App Inventor – 2° μάθημα (Σκύλος φύλακας)

- ✓ Καμβάς
- Φιγούρες
- Κίνηση
- ✓ Δόνηση
- ✓ SMS

Στόχος της 2ης εφαρμογής μας είναι να φτιάξουμε έναν «φύλακα», χρησιμοποιώντας έναν σκύλο που δεν θα επιτρέπει σε κανέναν να πειράξει τη συσκευή μας. Συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν κάποιος αγγίζει την οθόνη, ενώ επιπλέον τρέχει προς το σημείο που την αγγίζουμε.





<u>Βήμα 1: Δημιουργία νέου έργου, προσθήκη αρχείων</u>

 Ξεκινώντας, στην διεύθυνση <u>http://ai2.appinventor.mit.edu/</u>, αφού έχουμε συνδεθεί, δημιουργούμε ένα καινούργιο Project το οποίο ονομάζουμε *GuardDog*.



- 2. 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, θα τροποποιήσουμε κάποιες ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο Properties. Έχοντας επιλεγμένο το αντικείμενο Screen1 στο μενού Components θέτουμε τις ιδιότητες της κύλισης, Scrollable, σε no, ξετικάροντας το αντίστοιχο πεδίο, αν δεν είναι ξετικαρισμένο. Επιπλέον θέτουμε τον τίτλο, Title, σε Guard Dog.
- Στη συνέχεια, θα προσθέσουμε στο έργο μας όλα τα απαραίτητα αρχεία που θα βρείτε και στον φάκελό σας. Αυτά είναι τα:
 - Grass.jpg: Το γρασίδι στο οποίο κινείται ο σκύλος μας.
 - *SleepyDog.png*: Ο ήρεμος σκύλος.
 - *AngyDog.png*: Ο θυμωμένος σκύλος.
 - *Bark.mp3*: Ο σκύλος που γαβγίζει.

Media		
	Upload File	

Από το πλαίσιο Media, κάτω δεξιά, επιλέγουμε την εντολή Upload File... και βρίσκουμε τα παραπάνω αρχεία.

<u>Βήμα 2: Προσθήκη του καμβά</u>

Ο καμβάς, Canvas, είναι μία ορθογώνια επιφάνεια, εντός της οποίας μπορούμε να χειριζόμαστε φιγούρες, Sprites, ή να σχεδιάζουμε αγγίζοντας την.

- 4. Στο βήμα αυτό θα προσθέσουμε τον καμβά που θα αποτελέσει το χώρο όπου θα κινείται ο σκύλος και θα καθορίσουμε τις ιδιότητες του χώρου αυτού. Βρίσκουμε στην παλέτα αριστερά, Palette, στην ομάδα πλακιδίων Drawing and Animation, το αντικείμενο Canvas και το σέρνουμε στην οθόνη.
- 5. Μετονομάζουμε το νέο μας αντικείμενο από *Canvas1* σε *DogCanvas*, από το κουμπί *Rename*. Έχοντας το επιλεγμένο, στο πλαίσιο **Components**, μεταβάλουμε τις επόμενες ιδιότητές του· από το πλαίσιο **Properties**:
 - Width: Fill parent
 - 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.

Βήμα 3: Προσθήκη φιγούρας

- 6. Οι φιγούρες, ImageSprites, τοποθετούνται αναγκαστικά στον καμβά και όχι οπουδήποτε σε μία οθόνη και αλληλοεπιδρούν με τον καμβά, με άλλες φιγούρες, αλλά και αντιδρούν στις ενέργειες του χρήστη. Για να προσθέσουμε τη φιγούρα μας που είναι ο σκύλος, από την Palette και την ομάδα Drawing and Animation μεταφέρουμε το πλακίδιο ImageSprite μέσα στον καμβά.
- **7.** Δίνουμε στο νέο μας αντικείμενο το όνομα *DogSprite* και μεταβάλλουμε τις ιδιότητες:
 - **Interval**: *10*
 - **Picture**: *SleepyDog.png*
 - **Rotates**: *no* (ζετικαρισμένο)

Η ιδιότητα Interval έχει να κάνει με την ομαλότητα τις κίνησης. Στο

παράδειγμά μας έχουμε θέσει να ανανεώνεται η θέση του Sprite κάθε 10 msec. Το **Rotates** έχει να κάνει με το αν περιστρέφεται το Sprite όταν αλλάζει η κατεύθυνσή του. Αν δεν μας αρέσει το μέγεθος του σκύλου μεταβάλλουμε κατάλληλα και τις ιδιότητες **Width** και **Height**.

Palette	
User Interface	
Layout	
Media	
Drawing and Animation	
🔎 Ball	?
🌽 Canvas	?
🔎 ImageSprite	?
Sensors	

Pale	ette	
Us	er Interface	
La	yout	
Me	edia	
Dra	awing and Animation	
,0	Ball	?
<u>V</u>	Canvas	?
1	ImageSprite	?

<u> Βήμα 4: Προσθήκη ήχου</u>

- 8. 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. Δίνουμε στο νέο αντικείμενο το όνομα DogBarkingSound και μεταβάλλουμε τις ιδιότητες:
 - 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 ξαναπαίζει προτού περάσουν τουλάχιστον 0,3 δευτερόλεπτα. Το αντικείμενο **Sound** εκτός του πιθανού ήχου είναι απαραίτητο και για την δόνηση της συσκευής.

<u>Βήμα 5: Συγγραφή Προγράμματος</u>

Μέχρι τώρα, στο **Designer** σχεδιάζουμε την «οθόνη» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της. Για να γίνει αυτό, μεταβαίνουμε στο **Blocks** για να συσχετίσουμε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσουμε, προσθέτοντας τις κατάλληλες εντολές.

MIT App Inventor 2 Beta	Project +	Connect +	Build - Help	My Project	Guide	Report an Issue	akontoge nail.com +
GuardDog	Screen1 •	Add Screen	Remove Screen				Designer Blocks

Υπενθυμίζεται ότι για κάθε πλακίδιο που θέλουμε να προσθέσουμε, ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης, ανοίγουν οι διαθέσιμες εντολές, αναζητούμε και επιλέγουμε το πλακίδιο που χρειαζόμαστε και το σέρνουμε στον χώρο σύνταξης των προγραμμάτων.

9. 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, ανοίγουν οι διαθέσιμες εντολές και μεταφέρουμε το πλακίδιο when DogCanvas.TouchDown.



10. Στη συνέχεια επιλέγουμε το αντικείμενο *DogBarkingSound*, εντοπίζουμε το πλακίδιο **call DogBarkingSound.Play** και το «κουμπώνουμε» μέσα στο προηγούμενο πλακίδιο. Μεταφράζοντας το σενάριο που μόλις δημιουργήσαμε, όταν ο καμβάς πατηθεί ξεκίνα τον ήχο του

γαβγίσματος, βλέπουμε ότι υλοποιήσαμε το ζητούμενο. Παρατηρούμε στην εντολή when DogCanvas.TouchDown και δύο μεταβλητές, τις **x και y**. Αυτές προσδιορίζουν που ακριβώς πάτησε ο χρήστης στην οθόνη της συσκευής μας και θα χρησιμοποιηθούν στη συνέχεια.



11. Εάν επιθυμείτε μπορείτε να δοκιμάσετε τι έχετε υλοποιήσει μέχρι τώρα επιλέγοντας την εντολή Connect - AI Companion.

MIT App Inventor 2 Beta	Projects +	Connect - Build -	Help +
GuardDog	Screen1 • A	AI Companion	creen
Guardbog		Emulator	
Blocks	Viewer	USB	
Built-in		Reset Connection	
Control		Hard Reset	



Βήμα 6: Αλλαγή μορφής

12. 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 την αλλαγή της μορφής του σκύλου, το γεγονός της επαφής με τον καμβά. Θα χρειαστεί, λοιπόν, να συνδυάσουμε κάποιες εντολές από την ομάδα εντολών 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.Τouchdown, το πλακίδιο set DogSprite.Picture to ...

13. Στη συνέχεια θα πρέπει να παρέχουμε το όνομα αρχείου του αγριεμένου σκύλου. Στην ενσωματωμένη ομάδα πλακιδίων **Text** επιλέγουμε το πλακίδιο του κενού κειμένου που συμπληρώνουμε με το όνομα του αρχείου εικόνας *AngryDog.png*.





14. Κατά τον ίδιο τρόπο επαναφέρουμε την εικόνα στον αρχικό κοιμισμένο σκύλο όταν σταματήσει η επαφή, when DogCanvas.Touchup.



15. Εξυπακούεται ότι στο τέλος κάθε βήματος μπορείτε να δοκιμάζετε την εφαρμογή στη συσκευή σας.

Βήμα 7: Κίνηση προς το σημείο επαφής

16. Στη συνέχεια θα κάνουμε τον σκύλο-φύλακά μας λίγο πιο επιθετικό. Όταν αγγίζετε την οθόνη του κινητού σας, ο σκύλος θα κινείται προς το σημείο επαφής. Όπως είδαμε στο προηγούμενο βήμα, στην εντολή when DogCanvas.Touchdown προσδιορίζονται και οι δύο μεταβλητές, x και y, οι οποίες καθορίζουν τις συντεταγμένες του σημείου επαφής με την οθόνη. Αρχικά προσθέτουμε το πλακίδιο call DogSprite.PointInDirection της ομάδας DogSprite που στρέφει την φιγούρα προς το σημείο που θα τεθεί στα x και y.



17. Εάν αφήσουμε το ποντίκι πάνω από τα x και y εμφανίζονται τα πλακίδια get x και get y αντίστοιχα, τα οποία και κουμπώνουμε δίπλα από τα x και y της call DogSprite.PointInDirection



18. Στη συνέχεια πρέπει να ασχοληθούμε με τον τρόπο που θα κινηθεί ο σκύλος προς το σημείο επαφής. Αρχικά πρέπει να θέσουμε την ταχύτητα του,

κάτι που γίνεται με το πλακίδιο set DogSprite.Speed

set DogSprite . Speed to

ομάδα πλακιδίων Math επιλέγουμε αυτό με τον αριθμό 0, το θέτουμε 40 και το κουμπώνουμε στην προηγούμενη εντολή.

set DogSprite 🔹 . Speed 🔹 to 其	40	
--------------------------------	----	--

Εισάγουμε την νέα αυτή εντολή στο σενάριο μας στο when DogCanvas.Touchdown. Το σενάριο μας μέχρι στιγμής έχει την επόμενη μορφή. Το οποίο αν το μεταφράσουμε προκύπτει το εξής σενάριο: Όταν ο χρήστης αγγίξει τον καμβά *DogCanvas*, αναπαράγεται ο ήχος *DogBarkingSound*, η φιγούρα *DogSprite* στρέφεται προς το σημείο επαφής και κινείται με ταχύτητα 40.



19. Όταν ο χρήστης δεν ακουμπά πια το δάχτυλο στην οθόνη η ταχύτητα πρέπει να μηδενιστεί. Έτσι συνολικά το σενάριο που αφορά την εντολή **when DogCanvas.Touchup** παίρνει την επόμενη μορφή.

when	DogCanvas .TouchUp
X	У
do	set DogSprite . Picture to SleepyDog.png
	set DogSprite . Speed to 0

Η εφαρμογή μας είναι έτοιμη και μπορούμε να την δοκιμάσουμε στην android συσκευή μας.

<u> Βήμα 8: Δόνηση</u>

20. Μία βελτίωση στο App αυτό είναι η συσκευή μας επιπλέον να δονείται όταν ακουμπάμε τον σκύλο. Στην ομάδα εντολών DogSprite σέρνουμε στην επιφάνεια των σεναρίων μας την αντίστοιχη εντολή που αφορούσε μέχρι πριν τον καμβά, when DogSprite.Touchdown. Στη συνέχεια από την ομάδα εντολών του ήχου, DogBarkingSound, όπου βρίσκεται και η δόνηση εισάγουμε την εντολή call DogBarkingSound.Vibrate milisecs που προκαλεί τη δόνηση για έναν αριθμό milliseconds που τα θέτουμε από την ομάδα εντολών. Το ολοκληρωμένο σενάριο φαίνεται στη συνέχεια.



Μπορείτε να δοκιμάσετε και την επέκταση αυτή στην συσκευή σας.

Δραστηριότητες

 Αναπτύξτε μια εφαρμογή που θα δίνει τη δυνατότητα στον χρήστη της να παίζει πιάνο. Η εφαρμογή θα αποτελείται από 7 κουμπιά (buttons) - νότες (Ντο, Ρε, Μι, Φα, Σολ, Λα, Σι), όπως φαίνεται στη διπλανή εικόνα. Με το άγγιγμα κάθε κουμπιού θα αναπαράγεται ο ήχος της αντίστοιχης νότας. Τα απαραίτητα αρχεία για την εφαρμογή αυτή μπορείτε να τα βρείτε στο φάκελο Piano.



Δείτε ένα παράδειγμα του κώδικα που πρέπει να χρησιμοποιήσετε για το κουμπί της νότας Ντο!



Σημείωση: 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.

whe	n (Se	creen1 · Initialize
do	set	NotesSound . Source to finite nto.mp3
	set	NotesSound Source - to 4 re.mp3 *
	set	NotesSound . Source to mi.mp3
	set	NotesSound . Source to fa.mp3
	set	NotesSound . Source to sol.mp3 .
	set	NotesSound . Source to (1a.mp3)
	set	NotesSound . Source to isi.mp3

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.

- **2.** Αναπτύξτε μια εφαρμογή που θα λειτουργεί σαν κουμπί πανικού. Η εφαρμογή θα αποτελείται από δύο κουμπιά.
 - 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 ζητήσει βοήθεια. Το κουμπί πανικού θα απενεργοποιείται προσωρινά (ιδιότητα enabled), ώστε να είναι ορατό, αλλά ο χρήστης να μην μπορεί να το ξαναπατήσει. (τα αρχεία θα τα βρείτε στο φάκελο PanicButton)
 - 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.



call SMSTexting .SendMessage

Palette					
User Interface					
Layout					
Media					
Drawing and Animation					
Sensors					
Social					
ContactPicker	۲				
EmailPicker	3				
S PhoneCall	0				
PhoneNumberPicker	۲				
Texting	0				
Twitter	۲				
Storage					

<u>Σημείωση:</u>

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

Για να καθορίσετε το κείμενο του μηνύματος και τον τηλεφωνικό αριθμό που θα αποστέλλεται το μήνυμα, 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 call SMSTexting.SendMessage.

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. Επίσης έχει την ιδιότητα Loop, δηλαδή την αυτόματη επανεκκίνηση του ηχητικού κομματιού όταν αυτό ολοκληρωθεί.

!!! Προσοχή, τα μηνύματα μέσω της εφαρμογής χρεώνονται κανονικά.

<u>Ερωτήσεις</u>

Πηγή: <u>http://www.sepchiou.gr/</u>