

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : GUARDDOG – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

Θα επεκτείνουμε τη λειτουργικότητα της εφαρμογής Guard Dog, ώστε να λειτουργεί σαν συναγερμός, όταν κάποιος επιχειρήσει να σηκώσει τη συσκευή από τη θέση της.

Πιο συγκεκριμένα, ο σκύλος της κεντρικής οθόνης της εφαρμογής ενοχλείται και γαβγίζει όταν η συσκευή παύει να είναι ακινητοποιημένη σε μια σταθερή επιφάνεια. Επιπλέον, η φιγούρα του σκύλου δεν θα ηρεμεί παρά μόνο μετά το πέρας λ.χ. 2 δευτερολέπτων ακινησίας της συσκευής.

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

Βήμα 1 Άνοιγμα υπάρχοντος project

Ανοίγουμε το ήδη υπάρχον project με ονομασία GuardDog.

Βήμα 2 Αντίδραση του σκύλου στο σήκωμα της συσκευής

Οι φορητές ηλεκτρονικές συσκευές διαθέτουν πλήθος αισθητήρων, όπως τοποθεσίας (GPS), προσανατολισμού (Orientation), επιτάχυνσης (Acceleration) και ραβδωτού κώδικα (Barcode), οι οποίοι μπορούν να χρησιμοποιηθούν άμεσα στο App Inventor και να προσδώσουν ιδιαίτερα χαρακτηριστικά και αυτοματισμούς στην εφαρμογή μας. Εν προκειμένω, η συσκευή μας θα λειτουργεί σαν συναγερμός, αξιοποιώντας το επιταχυνσιόμετρο (accelerometer) της, σαν ανιχνευτή «σηκώματός» της.

Designer Γραφικό Περιβάλλον Εφαρμογής

Για να προσθέσουμε δυνατότητες ανίχνευσης της κίνησης της συσκευής, θα χρειαστεί να μεταβούμε στο Designer και να προσθέσουμε στην οθόνη της εφαρμογής μας το μη ορατό συστατικό AccelerometerSensor.



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

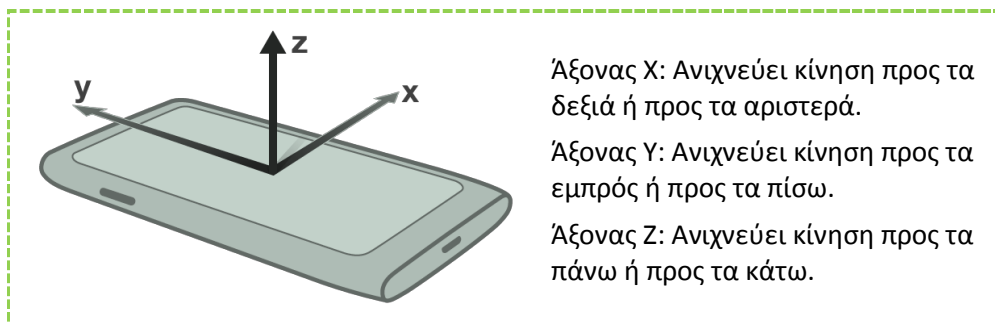
μεταβάλλουμε τις
ιδιότητες

Sensors

AccelerometerSensor

AccSensor

Ο συγκεκριμένος αισθητήρας ανιχνεύει το κούνημα της συσκευής, μετρώντας παράλληλα (κατά προσέγγιση) την επιτάχυνση της βαρύτητας (m/s^2) στις τρεις διαστάσεις.



Συγκεκριμένα, όταν η συσκευή βρίσκεται τοποθετημένη οριζόντια, με την οθόνη προς τα πάνω, η τιμή της παραμέτρου zAccel είναι περίπου 9.8 (επιτάχυνση της βαρύτητας). Όταν το τηλέφωνο κινείται προς τα πάνω, η τιμή της zAccel αυξάνεται.

Για τις ανάγκες του συναγερμού μας, αρκεί ο έλεγχος της τιμής της παραμέτρου zAccel.

Συγγραφή Προγράμματος

Blocks

Μεταβαίνουμε στο Blocks, όπου θέλουμε να χτίσουμε το κατάλληλο σενάριο, που συνοψίζεται στο: «Όταν ο αισθητήρας (στον κατακόρυφο άξονα z) ανιχνεύσει κίνηση, τότε ο σκύλος θυμώνει και γαβγίζει».

Προκειμένου, να επιτύχουμε αυτό το επιλεκτικό σενάριο, χρειαζόμαστε τα ακόλουθα:



από το
συρτάρι

επιλέγουμε την
εντολή

η οποία
σημαίνει

AcceleratorSensor

when AccSensor
AccelerationChanged do

όταν ανιχνευτεί οποιαδήποτε αλλαγή στις τιμές των επιταχύνσεων στους τρεις άξονες, τότε ...

Control

if ... then

Αν ισχύει η *συνθήκη*, τότε

Math

... = ...
και την τροποποιούμε σε
... > ...

Ελέγχεται η τιμή μιας συνθήκης (το αποτέλεσμα της σύγκρισης είναι true ή false)

Math

0
και αλλάζουμε την τιμή σε
11

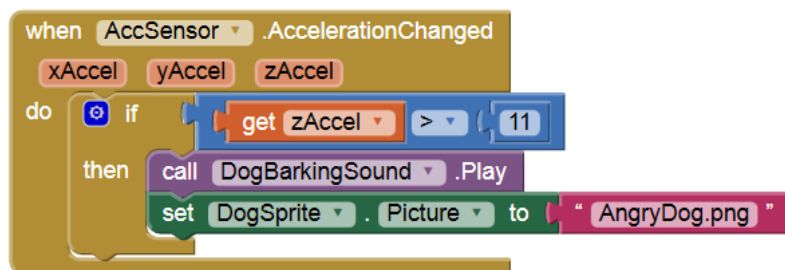
Ορίζει την τιμή 11

DogCalmClock

set DogCalmClock
TimerEnabled to

Ενεργοποίησε ή απενεργοποίησε το χρονόμετρο

Συνδυάζοντας τα παραπάνω, χτίζουμε το απαιτούμενο σενάριο:



Όποτε υπάρχει αλλαγή στο επιταχυνσιόμετρο (δηλαδή, η παραμικρή κίνηση), αν παρατηρηθεί τιμή της zAccel μεγαλύτερη από 11 (ο αριθμός αυτός πρέπει οπωσδήποτε να είναι μεγαλύτερος του 10 και μπορούμε να πειραματιστούμε με κοντινές τιμές), τότε κάλεσε την αναπαραγωγή του ήχου γαβγίσματος και άλλαξε τη φιγούρα του σκύλου σε θυμωμένη.

Ο σκύλος μας θύμωσε και παραμένει θυμωμένος, παρόλο που αφήνουμε τη συσκευή ακίνητη. Προς το παρόν, ο μόνος τρόπος να ηρεμήσουμε το σκύλο, είναι πρώτα να τον εκνευρίσουμε (π.χ. αγγίζοντάς τον), ώστε αφήνοντάς τον, να επανέλθει στη χαλαρή του κατάσταση.

Βήμα 3 Ο σκύλος ηρεμεί μετά από ένα διάστημα ακινησίας

Στο βήμα αυτό, και μετά από σήκωμα της συσκευής, ο σκύλος ηρεμεί από μόνος του, μετά από την πάροδο ενός χρονικού διαστήματος ηρεμίας.

Designer Γραφικό Περιβάλλον Εφαρμογής

Για το σενάριο αυτό θα χρειαστούμε το αντικείμενο του ρολογιού (Clock).



Το αντικείμενο Clock λειτουργεί σαν ρολόι, σαν χρονόμετρο και σαν βοηθητικό εργαλείο για χρονικούς υπολογισμούς ή απλούστερα για την έναρξη γεγονότων ανά συγκεκριμένα χρονικά διαστήματα.

Εδώ θα το χρησιμοποιήσουμε σαν χρονόμετρο. Όταν εκπνεύσει το χρονικό διάστημα που ορίζουμε στην ιδιότητα TimeInterval (σε msec) τότε ενεργοποιείται ένα γεγονός.



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

Αρχικά, δεν χρειάζεται το χρονόμετρο να είναι ενεργό, καθώς το χρειαζόμαστε μόνο όταν σηκωθεί η συσκευή.

Η διάρκεια του χρονομέτρου είναι 3000 msec, δηλαδή ορίζουμε να τερματίζει (και να ξεκινάει ξανά) κάθε 3 δευτερόλεπτα.

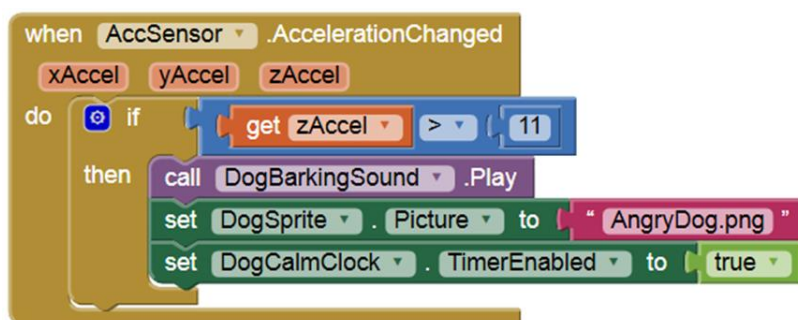
Συγγραφή Προγράμματος

Blocks

Για να ενεργοποιείται το χρονόμετρο, με σήκωμα της συσκευής, αλλά και για να απενεργοποιείται αυτόματα μετά από το χρονικό διάστημα που ορίσαμε προηγουμένως, χρειαζόμαστε την εντολή:

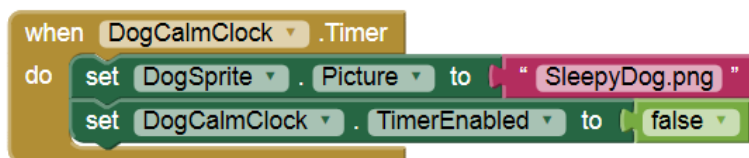
 από το συρτάρι	επιλέγουμε την εντολή	η οποία σημαίνει
DogCalmClock	set DogCalmClock TimerEnabled to	Ενεργοποίησε ή απενεργοποίησε το χρονόμετρο
Logic	true ή false	Αληθές ή Ψευδές

Έτσι, επεκτείνουμε τη λειτουργία ελέγχου της τιμής του επιταχυνσιομέτρου:



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

Αντιστοίχως, όταν το χρονόμετρο τερματίζει, θέτουμε τη φιγούρα του σκύλου στην ήρεμη εκδοχή της και εφόσον δεν χρειάζεται να λειτουργεί πλέον το χρονόμετρο, μπορούμε να το απενεργοποιήσουμε ξανά.



Όταν το χρονόμετρο DogCalmClock τερματίσει, θέσε στη μορφή του σκύλου την ήρεμη φιγούρα και απενεργοποίησε το χρονόμετρο.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΚΟΡΩΝΑ ΓΡΑΜΜΑΤΑ – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

Θα επεκτείνουμε την εφαρμογή Κορώνα – Γράμματα που δόθηκε για εξάσκηση στο σπίτι στο προηγούμενο μάθημα. Καταρχήν θα την μετατρέψουμε σε ένα απλό παιχνίδι, στο οποίο ο παίκτης θα επιλέγει Κορώνα ή Γράμματα και στη συνέχεια θα ρίχνει το νόμισμα. Επίσης, θα προσθέσουμε animation κατά την ρίψη του νομίσματος, για να μοιάζει το νόμισμα σαν να γυρίζει.

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

- Ο λογικός τελεστής NOT.
- Εντολή ελέγχου Αν ... Αλλιώς (If ... then ... else).
- Animation με τη βοήθεια ρολογιού.

Βήμα 1 Άνοιγμα του έργου «Κορώνα Γράμματα»

Αφού μεταβείτε στη σελίδα του [App Inventor](https://appinventor.mit.edu/) και συνδεθείτε με τα στοιχεία του λογαριασμού σας, ανοίξτε το έργο κορώνα γράμματα.

Designer Γραφικό Περιβάλλον Εφαρμογής

Τα αρχεία εικόνων και ήχου που θα χρειαστείτε για την εφαρμογή μπορείτε να τα κατεβάσετε από το <http://tiny.cc/aifiles> και το φάκελο CoinFlip. Θα χρειαστείτε δυο εικόνες για τις δυο όψεις του νομίσματος και τους ήχους για το στρίψιμο και την επιτυχή πρόβλεψη.

Βήμα 2 Προσθήκη κουμπιών επιλογής στην εφαρμογή

Θα προσθέσουμε δύο νέα κουμπιά στην εφαρμογή, ώστε να δίνουμε τη δυνατότητα στον χρήστη να επιλέξει κορώνα ή γράμματα πριν στρίψει το νόμισμα.

Τα δύο κουμπιά θα μπουν μέσα σε ένα αντικείμενο HorizontalArrangement ώστε να τοποθετηθούν το ένα δίπλα στο άλλο.



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

μεταβάλλουμε τις
ιδιότητες

Layout

HorizontalArrangement

CoinArea

Height : 100 pixels

AlignHorizontal : Center

UserInterface	Button	HeadButton	Image : 1.jpg Width : 60 pixels Height : 60 pixels
UserInterface	Button	TailButton	Image : 2.jpg Width : 60 pixels Height : 60 pixels



Εικόνα 75 - Προσθήκη κουμπιών επιλογής στην εφαρμογή

Βήμα 3 Προσθήκη κουμπιού για το στρίψιμο του νομίσματος

Αν δεν το έχετε κάνει ήδη από τις ασκήσεις του προηγούμενου μαθήματος προσθέστε και ένα κουμπί για το στρίψιμο του νομίσματος καθώς και ένα αντικείμενο Player για την αναπαραγωγή των ήχων. Προσέξτε ότι αρχικά ορίζουμε την ιδιότητα enabled (Ενεργοποιημένο) του κουμπιού στην τιμή false, ώστε να μην μπορεί να χρησιμοποιηθεί μέχρι να επιλέξει ο χρήστης την πλευρά του νομίσματος που θέλει.



από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	HorizontalArrangement	ButtonArea	Width : Fill Parent AlignHorizontal : Center
UserInterface	Button	CoinButton	Image : 1.jpg Visible : hidden Width : 150 pixels Height : 150 pixels
Media	Player	AppPlayer	

Βήμα 4 Προσθήκη ετικέτας με μήνυμα επιτυχίας

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



από την
ομάδα

UI

μεταφέρουμε το
αντικείμενο

Label

του δίνουμε το
όνομα

WinMessageLabel

μεταβάλλουμε τις
ιδιότητες

Font Size : 40

Text Alignment : center

Text :

Width : Fill Parent

Βήμα 5 Επιλογή χρήστη και στρίψιμο νομίσματος

Συγγραφή Προγράμματος

Blocks

Αρχικά θα δημιουργήσουμε μια μεταβλητή που θα αποθηκεύει την επιλογή του χρήστη. Η μεταβλητή αυτή θα παίρνει την τιμή 1 όταν ο χρήστης επιλέγει κορώνα και την τιμή 2 όταν ο χρήστης επιλέγει γράμματα.

Θα ονομάσουμε τη νέα μας μεταβλητή choice και θα της δώσουμε αρχικά την τιμή 0.

Επίσης θα χρειαστούμε μια μεταβλητή coin που θα αποθηκεύει το αποτέλεσμα από το στρίψιμο του νομίσματος, δηλαδή τον τυχαίο αριθμό 1 ή 2. Η μεταβλητή coin θα έχει και αυτή αρχικά την τιμή 0.

Στη συνέχεια θα πρέπει να υλοποιήσουμε την παρακάτω συμπεριφορά κατά το άγγιγμα των κουμπιών.

Όταν ο χρήστης επιλέγει το κουμπί της κορώνας, θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή choice θα παίρνει την τιμή 1.
- Το κουμπί επιλογής για τα γράμματα θα κρύβεται.
- Το κουμπί επιλογής για την κορώνα θα απενεργοποιείται (ιδιότητα Enabled) ώστε ο χρήστης να μην μπορεί να το ξαναεπιλέξει.
- Το κουμπί για το στρίψιμο του νομίσματος θα ενεργοποιείται.
- Το κείμενο της ετικέτας για το αποτέλεσμα του στρίψιματος θα γίνεται το κενό «».

Όταν ο χρήστης επιλέγει το κουμπί γράμματα θα γίνονται οι παρακάτω ενέργειες:

- Η μεταβλητή choice θα παίρνει την τιμή 2.
- Το κουμπί επιλογής για την κορώνα θα κρύβεται.
- Το κουμπί επιλογής για τα γράμματα θα απενεργοποιείται (ιδιότητα Enabled) ώστε ο χρήστης να μην μπορεί να το ξαναεπιλέξει.
- Το κουμπί για το στρίψιμο του νομίσματος θα ενεργοποιείται.

- Το κείμενο της ετικέτας για το αποτέλεσμα του στριψίματος θα γίνεται το κενό «».

Τέλος, θα προγραμματίσουμε τη συμπεριφορά του κουμπιού για το στρίψιμο του νομίσματος.

Όταν ο χρήστης αγγίζει το κουμπί για το στρίψιμο του νομίσματος θα πρέπει να γίνονται οι παρακάτω ενέργειες.

- Η μεταβλητή coin θα παίρνει ως νέα τιμή έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 2.
- Θα αναπαράγεται ο ήχος του στριψίματος του νομίσματος (θα πρέπει να θέσετε την πηγή του ήχου στο κατάλληλο ηχητικό αρχείο).
- Η εικόνα του κουμπιού του νομίσματος θα γίνεται κορώνα ή γράμματα ανάλογα με το αποτέλεσμα στη μεταβλητή coin.
- Αν η επιλογή του χρήστη και το αποτέλεσμα του στριψίματος είναι τα ίδια τότε θα εμφανίζεται το μήνυμα «Κέρδισες» στην ετικέτα και θα αναπαράγεται ο ήχος της επιτυχίας (θα πρέπει να θέσετε την πηγή του ήχου στο κατάλληλο ηχητικό αρχείο).
- Τα κουμπιά για την επιλογή Κορώνα ή Γράμματα θα είναι και τα δύο ορατά και ενεργοποιημένα.
- Το κουμπί για το στρίψιμο του νομίσματος θα απενεργοποιείται.



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

Μπορείτε να δείτε την μορφή της εφαρμογής μετά τις παραπάνω προσθήκες ανατρέχοντας στο [Παράρτημα με τις Λύσεις των Ασκήσεων](#).

Βήμα 6

Πέρα από την εντολή if – Η εντολή else

Παρατηρήστε ότι η εφαρμογή εμφανίζει μήνυμα επιτυχίας όταν ο χρήστης κερδίζει, αλλά δεν κάνει τίποτα σε περίπτωση που χάνει.

Θα την επεκτείνουμε ώστε όταν ο χρήστης δεν μαντεύει σωστά να του εμφανίζει το μήνυμα «Έχασες» αναπαράγοντας και έναν κατάλληλο ήχο αποτυχίας.

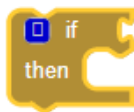
Η ΕΝΤΟΛΗ ELSE (ΑΛΛΙΩΣ)



Πολύ συχνά χρειάζεται να περιγράψουμε τις ενέργειες που θα εκτελεστούν όταν μια λογική πρόταση είναι αληθής και ταυτόχρονα να περιγράψουμε κάποιες άλλες ενέργειες όταν η **ίδια** λογική πρόταση ΔΕΝ ισχύει.

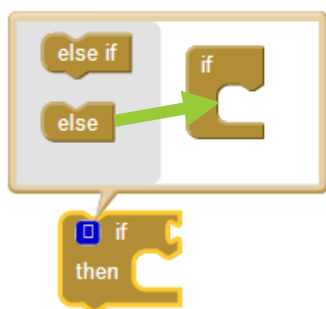
Για να το πετύχουμε θα πρέπει να επεκτείνουμε την εντολή if προσθέτοντας της την εντολή **else**.

Για να προσθέσουμε την εντολή `else` κάτω από την εντολή `if` κάνουμε κλικ στο μπλε τετράγωνο που βρίσκεται πάνω στην εντολή `if`, όπως φαίνεται παρακάτω.



Εικόνα 76 - Επεκτείνοντας την εντολή `if`

Θα ανοίξει ένα παράθυρο από το οποίο θα κουμπώσουμε την εντολή `else` κάτω από την εντολή `if`, σέρνοντας την με το ποντίκι μας ακριβώς κάτω από την `if`, όπως φαίνεται στο σχήμα.



Εικόνα 77 - Κουμπώνοντας το `else` κάτω από το `if`

Στη συνέχεια θα περιγράψουμε τις ενέργειες που θα γίνουν όταν ο χρήστης ΔΕΝ μαντέψει σωστά το νόμισμα και θα τις κουμπώσουμε κάτω από την εντολή `else`.

- Το κείμενο της ετικέτας θα γίνεται «Έχασες...»
- Θα αναπαράγεται ο ήχος αποτυχίας.

Η τελική μορφή της εντολής `if...else` φαίνεται παρακάτω.



Εικόνα 78 - Η εντολή `if...else`



Αν το νόμισμα και η επιλογή είναι τα ίδια, τότε θέσε το μήνυμα σε «Κέρδισες» και τον ήχο στο αρχείο `success`. Διαφορετικά, θέσε το μήνυμα σε «Έχασες» και τον ήχο στο αρχείο `failure`.
Παίξε τον ήχο.



Γιατί η εντολή `call [AppPlayer].Start` μετακινήθηκε μετά την εντολή `if...else ;`



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

Βήμα 7 Animation – Το νόμισμα που γυρίζει

Θα προσθέσουμε ένα απλό animation που θα αλλάζει την εικόνα του κουμπιού για το στρίψιμο του νομίσματος, έτσι ώστε να δίνει την αίσθηση ότι το νόμισμα γυρίζει.

Για το σκοπό αυτό θα χρειαστεί να προσθέσουμε 2 ρολόγια στην εφαρμογή. Το πρώτο θα καθορίζει τη συνολική διάρκεια του animation, για παράδειγμα 1 δευτερόλεπτο ενώ το 2^ο ρολόι θα καθορίζει κάθε πότε θα γίνεται η εναλλαγή της εικόνας του κουμπιού, για παράδειγμα κάθε 50 χιλιοστά του δευτερολέπτου.

Designer

Γραφικό Περιβάλλον Εφαρμογής



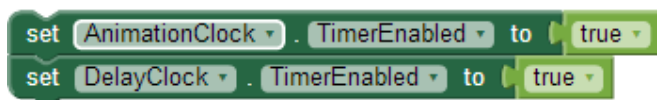
από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
UserInterface	Clock	DelayClock	TimerEnabled : OXI TimerInterval : 1000
UserInterface	Clock	AnimationClock	TimerEnabled : OXI TimerInterval : 50

Συγγραφή Προγράμματος

Blocks

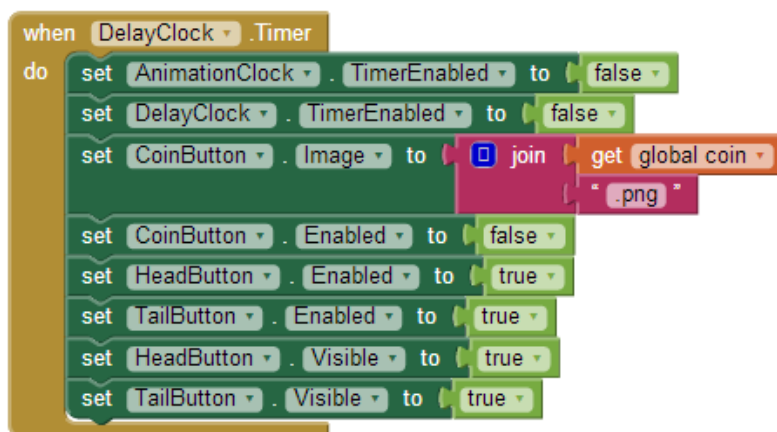
Αρχικά τα δύο ρολόγια είναι απενεργοποιημένα. Η ενεργοποίησή τους θα γίνεται κάθε φορά που ο παίκτης θα αγγίζει το κουμπί της ρίψης του νομίσματος. Όταν τελειώσει το animation μετά το πέρας 1 δευτερολέπτου τα ρολόγια θα απενεργοποιούνται ξανά.

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



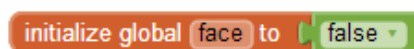
Εικόνα 79 - Εντολές ενεργοποίησης των ρολογιών

Οι εντολές που θα εκτελεί το ρολόι DelayClock θα είναι η απενεργοποίηση των ρολογιών και η ενεργοποίηση των υπολοίπων αντικειμένων του παιχνιδιού ώστε ο χρήστης να μπορεί να ξαναπαίξει. Οι εντολές αυτές **θα μετακινηθούν από το γεγονός του αγγίγματος του κουμπιού, στο γεγονός της «πυροδότησης» του ρολογιού**, όπως φαίνεται παρακάτω.



Εικόνα 80 - Η πυροδότηση του πρώτου ρολογιού

Το ρολόι AnimationClock θα μας βοηθήσει στην εναλλαγή των εικόνων του κουμπιού. Θα χρειαστεί να δημιουργήσουμε μια νέα μεταβλητή με όνομα face που θα παίρνει τις τιμές True/False. Θα τις δώσουμε αρχικά την τιμή false, όπως παρακάτω.

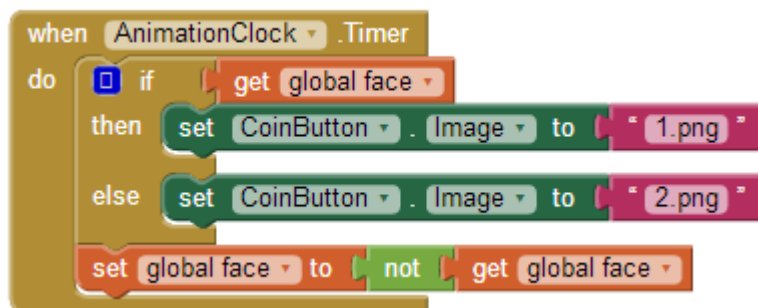


Εικόνα 81 - Αρχικοποίηση της μεταβλητής face

Στη συνέχεια κάθε φορά που θα πυροδοτείται το ρολόι AnimationClock θα γίνονται οι παρακάτω ενέργειες :

- Αν η μεταβλητή face έχει την τιμή false, στο κουμπί θα εμφανίζεται η εικόνα της κορώνας. Διαφορετικά θα εμφανίζεται η εικόνα των γραμμάτων.
- Η μεταβλητή face θα παίρνει την αντίθετη τιμή από αυτή που έχει. Δηλαδή αν είναι true θα γίνεται false, ενώ αν είναι false θα γίνεται true.

Οι απαραίτητες εντολές φαίνονται παρακάτω :



Εικόνα 82 - Η πυροδότηση του ρολογιού για το Animation

Ο ΛΟΓΙΚΟΣ ΤΕΛΕΣΤΗΣ NOT (ΌΧΙ)



Προκειμένου να αντιστρέψουμε την τιμή μιας λογικής μεταβλητής χρησιμοποιούμε τον τελεστή not που βρίσκεται στην ομάδα Logic. Η λειτουργία του είναι πολύ απλή. Αν μια λογική μεταβλητή έχει την τιμή true τότε βάζοντας μπροστά της τον λογικό τελεστή not παίρνει την τιμή false, και αντιστρόφως.

not true = false

not false = true

Εικόνα 83 - Ο λογικός τελεστής not

Εικόνα 84 - Η λειτουργία του τελεστή not

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ ΓΙΑ ΤΟ ΣΠΙΤΙ

1. Τροποποιήστε την εφαρμογή Ζάρια, ώστε να εμφανίζει σε μια ετικέτα το συνολικό αριθμό παιχνιδιών που έχει παίξει ο παίκτης καθώς και το ποσοστό των παιχνιδιών που ο παίκτης έφερε διπλές.

$$\text{Ποσοστό κερδισμένων παιχνιδιών} = \text{κερδισμένα παιχνίδια} / \text{συνολικά παιχνίδια} * 100$$

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

Η εφαρμογή θα εμφανίζει έναν έναν τους αριθμούς των δευτερολέπτων μετρώντας αντίστροφα. Όταν τα δευτερόλεπτα μηδενιστούν, η εφαρμογή θα 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

10 ... 9 ... 8 ... 7 ... 6 ... 5 ... 4 ... 3 ... 2 ... 1 ... 0

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.

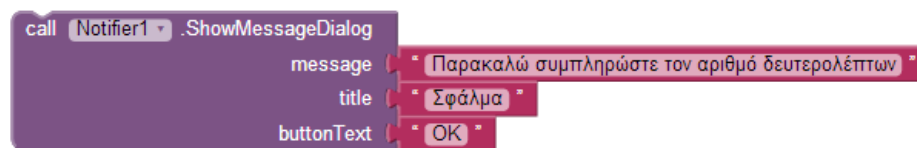


Θα χρειαστείτε το αντικείμενο Clock για να μετρά το πέρασμα 1 δευτερολέπτου.

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 UserInterface. Στο τμήμα των εντολών θα χρησιμοποιήσετε την εντολή που φαίνεται παρακάτω:

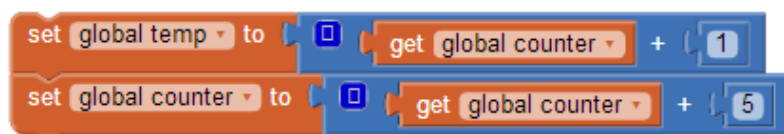


Για τον έλεγχο θα σας φανεί χρήσιμο το παρακάτω πλακίδιο της ομάδας Text, το οποίο επιστρέφει την τιμή Αληθές (true) εάν το κείμενο που περιέχεται στα δεξιά του έχει μηδενικό μήκος (δηλαδή είναι κενό):



3. Η ιδιότητα Source του αντικειμένου Player μπορεί να πάρει τιμή με δύο διαφορετικούς τρόπους. Να τους περιγράψετε και να καταγράψετε σε ποιες περιπτώσεις προτιμάμε τον έναν ή τον άλλον.

4. Έχουμε δύο μεταβλητές με ονόματα counter και temp. Η αρχική τους τιμή είναι 100 και 0 αντίστοιχα. Ποια θα είναι η τιμή των παρακάτω μεταβλητών μετά την εκτέλεση των εντολών που φαίνονται παρακάτω;



5. Έχουμε μια μεταβλητή με όνομα counter και αρχική τιμή 50, όπως φαίνεται στην παρακάτω εικόνα.



Έστω ότι πατάμε το κουμπί Button1 δύο φορές. Την πρώτη φορά ο τυχαίος αριθμός είναι το 60 και τη δεύτερη φορά ο τυχαίος αριθμός είναι το 10. Ποια θα είναι η τιμή της μεταβλητής counter μετά το δεύτερο πάτημα του κουμπιού; Να εξηγήσετε αναλυτικά την απάντησή σας.

6. Αναπτύξτε μια εφαρμογή που θα κάνει έναν υπολογισμό της αρεσκειάς σας. Ένα παράδειγμα είναι η μετατροπή μιας μονάδας μέτρησης σε κάποια άλλη, όπως μέτρα σε χιλιόμετρα, μέρες σε ώρες κλπ. Ο χρήστης θα δίνει μια τιμή σαν είσοδο και η εφαρμογή θα του εμφανίζει το αποτέλεσμα. Φροντίστε να εμπλουτίσετε την εφαρμογή σας με κάποια γραφικά και ήχους.

Δοκιμάστε να φτιάξετε δύο εκδοχές της εφαρμογής. Στην πρώτη ο χρήστης θα γράφει την τιμή μέσα σε ένα πεδίο κειμένου (textbox) και θα ζητά να γίνεται ο υπολογισμός με το άγγιγμα ενός κουμπιού. Θα πρέπει βέβαια να γίνεται έλεγχος ότι έχει γράψει μια τιμή μέσα στο πεδίο κειμένου, διαφορετικά να ενημερώνεται με μήνυμα σφάλματος.

Στη δεύτερη εκδοχή θα χρησιμοποιεί το αντικείμενο του μεταβολέα (slider). Με κάθε αλλαγή στην τιμή του μεταβολέα θα υπολογίζεται αυτομάτως και το νέο αποτέλεσμα.

Παρακάτω δίνεται σαν παράδειγμα μια εφαρμογή που μετατρέπει βαθμούς Κελσίου σε Φαρενάιτ.

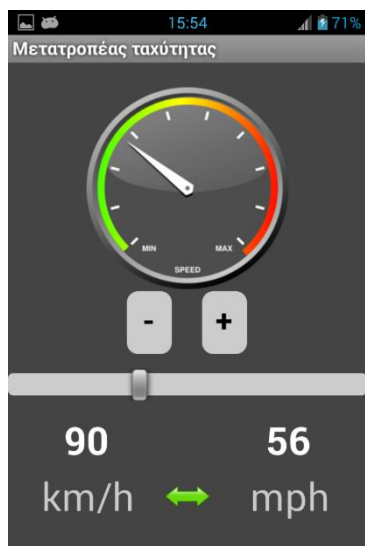
Εκδοχή 1^η με πεδίο κειμένου



Εκδοχή 2^η με μεταβολέα



Ενδεικτικό στιγμιότυπο μιας αντίστοιχης εφαρμογής, η οποία μετατρέπει ταχύτητες από χιλιόμετρα/ώρα σε μίλια/ώρα, φαίνεται παρακάτω. Η συγκεκριμένη εκδοχή διαθέτει μεταβολέα (slider) και επιπλέον κουμπιά για ακριβέστερη αυξομείωση της ταχύτητας. Για το συγχρονισμό των κουμπιών και του μεταβολέα χρησιμοποιήθηκε μεταβλητή.



7. Αναπτύξτε το παιχνίδι του κουλοχέρη. Σκοπός του παιχνιδιού είναι ο παίκτης να καταφέρει να κερδίσει πόντους - χρήματα με βάση τους κανόνες που περιγράφονται παρακάτω. Σε κάθε προσπάθεια ο παίκτης καταχωρεί το ποσό που θα ποντάρει και στη συνέχεια «τραβάει» το μοχλό του κουλοχέρη. Ανάλογα με τις εικόνες που θα εμφανιστούν στον κουλοχέρη, ο παίκτης κερδίζει κάποιο ποσό ή όχι.

Κανόνες παιχνιδιού

- Αν και οι 3 εικόνες είναι  τότε ο παίκτης κερδίζει ποσό ίσο με το 10πλάσιο του ποσού που ποντάρισε.
- Αν και οι 3 εικόνες είναι οι ίδιες (αλλά όχι το 7, όπως προηγουμένως), τότε ο παίκτης κερδίζει ποσό ίσο με το 5πλάσιο του ποσού που ποντάρισε.
- Αν 2 από τις 3 εικόνες είναι ίδιες τότε ο παίκτης κερδίζει ποσό ίσο με το 2πλάσιο του ποσού που ποντάρισε.
- Σε κάθε άλλη περίπτωση ο παίκτης δεν κερδίζει τίποτα.

Σε κάθε περίπτωση βέβαια ο παίκτης χάνει το ποσό που ποντάρισε.

Η εφαρμογή που θα φτιάξετε θα πρέπει να ελέγχει κάθε φορά που ο παίκτης «τραβάει» το μοχλό του κουλοχέρη ότι έχει δώσει ποσό πονταρίσματος.

Επίσης θα πρέπει να τον ενημερώνει κάθε φορά για το συνολικό ποσό που έχει κερδίσει ή που έχει χάσει. Φροντίστε να βάλετε animation στο γύρισμα των εικόνων και το τράβηγμα του μοχλού.

Μπορείτε να κατεβάσετε μια πρώτη έκδοση της εφαρμογής με έτοιμο το γραφικό της περιβάλλον από τη διεύθυνση : <http://tiny.cc/aifiles> και το φάκελο Slot Machine.



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