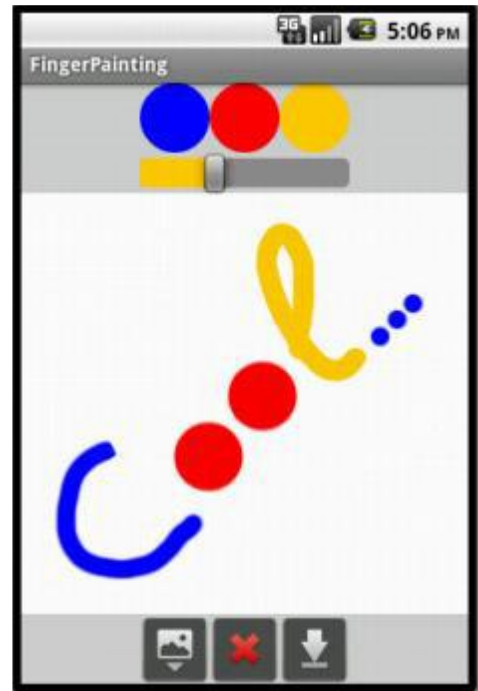


App Inventor – 8ο Μάθημα (Ζωγραφική με τα δάχτυλα)

- ✓ Υποπρόγραμμα – διαδικασία (procedure)
- ✓ Λήψη φωτογραφιών
- ✓ Σχεδίαση κύκλων και γραμμών σε καμβά
- ✓ Αποθήκευση σχεδίου στη συσκευή.

Θα κατασκευάσουμε μια εφαρμογή ζωγραφικής με τα δάχτυλα η οποία θα επιτρέπει να ζωγραφίζουμε ακόμα και πάνω σε φωτογραφίες που τραβάμε με την κάμερα της συσκευής, ενώ θα μπορούμε και να αποθηκεύσουμε τις δημιουργίες μας. Το περιβάλλον της εφαρμογής μας θα μοιάζει με το διπλανό στιγμιότυπο.

Βήμα 1: Δημιουργία νέου project και ρυθμίσεις οθόνης




1. Στο <http://ai2.appinventor.mit.edu/> ξεκινάμε δημιουργώντας ένα νέο Project με το όνομα *FingerPainting*, **Start new project**.
2. Αρχικά, το μοναδικό διαθέσιμο συστατικό της εφαρμογής είναι η οθόνη, Screen, και θα χρειαστεί να μεταβάλλουμε κάποιες από τις ιδιότητές της. Επιλέγουμε λοιπόν το αντικείμενο *Screen1* και θέτουμε τις επόμενες ιδιότητες:

- **AlignHorizontal:** *Center*
- **BackgroundColor:** *Light Gray*
- **Screen Orientation:** *Portrait*
- **Scrollable:** *No*
- **Title:** *FingerPainting*

Βήμα 2: Προσθήκη αρχείων πολυμέσων

3. Στη συνέχεια θα προσθέσουμε στην εφαρμογή μας τα αρχεία που θα χρειαστούμε. Επιλέγουμε, όπως πάντα, **Media – Upload File** και βρίσκουμε στον ίδιο φάκελο που βρίσκονται και οι παρούσες σημειώσεις τα επόμενα αρχεία εικόνας.

	ανεβάζουμε το αρχείο με όνομα	σύνομη περιγραφή
	clear.jpg	Εικόνα για το κουμπί καθαρισμού του καμβά
	takepic.png	Εικόνα για το κουμπί λήψης φωτογραφίας
	save.png	Εικόνα για το κουμπί αποθήκευσης εικόνας

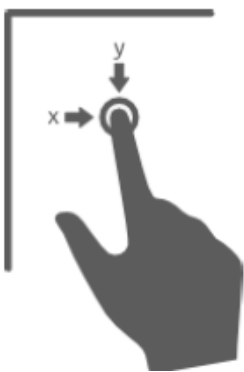
Βήμα 3: Σχέδια σε καμβά

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.

4. Στο **Designer** προσθέτουμε τον καμβά, **Canvas**, της ομάδας **Drawing and Animation**, του δίνουμε το όνομα *Paper* και μεταβάλλουμε τις επόμενες ιδιότητες.

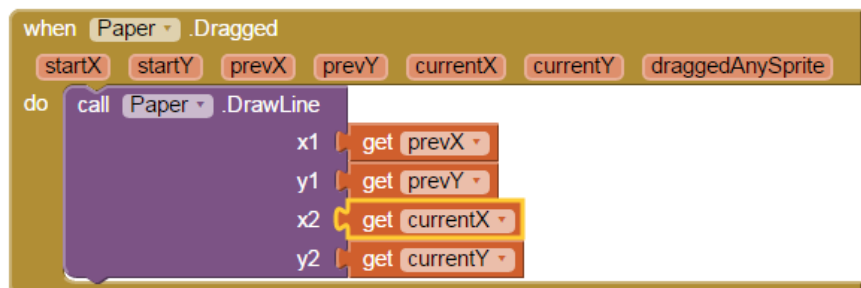
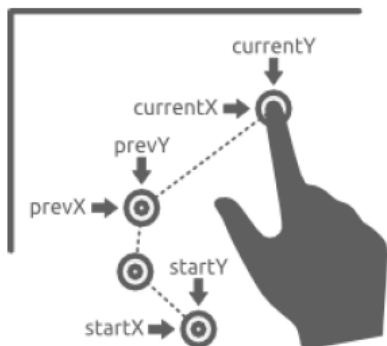
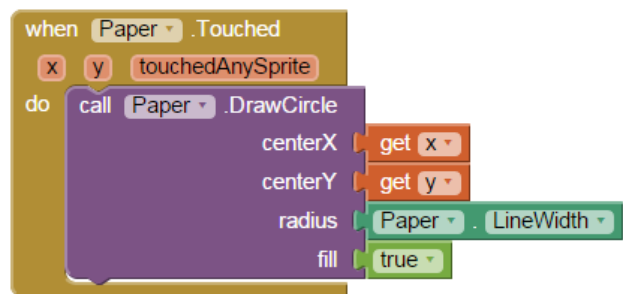
- **Paint Color:** *Blue*
- **Width:** *Fill Parent*
- **Height:** *Fill Parent*

5. Μεταβαίνουμε στο **Blocks** και εισάγουμε δύο ομάδες εντολών, που 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 ζωγράφισε έ-ναν κύκλο. Η ακτίνα του κύκλου προέρχεται από την ιδιότητα *LineWidth* του καμβά *Paper*.

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*.

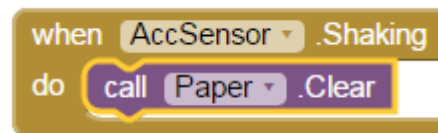
Το πάχος και το χρώμα της γραμμής δεν καθορίζονται από τις συγκεκριμένες εντολές, αλλά από τις ιδιότητες *LineWidth* και *PaintColor* του καμβά.

Όπως σε κάθε βήμα μπορείτε να δοκιμάσετε την εργασία σας μέχρι τη στιγμή αυτή στη συσκευή σας, **Connect – AI Companion**.

Βήμα 4: Καθαρισμός του καμβά

6. Κάθε συσκευή διαθέτει αισθητήρα επιτάχυνσης ο οποίος μπορεί να ανιχνεύσει, μεταξύ άλλων, πότε ο χρήστης ταρακουνά την συσκευή. Μεταβαίνουμε στο **Designer** και προσθέτουμε ένα αντικείμενο που θα μας επιτρέψει να χρησιμοποιήσουμε αυτόν τον αισθητήρα, **AccelerometerSensor** της ομάδας **Sensors**, στην εφαρμογή. Προσοχή, ο αισθητήρας αυτός είναι μη ορατός και ως εκ τούτου θα τοποθετηθεί στην αντίστοιχη ομάδα αντικειμένων κάτω από την οθόνη του κινητού, **Non-visible components**. Δώστε το όνομα *AccSensor* στον αισθητήρα σας.

7. Στο **Blocks** συνδυάζουμε το γεγονός της ανίχνευσης ταρακουνημάτων της συσκευής με την ενέργεια καθαρισμού της οθόνης.



Βήμα 5: Μία μικρή παλέτα με χρώματα

8. Στο **Designer** μεταφέρουμε μια οριζόντια τακτοποίηση, **HorizontalArrangement** της ομάδας **Layout**, πάνω ή κάτω από τον καμβά και τοποθετούμε μέσα σε αυτήν τρία κουμπιά, **Button** της ομάδας **User Interface**, έτσι ώστε να εμφανίζονται το ένα δίπλα στο άλλο. Θέτουμε τις επόμενες ιδιότητες στα αντικείμενά μας

HorizontalArrangement

- **Όνομα:** *ColorArrangement*
- **Width:** *Fill Parent*
- **AlignHorizontal:** *Center*

Button

- **Όνομα:** *BlueButton*
- **BackgroundColor:** *Blue*
- **Shape:** *Oval*
- **Text:**
- **Width:** *50 pixels*
- **Height:** *50 pixels*

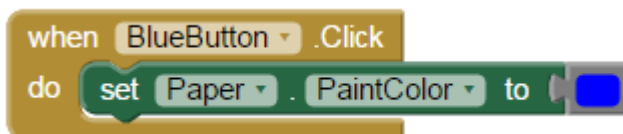
Button

- **Όνομα:** *RedButton*
- **BackgroundColor:** *Red*
- **Shape:** *Oval*
- **Text:**
- **Width:** *50 pixels*
- **Height:** *50 pixels*

Button

- **Όνομα:** *OrangeButton*
- **BackgroundColor:** *Orange*
- **Shape:** *Oval*
- **Text:**
- **Width:** *50 pixels*
- **Height:** *50 pixels*

9. Στο **Blocks** με το διπλανό σενάριο 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*).



10. Δοκιμάστε την προσθήκη της επιλογής χρώματος στη συσκευή σας.

Βήμα 6: Αλλάζοντας το πάχος της γραμμής

11. Στο **Designer** προσθέτουμε μια οριζόντια τακτοποίηση, **HorizontalArrangement** της ομάδας **Layout**, κάτω από τα κουμπιά επιλογής χρώματος και τοποθετούμε μέσα της έναν *μεταβολέα*. **Slider** της ομάδας **UserInterface**, δηλαδή έναν δείκτη που κινείται αριστερά και δεξιά. Ο κυλιόμενος αυτός

μεταβολέας θα χρησιμοποιηθεί για την αλλαγή του πάχους των γραμμών. Θέτουμε τις ιδιότητες που ακολουθούν στα νέα μας αντικείμενα:

HorizontalArrangement

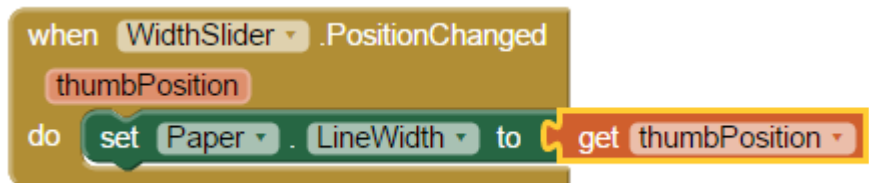
- Όνομα: *SliderArrangement*
- Width: *Fill Parent*
- AlignHorizontal: *Center*

Slider

- Όνομα: *WidthSlider*
- MaxValue: *30*
- MinValue: *2*
- ThumbPosition: *2*
- Width: *150 pixels*

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

12. Στο **Blocks** συσχετίζουμε το πάχος γραμμής του καμβά με τη θέση του μεταβολέα. Όταν αλλάξει η θέση **thumbPosition** του μεταβολέα *WidthSlider* τότε ενημέρωσε ανάλογα και το πάχος των γραμμών του καμβά *Paper* μέσω της ιδιότητας **LineWidth**.



Βήμα 7: Σχέδια πάνω σε φωτογραφίες

13. Σε αυτό το βήμα θα μπορέσουμε να ζωγραφίσουμε επάνω σε φωτογραφίες που τραβάμε με την συσκευή μας. Πίσω και πάλι στο **Designer** χρειαζόμαστε ένα κουμπί, **Button** της ομάδας **User Interface** με το οποίο θα ενεργοποιούμε την κάμερα της συσκευής, καθώς και ένα αντικείμενο-συστατικό που αντιστοιχεί στην ίδια την κάμερα, **Camera** της ομάδας **Media**. Τοποθετούμε το κουμπί μέσα σε μια νέα οριζόντια τακτοποίηση, **Horizontal Arrangement** της ομάδας **Layout**.

HorizontalArrangement

- Όνομα: *ButtonArrangement*
- Width: *Fill Parent*
- AlignHorizontal: *Center*

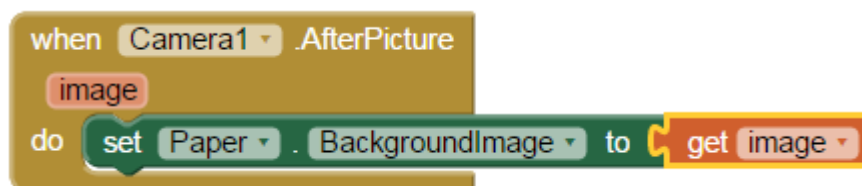
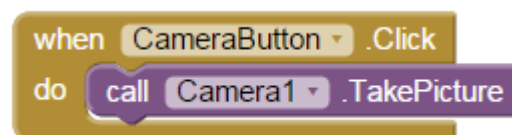
Button

- Όνομα: *CameraButton*
- Width: *50*
- Height: *40*
- Text:
- Image: *takepic.png*

Media

- Όνομα: *Camera1* (το εξορισμού όνομα)

14. Στο **Blocks**, καλούμε την ενσωματωμένη στο λειτουργία που επιτρέπει τη λήψη φωτογραφίας, με το πάτημα του κουμπιού *CameraButton*, αλλά και μετά το



τράβηγμα της φωτογραφίας θα θέσουμε σαν φόντο στον καμβά *Paper* τη φωτογραφία που μόλις τραβήξαμε.

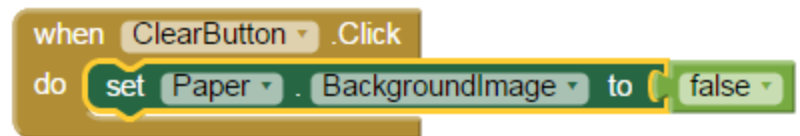
15. Δοκιμάστε με τον γνωστό τρόπο, Connect – AI Companion, τη λειτουργία της φωτογραφίας και ζωγραφίστε ένα μουστάκι στον διπλανό σας.

Βήμα 8: Καθαρισμός του καμβά

16. Με τον προηγούμενο καθαρισμό, αφήνουμε την εικόνα ανέπαφη. Για την περίπτωση που θέλουμε να σβήσουμε την εικόνα, προσθέτουμε ένα ακόμη **Button** στο *ButtonArrangement*.

Button

- Όνομα: *ClearButton*
- Width: 50
- Height: 40
- Text:
- Image: *clear.png*



17. Στο **Blocks** προσθέτουμε το διπλανό σενάριο και αν θέλουμε δοκιμάζουμε και πάλι της εφαρμογή μας.

Βήμα 9: Αποθήκευση των σχεδίων μας

18. Για να έχουμε τη δυνατότητα να αποθηκεύσουμε τα σχέδιά μας, προσθέτουμε καταρχάς ένα ακόμα **Button** στο *ButtonArrangement*. Προκειμένου να ερωτάται ο χρήστης για το όνομα του αρχείου προς αποθήκευση, θα χρησιμοποιήσουμε και ένα αντικείμενο **Notifier** της ομάδας **User Interface**.

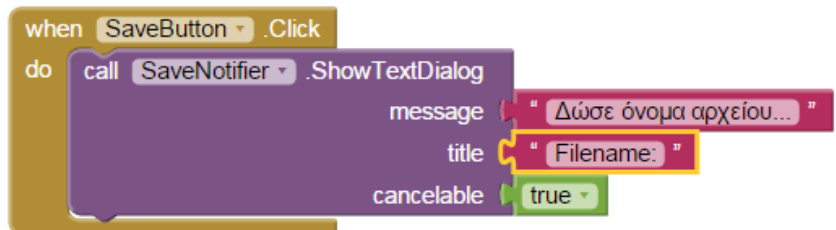
Button

- Όνομα: *SaveButton*
- Width: 50
- Height: 40
- Text:
- Image: *save.png*

Notifier

- Όνομα: *SaveNotifier*

19. Σε πρώτη φάση θα εμφανίζουμε ένα παράθυρο διαλόγου που θα προτρέπει το χρήστη να δώσει όνομα στο αρχείο προς αποθήκευση. Στην συνέχεια, αφού δοθεί το όνομα, το σχέδιο αποθηκεύεται στη συσκευή και εμφανίζεται ένα ενημερωτικό μήνυμα για μερικά δευτερόλεπτα με το όνομα του αρχείου και τη θέση αποθήκευσης.



20. Αμέσως μετά, θα αξιοποιήσουμε την απάντηση που δόθηκε για να αποθηκεύσουμε το σχέδιο με το επόμενο σενάριο που ουσιαστικά υλοποιεί την πρόταση που ακολουθεί. Αφού ολοκληρώθηκε η εισαγωγή του ονόματος, *response*, από το χρήστη, κάλεσε την έτοιμη διαδικασία αποθήκευσης που

διαθέτει ο καμβάς Paper, αποθηκεύοντας τη ζωγραφιά με όνομα *response.png* και εμφανίσε σχετικό ενημερωτικό μήνυμα.

21. Η εφαρμογή σας είναι έτοιμη. Μπορείτε να τη δοκιμάσετε στη συσκευή σας!

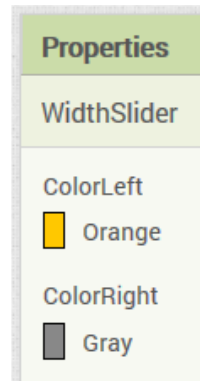
```

when SaveNotifier .AfterTextInput
  response
  do
    call SaveNotifier .ShowAlert
      notice
    call Paper .SaveAs
      fileName
    join
    get response
      ".png"
  
```

Δραστηριότητες

1. Προς το παρόν, δεν υπάρχει κάποια εμφανής ένδειξη για το ποιο είναι το ενεργό χρώμα σχεδίασης στον καμβά. Μια εύκολη λύση είναι, όποτε πατάμε κάποιο κουμπί χρώματος, να αλλάζει το χρώμα του μεταβολέα (slider) πάχους γραμμής, ώστε να υποδηλώνει το ενεργό χρώμα.

Αν μεταβείτε στις ιδιότητες του αντικειμένου WidthSlider θα διαπιστώσετε ότι προς το παρόν χρησιμοποιείται πορτοκαλί χρώμα αριστερά του δείκτη και γκρι χρώμα δεξιά του.



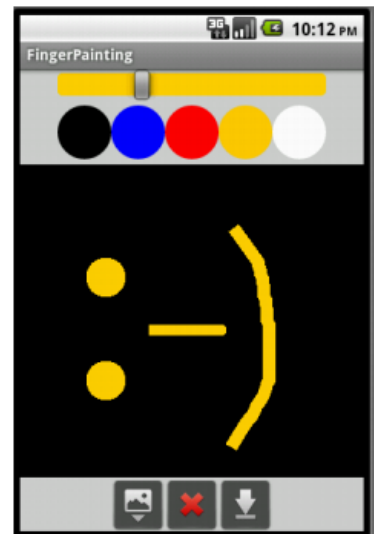
Προσθέστε τις κατάλληλες εντολές ώστε σε κάθε αλλαγή χρώματος, να χρωματίζονται και οι δυο πλευρές του μεταβολέα με το επιλεγμένο χρώμα.

2. Αν θέλετε να προσθέσετε, στην παλέτα χρωμάτων που φτιάξατε στο 5^ο βήμα, περισσότερα διαθέσιμα χρώματα, μαζί με τα ανάλογα κουμπιά, θα διαπιστώσετε ότι αυξάνονται και οι αντίστοιχες εντολές, οι οποίες ουσιαστικά επαναλαμβάνονται. Το μόνο στοιχείο που αλλάζει είναι το χρώμα, ιδιότητα *PaintColor*, του καμβά *Paper*.

Υπενθυμίζεται, από το Βήμα 5, ότι προκειμένου να ορίσουμε σαν χρώμα σχεδίασης στον καμβά π.χ. το μπλε, χρησιμοποιήσαμε το σετ εντολών ...

```

when BlueButton .Click
  do
    set Paper .PaintColor to
  
```



... ενώ στο προηγούμενο ερώτημα (1.α) έχετε προσθέσει κάποιες επιπλέον εντολές που σχετίζονται με το μπλε χρώμα.

Σας ζητείται να δημιουργήσετε μια διαδικασία *setColor [x]* η οποία θα καλείται όποτε πατάμε σε κάποιο κουμπί χρώματος και θα αναλαμβάνει να κάνει όλες τις απαραίτητες αλλαγές ιδιοτήτων.

Τροποποιήστε λοιπόν τα γεγονότα που συμβαίνουν με το πάτημα του μπλε, κόκκινου και πορτοκαλί κουμπιού ώστε να χρησιμοποιούν τη διαδικασία *setColor* και προσθέστε μερικά επιπλέον χρώματα της αρεσκείας σας, μαζί με τα αντίστοιχα κουμπιά στην παλέτα χρωμάτων.

3. Κάποιος συμμαθητής σας δεν χρησιμοποίησε μεταβολέα για τον ορισμό του πάχους γραμμής στον καμβά, αλλά μια μεταβλητή με όνομα *brushSize* και δυο κουμπιά που αυξομειώνουν το μέγεθος. Παρακάτω βλέπετε ένα μέρος του κώδικα που δημιούργησε:

```

initialize global brushSize to 0

when BiggerBrushButton .Click
  do
    set Paper .LineWidth to get global brushSize
    set global brushSize to get global brushSize + 1
  
```

Όταν, όμως επιχειρεί να σχεδιάσει κάτι στην οθόνη, δεν εμφανίζεται τίποτα, παρά μόνο όταν πατήσει το κουμπί BiggerBrushButton για δεύτερη φορά. Που βρίσκεται το πρόβλημα και τι αλλαγές προτείνετε για τη διόρθωσή του;

Πηγή: <http://www.sepchiou.gr/>