



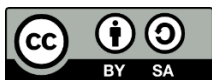
# Προγραμματισμός σε App Inventor

Β' μέρος



1η έκδοση

Μάρτιος 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 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.

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 3

<b>Στόχοι .....</b>	<b>6</b>
<b>Δραστηριότητα : Γκαλερί φωτογραφιών .....</b>	<b>7</b>
Περιγραφή εφαρμογής .....	7
Βασικές έννοιες.....	7
Προγραμματισμός με υποπρογράμματα – Η διαδικασία.....	10
Κλήση υποπρογραμμάτων .....	12
Υποπρογράμματα με παραμέτρους.....	14
Επεκτάσεις .....	17
<b>Δραστηριότητα : Αριθμομηχανή .....</b>	<b>18</b>
Περιγραφή εφαρμογής .....	18
Βασικές έννοιες.....	18
Επεκτάσεις .....	23
<b>Δραστηριότητα : Ζωγραφική με τα δάκτυλα .....</b>	<b>24</b>
Περιγραφή εφαρμογής .....	24
Βασικές έννοιες.....	24
Επεκτάσεις .....	35
<b>Δραστηριότητα : Πυξίδα .....</b>	<b>37</b>
Περιγραφή εφαρμογής .....	37
Βασικές έννοιες.....	37
Επεκτάσεις .....	40
<b>Δραστηριότητα : Πιάσε τη σημαία .....</b>	<b>41</b>
Περιγραφή εφαρμογής .....	41
Βασικές έννοιες.....	41
Επεκτάσεις .....	46
<b>Δραστηριότητες για το σπίτι .....</b>	<b>47</b>
<b>Παράρτημα – Λύσεις των ασκήσεων .....</b>	<b>54</b>
Photogallery (Επεκτάσεις).....	54
Αριθμομηχανή – Διαδικασία που διαβάζει τον αριθμό.....	54
Αριθμομηχανή – Διαδικασίες καθαρισμού και επιλογής αριθμητικής πράξης .....	55
Αριθμομηχανή – Διαδικασία υπολογισμού αποτελέσματος .....	56
FINGERPainting (Επεκτάσεις) .....	57
Πυξίδα (Επέκταση – Ηχητική ειδοποίηση για το Βορρά) .....	57
Πιάσε τη σημαία - Επεκτάσεις .....	57
MoleMash .....	59
Pong .....	60
Παιχνίδι με ζάρια (DiceGame).....	61
Μαθηματικά για παιδιά (Με πεδίο κειμένου).....	62
Μαθηματικά για παιδιά (Παραλλαγή με κουμπιά) .....	63

## Μάθημα 4

<b>Στόχοι .....</b>	<b>65</b>
<b>Δραστηριότητα : Μαθηματικά για παιδιά (Επέκταση).....</b>	<b>66</b>
Περιγραφή εφαρμογής .....	66
Βασικές έννοιες.....	66
Η εντολή επανάληψης WHILE .....	67
Επεκτάσεις .....	69
<b>Δραστηριότητα : Random Drops .....</b>	<b>70</b>
Περιγραφή εφαρμογής .....	70
Βασικές έννοιες.....	70
Υποπρογράμματα που επιστρέφουν τιμή.....	72
Η εντολή επανάληψης FOR.....	75
<b>Δραστηριότητα : Λεξικό.....</b>	<b>77</b>
Περιγραφή εφαρμογής .....	77
Βασικές έννοιες.....	77
Λίστες (list) .....	81
<b>Δραστηριότητα : Where is my bike? .....</b>	<b>92</b>
Περιγραφή εφαρμογής .....	92
Βασικές έννοιες.....	92
Ο εκκινητής δραστηριοτήτων (Activity Starter) .....	96
Επεκτάσεις .....	99
<b>Δραστηριότητα : Μπάλες που αναβοσβήνουν (Παιχνίδι Μνήμης) .....</b>	<b>100</b>
Περιγραφή εφαρμογής .....	100
Βασικές έννοιες.....	100
Επεκτάσεις .....	111
<b>Δραστηριότητες για το σπίτι .....</b>	<b>112</b>
<b>Παράρτημα – Λύσεις των ασκήσεων.....</b>	<b>120</b>
Where is my bike? – Επέκταση ελέγχου & αποθήκευσης σημείωσης στη ΒΔ .....	120
Κουμπί πανικού με δυνατότητα επιλογής & αποθήκευσης SMS & αριθμού .....	120
Μπάλες που αναβοσβήνουν (Παιχνίδι Μνήμης) - Επέκταση .....	121
Τυχερά Μπισκότα .....	121
Mix And Match .....	123
Κρυπτογραφία.....	124
HangMan .....	128
<b>Πηγές.....</b>	<b>130</b>

# App Inventor



## ΜΑΘΗΜΑ 3

### ΣΤΟΧΟΙ

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΓΚΑΛΕΡΙ ΦΩΤΟΓΡΑΦΙΩΝ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 1 - Εφαρμογή PhotoGallery

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα PhotoGallery.

## Designer Γραφικό Περιβάλλον Εφαρμογής

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



επιλέγουμε το  
**αντικείμενο**

Screen1

μεταβάλλουμε τις  
**ιδιότητες**

AlignHorizontal: Center

Screen Orientation: Portrait

Scrollable: No

Title: PhotoGallery

## Βήμα 2 Προσθήκη των απαραίτητων αρχείων πολυμέσων

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



ανεβάζουμε το αρχείο με  
**όνομα**

σύντομη  
**περιγραφή**

1.jpg

Εικόνα για προβολή στο άλμπουμ

2.jpg

Εικόνα για προβολή στο άλμπουμ

3.jpg

Εικόνα για προβολή στο άλμπουμ

4.jpg

Εικόνα για προβολή στο άλμπουμ

5.jpg

Εικόνα για προβολή στο άλμπουμ

6.jpg

Εικόνα για προβολή στο άλμπουμ

next.png

Κουμπί επόμενο

previous.png

Κουμπί προηγούμενο


Averil White - Death of the Hero.mp3

Μουσικό κομμάτι εφαρμογής

## Βήμα 3 Προσθήκη αντικειμένων στο γραφικό περιβάλλον

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



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

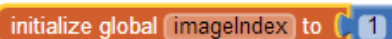
Συγγραφή Προγράμματος

Blocks

#### Βήμα 4 Εναλλαγή των εικόνων του άλμπουμ

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

Ας ονομάσουμε τη μεταβλητή μας `imageIndex` με αρχική τιμή 1.



Εικόνα 2 - Μεταβλητή για την εναλλαγή των εικόνων

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

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα.

```

set global imageIndex to (get global imageIndex + 1)
if (get global imageIndex > 6)
then
  set global imageIndex to 1
set GalleryCanvas.BackgroundImage to (join (get global imageIndex) ".jpg")
set PicNumberLabel.Text to (get global imageIndex)

```

Εικόνα 3 - Εντολές για την εναλλαγή των εικόνων



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

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

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

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



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

## ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΜΕ ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ – Η ΔΙΑΔΙΚΑΣΙΑ

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



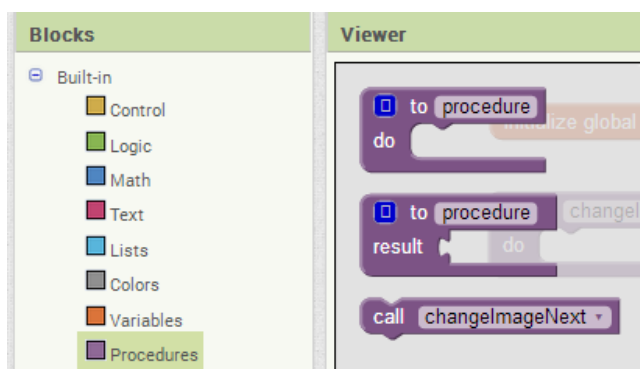
Η διαδικασία είναι ένα υποπρόγραμμα, δηλαδή ένα μικρό και αυτόνομο τμήμα της εφαρμογής στο οποίο τοποθετούμε μια σειρά εντολών. Στην ουσία χρησιμοποιώντας διαδικασίες δημιουργούμε νέες εντολές που μπορούμε στη συνέχεια να καλούμε (χρησιμοποιούμε) από οποιοδήποτε σημείο του προγράμματος.

Η χρήση υποπρογραμμάτων έχει μια σειρά πλεονεκτήματα, όπως:

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

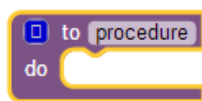
- Μείωση του χρόνου ανάπτυξης και τροποποιήσεων της εφαρμογής, αφού δεν χρειάζεται να επαναλαμβάνουμε όμοια τμήματα εντολών σε διάφορα σημεία της εφαρμογής.
- Ο κώδικας της εφαρμογής είναι πιο ευανάγνωστος και «συμπαγής».

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



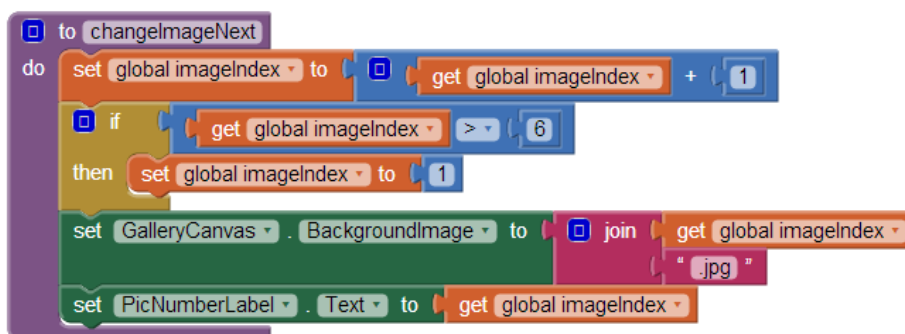
Εικόνα 4 - Ομάδα εντολών Procedures

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



Εικόνα 5 - Εντολή δημιουργίας διαδικασιών

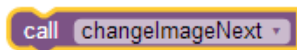
Στη συνέχεια θα μετονομάσουμε τη διαδικασία μας από **procedure** σε **changelImageNext**, ώστε να έχει κάποιο συγκεκριμένο νόημα για εμάς το όνομά της και θα τοποθετήσουμε τις εντολές για την εναλλαγή των εικόνων μέσα σε αυτή. Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα.



Εικόνα 6 - Διαδικασία για τη μετακίνηση στην επόμενη εικόνα

## ΚΛΗΣΗ ΥΠΟΠΡΟΓΡΑΜΜΑΤΩΝ

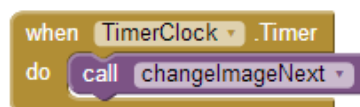
Για να εκτελεστούν οι εντολές που βρίσκονται μέσα σε μια διαδικασία θα πρέπει να την καλέσουμε, όπως συνηθίζεται να λέμε, δηλαδή να την χρησιμοποιήσουμε μέσα σε ένα ή περισσότερα σημεία της εφαρμογής. Η εντολή για την κλήση μιας διαδικασίας που αναπτύσσουμε βρίσκεται και αυτή στην ομάδα Procedures.



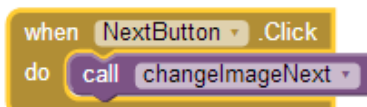
Εικόνα 7 - Εντολή κλήσης της διαδικασίας

### Βήμα 5 Κλήση της διαδικασίας κατά την εκτέλεση γεγονότων

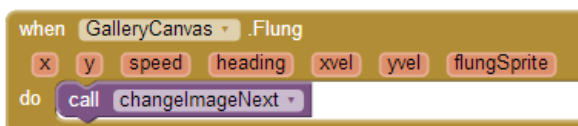
Στο σημείο αυτό το πρόβλημα έχει απλοποιηθεί πολύ, αφού το μόνο που μένει να κάνουμε είναι να καλέσουμε τη διαδικασία μας, με την εκτέλεση των 3 γεγονότων, όπως παρακάτω.



Εικόνα 8 - Πυροδότηση του χρονόμετρου



Εικόνα 9 - Άγγιγμα κουμπιού "Επόμενο"



Εικόνα 10 - Σύρσιμο δακτύλου στον καμβά

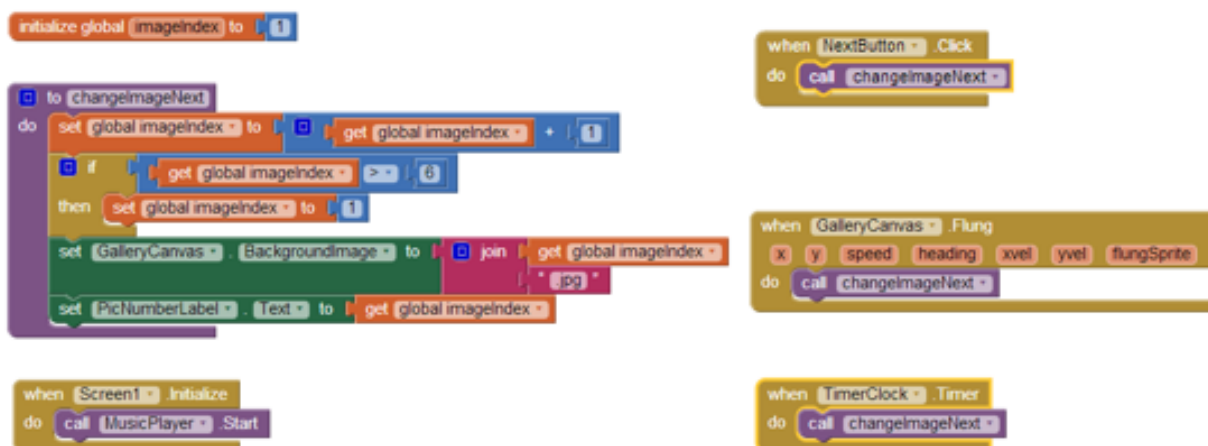
### Βήμα 6 Αναπαραγωγή μουσικής

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



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

Οι εντολές της εφαρμογής φαίνονται όλες μαζί στην παρακάτω εικόνα.



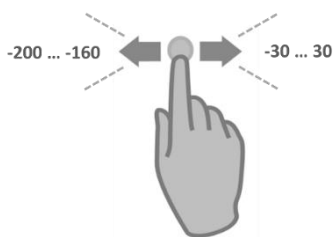
Εικόνα 11 - Αρχική εκδοχή της εφαρμογής PhotoGallery

## Βήμα 7 Μετακίνηση στην προηγούμενη εικόνα

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



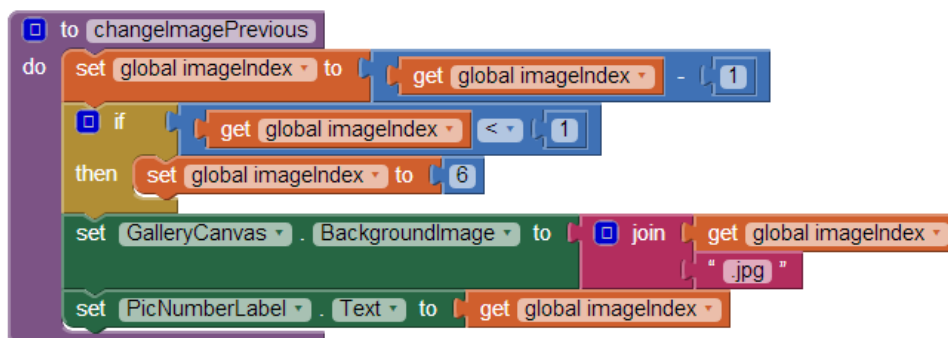
Για να ανιχνεύσετε τη φορά προς την οποία μετακινείται το δάκτυλο του χρήστη πάνω στον καμβά θα χρειαστείτε την ιδιότητα `heading` (κατεύθυνση), την τιμή της οποίας σας παρέχει το γεγονός **when [GalleryCanvas] Flung**.



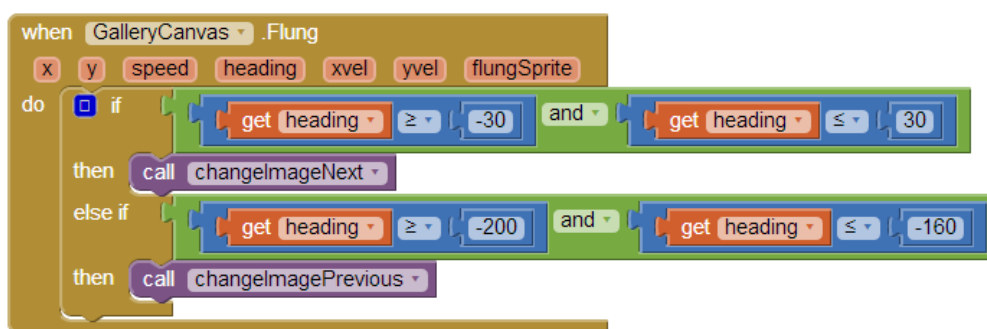
Για να ανιχνεύσετε την κίνηση του δάκτυλου προς τα δεξιά ελέγξτε αν η ιδιότητα `heading` έχει τιμές από -30 μέχρι 30.

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

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



Εικόνα 12 - Διαδικασία για την μετακίνηση σε προηγούμενη εικόνα



Εικόνα 13 - Έλεγχος για κατεύθυνση συρσίματος του δάκτυλου πάνω στον καμβά

## Βήμα 8 Συγχωνεύοντας τις δύο διαδικασίες σε μία

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

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



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

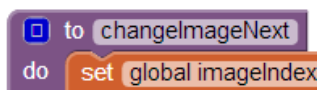
## ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΜΕ ΠΑΡΑΜΕΤΡΟΥΣ



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

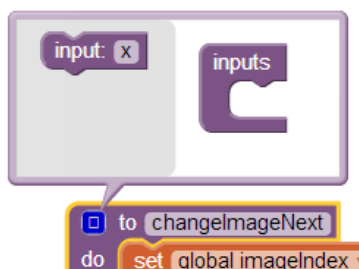
Στην ουσία οι παράμετροι είναι μεταβλητές που μεταφέρουν τιμές από την εφαρμογή προς τη διαδικασία.

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



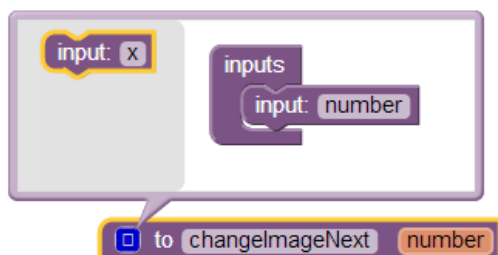
Εικόνα 14 - Προσθέτοντας παραμέτρους σε διαδικασίες

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



Εικόνα 15 - Παράθυρο προσθήκης παραμέτρων

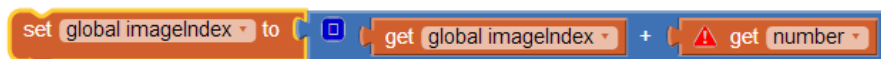
Στη θέση του x βάζουμε το όνομα που θέλουμε να έχει κάθε παράμετρος. Στην εφαρμογή μας θα την ονομάσουμε number.



Εικόνα 16 - Μετονομασία παραμέτρων

Επίσης, θα μετονομάσουμε τη διαδικασία **changelImageNext** σε **changelImage**.

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



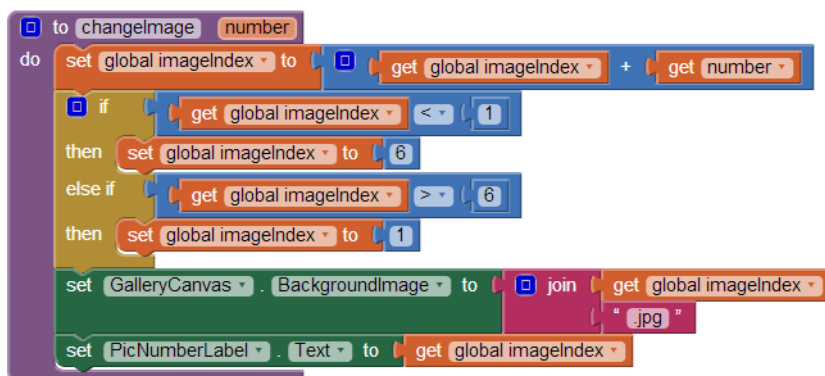
Εικόνα 17 - Χρησιμοποιώντας την παράμετρο

Επιπλέον θα προσθέσουμε στην εντολή if και τους 2 ελέγχους.



Εικόνα 18 - Έλεγχος για άνω και κάτω όριο

Η τελική μορφή της διαδικασίας μας φαίνεται στην παρακάτω εικόνα.

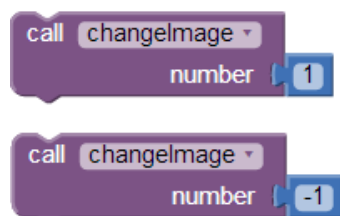


Εικόνα 19 - Τελική μορφή της παραμετρικής διαδικασίας

## Βήμα 9 Αλλάζοντας την κλήση της διαδικασίας

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

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



Εικόνα 20 - Κλήση της παραμετρικής διαδικασίας

Η τελική μορφή της εφαρμογής φαίνεται παρακάτω.



```

initialize global imageIndex to 1

when Screen1.Initialize
do call MediaPlayer.Start

when TimerClock.Timer
do call changelimage
number 1

to changelimage number
do
set global imageIndex to (get global imageIndex + get number)
if (get global imageIndex < 1)
then set global imageIndex to 6
else if (get global imageIndex > 6)
then set global imageIndex to 1
set GalleryCanvas.BackgroundImage to join (get global imageIndex) ".jpg"
set PicNumberLabel.Text to get global imageIndex

when PreviousButton.Click
do
call changelimage
number -1
set TimerClock.TimerEnabled to true
set TimerClock.TimerEnabled to false

when NextButton.Click
do
set TimerClock.TimerEnabled to false
call changelimage
number 1
set TimerClock.TimerEnabled to true

when GalleryCanvas.Flung
do
set TimerClock.TimerEnabled to false
if (get heading >= -30 and get heading <= 30)
then call changelimage
number 1
else if (get heading >= -200 and get heading <= -160)
then call changelimage
number -1
set TimerClock.TimerEnabled to true

```

Εικόνα 21 - Τελική εκδοχή της εφαρμογής PictureGallery

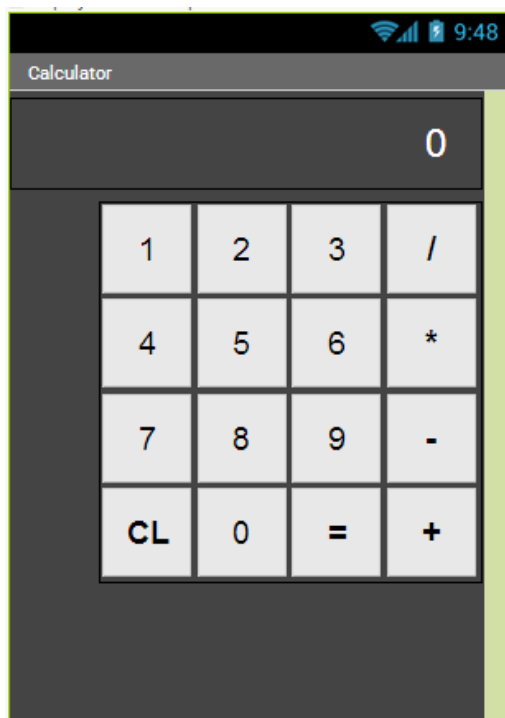
## ΕΠΕΚΤΑΣΕΙΣ

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΑΡΙΘΜΟΜΗΧΑΝΗ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 22 - Η εφαρμογή Αριθμομηχανή

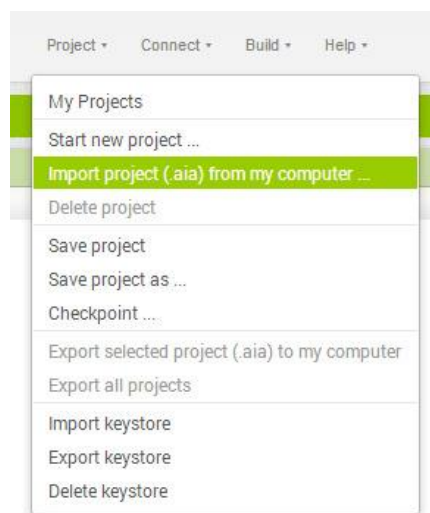
### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

#### Βήμα 1 Εισαγωγή εφαρμογής από αρχείο

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

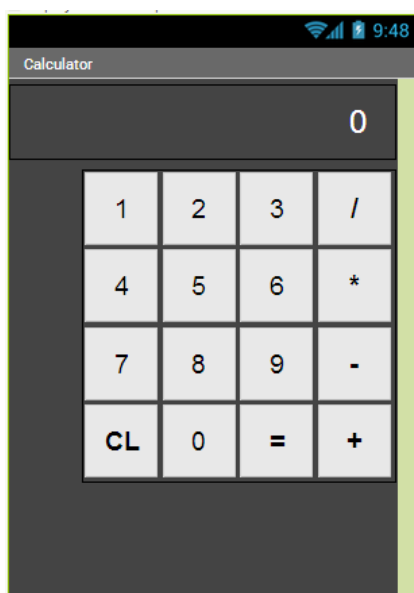
Για να εισάγουμε το αρχείο που περιέχει την μισοτελειωμένη έκδοσή της εφαρμογής θα κατεβάσουμε αρχικά στον υπολογιστή μας το αρχείο Calculator.aia από τη διεύθυνση <http://tiny.cc/aifiles> και το φάκελο Calculator. Στη συνέχεια από την κεντρική σελίδα των projects του App Inventor θα κάνουμε κλικ στο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



Εικόνα 23 - Εισαγωγή έτοιμου έργου στο περιβάλλον του App Inventor

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

## Βήμα 2 Το γραφικό περιβάλλον της εφαρμογής



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

### Βήμα 3 Ο έτοιμος κώδικας της εφαρμογής

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



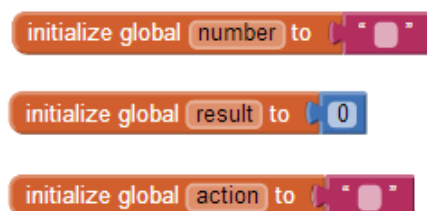
Εικόνα 24 - Ο έτοιμος κώδικας της εφαρμογής Αριθμομηχανή (1)

Επιπλέον υπάρχουν τρεις εντολές αρχικοποίησης μεταβλητών.

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

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

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



Εικόνα 25 - Εικόνα 24 - Ο έτοιμος κώδικας της εφαρμογής Αριθμομηχανή (2)

## Βήμα 4 «Διαβάζοντας» τον αριθμό που πληκτρολογείται

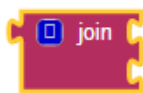
Θα υλοποιήσουμε μια διαδικασία (ονομάστε την **makeNumber**) που θα παίρνει σαν παράμετρο τον αριθμό του ψηφίου που «πατήθηκε» από τον χρήστη και θα το ενώνει με τον υπόλοιπο αριθμό που έχει ήδη πληκτρολογηθεί.

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

- A. να τα ενώνει (δηλαδή να φτιάχνει το 95),
- B. να αποθηκεύει το αποτέλεσμα στην μεταβλητή number και
- Γ. να εμφανίζει στην ετικέτα τον καινούριο αριθμό που σχηματίστηκε



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



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

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το [Παράρτημα – Λύσεις των Ασκήσεων](#) του μαθήματος.



Ώρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

## Βήμα 5 Καθαρίζοντας την αριθμομηχανή

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

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



Ώρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.

## Βήμα 6 Διαβάζοντας την αριθμητική πράξη

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

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

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

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

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

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

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

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το [Παράρτημα – Λύσεις των Ασκήσεων](#) του μαθήματος.

## Βήμα 7 Υπολογίζοντας το αποτέλεσμα

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

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

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

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

Δ. Θα καλεί τη διαδικασία που καθαρίζει την αριθμομηχανή.

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

Προσπαθήστε να ολοκληρώσετε μόνοι σας το παραπάνω βήμα. Σε περίπτωση που δεν τα καταφέρετε μπορείτε να συμβουλευθείτε το [Παράρτημα – Λύσεις των Ασκήσεων](#) του μαθήματος.



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

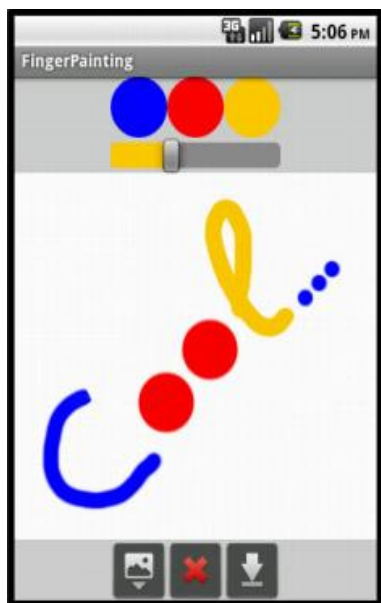
#### ΕΠΕΚΤΑΣΕΙΣ

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΩΓΡΑΦΙΚΗ ΜΕ ΤΑ ΔΑΚΤΥΛΑ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 26 – Στιγμιότυπο της εφαρμογής Finger Painting

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα FingerPainting.

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





επιλέγουμε το  
**αντικείμενο**

μεταβάλλουμε τις  
**ιδιότητες**

Screen1

AlignHorizontal: Center

BackgroundColor: LightGrey

Screen Orientation: Portrait

Scrollable: No

Title: FingerPainting

## Βήμα 2

### Προσθήκη των απαραίτητων αρχείων πολυμέσων

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



ανεβάζουμε το αρχείο με  
**όνομα**

σύνομη  
**περιγραφή**

clear.jpg

Εικόνα για το κουμπί καθαρισμού του καμβά

takepic.png

Εικόνα για το κουμπί λήψης φωτογραφίας

save.png

Εικόνα για το κουμπί αποθήκευσης εικόνας

## Βήμα 3

### Σχέδια επάνω στον καμβά

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

### Designer

#### Γραφικό Περιβάλλον Εφαρμογής

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



από την  
**ομάδα**

μεταφέρουμε το  
**αντικείμενο**

του δίνουμε το  
**όνομα**

μεταβάλλουμε τις  
**ιδιότητες**

Drawing and  
Animation

Canvas

Paper

Paint Color: Blue

Width: Fill Parent

Height: Fill Parent

## Συγγραφή Προγράμματος

## Blocks

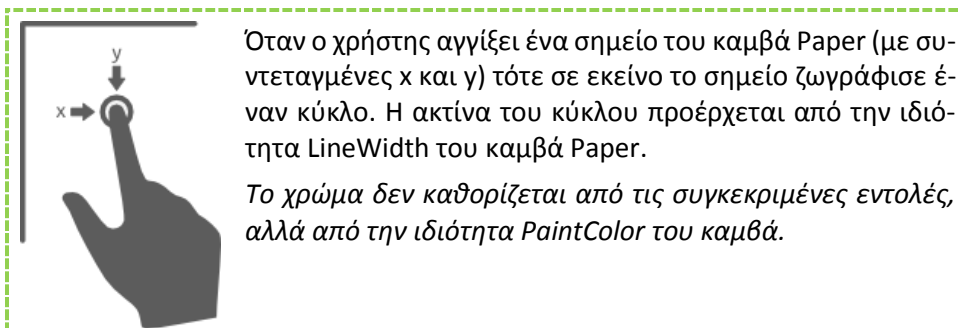
Οι δύο ομάδες εντολών που ακολουθούν ανιχνεύουν πότε ο χρήστης ακούμπησε την οθόνη της συσκευής ή έσυρε το δάχτυλο πάνω σε αυτήν. Στην πρώτη περίπτωση σχεδιάζεται μια τελεία, ενώ στη δεύτερη μια γραμμή.

```

when Paper .Touched
  x y touchedSprite
do call Paper .DrawCircle
  x get x
  y get y
  r Paper . LineWidth

```

Εικόνα 27 - Σχεδιασμός κύκλου στο σημείο επαφής

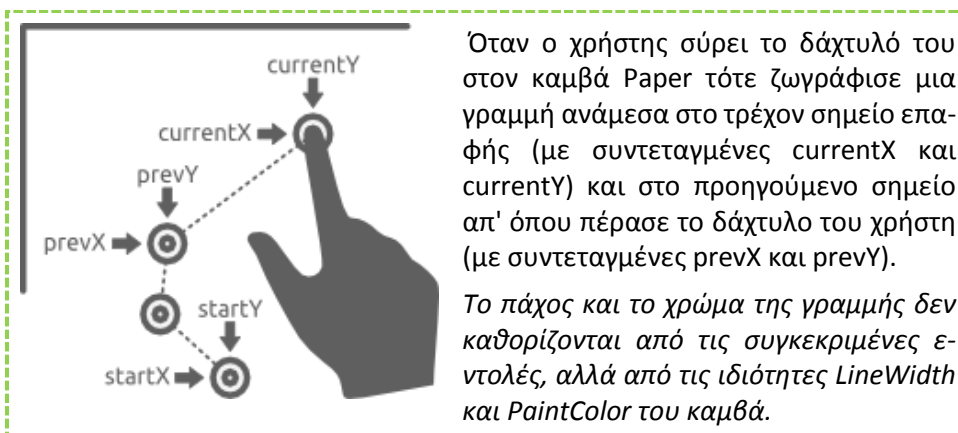


```

when Paper .Dragged
  startX startY prevX prevY currentX currentY
do call Paper .DrawLine
  x1 get prevX
  y1 get prevY
  x2 get currentX
  y2 get currentY

```

Εικόνα 28 - Σχεδιασμός γραμμής με σύριμο



## Βήμα 4 Καθαρισμός του καμβά

Σε αυτό το βήμα θα προσθέσουμε τη δυνατότητα καθαρισμού του σχεδίου μας με ένα κούνημα της συσκευής.

### Designer Γραφικό Περιβάλλον Εφαρμογής

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



από την  
**ομάδα**

Sensors

μεταφέρουμε το  
**αντικείμενο**

Accelerometer Sensor

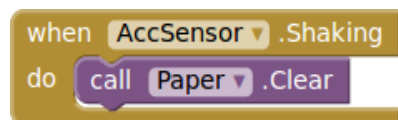
του δίνουμε το  
**όνομα**

AccSensor

### Συγγραφή Προγράμματος

### Blocks

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



Εικόνα 29 - Καθαρισμός καμβά με ταρακούνημα




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

## Βήμα 5 Μια μικρή παλέτα με χρώματα

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

### Designer Γραφικό Περιβάλλον Εφαρμογής

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

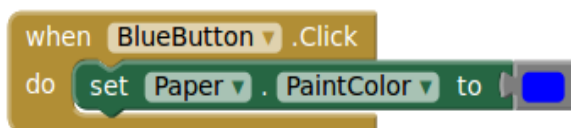
 από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	ColorArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Button	BlueButton	BackgroundColor: Blue Shape: Oval Text: Width: 50 pixels Height: 50 pixels
User Interface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

Συγγραφή Προγράμματος

Blocks

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

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



Εικόνα 30 - Ορισμός του χρώματος βαφής



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

## Βήμα 6 Αλλάζοντας το πάχος της γραμμής

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

### Designer Γραφικό Περιβάλλον Εφαρμογής

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



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

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

### Συγγραφή Προγράμματος

### Blocks

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

```

when WidthSlider .PositionChanged
  thumbPosition
do set Paper . LineWidth to get thumbPosition
  
```

Εικόνα 31 - Ορισμός πάχους γραμμής



Όταν αλλάξει η θέση *thumbPosition* του μεταβολέα *Width* τότε ενημέρωσε ανάλογα και το πάχος των γραμμών του καμβά *Paper* (μέσω της ιδιότητας *LineWidth*).

Όταν το πάχος της γραμμής γίνει μεγάλο τότε οι γραμμές δεν σχεδιάζονται σωστά. Μπορούμε να αντιμετωπίσουμε το πρόβλημα σχεδιάζοντας έναν κύκλο στο τρέχον σημείο επαφής με την οθόνη. Η ακτίνα του κύκλου θα πρέπει να είναι το μισό του πάχους γραμμής. Προσθέτουμε λοιπόν την ακόλουθη εντολή στο whenPaperDragged του Βήματος 3.

```

call Paper .DrawCircle
  x get currentX
  y get currentY
  r Paper . LineWidth / 2
    
```


Εικόνα 32 - Βελτίωση για γραμμές μεγάλου πάχους

## Βήμα 7 Σχέδια πάνω σε φωτογραφίες

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

### Designer Γραφικό Περιβάλλον Εφαρμογής

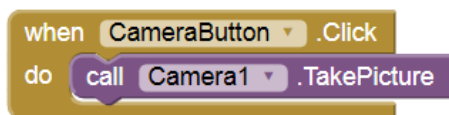
Χρειαζόμαστε ένα κουμπί (Button) με το οποίο θα ενεργοποιούμε την κάμερα της συσκευής, καθώς και ένα αντικείμενο-συστατικό που αντιστοιχεί στην ίδια την κάμερα (Camera). Τοποθετούμε το κουμπί μέσα σε μια νέα οριζόντια τακτοποίηση.

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

## Συγγραφή Προγράμματος

## Blocks

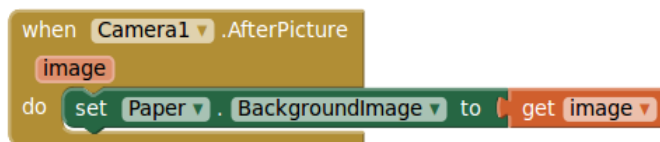
Καλούμε την ενσωματωμένη στο App Inventor λειτουργία που επιτρέπει τη λήψη φωτογραφίας, με το πάτημα του κουμπιού CameraButton.



Εικόνα 33 - Λήψη φωτογραφίας



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



Εικόνα 34 - Ορισμός της φωτογραφίας ως φόντου του καμβά

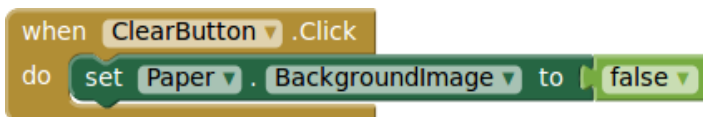


Όταν τραβηχτεί μια φωτογραφία image από την φωτογραφική μηχανή Camera1 τότε θέσε την φωτογραφία αυτή ως φόντο στον καμβά Paper.

## Βήμα 8

## Καθαρισμός του καμβά (συνέχεια)

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



Εικόνα 35 - Καθαρισμός φόντου με πάτημα κουμπιού



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

## Βήμα 9 Αποθήκευση των σχεδίων μας

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



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

### Designer Γραφικό Περιβάλλον Εφαρμογής

Εισάγουμε λοιπόν τα δυο προαναφερθέντα αντικείμενα.

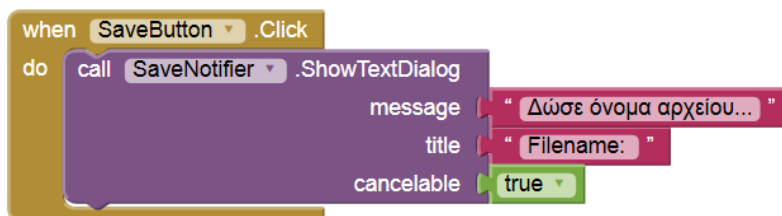


από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
User Interface	Button	SaveButton	Width: 50 Height: 40 Text: Image: save.png
User Interface	Notifier	SaveNotifier	

### Συγγραφή Προγράμματος

### Blocks

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



Εικόνα 36 - Ερώτημα για όνομα αρχείου προς αποθήκευση



Όταν πατηθεί το πλήκτρο SaveButton εμφανίσε ένα παράθυρο διαλόγου με τίτλο «Filename:», κυρίως μήνυμα «Δώσε όνομα αρχείου...», το οποίο μπορεί να ακυρωθεί από το χρήστη, αν το επιθυμεί.



Αμέσως μετά, θα αξιοποιήσουμε την απάντηση που δόθηκε για να αποθηκεύσουμε το σχέδιο.

```

when SaveNotifier .AfterTextInput
  response
do
  call SaveNotifier .ShowAlert
  notice
  call Paper .SaveAs
  fileName
  join
  get response
  ".png"
  
```

Εικόνα 37 - Αποθήκευση της ζωγραφιάς με το όνομα που δόθηκε



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

### ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ ΑΠΟΘΗΚΕΥΣΗΣ ΜΕ ΑΥΤΟΜΑΤΗ ΟΝΟΜΑΣΙΑ ΑΡΧΕΙΟΥ

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

```

when SaveButton .Click
do
  initialize local name to
  call Paper .SaveAs
  fileName
  join
  "fp-"
  call Clock1 .SystemTime
  ".png"
in
  call SaveNotifier .ShowMessageDialog
  message
  join
  "filename: "
  get name
  title
  "Saved Image"
  buttonText
  "OK"
  
```

Εικόνα 38 - Αποθήκευση του σχεδίου με αυτόματη καταχώρηση ονόματος

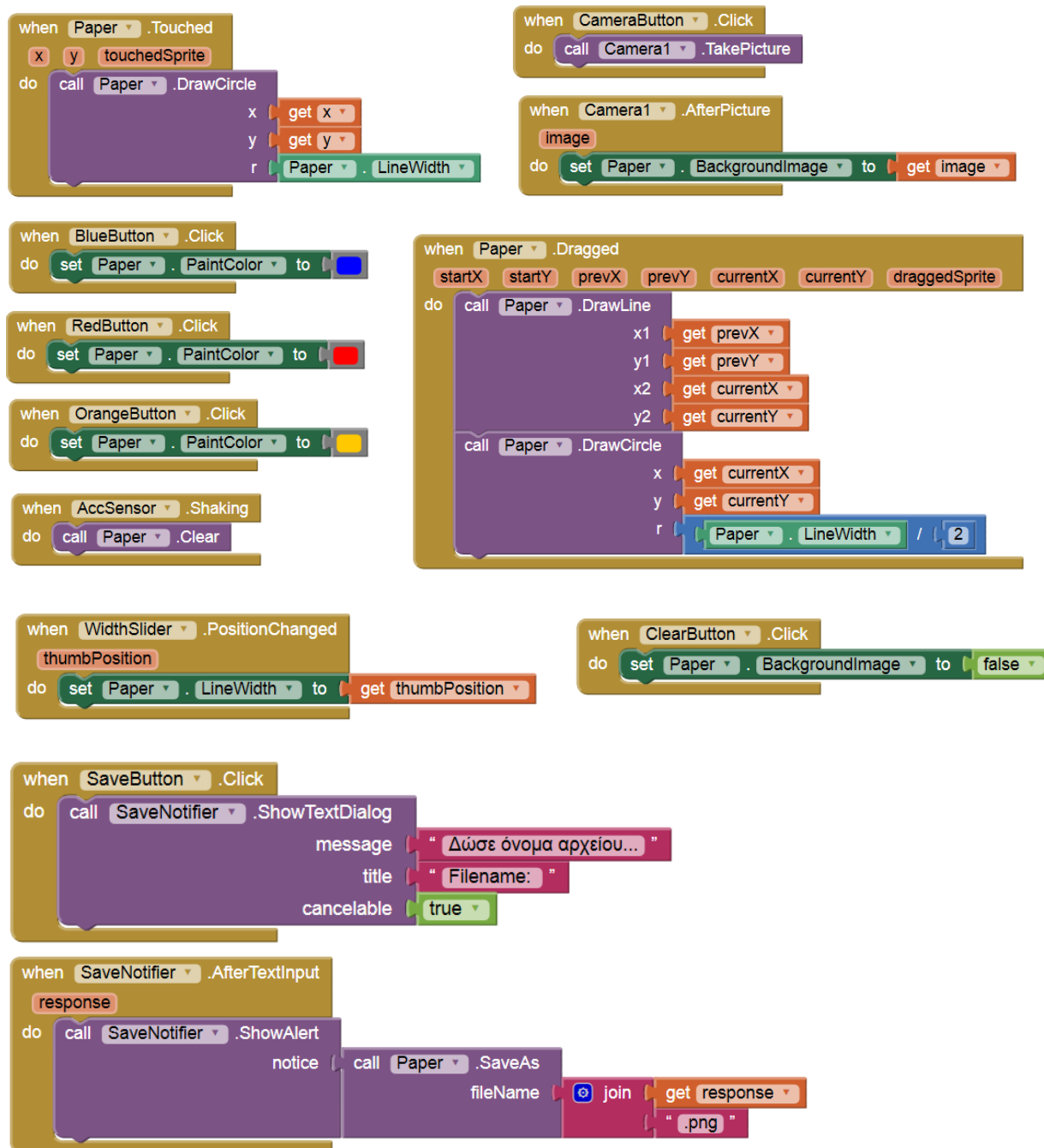


Όταν πατηθεί το πλήκτρο SaveButton, θέσε σε μια τοπική μεταβλητή με όνομα *name* το όνομα αρχείου *fp-[ώρα\_συστήματος].png* και αποθήκευσε τη ζωγραφιά του καμβά στη συσκευή με το συγκεκριμένο όνομα αρχείου. Εμφάνισε ένα παράθυρο διαλόγου με τίτλο Saved Image, με κυρίως μήνυμα *filename: [όνομα\_αποθηκευμένης\_εικόνας]* και με ένα κουμπί OK.



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

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



The image displays a collection of MIT App Inventor code blocks for a Finger Painting application. The blocks are organized as follows:

- when Paper .Touched:** A block with variables `x` and `y` for `touchedSprite`. The `do` block contains `call Paper .DrawCircle` with arguments `x` (from `get x`), `y` (from `get y`), and `r` (from `Paper . LineWidth`).
- when CameraButton .Click:** A block with `do` containing `call Camera1 .TakePicture`.
- when Camera1 .AfterPicture:** A block with variable `image` and `do` containing `set Paper . BackgroundImage` to `get image`.
- when BlueButton .Click:** A block with `do` containing `set Paper . PaintColor` to a blue color swatch.
- when RedButton .Click:** A block with `do` containing `set Paper . PaintColor` to a red color swatch.
- when OrangeButton .Click:** A block with `do` containing `set Paper . PaintColor` to an orange color swatch.
- when AccSensor .Shaking:** A block with `do` containing `call Paper .Clear`.
- when Paper .Dragged:** A large block with variables `startX`, `startY`, `prevX`, `prevY`, `currentX`, `currentY`, and `draggedSprite`. The `do` block contains:
  - `call Paper .DrawLine` with arguments `x1` (from `get prevX`), `y1` (from `get prevY`), `x2` (from `get currentX`), and `y2` (from `get currentY`).
  - `call Paper .DrawCircle` with arguments `x` (from `get currentX`), `y` (from `get currentY`), and `r` (from `Paper . LineWidth / 2`).
- when WidthSlider .PositionChanged:** A block with variable `thumbPosition` and `do` containing `set Paper . LineWidth` to `get thumbPosition`.
- when ClearButton .Click:** A block with `do` containing `set Paper . BackgroundImage` to `false`.
- when SaveButton .Click:** A block with `do` containing `call SaveNotifier .ShowTextDialog` with arguments `message` (from `" Δώσε όνομα αρχείου... "`), `title` (from `" Filename: "`), and `cancelable` (from `true`).
- when SaveNotifier .AfterTextInput:** A block with variable `response` and `do` containing `call SaveNotifier .ShowAlert` with argument `notice` (from `call Paper .SaveAs`), and `fileName` (from `join` of `get response` and `".png"`).

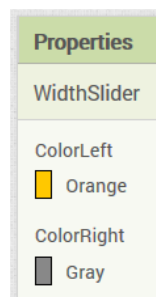
Εικόνα 39 - Οι εντολές της εφαρμογής Finger Painting

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

## ΕΠΕΚΤΑΣΕΙΣ

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

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

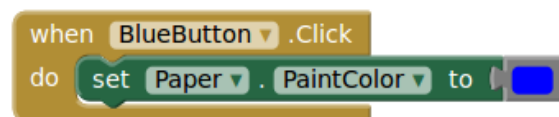


Εικόνα 40 - Οι ιδιότητες ColorLeft και ColorRight του μεταβολέα

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

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

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

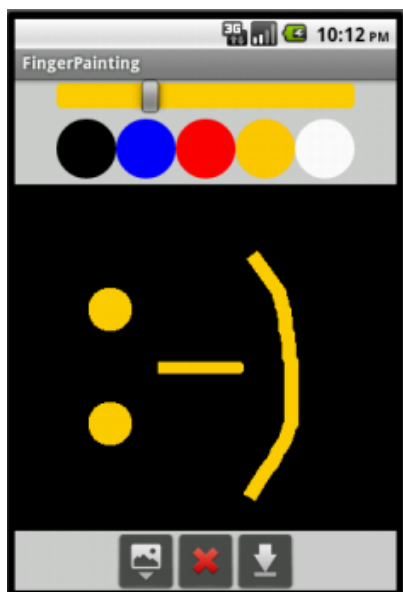


Εικόνα 41 - Ορισμός του ενεργού χρώματος σε μπλε

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

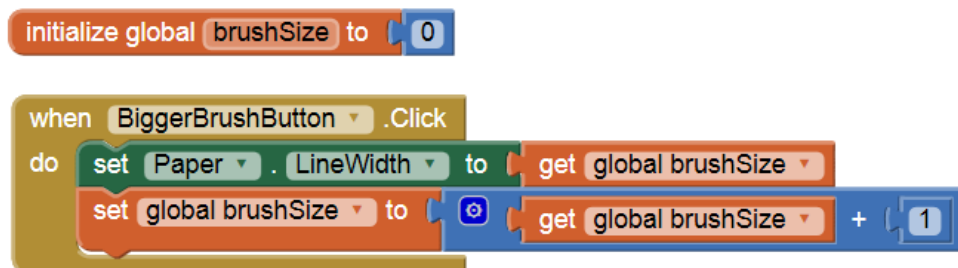
Σας ζητείται να δημιουργήσετε μια διαδικασία setColor [x] η οποία θα καλείται όποτε πατάμε σε κάποιο κουμπί χρώματος και θα αναλαμβάνει να κάνει όλες τις απαραίτητες αλλαγές ιδιοτήτων.

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



Εικόνα 42 - Ενδεικτικό στιγμιότυπο της νέας εφαρμογής FingerPainting

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



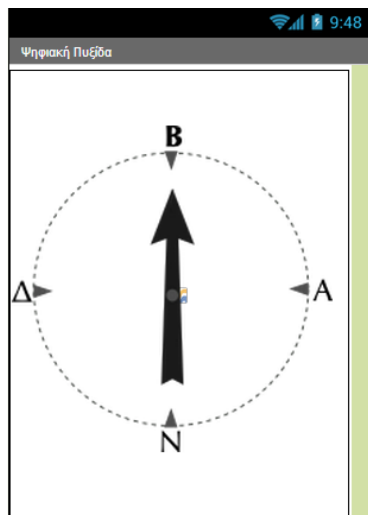
Εικόνα 43 - Τμήμα εντολών για αύξηση του μεγέθους γραμμής με το πάτημα του ανάλογου κουμπιού

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΠΥΞΙΔΑ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 44 - Το περιβάλλον της εφαρμογής Πυξίδα

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα Compass.

#### Designer Γραφικό Περιβάλλον Εφαρμογής

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



επιλέγουμε το  
**αντικείμενο**

Screen1

μεταβάλλουμε τις  
**ιδιότητες**

AlignHorizontal: Center  
Screen Orientation: Portrait  
Scrollable: No  
Title: Ψηφιακή Πυξίδα

## Βήμα 2 Προσθήκη αρχείων πολυμέσων

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



ανεβάζουμε το αρχείο με  
**όνομα**

σύνομη  
**περιγραφή**

compass-arrow.png

Εικόνα πυξίδας με βέλος

## Βήμα 3 Προσθήκη αντικειμένων στο γραφικό περιβάλλον

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



από την  
**ομάδα**

μεταφέρουμε το  
**αντικείμενο**

του δίνουμε το  
**όνομα**

μεταβάλλουμε τις  
**ιδιότητες**

Drawing and  
Animation

Canvas

CompassCanvas

Width: Fill Parent  
Height: Fill Parent

Drawing and  
Animation

ImageSprite

CompassSprite

X : 0  
Y : 50  
Width : 300 pixels  
Height : 300 pixels

Sensors

OrientationSensor

CompassSensor

## Βήμα 4 Ανιχνεύοντας την αλλαγή προσανατολισμού

Συγγραφή Προγράμματος

Blocks

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



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

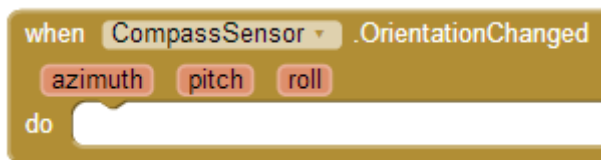
Roll : Έχει τιμή 0 όταν η συσκευή είναι επίπεδη, αυξάνεται στις 90° όταν η συσκευή είναι γυρισμένη στη δεξιά της πλευρά κ.ο.κ.

Pitch : Έχει τιμή 0 όταν η συσκευή είναι επίπεδη, αυξάνεται στις 90° όταν η πάνω πλευρά της συσκευής κοιτάει προς τα κάτω κ.ο.κ.

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

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

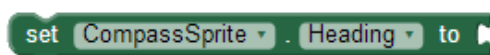
Από το αντικείμενο CompassSensor χρησιμοποιούμε την εντολή



Εικόνα 45 - Το γεγονός αλλαγής προσανατολισμού

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

Από το αντικείμενο CompassSprite επιλέγουμε την εντολή ...



Εικόνα 46 - Εντολή αλλαγής κατεύθυνσης μιας εικόνας

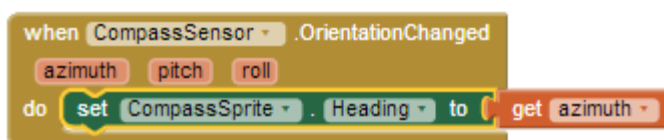
... και την τοποθετούμε μέσα στο γεγονός της αλλαγής προσανατολισμού.

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



Εικόνα 47 - Παίρνοντας την τιμή του αζιμούθιου

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



Εικόνα 48 - Θέτοντας τον προσανατολισμό της πυξίδας στο Βορρά



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

## ΕΠΕΚΤΑΣΕΙΣ

1. Τροποποιήστε την εφαρμογή της πυξίδας, ώστε όταν η συσκευή δείχνει προς το Βορρά (με απόκλιση +/- 10 μοίρες) να αναπαράγει ένα ηχητικό σήμα της επιλογής σας.
2. Προσθέστε 3 ετικέτες που θα εμφανίζουν στην οθόνη τις τιμές των ιδιοτήτων Roll, Pitch και Azimuth.
  - A. Περιγράψτε ποια είναι η θέση της κινητής συσκευής όταν η ιδιότητα Roll παίρνει την τιμή 180.
  - B. Περιγράψτε ποια είναι η θέση της κινητής συσκευής όταν η ιδιότητα Pitch παίρνει την τιμή 270.



## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΠΙΑΣΕ ΤΗ ΣΗΜΑΙΑ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 49 - Εφαρμογή "Πιάσε τη σημαία"

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα CatchTheFlag.

#### Designer Γραφικό Περιβάλλον Εφαρμογής

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



επιλέγουμε το  
**αντικείμενο**

μεταβάλλουμε τις  
**ιδιότητες**

Screen1

Screen Orientation: Portrait

Scrollable: No

Title: Πιάσε τη σημαία

## Βήμα 2

### Προσθήκη των απαραίτητων αρχείων πολυμέσων

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



ανεβάζουμε το αρχείο με  
**όνομα**

σύντομη  
**περιγραφή**

tiles.jpg

Εικόνα φόντου

Soccer\_Ball\_small.png

Γραφικό μπάλας

hole\_small.png

Γραφικό τρύπας

red\_flag.png

Γραφικό σημαίας

Smashing.mp3

Ήχος σύγκρουσης με τρύπα

Tada.mp3

Ήχος σύγκρουσης με σημαία

## Βήμα 3

### Προσθήκη αντικειμένων στο γραφικό περιβάλλον

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



από την  
**ομάδα**

μεταφέρουμε το  
**αντικείμενο**

του δίνουμε το  
**όνομα**

μεταβάλλουμε τις  
**ιδιότητες**

Drawing and  
Animation

Canvas

GameCanvas

BackgroundImage : tiles.jpg

Width : Fill Parent

Height : Fill Parent

Drawing and  
Animation

ImageSprite

BallSprite

Picture : Soccer\_Ball\_small.png

Width : 40 pixels

Height : 40 pixels

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

## Συγγραφή Προγράμματος

## Blocks

#### Βήμα 4 Αρχική θέση μπάλας και σημαίας

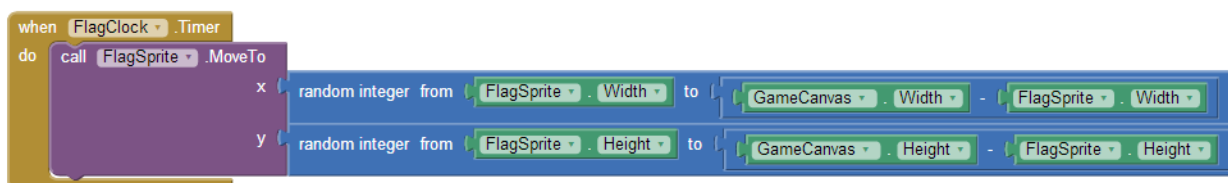
Στο ξεκίνημα του παιχνιδιού θέλουμε τα γραφικά της μπάλας και της σημαίας να τοποθετηθούν σε τυχαίες θέσεις, μέσα στα όρια του καμβά, φυσικά.



Εικόνα 50 - Αρχική θέση μπάλας και σημαίας

#### Βήμα 5 Αλλαγή θέσης σημαίας κάθε 4 δευτερόλεπτα

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

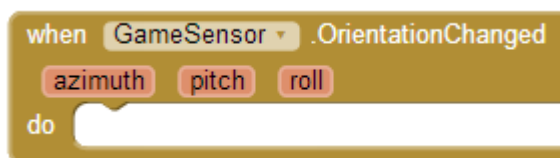


Εικόνα 51 - Μετακίνηση σημαίας κάθε 4 δευτερόλεπτα

## Βήμα 6 Μετακίνηση μπάλας με τη βοήθεια του αισθητήρα

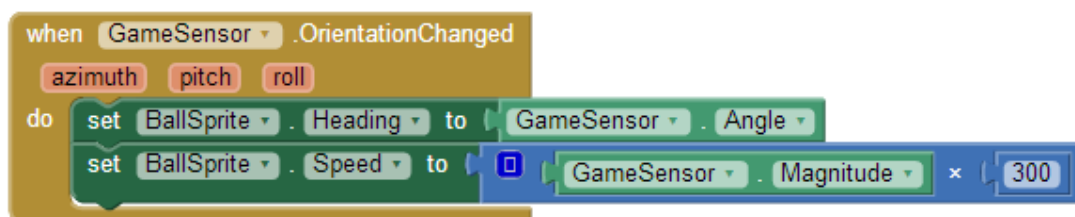
Όταν η κινητή μας συσκευή αλλάζει προσανατολισμό θα πρέπει η μπάλα να μετακινείται προς την κατεύθυνση που «έστριψε» και η συσκευή. Επίσης, η ταχύτητα της μπάλας θα εξαρτάται από το πόσο μεγάλη κλίση δώσαμε στη συσκευή μας.

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



Εικόνα 52 - Γεγονός αλλαγής προσανατολισμού

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



Εικόνα 53 - Αλλαγή ταχύτητας και προσανατολισμού της μπάλας



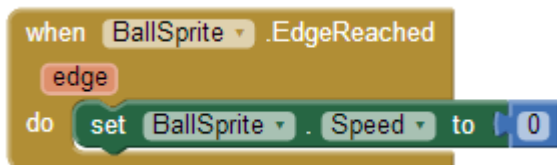
Η ιδιότητα **Angle** του αισθητήρα προσανατολισμού μας δίνει την διεύθυνση προς στην οποία είναι στραμμένη η συσκευή, σε σχέση με την επίπεδη – οριζόντια θέση.

Η ιδιότητα **Magnitude** του αισθητήρα προσανατολισμού μας δίνει έναν αριθμό μεταξύ 0 και 1 που δείχνει πόσο μεγάλη είναι η κλίση της συσκευής.

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

## Βήμα 7 Μηδενισμός ταχύτητας της μπάλας στα άκρα του καμβά

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



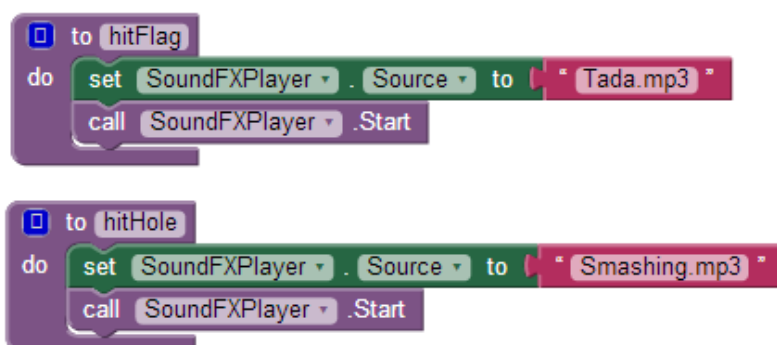
Εικόνα 54 - Μηδενισμός ταχύτητας της μπάλας

## Βήμα 8 Σύγκρουση μπάλας με άλλα αντικείμενα

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

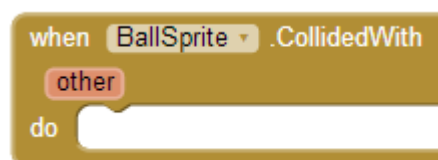
Στην πρώτη εκδοχή της εφαρμογής μας, οι διαδικασίες απλά θα θέτουν το κατάλληλο αρχείο ήχου για αναπαραγωγή στο αντικείμενο SoundFXPlayer. Στις επεκτάσεις που θα κάνετε θα εμπλουτιστούν με τις κατάλληλες εντολές.

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



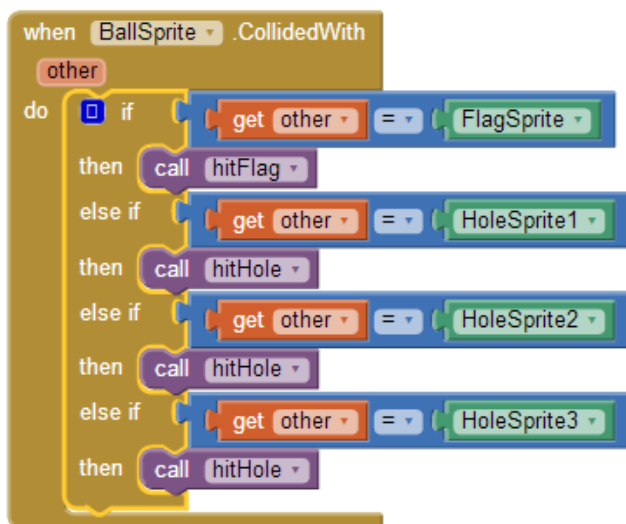
Εικόνα 55 - Διαδικασίες για τη σημαία και την τρύπα

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



Εικόνα 56 - Γεγονός σύγκρουσης μπάλας

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



Εικόνα 57 - Έλεγχος σύγκρουσης μπάλας με αντικείμενο



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

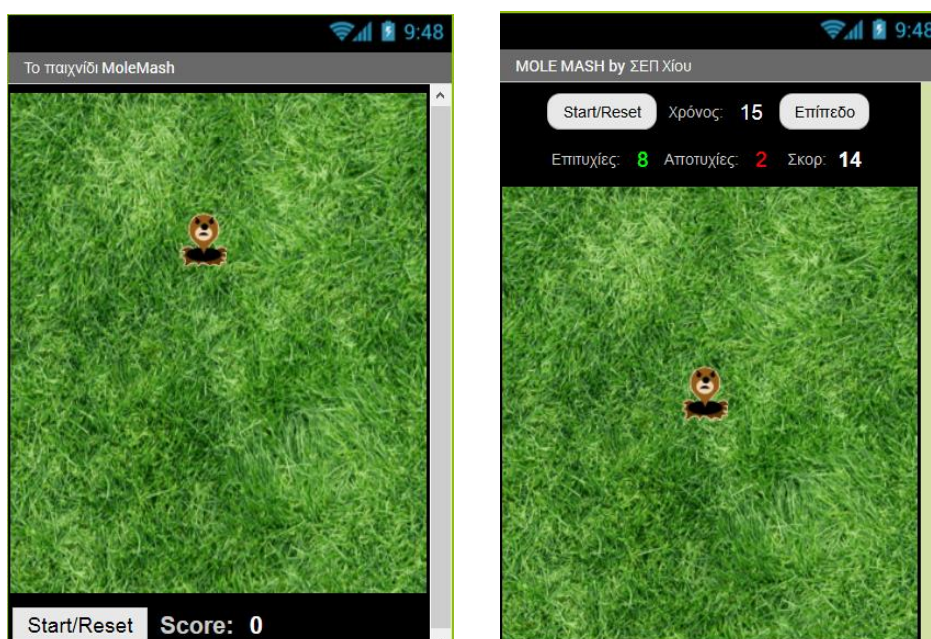
## ΕΠΕΚΤΑΣΕΙΣ

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

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

Ενδεικτικές εκδοχές του παιχνιδιού φαίνονται στις παρακάτω εικόνες:



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

### Βασικές λειτουργίες:

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

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



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

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

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

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

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

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

### Ιδέες για επέκταση:

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

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

- Εάν ο παίκτης αγγίξει την πίστα, αλλά όχι το mole, τότε χάνει ... πόντους.

*Το σκορ ενδεικτικά θα μπορούσε να προκύπτει π.χ. κερδίζοντας 2 πόντους για κάθε εύστοχο άγγιγμα και χάνοντας έναν πόντο για κάθε άστοχο άγγιγμα.*

- Ένα χρονόμετρο μετράει αντίστροφα, ώστε το παιχνίδι να τελειώνει σε ... δευτερόλεπτα.

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

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

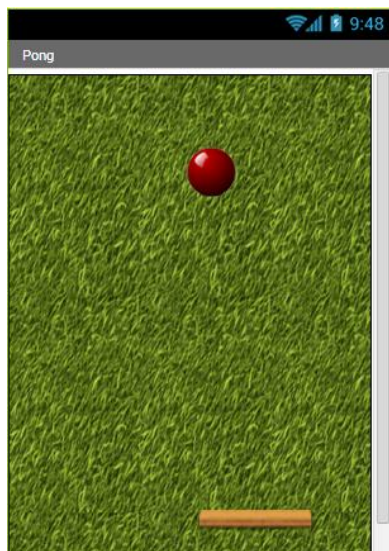
*Ξεκινώντας, μπορεί ο παίκτης να έχει συγκεκριμένο αριθμό ζωών και για κάθε «κακό» ζωάκι που αγγίζει να του αφαιρείται μια ζωή και να χάνει πόντους.*

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

*Θα χρειαστείτε ένα αντικείμενο ListPicker από την κατηγορία User Interface, το οποίο προσθέτει ένα κουμπί για την επιλογή κάποιου στοιχείου από λίστα. Ορίστε όσα επίπεδα θέλετε και για κάθε επίπεδο, φροντίστε να αλλάζει η κατάλληλη ιδιότητα.*



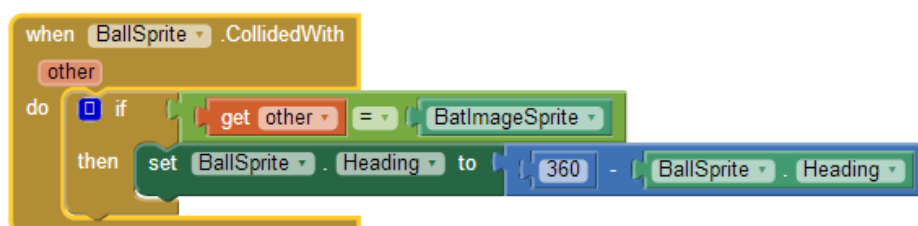
2. Αναπτύξτε μια εφαρμογή που θα υλοποιεί το παιχνίδι Pong. Το παιχνίδι αποτελείται από μια μπάλα που κινείται συνεχώς και αναπηδά στα τοιχώματα. Ο παίκτης χειρίζεται μια σανίδα, την οποία μετακινεί αριστερά δεξιά με το δάκτυλο. Ο παίκτης μετακινώντας τη σανίδα προσπαθεί να χτυπάει την μπάλα και να μην της επιτρέψει να αγγίξει την κάτω πλευρά της οθόνης (στην περίπτωση αυτή το παιχνίδι τερματίζει και ο παίκτης χάνει).



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

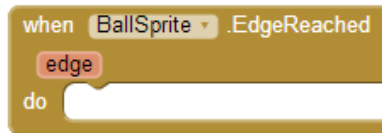
Για τη μετακίνηση της σανίδας θα σας χρειαστεί το γεγονός **when [.....] Dragged** που θα βρείτε στην ομάδα εντολών του γραφικού της σανίδας.

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



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

Επίσης για να ανιχνεύσετε τη σύγκρουση της μπάλας με τα «τοιχώματα» της οθόνης θα χρειαστείτε το γεγονός:

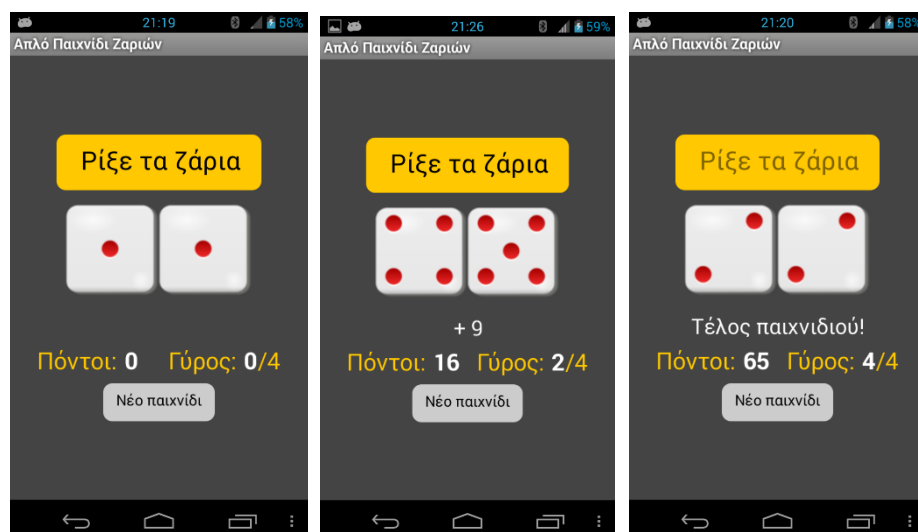


Η τιμή της ιδιότητας `edge` μας δείχνει με ποια πλευρά της οθόνης «συγκρούστηκε» η μπάλα. Η κάτω πλευρά της οθόνης αντιστοιχεί στην τιμή `-1` του `edge`.

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

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

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



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

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

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

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

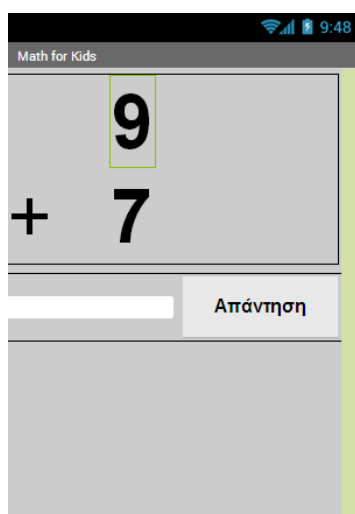
Φυσικά, αν το επιθυμείτε, μπορείτε να ορίσετε τους δικούς σας κανόνες...

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

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

Μπορείτε να υλοποιήσετε την εφαρμογή με δύο τρόπους:

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



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

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

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



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

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



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

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

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

## ΠΑΡΑΡΤΗΜΑ – ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ

## PHOTOGALLERY (ΕΠΕΚΤΑΣΕΙΣ)

```

when PreviousButton .Click
do
  set TimerClock . TimerEnabled to false
  call changelimage
    number -1
  set TimerClock . TimerEnabled to true

```

```

when NextButton .Click
do
  set TimerClock . TimerEnabled to false
  call changelimage
    number 1
  set TimerClock . TimerEnabled to true

```

```

when GalleryCanvas .Flung
  x y speed heading xvel yvel flungSprite
do
  set TimerClock . TimerEnabled to false
  if
    get heading ≥ -30 and get heading ≤ 30
  then
    call changelimage
      number 1
  else if
    get heading ≥ -200 and get heading ≤ -160
  then
    call changelimage
      number -1
  set TimerClock . TimerEnabled to true

```

## ΑΡΙΘΜΟΜΗΧΑΝΗ – ΔΙΑΔΙΚΑΣΙΑ ΠΟΥ ΔΙΑΒΑΖΕΙ ΤΟΝ ΑΡΙΘΜΟ

```

to makeNumber x
do
  set MainLabel . Text to " "
  set global number to join
    get global number
    get x
  set MainLabel . Text to get global number

```

```

when Button1.Click
do
  call makeNumber x 1

when Button2.Click
do
  call makeNumber x 2

when Button3.Click
do
  call makeNumber x 3

when Button4.Click
do
  call makeNumber x 4

when Button5.Click
do
  call makeNumber x 5

when Button6.Click
do
  call makeNumber x 6

when Button7.Click
do
  call makeNumber x 7

when Button8.Click
do
  call makeNumber x 8

when Button9.Click
do
  call makeNumber x 9

when Button0.Click
do
  call makeNumber x 0
    
```

ΑΡΙΘΜΟΜΗΧΑΝΗ – ΔΙΑΔΙΚΑΣΙΕΣ ΚΑΘΑΡΙΣΜΟΥ & ΕΠΙΛΟΓΗΣ ΠΡΑΞΗΣ

```

to clearCalc
do
  set global number to ""
  set global result to 0
  set global action to ""
    
```

Εικόνα 58 - Η διαδικασία καθαρισμού

```

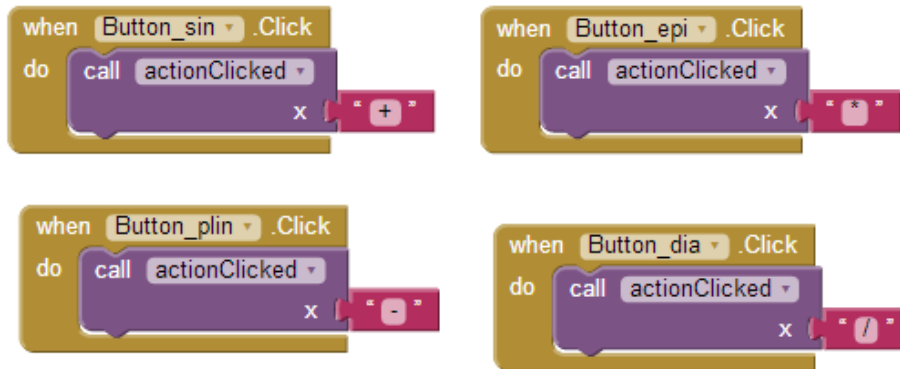
when Button_CLICK.Click
do
  call calcResult
  set MainLabel.Text to "0"
    
```

Εικόνα 59 - Κλήση της διαδικασίας καθαρισμού

```

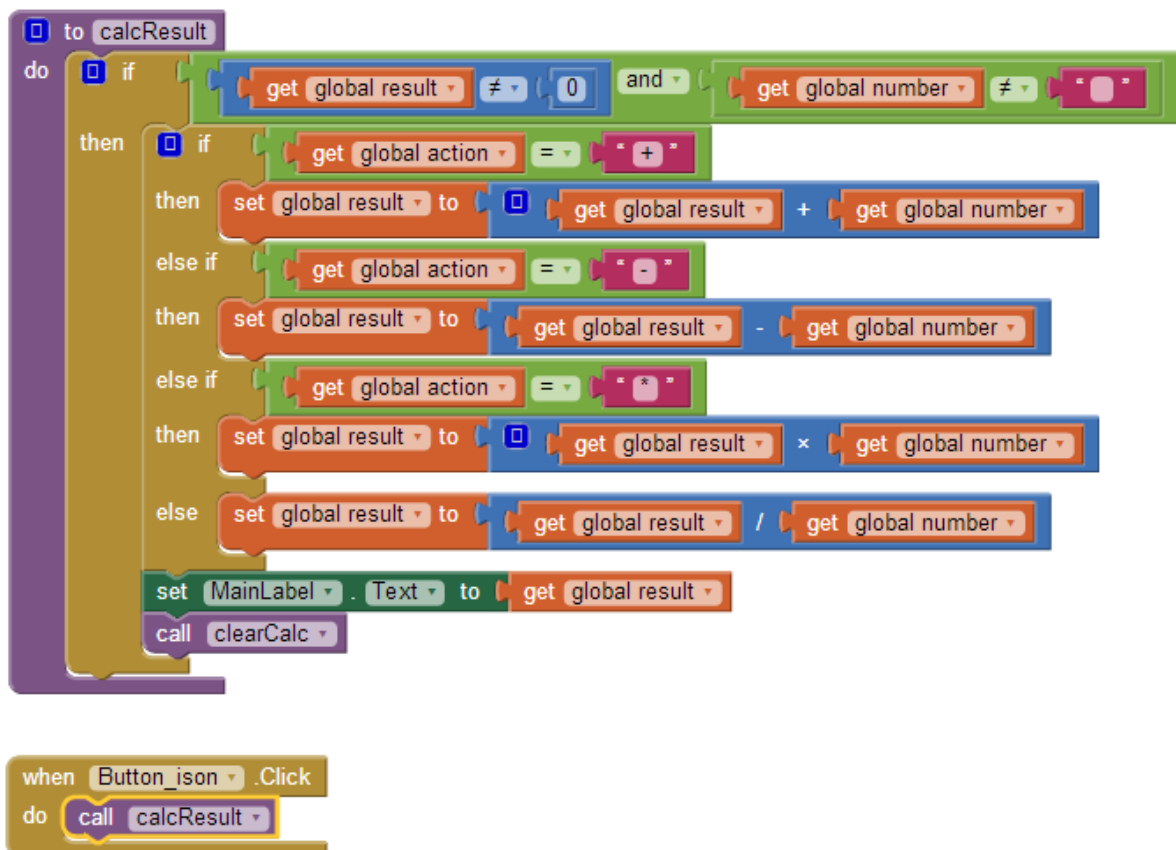
to actionClicked x
do
  if (get global number ≠ "")
  then
    set global action to get x
    set MainLabel.Text to get x
    set global result to get global number
    set global number to ""
    
```

Εικόνα 60 - Η διαδικασία επιλογής αριθμητικής πράξης



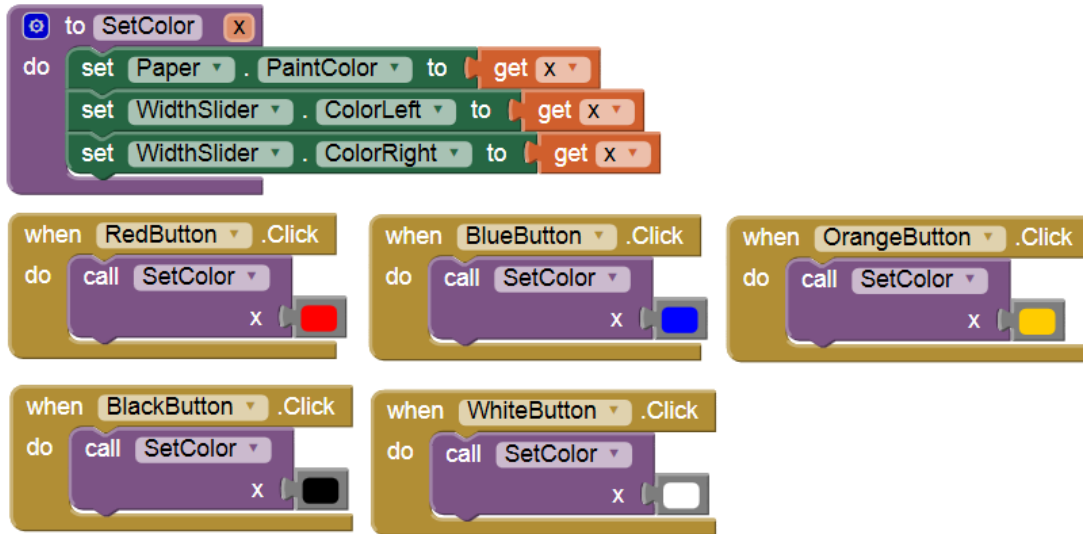
Εικόνα 61 - Κλήση της διαδικασίας επιλογής πράξης

### ΑΡΙΘΜΟΜΗΧΑΝΗ – ΔΙΑΔΙΚΑΣΙΑ ΥΠΟΛΟΓΙΣΜΟΥ ΑΠΟΤΕΛΕΣΜΑΤΟΣ





## FINGERPAINTING (ΕΠΕΚΤΑΣΕΙΣ)



```

function SetColor(x)
do
  set Paper.PaintColor to get x
  set WidthSlider.ColorLeft to get x
  set WidthSlider.ColorRight to get x

when RedButton.Click
do
  call SetColor x

when BlueButton.Click
do
  call SetColor x

when OrangeButton.Click
do
  call SetColor x

when BlackButton.Click
do
  call SetColor x

when WhiteButton.Click
do
  call SetColor x
  
```

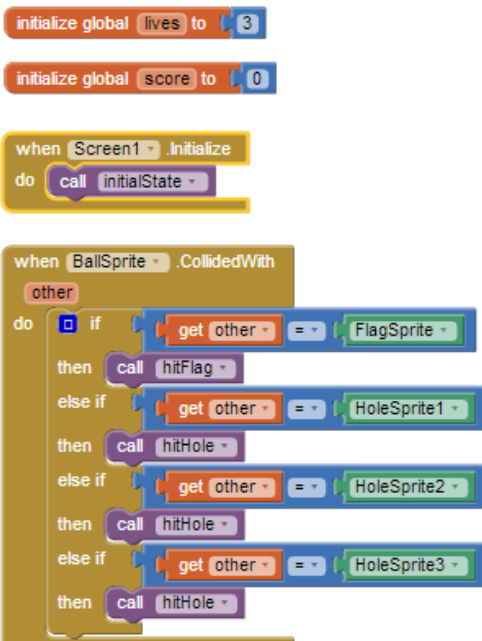
## ΠΥΞΙΔΑ (ΕΠΕΚΤΑΣΗ – ΗΧΗΤΙΚΗ ΕΙΔΟΠΟΙΗΣΗ ΓΙΑ ΤΟ ΒΟΡΡΑ)



```

when CompassSensor.OrientationChanged
  azimuth pitch roll
do
  set CompassSprite.Heading to get azimuth
  if
    (get azimuth >= 0 and get azimuth <= 10) or
    (get azimuth >= 350 and get azimuth <= 360)
  then
    call NorthSound.Play
  
```

## ΠΙΑΣΕ ΤΗ ΣΗΜΑΙΑ - ΕΠΕΚΤΑΣΕΙΣ



```

initialize global lives to 3
initialize global score to 0

when Screen1.Initialize
do
  call initialState

when BallSprite.CollidedWith other
do
  if
    get other = FlagSprite
  then
    call hitFlag
  else if
    get other = HoleSprite1
  then
    call hitHole
  else if
    get other = HoleSprite2
  then
    call hitHole
  else if
    get other = HoleSprite3
  then
    call hitHole
  
```

```

when FlagClock . Timer
do
  call FlagSprite . MoveTo
  x random integer from FlagSprite . Width to GameCanvas . Width - FlagSprite . Width
  y random integer from FlagSprite . Height to GameCanvas . Height - FlagSprite . Height

  call HoleSprite1 . MoveTo
  x random integer from HoleSprite1 . Width to GameCanvas . Width - HoleSprite1 . Width
  y random integer from HoleSprite1 . Height to GameCanvas . Height - HoleSprite1 . Height

  call HoleSprite2 . MoveTo
  x random integer from HoleSprite2 . Width to GameCanvas . Width - HoleSprite2 . Width
  y random integer from HoleSprite2 . Height to GameCanvas . Height - HoleSprite2 . Height

  call HoleSprite3 . MoveTo
  x random integer from HoleSprite3 . Width to GameCanvas . Width - HoleSprite3 . Width
  y random integer from HoleSprite3 . Height to GameCanvas . Height - HoleSprite3 . Height

```

```

when GameSensor . OrientationChanged
  azimuth pitch roll
do
  set BallSprite . Heading to GameSensor . Angle
  set BallSprite . Speed to GameSensor . Magnitude x 300

```

```

when BallSprite . EdgeReached
  edge
do
  set BallSprite . Speed to 0

```

```

to hitFlag
do
  set global score to get global score + 5
  set ScoreLabel . Text to join " Σκορ: " get global score
  set SoundFXPlayer . Source to " Tada.mp3 "
  call SoundFXPlayer . Start
  call FlagSprite . MoveTo
  x random integer from FlagSprite . Width to GameCanvas . Width - FlagSprite . Width
  y random integer from FlagSprite . Height to GameCanvas . Height - FlagSprite . Height

```

```

to initialState
do
  set FlagClock . TimerEnabled to true
  set GameSensor . Enabled to true
  set global lives to 3
  set LivesLabel . Text to join " Ζωές: " get global lives
  set global score to 0
  set ScoreLabel . Text to join " Σκορ: " get global score
  call BallSprite . MoveTo
  x random integer from BallSprite . Width to GameCanvas . Width - BallSprite . Width
  y random integer from BallSprite . Height to GameCanvas . Height - BallSprite . Height
  call FlagSprite . MoveTo
  x random integer from FlagSprite . Width to GameCanvas . Width - FlagSprite . Width
  y random integer from FlagSprite . Height to GameCanvas . Height - FlagSprite . Height

```

```

to hitHole
do
  set global lives to (get global lives) - 1
  set LivesLabel . Text to (join (" Ζωές: " (get global lives)))
  set SoundFXPlayer . Source to "Smashing.mp3"
  call SoundFXPlayer . Start
  if (get global lives = 0)
  then call gameOver
  
```

```

when PlayAgainSprite . Touched
x y
do
  call initialState
  set PlayAgainSprite . Visible to false
  
```

```

to gameOver
do
  set BangSprite . Visible to true
  set BallSprite . Speed to 0
  set FlagClock . TimerEnabled to false
  set SoundFXPlayer . Source to "explosion.mp3"
  call SoundFXPlayer . Start
  set GameoverClock . TimerEnabled to true
  set GameSensor . Enabled to false
  
```

```

when GameoverClock . Timer
do
  set BangSprite . Visible to false
  set GameoverClock . TimerEnabled to false
  set PlayAgainSprite . Visible to true
  
```

## MOLEMASH

```

when Pista . Touched
x y touchedSprite
do
  if (get touchedSprite)
  then
    set global Hits to (get global Hits) + 1
    set TimeClock . TimerEnabled to false
    call MoveMole
    set TimeClock . TimerEnabled to true
  else
    set global Missed to (get global Missed) + 1
  call UpdatePoints
  
```

```

initialize global TimeLeft to 15
initialize global Score to 0
initialize global Hits to 0
initialize global Missed to 0
  
```

```

when StartResetButton . Click
do
  set global Hits to 0
  set global Missed to 0
  set global Score to 0
  set global TimeLeft to 15
  call UpdatePoints
  set TimeClock . TimerEnabled to true
  set Mole . Visible to true
  
```

```

to UpdatePoints
do
  set HitsLabel . Text to (get global Hits)
  set MissedLabel . Text to (get global Missed)
  set ScoreLabel . Text to ((get global Hits) * 2) - (get global Missed)
  
```

```

to MoveMole
do
  call Mole . MoveTo
  x random integer from 0 to (Pista . Width) - (Mole . Width)
  y random integer from 0 to (Pista . Height) - (Mole . Height)
  
```

```

when TimeClock.Timer
do
  call MoveMole
  set global TimeLeft to (get global TimeLeft) - 1
  set TimeLeftLabel.Text to (get global TimeLeft)
  if (get global TimeLeft) ≤ 0
  then
    set TimeClock.TimerEnabled to false
    set MoleClock.TimerEnabled to false
    set Mole.Visible to false
  
```

---

## PONG

```

to ballStart
do
  call BallSprite.MoveTo
  x (Screen1.Width / 2)
  y (BallSprite.Height)
  set BallSprite.Heading to (random integer from 225 to 315)
  set BallSprite.Interval to 10
  set BallSprite.Speed to 10

```

```

when Screen1.Initialize
do
  call batStart
  call ballStart
  set GameoverSprite.Visible to false

```

```

when BallSprite.EdgeReached
edge
do
  if (get edge) = -1
  then
    set BallSprite.Speed to 0
    set GameoverSprite.Visible to true
  else
    call BallSprite.Bounce
    edge (get edge)
  
```

```

when BatImageSprite.Dragged
startX startY prevX prevY currentX currentY
do
  call BatImageSprite.MoveTo
  x (get currentX)
  y (BatImageSprite.Y)

```

```

to batStart
do
  call BatImageSprite . MoveTo
  x <= Screen1 . Width / 2 - BatImageSprite . Width / 2
  y <= PongCanvas . Height - BatImageSprite . Height + BallSprite . Height

when PongCanvas . Touched
x y touchedSprite
do
  set BatImageSprite . X to get x

when BallSprite . CollidedWith
other
do
  if get other = BatImageSprite
  then
    set BallSprite . Heading to 360 - BallSprite . Heading

when GameoverSprite . Touched
x y
do
  call batStart
  call ballStart
  set GameoverSprite . Visible to false
    
```

## ΠΑΙΧΝΙΔΙ ΜΕ ΖΑΡΙΑ (DICEGAME)

```

initialize global dice1 to 0
initialize global dice2 to 0
initialize global score to 0
initialize global points to 0
initialize global sum to 0
initialize global rounds to 0
initialize global MaxRounds to 4

when AccSensor . Shaking
do
  call runRound

when ThrowButton . Click
do
  call runRound

when ResetButton . Click
do
  set global rounds to 0
  set global score to 0
  set global points to 0
  set RoundsLabel . Text to get global rounds
  set ScoreLabel . Text to get global points
  set ResultLabel . Text to ""
  set TotalRoundsLabel . Text to join " / "
  get global MaxRounds
  set Dice1Image . Picture to "1.png"
  set Dice2Image . Picture to "1.png"
  set ThrowButton . Enabled to true
  set AccSensor . Enabled to true

when Screen1 . Initialize
do
  set TotalRoundsLabel . Text to join " / "
  get global MaxRounds

to runRound
do
  if get global rounds < get global MaxRounds
  then
    call throwDice
    call checkDicePoints
    set global rounds to get global rounds + 1
    set RoundsLabel . Text to get global rounds
  else
    call endGame
    
```

```

to throwDice
do
  set ResultLabel . Text to " "
  set global dice1 to random integer from 1 to 6
  set global dice2 to random integer from 1 to 6
  set global sum to (get global dice1 + get global dice2)
  set Dice1Image . Picture to (join (get global dice1) ".png")
  set Dice2Image . Picture to (join (get global dice2) ".png")
  call ThrowSound . Play

to getPoints x
do
  set global points to (get X)
  set ResultLabel . Text to (join (" + " (get global points)))

to endGame
do
  set ResultLabel . Text to " Τέλος παιχνιδιού! "
  set ThrowButton . Enabled to false
  set AccSensor . Enabled to false
  call Player1 . Vibrate milliseconds 200

to checkDicePoints
do
  if (get global dice1 = 1 and get global dice2 ≠ 1 or get global dice1 ≠ 1 and get global dice2 = 1)
  then
    call getPoints x 0
    call endGame
    call Player1 . Start
  else
    if (get global dice1 = 1 and get global dice2 = 1)
    then
      call getPoints x 40
    else if (get global dice1 = get global dice2)
    then
      call getPoints x 25
    else
      call getPoints x (get global sum)
  set global score to (get global score + get global points)
  set ScoreLabel . Text to (get global score)
  
```

### ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΜΕ ΠΕΔΙΟ ΚΕΙΜΕΝΟΥ)

```

initialize global num1 to 0
initialize global correctAnswer to 0
initialize global num2 to 0
initialize global wrongAnswer1 to 0
initialize global userAnswer to 0

when Screen1 . Initialize
do
  call createProblem

to createProblem
do
  set global num1 to random integer from 1 to 10
  set global num2 to random integer from 1 to 10
  set No1Label . Text to (get global num1)
  set No2Label . Text to (get global num2)
  set global correctAnswer to (get global num1 + get global num2)

when Answer1Button . Click
do
  set global userAnswer to AnswerTextBox . Text
  call acceptAnswer

to acceptAnswer
do
  if (get global correctAnswer = get global userAnswer)
  then
    set MediaPlayer . Source to " success.wav "
    call MediaPlayer . Start
    call createProblem
  else
    set MediaPlayer . Source to " failure.mp3 "
    call MediaPlayer . Start
  set AnswerTextBox . Text to " "
  
```

## ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΠΑΡΑΛΛΑΓΗ ΜΕ ΚΟΥΜΠΙΑ)

```

to acceptAnswer
do
  if (get global correctAnswer = get global userAnswer)
  then
    set MediaPlayer . Source to "success.wav"
    call MediaPlayer . Start
    call createProblem
  else
    set MediaPlayer . Source to "failure.mp3"
    call MediaPlayer . Start

initialize global correctButton to 0

initialize global num1 to 0
initialize global num2 to 0
initialize global userAnswer to 0

initialize global correctAnswer to 0
initialize global wrongAnswer1 to 0
initialize global wrongAnswer2 to 0
  
```

```

to createProblem
do
  set global num1 to random integer from 1 to 10
  set global num2 to random integer from 1 to 10
  set No1Label . Text to get global num1
  set No2Label . Text to get global num2
  set global correctButton to random integer from 1 to 3
  set global correctAnswer to (get global num1 + get global num2)
  set global wrongAnswer1 to random integer from (get global correctAnswer - 5) to (get global correctAnswer + 5)
  set global wrongAnswer2 to random integer from (get global correctAnswer - 5) to (get global correctAnswer + 5)

  if (get global correctButton = 1)
  then
    set Answer1Button . Text to get global correctAnswer
    set Answer2Button . Text to get global wrongAnswer1
    set Answer3Button . Text to get global wrongAnswer2
  else if (get global correctButton = 2)
  then
    set Answer1Button . Text to get global wrongAnswer1
    set Answer2Button . Text to get global correctAnswer
    set Answer3Button . Text to get global wrongAnswer2
  else
    set Answer1Button . Text to get global wrongAnswer1
    set Answer2Button . Text to get global wrongAnswer2
    set Answer3Button . Text to get global correctAnswer
  
```

```

when Screen1 . Initialize
do
  call createProblem

when Answer1Button . Click
do
  set global userAnswer to Answer1Button . Text
  call acceptAnswer

when Answer2Button . Click
do
  set global userAnswer to Answer2Button . Text
  call acceptAnswer

when Answer3Button . Click
do
  set global userAnswer to Answer3Button . Text
  call acceptAnswer
  
```





# App Inventor



## ΜΑΘΗΜΑ 4

### ΣΤΟΧΟΙ

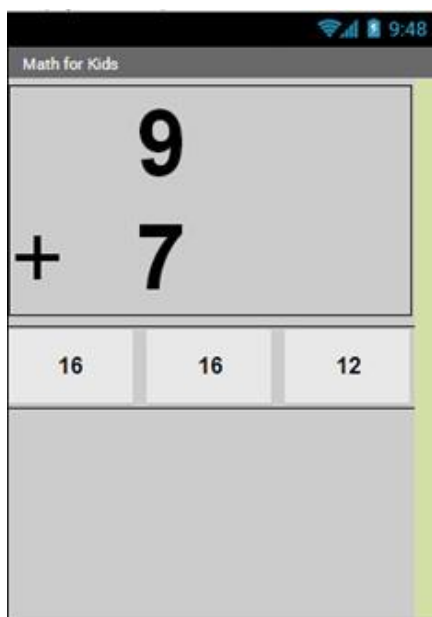
- Να αναγνωρίζετε τις εντολές επανάληψης και να τις συντάσσετε σωστά.
- Να χρησιμοποιείτε την κατάλληλη εντολή επανάληψης σε κάθε περίπτωση.
- Να χρησιμοποιείτε την τοπική βάση δεδομένων της κινητής συσκευής για «μόνιμη» αποθήκευση δεδομένων.
- Να χρησιμοποιείτε τον αισθητήρα GPS για τον εντοπισμό της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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΜΑΘΗΜΑΤΙΚΑ ΓΙΑ ΠΑΙΔΙΑ (ΕΠΕΚΤΑΣΗ)

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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

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



Εικόνα 62 – Το σφάλμα της εφαρμογής Μαθηματικά για παιδιά

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

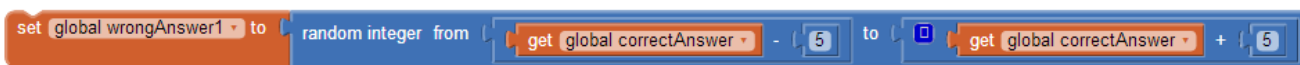
- Η εντολή επανάληψης while

#### Βήμα 1 Έλεγχος της πρώτης λανθασμένης απάντησης

Συγγραφή Προγράμματος

Blocks

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



Εικόνα 63 - Η τιμή της μεταβλητής για την πρώτη λανθασμένη απάντηση σε ένα διάστημα τυχαίων αριθμών

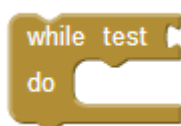
Στο σημείο αυτό θα πρέπει να προστεθεί ένας έλεγχος, ώστε στην περίπτωση που ο τυχαίος αριθμός που παράγεται είναι ίσος με τη σωστή απάντηση, η εφαρμογή να τον ξαναζητά.



Πόσες φορές όμως θα πρέπει να γίνεται η παραπάνω ενέργεια;

## Η ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ WHILE

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



Εικόνα 64 - Η εντολή while



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

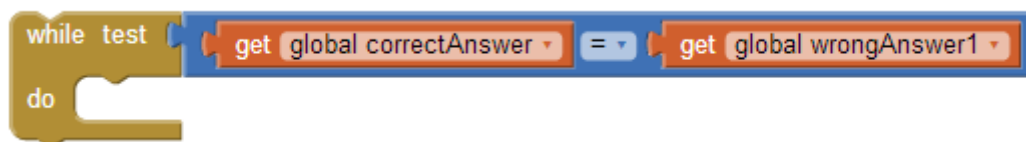
Στο παράδειγμα μας, η εντολή while θα εξετάζει αν η πρώτη λανθασμένη απάντηση έχει την ίδια τιμή με τη σωστή απάντηση. Όσο αυτό ισχύει θα παράγει ξανά έναν τυχαίο αριθμό και θα τον θέτει ως νέα τιμή της πρώτης λανθασμένης απάντησης.

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



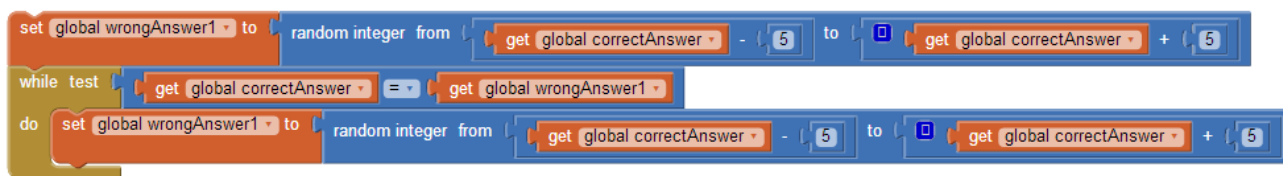
Εικόνα 65 - Έλεγχος αν η λανθασμένη απάντηση είναι ίση με τη σωστή

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



Εικόνα 66 - Έλεγχος με την εντολή while

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



Εικόνα 67 - Η ολοκληρωμένη εντολή while



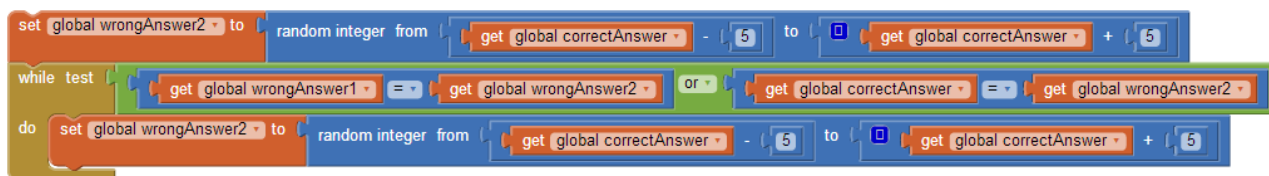
**Πόσες είναι οι λιγότερες φορές που μπορούν να εκτελεστούν οι εντολές μέσα στην while;**

## Βήμα 2 Έλεγχος της δεύτερης λανθασμένης απάντησης

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

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

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα :



Εικόνα 68 - Ο έλεγχος της δεύτερης λανθασμένης απάντησης



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

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

Θέσε εκ νέου την τιμή της σε έναν τυχαίο αριθμό



Η εφαρμογή μας είναι έτοιμη!

Ωρα να την ελέγξουμε με την κινητή μας συσκευή.

## ΕΠΕΚΤΑΣΕΙΣ

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : RANDOM DROPS

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 69 – Στιγμιότυπα της εφαρμογής Finger Painting

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

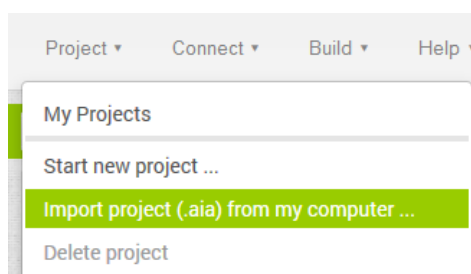
Το κουμπί καθαρισμού (πάνω δεξιά), πέρα του αυτονόητου να καθαρίζει τον καμβά, επιθυμούμε να έχει και μια «κρυμμένη» λειτουργικότητα: όταν πατηθεί παρατεταμένα (long click) θα αλλάζει το χρώμα φόντου του καμβά από άσπρο σε μαύρο και αντίστροφα.

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

## Βήμα 1 Εισαγωγή πηγαίου κώδικα για δημιουργία νέου project

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



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

Συγγραφή Προγράμματος

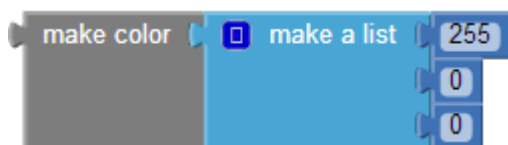
Blocks

## Βήμα 2 Σχεδίαση κύκλων και γραμμών με το δάκτυλο

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

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

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



Εικόνα 70 - Εντολή παραγωγής χρώματος

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



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

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

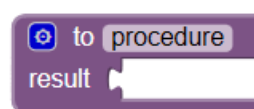


Εικόνα 71 - Παραγωγή υχάιου χρώματος

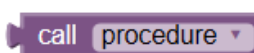
Τέλος, θα υλοποιήσουμε μια διαδικασία που θα παράγει το τυχαίο χρώμα, όπως παραπάνω και θα το επιστρέφει σαν αποτέλεσμα, ώστε να χρησιμοποιηθεί από την εντολή που την κάλεσε.

#### ΥΠΟΠΡΟΓΡΑΜΜΑΤΑ ΠΟΥ ΕΠΙΣΤΡΕΦΟΥΝ ΤΙΜΗ

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

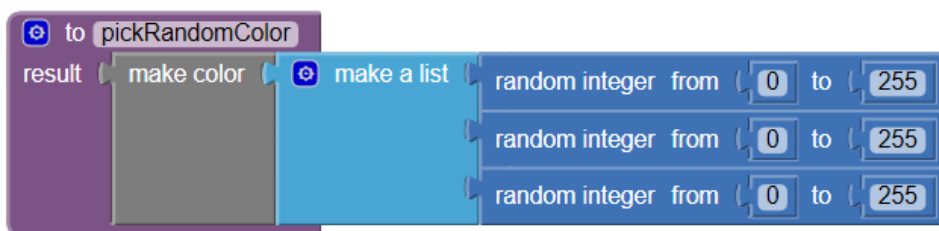


Έχουμε τη δυνατότητα να δημιουργούμε διαδικασίες που επιστρέφουν σαν αποτέλεσμα της εκτέλεσής τους μια τιμή (result).



Κατά αντιστοιχία, όταν τις καλούμε, πρέπει να τις «κουμπώσουμε» απευθείας σε κάποιο πλακίδιο που δέχεται μια τιμή.

Στη συνέχεια δίνεται ολοκληρωμένη η διαδικασία παραγωγής ενός τυχαίου χρώματος.



Εικόνα 72 - Παραγωγή τυχαίου χρώματος από διαδικασία





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

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

```
set Canvas1 . PaintColor to call pickRandomColor
```

Εικόνα 73 - Ορισμός χρώματος σχεδίασης του καμβά

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

```
when Canvas1 . TouchDown
do
  set Canvas1 . PaintColor to call pickRandomColor
  call Canvas1 . DrawCircle
  x get x
  y get y
  r random integer from 1 to 12

when Canvas1 . Dragged
do
  set Canvas1 . PaintColor to call pickRandomColor
  set Canvas1 . LineWidth to random integer from 1 to 4
  call Canvas1 . DrawLine
  x1 get prevX
  y1 get prevY
  x2 get currentX
  y2 get currentY
```

Εικόνα 74 - Σχεδίαση κύκλων ή γραμμών με άγγιγμα ή σύρσιμο του δακτύλου αντίστοιχα.

### Βήμα 3 Καθαρισμός και αλλαγή του χρώματος του καμβά

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

```
when ClearButton . Click
do
  call Canvas1 . Clear
```

Εικόνα 75 - Καθαρισμός του καμβά

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

```
when ClearButton . LongClick
do
```

Εικόνα 76 - Το γεγονός παρατεταμένου πατήματος του κουμπιού Clear



Ποιες εντολές θα κουμπώσουν μέσα στο πλακίδιο **When ClearButton.LongClick**;

## Βήμα 4 Ρίξιμο τυχαίων σταγόνων στον καμβά

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

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

```

when Screen1.Initialize
do
  set DropsLabel.Text to round(DropsSlider.ThumbPosition)

when DropsSlider.PositionChanged
thumbPosition
do
  set DropsLabel.Text to round(DropsSlider.ThumbPosition)
  
```


Εικόνα 77 - Ενημέρωση της ετικέτας που εμφανίζει την ακριβή θέση του μεταβολέα




Κατά την εκκίνηση της εφαρμογής, αλλά και σε οποιαδήποτε αλλαγή της θέσης του μεταβολέα DropsSlider, όρισε σαν κείμενο της ετικέτας DropsLabel, τον αριθμό που αντιστοιχεί στην τρέχουσα θέση του.

Η εντολή **round** στρογγυλοποιεί τις πιθανές δεκαδικές θέσεις που βρίσκεται ο δείκτης του μεταβολέα, στον κοντινότερο ακέραιο αριθμό.

Για παράδειγμα η εντολή

 θα επιστρέψει την τιμή 3,

ενώ η εντολή  θα επιστρέψει την τιμή 4.

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

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

```

set Canvas1.PaintColor to call pickRandomColor
call Canvas1.DrawCircle
  x random integer from 0 to Canvas1.Width
  y random integer from 0 to Canvas1.Height
  r random integer from 1 to 12
  
```

Εικόνα 78 - Τυχαία σχεδίαση ενός κύκλου



Δώσε ένα τυχαίο χρώμα ως χρώμα βαφής για τον καμβά.

Σχεδιάσε έναν κύκλο:

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

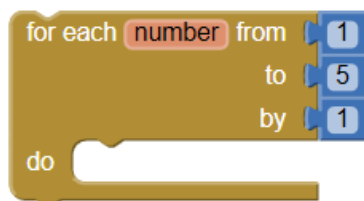
Θα χρειαστεί στη συνέχεια να επαναλάβουμε τις παραπάνω εντολές, ξανά και ξανά, με βάση την τιμή `DropsSlider.ThumbPosition`.



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

## Η ΕΝΤΟΛΗ ΕΠΑΝΑΛΗΨΗΣ FOR

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



Εικόνα 79 - Η εντολή επανάληψης For



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

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

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

Ολοκληρωμένος, ο κώδικας της εφαρμογής μας φαίνεται παρακάτω:

```

when Screen1.Initialize
do
  set DropsLabel.Text to round(DropsSlider.ThumbPosition)

when DropsSlider.PositionChanged
do
  set DropsLabel.Text to round(DropsSlider.ThumbPosition)

when AccSensor.Shaking
do
  call drawRandomCircles
  drops round(DropsSlider.ThumbPosition)

when Canvas1.TouchDown
do
  set Canvas1.PaintColor to call pickRandomColor
  call Canvas1.DrawCircle
  x get x
  y get y
  r random integer from 1 to 12

when Canvas1.Dragged
do
  set Canvas1.PaintColor to call pickRandomColor
  set Canvas1.LineWidth to random integer from 1 to 4
  call Canvas1.DrawLine
  x1 get prevX
  y1 get prevY
  x2 get currentX
  y2 get currentY

when ClearButton.Click
do
  call Canvas1.Clear

when ClearButton.LongClick
do
  if Canvas1.BackgroundColor = 
  then set Canvas1.BackgroundColor to 
  else set Canvas1.BackgroundColor to 

to drawRandomCircles drops
do
  for each number from 1 to get drops
  by 1
  do
    set Canvas1.PaintColor to call pickRandomColor
    call Canvas1.DrawCircle
    x random integer from 0 to Canvas1.Width
    y random integer from 0 to Canvas1.Height
    r random integer from 1 to 12

to pickRandomColor
result make color
make a list
  random integer from 0 to 255
  random integer from 0 to 255
  random integer from 0 to 255
  
```

Εικόνα 80 - Ο κώδικας της εφαρμογής Random Drops

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΛΕΞΙΚΟ

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 81 - Η εφαρμογή "Λεξικό"

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

### Βήμα 1 Δημιουργία του έργου και ρυθμίσεις οθόνης

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



επιλέγουμε το  
**συστατικό**

μεταβάλλουμε τις  
**ιδιότητες**

Screen1

Title : Λεξικό

Background color : Blue

**Designer** Γραφικό Περιβάλλον Εφαρμογής

**Βήμα 2** Προσθήκη των απαραίτητων αρχείων πολυμέσων

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



ανεβάζουμε το αρχείο με  
**όνομα**

Speaker.png

σύντομη  
**περιγραφή**

Εικόνα ηχείου

**Βήμα 3** Προσθήκη αντικειμένων στο γραφικό περιβάλλον

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



από την  
**ομάδα**

Layout

μεταφέρουμε το  
**αντικείμενο**

HorizontalAlignment

του δίνουμε το  
**όνομα**

ButtonsArea

μεταβάλλουμε τις  
**ιδιότητες**

AlignHorizontal : center  
Width : fill Parent  
Height : 80 pixels

UserInterface

ListPicker

WordListPicker

Background Color : Orange  
FontBold : Checked  
FontSize : 18  
Shape : rounded  
Text : Λεξικό  
Text Color : White  
TextAlignment : center

UserInterface

Label

PaddButtonLabel

Text :  
Width : 100 pixels

UserInterface

Button

AddWordButton

Background Color : Green  
FontBold : Checked  
FontSize : 18  
Shape : rounded  
Text : Λεξικό  
Text Color : White  
TextAlignment : center

Layout	HorizontalAlignment	TranslationArea	AlignHorizontal : center Width : fill Parent Height : 80 pixels
UserInterface	Label	GreekLabel	FontBold : checked FontSize : 20
UserInterface	Label	PaddLabel	Text : Width : 50 pixels
UserInterface	Label	GermanLabel	FontBold : checked FontSize : 20
UserInterface	Button	ListenButton	Image : speaker.png Width : 64 pixels Height : 64 pixels Visible : hidden
Layout	HorizontalAlignment	AddWordArea	AlignHorizontal : center Width : fill Parent Height : 100 pixels Visible : hidden
UserInterface	TextBox	GreekTextBox	Hint : Ελληνικά Text : Multiline : checked
UserInterface	TextBox	GermanTextBox	Hint : Γερμανικά Text : Multiline : checked
Layout	HorizontalAlignment	InsertArea	AlignHorizontal : center Width : fill Parent Height : 80 pixels Visible : hidden
UserInterface	Button	InsertButton	Background Color : Yellow FontBold : Checked FontSize : 18 Shape : rounded Text : Αποθήκευση Text Color : White TextAlignment : center
UserInterface	Label	PaddLabel2	Text : Width : 50 pixels
UserInterface	Button	ClearButton	Background Color : Cyan FontBold : Checked FontSize : 18 Shape : rounded

			Text : Ακύρωση
			Text Color : White
			TextAlignment : center
Storage	TinyDB	DictionaryDB	
Media	TextToSpeech	GermanTTS	Country : deu
			Language : DEU

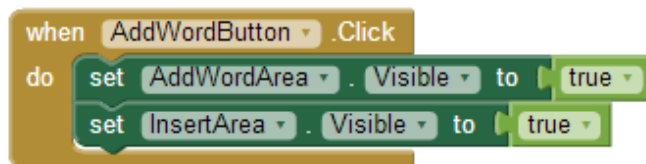
Συγγραφή Προγράμματος

Blocks

#### Βήμα 4 Προσθήκη νέας λέξης στο λεξικό

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

- τα πεδία κειμένου για να γράψει τις λέξεις στα ελληνικά και στα γερμανικά και
- τα κουμπιά για την καταχώρηση της νέας λέξης ή την ακύρωση της ενέργειας.

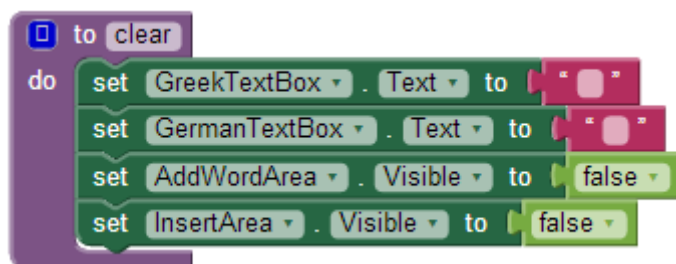


Εικόνα 82 - Εμφάνιση των περιοχών για την καταχώρηση νέας λέξης

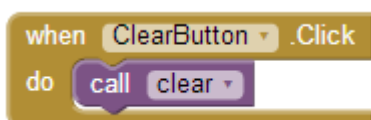
#### Βήμα 5 Ακύρωση της καταχώρισης

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





Εικόνα 83 - Διαδικασία για το καθάρισμα των πεδίων κειμένου και την απόκρυψη των περιοχών εισαγωγής λέξης



Εικόνα 84 - Όταν πατηθεί το κουμπί "Ακύρωση"

## Βήμα 6 Αποθήκευση νέας λέξης στο λεξικό

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



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

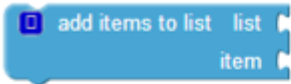

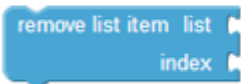
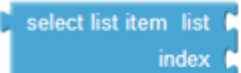
Μια πρώτη σκέψη είναι να χρησιμοποιήσουμε μεταβλητές για την καταχώριση των λέξεων. Αυτό δεν μπορεί να λειτουργήσει σωστά, διότι η μεταβλητή κρατάει μια μόνο τιμή, ενώ η εφαρμογή μας θα πρέπει να διαχειρίζεται πολλές τιμές (όσες και οι λέξεις στο λεξικό).

### ΛΙΣΤΕΣ (LIST)



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

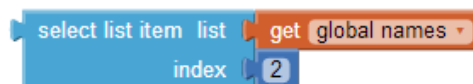
Οι λίστες μας δίνουν πολλές δυνατότητες σε σύγκριση με τις μεταβλητές, αφού μπορούμε εύκολα να εισάγουμε ή να διαγράψουμε τιμές από τη λίστα, να διαλέξουμε μια τυχαία τιμή της ή να αναζητήσουμε τιμές με βάση τη θέση τους. Παρακάτω δίνονται μερικές από τις πιο συχνά χρησιμοποιούμενες εντολές στις λίστες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

	Προσθήκη τιμής στο τέλος της λίστας
	Επιλογή τυχαίας τιμής
	Διαγραφή τιμής από τη θέση .....
	Επιλογή τιμής στη θέση .....



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

Αν για παράδειγμα εκτελέσουμε την εντολή



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

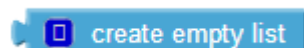
Εικόνα 85 = Παράδειγμα Λίστας

Καταρχήν από την ομάδα Variables δημιουργούμε δύο νέα συμβολικά ονόματα, όπως παρακάτω...



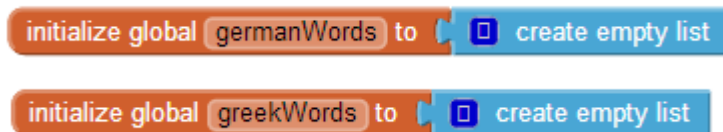
Εικόνα 86 - Τα ονόματα που θα έχουν οι λίστες της εφαρμογής

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



Εικόνα 87 - Εντολή δημιουργίας κενής λίστας

Το τελικό αποτέλεσμα φαίνεται στην παρακάτω εικόνα.

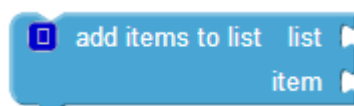


Εικόνα 88 - Δημιουργία δύο κενών λιστών



Δημιούργησε δύο κενές λίστες με ονόματα germanWords και greekWords αντίστοιχα.

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



Εικόνα 89 - Εντολή προσθήκης στοιχείου σε λίστα

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

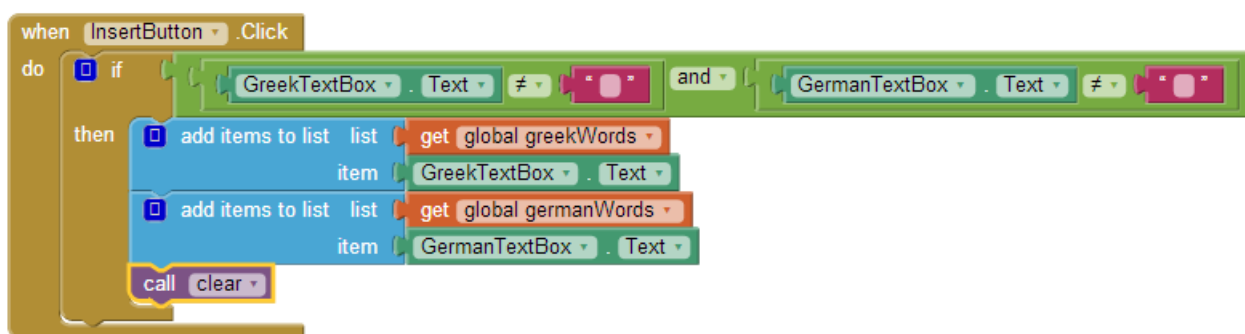


Εικόνα 90 - Προσθήκη λέξης στην αντίστοιχη λίστα

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

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

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



Εικόνα 91 - Άγγιγμα του κουμπιού "Αποθήκευση"



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

## Βήμα 7 Επιλέγοντας από το λεξικό

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



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

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

set WordListPicker . Elements to

Εικόνα 92 - Εντολή ανάθεσης τιμών σε αντικείμενα λίστας

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

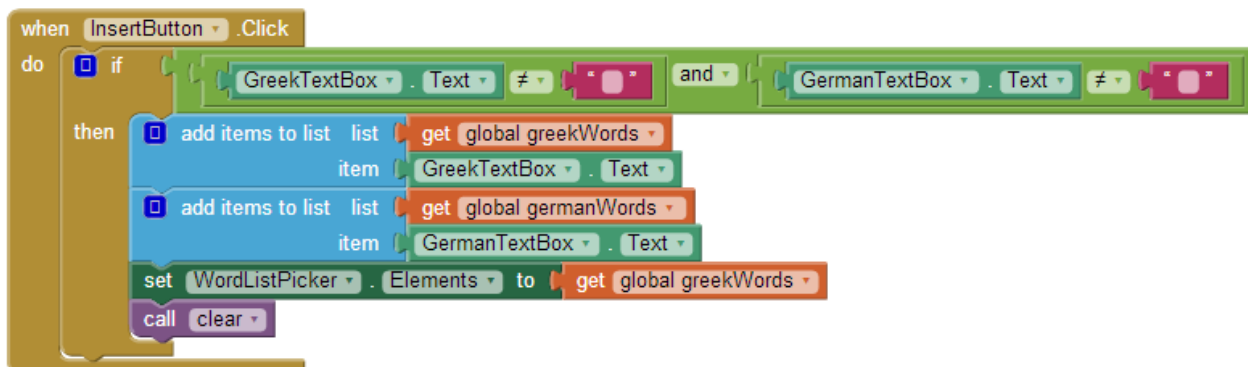
set WordListPicker . Elements to get global greekWords

Εικόνα 93 - Εμφάνιση τιμών από τη λίστα με τις ελληνικές λέξεις

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

when Screen1 . Initialize  
do set WordListPicker . Elements to get global greekWords

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



Εικόνα 94 - Ενημέρωση της λίστας με τη νέα λέξη

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



Εικόνα 95 - Γεγονός που ενεργοποιείται μόλις επιλεγεί τιμή από τη λίστα

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



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

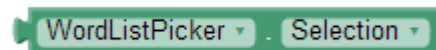


Οι τιμές που εισάγουμε σε μια λίστα ξεχωρίζουν μέσω της θέσης τους στη λίστα. Για παράδειγμα, η τιμή που εισάγουμε πρώτη έχει σαν θέση τον αριθμό 1, η τιμή που βάζουμε δεύτερη έχει σαν θέση τον αριθμό 2 κ.ο.κ.

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

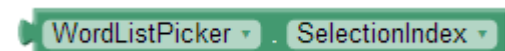
Επομένως αν για παράδειγμα η πρώτη τιμή που καταχωρούμε είναι το «Καλημέρα» και αντίστοιχα το «Guten Morgen» στις 2 λίστες (ελληνικά και γερμανικά αντίστοιχα) τότε βρίσκονται και οι δύο στη θέση 1 μέσα στις λίστες τους.

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



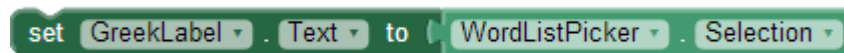
Εικόνα 96 - Η τιμή του επιλεγμένου στοιχείου

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



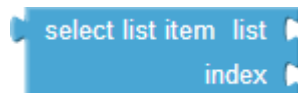
Εικόνα 97 - Η θέση του στοιχείου που επιλέχθηκε

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



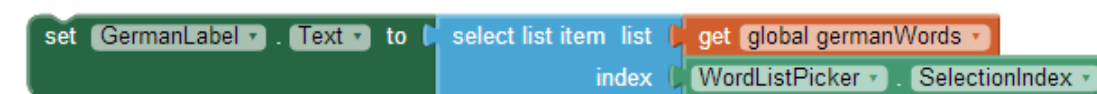
Εικόνα 98 - Τιμή της ετικέτας με την ελληνική λέξη

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



Εικόνα 99 - Εντολή επιλογής τιμής από λίστα

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



Εικόνα 100 - Τιμή της ετικέτας με τη γερμανική λέξη

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

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

```

when WordListPicker .AfterPicking
do
  set GreekLabel .Text to WordListPicker .Selection
  set GermanLabel .Text to select list item list
  index get global germanWords
  WordListPicker .SelectionIndex
  set ListenButton .Visible to true

```

Εικόνα 101 - Γεγονός επιλογής από λίστα



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

## Βήμα 8 Ακούγοντας τη λέξη!

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

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



Το αντικείμενο TextToSpeech (TTS) μετατρέπει ένα κείμενο σε λόγο. Το μόνο που χρειάζεται είναι να του καθορίσουμε τις ιδιότητες χώρα και γλώσσα (Country και Language αντίστοιχα) για να μπορέσει να διαβάσει σωστά το κείμενο που θα του δώσουμε.

Στη συνέχεια κάθε φορά που θέλουμε να διαβάσει ένα κείμενο χρησιμοποιούμε την εντολή

```

call GermanTTS .Speak
message

```

στην οποία κουμπώνουμε το κείμενο

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

```

when ListenButton .Click
do
  call GermanTTS .Speak
  message GermanLabel .Text

```

Εικόνα 102 - Διάβασμα της λέξης από την κινητή μας συσκευή



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

## Βήμα 9 Μόνιμη αποθήκευση στοιχείων στην κινητή συσκευή

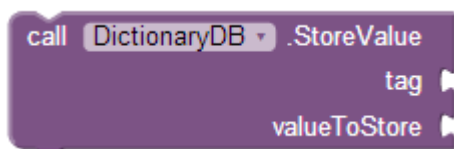
Όταν ο χρήστης αγγίζει το κουμπί Αποθήκευση θα πρέπει η νέα λέξη που έχει γράψει να αποθηκεύεται στην κινητή συσκευή, ώστε να μην χρειάζεται να καταγράφει κάθε φορά όλες τις λέξεις από την αρχή.

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



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

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



Εικόνα 103 - Εντολή αποθήκευσης τιμών στη Βάση Δεδομένων

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

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



Εικόνα 104 - Αποθήκευση λίστας με ελληνικά στη ΒΔ

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





Εικόνα 105 - Αποθήκευση λίστας με γερμανικά στη ΒΔ

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

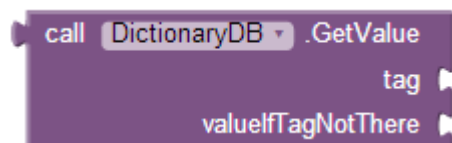


Εικόνα 106 - Αποθήκευση νέας λέξης στη ΒΔ

## Βήμα 10 Ανάκτηση αποθηκευμένων τιμών από την ΒΔ

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

Για να το πετύχουμε θα χρειαστούμε την εντολή...



Εικόνα 107 - Εντολή ανάκτησης τιμών από την ΒΔ

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

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

```

when Screen1.Initialize
do
  set global greekWords to call DictionaryDB.GetValue
    tag "greekWords"
    valueIfTagNotThere create empty list
  set global germanWords to call DictionaryDB.GetValue
    tag "germanWords"
    valueIfTagNotThere create empty list
  set WordListPicker.Elements to get global greekWords

```

Εικόνα 108 - Ενέργειες κατά την εκκίνηση της εφαρμογής



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

Ολοκληρωμένη, η εφαρμογή μας αποτελείται από τις παρακάτω εντολές:

```

initialize global germanWords to create empty list
initialize global greekWords to create empty list

```

```

when Screen1.Initialize
do
  set global greekWords to call DictionaryDB.GetValue
    tag "greekWords"
    valueIfTagNotThere create empty list
  set global germanWords to call DictionaryDB.GetValue
    tag "germanWords"
    valueIfTagNotThere create empty list
  set WordListPicker.Elements to get global greekWords

```

```

when AddWordButton .Click
do
  set AddWordArea .Visible to true
  set InsertArea .Visible to true
    
```

```

when ClearButton .Click
do
  call clear
    
```

```

to clear
do
  set GreekTextBox .Text to ""
  set GermanTextBox .Text to ""
  set AddWordArea .Visible to false
  set InsertArea .Visible to false
    
```

```

when WordListPicker .AfterPicking
do
  set GreekLabel .Text to WordListPicker .Selection
  set GermanLabel .Text to select list item list
  get global germanWords
  index WordListPicker .SelectionIndex
  set ListenButton .Visible to true
    
```

```

when ListenButton .Click
do
  call GermanTTS .Speak
  message GermanLabel .Text
    
```

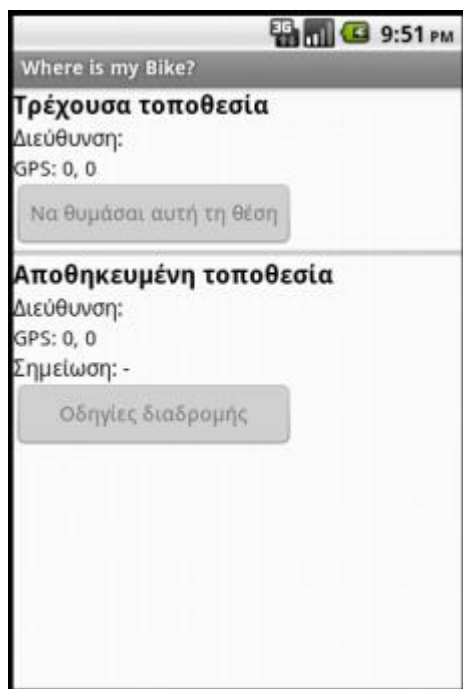
```

when InsertButton .Click
do
  if
    GreekTextBox .Text ≠ "" and GermanTextBox .Text ≠ ""
  then
    add items to list list
    get global greekWords
    item GreekTextBox .Text
    add items to list list
    get global germanWords
    item GermanTextBox .Text
    call DictionaryDB .StoreValue
    tag "greekWords"
    valueToStore get global greekWords
    call DictionaryDB .StoreValue
    tag "germanWords"
    valueToStore get global germanWords
    set WordListPicker .Elements to get global greekWords
    call clear
    
```

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : WHERE IS MY BIKE?

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 109 - Στιγμιότυπο της εφαρμογής κατά την εκκίνησή της

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

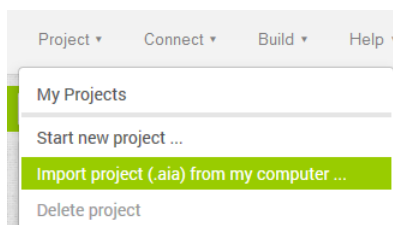
Όποτε το θελήσουμε, πατώντας το κουμπί «Οδηγίες διαδρομής», μεταφερόμαστε στην εφαρμογή Google Maps, όπου αυτόματα, με βάση τα γεωγραφικά στοιχεία, μπορούμε να πλοηγηθούμε στον προορισμό μας, στο σημείο δηλαδή που αποθηκεύσαμε πριν (εκεί που βρίσκεται το ποδήλατό μας).

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

## Βήμα 1 Εισαγωγή πηγαίου κώδικα για δημιουργία νέου project

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



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

Συγγραφή Προγράμματος

Blocks

## Βήμα 2 Εμφάνιση της τρέχουσας θέσης

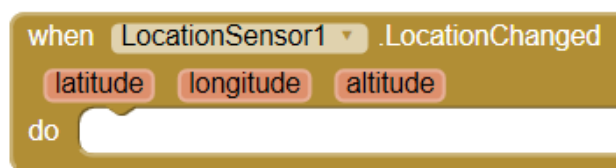
Προκειμένου να αξιοποιήσουμε τις δυνατότητες του GPS στην εφαρμογή μας χρειαζόμαστε το αντικείμενο LocationSensor, το οποίο έχει ήδη προστεθεί στην περιοχή σχεδίασης.

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



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

Θα αξιοποιήσουμε την παρακάτω εντολή ...



Εικόνα 110 - Γεγονός αλλαγής γεωγραφικής θέσης

... η οποία ανιχνεύει οποιαδήποτε αλλαγή στη γεωγραφική θέση της συσκευής. Το γεγονός αυτό πυροδοτείται ακόμα και όταν ο αισθητήρας λάβει σήμα από τους δορυφόρους για

πρώτη φορά, κάτι το οποίο μπορεί να πάρει από μερικά δευτερόλεπτα, μέχρι μερικά λεπτά, ανάλογα με τη συσκευή, τις συνθήκες υπό τις οποίες λειτουργεί και τη θέση που βρίσκεται.



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

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

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

```

when LocationSensor1 . LocationChanged
  latitude longitude altitude
do
  set CurrentAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set GPSCurrentLatLabel . Text to get latitude
  set GPSCurrentLongLabel . Text to get longitude
  set RememberCurrentButton . Enabled to true

```

Εικόνα 111 - Εμφάνιση της τρέχουσας θέσης μας



Όταν ο αισθητήρας ανίχνευσης θέσης ανιχνεύσει οποιαδήποτε αλλαγή της θέσης μας, εμφάνισε στις αντίστοιχες ετικέτες κειμένου την πλήρη διεύθυνση στην οποία βρισκόμαστε, τις ακριβείς συντεταγμένες (μήκος και πλάτος) και ενεργοποίησε το κουμπί για την αποθήκευση των παραπάνω στοιχείων.

### Βήμα 3

#### Αποθήκευση της τρέχουσας θέσης στη βάση δεδομένων

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

- A. να ενημερώνει τις τιμές των ετικετών στο κάτω μέρος της οθόνης και
- B. να αποθηκεύει τα στοιχεία αυτά στη βάση δεδομένων της εφαρμογής, ώστε να είναι διαθέσιμα ακόμα και αν η εφαρμογή τερματιστεί και ξεκινήσει ξανά.

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

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

Οι εντολές που θα χρειαστούν φαίνονται στην παρακάτω εικόνα :

```

when RememberCurrentButton . Click
do
  set RememberAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set GPSRememberLatLabel . Text to GPSCurrentLatLabel . Text
  set GPSRememberLongLabel . Text to GPSCurrentLongLabel . Text
  set ShowDirectionsButton . Enabled to true
  call TinyDB1 . StoreValue
    tag "Address"
    valueToStore CurrentAddressDataLabel . Text
  call TinyDB1 . StoreValue
    tag "Lat"
    valueToStore GPSCurrentLatLabel . Text
  call TinyDB1 . StoreValue
    tag "Long"
    valueToStore GPSCurrentLongLabel . Text
  call Notifier1 . ShowTextDialog
    message "(προαιρετική συμπλήρωση)"
    title "Σημείωση"
    cancelable true

```

Εικόνα 112 - Οι εντολές που εκτελούνται με το πάτημα του κουμπιού αποθήκευσης



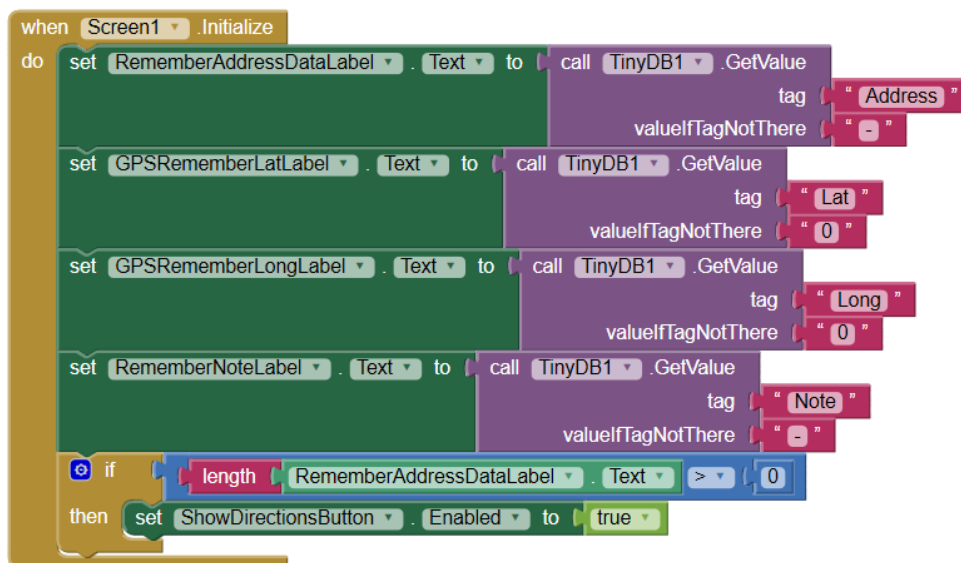
Όταν πατηθεί το κουμπί, ενημέρωσε τις αντίστοιχες ετικέτες, ενεργοποίησε το κουμπί λήψης οδηγιών και αποθήκευσε τις αντίστοιχες τιμές στη βάση δεδομένων (ΒΔ) σε πεδία με όνομα Address, Lat και Long. Επιπλέον, εμφάνισε ένα μήνυμα που να προτρέπει προαιρετικά στο χρήστη να εισάγει μια επιπρόσθετη σημείωση, λ.χ. σχετικά με τη θέση αυτή.

#### Βήμα 4 Ανάκτηση των αποθηκευμένων στοιχείων από τη ΒΔ

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

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

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



Εικόνα 113 - Ενέργειες κατά την εκκίνηση της εφαρμογής



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

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

## Βήμα 5





### Λήψη οδηγιών για την πλοήγηση μέσω Google Maps

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

#### Ο ΕΚΚΙΝΗΤΗΣ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ (ACTIVITY STARTER)



##### Connectivity

-  ActivityStarter
-  BluetoothClient
-  BluetoothServer
-  Web

Το αντικείμενο Activity Starter μας επιτρέπει να ξεκινήσουμε κάποια «δραστηριότητα», όπως:

- Άνοιγμα κάποιας εγκατεστημένης εφαρμογής
- Αναζήτηση στον Παγκόσμιο Ιστό
- Άνοιγμα κάποιας ιστοσελίδας σε φυλλομετρητή
- Άνοιγμα των Χαρτών Google σε συγκεκριμένη θέση.

Οι παράμετροι που θέλουμε να περάσουμε στην δραστηριότητα αυτή καθορίζουν τι ακριβώς και πως θα συμβεί.

Περισσότερα στο: <http://ai2.appinventor.mit.edu/reference/components/connectivity.html>



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

### Designer Γραφικό Περιβάλλον Εφαρμογής

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



από την  
**ομάδα**

μεταφέρουμε το  
**αντικείμενο**

μεταβάλλουμε τις  
**ιδιότητες**

Connectivity

ActivityStarter

Action: android.intent.action.VIEW

ActivityClass: com.google.android.maps.MapActivity

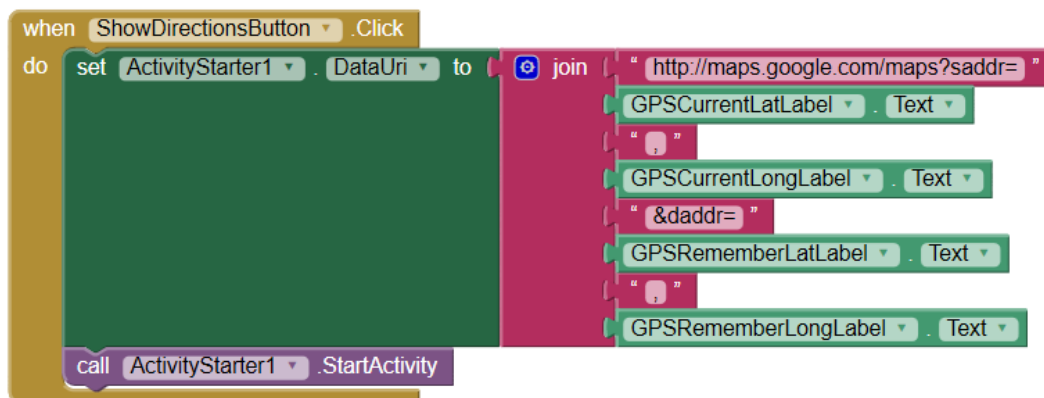
ActivityPackage: com.google.android.apps.maps

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

### Συγγραφή Προγράμματος

### Blocks

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



Όταν πατηθεί το κουμπί ShowDirectionsButton, δώσε στην ιδιότητα DataUri του ActivityStarter1 την τιμή που προκύπτει από την ένωση των αναφερόμενων σταθερών τμημάτων κειμένου και ετικετών.

*(Ουσιαστικά, προκύπτει μια διεύθυνση URL, με τα απαραίτητα στοιχεία για την εκκίνηση της πλοήγησης στην υπηρεσία Google Maps).*

Ξεκίνα τη δραστηριότητα του ActivityStarter1 (με τις δοσμένες παραμέτρους).



Η εφαρμογή μας είναι έτοιμη!

Ώρα να την ελέγξουμε με την κινητή μας συσκευή (ιδανικά σε εξωτερικό χώρο).

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

```

when LocationSensor1 . LocationChanged
  latitude longitude altitude
do
  set CurrentAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set GPSCurrentLatLabel . Text to get latitude
  set GPSCurrentLongLabel . Text to get longitude
  set RememberCurrentButton . Enabled to true

```

```

when RememberCurrentButton . Click
do
  set RememberAddressDataLabel . Text to LocationSensor1 . CurrentAddress
  set GPSRememberLatLabel . Text to GPSCurrentLatLabel . Text
  set GPSRememberLongLabel . Text to GPSCurrentLongLabel . Text
  set ShowDirectionsButton . Enabled to true
  call TinyDB1 . StoreValue
    tag Address
    valueToStore CurrentAddressDataLabel . Text
  call TinyDB1 . StoreValue
    tag Lat
    valueToStore GPSCurrentLatLabel . Text
  call TinyDB1 . StoreValue
    tag Long
    valueToStore GPSCurrentLongLabel . Text
  call Notifier1 . ShowTextDialog
    message (προαιρετική συμπλήρωση)
    title Σημείωση
    cancelable true

```

```

when Screen1 . Initialize
do
  set RememberAddressDataLabel . Text to call TinyDB1 . GetValue
    tag Address
    valueIfTagNotThere -
  set GPSRememberLatLabel . Text to call TinyDB1 . GetValue
    tag Lat
    valueIfTagNotThere 0
  set GPSRememberLongLabel . Text to call TinyDB1 . GetValue
    tag Long
    valueIfTagNotThere 0
  set RememberNoteLabel . Text to call TinyDB1 . GetValue
    tag Note
    valueIfTagNotThere -
  if length RememberAddressDataLabel . Text > 0
  then set ShowDirectionsButton . Enabled to true

```

```

when ShowDirectionsButton . Click
do
  set ActivityStarter1 . DataUri to join
    http://maps.google.com/maps?saddr=
    GPSCurrentLatLabel . Text
    ,
    GPSCurrentLongLabel . Text
    &daddr=
    GPSRememberLatLabel . Text
    ,
    GPSRememberLongLabel . Text
  call ActivityStarter1 . StartActivity

```

## ΕΠΕΚΤΑΣΕΙΣ

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

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



```
when Notifier1 .AfterTextInput  
  response  
do
```

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

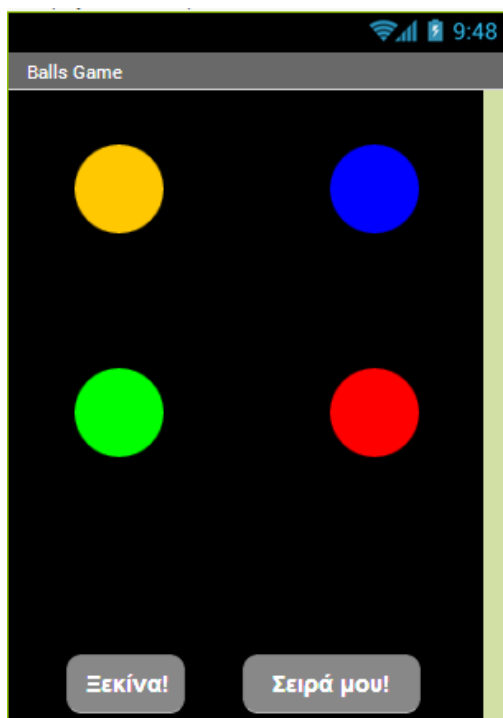
Εφόσον ο χρήστης συμπληρώσει κάποιο κείμενο, αυτό μεταφέρεται μέσω της παραμέτρου `response`. Εάν όμως ο χρήστης πατήσει το κουμπί `Cancel`, το γεγονός επιστρέφει την τιμή `"Cancel"` στην παράμετρο `response`, οπότε καλό είναι να μην αποθηκεύσουμε τη συγκεκριμένη τιμή στη ΒΔ.

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

## ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΜΠΑΛΕΣ ΠΟΥ ΑΝΑΒΟΣΒΗΝΟΥΝ (ΠΑΙΧΝΙΔΙ ΜΝΗΜΗΣ)

### ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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



Εικόνα 114 - Παιχνίδι μνήμης με μπάλες

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

## Βήμα 1 Εισαγωγή έτοιμου έργου

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

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

Συγγραφή Προγράμματος

Blocks

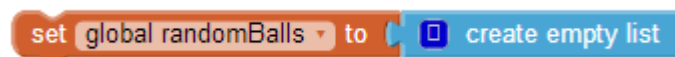
## Βήμα 2 Παραγωγή τυχαίας σειράς για το animation

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

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

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

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



Εικόνα 115 - Καθάρισμα της λίστας με τους τυχαίους αριθμούς

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

```

to prepareRandomBalls
do
  for each number from 1
    to get global level
    by 1
  do
    add items to list list
    item get global randomBalls
    random integer from 1 to 4
  
```

Εικόνα 116 - Η διαδικασία παραγωγής τυχαίων αριθμών



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

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

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

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

```

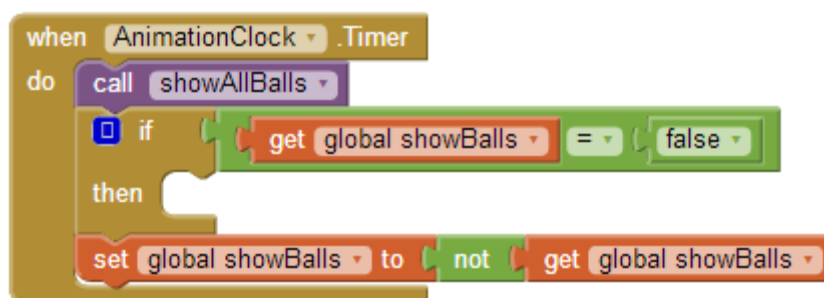
when StartAnimationButton .Click
do
  set global randomBalls to create empty list
  call prepareRandomBalls
  set StartAnimationButton . Enabled to false
  set InteractButton . Enabled to false
  set AnimationClock . TimerEnabled to true
  set global randomBallIndex to 1
  
```

Εικόνα 117 - Όταν πατηθεί το κουμπί "Ξεκίνα"

### Βήμα 3 Αναβοσβήνοντας τις μπάλες

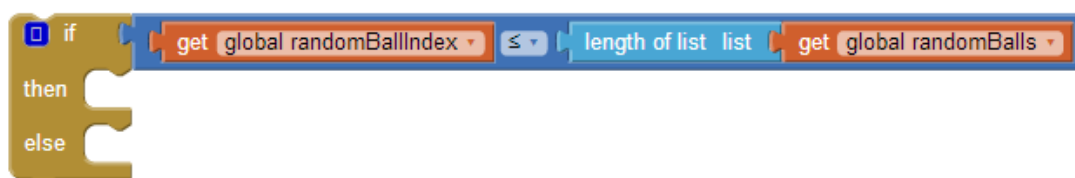
Στη συνέχεια θα περιγράψουμε τις ενέργειες που θα γίνονται κάθε φορά που πυροδοτείται το χρονόμετρο AnimationClock.

Το AnimationClock θα κάνει τα εξής: Αρχικά θα εμφανίζει όλες τις μπάλες. Έπειτα κάθε **δεύτερη** φορά που πυροδοτείται, θα «κρύβει» την μπάλα που πρέπει, με βάση τις τιμές της λίστας randomBalls, Αυτό επιτυγχάνεται με την χρήση μιας λογικής μεταβλητής με όνομα showBalls, η οποία παίρνει διαδοχικά τις τιμές true – false. Όταν η τιμή της είναι true, οι μπάλες παραμένουν όλες ορατές, ενώ όταν η τιμή της είναι false κρύβεται η αντίστοιχη μπάλα. Επίσης, κάθε φορά που πυροδοτείται το AnimationClock η τιμή της λογικής μεταβλητής αντιστρέφεται με τη βοήθεια του λογικού τελεστή not.



Εικόνα 118 - Ενεργοποίηση του AnimationClock

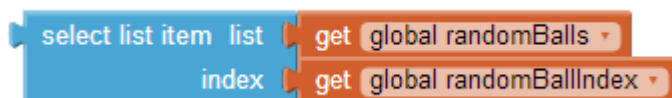
Μέσα στην if, θα εξετάζουμε αν η μεταβλητή randomIndex έχει ξεπεράσει το μέγεθος της λίστας, οπότε πρέπει να σταματήσουμε το animation...



Εικόνα 119 - Έλεγχος για την ολοκλήρωση των στοιχείων της λίστας

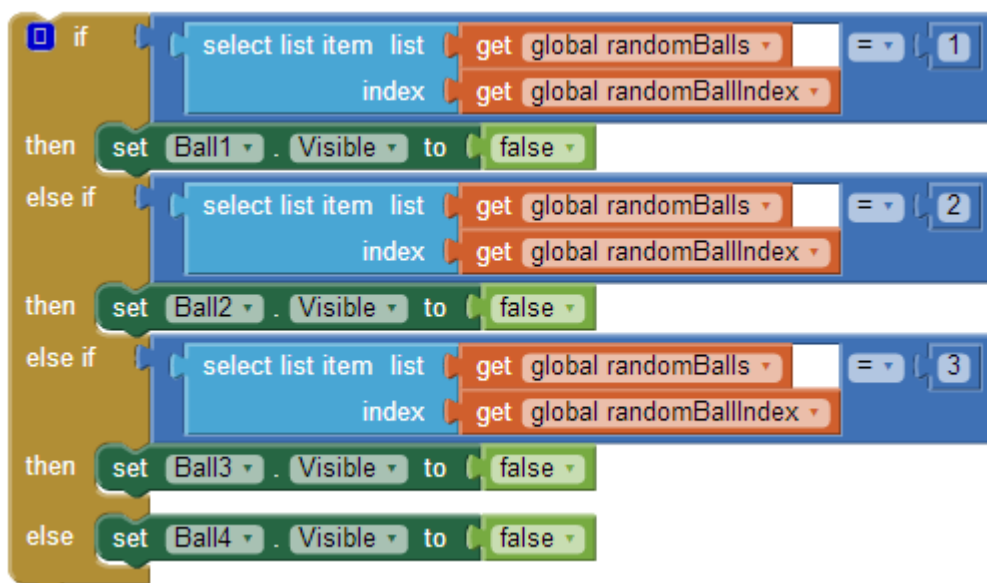
... ή σε διαφορετική περίπτωση να κρύψουμε την κατάλληλη μπάλα.

Για να κρύψουμε τώρα κάποια από τις μπάλες, θα «συμβουλευτούμε» τη λίστα με τους τυχαίους αριθμούς randomBalls και θα κρύψουμε την μπάλα που έχει τον ίδιο αριθμό με το στοιχείο της λίστας που βρίσκεται στη θέση randomIndex.



Εικόνα 120 - Επιλογή στοιχείου της λίστας

Το στοιχείο που παίρνουμε από τη λίστα θα έχει μια από τις τιμές 1 έως 4. Θα εξετάζουμε ποια από όλες τις τιμές έχει, ώστε να κρύβουμε την αντίστοιχη μπάλα, όπως φαίνεται παρακάτω :



Εικόνα 121 - Επιλογή της μπάλας που θα κρυφτεί

Τέλος ο μετρητής randomBallIndex θα αυξάνεται κατά 1.



Εικόνα 122 - Αύξηση του μετρητή

Σε περίπτωση που ο μετρητής randomBallIndex ξεπεράσει το μέγεθος της λίστας τότε το animation πρέπει να τερματιστεί, επομένως θα απενεργοποιήσουμε το AnimationClock και θα ενεργοποιήσουμε τα κουμπιά, ώστε ο παίκτης να μπορεί να συνεχίσει το παιχνίδι. Οι εντολές που αφορούν την ενεργοποίηση του AnimationClock φαίνονται παρακάτω.



```

when AnimationClock.Timer
do
  call showAllBalls
  if (get global showBalls = false)
  then
    if (get global randomBallIndex ≤ length of list list get global randomBalls)
    then
      if (select list item list get global randomBalls index get global randomBallIndex = 1)
      then
        set Ball1.Visible to false
      else if (select list item list get global randomBalls index get global randomBallIndex = 2)
      then
        set Ball2.Visible to false
      else if (select list item list get global randomBalls index get global randomBallIndex = 3)
      then
        set Ball3.Visible to false
      else
        set Ball4.Visible to false
      set global randomBallIndex to (get global randomBallIndex + 1)
    else
      set AnimationClock.TimerEnabled to false
      set StartAnimationButton.Enabled to true
      set InteractButton.Enabled to true
  set global showBalls to not (get global showBalls)
  
```

Εικόνα 123 - Ολοκλήρωση των ενεργειών κατά την ενεργοποίηση του χρονομέτρου



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

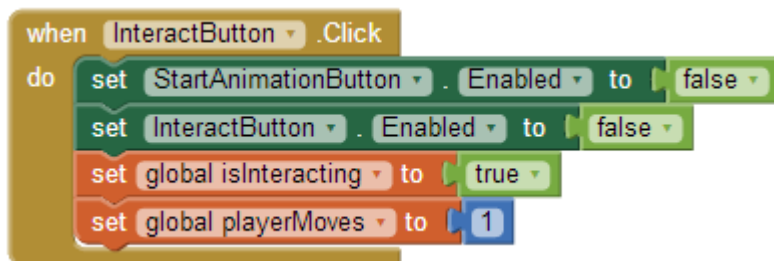
#### Βήμα 4 Ξεκινώντας το παιχνίδι

Όταν ο παίκτης αγγίξει το κουμπί InteractButton θα πρέπει να αγγίξει τις μπάλες με την ίδια σειρά με την οποία κρύφτηκαν.

Το πρώτο που θα κάνουμε είναι να απενεργοποιήσουμε τα δύο κουμπιά και να θέσουμε τις τιμές δύο μεταβλητών της εφαρμογής.

Η πρώτη μεταβλητή με όνομα playerMoves θα μετράει πόσες μπάλες έχει αγγίξει ο παίκτης και θα παίρνει αρχική τιμή 1.

Η δεύτερη μεταβλητή με όνομα `isInteracting` παίρνει τις τιμές `true/false` και την χρησιμοποιούμε για να μας δείχνει αν ο παίκτης βρίσκεται στη φάση της επιλογής (και όχι του `animation`), ώστε να του επιτρέψουμε να αγγίζει τις μπάλες. Οι εντολές που θα μας χρειαστούν φαίνονται παρακάτω...



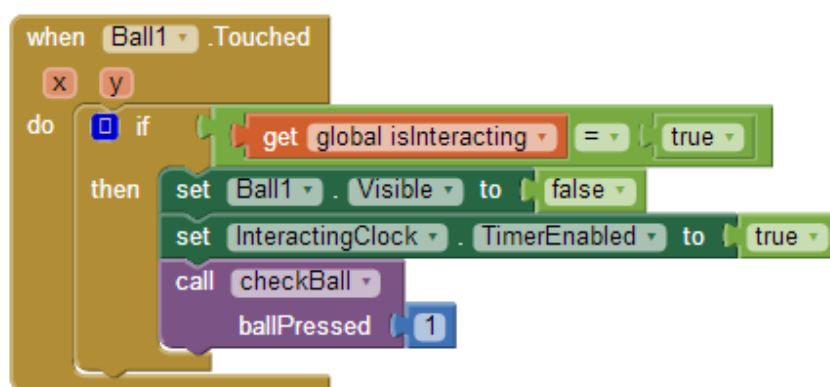
Εικόνα 124 - Το άγγιγμα του κουμπιού έναρξης του παιχνιδιού

## Βήμα 5 Η επιλογή της μπάλας

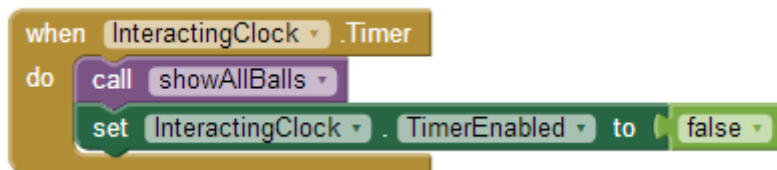
Όταν ο παίκτης αγγίξει μια μπάλα θα γίνουν οι παρακάτω ενέργειες : εφόσον η λογική μεταβλητή `isInteracting` είναι αληθής (`true`) η μπάλα θα κρύβεται για μισό δευτερόλεπτο με τη βοήθεια του χρονόμετρου – ρολογιού `InteractingClock`. Έπειτα θα γίνεται έλεγχος αν η μπάλα που άγγιξε ο παίκτης είναι στη σωστή σειρά με βάση το `animation` που είδε προηγουμένως.

Ο έλεγχος για το αν ο παίκτης άγγιξε τη σωστή μπάλα θα γίνεται από διαδικασία που θα υλοποιήσουμε για το σκοπό αυτό, ας την ονομάσουμε `checkBall`. Η διαδικασία θα παίρνει ως παράμετρο τον αριθμό της μπάλας που πατήθηκε.

Παρακάτω δίνονται οι εντολές που θα εκτελούνται όταν πατηθεί η 1<sup>η</sup> μπάλα καθώς και οι εντολές κατά την πυροδότηση του ρολογιού `InteractingClock`.



Εικόνα 125 - Το άγγιγμα της πρώτης μπάλας



Εικόνα 126 - Η πυροδότηση του χρονομέτρου InteractingClock

## Βήμα 6 Ο έλεγχος της μπάλας

Παρακάτω θα περιγράψουμε τη διαδικασία checkBalls. Η διαδικασία θα ελέγχει αν ο παίκτης διάλεξε τη σωστή μπάλα ή αν έκανε λάθος.

Σε περίπτωση λανθασμένης επιλογής θα αναπαράγει τον ήχο failure.mp3 και θα γίνονται οι απαραίτητες ενέργειες ώστε το παιχνίδι να αρχίζει από την αρχή.



Εικόνα 127 - Η λανθασμένη επιλογή

Σε περίπτωση σωστής επιλογής, η διαδικασία θα ελέγχει αν ο παίκτης ολοκλήρωσε σωστά όλες τις κινήσεις, δηλαδή αν ο αριθμός των σωστών επιλογών του είναι όσα και τα στοιχεία της λίστας με τους τυχαίους αριθμούς. Σε αυτή την περίπτωση θα αναπαράγει τον ήχο success.wav, θα γίνονται οι απαραίτητες ενέργειες ώστε το παιχνίδι να αρχίζει από την αρχή και ο παίκτης θα ανεβαίνει επίπεδο. Διαφορετικά θα αυξάνει τον αριθμό των σωστών επιλογών του παίκτη (μεταβλητή playerMoves).

```

if (get global playerMoves = length of list list get global randomBalls)
then
  set FXPlayer . Source to "success.wav"
  set global level to (get global level + 1)
  call FXPlayer . Start
  set StartAnimationButton . Enabled to true
  set InteractButton . Enabled to false
  set global isInteracting to false
else
  set global playerMoves to (get global playerMoves + 1)

```

Εικόνα 128 - Ενέργειες για τη σωστή επιλογή



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

Παρακάτω δίνεται ο κώδικας της ολοκληρωμένης εφαρμογής.

```

initialize global playerMoves to 1
initialize global showBalls to false
initialize global isInteracting to false
initialize global level to 4
initialize global randomBalls to create empty list
initialize global randomBallIndex to 1

when Screen1 . Initialize
do
  set AnimationClock . TimerEnabled to false
  set InteractButton . Enabled to false
  set StartAnimationButton . Enabled to true
  set InteractingClock . TimerEnabled to false

when StartAnimationButton . Click
do
  set global randomBalls to create empty list
  call prepareRandomBalls
  set StartAnimationButton . Enabled to false
  set InteractButton . Enabled to false
  set AnimationClock . TimerEnabled to true
  set global randomBallIndex to 1

```

```

when AnimationClock.Timer
do
  call showAllBalls
  if (get global showBalls = false)
  then
    if (get global randomBallIndex ≤ length of list list get global randomBalls)
    then
      if (select list item list get global randomBalls index get global randomBallIndex = 1)
      then set Ball1.Visible to false
      else if (select list item list get global randomBalls index get global randomBallIndex = 2)
      then set Ball2.Visible to false
      else if (select list item list get global randomBalls index get global randomBallIndex = 3)
      then set Ball3.Visible to false
      else set Ball4.Visible to false
      set global randomBallIndex to (get global randomBallIndex + 1)
    else
      set AnimationClock.TimerEnabled to false
      set StartAnimationButton.Enabled to true
      set InteractButton.Enabled to true
  end if
  set global showBalls to not get global showBalls

```

```

when InteractButton.Click
do
  set StartAnimationButton.Enabled to false
  set InteractButton.Enabled to false
  set global isInteracting to true
  set global playerMoves to 1

```

```

when Ball1.Touched
x y
do
  if (get global isInteracting = true)
  then
    set Ball1.Visible to false
    set InteractingClock.TimerEnabled to true
    call checkBall ballPressed 1

```

```

when Ball2.Touched
x y
do
  if (get global isInteracting = true)
  then
    set Ball2.Visible to false
    set InteractingClock.TimerEnabled to true
    call checkBall ballPressed 2

```

```

when Ball3 . Touched
do
  if
  then
    set Ball3 . Visible to false
    set InteractingClock . TimerEnabled to true
    call checkBall
      ballPressed 3
  
```

```

when Ball4 . Touched
do
  if
  then
    set Ball4 . Visible to false
    set InteractingClock . TimerEnabled to true
    call checkBall
      ballPressed 4
  
```

```

when InteractingClock . Timer
do
  call showAllBalls
  set InteractingClock . TimerEnabled to false
  
```

```

to prepareRandomBalls
do
  for each number from 1
  to get global level
  by 1
  do
    add items to list list
    item get global randomBalls
    random integer from 1 to 4
  
```

```

to showAllBalls
do
  set Ball1 . Visible to true
  set Ball2 . Visible to true
  set Ball3 . Visible to true
  set Ball4 . Visible to true
  
```

```

to checkBall ballPressed
do
  if
    get ballPressed ≠
    select list item list
    index
    get global randomBalls
    get global playerMoves
  then
    set FXPlayer . Source to "failure.mp3"
    call FXPlayer . Start
    set global randomBalls to create empty list
    set StartAnimationButton . Enabled to true
    set InteractButton . Enabled to false
    set global isInteracting to false
  else
    if
      get global playerMoves = length of list list
      get global randomBalls
    then
      set FXPlayer . Source to "success.wav"
      set global level to get global level + 1
      call FXPlayer . Start
      set StartAnimationButton . Enabled to true
      set InteractButton . Enabled to false
      set global isInteracting to false
    else
      set global playerMoves to get global playerMoves + 1
  
```

## ΕΠΕΚΤΑΣΕΙΣ

1. Ένας τρόπος να βοηθήσουμε τον παίκτη να θυμάται τη σειρά με την οποία αναβόσβησαν οι μπάλες είναι να συνδυάσουμε κάθε μια από αυτές με ένα διαφορετικό ήχο (για παράδειγμα μια μουσική νότα) που θα αναπαράγεται κατά το «σβήσιμο» της μπάλας.

Κάντε τις απαραίτητες ενέργειες για να υλοποιηθεί η παραπάνω λειτουργία. Μπορείτε να βρείτε τους ήχους για τις νότες στο φάκελο BallsFlashing της διεύθυνσης <http://tiny.cc/aifiles>.



Χρησιμοποιήστε μια λίστα για να αποθηκεύσετε τα ονόματα των αρχείων με τις νότες.

2. Κάντε τις απαραίτητες τροποποιήσεις στην εφαρμογή, ώστε ο παίκτης να βλέπει στην οθόνη το επίπεδο στο οποίο βρίσκεται.

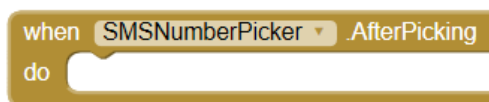
## ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Η εφαρμογή κουμπι Πανικού (βλ. δραστηριότητα για το σπίτι PanicButton μαθήματος 1), δεν είναι παραμετροποιήσιμη. Πιο συγκεκριμένα, ο αριθμός στον οποίο αποστέλλεται το μήνυμα SMS, καθώς και το κείμενο του μηνύματος είναι σταθερά και ενσωματωμένα στο τμήμα σχεδίασης της εφαρμογής (Designer), οπότε μπορούν να τροποποιηθούν μόνο από κάποιον προγραμματιστή και όχι από οποιονδήποτε χρήστη της εφαρμογής.

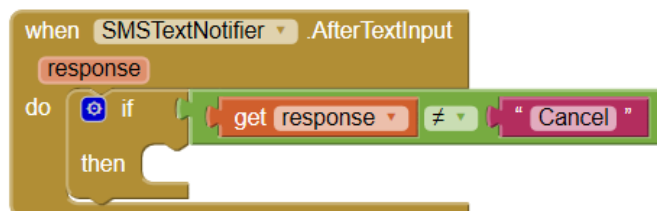
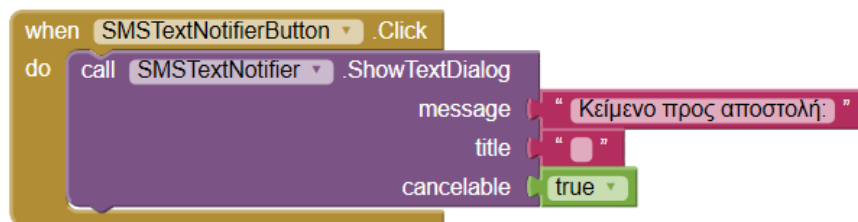
Σας δίνεται έτοιμο ένα μεγάλο μέρος της εφαρμογής (αρχείο PanicButton2.aia από το φάκελο PanicButton2 της διεύθυνσης <http://tiny.cc/aifiles>) ώστε να εσιτιάσετε στις παρακάτω νέες λειτουργίες:

- Να μπορεί ο χρήστης να επιλέξει από τις επαφές της συσκευής τον αριθμό, όπου θα σταλεί το SMS.

Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο παρακάτω γεγονός:

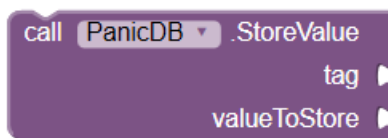


- Να μπορεί ο χρήστης να καθορίζει το κείμενο του μηνύματος που θα αποσταλεί. Η ερώτηση για εισαγωγή του νέου κειμένου προς αποστολή είναι έτοιμη. Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο δεύτερο από τα παρακάτω γεγονότα:



- Ο αριθμός και το κείμενο να αποθηκεύονται σε τοπική Βάση Δεδομένων, ώστε να παραμένουν αποθηκευμένα, ακόμα κι αν ο χρήστης κλείσει την εφαρμογή και μέχρι να τα τροποποιήσει ξανά.

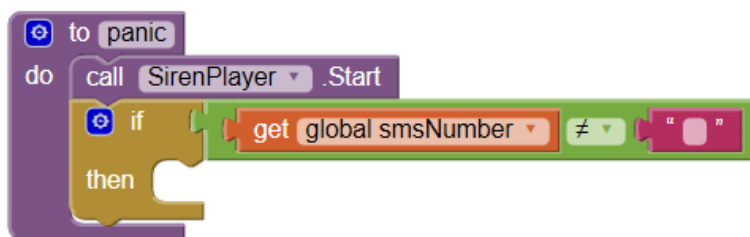
Θα χρειαστείτε εντολές τύπου StoreValue όπως την παρακάτω (αλλά και εντολές που ενημερώνουν τις αντίστοιχες τιμές των μεταβλητών smsNumber και message):



- Στο κείμενο του μηνύματος να προστίθεται αυτόματα στο τέλος η τρέχουσα διεύθυνση και οι συντεταγμένες στις οποίες βρίσκεται ο χρήστης, σύμφωνα με τα δεδομένα που λαμβάνει το GPS της συσκευής.

Θα χρειαστεί να συμπληρώσετε τις κατάλληλες εντολές στο παρακάτω γεγονός:

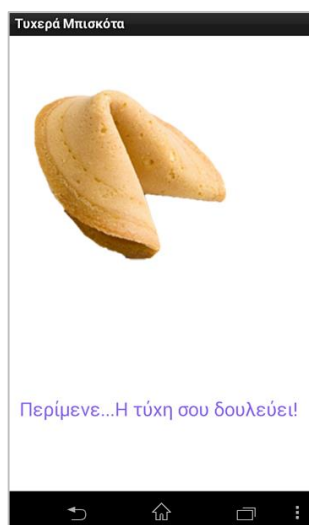




2. [Τυχερά μπισκότα]. Υλοποιήστε μια εφαρμογή που θα λειτουργεί σαν τυχερό μπισκότο. Ο χρήστης θα βλέπει αρχικά στην οθόνη του την εικόνα ενός τυχερού μπισκότου. Όταν την αγγίζει, η εφαρμογή θα παίζει έναν ήχο και μόλις σταματήσει η αναπαραγωγή του ήχου θα εμφανίζει στον χρήστη ένα τυχαίο μήνυμα.



Κατά το διάστημα της αναπαραγωγής του ήχου η εικόνα του μπισκότου δεν θα μπορεί να ξαναπατηθεί. Επίσης θα εμφανίζεται στον χρήστη το μήνυμα «Περίμενε... Η τύχη σου δουλεύει!», το οποίο και θα «αναβοσβήνει», με τυχαίο χρώμα κάθε φορά που εμφανίζεται.



Μόλις ολοκληρωθεί η αναπαραγωγή του ήχου θα εμφανίζεται ένα τυχαίο μήνυμα στον χρήστη και η εικόνα θα αλλάζει σε ανοικτό μπισκότο.

## Τυχερά Μπισκότα



Αποδέξου ότι κάποιες μέρες είσαι το περιστέρι και κάποιες άλλες το άγαλμα



Πλέον όταν ο παίκτης πατήσει πάνω στην εικόνα του ανοικτού μπισκότου, όλα θα ξεκινούν από την αρχή.

Όλα τα αρχεία που θα χρειαστείτε βρίσκονται στη διεύθυνση <http://tiny.cc/aifiles> και το φάκελο Fortune Cookies.

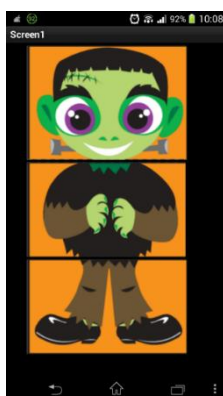
Μπορείτε να βάλετε όποια και όσα μηνύματα θέλετε στα τυχερά μπισκότα. Σε περίπτωση που δεν έχετε έμπνευση χρησιμοποιήστε κάποια από τα παρακάτω!

- Η τύχη σου κρύβεται σε άλλο μπισκότο.
- Ο Κομφούκιος λέει : Ο δρόμος για τα πλούτη κρύβεται στη δουλειά για το σχολείο! Διάβαζε!
- Αποδέξου ότι κάποιες μέρες είσαι το περιστέρι και κάποιες άλλες το άγαλμα.
- Όταν όλοι έρχονται προς το μέρος σου τότε είσαι σε λάθος λωρίδα!
- Θα πεινάσεις πάλι σε 1 ώρα!
- Η ευτυχία κρύβεται στο τελευταίο κουδούνι, Παρασκευή μεσημέρι!
- Σε λίγο καιρός θα είσαι απένταρος...πάλι!
- Η σκληρή δουλειά θα σου ανταποδώσει στο μέλλον. Η τεμπελιά θα σου ανταποδώσει άμεσα!
- Αυτό το τυχερό μπισκότο είναι άτυχο! Δοκίμασε κάποιο άλλο!

3. [Εφαρμογή Mix and Match]. Το Mix and Match είναι ένα διασκεδαστικό παιχνίδι που περιλαμβάνει μια σειρά εικόνων «τεμαχισμένων» σε 3 μέρη, όπως φαίνεται παρακάτω :

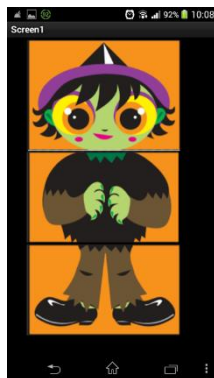


Κατά την εκκίνηση της εφαρμογής προβάλλεται στην οθόνη μια από τις σωστές εικόνες (η οποία επιλέγεται τυχαία κάθε φορά που ξεκινάει η εφαρμογή).



Στη συνέχεια ο παίκτης έχει 3 επιλογές :

- Να σύρει το δάκτυλό του σε ένα από τα 3 τμήματα (κεφάλι, μέση, πόδια), οπότε ΜΟΝΟ το συγκεκριμένο τμήμα εμφανίζει την επόμενη εικόνα.



- Να αγγίξει την οθόνη, οπότε και τα 3 τμήματα μετακινούνται στην επόμενη κατά σειρά εικόνα ταυτόχρονα.



- Να κινήσει τη συσκευή του πέρα δώθε (accelerometer), οπότε και για τα 3 τμήματα επιλέγεται μια διαφορετική τυχαία εικόνα. Δηλαδή, άλλη τυχαία εικόνα για το κεφάλι, άλλη για το σώμα και άλλη για τα πόδια.



Εννοείται ότι όταν ο χρήστης φτάνει στην τελευταία εικόνα της λίστας θα επιστρέφει πάλι στην πρώτη εικόνα. Μην ξεχάσετε να κάνετε τους απαραίτητους ελέγχους!

Στη διεύθυνση <http://tiny.cc/aifiles> και στο φάκελο Mix and Match θα βρείτε το έργο mixAndMatch.aia το οποίο και θα εισάγετε στο περιβάλλον του App Inventor. Εκτός από το έτοιμο γραφικό περιβάλλον υπάρχουν και 3 έτοιμες λίστες. Μια που περιλαμβάνει τις εικόνες για το κεφάλι, μια που περιλαμβάνει τις εικόνες για το σώμα και μια που περιλαμβάνει τις εικόνες για τα πόδια.

Χρησιμοποιήστε κατάλληλα τις λίστες αυτές για να ολοκληρώσετε την εφαρμογή.

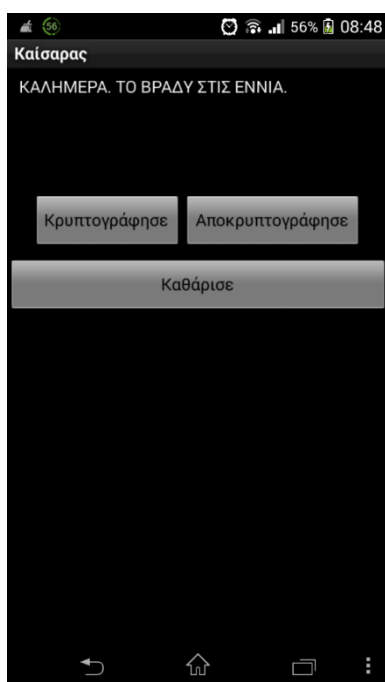
4. Περιγράψτε, σχεδιάστε και υλοποιήστε μια εφαρμογή της αρεσκείας σας, που θα κάνει χρήση του Αισθητήρα Τοποθεσίας (Location Sensor).

5. Η κρυπτογράφηση είναι ένας τρόπος «αλλοίωσης» ενός μηνύματος, ώστε να είναι κατανοητό μόνο από τον αποστολέα και τον παραλήπτη και όχι από άλλους.

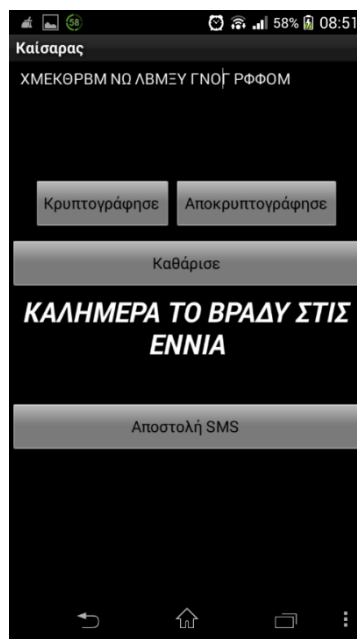
Ένας απλός τρόπος κρυπτογράφησης είναι να αντικαταστήσουμε κάθε γράμμα μιας πρότασης με κάποιο άλλο γράμμα που έχουμε συμφωνήσει από πριν με τον παραλήπτη.

Για παράδειγμα, η πρόταση «ΡΑΝΤΕΒΟΥ ΣΤΗΝ ΠΛΑΤΕΙΑ ΤΟ ΑΠΟΓΕΥΜΑ» θα μπορούσε να μετατραπεί σε «ΥΔΠΧΘΣΨ ΦΧΚΠ ΤΞΑΧΘΜΔ ΧΣ ΔΤΣΖΘΨΟΔ», μετακινώντας κάθε γράμμα της κατά 3 θέσεις στο αλφάβητο. Δηλαδή το γράμμα «Ρ» αντικαθίσταται από το γράμμα «Φ», το γράμμα «Α» αντικαθίσταται από το γράμμα «Δ» κ.ο.κ.

Ο πρώτος που χρησιμοποίησε την παραπάνω μέθοδο κρυπτογράφησης ήταν ο Ιούλιος Καίσαρας για το λόγο αυτό ονομάζεται αλγόριθμος του Καίσαρα.



Θα υλοποιήσουμε μια εφαρμογή ή οποία θα δίνει στον χρήστη της τη δυνατότητα να γράψει ένα μήνυμα, να το κρυπτογραφήσει και στη συνέχεια να το αποστείλει με SMS. Επίσης θα του δίνει τη δυνατότητα να γράψει ένα κρυπτογραφημένο μήνυμα και στη συνέχεια να ζητήσει την αποκρυπτογράφησή του.



Στη διεύθυνση <http://tiny.cc/aifiles> και στο φάκελο Encryption θα βρείτε το έργο Encryption.aia το οποίο και θα εισάγετε στο περιβάλλον του App Inventor. Εκτός από το έτοιμο γραφικό περιβάλλον υπάρχουν και 2 έτοιμες λίστες. Μια που περιλαμβάνει τα κεφαλαία γράμματα του ελληνικού αλφαβήτου με την κανονική τους σειρά και μια που περιλαμβάνει τα κεφαλαία γράμματα του ελληνικού αλφαβήτου με μπερδεμένη σειρά.

Χρησιμοποιήστε κατάλληλα τις λίστες αυτές για να ολοκληρώσετε την εφαρμογή.



Για να σπάσετε μια λέξη ή πρόταση σε μεμονωμένους χαρακτήρες θα χρειαστείτε τις παρακάτω εντολές.

```

for each number from 1
  to length get global plainText
  by 1
do set global charToEncrypt to segment text get global plainText
  start get number
  length 1
    
```

6. Ολοκληρώστε τη μισοτελειωμένη εφαρμογή του κλασικού παιχνιδιού της κρεμάλας. Το περιβάλλον της εφαρμογής είναι έτοιμο, όπως επίσης και ένα μεγάλο μέρος του κώδικα. Επεξεργαστείτε το έργο HangMan.aia από το φάκελο Hangman στη διεύθυνση <http://tiny.cc/aifiles> και υλοποιήστε τη λειτουργικότητα που λείπει, ουσιαστικά τη διαδικασία CheckLetter (letter), που καλείται για κάθε γράμμα που επιλέγεται από τον παίκτη:

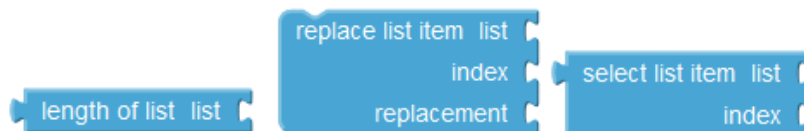
Όταν επιλεγεί ένα γράμμα, να ελέγχεται αν υπάρχει στην κρυμμένη λέξη. Όπου υπάρχει, αντικαθίστανται οι παύλες με το γράμμα αυτό. Αν το γράμμα δεν βρεθεί, το γράμμα εμφανίζεται στην ετικέτα με τα λάθος γράμματα και η κρεμάλα σταδιακά προχωράει, αλλάζοντας

εικόνα. Το παιχνίδι τελειώνει όταν έχουν γίνει 7 λάθη ή όταν βρεθεί η λέξη. Σε κάθε περίπτωση, στο τέλος εμφανίζονται μηνύματα νίκης ή ήττας με τη λέξη που δεν βρέθηκε και αντίστοιχοι ήχοι. Όλα τα σχετικά αρχεία έχουν προστεθεί ήδη στην εφαρμογή.

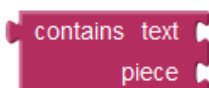
Η λίστα *Word* περιέχει τις λέξεις που επιλέγει τυχαία ο αλγόριθμος και την οποία μπορείτε να επεξεργαστείτε. Η λίστα *LettersList* δημιουργείται αυτόματα και περιέχει τα γράμματα της λέξης που επιλέγεται, ενώ η λίστα *HiddenList* περιέχει αντίστοιχα τις παύλες που αντιστοιχούν στα γράμματα της κρυμμένης λέξης.



Θα χρειαστείτε εντολές που σχετίζονται με λίστες, όπως τις ακόλουθες ...



... αλλά και το παρακάτω πλακίδιο που ελέγχει αν ένα γράμμα ή λέξη (piece) περιέχεται στο κείμενο (text).



ΠΑΡΑΡΤΗΜΑ – ΛΥΣΕΙΣ ΤΩΝ ΑΣΚΗΣΕΩΝ

WHERE IS MY BIKE? – ΕΠΕΚΤΑΣΗ ΕΛΕΓΧΟΥ & ΑΠΟΘΗΚΕΥΣΗΣ ΣΗΜΕΙΩΣΗΣ ΣΤΗ ΒΔ

```

when Notifier1 .AfterTextInput
  response
  do
    if (get response) ≠ "Cancel"
    then
      set RememberNoteLabel .Text to (get response)
    else
      set RememberNoteLabel .Text to "-"
    call TinyDB1 .StoreValue
      tag "Note"
      valueToStore RememberNoteLabel .Text
  
```

ΚΟΥΜΠΙ ΠΑΝΙΚΟΥ ΜΕ ΔΥΝΑΤΟΤΗΤΑ ΕΠΙΛΟΓΗΣ & ΑΠΟΘΗΚΕΥΣΗΣ SMS & ΑΡΙΘΜΟΥ

```

when Screen1 .Initialize
  do
    call updateValues
    initialize global message to ""
    initialize global smsNumber to ""

when PanicButton .Click
  do
    call panic

when PanicButton .LongClick
  do
    call SirenPlayer .Stop

to updateValues
  do
    set global smsNumber to call PanicDB .GetValue
      tag "sms"
      valueIfTagNotThere ""
    set global message to call PanicDB .GetValue
      tag "msg"
      valueIfTagNotThere "Βοήθεια βρίσκομαι σε κίνδυνο! Η θέση μου είναι: "
    set SMSNumberPicker .Text to (get global smsNumber)
    set SMSTextNotifierButton .Text to (get global message)

when SetupButton .Click
  do
    if (OptionsArea .Visible)
    then
      set OptionsArea .Visible to false
    else
      set OptionsArea .Visible to true

when SMSNumberPicker .AfterPicking
  do
    set global smsNumber to SMSNumberPicker .PhoneNumber
    set SMSNumberPicker .Text to (get global smsNumber)
    call PanicDB .StoreValue
      tag "sms"
      valueToStore (get global smsNumber)

when SMSTextNotifierButton .Click
  do
    call SMSTextNotifier .ShowTextDialog
      message "Κείμενο προς αποστολή:"
      title ""
      cancelable true

when SMSTextNotifier .AfterTextInput
  response
  do
    if (get response) ≠ "Cancel"
    then
      set global message to (get response)
      set SMSTextNotifierButton .Text to (get global message)
      call PanicDB .StoreValue
        tag "msg"
        valueToStore (get global message)

when ClearButton .Click
  do
    call PanicDB .ClearAll
    call updateValues

to panic
  do
    call SirenPlayer .Start
    if (get global smsNumber) ≠ ""
    then
      set SMSSender .PhoneNumber to (get global smsNumber)
      set SMSSender .Message to join
        (get global message)
        MyLocation .CurrentAddress
        " Lat: "
        MyLocation .Latitude
        " Long: "
        MyLocation .Longitude
      call SMSSender .SendMessage
  
```



## ΜΠΑΛΕΣ ΠΟΥ ΑΝΑΒΟΣΒΗΝΟΥΝ (ΠΑΙΧΝΙΔΙ ΜΝΗΜΗΣ) - ΕΠΕΚΤΑΣΗ

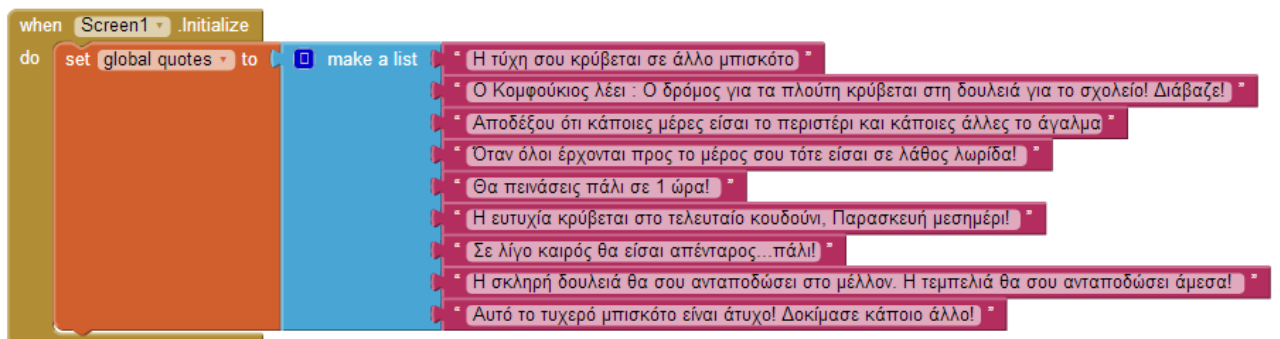


Εικόνα 129 - Αντίστοιχες εντολές θα προστεθούν σε κάθε μπάλα



Εικόνα 130 - Εντολή αλλαγής ετικέτας όταν αλλάζει η μεταβλητή level

## ΤΥΧΕΡΑ ΜΠΙΣΚΟΤΑ



```

initialize global showLabel to true
initialize global isOpen to false
initialize global quotes to create empty list

when CookieImageSprite .Touched
  x y
  do
    if get global isOpen = false
    then
      call CookiePlayer .Start
      call CookiePlayer .Vibrate
        milliseconds 500
      set global isOpen to true
      set MessageLabel .Text to "Περίμενε...Η τύχη σου δουλεύει!"
      set CookieImageSprite .Enabled to false
      set FortuneClock .TimerEnabled to true
    else
      set CookieImageSprite .Picture to "fortune_cookie.png"
      set global isOpen to false
      set MessageLabel .Text to ""

when CookiePlayer .Completed
do
  set FortuneClock .TimerEnabled to false
  set MessageLabel .Visible to true
  set CookieImageSprite .Picture to "fortunecookieopen2.png"
  set MessageLabel .Text to pick a random item list get global quotes
  set CookieImageSprite .Enabled to true

when FortuneClock .Timer
do
  if get global showLabel = true
  then
    set MessageLabel .TextColor to make color
      make a list
      random integer from 0 to 255
      random integer from 0 to 255
      random integer from 0 to 255
    set MessageLabel .Visible to true
  else
    set MessageLabel .Visible to false
  set global showLabel to not get global showLabel

```

## MIX AND MATCH

```

initialize global tops to make a list
  "dracula-top.png"
  "alien-top.png"
  "frankenstein-top.png"
  "witch-top.png"
  "wolfman-top.png"

initialize global middles to make a list
  "dracula-middle.png"
  "alien-middle.png"
  "frankenstein-middle.png"
  "witch-middle.png"
  "wolfman-middle.png"

initialize global bottoms to make a list
  "dracula-bottom.png"
  "alien-bottom.png"
  "frankenstein-bottom.png"
  "witch-bottom.png"
  "wolfman-bottom.png"

initialize global correctimage to random integer from 1 to 5

initialize global currentTops to 0

initialize global currentMiddles to 0

initialize global currentBottoms to 0

```

```

to refreshImages
do
  if get global currentTops > 5
  then set global currentTops to 1

  if get global currentMiddles > 5
  then set global currentMiddles to 1

  if get global currentBottoms > 5
  then set global currentBottoms to 1

  set TopImageSprite . Picture to select list item list get global tops
  index get global currentTops

  set MiddleImageSprite . Picture to select list item list get global middles
  index get global currentMiddles

  set BottomImageSprite . Picture to select list item list get global bottoms
  index get global currentBottoms

```

```

when Screen1.Initialize
do
  set global currentTops to get global correctimage
  set global currentMiddles to get global correctimage
  set global currentBottoms to get global correctimage
  call refreshImages

when ShakingSensor.Shaking
do
  set global currentTops to random integer from 1 to 5
  set global currentMiddles to random integer from 1 to 5
  set global currentBottoms to random integer from 1 to 5
  call refreshImages

when Canvas1.Touched
  x y touchedSprite
do
  set global currentTops to get global currentTops + 1
  set global currentMiddles to get global currentMiddles + 1
  set global currentBottoms to get global currentBottoms + 1
  call refreshImages

when TopImageSprite.Flung
  x y speed heading xvel yvel
do
  set global currentTops to get global currentTops + 1
  call refreshImages

when MiddleImageSprite.Flung
  x y speed heading xvel yvel
do
  set global currentMiddles to get global currentMiddles + 1
  call refreshImages

when BottomImageSprite.Flung
  x y speed heading xvel yvel
do
  set global currentBottoms to get global currentBottoms + 1
  call refreshImages
  
```

## ΚΡΥΠΤΟΓΡΑΦΙΑ

```

initialize global cryptLetters to create empty list
initialize global letters to create empty list
initialize global listKeys to create empty list
initialize global key to 4
initialize global letterIndex to 0
initialize global encryptedChar to ""
initialize global plainText to ""
initialize global charToEncrypt to ""
  
```

```

when Screen1.Initialize
do
  call fillLettersList
  call fillLettersCryptList

when ClearButton.Click
do
  set PlainMessageTextBox.Text to ""
  set HorizontalArrangement4.Visible to false
  set PhoneNumberPicker1.Visible to false
  
```

```

to encryptMessage
do
  set EncryptedMessageLabel.Text to ""
  for each number from 1
    to length get global plainText
    by 1
  do
    set global charToEncrypt to segment text
      uppercase get global plainText
      start get number
      length 1
    if is in list? thing get global charToEncrypt
      list get global letters
    then
      set EncryptedMessageLabel.Text to join
        EncryptedMessageLabel.Text
        select list item list get global cryptLetters
        index index in list thing get global charToEncrypt
        list get global letters
    else
      set EncryptedMessageLabel.Text to join
        EncryptedMessageLabel.Text
        get global charToEncrypt
  
```

```

to decryptMessage
do
  set EncryptedMessageLabel . Text to ""
  for each number from 1
    to length get global plainText
    by 1
  do
    set global charToEncrypt to segment text (uppercase get global plainText)
      start get number
      length 1
    if is in list? thing (get global charToEncrypt)
      list (get global cryptLetters)
    then
      set EncryptedMessageLabel . Text to join EncryptedMessageLabel . Text
        select list item list (get global letters)
        index index in list thing (get global charToEncrypt)
        list (get global cryptLetters)
    else
      set EncryptedMessageLabel . Text to join EncryptedMessageLabel . Text
        get global charToEncrypt
  
```

```

when EncryptButton .Click
do
  if trim PlainMessageTextBox . Text ≠ ""
  then
    set global plainText to trim PlainMessageTextBox . Text
    call encryptMessage
    set HorizontalArrangement4 . Visible to true
    set PhoneNumberPicker1 . Visible to true
  
```

```

when DecryptButton .Click
do
  set PhoneNumberPicker1 . Visible to false
  if trim PlainMessageTextBox . Text ≠ ""
  then
    set global plainText to trim PlainMessageTextBox . Text
    set HorizontalArrangement4 . Visible to true
    call decryptMessage
  
```

```

when PhoneNumberPicker1 .AfterPicking
do
  set SMSTexting . Message to EncryptedMessageLabel . Text
  set SMSTexting . PhoneNumber to PhoneNumberPicker1 . PhoneNumber
  call SMSTexting .SendMessage
  
```

```

to fillLettersList
do
  add items to list list get global letters
  item " A "
  add items to list list get global letters
  item " B "
  add items to list list get global letters
  item " Γ "
  add items to list list get global letters
  item " Δ "
  add items to list list get global letters
  item " E "
  add items to list list get global letters
  item " Z "
  add items to list list get global letters
  item " H "
  add items to list list get global letters
  item " Θ "
  add items to list list get global letters
  item " I "
  add items to list list get global letters
  item " K "
  add items to list list get global letters
  item " Λ "
  add items to list list get global letters
  item " M "
  add items to list list get global letters
  item " N "
  add items to list list get global letters
  item " Ξ "
  add items to list list get global letters
  item " O "
  add items to list list get global letters
  item " Π "
  add items to list list get global letters
  item " P "
  add items to list list get global letters
  item " Σ "
  add items to list list get global letters
  item " T "
  add items to list list get global letters
  item " Y "
  add items to list list get global letters
  item " Φ "
  add items to list list get global letters
  item " X "
  add items to list list get global letters
  item " Ψ "
  add items to list list get global letters
  item " Ω "
  
```

```

to fillLettersCryptList
do
  add items to list list get global cryptLetters
  item " M "
  add items to list list get global cryptLetters
  item " Λ "
  add items to list list get global cryptLetters
  item " H "
  add items to list list get global cryptLetters
  item " Ξ "
  add items to list list get global cryptLetters
  item " P "
  add items to list list get global cryptLetters
  item " I "
  add items to list list get global cryptLetters
  item " K "
  add items to list list get global cryptLetters
  item " T "
  add items to list list get global cryptLetters
  item " O "
  add items to list list get global cryptLetters
  item " X "
  add items to list list get global cryptLetters
  item " E "
  add items to list list get global cryptLetters
  item " Θ "
  add items to list list get global cryptLetters
  item " Φ "
  add items to list list get global cryptLetters
  item " Σ "
  add items to list list get global cryptLetters
  item " Ω "
  add items to list list get global cryptLetters
  item " Z "
  add items to list list get global cryptLetters
  item " B "
  add items to list list get global cryptLetters
  item " Γ "
  add items to list list get global cryptLetters
  item " N "
  add items to list list get global cryptLetters
  item " Y "
  add items to list list get global cryptLetters
  item " Ψ "
  add items to list list get global cryptLetters
  item " Δ "
  add items to list list get global cryptLetters
  item " A "
  add items to list list get global cryptLetters
  item " Π "
  
```

## HANGMAN

initialize global Word to " "

initialize global Mistakes to 0

initialize global correctLetters to 0

initialize global LettersList to create empty list

initialize global HiddenList to create empty list

when Screen1.Initialize

do set LetterListPicker.Enabled to false

to PickRandomWord

do set global Word to pick a random item list make a list

- " ΠΕΙΡΑΤΗΣ "
- " ΕΡΓΑΣΤΗΡΙΟ "
- " ΚΑΛΑΜΑΚΙ "
- " ΑΛΓΟΡΙΘΜΟΣ "
- " ΚΕΛΕΠΟΥΡΙ "

when StartButton.Click

do call PickRandomWord

set global Mistakes to 0

set global correctLetters to 0

set global HiddenList to create empty list

set global LettersList to create empty list

set LetterListPicker.Enabled to true

for each number from 1 to length get global Word by 1

do add items to list list get global LettersList

item segment text get global Word

start get number

length 1

add items to list list get global HiddenList

item " "

set WrongLettersLabel.Text to " "

set HiddenWordLabel.Text to get global HiddenList

set HangmanImage.Picture to " 1.png "

set MessageLabel.Text to " Θα καταφέρεις να βρεις τη λέξη. "

when LetterListPicker.AfterPicking

do call CheckLetter

Letter LetterListPicker.Selection



```

to CheckLetter Letter
do
  for each i from 1
  to length of list list get global LettersList
  by 1
  do
    if
    select list item list get global LettersList = get Letter
    index get i
    then
      replace list item list get global HiddenList
      index get i
      replacement get Letter
      set global correctLetters to get global correctLetters + 1
    if
    not contains text get global Word
    piece get Letter
    then
      set global Mistakes to get global Mistakes + 1
      set WrongLettersLabel . Text to join WrongLettersLabel . Text
      " "
      get Letter
      set HangmanImage . Picture to join get global Mistakes
      ".png"
      set HiddenWordLabel . Text to get global HiddenList
      if
      get global Mistakes = 7
      then
        set LetterListPicker . Enabled to false
        set MessageLabel . Text to join " Η λέξη ήταν. "
        get global Word
        set SoundFXPlayer . Source to " cartoon_hop.mp3 "
        call SoundFXPlayer .Start
      else if
      get global correctLetters = length get global Word
      then
        set LetterListPicker . Enabled to false
        set MessageLabel . Text to " ΚΕΡΔΙΣΕΣ! "
        set SoundFXPlayer . Source to " yee_ha.mp3 "
        call SoundFXPlayer .Start
  
```

## ΠΗΓΕΣ

Το πολυμεσικό υλικό που χρησιμοποιείται στις αναφερόμενες εφαρμογές διατίθεται δωρεάν από τον ιστότοπο: <http://www.freesfx.co.uk/>

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 «Ζωγραφική με τα δάκτυλα» βασίζεται και αποτελεί επέκταση του «Φύλλου εργασίας: FingerPainting – Προγραμματισμός Εφαρμογής με το AppInventor. Γιώργος Μπουκέας, Γιώργος Χατζηνικολάκης. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2013».

Η δραστηριότητα MoleMash αποτελεί επέκταση της αντίστοιχης δραστηριότητας από την επίσημη ιστοσελίδα του App Inventor: <http://beta.appinventor.mit.edu/learn/tutorials/>

Η δραστηριότητα Random Drops βασίζεται στο έργο “Jackson Pollock Splatter Painting” για Scratch από το χρήστη gogreen19: <http://scratch.mit.edu/projects/359758/>

Η δραστηριότητα Where is my Bike? Βασίζεται στο tutorial της επίσημης ιστοσελίδας του App Inventor “Android, Where’s My Car”: <http://beta.appinventor.mit.edu/learn/tutorials/whereismy-car/whereismycar.html>

Η δραστηριότητα Mix and Match βασίζεται σε σχετικό έργο από το Wellesley College: <https://sites.google.com/site/wellesleycs117fall11/>

Η δραστηριότητα με τις μπάλες που αναβοσβήνουν βασίζεται σε σχετικό έργο από το Trinity College: <http://turing.cs.trincoll.edu/~ram/cpsc110/>

Η δραστηριότητα HangMan βασίζεται σε αντίστοιχη εργασία του τμήματος πληροφορικής του Πανεπιστημίου της Alabama ([http://cs104.cs.ua.edu/index\\_files/Assignments.htm](http://cs104.cs.ua.edu/index_files/Assignments.htm)).



[www.sepchiou.gr](http://www.sepchiou.gr)

[sepchiou@gmail.com](mailto:sepchiou@gmail.com)

[chioscodeclub@gmail.com](mailto:chioscodeclub@gmail.com)

