



Δημιουργώ παιχνίδια στο

ω παιχνίδια
SCRATCH



Η συγγραφική ομάδα: Αλεξοπούλου Γ., Ανδρονικίδης Α., Αποστολίδης Α., Ασβεστόπουλος Β.-Α., Βαζάκας Α., Βουμβουράκης Ε., Γαληνός Δ., Γεωργίου Δ., Γιάκας Α., Γούσης Μ., Γρηγορόπουλος Α., Δημακόπουλος Θ., Ευθυμιάδης Δ., Ιακώβου Α., Καζάνης Κ., Καϊσίδης Β., Καλλίτης Μ., Καλογήρου Χ., Καματσέλου Β., Καραβασίλης Α., Καραμήτρου Ε. Καραμπέρης Σ., Καρνιαβούρας Γ., Κατσαρού Φ., Κατσιγιαννάκης Ε., Κεχαγιάς Σ., Κοσμανός Δ., Κότσιρα Μ., Κούλελη Μ., Κουμής Μ., Κούρφαλη Α., Κρόκου Γ., Κωνσταντινίδου Θ., Λαρίσης Ν., Λέκκας Κ., Μανιάτη Ο., Μεϊκόπουλος Ο.-Ι., Μήττας Α., Μιτσιγιώργης Π., Μπαλάση Π., Μπέσσας Κ., Νάκος Β., Νασιάκου Α., Νασόπουλος Γ., Νόνα Μ., Παλιαρούτης Γ.-Ι., Πάλλας Γ., Παπαδάκης Ι., Παπαδόπουλος Δ., Παπακωνσταντίνου Α., Πλωμαρίτου Κ., Πουλαράκης Κ., Προδρόμου Μ., Ροδίτη Α., Ρουμελιώτου Κ., Σκορδίλης Ε., Σπανάκης Ι., Σπέντζας Θ., Τζεβελεκίδης Κ., Τζήκας Ι., Τολίκας Γ., Τοπαλούδης Α., Τσακμάλης Κ., Τσιρογιάννη-κουλουμπρίδου Ι., Τζουμάνης Π., Χαλκιάς Π., Χασακός Σ., Χατζηνώτας Γ., Ψάλτης Α.

Επιμέλεια κειμένου - συντονισμός: Παλαιγεωργίου Γ.

Τμήμα Μηχανικών Η/Υ, Δικτύων και Τηλεπικοινωνιών
Πανεπιστήμιο Θεσσαλίας
Δεκέμβριος 2010



Το περιεχόμενο του παρόντος βιβλίου υπάγεται σε Άδεια Χρήσης [Creative Commons - Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/)

Αγαπητοί μαθητές

Το παρόν βιβλίο είναι γραμμένο από φοιτητές και απευθύνεται σε εσάς, στους μαθητές των ελληνικών σχολείων που ψάχνουν τρόπους για να ανακαλύψουν τη δημιουργικότητά τους, να διασκεδάσουν μαθαίνοντας, να δημιουργήσουν τα δικά τους αυθεντικά προϊόντα, να ανακαλύψουν τον κόσμο της πληροφορικής, να «κατασκευάσουν» τη δική τους τεχνολογία, τα δικά τους παιχνίδια!

Οπλίστε τον εαυτό σας με υπομονή, μεθοδικότητα και μεράκι και παίζετε με τα παραδείγματα του τελευταίου κεφαλαίου για να δείτε πόσο εντυπωσιακές είναι οι δεξιότητες που θα αποκτήσετε στο τέλος της μελέτης σας. Συμμαθητές σας από όλο τον κόσμο τα έχουν καταφέρει περίφημα, πολλές φορές εντυπωσιάζοντας ακόμη και τους πιο έμπειρους προγραμματιστές μεγάλων εταιριών λογισμικού. Είναι στο χέρι σας (κυριολεκτικά και μεταφορικά)!

Καλή διασκέδαση

Αγαπητοί καθηγητές

Το βιβλίο «*Δημιουργώ παιχνίδια στο Scratch*» δημιουργήθηκε στα πλαίσια του μαθήματος «Διδακτική της Πληροφορικής II» που πραγματοποιήθηκε στο εαρινό εξάμηνο του ακαδημαϊκού έτους 2009-2010, στο τμήμα Μηχανικών Η/Υ, Δικτύων και Τηλεπικοινωνιών, του πανεπιστημίου Θεσσαλίας. Τι φοιτητές! Το μεράκι τους ανεξάντλητο, η δημιουργικότητά τους προκλητική, η συναδελφικότητά τους απαράμιλλη, πάντα όμως με την προϋπόθεση να ανταποκρίνονται διαρκώς στις πολλές, πολύπλοκες και ετερογενείς προσδοκίες τους. Ισχύουν για τη συντριπτική πλειοψηφία της συγγραφικής ομάδας.

Οι φοιτητές ανέλαβαν πρωταγωνιστικό ρόλο στη δημιουργία ενός εκπαιδευτικού προϊόντος που ελπίζω ότι θα είναι χρήσιμο στην εκπαιδευτική κοινότητα και ίσως ωθήσει αριετούς μαθητές να μυηθούν στο κόσμο του προγραμματισμού από μια μοναδική σκοπιά, αυτή του Scratch. Επειδή όμως η διδασκαλία πρέπει να ενσωματώνει την προσωπική σας επιστημολογική θεώρηση, τις δικές σας διδακτικές προσεγγίσεις αλλά και τις ιδιαίτερες συνθήκες που αφορούν τους μαθητές σας, το βιβλίο αυτό διανέμεται στο διαδίκτυο με άδεια (cc) Creative Commons. Μπορείτε, δηλαδή, να δημιουργήσετε και να διανείμετε τις δικές σας εκδοχές του βιβλίου, με διαφορετικά κεφάλαια, διαφορετικά παραδείγματα, διαφορετικές περιγραφές. Με μεγάλη μας χαρά θα φιλοξενήσουμε στον ιστοχώρο του βιβλίου μας, τα δικά σας δημιουργήματα.

Ήταν η πρώτη προσπάθεια και συνεπώς χαρακτηρίζεται από αριετές αδυναμίες. Ελπίζω να υπάρχει επόμενη και να γίνει και με τη δική σας συμμετοχή!

Καλή δύναμη στο δύσκολο και μοναχικό σας έργο

[Γ.Παλαιεωργίου](#)

Περιεχόμενα

Κεφάλαιο 1^ο: Εισαγωγή

1.1 Τι είναι ο προγραμματισμός;	7
1.2 Τι χρειάζεται για να γράψουμε ένα πρόγραμμα;	8
1.3 Οφέλη προγραμματισμού	9
1.4 Scratch...ing	9
Περίληψη	11
Ερωτήσεις	11
Δραστηριότητες	11

Κεφάλαιο 2^ο: Πρώτη επαφή

2.1 Εγκατάσταση του Scratch	14
2.2 Παραδείγματα προγραμμάτων	16
2.3 Αντικείμενα, συμπεριφορές, γεγονότα	18
2.4 Αναγνώριση αντικειμένων	20
Περίληψη	21
Ερωτήσεις	21
Δραστηριότητες	21

Κεφάλαιο 3^ο: Το περιβάλλον

3.1 Γρήγορη ξενάγηση στο Περιβάλλον Εργασίας	23
3.2 Πρωταγωνιστές στα έργα μας ...τα αντικείμενα!	24
3.4 Η σκηνή...είναι δική σας	27
3.5 Εντολές	27
Περίληψη	29
Ερωτήσεις	29

Κεφάλαιο 4^ο: Αντικείμενα

4.1 Το σκηνικό του Scratch	31
4.2 Σχεδιάζοντας ένα σκηνικό	31
4.2.1 Δημιουργώντας το δικό μας σκηνικό	31
4.2.2 Εισαγωγή έτοιμου σκηνικού	35
4.2.3 Βγάζοντας φωτογραφίες για σκηνικό	36
4.2.4 Ακόμη ένα παράδειγμα δημιουργίας σκηνικού	36
4.3 Σχεδιάζοντας αντικείμενα	38
4.3.1 Εισαγωγή στο περιβάλλον των αντικειμένων	38
4.3.2 Δημιουργία νέων αντικειμένων	38
4.3.3 Ενδυμασίες	41
4.4 Ήχοι	44
4.4.1 Χρήση ήχων	44
Περίληψη	46
Ερωτήσεις	47

Δραστηριότητες	47
----------------	----

Κεφάλαιο 5^ο: Κίνηση

5.1 Σχετική κίνηση αντικειμένων	49
5.1.1 Τα πρώτα μου βήματα	49
5.1.2 Απόλυτη κατεύθυνση	50
5.1.3 Σχετική κατεύθυνση	51
5.1.4 Σχετική κατεύθυνση προς αντικείμενο	52
5.1.5 Μεταβολή!	53
5.1.6 Έλεγχος δυνατοτήτων κίνησης του αντικείμενου	53
5.2 Κίνηση με συντεταγμένες	53
5.2.1 Ραντεβού στο (x, y)	55
5.2.2 Μετακίνηση με σχετικές συντεταγμένες	56
5.2.3 Μετακίνηση προς αντικείμενο	56
5.3 Ομαλή κίνηση	57
5.4 "Καβουράκης και Αστεριάδης σε υποβρύχιες βόλτες"	57
Ερωτήσεις	60
Δραστηριότητες	60

Κεφάλαιο 6^ο: Ζωγραφική

6.1 Ζωγραφική	63
6.2 Απλά ζωγράφισε	63
6.2.1 Κατέβασε πένα	63
6.2.2 Σήκωσε πένα	63
6.2.3 Σφραγίδα	63
6.2.4 Καθάρισε	64
6.2.5 Παραδείγματα	64
6.3 Χρώμα, σκιά και μέγεθος πέννας	67
6.3.1 Ορισμός του χρώματος	67
6.3.2 Ορισμός της σκιάς	67
6.3.3 Μέγεθος της πέννας	68
6.4 Παράδειγμα	68
Περίληψη	69
Ερωτήσεις	69
Δραστηριότητες	69

Κεφάλαιο 7^ο: Όψεις

7.1 Αλλάζοντας όψεις εύκολα, γρήγορα και προσεκτικά	72
7.1.1 Ξεκινάμε να περπατάμε !!!	72
7.1.2 Χόρεψε με τη "σειρά"	73
7.1.3 Είμαι πιο "Μπροστά" απ' όλους... με τη βοήθεια εντολών	74
7.2 - Σκέψου πριν μιλήσεις	75

7.2.1 Παράδειγμα : Ενοχλητικός τύπος.....	76	Δραστηριότητες.....	105
7.2.2 Παράδειγμα : Scratch Rules	76	Κεφάλαιο 10^ο: Η επανάληψη	
7.2.3 Παράδειγμα : Δοκιμαστήριο	77	10.1 Εισαγωγή στην επανάληψη	107
7.3 Πρώτο πλάνο και στο σωστό μέγεθος.....	78	10.2 Θέλω να κάνω κάτι για πάντα..!	108
7.3.1 Παράδειγμα : Αστερίξ.....	79	10.3 Η εντολή «επανάλαβε Χ».....	110
7.3.2 Δε μπορούμε να αλλάξουμε το σκηνικό δυναμικά; 80		10.4 Άλλα παραδείγματα.....	112
7.3.3 Παράδειγμα – Υπόβαθρα.....	80	10.4.1 Ένας απλός ανεμόμυλος	112
7.4 Δυναμικά εφέ	81	10.4.2 Ρολόι!.....	112
7.4.1 Οι εντολές.....	81	Περίληψη	113
7.4.2 Εφέ χρώμα	81	Ερωτήσεις	114
7.4.3 Εφέ Εικονοστοιχειοποίηση	82	Δραστηριότητες.....	114
7.4.4 – Εφέ ψηφιδωτό	83	Κεφάλαιο 11^ο: Έλεγχος	
7.4.5 Εφέ φάντασμα.....	83	11.1 Εισαγωγή στον έλεγχο.	118
7.4.5 Εφέ φωτεινότητα	83	11.2 Οι εντολές ελέγχου του scratch.....	118
7.4.6 Εφέ στροβίλιση	83	11.2.1 Η εντολή "Εάν..."	118
7.4.7 Εφέ μάτι ψαριού.....	84	11.2.2 Η εντολή "Εάν..." και αισθητήρες	118
Περίληψη	85	11.2.3 Η εντολή "Εάν..." και οι τελεστές συνθηκών	120
Ερωτήσεις.....	85	11.2.2 Η εντολή "Εάν...Αλλιώς.."	122
Δραστηριότητες.....	85	11.3 Παραδείγματα	123
Κεφάλαιο 8^ο: Ήχοι		Περίληψη	125
8.1 Βασικές εντολές ήχου στο Scratch	87	Ερωτήσεις	125
8.1.1 Αναπαραγωγή και stop/pause έτοιμων ηχητικών clips	87	Δραστηριότητες.....	125
8.1.2 Ρύθμιση της έντασης.....	88	Κεφάλαιο 12^ο: Νέες επαναλήψεις	
8.2: Προγραμματίζοντας ήχους	89	12.1. Εντολή «για πάντα εάν...»	128
8.2.1: Παρουσίαση τυμπάνων και ρυθμού.....	89	12.2 Εντολή «επανάλαβε ώσπου».....	130
8.2.2 Συνθέτοντας τη μελωδία	90	12.3 Εντολή "περίμενε ώσπου..."	131
8.2.3 White Stripes.....	91	12.4 Συνδυασμός εντολών επανάληψης.....	133
8.3 Δυο ακόμη παραδείγματα	91	12.5 Τελεστές, συνθήκες και άλλα!	136
8.3.1 Ο Μπούκλας παίζει και μουσική	91	Περίληψη	138
8.3.2 Χορεύοντας μόνος σου.....	93	Ερωτήσεις	139
Σύνοψη.....	95	Δραστηριότητες.....	140
Ερωτήσεις.....	96	Κεφάλαιο 13^ο: Επικοινωνία	
Δραστηριότητες.....	96	13.1 Εντολές «μετάδωσε...» και «όταν λάβω...».....	142
Κεφάλαιο 9^ο: Αλληλεπίδραση		13.2 Εντολή "μετάδωσε... και περίμενε"	144
9.1 Γεγονότα Σφάλμα! Δεν έχει οριστεί σελιδοδείκτης.		13.3 Παραδείγματα	145
9.1.1 Εντολή «Όταν το πλήκτρο ... πατηθεί».....	99	Περίληψη	150
9.2 Εντολή «Ρώτησε και Περίμενε»	102	Ερωτήσεις	150
Περίληψη	104	Δραστηριότητες.....	151
Ερωτήσεις.....	105	Κεφάλαιο 14^ο: Μεταβλητές-Λίστες	
		14.1. Εισαγωγή στην έννοια των μεταβλητών	153

14.1.1 Οι μεταβλητές	153	Κυνηγός.....	186
14.1.2 Παραδείγματα εφαρμογών με μεταβλητές	154	Σφαίρα.....	186
14.1.3 Γενικές παρατηρήσεις για τις μεταβλητές	156	Λαγός.....	186
14.2 Λίστες.....	157	Ζωές	188
14.2.1 Τι είναι οι λίστες;	157	Αντικείμενο Τέλος.....	189
14.2.2 Η λίστα στο Scratch	157	Προτεινόμενες αλλαγές στο παιχνίδι.....	189
14.2.3 Μεταβλητές vs. Λίστες	160		
14.3 Παραδείγματα.....	160		
14.3.1 Μεταβλητές	160		
14.3.2 Λίστες	162		
Περίληψη	169		
Ερωτήσεις.....	169		
Δραστηριότητες.....	170		
Κεφάλαιο 15^ο: Παιχνίδια			
15.1 Super Pong	175		
Αντικείμενα & Σκηνικό	172		
Αντικείμενο 1: Σκηνικό.....	172		
Αντικείμενο 2: Η ρακέτα.....	172		
Αντικείμενο 3: Οι μπάλες.....	173		
Προτεινόμενες αλλαγές στο παιχνίδι.....	174		
15.2 Ναρκοπέδιο!.....	175		
Αντικείμενα και Σκηνικό	175		
Αντικείμενα νάρκες	175		
Αντικείμενο πλοίο	176		
Προτεινόμενες αλλαγές στο παιχνίδι.....	176		
15.3 Σκοποβολή.....	177		
Αντικείμενα και Σκηνικό	177		
Σκηνικό.....	177		
Αντικείμενα-στόχοι	178		
Προτεινόμενες αλλαγές στο παιχνίδι.....	179		
15.4 Ο καρχαρίας.....	180		
Τα αντικείμενα	180		
Το σκηνικό.....	180		
Ψάρια και χταπόδι.....	180		
Ο καρχαρίας.....	182		
Προτεινόμενες αλλαγές στο παιχνίδι.....	184		
15.5 Καροτοκυνηγός.....	185		
Αντικείμενα	185		
Μεταβλητές	185		
Σκηνικό.....	185		

Κεφάλαιο 1: Προγραμματισμός

Σε αυτό το κεφάλαιο:

- 1.1 Τι είναι ο προγραμματισμός
- 1.2 Τι χρειάζεται για να δημιουργήσουμε ένα πρόγραμμα
- 1.3 Οφέλη από τον προγραμματισμό
- 1.4 Scratch...ing

«Πρώτα λύσε το πρόβλημα. Μετά γράψε τον κώδικα».

(John Johnson)



1.1 Τι είναι ο προγραμματισμός

Πολύ πιθανόν να είστε εξοικειωμένοι με τη νέα τεχνολογία, να γράφετε και να στέλνετε ηλεκτρονικά μηνύματα από τον υπολογιστή σας, να παίζετε online παιχνίδια και να αναζητάτε στο διαδικτυο υλικό για τα ενδιαφέροντά σας. Περιορίζονται όμως οι δυνατότητες που σας παρέχει η νέα τεχνολογία μόνο στις συγκεκριμένες ενέργειες; Έχετε σκεφτεί ποτέ να δημιουργήσετε τα δικά σας παιχνίδια, τις δικές σας εφαρμογές; Θα θέλατε από παθητικοί χρήστες των νέων τεχνολογιών να μετατραπείτε σε δημιουργοί νέου λογισμικού, παιχνιδιών και παρουσιάσεων; Έχετε αναρωτηθεί αν κάτι τέτοιο είναι εφικτό και πόσο δύσκολο είναι; Αυτό το βιβλίο θα σας βοηθήσει να φτιάξετε τα δικά σας προγράμματα, αξιοποιώντας τη φαντασία και τη δημιουργικότητά σας και θα σας πείσει (ελπίζουμε...) ότι η δημιουργία προγραμμάτων είναι μια εύκολη, δημιουργική και χρήσιμη διαδικασία (στην επόμενη εικόνα, τα γυαλιά δεν είναι απαραίτητα ☺).

Θα μάθω να προγραμματίζω τα δικά μου παιχνίδια!!!



Το να χρησιμοποιούμε τεχνολογία αλλά να μην μπορούμε να δημιουργήσουμε, μίπως μοιάζει με το να είμαστε σε θέση να διαβάζουμε αλλά να μη μπορούμε να γράψουμε;

Η τέχνη του να μπορούμε να γράφουμε τα δικά μας προγράμματα ονομάζεται προγραμματισμός. Ο ορισμός του προ-γραμ-ματισμού από την ελληνική Wikipedia:

«Το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος για την πραγματοποίηση εργασιών ή για την επίλυση ενός δεδομένου προβλήματος. Ο προγραμματισμός πε-

ριλαμβάνει επίσης τον έλεγχο του προγράμματος για την επαλήθευση της ακρίβειάς του, και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εργασίες που καθορίζονται στις προδιαγραφές του προγράμματος».

Γενικότερα ως προγραμματιστικό πρόβλημα θεωρούμε κάθε ζήτημα που τίθεται προς επίλυση, κάθε κατάσταση που μας απασχολεί, κάθε ηλεκτρονική συμπεριφορά που επιθυμούμε να επιδειχθεί από τον υπολογιστή μας. Μην ανησυχείτε, τα προβλήματα που θα εξετάσουμε σε αυτό το βιβλίο αφορούν κυρίως την κατασκευή παιχνιδιών! Ακόμη όμως και τα παιχνίδια, όπως όλα τα προβλήματα, έχουν τα δικά τους συγκεκριμένα δεδομένα και ζητούμενα. Π.χ. σε ένα παιχνίδι ράλι, θέλουμε, αν ο χρήστης πατά το δεξί βέλος του πληκτρολογίου, το αυτοκινητάκι μας να στρίβει προς τα δεξιά. Πριν λύσουμε λοιπόν οποιοδήποτε πρόβλημα ως προγραμματιστές, οφείλουμε να κατανοήσουμε σε βάθος αυτά τα δυο στοιχεία, τα δεδομένα και τα ζητούμενα.

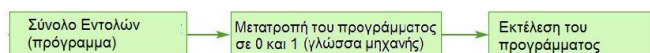
Τι σημαίνει όμως «λύνουμε ένα πρόβλημα», π.χ. δημιουργούμε ένα νέο παιχνίδι, ως προγραμματιστές; Σημαίνει ότι θα πρέπει να δώσουμε συγκεκριμένες και ακριβείς οδηγίες στον υπολογιστή για τον τρόπο με τον οποίο θα πρέπει να λειτουργεί. Πως θα φαίνεται η οθόνη μας; Πως θα αντιδρούν οι πρωταγωνιστές του παιχνιδιού στα διαφορετικά συμβάντα; Τι θα συμβαίνει όταν ο χρήστης χάσει μια ζωή; Τι θα γίνει αν η σφαίρα αγγίξει ένα ζωάκι; Η περιγραφή της λύσης, δηλαδή η διατύπωση των σωστών οδηγιών για την επίλυση του προβλήματος, περιέχει συχνά δυσκολίες.

Το κυριότερο συστατικό ενός προγράμματος είναι οι εντολές. Μπορούμε να φανταστούμε τις εντολές σαν οδηγίες του προγραμματιστή προς τον υπολογιστή για να κάνει κάτι (π.χ. να δημιουργήσει μια γραμμή, να μετακινήσει έναν στρατιώτη, να δείξει μια εικόνα στην οθόνη). Μία ακολουθία εντολών συνιστά το πρόγραμμα. Τα προγράμματα που χρησιμοποιείτε καθημερινά, όπως το Tetris ή ο ναρκαλιευτής, αποτελούνται από μια σειρά εντολών, μια σειρά οδηγιών για το πώς πρέπει να συμπεριφέρονται.

Τι μορφή όμως έχουν οι εντολές αυτές; Πολλές και διαφορετικές. Υπάρχουν δηλαδή πολλές «γλώσσες προγραμματισμού»

με τις οποίες μπορείτε να δημιουργήσετε τα παιχνίδια σας. Κάθε γλώσσα προγραμματισμού έχει το δικό της σύνολο διαθέσιμων εντολών και τους δικούς της κανόνες. Όπως θυμάστε όμως από την πρώτη γυμνασίου που είχαμε συζητήσει ότι ο υπολογιστής καταλαβαίνει μόνο 01011101..; Άρα, αφού γράψουμε το πρόγραμμά μας σε οποιαδήποτε γλώσσα, μετά απαιτείται ένα στάδιο το οποίο ονομάζεται μεταγλώττιση. Όπως αποκαλύπτει η ίδια η λέξη, κατά τη μεταγλώττιση οι εντολές μας μετατρέπονται σε μια μορφή που είναι κατανοητή από τον υπολογιστή, τη γνωστή μορφή 00101010... (γλώσσα μηχανής). Μετά από αυτό το βήμα, μπορούμε να τρέξουμε το πρόγραμμά μας (ή αλλιώς να «**εκτελέσουμε**» το πρόγραμμά μας) έτσι ώστε να ελέγξουμε αν όντως κάνει αυτό που αρχικά σχεδιάσαμε.

Σχηματικά, η παραπάνω διαδικασία απεικονίζεται στο παρακάτω σχήμα:



Άρα μπορούν να υπάρχουν πολλές γλώσσες προγραμματισμού αρκεί να περιέχουν έναν «μεταγλωττιστή». Ποια γλώσσα προγραμματισμού θα ήταν καλύτερα να μάθω; Γενικά, όσο και αν διαφέρουν οι διάφορες γλώσσες προγραμματισμού, στη βάση τους υπάρχουν πολλά στοιχεία που είναι κοινά, αφού οι εργασίες που μπορεί να κάνει ο υπολογιστής μας είναι συγκεκριμένες. Αυτό -ευτυχώς- συνεπάγεται ότι αν μάθουμε μία γλώσσα προγραμματισμού είναι σχετικά εύκολο να προχωρήσουμε και στην εκμάθηση κάποιας άλλης γλώσσας.

1.2 Τι χρειάζεται για να γράψουμε ένα πρόγραμμα;

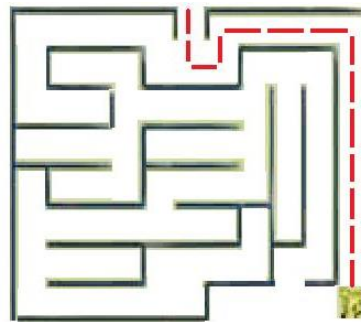
Ήρθε όμως η ώρα να δούμε με περισσότερη λεπτομέρεια τι χρειάζεται για να γράψουμε ένα πρόγραμμα.

Τι οδηγίες θα δίνουμε στο μικρό μας αδερφάκι για το πως θα λύσει το γνωστό σε όλους μας παιχνίδι του λαβυρίνθου, που φαίνεται στην εικόνα 1.3;

Σκοπός του παιχνιδιού είναι να φτάσουμε στην έξοδο του λαβυρίνθου μετακινώντας το μικρό κίτρινο ήρωα. Ο ήρωας δε μπορεί να περάσει μέσα από τους τοίχους. Ένα απλό σύνολο οδηγιών περιλαμβάνει:

1. *Κίνησε τον ήρωα επτά βήματα προς τα πάνω*
2. *Κίνησε τον ήρωα τρία βήματα προς τα αριστερά*
3. *Κίνησε τον ήρωα ένα βήμα προς τα κάτω*
4. *Κίνησε τον ήρωα ένα βήμα προς τα αριστερά*
5. *Κίνησε τον ήρωα δύο βήματα προς τα πάνω*

Με τον ίδιο τρόπο θα πρέπει να δίνουμε τις εντολές στον υπολογιστή για να του δώσουμε να καταλάβει τι θέλουμε να κάνει.



Αλλά το παραπάνω παράδειγμα ήταν πολύ απλό και στην ουσία δε θα μπορούσε να εφαρμοστεί σε οποιοδήποτε λαβύρινθο! Ας ασχοληθούμε με ένα διαφορετικό παιχνίδι, το γνωστό μας packman. Ο κεντρικός ήρωας μας θα πρέπει να κινείται μέσα σε ένα λαβύρινθο. Κατά μήκος του λαβυρίνθου υπάρχουν μικρά φρούτα, τα οποία επιδιώκει να τρώει για να ανεβάζει το σκορ του. Όμως υπάρχουν και μικρά φαντασματάκια που τον κυνηγούν και όταν πέφτει πάνω σε αυτά, χάνει τη ζωή του.



Πως δημιουργούμε ένα αντίστοιχο πρόγραμμα ως προγραμματιστές; Σε πρώτο στάδιο πρέπει να μελετήσουμε προσεκτικά το πρόβλημα και να εντοπίσουμε τα δεδομένα του και στη συνέχεια τα ζητούμενά του. Ποιοι είναι οι κανόνες του παιχνιδιού; Ποιες είναι οι συμπεριφορές που θέλουμε να επιδεικνύουν όλα τα στοιχεία του; Αμέσως μετά πρέπει να κάνουμε μια **αποδόμηση του προβλήματος σε μικρότερα υποπροβλήματα**, τα οποία είναι πιο εύκολο να λυθούν. Επιμέρους προβλήματα στο packman μπορούν να θεωρηθούν τα εξής:

- ✓ πρέπει να αποφασίσουμε για το ποιο είναι το **σκηνικό** μας, δηλαδή πρέπει να σχεδιάσουμε το λαβύρινθο.
- ✓ πρέπει να σχεδιάσουμε τους **χαρακτήρες** μας που δεν είναι άλλοι από τον packman, τα φαντασματάκια που τον κυνηγούν και τα φρούτάκια που τρώει ο πάκμαν.
- ✓ πρέπει να προσδιορίσουμε ποιες είναι οι **συμπεριφορές** του κάθε αντικειμένου; Ο packman κινείται όταν χρησιμοποιούμε τα βελάκια του πληκτρολογίου. Τα φαντασματάκια κινούνται διαρκώς τυχαία σε όλες τις δυνατές διαδρομές μέσα στο λαβύρινθο. Τα φρούτα εμφανίζονται με τυχαίο τρόπο μέσα στο λαβύρινθο.
- ✓ ποιες οι **αλληλεπιδράσεις** μεταξύ των χαρακτήρων μας; Όταν ο χαρακτήρας μας ακουμπήσει ένα φαντασματάκι, ο παίκτης χάνει μια ζωή και το αντικείμενο μας ξεκινά ξανά από το σημείο εκκίνησης. Όταν ακουμπήσει ένα φρούτο, τότε κερδίζει πόντους και το φρούτο εξαφανίζεται κτλ.



Με έναν αντίστοιχο τρόπο, απλό και συστηματικό, συνεχίζουμε να αποδομούμε λογικά το πρόβλημα σε μικρότερα προβλήματα έτσι ώστε στο τέλος να γράψουμε πολλά μικρά και εύκολα σύνολα εντολών-οδηγιών που όλα μαζί συνιστούν το παιχνίδι. Φυσικά, δεν πρέπει να ξεχνάμε ότι ένα πρόβλημα πολλές φορές μπορεί να λυθεί με πολλούς διαφορετικούς τρόπους.



Τι θα συμβεί τώρα; Μα φυσικά, ό,τι έχει προκαθοριστεί στο πρόγραμμα!

Η αλληλεπίδρασή μας με τους υπολογιστές πολλές φορές μας οδηγεί αναπόφευκτα στο συμπέρασμα ότι ο υπολογιστής εμπεριέχει μια μορφή ευφυΐας που του επιτρέπει να ανταποκρίνεται στις απαιτήσεις κάθε κατάστασης του παιχνιδιού. Αυτό όμως που γίνεται στην πραγματικότητα είναι ότι ο προγραμματιστής του παιχνιδιού (ή καλύτερα μια ομάδα προγραμματιστών) έχει προκαθορίσει ποια ενέργεια θα εκτελέσει ο υπολογιστής σε κάθε επιλογή του χρήστη. Για παράδειγμα, αν χρήστης πατήσει το βελάκι προς τα αριστερά, ο ήρωας μας πηγαίνει αριστερά και, αντίστοιχα, αν ο εχθρικός ήρωας α-κουμπήσει τον ήρωα του παιχνιδιού μειώνεται κατά ένα το πλήθος των ζώνων.

1.3 Οφέλη προγραμματισμού

Τι κερδίζουμε όμως μαθαίνοντας προγραμματισμό; Γιατί να μάθουμε προγραμματισμό; Καταρχάς, σε αυτό το βιβλίο θα μάθουμε προγραμματισμό για να δημιουργούμε τα δικά μας παιχνίδια και άρα προβλέπεται πολύ διασκέδαση! Επιπλέον όμως μπορούμε να σκεφτούμε τον προγραμματισμό σαν ένα καμβά πάνω στον οποίο μας δίνεται η δυνατότητα να φτιάχνουμε δικές μας δημιουργίες, να ζωγραφίζουμε και να δημιουργούμε εφαρμογές ανάλογα με τις προσωπικές μας προτιμήσεις και ανάγκες. Για παράδειγμα, μπορούμε να δημιουργήσουμε ένα ηλεκτρονικό ημερολόγιο που μας βοηθά στην οργάνωση της καθημερινότητάς μας ή ένα πρόγραμμα που εκτελεί μουσικές συν-

θέσεις και μας χαλαρώνει από τις υποχρεώσεις μας. Ο προγραμματισμός μπορεί να συνεισφέρει πρακτικά στην καθημερινή μας διασκέδαση και ψυχαγωγία.

Προγραμμάσέ με και θα σου κάνω τη ζωή πιο εύκολη!



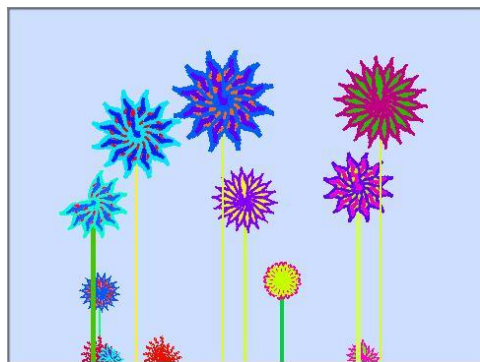
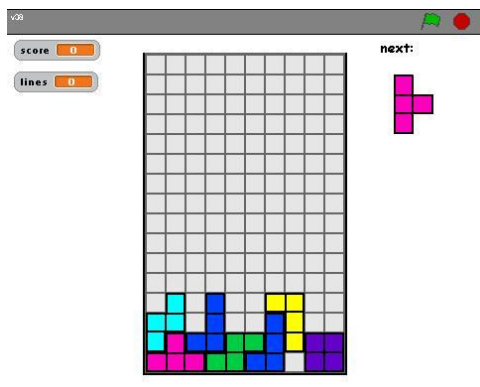
Ταυτόχρονα όμως, ο τρόπος σκέψης με τον οποίο θα εξοικειωθούμε, είναι πολύ πιθανόν να μας κάνει να σκεφτόμαστε πιο μεθοδικά, να λύνουμε πιο αποδοτικά τα καθημερινά σας προβλήματα. Η ενασχόληση με τη μεθοδική μελέτη, επίλυση και δοκιμή προβλημάτων και λύσεων, θα μας βοηθήσει να σκεφτόμαστε πιο γρήγορα, πιο έξυπνα, πιο συστηματικά. Θετικές επιδράσεις του προγραμματισμού έχουν βρεθεί ακόμη και στον τομέα των μαθηματικών! Φανταζόμαστε ότι όλοι θέλετε να βελτιώσετε την ταχύτητα και αποτελεσματικότητα της σκέψης σας.

Τέλος, η δημιουργία παιχνιδιών και εφαρμογών μέσω του προγραμματισμού, θα σας κάνει να συνειδητοποιήσετε πως όλες αυτές οι εφαρμογές με τις οποίες ασχολούμαστε καθημερινά δε δουλεύουν με ένα μαγικό τρόπο, αλλά αξιοποιούν απλές εντολές που καθορίζουν τη συμπεριφορά τους. Κατά τη μελέτη του βιβλίου θα πρέπει να πάψουμε να φοβόμαστε τους υπολογιστές. Θα διαπιστώσετε ότι και εσείς μπορείτε να δημιουργήσετε τεχνολογία! Συμμάθητές σας από όλο τον κόσμο τα έχουν καταφέρει περίφημα πολλές φορές εντυπωσιάζοντας ακόμη και τους πιο έμπειρους προγραμματιστές μεγάλων εταιριών λογισμικού.

1.4 Scratch...ing

Εφόσον συζητήσαμε για την χρησιμότητα του προγραμματισμού στην καθημερινότητα σας, έφτασε η στιγμή να γνωρίσετε ένα τρόπο για να γράφετε τα δικά σας προγράμματα. Ο τρόπος αυτός είναι το προγραμματιστικό περιβάλλον Scratch. Τί είναι το Scratch;

Το Scratch είναι μία νέα γλώσσα προγραμματισμού με την οποία μπορούμε να φτιάχνουμε τις δικές μας διαδραστικές ιστορίες, τα δικά μας παιχνίδια εύκολα και γρήγορα, ενώ παράλληλα θα συζητάμε για βασικές αρχές του προγραμματισμού. Με αυτή τη πλατφόρμα προγραμματισμού θα μπορέσουμε να φτιάξουμε το δικό μας tetris, packman ή το δικό μας κήπο, όπως βλέπουμε στις παρακάτω εικόνες.



Επίσης θα δημιουργούμε τα δικά μας κινούμενα σχέδια με διάλογους της επιλογής μας, καθώς και θα μοντελοποιούμε προβλήματα φυσικής όπως για παράδειγμα την κίνηση των πλανητών γύρω από τον ήλιο.



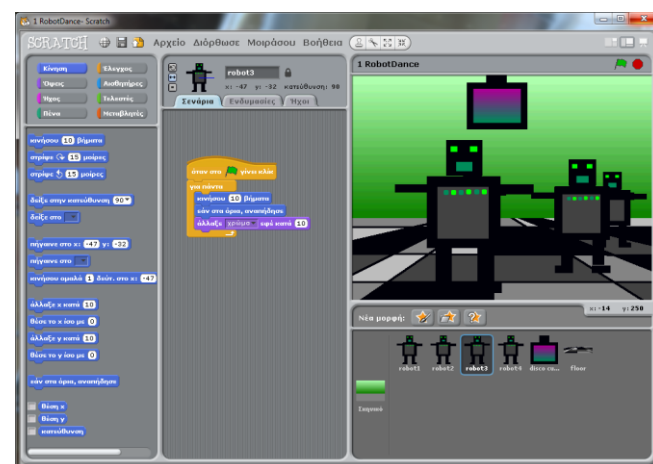
Ας δούμε όμως πρώτα μερικά ιστορικά στοιχεία για το Scratch. Αναπτύχθηκε από το Lifelong Kindergarten group στο MIT (όλο μας έχουμε ακούσει για το τρομερό ερευνητικό εργαστήριο) με επικεφαλή τον Mitchel Resnick και πρωτοεμφανίστηκε το καλοκαίρι του 2007, είναι δηλαδή σχετικά καινούριο περιβάλλον. Το λογισμικό διανέμεται δωρεάν για διαφορετικά λειτουργικά συστήματα (Windows, Mac OS X ή Linux) και η εγκατάστασή του

είναι πολύ απλή. Σήμερα χρησιμοποιείται ευρέως για τη διδασκαλία του προγραμματισμού, ενώ η διάδοσή του είναι ταχύτατη. Ενδεικτικά μπορούμε να αναφέρουμε ότι στην ιστοσελίδα του Scratch (<http://scratch.mit.edu/>) υπάρχουν γύρω στα 700.000 εγγεγραμμένα μέλη και γύρω στους 200.000 προγραμματιστές που δημοσιεύουν τα προγράμματά τους στον συγκεκριμένο ιστόχωρο!

Ενδιαφέρον είναι ότι το Scratch πήρε το όνομά του από την τεχνική των DJ's (scratching). Το βασικό χαρακτηριστικό της τεχνικής των DJ's είναι η επαναχρησιμοποίηση των μουσικών κομματιών. Αντίστοιχα στο Scratch όλα τα αντικείμενα, γραφικά, ήχοι, και κείμενα μπορούν εύκολα να εισαχθούν σε ένα νέο πρόγραμμα και να συνδυαστούν με ποικίλους τρόπους για την παραγωγή ενός προγράμματος, κάτι το οποίο δίνει κίνητρο για περαιτέρω ενασχόληση με αυτό.



Στην παρακάτω εικόνα εμφανίζεται το περιβάλλον του Scratch ενώ έχουμε ανοίξει ένα από τα έτοιμα παραδείγματα που μας προσφέρει.



Οι εντολές, που όπως είπαμε είναι τα δομικά συστατικά ενός προγράμματος, αναπαριστώνται ως **τουβλάκια**. Τα τουβλάκια, που από εδώ και στο εξής θα αποκαλούμε **εντολές**, συνθέτονται

σε στοίβες, οι οποίες συνιστούν τα **σενάρια ενεργειών**,

Παρατηρούμε πως ενώνονται τα τουβλάκια-εντολές και σχηματίζουν ένα μπλοκ, κατά τρόπο όμοιο με αυτόν που θα ενώναμε τουβλάκια Lego.



Όλες αυτές οι στοίβες από τουβλάκια δημιουργούν το πρόγραμμά μας. Ποια είναι όμως τα πλεονεκτήματα της χρήσης του Scratch;

Οι δημιουργοί του Scratch για να μας διευκολύνουν σχεδίασαν τις εντολές κατά τέτοιο τρόπο ώστε να μπορούν να συνδεθούν μεταξύ τους μόνο όταν ο συνδυασμός τους έχει νόημα. Επιπλέον, οι εντολές που μπορούμε να χρησιμοποιήσουμε είναι εκ των προτέρων γνωστές και εντοπίζονται εύκολα ανοίγοντας καθεμιά από τις διαθέσιμες **παλέτες εντολών** (βρίσκονται στα αριστερά της οθόνης του Scratch). Τα ονόματα των εντολών έχουν επιλεγεί ώστε να μπορούμε εύκολα να καταλάβουμε τι κάνει μία εντολή. Τέλος, το Scratch μας δίνει τη δυνατότητα να εξετάζουμε πολύ γρήγορα και εύκολα τα αποτελέσματα οποιασδήποτε εντολής. Αρκεί να πατήσουμε διπλό κλικ πάνω της (ακόμη και μέσα στην παλέτα). Τα ουσιαστικά πλεονεκτήματα του Scratch όμως θα τα συνειδητοποιήσετε στα κεφάλαια που ακολουθούν.

Περίληψη

Στο κεφάλαιο αυτό συζητήσαμε συνοπτικά για τον προγραμματισμό, μία έννοια που δεν πρέπει να μας φαίνεται άγνωστη, καθώς είτε το αντιλαμβανόμαστε είτε όχι, ο προγραμματισμός υπάρχει γύρω μας. Με το Scratch θα μπορέσουμε σχετικά εύκολα και γρήγορα να αναπτύξουμε μια πληθώρα διαδραστικών εφαρμογών χρησιμοποιώντας την ευρεία γκάμα δυνατοτήτων που μας παρέχει.

Στόχος αυτού του βιβλίου είναι η εξέταση των βασικών αρχών του προγραμματισμού με έναν όσο το δυνατόν πιο ευχάριστο και δημιουργικό τρόπο. Στα επόμενα δύο κεφάλαια, θα δούμε πως μπορούμε να κατεβάσουμε το Scratch και να το εγκαταστήσουμε στον υπολογιστή μας, καθώς και πώς μπορούμε να τρέξουμε έτοιμα παραδείγματα. Στο κεφάλαιο 3 γίνεται μία πρώτη περιγραφή του περιβάλλοντος εργασίας. Αυτά τα κεφάλαια είναι κυρίως εισαγωγικά.

Όσον αφορά το δεύτερο μέρος του βιβλίου, αρχικά στο κεφάλαιο 4, θα μάθουμε να δημιουργούμε τους ήρωες των παιχνιδιών μας και στη συνέχεια στα κεφάλαια 5 έως 8 θα τους κάνουμε να κινούνται, να ζωγραφίζουν, να παίζουν μουσική και θέατρο.

Στο τρίτο και τελευταίο μέρος του βιβλίου, οι ήρωες μας θα γίνουν πιο «έξυπνοι» και θα αποκτήσουν περισσότερες δυνατότητες. Πώς θα το πετύχουμε αυτό; Θα τους κάνουμε να αντιλαμβάνονται καλύτερα το περιβάλλον τους και να ανταλλάσσουν μηνύματα μεταξύ τους, να επαναλαμβάνουν διαδικασίες, όπως άλλωστε κάνουμε και εμείς καθημερινά μεταξύ μας και να

εκτελούν ενέργειες μόνο εφόσον ισχύουν ορισμένες συνθήκες. Το τελευταίο κεφάλαιο του βιβλίου δε θα μπορούσε παρά να παρουσιάζει τη δημιουργία 6 παιχνιδιών!

Καλή διασκέδαση!

Ερωτήσεις

- 1) Περιγράψτε τι είναι προγραμματισμός και ποιά τα βασικά χαρακτηριστικά ενός προγράμματος.
- 2) Τι πιστεύετε ότι μπορεί να σας προσφέρει ο προγραμματισμός στην καθημερινότητά σας;
- 3) Γιατί πρέπει να κατανοούμε καλά ένα πρόβλημα πριν το επιλύσουμε;
- 4) Ποιά είναι τα στάδια επίλυσης προβλημάτων που πρέπει να ακολουθούμε;
- 5) Αναφέρετε καθημερινές σας δραστηριότητες στις οποίες πιθανότατα ο προγραμματιστικός τρόπος σκέψης μπορεί να σας βοηθήσει.
- 6) Περιγράψτε τι είδους εφαρμογές πιστεύετε ότι μπορείτε να δημιουργήσετε με το Scratch και ποιές από αυτές μπορείτε να τις χρησιμοποιήσετε στην καθημερινότητά σας;

Δραστηριότητες

- 1) Αναλύστε το πρόβλημα της δημιουργίας του παιχνιδιού «tetris».
- 2) Δώστε τη σειρά των βημάτων που πρέπει να ακολουθήσουμε για να ζωγραφίσουμε ένα ρόμβο χωρίς να χρειαστεί σηκώσουμε το στυλό από το χαρτί.
- 3) Που πιστεύετε ότι είναι το λάθος στις παρακάτω οδηγίες που δώσαμε στον υπολογιστή για να δημιουργήσει την αρχική εικόνα του παιχνιδιού "τριλιζα":

Σχεδίασε ένα κατακόρυφο ευθύγραμμο τμήμα (τμήμα 1)

Σχεδίασε ένα κατακόρυφο ευθύγραμμο τμήμα (τμήμα 2) παράλληλο στο τμήμα 1

Σχεδίασε ένα κατακόρυφο ευθύγραμμο τμήμα (τμήμα 3) παράλληλο στο τμήμα 2

Σχεδίασε ένα ευθύγραμμο τμήμα (τμήμα 4) κάθετο στο τμήμα 2

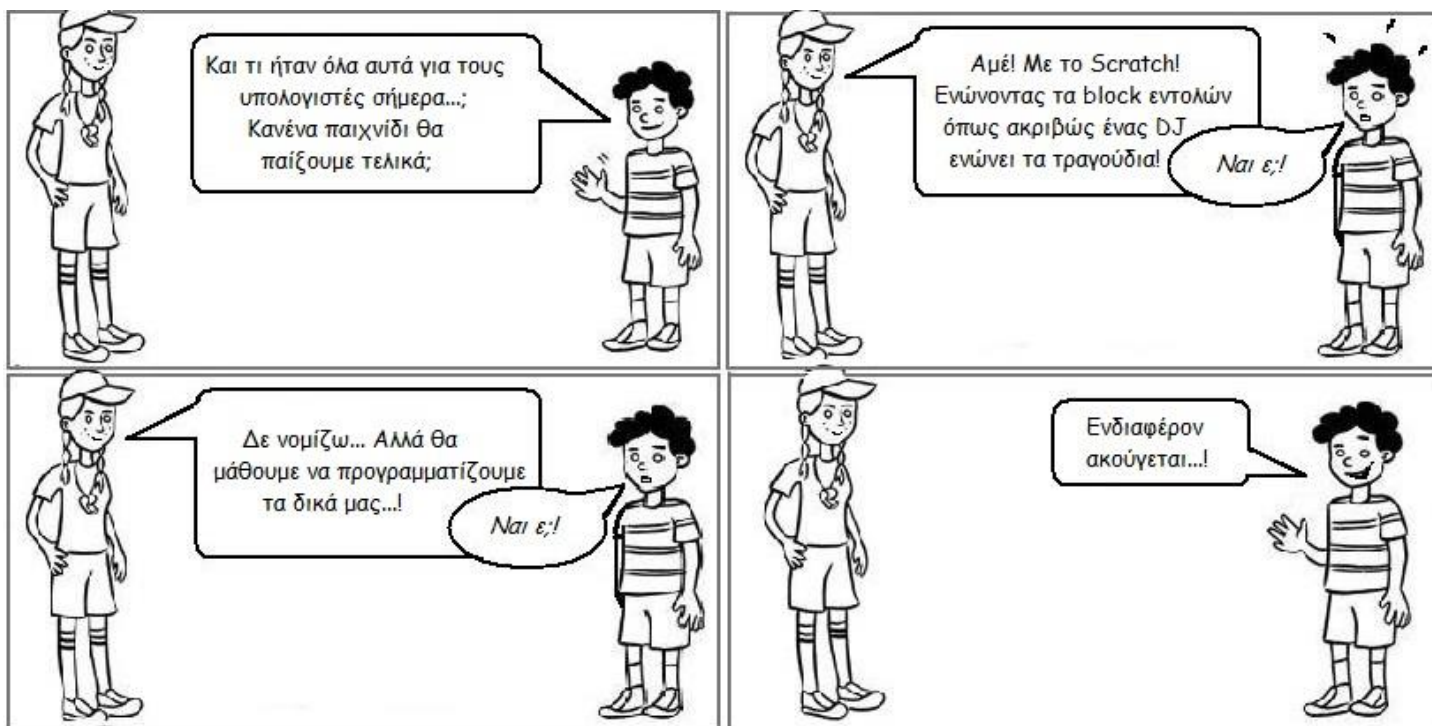
Σχεδίασε ένα ευθύγραμμο τμήμα (τμήμα 5) κάθετο στο τμήμα 3

Σχεδίασε ένα ευθύγραμμο τμήμα (τμήμα 6) κάθετο στο τμήμα 4

- 4) Ανακαλύψτε τις κρυμμένες λέξεις : προγραμματισμός, σύνθεση, μεταγλώττιση, κώδικας, αποδόμηση, έλεγχος, σφάλμα, γλώσσα, εκτέλεση, εντολές.

Π	Ε	Ρ	Τ	Υ	Θ	Ι	Σ	Υ	Ν	Θ	Ε	Σ	Η	Ι
Κ	Ρ	Ο	Π	Α	Σ	Δ	Φ	Γ	Η	Ξ	Ν	Κ	Λ	Ξ
Ζ	Ω	Ο	Χ	Ψ	Ω	Β	Ν	Μ	Ρ	Τ	Τ	Γ	Κ	Δ
Λ	Β	Δ	Γ	Λ	Ω	Σ	Σ	Α	Σ	Χ	Ο	Ν	Φ	Ψ
Α	Φ	Ξ	Ι	Ρ	Λ	Ω	Ρ	Σ	Φ	Α	Λ	Μ	Α	Υ
Ε	Φ	Γ	Η	Κ	Α	Ν	Ξ	Κ	Λ	Χ	Ε	Ρ	Κ	Ο
Ζ	Σ	Δ	Ε	Η	Α	Μ	Ν	Ξ	Ρ	Μ	Σ	Σ	Λ	Ψ

Ε	Ρ	Τ	Κ	Υ	Θ	Σ	Μ	Ι	Ο	Π	Λ	Κ	Ξ	Η
Η	Γ	Φ	Τ	Δ	Σ	Α	Β	Α	Μ	Β	Ω	Ψ	Χ	Ζ
Ρ	Η	Μ	Ε	Τ	Α	Γ	Λ	Ω	Τ	Τ	Ι	Σ	Η	Γ
Ε	Ρ	Τ	Λ	Υ	Θ	Ι	Ο	Π	Α	Ι	Δ	Φ	Γ	Ξ
Ζ	Ε	Λ	Ε	Γ	Χ	Ο	Σ	Ω	Β	Ν	Σ	Δ	Γ	Σ
Ε	Ρ	Τ	Σ	Υ	Θ	Ι	Ο	Π	Α	Δ	Γ	Μ	Φ	Γ
Ζ	Χ	Ψ	Η	Ω	Β	Ν	Μ	Ξ	Δ	Τ	Δ	Α	Ο	Β
Φ	Γ	Η	Α	Π	Ο	Δ	Ο	Μ	Η	Σ	Η	Λ	Κ	Σ



Κεφάλαιο 2: Πρώτη επαφή

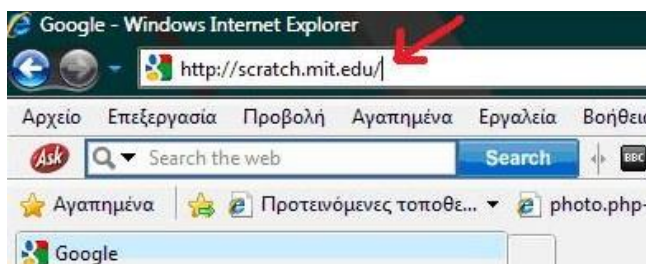
Σε αυτό το κεφάλαιο:

- 2.1 Εγκατάσταση του Scratch
- 2.2 Παραδείγματα προγραμμάτων
- 2.3 Αντικείμενα, συμπεριφορές, γεγονότα
- 2.4 Αναγνώριση αντικειμένων ...

«Θεωρητικά δεν υπάρχει καμία διαφορά μεταξύ θεωρίας και πράξης. Στην πράξη όμως, υπάρχει».
(Yogi Berra)

2.1 Εγκατάσταση του Scratch

Μετά την πρώτη μας επαφή με το περιβάλλον του Scratch, θα θέλετε να μάθετε πως θα το αποκτήσετε και εσείς στον προσωπικό σας υπολογιστή. Ας δούμε μαζί, λοιπόν, βήμα βήμα τον τρόπο για να το "κατεβάσουμε" και να το εγκαταστήσουμε, με τη βοήθεια εικόνων. Ανοίγουμε τον αγαπημένο μας φυλλομετρητή (π.χ. Internet Explorer) και στη μπάρα διευθύνσεων πληκτρολογούμε <http://scratch.mit.edu/>



Όπως μπορείτε να δείτε και μόνοι σας, στον ιστοχώρο αυτό, μπορείτε εύκολα να βρείτε πολυάριθμα παραδείγματα που έχουν δημιουργηθεί από έμπειρους χρήστες του λογισμικού, καθώς και πληροφορίες για όποια ερωτήματα σας δημιουργηθούν καθ' όλη τη διάρκεια της χρήσης του. Στη συνέχεια κάνουμε κλικ πάνω στην επιλογή «Download Scratch».



Έπειτα μας ζητείται προαιρετικά η συμπλήρωση μιας φόρμας με κάποια προσωπικά μας στοιχεία. Η φόρμα βοηθάει τους δημιουργούς του προγράμματος να γνωρίζουν τη διάδοση του Scratch. Μπορούμε να προσπεράσουμε τη φόρμα πατώντας στο σύνδεσμο «Συνεχίστε στο κατέβασμα του Scratch» - καλό

είναι να προσέχουμε όταν παρέχουμε τα προσωπικά μας δεδομένα στο διαδίκτυο.



Κατεβάστε το Scratch

Συμπληρώστε την προαιρετική φόρμα παρακάτω για να κατεβάσετε ανανεώσεις για το Scratch. Για να πάτε στη σε downloads, κάντε κλικ στο κουμπί που βρίσκεται στο τέλος της φόρμας.

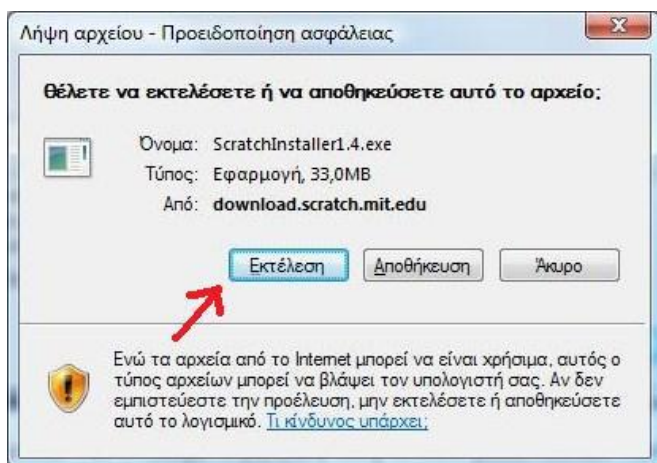
E-mail διεύθυνση:		Αγνοήστε αν είστε κάτω από 13 χρονών
Σχολείο ή Οργανισμός:		
Πόλη:		
Τόπος κατοικίας:		
Χώρα:		
Ρόλος:	☐ Μαθητής/Φοιτητής ☐ Εκπαιδευτικός ☐ Γονικός ☐ Ερευνητής	Άλλα:
Από που ακούσατε για το Scratch;		
Τι είναι αυτό που σας ενδιαφέρει στο Scratch;		
Κάτι άλλο που θέλετε να μοιραστείτε σε σχέση με το υπάδαθρό σας ή τη σχετική εμπειρία σας;		
Συνεχίστε στο κατέβασμα του Scratch		

Στη σελίδα που μας εμφανίζεται επιλέγουμε να κατεβάσουμε το πρόγραμμα ScratchInstaller1.4.exe, το οποίο βρίσκεται κάτω από την επιγραφή «Scratch Installer for Windows» (εφόσον βέβαια έχετε εγκατεστημένο στον υπολογιστή σας το αντίστοιχο λειτουργικό σύστημα της Microsoft).

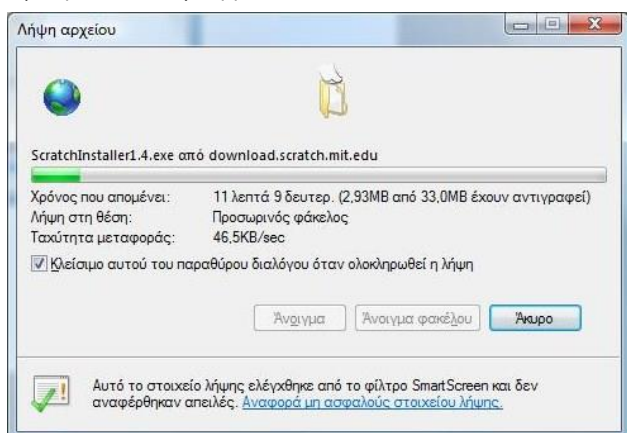


Στο επόμενο παράθυρο του οδηγού εγκατάστασης, μας δίνεται επιπλέον η δυνατότητα να επιλέξουμε το φάκελο του υπολογιστή μας, στον οποίο θα αποθηκευτεί το πρόγραμμα. Καλό θα ήταν να αφήσουμε τον προεπιλεγμένο φάκελο «Program Files». Μετά από αυτό κάνουμε κλικ πάλι στο κουμπί «Next» για να συνεχίσουμε την εγκατάσταση.

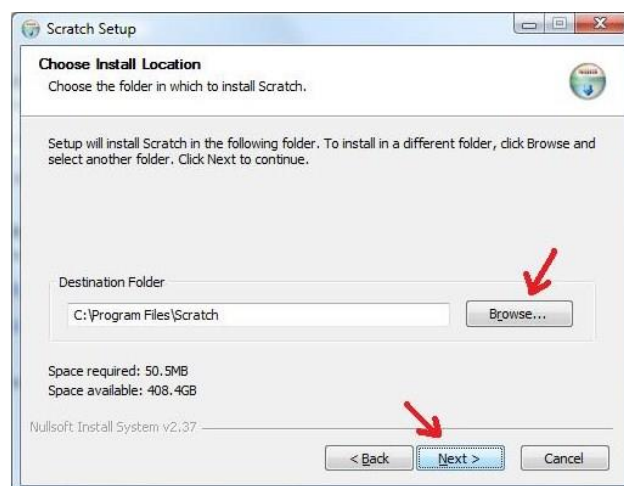
Έπειτα πατάμε «Εκτέλεση».



και περιμένουμε μέχρι ο υπολογιστής μας να τελειώσει με το κατέβασμα του προγράμματος.



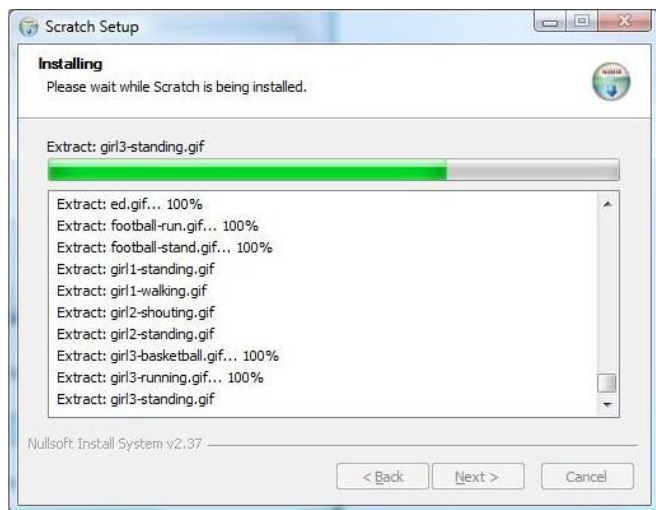
Μόλις «κατέβει» στον υπολογιστή μας, το αρχείο εκτελείται αυτόματα. Τότε ξεκινάει ο οδηγός εγκατάστασης του Scratch και το μόνο που μας μένει πλέον είναι να ακολουθήσουμε τα 3 βήματά του. Στην αρχική οθόνη πληροφοριών επιλέγουμε το κουμπί «Next».



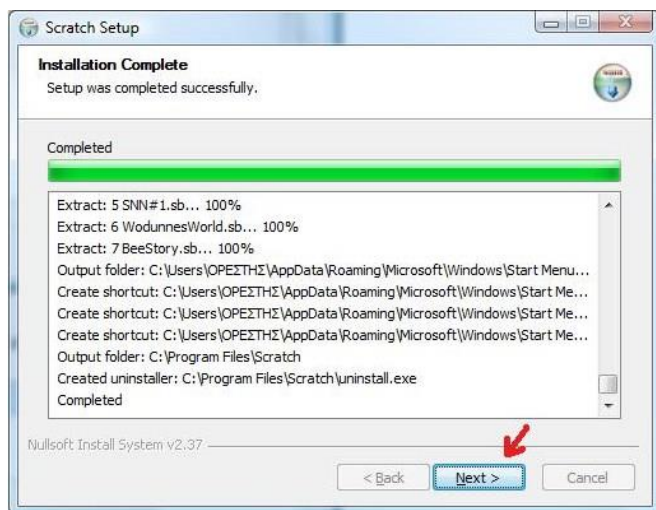
Έπειτα, ο οδηγός μας επιτρέπει να δώσουμε το όνομα της φάκελς μας στο φάκελο που θα εμφανιστεί στη μπάρα εκκίνησης των Windows και ο οποίος θα περιέχει συντόμευση προς την εφαρμογή μας. Στην προκειμένη περίπτωση εμείς δίνουμε το όνομα Scratch. Τέλος, πατάμε στην ένδειξη «Install» για να ξεκινήσει η εγκατάσταση.



Το Scratch εγκαθίσταται στον υπολογιστή μας και σε πολύ λίγο θα είναι έτοιμο για χρήση.



Μόλις η εγκατάσταση ολοκληρωθεί με επιτυχία πατάμε ξανά το κουμπί «Next».

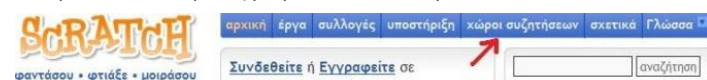


και ολοκληρώνουμε τη διαδικασία εγκατάστασης πατώντας την ένδειξη «Finish».



Ο οδηγός εγκατάστασης θα δημιουργήσει ένα εικονίδιο του Scratch (μια συντόμευση δηλαδή) στην επιφάνεια εργασίας σας, έτσι ώστε η πρόσβαση στο πρόγραμμα να είναι εύκολη.

Να σημειωθεί ότι υπάρχει ειδικά διαμορφωμένος διαδικτυακός τόπος συζήτησης για θέματα του Scratch, στον οποίο μπορείτε να βρεθείτε κάνοντας κλικ στο σύνδεσμο «χώροι συζητήσεων» που βρίσκεται στην αρχική σελίδα του εργαλείου.



Εκεί, εκτός των άλλων, μπορείτε να διατυπώσετε τις όποιες απορίες σας, που έχουν να κάνουν με τη χρήση του λογισμικού, αλλά και να έρθετε παράλληλα σε επαφή με χρήστες του Scratch από όλο τον κόσμο.

2.2 Παραδείγματα προγραμμάτων

Ας πάρουμε όμως μαζί μια πρώτη γεύση από τις δυνατότητες του. Αρχικά ανοίγουμε το Scratch πατώντας διπλό κλικ πάνω στο αντίστοιχο εικονίδιο στην επιφάνεια εργασίας.



Αφού ανοίξει το περιβάλλον του Scratch, θα τρέξουμε ορισμένα έτοιμα παραδείγματα, για να καταλάβετε τι θα μπορείτε να δημιουργήσετε στο τέλος αυτού του βιβλίου και για να έχουμε μια πρώτη επαφή με το περιβάλλον εργασίας του. Για να ανοίξουμε τα έτοιμα παραδείγματα που προσφέρει το Scratch, πατάμε στο κουμπί «Αρχείο» και στη συνέχεια «Άνοιγμα».



Στο παράθυρο που εμφανίζεται στην οθόνη μας («Άνοιξη Έργο») πατάμε στην ένδειξη «Παραδείγματα».



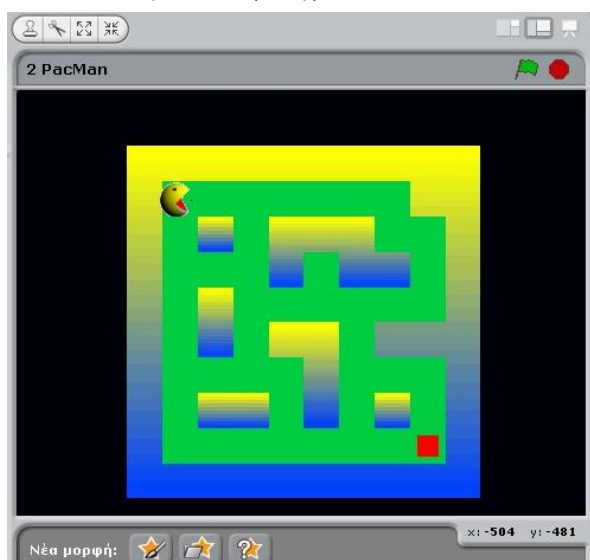
Στη συνέχεια επιλέγουμε μία από τις κατηγορίες που μας εμφανίζονται πατώντας διπλό κλικ πάνω σε μια από αυτές. Φυσικά θα ξεκινήσουμε από την κατηγορία «Games».



Ας εκτελέσουμε το παράδειγμα Packman. Αφού έχουμε εισέλθει στον φάκελο «Games» πατάμε διπλό κλικ στην επιγραφή «PacMan».



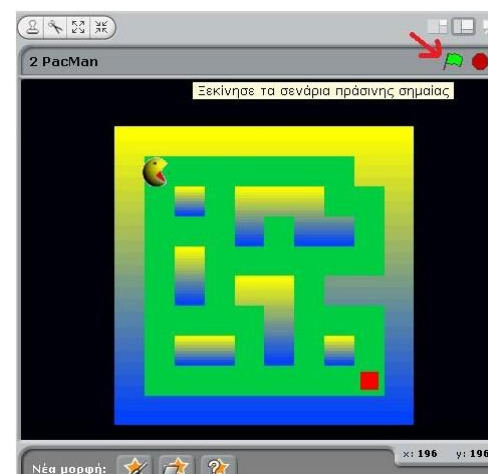
Το Scratch ξεκινάει την εκτέλεση του προγράμματος και εμφανίζει στην οθόνη πάνω δεξιά το περιβάλλον του παιχνιδιού. Ας επικεντρωθούμε σε αυτήν αρχικά.



Για να ξεκινήσουμε το παιχνίδι, πρέπει να πατήσουμε την πράσινη σημαϊούλα όπως φαίνεται στην παρακάτω εικόνα. Δίπλα

στη σημαία, υπάρχει το κόκκινο κουμπί «Stop» με το οποίο μπορούμε να σταματήσουμε το παιχνίδι όποτε το θελήσουμε.

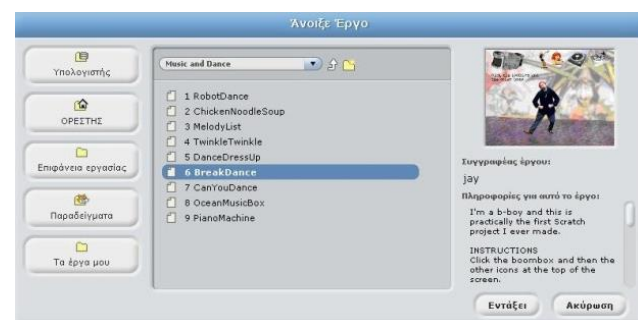
Ο τρόπος χειρισμού του pacman είναι εξαιρετικά απλός, καθώς χρησιμοποιούμε τα βελάκια του πληκτρολογίου μας για να κινήσουμε το χαρακτήρα του παιχνιδιού, με σκοπό κάθε φορά να τον οδηγήσουμε στο κόκκινο τετραγωνάκι, αποφεύγοντας τα εμπόδια του λαβυρίνθου.



Μη χάσετε πολύ ώρα στο PacMan, καθώς έχουμε να δούμε πολλά παιχνίδια κατά τη διάρκεια του συγκεκριμένου βιβλίου. Ας δούμε άλλο ένα παράδειγμα από μια διαφορετική κατηγορία αυτή τη φορά. Ακολουθώντας τα ίδια βήματα που περιγράψαμε παραπάνω (Αρχείο -> Άνοιγμα -> Παραδείγματα), επιλέγουμε την κατηγορία «Music and Dance»,



και εκτελούμε το παράδειγμα BreakDance.



Πατάμε την πράσινη σημαϊούλα και ο χορός ξεκινά...



Ο χαρακτήρας του παραδείγματος αρχίζει και κινείται στο ρυθμό της μουσικής που ακούγεται από το Scratch.

Πάνω από το χαρακτήρα υπάρχουν 5 εικονίδια. Πατώντας σε κάθε ένα από αυτά ο «Rapper» μας (ονόματι B boy) κάνει και από μια διαφορετική χορευτική φιγούρα. Για παράδειγμα, αν πατήσουμε το εικονίδιο της γυναίκας τραγουδίστριας (δηλαδή το μεσαίο) το αποτέλεσμα θα είναι άκρως εντυπωσιακό και διασκεδαστικό, όπως φαίνεται και στην αμέσως επόμενη εικόνα.



Αντιστοίχως και τα υπόλοιπα 4 εικονίδια αναδεικνύουν, με τη σειρά τους, περαιτέρω χορευτικές ικανότητες του χαρακτήρα μας, βάζοντάς μας παράλληλα στον πειρασμό να ακολουθήσουμε και εμείς το ρυθμό του.

Με παρόμοιο τρόπο μπορούμε να τρέξουμε και όλα τα υπόλοιπα παραδείγματα από κάθε μια κατηγορία. Και το παιχνίδι ξεκινά... Καλή μας διασκέδαση!!!

2.3 Αντικείμενα, συμπεριφορές, γεγονότα

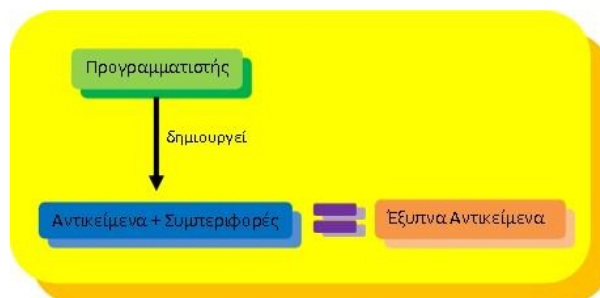
Όπως αναφέραμε και στο πρώτο κεφάλαιο, ο προγραμματισμός αποτελεί ένα τρόπο επίλυσης προβλημάτων κατά τον οποίο συνθέτουμε μια ακολουθία εντολών με σκοπό την επίτευξη συγκεκριμένων στόχων. Ας ξεκινήσουμε λοιπόν με την μελέτη των βασικών του στοιχείων.

Το πρώτο στοιχείο του προγραμματισμού στο Scratch που θα μας απασχολήσει καθ' όλη τη διάρκεια του βιβλίου είναι τα α-

ντικείμενα ή χαρακτήρες ή μορφές ή sprites (όπως αναφέρονται στο Scratch). Αν ρίξουμε μια ματιά γύρω μας, θα παρατηρήσουμε ότι το περιβάλλον μας αποτελείται από αντικείμενα, τα οποία μπορούμε εύκολα να αντιληφθούμε. Σαν παραδείγματα αντικειμένων μπορούμε να αναφέρουμε ένα αυτοκίνητο, μια μπάλα μπάσκετ ή έναν άνθρωπο. Κάθε αντικείμενο προσδιορίζεται και από κάποια συμπεριφορά άλλοτε απλή και άλλοτε πολύπλοκη. Αν αφήσουμε από τα χέρια μας μια μπάλα αυτή πέφτει στο έδαφος και αναπηδά. Παρομοίως ένα αυτοκίνητο έχει ανάλογη συμπεριφορά ανάλογα με το πετάλι που θα πατήσει ο οδηγός. Όμως τι σχέση έχουν όλα αυτά με τον προγραμματισμό στο Scratch;

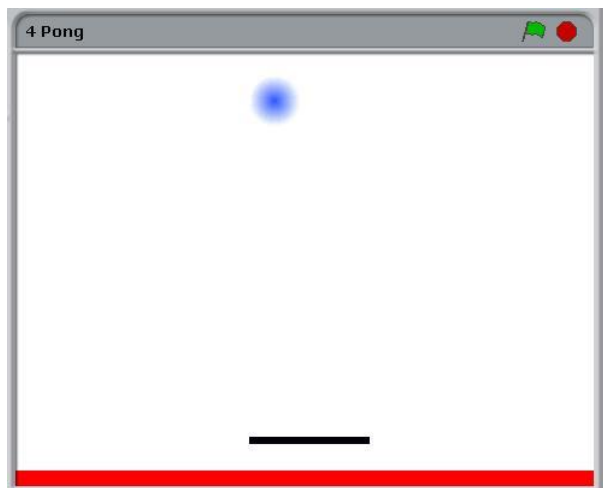


Ας υποθέσουμε ότι είμαστε κατασκευαστές ηλεκτρονικών παιχνιδιών και θέλουμε να φτιάξουμε ένα ράλι που να φαίνεται όσο πιο αληθινό γίνεται. Αρχικά σχεδιάζουμε εμφανισιακά τα διάφορα στοιχεία του παιχνιδιού, άλλωστε έχουμε δει τις δυνατότητες των προγραμμάτων ζωγραφικής στους υπολογιστές. Όμως αυτό δεν αρκεί. Βλέπετε τα αντικείμενα από μόνα τους δεν είναι καθόλου έξυπνα, δεν έχουν δική τους βούληση, αλλά ούτε και υπακούνε σε φυσικούς νόμους, όπως η μπάλα στον πραγματικό κόσμο που πέφτει λόγω του νόμου της βαρύτητας. Σε αυτό το σημείο αναλαμβάνει δράση ο προγραμματιστής δίνοντας ζωή στα αντικείμενα. Πως όμως το καταφέρνει αυτό; Αρκεί να γράψει ένα σύνολο από εντολές που θα καθορίζουν τη συμπεριφορά που θα ήθελε να έχουν τα αντικείμενα του. Έτσι απλά καταφέρνουμε να φέρουμε τον πραγματικό κόσμο στον υπολογιστή μας, μέσα από την φιλοσοφία των αντικειμένων. Ώρα λοιπόν να συμπεριφερθούμε σαν προγραμματιστές.



Ίσως όλοι γνωρίζουμε το γνωστό παιχνίδι Pong. Μπορούμε να το δούμε στα έτοιμα παραδείγματα του Scratch αφού ακολουθήσουμε τη διαδικασία ανοίγματος παραδειγμάτων (φάκελος games, αρχείο Pong). Στο παιχνίδι αυτό, υπάρχει μια μπάλα η οποία κινείται στη οθόνη και αναπηδά στα τοιχώματα, ενώ με μια μπάρα που ελέγχεται από τη θέση του ποντικιού, ο χρήστης προσπαθεί να μην αφήσει την μπάλα να αγγίξει το κόκκινο έδαφος. Μπορούμε να φανταστούμε το συγκεκριμένο παιχνίδι σαν ένα απλό φλιπεράκι. Σαν προγραμματιστές πλέον, είμαστε

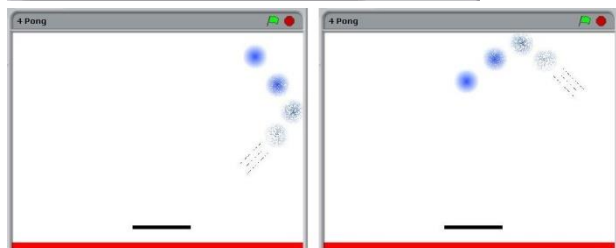
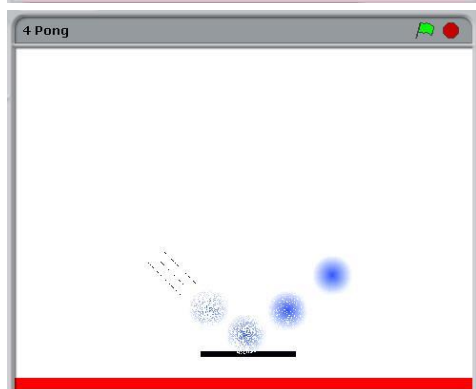
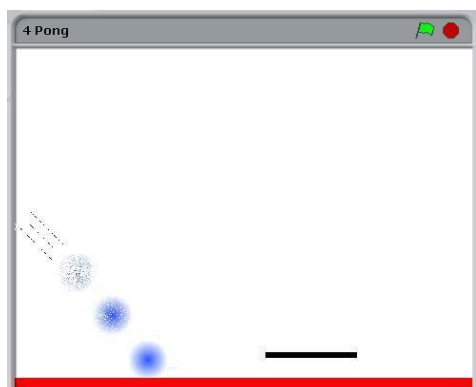
σε θέση να αναγνωρίσουμε τα αντικείμενα μας και τις συμπεριφορές που έχουν.



Αντικείμενα: Μπάρα, Μπάλα.

Η μπάρα — κινείται οριζόντια ανάλογα με την κίνηση του ποντικιού (αριστερά ή δεξιά). Δηλαδή η θέση της ορίζεται από την οριζόντια συντεταγμένη του δείκτη του ποντικιού.

Η μπάλα — τώρα, κινείται σε όλη την οθόνη διαρκώς, ενώ όταν έρθει σε επαφή με τη μπάρα ή με τα τοιχώματα της οθόνης αναπηδά και αλλάζει κατεύθυνση, όπως ακριβώς οι μπάλες σε ένα μπιλιάρδο.



Πως θα περιγράψουμε αυτές τις οδηγίες με προγραμματιστικές εντολές; Αν και τις εντολές αυτές θα τις δούμε αναλυτικά στα επόμενα κεφάλαια, μπορούμε να κάνουμε το εξής τρικ.

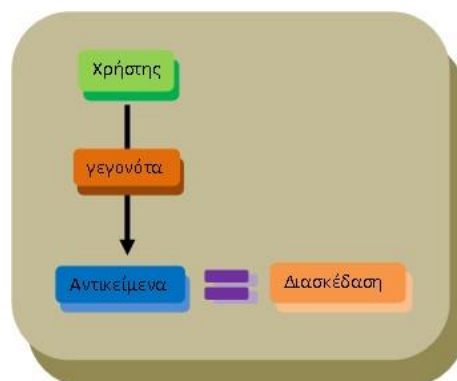
Ας υποθέσουμε ότι διαγράφουμε τις εντολές που υπάρχουν σε αυτό το έτοιμο παράδειγμα. Μη ξεχνάτε ότι οι εντολές είναι τα τουβλάκια που εμφανίζονται αριστερά από την κύρια οθόνη του παιχνιδιού. Αυτό μπορούμε να το πετύχουμε πατώντας «δεξί κλικ->διαγραφή» στη πρώτη εντολή κάθε ομάδας εντολών που βλέπουμε. Τι παρατηρούμε;

Τα αντικείμενα μας έχουν μετατραπεί σε ακίνητες φιγούρες χωρίς καμία λειτουργία και καμία συμπεριφορά όταν πατη-σουμε την πράσινη σημαούλα.

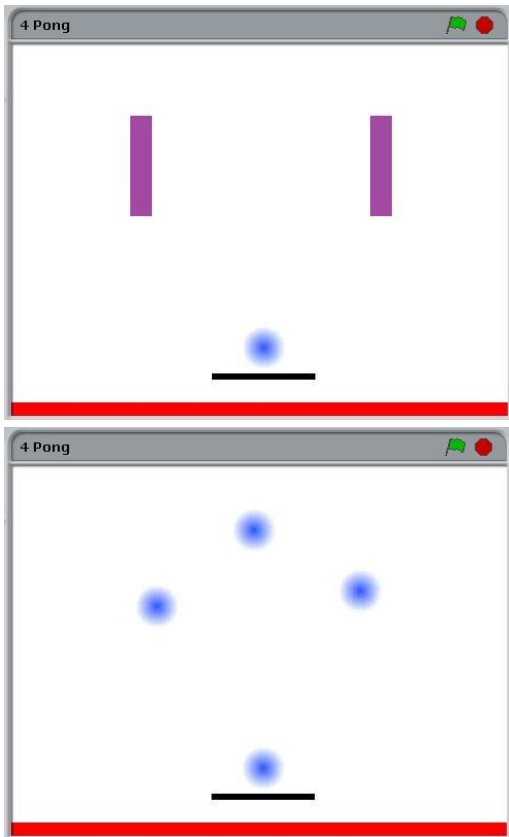
Συνεπώς τι ήταν αυτό που έδινε ζωή στα αντικείμενα μας, δηλαδή προσδιόριζε τη συμπεριφορά τους; Μήπως αυτό το σύνολο από γαλάζια, πορτοκαλί και πράσινα τουβλάκια δηλαδή οι εντολές; Άρα, μήπως όταν αναφερόμαστε σε προγραμματιστές εννοούμε «δημιουργούς αντικειμένων με συμπεριφορές»;



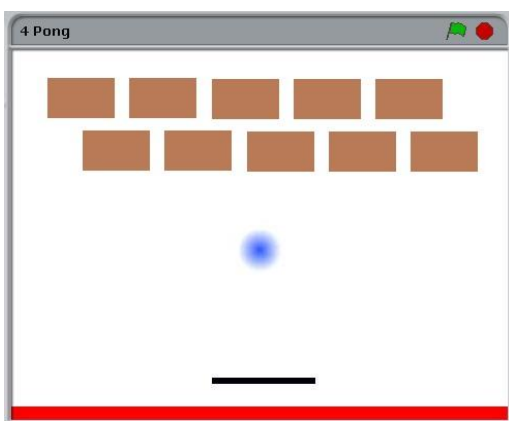
Χρειαζόμαστε κάτι ακόμα για να φέρουμε σε επαφή τον χρήστη με τα αντικείμενα; Τα αντικείμενα έχουν συμπεριφορές, αλλά πως αυτές ενεργοποιούνται; Πως οι χρήστες τα ελέγχουν; Εδώ μπαίνει μια καινούρια έννοια του προγραμματισμού, τα **γεγονότα**. Γεγονός είναι ένα συμβάν που γίνεται αντιληπτό από κάποιο αντικείμενο (π.χ. η κίνηση του ποντικιού, ένα κλικ κτλ.) και εξαιτίας του, το αντικείμενο ενεργοποιεί μια συγκεκριμένη σειρά ενεργειών. Στο pong όταν κουνάμε το ποντίκι, η μπάρα αντιλαμβάνεται την αλλαγή της θέσης του και μετακινεί τη θέση της ρακέτας ανάλογα με τη θέση του δείκτη του ποντικιού.



Αφού μελετήσαμε τα πρώτα βασικά στοιχεία, θα μπορούσαμε να επεκτείνουμε το παράδειγμα του Pong εμπλουτίζοντας το με νέα αντικείμενα ή αλλάζοντας τα γεγονότα. Σε πρώτη φάση θα μπορούσαμε να κάνουμε ποιο δύσκολο το στόχο του παιχνιδιού προσθέτοντας αντικείμενα στην οθόνη που να αλλάζουν πιο συχνά την κατεύθυνση της μπάλας ή ακόμα και να προσθέσουμε επιπλέον μπάλες.



Θα μπορούσαμε επίσης να αλλάξουμε και τη φιλοσοφία του παιχνιδιού προσθέτοντας τουβλάκια στο πάνω μέρος της οθόνης και ζητώντας από το χρήστη εκτός από το να διατηρήσει την μπάλα μέσα στην οθόνη να εξαφανίσει και όλα τα τουβλάκια. Όμως μην ξεχνάμε ότι θα πρέπει να καθορίσουμε και τη νέα συμπεριφορά των νέων μας αντικειμένων. Δηλαδή κάθε φορά που η μπάλα θα χτυπάει τα τουβλάκια αυτά θα πρέπει να εξαφανίζονται. Είναι προφανές, ότι μόλις μάθετε να δημιουργείτε το αρχικό παιχνίδι, πολύ εύκολα μπορείτε να σχεδιάσετε προκλητικές παραλλαγές του!



Αντικείμενο

Οποιαδήποτε μορφή έξυπνης ή μη φιγούρας που πρωταγωνιστεί στο πρόγραμμα μας.

Συμπεριφορά

Σειρά εντολών (πολύχρωμα τουβλάκια) με την βοήθεια των οποίων ένας προγραμματιστής μπορεί να προσδώσει συμπεριφορές σε ένα αντικείμενο.

Γεγονός

Ένα συμβάν που αναγνωρίζεται από το αντικείμενο και για το οποίο το αντικείμενο επιδεικνύει συγκεκριμένες συμπεριφορές.

2.4 Αναγνώριση αντικειμένων

Χωρίς να δυσκολευτούμε ιδιαίτερα μπορούμε να καταλάβουμε ότι ο τρόπος με τον οποίο προγραμματίζουμε έχει σαν βάση του τα αντικείμενα, και στόχος μας για την επίλυση όλων των προβλημάτων είναι να ανιχνεύσουμε τα κατάλληλα αντικείμενα, να τους αποδώσουμε κατάλληλες συμπεριφορές, να αποφασίσουμε πως σχετίζονται μεταξύ τους και πως αντιδρούν σε διαφορετικού τύπου γεγονότα. Οι άνθρωποι σκεφτόμαστε με βάσει τα αντικείμενα και τις συμπεριφορές τους.

Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να περιγράψουμε ένα παιχνίδι αγώνα αυτοκινήτων.



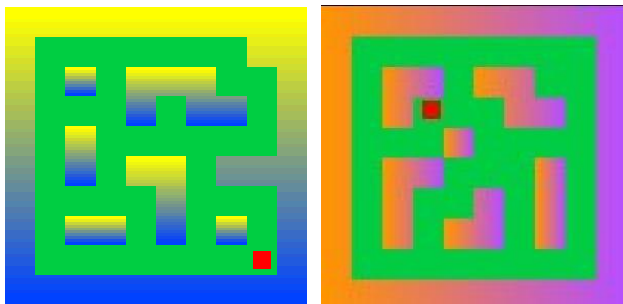
Αναγνωρίζουμε αρχικά το αυτοκίνητο που χειριζόμαστε εμείς ως αντικείμενο. Είναι όμως το μοναδικό αντικείμενο; Φυσικά και όχι! Το αυτοκίνητό μας δεν θα αγωνίζεται μόνο του! Θα έχει ως ανταγωνιστές του άλλα αυτοκίνητα, άρα έχουμε επιπλέον αντικείμενα. Και πάλι όμως δεν τελειώσαμε! Ως αντικείμενο μπορούμε ακόμα να αναγνωρίσουμε ένα εμπόδιο που θα συναντήσουμε μπροστά μας και θα μας αναγκάσει να κάνουμε έναν ελιγμό ή πιθανά λάδια στο δρόμο που θα κάνουν το αυτοκίνητό μας να γλιστρήσει.



Ακόμη, αντικείμενο μπορούμε να θεωρήσουμε και τη γραμμή λήξης του αγώνα. Ανάμεσα στα αντικείμενα αυτά υπάρχουν διάφορες σχέσεις. Όπως αναφέρθηκε, όταν ένα αυτοκίνητο πατήσει πάνω σε λάδια, τότε πιθανά θα γλιστρήσει, αλλάζοντας κατεύθυνση. Ή αν ένα αυτοκίνητο περάσει πρώτο από τη γραμμή τερματισμού, τότε θα είναι ο νικητής του αγώνα! Ακόμα, με κάποιον τρόπο πρέπει να ελέγχουμε και την κίνηση του αυτοκινήτου μας. Πότε θα ελαττώσει ταχύτητα το αυτοκίνητο; Προφανώς όταν πατήσουμε το φρένο. Ακόμη, για να αλλάξει κατεύθυνση χρειάζεται να στρίψουμε το τιμόνι. Το πάτημα του φρένου και το στρίψιμο του τιμονιού αντιστοιχούν σε γεγονότα που πρέπει να προκαλέσει ο χρήστης.

Ας εξετάσουμε τώρα το παράδειγμα που τρέξαμε στην πρώτη ενότητα του κεφαλαίου, το παιχνίδι Pacman. Σ' αυτό το παράδειγμα έχουμε δύο αντικείμενα. Το ένα προφανώς είναι ο ήρωάς μας 🍒. Στο παράδειγμά μας, παρατηρούμε ότι ο πρωταγωνιστής μας μπορεί να κινείται και να αλλάζει μορφή (καθώς ανοιγοκλείνει το «στόμα» του)! Το δεύτερο αντικείμενο που έχουμε είναι ο λαβύρινθος μέσα στον οποίο κινείται ο πακμαν.

Ο λαβύρινθος βλέπουμε ότι αλλάζει μορφή συμβολίζοντας έτσι την αλλαγή της πίστας όταν ο χαρακτήρας μας «φάει» την κόκκινη κουκίδα!



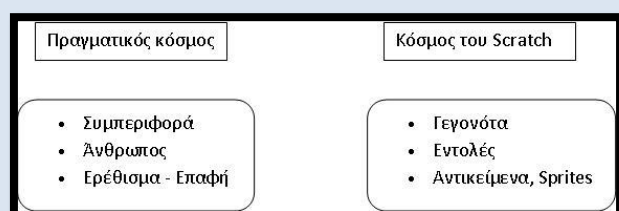
Παρατηρούμε λοιπόν ότι το παιχνίδι για να «τρέξει» χρησιμοποιεί δύο αντικείμενα που αλληλεπιδρούν μεταξύ τους! Ο πακμαν μπορεί να κινηθεί μέσα στο λαβύρινθο (αποφεύγοντας τους τοίχους) και ο λαβύρινθος αλλάζει μορφή όταν ο πακμαν ακουμπήσει πάνω στην κόκκινη κουκίδα που έχει ο λαβύρινθος! Άρα τα πάντα είναι αντικείμενα! Ή τουλάχιστον έτσι θα τα βλέπουμε όταν προγραμματίζουμε στο Scratch.

Περίληψη

Συνοψίζοντας, σε αυτό το κεφάλαιο είδαμε πως μπορούμε να εγκαταστήσουμε το Scratch και να εκτελέσουμε μια σειρά έτοιμων παραδειγμάτων που παρέχονται. Στη συνέχεια γνωρίσαμε τις πρώτες μας προγραμματιστικές έννοιες όσον αφορά τον αντικειμενοστραφή προγραμματισμού όπως τα αντικείμενα, τις συμπεριφορές και τα γεγονότα. Τονίσαμε ότι καθώς θα προγραμματίζουμε στο scratch θα σκεφτόμαστε βάσει αντικειμένων, των συμπεριφορών τους και των γεγονότων στα οποία αντιδρούν. Κάτι αντίστοιχο κάνουμε και στην καθημερινότητά μας (?).

Ερωτήσεις

1) Στο παρακάτω σχήμα αντιστοιχίστε τις έννοιες:



2) Δώστε παραδείγματα αντικειμένων που συμμετέχουν σε προβλήματα της καθημερινότητας σας.

3) Περιγράψτε τις συμπεριφορές των αντικειμένων του προηγούμενου ερωτήματος.

4) Δώστε 3 εντολές που δέχστε από τους γονείς σας οι οποίες διαμορφώνουν κάποια συμπεριφορά σας.

5) Αναφέρετε 3 επεκτάσεις που θα μπορούσατε να κάνετε στο παιχνίδι του Pacman οι οποίες θα έκαναν το παιχνίδι πιο δύσκολο και πιο ενδιαφέρον. Οι επεκτάσεις χρειάζονται νέα αντικείμενα; Με ποια συμπεριφορά;

Δραστηριότητες

1) Στο έτοιμο παράδειγμα του Scratch, το ρόλι, περιγράψτε ποια είναι τα αντικείμενα και ποια τα γεγονότα; Να περιγράψετε με μορφή οδηγιών τη συμπεριφορά των αντικειμένων.

2) Περιηγηθείτε στα διάφορα έτοιμα παραδείγματα του Scratch και προσπαθήστε να σκεφτείτε ένα πρόβλημα από τη δική σας καθημερινότητα, που θα μπορούσε να μεταφερθεί στο Scratch. Καταγράψτε αναλυτικά αντικείμενα, συμπεριφορές και γεγονότα.

3) Έστω ότι θέλουμε να προγραμματίσουμε το παιχνίδι του σκακιού στο Scratch. Κατονομάστε όλα τα αντικείμενα που θα λάβουν μέρος στο παιχνίδι, το σκηνικό στο οποίο θα διαδραματιστεί το παιχνίδι, και πως θα αλληλεπιδράσουν όλα αυτά μεταξύ τους.



Κεφάλαιο 3: Το περιβάλλον

Σε αυτό το κεφάλαιο:

- 3.1 Γρήγορη ξενάγηση
- 3.2 Πρωταγωνιστές...τα αντικείμενα
- 3.3 Η σκηνή είναι δική μας
- 3.4 Εντολές

«Το να είσαι ικανός σημαίνει να μπορείς να ελέγχεις και να χειρίζεσαι τα πράγματα στο περιβάλλον σου».
(L.Ron Hubbard)



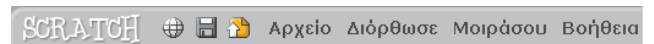
3.1 Γρήγορη ξενάγηση στο Περιβάλλον Εργασίας

Ήρθε η ώρα να πειραματιστούμε με τις πολυάριθμες δυνατότητες που παρέχει το προγραμματιστικό περιβάλλον του Scratch. Συγκεκριμένα, θα δούμε τα βασικά χαρακτηριστικά του περιβάλλοντος εργασίας, τις διάφορες παλέτες εντολών, το πώς μπορούμε να εισάγουμε μία εντολή, πώς δημιουργούμε αντικείμενα, σκηνικά και σενάρια. Αλλά ας ξεκινήσουμε με μια γρήγορη ξενάγηση στο περιβάλλον εργασίας.

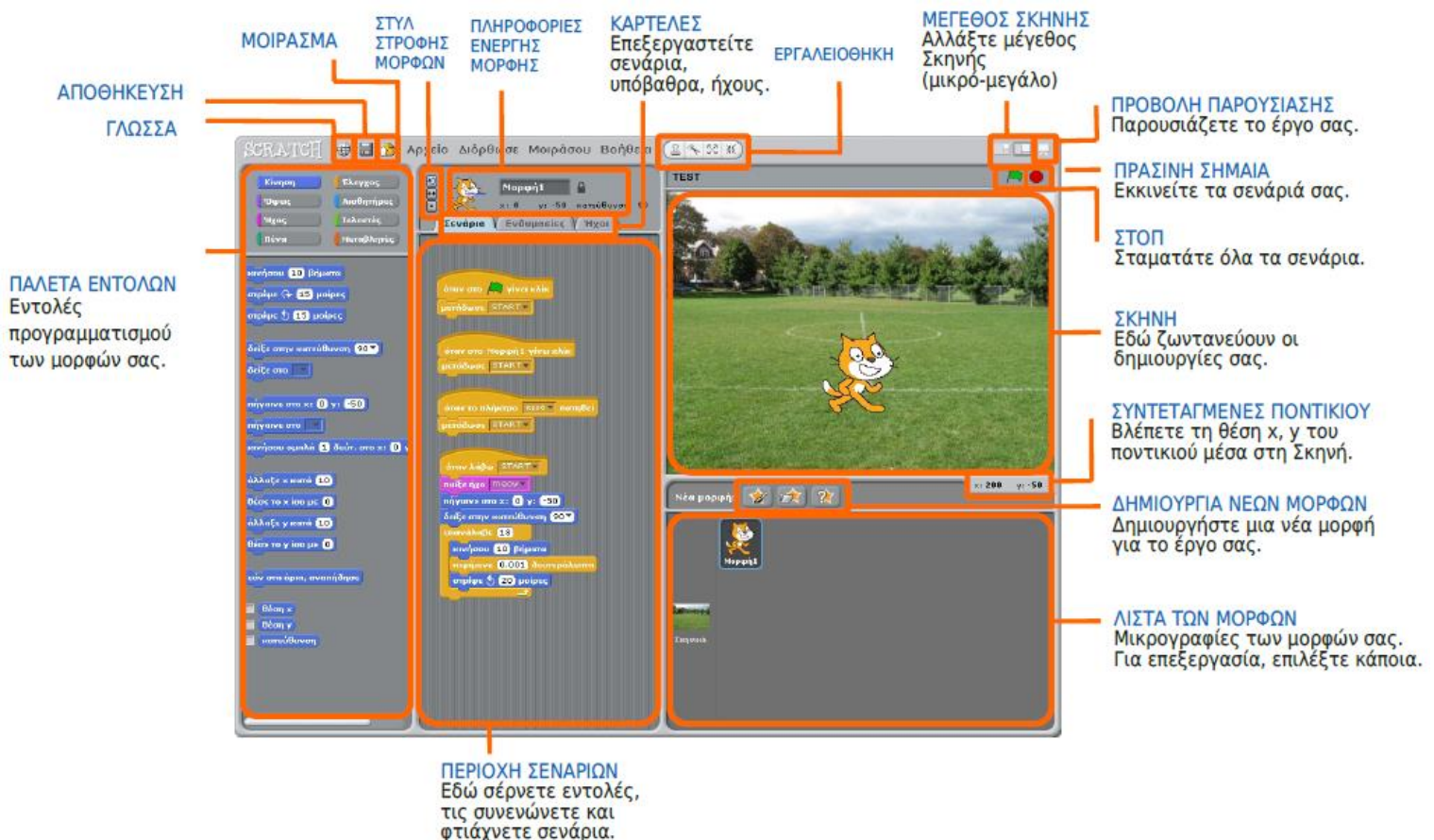
Η εικόνα που ακολουθεί παρουσιάζει τις βασικές περιοχές του περιβάλλοντος εργασίας του Scratch. Η περιοχή με τη φωτογραφία του γηπέδου στα δεξιά είναι η σκηνή, ο χώρος δράσης των αντικειμένων. Ακριβώς από κάτω βρίσκεται η λίστα αντικειμένων και σκηνικών. Στο κέντρο του παραθύρου βρίσκεται ο

χώρος στον οποίο εισάγουμε τις εντολές για κάθε αντικείμενο, ενώ στα αριστερά βλέπουμε τις παλέτες εντολών από όπου βλέπουμε τις διαθέσιμες.

Πριν ξεκινήσετε να κάνετε τις δικές σας δημιουργίες, είναι σημαντικό να γνωρίζετε επίσης κάποιες από τις λειτουργίες του μενού του Scratch οι οποίες θα μας χρειαστούν πολύ σύντομα, όπως η αποθήκευση αρχείου.



- Με το εικονίδιο γλώσσας (το πρώτο από αριστερά) αλλάζετε τη γλώσσα της επιφάνειας εργασίας του Scratch.
- Με το εικονίδιο αποθήκευσης (το δεύτερο από αριστερά) αποθηκεύετε το έργο σας.



- Με το εικονίδιο μοιράσματος (το τρίτο από αριστερά) ανεβάζετε το έργο σας στον ιστοχώρο του Scratch και το δημοσιοποιείτε στο διαδίκτυο.
- Από το μενού Αρχείο μπορείτε να δημιουργήσετε ένα νέο έργο, να ανοίξετε ένα αποθηκευμένο έργο και να αποθηκεύσετε το τρέχον έργο σας.
- Εισαγωγή έργου: εισάγει όλες τις μορφές και τα υπόβαθρα ενός άλλου έργου μέσα στο τρέχον. Αυτό είναι χρήσιμο όταν θέλετε να χρησιμοποιήσετε ταυτόχρονα τις μορφές πολλών έργων.
- Εξαγωγή μορφής: εξάγει τον ενεργό χαρακτήρα ως αρχείο .sprite, το οποίο μπορεί να εισαχθεί σε ένα άλλο έργο.
- Σημειώσεις έργου: σας επιτρέπει να γράψετε και να αποθηκεύσετε σημειώσεις σχετικά με το έργο σας, όπως περιγραφές των προγραμματιστικών επιλογών σας.
- Εξοδος: τερματίζει το Scratch.
- Το μενού Διόρθωση παρέχει διάφορες λειτουργίες επεξεργασίας του τρέχοντος έργου.
- Αναίρεση διαγραφής: αναίρει την τελευταία διαγραφή από ένα τουβλάκι, σενάριο, μορφή, ενδυμασία ή ήχο.
- Ξεκίνησε από βηματισμό: το Scratch τρέχει το έργο σας ένα βήμα κάθε φορά, ενώ ταυτόχρονα επισημαίνεται η τρέχουσα εντολή. Αυτή η λειτουργία είναι χρήσιμη για την εύρεση σφαλμάτων μέσα στο έργο, αλλά και ως βοήθεια σε νέους προγραμματιστές για να αντιληφθούν την πορεία εκτέλεσης ενός προγράμματος.
- Ορίσε από βηματισμό: επιλέγετε την ταχύτητα εκτέλεσης βήμα βήμα (αργή ή γρήγορη).
- Συμπίεσε ήχους και Συμπίεσε εικόνες: μειώνει το μέγεθος του αρχείου του έργου συμπιέζοντας τους ήχους και τις εικόνες του. Αυτό όμως μπορεί να υποβαθμίσει την ποιότητά τους.
- Εμφάνιση εντολών κινήτων: εμφανίζει εντολές κινήτηρα στην κατηγορία Κίνηση. Με αυτές τις εντολές μπορείτε να προγραμματίσετε έναν κινήτηρα συνδεδεμένο στον υπολογιστή σας. (π.χ. LEGO® Education WeDo™ (<http://www.legoeducation.com>)).
- Από το μενού Μοιράσου μπορείτε να ανεβάσετε το έργο σας στον ιστοχώρο του Scratch.
- Από το μενού Βοήθεια έχετε πρόσβαση στη σελίδα βοήθειας με υπερσυνδέσμους σε υλικό αναφοράς, οδηγίες χρήσης και συχνές ερωτήσεις.

Είναι λογικό να μην έχετε κατανοήσει πολλές από τις επιλογές. Μην ανησυχείτε όμως καθώς θα τις ξανασυζητήσουμε παρακάτω.

3.2 Πρωταγωνιστές στα έργα μας ...τα αντικείμενα!

Πως δημιουργούμε ένα αντικείμενο-χαρακτήρα-μορφή; (στο βιβλίο αυτό είτε χρησιμοποιούμε τον όρο «μορφή», είτε αναφερόμαστε στο «αντικείμενο», ή στο «χαρακτήρα», θα εννοούμε το ίδιο πράγμα). Για την δημιουργία ενός αντικειμένου χρησιμοποιούμε το μενού που φαίνεται στην επόμενη εικόνα. Τα τρία

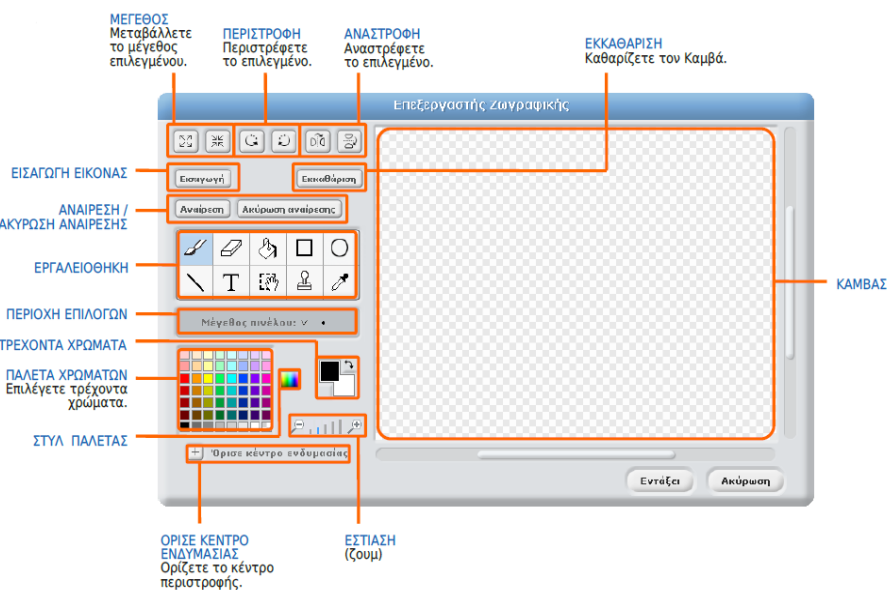
κουμπιά εικόνες που βλέπετε σας επιτρέπουν να δημιουργήσετε μια νέα μορφή με τρεις διαφορετικούς τρόπους.



«Ζωγράφισε μια νέα μορφή»: Πατώντας το πρώτο εικονίδιο εμφανίζεται ο επεξεργαστής ζωγραφικής. Εδώ μπορείτε να δημιουργήσετε τη δική σας εμφάνιση για το αντικείμενό σας χρησιμοποιώντας τη ζωγραφική (θα μάθετε περισσότερα στο επόμενο κεφάλαιο). Μπορείτε να ζωγραφίσετε ένα αυτοκίνητο, ένα καρότο ή ακόμη και να εισάγετε μια φωτογραφία ενός φίλου σας!

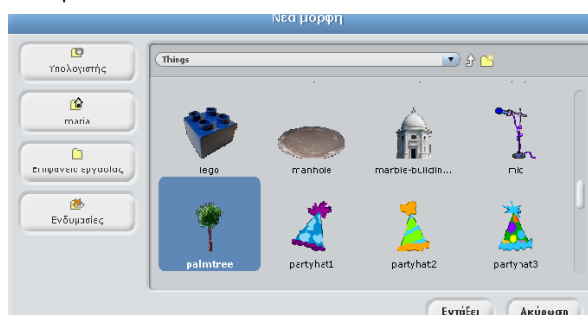


«Διάλεξε μια νέα μορφή από το αρχείο»: Πατώντας το δεύτερο εικονίδιο στη σειρά αποκτούμε πρόσβαση στη βιβλιοθήκη έτοιμων αντικειμένων που μας παρέχει το Scratch. Εμφανίζονται φάκελοι που περιέχουν προσχεδιασμένα αντικείμενα ό-

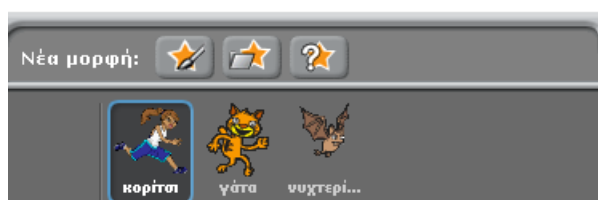


πως ζώα, μεταφορικά μέσα, φανταστικοί χαρακτήρες (δράκους, ρομπότ, χιονάνθρωπους κτλ.). Τις περισσότερες φορές αυτές οι μορφές θα καλύπτουν τις ανάγκες μας.

Για παράδειγμα, μπορούμε να εισάγουμε ως αντικείμενο στο έργο μας ένα δέντρο επιλέγοντας από το φάκελο Things την εικόνα palmtree .

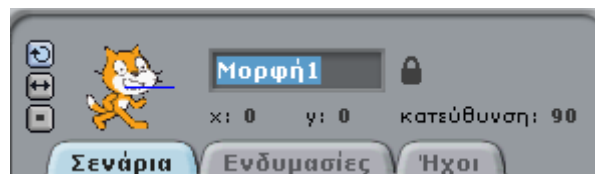


«**Πάρε μια μορφή έκπληξη**» : Κάνοντας κλικ στο τρίτο εικονίδιο θα εμφανιστεί στη σκηνή τυχαία ένα αντικείμενο έκπληξη! Το Scratch θα διαλέξει από μόνο του έναν από τους χαρακτήρες που έχει στη βιβλιοθήκη του και θα τον εισάγει στο έργο μας. Παρακάτω εμφανίζεται έργο στο οποίο πατήσαμε διαδοχικά το κουμπί «πάρε μια μορφή έκπληξη».



Κάθε φορά όμως που ανοίγετε το Scratch βλέπετε στη σκηνή τη χαρακτηριστική γάτα! Είναι το προεπιλεγμένο αντικείμενο του προγράμματος, το οποίο έχει το δικό του όνομα, το όνομα Μορφή1, όπως φαίνεται στην επόμενη εικόνα. Γιατί όμως οι χαρακτήρες-αντικείμενα που δημιουργούμε να μην έχουν τα ονόματα που θέλουμε εμείς; Θέλω το δράκο να τον ονομάσω Τάκη! Πως μπορεί να γίνει αυτό; Μπορείτε να **ονοματίσετε** όλα τα αντικείμενα κάνοντας κλικ στο προεπιλεγμένο όνομα κάθε αντικειμένου (π.χ. Μορφή1), στην κεντρική περιοχή του Scratch και

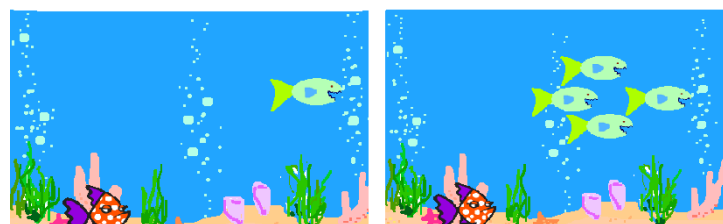
πληκτρολογώντας το όνομα που επιθυμείτε. Μην ξεχνάτε όμως ότι κατά την λογική επίλυση του προγραμματιστικού προβλήματός σας, καλό θα ήταν να αναφέρεστε στα αντικείμενα με ονόματα που είναι σχετικά με το πρόβλημα. Το όνομα «Τάκης» μπορεί να μη σας βοηθά αρκετά για να αναγνωρίσετε ότι αναφέρεται στη μορφή ενός δράκου!



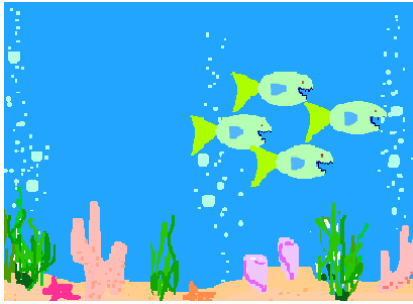
Πως μπορούμε να **αλλάξουμε** ή να **διαγράψουμε** τους χαρακτήρες που έχουμε εισάγει μέσα στο σκηνικό μας; Ας πάμε στην γραμμή εργαλείων που βρίσκεται δεξιά από την γραμμή του μενού και πάνω από την σκηνή.



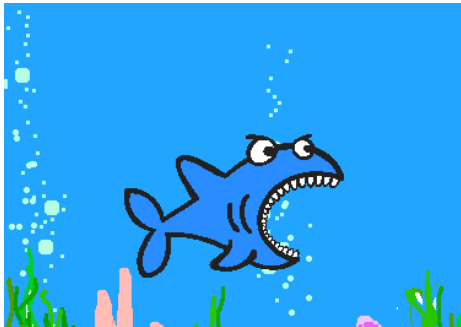
Στον παρακάτω ήρεμο βυθό, ας υποθέσουμε ότι θέλουμε το πράσινο ψάρι να συναντά ένα κοπάδι με όμοια ψάρια και να το ακολουθεί. Πως θα δημιουργήσουμε τους αντίστοιχους χαρακτήρες; Ο τυπικός τρόπος είναι να εισάγουμε 4 φορές το πράσινο ψάρι με τη χρήση της επιλογής «**Διάλεξε μια νέα μορφή από το αρχείο**». Κουραστικό! Το Scratch μας δίνει και άλλη δυνατότητα. Για να δημιουργήσετε το κοπάδι κάντε κλικ στο **Διπλασιασμός** (το κουμπί που περιέχει τη σφραγίδα) και στη συνέχεια κρατήστε πατημένο το πλήκτρο shift και κάνετε κλικ πάνω στο ψάρι τρεις φορές. Τρία νέα ψάρια δημιουργήθηκαν πολύ γρήγορα! Για τη δημιουργία ενός μόνο αντίγραφου θα μπορούσατε εναλλακτικά να κάνετε δεξί κλικ πάνω στο αντικείμενο που θέλετε να αντιγράψετε και να επιλέξετε από το μενού που εμφανίζεται την επιλογή «διπλασίασε».



Έστω ότι το πουά ψάρι φαίνεται να μην είναι χρήσιμο πια στο παιχνίδι που σχεδιάζουμε! Για να **διαγράψετε** το ψάρι κάντε κλικ στο Διέγραψε (το εικονίδιο ψαλίδι) και στη συνέχεια πάνω στο ψάρι. Εναλλακτικά μπορείτε να κάνετε δεξί κλικ πάνω στο ψάρι και να επιλέξετε «διαγραφή» από το μενού που εμφανίζεται.



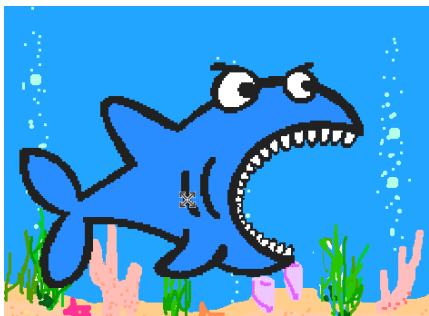
Αλλάζουμε όμως άποψη και θέλουμε να βάλουμε στο παιχνίδι μας ένας πεινασμένο καρχαρία.



Όπως και με τον διπλασιασμό, για να διαγράψουμε όλα τα προηγούμενα ψαράκια κάνουμε κλικ στο Διέγραψε (ψαλίδι στην Εικόνα 3.3) και μετά κρατάμε πατημένο το shift και κάνουμε κλικ πάνω στα ψάρια. Εισάγουμε στη συνέχεια τον καρχαρία από τη βιβλιοθήκη του Scratch.



Για να κάνετε πιο τρομακτικό τον καρχαρία μπροστά στα ψαράκια μπορείτε να τον **μεγεθύνετε**.



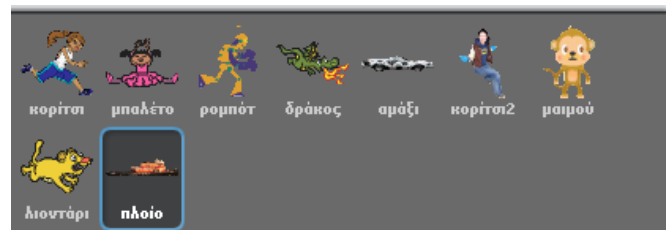
Για να αλλάξετε το μέγεθος ενός αντικειμένου κάντε κλικ στο «Μεγάλωσε την μορφή» ή «Σμίκρυνση μορφής» και στη συνέχεια πάνω στο αντικείμενο. Για περισσότερα βήματα μεγέθυνσης ή σμίκρυνσης μπορείτε και σε αυτήν την περίπτωση να κρατήσετε πατημένο το shift και να πατήσετε πολλαπλά κλικ πάνω στο αντίστοιχο αντικείμενο. Εναλλακτικά μπορείτε να κάνετε δεξί κλικ πάνω στο αντικείμενο και να επιλέξετε αλλαγή μεγέθους μορφής. Στην επόμενη εικόνα φαίνονται οι επιλογές που

αναφέρθηκαν προηγουμένως και οι οποίες εμφανίζονται με δεξί κλικ πάνω στο αντικείμενο.

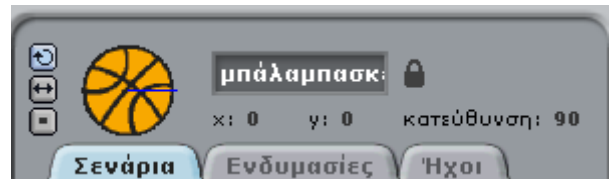


άρπαξε περιοχή οθόνης για νέα ενδυμασία
εξάγαγε αυτή τη μορφή
διπλασίασε
διαγραφή
αλλαγή μεγέθους μορφής
περίστρεψε αυτή τη μορφή

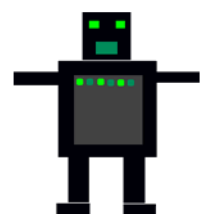
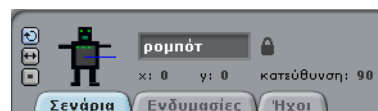
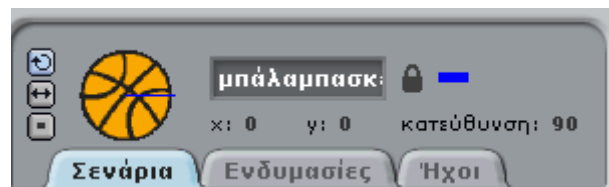
Η **λίστα αντικειμένων** που βρίσκεται κάτω από τη μπάρα δημιουργίας αντικειμένων, δείχνει όλα τα αντικείμενα που έχουν εισαχθεί σε ένα έργο σε μικρογραφίες με τα ονόματά τους. Το εικονίδιο του αντικειμένου που είναι επιλεγμένο περιβάλλεται από μπλε περίγραμμα. Εάν ένα αντικείμενο είναι εκτός σκηνής μπορείτε να το εμφανίσετε κάνοντας shift+κλικ πάνω στη μικρογραφία του.



Με κλικ σε μια μικρογραφία ή διπλό κλικ πάνω στο αντικείμενο μπορείτε να δείτε αναλυτικά τα χαρακτηριστικά του αντικείμενου που φαίνονται στο πάνω μέρος της επιφάνειας εργασίας:



Στην περιοχή αυτή φαίνονται το όνομα του αντικειμένου, οι x, και y συντεταγμένες του στη σκηνή (η θέση του δηλαδή μέσα στη σκηνή του προγράμματος), η κατεύθυνσή του (κάθε καινούριο αντικείμενο που εισάγεται εμφανίζεται με κατεύθυνση 90 μοιρών), και ακριβώς δίπλα, αν έχει επιλεγεί η πένα, μια μικρή γραμμή που υποδηλώνει ότι το αν η πένα είναι ενεργή για το συγκεκριμένο αντικείμενο. Περισσότερα για την πένα θα αναφερθούν σε επόμενο κεφάλαιο. Εισάγετε ένα αντικείμενο έκπληξη και με δεξί κλικ πάνω του, επιλέξτε «περίστρεψε αυτή τη μορφή». Παρατηρήστε πως μεταβάλλεται η κατεύθυνσή του.

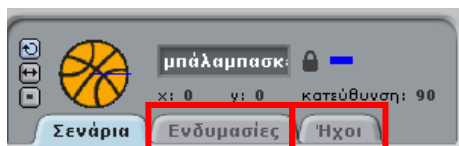




Η καρτέλα «Σενάρια» είναι αυτή που περιέχει τις εντολές που δημιουργούμε για κάθε ένα αντικείμενο ξεχωριστά, δηλαδή πολύχρωμες οδηγίες για τη συμπεριφορά του. Δείτε ένα τέτοιο σενάριο.



Είναι σημαντικό να κατανοήσουμε ότι το Scratch είναι μια γλώσσα που προωθεί τον παράλληλο προγραμματισμό καθώς ένα έργο μπορεί να περιέχει πολλά σενάρια και εντολές που τρέχουν ξεχωριστά και παράλληλα για κάθε αντικείμενο.



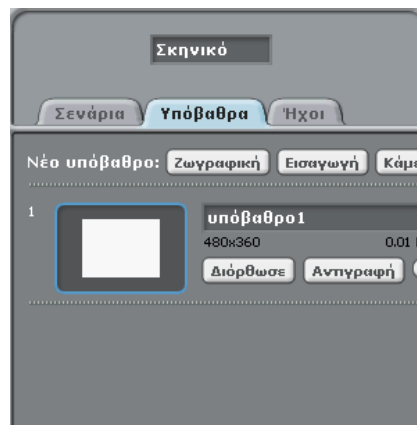
Κάθε αντικείμενο μπορεί να έχει εμφανίζεται στη σκηνή μας με διαφορετικές ενδυμασίες και να αναπαράγει διαφορετικούς ήχους. Δηλαδή το αντικείμενο δράκος μπορεί να εμφανίζεται με διαφορετικό δέρμα, διαφορετικά μάτια και διαφορετικά ρούχα σε συγκεκριμένες στιγμές του παιχνιδιού μας. Τις δυνατές εμφανίσεις των αντικειμένων μπορείτε να τις δείτε και να τις επεξεργαστείτε από την επιλογή "Ενδυμασίες" ενώ από την καρτέλα "Ήχοι" μπορείτε να επεξεργαστείτε τους ήχους που παράγουν τα αντικείμενα. Δεν αναφέρουμε περισσότερες πληροφορίες και συνεχίζουμε τη γρήγορη πλοήγησή μας στο περιβάλλον του Scratch

3.4 Η σκηνή...είναι δική σας.

Ποιο είναι το σκηνικό μέσα στο οποίο εκτυλίσσεται το παιχνίδι μας; Μια πεδιάδα; Ένα κάστρο; Μια έρημος; Το Scratch έχει προβλέψει ένα ειδικό αντικείμενο, **το σκηνικό**, για να εισάγουμε το υπόβαθρο-φόντο μπροστά από το οποίο οι πρωταγωνιστές μας εκτελούν τις εντολές μας. Για να το καταλάβουμε καλύτερα

μπορούμε να παρομοιάσουμε τη σκηνή με το γνωστό σε όλους μας λευκό πανί του Θεάτρου Σκιών όπου εκεί πάνω ο Καραγκιόζης και η παρέα του παίρνουν ζωή και μας κάνουν να διασκεδάσουμε μέσα από τις ιστορίες τους.

Είναι λογικό, όπως συμβαίνει σε μια ταινία ή στα παιχνίδια, η δράση να μην εξελίσσεται όλη μπροστά από ένα και μοναδικό σκηνικό. Πολλές φορές οι δημιουργίες μας απαιτούν την χρήση περισσότερων από ένα σκηνικών για να είναι ολοκληρωμένες. Κάνοντας διπλό-κλικ πάνω στην αρχική σκηνή, η κεντρική περιοχή δείχνει τα χαρακτηριστικά της σκηνής που έχουμε επιλέξει, δηλαδή το σενάριο, τα υπόβαθρα και τους ήχους που σχετίζονται με το σκηνικό του έργου μας.

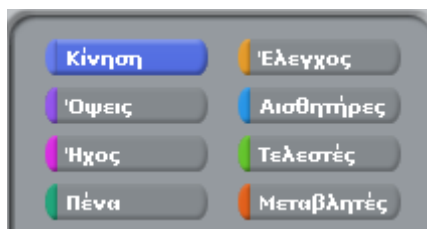


Σίγουρα θα παρατηρείτε κάποιες ομοιότητες μεταξύ του σκηνικού και του αντικείμενου. Δηλαδή ότι και το σκηνικό και τα αντικείμενα έχουν σενάριο, υπόβαθρα-ενδυμασίες και ήχους, αντίστοιχα. Μήπως θα ήταν πιο βολικό να χρησιμοποιούμε μόνο αντικείμενα και να εισάγουμε ένα αντικείμενο ως σκηνικό για να μην ταλαιπωρούμαστε;

Η απάντηση είναι αρνητική. Αυτό που έχει σημασία είναι να κατανοήσετε ότι ένα αντικείμενο αντιπροσωπεύει μια οντότητα. Αντίθετα το σκηνικό αποτελεί το χώρο μέσα στον οποίο δραστηριοποιείται η οντότητα. Τα αντικείμενα κινούνται και δρουν μέσα στο σκηνικό. Ο διαχωρισμός αυτός μεταξύ σκηνικού και αντικείμενου φαίνεται εξάλλου και από τις διαφορετικές διαθέσιμες εντολές που υπάρχουν για το καθένα. Για παράδειγμα κάνοντας κλικ στο σενάριο ενός σκηνικού θα διαπιστώσετε ότι δεν υπάρχουν διαθέσιμες εντολές κίνησης. Συνεπώς, ένα από τα πρώτα ερωτήματα που έχουμε να απαντήσουμε όταν δημιουργούμε ένα πρόγραμμα στο Scratch είναι το ποιο θα είναι το σκηνικό μας, μέσα σε ποια σκηνή ή σκηνές θα διαδραματιστεί η ιστορία μας.

3.5 Εντολές

Μπορείτε να δώσετε ζωή στις δημιουργίες σας, τοποθετώντας εντολές στο χώρο των σεναρίων κάθε αντικείμενου. Σημειώνουμε ξανά ότι κάθε αντικείμενο έχει τα δικά του σενάρια, τις δικές του συμπεριφορές και πριν εισάγουμε οποιαδήποτε εντολή πρέπει να διακρίνουμε το αντικείμενο που είναι επιλεγμένο από την λίστα αντικειμένων μας. Πρώτα επιλέγουμε αντικείμενο και μετά εισάγουμε εντολές στο συγκεκριμένο αντικείμενο.



Το βασικό δομικό στοιχείο ενός σεναρίου είναι οι εντολές. Για να εισάγουμε εντολές πρέπει να χρησιμοποιήσουμε την παλέτα εντολών που μας παρέχει το Scratch. Η παλέτα αυτή βρίσκεται στην πάνω αριστερή μεριά του παραθύρου του Scratch και οι εντολές είναι κατηγοριοποιημένες ανάλογα με το τι προκαλούν στον χαρακτήρα μας, όπως φαίνεται στην επόμενη εικόνα. Πατώντας σε κάθε κατηγορία, εμφανίζονται από κάτω οι αντίστοιχες διαθέσιμες εντολές. Μπορούμε να σύρουμε τις εντολές στο χώρο των σεναρίων, να τις ενώσουμε και να σχηματίσουμε σεσνάκια. Ή μπορούμε να κάνουμε διπλό κλικ στις εντολές μέσα στην παλέτα για να δούμε τι επίδραση θα έχουν στο επιλεγμένο αντικείμενο. Ας δούμε λίγο πιο αναλυτικά τις διαθέσιμες κατηγορίες εντολών:

Κίνηση: εντολές που κινούν τα αντικείμενα, αλλάζουν την κατεύθυνσή τους και καθορίζουν την θέση τους στο σκηνικό. Στο παράδειγμα που ακολουθεί το επιλεγμένο αντικείμενο θα κινηθεί κατά 10 βήματα μόλις πατήσουμε την πράσινη σημαία.



Όψεις: εντολές που αφορούν τη διαχείριση της εμφάνιση των αντικειμένων, όπως π.χ. την αλλαγή του μεγέθους ή της ενδυμασίας του αντικειμένου. Στο παράδειγμα που ακολουθεί, μόλις πατηθεί η πράσινη σημαία, το μέγεθος του αντικειμένου μεγαλώνει κατά 10%.



Ήχος: εντολές που αναπαράγουν μουσική και ηχογραφήσεις που έχουν εισαχθεί στο επιλεγμένο αντικείμενο ή που επιτρέπουν την αναπαραγωγή νοτών από συγκεκριμένα μουσικά όργανα. Στο παράδειγμα που ακολουθεί, μόλις πατήσουμε την πράσινη σημαία ο χαρακτήρας μας, θα κάνει «μιάου» (εφόσον έχουμε ανοικτά τα ηχεία μας...).



Πένα: εντολές που επιτρέπουν στο αντικείμενο να ζωγραφίζει στη σκηνή καθώς κινείται, δηλαδή να αφήνει ένα ίχνος στις θέσεις πάνω από τις οποίες κινείται. Στο παράδειγμα που ακολουθεί, μόλις πατηθεί η πράσινη σημαία, θα κατέβει η πένα και όσο το αντικείμενό μας κινείται, θα αφήνει και ένα αντίστοιχο ίχνος από κάτω του.



Έλεγχος: εντολές που καθορίζουν πότε θα τρέξει ένα σύνολο εντολών ή πόσες φορές θα εκτελεστεί το συγκεκριμένο σύνολο εντολών. Σύμφωνα με το παράδειγμα που φαίνεται στην επόμενη εικόνα, μόλις πατηθεί η πράσινη σημαία, το αντικείμενό μας θα κινηθεί κατά δέκα βήματα δύο φορές συνεχόμενα. Θα κινηθεί δηλαδή για 20 βήματα.



Αισθητήρες: εντολές που επιτρέπουν στο αντικείμενο να αντιλαμβάνεται το περιβάλλον του, όπως π.χ. το αν αγγίζει άλλα αντικείμενα ή χρώματα, και να αντιδρά ανάλογα. Στο παράδειγμα που ακολουθεί, μόλις πατηθεί η πράσινη σημαία, ο χαρακτήρας μας θα κινηθεί κατά 10 βήματα και αν μετά την κίνησή του αγγίζει το όριο της σκηνής, τότε θα στρίψει δεξιόστροφα κατά 15 μοίρες.



Τελεστές: εντολές που βοηθούν τα αντικείμενα να κάνουν μαθηματικές πράξεις, συγκρίσεις, υπολογισμούς κτλ. Στο παράδειγμα που ακολουθεί, το αντικείμενό μας θα εμφανίσει ένα μπαλονάκι με το αποτέλεσμα της πράξης, δηλαδή θα εμφανίσει τον αριθμό 6.



Μεταβλητές: εντολές που αφορούν τιμές που αποθηκεύουμε για τη συνέχεια του προγράμματος, όπως π.χ. το σκορ, ή τις ζωές του χρήστη σε ένα παιχνίδι. Στο παράδειγμα που ακολουθεί, όταν ο χρήστης κάνει κατά λάθος κλικ πάνω στο δράκο, χάνει μια ζωή.



Η κατηγοριοποίηση των εντολών με βάση τα χρώματα, θα σας διευκολύνει πολύ είτε για να βρείτε την εντολή που θέλετε να χρησιμοποιήσετε είτε για να κατανοήσετε τις εντολές που περιέχει ένα έτοιμο έργο του Scratch.

Για να καθαρίσουμε το χώρο του σεναρίου από τις διάφορες ε-
ντολές που εισάγαμε πειραματιζόμενοι, κάνουμε δεξί κλικ σε
κάποιο κενό σημείο του χώρου του σεναρίου και επιλέγουμε
«καθάρισε τα όλα» από το μενού. Είμαστε έτοιμοι για να προ-
χωρήσουμε παρακάτω.



Κεφάλαιο 4: Αντικείμενα

Σε αυτό το κεφάλαιο:

- 4.1 Το σκηνικό του Scratch
- 4.2 Σχεδιάζοντας ένα σκηνικό
- 4.3 Σχεδιάζοντας αντικείμενα

«Το κρίσιμο στοιχείο της δημιουργικότητας είναι
το να μη φοβάσαι να αποτύχεις»
(Edwin Herbert Land)



4.1 Το σκηνικό του Scratch

Στο κεφάλαιο αυτό θα ασχοληθούμε σε μεγαλύτερο βάθος με τη σχεδίαση σκηνικών και αντικειμένων. Θα μάθετε πώς να δημιουργείτε σκηνικά, χρησιμοποιώντας έτοιμες εικόνες ή ζωγραφίζοντας από την αρχή το δικό σας σκηνικό, θα μάθετε να δημιουργείτε αντικείμενα, και διαφορετικές ενδυμασίες. Τέλος θα δούμε πώς μπορείτε να συνδυάζετε τα παραπάνω με διάφορους ήχους, για ένα πιο ζωντανό αποτέλεσμα. Κάθε φορά που συναντάτε κάτι καινούργιο, καλό θα ήταν, να το δοκιμάζετε στην πράξη. Θα έχετε τη δυνατότητα έτσι, να κατανοήσετε καλύτερα τις λειτουργίες του Scratch αλλά και να πειραματιστείτε με τις πολλές δυνατότητες του.

4.2 Σχεδιάζοντας ένα σκηνικό

Το σκηνικό είναι στην ουσία το φόντο που βρίσκεται στην σκηνή μας το οποίο είναι αρχικά λευκό. Το Scratch μας δίνει τη δυνατότητα να εισάγουμε είτε ένα έτοιμο σκηνικό είτε να ζωγραφίσουμε ένα νέο από την αρχή, όπως είδαμε στο προηγούμενο κεφάλαιο.

Παρατηρείστε προσεκτικά ότι στην κάτω αριστερή πλευρά, της εικόνας που ακολουθεί, εμφανίζεται το εικονίδιο **σκηνικό**. Αυτό θα πρέπει να επιλέγετε όταν θέλετε να επεξεργαστείτε τη σκηνή πάνω στην οποία θα διαδραματιστεί το παιχνίδι σας.



4.2.1 Δημιουργώντας το δικό μας σκηνικό

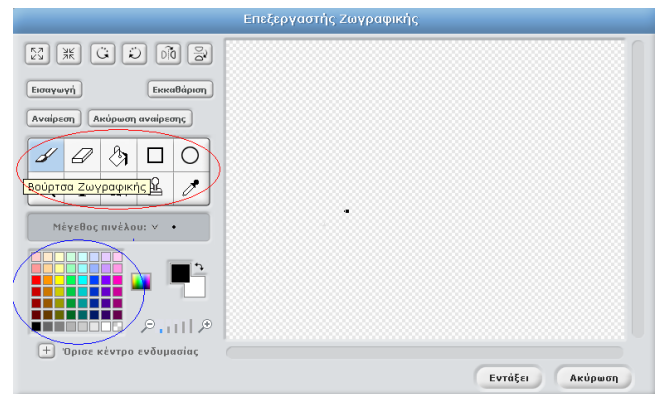
Για να αρχίσουμε να δημιουργούμε το πρώτο δικό μας σκηνικό, πατήστε το εικονίδιο Σκηνικό και στη συνέχεια πατήστε το εικονίδιο Υπόβαθρα.



Εμφανίζονται 3 επιλογές με 3 εικονίδια, η «ζωγραφική», η «εισαγωγή» και η «κάμερα».



Πατώντας το εικονίδιο ζωγραφική **Ζωγραφική** εμφανίζεται στην οθόνη μας ένα νέο παράθυρο με τίτλο Επεξεργαστής Ζωγραφικής, που μας δίνει τη δυνατότητα να σχεδιάσουμε τα σκηνικά του έργου μας από μηδενική βάση. Αν έχετε δουλέψει με άλλα προγράμματα επεξεργασίας εικόνων, όπως με τη Ζωγραφική των Windows, οι δυνατότητές του θα σας φανούν οικείες.



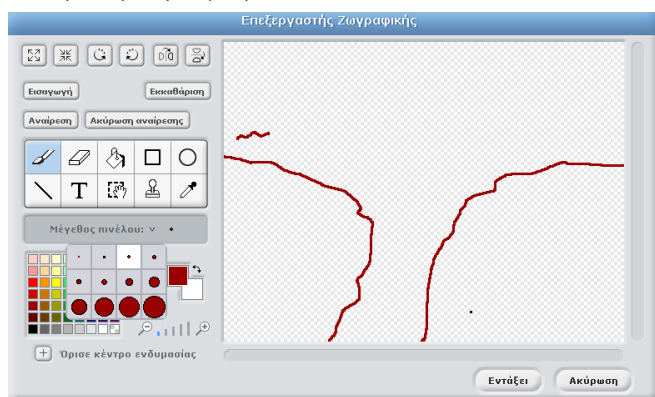
Στο λευκό φόντο (δεξιά) είναι ο χώρος στον οποίο μπορείτε να ζωγραφίσετε τα σκηνικά σας. Η μαύρη κουκίδα που εμφανίζεται δείχνει τη θέση του ποντικιού. Οι δυνατότητες επεξεργασίας

που έχουμε, εμφανίζονται στην αριστερή πλευρά του παραθύρου. Κάθε φορά που τοποθετείτε το ποντίκι σας πάνω από κάθε επιλογή, εμφανίζεται η ονομασία αυτής της επιλογής ώστε να μπορείτε να την αναγνωρίζετε ευκολότερα.

Βούρτσα ζωγραφικής:  επιλέξτε τη βούρτσα και δημιουργήστε ελεύθερα το δικό σας σχέδιο, έχοντας επιλέξει το επιθυμητό χρώμα από την παλέτα των χρωμάτων. Αν επιθυμείτε διαφορετικό πάχος στη βούρτσα σας, μπορείτε να επιλέξετε το μέγεθος του πινέλου.

Μέγεθος πινέλου: ▾ •

Στην επόμενη εικόνα έχοντας επιλέξει την βούρτσα, ένα συγκεκριμένο μέγεθος πινέλου και ως χρώμα το καφέ δημιουργούμε ένα μονοπάτι. Δοκιμάστε το και εσείς, είναι πολύ εύκολο και μια καλή αρχή για να αρχίσετε να κατανοείτε τις λειτουργίες του Επεξεργαστή Ζωγραφικής του Scratch.

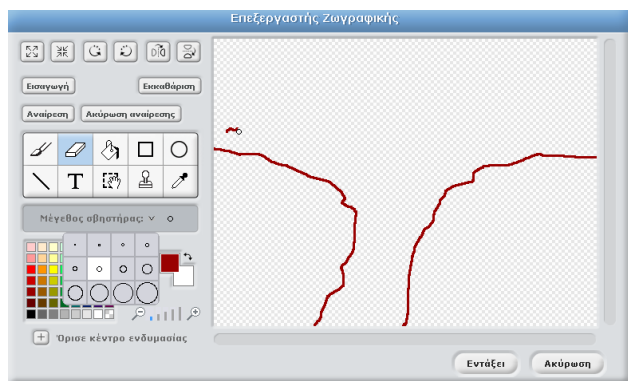


Σβηστήρα:  με αυτή μπορείτε να σβήσετε ότι έχετε ζωγραφίσει εφόσον δεν σας ικανοποιεί. Το μέγεθος της σβηστήρας καθορίζεται από το αντίστοιχο εικονίδιο

Μέγεθος σβηστήρας: ▾ ○

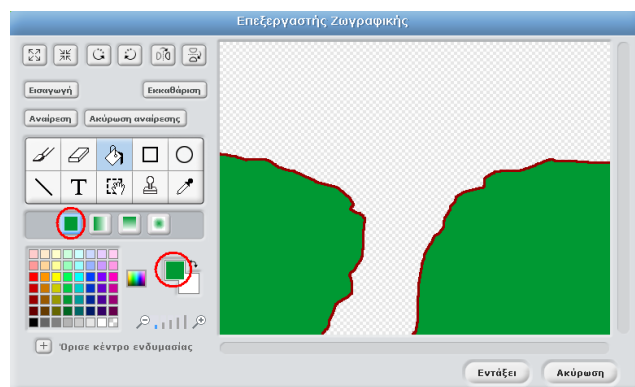
(παρατηρήστε ότι ανάλογα το επιλεγμένο εργαλείο, αλλάζουν οι επιλογές κάτω από την μπάρα εργαλείων)

Επιλέξτε τη σβήστρα και κατάλληλο μέγεθος και σβήστε ένα κομμάτι στο οποίο έχετε κάνει λάθος. Στην επόμενη εικόνα φαίνεται ότι έχοντας επιλέξει ένα μικρό μέγεθος σβηστήρας, σβήνουμε μία γραμμή που είχαμε τοποθετήσει σε λάθος θέση.

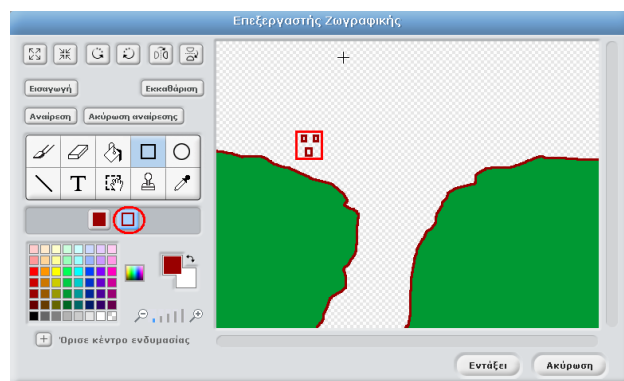


Εργαλείο γεμίσματος:  μπορείτε να γεμίσετε με χρώμα μια κλειστή περιοχή του σκηνικού που οριοθετείται από γραμμές. Δείτε στην επόμενη εικόνα πως χρωματίστηκε το εσωτερικό

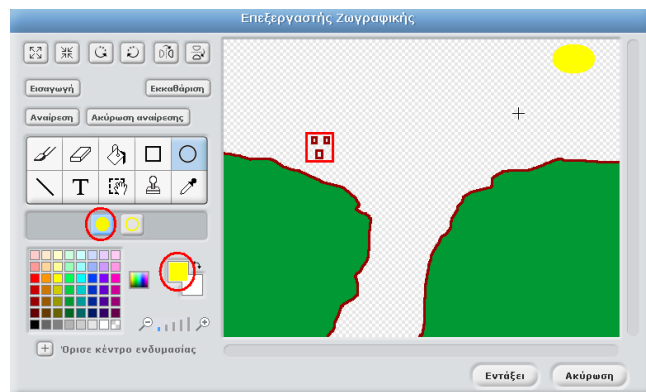
των δύο περιοχών που οριοθετήσαμε προηγουμένως με τη χρήση της βούρτσας με πράσινο χρώμα. Υποτίθεται ότι οι πράσινες περιοχές είναι τα λιβάδια μας.



Εργαλείο παραλληλογράμμου:  Επιλέγοντας το εργαλείο του παραλληλογράμμου μπορείτε να δημιουργήσετε παραλληλόγραμμα είτε γεμισμένα με χρώμα είτε σχεδιάζοντας μόνο το περίγραμμά τους. Στην επόμενη εικόνα έχοντας επιλέξει να δημιουργήσουμε το περίγραμμα ενός παραλληλογράμμου και προσδιορίζοντας ως χρώμα το κόκκινο δημιουργήσαμε ένα σπιτάκι. Δοκιμάστε το και εσείς.

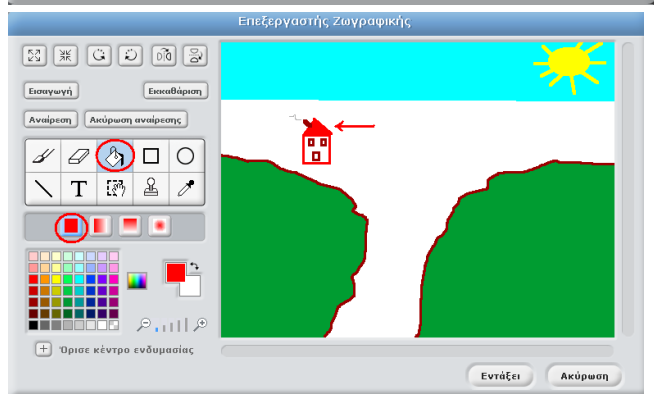
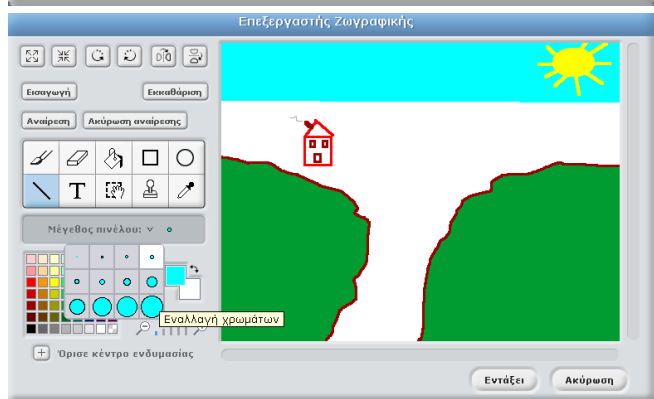
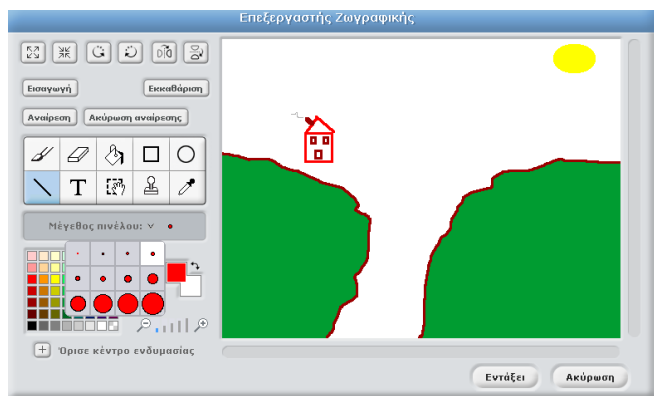


Εργαλείο έλλειψης:  Το εργαλείο της έλλειψης λειτουργεί ακριβώς όπως και το εργαλείο του παραλληλογράμμου με την διαφορά ότι μπορείτε να δημιουργείτε ελλείψεις γεμισμένες ή όχι με χρώμα. Στην επόμενη εικόνα δημιουργήσαμε μια κίτρινη έλλειψη γεμισμένη με κίτρινο χρώμα για να αναπαραστήσουμε τον ήλιο.

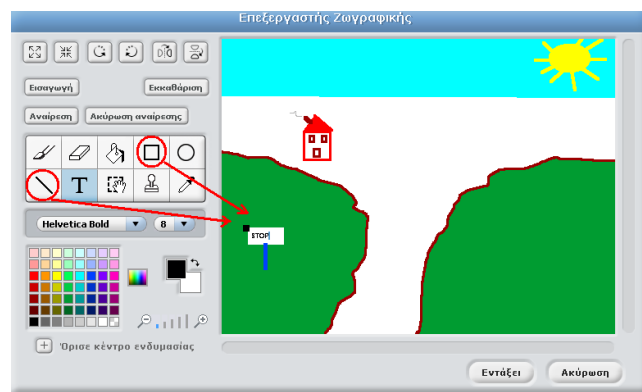


Εργαλείο γραμμής:  απλά, σχεδιάζουμε γραμμές κρατώντας πατημένο το αριστερό κουμπί του ποντικιού όσο προσπαθούμε

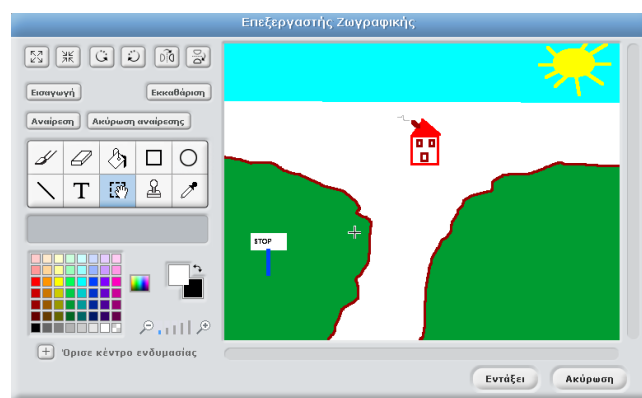
να επιλέξουμε κατάλληλη θέση και μέγεθος. Στο δικό μας παράδειγμα, επιλέξτε κάποιο μέγεθος γραμμής από το την επιλογή «μέγεθος πινέλου», διαλέξτε το κόκκινο χρώμα για τη γραμμή μας και στην συνέχεια τοποθετήστε μια σκεπή στο σπιτάκι σχεδιάζοντας δυο γραμμές. Επιπρόσθετα, αλλάξτε τα στοιχεία του πάχους και του χρώματος της γραμμής για να δημιουργήσετε έναν ουρανό. Μήπως να προσθέσουμε και στον ήλιο ηλιαχτίδες;



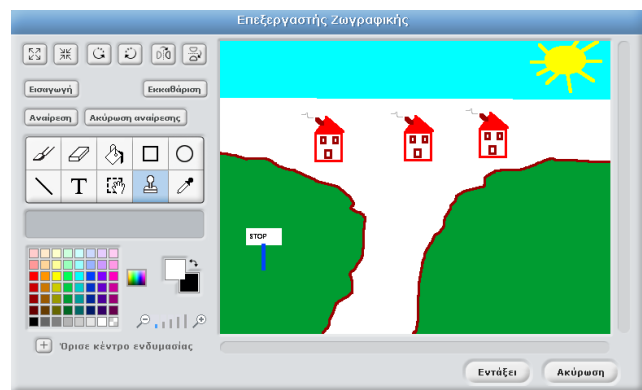
Εργαλείο κειμένου : Επιλέγοντας αυτό το εργαλείο μπορείτε να εισάγετε κείμενο στο σκηνικό σας. Στην επόμενη εικόνα δημιουργήσαμε μια ταμπέλα STOP η οποία μας απαγορεύει να πατάμε στην πράσινη περιοχή! Δοκιμάστε και εσείς να γράψετε κάτι στο σκηνικό που έχετε δημιουργήσει μέχρι τώρα. Κάτω από την μπάρα των εργαλείων σας δίνεται η δυνατότητα να αλλάξετε τον τύπο αλλά και το μέγεθος της γραμματοσειράς.



Εργαλείο επιλογής : Με αυτό το εργαλείο μπορείτε να επιλέξετε ένα τμήμα του σκηνικού σας και στη συνέχεια είτε να το διαγράψετε πατώντας απλά delete, είτε να το μετακινήσετε σέρνοντάς το σε άλλο σημείο της οθόνης. Όπως μπορείτε να δείτε στην επόμενη εικόνα μετακινήσαμε το σπιτάκι λίγο πιο δεξιά.



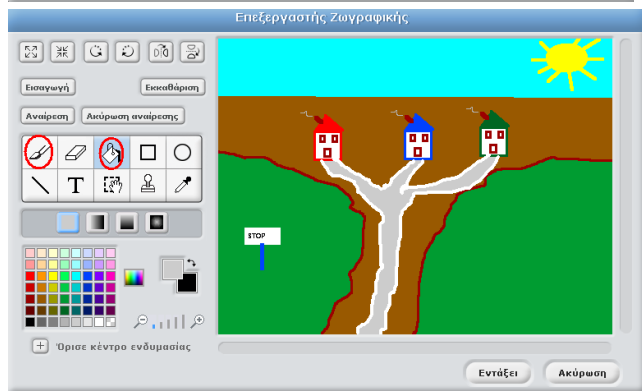
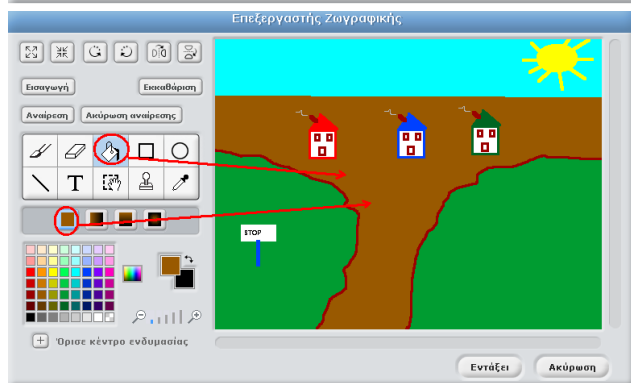
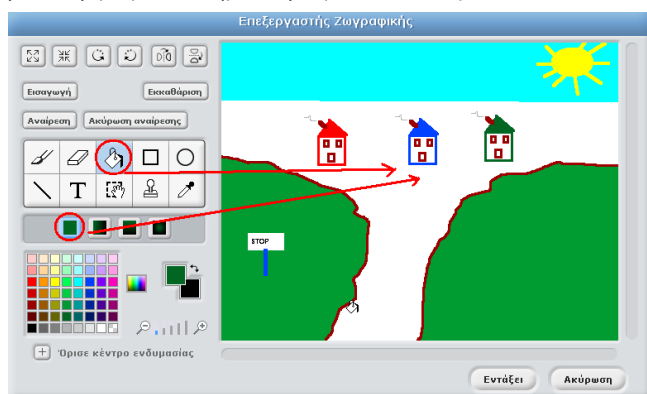
Εργαλείο σφραγίδας : Επιλέγοντας αυτό το εργαλείο μπορούμε να φτιάξουμε πολλά αντίγραφα ενός τμήματος του σκηνικού μας. Σέρνοντας το ποντίκι, επιλέγετε την περιοχή που θέλετε να αντιγράψετε. Μόλις αφήσετε το αριστερό κουμπί του ποντικιού, το τμήμα αυτό ακολουθεί το δείκτη του ποντικιού (σαν να είναι σφραγίδα...). Κάθε φορά που πατάμε ένα απλό κλικ, το επιλεγμένο τμήμα «τυπώνεται» πάνω στο σκηνικό μας. Αυτός είναι ένας εύκολος τρόπος για να δημιουργήσουμε πολλά σπιτάκια! Μη ξεχνάτε ότι πρέπει να επιλέξετε άλλο εργαλείο για να σταματήσει η συγκεκριμένη λειτουργία της σφραγίδας.



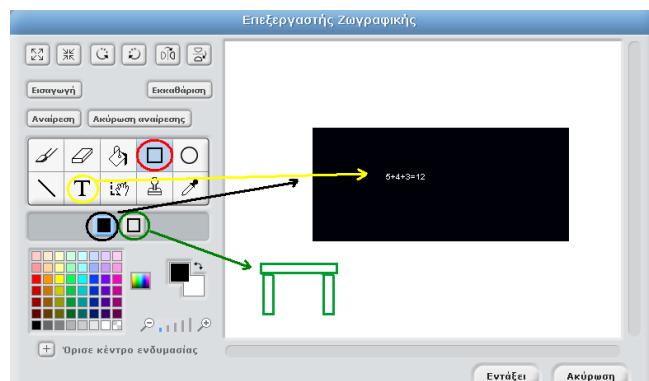
Εργαλείο επιλογής χρωμάτων ή σταγονόμετρο : Έχουν υπάρξει φορές στο παρελθόν που θέλατε να επαναχρησιμοποιή-

ήσετε ένα χρώμα που είχατε επιλέξει προηγουμένως και δε θυμάστε ποιο ακριβώς είναι; Ή να χρησιμοποιήσετε ένα χρώμα το οποίο ήδη υπάρχει πάνω στην εικόνα που επεξεργάζεστε; Δεν είναι τις περισσότερες φορές σχεδόν αδύνατο να βρείτε το αντίστοιχο χρώμα ψάχνοντας μόνο στην παλέτα των χρωμάτων; Με το εργαλείο επιλογής χρωμάτων, μπορούμε να ορίσουμε ως χρώμα οποιοδήποτε χρώμα βρίσκεται πάνω στο σκηνικό, αρκεί να πατήσουμε ένα κλικ πάνω σε αυτό. Αμέσως, μετά μπορούμε να χρησιμοποιήσουμε το συγκεκριμένο χρώμα για να δημιουργήσουμε μια γραμμή ή να γεμίσουμε με χρώμα μια κλειστή περιοχή.

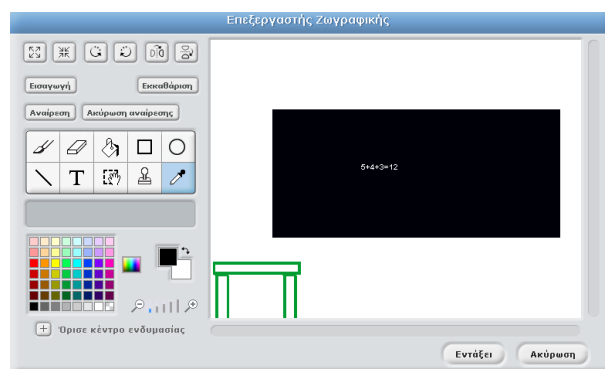
Ας ολοκληρώνουμε το σκηνικό μας με τρία ακόμα βήματα. Στην πρώτη από τις επόμενες εικόνες χρησιμοποιώντας το εργαλείο γεμίσματος χρωματίζουμε τα δύο νέα σπιτάκια μας με μπλε και πράσινο χρώμα αντίστοιχα. Στην δεύτερη εικόνα χρησιμοποιώντας το εργαλείο γεμίσματος χρωματίζουμε το μονοπάτι με χρώμα καφέ. Τέλος, χρησιμοποιώντας, τη βούρτσα και το εργαλείο γεμίσματος δημιουργούμε άλλα τρία μονοπάτια.



Τι λέτε για ένα ακόμη παράδειγμα; Στην εικόνα που ακολουθεί δημιουργούμε έναν πίνακα και ένα θρανίο (με κύκλους σημειώνονται τα εργαλεία που απαιτούνται για τη συγκεκριμένη δημιουργία).



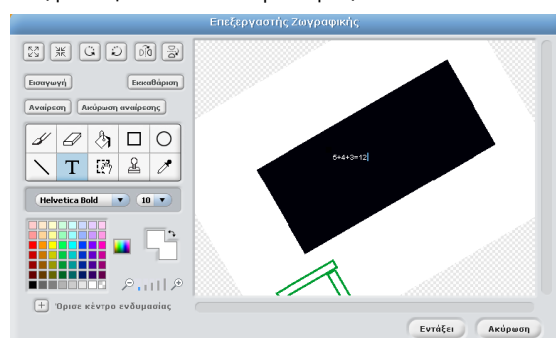
Πατήστε το **εικονίδιο μεγέθυνσης**  που βρίσκεται πάνω αριστερά στον επεξεργαστή ζωγραφικής. Μεγαλώνει όλο το σκηνικό μας! Δηλαδή έχουμε τη δυνατότητα να σχεδιάσουμε κάτι σε μικρογραφία και στη συνέχεια να το μεγαλώσουμε.



Αν θέλετε να επανέλθει το σκηνικό σας στην αρχική του κατάσταση, πατήστε το **εικονίδιο της αναιρέσης** . Είναι πραγματικά το πιο «χρήσιμο» εικονίδιο κατά τη διάρκεια του σχεδιασμού μας! Μας επιτρέπει να αναιρέσουμε τις αλλαγές που έχουμε κάνει στο σχέδιό μας και να το επαναφέρουμε σε προηγούμενη του κατάσταση. Δεν είναι και λίγες οι φορές που κάνουμε λάθη όταν ζωγραφίζουμε...

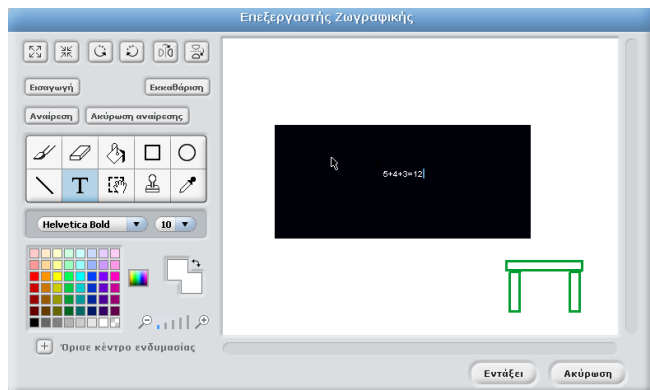
Πατήστε το **εικονίδιο της σμίκρυνσης**  για να μειώσετε το μέγεθος του σκηνικού σας. Έχει δηλαδή την ακριβώς αντίθετη λειτουργία με το εικονίδιο της μεγέθυνσης.

Πατήστε το **εικονίδιο της αριστερόστροφης περιστροφής**  για να «στρίψετε» το σκηνικό προς τα αριστερά. Αν δείτε ότι κάτι δεν πάει καλά, πατήστε το εικονίδιο της αναιρέσης. Δείτε πως μετατρέπεται το σκηνικό μας:

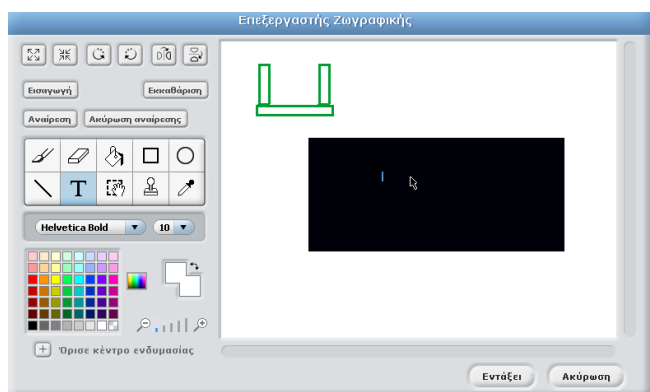


Πατήστε το **εικονίδιο της δεξιόστροφης περιστροφής** για «στρίψετε» το σκηνικό προς τα δεξιά: η αντίθετη εργασία από την αριστερόστροφη περιστροφή.

Πατήστε το **εικονίδιο της οριζόντιας αναστροφής** για να δείτε τον «καθρέπτη» της εικόνας που έχετε δημιουργήσει. Παρατηρήστε στην επόμενη εικόνα ότι το θρανίο μετακινήθηκε στα δεξιά και ο πίνακας στα αριστερά.



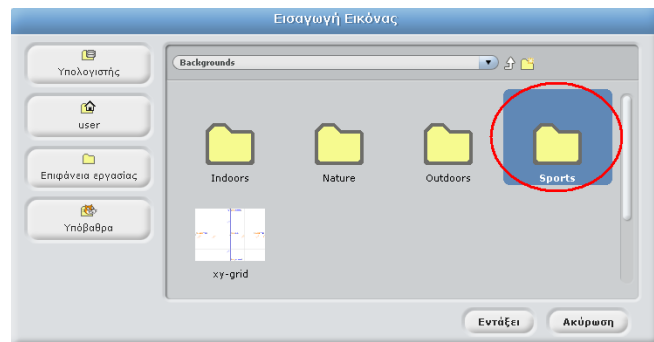
Πατήστε το **εικονίδιο της κάθετης αναστροφής** για να αναποδογυρίσετε το σκηνικό σας (σα να βάλουμε το καθρέπτη από πάνω!). Δείτε πως μετατράπηκε το σκηνικό μας:



Με το παράδειγμα αυτό ολοκληρώσαμε μια πρώτη περιήγηση στον επεξεργαστή ζωγραφικής. Θα δούμε επιπλέον παραδείγματα παρακάτω.

4.2.2 Εισαγωγή έτοιμου σκηνικού

Όπως είδαμε και στο προηγούμενο κεφάλαιο, το Scratch παρέχει επίσης μία πληθώρα από έτοιμα σκηνικά. Για να εισάγουμε ένα έτοιμο σκηνικό, πηγαίνουμε στην καρτέλα **Υπόβαθρα** του σκηνικού και κάνουμε «κλικ» στο **εικονίδιο Εισαγωγή** για να εμφανιστεί το παράθυρο της εισαγωγής υποβάθρου από τη βιβλιοθήκη.



Έστω ότι θέλουμε ως υπόβαθρο ένα γήπεδο στο οποίο στη συνέχεια θα δημιουργήσουμε ένα παιχνίδι ποδοσφαίρου. Πηγαίνουμε στο φάκελο Sports (αθλητικά), «εξερευνούμε» τις εναλλακτικές μας, επιλέγουμε την επιθυμητή και κάνουμε κλικ στο «Εντάξει».



Το σκηνικό μας είναι πλέον έτοιμο για εισάγουμε τα υπόλοιπα αντικείμενα του έργου μας.



{Σημείωση: Ένας άλλος τρόπος για να εισάγετε ένα έτοιμο σκηνικό είναι να εκκινήσετε τον Επεξεργαστή ζωγραφικής και να κάνετε "κλικ" στην επιλογή **Εισαγωγή**. Στη συνέχεια αρκεί να ακολουθήσετε τα ίδια βήματα. Επιπλέον, όμως, μπορείτε να αλλάξετε το προεπιλεγμένο σκηνικό}

Τι συμβαίνει όμως αν θέλουμε το παιχνίδι μας να εκτυλιχθεί σε πολλά σκηνικά; Αυτό δε συμβαίνει στο Pacman; Ένα παιχνίδι ράβι δεν περιέχει πολλές πίστες;

Καταρχάς μπορούμε να δημιουργήσουμε νέα σκηνικά χρησιμοποιώντας τις προηγούμενες επιλογές (ζωγραφική, εισαγωγή και κάμερα). Τα νέα υπόβαθρα εμφανίζονται στο τέλος της λίστας. Αν όμως θέλετε να βασίσετε τα νέα υπόβαθρα σε κάποια που έχετε ήδη δημιουργήσει, πατήστε το κουμπί της **Αντιγραφής (A)** στο υπόβαθρο που θέλετε να αντιγράψετε και στη συνέχεια πατήστε τη κουμπί της **Διόρθωσης (B)** για να το αλλάξετε. Το κουμπί της διόρθωσης, σας δίνει τη δυνατότητα να αλλάξετε ανά πάσα στιγμή όλα τα υπόβαθρα που έχουν εισαχθεί

στο έργο σας. Τέλος αν θέλετε να διαγράψετε ένα σκηνικό, αρκεί να πατήσετε το εικονίδιο **X** που βρίσκεται δίπλα από το κουμπί της αντιγραφής.

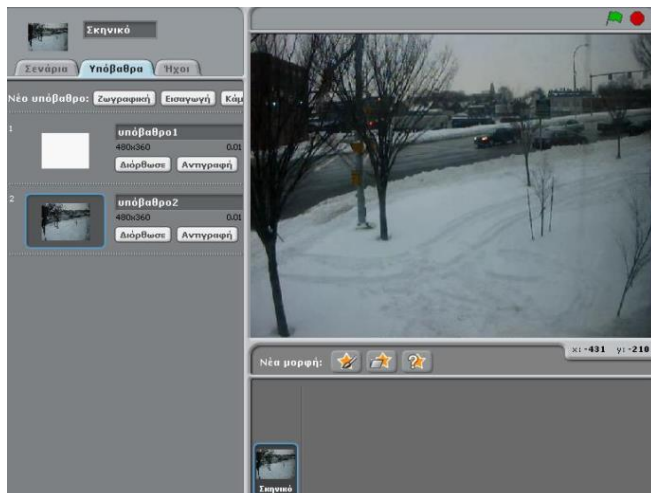


4.2.3 Βγάζοντας φωτογραφίες για σκηνικό

Αν έχετε κάμερα στον υπολογιστή σας, μπορείτε να βγάλετε μια φωτογραφία μέσω του Scratch και να την προσθέσετε ως σκηνικό στο έργο σας. Επιλέξτε για τη δημιουργία του νέου υποβάθρου το κουμπί **Κάμερα**. Εμφανίζεται ένα νέο παράθυρο που παρουσιάζει την εικόνα που λαμβάνει η κάμερα. Έστω ότι γυρνάμε την κάμερά μας προς ένα χιονισμένο τοπίο.



Πατώντας «Έγινε» έχουμε εισάγει ως σκηνικό τη συγκεκριμένη φωτογραφία.



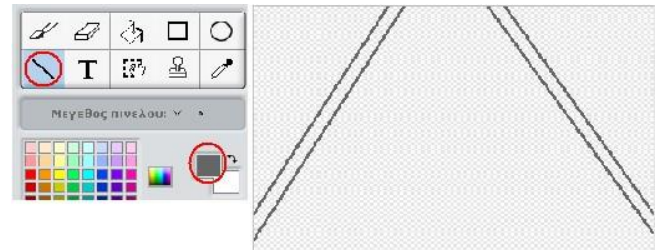
Προφανώς, μπορείτε να χρησιμοποιήσετε την κάμερα για να εισάγετε ως σκηνικό το δωμάτιό σας, ένα προσωπικό σας αντικείμενο, το πρόσωπο των φίλων σας ή και τους γονείς σας!

4.2.4 Ακόμη ένα παράδειγμα δημιουργίας σκηνικού

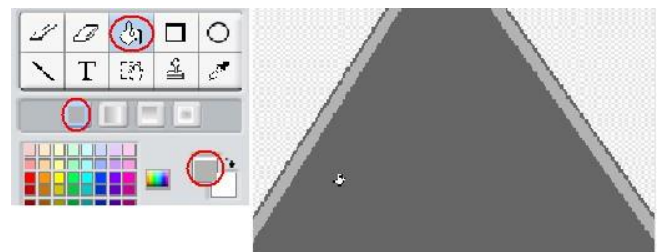
Ας περάσουμε πάλι από τη θεωρία στην πράξη και ας σχεδιάσουμε ένα δικό μας, λίγο πιο σύνθετο, σκηνικό αξιοποιώντας όλες τις προηγούμενες εντολές.

Έστω λοιπόν ότι θέλουμε να σχεδιάσουμε ως σκηνικό έναν δρόμο με μία διάβαση. Κάνουμε «κλικ» στο εικονίδιο Σκηνικό, κατόπιν επιλέγουμε **Υπόβαθρο** και στη συνέχεια επιλέγουμε **Ζωγραφική** για να δημιουργήσουμε από την αρχή το σκηνικό μας. Εμφανίζεται το παράθυρο του Επεξεργαστή Ζωγραφικής.

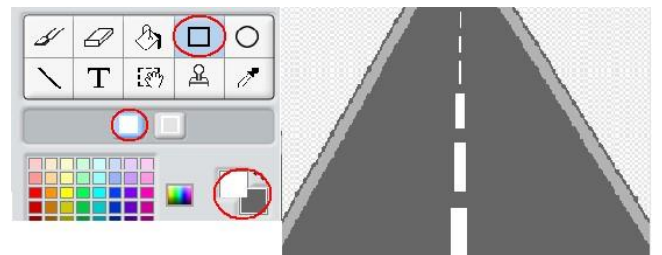
Αρχικά θα τραβήξουμε με τη βοήθεια του εργαλείου **γραμμή** και με γκρι χρώμα τέσσερις γραμμές ανά δύο παράλληλες μεταξύ τους για να χαράξουμε τα όρια του δρόμου με το πεζοδρόμιο. Φροντίστε οι γραμμές να εκτείνονται μέχρι τις άκρες του καμβά έτσι ώστε να μπορείτε να «γεμίσετε» με χρώματα τις διάφορες κλειστές περιοχές που σχηματίζονται.



Με το εργαλείο γεμίσματος και επιλέγοντας δύο άλλες αποχρώσεις του γκρι θα «γεμίσουμε» με χρώμα τον δρόμο και την άκρη του πεζοδρομίου.

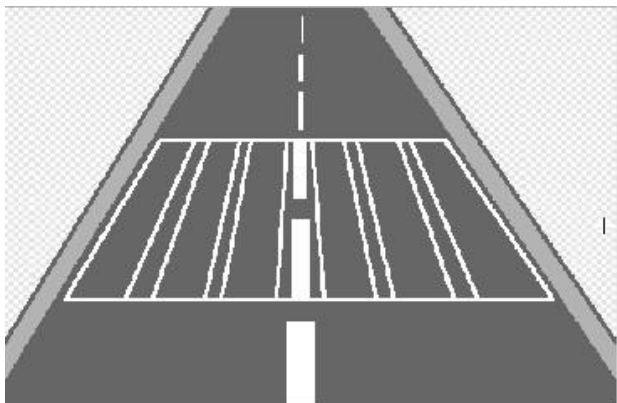


Στη συνέχεια επιλέγουμε το εργαλείο παραλληλογράμμου, κάνουμε «κλικ» στο γέμισμα (ώστε το παραλληλόγραμμο να είναι χρωματισμένο στο εσωτερικό του) και επιλέγουμε χρώμα άσπρο προκειμένου να κάνουμε τη διαγράμμιση του δρόμου. Δημιουργούμε παραλληλόγραμμο στο κέντρο του δρόμου, το ένα πάνω από το άλλο, μειώνοντας το μήκος και το πλάτος τους καθώς φτάνουμε στην κορυφή του καμβά για να δημιουργήσουμε της αίσθηση του βάθους στο υπόβαθρό μας.



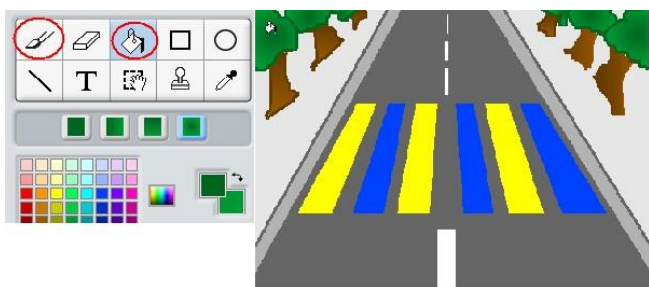
Ας σχεδιάσουμε τώρα την διάβαση. Επιλέγοντας πάλι το εργαλείο γραμμής με το ίδιο μέγεθος πινέλου αλλά αυτή τη φορά με χρώμα άσπρο, θα σχεδιάσουμε περίπου στη μέση του δρόμου ένα τραπέζιο το οποίο θα μας βοηθήσει να ζωγραφίσουμε τη διάβαση.

Τραβάμε δύο γραμμές οι οποίες θα είναι παράλληλες με τα δύο πεζοδρόμια και τις ενώνουμε μεταξύ τους με δύο άλλες οριζόντιες παράλληλες γραμμές. Έπειτα θα τραβήξουμε ορισμένες νέες γραμμές από την μεγάλη βάση του τραπεζίου στην μικρή για να μπορέσει να πάρει η διάβαση την τελική της μορφή.

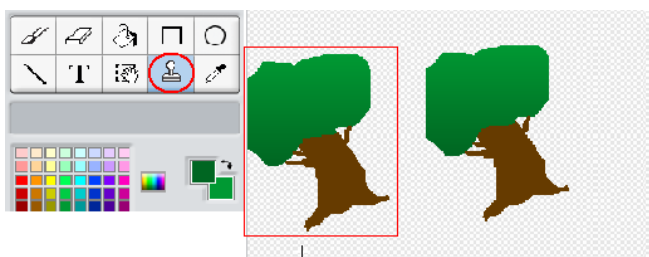


Κατόπιν γεμίζουμε με μπλε και κίτρινο χρώμα εναλλάξ τα κουτιά που δημιουργήθηκαν με χρήση του εργαλείου γεμίσματος. Τέλος, «γεμίζουμε» τις άσπρες γραμμές που «περίσσεψαν» με το γκρι χρώμα του δρόμου. Ο δρόμος μας με την διάβαση μόλις τελείωσε!

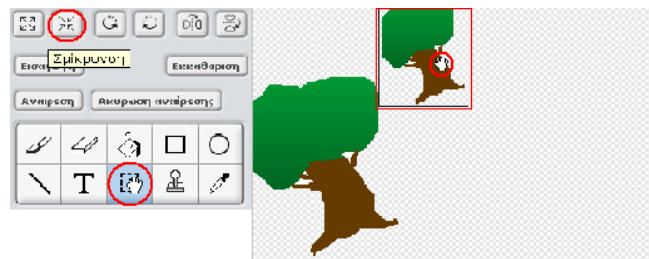
Ας κάνουμε όμως λίγο πιο «πλούσιο» το σκηνικό μας. Στα αριστερά και στα δεξιά του δρόμου μπορούμε να ζωγραφίσουμε δέντρα. Αυτό μπορεί να γίνει χρησιμοποιώντας το εργαλείο βούρτσα για να σχεδιάσουμε τον κορμό και τα φύλλα των δέντρων -με τα κατάλληλα χρώματα- και το εργαλείο γεμίσματος, όπως προηγουμένως.



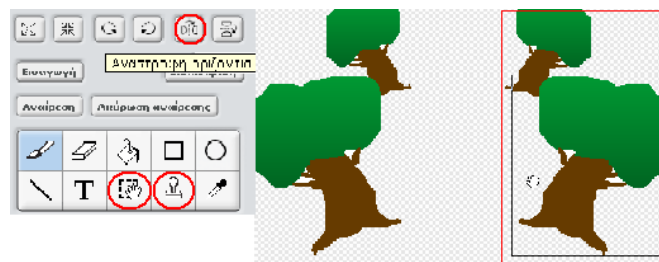
Εναλλακτικά, υπάρχει ένας πιο εύκολος και γρήγορος τρόπος για να «εμπλουτίσετε» το σκηνικό με τα δέντρα. Αρχικά ζωγραφίστε ένα δέντρο. Στη συνέχεια κάντε «κλικ» στο εργαλείο σφραγίδας και επιλέξτε το δέντρο όπως φαίνεται στην επόμενη εικόνα.



Έπειτα, κάντε «κλικ» στο εργαλείο Επέλεξε για να επιλέξετε το αντίγραφο του δέντρου. Κάντε σμίκρυνση του επιλεγμένου δέντρου με τη βοήθεια της αντίστοιχης επιλογής. Όσες περισσότερες φορές πατήσετε δεξί κλικ πάνω του τόσο πιο μικρό θα γίνει. Τοποθετείστε το ποντίκι πάνω στο αντίγραφο και σύρετε το δέντρο όπου εσείς θέλετε.

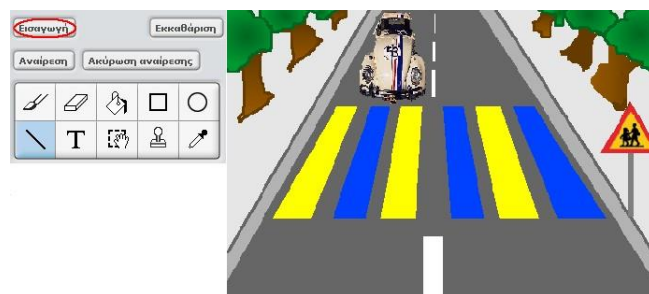


Για να «στολίσετε» τώρα και την δεξιά πλευρά του πεζοδρομίου δεν χρειάζεται να ζωγραφίσετε ξανά τα δέντρα. Το μόνο που χρειάζεται είναι να δημιουργήσετε από μία σφραγίδα των δέντρων, να επιλέξετε το κάθε αντίγραφο χωριστά και να κάνετε κλικ στην επιλογή Αναστροφή οριζόντια ούτως ώστε να υπάρχει μία συμμετρία των αριστερών δέντρων με τα δεξιά. Σας μένει να τα τοποθετείτε στο σημείο που θέλετε και το σκηνικό του πεζοδρομίου είναι έτοιμο.



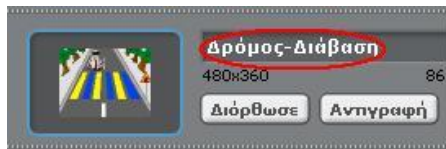
Με αυτόν τον τρόπο, μπορείτε να δημιουργήσετε πολλά αντίγραφα από οτιδήποτε υπάρχει πάνω στον καμβά και να επεξεργαστείτε τα αντίγραφα αυτά κατάλληλα (π.χ. μεγέθυνση, σμίκρυνση, αναστροφή, περιστροφή).

Μία ενδιαφέρουσα επιλογή για την δημιουργία σκηνικών είναι το κουμπί Εισαγωγή με το οποίο μπορούμε να εισάγουμε μία εικόνα από τον υπολογιστή μας στον τρέχοντα καμβά. Έτσι μπορούμε για παράδειγμα να εισάγουμε ένα αυτοκίνητο και να το τοποθετήσουμε πάνω στον δρόμο (προσοχή! όχι πάνω στην διάβαση!) αλλά και ένα σήμα που προειδοποιεί τους διερχόμενους οδηγούς ότι περνούν μαθητές κατά μήκος της διάβασης.



Το σκηνικό μας είναι έτοιμο! Για να ξεχωρίζετε τα διάφορα σκηνικά που σχεδιάζετε μπορείτε να τα μετονομάζετε κάνοντας «κλικ» στον τίτλο του υποβάθρου και πληκτρολογώντας τη δική σας ονομασία για το δημιούργημά σας. Στην παρακάτω εικόνα μετονομάζουμε το υπόβαθρο 2 σε Δρόμος-Διάβαση.





4.3 Σχεδιάζοντας αντικείμενα

4.3.1 Εισαγωγή στο περιβάλλον των αντικειμένων

Όπως συζητήσαμε και στο προηγούμενο κεφάλαιο, τα αντικείμενα στο Scratch αποτελούν τους πρωταγωνιστές των έργων μας. Το πρώτο βήμα κατά τη δημιουργία τους είναι να τους δίνουμε ένα κατάλληλο όνομα ώστε να μπορούμε να τα αναγνωρίζουμε στη συνέχεια. Θυμηθείτε ότι είναι καλό να επιλέγουμε ονόματα σχετικά με το πρόβλημα που έχουμε να επιλύσουμε ώστε εμμέσως να βοηθήσουμε τον τρόπο σκέψης μας.

Ας μετονομάσουμε τη γάτα που εμφανίζεται μόλις ανοίγουμε το Scratch. Επιλέγουμε το εικονίδιο Μορφή1 που βρίσκεται στη λίστα αντικειμένων.



Στη συνέχεια, μεταφέρουμε το δείκτη του ποντικιού στο πάνω τμήμα της οθόνης και κάνουμε κλικ στον υπάρχον όνομα Μορφή1. Πατάμε delete και γράφουμε γάτα, όπως φαίνεται στις εικόνες παρακάτω εικόνες.



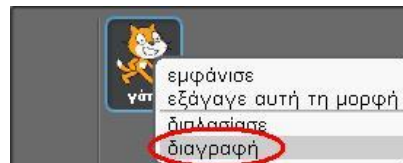
Κάθε αντικείμενο που δημιουργούμε, όπως και το σκηνικό, αποτελείται από τρία στοιχεία: τα Σενάρια, τις Ενδυμασίες και τους Ήχους. Επιλέγοντας το εικονίδιο Σενάρια και μεταφέροντας τις κατάλληλες εντολές από την παλέτα εντολών, προσδιορίζουμε τη συμπεριφορά του αντικειμένου σας. Πατώντας το εικονίδιο Ενδυμασίες μπορούμε να δούμε και να επεξεργαστούμε τις διαθέσιμες ενδυμασίες του αντικειμένου σας ή να δημιουργήσουμε νέες. Οι ενδυμασίες αναφέρονται στις εναλλακτικές μορφές εμφάνισης ενός αντικειμένου. Τέλος, από την επιλογή Ήχοι διαχειριζόμαστε τους ήχους που μπορούν να αναπαράγει κάθε αντικείμενο ξεχωριστά.

4.3.2. Δημιουργία νέων αντικειμένων

Υπάρχουν τρεις διαφορετικοί τρόποι για να δημιουργήσουμε ένα νέο αντικείμενο, παραπλήσιοι με αυτούς της δημιουργίας νέων υποβάρθρων.

Ζωγράφισε νέα μορφή: Φτιάχνουμε λοιπόν ένα μουσικό παιχνίδι και αποφασίζουμε να σχεδιάσουμε μια κιθάρα στην οποία θα προσθέσουμε στη συνέχεια κατάλληλους ήχους.

Αρχικά, επειδή για το σκοπό του έργου μας δεν χρειαζόμαστε τη γάτα θα πρέπει να την διαγράψουμε. Αυτό το πετυχαίνουμε κάνοντας δεξί κλικ πάνω στο αντικείμενο-γάτα στη λίστα αντικειμένων, στο κάτω δεξιό μέρος της οθόνης μας, και στη συνέχεια επιλέγοντας διαγραφή.



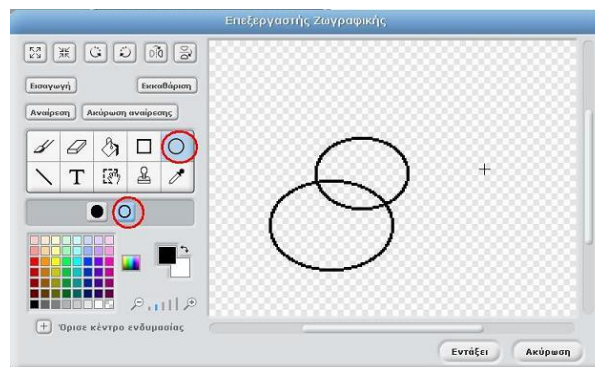
Κάτω από τη σκηνή μας βρίσκονται οι τρεις επιλογές για τη νέα μορφή. Για τη σχεδίαση του αντικειμένου μας επιλέγουμε Ζωγράφισε νέα μορφή.



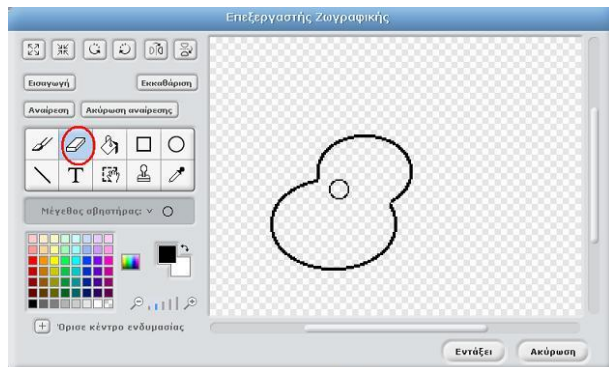
Όπως θα δείτε ανοίγει ο Επεξεργαστής Ζωγραφικής του Scratch. Πάμε να ζωγραφίσουμε την κιθάρα μας!



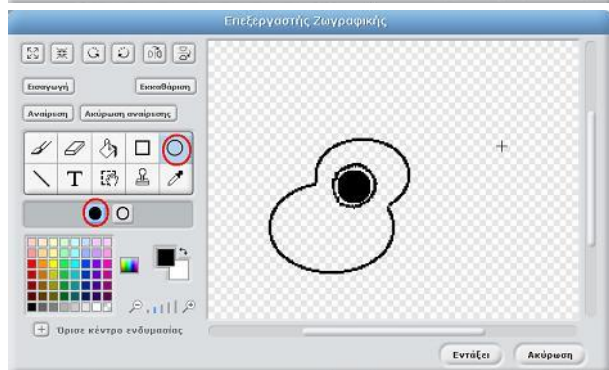
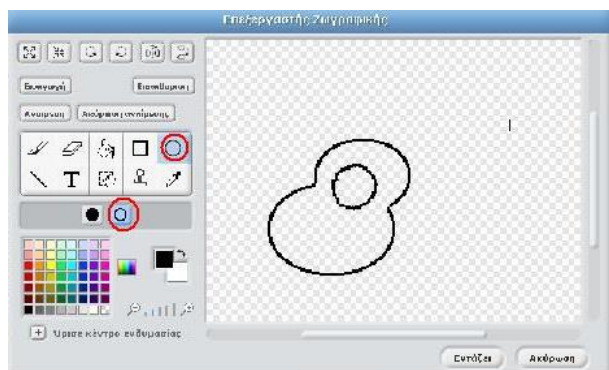
Επιλέγουμε, από την παλέτα των εργαλείων, το εργαλείο της Έλλειψης και στη συνέχεια από τις δυο προτεινόμενες μορφές έλλειψης διαλέγουμε αυτήν που δεν περιέχει χρώμα στο εσωτερικό της. Έτσι σχεδιάζουμε στη βάση του καμβά μας μια μεγάλη έλλειψη, και ακριβώς από πάνω της, μια έλλειψη μικρότερου μεγέθους.



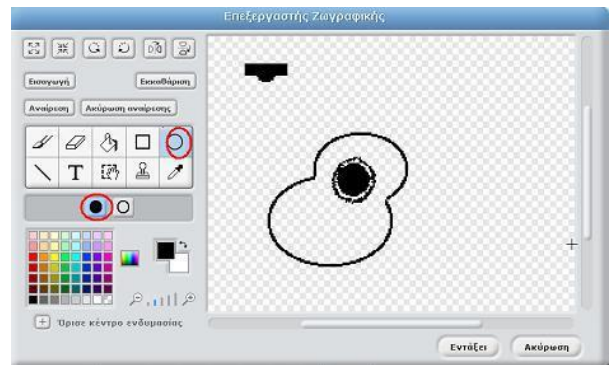
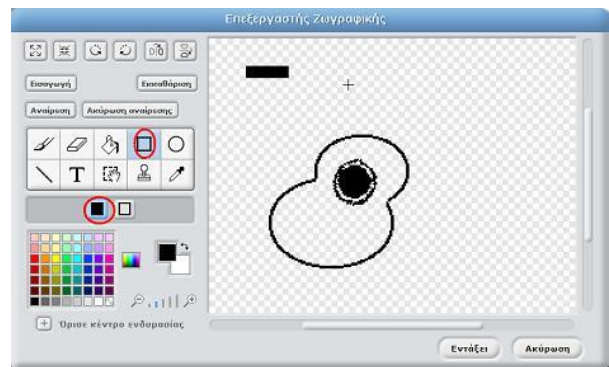
Στη συνέχεια με τη βοήθεια του εργαλείου Σβηστήρας διαγράφουμε την τομή των δυο ελλείψεων όπως φαίνεται στην παρακάτω εικόνα. Πολύ γρήγορα σχεδιάσαμε το εξωτερικό περιγράμμα μιας κιθάρας.



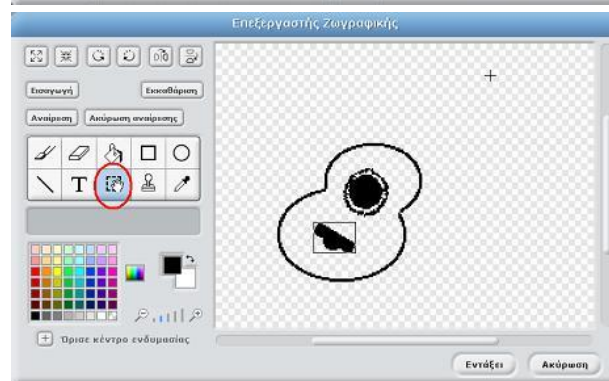
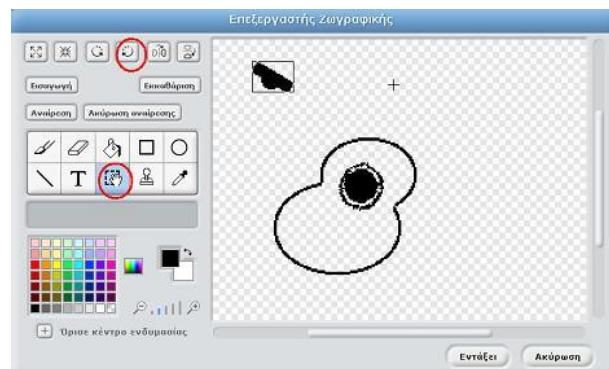
Έπειτα, χρησιμοποιώντας πάλι το εργαλείο της έλλειψης, σχεδιάζουμε έναν κύκλο στο εσωτερικό του περιγράμματος μας, περίπου στο κέντρο του. Στη συνέχεια, επιλέγουμε η έλλειψη μας να περιέχει χρώμα στο εσωτερικό της και ζωγραφίζουμε στο εσωτερικό του κύκλου που πριν λίγο δημιουργήσαμε ένα μικρότερο κύκλο μαύρου χρώματος. Έτσι δημιουργείται ο τυπικός λευκός δακτύλιος που έχουν οι περισσότερες κιθάρες.



Με τη βοήθεια του εργαλείου Παραλληλόγραμμα και επιλέγοντας να είναι γεμάτο με μαύρο χρώμα, σχεδιάζουμε ένα μαύρο ορθογώνιο παραλληλόγραμμα. Αυτό το κάνουμε εξωτερικά του περιγράμματος της κιθάρας για να μπορέσουμε να το επεξεργαστούμε ευκολότερα. Στη συνέχεια, με το εργαλείο Έλλειψη και επιλέγοντας η έλλειψη να είναι γεμάτη με μαύρο χρώμα, ζωγραφίζουμε ακριβώς πάνω στο παραλληλόγραμμα ένα κύκλο με τέτοιο τρόπο ώστε να εξέχει στη κάτω πλευρά του, ένα ημικύκλιο. Οι επόμενες εικόνες δείχνουν το αποτέλεσμα των ενεργειών μας.

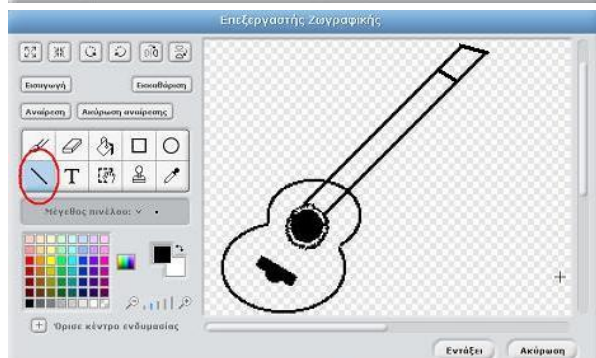
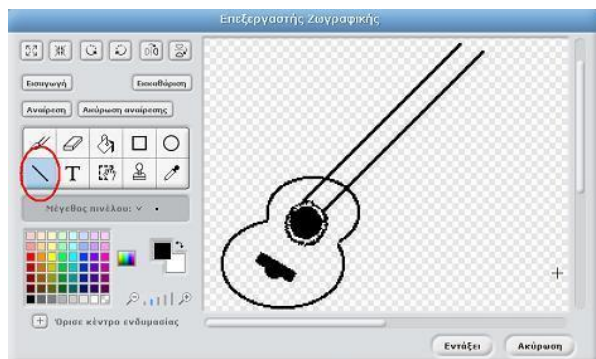


Κατόπιν με το εργαλείο Επιλογή, μαρκάρουμε την περιοχή με το παραλληλόγραμμα και την έλλειψη που μόλις ζωγραφίσαμε. Περιστρέφουμε την επιλογή μας με Δεξιόστροφη περιστροφή κάνοντας κλικ στο ανάλογο εικονίδιο. Χρησιμοποιώντας πάλι το εργαλείο Επιλογή, μετακινούμε το νέο τμήμα στο εσωτερικό του περιγράμματος της κιθάρας. Με αυτόν τον τρόπο προσπαθούμε να δώσουμε λεπτομέρειες, στο σώμα της κιθάρας, έτσι ώστε να είναι πιο κοντά στη μορφή μιας πραγματικής κιθάρας.

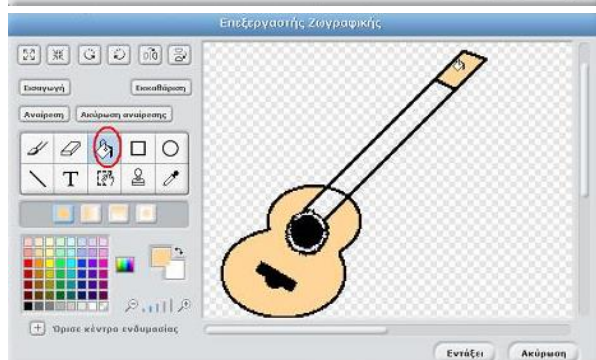
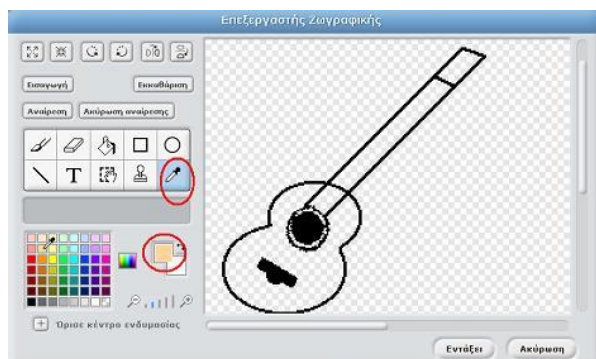


Ας σχεδιάσουμε τώρα το μπράτσο της κιθάρας. Με το εργαλείο γραμμής σχεδιάζουμε δυο παράλληλες γραμμές ξεκινώντας από τον δακτύλιο που βρίσκεται στο εσωτερικό του περιγράμματος της κιθάρας και κατευθυνόμενοι προς το εξωτερικό και

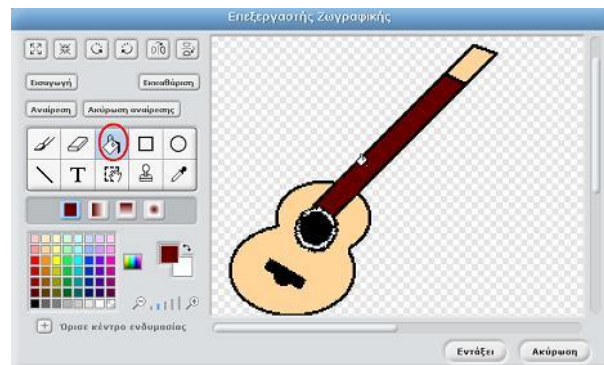
πάνω μέρος του. Έπειτα, ενώνουμε τις δυο παράλληλες γραμμές με δυο κάθετες, προς αυτές, γραμμές χρησιμοποιώντας το ίδιο εργαλείο. Θα έχουμε σχηματίσει έτσι ένα μικρότερο ορθογώνιο παραλληλόγραμμο στο πάνω μέρος του μπράτσου. Δείτε τις επόμενες εικόνες.



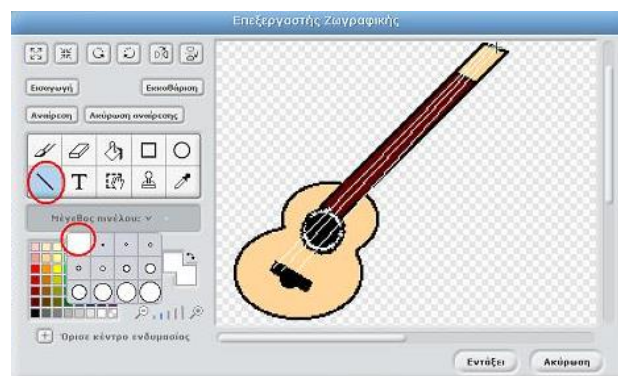
Τι θα λέγατε να προσθέσουμε χρώμα στο αντικείμενό μας; Με το εργαλείο επιλογής χρωμάτων διαλέγουμε το ανοιχτό πορτοκαλί και με το Εργαλείο Γεμίσματος χρωματίζουμε το σώμα της κιθάρας και το επάνω μέρος του μπράτσου.



Με τον ίδιο τρόπο γεμίζουμε με καφέ σκούρο χρώμα το υπόλοιπο μπράτσο της κιθάρας.



Πάμε να σχεδιάσουμε τις χορδές. Αφού επιλέξουμε το λευκό χρώμα από την παλέτα χρωμάτων τραβάμε με τη βοήθεια του εργαλείου Γραμμή παράλληλες ευθείες, από το κάτω μέρος του σώματος της κιθάρας μέχρι το πάνω μέρος του μπράτσου της.



Τέλος δίνουμε όνομα στο αντικείμενό μας.



Με αυτόν τον τρόπο ολοκληρώσαμε τη δημιουργία του αντικείμενου μας με τη χρήση του Επεξεργαστή Ζωγραφικής.

Διάλεξε νέα μορφή από αρχείο: Εκτός από το να σχεδιάσετε από μόνοι σας μια μορφή, μπορείτε να εισάγετε έτοιμα αντικείμενα από τη βιβλιοθήκη του Scratch. Ακόμη, θα μπορούσατε να χρησιμοποιήσετε εικόνες από το προσωπικό σας αρχείο ή το διαδίκτυο. Φανταστείτε την έκπληξη που θα προκαλέσετε, εάν φτιάξετε ένα πρόγραμμα στο Scratch χρησιμοποιώντας ως αντικείμενα τους φίλους σας με φωτογραφίες από το Facebook! Έστω ότι θέλουμε να δημιουργήσουμε ένα παιχνίδι ερωταποκρίσεων. Ο πρωταγωνιστής μας θα κάνει μια ερώτηση και αν η απάντηση του χρήστη είναι σωστή θα παίρνει μια χαρούμενη όψη, ενώ εάν είναι λανθασμένη θα παίρνει μια θυμωμένη όψη, ενώ μετά από συγκεκριμένο αριθμό λαθών το παιχνίδι θα τερματίζεται. Συνεπώς, για το παιχνίδι μας χρειαζόμαστε ένα αντικείμενο το οποίο θα κάνει κάποιες ερωτήσεις και ανάλογα με το αν η απάντηση είναι σωστή ή λάθος θα αλλάζει όψη κάνοντας αντίστοιχες γκριμάτσες.

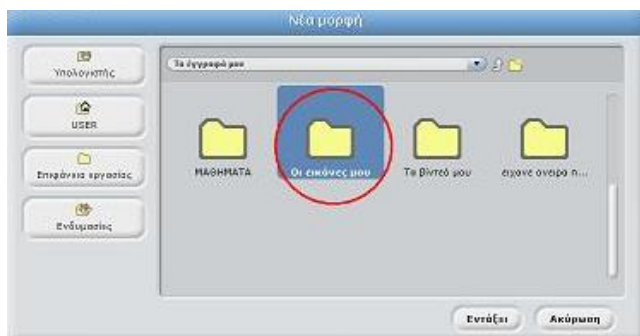
Χρησιμοποιήσαμε μια φιγούρα από τα γνωστά μας καρτούν, αυτή του Ντάφι Ντάκ. Βρήκαμε εικόνες του γνωστού μας ήρωα

κινουμένων σχεδίων στο διαδίκτυο και στη συνέχεια τις αποθηκεύσαμε στον υπολογιστή μας. Μπορείτε να κάνετε και εσείς το ίδιο, χρησιμοποιώντας την αγαπημένη σας μηχανή αναζήτησης (π.χ. <http://images.google.com>).

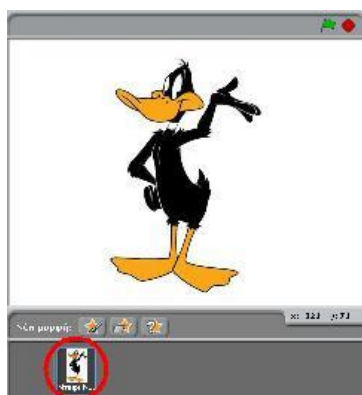
Για να εισάγουμε την εικόνα στο έργο μας, επιλεγούμε Διάλεξε νέα μορφή από αρχείο, όπως φαίνεται στην επόμενη εικόνα.



Αυτός είναι ο δεύτερος τρόπος με τον οποίο μπορούμε να δημιουργήσουμε αντικείμενα στο Scratch. Όπως θα δείτε στην οθόνη σας, ανοίγει ένα παράθυρο με τα διαθέσιμα από τη βιβλιοθήκη του Scratch αντικείμενα. Εμείς κατευθυνόμαστε στο φάκελο που έχουμε αποθηκεύσει την εικόνα μας και συγκεκριμένα στο φάκελο «Οι εικόνες μου».



Αφού ονομάσουμε το αντικείμενο μας, έχουμε ολοκληρώσει τη δημιουργία του.



Πως όμως ένα αντικείμενο μπορεί να έχει διαφορετικές όψεις;

4.3.3 Ενδυμασίες

Κάθε αντικείμενο στο Scratch μπορεί να έχει μια ή και περισσότερες ενδυμασίες. Οι ενδυμασίες ενός αντικείμενου, είναι τα διαφορετικά κοστούμια που θα θέλαμε να φοράει ο χαρακτήρας μας σε διαφορετικές στιγμές στο έργο μας. Εμείς θα προσθέσουμε τρεις ενδυμασίες στο αντικείμενό μας. Οι δυο θα αφορούν τις γκριμάτσες που θα κάνει αναλόγως με την απάντηση του χρήστη και η τελευταία θα αφορά τον τερματισμό του παιχνιδιού.

Αρχικά πατήστε πάνω στην καρτέλα Ενδυμασίες και παρατηρήστε πως στο μεσαίο τμήμα της οθόνης υπάρχει μια ενδυμασία για το αντικείμενο μας. Η πρώτη ενδυμασία είναι η εικόνα που εισάγαμε προηγουμένως.



Μετονομάζουμε την πρώτη ενδυμασία του αντικείμενου μας σε Ντάφι Ντάκ_ερωτήσεις, μιας και επιλέξαμε αυτήν την ενδυμασία να εμφανίζεται ως αρχική.

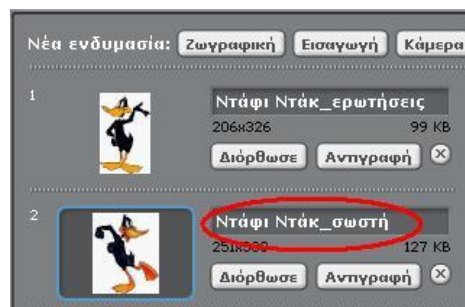


Όπως μπορείτε να διαπιστώσετε, πάνω ακριβώς από την ενδυμασία του αντικείμενου, το Scratch μας παρέχει τρεις διαφορετικούς τρόπους για να εισάγουμε νέες ενδυμασίες. Επιλέγοντας το εικονίδιο Ζωγραφική μπορείτε να σχεδιάσετε μια νέα ενδυμασία για το αντικείμενό σας, χρησιμοποιώντας τον Επεξεργαστή Ζωγραφικής. Πατώντας το εικονίδιο Εισαγωγή, μπορείτε να εισάγετε μια ενδυμασία από τη βιβλιοθήκη του Scratch ή επιλέγοντας κάποιο αρχείο σας όπως κάναμε σε αυτή την περίπτωση. Πατώντας το εικονίδιο Κάμερα, μπορείτε να τραβήξετε μια φωτογραφία και στη συνέχεια να την εισάγετε ως ενδυμασία. Δηλαδή, όλες οι επιλογές που είναι διαθέσιμες για τα υπόβαθρα είναι διαθέσιμες και για τα αντικείμενα.

Παραπάνω αναφέραμε ότι θα προσθέσουμε τρεις νέες ενδυμασίες. Αρχικά θα προσθέσουμε την ενδυμασία με την χαρούμενη όψη του αντικείμενου μας για την περίπτωση της σωστής απάντησης. Για να το κάνουμε αυτό επιλεγούμε Εισαγωγή και επιλέγουμε τη δεύτερη εικόνα του Ντάφυ που βρήκαμε για τη σωστή απάντηση.



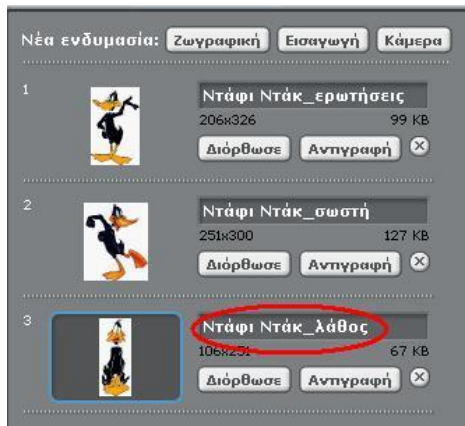
Μετονομάζουμε την νέα ενδυμασία του αντικείμενου μας και έχουμε ολοκληρώσει τη δημιουργία της.



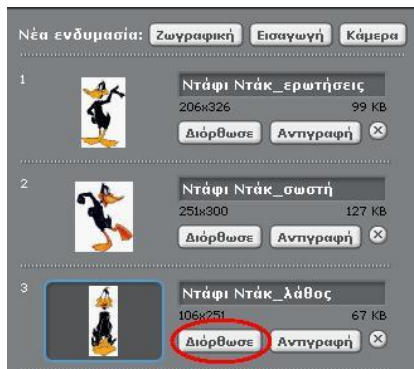
Σε αυτό το σημείο πρέπει να παρατηρήσετε πως η ενδυμασία με την οποία το αντικείμενο μας εμφανίζεται στη σκηνή του Scratch, είναι αυτή που είναι επιλεγμένη με μπλε περίγραμμα

στη λίστα των ενδυμασιών. Πατώντας σε διαφορετικές ενδυμασίες, αλλάζει αντίστοιχα η ενδυμασία του αντικειμένου και στη σκηνή.

Με τον ίδιο τρόπο, πατώντας το εικονίδιο Εισαγωγή, δημιουργούμε την τρίτη ενδυμασία του αντικειμένου μας. Την χρειαζόμαστε για την περίπτωση της λανθασμένης απάντησης. Κάθε λάθος απάντηση φέρνει το χρήστη κάθε φορά ένα βήμα πιο κοντά στο τέλος. Σε περίπτωση λάθους λοιπόν, η ενδυμασία μας θα είναι η κρεμάλα! Και έτσι ο χαρακτήρας μας θα είναι πολύ θυμωμένος...στον παράδεισο! Αυτή τη φορά αφού μετονομάσουμε την νέα ενδυμασία του αντικειμένου μας θα την επεξεργαστούμε για το σκοπό του παιχνιδιού μας.



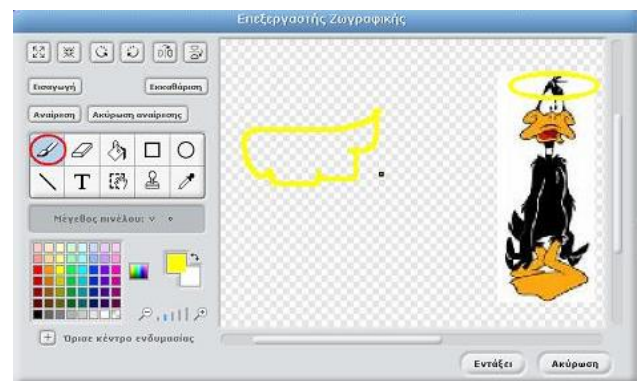
Επιλέγοντας το εικονίδιο Διόρθωσε μπορούμε να αλλάξουμε κάθε ενδυμασία των αντικειμένων μας χρησιμοποιώντας τον Επεξεργαστή Ζωγραφικής. Για να δώσουμε έμφαση στη λάθος απάντηση αλλά και μια δόση χιούμορ, θα σχεδιάσουμε ένα φωτοστέφανο συνοδευόμενο από τα χαρακτηριστικά φτερά των αγγέλων. Επιλέγουμε Διόρθωσε στην τελευταία ενδυμασία μας.



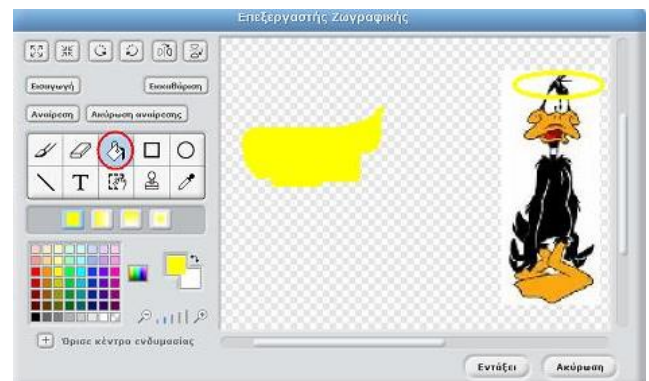
Θα ανοίξει το παράθυρο του Επεξεργαστή Ζωγραφικής και στον καμβά του θα εμφανιστεί η ενδυμασία που επιλέξαμε να επεξεργαστούμε. Για να σχεδιάσουμε το φωτοστέφανο δεν έχουμε παρά να κάνουμε δυο απλές κινήσεις. Αρχικά διαλέγουμε το κίτρινο χρώμα από την παλέτα χρωμάτων και στη συνέχεια επιλεγούμε το εργαλείο έλλειψης (χωρίς γέμισμα) για να σχεδιάσουμε μια έλλειψη γύρω από το κεφάλι του Ντάφυ.



Χρειαζόμαστε τα αναγκαία φτερά. Χρησιμοποιώντας τη βούρτσα ζωγραφικής, προσπαθήστε να σχεδιάσετε ένα περίγραμμα όπως φαίνεται στην επόμενη εικόνα. Εδώ σας προκαλούμε να δείξετε την καλλιτεχνική σας φύση. Μη φοβάστε να κάνετε λάθος. Υπάρχει πάντα το εικονίδιο της Ανίρεσης και έτσι μπορείτε να προσπαθήστε ξανά και ξανά, μέχρι να πετύχετε το επιθυμητό αποτέλεσμα. Αν παρ' όλα αυτά δεν τα καταφέρετε, μπορείτε να αναζητήσετε έτοιμες εικόνες στο διαδίκτυο..



Στη συνέχεια, με τη βοήθεια του εργαλείου γεμίσματος γεμίζουμε με κίτρινο το περίγραμμα του φτερού.



Το επόμενο βήμα είναι να δημιουργήσουμε το δεύτερο φτερό. Για να το πετύχουμε αυτό θα μας βοηθήσει το εργαλείο σφραγίδα. Πατάμε πάνω στο εργαλείο σφραγίδα και επιλέγουμε την περιοχή που θέλουμε να αντιγράψουμε, δηλαδή το φτερό. Στη συνέχεια με ένα απλό κλικ στο κατάλληλο σημείο, το αρχικό φτερό αντιγράφεται. Αφήνουμε το δεύτερο φτερό στη δεξιά πλευρά του Ντάφυ.



Για να αναστρέψουμε το πρώτο φτερό ώστε να έχει τη σωστή θέση και κατεύθυνση, θα χρησιμοποιήσουμε δυο εντολές. Αρχικά, επιλέγουμε το φτερό με τη χρήση του εργαλείου επιλογής.



Στη συνέχεια πατώντας το εικονίδιο οριζόντιας αναστροφής το φτερό αντιστρέφεται οριζόντια και έχει αποκτήσει την κατάλληλη κατεύθυνση. Μπορούμε, εάν θέλουμε, να μετακινήσουμε πιο πάνω ή κάτω το φτερό, χρησιμοποιώντας το εργαλείο επιλογής.



Ας κάνουμε λίγο πιο κωμικοτραγική την εικόνα με μια θηλιά! Αρχικά, διαλέγουμε το σκούρο καφέ χρώμα με το εργαλείο επιλογής χρωμάτων, για το χρώμα του σχοινιού και στη συνέχεια τραβάμε μια κάθετη ευθεία γραμμή, από το κεφάλι του Ντάφυ προς τα επάνω. Τέλος, χρησιμοποιώντας τη βούρτσα σχεδιάζουμε ένα ατελή κύκλο γύρω από το λαιμό του αντικειμένου μας.



Πάμε να δώσουμε την τελική πινελιά στο παιχνίδι μας και να εξηγήσουμε παράλληλα, με ποιο τρόπο μπορούμε να επεξεργαστούμε μια ενδυμασία που προήλθε από κάποια έτοιμη εικόνα ή φωτογραφία. Θα προσθέσουμε στο αντικείμενο μας μια τελευταία ενδυμασία. Η ενδυμασία αυτή θα είναι το τέλος παιχνιδιού, κάτι σαν το game over που έχετε δει στα παιχνίδια που παίζετε.

Αποφασίσαμε να χρησιμοποιήσουμε το γνωστό τίτλο τέλους των καρτούν Looney Tunes, έτσι ώστε το παιχνίδι μας να έχει το τέλος που του αρμόζει. Μετά από μια μικρή αναζήτηση στο διαδίκτυο καταλήξαμε στην επόμενη εικόνα.



Για να δημιουργήσουμε τη νέα ενδυμασία πατάμε Εισαγωγή και κατευθυνόμαστε στο φάκελο που έχουμε αποθηκεύσει την εικόνα μας.



Σκεφτήκαμε ότι θα ήταν πιο ενδιαφέρον να εμφανίζεται στο τέλος του παιχνιδιού μας, ο δημιουργός του έργου. Μπορείτε να φανταστείτε τα σχόλια των φίλων σας εάν στην παραπάνω εικόνα εμφανιζόσασταν εσείς αντί του γνωστού καρτούν...

Αρχικά επιλεγούμε το εικονίδιο Διόρθωση, όπως είδατε και παραπάνω, έτσι ώστε να ανοίξει ο Επεξεργαστής Ζωγραφικής. Στη συνέχεια πατάμε το εικονίδιο Εισαγωγή.



Θα ανοίξει το παράθυρο εισαγωγής εικόνας, με τη χρήση του οποίου θα επιλέξετε τη φωτογραφία σας.



Θυμηθείτε ότι αν το μέγεθός της είναι είτε μεγάλο είτε μικρό, ο επεξεργαστής ζωγραφικής παρέχει αντίστοιχα τα εργαλεία σμί-
κρυνσης και μεγέθυνσης για την φέρετε σε κατάλληλο μέγεθος.



Με τη βοήθεια της σβηστήρας μπορείτε να σβήσετε τα τμήματα της φωτογραφίας που δεν χρειάζεστε.

Το μόνο που έχουμε να κάνουμε είναι να τοποθετήσουμε τη φωτογραφία στο σωστό σημείο. Μπορείτε επίσης να περιστρέ-
ψετε την φωτογραφία σας αριστερόστροφα ή δεξιόστροφα ώστε να καταφέρετε το επιθυμητό αποτέλεσμα.



Μπορούμε να προσθέσουμε μια τελευταία λεπτομέρεια χρησιμοποιώντας το εργαλείο κείμενου. Έτσι επιλέξαμε το κίτρινο χρώμα, για να διακρίνεται πάνω στο σκούρο φόντο και γράψαμε «Τέλος παιχνιδιού».



Μετά από όλα αυτά τα παραδείγματα, γνωρίζουμε πλέον να προσθέτουμε ενδυμασίες σε ένα αντικείμενο ποικιλοτρόπως! Μήπως ήρθε η ώρα να προσθέσουμε μελωδίες στην κιθάρα μας;

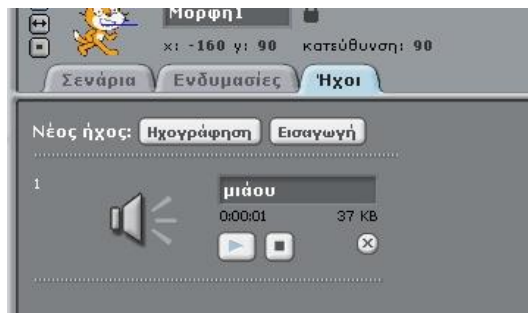
4.4 Ήχοι

4.4.1 Χρήση ήχων

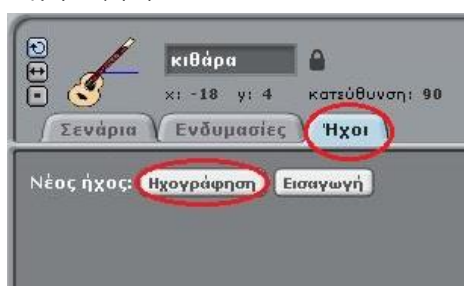
Ήρθε η ώρα να δώσουμε ζωή στα αντικείμενα ηχογραφώντας ήχους ή αναπαράγοντας έτοιμους ήχους ή αγαπημένα μας τραγούδια. Κάθε αντικείμενο έχει τους δικούς του ήχους και μόνο αυτό μπορεί να τους αναπαράγει με τη χρήση κατάλληλων εντολών. Τι ήχους θα μπορούσε να χρειάζεται ένα αντικείμενο; Ήχους ειδοποιήσεων, μουσικά κομμάτια, ηχογραφημένες εκφράσεις, κραυγές νίκης, κτλ.

Ο πιο εύκολος τρόπος για να εισάγουμε ήχους στο Scratch είναι μέσω του μικροφώνου μας. Είτε θέλουμε να εισάγουμε συγκεκριμένες εκφωνήσεις, είτε θέλουμε να εισάγουμε φυσικούς ήχους όπως τον ήχο του αέρα ή της βροχής, μπορούμε να αξιοποιήσουμε το μικρόφωνο του υπολογιστή μας.

Αφού έχουμε προνοήσει να υπάρχει ένα συνδεδεμένο μικρόφωνο στον ηλεκτρονικό υπολογιστή μας, δημιουργούμε ένα νέο έργο και επιλέγουμε από το μεσαίο παράθυρο του Scratch την καρτέλα «Ήχοι». Παρατηρήστε ότι το αντικείμενό μας περιέχει ήδη έναν ήχο, «μιάου».



Αν πατήσουμε το «παίξε»  θα ακούσουμε τον συγκεκριμένο ήχο. Η γάτα του Scratch δηλαδή έχει και μιλιά! Πως μπορούμε να εισάγουμε ήχο στο παράδειγμα με την κιθάρα της προηγούμενης ενότητας; Ανοίγουμε το αντίστοιχο έργο και επιλέγουμε το αντικείμενο «κιθάρα». Για να ηχογραφήσουμε τους ήχους μας επιλέγουμε την καρτέλα ήχοι και στη συνέχεια το κουμπί «Ηχογράφηση»,



με αποτέλεσμα να εμφανιστεί η παρακάτω εικόνα:



Θα πρέπει τώρα να ηχογραφήσουμε τον ήχο μιας κανονικής κιθάρας. Πατώντας το κουμπί  ξεκινάμε την ηχογράφηση μας και όταν τελειώσουμε πατάμε το κουμπί . Για να ακούσουμε την ηχογράφηση που πραγματοποιήσαμε πατάμε το .

Η μπάρα που φαίνεται να μετακινείται, κατά τη διάρκεια της ηχογράφησης δείχνει την ένταση του ήχου που ηχογραφούμε. Καλό είναι να φτάνει κατά τη διάρκεια της ηχογράφησης ως τη μέση περίπου. Αυτό το επίπεδο έντασης αρκεί για μια ικανοποιητική ηχογράφηση. Αν μετακινείται πολύ λίγο από το αρχικό της σημείο, μάλλον η πηγή του ήχου είναι μακριά από το μικρόφωνο ενώ αν διαρκώς βρίσκεται στη στο δεξί άκρο, μάλλον πρέπει να απομακρύνουμε το μικρόφωνο από την πηγή. Ενδιαφέρον είναι ότι Scratch ξεκινά την καταγραφή τη στιγμή που έχει αντιληφθεί κάποιον ήχο και όχι όταν απλά πατάμε το κουμπί της εγγραφής!

Αφού ακούσουμε την ηχογράφηση που κάναμε, και εφόσον είμαστε ικανοποιημένοι, πατάμε το «εντάξει» και η ηχογράφησή μας εμφανίζεται στη λίστα των ηχογραφήσεων.

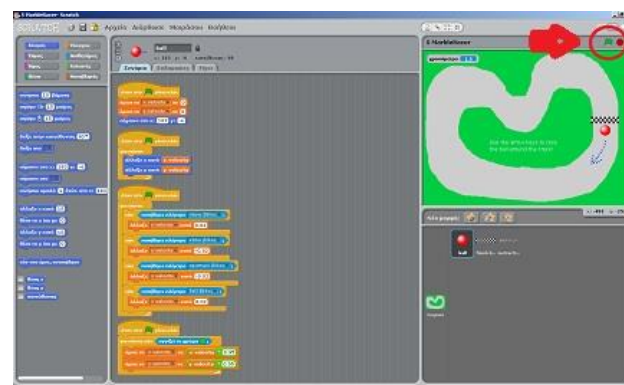
Τα περισσότερα προγράμματα, χρησιμοποιούν ήχους για να δηλώσουν διάφορα συμβάντα και γεγονότα. Πολύ συχνός ήχος

είναι ο ήχος «προειδοποίησης» όταν βγαίνουμε από ένα πρόγραμμα και δεν έχουμε κάνει αποθήκευση του αρχείου μας, ή όταν το λειτουργικό σύστημα εμφανίζει κάποιο πρόβλημα. Ορισμένα παιχνίδια αναπαράγουν διαρκώς μουσική π.χ. το «Bubble Bubble». Είναι πολλές οι περιπτώσεις λοιπόν που θέλουμε να χρησιμοποιήσουμε έτοιμους ήχους. Να βάλουμε ήχο σε ένα παιχνίδι;

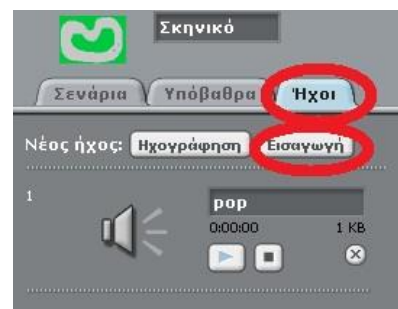
Θα χρησιμοποιήσουμε ένα έτοιμο παραδειγματικό έργο του Scratch. Επιλέγουμε «Αρχείο» μετά «Ανοίγμα..» και μετά «Παραδείγματα». Πατάμε διπλό κλικ στον κατάλογο «Games» και επιλέγουμε το παιχνίδι «5 MarbleRacer» και πατάμε «Εντάξει».

Η οθόνη του Scratch πρέπει να είναι παρόμοια με αυτήν της εικόνας που ακολουθεί. Αν πατήσουμε στη σημαία για να ξεκινήσει η εκτέλεση του έργου, θα παρατηρήσουμε ότι μπορούμε να οδηγήσουμε την κόκκινη μπάλα με τα βελάκια του πληκτρολογίου χωρίς όμως να ακούγεται κάποια μουσική. Ας δούμε πως μπορούμε να την προσθέσουμε εμείς!

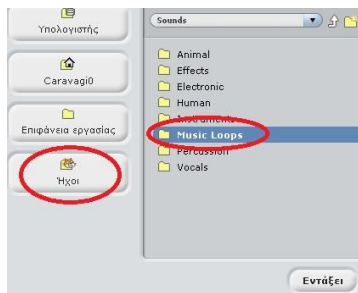
Αφού θέλουμε ο ήχος να ακούγεται σε όλη τη διάρκεια εκτέλεσης του παιχνιδιού θα τον εισάγουμε στο σκηνικό του έργου.



Επιλέγουμε το σκηνικό και στη συνέχεια την καρτέλα «Ήχοι» και πατάμε στο κουμπί εισαγωγή.



Από το παράθυρο που εμφανίζεται, μπορούμε είτε να εισάγουμε ένα ήχο που έχουμε στον υπολογιστή μας, είτε πατώντας στο αριστερό μέρος «Ήχοι» να αξιοποιήσουμε τη βιβλιοθήκη ήχων που μας παρέχει το Scratch. Εμείς θα κάνουμε το δεύτερο. Μπείτε στο φάκελο «Music Loops» και επιλέξτε τον ήχο «Triumph». Τέλος, πατήστε «εντάξει».



Ο ήχος που προσθέσαμε, εμφανίζεται στη λίστα ήχων του σκη- νικού.



Με αυτόν τον τρόπο εισάγουμε ήχους στο σκηνικό. Ας πάμε ό- μως λίγο παραπέρα. Ακολουθήστε την επόμενη διαδικασία πα- ρότι μπορεί να μην κατανοήσετε το νόημα ορισμένων βημάτων της.

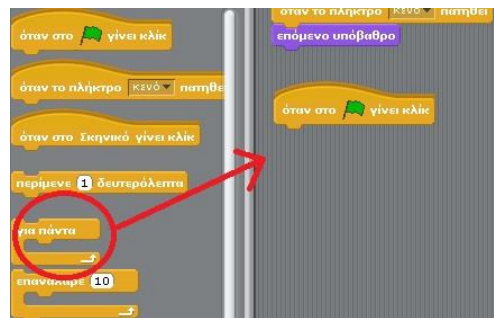
Επιλέξτε την καρτέλα «Σενάρια». Εκεί θα δείτε ότι υπάρχει ήδη ένα σύνολο εντολών:



Επιλέξτε την παλέτα εντολών «Έλεγχος» και σύρετε με το πο- ντίκι την εντολή «Όταν στη πράσινη σημαία γίνει κλικ» μέσα στο χώρο των σεναρίων του σκηνικού.



Ομοίως, μεταφέρετε την εντολή «για πάντα» μέσα στα σενάρια και ενώστε την με την προηγούμενη.



Από την παλέτα εντολών «Ήχοι» μεταφέρετε την εντολή παιξε ήχο «Triumph» μέχρι τέλους μέσα στην εντολή για πάντα όπως δείχνει η επόμενη εικόνα.



Θα πρέπει τώρα μέσα στα σενάρια του σκηνικού να βλέπετε δυο ομάδες εντολών όπως παρουσιάζονται στην επόμενη εικόνα:



Δεν είναι δύσκολο να καταλάβουμε τη λειτουργία των εντολών που προσθέσαμε: Όταν ο χρήστης πατήσει στην πράσινη ση- μαία, θέλουμε για πάντα να παίζεται ολόκληρος ο ήχος Tri- umph. Τρέξτε το έργο για να δείτε τα αποτελέσματά του.

Περίληψη

Στο κεφάλαιο αυτό είδαμε πως μπορούμε να δημιουργήσουμε αντικείμενα, σκηνικά, ενδυμασίες και ήχους. Πρέπει να θυμόμα- στε ότι το Scratch μας δίνει τέσσερις κατευθύνσεις εργασίας:

α) μπορούμε να δημιουργούμε-σχεδιάζουμε μόνοι μας αντικεί- μενα, σκηνικά, ενδυμασίες, ήχους κτλ.

β) μπορούμε να χρησιμοποιούμε τη βιβλιοθήκη του Scratch για να εισάγουμε αντικείμενα, σκηνικά, ενδυμασίες, ήχους κτλ

γ) μπορούμε να εισάγουμε εικόνες – αρχεία από το προσωπικό μας υπολογιστή ή το διαδίκτυο ως αντικείμενα, σκηνικά, ενδυμασίες, ήχους κτλ.

δ) μπορούμε να εισάγουμε αρχεία είτε από τη βιβλιοθήκη του Scratch είτε από τον υπολογιστή μας αλλά στη συνέχεια να τα επεξεργαζόμαστε στο Scratch

Συνεπώς, οι τεχνικές δυνατότητες που μας δίνει το Scratch για να δημιουργήσουμε τους πρωταγωνιστές των παιχνιδιών μας είναι πολλές. Το μόνο που έχετε να κάνετε είναι να χρησιμοποιήσετε τη φαντασία σας.

Ερωτήσεις

- 1) Περιγράψτε τη χρησιμότητα των σκηνικών και των αντικειμένων.
- 2) Για ποιο λόγο μπορούμε να προσδιορίσουμε διαφορετικές ενδυμασίες για τα αντικείμενά μας;
- 3) Ποια είναι η διαφορά ανάμεσα στο σκηνικό και στα αντικείμενα.
- 4) Πως εισάγουμε ήχους σε ένα αντικείμενο;

Δραστηριότητες

- 1) Κάντε τον εαυτό σας ήρωα μιας ιστορίας δημιουργώντας ένα αντικείμενο και τοποθετώντας για κύρια ενδυμασία τη δική σας φωτογραφία.
- 2) Φτιάξτε 5 διαφορετικές ενδυμασίες του αγαπημένου σας ήρωα κινουμένων σχεδίων και ηχογραφήστε ατάκες που συχνά χρησιμοποιεί.
- 3) Δημιουργήστε ένα αντικείμενο που θα χορεύει! Σχεδιάστε ένα αντικείμενο και μια ενδυμασία του. Το αντικείμενο και η ενδυμασία πρέπει να είναι τέτοια ώστε όταν εναλλάσσονται να φαίνεται ότι το αντικείμενο χορεύει. Παράδειγμα εικόνας.



Στη συνέχεια εισάγετε στους ήχους του αντικειμένου ένα μουσικό κομμάτι της αρεσκείας σας, ή ηχογραφήστε το από το μικρόφωνο. Δημιουργήστε το περιβάλλον στο οποίο θέλετε ο χαρακτήρας σας να χορεύει.

4) Προσπαθήστε χρησιμοποιώντας τις εντολές "για πάντα", "επόμενη ενδυμασία" καθώς και την "παίξε ήχο μέχρι τέλους" να κάνετε το χαρακτήρα σας να χορεύει!

5) Περιγράψτε εν συντομία, πως θα συνδυάζατε αντικείμενα, ενδυμασίες και ήχο για να δημιουργήσετε ένα παιχνίδι που σας αρέσει.

6) Αντιστοιχίστε τα παρακάτω εικονίδια με τις λειτουργίες που επιτελούν



Αναστροφή οριζόντια



Επέλεξε εργαλείο



Αναστροφή κάθετα



Εργαλείο επιλογής χρωμάτων



Δεξιόστροφη περιστροφή



Εργαλείο σφραγίδας



Σηστήρα



Αριστερόστροφη περιστροφή



Εργαλείο γραμμής



Εργαλείο κειμένου



Βούρτσα



Εργαλείο παραλληλογράμμου



Εργαλείο έλλειψης



Εργαλείο γεμίσματος



Κεφάλαιο 5: Κίνηση

Σε αυτό το κεφάλαιο:

- 5.1 Σχετική κίνηση αντικειμένων
- 5.2 Κίνηση με συντεταγμένες
- 5.3 Ομαλή κίνηση
- 5.4 Ολοκληρωμένο παράδειγμα

Δώσε μου τόπο να σταθώ και κινώ και την γη ακόμα...

(Αρχιμήδης)



5.1 Σχετική κίνηση αντικειμένων

Πως θα μπορούσε να κινηθεί ένας χαρακτήρας προς την έξοδο ενός λαβύρινθου; Πως θα μπορούσε το αυτοκινητάκι μας να κινείται μέσα στην πίστα; Πως θα μπορούσαμε να αναπαραστήσουμε τη γη καθώς κάνει μια περιστροφή γύρω από τον ήλιο με σταθερή ταχύτητα;

Στο κεφάλαιο αυτό, θα δημιουργήσουμε τα πρώτα ολοκληρωμένα προγράμματά μας, τα οποία για τη συνέχεια του βιβλίου θα τα αποκαλούμε «έργα». Τα σενάρια που θα εξετάσουμε, επικεντρώνονται στην κίνηση των αντικειμένων και οι εντολές που θα αναλυθούν θα μας επιτρέψουν να μετατρέψουμε το σκηνικό μας σε μια πραγματική θεατρική σκηνή!

Πριν εισάγουμε τις πρώτες μας εντολές πρέπει να θυμηθούμε ότι:

α) Οι εντολές αναφέρονται σε συγκεκριμένα αντικείμενα, **κάθε δηλαδή αντικείμενο περιέχει τα δικά του σενάρια**. Συνεπώς, πρώτα πρέπει να επιλέγουμε το κατάλληλο αντικείμενο και στη συνέχεια να σέρνουμε τις εντολές μέσα στο σενάριό του.

β) Η έναρξη ενός προγράμματος πραγματοποιείται όταν πατήσουμε την πράσινη σημαία. Πότε όμως τρέχουν οι εντολές του αντικείμενου; Αν σύρετε απλά μια εντολή μέσα στο σενάριο ενός αντικείμενου, τότε αυτή δε θα εκτελεστεί ποτέ. Αν θέλουμε να τρέχουν οι εντολές μας τη στιγμή που πατάμε την πράσινη σημαία, τότε από την παλέτα εντολών **Έλεγχος** πρέπει να σύρουμε την εντολή **όταν στο κουμπί πράσινη σημαία γίνει κλικ** στο χώρο του σεναρίου. Στη συνέχεια «κολλάμε» από κάτω την επιθυμητή ακολουθία εντολών και κάθε φορά που θα κάνουμε κλικ στην πράσινη σημαία, οι εντολές θα εκτελούνται.

Το Scratch μας παρέχει όχι έναν, αλλά τρεις διαφορετικούς τρόπους κίνησης των αντικειμένων μας!

Ο πρώτος τρόπος αφορά τη **«σχετική κίνηση των αντικειμένων»**, δηλαδή τη μετακίνησή τους χωρίς να γνωρίζουμε ούτε την ακριβή θέση τους, και χωρίς να προσδιορίζουμε με ακρίβεια την τελική τους θέση. Απλά λέμε στο αντικείμενο να πάει λίγο πιο δεξιά ή λίγο πιο αριστερά κτλ.

Ο δεύτερος τρόπος αφορά τη μετακίνηση των αντικειμένων **«βάσει συντεταγμένων»**. Σε αυτήν την περίπτωση προσδιορίζουμε με ακρίβεια το που θα βρίσκεται ο χαρακτήρας μας πάνω στην οθόνη του Scratch. Και οι δυο τρόποι που αναφέρθηκαν προκαλούν ακαριαία μετατόπιση του αντικείμενου, μεταφέρουν δηλαδή το αντικείμενο στη νέα του θέση χωρίς καθυστέρηση χρόνου.

Επιπλέον σε αυτούς τους τρόπους κίνησης, υπάρχει και η **«ομαλή μετακίνηση»**, κατά την οποία το αντικείμενο μεταφέρεται σε μια άλλη θέση στην οθόνη παρουσιάζοντας όμως την πορεία μετακίνησης. Δηλαδή δεν μεταφέρεται ακαριαία αλλά σιγά-σιγά.

Ας ξεκινήσουμε με τη σχετική κίνηση.

5.1.1 Τα πρώτα μου βήματα...

Σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε εντολές από την παλέτα εντολών **Κίνηση** (θυμίζουμε ότι οι παλέτες εντολών βρίσκονται στο αριστερό τμήμα της οθόνης του Scratch).

Η βασική και πιο απλή εντολή κίνησης είναι η **κινήσου...βήματα**. Η εντολή αυτή, μετατοπίζει το αντικείμενο κατά όσα βήματα ορίσετε εσείς (το ένα βήμα αντιστοιχεί σε λίγα χιλιοστά). Η προεπιλεγμένη τιμή είναι τα 10 βήματα, όμως μπορείτε να την αλλάξετε κάνοντας κλικ στο λευκό κουτάκι που περιέχει τον αριθμό 10 και προσδιορίζοντας ένα νέο αριθμό βημάτων. Αν δημιουργήσετε την εντολή **κινήσου 100 βήματα**, τότε το αντικείμενό σας θα κινηθεί προς τα δεξιά κατά 100 βήματα. Προσέξτε ότι η μετακίνηση είναι ακαριαία, δηλαδή τα βήματα αντιστοιχίζονται σε απόσταση και **δεν** χρειάζεται χρόνος ανάλογος με τον αριθμό των βημάτων.

Χρησιμοποιώντας το σενάριο που παρουσιάζεται παρακάτω, κινείτε το αντικείμενό σας κατά 500 βήματα. Δοκιμάστε να προσθέσετε πολλές φορές την εντολή **κινήσου...βήματα** (θα πρέπει να κολλήσετε την μια εντολή κάτω από την άλλη) και πατήστε ξανά την πράσινη σημαία (θυμάστε πως μπορούμε να

Κίνηση

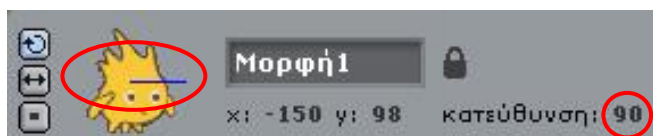
Έλεγχος

κινήσου 10 βήματα

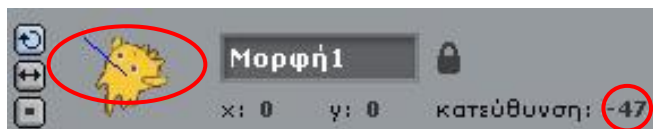
ξανα-εμφανίσουμε ένα αντικείμενο που καταλήγει εκτός σκηνικού;)



Γιατί όμως το αντικείμενό μας κινείται προς τα δεξιά; Είναι σημαντικό να κατανοήσουμε ότι κάθε αντικείμενο έχει συγκεκριμένη κατεύθυνση και η εντολή **κινήσου...βήματα** θα το μετακινήσει προς αυτήν την κατεύθυνση. Στο προηγούμενο παράδειγμα, η γάτα κοιτάει προς τα δεξιά και για αυτό και μετακινείται προς τα δεξιά. Για να διαπιστώσετε την κατεύθυνση ενός αντικειμένου δεν έχετε παρά να παρατηρήσετε δυο ιδιότητες του αντικειμένου στο πάνω μέρος του περιβάλλοντος εργασίας. Η μπλε γραμμή, που διακρίνεται στην επόμενη εικόνα, δείχνει την κατεύθυνση οπτικά, ενώ η κατεύθυνση προσδιορίζεται και αριθμητικά βάσει μοιρών (90 μοίρες για το φαντασματάκι μας).



Τι γίνεται όμως όταν θέλω να κινήσω το αντικείμενό μου προς τα αριστερά; Για να κινηθεί το αντικείμενο προς άλλη κατεύθυνση, πρέπει πρώτα να «στραφεί» προς εκείνη την κατεύθυνση και στη συνέχεια να κινηθεί. Ο πιο εύκολος τρόπος για να στρίψετε ένα αντικείμενο είναι να σύρετε τη δεξιά άκρη της μπλε γραμμής. Έτσι, το αντικείμενό σας θα μπορούσε να κοιτάει προς πάνω και αριστερά, όπως φαίνεται στην επόμενη εικόνα:



Παρατηρήστε ότι άλλαξαν και οι μοίρες της κατεύθυνσης (-47).

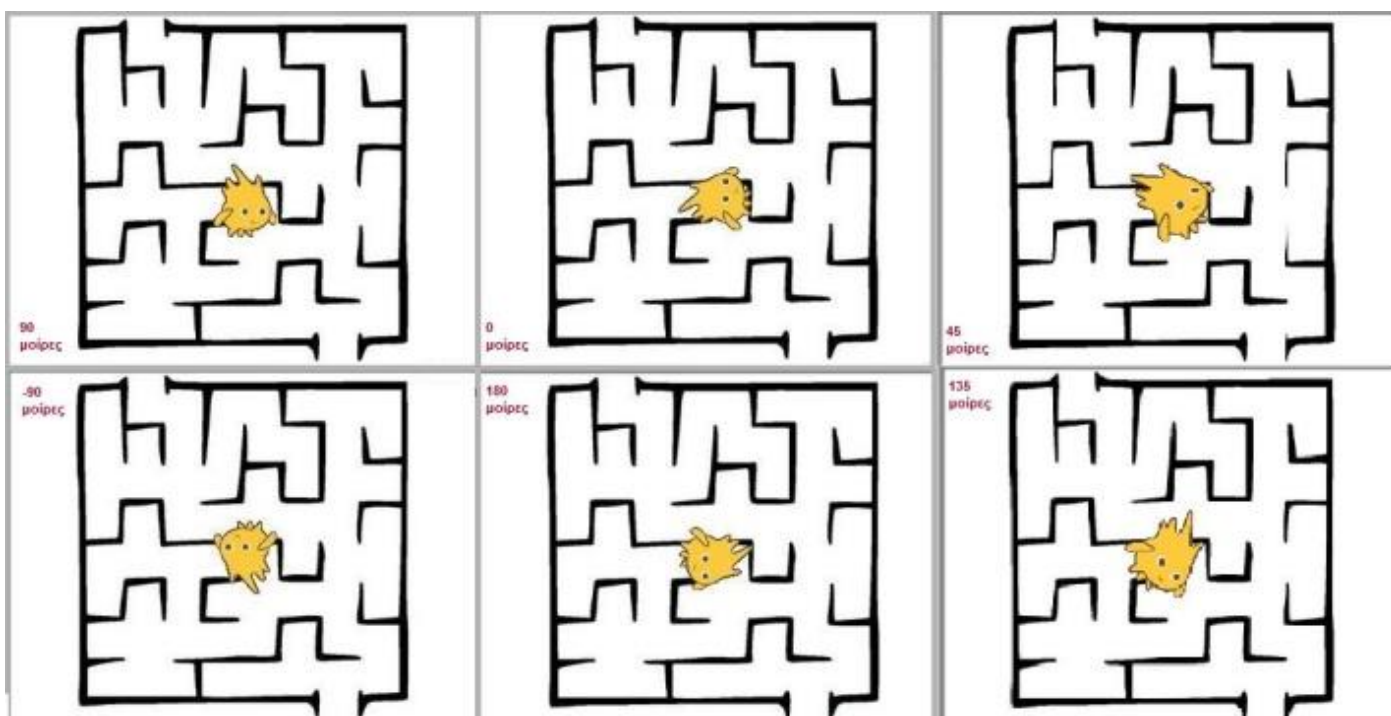
5.1.2 Απόλυτη κατεύθυνση

Πιθανόν όλοι σας να γνωρίζετε το παιχνίδι λαβύρινθος. Ο στόχος του παίκτη είναι να μετακινήσει έναν χαρακτήρα από τη μία άκρη στην άλλη, περπατώντας σε μια συγκεκριμένη διαδρομή και αλλάζοντας κατευθύνσεις. Ας δούμε πως είναι δυνατόν να κάνετε κάτι τέτοιο μέσω του Scratch. Οι 4 βασικές κατευθύνσεις προς τις οποίες μπορεί να στρέφεται το αντικείμενό σας είναι δεξιά (90 μοίρες), αριστερά (-90 μοίρες), πάνω (0 μοίρες) και κάτω (180 μοίρες). Όταν λέμε ότι το αντικείμενο στρέφεται προς μία κατεύθυνση, εννοούμε ότι κοιτάει προς αυτή την κατεύθυνση και κατά συνέπεια η εντολή **κινήσου...βήματα** τον μετακινεί προς αυτή την κατεύθυνση. Χρησιμοποιώντας την εντολή **δείξε στην κατεύθυνση...**, μπορείτε να επιλέξετε προς ποια από τις βασικές 4 κατευθύνσεις θέλετε να στραφεί το αντικείμενό σας. Επιπλέον, στη εντολή αυτή μπορείτε να εισάγετε τον ακριβή αριθμό μοιρών της επιθυμητής κατεύθυνσης κάνοντας κλικ στο λευκό κουτάκι. Το αντικείμενό σας στρέφεται προς όποια άλλη κατεύθυνση θέλετε αναλόγως τις μοίρες που έχετε ορίσει. Σε συνδυασμό με την εντολή **κινήσου...βήματα** το αντικείμενο μπορεί πλέον να κινηθεί προς οποιαδήποτε κατεύθυνση.

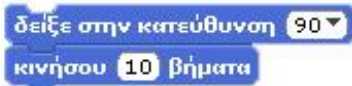
Δείξε στην κατεύθυνση 90



Χρησιμοποιώντας τις εντολές **δείξε στην κατεύθυνση...** και **κινήσου...βήματα** και αφού τοποθετήσετε το αντικείμενό σας σε μια από τις δυο εισόδους του λαβυρίνθου που εμφανίζεται στην επόμενη εικόνα, μετακινείτε το αντικείμενό σας μέσα στο λαβύρινθο έτσι ώστε να βρεθεί στην άλλη είσοδο/έξοδο. Δεν σας ζητείται προς το παρόν να δημιουργηθεί ένα ολοκληρωμένο πρόγραμμα αλλά να προσδιορίζετε τις κατάλληλες εντολές έτσι



ώστε κάνοντας διπλό κλικ διαδοχικά στο συνδυασμό των παρακάτω εντολών, το αντικείμενο να καταφέρει να φτάσει στο στόχο του (αν δε βρείτε κάποιον λαβύρινθο από το διαδίκτυο, σχεδιάστε έναν στον Επεξεργαστή Ζωγραφικής).

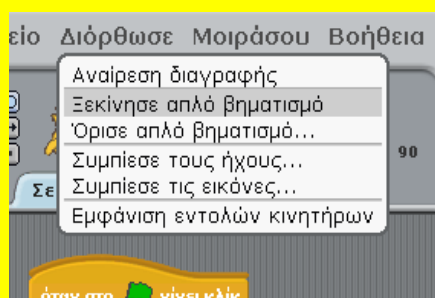


Έχοντας πειραματιστεί αρκετά, και κατανοώντας τη λειτουργία των δυο προηγούμενων εντολών, θέλουμε να δημιουργήσουμε πλέον το ολοκληρωμένο σενάριο. Αν τοποθετούσαμε το φαντασματάκι μας στην κάτω είσοδο, τότε για να καταλήξει στην πάνω είσοδο θα πρέπει να κάνει τα εξής βήματα [05_π01.sb]:



Προσπαθήστε να δημιουργήσετε το ολοκληρωμένο σενάριο για την αντίστροφη κίνηση.

Από τη στιγμή που οι εντολές κίνησης που παρουσιάστηκαν εκτελούνται ακαριαία, πως είναι δυνατόν να παρακολουθήσουμε την πορεία εκτέλεσης ενός έργου που περιλαμβάνει μόνο τέτοιες εντολές; Και πως θα το διορθώσουμε; Θα πρέπει να δοκιμάζουμε ξεχωριστά κάθε εντολή; Κάθε τμήμα του σεναρίου; Όχι. Το Scratch μας δίνει τη δυνατότητα να μελετήσουμε την εκτέλεση του σεναρίου μας βηματικά! Αν πάτε με το ποντίκι σας στο μενού «Διόρθωση», θα ανακαλύψετε δυο πολύ χρήσιμες επιλογές: α) Ξεκίνησε τον απλό βηματισμό β) Όρισε απλό βηματισμό.

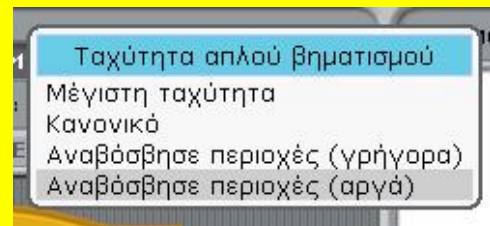


Η πρώτη επιλογή αναγκάζει το Scratch να εκτελεί τα έργα μας βήμα προς βήμα, δείχνοντάς μας ταυτόχρονα ποια εντολή επεξεργάζεται και ποιο είναι το αποτέλεσμα της. Η δεύτερη επιλογή μας επιτρέπει να προσδιορίσουμε την ταχύτητα του βηματισμού.

Στην παρακάτω εικόνα εμφανίζεται με λαδί χρώμα η εντολή που εκτελείται αλλά και η θέση του αντικείμενου μας μέσα στο λαβύρινθο μετά την εκτέλεση των εντολών μέχρι εκείνο το σημείο του σεναρίου



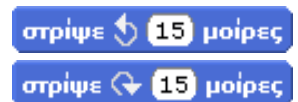
Για να ξεκινήσετε το βηματισμό, επιλέξτε καταρχάς «Ξεκίνησε απλό βηματισμό» και στη συνέχεια κάντε κλικ στην επιλογή «Όρισε απλό βηματισμό». Από το νέο μενού που εμφανίζεται επιλέξτε «Αναβόσβησε περιοχές αργά».



Κάθε φορά πλέον που θα πατάμε την πράσινη σημαία, το έργο μας θα εκτελείται βηματικά. Για να σταματήσουμε αυτή τη λειτουργία, πρέπει να επιλέξουμε πάλι από το μενού «Διόρθωση», την επιλογή «Σταμάτησε βηματισμό».

5.1.3 Σχετική κατεύθυνση

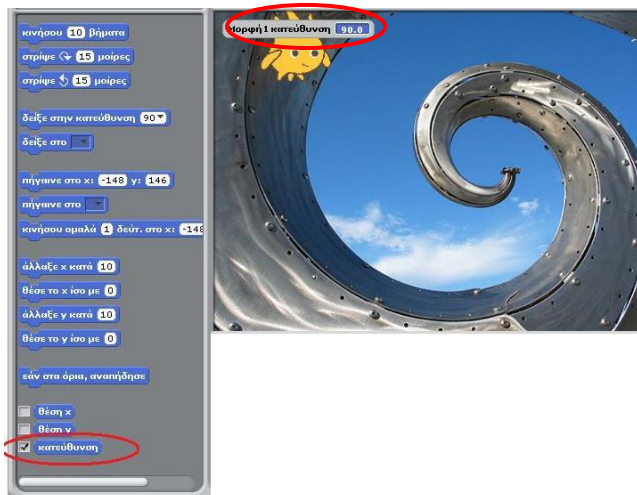
Τώρα που ξέρετε πως να κινηθείτε προς τις 4 κατευθύνσεις για να βγείτε από το λαβύρινθο, τι θα λέγατε να δυσκολέψουμε λίγο το πρόβλημα; Στην επόμενη εικόνα βλέπετε το φαντασματάκι μας στην αρχή ενός σπирάλ λαβυρίνθου. Πως θα το προγραμματίσετε έτσι ώστε να φτάσει στην άκρη του αγγίζοντας μόνο τη μια πλευρά του σχήματος πάνω στο οποίο βρίσκεται;



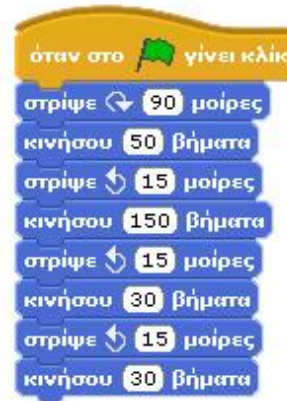


Η δυσκολία που συναντάμε σε αυτή την περίπτωση είναι ότι ο λαβύρινθος αποτελείται μόνο από καμπύλες. Για να κάνετε το αντικείμενό σας να κινηθεί σε αυτό το σπирάλ, θα πρέπει να στρίβει και να κινείται διαρκώς υπό διάφορες γωνίες. Για αυτό και το Scratch μας δίνει τις εντολές **στρίψε δεξιόστροφα...μοίρες** ή **στρίψε αριστερόστροφα...μοίρες**. Οι εντολές αυτές αλλάζουν την κατεύθυνση του αντικειμένου δεξιόστροφα ή αριστερόστροφα -κατά όσες μοίρες ορίσετε- σε σχέση με την κατεύθυνση την οποία έχει. Προσδιορίζουμε δηλαδή σχετική κατεύθυνση. Γιατί όμως να μη χρησιμοποιήσουμε την εντολή δείξε στην κατεύθυνση...; Γιατί είναι πιο εύκολο να λύσουμε το πρόβλημά μας λέγοντας στο αντικείμενό μας να στρίψει λίγο ακόμη προς τη μια ή την άλλη κατεύθυνση παρά να ορίζουμε απόλυτο αριθμό μοιρών της επιθυμητής κατεύθυνσης.

Για να διευκολυνθείτε στη δημιουργία του έργου σας μπορείτε, να τσεκάρετε την επιλογή **κατεύθυνση** από την παλέτα εντολών της κίνησης ώστε να εμφανίζεται η κατεύθυνση του αντικειμένου διαρκώς πάνω στη σκηνή.



Κάντε το αντικείμενό σας να διασχίσει το λαβύρινθο και να κινηθεί κατά μήκος της μεταλλικής πλευράς πάνω στην οποία είναι τοποθετημένο. Το σενάριό σας θα πρέπει να ξεκινάει κάπως έτσι:



Μπορείτε να συνεχίσετε μόνοι σας την υπόλοιπη διαδρομή. Μη ξεχνάτε ότι η βηματική εκτέλεση του έργου σας μπορεί να σας βοηθήσει σημαντικά

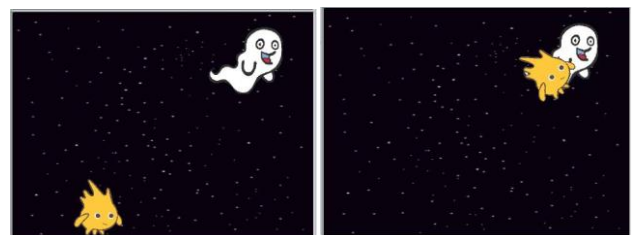
[05_π02.sb].

5.1.4 Σχετική κατεύθυνση προς αντικείμενο

Υπάρχουν πολλά παιχνίδια στα οποία οι "κακοί" χαρακτήρες του παιχνιδιού κυνηγούν τους "καλούς". Πως θα μπορούσατε κι εσείς στα δικά σας προγράμματα να κάνετε το κακό να κινείται προς την κατεύθυνση του καλού;

Πρέπει, αρχικά, να δημιουργήσετε ένα δεύτερο αντικείμενο και να το τοποθετήσετε στο σκηνικό. Πως όμως θα ξέρουμε ποια είναι ακριβώς η θέση του δεύτερου αντικειμένου σε σχέση με τη θέση του πρώτου; Επειδή αυτό δεν μπορείτε να το γνωρίζετε με σιγουριά και η θέση ενδέχεται διαρκώς να αλλάζει, το Scratch μας προσφέρει την εντολή **δείξε στο...** η οποία επιτρέπει σε ένα αντικείμενο να πάρει κατεύθυνση κίνησης προς οποιοδήποτε άλλο αντικείμενο του έργου μας. Επιλέγοντας το όνομα του νέου αντικειμένου μας μέσα στην εντολή **δείξε στο...**, το πρώτο αντικείμενο στρέφεται προς αυτό. Παρατηρήστε ότι αν εισάγετε επιπλέον αντικείμενα στο έργο σας, θα εμφανιστούν όλα στη λίστα αντικειμένων της εντολής. Αν στη συνέχεια χρησιμοποιήσετε την εντολή **κινήσου...βήματα**, τότε το πρώτο κινείται προς το δεύτερο. Έτσι, οπουδήποτε και να είναι το δεύτερο αντικείμενο, το πρώτο θα μπορεί πάντα να στραφεί προς αυτό μόνο με την εντολή **δείξε στο...**, ανεξάρτητα με τη θέση του δεύτερου.

Έχοντας σαν εκκίνηση το σκηνικό της επόμενης εικόνας, πιάστε το δεύτερο άσπρο φαντασματάκι με τις εντολές **δείξε στο...<μορφή2>** και **κινήσου...βήματα**.



Το σενάριό σας θα πρέπει να μοιάζει με το παρακάτω:



5.1.5 Μεταβολή!

Έχετε σκεφτεί ποτέ τι υπάρχει πέρα από τα όρια του σκηνικού; Που πηγαίνει το αντικείμενο αν οδηγήσουμε εκτός του σκηνικού; Τι θα κάνει μόλις χαθεί από το οπτικό μας πεδίο; Η

εάν στα όρια, αναπήδησε

απάντηση είναι πως το αντικείμενο συνεχίζει την εκτέλεση των εντολών που του έχουμε δώσει κανονικά, απλά εμείς δεν μπορούμε να δούμε τη συμπεριφορά του. Επειδή είναι εξαιρετικά πιθανό να μη θέλουμε το αντικείμενό μας να βγαίνει εκτός της οθόνης του Scratch, για αυτό προσφέρεται η εντολή **εάν στα όρια, αναπήδησε**. Με αυτή την εντολή το αντικείμενο, όταν αγγίξει τα όρια της οθόνης, κάνει μεταβολή δηλαδή αλλάζει η κατεύθυνσή του κατά 180 μοίρες. Έτσι, συνεχίζει να κινείται εντός ορίων του πλαισίου. Ας δούμε ένα πολύ απλό παράδειγμα.

Έστω ότι το σκηνικό μας είναι ένα γήπεδο του τένις και το αντικείμενό μας είναι ένα μπαλάκι.



Από την παλέτα εντολών της κίνησης έχουμε τσεκάρει την επιλογή **κατεύθυνση** έτσι ώστε να μπορούμε να παρακολουθούμε τις αλλαγές της. Δίνουμε την εντολή κινήσου 1000 βήματα στο μπαλάκι και αμέσως μετά βάζουμε την εντολή **εάν στα όρια αναπήδησε** για να μπορεί το αντικείμενο να στραφεί κατά 180 μοίρες εάν βρεθεί στα όρια.



Στην επόμενη εικόνα μπορείτε να δείτε ότι το μπαλάκι βρίσκεται εντός ορίων παρόλο που προσδιορίσαμε μεγάλο αριθμό βημάτων. Η κατεύθυνσή του έχει αλλάξει κατά 180 μοίρες, πράγμα που σημαίνει πως όταν ξεπέρασε τα όρια, επέστρεψε σε αυτά και έκανε μεταβολή.



5.1.6 Έλεγχος δυνατοτήτων κίνησης του αντικειμένου

Το Scratch μας δίνει επιπλέον πληροφορίες για τις δυνατότητες κίνησης του αντικειμένου αλλά και τη θέση και την κατεύθυνσή του κάθε στιγμή, όπως είδαμε προηγουμένως. Τα τρία αριστερά κουμπιά στην επόμενη εικόνα προσδιορίζουν το κατά πόσο επιτρέπουμε σε ένα αντικείμενο να περιστραφεί.



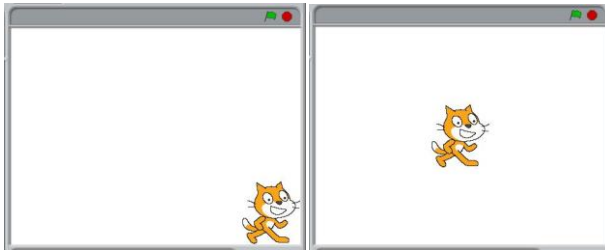
Αυτό σημαίνει ότι είχατε τη δυνατότητα να περιστρέψουμε τα αντικείμενά μας προς οποιαδήποτε κατεύθυνση γιατί ήταν ενεργή η συγκεκριμένη επιλογή (**περιστρέψιμο**). Η άλλη επιλογή, **μόνο το πρόσωπο αριστερά-δεξιά**, επιτρέπει σε ένα αντικείμενο να στρέφεται μόνο δεξιά ή αριστερά. Το αντικείμενο, σε αυτήν την περίπτωση, δεν μπορεί να στραφεί και να κινηθεί σε οποιαδήποτε άλλη γωνία εκτός των 90 και -90 μοιρών. Αν θέλετε να δημιουργήσετε μια εφαρμογή, όπως το παιχνίδι Super Mario στο οποίο ο ήρωας κοιτάει μόνο δεξιά και αριστερά, πρέπει να περιορίσετε τις αλλαγές κατεύθυνσης του αντικειμένου σας μόνο στον οριζόντιο άξονα. Στο ίδιο τμήμα της οθόνης του Scratch μπορούμε να δούμε ανά πάσα στιγμή την τρέχουσα θέση του. Αυτή η πληροφορία θα μας χρειαστεί στην επόμενη ενότητα.

5.2 Κίνηση με συντεταγμένες

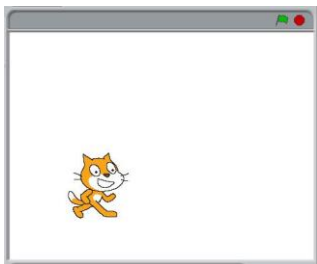
Στην προηγούμενη υποενότητα είδαμε πως μπορούμε να κάνουμε το χαρακτήρα σας να κινηθεί με την εντολή **κινήσου...βήματα** που αποτελεί και την απλούστερη εντολή της αντίστοιχης παλέτας **Κίνηση**. Ωστόσο υπάρχουν φορές που θέλουμε να ελέγξουμε με μεγαλύτερη ακρίβεια τη θέση του αντικειμένου μας πάνω στη σκηνή.

Αν παρατηρήσετε την παλέτα των εντολών κίνησης θα διαπιστώσετε ότι υπάρχει ένας μεγάλος αριθμός εντολών που αναφέρονται στις τιμές «κάποιων» x και y. Οι συγκεκριμένες εντολές έχουν και αυτές ως αποτέλεσμα την κίνηση του χαρακτήρα με τρόπο παρόμοιο με αυτόν της πρώτης εντολής κίνησης που μάθαμε. Τι ακριβώς αντιπροσωπεύουν όμως τα x και y; Θα ξεκινήσουμε τη διερεύνηση αυτών των εντολών αφού όμως πρώτα εξετάσουμε προσεκτικά το σύστημα συντεταγμένων του Scratch.

Τι θα απαντούσατε αν κάποιος σας ρωτούσε ποια είναι η θέση του χαρακτήρα στην οθόνη εμφάνισης για κάθε μια από τις περιπτώσεις των επόμενων δυο εικόνων; Το πιο πιθανό είναι να δίνετε μία απάντηση του τύπου «στην κάτω δεξιά γωνία» και «στο κέντρο» αντίστοιχα.

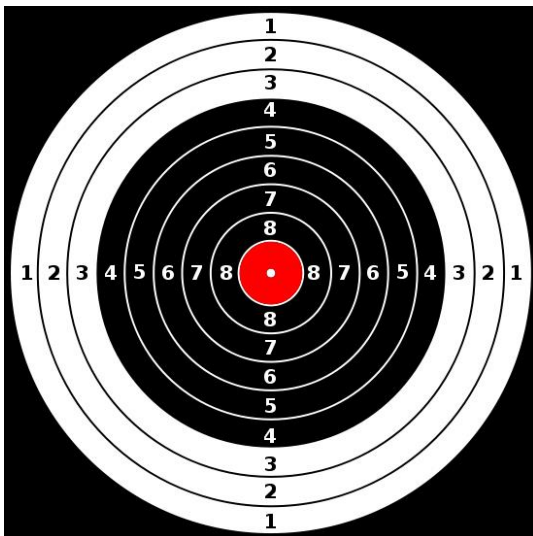


Τι θα απαντούσατε αλήθεια στην ίδια ερώτηση για την περίπτωση της επόμενης εικόνας;



Όπως μπορείτε να διαπιστώσετε και μόνοι σας στην τελευταία ερώτηση δεν μπορείτε να απαντήσετε ούτε καν περιγραφικά καθώς η θέση του χαρακτήρα στην οθόνη εμφάνισης είναι δύσκολο να προσδιοριστεί. Η μήπως δεν είναι;

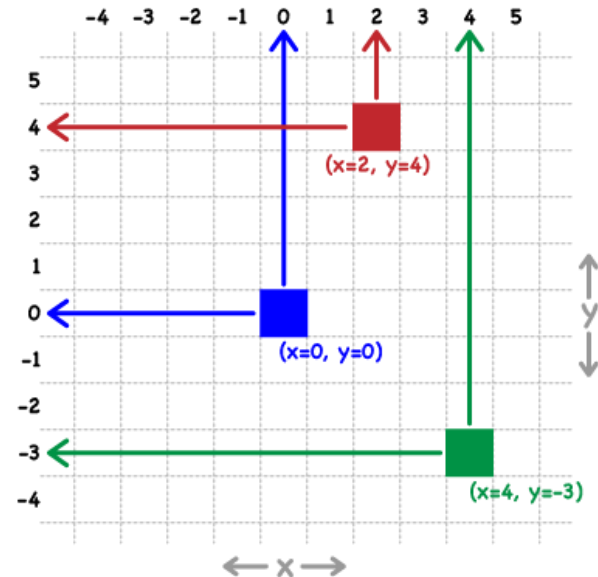
Φανταστείτε ότι έχετε στο έργο σας το σκηνικό της επόμενης εικόνας. Αν ο χρήστης έριχνε ένα αντικείμενο-βέλος σε κάποια θέση τι βαθμολογία θα έπαιρνε η «βολή» του; Η σωστή βαθμολογία εξαρτάται από την ικανότητά μας να αναγνωρίζουμε την ακριβή θέση του βέλους στον στόχο.



Κάτι αντίστοιχο κάνουμε και στο Scratch. Η σκηνή του Scratch αποτελείται από εικονοστοιχεία (pixels), μικρά τετραγωνάκια που υπάρχουν στην οθόνη σε μορφή ενός πίνακα που περιέχει γραμμές και στήλες. Έτσι μπορούμε να προσδιορίσουμε τη θέση τους βάσει της γραμμής και τη στήλης στην οποία ανήκουν. Στο Scratch ο αριθμός της στήλης συμβολίζεται με x (η οριζόντια θέση δηλαδή) και της γραμμής με y (η κάθετη θέση) ενώ ένα ζεύγος (x, y) δίνει την ακριβή θέση ενός εικονοστοιχείου στην οθόνη καθώς καθορίζει σε ποια γραμμή και σε ποια στήλη αυτό

ανήκει. Το ζεύγος αυτό ονομάζεται συντεταγμένες. Πόσο διαφέρει αυτό το σύστημα συντεταγμένων από το καρτεσιανό σύστημα;

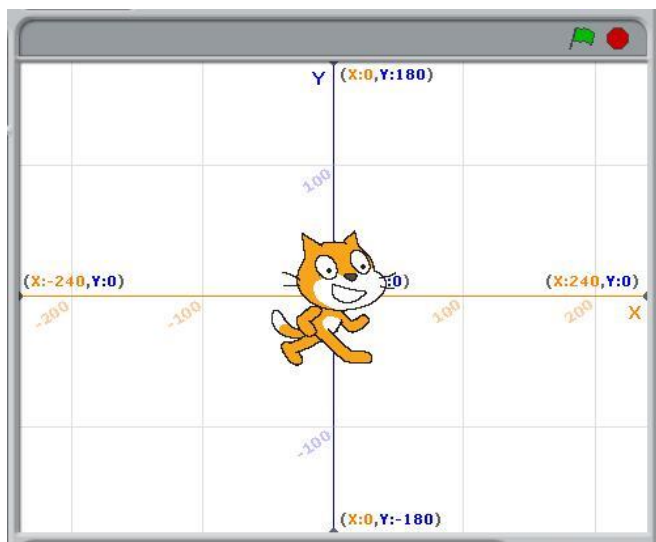
Στην επόμενη εικόνα έχουμε μεγεθύνει την οθόνη εμφάνισης του Scratch περίπου 30 φορές και έχουμε χρωματίσει 3 εικονοστοιχεία των οποίων και σημειώσαμε τη θέση από κάτω προσδιορίζοντας τις τιμές των x και y .



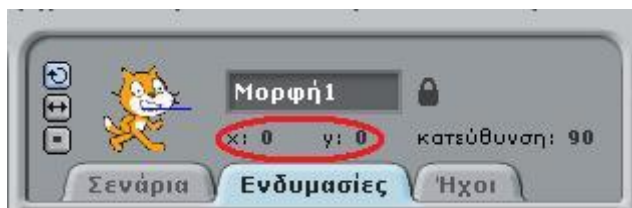
Παρατηρούμε λοιπόν ότι:

- ✓ Το μπλε εικονοστοιχείο είναι ακριβώς στο κέντρο της οθόνης και το Scratch θεωρεί τη συγκεκριμένη στήλη ως 0 και τη συγκεκριμένη γραμμή επίσης ως 0. Δηλαδή, οι συντεταγμένες ενός αντικειμένου στο κέντρο της οθόνης είναι $(x = 0, y = 0)$.
- ✓ Το κόκκινο εικονοστοιχείο είναι στη στήλη με αριθμό 2 και στη γραμμή με αριθμό 4 άρα έχει συντεταγμένες $(x = 2, y = 4)$.
- ✓ Το πράσινο εικονοστοιχείο βρίσκεται στη στήλη με αριθμό 4 και στη γραμμή με αριθμό -3 οπότε έχει συντεταγμένες $(x = 4, y = -3)$.

Πάμε τώρα να δούμε το χαρακτήρα μας πάνω στο σύστημα συντεταγμένων που μόλις αναφέραμε. Σε ένα νέο έργο δημιουργούμε ένα νέο σκηνικό από την εικόνα xy -grid που βρίσκεται στη βιβλιοθήκη υποβάθρων του Scratch. Εάν η διαδικασία γίνει σωστά η σκηνή μας θα πρέπει να μοιάζει με αυτήν της επόμενης εικόνας:



Το σκηνικό αυτό δείχνει τους άξονες των συντεταγμένων και έτσι μπορούμε να καταλάβουμε καλύτερα τις διάφορες θέσεις των αντικειμένων. Παρατηρήστε ότι η γάτα σε ένα νέο έργο εμφανίζεται στο κέντρο του σκηνικού και έχει συντεταγμένες $x=0$ και $y=0$. Όπως είπαμε και προηγουμένως, αυτό μπορείτε να το επιβεβαιώσετε πατώντας στο περιβάλλον του Scratch στο εικονίδιο «Μορφή 1» και παρατηρώντας στο πάνω μέρος του περιβάλλοντος εργασίας τις τιμές που έχουν τα x και y του συγκεκριμένου αντικειμένου.

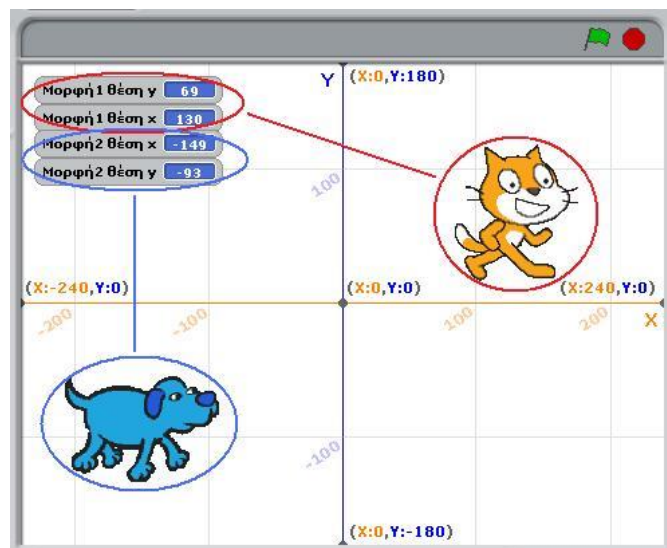


Ένας δεύτερος τρόπος να γνωρίζετε κάθε στιγμή τις συντεταγμένες της θέσης του χαρακτήρα σας είναι να τσεκάρετε από την παλέτα των εντολών κίνησης τις επιλογές θέση x και θέση y.



Το αποτέλεσμα θα είναι να εμφανιστούν στην πάνω αριστερή γωνία του σκηνικού οι συντεταγμένες του αντικειμένου, οι οποίες θα αλλάζουν αυτόματα κάθε φορά που ο χαρακτήρας σας αλλάζει θέση. Όπως ακριβώς συνέβαινε με την επιλογή Κατεύθυνση που είδαμε στην προηγούμενη υποενότητα. Όλες αυτές τις επιλογές θα τις ονομάζουμε μεταβλητές κατάστασης καθώς αναφέρονται σε χαρακτηριστικές ιδιότητες των αντικειμένων που γίνονται ορατές μέσα στη σκηνή.

Είναι σημαντικό να τονίσουμε ότι οι μεταβλητές κατάστασης x, y είναι διαφορετικές για κάθε αντικείμενο. Για να τις εμφανίσετε, πρέπει να επιλέξετε πρώτα το αντικείμενο και μετά να τσεκάρετε τις αντίστοιχες επιλογές. Για να τις απομακρύνετε από την οθόνη, πρέπει να κάνετε την αντίστροφη διαδικασία. Στην παρακάτω εικόνα βλέπετε διαφορετικές μεταβλητές κατάστασης για δυο αντικείμενα.



Δοκιμάστε και μόνοι σας. Μετακινήστε το χαρακτήρα σε διάφορες θέσεις στο σκηνικό και παρατηρήστε τις τιμές που παίρνουν οι μεταβλητές κατάστασης σε κάθε περίπτωση. Πλέον ανά πάσα στιγμή μπορείτε να γνωρίζετε την ακριβή θέση των αντικειμένων σας.

Καλό θα ήταν να θυμάστε:

Α) ότι στα **οριζόντια άκρα**, οι τιμές του x είναι οι **-240** στο αριστερό όριο και **240** στο δεξί όριο,

Β) ότι στα **κάθετα άκρα**, οι τιμές του y είναι οι **180** στο πάνω όριο και **-180** στο κάτω όριο.

Άρα, η οθόνη του Scratch έχει πλάτος $240 \times 2 = 480$ και ύψος $180 \times 2 = 360$.

Είμαστε έτοιμοι να αρχίσουμε να κινούμε τους χαρακτήρες μας βάσει του συστήματος συντεταγμένων

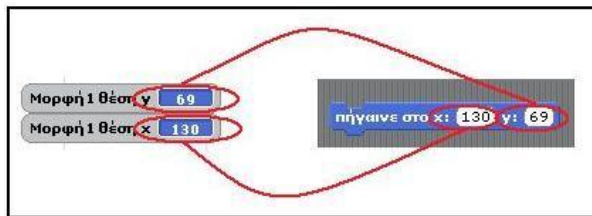
5.2.1 Ραντεβού στο (x, y)

Με την εντολή **πήγαινε στο x, y** που βρίσκεται στην παλέτα εντολών της κίνησης, το επιλεγμένο αντικείμενο μεταβαίνει στις συντεταγμένες (x, y) που του ορίζουμε. Η παραπάνω εντολή όταν εκτελεστεί έχει σαν αποτέλεσμα την άμεση μετακίνηση του αντικειμένου μας στη θέση (x, y) της οθόνης ανεξάρτητα από τη θέση που είχε το αντικείμενο πριν από την εκτέλεση της εντολής (σε αντίθεση με την εντολή κινήσου...βήματα).

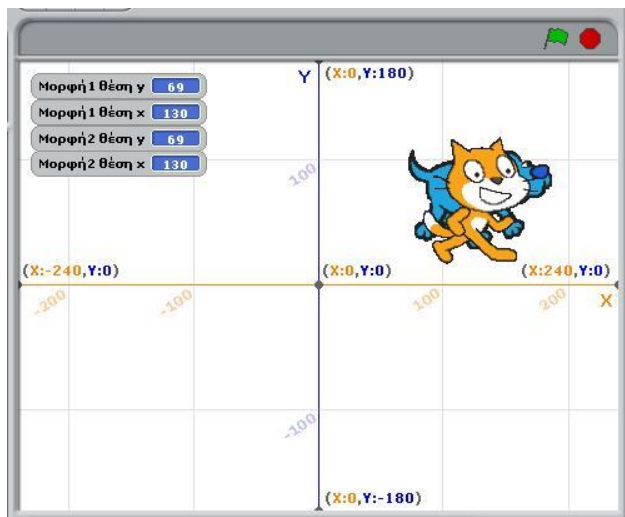
Για να δούμε τώρα πως με τη χρήση αυτής της νέας εντολής μπορεί ο σκύλος να μετακινηθεί στο σημείο που είναι και η γάτα στο σενάριο μας! Το μόνο που χρειάζεται είναι να γνωρίζουμε τις συντεταγμένες της γάτας και να στείλουμε το σκύλο απευθείας στις συντεταγμένες αυτές. Έχοντας ενεργοποιήσει τις μεταβλητές κατάστασης που αφορούν τη θέση, βλέπουμε τις συντεταγμένες της γάτας (Μορφή 1). Στο σενάριο του σκύλου (Μορφή 2) τοποθετούμε την παραπάνω εντολή συμπληρώνοντας στα x και y τις συντεταγμένες αυτές.

θέση x

θέση y



Εάν λοιπόν στο σενάριο για τη Μορφή 2 εκτελεστεί η παραπάνω εντολή, τότε το αποτέλεσμα θα είναι αυτό που φαίνεται στην επόμενη εικόνα.

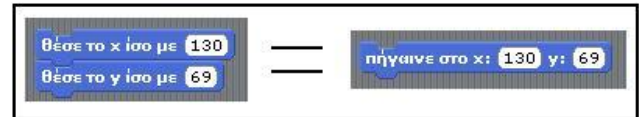


Παρατηρήστε στη δεύτερη εικόνα τις συντεταγμένες των δύο χαρακτήρων. Μετά την εκτέλεση του προγράμματος είναι ακριβώς οι ίδιες και οι δύο χαρακτήρες βρίσκονται πλέον στην ίδια ακριβώς θέση. Ο στόχος επιτεύχθηκε με τη χρήση μόλις μιας εντολής και χωρίς να μπλεχτείτε με εντολές αλλαγής κατεύθυνσης.

Υπάρχουν ωστόσο περιπτώσεις όπου χρειάζεται να αλλάξουμε μόνο το x ή μόνο το y από τις συντεταγμένες του χαρακτήρα μας. Όταν θέλουμε να μετακινήσουμε το χαρακτήρα μας μόνο οριζόντια ή μόνο κατακόρυφα, δεν χρειάζεται να αλλάξουμε και τις δύο συντεταγμένες του αλλά μόνο μία από αυτές ανάλογα με την περίπτωση. Σκεφτείτε για παράδειγμα ότι θέλετε ο χαρακτήρας σας να ανέβει μία κατακόρυφη σκάλα. Ποια από τις συντεταγμένες του θα αλλάζατε για να το καταφέρετε; Η άλλη θα χρειαζόταν να αλλάξει;

Για τέτοιες περιπτώσεις, το Scratch περιλαμβάνει δύο εντολές που μετακινούν ένα αντικείμενο μόνο στον οριζόντιο ή τον κατακόρυφο άξονα: οι εντολές θέσε το x ίσο με... και θέσε το y ίσο με... μετακινούν το χαρακτήρα αλλάζοντας τιμή μόνο στο x ή στο y αντίστοιχα.

Ο συνδυασμός των δύο παραπάνω εντολών ισοδυναμεί με την εντολή πήγαινε στο x, y οπότε το προηγούμενο σενάριο θα μπορούσε εναλλακτικά να υλοποιηθεί με αυτές τις δύο εντολές προκαλώντας το ίδιο ακριβώς αποτέλεσμα.

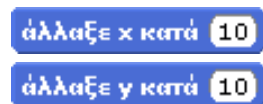


Αν και υπάρχει η παραπάνω ισοδυναμία, θα πρέπει να γνωρίζετε και τους δύο τρόπους και ανάλογα με την περίπτωση να διαλέγετε τη χρήση του πιο κατάλληλου. Θα δούμε τέτοιες περιπτώσεις σε επόμενα κεφάλαια και κυρίως στα παιχνίδια.

5.2.2 Μετακίνηση με σχετικές συντεταγμένες

Η εντολή κινήσου...βήματα μετακινούσε ένα αντικείμενο ανάλογα με την αρχική του θέση. Πόσο ακριβής ήταν όμως η μετακίνηση αυτή; Η μονάδα μετακίνησης ήταν τα βήματα και η μετακίνηση σχετιζόταν με την κατεύθυνση. Και αν απλά θέλουμε να μετακινήσουμε λίγο πιο δεξιά ένα αντικείμενο χωρίς να αλλάξουμε την κατεύθυνσή του; Ή αν θέλουμε να το μετακινήσουμε προς τα πάνω αλλά για μια απόσταση συγκεκριμένη;

Για να μας βοηθήσει στις προηγούμενες ενέργειες το Scratch μας παρέχει τις εντολές άλλαξε x κατά... και άλλαξε y κατά...



Με τις εντολές αυτές μπορούμε να μεταβάλλουμε κατά όσο θέλουμε τις τιμές των x και y αντίστοιχα. Προσοχή! Δεν προσδιορίζουμε σημείο στην οθόνη αλλά το κατά πόσα εικονοστροιχεία θέλουμε να μεταβάλλουμε την τρέχουσα θέση του αντικειμένου.

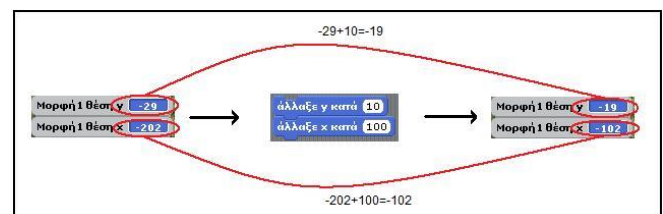
Η διαφορά δηλαδή είναι ότι οι εντολές αυτές μεταβάλλουν τα x και y σε σχέση με τις τιμές που αυτά έχουν τη δεδομένη στιγμή, σε αντίθεση με τις εντολές θέσε το x και θέσε το y οι οποίες μετακινούν το αντικείμενο στις συγκεκριμένες θέσεις x και y.

Δείτε ένα παράδειγμα χρήσης των νέων αυτών εντολών στην επόμενη εικόνα. Αρχικά οι συντεταγμένες της θέσης του χαρακτήρα μας ήταν x = -202 και y = -29. Μετά την εκτέλεση των εντολών άλλαξε x κατά 100 και άλλαξε y κατά 10 έχουμε:

$$A) X = -202 + 100 = -102 \text{ και}$$

$$B) Y = -29 + 10 = -19$$

Παρατηρείτε και μόνοι σας ότι το x που προκύπτει εξαρτάται από την τιμή που είχε πριν την εκτέλεση της εντολής (-202) και από τη μεταβολή που ορίζουμε στην αντίστοιχη εντολή (100). Αντίστοιχα προκύπτει και το νέο y.



Τι τιμές θα είχαν τελικά τα x και y αν με ίδια αρχική κατάσταση εκτελούσαμε τις εντολές θέσε το x ίσο 100 και θέσε το y ίσο με 10;

5.2.3 Μετακίνηση προς αντικείμενο

Δημιουργήστε ένα νέο έργο και τρέξτε το παρακάτω σενάριο. Η εντολή για πάντα βρίσκεται στην παλέτα εντολών ελέγχου.



Τι παρατηρείτε; Όσο το πρόγραμμά μας τρέχει, η γάτα μετακινείται στη θέση που βρίσκεται ο δείκτης του ποντικιού. Από την περιγραφή των εντολών, μπορούμε να συνοψίσουμε: όταν γίνει κλικ στην πράσινη σημαία, για πάντα (δηλαδή μέχρι το τέλος του προγράμματος) θέλουμε το αντικείμενό μας να πηγαίνει στο δείκτη του ποντικιού.

Η εντολή πήγαινε στο... μας δίνει δυο δυνατότητες:

α) να πούμε στο αντικείμενό μας να μετακινηθεί στη θέση του δείκτη του ποντικιού, και



β) να πούμε στο αντικείμενό μας να πάει στη θέση κάποιου άλλου αντικειμένου που βρίσκεται στη σκηνή.

Για να δοκιμάσετε την δεύτερη επιλογή, εισάγετε έναν δεύτερο χαρακτήρα στο έργο σας και δείτε ότι η εντολή πήγαινε στο... εμφανίζει στη λίστα της το όνομα του νέου αντικειμένου. Αν το επιλέξετε, το αντικείμενό σας θα μετακινηθεί ακριβώς στη θέση του δεύτερου αντικειμένου.

5.3 Ομαλή κίνηση

Όλες οι περιπτώσεις κίνησης που περιγράψαμε προηγουμένως, οδηγούν στην ακαριαία μετακίνηση του αντικειμένου τη στιγμή εκτέλεσης του προγράμματος. Είτε κινούσαμε ένα αντικείμενο με σχετικές συντεταγμένες είτε με απόλυτες, αυτομάτως βρισκόταν στην επόμενη θέση του. Δεν είναι λογικό όμως κάποιες φορές να θέλουμε να βλέπουμε την μετακίνηση, την πορεία του αντικειμένου;

Για το συγκεκριμένο σκοπό το Scratch μας παρέχει την εντολή κινήσου ομαλά...δευτ. στο x, y. Στη συγκεκριμένη εντολή ορίζουμε εμείς που επιθυμούμε να πάει



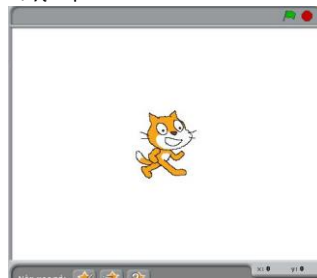
το αντικείμενό μας και σε πόσο χρόνο. Ας δούμε μέσω ενός παραδείγματος πως λειτουργεί αυτή η εντολή:

Έστω ότι το αντικείμενο μας βρίσκεται στην αρχική κατάσταση (0,0) και θέλουμε να το μετακινήσουμε με σταθερή ταχύτητα στο σημείο (166,96). Η μετακίνηση του από το σημείο (0,0) στο (166,96) επιθυμούμε να διαρκέσει 5 δευτερόλεπτα. Στο πρώτο λευκό κουτάκι της εντολής ορίζουμε τη διάρκεια της μετακίνησης σε 5 δευτερόλεπτα. Στις συντεταγμένες x και y θα θέσουμε το σημείο (166,96) στο οποίο θέλουμε να καταλήξει το αντικείμενο μας.

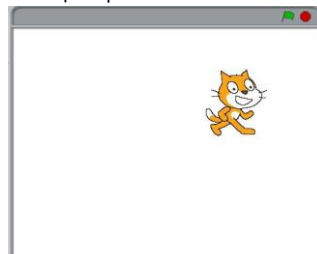


Πατώντας την πράσινη σημαία έχουμε τις εξής καταστάσεις:

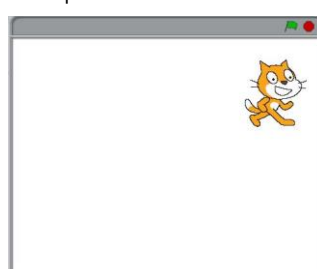
Αρχική:



Ενδιάμεση:



Τελική:



Και ναι! Ο ήρωας μας γλίστρησε στο σημείο και η κίνησή του διήρκεσε όσο χρόνο εμείς ορίσαμε! Είμαστε έτοιμοι να κάνουμε κι άλλες βόλτες με τους πρωταγωνιστές μας;

5.4 "Καβουράκης και Αστεριάδης σε υποβρύχιες βόλτες"

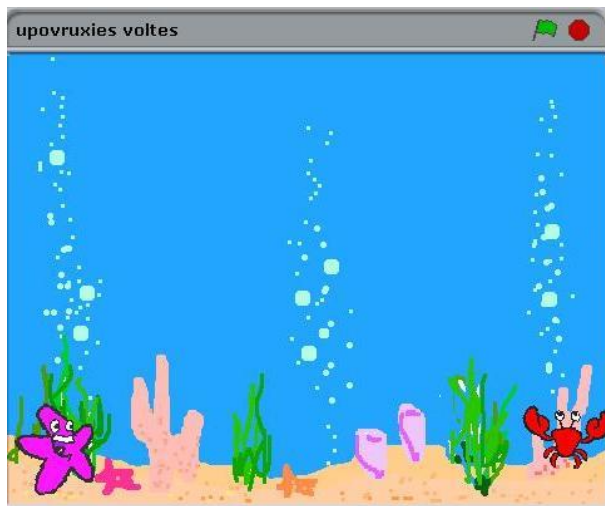
Στην εφαρμογή αυτή θα συνδυάσουμε τις προηγούμενες εντολές για να δημιουργήσουμε ένα ολοκληρωμένο έργο. Στόχος μας, να συντονίσουμε τις κινήσεις δυο αντικειμένων δημιουργώντας ένα υποθαλάσσιο χορευτικό! Ακολουθήστε τα επόμενα βήματα

[\[05_π03.sb\]](#):

Βήμα 1 : Εισάγουμε στο έργο μας τα αντικείμενα crab1-a και starfish1-a από τον φάκελο animals της βιβλιοθήκης αντικειμένων του Scratch. Ο κ. Καβουράκης (καβούρι) αντιστοιχίζεται στη μορφή 1 και κ. Αστεριάδης στη μορφή 2 (αστερίας).

Βήμα 2: Στην συνέχεια εισάγουμε για σκηνικό την επιλογή underwater από τον φάκελο nature της βιβλιοθήκης σκηνικών του Scratch.

Βήμα 3: Τοποθετούμε τα αντικείμενα στο κάτω μέρος του σκηνικού δεξιά και αριστερά όπως φαίνεται στην επόμενη εικόνα.



Βήμα 4: Το σενάριο για τις 2 μορφές θα είναι το εξής: Για τον κ. Καβουράκη



και για τον κύριο Αστεριάδη



ενώ η επιλογή που έχουμε κάνει για τις δυνατότητες κατεύθυνσης των 2 αντικειμένων είναι «περιστρέψιμο».

Τρέξτε το πρόγραμμα πατώντας την πράσινη σημαία και επιστρέψτε στο βιβλίο για να το αναλύσουμε εντολή προς εντολή:

Οι δύο πρώτες εντολές αφορούν τον προσδιορισμό της αρχικής θέσης των αντικειμένων. Οι εντολές αυτές τοποθετούν τους πρωταγωνιστές μας στις θέσεις όπου επιθυμούμε να ξεκινούν την κίνηση τους κάθε φορά που εκτελείται το πρόγραμμά μας. Συγκεκριμένα ο κ. Καβουράκης θα ξεκινάει πάντα από την θέση (200,-130) ενώ ο κ. Αστεριάδης από την θέση (-200,-130) όπως βλέπουμε παρακάτω:

Αρχικοποίηση για τον κ. Καβουράκη:



Αρχικοποίηση για τον κ. Αστεριάδη:



Είναι σημαντικό να επισημάνουμε ότι επειδή τα αντικείμενά μας μετακινούνται διαρκώς είτε επειδή τα σέρνουμε στην οθόνη είτε μέσω των σεναρίων που εκτελούμε, θα πρέπει να προνοούμε να ξεκινούν κάθε φορά από την ίδια θέση. Οι πρώτες δηλαδή εντολές κάθε προγράμματος πρέπει να τα τοποθετούν εκεί που θέλουμε να βρίσκονται όταν ξεκινάει η εκτέλεση του προγράμματος. Η διαδικασία αυτή ονομάζεται αρχικοποίηση θέσης και είναι ιδιαίτερα σημαντική για να μπορούμε να είμαστε βέβαιοι για την έκβαση των συμπεριφορών που αποδίδουμε στη συνέχεια στα αντικείμενα.

Με την τρίτη εντολή που παρουσιάζεται στις επόμενες εικόνες οι μορφές μεταβαίνουν ομαλά από την αρχική θέση τους στα σημεία (-200,130) και (200,-130) αντίστοιχα. Αλλάζουν δηλαδή θέσεις:

Καβουράκης



Αστεριάδης



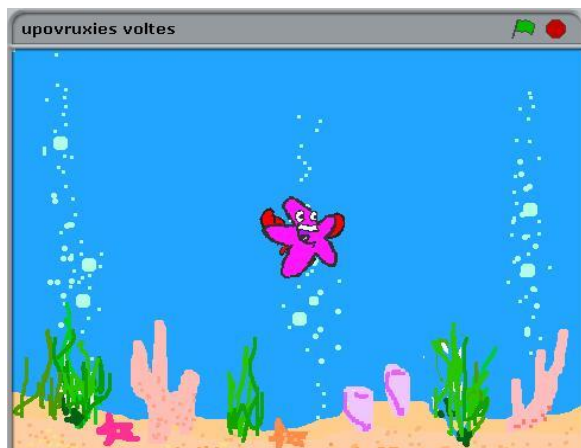
Με τις εντολές



για τον Αστεριάδη και



για τον Καβουράκι οι 2 μορφές μεταβαίνουν μετά την ομαλή τους μετακίνηση, ακαριαία στο ίδιο σημείο (3,13), όπως φαίνεται στην επόμενη εικόνα.



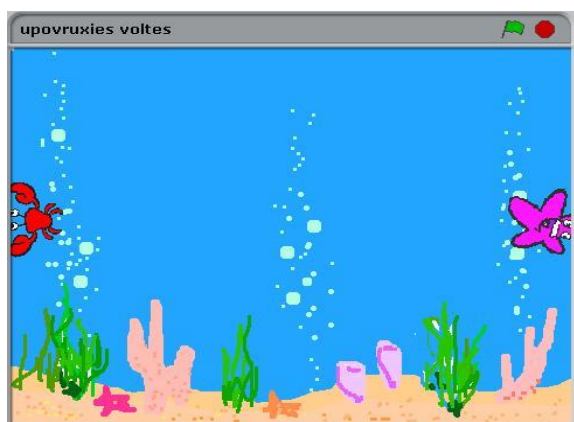
Με τις δυο επόμενες εντολές για τον Καβουράκι



και για τον Αστεριάδη



οι δύο μορφές κάνουν στροφή 90 μοιρών (αριστερά και δεξιά αντίστοιχα) και στη συνέχεια κινούνται ομαλά μέχρι τα σημεία (-220,20) και (220,20).

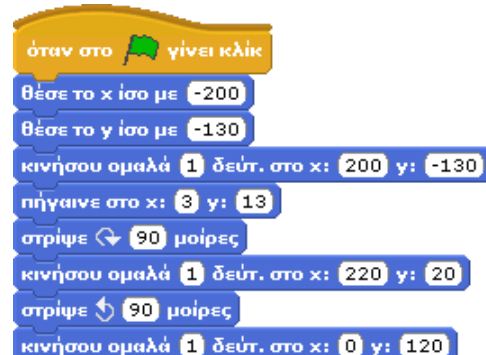


Επειδή θέλουμε οι 2 μορφές να ξανασυναντηθούν από το σημείο αυτό, στο σημείο (0,120), στρίβουμε κατά 90 μοίρες δεξιά τον Καβουράκι και 90 μοίρες αριστερά τον Αστεριάδη και τους δίνουμε την εντολή ομαλής κίνησης μέχρι το σημείο συνάντησης.

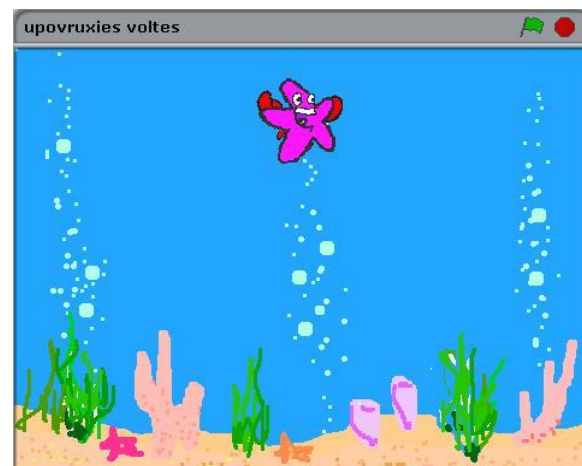
Καβουράκις:



Αστεριάδης:



Αποτέλεσμα:



Τέλος, θέλοντας να επαναφέρουμε τους ήρωες μας στην αρχική τους κατάσταση χρησιμοποιούμε την εντολή ομαλής κίνησης στα σημεία (200,-130) και (-200,-130) αντίστοιχα.

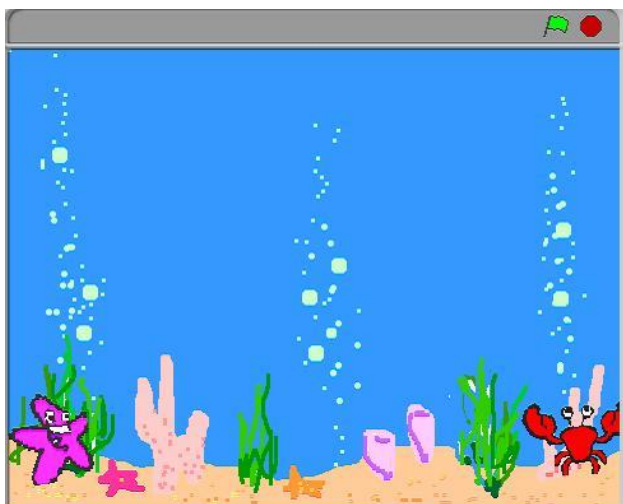
Σενάριο για τον κ.Καβουράκι:



Σενάριο για τον κ.Αστεριάδη:



και φτάνουμε στην τελική κατάσταση της ιστορίας μας:



Στο παράδειγμά μας έχουμε ορίσει η κάθε κίνηση να διαρκεί 1 δευτερόλεπτο. Πειραματιστείτε με τον χρόνο ανάλογα με το πόσο αργά ή γρήγορα θέλετε να μετακινούνται οι ήρωες σας!

Περίληψη

Σε αυτό το κεφάλαιο εξετάσαμε τις εντολές κίνησης των αντικειμένων μας που μας προσφέρονται από το Scratch. Συνοψίζοντας, μπορούμε να διακρίνουμε 3 τρόπους κίνησης:

A) (Κίνηση και κατεύθυνση) προσδιορίζουμε την κατεύθυνση και ορίζουμε συγκεκριμένο αριθμό βημάτων για το αντικείμενό μας,

B) (Κίνηση με συντεταγμένες) προσδιορίζουμε την απόλυτη θέση του αντικειμένου βάσει του συστήματος συντεταγμένων της σκηνής ή το μετακινούμε σχετικά βάση όμως του συστήματος συντεταγμένων,

Γ) (Ομαλή κίνηση) προσδιορίζουμε τις απόλυτες συντεταγμένες της νέας θέσης του αντικειμένου αλλά και το χρόνο που θέλουμε να διαρκέσει η κίνηση αυτή.

Δημιουργήσαμε τα πρώτα ολοκληρωμένα προγράμματά μας, αλλά μόλις αρχίσαμε...

Ερωτήσεις

- Όταν σε ένα έργο βλέπετε ένα χαρακτήρα να περπατάει στην οθόνη σας, και όταν αγγίζει τα όρια της οθόνης κάνει μεταβολή, ποιές εντολές θεωρείτε ότι έχουν χρησιμοποιηθεί;
- Σε τι διαφοροποιείται η εντολή **κινήσου ομαλά...δευτ. στο x,y**, σε σχέση με την εντολή **πήγαινε στο x,y**;
- Κατηγοριοποιήστε τους τρόπους κίνησης ενός αντικειμένου και δώστε τις διαφορές τους.

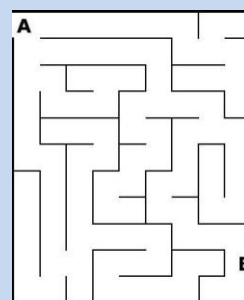
- Με ποιους διαφορετικούς τρόπους θα μπορούσατε να τοποθετήσετε ένα αντικείμενο στο σημείο (200,-130);
- Αν δημιουργούσατε ένα πρόγραμμα στο οποίο ο κεντρικός χαρακτήρας κινούνταν μόνο στη κάθετη διάσταση, ποια εντολή θα επιλέγατε να χρησιμοποιήσετε;
- Ποιά εντολή πιστεύετε ότι χρησιμοποιείται για την επανατοποθέτηση του rasman στην αρχική του θέση αφού τον πιάσει το φαντασματάκι;
- Αναγνώρισε τις παρακάτω εντολές κίνησης, συμπλήρωσε τα κενά και περιγράψτε σύντομα τη λειτουργία τους.



Δραστηριότητες

- 1) Στόχος σας είναι η προσομοίωση ενός παιχνιδιού, πιο συγκεκριμένα του παιχνιδιού «Λαβύρινθος».

Αφού κατεβάσετε από το διαδίκτυο μια εικόνα λαβυρίνθου και την θέσετε ως σκηνικό, με τα κατάλληλα εργαλεία, πρέπει να βρείτε όλες τις συντεταγμένες από τις οποίες θα περάσει ένα αντικείμενο, για να διασχίσει τη διαδρομή και να φτάσει από το σημείο εκκίνησης στο σημείο τερματισμού.



(Tip: χρησιμοποιήστε από την παλέτα εντολών της κίνησης τις μεταβλητές κατάστασης **θέση x**, **θέση y**)

Χρησιμοποιήστε τις συντεταγμένες που βρήκατε και προγραμματίστε το αντικείμενο, ώστε να διασχίσει την διαδρομή από το A σημείο, στο B.

- 2) Φτιάξτε τη δική σας τρομακτική ιστορία. Ο ήρωας σας θα διασχίζει ένα σκοτεινό μονοπάτι, και ξαφνικά, θα εμφανιστεί μπροστά του ένα φιλικό φάντασμα. Αυτός όμως αγνοώντας το γεγονός ότι είναι φιλικό, θα αρχίσει να τρέχει προς την αντίθετη κατεύθυνση.

(Tip: θυμηθείτε ότι ένα αντικείμενο μπορεί να βρίσκεται και εκτός της σκηνής του Scratch)



Κεφάλαιο 6: Ζωγραφική

Σε αυτό το κεφάλαιο:

- 6.1 Ζωγραφική
- 6.2 Απλά ζωγράφισε
- 6.3 Χρώμα, σκιά και μέγεθος
- 6.4 Παράδειγμα

«Ζωγραφίζω πράγματα που σκέφτομαι, όχι πράγματα που βλέπω!»
(Πικάσο)

6.1 Ζωγραφική

Στο προηγούμενο κεφάλαιο μάθαμε πως μπορούμε να δημιουργούμε σενάρια που κινούν τους χαρακτήρες μας. Τι θα λέγατε, αν μπορούσατε να πείτε στους χαρακτήρες σας να αφήνουν ένα ίχνος καθώς κινούνται; Να κινούνται και να ζωγραφίζουν δηλαδή ταυτόχρονα! Φαντάζεστε πολλούς χαρακτήρες να κινούνται ταυτόχρονα και να σχεδιάζουν πολύχρωμα σχήματα; Το Scratch μας δίνει αυτήν την εκπληκτική δυνατότητα, παρέχοντάς σχετικές εντολές. Αρκεί βεβαίως να μάθουμε να τις αξιοποιούμε...

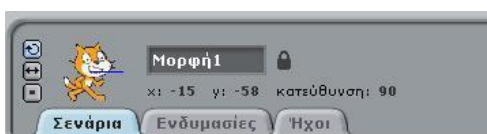
Οι εντολές αυτές βρίσκονται μέσα στην παλέτα εντολών **Πένα** και πολλά δημιουργικά παραδείγματα που θα ακολουθήσουν, θα σας κάνουν να εκπλαγείτε από τις δυνατότητές τους. Επειδή οι εντολές είναι απλές, θα αναλυθούν αρχικά όλες ακολουθούμενες από μικρά παραδείγματα και στο τέλος του κεφαλαίου θα τις συνδυάσουμε σε ένα μεγαλύτερο παράδειγμα.

6.2 Απλά ζωγράφισε

6.2.1 Κατέβασε πένα

Η εντολή **κατέβασε πένα** μετατρέπει το αντικείμενό μας σε ένα μολύβι-μαρκαδόρο και μας δίνει τη δυνατότητα να ζωγραφίζουμε καθώς το κινούμε. Είναι ανάλογη με το ακούμπημα του μολυβιού στο χαρτί (αφήνει το ίχνος της πέννας κατά την κίνηση του αντικειμένου). Όταν εκτελείτε η εντολή **κατέβασε πένα**, εμφανίζεται ένα μικρό ορθογώνιο παραλληλόγραμμο στις ιδιότητες του αντικειμένου που μας δείχνει κάθε στιγμή το χρώμα της πέννας.

Πριν...



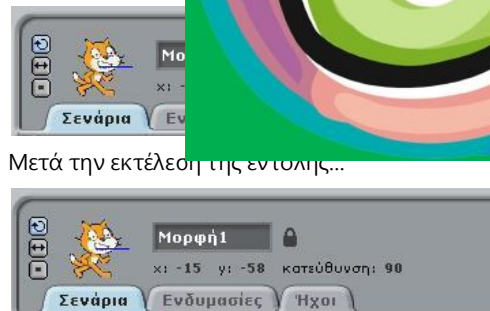
Μετά την εκτέλεση της εντολής...



6.2.2 Σήκωσε πένα

Για να "σηκώσουμε" την εντολή **σηκώσε πένα** χρωματιστό ορθογώνιο...

Πριν...



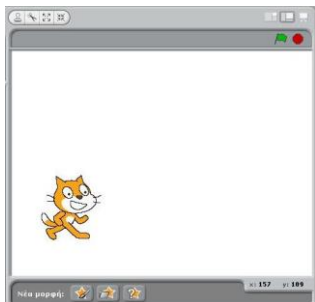
Μετά την εκτέλεση της εντολής...

Γιατί να σηκώσουμε την πένα; Γιατί απλά το κομμάτι της σχεδίασης τελειώνει κάποια στιγμή και γιατί ο πρωταγωνιστής μας μπορεί να έχει και άλλες δουλειές! Ή γιατί ίσως χρειαστεί να σχεδιάσουμε ένα σχήμα σε ένα άλλο σημείο της οθόνης και συνεπώς θέλουμε να μετακινήσουμε το αντικείμενό μας στη νέα θέση χωρίς να αφήσει το αντίστοιχο ίχνος.

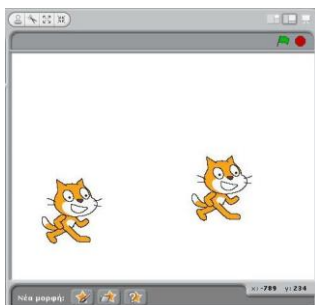
6.2.3 Σφραγίδα

Αν θέλουμε να αφήσουμε στην οθόνη το αποτύπωμα του αντικειμένου μας χρησιμοποιούμε την εντολή **σφραγίδα**. Θυμάστε την αντίστοιχη επιλογή στον επεξεργαστή Ζωγραφικής; Επιλέγαμε μια περιοχή από το καμβά μας και στη συνέχεια μπορούσαμε να την αναπαράγουμε όσες φορές θέλαμε. Η εντολή **σφραγίδα** προκαλεί το ίδιο αποτέλεσμα αφήνοντας ένα αποτύπωμα του αντικειμένου μας στο σημείο στο οποίο βρίσκεται. Φυσικά, αυτό θα φανεί αφού μετακινήσουμε το αντικείμενο.

Πριν...



Μετά την εκτέλεση της εντολής και τη μετακίνηση της γάτας...



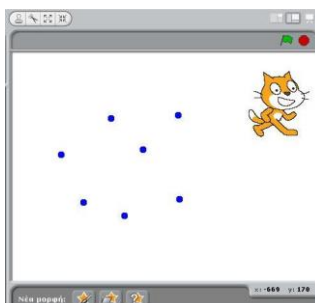
Δεν πρέπει να μπερδέψουμε, τη σφραγίδα με το ίδιο το αντικείμενο. Αφού το αποτύπωμα αφηθεί στην οθόνη, δεν μπορούμε να το χειριστούμε, ενσωματώνεται με το υπόβαθρο του έργου μας.

6.2.4 Καθάρισε

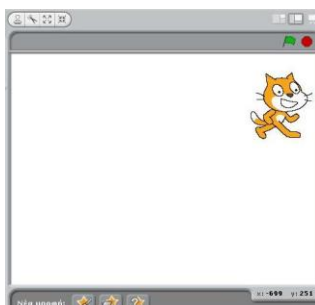
Και τι γίνεται αν έχουμε σχεδιάσει λίγο παραπάνω από όσο πρέπει; Η εντολή **καθάρισε** καθαρίζει την οθόνη από οτιδήποτε έχουμε ζωγραφίσει.

καθάρισε

Πριν...



Μετά την εκτέλεση της εντολής...

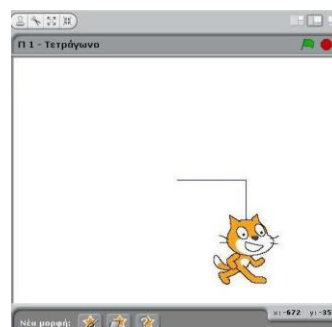
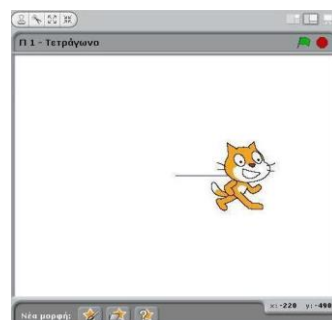
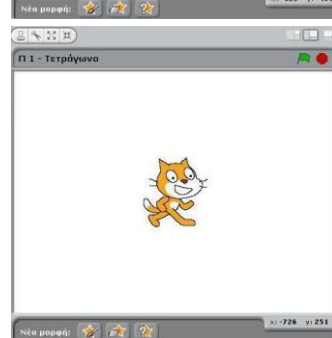
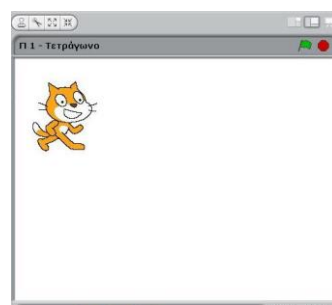


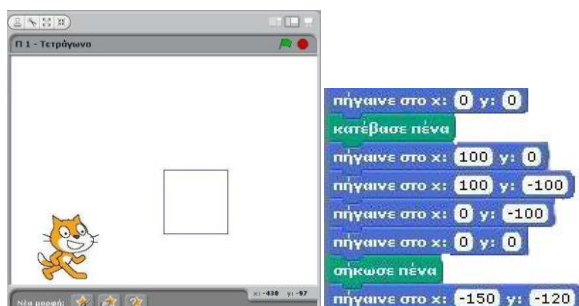
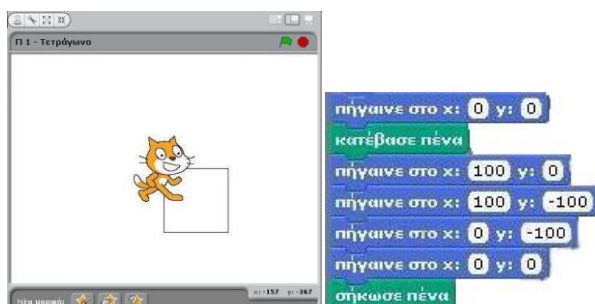
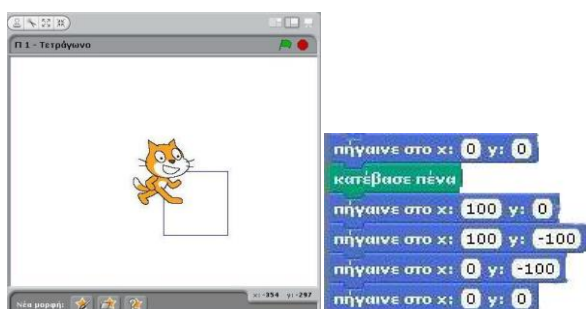
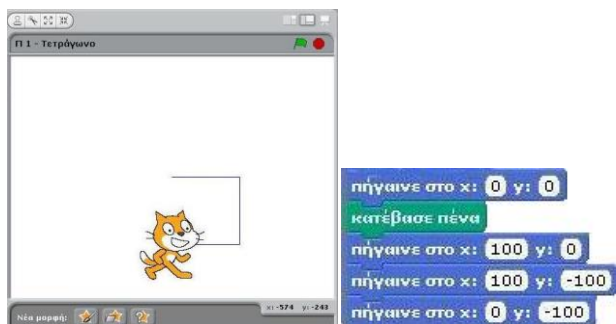
Είναι σημαντικό να σημειώσουμε ότι όταν σχεδιάζουμε στο Scratch, τότε τα σχέδιά μας δε διαγράφονται αυτόματα όταν εκτελούμε ξανά το έργο μας. Συνεπώς, η οθόνη μας αρχίζει και μπουτζουρώνεται πολύ! Για αυτό το λόγο συνιστάται η χρήση της **καθάρισε** στην αρχή κάθε έργου που περιλαμβάνει σχεδίαση με τους χαρακτήρες μας. Γιατί όχι στο τέλος;

6.2.5 Παραδείγματα

Ας δούμε τώρα πρακτικά πως μπορούμε να συνδυάσουμε την εντολή κίνησης **πήγαινε στο x:... y:...** με τις **κατέβασε πένα**, **σήκωσε πένα** για να δημιουργήσουμε ένα τετράγωνο. Έστω ότι θέλουμε να το δημιουργήσουμε ένα τετράγωνο από το κέντρο της οθόνης (συντεταγμένες $x=0, y=0$) με ακμές των 100 εικονοστοιχείων.

Αρχικά, θα το ζωγραφίσουμε δεξιόστροφα (με την φορά των δειχτών του ρολογιού), κινούμενοι κατά σειρά από την αρχική θέση $x=0, y=0$, στις θέσεις $x=100, y=0 \Rightarrow x=100, y=-100 \Rightarrow x=0, y=-100$ και τέλος θα επιστρέψουμε στο αρχικό σημείο $x=0, y=0$. Μελετήστε την εναλλαγή των εντολών και των αποτελεσμάτων τους.





Πριν δημιουργήσετε οποιοδήποτε σχήμα με τη χρήση εντολών της παλέτας **Πένα**,

α) σχεδιάστε σε ένα χαρτί το σύστημα συντεταγμένων της οθόνης του Scratch που είδαμε στο προηγούμενο κεφάλαιο,

β)) δημιουργήστε πάνω στο σύστημα συντεταγμένων το σχήμα που επιδιώκετε να σχεδιάσετε και προσδιορίστε τις συντεταγμένες των σημείων μεταξύ των οποίων πρέπει να κινηθούν τα αντικείμενα

γ) μεταφράστε το σχέδιό σας σε αντίστοιχες εντολές.

Δεν πρέπει να ξεχνάμε ότι πρώτα πρέπει να λύνουμε λογικά το πρόβλημά μας και στη συνέχεια να το υλοποιούμε στο περιβάλλον του Scratch.

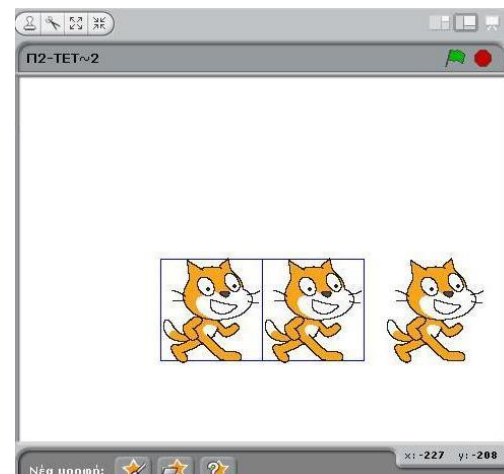
Όπως είπαμε και στα προηγούμενα κεφάλαια, στον προγραμματισμό υπάρχουν εναλλακτικοί τρόποι για να φτάσετε στο ζητούμενο αποτέλεσμα. Τα σενάρια που θα δημιουργείτε, δηλαδή ο συνδυασμός εντολών που θα επιλέγετε, δεν θα είναι μοναδικά και θα εξαρτώνται από την κατανόηση των διαθέσιμων εντολών, την προηγούμενη εμπειρία σας, και φυσικά τη φαντασία σας. Ας δούμε έναν εναλλακτικό τρόπο σχεδίασης του τετραγώνου.

Πώς θα ζωγραφίζατε το ίδιο τετράγωνο στο χαρτί; Το πιθανότερο είναι να ακουμπούσατε το μολύβι στην αρχική θέση και να χαράζατε μια οριζόντια γραμμή μήκους 100mm ως πούμε προς τα δεξιά. Στη συνέχεια θα τοποθετούσατε τον χάρακα κάθετα (90 μοίρες γωνία) και θα χαράζατε μια κατακόρυφη γραμμή μήκους 100mm προς τα κάτω. Μετά θα τοποθετούσατε τον χάρακα σε ορθή γωνία με την κατακόρυφη ακμή και θα χαράζατε μια οριζόντια γραμμή μήκους 100 mm προς τα αριστερά και τέλος θα επανερχόσασταν στο αρχικό σημείο με μια ευθεία γραμμή προς τα πάνω. Σκεπτόμενοι αντίστοιχα, θα μπορούσαμε να σχεδιάσουμε το τετράγωνο με το ακόλουθο σενάριο:

[06_π01.sb]



Δοκιμάστε τώρα από μόνοι σας να δημιουργήσετε ένα ομοίωμα του αντικειμένου σας μέσα στο τετράγωνο. Ποια εντολή της παλέτας **Πένα** θα χρησιμοποιήσετε; Χωράει το sprite σας μέσα στο τετράγωνο; Αν όχι κάντε τις απαραίτητες αλλαγές στο μέγεθος του τετραγώνου ώστε να χωρέσει. Αφού βρείτε τις κατάλληλες διαστάσεις του τετραγώνου βάλτε το αντικείμενό σας να κινηθεί ομαλά από τη μια άκρη της οθόνης στην άλλη, αφήνοντας το αποτύπωμα του σε δύο διαδοχικά τετράγωνα.



[06_π02.sb]

Ένα από τα εναλλακτικά σενάρια που λύνουν το συγκεκριμένο πρόβλημα παρουσιάζεται στην επόμενη εικόνα:



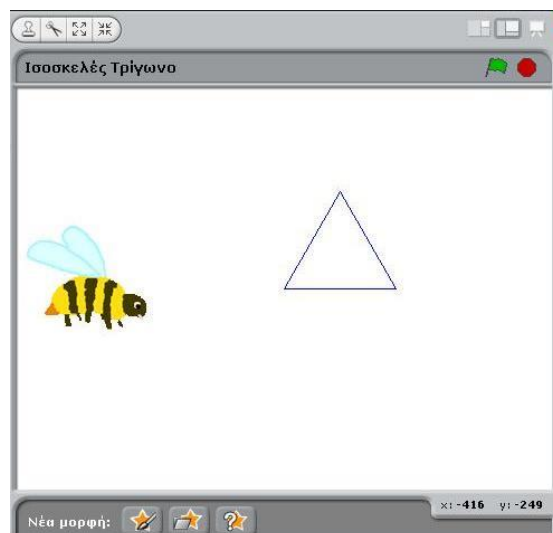
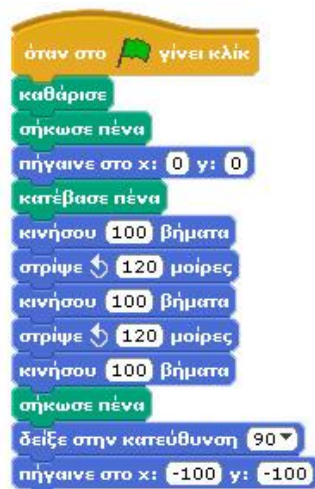
Μελετήστε πως η εντολή **σήκωσε πένα** στο παραπάνω σενάριο οριοθετεί τη δημιουργία των διαφορετικών σχημάτων. Μπορείτε να γράψετε ένα πιο σύντομο έργο που θα έχει τα ίδια αποτελέσματα;

Κύριος στόχος των προγραμματιστών είναι η δημιουργία έργων τα οποία χρησιμοποιούν όσο το δυνατόν λιγότερη επεξεργαστική ισχύ και τα οποία επιτρέπουν στον υπολογιστή μας να εκτελεί με ευκολία παράλληλες εργασίες. Τις περισσότερες φορές τα προγράμματα με τις λιγότερες εντολές είναι αυτά που είναι και πιο αποδοτικά. Συστήνουμε να επιδιώκετε τον περιορισμό των εντολών που χρησιμοποιείτε σε κάθε έργο σας.

Μη ξεχνάτε ότι μπορείτε να χρησιμοποιείτε την επιλογή «Ξεκίνησε απλό βηματισμό» για να παρακολουθείτε προσεκτικά τη χρησιμότητα κάθε εντολής που έχετε εισάγει στο σενάριό σας. Μέσω αυτής της επιλογής μπορείτε εύκολα να διακρίνεται πλεονασματικές εντολές...

Ας δοκιμάσουμε τώρα να δημιουργήσουμε ένα ισοσκελές τρίγωνο. Ισοσκελές ονομάζεται το τρίγωνο που έχει όλες τις πλευρές του ίσες και όλες τις γωνίες του ίσες.

Δεν είναι δύσκολο να φτιάξουμε ένα τέτοιο τρίγωνο όπως φαίνεται στο παρακάτω σενάριο.



[06_π02.sb]

Στο παραπάνω παράδειγμα χρησιμοποιήσαμε την εντολή **στρίψε ... μοίρες** με τιμή τις 120 μοίρες. Γιατί; Δεν θα έπρεπε να ορίσουμε 60 μοίρες; Ένα ισόπλευρο τρίγωνο δεν έχει γωνίες 60 μοιρών; Μήπως σχεδιάσαμε τις συμπληρωματικές γωνίες του τριγώνου (θυμηθείτε ότι $120 + 60 = 180$);

Πριν προχωρήσετε παρακάτω, σας συστήνουμε να σχεδιάσετε ένα μικρό καράβι, ένα σπιτάκι και έναν πύραυλο. Δύο χρήσιμες παρατηρήσεις:

A) Μη ξεχνάτε να σχεδιάζετε πρώτα πάνω σε ένα σύστημα συντεταγμένων στο χαρτί και στη συνέχεια να χρησιμοποιείτε τις εντολές του Scratch.

B) Χρησιμοποιήστε την εντολή **κινήσου ομαλά...δευτερόλεπτα στη θέση x...y...** προσδιορίζοντας επαρκείς χρόνους για να προλαβαίνετε να δείτε την εκτέλεση των εντολών του σεναρίου σας. Άλλωστε, έτσι γίνονται και πιο εντυπωσιακά τα έργα σας.

6.3 Χρώμα,σκιά και μέγεθος πέννας

6.3.1 Ορισμός του χρώματος

Δεν θα ήταν πιο ενδιαφέρον να μπορούμε να προσδιορίσουμε το χρώμα της πέννας κατά τη διάρκεια του έργου σας; Δεν είναι πιθανό να θέλετε να σχεδιάσετε ένα κόκκινο αυτοκίνητο και ένα πράσινο δέντρο; Το Scratch σας δίνει τη δυνατότητα αυτή με την εντολή **όρισε το χρώμα πέννας σε ...**

Πατώντας πάνω στο προεπιλεγμένο μπλε χρώμα, μπορούμε να αλλάξουμε το χρώμα της πέννας σε όποιο χρώμα επιθυμούμε.

Επομένως αν θέλαμε να δημιουργήσουμε μισή γραμμή μπλε και μισή πράσινη θα μπορούσαμε να εισάγουμε στο σενάριό μας την εξής ακολουθία εντολών:

```
κατέβασε πένα
όρισε το χρώμα πέννας σε [μπλε]
κινήσου 80 βήματα
όρισε το χρώμα πέννας σε [πράσινο]
κινήσου 80 βήματα
```



[06_π03.sb]

Μια άλλη δυνατότητα που υπάρχει για την αλλαγή του χρώματος της πέννας προκύπτει από την εντολή **άλλαξε χρώμα πέννας κατά** Σε αντίθεση με την προηγούμενη εντολή, στην εντολή **άλλαξε χρώμα πέννας κατά ...** προσδιορίζουμε πόσο θέλουμε να αλλάξουμε το χρώμα της πέννας σε σχέση με το ήδη επιλεγμένο. Για παράδειγμα, αν το επιλεγμένο χρώμα της πέννας είναι το μπλε, τότε αν αλλάξουμε το χρώμα της πέννας κατά +10 θα παρατηρήσουμε ότι η πένα θα πάρει γαλάζιο χρώμα ενώ αν ξανα-αλλάξουμε το χρώμα της πέννας κατά +10, το χρώμα της πέννας θα γίνει μωβ. Το χρώμα της πέννας δηλαδή αλλάζει βάσει μιας συνεχούς κλίμακας χρωμάτων.

Ας δούμε και οπτικά το αποτέλεσμα της **άλλαξε χρώμα πέννας κατά ...** Στις επόμενες εικόνες αλλάζουμε 300 φορές το χρώμα της πέννας και μετακινούμε κατά 1 pixel το αντικείμενο κάθε φορά. Με τον τρόπο αυτό «τυπώνεται» η κλίμακα χρωμάτων. Δεν είναι εντυπωσιακό το αποτέλεσμα;

```
όταν στο [πρόσκ] γίνει κλικ
πήγαινε στο x: -140 y: 0
δείξε στην κατεύθυνση 90
καθάρισε
κατέβασε πένα
όρισε το χρώμα πέννας σε [μπλε]
επανάλαβε 300
  κινήσου 1 βήματα
  άλλαξε χρώμα πέννας κατά 1
σηκώσε πένα
πήγαινε στο x: -140 y: -100
```



[06_π04.sb]

Παρότι δεν έχουμε συζητήσει ακόμη την εντολή **επανάλαβε ... φορές** μπορείτε εύκολα να κατανοήσετε ότι προκαλεί την επαναλαμβανόμενη εκτέλεση των εντολών που περιέχει στο εσωτερικό της. Δηλαδή οι εντολές που βρίσκονται μέσα στις «δαγκάνες» της εκτελούνται επαναλαμβανόμενα για όσες φορές προσδιορίζεται από τον αριθμό που βρίσκεται στο λευκό κουτάκι. Θα τη μελετήσουμε διεξοδικότερα παρακάτω.

Τι μέγεθος πιστεύετε ότι έχει η κλίμακα των χρωμάτων; Επαναλαμβάνονται ίδια χρώματα στην παραπάνω εικόνα; Μπορείτε μέσω πειραματισμού να βρείτε τον αριθμό των διαφορετικών χρωμάτων που μπορούν να τυπωθούν σε μια αντίστοιχη γραμμή;

6.3.2 Ορισμός της σκιάς

Τι μπορούμε να κάνουμε όμως αν θέλουμε να αλλάξουμε μόνο τη φωτεινότητα του χρώματος της πέννας χωρίς να αλλάξουμε το χρώμα; Δηλαδή πως μπορούμε απλά να κάνουμε πιο ανοιχτό ή πιο σκούρο το ήδη επιλεγμένο χρώμα της πέννας μας;

Μπορούμε να ορίσουμε τη **σκιά** της πέννας σύμφωνα με μία συγκεκριμένη τιμή (σκιά πέννας = 0 σημαίνει πολύ σκοτεινή, σκιά πέννας = 100 σημαίνει πολύ φωτεινή)

Η εντολή που μας δίνει αυτή την δυνατότητα είναι η **όρισε τη σκιά πέννας σε**

Ένα παράδειγμα για να την κατανοήσετε είναι το ακόλουθο:

```
όταν στο [πρόσκ] γίνει κλικ
πήγαινε στο x: 0 y: -100
καθάρισε
κατέβασε πένα
όρισε το μέγεθος πέννας σε 10
όρισε το χρώμα πέννας σε [μπλε]
όρισε τη σκιά πέννας σε 0
επανάλαβε 120
  κινήσου 2 βήματα
  όρισε τη σκιά πέννας σε [τυχαία επιλογή από 1 μέχρι 100]
σηκώσε πένα
```

[06_π05.sb]

Η εντολή **τυχαία επιλογή από 1 μέχρι 100** παράγει ένα τυχαίο αριθμό σκιάς από το 1 έως το 100 μετά από κάθε κίνηση 2 βημάτων. Για το λόγο αυτό βλέπετε στην αντίστοιχη εικόνα τυχαίες αλλαγές της σκιάς του μπλε.

Όπως, υπάρχει εντολή σχετικής αλλαγής του χρώματος (**άλλαξε χρώμα πέννας κατά X**), αντίστοιχα υπάρχει **άλλαξε σκιά πέννας κατά 1**

και εντολή σχετικής αλλαγής της σκιάς. Η εντολή άλλαξε τη σκιά πένας κατά μας επιτρέπει να προσδιορίσουμε πόσο πιο φωτεινό θέλουμε να είναι το ίχνος της σκιάς σε σχέση με το ήδη επιλεγμένο. Ένα παράδειγμα για να κατανοήσετε την παραπάνω εντολή είναι το ακόλουθο:



[06_π06.sb]

Σε αυτό το παράδειγμα βλέπετε την αλλαγή της φωτεινότητας του γαλάζιου, από σκιά 0, δηλαδή πολύ σκούρο γαλάζιο μέχρι σκιά 100 δηλαδή πολύ φωτεινό γαλάζιο.

6.3.3 Μέγεθος της πένας

Μέχρι τώρα όμως, δεν έχουμε αλλάξει το βασικό στοιχείο της πένας, το μέγεθός της.

Μπορούμε να ορίσουμε το πάχος της πένας με τη χρήση της εντολής **όρισε το μέγεθος πένας σε** Η

όρισε το μέγεθος πένας σε 1

μικρότερη τιμή που μπορούμε να δώσουμε είναι το 1 και αντιστοιχεί στη λεπτότερη πένα που μπορεί να υποστηρίξει το Scratch. Όσο μεγαλύτερη τιμή προσδιορίζουμε, τόσο πιο παχιά θα είναι η πένα μας. Το 1 λέτε να αντιστοιχεί σε εικονοστοιχείο;

Ας δούμε ένα παράδειγμα για την παραπάνω εντολή



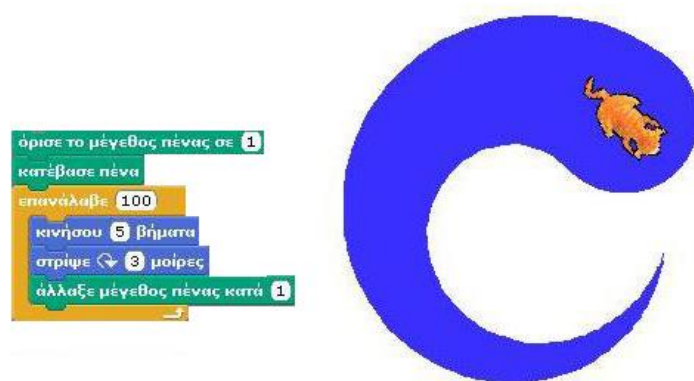
Τέλος, μπορούμε να αλλάζουμε το πάχος της πένας με τρόπο σχετικό, όπως ακριβώς μπορούμε να κάνουμε με το χρώμα και τη σκιά της πένας. Η αλλαγή προκύπτει

άλλαξε μέγεθος πένας κατά 1

βάσει της τρέχουσας τιμής του πάχους της πένας και αν είναι θετική η πένα γίνεται πιο παχιά ενώ αν είναι αρνητική η πένα

γίνεται πιο λεπτή. Η εντολή που μας δίνει αυτήν τη δυνατότητα είναι η άλλαξε το μέγεθος πένας κατά

Ένα εντυπωσιακό παράδειγμα για αυτήν την εντολή είναι το ακόλουθο:



[06_π07.sb]



Σε αυτό το παράδειγμα κινούμε 100 φορές το αντικείμενό μας στρίβοντάς το κάθε 5 βήματα κατά 3 μοίρες. Σε κάθε μια από τις 100 επαναλήψεις αυξάνουμε το μέγεθος της πένας κατά 1 με αποτέλεσμα να σχηματίζεται η εντυπωσιακή ουρά της εικόνας

6.4 Παράδειγμα

Ας συνδυάσουμε τις προηγούμενες εντολές για να δημιουργήσουμε ένα πιο σύνθετο παράδειγμα. Στόχος μας είναι να δημιουργήσουμε ένα πολύχρωμο σπίτι χρησιμοποιώντας περισσότερα από ένα αντικείμενα ταυτόχρονα.

Αρχικά αποφασίζουμε τον αριθμό των αντικειμένων που θέλουμε να χρησιμοποιήσουμε στο παράδειγμά μας και είτε τα επιλέγουμε από τη βιβλιοθήκη αντικειμένων του Scratch είτε δημιουργούμε δικά μας, όπως μάθαμε στο τέταρτο κεφάλαιο. Κατόπιν σκεφτόμαστε ποιο μέρος του τελικού σχεδίου θα αναλάβει το κάθε αντικείμενο π.χ. το πρώτο θα μπορούσε να αναλάβει τη σχεδίαση της σκεπής, το δεύτερο τα παράθυρα και το τρίτο το περίγραμμα του κτιρίου. Από τη στιγμή που όλα τα σενάρια μας θα ξεκινούν με την εντολή Όταν στην πράσινη σημαία γίνει κλικ, τα αντικείμενά μας θα ξεκινήσουν να δημιουργούν μαζί τα διαφορετικά τμήματα του σχεδίου μας. Πως όμως θα καταφέρουμε να ολοκληρώσουν και μαζί τη σχεδίαση; Μελετήστε τον παρακάτω κώδικα και προσπαθήστε να εξηγήσετε τις επιλογές του προγραμματιστή. Ποιο τμήμα του σχεδίου δημιουργεί το συγκεκριμένο αντικείμενο; Μπορείτε με αντίστοιχο τρόπο να δημιουργήσετε τα σενάρια των άλλων αντικειμένων για να δημιουργήσετε το σχήμα της εικόνας που ακολουθεί;



Επειδή τα αντικείμενά σας έχουν παραπλήσιο κώδικα, μπορείτε να χρησιμοποιήσετε την επιλογή «διπλασίασε» που εμφανίζεται με δεξί κλικ πάνω σε ένα σενάριο. Η επιλογή «διπλασίασε» δημιουργεί ένα αντίγραφο των εντολών που υπάρχουν στο σενάριο από το σημείο που πατήσαμε το κλικ και κάτω. Πατώντας στη συνέχεια πάνω σε οποιοδήποτε αντικείμενο οι συγκεκριμένες εντολές μεταφέρονται μέσα στο σενάριο του αντίστοιχου αντικείμενου. Για να αντιγράψετε όλο το σενάριο προφανώς πρέπει να πατήσετε δεξί κλικ πάνω στην εντολή **όταν στην πράσινη σημαία γίνει κλικ**

Περίληψη

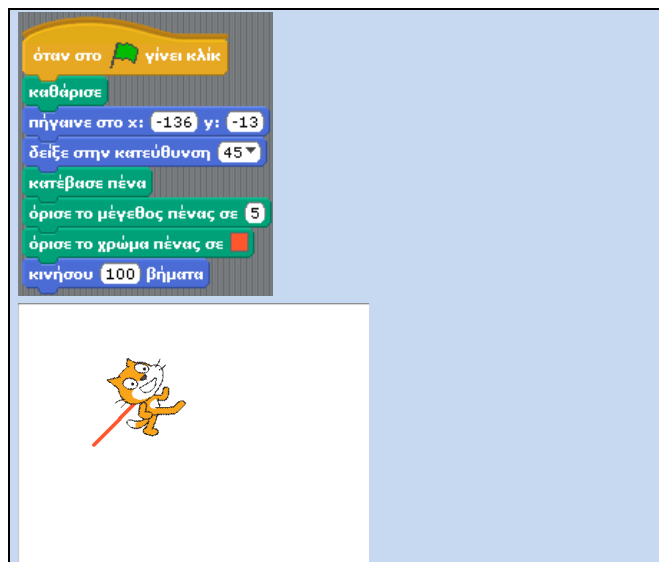
Οι εντολές της παλέτας **Πένα** είναι απλές στη χρήση τους αλλά και στην κατανόησή τους. Με μια απλή αναλογία με τον τρόπο που ζωγραφίζουμε στο χαρτί, μπορούμε εύκολα να αναγνωρίσουμε τις έννοιες της πέννας, του χρώματος, του πάχους, της σκιάς κτλ. Καλή χρήση των εντολών κίνησης και του συστήματος συντεταγμένων της οθόνης του Scratch είναι προαπαιτούμενα στοιχεία για τη δημιουργία σύνθετων σχημάτων. Οι δυνατότητες ζωγραφικής είναι πολλές και αξίζει να προσπαθήσετε να μάθετε περισσότερα για αυτές δημιουργώντας ποικίλα σχέδια, παίζοντας με τα χρώματα, τις σκιάς και τους διαφορετικούς χαρακτήρες-ζωγράφους.

Ερωτήσεις

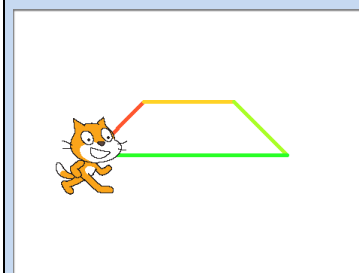
- 1) Ποια είναι η διαφορά ανάμεσα στο χρώμα μιας πέννας και στη σκιά της;
- 2) Γιατί προτείνετε η εντολή **καθάρισε** να χρησιμοποιείται στην αρχή κάθε έργου ζωγραφικής;
- 3) Γιατί προτείνετε η εντολή **σήκωσε πένα** να χρησιμοποιείται στο τέλος κάθε έργου ζωγραφικής;

Δραστηριότητες

- 1) Δίνεται ένα τμήμα ενός σεναρίου που υλοποιεί το αποτέλεσμα της εικόνας δ1 που ακολουθεί. Συμπληρώστε το σενάριο ώστε να επιτύχετε το αποτέλεσμα της εικόνας δ2.



Εικόνα δ1



Εικόνα δ2

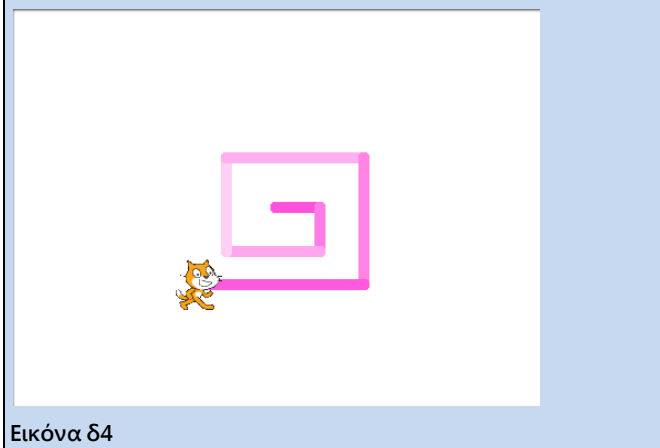
- 2) Μελετήστε το σενάριο που ακολουθεί και βρείτε την εντολή που λείπει ώστε να υλοποιηθεί το αποτέλεσμα της εικόνας δ3.





Εικόνα δ3

3) Αφού διορθώσετε το σενάριο της προηγούμενης δραστηριότητας, προσθέστε στο ίδιο έργο κατάλληλες εντολές έτσι ώστε να αλλάζει η σκιά της γραμμής όπως φαίνεται στην παρακάτω εικόνα δ4



Εικόνα δ4



Κεφάλαιο 7: Όψεις

Σε αυτό το κεφάλαιο:

7.1 Αλλάζοντας όψεις

7.2 Σκέψου πριν μιλήσεις

7.3 Πρώτο πλάνο και σωστό μέγεθος

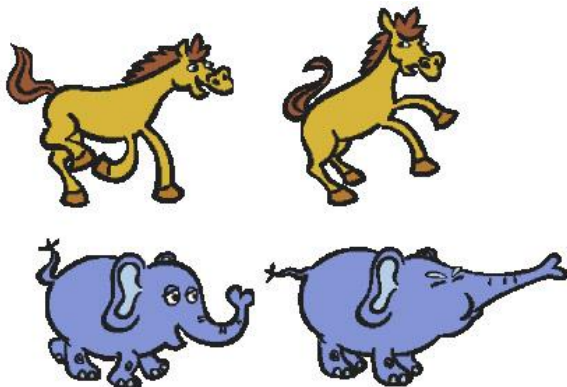
7.4 Δυναμικά εφέ

«Σημασία δεν έχει αυτό στο οποίο κοιτάς, αλλά αυτό που βλέπεις!»
(Henry David Thoreau)



7.1 Αλλάζοντας όψεις εύκολα, γρήγορα και προσεκτικά

Όπως και στα κόμικς, τα αντικείμενα που δημιουργούμε στο Scratch μπορούν να έχουν διαφορετικές ενδυμασίες, τις οποίες είναι σε θέση να «φορέσουν» σε διαφορετικές στιγμές εκτέλεσης των έργων μας. Στην παρακάτω εικόνα βλέπουμε δυο διαφορετικές ενδυμασίες για δυο αντικείμενα, ένα άλογο και έναν ελέφαντα!



Στο 4^ο κεφάλαιο είδαμε πολλούς τρόπους για να δημιουργούμε ενδυμασίες για κάθε αντικείμενο αλλά και διαφορετικά υπόβαθρα για το σκηνικό μας, χωρίς όμως να αξιοποιήσουμε τα στοιχεία αυτά στα έργα μας μέχρι τώρα. Στο αυτό το κεφάλαιο, θα συζητήσουμε για τις διαθέσιμες εντολές που αφορούν την επιλογή της επιθυμητής ενδυμασίας για τα αντικείμενα καθώς και γενικότερα για εντολές που μπορούν να επηρεάσουν την εμφάνιση του αντικειμένου.

Θα ξεκινήσουμε παρουσιάζοντας πρώτα τις διαθέσιμες εντολές της παλέτας **Όψεις** και στη συνέχεια θα τις αξιοποιήσουμε σε πρακτικά παραδείγματα. Μελετήστε λοιπόν από μόνοι σας τις εντολές του επόμενου πίνακα.

Εντολή	Επεξήγηση
αλλαγή σε ενδυμασία ενδυμασία2	Η εντολή μας επιτρέπει να επιλέξουμε την ενδυμασία που θέλουμε να «φορέσει» το αντικείμενό μας, προσδιορίζοντας απλά το όνομά της (μη ξεχνάτε

ότι κάθε ενδυμασία έχει το δικό της όνομα).

επόμενη ενδυμασία

Η εντολή αλλάζει την ενδυμασία του αντικειμένου με τη διαφορά όμως ότι δεν προσδιορίζουμε άμεσα μια όψη αλλά επιλέγεται αυτομάτως η επόμενη ενδυμασία στην αντίστοιχη λίστα ενδυμασιών κάθε αντικειμένου.

ενδυμασία #

Μια μεταβλητή κατάσταση (όπως η μεταβλητή κατάσταση **θέση x**) η οποία εμφανίζει στην οθόνη του έργου μας ποια ενδυμασία χρησιμοποιεί το αντικείμενό μας κάθε στιγμή.

πήγαινε σε πρώτο πλάνο

Με την εντολή αυτή προσδιορίζουμε ποιο αντικείμενό μας θέλουμε να είναι «μπροστά» από τα υπόλοιπα αντικείμενα του έργου μας (μέχρι βέβαια να ξανακάνουμε χρήση της εντολής για ένα άλλο αντικείμενο).

πήγαινε πίσω **1** επίπεδα

Με αυτή την εντολή μπορούμε να στείλουμε το αντικείμενό μας σε πιο πίσω επίπεδα ώστε να εμφανιστούν άλλα μπροστά του.

7.1.1 Ξεκινάμε να περπατάμε !!!

Παρότι μάθαμε να κινούμε ένα αντικείμενο, η κίνησή του μέχρι τώρα σήμαινε απλά την αλλαγή θέσης του στην οθόνη του Scratch. Δεν θα ήταν πιο αληθοφανές, καθώς ο μαθητής της παρακάτω εικόνας κινείται, να δημιουργείται η εντύπωση πως περπατάει; Γνωρίζετε ότι τα κινούμενα σχέδια που παρακολουθούμε στην τηλεόραση είναι μια σειρά από εικόνες οι οποίες αλλάζουν πολύ γρήγορα για να δημιουργηθεί η ψευδαίσθηση της κίνησης; Το Scratch μας δίνει τη δυνατότητα να κάνουμε κάτι αντίστοιχο.



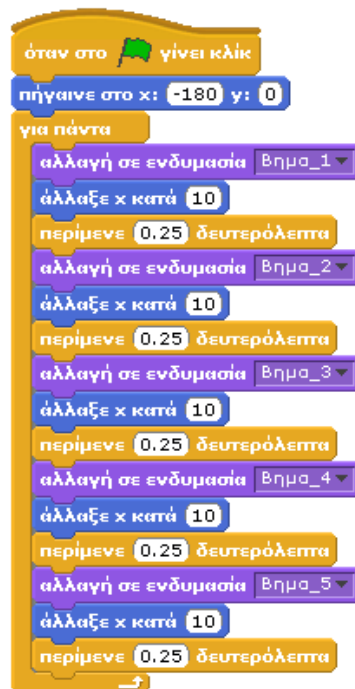
Απλά, θα πρέπει να δημιουργήσουμε διαφορετικές όψεις του μαθητή (διαφορετικές εικόνες) και με το συνδυασμό κατάλληλου χρονισμού και σωστής εναλλαγής ενδυμασιών να πετύχουμε το επιθυμητό αποτέλεσμα. Την πρώτη ενδυμασία του αντικειμένου την ονομάζουμε Βήμα_1, όπως φαίνεται στην επόμενη εικόνα και μπορείτε να τη βρείτε με το όνομα boy4-walking-a στη βιβλιοθήκη αντικειμένων, στην κατηγορία People. Εν συνεχεία, προσθέτουμε τις διαφορετικές όψεις boy4-walking-b, boy4-walking-c, boy4-walking-d και boy4-walking-e πατώντας το κουμπί «εισαγωγή» νέας ενδυμασίας, στην καρτέλα ενδυμασίες του αντικειμένου μας. Ονομάζουμε τις αντίστοιχες ενδυμασίες ως Βήμα_2, Βήμα_3, Βήμα_4, Βήμα_5.



Τώρα πρέπει να κάνουμε σωστό χρονισμό μεταξύ των εναλλαγών των ενδυμασιών μας και της κίνησης του αντικειμένου μας. Μελετήστε τις εντολές που εμφανίζονται παρακάτω.



[07_π01.sb]



Αρχικά, ορίζουμε στο σενάριο τη θέση από την οποία θέλουμε να ξεκινήσει να περπατάει το αντικείμενό μας. Στη συνέχεια επιλέγουμε την ενδυμασία Βήμα_1 με την εντολή αλλαγή σε ενδυμασία "Βήμα_1", ώστε να «φορέσουμε» στο χαρακτήρα μας την αρχική του ενδυμασία. Επιπλέον, μετακινούμε το χαρακτήρα μας στον άξονα X με τη βοήθεια της εντολής άλλαξε x κατά 10 και σταματούμε την εκτέλεση της επόμενης εντολής για 0,25 δευτερόλεπτα ώστε να δοθεί η εντύπωση του πρώτου βήματος. Η εντολή περίμενε X δευτερόλεπτα σταματά την εκτέλεση του σεναρίου για όσο χρονικό διάστημα προσδιορίζεται μέσα στο λευκό κουτάκι. Ουσιαστικά, δημιουργήσαμε ένα νέο «καρέ» και το αφήσαμε να εμφανίζεται για 0.25 δευτερόλεπτα. Αμέσως μετά το αντικείμενό μας φοράει την ενδυμασία Βήμα_2, μετακινείται για άλλα 10 σημεία και περιμένει για 0.25 δευτερόλεπτα. Έτσι αρχίζει να δημιουργείται η εντύπωση της κίνησης παρουσιάζοντας καρέ των 0.25 δευτερολέπτων. Στις εντολές έχει προστεθεί η εντολή για πάντα ώστε η ακολουθία αλλαγής όλων των ενδυμασιών και ταυτόχρονης μετακίνησης του αντικειμένου να επαναλαμβάνεται διαρκώς. Αν εκτελούσαμε τη συγκεκριμένη ακολουθία εντολών μόνο μια φορά, πόσο διάστημα θα διαρκούσε η κίνησή;

Σκεφτείτε πως θα μπορούσατε να δημιουργήσετε το ίδιο έργο με τη χρήση της εντολής κινήσου ομαλά για X δευτερόλεπτα στη θέση X και Y;

Ποια από τις δυο εκδοχές του έργου προτιμάτε; Ποιο έργο υλοποιείται με τις λιγότερες εντολές; Ποιο έργο παράγει το καλύτερο οπτικό αποτέλεσμα;

7.1.2 Χόρεψε με τη "σειρά"

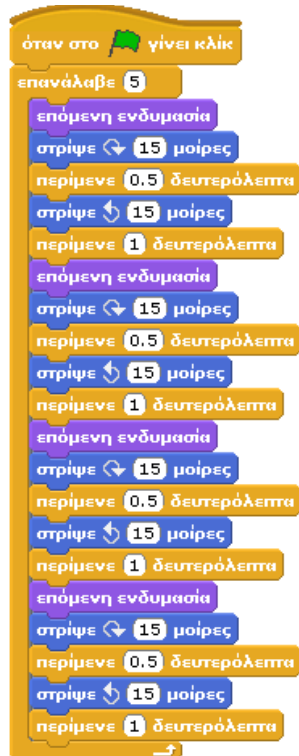
Μπορούμε στο προηγούμενο έργο μας να αντικαταστήσουμε όλες τις εντολές αλλαγή σε ενδυμασία... με την εντολή επόμενη ενδυμασία; Αν παρατηρήσουμε προσεκτικά το σενάριο, η σειρά επιλογής των ενδυμασιών είναι αντίστοιχη με τη σειρά των ενδυμασιών στην αντίστοιχη λίστα. Συνεπώς, δοκιμάστε να τρέξετε το πρώτο μας παράδειγμα αντικαθιστώντας την εντολή

αλλαγή σε ενδυμασία... με την εντολή επόμενη ενδυμασία.
Μήπως μπορείτε να κάνετε το έργο πολύ μικρότερο;

Ας δούμε άλλο ένα παράδειγμα κάνοντας χρήση της εντολής επόμενη ενδυμασία, αλλά και της μεταβλητής κατάστασης ενδυμασία για να βλέπουμε ποια ενδυμασία του αντικειμένου μας είναι επιλεγμένη κάθε στιγμή.

Μελετήστε το παρακάτω σενάριο που αφορά το κορίτσι της επόμενης εικόνας:

[07_π02.sb]



Το έργο μας περιλαμβάνει ως σκηνικό την εικόνα train-tracks-1 από τη βιβλιοθήκη σκηνικών (κατάλογος outdoor) και ως αντικείμενο την Cassy από τη βιβλιοθήκη αντικειμένων (κατάλογος People). Επιπλέον έχουμε εισάγει ως ενδυμασίες τις cassy-dancing-1, cassy-dancing-2, cassy-dancing-3, cassy-dancing-4 από τον ίδιο κατάλογο.

Ποιο είναι το αποτέλεσμα του συγκεκριμένου έργου; Θα μπορούσε το πρόγραμμα να παράγει τα ίδια αποτελέσματα με λιγότερες εντολές; Βρείτε τον τρόπο.

Για πρώτη φορά συνειδητοποιούμε ότι η σειρά με την οποία εισάγουμε τις ενδυμασίες στην αντίστοιχη λίστα μπορεί να έχει σημαντική επίδραση στην εκτέλεση του προγράμματός μας. Είναι εύκολο να αλλάξουμε αυτή τη σειρά κάνοντας κλικ στην εικόνα κάθε ενδυμασίας και σέρνοντας την στη θέση που επιθυμούμε.



7.1.3 Είμαι πιο "Μπροστά" απ' όλους... με τη βοήθεια εντολών....

Έστω, ότι προσπαθούμε να αναπαράγουμε στο Scratch μια ιστορία από το γνωστό σε όλους μας κόμικ «Αστερίξ και Οβελίξ». Προκύπτει ένα ερώτημα που δεν έχει απαντηθεί έως τώρα: πως μπορούμε να τοποθετήσουμε ένα συγκεκριμένο αντικείμενο μπροστά από ένα άλλο;



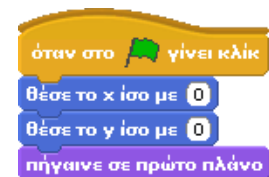
Αν θεωρήσουμε ως αντικείμενα του συγκεκριμένου έργου το χαλί και τους ήρωες που βρίσκονται πάνω στο χαλί, τότε μπορούμε να πούμε ότι σε πρώτο πλάνο βρίσκεται το ιπτάμενο χαλί καθώς η ενδυμασία του υπερκαλύπτει όλα τα υπόλοιπα αντικείμενα (π.χ. τα πόδια του Αστερίξ και του Οβελίξ). Επιπλέον μπορούμε να διακρίνουμε πως ο Οβελίξ είναι ένα επίπεδο πίσω από τον Ιντεφιξ, το σκύλο του, καθώς βρίσκεται πίσω από αυτόν αλλά και τον Αστερίξ καθώς διακρίνουμε ότι τα πόδια του Αστερίξ είναι μπροστά από το χέρι του Οβελίξ.

Η λογική των επιπέδων υπάρχει και στο Scratch και μας παρέχονται δυο εντολές για να προσδιορίζουμε κατά την εκτέλεση του έργου μας το επίπεδο στο οποίο εμφανίζεται κάθε αντικείμενο: η πήγαινε σε πρώτο πλάνο και η πήγαινε πίσω...επίπεδα. Παρατηρήστε ότι κάθε νέο αντικείμενο που εισάγουμε στο έργο μας τοποθετείται αυτομάτως σε πρώτο πλάνο. Επίσης, το αντικείμενο που χειριστήκαμε τελευταίο με το ποντίκι βρίσκεται και αυτό σε πρώτο πλάνο. Για να πειστείτε: α) εισάγετε διαδοχικά τρία αντικείμενα στο Scratch και θα παρατηρήσετε ότι το τελευταίο που εισάγατε βρίσκεται πάνω από τα προηγούμενα, β) μετακινήστε το πρώτο ή το δεύτερο αντικείμενο που εισάγατε ελάχιστα με το ποντίκι σας και θα παρατηρήσετε ότι αυτομάτως

πήγαινε σε πρώτο πλάνο

βρίσκεται μπροστά από τα υπόλοιπα.

Μπορούμε να δημιουργήσουμε ένα απλό έργο στο οποίο θα περιέχονται 3 αντικείμενα και τα οποία με τη χρήση της εντολής πήγαινε σε πρώτο πλάνο να βρεθούν και τα τρία σε πρώτο πλάνο; Έστω ότι δημιουργούμε το ίδιο σενάριο και για τα τρία αντικείμενα:



Προσδιορίζουμε να βρίσκονται στην ίδια θέση ($x=0$, $y=0$) και δίνουμε εντολή και στα 3 αντικείμενα να είναι σε πρώτο πλάνο.



Αν εκτελούμε πολλές φορές το έργο μας θα παρατηρήσουμε ότι οι εντολές μας δεν μας οδηγούν σε ένα σταθερό αποτέλεσμα, αλλά μπορεί να βλέπουμε:



Δεν είναι δυνατόν να βρίσκονται περισσότερα από ένα αντικείμενα σε πρώτο πλάνο αφού πάντα ένα αντικείμενο θα πρέπει να είναι πάντα μπροστά από κάποιο άλλο. Το έργο μας δεν λειτουργεί όπως θα θέλαμε και για το λόγο αυτό θα κάνουμε χρήση της εντολής πήγαινε πίσω...επίπεδα. Αν επιδιώκαμε το αποτέλεσμα της πρώτης από τις δυο εικόνες τότε τα σενάρια των τριών αντικειμένων θα έπρεπε να ήταν αντίστοιχα:



[07_π03.sb]



Κάθε φορά που θα εκτελείτε το συγκεκριμένο έργο, το αποτέλεσμα είναι το ίδιο; Ο αρχάριος προγραμματιστής μας έχει κάνει πάλι κάποιο λάθος! Πως μπορείτε να βελτιώσετε τα παραπάνω σενάρια χρησιμοποιώντας και την εντολή πήγαινε σε πρώτο πλάνο;

7.2 - Σκέψου πριν μιλήσεις...

Πέρα από την εμφάνιση, οι χαρακτήρες-αντικείμενα χρειάζονται κι άλλα μέσα έκφρασης για να μπορέσουμε να αναπαράγουμε ιστορίες. Σίγουρα έχετε δει στα κόμικς τα μπαλονάκια διαλόγου που εμφανίζονται όταν δυο χαρακτήρες συζητούν και τα συννεφάκια οποία χρησιμοποιούνται για την περίπτωση που ο χαρακτήρας σκέφτεται κάτι από μέσα του.



Για να κάνουμε στο Scratch ένα αντικείμενο να πει κάτι, υπάρχει η εντολή πες.... η οποία εμφανίζει ένα μπαλονάκι κειμένου πάνω από τον χαρακτήρα μας. Το κείμενο που εμφανίζεται μέσα στο μπαλονάκι το καθορίζουμε εμείς, αλλάζοντας την προεπιλεγμένη έκφραση «Γεια σου!».

πες Γεια σου!

σκέψου το Χμμ...

Η εντολή σκέψου το..., εμφανίζει ένα συννεφάκι πάνω ακριβώς από τον χαρακτήρα-αντικείμενο. Το κείμενο που περιέχει το συννεφάκι είναι αυτό που συμπληρώνουμε στο λευκό κουτάκι της εντολής που έχει ως προεπιλογή την λέξη: «Χμμμ».

Σε αντίθεση με τα κόμικς, στο Scratch οι ιστορίες είναι δυναμικές, οι χαρακτήρες συζητούν μεταξύ τους, μετακινούνται και αλλάζουν συμπεριφορά κατά τη διάρκεια του έργου. Πολλές φορές συνεπώς δεν μας ενδιαφέρει απλώς το τι θα πουν ή θα σκεφτούν οι χαρακτήρες μας αλλά για πόση ώρα θα εμφανίζονται τα αντίστοιχα μπαλονάκια ώστε να γίνει ο κατάλληλος συγχρονισμός μεταξύ τους. Για το λόγο αυτό το Scratch μας παρέχει δυο παρόμοιες εντολές που μας επιτρέπουν να προσδιορίσουμε το για πόσο χρόνο θα φαίνεται το συννεφάκι/μπαλονάκι. Αυτές οι εντολές είναι οι σκέψου το...για... δευτερόλε-

πτα και πες...για...δευτερόλεπτα. Το μόνο που έχουμε να κάνουμε επιπλέον, είναι να συμπληρώσουμε το πεδίο των δευτερολέπτων εμφάνισης για το συννεφάκι/μπαλονάκι.

Είναι σημαντικό να τονίσουμε ότι κατά τη διάρκεια των δευτερολέπτων που κάποιο αντικείμενο μιλάει ή σκέφτεται, το έργο μας δεν προχωράει στις εντολές που ακολουθούν. Ουσιαστικά, οι εντολές αυτές μπορούν να αντικατασταθούν από τις απλές εκδοχές τους (που δεν περιέχουν χρόνο) ακολουθούμενες από την εντολή περίμενε...δευτερόλεπτα. Αποτελούν δηλαδή συνδυασμό των δυο απλών εντολών, με μια όμως διαφορά! Ποια είναι αυτή;

7.2.1 Παράδειγμα : Ενοχλητικός τύπος

Έστω ότι θέλουμε να σκηνοθετήσουμε την πρώτη γνωριμία δυο συμμαθητών όπου ο πρώτος μας ήρωας (ας τον βαφτίσουμε Σορτσάκια) κάνει το πρώτο βήμα ανοίγοντας κουβέντα για το Scratch, αλλά ο δεύτερος (ο Μπούκλας) δεν έχει την ίδια διάθεση για γνωριμίες κι έτσι τον αποπαίρνει. Πιο αναλυτικά ο διάλογός τους:

- Σορτσάκιας λέει: **Γεια! Ξέρεις scratch;**
- Μπούκλας σκέφτεται: **Την παλεύει αυτός; Όρεξη για κουβέντα έχει;**
- Μπούκλας λέει: **Φίλε σπάσε, ενοχλείς!**
- Σορτσάκιας σκέφτεται: **Τι αγενής...**

Μελετώντας το διάλογο, προκύπτει ότι θα χρειαστούμε για τον κάθε ήρωα μια εντολή πες... και μία σκέψου το.... Από τη στιγμή όμως που έχουμε διάλογο και το κάθε μπαλονάκι πρέπει να εμφανιστεί την κατάλληλη στιγμή θα χρησιμοποιήσουμε την εκδοχή των εντολών που περιέχει και το χρόνο παρουσίας.

Ας διαλέξουμε τις μορφές boycurly και boyshorts που βρίσκονται στο φάκελο costumes/people, αφού πρώτα έχουμε σβήσει το αντικείμενο γάτα με δεξί κλικ και επιλέγοντας «διαγραφή». Κατόπιν αλλάζουμε τα ονόματα των αντικειμένων σε «Μπούκλας» για την μορφή boycurly και «Σορτσάκιας» για την μορφή boyshorts. Ως φόντο θα ορίσουμε το υπόβαθρο hall που βρίσκεται στον φάκελο backgrounds/Indoors.



Από την καρτέλα Έλεγχος σύρουμε στην καρτέλα "Σενάρια" την εντολή όταν στο κουμπί πράσινη σημαία γίνει κλικ για το αντικείμενο Σορτσάκιας. Κατόπιν, ενώνουμε την προηγούμενη εντολή με τις εντολές θέσε το x ίσο με... και θέσε το y ίσο με..., βάζοντας τις τιμές 170 και -50 για x και y αντίστοιχα. Επειδή θέλουμε να αφήσουμε χώρο μεταξύ των 2 ηρώων μας ώστε να εμφανιστούν οι διάλογοι, για το αντικείμενο Μπούκλας εισάγουμε τις ίδιες εντολές μόνο που για x και y βάζουμε τις τιμές -

170 και -50 αυτή τη φορά. Από το σημείο αυτό πρέπει να δημιουργήσουμε τον διάλογο, οπότε χρειάζονται και διαφορετικές εντολές για κάθε αντικείμενο.

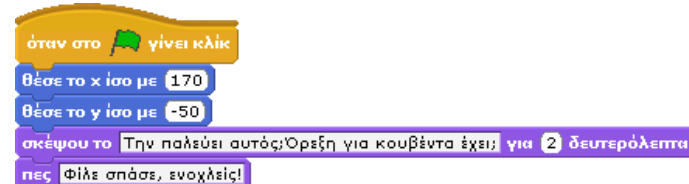


Για το αντικείμενο Σορτσάκιας, ο οποίος θα είναι ο ευγενικός μας ήρωας, σύρουμε από την καρτέλα Όψεις την εντολή πες...για...δευτερόλεπτα και συμπληρώνουμε τα δυο της πεδία με το κείμενο "Γεια! Ξέρεις scratch;" και την τιμή 3 δευτερόλεπτα. Ύστερα, από την ίδια καρτέλα σέρνουμε την εντολή σκέψου το... και συμπληρώνουμε το πεδίο της με το κείμενο "Τι αγενής..."



Ο Σορτσάκιας δηλαδή, πηγαίνει στην αρχική του θέση, κάνει την ερώτησή του που διαρκεί για 3 δευτερόλεπτα και μετά σκέφτεται «Τι αγενής».

Για το αντικείμενο Μπούκλας, ο οποίος θα είναι ο ήρωας που δεν σηκώνει και πολλά, σύρουμε από την καρτέλα Όψεις την εντολή σκέψου το... για ...δευτερόλεπτα, συμπληρώνοντας τα πεδία της με το κείμενο "Την παλεύει αυτός; Όρεξη για κουβέντα έχει;". Πόσο χρόνο όμως θα διαρκέσει η συγκεκριμένη σκέψη του από τη στιγμή που στα 3 δευτερόλεπτα ο Σορτσάκιας σκέφτεται «Τι αγενής...»; Προφανώς λιγότερο από 3 δευτερόλεπτα (π.χ. 2) ώστε να προλάβει να πει το κομμάτι του διαλόγου που του αντιστοιχεί και να ακολουθήσει η σκέψη του Σορτσάκια. Στη συνέχεια, προσθέτουμε την εντολή πες... γεμίζοντας το κουτί κειμένου με το «Φίλε σπάσε, ενοχλείς!».



[07_π04.sb]

Έρθε η ώρα να τρέξουμε το πρόγραμμά μας και να δούμε τον διάλογο που σχεδιάσαμε! Μπορείτε να κάνετε δοκιμές με τους αντίστοιχους χρόνους για να ανακαλύψετε διαφορετικούς τρόπους με τους οποίους μπορείτε να χτίσετε τους διαλόγους σας.

7.2.2 Παράδειγμα : Scratch Rules

Ας δοκιμάσουμε τις νέες εντολές σε ένα πιο μεγάλο έργο. Στο παράδειγμα αυτό, θα εξελίξουμε το διάλογο των 2 ηρώων βάζοντάς τους να πουν περισσότερα και με καλύτερο συγχρονισμό. Σ' αυτό το επεισόδιο, ο Μπούκλας μετανιωμένος για την προηγούμενη συμπεριφορά του προς τον Σορτσάκια, τον πλη-



σιάζει στο δωμάτιο διασκέδασης για να επανορθώσει, καλώντας τον να πουν μαζί ένα σύνθημα! Ακολουθεί το αναλυτικό σενάριο για το σκετς:

ΣΕΝΑΡΙΟ		
Χρό- νος (δευτ)	Μπούκλας	Σορτσάκιας
0	Σκέφτεται : Ας σπάσω τον πάγο...	Σκέφτεται : Ωχ! Λες να μ' αρχίσει στα μπινελίκια τώρα ο Νεάντερνταλ;
2	Λέει: Γεια σου!	
3		Λέει: Γεια σου!
3.5	Λέει: Έτοιμος;	
5	Λέει: Με το 1...	Λέει: Οκ!
6	Λέει: με το 2...	
7	Λέει: με το 3...	
8	Λέει: Scratch rules!!!	Λέει: Scratch rules!!!

Όπως καταλαβαίνετε, η συγκεκριμένη μορφή παρουσίασης του διαλόγου είναι απαραίτητη για να γίνουν συγχρονισμοί διαλόγων που διαρκούν περισσότερο από 2-3 δευτερόλεπτα. Αποτυπώνοντας με αυτόν τον τρόπο το διάλογο, διευκολύνουμε κατά πολύ τη δημιουργία των σεναρίων για κάθε αντικείμενο και σχεδιάζουμε πολύ ευκολότερα ολόκληρο το διάλογο. Καλό θα ήταν να αξιοποιείτε και εσείς τη συγκεκριμένη τεχνική. Πλέον οι εντολές που πρέπει να προστεθούν σε κάθε αντικείμενο είναι πολύ εύκολο να ανιχνευτούν:

όταν στο γίνει κλικ
θέσε το x ίσο με -170
θέσε το y ίσο με -50
σκέψου το Ας σπάσω τον πάγο... για 2 δευτερόλεπτα
πες Γεια σου! για 1.5 δευτερόλεπτα
πες Έτοιμος; για 1.5 δευτερόλεπτα
πες Με το 1... για 1 δευτερόλεπτα
πες με το 2... για 1 δευτερόλεπτα
πες με το 3... για 1 δευτερόλεπτα
πες Scratch rules!!! για 2 δευτερόλεπτα

όταν στο γίνει κλικ
θέσε το x ίσο με 170
θέσε το y ίσο με -50
σκέψου το Ωχ! Λες να μ' αρχίσει στα μπινελίκια τώρα ο νεάντερνταλ; για 3 δευτερόλεπτα
πες Γεια σου! για 1 δευτερόλεπτα
πες Οκ! για 4 δευτερόλεπτα
πες Scratch rules!!! για 2 δευτερόλεπτα

[07_π05.sb]

Δεν ήταν και τόσο δύσκολο τελικά!

7.2.3 Παράδειγμα : Δοκιμαστήριο

Όρα να τεστάρουμε αυτά που μάθαμε σ' αυτό το κεφάλαιο με ένα πιο μεγάλο παράδειγμα. Οι δύο ήρωες μας βρίσκονται αυ-

τήν την φορά σε ένα κατάστημα ρούχων, προκειμένου να αγοράσουν παραδοσιακές στολές διάφορων χωρών, για το ταξίδι που σχεδιάζουν. Το σενάριο:

ΣΕΝΑΡΙΟ		
Χρό- νος (δευτ)	Μπούκλας	Σορτσάκιας
0	Βρίσκεται πίσω από το παραβάν ντυμένος με την πρώτη ενδυμασία και λέει: « Έτοιμος;»	Βρίσκεται μακριά από το παραβάν περιμένοντας και λέει: « Για να δω...»
1	Βγαίνει από το παραβάν	
2		Λέει: « Χμμμ..»
2,5	Λέει: «Εεεε; Απευθείας απ'τα Χάιλαντς!»	
3,5		Λέει: «Μπα!»
4,5	Λέει: «Οκ, αλλάζω.»	
5,5	Πηγαίνει πίσω από το παραβάν κι επιλέγει την επόμενη ενδυμασία	
7	Βγαίνει από το παραβάν	
8,5	Λέει: «Ολέ!!! Ματαντόρ. Ε;»	Λέει: «Χμμμ..»
10		Λέει: «Μπα!»
10,5	Λέει: «Οκ, αλλάζω.»	
11,5	Πηγαίνει πίσω από το παραβάν κι επιλέγει την επόμενη ενδυμασία	
13	Βγαίνει από το παραβάν	
14		Λέει: «Μμμ..»
14,5	Λέει: «Τελευταίο, τσανιζί!»	
15,5		Λέει: «Μπα! Αυτά ήταν όλα;»
17,5	Λέει: «Λοιπόν, τα παίρνουμε όλα και διαλέγουμε μετά»	
18		
19,5	Πηγαίνει πίσω από το παραβάν	Λέει: «ΟΚ!»

Διαλέγουμε ως υπόβαθρο για το έργο μας το clothing-store backgrounds που βρίσκεται στον κατάλογο Indoors. Επιπλέον, στο αντικείμενο Μπούκλας θα προσθέσουμε δυο νέες ενδυμασίες, τις boycurlg2 και boycurlg3. Τις ενδυμασίες αυτές μπορείτε να τις δημιουργήσετε πατώντας τρεις φορές αντιγραφή στην

αρχική ενδυμασία του Μπούκλα και στη συνέχεια χρησιμο-
ποιώντας τον επεξεργαστή ζωγραφικής για να δημιουργήσετε
ενδυμασίες παρόμοιες με αυτές των εικόνων που ακολουθούν.



Επιπλέον θα προσθέσουμε ένα αντικείμενο παραβάν πίσω από
το οποίο ο χαρακτήρας μας θα αλλάζει τα ρούχα του! (ψάξτε
στο διαδίκτυο για curtain δηλαδή κουρτίνα)



Θέλουμε ο Μπούκλας να δοκιμάζει στολές και ο Σορτσάκις να
αποκρίνεται αν του αρέσουν ή όχι. Έτσι, πρέπει αρχικά να το-
ποθετήσουμε το αντικείμενο παραβάν σε σημείο που να επιτρέ-
πει στον ήρωα που δοκιμάζει τις στολές να βγαίνει από αυτό και
τα μπαλονάκια των δυο ηρώων να μην επικαλύπτονται κατά
τον διάλογο.

Επειδή θέλουμε ο Μπούκλας να μπορεί να πηγαίνει πίσω από
το παραβάν, εισάγουμε στο σενάριό του την εντολή πήγαινε
πίσω 1 επίπεδα. Από το σημείο και μετά απλά προσθέτουμε τις
απαιτούμενες εντολές όπως φαίνονται στο επόμενο σενάριο.
Πρέπει να σημειώσουμε ότι από τη στιγμή που χρησιμοποιούμε
την εντολή επόμενη ενδυμασία, όπως είπαμε και προηγουμέ-
νως, θα πρέπει να προσέξουμε τη σειρά των ενδυμασιών στην
αντίστοιχη λίστα του Μπούκλα.

```

όταν στο  γίνει κλικ
  πήγαινε πίσω 1 επίπεδα
  θέσε το x ίσο με 180
  θέσε το y ίσο με -10
  αλλαγή σε ενδυμασία boycurly1
  πες 'Ετοιμος; για 1 δευτερόλεπτα
  κινήσου ομαλά 1.5 δεύτ. στο x: 0 y: -20
  πες 'Εεε, Απευθείας απ'τα Χάιλαντς! για 2 δευτερόλεπτα
  πες 'Οκ, αλλάζω. για 1 δευτερόλεπτα
  κινήσου ομαλά 1.5 δεύτ. στο x: 180 y: -20
  επόμενη ενδυμασία
  κινήσου ομαλά 1.5 δεύτ. στο x: 0 y: -20
  πες 'Ολέ!!!Ματαντόρ. Ε; για 2 δευτερόλεπτα
  πες 'Οκ, αλλάζω. για 1 δευτερόλεπτα
  κινήσου ομαλά 1.5 δεύτ. στο x: 180 y: -20
  επόμενη ενδυμασία
  κινήσου ομαλά 1.5 δεύτ. στο x: 0 y: -20
  πες 'Τελευταίο, τσαινίζ! για 3 δευτερόλεπτα
  πες 'Λοιπόν, τα παίρνουμε όλα και διαλέγουμε μετά, για 2 δευτερόλεπτα
  κινήσου ομαλά 1.5 δεύτ. στο x: 180 y: -20

```

Όσον αφορά το Σορτσάκια, τα πράγματα είναι πιο απλά, αφού
το μόνο που κάνει είναι να εκφράζει την άποψή του για τις στο-
λές που δοκιμάζει ο φίλος του:

```

όταν στο  γίνει κλικ
  θέσε το x ίσο με -170
  θέσε το y ίσο με -75
  πες 'Για να δω... για 2 δευτερόλεπτα
  σκέψου το Μμμ... για 1.5 δευτερόλεπτα
  πες 'Μπα! για 5 δευτερόλεπτα
  σκέψου το Μμμ... για 1.5 δευτερόλεπτα
  πες 'Μπα! για 5 δευτερόλεπτα
  σκέψου το Μμμ... για 1.5 δευτερόλεπτα
  πες 'Μπα! Αυτά ήταν όλα; για 5 δευτερόλεπτα
  πες 'ΟΚ!

```

[07_π05.sb]



Τονίζουμε πως ο συγχρονισμός επετεύχθη όπως και στα προη-
γούμενα παραδείγματα, αφαιρώντας κάθε φορά στο χρονο-
διάγραμμα του σεναρίου την χρονική στιγμή της επόμενης α-
τάκας από την χρονική στιγμή της τρέχουσας. Το αποτέλεσμα
της αφαίρεσης αντιστοιχούσε στο χρόνο που πρέπει να διαρκεί
η τρέχουσα ατάκα.

7.3 Πρώτο πλάνο και στο σωστό μέγεθος

Έχουμε ήδη ντύσει και έχουμε μάθει τους ήρωες μας να μιλούν
και να σκέφτονται! Ήρθε λοιπόν η στιγμή, να αυξομειώσουμε
το μέγεθός τους!

Θα έχετε σίγουρα παρατηρήσει στα κόμικ, ότι για να δείξει ο
σκιτσογράφος πόσο κοντά ή πόσο μακριά είναι κάποιος χαρα-
κτήρας, αλλάζει το μέγεθός του. Έτσι, όταν ένας χαρακτήρας εί-
ναι πιο κοντά μας, θα πρέπει να φαίνεται πιο μεγάλος, ενώ όταν
απομακρύνεται θα πρέπει να αποκτά μικρότερο μέγεθος. Το

Scratch, όπως ήδη θα φαντάζεστε, μας δίνει την δυνατότητα να αυξήσουμε ή να μειώσουμε το μέγεθος ενός αντικειμένου!

Ας δούμε τις εντολές που μας λύνουν τα χέρια. Με την εντολή άλλαξε μέγεθος κατά..., αλλάζουμε το μέγεθος κατά ένα ποσό! Μπορούμε να μειώσουμε το μέγεθος ενός χαρακτήρα, αν αντί για θετική τιμή, βάλουμε αρνητική. Δηλαδή, μπορούσαμε να βάλουμε τιμή -10 και να δούμε τον χαρακτήρα μας να μικραίνει! Παίξτε με τις τιμές για να εξοικειωθείτε.

άλλαξε μέγεθος κατά 10

Με την εντολή όρισε το μέγεθος σε...%, ορίζουμε το μέγεθος ενός χαρακτήρα μας, σε σχέση με το αρχικό του μέγεθος. Για παράδειγμα, αν βάλουμε τιμή 50 (όρισε μέγεθος σε 50%) τότε ο χαρακτήρας μας θα γίνει μισός σε μέγεθος. Μπορούμε να βάλουμε τιμές πάνω από το 100% (π.χ. με 200% γίνεται διπλάσιος). Η διαφορά της συγκεκριμένης εντολής με την προηγούμενη έγκειται στο γεγονός ότι όσες φορές και αν την επαναλάβουμε το μέγεθος του αντικειμένου θα παραμείνει το ίδιο, ενώ η εντολή άλλαξε μέγεθος κατά... θα αλλάζει το μέγεθος του αντικειμένου κάθε φορά που εκτελείται.

Επιπλέον, το Scratch μας δίνει 2 εντολές με τις οποίες μπορούμε να εμφανίσουμε ή να εξαφανίσουμε το χαρακτήρα μας. Αυτές είναι οι: εμφάνισε και η απόκρυψη. Ο ρόλος τους είναι προφανής! Με την εντολή εμφάνισε, εμφανίζεται ο χαρακτήρας μας στο σκηνικό, ενώ με την εντολή απόκρυψη, ο χαρακτήρας εξαφανίζεται από την οθόνη μας!

εμφάνισε

απόκρυψη

Ας δούμε την χρήση των εντολών που μόλις αναλύσαμε, μέσα από ένα παράδειγμα. Τι θα λέγατε για να χρησιμοποιήσουμε ως αντικείμενα τους Αστερίξ και Οβελίξ;

7.3.1 Παράδειγμα : Αστερίξ

Αφού έχουμε κατεβάσει και αποθηκεύσει τα σκίτσα που θα χρειαστούμε για το κόμικ μας (αναζητώντας στο διαδίκτυο...), ανοίγουμε το Scratch. Στο παράδειγμά μας χρειαζόμαστε 2 εικόνες. Μία του Αστερίξ μαζί με τον Οβελίξ, και μία του αρχηγού Μαζεστίξ.

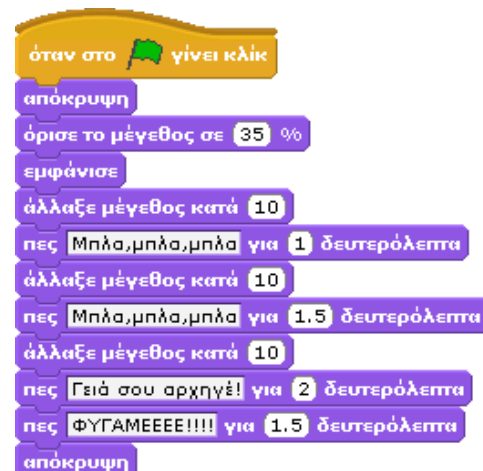
Στο σενάριο, ο αρχηγός θα περιμένει τους δύο πολεμιστές του για να τους αναθέσει μία αποστολή. Αυτοί εμφανίζονται στον ορίζοντα επομένως το μέγεθός τους πρέπει να είναι αρχικά μικρό και στην συνέχεια καθώς πλησιάζουν προς το μέρος του αρχηγού, το μέγεθός τους θα μεγαλώνει. Όταν φτάσουν αρ-

κετά κοντά, ανταλλάσσουν χαιρετισμό και ο αρχηγός τους αναθέτει την αποστολή, την οποία αποδέχονται και εξαφανίζονται για να την εκτελέσουν.

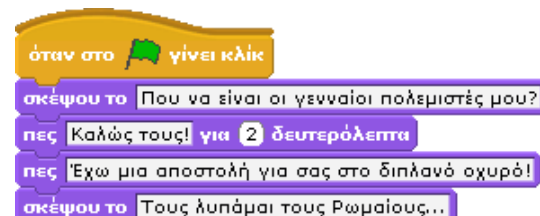
Όπως σε όλα τα προηγούμενα παραδείγματα, διαγράφουμε τη γάτα που εμφανίζεται στην οθόνη (δεξί κλικ --> διαγραφή), επιλέγουμε νέα μορφή ώστε να εισάγουμε τα δυο αντικείμενά μας.



Το σενάριο για τους δυο ήρωες πρέπει να περιλαμβάνει α) τον προσδιορισμό ενός μικρού αρχικού μεγέθους β) τη σταδιακή αύξηση του μεγέθους καθώς συζητούν μεταξύ τους γ) το διάλογο με τον αρχηγό δ) την απόκρυψή τους. Επειδή στο τέλος κάθε εκτέλεσης του προγράμματος τα αντικείμενα αποκρύπτονται, στην αρχή του προγράμματος θα πρέπει να χρησιμοποιήσουμε την εντολή εμφάνισε.



Για να πούμε στον αρχηγό τί να κάνει, κάνουμε κλικ στο σκίτσο του στη λίστα αντικειμένων, πάμε στο μενού Όψεις και επιλέγουμε τις παρακάτω εντολές



Αυτό ήταν, πατάμε κλικ στο πράσινο σημαδάκι και απολαμβάνουμε το νέο μας δημιούργημα! Βέβαια πάλι υπάρχει κάτι που μπορεί να βελτιωθεί στο σενάριο των δυο συντρόφων. Τι είναι αυτό;

Μπορείτε σαν άσκηση να κάνετε το εξής: Μπείτε στο facebook και κατεβάστε φωτογραφίες φίλων σας. Στην συνέχεια, μπείτε στο Scratch και μέσω της ζωγραφικής «κολλήστε» τα πρόσωπα των φίλων σας, στη θέση των κεφαλιών κάποιων ζώων (όχι και τόσο ευγενικό!) Βάλτε τους φίλους σας με τα νέα σώματα να αυξομειώνουν το σώμα τους και να μιλούν μεταξύ τους. Δείξτε το

δημιούργημα σας σε αυτούς που έχουν αίσθηση του χιούμορ και γελάστε μαζί τους!

Πέρα από τη μεταβλητή κατάσταση ενδυμασία, το Scratch μας προσφέρει και τη μεταβλητή κατάσταση μέγεθος με την οποία μπορούμε να παρακολουθούμε κάθε στιγμή το μέγεθος ενός αντικειμένου. Το φυσικό μέγεθος των μορφών, είναι το 100 ενώ π.χ. με μια αύξηση κατά 10 γίνεται 110. Η συγκεκριμένη μεταβλητή κατάσταση μας επιτρέπει να αναγνωρίζουμε πιο εύκολα πόσο κοντά ή μακριά είμαστε από το φυσικό μέγεθος των αντικειμένων μας και να σχεδιάζουμε πιο αποτελεσματικά τα έργα μας.

7.3.2 Δε μπορούμε να αλλάξουμε το σκηνικό δυναμικά;

Ας αφήσουμε λίγο τους χαρακτήρες μας ήσυχους για να ασχοληθούμε με το σκηνικό μας. Μην νομίζετε ότι δεν είναι και τόσο σημαντικό, καθώς όπως θα έχετε ήδη δει σε κόμικς και σε παιχνίδια, το εναλλασσόμενο σκηνικό αποτελεί απαραίτητη προϋπόθεση για να προκαλείται μια έντονη εμπειρία στο θεατή. Το σκηνικό είναι αυτό που μας μεταφέρει στις διάφορες τοποθεσίες μέσα στις οποίες οι ιστορίες μας διαδραματίζονται και στις διάφορες πίστες μέσα στις οποίες εξελίσσεται ένα παιχνίδι. Υπάρχει, λοιπόν, η ανάγκη να μπορούμε να αλλάζουμε το σκηνικό μας κατά την διάρκεια του έργου μας.

Στο 4^ο κεφάλαιο συζητήσαμε για το πώς μπορούμε να δημιουργήσουμε πολλαπλά υπόβαθρα για το σκηνικό μας. Τώρα θα συζητήσουμε τρεις εντολές που μας παρέχει το Scratch για να αλλάζουμε τα υπόβαθρα του έργου μας. Κάνουμε διπλό κλικ πάνω στο σκηνικό για να το επιλέξουμε και στη συνέχεια πηγαίνουμε στην παλέτα Όψεις. Παρατηρούμε αμέσως ότι οι διαθέσιμες εντολές για το σκηνικό είναι πλέον διαφορετικές συγκριτικά με τις εντολές για τα αντικείμενα. Έχουμε και λέμε λοιπόν:

αλλαγή σε υπόβαθρο...:

με τη χρήση της συγκεκριμένης εντολής δηλώνουμε ποιο υπόβαθρο θέλουμε να παρουσιαστεί (αντίστοιχη εντολή με την αλλαγή σε ενδυμασία...).

επόμενο υπόβαθρο: όταν χρησιμοποιείται η συγκεκριμένη εντολή, επιλέγεται αυτόματα το επόμενο υπόβαθρο από τη λίστα υποβάθρων του σκηνικού μας (εντολή αντίστοιχη με την επόμενη ενδυμασία).

υπόβαθρο: όταν κάνουμε κλικ πάνω στη συγκεκριμένη εντολή εμφανίζεται στη οθόνη

αλλαγή σε υπόβαθρο υπόβαθρο1 ▾

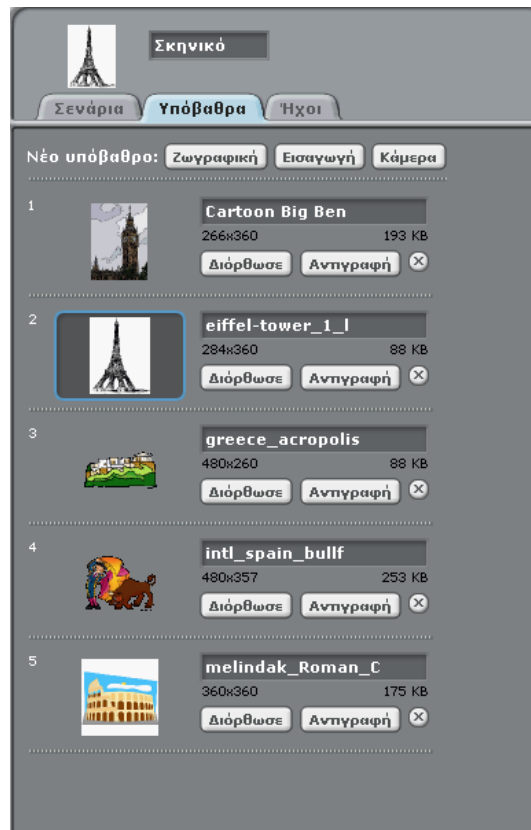
επόμενο υπόβαθρο

υπόβαθρο #

του έργου μας μία μεταβλητή κατάσταση που αναφέρει τον αριθμό του υποβάθρου που είναι επιλεγμένο κάθε στιγμή.

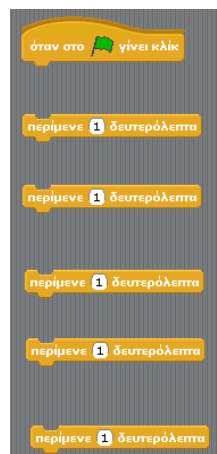
7.3.3 Παράδειγμα – Υπόβαθρα

Έστω λοιπόν, ότι κάνουμε ένα τουρ σε κάποιες Ευρωπαϊκές Πρωτεύουσες! Έχουμε λοιπόν κάποια υπόβαθρα τα οποία έχουμε εισάγει στο σκηνικό μας:



Θέλουμε τα υπόβαθρα να εναλλάσσονται στην οθόνη μας με την σειρά και στο τέλος να εμφανίζουμε αυτό που μας άρεσε περισσότερο, τον πύργο του Άιφελ.

Πηγαίνουμε στο μενού Έλεγχος, και εισάγουμε την εντολή όταν στην πράσινη σημαία γίνει κλικ. Εισάγουμε ακόμη 4 φορές την εντολή περίμενε 1 δευτερόλεπτα. Θα τη χρησιμοποιήσουμε ώστε να προσδιορίσουμε πόσο χρόνο θα παραμείνει στην οθόνη κάθε υπόβαθρο.



Ανάμεσα στις 4 αυτές εντολές, εισάγουμε 5 φορές την εντολή επόμενο υπόβαθρο από το μενού Όψεις ώστε να εμφανιστούν

με τη σειρά τα 5 σκηνικά. Στο τέλος του σεναρίου με τη χρήση της εντολής αλλαγή σε υπόβαθρο... προσδιορίζουμε ότι θέλουμε να παρουσιαστεί το υπόβαθρο που περιέχει τον πύργο του Άιφελ.



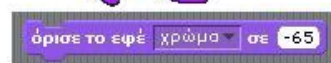
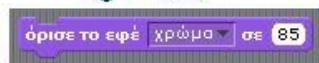
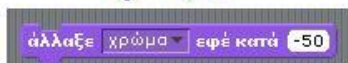
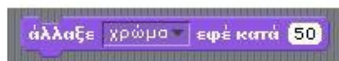
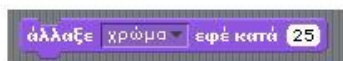
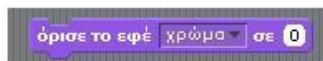
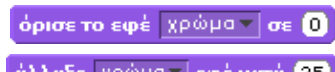
[07_π07.sb]

Μήπως όμως ο προγραμματιστής μας ήταν λίγο αμελής για άλλη μια φορά; Εντοπίστε και διορθώστε τα δυο προβλήματα που περιέχει το συγκεκριμένο σενάριο.

7.4 Δυναμικά εφέ

7.4.1 Οι εντολές

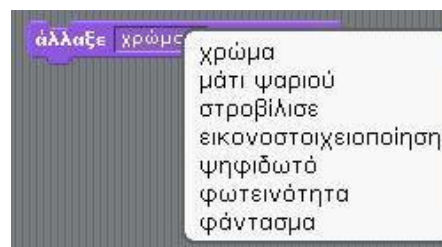
Το Scratch μας παρέχει ακόμη περισσότερα εργαλεία για να αλλάζουμε δυναμικά την εμφάνιση των αντικειμένων σας. Από την παλέτα Όψεις οι εντο-



λές, αλλάξε...εφέ κατά... και όρισε το εφέ...σε... μας επιτρέπουν να εφαρμόσουμε εντυπωσιακά εφέ πάνω στην τρέχουσα ενδυμασία των χαρακτήρων μας. Τα εφέ φέρουν τους τίτλους:

- Χρώμα,
- Μάτι ψαριού,
- Στροβίλισε,
- Εικονοστοιχείοποίηση,
- Ψηφιδωτό,
- Φωτεινότητα,
- Φάντασμα,

ενώ για κάθε εφέ καλούμαστε να ορίσουμε μια αριθμητική τιμή που προσδιορίζει την ένταση του εφέ πάνω στον χαρακτήρα μας.



Ευτυχώς, υπάρχει και η εντολή καθάρισε τα γραφικά εφέ ώστε να επαναφέρουμε το αντικείμενό μας στην αρχική του εμφάνιση. Τα πειράματά μας θα είναι πολλά και ίσως φτάσετε πολλές φορές σε σημείο που δεν θα αναγνωρίζετε τα αντικείμενά σας. Μη ξεχνάτε αυτήν την εντολή!

7.4.2 Εφέ χρώμα

Θα ξεκινήσουμε με τη σειρά και θα μελετήσουμε το εφέ «Χρώμα». Δείτε τις επιδράσεις που έχει το συγκεκριμένο εφέ με διαφορετικές τιμές έντασης στη χαρακτηριστική γάτα του Scratch, στην εικόνα που ακολουθεί. Είναι προφανές, ότι με το συγκεκριμένο εφέ αλλάζουμε τα χρώματα του αντικειμένου μας.

Ας δούμε ένα πιο ολοκληρωμένο παράδειγμα. Όλοι μας μετά από μια δυνατή βροχή, αναζητούμε στον ορίζοντα, από την αντίθετη μεριά του ήλιου, να δούμε το ουράνιο τόξο. Πως θα μπορούσαμε να δημιουργήσουμε και εμείς ένα ουράνιο τόξο με το Scratch;

Μπορούμε να δημιουργήσουμε 5 αντικείμενα-γραμμές, όπως οι πολύχρωμες ρίγες που εμφανίζονται στην επόμενη εικόνα και στο σενάριο τους να ενσωματώσουμε τις εντολές που παρουσιάζονται στο παρακάτω σενάριο. Τρέξτε το παράδειγμα και παρατηρείστε την εναλλαγή των χρωμάτων. Στο συγκεκριμένο σενάριο, όπως μπορείτε εύκολα να διακρίνετε αλλάζουμε κατά 47 το εφέ «Χρώμα» κάθε αντικειμένου κάθε δυο δευτερόλεπτα και για πάντα. Δεν είναι εντυπωσιακό το αποτέλεσμα; Δοκιμάστε να πειραματιστείτε με διαφορετικές τιμές.



[07_π08.sb]

Σε ποια άλλα παραδείγματα θα μπορούσε να είχε εφαρμογή το συγκεκριμένο εφέ; Σκεφτείτε ένα παιχνίδι όπου ο ήρωας καθώς ακουμπά εμπόδια που θα έπρεπε να αποφεύγει η «Μπάρα Ε-νέργειας» αλλάζει χρώμα και από πράσινη, γίνεται κίτρινη και κόκκινη λίγο πριν το GAME OVER... Δείτε την εικόνα που ακολουθεί και σκεφτείτε ένα σενάριο με λίγες εντολές που θα υλοποιούσε αυτή την εναλλαγή στα χρώματα.

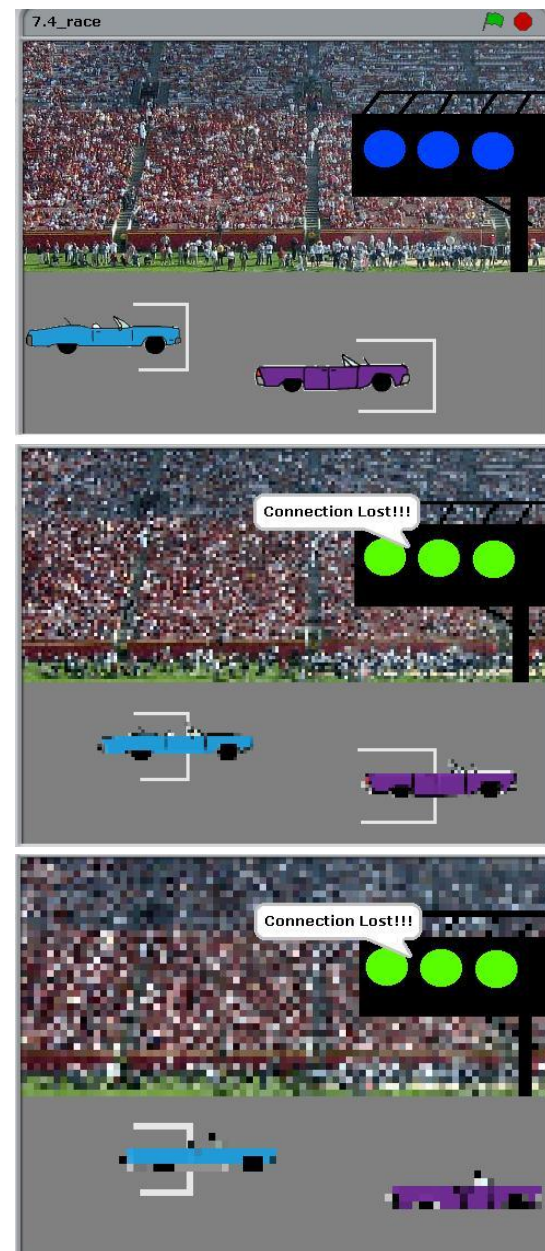


7.4.3 Εφέ Εικονοστοιχειοποίησης

Το εφέ εικονοστοιχειοποίησης παρουσιάζει το αντικείμενο μας σαν ένα σύνολο από κουτάκια ενός χρώματος. Όπως βλέπετε στην εικόνα που ακολουθεί, η ποιότητα της αρχικής εικόνας της γάτας είναι χειρότερη καθώς τα δομικά στοιχεία της είναι κουτάκια διαφορετικών μεγεθών. Τι παρατηρείτε ως προς το πρόσημο της έντασης του εφέ;



Ακόμα ένα παράδειγμα του συγκεκριμένου εφέ φαίνεται στις παρακάτω τρεις εικόνες. Η πρώτη είναι χωρίς εφέ ενώ στις επόμενες δυο έχουμε εφαρμόσει δυο φορές διαδοχικά την εντολή άλλαξε εικονοστοιχειοποίηση εφέ κατά 25.



[07_π09.sb]

Όπως είναι φανερό, τα κουτάκια της τελευταίας εικόνας είναι τα μεγαλύτερα και η σαφήνεια της εικόνας, συνεπώς, η χειρότερη.

7.4.4 – Εφέ ψηφιδωτό

Το εφέ ψηφιδωτό παρουσιάζει πολλαπλασιάζεις τις εμφανίσεις του αντικειμένου στο χώρο όμως που καταλάμβανε το αρχικό αντικείμενο. Μελετήστε την παρακάτω εικόνα για να κατανοήσετε καλύτερα το εφέ και τα αποτελέσματα διαφορετικών τιμών έντασης του εφέ.



Δείτε το αποτέλεσμα της εντολής άλλαξε ψηφιδωτό εφέ κατά 15 σε μια περίπτωση όπου το φεγγάρι προκάλεσε παραισθήσεις...



[07_π10.sb]



7.4.5 Εφέ φάντασμα

Το εφέ φάντασμα κάνει τα αντικείμενά μας αόρατα! Δείτε στην επόμενη εικόνα τα αποτελέσματα του εφέ με διαφορετικές τιμές στην ένταση του εφέ:



Πρέπει να έχετε παρατηρήσει ότι σε ορισμένα εφέ οι αρνητικές τιμές στην ένταση του εφέ προκαλούν ακριβώς τα ίδια αποτελέσματα με τις θετικές τιμές.

7.4.5 Εφέ φωτεινότητα

Το εφέ φωτεινότητα μας επιτρέπει να προσδιορίσουμε πόσο φωτεινή είναι η εμφάνιση του αντικειμένου μας. Στην παρακάτω εικόνα εφαρμόζουμε το εφέ στο αντικείμενο-εικόνα που βρίσκεται στην οθόνη του υπολογιστή. Ποιες τιμές κάνουν ποιο φωτεινό και ποιες ποιο σκοτεινό το αντικείμενο; Πειραματιστείτε με θετικές και αρνητικές τιμές στην ένταση του εφέ.



7.4.6 Εφέ στροβίλιση

Θα μπορούσαμε να φανταστούμε ότι η εντολή άλλαξε στροβίλιση εφέ κατά... αλλάζει τη εμφάνιση του αντικειμένου μας σαν να υπήρχε μια δίνη στη μέση ακριβώς του αντικειμένου που το να το τράβαγε προς το εσωτερικό. Το θετικό ή αρνητικό πρόσημο στην ένταση του εφέ προκαλεί διαφορετική φορά περιστροφής. Παρατηρήστε στην παρακάτω εικόνα τις συνέπειες που έχει η διαδοχική εφαρμογή συγκεκριμένου εφέ.



[07_π11.sb]

7.4.7 Εφέ μάτι ψαριού

Πολλοί από εσάς, ενώ δε θα έπρεπε, έχετε μια αγάπη για πατατάκια, γαριδάκια κτλ. Αν τα φάτε γρήγορα, δεν αισθάνεστε την κοιλιά σας να πρήζεται; Η εντολή άλλαξε μάτι ψαριού εφέ κατά... αρχίζει «φουσκώνει» το αντικείμενό μας από το κέντρο του ανάλογα με την ένταση του εφέ που επιλέγουμε. Πειραματιστείτε με διαφορετικές τιμές έντασης και δείτε τα αποτελέσματα μια θετικής τιμής στην κοιλιά ενός χαρακτήρα, όπως φαίνεται στην επόμενη εικόνα.

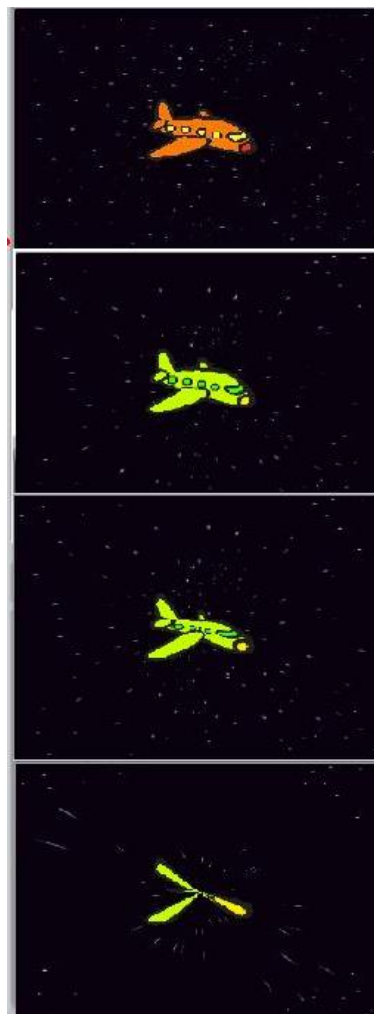


[07_π12.sb]

Αν για θετικό πρόσημο στην ένταση του εφέ, φουσκώνει το αντικείμενο τότε για αρνητικό μήπως θα ρουφούσε την κοιλιά του; Διαβάστε το σενάριο στο αριστερό μέρος της εικόνα 7.69 και δείτε πως επιδρά η εντολή άλλαξε μάτι ψαριού εφέ κατά -10 σε ένα αστέρι και στα αστέρια του γαλαξία καθώς αυτά αναπτύσσουν ταχύτητες κοντά στην ταχύτητα του φωτός.



[07_π13.sb]



Περίληψη

Το Scratch μετά από αυτό το κεφάλαιο απέκτησε σίγουρα περισσότερο ενδιαφέρον, αφού τα αντικείμενά μας πλέον μιλάνε, σκέφτονται και μπορούν να αλλάξουν εμφάνιση με πάρα πολλούς τρόπους.

Αρχικά μάθαμε πως να αλλάζουμε την εμφάνιση του αντικείμενου μέσω της αλλαγής της ενδυμασίας του. Οι εντολές αλλαγή σε ενδυμασία..., επόμενη ενδυμασία και ενδυμασία # μας βοήθησαν στο να μεταμορφώνουμε τα αντικείμενα μας, εύκολα και γρήγορα, ενώ οι εντολές αλλαγή σε υπόβαθρο..., επόμενο υπόβαθρο και υπόβαθρο # μας βοηθούν να αλλάξουμε την εμφάνιση του σκηνικού. Οι εντολές πήγαινε σε πρώτο πλάνο και πήγαινε πίσω...επίπεδα μας βοήθησαν να «τακτοποιήσουμε» τη σειρά με την οποία εμφανίζονται τα αντικείμενά μας. Με τα μπαλονάκια (εντολή πες...για...δευτερόλεπτα και πες...) και τα συννεφάκια κειμένων (εντολή σκέψου το...για...δευτερόλεπτα και σκέψου το...), δώσαμε τη δυνατότητα στα αντικείμενά μας να μιλάνε και να σκέφτονται. Έπειτα, με τις εντολές άλλαξε μέγεθος κατά..., όρισε το μέγεθος σε...%, μέγεθος, εμφάνισε, απόκρυψη μικρύναμε, μεγαλώσαμε, κρύψαμε και επανεμφανίσαμε τα αντικείμενά μας.

Τέλος, μελετήσαμε τα διαθέσιμα εφέ μέσα από τις εντολές άλλαξε...εφέ κατά..., όρισε το εφέ...σε... και καθάρισε τα γραφικά εφέ που μπορούν να εφαρμοστούν τόσο για τα αντικείμενα όσο και για το σκηνικό.

Μετά το τέλος του συγκεκριμένου κεφαλαίου θα πρέπει να είστε σε θέση να δημιουργήσετε πραγματικά εντυπωσιακές ιστορίες.

Ερωτήσεις

1) Ποια είναι η διαφορά μεταξύ των εντολών πες... και πες...για...δευτερόλεπτα.

2) Περιγράψτε με δικά σας λόγια την επίδρασης του εφέ "Ψηφιδωτό" στα αντικείμενα.

3) Αναλύστε πως οι αλλαγές πρόσημου στην ένταση του εφέ "στροβίλιση" επιδρούν στη φορά περιστροφής του αντικείμενου.

4) Σωστό ή Λάθος και εξηγήστε γιατί!

Σ-Λ Η εντολή επόμενη ενδυμασία είναι ολόιδια με την αλλαγή σε ενδυμασία #

Σ-Λ Με την εντολή πήγαινε σε πρώτο πλάνο πηγαίνει το αντικείμενο μας ακριβώς ένα επίπεδο μπροστά.

Σ-Λ Η εντολή πες...για...δευτερόλεπτα δεν μπορεί να ακολουθείται από κάποια άλλη εντολή.

Σ-Λ Η εντολή καθάρισε τα γραφικά εφέ απενεργοποιεί όλα τα εφέ που έχουν καθοριστεί για ένα αντικείμενο.

5) Επιλέξτε τη σωστή απάντηση και βρείτε την κρυμμένη λέξη.

@Εμφανίζει ένα συννεφάκι κειμένου πάνω από το αντικείμενό μας.

Ο) σκέψου ... Χ) πες ...για # δευτερόλεπτα Δ) όρισε το εφέ σε...

@Δεν είναι ακριβώς εντολή αλλά μεταβλητή κατάσταση

Φ) άλλαξε...εφέ κατά... Ε)όρισε το εφέ σε... Ψ) ενδυμασία #

@Αυτή η εντολή μοιάζει πολύ με μία άλλη με την διαφορά ότι επιλέγει έμμεσα ενδυμασία

Κ) ενδυμασία # Α) αλλαγή σε ενδυμασία # Ε)επόμενη ενδυμασία

@Εντολή για τον απόλυτο προσδιορισμό της έντασης του εφέ

Ι) όρισε το εφέ σε... Σ) άλλαξε...εφέ κατά... Λ) καθάρισε τα γραφικά εφέ

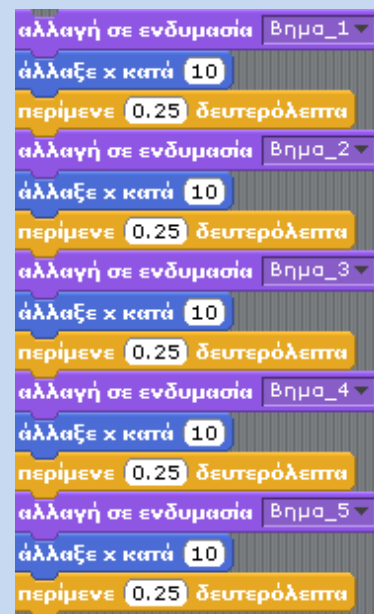
@Με την εντολή αυτή διαλέγουμε το αντικείμενό μας να είναι "μπροστά" από οποιονδήποτε άλλο αντικείμενο του σκηνικού

Α) Πήγαινε πίσω # επίπεδα Σ)Πήγαινε μπροστά

ΚΡΥΜΜΕΝΗ ΛΕΞΗ : _ _ _ _ _

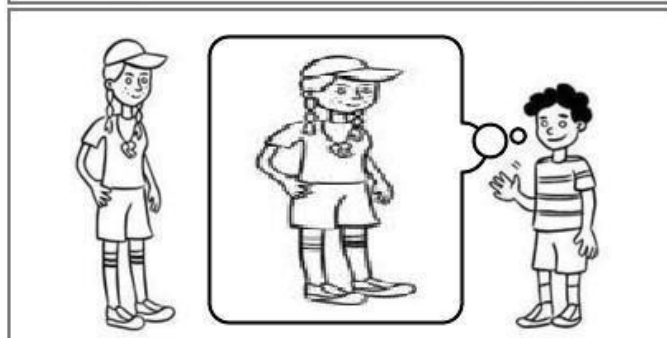
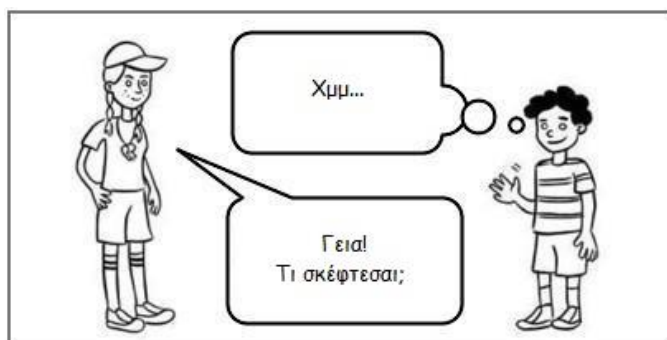
Δραστηριότητες

1) Εξηγήστε τι κάνει το παρακάτω σενάριο:



2) Επιλέξτε ένα επεισόδιο από ένα κόμικς στην τάξη και προσπαθήστε να το αναπαραστήσετε ζωντανά στο Scratch. Χωρίστε σε ομάδες των δυο ατόμων και παρουσιάστε το αποτέλεσμα στους συμμαθητές σας. Ποια ομάδα είχε την περισσότερη δημιουργικότητα;

3) Δημιουργήστε στο Scratch ένα σκετσάκι για την τάξη σας. Επιλέξτε 4-5 συμμαθητές σας, δημιουργήστε ως σκηνικό μια τάξη, και προσπαθήστε να αναπαράγετε χαρακτηριστικά συμβάντα που σας έκαναν να γελάσετε. Μη ξεχνάτε να αλλάζετε κατά τη διάρκεια του έργου σας το μέγεθος των αντικειμένων, τις ενδυμασίες τους και το επίπεδο στο οποίο βρίσκονται ώστε να κάνετε πιο αναπαραστατική την παρουσίασή σας. (δημιουργήστε το σενάριό σας πρώτα σε ένα πίνακα-χρονοδιάγραμμα και στη συνέχεια να παράγετε τα σενάρια των χαρακτήρων σας).



Κεφάλαιο 8: Ήχοι

Σε αυτό το κεφάλαιο:

8.1 Βασικές εντολές ήχου

8.2 Προγραμματίζοντας ήχους

8.3 Παραδείγματα

«Όταν αλλάζει η μουσική, αλλάζει κι ο χορός»

(Αφρικανική παροιμία)



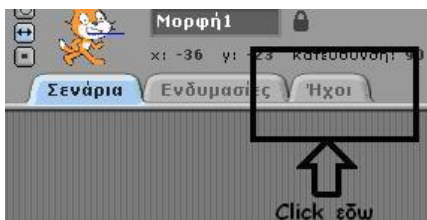
8.1 Βασικές εντολές ήχου στο Scratch

Στα προηγούμενα κεφάλαια έχουμε δει πως μπορούμε να κινήσουμε αντικείμενα, να ζωγραφίσουμε στην οθόνη του Scratch και να αλλάξουμε τις ενδυμασίες των αντικειμένων μας. Προσπαθήσαμε να αναπαραστήσουμε ιστορίες αλλά μέχρι τώρα μιλούσαμε μάλλον για βουβό κινηματογράφο!

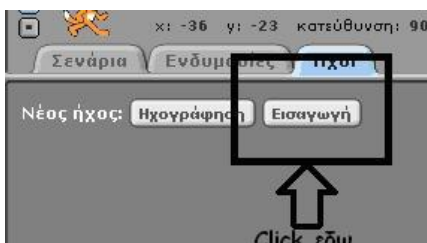
Από το τέταρτο κεφάλαιο, όμως, είχαμε αναφέρει ότι κάθε αντικείμενο πέρα από το σενάριο και τις ενδυμασίες του, μπορεί να περιέχει και τους δικούς του ήχους. Στο κεφάλαιο αυτό, θα δούμε πως μπορούμε να χρησιμοποιούμε είτε έτοιμους ήχους και μουσικά κομμάτια είτε ήχους που θα συνθέτουμε με την παλέτα εντολών **Ήχοι** που μας παρέχει το Scratch.

8.1.1 Αναπαραγωγή και stop/pause έτοιμων ηχητικών clips

Για να εισάγουμε ένα ή περισσότερα μουσικά κλιπ σε ένα αντικείμενο, επιλέγουμε από το μεσαίο παράθυρο του Scratch την καρτέλα «Ήχοι».

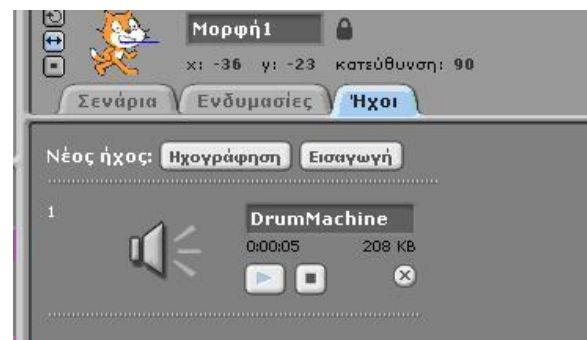


Το Scratch μας δίνει δυο επιλογές για την εισαγωγή ήχων. Μας επιτρέπει είτε να ηχογραφήσουμε ένα νέο ήχο με τη χρήση του μικροφώνου του υπολογιστή μας, είτε να εισάγουμε έναν έτοιμο ήχο από τη βιβλιοθήκη του Scratch ή την προσωπική μουσική μας βιβλιοθήκη. Εμείς προς το παρόν θα επιλέξουμε να εισάγουμε έναν έτοιμο ήχο πατώντας το κουμπί «Εισαγωγή».



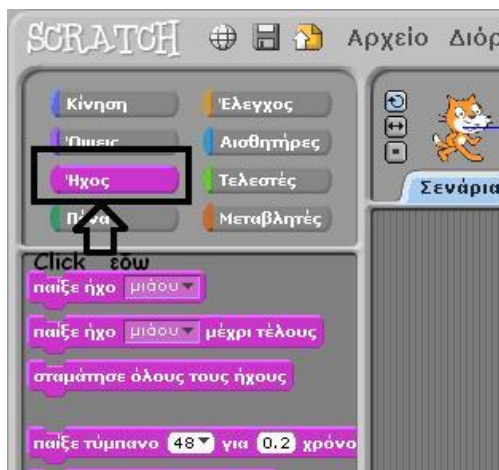
Αμέσως ανοίγει ένα παράθυρο που μας παρουσιάζει ομαδοποιημένα σε φακέλους όλα τα ηχητικά κλιπ της βιβλιοθήκης του

Scratch. Εμείς από το φάκελο Music Loops επιλέγουμε το ηχητικό κλιπ DrumMachine και κάνοντας κλικ στο «Εντάξει», το clip εισάγεται στη λίστα ήχων του αντικειμένου μας που στην προκειμένη περίπτωση είναι η γάτα.



Μπορούμε να ακούσουμε το κλιπ που μόλις εισάγαμε πατώντας το κουμπί «Play» ενώ με το κουμπί «Stop» η αναπαραγωγή σταματά. Το συγκεκριμένο κλιπ «DrumMachine» έχει διάρκεια 5 δευτερολέπτων και μέγεθος 208Kb, όπως φαίνεται κάτω από το όνομά του. Καλό είναι να παρατηρούμε το μέγεθος των ήχων που εισάγουμε στα έργα μας καθώς αυτό επηρεάζει σημαντικά το συνολικό μέγεθος των έργων.

Πως μπορούμε να ακούσουμε το συγκεκριμένο ήχο κατά τη διάρκεια εκτέλεσης του έργου μας; Επιλέγοντας την παλέτα εντολών **Ήχος**, θα εμφανιστούν οι εντολές που σχετίζονται με την αναπαραγωγή των ήχων.



Για την αναπαραγωγή ενός ήχου, το Scratch μας παρέχει δυο εντολές, τις παιξε ήχο... και παιξε ήχο...μέχρι τέλους. Και στις δυο περιπτώσεις μπορούμε να επιλέξουμε τον ήχο που θέλουμε να ακούσουμε από την αντίστοιχη λίστα που διαθέτουν οι δυο εντολές. Θυμηθείτε όμως ότι η λίστα παρουσιάζει μόνο τους ήχους που είναι διαθέσιμοι στο συγκεκριμένο και μόνο αντικείμενο.

Η πρώτη εντολή, παιξε ήχο..., ξεκινά την αναπαραγωγή του σχετικού ήχου επιτρέποντας όμως ταυτόχρονα και την άμεση εκτέλεση της επόμενης εντολής του σεναρίου. Δηλαδή ακούμε τον ήχο και, ανεξάρτητα από τη διάρκειά του, οι εντολές του σεναρίου που ακολουθούν, εκτελούνται κανονικά. Αντίθετα, η εντολή παιξε ήχο... μέχρι τέλους σταματά το σενάριο μέχρι να ολοκληρωθεί η αναπαραγωγή του επιλεγμένου ήχου και μόνο τότε αρχίζει η εκτέλεση των υπολοίπων εντολών του σεναρίου.

Για να δοκιμάσετε πρακτικά τη διαφορά των δυο εντολών, μπορείτε να εισάγετε ως ήχο στο αντικείμενό σας ένα αγαπημένο σας τραγούδι και στη συνέχεια να δημιουργήσετε δυο σενάρια όπως φαίνονται στην επόμενη εικόνα. Στο πρώτο συνδυάστε την εντολή παιξε ήχο... με την εντολή κινήσου...βήματα βάζοντας ένα σχετικά μεγάλο αριθμό βημάτων. Στο δεύτερο σενάριο, συνδυάστε την εντολή παιξε ήχο ...μέχρι τέλους με την εντολή κινήσου...βήματα. Τρέξτε τα δυο σενάρια ξεχωριστά. Τι παρατηρείτε; Σε ποια περίπτωση, ο χαρακτήρας μας κινείται μόνο μετά το τέλος του τραγουδιού μας;



Επιπλέον, με τη χρήση της εντολής σταμάτησε όλους τους ήχους μπορούμε να σταματήσουμε την αναπαραγωγή των ηχητικών κλιπ οποιαδήποτε στιγμή και από οποιοδήποτε αντικείμενο. Προσέξτε όμως ότι η εντολή παιξε ήχο...μέχρι τέλους δε θα σταματήσει την αναπαραγωγή του ήχου αν η εντολή σταμάτησε όλους τους ήχους την ακολουθεί. Πρώτα θα ολοκληρωθεί ο ήχος και μετά θα εκτελεστεί η σταμάτησε όλους τους ήχους, οπότε η εντολή δε θα έχει νόημα. Η συγκεκριμένη εντολή όμως μπορεί να σταματήσει τον ήχο που

αναπαράγεται από την εντολή παιξε ήχο_μέχρι τέλους αν εκτελεστεί από ένα άλλο αντικείμενο τη στιγμή που αναπαράγεται ο ήχος. Δείτε ακόμη ένα παράδειγμα στην παρακάτω εικόνα:



8.1.2 Ρύθμιση της έντασης

Είναι πιθανόν να θέλουμε οι χρήστες μας να ακούσουν έναν ήχο κινδύνου πολύ δυνατά, ενώ να ακούν τη μουσική του παιχνιδιού μας σε χαμηλή ένταση. Με τις εντολές όρισε ένταση σε...% και άλλαξε ένταση κατά... μπορούμε να ρυθμίσουμε την ένταση των ήχων κατά τη στιγμή που αναπαράγονται. Με την εντολή όρισε ένταση σε...% προσδιορίζουμε την ένταση που θέλουμε να έχει ο ήχος σε σχέση με το κανονικό επίπεδο έντασης. Επομένως, προσδιορίζοντας 110% ένταση, ζητάμε από το Scratch να αναπαράγει τον ήχο 10% πιο δυνατά από τα κανονικά επίπεδα αναπαραγωγής του. Από την άλλη μεριά, η εντολή άλλαξε ένταση κατά... κάνει σχετικές τροποποιήσεις στην ένταση των ήχων. Επομένως, μια αλλαγή κατά 10 μονάδες μετά από μια εντολή όρισε ένταση σε 110% θα προκαλούσε επίπεδα έντασης της τάξης του 120%. Οι διαφορές αυτών των εντολών είναι ανάλογες με αυτές των εντολών θέσε το X ίσο με... και άλλαξε X κατά..., όρισε το μέγεθος σε % και άλλαξε μέγεθος κατά..., όρισε το χρώμα της πέννας σε ... και άλλαξε χρώμα της πέννας κατά...

Ας δούμε ένα παράδειγμα στο οποίο θα προσπαθήσουμε να προσομοιώσουμε το εφέ fade out (σβήσιμο ήχου) σε ένα τραγούδι. Αυτό που θέλουμε να κάνουμε δηλαδή είναι να βρούμε έναν τρόπο ώστε ένα τραγούδι που εισάγαμε στο Scratch να τελειώνει την αναπαραγωγή του με σταδιακή μείωση της έντασης του ήχου του. Δείτε το αντίστοιχο έργο στην επόμενη εικόνα.

[08_π01.sb]



Αρχικά πρέπει να εισάγουμε το τραγούδι που επιθυμούμε σε ένα αντικείμενο ή στους ήχους του σκηνικού μας. Στη συνέχεια, θα χρησιμοποιήσουμε την εντολή όρισε ένταση σε...% και θα ορίσουμε την ένταση στο 100% ώστε αρχικά το κομμάτι μας να αναπαράγεται κανονικά. Στο επόμενο βήμα, θα χρησιμοποιήσουμε την εντολή παιξε ήχο... για το τραγούδι μας και θα περιμένουμε για ένα σχετικά μεγάλο διάστημα, έστω ένα λεπτό, πριν αρχίσουμε να εφαρμόζουμε το εφέ μας. Η εντολή περίμενε για...χρόνους μας επιτρέπει να καθυστερήσουμε την εκτέλεση της επόμενης εντολής όπως ακριβώς η εντολή περίμενε...δευτερόλεπτα. Ο τρόπος με τον οποίο μετρά το χρόνο η κάθε μια είναι διαφορετικός αλλά αυτό θα το συζητήσουμε αργότερα.

Σε κάθε βήμα λοιπόν θα αλλάζουμε την ένταση του κατά -15 χρησιμοποιώντας την εντολή άλλαξε ένταση κατά... και θα περιμένουμε 0.5 χρόνους για την επόμενη μείωση. Έτσι, θα μειώνουμε σταδιακά την ένταση του ήχου.

Δυο ερωτήσεις:

- οι προγραμματιστές μας κάνουν συνεχώς λάθη: ποιο λάθος περιέχει το προηγούμενο σενάριο;
- πόσα βήματα χρειάζομαστε ώστε να μην ακούγεται καθόλου στο τέλος του σεναρίου μας το κομμάτι, αν το βήμα μείωσης της έντασης ήταν το -10;

8.2: Προγραμματίζοντας ήχους

Το Scratch μας παρέχει εντολές για να προγραμματίσουμε ήχους, δηλαδή να ζητήσουμε την αναπαραγωγή συγκεκριμένων νοτών από διαφορετικά μουσικά όργανα σε διαφορετικές χρονικές στιγμές και με συγκεκριμένο ρυθμό! Ουσιαστικά μας δίνει τη δυνατότητα να συνθέσουμε από την αρχή τα δικά μας μουσικά έργα με μουσικούς τα αντικείμενά μας!

8.2.1: Παρουσίαση τυμπάνων και ρυθμού

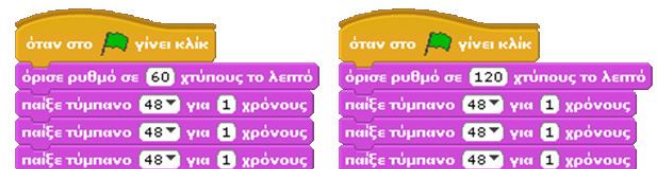
Πρώτα από όλα ο ρυθμός. Και το όργανο που παραδοσιακά δίνει το ρυθμό δεν είναι άλλο από τα τύμπανα. Στην παλέτα εντολών Ήχοι υπάρχει η εντολή παιξε τύμπανο...για...χρόνους η οποία αναπαράγει ήχους κρουστών. Η πρώτη παράμετρος της εντολής μας δίνει τη δυνατότητα να επιλέξουμε το επιθυμητό κρουστό από μια εκπληκτική λίστα εναλλακτικών. Η λίστα περιλαμβάνει τα βασικά στοιχεία ενός σετ τυμπάνων όπως μπάσο, πιατίνια, ταμπούρο κτλ. αλλά και εξωτικά κρουστά όπως μαράκες, καμπάσα και αγκόλο. Αξίζει να πειραματιστείτε με τις εναλλακτικές που έχετε στη διάθεσή σας.

Σας θυμίζουμε ότι για να δοκιμάσετε μια εντολή μπορείτε απλά να τη μεταφέρετε στο χώρο του σεναρίου, να αλλάξετε τις επιλογές της (π.χ. την επιλογή του τύμπανου) και να κάνετε διπλό κλικ πάνω στην εντολή. Η ενέργεια αυτή εκτελεί τη συγκεκριμένη εντολή, οπότε γρήγορα ακούτε τον αντίστοιχο ήχο.

Η δεύτερη παράμετρος της εντολής παιξε τύμπανο...για...χρόνους είναι λίγο πιο σύνθετη. Κατ'αρχάς, αναφέρετε σε μονάδα χρόνου, δηλαδή πόσο θα διαρκέσει

ο ήχος του τυμπάνου. Όπως όμως μπορείτε να παρατηρήσετε και οι υπόλοιπες εντολές στην παλέτα των ήχων δεν χρησιμοποιούν τα δευτερόλεπτα ως μονάδα χρόνου αλλά τους «χρόνους». Τι είναι οι χρόνοι; Θυμάστε την έννοια του μετρονόμου από το μάθημα της μουσικής; Ο μετρονόμος παράγει ένα τακτικά επαναλαμβανόμενο τικ βάσει του οποίου οι μουσικοί παίζουν τα κομμάτια τους, αποφασίζουν δηλαδή το ρυθμό του τραγουδιού τους. Ο μετρονόμος μπορεί να ρυθμιστεί ώστε να επαναλαμβάνει το τικ πιο αργά ή πιο γρήγορα. Ανάλογα με το ρυθμό του μετρονόμου ένας μουσικός θα παίξει αντίστοιχα το κομμάτι του πιο αργά ή πιο γρήγορα.

Το Scratch χρησιμοποιεί την έννοια του χρόνου όπως τα τικ του μετρονόμου. Δηλαδή ο χρόνος είναι σχετική έννοια (ας θεωρήσουμε ότι αντιστοιχίζεται σε ένα τικ ενός μετρονόμου) και ρυθμίζεται από την εντολή όρισε ρυθμό σε...χτύπους το λεπτό. Παρατηρήστε ότι η αρχική τιμή του ρυθμού είναι 60 χτύποι το λεπτό δηλαδή ένας χτύπος το δευτερόλεπτο. Σε αυτήν την περίπτωση ο χτύπος ταυτίζεται με το δευτερόλεπτο και οι εντολές περίμενε για 1 χρόνο και περίμενε 1 δευτερόλεπτο θα πρέπει να έχουν το ίδιο αποτέλεσμα. Αν όμως ορίσουμε το ρυθμό σε 120 χτύπους το λεπτό, τότε η εντολή περίμενε για 1 χρόνο αντιστοιχίζεται με την εντολή περίμενε 0.5 δευτερόλεπτο ($120/60=2$ χτύποι ανά δευτερόλεπτο). Στην παρακάτω εικόνα παρότι οι εντολές παιξε τύμπανο...για...χρόνους είναι ίδιες, θα ακούσουμε δυο διαφορετικά αποτελέσματα. Στο αριστερό σενάριο, θα ακούσουμε ένα χτύπημα κάθε δευτερόλεπτο και στο δεξί σενάριο, 2 χτυπήματα το δευτερόλεπτο.



Γιατί όμως όλο αυτό το μπέρδεμα; Γιατί δεν ορίζουμε τους χρόνους σε δευτερόλεπτα για να είναι πιο εύκολη η δημιουργία του έργου μας; Γιατί έτσι μας παρέχεται μια εξαιρετική ευελιξία: δημιουργούμε μια σύνθεση με τους «χρόνους» και στη συνέχεια μπορούμε να αλλάξουμε το ρυθμό της οποιαδήποτε στιγμή επιθυμούμε, μπορούμε να δώσουμε ένα πιο γρήγορο ρυθμό αν υπάρχει περισσότερη αγωνία ή ένα πιο αργό ρυθμό αν ο χρήστης μας χαλαρώνει.

Είναι σημαντικό να τονίσουμε ότι οι εντολές εκτελούνται για όσο χρονικό διάστημα προσδιορίζεται από τη δεύτερη παράμετρο και μόνο μετά από αυτό το διάστημα εκτελείται οποιαδήποτε εντολή ακολουθεί. Δοκιμάστε να βάλετε την κινήσου...βήματα μετά την παιξε τύμπανο...για...χρόνους για να δείτε ότι το αντικείμενό σας θα κινηθεί μετά το πέρασμα των χρόνων που έχουν προσδιοριστεί στην πρώτη εντολή.

Πέρα από την εντολή όρισε ρυθμό σε...χτύπους το λεπτό υπάρχει και η εντολή άλλαξε ρυθμό κατά... η οποία αλλάζει σχετικά το ρυθμό της μουσικής μας. Έτσι αν είχαμε ρυθμό 60 και

τον αλλάξουμε κατά 10, ο ρυθμός μας θα γίνει 70 χτύποι το δευτερόλεπτο. Αρνητική αλλαγή σημαίνει πιο αργή αναπαραγωγή της σύνθεσής μας. Ευτυχώς, υπάρχει επιπλέον η μεταβλητή κατάσταση **Ρυθμός** η οποία μας επιτρέπει να παρατηρούμε τις τιμές του ρυθμού κατά την εκτέλεση του έργου μας.

Τρέξτε το παρακάτω σενάριο και προσπαθήστε να εξηγήσετε τη λειτουργία του. Για να βοηθηθείτε επιλέξτε να βλέπετε τη μεταβλητή κατάσταση **Ρυθμός**.



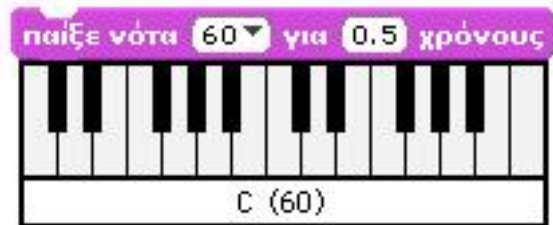
[08_π02.sb]

8.2.2 Συνθέτοντας τη μελωδία

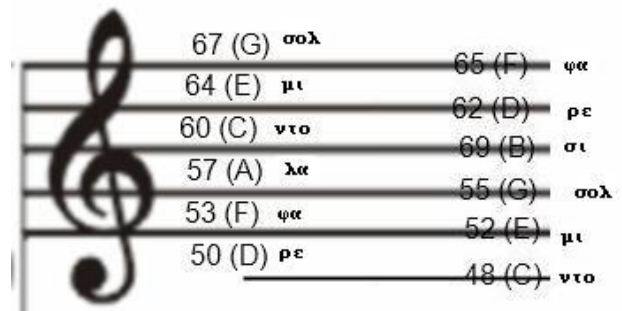
Αφού μάθαμε να δίνουμε το ρυθμό με τα κρουστά, ήρθε η ώρα να ασχοληθούμε με τη μελωδία της μουσικής μας σύνθεσης! Χρειαζόμαστε δυο λειτουργίες: α) τον προσδιορισμό του οργάνου που θέλουμε να παίξει το αντικείμενό μας, β) τον προσδιορισμό των νोटών της μελωδίας μας.

Δυο αντίστοιχες εντολές μας προσφέρει και το Scratch, τις εντολές **όρισε όργανο...** και **παίξε νότα...για...χρόνους**. Η εντολή **όρισε όργανο...** επιτρέπει να επιλέξουμε το όργανο που θα παίξει τη μελωδία μέσα από μια εκπληκτική ποικιλία οργάνων που ξεκινά από τη βιόλα και την κιθάρα και φτάνει μέχρι το ξυλόφωνο, το όμποε και το πίκολο!

Η εντολή **παίξε νότα...για...χρόνους** μας επιτρέπει να επιλέξουμε ποια νότα θα παιχθεί και πόσο θα διαρκέσει η αναπαραγωγή της. Όταν πατήσουμε στο βελάκι της πρώτης παραμέτρου θα εμφανιστεί ένα πιάνο που θα μας βοηθήσει να επιλέξουμε την επιθυμητή νότα. Καθώς κινούμαστε με το ποντίκι πάνω στα πλήκτρα, παρατηρούμε ότι εμφανίζονται τα ονόματα των αντίστοιχων νोटών.



Οι ονομασίες των νोटών στα ελληνικά εμφανίζονται στην επόμενη εικόνα. Αυτό μπορεί να μας φανεί πολύ χρήσιμο!



Βάζοντας, συνεπώς, διαδοχικές εντολές **παίξε νότα...για...χρόνους** μπορούμε να αναπαράγουμε οποιαδήποτε μελωδία θελήσουμε.

Παραπάνω δόθηκε ένα μικρό παράδειγμα. Η μελωδία είναι τα κάλαντα των Χριστουγέννων. Χρησιμοποιώντας τις εντολές **όρισε ρυθμό σε...χτύπους το λεπτό**, **παίξε νότα...για...χρόνους** και **περίμενε για...χρόνους** βάζουμε το χαρακτήρα μας να παίξει τα κάλαντα. Μπορείτε να μεταφέρετε το έργο στο Scratch και να εισάγετε επιπλέον μια εντολή **όρισε όργανο...** στην αρχή

όρισε ρυθμό σε 70 χτύπους το λεπτό	ορίζουμε το ρυθμό που παίζει το τραγούδι
παίξε νότα 55 για 0.4 χρόνους	διαλέγουμε τις σωστές νότες όπως τις βλέπουμε στην παρτιτούρα και βάζουμε και το χρόνο που θα παίξουν
παίξε νότα 57 για 0.4 χρόνους	
παίξε νότα 59 για 0.4 χρόνους	η νότα κρατάει πιο πολύ και μεγαλώνουμε το χρόνο
παίξε νότα 60 για 0.8 χρόνους	ο ρυθμός πρέπει να αυξηθεί άρα τον αλλάζουμε
όρισε ρυθμό σε 90 χτύπους το λεπτό	διαλέγουμε πάλι σωστά νότες και χρόνους
παίξε νότα 59 για 0.4 χρόνους	
παίξε νότα 57 για 0.4 χρόνους	πρέπει να γυρίσουμε στον αρχικό ρυθμό του τραγουδιού
όρισε ρυθμό σε 70 χτύπους το λεπτό	διαλέγουμε πάλι σωστά νότες και χρόνους
παίξε νότα 55 για 0.8 χρόνους	
παίξε νότα 53 για 0.8 χρόνους	
παίξε νότα 52 για 1.2 χρόνους	
παίξε νότα 55 για 0.8 χρόνους	
παίξε νότα 53 για 0.4 χρόνους	
παίξε νότα 52 για 0.4 χρόνους	
παίξε νότα 50 για 0.8 χρόνους	
περίμενε για 0.2 χρόνους	έχουμε παύση

του σεναρίου ώστε μετά από δοκιμές να καταλήξετε στο όργανο που είναι το καταλληλότερο για εσάς.



[08_π03.sb]

Πρέπει να σας έχει δημιουργηθεί μια απορία: ορθώς αυξάνεται ο ρυθμός του τραγουδιού στο προηγούμενο σενάριο; Αν κάποιες νότες πρέπει να παιχτούν γρηγορότερα πρέπει να αυξήσουμε το ρυθμό; Μήπως μπορούμε να μειώσουμε τους χρόνους; Μήπως οι προγραμματιστές μας δεν έχουν κατανοήσει την έννοια του ρυθμού; Εσείς την κατανοήσατε;

8.2.3 White Stripes

Δεν χρειάζεται να έχουμε ιδιαίτερες μουσικές γνώσεις για να κάνουμε το Scratch να αναπαράγει μια απλή μελωδία όπως το Seven Nation Army των White Stripes. Τις νότες οποιουδήποτε τραγουδιού μπορούμε εύκολα να τις αναζητήσουμε στο διαδικτυο ψάχνοντας με το όνομα του κομματιού συνοδευόμενο με τη λέξη tab (από το tablature = ταμπλατούρα).

Στην επόμενη εικόνα φαίνονται οι νότες που χρειαζόμαστε σε μορφή ταμπλατούρας για κιθάρα. Τα γράμματα αναφέρονται στην ακολουθία των νοτών που συνιστούν τη μελωδία.

```
3 | --- e - e - g - - e - d - c - - - - -
2 | --- - - - - - - - - - - b - - - - -

3 | --- e - e - g - - e - d - c - d - c - - - - -
2 | --- - - - - - - - - - - h - - - - -
```

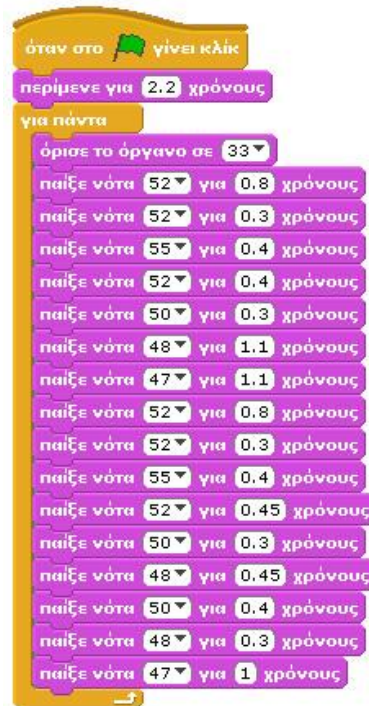
Ας δημιουργήσουμε και μια μικρή μπάντα για να παίξει το τραγούδι μας. Εισάγουμε δυο αντικείμενα, έναν ντράμερ, και έναν μπασιίστα, όπως στην παρακάτω εικόνα.



Ο ντραμίστας μας θα παίξει τύμπανο και θα κρατάει το ρυθμό ενώ ο μπασιίστας μας θα παίξει τη μελωδία. Το σενάριο του ντραμίστα που κρατά ένα σταθερό ρυθμό:



Και το σενάριο του μπασιίστα που αναπαράγει τη μελωδία:



[08_π04.sb]

Πρέπει λογικά να σας προκαλούνται πάλι ορισμένα ερωτήματα: Γιατί στο σενάριο του ντραμίστα υπάρχουν τέσσερις πανομοιότυπες εντολές που επαναλαμβάνονται για πάντα; Θα μπορούσε το συγκεκριμένο σενάριο να γίνει μικρότερο;

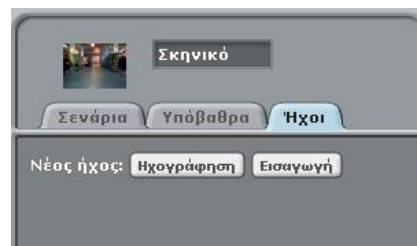


8.3 Δυο ακόμη παραδείγματα

8.3.1 Ο Μπούκλας παίζει και μουσική

Στο παράδειγμα «Δοκιμαστήριο» του κεφαλαίου 7, είχαμε 3 αντικείμενα: δυο χαρακτήρες, τον Μπούκλα και το Σορτσάκια και το Παραβάν πίσω από το οποίο άλλαζε ενδυμασίες ο Μπούκλας. Ο Μπούκλας δοκίμαζε παραδοσιακές στολές της Σκωτίας, της Ισπανίας και της Κίνας, ενώ ο φίλος του Σορτσάκιας έλεγε την άποψή του για κάθε μία από αυτές. Δε θα ήταν ωραίο να ζωντανέψουμε τη σκηνή αυτή προσθέτοντας χαρακτηριστικά μουσικά κλιπ από κάθε χώρα;

Το μόνο που θα πρέπει να προσέξουμε είναι τις χρονικές στιγμές που παίζουμε τα κομμάτια και τη διάρκειά τους, ώστε να είναι συγχρονισμένα με τις εμφανίσεις του Μπούκλα. Οι ήχοι μπορούν να αναπαραχθούν από οποιοδήποτε αντικείμενο αλλά έστω ότι εμείς αποφασίζουμε να τους εισάγουμε στην καρτέλα «Ήχοι» του σκηνικού μας όπως φαίνεται στην παρακάτω.



Αφού πατήσουμε το κουμπί «Εισαγωγή», εμφανίζεται ένα παράθυρο για την εισαγωγή ήχων μέσω του οποίου εισάγουμε 3 ηχητικά clips με ονόματα "matador.wav",

“Scottish_bagpipes.wav” και “chinese_riff.wav” από το συ-
νοδευτικό υλικό του βιβλίου. Μετά την εισαγωγή των ήχων, η
καρτέλα «Ήχοι» θα πρέπει να έχει την παρακάτω μορφή.



Μεταφερόμαστε πλέον στην καρτέλα «Σενάρια» του σκηνικού
μας. Το πρώτο πράγμα που πρέπει να κάνουμε είναι να σύρουμε
στην καρτέλα «Σενάρια» την εντολή όταν στην πράσινη ση-
μαία γίνει κλικ. Η πρώτη εμφάνιση με την οποία εμφανίζεται ο
ήρωας είναι η σκωτσέζικη, οπότε θα πρέπει να παίξουμε πρώτα
το clip «scottish_bagpipes». Αυτό θα γίνει με την εντολή παίξε
ήχο ...μέχρι τέλους της παλέτας «Ήχος». Προσοχή όμως! Δε θέ-
λουμε να αρχίσει να ακούγεται από την αρχή του προγράμμα-
τος αλλά μόνο όταν εμφανιστεί ο ήρωας μας. Γι’ αυτό χρησι-
μοποιούμε την εντολή περίμενε για ...χρόνους. Να θυμίσουμε
ότι από τη στιγμή που δεν έχουμε αλλάξει το ρυθμό, η εντολή
αυτή ισοδυναμεί με την εντολή περίμενε για...δευτερόλεπτα.
Πόσο όμως πρέπει να περιμένουμε; Ας ξαναδούμε το σενάριο
του Μπούκλα:



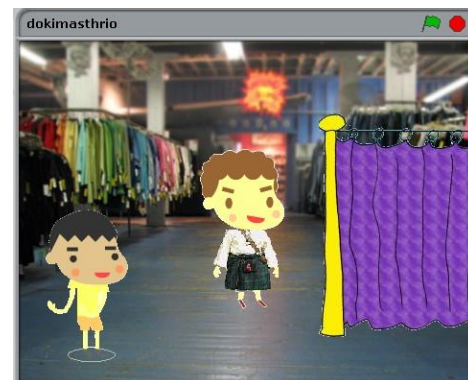
Παρατηρήστε ότι Μπούκλας ξεκινά να βγαίνει από το δοκιμα-
στήριο μετά από 1 δευτερόλεπτο αφού έχει ρωτήσει «Έτοιμος;»
ενώ στη συνέχεια κινείται για 1.5 δευτερόλεπτα μέχρι τη νέα
θέση του. Άρα κάπου ανάμεσα στο 1 και 2.5 (1+1.5) δευτερόλε-
πτο του χρονοδιαγράμματος του σεναρίου μας πρέπει να αρχί-
σει να ακούγεται ο αντίστοιχος ήχος.



Το πόσο θα περιμένουμε στη συνέχεια μέχρι να παίξουμε το ε-
πόμενο κλιπ δεν είναι τόσο απλό αφού αυτό εξαρτάται από τη
διάρκεια του προηγούμενου. Για αυτό το λόγο αξιοποιούμε και
τη διάρκεια του κάθε κλιπ όπως φαίνονται στην καρτέλα Ήχοι
για τους υπολογισμούς μας. Από τη στιγμή που έχουμε επιλέξει
την εντολή παίξε ήχο ...μέχρι τέλους η επόμενη εντολή θα εκτε-
λεστεί μετά από 6 δευτερόλεπτα που κρατάει το κλιπ matador.
Υπολογίζουμε τους υπόλοιπους χρόνους και με το σκεπτικό
αυτό καταλήγουμε στο εξής σενάριο:



Το πρόγραμμά μας είναι πια έτοιμο για δοκιμή!



[08_π05.sb]

Υπάρχουν δυο σημαντικά ερωτήματα:

Α) Ήταν σωστή η επιλογή μας να βάλουμε τους ήχους στο σκη-
νικό; Η απάντηση είναι όχι! Από τη στιγμή που οι εντολές ανα-
παραγωγής του ήχου εξαρτώνται αποκλειστικά από το σενά-
ριο του Μπούκλα θα ήταν πολύ ευκολότερο να επιλέξουμε να
τις βάλουμε στη σωστή θέση στο συγκεκριμένο σενάριο!

Β) Ήταν σωστή η επιλογή της εντολής παίξε ήχο...μέχρι τέλους
που μας ανάγκασε να μετράμε το χρόνο του κλιπ για να υπολο-
γίσουμε πότε θα αναπαραχθεί το επόμενο κλιπ; Σκεφτήκατε τι
θα συνέβαινε αν η διάρκεια του κλιπ ήταν μεγαλύτερη από το
χρόνο που ήθελε ο Μπούκλας για να βγει έξω και να ξαναμπει
στο δοκιμαστήριο. Η επιλογή μας ήταν κακή. Θα ήταν καλύτερη
η επιλογή της εντολής παίξε ήχο... και ο συνδυασμός της με την
εντολή σταμάτησε όλους τους ήχους πριν την αναπαραγωγή
του επόμενου ήχου.

Άρα υπάρχουν δυο εναλλακτικές που καλό θα ήταν να αναπτύ-
ξετε από μόνοι σας:

Α) διατηρήστε τους ήχους στο σκηνικό και αναπτύξτε το αντίστοιχο σενάριο χρησιμοποιώντας τις εντολές παιξε ήχο..., σταμάτησε όλους τους ήχους, και περίμενε για...χρόνους.

Β) εισάγετε τους ήχους στον Μπούκλα και ενσωματώστε τις εντολές αναπαραγωγής του ήχου μέσα στο δικό του σενάριο. Θα χρειαστείτε μόνο τις παιξε ήχο... και περίμενε για...χρόνους. Παρατηρήστε πόσο πιο εύκολο γίνεται το συγκεκριμένο έργο.

Είναι σημαντικό λοιπόν να κάνουμε σωστή αναγνώριση του αντικείμενου που θα αναπαράγει τους ήχους μας καθώς αυτή μπορεί να μας βοηθήσει δραματικά για τη δημιουργία του έργου μας. Άλλωστε θα έπρεπε από την αρχή να συνειδητοποιήσουμε ότι αυτός που θα έπρεπε να αναπαράγει τους ήχους ήταν ο Μπούκλας!

8.3.2 Χορεύοντας μόνος σου

Στο παράδειγμα αυτό θα δούμε πώς μπορούμε να συνδυάσουμε ήχο και κίνηση, ώστε να κάνουμε έναν χαρακτήρα να χορεύει. Πιο συγκεκριμένα, θέλουμε ο χαρακτήρας μας να χορεύει στο δωμάτιό του ενώ ακούει μουσική από το ραδιοκασετόφωνό του.

Ποια είναι τα αντικείμενα λοιπόν σε αυτό το παράδειγμα και τι ακριβώς κάνουν; Διακρίνουμε 2 αντικείμενα. Το πρώτο είναι ο μαλλιαρός χαρακτήρας μας, ο οποίος αρχικά σκέφτεται κάτι και στη συνέχεια χορεύει και το δεύτερο αντικείμενο είναι το ραδιοκασετόφωνο το οποίο παίζει μουσική και πάλλεται στο ρυθμό της. Τα αντικείμενα αυτά βρίσκονται στο δωμάτιο του χαρακτήρα μας το οποίο αποτελεί και το σκηνικό του έργου μας.

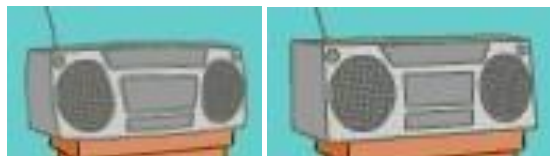


Ο πρωταγωνιστής μας θα στέκεται στο κέντρο του δωματίου του ακούγοντας μουσική από το ραδιοκασετόφωνό του και θα σκέφτεται για λίγο αν πρέπει να χορέψει ή όχι.

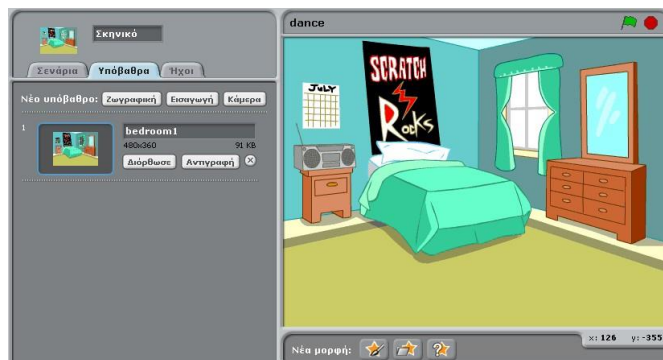


Τη στιγμή που το αποφασίζει, θα αλλάξει ενδυμασία και θα σκέφτεται «Θα χορέψω!!!» και θα αρχίζει ο χορός. Πως θα κάνουμε το χαρακτήρα μας να φαίνεται ότι χορεύει; Μπορούμε να τον κινούμε έστω και λίγο και να του αλλάζουμε ενδυμασίες διαρκώς ώστε να φαίνεται ότι χορεύει.

Το δεύτερο αντικείμενο, το ραδιοκασετόφωνο θα πρέπει να πάλλεται όσο παίζει η μουσική δηλαδή να «φουσκώνει» και να «ξεφουσκώνει» στο ρυθμό της μουσικής. Και εδώ, όπως και πριν, αυτό θα πρέπει αυτό να επαναλαμβάνεται σε όλη τη διάρκεια του έργου μας.



Αρχικά πρέπει να επιλέξουμε σκηνικό. Ακολουθώντας τις γνωστές διαδικασίες επιλέγουμε ως σκηνικό το «bedroom2» που περιέχεται στο φάκελο «indoors» στη βιβλιοθήκη σκηνικών του Scratch. Το παράδειγμά μας θα έχει τα 2 αντικείμενα που αναφέραμε: ένα χαρακτήρα που χορεύει μέσα στο δωμάτιο και το ραδιοκασετόφωνο που παίζει τη μουσική. Όπως παρατηρούμε στην επόμενη εικόνα, υπάρχει ήδη ένα ραδιοκασετόφωνο μέσα στο σκηνικό που επιλέξαμε. Μήπως υπάρχει κάποιος τρόπος να επιλέξουμε το ραδιοκασετόφωνο του σκηνικού ως νέο αντικείμενο αντί να ψάχνουμε καινούριο; Και βέβαια! Ας δούμε πώς!



Κάντε δεξί κλικ σε οποιοδήποτε σημείο του υποβάθρου (όχι όμως πάνω σε άλλο αντικείμενο) και επιλέξτε «**άρπαξε περιοχή οθόνης για νέα μορφή**».



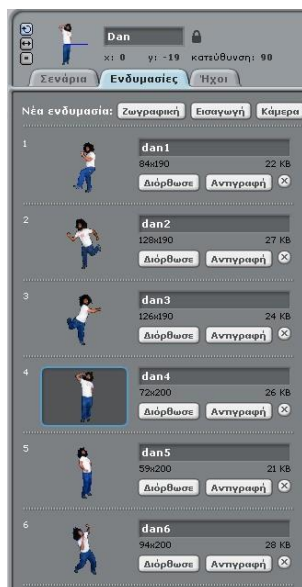
Στη συνέχεια μαρκάρετε μια ορθογώνια περιοχή που να περιέχει το ραδιοκασετόφωνο (επιδιώκουμε να είναι όσο πιο κοντά στο περίγραμμά του γίνεται).



Η περιοχή που διαλέξατε εμφανίζεται τώρα ως νέο αντικείμενο στο κέντρο της οθόνης καθώς και στη λίστα των αντικειμένων! Σύρετε το νέο αντικείμενο πάνω από το ραδιοκασετόφωνο του σκηνικού ώστε να το καλύψει. Ονομάζουμε το νέο αντικείμενο «ραδιοκασετόφωνο». Μπορούμε πλέον να εφαρμόσουμε στο νέο μας αντικείμενο διάφορα εφέ.

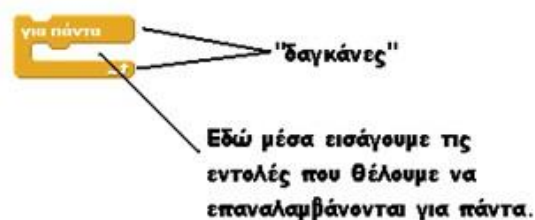
Με τον τρόπο αυτό μπορείτε να εισάγετε οποιαδήποτε φωτογραφία ως υπόβαθρο και στην συνέχεια να δημιουργήσετε αντικείμενα από τα περιεχόμενά της. Απλά κάνετε δεξί κλικ και επιλέγετε «**άρπαξε περιοχή οθόνης για νέα μορφή**». Το νέο αντικείμενο που μόλις δημιουργήσαμε «αρπάζοντάς» το από το σκηνικό, μπορεί να χρησιμοποιηθεί προς το παρόν μόνο στο τρέχον έργο μας. Αν θέλουμε να το αποθηκεύσουμε εκτός προγράμματος για να το ξαναχρησιμοποιήσουμε σε άλλο έργο, κάνουμε δεξί κλικ στο αντικείμενο (στη θέση του στη λίστα αντικειμένων) και επιλέγουμε «**εξάγαγε αυτή τη μορφή**».

Η δεύτερη μορφή στο πρόγραμμά μας είναι ο χαρακτήρας. Θα πάρουμε ως χαρακτήρα τον Dan από το φάκελο People της βιβλιοθήκης μορφών του Scratch. Θέλουμε όμως, ο Dan να χορεύει, οπότε θα χρειαστούμε διαφορετικές ενδυμασίες για κάθε χορευτική κίνηση. Για το λόγο αυτό εισάγουμε τις μορφές dan1 έως dan6 στην καρτέλα «**Ενδυμασίες**».



Ας βάλουμε τώρα μουσική στο παράδειγμά μας. Στην καρτέλα «**Σενάρια**» του σκηνικού, μετά την κλασική εντολή όταν στη

σημαία γίνει κλικ, τοποθετούμε την εντολή για πάντα που βρίσκεται στην παλέτα «**Ελεγχος**». Τι κάνει η εντολή αυτή; Όπως έχουμε ξαναπει, επαναλαμβάνει συνεχώς (για πάντα δηλαδή!) όσες εντολές βρίσκονται μέσα στις «**δαγκάνες**» της.



Τι θέλουμε εμείς να κάνουμε για πάντα όμως στο παράδειγμά μας; Μα να ακούμε μουσική! Γι αυτό και μέσα στην εντολή για πάντα τοποθετούμε την εντολή παιξε ήχο...μέχρι τέλους, ώστε να επαναλαμβάνεται συνεχώς το μουσικό κλιπ Techno1. Ο ήχος αυτός είναι διαθέσιμος στη βιβλιοθήκη ήχων του Scratch. Εσείς βέβαια μπορείτε να βάλετε όποιο τραγούδι σας αρέσει!

Μετά από τα παραπάνω, οι καρτέλες «**Σενάρια**» και «**Ηχοί**» του σκηνικού θα μοιάζουν με τις αντίστοιχες εικόνες:



Έτοιμη η μουσική! Όρα να δούμε πώς θα αντιδρούν τα άλλα αντικείμενα στη μουσική. Στην επόμενη εικόνα βλέπουμε τις εντολές για το ραδιοκασετόφωνο.



Με τη χρήση του εφέ «**μάτι ψαριού**» κάνουμε το ραδιοκασετόφωνο να πάλλεται (φουσκώνει και ξεφουσκώνει) για πάντα.

Τέλος, το σενάριο του χορευτή μας φαίνεται στην εικόνα που ακολουθεί.



Με το που γίνει λοιπόν κλικ στη σημαία, θέτουμε το x ίσο με 0 δηλαδή τοποθετούμε τον Dan στο κέντρο του σκηνικού. Αυτό το κάνουμε, επειδή κάθε φορά που τελειώνει το πρόγραμμά μας, ο Dan πιθανώς να βρίσκεται σε άλλο σημείο της σκηνής, χορευτής είναι εξάλλου! Στη συνέχεια, ο Dan «φορώντας» μια πιο «σκεπτική» ενδυμασία (dan4) αναλογίζεται για 6 δευτερόλεπτα αν πρέπει να χορέψει. Τελικά σκέφτεται «Θα χορέψω!» έχοντας φορέσει μια πιο «αποφασιστική» στάση-ενδυμασία.



Το δεύτερο μέρος του σεναρίου κάνει τον Dan να χορεύει. Αυτό το καταφέρνει αλλάζοντας ανά μικρά χρονικά διαστήματα (0.2 δευτερόλεπτα) τις ενδυμασίες του Dan και μετακινώντας τον ελαφρώς δεξιά-αριστερά στη σκηνή.

Αρχικά, θα πηγαίνει 20 σημεία δεξιά στον οριζόντιο άξονα. Στη συνέχεια, θα αλλάζει 4 φιγούρες-ενδυμασίες ανά 0.2 δευτερόλεπτα δίνοντάς μας την αίσθηση ότι χορεύει ρυθμικά, και στο τέλος θα επιστρέφει στην αρχική θέση ($x=0$) στον άξονα x . Τέλος, θα περιμένει 0.2 δευτερόλεπτα πριν αρχίσει πάλι από την αρχή να χορεύει.

Αυτό ήταν! Ο Dan και το ραδιοκασετόφωνό του είναι έτοιμοι για χορό!

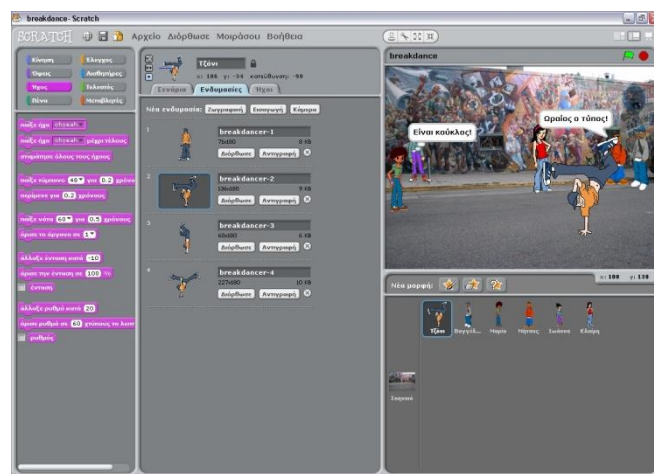
[08_π06.sb]

Θα πρέπει να σας έχουν προβληματίσει πάλι (!) δυο ερωτήματα:

A) τι θα συνέβαινε αν στο σενάριο του σκηνικού αντί να χρησιμοποιήσουμε την εντολή **παιξε ήχο...μέχρι τέλους**, είχαμε χρησιμοποιήσει την εντολή **παιξε ήχο...**; Δημιουργήστε ένα αντίστοιχο σενάριο, ακούστε τι συμβαίνει και προσπαθήστε να το εξηγήσετε.

B) πως θα μπορούσατε να δημιουργήσετε ένα πιο ενδιαφέρον σενάριο χορευτικών κινήσεων για τον χαρακτήρα σας. Δοκιμάστε να αλλάξετε χρόνους και εντολές στο αντίστοιχο σενάριο.

Για όσους θα ήθελαν να δουν ένα ακόμη πιο σύνθετο παράδειγμα με συνδυασμό μουσικής, χορού και εικόνων, υπάρχει διαθέσιμο το παράδειγμα dance.sb. Στο παράδειγμα αυτό κεντρικός χαρακτήρας είναι ένας break-dancer ο οποίος χορεύει στο ρυθμό της αγαπημένης του (ή...σας) μουσικής. Ο χορός του έχει προσελκύσει διάφορα άτομα που τον παρατηρούν, σχολιάζουν και αντιδρούν καθώς ο break-dancer χορεύει χωρίς σταματημό μέρα-νύχτα. Δείτε τα σενάρια των διαφορετικών χαρακτήρων και προσπαθήστε να τα κατανοήσετε συζητώντας με τους συμμαθητές και το δάσκαλό σας. Τροποποιήστε και εμπλουτίστε το παράδειγμα με δικές σας ιδέες!



Σύνοψη

Στο κεφάλαιο 8 είδαμε πώς μπορούμε να εισάγουμε έτοιμους ήχους, να ηχογραφήσουμε δικούς μας ήχους μέσα από το Scratch και στη συνέχεια να τους χειριστούμε προγραμματιστικά. Δεν πρέπει να ξεχνάμε ότι οι ήχοι εισάγονται είτε σε συγκεκριμένα αντικείμενα είτε στο σκηνικό και ότι η επιλογή του αντικειμένου που θα τους βάλουμε μπορεί να είναι καθοριστική για την ανάπτυξη του έργου μας.

Το Scratch μας δίνει τη δυνατότητα ακόμη και να συνθέσουμε τη δική μας μουσική χρησιμοποιώντας τα όργανα και τα τύμπανα που μας παρέχει. Κάποιες βασικές μουσικές γνώσεις είναι απαραίτητες αλλά με τη βοήθεια του βιβλίου μπορείτε να αναπαράγετε οποιαδήποτε μελωδία θελήσετε.

Έχουμε πια τις βασικές γνώσεις που αφορούν την κίνηση, τις όψεις, τη ζωγραφική, και τους ήχους. Τα αντικείμενά μας μπορούν να γίνουν ολοκληρωμένοι χαρακτήρες ιστοριών ή παιχνιδιών. Ο προγραμματισμός όμως μόλις ξεκινάει.

Ερωτήσεις

1) Ποια είναι η διαφορά της εντολής παίξε ήχο... και παίξε ήχο...μέχρι τέλους;

2) Με ποια εντολή ορίζεις την ένταση ενός ήχου;

3) Ποια εντολή θα αξιοποιούσες για να αναπαραστήσεις τη φωνή κάποιου που απομακρύνεται αργά αργά;

4) Γιατί χρησιμοποιείτε η έννοια των «χρόνων» στις εντολές της παλέτας Ήχος και όχι τα «δευτερόλεπτα»;

6) Με ποια εντολή ρυθμίζεις πόσο γρήγορα ή αργά θα παίζει η μελωδία σου;

7) Τι θα συνέβαινε αν μέσα σε μια εντολή για πάντα εισάγαμε μια εντολή παίξε ήχο... και επιλέγαμε ένα τραγούδι; Δοκιμάστε το και εξηγήστε τι συμβαίνει.

8) Σωστό ή Λάθος και εξηγήστε γιατί!

Σ-Λ Στην εντολή παίξε ήχο... μπορούμε να ακούσουμε και την δική μας φωνή

Σ-Λ Με την εντολή άλλαξε το ρυθμό κατά... δίνουμε μια συγκεκριμένη τιμή στο ρυθμό

Σ-Λ Η εντολή παίξε ήχο... συνεχίζει την αναπαραγωγή του ήχου που έχουμε επιλέξει ακόμη και αν ακολουθεί εντολή σταμάτησε όλους τους ήχους.

Δραστηριότητες

1) Φτιάξτε το δικό σας βιντεοκλιπ. Επιλέξτε τραγούδι και πρωταγωνιστές από την τάξη σας και δημιουργήστε ένα βιντεοκλιπ που ταιριάζει με τους στίχους του τραγουδιού.

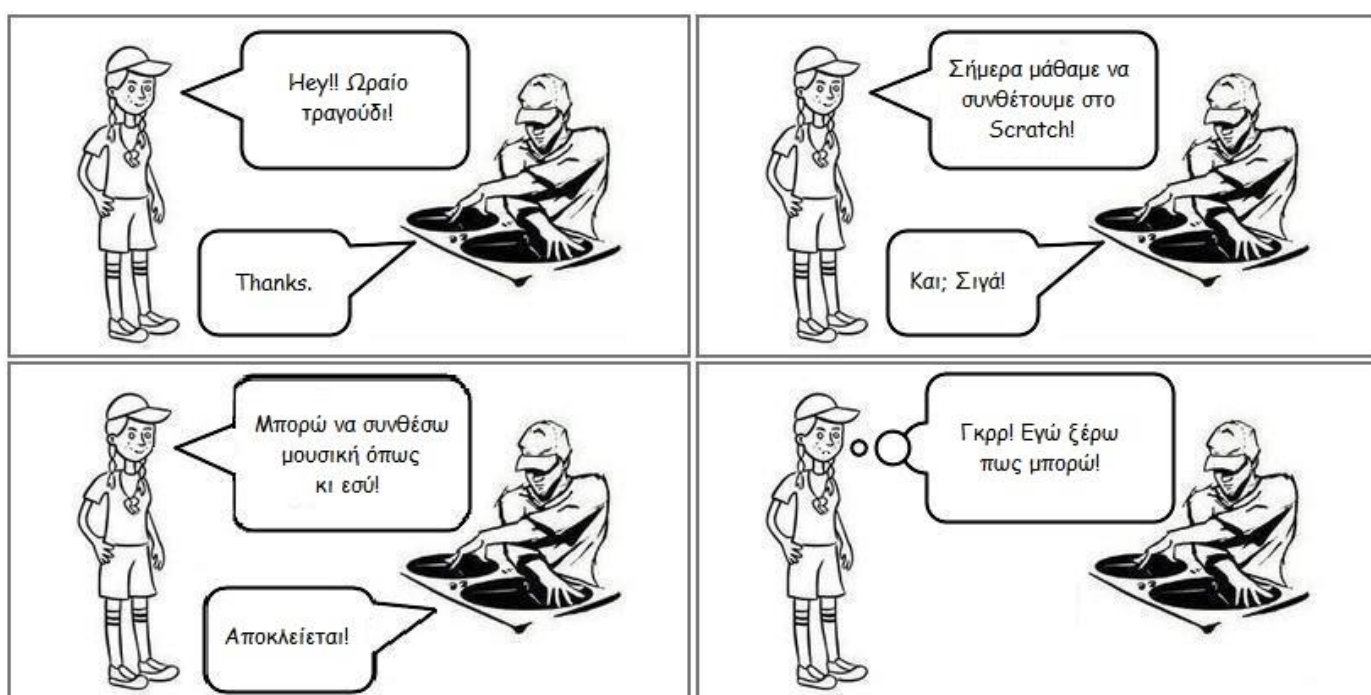
2) Εργαζόμενοι πάνω στο παράδειγμα «Δοκιμαστήριο», προσπαθήστε σε ομάδες των 2 να ηχογραφήσετε μέρος της συνομιλίας των 2 χαρακτήρων που παρουσιάζεται σε συννεφάκια. Ηχογραφήστε τη συνομιλία κομμάτι-κομμάτι και στη συνέχεια προσπαθήστε να παίξετε τις ηχογραφήσεις σας ώστε να ακούγονται οι ήρωές σας την ίδια στιγμή που εμφανίζονται τα αντίστοιχα συννεφάκια. Προσοχή! Ίσως χρειαστείτε να μειώσετε την ένταση των μουσικών κομματιών που χρησιμοποιήσαμε στο παράδειγμα ώστε να ακούγονται καθαρά οι δικές σας ηχογραφήσεις.

3) Παρακάτω δίνεται η παρτιτούρα από το τραγούδι "twinkle twinkle little star". Προσπαθήστε να δημιουργήσετε τη μελωδία στο Scratch. Μην ξεχάσετε να συμβουλευέστε την αντίστοιχη εικόνα του κεφαλαίου για να επιλέξετε σωστά τις νότες.

Twin-kle, twin-kle, lit-tle star, how I won-der what you are!

Up a-bove the world so high, like a dia-mond in the sky.

Twin-kle, twin-kle, lit-tle star, how I won-der what you are!



Κεφάλαιο 9: Αλληλεπίδραση

Σε αυτό το κεφάλαιο:

9.1 Εισαγωγή στην αλληλεπίδραση

9.2 Γεγονότα

9.3 Εντολή «Ρώτησε και Περίμενε»

«Το οικοσύστημα αυτοπροσαρμόζεται και για κάθε δράση υπάρχει και μία αντίδραση»
(Steve Mils)



9.1 Γεγονότα

Τα έργα που έχουμε αναπτύξει έως τώρα τρέχουν ένα σενάριο και σταματούν. Τα αντικείμενά μας αλλάζουν θέση και ενδυμασίες, παίζουν διαφορετικούς ήχους και ζωγραφίζουν διάφορα περίεργα σχήματα. Εμείς, όμως, απλά τα παρακολουθούμε καθώς εκτελούνται! Όσες φορές και να τρέξουν, το αποτέλεσμα είναι το ίδιο. Δεν μπορούμε να χειριστούμε τους χαρακτήρες, δεν μπορούμε να χρησιμοποιήσουμε ποντίκι και πληκτρολόγιο για να αλληλεπιδράσουμε με την εφαρμογή! Αναρωτηθήκατε πως θα κινήσουμε ένα αυτοκίνητο ή τον θρυλικό πλέον σε αυτό το βιβλίο πακ-μαν; Αναρωτηθήκατε πως ο χρήστης θα «αλληλεπιδρά» με το έργο σας;

Στο κεφάλαιο αυτό θα εξετάσουμε δυο τρόπους αλληλεπίδρασης με τα έργα μας: τα γεγονότα και τις ερωταποκρίσεις.

Τι συμβαίνει συνήθως στο αυτοκινητάκι μας σε ένα παιχνίδι ράλι όταν πατάμε το δεξί βελάκι; Το αυτοκίνητο στρίβει προς τα δεξιά.

Τι συμβαίνει όταν πατάμε διπλό κλικ πάνω σε ένα αρχείο στην επιφάνεια εργασίας του υπολογιστή μας; Το σύστημά μας εκτελεί ή ανοίγει το συγκεκριμένο αρχείο.

Τι συμβαίνει όταν πατάμε το κουμπί Bold στον επεξεργαστή κειμένου; Το πρόγραμμα κάνει πιο έντονα τα επιλεγμένα γράμματα στο κείμενο που επεξεργαζόμαστε.

Τι συμβαίνει σε ένα παιχνίδι δράσης όταν κινούμε το ποντίκι μας προς τα αριστερά; Στρίβει ο πρωταγωνιστής μας προς τα αριστερά και ενημερώνεται η οθόνη με τη νέα θέα του ήρωα μας.

Όλα τα προηγούμενα παραδείγματα βασίζονται σε συμπεριφορές που επιδεικνύουμε εμείς προς το σύστημα και τις οποίες το σύστημα αντιλαμβάνεται και αντιδρά. Στη γλώσσα του προγραμματισμού, οι συμπεριφορές αυτές αποκαλούνται «γεγονότα»: Ο χρήστης προκαλεί «γεγονότα» που αντιλαμβάνεται το σύστημα και ανάλογα με το γεγονός εκτελεί ένα συγκεκριμένο σχετικό σενάριο.

Τι γεγονότα μπορούμε να προκαλέσουμε στις εφαρμογές μας γενικά; Απλό και διπλό κλικ στο ποντίκι, αριστερό και δεξί κλικ στο ποντίκι, το πάτημα ενός πλήκτρου του πληκτρολογίου, το

πάτημα συνδυασμού πλήκτρων του πληκτρολογίου κτλ. Αυτή είναι η γλώσσα με την οποία ο χρήστης διαχειρίζεται τον υπολογιστή του, αλληλεπιδρά με αυτόν αλλά και η γλώσσα με την οποία μας καταλαβαίνει ο υπολογιστής. Αυτή είναι και η γλώσσα με την οποία δημιουργούμε εφαρμογές στον υπολογιστή.

Ακολουθώντας το μοντέλο του «γεγονο-στραφή» προγραμματισμού τα έργα μας αποκτούν μια απρόβλεπτη εκτέλεση που εξαρτάται από την είσοδο του χρήστη.

Ένα γεγονός που έχουμε συναντήσει στο Scratch αρκετές φορές είναι η εντολή: όταν στην πρά-

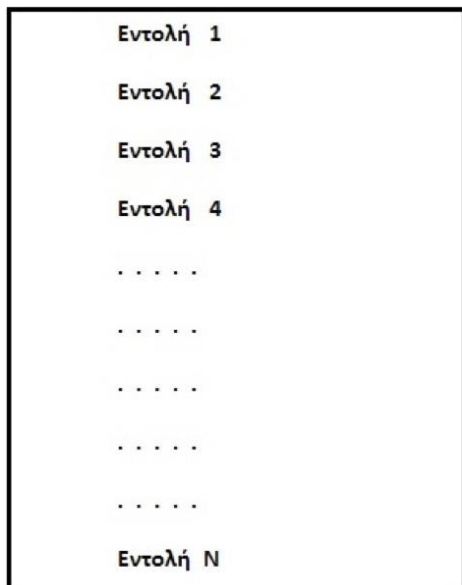
σινη σημαία γίνει κλικ. Η εντολή



αυτή ουσιαστικά προχωρά στην εκτέλεση των εντολών που την ακολουθούν, όταν ο χρήστης πατήσει στην πράσινη σημαία, δηλαδή όταν προκαλέσει το αντίστοιχο γεγονός. Όταν όντως το κουμπί πατηθεί, θα δημιουργηθεί ένα γεγονός, το Scratch θα το καταλάβει και θα επιτρέψει την εκτέλεση όλων των σεναρίων όλων των αντικειμένων που ξεκινούν από τη συγκεκριμένη εντολή.

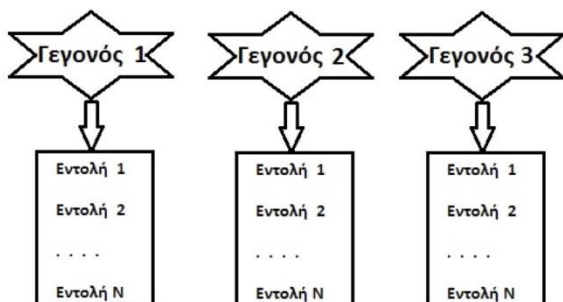
Η έννοια είναι ιδιαίτερα σημαντική για τη σχεδίαση αλληλεπιδραστικών εφαρμογών και για αυτό ας τη δούμε ακόμη περισσότερο. Μέχρι στιγμής στο βιβλίο έχουμε δει έργα τα οποία εκτελούν σενάρια ολοκληρωμένα από την αρχή ως το τέλος. Ο

τρόπος αυτός εκτέλεσης παρουσιάζεται σχεδιαγραμματικά στην επόμενη εικόνα.



Σε αυτή την προσέγγιση, στόχος του προγραμματιστή είναι ο σχεδιασμός και η υλοποίηση ενός προγράμματος το οποίο θα πετυχαίνει έναν προδιαγεγραμμένο στόχο, που μπορεί να είναι η κίνηση ενός αντικειμένου σε κάποια θέση ή κάποιοι μαθηματικοί υπολογισμοί. Ο χρήστης δεν έχει τρόπο να αλλάξει την ροή αυτής της εκτέλεσης του προγράμματος. Ξέρουμε από την αρχή, ποια θα είναι η λειτουργία του προγράμματος.

Στην ακριβώς αντίθετη περίπτωση βρίσκεται το μοντέλο του γεγονοστραφούς προγραμματισμού, όπου οι εντολές εκτελούνται ανάλογα με τα γεγονότα που προκαλεί ο χρήστης. Στην επόμενη εικόνα βλέπουμε το μοντέλο ενός τέτοιου προγράμματος.



Σε αυτό το μοντέλο προγραμματισμού βασική ιδέα είναι ότι σχεδιάζουμε την αντίδραση σε μια δράση. Στα συγκεκριμένα έργα δε γνωρίζουμε από την αρχή ποιο θα είναι το αποτέλεσμα τους, καθώς κάθε φορά μπορεί να προκαλούνται διαφορετικά γεγονότα από το χρήστη και έτσι να εκτελούνται διαφορετικά τμήματα του προγράμματος μας.

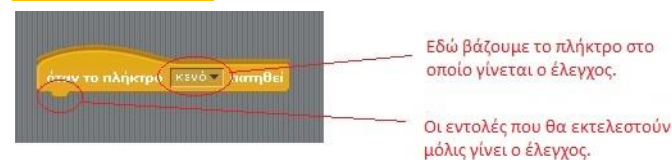
Αυτό, όμως, δε σημαίνει ότι το δεύτερο μοντέλο δεν ενσωματώνει το πρώτο. Κάθε άλλο. Ουσιαστικά όλα τα προγράμματα έχουν κάποια σενάρια που εκτελούνται μόλις γίνει η έναρξη της εκτέλεσής τους και μια σειρά από άλλα σενάρια που εκτελούνται βάσει των γεγονότων που προκαλούν οι χρήστες.

Θα αναρωτιέστε πως όλα αυτά που είπαμε μπορούμε να τα συνδέσουμε με το Scratch. Πώς δηλαδή μπορούμε εμείς οι προγραμματιστές του Scratch να φτιάξουμε ένα πρόγραμμα το οποίο θα αντιλαμβάνεται γεγονότα και θα παράγει συγκεκριμένες αντιδράσεις; Όπως συζητήσαμε και παραπάνω το έχουμε ήδη κάνει με την εντολή όταν στην πράσινη σημαία γίνει κλικ. Ας δούμε όμως και πως μπορούμε να χρησιμοποιήσουμε το πληκτρολόγιό μας για να χειριστούμε τα αντικείμενά μας.

9.1.1 Εντολή «Όταν το πλήκτρο ... πατηθεί»

Τι λειτουργίες πραγματοποιούν τα πλήκτρα ενός πληκτρολογίου; Σε ένα επεξεργαστή κειμένου, τυπώνονται τα αντίστοιχα γράμματα, ενώ σε ένα παιχνίδι ελέγχουμε τη θέση και την κίνηση των χαρακτήρων μας. Το κάθε πρόγραμμα, δηλαδή, συνδέει το γεγονός του πατήματος ενός πλήκτρου με διαφορετικό σενάριο εντολών.

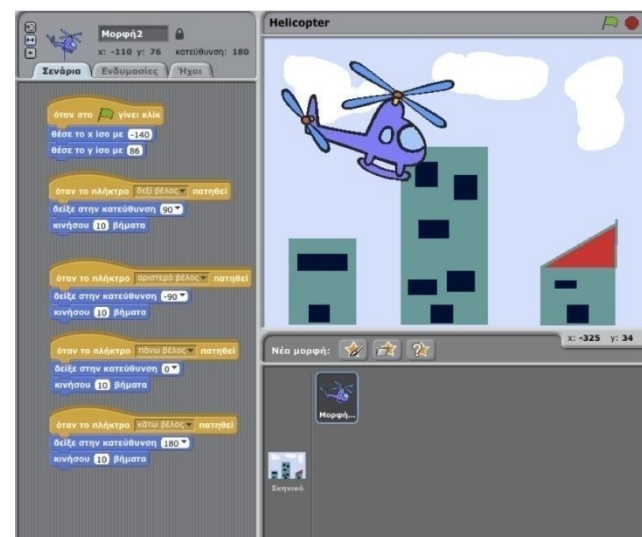
Η εντολή που μας επιτρέπει να προσδιορίσουμε διάφορες συμπεριφορές στα πλήκτρα μας είναι η όταν το πλήκτρο... πατηθεί.



Παρατηρείτε ότι το σχήμα της μοιάζει με αυτό της εντολής όταν στο κουμπί πράσινη σημαία γίνει κλικ; Όλες οι εντολές που στο πάνω μέρος τους περιέχουν αυτήν την ιδιαίτερη καμπύλη μοιράζονται δυο χαρακτηριστικά:

- α) δεν μπορούν να ενσωματωθούν στο εσωτερικό σεναρίων, π.χ. δεν μπορούν να ακολουθούν μια εντολή κινήσου...βήματα κάτι το οποίο επιβεβαιώνεται και από την έλλειψη αντίστοιχης υποδοχής στο πάνω μέρος τους,
- β) περιέχουν δικά τους σενάρια τα οποία εκτελούνται όταν προκληθούν τα γεγονότα που περιγράφουν.

Ας δούμε με ένα απλό έργο την καινούργια εντολή. Στόχος μας στο παράδειγμα είναι να κινούμε ένα ελικόπτερο με τη χρήση των πλήκτρων-βελάκια του πληκτρολογίου. Το πρόβλημά μας έχει 1 αντικείμενο και το σκηνικό του, όπως φαίνεται στην επόμενη εικόνα.



Η επιθυμητή συμπεριφορά του ελικοπτέρου είναι η εξής:

- ✓ Όταν πατάμε το πάνω πλήκτρο, θέλουμε το ελικόπτερο να αλλάζει κατεύθυνση προς τα πάνω και να κινείται προς τα πάνω.
- ✓ Όταν πατάμε το κάτω πλήκτρο, θέλουμε το ελικόπτερο να αλλάζει κατεύθυνση προς τα κάτω και να κινείται προς τα κάτω.
- ✓ Όταν πατάμε το δεξί πλήκτρο, θέλουμε το ελικόπτερο να αλλάζει κατεύθυνση προς τα δεξιά και να κινείται δεξιά.
- ✓ Όταν πατάμε το αριστερό πλήκτρο, θέλουμε το ελικόπτερο να αλλάζει κατεύθυνση προς τα αριστερά και να κινείται αριστερά.

Η εντολή μας, όταν το πλήκτρο... πατηθεί, έχει τη μορφή ενός ελέγχου. Δηλαδή όταν συμβεί ένα συγκεκριμένο γεγονός στο πρόγραμμά μας, τότε θα εκτελεστούν οι εντολές που ακολουθούν την αντίστοιχη εντολή ελέγχου. Στο κομμάτι της εντολής το πλήκτρο...πατηθεί πρέπει να προσδιορίσουμε ένα συγκεκριμένο πλήκτρο από το πληκτρολόγιο από την αντίστοιχη λίστα.

Σε σχέση με την κίνηση του ελικοπτέρου: όπως είδαμε στο κεφάλαιο 5, υπάρχουν τρεις διαφορετικοί τρόποι κίνησης. Εμείς θέλουμε κάθε φορά που πατάμε ένα βελάκι να πραγματοποιείται μια μικρή μετακίνηση του αντικειμένου μας, με ταυτόχρονη αλλαγή της κατεύθυνσής του. Συνεπώς, η καλύτερη επιλογή κίνησης είναι ο συνδυασμός των εντολών δείξε στην κατεύθυνση... και κινήσου...βήματα. Εμείς θέλουμε π.χ. όταν πατάμε το δεξί βελάκι, το ελικόπτερο να δείχνει στην κατεύθυνση 90° και να κινείται 10 βήματα. Άρα μέσα στο σενάριο του ελικοπτέρου εισάγουμε τις εντολές:



Με αντίστοιχο τρόπο δημιουργούμε και τα σενάρια για τα υπόλοιπα πλήκτρα όπως φαίνεται παρακάτω.



[09_π01.sb]

Το σενάριο του ελικοπτέρου τώρα περιέχει 5 σενάρια για 5 γεγονότα: ένα σενάριο για το γεγονός όταν πατήσουμε την πρώτη σημαία (όπου αρχικοποιούμε τη θέση του ελικοπτέρου) και 4 σενάρια για τα γεγονότα όταν το πλήκτρο...πατηθεί.

Τρέξτε το έργο σας και δείτε πως μπορείτε να μετακινείτε το ελικόπτερο. Είμαστε σαφώς πιο κοντά στη δημιουργία παιχνιδιών!

Τι άλλο θα μπορούσαμε να κάνουμε; Πραγματικά ότι σας περνάει από το μυαλό! Για παράδειγμα θέλουμε όταν ο χρήστης πατήσει «το κενό» να αλλάζει το υπόβαθρο του έργου μας. Ας το υλοποιήσουμε.

Δημιουργούμε με τη βοήθεια της ζωγραφικής 3 διαφορετικά υπόβαθρα.



Το σενάριο που μας επιτρέπει να αλλάζουμε πίστα για τη κίνηση του ελικοπτέρου μας εκτελείται όταν πατήσουμε το πλήκτρο «κενό»:

όταν το πλήκτρο ΚΕΝΟ πατηθεί

επόμενο υπόβαθρο

Όπως παρατηρούμε, κάνουμε πάλι χρήση της εντολής όταν το πλήκτρο... πατηθεί, αυτή τη φορά σε συνδυασμό με την εντολή Επόμενο υπόβαθρο από την παλέτα Όψεις.

Δεν είναι απίστευτο το πόσο μικρά είναι αυτά τα σενάρια και πόσο ενδιαφέροντα μας φαίνονται;

Στο κεφάλαιο 8 είδαμε πως μπορούμε να δημιουργήσουμε μια μελωδία χρησιμοποιώντας το Scratch. Ας δούμε τώρα πως θα μπορούσαμε με τη χρήση της εντολής όταν το πλήκτρο... πατηθεί να επεκτείνουμε τα παραδείγματα που είδαμε, με σκοπό να παίζουμε εμείς το ρόλο του πιανίστα. Συγκεκριμένα θα υλοποιήσουμε ένα πρόγραμμα το οποίο θα περιέχει ένα μόνο αντικείμενο, το πιάνο.



Η δημιουργία του πιάνου στο Scratch είναι κάτι ανάλογο με το παιχνίδι του ελικοπτερού, μόνο που τώρα θα πρέπει να αντιστοιχίσουμε σε κάθε πλήκτρο του πληκτρολογίου μια νότα του πιάνου.



Απλά, λοιπόν, συνδυάζουμε την εντολή όταν το πλήκτρο ... πατηθεί με την εντολή παίξε νότα ... για ... χρόνους από την παλέτα Ήχος. Με αυτόν τον τρόπο πατώντας ένα πλήκτρο στο

πληκτρολόγιο μας, το πρόγραμμα παίζει τη νότα που έχει συσχετιστεί με το συγκεκριμένο αυτό πλήκτρο.

όταν το πλήκτρο 3 πατηθεί

παίξε νότα 48 για 0.5 χρόνους

Ας ρίξουμε μια ματιά σε όλες τις εντολές του προγράμματος μας.



[09_π02.sb]

Ας σκεφτούμε, επίσης, το παιχνίδι Pacman. Το αντικείμενο Pacman μετακινείται μέσα στις πίστες με σκοπό να μαζεύει φρουτάκια και να αποφεύγει τους κακούς. Μπορούμε να υλοποιήσουμε την κίνηση του Pacman με τη χρήση της όταν το πλήκτρο...πατηθεί για 4 φορές, μία για κάθε βελάκι.

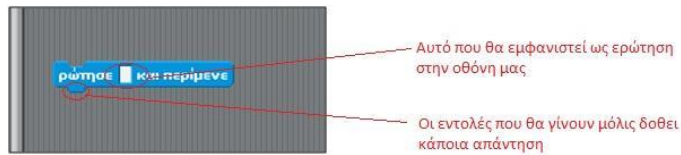
Μια εύλογη ερώτηση είναι, το τι συμβαίνει αν έχουμε πολλά αντικείμενα σε ένα έργο. Μπορούν όλα να «παρακολουθούν» τα ίδια γεγονότα ή αν αναθέσουμε το πλήκτρο «πάνω βελάκι» στο ελικοπτερό, δεν μπορούμε να το αναθέσουμε σε άλλο αντικείμενο;

Όλα τα αντικείμενα μπορούν να παρακολουθούν όλα τα γεγονότα ανεξάρτητα της χρήσης τους. Για να το επιβεβαιώσετε, εισάγετε ένα δεύτερο αντικείμενο στο έργο με το ελικοπτεράκι και αντιγράψτε τα σενάρια του πρώτου αντικειμένου στο δεύτερο. Τι παρατηρείτε; Ότι και τα δυο αντικείμενα αντιλαμβάνονται τα ίδια γεγονότα και στη συγκεκριμένη περίπτωση αντιδρούν με τον ίδιο τρόπο, δηλαδή μετακινούνται μαζί.

Άρα, όταν σχεδιάζουμε ένα έργο στο Scratch, πρέπει να διακρίνουμε σε ποια γεγονότα θέλουμε να αντιδράσει κάθε αντικείμενο και ποιες συμπεριφορές θέλουμε να επιδείξει το αντικείμενο ως αντίδραση στα γεγονότα αυτά. Η αντίδραση των διαφορετικών αντικειμένων στο ίδιο γεγονός (π.χ. στο πάτημα ενός συγκεκριμένου πλήκτρου) εκτελείτε παράλληλα, δηλαδή όλα τα αντικείμενα αντιδρούν ταυτόχρονα.

9.2 Εντολή «Ρώτησε και Περιμένε»

Η δεύτερη εντολή, που επιτρέπει στο έργο μας να αλληλεπιδράσει με το χρήστη, είναι η **ρώτησε...και περιμένε** που βρίσκεται στην παλέτα **Αισθητήρες**. Η μορφή της παρουσιάζεται παρακάτω και είναι πολύ απλή στη χρήση της:



Όπως προκύπτει και από το κείμενο που περιέχει, η συγκεκριμένη εντολή παρουσιάζει μια ερώτηση στο χρήστη και σταματά την εκτέλεση του σεναρίου μέχρι να απαντήσει ο χρήστης. Μόλις ο χρήστης εισάγει απάντηση στο έργο μας, το σενάριο συνεχίζει να εκτελείται.

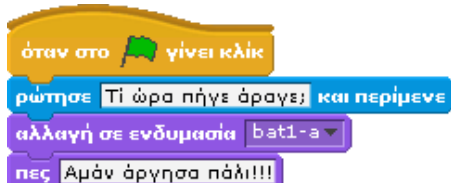
Ας γνωρίσουμε την εντολή με ένα παράδειγμα. Στο έργο μας θα έχουμε δύο αντικείμενα, τον αργοπορημένο Κόμη και το σκηνικό. Οι ενέργειες του Κόμη είναι οι εξής: θα ρωτά το χρήστη τι ώρα είναι και αφού θα λάβει απάντηση, θα λέει «αμάν, άργησα πάλι», θα αλλάζει ενδυμασία και θα αποχωρεί από το σκηνικό.



Πηγαίνουμε στο σενάριο του κόμη και σύρουμε την εντολή **όταν στην πράσινη σημαία γίνει κλικ** από την παλέτα του Ελέγχου. Συνεχίζουμε με την εντολή **ρώτησε ... και περιμένε**, στην οποία συμπληρώνουμε την ερώτηση που θέλουμε να εμφανιστεί από το χαρακτήρα μας.



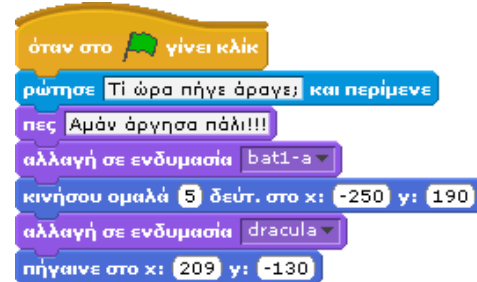
Στη συνέχεια χρησιμοποιούμε τις εντολές **πες...** και **αλλαγή σε ενδυμασία...** από την παλέτα **Όψεις**:



Προσθέτουμε την εντολή **κινήσου ομαλά...δεύτ. στο x,y** από την παλέτα **Κίνηση** έτσι ώστε η νυχτερίδα μας να φύγει εκτός της οθόνης του Scratch:



Τέλος, προσθέτουμε δύο ακόμα εντολές, έτσι ώστε όταν το πρόγραμμα τελειώνει να επανατοποθετείται ο κόμης στην αρχική του θέση και με την κατάλληλη μορφή (αυτό πρέπει να βρίσκεται στο τέλος ή στην αρχή του σεναρίου μας):

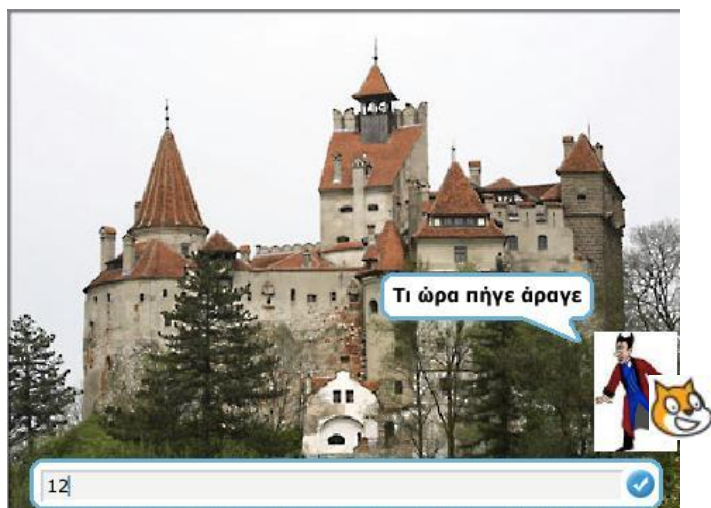


[09_π03.sb]

Πατώντας την πράσινη σημαία για να ξεκινήσει η εκτέλεση βλέπουμε το εξής: η χρήση της εντολής **ρώτησε ... και περιμένε** έχει ως αποτέλεσμα να εμφανιστεί το συγκριμένο ερώτημα ως μπαλονάκι πάνω από τον χαρακτήρα μας και επιπλέον στο κάτω μέρος της οθόνης να εμφανιστεί ένα πεδίο στο οποίο ο χρήστης πρέπει να δώσει μια απάντηση! Πρέπει να σημειώσουμε ότι ο Κόμης είναι πολύ πεισματάρης, και δεν θα μας αφήσει σε ησυχία μέχρι να του δώσουμε μια απάντηση. Δηλαδή το έργο μας δεν θα προχωρήσει στην εκτέλεση των υπόλοιπων εντολών, αν δε λάβει μια δική μας απάντηση! Βλέπουμε δηλαδή ότι η εντολή **ρώτησε ... και περιμένε** "σπάει" το πρόγραμμά μας σε δύο μέρη, περιμένοντας μια δική μας ενέργεια για να μπορέσει να συνεχίσει την εκτέλεση του.



Ας δώσουμε λοιπόν στον Κόμη μας μια απάντηση, ας πούμε ότι η ώρα πήγε 12, όπως βλέπουμε και στην επόμενη εικόνα:



Πληκτρολογώντας «12» στο πεδίο κειμένου και πατώντας enter βλέπουμε τον Κόμη να μεταμορφώνεται σε νυχτερίδα, και να φεύγει βιαστικά και πάνω από όλα, με στυλ...:



Ας δούμε ακόμη ένα παράδειγμα, στο οποίο το σκηνικό μας θα είναι η αυλή ενός σχολείου και θα υπάρχει ως αντικείμενο ένα κορίτσι. Το κορίτσι θα μας κάνει ερωτήσεις και εμείς με τη σειρά μας θα απαντάμε δίνοντας συνέχεια στο διάλογο.



Θα έχουμε μια σύντομη και ευγενική συνομιλία με τη φίλη μας. Θα ξεκινήσουμε με την απαραίτητη εντολή εκκίνησης:



Η εντολή ρώτησε... και περίμενε θα χρησιμοποιηθεί έτσι ώστε το κορίτσι να ρωτήσει το όνομά μας.

ρώτησε Πώς σε λένε; και περίμενε

Η απάντηση από το κορίτσακι.

πες Γειά σου! για 2 δευτερόλεπτα

πες Εμένα με λένε Ελένη! για 2 δευτερόλεπτα

Ακόμη μια ερώτηση, και τέλος της συζήτησης από την ίδια.

ρώτησε Τι κάνεις; και περίμενε

πες Ενδιαφέρον.. για 2 δευτερόλεπτα

πες Εγώ είμαι πολύ καλά!!! για 2 δευτερόλεπτα

[09_π04.sb]

Ας δούμε τώρα το παράδειγμα μας σε λειτουργία. Η φίλη μας θα ξεκινήσει τη συνομιλία!



Ας απαντήσουμε (δεν ξεχνάμε στο τέλος να πατήσουμε είτε enter είτε το χαρακτηριστικό κουμπί που βρίσκεται στα δεξιά του πεδίου κειμένου)



Τώρα θα λάβουμε μια απάντηση.



Το κοριτσάκι θα μας κάνει άλλη μια ερώτηση!



Και πάλι το πρόγραμμα περιμένει την είσοδο από το πληκτρολόγιο.



Απαντάμε με ευγένεια. Ο χαρακτήρας ανταποκρίνεται με ακόμα δυο φράσεις.



Και δεν μπορούμε δηλαδή να χρησιμοποιήσουμε με κάποιο τρόπο τις τιμές που δίνει ο χρήστης στο Scratch; Προφανώς και μπορούμε να τις χρησιμοποιήσουμε. Το Scratch αποθηκεύει την απάντηση του χρήστη στη μεταβλητή απάντηση που βρίσκεται

στην παλέτα Αισθητήρες. Παρατηρήστε ότι το σχήμα της συγκεκριμένης μεταβλητής δεν περιέχει ούτε εσοχές, ούτε εξοχές, δηλαδή δεν μπορεί να εισαχθεί ως εντολή μέσα σε ένα σενάριο. Αντίθετα, μπορεί να εισαχθεί μέσα στα λευκά κουτάκια των εντολών π.χ. μπορεί να εισαχθεί μέσα σε μια εντολή πες...για...δευτερόλεπτα. Δείτε το παρακάτω παράδειγμα:



Σε αυτό το παράδειγμα, το αντικείμενο ζητά από το χρήστη να προσδιορίσει το όνομά του και στη συνέχεια το αντικείμενο αναφέρει το όνομα που του έδωσε ο χρήστης και σχολιάζει ότι είναι ωραίο. Με τον τρόπο αυτό, οι διάλογοί μας μπορούν να γίνουν πιο ρεαλιστικοί αφού ο χαρακτήρας μας φαίνεται να «ακούει» το χρήστη. Ας δούμε μια προέκταση του προηγούμενου σεναρίου:



[09_π05.sb]

Στη συνέχεια του διαλόγου, το αντικείμενό μας ρωτάει το χρήστη για το που γεννήθηκε. Μετά την απάντηση του χρήστη, το αντικείμενο αναφέρει την απάντηση του χρήστη ενωμένη με 4 θαυμαστικά. Ο τελεστής ένωσε μας δίνει τη δυνατότητα να ενώσουμε δυο εκφράσεις και να τις παρουσιάσουμε ως μια. Αν ο χρήστης έγραψε ότι γεννήθηκε στην «Καβάλα», τότε ο χαρακτήρας μας θα έλεγε διαδοχικά:



Προσπαθήστε να δημιουργήσετε αντίστοιχους αληθοφανείς διαλόγους που θα εντυπωσιάσουν τους συμμαθητές σας.

Περίληψη

Ανακεφαλαιώνοντας, στο συγκεκριμένο κεφάλαιο καταφέραμε να σπάσουμε τη «μονοτονία» των έργων μας, διερευνώντας δυο τρόπους με τους οποίους ο χρήστης μπορεί να αλληλεπιδράσει με την εφαρμογή. Ο πρώτος τρόπος, τα γεγονότα, θα είναι κομβικός για τα επόμενα κεφάλαια καθώς αναφέρεται σε ένα γενικότερο τρόπο προσέγγισης της αλληλεπίδρασης του

χρήστη με το σύστημα. Ο χρήστης παράγει γεγονότα και το έργο μας αντιδρά στα γεγονότα αυτά. Ως προγραμματιστές πρέπει να προσδιορίζουμε σε ποια από τα γεγονότα που μπορεί να προκαλέσει ο χρήστης, τα αντικείμενά μας θα αντιδρούν και πως ακριβώς θα αντιδρούν, ποιο θα είναι δηλαδή το αντίστοιχο σενάριο. Ο δεύτερος τρόπος αλληλεπίδρασης, η εντολή ρώτησε ... και περίμενε, είναι ιδιαίτερα απλός και επιτρέπει τη δημιουργία ερωταποκρίσεων μεταξύ του έργου μας και του χρήστη. Τα αντικείμενά μας άρχισαν να αποκτούν περισσότερο «ευφυή» συμπεριφορά καθώς αντιδρούν στις διαφορετικές επιλογές μας. Είμαστε όμως ακόμη στην αρχή, από το επόμενο κεφάλαιο αρχίζουν να ξεδιπλώνονται οι πραγματικές δυνατότητες του προγραμματισμού.

ποιος χαρακτήρας του έργου θα εμφανιστεί, τι θα πει και τι θα κάνει με το πάτημα διαφορετικών πλήκτρων του πληκτρολογίου! Χρησιμοποιείστε ως βάση σας σενάριο από μια γνωστή σας θεατρική παράσταση, ή μια ταινία που παρακολουθήσατε πρόσφατα.

Ερωτήσεις

- 1) Να καταγραφούν τα γεγονότα που προκαλεί ο χρήστης και γίνονται αντιληπτά και αξιοποιήσιμα από το παιχνίδι Pacman.
- 2) Ποιά είναι η ιδιαιτερότητα του γεγονοστραφή προγραμματισμού;
- 3) Συζητήστε με τους συμμαθητές σας σχετικά με τον αν θα ήταν δυνατή η υλοποίηση του προγράμματος «Ελικόπτερο» χωρίς την χρήση της εντολής Όταν το πλήκτρο...πατηθεί.
- 5) Ποιό είναι το βασικό χαρακτηριστικό της ρώτησε.... και περίμενε.

6) Σε ποια προγράμματα θα μπορούσε να είναι χρήσιμη η εντολή ρώτησε.... και περίμενε.

7) Βρείτε τις επόμενες λέξεις.

- 1) _ _ Γ _ _ _ _ _
- 2) _ _ _ _ E _ _ _ _ _
- 3) _ _ _ Γ _ _ _
- 4) _ _ _ _ _ _ O _ _ _ _ _
- 5) _ _ _ _ _ _ _ _ N _
- 6) _ _ _ _ _ O _ _ _

1. Προγραμματισμός που βασίζεται σε γεγονότα.
2. Προγραμματισμός που βασίζεται σε αντικείμενα.
3. Παλέτα στην οποία βρίσκεται η εντολή όταν το πλήκτρο...πατηθεί (μια λέξη).
4. Εντολή που αντιλαμβάνεται γεγονότα που προκαλούνται από το πάτημα των πλήκτρων.
5. Απαραίτητη εντολή για να ξεκινήσει μια συζήτηση.
6. Μονάδα εισόδου που προκαλεί γεγονότα.

Δραστηριότητες

- 1) Υλοποιείτε ένα εικονικό συνθεσάιζερ, στο οποίο θα παίζετε εσείς οι ίδιοι πατώντας πλήκτρα στο πληκτρολόγιο. Θα πρέπει να μπορείτε να παίζετε ταυτόχρονα μια μελωδία με ένα πίκολο και να δίνεται ρυθμός με ένα πιατίνι και ένα μπάσο.
- 2) Υλοποιείτε με τη βοήθεια της εντολής ρώτησε ... και περίμενε μια εικονική συναλλαγή με το περίπτερο της γειτονιάς σας.
- 3) Σχεδιάστε και υλοποιείτε μια θεατρική παράσταση, στην οποία ο ίδιος ο χρήστης θα είναι ο σκηνοθέτης, επιλέγοντας



Κεφάλαιο 10: Η επανάληψη

Σε αυτό το κεφάλαιο:

10.1 Εισαγωγή στην επανάληψη

10.2 Για πάντα

10.3 Η εντολή Επανάλαβε Χ

10.4 Παραδείγματα

«Επανάληψη μήτηρ πάσης μαθήσεως»
ρητό



10.1 Εισαγωγή στην επανάληψη

Στο κεφάλαιο αυτό ήρθε η ώρα να μελετήσουμε την επανάληψη στον προγραμματισμό λίγο πιο διεξοδικά! Έχετε ήδη χρησιμοποιήσει, χωρίς πολλές επεξηγήσεις, σε προηγούμενα κεφάλαια τις εντολές **για πάντα** και **επανάλαβε Χ**.. Πρέπει πλέον να κατανοήσετε τη γενικότερη χρησιμότητα της επανάληψης στον προγραμματισμό ώστε να είστε σε θέση να την εφαρμόζεται σε σύνθετα παραδείγματα.

Τι εννοούμε όταν λέμε επανάληψη γενικά; Ας δούμε ορισμένα παραδείγματα από την καθημερινότητά μας. Σκεφτείτε το περιεχόμενο της επόμενης εικόνας:



Η αδιάκοπη εναλλαγή των τεσσάρων εποχών αποτελεί ένα ξεκάθαρο παράδειγμα επανάληψης στην ζωή μας! Η σειρά Άνοιξη – Καλοκαίρι – Φθινόπωρο – Χειμώνας επαναλαμβάνεται...για πάντα!



Επίσης, χαρακτηριστικό παράδειγμα επανάληψης είναι ένας αγώνας αυτοκινήτων. Σε αυτή την περίπτωση, η επανάληψη πρέπει κάποια στιγμή να σταματήσει. Οι οδηγοί δεν μπορούν να τρέχουν για πάντα, κυνηγώντας ο ένας τον άλλον! Έπειτα από ένα συγκεκριμένο αριθμό γύρων, ο αγώνας φτάνει σε ένα τέλος. Οι οδηγοί ακολουθούν πιστά τη διαδρομή τους και να την επαναλαμβάνουν τόσες φορές όσες έχει οριστεί από την αρχή του αγώνα.

Όπως ξέρουμε από το μάθημα της Βιολογίας, η καρδιά μας είναι ο μοναδικός μυς του ανθρώπινου σώματος ο οποίος δεν χρειάζεται να πάρει εντολή από τον εγκέφαλο για να λειτουργήσει. Η καρδιά λειτουργεί ασταμάτητα σε μια συνεχή επανάληψη.

Τέλος, όπως βλέπουμε στο ρολόι μας, ο δείκτης των δευτερολέπτων του ρολογιού μας πραγματοποιεί 60 κινήσεις κάθε λεπτό, ενώ ο δείκτης των λεπτών πραγματοποιεί 60 κινήσεις κάθε μια ώρα.

Υπάρχουν, άραγε, περιπτώσεις που εμφανίζεται κάποιο είδος επανάληψης στον προγραμματισμό; Μα, φυσικά!!

Σκεφτείτε ένα παιχνίδι στρατηγικής όπου χρειάζεται να αναπαραστήσουμε στρατιώτες σε κάποιο πεδίο μάχης. Είναι δυνατόν ο σχεδιαστής του παιχνιδιού να δημιουργεί έναν-έναν όλους αυτούς τους στρατιώτες; Προφανώς όχι. Σχεδιάζει ένα μοντέλο του στρατιώτη, το οποίο επαναλαμβάνει αρκετές φορές στην οθόνη μας. Χωρίς την χρήση της επανάληψης, ο προγραμματιστής θα έπρεπε να γράψει πολλές εκατοντάδες φορές την ίδια εντολή, για να πετύχει το αποτέλεσμα της επόμενης εικόνας!



Θυμηθείτε ότι όταν θέλαμε να αναπαράγουμε μια νότα για 10 φορές, έπρεπε να γράψουμε 10 φορές την αντίστοιχη εντολή ενώ όταν θέλαμε να εμφανίσουμε το χαρακτήρα μας να περπατάει επαναλάβαμε πολλές φορές τον ίδιο συνδυασμό εντολών (κινήσου...βήματα και επόμενη ενδυμασία). Όταν προγραμματίζουμε τη συμπεριφορά ενός αντικειμένου, πολλές φορές θέλουμε να το κάνουμε να επαναλαμβάνει κάποια στοιχεία της συμπεριφοράς του για ορισμένο ή όχι αριθμό φορών.

Από τα προηγούμενα παραδείγματα προκύπτουν και τα δύο είδη επανάληψης που θα εξετάσουμε σε αυτό το κεφάλαιο: την επανάληψη που διαρκεί για όλη τη διάρκεια εκτέλεσης του έργου και την επανάληψη που συμβαίνει για συγκεκριμένο αριθμό φορών.

Όλες οι γλώσσες προγραμματισμού δίνουν εντολές που επιτρέπουν την επανάληψη συγκεκριμένων τμημάτων εντολών και σημειώστε ότι θα συζητήσουμε για πιο σύνθετες μορφές επανάληψης στο κεφάλαιο 12.

10.2 Θέλω να κάνω κάτι για πάντα...

Η δυνατότητα να επαναλαμβάνεται διαρκώς μια ακολουθία εντολών π.χ. η αναπαραγωγή ενός ήχου, μπορεί να υλοποιηθεί στο Scratch με την χρήση της εντολής για πάντα.

Όπως φαίνεται και από την απεικόνιση της εντολής, έχει δυο νέα χαρακτηριστικά που δεν έχουμε συναντήσει σε άλλες εντολές μέχρι τώρα:

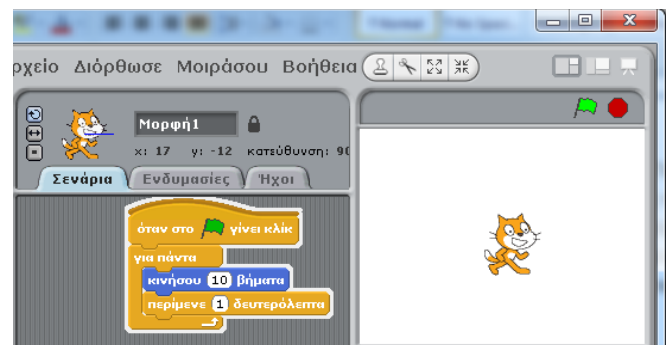


- ✓ Μπορεί να περιέχει άλλες εντολές μέσα στις «δαγκάνες» της. Δηλαδή σέρνουμε εντολές από τις παλέτες εντολών ανάμεσα στις δαγκάνες της και δημιουργούμε μικρο-σενάρια που περιέχονται μέσα στην εντολή για πάντα. Αυτές είναι οι εντολές που θα εκτελούνται διαρκώς και με τη συγκεκριμένη σειρά με την οποία τις τοποθετούμε.
- ✓ Στο κάτω μέρος της, η εντολή δεν έχει την εξοχή εκείνη που θα μας επέτρεπε να προσθέσουμε άλλη εντολή μετά από αυτήν. Δηλαδή η εντολή για πάντα μπορεί να είναι μόνο η τελευταία σε ένα σενάριο. Αυτό είναι πολύ λογικό καθώς από τη στιγμή που οι εντολές που περιέχει εκτελούνται για πάντα, τότε θα εκτελεστούν αυτές που την ακολουθούν;

Για να κατανοήσετε καλύτερα τη λειτουργία της εντολής για πάντα, δεν έχετε παρά να επαναλάβετε τις εντολές που περιέχει, όπως φαίνεται στο παρακάτω σχήμα:



Και τότε σταμάτα το συγκεκριμένο έργο, αφού οι εντολές τρέχουν για πάντα; Παρατηρήστε ότι όταν πατήσουμε την πράσινη σημαία, αυτή παραμένει διαρκώς φωτισμένη, γεγονός που σημαίνει ότι το έργο μας συνεχίζει να εκτελείται. Αυτό επίσης μπορεί να βεβαιωθεί από το άσπρο περίγραμμα στο σενάριο μας που μας δείχνει ότι το συγκεκριμένο σενάριο συνεχίζει να εκτελείται.



Μόνο όταν πατήσουμε το κόκκινο κουμπί που βρίσκεται δίπλα στην πράσινη σημαία θα σταματήσει να εκτελείτε το έργο μας.

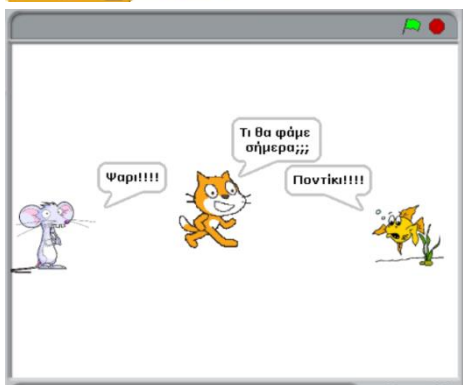
Ας δούμε δυο παραδείγματα ακόμη. Δε θα ήταν πιο ευχάριστο κατά τη διάρκεια ενός παιχνιδιού να ακούγεται διαρκώς μουσική; Για να το καταφέρετε αυτό επιλέξτε και σύρετε την εντολή για πάντα και μέσα της εισάγετε την εντολή Παίξε ήχο...μέχρι τέλους. Επιλέξτε το μουσικό κομμάτι ή τον ήχο που επιθυμείτε και διασκεδάστε με το αποτέλεσμα!



Θυμάστε τι θα συνέβαινε αν χρησιμοποιούσατε την εντολή Παίξε ήχο... Δοκιμάστε την αντίστοιχη εντολή και εξηγήστε το αποτέλεσμα.

Ας δούμε, ακόμη ένα πιο σύνθετο παράδειγμα χρήσης της εντολής για πάντα με πρωταγωνιστή τον "Αναποφάσιστο κ. Γατίδη...". Φανταστείτε τον κ. Γατίδη να βρίσκεται ανάμεσα σε ένα φοβισμένο ποντικό και ένα τρομαγμένο ψάρι. Δυσκολεύεται να αποφασίσει για το μεσημεριανό του γεύμα, κινείται δεξιά και αριστερά προσπαθώντας να πιάσει κάποιο από τα δύο αντικείμενα, όμως παραμένει αναποφάσιστος και αυτό τον αφήνει (δυστυχώς ή ευτυχώς) νηστικό. Ταυτόχρονα, οι δυο χαρακτήρες προτρέπουν τον κ.Γατίδη να φάει τον απέναντί τους κάθε φορά που τους πλησιάζει. Ας υλοποιήσουμε αυτό το μικρό σκετς.

Αφού βρούμε τις εικόνες για το ψάρι και το ποντίκι από το διαδίκτυο, τις εισάγουμε ως ξεχωριστά αντικείμενα στο Scratch. Τοποθετούμε το ποντίκι αριστερά και το ψάρι δεξιά και τον γάτο αρχικά κοντά στο ψάρι. Ο κ. Γατίδης θα κινείται οριζόντια αλλάζοντας τη θέση του κατά 170 σημεία από τα αριστερά στα δεξιά και αντίστροφα. Αρχικά, θα ρωτά το ψάρι «τι θα φάμε σήμερα;» και μετά θα πηγαίνει στον άλλο χαρακτήρα και θα κάνει την ίδια ερώτηση, για να τους τρομάξει. Αυτή η διαδικασία θα επαναλαμβάνεται για πάντα και συνεπώς πρέπει να την εισάγουμε μέσα σε μια εντολή για πάντα. Το σενάριο της γάτας φαίνεται στην επόμενη εικόνα:



Από την πλευρά τους, τα δυο κακόμοιρα ζωάκια μόλις πλησιάζει σε αυτά, ο κ.Γατίδης θα τον προτρέπουν να φάει τον απέναντι φίλο τους! Και θα το κάνουν για πάντα!

Το σενάριο για το ποντίκι:



Και το σενάριο για το ψάρι:

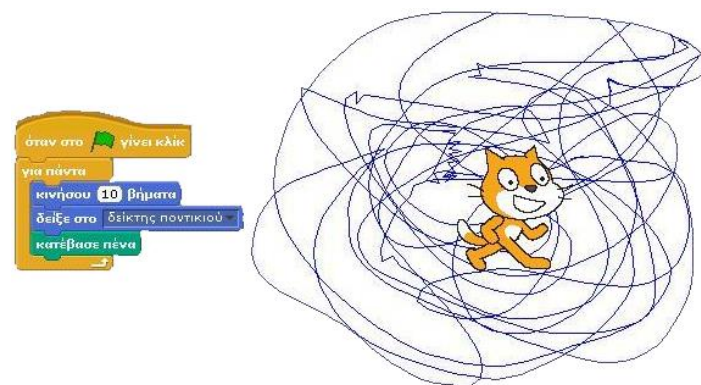


[10_π01.sb]

Μη ξεχνάτε ότι ένας πίνακας που περιέχει το χρονοδιάγραμμα των συμβάντων μπορεί να σας βοηθήσει πολύ στη σχεδίαση αντίστοιχων προγραμμάτων. Τώρα, προσπαθήστε και εσείς με την σειρά σας να δώσετε κίνηση στον κ. Ποντίκι και στον κ. Ψαρίδι ώστε να τρέχουν γύρω του με σκοπό να τον μπερδέψουν...για πάντα.

Η εντολή για πάντα μπορεί επίσης να χρησιμοποιηθεί όταν θέλουμε ένα αντικείμενο να ακολουθεί τον δείκτη του ποντικιού όπου και να το κατευθύνει ο χρήστης. Πως θα σας φαινόταν αν ο κύριος Γατίδης ακολουθούσε κάθε σας βήμα;

Έχουμε μάθει από τα προηγούμενα κεφάλαια την εντολή δείξε στο... όπου σε συνδυασμό με την κινήσου...βήματα μπορούμε να κινήσουμε ένα αντικείμενο προς κάποιο άλλο. Στην πρώτη εντολή, όμως, υπάρχει και η επιλογή δείξε στο δείκτη ποντικιού που δεν έχουμε χρησιμοποιήσει μέχρι τώρα. Αν για πάντα πούμε στον κ.Γατίδη να κινείται προς το δείκτη του ποντικιού και ταυτόχρονα κατεβάσουμε την πένα ώστε να καταγράφεται το ίχνος της κίνησής του, τότε το αποτέλεσμα του έργου μας μπορεί να μοιάζει με αυτό της επόμενης εικόνας.



[10_π02.sb]

Ο προγραμματιστής μας, πάλι αφηρημένος, έχει κάνει ένα λάθος στο έργο του. Μπορείτε να το εντοπίσετε; Καλό θα ήταν επιπλέον να πειραματιστείτε με διαφορετικό μέγεθος βήματος. Ποια η επίδρασή του στο τελικό σχέδιο που εμφανίζεται στην οθόνη του Scratch;

10.3 Η εντολή «επανάλαβε X»

Όπως είδαμε πιο πάνω, η εντολή **για πάντα** είναι χρήσιμη σε διάφορα προγράμματα όταν π.χ. θέλουμε να δείξουμε την κίνηση της γης γύρω από τον ήλιο ή για να αναπαραστήσουμε το δίλημμα του κ. Γατίδη όσον αφορά το φαγητό του.

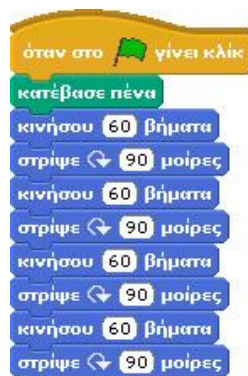
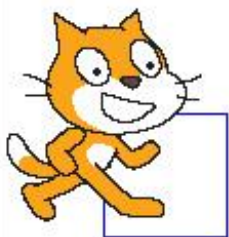
Η εντολή **επανάλαβε...** λειτουργεί με ακριβώς τον ίδιο τρόπο με την εντολή **για πάντα** με τη διαφορά ότι σε αυτήν την περίπτωση, οι επαναλήψεις των εντολών που περιέχονται μέσα στις δαγκάνες της, είναι συγκεκριμένες. Ο αριθμός των επαναλήψεων προσδιορίζεται από το νούμερο που συμπληρώνουμε στο λευκό κουτάκι. Υπάρχει, όμως, ακόμη μια πολύ σημαντική διαφορά. Από τη στιγμή που ο αριθμός των επαναλήψεων είναι συγκεκριμένος, άλλες εντολές μπορούν να ακολουθούν την εντολή **επανάλαβε...**

Χρησιμοποιώντας το παράδειγμα για τη μείωση της έντασης του ήχου που συζητήσαμε στο κεφάλαιο 8, θα δείξουμε πως κάνοντας χρήση της εντολής **επανάλαβε...** μπορούμε να ελαττώσουμε το μέγεθος του σεναρίου μας και ταυτόχρονα να το κάνουμε πιο κομψό και κατανοητό. Στα αριστερά της επόμενης εικόνας βλέπουμε το σενάριο μείωσης της έντασης του ήχου με τη χρήση 10 εντολών. Στα δεξιά βλέπουμε το ίδιο παράδειγμα με τα ίδια αποτελέσματα, αλλά με τη χρήση εντολής **επανάλαβε** ... Οι εντολές έχουν μειωθεί στις 4!.



[10_π03.sb]

Η εντολή **επανάλαβε X** όπως βλέπουμε είναι πολύ απλή στη χρήση της. Παρακάτω ακολουθεί ένα παράδειγμα σχεδιασμού ενός τετραγώνου αρχικά με χρήση της εντολής **επανάλαβε...** και στη συνέχεια χωρίς τη χρήση της.



[10_π04.sb]

Το συμπέρασμα και από αυτό το παράδειγμα είναι ότι η **επανάλαβε...** κάνει εφικτή τη συγγραφή μικρότερων έργων, επιτρέποντας την ευκολότερη ανάγνωσή τους. Φαντάζεστε πόσες εντολές χρειαζόμαστε για να πούμε στο αντικείμενό μας να σχεδιάσει 5 τετράγωνα χωρίς τη χρήση της συγκεκριμένης εντολής;

Άλλο ένα παράδειγμα με το οποίο μπορούμε να δείξουμε ότι με τη χρήση της εντολής **επανάλαβε X** μπορούμε να κάνουμε μαγικά! Συγκεκριμένα θα δείξουμε ότι ο κ. Γατίδης μπορεί να εξαφανίζεται και να εμφανίζεται σταδιακά στη σκηνή χρησιμοποιώντας την **άλλαξε...εφέ κατά X**. Θα χρειαστούμε την **επανάλαβε...** όπου ο κ. Γατίδης σε κάθε βήμα της θα εξαφανίζεται όλο και περισσότερο μέχρι να χαθεί τελείως από την οθόνη και μόλις εξαφανιστεί πλήρως τότε, ξανά με την ίδια εντολή θα εμφανιστεί πάλι σταδιακά. Η εντολή που θα μας βοηθήσει να εξαφανίσουμε τον κ.Γατίδη είναι η **άλλαξε...εφέ κατά...** στην οποία θα επιλέξουμε ως εφέ το φάντασμα. Για να είναι όμως ορατό το αποτέλεσμα θα χρησιμοποιήσουμε και την **περίμενε...δευτερόλεπτα** ώστε να «προλαβαίνουμε» να δούμε την εφαρμογή του εφέ. Το σενάριο παρουσιάζεται στην επόμενη εικόνα.

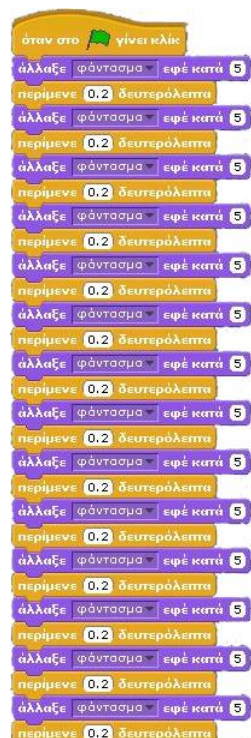


[10_π05.sb]

1 ^ο μπλοκ επανάληψης Αρχική κατάσταση	
Μετά την 10 ^η επανάληψη	

Μετά την 15 ^η Επανάληψη	
Μετά την 20 ^η Επανάληψη	
2 ^ο μπλοκ Μετά την 5 ^η Επανάληψη	
Μετά την 10 ^η επανάληψη	
Τελική κατάσταση	

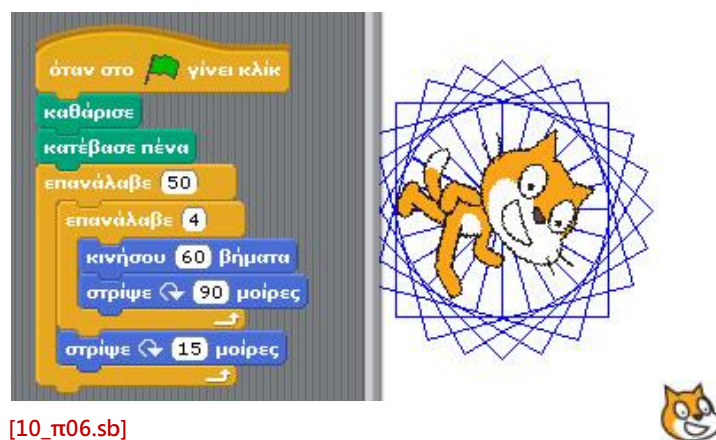
Οι εντολές που θα χρειαζόμασταν για να υλοποιήσουμε τον ίδιο στόχο χωρίς την επανάληψη, δεν χωράει ολόκληρος στο παράθυρο του Scratch. Θα έπρεπε να χρησιμοποιήσουμε $20 \times 2 + 20 \times 2$ εντολές, δηλαδή 80 εντολές! Στην επόμενη εικόνα εμφανίζονται μόνο 28...



Για το τέλος, κρατήσαμε το πιο δύσκολο κομμάτι αυτής της εντολής, το οποίο είναι η *εμφωλευμένη* επανάληψη. Όπως μας λέει η ίδια φράση, έχουμε τη δυνατότητα να εισάγουμε μια επανάληψη μέσα σε μια άλλη επανάληψη κάτι το οποίο μας δίνει νέες δυνατότητες π.χ. στον σχεδιασμό πολύπλοκων γεωμετρικών σχημάτων.

Σκεφτείτε το παράδειγμα με το σχεδιασμό του τετραγώνου. Πως υλοποιήθηκε; Χρησιμοποιήθηκε η εντολή επανάλαβε..., η οποία σχεδίαζε γραμμή-γραμμή ένα τετράγωνο. Εάν όμως θέλετε να σχεδιάσετε περισσότερα από ένα τετράγωνα στο ίδιο πρόγραμμα, θα πρέπει να γράψετε το ίδιο κομμάτι εντολών πολλές φορές; Μήπως υπάρχει άλλη λύση;

Το μόνο που έχουμε να κάνουμε είναι να εισάγουμε μία εντολή επανάληψης που θα περιλαμβάνει το σενάριο δημιουργίας του τετραγώνου, άρα και την αρχική εντολή επανάληψης που χρησιμοποιήσαμε.



[10_π06.sb]

Για κάθε νέα εξωτερική επανάληψη, εκτελούνται όλα τα βήματα της εσωτερικής επανάληψης όσα κι αν είναι. Δηλαδή π.χ. εάν η εξωτερική επανάληψη γίνεται 50 φορές και η εσωτερική γίνεται 4, τότε για κάθε 1 εξωτερική επανάληψη θα εκτελούνται και οι 4 εσωτερικές επαναλήψεις. Με μια μικρή προσθήκη (εντολή στρίψε δεξιόστροφα... μοίρες για την αλλαγή της κατεύθυνσης του κάθε τετραγώνου) το αποτέλεσμα είναι ένα όμορφο γεωμετρικό σχήμα.

Θα πρέπει να θυμάστε ότι κάθε φορά που χρησιμοποιούμε εμφωλευμένη εντολή επανάλαβε..., μπορούμε να κατανοήσουμε το σενάριο απλά δημιουργώντας την ξανά στο μυαλό αντικαθιστώντας την εσωτερική επανάλαβε X, με μια ισοδύναμη σειρά εντολών. Δείτε πως το προηγούμενο σενάριο απλοποιείται μετά την αντικατάσταση της εσωτερικής επανάληψης με ισοδύναμη ακολουθία εντολών:



Τα δυο σενάρια είναι ισοδύναμα. Αρκεί να θυμάστε ότι για κάθε εξωτερική επανάληψη, θα εκτελούνται όλες οι εσωτερικές επαναλήψεις. Παρατηρήστε ότι είναι πολύ ευκολότερο να κατανοήσετε το τι κάνει το έργο όταν μελετάτε το σενάριο με την εμφωλευμένη επανάληψη.

10.4 Άλλα παραδείγματα

10.4.1 Ένας απλός ανεμόμυλος

Το επόμενο παράδειγμα μας δείχνει πως μπορούμε να δημιουργήσουμε την κίνηση της έλικας ενός ανεμόμυλου αλλά και την απλή συνεχόμενη κίνηση ενός πουλιού που διασχίζει το σκηνικό μας.



Δημιουργούμε ένα σκηνικό που περιέχει το οίκημα και δυο αντικείμενα, την έλικα του ανεμόμυλου και το πουλί. Θέλουμε η έλικα να γυρίζει γύρω από τον εαυτό της ενώ το πουλί να κινείται από τη μια άκρη της οθόνης στην άλλη και να επαναλαμβάνει αυτήν την κίνηση από την αρχή.

Για την έλικα του ανεμόμυλου θα έχουμε το πολύ απλό σενάριο:



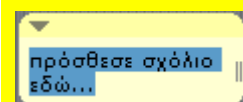
Το σενάριο για το πουλί θα περιέχει την ομαλή αλλαγή θέσης από τη μια άκρη της οθόνης στην άλλη και στη συνέχεια την επανατοποθέτηση του πουλιού στην αρχική του θέση ώστε στην επόμενη επανάληψη να πραγματοποιηθεί η ίδια κίνηση:



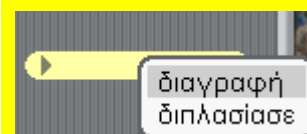
[10_π07.sb]



Μπορούμε να προσθέσουμε και σχόλια στο σενάριο μας πατώντας δεξί κλικ και επιλέγοντας «πρόσθεσε σχόλιο». Μπορούμε να σχολιάσουμε το σκεπτικό ανάπτυξης του προγράμματος, ή να εισάγουμε μια υπενθύμιση για την ανάπτυξη του έργου μας. Με τα σχόλια ο κώδικας γίνεται πιο ευανάγνωστος και πιο κατανοητός τόσο για αυτόν που το γράφει, όσο και για τον απλό αναγνώστη. Μη διστάζετε λοιπόν να βάλετε σχόλια στα σενάρια σας. Θα σας βοηθήσουν να δομήσετε καλύτερα τον κώδικα σας και να θυμάστε τι έχετε κάνει σε κάποια σημεία που αντιμετωπίστε δυσκολίες. Αν επιλέξετε "πρόσθεσε σχόλιο" τότε θα εμφανιστεί η κίτρινη περιοχή σχολίων όπως φαίνεται στην παρακάτω εικόνα, στην οποία μπορείτε να εισάγετε το κείμενό σας. Για να μεταβάλετε το πλάτος μιας περιοχής σχολίων, χρησιμοποιήστε τη λαβή που βρίσκεται στη δεξιά της άκρη. Πατήστε το τριγωνάκι πάνω αριστερά για να διπλώσετε ή να ξεδιπλώσετε την περιοχή των σχολίων. Για να συνδέσετε ένα σχόλιο με μια εντολή, σέρνετε το σχόλιο πάνω στην εντολή. Για να αποσυνδέσετε το σχόλιο, το απομακρύνετε από την εντολή.



Για να διαγράψετε ένα σχόλιο κάντε δεξί κλικ πάνω του και επιλέξτε διαγραφή.



10.4.2 Ρολόι!

Το επόμενο παράδειγμα μας δείχνει πως μπορούμε να χρησιμοποιήσουμε την εντολή **επανάλαβε X**, ώστε να προσομοιώσουμε την κίνηση των δεικτών του ρολογιού για μία ολόκληρη ημέρα!



Χρειαζόμαστε 3 αντικείμενα και ένα υπόβαθρο. Τα αντικείμενα αντιστοιχίζονται στους 3 διαφορετικούς δείκτες του ρολογιού και το υπόβαθρο στο καντράν του. Για την επίλυση του προβλήματός μας θα βασιστούμε στον υπολογισμό του αριθμού των φορών που ο κάθε δείκτης του ρολογιού θα στρέφεται σε επίπεδο λεπτού, ώρας και 24ωρου.

Για το δείκτη των δευτερολέπτων θέλουμε απλά να πραγματοποιεί μια πλήρη περιστροφή γύρω από τον εαυτό του σε 60 δευτερόλεπτα. Αυτό σημαίνει ότι αν θέλουμε να αλλάζει θέση κάθε ένα δευτερόλεπτο, τότε θα πρέπει να γυρίζει κατά 360 μοίρες/60 δευτερόλεπτα = 6 μοίρες κάθε φορά. Με τον τρόπο αυτό, στο ένα λεπτό θα βρίσκεται στη θέση από την οποία ξεκίνησε.

Για το δείκτη των λεπτών, επιζητούμε το ίδιο αποτέλεσμα με τη διαφορά ότι ο δείκτης θα πρέπει να βρεθεί ξανά στην αρχική του θέση μετά από 60 λεπτά. Αν θέλουμε να αλλάζει θέση κάθε λεπτό, τότε θα πρέπει να περιστρέφεται κατά 6 μοίρες κάθε 60 δευτερόλεπτα.

Τέλος, για το δείκτη των ωρών, θέλουμε να μετακινείται 1 φορά κάθε ώρα και σε 12 μετακινήσεις να έχει περιστραφεί γύρω από τον εαυτό του. Άρα θα μετακινείται κάθε 60λεπτά = $60 \times 60 = 3600$ δευτερόλεπτα, ενώ η περιστροφή του θα πρέπει να είναι $360 \text{ μοίρες} / 12 \text{ φορές} = 15 \text{ μοίρες}$ κάθε φορά.

Αν θέλουμε να προσομοιώσουμε το ρολόι μας μόνο για μια ημέρα, τότε χρειαζόμαστε:

- $60 \times 60 \times 24 = 86400$ κινήσεις του δείκτη των δευτερολέπτων
- $60 \times 24 = 1440$ κινήσεις του δείκτη των λεπτών
- Και 24 κινήσεις του δείκτη των ωρών.

Το σενάριο για τα αντικείμενα «δείκτης δευτερολέπτων»,



για το αντικείμενο «δείκτης λεπτών»,



Για το αντικείμενο «δείκτης ωρών»

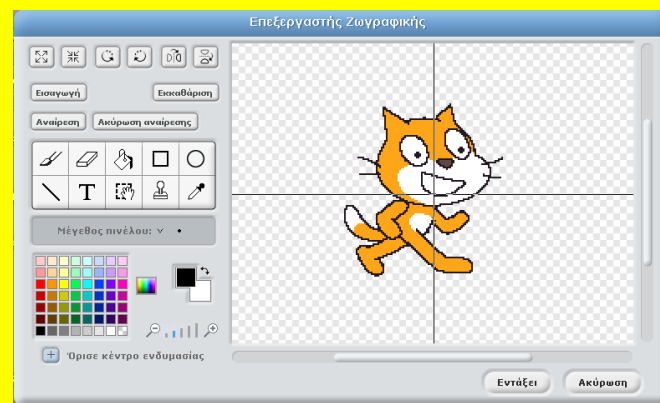


[10_π08.sb]



Το αποτέλεσμα του συγκεκριμένου έργου είναι οι δείκτες να περιστρέφονται για μία ολόκληρη ημέρα, γύρω από τον άξονα τους, δείχνοντας μας τα δευτερόλεπτα, τα λεπτά και την ώρα που πέρασαν από την εκκίνηση του προγράμματος. Πόσο δύσκολο είναι να μετατρέψετε το συγκεκριμένο έργο σε ένα ρολόι που τρέχει για πάντα με τη χρήση της εντολής για πάντα. Μήπως απλά με μια αντικατάσταση των επανάλαβε... με την για πάντα πετυχαίνουμε το επιθυμητό αποτέλεσμα;

Αν δοκιμάσατε να δημιουργήσετε το συγκεκριμένο έργο, λογικά θα συναντήσατε μια δυσκολία: να περιστρέψετε τους δείκτες γύρω από ένα συγκεκριμένο σημείο τους και όχι από το κέντρο τους! Για να λύσετε το συγκεκριμένο πρόβλημα, επιλέξτε τον επεξεργαστή ζωγραφικής για κάθε δείκτη και πατήστε την επιλογή όρισε κέντρο ενδυμασίας που βρίσκεται κάτω αριστερά στο αντίστοιχο παράθυρο, όπως φαίνεται στην επόμενη εικόνα.



Μόλις πατήσετε τη συγκεκριμένη επιλογή, εμφανίζονται δυο άξονες που προσδιορίζουν το κέντρο περιστροφής από την τομή τους. Μπορείτε να σύρετε την τομή, στο σημείο που επιθυμείται να γίνει το νέο κέντρο περιστροφής. Στους δείκτες του ρολογιού, το κέντρο του ρολογιού θα είναι κοντά στις άκρες τους.

Περίληψη

Στο κεφάλαιο αυτό, συζητήσαμε για την αξία της επανάληψης στον προγραμματισμό και μελετήσαμε δυο μορφές της: την επανάληψη που διαρκεί για πάντα (με την εντολή για πάντα) και

την επανάληψη που πραγματοποιείται για συγκεκριμένο αριθμό φορών (με την εντολή **επανάλαβε...**). Και στις δυο περιπτώσεις ένα μικρο-σενάριο επαναλαμβάνεται ενώ στην δεύτερη περίπτωση είναι δυνατόν να εκτελούνται και άλλες εντολές μετά την εντολή **επανάλαβε...** Επιπλέον, μια εντολή **επανάλαβε...**, μπορεί να βρίσκεται μέσα σε μια άλλη εντολή **επανάλαβε X**. Είναι σημαντικό να θυμόμαστε ότι για κάθε εξωτερική επανάληψη, πραγματοποιούνται όλες οι εσωτερικές επαναλήψεις των περιεχόμενων εντολών. Μην ανησυχείτε για την κατανόηση αυτών των χαρακτηριστικών, γιατί θα αξιοποιούμε τις δυο αυτές εντολές σε όλα τα επόμενα κεφάλαια και θα έχετε πολλές ευκαιρίες για να καταλάβετε διαφορετικές περιστάσεις χρήσης τους.

Ερωτήσεις

1) Διαβάστε τις παρακάτω προτάσεις και σημειώστε Σ για την σωστή και Λ για την λάθος πρόταση:

Όταν έχουμε μια εντολή επανάληψης **για πάντα** δεν θα εκτελεστεί καμία εντολή που βρίσκεται αμέσως μετά από αυτήν.

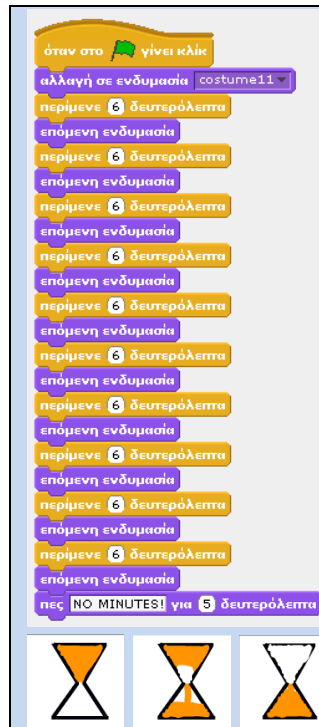
Μπορούμε να έχουμε εμφωλευμένη μια εντολή **για πάντα** μέσα σε μία εντολή **επανάλαβε X**.

2) Περιγράψτε το αποτέλεσμα που θα έχουν τα παρακάτω σενάρια για το κ. Γατίδη:



Δραστηριότητες

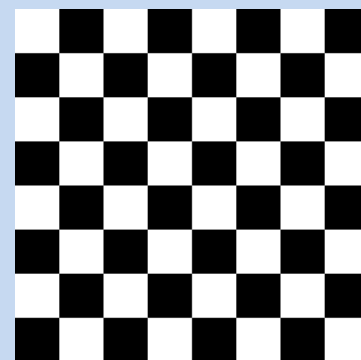
1) Ο παρακάτω κώδικας υλοποιεί μια κλεψύδρα που μετρά το χρόνο. Στις εικόνες που ακολουθούν φαίνονται αντίστοιχα η αρχική, μια ενδιάμεση και η τελική ενδυμασία της κλεψύδρας. Δώστε μια εναλλακτική υλοποίηση του σεναρίου χρησιμοποιώντας την κατάλληλη δομή επανάληψης.

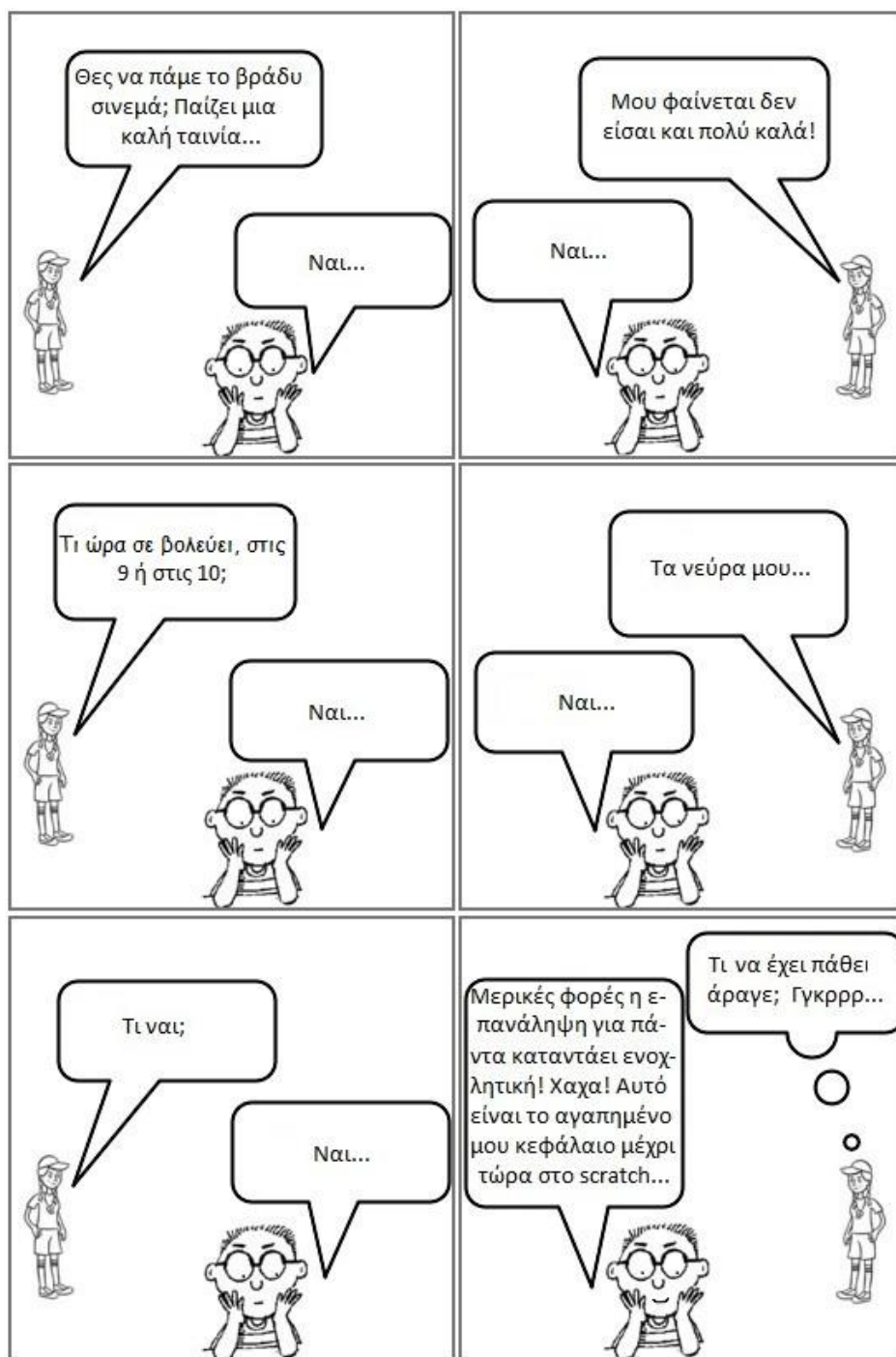


2) Είδαμε προηγουμένως, πως μπορούμε με ένα μικρό σενάριο να φτιάξουμε ένα τετράγωνο. Τροποποιήστε το παράδειγμα έτσι ώστε το αποτέλεσμα να είναι ένα ισόπλευρο τρίγωνο. Επιπλέον, δημιουργήστε ένα σενάριο που δημιουργεί πολλαπλά τρίγωνα διαφορετικού χρώματος περιστρέφοντας το αντικείμενό σας κατά 5 μοίρες κάθε φορά πριν τη σχεδίαση κάθε τριγώνου.

3) Σε παράδειγμα του κεφαλαίου εξηγήθηκε πως μπορούμε να προσομοιώσουμε την κίνηση της Γης γύρω από τον Ήλιο. Γράψτε ένα σενάριο στο οποίο θα πραγματοποιείται και η κίνηση του φεγγαριού γύρω από τη Γη ταυτόχρονα με την κίνηση της Γης γύρω από τον Ήλιο.

4) Σχεδιάστε με τη χρήση εντολών επανάληψης μια σκακιέρα





Κεφάλαιο 11: Ο έλεγχος

Σε αυτό το κεφάλαιο:

11.1 Εισαγωγή στον έλεγχο

11.2 Οι εντολές ελέγχου

Η εντολή "Εάν..."

Η εντολή "Εάν..." και οι τελεστές συνθηκών

Η εντολή "Εάν...Αλλιώς..."

11.3 Παραδείγματα

«Αν εισβάλλω στη Λακωνία, θα ισοπεδώσω την πόλη της Σπάρτης»
(Βασιλιάς Φίλιππος Β')



11.1 Εισαγωγή στον έλεγχο

Ο κόσμος γύρω μας είναι γεμάτος επιλογές οι οποίες εξαρτώνται από διάφορες συνθήκες. Εάν βρέχει, παίρνουμε ομπρέλα. Εάν έχει ήλιο, φοράμε καπέλο. Εάν είναι σαββατοκύριακο, ξεκουραζόμαστε, αλλιώς πάμε σχολείο. Το ίδιο όμως ισχύει και στον υπολογιστικό κόσμο. Όταν ένα αυτοκινητάκι πατάει στο γρασίδι, τότε επιβραδύνεται. Όταν ένα κακό μπαρμπαδελάκι ακουμπήσει τον Packman, τότε αφαιρείται μια ζωή. Αν ο Packman ακουμπήσει ένα φρουτάκι, τότε το φρουτάκι εξαφανίζεται και εμείς κερδίζουμε πόντους. Αν προσπαθήσουμε να αντιγράψουμε ένα αρχείο σε ένα φάκελο που περιέχει ένα αρχείο με το ίδιο όνομα, τότε το λειτουργικό σύστημα θα μας ρωτήσει τι ακριβώς θέλουμε να κάνουμε. Όταν ένα αντικείμενο στο Scratch θέλει να προσδιορίσει τη συμπεριφορά του ανάλογα με το τι συμβαίνει γύρω του, τότε χρησιμοποιούμε τις εντολές εάν... και εάν... αλλιώς...

Είδαμε ότι ο οδηγός σε έναν αγώνα ράλι επαναλαμβάνει το γύρο της πίστας όσες φορές έχει οριστεί από τον αγώνα. Τι γίνεται όμως όταν του τελειώνουν τα καύσιμα; Εάν του τελειώνουν τα καύσιμα τότε ο οδηγός θα μπει στο pit stop για ανεφοδιασμό, όπως φαίνεται στην επόμενη εικόνα. Αλλιώς, ο οδηγός θα συνεχίσει να επαναλαμβάνει την πορεία του.



Η καθημερινότητα μας θα ήταν πολύ μονότονη και με περιορισμένες δυνατότητες αν δεν είχαμε διαφορετικές επιλογές. Όμως πριν κάνουμε μια επιλογή εξετάζουμε τουλάχιστον μια συνθήκη που μπορεί να αφορά τα κίνητρά μας, τις ανάγκες μας, τις υποχρεώσεις μας κτλ. Όλοι σας έχετε δώσει την απάντηση «Εάν βρέξει, δεν θα μπορέσουμε» στην ερώτηση «Θα πάμε μετά το σχολείο να παίξουμε;». Στο συγκεκριμένο παράδειγμα, η συνθήκη βρίσκεται στην έκφραση «Εάν βρέξει». Υπάρχουν δυο πιθανές περιπτώσεις. Είτε να βρέξει είτε να μην βρέξει και η επιλογή μας για το αν τελικά

θα πάμε για παιχνίδι εξαρτάται από αυτήν ακριβώς τη συνθήκη.

Στο προγραμματισμό μπορούμε να επιλέξουμε την υπό συνθήκη εκτέλεση ορισμένων εντολών, δηλαδή να προσδιορίσουμε ότι κάποιες εντολές θα τρέχουν μόνο εφόσον ισχύει μια συνθήκη. Η δομή της εντολής εάν... στον προγραμματισμό είναι η ακόλουθη:

Εάν (συνθήκη) (εντολές)

Η συνθήκη μπορεί να αποτιμηθεί είτε σε αληθής (δηλαδή να ισχύει) είτε σε ψευδής (δηλαδή να μην ισχύει). Σε περίπτωση που η συνθήκη είναι αληθής, τότε θα εκτελεστούν οι εντολές που περιέχει, ενώ αν είναι ψευδής δεν θα εκτελεστούν και το πρόγραμμα θα συνεχίσει στις επόμενες εντολές. Ας προχωρήσουμε σε ένα μικρό παράδειγμα πάλι από την καθημερινότητα.

Υποθέστε ότι ο καθηγητής των μαθηματικών σας ανακοινώνει ότι την Δευτέρα θα έχετε διαγώνισμα στην ύλη που έχετε καλύψει μέχρι τώρα. Πηγαίνετε σπίτι και το ανακοινώνετε στους γονείς σας. Εκείνοι με την σειρά τους σας λένε ότι θα πάτε για παιχνίδι μόνο αν σας ξετάσουν και δουν ότι είστε διαβασμένοι. Επίσης σας λένε ότι, επειδή θεωρούν το μάθημα αυτό πολύ σημαντικό για την εξέλιξη σας, αν γράψετε είκοσι θα σας αγοράσουν το πολυπόθητο παιχνίδι του υπολογιστή που τόσο καιρό ζητούσατε. Οπότε εσείς σκέφτεστε «Να η ευκαιρία μου!!». Και επειδή όλοι ξέρουμε ότι το παιχνίδι δεν βγαίνει ποτέ από το μυαλό σας σκέφτεστε: «αν αρχίσω το διάβασμα τώρα και τελειώσω νωρίς, τότε θα έχω χρόνο και για παιχνίδι με τον φίλο μου». Βέβαια υπάρχει και το ενδεχόμενο ο φίλος σας να διαβάζει ακόμα αλλά αυτό δεν σας ενοχλεί καθόλου καθώς αν διαβάζει ακόμα, τότε εσείς θα σερφάρετε στις αγαπημένες σας ιστοσελίδες στο διαδίκτυο.

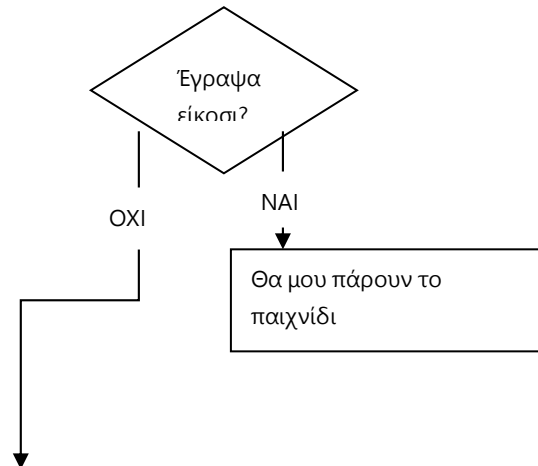
Όπως θα έχετε ήδη διαπιστώσει αυτή η ιστορία έχει αρκετές συνθήκες και πολλές διαφορετικές ενδεχόμενες καταλήξεις. Αλλά αυτή δεν είναι η πραγματικότητα; Ας προσπαθήσουμε να τις αναλύσουμε με την βοήθεια ενός διαγράμματος ροής. Το διάγραμμα ροής απεικονίζει κάθε συνθήκη σε ένα ρόμβο και κάθε δυνατή ενέργεια που μπορεί να προκύψει μετά τον έλεγχο

της συνθήκης σε ένα ορθογώνιο. Με βέλη σχηματίζουμε τα διάφορα πιθανά μονοπάτια. Κάθε συνθήκη έχει δυο πιθανά μονοπάτια:

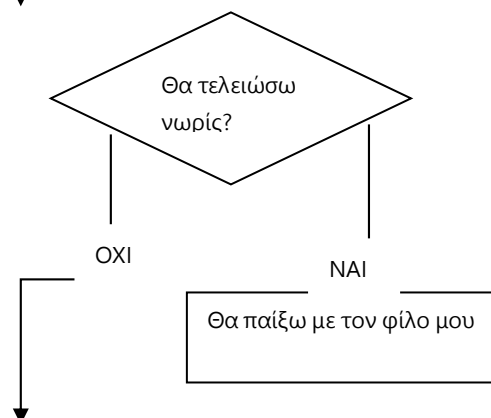
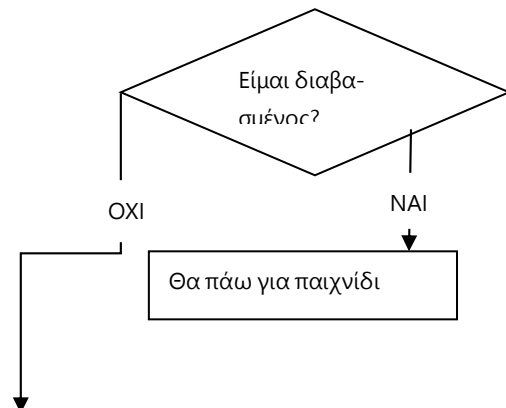
A) αν ισχύει, θα πραγματοποιηθούν οι ενέργειες που είναι υπό συνθήκη

B) αν δεν ισχύει, δεν θα πραγματοποιηθούν.

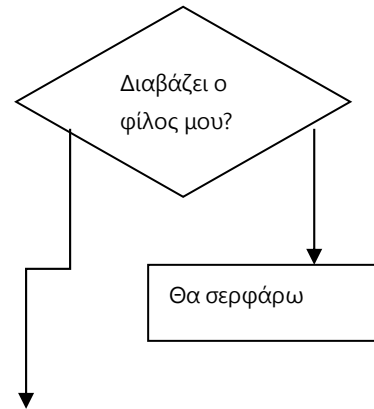
Ξεκινώντας από την αρχή συναντάμε την πρώτη συνθήκη που αφορά το βαθμό που μπορεί να πάρουμε στην εξέταση. Ας δούμε το διάγραμμα ροής της.



Και τα διαγράμματα ροής για τις επόμενες συνθήκες:



OXI NAI



Υποθέστε ότι τελικά ήρθε η μέρα του διαγωνίσματος και την προηγούμενη μέρα τελικά συνέβησαν τα ακόλουθα: Ήσασταν διαβασμένοι όταν σας εξέτασαν οι γονείς σας, ο βαθμός σας ήταν δεκαοχτώ, τελειώσατε νωρίς το διάβασμα και ο φίλος σας όταν εσείς είχατε τελειώσει, διάβαζε ακόμα. Ποια μονοπάτια θα ακολουθήσετε και ποιες ενέργειες θα εκτελεστούν; Ποιες όχι;

Στην πραγματικότητα, οι συνθήκες μπορεί να είναι ακόμη περισσότερες. Για παράδειγμα για να πάρετε τον φίλο σας τηλέφωνο θα πρέπει να έχετε και χρόνο ομιλίας στο κινητό σας αλλά και να είστε σε ένα μέρος που θα έχετε καλό σήμα. Εδώ βλέπουμε ότι έχουμε δυο συνθήκες. Ποιες είναι αυτές; Τι θα συμβεί αν δεν έχετε καθόλου σήμα; Τι θα γίνει αν δεν έχετε χρόνο ομιλίας; Τι θα γίνει αν έχετε και χρόνο ομιλίας αλλά και καλό σήμα; Τι θα γίνει αν έχετε καλό σήμα αλλά δεν έχετε χρόνο ομιλίας;

Για να μπορούν και τα αντικείμενα στον προγραμματισμό να εξετάζουν πολύπλοκες συνθήκες πριν επιδείξουν τη συμπεριφορά τους, οι γλώσσες προγραμματισμού διαθέτουν ένα σύνολο τελεστών συσχέτισης των συνθηκών. Δηλαδή, αν έχουμε περισσότερες από μια συνθήκες που θέλουμε να ελέγξουμε, υπάρχουν τρόποι με τους οποίους μπορούμε να περιγράψουμε πώς σχετίζονται οι συγκεκριμένες συνθήκες.

Ένας από αυτούς τους τελεστές είναι ο **KAI**. Αν χρησιμοποιήσουμε αυτόν τον τελεστή ανάμεσα στις συνθήκες μας, τότε για να εκτελεστεί το αντίστοιχο σύνολο εντολών θα πρέπει να ισχύουν και οι δυο συνθήκες μεταξύ των οποίων βρίσκεται ο τελεστής. Το προηγούμενο παράδειγμα θα μπορούσε να απεικονιστεί ως εξής:

Αν (έχω χρόνο ομιλίας) **KAI** (έχω σήμα), **τότε** (καλώ τον φίλο μου).

Αν έστω και μια συνθήκη δεν ισχύει τότε δεν εκτελείται και η αντίστοιχη ενέργεια.

Ας δούμε ένα διαφορετικό παράδειγμα. Έστω ότι θέλετε να δείτε κάποιες φωτογραφίες από ένα πρόσωπο στο facebook. Για να το καταφέρετε αυτό είτε θα πρέπει να είστε φίλος με αυτό το πρόσωπο είτε να έχει ένα δημόσιο προφίλ. Πάλι έχουμε δυο συνθήκες. Με τη διαφορά όμως ότι τώρα για να πραγματοποιήσουμε την ενέργειά μας αρκεί να ισχύει μια από τις δυο συνθήκες. Για αυτές τις περιπτώσεις είναι διαθέσιμος ο τελεστής συσχέτισης συνθηκών **Η**. Το παράδειγμά μας θα μπορούσε να απεικονιστεί ως εξής:

Αν (είμαι φίλος με το άτομο) **Η'** (το άτομο έχει δημόσιο προφίλ) **τότε** (βλέπω φωτογραφίες)

Για να μην καταφέρετε να δείτε τις φωτογραφίες, θα πρέπει όλες οι συνθήκες να είναι ψευδείς, δηλαδή να μην ισχύει καμία.

Συνοψίζοντας τα παραπάνω, στην επόμενη εικόνα παρουσιάζεται ο πίνακας αληθείας των δυο τελεστών Η' και ΚΑΙ. Πίνακας αληθείας; Σε αυτόν μπορούμε να δούμε αν ο τελικός συνδυασμός συσχέτισης των συνθηκών με τους αντίστοιχους τελεστές αποτιμάται ως ψευδής ή αληθής.

Συνθήκη 1	Συνθήκη 2	Σ1 ΚΑΙ Σ2	Σ1 Η Σ2
Αληθής	Αληθής	Αληθής	Αληθής
Αληθής	Ψευδής	Ψευδής	Αληθής
Ψευδής	Αληθής	Ψευδής	Αληθής
Ψευδής	Ψευδής	Ψευδής	Ψευδής

Πήραμε μια σύντομη θεωρητική γεύση από αυτά που θα ακολουθήσουν παρακάτω. Ας ξαναγυρίσουμε στον κόσμο του Scratch.

11.2 Οι εντολές ελέγχου του scratch

11.2.1 Η εντολή "Εάν..."

Πως λοιπόν το Scratch μας επιτρέπει να εκτελούμε ορισμένες εντολές μόνο εφόσον ισχύει μια συνθήκη; Καταρχάς, μας παρέχει την εντολή **εάν** που βρίσκεται στην παλέτα **Έλεγχος**. Η σύνταξη της φαίνεται στην επόμενη εικόνα.



Το άδαιο εξάγωνο που ακολουθεί την ετικέτα «εάν», είναι το σημείο μέσα στο οποίο πρέπει να προσδιορίσουμε τις συνθήκες που πρέπει να ελεγχθούν πριν εκτελεστούν οι εντολές που περιέχονται μέσα στις δαγκάνες της **εάν**. Αν η συνθήκη αποτιμηθεί ως αληθής κατά την εκτέλεση του έργου μας (δηλαδή ισχύει), **τότε** το σώμα εντολών που περιέχεται μέσα στην **εάν** θα εκτελεστεί, **αλλιώς**, αν η συνθήκη είναι ψευδής, θα συνεχιστεί η εκτέλεση των εντολών που ακολουθούν την **εάν**. Παρατηρήστε την εξοχή που υπάρχει στο κάτω μέρος της **εάν** και η οποία υποδηλώνει ότι άλλες εντολές μπορούν να την ακολουθούν.

Πρέπει να σημειώσουμε ότι η συνθήκη ελέγχεται μόνο τη στιγμή που φτάνει η εκτέλεση του προγράμματος στην εντολή **εάν**. Η **εάν** δεν σταματά το χρόνο αν είναι ψευδής. Αντίθετα, ακαριαία δίνει τον έλεγχο στην επόμενη εντολή που την ακολουθεί.

Μπορείτε να διακρίνετε ότι η συνθήκη της εντολής έχει εξάγωνο σχήμα; Ρίξτε μια ματιά στα παλέτες που έχουμε δει έως σήμερα (κίνηση, όψεις, πένα, έλεγχος, ήχος) και προσπαθήστε

να ανιχνεύσετε εντολές με εξάγωνο σχήμα. Καμία. Όπως μπορείτε να παρατηρήσετε μπλοκ μόνο από τις παλέτες **Αισθητήρες**, **Τελεστές** και **Μεταβλητές** έχουν αντίστοιχο σχήμα και μπορούν να χρησιμοποιηθούν ως στοιχεία των συνθηκών.

11.2.2 Η εντολή "Εάν..." και αισθητήρες

Ας ξεκινήσουμε να χρησιμοποιούμε τους αισθητήρες. Όπως υποδηλώνει και το όνομά τους, αφορούν το τι μπορεί να αισθανθεί ο χαρακτήρας μας! Ναι, οι χαρακτήρες μας στο Scratch έχουν αισθητήρια όργανα και μπορούν να αναγνωρίσουν ποιους αγγίζουν, αν πατήθηκε κάποιο κουμπί του πληκτρολογίου ή του ποντικιού κτλ. και να δρουν ανάλογα. Οι αισθητήρες είναι ουσιαστικά συνθήκες οι οποίες μας ενημερώνουν για το τι αντιλαμβάνονται οι χαρακτήρες μας. Για παράδειγμα, η συνθήκη **αγγίζει το....** είναι αληθής όταν το αντικείμενο αγγίζει το στοιχείο που έχει επιλεγεί από την αντίστοιχη λίστα.

π.χ. θέλουμε, αν ο γάτος ακουμπά τον κύριο στρείδι, να τον χαιρετά και να αλλάζει κατεύθυνση, όπως παρουσιάζεται στο παρακάτω σενάριο:



[11_π01.sb]

Στο σενάριο αυτό, σε κάθε επανάληψη της **για πάντα**, το αντικείμενο κινείται 10 βήματα και μετά γίνεται έλεγχος για το αν το αντικείμενο αγγίζει το στρείδι. Αν δεν το αγγίζει, τότε εκτελείται η **εάν στα όρια, αναπήδησε**, η οποία ελέγχει αν το αντικείμενο βρίσκεται στα όρια της οθόνης του Scratch. Αν βρίσκεται στα όρια, αντιστρέφει την κατεύθυνσή του. Στην περίπτωση που το αντικείμενό μας ακουμπά το στρείδι, πρώτα λέει «Γεια σου» για 2 δευτερόλεπτα και αλλάζει κατεύθυνση και μετά εκτελείται η εντολή **εάν στα όρια, αναπήδησε**.



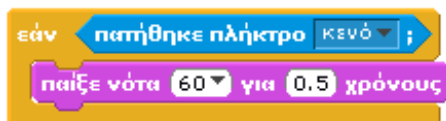
Η συνθήκη **αγγίζει το....** μπορεί να πάρει τρεις διαφορετικές:

αγγίζει το **Μορφή2** : Ελέγχει αν το αντικείμενό μας αγγίζει άλλη μορφή, όπως στο προηγούμενο παράδειγμα. Η διαθέσιμη λίστα μας επιτρέπει να επιλέξουμε οποιοδήποτε από τα αντικείμενα που έχουν εισαχθεί στο έργο μας.

αγγίζει το **δείκτης ποντικιού** : Ελέγχει αν το αντικείμενό μας αγγίζει το δείκτη του ποντικιού. Ή αντιστρόφως, ελέγχεται το κατά πόσο ο δείκτης του ποντικιού ακουμπά οποιοδήποτε σημείο του αντικείμενού μας.

αγγίζει το **όριο** : Ελέγχει αν το αντικείμενό μας αγγίζει οποιοδήποτε από τα όρια της οθόνης.

Η συνθήκη **πατήθηκε πλήκτρο...** μας ενημερώνει για το αν πατήθηκε το πλήκτρο που έχουμε προσδιορίσει. Έτσι, αν θέλουμε όταν πατάμε το κενό να ακούγεται μια συγκεκριμένη νότα:



Στη θέση του πλήκτρου μπορούμε να επιλέξουμε κάποιο από τα γράμματα (a-z), τους αριθμούς (0-9) καθώς και τα βελάκια. Περίεργο! Δεν σας θυμίζει ο αισθητήρας αυτός, την εντολή **όταν το πλήκτρο...πατηθεί**; Ποια είναι η διαφορά των δυο επόμενων σεναρίων;



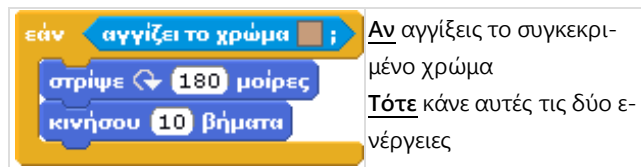
[11_π02.sb]

Δημιουργήστε ένα έργο στο οποίο δυο αντικείμενα θα ξεκινούν από την ίδια θέση και το καθένα θα έχει ένα από τα δυο προηγούμενα σενάρια. Λογικά οι δυο εντολές πρέπει να παράγουν το ίδιο αποτέλεσμα. Παρατηρήστε όμως, ότι όταν πατάτε το πλήκτρο κενό μια φορά, το πρώτο σενάριο προκαλεί την κίνηση του αντικείμενου κατά 10 βήματα ενώ το δεύτερο σενάριο κινεί το αντικείμενο κατά πολλαπλάσιο αριθμό βημάτων, δηλαδή το κινεί περισσότερο. Αυτό οφείλεται στο ότι στην πρώτη περίπτωση το πάτημα του πλήκτρου πυροδοτεί το αντίστοιχο γεγονός και την εκτέλεση του πρώτου σεναρίου ενώ στη δεύτερη το έργο μας διαρκώς ελέγχει το αν είναι πατημένο το πλήκτρο κενό. Με την εντολή επανάληψης **για πάντα**, το έργο μας προλαβαίνει να κάνει πολλούς ελέγχους το δευτερόλεπτο, με συνέπεια σε ένα απλό πάτημα του «κενού», η **εάν** να «προλάβει» πατημένο το κουμπί παραπάνω από 1 φορές και συνεπώς να τρέξει αντιστοίχως περισσότερες φορές τις εντολές που περιέχει.

Επιπλέον, υπάρχουν αισθητήρες που επιτρέπουν στα αντικείμενα να αλληλεπιδρούν μεταξύ τους ανάλογα με τα διάφορα χρώματα που έχουν:

Η συνθήκη **αγγίζει το χρώμα...** ελέγχει κατά πόσο κάποιο σημείο του αντικείμενου μας αγγίζει το χρώμα που έχουμε προσδιορίσει. Το χρώμα αυτό μπορεί να βρίσκεται είτε στο σκηνικό είτε σε πάνω σε κάποιον άλλο χαρακτήρα.

π.χ. θέλουμε όποτε ο κ.Γατίδης ακουμπάει στον κορμό ενός δέντρου να γυρνάει προς τα πίσω:



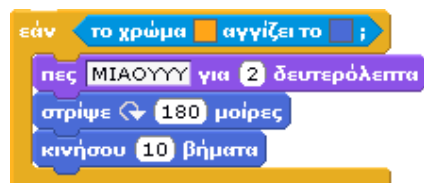
[11_π03.sb]



Πως προσδιορίζουμε το χρώμα του δέντρου; Κάνοντας κλικ στο χρωματιστό τετράγωνο του αισθητήρα, ενεργοποιείται το εργαλείο επιλογής χρώματος με το οποίο μπορούμε να κάνουμε κλικ σε οποιοδήποτε σημείο του σκηνικού και των αντικείμενων μας και να επιλέξουμε το χρώμα του συγκεκριμένου σημείου. Είναι προφανές ότι μας ενδιαφέρουν μόνο τα χρώματα που έχουν τα αντικείμενα και το σκηνικό του έργου μας και όχι οποιοδήποτε άλλο χρώμα.

Η συνθήκη **το χρώμα...αγγίζει το...** μας δίνει τη δυνατότητα να ελέγξουμε αν τα σημεία του αντικείμενού μας που έχουν ένα συγκεκριμένο χρώμα (π.χ. κόκκινα παπούτσια, τα μαύρα μαλλιά, το κίτρινο σπαθί...) αγγίζουν κάποιο άλλο χρώμα.

π.χ. θέλουμε η αλεπού να μην μπορεί να πατάει στην λίμνη της επόμενης εικόνας και αν την ακουμπάει να απομακρύνεται προς την άλλη κατεύθυνση:



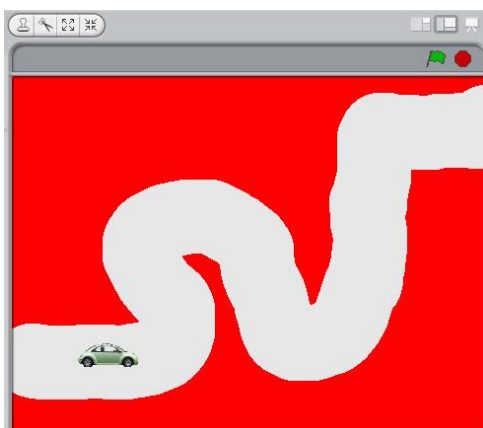
Σύμφωνα με το παραπάνω σενάριο, αν το σκούρο πορτοκαλί χρώμα της αλεπούς, ακουμπήσει στο γαλάζιο της λίμνης, τότε η αλεπού θα κάνει μιάου (!!) και θα φεύγει προς τα πίσω.



[11_π04.sb]

Προφανώς, η χρησιμότητα του συγκεκριμένου αισθητήρα είναι μεγαλύτερη όταν το αντικείμενό μας έχει διαφορετικές περιοχές χρωμάτων (π.χ. κόκκινα παπούτσια ή πράσινα μαλλιά). Σε αυτές τις περιπτώσεις, μπορούμε να δημιουργήσουμε συνθήκες που αφορούν αυτές τις διαφορετικές περιοχές).

Ας δούμε ακόμη ένα παράδειγμα. Έστω ότι θέλουμε να κατασκευάσουμε ένα παιχνίδι στο οποίο ένας χρήστης οδηγεί ένα αυτοκίνητο σε μια πίστα με τα βελάνια του πληκτρολογίου. Αν τι αυτοκίνητο ακουμπήσει την περιοχή γύρω από την πίστα που είναι κόκκινη τότε χάνει. Πως θα μπορούσαμε να υλοποιήσουμε το παιχνίδι με τις εντολές εάν και τη συνθήκη αγγίζει το χρώμα...;



Μελετήστε το παρακάτω σενάριο



[11_π05.sb]



Μπορείτε να συμπληρώσετε τα παρακάτω κενά;

-Εάν__ τότε το σκίτσο μας κινείται προς τα αριστερά

-Εάν__ τότε το σκίτσο μας κινείται προς τα κάτω.

Πότε θα εκτελεστεί η εντολή πες «Έχασα» σύμφωνα με το σενάριο;

11.2.3 Η εντολή "Εάν..." και οι τελεστές συνθηκών

Στην αρχή του κεφαλαίου είπαμε ότι οι περισσότερες γλώσσες προγραμματισμού προσφέρουν τελεστές για να δημιουργούμε πολυπλοκότερες εκφράσεις συνθηκών βάσει των οποίων να εκτελείτε ή όχι ένα σενάριο. Πως εκφράζουμε το «και», το «=» και το «ή» στο Scratch;

Οι τελεστές βρίσκονται στην παλέτα τελεστές.

Παρατηρήστε ότι τα κενά κουτάκια τους έχουν μορφή εξαγώνου, γεγονός που σημαίνει ότι παίρνουν ως παραμέτρους τους αισθητήρες ή άλλους τελεστές.

Όπως είδαμε προηγουμένως, τον τελεστή ...και... το χρησιμοποιούμε όταν θέλουμε να ελέγξουμε αν ισχύουν δύο διαφορετικές συνθήκες ταυτόχρονα. Έτσι, για να αντιδρά κ.Γατίδης όταν ο χρήστης πατάει πάνω του με το ποντίκι τότε θα πρέπει να χρησιμοποιήσουμε μια σύνθετη συνθήκη σαν αυτή που φαίνεται στην επόμενη εικόνα.



[11_π06.sb]



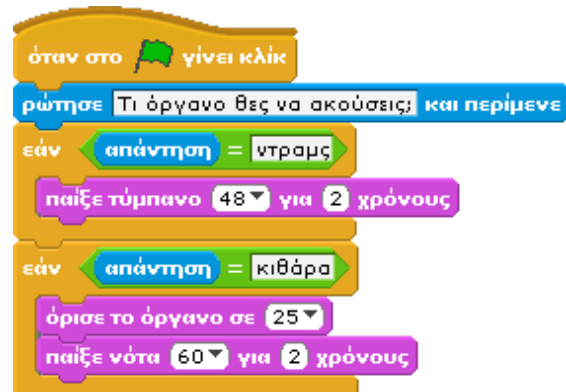
Σύμφωνα με το σενάριο, το αντικείμενό μας ελέγχει για πάντα το κατά πόσο ο δείκτης του ποντικιού βρίσκεται πάνω στο αντικείμενο ΚΑΙ ταυτόχρονα το αν πατήθηκε το αριστερό πλήκτρο του ποντικιού. Όσο δε συμβαίνουν και τα δυο μαζί, το αντικείμενό μας δεν αντιδρά. Δηλαδή, δεν αντιδρά αν κάνουμε κλικ στο ποντίκι όταν ο δείκτης του είναι μακριά από το αντικείμενο ή όταν απλά ο δείκτης του ποντικιού είναι πάνω από το αντικείμενο.

Ο τελεστής ...ή... χρησιμοποιείται όταν θέλουμε να εκτελεστεί ένα σενάριο εφόσον ισχύει τουλάχιστον μια από δυο συνθήκες. Εάν μία από τις δύο συνθήκες ισχύει ή και οι δύο ισχύουν τότε οι εντολές της εάν εκτελούνται. Στο παρακάτω παράδειγμα βλέπουμε πως ο χαρακτήρας μας αλλάζει κατεύθυνση αν ακουμπά τα όρια της οθόνης ή αν τον ακουμπήσουμε εμείς με το δείκτη του ποντικιού.



[11_π07.sb]

Τον τελεστή ...=... τον χρησιμοποιούμε όταν θέλουμε να εκτελεστεί ένα σύνολο εντολών εφόσον υπάρχει ισότητα ανάμεσα σε δυο στοιχεία. Εφόσον ισχύει η ισότητα, τότε η έκφραση αποτιμάται ως αληθής. Στο παρακάτω παράδειγμα ρωτάμε το χρήστη τι όργανο θέλει να ακούσει και ανάλογα με τη λέξη που γράφει, παράγουμε και τον αντίστοιχο ήχο.

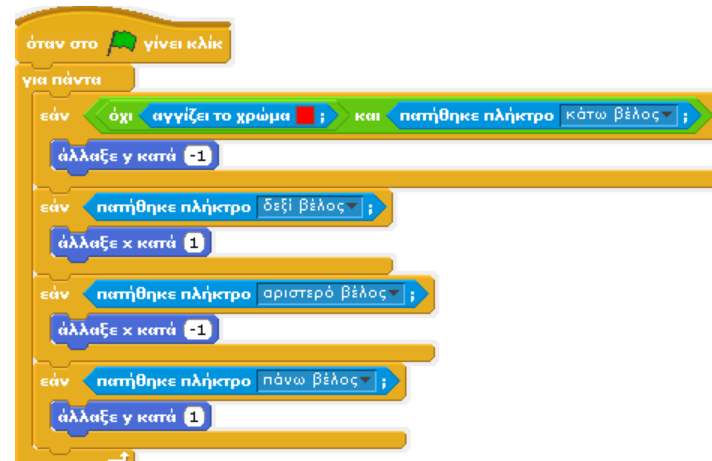


[11_π08.sb]

Αν ο χρήστης δώσει απάντηση διαφορετική από «ντραμς» ή «κιθάρα», τότε δε θα συμβεί τίποτε.

Ο τελεστής όχι... χρησιμοποιείται όταν θέλουμε να διατυπώσουμε μια συνθήκη αρνητικά. Π.χ. πως θα μπορούσαμε να πούμε σε ένα αυτοκινητάκι να κινείται μόνο εφόσον ΔΕΝ αγγίζει τα πράσινα εμπόδια. Δείτε το παρακάτω παράδειγμα. Έστω ότι έχουμε ένα ελικόπτερο που το χειρίζεται ο χρήστης με τα βελάκια του πληκτρολογίου. Υπάρχει μια πλατφόρμα από την οποία απογειώνεται και προσ-

γιώνεται το ελικόπτερο. Το ελικόπτερο δεν πρέπει να μπορούμε να το μετακινούμε προς τα κάτω όταν βρίσκεται στην πλατφόρμα. Πως μπορούμε να το καταφέρουμε αυτό;

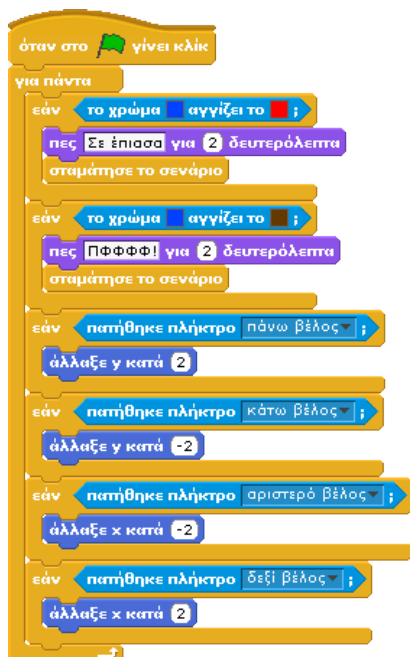


[11_π09.sb]

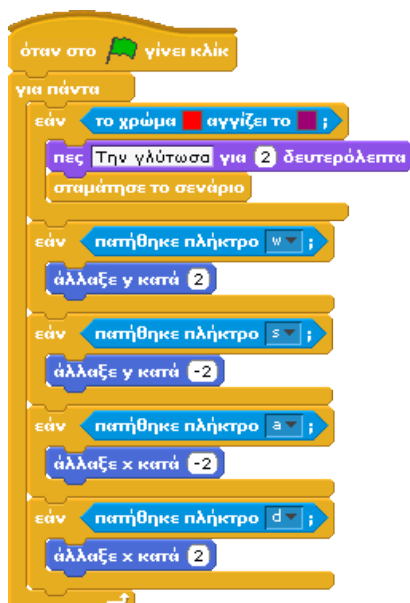
Δεν θέλουμε να κινείται προς τα κάτω το ελικόπτερο όταν αγγίζει το κόκκινο χρώμα. Εμείς όμως προσδιορίζουμε στο εσωτερικό της εάν... πότε θέλουμε να κινείται προς τα κάτω. Άρα θέλουμε να κινείται προς τα κάτω όχι μόνο όταν πατιέται το κάτω βέλος αλλά και όταν ΔΕΝ αγγίζει το κόκκινο χρώμα.

Για πρακτική εξάσκηση, ας θεωρήσουμε ότι θέλουμε να κατασκευάσουμε ένα παιχνίδι στο οποίο μια μπάλα αναπηδά στο δεξί, στο αριστερό και στο πάνω όριο της οθόνης και πρέπει να την αποκρούσουμε με μια μπάρα που ελέγχουμε με το πληκτρολόγιο ώστε να μην ακουμπήσει στο κάτω όριο της οθόνης. Έστω ότι το χρώμα της μπάρας είναι μαύρο, της μπάλας κόκκινο και έχουμε δημιουργήσει μια κίτρινη γραμμή στο κάτω όριο της οθόνης. Ποια είναι η συνθήκη που προσδιορίζει ότι έχουμε χάσει; Μπορείτε να γράψετε όλες τις συνθήκες που θα περιέχει το πρόγραμμα;

Ας υποθέσουμε τώρα ότι θέλουμε να κάνουμε ένα παιχνίδι που θα παίζεται από δυο παίκτες. Ο ένας παίχτης θα ελέγχει με το πληκτρολόγιο έναν καρχαρία και ο άλλος θα ελέγχει με άλλα πλήκτρα του πληκτρολογίου ένα μικρό ψάρι. Υπάρχουν επίσης δυο ακόμα αντικείμενα. Το ένα είναι το δόλωμα από ένα καλάμι ενός ψαρά και το άλλο η φωλιά του μικρού ψαριού. Υποθέστε ότι το χρώμα του καρχαρία είναι μπλε, του μικρού ψαριού κόκκινο, της φωλιάς μωβ και του δολώματος καφέ. Ο καρχαρίας προσπαθεί να πιάσει το μικρό ψάρι χωρίς όμως να ακουμπήσει το δόλωμα και το μικρό ψάρι να πάει στην φωλιά του. Πως θα μπορούσαμε να αναπτύξουμε το σενάριο του καρχαρία;

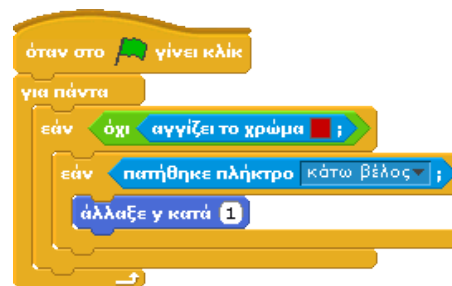


Το σενάριο για το μικρό ψάρι:



Όπως βλέπουμε όταν ο καρχαρίας αγγίζει το μικρό ψάρι λέει «Σε έπιασα» ενώ αν αγγίξει το δόλωμα λέει «ΠΦΦΦΦ». Το μικρό ψάρι αν αγγίξει την φωλιά του λέει «Τη γλίστωσα». Δοκιμάστε να αναπτύξετε μόνοι σας αυτό το έργο.

Τέλος, είναι επίσης σημαντικό να αναγνωρίσουμε ότι μπορούμε να αναπαραστήσουμε μια εντολή εάν... που περιέχει δυο συνθήκες που συσχετίζονται με τον τελεστή ΚΑΙ, ως δυο εμφωλευμένες εάν... Δείτε τα δυο σενάρια που ακολουθούν και διαπιστώστε ότι είναι ισοδύναμα:

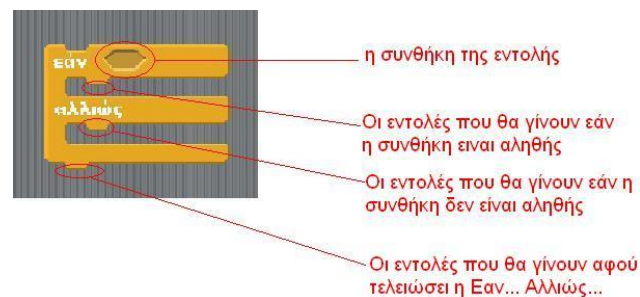


Ωστόσο η αναπαράσταση με εμφωλευμένες εάν... περισσότερο περιπλέκει το έργο μας παρά το απλουστεύει και καλό θα ήταν οι εμφωλευμένες εντολές εάν... αυτής της μορφής να αποφεύγονται.

11.2.2 Η εντολή "Εάν...Αλλιώς.."

Η εντολή εάν βασίζεται στην υπόθεση ότι σε ένα ερώτημα τύπου «Θα πάμε μετά το σχολείο να παίξουμε;» μπορεί να απαντήσουμε «Εάν δε βρέξει θα παίξουμε». Δεν υπάρχουν όμως φορές που δίνουμε απαντήσεις της μορφής «Εάν δε βρέξει, τότε θα παίξουμε αλλιώς μπορούμε να παίξουμε στο σπίτι». Με τον ίδιο τρόπο λειτουργεί η εντολή εάν ...αλλιώς... στον προγραμματισμό.

Όπως φαίνεται και από το σχήμα της, κάτω από την ετικέτα «αλλιώς», η εάν ...αλλιώς... έχει έναν επιπλέον χώρο στον οποίο μπορούμε να σύρουμε εντολές. Οι εντολές αυτές εκτελούνται στην περίπτωση που η συνθήκη δεν είναι αληθής. Ενώ δηλαδή στην εντολή εάν..., όταν η συνθήκη δεν ήταν αληθής, η εκτέλεση του προγράμματος προχωρούσε με τις εντολές που την ακολουθούσαν, στην εάν ...αλλιώς... όταν η συνθήκη δεν είναι αληθής τότε εκτελούνται οι εντολές που βρίσκονται στο δεύτερο τμήμα της.



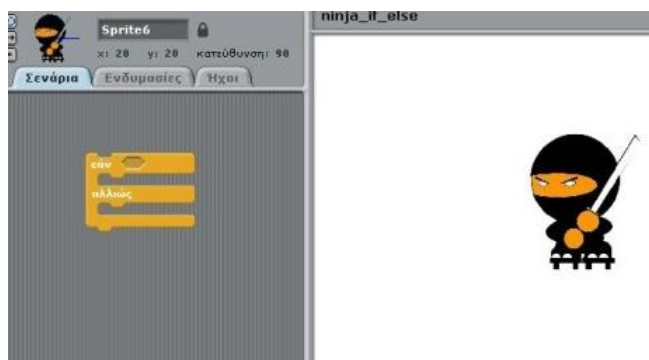
Όταν εισάγουμε δηλαδή μια εάν...αλλιώς... στο σενάριό μας, τότε κάθε φορά που έρχεται η σειρά εκτέλεσής της, ένα από τα δυο μικροσενάρια εντολών που περιέχει θα εκτελείται. Αντίθετα, στην εάν..., οι εντολές εκτελούνταν μόνο εφόσον η συνθήκη ήταν αληθής. Άρα, πρέπει να έχουμε στο νου μας ότι η εάν...αλλιώς... μας επιτρέπει ουσιαστικά να δημιουργούμε δυο διαφορετικές εκδοχές του σεναρίου μας με την εκτέλεση δυο διαφορετικών τμημάτων εντολών ανάλογα με τις συνθήκες. Αυτό μας δίνει τρομακτική ευελιξία.

Έχουμε την τιμή να σας παρουσιάσουμε τον νιντζά I-fel-se-kun. Έχοντας μάθει την τέχνη της κατασκοπείας και της μυστικότητας όταν πάμε να τον περιεργαστούμε εξαφανίζεται. Με λίγα λόγια ο νιντζά εξαφανίζεται κάθε φορά που το ποντίκι περνά από πάνω του.

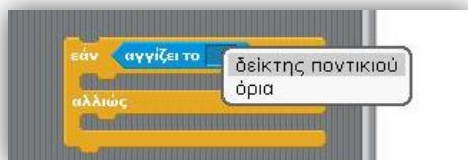


Πώς θα μπορούσαμε να δώσουμε στον ninja την παραπάνω ικανότητα στο Scratch χρησιμοποιώντας την εάν...αλλιώς...; Ας κατασκευάσουμε βήμα-βήμα το έργο μας:

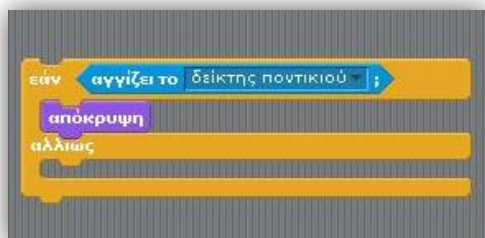
Πρώτα εισάγουμε την εάν...αλλιώς... στο πρόγραμμα του ninja.



Τι θέλουμε να ελέγχουμε; Εάν το ποντίκι ακουμπά τον ninja. Αν τον ακουμπά, ο ninja θα εξαφανίζεται αλλιώς (αν δεν τον ακουμπά) θα πρέπει να εμφανίζεται. Άρα η συνθήκη της εντολής θα σχηματιστεί χρησιμοποιώντας τον αισθητήρα αγγίζει το... και θα επιλέξουμε «δείκτη ποντικιού» όπως φαίνεται στην παρακάτω εικόνα.



Συνεπώς, η συνθήκη θα είναι αληθής όταν το αντικείμενό μας αγγίζει το δείκτη του ποντικιού. Όταν συμβαίνει αυτό, θέλουμε το αντικείμενό μας να εξαφανίζεται και αυτό θα το επιτύχουμε με την εντολή απόκρυψη από την παλέτα Όψεις.

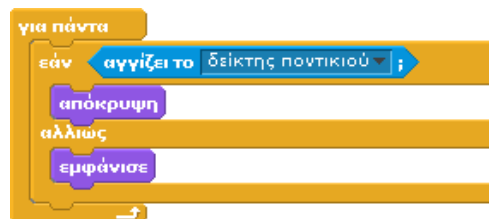


Εάν τώρα ο ninja δεν αγγίζει το δείκτη του ποντικιού, θα εκτελεστούν οι εντολές της «αλλιώς». Εισάγουμε στις επόμενες δαγκάνες την εντολή εμφάνισε από την παλέτα Όψεις.

Με αυτό τον τρόπο δημιουργήσαμε στο πρόγραμμά μας τη λογική:

Εάν ο ninja αγγίζει τον δείκτη του ποντικιού τότε απόκρυψη αλλιώς εμφάνισε.

Αν τρέξουμε το έργο μας προσθέτοντας την εντολή όταν στην πράσινη σημαία γίνει κλικ τότε τι θα συμβεί; Θα γίνει ο έλεγχος μια φορά και το έργο μας θα τελειώσει. Ο έλεγχος θα έχει εκτελεστεί ΜΙΑ φορά και μετά θα σταματήσει. Εμείς όμως θέλουμε ο έλεγχος να τρέχει για πάντα. Έτσι εισάγουμε την εντολή για πάντα για άλλη μια φορά όπως φαίνεται στην επόμενη εικόνα.



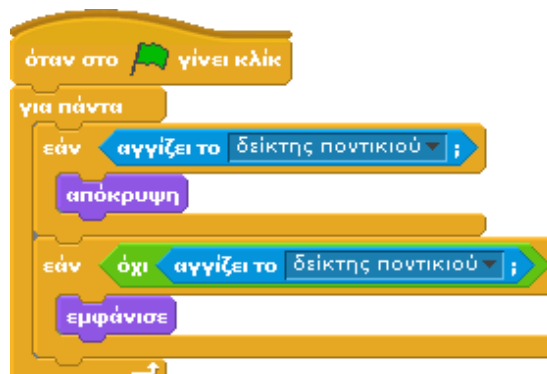
[11_π10.sb]



Το πρόγραμμά μας είναι έτοιμο και ο ninja εξαφανίζεται όταν ο δείκτης του ποντικιού βρίσκεται από πάνω του.

Μπορούμε να υλοποιήσουμε την εάν...αλλιώς... από εντολές εάν...; Μήπως μια εάν...αλλιώς... είναι ουσιαστικά δυο εάν...; Τι σημαίνει η έκφραση αλλιώς; Σημαίνει αν ΔΕΝ ισχύει η αρχική συνθήκη.

Μελετήστε και τρέξτε το επόμενο σενάριο. Σας θυμίζουμε ότι ο τελεστής όχι αντιστρέφει ουσιαστικά το αποτέλεσμα της συνθήκης που περιέχει. Άρα η «αλλιώς» έχει γίνει «όχι η αρχική» που είναι πρακτικά το ίδιο πράγμα.



Ο ninja αντιδρά ακριβώς όπως και πριν.

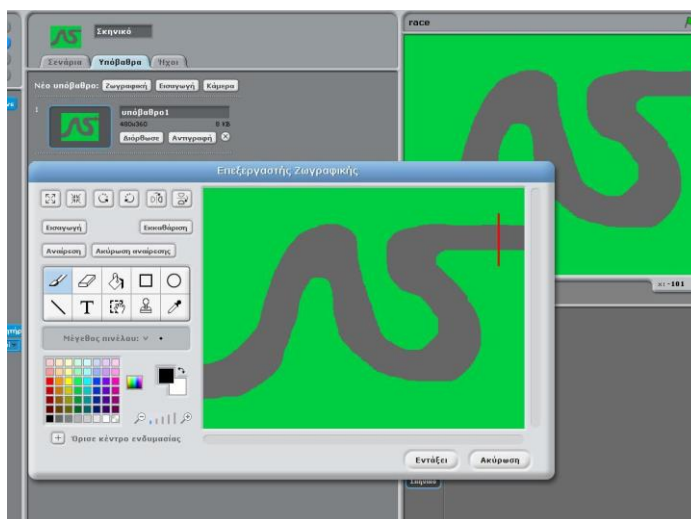
Γιατί να έχουμε την εάν...αλλιώς... αφού μπορούμε να την αντικαταστήσουμε με δύο εάν...; Η απάντηση είναι ότι στη δεύτερη περίπτωση, ο υπολογιστής μας αφού ελέγχει μία φορά τη συνθήκη μετά πρέπει να την ελέγχει και δεύτερη φορά στην δεύτερη εάν... που σημαίνει περισσότερη δουλειά για τον επεξεργαστή. Το έργο μας δεν είναι τόσο αποδοτικό. Όταν τα έργα σας αρχίζουν να μεγαλώνουν τότε αυτά τα στοιχεία απόδοσης θα αρχίσουν να παίζουν σημαντικό ρόλο. Με την εάν...αλλιώς... ο υπολογιστής αποφεύγει τον δεύτερο έλεγχο γλιτώνοντας έτσι χρόνο και ενέργεια.

11.3 Παραδείγματα.

Παράδειγμα 1°: Οδηγούμε ένα αυτοκινητάκι σε μια πίστα η οποία θέλουμε να μας ενημερώνει πότε βρισκόμαστε εκτός του

βασικού χρόνου ενώ όταν φτάσουμε στο σημείο τερματισμού να μας αναφέρει ότι ολοκληρώθηκε το παιχνίδι.

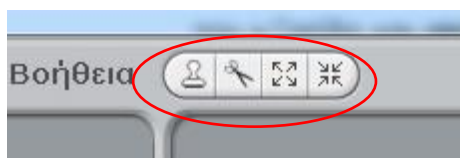
Αρχίζουμε σχεδιάζοντας στον επεξεργαστή ζωγραφικής την πίστα που θέλουμε:



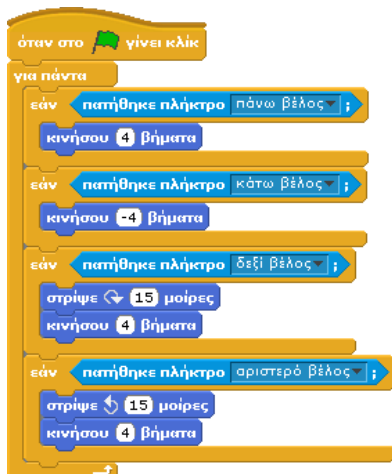
Μετά εισάγουμε ένα αντικείμενο που θα αντιπροσωπεύει το αμάξι μας.



Χρησιμοποιούμε, εφόσον χρειάζεται τη μπάρα εργαλείων που βρίσκεται πάνω από την οθόνη του Scratch για να προσδιορίσουμε το κατάλληλο μέγεθος για το αντικείμενό μας.



Για να κινήσουμε το αμάξι με τα βελάκια του πληκτρολογίου χρησιμοποιούμε διαδοχικές εάν... μέσα σε μια για πάντα και αξιοποιούμε τον αισθητήρα πατήθηκε πλήκτρο... ως συνθήκη για τις εντολές:



Σύμφωνα με τα παραπάνω, εάν πατηθεί το πάνω βέλος, τότε το αυτοκινητάκι μας μετακινείται προς τα εμπρός, με το πίσω «βάζει την όπισθεν» και με τα δεξιά και αριστερά στρίβει και μετακινείται ταυτόχρονα.

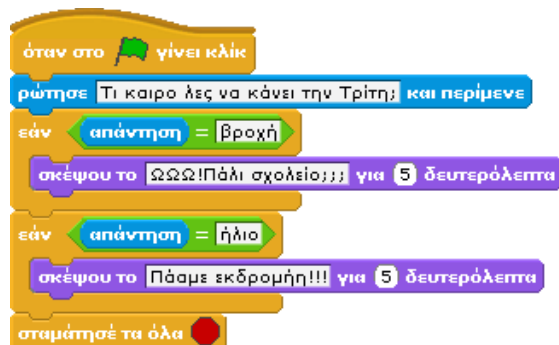
Η περιγραφή του προβλήματός μας όμως μας έλεγε ότι θα πρέπει να ενημερώνουμε τον οδηγό όταν βγαίνει εκτός της ασφάλτου αλλά και όταν φτάνει στη γραμμή τερματισμού.



[11_π11.sb]

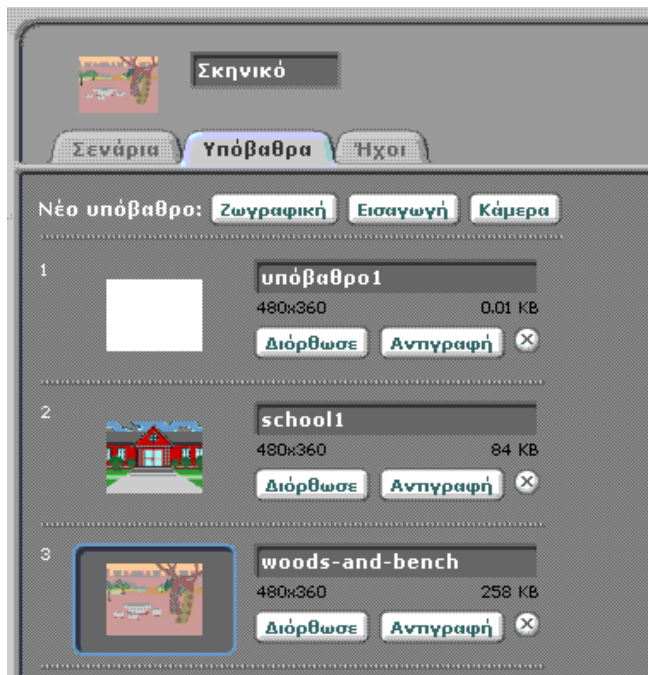


Παράδειγμα 2°: Έστω ότι έχουμε τον κ. Γατίδη σε ένα κενό σκηνικό και θέλουμε να ρωτάει τι καιρό θα έχει την ημέρα που θα πάει την εκδρομή του. Ο κ. Γατίδης θα πρέπει να αντιδρά ανάλογα με τη πρόγνωση του καιρού που θα προσδιορίζουμε εμείς. Μπορούμε να υλοποιήσουμε το μικρό αυτό διάλογο χρησιμοποιώντας τις εντολές ρώτησε...και περίμενε για την ερώτηση του κ.Γατίδη και απάντηση για να κρατήσουμε την απάντηση που θα δώσει ο χρήστης. Μελετήστε το παρακάτω σενάριο:



Μπορείτε να αιτιολογήσετε γιατί δε χρησιμοποιούμε την εάν...αλλιώς... στο παραπάνω παράδειγμα; Είναι πάλι ο προγραμματιστής μας αφηρημένος;

Πολύ μονότονο δεν είναι το σκηνικό; Τι θα λέγατε να βλέπαμε να υλοποιούνται οι σκέψεις του κ. Γατίδη; Κρατώντας το σενάριο λοιπόν για τον κ. Γατίδη ως έχει, θα χρειαστεί απλά να εισάγουμε δύο διαφορετικά υπόβαθρα στο σκηνικό μας, ένα για την εκδρομή (woods-and-bench) και ένα για το σχολείο (school1).



Έπειτα, μπορούμε να δημιουργήσουμε ένα σενάριο στο σκηνικό το οποίο ανάλογα με την απάντηση του κ. Γατίδη να αλλάζει υπόβαθρο.



Ας δοκιμάσουμε τώρα να βάλουμε και μια ακόμα συνθήκη για το εάν τελικά θα πάει ο κ. Γατίδης εκδρομή ή όχι. Δε χρειάζεται μόνο ο καλός καιρός, μα και η συγκατάθεση των γονιών του! Το σενάριο του κ.Γατίδη θα μείνει ίδιο μέχρι την δεύτερη εντολή **εάν....** Μέσα στη δεύτερη εντολή **εάν....**, εφόσον δηλαδή έχουμε απαντήσει ότι θα έχει ήλιο, ο κ.Γατίδης θα μας ρωτά αν θα τον αφήσουν οι γονείς του να πάει εκδρομή και ανάλογα με την απάντησή μας θα αντιδρά.



[11_π12.sb]

Περίληψη

Στο κεφάλαιο συζητήσαμε θεωρητικά για το πώς χρησιμοποιούμε τον έλεγχο στην καθημερινότητά μας και ποια είναι η χρησιμότητά του στον προγραμματισμό. Οι εντολές ελέγχου **εάν...** και **εάν... αλλιώς...** μας επιτρέπουν να εκτελούμε μικροσενάρια εντολών εφόσον ισχύουν ορισμένες συνθήκες. Μας δίνουν δηλαδή τη δυνατότητα να δημιουργούμε προγράμματα τα οποία διαφοροποιούν την εκτέλεσή τους ανάλογα με την είσοδο του χρήστη. Οι συνθήκες αυτές μπορεί να είναι είτε απλές είτε σύνθετες με τη χρήση των κατάλληλων τελεστών (και, ή, όχι). Στο συγκεκριμένο κεφάλαιο διερευνήσαμε κυρίως συνθήκες που αφορούν αισθητήρες, εντολές δηλαδή που μας ενημερώνουν για το τι αντιλαμβάνεται ο χαρακτήρας μας (π.χ. άγγιξε κάποιο χρώμα, άγγιξε το δείκτη του ποντικιού κτλ.). Ίσως ήρθε η ώρα να λέμε ότι είμαστε σε θέση να δημιουργήσουμε ευφυή έργα; Τρέξε τα παραδειγματικά έργα με τους συμμαθητές σας, δημιουργήστε παραλλαγές τους και θα διαπιστώστε ότι πραγματικά αρχίζουμε να γνωρίζουμε τις εξαιρετικές δημιουργικές δυνατότητες που μας προσφέρει ο προγραμματισμός.

Ερωτήσεις

1) Διακρίνετε αν οι παρακάτω δηλώσεις είναι σωστές ή λανθασμένες

- Μπορούμε να εισάγουμε μια **Εάν...** μέσα σε μια άλλη **Εάν...**
- Σε μια **Εάν...** μπορούμε να εισάγουμε 2 συνθήκες.

2) Προσπαθήστε να εντοπίσετε παραδείγματα εφαρμογής της εντολής **Εάν....** σε προγράμματα που χρησιμοποιείτε καθημερινά στον υπολογιστή σας.

3) Τι θα γίνει αν από μια συνθήκη αλλάξουμε τον τελεστή **...και...** σε τελεστή **...ή...**;

4) Μετατρέψτε τις παρακάτω **Εάν...** σε μία **Εάν...αλλιώς...**

Εάν (αγγίζει δείκτη ποντικιού) {Παίξε ήχο}

Εάν ΔΕΝ (αγγίζει δείκτη ποντικιού) {Πες ησυχία}

Δημιουργήστε τα αντίστοιχα σενάρια στο Scratch.

5) Υποθέστε ότι δεν υπάρχει η εντολή **Εάν ...αλλιώς**. Κρίνετε πως το γεγονός αυτό θα επηρέαζε τα προγράμματα σας.

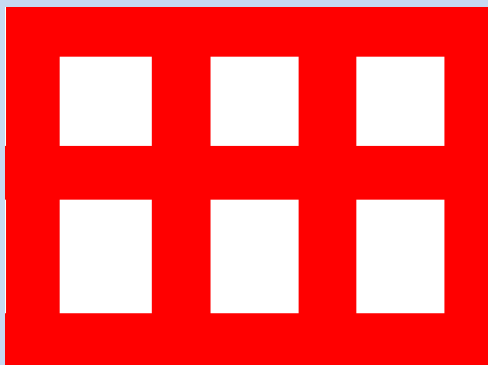
Δραστηριότητες

1) Δημιουργήστε το ακόλουθο παιχνίδι. Ο χρήστης έχει ένα ελικόπτερο το οποίο πρέπει να το απογειώσει από την πλατφόρμα με κόκκινο χρώμα και να το προσγειώσει στην πλατφόρμα με κίτρινο χρώμα. Η μετακίνηση γίνεται με τα τρία βελάκια του πληκτρολογίου (πάνω, δεξιά, αριστερά) ενώ όταν δεν πατά το πάνω βέλος ο χρήστης, το ελικόπτερο πέφτει με σταθερό ρυθμό (εκτός από την περιοχή των πλατφορμών). Επιπλέον, στη διαδρομή υπάρχουν διάφορα εμπόδια που αν τα ακουμπήσει ο χρήστης χάνει.

2) Σκοπός αυτής της δραστηριότητας είναι η δημιουργία ενός λαβυρίνθου με ένα κρυμμένο εμπόδιο. Ο λαβύρινθος θα έχει την μορφή της εικόνας που παρουσιάζεται παρακάτω και θα δημιουργηθεί με τη βοήθεια του επεξεργαστή ζωγραφικής. Μικρύνετε το αντικείμενο γάτα ώστε να χωράει στις κόκκινες γραμμές

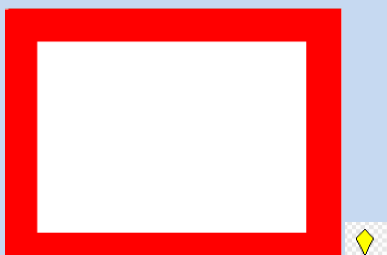


του λαβυρίνθου. Εισάγετε ένα δεύτερο αντικείμενο μέσω της ζωγραφικής το οποίο θα έχει το ίδιο χρώμα με το λαβύρινθο και πρέπει να τοποθετηθεί σε σημείο του λαβυρίνθου ώστε να μην ορατό. Ο χρήστης θα κινεί τη γάτα μέχρι να βρει το κρυμμένο εμπόδιο.



3) Σε αυτή τη δραστηριότητα θα δημιουργήσετε ένα πίνακα όπως παρουσιάζεται στην εικόνα που ακολουθεί, πάνω στον οποίο θα μπορείτε να ζωγραφίσετε παραλληλόγραμμα σχήματα και να αφήνετε αποτυπώματα με τη βοήθεια ενός αντικειμένου ρόμβου (τελευταία εικόνα), τον οποίο θα κινείτε με τα βέλη. Ο ρόμβος όταν θα αγγίζει το κόκκινο χρώμα θα σταματά και δεν θα ζωγραφίζει τίποτε πάνω στο κάδρο μας. Επιπλέον φροντίστε με τη χρήση της εντολής εάν...

1. Εάν πατηθεί το πλήκτρο c να καθαρίζει ο πίνακας.
2. Εάν πατηθεί το πλήκτρο d ο ρόμβος να ζωγραφίζει καθώς κινείται.
3. Εάν πατηθεί το πλήκτρο u ο ρόμβος να μη ζωγραφίζει καθώς κινείται.
4. Εάν πατηθεί το πλήκτρο κενό ο ρόμβος να αφήνει το αποτύπωμά του.
5. Εάν πατηθεί το πλήκτρο r το χρώμα της πέννας να γίνεται κόκκινο.
6. Εάν πατηθεί το πλήκτρο g το χρώμα της πέννας να γίνεται πράσινο.
7. Εάν πατηθεί το πλήκτρο p το χρώμα της πέννας να γίνεται μωβ.
8. Εάν πατηθεί το πλήκτρο y το χρώμα της πέννας να γίνεται κίτρινο.
9. Εάν πατηθεί το πλήκτρο s το μέγεθος της πέννας να αλλάζει κατά -1.
10. Εάν πατηθεί το πλήκτρο b το μέγεθος της πέννας να αλλάζει κατά 1.





Κεφάλαιο 12: Νέες επαναλήψεις!

Σε αυτό το κεφάλαιο:

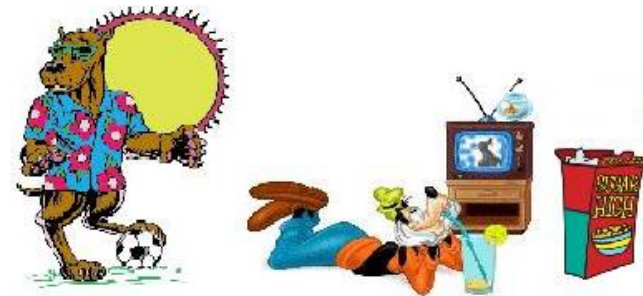
- 12.1 Εντολή «για πάντα εάν...»
- 12.2 Εντολή «επανέλαβε ώσπου...»
- 12.3 Εντολή «περίμενε ώσπου...»
- 12.4 Συνδυαστικά παραδείγματα

«Ό, τι επαναλαμβάνουμε, μας καθορίζει»
(Αριστοτέλης)

12.1. Εντολή «για πάντα εάν...»

Όπως αναφέρθηκε και στο κεφάλαιο 10, η έννοια της επανάληψης παίζει κυρίαρχο ρόλο στη λογική της καθημερινότητάς μας. Πέρα όμως από τις δυο συγκεκριμένες μορφές επαναλήψεων που εξετάσαμε, δηλαδή τις επαναλήψεις που διαρκούν για πάντα (εντολή για πάντα) και τις επαναλήψεις που συμβαίνουν για γνωστό και συγκεκριμένο αριθμό φορών (εντολή επανέλαβε X), υπάρχουν και μορφές επαναλήψεων που διαρκούν όσο ισχύουν ορισμένες συνθήκες. Σε αυτές τις μορφές επαναλήψεων που γνωρίζουμε από την αρχή τον αριθμό των επαναλήψεων τους.

Πολλές φορές ξεκινάμε να βλέπουμε κάθε βδομάδα μια καινούρια τηλεοπτική σειρά που μας αρέσει εφόσον όμως δεν έχουμε κανονίσει να βγούμε με τους φίλους μας. Ή παίζουμε μπάλα συγκεκριμένη ημέρα, όταν όμως έχει καλό καιρό.



Υπάρχουν δηλαδή κάποιες ενέργειες που επαναλαμβάνονται μόνο εφόσον ισχύουν ορισμένες συνθήκες και για το λόγο αυτό δε γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων που θα συμβούν.

Πως θα μπορούσαμε να κάνουμε τον κ.Γατίδη να πηγαίνει προς το δείκτη του ποντικιού κάθε φορά που κρατάμε πατημένο το δεξί πλήκτρο του ποντικιού μας; Θέλουμε δηλαδή να κινείται για όσο όμως πατάμε το ποντίκι μας. Θα κυνηγάει δηλαδή το ποντίκι όσο του λέμε που βρίσκεται δηλαδή! Λογικό... Με βάση τις προηγούμενες γνώσεις μας κάτι τέτοιο δεν είναι δύσκολο. Χρειαζόμαστε μια εντολή για πάντα η οποία μέσα της θα περιέχει μια εντολή εάν... το ποντίκι πατήθηκε. Όταν θα είναι πατημένο το ποντίκι, ο γάτος μας θα δείχνει (κατεύθυνση) προς το

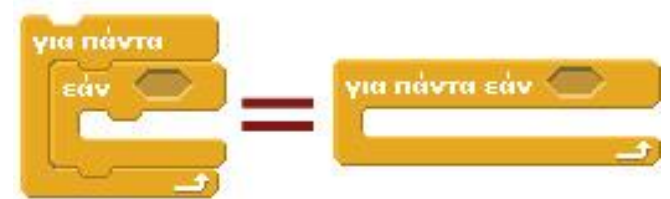
δείκτη του ποντικιού και θα κινείται χρησιμοποιώντας την εντολή κινή-



σου...βήματα. Δείτε το αριστερό από τα δυο σενάρια της παρακάτω εικόνας:



Αυτή η υπό συνθήκη επανάληψη είναι πολύ συχνή στον προγραμματισμό και για αυτό το Scratch μας δίνει την εντολή για πάντα εάν... που εμφανίζεται στο δεξί σενάριο. Η εντολή αυτή συμπεριφέρεται όπως περίπου μια εντολή για πάντα που περιέχει μέσα της μόνο μια εάν... και μας επιτρέπει να εκτελούμε ένα σύνολο εντολών για όσο ισχύει μια συνθήκη.



Ας δούμε όμως τη χρήση της εντολής σε ένα μεγαλύτερο παράδειγμα.

Όλοι κάποια στιγμή της ζωής μας έτυχε να παίξουμε με εφαρμογές κινητού. Μία αρκετά γνωστή εφαρμογή είναι η FaceWarp. Αυτή η εφαρμογή δίνει τη δυνατότητα στο χρήστη να παραμορφώνει είτε μία φωτογραφία που έχει στους φακέλους του είτε μία φωτογραφία που θα τραβήξει εκείνη τη στιγμή. Μεγαλώννει και μικραίνει τη μύτη και το χαμόγελο, αλλάζει το σχήμα του προσώπου και γενικά αλλάζει όλη τη φωτογραφία προκαλώντας πολύ γέλιο στην παρέα.

Τί σχέση όμως έχουν όλα αυτά με το Scratch;

Στο κεφάλαιο 7 μελετήσαμε τις εντολές της παλέτας **Όψεις** που μας έδιναν αντίστοιχα χαρακτηριστικά. Τί θα λέγατε αν συνδυάζαμε εκείνες τις εντολές, με την εντολή επανάληψης **για πάντα εάν...** για να φτιάξουμε μία εφαρμογή παρόμοια με αυτή των κινητών; Πιστεύετε ότι κάτι τέτοιο είναι δυνατό; Για πάμε να δούμε.

Σαν φωτογραφία που θα πειράζουμε, επιλέγουμε το αντικείμενο squareface από τη βιβλιοθήκη του Scratch. Αυτή η επιλογή δεν είναι δεσμευτική, καθώς εσείς μπορείτε να βάλετε μία δική σας φωτογραφία ή μία φωτογραφία κάποιου φίλου σας.



Ορίζουμε τη συγκεκριμένη φωτογραφία ως υπόβαθρο στο σκηνικό μας, αφού πρώτα τη μεγεθύνουμε για να καλύπτει όλη την οθόνη. Στη συνέχεια δημιουργούμε 4 αντικείμενα με την ίδια μορφή και μικρότερο μέγεθος. Αυτά θα είναι τα αντικείμενα που θα τροποποιήσουμε.

Με την εντολή **πήγαινε στο x,y** τοποθετούμε τα 4 αντικείμενά μας στις 4 γωνίες του σκηνικού μας για την εκκίνηση του έργου και μέσω της εντολής **κινήσου ομαλά Χ δεύτ. στο x,y** τα μετακινούμε στο κέντρο του σκηνικού ώστε να ξεκινήσουμε με μία φαντασμαγορική αρχή! Αφού τα τοποθετήσαμε σωστά, πάμε να κάνουμε στο καθένα και μία μικρή αλλαγή!

Θέλουμε κάθε φορά που πατάμε το ποντίκι το κάθε αντικείμενο να κάνει μία αλλαγή στη μορφή του.

Έστω ότι το πρώτο αντικείμενο περιστρέφεται κατά 20 μοίρες, το δεύτερο χρησιμοποιεί το εφέ στροβίλιση, το τρίτο τα εφέ ψηφιδωτό και άλλαξε χρώμα και το τέταρτο αλλάζει το μέγεθός του.

Και τώρα αναρωτιόμαστε: Θέλουμε αυτή η αλλαγή να γίνει μία φορά; Όχι! Θέλουμε κάθε φορά που κάνουμε κλικ να γίνονται οι αλλαγές. Άρα, σίγουρα χρειαζόμαστε επανάληψη! Θέλουμε κάποια άλλη συνθήκη; Όχι. Μπορούμε λοιπόν να χρησιμοποιήσουμε την **Για πάντα-εάν**; Φυσικά.

Το σκηνικό μας αρχικά έχει την εξής μορφή:



Ένα στιγμιότυπο αυτού είναι όταν έχουμε κάνει κλικ στο ποντίκι μας:



ενώ τα σενάρια για κάθε αντικείμενο είναι:



[12_π01.sb]



Υπάρχουν πολλά περιθώρια βελτίωσης των σεναρίων. Μπορείτε να εντοπίσετε τα προβλήματά τους ή ευκαιρίες για να γίνουν πιο αποδοτικά;

Είναι σημαντικό να παρατηρήσουμε ότι και η εντολή **για πάντα εάν...**, όπως η εντολή **για πάντα**, δεν δέχεται άλλες εντολές από κάτω της, δεν έχει την χαρακτηριστική εξοχή στην οποία μπορούμε να «κολλήσουμε» μια εντολή. Αυτό συμβαίνει γιατί όταν ο έλεγχος του έργου μας φτάσει στη συγκεκριμένη εντολή, τότε μέχρι το τέλος του έργου το Scratch θα ελέγχει τη συνθήκη και εφόσον ικανοποιείται θα εκτελεί τις εντολές που περιέχονται μέσα σε αυτήν.

Ας θεωρήσουμε ότι πρέπει να υλοποιήσουμε στο Scratch την εξής περιγραφή «Αν το ελεφαντάκι ακουμπήσει το έντομο, τότε το έντομο πρέπει να πάει πίσω στην αρχική του θέση και να πει ότι έχασε». Σε ποιου αντικείμενου το σενάριο, θα εισάγουμε την

εντολή εάν... ; Αρχικά θα σκεφτόμασταν να εισάγουμε μια εντολή εάν... στο αντικείμενο ελεφαντάκι. Αλλά αυτό θα μας οδηγούσε σε ένα αδιέξοδο, καθώς από το σενάριο που αφορά το ελεφαντάκι δεν μπορούμε να κινήσουμε το έντομο αλλά μόνο το ελεφαντάκι.

Το σημείο αυτό θέλει προσοχή. Οι συνθήκες των εντολών εάν... μπορούν να περιέχουν αντικείμενα με οποιαδήποτε σειρά αλλά εμείς ενδιαφερόμαστε να προγραμματίσουμε το αποτέλεσμα, το τι θα συμβεί αν ισχύσει η συνθήκη. Αυτά που θα συμβούν ποιο αντικείμενο αφορούν; Το έντομο. Σε αυτό το αντικείμενο θα εισάγουμε την αντίστοιχη εντολή.

12.2 Εντολή «επανάλαβε ώσπου»

Είναι 11 η ώρα το πρωί και δεν έχετε σχολείο. Αποφασίζετε λοιπόν να κοιμηθείτε λίγο παραπάνω! Ξαφνικά ακούτε από μέσα μία φωνή πολύ γνωστή που να λέει: «Ξύπνα! Πρέπει να σηκωθείς για να συμμαζέψεις το δωμάτιο σου!». Και ναι! Είναι η φωνή της μητέρας. Τί σχέση όμως μπορεί να έχει αυτό με το μάθημά μας;

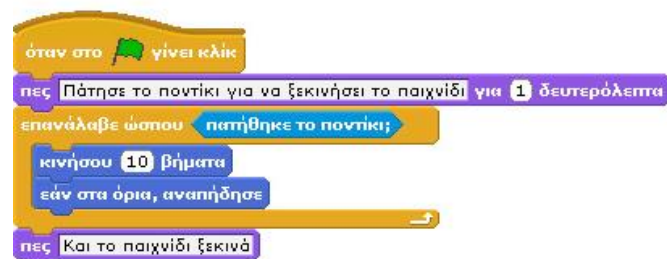
Επανάληψη: Η μαμά θα συνεχίσει να φωνάζει ότι πρέπει να σηκωθείτε για να συμμαζέψετε το δωμάτιο.

Συνθήκη: Μέχρι να σηκωθείτε από το κρεβάτι.

Το σίγουρο λοιπόν είναι ότι θα συνεχίσει να φωνάζει μέχρι εσείς να ξυπνήσετε και να σηκωθείτε από το κρεβάτι. Δηλαδή μέχρι η συνθήκη (αν σηκωθήκατε από το κρεβάτι) να γίνει αληθής!!!

Στην παλέτα Έλεγχος υπάρχει η εντολή της μαμάς ☺ επανάλαβε ώσπου. Μέσω αυτής της επιλογής μπορούμε να εξασφαλίσουμε τη συνεχή επανάληψη εντολών ώσπου η συνθήκη ελέγχου να γίνει αληθής. Για παράδειγμα, μπορούμε να σκεφτούμε την πρώτη οθόνη ενός παιχνιδιού, στην οποία ο χαρακτήρας κινείται ασταμάτητα μέχρι να πατήσουμε το ποντίκι και να ξεκινήσει το παιχνίδι. Πως θα μπορούσαμε να υλοποιήσουμε κάτι τέτοιο:

Μελετήστε το παρακάτω πολύ απλό σενάριο:



Η επανάλαβε ώσπου κάνει τον πρωταγωνιστή μας να κινείται μεταξύ των ορίων της οθόνης μέχρι να πατήσουμε το ποντίκι.

Η εντολή επανάλαβε ώσπου δεν επαναλαμβάνει τις εντολές που περιέχει για συγκεκριμένο αριθμό φορών όπως η επανάλαβε Χ. Αντίθετα, στην επανάλαβε ώσπου:

- ✓ αν η συνθήκη είναι αληθής από την αρχή, τότε οι περιεχόμενες εντολές δεν θα εκτελεστούν καμία φορά!
- ✓ αν η συνθήκη είναι ψευδής, θα επαναλαμβάνει τις εντολές μέχρι η συνθήκη να γίνει αληθής.

Μοιάζει επίσης με την εντολή για πάντα εάν... με τη διαφορά ότι

- ✓ οι συνθήκες είναι αντίστροφες, αφού στην για πάντα εάν... αν η συνθήκη είναι αληθής, πραγματοποιούνται οι επαναλήψεις, ενώ στην εντολή επανάλαβε ώσπου οι επαναλήψεις πραγματοποιούνται όσο η συνθήκη είναι ψευδής
- ✓ ο έλεγχος της συνθήκης στην επανάλαβε ώσπου δεν γίνεται καθ' όλη τη διάρκεια του έργου αλλά μέχρι να ικανοποιηθεί η συνθήκη. Αν ικανοποιηθεί η συνθήκη το πρόγραμμα εκτελεί την επόμενη εντολή και για αυτό επιτρέπεται να προστεθούν άλλες εντολές από κάτω της. Αντίθετα, όταν ξεκινήσει μια για πάντα εάν... ο έλεγχος δεν σταματά παρά μόνο όταν τερματιστεί ο έλεγχος.

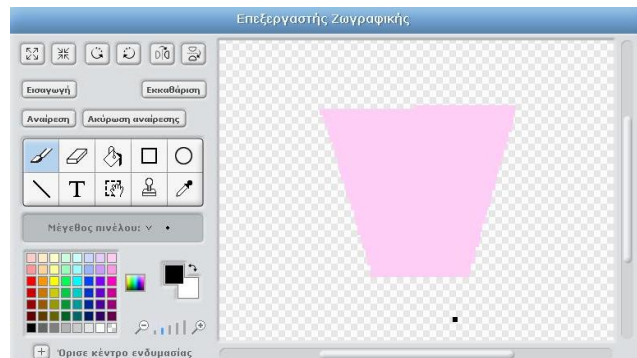
Ας δούμε τη χρήση της μέσα σε ένα παράδειγμα. Θέλουμε ο ήρωας μας να μαζεύει όλα τα επικίνδυνα κοχύλια που δεν σταματούν να κουνιούνται μέσα σε έναν κουβά για να καθαρίσει την παραλία!

Τί χρειαζόμαστε; Χρειαζόμαστε σίγουρα κάποιον να κάνει τη βρώμικη(?) δουλειά! Έστω ότι επιλέγουμε το αντικείμενο fantasy1-b από τον κατάλογο fantasy της βιβλιοθήκης αντικειμένων του Scratch.



Επίσης πρέπει να έχουμε έναν κουβά για να μαζεύουμε τα κοχύλια. Ας τον δημιουργήσουμε εμείς!

Με τον επεξεργαστή ζωγραφικής φτιάχνουμε ένα απλό τραπέζιο σε μορφή κουβά και το χρωματίζουμε στο χρώμα της αρεσκείας μας. Εμείς θα επιλέξουμε το ροζ κι έτσι θα έχουμε:



Τέλος χρειαζόμαστε τα κοχύλια τα οποία επιλέγουμε να είναι τα (κατάλογος animals, μορφή clam1):



Φυσικά για σκηνικό επιλέγουμε μία εξωτική παραλία! Το παιχνίδι μας θα ξεκινήσει από την εξής κατάσταση:



Πάμε όμως να δούμε τι θέλουμε να κάνει το κάθε αντικείμενο ξεχωριστά.

Ο... συλλέκτης: Σαν χρήστες της εφαρμογής, θέλουμε να μπορούμε να μετακινούμε το πρωταγωνιστή μας με τα βελάκια προς όποια κατεύθυνση επιθυμούμε. Επιπρόσθετα, ο πρωταγωνιστής μας θα πρέπει να βρίσκεται σε συγκεκριμένη θέση στην αρχή του παιχνιδιού. Το σενάριο του λοιπόν είναι το παρακάτω:



Κοχύλια: Θέλουμε να δώσουμε στα κοχύλια μία συνεχόμενη κίνηση μπρος-πίσω για να τα κάνουμε να μοιάζουν λίγο επικίνδυνα! Αυτή η κίνηση σταματάει μόλις ο συλλέκτης τα πιάσει (στη γλώσσα μας = ακουμπήσει) και αυτά αμέσως μετακινούνται στον ροζ κουβά μας! Θέλουμε δηλαδή να κινούνται συνέχεια, ώπου ο συλλέκτης να τα πιάσει. Αυτή είναι δουλειά της **επανάλαβε ώπου...** Επομένως, αφού δημιουργήσουμε 5 κοχύλια, και τα τοποθετούμε στην αρχή του έργου σε συγκεκριμένες θέσεις, χρησιμοποιώντας την εντολή **επανάλαβε ώπου** τα κάνουμε να κινούνται δεξιά και αριστερά με τη χρήση της εντολής **κινήσου ομαλά Χ δεύτ. στο x,y**. Η κίνηση θα σταματά όταν θα τα αγγίζει ο συλλέκτης, δηλαδή θα κινούνται «**ώπου να τα αγγίξει ο συλλέκτης**». Όταν τα πιάνει, θα μετακινούνται στον κουβά. Τα σενάρια όλων των κοχυλιών είναι πανομοιότυπα και οι μόνες τιμές που αλλάζουν είναι κάθε φορά οι αρχικές και τελικές θέσεις τους. Ένα από αυτά τα σενάρια:



[12_π02.sb]

Μέσω της εντολής **επανάλαβε ώπου** έχουμε τη δυνατότητα να εκτελούμε επαναληπτικά κάποιες εντολές χωρίς προκαθορισμένο αριθμό επαναλήψεων και μετά το πέρας αυτής της επανάληψης να βάζουμε επιπλέον εντολές! Αυτό είναι πολύ σημαντικό γιατί μέχρι τώρα με την **για πάντα** και την **για πάντα εάν...** δε μας δινόταν αυτή η δυνατότητα.

Για να καταλάβετε καλύτερα τις διαφορές τους όμως, δοκιμάστε να αλλάξετε τις εντολές κίνησης του συλλέκτη χρησιμοποιώντας αρχικά την εντολή **για πάντα εάν...** και μετά την **επανάλαβε ώπου**. Να είστε σίγουροι ότι με αυτόν τον τρόπο θα καταλάβετε όλες τις διαφορές τους!

12.3 Εντολή "περίμενε ώπου..."

Χτυπάει το κουδούνι για το μεγάλο διάλειμμα! Η μυρωδιά του κρουασάν που ψήθηκε είναι ακαταμάχητη! Η μόνη λύση είναι να πάτε να το αγοράσετε από το κυλικείο του σχολείου σας. Με ένα μεγάλο όμως τίμημα: θα πρέπει να περιμένετε ώπου να έρθει η σειρά σας...

Η έννοια της αναμονής είναι ένα επίσης αναπόσπαστο κομμάτι της καθημερινότητας μας: η αναμονή στο super-market μέχρι να πληρώσετε, η αναμονή σε ένα τηλεφώνημα που κάνετε μέχρι ο συνομιλητής σας να σηκώσει το τηλέφωνο, αναμονή όταν περιμένετε να ξεκινήσετε να παίζετε ένα παιχνίδι μέχρι να επιλέξετε τον ήρωά σας, αναμονή για την εμφάνιση ενός κακού χαρακτήρα στο παιχνίδι που θέλει να σας εξοντώσει.

Υπάρχουν φορές στον προγραμματισμό που θέλουμε να σταματήσει η εκτέλεση ενός σεναρίου μέχρι να συμβεί κάτι. Πολλές. Για το λόγο αυτό, το Scratch μας προσφέρει την εντολή **περίμενε ώπου...** η οποία σταματά την εκτέλεση της ροής του σεναρίου μέχρι να γίνει η συνθήκη που περιέχει η εντολή αληθής. Η εντολή **περίμενε ώπου Χ** δεν περιέχει στο εσωτερικό της άλλες εντολές. Όλες οι εντολές που την ακολουθούν, θα εκτελεστούν εφόσον η συνθήκη της γίνει αληθής

Έστω για παράδειγμα ότι ο κ.Γατίδης θέλουμε να δημιουργεί ένα σχέδιο, π.χ. δυο τετράγωνα, με μια συγκεκριμένη σειρά αλλά μόνο αφού του δώσουμε το OK. Πως θα μπορούσαμε να το κάνουμε χρησιμοποιώντας την εντολή **περίμενε ώπου Χ**; Μελετήστε το παρακάτω σενάριο:





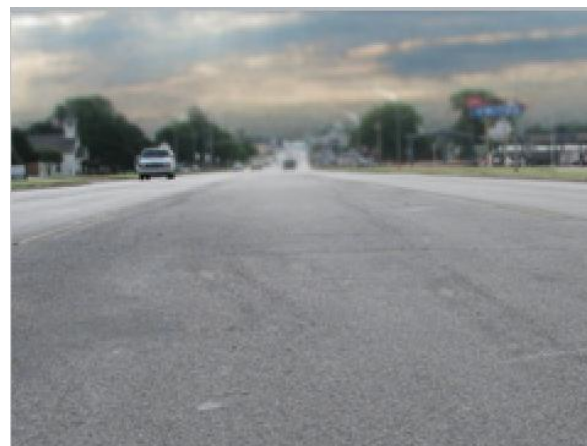
[12_π03.sb]



Όταν τρέξουμε το έργο μας, αρχικά θα εκτελεστούν οι τρεις πρώτες εντολές και το σενάριο θα σταματήσει στην πρώτη εντολή περίμενε ώσπου... Η εντολή θα επιτρέψει την εκτέλεση των επόμενων εντολών μόνο εφόσον ικανοποιηθεί η συνθήκη της, δηλαδή όταν πατηθεί το πλήκτρο 1. Αν πατήσουμε το αντίστοιχο πλήκτρο, μέσω της εντολής επανάλαβε... ο κ.Γατίδης θα σχεδιάσει ένα τετράγωνο και στη συνέχεια η εκτέλεση των εντολών θα σταματήσει πάλι στην επόμενη εντολή περίμενε ώσπου... Πρέπει να πατήσουμε 2 ώστε να σχεδιαστεί και το δεύτερο τετράγωνο. Η χρησιμότητα της περίμενε ώσπου... είναι ολοφάνερη. Δεν θα μπορούσαμε με καμία άλλη εντολή να καταφέρουμε το ίδιο αποτέλεσμα μέσα σε ένα μόνο σενάριο (ίσως μόνο με ορισμένα τρικ). Συνεπώς, με την περίμενε ώσπου... μπορούμε να δημιουργούμε σενάρια που τα διαδοχικά τους βήματα θα εκτελούνται σε διαφορετικές χρονικές στιγμές ανάλογα με τις συνθήκες που θα έχουμε ορίσει.

Θυμηθείτε ότι έχουμε χρησιμοποιήσει αρκετές φορές και την εντολή περίμενε...δευτερόλεπτα, η οποία λειτουργεί κατά αντίστοιχο τρόπο, με τη διαφορά όμως ότι απαιτεί να γνωρίζουμε εκ των προτέρων σε πόσο χρόνο θα αρχίσει να εκτελείται το επόμενο κομμάτι εντολών που την ακολουθεί. Είναι εξαιρετικά δύσκολο σε ένα παιχνίδι να γνωρίζουμε τη συμπεριφορά των αντικειμένων σε επίπεδο δευτερολέπτων από πριν και εκεί η εντολή περίμενε ώσπου... είναι ακόμη πιο καθοριστική.

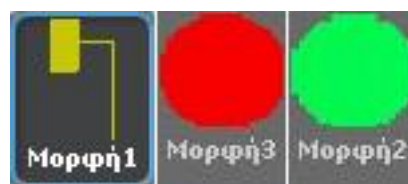
Ένα μεγαλύτερο παράδειγμα; Πώς θα σας φαινόταν η ιδέα να ζούσαμε σε έναν ιδανικό κόσμο όπου τα φανάρια θα κοκκίνιζαν για τους οδηγούς και θα πρασίνιζαν για τους πεζούς με το πάτημα ενός μόνο κουμπιού; Πάμε να το φτιάξουμε στο Scratch! Αρχικά χρειαζόμαστε το σκηνικό να είναι κάποιος δρόμος. Έστω ότι επιλέγουμε την εικόνα «route66» που βρίσκεται στο φάκελο έτοιμων υποβάθρων «outdoors»:



Μέσω της ζωγραφικής, φτιάχνουμε διάβαση για τους πεζούς και την μπροστινή πλευρά ενός αμαξίου. Το αποτέλεσμα είναι κάπως έτσι (μάλλον εσείς θα τα καταφέρετε καλύτερα ☺):



Στη συνέχεια και πάλι με τη βοήθεια του επεξεργαστή ζωγραφικής δημιουργούμε ως ξεχωριστά αντικείμενα το φανάρι, το κόκκινο και το πράσινο, όπως φαίνεται στις 3 εικόνες που ακολουθούν:



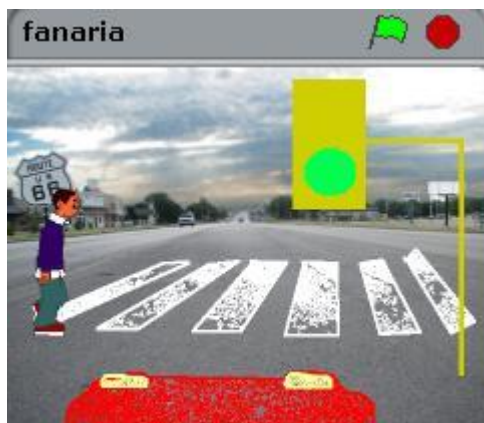
Τέλος, μένει να επιλέξουμε το χαρακτήρα που θα περάσει το δρόμο καθώς και να δημιουργήσουμε το κουμπί.

Όπως σε προηγούμενο παράδειγμα, επιλέξαμε από τη βιβλιοθήκη του Scratch και το φάκελο People, το αντικείμενο boy4-walking και τις τέσσερις σχετικές ενδυμασίες για να του δώσουμε την αίσθηση της κίνησης.



Το κουμπί το φτιάχνουμε πολύ απλά και πάλι μέσω της ζωγραφικής, φτιάχνοντας ένα μήνυμα και από κάτω ένα κουμπί.

Το αρχικό σκηνικό μας θα είναι:



Το πράσινο φανάρι για τα αυτοκίνητα είναι πάντα αναμμένο πράσινο εκτός από τη στιγμή που κάποιος πεζός πατήσει το κουμπί. Τότε αυτόματα κοκκινίζει το φανάρι και ο πεζός έχει 10 δευτερόλεπτα για να περάσει απέναντι. Σημειώνουμε ότι ο πεζός πρέπει να περιμένει μέχρι το φανάρι των αυτοκινήτων να γίνει κόκκινο και δεν μπορεί να κάνει καμία κίνηση.

Πάμε να το δούμε αναλυτικότερα:

Κόκκινο και Πράσινο: Το πράσινο είναι αναμμένο και σταματάει για 10 δευτερόλεπτα αν ο πεζός πατήσει το κουμπί και το κόκκινο το αντίθετο. Τα σενάρια τους παρουσιάζονται στις παρακάτω εικόνες:



Ο πεζός: Ο πεζός θα προχωράει με την εναλλαγή ενδυμασιών κάθε φορά που θα πατάμε το δεξί βελάκι, αλλά πρώτα θα πρέπει να περιμένει να πατηθεί το ποντίκι, γιατί αλλιώς δεν μπορεί να περάσει.

Πώς όμως θα το πετύχουμε αυτό; Με την εντολή **περίμενε ώσπου...** μπορούμε να αναγκάσουμε το αντικείμενο να περιμένει, χωρίς να εκτελεί καμία εντολή μέχρι η συνθήκη να γίνει αληθής. Στο συγκεκριμένο παράδειγμα, η συνθήκη είναι αν πατήθηκε το ποντίκι οπότε το σενάριο θα είναι το παρακάτω:



[12_π04.sb]

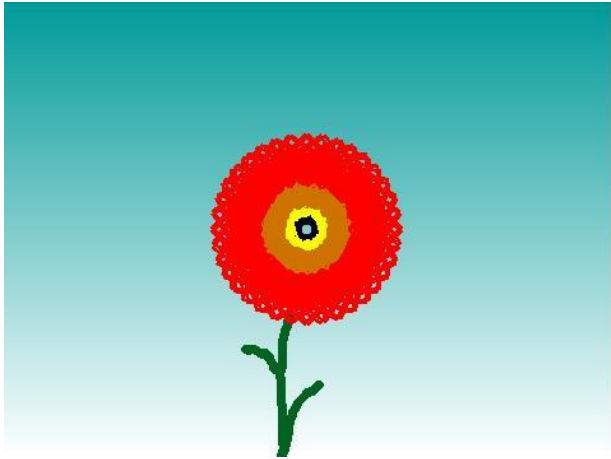
Σύμφωνα με το παραπάνω σενάριο, θα πρέπει να πατήσουμε πρώτα το ποντίκι ώστε να αλλάξει το χρώμα του φαναριού και στη συνέχεια, θα μπορέσουμε με το δεξί βέλος να κινήσουμε προς τα δεξιά τον πρωταγωνιστή μας. Στο σενάριο έχουμε ενσωματώσει μια σειρά από λαθάκια τα οποία μπορούν να αποφευχθούν. Προσπαθήστε να απαντήσετε στα επόμενα ερωτήματα:

- Ποιος είναι ο ελάχιστος αριθμός αντικειμένων με τα οποία θα μπορούσαμε να είχαμε το ίδιο αποτέλεσμα;
- Πως θα μπορούσε να γίνει πιο «αποδοτικό» το σενάριο των δυο φαναριών;
- Πως θα μπορούσε να γίνει πιο «αποδοτικό» το σενάριο του χαρακτήρα που περνά από το δρόμο;
- Υπάρχει δυνατότητα να ξαναγραφτεί το σενάριο του χαρακτήρα ώστε αφού περάσει το δρόμο, να μπορεί να γίνει και η αντίστροφη κίνηση με το πάτημα του ποντικιού;
- Πως θα μπορούσαμε να ελέγξουμε το κατά πόσο ο χαρακτήρας μας πρόλαβε να περάσει απέναντι όσο το φανάρι ήταν κόκκινο για τα αυτοκίνητα;

12.4 Συνδυασμός εντολών επανάληψης

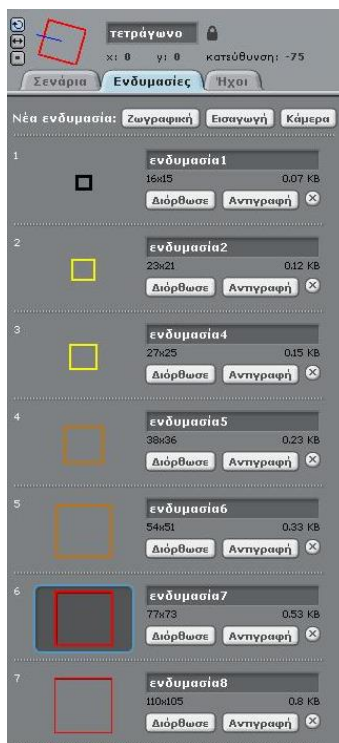
Στην ενότητα αυτή θα προσπαθήσουμε να κατανοήσουμε καλύτερα τις διαφορετικές εντολές επανάληψης που έχουμε παρουσιάσει μέσα από διαφορετικά παραδείγματα.

Παράδειγμα «Ένα λουλουδί»: Αυτό το παράδειγμα έχει ως στόχο τη δημιουργία του άνθους ενός λουλουδιού σε ένα μπλε φόντο όπου υπάρχει ήδη ο μίσχος του, ενώ παράλληλα ακούγεται το τραγούδι Suspense. Ο μίσχος θέλουμε να είναι τοποθετημένος στο κέντρο του παραθύρου ενώ το άνθος θα δημιουργηθεί στο πάνω άκρο του όπως φαίνεται στην επόμενη εικόνα:



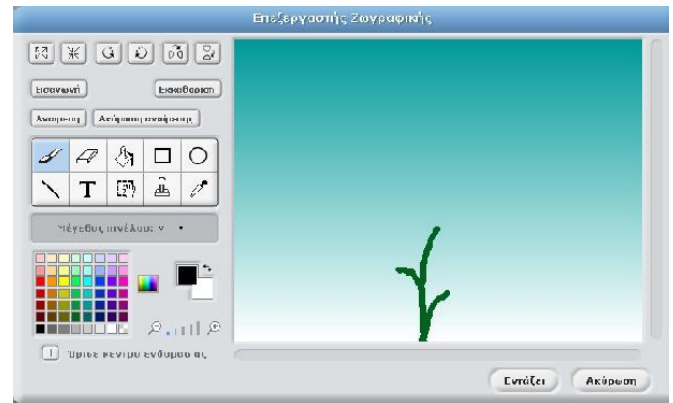
Όπως είδαμε και σε προηγούμενα κεφάλαια, η δημιουργία τέτοιων πολύπλοκων σχημάτων βασίζεται συνήθως σε σχήματα που αφήνουν επαναλαμβανόμενα τη σφραγίδα τους σε διαφορετικά σημεία ή με διαφορετική κατεύθυνση. Κάτι αντίστοιχο θα κάνουμε και στο συγκεκριμένο παράδειγμα.

Η δημιουργία του άνθους θα βασιστεί σε ένα αντικείμενο με εφτά διαφορετικές ενδυμασίες. Όλες οι ενδυμασίες θα είναι περιγράμματα τετραγώνων διαφορετικού μεγέθους και διαφορετικών χρωμάτων. Πιο συγκεκριμένα χρειαζόμαστε τα περιγράμματα ενός μαύρου τετραγώνου 16x15, δύο κίτρινων τετραγώνων 23x21 και 27x25, δύο πορτοκαλί τετραγώνων 38x36 και 54x51 και δύο κόκκινων διαστάσεων 77x73 και 110x105. Οι διαφορετικές ενδυμασίες φαίνονται στην επόμενη εικόνα.



Η δημιουργία του άνθους θα ξεκινάει από το μαύρο τετράγωνο, την αρχική δηλαδή ενδυμασία, το οποίο θα αφήνει μια σφραγίδα ενώ στη συνέχεια θα επιλέγεται η επόμενη ενδυμασία, το αντικείμενό μας θα στρίβει δεξιά κατά 10 μοίρες και θα αφήνει το αποτύπωμα της νέας ενδυμασίας. Η διαδικασία αυτή θέλουμε να επαναλαμβάνεται έως ότου σχηματιστεί το άνθος που φαίνεται στην προηγούμενη εικόνα.

Δημιουργούμε το φόντο χρησιμοποιώντας τον επεξεργαστή ζωγραφικής.



Για να ξεκινάει η δημιουργία του άνθους από το μαύρο τετράγωνο θα χρησιμοποιήσουμε την εντολή αλλαγή σε ενδυμασία (ενδυμασία1). Για να δοκιμάσουμε την προσοχή σας στις προηγούμενες εικόνες, για την επανάληψη των βημάτων επιλέγουμε την εντολή επανέλαβε ώσπου με συνθήκη «ενδυμασία # = 8» (τι σημαίνει αυτό;). Στο σώμα της επανάληψης θα περιλάβουμε τη δημιουργία αποτυπώματος με την εντολή σφραγίδα, την περιστροφή του αντικείμενου κατά 10 μοίρες και μια μικρή αναμονή πριν προχωρήσει το έργο μας στην επόμενη σφραγίδα για να προλάβουμε να παρατηρήσουμε πως σχηματίζεται το άνθος. Το σενάριο παρουσιάζεται παρακάτω:



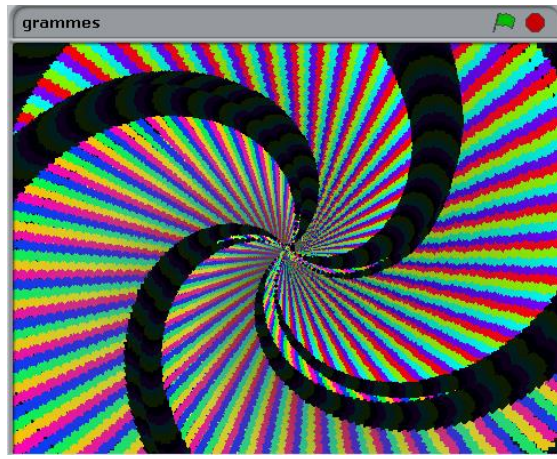
[12_π05.sb]



Η επανάληψη σε συνδυασμό με την εντολή επόμενη ενδυμασία επιτυγχάνει το στόχο μας. Ή μήπως όχι;

Παρατηρήστε ότι παρότι το όνομα της τελευταίας ενδυμασίας είναι το «ενδυμασία 8», οι ενδυμασίες είναι 7 και η τελευταία ενδυμασία προσδιορίζεται με τον ίδιο αριθμό (λείπει η ενδυμασία 3!). Αρα η συνθήκη δε θα γίνει ποτέ αληθής και η επανάληψη θα συνεχίσει μέχρι να τερματίσει ο χρήστης το έργο. Αρα η συγκεκριμένη μορφή της εντολής επανέλαβε ώσπου... λειτουργεί όπως μια εντολή για πάντα! Τι θα συνέβαινε αν στη συνθήκη αλλάζαμε το 8 σε 7; Θα τυπωνόταν το άνθος; Ποια θεωρείτε ότι είναι η καταλληλότερη εντολή για το συγκεκριμένο στόχο (για πάντα, επανέλαβε ώσπου..., επανάλαβε X). Αν απαντήσατε η Τρίτη, μαντέψατε σωστά. Μπορείτε να μας εξηγήσετε και το γιατί;

Μελετήστε από μόνοι σας ένα ανάλογο παράδειγμα που έχει ως αποτέλεσμα το σχήμα της επόμενης εικόνας:



[12_π06.sb]

Παράδειγμα «Το ελικόπτερο πέφτει...»: Έχουμε εξετάσει σε προηγούμενο κεφάλαιο αντίστοιχο παράδειγμα. Ζητούμενό μας είναι ο χρήστης να μπορεί να ελέγχει ένα ελικόπτερο που πρέπει να απογειώνεται από μια συγκεκριμένη θέση στην τάρταρα ενός κίτρινου ουρανοξύστη με στόχο να το προσγειώσει σε ένα μπλε ελικοδρόμιο, οπότε και τερματίζει το παιχνίδι.

Πιο συγκεκριμένα, το ελικόπτερο θα πρέπει να κινείται προς τα πάνω όταν πατάμε το πάνω βελάκι του πληκτρολογίου, ενώ αν δεν το πατάμε (και το ελικόπτερο δεν βρίσκεται πάνω στον κίτρινο ουρανοξύστη) θα πρέπει να χάνει ύψος. Όταν πατάμε το δεξί βελάκι θα μετακινείται προς τα δεξιά, ενώ διόρθωση με το αριστερό βελάκι δεν θα υπάρχει! Όσο κινείται στον αέρα θα πρέπει να ακούγεται αντίστοιχος ήχος για τον έλικα, ενώ όταν ακουμπήσει την μπλε πλατφόρμα θα πρέπει να εμφανίζει η εκτέλεση του έργου μας. Αν ακουμπήσει σύννεφο, τότε θα πρέπει να γυρίζει στο σημείο εκκίνησης εμφανίζοντας το μήνυμα «Προσπάθησε ξανά!» για 1 δευτερόλεπτο.

Για να δημιουργήσουμε το σκηνικό θα χρησιμοποιήσουμε τον επεξεργαστή ζωγραφικής. Χρειαζόμαστε ένα γκρι ανοιχτό φόντο, στην κάτω αριστερή γωνία ένα κίτρινο ουρανοξύστη, στην κάτω δεξιά ένα μπλε ελικοδρόμιο και τρία σύννεφα χρώματος γκρι σκούρο διάσπαρτα στον ουρανό. Το ολοκληρωμένο σκηνικό μαζί με το ελικόπτερο φαίνονται στην επόμενη εικόνα.



Για να δημιουργήσουμε το ελικόπτερο, επιλέγουμε τη μορφή helicopter1 από τον κατάλογο transportation της βιβλιοθήκης αντικειμένων του Scratch. Η συμπεριφορά του ελικοπτερού εί-

ναι σχετικά απλή. Για την ανοδική κίνηση θα πρέπει να αυξάνουμε τη συντεταγμένη y της θέσης του, ενώ για την κίνηση προς τα δεξιά θα πρέπει να αυξάνουμε την συντεταγμένη x της θέσης του. Για να πέφτει το ελικόπτερο θα μειώνουμε την συντεταγμένη y της θέσης του. Για να ακούγεται ο ήχος του έλικα, εισάγουμε στους ήχους τους αντικείμενου τον ήχο Peculiar από το φάκελο Electronic του Scratch.

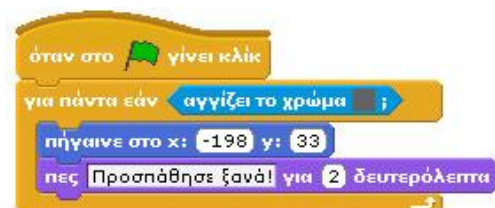
Θα βάλουμε το ελικόπτερο αρχικά πάνω στην κίτρινη πολυκατοικία και στη συνέχεια με το συνδυασμό των εντολών για πάντα και εάν... θα ελέγχουμε για όλες τις παραπάνω συνθήκες κίνησης.

Οι συνθήκες που πρέπει να εξεταστούν είναι:

- A) *αν αγγίζει το κίτρινο χρώμα*, δηλαδή αν βρίσκεται πάνω στην πολυκατοικία, όπου δεν πρέπει να ακούγεται ο ήχος του έλικα,
- B) *αν δεν αγγίζει το κίτρινο χρώμα*, δηλαδή αν βρίσκεται στον αέρα, οπότε πρέπει να χάνει λίγο ύψος και να ακούγεται ο ήχος του έλικα,
- Γ) *αν πατήθηκε το πάνω βέλος*, οπότε πρέπει να αυξάνουμε το ύψος του,
- Δ) *αν πατήθηκε το δεξί βέλος*, οπότε πρέπει να μετακινείται προς τα δεξιά,
- Ε) *αν αγγίζει το μπλε χρώμα*, οπότε πρέπει να βγει το αντίστοιχο μήνυμα τέλους και να ολοκληρωθεί το έργο μας.



Επιπλέον, για την περίπτωση που το ελικοπτεράκι πέσει πάνω στα σύννεφα, θα χρησιμοποιήσουμε την εντολή για πάντα εάν ... με συνθήκη το αν αγγίζει το χρώμα γκρι.



[12_π07.sb]

Φαντάζεστε κάποιο για ποιο λόγο εισάγαμε το συγκεκριμένο σενάριο ξεχωριστά από τους υπόλοιπους ελέγχους; Θα μπορούσαμε να είχαμε εισάγει τις αντίστοιχες εντολές στο πρώτο σενάριο χωρίς καμία αλλαγή; Αν όχι, με ποιες αλλαγές θα μπορούσαν να εισαχθούν οι συγκεκριμένες εντολές στο πρώτο σενάριο;

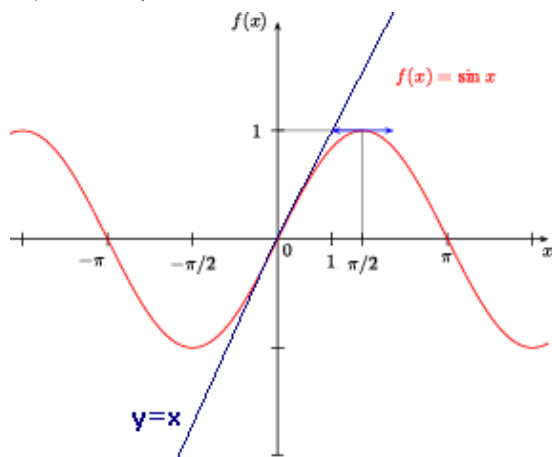
Θα μπορούσαμε να έχουμε μόνο μια εάν...αλλιώς στο πρώτο σενάριο για να ελέγξουμε την κίνηση στον κάθετο άξονα (όταν είναι στον αέρα και όταν πατιέται το πάνω βελάκι); Επίσης, ο έλεγχος της κίνησης θα μπορούσε να γίνει με τη χρήση της εντολής όταν πατηθεί το πλήκτρο...;

Όπως μπορείτε να παρατηρήσετε, οι εναλλακτικοί δρόμοι που μας παρέχει το Scratch για να λύσουμε τα προβλήματά μας είναι πολλοί και κάποιες φορές έντονα διαφορετικοί. Δεν υπάρχει μια λύση, δεν υπάρχει μια μόνο εντολή επανάληψης που μπορεί χρησιμοποιηθεί.

12.5 Τελεστές, συνθήκες και άλλα!

Όπως έχει διαφανεί από όλα τα προηγούμενα παραδείγματα, η κατασκευή κατάλληλων συνθηκών στις εντολές εάν..., εάν...αλλιώς, για πάντα εάν..., περίμενε ώσπου..., επανέλαβε ώσπου..., είναι εξαιρετικά σημαντική. Συζητήσαμε στο προηγούμενο κεφάλαιο για τους τελεστές συσχέτισης συνθηκών (...ή..., ...και..., όχι...) αλλά βλέπουμε στα παραδείγματά μας ότι υπάρχουν πολλά διαφορετικά στοιχεία και επιπλέον τελεστές που μπορούμε να αξιοποιήσουμε στη διαμόρφωση συνθηκών και όχι μόνο.

Παράδειγμα «Ημίτονο και ζωγραφική!»: Έχει δει στα μαθηματικά τη γραφική παράσταση της συνάρτησης του ημιτόνου; Σας θυμίζει κάτι η παρακάτω εικόνα;



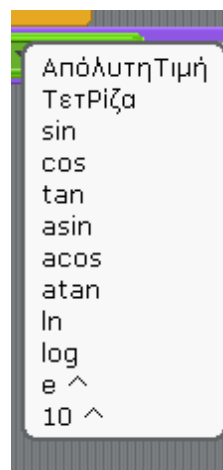
Θα προσπαθήσουμε να την αναπαράγουμε στο Scratch. Ας ξεκινήσουμε όμως από τα βασικά. Το Scratch εμπεριέχει τον αυτόματο υπολογισμό πληθώρας μαθηματικών συναρτήσεων. Η αντίστοιχη εντολή που μας επιστρέφει το αποτέλεσμα των μαθηματικών συναρτήσεων βρίσκεται στην παλέτα **Τελεστές** και έχει ως προεπιλεγμένη συνάρτηση την τετραγωνική ρίζα.

Όλες οι μαθηματικές συναρτήσεις απαιτούν μια τιμή εισόδου (το λευκό κουτάκι στην εντολή)

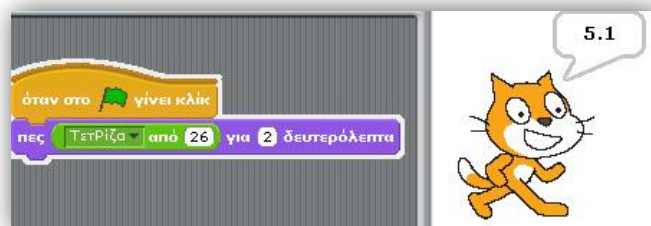
και παράγουν μια τιμή εξόδου. Οι διαθέσιμες συναρτήσεις φαίνονται στην επόμενη εικόνα (απόλυτη τιμή, τετραγωνική ρίζα,

ΤετΡΙζα από 10

ημίτονο, συνημίτονο, εφαπτομένη, αντίστροφο ημίτονο, αντίστροφο συνημίτονο, αντίστροφη εφαπτομένη, λογάριθμος, δεκαδικός λογάριθμος, εκθετική συνάρτηση και δύναμη του 10)



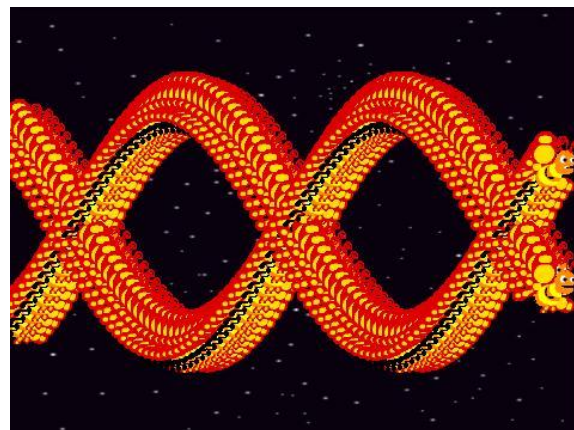
Το αποτέλεσμα της μαθηματικής συνάρτησης μπορεί να χρησιμοποιηθεί είτε ως είσοδος σε οποιαδήποτε εντολή περιέχει λευκό κουτάκι είτε για κάποιον έλεγχο σε μια συνθήκη. Μπορείτε συνεπώς να βάλετε για παράδειγμα το χαρακτήρα σας να σας λέει την τετραγωνική ρίζα του 26:



ή να κινηθεί κατά εκατό βήματα δηλώνοντάς το με έναν άλλο τρόπο.



Ας επιστρέψουμε όμως στο παράδειγμα του ημιτόνου. Θέλουμε δυο πανομοιότυπα αντικείμενα να δημιουργήσουν το σχήμα της παρακάτω εικόνας.

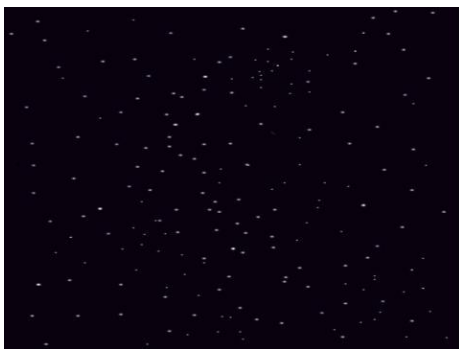


Θυμηθείτε ότι η θέση κάθε αντικειμένου στο Scratch προσδιορίζεται βάση του καρτεσιανού συστήματος συντεταγμένων και η διάσταση στον άξονα X παίρνει νούμερα από -240 έως 240. Τι θα συνέβαινε αν ζητούσαμε επαναλαμβανόμενα να υπολογί-

στεί το ημίτονο του $-235, -230, -230, \dots, 0, 5, 10, 15, \dots, 210$; Θα παράγονταν οι τιμές του ημιτόνου που θα θέλαμε να αναπαράστησουμε στον άξονα Y. Το ημίτονο παίρνει τιμές από -1 έως 1 . Αν όμως πολλαπλασιάζαμε με το 100 αυτές τις τιμές, δε θα δημιουργούνταν συντεταγμένες Y και για την οθόνη του Scratch; Βάσει αυτής ακριβώς της λογικής θα δημιουργήσουμε την προηγούμενη εικόνα. Αρχικά, θα εισάγουμε 2 αντικείμενα με την ίδια μορφή από τη βιβλιοθήκη αντικειμένων του Scratch (butterfly2 από τον φάκελο Animals), και θα τα τοποθετήσουμε στις αντισυμμετρικές θέσεις (235,0) και (-235,0) αντίστοιχα.



Ο στόχος μας είναι να ρυθμίσουμε την κίνησή τους έτσι ώστε, το ένα αντικείμενο να σχεδιάζει τη συνάρτηση του ημιτόνου της θέσης του x, ενώ το άλλο να σχεδιάζει την αντισυμμετρική της προηγούμενης γραφική παράσταση. Όσον αφορά το σκηνικό, επιλέγουμε το stars από τον φάκελο Nature των υποβόθρων του Scratch.



Ο σχεδιασμός των γραφημάτων θέλουμε να εκτελείται για τα δύο αντικείμενα, όσο η τιμή του x είναι μικρότερη του 210, μέχρι δηλαδή το δεξί άκρο της οθόνης. Αφού η συμπεριφορά του αντικειμένου θα εκτελείται επαναληπτικά όσο η συντεταγμένη X των αντικειμένων είναι μικρότερη του 210, θα χρησιμοποιήσουμε τη δομή επανάληψης **για πάντα εάν X** με την αντίστοιχη συνθήκη. Στην παλέτα Τελεστές, το Scratch μας προσφέρει όλους τους κλασσικούς αριθμητικούς τελεστές σύγκρισης:



Το σχήμα τους είναι εξαγωνικό, γεγονός που σημαίνει ότι οι τελεστές αυτοί μπορούν να χρησιμοποιηθούν ως συνθήκες. Επιλέγουμε τον κατάλληλο τελεστή και στη συνέχεια μεταφέρουμε μέσα στα λευκά κουτάκια τα στοιχεία της σύγκρισης.

Τέλος, για την αναπαράσταση της συνάρτησης, θα αφήνουμε τη σφραγίδα των αντικειμένων σε κάθε σημείο υπολογισμού του ημιτόνου. Όπως προαναφέρθηκε θα υπολογίζουμε τη νέα θέση Y του αντικειμένου πολλαπλασιάζοντας το αποτέλεσμα της συνάρτησης του ημιτόνου του x με το 100. Πως μπορούμε να κάνουμε τον πολλαπλασιασμό; Το Scratch παρέχει τη δυνατότητα υπολογισμού όλων των τυπικών μαθηματικών πράξεων:



Παρατηρήστε ότι το σχήμα των εντολών αυτών είναι στρογγυλοποιημένο στα άκρα τους γεγονός που σημαίνει ότι μπορούν να χρησιμοποιούν σε οποιοδήποτε λευκό κουτάκι αλλά όχι ως συνθήκες, όπως είναι αυτονόητο.

Μελετήστε και τρέξτε τα σενάρια των δυο αντικειμένων:



[12_π08.sb]



Το αποτέλεσμα είναι πραγματικά εντυπωσιακό. Θα μπορούσαμε να χρησιμοποιήσουμε την εντολή **επανάλαβε ώσπου** για την υλοποίηση της συγκεκριμένης επανάληψης; Ποια θα ήταν η κατάλληλη συνθήκη;

Συνοψίζοντας τα συμπεράσματα του παραδείγματος:

- Α) Όλες οι μεταβλητές κατάστασης που έχουμε δει μέχρι αυτό το κεφάλαιο μπορούν να χρησιμοποιηθούν είτε ως είσοδοι στα λευκά κουτάκια των εντολών είτε ως στοιχεία σύγκρισης στις συνθήκες.
- Β) στα λευκά κουτάκια μπορούμε να εισάγουμε υπολογιζόμενες τιμές από σύνθετες αριθμητικές αναπαραστάσεις.
- Γ) οι μαθηματικές συναρτήσεις μας επιτρέπουν να φέρουμε άλλα μαθήματα μέσα στο Scratch! Φανταστείτε και προτείνεται προβλήματα που μπορούν να σας ζητήσουν να υλοποιήσετε στο Scratch οι καθηγητές των Μαθηματικών και της Φυσικής!

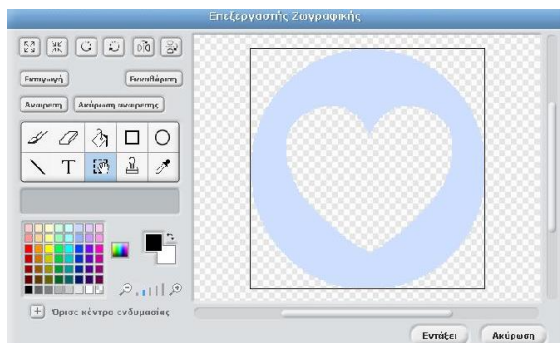
Παράδειγμα «Κρυμμένο σχήμα»: Στο παράδειγμα που ακολουθεί θα προσκαλέσουμε τους χρήστες μας να αποκαλύψουν ένα κρυμμένο σχήμα πατώντας επαναλαμβανόμενα κλικ με το ποντίκι τους σε ένα φαινομενικά κενό σκηνικό. Τα κλικ μας θα δημιουργούν το αποτύπωμα μιας κουκίδας που ακολουθεί το δείκτη του ποντικιού μόνο όμως όταν η κουκίδα δεν βρίσκεται πάνω από το κρυμμένο αντικείμενο.

Το έργο μας θα περιλαμβάνει δύο αντικείμενα. Το πρώτο θα είναι το κρυμμένο σχήμα το οποίο θα έχει το ίδιο χρώμα με το σκηνικό του έργου μας ώστε να μη φαίνεται. Το δεύτερο θα είναι μία κουκίδα η οποία θα μετακινείται μαζί με το δείκτη του ποντικιού και θα αφήνει το αποτύπωμά της με διαφορετικό

χρώμα κάθε φορά που θα πατάμε στο ποντίκι μας (εφόσον δεν βρισκόμαστε πάνω από το κρυμμένο αντικείμενο). Για να είναι λίγο δυσκολότερο το έργο μας, η κουκίδα δεν θα μετακινείται ακριβώς κάτω από το δείκτη του ποντικιού, αλλά ανάλογα με το πόσο μακριά βρίσκεται ο δείκτης του ποντικιού στον άξονα Χ από το κέντρο της οθόνης του Scratch, η κουκίδα θα απομακρύνεται αναλόγως από το δείκτη του ποντικιού! Τέλος, θέλουμε όταν ο χρήστης πατά το πλήκτρο c να καθαρίζει η οθόνη ώστε να μπορεί να ξεκινά ξανά την προσπάθειά του.



Το κρυφό μας σχήμα:



Στην επόμενη εικόνα βλέπετε την οθόνη του έργου μας μετά από πολλά κλικ...



Ποιες θα είναι οι συμπεριφορές των αντικειμένων μας;

Κουκίδα: Καταρχάς, η κουκίδα θα πρέπει να μετακινείται μαζί με το δείκτη του ποντικιού. Η συντεταγμένη Y θα είναι ίδια με τη θέση Y του δείκτη του ποντικιού. Για να καταφέρουμε να μετακινούμε την κουκίδα στον άξονα Χ ανάλογα με την απόστασή της από το κέντρο θα πρέπει να δημιουργήσουμε μια μαθηματική έκφραση. Για παράδειγμα,

Νέα Θέση = Θέση Χ του δείκτη του ποντικιού + α * Θέση Χ αντικειμένου

Η θέση Χ της κουκίδας αποτελεί την απόστασή της από το κέντρο. Άρα, αν στην τρέχουσα θέση του δείκτη του ποντικιού προσθέσουμε μια «ποσότητα» ανάλογη της θέσης Χ του αντικειμένου, τότε η κουκίδα θα απομακρύνεται από το δείκτη του

ποντικιού όσο ο δείκτης απομακρύνεται από το κέντρο της οθόνης του Scratch. Για να δημιουργήσουμε την παραπάνω έκφραση, χρειαζόμαστε μια πρόσθεση και έναν πολλαπλασιασμό,

και συνεπώς θα χρησιμοποιήσουμε τους τελεστές  εισάγοντας το δεύτερο μέσα στον πρώτο.

Η κουκίδα θα πρέπει να αφήνει το αποτύπωμά της μόνο εφόσον δεν ακουμπά το κρυμμένο αντικείμενο και μόνο όταν ταυτόχρονα πατιέται το πλήκτρο του ποντικιού. Αυτή την απαίτηση μπορούμε να την ικανοποιήσουμε με την εντολή για **πάντα εαν...** και τον τελεστή **...ΚΑΙ...** ο οποίος απαιτεί και οι δυο συνθήκες να είναι αληθείς ώστε να αποτιμηθεί ως αληθής και η έκφραση. Μελετήστε τα παρακάτω 2 σενάρια που αφορούν την κουκίδα



Η τελευταία απαίτηση του προβλήματος είναι να καθαρίζουμε την οθόνη όποτε επιθυμούμε, με το πάτημα του πλήκτρου c. Για το σκοπό αυτό αρκεί το σενάριο της επόμενης εικόνας.



[12_π09.sb]



Είναι σημαντικό να διακρίνουμε ότι έχουμε δει τρία διαφορετικά είδη συνθηκών:

Α) συνθήκες που αφορούν την αλληλεπίδραση ενός χαρακτήρα με το χρήστη (π.χ. αν πατήθηκε το ποντίκι ή κάποιο άλλο πλήκτρο),

Β) συνθήκες που αφορούν τη σχέση του αντικειμένου με άλλα αντικείμενα και το σκηνικό (π.χ. αν ακουμπά άλλο αντικείμενο, ή αν αγγίζει άλλο χρώμα),

Γ) συνθήκες που αφορούν τη συσχέτιση μιας ιδιότητας του αντικειμένου (π.χ. θέση, ενδυμασία κτλ.) με μια μαθηματική έκφραση.

Η ποικιλία αυτή συνθηκών μας δίνει μια εξαιρετική ευελιξία για να δημιουργήσουμε ενδιαφέροντα και απρόβλεπτα παιχνίδια στη συνέχεια.

Περίληψη

Ανακεφαλαιώνοντας, στην παρούσα ενότητα ασχοληθήκαμε και πάλι με την έννοια της επανάληψης που γνωρίσαμε αρχικά στο κεφάλαιο 10, ενώ εμβαθύνσαμε περισσότερο στις έννοιες των τελεστών και των αισθητήρων, που γνωρίσαμε στο κεφάλαιο 11. Συζητήσαμε για τρεις νέες εντολές επανάληψης, την **για πάντα εάν Χ**, με την οποία ένα σώμα εντολών εκτελείται για πά-

ντα στο έργο μας όσο ικανοποιείται μια συνθήκη, την επανάλαβε ώσπου, στην οποία οι περιεχόμενες εντολές εκτελούνται όσο η συνθήκη είναι ψευδής και την περίμενε ώσπου X, η οποία διακόπτει την εκτέλεση ενός σεναρίου μέχρι να ικανοποιηθεί μια συνθήκη. Συζητήσαμε και είδαμε ότι πολλά προγραμματιστικά προβλήματα μπορούν να λυθούν με διαφορετικούς συνδυασμούς των συγκεκριμένων εντολών. Είστε έτοιμοι για να μπείτε στα πιο βαθιές; Πως επικοινωνούν τότε τα αντικείμενα μεταξύ τους;

Ερωτήσεις

1) Διακρίνεται αν οι παρακάτω δηλώσεις είναι σωστές ή λανθασμένες:

Σ-Λ : Μετά την ολοκλήρωση της δομής για πάντα εάν..., μπορούμε να προσθέσουμε κι άλλες εντολές.

Σ-Λ : Η εντολή επανάλαβε ώσπου... έχει τον ίδιο αποτέλεσμα με την επανάλαβε...

Σ-Λ : Ο αισθητήρας αγγίζει το X μπορεί να παίρνει ως επιλογή άλλα αντικείμενα που υπάρχουν στο έργο μας.

Σ-Λ : Μπορούμε να συγκρίνουμε τις συντεταγμένες X δύο θέσεων μέσω των λογικών τελεστών ...και..., ...ή..., όχι...

Σ-Λ : Κάθε φορά που γίνεται αληθής η συνθήκη της εντολής περίμενε ώσπου... εκτελείται το σώμα εντολών που την ακολουθεί.

2) Συμπληρώστε τα κενά

Ο τελεστής γίνεται αληθής μόνο αν και οι δύο συνθήκες του είναι αληθείς, ενώ για τον τελεστή αρκεί μία από τις δύο συνθήκες να αληθεύει.

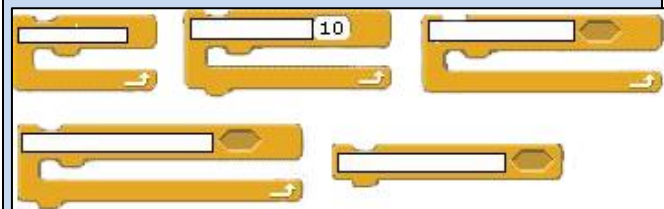
Η δομή..... και η δομή τερματίζουν όταν η συνθήκη τους γίνει αληθής.

3) Εντοπίστε τις κρυμμένες λέξεις :

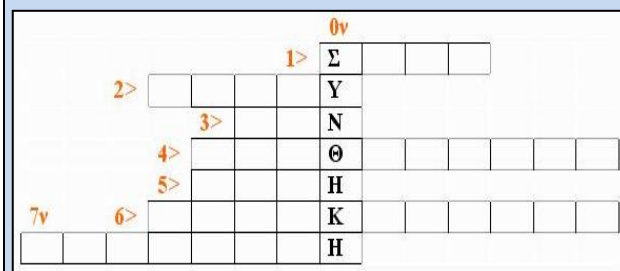
περίμενε ώσπου, επανάλαβε ώσπου, για πάντα εάν, αισθητήρας, τελεστής, συνθήκη.

Ε	Τ	Γ	Β	Υ	Ξ	Θ	Ο	Χ	Ζ	Α	Δ	Ψ	Σ
Α	Π	Ε	Ρ	Ι	Μ	Ε	Ν	Ε	Ω	Σ	Π	Ο	Υ
Σ	Γ	Α	Ρ	Ξ	Σ	Ν	Κ	Σ	Δ	Β	Π	Ψ	Ν
Δ	Χ	Ι	Ν	Α	Ν	Τ	Σ	Κ	Ω	Σ	Λ	Κ	Θ
Φ	Ζ	Σ	Β	Ε	Ε	Γ	Ν	Φ	Κ	Α	Ρ	Π	Η
Γ	Ε	Θ	Ε	Ρ	Λ	Ι	Τ	Τ	Υ	Ι	Α	Β	Κ
Η	Ζ	Η	Ε	Φ	Γ	Α	Η	Φ	Ν	Θ	Ν	Χ	Η
Ξ	Δ	Τ	Γ	Η	Ξ	Π	Β	Ψ	Β	Ζ	Μ	Ζ	Ι
Κ	Τ	Η	Ξ	Κ	Χ	Α	Α	Ε	Β	Κ	Β	Ε	Θ
Λ	Γ	Ρ	Κ	Σ	Ο	Ν	Ε	Ψ	Ω	Α	Β	Ε	Ρ
Σ	Λ	Α	Ζ	Δ	Ρ	Τ	Ε	Λ	Ε	Σ	Τ	Η	Σ
Ε	Ω	Σ	Η	Μ	Δ	Α	Ε	Υ	Μ	Δ	Π	Σ	Μ
Ρ	Π	Η	Μ	Σ	Α	Ε	Λ	Χ	Π	Γ	Ε	Ο	Κ
Τ	Α	Δ	Φ	Γ	Β	Α	Σ	Π	Φ	Ξ	Α	Α	Υ
Υ	Ζ	Χ	Ψ	Ω	Β	Ν	Ν	Μ	Ι	Π	Ξ	Β	Δ

4) Αναγνωρίστε τις δομές επανάληψης και συμπληρώστε τα κενά.



5) Συμπληρώστε το σταυρόλεξο.



.....εντολών: τοποθετείται στο εσωτερικό μίας δομής επανάληψης.

Επανάλαβε και Περίμενε

Για πάντα...

Είναι ο «αγγίζει το...»

Μπορεί να είναι ελέγχου, επανάληψης

Εισάγονται μπροστά από το σκηνικό.

Υλοποιείται μέσω των τελεστών <, > και =

6) Επιλέξτε τη σωστή απάντηση και βρείτε την κρυμμένη λέξη.

-Αληθεύει, αν και οι δύο συνθήκες του αληθεύουν

Ε) και Η) ή Θ) όχι

-Ποια από τις παρακάτω δομές δεν εκτελεί σώμα εντολών

Γ) Για πάντα-εάν Μ) Επανάλαβε-ώσπου Λ) Περίμενε-ώσπου

-Μπορεί να κάνει άπειρες επαναλήψεις.

Ε) Περίμενε-ώσπου Β) εάν Σ) Επανάλαβε τόσες φορές

-Πρέπει το αντικείμενο να κινείται μέχρι να φτάσει στα όρια.



-Ποια συνθήκη λείπει;

Ρ) αγγίζει το όριο ;

Γ) όχι αγγίζει το όριο ;

Χ) πιατήθηκε το ποντίκι;

-Ποιο είναι ένα από τα κοινά των δομών επανάλαβε ώσπου και περίμενε ώσπου X;

Χ) Σταματούν να εκτελούνται όταν η συνθήκη γίνει αληθής

Ν) Σταματούν να εκτελούνται όταν η συνθήκη γίνει ψευδής

Λ) Δεν ελέγχουν κάποια συνθήκη

-Μέσα στη συνθήκη των τελεστών δεν μπορούμε να βάλουμε:

Θ) Άλλους τελεστές Κ) Αισθητήρες Ο) Άλλες εντολές

-Ποιόν τελεστή θα χρησιμοποιήσουμε για να βρούμε την εφάπτομένη των 30 μοιρών

Ω) cos από 30 Σ) tan από 30 Ο) atan από 30

ΚΡΥΜΜΕΝΗ ΛΕΞΗ : _____

7) Περιγράψτε τί κάνουν τα παρακάτω κομμάτια σεναρίων. Σε ποιο από τα δυο υπάρχει πρόβλημα;

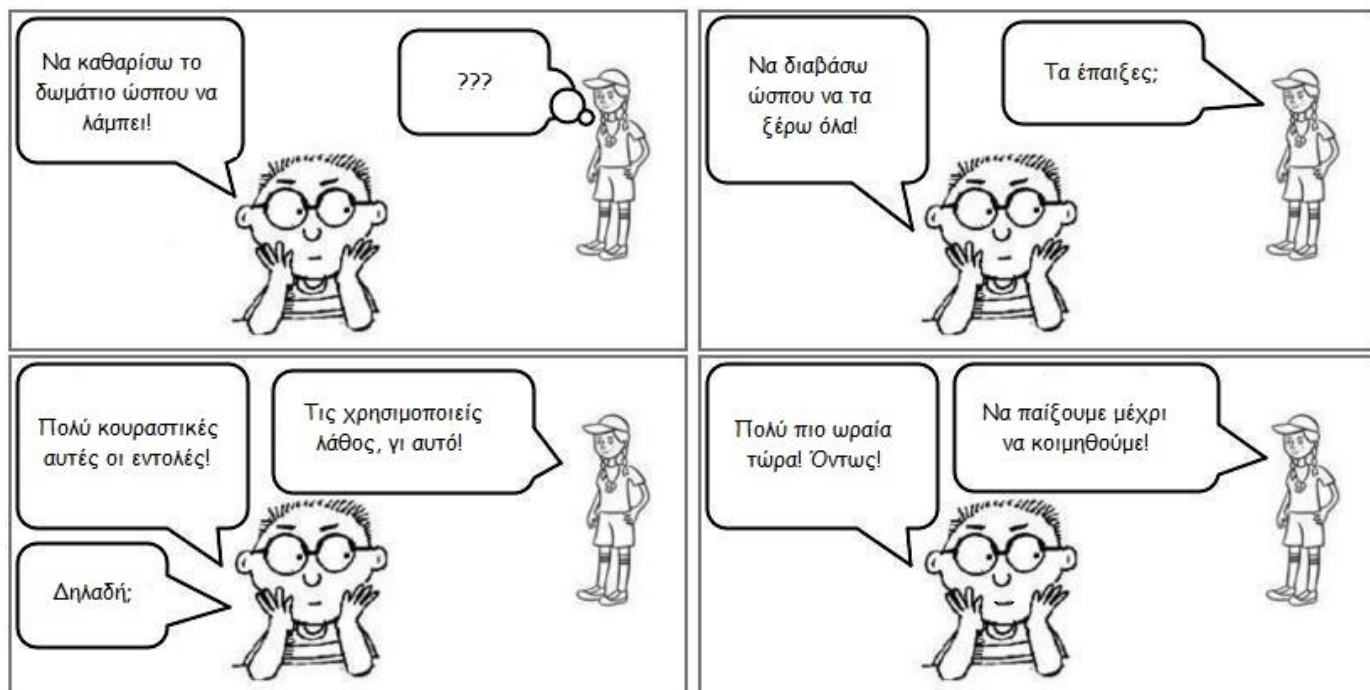


Δραστηριότητες

1) Παιχνίδι με το όνομά σας: Το αποτέλεσμα της εφαρμογής αυτής θα πρέπει να είναι η εμφάνιση στην οθόνη αντικειμένων ισοσάριθμων με τα γράμματα του ονόματός σας. Όταν πατήσουμε σε κάθε ένα από τα αντικείμενα, αυτό θα εμφανίζει το γράμμα που του έχετε αντιστοιχίσει στην σωστή θέση έτσι ώστε αφού θα έχετε πατήσει όλα τα αντικείμενα να εμφανίζεται το όνομά σας. Για να είναι όμως πιο ενδιαφέρον το παιχνίδι, τα αντικείμενα θα κινούνται διαρκώς, όταν ακουμπούν στα όρια του παραθύρου θα αναπηδούν και θα στρίβουν δεξιόστροφα κατά 1 μοίρα και η κίνησή τους θα επαναλαμβάνεται μέχρι να κάνουμε κλικ με το ποντίκι πάνω τους.

2) Θέλουμε να κατασκευάσουμε ένα παιχνίδι με δύο αντικείμενα, έναν λαβύρινθο και ένα ανθρωπάκι που θα κινείται μέσα σ' αυτόν. Το παιχνίδι μας θα έχει 3 διαφορετικά επίπεδα λαβυρίνθων και με την ολοκλήρωση του προηγούμενου αυτομάτως θα μεταφέρει το χρήστη στο επόμενο επίπεδο. Το φόντο σε κάθε λαβύρινθο θα είναι κίτρινο, ενώ τα εμπόδια θα έχουν διαφορετικό χρώμα. Το σημείο τερματισμού θα είναι μπλε στα 2 πρώτα επίπεδα, ενώ στο τελευταίο θα είναι μωβ. Οι λαβύρινθοι θα είναι κλιμακούμενης δυσκολίας της επιλογής σας.

Το ανθρωπάκι θα κινείται με τα αντίστοιχα βέλη πάνω, κάτω, δεξιά, αριστερά κατά 5 βήματα κάθε φορά. Όμως κάθε στιγμή το ανθρωπάκι θα κινείται κατά τουλάχιστον 3 βήματα σε οποιαδήποτε κατεύθυνση είναι στραμμένο. Όταν πέφτει σε εμπόδιο ή τοίχο θα πρέπει να επιστρέφει στο σημείο εκκίνησης του εκάστοτε λαβυρίνθου που ορίζετε.



Κεφάλαιο 13: Επικοινωνία

Σε αυτό το κεφάλαιο:

13.1 Εντολές «μετάδωσε...» και «όταν λάβω...»

13.2 Εντολή «μετάδωσε...και περίμενε»

13.3 Παραδείγματα

«Το πρώτο μισό της ζωής μας ψάχνουμε κυρίως να βρούμε ποιοί είμαστε ανακαλύπτοντας τους εαυτούς μας μέσα από την αλληλεπίδρασή μας με άλλους».



13.1 Εντολές «μετάδωσε...» και «όταν λάβω...»

Στα προηγούμενα κεφάλαια μάθαμε πώς μπορούμε να κινήσουμε τα αντικείμενά μας, να ζωγραφίσουμε με αυτά, να αλλάξουμε τις όψεις τους και προσδιορίσαμε σύνθετα σενάρια που περιείχαν αλληλεπιδράσεις με τους χρήστες, επαναλήψεις, ελέγχους κτλ. Σε γενικές γραμμές όμως οι χαρακτήρες μας εκτελούσαν τις ενέργειές τους ανεξάρτητα από τα υπόλοιπα αντικείμενα του σκηνικού. Ο συγχρονισμός των διαφορετικών αντικειμένων μπορούσε να γίνει

- ✓ είτε **με εντολές χρόνου** για το συντονισμό τους πάνω σε ένα συγκεκριμένο χρονοδιάγραμμα, όπως με την εντολή **περίμενε...δευτερόλεπτα**
- ✓ είτε **με γεγονότα** που προκαλούσε ο χρήστης και που μπορούσαν να γίνουν ταυτόχρονα αντιληπτά από όλα τα αντικείμενα, όπως συμβαίνει με την εντολή **όταν το πλήκτρο...πατηθεί**
- ✓ είτε εφόσον υπήρχε **χωρική συσχέτιση** μεταξύ των αντικειμένων, για παράδειγμα με τη χρήση των τελεστών **αγγίζει το χρώμα ή το αντικείμενο**.

Σε αυτό το κεφάλαιο θα συζητήσουμε πως μπορούν να επικοινωνούν άμεσα και όχι έμμεσα τα αντικείμενα μεταξύ τους! Δηλαδή πως μπορούν να στέλνουν μηνύματα μεταξύ τους έτσι ώστε ο συγχρονισμός της συμπεριφοράς τους να γίνει πολύ ευκολότερος.

Ας διοργανώσουμε μια συναυλία στο Scratch! Καλέσαμε τον κύριο Μασετρίδη για μαέστρο, την Καλλιφωνίδου στα φωνητικά, τον Ντριγκιντράγκα στην κιθάρα και τον μοναδικό στο είδος του, κ. Πιατίνη στα ντραμς. Ο κύριος Μασετρίδης θα διευθύνει τη συναυλία μας, στέλνοντας μηνύματα στα αντικείμενα του σκηνικού προκειμένου ο καθένας να εκτελέσει την ενέργεια που του αναλογεί.



[13_π01.sb]

Δημιουργία αντικειμένων και σκηνικού: Η ορχήστρα μας λοιπόν αποτελείται από δύο μουσικούς (Ντριγκιντράγκα, Πιατίνη), έναν μαέστρο (Μασετρίδης) και μία τραγουδίστρια (Καλλιφωνίδου). Τους χαρακτήρες τους έχουμε βρει με την χρήση

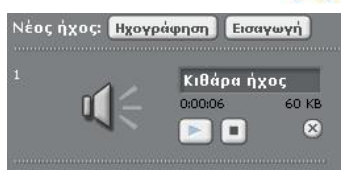
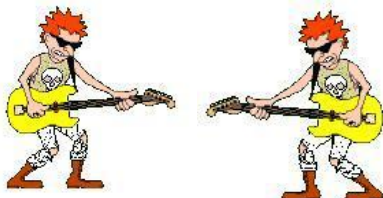
μιας μηχανής αναζήτησης (π.χ. Google) βάζοντας σαν όρους αναζήτησης τα όργανα που παίζουν ή την ιδιότητα τους στην παράσταση (για παράδειγμα για τον Ντριγκιντράγκα βάλαμε στα αγγλικά guitar player clipart, όπου το clipart προσδιορίζει ότι αναζητούμε κάποιο ήρωα με μορφή γραφικού). Το σκηνικό είναι διαθέσιμο στη βιβλιοθήκη υποβόθρων του Scratch και πιο συγκεκριμένα στο φάκελο «indoors» με το όνομα «spotlight-stage». Επίσης εισάγουμε και επιπλέον ενδυμασίες για τα αντικείμενα χρησιμοποιώντας των επεξεργαστή ζωγραφική και παραλλάσσοντας τη βασική ενδυμασία κάθε αντικειμένου. Θα κάνουμε εναλλαγή στις ενδυμασίες των αντικειμένων για να γίνει πιο αληθινή η συναυλία μας.

Μασετρίδης: Για τον μαέστρο μας έχουμε μία ενδυμασία. Ο ρόλος του είναι ο σημαντικότερος καθώς αυτός μεταδίδει τα μηνύματα που «προκαλούν» τις ενέργειες των άλλων αντικειμένων. Θα πρέπει να στέλνει μια οδηγία σε όλους τους μουσικούς της ορχήστρας ξεχωριστά, για να ακούσει τα όργανά τους, πριν τη συναυλία.





Ντριγκιντράγκας: Για τον Ντριγκιντράγκα θα δημιουργήσουμε δύο ενδυμασίες αλλά κυρίως θα πρέπει να προσθέσουμε τον ήχο που θα αναπαράγει όταν θα παίζει την κιθάρα του. Εμείς επιλέξαμε τον «GuitarChords2» από το φάκελο «music loops». Σε όλες τις περιπτώσεις, ο Ντριγκιντράγκας όταν λάβει σήμα από το Μασετρίδη, ξεκινάει αλλάζοντας ενδυμασία, στην συνέχεια παίζει τον ήχο και κάνει φιγούρες χρησιμοποιώντας εντολές που έχουμε μάθει στα προηγούμενα κεφάλαια.



Πιατίνης: Για τον Πιατίνη έχουμε μία ενδυμασία και έναν ήχο «Drum» που τον εισάγουμε επίσης από το φάκελο «music loops». Και ο κ.Πιατίνης όταν λάβει το σήμα που τον αφορά από το μάεστρο, ξεκινάει αλλάζοντας την όψη του με διάφορα εφέ και αναπαράγει τον αντίστοιχο ήχο.



Καλλιφωνίδου: Για την τραγουδίστρια μας δημιουργούμε δύο ενδυμασίες και εισάγουμε τον ήχο «got-inspiration» από το φά-

κελο «vocals». Η Καλλιφωνίδου όταν λάβει το σήμα αλλάζει ενδυμασία, κινείται στο σκηνικό και ταυτόχρονα αναπαράγει τον ήχο.



Σκηνικό: Για το σκηνικό θα χρησιμοποιήσουμε μόνο ένα υπόβαθρο, το οποίο θα αλλάζει με τη χρήση εφέ όταν ο κ. Μασετρίδης δίνει σήμα στους μουσικούς του να παίξουν μουσική. Π.χ.:



Η θεωρία πίσω από τα μηνύματα

Πως μπορεί ο Μασετρίδης να συντονίσει την ορχήστρα του; Ο πιο εύκολος τρόπος που θα μπορούσαμε να φανταστούμε, είναι να στέλνει μια μορφή μηνύματος κάθε φορά στο αντίστοιχο αντικείμενο-μουσικό και όταν ο μουσικός λαμβάνει το αντίστοιχο μήνυμα, να αντιδρά αναλόγως. Όπως δηλαδή ο χρήστης προκαλεί τη δημιουργία γεγονότων με την αλληλεπίδραση του με το ποντίκι και το πληκτρολόγιο, έτσι θα μπορούσε να προκαλέσει ένα γεγονός ένα αντικείμενο, το οποίο θα το «πιάσουν» και θα «χειριστούν» οι άλλοι χαρακτήρες.

Αυτή ακριβώς τη δυνατότητα μας δίνει το Scratch με τις εντολές **μετάδωσε...** και **όταν λάβω....** Η πρώτη δίνει τη δυνατότητα σε ένα αντικείμενο να στέλνει ένα μήνυμα, ενώ η δεύτερη επιτρέπει στα αντικείμενα να χειρίζονται τα μηνύματα που στέλνονται.



Και ποια είναι η μορφή του μηνύματος; Τα μηνύματα είναι απλές εκφράσεις π.χ. «ξεκινάμε», «τέλος», «γεια χαρά», κτλ. τις οποίες δημιουργούμε ελεύθερα κατά τη χρήση της εντολής **μετάδωσε...** Επιλέγουμε πατώντας το βελάκι «**νέο**» και μας εμφανίζεται ένα παράθυρο



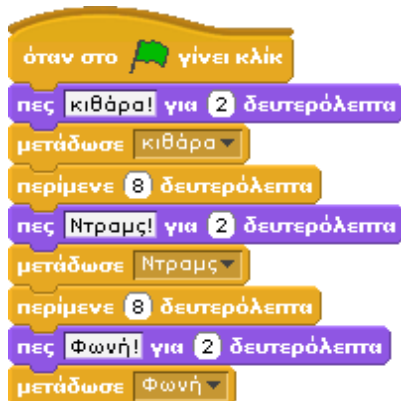
που μας ζητάει να πληκτρολογήσουμε το όνομα του μηνύματος. Μπορούμε να γράψουμε ότι μήνυμα θέλουμε χωρίς κανέναν περιορισμό τόσο για το μέγεθος όσο και για τους χαρακτήρες του πληκτρολογίου που θα συμπεριλάβουμε (κεφαλαία, μικρά, σύμβολα πράξεων). Καλό θα ήταν όμως να δημιουργούμε μηνύματα που θα μας θυμίζουν το «ρόλο» τους μέσα στα σενάρια γιατί θα τα χρησιμοποιούμε στη συνέχεια στις εντολές όταν λάβω.... στα αντικείμενα που θα «ακούν» τα συγκεκριμένα μηνύματα.

Η εντολή όταν λάβω.... έχει το ίδιο σχήμα με τις εντολές, όταν στην πράσινη σημαία γίνει κλικ, και όταν το πλήκτρο... πατηθεί και αυτό αποκαλύπτει το γεγονός ότι ουσιαστικά αποτελεί την αρχή ενός μικροσεναρίου. Δηλαδή αν το αντικείμενο λάβει το συγκεκριμένο μήνυμα, τότε εκτελεί ένα σενάριο που αντιστοιχεί μόνο σε αυτό το μήνυμα.

Λογικά, πρέπει να έχετε μια απορία. Ποιος είναι ο παραλήπτης του μηνύματος, αφού δεν τον προσδιορίζουμε;

Δημιουργούμε ένα μήνυμα αλλά σε ποιον το στέλνουμε; Η απάντηση είναι ότι στο Scratch κάθε μήνυμα στέλνεται σε όλα τα αντικείμενα και για αυτό δεν υπάρχει παραλήπτης. Κάθε μήνυμα που στέλνει ένα αντικείμενο, το «ακούν» όλα τα υπόλοιπα αντικείμενα. Αντιδρούν όμως μόνο αυτά που περιέχουν την εντολή όταν λάβω.... με παράμετρο το συγκεκριμένο μήνυμα. Άρα ένα μήνυμα το λαμβάνουν όλα τα αντικείμενα αλλά συνήθως το αξιοποιούν λίγα.

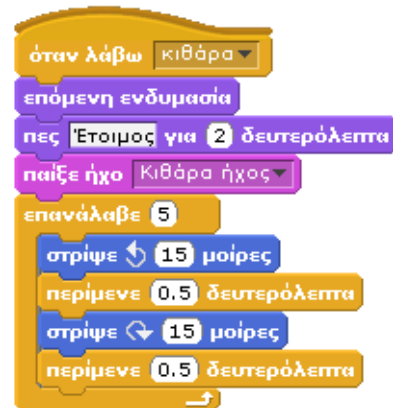
Ας επιστρέψουμε όμως στο παράδειγμά μας. Πως θα μπορούσε ο κ.Μαεστρίδης να δώσει με τη σειρά ένα μήνυμα στους διαφορετικούς μουσικούς για να τους ακούσει; Μελετήστε το παρακάτω σενάριο για τον κ.Μαεστρίδη:



Ο κ.Μαεστρίδης λέει το όργανο που θέλει να ακούσει και στέλνει το αντίστοιχο μήνυμα. Περιμένει για ένα χρονικό διάστημα για να ολοκληρώσει ο κάθε μουσικός το σενάριο-αντίδραση

στο μήνυμα και στη συνέχεια επαναλαμβάνει το ίδιο και με τους υπόλοιπους μουσικούς.

Τι μας μένει να δούμε; Το σενάριο σε έναν από τους μουσικούς που «ακούει» για ένα μήνυμα. Ας πάρουμε τον κ. Ντριγκιντράγκα που θα πρέπει να χειρίζεται και το πρώτο από τα τρία μηνύματα:



Σύμφωνα με το σενάριο αυτό, μόλις σταλεί το μήνυμα κιθάρα, ο κ. Ντριγκιντράγκας δηλώνει παρών παίζοντας ταυτόχρονα μία μελωδία στην κιθάρα του και κάνοντας χορευτικές φιγούρες. Ο χρόνος για την ολοκλήρωση του σεναρίου είναι 7 δευτερόλεπτα (2 για το έτοιμος και $5 * [0.5 + 0.5]$ για τις φιγούρες) και συνεπώς ο κ. Μαεστρίδης θα στείλει το επόμενο μήνυμα αφού ο κ. Ντριγκιντράγκας ολοκληρώσει την παρουσίασή του. Ήταν τόσο απλό. Με την εξαίρεση βέβαια του ότι πρέπει να υπολογίζουμε πάλι χρόνους για βεβαιωθούμε πότε πρέπει να σταλεί το επόμενο μήνυμα!

13.2 Εντολή "μετάδωσε... και περίμενε"

Το scratch όμως λύνει και αυτό το πρόβλημα με την εντολή μετάδωσε...και περίμενε. Με αυτή την εντολή, το αντικείμενο που μεταδίδει το μήνυμα, πριν συνεχίσει με την εκτέλεση της επόμενης εντολής, περιμένει τα υπόλοιπα αντικείμενα που θα λάβουν το μήνυμα να τελειώσουν την εκτέλεση των σχετικών σεναρίων. Δείτε πόσο πιο εύκολο γίνεται το προηγούμενο παράδειγμα με τη χρήση της εντολής μετάδωσε...και περίμενε:



[13_π02.sb]

Δηλαδή, αφού ο κ.Μαεστρίδης μεταδώσει το μήνυμα «κιθάρα» και αφού ολοκληρωθεί το σενάριο του Ντριγκιντράγκα, τότε θα εκτελεστεί η εντολή που ακολουθεί τη μετάδοση του μηνύματος. Είναι προφανές ότι είναι πλέον πολύ ευκολότερο να συγ-



χρονίσουμε τα αντικείμενα του σκηνικού μας και να υλοποιήσουμε πιο σύνθετα προγράμματα μέσα από το Scratch. Στη συνέχεια θα παρουσιάσουμε επιπλέον παραδείγματα επικοινωνίας αντικειμένων που αναδεικνύουν ακόμη καλύτερα τη χρησιμότητα των εντολών μετάδωσε..., μετάδωσε...και περίμενε και όταν λάβω....

13.3 Παραδείγματα

Παράδειγμα 1: Η τάξη: Ας αρχίσουμε με ένα απλό παράδειγμα. Σκεφτείτε το χώρο μάθησης, την τάξη σας, και αναλογιστείτε πόσες φορές οι καθηγητές σας θέτουν ερωτήσεις, δίνοντας έτσι το έναυσμα για μία συζήτηση. Από την πρώτη κιόλας τάξη του δημοτικού γνωρίζετε πως όταν θέλετε να απαντήσετε στον διδάσκοντα, σηκώνετε το χέρι προκειμένου να του γνωστοποιήσετε την επιθυμία σας. Δεν απαντάτε όμως αν αυτός δε σας δώσει το λόγο. Πώς όμως αυτός ο "άγραφος νόμος" της τάξης θα μπορούσε να εφαρμοστεί στο Scratch για να πραγματοποιηθεί ένας σύντομος διάλογος;

Ας υλοποιήσουμε, λοιπόν, μια αίθουσα διδασκαλίας στην οποία η καθηγήτρια των μαθηματικών θα θέτει στους μαθητές της Νίκο, Γιώργο, Σπύρο και Κώστα την ερώτηση «Ποιο είναι το αποτέλεσμα της πράξης $1+1$;» Μετά τη διατύπωση του ερωτήματος οι μαθητές προσπαθούν να βρουν τη σωστή απάντηση και ζητούν το λόγο σηκώνοντας το χέρι τους. Μπείτε εσείς στο ρόλο του καθηγητή και αποφασίστε, με τη βοήθεια του πληκτρολογίου σας, ποιος από τους 4 μαθητές θα απαντήσει. Όταν η καθηγήτρια θα απευθύνει το λόγο σε κάποιον μαθητή, θα πρέπει οι συμμαθητές του να το αντιλαμβάνονται και να σταματούν να ζητούν το λόγο κατεβάζοντας τα χέρια.

Το πρώτο πράγμα που μας απασχολεί για να ξεκινήσουμε τη σύνθεση του παραδείγματος είναι το ποια αντικείμενα συμμετέχουν. Εφόσον έχουμε μια καθηγήτρια και 4 μαθητές τα αντικείμενά μας θα είναι 5.

Τις ενδυμασίες για τα αντικείμενα μπορούμε να τα βρούμε από μια διαδικτυακή μηχανή αναζήτησης. Η καθηγήτρια απαιτεί μόνο μια ενδυμασία. Οι μαθητές όμως πρέπει να έχουν διαφορετικές ενδυμασίες προκειμένου να είναι διακριτή η διαφορετική κατάσταση στην οποία βρίσκονται. Για αυτό το λόγο χρειάζονται και τρεις ενδυμασίες: 1) παρακολουθώ, 2) σηκώνω το χέρι-ζητώ το λόγο και 3) απαντάω.



Ας "μπούμε", λοιπόν, στην τάξη που περιγράψαμε... Η καθηγήτρια με την έναρξη του σεναρίου πρέπει να θέσει την ερώτηση «ποιο το αποτέλεσμα της πράξης $1+1$ » στους μαθητές της και στη συνέχεια να μεταδώσει ένα μήνυμα «σκέψου» προκειμένου αυτοί να αρχίσουν να σκέφτονται την απάντηση. Θα χρησιμοποιήσει την εντολή μετάδωσε...και περίμενε ώστε αφού οι μα-

θητές έχουν σκεφτεί (έχουν δηλαδή ολοκληρώσει τα αντίστοιχα μικροσενάρια) στη συνέχεια θα μας ρωτήσει ποιος θέλουμε να απαντήσει.



Εφόσον εμείς επιλέξουμε ποιος μαθητής θα απαντήσει, η καθηγήτρια λαμβάνει το ερέθισμα μέσω της εντολής όταν στο πληκτρο...πατηθεί, και στέλνει ένα αντίστοιχο μήνυμα για να δώσει το λόγο μέσω της εντολής μετάδωσε... Η έκφραση που θα χρησιμοποιήσει μέσα στο μήνυμα είναι το όνομα του αντίστοιχου μαθητή που επιλέχθηκε.

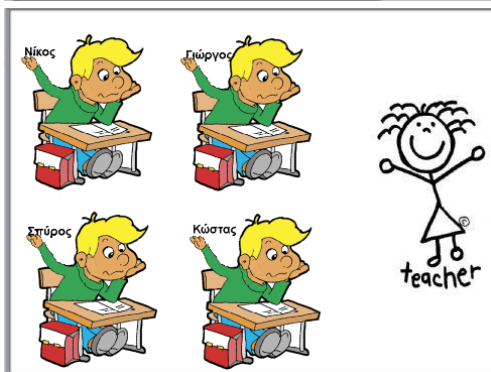
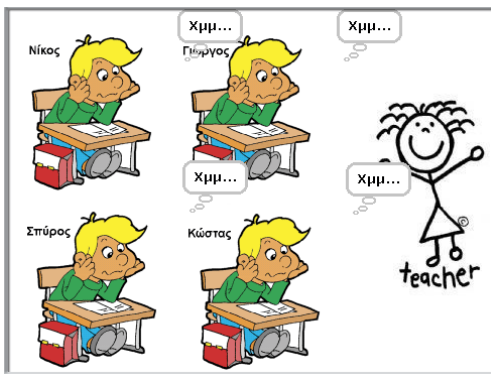
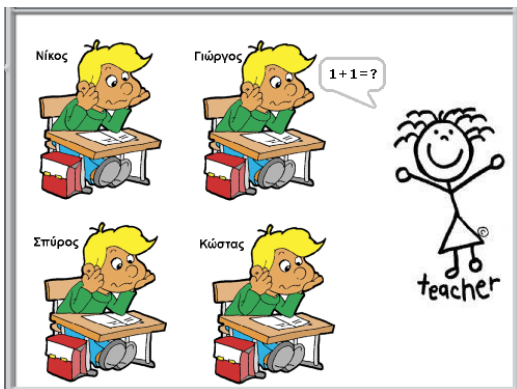
Έτσι, στα σενάρια της καθηγήτριας πρέπει να περιληφθούν και τα παρακάτω:

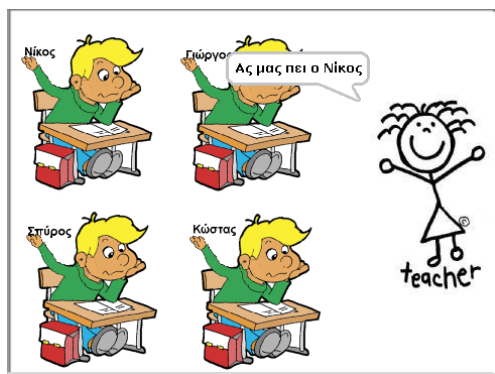
όταν το πλήκτρο 1 πατηθεί
πες Ας μας πει ο Νίκος για 2 δευτερόλεπτα
μετάδωσε Νίκος

όταν το πλήκτρο 2 πατηθεί
πες Ας μας πει ο Γιώργος για 2 δευτερόλεπτα
μετάδωσε Γιώργος

όταν το πλήκτρο 3 πατηθεί
πες Ας μας πει ο Σπύρος για 2 δευτερόλεπτα
μετάδωσε Σπύρος

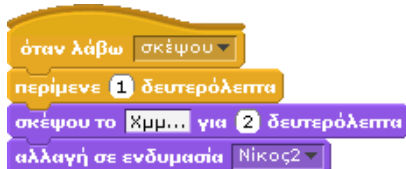
όταν το πλήκτρο 4 πατηθεί
πες Ας μας πει ο Κώστας για 2 δευτερόλεπτα
μετάδωσε Κώστας





Όπως είπαμε για τους μαθητές έχουμε 3 διαφορετικές ενδυμασίες. Στην πρώτη σκέφτονται, στη δεύτερη ζητούν το λόγο σηκώνοντας το χέρι και στην τρίτη απαντούν.

Αμέσως μετά τη διατύπωση της ερώτησης οι μαθητές λαμβάνουν το μήνυμα "σκέψου" που έχει σταλεί από την καθηγήτρια και αρχίζουν να σκέφτονται. Έτσι στο σενάριο κάθε μαθητή θα προσθέσουμε μια εντολή όταν λάβω... ώστε να αντιδράσουν στο αντίστοιχο μήνυμα:



Τέλος θα πρέπει ο κάθε μαθητής να χειρίζεται το μήνυμα που τον αφορά. Έτσι, στο μαθητή Νίκο θα προσθέσουμε μια εντολή όταν λάβω... προσδιορίζοντας ως μήνυμα το «Νίκος». Στο μικρο-σενάριο της συγκεκριμένης εντολής θα προσθέσουμε μια εντολή που θα κάνει το Νίκο να λείει το αποτέλεσμα και να φοράει την κατάλληλη τρίτη ενδυμασία του:



[13_π03.sb]

Δημιουργούμε αντίστοιχα σενάρια για τους υπόλοιπους μαθητές και το έργο μας ολοκληρώθηκε.

Είναι σημαντικό να κατανοήσουμε ότι

- ✓ ένα αντικείμενο μπορεί να λαμβάνει και να χειρίζεται περισσότερα από ένα μηνύματα
- ✓ και ότι το ίδιο μήνυμα μπορούν να το λαμβάνουν και να το χειρίζονται περισσότερα από ένα αντικείμενα.

Μελετήστε μόνοι σας ένα παραπλήσιο παράδειγμα από ένα φεστιβάλ χορού όπου ο χρήστης καλείται να επιλέξει ποιους χορευτές επιθυμεί να δει να χορεύουν:

[13_π04.sb]

Πως χρησιμοποιούνται τα μηνύματα στο συγκεκριμένο παράδειγμα;

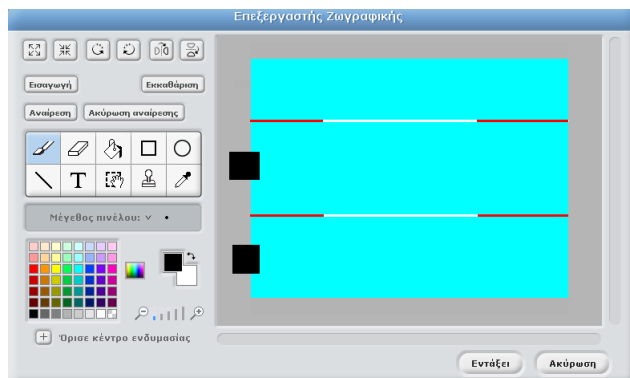


Παράδειγμα «Κολυμβητικοί αγώνες»: Οι θερινοί κολυμβητικοί αγώνες έχουν φτάσει στην τελική τους φάση. Η μοβ και η πορτοκαλί ομάδα είναι στον τελικό!!!!!! Οι κολυμβητές μας κα-λού-νται να τρέξουν το αγώνισμα της σκυτάλης 2x100m, δηλαδή οι ομάδες αποτελούνται από 2 κολυμβητές. Οι αθλητές μας ξεκινάνε από το βαθύρα της πισίνας (50m) και με το σφύριγμα του διαιτητή φεύγει η πρώτη σειρά. Το στυλ κολύμβησης είναι για τα πρώτα 50 μέτρα ελεύθερο. Όταν αυτοί φτάσουν και αγγίξουν το απέναντι άκρο της πισίνας, οι αθλητές της δεύτερης σειράς λαμβάνουν θέση εκκίνησης. Πέφτουν όμως μέσα στην πισίνα μόνο όταν ο συμπαίκτης τους έχει επιστρέψει - τερματίσει. Στην επιστροφή τους οι κολυμβητές κολυμπούν σε στυλ ύπτιο. Ο τερματισμός του κάθε παίκτη καθορίζεται από το άγγιγμά του στο βαθύρα. Με τον τερματισμό της δεύτερης σειράς αναδεικνύεται η νικήτρια ομάδα. Ας δημιουργήσουμε το πρόγραμμα που θα παρουσιάζει την παραπάνω σκυταλοδρομία.



Σκέψεις πριν την υλοποίηση: Αρχικά πρέπει να σκεφτούμε τα αντικείμενα που θα πρωταγωνιστούν καθώς και το σκηνικό στο οποίο θα τοποθετηθούν. Εφόσον έχουμε 2 ομάδες των 2 αθλητών, χρειαζόμαστε 4 αντικείμενα, (2 μοβ – 2 πορτοκαλί για να ξεχωρίζουν οι διαφορετικές ομάδες).

Τα αντικείμενα θα πρέπει να τοποθετηθούν σε ένα σκηνικό που θα έχει τη μορφή πισίνας, στο οποίο θα απεικονίζονται και οι βαθύρες που αποτελούν τα σημεία έναρξης και τερματισμού της κούρσας. Για να δημιουργήσουμε το σκηνικό θα χρησιμοποιήσουμε τον επεξεργαστή ζωγραφικής και θα δημιουργήσουμε ένα υπόβαθρο όμοιο με αυτό της επόμενης εικόνας.



Ως αντικείμενα-κολυμβητές θα επιλέξουμε τα «diver1» και «diver2» από τον κατάλογο «People» της βιβλιοθήκη αντικειμένων του Scratch ενώ για διαιτητή μας θα επιλέξουμε τη μορφή «referee2». Ας μελετήσουμε τα αντικείμενά μας ξεχωριστά.

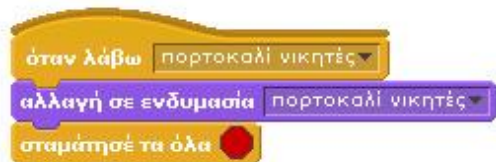
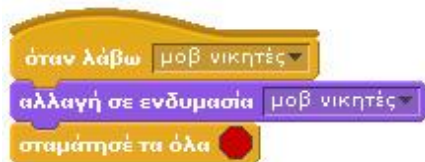


Διαιτητής: Ο διαιτητής διαδραματίζει ένα σημαντικό ρόλο στο πρόγραμμά μας καθώς α) είναι το αντικείμενο που πρώτο δίνει το έναυσμα για να ξεκινήσει η σκυταλοδρομία και β) δεύτερον, θα πρέπει να αναδείξει τον νικητή με τον τερματισμό των αθλητών. Άρα στην αρχή του έργου μας ο διαιτητής θα πρέπει να στείλει το μήνυμα «πάμε», ενώ στο τέλος του προγράμματος θα πρέπει να περιμένει τα μήνυμα από τον κολυμβητή που άγγιξε το βατήρα πρώτος, για να τον ανακηρύξει και νικητή:

Για την αποστολή του πρώτου μηνύματος, θα δημιουργήσουμε το σενάριο:



Για τη λήψη μηνύματος από τον νικητή, θα δημιουργήσουμε τα σενάρια:



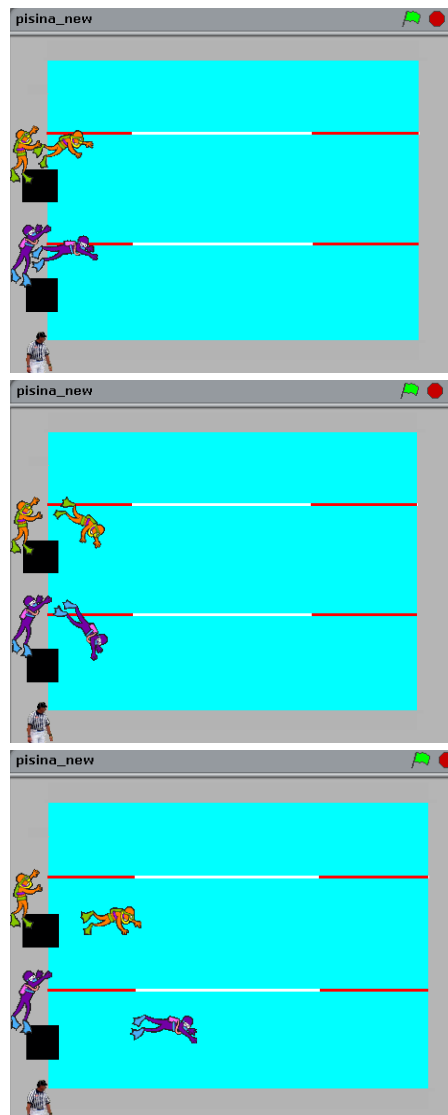
Για την ανάδειξη του νικητή, έχουμε δημιουργήσει για το διαιτητή δυο επιπλέον ενδυμασίες, μια στην οποία έχουμε γράψει μωβ νικητές και μια στην οποία έχουμε γράψει πορτοκαλί νικητές. Με τη λήψη του μηνύματος, ο διαιτητής φοράει την κατάλληλη ενδυμασία. Λίγο περίεργο; Γιατί δεν χρησιμοποιήσαμε την

εντολή πες...για...δευτερόλεπτα; Ας απαντήσουμε σε αυτό το ερώτημα στο τέλος.

Κολυμβητές 1^{ης} σειράς: Πρέπει αρχικά να φέρουμε τους κολυμβητές μας στην κατάλληλη τους θέση, δηλαδή να τους περιστρέψουμε (για να εμφανίζονται όρθιοι) και έπειτα να προσδιορίσουμε τη θέση τους στο βατήρα.

Οι αθλητές της πρώτης σειράς ξεκινούν το αγώνισμα με το σφύριγμα του διαιτητή. Επομένως στα σενάρια τους θα πρέπει να χειρίζονται το μήνυμα «πάμε» με τη χρήση της εντολής όταν λάβω.... Για να μη μπερδευτούμε, πρέπει να σημειώσουμε ότι τα αντικείμενα της δεύτερης σειράς δεν έχουν καμία σχέση με αυτό το μήνυμα.

Για να βουτήξουν οι αθλητές μας, θα πρέπει οι κολυμβητές να ακολουθήσουν μια καμπυλοειδή κίνηση κατά τη διάρκεια της βουτιάς τους στο νερό. Θα προσπαθήσουμε να οπτικοποιήσουμε όσο το δυνατόν καλύτερα την πραγματική τους κίνηση χρησιμοποιώντας τις εντολές κινήσου...βήματα, στρίψε δεξιό-στροφα...μοίρες μέσα σε ένα βρόχο επανάληψης. Θα περιστρέψουμε τους αθλητές πρώτα αριστερόστροφα και στη συνέχεια δεξιόστροφα για να είμαστε πειστικοί. Τέλος με τη χρήση της εντολής δείξε στην κατεύθυνση 90 ο αθλητής μας ακολουθεί οριζόντια κίνηση μέσα στην πισίνα. Οι προηγούμενες κινήσεις φαίνονται στις επόμενες εικόνες.



Όταν πέσουν στην πισίνα μας, οι πρώτοι κολυμβητές μας κινούνται προς το δεξί άκρο της πισίνας μέχρι να το φτάσουν (γκρι χρώμα) οπότε αλλάζουν κατεύθυνση και μεταδίδουν μήνυμα στους συμπαίκτες τους (κάθε αθλητής στον συμπαίκτη του) να λάβουν θέση στο βατήρα. Η κίνηση των αθλητών συνεχίζεται μέχρι να ακουμπήσουν στο βατήρα, οπότε στη συνέχεια πρέπει να στείλουν νέο μήνυμα στους συμπαίκτες τους για να ξεκινήσουν με τη σειρά τους το κολύμπι.

Άρα,

Α) χρειαζόμαστε μια επανάληψη της κίνησης έως ότου οι κολυμβητές ακουμπήσουν το μαύρο χρώμα του βατήρα, οπότε στη συνέχεια βγαίνουν από την πισίνα (απόκρυψη) και στέλνουν μήνυμα «πάμε Χ» για να πάσει ο συμπαίκτης τους

Β) όταν ακουμπήσουν το γκρι χρώμα πρέπει να αλλάξουν κατεύθυνση και να κινηθούν προς τους βατήρες, ενώ ταυτόχρονα πρέπει να στείλουν το μήνυμα «λάβε θέση Χ» στους συμπαίκτες τους,

Γ) με κάποιο τρόπο θα πρέπει να μην είναι προκαθορισμένη η ταχύτητα της κίνησής τους ώστε ο τελικός νικητής να μπορεί να είναι διαφορετικός κάθε φορά.

Για να πετύχουμε τον

τελευταίο στόχο, θα **τυχαία επιλογή από 1 μέχρι 10** χρησιμοποιήσουμε την εντολή **κινήσου...βήματα** προσδιορίζοντας όμως τυχαίο αριθμό βημάτων με τη χρήση του τελεστή **τυχαία επιλογή από...μέχρι.....** Πρέπει να κάνουμε το παιχνίδι λίγο πιο ενδιαφέρον. Η εντολή **τυχαία επιλογή από...μέχρι.....** κάθε φορά που εκτελείται η **κινήσου...βήματα** παράγει μια νέα τιμή τυχαία ανάμεσα στα όρια τα οποία θέτουμε στα δυο λευκά κουτάκια της. Επομένως, τα βήματα δεν είναι σταθερά.

Με βάση τις προηγούμενες περιγραφές, τα σενάρια ενός εκ των δυο κολυμβητών της πρώτης σειράς είναι τα εξής:



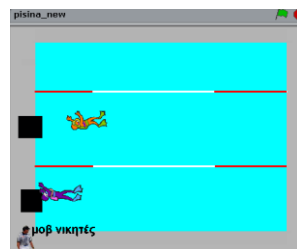
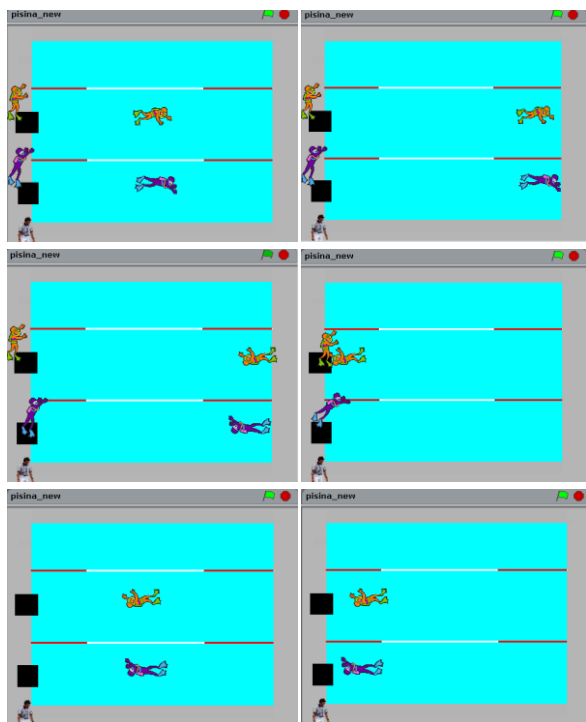
Κολυμβητές 2^{ης} σειράς: Όπως αναφέραμε και αρχικά οι αθλητές της δεύτερης σειράς είναι όμοιοι(σε εμφάνιση) με εκείνους της πρώτης. Αφού λοιπόν τους περιστρέψουμε κατάλληλα, τους τοποθετούμε σε θέση αναμονής δίπλα στο βατήρα όπως φάνηκε στις προηγούμενες εικόνες.

Στη συνέχεια οι αθλητές μας πρέπει να περιμένουν για να λάβουν το μήνυμα των συναθλητών τους με την εντολή **όταν λάβω....** προκειμένου να πάρουν θέση εκκίνησης (ο κάθε ένας ξεχωριστά). Όταν σταλεί το μήνυμα προχωρούν στο βατήρα και ετοιμάζονται για να βουτήξουν. Μόλις οι αθλητές της πρώτης σειράς ακουμπήσουν το μαύρο βατήρα, στέλνουν δεύτερο μήνυμα που λαμβάνεται πάλι από τους συναθλητές τους (κάθε έναν ξεχωριστά) και βάση του οποίου ξεκινά και για αυτούς το κολύμπι, όπως με τους πρώτους αθλητές. Οι δυο αθλητές όμως στο τέλος δε στέλνουν δυο μηνύματα, όπως οι προηγούμενοι συναθλητές τους αλλά αυτός που προλαβαίνει να φτάσει πρώτος καταφέρνει να στείλει ένα μήνυμα που θα λάβει ο διαιτητής. Παρακάτω παρουσιάζεται το σενάριο ενός αθλητή της δεύτερης σειράς



[13_π05.sb]

Το αποτέλεσμα εντυπωσιακό, πολύ αναπαραστατικό και απρόβλεπτο! Τα σημαντικότερα στιγμιότυπα του αγώνα:



Μπορείτε τώρα να απαντήσετε για ποιο λόγο στο διαιτητή δεν χρησιμοποιήσαμε την εντολή πες...για...δευτερόλεπτα και χρησιμοποιήσαμε την εντολή άλλαξε ενδυμασία; Παρατηρήστε ποια εντολή ακολουθεί την εντολή άλλαξε ενδυμασία..

Περίληψη

Η επικοινωνία μεταξύ των αντικειμένων στο Scratch ήταν ένα στοιχείο που μας είχε προβληματίσει στα προηγούμενα κεφάλαια. Σε αυτό το κεφάλαιο είδαμε ένα πολύ απλό μηχανισμό επικοινωνίας, όπου τα αντικείμενα στέλνουν μέσω των εντολών μετάδωσε και μετάδωσε και περίμενε μηνύματα ενώ μέσω της εντολής όταν λάβω... μπορούν να χειριστούν όσα από αυτά επιθυμούν. Καλό είναι να επιλέγουμε ονόματα μηνυμάτων που είναι σχετικά με τα αποτελέσματα που προκαλούν, ώστε να τα αναγνωρίζουμε ευκολότερα.

Ερωτήσεις

- 1) Ποιες εντολές χρησιμοποιούμε για την ανταλλαγή μηνυμάτων μεταξύ των αντικειμένων;
- 2) Έστω ότι έχουμε 2 αντικείμενα και το 1ο στέλνει ένα μήνυμα στο 2ο με την εντολή μετάδωσε ξεκίνα.

Ποια εντολή θα πρέπει να έχει στο σενάριό του το 2ο αντικείμενο ώστε να λάβει το μήνυμα;

- 3) Συμπληρώστε τα κενά

- 'Όταν θέλουμε να πετύχουμε την μεταξύ αντικειμένων, τότε ένα αντικείμενο στέλνει σε κάποιο άλλο. Το δεύτερο, όταν αυτό το μήνυμα, εκτελεί κάποιο σενάριο.

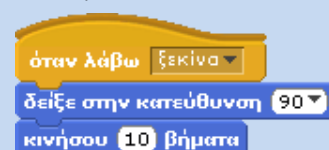
- Με την εντολή το αντικείμενο που την εκτελεί περιμένει, αφού στείλει το μήνυμα, μέχρι να εκτελέσει το σενάριο του το αντικείμενο ή τα αντικείμενα που έλαβαν αυτό το μήνυμα.

- 4) Αναλύστε τι προκαλούν τα παρακάτω σενάρια:

Αντικείμενο1



Αντικείμενο 2



- 5) Θα συγχαρεί ποτέ ο προπονητής τους αθλητές του στα παρακάτω σενάρια; Δικαιολογήστε την απάντησή σας.

Προπονητής

όταν στο  γίνει κλικ
μετάδωσε ξεκίνα και περίμενε
πες Μπράβο!

Αθλητής 1

όταν λάβω ξεκίνα
επανάλαβε 10
 δείξε στην κατεύθυνση 90
 κινήσου 10 βήματα

Αθλητής 2

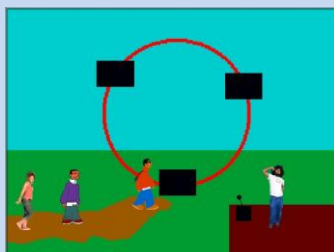
όταν λάβω ξεκίνα
για πάντα
 δείξε στην κατεύθυνση 90
 κινήσου 10 βήματα

6) Επιλέξτε ένα πρόγραμμα του κεφαλαίου που περιέχει την εντολή μετάδωσε...και περίμενε και τροποποιήστε τον κώδικα ώστε να εκτελεί την ίδια λειτουργία με το συνδυασμό των εντολών μετάδωσε... και περίμενε...δευτερόλεπτα. Μπορούμε πάντα να αντικαταστήσουμε την μετάδωσε...και περίμενε με άλλες εντολές;

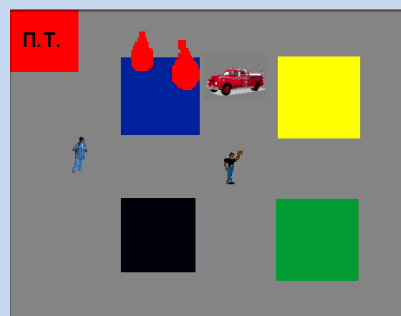
Δραστηριότητες

1) Βρείτε 10 διαφορετικές σημαίες χωρών από το διαδίκτυο και εισάγετε τις εικόνες αυτές ως αντικείμενα. Με τη βοήθεια μηνυμάτων υλοποιήστε ένα σενάριο στο οποίο όταν ο χρήστης κάνει κλικ με το ποντίκι πάνω σε κάποια σημαία, τότε να εμφανίζεται το όνομά της.

2) Βρισκόμαστε σε ένα λούνα παρκ και συγκεκριμένα στη ρόδα. Στην ουρά περιμένουν τρία παιδιά και απέναντί τους βρίσκεται ο εισπράκτορας, ο οποίος ελέγχει τη ρόδα με ένα μοχλό. Η ρόδα περιέχει τρία βαγόνια, ένα για κάθε παιδί. Ο πρώτος στην ουρά περιμένει σήμα από τον εισπράκτορα και όταν το λάβει ανεβαίνει στο πρώτο βαγόνι, ενώ ο δεύτερος κατά σειρά παίρνει τη θέση του για να μπει στη συνέχεια στο δεύτερο βαγόνι. Ο εισπράκτορας αλλάζει τη θέση του μοχλού και η ρόδα ξεκινά να γυρίζει. Όταν το επόμενο ελεύθερο βαγόνι βρεθεί σε θέση επιβίβασης, ο εισπράκτορας σταματά τη ρόδα και ειδοποιεί το επόμενο παιδί να ανέβει. Δημιουργήστε το κατάλληλο σκηνικό, αναγνωρίστε τα αντικείμενα που συμμετέχουν και με τη βοήθεια των εντολών του κεφαλαίου υλοποιήστε το παραπάνω σενάριο.



3) Έστω ότι έχουμε την κάτοψη μίας πόλης με πέντε κτήρια και το δρόμο ανάμεσά τους. Τα τέσσερα κτήρια είναι διαφορετικού χρώματος το καθένα (μαύρο, πράσινο, μπλε, κίτρινο) και το ένα είναι το πυροσβεστικό τμήμα της πόλης, το οποίο βρίσκεται στην πάνω αριστερή γωνία της. Στην πόλη κυκλοφορεί ένας εμπρηστής, ο οποίος κινείται από το χρήστη πάνω, κάτω, αριστερά και δεξιά. Όταν ακουμπήσει κάποιο κτήριο, αυτό παίρνει φωτιά (αλλαγή ενδυμασίας) και αυτός κρύβεται. Τα κτήρια παρακολουθούνται από ένα φύλακα, ο οποίος βρίσκεται στο κέντρο της πόλης και καλεί την πυροσβεστική μόλις αντιληφθεί τη φωτιά. Ένα όχημα έρχεται από το τμήμα, σβήνει τη φωτιά και επιστρέφει σε αυτό, ενώ τότε ο εμπρηστής εμφανίζεται στην αρχική του θέση. Υλοποιήστε το παραπάνω πρόγραμμα χρησιμοποιώντας μηνύματα στις ακόλουθες περιπτώσεις: α) Ο εμπρηστής στέλνει μήνυμα στο κτήριο και στο φύλακα ότι άναψε φωτιά, β) ο φύλακας στέλνει μήνυμα στην πυροσβεστική για να στείλει πυροσβεστικό όχημα, γ) το όχημα στέλνει μήνυμα στο κτήριο ότι έσβησε τη φωτιά και στον εμπρηστή να εμφανιστεί πάλι στην αρχική του θέση.





Κεφ. 14: Μεταβλητές-Λίστες

Σε αυτό το κεφάλαιο:

14.1 Εισαγωγή στην έννοια των μεταβλητών

14.2 Λίστες

14.3 Παραδείγματα

«Το πιο απίθανο πράγμα με τη ζωή είναι ότι πάντα θα υπάρχουν μεταβλητές.

Θα πρέπει να τις δεις και να τις αναγνωρίσεις για να μάθεις πώς να αλληλεπιδράς μ' αυτές».

(Suzanne Farrell)



14.1. Εισαγωγή στην έννοια των μεταβλητών

14.1.1 Οι μεταβλητές

Θα πρέπει να έχετε παρατηρήσει ότι έχουμε φτιάξει τόσα παιχνίδια μέχρι αυτό το σημείο και δεν έχουμε αναφερθεί πουθενά για το πως μπορούμε να δημιουργήσουμε τα πιο σημαντικά στοιχεία όλων των παιχνιδιών: ζωές και σκορ! Θα μπορούσαμε με κάποιο τρόπο να κρατάμε το σκορ στις προσπάθειες που κάναμε για μεταφερθεί το ελικοπτεράκι από την κίτρινη πλατφόρμα στην κόκκινη; Θα μπορούσαμε να έχουμε ένα μέγιστο αριθμό προσπαθειών πέρα από τον οποίο θα σταματούσε η εκτέλεση του έργου μας, δηλαδή θα τελείωνε το παιχνίδι; Θα μπορούσαμε να εμφανίσουμε τις ζωές του rasman που μας απομένουν και το σκορ που μαζεύει τρώγοντας φρουτάκια;

Προφανώς! Είπαμε ότι στο τέλος του βιβλίου θα γνωρίζετε να δημιουργείτε ολοκληρωμένα παιχνίδια. Δεν θα ξελέμε τώρα!

Πως, λοιπόν, θα διατηρούμε το σκορ καθ' όλη τη διάρκεια εκτέλεσης ενός έργου; Θα το αποθηκεύσουμε ως μια μεταβλητή. Που; Στη μνήμη του υπολογιστή.

Οι μεταβλητές είναι συμβολικά ονόματα που αντιστοιχούν σε θέσεις μνήμης του υπολογιστή. Στις θέσεις μνήμης αυτές, αποθηκεύουμε διάφορες τιμές (π.χ. το σκορ, τον αριθμό ζωών, την απάντηση ενός χρήστη κτλ). Οι τιμές αυτές μπορούν να αλλάξουν, να διαβαστούν ή να αντιγραφούν. Μπορείτε να τις φανταστείτε σαν κουτάκια που έχουν ένα όνομα και μια τιμή. Στην παρακάτω εικόνα, θεωρώντας ότι ο πίνακας είναι η μνήμη του υπολογιστή, βλέπουμε δυο θέσεις στη μνήμη που αντιστοιχίζονται στις μεταβλητές «σκορ» και «ζωές» που έχουν τιμές 300 και 3 αντίστοιχα.

Όνομα	Τιμή	Όνομα	Τιμή
σκορ	300	ζωές	3

Άρα, όταν δημιουργούμε μια μεταβλητή, ουσιαστικά δεσμεύουμε μια θέση στη μνήμη του υπολογιστή στην οποία μπορούμε να αποθηκεύσουμε όποια τιμή θελήσουμε.

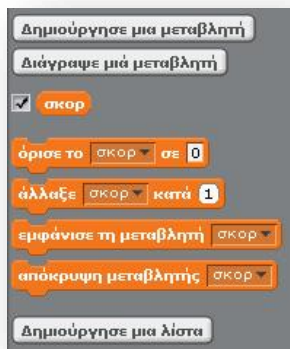
Στο Scratch, για να δημιουργήσουμε μια μεταβλητή πρέπει αρχικά να μεταφερθούμε στην παλέτα εντολών μεταβλητές και να πατήσουμε στην επιλογή «Δημιούργησε μια μεταβλητή» οπότε εμφανίζεται το παρακάτω παράθυρο:

Στο παράθυρο αυτό πρέπει να προσδιορίσουμε:

α) το όνομα της μεταβλητής: με την ονομασία της μεταβλητής προσδιορίζουμε τον τρόπο με τον οποίο θα αναφερόμαστε στη συγκεκριμένη θέση μνήμης και θα αλλάζουμε τα περιεχόμενά της, δηλαδή την τιμή της. Καλό είναι στις μεταβλητές που δημιουργούμε να δίνουμε ονόματα με νόημα, δηλαδή ονόματα που προσδιορίζουν τη χρησιμότητα της μεταβλητής. Στο συγκεκριμένο παράδειγμα θα ονομάσουμε τη μεταβλητή μας «σκορ».

β) το εύρος χρήσης της μεταβλητής: θα πρέπει επίσης να προσδιορίσουμε αν η νέα μεταβλητή θέλουμε να είναι ορατή και να τροποποιείται από όλα τα αντικείμενα του έργου μας ή αν θέλουμε να αξιοποιείται μόνο από ένα συγκεκριμένο αντικείμενο. Στη δεύτερη περίπτωση, η μεταβλητή αυτή μπορεί να χρησιμοποιηθεί μόνο σε σενάρια του συγκεκριμένου αντικειμένου ενώ στην πρώτη, όλα τα αντικείμενα μπορούν να αλλάξουν την τιμή της μεταβλητής.

Αφού δημιουργηθεί η μεταβλητή, ως δια μαγείας, η παλέτα μεταβλητές εμφανίζει νέες εντολές!



Η πρώτη εντολή όρισε το X σε 0, είναι η σημαντικότερη, καθώς μας δίνει τη δυνατότητα να δώσουμε μια τιμή στη μεταβλητή μας συμπληρώνοντας το λευκό κουτάκι. Όταν δημιουργούμε μια μεταβλητή, η αρχική της τιμή είναι κενή. Με τη χρήση της εντολής όρισε το X σε 0 μπορούμε να της δώσουμε αρχική τιμή αλλά και να αλλάξουμε την τιμή αυτή. Π.χ. σε ένα παιχνίδι θα ορίζαμε τη μεταβλητή «ζωές» σε 3 αρχικά ενώ τη μεταβλητή «σκορ» σε 0. Οι μεταβλητές μπορούν να πάρουν και αρνητικές τιμές.

Η δεύτερη εντολή που γίνεται διαθέσιμη είναι η άλλαξε X κατά 1, η οποία αλλάζει την τρέχουσα τιμή της μεταβλητής κατά την τιμή που προσδιορίζουμε στο λευκό κουτάκι. Αλλάξε την τιμή κατά 1, σημαίνει αύξηση της τιμής της μεταβλητής κατά 1 μονάδα ενώ αύξηση κατά -5 σημαίνει μείωση της αρχικής τιμής κατά 5 μονάδες. Το μεγάλο πλεονέκτημα της συγκεκριμένης εντολής είναι μας επιτρέπει να αλλάζουμε την τιμή της μεταβλητής χωρίς να ενδιαφερόμαστε για το ποια είναι η τρέχουσα τιμή της. Αν εκτελούσαμε τη συγκεκριμένη εντολή για τη μεταβλητή «σκορ», τότε το σκορ θα αυξανόταν κατά 1 ενώ αν εκτελούσαμε την εντολή για τη μεταβλητή «ζωές» με παράμετρο -1, η τιμή της αντίστοιχης μεταβλητής θα γινόταν 2.

Η τρίτη και η τέταρτη εντολή (εμφάνισε τη μεταβλητή X, απόκρυψη μεταβλητής X) αναφέρονται στην εμφάνιση και στην απόκρυψη μια μεταβλητής από την οθόνη του Scratch. Σε αντίθεση με άλλες μεταβλητές κατάστασης, όπως η θέση X και η ενδυμασία #, η εμφάνιση των τιμών των μεταβλητών στην οθόνη του Scratch ελέγχεται προγραμματιστικά. Έτσι, για παράδειγμα, όταν ο χρήστης μας χάσει μια ζωή, μπορούμε να εμφανίσουμε τη μεταβλητή «ζωές» για λίγα δευτερόλεπτα και στη συνέχεια να την αποκρύψουμε.

Ας δούμε πιο πρακτικά τη χρησιμότητα και τη λειτουργία των μεταβλητών μέσα από παραδείγματα. Δεν θα παρουσιάσουμε ολοκληρωμένα παραδείγματα αλλά μόνο τα τμήματά τους που αφορούν τη λειτουργία των μεταβλητών.

14.1.2 Παραδείγματα εφαρμογών με μεταβλητές

Παράδειγμα: Ο «κ.Γατίδης μετράει ως το 10»: Τι θα λέγατε να βάλουμε τον κ.Γατίδη να μετράει μέχρι το 10 με τη χρήση των μεταβλητών. Η πρώτη λύση που μας έρχεται στο μυαλό:



Τι θα συνέβαινε όμως αν θέλαμε να μετρήσει μέχρι το 30; Θα δημιουργούσαμε 30 εντολές; Θα μπορούσαμε αντί για αυτή τη λύση, να δημιουργήσουμε μια μεταβλητή, να αυξάνουμε κατά μια μονάδα την τιμή της και να βάλουμε το χαρακτήρα μας να αναφέρει την τιμή αυτή μέσα σε μια επανάληψη; Δεν θα ήταν πιο εύκολο;

Χρειαζόμαστε λοιπόν μια μεταβλητή που θα την ονομάσουμε «αριθμός» και την οποία θα κάνουμε διαθέσιμη μόνο στον κ.Γατίδη. Από την παλέτα Μεταβλητές επιλέγουμε «δημιούργησε μια μεταβλητή», προσδιορίζουμε ως όνομα της μεταβλητής το «αριθμός» και επιλέγουμε η μεταβλητή να είναι διαθέσιμη μόνο για το συγκεκριμένο αντικείμενο.

Θα δώσουμε στη μεταβλητή, αρχική τιμή 1 και στη συνέχεια μέσα σε μια επανάληψη θα βάλουμε τον κ.Γατίδη να λέει την τιμή της μεταβλητής ενώ θα αλλάζουμε την τιμή της κατά 1. Μελετήστε το παρακάτω σενάριο:



[14_π01.sb]

Όπως παρατηρούμε από το συγκεκριμένο παράδειγμα, θα μπορούσαμε πολύ εύκολα να βάλουμε τον κ.Γατίδη να μετράει μέχρι το 1000 απλά αλλάζοντας τον αριθμό των επαναλήψεων στην εντολή επανάλαβε X!

Για να μεταφέρουμε τη μεταβλητή μέσα σε ένα λευκό κουτάκι, επιλέγουμε και σέρνουμε την αντίστοιχη μεταβλητή κατάστασης που εμφανίζεται στην παλέτα Μεταβλητές.

Επίσης, είναι σημαντικό να θυμόμαστε:

- α) ότι για κάθε μεταβλητή στην αρχή ενός προγράμματος θα πρέπει να προσδιορίζουμε την αρχική τιμή της,
- β) η μεταβλητή μπορεί να χρησιμοποιηθεί σε οποιοδήποτε λευκό κουτάκι των εντολών του Scratch (π.χ. στην κίνηση, στη ζωγραφική, στον ήχο, στις συνθήκες κτλ.).

Παράδειγμα: χρονομετρητής: Έστω λοιπόν ότι θέλουμε να δημιουργήσουμε ένα χρονόμετρο για ένα παιχνίδι μας. Ο παίκτης

θα μπορεί να παίξει το παιχνίδι για 60 δευτερόλεπτα. Πως θα μπορούσαμε να το καταφέρουμε;



[14_π02.sb]

Όπως βλέπετε στο προηγούμενο σενάριο, ο προγραμματιστής μας, δημιούργησε μια μεταβλητή που την ονόμασε «ρολόι». Στόχος του ήταν η μεταβλητή αυτή να κρατά κάθε στιγμή το χρόνο που απομένει στον παίκτη. Στη συνέχεια, ο προγραμματιστής μας προσδιόρισε ως αρχική τιμή της μεταβλητής το 60, δηλαδή 60 δευτερόλεπτα παιχνιδιού. Χρησιμοποιώντας την εντολή εμφάνισε μεταβλητή... εμφάνισε τη μεταβλητή αυτή στην οθόνη του χρήστη ώστε ο παίκτης να γνωρίζει το χρόνο που του απομένει. Στη συνέχεια ζήτησε κάθε ένα δευτερόλεπτο να αλλάζει η τιμή της μεταβλητής «ρολόι», μέχρι η τιμή της να γίνει μηδέν. Αν η μεταβλητή μηδενιστεί, η επανάληψη σταματά, εκτελούνται οι δυο τελευταίες εντολές του σεναρίου και το παιχνίδι σταματά!

Προσέξτε ότι:

- ✓ έχουμε αρνητική τιμή στην εντολή άλλαξε ρολόι κατά...
- ✓ με την εντολή εμφάνισε μεταβλητή... παρουσιάζεται η τιμή της μεταβλητής στην οθόνη του Scratch,
- ✓ η μεταβλητή χρησιμοποιήθηκε μέσα σε μια συνθήκη.

Αν θέλαμε κάθε φορά που ο χρήστης έτρωγε ένα φρουτάκι να κερδίζει έξτρα χρόνο, τι θα έπρεπε να κάνουμε; Πολύ απλά, θα μπορούσαμε να χρησιμοποιήσουμε την εντολή άλλαξε ρολόι κατά... σε κάποιο διαφορετικό σενάριο και κάθε φορά που έτρωγε ο πρωταγωνιστής μας ένα φρουτάκι να προσθέταμε τον αντίστοιχο χρόνο που κέρδιζε στη μεταβλητή «ρολόι». Το παραπάνω σενάριο θα λειτουργούσε χωρίς κανένα πρόβλημα. Αυτό είναι και το πλεονέκτημα της εντολής επανάλαβε ώσπου... αντί της επανάλαβε X στη συγκεκριμένη περίπτωση. Μπορείτε να το εξηγήσετε καλύτερα;

Παράδειγμα: «Μακριά από το δράκο»: Έστω, λοιπόν, ότι έχουμε δημιουργήσει ένα παιχνίδι στο οποίο ο χρήστης προσπαθεί να κάνει κλικ σε όσο το δυνατόν περισσότερα Στρουμφάκια για να τα απελευθερώσει από το κάστρο του Δρακουμέλ. Αυτό όμως πρέπει να γίνει χωρίς να κάνουμε κλικ πάνω στο Δρακουμέλ που κινείται μανιωδώς προς το δέικτη του ποντικιού. Έστω ότι έχουμε 3 ζωές. Πως θα μπορούσαμε να υλοποιήσουμε το συγκεκριμένο παιχνίδι. Μελετήστε το παρακάτω σενάριο:



Στο συγκεκριμένο παράδειγμα, δημιουργήσαμε μια μεταβλητή που ονομάσαμε «ζωές» και η οποία προσδιορίσαμε ότι θέλουμε να είναι ορατή από όλα τα αντικείμενα του έργου μας. Στην αρχή του έργου μας, ο ήρωας έχει 3 ζωές και στη συνέχεια ζητήσαμε να συνεχίσει να εκτελείται το έργο μας μέχρι οι ζωές να γίνουν 0. Τότε εμφανίζεται μήνυμα τέλους του παιχνιδιού και σταματά η εκτέλεση του έργου. Πως όμως θα μειώνονται οι ζωές μας; Από το σενάριο του Δρακουμέλ. Μη ξεχνάτε ότι και ο Δρακουμέλ μπορεί να πειράζει αυτή τη μεταβλητή αφού είναι ορατή σε αυτόν.

Αρκεί να προσθέσουμε το παρακάτω σενάριο στο αντικείμενο δράκος:



Οι μεταβλητές, επομένως, μπορούν να χρησιμοποιηθούν και ως ένας τρόπος έμμεσης επικοινωνίας μεταξύ των αντικείμενων όταν είναι ορατές σε αυτά. Πολλαπλά αντικείμενα μπορούν να αλλάζουν τις τιμές μιας μεταβλητής, ενώ όλα τα αντικείμενα μπορεί να περιέχουν στα σενάρια τους συνθήκες που εξαρτώνται από αυτήν.

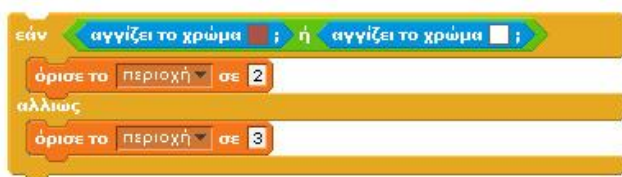
Παράδειγμα: Μπάσκετ: Έστω ότι θέλουμε να φτιάξουμε ένα πρόγραμμα με το οποίο να παίζουμε μπάσκετ. Στόχος του παιχνιδιού είναι να συγκεντρώσουμε όσους περισσότερους πόντους μπορούμε. Στο παιχνίδι αυτό λοιπόν θα προσομοιώσουμε ένα παίκτη που σουτάρει από διαφορετικές θέσεις στο γήπεδο! Για να το πετύχουμε αυτό, θα χρειαστούμε ένα αντικείμενο-χέρι για τον παίκτη που σουτάρει, ένα αντικείμενο-μπάλα και ένα αντικείμενο-καλάθι! Ο χρήστης μετακινεί το αντικείμενο-χέρι με τα βελάκια του πληκτρολογίου ενώ σουτάρει τη μπάλα πατώντας το πλήκτρο κενό για κατάλληλο χρονικό διάστημα. Όποτε η μπάλα αγγίζει τη στεφάνη του καλάθιού, θεωρούμε ότι ο παίκτης πέτυχε καλάθι. Κάθε φορά που ο παίκτης ρίχνει μία βολή και πετυχαίνει καλάθι από την περιοχή κάτω από το καλάθι, το σκορ του αυξάνεται κατά 2 πόντους. Κάθε φορά που πετυχαίνει καλάθι ενώ βρίσκεται έξω από την περιοχή, το σκορ του αυξάνεται κατά 3 πόντους. Στο τέλος του παιχνιδιού αυτό που θέλουμε να ξέρουμε είναι τους πόντους που έχει πετύχει ο παίκτης.



Είναι φανερό ότι κατά τη διάρκεια του παιχνιδιού πρέπει με κάποιο τρόπο να «κρατάμε» - αποθηκεύουμε τους πόντους του χρήστη και κάθε φορά που ο χρήστης πετυχαίνει καλάθι, να ενημερώνουμε το σκορ. Στο μυαλό μας έρχονται αμέσως οι μεταβλητές. Θα χρειαστούμε μία μεταβλητή στην οποία θα αποθηκεύουμε το σκορ του παίκτη κατά τη διάρκεια του παιχνιδιού. Πότε θα αλλάζουμε τη μεταβλητή «σκορ»; Όποτε το αντικείμενο «μπάλα» αγγίζει το αντικείμενο «στεφάνη». Σε ποιο αντικείμενο θα δημιουργήσουμε τη μεταβλητή; Σε ένα από τα δυο. Έστω ότι επιλέγουμε το πρώτο και ότι η μεταβλητή είναι ορατή μόνο στο συγκεκριμένο αντικείμενο.

Επιπλέον, θέλουμε να γνωρίζουμε από ποια περιοχή προήλθε η βολή που κατέληξε σε καλάθι έτσι ώστε να αυξήσουμε το σκορ κατά 2 ή κατά 3 πόντους.

Το πρόβλημά μας είναι πως το αντικείμενο-χέρι θα «μιλήσει» με το αντικείμενο-μπάλα ώστε να προσθέσει τον κατάλληλο αριθμό πόντων. Στο σημείο αυτό πρέπει να σκεφτούμε ότι μπορούμε να δημιουργήσουμε μία μεταβλητή που θα αποθηκεύει σε ποια περιοχή βρισκόταν το αντικείμενο «παίκτης» την ώρα της βολής. Έστω ότι την ονομάζουμε «περιοχή» και την ενημερώνει το αντικείμενο «παίκτης» ανάλογα με το που βρίσκεται την ώρα που επιχειρεί ο χρήστης το σουτ. Η μεταβλητή θα είναι ορατή από όλα τα αντικείμενα ώστε να τη λαμβάνει υπόψη του το αντικείμενο-μπάλα που μετράει το σκορ. Θα παίρνει τιμές με τον εξής απλό τρόπο:



Η περιοχή κάτω από τη στεφάνη έχει χρώμα μπορντό ή λευκό. Αν λοιπόν το αντικείμενο «παίκτης» αγγίζει αυτά τα χρώματα τότε ορίζουμε τη τιμή της μεταβλητής «περιοχή» σε 2. Διαφορετικά καταλαβαίνουμε ότι βρίσκεται έξω από τη περιοχή των δίποντων, οπότε και ορίζουμε τη μεταβλητή «περιοχή» σε 3 χρησιμοποιώντας την εντολή όρισε το...σε...

Η μεταβλητή «σκορ» ελέγχεται και τροποποιείται μόνο από το αντικείμενο «μπάλα». Αρχικά το σκορ είναι μηδέν αφού δεν έχει μπει κανένα καλάθι.



Έπειτα, κάθε φορά που το αντικείμενο μπάλα αγγίζει το αντικείμενο «στεφάνη», πρέπει το αντικείμενο «μπάλα» να ελέγχει

τη μεταβλητή «περιοχή». Αν η μεταβλητή αυτή έχει τιμή 2 τότε πρέπει να αυξήσει τη μεταβλητή «σκορ» κατά 2. Αυτό το πετυχαίνουμε με την εντολή άλλαξε σκορ κατά 2. Διαφορετικά, η μεταβλητή «περιοχή» θα έχει τιμή 3, καθώς μπορεί να πάρει μόνο αυτές τις δύο τιμές.



Μήπως όμως ο προγραμματιστής μας μπορούσε να κάνει κάτι πιο αποδοτικό; Μελετήστε προσεκτικά την παρακάτω εναλλακτική:

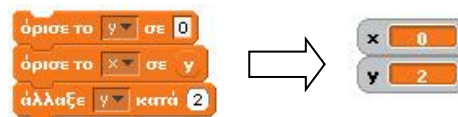


Αυτό ήταν ακόμη ένα παράδειγμα στο οποίο οι μεταβλητές χρησιμοποιήθηκαν για την έμμεση επικοινωνία μεταξύ αντικειμένων.

14.1.3 Γενικές παρατηρήσεις για τις μεταβλητές

Υπάρχουν δυο σημαντικά στοιχεία που πρέπει να αποσαφηνίσουμε για τις μεταβλητές στον προγραμματισμό.

A) Πρώτον, οι μεταβλητές που χρησιμοποιούμε στον προγραμματισμό έχουν διαφορετική έννοια από αυτές που χρησιμοποιούμε στα μαθηματικά. Στον προγραμματισμό, το περιεχόμενο μιας μεταβλητής αλλάζει σύμφωνα με τη σειρά εκτέλεσης των εντολών ενός έργου-προγράμματος. Αυτό σημαίνει ότι η τιμή που θα είναι αποθηκευμένη στη μεταβλητή θα είναι η τελευταία τιμή την οποία εκχωρήσαμε σε αυτή. Αν για παράδειγμα έχουμε τις μεταβλητές «x» και «y», και αρχικά αποθηκεύσουμε στη μεταβλητή «x» το περιεχόμενο της μεταβλητής «y» και στη συνέχεια αλλάξουμε την τιμή της μεταβλητής «y», αυτό δε σημαίνει ότι θα αλλάξει και η τιμή της μεταβλητής «x».



Στο πρόγραμμα της παραπάνω εικόνας οι μεταβλητές θα πάρουν τις εξής τιμές μετά από την εκτέλεση κάθε εντολής:

Όρισε το y σε μηδέν $y=0, x=\text{κενό}$

Όρισε το x σε y $y=0, x=0$

Αλλάξε y κατά 2 $y=2, x=0$

Δηλαδή το όρισε το x σε y ΔΕΝ σημαίνει εξίσωση των δυο τιμών αλλά ότι απλά θα βάλουμε ως τιμή του x, την τιμή του y τη στιγμή που εκτελείται η συγκεκριμένη εντολή. Αν στη συνέχεια η τιμή του y αλλάξει, η τιμή του x δε θα επηρεαστεί.

B) Δεύτερον, τα λευκά κουτάκια που περιέχουν οι εντολές των μεταβλητών, μπορούν να περιέχουν άλλες μεταβλητές.

Έτσι για παράδειγμα μπορούμε να αναθέσουμε μια σύνθετη μαθηματική έκφραση ως τιμή μιας μεταβλητής π.χ.



Σε αυτές τις αναθέσεις τιμών, υπολογίζεται ο τύπος που βρίσκεται στο δεύτερο πεδίο της εντολής και το νούμερο που θα βρεθεί γίνεται η τιμή της μεταβλητής. Είναι πολύ ενδιαφέρον ότι μπορούμε να χρησιμοποιήσουμε σε μια τέτοια έκφραση την ίδια τη μεταβλητή στην οποία αναθέτουμε την τιμή! Π.χ.:



Στο τέλος του προηγούμενου σεναρίου το x θα πάρει την τιμή 2. Το Scratch δε βλέπει τη συγκεκριμένη έκφραση ως λανθασμένη (παρότι φαίνεται παράξενη), καθώς υπολογίζει πρώτα την τιμή της μαθηματικής έκφρασης βάσει της προηγούμενης τιμής του x και στη συνέχεια αναθέτει την νέα υπολογισμένη τιμή, στο x. Φυσικά, η συγκεκριμένη έκφραση είναι περιττή καθώς μπορεί να αντικατασταθεί από την εντολή άλλαξε...κατά... Θα δούμε όμως αργότερα παραδείγματα που κάτι ανάλογο είναι απαραίτητο.

14.2 Λίστες

14.2.1 Τι είναι οι λίστες;

Πολλές φορές στην καθημερινή μας ζωή, χωρίς να το συνειδητοποιούμε, χρησιμοποιούμε λίστες. Τέτοια παραδείγματα είναι η λίστα του super market η οποία είναι ένας κατάλογος αντικειμένων που θέλουμε να αγοράσουμε. Ένα άλλο παράδειγμα είναι ένας τηλεφωνικός κατάλογος, ο οποίος περιλαμβάνει το όνομα και το τηλέφωνο των κατοίκων μιας περιοχής, όπως επίσης και η κατάσταση με τα ονόματα και τους βαθμούς που αναρτάται στο σχολείο μετά τις τελικές εξετάσεις της πληροφορικής. Η λίστα είναι μία συλλογή αντικειμένων με κοινά χαρακτηριστικά ή κοινό σκοπό. Για παράδειγμα, ο κοινός σκοπός των αντικειμένων της λίστας του supermarket είναι να αγοράσουμε όλα τα υλικά που χρειαζόμαστε για μία συνταγή μαγειρικής. Όμοια, το κοινό χαρακτηριστικό της λίστας με τα ονόματα και τις βαθμολογίες των μαθητών είναι ότι αφορούν το ίδιο μάθημα, δηλαδή αυτό της πληροφορικής.

Αντίστοιχα και στον προγραμματισμό συναντάμε συχνά περιπτώσεις που οι λίστες είναι χρήσιμες για την υλοποίηση ενός προγράμματος. Σκεφτείτε για παράδειγμα ότι θέλουμε να δημιουργήσουμε μια εφαρμογή αγώνων ταχύτητας (Formula 1) και θα πρέπει ο χρήστης να μπορεί να διαλέξει πιο όχημα θα οδηγήσει μέσα από μια λίστα από 20 διαφορετικά μονοθέσια! Αντίστοιχα, αναλογιστείτε πόσο μεγάλο ενδιαφέρον θα είχε για το χρήστη να διαλέγει ακόμα και τις πίστες στις οποίες θέλει να αγωνιστεί οι οποίες θα βρίσκονται αποθηκευμένες σε μια δεύτερη λίστα παρόμοια με αυτή των μονοθέσιων. Το σίγουρο είναι ότι με χρήση των λιστών μπορούμε να δημιουργήσουμε πάρα πολλά παιχνίδια από πολύ απλά μέχρι αρκετά πολύπλοκα, παιχνίδια που συναντούμε στην καθημερινότητά μας

και δεν φανταζόμασταν ποτέ ότι θα μπορούσαμε να γί-νουμε οι δημιουργοί τους!

14.2.2 Η λίστα στο Scratch

Πως θα μπορούσαμε όμως στο Scratch να δημιουργήσουμε μία λίστα; Παρατηρούμε ότι στην παλέτα «Μεταβλητές» υπάρχει η επιλογή «Δημιούργησε μια λίστα».



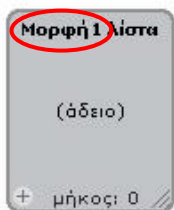
Πατώντας την επιλογή αυτή, ανοίγει ένα παράθυρο που έχει την ίδια μορφή με αυτό που εμφανίζεται όταν δημιουργούμε μία μεταβλητή.



Στο παράθυρο αυτό προσδιορίζουμε το όνομα που θέλουμε να δώσουμε στη λίστα και καθορίζουμε αν η λίστα αυτή θα είναι κοινή για όλα τα αντικείμενα ή μόνο για κάποιο, όπως ακριβώς και στις μεταβλητές. Η διαφορά των κοινών λιστών με τις λίστες που αφορούν ένα μόνο αντικείμενο μπορεί να φανεί καλύτερα μέσα ένα παράδειγμα. Σκεφτείτε για παράδειγμα μια εφαρμογή, όπως το γνωστό σε όλους μας παιχνίδι «Scrabble». Έστω ότι η εφαρμογή έχει δημιουργηθεί για τέσσερις χρήστες. Στο παιχνίδι υπάρχει μια λίστα που περιέχει όλα τα πιθανά γράμματα που μπορεί να τραβήξει κάθε παίχτης, η οποία πρέπει να είναι κοινή, δηλαδή έχει δημιουργηθεί με την επιλογή «Για όλες τις μορφές (κοινή)», αφού όλοι οι χρήστες πρέπει να μπορούν να επέμβουν πάνω της, δηλαδή να διαλέγουν γράμματα από αυτή. Ο κάθε παίχτης ξεχωριστά όμως πρέπει να κρατάει σε μια προσωπική του λίστα τα γράμματα που έχει επιλέξει σε κάθε γύρο. Επομένως είναι αναγκαία η δημιουργία τεσσάρων ακόμη λιστών οι οποίες θα δημιουργηθούν με την επιλογή «Μόνο γι' αυτή τη μορφή» αφού θα αφορούν κάθε χρήστη αποκλειστικά. Πατάμε «Εντάξει» στο προηγούμενο παράθυρο και η λίστα μας έχει δημιουργηθεί. Παρατηρήστε ότι εμφανίζεται ένα μικρό ορθογώνιο για τη λίστα στην οθόνη του Scratch, όπως φαίνεται στην παρακάτω εικόνα.



Αν η λίστα δημιουργήθηκε με το ίδιο όνομα αλλά με την επιλογή «Μόνο γι' αυτή τη μορφή» τότε εμφανίζεται και το όνομα του αντικείμενου στον τίτλο του παραθύρου, όπως φαίνεται στην επόμενη εικόνα.



Παρατηρούμε ότι αμέσως μετά τη δημιουργία της λίστας, στην παλέτα «Μεταβλητές» εμφανίζονται εντολές μέσω των οποίων μπορούμε να επεμβούμε στη λίστα μας (όπως συνέβη με τις μεταβλητές).



Ας συζητήσουμε τη χρήση των εντολών της λίστας μέσα από ρεαλιστικά παραδείγματα.

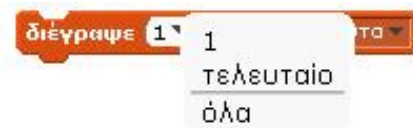
Η πρώτη εντολή που φαίνεται είναι η **πρόσθεσε [αντικείμενο] στο [λίστα]**. Με την εντολή αυτή, μπορούμε να προσθέσουμε ένα αντικείμενο στη λίστα μας, το οποίο προστίθεται στην τελευταία θέση της λίστας. Σκεφτείτε, για παράδειγμα, μια παραλλαγή της εφαρμογής «φιδάκι» όπου το φίδι μας δεν τρέφεται με τελίτσες αλλά με φρούτα! Έστω ότι θέλουμε να ξέρουμε κάθε στιγμή ποια φρούτα έχει ήδη φάει το φίδι κρατώντας τα σε μια λίστα. Τότε σε κάθε διαφορετικό φρούτο της εφαρμογής μας (θεωρώντας ότι έχουμε δημιουργήσει τόσα αντικείμενα όσα είναι και τα φρούτα που θέλουμε στο παιχνίδι), πρέπει να δημιουργήσουμε μια συνθήκη που αφορά το αν αγγίζει το φρούτο το αντικείμενο-φίδι. Αν συνθήκη γίνεται αληθής και τότε θέλουμε το όνομα του αντικείμενου να προστίθεται στη λίστα (την οποία ενδεικτικά μπορούμε να ονομάσουμε «Φρούτα»). Το παρακάτω σενάριο δείχνει την ομάδα των εντολών που προσθέτουν το όνομα «μήλο» στη λίστα «Φρούτα» αν αυτό αγγίξει το αντικείμενο «φίδι».



Μετά την εκτέλεση της εντολής **πρόσθεσε [Μήλο] στο [Φρούτα]** η λίστα των φρούτων θα έχει τη μορφή της επόμενης εικόνας.



Συνεχίζουμε με τη δεύτερη εντολή η οποία είναι η **διέγραψε [1] από το [λίστα]**. Με την εντολή αυτή μπορούμε να διαγράψουμε το πρώτο στοιχείο της λίστας μας. Παρατηρούμε όμως ότι στην επιλογή 1 υπάρχει ένα βελάκι το οποίο αν το πατήσουμε μας δίνει τρεις επιλογές σχετικά με το τι θέλουμε να διαγράψουμε από τη λίστα μας.



Με τη επιλογή «1» διαγράφουμε το πρώτο στοιχείο της λίστας μας, με την επιλογή «τελευταίο» όπως είναι φανερό διαγράφουμε το τελευταίο στοιχείο της, ενώ με την επιλογή «όλα» διαγράφουμε όλα τα αντικείμενά της. Μπορούμε επίσης κάνοντας κλικ στο «1» και όχι στο βελάκι να δώσουμε εμείς πιο αντικείμενο επιθυμούμε να διαγραφεί από τη λίστα π.χ. μπορούμε να εκτελέσουμε την εντολή **διέγραψε [4] από το [λίστα]**. Σκεφτείτε για παράδειγμα την προηγούμενη εφαρμογή με το φιδάκι που τρώει φρούτα. Έστω ότι ο πρώτος χρήστης έχασε όλες τις ζωές του και είναι η σειρά μας να παίξουμε. Αν εκκινήσουμε το πρόγραμμά μας τότε η λίστα «Φρούτα» θα περιέχει αρχικά όλα τα φρούτα που ο προηγούμενος χρήστης είχε φάει πριν χάσει. Αφού δε θέλουμε κάτι τέτοιο, είναι λογικό στην αρχή του προγράμματος να διαγράψουμε όλα τα περιεχόμενα της λίστας.



Γενικά είναι μια καλή στρατηγική κάθε φορά που χρησιμοποιούμε λίστες στην εφαρμογή μας να μην ξεχνάμε να διαγράφουμε όλα τα στοιχεία τους με το ξεκίνημα του προγράμματος. Η τρίτη κατά σειρά εντολή της λίστας είναι η **παρέμβαλε το [αντικείμενο] στη θέση [1] του [λίστα]**. Αυτή η εντολή μας δίνει τη δυνατότητα να προσθέσουμε ένα αντικείμενο στη λίστα μας, σε όποια θέση επιθυμούμε και όχι αναγκαστικά στη τελευταία θέση όπως κάνει η

εντολή πρόσθεσε [αντικείμενο] στο [λίστα]. Η επιλογή της θέσης στην οποία θα τοποθετηθεί το αντικείμενο γίνεται κατά τον ίδιο τρόπο με την επιλογή της θέσης του αντικειμένου που θέλουμε να διαγράψουμε, όπως εξηγήσαμε στη προηγούμενη εντολή. Η μόνη διαφορά είναι ότι εδώ στη θέση της επιλογής «όλα» που υπήρχε στη διαγραφή, έχουμε την επιλογή «οποιοδήποτε» με την οποία το αντικείμενο προστίθεται τυχαία σε κάποια θέση της λίστας.

παρέμβαλε το αντικείμενο στη θέση 1 1
τελευταίο
οποιοδήποτε

Έστω ότι θέλουμε να δημιουργήσουμε το πολύ γνωστό παιχνίδι κρεμάλα! Πως θα μας βοηθούσαν οι λίστες στην υλοποίησή τους; Μια λίστα θα μπορούσε να κρατούσε τα γράμματα της κρυμμένης λέξης, και μια δεύτερη λίστα θα μπορούσε να κρατάει τα γράμματα που δίνει ο χρήστης και είναι πετυχημένα στην προσπάθειά του να ανακαλύψει την κρυμμένη λέξη. Προφανώς, στη δεύτερη λίστα αρχικά θα εμφανίζονται το πρώτο και το τελευταίο γράμμα της λέξης. Τα δύο γράμματα θα εμφανίζονται στη πρώτη και στη τελευταία θέση της αντίστοιχης λίστας. Ανάμεσα στα δύο γράμματα πρέπει να παρεμβάλουμε τόσα κενά όσα είναι τα υπόλοιπα γράμματα της λέξης έτσι ώστε ο δεύτερος παίχτης να έχει μια αντίληψη της λέξης που ψάχνει. Ένα σενάριο που κάνει κάτι τέτοιο:

επανάλαβε μήκος του κρυφή - 2
παρέμβαλε το στη θέση 2 του φανερή

Για να μπορέσουμε να γεμίσουμε τη φανερή λίστα μας με τον κατάλληλο αριθμό κενών, χρησιμοποιούμε όπως βλέπουμε μια ακόμα εντολή των λιστών την μήκος του [λίστα]. Αυτή η εντολή μας δίνει τη δυνατότητα να γνωρίζουμε πόσα στοιχεία είναι αποθηκευμένα στη λίστα μας κάθε στιγμή. Στο συγκεκριμένο παράδειγμα θέλουμε να γεμίσουμε τη φανερή λίστα με τόσα κενά όσα αντιστοιχούν στο συνολικό μήκος της κρυφής λίστας μείον 2 για τα γράμματα που θα φαίνονται κανονικά. Με τον τρόπο αυτό πετυχαίνουμε τις επιθυμητές επαναλήψεις. Παρατηρήστε ότι παρεμβάλουμε τα κενά επαναλαμβανόμενα στη θέση 2 της φανεράς λίστας, δηλαδή ανάμεσα στο πρώτο και το τελευταίο γράμμα. Κάθε νέο κενό που προστίθεται σε κάθε επανάληψη μπαίνει στη δεύτερη θέση της λίστας «σπρώχνοντας» ουσιαστικά τα υπόλοιπα κενά και το τελευταίο γράμμα προς τα κάτω.

Η πέμπτη εντολή που συναντούμε στην παλέτα των μεταβλητών είναι η αντικατέστησε στοιχείο [1] του [λίστα] με [αντικείμενο]. Με την εντολή αυτή μπορούμε να αντικαταστήσουμε ένα στοιχείο της λίστας μας με ένα άλλο αντικείμενο. Η χρήση της είναι παρόμοια με την εντολή παρέμβαλε το [αντικείμενο] στη θέση [1] του [λίστα] που είδαμε πριν, με τη διαφορά ότι εδώ το νέο

αντικατέστησε στοιχείο 1 του ψ με αντικείμενο

μας αντικείμενο δεν μπαίνει στη λίστα μετακινώντας άλλα στοιχεία αλλά αντικαθιστά το αντικείμενο που βρισκόταν στη θέση που προσδιορίζουμε στην εντολή. Οι επιλογές που έχουμε για το

ποιο στοιχείο θέλουμε να αντικαταστήσει το καινούριο αντικείμενο είναι ίδιες με πριν. Μπορούμε δηλαδή να αντικαταστήσουμε το πρώτο αντικείμενο της λίστας, το τελευταίο της ή οποιοδήποτε άλλο. Με την επιλογή «οποιοδήποτε» η αντικατάσταση είναι τυχαία ενώ αν θέλουμε να αντικαταστήσουμε ένα συγκεκριμένο αντικείμενο αλλά όχι το πρώτο ή το τελευταίο, αρκεί να κάνουμε κλικ στον αριθμό 1 και να προσδιορίσουμε τον αριθμό της θέσης του στοιχείου προς αντικατάσταση.

Για να καταλάβουμε τη λειτουργία όμως αυτής της εντολής ας σκεφτούμε ξανά το προηγούμενο παράδειγμα της κρεμάλας. Έστω ότι ήρθε η ώρα του δεύτερου χρήστη να παίξει μαντεύοντας γράμματα της λέξης που έβαλε ο πρώτος. Αν το γράμμα που πληκτρολογεί ο παίχτης, ανήκει στη κρυφή λίστα, γεγονός που σημαίνει ότι είναι σωστή επιλογή, τότε πρέπει αυτό το γράμμα να αντικαταστήσει το κενό στην αντίστοιχη θέση στη φανερή λίστα. Για να ελέγξουμε αν το γράμμα που έδωσε ο χρήστης ανήκει στη κρυφή λίστα, θα χρησιμοποιήσουμε την εντολή των λιστών [λίστα] περιέχει [αντικείμενο]. Αυτή η εντολή πραγματοποιεί μια αναζήτηση στη λίστα μας και επιστρέφει «αληθές» αν το αντικείμενο υπάρχει ή «ψευδές» αν το αντικείμενο δεν υπάρχει σε αυτή.

ψ περιέχει αντικείμενο

Αυτό όμως από μόνο του δεν είναι αρκετό. Αν τελικά το γράμμα ανήκει στη κρυφή λίστα, πρέπει να ανακαλύψουμε σε ποια θέση ή θέσεις είναι ακριβώς, ώστε να αντικαταστήσουμε τα κενά με το γράμμα το οποίο πληκτρολογήθηκε. Αυτό δυστυχώς θα μας αναγκάσει να ελέγξουμε όλα τα στοιχεία της λίστας ένα προς ένα. Για να κάνουμε τον έλεγχο θα χρειαστούμε μια επανάληψη και τη χρήση της εντολής στοιχείο [1] του [λίστα]. Η εντολή αυτή επιστρέφει το όνομα του αντικειμένου της λίστας που βρίσκεται στη θέση που προσδιορίζουμε στην εντολή. Αν δηλαδή εκτελεστεί η εντολή στοιχείο [3] του [λίστα] τότε αυτή επιστρέφει το τρίτο αντικείμενο της λίστας. Το κομμάτι του σεναρίου της κρεμάλας που πραγματοποιεί αυτά που αναφέραμε:

εάν κρυφή περιέχει απάντηση
επανάλαβε μήκος του κρυφή - 2
εάν στοιχείο x του κρυφή = απάντηση
αντικατέστησε στοιχείο x του φανερή με απάντηση
άλλαξε x κατά 1

Η μεταβλητή απάντηση περιέχει την απάντηση του χρήστη σε ερώτηση για το επόμενο γράμμα με τη χρήση της εντολής ρώτησε...και περίμενε. Στο σενάριο αυτό εξετάζουμε αρχικά αν υπάρχει το γράμμα μέσα στη λίστα. Εφόσον υπάρχει, τότε με μια επανάληψη ελέγχουμε όλα τα γράμματα της κρυφής λίστας ξεχωριστά για να βρούμε τις θέσεις του γράμματος. Κάθε φορά που βρίσκουμε μια θέση στην κρυφή λίστα στην οποία υπάρχει το γράμμα, αντικαθιστούμε στην αντίστοιχη θέση το κενό της φανεράς λίστας!

Λείπει όμως κάτι από το σενάριο. Τι είναι το x ; Το x είναι μια μεταβλητή που έχουμε δημιουργήσει και της οποίας αρχική τιμή είναι το 2. Παρατηρήστε ότι λόγω αυτής της τιμής ο πρώτος έλεγχος στη κρυφή λίστα θα γίνει για το δεύτερο στοιχείο της. Η μεταβλητή x είναι ουσιαστικά ένας μεταβλητός δείκτης στη λίστα. Στην πρώτη επανάληψη παίρνει την τιμή 2, στην επόμενη 3 κ.ο.κ. Τέτοιες μεταβλητές, που αυξάνονται με τις επαναλήψεις ονομάζονται **μετρητές** και είναι ιδιαίτερα χρήσιμες στον προγραμματισμό.

Ο αριθμός των επαναλήψεων είναι ίσος είναι το μήκος της κρυφής λίστας μείον 2 αφού το πρώτο και το τελευταίο γράμμα δεν χρειάζεται να τα ελέγχουμε.

14.2.3 Μεταβλητές vs. Λίστες

Ίσως, να μην είναι εντελώς ξεκάθαρο πότε πρέπει να χρησιμοποιούμε μια μεταβλητή και πότε μια λίστα. Από τις παραπάνω περιγραφές βλέπουμε ότι τα δύο αυτά στοιχεία έχουν αρκετές ομοιότητες μεταξύ τους. Παρατηρούμε ότι και τα δύο αποτελούν δομές αποθήκευσης. Παρόλα αυτά, δεν πρέπει να συγχέουμε τη χρήση και το λόγο ύπαρξης του καθενός.

Αρχικά, πρέπει να έχουμε ξεκαθαρίσει στο μυαλό μας ότι οι μεταβλητές χρησιμοποιούνται για την απόδοση τιμής σε ένα χαρακτηριστικό ενός αντικειμένου, για παράδειγμα το «χρώμα_ματιών» του αντικειμένου «Κατερίνα». Ακόμη, οι μεταβλητές μπορούν να χρησιμοποιηθούν ως θέσεις προσωρινής αποθήκευσης για χαρακτηριστικά που η τιμή τους μπορεί να αλλάξει κατά την εκτέλεση ενός προγράμματος. Ένα τέτοιο παράδειγμα θα μπορούσε να είναι μεταβλητές που κρατούν το σκορ του αγώνα για τις ομάδες μας ή η μεταβλητή που κρατάει τις ελεύθερες θέσεις σε ένα parking για να δούμε αν υπάρχει χώρος για άλλα αυτοκίνητα.

Από την άλλη πλευρά, η λίστα μας προσφέρει τη δυνατότητα αποθήκευσης δεδομένων, χωρίς να απαιτείται να γνωρίζουμε εξ αρχής τον αριθμό των δεδομένων που θέλουμε να κρατήσουμε. Για παράδειγμα, σε έναν τηλεφωνικό κατάλογο δε γνωρίζουμε από την αρχή τον αριθμό των τηλεφώνων που θα καταχωρηθούν, οπότε για κάθε νέο αριθμό που θέλουμε να έχουμε στον κατάλογο μας, προσθέτουμε ακόμη μία θέση στη λίστα. Επίσης, μία ιδιότητα της λίστας που την καθιστά ιδιαίτερα χρήσιμη είναι ότι μπορεί να διαχειρίζεται μεγάλο πλήθος δεδομένων. Ας σκεφτούμε το προηγούμενο παράδειγμα, όπου ο τηλεφωνικός κατάλογος μπορεί να είναι κάποιας μεγάλης πόλης, για την οποία ο αριθμός τηλεφώνων των κατοίκων της είναι από την αρχή γνωστός. Αν θέλαμε να κρατήσουμε το κάθε τηλέφωνο σε μία ξεχωριστή μεταβλητή θα ήταν πολύ δύσκολο να δημιουργήσουμε και να διαχειριστούμε τόσες πολλές μεταβλητές. Επιπλέον, η λίστα μας προσφέρει τη δυνατότητα της ταξινόμησης των δεδομένων αλλά και τη δυνατότητα της αναζήτησης. Αυτές οι λειτουργίες θα ήταν χρήσιμες για παράδειγμα σε μία εφαρμογή για ένα κατάστημα ενοικίασης DVD ταινιών. Ο πωλητής θα μπορούσε να χειρίζεται το πρόγραμμα αυτό για να κάνει αναζήτηση στη λίστα των ταινιών που διαθέτει το κατά-

στημα έτσι ώστε να βρει αν υπάρχει η ταινία που ζήτησε κάποιος πελάτης. Ακόμη, θα έχει τη δυνατότητα να ταξινομή τη λίστα με τις ταινίες με αλφαβητική σειρά.

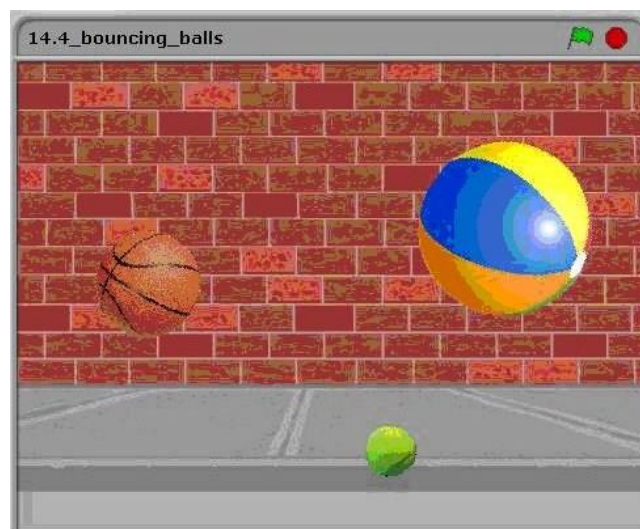
Ευτυχώς που σε αυτό το βιβλίο επιδιώκουμε την κατανόηση μέσα από παραδείγματα, και κυρίως παραδείγματα-παιχνίδια ☺

14.3 Παραδείγματα

14.3.1 Μεταβλητές

Παράδειγμα Bouncing Balls: Στη σκηνή υπάρχουν τρεις μπάλες και κάθε μία έχει διαφορετικό μέγεθος από τις άλλες. Όλες οι μπάλες χοροπηδούν ταυτόχρονα προς όλες τις κατευθύνσεις. Κάθε φορά που πετυχαίνεις μία μπάλα με το ποντίκι σου, το σκορ σου αυξάνεται. Όσο πιο μικρή η μπάλα, τόσο περισσότεροι και οι βαθμοί σου! Πόσο καλό στόχο έχεις;

Αντικείμενα: Αρχικά πρέπει να ανιχνεύσουμε ποια θα είναι τα αντικείμενά μας. Παρατηρούμε ότι πρέπει να υπάρχουν τρεις μπάλες οι οποίες θα κινούνται συνεχώς, επομένως και τρία αντικείμενα. Θα διαλέξουμε μπάλες με διαφορετικό μέγεθος. Μπορούμε να χρησιμοποιήσουμε μία μπάλα θαλάσσης ως τη μεγαλύτερη, μία μπάλα του μπάσκετ ως μεσαίου μεγέθους μπάλα και μία μπάλα του τένις ως τη μικρότερη μπάλα. Τα αντίστοιχα ονόματα των αντικειμένων θα είναι μπάλα_θαλάσσης, μπάλα_μπάσκετ και μπάλα_τένις. Όλες τις μπάλες μπορείτε να τις βρείτε μέσω διαδικτύου. Για σκηνικό, επιλέξαμε το υπόβαθρο «brick-wall1» της βιβλιοθήκης σκηνικών του Scratch για να φαίνεται ότι κυνηγάμε μπάλες στο δρόμο.



Ποιες είναι οι συμπεριφορές των αντικειμένων μας;

Μπάλα του μπάσκετ: Η μπάλα του μπάσκετ θα κινείται από την έναρξη του παιχνιδιού μέχρις ότου ο παίκτης να τερματίσει ο ίδιος το παιχνίδι. Όταν φτάσει στα όρια της σκηνής, θα πρέπει να συνεχίσει να κινείται αλλάζοντας κατεύθυνση.

Παρότι έχουμε αντιμετωπίσει πολλά παραπλήσια προβλήματα στα προηγούμενα κεφάλαια, ας θυμηθούμε ξανά όλες τις απαιτούμενες εντολές. Για την κίνηση της μπάλας θα χρησιμοποιήσουμε την εντολή **κινήσου...βήματα**. Εφόσον θέλουμε να μη σταματάει να κινείται, θα πρέπει η εντολή αυτή να συνδυαστεί με την εντολή **για πάντα**. Επίσης, για να αναπηδήσει και να αλλάξει κατεύθυνση στα όρια θα πρέπει να χρησιμοποιήσουμε

την εντολή εάν στα όρια, αναπήδησε. Για να γίνει το παιχνίδι ενδιαφέρον και η θέση της μπάλας να μην είναι απολύτως προβλέψιμη, μπορούμε να χρησιμοποιήσουμε τον τελεστή τυχαία επιλογή από...μέχρι... για να προσδώσουμε μια τυχαία αρχική κατεύθυνση στο αντικείμενο και στη συνέχεια με στόχο τα βήματα της μπάλας να μην είναι σταθερά σε κάθε επανάληψη. Το σενάριο της κίνησης που προκύπτει για την μπάλα:



Μπάλα του τένις και μπάλα θαλάσσης: Οι μπάλες αυτές θα πρέπει να κινούνται πάνω στη σκηνή με τον ίδιο τρόπο που κινείται και η μπάλα του μπάσκετ ενώ όταν ο παίκτης πετύχει με το ποιντίκι του κάποια από τις μπάλες αυτές, θα πρέπει να αυξάνεται το σκορ του με τη διαφορά όμως ότι για την μπάλα του τένις οι πόντοι του θα είναι περισσότεροι αφού είναι και η πιο μικρή.



Μη ξεχνάτε ότι για να κάνετε τις μπάλες να ξεκινήσουν από διαφορετικές κατευθύνσεις μπορείτε επίσης να στρίψετε την μπλε γραμμή κατεύθυνσης που βρίσκεται πάνω στο κάθε αντικείμενο όπως φαίνεται στην παρακάτω εικόνα.



Πόντοι: Έχοντας ολοκληρώσει την κίνηση κάθε μπάλας, ήρθε η στιγμή να υλοποιήσουμε τον τρόπο με τον οποίο ο παίκτης θα συγκεντρώνει πόντους. Οι πόντοι θα πρέπει να αποθηκεύονται σε μία μεταβλητή. Συνεπώς δημιουργούμε μια νέα μεταβλητή με το όνομα «πόντοι» και επιλέγουμε να είναι «Κοινή για όλες τις μορφές» αφού θα πρέπει να μπορεί να τροποποιηθεί από όλες τις μπάλες, δηλαδή από όλα τα αντικείμενα του έργου μας.

Κάθε φορά που ξεκινάει το πρόγραμμα η μεταβλητή πρέπει να έχει μηδενική τιμή για να είναι αξιόπιστο το σκορ του παίκτη! Άρα στα σενάρια για τις μπάλες προσθέτουμε την εντολή όρισε το [score] σε [0], πριν τις εντολές της κίνησης που είδαμε παραπάνω.



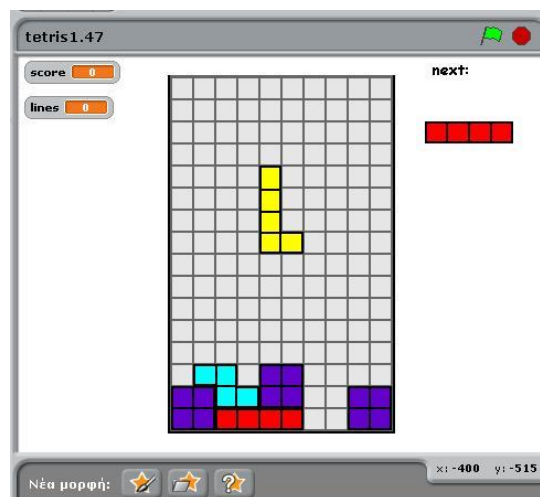
Κάθε φορά που γίνει κλικ πάνω σε μία μπάλα, οι πόντοι που συλλέγει ο παίκτης πρέπει να αυξάνονται. Αν πετύχει την μπάλα θαλάσσης κερδίζει 1 πόντο, την μπάλα του μπάσκετ 2 και την μπάλα του τένις 4 πόντους. Επομένως, πρέπει να χρησιμοποιήσουμε την εντολή όταν στο...γίνει κλικ για κάθε αντικείμενο, με στόχο να αλλάζουμε την τιμή της μεταβλητής ανάλογα με το ποια μπάλα έχει πετύχει ο παίκτης.

Για την αύξηση της τιμής της μεταβλητής θα χρειαστούμε την εντολή άλλαξε [πόντοι] κατά [...], όπου ο αριθμός αύξησης εξαρτάται από το είδος της μπάλας (1 για την μπάλα θαλάσσης, 2 για την μπάλα του μπάσκετ και 4 για την μπάλα του τένις). Προσθέτουμε τα παρακάτω σενάρια στα αντικείμενα μας (ένα σε κάθε αντικείμενο) και το παιχνίδι μας έχει ολοκληρωθεί!!



[14_π05.sb]

Παράδειγμα μεταβλητών στο Tetris: Για όσους δε γνωρίζουν το Tetris, σκοπός του παιχνιδιού είναι να δημιουργούμε ολοκληρωμένες γραμμές χωρίς κενά, τοποθετώντας κατάλληλα τα τουβλάκια που μας έρχονται από το πάνω μέρος της οθόνης. Κάθε φορά που ολοκληρώνεται μια γραμμή, τότε διαγράφεται και όλα τα τουβλάκια μετακινούνται προς τα κάτω. Στόχος του παίκτη είναι να μην φτάσουν τα στοιβαγμένα τουβλάκια στο πάνω μέρος της οθόνης. Εδώ δεν θα αναλύσουμε την λειτουργία ολόκληρου του παιχνιδιού αλλά θα εστιάσουμε στον τρόπο με τον οποίο επιλέγουμε με τυχαίο τρόπο τα τουβλάκια που πέφτουν.



Από τη στιγμή που υπάρχουν 7 διαφορετικά τουβλάκια διαθέσιμα, προφανώς θα χρησιμοποιήσουμε την εντολή τυχαία επιλογή από 1 μέχρι 7 ώστε να προκύψει ένας τυχαίος αριθμός από 1 έως 7. Ανάλογα με αυτό τον αριθμό θα στείλουμε μήνυμα στο κατάλληλο αντικείμενο ώστε να πάρει θέση και να αρχίσει να πέφτει μόλις έρθει η σειρά του. Από τη στιγμή που κάθε φορά που εκτελείται η εντολή τυχαία επιλογή από 1 μέχρι 7 παράγει διαφορετικό αριθμό, για να κάνουμε ελέγχους με τον αριθμό που παράγει, πρέπει να αποθηκεύσουμε το αποτέλεσμα της σε μια μεταβλητή. Μη ξεχνάτε ότι κάθε φορά που τρέχει η τυχαία

επιλογή από 1 μέχρι 7 παράγει έναν διαφορετικό αριθμό. Έπειτα, η συγκεκριμένη μεταβλητή θα συσχετιστεί στους διάφορους ελέγχους που πρέπει να γίνουν. Μελετήστε τον παρακάτω μέρος του σεναρίου του βασικού αντικειμένου του Tetris:



Όλο το σενάριο βρίσκεται μέσα σε μια επανάληψη. Στη τελευταία γραμμή του σεναρίου δημιουργείται ένας τυχαίος αριθμός ο οποίος αποθηκεύεται στη μεταβλητή next, ενώ στην πρώτη γραμμή του σεναρίου, η μεταβλητή randomnum παίρνει την τιμή της μεταβλητής next και με βάση αυτή τη μεταβλητή στέλνεται το κατάλληλο μήνυμα. Δεν θα μπορούσαν να γίνουν οι έλεγχοι αν δεν αποθηκεύαμε τον τυχαίο αριθμό που παράχθηκε μέσα σε μια μεταβλητή και για αυτό η χρήση των μεταβλητών για την αποθήκευση τυχαίων αριθμών είναι πολύ συχνή.

Γιατί ο προγραμματιστής επέλεξε να φωλιάσει τις εντολές εάν...αλλιώς και δεν τις έβαλε απλώς τη μια κάτω από την άλλη; Αξιολογήστε την απόδοση των δυο αντίστοιχων προγραμμάτων. Πόσοι έλεγχοι πραγματοποιούνται κάθε φορά;

Αντίστοιχα, αν θέλαμε να μεταφέρουμε ένα αντικείμενό μας μέσα σε ένα συγκεκριμένο τμήμα της οθόνης του Scratch με τυχαίο όμως τρόπο, θα δημιουργούσαμε ένα σενάριο σαν αυτό που φαίνεται στην επόμενη εικόνα.



Στο παραπάνω σενάριο, τα άλλα αντικείμενα μπορούν να ελέγξουν τη νέα θέση του συγκεκριμένου αντικειμένου μέσω των μεταβλητών x και y.

Αν κουρασθήκατε και θέλετε να παίξετε Tetris, μπορείτε να επισκεφτείτε τη διεύθυνση

<http://scratch.mit.edu/projects/amyv/11292>

14.3.2 Λίστες

Παράδειγμα «Κρεμάλα»: Τι θα λέγατε να παίξουμε μία κρεμάλα; Ας επαναλάβουμε τους κανόνες για να είμαστε σίγουροι ότι το παιχνίδι θα είναι δίκαιο! Στο παιχνίδι αυτό χρειαζόμαστε δύο παίκτες τουλάχιστον. Ο πρώτος παίκτης είναι αυτός που διαλέγει μια λέξη και φανερώνει μόνο το πρώτο και το τελευταίο γράμμα της καθώς και πόσες κενές θέσεις υπάρχουν ανάμεσά τους. Ο δεύτερος παίκτης ψάχνει να βρει τη λέξη προτείνοντας κάθε φορά κάποιο γράμμα. Όμως οι ευκαιρίες του δεν είναι απεριόριστες! Έτσι όταν τις εξαντλήσει, χάνει! Στην παρακάτω εικόνα, βλέπετε ένα στιγμιότυπο του προγράμματος, που ο πρώτος χρήστης έχει βάλει τη λέξη «γιάννης» και ο δεύτερος χρήστης έχει έως τώρα δοκιμάσει τα γράμματα ι,ο και επιτρέπεται να κάνει άλλα 6 λάθη.



[14_p05.sb]

Αντικείμενα και Σκηνικό: Στο συγκεκριμένο παιχνίδι παρατηρούμε ότι δε χρειαζόμαστε παραπάνω από ένα αντικείμενο. Το αντικείμενο μας θα λειτουργεί ως παρουσιαστής του παιχνιδιού που ανακοινώνει στους παίκτες τους κανόνες. Στο παράδειγμά μας, θα αφήσουμε τον κ.Γατίδη ως παρουσιαστή.

Υλοποίηση του παιχνιδιού: Ας αναλύσουμε το παιχνίδι λίγο πιο προσεκτικά. Τι συμβαίνει στο παιχνίδι;

- ✓ Πραγματοποιείται η εισαγωγή μίας λέξης από τον πρώτο παίκτη.

- ✓ Αποκρύπτεται η λέξη αυτή ώστε ο δεύτερος παίκτης να μην τη γνωρίζει.
- ✓ Εμφανίζεται μια λίστα που περιέχει μόνο το αρχικό και τελευταίο γράμμα της λέξης με ενδιάμεσα κενά για τα γράμματα που λείπουν.
- ✓ Ο δεύτερος παίκτης αρχίζει να εισάγει γράμματα.
- ✓ Αν τα γράμματα αυτά υπάρχουν στη λέξη, προστίθενται στις σωστές θέσεις.
- ✓ Όλα τα γράμματα που εισάγει ο χρήστης αποθηκεύονται σε μια άλλη λίστα ώστε να γνωρίζει ποια έχει δοκιμάσει κάθε στιγμή.
- ✓ Το παιχνίδι συνεχίζεται είτε μέχρι είτε να βρει τη λέξη ο χρήστης, είτε να φτάσει το μέγιστο αριθμό λαθών που επιτρέπονται.

Όπως παρατηρείτε, το συγκεκριμένο έργο έχει μια ακολουθιακή δομή, στην οποία η μια ενέργεια διαδέχεται την επόμενη και συνεπώς δε θα χρειαστούμε πολλαπλά σενάρια αλλά μόνο 1! Μετά από πολλά κεφάλαια θα δημιουργήσουμε ένα παιχνίδι με ένα μόνο σενάριο!

Εισαγωγή λέξης από τον πρώτο παίκτη.

Επειδή όλο το παιχνίδι εκτυλίσσεται σε επίπεδο γραμμάτων, θα ζητήσουμε από το χρήστη να εισάγει τη λέξη προσδιορίζοντας τα γράμματά της ξεχωριστά. Ο μόνος τρόπος που έχουμε μάθει για την εισαγωγή δεδομένων από το χρήστη είναι μέσω της εντολής **ρώτησε...και περίμενε** και της μεταβλητής **απάντηση** από την παλέτα **Αισθητήρες**. Που όμως θα αποθηκεύσουμε τα γράμματα; Χρειαζόμαστε τη δημιουργία μιας λίστας την οποία θα ονομάσουμε «κρυφή» και στην οποία θα αποθηκεύουμε τα γράμματα της λέξης. Αυτή η λίστα δεν θα πρέπει να είναι ορατή στο χρήστη.

Αρχικά, κατανοούμε ότι εφόσον ο παίκτης θα δίνει τα γράμματα ξεχωριστά, θα χρειαστούμε κάποια εντολή επανάληψης έτσι ώστε να επαναλαμβάνεται η διαδικασία εισαγωγής γραμμάτων. Μέχρι τότε θα εκτελείτε η επανάληψη; Κάθε λέξη έχει διαφορετικό αριθμό γραμμάτων και συνεπώς δε μπορούμε να γνωρίζουμε εκ των προτέρων τον ακριβή αριθμό των επαναλήψεων. Μπορούμε όμως να δημιουργήσουμε εμείς μια συνθήκη τερματισμού, π.χ. να πούμε ότι όταν η λέξη ολοκληρωθεί, ο χρήστης μπορεί να εισάγει ως γράμμα το κενό, οπότε το πρόγραμμα αντιλαμβάνεται ότι τα γράμματα της λέξης τελείωσαν. Άρα από τη στιγμή που δεν γνωρίζουμε τον αριθμό των επαναλήψεων πρέπει να χρησιμοποιήσουμε την εντολή **επανάλαβε...** από την παλέτα **Έλεγχος**. Στο εσωτερικό της επανάληψης αυτής θα χρειαστούμε την εντολή **ρώτησε...και περίμενε** και τη μεταβλητή **απάντηση** και κάθε γράμμα που προσδιορίζει ο χρήστης θα το εισάγουμε στην κρυφή λίστα μας μέσω της εντολής **πρόσθεσε [...] στο [κρυφή]**. Το παρακάτω σενάριο υλοποιεί αυτή τη διαδικασία:



Δυο παρατηρήσεις σχετικές με το προηγούμενο σενάριο:

- ✓ Με βάση το προηγούμενο σενάριο, όταν ο παίκτης πατήσει το κενό, αυτό θα προστεθεί στη λίστα, κάτι που θέλουμε να αποφύγουμε αφού δεν αποτελεί μέρος της λέξης. Για αυτό αφαιρούμε από τη λίστα το τελευταίο στοιχείο που προστέθηκε με την εντολή **διέγραψε [τελευταίο] από το [κρυφή]**.
- ✓ Η συνθήκη της επανάληψης παίρνει τη μορφή «**[απάντηση] = []**» με τη βοήθεια του τελεστή της ισότητας από την παλέτα Τελεστές. Το λευκό κουτάκι στον τελεστή φαίνεται άδαιο αλλά εμείς έχουμε πατήσει το χαρακτήρα του κενού.

Από τη στιγμή που δεν υπάρχουν εντολές για την απόκρυψη και την εμφάνιση μιας λίστας στην οθόνη του Scratch, η απόκρυψη της κρυφής λίστας πρέπει να γίνει απενεργοποιώντας την αντίστοιχη μεταβλητή κατάστασης «κρυφή» που βρίσκεται στην παλέτα **Μεταβλητές**.

Δημιουργία λέξης με κενά

Ήρθε η στιγμή να προσπαθήσουμε να δημιουργήσουμε τη λέξη που θα εμφανίζεται στον δεύτερο παίκτη και η οποία θα έχει ένα γράμμα στην αρχή και ένα στο τέλος της και κενά στις ενδιάμεσες θέσεις. Σκεφτόμενοι όπως και πριν, και αυτή η λέξη θα αποθηκευτεί σε μορφή λίστας έτσι ώστε να είμαστε σε θέση να αντικαταστήσουμε τα κενά της με τις επιλογές γραμμάτων του παίκτη. Θα δημιουργήσουμε λοιπόν μία λίστα με όνομα «φανερή». Για την εισαγωγή στη λίστα «φανερή» του πρώτου και του τελευταίου στοιχείου της λίστας «κρυφή», θα χρειαστούμε τις εντολές **πρόσθεσε [...] στο [φανερή]**, **στοιχείο [τελευταίο] του [κρυφή]** και **στοιχείο [1] του [κρυφή]**. Οι δυο πρώτες εντολές του παρακάτω σεναρίου (που αποτελούν τη συνέχεια του προηγούμενου) προσθέτουν το πρώτο και τελευταίο γράμμα της κρυφής λίστας στη φανερή.



Τα στοιχεία αυτά θα εισαχθούν στην πρώτη και δεύτερη θέση της «φανερής» λίστας, ενώ δεν υπάρχουν προς το παρόν κενά στοιχεία για τα γράμματα που λείπουν. Πρέπει όμως να εισάγουμε αυτά τα κενά. Αυτό κάνει η επανάληψη που ακολουθεί

στο προηγούμενο σενάριο. Βάζει όσα κενά χρειαζόμαστε μεταξύ των δυο γραμμάτων. Πόσα κενά χρειαζόμαστε; Όσο είναι το πλήθος των γραμμάτων της λίστας «κρυφή» μείον 2, αφού έχουμε ήδη εισάγει δύο από τα γράμματα της λέξης. Το πλήθος των γραμμάτων της κρυφής λίστας θα το πάρουμε με την εντολή μήκος του [κρυφή], ενώ η αφαίρεση θα πραγματοποιηθεί με τη βοήθεια του τελεστή αφαίρεσης ...- Για την εισαγωγή των κενών χρησιμοποιήσαμε την εντολή παρέμβαλε το [] στη θέση [2] του [φανερή]. Με την εντολή αυτή ουσιαστικά παρεμβάλλουμε κάθε φορά στη θέση 2 της φανερής λίστας ένα κενό μετατοπίζοντας τα στοιχεία που ακολουθούν προς τα κάτω και δημιουργώντας έτσι τα απαραίτητα κενά. Η λίστα «φανερή» θέλουμε να εμφανίζεται στον παίκτη, οπότε πρέπει να επιλέξουμε την εμφάνιση της από την αντίστοιχη μεταβλητή κατάσταση.

Εισαγωγή γραμμάτων από το δεύτερο παίκτη

Η συνέχεια του προγράμματός μας είναι μια επανάληψη. Ο δεύτερος παίκτης πρέπει να μπορεί να δίνει διαρκώς γράμματα και εμείς να εξετάζουμε τις επόμενες εναλλακτικές:

- ✓ το γράμμα να ανήκει μέσα στην κρυφή λίστα, οπότε πρέπει να το προσθέσουμε στη φανερή στις κατάλληλες θέσεις,
- ✓ το γράμμα να μην ανήκει μέσα στη κρυφή λίστα, οπότε πρέπει να σημειώσουμε ότι ο παίκτης έκανε ένα λάθος,
- ✓ η φανερή λίστα να μην περιέχει άλλα κενά, γεγονός που σημαίνει ότι ο δεύτερος παίκτης τα κατάφερε και βρήκε τη λέξη,
- ✓ ο χρήστης να φτάνει το όριο των λαθών που του επιτρέπονταν και συνεπώς το παιχνίδι να τερματίζεται.

Ας δούμε αυτές τις εναλλακτικές ξεχωριστά:

A) Το γράμμα ανήκει μέσα στην κρυφή λίστα, οπότε πρέπει να το προσθέσουμε και στη φανερή λίστα

Εφόσον, πλέον, ο δεύτερος παίκτης μπορεί να δει τη λέξη με την οποία θα παίξει, πρέπει να μπορεί να εισάγει τα γράμματα που φαντάζεται ότι κρύβονται πίσω από τα κενά. Για να εισάγει ο δεύτερος παίκτης τα γράμματα θα χρησιμοποιήσουμε ξανά την εντολή ρώτησε [] και περίμενε απάντηση.

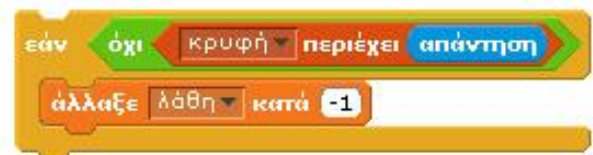
Όταν ο παίκτης εισάγει ένα γράμμα, πρέπει να ελέγχεται αν υπάρχει μέσα στα γράμματα της κρυφής λίστας. Ο έλεγχος αυτός θα γίνει με τη βοήθεια της εντολής [κρυφή] περιέχει [απάντηση]. Εάν ο έλεγχος αυτός είναι αληθής, δηλαδή αν το γράμμα περιέχεται στη λέξη, τότε θα πρέπει να βρούμε σε ποια θέση της λέξης βρίσκεται. Για να βρούμε τη θέση/τις θέσεις, θα χρειαστούμε μια επανάληψη όπως αυτή που αναλύσαμε προηγουμένως στο κεφάλαιο. Στην επανάληψη θα εξετάσουμε αν οποιοδήποτε από τα ενδιάμεσα γράμματα της κρυφής λίστας ταιριάζει με την απάντηση του χρήστη. Αν ταιριάζει, θα αντικαταστήσουμε το κενό της φανερής λίστας, με την απάντηση του χρήστη, στη θέση που εντοπίσαμε το γράμμα στην κρυφή λίστα! Μελετήστε το παρακάτω σενάριο:



Θυμηθείτε ότι το x είναι ένας μια μεταβλητή-μετρητής για να ελέγχουμε διαφορετικές θέσεις στις λίστες και για αυτό πρέπει σε κάθε επανάληψη την αυξάνουμε.

B) Το γράμμα δεν ανήκει μέσα στη κρυφή λίστα, οπότε πρέπει να σημειώσουμε ότι ο παίκτης έκανε ένα λάθος

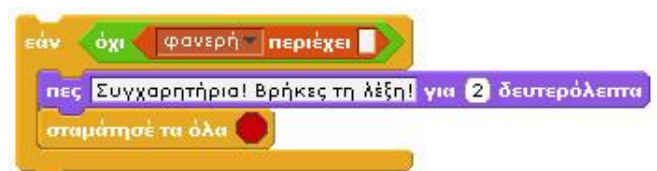
Τι συμβαίνει όμως αν το γράμμα δεν ανήκει μέσα στη κρυφή λίστα; Τότε ο δεύτερος παίκτης έχει κάνει ένα λάθος. Αλλά πόσα λάθη θα επιτρέψουμε στον παίκτη να κάνει; Έστω ότι δίνουμε στον παίκτη επτά ευκαιρίες. Αυτός ο αριθμός θα πρέπει να μειώνεται κάθε φορά που ο παίκτης κάνει λάθος και όταν μηδενιστεί, τότε να τερματίζεται και το παιχνίδι. Για να μπορούν όμως να γίνονται οι ενέργειες αυτές πρέπει ο αριθμός των λαθών να αποθηκεύεται σε μία μεταβλητή. Δημιουργήστε, λοιπόν, τη μεταβλητή με όνομα «λάθη» και στη συνέχεια προσθέστε στο προηγούμενο σενάριο μας, τις παρακάτω εντολές:



Οι εντολές αυτές απλά εξετάζουν αν η κρυφή λίστα δεν περιέχει το γράμμα που έδωσε ο χρήστης, και εφόσον κάτι τέτοιο ισχύει, μειώνουν τα διαθέσιμα λάθη του χρήστη.

Γ) Η φανερή λίστα δεν περιέχει άλλα κενά, γεγονός που σημαίνει ότι ο δεύτερος παίκτης τα κατάφερε και βρήκε τη λέξη.

Στην περίπτωση που ο χρήστης έχει βρει τη λέξη, η φανερή λίστα δεν θα περιέχει κενά, γεγονός που πρέπει να οδηγήσει στον τερματισμό του παιχνιδιού. Ο έλεγχος θα γίνει και σε αυτή την περίπτωση με τη βοήθεια του λογικού τελεστή άρνησης όχι... και της εντολής συνθήκης [φανερή] περιέχει []. Όταν η συνθήκη αυτή γίνει αληθής, τότε θα πρέπει ο παρουσιαστής να δηλώνει στον παίκτη ότι κέρδισε και να τερματίζεται το παιχνίδι. Όλες αυτές οι εντολές εμφανίζονται στην παρακάτω εικόνα.



Δ) Ο χρήστης φτάνει το όριο των λαθών που τέθηκε από την αρχή

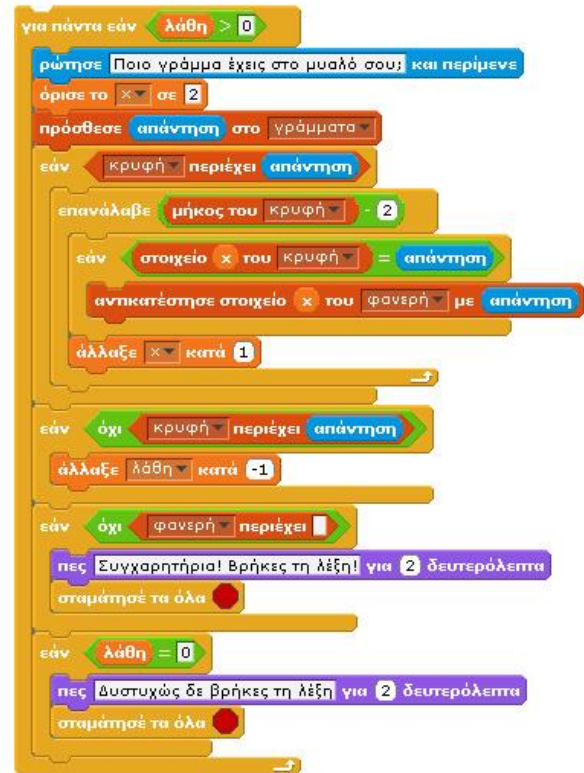
Στην περίπτωση, που ο παίκτης έχει εξαντλήσει όλες του τις ευκαιρίες, δηλαδή όταν η μεταβλητή «λάθη» έχει γίνει 0, το παιχνίδι πρέπει να τερματίζεται και να ανακοινώνεται στον παίκτη ότι δε βρήκε τη λέξη. Ο έλεγχος σε αυτή την περίπτωση είναι πιο απλός και υλοποιείται με τη βοήθεια του τελεστή ισότητας [λάθη] = [0].



Το μόνο που δεν έχουμε κάνει μέχρι αυτή τη στιγμή είναι να μπορεί ο χρήστης να βλέπει ποια γράμματα έχει εισάγει. Για να το καταφέρουμε και αυτό, θα δημιουργήσουμε μια νέα λίστα με όνομα «γράμματα», θα την εμφανίσουμε στην οθόνη του Scratch και θα προσθέτουμε σε αυτήν όλες τις απαντήσεις του δεύτερου παίκτη με τη χρήση της εντολής πρόσθεσε «απάντηση» στα «γράμματα».

Επίσης δεν πρέπει να ξεχνάμε, ότι για να ξαναπαίξουμε το παιχνίδι θα πρέπει οι λίστες να είναι άδειες και για το λόγο αυτό, στην αρχή του σεναρίου μας πρέπει να διαγράψουμε όλα τα περιεχόμενά τους με την εντολή διέγραψε [όλα] από το [φανερή/κρυφή/γράμματα].

Μελετήστε προσεκτικά το ολοκληρωμένο σενάριο.



Και φυσικά μια διαπίστωση για τους αποδοτικούς προγραμματιστές μας: υπάρχει μια εύκολη αλλαγή που μπορούμε να κάνουμε και η οποία θα κάνει το πρόγραμμά μας λίγο γρηγορότερο από όσο είναι τώρα. Ποια είναι αυτή; Αναζητήστε την απάντηση στις εντολές επανάληψης του σεναρίου.

Παράδειγμα «Love chase»:

Τι θα λέγατε για ένα κυνηγητό; Όταν ακούμε κυνηγητό μας έρχονται στο μυαλό διάφορες ιστορίες όπως ένα περιπολικό να κυνηγάει το αυτοκίνητο των ληστών ή ακόμη και το παιχνίδι όπου ένας παίκτης καλείται να πιάσει όλους τους υπόλοιπους. Εδώ θα δημιουργήσουμε ένα διαφορετικό είδος κυνηγητού. Πόσοι από εσάς δεν έχετε φανταστεί να σας κυνηγάει το αγόρι ή το κορίτσι των ονείρων σας; Αυτό όμως μερικές φορές μπορεί να καταλήξει ενοχλητικό! Στη δική μας ιστορία ο κύριος της «παρέας» δεν αντέχει άλλο το στενό μαρκάρισμα της θαυμάστριάς του και τρέχει να γλιτώσει! Εκείνη όμως τον ακολουθεί κατά βήμα, αφού μπορεί και καταγράφει κάθε του κίνηση! Δημιουργήστε δύο φιγούρες της επιλογής σας, έτσι ώστε η μία να κινείται από τον παίκτη και η δεύτερη να ακολουθεί την πορεία της πρώτης. Η επόμενη εικόνα είναι αποκαλυπτική για το τι πρέπει να φαίνεται στο έργο σας! Έχει πολύ πλάκα, δοκιμάστε το.

[14_π06.sb]



Αντικείμενα και Σκηνικό: Στο κунηγητό που θα δημιουργήσουμε συμμετέχουν δύο αντικείμενα. Το ένα αντικείμενο θα μοιάζει με αγόρι ενώ το άλλο με κορίτσι. Μπορείτε να επιλέξετε ό, τι θέλετε. Στο συγκεκριμένο παράδειγμα έχουν χρησιμοποιηθεί οι μορφές fantasy10 και fantasy14 από τον κατάλογο Fantasy της βιβλιοθήκης μορφών του Scratch. Στη μορφή fantasy14 έχουν γίνει τροποποιήσεις στο χρώμα με τη βοήθεια της ζωγραφικής, ενώ και στις δύο μορφές έχει γίνει σμίκρυνση αφού το κунηγητό θέλει χώρο! Το σκηνικό επίσης δεν παίζει ιδιαίτερο ρόλο και για το λόγο αυτό χρησιμοποιούμε ένα απλό λευκό σκηνικό. Στα αντικείμενα μας έχουμε δώσει τα ονόματα Mr.Blue και Mrs.Pink. Ας δούμε τα δυο αντικείμενα ξεχωριστά.

Mr.Blue

Ας σκεφτούμε πιο αναλυτικά τι πρέπει να κάνει αρχικά το αντικείμενο Mr.Blue, ο ήρωας που τρέχει για να ξεφύγει από την κατά τα άλλα πολυαγαπημένη του. Ο χαρακτήρας αυτός ελέγχεται από το χρήστη και πρέπει να μετακινείται προς όλες τις κατευθύνσεις, χρησιμοποιώντας τα βελάκια του πληκτρολογίου. Όμως, εφόσον η Mrs.Pink πρέπει να τον ακολουθεί, σημαίνει ότι ο Mr.Blue αφήνει πίσω του ίχνη της κίνησής του, κάτι που μπορεί να γίνει με τη βοήθεια της πένα. Αλλά είναι αυτό από μόνο του αρκετό για να τον εντοπίσει; Μάλλον όχι. Για να μπορεί η Mrs.Pink να τον ακολουθεί κατά βήμα πρέπει να γνωρίζει τις ακριβείς συντεταγμένες της πορείας του κάθε στιγμή. Αυτό μπορεί να συμβεί μόνο αν ο Mr.Blue αποθηκεύει τις συντεταγμένες από τις οποίες έχει περάσει και η Mrs.Pink να έχει πρόσβαση σε αυτές. Μπορείτε να σκεφτείτε που θα καταγράφονται οι συντεταγμένες του Mr.Blue;

Mrs.Pink

Η Mrs.Pink, από την άλλη πλευρά, φροντίζει να ενημερώνεται για τις θέσεις από τις οποίες έχει περάσει ο Mr.Blue, διαβάζοντας τις καταχωρήσεις του και πηγαίνοντας προς αυτές ελπίζοντας ότι θα τον πετύχει. Όμως, ως φανατική θαυμάστρια που είναι, θέλει να σβήνει τα ίχνη του Mr.Blue έτσι ώστε καμία άλλη να μην μπορεί να τον εντοπίσει!

Κίνηση του Mr.Blue

Ας ξεκινήσουμε την υλοποίηση του παιχνιδιού δημιουργώντας την κίνηση του Mr.Blue από τον παίκτη. Για την κίνηση αυτή θα χρησιμοποιήσουμε τα βελάκια του πληκτρολογίου. Έχουμε δει πολλαπλά τέτοια παραδείγματα σε προηγούμενα παιχνίδια. Μελετήστε τα παρακάτω σενάρια και διαπιστώστε ποια είναι η διαφορά από τους τρόπους κίνησης που έχουμε δει μέχρι τώρα.



Σε αντίθεση με τα προηγούμενα παραδείγματα, στα οποία προσδιορίζαμε ένα συγκεκριμένο αριθμό βημάτων για την αλλαγή της θέσης του αντικείμενου, εδώ προσδιορίζουμε την αλλαγή μέσω μιας μεταβλητής, της μεταβλητής «βήμα». Μέσω αυτής της μεταβλητής μπορούμε να αλλάζουμε το βήμα με το οποίο κινείται το αντικείμενο προς όλες τις κατευθύνσεις αλλάζοντας μόνο την τιμή της μεταβλητής. Ανάλογα με την ταχύτητα που θέλουμε να κινείται ο ήρωάς μας, μπορούμε απλά να αλλάζουμε το βήμα στην αρχή του παιχνιδιού. Παρατηρήστε ότι οι αρνητικές που θέλαμε να δοθούν όταν αλλάζουν οι θέσεις x και y προς τα αριστερά και προς τα κάτω αντίστοιχα, δίνονται με τη βοήθεια του τελεστή πολλαπλασιασμού «...*...» από την παλέτα **Τελεστές**, πολλαπλασιάζοντας τη μεταβλητή «βήμα» με το -1.

Καταγραφή της κίνησης του Mr.Blue

Που θα καταγράφονται οι θέσεις του Mr.Blue; Από τη στιγμή που θέλουμε να έχουμε στη διάθεση μας όλες τις θέσεις από τις οποίες περνάει ο Mr.Blue, χρειαζόμαστε μια λίστα που να αποθηκεύει τις συντεταγμένες της μετακίνησής του. Μόνο μία λίστα όμως είναι αρκετή; Αν και οι δύο συντεταγμένες του Mr.Blue αποθηκεύονται σε μία θέση της λίστας θα μπορούσαμε να τις επεξεργαστούμε; Η απάντηση είναι πώς δεν μπορούμε να ξεχωρίσουμε αυτές τις τιμές όταν αποθηκεύονται στην ίδια θέση και για το λόγο αυτό θα πρέπει να δημιουργήσουμε δύο λίστες: στη μία θα αποθηκεύονται οι θέσεις x του Mr.Blue και στην άλλη οι θέσεις y, προσέχοντας όμως η θέση αποθήκευσης κάθε φορά στις δυο λίστες να είναι αντίστοιχη.

Ας ονομάσουμε τις λίστες αυτές «x» και «y» αντίστοιχα. Η Mrs.Pink μπορεί και ακολουθεί κατά βήμα τον Mr.Blue γιατί έχει πρόσβαση στις τιμές αυτές. Άρα, οι λίστες «x» και «y» πρέπει να είναι «Για όλες τις μορφές», δηλαδή κοινές.

Τώρα πρέπει να αποφασίσουμε πότε θα καταγράφονται οι τιμές των θέσεων στις οποίες βρίσκεται ο Mr.Blue. Οι θέσεις αυτές μπορούν να καταγράφονται κάθε φορά που ο παίκτης πατήσει κάποιο βελάκι. Άλλωστε, μόνο τότε αλλάζει η θέση του Mr.Blue και τότε πρέπει να εισάγουμε στις λίστες τις νέες συντεταγμένες του.

Ο στόχος μας θα πραγματοποιηθεί με τις εντολές πρόσθεσε [θέση x] στο [x] και πρόσθεσε [θέση y] στο [y] από την παλέτα Μεταβλητές. Οι προσθήκες πρέπει να γίνουν ακριβώς μετά τις εντολές μετακίνησης του αντικειμένου. Το προηγούμενο σενάριο έχει γίνει πλέον:



Όπως είπαμε και στην αρχική περιγραφή του Mr.Blue, ο χαρακτήρας θα πρέπει όταν μετακινείται να αφήνει πίσω του ίχνη. Για το λόγο αυτό θα χρησιμοποιήσουμε την πένα. Αρχικά, πρέπει να ορίσουμε το χρώμα και το μέγεθος της πέννας. Οι ιδιότητες αυτές καλό είναι να καθοριστούν κατά την έναρξη του προγράμματος. Έτσι, προσθέτουμε τις εντολές όρισε το χρώμα πέννας σε... και όρισε το μέγεθος πέννας σε [2] από την παλέτα Πένα. Επίσης, για να μπορέσει να αφήνει τα ίχνη του πρέπει το αντικείμενο να κατεβάσει την πένα με τη βοήθεια της εντολής κατέβασε πένα. Ανανεώνουμε για άλλη μια φορά το αρχικό μας σενάριο.

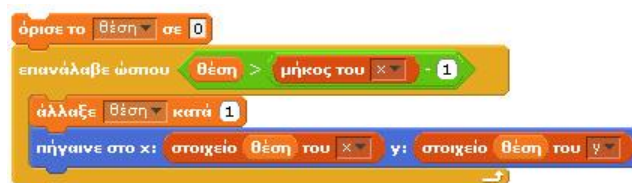
Τέλος, είναι αναγκαίο όταν ξεκινάει το παιχνίδι, οι λίστες με τις θέσεις από τις οποίες έχει περάσει ο Mr.Blue να είναι άδειες και να μην περιέχουν τιμές από προηγούμενο παιχνίδι. Για το λόγο αυτό, κατά την έναρξη, πρέπει να διαγράφονται όλα τα στοιχεία που πιθανόν να περιέχουν. Αυτό θα πραγματοποιηθεί με τις εντολές διέγραψε [όλα] από το [x] και διέγραψε [όλα] από το [y]. Επιπλέον, πρέπει να καθαρίζεται και η σκηνή από τυχόν υπολειπόμενα ίχνη του Mr.Blue με την εντολή καθάρισε. Το ολοκληρωμένο πλέον σενάριο του Mr.Blue όταν πατηθεί η πράσινη σημαία είναι το παρακάτω:



Κίνηση της Mrs.Pink

Τι θα λέγατε να βοηθήσουμε την Mrs.Pink να έρθει πιο κοντά στον Mr.Blue; Όπως έχουμε αναφέρει μέχρι τώρα, το αντικείμενο αυτό ακολουθεί κατά βήμα την κίνηση του Mr.Blue. Για να το πετύχει αυτό πρέπει να εκμεταλλευτεί τις λίστες «x» και «y» που περιέχουν τις πληροφορίες μετακίνησης του άλλου αντικείμενου. Σκοπός της Mrs.Pink είναι να περάσει από όλες τις θέσεις από τις οποίες πέρασε και ο Mr.Blue. Επομένως, όσο βρίσκει στοιχεία στις λίστες αυτές, ίχνη δηλαδή του Mr. Blue, η κίνηση της δε σταματάει. Πως όμως θα διαβάσει τις διαφορετικές συντεταγμένες που είναι αποθηκευμένες στις λίστες; Με μια επανάληψη και μια μεταβλητή-μετρητή. Η επανάληψη θα διαρκεί όσο ο μετρητής είναι μικρότερος από το μήκος της λίστας, δηλαδή όσο υπάρχουν στοιχεία μέσα στη λίστα που δεν έχει επισκεφτεί. Όσο δηλαδή δεν έχει φτάσει στον αγαπημένο της!

Αρχικά, η μεταβλητή μετρητής θα πρέπει να παίρνει την τιμή 0 και στη συνέχεια θα πρέπει να αλλάζει κατά ένα σε κάθε επανάληψη έτσι ώστε κάθε φορά να δείχνει στα επόμενα στοιχεία των λιστών x και y. Η κίνηση προς τα στοιχεία αυτά για την Mrs. Pink θα πραγματοποιείται με την εντολή πήγαινε στο x... y... από την παλέτα Κίνηση. Οι συντεταγμένες του σημείου μετακίνησης θα προκύψουν από τη χρήση των εντολών στοιχείο [] του [x] και στοιχείο [] του [y] σε συνδυασμό με τη μεταβλητή-μετρητή «θέση». Μελετήστε το παρακάτω σενάριο:



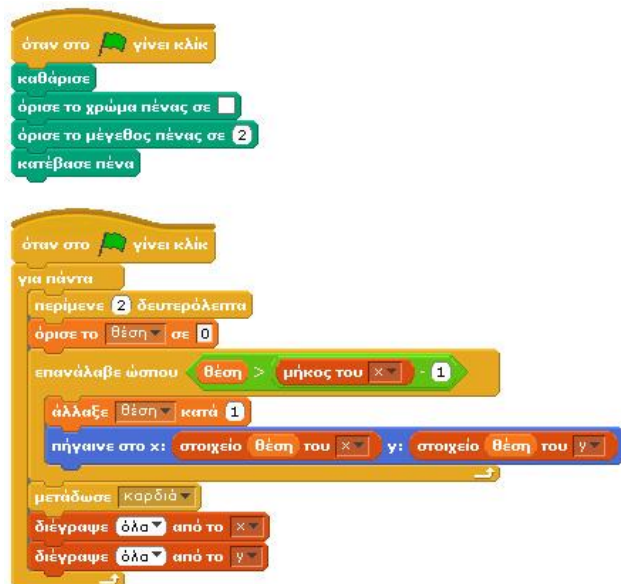
Γιατί έχουμε τη συγκεκριμένη συνθήκη στην επανάληψη και όχι τη συνθήκη (θέση > μήκος του x); Πως θα ξαναγράφατε τη συνθήκη χρησιμοποιώντας τον τελεστή της ισότητας;

Ας γυρίσουμε όμως στο παιχνίδι. Ακόμη κι αν η Mrs.Pink φτάσει τον αγαπημένο της, το παιχνίδι δε σταματά αφού ο Mr.Blue μπορεί να ξεφύγει και να ξεκινήσει νέο κυνηγητό. Για το λόγο αυτό η διαδικασία της κίνησης της Mrs.Pink θα πρέπει να εκτελείται συνεχώς. Αυτό το εξασφαλίζει η εντολή «για πάντα...».

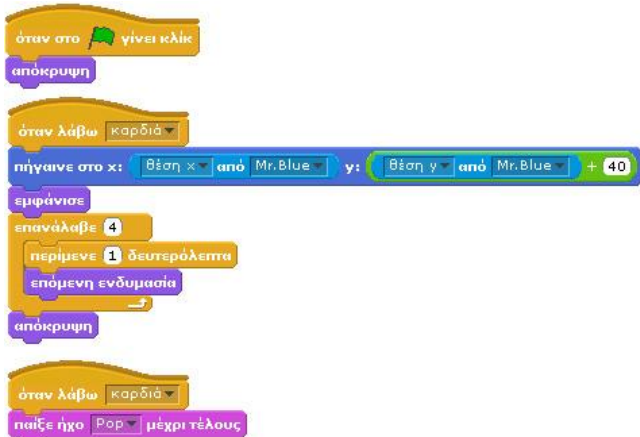
Ακόμη, η Mrs.Pink επιθυμεί να σβήνει τα ίχνη του αγαπημένου της. Για το λόγο αυτό θα χρησιμοποιήσει πένα με χρώμα λευκό και πάχος ίσο με αυτό της πέννας του Mr.Blue. Οι ιδιότητες της πέννας θα ορίζονται κατά την έναρξη του προγράμματος, όπως και το κατέβασμά της. Ίχνη βέβαια δεν αποτελούν μόνο τα α-

ποτυπώματα που αφήνει η πένα του Mr.Blue αλλά και οι συντεταγμένες των θέσεων από τις οποίες πέρασε. Για το λόγο αυτό η Mrs.Pink κάθε φορά που συναντά τον Mr.Blue διαγράφει τα περιεχόμενα των λιστών «x» και «y» με τη χρήση της εντολής **διέγραψε [όλα] από το [x/y]**.

Τέλος, επειδή η Mrs.Pink πρέπει να αφήνει τον Mr.Blue για λίγο ελεύθερο για να μετακινηθεί, θα χρειαστούμε και την εντολή **περίμενε...δευτερόλεπτα**. Τα ολοκληρωμένα σενάρια για την Mrs.Pink φαίνεται στην παρακάτω εικόνα.



Τι θα λέγατε να προσθέταμε ένα ακόμη αντικείμενο το οποίο να έχει σχήμα καρδιάς και κάθε φορά που η Mrs.Pink φτάνει τον Mr.Blue να εμφανίζεται στη σκηνή; Για να γνωρίζει βέβαια αυτό το αντικείμενο πότε να εμφανιστεί, θα πρέπει η Mrs.Pink να μεταδίδει ένα μήνυμα τη στιγμή που φτάνει τον αγαπημένο της. Ας ονομάσουμε το αντικείμενο αλλά και το συγκεκριμένο μήνυμα ως «καρδιά». Όταν λοιπόν το μήνυμα αυτό ληφθεί από το αντικείμενο Καρδιά, τότε ο έρωτας τις Mrs.Pink θα γίνεται ακόμη πιο φανερός, ενώ το αντικείμενο μπορεί να αλλάζει ενδυμασίες έτσι ώστε να φαίνεται ότι μεγαλώνει και να αναπαράγει και κάποιον ήχο. Μία ενδεικτική υλοποίηση του αντικειμένου αυτού παρουσιάζεται στην επόμενη εικόνα.



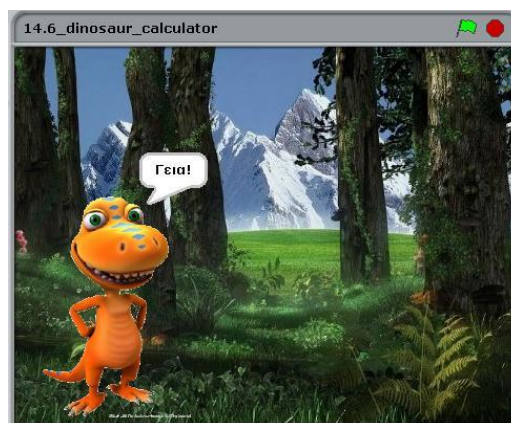
Παράδειγμα «ο δεινόσαυρος-μαθηματικός»: Ας δούμε ένα πρόγραμμα το οποίο υπολογίζει απλές μαθηματικές εκφράσεις χρησιμοποιώντας, τις μεταβλητές, τους τελεστές της πρόσθε-

σης (+) και του πολλαπλασιασμού (*). Σκοπός μας είναι να δημιουργήσουμε ένα πρόγραμμα το οποίο να μπορεί να υπολογίζει και να υπολογίζει το αποτέλεσμα των παρακάτω μαθηματικών εκφράσεων ανάλογα με την επιλογή του χρήστη.

- ✓ Εμβαδό ορθογωνίου = $\alpha * \beta$, όπου α και β οι πλευρές του ορθογωνίου.
- ✓ Περίμετρος ορθογωνίου = $2*\alpha + 2*\beta$, όπου α και β οι πλευρές του ορθογωνίου.
- ✓ Εμβαδό κύκλου = $3.14*\alpha*\alpha$, όπου α η ακτίνα του κύκλου.
- ✓ Περίμετρος κύκλου = $2*(3.14)*\alpha$, όπου α η ακτίνα του κύκλου.

Τις παραμέτρους α και β θα πρέπει να τις δίνει ο χρήστης ως είσοδο στο πρόγραμμα και το έργο μας θα κάνει τους υπολογισμούς. Δηλαδή θα δημιουργήσουμε ένα γεωμετρικό υπολογιστή!

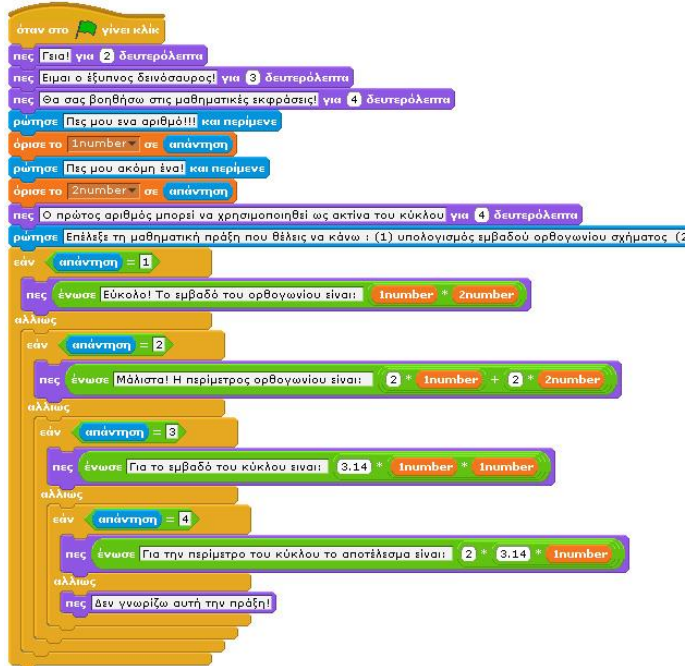
Για τον υπολογισμό των παραπάνω μαθηματικών εκφράσεων θα μας βοηθήσει ένας μικρός δεινόσαυρος. Το σκηνικό και ο δεινόσαυρος φαίνονται στην επόμενη εικόνα.



Όπως αναφέραμε, ο χρήστης θα πρέπει να δίνει δύο αριθμούς βάσει των οποίων θα γίνονται οι υπολογισμοί με τη χρήση της εντολής **ρώτση...και περίμενε** και της μεταβλητής **απάντηση**. Για να μπορέσουμε να χρησιμοποιήσουμε τους αριθμούς αυτούς πολλές φορές και σε διαφορετικές εντολές, θα πρέπει να τους αποθηκεύσουμε. Για το λόγο αυτό θα χρειαστούμε τη δημιουργία δυο αντίστοιχων μεταβλητών με όνομα 1number και 2number. Η πρώτη τιμή που εισάγει ο χρήστης θα αποθηκευτεί στη μεταβλητή με όνομα 1number, ενώ η δεύτερη στη 2number. Έπειτα, ο χρήστης καλείται να επιλέξει ποια πράξη επιθυμεί να εκτελεστεί. Αν επιλέξει τον αριθμό 1, τότε πραγματοποιείται η πράξη για την εύρεση του εμβαδού ενός ορθογωνίου. Αν επιλέξει τον αριθμό 2, πραγματοποιείται ένας συνδυασμός πράξεων για την εύρεση της περιμέτρου του ορθογωνίου. Αν επιλέξει 3, ο δεινόσαυρος εκτελεί τον υπολογισμό του εμβαδού του κύκλου. Αν επιλέξει 4, εκτελείται η πράξη για την εύρεση της περιμέτρου του κύκλου. Όλοι υπολογισμοί μπορούν να γίνουν με τη χρήση των τελεστών πολλαπλασιασμού [*****] και πρόσθεσης [**+**]. Για να γίνει όμως πιο πειστική η ανακοίνωση του αποτελέσματος από το δεινόσαυρο, χρησιμοποιούμε ακόμη μία εντολή της παλέτας των Τελεστών, την **ένωσε [] []** η οποία μας επιτρέπει να ενώσουμε ένα σύντομο κείμενο με μια μεταβλητή

και να δημιουργήσουμε ένα ενιαίο κείμενο που μπορεί να «ειπωθεί» από τον ήρωα μας.

Μελετήστε το παρακάτω σενάριο:



[14_π07.sb]

Περίληψη

Στο κεφάλαιο αυτό ασχοληθήκαμε με δύο ιδιαίτερα σημαντικές έννοιες τόσο του Scratch, όσο και του προγραμματισμού γενικότερα. Δεν είναι οι πιο εύκολες έννοιες που συναντήσαμε! Αλλά είναι αυτές που πραγματικά θα μας επιτρέψουν στη συνέχεια να αναπτύξουμε εκπληκτικά παιχνίδια. Ασχοληθήκαμε με τις μεταβλητές, που είναι δομές αποθήκευσης τιμών που χρειάζεται ένα έργο κατά τη διάρκεια εκτέλεσής του. Οι τιμές αυτές μπορεί να αφορούν ένα και μόνο αντικείμενο ή να διαβάζονται και να αλλάζουν από όλα τα αντικείμενα. Συνήθως, δίνουμε μια αρχική τιμή σε κάθε μεταβλητή στην αρχή του προγράμματος και αλλάζουμε την τιμή αυτή κατά τη διάρκεια εκτέλεσης του έργου. Σκορ, ζωές, χρόνος κτλ θα αποθηκεύονται τις περισσότερες φορές σε μεταβλητές. Στις λίστες μπορούν να αποθηκευτούν περισσότερες από μία τιμές χωρίς να χρειάζεται να γνωρίζουμε από την έναρξη του προγράμματος πόσες τιμές θέλουμε να αποθηκεύσουμε. Σε μία λίστα, μπορούμε να προσθέσουμε, να διαγράψουμε αλλά και να αντικαταστήσουμε τα στοιχεία της. Επιπλέον, το Scratch μας δίνει τη δυνατότητα να εξετάσουμε εύκολα το περιεχόμενο μιας λίστας και να γνωρίζουμε το μήκος της. Έχουμε λοιπόν δυο αποθηκευτικά εργαλεία που είναι απολύτως απαραίτητα όπως θα καταλάβετε και στο τελευταίο κεφάλαιο του βιβλίου μας.

Ερωτήσεις

1) Έστω ότι σε μία εφαρμογή έχουμε τη μεταβλητή «άθροισμα» την οποία αρχικοποιούμε με την τιμή 0. Ποια τιμή θα έχει η μεταβλητή μετά την εκτέλεση της εντολής άλλαξε [άθροισμα] κατά [-2];

2) Στην παρακάτω εικόνα εμφανίζεται η λίστα «αριθμοί» με τα περιεχόμενά της.

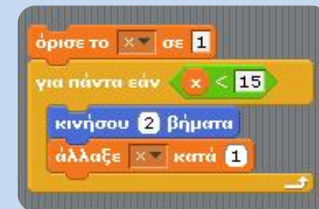


Ποια θα είναι η μορφή της λίστας μετά την εκτέλεση της εντολής παρέμβαλλε το [4] στη θέση [2] του [αριθμοί];

3) Αν μία λίστα περιέχει 3 στοιχεία και εμείς εκτελέσουμε την εντολή παρέμβαλλε το [αντικείμενο] στη θέση [οποιοδήποτε] του [λίστα], σε πόσα σημεία είναι δυνατό να μπει το αντικείμενο;

4) Αν θέλετε κάθε στοιχείο που προσθέτετε σε μία λίστα να μπαίνει στην πρώτη και όχι στην τελευταία της θέση, ποια εντολή πρέπει να χρησιμοποιήσετε;

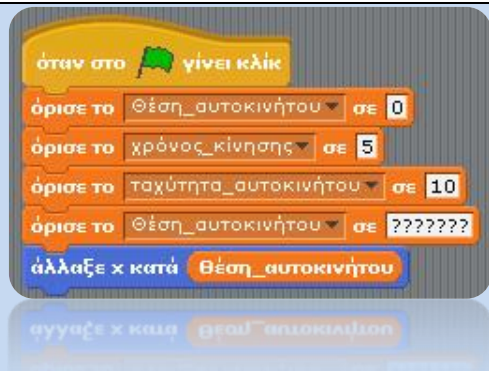
5) Στην παρακάτω εικόνα δίνεται ένα σενάριο επανάληψης. Πόσες φορές θα εκτελεστεί ο κώδικας που περιέχεται μέσα στην εντολή για πάντα εάν ...;



6) Δίνεται το παρακάτω σενάριο. Με πόσους διαφορετικούς τρόπους μπορεί να τερματιστεί η επανάληψη;



7) Δίνεται το παρακάτω σενάριο στο οποίο θέλουμε να υπολογίζουμε τη θέση ενός αυτοκινήτου που κινείται με ταχύτητα «ταχύτητα_αυτοκινήτου» μετά από χρόνο «χρόνος_κίνησης». Στον κώδικα αυτό λείπει ο τύπος για τον υπολογισμό της θέσης αυτής. Μπορείτε να τον συμπληρώσετε με τη βοήθεια των αριθμητικών τελεστών; Θυμίζουμε ότι ο τύπος για τη μετατόπιση ενός αντικείμενου που κινείται ομαλά είναι: $x_{\text{τελικό}} = x_{\text{αρχικό}} + u \cdot \Delta t$.



8) Δημιουργήστε ένα έργο το οποίο θα ζητάει από το χρήστη την είσοδο δύο αριθμών που θα αντιστοιχούν στο μήκος των δύο κάθετων πλευρών ενός ορθογωνίου τριγώνου. Το έργο σας θα υπολογίζει το μήκος της υποτείνουσας με τη βοήθεια του Πυθαγορείου Θεωρήματος. Το Πυθαγόρειο Θεώρημα είναι το ακόλουθο: $\alpha^2 = \beta^2 + \gamma^2$.

9) Δημιουργήστε ένα έργο στο οποίο ο χρήστης θα δίνει τυχαία αριθμούς τους οποίους το έργο σας θα αποθηκεύει σε μία λίστα αλλά πάντα με αριθμητική σειρά από το μικρότερο προς το μεγαλύτερο αριθμό.

10) Δημιουργήστε ένα πρόγραμμα που θα υλοποιεί την αντίστροφη μέτρηση. Στο πρόγραμμα αυτό, ο χρήστης θα δίνει τον αριθμό των δευτερολέπτων και έπειτα ο χρόνος θα μετράει αντίστροφα.

Δραστηριότητες

1) Δημιουργήστε μία εφαρμογή στην οποία θα υπάρχει μία φιγούρα της επιλογής σας, όπως ένα άτομο ή ένα ζώο, το οποίο θα προσπαθεί να συλλέξει τη μεγαλύτερη αξία σε κέρματα που μπορεί μέσα σε 15 δευτερόλεπτα. Τα κέρματα που θα εμφανίζονται στη σκηνή θα πρέπει να είναι τόσα ώστε να μην είναι εύκολο να τα μαζέψει όλα μέσα σε αυτό το χρονικό διάστημα. Ένας ικανοποιητικός αριθμός κερμάτων θα ήταν 30. Επιπλέον, τα κέρματα θα πρέπει να έχουν διαφορετική αξία, π.χ. να υπάρχουν χρυσά και ασημένια τα οποία θα προσφέρουν διαφορετικούς πόντους στον παίκτη. Μία ενδεικτική τιμή θα ήταν οι 3 πόντοι για τα χρυσά και 1 πόντος για τα ασημένια. Επίσης, θα είχε ενδιαφέρον αν στη σκηνή βάζατε και κάποια εμπόδια που θα κάνουν δύσκολη την πρόσβασή του χαρακτήρα στα κέρματα. Το πλήθος, τη μορφή και τις θέσεις των εμποδίων μπορείτε να τις αποφασίσετε μόνοι σας. Τα εμπόδια θα μπορούσαν να έχουν για παράδειγμα τη μορφή βράχων. Σε περίπτωση που τα ακουμπήσει θα χάνει. Νικητής είναι αυτός που θα συγκεντρώσει τους περισσότερους πόντους μέσα στα 15 αυτά δευτερόλεπτα χωρίς να πέσει πάνω στα εμπόδια.

2) Δημιουργήστε ένα τυχερό παιχνίδι! Στο παιχνίδι αυτό θα υπάρχει μία λίστα όλων των ακεραίων αριθμών από το 1 έως το 40. Ο χρήστης αρχικά θα πληκτρολογεί έξι αριθμούς της επιλογής του. Στη συνέχεια, με τυχαίο τρόπο 34 από τους 40 αριθμούς της λίστας θα διαγράφονται. Ανάλογα με το πόσοι από τους έξι αριθμούς που έδωσε ο χρήστης βρίσκονται στη νέα λίστα θα προσδιορίζεται και το ποσοστό της επιτυχίας του.

3) Δημιουργήστε μία εφαρμογή στην οποία θα υπάρχει ένα κανόνι που θα κινείται από το χρήστη. Η θέση του κανονιού θα παραμένει σταθερή στο κάτω μέρος της σκηνής, ενώ η κατεύθυνση του θα μπορεί να αλλάζει με το δεξί και το αριστερό βέλος του πληκτρολογίου. Μπορείτε να επιλέξετε εσείς ποιο θα είναι το βήμα με το οποίο θα αλλάζει η κατεύθυνση του κανονιού, μία ενδεικτική τιμή είναι οι 5 μοίρες. Το κανόνι θα ρίχνει σφαίρες στην κατεύθυνση στην οποία δείχνει, οι οποίες θα κινούνται μέχρι να συναντήσουν τα όρια της σκηνής. Με τις σφαίρες ο παίκτης θα προσπαθεί να πετύχει τις τέσσερις πάπιες που θα υπάρχουν στη σκηνή. Οι πάπιες αυτές θα κινούνται γύρω από τη θέση τους δεξιά και αριστερά. Τις αρχικές τους θέσεις μπορείτε να τις αποφασίσετε μόνοι σας. Άλλες μπορούν να είναι πιο κοντά στο κανόνι και άλλες πιο μακριά από αυτό. Κάθε πάπια έχει τρεις ζωές. Κάθε φορά που μία σφαίρα πετύχει κάποια πάπια, η ζωή της μειώνεται κατά μία. Όταν οι ζωές κάποιας πάπιας τελειώσουν, τότε αυτή θα πρέπει να βγαίνει από το παιχνίδι. Για να τερματιστεί το παιχνίδι θα πρέπει όλες οι πάπιες να έχουν χάσει τις ζωές τους. Επιπλέον, θα μπορούσατε να χρησιμοποιήσετε ένα χρονόμετρο για να περιορίσετε το χρόνο που έχει ο παίκτης στη διάθεση του.



Κεφάλαιο 15: Παιχνίδια

Σε αυτό το κεφάλαιο:

- 15.1 Super Pong
- 15.1 Ναρκοπέδιο!
- 15.3 Σκοποβολή
- 15.4 Ο καρχαρίας
- 15.5 Παραλλαγή Pacman
- 15.6 Καροτοκυνηγός



15.1 Super Pong

[15_π01.sb]

Στο παιχνίδι Super Pong που θα αναπτύξουμε, δύο μπάλες κινούνται ακατάπαυστα, αναπηδούν στα όρια της οθόνης του Scratch αλλά δεν επιτρέπεται να ακουμπήσουν στο κάτω μέρος της οθόνης. Για ακριβώς αυτό το λόγο, ο χρήστης αναλαμβάνει να αποκρούσει οποιαδήποτε μπάλα πλησιάζει προς το κάτω μέρος της οθόνης μέσω μιας ρακέτας που κινείται οριζόντια ανάλογα με τη θέση του δείκτη του ποντικιού. Το σκορ του χρήστη διαμορφώνεται από τις αποκρούσεις που κατάφερε να κάνει ενώ αν έστω και μια μπάλα φτάσει στο κάτω όριο, το παιχνίδι τερματίζεται.

Δεν πρέπει να ξεχνάμε ότι τα προβλήματα στο Scratch λύνονται πιο εύκολα αν αποκτήσουμε μια μεθοδική προσέγγιση στην ανάλυσή τους. Τα ερωτήματα που έχουμε να απαντήσουμε είναι τα ακόλουθα:

- 1) Ποια είναι τα διαφορετικά αντικείμενα που διακρίνουμε και ποια η γενική περιγραφή τους; Ποιο θα είναι το σκηνικό μας και πως θα συμμετάσχει στην επίλυση του προβλήματος;
- 2) Για κάθε αντικείμενο του προηγούμενου βήματος (αντικείμενα και σκηνικό):

- ✓ Χρειάζονται διαφορετικές **εμφανίσεις** και **πολλαπλοί ήχοι** για το αντικείμενο;
- ✓ Ποιες είναι οι **συμπεριφορές** του αντικειμένου (τι θα πρέπει να κάνει);
- ✓ Σε ποια **γεγονότα αντιδρά** το αντικείμενο; (γεγονότα και αισθητήρες)
- ✓ Χρειάζεται να **επικοινωνήσει** το συγκεκριμένο αντικείμενο με άλλα;
- ✓ Χρειάζεται το αντικείμενο να αποθηκεύει πληροφορίες σε **μεταβλητές ή λίστες**;
- ✓ Χρειαζόμαστε **τυχαιότητα** στη συμπεριφορά του αντικειμένου; Σε ποια σημεία;

Αντικείμενα & Σκηνικό

Από την περιγραφή του προβλήματος, είναι σαφές ότι τα αντικείμενά μας θα είναι οι δυο μπάλες που κινούνται αυτόνομα και η ρακέτα η οποία ελέγχεται από το χρήστη του παιχνιδιού. Το

παιχνίδι μας θα περιλαμβάνει ως σκηνικό ένα τοπίο που θα δυσκολεύει οπτικά τη διάκριση κάθε μπάλας.



Αντικείμενο 1: Σκηνικό

Το σκηνικό μας δεν θα συμμετάσχει ενεργά στο έργο μας αφού δεν αλλάζει, ούτε εκτελεί κάποιες γενικές λειτουργίες (αν και θα μπορούσαμε να βάλουμε ένα τραγούδι στο παιχνίδι μας...). Έχουμε όμως μια απαίτηση από αυτό! Για να διευκολύνουμε τη δημιουργία σεναρίων για τον τερματισμό του παιχνιδιού, θα δημιουργήσουμε μια κόκκινη περιοχή στο κάτω μέρος του σκηνικού έτσι ώστε όταν οι μπάλες ακουμπούν αυτή τη γραμμή, το παιχνίδι να τερματίζεται. Το σκηνικό που φαίνεται στην προηγούμενη εικόνα έχει επιλεγεί από το φάκελο nature της βιβλιοθήκης υποβάθρων και έχει όνομα lake.

Αντικείμενο 2: Η ρακέτα

Θα σχεδιάσουμε τη ρακέτα στον επεξεργαστή ζωγραφικής τραβώντας μια μικρή οριζόντια μαύρη γραμμή. Θα τοποθετήσουμε τη ρακέτα κοντά στο κάτω όριο της οθόνης του Scratch. Θέλουμε ο παίχτης του παιχνιδιού να χειρίζεται τη ρακέτα, μέσω του δείκτη του ποντικιού του. Επομένως μεταφέροντας το ποντίκι προς τα δεξιά, θα πρέπει να μεταφέρεται και η ρακέτα

προς τα δεξιά και, συνεπώς, πρέπει να συνδέσουμε την οριζόντια θέση της ρακέτας με την οριζόντια θέση του δείκτη του ποντικιού. Μελετήστε το παρακάτω σενάριο:



Με τη συγκεκριμένη εντολή (θέσε το x ίσο με...) και τη χρήση της μεταβλητής **ποντίκι x**, καταφέρνουμε η ρακέτα να ακολουθεί οριζόντια το δείκτη του ποντικιού.

Αντικείμενο 3: Οι μπάλες

Ας συνοψίσουμε τη συμπεριφορά που θέλουμε να έχουν οι δύο μπάλες:

- ✓ Θέλουμε να κινούνται συνεχώς και να αναπηδούν όταν ακουμπούν τα όρια της οθόνης,
- ✓ Θέλουμε να τερματίζεται το παιχνίδι, αν οι μπάλες ακουμπήσουν την κόκκινη περιοχή,
- ✓ Θέλουμε να αναπηδούν όταν ακουμπούν την ρακέτα. Κάθε φορά που αναπηδούν λόγω της ρακέτας το σκορ του χρήστη θα πρέπει να αυξάνεται.

Ας ονομάσουμε τις δυο μπάλες ως Μπάλα1 και Μπάλα2. Εισάγουμε τις αντίστοιχες μορφές από το φάκελο things της βιβλιοθήκης αντικειμένων του Scratch. Τα σενάρια για τις δυο μπάλες είναι πανομοιότυπα και επομένως θα αναλύσουμε ένα από τα δυο.

Διακρίναμε παραπάνω τρεις διαφορετικές συμπεριφορές. Μέχρι τώρα προσπαθήσαμε να εντάξουμε τις διαφορετικές συμπεριφορές ενός αντικείμενου μέσα σε ένα και μόνο σενάριο. Μήπως όμως υπάρχει και άλλη λύση; Τι θα λέγατε αν, για αλλαγή, δημιουργούσαμε τρία διαφορετικά σενάρια που αφορούν τις τρεις διαφορετικές συμπεριφορές και τα οποία τρέχουν ταυτόχρονα όταν πατηθεί η πράσινη σημαία;

Το πρώτο σενάριο που αφορά τη συνεχή κίνησης της μπάλας και την αναπήδηση όταν ακουμπά τα όρια της οθόνης, είναι πολύ απλό. Πρέπει να προσδιορίσουμε την αρχική θέση της μπάλας, να της δώσουμε μια τυχαία κατεύθυνση προς τα κάτω (δηλαδή από 90 έως 270 μοίρες) και στη συνέχεια να εξασφαλίσουμε ότι η μπάλα θα κινείται για πάντα:



Το δεύτερο σενάριο αφορά τον τερματισμό του παιχνιδιού. Χρειαζόμαστε ένα σενάριο ώστε όταν η μπάλα αγγίξει την κόκκινη περιοχή να βγαίνει ένα σύντομο μήνυμα και να σταματούν όλα τα σενάρια:



Το τρίτο σενάριο αφορά την συμπεριφορά της μπάλας όταν έρχεστε σε επαφή με την ρακέτα. Καταρχάς, θα χρειαστούμε μια νέα μεταβλητή που θα μπορεί να χρησιμοποιηθεί και από τις δυο μπάλες και στην οποία θα αποθηκεύουμε το σκορ του χρήστη. Έστω ότι ονομάζουμε τη μεταβλητή αυτή σκορ. Κατά την έναρξη του προγράμματος θα πρέπει να μηδενίζουμε την τιμή της σε μια από τις δυο μπάλες. Από και και πέρα, όποτε η μπάλα αγγίζει τη ρακέτα (οι αισθητήρες θα μας βοηθήσουν να αναγνωρίσουμε το γεγονός αυτό), θα πρέπει να αυξάνεται το σκορ και η μπάλα να αναπηδά προς τα πάνω. Για να αναπηδά προς τα πάνω, θα χρησιμοποιήσουμε μια μαθηματική έκφραση για να αφαιρέσουμε την τρέχουσα κατεύθυνση από το 180 ώστε να δώσουμε συμπληρωματική κατεύθυνση στη μπάλα. Για να μην είναι όμως μονότονο το παιχνίδι, θα αλλάζουμε ελαφρώς την κατεύθυνση της μπάλας με τυχαίο τρόπο.



Υπάρχουν δυο ενδιαφέροντα σημεία στο συγκεκριμένο σενάριο για τους παρατηρητικούς:

- ✓ Έχουμε εισάγει μια εντολή κίνησης όταν το αντικείμενο αγγίζει τη ρακέτα. Γιατί δεν αφήνουμε στο πρώτο από τα τρία σενάρια την κίνηση; Και γιατί το κινούμε με μεγαλύτερο βήμα από το «κανονικό»; Η κίνηση με βήματα έχει ως αποτέλεσμα να είναι πιθανόν η επαφή της μπάλας με τη ρακέτα να γίνει σε διαφορετικό βάθος (από 1-4 βήματα), δηλαδή η μπάλα να «ακουμπάει» λιγότερο ή περισσότερο τη ρακέτα. Αυτό μπορεί να γίνει ακόμη πιο έντονο αν η ρακέτα αγγίξει την μπάλα από τις πλευρές της και όχι από πάνω. Σε όλες αυτές τις περιπτώσεις θα πρέπει να εξασφαλίσουμε ότι μπάλα θα φύγει μακριά από την ρακέτα μέσα στην ίδια επανάληψη ώστε να μην ξαναγίνει αληθής ο έλεγχος «εάν αγγίζει τη ρακέτα». Δοκιμάστε να αφαιρέσετε τις συγκεκριμένες γραμμές και συζητήστε τι συμβαίνει.
- ✓ γιατί αρχικά αλλάζουμε κατεύθυνση, στη συνέχεια κινούμε λίγο το αντικείμενο και μετά ξανα-αλλάζουμε κατεύθυνση; Μήπως είναι λίγο υπερβολικό αυτό; Πως θα μπορούσε να γίνει το σενάριο πιο συμπαγές και αποδοτικό;

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Δοκιμάστε να δημιουργήσετε ξανά το παιχνίδι χρησιμοποιώντας μόνο ένα και όχι τρία σενάρια.
- 2) Προσπαθήστε να δημιουργήσετε ένα τρόπο αξιολόγησης της απόδοσης του παιχνιδιού για τις δυο εναλλακτικές υλοποίησής του, χρησιμοποιώντας έναν μετρητή χρόνου και χρονομετρώντας συγκεκριμένες ενέργειες του παιχνιδιού. Ποιος από τους δυο τρόπους υλοποίησης του παιχνιδιού είναι γρηγορότερος; Αιτιολογήστε την απάντησή σας.



15.2 Ναρκοπέδιο!

[15_π02.sb]

Όλοι μας έχουμε παίξει παιχνίδια όπου το χαρακτηριστικό που θα κρίνει το νικητή είναι το ποιος θα καταφέρει να πετύχει ένα συγκεκριμένο στόχο στο μικρότερο δυνατό χρόνο. Και στο Scratch μπορούμε να δημιουργήσουμε ενδιαφέροντα παιχνίδια έχοντας αντίπαλο το χρόνο.

Για να το πετύχουμε αυτό χρειαζόμαστε το χρονόμετρο, μια μεταβλητή κατάστασης η οποία βρίσκεται στην παλέτα Αισθητήρες. Για να εμφανιστεί το χρονόμετρο πρέπει να επιλέξουμε την προβολή του χρονομέτρου στη σκηνή τσεκάροντας την αντίστοιχη επιλογή.



Το χρονόμετρο ξεκινά να μετράει δευτερόλεπτα αυτόματα από τη στιγμή που εκκινούμε το Scratch! Αν τσεκάρουμε την αντίστοιχη επιλογή για να εμφανιστεί στην οθόνη, τότε θα παρατηρήσουμε ότι α) ο χρόνος κυλάει παρότι δεν εκτελούμε κάποιο έργο (!) β) η αρχική του τιμή δεν είναι 0 αλλά αντιστοιχεί στα δευτερόλεπτα που πέρασαν από τη στιγμή που ανοίξαμε το Scratch(!). Επειδή, ο χρόνος αυτός δεν μας ενδιαφέρει στα έργα μας, το Scratch μας παρέχει μια εντολή για να μηδενίζουμε την τιμή του, την εντολή μηδένισε το χρονόμετρο.

Αν λοιπόν θέλουμε να χρησιμοποιήσουμε το χρονόμετρο σε κάποιο πρόγραμμα, θα πρέπει απλά στην αρχή του έργου να το μηδενίσουμε.

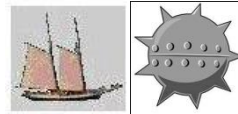


Πάμε τώρα στο νέο παιχνίδι που θα δημιουργήσουμε. Έστω ότι είμαστε καπετάνιοι σε ένα πλοίο που έχει μπλεχτεί σε ένα θαλάσσιο ναρκοπέδιο και προσπαθεί να φτάσει στη στεριά. Όποιος καπετάνιος κάνει το καλύτερο χρόνο, αυτός είναι και ο νικητής. Φυσικά χρειάζεται πολύ προσοχή στις νάρκες οι οποίες είναι έτοιμες να εκραγούν! Οι νάρκες δε μένουν σταθερές στη θέση τους αλλά κινούνται λόγω θαλασσίων ρευμάτων κάνοντας πιο δύσκολη την προσπάθεια του πλοίου να φτάσει στη στεριά. Αν το πλοίο ακουμπήσει σε μία από αυτές τις νάρκες, τότε αυτή θα εκραγεί και το πλοίο θα πάρει φωτιά και φυσικά ο παίκτης θα χάσει.



Αντικείμενα και Σκηνικό

Σίγουρα θα χρειαστούμε ένα αντικείμενο πλοίο και ένα αντικείμενο-νάρκη το οποίο θα αντιγράψουμε πολλές φορές ώστε να γεμίσουμε το ναρκοπέδιό μας. Ψάχνοντας στο διαδίκτυο αλλά και στη βιβλιοθήκη του Scratch επιλέξαμε τις εξής ενδυμασίες για τα αντικείμενα:



Τι σκηνικό θα θέλαμε να χρησιμοποιήσουμε για μια τέτοιου είδους εφαρμογή; Σκεφτόμαστε ένα φόντο θάλασσας με τη στεριά που να φαίνεται σε κάποια γωνία. Ενδεικτικά λοιπόν το σκηνικό θα μπορούσε να ήταν αυτό της επόμενης εικόνας.



Αντικείμενα νάρκες

Τώρα ας σκεφτούμε τα εμπόδια που θα υπάρχουν στο σκηνικό μας. Καλό θα ήταν να υπάρχουν αρκετές νάρκες έτσι ώστε να κάνουμε τη διαδρομή του πλοίου δυσκολότερη. Έστω, λοιπόν, ότι εισάγουμε πέντε νάρκες, οι οποίες θα επιπλέουν στη θάλασσά μας και θα είναι διασκορπισμένες σε όλη την επιφάνεια του νερού. Ποια θα είναι η συμπεριφορά τους; Απλά θα κινούνται ανάμεσα σε σταθερές θέσεις. Καλό θα ήταν κάποιες από αυτές να κινούνται κάθετα, άλλες οριζόντια και άλλες διαγώνια. Το ολοκληρωμένο σενάριο για μια από τις νάρκες:



Ορίζουμε μια αρχική θέση, και με την εντολή επανάληψης για πάντα, προκαλούμε τις νάρκες να μετακινούνται ομαλά μεταξύ δυο θέσεων. Ανάλογα με το χρόνο κίνησης που θα επιλέξετε στις εντολές κινήσου ομαλά...δευτ, θα προσδιορίσετε και τη δυσκολία του παιχνιδιού. Δημιουργήστε ένα μόνο αντικείμενο και στη συνέχεια πατήστε δεξί κλικ πάνω του (στη λίστα αντικειμένων) και επιλέξτε διπλασίασε. Εύκολα μπορείτε να δημιουργήσετε άλλα τέσσερα αντικείμενα που έχουν ενσωματωμένο μέσα τους το ίδιο σενάριο. Απλά αλλάξτε τις συντεταγμένες στις αντίστοιχες εντολές.

Αντικείμενο πλοίο

Το πλοίο μας χρειάζεται μια δεύτερη ενδυμασία (έκρηξη) έτσι ώστε σε περίπτωση που ακουμπήσει μια νάρκη να φανεί πιο ρεαλιστικά το αποτέλεσμα. Η ενδυμασία αυτή θα μπορούσε να μοιάζει με αυτήν που εμφανίζεται στην παρακάτω εικόνα.



Η συμπεριφορά του καραβιού είναι εύκολη. Ας διακρίνουμε της διαστάσεις της:

- ✓ το πλοίο πρέπει να κινείται με τα βελάκια του πληκτρολογίου ώστε να μπορεί να μετακινηθεί μέχρι τη στεριά,
- ✓ αν το πλοίο καταφέρει να «αγγίξει» τη στεριά τότε το παιχνίδι μας θα έχει τερματιστεί με επιτυχία. Η άφιξη του πλοίου στη στεριά μπορεί να γίνει αντιληπτή μόλις το πλοίο ακουμπήσει το νησάκι που βρίσκεται στην κάτω αριστερή γωνία του σκηνικού.
- ✓ αν το πλοίο «αγγίξει» μια νάρκη, τότε φοράει την ενδυμασία «έκρηξη» και τερματίζεται το παιχνίδι.

Στην παρουσίαση του προηγούμενου παιχνιδιού, εισάγαμε τις τρεις διαφορετικές συμπεριφορές του αντικειμένου «ρακέτα» σε τρία διαφορετικά σενάρια. Το ίδιο θα μπορούσε να γίνει και σε αυτήν την περίπτωση οπότε θα είχαμε: 4 μικρά σενάρια για την κίνηση του πλοίου, ένα σενάριο που θα αφορούσε το τι θα συμβεί όταν το πλοίο αγγίζει τη νάρκη και ένα σενάριο που θα αφορούσε το τι θα συμβεί όταν το πλοίο αγγίζει τη στεριά. Θα προσπαθήσουμε όμως να μειώσουμε τον αριθμό των σεναρίων.

Όταν αρχίζει το πρόγραμμα πρέπει να φροντίσουμε το πλοίο να έχει την κατάλληλη ενδυμασία. Φυσικά, όπως και με τις νάρκες, τοποθετούμε το πλοίο στην αρχική του θέση και δεν ξεχνάμε να μηδενίσουμε το χρονόμετρο για ξεκινήσει η προσπάθεια του χρήστη. Το σενάριό μας προς το παρόν μοιάζει με το παρακάτω:



Χρειαζόμαστε να προσθέσουμε τους δυο ελέγχους για τις δυο διαφορετικές συνθήκες τερματισμού που περιγράφηκαν προηγουμένως. Προφανώς θα τις εντάξουμε μέσα σε μια εντολή **για πάντα** αφού θέλουμε να οι έλεγχοι να πραγματοποιούνται διαρκώς. Οι συνθήκες μας θα βασιστούν στο χρώμα των αντίστοιχων αντικειμένων:

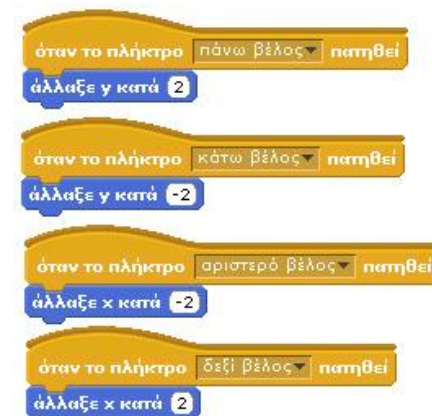
- ✓ στην περίπτωση που το πλοίο μας αγγίζει το χρώμα της νάρκης θα προσδιορίζουμε την ενδυμασία του πλοίου σε «έκρηξη» και θα σταματά το σενάριο

- ✓ στην περίπτωση που αγγίζει το χρώμα της στεριάς, τότε το πλοίο θα μας ανακοινώσει για 2 δευτερόλεπτα το χρόνο μας και θα τερματίσει το έργο.

Το αρχικό σενάριο θα γίνει:



Αφού ορίσαμε τα αρχικά δεδομένα του πλοίου μας, και τις συνθήκες τερματισμού του παιχνιδιού, θέλουμε να το κινούμε με τα βελάκια του πληκτρολογίου. Τα γνωστά τέσσερα σενάρια:



Και το παιχνίδι μας είναι έτοιμο. Πειραματιστείτε με τον αριθμό και τις θέσεις των ναρκών για να το κάνετε πιο ενδιαφέρον.

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Αλλάξτε τα σενάρια ώστε να αλλάζει ενδυμασία και η νάρκη κατά τη στιγμή της έκρηξης.
- 2) Μετασχηματίστε το παιχνίδι ώστε ο χρόνος να αποθηκεύεται σε μια μεταβλητή και στη συνέχεια μετατρέψτε το παιχνίδι ώστε να παίζεται στη σειρά από δυο παίκτες. Το παιχνίδι θα πρέπει να ανακοινώνει ποιος από τους δυο παίκτες ήταν ο τελικός νικητής.
- 3) Προσδιορίστε το χρόνο στις εντολές κίνησης των ναρκών (κινήσου ομαλά...δευτ.) με μια μεταβλητή και στη συνέχεια να εισάγετε διαφορετικά επίπεδα δυσκολίας στο παιχνίδι αλλάζοντας την τιμή αυτής της μεταβλητής. Ζητήστε από το χρήστη να επιλέξει βαθμό δυσκολίας στην αρχή του παιχνιδιού.

15.3 Σκοποβολή



[15_π03.sb]

Ας δούμε πως θα μπορέσετε να φτιάξετε ένα παιχνίδι σκοποβολής, δηλαδή ένα παιχνίδι αντανakλαστικών και ακρίβειας! Στο παιχνίδι θα υπάρχουν στόχοι οι οποίοι θα κινούνται από το ένα άκρο της οθόνης μέχρι το άλλο με διαφορετικές ταχύτητες και διαφορετικές κατευθύνσεις κάθε φορά και ο χρήστης θα προσπαθεί να τους πετύχει κάνοντας κλικ πάνω τους. Αν ευστοχήσει στο κλικ του, ο στόχος εκρήγνυται, ακούγεται σχετικός ήχος και το σκορ αυξάνει. Σε διαφορετική περίπτωση απλά ακούγεται ήχος αστοχίας! Επιπλέον, πριν ξεκινήσει το παιχνίδι, ο παίκτης θα πρέπει να καθορίσει το βαθμό δυσκολίας του. Μεγαλύτερη δυσκολία θα πρέπει να ισοδυναμεί με γρηγορότερη κίνηση των στόχων. Η διάρκεια του παιχνιδιού περιορίζεται σε 60" με το χρονόμετρο. Δείτε ένα στιγμιότυπο του παιχνιδιού στην παρακάτω εικόνα:



Αντικείμενα και Σκηνικό

Πρώτη μας δουλειά για να σχεδιάσουμε το παιχνίδι μας είναι να διακρίνουμε τα κατάλληλα αντικείμενα που θα χρειαστούμε. Είναι σαφές ότι θα έχουμε πολλά πανομοιότυπα αντικείμενα-στόχους, δηλαδή πολλές μπάλες. Στο συγκεκριμένο όμως παράδειγμα δεν διαφαίνεται να έχουμε κάποιο αντικείμενο-πρωταγωνιστή που να επιδεικνύει συγκεκριμένες συμπεριφορές. Ο χρήστης είτε κάνει κλικ στις μπάλες είτε στο σκηνικό. Μπορούμε επομένως να χρησιμοποιήσουμε το σκηνικό μας ως βασικό μας αντικείμενο για τις εντολές έναρξης του έργου.

Σκηνικό

Το σκηνικό μας θα πρέπει να έχει δύο υπόβαθρα. Το πρώτο αφορά την αρχική οθόνη υποδοχής του παίκτη, στην οποία θα ορίζεται ο βαθμός δυσκολίας του παιχνιδιού. Η αρχική οθόνη θα καλωσορίζει τον παίκτη και θα του ζητάει να εισάγει ένα γράμμα e(easy), m (medium), h (hard), για να ορίσει το βαθμό δυσκολίας του παιχνιδιού. Μπορούμε εύκολα να ζωγραφίσουμε το συγκεκριμένο υπόβαθρο στη ζωγραφική. Το αποτέλεσμα θα πρέπει να είναι κάπως έτσι:

Το δεύτερο υπόβαθρο θα αφορά την οθόνη στην οποία θα εμφανίζονται οι στόχοι και η οποία θα είναι μονόχρωμη για να είναι ευδιάκριτοι οι στόχοι.

Καλωσορίσατε

Επιλέξτε δυσκολία παιχνιδιού:

Εύκολο: e

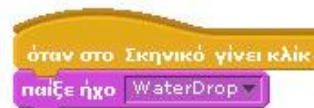
Μέτριο: m

Δύσκολο: h

Χρειάζεται επίσης να ενσωματώσουμε στο σκηνικό μας έναν ήχο που θα ακούγεται όταν ο παίκτης κάνει κλικ πάνω στο υπόβαθρο, δηλαδή όταν αποτύχει να χτυπήσει κάποιο στόχο. Θα πρέπει να είναι ένας σύντομος ήχος ώστε να μπορεί να γίνει αντιληπτός όταν ο χρήστης επαναλαμβανόμενα αστοχήσει. Στο παράδειγμά μας θα χρησιμοποιήσουμε τον ήχο WaterDrop από το φάκελο effects της βιβλιοθήκης ήχων του Scratch.

Το σκηνικό μας χρειάζεται δυο σενάρια, ένα για την περίπτωση που ο χρήστης κάνει κλικ πάνω στο σκηνικό και ένα για την παρουσίαση της οθόνης εισαγωγής αλλά και τη διαχείριση του χρόνου και του σκορ.

Το πρώτο σενάριο που αφορά την αναπαραγωγή ενός ήχου όταν ο χρήστης κάνει κλικ στο υπόβαθρο και όχι στους στόχους:



Το δεύτερο σενάριο θα πρέπει να έχει την εξής δομή:

- ✓ επιλογή επιπέδου δυσκολίας από το χρήστη,
- ✓ εκκίνηση παιχνιδιού,
- ✓ τερματισμός παιχνιδιού.

Πριν ξεκινήσουμε να βλέπουμε το σενάριο, ας σκεφτούμε τι μεταβλητές χρειαζόμαστε καθώς έχουμε να διαχειριστούμε το σκορ, τη δυσκολία και το χρονόμετρο. Επιλέγουμε το χρονόμετρο από την παλέτα **Αισθητήρες** ενώ για το σκορ δημιουργούμε μια μεταβλητή που πρέπει να είναι ορατή από όλα τα αντικείμενα. Πως όμως θα ελέγξουμε τη δυσκολία του παιχνιδιού; Όπως είδαμε και από προηγούμενα παραδείγματα, η δυσκολία ουσιαστικά αντιστοιχεί στην ταχύτητα κίνησης των στόχων, δηλαδή στο χρόνο που προσδιορίζουμε για να μετακινηθούν σε ένα σημείο της οθόνης. Όσο μικρότερος ο χρόνος, τόσο πιο γρήγορα κινούνται οι στόχοι. Πως θα ελέγξουμε τους χρόνους όλων των αντικειμένων; Χρησιμοποιώντας μια μεταβλητή, που θα ονομάσουμε «χρόνος» και η οποία θα εισαχθεί σε όλες τις εντολές **κινήσου ομαλά...δευτερόλεπτα στο x...y...**

Άρα στο δεύτερο σενάριο του σκηνικού, αρχικά θα επιλέξουμε να εμφανιστεί το υπόβαθρο με τις οδηγίες και στη συνέχεια με τη χρήση της εντολής **ρώτησε...και περίμενε** θα ζητήσουμε από το χρήστη να επιλέξει το επίπεδο δυσκολίας προσδιορίζοντας το αντίστοιχο επίπεδο με ένα γράμμα. Ανάλογα με την τιμή της μεταβλητής **απάντηση** θα μπορέσουμε στη συνέχεια να προσδιορίσουμε την τιμή της μεταβλητής «χρόνος». Υπάρχει όμως ένα πρόβλημα. Τι θα συμβεί αν ο χρήστης δώσει ως είσοδο ένα

διαφορετικό χαρακτήρα; Πως θα γνωρίζουμε το επιθυμητό επίπεδο δυσκολίας; Το έργο μας δεν πρέπει να προχωρά αν ο χρήστης δεν εισάγει ένα από τα τρία γράμματα. Αυτό μπορούμε να το επιτύχουμε αν δημιουργήσουμε μια επανάληψη η οποία σταματά μόνο εφόσον το γράμμα εισόδου είναι ένα από τα επιθυμητά. Μελετήστε την επόμενη εντολή:



Η συγκεκριμένη τεχνική χρησιμοποιείται ευρέως. Δηλαδή αν θέλετε η είσοδος του χρήστη να περιοριστεί σε ορισμένες μόνο επιλογές, δεν έχετε παρά να συνδυάσετε μια εντολή επανάληψης με μια εντολή ερωτήματος που συνεχίζει να ζητάει από το χρήστη μια τιμή έως ότου αυτός να δώσει μια από τις επιθυμητές.

Για να εξετάσουμε τη συγκεκριμένη συνθήκη χρειαστήκαμε 2 συνθήκες ...ή... αφού χρειαζόμαστε τον έλεγχο τριών ισοτήτων.

Τώρα, αφού έχουμε καταφέρει να πάρουμε μια ικανοποιητική απάντηση, θα χρησιμοποιήσουμε τρεις φορές την εντολή **εάν..._** για να ορίσουμε την κατάλληλη τιμή της μεταβλητής «χρόνος».



Σε αυτή τη φάση του προγράμματος συνεπώς ο χρήστης έχει επιλέξει επίπεδο και είναι έτοιμος να ξεκινήσει το παιχνίδι. Άρα, πρέπει να ξεκινήσουμε το χρονόμετρο, να μηδενίσουμε το σκορ και να εμφανίσουμε το μονόχρωμο υπόβαθρο, πάνω στο οποίο θα εμφανιστούν οι στόχοι.



Και μας μένει μια ακόμη μόνο δουλειά: να εμφανιστούν και να ξεκινήσουν την κίνησή τους οι στόχοι και να τερματιστεί το παιχνίδι μετά από 60 δευτερόλεπτα. Παρατήστε ότι τα αντικείμενα στόχοι δεν πρέπει να κάνουν τίποτε μέχρι να φτάσουμε σε αυτό το σημείο και πρέπει να σταματήσουν όταν ο χρόνος ξεπεράσει τα 60 δευτερόλεπτα. Πως θα μπορούσαμε να πετύχουμε κάτι τέτοιο στα σενάρια τους; Ο πιο απλός τρόπος είναι να τους στείλει το σκηνικό ένα μήνυμα τη στιγμή που ξεκινά το παιχνίδι και ένα δεύτερο μήνυμα τη στιγμή που τελειώνει, δηλαδή μετά από 60 δευτερόλεπτα. Οι στόχοι δε θα έχουν παρά να χειριστούν αυτά τα δυο μηνύματα. Δείτε το τελευταίο τμήμα του δεύτερου σεναρίου για το σκηνικό:



Όπως βλέπετε, η εντολή **περίμενε ώσπου...** με συνθήκη το χρονόμετρο να μη ξεπεράσει τα 60 δευτερόλεπτα, μπορεί να χρησιμοποιηθεί ως μηχανισμός για να σταλεί το δεύτερο σήμα «end» την κατάλληλη στιγμή. Ολοκληρωμένο πλέον το σενάριο του σκηνικού:



Αντικείμενα-στόχοι

Οι στόχοι είναι τα αντικείμενα πάνω στα οποία θα επιδιώκει ο χρήστης να κάνει κλικ. Στο παράδειγμά μας μπορούμε να εισάγουμε μια σειρά από διαφορετικές μπάλες που βρίσκονται μέσα στο φάκελο things της βιβλιοθήκης αντικειμένων του Scratch! Θα χρειαστείτε αρκετές μπάλες γι' αυτό μη ξεχνάτε ότι μπορείτε να διπλασιάσετε τα αντικείμενά σας (δεξί κλικ πάνω στο αντικείμενο και επιλέγουμε «διπλασιάσε»!).



Οι στόχοι θα πρέπει να έχουν δυο ενδυμασίες: την αρχική ενδυμασία και την ενδυμασία έκρηξης «mroum» την οποία θα δημιουργήσετε στην ζωγραφική! Δεν είναι δύσκολο! Το αποτέλεσμα θα πρέπει να είναι κάτι τέτοιο:



Ποιες είναι οι συμπεριφορές των συγκεκριμένων αντικειμένων;

- ✓ όταν λάβουν το μήνυμα «start» θα πρέπει να ξεκινά η αδιάκοπη και τυχαία κίνησή τους,
- ✓ όταν ο χρήστης πατήσει κλικ πάνω στους στόχους, οι στόχοι θα πρέπει να αλλάζουν εμφάνιση,
- ✓ όταν ξεκινά το έργο θέλουμε τα αντικείμενα-στόχοι να μην φαίνονται,
- ✓ όταν τελειώνει το έργο θέλουμε τα αντικείμενα-στόχοι να εξαφανίζονται και να σταματά η κίνησή τους.

Ελπίζουμε να έχει γίνει φανερό πλέον, ότι αν πριν ξεκινήσουμε να δημιουργούμε το σενάριο ενός αντικειμένου, έχουν αναλύσει προσεκτικά τις συμπεριφορές του, η κατασκευή του διευκολύνεται κατά πολύ.

Επειδή όλες οι συμπεριφορές βασίζονται σε γεγονότα που προκαλούνται είτε από το χρήστη (κλικ), είτε από άλλα αντικείμενα (μήνυματα start, end), είτε από το έργο (πάτημα στην πράσινη σημαία), θα χρειαστεί να δημιουργήσουμε 4 αντίστοιχα σενάρια. Τα δυο τελευταία είναι εύκολα και δεν χρειάζονται επεξήγηση:



Όταν ο παίκτης κάνει κλικ πάνω στο στόχο, δηλαδή πετυχαίνει μια βολή, τότε θα πρέπει

- ✓ να αλλάζει ενδυμασία και να φοράει την ενδυμασία «μπουμ»,
- ✓ να παίζει τον αντίστοιχο ήχο (εισάγουμε τον ήχο Pop από τον φάκελο effects των έτοιμων ήχων στους στόχους μας),
- ✓ να αυξάνει το σκορ,
- ✓ να κρύβεται.

Αυτά ακριβώς εκτελεί το παρακάτω σενάριο



Η εντολή **περίμενε...δευτερόλεπτα**, επιτρέπει στο χρήστη να δει έστω και για λίγο την εμφάνιση έκρηξης του στόχου.

Τέλος, το σενάριο της κίνησης πρέπει να ξεκινά όταν το σκηνικό μεταδώσει το σήμα «start». Γι' αυτό, το σενάριο θα ξεκινάει με την εντολή **όταν λάβω....** Το αντικείμενο θα ξεκινάει πάντα από το αριστερό ή το δεξί άκρο του σκηνικού και θα καταλήγει στο απέναντι. Επίσης, για λόγους συμμετρίας, τα μισά αντικείμενα θα ξεκινούν από τα αριστερά και τα άλλα μισά από τα δεξιά. Για να είναι τυχαία η κίνησή τους θα πρέπει να αλλάζουμε διαρκώς το ύψος, τη συντεταγμένη y δηλαδή, από την οποία ξεκινούν

και τελειώνουν κάθε φορά που διανύουν την οθόνη του Scratch.

Επιπλέον για να ενισχύσουμε το τυχαίο χαρακτήρα της εμφάνισης των στόχων, μπορούμε να προσθέσουμε ένα τυχαίο χρονικό διάστημα πριν εμφανιστεί κάθε αντικείμενο, με τη χρήση της εντολής **περίμενε... δευτερόλεπτα**. Και φυσικά δεν πρέπει να ξεχάσουμε ότι στην εντολή **κινήσου ομαλά... δεύτ. στο x, y** εισάγουμε ως χρόνο τη μεταβλητή «χρονός» ώστε οι μπάλες να κινούνται με διαφορετική ταχύτητα στα διαφορετικά επίπεδα δυσκολίας του παιχνιδιού. Μελετήστε το παρακάτω σενάριο για ένα από τα αντικείμενα-στόχους:



Γιατί στο σενάριο υπάρχει η εντολή **αλλαγή σε ενδυμασία...**; Μη ξεχνάτε ότι μπορεί το αντικείμενό μας να έχει «σκάσει» από το χτύπημα του χρήστη και να φορά συνεπώς την ενδυμασία «έκρηξη». Άρα με την εντολή **αλλαγή σε ενδυμασία...** εξασφαλίζουμε ότι θα φορά την αρχική του ενδυμασία. Μήπως όμως υπάρχει καλύτερο μέρος για να εισάγουμε την εντολή αυτή; Πως θα μπορούσαμε να κάνουμε πιο «αποδοτικό» το σενάριό μας;

Για να αποκτήσουν όλα τα αντικείμενα την ίδια ενδυμασία σύρτε το εικονίδιο της από τη **λίστα ενδυμασιών**, πάνω στα αντικείμενα που βρίσκονται στη **λίστα αντικειμένων**. Με πολύ γρήγορο τρόπο μπορείτε να εισάγετε μια κοινή ενδυμασία σε πολλαπλά αντικείμενα. Το ίδιο ισχύει και για τους ήχους. Αλλά ακριβώς το ίδιο μπορείτε να κάνετε και όταν θέλετε να αντιγράψετε ένα σενάριο σε ένα άλλο αντικείμενο. Δεν έχετε παρά να το σύρτε πάνω στο αντικείμενο που βρίσκεται στη λίστα αντικειμένων

Επιπλέον, όμως, όταν θεωρούμε ότι μια σειρά από αντικείμενα θα έχουν στο πρόγραμμά μας πανομοιότυπη συμπεριφορά, μπορούμε να δημιουργήσουμε μόνο ένα αρχικά, και στη συνέχεια τα άλλα να προκύψουν ως αντίγραφα του πρώτου.

Με όλες τις προηγούμενες οδηγίες μπορείτε να κερδίσετε πολύτιμο χρόνο.

Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Δημιουργήστε διαφορετικού μεγέθους στόχους που δίνουν και διαφορετική βαθμολογία.
- 2) Δημιουργήστε ένα τέταρτο επίπεδο δυσκολίας στην αρχική οθόνη του παιχνιδιού σύμφωνα με το οποίο η δυσκολία του παιχνιδιού θα αυξάνεται με το πέρασμα του χρόνου (π.χ. κάθε 20 δευτερόλεπτα).



15.4 Ο καρχαρίας

[15_π04.sb]

Το επόμενο παιχνίδι που θα αναλύσουμε λαμβάνει χώρο σε υδάτινο περιβάλλον και βασικός ήρωας είναι ένας πεινασμένος καρχαρίας. Ο χρήστης ελέγχει τον καρχαρία με τα βελάκια του πληκτρολογίου με στόχο να τον ταΐσει με τα καημένα ψαράκια που περιπλανώνται στο βυθό της θάλασσας. Πρέπει όμως να αποφύγει αυτά που είναι δηλητηριώδη. Ο χρήστης κερδίζει πόντους ανάλογα με το είδος των ψαριών που τρώει ο καρχαρίας, ενώ αν φάει δηλητηριώδες ψάρι, τότε χάνει μια ζωή.

Ποιο συγκεκριμένα, τα επόμενα ψάρια δίνουν τους εξής βαθμούς:

- ✓ Κίτρινα ψάρια (1 βαθμός).
- ✓ Μοβ ψάρια (3 βαθμοί).
- ✓ Χταπόδι (5 βαθμοί).

Το κόκκινο ψάρι είναι δηλητηριώδες και όταν ακουμπά τον καρχαρία, αφαιρείται μια ζωή από το χρήστη. Κατά την εκκίνηση του παιχνιδιού ο χρήστης διαθέτει 3 ζωές.

Στο παιχνίδι συμμετέχουν και μικρά ψαράκια που δεν αποτελούν τροφή για τον καρχαρία και κάνουν τη ζωή του παίχτη δυσκολότερη. Παρακάτω φαίνεται μια εικόνα του παιχνιδιού



Τα αντικείμενα

Τα αντικείμενα ανιχνεύονται σχετικά εύκολα αφού για κάθε μορφή ζωής μέσα στον υδάτινο κόσμο μας χρειαζόμαστε ένα αντικείμενο. Το αντικείμενο καρχαρίας θα ελέγχεται από το χρήστη ενώ όλα τα υπόλοιπα θα κινούνται από μόνα τους.

Το σκηνικό

Το σκηνικό μας είναι απλό και η μόνη ζητούμενη συμπεριφορά είναι η αναπαραγωγή του ήχου bubbles (από τη βιβλιοθήκη ήχων του Scratch) για να δώσουμε την εντύπωση στο χρήστη ότι βρίσκεται πραγματικά μέσα σε ένα υδάτινο περιβάλλον. Το σκηνικό που έχει επιλεγεί ονομάζεται underwater και βρίσκεται στο φάκελο nature της βιβλιοθήκης υποβάθρων του Scratch. Στο σενάριο του σκηνικού, είναι σημαντικό να θυμηθούμε ότι μέσα στην εντολή επανάληψης για τον ήχο πρέπει να επιλέξουμε την εντολή αναπαραγωγής παίξε ήχο...μέχρι τέλους και όχι την παίξε ήχο... Δοκιμάστε το αντίστροφο και εξηγήστε τι συμβαίνει.



Ψάρια και χταπόδι

Ας μελετήσουμε την εμφάνιση και τη συμπεριφορά των ψαριών και του χταποδιού.

Για το κίτρινο ψάρι θα επιλέξουμε ως βασική ενδυμασία την εικόνα fish3 από τον κατάλογο animals της βιβλιοθήκης αντικειμένων του Scratch. Το κίτρινο ψάρι εμφανίζεται σε τυχαία θέση στο σκηνικό στην αρχή του παιχνιδιού αλλά και αφού φαγωθεί (!). Κινείται συνεχώς με σταθερή ταχύτητα και σε κατεύθυνση που μπορεί να αλλάζει λίγο με τυχαίο τρόπο, ενώ στην περίπτωση που φτάσει στα όρια πρέπει να αναπηδά. Το κίτρινο ψάρι δίνει 1 βαθμό στον καρχαρία όταν το φάει. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα.



Για το μοβ ψάρι επιλέγουμε ως ενδυμασία την εικόνα fish2 από τον κατάλογο animals. Και το μοβ ψάρι εμφανίζεται σε τυχαία θέση στο σκηνικό και κινείται συνεχώς από μόνο του με σταθερή ταχύτητα και σε τυχαίες κατευθύνσεις χώρο. Το μοβ ψάρι δίνει 3 βαθμούς στον καρχαρία. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα.



Για το κόκκινο ψάρι επιλέγουμε ως ενδυμασία την εικόνα fish4 από τον κατάλογο animals. Το κόκκινο ψάρι θα πρέπει να εμφανίζεται αρχικά σε τυχαία θέση στο σκηνικό μας, εκτός όμως από την περιοχή στην οποία βρίσκεται ο καρχαρίας (π.χ. στην αρχή του παιχνιδιού τοποθετείται στο πάνω αριστερό άκρο της οθόνης). Αν ακουμπήσει το κόκκινο ψάρι τον καρχαρία, ο χρήστης χάνει μια ζωή και θα πρέπει να αποφεύγουμε να συμβαίνει κάτι τέτοιο στην αρχή του παιχνιδιού και μετά από κάθε «χάσιμο» ζωής.



Το χταπόδι εμφανίζεται αρχικά στα κάτω όρια της οθόνης σε τυχαία οριζοντίως θέση, ενώ κινείται διαρκώς κατακόρυφα μέχρι τα πάνω όρια. Για να δείξουμε την κίνηση του χταποδιού όταν ανεβαίνει από τον βυθό προς τα πάνω, θα προσπαθήσουμε να φαίνεται ότι σπρώχνει το νερό προς τα κάτω οπότε και δημιουργείται η απαιτούμενη ώθηση. Για το λόγο αυτό θα χρησιμοποιήσουμε ως ενδυμασίες, τις εικόνες octopus1-a και octopus1-b από το φάκελο animals και θα τις εμφανίζουμε εναλλάξ.



Το χταπόδι δίνει 5 βαθμούς στον καρχαρία. Όταν φαγωθεί από τον καρχαρία εξαφανίζεται, και εμφανίζεται μετά από κάποιο τυχαίο χρονικό διάστημα σε άλλη θέση στο κάτω μέρος της οθόνης. Όπως φαίνεται από τις προηγούμενες εικόνες, οι ενδυμασίες δεν έχουν κατεύθυνση ακριβώς προς τα πάνω αλλά προς πάνω-δεξιά και για αυτό θα πρέπει να αλλάξουμε λίγο την κατεύθυνση τους.



Τέλος, υπάρχει και το μικρό ψάρι που δεν αποτελεί τροφή για το καρχαρία. Για τα ψάρια αυτά επιλέγουμε την εικόνα fish1-a. Η συμπεριφορά τους είναι απλή καθώς τριγυρίζουν στον βυθό με σταθερή οριζόντια κατεύθυνση και κινούνται με χαμηλή σταθερή ταχύτητα.

Θα έχετε διαπιστώσει ότι οι εικόνες που τοποθετήσαμε στο παιχνίδι μας για τους διάφορους ήρωες είναι πολύ μεγάλες σε σχέση με μέγεθος της οθόνης του Scratch. Σε αυτό το παράδειγμα θα τις μικρύνουμε προγραμματιστικά χρησιμοποιώντας την εντολή όρισε το μέγεθος σε... % από την παλέτα Όψεις.

- Για τα κίτρινα ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 20%.
- Για τα κόκκινα ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 15%.
- Για τα μοβ ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 20%.
- Για τα μικρά ψάρια θα πρέπει να μειώσουμε το μέγεθος κατά 10%.

Ποια είναι τα σενάρια των προηγούμενων αντικειμένων; Ποιες συμπεριφορές επιδεικνύουν; Θα μπορούσαμε να διακρίνουμε δυο είδη συμπεριφορών:

A) Αρχικοποίηση θέσης και εμφάνισης.

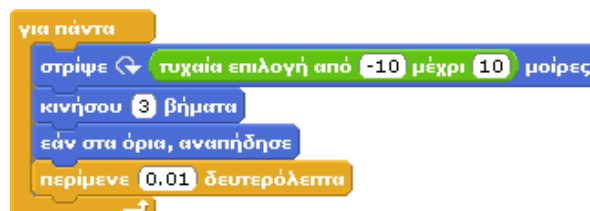
B) Συνεχής κίνηση.

Γ) Αντίδραση όταν έρχονται σε επαφή με τον καρχαρία.

Βήμα πρώτο: Όλα τα ψάρια εμφανίζονται σε τυχαία θέση στο σκηνικό και συνεπώς θα χρησιμοποιήσουμε την εντολή τυχαία επιλογή από...μέχρι.... Ταυτόχρονα, πρέπει να ορίσουμε και το κατάλληλο μέγεθος. Άρα το σενάριό τους ξεκινά κάπως έτσι:



Βήμα δεύτερο: Όλα τα ψάρια κινούνται διαρκώς. Συνεπώς οι εντολές κίνησης θα πρέπει να βρίσκονται μέσα σε μια εντολή επανάληψης για πάντα. Λόγω του ότι το ψάρι καθώς κινείται θα πρέπει να αλλάζει κατευθύνσεις, καλό θα ήταν να χρησιμοποιήσουμε συνδυαστικά τις εντολές κινήσου... βήματα και στρίψε δεξιόστροφα... μοίρες. Θέλουμε το ψάρι να κινείται τυχαία, χωρίς όμως απότομες αλλαγές κατεύθυνσης. Έτσι θα χρησιμοποιήσουμε την εντολή τυχαία επιλογή από...μέχρι... προσδιορίζοντας ένα μικρό εύρος μοιρών (π.χ. -10 μέχρι 10). Για να μειώσουμε την ταχύτητα με την οποία κινείται το ψάρι θα βάλουμε επίσης την εντολή περίμενε... δευτερόλεπτα και να δώσουμε μια πολύ μικρή τιμή καθυστέρησης (π.χ. 0.01). Το σενάριο του ψαριού θα πρέπει να περιέχει τον επόμενο συνδυασμό εντολών:



Ανάλογα με τις ιδιότητες που θέλουμε να έχει κάθε είδος ψαριού μπορούμε να χρησιμοποιήσουμε διαφορετικές τιμές στα πιο πάνω πεδία.

Βήμα τρίτο: Θέλουμε να ελέγχουμε διαρκώς το κατά πόσο το αντικείμενο άγγιξε τον καρχαρία. Για να το καταφέρουμε αυτό θα βάλουμε μέσα στην προηγούμενη εντολή επανάληψης μια εντολή εάν... που θα ελέγχει ακριβώς για αυτή τη συνθήκη. Συγκεκριμένα, συνδυάζουμε την εντολή εάν... με τον αισθητήρα αγγίζει το...



Ποια είναι η συμπεριφορά του ψαριού όταν αγγίζει τον καρχαρία; Καταρχάς, πρέπει να αυξηθεί η τιμή της μεταβλητής σκορ, ανάλογα με το ψάρι για το οποίο δημιουργούμε το σενάριο (π.χ. κίτρινο ψάρι 1 βαθμός). Αμέσως μετά, το αντικείμενο πρέπει να κρυφτεί καθώς θα βρίσκεται στη κοιλιά του καρχαρία. Στη συνέχεια πρέπει να ξαναεμφανιστεί μετά από κάποιο τυχαίο χρονικό διάστημα και, τέλος, να βρεθεί σε μια τυχαία θέση του υδάτινου κόσμου. Για να γίνει όμως πιο πειστικό το παιχνίδι, το ψάρι θα ξανα-εμφανίζεται από τα άκρα της οθόνης ως νέο ψάρι (!) για το χρήστη, αφού το προηγούμενο το έχει φάει ο καρχαρία. Θα προσπαθήσουμε το ψάρι να ξεκινά είτε από το αριστερό είτε από το δεξί άκρο της οθόνης τυχαία. Πως θα προγραμματίσουμε αυτές τις συμπεριφορές;

Για την απόκρυψη του ψαριού θα χρησιμοποιήσουμε την εντολή απόκρυψη. Για την τυχαία καθυστέρηση μέχρι την επανεμφάνιση θα χρησιμοποιούμε το συνδυασμό εντολών τυχαία

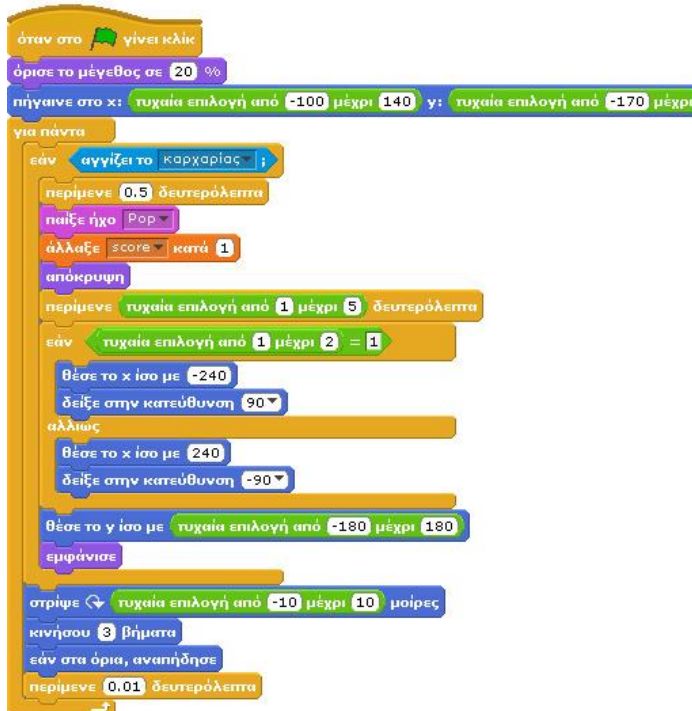
επιλογή από...μέχρι... και περίμενε...δευτερόλεπτα, όπως κάναμε και στο προηγούμενο παιχνίδι.

περίμενε τυχαία επιλογή από 1 μέχρι 5 δευτερόλεπτα

Για την επιλογή τυχαίας θέσης επανεμφάνισης, θα εφαρμόσουμε ένα τρικ. Θα δημιουργήσουμε μια συνθήκη που θα ελέγχει αν το αποτέλεσμα του τελεστή τυχαία επιλογή από...μέχρι... είναι 1 ή 2 και ανάλογα θα προσδιορίζουμε συντεταγμένες εμφάνισης για το ψάρι από το αριστερό ή το δεξί άκρο της οθόνης. Μελετήστε το επόμενο σενάριο:

εάν τυχαία επιλογή από 1 μέχρι 2 = 1
αλλιώς

Στις δυο περιπτώσεις που προκύπτουν από την εντολή εάν...αλλιώς θα προσδιορίσουμε τη θέση x (240 ή -240), τη κατεύθυνση (-90 ή 90 αντίστοιχα) και θα θέσουμε τυχαίο y για τις συντεταγμένες του ψαριού. Έτσι το ολοκληρωμένο σενάριο για το κίτρινο ψάρι είναι το ακόλουθο:



Προσέξτε ότι έχει προστεθεί η εντολή περίμενε 0.5 δευτερόλεπτα αμέσως μετά τη συνθήκη που αφορά το άγγιγμα του καρχαρία. Γιατί; Μήπως για να προλάβει ο καρχαρίας να αλλάξει ενδυμασία, δηλαδή να ανοίξει το στόμα του και στη συνέχεια να φάει το ψάρι;

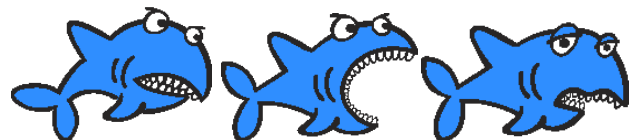
Αυτό το σενάριο είναι πανομοιότυπο για τα μοβ και τα κίτρινα ψάρια. Τα μικρά ψάρια απλά κινούνται και δεν περιέχουν το μπλοκ εντολών που βρίσκεται μέσα στην πρώτη εντολή εάν... ενώ για τα εχθρικά ψάρια, υπάρχουν δυο αλλαγές: α) στο άγγιγμα του καρχαρία αντί να αυξάνουν τη μεταβλητή «σκορ», μειώνουν τη μεταβλητή «ζωές», και β) προσδιορίζουμε ως συντεταγμένες νέας εμφάνισης, θέσεις εκτός της γωνίας στην οποία θα επανεμφανιστεί ο καρχαρίας. Μια λίγο μεγαλύτερη διαφοροποίηση έχει το σενάριο του χταποδιού που κινείται κατακόρυφα:



Μπορείτε να διακρίνετε ότι η κίνηση του χταποδιού γίνεται ουσιαστικά μέσα στην εντολή επανάλαβε ώπου... και διαρκεί μέχρι το χταπόδι να φτάσει στο πάνω όριο της οθόνης. Κατά τη διάρκεια της κίνησης, γίνεται ο γνωστός έλεγχος για το κατά πόσο το χταπόδι άγγιξε το καρχαρία. Ταυτόχρονα, όμως, το αντικείμενο αλλάζει ενδυμασίες και αυξάνει μόνο τη συντεταγμένη y.

Ο καρχαρίας

Ο καρχαρίας εμφανίζεται αρχικά πάνω αριστερά στην οθόνη, ελέγχουμε τη θέση του με τα βελάκια του πληκτρολογίου και μπορεί να φάει διαφορετικά είδη ψαριών ενώ όταν ακουμπήσει κόκκινο ψάρι, τότε του αφαιρείται μια από τις διαθέσιμες ζωές του. Η συμπεριφορά του, όταν ακουμπήσει κάποιο ψάρι, είναι να στρέφεται προς το μέρος του θύματος, να ανοίγει το στόμα του και να το τρώει. Πιο συγκεκριμένα, θα προσδιορίσουμε ως αρχική μορφή του καρχαρία την ενδυμασία shark-b από τη βιβλιοθήκη αντικειμένων του Scratch, ενώ όταν ο καρχαρίας ακουμπήσει ένα ψάρι, στρέφεται προς το μέρος του και παίρνει τη μορφή shark-b.



Στη συνέχεια, το ψάρι πρέπει να εξαφανίζεται και ο καρχαρίας να επανέρχεται στην αρχική του μορφή. Επιπλέον, όταν ο καρχαρίας ακουμπήσει ένα κόκκινο ψάρι τότε θα πρέπει να φορέσει την ενδυμασία απογοήτευσης shark-c, να χάνει το εφέ του χρώματός του κατά 20 (να φαίνεται πιο αδύναμος) και να επιστρέφει στην αρχική του θέση.

Άρα ο καρχαρίας:

Α) Πρέπει να αρχικοποιεί τις μεταβλητές που αφορούν το σκορ και τις ζωές, να αλλάζει μέγεθος και να πηγαίνει σε μια αρχική θέση,

Β) πρέπει διαρκώς να ελέγχει για το αν άγγιξε κάποιο από τα αντικείμενα που το επηρεάζουν, και

Γ) πρέπει να μετακινείται με τα βελάκια του πληκτρολογίου.

Το πρώτο κομμάτι της συμπεριφοράς του μπορούμε να το δημιουργήσουμε εύκολα:



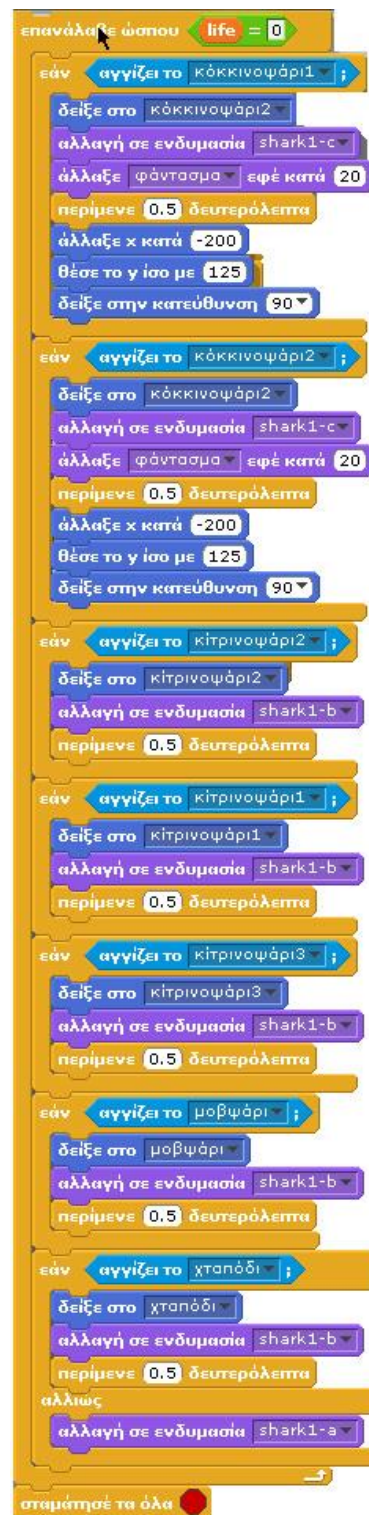
Για το δεύτερη συμπεριφορά του, πρέπει να δημιουργήσουμε μια επανάληψη μέσα στην οποία θα ελέγχουμε το αν άγγιξε άλλα αντικείμενα. Άλλα πόσο θα διαρκεί αυτή η επανάληψη; Μέχρι η μεταβλητή ζωές να πάρει την τιμή 0. Για το λόγο αυτό θα χρησιμοποιήσουμε την εντολή επανάλαβε ώσπου... και μετά από αυτή θα χρησιμοποιήσουμε την εντολή σταμάτησέ τα όλα ώστε να τερματίζεται το παιχνίδι όταν δεν υπάρχουν άλλες διαθέσιμες ζωές. Δηλαδή πρέπει να προσθέσουμε τις εντολές:



Επιπλέον, χρειαζόμαστε τις συνθήκες ελέγχου για το αν άγγιξε άλλο αντικείμενο. Αν άγγιξει άλλο αντικείμενο, ο καρχαρίας πρέπει αλλάξει την κατεύθυνσή του για να δείξει προς αυτό (ώστε όταν ανοίξει το στόμα του να είναι κοντά το ψάρι!), να ανοίξει το στόμα του και να περιμένει λίγο ώστε να προλάβει ο χρήστης να παρακολουθήσει το ανοιγμένο στόμα του. Στην περίπτωση που ακουμπήσει κόκκινο ψάρι, τότε φοράει την ενδυμασία του απογοητευμένου καρχαρία, γίνεται πιο αδύναμος με το εφέ «φάντασμα» και στη συνέχεια μεταφέρεται στην αρχική του θέση. Παρακάτω παρουσιάζονται τα δυο είδη συνθηκών που θα περιληφθούν στην εντολή επανάλαβε ώσπου...



Μια από τις δυο συνθήκες χρειαζόμαστε για όλα τα αντικείμενα. Το δεύτερο τμήμα του σεναρίου εμφανίζεται στην παρακάτω εικόνα:



Τέλος, για το κομμάτι της κίνησης του καρχαρία, δεν μπορούμε να χρησιμοποιήσουμε τις εντολές κίνησης άλλαξε x κατά... και άλλαξε y κατά... καθώς ο καρχαρίας θα φαινόταν να πηγαίνει ακόμη και με την όπισθεν! Χρειαζόμαστε εντολές κίνησης που ενσωματώνουν κατεύθυνση. Επομένως, θα χρησιμοποιήσουμε τις εντολές δείξε στην κατεύθυνση... και κινήσου...βήματα:



Το παιχνίδι μας είναι έτοιμο και είναι πραγματικά απολαυστικό.
Ως προγραμματιστές το δοκιμάσαμε για πολύ ώρα ☺

Προτεινόμενες αλλαγές στο παιχνίδι

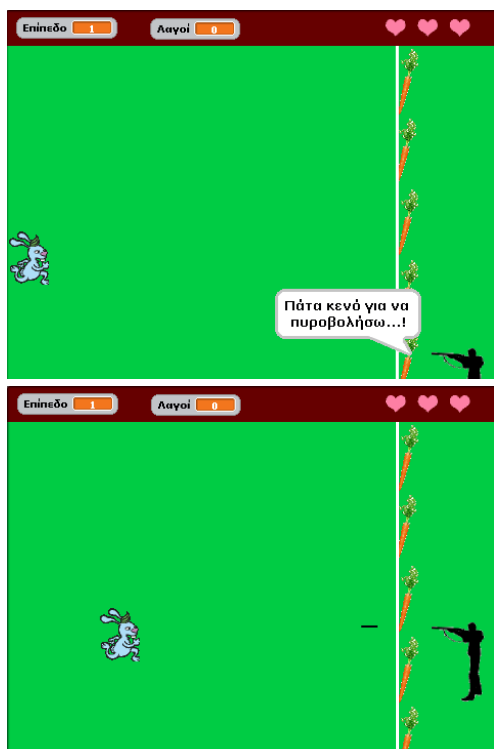
- 1) Δημιουργήστε στο σκηνικό φυσικά εμπόδια με χρώμα τα οποία περιορίζουν τις δυνατότητες κίνησης του καρχαρία. Δεν είναι και μικρός!
- 2) Ζητήστε από το χρήστη να προσδιορίσει στην αρχή του παιχνιδιού το επιθυμητό επίπεδο δυσκολίας, όπως στο προηγούμενο παιχνίδι, και εμφανίστε διαφορετικό αριθμό κόκκινων ψαριών για τα διαφορετικά επίπεδα.
- 3) Δημιουργήστε ένα turbo-boost για τον καρχαρία. Δώστε τη δυνατότητα στο χρήστη με ένα πλήκτρο να μετακινεί το καρχαρία πολύ γρηγορότερα για κάποια δευτερόλεπτα. Το turbo-boost, θα μπορεί να χρησιμοποιηθεί μόνο 2 φορές κατά τη διάρκεια του παιχνιδιού.



15.5 Καροτοκυνηγός

[15_π05.sb]

Το παιχνίδι λαμβάνει χώρα σε ένα κτήμα, όπου στη δεξιά του πλευρά του υπάρχει ένα χωράφι με καρότα τα οποία οριοθετούνται από μια λευκή ευθεία γραμμή αριστερά τους (βλ. επόμενη εικόνα). Το πρόβλημα όμως είναι ότι διάφοροι επιτήδριοι (!) λαγοί θέλουν να φάνε τα καρότα και ο κτηματίας είναι αναγκασμένος να πάρει σκληρά μέτρα. Για αυτό και δεξιά από το χωράφι τοποθετεί ένα κυνηγό (!), που ελέγχεται από το χρήστη και ο οποίος μετακινείται προς τα πάνω ή προς τα κάτω με τα δυο αντίστοιχα βελάκια του πληκτρολογίου. Ο κυνηγός πυροβολεί όταν ο χρήστης πατάει το πλήκτρο κενό (space) με στόχο να σκοτώσει τους λαγούς. Όταν πυροβολεί μία σφαίρα, αυτή πρέπει να εξέρχεται από το όπλο του και να κινείται μέχρι τα αριστερά όρια του σκηνικού ή μέχρι να πετύχει το λαγό ενώ στη συνέχεια θα πρέπει να εξαφανίζεται.



Ο λαγός εμφανίζεται σε κάποια τυχαία θέση στα αριστερά όρια του σκηνικού και κινείται προς τα καρότα φυσικά.

Αν ο κυνηγός πετύχει το λαγό, πριν αυτός αγγίξει τη λευκή γραμμή του χωραφιού, τότε ο λαγός εξαφανίζεται και επανεμφανίζεται στα αριστερά όρια του σκηνικού, συνεχίζοντας την κίνησή του προς το χωράφι, ενώ ταυτόχρονα πρέπει να αυξάνεται η βαθμολογία του χρήστη. Όταν ο χρήστης πετύχει το λαγό αριθμό φορές πολλαπλάσιο του δέκα, η δυσκολία του παιχνιδιού πρέπει να μεγαλώνει και το μέγεθος του λαγού να μειώνεται.

Αν ο λαγός καταφέρει να φτάσει στην άσπρη γραμμή δίπλα από τα καρότα, τότε ο χρήστης χάνει μια ζωή και ο λαγός αρχίζει πάλι την κίνησή του από τα αριστερά όρια του σκηνικού μετά από ένα δευτερόλεπτο. Οι ζωές του χρήστη εμφανίζονται αρχικά με τρεις καρδιές στο πάνω δεξιά μέρος του σκηνικού, και πρέπει αντίστοιχα να μειώνονται κάθε φορά που ο χρήστης χάνει μια ζωή.

Στόχος του παιχνιδιού είναι ο κυνηγός να πετύχει όσο το δυνατόν μεγαλύτερη βαθμολογία, ενώ το παιχνίδι τελειώνει όταν ο κυνηγός χάσει και τις τρεις ζωές του. Στο τέλος του παιχνιδιού πρέπει να εμφανίζεται το μήνυμα «Έχασες...!».

Αντικείμενα

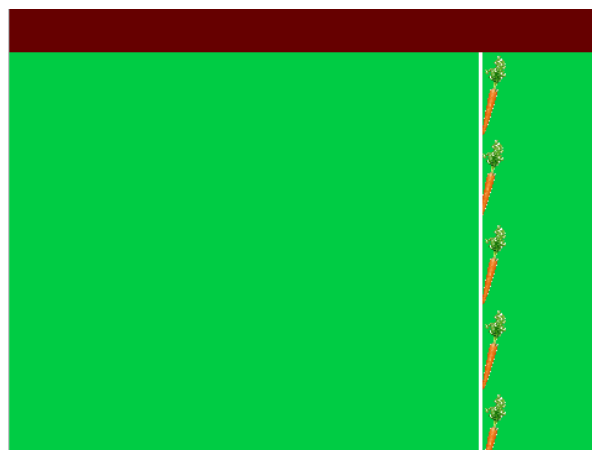
Ποια θα είναι τα αντικείμενα του έργου μας; Ο λαγός και ο κυνηγός είναι τα δυο προφανή. Επιπλέον, για να κινείται η σφαίρα θα πρέπει να δημιουργήσουμε ένα αντίστοιχο αντικείμενο ενώ και οι καρδιές που πρέπει να μειώνονται υποδηλώνουν ένα αντικείμενο με διαφορετικές ενδυμασίες. Τέλος, χρειαζόμαστε ένα αντικείμενο το οποίο θα παρουσιάσει το μήνυμα ολοκλήρωσης του παιχνιδιού.

Μεταβλητές

Πρέπει να αποθηκεύουμε κάπου τη βαθμολογία του παίχτη, τις ζωές του και το επίπεδο του παιχνιδιού. Για αυτό και θα χρειαστούμε τρεις μεταβλητές, με ονόματα «Λαγοί», «Ζωές» και «Επίπεδο», και αρχικές τιμές «0», «3» και «1» αντίστοιχα. Οι μεταβλητές μπορούν να αρχικοποιούνται από το σκηνικό με το κλικ στην πράσινη σημαία.

Σκηνικό

Σημαντικό ρόλο στο πρόβλημά μας διαδραματίζει η λευκή ευθεία γραμμή που οριοθετεί το χωράφι με τα καρότα και την οποία προσπαθεί να αγγίξει ο λαγός. Το σκηνικό μας πρέπει να την περιέχει, οπότε θα χρειαστεί να σχεδιάσουμε εμείς ένα σκηνικό με τη βοήθεια του επεξεργαστή ζωγραφικής του Scratch. Στο σκηνικό μπορούμε να βάλουμε εικόνες από καρότα για το χωράφι, ενώ καλό θα ήταν να υπάρχει χώρος στο πάνω μέρος για να φαίνονται καθαρά οι ζωές, το επίπεδο του παιχνιδιού και η βαθμολογία του κυνηγού. Δε χρειάζεται να προσθέσουμε διαφορετικά υπόβαθρα στο σκηνικό μας, ενώ το μοναδικό σενάριο που θα περιέχει το σκηνικό, είναι η αρχικοποίηση των μεταβλητών του προβλήματος και μια επανάληψη για πάντα, μέσα στην οποία θα αναπαράγεται ένα μουσικό κομμάτι. Προσπαθήστε να δημιουργήσετε ένα σκηνικό παρόμοιο με αυτό που παρουσιάζεται παρακάτω.



Το σενάριο του σκηνικού είναι απλό:



Κυνηγός

Για το αντικείμενο του κυνηγού μπορούμε να βρούμε μια εικόνα από το διαδίκτυο, ενώ δεν είναι απαραίτητο να χρησιμοποιήσουμε επιπρόσθετες ενδυμασίες ή κάποιον ήχο. Ο κυνηγός κινείται από το χρήστη προς τα πάνω και προς τα κάτω, με το πάτημα των πλήκτρων «πάνω βέλος» και «κάτω βέλος».



Άρα, η συμπεριφορά του κυνηγού περιλαμβάνει απλά την κάθετη κίνησή του στην περιοχή που υπάρχουν τα καρότα. Στην έναρξη του παιχνιδιού, ο κυνηγός μας πρέπει να πληροφορήσει το χρήστη ότι πυροβολεί όταν πατηθεί το πλήκτρο «κενό», οπότε χρησιμοποιούμε την εντολή πες...για...δευτερόλεπτα, σε συνδυασμό με τη φράση «Πάτα κενό για να πυροβολήσω...!». Θα χρησιμοποιούμε την εντολή όταν το πλήκτρο πατηθεί για να αλλάξουμε τη κάθετη θέση του κυνηγού, δηλαδή τη συντεταγμένη y . Όμως θα πρέπει να περιορίσουμε την κίνηση του κυνηγού μόνο στην περιοχή του χωραφιού που υπάρχουν τα καρότα. Για αυτό θα πρέπει να κάνουμε χρήση μιας εντολής εάν..., η οποία δε θα επιτρέπει την κίνησή του όταν αυτός φτάσει στα άκρα της επιτρεπόμενης περιοχής κίνησης. Μελετήστε το σενάριο του κυνηγού.



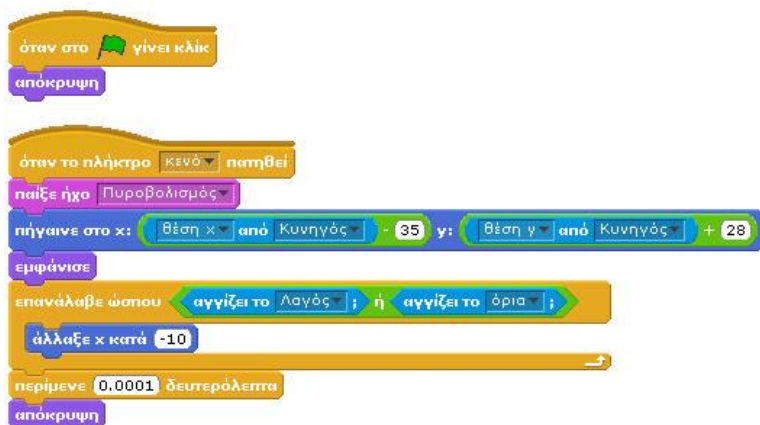
Προκύπτει μια εύλογη ερώτηση: γιατί δε βάζουμε στο αντικείμενο-κυνηγός, τη διαχείριση των ζώων του κυνηγού ή της βαθμολογίας; Η απάντηση είναι γιατί απλά δεν εξαρτώνται από αυτόν και τη συμπεριφορά του. Από τη στιγμή που ο κυνηγός χάνει μια ζωή κάθε φορά που ο λαγός ακουμπά την

άσπρη γραμμή, ως αντικείμενο (ο κυνηγός) δεν εμπλέκεται στο συγκεκριμένο συμβάν. Το ίδιο ισχύει και για κάθε φορά που μια σφαίρα βρίσκει το λαγό και κερδίζεται ένας πόντος. Άρα τη διαχείριση των ζώων και τη βαθμολογία, θα πρέπει να την ανιχνεύσει ο λαγός που με τη σειρά του θα στέλνει τα κατάλληλα μηνύματα.

Σφαίρα

Μπορούμε να σχεδιάσουμε εμείς το αντικείμενο της σφαίρας με μια μικρή ευθεία μαύρη γραμμή, ενώ πρέπει να βρούμε και να εισάγουμε κάποιον ήχο πυροβολισμού. Η σφαίρα εμφανίζεται στο σκηνικό όταν πατηθεί το πλήκτρο «κενό» από το χρήστη και κινείται προς τα αριστερά μέχρι να πετύχει το λαγό ή να φτάσει τα αριστερά όρια του σκηνικού. Όταν μια από τις δυο συνθήκες ικανοποιηθεί, η σφαίρα θα πρέπει να εξαφανιστεί. Σε ποιο σημείο του σκηνικού πρέπει όμως να εμφανιστεί; Στην περιγραφή του προβλήματός μας είπαμε ότι η σφαίρα εξέρχεται από το όπλο του κυνηγού, δηλαδή από το πάνω αριστερό άκρο του αντικειμένου «Κυνηγός». Άρα το αντικείμενο «Σφαίρα» πρέπει να εμφανιστεί αριστερότερα από το «Κυνηγός» (δηλαδή με x μειωμένο σε σχέση με τη θέση x του «Κυνηγός») και λίγο πιο πάνω (δηλαδή y μεγαλύτερο σε σχέση με τη θέση y του «Κυνηγός»). Για να πετύχουμε τη συγκεκριμένη θέση, θα πρέπει να χρησιμοποιήσουμε τον αισθητήρα θέση x/y από μορφή..., ο οποίος δίνει τη δυνατότητα σε ένα αντικείμενο να αξιοποιεί τις συντεταγμένες ενός άλλου αντικειμένου. Στη συγκεκριμένη περίπτωση η σφαίρα εξέρχεται σε θέση σχετική με τις συντεταγμένες του κυνηγού. Ίσως χρειαστεί να δοκιμάσουμε ορισμένες τιμές για να καταφέρουμε να βρούμε την κατάλληλη θέση εμφάνισης της σφαίρας.

Είναι επίσης σαφές ότι πρέπει να χρησιμοποιήσουμε μια εντολή επανάληψης (όπως π.χ. την εντολή επανάλαβε ώσπου), ώστε η κίνηση της σφαίρας να συνεχίζεται μέχρι να αγγίξει το αντικείμενο Λαγός ή τα όρια του σκηνικού. Οπότε, για τη συνθήκη τερματισμού θα χρειαστούμε και τον τελεστή ...ή.... Η σφαίρα θα πρέπει να είναι αρχικά κρυμμένη ενώ θα πρέπει να εμφανίζεται όταν πατηθεί το κενό πλήκτρο.



Λαγός

Σε ό,τι αφορά το αντικείμενο του λαγού, επιλέγουμε να εισάγουμε ως ενδυμασία το αρχείο rabbit1 από το φάκελο Animals, της βιβλιοθήκης ενδυμασιών του Scratch. Επίσης,

Όταν ο λαγός σταματήσει να κινείται (δηλαδή όταν τερματιστεί η επανάληψη), θα έχει αγγίξει τη σφαίρα ή θα έχει αγγίξει τη λευκή γραμμή. Αν τον έχει πετύχει η σφαίρα, θα πρέπει να εξαφανιστεί αφού «χτυπήθηκε», και να αυξηθεί κατά ένα η τιμή της μεταβλητής «Λαγοί», πριν ξανά-εμφανιστεί το αντικείμενο στα αριστερά του σκηνικού. Η νέα θέση του λαγού θα πρέπει φυσικά να είναι μια τυχαία θέση. Τυχαία σε ό,τι αφορά τη κάθετη συντεταγμένη y του αντικειμένου, αφού το x πρέπει να είναι σταθερό και με τιμή μικρότερη από -200 , για να ξεκινάει το αντικείμενο από τα αριστερά όρια του σκηνικού (-217 στην υλοποίηση που παρουσιάζουμε). Για να κάνουμε το y να δέχεται τυχαίες τιμές πρέπει να χρησιμοποιήσουμε την εντολή τυχαία επιλογή από...μέχρι..., με τιμές τα κάθετα όρια του κτήματος (πράσινη περιοχή του σκηνικού).



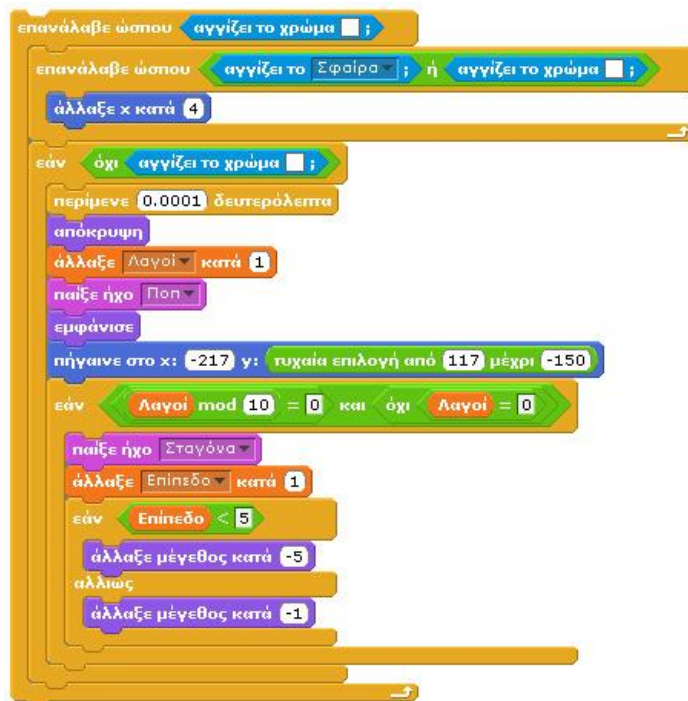
Στη συνέχεια, πρέπει να ελέγξουμε αν πρέπει να αυξηθεί το επίπεδο, δηλαδή αν ο συνολικός αριθμός των λαγών που έχει σκοτώσει ο χρήστης είναι πολλαπλάσιος του 10. Ουσιαστικά, το επίπεδο αυξάνεται κάθε φορά που ο χρήστης πυροβολεί άλλους 10 λαγούς. Ένας αριθμός είναι πολλαπλάσιος του 10, αν το υπόλοιπο της διαίρεσής του με το 10 έχει αποτέλεσμα 0! Σε αυτή την περίπτωση, θα πρέπει να παίζει ο ήχος WaterDrop, να αυξάνεται η τιμή της μεταβλητής «Επίπεδο» κατά ένα και ανάλογα με το επίπεδο, το αντικείμενο «Λαγός» να μειώνει το μέγεθός του. Όσο το επίπεδο του παιχνιδιού είναι μικρότερο από 5, το μέγεθος του λαγού μειώνεται κατά -5, αλλιώς μειώνεται κατά -1.



Αν συνεχίζαμε την υλοποίηση του παιχνιδιού, αφήνοντας το προηγούμενο σενάριο και παίζαμε το παιχνίδι, πιθανόν να μην αντιλαμβανόμασταν ποτέ ένα σημαντικό λάθος. Η διαίρεση mod του 0 με το 10 έχει και αυτή αποτέλεσμα 0. Συνεπώς, αν δεν πετύχουμε τον πρώτο λαγό με σφαίρα και αυτός έφτανε στο χωράφι, η τιμή της μεταβλητής «Λαγοί» θα παρέμενε 0 και θα άλλαζε το επίπεδο χωρίς να έχουμε πετύχει κάποιον λαγό! Οπότε, πρέπει να τροποποιήσουμε τη συνθήκη ελέγχου, προσθέτοντας άλλη μια πρόταση που πρέπει να είναι αληθής για να εκτελεστεί το μπλοκ εντολών.



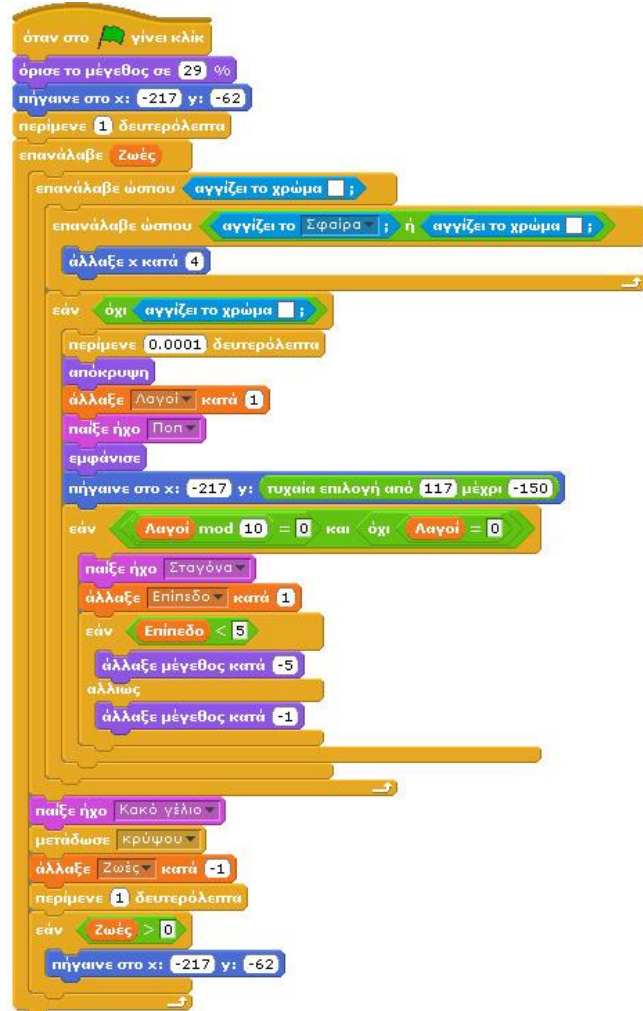
Όλα τα προηγούμενα πρέπει να επαναλαμβάνονται, όσο ο κυνηγός δε καταφέρνει να σκοτώνει το λαγό και όσο ο λαγός δε καταφέρνει να φτάσει στο χωράφι. Αν ο λαγός φτάσει στο χωράφι και αγγίξει το λευκό χρώμα της γραμμής, η επανάληψη θα πρέπει να σταματήσει (προσωρινά). Συμπεριλαμβανουμε, λοιπόν, όλα τα παραπάνω σε μία εντολή **επανάλαβε ώσπου**.



Η έξοδος από την παραπάνω επανάληψη, δείχνει ότι ο λαγός κατάφερε να φτάσει στο χωράφι. Τότε, σύμφωνα με την περιγραφή του προβήματος, ο χρήστης χάνει μία από τις ζωές του. Έτσι, μειώνεται η τιμή της μεταβλητής «Ζωές» κατά 1, παίζει ο ήχος «σατανικού γέλιου» και μεταδίδουμε μήνυμα «κρύψου», ώστε να το λάβει το αντικείμενο «Ζωές» και να αλλάξει ενδυμασία.

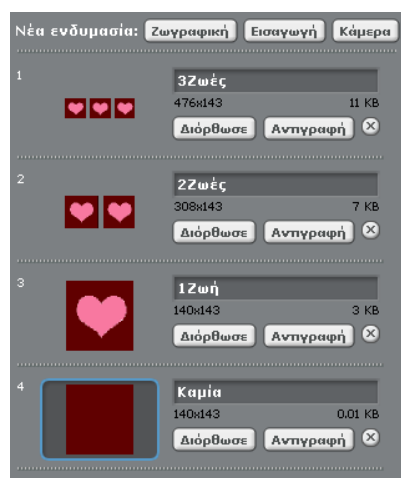


Τελειώνοντας, θέλουμε να έχει τρεις ζωές ο χρήστης, οπότε εσωκλείουμε τον κώδικα που έχουμε περιγράψει ως τώρα σε μια επανάληψη, όπου στη θέση του αριθμού των επαναλήψεων βάζουμε τη μεταβλητή «Ζωές».



Ζωές

Οι ζωές του παίχτη αναπαρίστανται με καρδιές, οπότε κατεβάζουμε από το διαδίκτυο μια εικόνα καρδιάς και δημιουργούμε τέσσερις ενδυμασίες για το αντικείμενό μας με τον επεξεργαστή ζωγραφικής (τρεις καρδιές, δύο καρδιές, μία καρδιά, καμία καρδιά), προσέχοντας όμως να έχουμε σε όλες τις ενδυμασίες μας χρώμα φόντου ίδιο με το χρώμα του πάνω μέρους του σκηνικού.



Στην έναρξη του παιχνιδιού θα πρέπει να φαίνονται τρεις καρδιές (όσες και οι ζωές του παίχτη) και θα πρέπει να μειώνονται μέσω αλλαγής ενδυμασίας όταν ο λαγός καταφέρει να αγγίξει τη λευκή γραμμή. Και πάλι η πληροφορία θα έρχεται μέσω μηνύματος από άλλο αντικείμενο («Λαγός»). Επιπλέον, θα πρέπει

το αντικείμενό μας να στέλνει μήνυμα στο αντικείμενο «Τέλος» για να το ενημερώνει ότι οι ζωές τελείωσαν και ο παίχτης έχασε. Το σενάριο είναι ιδιαίτερα απλό.



Αντικείμενο Τέλος

Το πιο εύκολο προς υλοποίηση αντικείμενο είναι το αντικείμενο «Τέλος». Αυτό είναι κρυμμένο στην έναρξη του παιχνιδιού και εμφανίζεται, παίζοντας τον ήχο «Έχασες», όταν ο κυνηγός χάσει και τις τρεις ζωές του. Για να είναι σε θέση να αντιδρά όταν τελειώσουν οι ζωές του κυνηγού, θα πρέπει να λαμβάνει το μήνυμα «έχασες» που στέλνει το αντικείμενο «Ζωές». Θα χρησιμοποιήσουμε συνεπώς την εντολή όταν λάβω... συμπληρώνοντας όνομα μηνύματος ίδιο με αυτό που θα συμπληρώσουμε στην εντολή μετάδωσε στο σενάριο του αντικειμένου «Ζωές». Ας δώσουμε το όνομα «έχασες» στο μήνυμα.



Στην επόμενη εικόνα φαίνεται η οθόνη του παιχνιδιού όταν εμφανιστεί το αντικείμενο «τέλος».



Προτεινόμενες αλλαγές στο παιχνίδι

- 1) Αλλάξτε το έργο ώστε ο κυνηγός να μπορεί να ρίχνει δυο σφαίρες και ταυτόχρονα να εμφανίζονται 2 λαγοί.
- 2) Αλλάξτε το έργο ώστε να παίζεται ταυτόχρονα από 2 χρήστης που θα ελέγχουν 2 διαφορετικούς κυνηγούς.
- 3) Δώστε επιπλέον όπλα στους κυνηγούς (π.χ. μια πέτρα που πέφτει σε συγκεκριμένη θέση - αν τη στιγμή που πέφτει αγγίζει το

λαγό τότε τον σκοτώνει). Επιπλέον προσδιορίστε πιο σύνθετη κίνηση στους λαγούς (π.χ. με δυνατότητα τυχαίας τροχιάς +/- 10 μοίρες από την οριζόντια κίνηση) ώστε να κάνετε το παιχνίδι πιο ενδιαφέρον και δύσκολο.



64 φοιτητές συνεργάστηκαν σε μια πλατφόρμα Wiki για να δημιουργήσουν το παρόν σύγγραμμα και επιπλέον συνοδευτικό μαθησιακό υλικό. Χρειάστηκαν:

86595 επισκέψεις στο Wiki
664 μηνύματα συζητήσεων
3358 ενημερώσεις ιστοσελίδων
17 ομάδες
1 συντονιστής
πολλή δουλειά....

Σε διάστημα 4 μηνών προετοιμάστηκε η πρώτη έκδοση του βιβλίου, ενώ για αρκετές εβδομάδες μετά, το υλικό μελετήθηκε προσεκτικά για να διορθωθούν λάθη, να βελτιωθούν οι περιγραφές και οι εικόνες, και να δημιουργηθεί η τελική φόρμα του κειμένου.

Το βιβλίο συνοδεύεται από τα αρχεία των παραδειγμάτων, παρουσιάσεις, screencasts, σχετικούς πόρους, κτλ. που μπορείτε να βρείτε στον ιστοτόπο:

<http://www.scratchplay.gr>