



Σχεδιασμός και αξιολόγηση εκπαιδευτικού λογισμικού

Εργαστηριακή Ενότητα 3 : App Inventor /
Πρόγραμμα Ζωγραφικής

Διδάσκων: Νικόλαος Τσέλιος

Τμήμα Επιστημών της Εκπαίδευσης και της Αγωγής
στην Προσχολική Ηλικία

Σκοποί ενότητας

- Να σχεδιάσει μια εφαρμογή ζωγραφικής με το App Inventor
- Να τη δοκιμάσει στη συσκευή του
- Να εξοικειωθεί με το αντικείμενο canvas για
ζωγραφική/σχεδίαση
- Να μάθει να διαχειρίζεται γεγονότα σχετικά με την επαφή
- Να χρησιμοποιεί μεταβλητές για να αποθηκεύσει κάποιες τιμές

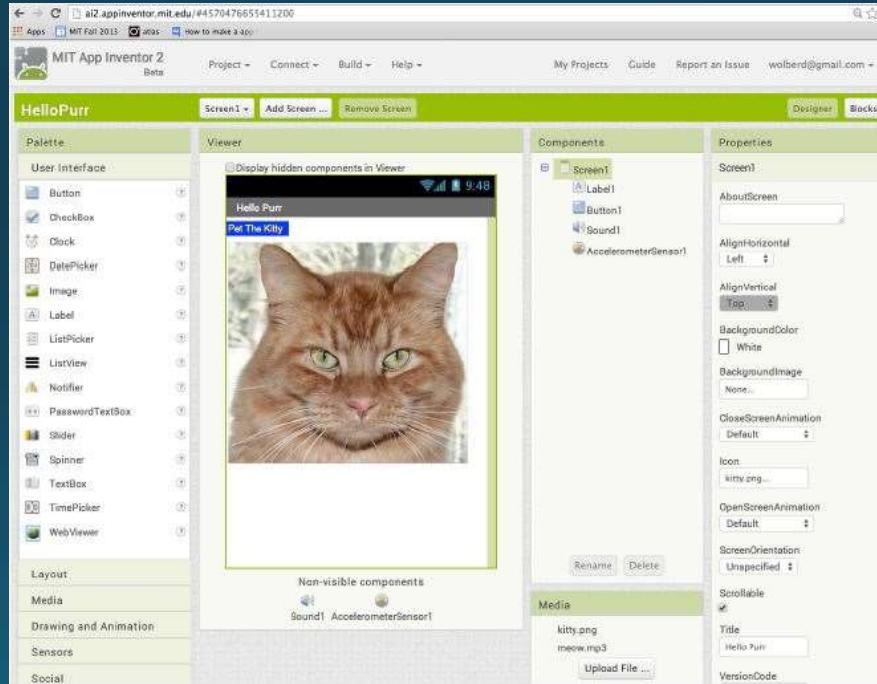
Περιεχόμενα ενότητας

- Περιγραφή: Δημιουργούμε, δοκιμάζουμε και τροποποιούμε μια ολοκληρωμένη εφαρμογή με τη χρήση του App Inventor
- Λέξεις Κλειδιά: App Inventor, Mobile devices, designer, block, γεγονότα, canvas

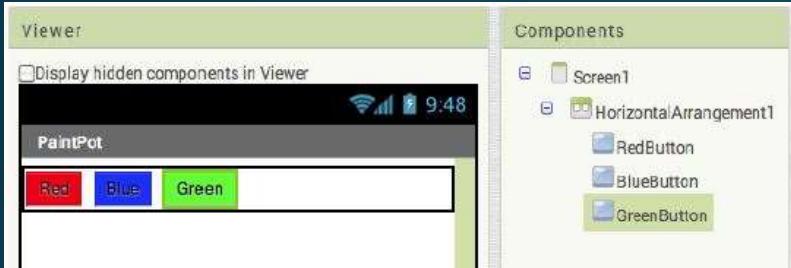
Η εφαρμογή PaintPot

- Θα σχεδιάσουμε ένα πρόγραμμα ζωγραφικής
- Θα μπορούμε να ζωγραφίζουμε πάνω σε μια φωτογραφία
- Θα μπορούμε να αλλάζουμε χρώματα, να σχεδιάζουμε γραμμές και να καθαρίζουμε την οθόνη
- Θα μπορούμε να αλλάζουμε το πάχος της γραμμής
- Θα μπορούμε να παίρνουμε φωτογραφία με τη κάμερα και να ζωγραφίζουμε πάνω σε αυτή

Η εφαρμογή PaintPot (designer)

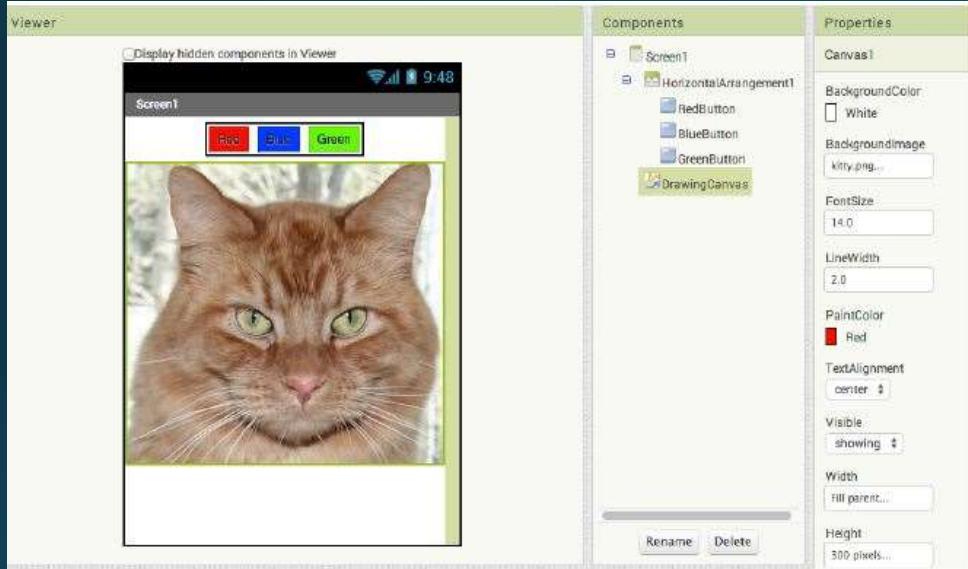


Βήμα 1



- Επιλέγουμε νέο έργο (project)
- Επιλέγουμε designer
- Προσθέτουμε σταδιακά 3 buttons
- Παρατηρήστε ότι δεν μπορούμε να βάλουμε κενά στα ονόματα των αντικειμένων
- Ακολούθως βάζουμε ένα αντικείμενο HorizontalArrangement για να 'τακτοποιήσουμε' τα 3 buttons
- Βάζουμε FillParent στις ιδιότητες του HorizontalArrangement (έτσι θα καταλαμβάνει το πλήρες πλάτος της οθόνης)

Βήμα 2



- Σύρουμε και αφήνουμε από την παλέτα
- Το αντικείμενο Canvas
- BackgroundImage (που θα εμφανίζει τη γάτα)
(<http://appinventor.org/bookFiles/HelloPurr/kitty.png>)
- Θα πρέπει να αποδηκεύσουμε στον υπολογιστή την εικόνα και να τη μεταφορτωσουμε
- Πλάτος βάζουμε πάλι Fillparent

Βήμα 3



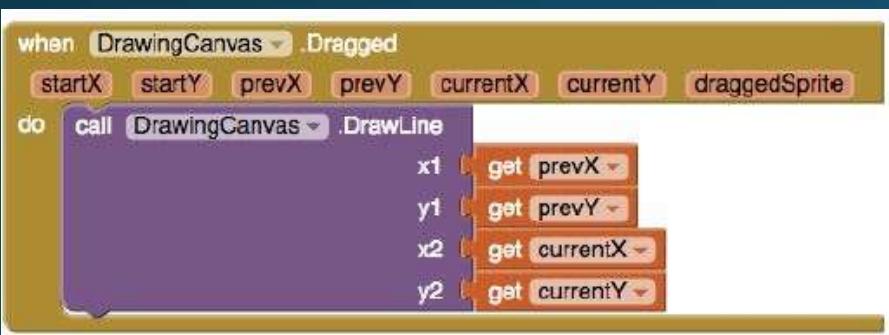
- Προσθέτουμε τα πλήκτρα
- Takepicture, WipeButton,
BigButton, SmallButton
- Επίσης βάζουμε το αντικείμενο
camera
- *Βάζουμε αντικείμενο*
HorizontalArrangement για να
τακτοποιήσουμε τα αντικείμενα
- Πλάτος βάζουμε πάλι Fillparent

Βήμα 4



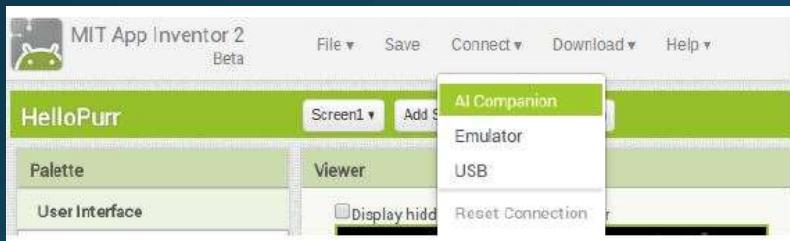
- Πηγαίνουμε στο block
- Επιλέγουμε το Drawingcanvas.touched
- Επιλέγουμε την ενέργεια DrawCircle

Βήμα 5



- Για τα σημεία x και y θα πρέπει να σημειώσουμε να δηλώσουμε τις συντεταγμένες που αναγνωρίζει η εφαρμογή
- Θα μπορούσαμε να σχεδιάσουμε οπουδήποτε αλλά μας ενδιαφέρει να αποτυπώσουμε μια κουκίδα ακριβώς εκεί που πάτησε ο χρήστης
- Βάζουμε τα αντικείμενα $getx$ και $gety$ (εμφανίζονται είτε στη λίστα είτε αν σύρουμε και αφήσουμε το ποντίκι πάνω στο γεγονός `DrawingCanvas.touched`)
- Στην εσοχή r θα πρέπει να ορίσουμε το πάχος της κουκίδας. Βάζουμε την τιμή 5
- Αντίστοιχα, προσθέτουμε εντολές για το αντικείμενο `drawingcanvas.dragged`
- Πρακτικά, δίνουμε την εντολή να σχεδιαστεί μια γραμμή που ενώνει τα σημεία (x_1, y_1) και (x_2, y_2) ($prev$ και $current$ αντίστοιχα)

Δοκιμή



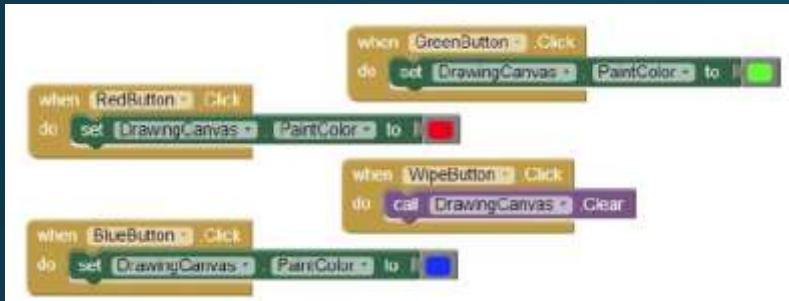
- Μπορούμε να δοκιμάσουμε την εφαρμογή στο κινητό μας!
- Καλό είναι να το κάνουμε κάθε φορά που προσθέτουμε / τροποποιούμε κάτι στην εφαρμογή μας
- Πρέπει να έχουμε εγκαταστήσει το "MIT AI2 Companion" από το Google Play store
- Κινητό και υπολογιστής πρέπει να είναι στο ίδιο ασύρματο δίκτυο
- Επιλέγουμε Connect -> AI companion

Δοκιμή (2)



- Ανοίγουμε στο κινητό μας την εφαρμογή “MIT AI2 Companion”
- Μπορούμε να βάλουμε τον 6ψήφιο κωδικό που εμφανίζεται ή
- Να σκανάρουμε το QR code (βάζουμε το κινητό μπροστά στην οθόνη αφού επιλέξουμε τη σχετική επιλογή)
- Γενικές οδηγίες στο <http://appinventor.mit.edu/explore/ai2/setup.html>

Βήμα 6



- Προσθέτουμε γεγονότα και εντολές για να διαχειριστούμε την επιλογή button για την αλλαγή χρώματος και καθαρισμού
- Προσθέτουμε γεγονότα για τη λήψη φωτογραφίας

Βήμα 7

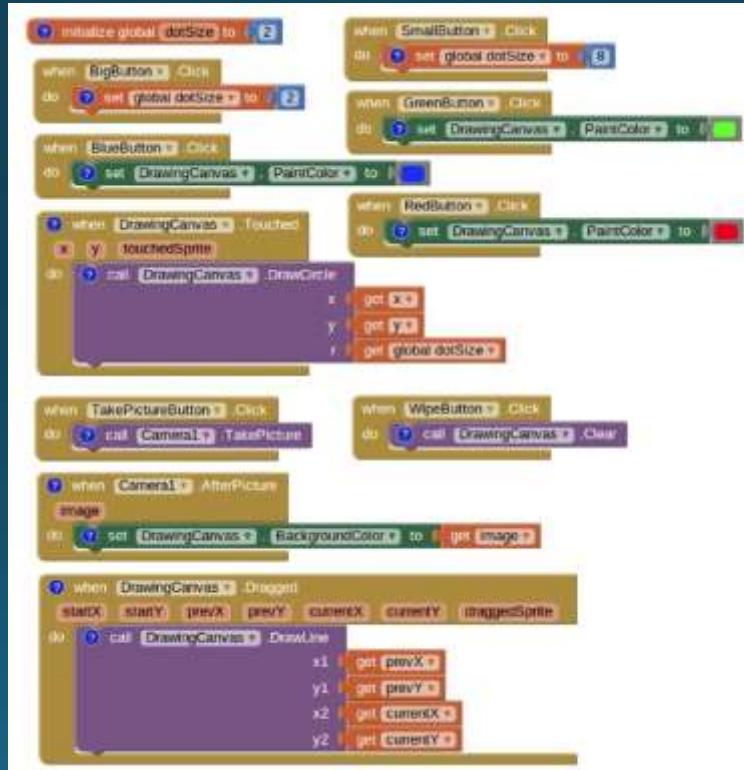


- Πως θα διαφοροποιούμε το πάχος της γραμμής;
- Θα πρέπει να έχουμε 2 διαφορετικές τιμές για το πάχος
- Αυτές θα πρέπει να τις καταχωρίσουμε σε μια μεταβλητή
- Για να το κάνουμε αυτό πηγαίνουμε Blocks->variables και εισάγουμε τη δήλωση `initialize global`
- Το 'name' το αλλάζουμε σε 'dotSize' (περιγράφει το τι αποθηκεύει η μεταβλητή με ακρίβεια) και θέτουμε αρχική τιμή το 2 (μικρό πάχος)
- Οπότε πλέον στο πάχος της γραμμής δεν θα έχουμε 5 αλλά την τιμή της `dotSize`, οπότε κάνουμε την αλλαγή στο σχετικό γεγονός
- Αντίστοιχα προσαρμόζουμε την τιμή `dotSize` σε 2 και 8 κάθε φορά που επιλέγει ο χρήστης το πλήκτρο μικρού- μεγάλου πάχους

Τελική μεταφόρτωση εφαρμογής

- Επιλέγουμε εικονίδιο για την εφαρμογή (Designer ->Screen, φορτώνουμε μια εικόνα)
- Θα πρέπει να επιτρέψουμε στη συσκευή μας την εγκατάσταση εφαρμογών εκτός Google Play
 - Settings -> Applications επιλέγουμε το 'Unknown sources'
- Επιλέγουμε Build ->APK
- Πιθανά θα χρειαστούμε ένα QR code scanner (δωρεάν διαθέσιμοι στο Google Play Store)
- Μπορούμε να αποθηκεύσουμε τοπικά το αρχείο .apk και να το στείλουμε και σε άλλους χρήστες ή να το ανεβάσουμε στο δίκτυο (πχ dropbox)

Η τελική εφαρμογή Paintpot (block)



Τι μάθαμε;

- Φτιάχνουμε μια εφαρμογή ζωγραφικής
- Χρησιμοποιήσαμε την κάμερα
- Αποθηκεύουμε σε μεταβλητή το πάχος της κουκίδας
- Διαφοροποιούμε το χρώμα

Πιθανές επεκτάσεις;

- Προσθήκη ήχου κάθε φορά που ζωγραφίζουμε ή όταν παίρνουμε μια φωτογραφία
- Δεν υπάρχει ανάδραση για το ποιο είναι το πάχος της γραμμής.
Πως θα μπορούσαμε να το κάνουμε αυτό;
- Προσθήκη αντικειμένου slider για να μεταβάλλουμε το πάχος της γραμμής

Βιβλιογραφία

- Wolber, D., Abelson, H., Spertus, E., & Looney, L. (2014). *App Inventor 2: Create Your Own Android Apps*. O'Reilly Media, Inc.

Τέλος Ενότητας