

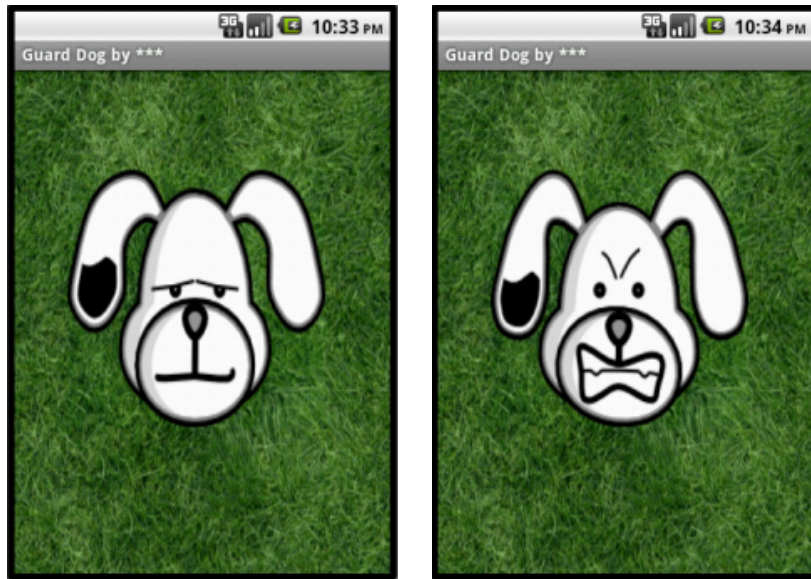


GuardDog Φύλλο Εργασίας

Προγραμματισμός Εφαρμογής στο AppInventor
<http://ai2.appinventor.mit.edu>

Ακολουθώντας τα βήματα αυτού του φύλλου εργασίας, θα αναπτύξουμε μια μικρή εφαρμογή για Android συσκευές (κινητά ή tablets), η οποία θα μας εξοικειώσει με χρήσιμες λειτουργίες που θα αξιοποιήσουμε και στη συνέχεια.

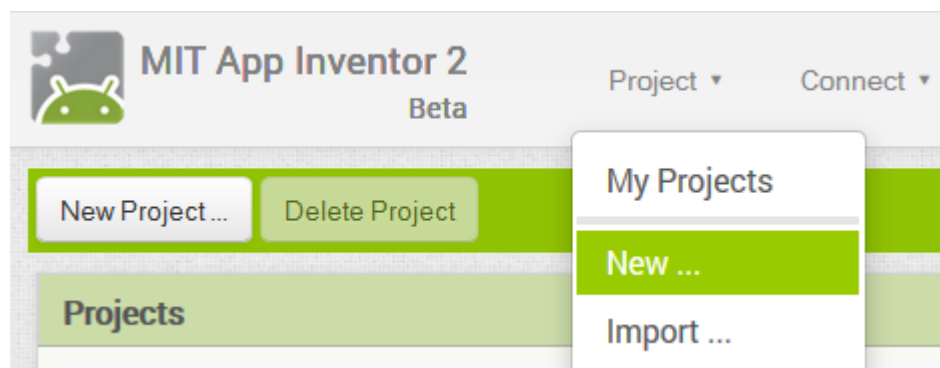
Θα αναπτύξουμε σταδιακά μια **εφαρμογή «συναγερμό»**, χρησιμοποιώντας έναν σκύλο που δεν θα επιτρέπει σε κανέναν να πειράζει την συσκευή μας.



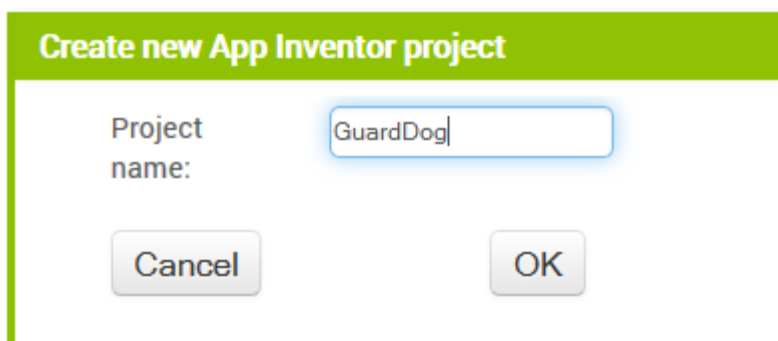
Η εφαρμογή που θα κατασκευάσουμε είναι απλή, πρόκειται για ένα εισαγωγικό παράδειγμα. Οι οδηγίες που ακολουθούν είναι αναλυτικές, με σκοπό να προσφέρουν μια στοιχειώδη εξοικείωση με το περιβάλλον του AppInventor.

Βήμα 1 Δημιουργία νέου project

Ξεκινώντας, δημιουργούμε ένα καινούργιο project...



...το οποίο ονομάζουμε GuardDog.

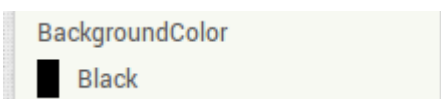


Αρχικά, βρισκόμαστε στην ενότητα **Designer**, στην οποία σχεδιάζουμε τη διεπαφή (interface) της εφαρμογής μας, προσθέτοντας τα απαραίτητα συστατικά (components) και ορίζοντας ιδιότητες (properties) για αυτά.

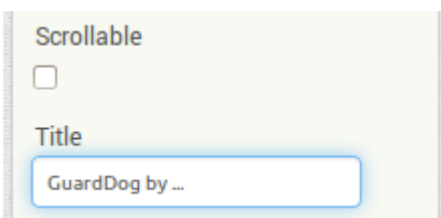
Το μοναδικό συστατικό μέχρι στιγμής, είναι η οθόνη (Screen1). Προτού προσθέσουμε άλλα συστατικά, κάνουμε ορισμένες απαραίτητες τροποποιήσεις στις ιδιότητες της οθόνης, που βρίσκονται στο πλαίσιο **Properties**, στα δεξιά.



Επιλέγουμε το χρώμα του φόντου (BackgroundColor) να είναι μαύρο.



Η ιδιότητα Scrollable δεν πρέπει να είναι επιλεγμένη. Στον τίτλο (Title) προσθέτουμε το όνομα της ομάδας μας, τα ονόματά μας ή τα αρχικά μας.



Οι τροποποιήσεις στις ιδιότητες της οθόνης Screen1 συνοψίζονται ως εξής:



επιλέγουμε το συστατικό	μεταβάλλουμε τις ιδιότητες
Screen1	BackgroundColor: Black Scrollable: No Title: GuardDog by ...

Βήμα 2 Προσθήκη του καμβά

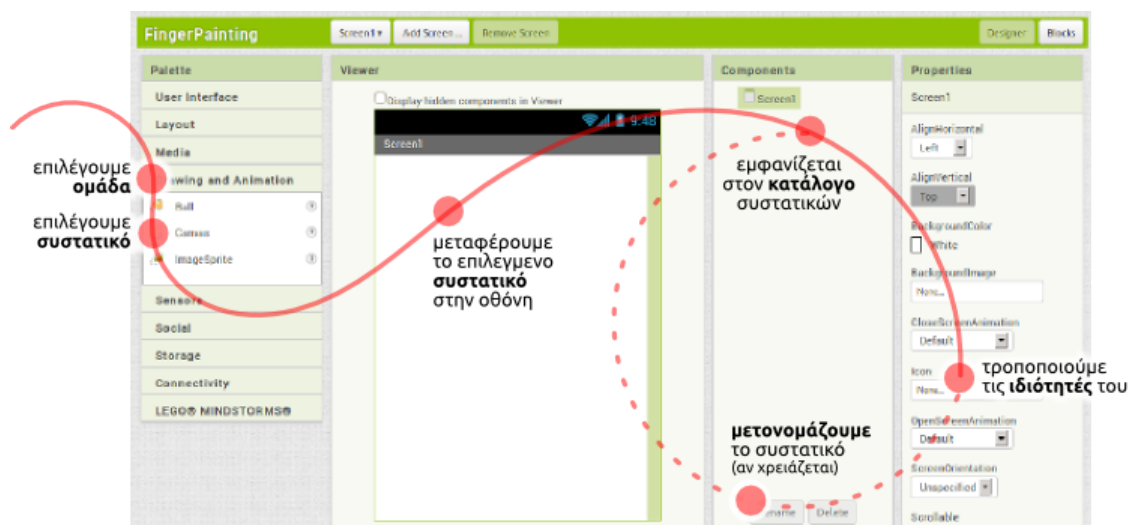
Στο βήμα αυτό θα προσθέσουμε το συστατικό που θα αποτελέσει το χώρο όπου θα κινείται ο σκύλος και θα καθορίσουμε τις ιδιότητες του χώρου αυτού.

Designer

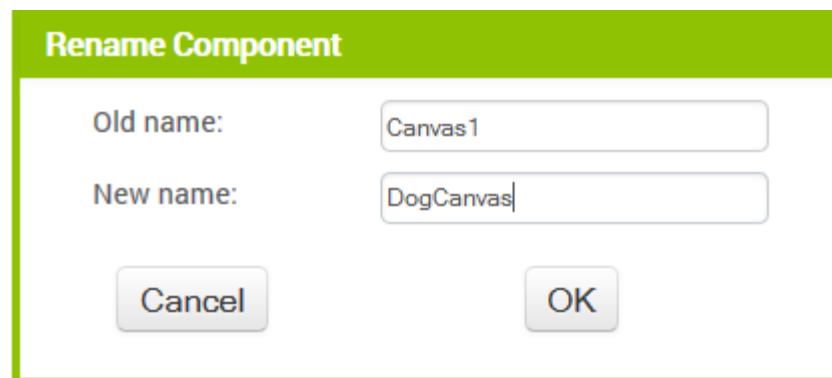
Ο καμβάς (Canvas) είναι μια ορθογώνια επιφάνεια, εντός της οποίας μπορούμε να χειριζόμαστε φιγούρες (Sprites) ή να σχεδιάζουμε αγγίζοντάς την.

Προκειμένου να ορίσουμε μια περιοχή στην οποία θα είναι δυνατή η μετακίνηση της φιγούρας του σκύλου, θα προσθέσουμε στην οθόνη μας έναν καμβά.

Σέρνουμε λοιπόν από την παλέτα αριστερά (Palette), από την ομάδα πλακιδίων Drawing and Animation, το συστατικό Canvas, και το αποθέτουμε στην οθόνη.



Είναι σημαντικό (αλλά όχι υποχρεωτικό), να ονομάζουμε τα συστατικά που χρησιμοποιούμε με τέτοιο τρόπο, ώστε να τα αναγνωρίζουμε ευκολότερα. Γι' αυτό το λόγο, μετονομάζουμε τον καμβά Canvas1 σε DogCanvas.



Παρατήρηση: Τα ονόματα των συστατικών πρέπει να αποτελούνται από λατινικούς χαρακτήρες, αριθμούς ή κάτω παύλες και πρέπει να ξεκινάνε με χαρακτήρα, οπότε δεν μπορούμε να χρησιμοποιήσουμε π.χ. ελληνικούς χαρακτήρες ή κενά.

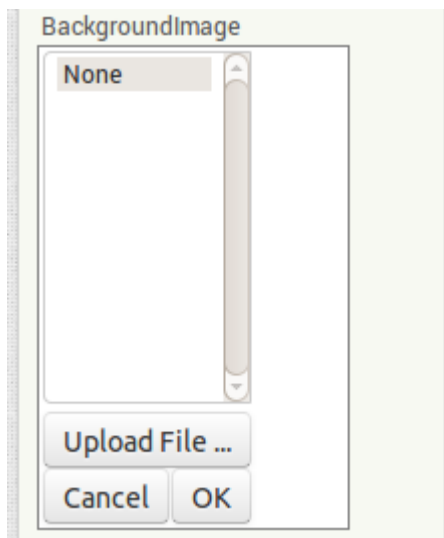
Τέλος, μεταβάλλουμε τις ιδιότητες του καμβά DogCanvas ως εξής:



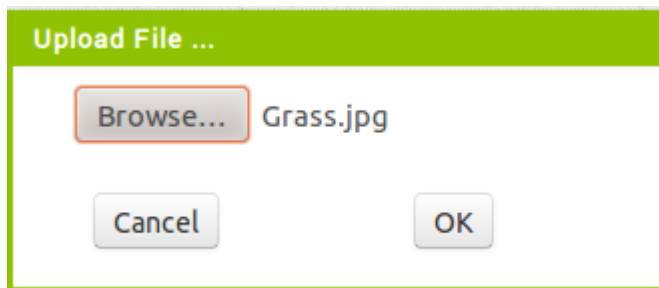
από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Drawing and Animation	Canvas	DogCanvas	Width: Fill Parent Height: Fill Parent BackgroundImage: Grass.jpg

Η επιλογή Fill Parent για τις ιδιότητες Width και Height του καμβά του επιτρέπει να επεκταθεί και να καταλάβει όλο το διαθέσιμο χώρο. Έτσι, ο σκύλος θα μπορεί να κινείται κατά μήκος και κατά πλάτος όλης της οθόνης.

Για να χρησιμοποιηθεί ως φόντο του καμβά DogCanvas η εικόνα με το γρασίδι (Grass.jpg), θα πρέπει πρώτα αυτή να «ανέβει» στο project μας. Κάνουμε κλικ στην ιδιότητα BackgroundImage, επιλέγουμε Upload File...



...μετά Αναζήτηση... (Browse...) κι επιλέγουμε το αρχείο Grass.jpg.



Παρόμοιες κινήσεις κάνουμε για κάθε αρχείο εικόνας ή ήχου που σκοπεύουμε να χρησιμοποιήσουμε στο project, ενώ στο Βήμα 5 περιγράφεται κι ένας εναλλακτικός τρόπος να ανεβάζουμε αρχεία στο project μας.

Βήμα 3 Προσθήκη του σκύλου

Στο βήμα αυτό θα προσθέσουμε τη φιγούρα του σκύλου-φύλακα.

Designer

Οι φιγούρες τοποθετούνται εντός του καμβά και μπορούν να αλληλεπιδρούν με τον καμβά, με άλλες φιγούρες εντός του καμβά και φυσικά να αντιδρούν στις δικές μας ενέργειες.

Προκειμένου να προσθέσουμε τον σκύλο, σέρνουμε μια φιγούρα (ImageSprite) μέσα στον καμβά. Μέσω της ιδιότητας Picture της φιγούρας, ορίζουμε σαν εικόνα της την ήρεμη εκδοχή του σκύλου, κατά παρόμοιο τρόπο με το Βήμα 2.



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Drawing and Animation	ImageSprite	DogSprite	Interval: 10 Picture: SleepyDog.jpg Rotates: no

Αν ο σκύλος φαίνεται πολύ μεγάλος, μπορούμε να αλλάξουμε τις διαστάσεις του σε pixel, μέσω των ιδιοτήτων Width και Height.

Βήμα 4 Γάβγισμα!

Στο βήμα αυτό θα προσθέσουμε τον ήχο του γαβγίσματος, καθώς και τις πρώτες μας εντολές, ώστε ο σκύλος να γαβγίζει όταν αγγίζουμε την οθόνη.

Designer

Όταν η εφαρμογή μας χρησιμοποιεί ήχους τότε χρειάζεται να προσθέσουμε τους ήχους αυτούς, καθώς και τα συστατικά ήχου (Sound) που αναλαμβάνουν την αναπαραγωγή τους.

Για να μπορεί ο σκύλος να γαβγίζει, προσθέτουμε ένα συστατικό Sound της ομάδας Media, σέρνοντάς το στην οθόνη. Αυτό θα αναπαράγει τον ήχο, όταν το θελήσουμε. Παρατηρούμε ότι δεν εμφανίζεται εντός της οθόνης, αλλά κάτω από αυτήν, αποτελώντας για τον χρήστη της εφαρμογής ένα μη-ορατό συστατικό (non-visible component). Το μετονομάζουμε και ορίζουμε ως πηγή του ήχου το αρχείο Bark.mp3, μέσω της ιδιότητας Source:



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Media	Sound	DogBarkingSound	Source: Bark.mp3 MinimumInterval: 300

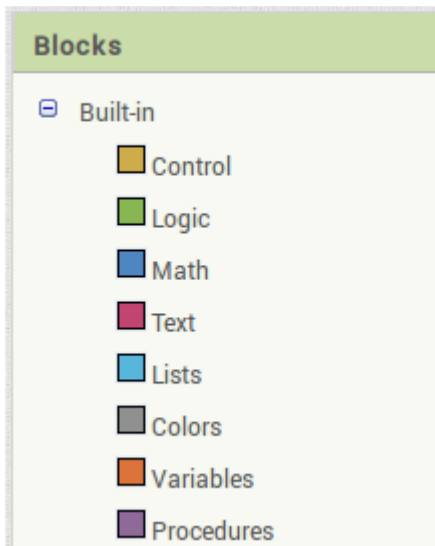
Η ιδιότητα `MinimumInterval` είναι ο ελάχιστος χρόνος πριν την επανάληψη του ήχου. Αν λοιπόν θέσουμε `MinimumInterval: 300 msec (0,3 sec)` τότε ο ήχος δεν θα μπορεί να ξαναπαίξει προτού περάσουν τουλάχιστον 0,3 δευτερόλεπτα.

Παρατήρηση: Το συστατικό `Sound`, ακόμα κι αν δεν ορίσουμε συγκεκριμένο αρχείο ήχου για αναπαραγωγή, είναι απαραίτητο, αν θέλουμε η συσκευή να δονείται σαν αποτέλεσμα κάποιας ενέργειας.

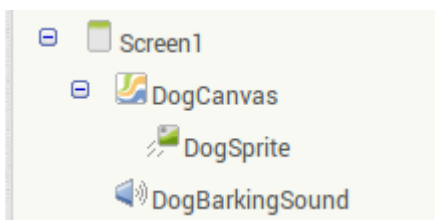
Blocks

Μέχρι στιγμής, στο Designer έχουμε σχεδιάσει ένα μέρος της «οθόνης» της εφαρμογής μας, αλλά δεν έχουμε ορίσει καθόλου την συμπεριφορά της. Μεταβαίνουμε λοιπόν στο `Blocks` (το κουμπί μετάβασης βρίσκεται πάνω δεξιά) για να συσχετίσουμε ενέργειες με γεγονότα και, ουσιαστικά, να προγραμματίσουμε, προσθέτοντας τις κατάλληλες εντολές.

Στα αριστερά της οθόνης διακρίνουμε τρεις ομάδες πλακιδίων (`blocks`). Πρώτα βρίσκουμε τα ενσωματωμένα (`built-in`) πλακίδια, την λειτουργία των οποίων θα εξετάσουμε αργότερα.



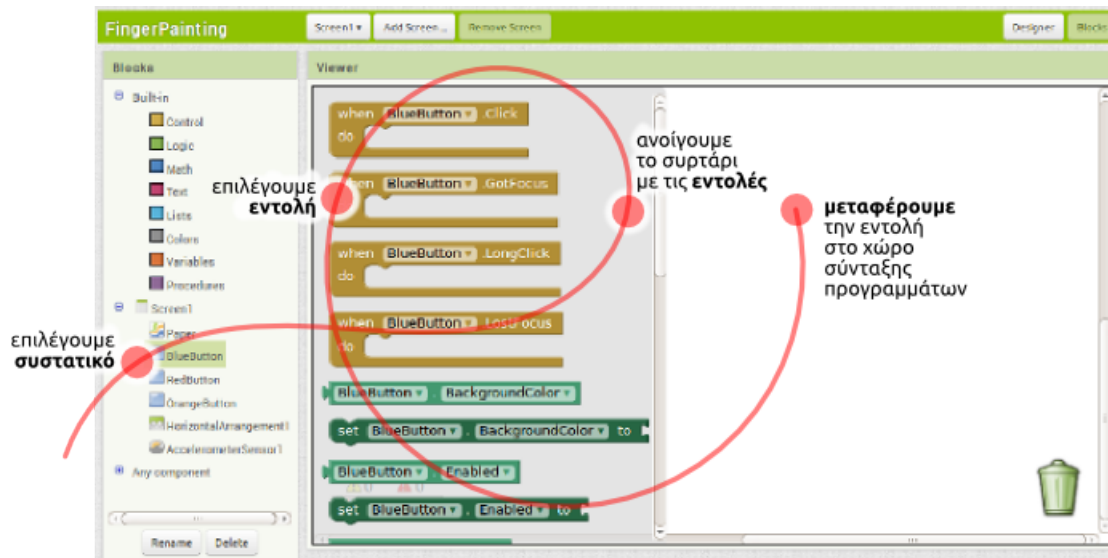
Στη συνέχεια, βλέπουμε τα πλακίδια που αφορούν τα συστατικά που έχουμε προσθέσει στην εφαρμογή μας.



Ακολουθεί η τελευταία ομάδα, που δε θα μας απασχολήσει εδώ.

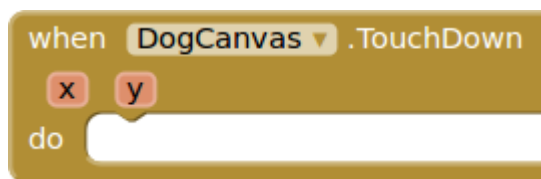
Όταν προγραμματίζουμε μια συγκεκριμένη συμπεριφορά για την εφαρμογή μας πρακτικά συναρμολογούμε πλακίδια εντολών (και όχι μόνο). Για κάθε πλακίδιο που θέλουμε να προσθέσουμε **ανατρέχουμε στην κατάλληλη ομάδα στα αριστερά της οθόνης**, ανοίγει το αντίστοιχο «συρτάρι» με τις διαθέσιμες

εντολές, αναζητούμε και επιλέγουμε το πλακίδιο που χρειαζόμαστε και το σέρνουμε στον χώρο σύνταξης των προγραμμάτων.

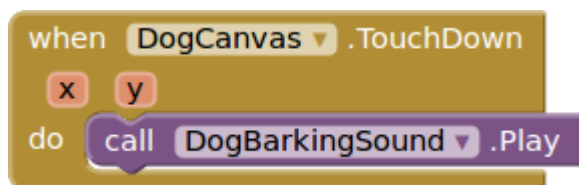


Στην συγκεκριμένη περίπτωση, θέλουμε να ακούγεται ο ήχος του γαβγίσματος DogBarkingSound κάθε φορά που ο χρήστης αγγίζει τον καμβά DogCanvas.

Επιλέγουμε λοιπόν το συστατικό DogCanvas, ανοίγει το αντίστοιχο συρτάρι και μεταφέρουμε το πλακίδιο when DogCanvas.TouchDown.



Στη συνέχεια επιλέγουμε το συστατικό DogBarkingSound, εντοπίζουμε το πλακίδιο call DogBarkingSound.Play και το «κουμπώνουμε» μέσα στο προηγούμενο πλακίδιο.



Ας διαβάσουμε τα πλακίδια που συναρμολογήσαμε. Δίνουν τις εντολές που είναι απαραίτητες έτσι ώστε η εφαρμογή μας να κάνει αυτό που θέλουμε.



Όταν ο χρήστης αγγίζει τον καμβά DogCanvas, τότε αναπαράγεται ο ήχος DogBarkingSound.

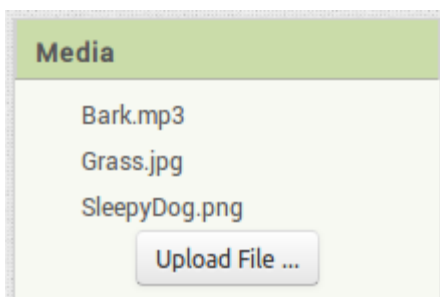
Παρατήρηση: Σημειώστε ότι το πλακίδιο when DogCanvas.TouchDown επιστρέφει και δύο νούμερα, τα x και y. Αυτά τα νούμερα προσδιορίζουν που ακριβώς ακούμπησε το δάχτυλό του ο χρήστης στην οθόνη και θα τα χρησιμοποιήσουμε στο Βήμα 6.

Βήμα 5 Αλλαγή της μορφής του σκύλου

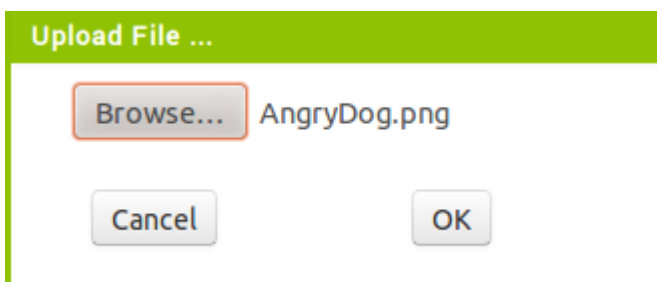
Στο βήμα αυτό θα επεκτείνουμε την συμπεριφορά του σκύλου έτσι ώστε να αλλάζει μορφή και να αγριεύει όταν τον αγγίζουμε.

Designer

Προσθέτουμε στο project τη θυμωμένη εκδοχή του σκύλου (AngryDog.png). Στο το πλαίσιο Media πατάμε Upload File...



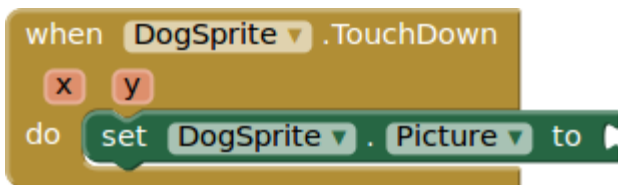
στη συνέχεια Browse... και επιλέγουμε το αρχείο AngryDog.png.



Blocks

Εδώ θέλουμε η φιγούρα του σκύλου DogSprite να αλλάζει μορφή όταν ο χρήστης την αγγίζει και να επανέρχεται όταν η επαφή σταματά. Όλα τα πλακίδια αφορούν την μορφή DogSprite και θα τα βρούμε στο συρτάρι που εμφανίζεται όταν την επιλέξουμε.

Αρχικά, εντοπίζουμε και συναρμολογούμε τα παρακάτω πλακίδια:



Παρατηρούμε ότι οι εντολές μας είναι ελλιπείς. Πώς θα προσδιορίσουμε την εικόνα που θα έχει η φιγούρα DogSprite όταν αγγιχθεί; Θα πρέπει να παρέχουμε το αντίστοιχο όνομα αρχείου. Ανατρέχουμε λοιπόν στην ενσωματωμένη ομάδα πλακιδίων Text και επιλέγουμε το πλακίδιο του κενού κειμένου...



...στο οποίο συμπληρώνουμε το όνομα του αρχείου εικόνας "AngryDog.png" και το προσθέτουμε στις εντολές μας.

```
when DogSprite .TouchDown
  x y
do set DogSprite . Picture to " AngryDog.png "
```

Αναλόγως προσδιορίζουμε την εικόνα στην οποία θα επανέρχεται ο σκύλος όταν σταματάει η επαφή.

```
when DogSprite .TouchUp
  x y
do set DogSprite . Picture to " SleepyDog.png "
```



Όταν ο χρήστης αγγίζει τη φιγούρα DogSprite, τότε αυτή παίρνει την μορφή AngryDog.png. Όταν σταματήσει η επαφή με την φιγούρα DogSprite τότε αυτή παίρνει την μορφή SleepyDog.png.

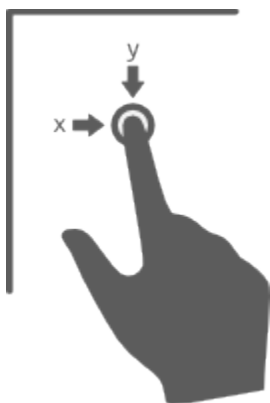
Βήμα 6 Προσοχή στα δάχτυλα

Στο βήμα αυτό θα κάνουμε τον φύλακά μας λίγο πιο επιθετικό. Όταν ο χρήστης αγγίζει την οθόνη τότε ο σκύλος θα κινείται προς το σημείο επαφής.

Blocks

Εδώ θέλουμε όταν ο χρήστης αγγίζει τον καμβά DogCanvas η φιγούρα DogSprite να στρέφεται και να κινείται προς το σημείο επαφής. Στο Βήμα 4 είχαμε προσδιορίσει πως όταν ο χρήστης ακουμπά τον καμβά DogCanvas τότε ακούγεται ο ήχος του γαβγίσματος.

```
when DogCanvas .TouchDown
  x y
do call DogBarkingSound .Play
```



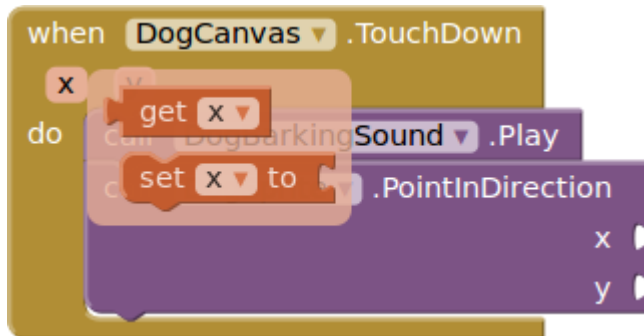
Διαθέτουμε λοιπόν ήδη το when DogCanvas.TouchDown, το οποίο προσδιορίζει και το σημείο στο οποίο έγινε η επαφή, δηλαδή την οριζόντια απόσταση x από το αριστερό άκρο του καμβά και την κάθετη απόσταση y από το πάνω άκρο του καμβά. Οι αριθμοί x και y ονομάζονται και *συντεταγμένες* του σημείου επαφής.

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

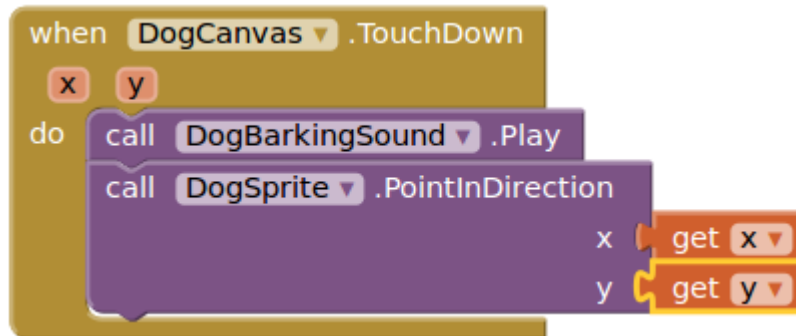
Αρχικά, προσθέτουμε το πλακίδιο call DogSprite.PointInDirection, το οποίο στρέφει τη φιγούρα προς ένα συγκεκριμένο σημείο.



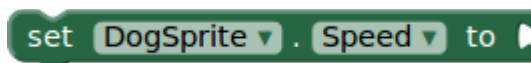
Για να πάρουμε την τιμή της x στην οποία έγινε η επαφή αφήνουμε το δείκτη του ποντικιού πάνω από την ετικέτα x κι εμφανίζεται το πλακίδιο get x...



...το οποίο σέρνουμε και κουμπώνουμε στο x της PointInDirection. Το ίδιο κάνουμε και για την y.



Τώρα απομένει να ασχοληθούμε με την κίνηση του σκύλου. Θα πρέπει, όταν ο χρήστης ακουμπά τον καμβά, να τροποποιήσουμε την ταχύτητά του με το πλακίδιο set DogSprite.Speed.



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



...τον οποίο τροποποιούμε σε 40 και κουμπώνουμε στην set DogSprite.Speed.

```
set DogSprite . Speed to 40
```

Εισάγουμε πλέον ολόκληρη την εντολή στην when DogCanvas.TouchDown.

```
when DogCanvas . TouchDown
  x y
do
  call DogBarkingSound . Play
  call DogSprite . PointInDirection
  x get x
  y get y
  set DogSprite . Speed to 40
```



Όταν ο χρήστης αγγίζει τον καμβά DogCanvas, τότε αναπαράγεται ο ήχος DogBarkingSound, η φιγούρα DogSprite στρέφεται προς το σημείο επαφής (με συντεταγμένες x και y και κινείται με ταχύτητα 40.

Με παρόμοιο τρόπο, μηδενίζουμε την ταχύτητα του σκύλου όταν το δάχτυλο του χρήστη δεν είναι πια σε επαφή με την οθόνη.

```
when DogCanvas . TouchUp
  x y
do
  set DogSprite . Speed to 0
```



Όταν ο χρήστης παύει να αγγίζει τον καμβά DogCanvas η φιγούρα DogSprite ακινητοποιείται.



Αυτό το φύλλο εργασίας βασίζεται στο παράδειγμα HelloPurr και διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού 3.0 (<http://goo.gl/oyehU8>). Η αναφορά σε αυτό θα πρέπει να γίνεται ως εξής:

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

Τα εικονίδια που χρησιμοποιήθηκαν στο φύλλο εργασίας προέρχονται από το σετ εικονιδίων Faenza, του Matthieu James.