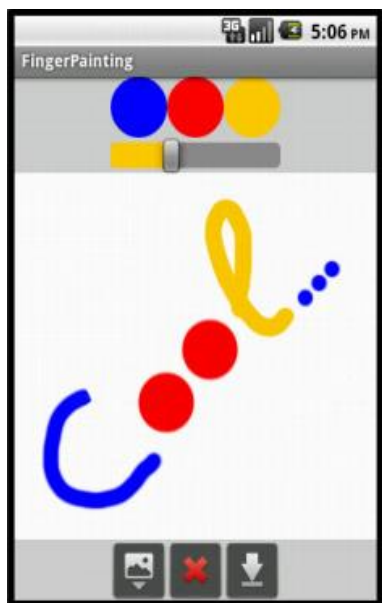


ΔΡΑΣΤΗΡΙΟΤΗΤΑ : ΖΩΓΡΑΦΙΚΗ ΜΕ ΤΑ ΔΑΚΤΥΛΑ

ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

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 παρακάτω.



Εικόνα 110 – Στιγμιότυπο της εφαρμογής Finger Painting

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

- Υποπρόγραμμα – διαδικασία (procedure).
- Λήψη φωτογραφιών με την κάμερα της συσκευής.
- Σχεδίαση κύκλων και γραμμών σε καμβά.
- Αποθήκευση σχεδίου στη συσκευή.

Βήμα 1 Δημιουργία νέου project και ρυθμίσεις οθόνης

Ξεκινώντας, δημιουργούμε ένα καινούργιο project με όνομα FingerPainting.

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



επιλέγουμε το
αντικείμενο

μεταβάλλουμε τις
ιδιότητες

Screen1

AlignHorizontal: Center

BackgroundColor: LightGrey

Screen Orientation: Portrait

Scrollable: No

Title: FingerPainting

Βήμα 2**Προσθήκη των απαραίτητων αρχείων πολυμέσων**

Ας προσθέσουμε στο project μας τα αρχεία εικόνας που θα χρησιμοποιεί η εφαρμογή μας.



ανεβάζουμε το αρχείο με
όνομα

σύντομη
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 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**Γραφικό Περιβάλλον Εφαρμογής**

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



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

μεταβάλλουμε τις
ιδιότητες

Drawing and Animation

Canvas

Paper

Paint Color: Blue

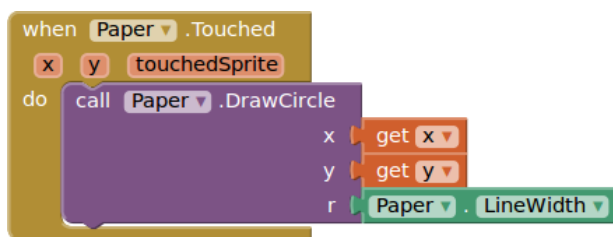
Width: Fill Parent

Height: Fill Parent

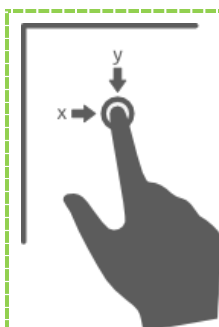
Συγγραφή Προγράμματος

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.

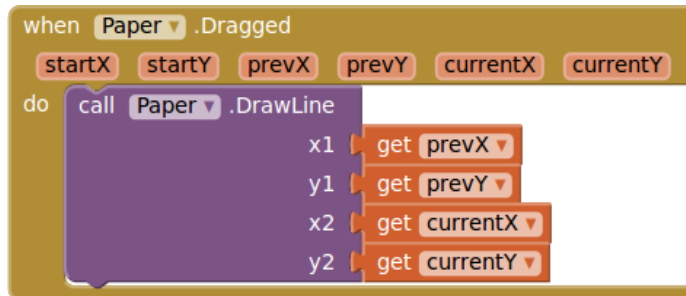


Εικόνα 111 - Σχεδιασμός κύκλου στο σημείο επαφής

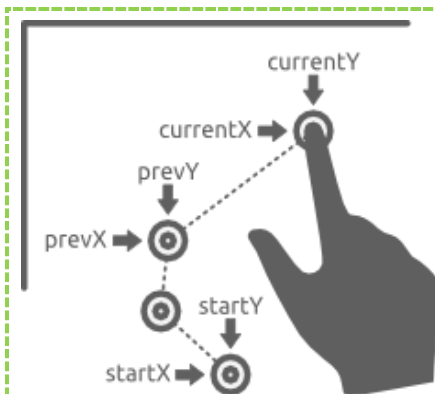


Όταν ο χρήστης αγγίζει ένα σημείο του καμβά Paper (με συντεταγμένες x και y) τότε σε εκείνο το σημείο ζωγράφισε έναν κύκλο. Η ακτίνα του κύκλου προέρχεται από την ιδιότητα LineWidth του καμβά Paper.

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



Εικόνα 112 - Σχεδιασμός γραμμής με σύρσιμο



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 4 Καθαρισμός του καμβά

Σε αυτό το βήμα θα προσθέσουμε τη δυνατότητα καθαρισμού του σχεδίου μας με ένα κούνημα της συσκευής.

Designer Γραφικό Περιβάλλον Εφαρμογής

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



από την
ομάδα

Sensors

μεταφέρουμε το
αντικείμενο

Accelerometer Sensor

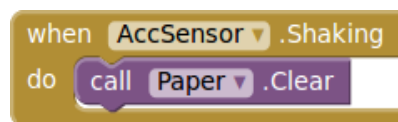
του δίνουμε το
όνομα

AccSensor

Συγγραφή Προγράμματος

Blocks

Συνδυάζουμε το γεγονός της ανίχνευσης ταρακουνήματος της συσκευής με την ενέργεια καθαρισμού της οθόνης.



Εικόνα 113 - Καθαρισμός καμβά με ταρακούνημα



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

Βήμα 5 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 Γραφικό Περιβάλλον Εφαρμογής

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

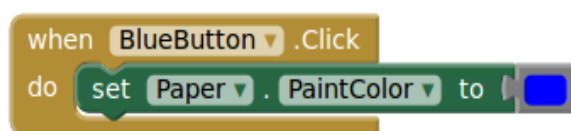
	από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
	Layout	Horizontal Arrangement	ColorArrangement	Width: Fill Parent AlignHorizontal: Center
	User Interface	Button	BlueButton	BackgroundColor: Blue Shape: Oval Text: Width: 50 pixels Height: 50 pixels
	User Interface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

Συγγραφή Προγράμματος

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).



Εικόνα 114 - Ορισμός του χρώματος βαφής



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

Βήμα 6 Αλλάζοντας το πάχος της γραμμής

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

Designer Γραφικό Περιβάλλον Εφαρμογής

Προσθέτουμε μια οριζόντια τακτοποίηση (HorizontalArrangement) κάτω από τα κουμπιά επιλογής χρώματος και τοποθετούμε μέσα της έναν μεταβολέα (Slider), δηλαδή έναν δείκτη που κινείται αριστερά και δεξιά. Ο κυλιόμενος αυτός μεταβολέας θα χρησιμοποιηθεί για την αλλαγή του πάχους των γραμμών.



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

μεταβάλλουμε τις
ιδιότητες

Layout

Horizontal
Arrangement

SliderArrangement

Width: Fill Parent
AlignHorizontal: Center

User Interface

Slider

WidthSlider

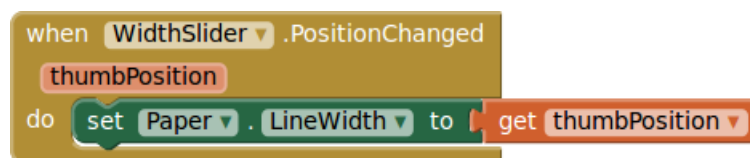
MaxValue: 30
MinValue: 2
ThumbPosition: 2
Width: 150 pixels

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

Συγγραφή Προγράμματος

Blocks

Συσχετίζουμε το πάχος γραμμής του καμβά με τη θέση του μεταβολέα.

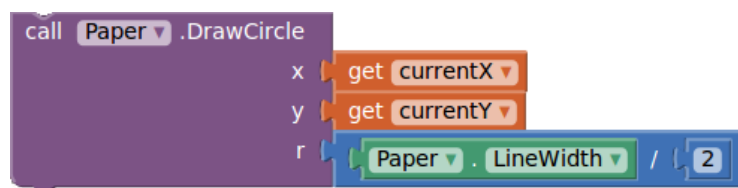


Εικόνα 115 - Ορισμός πάχους γραμμής



Όταν αλλάξει η θέση *thumbPosition* του μεταβολέα *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. Προσθέτουμε λοιπόν την ακόλουθη εντολή στο whenPaperDragged του Βήματος 3.



Εικόνα 116 - Βελτίωση για γραμμές μεγάλου πάχους

Βήμα 7 Σχέδια πάνω σε φωτογραφίες

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

Designer Γραφικό Περιβάλλον Εφαρμογής

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

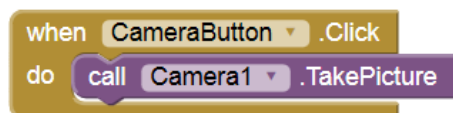


από την ομάδα	μεταφέρουμε το αντικείμενο	του δίνουμε το όνομα	μεταβάλλουμε τις ιδιότητες
Layout	Horizontal Arrangement	ButtonArrangement	Width: Fill Parent AlignHorizontal: Center
User Interface	Button	CameraButton	Width: 50 Height: 40 Text: Image: takepic.png
Media	Camera	Camera1	

Συγγραφή Προγράμματος

Blocks

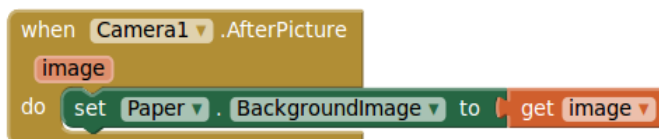
Καλούμε την ενσωματωμένη στο App Inventor λειτουργία που επιτρέπει τη λήψη φωτογραφίας, με το πάτημα του κουμπιού CameraButton.



Εικόνα 117 - Λήψη φωτογραφίας



Όταν ο χρήστης πατήσει το κουμπί CameraButton τότε ενεργοποιήσει την φωτογραφική μηχανή Camera1 της συσκευής.



Εικόνα 118 - Ορισμός της φωτογραφίας ως φόντου του καμβά

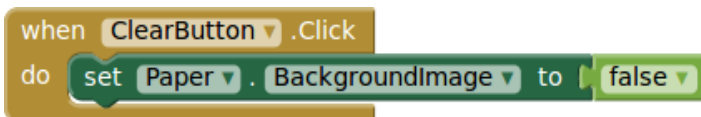


Όταν τραβηχτεί μια φωτογραφία image από την φωτογραφική μηχανή Camera1 τότε θέσει την φωτογραφία αυτή ως φόντο στον καμβά Paper.

Βήμα 8

Καθαρισμός του καμβά (συνέχεια)

Ο καθαρισμός του καμβά στο Βήμα 4 αφήνει ανέπαφη την εικόνα που πιθανώς να έχουμε θέσει ως φόντο. Μπορούμε να εισάγουμε ένα νέο κουμπί με όνομα ClearButton (εικόνα clear.png), 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).



Εικόνα 119 - Καθαρισμός φόντου με πάτημα κουμπιού



Όταν πατηθεί το πλήκτρο ClearButton ακύρωσε το τυχόν υπάρχον φόντο του καμβά.

Βήμα 9 Αποθήκευση των σχεδίων μας

Για να έχουμε τη δυνατότητα να αποθηκεύσουμε τα σχέδιά μας, προσθέτουμε καταρχάς ένα ακόμα κουμπί στην εφαρμογή μας, με όνομα SaveButton. Προκειμένου να ερωτάται ο χρήστης για το όνομα του αρχείου προς αποθήκευση, θα χρησιμοποιήσουμε και ένα αντικείμενο Notifier.



Το αντικείμενο Notifier επιτρέπει την εμφάνιση μηνυμάτων ή παραθύρων διαλόγου, από τα οποία μπορεί ο χρήστης να ενημερωθεί για κάτι, να επιλέξει ανάμεσα από δυο επιλογές ή ακόμα και να εισάγει την απάντηση που επιθυμεί, ώστε αυτή να χρησιμοποιηθεί στην εφαρμογή.

Designer Γραφικό Περιβάλλον Εφαρμογής

Εισάγουμε λοιπόν τα δυο προαναφερθέντα αντικείμενα.



από την
ομάδα

μεταφέρουμε το
αντικείμενο

του δίνουμε το
όνομα

μεταβάλλουμε τις
ιδιότητες

User Interface

Button

SaveButton

Width: 50

Height: 40

Text:

Image: save.png

User Interface

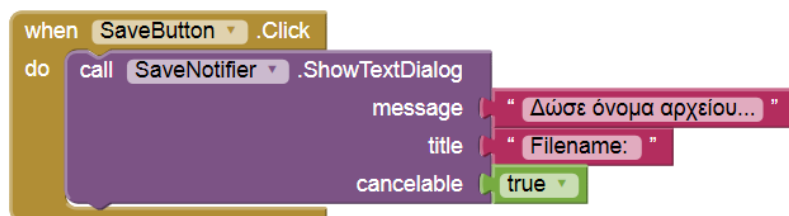
Notifier

SaveNotifier

Συγγραφή Προγράμματος

Blocks

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.

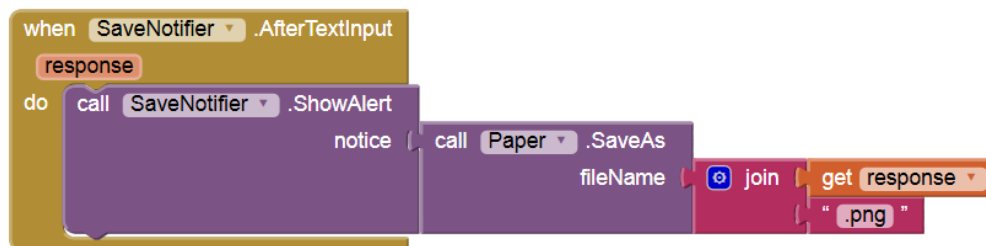


Εικόνα 120 - Ερώτημα για όνομα αρχείου προς αποθήκευση



Όταν πατηθεί το πλήκτρο SaveButton εμφανίσει ένα παράθυρο διαλόγου με τίτλο «Filename:», κυρίως μήνυμα «Δώσε όνομα αρχείου...», το οποίο μπορεί να ακυρωθεί από το χρήστη, αν το επιθυμεί.

Αμέσως μετά, θα αξιοποιήσουμε την απάντηση που δόθηκε για να αποθηκεύσουμε το σχέδιο.



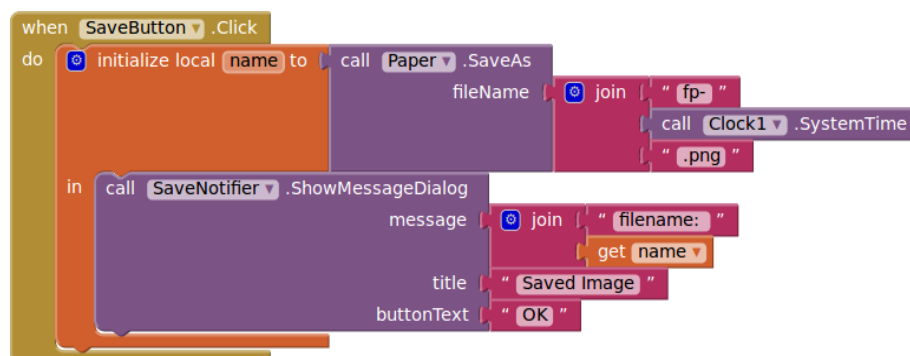
Εικόνα 121 - Αποθήκευση της ζωγραφιάς με το όνομα που δόθηκε



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 εμφάνισε σχετικό ενημερωτικό μήνυμα.

ΕΝΑΛΛΑΚΤΙΚΟΣ ΤΡΟΠΟΣ ΑΠΟΘΗΚΕΥΣΗΣ ΜΕ ΑΥΤΟΜΑΤΗ ΟΝΟΜΑΣΙΑ ΑΡΧΕΙΟΥ

Εναλλακτικά, η ζωγραφιά μας αποθηκεύεται σε ένα αρχείο με όνομα που ξεκινάει με *fp-* και ακολουθείται από έναν μοναδικό αριθμό. Ο μοναδικός αυτός αριθμός προέρχεται από το ρολόι του συστήματος (της συσκευής). Για να χρησιμοποιήσουμε τη δυνατότητα αυτή, εισάγουμε επιπλέον στην εφαρμογή μας ένα συστατικό Clock (με όνομα Clock1).



Εικόνα 122 - Αποθήκευση του σχεδίου με αυτόματη καταχώρηση ονόματος

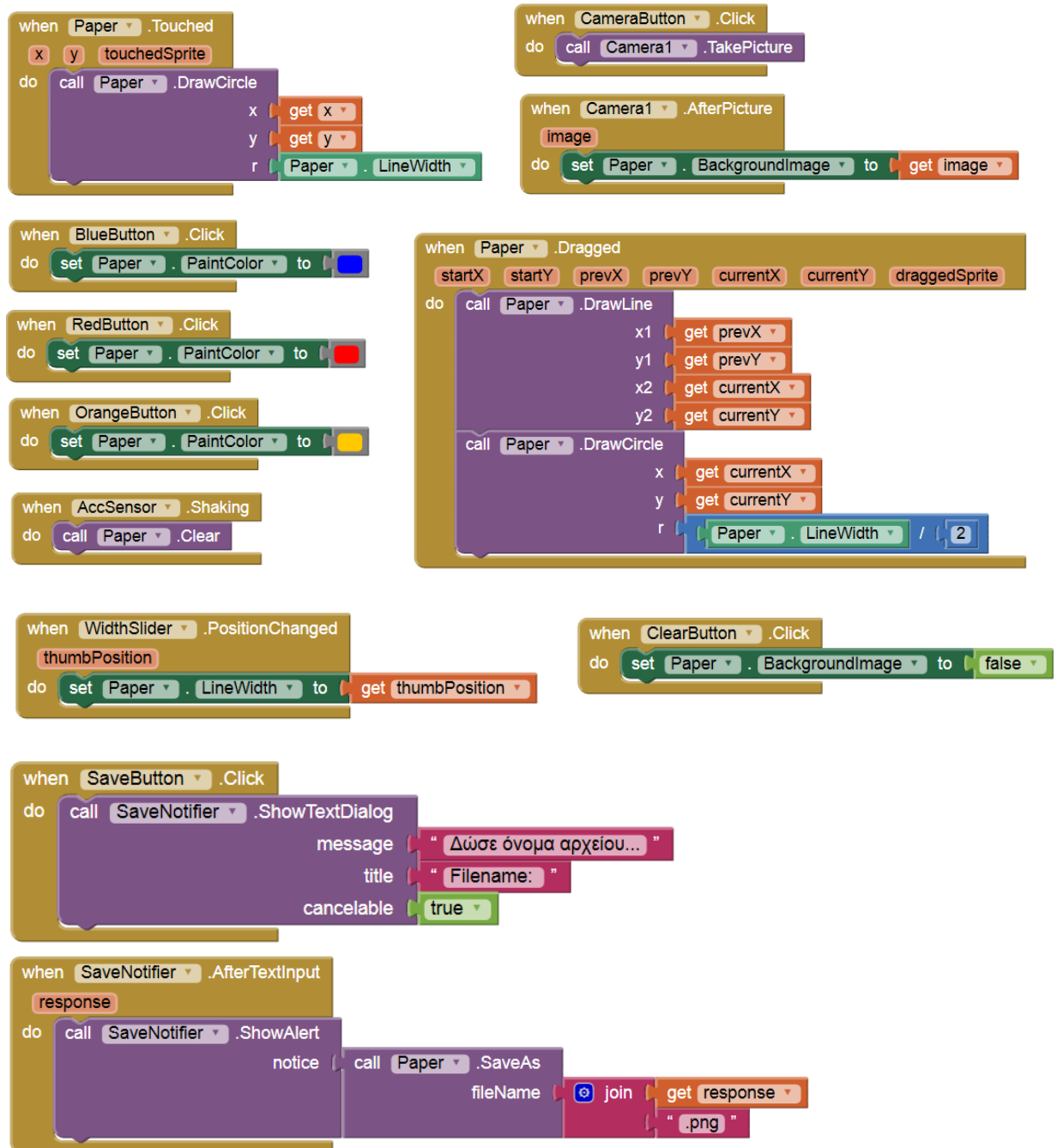


Όταν πατηθεί το πλήκτρο SaveButton, θέσε σε μια τοπική μεταβλητή με όνομα *name* το όνομα αρχείου *fp-[ώρα_συστήματος].png* και αποθήκευσε τη ζωγραφιά του καμβά στη συσκευή με το συγκεκριμένο όνομα αρχείου. Εμφάνισε ένα παράθυρο διαλόγου με τίτλο *Saved Image*, με κυρίως μήνυμα *filename: [όνομα_αποθηκευμένης_εικόνας]* και με ένα κουμπί *OK*.



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

Ολοκληρωμένη η εφαρμογή Finger Painting περιλαμβάνει τις παρακάτω εντολές ...



Εικόνα 123 - Οι εντολές της εφαρμογής Finger Painting

και το πηγαίο αρχείο με το πλήρες σετ εντολών που μπορείτε να εισάγετε απευθείας στο App Inventor βρίσκεται στο φάκελο FingerPainting του συνδέσμου <http://tiny.cc/aifiles>.