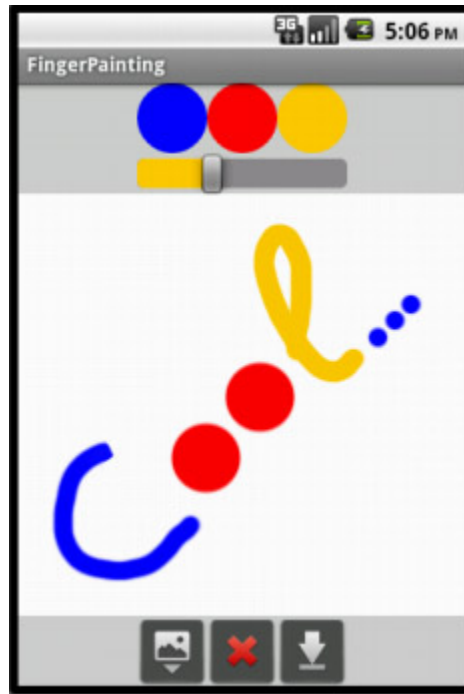




## FingerPainting Φύλλο Εργασίας

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

Ακολουθώντας τα βήματα αυτού του φύλλου εργασίας, θα αναπτύξουμε σταδιακά μια απλή **εφαρμογή ζωγραφικής με τα δάχτυλα**.



Αν και η εφαρμογή που θα κατασκευάσουμε είναι αρκετά απλή, δεν πρόκειται για ένα εισαγωγικό παράδειγμα. Οι οδηγίες που ακολουθούν υποθέτουν μια στοιχειώδη εξοικείωση με το περιβάλλον του AppInventor.

### Βήμα 1 Δημιουργία νέου project

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με τίτλο FingerPainting.

Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη (Screen) και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της.



επιλέγουμε το  
**συστατικό**

Screen1

μεταβάλλουμε τις  
**ιδιότητες**

BackgroundColor: LightGrey  
Screen Orientation: Portrait  
Scrollable: No  
Title: FingerPainting

## Βήμα 2 Σχέδια επάνω στον καμβά

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

### Designer

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



από την  
**ομάδα**

μεταφέρουμε το  
**συστατικό**

του δίνουμε το  
**όνομα**

μεταβάλλουμε τις  
**ιδιότητες**

Drawing and  
Animation

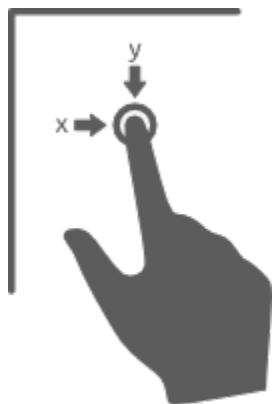
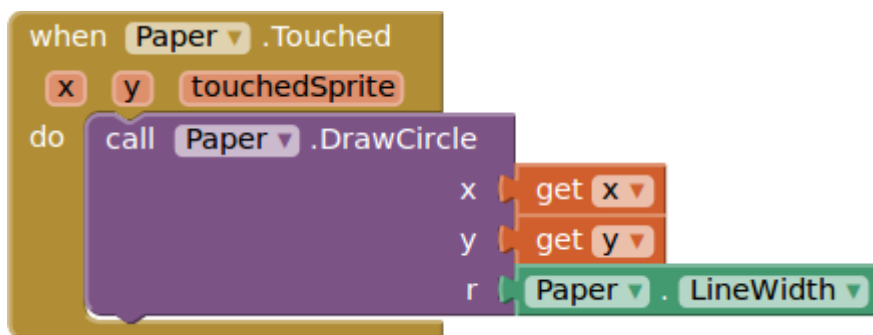
Canvas

Paper

Paint Color: Blue  
Width: Fill Parent  
Height: Fill Parent

### Blocks

Οι δύο ομάδες εντολών που ακολουθούν ανιχνεύουν πότε ο χρήστης ακούμπησε την οθόνη της συσκευής ή έσυρε το δάχτυλο πάνω σε αυτήν. Στην πρώτη περίπτωση σχεδιάζεται μια τελεία, ενώ στη δεύτερη μια γραμμή.



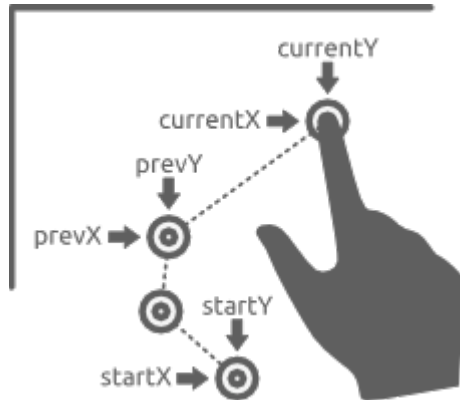
Όταν ο χρήστης αγγίζει ένα σημείο του καμβά Paper (με συντεταγμένες  $x$  και  $y$ ) τότε σε εκείνο το σημείο ζωγραφίσε έναν κύκλο. Η ακτίνα του κύκλου προέρχεται από την ιδιότητα `LineWidth` του καμβά Paper.

Το χρώμα δεν καθορίζεται από τις συγκεκριμένες εντολές, αλλά από την ιδιότητα `PaintColor` του καμβά.

```

when Paper ▾ .Dragged
  startX startY prevX prevY currentX currentY
do
  call Paper ▾ .DrawLine
    x1 get prevX ▾
    y1 get prevY ▾
    x2 get currentX ▾
    y2 get currentY ▾

```



Όταν ο χρήστης σύρει το δάχτυλό του στον καμβά Paper τότε ζωγράφισε μια γραμμή ανάμεσα στο τρέχον σημείο επαφής (με συντεταγμένες currentX και currentY) και στο προηγούμενο σημείο απ' όπου πέρασε το δάχτυλο του χρήστη (με συντεταγμένες prevX και prevY).

Το πάχος και το χρώμα της γραμμής δεν καθορίζονται από τις συγκεκριμένες εντολές, αλλά από τις ιδιότητες LineWidth και PaintColor του καμβά.

### Βήμα 3 Καθαρισμός του καμβά

Στο δεύτερο βήμα θα προσθέσουμε τη δυνατότητα καθαρισμού του σχεδίου μας με ένα κούνημα της συσκευής.

#### Designer

Κάθε συσκευή διαθέτει αισθητήρα επιτάχυνσης ο οποίος μπορεί να ανιχνεύσει, μεταξύ άλλων, πότε ο χρήστης ταρακουνά την συσκευή. Προσθέτουμε ένα συστατικό που θα μας επιτρέψει να χρησιμοποιήσουμε αυτόν τον αισθητήρα (AccelerometerSensor) στην εφαρμογή.



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα
Sensors	Accelerometer Sensor	AccSensor

## Blocks

```
when AccSensor .Shaking  
do call Paper .Clear
```



Όταν ο αισθητήρας AccSensor ανιχνεύσει ταρακούνημα της συσκευής τότε καθάρισε τον καμβά Paper.

## Βήμα 4 Μια μικρή παλέτα με χρώματα

Στο βήμα αυτό θα προσθέσουμε στην εφαρμογή τα κουμπιά που θα μας επιτρέψουν να επιλέγουμε και να χρησιμοποιούμε διαφορετικά χρώματα.

## Designer

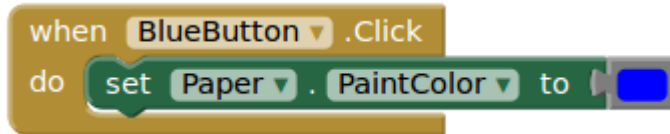
Μεταφέρουμε μια οριζόντια *διευθέτηση* (HorizontalArrangement) πάνω ή κάτω από τον καμβά και τοποθετούμε μέσα σε αυτήν τρία κουμπιά (Button), έτσι ώστε να εμφανίζονται το ένα δίπλα στο άλλο.



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	ButtonArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Button	BlueButton	BackgroundColor: Blue Shape: Oval Text: Width: 50 pixels Height: 50 pixels
User Interface	Button	RedButton	BackgroundColor: Red Shape: Oval Text: Width: 50 pixels Height: 50 pixels
User Interface	Button	OrangeButton	BackgroundColor: Orange Shape: Oval Text: Width: 50 pixels Height: 50 pixels

## Blocks

Η παρακάτω ομάδα εντολών συσχετίζει το άγγιγμα ενός εκ των κουμπιών με το χρώμα ζωγραφικής. Θα πρέπει να επαναληφθεί μία φορά για κάθε κουμπί χρώματος (δηλαδή και για το RedButton και το OrangeButton).



Όταν ο χρήστης αγγίζει το κουμπί BlueButton τότε το χρώμα με το οποίο ζωγραφίζουμε στον καμβά Paper γίνεται μπλε (μέσω της ιδιότητας PaintColor).

## Βήμα 5 Αλλάζοντας το πάχος της γραμμής

Στο βήμα αυτό θα προσθέσουμε τη δυνατότητα μεταβολής του πάχους που θα έχουν οι γραμμές που σχεδιάζουμε.

## Designer

Προσθέτουμε μια οριζόντια διεύθυνση (HorizontalArrangement) κάτω από τα κουμπιά επιλογής χρώματος και τοποθετούμε μέσα της έναν ολισθητή (Slider), δηλαδή έναν δείκτη που κινείται αριστερά και δεξιά. Ο ολισθητής θα χρησιμοποιηθεί για την μεταβολή του πάχους των γραμμών.



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	SliderArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Slider	WidthSlider	MaxValue: 30 MinValue: 2 ThumbPosition: 2 Width: 150 pixels

Οι ιδιότητες MinValue και MaxValue που χρησιμοποιούνται για τον ολισθητή αντιστοιχούν στο ελάχιστο και στο μέγιστο πάχος γραμμής. Η ιδιότητα ThumbPosition είναι η τρέχουσα τιμή του ολισθητή (άρα και το τρέχον πάχος).

## Blocks

```
when WidthSlider .PositionChanged  
  thumbPosition  
do set Paper . LineWidth to get thumbPosition
```



Όταν αλλάξει η θέση thumbPosition του ολισθητή Width τότε ενημέρωσε ανάλογα και το πάχος των γραμμών του καμβά Paper (μέσω της ιδιότητας LineWidth).

Όταν το πάχος της γραμμής γίνει μεγάλο τότε οι γραμμές δεν σχεδιάζονται σωστά. Μπορούμε να αντιμετωπίσουμε το πρόβλημα σχεδιάζοντας έναν κύκλο στο τρέχον σημείο επαφής με την οθόνη. Η ακτίνα του κύκλου θα πρέπει να είναι το μισό του πάχους γραμμής. Προσθέστε λοιπόν την ακόλουθη εντολή στο whenPaperDragged του Βήματος 1.

```
call Paper .DrawCircle  
  x get currentX  
  y get currentY  
  r Paper . LineWidth / 2
```

## Βήμα 6 Σχέδια επάνω σε φωτογραφίες

Σε αυτό το βήμα θα μπορέσουμε να ζωγραφίσουμε επάνω σε φωτογραφίες που τραβάμε με την συσκευή μας.

## Designer

Χρειαζόμαστε ένα κουμπί (Button) με το οποίο θα ενεργοποιούμε την κάμερα της συσκευής, καθώς και ένα συστατικό που αντιστοιχεί στην ίδια την κάμερα (Camera). Τοποθετούμε το κουμπί μέσα σε μια νέα οριζόντια διεύθετηση.



από την ομάδα	μεταφέρουμε το συστατικό	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	PhotoArrangement	Width: Fill Parent AlignHorizontal: Center



από την  
**ομάδα**

μεταφέρουμε  
το **συστατικό**

του δίνουμε το  
**όνομα**

μεταβάλλουμε τις  
**ιδιότητες**

User  
Interface

Button

ClickButton

Width: 50  
Height: 40  
Text:  
Image: photograph.png

Media

Camera

Camera1

## Blocks

```
when ClickButton .Click  
do call Camera1 .TakePicture
```



Όταν ο χρήστης πατήσει το κουμπί ClickButton τότε ενεργοποίησε την φωτογραφική μηχανή Camera1 της συσκευής.

```
when Camera1 .AfterPicture  
  image  
do set Paper . BackgroundImage to get image
```



Όταν τραβηχτεί μια φωτογραφία image από την φωτογραφική μηχανή Camera1 τότε θέσε την φωτογραφία αυτή ως φόντο στον καμβά Paper.

## Βήμα 7 προαιρετικό

### Καθαρισμός του καμβά (συνέχεια)

Ο καθαρισμός του καμβά στο Βήμα 3 αφήνει ανέπαφη την εικόνα που πιθανώς να έχουμε θέσει ως φόντο. Μπορούμε να εισάγουμε ένα νέο κουμπί με όνομα ClearButton, να το τοποθετήσουμε δίπλα στο SaveButton και να προσθέσουμε τις ακόλουθες εντολές, έτσι ώστε το πάτημα του κουμπιού να καθαρίζει το φόντο (και το σχέδιό μας).

```
when ClearButton .Click
do
  set Paper . BackgroundImage to false
```

## Βήμα 8 προαιρετικό

### Αποθήκευση των σχεδίων μας

Για να έχουμε τη δυνατότητα να αποθηκεύσουμε τα σχέδιά μας προσθέτουμε ένα ακόμα κουμπί στην εφαρμογή μας, με όνομα SaveButton. Το πάτημα του κουμπιού αποθηκεύει το σχέδιό μας σε ένα αρχείο με όνομα fp, ακολουθούμενο από έναν μοναδικό αριθμό. Για να χρησιμοποιήσουμε τα παρακάτω πλακίδια εισάγουμε επίσης στην εφαρμογή ένα συστατικό Clock (με όνομα Clock1) και έναν Notifier (με όνομα SaveNotifier).

```
when SaveButton .Click
do
  initialize local name to call Paper .SaveAs
  join "fp-"
  call Clock1 .SystemTime
  join ".png"
  in call SaveNotifier .ShowMessageDialog
  message join "filename:"
  get name
  title "Saved Image"
  buttonText "OK"
```



Αυτό το φύλλο εργασίας βασίζεται στο παράδειγμα PaintPot (<http://goo.gl/ny3PEj>) 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: *FingerPainting – Προγραμματισμός Εφαρμογής με το AppInventor*. Γιώργος Μπουκέας, Γιώργος Χατζηνικολάκης. Σύλλογος Εκπαιδευτικών Πληροφορικής Χίου, 2013.

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