Εκπαιδευτικές δραστηριότητες ανάπτυξης εφαρμογών για Android συσκευές

ΠΕΡΙΕΧΟΜΕΝΑ

Μάθημα 1

Στόχοι	
Δραστηριότητα : Ζάρια	4
Δραστηριότητα : Guard Dog	17
Δραστηριότητες για το σπίτι	26
Ενδεικτικές Λύσεις των ασκήσεων	30

Μάθημα 2

Στόχοι	. 32
Δραστηριότητα : Ζάρια – Επέκταση	.33
Δραστηριότητα : GuardDog – Επέκταση	.42
Δραστηριότητα : Κορώνα Γράμματα – Επέκταση	.46
Δραστηριότητες για το σπίτι	. 54
Ενδεικτικές λύσεις των ασκήσεων	.58

Μάθημα 3

Στόχοι	65
Δραστηριότητα : Γκαλερί φωτογραφιών	66
Δραστηριότητα : Αριθμομηχανή	77
Δραστηριότητα : Ζωγραφική με τα δάκτυλα	
Δραστηριότητα : Πυξίδα	96
Δραστηριότητα : Πιάσε τη σημαία	
Δραστηριότητες για το σπίτι	106
Ενδεικτικές λύσεις των ασκήσεων	113

Μάθημα 4

Στόχοι	. 123
Δραστηριότητα : Μαθηματικά για παιδιά (Επέκταση)	.124
Δραστηριότητα : Random Drops	. 128
Δραστηριότητα : Λεξικό	. 135
Δραστηριότητα : Where is my bike?	. 149
Δραστηριότητα : Μπάλες που αναβοσβήνουν (Παιχνίδι Μνήμης)	.157
Δραστηριότητες για το σπίτι	. 169
Ενδεικτικές λύσεις των ασκήσεων	. 177

Παράρτημα

Πηγές	. 187
Προαπαιτούμενα για τη χρήση του App Inventor	. 188
Πακετάρισμα εφαρμογής και διαμοιρασμός της	.192

- Να διακρίνετε το τμήμα της σχεδίασης και να μπορείτε να τοποθετείτε αντικείμενα (Components) πάνω στην περιοχή σχεδίασης της εφαρμογής.
- Να διακρίνετε το τμήμα των εντολών και να τοποθετείτε εντολές των διαφόρων αντικειμένων στην περιοχή συγγραφής του προγράμματος.
- Να μεταφορτώνετε αρχεία πολυμέσων (εικόνας και ήχου) στο περιβάλλον του App Inventor.
- Να διακρίνετε τα διάφορα γεγονότα που μπορούν να προκληθούν σε μια κινητή συσκευή, όπως το άγγιγμα της οθόνης.
- Να συνδέετε τα γεγονότα με συγκεκριμένες ενέργειες.
- Να τοποθετείτε τις ενέργειες αυτές με τη σωστή σειρά εκτέλεσης.
- Να δίνετε τιμές στις ιδιότητες των αντικειμένων που χρησιμοποιείτε στην εφαρμογή.
- Να χρησιμοποιείτε τις εντολές παραγωγής τυχαίων αριθμών.
- Να δημιουργείτε κίνηση (animation) με τη βοήθεια του καμβά.
- Να εκτελείτε την εφαρμογή που δημιουργήσατε στην φορητή σας συσκευή ή στον ενσωματωμένο προσομοιωτή (emulator) του App Inventor.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΑΡΙΑ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή που θα δίνει τη δυνατότητα στον χρήστη της να ρίχνει δύο ζάρια με το πάτημα ενός κουμπιού, όπως ακριβώς και στο τάβλι. Το περιβάλλον της εφαρμογής μας θα μοιάζει με το παρακάτω.

Ζάρια		a e	8:22 pm
	Ρίξε τ	α ζάρια	
	-		

Εικόνα 1 - Περιβάλλον της εφαρμογής Ζάρια

βασικές εννοιές

- Γεγονότα (events).
- Τυχαίοι (random) αριθμοί.
- Επιταχυνσιόμετρο (accelerometer).

Δημιουργίανέου έργου στο App Inventor

Επισκεπτόμαστε τη σελίδα του MIT για το App Inventor <u>http://appinventor.mit.edu/</u> και κάνουμε κλικ στο "Create" στην άνω δεξιά γωνία της σελίδας ...

MIT App Inventor	Home	Blog •	Support •		Create
Follow Us: 👖 💟 🔠 🕅				Google"" Custom Se,	٩

Εικόνα 2- Αρχική σελίδα ΜΙΤ App Inventor

... ή πληκτρολογούμε τη διεύθυνση <u>http://ai2.appinventor.mit.edu/</u> για απευθείας μετάβαση στο περιβάλλον δημιουργίας εφαρμογών. Θα εμφανιστεί η σελίδα για σύνδεση στο λογαριασμό που έχουμε στη Google. Δίνουμε το όνομα χρήστη και το συνθηματικό του λογαριασμού μας και πατάμε "Sign In" (Σύνδεση).

Στη σελίδα που εμφανίζεται, για να δημιουργήσουμε μια νέα εφαρμογή στο περιβάλλον του App Inventor, κάνουμε κλικ στο κουμπί "New Project" (Νέο έργο).

<u>м</u>	IT App Inventor 2 Beta	Project *	Connect •	Build 🔻	Help *	My Projects	Guide
New Proje	ect Delete Project						

Εικόνα 3- Δημιουργία νέου έργου

Θα εμφανιστεί ένα παράθυρο για να καταχωρήσουμε το όνομα που θα δώσουμε στην εφαρμογή μας.

Create new App	o Inventor project
Project name:	
Cancel	ОК

Εικόνα 4 - Ονομάζοντας ένα νέο έργο

Φροντίζουμε το όνομα που θα δώσουμε να αποτελείται από λατινικούς χαρακτήρες και να μην περιέχει κενά ή άλλα σημεία στίξης εκτός από την κάτω παύλα (_). *Προτείνεται το όνομα RollDice*. Στη συνέχεια πατάμε το κουμπί ΟΚ.



Γραφικό Περιβάλλον Εφαρμογής

Αρχικά, βρισκόμαστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη διεπαφή (interface) της εφαρμογής μας, προσθέτοντας τα απαραίτητα αντικείμενα-συστατικά (components) και ορίζοντας ιδιότητες (properties) για αυτά.

Βήμα 2 Μεταφ όρτωση αρχείων στο App Inventor

Θα χρειαστούμε κάποιες εικόνες για τα ζάρια και έναν ήχο για το ρίξιμο των ζαριών.

Αρχικά επισκεπτόμαστε το σύνδεσμο <u>http://tiny.cc/aifiles</u>. Μπαίνουμε στο φάκελο RollDice και κατεβάζουμε στον υπολογιστή μας τις εικόνες των ζαριών, καθώς και τον σχετικό ήχο.

Στη συνέχεια κάνουμε κλικ στο κουμπί "Upload File..." στην περιοχή Media.

Media		
	Upload File	

Εικόνα 5 - Μεταφόρτωση αρχείων

Από το παράθυρο διαλόγου που εμφανίζεται, κάνουμε κλικ στο «Επιλογή Αρχείου» ...



Εικόνα 6 - Επιλογή αρχείου για μεταφόρτωση

... και επιλέγουμε ένα ένα τα αρχεία των εικόνων και του ήχου, όπως περιγράφονται και στον ακόλουθο πίνακα, για να τα μεταφορτώσουμε στο App Inventor.

t	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή
	1.png	Ζάρι με αριθμό 1
	2.png	Ζάρι με αριθμό 2
	3.png	Ζάρι με αριθμό 3
	4.png	Ζάρι με αριθμό 4
	5.png	Ζάρι με αριθμό 5
	6.png	Ζάρι με αριθμό 6
	RollDiceSound.mp3	Ήχος ριξίματος ζαριών

ήμα3 Εισαγωγή αντικειμένων στην περιοχή σχεδίασης

Μέχρι στιγμής, το μοναδικό αντικείμενο της εφαρμογής είναι η οθόνη (Screen1). Προτού προσθέσουμε στο έργο μας τα απαραίτητα αρχεία ήχου και εικόνων, καθώς και επιπλέον αντικείμενα που θα χρειαστούμε, ας τροποποιήσουμε ορισμένες ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties**, ως εξής:

Title : Ζάρια
Background Color : Dark Gray

Στη συνέχεια θα εισάγουμε στο έργο μας τα απαραίτητα αντικείμενα.

Η διαδικασία που πρέπει να ακολουθήσουμε, είναι να επιλέγουμε τα αντικείμενα που μας χρειάζονται από την εκάστοτε ομάδα της παλέτας (Palette) αριστερά και να τα αποθέτουμε στην οθόνη, όπως ενδεικτικά περιγράφεται στην παρακάτω εικόνα.



Εικόνα 7 - Η διαδικασία προσθήκης αντικειμένων στην περιοχή σχεδίασης

Η εφαρμογή που θα φτιάξουμε θα αποτελείται από τα παρακάτω αντικείμενα :

- από την ομάδα User Interface: ένα αντικείμενο Button
- από την ομάδα Layout: ένα αντικείμενο Horizontal Arrangement
- από την ομάδα User Interface: δύο αντικείμενα Image
- από την ομάδα Media: ένα αντικείμενο Sound

Τα τοποθετούμε με την παραπάνω σειρά στην περιοχή σχεδίασης και φροντίζουμε τα αντικείμενα Image να τοποθετηθούν μέσα στο αντικείμενο Horizontal Arrangement.

Όταν ολοκληρωθεί αυτό το βήμα, η περιοχή σχεδίασης πρέπει να είναι όπως παρακάτω:

Display hidden components in Viewer
9:48 🖬
Screen1
Text for Button 1
Non-visible components
Sound1
ικόνα 8 - Αρχική μορφή εφαρμογής Ζάρια



Βήμα4 Καθορισμός Ιδιοτήτων αντικειμένων

Η πρώτη μας κίνηση είναι να μετονομάσουμε τα αντικείμενα που έχουμε εισάγει στην εφαρμογή μας, ώστε να έχουν χαρακτηριστικά ονόματα και όχι button1, image1 κ.λπ. ώστε να τα αναγνωρίζουμε ευκολότερα.

Για να μετονομάσουμε ένα αντικείμενο το επιλέγουμε από την περιοχή των αντικειμένων (Components) και στη συνέχεια κάνουμε κλικ στο κουμπί Rename, όπως φαίνεται στην εικόνα που ακολουθεί.



Τα ονόματα των αντικειμένων πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.



Εικόνα 9 - Μετονομασία Αντικειμένου

Μετονομάζουμε τα αντικείμενα, ώστε όταν ολοκληρωθεί η διαδικασία, να φαίνονται όπως στην παρακάτω εικόνα:

Components				
😑 📃 Screen1				
ThrowButton				
😑 🌅 DiceArea				
🌌 Dice1 Image				
Cice2Image				
⊲ [⊚] ThrowSound				
Rename Delete				

Εικόνα 10 - Μετά την μετονομασία

Στη συνέχεια θα καθορίσουμε κάποιες ιδιότητες που θα έχουν τα αντικείμενα της εφαρμογής.

Για να δούμε τις ιδιότητες κάποιου αντικειμένου καταρχήν **το επιλέγουμε** από την περιοχή Components. Στον πίνακα που ακολουθεί φαίνονται αναλυτικά τα νέα ονόματα και οι ιδιότητες των αντικειμένων.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
UserInterface	Button	ThrowButton	Background Color : Orange
			Font Size : 30
			Shape : rounded
			Text : Ρίξε τα ζάρια
			Width : Fill Parent
			Height : 80 (pixels)
Layout	HorizontalAlignment	DiceArea	AlignHorizontal : Centre
			Width : Fill Parent
			Height : 100 pixels
UserInterface	Image	Dice1Image	Picture : 1.png
UserInterface	Image	Dice2Image	Picture : 1.png
Media	Sound	ThrowSound	Source : RollDiceSound.mp3

Η οθόνη της εφαρμογής, μετά και από τις παραπάνω αλλαγές, έχει διαμορφωθεί όπως στην εικόνα.



Εικόνα 11 - Τελική μορφή εφαρμογής Ζάρια

Συγγραφή Προγράμματος

Blocks

Μέχρι στιγμής, στο παράθυρο σχεδίασης (Designer) έχουμε μεν σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της.

Θα μεταφερθούμε λοιπόν στο παράθυρο συγγραφής προγράμματος Blocks (το κουμπί μετάβασης βρίσκεται πάνω δεξιά) για να συσχετίσουμε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσουμε, προσθέτοντας τις κατάλληλες εντολές.

nove Screen Blocks Components Properties Screen1 Screen1 Display hidden components in View 🗐 🚺 🧐 ThrowDiceButton AlignHorizontal 🖯 🖸 DiceArea Left 💌 Dice1 Image AlignVertical Dice2Image Ρίξε ζάρια Тор 👻 Sound1 BackgroundColor Black BackgroundImage None. CloseScreenAnimation Default -

Μεταβαίνουμε στο τμήμα εντολών της εφαρμογής, κάνοντας κλικ στο κουμπί Blocks.

Βήμα**5** Γεγονός - Άγγιγμα του κουμπιού «Ρίξε ζάρια»

Το ρίξιμο των ζαριών θα ενεργοποιείται όταν ο χρήστης αγγίζει το κουμπί «Ρίξε ζάρια». Θα πρέπει να συνδέσουμε το γεγονός αυτό (το άγγιγμα του κουμπιού) με τις κατάλληλες ενέργειες.

Πρέπει, λοιπόν, να τοποθετήσουμε στην περιοχή εντολών την κατάλληλη εντολή που θα «ανιχνεύει» το άγγιγμα του κουμπιού.

Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή μας πρακτικά συναρμολογούμε πλακίδια εντολών (και όχι μόνο). Για κάθε πλακίδιο που θέλουμε να προσθέσουμε **ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης**, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες εντολές, αναζητούμε και επιλέγουμε το πλακίδιο που χρειαζόμαστε και το σέρνουμε στον χώρο σύνταξης των προγραμμάτων.

Η διαδικασία μοιάζει, λοιπόν, με εκείνη που ακολουθήσαμε για να προσθέσουμε αντικείμενα στην περιοχή σχεδίασης στο Βήμα 3 και περιγράφεται ενδεικτικά στην παρακάτω εικόνα:

Εικόνα 12 - Κουμπί Blocks



Εικόνα 13 - Διαδικασία προσθήκης εντολών για τα αντικείμενα

Στα αριστερά της οθόνης διακρίνουμε τρεις ομάδες πλακιδίων (blocks).



Πρώτα βρίσκουμε τα ενσωματωμένα (Built-in) πλακίδια, την λειτουργία των οποίων θα εξετάσουμε αργότερα.

Στη συνέχεια, βλέπουμε τα πλακίδια που αφορούν τα συγκεκριμένα αντικείμενα που έχουμε προσθέσει στην εφαρμογή μας.

Ακολουθεί η τελευταία ομάδα (Any component), που δε θα μας απασχολήσει εδώ.

Για το σκοπό μας (Όταν πατηθεί το κουμπί ThrowButton να εκτελεστεί κάποια ενέργεια), επιλέγουμε το αντικείμενο ThrowButton από την αριστερή πλευρά της σελίδας, όπου περιλαμβάνονται όλα τα αντικείμενα της εφαρμογής μας. Στη συνέχεια σέρνουμε με το ποντίκι, την εντολή **when [throwButton].click ... do** στο χώρο σύνταξης του προγράμματος, δεξιά.

Βήμα6 Ρίξιμο ζαριού – τυχαίοι αριθμοί

Για να προσομοιώσουμε το ρίξιμο των ζαριών θα πρέπει να παράγουμε 2 τυχαίους αριθμούς, έναν για το κάθε ζάρι και να θέτουμε, ανάλογα με τον αριθμό που θα παραχθεί, και την κατάλληλη εικόνα ζαριού στα αντίστοιχα σημεία.

Αρχικά για να αλλάξουμε την εικόνα που απεικονίζει το αντικείμενο Dice1Image, θα επιλέξουμε το αντικείμενο Dice1Image, για να βρούμε την εντολή **set Dice1Image.Picture to** ...



Εικόνα 15 - Εντολή για την αλλαγή εικόνας

Η εντολή αυτή θα «κουμπώσει» κάτω από την εντολή ενεργοποίησης του γεγονότος που είδαμε προηγουμένως. Το αποτέλεσμα θα είναι όπως παρακάτω.



Εικόνα 16 - Μετά την προσθήκη της εντολή αλλαγής εικόνας

Στη συνέχεια θα πρέπει να προσθέσουμε στην «εγκοπή» της εντολής το όνομα του αρχείου εικόνας που απεικονιστεί. Εδώ πρέπει να προσθέσουμε ένα κείμενο που θα περιέχει το όνομα του αρχείου.

Το ένα τμήμα του ονόματος θα είναι ο αριθμός της εικόνας, τον οποίο θα παράγουμε με τυχαίο τρόπο και το δεύτερο τμήμα του κειμένου θα είναι η επέκτασή του («.png»).

Επομένως θα πρέπει να συνδυάσουμε δύο τμήματα κειμένου. Για το σκοπό αυτό θα χρησιμοποιήσουμε, από την ομάδα **Text**, την εντολή ...



Εικόνα 17 - Εντολή join

... την οποία και θα κουμπώσουμε στην εγκοπή της εντολής αλλαγής εικόνας.



Εικόνα 18 - Μετά την προσθήκη της εντολής join

Στην πρώτη εγκοπή της εντολής θα βάλουμε το όνομα του αρχείου εικόνας και στη δεύτερη εγκοπή την κατάληξή του.

Το όνομα του αρχείου της εικόνας είναι ένας αριθμός από 1 μέχρι 6. Στο σημείο αυτό θα πρέπει να παράγουμε έναν τυχαίο αριθμό σε αυτό το διάστημα που θα τον χρησιμοποιήσουμε σαν όνομα της εικόνας. Για το σκοπό αυτό θα ανατρέξουμε στην ομάδα **Math**, όπου εντοπίζουμε την εντολή ...

٢	random integer	from	ί1	to	L <mark>100</mark>

Εικόνα 19 - Εντολή παραγωγής τυχαίων αριθμών

Θα προσαρμόσουμε τους αριθμούς που παράγονται στο διάστημα 1 - 6 (όσοι είναι και οι αριθμοί των ζαριών μας), αλλάζοντας το δεύτερο όριο τις εντολής από 100 σε 6:



Εικόνα 20 - Προσαρμογή τυχαίων αριθμών στο διάστημα 1 έως 6

Τέλος θα κουμπώσουμε αυτή την εντολή στην 1^η εγκοπή της εντολής join, όπως παρακάτω.



Εικόνα 21 - Μετά την προσθήκη της εντολής τυχαίων αριθμών

Στη 2^η εγκοπή της εντολής join θα προσθέσουμε το κείμενο «.png» που είναι η κατάληξη του ονόματος αρχείου. Για το σκοπό αυτό, θα χρειαστούμε, από την ομάδα **Text**, το πλακίδιο κειμένου που φαίνεται παρακάτω.



Εικόνα 22 - Πλακίδιο κειμένου

Η τελική μορφή του κώδικα μετά την προσθήκη της εντολής κειμένου φαίνεται στο επόμενο σχήμα.

whe	n (ThrowButton -).Click	
do	set Dice1Image . Picture to	i join C random integer from 1 to 6
		ζ " .png "

Εικόνα 23 - Ολοκλήρωση των εντολών για την αλλαγή της 1ης εικόνας με τυχαίο τρόπο



Όταν πατηθεί το κουμπί θέσε την εικόνα του πρώτου ζαριού σε ένα αρχείο με όνομα (τυχαίος ακέραιος αριθμός από 1 μέχρι 6).png

Βήμα **7** Ρίξιμο δεύτερου ζαριού - Διπλασιασμός εντολών

Είναι προφανές ότι θέλουμε να εκτελεστούν ακριβώς οι ίδιες ενέργειες για το αντικείμενο Dice2Image. Αντί να ξανακάνουμε τη διαδικασία που περιεγράφηκε προηγουμένως μπορούμε απλά να διπλασιάσουμε την εντολή **set Dice1Image.Picture to** και να την τροποποιήσουμε κατάλληλα για να λειτουργεί με την εικόνα του 2^{ου} ζαριού.

Κάνουμε δεξί κλικ πάνω στην εντολή που θέλουμε να διπλασιάσουμε και επιλέγουμε την εντολή Duplicate (διπλασιασμός).



Εικόνα 24 - Διπλασιάζοντας εντολές

Το μόνο που χρειάζεται να αλλάξουμε είναι το όνομα του αντικειμένου από Dice1Image σε Dice2Image.

whe	n ThrowButton .Click		
do	set Dice1Image • . Picture • to	🔲 join 🕼	random integer from 1 1 to 1 6
		1	" .png "
	set Dice2Image . Picture to	🔲 join 🖡	random integer from 1 1 to 1 6
		Ļ	" .png) "

Εικόνα 25 - Οι εντολές μετά το διπλασιασμό

μα 8 Αναπαραγωγήήχου ζαριών

Η αναπαραγωγή του ήχου των ζαριών θα ενεργοποιείται και αυτή με το γεγονός του αγγίγματος του κουμπιού «Ρίξε ζάρια».

Για να ξεκινήσει η αναπαραγωγή του ήχου θα χρησιμοποιήσουμε την εντολή ...

call ThrowSound .Play

Εικόνα 26 - Εντολή αναπαραγωγής ήχου

... την οποία και θα κουμπώσουμε κάτω από τις υπόλοιπες.

Το ολοκληρωμένο πρόγραμμα φαίνεται στην εικόνα που ακολουθεί.

whe	n ThrowButton .Click		
do	set Dice1Image . Picture . to	🔲 join 🛿	random integer from 1 1 to 1 6
		1	" [png] "
	set Dice2Image . Picture to	🔲 join 🖡	random integer from 1 1 to 1 6
		1	" (png) "
	call ThrowSound . Play		

Εικόνα 27 - Τελικό πρόγραμμα Ζάρια



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

Προς το παρόν το ρίξιμο των ζαριών ενεργοποιείται με το άγγιγμα του κουμπιού «Ρίξε ζάρια». Προτείνετε έναν εναλλακτικό τρόπο ενεργοποίησης των ζαριών που ΔΕΝ θα περιλαμβάνει το άγγιγμα της οθόνης.



TO ANTIKEIMENO ACCELEROMETER



Οι κινητές συσκευές τύπου tablet ή κινητού τηλεφώνου περιλαμβάνουν μια σειρά από αισθητήρες, δηλαδή από εξαρτήματα που μετρούν διάφορα μεγέθη που αφορούν την κινητή συσκευή.

Ένας τέτοιος αισθητήρας είναι το Accelerometer ή επιταχυνσιόμετρο στα ελληνικά, με τη βοήθεια του οποίου η κινητή συσκευή παίρνει πληροφορίες για το αν επιταχύνεται και σε ποιον άξονα (X,Y,Z) καθώς και για το αν την μετακινούμε πέρα δώθε (κάτι σαν φραπέ δηλαδή ^(C)).

Περισσότερα για το επιταχυνσιόμετρο θα μάθουμε στο Μάθημα 2.

<mark>Βήμα8</mark> Προσθήκη αντικειμένου Accelerometer

Επανερχόμαστε στο τμήμα σχεδίασης της εφαρμογής (design).

Designer Γραφικό Περιβάλλον Εφαρμογής

Από την ομάδα **Sensors** επιλέγουμε και τοποθετούμε στην περιοχή σχεδίασης το αντικείμενο Accelerometer.

Palette				
Us	er Interface			
La	yout			
Me	edia			
Dr	awing and Animation	۱		
Sensors				
	AccelerometerSensor	40		
	BarcodeScanner	?		
0	LocationSensor	•		
0:	NearField	•		
	OrientationSensor	1		
Εικόνα 28 - Επιλογή του επιταχυνσιόμετρου				

Συγγραφή Προγράμματος

Blocks

Στη συνέχεια μεταβαίνουμε στο τμήμα εντολών (Blocks) και επιλέγουμε το κατάλληλο γεγονός «ανίχνευσης» του κουνήματος της συσκευής, που βρίσκεται κάτω από το αντικείμενο AccelerometerSensor1, για να το προσθέσουμε στις εντολές της εφαρμογής.



Εικόνα 29- Εντολή ανίχνευσης κίνησης της συσκευής

Οι εντολές που θα εκτελούνται με την κίνηση της συσκευής θα είναι ακριβώς οι ίδιες με αυτές του αγγίγματος του κουμπιού. Τις πρσθέτουμε, αξιοποιώντας τη λειτουργία του διπλασιασμού εντολών (duplicate).

Μπορείτε να δείτε την τελική μορφή της εφαρμογής μετά την προσθήκη του Accelerometer στο Παράρτημα «<u>Λύσεις των ασκήσεων</u>».

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : GUARD DOG

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα αναπτύξουμε σταδιακά μια εφαρμογή «φύλακα», χρησιμοποιώντας έναν σκύλο που δεν θα επιτρέπει σε κανέναν να πειράξει τη συσκευή μας.





Εικόνα 30 - Στιγμιότυπα της οθόνης του Guard Dog

Συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν κάποιος αγγίζει την οθόνη της συσκευής, ενώ επιπλέον τρέχει προς το σημείο που αγγίζουμε την οθόνη.

βασικές εννοιές

- Καμβάς (canvas).
- Επαφή με οθόνη (touchdown και touchup).
- Κίνηση σε καμβά με βάση συντεταγμένες x, y.

ήμα 1 <mark>Δ</mark>ι

Δημιουργίανέου project

Ξεκινώντας, δημιουργούμε ένα καινούργιο project...



Εικόνα 31 - Δημιουργία ενός νέου project

...το οποίο ονομάζουμε GuardDog.

Designer Γραφικό Περιβάλλον Εφαρμογής

Μέχρι στιγμής, το μοναδικό αντικείμενο της εφαρμογής είναι η οθόνη (Screen1). Προτού προσθέσουμε στο έργο μας τα απαραίτητα αρχεία ήχου και εικόνων, καθώς και επιπλέον αντικείμενα που θα χρειαστούμε, ας τροποποιήσουμε ορισμένες ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties**, ως εξής:

*	επιλέγουμε το αντικείμενο	μεταβάλλουμε τις ιδιότητες	
	Screen1	Scrollable: No Title: Guard Dog	

ήμα 2 Προσθήκη των απαραίτητων αρχείωνπολυμέσων

Ας προσθέσουμε στο project μας όλα τα απαραίτητα αρχεία ήχου και εικόνας που θα χρησιμοποιεί η εφαρμογή μας (<u>http://tiny.cc/aifiles</u>) από το φάκελο GuardDog.

Από το πλαίσιο Media, κάτω δεξιά, επιλέγουμε την εντολή Upload File...

Media		
	Upload File	

Εικόνα 32 - Προσθήκη αρχείων πολυμέσων

...ώστε να εντοπίσουμε και να ανεβάσουμε στο project ένα ένα τα σχετικά αρχεία, όπως περιγράφονται και στον παρακάτω πίνακα:

F	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή	
	Grass.jpg	Το γρασίδι στο οποίο κινείται ο σκύλος	
SleepyDog.png		Η ήρεμη φιγούρα του σκύλου	
	AngryDog.png	Η θυμωμένη φιγούρα του σκύλου	
	Bark.mp3	Ο ήχος του γαβγίσματος	

ΒήμαΞ

Προσθήκη τουκαμβά

Στο βήμα αυτό θα προσθέσουμε το αντικείμενο που θα αποτελέσει το χώρο όπου θα κινείται ο σκύλος και θα καθορίσουμε τις ιδιότητες του χώρου αυτού.



Ο καμβάς (Canvas) είναι μια ορθογώνια επιφάνεια, εντός της οποίας μπορούμε να χειριζόμαστε φιγούρες (Sprites) ή να σχεδιάζουμε αγγίζοντάς την. Προκειμένου, λοιπόν, να ορίσουμε της περιοχή στην οποία θα είναι δυνατή η μετακίνηση της φιγούρας του σκύλου, θα προσθέσουμε στην οθόνη μας έναν καμβά.

Σέρνουμε από την παλέτα αριστερά (Palette), από την ομάδα πλακιδίων Drawing and Animation, το αντικείμενο Canvas, και το αποθέτουμε στην οθόνη.

Υπενθυμίζεται πως είναι σημαντικό (αλλά όχι υποχρεωτικό), να ονομάζουμε τα αντικείμενα που χρησιμοποιούμε με τέτοιο τρόπο, ώστε να τα αναγνωρίζουμε ευκολότερα. Γι' αυτό το λόγο, μετονομάζουμε τον καμβά Canvas1 σε DogCanvas. Τέλος, μεταβάλλουμε τις ιδιότητες του καμβά DogCanvas ως εξής:

\$ από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Drawing and	Canvas	DogCanvas	Width: Fill parent
Ammation			Height: Fill Parent
			BackgroundImage: Grass.jpg

Η επιλογή Fill Parent για τις ιδιότητες Width και Height του καμβά του επιτρέπει να επεκταθεί και να καταλάβει όλο το διαθέσιμο χώρο. Έτσι, ο σκύλος θα μπορεί να κινείται κατά μήκος και κατά πλάτος όλης της οθόνης.

Βήμα4

Προσθήκη τουσκύλου

Στο βήμα αυτό θα προσθέσουμε τη φιγούρα του σκύλου-φύλακα.



Οι φιγούρες (image sprites) τοποθετούνται εντός του καμβά και μπορούν να αλληλεπιδρούν με τον καμβά, με άλλες φιγούρες εντός του καμβά και φυσικά να αντιδρούν στις δικές μας ενέργειες.

Προκειμένου να προσθέσουμε τον σκύλο, σέρνουμε μια φιγούρα (ImageSprite) μέσα στον καμβά και ορίζουμε τις ιδιότητές της, όπως παρακάτω:

\$ από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Drawing and Ani- mation	ImageSprite	DogSprite	Interval: 10 Picture: SleepyDog.png
			Rotates: no

Αν ο σκύλος φαίνεται πολύ μεγάλος, μπορούμε να αλλάξουμε τις διαστάσεις του σε pixel, μέσω των ιδιοτήτων Width και Height.

Βήμα **5** Γάβγισμα

Στο βήμα αυτό θα προσθέσουμε τον ήχο του γαβγίσματος, καθώς και τις πρώτες μας εντολές, ώστε ο σκύλος να γαβγίζει όταν αγγίζουμε την οθόνη.

Για να μπορεί ο σκύλος να γαβγίζει, προσθέτουμε ένα αντικείμενο Sound της ομάδας Media, σέρνοντάς το στην οθόνη. Αυτό θα αναπαράγει τον ήχο, όταν το θελήσουμε. Παρατηρούμε ότι δεν εμφανίζεται εντός της οθόνης, αλλά κάτω από αυτήν, αποτελώντας για τον χρήστη της εφαρμογής ένα μη-ορατό αντικείμενο (non-visible component).

*	από την	μεταφέρουμε το	του δίνουμε το	μεταβάλλουμε τις
	ομάδα	αντικείμενο	όνομα	ιδιότητες
	Media	Sound	DogBarkingSound	Source: Bark.mp3 MinimumInterval: 300

Η ιδιότητα MinimumInterval είναι ο ελάχιστος χρόνος πριν την επανάληψη του ήχου. Αν λοιπόν θέσουμε MinimumInterval: 300 msec (0,3 sec) τότε ο ήχος δεν θα μπορεί να ξαναπαίξει προτού περάσουν τουλάχιστον 0,3 δευτερόλεπτα.



Το αντικείμενο Sound, ακόμα κι αν δεν ορίσουμε συγκεκριμένο αρχείο ήχου για αναπαραγωγή, είναι απαραίτητο, αν θέλουμε η συσκευή να δονείται σαν αποτέλεσμα κάποιας ενέργειας.

Συγγραφή Προγράμματος

Blocks

Μέχρι στιγμής, στο Designer έχουμε σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της.

Μεταβαίνουμε λοιπόν στο Blocks (το κουμπί μετάβασης βρίσκεται πάνω δεξιά) για να συσχετίσουμε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσουμε, προσθέτοντας τις κατάλληλες εντολές.

Υπενθυμίζεται ότι για κάθε πλακίδιο που θέλουμε να προσθέσουμε, **ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης**, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες εντολές, αναζητούμε και επιλέγουμε το πλακίδιο που χρειαζόμαστε και το σέρνουμε στον χώρο σύνταξης των προγραμμάτων (βλ. εικόνα 14).

Στην συγκεκριμένη περίπτωση, θέλουμε να ακούγεται ο ήχος του γαβγίσματος DogBarkingSound κάθε φορά που ο χρήστης αγγίζει τον καμβά DogCanvas.

Επιλέγουμε λοιπόν το αντικείμενο DogCanvas, ανοίγει το αντίστοιχο συρτάρι και μεταφέρουμε το πλακίδιο when DogCanvas.TouchDown.



Εικόνα 33 - Η εντολή ανταπόκρισης στο γεγονός ότι αγγίχθηκε ο καμβάς

Στη συνέχεια επιλέγουμε το αντικείμενο DogBarkingSound, εντοπίζουμε το πλακίδιο call DogBarkingSound.Play και το «κουμπώνουμε» μέσα στο προηγούμενο πλακίδιο.

whe	n DogCanvas 🔻 .TouchDo	own
X	У	
do	call DogBarkingSound 🔻).Play

Εικόνα 34 - Συνδυασμός εντολών για αναπαραγωγή ήχου

Ας διαβάσουμε τα πλακίδια που συναρμολογήσαμε. Δίνουν τις εντολές που είναι απαραίτητες έτσι ώστε η εφαρμογή μας να κάνει αυτό που θέλουμε.



Όταν ο χρήστης αγγίξει τον καμβά DogCanvas, τότε αναπαράγεται ο ήχος DogBarkingSound.

Σημειώστε ότι το πλακίδιο **when DogCanvas.TouchDown** επιστρέφει και δύο νούμερα, τα x και y. Αυτά τα νούμερα προσδιορίζουν που ακριβώς ακούμπησε το δάχτυλό του ο χρήστης στην οθόνη και θα τα χρησιμοποιήσουμε στο Βήμα 7.

Βήμα 6 Αλλαγή της μορφής του σκύλου

Στο βήμα αυτό θα επεκτείνουμε την συμπεριφορά του σκύλου έτσι ώστε να αλλάζει μορφή και να αγριεύει όταν αγγίζουμε είτε τον ίδιο, είτε την οθόνη.

Εδώ θέλουμε η φιγούρα του σκύλου DogSprite να αλλάζει μορφή, όταν ο χρήστης αγγίζει είτε την οθόνη (τον καμβά με το χορτάρι), είτε τον ίδιο το σκύλο και να επανέρχεται όταν η επαφή σταματά.

Στο συγκεκριμένο σενάριο, επειδή όταν κάποιος αγγίξει το σκύλο (ο οποίος βρίσκεται εντός του καμβά) αγγίζει αναγκαστικά και τον ίδιο τον καμβά, αρκεί να χρησιμοποιήσουμε σαν αφορμή για της αλλαγής της μορφής του σκύλου, το γεγονός της επαφής με τον καμβά.

Θα χρειαστεί, λοιπόν, να συνδυάσουμε κάποιες εντολές από το συρτάρι που αφορά στον σκύλο DogSprite με το ήδη υπάρχον γεγονός αγγίγματος του καμβά DogCanvas.

Εντοπίζουμε και συναρμολογούμε κάτω από το ήδη υπάρχον σχετικό τμήμα when DogCanvas.Touchdown, το πλακίδιο set DogSprite.Picture to ... όπως παρακάτω:



Εικόνα 35 - Μετά και από την προσθήκη της εντολής ορισμού εικόνας

Παρατηρούμε ότι η εντολή μας είναι ελλιπής. Πώς θα προσδιορίσουμε την εικόνα που θα έχει η φιγούρα DogSprite όταν αγγίζει κάποιος τον καμβά; Θα πρέπει να παρέχουμε το αντίστοιχο όνομα αρχείου. Ανατρέχουμε λοιπόν στην ενσωματωμένη ομάδα πλακιδίων Text και επιλέγουμε το πλακίδιο του κενού κειμένου...



Εικόνα 36 - Το πλακίδιο για την εισαγωγή κειμένου

...στο οποίο συμπληρώνουμε το όνομα του αρχείου εικόνας "AngryDog.png" και το προσθέτουμε στις εντολές μας.

when DogCanvas . TouchDown				
X	У			
do	call DogBarkingSound .Play			
	set DogSprite 🔹 . Picture 🔹 to 🕨 "AngryDog.png "			

Εικόνα 37 - Το ολοκληρωμένο πακέτο εντολών

Αναλόγως προσδιορίζουμε την εικόνα στην οποία θα επανέρχεται ο σκύλος όταν σταματάει η επαφή.



Εικόνα 38 - Οι αντίστοιχες εντολές για την επαναφορά της αρχικής εικόνας



Όταν ο χρήστης αγγίξει τον καμβά DogCanvas, τότε ακούγεται ο ήχος Dog-BarkingSound και η φιγούρα του σκύλου παίρνει την μορφή AngryDog.png. Όταν σταματήσει η επαφή με τον καμβά DogCanvas τότε η φιγούρα του σκύλου παίρνει την μορφή SleepyDog.png.

Βήμα**7**

Προσοχή στα δάχτυλα

Στο βήμα αυτό θα κάνουμε τον φύλακά μας λίγο πιο επιθετικό. Όταν ο χρήστης αγγίζει την οθόνη τότε ο σκύλος θα κινείται προς το σημείο επαφής.

Πιο συγκεκριμένα, εδώ θέλουμε όταν ο χρήστης αγγίζει τον καμβά DogCanvas η φιγούρα DogSpite να στρέφεται και να κινείται προς το σημείο επαφής.



Διαθέτουμε ήδη από το προηγούμενο βήμα το when DogCanvas.TouchDown, το οποίο προσδιορίζει και το σημείο στο οποίο έγινε η επαφή, δηλαδή την οριζόντια απόσταση x από το αριστερό άκρο του καμβά και την κάθετη απόσταση y από το πάνω άκρο του καμβά.

Οι αριθμοί x και y ονομάζονται και συντεταγμένες του σημείου επαφής.

Στο βήμα αυτό θα χρησιμοποιήσουμε τις συντεταγμένες x και y όπου έγινε η επαφή, για να καθορίσουμε προς τα που πρέπει να στραφεί ο σκύλος.

Αρχικά, προσθέτουμε το πλακίδιο call DogSprite.PointInDirection, το οποίο στρέφει τη φιγούρα προς ένα συγκεκριμένο σημείο.



Εικόνα 39 - Προσθήκη της εντολής για στροφή της φιγούρας προς κάποια κατεύθυνση

Για να πάρουμε την τιμή της x στην οποία έγινε η επαφή αφήνουμε το δείκτη του ποντικιού πάνω από την ετικέτα x κι εμφανίζεται το πλακίδιο get x...



Εικόνα 40 - Εμφάνιση επιπλέον πλακιδίων μέσω των ετικετών

...το οποίο σέρνουμε και κουμπώνουμε στο x της PointInDirection. Το ίδιο κάνουμε και για την y.

whe	n DogCanvas 🔪 .TouchDown
X	y y
do	call DogBarkingSound 🔪 .Play
	set DogSprite . Picture to (AngryDog.png "
	call DogSprite .PointInDirection
	x (get x ·
	y (get y y

Εικόνα 41 - Το ολοκληρωμένο σετ εντολών στροφής του σκύλου

Τώρα απομένει να ασχοληθούμε με την κίνηση του σκύλου. Θα πρέπει, όταν ο χρήστης ακουμπά τον καμβά, να τροποποιήσουμε την ταχύτητά του με το πλακίδιο **set DogSprite.Speed**.

set	DogSprite 🔻).	Speed v	to 🖡

Εικόνα 42 - Εντολή ορισμού ταχύτητας της φιγούρας

Χρειάζεται ένας αριθμός για να προσδιορίσουμε την ταχύτητα. Ανατρέχουμε λοιπόν στην ενσωματωμένη ομάδα πλακιδίων Math κι επιλέγουμε το πλακίδιο με τον αριθμό 0...



Εικόνα 43 - Το πλακίδιο εισαγωγής αριθμού

...τον οποίο τροποποιούμε σε 40 και κουμπώνουμε στην set DogSprite.Speed.



Εικόνα 44 - Η εντολή καθορισμού της ταχύτητας με συγκεκριμένη τιμή

Εισάγουμε πλέον ολόκληρη την εντολή στην when DogCanvas.TouchDown.







Όταν ο χρήστης αγγίξει τον καμβά DogCanvas, τότε αναπαράγεται ο ήχος DogBarkingSound, η φιγούρα DogSprite στρέφεται προς το σημείο επαφής (με συντεταγμένες x και y) και κινείται με ταχύτητα 40. Με παρόμοιο τρόπο, μηδενίζουμε την ταχύτητα του σκύλου όταν το δάχτυλο του χρήστη δεν είναι πια σε επαφή με την οθόνη, συμπληρώνοντας το αντίστοιχο τμήμα.



Εικόνα 46 - Οι εντολές που εκτελούνται όταν παύει η επαφή με την οθόνη



Όταν ο χρήστης παύει να αγγίζει τον καμβά DogCanvas, η φιγούρα του σκύλου DogSprite παίρνει τη μορφή SleepyDog.png και ακινητοποιείται.

Ολοκληρωμένη, η εφαρμογή μας αποτελείται από τις παρακάτω εντολές:



Εικόνα 47 - Το ολοκληρωμένο σετ εντολών



Η εφαρμογή μας είναι έτοιμη!

Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

Όπως αναφέρθηκε και στο βήμα 5, το αντικείμενο sound, ακόμα κι αν δεν έχουμε προσθέσει κάποιο αρχείο ήχου στην εφαρμογή μας, είναι χρήσιμο αν θέλουμε η συσκευή να δονείται αντιδρώντας σε ενέργειές μας.

Προσθέστε τις απαραίτητες εντολές, ώστε όταν κάποιος αγγίζει το σκύλο, πέραν όλων των άλλων που συμβαίνουν ήδη, η συσκευή να δονείται για χρονικό διάστημα της αρεσκείας σας.

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Επισκεφθείτε το Google Play Store και συγκεκριμένα τη διεύθυνση:

https://play.google.com/store/apps/category/EDUCATION

Θα σας εμφανίσει μια σειρά από εφαρμογές για κινητές συσκευές στην κατηγορία «Εκπαίδευση». Καταγράψτε στον παρακάτω πίνακα μια εφαρμογή της συγκεκριμένης κατηγορίας που σας τράβηξε το ενδιαφέρον. Περιγράψτε τι ακριβώς σας προσέλκυσε την προσοχή, όπως στο παράδειγμα. Αν την εγκαταστήσετε, καταγράψτε επίσης αν εντοπίσατε κάποια αρνητικά σημεία στην εφαρμογή.

Τίτλος	Εικόνα	Περιγραφή	Λόγοι προσοχής	Αρνητικά
Solar System 3D		Μια 3-D ξενά- γηση στο η- λιακό μας σύ- στημα.	Εντυπωσιακά γραφικά. Χρήσιμες πλη- ροφορίες.	Δεν επιτρέπει τον κατακό- ρυφο προσανατολισμό του τηλεφώνου.

2. Κατηγοριοποιήστε τις παρακάτω εντολές σε Γεγονότα και Ενέργειες.

set Dice1Image ▼ . Width ▼ to ▶	call ThrowSound .Stop	when Screen1 .Initialize
when ThrowButton .LongClick do	initialize global time to 🖡	🕻 🙆 make a list 🕻

3. Αναπτύξτε μια εφαρμογή που θα προσομοιώνει το ρίξιμο ενός κέρματος (κορώνα ή γράμματα). Το παιχνίδι θα αποτελείται από ένα κουμπί που θα δείχνει αρχικά τη μια πλευρά του νομίσματος. Όταν ο χρήστης αγγίξει στο κουμπί – νόμισμα θα αναπαράγεται ο ήχος ρίψης του νομίσματος και θα επιλέγεται με τυχαίο τρόπο η νέα του εικόνα (η εικόνα της κορώνας ή η εικόνα των γραμμάτων).

Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <u>http://tiny.cc/aifiles</u> και το φάκελο CoinFlip.



Εικόνα 48 - Εφαρμογή Κορώνα - Γράμματα

4. Αναπτύξτε μια εφαρμογή που θα δίνει τη δυνατότητα στον χρήστη της να παίζει πιάνο. Η εφαρμογή θα αποτελείται από 7 κουμπιά - νότες (Ντο, Ρε, Μι, Φα, Σολ, Λα, Σι). Με το άγγιγμα κάθε κουμπιού θα αναπαράγεται ο ήχος της αντίστοιχης νότας.

Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <u>http://tiny.cc/aifiles</u> και το φάκελο Piano.



Εικόνα 49 - Η εφαρμογή Πιάνο



Αν χρησιμοποιήσετε το αντικείμενο Sound, θα πρέπει να κάνετε προφόρτωση των ήχων, δηλαδή να ορίσετε κατά την εκκίνηση της εφαρμογής έναν έναν τους ήχους σαν πηγές του αντικειμένου Sound. Με αυτό τον τρόπο οι ήχοι φορτώνονται όλοι κατά την εκκίνηση της εφαρμογής με αποτέλεσμα, στη συνέχεια, το πιάνο να λειτουργεί χωρίς καθυστερήσεις. Για να το πετύχετε αυτό, μπορείτε να χρησιμοποιήσετε το παρακάτω μπλοκ εντολών.

when Screen1 .Initialize						
do	set	NotesSound . Source to ("c.mp3"				
	set	NotesSound . Source to d.mp3 "				
	set	NotesSound . Source to f e.mp3 "				
	set	NotesSound . Source to f.mp3				
	set	NotesSound . Source to (g.mp3)				
	set	NotesSound . Source to 4 a.mp3				
	set	NotesSound 🔹 . Source 🔹 to 🌔 " (b.mp3) "				

Εικόνα 50 - Προφόρτωση ήχων στο περιβάλλον του App Inventor

Εναλλακτικά μπορείτε να χρησιμοποιήσετε το αντικείμενο Player από την ομάδα Media αντί για το Sound, το οποίο λειτουργεί με διαφορετικό τρόπο και δεν απαιτείται να προφορτώσετε τα αρχεία ήχου.

5. Αναπτύξτε μια εφαρμογή που θα λειτουργεί σαν κουμπί πανικού. Η εφαρμογή θα αποτελείται από δύο κουμπιά.



Εικόνα 51 - Μια εκδοχή της εφαρμογής Κουμπί Πανικού

Το πρώτο θα είναι το κουμπί πανικού. Όταν ο χρήστης αγγίζει το κουμπί πανικού θα ξεκινάει ο ήχος μιας σειρήνας, ο οποίος θα αναπαράγεται ξανά και ξανά. Ταυτόχρονα, η συσκευή θα στέλνει ένα μήνυμα SMS σε έναν προεπιλεγμένο αριθμό για να ζητάει βοήθεια. Το κουμπί πανικού θα απενεργοποιείται προσωρινά (ιδιότητα enabled), ώστε να είναι ορατό, αλλά ο χρήστης να μην μπορεί να το ξαναπατήσει.

Το δεύτερο κουμπί (stop) θα τερματίζει τον ήχο της σειρήνας και θα ενεργοποιεί εκ νέου το κουμπί πανικού, ώστε ο χρήστης να μπορεί να το ξαναπατήσει.

Για την αποστολή του SMS θα χρειαστείτε το αντικείμενο **Texting** που βρίσκεται στην ομάδα **Social**.



Εικόνα 52 - Ομάδα εντολών Social

Για να καθορίσετε το κείμενο του μηνύματος και τον τηλεφωνικό αριθμό όπου θα αποστέλλεται, θα χρειαστεί να αλλάξετε τις ιδιότητες Message (Μήνυμα) και PhoneNumber (Τηλεφωνικός αριθμός) του αντικειμένου Texting, αντίστοιχα.

Για να στείλετε το μήνυμα θα χρησιμοποιήσετε την εντολή που φαίνεται παρακάτω.



Εικόνα 53 - Η εντολή για την αποστολή SMS



Για την αναπαραγωγή του ήχου καλύτερα να χρησιμοποιήσετε το αντικείμενο Player από την ομάδα Media και όχι το αντικείμενο Sound.

Το αντικείμενο Player έχει περισσότερες δυνατότητες, όπως είναι η αναπαραγωγή βίντεο, η ρύθμιση της έντασης του ήχου και η δυνατότητα παύσης. Επίσης έχει την ιδιότητα Loop, δηλαδή την αυτόματη επανεκκίνηση του ηχητικού κομματιού όταν αυτό ολοκληρωθεί.

Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <u>http://tiny.cc/aifiles</u> και το φάκελο Panic Button.

ΕΝΔΕΙΚΤΙΚΕΣ ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ

ZAPIA – ΠΡΟΣΘΗΚΗ ACCELEROMETER

when ThrowButton .Click
do set Dice1Image . Picture to L i join random integer from 1 to 6
(".png "
set Dice2Image • . Picture • to L 🖸 join L random integer from L 1 to L 6
(⁴ ".png "
call ThrowSound .Play
when AccelerometerSensor1 . Shaking
do set Dice1Image . Picture to L i join random integer from 1 to 6
(png "
set Dice2Image . Picture to () join (random integer from () to (6
(png "
call ThrowSound .Play

κορωνα – γραμματα



ΚΟΥΜΠΙ ΠΑΝΙΚΟΥ



ΠΙΑΝΟ

when CButton . Click	when AButton Click
do set NotesSound . Source to (C.mp3)	do set NotesSound C. Source to (a.mp3)
call NotesSound .Play	call (NotesSound C.Play
when DButton . Click	when BButton . Click
do set NotesSound . Source to (d.mp3	do set NotesSound . Source to (b.mp3 :
call NotesSound . Play	call NotesSound . Play
when FButton Click	when EButton .Click
do set NotesSound . Source to ff.mp3	do set NotesSound .Source to fe.mp3
call NotesSound .Play	call NotesSound .Play
when GButton . Click do set NotesSound . Source to g.mp3 call NotesSound .Play	when Screen1 Initialize do set NotesSound Source to C.mp3 set NotesSound Source

GUARD DOG



στοχοι

- Να δημιουργείτε μεταβλητές και να τους δίνετε αρχικές τιμές.
- Να αλλάζετε την τιμή των μεταβλητών μέσα στην εφαρμογή.
- Να χρησιμοποιείτε μεταβλητές για να υπολογίσετε το πλήθος κάποιων συμβάντων (μετρητές).
- Να χρησιμοποιείτε τις λογικές τιμές True/False (Αληθής/Ψευδής).
- Να ελέγχετε την τιμή λογικών προτάσεων για να παίρνετε αποφάσεις με την εντολή If (Av).
- Να χρησιμοποιείτε το αντικείμενο Clock (ρολόι) για την παραγωγή animation κινούμενης εικόνας.
- Να χρησιμοποιείτε το αντικείμενο Clock (ρολόι) για την εκτέλεση ενεργειών ανά τακτά χρονικά διαστήματα.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΑΡΙΑ – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

Θα επεκτείνουμε την εφαρμογή ζάρια που αναπτύξαμε στο προηγούμενο μάθημα ώστε να εμφανίζει μήνυμα επιτυχίας όταν ο χρήστης φέρει εξάρες και να κρατάει σκορ, δηλαδή να ενημερώνει τον χρήστη πόσες φορές έχει πετύχει εξάρες.

βασικές εννοιές

- Μεταβλητές (variables).
- Λογικές συνθήκες.
- Εντολή ελέγχου Αν ... τότε ... (If ... then ...).

Βήμα1

Άνοιγμα του έργου «Ζάρια»

Αφού μεταβείτε στη σελίδα του <u>App Inventor</u> και συνδεθείτε με τα στοιχεία του λογαριασμού σας, ανοίξτε το έργο με τα ζάρια που είχατε δημιουργήσει στο προηγούμενο μάθημα.

3ήμα2 Προσθήκη ετικέτας με μήνυμα επιτυχίας

Στο βήμα 2 θα προσθέσουμε μια ετικέτα κειμένου που θα ενημερώνει το χρήστη για το αν κέρδισε.



Οι ετικέτες κειμένου (text labels) είναι χρήσιμες για να εμφανίζουμε μηνύματα κειμένου σε διάφορα σημεία της οθόνης της εφαρμογής.

Designer

Γραφικό Περιβάλλον Εφαρμογής

Καταρχήν θα προσθέσουμε στο γραφικό περιβάλλον της εφαρμογής ένα αντικείμενο **Label** (ετικέτα) από την ομάδα αντικειμένων User Interface.

\$	από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
	UserInterface	Label	ResultLabel	Font Size : 40
				Text Alignment : center
				Text :
				Width : Fill Parent

Βήμα3 Προσθήκη ετικέτας γιατο σκορ

Θα προσθέσουμε ακόμα μια ετικέτα που θα ενημερώνει τον χρήστη για το πόσες φορές έχει φέρει εξάρες.

*	από την ομάδα	μεταφ αντικ	έρουμε το ε έμενο	του δ όνοι	ίνουμε το ια	μεταβάλλουμε τις ιδιότητες
	UserInterface	Label		Score	Label	Font Size : 20
						TextColor : White
						Text : Σкор : 0
						Width : Fill Parent
Συγγραφή Προγράμματος		τος	Blocks			

Η ΕΝΝΟΙΑ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ

Για να μπορέσουμε να συγκρίνουμε το αποτέλεσμα των δύο ζαριών θα αποθηκεύσουμε τους δύο τυχαίους αριθμούς που παράγονται από την εντολή **random integer from...**



Για το σκοπό αυτό θα χρειαστούμε μια μεταβλητή, δηλαδή ένα χώρο μνήμης στον υπολογιστή μας που μπορεί να αποθηκεύσει μια τιμή. Η αναφορά σε αυτόν τον χώρο γίνεται με ένα συμβολικό όνομα που επιλέγουμε εμείς.

Μπορείτε να φανταστείτε την μεταβλητή σαν ένα πολύ μικρό σάκο που χωράει μονάχα ένα αντικείμενο. Μπορούμε να μεταβάλλουμε το περιεχόμενο του σάκου, δηλαδή να αλλάξουμε την τιμή μιας μεταβλητής, αντικαθιστώντας πρακτικά το προηγούμενο αντικείμενο που έχει, με το νέο αντικείμενο που θα τοποθετήσουμε.

Παρακάτω φαίνεται ένα τέτοιο παράδειγμα. Έστω ότι έχουμε μια μεταβλητή με όνομα σκορ στην οποία δίνουμε αρχικά την τιμή 0. Στη συνέχεια της δίνουμε την τιμή 50, οπότε η προηγούμενη τιμή της (δηλαδή το 0) αντικαθίσταται από τη νέα τιμή.



Εικόνα 54 - Η έννοια της μεταβλητής

3ήμα4 Δημιουργίαμεταβλητών για τα ζάρια

Στην εφαρμογή μας θα χρειαστούμε δύο μεταβλητές, κάθε μια για να αποθηκεύσει τον αριθμό του αντίστοιχου ζαριού.

Για να εισάγουμε μια μεταβλητή στην εφαρμογή μας, κάνουμε κλικ στην ομάδα Variables (Μεταβλητές) και επιλέγουμε την εντολή initialize global [name] to ..., όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 55 - Εντολές στην ομάδα Variables

Στο πεδίο [name] της εντολής γράφουμε το όνομα που θέλουμε να δώσουμε σε κάθε μεταβλητή. Στην εφαρμογή μας θα ονομάσουμε την πρώτη μεταβλητή dice1 και τη δεύτερη μεταβλητή dice2, όπως στην παρακάτω εικόνα. Τέλος, στο αντίστοιχο κενό θα κουμπώσουμε τον αριθμό 0 (από την ομάδα Math), που θα αποτελέσει την αρχική, δηλαδή την πρώτη τιμή των μεταβλητών.



Εικόνα 56 - Δίνοντας αρχικές τιμές σε μεταβλητές



Δημιούργησε μια μεταβλητή με όνομα dice1 και εκχώρησε σαν αρχική τιμή της μεταβλητής αυτής το 0.

Δημιούργησε μια μεταβλητή με όνομα dice2 και εκχώρησε σαν αρχική τιμή της μεταβλητής αυτής το 0.

Βήμα 5 Ανάθεση αριθμών ζαριών στις μεταβλητές

Για να αποθηκεύσουμε τους τυχαίους αριθμούς στις μεταβλητές που δημιουργήσαμε θα πρέπει να τροποποιήσουμε το τμήμα των εντολών που ενεργοποιείται με το άγγιγμα του κουμπιού Ρίξε Ζάρια.



Εικόνα 57 - Το αρχικό πρόγραμμα της εφαρμογής Ζάρια

Η εντολή για την παραγωγή των τυχαίων αριθμών θα κουμπώσει με την εντολή ανάθεσης τιμής σε μεταβλητή **set [] to...**, όπως φαίνεται παρακάτω.

set global di	ce1 to C random integer from (1) to (6
Εικόνα 58 - Ανάθεσ	η τιμής σε μεταβλητή
Q	Θέσε την τιμή της μεταβλητής dice1 σε έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 6.

Προφανώς θα διπλασιάσουμε (duplicate) την παραπάνω εντολή για να την εφαρμόσουμε και στη μεταβλητή με όνομα dice2.

Στη συνέχεια θα μεταφέρουμε τις παραπάνω εντολές ακριβώς κάτω από το κοινό γεγονός για το άγγιγμα του κουμπιού.



Εικόνα 59 - Ανάθεση τιμών στις μεταβλητές με το άγγιγμα του κουμπιού

Τέλος, στην εντολή που θέτει την κατάλληλη εικόνα του ζαριού **set Dice1Image.Picture to...** θα αντικαταστήσουμε στην εντολή **join** τον τυχαίο αριθμό, με την τιμή της κάθε μεταβλητής.

Για να χρησιμοποιήσουμε την τρέχουσα τιμή μιας μεταβλητής χρησιμοποιούμε την εντολή get [], που βρίσκεται στην ομάδα εντολών Variable, επιλέγοντας το όνομα της μεταβλητής από τη λίστα που ακολουθεί την εντολή.


Εικόνα 60 - Εντολή get

Το πρόγραμμα μας πρέπει να μοιάζει με την παρακάτω εικόνα.

whe	n ThrowButton - Click
do	set global dice1 to C random integer from (1) to (6)
	set global dice2 • to L random integer from L1 to L6
	set (Dice1Image 🔹 . (Picture 🔹 to 🚺 📋 join 🕻 get (global dice1 🔹
	(* .png *
	set (Dice2Image 🔹 . (Picture 🔹 to 🔓 🔲 join 🕻 get (global dice2 🔹
	(* .png *
	call ThrowSound . Play

Εικόνα 61 - Η εφαρμογή Ζάρια μετά την προσθήκη μεταβλητών



H ENTOAH IF (AN)

Για να δώσουμε τη δυνατότητα στην εφαρμογή μας να ανιχνεύει το αν ο χρήστης έφερε εξάρες, δηλαδή τον αριθμό 6 και στα δύο ζάρια, πρέπει να ελέγχουμε αν η τιμή των μεταβλητών dice1 και dice2 είναι ίση με 6.



Για το σκοπό αυτό θα χρησιμοποιήσουμε μια εντολή ελέγχου που ονομάζεται **if** [] **then** ... (Αν [] τότε) και βρίσκεται στην ομάδα εντολών **Control**.



Εικόνα 62 - Η εντολή If

Η εντολή Αν λειτουργεί ως εξής : ελέγχει μια λογική πρόταση, δηλαδή μια πρόταση που μπορεί να είναι είτε Αληθής είτε Ψευδής (να ισχύει ή να μην ισχύει). Σε περίπτωση που η πρόταση είναι Αληθής τότε εκτελούνται οι εντολές που «περιέχονται» κάτω από την εντολή Αν. Διαφορετικά, οι εντολές αυτές αγνοούνται και ΔΕΝ εκτελούνται.

Η ΛΟΓΙΚΗ ΠΡΟΤΑΣΗ

Η πρώτη μας ενέργεια είναι να σχεδιάσουμε τη λογική πρόταση που θέλουμε να εξετάσουμε. Καταρχάς ελέγχουμε αν η μεταβλητή dice1 έχει την τιμή 6, όπως φαίνεται παρακάτω.



Εικόνα 63 - Έλεγχος τιμής μεταβλητής

Το ίδιο ακριβώς θέλουμε να εξετάσουμε και για τη δεύτερη μεταβλητή της εφαρμογής dice2. Θέλουμε όμως να ενεργοποιηθούν οι εντολές κάτω από την εντολή Αν, εφόσον ισχύουν ΚΑΙ οι δύο προτάσεις που εξετάζουμε. Στο σημείο αυτό θα χρειαστεί να ενώσουμε τις δύο προτάσεις με το λογικό τελεστή and (και).



Εικόνα 65 - Ένωση προτάσεων με το AND

Τελικά, κουμπώνουμε τη σύνθετη λογική πρόταση που δημιουργήσαμε στην εντολή if.

O if C	get global dice1 • = • (6	and - I get global dice2 - = 1 6
then C		

Εικόνα 66 - Η εντολή if με ενσωματωμένη τη λογική πρόταση

οι ενεργείες κατώ από την εντολή if

Στην περίπτωση που ο χρήστης της εφαρμογής φέρει εξάρες θέλουμε να εμφανίζεται το μήνυμα «Κέρδισες!» στην ετικέτα που τοποθετήσαμε στο γραφικό περιβάλλον της εφαρμογής.

Για να θέσουμε το κείμενο της ετικέτας χρησιμοποιούμε την εντολή **set [ResultLabel].Text to...** την οποία θα βρούμε μόλις επιλέξουμε το αντικείμενο ResultLabel.



Εικόνα 67 - Η εντολή για να θέσουμε το κείμενο μιας ετικέτας

Θα κουμπώσουμε στην εντολή το κείμενο «Κέρδισες!». Η τελική μορφή της εντολής if θα μοιάζει όπως παρακάτω.



Εικόνα 68 - Η τελική μορφή της εντολής if

		Q
--	--	---

Αν η τιμή της μεταβλητής dice1 είναι 6 ΚΑΙ η τιμή της μεταβλητής dice2 είναι 6, τότε θέσε το κείμενο της ετικέτας σε «Κέρδισες!»

Ώρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

initia	initialize global dice1 to 10						
initi	alize global (dice2) to 📫 🕕						
whe	n ThrowButton Click						
do	set global dice1 • to (random integer from (1 to (6						
	set global dice2 to c random integer from (1 to (6						
	set Dice1Image . Picture to I D join I get global dice1						
	in the second						
	set Dice2Image • . Picture • to L D join L get global dice2 •						
	call ThrowSound . Play						
	if (get global dice1 = (6) and (get global dice2 = (6)						
	then set ResultLaber . Text to Képőiosci						



ήμα 7 Προσθήκη σκορ στηνεφαρμογή

Καλό θα είναι η εφαρμογή να δείχνει στον χρήστη της πόσες φορές έχει φέρει εξάρες, δηλαδή να μετράει σε πόσα παιχνίδια κέρδισε.

Για το σκοπό αυτό θα χρειαστούμε ακόμα μια μεταβλητή. Ας την ονομάσουμε score και ας της δώσουμε σαν αρχική τιμή το 0.



Στη συνέχεια θα πρέπει, όταν ο χρήστης φέρνει εξάρες, η τιμή της μεταβλητής να αυξάνεται κατά 1. Καταρχήν φτιάχνουμε την εντολή αύξησης της τιμής της μεταβλητής κατά 1.



Εικόνα 71 - Αύξηση τιμής score κατά 1

Θέτουμε το παραπάνω σαν νέα τιμή της μεταβλητής score.

set global score • to	set global score 🔹 to 🌘	get global score •	+ (1
-----------------------	-------------------------	--------------------	------

Εικόνα 72 - Ανάθεση νέας τιμής στην μεταβλητή score



Αύξησε την τιμή της μεταβλητής score κατά 1.

Τέλος τοποθετούμε την εντολή κάτω από την εντολή if.

Στη συνέχεια για να εμφανίζεται το σκορ στην ετικέτα ScoreLabel χρησιμοποιούμε την εντολή **set [ScoreLabel].Text to...** την οποία θα βρούμε μόλις επιλέξουμε το αντικείμενο ScoreLabel. Το κείμενο που θα εμφανίζεται θα είναι η λέξη «Σκορ :» και η τιμή της μεταβλητής score. Η τελική μορφή της εντολής θα είναι η παρακάτω.

set	ScoreLabel 🔹 . Text 🔹	to 🚺	🔲 join	ť.	Σкор : С	
				4	get (global so	core 🔹

Εικόνα 7	3.4	λλά7α	DALAC .	το κεί	nc s	ετικέτας	VIO TO	GKOO
	• •		o v cury	to net	12.5	concours	rea co	onop

Οι εντολές που εκτελεί η εφαρμογή μας μετά την προσθήκη του σκορ φαίνονται στην παρακάτω εικόνα.

initia	alize global dice1 to 40 (initialize global score) to 40						
initia	alize global dice2 to 10						
whe	n ThrowButton . Click						
do	do set global dice1 to (random integer from (1) to (6)						
	set global dice2 to (random integer from (1) to (6)						
	set Dice1Image . Picture to (D join) get global dice1						
	png *						
	set Dice2Image . Picture s to 🖡 🔲 join 🖡 get global dice2 🔹						
	(png)						
	call ThrowSound . Play						
	if if get global dice1 = ; 6 and v if get global dice2 = ; 6						
	then set ResultLabel . Text to [Κέρδισες]						
	set global score • to L O (get global score •) + (1)						
	set ScoreLabel . Text . to D join 1 Σκορ						

Εικόνα 74 - Η εφαρμογή ζάρια με σκορ



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

1. Παρατηρήστε ότι όταν ο χρήστης της εφαρμογής φέρει εξάρες και στη συνέχεια ξαναπαίξει, η ετικέτα συνεχίζει να δείχνει το κείμενο «Κέρδισες!». Κάντε τις απαραίτητες ενέργειες, ώστε όταν ο παίκτης ρίχνει τα ζάρια το κείμενο της ετικέτας να γίνεται το κενό («»).

2. Τροποποιήστε την εφαρμογή, έτσι ώστε να αυξάνει το σκορ του παίκτη όχι μόνο στην περίπτωση που φέρει εξάρες, αλλά κάθε φορά που φέρνει διπλές, δηλαδή τον ίδιο αριθμό και στα δύο ζάρια.

3. Προσθέστε ένα νέο κουμπί στην εφαρμογή με τίτλο «Μηδένισε σκορ» ή προσθέστε στο ήδη υπάρχον κουμπί «Ρίξε ζάρια» (π.χ. με παρατεταμένο πάτημα longClick) τη δυνατότητα ο χρήστης να μηδενίζει το σκορ.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : GUARDDOG – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

Θα επεκτείνουμε τη λειτουργικότητα της εφαρμογής Guard Dog, ώστε να λειτουργεί σαν συναγερμός, όταν κάποιος επιχειρήσει να σηκώσει τη συσκευή από τη θέση της.

Πιο συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν η συσκευή παύει να είναι ακινητοποιημένη σε μια σταθερή επιφάνεια. Επιπλέον, η φιγούρα του σκύλου δεν θα ηρεμεί παρά μόνο μετά το πέρας λ.χ. 2 δευτερολέπτων ακινησίας της συσκευής.

βασικές εννοιές

- Αισθητήρας επιτάχυνσης (accelerometer).
- Έλεγχος για ικανοποίηση λογικής συνθήκης (if).
- Ρολόι χρονόμετρο (clock).

Βήμα1 Άνοιγμα υπάρχοντος project

Ανοίγουμε το ήδη υπάρχον project με ονομασία GuardDog.

Βήμα 2 Αντίδρασητου σκύλου στο σήκωμα της συσκευής

Οι φορητές ηλεκτρονικές συσκευές διαθέτουν πλήθος αισθητήρων, όπως τοποθεσίας (GPS), προσανατολισμού (Orientation), επιτάχυνσης (Accelaration) και ραβδωτού κώδικα (Barcode), οι οποίοι μπορούν να χρησιμοποιηθούν άμεσα στο App Inventor και να προσδώσουν ιδιαίτερα χαρακτηριστικά και αυτοματισμούς στην εφαρμογή μας. Εν προκειμένω, η συσκευή μας θα λειτουργεί σαν συναγερμός, αξιοποιώντας το επιταχυνσιόμετρο (accelererometer) της, σαν ανιχνευτή «σηκώματός» της.

Designer Γραφικό Περιβάλλον Εφαρμογής

Για να προσθέσουμε δυνατότητες ανίχνευσης της κίνησης της συσκευής, θα χρειαστεί να μεταβούμε στο Designer και να προσθέσουμε στην οθόνη της εφαρμογής μας το μη ορατό συστατικό AccelerometerSensor.

*	από την	μεταφέρουμε το	του δίνουμε το	μεταβάλλουμε τις
	ομάδα	αντικείμενο	όνομα	ιδιότητες
	Sensors	AccelerometerSensor	AccSensor	

Ο συγκεκριμένος αισθητήρας ανιχνεύει το κούνημα της συσκευής, μετρώντας παράλληλα (κατά προσέγγιση) την επιτάχυνση της βαρύτητας (m/s²) στις τρεις διαστάσεις.



Συγκεκριμένα, όταν η συσκευή βρίσκεται τοποθετημένη οριζόντια, με την οθόνη προς τα πάνω, η τιμή της παραμέτρου zAccel είναι περίπου 9.8 (*επιτάχυνση της βαρύτητας*). Όταν το τηλέφωνο κινείται προς τα πάνω, η τιμή της zAccel αυξάνεται.

Για τις ανάγκες του συναγερμού μας, αρκεί ο έλεγχος της τιμής της παραμέτρου zAccel.

```
Συγγραφή Προγράμματος Blocks
```

Μεταβαίνουμε στο Blocks, όπου θέλουμε να χτίσουμε το κατάλληλο σενάριο, που συνοψίζεται στο: «Όταν ο αισθητήρας (στον κατακόρυφο άξονα z) ανιχνεύσει κίνηση, τότε ο σκύλος θυμώνει και γαβγίζει».

Προκειμένου, να επιτύχουμε αυτό το επιλεκτικό σενάριο, χρειαζόμαστε τα ακόλουθα:

από το συρτάρι	επιλέγουμε την εντολή	η οποία σημαίνει
AcceleratorSensor	when AccSensor AccelerationChanged do	όταν ανιχνευτεί οποιαδήποτε αλ- λαγή στις τιμές των επιταχύνσεων στους τρεις άξονες, τότε
Control	if then	Αν ισχύει η <i>συνθήκη,</i> τότε
Math	= και την τροποποιούμε σε >	Ελέγχεται η τιμή μιας συνθήκης (το αποτέλεσμα της σύγκρισης είναι true ή false)
Math	0 και αλλάζουμε την τιμή σε 11	Ορίζει την τιμή 11
DogCalmClock	set DogCalmClock TimerEnabled to	Ενεργοποίησε ή απενεργοποίησε το χρονόμετρο

Συνδυάζοντας τα παραπάνω, χτίζουμε το απαιτούμενο σενάριο:

whe	n (Acc	Sensor 🔪 .AccelerationChanged
X/	Accel	yAccel zAccel
do	if 💿	C L get ZAccel ▼ > ▼ L 11
	then	call [DogBarkingSound] .Play
		set [DogSprite •]. Picture •] to ["AngryDog.png "



Όποτε υπάρχει αλλαγή στο επιταχυνσιόμετρο (δηλαδή, η παραμικρή κίνηση), αν παρατηρηθεί τιμή της zAccel μεγαλύτερη από 11 (ο αριθμός αυτός πρέπει οπωσδήποτε να είναι μεγαλύτερος του 10 και μπορούμε να πειραματιστούμε με κοντινές τιμές), τότε κάλεσε την αναπαραγωγή του ήχου γαβγίσματος και άλλαξε τη φιγούρα του σκύλου σε θυμωμένη.

Ο σκύλος μας θύμωσε και παραμένει θυμωμένος, παρόλο που αφήνουμε τη συσκευή ακίνητη. Προς το παρόν, ο μόνος τρόπος να ηρεμήσουμε το σκύλο, είναι πρώτα να τον εκνευρίσουμε (π.χ. αγγίζοντάς τον), ώστε αφήνοντάς τον, να επανέλθει στη χαλαρή του κατάσταση.

Βήμα 3 Οσκύλος ηρεμεί μετά από ένα διάστημα ακινησίας

Στο βήμα αυτό, και μετά από σήκωμα της συσκευής, ο σκύλος ηρεμεί από μόνος του, μετά από την πάροδο ενός χρονικού διαστήματος ηρεμίας.

Designer Γρα

Γραφικό Περιβάλλον Εφαρμογής

Για το σενάριο αυτό θα χρειαστούμε το αντικείμενο του ρολογιού (Clock).



Το αντικείμενο Clock λειτουργεί σαν ρολόι, σαν χρονόμετρο και σαν βοηθητικό εργαλείο για χρονικούς υπολογισμούς ή απλούστερα για την έναρξη γεγονότων ανά συγκεκριμένα χρονικά διαστήματα.

Εδώ θα το χρησιμοποιήσουμε σαν χρονόμετρο. Όταν εκπνεύσει το χρονικό διάστημα που ορίζουμε στην ιδιότητα TimerInterval (σε msec) τότε ενεργοποιείται ένα γεγονός.

από την	μεταφέρουμε το	του δίνουμε το	μεταβάλλουμε τις
ομάδα	αντικείμενο	όνομα	ιδιότητες
User Interface	Clock	DogCalmClock	

Αρχικά, δεν χρειάζεται το χρονόμετρο να είναι ενεργό, καθώς το χρειαζόμαστε μόνο όταν σηκωθεί η συσκευή.

Η διάρκεια του χρονομέτρου είναι 3000 msec, δηλαδή ορίζουμε να τερματίζει (και να ξεκινάει ξανά) κάθε 3 δευτερόλεπτα.

```
Συγγραφή Προγράμματος
```

Blocks

Για να ενεργοποιείται το χρονόμετρο, με σήκωμα της συσκευής, αλλά και για να απενεργοποιείται αυτόματα μετά από το χρονικό διάστημα που ορίσαμε προηγουμένως, χρειαζόμαστε την εντολή:

\$ από το συρτάρι	επιλέγουμε την εντολή	η οποία σημαίνει
DogCalmClock	set DogCalmClock TimerEnabled to	Ενεργοποίησε ή απενεργο- ποίησε το χρονόμετρο
Logic	true ή false	Αληθές ή Ψευδές

Έτσι, επεκτείνουμε τη λειτουργία ελέγχου της τιμής του επιταχυνσιομέτρου:



Αντιστοίχως, όταν το χρονόμετρο τερματίζει, θέτουμε τη φιγούρα του σκύλου στην ήρεμη εκδοχή της και εφόσον δεν χρειάζεται να λειτουργεί πλέον το χρονόμετρο, μπορούμε να το απενεργοποιήσουμε ξανά.





Όταν το χρονόμετρο DogCalmClock τερματίσει, θέσε στη μορφή του σκύλου την ήρεμη φιγούρα και απενεργοποίησε το χρονόμετρο.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΚΟΡΩΝΑ ΓΡΑΜΜΑΤΑ – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

Θα επεκτείνουμε την εφαρμογή Κορώνα – Γράμματα που δόθηκε για εξάσκηση στο σπίτι στο προηγούμενο μάθημα. Καταρχήν θα την μετατρέψουμε σε ένα απλό παιχνίδι, στο οποίο ο παίκτης θα επιλέγει Κορώνα ή Γράμματα και στη συνέχεια θα ρίχνει το νόμισμα. Επίσης, θα προσθέσουμε animation κατά την ρίψη του νομίσματος, για να μοιάζει το νόμισμα σαν να γυρίζει.

βασικές εννοιές

- Ο λογικός τελεστής NOT.
- Εντολή ελέγχου Αν ... Αλλιώς (If ... then ... else).
- Animation με τη βοήθεια ρολογιού.

Βήμα**1** Άνοιγμα του έργου «Κορώνα Γράμματα»

Αφού μεταβείτε στη σελίδα του <u>App Inventor</u> και συνδεθείτε με τα στοιχεία του λογαριασμού σας, ανοίξτε το έργο κορώνα γράμματα.

Designer

Γραφικό Περιβάλλον Εφαρμογής

Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <u>http://tiny.cc/aifiles</u> και το φάκελο CoinFlip. Θα χρειαστείτε δυο εικόνες για τις δυο όψεις του νομίσματος και τους ήχους για το στρίψιμο και την επιτυχή πρόβλεψη.

Βήμα 2 Προσθήκη κουμπιών επιλογής στηνεφαρμογή

Θα προσθέσουμε δύο νέα κουμπιά στην εφαρμογή, ώστε να δίνουμε τη δυνατότητα στον χρήστη να επιλέξει κορώνα ή γράμματα πριν στρίψει το νόμισμα.

Τα δύο κουμπιά θα μπουν μέσα σε ένα αντικείμενο HorizontalArrangement ώστε να τοποθετηθούν το ένα δίπλα στο άλλο.

*	από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
	Layout	HorizontalArrangement	CoinArea	Height : 100 pixels
				AlignHorizontal : Center

UserInterface	Button	HeadButton	Image : 1.jpg
			Width : 60 pixels
			Height : 60 pixels
UserInterface	Button	TailButton	Image : 2.jpg
			Width : 60 pixels
			Height : 60 pixels



Εικόνα 75 - Προσθήκη κουμπιών επιλογής στην εφαρμογή

Βήμα 3 Προσθήκη κουμπιούγια το στρίψιμο του νομίσματος

Αν δεν το έχετε κάνει ήδη από τις ασκήσεις του προηγούμενου μαθήματος προσθέστε και ένα κουμπί για το στρίψιμο του νομίσματος καθώς και ένα αντικείμενο Player για την αναπαραγωγή των ήχων. Προσέξτε ότι αρχικά ορίζουμε την ιδιότητα enabled (Ενεργοποιημένο) του κουμπιού στην τιμή false, ώστε να μην μπορεί να χρησιμοποιηθεί μέχρι να επιλέξει ο χρήστης την πλευρά του νομίσματος που θέλει.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	HorizontalArrangement	ButtonArea	Width : Fill Parent AlignHorizontal : Center
UserInterface	Button	CoinButton	lmage : 1.jpg Visible : hidden Width : 150 pixels Height : 150 pixels
Media	Player	AppPlayer	

Βήμα4 Προσθήκη ετικέτας με μήνυμα επιτυχίας

Θα προσθέσουμε μια ετικέτα για να εμφανίζουμε μήνυμα επιτυχίας στον χρήστη όταν μαντεύει σωστά το νόμισμα.

\$ από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
UserInterface	Label	WinMessageLabel	Font Size : 40
			Text Alignment : center
			Text :
			Width : Fill Parent

Βήμα5

Επιλογή χρήστη και στρίψιμο νομίσματος

Συγγραφή Προγράμματος

Blocks

Αρχικά θα δημιουργήσουμε μια μεταβλητή που θα αποθηκεύει την επιλογή του χρήστη. Η μεταβλητή αυτή θα παίρνει την τιμή 1 όταν ο χρήστης επιλέγει κορώνα και την τιμή 2 όταν ο χρήστης επιλέγει γράμματα.

Θα ονομάσουμε τη νέα μας μεταβλητή choice και θα της δώσουμε αρχικά την τιμή 0.

Επίσης θα χρειαστούμε μια μεταβλητή coin που θα αποθηκεύει το αποτέλεσμα από το στρίψιμο του νομίσματος, δηλαδή τον τυχαίο αριθμό 1 ή 2. Η μεταβλητή coin θα έχει και αυτή αρχικά την τιμή 0.

Στη συνέχεια θα πρέπει να υλοποιήσουμε την παρακάτω συμπεριφορά κατά το άγγιγμα των κουμπιών.

Όταν ο χρήστης επιλέγει το κουμπί της κορώνας, θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή choice θα παίρνει την τιμή 1.
- Το κουμπί επιλογής για τα γράμματα θα κρύβεται.
- Το κουμπί επιλογής για την κορώνα θα απενεργοποιείται (ιδιότητα Enabled) ώστε ο χρήστης να μην μπορεί να το ξαναεπιλέξει.
- Το κουμπί για το στρίψιμο του νομίσματος θα ενεργοποιείται.
- Το κείμενο της ετικέτας για το αποτέλεσμα του στριψίματος θα γίνεται το κενό «».

Όταν ο χρήστης επιλέγει το κουμπί γράμματα θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή choice θα παίρνει την τιμή 2.
- Το κουμπί επιλογής για την κορώνα θα κρύβεται.
- Το κουμπί επιλογής για τα γράμματα θα απενεργοποιείται (ιδιότητα Enabled) ώστε ο χρήστης να μην μπορεί να το ξαναεπιλέξει.
- Το κουμπί για το στρίψιμο του νομίσματος θα ενεργοποιείται.

• Το κείμενο της ετικέτας για το αποτέλεσμα του στριψίματος θα γίνεται το κενό «».

Τέλος, θα προγραμματίσουμε τη συμπεριφορά του κουμπιού για το στρίψιμο του νομίσματος.

Όταν ο χρήστης αγγίζει το κουμπί για το στρίψιμο του νομίσματος θα πρέπει να γίνονται οι παρακάτω ενέργειες.

- Η μεταβλητή coin θα παίρνει ως νέα τιμή έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 2.
- Θα αναπαράγεται ο ήχος του στριψίματος του νομίσματος (θα πρέπει να θέσετε την πηγή του ήχου στο κατάλληλο ηχητικό αρχείο).
- Η εικόνα του κουμπιού του νομίσματος θα γίνεται κορώνα ή γράμματα ανάλογα με το αποτέλεσμα στη μεταβλητή coin.
- Αν η επιλογή του χρήστη και το αποτέλεσμα του στριψίματος είναι τα ίδια τότε θα εμφανίζεται το μήνυμα «Κέρδισες» στην ετικέτα και θα αναπαράγεται ο ήχος της επιτυχίας (θα πρέπει να θέσετε την πηγή του ήχου στο κατάλληλο ηχητικό αρχείο).
- Τα κουμπιά για την επιλογή Κορώνα ή Γράμματα θα είναι και τα δύο ορατά και ενεργοποιημένα.
- Το κουμπί για το στρίψιμο του νομίσματος θα απενεργοποιείται.



Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή.

Μπορείτε να δείτε την μορφή της εφαρμογής μετά τις παραπάνω προσθήκες ανατρέχοντας στο <u>Παράρτημα με τις Λύσεις των Ασκήσεων</u>.

Βήμα6 Πέρααπό την εντολή if - Η εντολή else

Παρατηρήστε ότι η εφαρμογή εμφανίζει μήνυμα επιτυχίας όταν ο χρήστης κερδίζει, αλλά δεν κάνει τίποτα σε περίπτωση που χάνει.

Θα την επεκτείνουμε ώστε όταν ο χρήστης δεν μαντεύει σωστά να του εμφανίζει το μήνυμα «Έχασες» αναπαράγοντας και έναν κατάλληλο ήχο αποτυχίας.

h entoah else (aaai $\Omega\Sigma)$



Πολύ συχνά χρειάζεται να περιγράψουμε τις ενέργειες που θα εκτελεστούν όταν μια λογική πρόταση είναι αληθής και ταυτόχρονα να περιγράψουμε κάποιες άλλες ενέργειες όταν η **ίδια** λογική πρόταση ΔΕΝ ισχύει.

Για να το πετύχουμε θα πρέπει να επεκτείνουμε την εντολή if προσθέτοντας της την εντολή **else**.

Για να προσθέσουμε την εντολή else κάτω από την εντολή if κάνουμε κλικ στο μπλε τετράγωνο που βρίσκεται πάνω στην εντολή if, όπως φαίνεται παρακάτω.



Εικόνα 76 - Επεκτείνοντας την εντολή if

Θα ανοίξει ένα παράθυρο από το οποίο θα κουμπώσουμε την εντολή else κάτω από την εντολή if, σέρνοντας την με το ποντίκι μας ακριβώς κάτω από την if, όπως φαίνεται στο σχήμα.



Εικόνα 77 - Κουμπώνοντας το else κάτω από το if

Στη συνέχεια θα περιγράψουμε τις ενέργειες που θα γίνουν όταν ο χρήστης ΔΕΝ μαντέψει σωστά το νόμισμα και θα τις κουμπώσουμε κάτω από την εντολή else.

- Το κείμενο της ετικέτας θα γίνεται «Έχασες...»
- Θα αναπαράγεται ο ήχος αποτυχίας.

Η τελική μορφή της εντολής if...else φαίνεται παρακάτω.



Εικόνα 78 - Η εντολή if...else



Αν το νόμισμα και η επιλογή είναι τα ίδια, τότε θέσε το μήνυμα σε «Κέρδισες» και τον ήχο στο αρχείο success. Διαφορετικά, θέσε το μήνυμα σε «Έχασες» και τον ήχο στο αρχείο failure. Παίξε τον ήχο.



Γιατί η εντολή call [AppPlayer].Start μετακινήθηκε μετά την εντολή if...else ;



Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή.

Βήμα 7 Animation – Το νόμισμα που γυρίζει

Θα προσθέσουμε ένα απλό animation που θα αλλάζει την εικόνα του κουμπιού για το στρίψιμο του νομίσματος, έτσι ώστε να δίνει την αίσθηση ότι το νόμισμα γυρίζει.

Για το σκοπό αυτό θα χρειαστεί να προσθέσουμε 2 ρολόγια στην εφαρμογή. Το πρώτο θα καθορίζει τη συνολική διάρκεια του animation, για παράδειγμα 1 δευτερόλεπτο ενώ το 2° ρολόι θα καθορίζει κάθε πότε θα γίνεται η εναλλαγή της εικόνας του κουμπιού, για παράδειγμα κάθε 50 χιλιοστά του δευτερολέπτου.

α Ο	πό την μάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
U	lserInterface	Clock	DelayClock	TimerEnabled : OXI TimerInterval : 1000
U	lserInterface	Clock	AnimationClock	TimerEnabled : OXI TimerInterval : 50

Αρχικά τα δύο ρολόγια είναι απενεργοποιημένα. Η ενεργοποίησή τους θα γίνεται κάθε φορά που ο παίκτης θα αγγίζει το κουμπί της ρίψης του νομίσματος. Όταν τελειώσει το animation μετά το πέρας 1 δευτερολέπτου τα ρολόγια θα απενεργοποιούνται ξανά.

Καταρχήν προσθέτουμε κάτω από το γεγονός του αγγίγματος στο κουμπί της ρίψης τις εντολές:



Εικόνα 79 - Εντολές ενεργοποίησης των ρολογιών

Οι εντολές που θα εκτελεί το ρολόι DelayClock θα είναι η απενεργοποίηση των ρολογιών και η ενεργοποίηση των υπολοίπων αντικειμένων του παιχνιδιού ώστε ο χρήστης να μπορεί να ξαναπαίξει. Οι εντολές αυτές **θα μετακινηθούν από το γεγονός του αγγίγματος του κου**μπιού, στο γεγονός της «πυροδότησης» του ρολογιού, όπως φαίνεται παρακάτω.



Εικόνα 80 - Η πυροδότηση του πρώτου ρολογιού

Το ρολόι AnimationClock θα μας βοηθήσει στην εναλλαγή των εικόνων του κουμπιού. Θα χρειαστεί να δημιουργήσουμε μια νέα μεταβλητή με όνομα face που θα παίρνει τις τιμές True/False. Θα τις δώσουμε αρχικά την τιμή false, όπως παρακάτω.



Εικόνα 81 - Αρχικοποίηση της μεταβλητής face

Στη συνέχεια κάθε φορά που θα πυροδοτείται το ρολόι AnimationClock θα γίνονται οι παρακάτω ενέργειες :

- Αν η μεταβλητή face έχει την τιμή false, στο κουμπί θα εμφανίζεται η εικόνα της κορώνας. Διαφορετικά θα εμφανίζεται η εικόνα των γραμμάτων.
- Η μεταβλητή face θα παίρνει την αντίθετη τιμή από αυτή που έχει. Δηλαδή αν είναι true θα γίνεται false, ενώ αν είναι false θα γίνεται true.

Οι απαραίτητες εντολές φαίνονται παρακάτω :



Εικόνα 82 - Η πυροδότηση του ρολογιού για το Animation

Ο ΛΟΓΙΚΟΣ ΤΕΛΕΣΤΗΣ ΝΟΤ (ΌΧΙ)



Προκειμένου να αντιστρέψουμε την τιμή μιας λογικής μεταβλητής χρησιμοποιούμε τον τελεστή not που βρίσκεται στην ομάδα Logic. Η λειτουργία του είναι πολύ απλή. Αν μια λογική μεταβλητή έχει την τιμή true τότε βάζοντας μπροστά της τον λογικό τελεστή not παίρνει την τιμή false, και αντιστρόφως.

	not (true -	false
D not D	not (false -	true 🔹
Εικόνα 83 - Ο λογικός τελεστής not	Εικόνα 84 - Η λειτουργία το	ου τελεστή not

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Τροποποιήστε την εφαρμογή Ζάρια, ώστε να εμφανίζει σε μια ετικέτα το συνολικό αριθμό παιχνιδιών που έχει παίξει ο παίκτης καθώς και το ποσοστό των παιχνιδιών που ο παίκτης έφερε διπλές.

Ποσοστό κερδισμένων παιχνιδιών = κερδισμένα παιχνίδια / συνολικά παιχνίδια * 100

2. Αναπτύξτε μια εφαρμογή που θα λειτουργεί σαν αντίστροφη μέτρηση. Αρχικά ο χρήστης θα ορίζει τα δευτερόλεπτα που θα μετρά αντίστροφα η εφαρμογή. Στη συνέχεια θα ενεργοποιεί την αντίστροφη μέτρηση με το άγγιγμα ενός κουμπιού.

Η εφαρμογή θα εμφανίζει έναν έναν τους αριθμούς των δευτερολέπτων μετρώντας αντίστροφα. Όταν τα δευτερόλεπτα μηδενιστούν, η εφαρμογή θα παίζει έναν ήχο δική σας επιλογής. Για παράδειγμα αν ο χρήστης δώσει σαν αριθμό δευτερολέπτων το 10, η εφαρμογή θα πρέπει ανά δευτερόλεπτο να εμφανίζει έναν έναν τους αριθμούς

10 ... 9 ... 8 ... 7 ... 6 ... 5 ... 4 ... 3 ... 2 ... 1 ... 0

Φροντίστε να είναι ορατά κάθε στιγμή, μόνο τα αντικείμενα που χρειάζονται, δηλαδή κατά την αντίστροφη μέτρηση δεν θα εμφανίζεται το κουτί για τα δευτερόλεπτα ούτε το κουμπί για την ενεργοποίηση της μέτρησης.



Θα χρειαστείτε το αντικείμενο Clock για να μετρά το πέρασμα 1 δευτερολέπτου.

Η εφαρμογή σας θα πρέπει να ελέγχει αν ο χρήστης έχει γράψει κάποιον αριθμό στο πεδίο κειμένου των δευτερολέπτων ή αν το έχει αφήσει κενό. Στη δεύτερη περίπτωση θα του εμφανίζεται ένα μήνυμα σφάλματος, ζητώντας να διορθώσει το λάθος του.



Θα χρειαστείτε το αντικείμενο Notifier που βρίσκεται στην ομάδα UserInterface. Στο τμήμα των εντολών θα χρησιμοποιήσετε την εντολή που φαίνεται παρακάτω:



Για τον έλεγχο θα σας φανεί χρήσιμο το παρακάτω πλακίδιο της ομάδας Text, το οποίο επιστρέφει την τιμή Αληθές (true) εάν το κείμενο που περιέχεται στα δεξιά του έχει μηδενικό μήκος (δηλαδή είναι κενό):



3. Η ιδιότητα Source του αντικειμένου Player μπορεί να πάρει τιμή με δύο διαφορετικούς τρόπους. Να τους περιγράψετε και να καταγράψετε σε ποιες περιπτώσεις προτιμάμε τον έναν ή τον άλλον. **4.** Έχουμε δύο μεταβλητές με ονόματα counter και temp. Η αρχική τους τιμή είναι 100 και 0 αντίστοιχα. Ποια θα είναι η τιμή των παρακάτω μεταβλητών μετά την εκτέλεση των εντολών που φαίνονται παρακάτω;

set global temp 🔹 to 🕻 🔲 🕻 get global counter 🔹 +	(1
set global counter 🔹 to 🕻 🛄 🕻 get global counter 🔹	+ (5

5. Έχουμε μια μεταβλητή με όνομα counter και αρχική τιμή 50, όπως φαίνεται στην παρακάτω εικόνα.

initialize global counter to (150)					
whe	n Butt	on1Click			
do	l if	Get global counter ▼ C random integer from (1) to (100)			
	then	set global counter T to (get global counter + (1			
	else	set global counter T to C get global counter T - C 1			
	L_				

Έστω ότι πατάμε το κουμπί Button1 δύο φορές. Την πρώτη φορά ο τυχαίος αριθμός είναι το 60 και τη δεύτερη φορά ο τυχαίος αριθμός είναι το 10. Ποια θα είναι η τιμή της μεταβλητής counter μετά το δεύτερο πάτημα του κουμπιού; Να εξηγήσετε αναλυτικά την απάντησή σας.

6. Αναπτύξτε μια εφαρμογή που θα κάνει έναν υπολογισμό της αρεσκείας σας. Ένα παράδειγμα είναι η μετατροπή μιας μονάδας μέτρησης σε κάποια άλλη, όπως μέτρα σε χιλιόμετρα, μέρες σε ώρες κλπ. Ο χρήστης θα δίνει μια τιμή σαν είσοδο και η εφαρμογή θα του εμφανίζει το αποτέλεσμα. Φροντίστε να εμπλουτίσετε την εφαρμογή σας με κάποια γραφικά και ήχους.

Δοκιμάστε να φτιάξετε δύο εκδοχές της εφαρμογής. Στην πρώτη ο χρήστης θα γράφει την τιμή μέσα σε ένα πεδίο κειμένου (textbox) και θα ζητά να γίνεται ο υπολογισμός με το άγγιγμα ενός κουμπιού. Θα πρέπει βέβαια να γίνεται έλεγχος ότι έχει γράψει μια τιμή μέσα στο πεδίο κειμένου, διαφορετικά να ενημερώνεται με μήνυμα σφάλματος.

Στη δεύτερη εκδοχή θα χρησιμοποιεί το αντικείμενο του μεταβολέα (slider). Με κάθε αλλαγή στην τιμή του μεταβολέα θα υπολογίζεται αυτομάτως και το νέο αποτέλεσμα.

Παρακάτω δίνεται σαν παράδειγμα μια εφαρμογή που μετατρέπει βαθμούς Κελσίου σε Φαρενάιτ.









Ενδεικτικό στιγμιότυπο μιας αντίστοιχης εφαρμογής, η οποία μετατρέπει ταχύτητες από χιλιόμετρα/ώρα σε μίλια/ώρα, φαίνεται παρακάτω. Η συγκεκριμένη εκδοχή διαθέτει μεταβολέα (slider) και επιπλέον κουμπιά για ακριβέστερη αυξομείωση της ταχύτητας. Για το συγχρονισμό των κουμπιών και του μεταβολέα χρησιμοποιήθηκε μεταβλητή.



7. Αναπτύξτε το παιχνίδι του κουλοχέρη. Σκοπός του παιχνιδιού είναι ο παίκτης να καταφέρει να κερδίσει πόντους - χρήματα με βάση τους κανόνες που περιγράφονται παρακάτω. Σε κάθε προσπάθεια ο παίκτης καταχωρεί το ποσό που θα ποντάρει και στη συνέχεια «τραβάει» το μοχλό του κουλοχέρη. Ανάλογα με τις εικόνες που θα εμφανιστούν στον κουλοχέρη, ο παίκτης κερδίζει κάποιο ποσό ή όχι.

Κανόνες παιχνιδιού

- Αν και οι 3 εικόνες είναι ជ тότε ο παίκτης κερδίζει ποσό ίσο με το 10πλάσιο του ποσού που ποντάρισε.
- Αν και οι 3 εικόνες είναι οι ίδιες (αλλά όχι το 7, όπως προηγουμένως), τότε ο παίκτης κερδίζει ποσό ίσο με το 5πλασιο του ποσού που ποντάρισε.
- Αν 2 από τις 3 εικόνες είναι ίδιες τότε ο παίκτης κερδίζει ποσό ίσο με το 2πλασιο του ποσού που ποντάρισε.
- Σε κάθε άλλη περίπτωση ο παίκτης δεν κερδίζει τίποτα.

Σε κάθε περίπτωση βέβαια ο παίκτης χάνει το ποσό που ποντάρισε.

Η εφαρμογή που θα φτιάξετε θα πρέπει να ελέγχει κάθε φορά που ο παίκτης «τραβάει» το μοχλό του κουλοχέρη ότι έχει δώσει ποσό πονταρίσματος.

Επίσης θα πρέπει να τον ενημερώνει κάθε φορά για το συνολικό ποσό που έχει κερδίσει ή που έχει χάσει. Φροντίστε να βάλετε animation στο γύρισμα των εικόνων και το τράβηγμα του μοχλού.

Μπορείτε να κατεβάσετε μια πρώτη έκδοση της εφαρμογής με έτοιμο το γραφικό της περιβάλλον από τη διεύθυνση : <u>http://tiny.cc/aifiles</u> και το φάκελο Slot Machine.



Μπορείτε να δείτε τις εντολές όλων των παραπάνω εφαρμογών στην ενότητα «<u>Λύσεις των</u> <u>Ασκήσεων</u>».

ενδεικτικές λύσεις των ασκήσεων

ΖΑΡΙΑ – ΕΠΕΚΤΑΣΗ ΓΙΑ ΔΙΠΛΕΣ, ΚΑΘΑΡΙΣΜΟ ΕΤΙΚΕΤΑΣ & ΜΗΔΕΝΙΣΜΟ ΣΚΟΡ



ΖΑΡΙΑ – ΕΠΕΚΤΑΣΗ ΜΕ ΕΜΦΑΝΙΣΗ ΣΥΝΟΛΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ & ΠΟΣΟΣΤΟΥ ΝΙΚΩΝ



ΚΟΡΩΝΑ ΓΡΑΜΜΑΤΑ – ΕΠΕΚΤΑΣΗ ΜΕ ΕΠΙΛΟΓΗ ΧΡΗΣΤΗ

initialize glob	al coin) to	C	1
initialize global	choice	to	C	0

whe	n (He	eadButton 🔹 .Click
do	set	global choice 🔹 to 🔓 🚺
	set	(WinMessageLabel 🔹 . (Text 🔹 to 🙌 🚺 🚺
	set	CoinButton . Enabled to true
	set	(HeadButton 🔹). (Enabled 🔹 to 🔰 false 🔹
	set	TailButton 🔹 . (Visible 🔹 to 🕴 false 🔹
	_	

wher	n (Ta	ailButton 🔹 .Click			
do	set	global choice to (2)			
	set	(WinMessageLabel -) . (Text -) to 📭			
	set	CoinButton . Enabled to true			
	set	(TailButton 🔹 . (Enabled 💌 to 🔰 false 🔪			
	set	(HeadButton 🔹). (Visible 🔹 to 🔰 false 🔪			



κορωνά γραμματά – επεκτάση με ανιματιόν

call (AppPlayer).Start

initialize global face to false initialize global coin to finitialize global coin to finitialize global choice to for	when Screen1 . Initialize do set AnimationClock . TimerEnabled to false . set DelayClock . TimerEnabled to false .
when CoinButton • .Click do set global coin • to f random integer from (1 to (2 set AnimationClock • . TimerEnabled • to (true • set DelayClock • . TimerEnabled • to (true • set AppPlayer • . Source • to (CoinFlip.mp3 * call AppPlayer • . Start	when AnimationClock • Timer do
<pre>when DelayClock • Timer do set AnimationClock • TimerEnabled • to false • set DelayClock • TimerEnabled • to false • • if fget global coin • = • 1 then set CoinButton • Image • to f 1,png * else set CoinButton • Image • to f 2,png * set CoinButton • Enabled • to false • set HeadButton • Enabled • to false •</pre>	when HeadButton • Click do set global choice • to (1 set WinMessageLabel • Fext • to (" " " set CoinButton • Enabled • to (true • set HeadButton • Enabled • to (false • set TailButton • Visible • to (false •
set TailButton • . Enabled • to [true • set HeadButton • . Visible • to] true • set TailButton • . Visible • to [true • • if [get global coin • = •] get global choice • then set WinMessageLabel • . Text • to [* Κέρδισες] set AppPlayer • . Source • to [* Success.wav * else set WinMessageLabel • . Text • to [* Έχασες set AppPlayer • . Source • to [* failure.mp3 *	<pre>when TailButton • .Click do set global choice • to (2) set WinMessageLabel • . Text • to (* * set CoinButton • . Enabled • to (true • set TailButton • . Enabled • to (false • set HeadButton • . Visible • to (false • </pre>

ΑΝΤΙΣΤΡΟΦΗ ΜΕΤΡΗΣΗ

initialize global countSeconds to 📫 🚺



ΜΕΤΑΤΡΟΠΗ ΒΑΘΜΩΝ ΚΕΛΣΙΟΥ ΣΕ ΦΑΡΕΝΑΙΤ ΜΕ ΠΕΔΙΟ ΚΕΙΜΕΝΟΥ



ΜΕΤΑΤΡΟΠΗ ΒΑΘΜΩΝ ΚΕΛΣΙΟΥ ΣΕ ΦΑΡΕΝΑΙΤ ΜΕ ΜΕΤΑΒΟΛΕΑ (SLIDER)



ΜΕΤΑΤΡΟΠΗ ΤΑΧΥΤΗΤΑΣ ΜΕ ΜΕΤΑΒΟΛΕΑ (SLIDER)



SLOT MACHINE (KOY Λ OXEPH Σ)



whe	n Ani	mationClock Timer			
do	🛛 if	((get global up) =) (false)			
	then	set (LeverUpSprite) . (Visible) to (true -)			
	set (LeverDownSprite) . (Visible) to (false)				
		set Animation2Clock . TimerEnabled to true :			
	else	set (Animation2Clock) . (TimerEnabled) to (false -			
		set AnimationClock . TimerEnabled to I false -			
		set global totalAmount to get global totalAmount - get global pontarisma			
		if (get global s1 = 3 and) (get global s2 = 3 and) (get global s3 = 3 (3 and)			
		then set global totalAmount to			
		set (MusicPlayer, Source, to (slotwins.mp3)			
		call Music layers .start			
		enden in get global s1 = i get global s2 and i get global s2 = i get global s3			
		then set global totalAmount to			
		set (MusicPlayer, Source) to (slotwins.mp3)			
		call MusicPlayer Start			
		else it i get global s12 = i get global s2 or i get global s2 = i get global s3 or i get global s1 = i get global s3			
		then set global totalAmount to get global totalAmount + I (2 × (get global pontarisma)			
		set (MusicPlayer) . Source to (Slotwins.mp3) call (MusicPlayer) .Start			
		set global pontarisma to (0			
		set AmountLabel). Text • to I II join (Συνολικό κέρδος) • get global totalAmount • € •			
		set PontarismaArea . Visible to I true -			
	set 👔	lobal up to (not) get global up			

ΣΤΟΧΟΙ

- Να αναλύετε το πρόβλημα σε απλούστερα τμήματα και να επιλύετε το κάθε τμήμα με ένα ξεχωριστό υποπρόγραμμα.
- Να δημιουργείτε υποπρογράμματα και να τα καλείτε, όπου χρειάζονται.
- Να δημιουργείτε υποπρογράμματα με παραμέτρους και να τα καλείτε κατάλληλα.
- Να σχεδιάζετε στην οθόνη με τη βοήθεια του καμβά.
- Να ανιχνεύετε τον προσανατολισμό της κινητής συσκευής με χρήση του κατάλληλου αισθητήρα.
- Να χρησιμοποιείτε τον αισθητήρα προσανατολισμού για κίνηση αντικειμένων εντός του γραφικού περιβάλλοντος.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΓΚΑΛΕΡΙ ΦΩΤΟΓΡΑΦΙΩΝ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή που θα δίνει τη δυνατότητα να παρουσιάζουμε ένα σύνολο από φωτογραφίες. Ο χρήστης της εφαρμογής θα έχει τη δυνατότητα να μετακινηθεί από τη μια φωτογραφία στην επόμενη με 3 διαφορετικούς τρόπους. Αγγίζοντας ένα κουμπί, σύροντας το δάκτυλό του πάνω στην οθόνη και αυτόματα κάθε 3 δευτερόλεπτα.



Εικόνα 85 - Εφαρμογή PhotoGallery

βασικές εννοιές

- Υποπρόγραμμα διαδικασία (procedure).
- Εντολή επιλογής if ... else if ... (Αν ... Αλλιώς Αν ...).

α1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα PhotoGallery.

Designer Γραφικό Περιβάλλον Εφαρμογής

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen1) και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της.

*	επιλέγουμε το αντικείμενο	μεταβάλλουμε τις ιδιότητες
	Screen1	AlignHorizontal: Center
		Screen Orientation: Portrait
		Scrollable: No
		Title: PhotoGallery

Βήμα **2**

Προσθήκη των απαραίτητων αρχείωνπολυμέσων

Ας προσθέσουμε στο project μας τα αρχεία εικόνας και τον ήχο που θα χρησιμοποιεί η εφαρμογή μας. Επισκεφθείτε τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο PhotoGallery.

ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή
1.jpg	Εικόνα για προβολή στο άλμπουμ
2.jpg	Εικόνα για προβολή στο άλμπουμ
3.jpg	Εικόνα για προβολή στο άλμπουμ
4.jpg	Εικόνα για προβολή στο άλμπουμ
5.jpg	Εικόνα για προβολή στο άλμπουμ
6.jpg	Εικόνα για προβολή στο άλμπουμ
next.png	Κουμπί επόμενο
previous.png	Κουμπί προηγούμενο
Averil White - Death of the Hero.mp3	Μουσικό κομμάτι εφαρμογής

μα 3 Προσθήκη αντικειμένωνστο γραφικό περιβάλλον

Θα χρειαστούμε το αντικείμενο του καμβά, πάνω στο οποίο θα εμφανίζονται οι φωτογραφίες του άλμπουμ. Επιπλέον θα προσθέσουμε τα κουμπιά για μετακίνηση σε προηγούμενη και επόμενη φωτογραφία, το ρολόι για την αυτόματη εναλλαγή και ένα αντικείμενο Player για την αναπαραγωγή της μουσικής. Τέλος σε μια ετικέτα θα εμφανίζουμε τον αριθμό της εικόνας που απεικονίζεται εκείνη τη στιγμή στην οθόνη.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	HorizontalArrangement	ButtonsArea	AlignHorizontal : Center
User Interface	Button	PreviousButton	Image : previous.png Text :
User Interface	Label	PicNumberLabel	FontSize : 30 Text : TextAlignment: center Width : 100 pixels
User Interface	Button	NextButton	Image : next.png Text :
Drawing and Animation	Canvas	GalleryCanvas	BackgroundImage : 1.jpg
User Interface	Clock	TimerClock	TimerInterval : 3000
Media	Player	MusicPlayer	Source : Averil White - Death of the Hero.mp3

Βήμα 4 Εναλλαγή των εικόνων του άλμπουμ

Για να επιτύχουμε την εναλλαγή των εικόνων θα χρειαστεί να δημιουργήσουμε μια μεταβλητή, η οποία θα παίρνει διαδοχικά τις τιμές από 1 μέχρι 6 και ξανά από την αρχή. Η τρέχουσα τιμή της μεταβλητής θα μας δείχνει και τη φωτογραφία που πρέπει να απεικονίσουμε πάνω στον καμβά (για το λόγο αυτό και τα αρχεία των εικόνων έχουν ονομαστεί κατάλληλα με αριθμούς από το 1 μέχρι το 6).

Ας ονομάσουμε τη μεταβλητή μας imageIndex με αρχική τιμή 1.

initialize global (imageIndex) to [1]

Εικόνα 86 - Μεταβλητή για την εναλλαγή των εικόνων

Η εφαρμογή μας θα αυξάνει κατά 1 την τιμή της μεταβλητής κάνοντας έλεγχο, όταν ξεπεράσει την τιμή 6, να της δίνει ξανά την τιμή 1. Με τον τρόπο αυτό οι εικόνες θα προβάλλονται κυκλικά. Στη συνέχεια η εφαρμογή θα ανανεώνει την εικόνα φόντου του αντικειμένου GalleryCanvas και το κείμενο της ετικέτας με τον αριθμό της εικόνας που προβάλλεται.

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα.

set global imageIndex - to 🕻 💷 🌔 get global imageIndex - + 🕻 🚺
if (get global imageIndex) >> (6)
then set global imageIndex to (1)
set GalleryCanvas . BackgroundImage to () join (get global imageIndex .
set [PicNumberLabel -]. Text - to get global imageIndex -

Εικόνα 87 - Εντολές για την εναλλαγή των εικόνων



Αύξησε την τιμή της μεταβλητής κατά 1. Αν αυτή ξεπεράσει το 6, δώσε της την τιμή 1.

Θέσε ως εικόνα φόντου του καμβά το αρχείο με όνομα ίδιο με την τιμή της μεταβλητής και κατάληξη «.jpg» και το κείμενο της ετικέτας ίσο με τον αριθμό της εικόνας που προβάλλεται.

Παρατηρήστε τώρα, ότι πρέπει να τοποθετήσουμε τις παραπάνω εντολές κάτω από τα 3 διαφορετικά γεγονότα, δηλαδή το άγγιγμα του κουμπιού, την πυροδότηση του ρολογιού και το σύρσιμο του δάκτυλου στην οθόνη.

Κάτι τέτοιο δε φαίνεται και πολύ λογικό, αφού θα περιγράφουμε ακριβώς τις ίδιες ενέργειες και στις 3 περιπτώσεις. Με τον τρόπο αυτό φορτώνουμε την εφαρμογή μας με τον ίδιο κώδικα ξανά και ξανά, κάνοντας την δυσανάγνωστη και λιγότερο κατανοητή σε κάποιον που τη διαβάζει. Σκεφτείτε την περίπτωση που οι εντολές δεν επαναλαμβάνονται 3, αλλά 10 ή 50 φορές!!

?

Πώς θα καταφέρουμε να μην αντιγράψουμε τις ίδιες εντολές ξανά και ξανά;

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ – Η ΔΙΑΔΙΚΑΣΙΑ

Ευτυχώς υπάρχει τρόπος να επιλύσουμε το παραπάνω πρόβλημα. Μπορούμε να τοποθετήσουμε τις εντολές για την εναλλαγή των εικόνων μέσα σε μια διαδικασία.



Η διαδικασία είναι ένα υποπρόγραμμα, δηλαδή ένα μικρό και αυτόνομο τμήμα της εφαρμογής στο οποίο τοποθετούμε μια σειρά εντολών. Στην ουσία χρησιμοποιώντας διαδικασίες δημιουργούμε νέες εντολές που μπορούμε στη συνέχεια να καλούμε (χρησιμοποιούμε) από οποιοδήποτε σημείο του προγράμματος.

Η χρήση υποπρογραμμάτων έχει μια σειρά πλεονεκτήματα, όπως:

 Διάσπαση του προβλήματος σε μικρότερα τμήματα και επίλυση των προβλημάτων αυτών με αυτόνομα τμήματα εντολών.

- Μείωση του χρόνου ανάπτυξης και τροποποιήσεων της εφαρμογής, αφού δεν χρειάζεται να επαναλαμβάνουμε όμοια τμήματα εντολών σε διάφορα σημεία της εφαρμογής.
- Ο κώδικας της εφαρμογής είναι πιο ευανάγνωστος και «συμμαζεμένος».

Για να δημιουργήσουμε μια νέα διαδικασία μεταβαίνουμε στην ομάδα Procedures (Διαδικασίες) ...

Blocks	Viewer
Built-in Control Logic Math	to procedure do
Text	to procedure changely result do
Colors Variables Procedures	call (changelmageNext -)

Εικόνα 88 - Ομάδα εντολών Procedures

.... και επιλέγουμε την εντολή to [procedure] do που φαίνεται παρακάτω.



Εικόνα 89 - Εντολή δημιουργίας διαδικασιών

Στη συνέχεια θα μετονομάσουμε τη διαδικασία μας από **procedure** σε **changelmageNext**, ώστε να έχει κάποιο συγκεκριμένο νόημα για εμάς το όνομά της και θα τοποθετήσουμε τις εντολές για την εναλλαγή των εικόνων μέσα σε αυτή. Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα.

	to changelmageNext
do	set global imageIndex • to (😐 (get global imageIndex •) + (1
	if (get global imageIndex >> (6) then set global imageIndex > (1)
	set GalleryCanvas • . Backgroundimage • to I i join f get global imageIndex •
	set PicNumberLabel . Text . to get global imageIndex .



κλήση υποπρογραμματών

Για να εκτελεστούν οι εντολές που βρίσκονται μέσα σε μια διαδικασία θα πρέπει να την καλέσουμε, όπως συνηθίζεται να λέμε, δηλαδή να την χρησιμοποιήσουμε μέσα σε ένα ή περισσότερα σημεία της εφαρμογής. Η εντολή για την κλήση μιας διαδικασίας που αναπτύσσουμε βρίσκεται και αυτή στην ομάδα Procedures.



Εικόνα 91 - Εντολή κλήσης της διαδικασίας

ήμα 5 Κλήσητης διαδικασίας κατά την εκτέλεση γεγονότων

Στο σημείο αυτό το πρόβλημα έχει απλοποιηθεί πολύ, αφού το μόνο που μένει να κάνουμε είναι να καλέσουμε τη διαδικασία μας, με την εκτέλεση των 3 γεγονότων, όπως παρακάτω.





Εικόνα 93 - Άγγιγμα κουμπιού "Επόμενο"







Αναπαραγωγή μουσικής

Κάντε τις απαραίτητες ενέργειες ώστε κατά την εκκίνηση της εφαρμογής να ξεκινάει και η αναπαραγωγή της μουσικής. Ρυθμίστε κατάλληλα το αντικείμενο Player, ώστε όταν η μουσική τελειώνει να ξεκινάει ξανά από την αρχή (ιδιότητα loop).



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή. Οι εντολές της εφαρμογής φαίνονται όλες μαζί στην παρακάτω εικόνα.

enti do	inze global (mageindex) to ()	when [NextButton II] Click do coll [changeImageNext]:::
	Image: set (sobal image: set (s	when GateryCanvas Filing x y speed heading xvel yvet filingSprite do cal changerimageNext
wf do	cal Musicipistratia Start	when STIMERClocks Timer do call Changelmapel/extra

Εικόνα 95 - Αρχική εκδοχή της εφαρμογής PhotoGallery

7 Μετακίνηση στην προηγούμενη εικόνα

Κάντε τις απαραίτητες ενέργειες, ώστε ο χρήστης της εφαρμογής να μπορεί να μετακινείται στην προηγούμενη εικόνα με δύο τρόπους. Είτε αγγίζοντας το κουμπί Προηγούμενο, είτε σέρνοντας το δάκτυλό του προς τα αριστερά. Υλοποιήστε μια διαδικασία με όνομα changelmagePrevious που θα εκτελεί τις απαραίτητες ενέργειες, όπως και πριν.

> Για να ανιχνεύσετε τη φορά προς την οποία μετακινείται το δάκτυλο του χρήστη πάνω στον καμβά θα χρειαστείτε την ιδιότητα heading (κατεύθυνση), την τιμή της οποίας σας παρέχει το γεγονός **when [GalleryCanvas] Flung.**



Για να ανιχνεύσετε την κίνηση του δάκτυλου προς τα δεξιά ελέγξτε αν η ιδιότητα heading έχει τιμές από -30 μέχρι 30.

Για να ανιχνεύσετε την κίνηση του δάκτυλου προς τα αριστερά ελέγξτε αν η ιδιότητα heading έχει τιμές από -200 μέχρι -160.

Η διαδικασία changeImagePrevious και η ανανεωμένη εκδοχή του γεγονότος GalleryCanvas.Flung φαίνονται στις παρακάτω εικόνες.



Εικόνα 96 - Διαδικασία για την μετακίνηση σε προηγούμενη εικόνα
whe	n GalleryCanvas T. Flung
x	y speed heading xvel yvel flungSprite
do	■ f f get heading > ≥ (-30) and > f get heading > ≤ (30)
	then call changelmageNext -
	else if leading < 2 , -200 and < leading < < , -160
	then call changelmagePrevious

Εικόνα 97 - Έλεγχος για κατεύθυνση συρσίματος του δάκτυλου πάνω στον καμβά

Βήμα8 Συγχωνεύοντας τιςδύο διαδικασίες σε μία

Αν παρατηρήσουμε προσεκτικά τις δύο διαδικασίες που υλοποιήσαμε για την μετακίνηση στην επόμενη και στην προηγούμενη εικόνα, θα δούμε ότι εκτελούν στην ουσία σχεδόν τις ίδιες εντολές.

Αυτό που τις διαφοροποιεί είναι η αύξηση ή η μείωση του αντίστοιχου αριθμού εικόνας κατά ένα. Επίσης, διαφέρουν στον έλεγχο για το αν ο αριθμός ξεπέρασε το 6 ή έγινε μικρότερος του 1, αλλά και αυτό εξαρτάται από το αν ο αριθμός αυξάνεται ή μειώνεται.



Μπορούμε να πετύχουμε τους σκοπούς μας με μια διαδικασία αντί για δυο;

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ



Έχουμε τη δυνατότητα να στέλνουμε τιμές σε μια διαδικασία τη στιγμή που την καλούμε. Οι τιμές αυτές ονομάζονται παράμετροι της διαδικασίας.

Στην ουσία οι παράμετροι είναι μεταβλητές που μεταφέρουν τιμές από την εφαρμογή προς τη διαδικασία.

Για να προσθέσουμε μια ή περισσότερες παραμέτρους σε μια διαδικασία κάνουμε κλικ στο μπλε τετράγωνο που βρίσκεται στα αριστερά του ονόματος της διαδικασίας. Ας επιλέξουμε τη διαδικασία με όνομα changelmageNext.



Εικόνα 98 - Προσθέτοντας παραμέτρους σε διαδικασίες

Από το παράθυρο που εμφανίζεται, σέρνουμε το πλακίδιο input μέσα στο πλακίδιο inputs. Βάζουμε τόσα πλακίδια, όσες και οι παράμετροι που θέλουμε να προσθέσουμε στη διαδικασία.

input: X	inputs
🔲 to Ch	nangelmageNext
do set	global imageIndex

Εικόνα 99 - Παράθυρο προσθήκης παραμέτρων

Στη θέση του x βάζουμε το όνομα που θέλουμε να έχει κάθε παράμετρος. Στην εφαρμογή μας θα την ονομάσουμε number.

input: 🛛	inputs input: numbe	1
🔲 to	changelmageNext	(number)

Εικόνα 100 - Μετονομασία παραμέτρων

Επίσης, θα μετονομάσουμε τη διαδικασία changelmageNext σε changelmage.

Η πρώτη αλλαγή που θα κάνουμε στις εντολές της διαδικασίας είναι να προσθέτουμε την τιμή της παραμέτρου number αντί να προσθέτουμε το 1.



Εικόνα 101 - Χρησιμοποιώντας την παράμετρο

Επιπλέον θα προσθέσουμε στην εντολή if και τους 2 ελέγχους.



Εικόνα 102 - Έλεγχος για άνω και κάτω όριο

to changelmage number
 set global imagelndex • to f e f get global imagelndex • + f get number •
 if f get global imagelndex • < f 1
 then set global imagelndex • to f 6
 else if f get global imagelndex • > (6
 then set global imagelndex • to f 1

Η τελική μορφή της διαδικασίας μας φαίνεται στην παρακάτω εικόνα.

Εικόνα 103 - Τελική μορφή της παραμετρικής διαδικασίας

PicNumberLabel .

set

GalleryCanvas . BackgroundImage . to

α 9 Αλλάζοντας την κλήσητης διαδικασίας

Text To get global imageIndex

Η διαδικασία που υλοποιήσαμε πρέπει κάθε φορά που καλείται να συνοδεύεται από έναν αριθμό, που είναι η τιμή που θέλουμε να δώσουμε στην παράμετρο number.

join (get global imageIndex * ".jpg "

Επομένως, στα σημεία που θέλουμε να μετακινούμαστε στην επόμενη εικόνα, θα δίνουμε σαν τιμή το +1, ενώ στα σημεία που θέλουμε να γίνεται μετακίνηση στην προηγούμενη εικόνα, θα δίνουμε σαν τιμή το -1.



Εικόνα 104 - Κλήση της παραμετρικής διαδικασίας

Η τελική μορφή της εφαρμογής φαίνεται παρακάτω.

initialize global (imageIndex) to	when Screen1 Initialize do call MusicPlayer Start	when TimerClock Timer do call changelmage I number [1]
o to changelmage number		
do set global imageIndex • to • • • • • • • • • • • • • • • • •	global imageIndex •) + (get number •)	when PreviousButton Click do call changeImage number [1] set TimerClock . TimerEnabled to [true set TimerClock . TimerEnabled to [false]
set GalleryCanvas . Backgroundimage set PicNumberLabel . Text to fig	to to join get global imageIndex •	when NextButton Click do set TimerClock . TimerEnabled to [false . call ChangeImage . number [] set TimerClock . TimerEnabled to [true .
get heading • ≥•	-30 and get heading < 30	
then call changelmage number / 1		
else if get heading • ≥ • (-200) and V (get heading) < V (-	160
then call changelmage - number [=1		
set TimerClock • . TimerEnabled • to	true •	

Εικόνα 105 - Τελική εκδοχή της εφαρμογής PictureGallery

επεκτάσεις

Παρατηρήστε ότι όταν μετακινούμαστε στην επόμενη εικόνα με το δάκτυλο ή με το κουμπί Επόμενο, ο μετρητής του χρονομέτρου δεν ξεκινάει να μετράει από τη νέα αρχή 3 δευτερόλεπτα, οπότε η μετάβαση στην επόμενη εικόνα μπορεί να γίνει σε μικρότερο χρονικό διάστημα. Κάντε τις απαραίτητες ενέργειες για να διορθωθεί το πρόβλημα.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΑΡΙΘΜΟΜΗΧΑΝΗ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή αριθμομηχανής, η οποία θα εκτελεί όλες τις βασικές αριθμητικές πράξεις.



Εικόνα 106 - Η εφαρμογή Αριθμομηχανή

βασικές εννοιές

- Υποπρόγραμμα διαδικασία (procedure) με ή χωρίς παραμέτρους.
- Εντολή επιλογής if ... else if ... (Αν ... Αλλιώς Αν ...).

Βήμα1 Εισαγωγή εφαρμογής από αρχείο

Θα εισάγουμε μια αρχική έκδοση της εφαρμογής που περιλαμβάνει έτοιμο το γραφικό της περιβάλλον. Σκοπός της δραστηριότητας είναι να κατασκευάσετε τις διαδικασίες που θα χρησιμοποιηθούν στην ολοκληρωμένη εφαρμογή.

Για να εισάγουμε το αρχείο που περιέχει την μισοτελειωμένη εκδοχή της εφαρμογής θα κατεβάσουμε αρχικά στον υπολογιστή μας to αρχείο Calculator.aia από τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο Calculator. Στη συνέχεια από την κεντρική σελίδα των projects του App Inventor θα κάνουμε κλικ στο μενού **Project** και θα επιλέξουμε το **Import project (.aia) from my computer**, όπως φαίνεται στην παρακάτω εικόνα.

My Projec	ots
Start new	project
Import pro	oject (.aia) from my computer
Delete pro	oject
Save proj	ect
Save proj	ect as
Checkpoi	nt
Export sel	lected project (.aia) to my compute
Export all	projects
Import ke	ystore
Export ke	ystore
0.1.1.1.1	

Εικόνα 107 - Εισαγωγή έτοιμου έργου στο περιβάλλον του App Inventor

Από το παράθυρο διαλόγου που θα ανοίξει θα επιλέξουμε το αρχείο που κατεβάσαμε στον υπολογιστή μας (Calculator.aia) και μετά από μερικά δευτερόλεπτα αναμονής θα το δούμε στη λίστα με τα Project μας.

Βήμα 2 Το γραφικό περιβάλλοντης εφαρμογής



Το γραφικό περιβάλλον της εφαρμογής αποτελείται από μια ετικέτα, στην οποία θα εμφανίζονται οι αριθμοί και το αποτέλεσμα της πράξης, καθώς και από 16 κουμπιά, που περιλαμβάνουν τα ψηφία από 0 - 9, τις 4 βασικές αριθμητικές πράξεις, το ίσον (=) για την εκτέλεση της πράξης και το κουμπί CL που θα «καθαρίζει» την αριθμομηχανή σε περίπτωση που θέλουμε να ακυρώσουμε έναν υπολογισμό. Blocks

ήμα3 Οέτοιμος κώδικας της εφαρμογής

Ας ρίξουμε μια ματιά στον κώδικα που υπάρχει ήδη έτοιμος στην εφαρμογή μας. Πρώτα απ' όλα υπάρχουν έτοιμα τα γεγονότα που ενεργοποιούνται από το άγγιγμα όλων των κουμπιών. Προς το παρόν δεν υπάρχουν εντολές κάτω από την ενεργοποίηση των γεγονότων, μιας και θα τις προσθέσουμε αργότερα.

when Button1 • .Click	when Button2 .Click	when Button3 .Click	when <u>Button_sin</u> .Click
do	do	do	do
when Button4 - Click	when Button5 Click	when Button6 .Click	when Button_plin • .Click
do		do	do
when Button7 • .Click	when Button8 • Click	when Button9 • Click	when Button_epiClick
do	do	do	do
when Button0 .Click	when Button_CL • Click	when Button_isonClick	when Button_dia • Click
do	do	do	do

Εικόνα 108 - Ο έτοιμος κώδικας της εφαρμογής Αριθμομηχανή (1)

Επιπλέον υπάρχουν τρεις εντολές αρχικοποίησης μεταβλητών.

Η μεταβλητή **number** θα αποθηκεύει τον αριθμό που «πληκτρολογεί» ο χρήστης εκείνη τη στιγμή στην αριθμομηχανή.

Η μεταβλητή **result** θα αποθηκεύει τον πρώτο αριθμό που πληκτρολόγησε ο χρήστης καθώς και το αποτέλεσμα της πράξης.

Τέλος η μεταβλητή action θα αποθηκεύει το σύμβολο της πράξης που ζήτησε ο χρήστης.

initialize global number to 📫 🔲 🎽
initialize global result to 📫 🚺
initialize global (action) to 🌾 🔵 🎽

Εικόνα 109 - Εικόνα 24 - Ο έτοιμος κώδικας της εφαρμογής Αριθμομηχανή (2)

Βήμα 4 «Διαβάζοντας » τον αριθμό που πληκτρολογείται

Θα υλοποιήσουμε μια διαδικασία (ονομάστε την **makeNumber**) που θα παίρνει σαν παράμετρο τον αριθμό του ψηφίου που «πατήθηκε» από τον χρήστη και θα το ενώνει με τον υπόλοιπο αριθμό που έχει ήδη πληκτρολογηθεί.

Για παράδειγμα, αν ο χρήστης έχει ήδη πληκτρολογήσει το ψηφίο 9 και στη συνέχεια πατήσει το ψηφίο 5, η διαδικασία μας θα πρέπει :

Α. να τα ενώνει (δηλαδή να φτιάχνει το 95),

🔲 join 🚺

B. να αποθηκεύει το αποτέλεσμα στην μεταβλητή number και

Γ. να εμφανίζει στην ετικέτα τον καινούριο αριθμό που σχηματίστηκε



Για την ένωση του νέου ψηφίου με τα προηγούμενα θα χρειαστείτε την εντολή

Τέλος, αφού ολοκληρώσουμε τη διαδικασία θα πρέπει να την καλέσουμε κατάλληλα όταν ο χρήστης αγγίζει κάποιο από τα αριθμητικά ψηφία.

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το <u>Παράρτημα – Λύσεις των Ασκήσεων</u> του μαθήματος.



Ώρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

ήμα 5 Καθαρίζοντας την αριθμομηχανή

Θα υλοποιήσουμε μια απλή διαδικασία (ονομάστε την **clearCalc**) που θα «καθαρίζει» τα δεδομένα που έχουν πληκτρολογηθεί στην αριθμομηχανή. Συγκεκριμένα, θα θέτει ξανά τις αρχικές τιμές των 3 μεταβλητών.

Αφού ολοκληρώσουμε την υλοποίηση της διαδικασίας, θα την καλέσουμε όταν πατηθεί το κουμπί καθαρισμού CL. Επίσης, θα θέτουμε στο κείμενο της ετικέτας της αριθμομηχανής το 0.



Ώρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

Βήμα6 Διαβάζοντας την αριθμητική πράξη

Η διαδικασία που θα υλοποιήσουμε σε αυτό το βήμα (ονομάστε την **actionClicked**) θα ενεργοποιείται όταν ο χρήστης αγγίζει ένα από τα κουμπιά με τις αριθμητικές πράξεις. Θα δέχεται σαν παράμετρο το σύμβολο της πράξης που επέλεξε ο χρήστης και στη συνέχεια θα εκτελεί τις παρακάτω ενέργειες :

A. Θα ελέγχει αν η μεταβλητή number έχει κάποια τιμή διαφορετική από το κενό αλφαριθμητικό «». Αυτός ο έλεγχος είναι απαραίτητος, ώστε να μην επιτρέπουμε στον χρήστη να επιλέγει αριθμητική πράξη πριν πληκτρολογήσει κάποιον αριθμό. Στην περίπτωση που η μεταβλητή number έχει τιμή:

B. Θα αποθηκεύει το σύμβολο της πράξης που επιλέχθηκε στην μεταβλητή action.

Γ. Θα αποθηκεύει την τιμή της μεταβλητής number στην μεταβλητή result.

Δ. Θα θέτει το κείμενο της ετικέτας ίσο με το σύμβολο της πράξης που επιλέχθηκε.

Ε. Θα θέτει την τιμή της μεταβλητής number ίση με το κενό αλφαριθμητικό «».

Τέλος, αφού ολοκληρώσουμε τη διαδικασία θα πρέπει να την καλέσουμε κατάλληλα όταν ο χρήστης αγγίζει κάποιο από τα κουμπιά των αριθμητικών πράξεων.

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το <u>Παράρτημα – Λύσεις των Ασκήσεων</u> του μαθήματος.

Βήμα 7 Υπολογίζοντας το αποτέλεσμα

Η τελευταία και πιο σημαντική διαδικασία που θα φτιάξουμε (ονομάστε την **calcResult**) θα ενεργοποιείται όταν ο χρήστης αγγίξει το κουμπί =. Οι ενέργειες που εκτελεί η διαδικασία είναι:

Α. Θα ελέγχει αν η μεταβλητή result έχει τιμή διαφορετική από το 0 και ταυτόχρονα η μεταβλητή number τιμή διαφορετική από το κενό αλφαριθμητικό. Αυτό είναι απαραίτητο προκειμένου να μην γίνονται πράξεις χωρίς να έχει πληκτρολογήσει ο χρήστης και τους δύο αριθμούς. Σε περίπτωση που ο παραπάνω έλεγχος είναι εντάξει, τότε η εφαρμογή :

B. Θα ελέγχει ποια από τις 4 πράξεις ζήτησε ο χρήστης. Ανάλογα θα ενημερώνει την τιμή της μεταβλητής result.

Γ. Θα θέτει το κείμενο της ετικέτας ίσο με την τιμή της μεταβλητής result.

Δ. Θα καλεί τη διαδικασία που καθαρίζει την αριθμομηχανή.

Τέλος, αφού ολοκληρώσουμε τη διαδικασία θα πρέπει να την καλέσουμε κατάλληλα όταν ο χρήστης αγγίζει το κουμπί = .

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το <u>Παράρτημα – Λύσεις των Ασκήσεων</u> του μαθήματος.



Η εφαρμογή μας είναι έτοιμη.

Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

Προσθέστε έναν προειδοποιητικό ήχο σφάλματος όταν ο χρήστης πατήσει κάποιο σύμβολο πράξης ή το = χωρίς να έχει δώσει αριθμό/αριθμούς.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΩΓΡΑΦΙΚΗ ΜΕ ΤΑ ΔΑΚΤΥΛΑ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή ζωγραφικής με τα δάχτυλα, η οποία θα επιτρέπει να ζωγραφίζουμε ακόμα και πάνω σε φωτογραφίες που τραβάμε με την κάμερα της συσκευής, ενώ θα μπορούμε και να αποθηκεύσουμε τις δημιουργίες μας. Το περιβάλλον της εφαρμογής μας θα μοιάζει με το παρακάτω.



Εικόνα 110 – Στιγμιότυπο της εφαρμογής Finger Painting

βασικές εννοιές

- Υποπρόγραμμα διαδικασία (procedure).
- Λήψη φωτογραφιών με την κάμερα της συσκευής.
- Σχεδίαση κύκλων και γραμμών σε καμβά.
- Αποθήκευση σχεδίου στη συσκευή.

ήμα1 Δημιουργίανέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα FingerPainting.

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen) και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της.

*	επιλέγουμε το αντικείμενο	μεταβάλλουμε τις ιδιότητες
	1	

Screen1	AlignHorizontal: Center
	BackgroundColor: LightGrey
	Screen Orientation: Portrait
	Scrollable: No
	Title: FingerPainting

ήμα 2 Προσθήκη των απαραίτητων αρχείωνπολυμέσων

Ας προσθέσουμε στο project μας τα αρχεία εικόνας που θα χρησιμοποιεί η εφαρμογή μας.

+	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή
	clear.jpg	Εικόνα για το κουμπί καθαρισμού του καμβά
	takepic.png	Εικόνα για το κουμπί λήψης φωτογραφίας
	save.png	Εικόνα για το κουμπί αποθήκευσης εικόνας

ΒήμαΞ

Σχέδια επάνω στον καμβά

Στο βήμα αυτό θα προσθέσουμε τα συστατικά και τις εντολές που θα μας επιτρέψουν να ζωγραφίζουμε τελείες και γραμμές με τα δάχτυλα. Αρχικά θα χρησιμοποιούμε μόνο ένα χρώμα και οι γραμμές θα έχουν προκαθορισμένο πάχος. Αργότερα ωστόσο θα δώσουμε στην εφαρμογή μεγαλύτερη ευελιξία.

Designer

Γραφικό Περιβάλλον Εφαρμογής

Προσθέτουμε τον καμβά (Canvas) επάνω στον οποίο θα ζωγραφίζουμε.

			F	ισιστητές
Drav Anin	ving and nation	Canvas	Paper	Paint Color: Blue Width: Fill Parent
				Height: Fill Parent

Συγγραφή Προγράμματος

Blocks

Οι δύο ομάδες εντολών που ακολουθούν ανιχνεύουν πότε ο χρήστης ακούμπησε την οθόνη της συσκευής ή έσυρε το δάχτυλο πάνω σε αυτήν. Στην πρώτη περίπτωση σχεδιάζεται μια τελεία, ενώ στη δεύτερη μια γραμμή.



Εικόνα 111 - Σχεδιασμός κύκλου στο σημείο επαφής



when Paper . Dragged							
st	artX	startY	prevX	P	revY	currentX	currentY
do	call	Paper v	.DrawLine				
			×1	. I	get	prevX 🔻	
	y1			. I	get	prevY 🔻	
			x2	: 🌔	get	currentX 🔻	
			y2		get	currentY 🔻	





Όταν ο χρήστης σύρει το δάχτυλό του στον καμβά Paper τότε ζωγράφισε μια γραμμή ανάμεσα στο τρέχον σημείο επαφής (με συντεταγμένες currentX και currentY) και στο προηγούμενο σημείο απ' όπου πέρασε το δάχτυλο του χρήστη (με συντεταγμένες prevX και prevY).

Το πάχος και το χρώμα της γραμμής δεν καθορίζονται από τις συγκεκριμένες εντολές, αλλά από τις ιδιότητες LineWidth και PaintColor του καμβά. Σε αυτό το βήμα θα προσθέσουμε τη δυνατότητα καθαρισμού του σχεδίου μας με ένα κούνημα της συσκευής.

Designer Γραφικό Περιβάλλον Εφαρμογής

Κάθε συσκευή διαθέτει αισθητήρα επιτάχυνσης ο οποίος μπορεί να ανιχνεύσει, μεταξύ άλλων, πότε ο χρήστης ταρακουνά την συσκευή. Προσθέτουμε ένα αντικείμενο που θα μας επιτρέψει να χρησιμοποιήσουμε αυτόν τον αισθητήρα (AccelerometerSensor) στην εφαρμογή.

από την	μεταφέρουμε το	του δίνουμε το	
ομάδα	αντικείμενο	όνομα	
Sensors	Accelerometer Sensor	AccSensor	

Συγγραφή Προγράμματος

Blocks

Συνδυάζουμε το γεγονός της ανίχνευσης ταρακουνήματος της συσκευής με την ενέργεια καθαρισμού της οθόνης.



Εικόνα 113 - Καθαρισμός καμβά με ταρακούνημα



Όταν ο αισθητήρας AccSensor ανιχνεύσει ταρακούνημα της συσκευής τότε καθάρισε τον καμβά Paper.

Βήμα**5**

Μιαμικρή παλέτα με χρώματα

Στο βήμα αυτό θα προσθέσουμε στην εφαρμογή τα κουμπιά που θα μας επιτρέψουν να επιλέγουμε και να χρησιμοποιούμε διαφορετικά χρώματα.

Designer Γραφικό Περιβάλλον Εφαρμογής

Μεταφέρουμε μια οριζόντια τακτοποίηση (HorizontalArrangement) πάνω ή κάτω από τον καμβά και τοποθετούμε μέσα σε αυτήν τρία κουμπιά (Button), έτσι ώστε να εμφανίζονται το ένα δίπλα στο άλλο.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	ColorArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Button	BlueButton	BackgroundColor: Blue Shape: Oval Text: Width: 50 pixels Height: 50 pixels
User Interface	Button	RedButton	BackgroundColor: Red Shape: Oval Text: Width: 50 pixels Height: 50 pixels
User Interface	Button	OrangeButton	BackgroundColor: Orange Shape: Oval Text: Width: 50 pixels Height: 50 pixels

Συγγραφή Προγράμματος

Blocks

Η παρακάτω ομάδα εντολών συσχετίζει το άγγιγμα ενός εκ των κουμπιών με το χρώμα ζωγραφικής.

Θα πρέπει να επαναληφθεί μία φορά για κάθε κουμπί χρώματος (δηλαδή και για το RedButton και το OrangeButton).



Εικόνα 114 - Ορισμός του χρώματος βαφής



Όταν ο χρήστης αγγίξει το κουμπί BlueButton τότε το χρώμα με το οποίο ζωγραφίζουμε στον καμβά Paper γίνεται μπλε (μέσω της ιδιότητας PaintColor).

Βήμα6 Αλλάζοντας το πάχος της γραμμής

Στο βήμα αυτό θα προσθέσουμε τη δυνατότητα μεταβολής του πάχους που θα έχουν οι γραμμές που σχεδιάζουμε.

Designer Γραφικό Περιβάλλον Εφαρμογής

Προσθέτουμε μια οριζόντια τακτοποίηση (HorizontalArrangement) κάτω από τα κουμπιά επιλογής χρώματος και τοποθετούμε μέσα της έναν μεταβολέα (Slider), δηλαδή έναν δείκτη που κινείται αριστερά και δεξιά. Ο κυλιόμενος αυτός μεταβολέας θα χρησιμοποιηθεί για την αλλαγή του πάχους των γραμμών.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	SliderArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Slider	WidthSlider	MaxValue: 30 MinValue: 2 ThumbPosition: 2 Width: 150 pixels

Οι ιδιότητες MinValue και MaxValue που χρησιμοποιούνται για τον μεταβολέα αντιστοιχούν στο ελάχιστο και στο μέγιστο πάχος γραμμής. Η ιδιότητα ThumbPosition είναι η τρέχουσα τιμή του μεταβολέα (άρα και το τρέχον πάχος).

Συγγραφή Προγράμματος

Blocks

Συσχετίζουμε το πάχος γραμμής του καμβά με τη θέση του μεταβολέα.



Εικόνα 115 - Ορισμός πάχους γραμμής



Όταν αλλάξει η θέση thumbPosition του μεταβολέα Width τότε ενημέρωσε ανάλογα και το πάχος των γραμμών του καμβά Paper (μέσω της ιδιότητας LineWidth). Όταν το πάχος της γραμμής γίνει μεγάλο τότε οι γραμμές δεν σχεδιάζονται σωστά. Μπορούμε να αντιμετωπίσουμε το πρόβλημα σχεδιάζοντας έναν κύκλο στο τρέχον σημείο επαφής με την οθόνη. Η ακτίνα του κύκλου θα πρέπει να είναι το μισό του πάχους γραμμής. Προσθέτουμε λοιπόν την ακόλουθη εντολή στο whenPaperDragged του Βήματος 3.



Εικόνα 116 - Βελτίωση για γραμμές μεγάλου πάχους

Βήμα΄

Σχέδια πάνωσε φωτογραφίες

Σε αυτό το βήμα θα μπορέσουμε να ζωγραφίσουμε επάνω σε φωτογραφίες που τραβάμε με την συσκευή μας.

Designer Γραφικό Περιβάλλον Εφαρμογής

Χρειαζόμαστε ένα κουμπί (Button) με το οποίο θα ενεργοποιούμε την κάμερα της συσκευής, καθώς και ένα αντικείμενο-συστατικό που αντιστοιχεί στην ίδια την κάμερα (Camera). Τοποθετούμε το κουμπί μέσα σε μια νέα οριζόντια τακτοποίηση.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	ButtonArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Button	CameraButton	Width: 50 Height: 40 Text: Image: takepic.png
Media	Camera	Camera1	

Συγγραφή Προγράμματος

Blocks

Καλούμε την ενσωματωμένη στο App Inventor λειτουργία που επιτρέπει τη λήψη φωτογραφίας, με το πάτημα του κουμπιού CameraButton.



Εικόνα 117 - Λήψη φωτογραφίας



Όταν ο χρήστης πατήσει το κουμπί CameraButton τότε ενεργοποίησε την φωτογραφική μηχανή Camera1 της συσκευής.



Εικόνα 118 - Ορισμός της φωτογραφίας ως φόντου του καμβά



Όταν τραβηχτεί μια φωτογραφία image από την φωτογραφική μηχανή Camera1 τότε θέσε την φωτογραφία αυτή ως φόντο στον καμβά Paper.

Βήμα8 Καθαρισμός του καμβά(συνέχεια)

Ο καθαρισμός του καμβά στο Βήμα 4 αφήνει ανέπαφη την εικόνα που πιθανώς να έχουμε θέσει ως φόντο. Μπορούμε να εισάγουμε ένα νέο κουμπί με όνομα ClearButton (εικόνα clear.png), να το τοποθετήσουμε δίπλα στο SaveButton και να προσθέσουμε τις ακόλουθες εντολές, έτσι ώστε το πάτημα του κουμπιού να καθαρίζει το φόντο (και το σχέδιό μας).



Εικόνα 119 - Καθαρισμός φόντου με πάτημα κουμπιού



Όταν πατηθεί το πλήκτρο ClearButton ακύρωσε το τυχόν υπάρχον φόντο του καμβά.

Βήμα 9 Αποθήκευσητων σχεδίων μας

Για να έχουμε τη δυνατότητα να αποθηκεύσουμε τα σχέδιά μας, προσθέτουμε καταρχάς ένα ακόμα κουμπί στην εφαρμογή μας, με όνομα SaveButton. Προκειμένου να ερωτάται ο χρήστης για το όνομα του αρχείου προς αποθήκευση, θα χρησιμοποιήσουμε και ένα αντικείμενο Notifier.



Το αντικείμενο Notifier επιτρέπει την εμφάνιση μηνυμάτων ή παραθύρων διαλόγου, από τα οποία μπορεί ο χρήστης να ενημερωθεί για κάτι, να επιλέξει ανάμεσα από δυο επιλογές ή ακόμα και να εισάγει την απάντηση που επιθυμεί, ώστε αυτή να χρησιμοποιηθεί στην εφαρμογή.

Designer Γραφικό Περιβάλλον Εφαρμογής

μεταφέρουμε το του δίνουμε το μεταβάλλουμε τις από την ομάδα αντικείμενο όνομα ιδιότητες User Interface Width: 50 Button SaveButton Height: 40 Text: Image: save.png **User Interface** SaveNotifier Notifier

Εισάγουμε λοιπόν τα δυο προαναφερθέντα αντικείμενα.



Σε πρώτη φάση θα εμφανίζουμε ένα παράθυρο διαλόγου που θα προτρέπει το χρήστη να δώσει όνομα στο αρχείο προς αποθήκευση. Στην συνέχεια, αφού δοθεί το όνομα, το σχέδιο αποθηκεύεται στη συσκευή και εμφανίζεται ένα ενημερωτικό μήνυμα για μερικά δευτερόλεπτα με το όνομα του αρχείου και τη θέση αποθήκευσης.

Blocks



Εικόνα 120 - Ερώτημα για όνομα αρχείου προς αποθήκευση



Όταν πατηθεί το πλήκτρο SaveButton εμφάνισε ένα παράθυρο διαλόγου με τίτλο «Filename:», κυρίως μήνυμα «Δώσε όνομα αρχείου...», το οποίο μπορεί να ακυρωθεί από το χρήστη, αν το επιθυμεί. Αμέσως μετά, θα αξιοποιήσουμε την απάντηση που δόθηκε για να αποθηκεύσουμε το σχέδιο.



Εικόνα 121 - Αποθήκευση της ζωγραφιάς με το όνομα που δόθηκε



Αφού ολοκληρώθηκε η εισαγωγή του ονόματος [response] από το χρήστη, κάλεσε την έτοιμη διαδικασία αποθήκευσης που διαθέτει ο καμβάς Paper, αποθηκεύοντας τη ζωγραφιά με όνομα [response].png και εμφάνισε σχετικό ενημερωτικό μήνυμα.

εναλλακτικός τρόπος αποθηκεύσης με αυτοματή ονομασία αρχείου

Εναλλακτικά, η ζωγραφιά μας αποθηκεύεται σε ένα αρχείο με όνομα που ξεκινάει με fp- και ακολουθείται από έναν μοναδικό αριθμό. Ο μοναδικός αυτός αριθμός προέρχεται από το ρολόι του συστήματος (της συσκευής). Για να χρησιμοποιήσουμε τη δυνατότητα αυτή, εισάγουμε επιπλέον στην εφαρμογή μας ένα συστατικό Clock (με όνομα Clock1).



Εικόνα 122 - Αποθήκευση του σχεδίου με αυτόματη καταχώρηση ονόματος

Q

Όταν πατηθεί το πλήκτρο SaveButton, θέσε σε μια τοπική μεταβλητή με όνομα name το όνομα αρχείου *fp-[ώρα_συστήματος].png* και αποθήκευσε τη ζωγραφιά του καμβά στη συσκευή με το συγκεκριμένο όνομα αρχείου.

Εμφάνισε ένα παράθυρο διαλόγου με τίτλο Saved Image, με κυρίως μήνυμα filename: [*όνομα_αποθηκευμένης_εικόνας]* και με ένα κουμπί ΟΚ.



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

when Paper .Touched x y touchedSprite do call Paper .DrawCircle x get x y get y r r Paper . LineWit	when CameraButton .Click do call Camera1 .TakePicture when Camera1 .AfterPicture image do set Paper . BackgroundImage to get image
when BlueButton Click do set Paper PaintColor to when RedButton Click do set Paper PaintColor to when OrangeButton Click do set Paper PaintColor to when AccSensor Shaking do call Paper Clear	vhen Paper • .Dragged startX startY prevX prevY call Paper • .DrawLine x1 get prevX • y1 get prevY • x2 get currentX • y2 get currentX • y2 get currentX • y2 get currentX • y2 get currentX • y get currentX • y get currentX • y get currentY • r (
when WidthSlider .PositionChanged thumbPosition do set Paper .LineWidth to get thum	bPosition
when SaveButton Click do call SaveNotifier Click message title cancelable	[•] Δώσε όνομα αρχείου " • Filename: " true "
when SaveNotifier .AfterTextInput response do call SaveNotifier .ShowAlert notice (call Pap	er Y .SaveAs fileName join get response Y ".png "

Ολοκληρωμένη η εφαρμογή Finger Painting περιλαμβάνει τις παρακάτω εντολές ...

Εικόνα 123 - Οι εντολές της εφαρμογής Finger Painting

και το πηγαίο αρχείο με το πλήρες σετ εντολών που μπορείτε να εισάγετε απευθείας στο App Inventor βρίσκεται στο φάκελο FingerPainting του συνδέσμου <u>http://tiny.cc/aifiles.</u>

επεκτάσεις

1. Προς το παρόν, δεν υπάρχει κάποια εμφανής ένδειξη για το ποιο είναι το ενεργό χρώμα σχεδίασης στον καμβά. Μια εύκολη λύση είναι, όποτε πατάμε κάποιο κουμπί χρώματος, να αλλάζει το χρώμα του μεταβολέα (slider) πάχους γραμμής, ώστε να υποδηλώνει το ενεργό χρώμα.

Αν μεταβείτε στις ιδιότητες του αντικειμένου WidthSlider θα διαπιστώσετε ότι προς το παρόν χρησιμοποιείται πορτοκαλί χρώμα αριστερά του δείκτη και γκρι χρώμα δεξιά του.



Εικόνα 124 - Οι ιδιότητες ColorLeft και ColorRight του μεταβολέα

Προσθέστε τις κατάλληλες εντολές ώστε σε κάθε αλλαγή χρώματος, να χρωματίζονται και οι δυο πλευρές του μεταβολέα με το επιλεγμένο χρώμα.

2. Αν θέλετε να προσθέσετε στην παλέτα χρωμάτων (βλ. Βήμα 5) περισσότερα διαθέσιμα χρώματα, μαζί με τα ανάλογα κουμπιά, θα διαπιστώσετε ότι αυξάνονται και οι αντίστοιχες εντολές, οι οποίες ουσιαστικά επαναλαμβάνονται. Το μόνο στοιχείο που αλλάζει είναι το χρώμα (ιδιότητα PaintColor) του καμβά Paper.

Υπενθυμίζεται, από το Βήμα 5, ότι προκειμένου να ορίσουμε σαν χρώμα σχεδίασης στον καμβά π.χ. το μπλε, χρησιμοποιήσαμε το σετ εντολών ...



Εικόνα 125 - Ορισμός του ενεργού χρώματος σε μπλε

... ενώ στο προηγούμενο ερώτημα (1.α) έχετε προσθέσει κάποιες επιπλέον εντολές που σχετίζονται με το μπλε χρώμα.

Σας ζητείται να δημιουργήσετε μια διαδικασία setColor [x] η οποία θα καλείται όποτε πατάμε σε κάποιο κουμπί χρώματος και θα αναλαμβάνει να κάνει όλες τις απαραίτητες αλλαγές ιδιοτήτων.

Τροποποιήστε λοιπόν τα γεγονότα που συμβαίνουν με το πάτημα του μπλε, κόκκινου και πορτοκαλί κουμπιού ώστε να χρησιμοποιούν τη διαδικασία setColor και προσθέστε μερικά επιπλέον χρώματα της αρεσκείας σας, μαζί με τα αντίστοιχα κουμπιά στην παλέτα χρωμάτων.



Εικόνα 126 - Ενδεικτικό στιγμιότυπο της νέας εφαρμογής FingerPainting

3 [Επανάληψη στις μεταβλητές]. Κάποιος συμμαθητής σας δεν χρησιμοποίησε μεταβολέα για τον ορισμό του πάχους γραμμής στον καμβά, αλλά μια μεταβλητή με όνομα brushSize και δυο κουμπιά που αυξομειώνουν το μέγεθος. Παρακάτω βλέπετε ένα μέρος του κώδικα που δημιούργησε:



Εικόνα 127 - Τμήμα εντολών για αύξηση του μεγέθους γραμμής με το πάτημα του ανάλογου κουμπιού

Όταν, όμως επιχειρεί να σχεδιάσει κάτι στην οθόνη, δεν εμφανίζεται τίποτα, παρά μόνο όταν πατήσει το κουμπί BiggerBrushButton για δεύτερη φορά. Που βρίσκεται το πρόβλημα και τι αλλαγές προτείνετε για τη διόρθωσή του;

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΠΥΞΙΔΑ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή - πυξίδα που θα μας δείχνει πού είναι ο Βορράς.



Εικόνα 128 - Το περιβάλλον της εφαρμογής Πυξίδα

βασικές εννοιές

- Αισθητήρας ανίχνευσης προσανατολισμού της συσκευής (Orientation Sensor).
- Εντολή επιλογής if (Av).

ήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα Compass.

Designer

Γραφικό Περιβάλλον Εφαρμογής

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen) και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της.

*		επιλέγουμε το αντικείμενο	μεταβάλλουμε τις ιδιότητες
		Screen1	AlignHorizontal: Center
			Screen Orientation: Portrait
			Scrollable: No
			Title: Ψηφιακή Πυξίδα
Βήμα 2	Προσθήκη	αρχείων πολυ μέ	σων

Ας προσθέσουμε στο project μας το αρχείο εικόνας που θα χρησιμοποιεί η εφαρμογή μας. Θα το βρείτε στη διεύθυνση <u>http://tiny.cc/aifiles</u> στο φάκελο με όνομα Compass.

+	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή	
	compass-arrow.png	Εικόνα πυξίδας με βέλος	

α 3 Προσθήκη αντικειμένωνστο γραφικό περιβάλλον

Θα χρειαστούμε το αντικείμενο του καμβά, πάνω στο οποίο θα τοποθετηθεί το γραφικό της πυξίδας (ImageSprite), καθώς και τον αισθητήρα προσανατολισμού της συσκευής.

0	από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
	Drawing and Animation	Canvas	CompassCanvas	Width: Fill Parent Height: Fill Parent
	Drawing and Animation	ImageSprite	CompassSprite	X : 0 Y : 50 Width : 300 pixels Height : 300 pixels
	Sensors	OrientationSensor	CompassSensor	

Βήμα4

Ανιχνεύοντας την αλλαγή προσανατολισμού

Συγγραφή Προγράμματος

Blocks

Η εφαρμογή μας θα ανιχνεύει το γεγονός της αλλαγής του προσανατολισμού της κινητής συσκευής και θα προσαρμόζει κατάλληλα την εικόνα της πυξίδας.



Το αντικείμενο OrientationSensor μας δίνει 3 σημαντικές πληροφορίες σχετικές με την κλίση της κινητής μας συσκευής.

Roll : Έχει τιμή 0 όταν η συσκευή είναι επίπεδη, αυξάνεται στις 90° όταν η συσκευή είναι γυρισμένη στη δεξιά της πλευρά κ.ο.κ.

Pitch : Έχει τιμή Ο όταν η συσκευή είναι επίπεδη, αυξάνεται στις 90° όταν η πάνω πλευρά της συσκευής κοιτάει προς τα κάτω κ.ο.κ.

Azimuth : Έχει τιμή 0 όταν η συσκευή κοιτάζει στο Βορρά, 90° όταν κοιτάζει προς την Ανατολή, 180° στο Νότο και 270° στη Δύση.

Εμείς θα χρειαστούμε την τιμή της ιδιότητας Azimuth (Αζιμούθιο) μιας και αυτή μας δείχνει τον προσανατολισμό της συσκευής στον άξονα Βορρά – Νότου.

Από το αντικείμενο CompassSensor χρησιμοποιούμε την εντολή



Εικόνα 129 - Το γεγονός αλλαγής προσανατολισμού

Στη συνέχεια θα πρέπει κάθε φορά που αλλάζει ο προσανατολισμός της συσκευής να προσαρμόζουμε την κατεύθυνση (heading) της εικόνας της πυξίδας.

Από το αντικείμενο CompassSprite επιλέγουμε την εντολή ...



Εικόνα 130 - Εντολή αλλαγής κατεύθυνσης μιας εικόνας

... και την τοποθετούμε μέσα στο γεγονός της αλλαγής προσανατολισμού.

Για να πάρουμε την τιμή του αζιμούθιου της συσκευής τοποθετούμε το δείκτη του ποντικιού πάνω στο πεδίο **azimuth** και τραβάμε την αντίστοιχη εντολή, όπως φαίνεται στην παρακάτω εικόνα.

when Cor	npassSensor 🔹 .OrientationChanged
do azimuth	get azimuth •
	set azimuth to

Εικόνα 131 - Παίρνοντας την τιμή του αζιμούθιου

Τέλος, κουμπώνουμε την παραπάνω εντολή με το αζιμούθιο της συσκευής.



Εικόνα 132 - Θέτοντας τον προσανατολισμό της πυξίδας στο Βορρά



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

 Τροποποιήστε την εφαρμογή της πυξίδας, ώστε όταν η συσκευή δείχνει προς το Βορρά (με απόκλιση +/- 10 μοίρες) να αναπαράγει ένα ηχητικό σήμα της επιλογής σας.

2. Προσθέστε 3 ετικέτες που θα εμφανίζουν στην οθόνη τις τιμές των ιδιοτήτων Roll, Pitch και Azimuth.

A. Περιγράψτε ποια είναι η θέση της κινητής συσκευής όταν η ιδιότητα Roll παίρνει την τιμή 180.

B. Περιγράψτε ποια είναι η θέση της κινητής συσκευής όταν η ιδιότητα Pitch παίρνει την τιμή 270.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΠΙΑΣΕ ΤΗ ΣΗΜΑΙΑ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα υλοποιήσουμε ένα παιχνίδι στο οποίο ο παίκτης θα χειρίζεται μια μπάλα με τη βοήθεια του αισθητήρα προσανατολισμού (Orientation Sensor). Σκοπός τους παιχνιδιού είναι να αγγίξει η μπάλα τη σημαία χωρίς να πέσει μέσα στις τρύπες. Η θέση της σημαίας θα αλλάζει κάθε 4 δευτερόλεπτα.



Εικόνα 133 - Εφαρμογή "Πιάσε τη σημαία"

βασικές εννοιές

- Υποπρόγραμμα διαδικασία (procedure).
- Εντολή επιλογής if ... else if ... (Αν ... Αλλιώς Αν ...).
- Χρήση του Orientation Sensor ως χειριστηρίου.
- «Σύγκρουση» αντικειμένων.

Βήμα1 Δημιουργίανέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα CatchTheFlag.

Designer Γραφικό Περιβάλλον Εφαρμογής

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen1) και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της.

*	ε: Ο	πιλέγουμε το ιντικείμενο	μεταβάλλουμε τις ιδιότητες
	S	creen1	Screen Orientation: Portrait
			Scrollable: No
			Title: Πιάσε τη σημαία

Βήμα 2 Προσθήκη των απαραίτητων αρχείωνπολυμέσων

Ας προσθέσουμε στο project μας τα αρχεία εικόνας και τους ήχους που θα χρησιμοποιεί η εφαρμογή μας. Επισκεφθείτε τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο CatchTheFlag.

ł	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή	
	tiles.jpg	Εικόνα φόντου	
	Soccer_Ball_small.png	Γραφικό μπάλας Γραφικό τρύπας	
	hole_small.png		
	red_flag.png	Γραφικό σημαίας	
	Smashing.mp3	Ήχος σύγκρουσης με τρύπα	
	Tada.mp3	Ήχος σύγκρουσης με σημαία	

Βήμα**3**

Προσθήκη αντικειμένωνστο γραφικό περιβάλλον

Θα χρειαστούμε το αντικείμενο του καμβά, πάνω στο οποίο θα εμφανίζονται τα γραφικά. Επιπλέον, θα προσθέσουμε 5 γραφικά (ImageSprites) για την μπάλα, τη σημαία και τις 3 τρύπες, ένα αντικείμενο Player για την αναπαραγωγή των ηχητικών εφέ και ένα ρολόι για τη μετακίνηση της σημαίας. Τέλος θα χρειαστούμε και τον αισθητήρα προσανατολισμού (Orientation Sensor).

\$ από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Drawing and Animation	Canvas	GameCanvas	BackgroundImage : tiles.jpg Width : Fill Parent Height : Fill Parent
Drawing and Animation	ImageSprite	BallSprite	Picture : Soccer_Ball_small.png Width : 40 pixels Height : 40 pixels

Drawing and Animation	ImageSprite	FlagSprite	Picture : red_flag.png Width : 50 pixels Height : 50 pixels
Drawing and Animation	ImageSprite	HoleSprite1	Picture : hole_small.png
Drawing and Animation	ImageSprite	HoleSprite2	Picture : hole_small.png
Drawing and Animation	ImageSprite	HoleSprite3	Picture : hole_small.png
UserInterface	Clock	FlagClock	TimerInterval : 4000
Media	Player	SoundFXPlayer	
Sensors	OrientationSensor	GameSensor	

Συγγραφή Προγράμματος

Blocks

Bńuα4

Αρχική θέσημπάλας και σημαίας

Στο ξεκίνημα του παιχνιδιού θέλουμε τα γραφικά της μπάλας και της σημαίας να τοποθετηθούν σε τυχαίες θέσεις, μέσα στα όρια του καμβά, φυσικά.

wher	Screen1 .Initialize	
	call BallSprite .MoveTo	
		random integer from BallSprite Width to GameCanvas Width - BallSprite Width
	у	random integer from (BallSprite - Height - to GameCanvas - Height - to BallSprite - Height -
	call FlagSprite . MoveTo	
		random integer from (FlagSprite . Width . to (GameCanvas . Width FlagSprite . Width .
	y f	random integer from L FlagSprite . Height . to GameCanvas . Height FlagSprite . Height .



μα 5 Αλλαγή θέσης σημαίας κάθε 4 δευτερόλεπτα

Κάθε φορά που το χρονόμετρο μετράει 4 δευτερόλεπτα η σημαία θα μετακινείται σε μια νέα τυχαία θέση, όπως φαίνεται παρακάτω.

whe	n FlagClock - Timer	
do	call FlagSprite . MoveTo	
		random integer from Care FlagSprite . Width to CareCarvas . Width - CareCarvas . Width - Care Carvas . Width -
	у (random integer from Carl FlagSprite . Height . to Carl GameCanvas . Height . (FlagSprite . Height .

Εικόνα 135 - Μετακίνηση σημαίας κάθε 4 δευτερόλεπτα

Βήμα6 Μετακίνηση μπάλας με τη βοήθεια του αισθητήρα

Όταν η κινητή μας συσκευή αλλάζει προσανατολισμό θα πρέπει η μπάλα να μετακινείται προς την κατεύθυνση που «έστριψε» και η συσκευή. Επίσης, η ταχύτητα της μπάλας θα εξαρτάται από το πόσο μεγάλη κλίση δώσαμε στη συσκευή μας.

Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε καταρχήν το γεγονός αλλαγής κατεύθυνσης του αισθητήρα προσανατολισμού...



Εικόνα 136 - Γεγονός αλλαγής προσανατολισμού

... και θα αλλάξουμε την ταχύτητα και την κατεύθυνση του γραφικού της μπάλας ανάλογα με τα μεγέθη Angle (γωνία) και Magnitude (Μέγεθος) του αισθητήρα, όπως φαίνεται στην παρακάτω εικόνα.





Η ιδιότητα **Angle** του αισθητήρα προσανατολισμού μας δίνει την διεύθυνση προς στην οποία είναι στραμμένη η συσκευή, σε σχέση με την επίπεδη — οριζόντια θέση.

Η ιδιότητα **Magnitude** του αισθητήρα προσανατολισμού μας δίνει έναν αριθμό μεταξύ Ο και 1 που δείχνει πόσο μεγάλη είναι η κλίση της συσκευής.

Επειδή η τιμή του Magnitude είναι πολύ μικρή την πολλαπλασιάζουμε συνήθως με έναν συντελεστή της αρεσκείας μας (εδώ τον αριθμό 300) προκειμένου να χρησιμοποιηθεί σαν τιμή για την ταχύτητα κίνησης της μπάλας.

Βήμα 7 Μηδενισμός ταχύτητας της μπάλας στα άκρα του καμβά

Όταν η μπάλα αγγίζει κάποια από τις 4 πλευρές του καμβά θέλουμε να σταματάει να κινείται. Επομένως θα πρέπει να μηδενίζουμε την ταχύτητά της όταν αγγίζει τα άκρα, όπως φαίνεται παρακάτω.



Εικόνα 138 - Μηδενισμός ταχύτητας της μπάλας

Βήμα8 Σύγκρουσημπάλας με άλλα αντικείμενα

Πρώτα απ' όλα θα υλοποιήσουμε δύο διαδικασίες, που θα περιγράφουν τις ενέργειες που θα εκτελούνται όταν η μπάλα αγγίζει τη σημαία και τις ενέργειες που θα εκτελούνται όταν η μπάλα αγγίζει κάποια από τις τρύπες.

Στην πρώτη εκδοχή της εφαρμογής μας, οι διαδικασίες απλά θα θέτουν το κατάλληλο αρχείο ήχου για αναπαραγωγή στο αντικείμενο SoundFXPlayer. Στις επεκτάσεις που θα κάνετε θα εμπλουτιστούν με τις κατάλληλες εντολές.

Ας ονομάσουμε την πρώτη μας διαδικασία hitFlag και τη δεύτερη hitHole. Οι εντολές που θα εκτελούν φαίνονται στην παρακάτω εικόνα.



Εικόνα 139 - Διαδικασίες για τη σημαία και την τρύπα

Στη συνέχεια θα πρέπει όταν η μπάλα συγκρούεται με κάτι να ελέγχουμε με ποιο αντικείμενο έγινε η σύγκρουση και να εκτελούμε τις κατάλληλες ενέργειες. Καταρχήν, θα χρειαστούμε το γεγονός σύγκρουσης της μπάλας.

when	BallSprite . CollidedWith
oth	ier
do	

Εικόνα 140 - Γεγονός σύγκρουσης μπάλας

Η ιδιότητα other συμβολίζει το αντικείμενο με το οποίο έγινε η σύγκρουση. Στο σημείο αυτό θα πρέπει να συγκρίνουμε την τιμή του other με κάθε ένα από τα αντικείμενα της εφαρμογής, όπως φαίνεται παρακάτω.

whe	n (BallSprite	CollidedWith
ot	iner	
do	🛈 if 🏮	get other 🔹 🖘 (FlagSprite 🔹
	then call	(hitFlag 🔹
	else if 🌘	get other 🗴 💷 (HoleSprite1 🗴
	then call	(hitHole •
	else if 🌘	get other 🔪 = 🗙 (HoleSprite2 🔹
	then call	(hitHole 🔹
	else if 🌘	get other 🔹 = 🔹 (HoleSprite3 🔹
	then call	hitHole •

Εικόνα 141 - Έλεγχος σύγκρουσης μπάλας με αντικείμενο



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

- Η σημαία θα πρέπει να μετακινείται σε τυχαία θέση κάθε φορά που την «πιάνει» ο παίκτης.
- 2. Δημιουργήστε σκορ στην εφαρμογή και εμφανίστε το στον παίκτη. Κάθε φορά που ο παίκτης πιάνει τη σημαία το σκορ θα αυξάνεται κατά 5.
- Κάντε τις τρύπες να μετακινούνται σε τυχαίες θέσεις κάθε 4 δευτερόλεπτα όπως και η σημαία.
- 4. Δώστε στον παίκτη 3 ζωές. Κάθε φορά που η μπάλα πέφτει σε μια από τις τρύπες οι ζωές θα μειώνονται. Όταν μηδενιστούν το παιχνίδι θα τερματίζει. Μπορείτε να χρησιμοποιήσετε τις κατάλληλες εικόνες και ήχο που θα βρείτε στο φάκελο CatchTheFlag, ώστε όταν ο παίκτης χάνει :
 - a. Να εμφανίζεται η εικόνα bang και να ακούγεται ο ήχος της έκρηξης.
 - b. Μετά από 3 δευτερόλεπτα να εμφανίζεται η εικόνα PlayAgain. Όταν ο παίκτης κάνει κλικ πάνω της το παιχνίδι θα ξεκινά από την αρχή. Όταν το παιχνίδι τερματίζει θα πρέπει όλα να σταματούν μέχρι να πατηθεί το PlayAgain.
- 5. Προσθέστε μουσική που θα αναπαράγεται κατά τη διάρκεια του παιχνιδιού.

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Σχεδιάστε και αναπτύξτε το παιχνίδι MoleMash, εμπνευσμένο από ένα κλασσικό arcade παιχνίδι, στο οποίο εμφανίζονται αντικείμενα σε τυχαίες θέσεις και ο παίκτης πρέπει να προλάβει να τα χτυπήσει πριν αυτά εξαφανιστούν. Στη δική μας εκδοχή, για να μην γινόμαστε βίαιοι [©], ας υποθέσουμε ότι εμφανίζεται ένα χαριτωμένο ζωάκι και θέλουμε απλά να το αγγίξουμε πριν εξαφανιστεί και εμφανιστεί σε κάποια άλλη θέση.



Ενδεικτικές εκδοχές του παιχνιδιού φαίνονται στις παρακάτω εικόνες:

Τα απαραίτητα γραφικά για την εφαρμογή θα τα βρείτε στη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο MoleMash.

Βασικές λειτουργίες:

 Το ζωάκι (mole) εμφανίζεται κάθε ... δευτερόλεπτα σε κάποιο τυχαίο σημείο της πίστας, δηλαδή σε κάποιες τυχαίες συντεταγμένες (X,Y), οι οποίες πρέπει να λαμβάνουν υπόψη τις διαστάσεις της πίστας και του mole.

Μπορείτε να δημιουργήσετε μια διαδικασία moveMole η οποία θα μετακινεί το mole σε τυχαία θέση εντός του καμβά, χρησιμοποιώντας την εντολή ...

call Mole MoveTo	
× (random integer from [0] to [Pista]. Width] - [Mole]. Width]
у (random integer from (0 to (Pista . Height - Mole . Height)

... ώστε να βεβαιωθείτε ότι θα κινείται εντός της πίστας.

Η διαδικασία moveMole θα καλείται με τη βοήθεια χρονομέτρου (Clock).

 Εάν ο παίκτης προλάβει να αγγίξει το ζωάκι, τότε ο παίκτης κερδίζει ... πόντους και το mole εμφανίζεται αμέσως σε επόμενη τυχαία θέση.

Όταν συμβεί το γεγονός αγγίγματος της πίστας (καμβά), έλεγξε αν αγγίχθηκε το mole, ανανέωσε το σκορ (προτείνεται η δημιουργία μιας διαδικασίας με όνομα updateScore) και κάλεσε τη διαδικασία moveMole.

 Όταν πατηθεί το πλήκτρο "Start/Reset" μηδενίζεται το σκορ για την εκκίνηση ενός νέου παιχνιδιού.

Με το πάτημα του κουμπιού, δώσε τις κατάλληλες τιμές στις σχετικές μεταβλητές.

Ιδέες για επέκταση:

 Αντί η εικόνα του ζώου να είναι το mole, να επιλέγει ο παίκτης μια εικόνα από τις επαφές του τηλεφώνου του.

Θα χρειαστείτε το αντικείμενο ContactPicker από την κατηγορία Social, το οποίο προσθέτει το σχετικό κουμπί και να αξιοποιήσετε την ιδιότητα Picture.

- Εάν ο παίκτης αγγίξει την πίστα, αλλά όχι το mole, τότε χάνει ... πόντους.
 Το σκορ ενδεικτικά θα μπορούσε να προκύπτει π.χ. κερδίζοντας 2 πόντους για κάθε εύστοχο άγγιγμα και χάνοντας έναν πόντο για κάθε άστοχο άγγιγμα.
- Ένα χρονόμετρο μετράει αντίστροφα, ώστε το παιχνίδι να τελειώνει σε ... δευτερόλεπτα.

Ένα διαφορετικό χρονόμετρο από εκείνο του mole, μετράει το χρόνο και σταματάει το παιχνίδι όταν μηδενιστεί ο χρόνος.

 Να εμφανίζεται, σε τυχαία χρονικά διαστήματα, πέρα από το «καλό» mole και ένα «κακό» ζωάκι ή αντικείμενο, το οποίο αν αγγίξει ο παίκτης να αφαιρούνται ... πόντοι ή να χάνει ζωές.

Ξεκινώντας, μπορεί ο παίκτης να έχει συγκεκριμένο αριθμό ζωών και για κάθε «κακό» ζωάκι που αγγίζει να του αφαιρείται μια ζωή και να χάνει πόντους.

 Πριν την εκκίνηση, να επιλέγει ο παίκτης επίπεδο (π.χ. εύκολο - μέτριο - δύσκολο), με ανάλογη αλλαγή στην ταχύτητα εμφάνισης των moles. Εναλλακτικά, η ταχύτητα των moles μπορεί να αυξάνεται σταδιακά καθώς περνάει ο χρόνος ή όταν ο παίκτης φτάνει σε συγκεκριμένα σκορ.

Θα χρειαστείτε ένα αντικείμενο ListPicker από την κατηγορία User Interface, το οποίο προσθέτει ένα κουμπί για την επιλογή κάποιου στοιχείου από λίστα. Ορίστε όσα επίπεδα θέλετε και για κάθε επίπεδο, φροντίστε να αλλάζει η κατάλληλη ιδιότητα. 2. Αναπτύξτε μια εφαρμογή που θα υλοποιεί το παιχνίδι Pong. Το παιχνίδι αποτελείται από μια μπάλα που κινείται συνεχώς και αναπηδά στα τοιχώματα. Ο παίκτης χειρίζεται μια σανίδα, την οποία μετακινεί αριστερά δεξιά με το δάκτυλο. Ο παίκτης μετακινώντας τη σανίδα προσπαθεί να χτυπάει την μπάλα και να μην της επιτρέπει να αγγίζει την κάτω πλευρά της οθόνης (στην περίπτωση αυτή το παιχνίδι τερματίζει και ο παίκτης χάνει).



Προσπαθήστε να «σπάσετε» την εφαρμογή σας σε μικρά τμήματα, τα οποία και θα υλοποιήσετε με διαδικασίες.

Για τη μετακίνηση της σανίδας θα σας χρειαστεί το γεγονός **when [...] Dragged** που θα βρείτε στην ομάδα εντολών του γραφικού της σανίδας.

Επίσης για να ανιχνεύσετε το γεγονός της σύγκρουσης της σανίδας (BatImageSprite) με την μπάλα (BallSprite) θα χρειαστείτε τις παρακάτω εντολές.





Όταν το γραφικό της μπάλας συγκρουστεί με ένα άλλο αντικείμενο, αν το αντικείμενο αυτό είναι η σανίδα, τότε άλλαξε την φορά της μπάλας στην αντίθετη κατεύθυνση.

Επίσης για να ανιχνεύσετε τη σύγκρουση της μπάλας με τα «τοιχώματα» της οθόνης θα χρειαστείτε το γεγονός:
when	BallSprite . EdgeReached
edg	9
do	

Η τιμή της ιδιότητας edge μας δείχνει με ποια πλευρά της οθόνης «συγκρούστηκε» η μπάλα. Η κάτω πλευρά της οθόνης αντιστοιχεί στην τιμή -1 του edge.

Όταν ο παίκτης χάνει η μπάλα θα σταματά, και θα εμφανίζεται το μήνυμα "Game Over". Σε περίπτωση που ο παίκτης κάνει κλικ πάνω στο μήνυμα το παιχνίδι θα ξαναρχίζει από την αρχή.

Τα απαραίτητα γραφικά για την εφαρμογή θα τα βρείτε στη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο Pong.

3. Υποθέστε ότι συνεργάζεστε με άλλους προγραμματιστές για τη δημιουργία ενός παιχνιδιού με ζάρια. Στο παιχνίδι αυτό, ο παίκτης έχει το δικαίωμα να ρίξει τα ζάρια μέχρι π.χ. 4 φορές (4 rounds) και ανάλογα με τη ζαριά του, κερδίζει κάποιους πόντους και συνεχίζει ή χάνει και σταματάει.



Καθένας σας έχει αναλάβει να υλοποιήσει από ένα μέρος του παιχνιδιού. Οι συνεργάτες σας υλοποίησαν με τη χρήση διαδικασιών ένα μεγάλο τμήμα του παιχνιδιού (βλ. **DiceGame0.aia** στη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο DiceGame), ενώ εσείς έχετε αναλάβει το κομμάτι που ελέγχει τη ζαριά και αναλόγως του αποτελέσματος, δίνει x πόντους (points) στον παίκτη (βλ. διαδικασία getPoints (x)) ή τερματίζει το παιχνίδι (βλ. διαδικασία endgame), αναπαράγοντας μέσω του player το αρχείο failure.mp3. Σε κάθε γύρο, φυσικά ενημερώνεται και η ένδειξη του σκορ (scoreLabel) με βάση την τιμή της μεταβλητής score.

Καλείστε λοιπόν να μελετήσετε τον κώδικα που σας δίνεται και βασιζόμενοι σε αυτόν να υλοποιήσετε τη διαδικασία **checkDicePoints** η οποία κάνει τους απαραίτητους ελέγχους, συνεχίζει ή σταματάει το παιχνίδι και ενημερώνει το σκορ.

Ενδεικτικά, οι κανόνες του παιχνιδιού θα μπορούσαν να είναι οι εξής:

- Αν ο παίκτης φέρει δυο άσσους (1 και 1), τότε κερδίζει 40 πόντους.
- Αν ο παίκτης φέρει διπλές (εκτός από 1 και 1), τότε κερδίζει 25 πόντους.
- Αν ο παίκτης φέρει έναν άσσο και οτιδήποτε άλλο (εκτός από άσσο), τότε χάνει!
- Αν ο παίκτης φέρει οποιονδήποτε άλλο συνδυασμό, κερδίζει πόντους ίσους με το άθροισμα (sum) της ζαριάς του.

Φυσικά, αν το επιθυμείτε, μπορείτε να ορίσετε τους δικούς σας κανόνες...

4. Αναπτύξτε μια εφαρμογή η οποία θα εμφανίζει προσθέσεις μεταξύ τυχαίων αριθμών από το 1 μέχρι το 10 (ή σε όποιο εύρος θέλετε εσείς) και θα ζητά από τον χρήστη να υπολογίσει το αποτέλεσμα. Στη συνέχεια θα τον ενημερώνει αν το αποτέλεσμα που έδωσε είναι σωστό ή λανθασμένο (μπορείτε απλά να αναπαράγετε ένα ηχητικό εφέ για την επιτυχία ή την αποτυχία).

Σε περίπτωση που το αποτέλεσμα είναι σωστό, η εφαρμογή θα εμφανίζει την επόμενη πράξη, διαφορετικά θα παραμένει στην ίδια πράξη και θα περιμένει εκ νέου το αποτέλεσμα.

Μπορείτε να υλοποιήσετε την εφαρμογή με δύο τρόπους:

Α. Ο απλούστερος τρόπος υλοποίησης είναι όταν ο χρήστης γράφει το αποτέλεσμα σε ένα πεδίο κειμένου. Ένα παράδειγμα φαίνεται στην παρακάτω εικόνα.

Math for Kids	╤₄∥ 9:48
9	
+ 7	
	Απάντηση

Βέβαια, μιας και η εφαρμογή απευθύνεται σε παιδιά μικρής ηλικίας, καλό θα είναι να μην γίνεται η εισαγωγή του αποτελέσματος σε πεδίο κειμένου, ώστε τα παιδιά να μην χρειάζεται να γράφουν τους αριθμούς.

B. Καλύτερη μέθοδος είναι η επιλογή του αποτελέσματος μεταξύ 3 πιθανών απαντήσεων που θα εμφανίζει η εφαρμογή σε στυλ κουμπιού, όπως φαίνεται παρακάτω.

Οι δύο από τις 3 απαντήσεις στα κουμπιά θα παράγονται με τυχαίο τρόπο στο διάστημα [χ-5, χ+5], όπου χ η σωστή απάντηση.

Math for Kids		⊜⊿ 9:48
	9	
+	7	
16	17	12

Τα γραφικά που φαίνονται στις εικόνες είναι ενδεικτικά. Μπορείτε να σχεδιάσετε το γραφικό περιβάλλον της εφαρμογής, όπως σας αρέσει!

Σημείωση : Τέτοιου είδους εφαρμογές κυκλοφορούν ευρύτατα στο Google Play Store και απευθύνονται κυρίως σε παιδιά μικρών ηλικιών που θέλουν να εξασκηθούν στις αριθμητικές πράξεις με έναν ευχάριστο τρόπο. Για την ακρίβεια απευθύνονται στους γονείς των παιδιών!



Η εφαρμογή μας (παραλλαγή με κουμπιά) υπάρχει περίπτωση να μην δουλέψει σωστά (bug) . Μπορείτε να σκεφθείτε σε ποια περίπτωση;

Η απάντηση και ο τρόπος για να το διορθώσουμε, στο επόμενο επεισόδιο... 😅

5. Υλοποιήστε μια δική σας εφαρμογή – παιχνίδι που θα χρησιμοποιεί σαν χειριστήριο κίνησης τον Αισθητήρα Προσανατολισμού (Orientation Sensor).

ενδεικτικές γύσεις των ασκήσεων

PHOTOGALLERY (ΕΠΕΚΤΑΣΕΙΣ)



ΑΡΙΘΜΟΜΗΧΑΝΗ - ΔΙΑΔΙΚΑΣΙΑ ΠΟΥ ΔΙΑΒΑΖΕΙ ΤΟΝ ΑΡΙΘΜΟ





ΑΡΙΘΜΟΜΗΧΑΝΗ – ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΘΑΡΙΣΜΟΥ & ΕΠΙΛΟΓΗΣ ΠΡΑΞΗΣ



Εικόνα 142 - Η διαδικασία καθαρισμού



Εικόνα 143 - Κλήση της διαδικασίας καθαρισμού



Εικόνα 144 - Η διαδικασία επιλογής αριθμητικής πράξης









when Button_ison • .Click do call calcResult •

FINGERPAINTING (EΠΕΚΤΑΣΕΙΣ)

to SetColor X do Set Paper . PaintColo set WidthSlider . Colo set WidthSlider . Colo	or T to get X T orLeft to get X T orRight to get X T	
when RedButton Click do call SetColor x	when BlueButton Click do call SetColor x	when OrangeButton V.Click do call SetColor V x
when BlackButton Click do call SetColor x	when WhiteButton Click do call SetColor	

ΠΥΞΙΔΑ (ΕΠΕΚΤΑΣΗ – ΗΧΗΤΙΚΗ ΕΙΔΟΠΟΙΗΣΗ ΓΙΑ ΤΟ ΒΟΡΡΑ)



πιάσε τη σημαία - επεκτάσεισ



whe	n FlagClock Timer	
do	call FlagSprite . MoveTo	
	x	random integer from FlagSprite . Width to GameCanvas . Width - FlagSprite . Width -
	У	random integer from FlagSprite . Height . to (GameCanvas . Height FlagSprite . Height .
	call HoleSprite1 .MoveTo	
	×	random integer from (HoleSprite1 • . Width • to GameCanvas • . Width • - (HoleSprite1 • . Width •
	У	f random integer from (HoleSprite1 • . Height • to I (GameCanvas • . Height • . I HoleSprite1 • . Height •
	call HoleSprite2 MoveTo	
	×	random integer from (HoleSprite2 • . Width • to (GameCanvas • . Width • - (HoleSprite2 • . Width •
	У	random integer from (HoleSprite2 • . Height • to (GameCanvas • . Height • - HoleSprite2 • . Height •
	call (HoleSprite3	
	×	random integer from (HoleSprite3 • . Width • to GameCanvas • . Width • - GHoleSprite3 • . Width •
	y	random integer from (HoleSprite3 • . Height • to f (GameCanvas • . Height • - I HoleSprite3 • . Height •





	to (hitHole)	
do	set global lives • to left get global lives • - 1 1	
	set LivesLabel . Text to () join (Zwick ? get global lives .	
	set SoundFXPlayer . Source to Smashing.mp3	
	call SoundFXPlayer	when PlayAgainSprite . Touched
	I if oet olobalives 0	× y
		do call (initialState -
	then cal gameover -	set PlayAgainSprite . Visible to false
2		
-	-	
E	to gameover	when GameoverClock Timer
do	set BangSprite . Visible to true .	do set BangSprite . Visible to false
	set BallSprite . Speed to 0	set (GameoverClock -) . TimerEnabled - to false -)
	set FlagClock . TimerEnabled . to false	set (PlayAgainSprite .) Visible to [true
	set SoundFXPlayer . Source . to sexplosion.mp3	
	call SoundFXPlayer	

MOLEMASH

set GameoverClock . TimerEnabled to true . set GameSensor . Enabled to false .





PONG



	o (batStart)		
do	call BatimageSprite		
	× I	Screen1 Width / 2 - GatimageSprite Width / 2	
	y I	(PongCanvas • Height • - D (BatlmageSprite •). Height • + (BallSprite •). Heigh	it -

when	PongCanvas
do	set BatimageSprite . X to I get x .
whe	n BallSpritess .CollidedWith
do	If get other BatimageSprite
	then set BallSprite • . Heading • to 1 360 - (BallSprite • . Heading •
whe	en GameoverSprite

A
 Sameoversprite
 Touched
 X
 Y

 do
 call balStart
 set GameoverSprite
 . Visible
 to (false

ΠΑΙΧΝΙΔΙ ΜΕ ΖΑΡΙΑ (DICEGAME)



to throwDice set ResultLabel • . Text • to (• • • • set global dice1 • to (• random integer from (1 to (6 set global dice2 • to (• random integer from (1 to (6	to getPoints to (get x *) set global points * to (get x *) set ResultLabel * . Text * to (join (get global points *) (get global points *)
set global sum • to f get global dice1 • + f get global dice2 • set Dice1Image • . Picture • to f ioin get global dice1 •	to endGame do set ResultLabel . Text to Ι Tέλος παιχνιδιού]
set Dice2Image . Picture to () join (get global dice2 .)	set AccSensor . Enabled to false call Player1 . Vibrate
call ThrowSound • .Play	milliseconds (200

o to checkDicePoints do if 💿 t u get global dicet 🔹 💷 (1) and 🔹 u get global dice2 🔹 💷 (1) or 🔹 u t get global dice1 🔹 💷 (1) and 🔹 u t get global dice2 🔹 💷 (1) call getPoints · 0 call endGame * call Player1 .Start 0 if [(] get global dice1 • EV [] and • (] get global dice2 • EV [] then call getPoints * x (40 get global dice1 v =v } get global dice2 v then call getPoints x (25 else call getPoints · x (get global sum • set global score • to t 😟 t get global score • + t get global points • set ScoreLabel • Text • to t get global score •

ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΜΕ ΠΕΔΙΟ ΚΕΙΜΕΝΟΥ)



ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΠΑΡΑΛΛΑΓΗ ΜΕ ΚΟΥΜΠΙΑ)







ΜΑΘΗΜΑ 4

στοχοι

- Να αναγνωρίζετε τις εντολές επανάληψης και να τις συντάσσετε σωστά.
- Να χρησιμοποιείτε την κατάλληλη εντολή επανάληψης σε κάθε περίπτωση.
- Να χρησιμοποιείτε την τοπική βάση δεδομένων της κινητής συσκευής για «μόνιμη» αποθήκευση δεδομένων.
- Να χρησιμοποιείτε τον αισθητήρα GPS για τον εντοπισμό της θέσης της κινητής συσκευής.
- Να εκκινείτε άλλες εφαρμογές της κινητής συσκευής μέσα από την εφαρμογή σας.
- Να χρησιμοποιείτε έτοιμες βιβλιοθήκες (APIs).

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΕΠΕΚΤΑΣΗ)

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή που υλοποιήθηκε στο προηγούμενο μάθημα σαν δραστηριότητα για το σπίτι έχει το παρακάτω σφάλμα. Υπάρχει περίπτωση στα 2 κουμπιά που εμφανίζονται με τυχαίο τρόπο οι λανθασμένες απαντήσεις να εμφανιστεί η σωστή απάντηση ή να εμφανιστεί η ίδια απάντηση και στα δύο κουμπιά.

Θα επεκτείνουμε την εφαρμογή, προσθέτοντας τους κατάλληλους ελέγχους, ώστε αν ισχύει κάποια από τις περιπτώσεις που αναφέρθηκαν προηγουμένως, η εφαρμογή να δημιουργεί εκ νέου έναν τυχαίο αριθμό για τα κουμπιά με τις λανθασμένες απαντήσεις.



Εικόνα 146 – Το σφάλμα της εφαρμογής Μαθηματικά για παιδιά

βασικές εννοιές

Η εντολή επανάληψης while





Μέχρι τώρα η εφαρμογή μας παράγει έναν τυχαίο αριθμό στο διάστημα [ΣωστήΑπάντηση -5, ΣωστήΑπάντηση + 5] και τον θέτει ως νέα τιμή της μεταβλητής wrongAnswer1, όπως φαίνεται στην παρακάτω εικόνα...



Εικόνα 147 - Η τιμή της μεταβλητής για την πρώτη λανθασμένη απάντηση σε ένα διάστημα τυχαίων αριθμών

Στο σημείο αυτό θα πρέπει να προστεθεί ένας έλεγχος, ώστε στην περίπτωση που ο τυχαίος αριθμός που παράγεται είναι ίσος με τη σωστή απάντηση, η εφαρμογή να τον ξαναζητά.



Πόσες φορές όμως θα πρέπει να γίνεται η παραπάνω ενέργεια;

Η ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ WHILE

Για να πετύχουμε όσα περιεγράφηκαν παραπάνω, θα χρειαστούμε την εντολή **while test** ... **do** (Όσο ισχύει.... κάνε) που βρίσκεται στην ομάδα εντολών Control.



Εικόνα 148 - Η εντολή while



Η εντολή while είναι μια εντολή επανάληψης. Οι εντολές που τοποθετούνται «μέσα» σε αυτήν εκτελούνται ξανά και ξανά, εφόσον συνεχίζει να ισχύει η λογική πρόταση που εξετάζει.

Στο παράδειγμα μας, η εντολή while θα εξετάζει αν η πρώτη λανθασμένη απάντηση έχει την ίδια τιμή με τη σωστή απάντηση. Όσο αυτό ισχύει θα παράγει ξανά έναν τυχαίο αριθμό και θα τον θέτει ως νέα τιμή της πρώτης λανθασμένης απάντησης.

Η λογική πρόταση που θέλουμε να εξετάσουμε φαίνεται παρακάτω...



Εικόνα 149 - Έλεγχος αν η λανθασμένη απάντηση είναι ίση με τη σωστή

... και θα την «κουμπώσουμε» στην εντολή while.



while test 🕻	🗘 get (global correctAnswer 🔹 💷 🕻 get (global wrongAnswer1 🔹
do	

Εικόνα 150 - Έλεγχος με την εντολή while

Η ενέργεια που θα μπει μέσα στην εντολή while είναι η εκ νέου δημιουργία ενός τυχαίου αριθμού για την μεταβλητή wrongAnswer1.

set	global wrongAnswer1 🔹 to 🌘	random integer from L get global correctAnswer - (5) to L get global correctAnswer + (5)
while	e test 📁 🕻 get (global corre	ctAnswer 🔹 💶 🚽 get (global wrongAnswer1 🔹
do	set global wrongAnswer1 -	to 🔓 random integer from 🔓 get global correctAnswer 🔹 - 4,5 to 4 😐 🗘 get global correctAnswer 🔹 + 4,5

Εικόνα 151 - Η ολοκληρωμένη εντολή while



Πόσες είναι οι λιγότερες φορές που μπορούν να εκτελεστούν οι εντολές μέσα στην while;

Βήμα 2 Έλεγχος της δεύτερης λανθασμένης απάντησης

Με παρόμοιο τρόπο θα ελέγξουμε την τιμή και της δεύτερης λανθασμένης απάντησης που αποθηκεύεται στην μεταβλητή wrongAnswer2.

Στο σημείο αυτό πρέπει να γίνει ένας επιπλέον έλεγχος, ώστε η δεύτερη λανθασμένη απάντηση να μην είναι ίδια με την πρώτη λανθασμένη απάντηση.

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα :

set global wrongAnswer2 🔹 to 🚺	random integer from 🔓 get global correctAnswer 🕤 - 🕼 📁 🕴 🖉 get global correctAnswer 🕤 + L 🕤
while test (get global w	rongAnswer1 🕤 💷 🐙 get (global wrongAnswer2 🕤 🔽 🖓 🖓 get (global correctAnswer 🕥 💷 🐙 get (global wrongAnswer2 🕤
do set global wrongAnswer2 •	to C random integer from C get global correctAnswer - (5 to C get global correctAnswer + (5





Q

Θέσε την τιμή της μεταβλητής wrongAnswer2 σε έναν τυχαίο αριθμό στο διάστημα [ΣωστήΑπάντηση - 5, ΣωστήΑπάντηση + 5]

Όσο η τιμή της μεταβλητής είναι ίση με την τιμή της σωστής απάντησης Ή είναι ίση με την τιμή της πρώτης λανθασμένης απάντησης

Θέσε εκ νέου την τιμή της σε έναν τυχαίο αριθμό



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

επεκτάσεις

Κάντε τις απαραίτητες τροποποιήσεις, ώστε η εφαρμογή να υπολογίζει και να εμφανίζει τη βαθμολογία του παίκτη. Για κάθε σωστή απάντηση θα κερδίζει 3 βαθμούς, ενώ για κάθε λανθασμένη απάντηση θα χάνει ένα βαθμό.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ : RANDOM DROPS

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα κατασκευάσουμε μια εφαρμογή απλοποιημένης ζωγραφικής με την τεχνική του dripping, την οποία καθιέρωσε ο Αμερικανός ζωγράφος Jackson Pollock, σημαντικός εκπρόσωπος του Αφηρημένου Εξπρεσιονισμού. Στη συγκεκριμένη εκδοχή, ο χρήστης θα «στάζει» έναν αριθμό από σταγόνες διαφορετικού μεγέθους και χρώματος με τυχαίο τρόπο στον καμβά. Επίσης, όταν αγγίζει το δάκτυλό του στην οθόνη θα «στάζει» στο σημείο επαφής μια σταγόνα τυχαίου χρώματος και μεγέθους, ενώ καθώς σέρνει το δάκτυλό του, θα σχεδιάζονται γραμμές τυχαίου πάχους και χρώματος. Ενδεικτικά στιγμιότυπα της εφαρμογής μας φαίνονται στις επόμενες εικόνες.



Εικόνα 153 – Στιγμιότυπα της εφαρμογής Finger Painting

Ο μεταβολέας (slider) καθορίζει τον αριθμό των σταγόνων που θα πέσουν σε τυχαία σημεία του καμβά, όποτε ταρακουνήσουμε τη συσκευή.

Το κουμπί καθαρισμού (πάνω δεξιά), πέρα του αυτονόητου να καθαρίζει τον καμβά, επιθυμούμε να έχει και μια «κρυμμένη» λειτουργικότητα: όταν πατηθεί παρατεταμένα (long click) θα αλλάζει το χρώμα φόντου του καμβά από άσπρο σε μαύρο και αντίστροφα.

βασικές εννοιές

- Διαδικασία που επιστρέφει τιμή.
- Παραγωγή τυχαίου χρώματος.
- Η εντολή επανάληψης For.



Βήμα1 Εισαγωγή πηγαίου κώδικαγια δημιουργία νέου project

Προκειμένου να εστιάσουμε σε συγκεκριμένα κομμάτια της εφαρμογής, σας διαθέτουμε μια έτοιμη μορφή της βασικής οθόνης της εφαρμογής (GUI), μαζί με ένα μέρος των εντολών που απαιτούνται. Αντί, λοιπόν, να δημιουργήσουμε ένα νέο project, θα εισάγουμε απευθείας στο λογαριασμό μας το αρχείο RandomDrops.aia από τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο RandomDrops.

	Project •	Connect •	Build 🔻	Help y				
	My Projects							
	Start new project							
	Import project (.aia) from my computer							
ALC: NOT	Delete projec	ct						

Προς το παρόν δεν θα χρειαστεί να αλλάξουμε ή να προσθέσουμε κάτι στο τμήμα σχεδίασης της εφαρμογής, οπότε μεταβαίνουμε στο τμήμα συγγραφής του προγράμματος για να μελετήσουμε, να κατανοήσουμε και να επεκτείνουμε τις υπάρχουσες εντολές.



Στο βήμα αυτό θα περιγράψουμε τις απαραίτητες ενέργειες, ώστε όταν ο χρήστης αγγίζει τον καμβά να σχεδιάζεται ένας κύκλος τυχαίου πάχους και χρώματος και όταν «σέρνει» το δάκτυλό του στον καμβά να σχεδιάζεται μια γραμμή τυχαίου πάχους και χρώματος, επίσης.

Οι εντολές που θα χρειαστούμε είναι παρόμοιες με το έργο «Ζωγραφική με τα δάκτυλα» που μελετήσαμε στο 3° Μάθημα. Οι διαφορές βρίσκονται στο τυχαίο πάχος και χρώμα κατά τη σχεδίαση.

Για την παραγωγή του τυχαίου χρώματος σχεδίασης θα χρειαστούμε την εντολή **make color**, η οποία βρίσκεται στην ομάδα εντολών colors.

H	make color	C	make a list	C	255	5
				C	0	
l				C	0	

Εικόνα 154 - Εντολή παραγωγής χρώματος

Όπως φαίνεται από την παραπάνω εικόνα, η εντολή make color πρέπει να κουμπώσει με μια λίστα 3 αριθμών. Οι αριθμοί αυτοί αντιπροσωπεύουν την ποσότητα Κόκκινου, Πράσινου και Μπλε χρώματος που θα αναμιχθούν για να παράγουν το τελικό χρώμα.





Όλα τα χρώματα στον υπολογιστή μπορούν να παραχθούν με την ανάμιξη των 3 βασικών χρωμάτων Red, Green, Blue. Η ποσότητα που θα χρησιμοποιηθεί από το κάθε χρώμα προσδιορίζεται από μια αριθμητική τιμή στο διάστημα 0 εως 255. Από το συνδυασμό των 3 τιμών μπορούν τελικά να παραχθούν 256 * 256 * 256 = 16,7 εκατομμύρια διαφορετικές χρωματικές αποχρώσεις!

Εμείς θα επιλέγουμε τις τιμές των 3 χρωμάτων με τυχαίο τρόπο, όπως φαίνεται στην παρακάτω εικόνα.

🕽 make color 🏮	0	make a list	ď	random integer	from	C <mark>0</mark>	to	(255
			ď	random integer	from	ς0	to	255
			ď	random integer	from	ι Ο	to	255

Εικόνα 155 - Παραγωγή υχαίου χρώματος

Τέλος, θα υλοποιήσουμε μια διαδικασία που θα παράγει το τυχαίο χρώμα, όπως παραπάνω και θα το επιστρέφει σαν αποτέλεσμα, ώστε να χρησιμοποιηθεί από την εντολή που την κάλεσε.

ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΠΟΥ ΕΠΙΣΤΡΕΦΟΥΝ ΤΙΜΗ

Μέχρι τώρα οι διαδικασίες που υλοποιήσαμε έκαναν κάποιες λειτουργίες όταν τις καλούσαμε, χωρίς όμως να επιστρέφουν κάποιο αποτέλεσμα. Συχνά χρειαζόμαστε διαδικασίες που μετά την κλήση τους θα επιστρέφουν μια τιμή σαν αποτέλεσμα.



Στη συνέχεια δίνεται ολοκληρωμένη η διαδικασία παραγωγής ενός τυχαίου χρώματος.

💿 to 📭	oickRandomColo	DT .	
result (make color 🏮	💿 make a list 🏼	random integer from (0 to (255
		0	random integer from (0 to (255)
		0	random integer from (0 to (255







Παρήγαγε και επέστρεψε ένα χρώμα, συνδυάζοντας τυχαία κάποιες από τις 256 τυχαίες αποχρώσεις του κόκκινου, πράσινου και μπλε.

Στη συνέχεια για να καλέσουμε τη διαδικασία και να πάρουμε το αποτέλεσμα της θα κουμπώσουμε την εντολή κλήσης στην ιδιότητα που θέτει το χρώμα σχεδίασης του καμβά.

set	Canvas1	•	. PaintColor •	to 🔰	call	pickRandomColor •

Εικόνα 157 - Ορισμός χρώματος σχεδίασης του καμβά

Παρακάτω δίνονται οι ενέργειες που θα εκτελούνται κατά το άγγιγμα και το σύρσιμο του καμβά.

when Canvas1 . TouchDown	when Canvas1 . Dragged
X Y	startX startY prevX prevY currentX currentY draggedSprite
do set Canvas1 . PaintColor to call pickRandomColor .	do set Canvas1 . PaintColor to Call pickRandomColor
call Canvas1 . DrawCircle	set Canvas1 . LineWidth . to random integer from (1) to (4)
x l <mark>get x *</mark>	call Canvast Drawline
y l <mark>get y v</mark>	x1 (get prevX -
r (random integer from (1 to (12)	v1 get prevY
	x2 (get currentX •
	y2 (get currentY -

Εικόνα 158 - Σχεδίαση κύκλων ή γραμμών με άγγιγμα ή σύρσιμο του δακτύλου αντίστοιχα.

Βήμα 3 Καθαρισμός και αλλαγή του χρώματος τουκαμβά

Το πάτημα του κουμπιού ClearButton καθαρίζει τον καμβά, καλώντας την ενσωματωμένη διαδικασία Clear που διατίθεται στις εντολές του Canvas1.



Εικόνα 159 - Καθαρισμός του καμβά

Σκοπός μας είναι να επεκτείνουμε την λειτουργία του καθαρισμού, ώστε με παρατεταμένο πάτημα του κουμπιού Clear, να αλλάζει χρώμα το φόντο του καμβά από άσπρο σε μαύρο και αντίστροφα.

when	ClearButton 🔹	.LongClick
do		

Εικόνα 160 - Το γεγονός παρατεταμένου πατήματος του κουμπιού Clear



Ποιες εντολές θα κουμπώσουν μέσα στο πλακίδιο **When ClearButton.Long-**Click;



Βήμα4 Ρίξιμο τυχαίωνσταγόνων στον καμβά

Η θέση του μεταβολέα καθορίζει πόσες σταγόνες θα πέσουν στον καμβά, μόλις ταρακουνήσουμε τη συσκευή.

Καταρχάς, οι παρακάτω εντολές αναλαμβάνουν να ενημερώνουν κατάλληλα την ετικέτα κειμένου DropsLabel με βάση την τρέχουσα θέση ThumbPosition του μεταβολέα DropsSlider.

when Screen1 .Initialize			
do set DropsLabel . Text to b	round 🔹 🖡	DropsSlider .	ThumbPosition •
when DropsSlider PositionChanged			
thumbPosition			
unumbr oblight			



2	Κατά την εκκίνηση της εφαρμογής, αλλά και σε οποιαδήποτε αλλαγή της θέ- σης του μεταβολέα DropsSlider, όρισε σαν κείμενο της ετικέτας DropsLabel, τον αριθμό που αντιστοιχεί στην τρέχουσα θέση του.		
Η εντολή round στρογγυλοποιεί τις πιθανές δεκαδικές θέσεις που βρία ο δείκτης του μεταβολέα, στον κοντινότερο ακέραιο αριθμό.			
	Για παράδειγμα η εντολή		
	round (3.42) θα επιστρέψει την τιμή 3,		
	ενώ η εντολή (round) (3.67) θα επιστρέψει την τιμή 4.		

Αυτό που χρειάζεται να κάνουμε είναι να σχεδιάζονται τόσοι κύκλοι, όσοι καθορίζονται από την τιμή που δείχνει ο μεταβολέας, ο καθένας σε κάποιο τυχαίο σημείο εντός του καμβά.

Ας ξεκινήσουμε από τη σχεδίαση ενός κύκλου, ο οποίος για να έχει τυχαίο χρώμα και μέγεθος, αλλά και για να «πέσει» εντός του καμβά, χρειάζεται να σχεδιαστεί ως εξής:



Εικόνα 162 - Τυχαία σχεδίαση ενός κύκλου





Δώσε ένα τυχαίο χρώμα ως χρώμα βαφής για τον καμβά.

Σχεδίασε έναν κύκλο:

- σε τυχαία θέση στον οριζόντιο άξονα x (οπουδήποτε από το σημείο
 Ο μέχρι και το μήκος του καμβά).
- σε τυχαία θέση στον κατακόρυφο άξονα y (οπουδήποτε από το σημείο 0 μέχρι και το ύψος του καμβά).
- με μέγεθος ακτίνας έναν τυχαίο αριθμό από το 1 μέχρι και το 12.

Θα χρειαστεί στη συνέχεια να επαναλάβουμε τις παραπάνω εντολές, ξανά και ξανά, με βάση την τιμή DropsSlider.ThumbPosition.



Πώς θα καταφέρουμε να καλέσουμε το παραπάνω σετ εντολών, όσες φορές χρειαστεί, χωρίς φυσικά να τις αντιγράψουμε, αφού δεν γνωρίζουμε που βρίσκεται κάθε φορά ο δείκτης του μεταβολέα;

Η ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ FOR

Για διευκόλυνσή μας, αλλά και για ευκολότερη ανάγνωση και διόρθωση του κώδικα, όταν θέλουμε κάποιες εντολές να εκτελεστούν πολλές φορές (με εκ των προτέρων γνωστό πλήθος επαναλήψεων), μπορούμε να χρησιμοποιήσουμε την εντολή επανάληψης Για (For) από την ομάδα εντολών Control.



Εικόνα 163 - Η εντολή επανάληψης For



Για κάθε αριθμό από 1 μέχρι 5, με βήμα 1, εκτέλεσε τις εντολές που αναφέρονται.

Προφανώς η αρχική τιμή, η τελική τιμή και το βήμα αύξησης ή μείωσης μεταβάλλονται κατ' επιλογή του χρήστη.

Αξιοποιώντας κατάλληλα τα παραπάνω, φτιάξτε μια διαδικασία drawRandomCircles που θα δέχεται ως παράμετρο τον αριθμό που αντιστοιχεί στην τρέχουσα θέση του μεταβολέα και θα καλεί το σετ εντολών τυχαίας σχεδίασης ενός κύκλου (βλ. εικόνα 77), όσες φορές απαιτείται με τη βοήθεια της εντολής For.



Ολοκληρωμένος, ο κώδικας της εφαρμογής μας φαίνεται παρακάτω:

when Screen1 . Initialize	
do set DropsLabel . Text to round DropsSlider . ThumbPosition	when ClearButton Click
when DropsSlider PositionChanged ThumbPosition do set DropsLabel Text to Tound DropsSlider ThumbPosition	do call Canvas1 • Clear when ClearButton • LongClick do O if Canvas1 • BackgroundColor • = • (
when AccSensor . Shaking	then set Canvas1 . BackgroundColor . to
do call drawRandomCircles drops (round DropsSlider ThumbPosition	else set Canvas1 • . (BackgroundColor • to)
to drawRandomCircles drops to pickR	RandomColor
do for each number from	ake color 1 💿 make a list 🔓 random integer from 1 🕕 to 1 255
to L get drops •	random integer from [0] to [255]
do set Canvas1 . PaintColor to call pickRandomColor	random integer from 0 to 255
call Canvas1 . DrawCircle	
× I random integer from I 0 to I Canvas1 • . V	Vidth -
y random integer from (0 to (Canvas1 • . H	leight
r (random integer from (1 to (12)	
when Car	nvas1 Dragged

whe	n Canvas1 🕥 .TouchDown	st	tartX startY prevX prevY currentX currentY draggedSprite
x	(Y)	do	set Canvas1 . PaintColor . to Call pickRandomColor
do	set Canvas1 . PaintColor . to Call pickRandomColor		set Canvas1 . LineWidth . to random integer from (1 to (4
	call Canvas1 .DrawCircle		call Canvas1 . DrawLine
	x L get x 🔪		x1 (get prevX v)
	y 🔓 get (y 🔪		y1 (get prevY v
	r I random integer from (1) to (12)		x2 (get currentX -
			y2 (get currentY v

Εικόνα 164 - Ο κώδικας της εφαρμογής Random Drops



ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΛΕΞΙΚΟ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα υλοποιήσουμε ένα ελληνο - γερμανικό λεξικό. Ο χρήστης του λεξικού θα έχει τη δυνατότητα να προσθέτει νέες λέξεις στα ελληνικά καθώς και την μετάφρασή τους. Επίσης, όταν επιλέγει μια από τις ελληνικές λέξεις θα βλέπει τη μετάφρασή της στα γερμανικά και θα ακούει την προφορά της γερμανικής λέξης από τη συσκευή του.

	🖫 📶 🕼 7:37 рм
Λεξικό	and the second second
Λεξικό	Νέα λέξη
Καλημέρα	Guten morgen
	())
Ελληνικά	Τερμανικά
Αποθήκευση	Ακύρωση

Εικόνα 165 - Η εφαρμογή "Λεξικό"

βασικές εννοιές

- Λίστες (Lists) και Κουμπί επιλογής στοιχείου από λίστα (ListPicker)
- Βάσεις Δεδομένων (TinyDB)
- Μετατροπή κειμένου σε ομιλία (TTS Text to Speech)

ια 1 Δημιουργίατου έργου και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα νέο έργο με όνομα Dictionary. Ας τροποποιήσουμε ορισμένες ιδιότητες της οθόνης, ως εξής:

\$	επιλέγουμε το συστατικό	μεταβάλλουμε τις ιδιότητες
	Screen1	Title : Λεξικό
		Background color : Blue



Designer Γραφικό Περιβάλλον Εφαρμογής

ήμα 2 Προσθήκη των απαραίτητων αρχείωνπολυμέσων

Ας προσθέσουμε στο project μας ένα αρχείο εικόνας που θα χρησιμοποιεί η εφαρμογή μας. Επισκεφθείτε τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο Dictionary.

+	ανεβάζουμε το αρχείο με όνομα	σύντομη περιγραφή	
	Speaker.png	Εικόνα ηχείου	

ΒήμαΞ

Προσθήκη αντικειμένωνστο γραφικό περιβάλλον

Θα χρειαστεί να εισάγουμε στην εφαρμογή μας τα αντικείμενα που περιγράφονται παρακάτω.

από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	HorizontalAlignment	ButtonsArea	AlignHorizontal : center
			Width : fill Parent Height : 80 pixels
UserInterface	ListPicker	WordListPicker	Background Color : Orange
			FontBold : Checked
			FontSize : 18
			Shape : rounded
			Text : Λεξικό
			Text Color : White
			TextAlignment : center
UserInterface	Label	PaddButtonLabel	Text :
			Width : 100 pixels
UserInterface	Button	AddWordButton	Background Color : Green
			FontBold : Checked
			FontSize : 18
			Shape : rounded
			Text : Λεξικό
			Text Color : White
			TextAlignment : center
Layout	HorizontalAlignment	TranslationArea	AlignHorizontal : center

			Width : fill Parent
			Height : 80 pixels
UserInterface	label	Greeki abel	FontBold · checked
		0.00.1100	FontSize · 20
llserInterface	label	Paddi abel	Toyt :
Osermeriace	Laber	raddeaber	Width : 50 pixels
Licorintorfaco	Labal	Cormoniohol	FontBold - shoeled
Oserinterrace	Label	GermanLaber	
	.		FontSize : 20
UserInterface	Button	ListenButton	Image : speaker.png
			Width : 64 pixels
			Height : 64 pixels
			Visible : hidden
Layout	HorizontalAlignment	AddWordArea	AlignHorizontal : center
			Width : fill Parent
			Height : 100 pixels
			Visible : hidden
UserInterface	TextBox	GreekTextBox	Hint : Ελληνικά
			Text :
			Multiline : checked
UserInterface	TextBox	GermanTextBox	Hint : Γερμανικά
			Text :
			Multiline : checked
Layout	HorizontalAlignment	InsertArea	AlignHorizontal : center
			Width : fill Parent
			Height : 80 pixels
			Visible : hidden
UserInterface	Button	InsertButton	Background Color : Yellow
			FontBold · Checked
			FontSize : 18
			Shape : rounded
			Text : Αποθήκεμση
			Text Color : White
			Text Alignment : center
UserInterface	Lahel	Paddl abel?	Tovt ·
Usermenace	Lavei		Width : 50 nivels
licerinterface	Button	ClearPutton	Pooleground Colors Corr
Usermende	Button		Background Color : Cyan
			FontSize : 18
			Shape : rounded
			Text : Ακύρωση



				Text Color : White
				TextAlignment : center
	Storage	TinyDB	DictionaryDB	
	Media	TextToSpeech	GermanTTS	Country : deu
				Language : DEU
Συγγρα	ιφή Προγρό	αμματος ΒΙα	ocks	

Όταν ο χρήστης αγγίζει το κουμπί «Νέα Λέξη», η εφαρμογή θα του εμφανίζει 2 περιοχές που περιέχουν:

α) τα πεδία κειμένου για να γράψει τις λέξεις στα ελληνικά και στα γερμανικά και

β) τα κουμπιά για την καταχώρηση της νέας λέξης ή την ακύρωση της ενέργειας.



Εικόνα 166 - Εμφάνιση των περιοχών για την καταχώρηση νέας λέξης

Βήμα 5 Ακύρωσητης καταχώρισης

Όταν ο χρήστης αγγίξει το κουμπί «Ακύρωση», θα πρέπει τα 2 πεδία κειμένου με την ελληνική και τη γερμανική λέξη να «καθαρίζονται» και οι 2 περιοχές να κρύβονται ξανά. Επειδή, παρόμοιες ενέργειες θα χρειαστεί να γίνουν και στην περίπτωση που πατηθεί το κουμπί «Αποθήκευση» είναι καλύτερο να δημιουργήσουμε **μια διαδικασία** και να τοποθετήσουμε τις ενέργειες σε αυτή. Στη συνέχεια θα την καλέσουμε κατάλληλα, όπου χρειάζεται.



Εικόνα 167 - Διαδικασία για το καθάρισμα των πεδίων κειμένου και την απόκρυψη των περιοχών εισαγωγής λέξης





Εικόνα 168 - Όταν πατηθεί το κουμπί "Ακύρωση"

Βήμα**6**

Αποθήκευσηνέας λέξης στο λεξικό

Όταν ο χρήστης αγγίξει το κουμπί «Αποθήκευση» η εφαρμογή θα πρέπει να ενημερώνει το λεξικό με την καινούρια λέξη και την μετάφρασή της.



Με ποιον τρόπο η εφαρμογή μας θα κρατάει όλες τις λέξεις που θα αποθηκεύει ο χρήστης στο λεξικό;

Μια πρώτη σκέψη είναι να χρησιμοποιήσουμε μεταβλητές για την καταχώριση των λέξεων. Αυτό δεν μπορεί να λειτουργήσει σωστά, διότι η μεταβλητή κρατάει μια μόνο τιμή, ενώ η εφαρμογή μας θα πρέπει να διαχειρίζεται πολλές τιμές (όσες και οι λέξεις στο λεξικό).

ΛΙΣΤΕΣ (LIST)



Όταν θέλουμε να καταχωρίσουμε ταυτόχρονα πολλές τιμές σε ένα αντικείμενο, ώστε να αναφερόμαστε σε αυτές χρησιμοποιώντας ένα κοινό όνομα, τότε χρησιμοποιούμε ένα στοιχείο που ονομάζεται λίστα.

Οι λίστες μας δίνουν πολλές δυνατότητες σε σύγκριση με τις μεταβλητές, αφού μπορούμε εύκολα να εισάγουμε ή να διαγράψουμε τιμές από τη λίστα, να διαλέξουμε μια τυχαία τιμή της ή να αναζητήσουμε τιμές με βάση τη θέση τους. Παρακάτω δίνονται μερικές από τις πιο συχνά χρησιμοποιούμενες εντολές στις λίστες :







Στο διπλανό σχήμα απεικονίζεται μια λίστα με όνομα Names. Η λίστα περιέχει 5 τιμές («Γιάννης», «Μαρία», «Ελένη», «Κώστας» και «Αργυρώ»). Η κάθε τιμή διαφέρει από την άλλη εξαιτίας της θέσης που βρίσκεται μέσα στη λίστα (αριθμοί 1 έως 5 αντίστοιχα).

Αν για παράδειγμα εκτελέσουμε την εντολή



θα πάρουμε την τιμή «Μαρία» (αφού αυτή η τιμή βρίσκεται στη θέση 2)

Εικόνα 169 = Παράδειγμα Λίστας

Προκειμένου, λοιπόν, να δημιουργήσουμε τις απαραίτητες λίστες για την εφαρμογή μας, από την ομάδα Variables δημιουργούμε δύο νέα συμβολικά ονόματα, όπως παρακάτω...



Εικόνα 170 - Τα ονόματα που θα έχουν οι λίστες της εφαρμογής

... και στη συνέχεια από την ομάδα αντικειμένων Lists τους αναθέτουμε την τιμή

Create empty list

Εικόνα 171 - Εντολή δημιουργίας κενής λίστας

Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα.



Εικόνα 172 - Δημιουργία δύο κενών λιστών



Δημιούργησε δύο κενές λίστες με ονόματα germanWords και greekWords αντίστοιχα.



Στη συνέχεια θα χρησιμοποιήσουμε την εντολή **add items to list** ... από την ομάδα αντικειμένων Lists...



Εικόνα 173 - Εντολή προσθήκης στοιχείου σε λίστα

... για να προσθέτουμε στη λίστα με τις ελληνικές λέξεις, τη νέα λέξη που έγραψε ο χρήστης στο αντίστοιχο πεδίο κειμένου.



Εικόνα 174 - Προσθήκη λέξης στην αντίστοιχη λίστα

Προφανώς θα κάνουμε ακριβώς το ίδιο για τη λέξη στα γερμανικά.

Καλό είναι να προσθέσουμε και έναν έλεγχο πριν βάλουμε τις λέξεις στη λίστα ότι ο χρήστης έχει πληκτρολογήσει κάποιο κείμενο στα αντίστοιχα πεδία και δεν τα έχει αφήσει κενά.

Οι ενέργειες που θα ενεργοποιηθούν με το πάτημα του κουμπιού «Αποθήκευση» φαίνονται παρακάτω.



Εικόνα 175 - Άγγιγμα του κουμπιού "Αποθήκευση"

Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή. Χρησιμοποιήστε ως παράδειγμα για την καταχώριση τις λέξεις: «Καλημέρα» - «Guten Morgen» «Καλησπέρα» - «Guten Abend»

α 7 Επιλέγοντας απότο λεξικό

Όταν ο χρήστης αγγίζει το κουμπί «Λεξικό» θα εμφανίζεται μια λίστα με όλες τις λέξεις που έχουν καταχωρηθεί στην εφαρμογή. Μόλις ο χρήστης αγγίξει μια από τις λέξεις θα του εμφανίζεται στις αντίστοιχες ετικέτες μαζί με τη μετάφρασή της.





Το κουμπί «Λεξικό» διαφέρει από κοινά κουμπιά που έχουμε χρησιμοποιήσει μέχρι τώρα γιατί με το άγγιγμα του η εφαρμογή εμφανίζει μια λίστα με στοιχεία. Στην πραγματικότητα, αν ανατρέξουμε λίγο παραπάνω θα παρατηρήσουμε ότι το κουμπί «Λεξικό» είναι ένα αντικείμενο ListPicker δηλαδή ένα αντικείμενο που παρουσιάζει μια σειρά από τιμές σε μορφή λίστας.

Η πρώτη μας δουλειά είναι να καθορίσουμε τη λίστα από την οποία θα παίρνει το αντικείμενο ListPicker τις τιμές που θα εμφανίσει. Για το σκοπό αυτό χρησιμοποιούμε την εντολή...



Εικόνα 176 - Εντολή ανάθεσης τιμών σε αντικείμενα λίστας

... του αντικειμένου ListPicker και την κουμπώνουμε με το όνομα της λίστας από την οποία θα τραβήξει τιμές.



```
Εικόνα 177 - Εμφάνιση τιμών από τη λίστα με τις ελληνικές λέξεις
```

Ένα σημείο στο οποίο πρέπει να θέτουμε τις τιμές που θα εμφανίζει το αντικείμενο ListPicker είναι στην εκκίνηση της εφαρμογής, όπως παρακάτω.



Φυσικά, η ίδια εντολή πρέπει να προστεθεί κάθε φορά που καταχωρούμε μια νέα λέξη στη λίστα, δηλαδή μέσα στο γεγονός του κουμπιού «Αποθήκευση».



Εικόνα 178 - Ενημέρωση της λίστας με τη νέα λέξη



Μόλις ο χρήστης επιλέγει μια τιμή από το αντικείμενο ListPicker ενεργοποιείται αυτόματα το γεγονός when[....].AfterPicking του αντικειμένου.

when	WordListPicker . AfterPicking
do	

Εικόνα 179 - Γεγονός που ενεργοποιείται μόλις επιλεγεί τιμή από τη λίστα

Θα το χρησιμοποιήσουμε για να εμφανίσουμε στις 2 ετικέτες την ελληνική λέξη που επέλεξε και την αντίστοιχη λέξη στα γερμανικά.



Με ποιον τρόπο σχετίζονται οι δύο λίστες μεταξύ τους; Πώς, επιλέγοντας τη λέξη από τη λίστα με τα ελληνικά, θα μπορέσουμε να βρούμε την αντίστοιχή λέξη στα γερμανικά;



Οι τιμές που εισάγουμε σε μια λίστα ξεχωρίζουν μέσω της θέσης τους στη λίστα. Για παράδειγμα, η τιμή που εισάγουμε πρώτη έχει σαν θέση τον αριθμό 1, η τιμή που βάζουμε δεύτερη έχει σαν θέση τον αριθμό 2 κοκ.

Επομένως όταν βάζουμε παράλληλα τιμές σε 2 λίστες τότε τα αντίστοιχα στοιχεία τους χαρακτηρίζονται από την ίδια θέση.

Επομένως αν για παράδειγμα η πρώτη τιμή που καταχωρούμε είναι το «Καλημέρα» και αντίστοιχα το «Guten Morgen» στις 2 λίστες (ελληνικά και γερμανικά αντίστοιχα) τότε βρίσκονται και οι δύο στη θέση 1 μέσα στις λίστες τους.

Το αντικείμενο ListPicker μας δίνει δύο πολύ χρήσιμες εντολές. Η πρώτη εντολή είναι η τιμή του στοιχείου που επιλέχθηκε...

WordListPicker 🔹 Selection •

Εικόνα 180 - Η τιμή του επιλεγμένου στοιχείου

... ενώ η δεύτερη μας δίνει τη θέση του στοιχείου που επιλέχθηκε.

WordListPicker • . SelectionIndex •

Εικόνα 181 - Η θέση του στοιχείου που επιλέχθηκε

Η πρώτη εντολή θα μας χρειαστεί για να θέσουμε την τιμή της ετικέτας με την ελληνική λέξη...

set GreekLabel • . Text • to WordListPicker • . Selection •

Εικόνα 182 - Τιμή της ετικέτας με την ελληνική λέξη

Για να θέσουμε την τιμή της ετικέτας με τη γερμανική λέξη θα πρέπει να επιλέξουμε την τιμή που βρίσκεται στην ίδια θέση από τη λίστα με τις γερμανικές λέξεις. Θα μας χρειαστεί η εντολή που βρίσκεται στην ομάδα αντικειμένων Lists...



Εικόνα 183 - Εντολή επιλογής τιμής από λίστα

... στην οποία θα κουμπώσουμε το όνομα της λίστας με τις γερμανικές λέξεις και τη θέση του αντικειμένου που επιλέχθηκε...

set GermanLabel 🔹 . Text 🔹 to 🖡	select list item list 📕	get global germanWords -
	index I,	WordListPicker • . SelectionIndex •

Εικόνα 184 - Τιμή της ετικέτας με τη γερμανική λέξη

Τέλος για να επιτρέψουμε στον χρήστη να ακούει τη λέξη στα γερμανικά θα κάνουμε ορατό το κουμπί με το ηχείο, μόλις ο χρήστης επιλέγει τιμή από τη λίστα.

Ο ολοκληρωμένος κώδικας που εκτελείται με την επιλογή τιμής από το Λεξικό φαίνεται στην παρακάτω εικόνα.



Εικόνα 185 - Γεγονός επιλογής από λίστα



Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή.

Βήμα**8**

Ακούγοντας τη λέξη!

Όταν ο χρήστης αγγίζει το κουμπί με την εικόνα του ηχείου, η κινητή συσκευή θα του διαβάζει τη λέξη στα γερμανικά.

Για να το πετύχουμε αυτό θα χρησιμοποιήσουμε το αντικείμενο TextToSpeech που έχουμε εισάγει στην εφαρμογή μας.



Το αντικείμενο TextToSpeech (TTS) μετατρέπει ένα κείμενο σε λόγο. Το μόνο που χρειάζεται είναι να του καθορίσουμε τις ιδιότητες χώρα και γλώσσα




Οι εντολές που για την ενεργοποίηση του αντικειμένου TTS φαίνονται παρακάτω :



Εικόνα 186 - Διάβασμα της λέξης από την κινητή μας συσκευή



Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή.

Βήμα9 Μόνιμη αποθήκευση στοιχείων στην κινητή συσκευή

Όταν ο χρήστης αγγίξει το κουμπί Αποθήκευση θα πρέπει η νέα λέξη που έχει γράψει να αποθηκεύεται στην κινητή συσκευή, ώστε να μην χρειάζεται να καταγράφει κάθε φορά όλες τις λέξεις από την αρχή.

Για να αποθηκεύσουμε μόνιμα κάποιες τιμές στη συσκευή μας χρειαζόμαστε ένα αντικείμενο που λέγεται Βάση Δεδομένων (TinyDB και TinyWebDB από την ομάδα Storage).



Η βάση δεδομένων είναι ένας οργανωμένος τρόπος αποθήκευσης. Η κάθε τιμή που αποθηκεύεται έχει μια δική της ξεχωριστή ονομασία με την οποία θα μπορούμε αργότερα να τη «βρούμε» στη βάση δεδομένων που την αποθηκεύσαμε.

Για να αποθηκεύσουμε μια τιμή ή μια λίστα τιμών, όπως στην περίπτωσή μας στη βάση δεδομένων χρησιμοποιούμε την εντολή....



Εικόνα 187 - Εντολή αποθήκευσης τιμών στη Βάση Δεδομένων



... του αντικειμένου DictionaryDB. Στην εντολή αυτή θα κουμπώσουμε την ονομασία που θέλουμε να έχει το αντικείμενο που θα αποθηκεύσουμε στη βάση (tag), καθώς και την τιμή ή τη λίστα τιμών που θα αποθηκεύσουμε (valueToStore).

Για παράδειγμα μπορούμε να αποθηκεύσουμε τη λίστα με τις ελληνικές λέξεις στη ΒΔ δίνοντας της την ονομασία greekwords, όπως φαίνεται παρακάτω...



Εικόνα 188 - Αποθήκευση λίστας με ελληνικά στη ΒΔ

Με τον ίδιο ακριβώς τρόπο θα αποθηκεύσουμε στη ΒΔ τη λίστα με τις γερμανικές λέξεις. Ας της δώσουμε την ονομασία germanWords.



Εικόνα 189 - Αποθήκευση λίστας με γερμανικα στη ΒΔ

when InsertButton . Click do 0 i and * GreekTextBox . Text . (#**) GermanTextBox -Text 🖌 🗲 get global greekWords * then 🔲 add items to list 🛛 list GreekTextBox Text item add items to list list get global germanWords GermanTextBox • Text call DictionaryDB . StoreValue tao greekWords valueToStore get global greekWords call DictionaryDB . StoreValue germanWords tag valueToStore get global germanWords • set WordListPicker . Elements to get global greekWords call clear

Τέλος θα τοποθετήσουμε τις εντολές αυτές μέσα στο γεγονός της αποθήκευσης νέας λέξης.

Εικόνα 190 - Αποθήκευση νέας λέξης στη ΒΔ



Βήμα 10 Ανάκτηση αποθηκευμένων τιμών από την ΒΔ

Κατά την εκκίνηση της εφαρμογής μας θα πρέπει να κάνουμε ανάκτηση των τιμών που «πιθανόν» έχουν αποθηκευτεί στη βάση δεδομένων από τον χρήστη και να τις «φορτώνουμε» στις λίστες με τα ελληνικά και τα γερμανικά.

Για να το πετύχουμε θα χρειαστούμε την εντολή...



Εικόνα 191 - Εντολή ανάκτησης τιμών από την ΒΔ

... στην οποία θα κουμπώσουμε την ονομασία της τιμής που θέλουμε να ανακτήσουμε (πχ. greekWords) και μια τιμή που θα επιστραφεί σε περίπτωση που δεν υπάρχει η τιμή που προσπαθούμε να ανακτήσουμε στη ΒΔ.

Οι τιμές αυτές θα δοθούν στις λίστες greekWords και germanWords της εφαρμογής.

whe	en Screen1 v Initialize		
do	set global greekWords • to 🕻 call DictionaryDB • .GetValue	e	
	ta	, d <mark>e</mark>	greekWords *
	valuelfTagNotThere		create empty list
	set global germanWords T to C call DictionaryDB .GetVa	ue	
		ag 🌗	* germanWords) *
	valuelfTagNotThe	ere 🌔	create empty list
	set WordListPicker • . Elements • to pget global greekV	/ords	



	•	

Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

Ολοκληρωμένη, η εφαρμογή μας αποτελείται από τις παρακάτω εντολές:



initialize global germanWords to 🚺 🔲 create empty list

initialize global greekWords to 🖡 🔲 create empty list

whe	n Screen1 . Initialize	
do	set global greekWords 🔹 to 🔓 call (DictionaryDB 🔹 .GetValue	
	tag 🖡	* greekWords *
	valuelfTagNotThere	create empty list
	set global germanWords • to 🕻 call DictionaryDB • .GetValue	
	tag	C germanWords *
	valuelfTagNotThere	🕻 🔲 create empty list
	set (WordListPicker •). Elements •) to pet global greekWord	S 🔨

whe	n (Ac	IdWordButton 🔪 .Click
do	set	AddWordArea 🔹 . Visible 🔹 to 🕴 true 🔹
	set	InsertArea 🔹 . (Visible 🔹 to 🕴 true 💌

when		Cle	earButton	•	.Click
do	C	all	clear 🔹		

	to Cle	ear
do	set	GreekTextBox 🔹 . (Text 🔹 to 🕌 🦲 📲
	set	GermanTextBox 🔹 . (Text 🔹 to 🖊 🍎 🦉
	set	AddWordArea 🔹 . Visible 🔹 to 🔓 false 🔹
	set	InsertArea •). Visible •) to 🔰 false •)





whe	n Inse	Button Click
do	if 🛛	GreekTextBox • . Text • 🗲 • • • • • • • • • • • • • • • • •
	then	🔲 add items to list 🛛 list 🖉 get global greekWords 🔹
		item GreekTextBox Text -
		🔲 add items to list 🛛 list 🖉 get global germanWords 🔹
		item (GermanTextBox -). Text -
		call DictionaryDB .StoreValue
		tag (greekWords)
		valueToStore le get global greekWords
		call DictionaryDB - StoreValue
		tag (germanWords)
		valueToStore 🚺 get global germanWords 🚽
		set WordListRicker 🕤 . Elements 🕤 to 🔓 get global greekWords 🔹
		call clear •

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : WHERE IS MY BIKE?

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα αναπτύξουμε μια εφαρμογή που θα μας επιτρέπει να αποθηκεύουμε τη διεύθυνση και τις συντεταγμένες της τοποθεσίας μας, όπου λ.χ. παρκάραμε το ποδήλατό μας και αξιοποιώντας τις δυνατότητες του δέκτη GPS και της υπηρεσίας Google Maps, θα μπορούμε να λάβουμε οδηγίες για το πώς θα επιστρέψουμε στην αποθηκευμένη τοποθεσία.

	9:51 рм
Where is my Bike?	
Τρέχουσα τοποθεσία Διεύθυνση: GPS: 0, 0	1
Αποθηκευμένη τοπο Διεύθυνση: GPS: 0, 0 Σημείωση: -	θεσία
Οδηγίες διαδρομής	

Εικόνα 193 - Στιγμιότυπο της εφαρμογής κατά την εκκίνησή της



Στο πάνω μέρος της οθόνης εμφανίζεται η τρέχουσα θέση μας (διεύθυνση και γεωγραφικές συντεταγμένες), ενώ όταν ο δέκτης GPS είναι σε θέση να προσδιορίσει την τοποθεσία μας, ενεργοποιείται το κουμπί «Να θυμάσαι αυτή τη θέση», που μας επιτρέπει να αποθηκεύσουμε τη θέση μας στη βάση δεδομένων της εφαρμογής και να την εμφανίσουμε στο κάτω μέρος της οθόνης.

Όποτε το θελήσουμε, πατώντας το κουμπί «Οδηγίες διαδρομής», μεταφερόμαστε στην εφαρμογή Google Maps, όπου αυτόματα, με βάση τα γεωγραφικά στοιχεία, μπορούμε να πλοηγηθούμε στον προορισμό μας, στο σημείο δηλαδή που αποθηκεύσαμε πριν (εκεί που βρίσκεται το ποδήλατό μας).

βασικές εννοιές

- Βάση Δεδομένων (TinyDB)
- Αισθητήρας τοποθεσίας (Location Sensor)
- Εκκινητής δραστηριοτήτων (Activity Starter)

Βήμα1 Εισαγωγή πηγαίου κώδικαγια δημιουργία νέου project

Προκειμένου να εστιάσουμε σε συγκεκριμένα κομμάτια της εφαρμογής, σας διαθέτουμε μια έτοιμη μορφή της βασικής οθόνης της εφαρμογής (GUI), μαζί με ένα μέρος των εντολών που απαιτούνται. Αντί, λοιπόν, να δημιουργήσουμε ένα νέο project, θα εισάγουμε απευθείας στο λογαριασμό μας το αρχείο WherelsMyBike.aia από τη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο WherelsMyBike.



Προς το παρόν δεν θα χρειαστεί να αλλάξουμε ή να προσθέσουμε κάτι στο τμήμα σχεδίασης της εφαρμογής, οπότε μεταβαίνουμε στο τμήμα συγγραφής του προγράμματος για να μελετήσουμε, να κατανοήσουμε και να επεκτείνουμε τις υπάρχουσες εντολές.



Προκειμένου να αξιοποιήσουμε τις δυνατότητες του GPS στην εφαρμογή μας χρειαζόμαστε το αντικείμενο LocationSensor, το οποίο έχει ήδη προστεθεί στην περιοχή σχεδίασης.

Έχουμε λοιπόν αμέσως πρόσβαση σε λειτουργίες που σχετίζονται με το στίγμα της συσκευής, εφόσον ο δέκτης GPS είναι ενεργός και λαμβάνει επαρκές σήμα.





Προκειμένου να καταφέρει ο αισθητήρας να εντοπίσει τη θέση μας, θα πρέπει ιδανικά, να βρισκόμαστε σε εξωτερικό χώρο ώστε να υπάρχει οπτική επαφή με τουλάχιστον τρεις δορυφόρους του συστήματος GPS ή εναλλακτικά να έχουμε ενεργοποιήσει τη λειτουργία ανίχνευσης της θέσης της συσκευής μέσω ασύρματων δικτύων (οπότε απαιτείται ενεργή πρόσβαση στο Διαδίκτυο.

Θα αξιοποιήσουμε την παρακάτω εντολή ...



Εικόνα 194 - Γεγονός αλλαγής γεωγραφικής θέσης

... η οποία ανιχνεύει οποιαδήποτε αλλαγή στη γεωγραφική θέση της συσκευής. Το γεγονός αυτό πυροδοτείται ακόμα και όταν ο αισθητήρας λάβει σήμα από τους δορυφόρους για πρώτη φορά, κάτι το οποίο μπορεί να πάρει από μερικά δευτερόλεπτα, μέχρι μερικά λεπτά, ανάλογα με τη συσκευή, τις συνθήκες υπό τις οποίες λειτουργεί και τη θέση που βρίσκεται.



Ο αισθητήρας Location Sensor ανιχνεύει τρεις τιμές σχετικά με τη θέση:

- Latitude: απόσταση από τον Ισημερινό (τιμή μηδέν), βόρεια (θετικές τιμές) ή νότια (αρνητικές τιμές) στο εύρος -90 ως 90.
- Longitude: απόσταση από τον Πρώτο Μεσημβρινό ή Μεσημβρινό του Γκρίνουιτς (τιμή μηδέν), ανατολικά (θετικές τιμές) ή δυτικά (αρνητικές τιμές) στο εύρος -180 ως 180.
- Altitude: το ύψος από το επίπεδο της θάλασσας (εφόσον το υποστηρίζει η συσκευή).

Οι παρακάτω εντολές, ενημερώνουν διαρκώς τις ετικέτες στο πάνω μέρος της οθόνης για τη διεύθυνση και τις συντεταγμένες (γεωγραφικό μήκος και πλάτος) με βάση τις τιμές που λαμβάνει ο δέκτης GPS της συσκευής μας. Με βάση τις τιμές αυτές, εντοπίζεται μέσω της υπηρεσίας Google Maps και η πλήρης διεύθυνση της θέσης μας, εφόσον διατίθεται.



Εικόνα 195 - Εμφάνιση της τρέχουσας θέσης μας



Όταν ο αισθητήρας ανίχνευσης θέσης ανιχνεύσει οποιαδήποτε αλλαγή της θέσης μας, εμφάνισε στις αντίστοιχες ετικέτες κειμένου την πλήρη διεύ-



θυνση στην οποία βρισκόμαστε, τις ακριβείς συντεταγμένες (μήκος και πλάτος) και ενεργοποίησε το κουμπί για την αποθήκευση των παραπάνω στοιχείων.

Βήμα 3 Αποθήκευσητης τρέχουσας θέσης στη βάση δεδομένων

Όταν ο χρήστης αγγίζει το κουμπί «Να θυμάσαι αυτή τη θέση», η εφαρμογή πρέπει να κάνει τα παρακάτω:

Α. να ενημερώνει τις τιμές των ετικετών στο κάτω μέρος της οθόνης και

Β. να αποθηκεύει τα στοιχεία αυτά στη βάση δεδομένων της εφαρμογής, ώστε να είναι διαθέσιμα ακόμα και αν η εφαρμογή τερματιστεί και ξεκινήσει ξανά.

Θα αποθηκεύσουμε τις τιμές της πλήρης διεύθυνσης καθώς και του γεωγραφικού μήκους και πλάτους στη βάση δεδομένων, χρησιμοποιώντας τα αναγνωριστικά "Address", "Lat" και «Long" αντίστοιχα.

Επίσης θα εμφανίζουμε ένα παράθυρο διαλόγου στον χρήστη με τη βοήθεια του αντικειμένου Notifier, ώστε να συμπληρώνει αν θέλει κάποια επιπλέον παρατήρηση ή σημείωση.

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα :



Εικόνα 196 - Οι εντολές που εκτελούνται με το πάτημα του κουμπιού αποθήκευσης



Όταν πατηθεί το κουμπί, ενημέρωσε τις αντίστοιχες ετικέτες, ενεργοποίησε το κουμπί λήψης οδηγιών και αποθήκευσε τις αντίστοιχες τιμές στη βάση δεδομένων (BΔ) σε πεδία με όνομα Address, Lat και Long.

Επιπλέον, εμφάνισε ένα μήνυμα που να προτρέπει προαιρετικά στο χρήστη να εισάγει μια επιπρόσθετη σημείωση, λ.χ. σχετικά με τη θέση αυτή.



Βήμα 4 Ανάκτησητων αποθηκευμένων στοιχείων από τη ΒΔ

Όταν ανοίγουμε την εφαρμογή, θέλουμε να εμφανίζονται στις ετικέτες στο κάτω μέρος της οθόνης, τα στοιχεία της θέσης που αποθηκεύσαμε. Κάτι τέτοιο δεν γίνεται από μόνο του, εφόσον έχουμε κλείσει και ξανανοίξει την εφαρμογή, οπότε είναι απαραίτητο να τα ανακτήσουμε από τη ΒΔ.

Το γεγονός που ταιριάζει στην περίπτωσή μας είναι το γεγονός εκκίνησης της εφαρμογής, δηλαδή της αρχικοποίησης της οθόνης (when Screen1.Initialize) από την ομάδα Control.

Ενημερώνουμε τις αντίστοιχες ετικέτες στο κάτω μέρος της οθόνης, ενεργοποιώντας το κουμπί λήψης οδηγιών, εφόσον βρεθεί αποθηκευμένη διεύθυνση στη ΒΔ.





Q

Όταν ξεκινάει η εφαρμογή, ενημέρωσε τις αντίστοιχες ετικέτες κειμένου με τις αποθηκευμένες τιμές των πεδίων (Address, Lat, Long και Note). Αν δεν βρεθεί αποθηκευμένη κάποια τιμή, πρόσθεσε στις ετικέτες τις τιμές «-», «0», «0» και «-» αντίστοιχα.

Εάν, μετά την ενημέρωση, το μήκος του κειμένου της ετικέτας RememberAddressDataLabel είναι μεγαλύτερο από το μηδέν, δηλαδή αν έχει αποθηκευτεί κάποια τιμή, ενεργοποίησε το κουμπί λήψης οδηγιών.

Βήμα **5** Λήψη οδηγιώνγια την πλοήγηση μέσω Google Maps

Με το πάτημα του κουμπιού «Οδηγίες διαδρομής» επιθυμούμε να λάβουμε τις απαραίτητες οδηγίες για τη μετάβασή μας στο σημείο που αποθηκεύσαμε και στο οποίο βρίσκεται το ποδήλατό μας. Αυτό, ευτυχώς για εμάς, δεν απαιτείται να υλοποιηθεί από το μηδέν, καθώς μπορούμε να αξιοποιήσουμε μια πολύ χρήσιμη δυνατότητα που μας παρέχει το App Inventor με την ονομασία Activity Starter (Εκκινητής Δραστηριοτήτων).



Ο ΕΚΚΙΝΗΤΗΣ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ (ACTIVITY STARTER)



Στην προκειμένη περίπτωση μας ενδιαφέρει να ανοίξουμε την εφαρμογή Google Maps με τις κατάλληλες πληροφορίες (τρέχουσα διεύθυνση και διεύθυνση προορισμού) ώστε να μας βοηθήσει στην πλοήγησή μας στο σημείο ενδιαφέροντος.

Designer Γραφικό Περιβάλλον Εφαρμογής

Προκειμένου να επιτύχουμε τα παραπάνω, προσθέτουμε αρχικά στην οθόνη μας, ένα μηορατό αντικείμενο Activity Starter, σύμφωνα με τον ακόλουθο πίνακα:

\$	από την ομάδα	μεταφέρουμε το αντικείμενο	μεταβάλλουμε τις ιδιότητες
	Connectivity	ActivityStarter	Action: android.intent.action.VIEW
			ActivityClass: com.google.android.maps.MapsActivity
			ActivityPackage: com.google.android.apps.maps

Δεν χρειάζεται να ορίσουμε κάποια άλλη ιδιότητα τώρα, καθώς οι υπόλοιπες απαιτούμενες τιμές δεν είναι σταθερές.



Blocks

Επιστρέφουμε στο χώρο συγγραφής του προγράμματος για να ορίσουμε τα γεγονότα που θα συμβούν για τη λήψη των οδηγιών.





Όταν πατηθεί το κουμπί ShowDirectionsButton, δώσε στην ιδιότητα DataUri του ActivityStarter1 την τιμή που προκύπτει από την ένωση των αναφερόμενων σταθερών τμημάτων κειμένου και ετικετών.
 (Ουσιαστικά, προκύπτει μια διεύθυνση URL, με τα απαραίτητα στοιχεία για την εκκίνηση της πλοήγησης στην υπηρεσία Google Maps).
 Ξεκίνα τη δραστηριότητα του ActivityStarter1 (με τις δοσμένες παραμέτρους).
 Η εφαρμογή μας είναι έτοιμη!
 Ώρα να την ελέγξουμε με την κινητή μας συσκευή (ιδανικά σε εξωτερικό χώρο).

Ολοκληρωμένος, ο κώδικας της εφαρμογής μας (μέχρι στιγμής) έχει ως εξής:



call TinyDB1 .StoreValue	
tag (" (Address) "
valueToStore	CurrentAddressDataLabel
call TinyDB1 .StoreValue	
tag (" (Lat) "
valueToStore	GPSCurrentLatLabel
call TinyDB1 .StoreValue	
tag (" (Long) "
valueToStore	GPSCurrentLongLabel
call (Notifier1). ShowTextDial	og
messa	ge 🖵 " ((προαιρετική συμπλήρωση)) "
ti	tle (ἕ Σημείωση) *
cancelat	ble 👔 true 📰

whe	n Screen1 · Initialize
do	set RememberAddressDataLabel . Text . to / call TinyDB1 . GetValue
	tag (# Address *
	valuelfTagNotThere (/ " 😑 "
	set GPSRememberLatLabel • . Text • to / call TinyDB1 • .GetValue
	tag (<mark>, " Lat</mark> "
	valuelfTagNotThere
	set GPSRememberLongLabel • . Text • to (call TinyDB1 • . GetValue
	tag (🖕 " Long "
	valuelfTagNotThere (" 🔘 "
	set RememberNoteLabel . Text to (call TinyDB1 . GetValue
	tag (Note)
	valuelfTagNotThere 🔰 " 🖃 "
	e if I (length (RememberAddressDataLabel ▼). Text ▼ > ▼ (0
	then set ShowDirectionsButton . Enabled to true .
1.1	





επεκτάσεις

Στο Βήμα 3 (βλ. εικόνα 112) προσθέσαμε τις απαραίτητες εντολές για την αποθήκευση κάποιων στοιχείων στη ΒΔ. Προσθέσαμε μάλιστα ένα παράθυρο διαλόγου (μέσω του Notifier) για την εμφάνιση ενός μηνύματος που προτρέπει (προαιρετικά) το χρήστη να εισάγει μια επιπρόσθετη σημείωση, λ.χ. κάποια λεπτομέρεια σχετικά με τη θέση που άφησε το ποδήλατό του.

Παρ' όλα αυτά η σημείωση αυτή, προς το παρόν, δεν αποθηκεύεται. Κάντε τις απαραίτητες προσθήκες εντολών, ώστε μετά την εισαγωγή (ή όχι) της σημείωσης, αυτή να αποθηκεύεται στη ΒΔ, όπως αποθηκεύονται ήδη η διεύθυνση και οι συντεταγμένες.

when Notifier1 .AfterTextInput
response
do

Για την αποθήκευση της προαιρετικής σημείωσης, θα χρειαστείτε το πλακίδιο when Notifier1.AfterTextInput που πυροδοτείται με το που ο χρήστης ολοκληρώσει την εισαγωγή (ή όχι) του κειμένου στο αναδυόμενο παράθυρο.

Εφόσον ο χρήστης συμπληρώσει κάποιο κείμενο, αυτό μεταφέρεται μέσω της παραμέτρου response. Εάν όμως ο χρήστης πατήσει το κουμπί Cancel, το γεγονός επιστρέφει την τιμή "Cancel" στην παράμετρο response, οπότε καλό είναι να μην αποθηκεύσουμε τη συγκεκριμένη τιμή στη ΒΔ.

Υλοποιήστε το τμήμα εντολών που ελέγχει την τιμή της παραμέτρου response και ενημερώνει την αντίστοιχη ετικέτα (RememberNoteLabel) στο κάτω μέρος της οθόνης και αποθηκεύει τη σημείωση στη ΒΔ σε ένα πεδίο με όνομα Note.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΜΠΑΛΕΣ ΠΟΥ ΑΝΑΒΟΣΒΗΝΟΥΝ (ΠΑΙΧΝΙΔΙ ΜΝΗΜΗΣ)

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

Θα υλοποιήσουμε ένα παιχνίδι μνήμης. Στην οθόνη θα εμφανίζονται 4 μπάλες διαφορετικού χρώματος. Όταν το παιχνίδι ξεκινάει οι μπάλες θα αναβοσβήνουν με τυχαία σειρά. Στη συνέχεια ο παίκτης θα πρέπει να αγγίξει τις μπάλες με την ίδια σειρά που αναβόσβησαν. Κάθε φορά που ο παίκτης τα καταφέρνει θα ανεβαίνει ένα επίπεδο, δηλαδή θα αυξάνονται οι φορές που αναβοσβήνουν οι μπάλες, κάνοντας το παιχνίδι σταδιακά όλο και πιο δύσκολο.



Εικόνα 198 - Παιχνίδι μνήμης με μπάλες

βασικές εννοιές

- Λίστες
- Διαδικασίες
- Δομή επανάληψης
- Δομή επιλογής
- Λογικές μεταβλητές



Βήμα1 Εισαγωγή έτοιμου έργου

Σας δίνεται έτοιμο ένα μέρος της εφαρμογής (αρχείο BallsFlashing.aia από το φάκελο BallsFlashing της διεύθυνσης <u>http://tiny.cc/aifiles</u>), που περιλαμβάνει το γραφικό της περιβάλλον, μια σειρά από μεταβλητές, τις εντολές που εκτελούνται κατά την εκκίνηση της εφαρμογής και μια έτοιμη διαδικασία που κάνει ορατές όλες τις μπάλες. Όλα αυτά θα μας χρειαστούν παρακάτω.

Κάντε εισαγωγή του έργου από το περιβάλλον του App Inventor, μενού Project → Import project (.aia) from my computer.

Συγγραφή Προγράμματος

Blocks

Βήμα 2 Παραγωγήτυχαίας σειράς για το animation

Το πρώτο κομμάτι της εφαρμογής μας αφορά την υλοποίηση του animation, που θα κάνει τις μπάλες να αναβοσβήνουν με τυχαία σειρά. Το animation θα ξεκινά όταν ο χρήστης πατήσει το κουμπί «Ξεκίνα».

Παρατηρήστε ότι στο έτοιμο κομμάτι της εφαρμογής μας υπάρχει μια λίστα με όνομα **ran**domBalls.

Για να κάνουμε τις μπάλες να αναβοσβήνουν με τυχαία σειρά, θα γεμίζουμε την παραπάνω λίστα με τυχαίους αριθμούς από το 1 μέχρι το 4 (καθένας αντιστοιχεί σε μια από τις 4 μπάλες) και στη συνέχεια θα αναβοσβήνουμε τις μπάλες με βάση τους αριθμούς που υπάρχουν στη λίστα.

Η πρώτη μας δουλειά θα είναι να «καθαρίζουμε» τη λίστα κάθε φορά που ο παίκτης αγγίζει το κουμπί «Ξεκίνα».



Εικόνα 199 - Καθάρισμα της λίστας με τους τυχαίους αριθμούς

Στη συνέχεια θα δημιουργούμε ένα πλήθος τυχαίων αριθμών και θα το αποθηκεύουμε στη λίστα randomBalls. Οι τυχαίοι αριθμοί θα είναι τόσοι, όσο και το επίπεδο του παίκτη που απεικονίζεται στη μεταβλητή level. Η παραπάνω λειτουργία θα γίνεται από διαδικασία που θα κατασκευάσουμε για το σκοπό αυτό.





Εικόνα 200 - Η διαδικασία παραγωγής τυχαίων αριθμών



Πρόσθεσε στη λίστα randomBalls έναν τυχαίο αριθμό από το 1 μέχρι το 4. Επανάλαβε τόσες φορές, όσες η τιμή της μεταβλητής level.

Στη συνέχεια θα προσθέσουμε κάποιες εντολές που θα απενεργοποιούν προσωρινά τα κουμπιά της εφαρμογής, ώστε να μην μπορούν να πατηθούν κατά τη διάρκεια του animation, θα ενεργοποιούμε το χρονόμετρο – ρολόι που είναι υπεύθυνο για το animation και θα θέτουμε την τιμή της μεταβλητής randomBallIndex στο 1.

Η μεταβλητή randomBallIndex είναι η τρέχουσα θέση της λίστα randomBalls και φυσικά ξεκινάει από το 1, δηλαδή την πρώτη τιμή της λίστας. Κατά τη διάρκεια του animation, η τιμή της μεταβλητής θα αυξάνεται, ώστε να μετακινούμαστε διαδοχικά σε όλες τις θέσεις της λίστας randomBalls.

Ο ολοκληρωμένος κώδικας που αντιστοιχεί στο άγγιγμα του κουμπιού «Ξεκίνα» φαίνεται στην παρακάτω εικόνα.



Εικόνα 201 - Όταν πατηθεί το κουμπί "Ξεκίνα"



Βήμα**3** Αναβοσβήνοντας τις μπάλες

Στη συνέχεια θα περιγράψουμε τις ενέργειες που θα γίνονται κάθε φορά που πυροδοτείται το χρονόμετρο AnimationClock.

To AnimationClock θα κάνει τα εξής: Αρχικά θα εμφανίζει όλες τις μπάλες. Έπειτα κάθε **δεύτερη** φορά που πυροδοτείται, θα «κρύβει» την μπάλα που πρέπει, με βάση τις τιμές της λίστας randomBalls, Αυτό επιτυγχάνεται με την χρήση μιας λογικής μεταβλητής με όνομα showBalls, η οποία παίρνει διαδοχικά τις τιμές true – false. Όταν η τιμή της είναι true, οι μπάλες παραμένουν όλες ορατές, ενώ όταν η τιμή της είναι false κρύβεται η αντίστοιχη μπάλα. Επίσης, κάθε φορά που πυροδοτείται το AnimationClock η τιμή της λογικής μεταβλητής αντιστρέφεται με τη βοήθεια του λογικού τελεστή not.



Εικόνα 202 - Ενεργοποίηση του AnimationClock

Μέσα στην if, θα εξετάζουμε αν η μεταβλητή randomIndex έχει ξεπεράσει το μέγεθος της λίστας, οπότε πρέπει να σταματήσουμε το animation...



Εικόνα 203 - Έλεγχος για την ολοκλήρωση των στοιχείων της λίστας

... ή σε διαφορετική περίπτωση να κρύψουμε την κατάλληλη μπάλα.

Για να κρύψουμε τώρα κάποια από τις μπάλες, θα «συμβουλευτούμε» τη λίστα με τους τυχαίους αριθμούς randomBalls και θα κρύψουμε την μπάλα που έχει τον ίδιο αριθμό με το στοιχείο της λίστας που βρίσκεται στη θέση randomIndex.



Εικόνα 204 - Επιλογή στοιχείου της λίστας



Το στοιχείο που παίρνουμε από τη λίστα θα έχει μια από τις τιμές 1 εως 4. Θα εξετάζουμε ποια από όλες τις τιμές έχει, ώστε να κρύβουμε την αντίστοιχη μπάλα, όπως φαίνεται παρακάτω :

if f select list item list get global randomBalls = 1 (1) index get global randomBallIndex =				
then set Ball1 . Visible to false				
else if select list item list get global randomBalls = 2 index get global randomBallIndex =				
then set Ball2 . Visible to false				
else if C select list item list get global randomBalls = 3 (3) index get global randomBallIndex =				
then set Ball3 . Visible to false				
else set Ball4 . Visible to false .				



Τέλος ο μετρητής randomBallIndex θα αυξάνεται κατά 1.

set global randomBallIndex • to (get global randomBallIndex •	+ (1

Εικόνα	206 -	Αύξηση του	μετρητή
--------	-------	------------	---------

Σε περίπτωση που ο μετρητής randomBallIIndex ξεπεράσει το μέγεθος της λίστας τότε το animation πρέπει να τερματιστεί, επομένως θα απενεργοποιήσουμε το AnimationClock και θα ενεργοποιήσουμε τα κουμπιά, ώστε ο παίκτης να μπορεί να συνεχίσει το παιχνίδι. Οι εντολές που αφορούν την ενεργοποίηση του AnimationClock φαίνονται παρακάτω.





Εικόνα 207 - Ολοκλήρωση των ενεργειών κατά την ενεργοποίηση του χρονομέτρου



Ώρα να ελέγξουμε την εφαρμογή μας με την κινητή μας συσκευή.

ήμα4 Ξεκ

Ξεκινώντας τοπαιχνίδι

Όταν ο παίκτης αγγίξει το κουμπί InteractButton θα πρέπει να αγγίζει τις μπάλες με την ίδια σειρά με την οποία κρύφτηκαν.

Το πρώτο που θα κάνουμε είναι να απενεργοποιήσουμε τα δύο κουμπιά και να θέσουμε τις τιμές δύο μεταβλητών της εφαρμογής.

Η πρώτη μεταβλητή με όνομα playerMoves θα μετράει πόσες μπάλες έχει αγγίξει ο παίκτης και θα παίρνει αρχική τιμή 1.



Η δεύτερη μεταβλητή με όνομα isInteracting παίρνει τις τιμές true/false και την χρησιμοποιούμε για να μας δείχνει αν ο παίκτης βρίσκεται στη φάση της επιλογής (και όχι του animation), ώστε να του επιτρέψουμε να αγγίζει τις μπάλες. Οι εντολές που θα μας χρειαστούν φαίνονται παρακάτω...



Εικόνα 208 - Το άγγιγμα του κουμπιού έναρξης του παιχνιδιού

Βήμα**5** Ηεπιλογή της μπάλας

Όταν ο παίκτης αγγίξει μια μπάλα θα γίνονται οι παρακάτω ενέργειες : εφόσον η λογική μεταβλητή isInteracting είναι αληθής (true) η μπάλα θα κρύβεται για μισό δευτερόλεπτο με τη βοήθεια του χρονόμετρου – ρολογιού InteractingClock. Έπειτα θα γίνεται έλεγχος αν η μπάλα που άγγιξε ο παίκτης είναι στη σωστή σειρά με βάση το animation που είδε προηγούμενως.

Ο έλεγχος για το αν ο παίκτης άγγιξε τη σωστή μπάλα θα γίνεται από διαδικασία που θα υλοποιήσουμε για το σκοπό αυτό, ας την ονομάσουμε checkBall. Η διαδικασία θα παίρνει ως παράμετρο τον αριθμό της μπάλας που πατήθηκε.

Παρακάτω δίνονται οι εντολές που θα εκτελούνται όταν πατηθεί η 1^η μπάλα καθώς και οι εντολές κατά την πυροδότηση του ρολογιού InteractingClock.



Εικόνα 209 - Το άγγιγμα της πρώτης μπάλας





Εικόνα 210 - Η πυροδότηση του χρονομέτρου InteractingClock

Βήμα6 Οέλεγχος της μπάλας

Παρακάτω θα περιγράψουμε τη διαδικασία checkBalls. Η διαδικασία θα ελέγχει αν ο παίκτης διάλεξε τη σωστή μπάλα ή αν έκανε λάθος.

Σε περίπτωση λανθασμένης επιλογής θα αναπαράγει τον ήχο failure.mp3 και θα γίνονται οι απαραίτητες ενέργειες ώστε το παιχνίδι να αρχίζει από την αρχή.





Σε περίπτωση σωστής επιλογής, η διαδικασία θα ελέγχει αν ο παίκτης ολοκλήρωσε σωστά όλες τις κινήσεις, δηλαδή αν ο αριθμός των σωστών επιλογών του είναι όσα και τα στοιχεία της λίστας με τους τυχαίους αριθμούς. Σε αυτή την περίπτωση θα αναπαράγει τον ήχο success.wav, θα γίνονται οι απαραίτητες ενέργειες ώστε το παιχνίδι να αρχίζει από την αρχή και ο παίκτης θα ανεβαίνει επίπεδο. Διαφορετικά θα αυξάνει τον αριθμό των σωστών επιλογών του παίκτη (μεταβλητή playerMoves).





Εικόνα 212 - Ενέργειες για τη σωστή επιλογή



Η εφαρμογή μας είναι έτοιμη! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

Παρακάτω δίνεται ο κώδικας της ολοκληρωμένης εφαρμογής.

initialize global playerMoves to [1]	
initialize global (showBalls) to 🔓 false 🔪	
initialize global (isInteracting) to (false v)	
initialize global (level) to 📫 🚺	
initialize global randomBalls to L Create empty list	
initialize global (randomBallIndex) to	
when Screen1 . Initialize	_
do set AnimationClock . TimerEnabled to false	
set (InteractButton v). Enabled v to (false v)	
set (StartAnimationButton . Enabled . to (true	•
set InteractingClock • . TimerEnabled • to 🕻 fals	
when StartAnimationButton . Click	
do 🛛 set global randomBalls 🔹 to 其 🔲 create empty list	
call (prepareRandomBalls -)	
set StartAnimationButton • . Enabled • to (false	•
set InteractButton 🔹 . Enabled 🔻 to 🔰 false 💌	
set (AnimationClock v). (TimerEnabled v) to (true	•
set global randomBallindex v to (1)	











wrie	Dall	
x	У	
do	🚺 if	🔓 🖟 get (global isInteracting 🔻 (true 🗸
	then	set Ball2 Visible - to false -
		set InteractingClock 🔹 . TimerEnabled 🔹 to 📫 true 🔹
		call checkBall
		ballPressed (2





when Ball3 .Touched
xy
do 📋 if 📫 get global isInteracting 🔻 💷 🕻 true 🗸
then set Ball3 . Visible to (false .
set InteractingClock • . TimerEnabled • to I true •
call checkBall
ballPressed (3)
when Ball4 v .Touched
x y
do 🕛 if 🗘 🕻 get global isInteracting 🔻 🖃 🕻 true 🔪
then set Ball4 . Visible . to (false .
set InteractingClock . TimerEnabled . to true .
call CheckBall
ballPressed (4
when InteractingClock . Timer
do call showAllBalls
set InteractingClock • . TimerEnabled • to I false •
to prepareRandomBalls
do for each number from (
by (1)
do D add items to list list and alobal randomBalls
item
random integer from (1) to (4)

to showAllBalls					
do	set	Ball1 . Visible to	true 🔻		
	set	(Ball2 • . Visible • to	(true 🔻		
	set	(Ball3 🔹 . Visible 🔹 to 🏮	true 🔻		
	set	Ball4 . Visible to	true 🔹		





επεκτάσεις

1. Ένας τρόπος να βοηθήσουμε τον παίκτη να θυμάται τη σειρά με την οποία αναβόσβησαν οι μπάλες είναι να συνδυάσουμε κάθε μια από αυτές με ένα διαφορετικό ήχο (για παράδειγμα μια μουσική νότα) που θα αναπαράγεται κατά το «σβήσιμο» της μπάλας.

Κάντε τις απαραίτητες ενέργειες για να υλοποιηθεί η παραπάνω λειτουργία. Μπορείτε να βρείτε τους ήχους για τις νότες στο φάκελο BallsFlashing της διεύθυνσης <u>http://tiny.cc/aifiles.</u>



Χρησιμοποιήστε μια λίστα για να αποθηκεύσετε τα ονόματα των αρχείων με τις νότες.

2. Κάντε τις απαραίτητες τροποποιήσεις στην εφαρμογή, ώστε ο παίκτης να βλέπει στην οθόνη το επίπεδο στο οποίο βρίσκεται.



ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Η εφαρμογή κουμπί Πανικού (βλ. δραστηριότητα για το σπίτι PanicButton μαθήματος 1), δεν είναι παραμετροποιήσιμη. Πιο συγκεκριμένα, ο αριθμός στον οποίο αποστέλλεται το μήνυμα SMS, καθώς και το κείμενο του μηνύματος είναι σταθερά και ενσωματωμένα στο τμήμα σχεδίασης της εφαρμογής (Designer), οπότε μπορούν να τροποποιηθούν μόνο από κάποιον προγραμματιστή και όχι από οποιονδήποτε χρήστη της εφαρμογής.

Σας δίνεται έτοιμο ένα μεγάλο μέρος της εφαρμογής (αρχείο PanicButton2.aia από το φάκελο PanicButton2 της διεύθυνσης <u>http://tiny.cc/aifiles</u>) ώστε να εστιάσετε στις παρακάτω νέες λειτουργίες:

 Να μπορεί ο χρήστης να επιλέξει από τις επαφές της συσκευής τον αριθμό, όπου θα σταλεί το SMS.

Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο παρακάτω γεγονός:



Να μπορεί ο χρήστης να καθορίζει το κείμενο του μηνύματος που θα αποσταλεί.
 Η ερώτηση για εισαγωγή του νέου κειμένου προς αποστολή είναι έτοιμη. Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο δεύτερο από τα παρακάτω γεγονότα:



 Ο αριθμός και το κείμενο να αποθηκεύονται σε τοπική Βάση Δεδομένων, ώστε να παραμένουν αποθηκευμένα, ακόμα κι αν ο χρήστης κλείσει την εφαρμογή και μέχρι να τα τροποποιήσει ξανά.

Θα χρειαστείτε εντολές τύπου StoreValue όπως την παρακάτω (αλλά και εντολές που ενημερώνουν τις αντίστοιχες τιμές των μεταβλητών smsNumber και message):



 Στο κείμενο του μηνύματος να προστίθεται αυτόματα στο τέλος η τρέχουσα διεύθυνση και οι συντεταγμένες στις οποίες βρίσκεται ο χρήστης, σύμφωνα με τα δεδομένα που λαμβάνει το GPS της συσκευής.

Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο παρακάτω γεγονός:





2. [Τυχερά μπισκότα]. Υλοποιήστε μια εφαρμογή που θα λειτουργεί σαν τυχερό μπισκότο. Ο χρήστης θα βλέπει αρχικά στην οθόνη του την εικόνα ενός τυχερού μπισκότου. Όταν την αγγίζει, η εφαρμογή θα παίζει έναν ήχο και μόλις σταματήσει η αναπαραγωγή του ήχου θα εμφανίζει στον χρήστη ένα τυχαίο μήνυμα.



Κατά το διάστημα της αναπαραγωγής του ήχου η εικόνα του μπισκότου δεν θα μπορεί να ξαναπατηθεί. Επίσης θα εμφανίζεται στον χρήστη το μήνυμα «Περίμενε... Η τύχη σου δουλεύει!», το οποίο και θα «αναβοσβήνει», με τυχαίο χρώμα κάθε φορά που εμφανίζεται.



Μόλις ολοκληρωθεί η αναπαραγωγή του ήχου θα εμφανίζεται ένα τυχαίο μήνυμα στον χρήστη και η εικόνα θα αλλάζει σε ανοικτό μπισκότο.







Πλέον όταν ο παίκτης πατήσει πάνω στην εικόνα του ανοικτού μπισκότου, όλα θα ξεκινούν από την αρχή.

Όλα τα αρχεία που θα χρειαστείτε βρίσκονται στη διεύθυνση <u>http://tiny.cc/aifiles</u> και το φάκελο Fortune Cookies.

Μπορείτε να βάλετε όποια και όσα μηνύματα θέλετε στα τυχερά μπισκότα. Σε περίπτωση που δεν έχετε έμπνευση χρησιμοποιήστε κάποια από τα παρακάτω!

- Η τύχη σου κρύβεται σε άλλο μπισκότο.
- Ο Κομφούκιος λέει : Ο δρόμος για τα πλούτη κρύβεται στη δουλειά για το σχολείο!
 Διάβαζε!
- Αποδέξου ότι κάποιες μέρες είσαι το περιστέρι και κάποιες άλλες το άγαλμα.
- Όταν όλοι έρχονται προς το μέρος σου τότε είσαι σε λάθος λωρίδα!
- Θα πεινάσεις πάλι σε 1 ώρα!
- Η ευτυχία κρύβεται στο τελευταίο κουδούνι, Παρασκευή μεσημέρι!
- Σε λίγο καιρός θα είσαι απένταρος...πάλι!
- Η σκληρή δουλειά θα σου ανταποδώσει στο μέλλον. Η τεμπελιά θα σου ανταποδώσει άμεσα!
- Αυτό το τυχερό μπισκότο είναι άτυχο! Δοκίμασε κάποιο άλλο!



3. [Εφαρμογή Mix and Match]. Το Mix and Match είναι ένα διασκεδαστικό παιχνίδι που περιλαμβάνει μια σειρά εικόνων «τεμαχισμένων» σε 3 μέρη, όπως φαίνεται παρακάτω :



Κατά την εκκίνηση της εφαρμογής προβάλλεται στην οθόνη μια από τις σωστές εικόνες (η οποία επιλέγεται τυχαία κάθε φορά που ξεκινάει η εφαρμογή).



Στη συνέχεια ο παίκτης έχει 3 επιλογές :

Να σύρει το δάκτυλό του σε ένα από τα 3 τμήματα (κεφάλι, μέση, πόδια), οπότε
 MONO το συγκεκριμένο τμήμα εμφανίζει την επόμενη εικόνα.





 Να αγγίξει την οθόνη, οπότε και τα 3 τμήματα μετακινούνται στην επόμενη κατά σειρά εικόνα ταυτόχρονα.



 Να κινήσει τη συσκευή του πέρα δώθε (accelerometer), οπότε και για τα 3 τμήματα επιλέγεται μια διαφορετική τυχαία εικόνα. Δηλαδή, άλλη τυχαία εικόνα για το κεφάλι, άλλη για το σώμα και άλλη για τα πόδια.



Εννοείται ότι όταν ο χρήστης φτάνει στην τελευταία εικόνα της λίστας θα επιστρέφει πάλι στην πρώτη εικόνα. Μην ξεχάσετε να κάνετε τους απαραίτητους ελέγχους!

Στη διεύθυνση <u>http://tiny.cc/aifiles</u> και στο φάκελο Mix and Match θα βρείτε το έργο mixAndMatch.aia το οποίο και θα εισάγετε στο περιβάλλον του App Inventor. Εκτός από το έτοιμο γραφικό περιβάλλον υπάρχουν και 3 έτοιμες λίστες. Μια που περιλαμβάνει τις εικόνες για το κεφάλι, μια που περιλαμβάνει τις εικόνες για το σώμα και μια που περιλαμβάνει τις εικόνες για τα πόδια.

Χρησιμοποιήστε κατάλληλα τις λίστες αυτές για να ολοκληρώσετε την εφαρμογή.

4. Περιγράψτε, σχεδιάστε και υλοποιήστε μια εφαρμογή της αρεσκείας σας, που θα κάνει χρήση του Αισθητήρα Τοποθεσίας (Location Sensor).

5. Η κρυπτογράφηση είναι ένας τρόπος «αλλοίωσης» ενός μηνύματος, ώστε να είναι κατανοητό μόνο από τον αποστολέα και τον παραλήπτη και όχι από άλλους.



Ένας απλός τρόπος κρυπτογράφησης είναι να αντικαταστήσουμε κάθε γράμμα μιας πρότασης με κάποιο άλλο γράμμα που έχουμε συμφωνήσει από πριν με τον παραλήπτη.

Για παράδειγμα, η πρόταση «PANTEBOY ΣΤΗΝ ΠΛΑΤΕΙΑ ΤΟ ΑΠΟΓΕΥΜΑ» θα μπορούσε να μετατραπεί σε «ΥΔΠΧΘΣΨ ΦΧΚΠ ΤΞΑΧΘΜΔ ΧΣ ΔΤΣΖΘΨΟΔ», μετακινώντας κάθε γράμμα της κατά 3 θέσεις στο αλφάβητο. Δηλαδή το γράμμα «P» αντικαθίσταται από το γράμμα «Φ», το γράμμα «A» αντικαθίσταται από το γράμμα «Δ» κοκ.

Ο πρώτος που χρησιμοποίησε την παραπάνω μέθοδο κρυπτογράφησης ήταν ο Ιούλιος Καίσαρας για το λόγο αυτό ονομάζεται αλγόριθμος του Καίσαρα.



Θα υλοποιήσουμε μια εφαρμογή ή οποία θα δίνει στον χρήστης της τη δυνατότητα να γράψει ένα μήνυμα, να το κρυπτογραφήσει και στη συνέχεια να το αποστείλει με SMS. Επίσης θα του δίνει τη δυνατότητα να γράψει ένα κρυπτογραφημένο μήνυμα και στη συνέχεια να ζητήσει την αποκρυπτογράφησή του.

Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου





Στη διεύθυνση <u>http://tiny.cc/aifiles</u> και στο φάκελο Encryption θα βρείτε το έργο Encryption.aia το οποίο και θα εισάγετε στο περιβάλλον του App Inventor. Εκτός από το έτοιμο γραφικό περιβάλλον υπάρχουν και 2 έτοιμες λίστες. Μια που περιλαμβάνει τα κεφαλαία γράμματα του ελληνικού αλφαβήτου με την κανονική τους σειρά και μια που περιλαμβάνει τα κεφαλαία γράμματα του ελληνικού αλφαβήτου με την κανονική τους σειρά.

Χρησιμοποιήστε κατάλληλα τις λίστες αυτές για να ολοκληρώσετε την εφαρμογή.

Για να σπάσετε μια λέξη ή πρόταση σε μεμονωμένους χαρακτήρες θα χρεια- στείτε τις παρακάτω εντολές.				
for each number from (1) to () length () get global plainText • by (1)				
do set global charToEncrypt v to i segment text i get global plainText v start i get number v length i 1				

6. Ολοκληρώστε τη μισοτελειωμένη εφαρμογή του κλασικού παιχνιδιού της κρεμάλας. Το περιβάλλον της εφαρμογής είναι έτοιμο, όπως επίσης και ένα μεγάλο μέρος του κώδικα. Επεξεργαστείτε το έργο HangMan.aia από το φάκελο Hangman στη διεύθυνση <u>http://tiny.cc/aifiles</u> και υλοποιήστε τη λειτουργικότητα που λείπει, ουσιαστικά τη διαδικασία CheckLetter (letter), που καλείται για κάθε γράμμα που επιλέγεται από τον παίκτη:

Όταν επιλεγεί ένα γράμμα, να ελέγχεται αν υπάρχει στην κρυμμένη λέξη. Όπου υπάρχει, αντικαθίστανται οι παύλες με το γράμμα αυτό. Αν το γράμμα δεν βρεθεί, το γράμμα εμφανίζεται στην ετικέτα με τα λάθος γράμματα και η κρεμάλα σταδιακά προχωράει, αλλάζοντας



εικόνα. Το παιχνίδι τελειώνει όταν έχουν γίνει 7 λάθη ή όταν βρεθεί η λέξη. Σε κάθε περίπτωση, στο τέλος εμφανίζονται μηνύματα νίκης ή ήττας με τη λέξη που δεν βρέθηκε και αντίστοιχοι ήχοι. Όλα τα σχετικά αρχεία έχουν προστεθεί ήδη στην εφαρμογή.

Η λίστα Word περιέχει τις λέξεις που επιλέγει τυχαία ο αλγόριθμος και την οποία μπορείτε να επεξεργαστείτε. Η λίστα LettersList δημιουργείται αυτόματα και περιέχει τα γράμματα της λέξης που επιλέγεται, ενώ η λίστα HiddenList περιέχει αντίστοιχα τις παύλες που αντιστοιχούν στα γράμματα της κρυμμένης λέξης.





ενδεικτικές λύσεις των ασκήσεων

WHERE IS MY BIKE? - ΕΠΕΚΤΑΣΗ ΕΛΕΓΧΟΥ & ΑΠΟΘΗΚΕΥΣΗΣ ΣΗΜΕΙΩΣΗΣ ΣΤΗ ΒΔ



κούμπι πανικού με δύνατοτητα επιλογής & αποθηκεύσης sms & αριθμού





ΜΠΑΛΕΣ ΠΟΥ ΑΝΑΒΟΣΒΗΝΟΥΝ (ΠΑΙΧΝΙΔΙ ΜΝΗΜΗΣ) - ΕΠΕΚΤΑΣΗ

initialize global notes to 🚺	0	make a list		c.mp3 *
				d.mp3) *
				e.mp3 *
			¢.	f.mp3) *

0 if	C (select list item list get global randomBalls • = • (1)				
		index get global randomBallindex •				
then	set (Ball T . Visible T to) false T					
	set [FXPlayer 🔹 . (Source 🔹 to 🚺 select list item list 🄰 get (global notes 🔻				
		index 🔰 🔳				
	call (FXPlayer 🔹 .Start				

Εικόνα 213 - Αντίστοιχες εντολές θα προστεθούν σε κάθε μπάλα

set (LevelLabel 🔻 . (Text 🔹 to 👂	🔲 join (* (Επίπεδο :) [*]
	(🗘 get global level 🔹 - (3

Εικόνα 214 - Εντολή αλλαγής ετικέτας όταν αλλάζει η μεταβλητή level

τύχερα μπισκότα

whe	screen1 .Initialize		
do	set global quotes 🔹 to 🌔	🔲 make a list	岸 🔭 Η τύχη σου κρύβεται σε άλλο μπισκότο) *
			🖕 🔭 Ο Κομφούκιος λέει : Ο δρόμος για τα πλούτη κρύβεται στη δουλειά για το σχολείο! Διάβαζε! 🎽
			🕻 🔭 Αποδέξου ότι κάποιες μέρες είσαι το περιστέρι και κάποιες άλλες το άγαλμα
			🖕 🔭 Όταν όλοι έρχονται προς το μέρος σου τότε είσαι σε λάθος λωρίδα! 🔭
			ς · Θα πεινάσεις πάλι σε 1 ώρα!
			岸 🔭 Η ευτυχία κρύβεται στο τελευταίο κουδούνι, Παρασκευή μεσημέρι! 📑
			Σε λίγο καιρός θα είσαι απένταροςπάλι!
			🕨 🐪 Η σκληρή δουλειά θα σου ανταποδώσει στο μέλλον. Η τεμπελιά θα σου ανταποδώσει άμεσα! 🤎
			📙 🔭 Αυτό το τυχερό μπισκότο είναι άτυχο! Δοκίμασε κάποιο άλλο!



initialize global showLabel to true			
initialize global isOpen to 🌾 false 🔹			
initialize global quotes to 🕻 🔲 create empty list			
when CookielmageSprite . Touched			
do 🔲 if 🗘 get global isOpen 🔪 = 🗸 🕻 false 🗸			
then call CookiePlayer .Start			
call CookiePlayer .Vibrate			
milliseconds (500			
set global isOpen • to t true •			
set MessageLabel 🔻 . Text 🔹 to 🚺 ΠερίμενεΗ τύχη σου δουλεύει!			
set CookielmageSprite . Enabled . to false .			
set FortuneClock . TimerEnabled to true .			
else set CookielmageSprite . Picture to for fortune_cookie.png			
set global isOpen - to false -			
set MessageLabel Text - to			
set MessageLabel • . Text • to			

when	Coo	okiePlayer 🕤 Completed	
do	set	FortuneClock • . TimerEnabled • to I false •	
	set	MessageLabel 🕤 . (Visible 🗊 to 🕴 true 🔹	
	set	CookielmageSprite -). Picture - to (fortunecookieopen2.png)	
	set	MessageLabel 🔹 . (Text 💌 to 🕩 pick a random item list 📮 get global quotes 📢	
	set	CookielmageSprite Enabled - to I true -	
when FortuneClock . Timer			
do	0 i	if get global showLabel - (true -	
	then	set MessageLabel • . TextColor • to 🖡 make color 🖡 💷 make a list 🖡 random integer from 🗐 to	





MIX AND MATCH

initialize global tops to L make a list a dracula-top.png alien-top.png frankenstein-top.png witch-top.png wolfman-top.png
initialize global (middles) to () make a list () (dracula-middle.png) (alien-middle.png) (frankenstein-middle.png) (witch-middle.png) (wolfman-middle.png)
initialize global bottoms to (make a list
initialize global correctimage to (random integer from (1 to (5
initialize global currentTops to
initialize global <u>currentMiddles</u> to 40




whe	n Screen1 .Initialize
do	set global currentTops • to 🔓 get global correctimage •
	set global currentMiddles 🔹 to 🔓 get global correctimage 🕤
	set global currentBottoms • to 🔓 get global correctimage •
	call (refreshimages -
whe	n (ShakingSensor 🔹 .Shaking
do	set global currentTops • to C random integer from (1) to (5)
	set global currentMiddles v to G random integer from (1) to (5)
	set global currentBottoms to random integer from 1 to 5
	call refreshimages
whe	n Canvas1 J.Touched
x	y touchedSprite
do	set global currentTops T to 🕻 🔲 🚽 get global currentTops T + 💭
	set global currentMiddles to () get global currentMiddles t + (1)
	set global currentBottoms in to / III a set global currentBottoms in to
	get global currentBottoms + 1
	call (refreshimages *)
wner	
do	
uu	set global currentTops • to t get global currentTops • + 11
	call refreshimages •
wher	n (MiddleImageSprite 🕤 .Flung
x	y speed (heading xvel) (yvel)
do	set global currentMiddles 🔹 to 🗘 🔲 💭 get global currentMiddles 🔪 + 💭
	Can Teneshimages
whee	BottomimageSprite - Flung
where	v speed heading xvel (wel)
do	set clobal currentBottoms at a light and clobal currentBottoms at a
	call refreshImages *

κρύπτογραφία





whe	en Screen1 . Initialize	
do	call (fillLettersList •	
	call fillLettersCryptList	3
whe	on ClearButton Click	
do	oct DisinMassareText	
do	set PlainMessageText	ox ▼. Text ▼ to 👫 🔭
do	set (HorizontalArrangen	Box ▼ . Text ▼ to ♥ ■ ■ ■ nent4 ▼ . Visible ▼ to ♥ false ▼
do	set PlainMessageTextE set HorizontalArrangen set PhoneNumberPicke	Box • . Text • to • • • • • • • • • • • • • • • • •







whe	n Enc	ryptButton 🔹 .Click
do	i i	🕴 🗘 trim 🚺 PlainMessageTextBox 🕤 . Text 🕤 🗲 🕬 🗯 🔵 "
	then	set global plainText • to 🚺 trim 🚺 PlainMessageTextBox •). (Text •)
		call encryptMessage ·
		set (HorizontalArrangement4 . Visible . to true .
		set PhoneNumberPicker1 . Visible to true
1	<u> </u>	
whe	n Dec	ryptButton - Click
do	set (PhoneNumberPicker1 🔹 . Visible 🔹 to 📫 false 🔹
	if 🖸	k trim k PlainMessageTextBox • . Text • ≠ • k • ● *
	then	set global plainText To C trim C PlainMessageTextBox T. Text T
		set (HorizontalArrangement4 . Visible) to true

call decryptMessage *

ſ	whe	n PhoneNumberPicker1 . AfterPicking
	do	set SMSTexting • . Message • to CEncryptedMessageLabel • . Text •
		set SMSTexting • . PhoneNumber • to PhoneNumberPicker1 • . PhoneNumber •
		call SMSTexting .SendMessage



۲	to fi	Lett	ersList	J				
do	Õ	add	items	to li	st	list	C	get global letters 🔹
						item	C	" A "
	0	add	items	to li		list	q	get global letters *
						item	9	"В"
	0	add	items	to li	st	list	9	get global letters *
						item	9	" 🚺 "
	0	add	items	to li	st	list	9	get global letters *
						item	C	" (Δ) "
	0	add	items	to li	st	list	3	get global letters •
	$ \sim$					item	4	" B "
	0	add	items	to li		list 	3	get global letters V
							4	
	0	add	items	to li	st	list 	1	get global letters •
	\vdash							
	0	add	items	to li			1	get global letters •
	\vdash					tem	4	······································
	•	add		to II			1	get global letters
						tem	2	"U"
	0	add	items	to li	st	list 	1	get global letters •
								K
	0	add	items	to li	st	list	1	get global letters
		e d d	it a marc	4- 1i	-	liem	H	
	•	add		to II		list	1	get global letters •
			14	4 - 10	-4	liem		
		auu		IO II		list	1	get global letters
		odd	itome	to li	et	liet	H	
		auu		10 11		itom	1	
		add	itome	to li		liet	H	
		auu	items					get global letters
	0	add	items	to li	st	list 	1	get global letters *
						item	4	" O "
	0	add	items	to li	st	list	3	get global letters •
	0	add	items	to II	st		1	get global letters •
		ordet	itorea	to li	a t	lint	2	P ant alabal inffarment
	0	add		to li		list	1	get global letters
		add	itome	to li	et	liet	2	cet (global letters -
		add			51	item	3	" T "
		add	itome	to li	et	liet	1	get global letters
		add		10 11		item.	1	
	6	add	items	to li	st	list	2	get global letters
			Romo	10 H		item	1	"Φ"
	6	add	items	to li	st	list	2	get global letters
		aaa		10 N		item	1	"X"
	õ	add	items	to li	st	list	7	get global leffers
						item	1	"Ψ"
	õ	add	items	to li	st	list	2	get global leffers
						item	1	"Ω"
	6	_				a contra	1	

0	to (fi	llLett	ersCry	/ptList			
do	0	add	items	to list	list	C	get global cryptLetters 🔹
	L.				item	C	" M "
	0	add	items	to list		G	get global cryptLetters *
					item	4	" ^ "
	0	add	items	to list	list	5	get global cryptLetters *
					item	4	<u>"Н"</u>
	0	add	items	to list	list	1	get global cryptLetters
	_						
	•	add		to list	list	1	get global cryptLetters
	~	odd	itomo	to liet	liet	h	
	•	auu		to list	item	1	
	ñ	add	items	to list	list	H	det global crypti etters
	Ŭ	uuu		10 1101	item	1	"K"
	ō	add	items	to list	list	7	det (dlobal crypt) etters
					item	1	" T "
	Ø	add	items	to list	list	d	get global cryptLetters
					item		"O"
	0	add	items	to list	list	C	get global cryptLetters *
					item	C	" X "
	Ø	add	items	to list	list	C	get global cryptLetters 🔹
					item	C	" 🖪 "
	٥	add	items	to list	list	C	get global cryptLetters 🔹
	L.				item	C	" Θ "
	0	add	items	to list		C	get global cryptLetters 🔹
	L				item	0	" Ф "
	٥	add	items	to list		9	get global cryptLetters *
					item	4	"Σ"
	0	add	items	to list	list	5	get global cryptLetters *
	0	add	items	to list		٩	get global cryptLetters *
					item	C	"Ω"
	0	add	items	to list	list	5	get global cryptLetters *
					item	5	" Z "
	٢	add	items	to list	list	1	get global cryptLetters
		~	it.	40 U 4	item	2	
	0	add	items	to list	itom	1	
	6	add	iteme	to list	liet	2	det diobal countil etters
		auu		to list	item	1	
	0	add	items	to list	list	7	get global crypti etters
		cicica		to not	item	1	"Y"
	Ø	add	items	to list	list	1	get global cryptLetters
					item		"Ψ"
	0	add	items	to list	list	C	get global cryptLetters 🔹
					item	C	" 🛆 "
	0	add	items	to list	list	C	get global cryptLetters 🔹
					item	C	" A "
	0	add	items	to list	list	C	get global cryptLetters 🔹
					item	C	" 🔲 "



HANGMAN

initia	lize global Word) to	initialize globa	listakes) to (0	initialize globa	correctLetters to 1
	initialize global LettersLis	t to 📔 🗿 create empty lis	t (initialize globa	al HiddenList to 📫	Create empty list
		when Screen			
		do set Lett	erListPicker 💙 . (Er	nabled 🔹 to 🕴 fal	se
0	to PickRandomWord				
do	set global Word • to b	pick a random item list {	make a list	ΠΕΙΡΑΤΗΣ ΕΡΓΑΣΤΗΡΙΟ ΚΑΛΑΜΑΚΙ ΑΛΓΟΡΙΘΜΟΣ ΚΕΛΕΠΟΥΡΙ	

whe	n StartButton Click
do	call PickRandomWord
	set global Mistakes to 0
	set global correctLetters to 10
	set global HiddenList to I o create empty list do call CheckLetter
	set global LettersList • to 1 O create empty list
	set LetterListPicker . Enabled . to I true .
	for each number from
	to 📔 length 🚺 get (global Word 🔹
	by 🕽 🚺
	do o add items to list list (get global LettersList)
	item 🚺 segment text l get global Word 💌
	start (get number -
	length [1]
	add items to list list get global HiddenList
	item 🙀 💭 👘
	set WrongLettersLabel . Text . to I
	set (HiddenWordLabel 🔹). Text 🔹 to (get global HiddenList 🔹
	set (Hangmanimage). Picture) to (1.png)
	set MessageLabel 🔻 . Text 💌 to 🕼 " Θα καταφέρεις να βρεις τη λέξη; "
1.1	







ΠΑΡΑΡΤΗΜΑ

ΠΗΓΕΣ

Το πολυμεσικό υλικό που χρησιμοποιείται στο έγγραφο και στις αναφερόμενες εφαρμογές διατίθεται δωρεάν από τους παρακάτω ιστότοπους:

http://www.freesfx.co.uk/	http://www.freevector.com	http://www.freesound.org
http://www.clker.com	http://soundbible.com	http://soundfxcenter.com

Η εικόνα του εξωφύλλου βασίζεται σε δημιουργία του Rich Dellinger: http://richd.com

Η εικόνα στην αρχή κάθε κεφαλαίου βασίζεται σε δημιουργία του Jesse Potter: <u>http://fetuscakemix.deviantart.com</u>

Η δραστηριότητα GuardDog βασίζεται στο Φύλλο Εργασίας: *GuardDog - Προγραμματισμός Εφαρμο*γής με το AppInventor. Γιώργος Χατζηνικολάκης, Γιώργος Μπουκέας. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2013.

Η δραστηριότητα «Ζωγραφική με τα δάκτυλα» βασίζεται και αποτελεί επέκταση του «Φύλλου εργασίας: FingerPainting – Προγραμματισμός Εφαρμογής με το AppInventor. Γιώργος Μπουκέας, Γιώργος Χατζηνικολάκης. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2013».

Η δραστηριότητα MoleMash αποτελεί επέκταση της αντίστοιχης δραστηριότητας από την επίσημη ιστοσελίδα του App Inventor: <u>http://beta.appinventor.mit.edu/learn/tutorials/</u>

Η δραστηριότητα Random Drops βασίζεται στο έργο "Jackson Pollock Splatter Painting" για Scratch από το χρήστη gogreen19: <u>http://scratch.mit.edu/projects/359758/</u>

Η δραστηριότητα Where is my Bike? Βασίζεται στο tutorial της επίσημης ιστοσελίδας του App Inventor "Android, Where's My Car": <u>http://beta.appinventor.mit.edu/learn/tutorials/whereismycar/whereismycar.htm</u>l

Η δραστηριότητα Mix and Match βασίζεται σε σχετικό έργο από το Wellesley College: <u>https://sites.google.com/site/wellesleycs117fall11/</u>

Η δραστηριότητα με τις μπάλες που αναβοσβήνουν βασίζεται σε σχετικό έργο από το Trinity College: http://turing.cs.trincoll.edu/~ram/cpsc110/

Η δραστηριότητα HangMan βασίζεται σε αντίστοιχη εργασία του τμήματος πληροφορικής του Πανεπιστημίου της Alabama: <u>http://cs104.cs.ua.edu/index_files/Assignments.htm</u>.



ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΓΙΑ ΤΗ ΧΡΗΣΗ ΤΟΥ ΑΡΡ INVENTOR

Η επίσημη ιστοσελίδα του App Inventor είναι η <u>http://appinventor.mit.edu</u> στην οποία μπορείτε να βρείτε πλούσιο υλικό αναφοράς, οδηγούς χρήσης και παραδείγματα δημιουργίας εφαρμογών, κ.α., ενώ φυσικά μπορείτε να δημιουργήσετε άμεσα εφαρμογές χωρίς να εγκαταστήσετε τίποτα στον Η/Υ σας.



3ήμα1 Δημιουργίαλογαριασμού Google

To online περιβάλλον δημιουργίας προγραμμάτων, μέσω του οποίου αποθηκεύετε τα έργα που δημιουργείτε στον προσωπικό σας λογαριασμό για το App Inventor, απαιτεί να διαθέτετε λογαριασμό Google (Google Account), τον οποίο μπορείτε να αποκτήσετε δωρεάν.

Google
Ένας λογαριασμός. Όλες οι υπηρεσίες Google.
Συνδεθείτε στο Λογαριασμό σας Google
8
Διεύθυνση ηλεκτρονικού ταχυδρομε Κωδικός πρόσβασης
Σύνδεση
Μείνετε συνδεδεμένοι Χρειάζεστε βοήθεια;
Δημιουργία λογαριασμού
Ένας Λογαριασμός Google για όλες τις υπηρεσίες Google 💈 M 👍 🕩 🌺 医



Επιλέξτε το όνομα χρή	ήστη σας
	@gmail.com
Ίροτιμώ να χρησιμοποι ιου ηλεκτρονικού ταχυδ	ιήσω την τρέχουσα διεύθυνσι δρομείου
Δημιουργία κωδικού τ	τρόσβασης
Ξπιβεβαιώστε τον κωδ	δικό πρόσβασής σας
ενέθλια	
Ημέρα Μήνας	\$ Έτος
Φύλο	
Είμαι	:
(ινητό τηλέφωνο	
- +30	
Η τρέχουσα διεύθυνσή	ή σας ηλεκτρονικού
αγυορομείου	

Παραλείψτε αυτήν την επαλήθευση (ενδέχεται να απαιτείται επαλήθευση μέσω τηλεφώνου) Στη φόρμα που καλείστε να συμπληρώσετε, επιλέγετε αν θα χρησιμοποιήσετε την ήδη υπάρχουσα διεύθυνση ηλεκτρονικού ταχυδρομείου, που ενδεχομένως διαθέτετε ή αν θα δημιουργήσετε νέο λογαριασμό ηλεκτρονικού ταχυδρομείου στο Gmail (...@gmail.com).

Σημειώνεται ότι, παρόλο που δεν αναφέρεται ξεκάθαρα, δεν είναι υποχρεωτική η συμπλήρωση του κινητού τηλεφώνου στη φόρμα εγγραφής, ενώ προφανές είναι και το ότι δεν είναι επίσης υποχρεωτικό να δηλώσετε το πραγματικό σας ονοματεπώνυμο.

Ο κωδικός πρόσβασης είναι στοιχείο αυστηρά προσωπικό, πρέπει να παραμένει κρυφός και φυσικά να τον απομνημονεύσετε ώστε να μην τον ξεχάσετε, μαζί και με το όνομα χρήστη!

Βήμα2 Εγκατάστασηεφαρμογής QR Code Scanner

Σίγουρα θα έχετε ξαναδεί αυτά τα τετράγωνα barcodes. Με τη βοήθεια της έξυπνης φορητής συσκευής και της κάμεράς της, ο οποιοσδήποτε μπορεί να σαρώσει έναν τέτοιο κώδικα και να τον αποκωδικοποιήσει, ώστε πολύ εύκολα και γρήγορα, να μεταβεί στην ιστοσελίδα (URL) που αντιστοιχεί στο συγκεκριμένο QR code.



Προκειμένου να μπορείτε να σαρώνετε κι εσείς τέτοιους κωδικούς και να διευκολυνθείτε στη μετάβασή σας σε διάφορες χρήσιμες διευθύνσεις, συνίσταται η εγκατάσταση στη συσκευή σας κάποιας εφαρμογής σάρωσης QR κωδικών (QR Code Scanner), όπως για παράδειγμα **ο**ποιασδήποτε από τις δύο παρακάτω δωρεάν εφαρμογές στο Play Store:

> <u>BarCode Scanner</u> από την ZXing Team



<u>QR Code Reader</u> από την Scan, Inc.





Βήμα3

Εγκατάστασητου ΜΙΤ AI2 Companion

Η ανάπτυξη και ο ταυτόχρονος έλεγχος μιας εφαρμογής μπορεί να γίνει ακόμα κι αν δεν διαθέτετε συσκευή Android, καθώς το App Inventor παρέχει έναν εξομοιωτή Android (emulator).



Build your project on your computer your computer with the onscreen emulator

Έτσι, ο έλεγχος της εφαρμογής μπορεί να γίνεται επιτόπου στην οθόνη του υπολογιστή, στην εικονική συσκευή που θα υφίσταται σε ένα ξεχωριστό παράθυρο (βλ. <u>οδηγίες</u>). Πέρα όμως του προφανούς μειονεκτήματος ότι δεν θα μπορείτε να αξιοποιήσετε όλα τα χαρακτηριστικά μιας πραγματικής συσκευής, ο emulator είναι σχετικά αργός, ενώ απαιτείται να εγκαταστήσετε στον υπολογιστή σας έξτρα λογισμικό (<u>Al Setup In-</u> <u>staller</u>).

Για τους παραπάνω λόγους, απεναντίας, **συνιστάται** η επιτόπου σύνδεση της Android συσκευής σας με τον Η/Υ μέσω ασύρματου δικτύου Wi-Fi.

Κατ' αυτό τον τρόπο, ο έλεγχος της εφαρμογής μπορεί να γίνεται επιτόπου στη συσκευή σας, εφόσον Η/Υ και συσκευή είναι συνδεδεμένα στο ίδιο δίκτυο (βλ. έξτρα <u>οδηγίες</u>).





Build your project on your computer

Test it in real-time on your device

Απαραίτητη προϋπόθεση για τη σύνδεση μέσω Wi-Fi είναι η εγκατάσταση στην Android συσκευή της εφαρμογής **MIT AI2 Companion**:

είτε από το <u>Play Store</u>

QR code:



είτε από τη σελίδα του ΑΙ (<u>.apk</u>)

QR code:



Προτείνεται η εγκατάσταση από το Play Store λόγω δυνατότητας αυτόματων ενημερώσεων.

Έτσι, όταν θέλετε να ελέγξετε την εφαρμογή σας, επιλέγετε από το άνω μενού του App Inventor την εντολή Connect --> Al Companion ...

	Al Composion	the second of the second second		
enl •	Emulator	ove Screen		
wer	USB		Components	
Dis	Reset Connectio	on s in Viewer	Screen1	
	ewer	Emulator USB	Emulator USB Dis Reset Connection	Emulator USB Components Dis Reset Connection 5 in Viewer



... οπότε και εμφανίζεται ένα παράθυρο με έναν QR κωδικό. Ανοίγετε τώρα την εφαρμογή MIT AI2 Companion στη συσκευή σας και πατώντας το κουμπί "Scan QR Code" σαρώνετε τον κωδικό (για να μην τον πληκτρολογείτε) και η σύνδεση "live testing" ολοκληρώνεται σε ελάχιστα δευτερόλεπτα.



Μπορείτε πλέον να δοκιμάσετε την εφαρμογή σας σε πραγματικό χρόνο, ακόμα και να προσθέσετε ή να αφαιρέσετε αντικείμενα στο παράθυρο σχεδίασης, να προσθέσετε ή να τροποποιήσετε εντολές στο παράθυρο εντολών και να δείτε απευθείας τι συνέπειες έχουν οι ενέργειές σας.



πακεταρισμα εφαρμογής και διαμοιράσμος της

Με τον προαναφερόμενο τρόπο μπορούμε μεν να δοκιμάσουμε την εφαρμογή που αναπτύσσουμε εύκολα και άμεσα, αλλά αυτό δεν αρκεί, αν θέλουμε η εφαρμογή να εγκατασταθεί μόνιμα στη συσκευή μας ή να την μοιραστούμε με άλλους. Απαιτείται λοιπόν να την «πακετάρουμε», δημιουργώντας το σχετικό αρχείο εγκατάστασής της (.apk).

Βήμα1 Προσθήκη εικονιδίουκαι συνοδευτικού κειμένου

Προτού δημιουργήσουμε το .apk αρχείο, καλό είναι να ορίσουμε μια εικόνα ως βασικό εικονίδιο της εφαρμογής, το οποίο θα εμφανίζεται, μετά την εγκατάστασή της στη συσκευή μας.

Η πιο απλή λύση είναι να επιλέξουμε (εφόσον υπάρχει) μια χαρακτηριστική εικόνα από αυτές που χρησιμοποιούνται ήδη στην εφαρμογή μας και να την θέσουμε ως Icon της Screen1 από το περιβάλλον σχεδίασης Designer.

Επιπρόσθετα, μπορούμε (πάλι προαιρετικά) να προσθέσουμε ένα μικρό κείμενο στην ιδιότητα AboutScreen της Screen1, το οποίο θα περιγράφει εν συντομία την εφαρμογή ή θα αναφέρει λίγα λόγια για το δημιουργό της.

Properties	
Screen1	
AboutScreen	
Icon	
None	

Βήμα 2 Πακετάρισματης εφαρμογής για εγκατάσταση

Προκειμένου να δημιουργήσουμε το .apk αρχείο, μεταβαίνουμε στο μενού Build της βασικής γραμμής εργαλείων, όπως φαίνεται και στην παρακάτω εικόνα, όπου έχουμε δυο επιλογές:



α) να εμφανιστεί ένας QR κωδικός, ώστε σαρώνοντάς τον να κατεβάσουμε το αρχείο εγκατάστασης απευθείας στη συσκευή μας (ο QR κωδικός λειτουργεί μόνο για το δημιουργό της εφαρμογής),

β) να αποθηκεύσουμε το αρχείο εγκατάστασης στον υπολογιστή μας για μετέπειτα διαμοιρασμό και εγκατάσταση με όποιον τρόπο προτιμάμε.

Αν επιθυμούμε να διαθέσουμε την εφαρμογή στο Google Play, απαιτείται η δημιουργία Publisher Account για τη μεταφόρτωση του .apk αρχείου. Για περισσότερες πληροφορίες: <u>http://appinventor.mit.edu/explore/ai2/google-play.htm</u>l

Τέλος, αν θέλουμε να μοιραστούμε τον **κώδικα** της εφαρμογής μας (μαζί με το γραφικό περιβάλλον και τα συνοδευτικά αρχεία), μπορούμε να εξάγουμε το πηγαίο αρχείο (.aia) και να το διαθέσουμε σε άλλους, ώστε να το εισάγουν ως έργο (project) στο δικό τους λογαριασμό στο App Inventor. Αρκεί να μεταβούμε στη λίστα με τα έργα μας (*My Projects*), έπειτα να επιλέξουμε το έργο (ή τα έργα) που θέλουμε να εξάγουμε και στη συνέχεια, από το μενού *Project*, να επιλέξουμε την εντολή *Export selected project (.aia) to my computer*. υ