



Προγραμματισμός σε App Inventor

Εκπαιδευτικές δραστηριότητες
ανάπτυξης εφαρμογών για Android συσκευές



1η έκδοση

Ιανουάριος 2014

ISBN 978-618-81581-0-8



Αυτό το υλικό διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/>). Η αναφορά σε αυτό θα πρέπει να γίνεται ως εξής:

Προγραμματισμός σε App Inventor. Βασίλης Βασιλάκης, Γιώργος Χατζηνικολάκης. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2014.

ΠΡΟΛΟΓΟΣ

Η τεράστια διείσδυση των έξυπνων φορητών συσκευών στην παγκόσμια αγορά και η μεγάλη τους απήχηση στις νεότερες γενιές είναι γεγονότα που δεν μπορούν να αμφισβητηθούν από κανέναν. Τα γεγονότα αυτά, θέτουν ιδανικές βάσεις για τη μάθηση βασικών προγραμματιστικών εννοιών μέσα από την ανάπτυξη εφαρμογών για τις αγαπημένες συσκευές των νέων, είτε πρόκειται για έξυπνα τηλέφωνα, είτε για ταμπλέτες.

Το μεγάλο μερίδιο της πλατφόρμας Android στην ελληνική αλλά και στην παγκόσμια αγορά, αποτέλεσε ένα λόγο παραπάνω να εστιάσουμε στην ιδιαιτέρως δημοφιλή πλατφόρμα του Android. Προς την κατεύθυνση αυτή, η Google ανέπτυξε το App Inventor, ένα ελεύθερο και ανοικτό περιβάλλον οπτικού προγραμματισμού, ιδανικό για χρήστες με ελάχιστη ή και καθόλου σχετική εμπειρία, το οποίο πλέον αναπτύσσεται και συντηρείται από το MIT (Massachusetts Institute of Technology), ένα από τα μεγαλύτερα και ισχυρότερα Πανεπιστήμια της Αμερικής στον τομέα της πληροφορικής.

Θεωρούμε ότι η πλατφόρμα του App Inventor, ως κατάλληλα σχεδιασμένη για αρχάριους στον προγραμματισμό χρήστες και βελτιστοποιημένη για εκπαιδευτική χρήση, δίνει κίνητρα στους μαθητές, καθώς τους επιτρέπει να δημιουργούν εύκολα εφαρμογές, άμεσα αξιοποιήσιμες από τους ίδιους αλλά και από τον περίγυρό τους, για συσκευές που αγαπάνε και χρησιμοποιούν σε καθημερινή βάση.

Στο πλαίσιο αυτό, αναπτύξαμε το παρόν εκπαιδευτικό υλικό, προσπαθώντας μέσα από προσεκτικά επιλεγμένες και όσο το δυνατόν πρωτότυπες δραστηριότητες να εισαγάγουμε τους μαθητές σε βασικές έννοιες προγραμματισμού, ώστε σταδιακά να είναι ικανοί να μετατρέπουν τις ιδέες τους σε πράξη, σχεδιάζοντας και αναπτύσσοντας εφαρμογές που να έχουν νόημα για τους ίδιους. Κατ' αυτό τον τρόπο, μετατρέπονται από παθητικοί χρήστες και απλοί "καταναλωτές" της τεχνολογίας σε δημιουργοί και "εφευρέτες", σε μια περίοδο που, στην ελληνική παιδεία, η πληροφορική λογίζεται κυρίως ως βοηθητικό εργαλείο για άλλα μαθήματα (ΤΠΕ) και όχι ως αυτόνομο και δημιουργικό γνωστικό αντικείμενο.

Το υλικό αυτό, λοιπόν, αποτελείται από έναν αριθμό μαθημάτων με συγκεκριμένους μαθησιακούς στόχους, οι οποίοι περιγράφονται στην αρχή κάθε μαθήματος. Κάθε μάθημα περιλαμβάνει αναλυτικά βήματα για το σχεδιασμό και την ανάπτυξη διαφόρων εφαρμογών, ενώ σταδιακά οι οδηγίες γίνονται πιο συνοπτικές, όταν αναφέρονται σε έννοιες και διαδικασίες που θεωρούνται γνωστές.

Ορισμένες επεκτάσεις των εφαρμογών αφήνονται ως ασκήσεις για τους μαθητές, ενώ στο τέλος κάθε μαθήματος υπάρχουν διάφορες δραστηριότητες για το σπίτι, ώστε οι μαθητές να επιλέγουν εκείνες που τους ενδιαφέρουν περισσότερο. Στο τέλος κάθε μαθήματος παρατίθενται επίσης ενδεικτικές λύσεις όλων των δραστηριοτήτων και επεκτάσεων. Τέλος, στο παράρτημα περιγράφονται τα προαπαιτούμενα για τη χρήση του App Inventor και την άμεση δοκιμή των εφαρμογών σε προσομοιωτή ή σε συσκευή Android.

Εκ μέρους του Συλλόγου Εκπαιδευτικών Πληροφορικής Χίου,

Βασίλης Βασιλάκης και Γιώργος Χατζηνικολάκης

ΠΕΡΙΕΧΟΜΕΝΑ

Μάθημα 1

Στόχοι.....	3
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	83
Δραστηριότητα : Πυξίδα	96
Δραστηριότητα : Πιάσε τη σημαία.....	100
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

App Inventor



ΜΑΘΗΜΑ 1

ΣΤΟΧΟΙ

- 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.



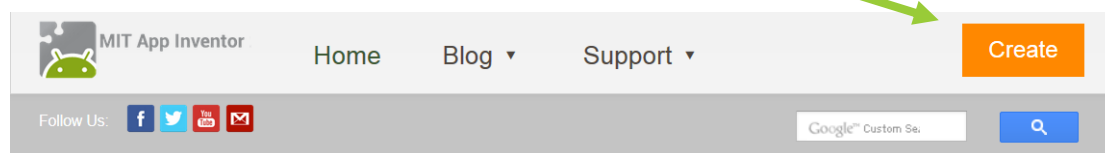
Εικόνα 1 - Περιβάλλον της εφαρμογής Ζάρια

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

Βήμα 1 Δημιουργία νέου έργου στο App Inventor

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

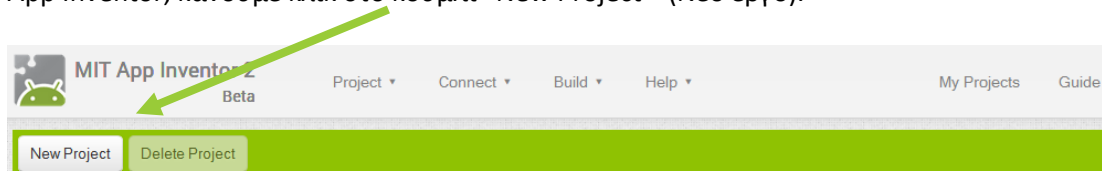


Εικόνα 2- Αρχική σελίδα MIT App Inventor

... ή πληκτρολογούμε τη διεύθυνση <http://ai2.appinventor.mit.edu/> για απευθείας μετάβαση στο περιβάλλον δημιουργίας εφαρμογών.

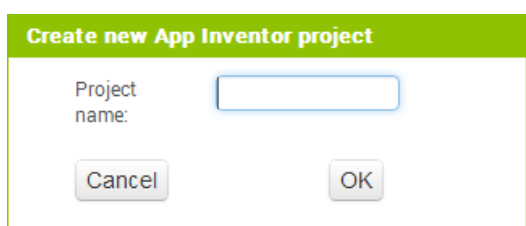
Θα εμφανιστεί η σελίδα για σύνδεση στο λογαριασμό που έχουμε στη Google. Δίνουμε το όνομα χρήστη και το συνθηματικό του λογαριασμού μας και πατάμε “Sign In” (Σύνδεση).

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



Εικόνα 3- Δημιουργία νέου έργου

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



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 OK.

Designer Γραφικό Περιβάλλον Εφαρμογής

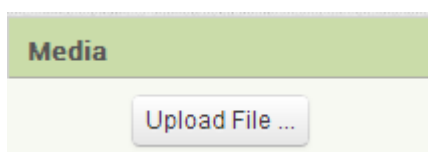
Αρχικά, βρισκόμαστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη *διεπαφή* (interface) της εφαρμογής μας, προσθέτοντας τα απαραίτητα *αντικείμενα-συστατικά* (components) και ορίζοντας *ιδιότητες* (properties) για αυτά.

Βήμα 2 Μεταφόρτωση αρχείων στο App Inventor

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

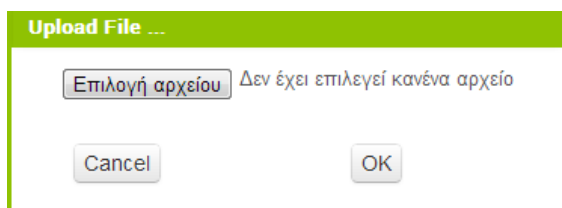
Αρχικά επισκεπτόμαστε το σύνδεσμο <http://tiny.cc/aifiles>. Μπαίνουμε στο φάκελο RollDice και κατεβάζουμε στον υπολογιστή μας τις εικόνες των ζαριών, καθώς και τον σχετικό ήχο.

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




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.

	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

Βήμα 3

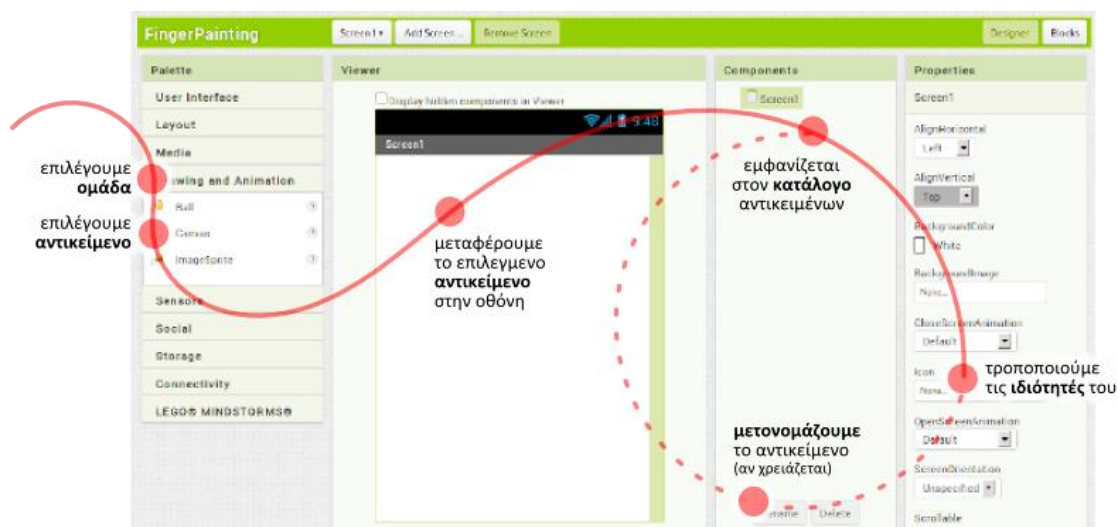
Εισαγωγή αντικειμένων στην περιοχή σχεδίασης

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:

	επιλέγουμε το συστατικό	μεταβάλλουμε τις ιδιότητες
	Screen1	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.



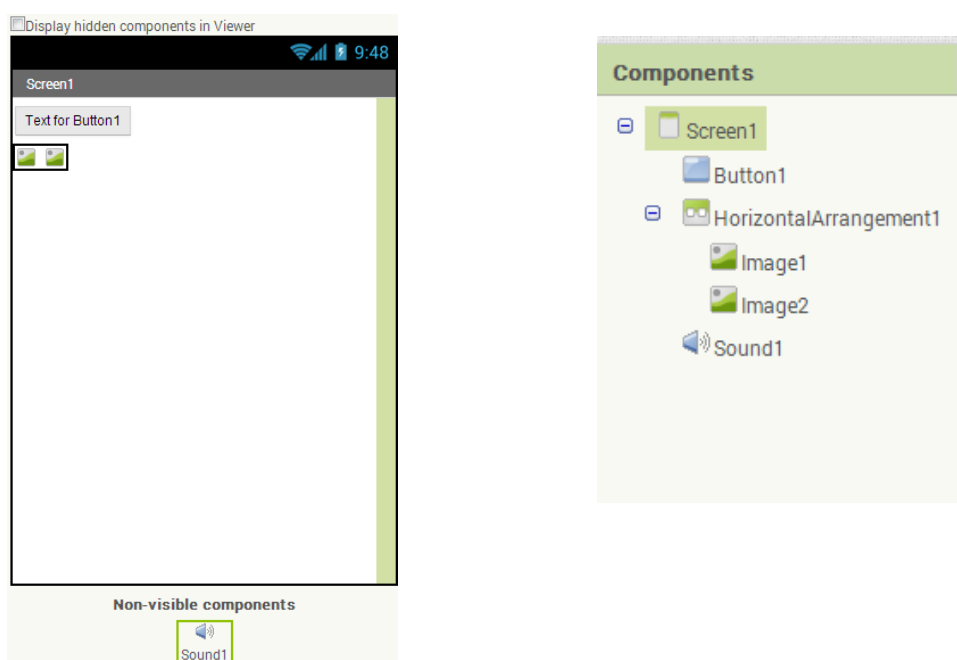
Εικόνα 7 - Η διαδικασία προσθήκης αντικειμένων στην περιοχή σχεδίασης

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:



Εικόνα 8 - Αρχική μορφή εφαρμογής Ζάρια

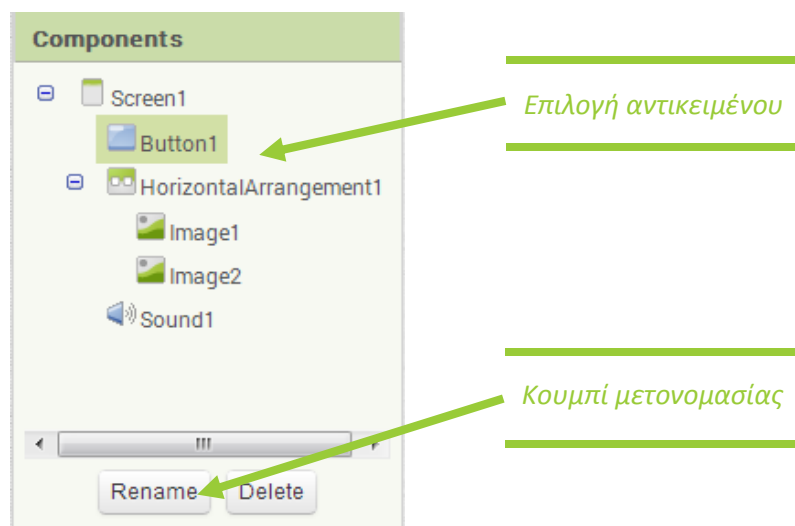
Βήμα 4 Καθορισμός Ιδιοτήτων αντικειμένων

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.

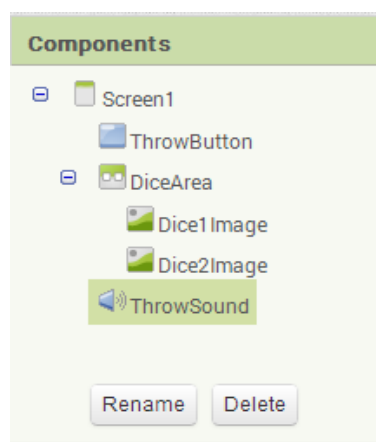


Τα ονόματα των αντικειμένων πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.



Εικόνα 9 - Μετονομασία Αντικειμένου

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



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.

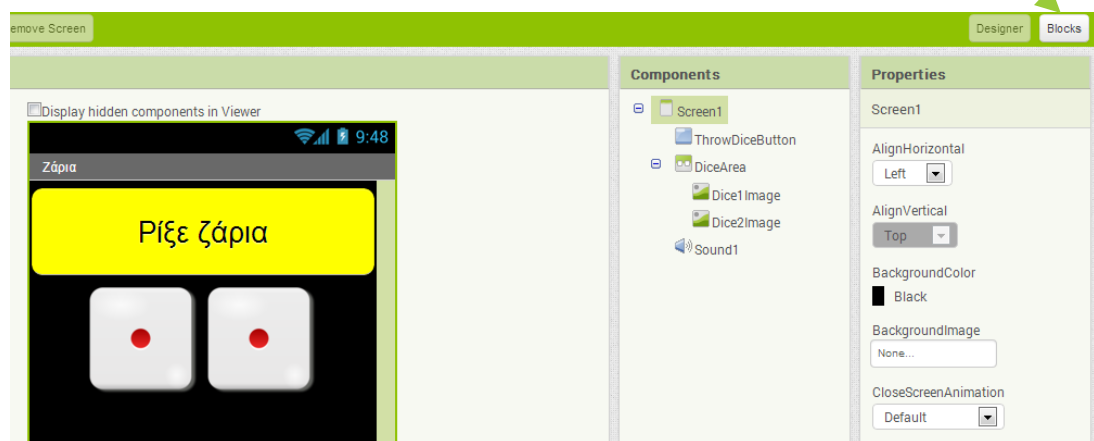


Εικόνα 11 - Τελική μορφή εφαρμογής Ζάρια

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.

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



Εικόνα 12 - Κουμπί Blocks

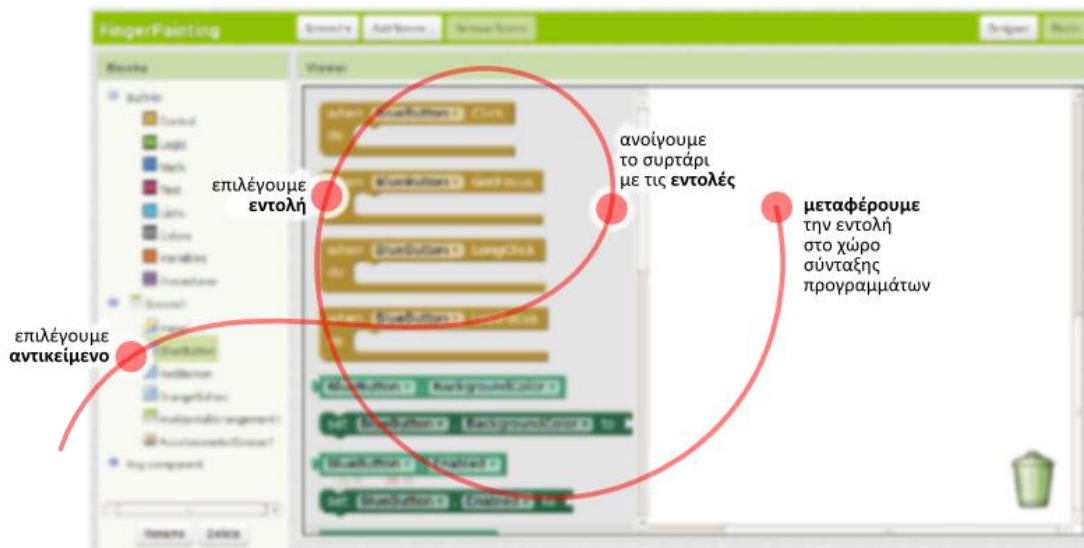
Βήμα 5 Γεγονός - Άγγιγμα του κουμπιού «Ρίξε ζάρια»

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.

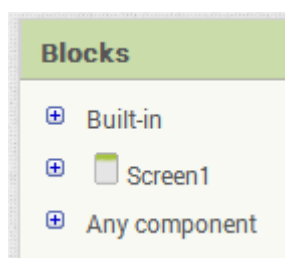
Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή μας πρακτικά συναρμολογούμε πλακίδια εντολών (και όχι μόνο). Για κάθε πλακίδιο που θέλουμε να προσθέσουμε **ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης**, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες εντολές, αναζητούμε 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 13 - Διαδικασία προσθήκης εντολών για τα αντικείμενα

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



Εικόνα 14 - Ομάδες πλακιδίων

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.

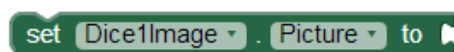
Ακολουθεί η τελευταία ομάδα (Any component), που δε θα μας απασχολήσει εδώ.

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.

Βήμα 6 Ρίξιμο ζαριού – τυχαίοι αριθμοί

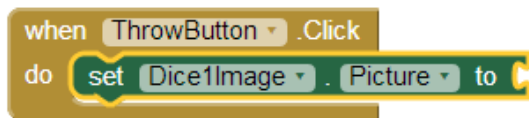
Για να προσομοιώσουμε το ρίξιμο των ζαριών θα πρέπει να παράγουμε 2 τυχαίους αριθμούς, έναν για το κάθε ζάρι και να θέτουμε, ανάλογα με τον αριθμό που θα παραχθεί, και την κατάλληλη εικόνα ζαριού στα αντίστοιχα σημεία.

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.

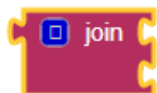


Εικόνα 16 - Μετά την προσθήκη της εντολής αλλαγής εικόνας

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.

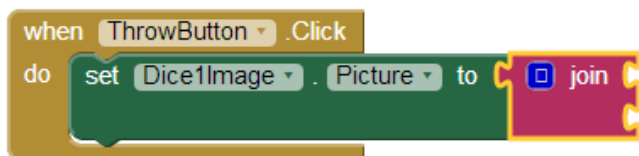
Το ένα τμήμα του ονόματος θα είναι ο αριθμός της εικόνας, τον οποίο θα παράγουμε με τυχαίο τρόπο και το δεύτερο τμήμα του κειμένου θα είναι η επέκτασή του («.png»).

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



Εικόνα 17 - Εντολή join

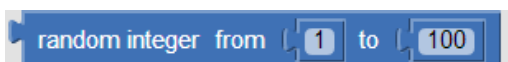
... την οποία και θα κουμπώσουμε στην εγκοπή της εντολής αλλαγής εικόνας.



Εικόνα 18 - Μετά την προσθήκη της εντολής join

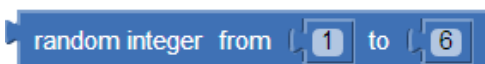
Στην πρώτη εγκοπή της εντολής θα βάλουμε το όνομα του αρχείου εικόνας και στη δεύτερη εγκοπή την κατάληξή του.

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 ...



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 της 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.



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

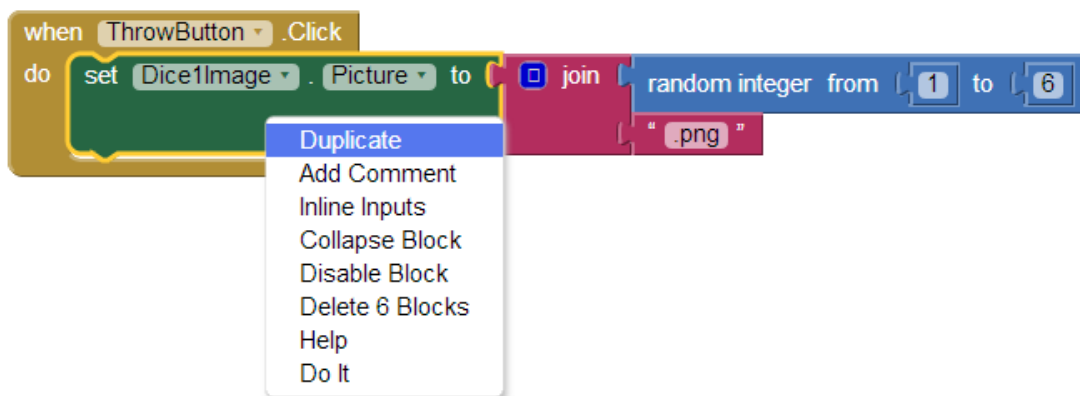


Όταν πατηθεί το κουμπί θέσε την εικόνα του πρώτου ζαριού σε ένα αρχείο με όνομα (τυχαίος ακέραιος αριθμός από 1 μέχρι 6).png

Βήμα 7 Ρίξιμο δεύτερου ζαριού - Διπλασιασμός εντολών

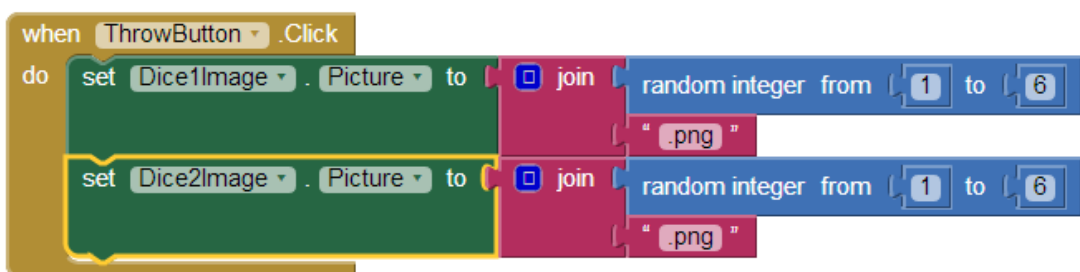
Είναι προφανές ότι θέλουμε να εκτελεστούν ακριβώς οι ίδιες ενέργειες για το αντικείμενο Dice2Image. Αντί να ξανακάνουμε τη διαδικασία που περιεγράφηκε προηγουμένως μπορούμε απλά να διπλασιάσουμε την εντολή **set Dice1Image.Picture to** και να την τροποποιήσουμε κατάλληλα για να λειτουργεί με την εικόνα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

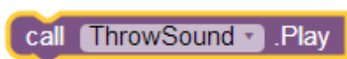


Εικόνα 25 - Οι εντολές μετά το διπλασιασμό

Βήμα 8 Αναπαραγωγή ήχου ζαριών

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

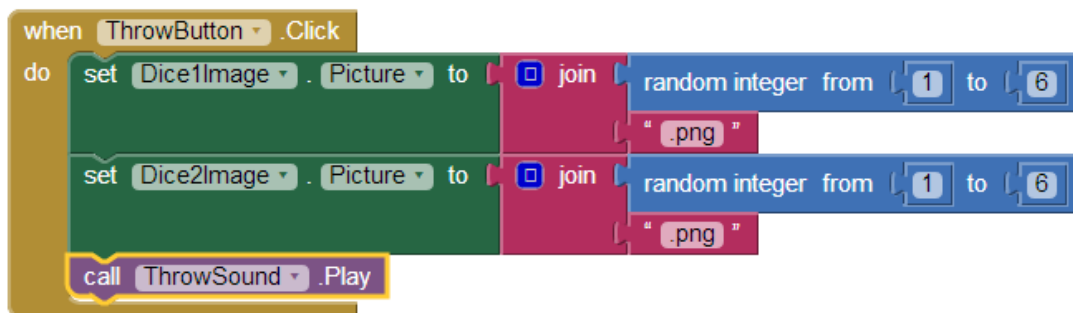
Για να ξεκινήσει η αναπαραγωγή του ήχου θα χρησιμοποιήσουμε την εντολή ...



Εικόνα 26 - Εντολή αναπαραγωγής ήχου

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

Το ολοκληρωμένο πρόγραμμα φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 27 - Τελικό πρόγραμμα Ζάρια



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

ΕΠΕΚΤΑΣΕΙΣ

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.



ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ 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 ☺).

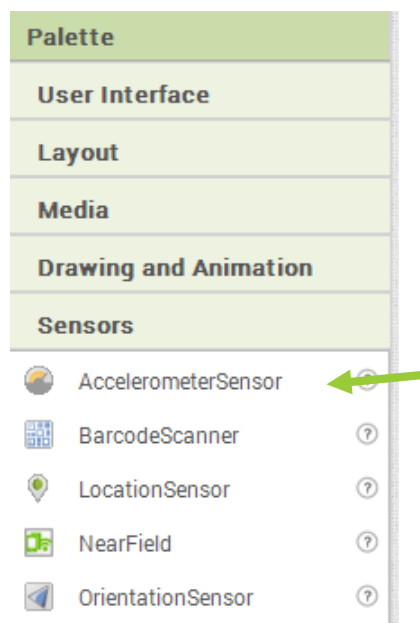
Περισσότερα για το επιταχυνσιόμετρο θα μάθουμε στο Μάθημα 2.

Βήμα 8 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.

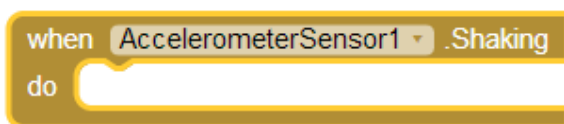


Εικόνα 28 - Επιλογή του επιταχυνσιόμετρου

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

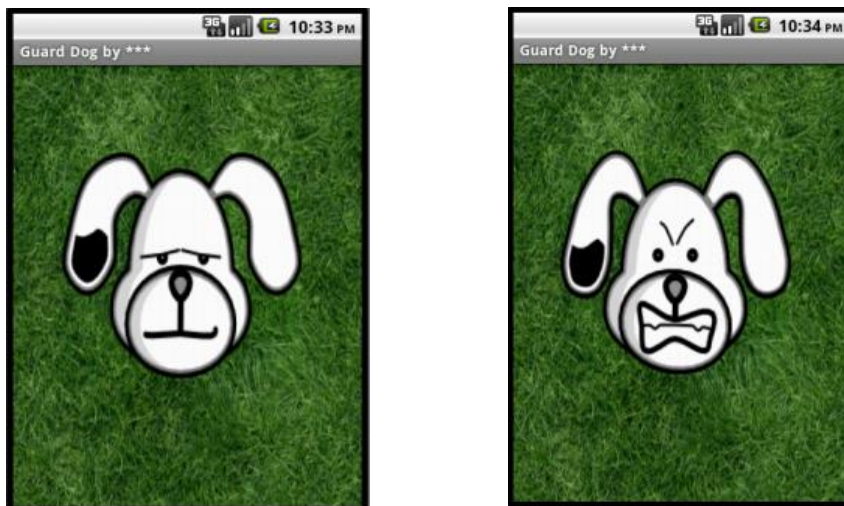
Οι εντολές που θα εκτελούνται με την κίνηση της συσκευής θα είναι ακριβώς οι ίδιες με αυτές του αγγίγματος του κουμπιού. Τις προσθέτουμε, αξιοποιώντας τη λειτουργία του διπλασιασμού εντολών (duplicate).

Μπορείτε να δείτε την τελική μορφή της εφαρμογής μετά την προσθήκη του Accelerometer στο Παράρτημα «[Λύσεις των ασκήσεων](#)».

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

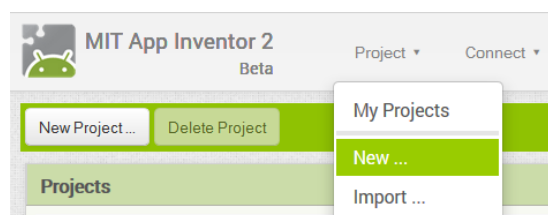
Συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν κάποιος αγγίζει την οθόνη της συσκευής, ενώ επιπλέον τρέχει προς το σημείο που αγγίζουμε την οθόνη.

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

Βήμα 1 Δημιουργία νέου 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

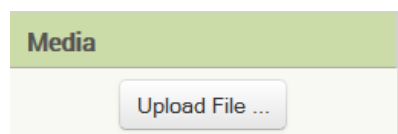
Title: Guard Dog

Βήμα 2

Προσθήκη των απαραίτητων αρχείων πολυμέσων

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

Από το πλαίσιο Media, κάτω δεξιά, επιλέγουμε την εντολή **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:



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

Βήμα 3

Προσθήκη του καμβά

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 Animation	Canvas	DogCanvas	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 4 Προσθήκη του σκύλου

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 Animation	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
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

μεταβάλλουμε τις
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

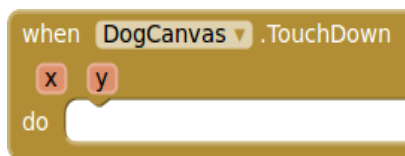
Μέχρι στιγμής, στο Designer έχουμε σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της.

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).

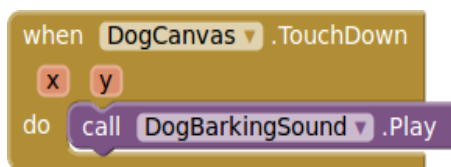
Στην συγκεκριμένη περίπτωση, θέλουμε να ακούγεται ο ήχος του γαβγίσματος DogBarkingSound κάθε φορά που ο χρήστης αγγίζει τον καμβά DogCanvas.

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.



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

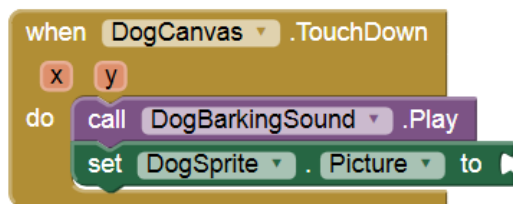
Στο βήμα αυτό θα επεκτείνουμε την συμπεριφορά του σκύλου έτσι ώστε να αλλάζει μορφή και να αγριεύει όταν αγγίζουμε είτε τον ίδιο, είτε την οθόνη.

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 αγγίζει 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:



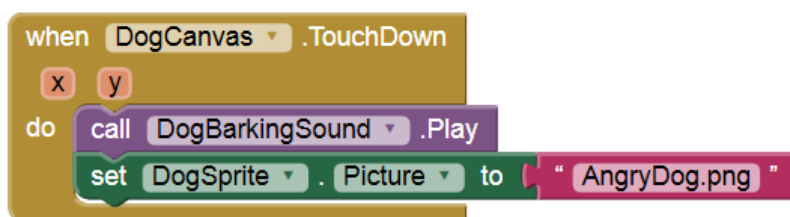
Εικόνα 35 - Μετά και από την προσθήκη της εντολής ορισμού εικόνας

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...



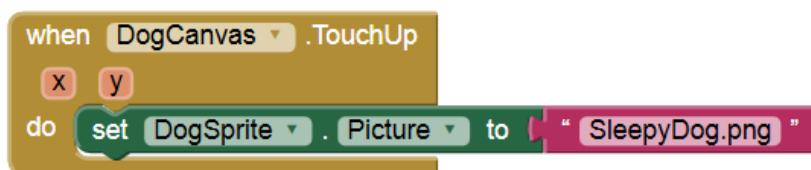
Εικόνα 36 - Το πλακίδιο για την εισαγωγή κειμένου

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



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

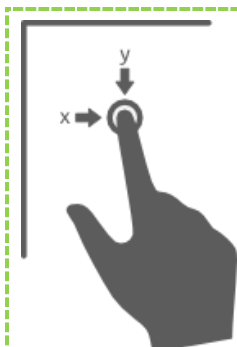


Όταν ο χρήστης αγγίζει τον καμβά DogCanvas, τότε ακούγεται ο ήχος DogBarkingSound και η φιγούρα του σκύλου παίρνει την μορφή AngryDog.png.
Όταν σταματήσει η επαφή με τον καμβά DogCanvas τότε η φιγούρα του σκύλου παίρνει την μορφή SleepyDog.png.

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 DogSprite να στρέφεται και να κινείται προς το σημείο επαφής.

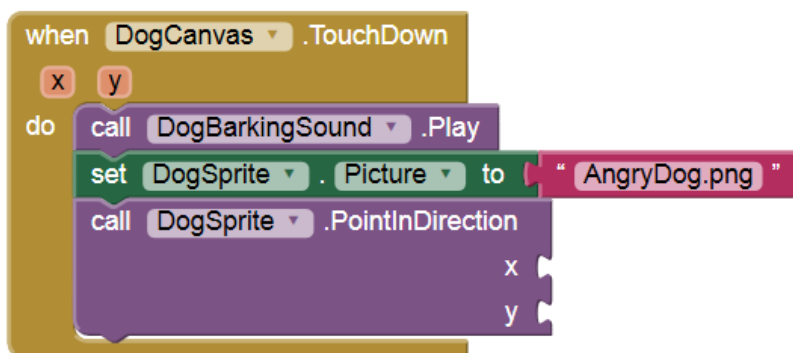


Διαθέτουμε ήδη από το προηγούμενο βήμα το when DogCanvas.TouchDown, το οποίο προσδιορίζει και το σημείο στο οποίο έγινε η επαφή, δηλαδή την οριζόντια απόσταση x από το αριστερό άκρο του καμβά και την κάθετη απόσταση y από το πάνω άκρο του καμβά.

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

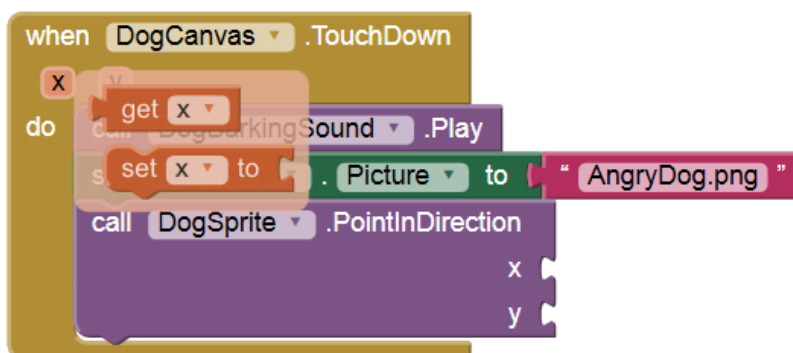
Στο βήμα αυτό θα χρησιμοποιήσουμε τις συντεταγμένες x και y όπου έγινε η επαφή, για να καθορίσουμε προς τα που πρέπει να στραφεί ο σκύλος.

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



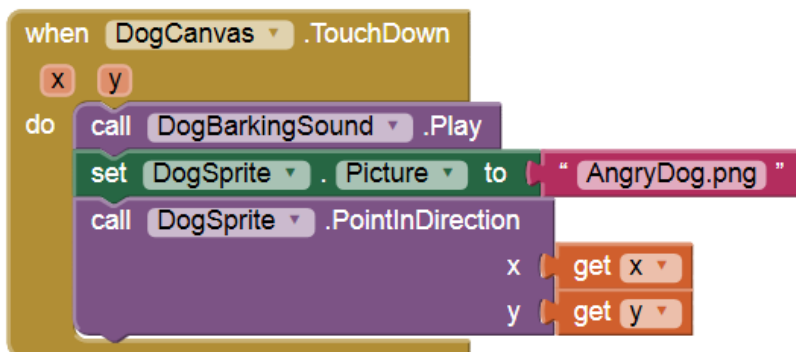
Εικόνα 39 - Προσθήκη της εντολής για στροφή της φιγούρας προς κάποια κατεύθυνση

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...



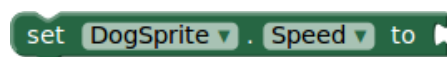
Εικόνα 40 - Εμφάνιση επιπλέον πλακιδίων μέσω των ετικετών

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



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**.



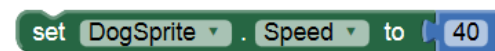
Εικόνα 42 - Εντολή ορισμού ταχύτητας της φιγούρας

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



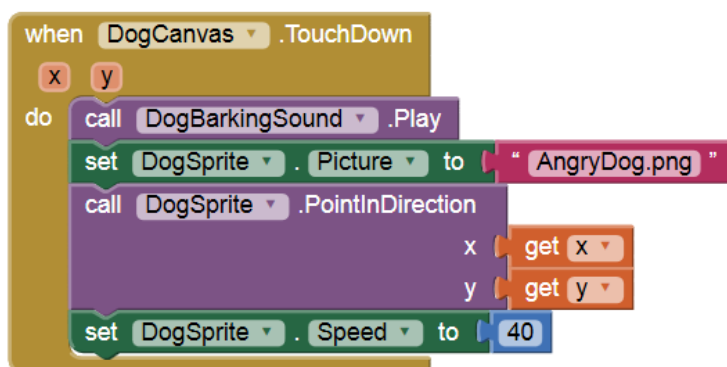
Εικόνα 43 - Το πλακίδιο εισαγωγής αριθμού

...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.



Εικόνα 45 - Το ολοκληρωμένο σετ εντολών για το συμβάν αγγίγματος της οθόνης



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.

```

when DogCanvas .TouchUp
  x y
do
  set DogSprite . Picture to "SleepyDog.png"
  set DogSprite . Speed to 0
  
```

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:

```

when DogCanvas .TouchDown
  x y
do
  call DogBarkingSound .Play
  set DogSprite . Picture to "AngryDog.png"
  call DogSprite .PointInDirection
    x get x
    y get y
  set DogSprite . Speed to 40

when DogCanvas .TouchUp
  x y
do
  set DogSprite . Picture to "SleepyDog.png"
  set DogSprite . Speed to 0
  
```

Εικόνα 47 - Το ολοκληρωμένο σετ εντολών



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

ΕΠΕΚΤΑΣΕΙΣ

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

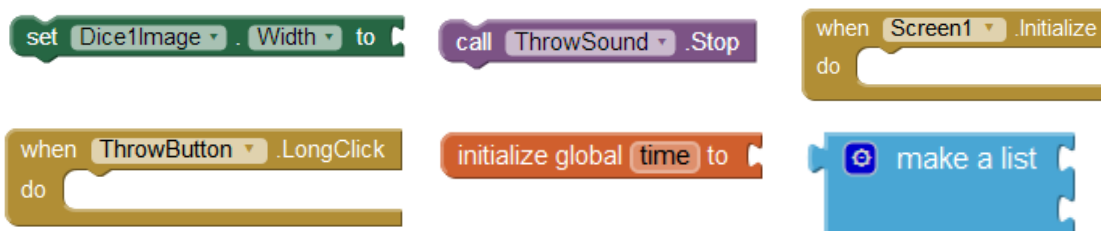
1. Επισκεφθείτε το Google Play Store και συγκεκριμένα τη διεύθυνση:

<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 ξενάγηση στο ηλιακό μας σύστημα.	Εντυπωσιακά γραφικά. Χρήσιμες πληροφορίες.	Δεν επιτρέπει τον κατακόρυφο προσανατολισμό του τηλεφώνου.

2. Κατηγοριοποιήστε τις παρακάτω εντολές σε **Γεγονότα** και **Ενέργειες**.



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 <http://tiny.cc/aifiles> 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.

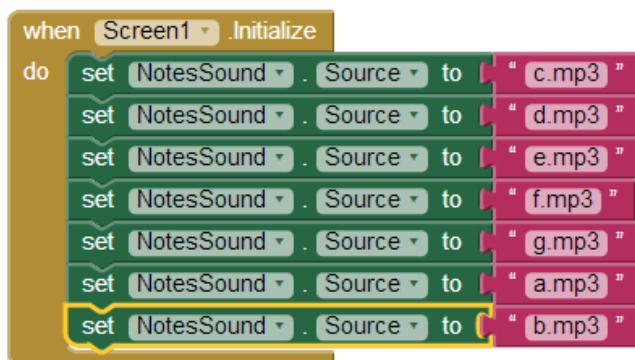
Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <http://tiny.cc/aifiles> και το φάκελο Ριάνο.



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.



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.



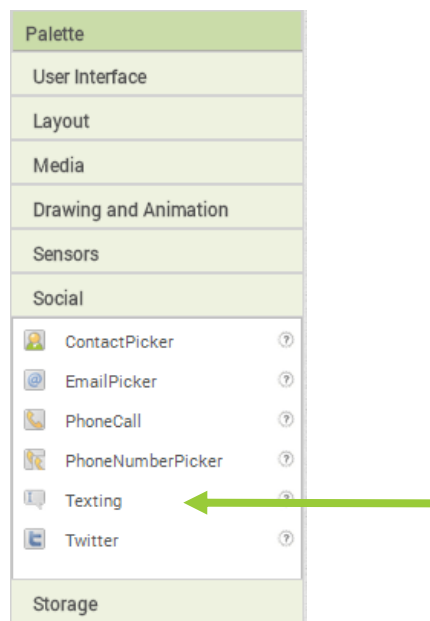
Εικόνα 51 - Μια εκδοχή της εφαρμογής Κουμπί Πανικού

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.

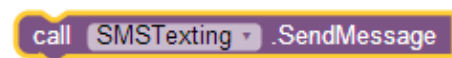
Για την αποστολή του SMS θα χρειαστείτε το αντικείμενο **Texting** που βρίσκεται στην ομάδα **Social**.



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 <http://tiny.cc/aifiles> και το φάκελο Panic Button.