# To App Inventor of 12 wpes

# Ανάπτυξη εφαρμογών για συσκευέs Android





Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου Μάρτιος 2015

# ΠΕΡΙΕΧΟΜΕΝΑ

1η Δραστηριότητα: Ζάρια	3
2η Δραστηριότητα: Το μήνυμα που αναβοσβήνει	8
Επεκτάσεις	10
3η Δραστηριότητα: Σχεδιάζοντας με το δάκτυλο	11
Επεκτάσεις	12
4η Δραστηριότητα: Πυξίδα	13
5η Δραστηριότητα: Ζάρια - Επέκταση με σκορ	17
Επεκτάσεις	19
6 <sup>η</sup> Δραστηριότητα: Παιχνίδι με Λογότυπα	21
Επεκτάσεις	24
Δραστηριότητες για Εξάσκηση	25
Ευρετήριο	
Ενδεικτικές λύσεις	31
1 <sup>η</sup> Δραστηριότητα: Ζάρια	31
2 <sup>η</sup> Δραστηριότητα: Το μήνυμα που αναβοσβήνει	31
Επεκτάσεις	31
3η Δραστηριότητα: Σχεδιάζοντας με το δάκτυλο	32
Επεκτάσεις	32
4η Δραστηριότητα: Πυξίδα	33
5 <sup>η</sup> Δραστηριότητα: Ζάρια - Επέκταση με σκορ	33
Επεκτάσεις	33
6 <sup>η</sup> Δραστηριότητα: Παιχνίδι με Λογότυπα	35
Επεκτάσεις	35
Δραστηριότητες για εξάσκηση	
Κορώνα ή Γράμματα	
Πιάνο	
Αντίστροφη μέτρηση	
Ξυστό	39
Κουμπί πανικού	40
Εφαρμογή «I see you»	40
Παιχνίδι Molemash	40
Παιχνίδι μνήμης «Σύγκριση αριθμών»	41



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:

Το App Inventor σε 12 ώρες. Βασιλάκης Βασίλης, Χατζηνικολάκης Γιώργος. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου - ΚΕΠΛΗΝΕΤ Χίου, 2015.

Ευχαριστούμε τους συναδέλφους Επιτροπάκη Κατερίνα, Καραμανή Στέλιο και Παπαδόπουλο Ηλία για τις πολύτιμες συμβουλές και διορθώσεις τους.

# 1<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΖΑΡΙΑ



Θα κατασκευάσουμε μια εφαρμογή που θα δίνει τη δυνατότητα στον χρήστη της να ρίχνει δύο ζάρια, όπως ακριβώς και στο τάβλι, είτε με το πάτημα ενός κουμπιού, είτε ταρακουνώντας τη συσκευή του.

#### τι θα ΜαθΩ

- Να σχεδιάζω το γραφικό περιβάλλον μιας εφαρμογής
- Να αναγνωρίζω τα αντικείμενα του ΑΙ και να τροποποιώ τις ιδιότητές τους
- Να συνδέω την εκτέλεση ενεργειών με γεγονότα

# Βήμα 1 Δημιουργία νέου έργου στο App Inventor

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

## ήμα 2 Μεταφόρτωση αρχείων στο App Inventor

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

#### Designer

Γραφικό Περιβάλλον Εφαρμογής

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

+	ανεβάζουμε το αρχείο με <b>όνομα</b>	σύντομη <b>περιγραφή</b>
	1.png	Ζάρι με αριθμό 1
	2.png	Ζάρι με αριθμό 2
	3.png	Ζάρι με αριθμό 3
	4.png	Ζάρι με αριθμό 4
	5.png	Ζάρι με αριθμό 5

6.png	Ζάρι με αριθμό 6
RollDiceSound.mp3	Ήχος ριξίματος ζαριών

# ήμα 3 Εισαγωγή αντικειμένων στην περιοχή σχεδίασης

Η οθόνη της εφαρμογής μας προς το παρόν είναι κενή και δεν περιέχει τίποτα άλλο παρά μόνο το ίδιο το αντικείμενο της οθόνης. Θα χρειαστούμε ορισμένα αντικείμενα ακόμα:

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

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

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

# Βήμα 4 Καθορισμός Ιδιοτήτων αντικειμένων

Ας τροποποιήσουμε αρχικά ορισμένες ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties**, ως εξής:

*	επιλέγουμε το <b>συστατικό</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
	Screen1	Title: Ζάρια
		Background Color: Dark Gray

Θα μετονομάσουμε επίσης τα αντικείμενα που έχουμε εισάγει στην εφαρμογή μας, ώστε να έχουν χαρακτηριστικά ονόματα και όχι button1, image1 κ.λπ. ώστε να τα αναγνωρίζουμε ευκολότερα. Στη συνέχεια θα τροποποιήσουμε κάποιες από τις ιδιότητες τους.

Στον πίνακα που ακολουθεί φαίνονται αναλυτικά τα νέα ονόματα και οι ιδιότητες των αντικειμένων.

*	επιλέγουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
	Button1	ThrowButton	Background Color: Orange
			Font Size: 30
			Shape: rounded
			Text: Ρίξε τα ζάρια

		Width: Fill Parent Height: 80 (pixels)
HorizontalAlignment1	DiceArea	AlignHorizontal: Centre Width: Fill Parent Height: 100 pixels
lmage1	Dice1Image	Picture: 1.png
Image2	Dice2Image	Picture: 1.png
Sound1	ThrowSound	Source: RollDiceSound.mp3

## <mark>Βήμα 5</mark> Γεγονός - Άγγιγμα του κουμπιού «Ρίξε ζάρια»

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

## Συγγραφή Προγράμματος Blocks

Για το σκοπό αυτό, επιλέγουμε το αντικείμενο ThrowButton από την αριστερή πλευρά της σελίδας και σέρνουμε με το ποντίκι, την εντολή **when [throwButton].click ... do** στο χώρο σύνταξης του προγράμματος, δεξιά.

## Βήμα 6 Ρίξιμο ζαριού - τυχαίοι αριθμοί

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

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

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



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

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

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



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

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

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

```
random integer from [ 1 to [ 100]
```

Θα προσαρμόσουμε τους αριθμούς που παράγονται, στο διάστημα 1 - 6.

Τέλος θα κουμπώσουμε αυτή την εντολή στην 1<sup>η</sup> εγκοπή της εντολής join.

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



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



## Βήμα 7 Ρίξιμο δεύτερου ζαριού - Διπλασιασμός εντολών

Είναι προφανές ότι θέλουμε να εκτελεστούν ακριβώς οι ίδιες ενέργειες για το αντικείμενο Dice2Image. Κάνουμε δεξί κλικ πάνω στην εντολή που θέλουμε να διπλασιάσουμε και επιλέγουμε την εντολή Duplicate (διπλασιασμός).

# Βήμα 8 Αναπαραγωγή ήχου ζαριών

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

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

call (ThrowSound ).Play

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



Η εφαρμογή μας έχει σχεδόν ολοκληρωθεί! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

## Βήμα 9

## Προσθήκη αντικειμένου Accelerometer

Προς το παρόν η ρίψη των ζαριών γίνεται μόνο πατώντας το κουμπί. Ας προσθέσουμε τη δυνατότητα ρίψης των ζαριών εναλλακτικά και με ταρακούνημα της συσκευής.

Designer

Γραφικό Περιβάλλον Εφαρμογής

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

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



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



Η εφαρμογή μας έχει πλέον ολοκληρωθεί! Ώρα να την ξαναελέγξουμε στην κινητή μας συσκευή.

# 2<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΤΟ ΜΗΝΥΜΑ ΠΟΥ ΑΝΑΒΟΣΒΗΝΕΙ

Θα υλοποιήσουμε μια εφαρμογή που θα εμφανίζει στην οθόνη ένα μήνυμα που αναβοσβήνει, ενώ παράλληλα θα αλλάζει και το χρώμα της οθόνης.

#### τι θα ΜαθΩ

 Να «παράγω» ένα γεγονός σε τακτά χρονικά διαστήματα, αξιοποιώντας το αντικείμενο του χρονομέτρου (Clock).

# Βήμα 1 Δημιουργία νέου project

Δημιουργούμε ένα καινούργιο project με όνομα Blinkingmessage.

## μα 2 Προσθήκη αντικειμένων στο γραφικό περιβάλλον

Θα χρειαστούμε μια ετικέτα για να εμφανίζει το μήνυμα που θέλουμε και ένα χρονόμετρο για να μετρά το χρόνο που θα εμφανίζεται και θα εξαφανίζεται το μήνυμα.

## Designer

Γραφικό Περιβάλλον Εφαρμογής

από την <b>ομάδα</b>	μεταφέρουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
Layout	HorizontalArrangement	MessageArea	AlignHorizontal: Center Width: Fill Parent
UserInterface	Label	MessageLabel	FontBold: True FontSize: 50 Text: HELLO WORLD TextColor: Orange Width: Fill Parent
Sensors	Clock	AnimationClock	

## βήμα 3 Η οθόνη που αλλάζει τυχαία χρώμα

Συγγραφή Προγράμματος

Blocks

Για να καθορίσουμε το χρώμα φόντου της οθόνης θα χρησιμοποιήσουμε το πλακίδιο



και για να «φτιάξουμε ένα νέο χρώμα θα χρειαστούμε το πλακίδιο



από την ομάδα εντολών Colors.

Οι τρεις τιμές που φαίνονται στην παραπάνω εικόνα καθορίζουν την ποσότητα κόκκινου, πράσινου και μπλε χρώματος αντίστοιχα, με ελάχιστη δυνατή τιμή το 0 και μέγιστη τιμή το 255 για κάθε ένα από αυτά. Επομένως, ο αρχικός συνδυασμός 255-0-0 παράγει το κόκκινο χρώμα, ενώ αλλάζοντας αυτές τις τιμές μπορούμε να παράγουμε οποιαδήποτε χρωματική απόχρωση επιθυμούμε. Εμείς θα τις αντικαταστήσουμε με τυχαίους αριθμούς από το 0 μέχρι και το 255, όπως παρακάτω, ώστε να παράγεται ένα τυχαίο χρώμα:

set Screen1 • . BackgroundColor • to D	make color 🏮	🔲 make a list 🚺	random integer from [ 0 to [ 255]
		0	random integer from { 0 to ( 255
		0	random integer from { 0 to ( 255

Η εντολή αυτή θα εκτελείται κάθε φορά που εκπνέει ο χρόνος που έχουμε θέσει στο χρονόμετρό μας, δηλαδή:



## βήμα 4 Η ετικέτα που αναβοσβήνει

Για να κάνουμε την ετικέτα να εμφανίζεται και να εξαφανίζεται θα θέτουμε κατάλληλα την την τιμή της ιδιότητα Visible του αντικειμένου MessageLabel χρησιμοποιώντας το πλακίδιο



Εξ ορισμού, όλα τα γραφικά αντικείμενα που προσθέτουμε στην οθόνη της εφαρμογής (όπως η ετικέτα κειμένου μας) είναι ορατά, δηλαδή η ιδιότητα Visible έχει αρχικά την τιμή **True**). Στην περίπτωσή μας θέλουμε η τιμή που θα παίρνει η ιδιότητα Visible να είναι κάθε φορά η αντίθετη από αυτήν που έχει.



Και αυτή η εντολή θα ενεργοποιείται με την εκπνοή του χρονομέτρου.



Η εφαρμογή μας έχει ολοκληρωθεί! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

\_\_\_\_\_

## επεκτάδεις

Προσθέστε τρία κουμπιά με τα οποία ο χρήστης θα ελέγχει τον χρόνο που αναβοσβήνει το μήνυμα. Με το ένα κουμπί θα αυξάνει τον χρόνο του χρονομέτρου κατά 250 ms κάθε φορά που το πατάει, με το δεύτερο κουμπί θα τον μειώνει κατά 250 ms, ενώ με το τρίτο κουμπί θα επαναφέρει τον χρόνο στα 1000 ms.

## 3<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΣΧΕΔΙΑΖΟΝΤΑΣ ΜΕ ΤΟ ΔΑΚΤΥΛΟ

Θα κατασκευάσουμε μια εφαρμογή σχεδίασης με τα δάχτυλα.

#### τι θα ΜαθΩ

 Να σχεδιάζω ελεύθερα στην οθόνη χρησιμοποιώντας το αντικείμενο Canvas (Καμβάς).

# 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 <b>αντικείμενο</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
	Screen1	AlignHorizontal: Center
		BackgroundColor: LightGrey
		Screen Orientation: Portrait
		Title: FingerPainting

## Βήμα 2

Σχέδια επάνω στον καμβά

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

#### Designer

## Γραφικό Περιβάλλον Εφαρμογής

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

\$ από την <b>ομάδα</b>	μεταφέρουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
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 δάχτυλό του πάνω σε αυτήν. Στην πρώτη περίπτωση σχεδιάζεται ένας κύκλος, ενώ στη δεύτερη μια γραμμή.





Όταν ο χρήστης αγγίξει ένα σημείο του καμβά Paper (με συντεταγμένες x και y) τότε σε εκείνο το σημείο ζωγράφισε έναν κύκλο ακτίνας 10 και με γέμισμα.

Το χρώμα του κύκλου καθορίζεται από την ιδιότητα paintColor του αντικειμένου Canvas.







Η εφαρμογή μας έχει ολοκληρωθεί! Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

#### επεκτασεισ

**1**. Κάντε τις απαραίτητες ενέργειες, ώστε όταν ταρακουνάμε τη συσκευή ο καμβάς να καθαρίζει από τα σχέδια που έχουμε κάνει. Θα χρειαστείτε το πλακίδιο **Clear** του αντικειμένου Canvas.

**2**. Προσθέστε στην εφαρμογή 3 κουμπιά με τα οποία ο χρήστης θα μπορεί να θέτει το χρώμα σχεδίασης σε πράσινο, κόκκινο και μπλε ανάλογα με το κουμπί που θα πατάει. Θα χρειαστείτε το πλακίδιο **set [...].paintcolor to [...]** του αντικειμένου Canvas.

# 4<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΠΥΞΙΔΑ



Θα υλοποιήσουμε μια εφαρμογή - πυξίδα που με τη βοήθεια μιας κατάλληλης εικόνας θα μας δείχνει πού είναι ο Βορράς. Όταν μάλιστα η συσκευή δείχνει προς το Βορρά (με απόκλιση +/- 10 μοίρες) θα αναπαράγει ένα σύντομο ηχητικό σήμα.

## τι θα μαθώ

• Να εξετάζω περιπτώσεις και να εκτελώ διαφορετικές εντολές ανά περίπτωση.

# Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Δημιουργούμε ένα καινούργιο project με όνομα Compass.

## Designer Γραφικό Περιβάλλον Εφαρμογής

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen). Μεταβάλλουμε τις ιδιότητές της, όπως παρακάτω.

*	επιλέγουμε το <b>αντικείμενο</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
	Screen1	AlignHorizontal: Center
		Screen Orientation: Portrait
		Title: Ψηφιακή Πυξίδα

#### 3ήμα **2**

Προσθήκη αρχείων πολυμέσων

Προσθέτουμε στο project σας τα αρχεία εικόνας και ήχου που θα χρησιμοποιεί η εφαρμογή. Θα τα βρούμε στη διεύθυνση <u>http://tiny.cc/aifiles</u> στο φάκελο με όνομα Compass.

+	ανεβάζουμε το αρχείο με <b>όνομα</b>	σύντομη <b>περιγραφή</b>	
	compass-arrow.png beep.mp3	Εικόνα πυξίδας με βέλος Ηχητική ειδοποίηση	
Βήμα <b>3</b>	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. Προσθέτουμε στην εφαρμογή τα αντικείμενα που φαίνονται στον πίνακα που ακολουθεί και ρυθμίζουμε κατάλληλα τις ιδιότητές τους.

*	από την <b>ομάδα</b>	μεταφέρουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>		
	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			

## ήμα 4 Ανιχνεύοντας την αλλαγή προσανατολισμού

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

## Συγγραφή Προγράμματος Β

Blocks

Το αντικείμενο OrientationSensor μας δίνει σημαντικές πληροφορίες σχετικές με την κλίση της κινητής μας συσκευής. Εδώ θα χρειαστούμε την τιμή της ιδιότητας Azimuth (Αζιμούθιο) μιας και αυτή μας δείχνει τον προσανατολισμό της συσκευής στον άξονα Βορρά - Νότου.



Azimuth: Έχει τιμή 0 όταν η συσκευή κοιτάζει στο Βορρά, 90° όταν κοιτάζει προς την Ανατολή, 180° στο Νότο και 270° στη Δύση, ενώ η μέγιστη τιμή της είναι η 360°.

Από το αντικείμενο CompassSensor χρησιμοποιούμε την εντολή

when	Com	passSen	isor 🔻	.OrientationChanged
az	imuth	pitch	roll	
do				

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 set [CompassSprite].Heading to... 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.

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





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

Βήμα !

## Ηχητικό σήμα όταν πλησιάζουμε το Βορρά

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

Όπως έχουμε ήδη αναφέρει, το αζιμούθιο μπορεί να πάρει τιμές από 0 μέχρι 360, με αυτές τις ακραίες τιμές να αντιστοιχούν ουσιαστικά στο Βορρά. Στην προκειμένη περίπτωση, αυτό που μας ενδιαφέρει να ελέγξουμε είναι: αν η τιμή του αζιμούθιου είναι μέχρι 10 ή πάνω από 350.

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



Η λογική πρόταση που θέλουμε να ελέγξουμε αποτελείται από δυο μέρη, στα οποία θα χρειαστούμε την τιμή του αζιμούθιου (πλακίδιο get azimuth), για να τη συγκρίνουμε τόσο με το 10, όσο και με το 350. Για τις συγκρίσεις θα χρησιμοποιήσετε το παρακάτω πλακίδιο από την ομάδα Math θέτοντας κατάλληλα τη φορά της ανίσωσης:



Επιπλέον, προκειμένου να συνδυάσετε τις δυο συγκρίσεις και να ολοκληρώσετε τη δημιουργία του διπλού ελέγχου, χρησιμοποιήστε κατάλληλα το παρακάτω πλακίδιο που θα βρείτε στην ομάδα εντολών **Logic** 



και κουμπώστε το στην πρώτη εγκοπή της εντολής if.



Στη συνέχεια εισάγετε τα απαραίτητα αντικείμενα και πλακίδια για την αναπαραγωγή του ήχου.

# 5<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΖΑΡΙΑ - ΕΠΕΚΤΑΣΗ ΜΕ ΣΚΟΡ

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

### τι θα μαθώ

- Να δίνω τη δυνατότητα στην εφαρμογή μου να «θυμάται» τιμές, χρησιμοποιώντας μεταβλητές
- Να μεταβάλλω την τιμή των μεταβλητών αυτών όποτε χρειαστεί

# Βήμα 1 Άνοιγμα του έργου «Ζάρια»

Ανοίγουμε το έργο με τα ζάρια που έχουμε δημιουργήσει στην 1<sup>n</sup> δραστηριότητα.

## μα 2 Προσθήκη ετικετών με σκορ και μήνυμα επιτυχίας

Προσθέτουμε δυο ετικέτες κειμένου, που θα ενημερώνει το χρήστη για το αν κέρδισε καθώς και για το πόσες φορές έχει φέρει εξάρες, με τις ιδιότητες που φαίνονται παρακάτω.

από τr <b>ομάδ</b>	יע מ	μεταφέρουμε το <b>αντικείμενο</b>	του δίνουμε το <b>όνομα</b>	μεταβάλλουμε τις <b>ιδιότητες</b>
UserIn	terface	Label	ResultLabel	Font Size: 40
				Text Alignment: center
				Text:
				Width: Fill Parent
UserIn	iterface	Label	ScoreLabel	Font Size: 20
				TextColor: White
				Text: Σκορ: 0
				Width: Fill Parent

# Βήμα 3 Δημιουργία μεταβλητών για τα ζάρια

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

## Συγγραφή Προγράμματος

Blocks

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



## βήμα 4 Ανάθεση αριθμών ζαριών στις μεταβλητές

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

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



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

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

Τέλος, στην εντολή που θέτει την κατάλληλη εικόνα του ζαριού **set Dice1Image.Picture to...** θα αντικαταστήσουμε στην εντολή **join** τον τυχαίο αριθμό, με την τιμή της κάθε μεταβλητής, χρησιμοποιώντας το πλακίδιο get [...] από την ομάδα εντολών Variables.

get global dice1 •

## Βήμα **5** Έλεγχος για εξάρες

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

Χρησιμοποίηστε κατάλληλα την εντολη **if** για να εξετάσετε κατά πόσο και οι δύο μεταβλητές έχουν την τιμή 6. Θα χρειαστείτε το πλακίδιο



για να συνδυάσετε τις λογικές προτάσεις.

#### οι ενεργείες κατώ από την εντολή if

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

# Βήμα 6 Προσθήκη σκορ στην εφαρμογή

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

Για το σκοπό αυτό θα χρειαστούμε ακόμα μια μεταβλητή. Ονομάστε την *score* και δώστε της σαν αρχική τιμή το 0.

Όταν ο χρήστης φέρνει εξάρες, η τιμή της μεταβλητής πρέπει να αυξάνεται κατά 1.

Στη συνέχεια κάντε τις απαραίτητες ενέργειες, ώστε να εμφανίζεται το σκορ στην ετικέτα ScoreLabel. Το κείμενο της θα είναι η λέξη «Σκορ:» και η τιμή της μεταβλητής score.



Η εφαρμογή μας είναι έτοιμη!

Ώρα να την ελέγξουμε με την κινητή μας συσκευή.

## επεκτασεισ

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

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

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

**4**. Χρησιμοποιήστε το αντικείμενο **TextToSpeech**, ώστε η εφαρμογή να εκφωνεί τους αριθμούς των ζαριών που έφερε ο χρήστης λέγοντας για παράδειγμα το μήνυμα «2 και 4».

**5**. Οι αλλαγές και προσθήκες που κάναμε στην εφαρμογή των ζαριών αφορούν μέχρι στιγμής το ρίξιμο των ζαριών μόνο με τη χρήση του κουμπιού. Κάντε τις απαραίτητες ενέργειες ώστε η ρίψη των ζαριών να γίνεται εναλλακτικά και με το ταρακούνημα της συσκευής.

**6.** Ρίχνοντας μια ματιά στις εντολές που εκτελούνται με το πάτημα του κουμπιού και με το ταρακούνημα της συσκευής παρατηρούμε ότι είναι πανομοιότυπες.

Σε τέτοιες περιπτώσεις αντί να τοποθετούμε ξανά όλα τα πλακίδια, έστω και με διπλασιασμό, δημιουργώντας κώδικα πολύπλοκο και επαναλαμβανόμενο προτιμούμε να «δημιουργήσουμε» ένα νέο πλακίδιο, μια νέα εντολή στην ουσία, που ονομάζουμε διαδικασία. Στη συνέχεια μπορούμε να χρησιμοποιήσουμε την εντολή αυτήοπουδήποτε μέσα στο πρόγραμμά μας.

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

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

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



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



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.



Επομένως κατά την ενεργοποίηση των γεγονότων που αφορούν το κουμπί και το ταρακούνημα της συσκευής θα εκτελούμε απλά την κλήση της διαδικασίας.



# 6<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΠΑΙΧΝΙΔΙ ΜΕ ΛΟΓΟΤΥΠΑ



Θα υλοποιήσουμε ένα παιχνίδι - κουιζ στο οποίο ο παίκτης θα πρέπει αρχικά να αναγνωρίσει ένα λογότυπο και να πληκτρολογήσει το όνομα που του αντιστοιχεί σε ένα πεδίο κειμένου.

Σε επόμενο στάδιο, θα επεκτείνουμε το παιχνίδι ώστε να αποτελείται από περισσότερες ερωτήσεις και λογότυπα.



#### τι θα ΜαθΩ

- Να δίνω τη δυνατότητα στην εφαρμογή μου να «θυμάται» πολλές τιμές, κρατώντας τις όλες μαζί σε ένα αντικείμενο (λίστα)
- Να αποθηκεύω μόνιμα την κατάσταση της εφαρμογής μου στην κινητή συσκευή, ώστε να μπορώ να την επαναφέρω στο σημείο που τερματίστηκε

# Βήμα 1 Δημιουργία απλής εκδοχής με μια ερώτηση

Για αρχή θα υλοποιήσετε μια απλή εκδοχή του παιχνιδιού, η οποία θα περίεχει μόνο ένα λογότυπο (για παράδειγμα του Firefox που φαίνεται στην παραπάνω εικόνα). Όταν ο χρήστης πατήσει το κουμπί «Υποβολή Απάντησης», η εφαρμογή θα ελέγχει αν αυτή είναι η σωστή και θα παίζει είτε έναν ήχο επιτυχίας είτε έναν ήχο αποτυχίας.

Τα αρχεία εικόνας και ήχου που θα χρησιμοποιεί η εφαρμογή θα τα βρείτε στη διεύθυνση <u>http://tiny.cc/aifiles</u> στο φάκελο με όνομα LogoGame. Για την αρχική εκδοχή της εφαρμογής θα χρειαστείτε μόνο την εικόνα με όνομα αρχείου «1.png».





Πώς θα καθαρίζουμε το πεδίο κειμένου κάθε φορά που ο παίκτης υποβάλλει την απάντησή του;

# Βήμα 2 Προσθήκη αρχείων πολυμέσων

Προσθέστε στο project σας τα υπόλοιπα αρχεία εικόνας που θα χρησιμοποιεί η εφαρμογή. Θα τα βρείτε στη διεύθυνση <u>http://tiny.cc/aifiles</u> στο φάκελο με όνομα LogoGame.

# Βήμα 3 Αποθηκεύοντας πολλές τιμές - Το αντικείμενο λίστα

Πώς θα καταφέρουμε να αποθηκεύσουμε τις σωστές απαντήσεις; Θα χρειαστούμε το αντικείμενο της λίστας.

Αρχικά, δημιουργήστε μια νέα μεταβλητή με όνομα answers.

Στη συνέχεια, από την ομάδα εντολών Lists επιλέξτε το πλακίδιο



και κουμπώστε το με την εντολή δημιουργίας της μεταβλητής answers.

Κάνοντας κλικ στο μπλε εικονίδιο του πλακιδίου make a list προσθέστε αντικείμενα, αυξάνοντας τον αριθμό των εγκοπών του πλακιδίου. Συνολικά, στην εφαρμογή μας θα χρειαστούμε 5 θέσεις στη λίστα. Το τελικό αποτέλεσμα θα είναι όπως παρακάτω.

initialize global (answers) to 🚺	🔲 make a list 🏮 * (FIREFOX) *
	LINUX) *
	OPERA *
	APPLE)

Βήμα 4 Επιλέγοντας την κατάλληλη εικόνα

Πώς και πότε θα μετακινούμαστε κάθε φορά στη νέα εικόνα;

Δημιουργείστε μια νέα μεταβλητή, έστω **selection** με αρχική τιμή 1.

Θέστε την πηγή της εικόνας που θα προβάλλεται στην οθόνη με βάση την τιμή της μεταβλητής selection, όπως κάναμε και στη δραστηριότητα με τα ζάρια.

Όταν ο παίκτης καταφέρει να βρίσκει την απάντηση θα εμφανίζεται η επόμενη εικόνα.

# Βήμα 5 Ελέγχοντας την απάντηση του παίκτη

Στην πρώτη εκδοχή της εφαρμογής που υλοποιήσατε εξετάζατε αν η απάντηση του δείχνει ο παίκτης είναι η λέξη FIREFOX μιας και έβλεπε μόνο μια εικόνα.



Το μόνο που πρέπει να αντικαταστήσουμε είναι η λέξη FIREFOX με την αντίστοιχη απάντηση που περιέχει η λίστα, ανάλογα με την τιμή της μεταβλητής **selection**.



## α 6 Φτάνοντας στο τέλος της λίστας

Τι θα γίνει όταν ο παίκτης καταφέρει να βρει όλα τα λογότυπα; Μια ιδέα είναι το παιχνίδι να ξεκινάει από την αρχή επιστρέφοντας στην πρώτη εικόνα.

Κάντε τις απαραίτητες ενέργειες, ώστε όταν η τιμή της μεταβλητής **selection** γίνει μεγαλύτερη από το μέγεθος της λίστας να ξαναπαίρνει την αρχική της τιμή. Θα χρειαστείτε το πλακίδιο

length of list list 🕻

από την ομάδα εντολών Lists, με το οποίο μπορούμε να μάθουμε το πλήθος των τιμών μιας λίστας.

# Βήμα **7** Αποθήκευση παιχνιδιού

Τι θα γίνει στην περίπτωση που το παιχνίδι μας έχει 200 ή περισσότερες εικόνες; Θα μπορέσει ο παίκτης να το ολοκληρώσει μονομιάς; Πώς μπορούμε να αποθηκεύσουμε την τιμή της μεταβλητής selection, ώστε ο παίκτης να συνεχίζει το παιχνίδι από το σημείο που σταμάτησε;

Θα προσθέσουμε καταρχήν ακόμα ένα κουμπί, για να δώσουμε στον παίκτη τη δυνατότητα να κάνει αποθήκευση.

Για να αποθηκεύσουμε μόνιμα τιμές από την εφαρμογή στο κινητό μας θα χρησιμοποιήσουμε το αντικείμενο **TinyDB** που βρίσκεται στην ομάδα αντικειμένων **Storage**.

Κάντε τις απαραίτητες ενέργειες, ώστε η εφαρμογή σας να πάρει την παρακάτω μορφή.

Screen1	<b>⊜₁∥ 2</b> 9:48
Υποβολή Απάντησης	; Αποθήκευση
Non-visible	e components

Όταν ο παίκτης πατήσει το κουμπί Αποθήκευση θα πρέπει να αποθηκεύουμε την τιμή της μεταβλητής **selection** στην κινητή μας συσκευή δίνοντας της ένα μοναδικό αναγνωριστικό όνομα, όπως παρακάτω.



ενώ κατά την εκκίνηση της εφαρμογής θα αναζητούμε την τιμή με το αναγνωριστικό "selection" στην κινητή μας συσκευή και εφόσον τη βρούμε θα τη θέτουμε ως τιμή της μεταβλητής selection. Διαφορετικά, θα βάζουμε την τιμή 1.



Πλέον, εφόσον ο παίκτης κάνει αποθήκευση, ακόμα και να κλείσει την εφαρμογή, όταν την ανοίξει ξανά θα συνεχίσει το παιχνίδι από εκεί που σταμάτησε και όχι από την αρχή.



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

### επεκτασεισ

 Προσθέστε τις εικόνες και από άλλα λογότυπα, ώστε το παιχνίδι να ολοκληρωθεί (βάλτε τουλάχιστον 10 ερωτήσεις).

**2**. Δώστε στον παίκτη τη δυνατότητα να πηγαίνει στην επόμενη εικόνα χωρίς να δώσει απάντηση.

**3**. Με δεδομένη την παραπάνω προσθήκη υπολογίστε με τη βοήθεια μεταβλητών σε πόσες ερωτήσεις ο παίκτης απαντάει σωστά και πόσες παραλείπει. Τα αποτελέσματα θα εμφανίζονται καθόλη τη διάρκεια του παιχνιδιού σε ετικέτες (labels) που θα τοποθετήσετε στην οθόνη.

# ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

## 1. Κορώνα ή Γράμματα



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

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

Αφού ολοκληρώσετε την αρχική εκδοχή της εφαρμογής που περιεγράφηκε παραπάνω σκεφτείτε με ποιους άλλους τρόπους θα μπορούσε ο παίκτης να ρίχνει το νόμισμα, πέραν του να αγγίζει το κουμπί. Προσθέστε τους στην εφαρμογή που υλοποιήσατε.

## 2. Πιάνο



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

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

# 3. Αντίστροφη μέτρηση



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

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



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



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

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

Θα χρειαστείτε το αντικείμενο Notifier που βρίσκεται στην ομάδα UserInterfac Στο τμήμα των εντολών θα χρησιμοποιήσετε την εντολή που φαίνεται παρακάτ		
call (Notifier1 - ShowMessageDialog message ( Παρακαλώ συμπληρώστε τον αριθμό δευτερολέπτων title ( Σφάλμα buttonText ( ΟΚ -		
Για τον έλεγχο θα σας φανεί χρήσιμο το παρακάτω πλακίδιο της ομάδας Text, το οποίο επιστρέφει την τιμή Αληθές (true) εάν το κείμενο που περιέχεται στα δεξιά του έχει μηδενικό μήκος (δηλαδή είναι κενό):		
is empty		

## 4. Ξυστό

Δημιουργήστε μια εφαρμογή «Ξυστό». Θέστε μια εικόνα της επιλογής σας ως φόντο της οθόνης και καλύψτε τη με έναν τεράστιο άσπρο κύκλο που θα σχεδιάζεται πάνω στον καμβά με την εκκίνηση της εφαρμογής. Στη συνέχεια θα μπορεί ο χρήστης να «σβήνει» τον άσπρο κύκλο, σχεδιάζοντας με το δάκτυλό του γραμμές πάνω στην οθόνη και αποκαλύπτοντας σιγά σιγά την εικόνα που κρύβεται από κάτω. Για να μπορέσει ο χρήστης να «σβήσει» τμήματα του καμβά θέστε το χρώμα σχεδίασης (ιδιότητα PaintColor) όπως φαίνεται παρακάτω:



Παρατηρήστε ότι πρέπει να προσθέσουμε και μια τέταρτη τιμή στο πλακίδιο και να τη θεσουμε ίση με μηδέν, ώστε ο συνδυασμός 255-255-255-0 να παράγει διαφανές χρώμα.

# 5. Κουμπί πανικού



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

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

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

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

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

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

call SMSTexting .SendMessage

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

## 6. Εφαρμογή «I see you»

Χρησιμοποιήστε κατάλληλα τον αισθητήρα προσανατολισμού και την ιδιότητα Pitch, ώστε όταν ο χρήστης της εφαρμογής γέρνει τη συσκευή προς το μέρος του να αναπαράγει τον ήχο "I see you.mp3" που θα κατεβάσετε από το <u>http://tiny.cc/aifiles</u> και το φάκελο Scary.



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.

# 7. Παιχνίδι Molemash



Σχεδιάστε και αναπτύξτε το παιχνίδι **MoleMash**, εμπνευσμένο από ένα κλασσικό arcade παιχνίδι, στο οποίο εμφανίζονται αντικείμενα σε τυχαίες θέσεις και ο παίκτης πρέπει να προλάβει να τα χτυπήσει πριν αυτά εξαφανιστούν. Στη δική μας εκδοχή, για να μην γινόμαστε βίαιοι <sup>(2)</sup>, ας υποθέσουμε ότι εμφανίζεται ένα χαριτωμένο ζωάκι και θέλουμε απλά να το αγγίξουμε πριν εξαφανιστεί και εμφανιστεί σε κάποια άλλη θέση.

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

#### Βασικές λειτουργίες:

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

Μπορείτε να μετακινείτε το mole σε τυχαία θέση εντός του καμβά, χρησιμοποιώντας την εντολή ...

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

- Εάν ο παίκτης προλάβει να αγγίξει το ζωάκι, τότε ο παίκτης κερδίζει ... πόντους και το mole εμφανίζεται αμέσως σε επόμενη τυχαία θέση.
- Όταν πατηθεί το πλήκτρο "Start/Reset" μηδενίζεται το σκορ για την εκκίνηση ενός νέου παιχνιδιού.

#### 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.
- 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.

## 8. Παιχνίδι μνήμης «Σύγκριση αριθμών»



Αναπτύξτε ένα παιχνίδι μνήμης, που εμφανίζει τριψήφιους αριθμούς στην οθόνη του κινητού και ο παίκτης καλείται να επιλέξει αν ο κάθε επόμενος αριθμός που εμφανίζεται είναι μεγαλύτερος ή μικρότερος από τον προηγούμενο, πατώντας αντίστοιχα το κουμπί πάνω ή κάτω από τον αριθμό που εμφανίζεται (ενδεικτικά η οθόνη αριστερά περιέχει δυο τέτοια διαφανή κουμπιά πάνω και κάτω από τον τυχαίο αριθμό 202).

Σταδιακά το χρονικό διάστημα εντός του οποίου καλείται ο παίκτης να απαντήσει μειώνεται και το παιχνίδι τελειώνει είτε όταν δοθεί λάθος απάντηση, είτε όταν ο παίκτης δεν προλάβει να απαντήσει.

Στο ενδεικτικό στιγμιότυπο της παραπάνω εικόνας, ο παίκτης έχει ήδη πατήσει το κουμπί Initialize (το οποίο και απενεργοποιήθηκε προσωρινά), ώστε να εμφανιστεί στην οθόνη ο πρώτος τυχαίος αριθμός. Πατώντας το κουμπί Start ο χρόνος θα αρχίσει να μετράει αντίστροφα και θα εμφανιστεί ο επόμενος αριθμός στην οθόνη, ώστε ο παίκτης να επιλέξει την απάντησή του. Για κάθε σωστή απάντηση, το σκορ του θα αυξάνεται και θα εμφανίζεται ο επόμενος τυχαίος αριθμός.

Ŷ

Θα χρειαστείτε το αντικείμενο Clock για να μετρά το χρόνο που διατίθεται στον παίκτη για να απαντήσει. Το χρονικό διάστημα αυτό μπορεί πχ να ξεκινάει από 1500 ms και σταδιακά να μειώνεται ώστε το παιχνίδι να δυσκολεύει.

Ουμηθείτε, με βάση και την περιγραφή του παιχνιδιού, ότι αν το χρονόμετρο τερματίσει, σημαίνει ότι δεν πρόλαβε ο παίκτης να πατήσει τη σωστή απάντηση και το παιχνίδι τελειώνει.

Εάν βέβαια η απάντηση που δίνεται είναι η σωστή, μην ξεχάσετε να επαναφέρετε/μηδενίσετε το χρονόμετρο, χρησιμοποιώντας κατάλληλα την εντολή:



Φροντίστε επίσης με τις κατάλληλες εντολές ελέγχου να μην τύχει να εμφανιστεί ο ίδιος αριθμός δυο συνεχόμενες φορές, καθώς θα είναι αδύνατο για τον παίκτη να επιλέξει κάποια σωστή απάντηση! <sup>©</sup>

Φυσικά μπορείτε να σκεφτείτε τους δικούς σας κανόνες, καθώς και το δικό σας τρόπο που θα δίνονται απαντήσεις από το χρήστη.



Μπορείτε να κατεβάσετε και να παίξετε μια απλή εκδοχή του παιχνιδιού από το Google Play του Συλλόγου Εκπαιδευτικών Πληροφορικής Χίου στο <u>http://tiny.cc/sepchiouapps</u> η οποία αποτελεί μια απλούστερη παραλλαγή της εφαρμογής "Number - remember the previous" του συναδέλφου Αργύρη Τερζάκη (<u>http://tiny.cc/numbersgame</u>).

# **EYPETHPIO**

#### Δραστηριότητα 1

- Τυχαίοι αριθμοί
- Αντικείμενο Επιταχυνσιόμετρο (Accelerometer Sensor)

#### Δραστηριότητα 2

- Αντικείμενο *Χρονόμετρο* (Clock)
- Δημιουργία χρώματος
- Ιδιότητα αντικειμένου: Ορατό (Visible)
- Λογικός τελεστής OXI (not)

#### Δραστηριότητα 3

- Αντικείμενο Καμβάς (Canvas)
- Σχεδιασμός γραμμών και κύκλων

#### Δραστηριότητα 4

- Αντικείμενο Αισθητήρας προσανατολισμού (Orientation Sensor)
- Εντολή επιλογής if
- Λογικός τελεστής Ή (or)

#### Δραστηριότητα 5

- Μεταβλητές
- Λογικός τελεστής KAI (and)
- Εντολή επιλογής if

#### Δραστηριότητα 6

- Εντολή επιλογής if
- Εντολές χειρισμού κειμένου
- Λίστες
- Αποθήκευση τιμών στην κινητή συσκευή

# ΕΝΔΕΙΚΤΙΚΕΣ ΛΥΣΕΙΣ

## 1<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΖΑΡΙΑ



## 2<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΤΟ ΜΗΝΥΜΑ ΠΟΥ ΑΝΑΒΟΣΒΗΝΕΙ

whe	when AnimationClock Timer							
do	set Screen1 . BackgroundColor . to (	make color 🌔	🔲 make a list 🖡	random integer from ( 0 to ( 255				
			0	random integer from ( 0 to ( 255				
			0	random integer from ( 0 to ( 255				
	set MessageLabel • . Visible • to 🖡 not	( MessageLa	abel 🔹 . Visible 🔹					

#### επεκτάσεις



3<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΣΧΕΔΙΑΖΟΝΤΑΣ ΜΕ ΤΟ ΔΑΚΤΥΛΟ when Canvas1 . Touched touchedAnySprite X V call Canvas1 . DrawCircle centerX get x 🔹 centerY get y • radius get global dot 🔹 fill true 🔻 when Canvas1 . Dragged prevX prevY currentX currentY draggedAnySprite startX startY call Canvas1 . DrawLine do x1 get prevX y1 get prevY • get currentX **x**2 get currentY y2

#### επεκτάσεις

1.



2.



## 4<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΠΥΞΙΔΑ



## 5<sup>Η</sup> ΔΡΑΣΤΗΡΙΟΤΗΤΑ: ΖΑΡΙΑ - ΕΠΕΚΤΑΣΗ ΜΕ ΣΚΟΡ

initia	lize glo	bal (dice1) to 40 initialize global score to 40					
initi	initialize global dice2 to						
whe	n Thro	wButton  .Click					
do	set g	lobal dice1 v to C random integer from (1) to (6					
	set g	lobal dice2 to C random integer from (1) to (6)					
	set 🚺	Dice1Image 🔹 . Picture 🔹 to 🖡 🔲 join 🕻 get global dice1 🔹					
	$L_{-}$	( <sup>1</sup> png)					
	set 🚺	Dice2Image 🔹 . Picture 🔹 to 🖡 🔲 join 🌔 get global dice2 🔹					
		( png)					
	call 🚺	ThrowSound  .Play					
	() if	f f get global dice1 • = • (6) and • f get global dice2 • = • (6)					
	then	then set ResultLabel . Text . to ( Kέρδισες! )					
	set global score T to C 🔲 🖌 get global score T + C 1						
		set ScoreLabel • . Text • to [ ] join [ * Σκορ : *					
		get global score •					
	<u> </u>						

## επεκτάσεις

1 - 2 - 3

initi	alize global die	ce1 to [ 🛛	initialize g	lobal dice2	to CO		
whe	n ThrowButto	n 🔨 .Click	initialize g	lobal score	to ( 0		
do	set ResultLabel T. Text T to ( " )"						
	set global die	e1 🔹 to 🥻 ran	dom integer	from (	to ( <b>6</b>		
	set global die	ce2 🔹 to 🕻 ran	dom integer	from (	to (6		
	set Dice1Im	age 🔹 Picture	🔹 to 🖡 🧕	) join ( ge	global dice1 🔪		
	set Dice2Image . Picture to ( ) join ( get global dice2 .						
	call ThrowSound .Play						
	If ↓ get global dice1 ▼ = ▼ ↓ get global dice2 ▼						
	then set	ResultLabel 🔹 .	Text ▼ to	ι " (Κέρδισε	ς!) "		
	set g	lobal score 🔹 to	( 🕑 ( g	et global sco	re • 1 + (1		
	set 🕃	ScoreLabel 🔹 . (	Text 🔹 to 🚺	🧿 join (	" [ Хкор : ] "		
					get global score *		

whe	n ThrowButton 🔻	.LongClick		
do	set global score	• to (0		
	set ScoreLabel	• . Text •	to 🔰 🙆 join	( <sup>1</sup> " Σκορ : "
				get global score 🔻

4.

call (TextToSpeech1 · ).Speak		
message (	🔲 join 🖡	get global dice1 •
	L C	" and "
		get global dice2 -

5.

whe	n AccelerometerSensor1 · Shaking
do	set global dice1 • to ( random integer from ( 1 to ( 6
	set global dice2 to ( random integer from (1) to (6)
	set Dice1 • . Picture • to ( D join ( get global dice1 •
	( <sup>1</sup> ".png "
	set Dice2 • . Picture • to ( D join ( get global dice2 •
	(, " .png "
	call TextToSpeech1 . Speak
	message ( 🔲 join ( get global dice1 🔻
	( and "
	get global dice2 ▼



#### επεκτάσεις

2.

when IgnoreButtonClick	
do set global selection • to ( 🔲 ( get global selection • + (	1
If ( get global selection ▼ ≥ ▼ ( length of list list )	get (global answers -
then set global selection • to [1]	
3	
initialize global (ignored) to (	
initialize global correct to	
initialize global selection to 41	
initialize global (answers) to ( make a list ( FIREFOX) "	
" OPERA "	
(APPLE)"	
when Screen1 - Initialize	_
do set global selection • to call SelectionValueDB • .GetValue	
tag valuelfTagNotThere	selection
set global correct to ( call SelectionValueDB .GetValue	
tag (	" correct "
valuelfTagNotThere	*0*
set global ignored v to ( call SelectionValueDB v .GetValue	
tag valuelfTagNotThere	f (gnored)
set [Logolmage -]. Picture - to ( D join ( get global select	tion •
( png *	







## when SubmitButton . Click



### ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΕΞΑΣΚΗΣΗ

#### κορώνα η γραμματά

whe	n CoinButton 🔻 .Click		
do	set CoinButton 🔨 . Image 🔨 to 🖡	🧿 join 🖡	random integer from [ 1 to [ 2
		L L	" .png "
	call Sound1 .Play		

#### ΠΙΑΝΟ

when CButton . Click do set NotesSound . Source to c.mp3 call NotesSound . Play	when AButton . Click do set NotesSound . Source to ( a.mp3 ' call NotesSound . Play
when DButton . Click	when BButton . Click
do set NotesSound . Source to d.mp3	do set NotesSound . Source to b.mp3
call NotesSound .Play	call NotesSound .Play
when FButton . Click	when EButton . Click
do set NotesSound . Source to f.mp3	do set NotesSound . Source to the e.mp3
call NotesSound . Play	call NotesSound .Play
when GButton . Click do set NotesSound . Source to f g.mp3 call NotesSound . Play	when Screen1 • .Initialize do set NotesSound • . Source • to [ • c.mp3 • set NotesSound • . Source • to [ • d.mp3 • set NotesSound • . Source • to [ • e.mp3 • set NotesSound • . Source • to [ • f.mp3 • set NotesSound • . Source • to [ • g.mp3 • set NotesSound • . Source • to [ • a.mp3 • set NotesSound • . Source • to [ • a.mp3 • set NotesSound • . Source • to [ • b.mp3 •

#### ΑΝΤΙΣΤΡΟΦΗ ΜΕΤΡΗΣΗ







#### ΞΥΣΤΟ

whe	Screen1 .Initialize	
do	set Canvas1 . PaintColor .	to 🕻 📒
	call Canvas1 . DrawCircle	
	centerX 🙀	0
	centerY 🕌	0
	radius 🕌	2000
	fill 🙀	true 🔹

whe	n (C	anvas1 🔹 .D	ragged				
st	artX	startY p	orevX prevY	) (curi	rentX current	tY dragge	dAnySprite
do	set	Canvas1 •	. PaintColor	🔹 to 🏮	make color 💧	🗖 🔲 make a	a list 🗘 255
							255
							255
							0
	set	Canvas1 •	. LineWidth	🛭 to 🏮	50		
	call	Canvas1 •	.DrawLine				
			x1 🌗	get st	tartX 🔹		
			y1 🔰	get st	tartY 🔹		
			x2 🔰	get C	urrentX 🔹		
			y2 🔰	get C	urrentY 🔹		

#### ΚΟΥΜΠΙ ΠΑΝΙΚΟΥ

whe	n PanicButton . Click
do	call SMSTexting .SendMessage
	call SirenPlayer Start
	set PanicButton 🔹 . Enabled 🔹 to 🕻 false 🔹
whe	n StopButton . Click
whe do	n StopButton Click call SirenPlayer Stop

### ΕΦΑΡΜΟΓΗ «I SEE YOU»

ΠΑΙΧΝΙΔΙ MOLEMASH



#### initialize global (TimeLeft) to 115 initialize global Score to when Pista .Touched initialize global Missed to 📢 🔘 initialize global Hits to 埍 🛛 x y touchedSprite do 🧿 if 🛛 🕴 get touchedSprite 🔻 then set global Hits T to C o get global Hits + (1 when StartResetButton .Click do set global Hits to CO set TimeClock . TimerEnabled . to false set global Missed T to 0 call MoveMole • set global Score T to CO set TimeClock • . TimerEnabled • to I true • set global TimeLeft T to (15) set global Missed 🔹 to 🕻 💿 🕻 get global Missed 💙 + 🕻 1 call UpdatePoints set TimeClock . TimerEnabled to to call UpdatePoints · set Mole . Visible to true

<b>0</b> 1	to (UpdatePoints)
do	set HitsLabel 🔽 . Text 💌 to 🕴 get global Hits 🔪
	set MissedLabel T . Text T to G get global Missed T
	set ScoreLabel • . Text • to • • • • • • • • • • • • • • • • •
0	to MoveMole
do	call Mole T .MoveTo
	x ( random integer from ( O to ( ) Pista • . Width • - ( Mole • . Width • )
	y ( random integer from ( 0 to ( ) Pista • . (Height • ) - ( Mole • . (Height •

#### ΠΑΙΧΝΙΔΙ ΜΝΗΜΗΣ «ΣΥΓΚΡΙΣΗ ΑΡΙΘΜΩΝ»

