



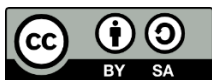
# Προγραμματισμός σε App Inventor

Α' μέρος



1η έκδοση

Ιανουάριος 2014



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 ιδιαίτερως 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.

Πιστεύουμε 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.....	6
Δραστηριότητα : Ζάρια.....	7
Περιγραφή εφαρμογής.....	7
Βασικές έννοιες.....	7
Επεκτάσεις .....	18
Το αντικείμενο Accelerometer .....	18
Δραστηριότητα : Guard Dog.....	20
Περιγραφή εφαρμογής.....	20
Βασικές έννοιες.....	20
Επεκτάσεις .....	28
Δραστηριότητες για το σπίτι .....	29
Ενδεικτικές λύσεις των ασκήσεων .....	33
Ζάρια – Προσθήκη Accelerometer .....	33
Κορώνα – Γράμματα .....	33
Κουμπί Πανικού.....	33
Πιάνο .....	34
Guard Dog.....	34

### Μάθημα 2

Στόχοι.....	35
Δραστηριότητα : Ζάρια – Επέκταση .....	36
Περιγραφή επέκτασης .....	36
Βασικές έννοιες.....	36
Η έννοια της μεταβλητής.....	37
Η εντολή If (Αν) .....	40
Επεκτάσεις .....	44
Δραστηριότητα : GuardDog – Επέκταση .....	45
Περιγραφή Επέκτασης .....	45
Βασικές έννοιες.....	45
Δραστηριότητα : Κορώνα Γράμματα – Επέκταση .....	49
Περιγραφή επέκτασης .....	49
Βασικές έννοιες.....	49
Η εντολή Else (Αλλιώς).....	52
Ο λογικός τελεστής Not (Όχι) .....	56

Δραστηριότητες για το σπίτι .....	57
Ενδεικτικές λύσεις των ασκήσεων .....	60
Ζάρια – Επέκταση με έλεγχο για διπλές και καθαρισμό ετικέτας.....	60
Ζάρια – Επέκταση με εμφάνιση συνολικών παιχνιδιών & ποσοστού νικών .....	60
Κορώνα Γράμματα – Επέκταση με επιλογή χρήστη .....	61
Αντίστροφη Μέτρηση .....	62
Μετατροπή βαθμών Κελσίου σε Φαρενάιτ με πεδίο κειμένου .....	62
Μετατροπή βαθμών Κελσίου σε Φαρενάιτ με μεταβολέα (Slider).....	63
Μετατροπή ταχύτητας με μεταβολέα (Slider) .....	63
Πηγές .....	64
Παράρτημα – Προαπαιτούμενα για τη χρήση του app Inventor .....	65
Παράρτημα – Προετοιμασία για πακετάρισμα εφαρμογής και διαμοιρασμό της.....	69

# 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.



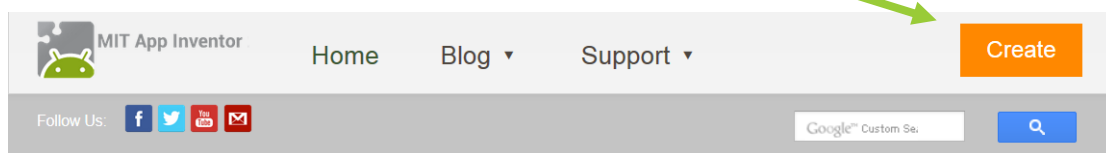
Εικόνα 1 - Περιβάλλον της εφαρμογής Ζάρια

### 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 ...

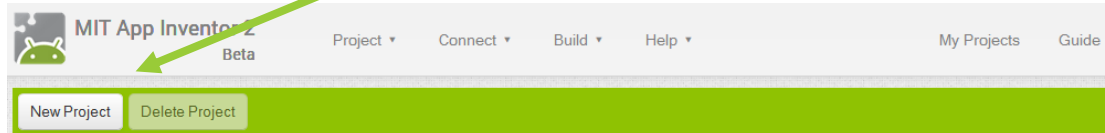


Εικόνα 2- Αρχική σελίδα MIT App Inventor

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

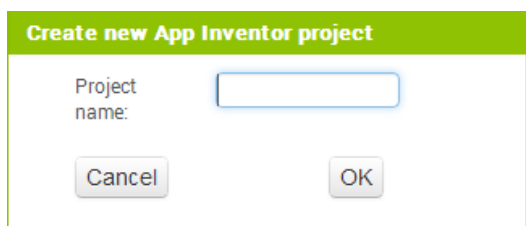
Θα εμφανιστεί η σελίδα για σύνδεση στο λογαριασμό που έχουμε στη Google. Δίνουμε το όνομα χρήστη και το συνθηματικό του λογαριασμού μας και πατάμε “Sign In” (Σύνδεση).

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

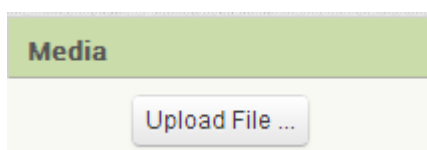
Αρχικά, βρισκόμαστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη *διεπαφή* (interface) της εφαρμογής μας, προσθέτοντας τα απαραίτητα *αντικείμενα-συστατικά* (components) και ορίζοντας *ιδιότητες* (properties) για αυτά.

## 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.

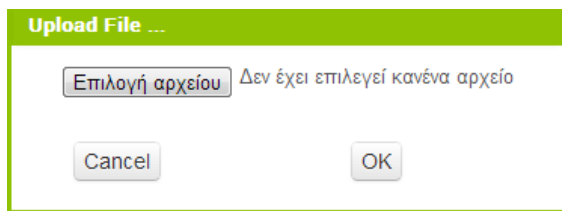
Στη συνέχεια κάνουμε κλικ στο κουμπί “Upload File...” στην περιοχή Media.



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 <b>όνομα</b>	σύντομη <b>περιγραφή</b>
	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

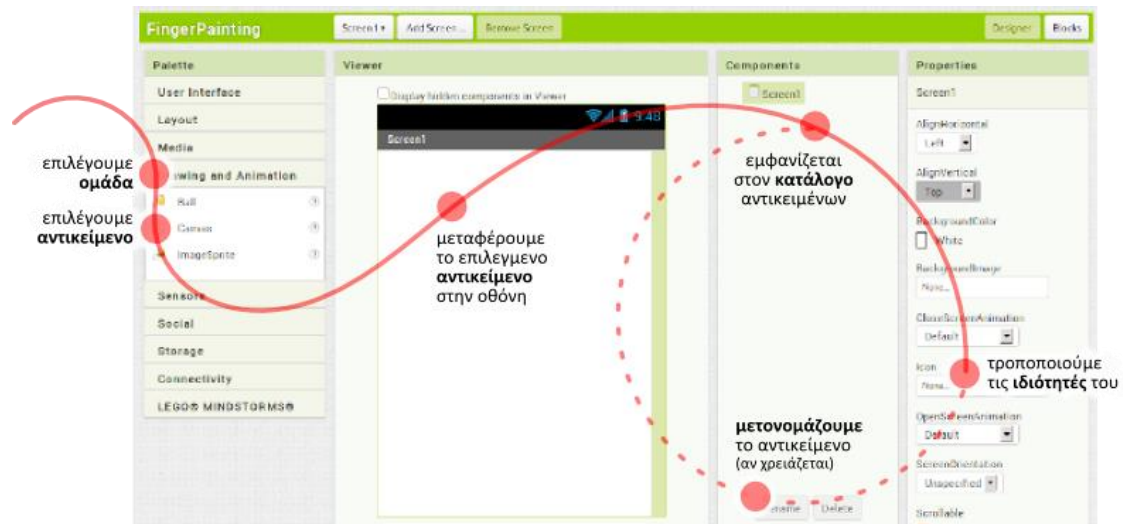
### Βήμα 3 Εισαγωγή αντικειμένων στην περιοχή σχεδίασης

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 <b>συστατικό</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
	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.



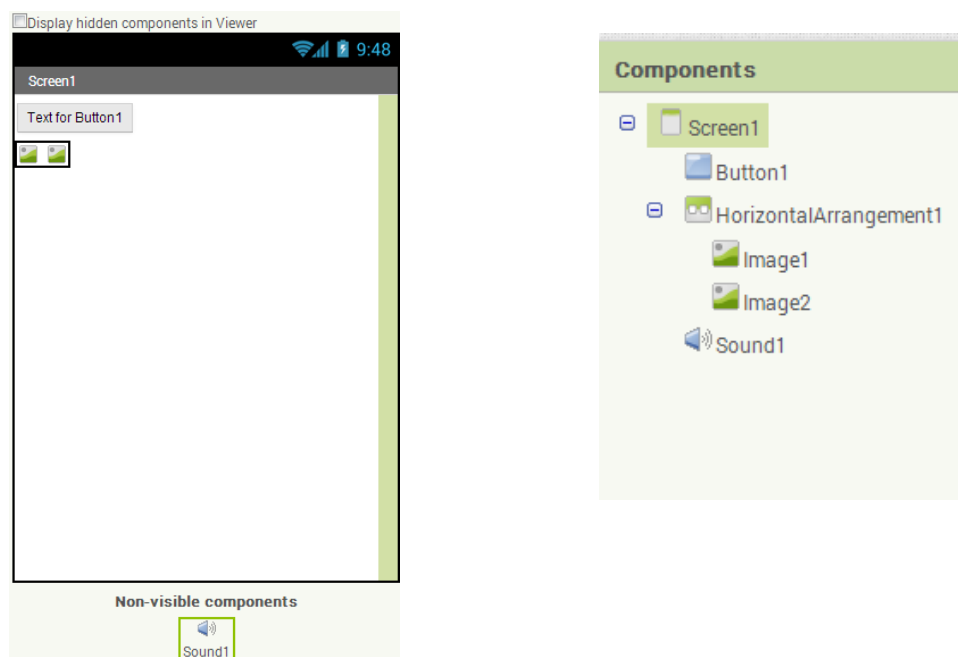
Εικόνα 7 - Η διαδικασία προσθήκης αντικειμένων στην περιοχή σχεδίασης

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

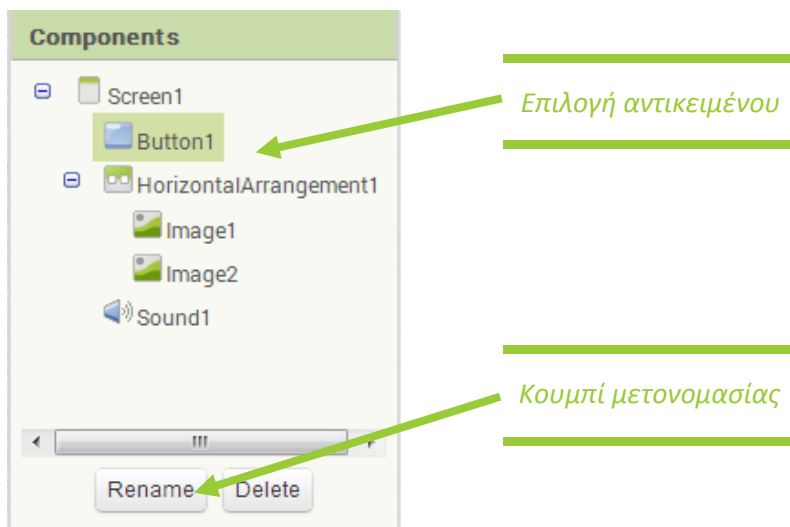
## Βήμα 4 Καθορισμός Ιδιοτήτων αντικειμένων

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.

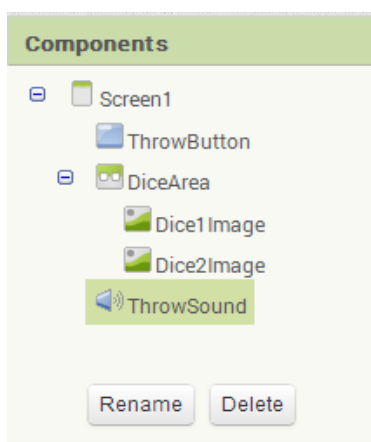


Τα ονόματα των αντικειμένων πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.



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.

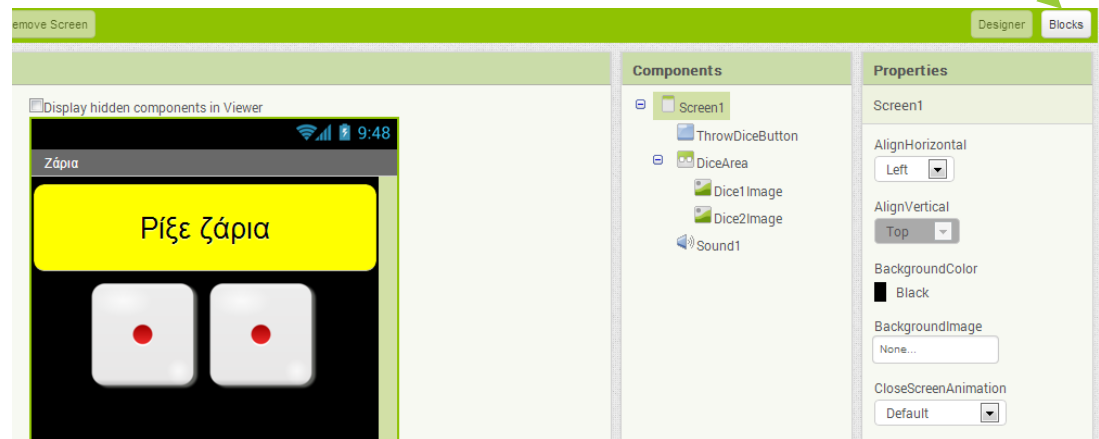


Εικόνα 11 - Τελική μορφή εφαρμογής Ζάρια

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

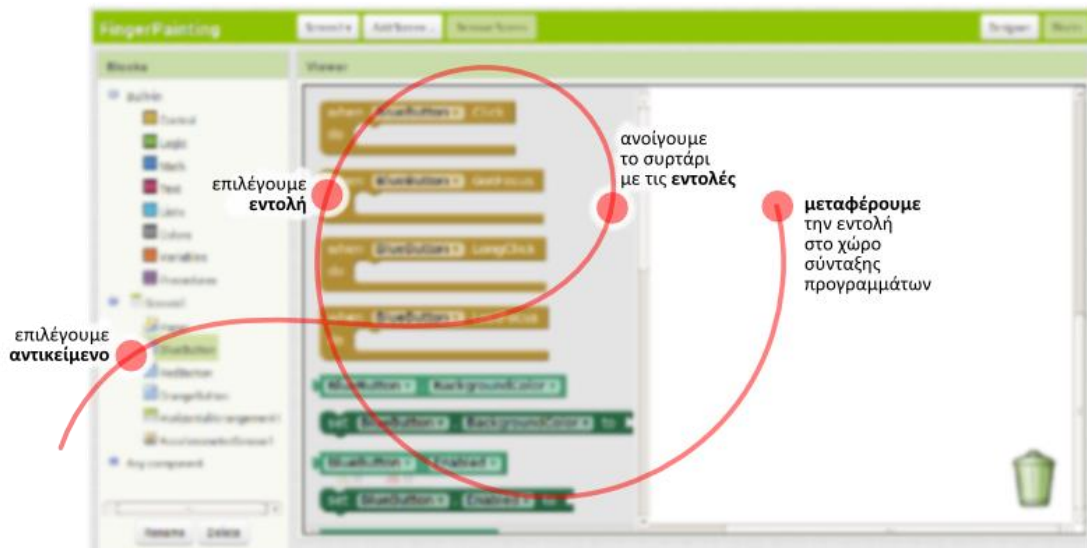
## Βήμα 5 Γεγονός - Άγγιγμα του κουμπιού «Ρίξε ζάρια»

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.

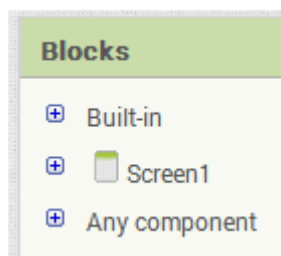
Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή μας πρακτικά συναρμολογούμε πλακίδια εντολών (και όχι μόνο). Για κάθε πλακίδιο που θέλουμε να προσθέσουμε **ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης**, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες εντολές, αναζητούμε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

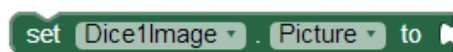
Ακολουθεί η τελευταία ομάδα (Any component), που δε θα μας απασχολήσει εδώ.

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.

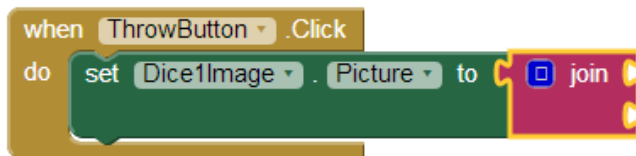
Το ένα τμήμα του ονόματος θα είναι ο αριθμός της εικόνας, τον οποίο θα παράγουμε με τυχαίο τρόπο και το δεύτερο τμήμα του κειμένου θα είναι η επέκτασή του («.png»).

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

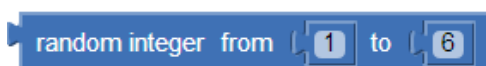
Στην πρώτη εγκοπή της εντολής θα βάλουμε το όνομα του αρχείου εικόνας και στη δεύτερη εγκοπή την κατάληξή του.

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 ...



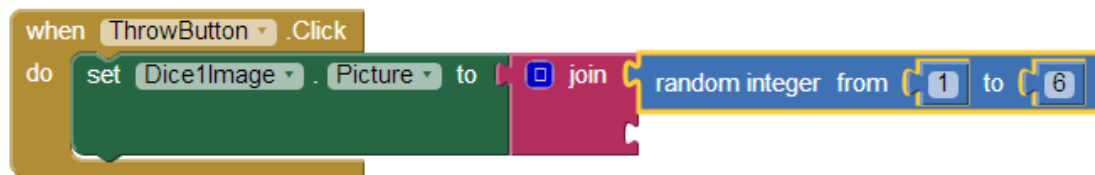
Εικόνα 19 - Εντολή παραγωγής τυχαίων αριθμών

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<sup>η</sup> εγκοπή της εντολής join, όπως παρακάτω.



Εικόνα 21 - Μετά την προσθήκη της εντολής τυχαίων αριθμών

Στη 2<sup>η</sup> 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



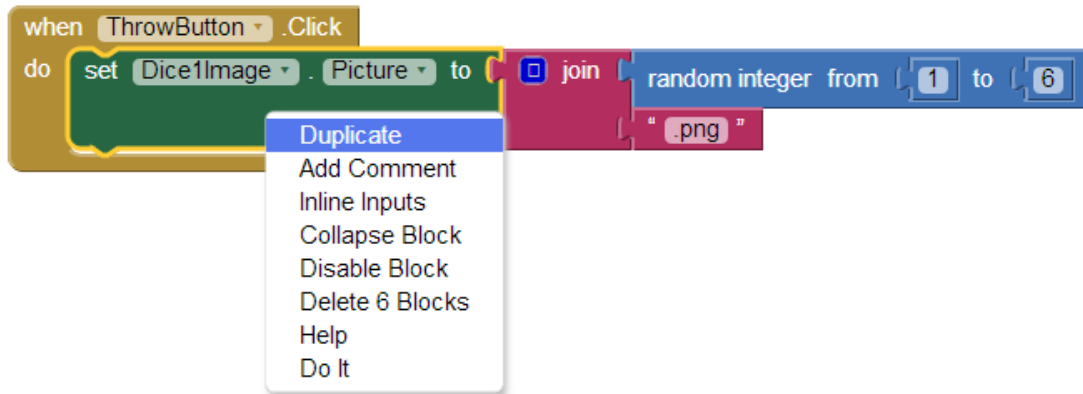
Όταν πατηθεί το κουμπί θέσε την εικόνα του πρώτου ζαριού σε ένα αρχείο με όνομα (τυχαίος ακέραιος αριθμός από 1 μέχρι 6).png

## 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<sup>ου</sup> ζαριού.

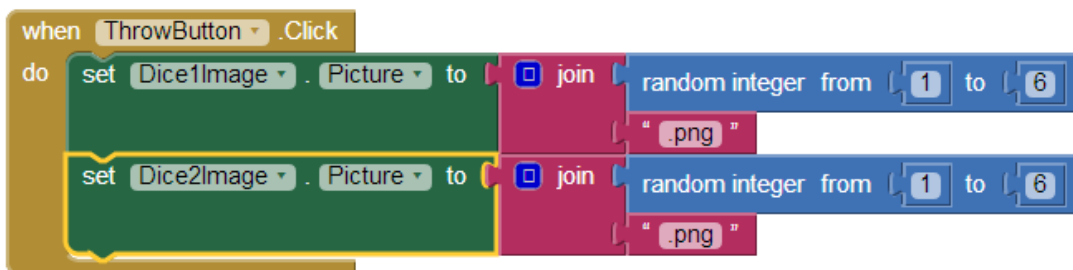
Κάνουμε δεξί κλικ πάνω στην εντολή που θέλουμε να διπλασιάσουμε και επιλέγουμε την εντολή Duplicate (διπλασιασμός).





Εικόνα 24 - Διπλασιάζοντας εντολές

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

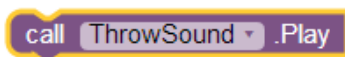


Εικόνα 25 - Οι εντολές μετά το διπλασιασμό

## 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.

```

when ThrowButton.Click
do
  set Dice1Image.Picture to join random integer from 1 to 6 ".png"
  set Dice2Image.Picture to join random integer from 1 to 6 ".png"
  call ThrowSound.Play
  
```

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**



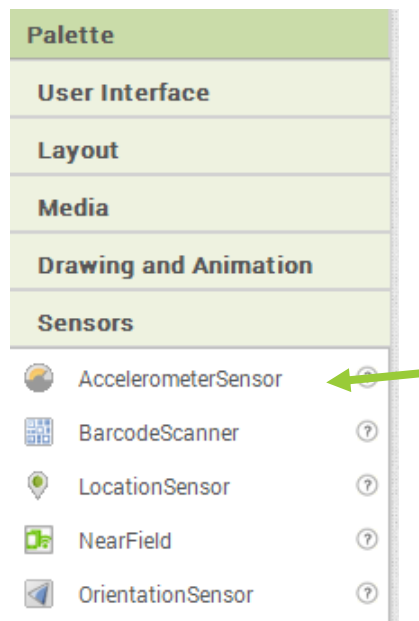
Οι κινητές συσκευές τύπου tablet ή κινητού τηλεφώνου περιλαμβάνουν μια σειρά από αισθητήρες, δηλαδή από εξαρτήματα που μετρούν διάφορα μεγέθη που αφορούν την κινητή συσκευή.  
 Ένας τέτοιος αισθητήρας είναι το Accelerometer ή επιταχυνσιόμετρο στα ελληνικά, με τη βοήθεια του οποίου η κινητή συσκευή παίρνει 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.

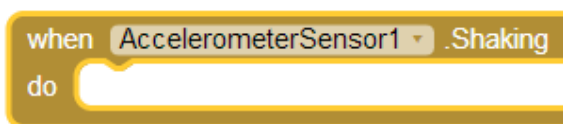


Εικόνα 28 - Επιλογή του επιταχυνσιόμετρου

### 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

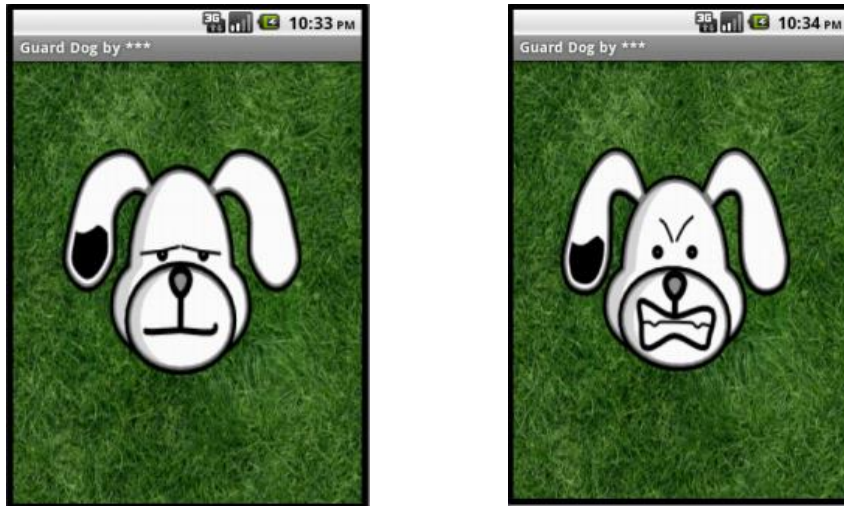
Οι εντολές που θα εκτελούνται με την κίνηση της συσκευής θα είναι ακριβώς οι ίδιες με αυτές του αγγίγματος του κουμπιού. Τις προσθέτουμε, αξιοποιώντας τη λειτουργία του διπλασιασμού εντολών (duplicate).

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

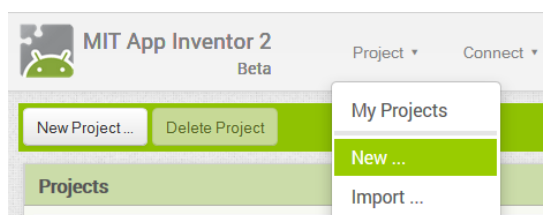
Συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν κάποιος αγγίζει την οθόνη της συσκευής, ενώ επιπλέον τρέχει προς το σημείο που αγγίζουμε την οθόνη.

## 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

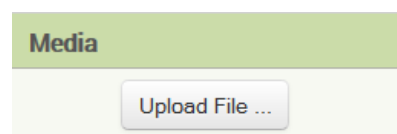
Scrollable: No

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**

σύνομη  
**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**

Media

μεταφέρουμε το  
**αντικείμενο**

Sound

του δίνουμε το  
**όνομα**

DogBarkingSound

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 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

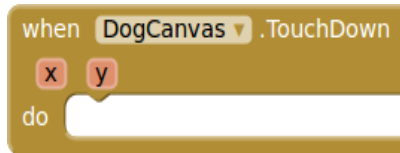
Μέχρι στιγμής, στο Designer έχουμε σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της.

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).

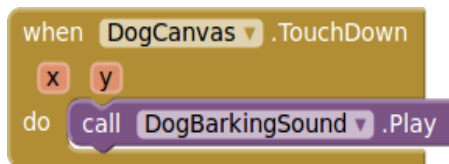
Στην συγκεκριμένη περίπτωση, θέλουμε να ακούγεται ο ήχος του γαβγίσματος DogBarkingSound κάθε φορά που ο χρήστης αγγίζει τον καμβά DogCanvas.

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.

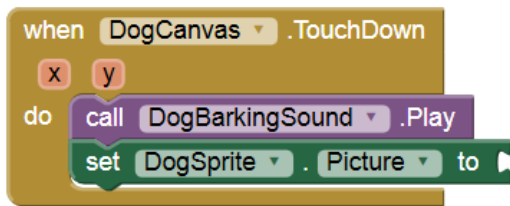
Εδώ θέλουμε η φιγούρα του σκύλου DogSprite να αλλάζει μορφή, όταν ο χρήστης αγγίζει είτε την οθόνη (τον καμβά με το χορτάρι), είτε τον ίδιο το σκύλο και να επανέρχεται όταν η επαφή σταματά.

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:





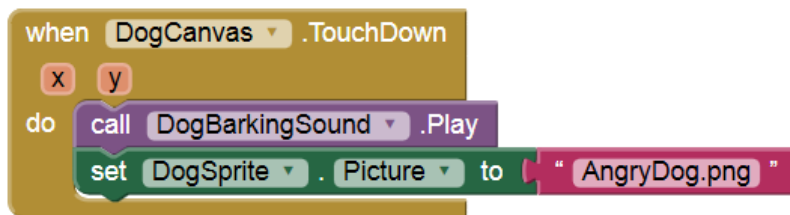
Εικόνα 35 - Μετά και από την προσθήκη της εντολής ορισμού εικόνας

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...



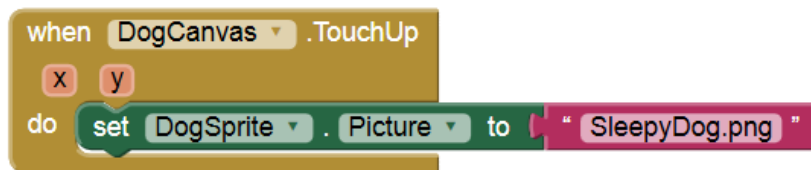
Εικόνα 36 - Το πλακίδιο για την εισαγωγή κειμένου

...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

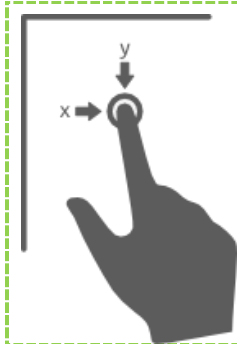


Όταν ο χρήστης αγγίζει τον καμβά DogCanvas, τότε ακούγεται ο ήχος DogBarkingSound και η φιγούρα του σκύλου παίρνει την μορφή AngryDog.png.  
Όταν σταματήσει η επαφή με τον καμβά DogCanvas τότε η φιγούρα του σκύλου παίρνει την μορφή SleepyDog.png.

## 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.

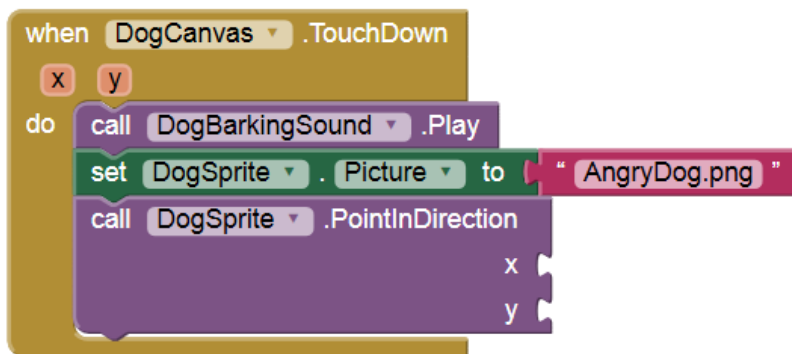
Πιο συγκεκριμένα, εδώ θέλουμε όταν ο χρήστης αγγίζει τον καμβά DogCanvas η φιγούρα DogSprite να στρέφεται και να κινείται προς το σημείο επαφής.



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.

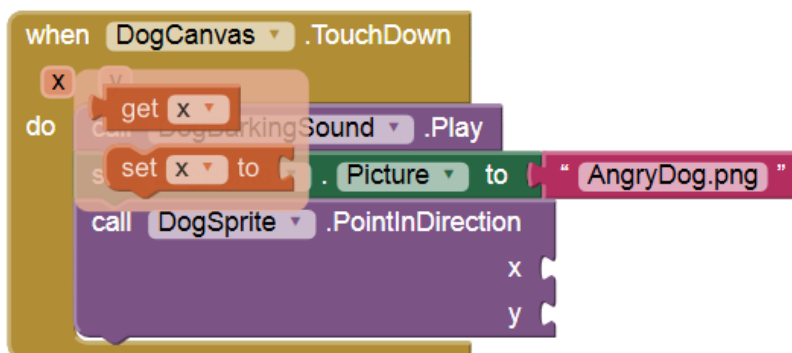
Στο βήμα αυτό θα χρησιμοποιήσουμε τις συντεταγμένες x και y όπου έγινε η επαφή, για να καθορίσουμε προς τα που πρέπει να στραφεί ο σκύλος.

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



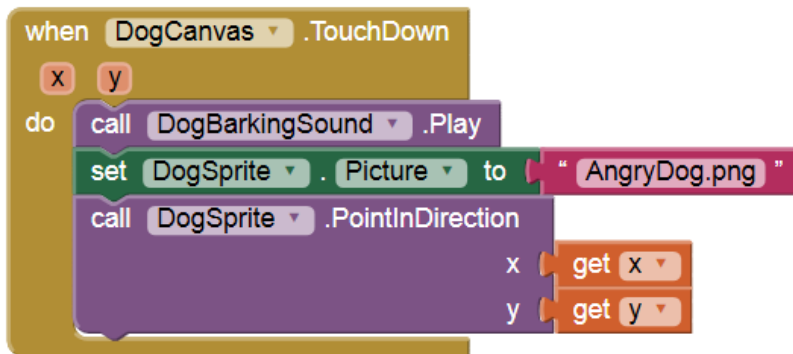
Εικόνα 39 - Προσθήκη της εντολής για στροφή της φιγούρας προς κάποια κατεύθυνση

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...



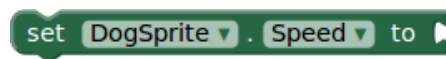
Εικόνα 40 - Εμφάνιση επιπλέον πλακιδίων μέσω των ετικετών

...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**.



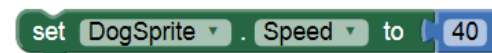
Εικόνα 42 - Εντολή ορισμού ταχύτητας της φιγούρας

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



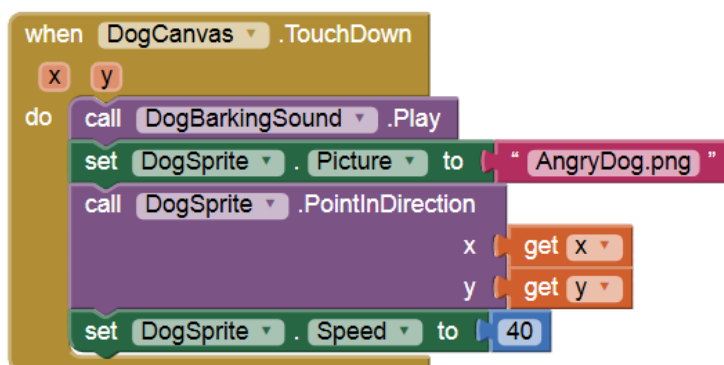
Εικόνα 43 - Το πλακίδιο εισαγωγής αριθμού

...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

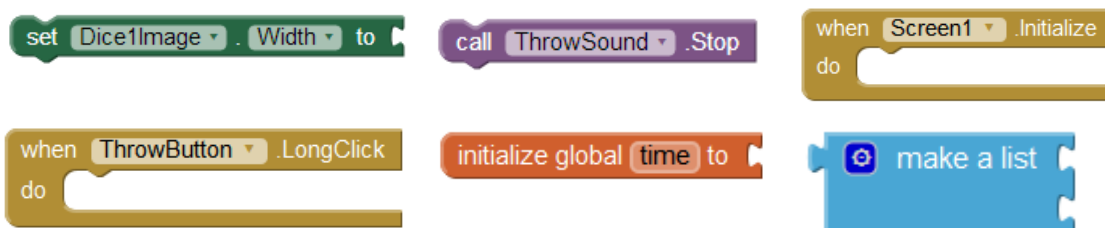
1. Επισκεφθείτε το Google Play Store και συγκεκριμένα τη διεύθυνση:

<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 στρίψιμο 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 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.

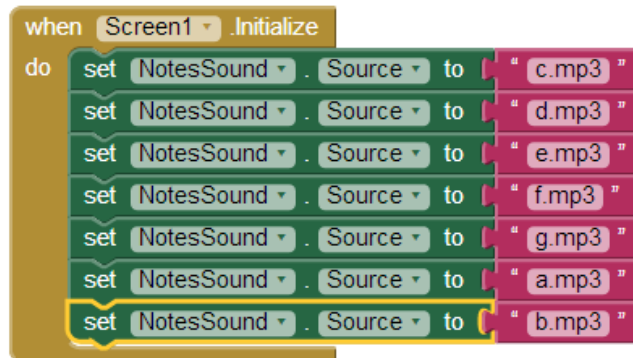
Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <http://tiny.cc/aifiles> και το φάκελο Ριάνο.



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.



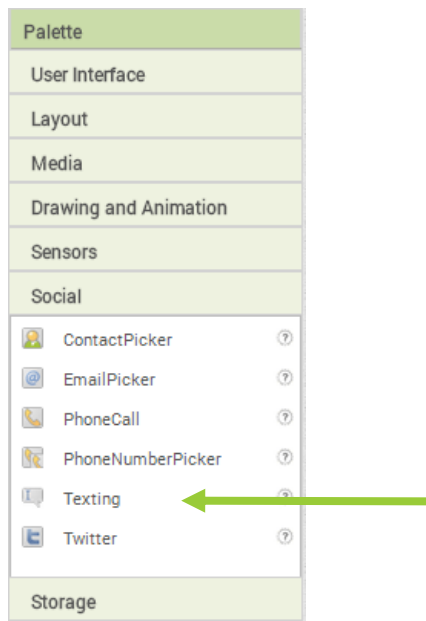
Εικόνα 51 - Μια εκδοχή της εφαρμογής Κουμπί Πανικού

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.

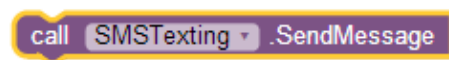
Για την αποστολή του SMS θα χρειαστείτε το αντικείμενο **Texting** που βρίσκεται στην ομάδα **Social**.



Εικόνα 52 - Ομάδα εντολών Social

Για να καθορίσετε το κείμενο του μηνύματος και τον τηλεφωνικό αριθμό που 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. 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.



# App Inventor



## ΜΑΘΗΜΑ 2

### 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 (Αν).
- 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.

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

#### Βήμα 1 Άνοιγμα του έργου «Ζάρια»

Αφού μεταβείτε στη σελίδα του [App Inventor](#) και συνδεθείτε με τα στοιχεία του λογαριασμού σας, ανοίξτε το έργο με τα ζάρια που είχατε δημιουργήσει στο προηγούμενο μάθημα.

#### Βήμα 2 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 <b>ομάδα</b>	μεταφέρουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
-------------------------	--------------------------------------	--------------------------------	--------------------------------------

UserInterface

Label

ResultLabel

Font Size : 40

Text Alignment : center

Text :

Width : Fill Parent

### Βήμα 3 Προσθήκη ετικέτας για το σκορ

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



από την  
**ομάδα**

UserInterface

μεταφέρουμε το  
**αντικείμενο**

Label

του δίνουμε το  
**όνομα**

ScoreLabel

μεταβάλλουμε τις  
**ιδιότητες**

Font Size : 20

TextColor : White

Text : Σκορ : 0

Width : Fill Parent

Συγγραφή Προγράμματος

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.

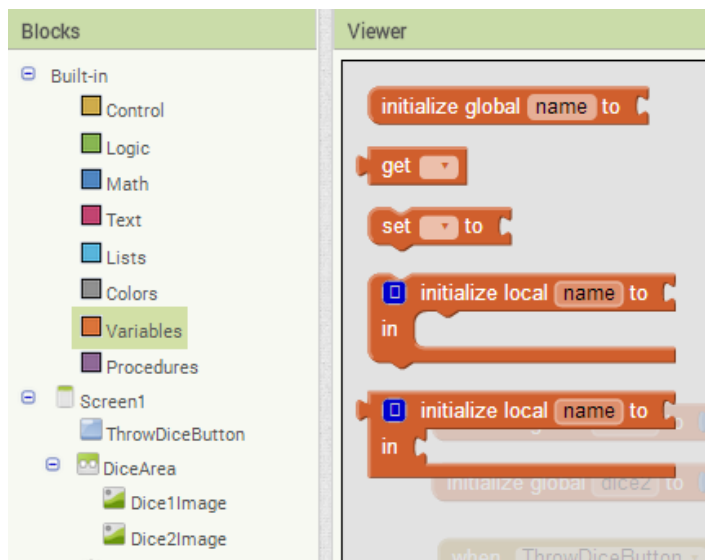


Εικόνα 54 - Η έννοια της μεταβλητής

## Βήμα 4 Δημιουργία μεταβλητών για τα ζάρια

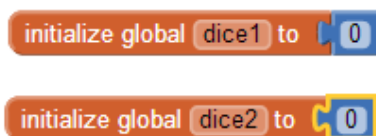
Στην εφαρμογή μας θα χρειαστούμε δύο μεταβλητές, κάθε μια για να αποθηκεύσει τον αριθμό του αντίστοιχου ζαριού.

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.



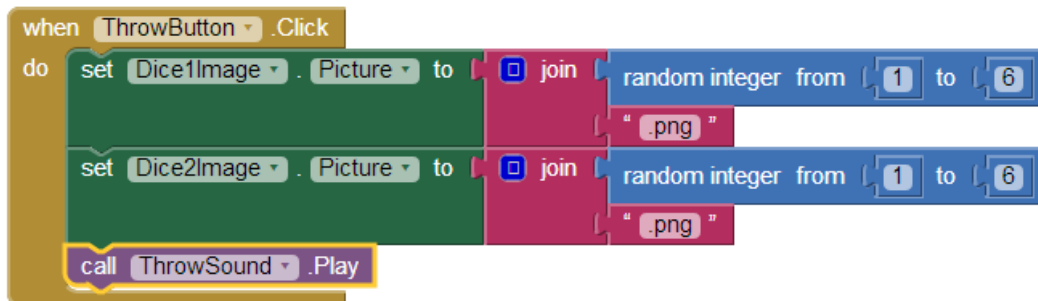
Εικόνα 56 - Δίνοντας αρχικές τιμές σε μεταβλητές



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.

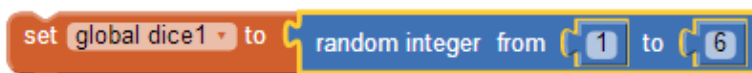
## Βήμα 5 Ανάθεση αριθμών ζαριών στις μεταβλητές

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.



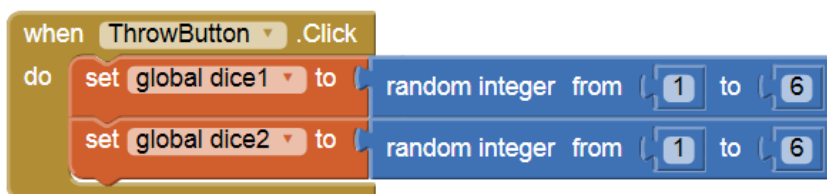
Εικόνα 58 - Ανάθεση τιμής σε μεταβλητή



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

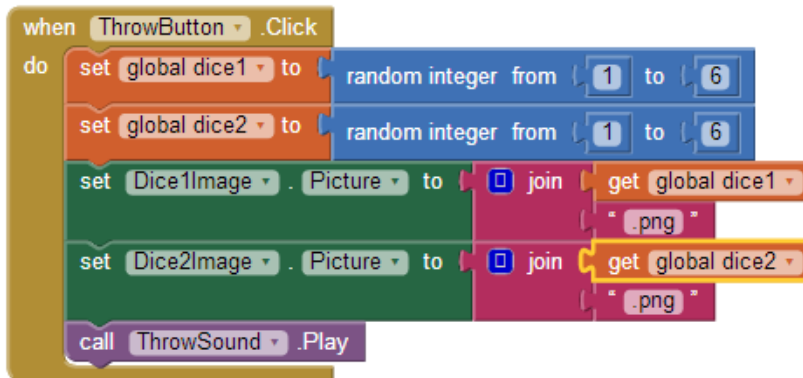
Τέλος, στην εντολή που θέτει την κατάλληλη εικόνα του ζαριού `set Dice1Image.Picture to...` θα αντικαταστήσουμε στην εντολή `join` τον τυχαίο αριθμό, με την τιμή της κάθε μεταβλητής.

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.



Εικόνα 61 - Η εφαρμογή Ζάρια μετά την προσθήκη μεταβλητών

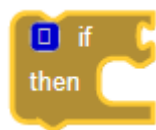
## Βήμα 6 Έλεγχος για εξάρεις

### Η ΕΝΤΟΛΗ IF (ΑΝ)

Για να δώσουμε τη δυνατότητα στην εφαρμογή μας να ανιχνεύει το αν ο χρήστης έφερε εξάρεις, 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**.

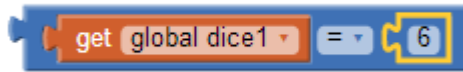


Εικόνα 62 - Η εντολή If

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.

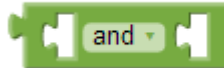
### Η ΛΟΓΙΚΗ ΠΡΟΤΑΣΗ

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

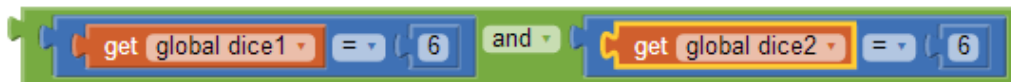
Το ίδιο ακριβώς θέλουμε να εξετάσουμε και για τη δεύτερη μεταβλητή της εφαρμογής dice2. Θέλουμε όμως να ενεργοποιηθούν οι εντολές κάτω από την εντολή Αν, εφόσον ισχύουν ΚΑΙ οι δύο προτάσεις που εξετάζουμε. Στο σημείο αυτό θα χρειαστεί να ενώσουμε τις δύο προτάσεις με το λογικό τελεστή and (και).



Εικόνα 64 - Ο λογικός τελεστής AND

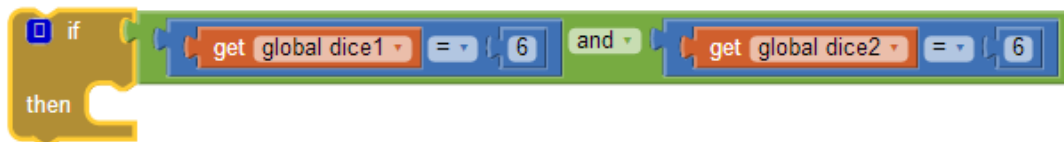


Ο τελεστής AND ενώνει δύο προτάσεις σε μια, που το αποτέλεσμα της είναι Αληθές μόνο αν ισχύουν ΚΑΙ οι δύο αρχικές προτάσεις.



Εικόνα 65 - Ένωση προτάσεων με το AND

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

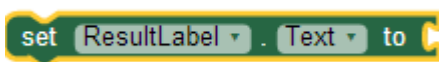


Εικόνα 66 - Η εντολή if με ενσωματωμένη τη λογική πρόταση

## ΟΙ ΕΝΕΡΓΕΙΕΣ ΚΑΤΩ ΑΠΟ ΤΗΝ ΕΝΤΟΛΗ IF

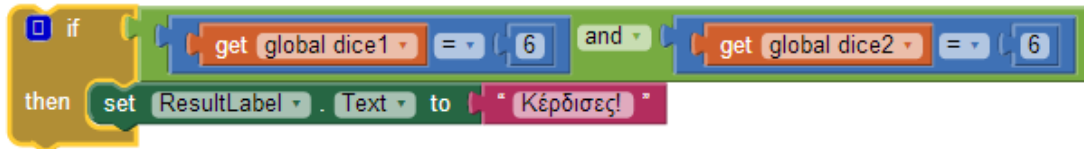
Στην περίπτωση που ο χρήστης της εφαρμογής φέρει εξάρεις θέλουμε να εμφανίζεται το μήνυμα «Κέρδισες!» στην ετικέτα που τοποθετήσαμε στο γραφικό περιβάλλον της εφαρμογής.

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.



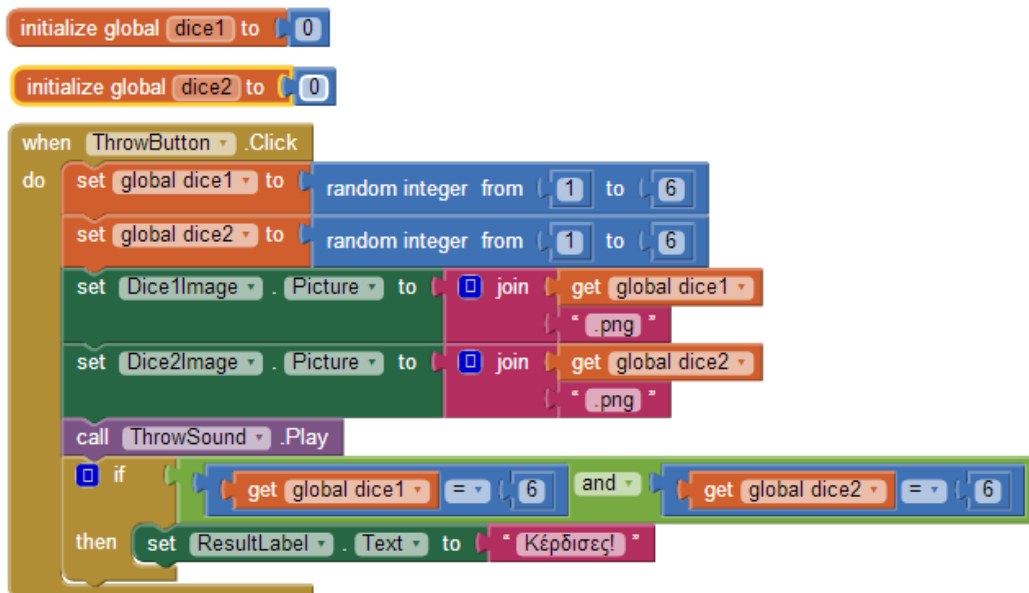
Εικόνα 68 - Η τελική μορφή της εντολής if



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 την ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

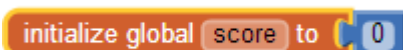


Εικόνα 69 - Η εφαρμογή Ζάρια με έλεγχο για τις εξάρες

## Βήμα 7 Προσθήκη σκορ στην εφαρμογή

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

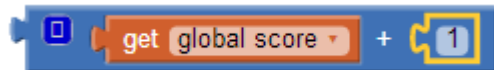
Για το σκοπό αυτό θα χρειαστούμε ακόμα μια μεταβλητή. Ας την ονομάσουμε score και ας της δώσουμε σαν αρχική τιμή το 0.



Εικόνα 70 - Αρχικοποίηση μεταβλητής score



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.



Εικόνα 72 - Ανάθεση νέας τιμής στην μεταβλητή score



Αύξησε την τιμή της μεταβλητής score κατά 1.

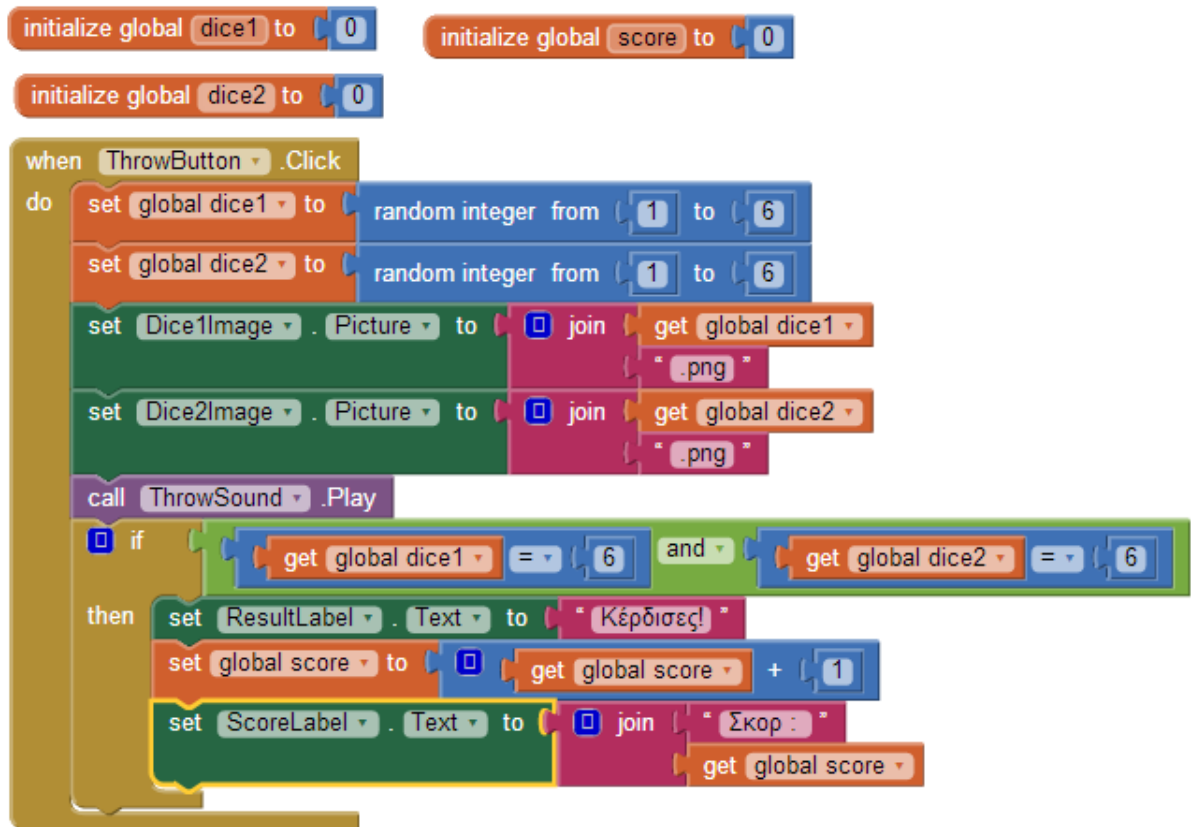
Τέλος τοποθετούμε την εντολή κάτω από την εντολή if.

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.



Εικόνα 73 - Αλλάζοντας το κείμενο της ετικέτας για το σκορ

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



Εικόνα 74 - Η εφαρμογή ζάρια με σκορ



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

## ΕΠΕΚΤΑΣΕΙΣ

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.

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Άνοιγμα υπάρχοντος project

Ανοίγουμε το ήδη υπάρχον project με ονομασία GuardDog.

### Βήμα 2 Αντίδραση του σκύλου στο σήκωμα της συσκευής

Οι φορητές ηλεκτρονικές συσκευές διαθέτουν πλήθος αισθητήρων, όπως τοποθεσίας (GPS), προσανατολισμού (Orientation), επιτάχυνσης (Acceleration) 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 (accelerometer) 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  
**ομάδα**

Sensors

μεταφέρουμε το  
**αντικείμενο**

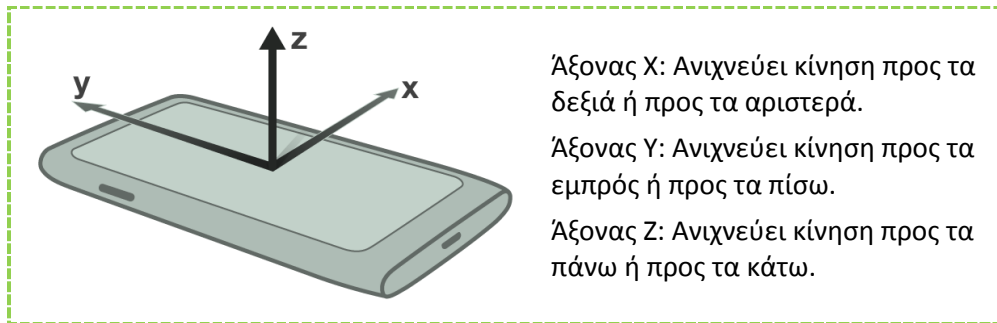
AccelerometerSensor

του δίνουμε το  
**όνομα**

AccSensor

μεταβάλλουμε τις  
**ιδιότητες**

Ο συγκεκριμένος αισθητήρας ανιχνεύει το κούνημα της συσκευής, μετρώντας παράλληλα (κατά προσέγγιση) την επιτάχυνση της βαρύτητας ( $m/s^2$ ) 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 <b>συρτάρι</b>	επιλέγουμε την <b>εντολή</b>	η οποία σημαίνει
AcceleratorSensor	when AccSensor AccelerationChanged do	όταν ανιχνευτεί οποιαδήποτε αλλαγή 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:

```

when AccSensor .AccelerationChanged
  xAccel yAccel zAccel
do
  if get zAccel > 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 Ο σκύλος 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 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 TimeInterval (σε msec) τότε ενεργοποιείται ένα γεγονός.




από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
User Interface	Clock	DogCalmClock	TimerEnabled: No TimeInterval: 3000

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.

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 <b>συρτάρι</b>	επιλέγουμε την <b>εντολή</b>	η οποία σημαίνει
DogCalmClock	set DogCalmClock TimerEnabled to	Ενεργοποίησε ή απενεργοποίησε το χρονόμετρο
Logic	true ή false	Αληθές ή Ψευδές

Έτσι, επεκτείνουμε τη λειτουργία ελέγχου της τιμής του επιταχυνσιομέτρου:

```

when AccSensor .AccelerationChanged
  xAccel yAccel zAccel
do
  if get zAccel > 11
  then
    call DogBarkingSound .Play
    set DogSprite .Picture to "AngryDog.png"
    set DogCalmClock .TimerEnabled to true
    
```



Όταν ανιχνεύσεις «σήκωμα» της συσκευής, μεταξύ άλλων, ενεργοποίησε το χρονόμετρο.

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.

```

when DogCalmClock .Timer
do
  set DogSprite .Picture to "SleepyDog.png"
  set DogCalmClock .TimerEnabled to false
    
```



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.

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

- 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 [App Inventor](#) 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 <http://tiny.cc/aifiles> και το φάκελο CoinFlip. Θα χρειαστείτε 2 εικόνες για τις 2 όψεις του νομίσματος και τους ήχους για το στρίψιμο και την επιτυχή πρόβλεψη.

### Βήμα 2 Προσθήκη κουμπιών επιλογής στην εφαρμογή

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**

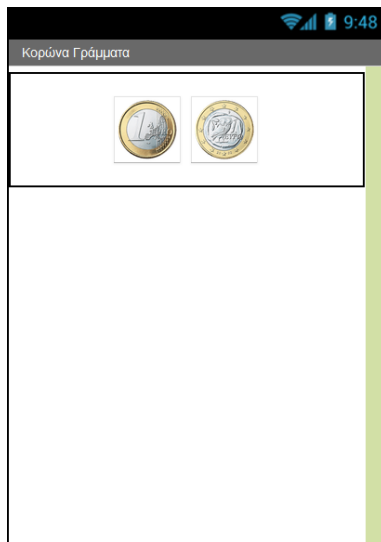
Layout

HorizontalArrangement

CoinArea

Height : 100 pixels

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 κουμπιού για 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	Image : 1.jpg Visible : hidden Width : 150 pixels Height : 150 pixels
Media	Player	AppPlayer	



## Βήμα 4 Προσθήκη ετικέτας με μήνυμα επιτυχίας

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



από την  
**ομάδα**

UserInterface

μεταφέρουμε το  
**αντικείμενο**

Label

του δίνουμε το  
**όνομα**

WinMessageLabel

μεταβάλλουμε τις  
**ιδιότητες**

Font Size : 40

Text Alignment : center

Text :

TextColor : Red

Width : Fill Parent

## Βήμα 5 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 γίνεται ορατό.
- 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 γίνεται ορατό.
- 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 [Παράρτημα με τις Λύσεις των Ασκήσεων](#).

## Βήμα 6 Πέρα από 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.

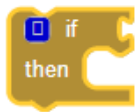
### Η ΕΝΤΟΛΗ ELSE (ΑΛΛΙΩΣ)



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.

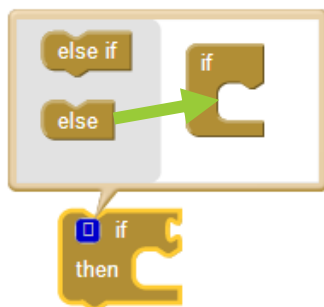
Για να το πετύχουμε θα πρέπει να επεκτείνουμε την εντολή if προσθέτοντας της την εντολή **else**.

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.

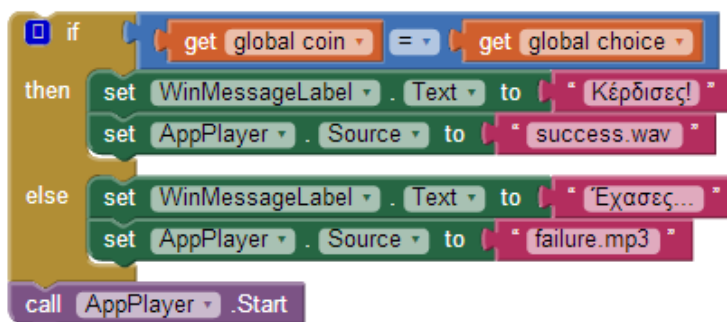


Εικόνα 77 - Κουμπώνοντας το else κάτω από το if

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<sup>ο</sup> ρολόι θα καθορίζει κάθε πότε θα γίνεται η εναλλαγή της εικόνας του κουμπιού, για παράδειγμα κάθε 50 χιλιοστά του δευτερολέπτου.

### Designer Γραφικό Περιβάλλον Εφαρμογής



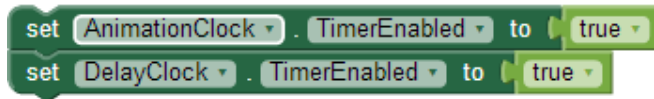
από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
UserInterface	Clock	DelayClock	TimerEnabled : OXI TimerInterval : 1000
UserInterface	Clock	AnimationClock	TimerEnabled : OXI TimerInterval : 50

### Συγγραφή Προγράμματος

### Blocks

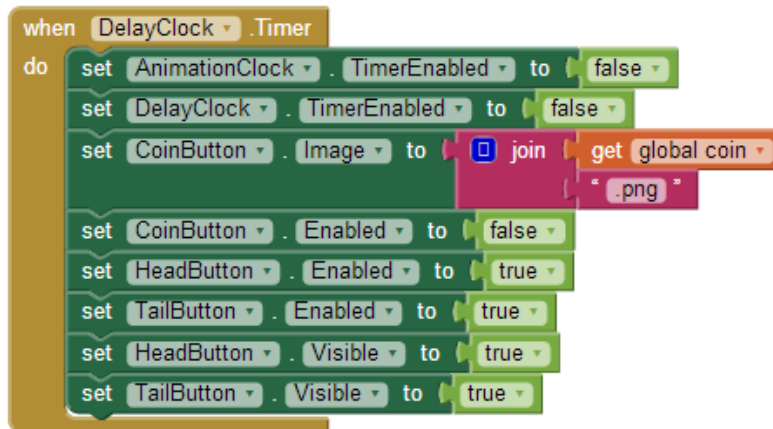
Αρχικά τα δύο ρολόγια είναι απενεργοποιημένα. Η ενεργοποίησή τους θα γίνεται κάθε φορά που ο παίκτης θα αγγίζει το κουμπί της ρίψης του νομίσματος. Όταν τελειώσει το animation μετά το πέρασμα 1 δευτερολέπτου τα ρολόγια θα απενεργοποιούνται ξανά.

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



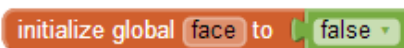
Εικόνα 79 - Εντολές ενεργοποίησης των ρολογιών

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 κουμπιού, 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.

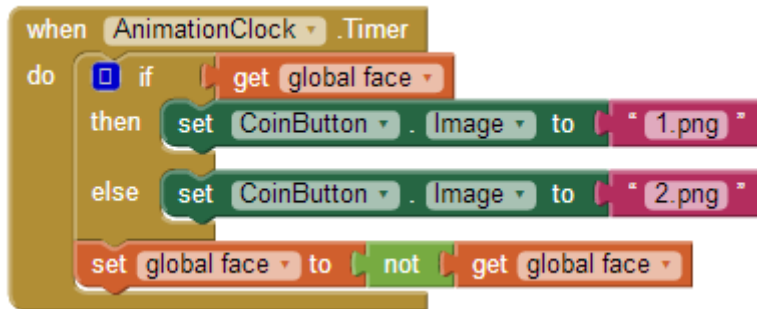


Εικόνα 81 - Αρχικοποίηση της μεταβλητής face

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 :

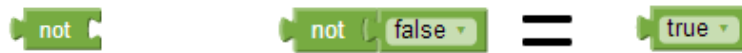


Εικόνα 82 - Η πυροδότηση του ρολογιού για το Animation

## Ο ΛΟΓΙΚΟΣ ΤΕΛΕΣΤΗΣ NOT (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.



Εικόνα 83 - Ο λογικός τελεστής not

Εικόνα 84 - Η λειτουργία του 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 παίξει 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 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.

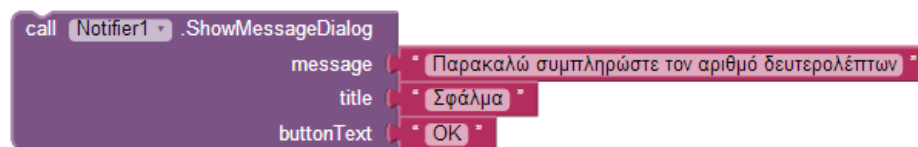


Θα χρειαστείτε το αντικείμενο Clock για να μετρά το πέρασμα 1 δευτερολέπτου.

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



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 50

when Button1.Click
do
  if (get global counter < random integer from 1 to 100)
  then
    set global counter to (get global counter + 1)
  else
    set global counter to (get global counter - 1)

```

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.

Εκδοχή 1<sup>η</sup> με πεδίο κειμένου



Εκδοχή 2<sup>η</sup> με μεταβολέα



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.



Μπορείτε να δείτε τις εντολές όλων των παραπάνω εφαρμογών στην ενότητα «[Λύσεις των Ασκήσεων](#)».

## ΠΗΓΕΣ

Το πολυμεσικό υλικό που χρησιμοποιείται στο έγγραφο και στις αναφερόμενες εφαρμογές διατίθεται δωρεάν από τους παρακάτω ιστότοπους:

<http://www.clker.com>

<http://www.freevector.com>

<http://soundbible.com>

<http://www.freesound.org>

<http://soundfxcenter.com>

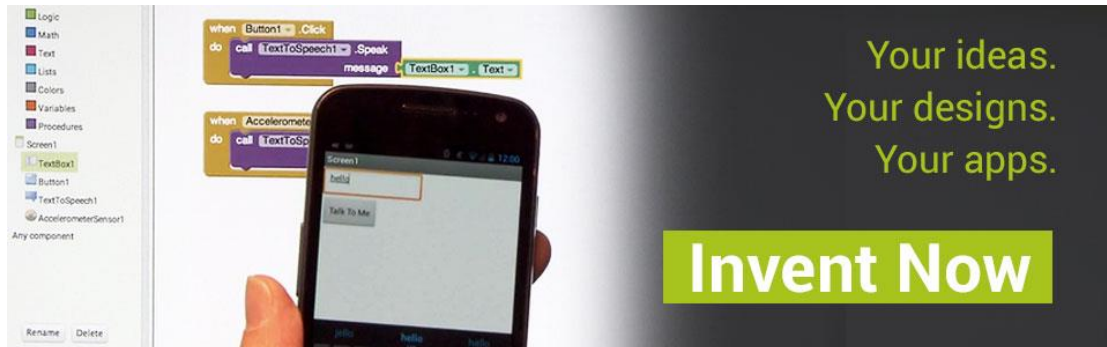
Η εικόνα του εξωφύλλου βασίζεται σε δημιουργία του Rich Dellinger: <http://richd.com>

Η εικόνα στην αρχή κάθε κεφαλαίου βασίζεται σε δημιουργία του Jesse Potter: <http://fetuscakemix.deviantart.com>

Η δραστηριότητα GuardDog βασίζεται στο Φύλλο Εργασίας: *GuardDog - Προγραμματισμός Εφαρμογής με το AppInventor*. Γιώργος Χατζηνικολάκης, Γιώργος Μπουκέας. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2013.

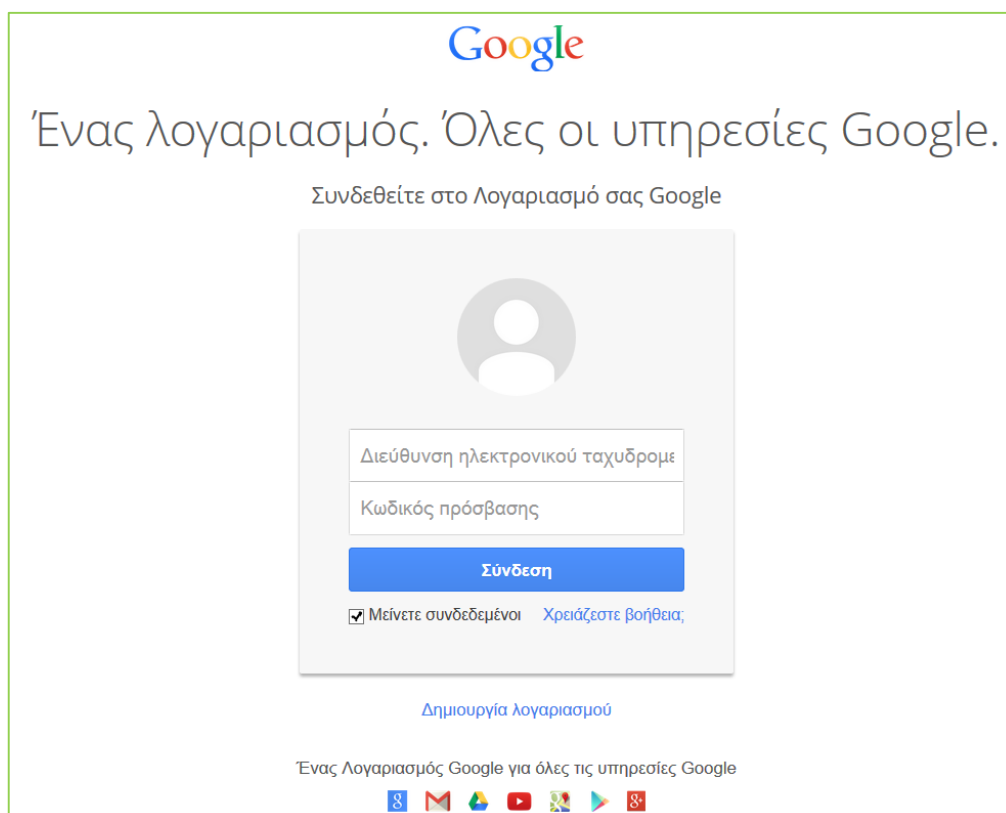
## ΠΑΡΑΡΤΗΜΑ – ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ ΓΙΑ ΤΗ ΧΡΗΣΗ ΤΟΥ APP INVENTOR

Η επίσημη ιστοσελίδα του App Inventor είναι η <http://appinventor.mit.edu> 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.



### Βήμα 1 Δημιουργία λογαριασμού Google

Το 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.



**Όνομα**

Όνομα  Επώνυμο

**Επιλέξτε το όνομα χρήστη σας**

@gmail.com

Προτιμώ να χρησιμοποιήσω την τρέχουσα διεύθυνσή μου ηλεκτρονικού ταχυδρομείου

**Δημιουργία κωδικού πρόσβασης**

**Επιβεβαιώστε τον κωδικό πρόσβασης σας**

**Γενέθλια**

Ημέρα  Μήνας  Έτος

**Φύλο**

Είμαι...

**Κινητό τηλέφωνο**

+30

**Η τρέχουσα διεύθυνσή σας ηλεκτρονικού ταχυδρομείου**

**Αποδείξτε ότι δεν είστε ρομπότ**

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!

## Βήμα 2 Εγκατάσταση εφαρμογής 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 **οποιασδήποτε** από τις δύο παρακάτω δωρεάν εφαρμογές στο Play Store:

[BarCode Scanner](#)  
από την ZXing Team



[QR Code Reader](#)  
από την Scan, Inc.



## Βήμα 3 Εγκατάσταση του MIT 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    Test it in real-time on 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. [οδηγίες](#)). 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 ([AI Setup Installer](#)).

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 [οδηγίες](#)).



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 [Play Store](#)

QR code:



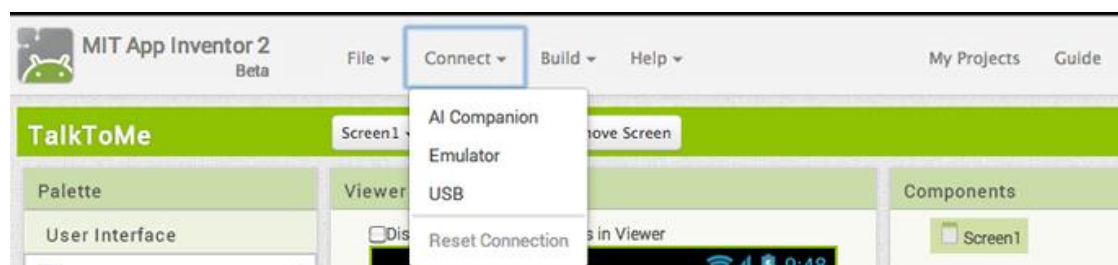
είτε από τη σελίδα του AI ([.apk](#))

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 --> AI Companion ...



... οπότε και εμφανίζεται ένα παράθυρο με έναν QR κώδικα. Ανοίγετε τώρα την εφαρμογή MIT AI2 Companion στη συσκευή σας και πατώντας το κουμπί “Scan QR Code” σαρώνετε τον κώδικα (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.

## ΠΑΡΑΡΤΗΜΑ – ΠΑΚΕΤΑΡΙΣΜΑ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΔΙΑΜΟΙΡΑΣΜΟΣ ΤΗΣ

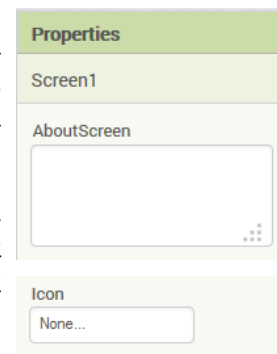
Με τον προαναφερόμενο τρόπο μπορούμε μεν να δοκιμάσουμε την εφαρμογή που αναπτύσσουμε εύκολα και άμεσα, αλλά αυτό δεν αρκεί, αν θέλουμε η εφαρμογή να εγκατασταθεί μόνιμα στη συσκευή μας ή να την μοιραστούμε με άλλους. Απαιτείται λοιπόν να την «πακετάρουμε», δημιουργώντας το σχετικό αρχείο εγκατάστασής της (.apk).

### Βήμα 1 Προσθήκη εικονιδίου και 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.

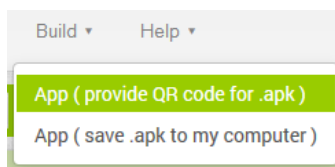
Η πιο απλή λύση είναι να επιλέξουμε (εφόσον υπάρχει) μια χαρακτηριστική εικόνα από αυτές που χρησιμοποιούνται ήδη στην εφαρμογή μας και να την θέσουμε ως Icon της Screen1 από το περιβάλλον σχεδίασης Designer.

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.



### Βήμα 2 Πακετάρισμα της 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: <http://appinventor.mit.edu/explore/ai2/google-play.html>

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*.

