

App Inventor



ΜΑΘΗΜΑ 2

ΣΤΟΧΟΙ

- Να δημιουργείτε μεταβλητές και να τους δίνετε αρχικές τιμές.
- Να αλλάζετε την τιμή των μεταβλητών μέσα στην εφαρμογή.
- Να χρησιμοποιείτε μεταβλητές για να υπολογίσετε το πλήθος κάποιων συμβάντων (μετρητές).
- Να χρησιμοποιείτε τις λογικές τιμές True/False (Αληθής/Ψευδής).
- Να ελέγχετε την τιμή λογικών προτάσεων για να παίρνετε αποφάσεις με την εντολή If (Αν).
- Να χρησιμοποιείτε το αντικείμενο Clock (ρολόι) για την παραγωγή animation – κινούμενης εικόνας.
- Να χρησιμοποιείτε το αντικείμενο Clock (ρολόι) για την εκτέλεση ενεργειών ανά τακτά χρονικά διαστήματα.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΑΡΙΑ – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

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

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

Βήμα 1 Άνοιγμα του έργου «Ζάρια»

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

Βήμα 2 Προσθήκη ετικέτας με μήνυμα επιτυχίας

Στο βήμα 2 θα προσθέσουμε μια ετικέτα κειμένου που θα ενημερώνει το χρήστη για το αν κέρδισε.



Οι ετικέτες κειμένου (text labels) είναι χρήσιμες για να εμφανίζουμε μηνύματα κειμένου σε διάφορα σημεία της οθόνης της εφαρμογής.

Designer

Γραφικό Περιβάλλον Εφαρμογής

Καταρχήν θα προσθέσουμε στο γραφικό περιβάλλον της εφαρμογής ένα αντικείμενο **Label** (ετικέτα) από την ομάδα αντικειμένων *User Interface*.



από την
ομάδα

UserInterface

μεταφέρουμε το
αντικείμενο

Label

του δίνουμε το
όνομα

ResultLabel

μεταβάλλουμε τις
ιδιότητες

Font Size : 40

Text Alignment : center

Text :

Width : Fill Parent

Βήμα 3 Προσθήκη ετικέτας για το σκορ

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



από την
ομάδα

UIInterface

μεταφέρουμε το
αντικείμενο

Label

του δίνουμε το
όνομα

ScoreLabel

μεταβάλλουμε τις
ιδιότητες

Font Size : 20

TextColor : White

Text : Σκορ : 0

Width : Fill Parent

Συγγραφή Προγράμματος

Blocks

Η ΕΝΝΟΙΑ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ

Για να μπορέσουμε να συγκρίνουμε το αποτέλεσμα των δύο ζαριών θα αποθηκεύσουμε τους δύο τυχαίους αριθμούς που παράγονται από την εντολή **random integer from...**



Για το σκοπό αυτό θα χρειαστούμε μια μεταβλητή, δηλαδή ένα χώρο μνήμης στον υπολογιστή μας που μπορεί να αποθηκεύσει μια τιμή. Η αναφορά σε αυτόν τον χώρο γίνεται με ένα συμβολικό όνομα που επιλέγουμε εμείς.

Μπορείτε να φανταστείτε την μεταβλητή σαν ένα πολύ μικρό σάκο που χωράει μονάχα ένα αντικείμενο. Μπορούμε να μεταβάλλουμε το περιεχόμενο του σάκου, δηλαδή να αλλάξουμε την τιμή μιας μεταβλητής, αντικαθιστώντας πρακτικά το προηγούμενο αντικείμενο που έχει, με το νέο αντικείμενο που θα τοποθετήσουμε.

Παρακάτω φαίνεται ένα τέτοιο παράδειγμα. Έστω ότι έχουμε μια μεταβλητή με όνομα σκορ στην οποία δίνουμε αρχικά την τιμή 0. Στη συνέχεια της δίνουμε την τιμή 50, οπότε η προηγούμενη τιμή της (δηλαδή το 0) αντικαθίσταται από τη νέα τιμή.

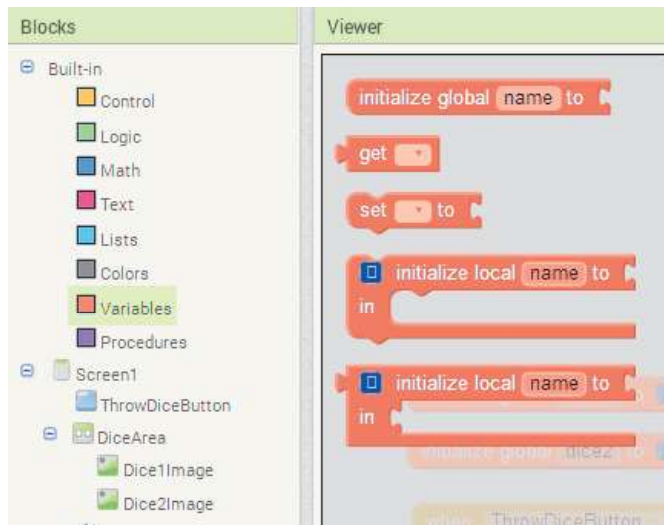


Εικόνα 54 - Η έννοια της μεταβλητής

Βήμα 4 Δημιουργία μεταβλητών για τα ζάρια

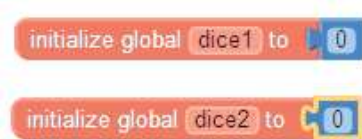
Στην εφαρμογή μας θα χρειαστούμε δύο μεταβλητές, κάθε μια για να αποθηκεύσει τον αριθμό του αντίστοιχου ζαριού.

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



Εικόνα 55 - Εντολές στην ομάδα Variables

Στο πεδίο [name] της εντολής γράφουμε το όνομα που θέλουμε να δώσουμε σε κάθε μεταβλητή. Στην εφαρμογή μας θα ονομάσουμε την πρώτη μεταβλητή dice1 και τη δεύτερη μεταβλητή dice2, όπως στην παρακάτω εικόνα. Τέλος, στο αντίστοιχο κενό θα κουμπώσουμε τον αριθμό 0 (από την ομάδα Math), που θα αποτελέσει την αρχική, δηλαδή την πρώτη τιμή των μεταβλητών.



Εικόνα 56 - Δίνοντας αρχικές τιμές σε μεταβλητές

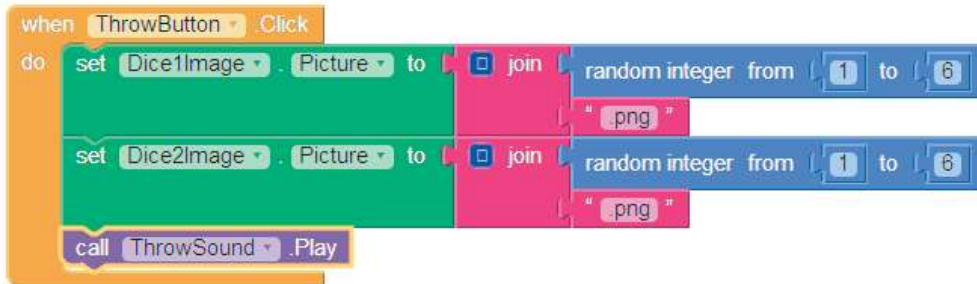


Δημιούργησε μια μεταβλητή με όνομα dice1 και εκχώρησε σαν αρχική τιμή της μεταβλητής αυτής το 0.

Δημιούργησε μια μεταβλητή με όνομα dice2 και εκχώρησε σαν αρχική τιμή της μεταβλητής αυτής το 0.

Βήμα 5 Ανάθεση αριθμών ζαριών στις μεταβλητές

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



Εικόνα 57 - Το αρχικό πρόγραμμα της εφαρμογής Ζάρια

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



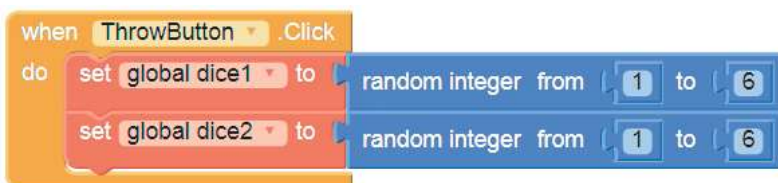
Εικόνα 58 - Ανάθεση τιμής σε μεταβλητή



Θέσε την τιμή της μεταβλητής dice1 σε έναν τυχαίο ακέραιο αριθμό από 1 μέχρι 6.

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

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



Εικόνα 59 - Ανάθεση τιμών στις μεταβλητές με το άγγιγμα του κουμπιού

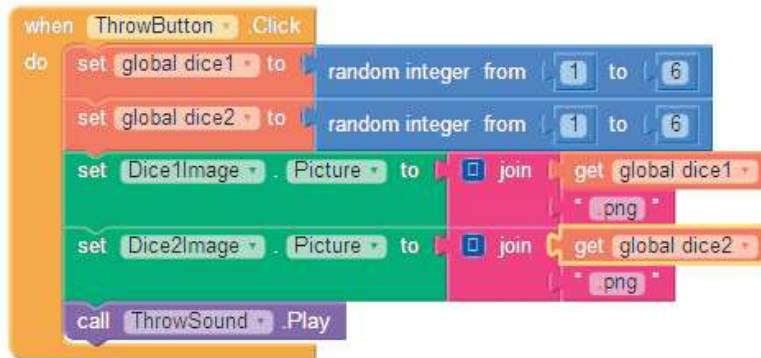
Τέλος, στην εντολή που θέτει την κατάλληλη εικόνα του ζαριού **set Dice1Image.Picture to...** θα αντικαταστήσουμε στην εντολή **join** τον τυχαίο αριθμό, με την τιμή της κάθε μεταβλητής.

Για να χρησιμοποιήσουμε την τρέχουσα τιμή μιας μεταβλητής χρησιμοποιούμε την εντολή **get []**, που βρίσκεται στην ομάδα εντολών **Variable**, επιλέγοντας το όνομα της μεταβλητής από τη λίστα που ακολουθεί την εντολή.



Εικόνα 60 - Εντολή get

Το πρόγραμμα μας πρέπει να μοιάζει με την παρακάτω εικόνα.



Εικόνα 61 - Η εφαρμογή Ζάρια μετά την προσθήκη μεταβλητών

Βήμα 6 Έλεγχος για εξάρες

Η ΕΝΤΟΛΗ IF (ΑΝ)

Για να δώσουμε τη δυνατότητα στην εφαρμογή μας να ανιχνεύει το 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.



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



Εικόνα 62 - Η εντολή If

Η εντολή Αν λειτουργεί ως εξής : ελέγχει μια λογική πρόταση, δηλαδή μια πρόταση που μπορεί να είναι είτε Αληθής είτε Ψευδής (να ισχύει ή να μην ισχύει). Σε περίπτωση που η πρόταση είναι Αληθής τότε εκτελούνται οι εντολές που «περιέχονται» κάτω από την εντολή Αν. Διαφορετικά, οι εντολές αυτές αγνοούνται και ΔΕΝ εκτελούνται.

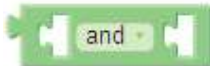
Η ΛΟΓΙΚΗ ΠΡΟΤΑΣΗ

Η πρώτη μας ενέργεια είναι να σχεδιάσουμε τη λογική πρόταση που θέλουμε να εξετάσουμε. Καταρχάς ελέγχουμε αν η μεταβλητή dice1 έχει την τιμή 6, όπως φαίνεται παρακάτω.



Εικόνα 63 - Έλεγχος τιμής μεταβλητής

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



Εικόνα 64 - Ο λογικός τελεστής AND



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



Εικόνα 65 - Ένωση προτάσεων με το AND

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



Εικόνα 66 - Η εντολή if με ενσωματωμένη τη λογική πρόταση

ΟΙ ΕΝΕΡΓΕΙΕΣ ΚΑΤΩ ΑΠΟ ΤΗΝ ΕΝΤΟΛΗ IF

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

Για να θέσουμε το κείμενο της ετικέτας χρησιμοποιούμε την εντολή **set [ResultLabel].Text to...** την οποία θα βρούμε μόλις επιλέξουμε το αντικείμενο ResultLabel.



Εικόνα 67 - Η εντολή για να θέσουμε το κείμενο μιας ετικέτας

Θα κουμπώσουμε στην εντολή το κείμενο «Κέρδισες!». Η τελική μορφή της εντολής if θα μοιάζει όπως παρακάτω.



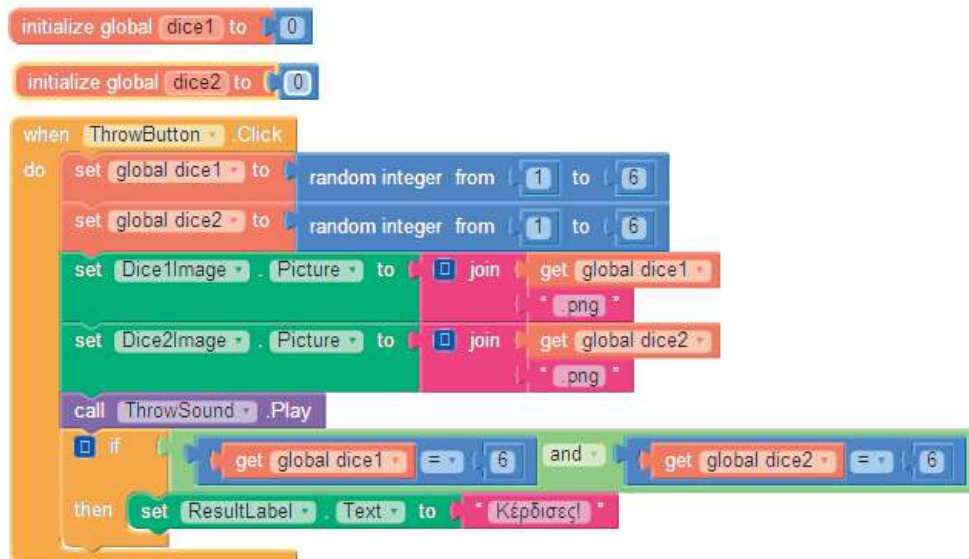
Εικόνα 68 - Η τελική μορφή της εντολής if



Αν η τιμή της μεταβλητής dice1 είναι 6 ΚΑΙ η τιμή της μεταβλητής dice2 είναι 6, τότε θέσε το κείμενο της ετικέτας σε «Κέρδισες!»



Όρα να ελέγξουμε την εφαρμογή με την κινητή μας συσκευή.



Εικόνα 69 - Η εφαρμογή Ζάρια με έλεγχο για τις εξάρες

Βήμα 7 Προσθήκη σκορ στην εφαρμογή

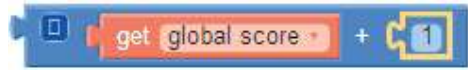
Καλό θα είναι η εφαρμογή να δείχνει στον χρήστη της πόσες φορές έχει φέρει εξάρες, δηλαδή να μετράει σε πόσα παιχνίδια κέρδισε.

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



Εικόνα 70 - Αρχικοποίηση μεταβλητής score

Στη συνέχεια θα πρέπει, όταν ο χρήστης φέρνει εξάρεις, η τιμή της μεταβλητής να αυξάνεται κατά 1. Καταρχήν φτιάχνουμε την εντολή αύξησης της τιμής της μεταβλητής κατά 1.



Εικόνα 71 - Αύξηση τιμής score κατά 1

Θέτουμε το παραπάνω σαν νέα τιμή της μεταβλητής score.



Εικόνα 72 - Ανάθεση νέας τιμής στην μεταβλητή score



Αύξησε την τιμή της μεταβλητής score κατά 1.

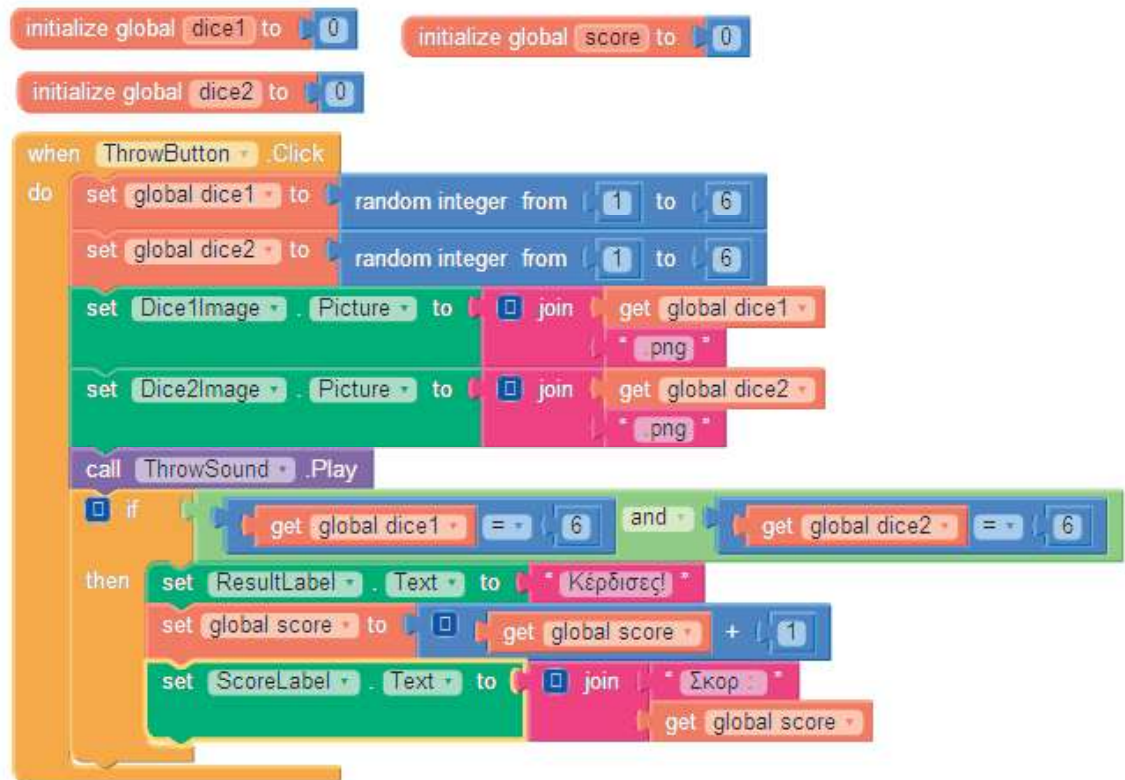
Τέλος τοποθετούμε την εντολή κάτω από την εντολή if.

Στη συνέχεια για να εμφανίζεται το σκορ στην ετικέτα ScoreLabel χρησιμοποιούμε την εντολή **set [ScoreLabel].Text to...** την οποία θα βρούμε μόλις επιλέξουμε το αντικείμενο ScoreLabel. Το κείμενο που θα εμφανίζεται θα είναι η λέξη «Σκορ :» και η τιμή της μεταβλητής score. Η τελική μορφή της εντολής θα είναι η παρακάτω.



Εικόνα 73 - Αλλάζοντας το κείμενο της ετικέτας για το σκορ

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



Εικόνα 74 - Η εφαρμογή ζάρια με σκορ



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

ΕΠΕΚΤΑΣΕΙΣ

1. Παρατηρήστε ότι όταν ο χρήστης της εφαρμογής φέρει εξάρες και στη συνέχεια ξαναπαίξει, η ετικέτα συνεχίζει να δείχνει το κείμενο «Κέρδισες!». Κάντε τις απαραίτητες ενέργειες, ώστε όταν ο παίκτης ρίχνει τα ζάρια το κείμενο της ετικέτας να γίνεται το κενό («»).
2. Τροποποιήστε την εφαρμογή, έτσι ώστε να αυξάνει το σκορ του παίκτη όχι μόνο στην περίπτωση που φέρει εξάρες, αλλά κάθε φορά που φέρνει διπλές, δηλαδή τον ίδιο αριθμό και στα δύο ζάρια.
3. Προσθέστε ένα νέο κουμπί στην εφαρμογή με τίτλο «Μηδένισε σκορ» ή προσθέστε στο ήδη υπάρχον κουμπί «Ρίξε ζάρια» (π.χ. με παρατεταμένο πάτημα longClick) τη δυνατότητα ο χρήστης να μηδενίζει το σκορ.

ΔΡΑΣΤΗΡΙΟΤΗΤΑ : GUARDDOG – ΕΠΕΚΤΑΣΗ

ΠΕΡΙΓΡΑΦΗ ΕΠΕΚΤΑΣΗΣ

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

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

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

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

Βήμα 1 Άνοιγμα υπάρχοντος project

Ανοίγουμε το ήδη υπάρχον project με ονομασία GuardDog.

Βήμα 2 Αντίδραση του σκύλου στο σήκωμα της συσκευής

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

Designer Γραφικό Περιβάλλον Εφαρμογής

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



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

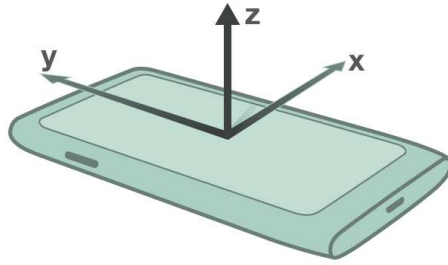
μεταβάλλουμε τις
ιδιότητες

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.



Άξονας X: Ανιχνεύει κίνηση προς τα δεξιά ή προς τα αριστερά.

Άξονας Y: Ανιχνεύει κίνηση προς τα εμπρός ή προς τα πίσω.

Άξονας Z: Ανιχνεύει κίνηση προς τα πάνω ή προς τα κάτω.

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 ρολογιού είναι 3000 msec, δηλαδή ορίζουμε να τερματίζει (και να ξεκινάει ξανά) κάθε 3 δευτερόλεπτα.

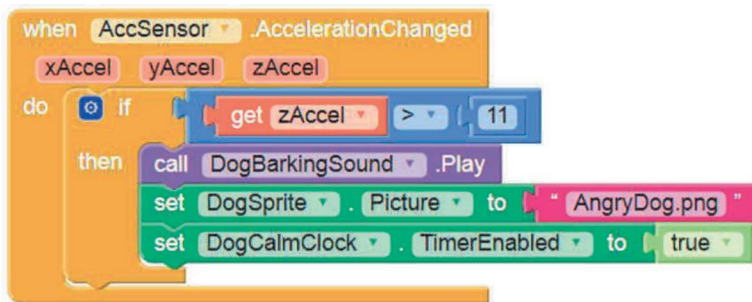
Συγγραφή Προγράμματος

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.