



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΦΑΡΜΟΓΗ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΖΩΓΡΑΦΙΚΗΣ ΣΕ ΓΛΩΣΣΑ

KOTLIN ΓΙΑ ΣΥΣΚΕΥΕΣ ANDROID

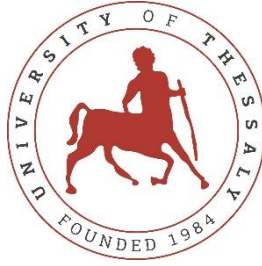
(CANVAS)

Διπλωματική Εργασία

Μηνάς Παπαδάκης

Επιβλέπων: Γεώργιος Θάνος

Βόλος, Σεπτέμβριος 2021



UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**PAINTING ANDROID APPLICATION USING KOTLIN
PROGRAMMING LANGUAGE
(CANVAS)**

Diploma Thesis

Minas Papadakis

Supervisor: George Thanos

Volos, September 2021

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα **Γεώργιος Θάνος**

Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Χρήστος Αντωνόπουλος**

Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών
Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Αθανάσιος Φεύγας**

Αναπληρωτής Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 23-09-2021

ΣΧΟΛΙΑ

Στα πλαίσια των σπουδών μου σε Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, αποφάσισα να επικεντρωθώ στο δεύτερο κομμάτι του τίτλου: το Software. Συγκεκριμένα, η δημιουργία εφαρμογών για κινητά, μου κίνησε σε μεγάλο βαθμό το ενδιαφέρον. Τόσο μεγάλο, που επιθυμούσα μέχρι και η διπλωματική μου εργασία να είναι μια ευκαιρία να δουλέψω σε μια τέτοια, πολύπλοκη εφαρμογή· και μέσα από την έρευνα και εργασία, να πετύχω *πρώτα* την αυτοβελτίωση, την παροχή υλικού σε συναδέλφους με ίδια ενδιαφέροντα, και *τελευταία* την απόκτηση ενός βαθμού ή και πτυχίου. Δημιουργώντας λοιπόν την «Canvas», μια εφαρμογή ζωγραφικής και επεξεργασίας εικόνων η οποία παρουσιάζεται στην παρούσα διπλωματική, είμαι βέβαιος ότι έχω προσεγγίσει όσο περισσότερο μπορούσα τους παραπάνω σκοπούς.

Θέλω να ευχαριστήσω θερμά τον Δρ. Θάνο Γεώργιο για την άψογη συνεργασία και την προθυμία να βοηθήσει σε κάθε πρόβλημα μου – και στην παρούσα διπλωματική, αλλά και στα πλαίσια των μαθημάτων.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών



Μηνάς Παπαδάκης
20-09-2021

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία πραγματεύεται την κατασκευή μιας Android εφαρμογής ζωγραφικής και σχεδίασης για κινητά ή tablet. Ο χρήστης έχει στην διάθεση του διάφορα εργαλεία και λειτουργίες, ώστε να ζωγραφίσει πάνω σε καμβάδες διαφόρων διαστάσεων, να αποθηκεύσει την δημιουργία του και να την κοινοποιήσει. Δημιουργήθηκε με την βοήθεια του περιβάλλοντος ανάπτυξης «Android Studio», και ο κώδικας είναι γραμμένος στην αντικειμενοστραφή γλώσσα *Kotlin*, η επίσημη γλώσσα δημιουργίας Android εφαρμογών. Στόχος είναι η εκμάθηση και απόκτηση εμπειρίας στο πλαίσιο αυτό μέσω της ανάπτυξης αυτής της εφαρμογής.

ABSTRACT

This thesis presents the implementation of an Android painting and drawing application, targeting mobile and tablet devices. The user has at their disposal various tools and functions, in order to draw on canvases of different sizes, save their creation, and share it. The app was created in the context of «Android Studio» development environment, and the code is written in the object-oriented language *Kotlin*, the official programming language for Android app development. The purpose of developing this application is learning, and gaining experience in the field during its creation.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ	<i>xī</i>
ABSTRACT	<i>xiii</i>
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	<i>xv</i>
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	<i>1</i>
1.1 Γνωστές εφαρμογές ζωγραφικής.....	<i>2</i>
1.2.1 - Autodesk Sketchbook	3
1.2.2 - Clip Studio Paint.....	3
1.2.3 - IbisPaint	4
1.2 Εισαγωγή στην εφαρμογή Canvas	<i>5</i>
1.3 Δυνατότητες της Canvas	<i>8</i>
1.4 Διαμόρφωση της εργασίας	<i>10</i>
ΚΕΦΑΛΑΙΟ 2 ΛΕΙΤΟΥΡΓΙΚΗ ΠΕΡΙΓΡΑΦΗ.....	<i>11</i>
2.1 Κυρίως Μενού	<i>11</i>
2.1.1 - Gallery	12
2.1.2 - New Canvas.....	13
2.1.3 - About	16
2.1.4 - Feedback.....	16
2.1.5 - Rate app.....	16
2.2 Ο Καμβάς	<i>17</i>
2.2.1 - Αρχεία CANVAS (.cv)	19
2.2.2 - Move / Hide Toolbar	19
2.2.3 - Εργαλείο Brush	21
2.2.4 - Εργαλείο Eraser	24
2.2.5 - Εργαλείο Bucket	25
2.2.6 - Εργαλείο Eyedropper.....	26
2.2.7 - Εργαλείο Shape.....	26

2.2.8 - Εργαλείο Text.....	29
2.2.9 - Layer List	30
2.2.10 - Επιλογή Transform Canvas	33
2.2.11 - Save Canvas	34
2.2.12 - Undo - Redo	35
ΚΕΦΑΛΑΙΟ 3 ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID.....	36
3.1 Εισαγωγή.....	36
Μοντέλο Android εφαρμογών.....	37
3.2 Το Android Studio	40
3.3 Απαιτήσεις της εφαρμογής	43
3.3.1 - JetBrains.kotlin	44
3.3.2 - Material Design.....	45
3.3.3 - Google Firebase	45
3.3.4 - ViewModel.....	47
3.3.5 - Navigation UI	48
3.4 Η γλώσσα Kotlin	48
ΚΕΦΑΛΑΙΟ 4 ΛΕΠΤΟΜΕΡΗΣ ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ.....	50
4.1 Android Manifest.....	51
4.2 Πακέτα κώδικα	53
4.2.1 - Γνωστές Κλάσεις	54
4.2.2 – Ο κώδικας της Canvas	55
4.3 Resources	61
4.3.1 - drawable	62
4.3.2 - font	62
4.3.3 - layout	63
4.3.4 - menu	64
4.3.5 - navigation	64

4.3.6 - values	65
4.3.7 - xml	66
4.4 Gradle Scripts	66
ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	68
5.1 Συμπεράσματα	68
5.2 Μελλοντικές επεκτάσεις	69
5.2.1 - Bugs.....	69
5.2.2 - Προσθήκη έξτρα λειτουργιών	70
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	73
ΠΑΡΑΡΤΗΜΑΤΑ.....	78
ΠΑΡΑΡΤΗΜΑ Α ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	80
ΠΑΡΑΡΤΗΜΑ Β Υλοποίηση ενός <i>OnClickListener</i>	82
ΠΑΡΑΡΤΗΜΑ Γ Παράδειγμα ενός <i>layout</i>	84

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η τεχνολογία γίνεται ολοένα και περισσότερο κομμάτι της καθημερινότητας των ανθρώπων με τα κινητά «smartphones» να διεκδικούν ιδιαίτερα υψηλή θέση σε αυτές. Με την χρήση τους, η καθοδήγηση με εφαρμογές GPS, η λήψη και κοινοποίηση φωτογραφιών, η επικοινωνία, η ενημέρωση, και η παροχή διαφόρων υπηρεσιών έχουν διευκολυνθεί. Μπορεί να υπάρχουν συσκευές ή εργαλεία όπου κάνουν τις ίδιες δουλειές, ίσως και καλύτερα, όπως ενδεχομένως οι φωτογραφικές μηχανές. Όμως, ο συνδυασμός όλων αυτών των παροχών σε μια συσκευή τόσο μικρού μεγέθους, είναι αυτό που κάνει το smartphone τόσο χρηστικό. Σε όλες τις πτυχές της ζωής του ανθρώπου, το εργαλείο αυτό προσφέρει εύκολες λύσεις σε πολλές ανάγκες και προβλήματα, με όποιον τρόπο μπορεί.

Η ζωγραφική και η επεξεργασία φωτογραφιών, λοιπόν, δεν είναι σε καμία περίπτωση εξαίρεση σε αυτό. Σε σχέση με την παραδοσιακή, η ψηφιακή ζωγραφική με smartphone ή tablet διαθέτει κάποια προνόμια, όπως η εύκολη πρόσβαση σε μεγάλη ποικιλία από εργαλεία και καμβάδες, και επιπλέον, η έλλειψη της ανάγκης για την μεταφορά τους. Σημαντικότερο όμως προνόμιο αποτελεί η αποδοτικότητα. Με την ψηφιακή ζωγραφική, η γόμα δεν αφήνει κανένα ίχνος, τα χρώματα και το μέγεθος αντικειμένων μπορούν να αλλάξουν μέσα σε δευτερόλεπτα. Μέχρι και το ταξίδι «πίσω στον χρόνο» είναι δυνατό με ένα τόσο απλό εργαλείο όσο το «Undo». Σίγουρα, βέβαια, διευκρινίζεται ότι δεν απορρίπτεται η σημασία της φυσικής, απτικής εμπειρίας, και της αυθεντικότητας που προσφέρει η παραδοσιακή ζωγραφική. [1]

Φυσικά όλα αυτά τα προνόμια γίνονται όλο και πιο εμφανή όταν αξιοποιούνται οι τελευταίες τεχνολογίες στην συσκευή σχεδίασης, τόσο σε λογισμικό όσο και υλικό. Για τα smartphones, συγκεκριμένα, υπάρχουν σήμερα πάρα πολλές εφαρμογές οι οποίες το επιδιώκουν αυτό, με αποτέλεσμα να χρησιμοποιούνται πλέον και επαγγελματικά. Στο κεφάλαιο αυτό θα παρουσιαστούν πολλές τέτοιες γνωστές εφαρμογές, όπως και μια εισαγωγή στην «Canvas», την Android εφαρμογή ζωγραφικής στην οποία επικεντρώνεται η παρούσα διπλωματική.

1.1 Γνωστές εφαρμογές ζωγραφικής

Για κάθε είδος πλατφόρμας, υπάρχει μεγάλη ποικιλία λογισμικών για ψηφιακή σχεδίαση και επεξεργασία φωτογραφιών. Κάποια παρέχουν πληθώρα λειτουργιών και εργαλείων, ίσως με κόστος την απαίτηση αρκετού χρόνου για εξοικείωση, κάποια άλλα το αντίθετο. Κάποια όμως, έχουν φτάσει πολύ κοντά στην χρυσή τομή ανάμεσα στην ευχρηστία και την λειτουργικότητα. Συνδυάζοντας κάποιες γνωστές ιστοσελίδες [2, 3, 4, 5, 6, 7, 8, 9], ορίστε μια λίστα με τα καλύτερα προγράμματα ψηφιακής ζωγραφικής του 2021, αναλογίζοντας την βαθμολογία τους στα συγκεκριμένα άρθρα αλλά και την δημοτικότητα τους:

1. Photoshop	Windows, macOS, iPad
2. Corel Painter	Windows, macOS
3. Rebelle	Windows, macOS
4. Clip Studio Paint	Windows, macOS, iOS, Android, Chromebook
5. Affinity Photo	Windows, macOS, iOS
6. Krita	Windows, macOS, Linux
7. ArtRage	Windows, macOS
8. Autodesk SketchBook	Windows, macOS, iOS, Android
9. IbisPaint	iOS, Android
10. Procreate	iPad
11. Astropad Studio	macOS, iPad
12. Pixelmator	macOS
13. Adobe Illustrator	Windows, macOS
14. Adobe Photoshop Sketch	iOS, Android
15. Artweaver	Windows
16. Paint Tool SAI	Windows

Εύκολα παρατηρεί κανείς, έστω και από αυτό το μικρό δείγμα, ότι η πλειοψηφία των εφαρμογών είναι φτιαγμένες για επιτραπέζιο υπολογιστή ή για tablet, καθώς προφανώς παρέχουν καλύτερη υπολογιστική ισχύ και μεγαλύτερες οθόνες από smartphones.

Παρόλα αυτά, οι εφαρμογές για κινητά που υπάρχουν σήμερα είναι παραπάνω από επαρκείς για ψηφιακή ζωγραφική. Οι επόμενες υπο-ενότητες πραγματεύονται τον ισχυρισμό αυτό.

1.2.1 - Autodesk Sketchbook

Διαθέσιμο για Windows, macOS, iOS και Android.

Το Sketchbook είναι μια βραβευμένη εφαρμογή σκίτσων της Autodesk η οποία χρησιμοποιείται από επαγγελματίες όπως αρχιτέκτονες, σχεδιαστές και καλλιτέχνες σε πολλές διαφορετικές πλατφόρμες. Επικεντρώνεται στην ζωγραφική και το σχέδιο σε λιγότερο χρόνο, με ένα σετ χαρακτηριστικών επαγγελματικού επιπέδου, οδηγούς, χάρακες, και προσαρμόσιμα εργαλεία όπως μολύβια, μαρκαδόροι, αερογράφοι, επιχρίσματα και άλλα εργαλεία που μοιάζουν με τα φυσικά τους αντίστοιχα. Η διεπαφή του χρήστη είναι διακριτική, ώστε να μπορεί ο χρήστης να εστιάσει στο σχέδιο. Τέλος, παρέχεται και η δυνατότητα χρήσης της κάμερας της συσκευής για την εισαγωγή φωτογραφιών στο έργο. [10, 11]



Εικόνα 1.1 – Το logo του Sketchbook [12]

1.2.2 - Clip Studio Paint

Διαθέσιμο για Windows, macOS, iOS, Android και Chromebook.

Την χρησιμοποιούν πάνω από 10 εκατομμύρια χρήστες και επαγγελματίες παγκοσμίως, για εικονογράφηση ή δημιουργία κόμικς και κινουμένων σχεδίων, με, ή χωρίς ήχο.

Παρέχονται πολλά εργαλεία για δημιουργία προσαρμόσιμων αντικειμένων μέχρι και στις 3 διαστάσεις. Όλα αυτά μπορεί ο χρήστης να τα μάθει μέσω των δωρεάν μαθημάτων που παρέχονται στον επίσημο ιστότοπο του Clip Studio Paint ή στο κανάλι YouTube. [13]



Εικόνα 1.2 – Το logo του Clip Studio [14]

1.2.3 - IbisPaint

Διαθέσιμο για iOS και Android.

Το IbisPaint είναι μια δημοφιλής και ευέλικτη εφαρμογή σχεδίασης που έχει ληφθεί περισσότερες από 150 εκατομμύρια φορές συνολικά. Παρέχει πάνω από 2700 υλικά (όπως πινέλα, γραμματοσειρές, φίλτρα κ.α.), εγγραφή timelapse σχεδίασης, δυνατότητα σταθεροποίησης γραμμών και χάρακες όπως χάρακες ακτινικής γραμμής ή χάρακες συμμετρίας, και η premium έκδοση παρέχει ακόμα παραπάνω. Πολλά εκπαιδευτικά βίντεο για το IbisPaint ανεβαίνουν στο κανάλι του στο YouTube.[15]



Εικόνα 1.3 – Το logo του IbisPaint [16]

Η συγκεκριμένη εφαρμογή έχει χρησιμοποιηθεί ως έμπνευση για την δημιουργία της εφαρμογής που παρουσιάζεται.

1.2 Εισαγωγή στην εφαρμογή Canvas

Η εφαρμογή της παρούσας διπλωματικής, είναι ένα αποτέλεσμα έξι μηνών προγραμματισμού με σκοπό αυτή να περιέχει όσα το δυνατόν περισσότερα από τα χαρακτηριστικά των παραπάνω γνωστών εφαρμογών. Ονομάζεται «Canvas», και προορίζεται για smartphones ή tablets τα οποία χρησιμοποιούν λειτουργικό *Android*. Ειδικεύεται στην σχεδίαση και ζωγραφική πάνω σε κενό καμβά ή σε φωτογραφίες.



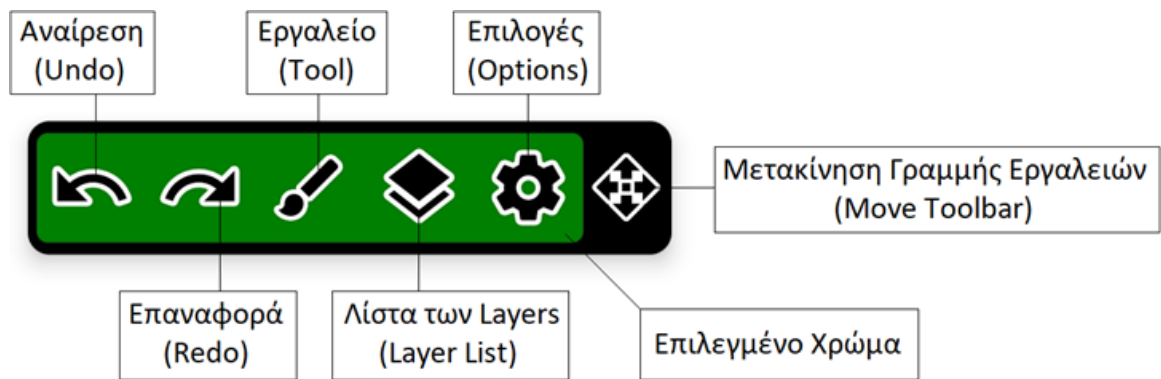
Εικόνα 1.4 – Το logo της εφαρμογής Canvas

Βασικός στόχος κατά την σχεδίαση της εφαρμογής πέρα από την λειτουργικότητα, ήταν και η ευχρηστία της. Χρησιμοποιώντας άλλες παρόμοιες εφαρμογές, ο χρήστης συνήθως βρίσκεται κατακλυσμένος από την χαστική παροχή κουμπιών-εργαλείων και στρωμάτων (layers¹). Ως αποτέλεσμα, χρειάζεται σπατάλη πολλών ωρών στην προσπάθεια για εξοικείωση, ώστε να μπορεί να τις χειριστεί στο έπακρο. Προτεραιότητα, επομένως, ήταν να εξασφαλιστεί στο *User Interface*² της εφαρμογής η απλότητα, αλλά όχι η απλοϊκότητα.

Στην εικόνα 1.5, απεικονίζεται η μπάρα εργαλείων (Toolbar), η οποία βρίσκεται συνήθως στην κορυφή της οθόνης. Αυτή περιέχει μόνο 6 κουμπιά, τα οποία φέρουν εικονίδια σύμφωνα με παγκόσμια πρότυπα για να είναι ευνόητη η χρήση τους:

¹ *Layers*: Η συνολική εικόνα του καμβά απαρτίζεται από ένα σύνολο στρώσεων, η μια πάνω στην άλλη. Αυτές οι στρώσεις, είναι γνωστές ως layers.

² *User Interface (UI)*: Διεπαφή χρήστη. Ένα σύνολο εικονικών στοιχείων που εμφανίζονται στην οθόνη μιας ψηφιακής συσκευής, με τα οποία αλληλοεπιδρά ο χρήστης.



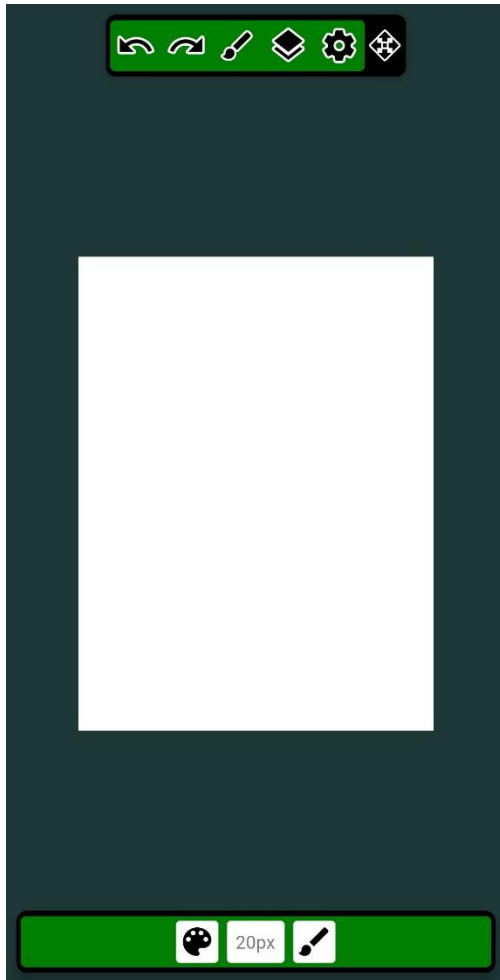
Εικόνα 1.5 – Toolbar

Η βασική ιδέα πίσω από την απλότητα του συγκεκριμένου User Interface βασίζεται στο ότι όταν ο χρήστης επιλέξει κάποιο εργαλείο ή λειτουργία από το Toolbar, τότε θα εμφανίζεται το “Bottom Toolbar”, μια δεύτερη γραμμή εργαλείων στο κάτω μέρος της οθόνης, στο οποίο η εφαρμογή θα παρέχει τις ενδεχόμενες ιδιότητες κάθε επιλογής.

Για παράδειγμα, όπως φαίνεται στην εικόνα 1.6, όταν ο χρήστης επιλέξει πινέλο, τότε οι ιδιότητες του στην κάτω γραμμή εργαλείων θα περιλαμβάνουν την επιλογή χρώματος, μεγέθους και μοτίβου του πινέλου. Όταν, από τη άλλη, πατηθεί το κουμπί «List of Layers», τότε θα εμφανίζεται η λίστα (εικόνα 1.7).

Με αυτόν τον τρόπο, ο χρήστης μπορεί μεθοδικά να βρει τις ιδιότητες από οποιαδήποτε λειτουργία θέλει να εφαρμόσει, χωρίς να του προβάλλονται «αποπνικτικά» όλες αυτές μαζί.

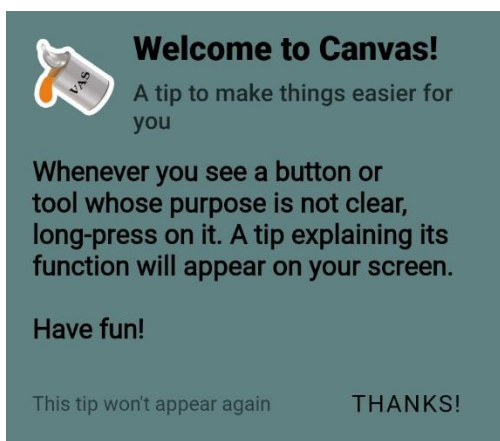
Τέλος, αξίζει να σημειωθεί ότι στην προσπάθεια να γίνει ακόμα πιο εύχρηστη η εφαρμογή, προστέθηκαν σημειώσεις «tips», για κάθε εικονικό στοιχείο του οποίου η χρήση δεν είναι ευνόητη. Το μόνο που έχει να κάνει ο χρήστης είναι να πατήσει παρατεταμένα στο στοιχείο αυτό, και ένας διάλογος που εξηγεί την χρήση του εμφανίζεται στην οθόνη. Παραδείγματα τέτοιων σημειώσεων φαίνονται στις εικόνες 1.8 και 1.9.



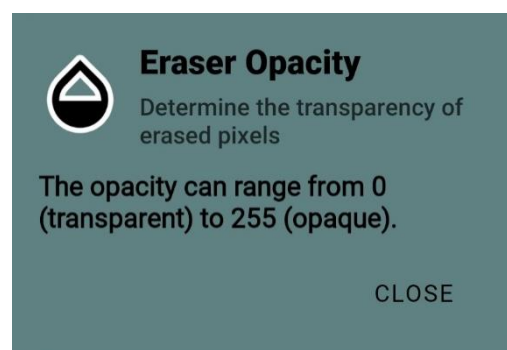
Εικόνα 1.6 – Bottom Toolbar – Brush Properties



Εικόνα 1.7 – Bottom Toolbar – Layer List



Εικόνα 1.8 – Welcoming Tip



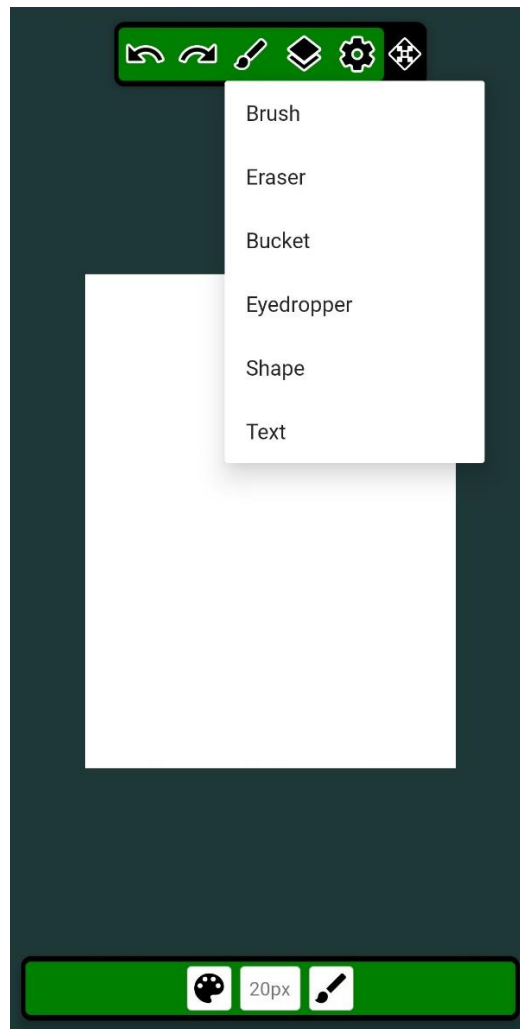
Εικόνα 1.9 – Long-Press σε ένα στοιχείο

1.3 Δυνατότητες της Canvas

Έχοντας δουλέψει στο παρελθόν σε προγράμματα όπως αρχικά η ζωγραφική των Windows, το IbisPaint, και πιο πρόσφατα το Photoshop της Adobe, ήταν σαφές ποια εργαλεία και λειτουργίες δεν θα έπρεπε να λείπουν από την εφαρμογή. Μαζί με κάποιες έξτρα προσθήκες, η Canvas έχει τις εξής δυνατότητες:

1. Δημιουργία καμβά συγκεκριμένων διαστάσεων: είτε προεπιλεγμένες, είτε με παραμέτρους pixel/ίντσες/χιλιοστά/dpi.
2. Δυνατότητα εύκολης μεγέθυνσης, ελαχιστοποίησης, μετακίνησης και περιστροφής του καμβά, χρησιμοποιώντας τα δυο δάχτυλα. Φυσικά, δίνεται και δυνατότητα επαναφοράς στην αρχική κατάσταση.
3. Δυνατότητα να ελαχιστοποιούνται τα πάνω και κάτω Toolbar ώστε να φαίνεται μεγαλύτερο ποσοστό του καμβά.
4. Αποθήκευση του καμβά ως αρχείο εικόνας στην συσκευή και δυνατότητα επεξεργασίας, διαγραφής ή κοινοποίησης αυτού.
5. Undo/Redo.
6. Εργαλεία ζωγραφικής (*Tools*, εικόνα 1.10):
 - a. Βούρτσα (*Brush*): Το πιο βασικό εργαλείο για ζωγραφική. Αποτυπώνει το μονοπάτι που διανύει το δάχτυλο πάνω στον καμβά.
 - b. Σβήστρα (*Eraser*): Λειτουργεί σαν ένα πινέλο, εκτός ότι σβήνει αντί για να ζωγραφίζει.
 - c. Κουβάς (*Bucket*): Πατώντας πάνω σε ένα σημείο του καμβά, ο κουβάς χρωματίζει αυτό και όλη την περιοχή γύρω από εκείνο το σημείο η οποία είχε το ίδιο χρώμα με το σημείο.
 - d. Σταγονόμετρο (*Eyedropper*): Όταν ο χρήστης πατήσει πάνω σε ένα σημείο του καμβά, επιλέγεται ως χρώμα το χρώμα εκείνου του σημείου.
 - e. Σχήμα (*Shape*): Ο χρήστης έχει την δυνατότητα να εισάγει γεωμετρικά σχήματα όπως ο κύκλος και το παραλληλόγραμμο στον καμβά.
 - f. Κείμενο (*Text*): Εισάγοντας μια λέξη-φράση και πατώντας πάνω στον καμβά, το συγκεκριμένο κείμενο ζωγραφίζεται πάνω του.

7. Layers: ο χρήστης μπορεί να επεξεργαστεί τα layers του καμβά ως εξής:
- a. Αλλαγή προτεραιότητας στη λίστα
 - b. Προσθήκη
 - c. Εμφάνιση/κρύψιμο
 - d. Ένωση δυο ή παραπάνω layers σε ένα
 - e. Δημιουργία αντιγράφου
 - f. Αλλαγή αδιαφάνειας (opacity)
 - g. Σβήσιμο περιεχομένου
 - h. Διαγραφή



Εικόνα 1.10 – Μενού με τα Εργαλεία Ζωγραφικής

1.4 Διαμόρφωση της εργασίας

Στο επόμενο κεφάλαιο, γίνεται εκτενής ανάλυση των λειτουργιών της εφαρμογής, με καθοδήγηση από την πρώτη οθόνη που συναντά ο χρήστης μέχρι και την τελευταία που ενδέχεται να συναντήσει, αναλύοντας παράλληλα τις επιλογές που του παρέχονται.

Στο Κεφάλαιο 3 παρουσιάζεται το λειτουργικό Android, το πρόγραμμα «Android Studio» με το οποίο δημιουργήθηκε η εφαρμογή, η γλώσσα προγραμματισμού Kotlin, και όλα τα έτοιμα εργαλεία λογισμικού που χρησιμοποιούνται από την Canvas.

Στο Κεφάλαιο 4 γίνεται μια λεπτομερής τεχνική περιγραφή της εφαρμογής, από άποψη αρχιτεκτονικής κώδικα και πόρων.

Στο Κεφάλαιο 5 παρουσιάζονται κάποια συμπεράσματα, όπως και πιθανές μελλοντικές επεκτάσεις που μπορούν να εφαρμοστούν στην Canvas.

Έπειτα ακολουθεί η Βιβλιογραφία, στην οποία παραπέμπει κάθε αριθμός κλεισμένος σε αγκύλες [] μέσα στο κείμενο.

Τέλος, ακολουθούν τα παραρτήματα. Το παράρτημα Α «Συντομογραφίες», επεξηγεί με αλφαβητική σειρά τυχόν ακρωνύμια που εμφανίζονται στο κείμενο.

Το παράρτημα Β παρέχει ένα παράδειγμα προγραμματισμού ενός κουμπιού ώστε να πετυχαίνει έναν συγκεκριμένο σκοπό.

Το παράρτημα Γ παρέχει ένα παράδειγμα προγραμματιστικής δημιουργίας ενός *layout*, δηλαδή μίας συγκεκριμένης διάταξης UI στοιχείων.

Το τελευταίο, παράρτημα Δ, περιέχει οδηγίες για να μπορέσει ο αναγνώστης να κατεβάσει και εγκαταστήσει την Canvas και στην δικιά του συσκευή Android.

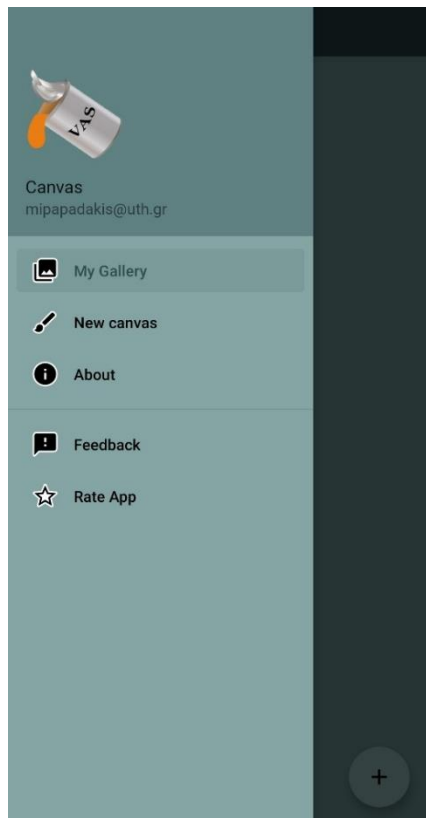
ΚΕΦΑΛΑΙΟ 2

ΛΕΙΤΟΥΡΓΙΚΗ ΠΕΡΙΓΡΑΦΗ

Σε αυτό το κεφάλαιο παρουσιάζονται βήμα-βήμα κάθε οθόνη στην οποία μπορεί να οδηγηθεί ο χρήστης. Έπειτα, εξηγείται η κάθε λειτουργία για την κατασκευή των εικόνων.

2.1 Κυρίως Μενού

Στο κυρίως μενού, συμπεριλαμβάνονται πέντε επιλογές, όπως φαίνεται στην εικόνα 2.1:



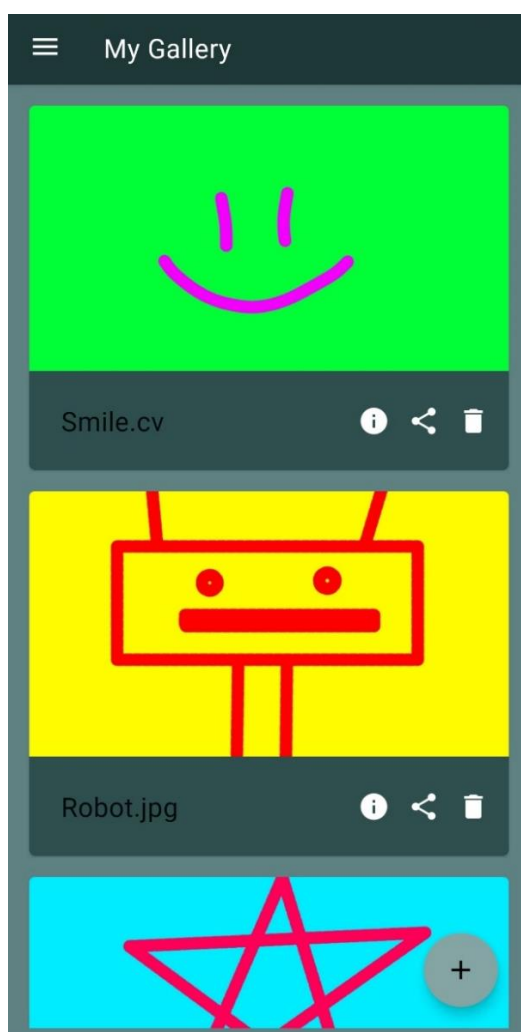
Εικόνα 2.1 – Κυρίως Μενού

Το μενού αυτό μπορεί να εμφανιστεί όταν ο χρήστης σύρει το δάχτυλο από την άκρη στα αριστερά προς τα δεξιά, ή πατώντας το κουμπί το οποίο βρίσκεται στην πάνω αριστερά γωνία όταν το μενού είναι κλειστό.

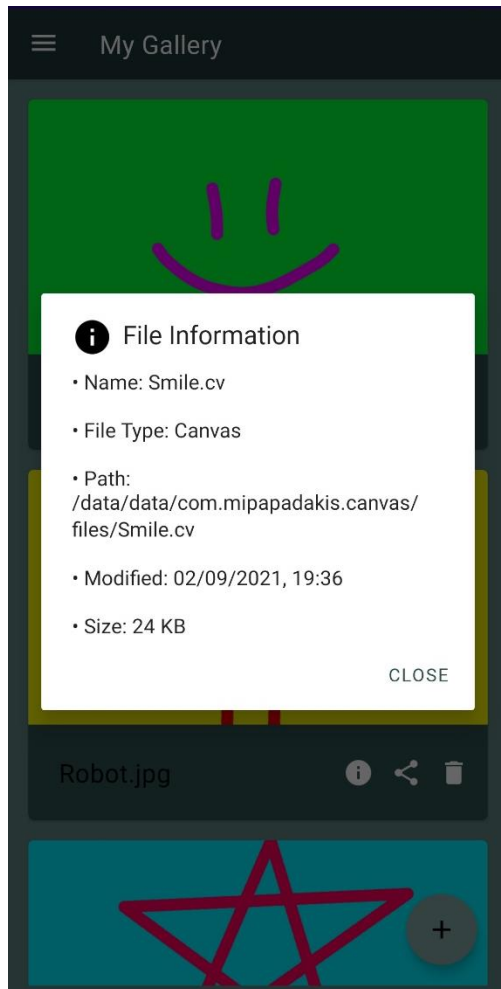
2.1.1 - Gallery

Η πρώτη επιλογή του κυρίως μενού, και η πρώτη οθόνη που συναντάται ανοίγοντας την εφαρμογή. Το Gallery περιλαμβάνει μια λίστα με τις αποθηκευμένες εικόνες που έχουν δημιουργηθεί στο παρελθόν (εικόνα 2.2). Για κάθε αρχείο από αυτά ο χρήστης να δει τις πληροφορίες του (εικόνα 2.3), να το κοινοποιήσει σε άλλες εφαρμογές (εικόνα 2.4), να το διαγράψει, και τέλος, πατώντας πάνω στην εικόνα, να το ανοίξει για επεξεργασία.

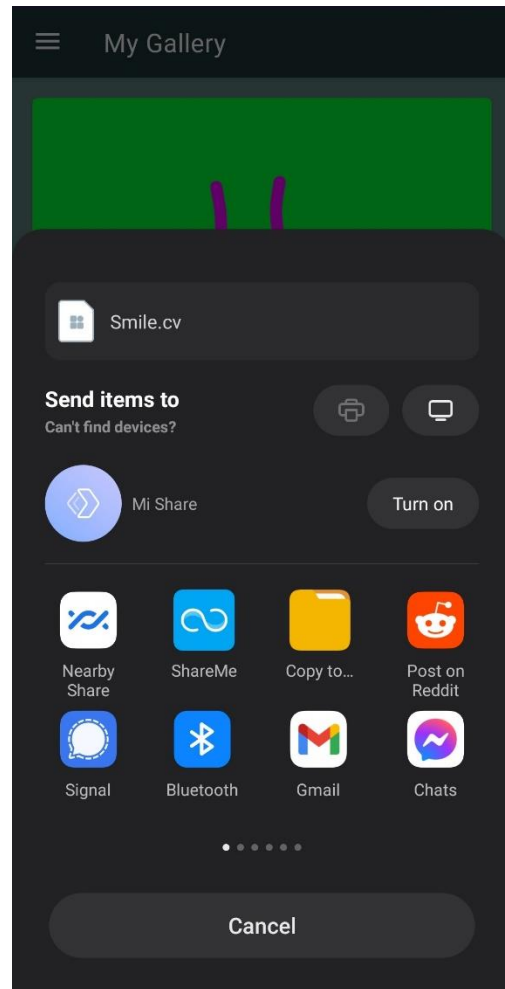
Η επεξεργασία περιγράφεται στην υπο-ενότητα «Ο Καμβάς» του κεφαλαίου στο οποίο βρισκόμαστε.



Εικόνα 2.2 – Gallery



Εικόνα 2.3 – Πληροφορίες αρχείου

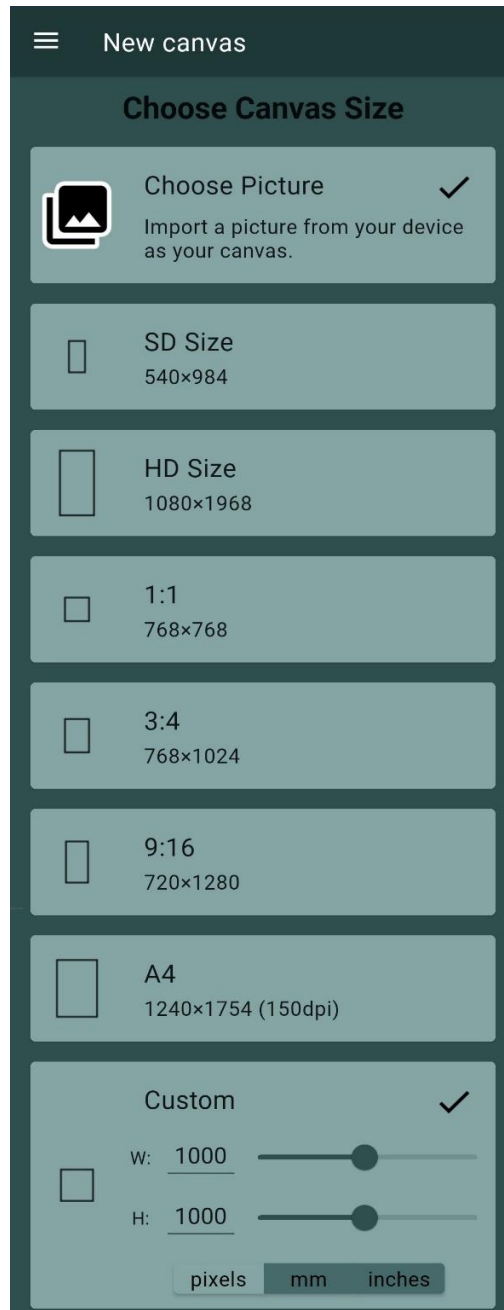


Εικόνα 2.4 – Κοινοποίηση αρχείου

2.1.2 - New Canvas

Η δεύτερη επιλογή του κυρίως μενού αφορά την αρχικοποίηση ενός νέου καμβά, όσον αφορά τις διαστάσεις και το φόντο. Υπάρχουν 8 επιλογές (εικόνα 2.5), εκ των οποίων η πρώτη προτρέπει τον χρήστη να χρησιμοποιήσει μια εικόνα από τα αρχεία της συσκευής του ως φόντο του καμβά, οι 6 επόμενες είναι προκαθορισμένες (HD, SD, 1:1, 3:4, 9:16, A4), και η τελευταία επιτρέπει την δημιουργία Custom διαστάσεων, με τρεις τρόπους:

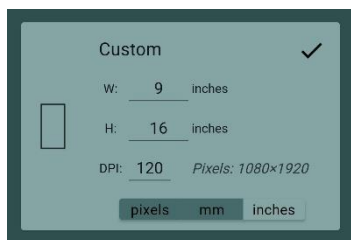
1. Εισαγωγή διαστάσεων (ύψος, πλάτος) σε pixels, από 1 έως και 2048.
2. Εισαγωγή ύψους και πλάτους σε ίντσες, και ανάλυσης σε dpi.
3. Εισαγωγή ύψους και πλάτους σε χιλιοστά, και ανάλυσης σε dpi.



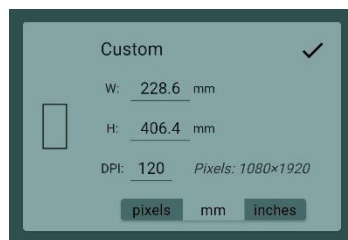
Εικόνα 2.5 – New Canvas

Σημειώνεται ότι όταν ο user εισάγει κάποιες διαστάσεις στο inches και μετά πατήσει mm, γίνεται αυτόματα μετατροπή αυτών των διαστάσεων από ίντσες σε χιλιοστά. Το ίδιο συμβαίνει και αντίστροφα (εικόνες 2.6, 2.7).

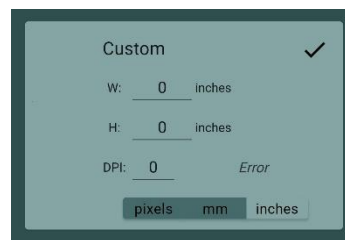
Όταν παρόλα αυτά ο καμβάς που προκύπτει από τις παραμέτρους έχει διαστάσεις σε pixel εκτός των ορίων, εμφανίζεται μήνυμα «Error» (εικόνα 2.8).



Εικόνα 2.6 – Custom Size: inch

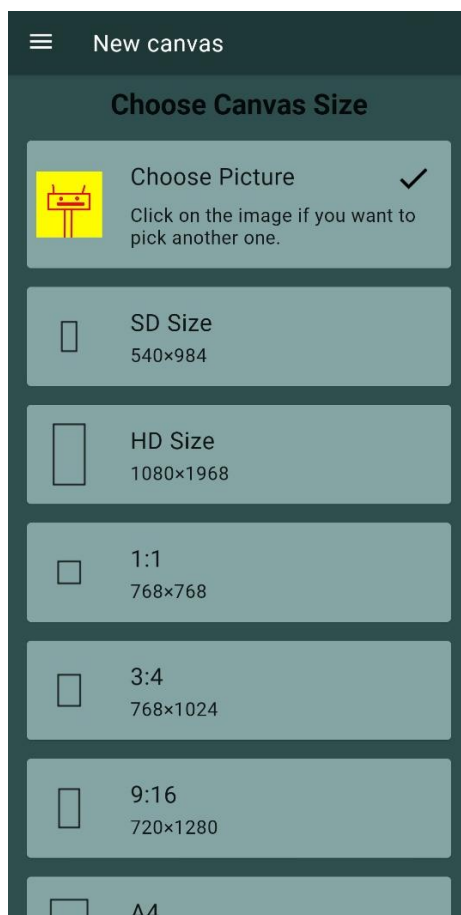


Εικόνα 2.7 – Custom Size: mm



Εικόνα 2.8 – Error

Τέλος, αξίζει να σημειωθεί ότι εφόσον ο χρήστης έχει επιλέξει μια εικόνα από την συσκευή του στην κάρτα «Choose Picture», το εικονίδιο της κάρτας δείχνει μια προεπισκόπηση της επιλεγμένης εικόνας, όπως φαίνεται στην εικόνα 2.9.

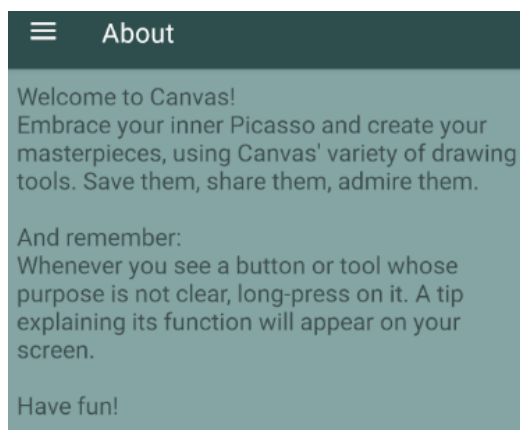


Εικόνα 2.9 – Επιλογή εικόνας ως καμβά

Πατώντας πάλι στο εικονίδιο, μπορεί να επιλέξει μια διαφορετική εικόνα.

2.1.3 - About

Η τρίτη επιλογή του μενού, περιλαμβάνει ένα απλό κείμενο το οποίο δίνει στον χρήστη μια σύντομη περιγραφή της εφαρμογής και των δυνατοτήτων της.



Εικόνα 2.10 – About

2.1.4 - Feedback

Η επιλογή feedback του κυρίως μενού, έχει σκοπό να επιτρέπει στον χρήστη την επικοινωνία με τον σχεδιαστή της εφαρμογής. Η επικοινωνία γίνεται μέσω email, το οποίο μπορεί να παρέχει πληροφορίες που θα χρησιμεύσουν στην βελτίωση της εφαρμογής, αναφορές προβλημάτων που μπορεί να μην έχουν λυθεί, θετικά ή αρνητικά σχόλια, και πολλά άλλα.

Με το πάτημα του κουμπιού η Canvas ζητάει από τον χρήστη να επιλέξει την εφαρμογή με την οποία θα στείλει την ανατροφοδότηση. Αφού επιλέξει, αυτή ανοίγει και ο παραλήπτης και το θέμα του email είναι ήδη συμπληρωμένα. Το μόνο που μένει πια για τον χρήστη είναι να συμπληρώσει το σώμα του μηνύματος.

2.1.5 - Rate app

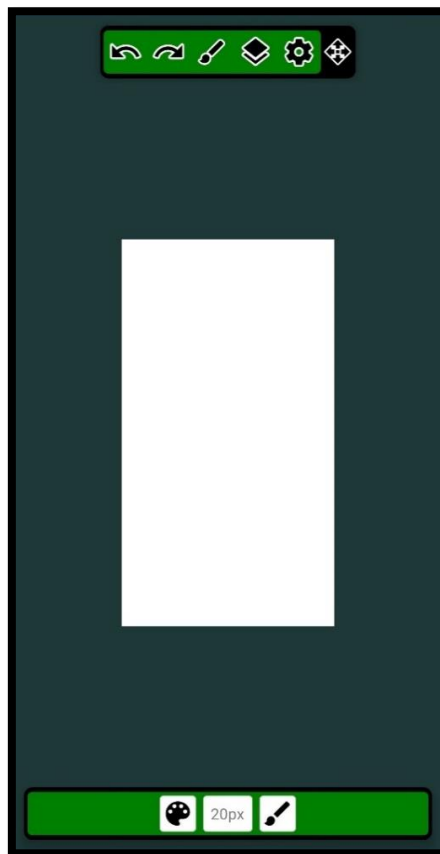
Τελευταία επιλογή του μενού αποτελεί το κουμπί «Rate App». Όταν πατηθεί, η Canvas ανοίγει το Play Store, κατευθείαν στην σελίδα της εφαρμογής, διευκολύνοντας έτσι τον χρήστη στη διαδικασία της αξιολόγησης.

2.2 Ο Καμβάς

Η ενότητα αυτή περιγράφει λεπτομερώς την οθόνη επεξεργασίας του καμβά. Σε αυτό το σημείο της εφαρμογής, ο χρήστης μπορεί να φτάσει με τρεις τρόπους:

1. Από το *Κυρίως Μενού* -> *New Canvas*, μόλις επιλεγεί μια από τις 8 διαστάσεις.
2. Όταν ο χρήστης επιλέξει μια εικόνα από το *Κυρίως Μενού* -> *Gallery*.
3. Όταν ο χρήστης επιλέξει να ανοίξει ένα οποιοδήποτε αρχείο στην συσκευή με την εφαρμογή Canvas. Προϋπόθεση βέβαια αποτελεί το αρχείο αυτό να είναι εικόνα μορφής PNG (.png), JPEG (.jpg) ή CANVAS (.cv). Η τελευταία, εξηγείται παρακάτω.

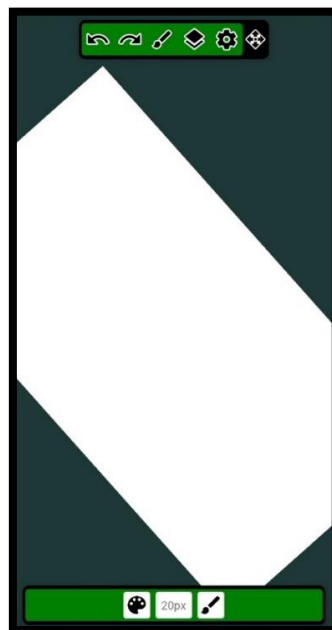
Έπειτα, δημιουργείται ο καμβάς όπως βλέπετε στην παρακάτω εικόνα (εικόνα 2.11). Στο παράδειγμα των εικόνων, έχει επιλεγεί *SD size* από το *Κυρίως Μενού* -> *New Canvas*.



Εικόνα 2.11 – Καμβάς μεγέθους SD size

Η τροποποίηση του καμβά, με την έννοια της θέσης, μεγέθυνσης και γωνίας, μπορούν να γίνουν εύκολα και γρήγορα χρησιμοποιώντας δύο δάχτυλα πάνω στην οθόνη.

- Όταν πλησιάσουν τα δύο δάχτυλα μεταξύ τους, γίνεται zoom out από τον καμβά.
- Όταν απομακρυνθούν, γίνεται μεγέθυνση.
- Όταν περιστραφούν τα δάχτυλα, ο καμβάς περιστρέφεται ανάλογα.
- Όταν μετακινηθούν και τα δύο προς μια κατεύθυνση, ο καμβάς ακολουθεί.



Εικόνα 2.12 – Επίδειξη τροποποίησης του καμβά

Πολύ συχνά, θα χρειαστεί η επαναφορά του καμβά στην αρχική του κατάσταση. Αυτό επιτυγχάνεται με δυο τρόπους:

1. Με κλικ τριών δακτύλων, ο καμβάς επιστρέφει στις αρχικές του διαστάσεις, θέση, και γωνία.
2. Με διπλό κλικ δυο δακτύλων, ο καμβάς επιστρέφει στην αρχική του θέση και γωνία, και μεγεθύνεται ή ελαχιστοποιείται ανάλογα, έτσι ώστε να εφάπτεται ακριβώς στην οθόνη του κινητού.

2.2.1 - Αρχεία CANVAS (.cv)

Σε πολλές εφαρμογές ψηφιακής ζωγραφικής, συνηθίζεται να έχουν δικό τους τύπο αρχείων, όπως ο τύπος PSD (.psd) του Photoshop. Αυτό συμβαίνει γιατί οι συνηθισμένοι τύποι αρχείων εικόνων, όπως JPEG, PNG και BITMAP, αποθηκεύουν μόνο τα δεδομένα εικόνας και τίποτε άλλο. Ο χρήστης, όμως, μπορεί να επιθυμεί να αποθηκεύσει όχι μόνο την συνολική εικόνα του δημιουργήματος του, αλλά και τα layers του έργου, συν πολλά άλλα δεδομένα τα οποία θα χαθούν εάν για παράδειγμα επιλέξει αποθήκευση σε PNG.

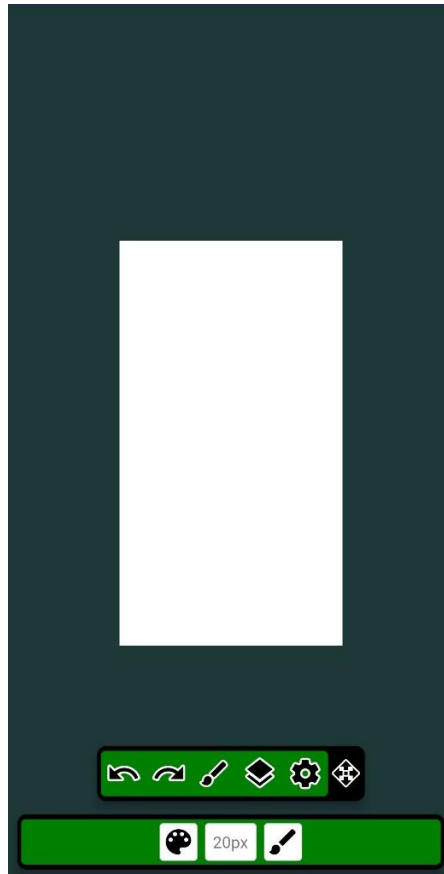
Για τον ίδιο λόγο, έχει δημιουργηθεί και το format «.cv», αποκλειστικά για την εφαρμογή Canvas και αναγνωρίζεται μόνο από αυτή. Σε αντίθεση με τους συνηθισμένους τύπους αρχείων εικόνας, ένα αρχείο CANVAS περιέχει αποθηκευμένα και τα layers, και όσα δεδομένα συμπεριλαμβάνονται σε κάθε layer. Επίσης, σημαντικό προτέρημα αποτελεί το γεγονός ότι τα CANVAS αρχεία μπορούν να αποθηκεύσουν εικόνες με *διαφανή pixel*, κάτι που, για παράδειγμα, τα αρχεία JPEG δεν μπορούν.

Η διαφάνεια, είναι ένα χαρακτηριστικό που διαθέτουν κάποιες μορφές εικόνων, οι οποίες αποθηκεύουν για κάθε pixel όχι μόνο χρώμα, αλλά και αδιαφάνεια. Η ιδιότητα αυτή παίρνει συνήθως το όνομα «opacity» ή «alpha». Ένα pixel με μηδενική αδιαφάνεια παίρνει το χρώμα του φόντου, δηλαδή του σημείου της οθόνης που βρίσκεται «πίσω» από την εικόνα που περιέχει το pixel.

Στην Canvas συγκεκριμένα, χρησιμοποιείται ως φόντο του καμβά ένα μοτίβο με άσπρα και γκρι τετραγωνάκια, όπως συνηθίζεται (θυμίζει σκάκι). Έτσι, εξασφαλίζεται το να είναι ευδιάκριτες οι διαφανείς περιοχές του καμβά.

2.2.2 - Move / Hide Toolbar

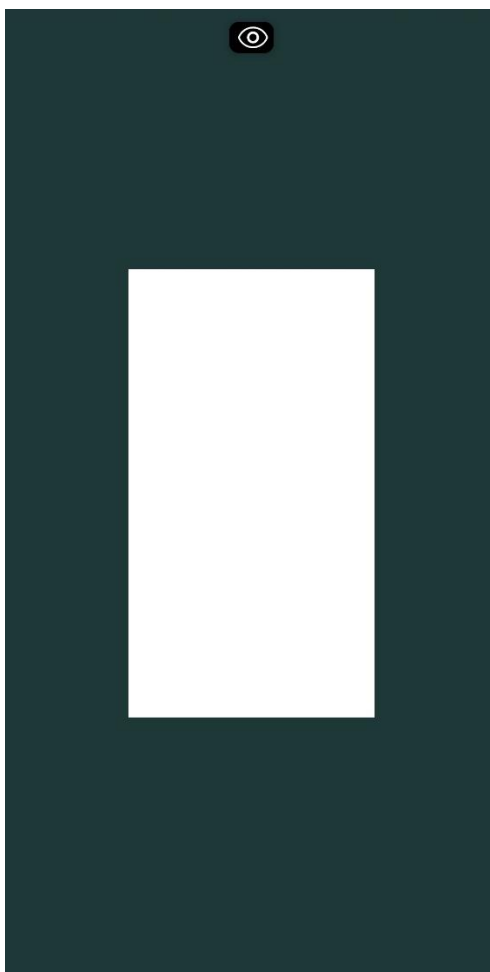
Στην δεξιά άκρη του Toolbar, υπάρχει το κουμπί μετακίνησης του. Πατώντας παρατεταμένα πάνω του, το Toolbar είναι ελεύθερο να μετακινηθεί πάνω ή κάτω, για την ενδεχόμενη διευκόλυνση του χρήστη.



*Εικόνα 2.13 – Η Γραμμή Εργαλείων, μετά από μετακίνηση της
(πριν την μετακίνηση: βλ. εικόνα 2.10).*

Επιπρόσθετα, πατώντας μια φορά σε αυτό το κουμπί, τα Toolbar και Bottom Toolbar κρύβονται από την οθόνη, ξανά για την ενδεχόμενη διευκόλυνση του χρήστη. Θεωρητικά, εφόσον αυτός έχει επιλέξει το εργαλείο και τις παραμέτρους που χρειάζεται, δεν υπάρχει ανάγκη τα toolbar να καταλαμβάνουν χώρο στην οθόνη. Το μόνο που χρειάζεται να φαίνεται είναι ο καμβάς.

Την στιγμή που θα κρυφτούν οι γραμμές εργαλείων, εμφανίζεται ένα μικρό κουμπί με την εικόνα ενός ματιού, σύμβολο που είναι καθιερωμένο παγκόσμια να υποδηλώνει την ορατότητα ενός αντικειμένου. Όταν πατηθεί λοιπόν το μάτι αυτό, τα toolbars επανέρχονται ξανά στην οθόνη. Όταν, από την άλλη, πατηθεί παρατεταμένα, μπορεί και αυτό να μετακινηθεί πάνω η κάτω, όπως και το κουμπί μετακίνησης της γραμμής εργαλείων.



Εικόνα 2.14 – Ο καμβάς, με κρυμμένες τις Γραμμές Εργαλείων

Σημειώνεται ότι υπάρχει και δεύτερος τρόπος να κρυφτούν ή να εμφανιστούν τα toolbars, απλά, πατώντας το γκρι φόντο πίσω από τον καμβά.

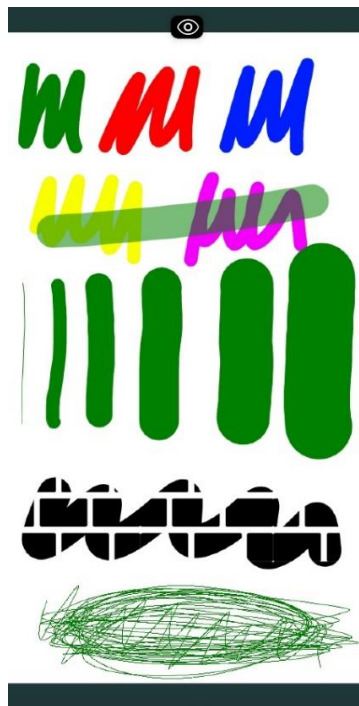
2.2.3 - Εργαλείο Brush

Πριν την επέκταση στα εργαλεία ζωγραφικής του Canvas, θα βοηθήσει μια μικρή διευκρίνιση: Όλα, εκτός της σβήστρας προφανώς, χρησιμοποιούν το ίδιο χρώμα, το οποίο υποδεικνύεται από το χρώμα των Toolbar και Bottom Toolbar.

Το πινέλο, είναι το πρώτο και πιο βασικό εργαλείο. Αποτυπώνει το μονοπάτι που διανύει το δάχτυλο πάνω στον καμβά. Όπως και κάθε άλλο εργαλείο, έχει τις δικές του ιδιότητες.

Ιδιότητες Πινέλου:

- Χρώμα: χρησιμοποιώντας τρεις μπάρες, μία για το χρώμα, μία για την φωτεινότητα και μια για την αδιαφάνεια, ο χρήστης μπορεί να καθορίσει το χρώμα της αρεσκείας του. Πάντα όμως έχει και την δυνατότητα να καθορίσει συγκεκριμένες τιμές ARGB³. Έπειτα, αφού έχει βρει το χρώμα που επιθυμεί, μπορεί να πατήσει το κουμπί «✓» με το νέο χρώμα, για να οριστικοποιήσει την επιλογή. Αξίζει να σημειωθεί ότι ο λόγος που υπάρχει αυτό το κουμπί είναι για να μπορεί ο χρήστης, παράλληλα με τον καθορισμό του νέου χρώματος, να το συγκρίνει με το παλιό.
- Μέγεθος: καθορίζει το πλάτος του πινέλου, από 1 έως και 100 pixel.
- Μοτίβο: πέρα από το κλασικό πινέλο, παρέχεται μολύβι για λεπτομέρειες επιπέδου pixel, και κάποια μοτίβα με τα οποία μπορεί να ζωγραφίσει ο χρήστης, όπως φαίνεται και στις εικόνες 2.18 και 2.19.



Εικόνα 2.15 – Επίδειξη λειτουργίας Πινέλου

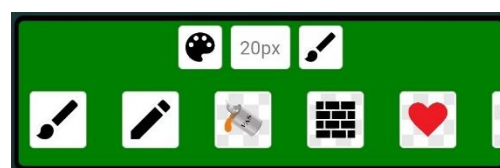
³ ARBG (Alpha, Red, Green, Blue): Μοντέλο χρώματος. Κάθε ένα από τα ARGB παίρνει τιμές από 0 έως 255. Ο συνδυασμός των RGB καθορίζει το χρώμα, και το Alpha καθορίζει την αδιαφάνεια, με το 255 να σημαίνει πλήρης αδιαφάνεια, ενώ το 0 πλήρης διαφάνεια.



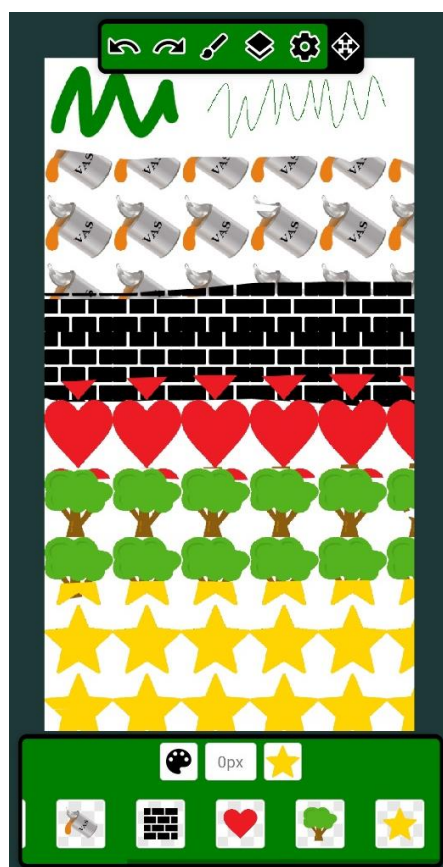
Εικόνα 2.16 – Ιδιότητες Πινέλου: Χρώμα



Εικόνα 2.17 – Ιδιότητες Πινέλου: Μέγεθος



Εικόνα 2.18 – Ιδιότητες Πινέλου: Μοτίβο



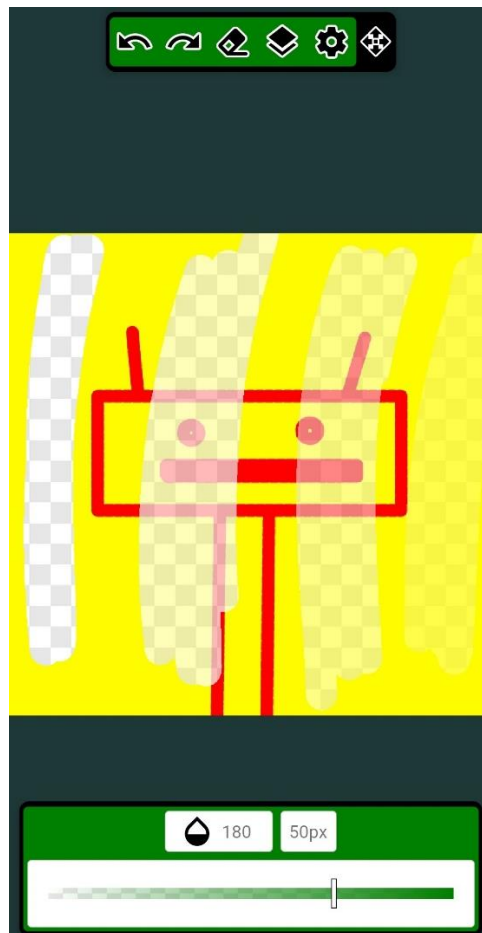
Εικόνα 2.19 – Ζωγραφική με διάφορα μοτίβα

2.2.4 - Εργαλείο Eraser

Όπως και το πινέλο, η σβήστρα αποτυπώνει το μονοπάτι του δαχτύλου στον καμβά, με την διαφορά όμως σβήνει αντί να ζωγραφίζει. Ουσιαστικά, μετατρέπει τα pixels του μονοπατιού σε διαφανή.

Ιδιότητες Σβήστρας:

- Αδιαφάνεια (Opacity): Με τιμές από 0 έως 255, καθορίζει το πόσο διαφανή κάνει τα pixels από τα οποία διέρχεται. Με 0 αδιαφάνεια, γίνονται εξολοκλήρου διαφανή, ενώ με 255 παραμένουν άθικτα.
- Μέγεθος: ομοίως με το Πινέλο.



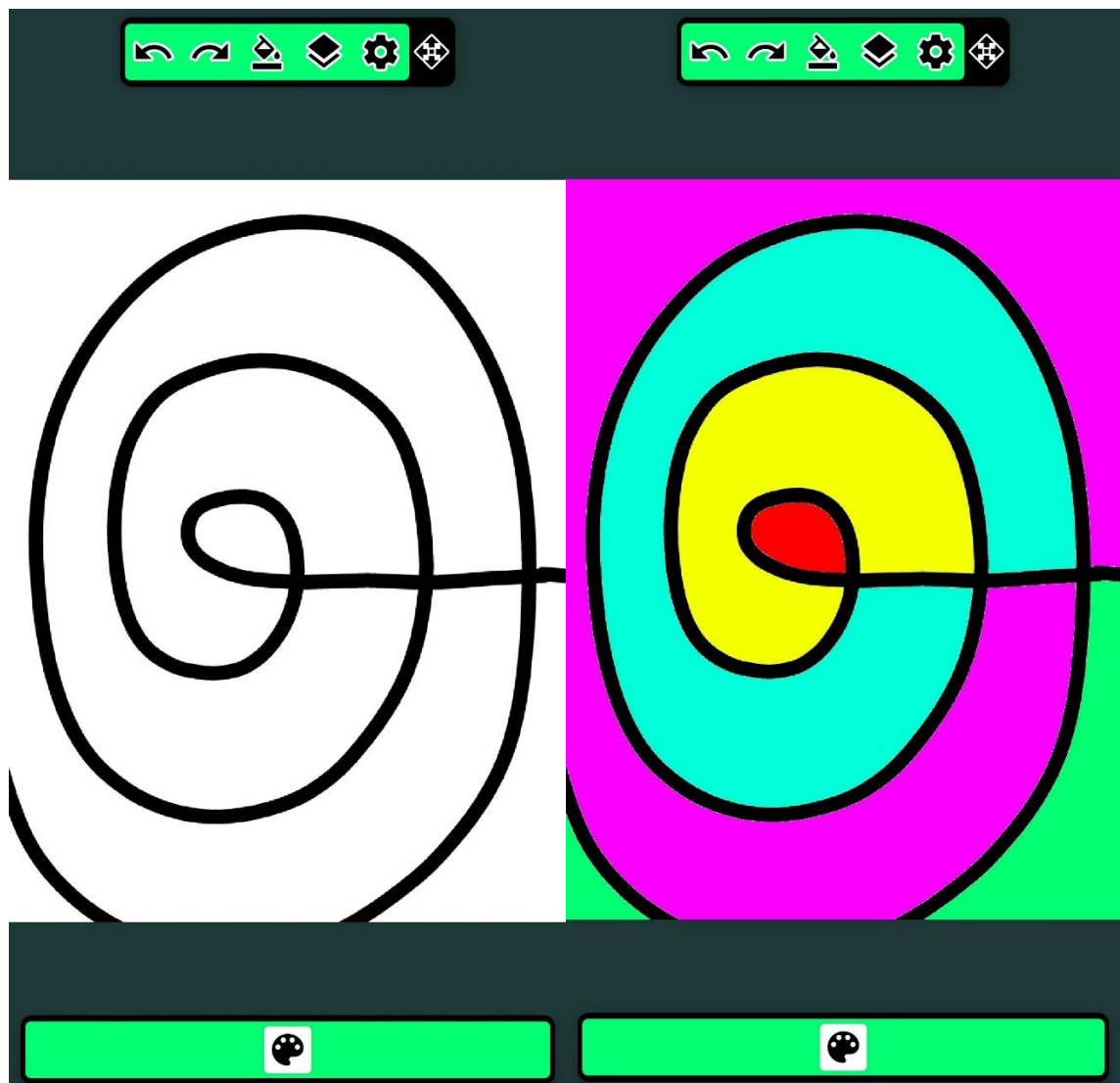
Εικόνα 2.20 – Ιδιότητες Σβήστρας και επίδειξη λειτουργίας

2.2.5 - Εργαλείο Bucket

Το εργαλείο αυτό βοηθάει τον χρήστη να χρωματίσει κλειστές περιοχές ίδιου χρώματος. Ο τρόπος για να το κάνει είναι να πατήσει σε ένα οποιοδήποτε σημείο μέσα στην επιθυμητή περιοχή.

Ιδιότητες Κουβά:

- Χρώμα: ομοίως με το Πινέλο



Εικόνα 2.21 –Ιδιότητες Κουβά και επίδειξη λειτουργίας (πριν και μετά)

2.2.6 - Εργαλείο Eyedropper

Η απλότητα του εργαλείου Eyedropper δεν αφήνει χώρο για ιδιότητες. Το μόνο που χρειάζεται να κάνει ο χρήστης είναι να πατήσει πάνω στον καμβά το σημείο με το χρώμα της αρεσκείας του.

Μπορεί, ακόμη, να σύρει το δάχτυλο του πάνω στην οθόνη. Το χρώμα θα αλλάζει ανάλογα με το ποιο pixel είναι πατημένο κάθε στιγμή.

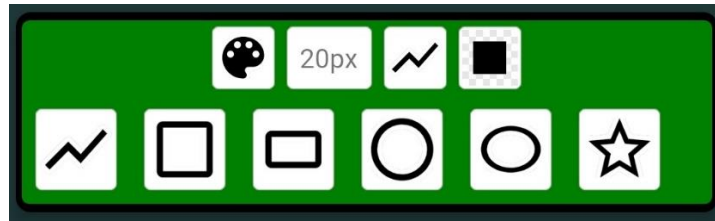
2.2.7 - Εργαλείο Shape

Ο χρήστης έχει την δυνατότητα να εισάγει γεωμετρικά σχήματα στον καμβά.

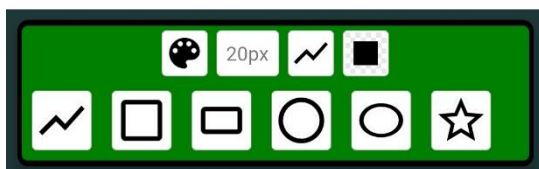
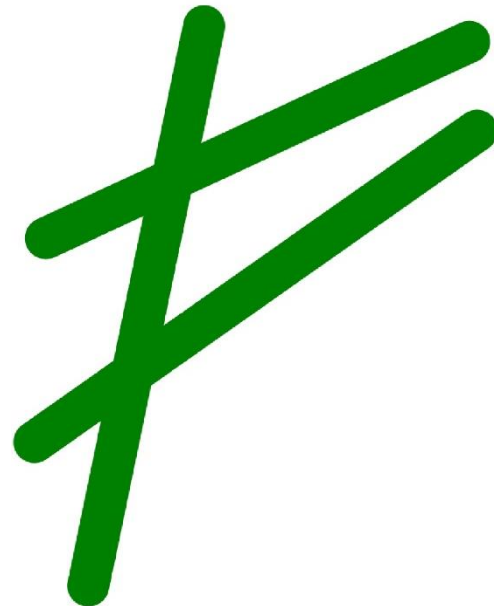
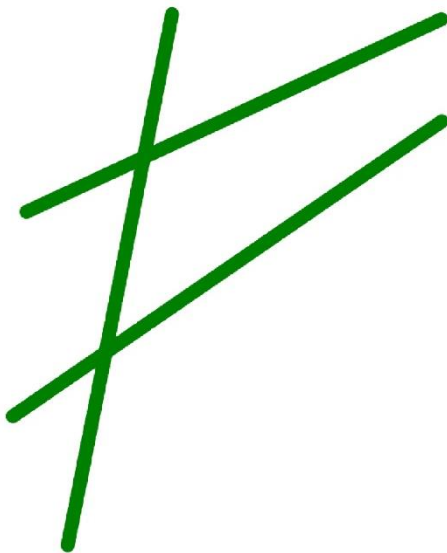
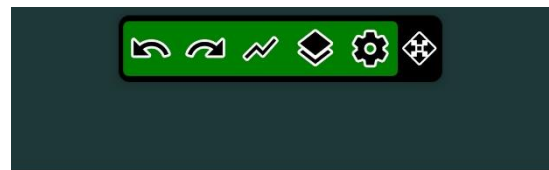
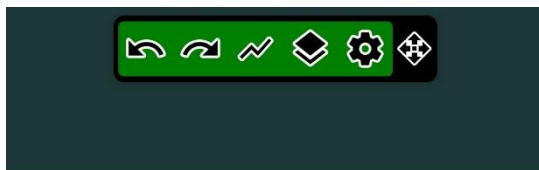
Ιδιότητες Σχήματος:

- Χρώμα: ομοίως με το Πινέλο.
- Μέγεθος: καθορίζει το πλάτος της γραμμής / περιγράμματος του σχήματος, από 1px έως και 100px.
- Τύπος σχήματος:
 - Γραμμή
 - Τετράγωνο
 - Ορθογώνιο παραλληλόγραμμο
 - Κύκλος
 - Οβάλ
 - Πολύπλευρο/Πολύγωνο⁴
- Γέμισμα: είτε περίγραμμα του σχήματος, είτε γεμάτο.

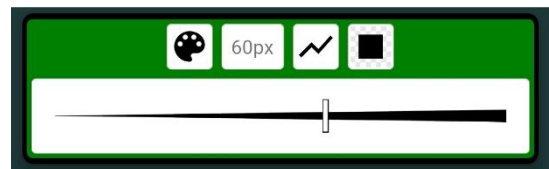
⁴ Στο πολύπλευρο/πολύγωνο, η δημιουργία του σχήματος γίνεται ζωγραφίζοντας διαδοχικές ευθείες. Η βασική διαφορά του Πολυγώνου με το εργαλείο της απλής Ευθείας, είναι ότι κάθε πλευρά ξεκινάει από το τέλος της προηγούμενης. Στην Ευθεία, από την άλλη, πρέπει πάντα να καθορίζονται οι θέσεις και των δύο άκρων της.



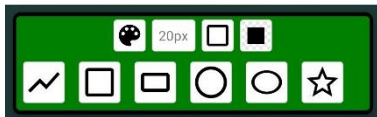
Εικόνα 2.22– Ιδιότητες Σχήματος: Τύπος Σχήματος



Εικόνα 2.23– Ευθεία, με μέγεθος 20px



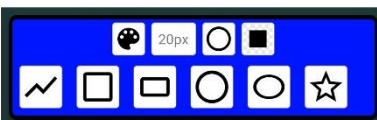
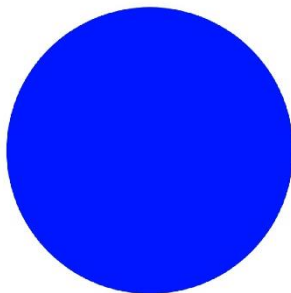
Εικόνα 2.24– Ευθεία, με μέγεθος 60px



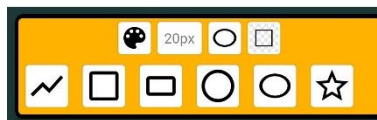
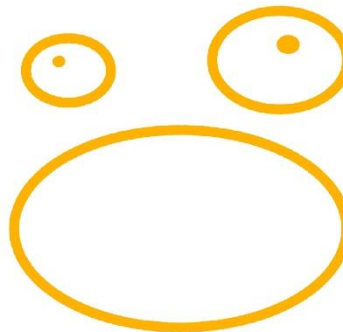
Εικόνα 2.25– Τετράγωνο



Εικόνα 2.26– Ορθ. Παραλληλόγραμμα



Εικόνα 2.27– Κύκλος



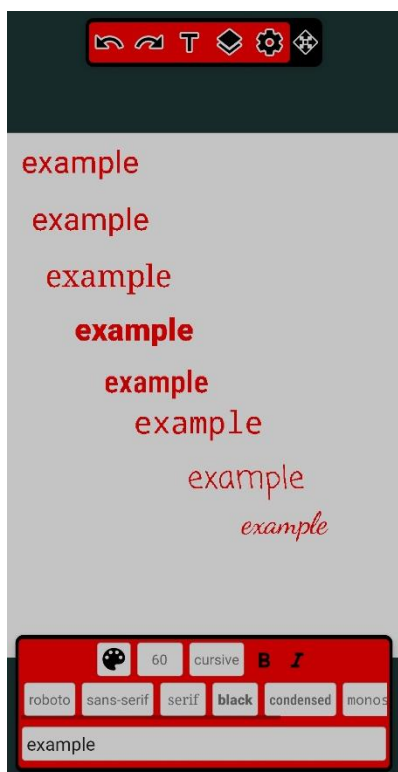
Εικόνα 2.28– Οβάλ
(με τύπο γεμίσματος «Περίγραμμα»)

2.2.8 - Εργαλείο Text

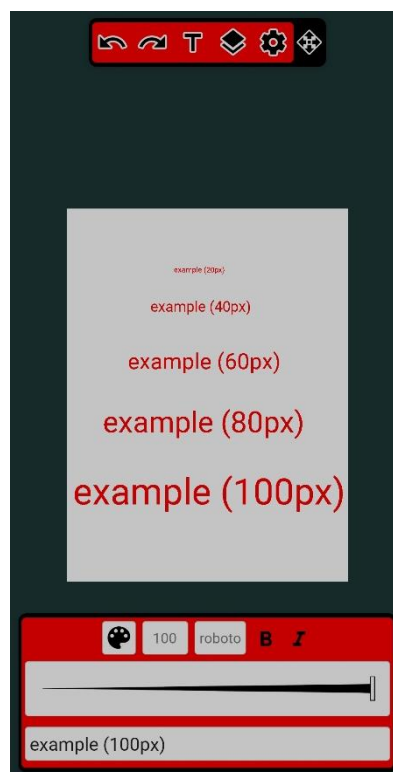
Με το εργαλείο κειμένου, είναι δυνατή η εισαγωγή μιας φράσης στον καμβά. Ο χρήστης πρέπει να εισάγει την φράση στο Bottom Toolbar και τέλος να πατήσει στον καμβά για να την γράψεις σε αυτόν, στην θέση που αφήνει το δάχτυλο του.

Ιδιότητες Κειμένου:

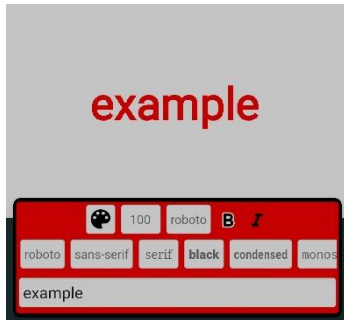
- Εισαγωγή κειμένου: Πλαίσιο το οποίο δέχεται την λέξη-φράση
- Χρώμα: ομοίως με το Πινέλο
- Μέγεθος: καθορίζει το ύψος των χαρακτήρων, από 1px έως και 100px.
- Font: παρέχονται κάποιες βασικές γραμματοσειρές
- Bold / Italics



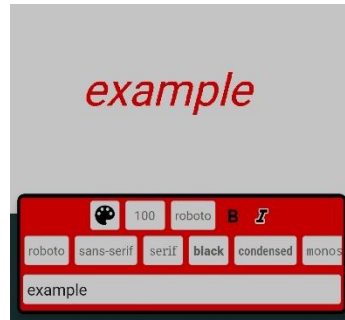
Εικόνα 2.29 – Ιδιότητα Κειμένου: Γραμματοσειρά



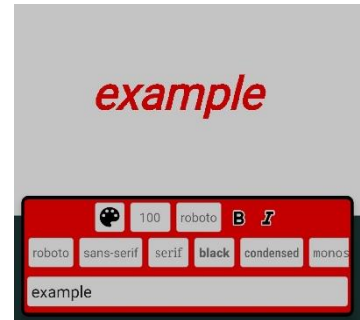
Εικόνα 2.30 – Ιδιότητα Κειμένου: Μέγεθος



Εικόνα 2.31 – Bold



Εικόνα 2.32– Italics



Εικόνα 2.33– Bold Italics

2.2.9 - Layer List

Πατώντας το κουμπί Layer List, εμφανίζεται στο Bottom Toolbar η λίστα, όπως φαίνεται στις επόμενες εικόνες. Κάθε στοιχείο της λίστας περιέχει την εικόνα, και το όνομα του layer. Η λίστα είναι οριζόντια, και η σειρά των στοιχείων υποδεικνύει την ιεραρχία τους. Με τον όρο «ιεραρχία» των layers, εννοείται ότι ο καμβάς απαρτίζεται από τα layers της λίστας, με τον εξής τρόπο:

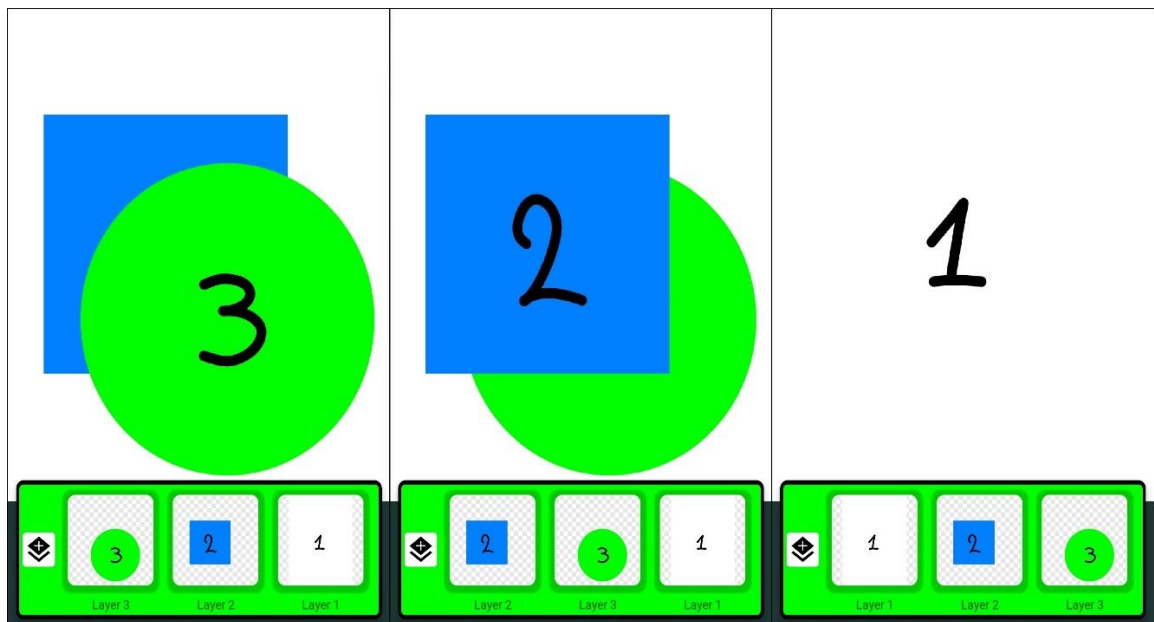
Πρώτα, ζωγραφίζεται σε αυτόν το τελευταίο (δεξιότερο) layer της λίστας. Έπειτα, ζωγραφίζεται στον καμβά το αμέσως επόμενο layer. Η διαδικασία συνεχίζεται μέχρι να ζωγραφίζεται και το layer ύψιστης ιεραρχίας, το αριστερότερο. Αυτό, θα βρίσκεται στην κορυφή του καμβά, «πάνω» από όλα τα υπόλοιπα.

Ιδιότητες λίστας στρωμάτων:

- Αριστερά από την λίστα, υπάρχει το κουμπί που δημιουργεί καινούργιο layer. Αυτό, είναι άδαιο (διαφανές), και προστίθεται στην αρχή της λίστας (εικόνα 2.34).
- Με παρατεταμένο πάτημα ενός στοιχείου της λίστας, μπορεί να συρθεί και να αλλάξει θέση στη λίστα (drag and drop).
- Επιλογή layer: Πατώντας πάνω σε ένα στοιχείο της λίστας, αυτό επιλέγεται. Όταν γίνεται αυτό, εμφανίζεται ένα άσπρο περίγραμμα γύρω από την εικόνα του στοιχείου, όπως και δυο κουμπιά: το *Layer Menu* κάτω αριστερά, και το *Visibility* κάτω δεξιά. Σημειώνεται ότι μπορούν να επιλεγούν πολλά layers μαζί.

- Κάθε πάτημα του κουμπιού *Visibility*, εμφανίζει ή κρύβει το layer από τον καμβά.
- Το *Layer Menu* ενός επιλεγμένου layer, περιέχει τα παρακάτω (εικόνα 2.37):
 - To Front: φέρνει το στοιχείο στην αρχή της λίστας, κι έπειτα ακολουθούν όλα τα υπόλοιπα επιλεγμένα στοιχεία, αν υπάρχουν, αλλάζοντας έτσι την ιεραρχία των layers.
 - Merge: Ενώνει όλα τα επιλεγμένα layers σε ένα.
 - Duplicate: Δημιουργεί αντίγραφα όλων των επιλεγμένων layers
 - Opacity: Εμφανίζεται ένας διάλογος ο οποίος δέχεται μια τιμή opacity, από 0 έως 255. Έπειτα, εφαρμόζεται αυτή σε όλα τα επιλεγμένα layers.
 - Clear: Σβήνει το περιεχόμενο όλων των επιλεγμένων layers, αφήνοντας τα άδεια (διαφανή).
 - Delete: Διαγράφει όλα τα επιλεγμένα layers.

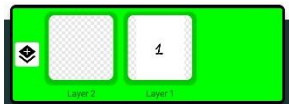
Στην παρακάτω εικόνα (2.34) φαίνεται η σημασία της ιεραρχίας των layers. Απεικονίζονται τρεις διαφορετικές διατάξεις των layers του καμβά. Το Layer 3 περιέχει ένα πράσινο κύκλο, το Layer 2 περιέχει ένα μπλε τετράγωνο, και τέλος το Layer 1 περιέχει έναν άσπρο καμβά.



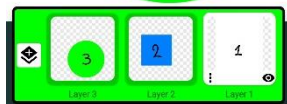
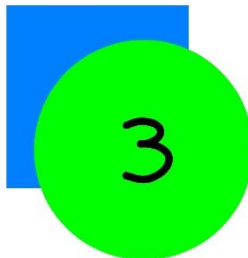
Εικόνα 2.34 – Η σημασία της ιεραρχίας στα Layers

Παραδείγματα επεξεργασίας της λίστας:

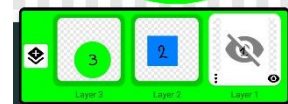
1



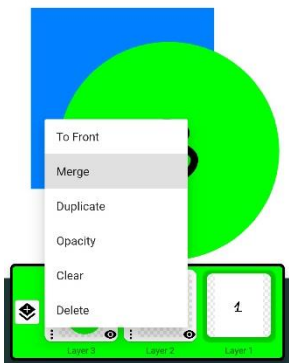
Εικόνα 2.35 – Add



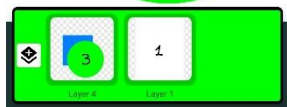
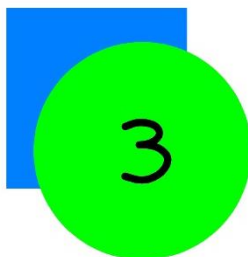
Εικόνα 2.36 – Select Layer 1



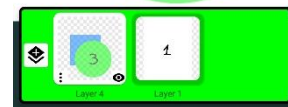
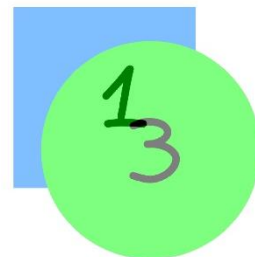
Εικόνα 2.37 – Hide Layer 1



Εικόνα 2.38 – Menu of Layer 3

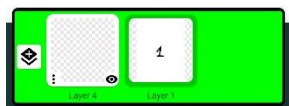


Εικόνα 2.39 – Merge 2 & 3



Εικόνα 2.40 – Change Opacity

1



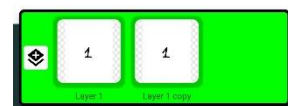
Εικόνα 2.41 – Clear Layer 4

1



Εικόνα 2.42 – Delete Layer 4

1



Εικόνα 2.43 – Duplicate Layer 1

2.2.10 - Επιλογή Transform Canvas

Στα Options του Toolbar, παρέχεται η επιλογή «Transform Canvas». Με την επιλογή της, παρέχονται οι επιλογές

- Flip Horizontally: ο καμβάς «καθρεπτίζεται» οριζόντια (εικόνα 2.45)
- Flip Vertically: ο καμβάς «καθρεπτίζεται» κάθετα (εικόνα 2.46)



Εικόνα 2.44 – Αρχική εικόνα



Εικόνα 2.45 – Οριζόντιο Flip



Εικόνα 2.46 – Κάθετο Flip

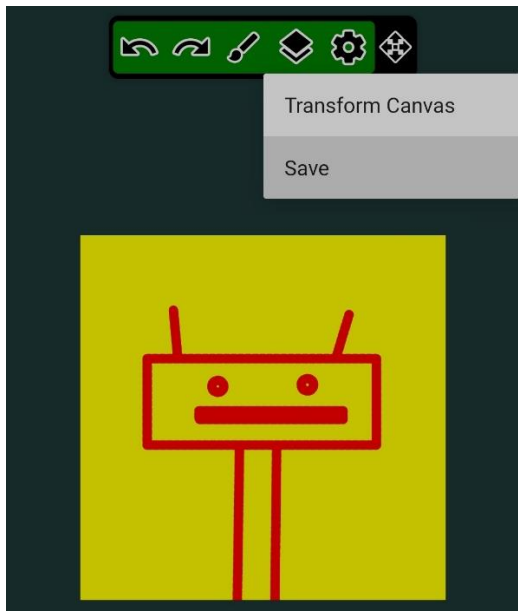
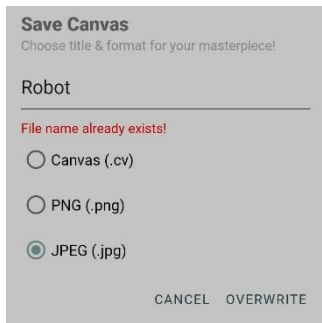
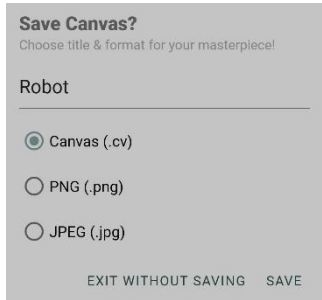
2.2.11 - Save Canvas

Η Canvas παρέχει την δυνατότητα αποθήκευσης του καμβά σε αρχείο. Υπάρχουν δυο τρόποι να επιτευχθεί αυτό:

1. Στα Options, πατώντας την επιλογή «Save».
2. Πατώντας πίσω, ενώ ο χρήστης βρίσκεται στην οθόνη με τον καμβά.

Και στις δυο παραπάνω περιπτώσεις, εμφανίζεται την οθόνη ένα παράθυρο με τον διάλογο αποθήκευσης αρχείου. Αυτός περιλαμβάνει ένα πλαίσιο για την εισαγωγή ονόματος, και παρακάτω 3 επιλογές τύπου αρχείων με τους οποίους μπορεί η εικόνα να αποθηκευτεί. Οι τύποι αρχείων είναι CANVAS, PNG και JPEG.

Σε περίπτωση που το όνομα του αρχείου (μαζί με το extension) υπάρχει ήδη, ο διάλογος ενημερώνει τον χρήστη με ένα κόκκινο μήνυμα κάτω από το όνομα. Εάν επιλέξει παρόλα αυτά να το αποθηκεύσει με ίδιο όνομα, το παλιό αρχείο αντικαταστάται από το νέο.

 <p>Εικόνα 2.47 – Επιλογή «Save» (από Toolbar -> Options)</p>	 <p>Εικόνα 2.48 – Διάλογος Επιλογής «Save»</p>  <p>Εικόνα 2.49 – Διάλογος μετά από πάτημα «Πίσω»</p>
---	--

Αφού αποθηκευτεί το αρχείο, ο χρήστης μπορεί μελλοντικά να το βρει στο *Gallery*.

2.2.12 - Undo - Redo

Η αναίρεση και επαναφορά είναι απαγορευτικό να λείπουν από μια εφαρμογή ζωγραφικής και επεξεργασίας φωτογραφιών. Στην Canvas δεν έχει μπει όριο ιστορικού, με την έννοια ότι όσες *ενέργειες* κι αν επιτελέσει ο χρήστης, πάντα μπορεί να φτάσει στην αρχική κατάσταση του καμβά με συνεχόμενα undo.

Οι *ενέργειες* οι οποίες μπορούν να αναιρούνται και επαναφέρονται είναι οι εξής:

- Ζωγραφική στον καμβά με εργαλεία:
 - Brush
 - Eraser
 - Bucket
 - Shape
 - Text
- Όλες οι αλλαγές στην λίστα των layers:
 - Αλλαγές ιεραρχίας
 - Show/Hide
 - Add
 - Merge
 - Duplicate
 - Opacity
 - Clear
 - Delete
- Horizontal flip
- Vertical flip

ΚΕΦΑΛΑΙΟ 3

ΤΟ ΛΕΙΤΟΥΡΓΙΚΟ ANDROID

Η Canvas έχει δημιουργηθεί για το λειτουργικό Android, χρησιμοποιώντας το πρόγραμμα Android Studio για τον προγραμματισμό της. Σε αυτό το κεφάλαιο θα παρουσιαστούν αυτά τα δύο εκτενώς, καθώς και κάποια εργαλεία του Android που χρησιμοποιήθηκαν από την εφαρμογή. Το κεφάλαιο κλείνει με πληροφορίες για την γλώσσα προγραμματισμού Κότλιν, με την οποία προγραμματίστηκε εξ ολοκλήρου η Canvas.

3.1 Εισαγωγή

Το Android είναι ένα λειτουργικό σύστημα για κινητά, σχεδιασμένο κυρίως για κινητές συσκευές με οθόνη αφής, όπως smartphone και tablet. Εξειδικευμένες μορφές του χρησιμοποιούνται από μια σειρά άλλων ηλεκτρονικών συσκευών, όπως κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, φορητές συσκευές αναπαραγωγής πολυμέσων, υπολογιστές και άλλα.



Εικόνα 3.1 – Το λογότυπο του Android (2019)[17]

Το Android αναπτύσσεται από μια κοινοπραξία προγραμματιστών γνωστή ως Open Handset Alliance και εμπορικά χορηγείται από την Google [18]. Παρουσιάστηκε τον Νοέμβριο του 2007, με την πρώτη εμπορική συσκευή Android, το HTC Dream, που κυκλοφόρησε τον Σεπτέμβριο του 2008 [19].

Οι περισσότερες συσκευές με Android έχουν κάποιες προ-εγκατεστημένες πρόσθετες εφαρμογές (Google Mobile Services [20]), όπως το Google Chrome, το Google Play κ.α.

Οι εφαρμογές στο Android, οι οποίες χρησιμοποιούν την μορφή APK, διανέμονται γενικά μέσω ιδιόκτητων καταστημάτων εφαρμογών όπως το Google Play Store, το Samsung Galaxy Store, το Huawei AppGallery, το Cafe Bazaar και το GetJar, ή πλατφόρμες ανοιχτού κώδικα όπως το Aptoide ή το F-Droid. [18]

Το Android είναι το λειτουργικό με τις περισσότερες πωλήσεις παγκοσμίως σε smartphone από το 2011 και σε tablet από το 2013 [18]. Από τον Μάιο του 2021, έχει πάνω από τρία δισεκατομμύρια μηνιαίους ενεργούς χρήστες, τη μεγαλύτερη εγκατεστημένη βάση οποιουδήποτε λειτουργικού συστήματος, και από τον Ιανουάριο του 2021, το Google Play Store διαθέτει περισσότερες από 3 εκατομμύρια εφαρμογές [18,21]. Η τρέχουσα σταθερή έκδοση είναι το Android 11, που κυκλοφόρησε στις 8 Σεπτεμβρίου 2020.

Μοντέλο Android εφαρμογών

Οι εφαρμογές Android μπορούν να γραφτούν χρησιμοποιώντας γλώσσες Kotlin, Java και C++. Κάθε εφαρμογή, από προεπιλογή, έχει πρόσβαση μόνο στα στοιχεία που απαιτούνται για να κάνει τη δουλειά της και όχι περισσότερα.

Τα βασικά δομικά στοιχεία μιας εφαρμογής Android, τα οποία καθορίζουν στην επικοινωνία της με το σύστημα ή με τον χρήστη, διακρίνονται σε τέσσερις διαφορετικούς τύπους:

1. Activities – επικοινωνία του χρήστη με την εφαρμογή.
2. Services – επικοινωνία με το σύστημα, με σκοπό την διατήρηση της εκτέλεσης μιας εφαρμογής στο background (πχ. Αναπαραγωγή μουσικής).
3. Broadcast receivers – επιτρέπει στο σύστημα να παραδίδει συμβάντα (πχ. Χαμηλό ποσοστό μπαταρίας) σε εφαρμογές που δεν τρέχουν απαραίτητα εκείνη τη στιγμή.
4. Content Providers – διαχειρίζεται ένα κοινόχρηστο από όλες τις εφαρμογές σύνολο δεδομένων. Αυτό μπορεί να είναι το σύστημα αρχείων, μια βάση δεδομένων SQLite, το Web ή οποιαδήποτε άλλη σταθερή τοποθεσία αποθήκευσης στην οποία μπορεί να έχει πρόσβαση μια εφαρμογή. [22]

Το δομικό στοιχείο που ενδιαφέρει για την συγκεκριμένη εργασία είναι το **Activity**.

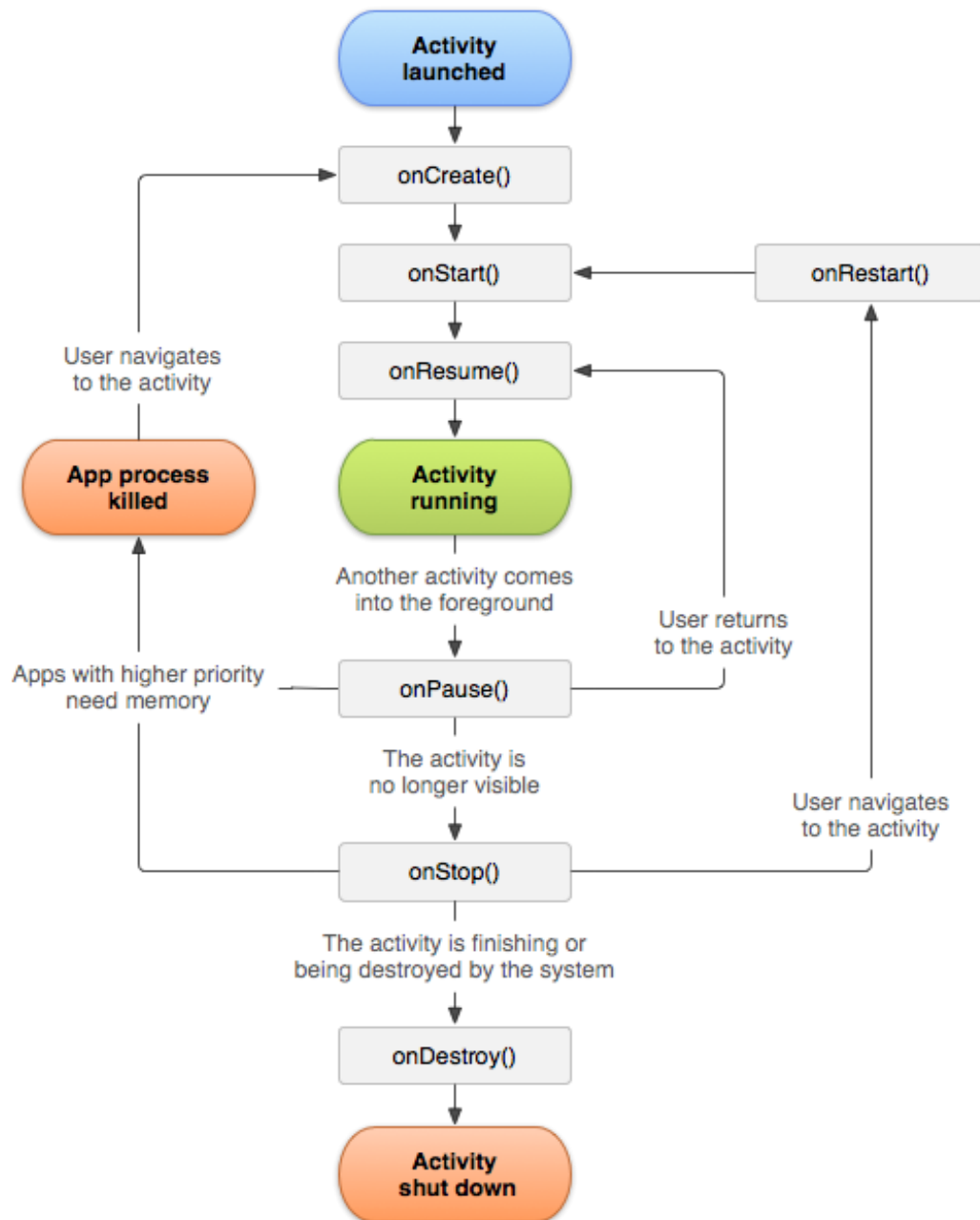
Ένα *Activity* αντιπροσωπεύει μια μεμονωμένη και εστιασμένη δραστηριότητα την οποία εκτελεί ο χρήστης. Η κλάση *Activity* πρώτον δημιουργεί ένα παράθυρο στην οθόνη το οποίο περιέχει κάθε απαραίτητο στοιχείο UI για την συγκεκριμένη δραστηριότητα, και δεύτερον, περιέχει τον κώδικα ο οποίος καθορίζει τι γίνεται και πότε. Το «πότε», θα μπορούσε για παράδειγμα να αναφέρεται στην εκκίνηση της εφαρμογής, σε ένα πάτημα κουμπιού, σε ένα scroll down, και πολλά άλλα. Για όσους είναι λίγο εξοικειωμένοι με τις έννοιες του προγραμματισμού, η κλάση *Activity* θα μπορούσε να χαρακτηριστεί και ως η συνάρτηση *main()* μιας Android εφαρμογής. Σημειώνεται βέβαια ότι μια εφαρμογή συνηθίζεται να έχει παραπάνω από μία κλάση *Activity*, όμως η λογική παραμένει η ίδια.

Τα *Activities* στο σύστημα διαχειρίζονται μέσω στοιβών (*activity stacks*). Όταν ξεκινάει ένα νέο *Activity*, συνήθως τοποθετείται στην κορυφή της τρέχουσας στοίβας και γίνεται το τρέχον *Activity* – το προηγούμενο *Activity* παραμένει πάντα κάτω από αυτό στη στοίβα και δεν θα εμφανιστεί ξανά στο προσκήνιο έως ότου εξέλθει το νέο *Activity*. Ένας τρόπος να γίνει αυτό είναι για παράδειγμα το πάτημα «Πίσω» στην συσκευή. Πάντως, μπορεί να υπάρχουν μία, αλλά και περισσότερες στοίβες ορατές στην οθόνη.

Τα *Activities* έχουν ουσιαστικά τέσσερις καταστάσεις:

1. *Running* - Εάν ένα *Activity* βρίσκεται στο πρώτο πλάνο της οθόνης (στην υψηλότερη θέση της κορυφαίας στοίβας), είναι *ενεργό ή εκτελείται*. Αυτό είναι το *Activity* με το οποίο ο χρήστης αλληλοεπιδρά αυτήν τη στιγμή.
2. *Visible* - Εάν ένα *Activity* έχει χάσει την εστίαση αλλά εξακολουθεί να παρουσιάζεται στον χρήστη, είναι *ορατό*.
3. *Hidden* - Εάν ένα *Activity* αποκρύπτεται εντελώς από ένα άλλο *Activity*, *διακόπτεται ή αποκρύπτεται*. Διατηρεί ακόμα όλες τις πληροφορίες που είχε, όμως δεν είναι πλέον ορατό στον χρήστη, άρα το παράθυρό του είναι κρυφό και συχνά θα σκοτώνεται από το σύστημα όταν χρειάζεται μνήμη αλλού.
4. *Destroyed* - Το σύστημα μπορεί να αφαιρέσει ένα *Activity* από τη μνήμη είτε ζητώντας να τελειώσει, είτε απλά σκοτώνοντας τη διεργασία (*process*) του, κάνοντάς το να *καταστραφεί*. Όταν εμφανίζεται ξανά στον χρήστη, πρέπει να επανεκκινηθεί πλήρως και να αποκατασταθεί στην προηγούμενη κατάσταση. [23]

Το παρακάτω διάγραμμα δείχνει τις σημαντικές διαδρομές κατάστασης ενός Activity. Τα τετράγωνα ορθογώνια αντιπροσωπεύουν μεθόδους που καλούνται όταν το Activity μετακινείται μεταξύ καταστάσεων. Τα χρωματιστά οβάλ είναι σημαντικές καταστάσεις στις οποίες μπορεί να βρίσκεται το Activity.



Εικόνα 3.2 – Android Activity Lifecycle [24]

3.2 To Android Studio

Το Android Studio είναι το επίσημο, ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για το λειτουργικό σύστημα Android της Google, βασισμένο στο λογισμικό IntelliJ IDEA της JetBrains και σχεδιασμένο ειδικά για ανάπτυξη Android. Είναι διαθέσιμο για λήψη σε λειτουργικά συστήματα Windows, macOS και Linux. [25]



Εικόνα 3.3 – Android Studio Logo [26]

Το Android Studio ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O. Ήταν σε στάδιο προεπισκόπησης πρώιμης πρόσβασης που ξεκινούσε από την έκδοση 0.1 τον Μάιο του 2013 και μετά μπήκε σε στάδιο beta ξεκινώντας από την έκδοση 0.8 που κυκλοφόρησε τον Ιούνιο του 2014. [27] Η πρώτη σταθερή κατασκευή κυκλοφόρησε τον Δεκέμβριο του 2014, ξεκινώντας από την έκδοση 1.0. [28] Στις 7 Μαΐου 2019, ο Kotlin αντικατέστησε την Java ως προτιμώμενη γλώσσα της Google για την ανάπτυξη εφαρμογών Android. [29] Η Java εξακολουθεί να υποστηρίζεται, όπως και η C++. [30]

Προσωπικά, θεωρώ το Android Studio ως το πιο εύχρηστο και λειτουργικό από όλα τα IDEs τα οποία έχω χρησιμοποιήσει για προγραμματισμό στην φοιτητική μου καριέρα. Λύνει τα χέρια σε πάρα πολλά θέματα όπως debugging, χρήση εξωτερικών βιβλιοθηκών⁵, οργάνωση κώδικα και αρχείων.

⁵ Βιβλιοθήκη, στην επιστήμη των υπολογιστών, είναι μια συλλογή πόρων που χρησιμοποιούνται από προγράμματα υπολογιστών, συχνά για ανάπτυξη λογισμικού. Αυτά μπορεί να περιλαμβάνουν δεδομένα διαμόρφωσης, τεκμηρίωση, δεδομένα βοήθειας, πρότυπα μηνυμάτων, προκαθορισμένο κώδικα και υπορουτίνες, κλάσεις, τιμές ή προδιαγραφές τύπου. [31]

Πιο συγκεκριμένα, σημαντικές παροχές του είναι οι εξής:

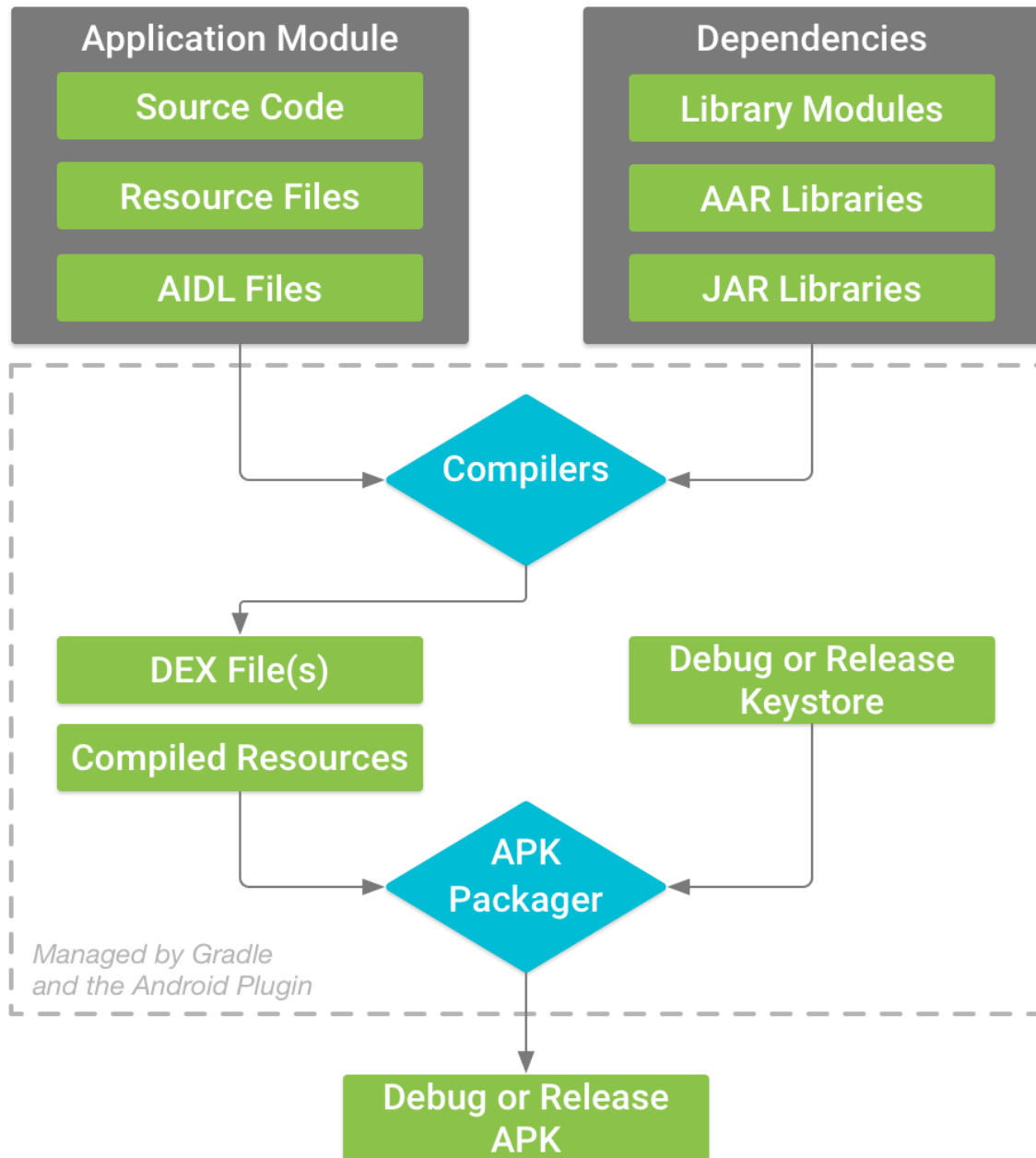
- Εργαλεία αναδιαμόρφωσης (refactoring), τα οποία συμβάλλουν στις έξυπνες και γρήγορες επιδιορθώσεις του κώδικα. Για παράδειγμα, το εργαλείο της μετονομασίας μιας μεταβλητής με τον refactor, την μετονομάζει σε κάθε σημείο που χρησιμοποιείται, ακόμα και σε διαφορετικά αρχεία κώδικα και πακέτα.
- Έξυπνη συγγραφή κώδικα: σε πολλές περιπτώσεις όπου χρειάζεται να γραφεί τυπικό κομμάτι κώδικα (που είναι πάντα ίδιο), το Android Studio παρέχει αυτά τα κομμάτια, σαν αυτόματη συμπλήρωση, ακολουθώντας τα πιο πρόσφατα πρότυπα.
- Εύκολος σχεδιασμός: παρέχονται εργαλεία για κατασκευή και προεπισκόπηση του UI (layout editor), σε πολλές διαμορφώσεις οθόνης. Ο προγραμματιστής έχει την επιλογή να σχεδιάσει το layout της εφαρμογής του είτε με κώδικα (XML), είτε με drag & drop των διάφορων UI στοιχείων στην προεπισκόπηση του.
- Σύνδεση με *GitHub* [32, 33], μια πλατφόρμα η οποία χρησιμοποιείται ευρέως από προγραμματιστές για αποθήκευση κώδικα σε cloud, και γενικότερα προσφέρει οργανωμένο και κατανεμημένο έλεγχο των εκδόσεων του. Το Android Studio επιτρέπει στον προγραμματιστή, με το πάτημα ενός κουμπιού, να ανεβάσει τον κώδικα του στην πλατφόρμα μαζί με σχόλια σχετικά με την έκδοση.



Εικόνα 3.4 – *GitHub Logo* [34]

- Χρησιμοποιεί *Gradle*, το οποίο είναι ένα εργαλείο που βοηθά στην κατασκευή (build, εικόνα 3.5), αυτοματοποίηση και παροχή καλύτερου λογισμικού, γρηγορότερα [35]. Αυτό που κάνει είναι ουσιαστικά η τακτοποίηση θεμάτων όπως η μεταγλώττιση, η ένωση, και το πακετάρισμα αρχείων κώδικα, με την χρήση του αρχείου build.gradle το οποίο τα συνδυάζει όλα αυτά σε ένα σύνολο εντολών.

- Δημιουργία APK αρχείων, τα οποία είναι ο τύπος αρχείου (.apk) ο οποίος χρησιμοποιείται από το λειτουργικό Android για διανομή και εγκατάσταση εφαρμογών [36].



Εικόνα 3.5 – The build process of a typical Android app module [37]

- Συχνές αναβαθμίσεις με τις τελευταίες τεχνολογίες και εκδόσεις Android.

- Εικονική συσκευή Android (emulator) για την γρήγορη εκτέλεση και τον εντοπισμό σφαλμάτων της εφαρμογής. Υπάρχουν πολλές επιλογές από emulator συσκευές, διαφορετικών κατασκευαστών και διαστάσεων, το οποίο βοηθάει τον προγραμματιστή να δει την συμπεριφορά της εφαρμογής του σε διαφορετικές συνθήκες και διατάξεις, και διορθώνοντας τυχόν λάθη, να εξασφαλίσει την καθολικότητα της.
- Επιθεώρηση Βάσης Δεδομένων, ο οποίος επιτρέπει την αναζήτηση και τροποποίηση της βάσης ενώ εκτελείται η εφαρμογή. Αυτό είναι ιδιαίτερα χρήσιμο για τον εντοπισμό σφαλμάτων της βάσης δεδομένων.
- Έξυπνος debugger, με δυνατότητα εκτέλεσης σε emulators πολλών διαφορετικών συσκευών, δυνατότητα παύσης (breakpoint) της ροής του προγράμματος, και παράλληλα, προεπισκόπηση τιμών των μεταβλητών και εκφράσεων.

3.3 Απαιτήσεις της εφαρμογής

Η Canvas δεν έχει υψηλές απαιτήσεις Hardware, σε αντίθεση με Software.

Software (λογισμικό), ορίζεται ως μια συλλογή οδηγιών και δεδομένων που λένε στον υπολογιστή πως να λειτουργεί. Στην επιστήμη και μηχανική των υπολογιστών, το λογισμικό είναι όλες οι πληροφορίες που επεξεργάζονται από συστήματα υπολογιστών, συμπεριλαμβανομένων προγραμμάτων και δεδομένων. Περιλαμβάνει προγράμματα υπολογιστών, βιβλιοθήκες και συναφή μη εκτελέσιμα δεδομένα, όπως ηλεκτρονική τεκμηρίωση (documentation) ή ψηφιακά μέσα. [38]

Στο project της Canvas, έχουν χρησιμοποιηθεί μια γκάμα βιβλιοθηκών, Api, πόροι (resources) όπως εικονίδια και γραμματοσειρές, και τέλος έρχεται η γλώσσα Kotlin, η οποία με μια σειρά αρχείων κώδικα έρχεται να τα ενώσει όλα για να ληφθεί το τελικό αποτέλεσμα.

Στην Canvas, πέρα από τις βασικές βιβλιοθήκες της Android, χρησιμοποιήθηκαν και κάποια έξτρα dependencies, τα οποία αναλύονται στο υπόλοιπο της ενότητας.

Σημειώνεται ότι τα Dependencies (εξαρτήσεις), είναι κάποιες εντολές οι οποίες προστίθενται μέσα στο αρχείο build.gradle (το οποίο προαναφέρθηκε στα χαρακτηριστικά του Android Studio) και καθορίζουν ποιες πρόσθετες βιβλιοθήκες πρέπει να κατέβουν έτσι ώστε να χρησιμοποιηθούν από το project.

3.3.1 - JetBrains.kotlin

Η JetBrains είναι μια εταιρεία ανάπτυξης λογισμικού που κατασκευάζει εργαλεία για προγραμματιστές και διαχειριστές project [39]. Αυτοματοποιεί κοινές, επαναλαμβανόμενες εργασίες και παρέχει εργαλεία για γρηγορότερη εξερεύνηση και εξοικείωση με τις βάσεις κώδικα. Τα προϊόντα της διευκολύνουν τον προγραμματιστή να φροντίζει για την ποιότητα σε όλα τα στάδια ανάπτυξης και να αφιερώνει λιγότερο χρόνο σε εργασίες συντήρησης. [40]



Εικόνα 3.6 – JetBrains Logo [41]

Η JetBrains έχει αναπτύξει την Kotlin, η οποία είναι η γλώσσα προγραμματισμού που χρησιμοποιήθηκε στην Canvas, και θα αναλυθεί στο τέλος του κεφαλαίου.

3.3.2 - Material Design

Ένα προσαρμόσιμο σύστημα οδηγιών, εξαρτημάτων και εργαλείων που υποστηρίζουν τις βέλτιστες πρακτικές του σχεδιασμού UI. Υποστηριζόμενο από κώδικα open-source, το Material Design βελτιώνει τη συνεργασία μεταξύ σχεδιαστών και προγραμματιστών και βοηθά τις ομάδες να δημιουργήσουν γρήγορα όμορφα προϊόντα.[42]

Στην Canvas, η βιβλιοθήκη Material Design χρησιμοποιήθηκε στην λίστα εικόνων του Gallery.



Εικόνα 3.7 – Material Design Logo [43]

3.3.3 - Google Firebase

Είναι λογισμικό ανάπτυξης εφαρμογών που υποστηρίζεται από την Google και επιτρέπει στους προγραμματιστές να αναπτύσσουν εφαρμογές iOS, Android και Web. Το Firebase παρέχει εργαλεία για την παρακολούθηση αναλυτικών στοιχείων, την αναφορά και τη διόρθωση σφαλμάτων εφαρμογών, τη δημιουργία πειραμάτων marketing και προϊόντων.
[44]



Εικόνα 3.8 – Firebase Logo [45]

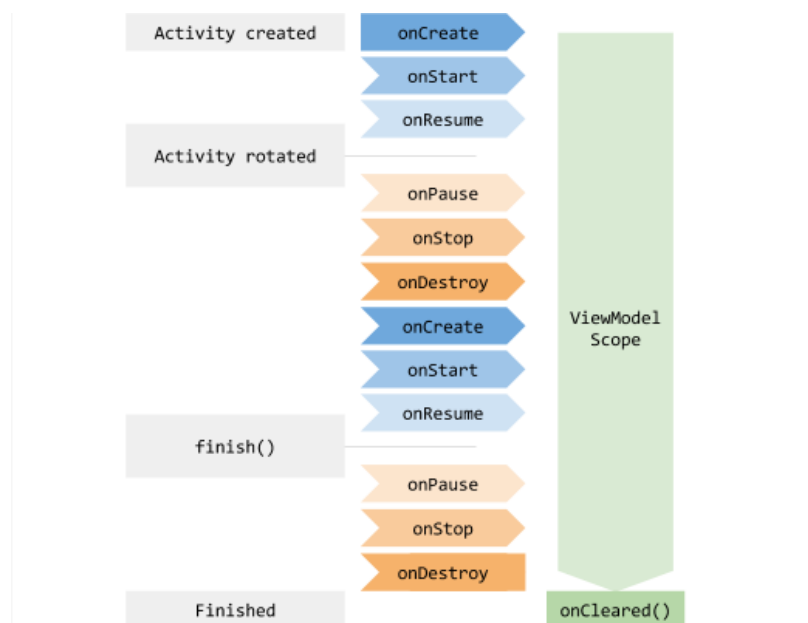
Το Firebase προσφέρει διάφορες υπηρεσίες, όπως:

- **Analytics** - Παρουσιάζει δεδομένα σχετικά με τη συμπεριφορά των χρηστών σε εφαρμογές iOS και Android, επιτρέποντας καλύτερη λήψη αποφάσεων σχετικά με τη βελτίωση της απόδοσης και του marketing εφαρμογών.[44]
 - ✓ Συμπεριλαμβάνεται στην Canvas.
- **Authentication** - Διευκολύνει τους προγραμματιστές να δημιουργήσουν ασφαλή συστήματα ελέγχου ταυτότητας.
- **Cloud messaging** - Εργαλείο ανταλλαγής μηνυμάτων πολλαπλών πλατφόρμων που επιτρέπει στις εταιρείες να λαμβάνουν και να παραδίδουν αξιόπιστα μηνύματα σε iOS, Android και Web, χωρίς κόστος.
- **Real-time database** - Βάση NoSQL που επιτρέπει την αποθήκευση και συγχρονισμό δεδομένων σε πραγματικό χρόνο μεταξύ των χρηστών.
- **Crashlytics** - Είναι ένας ρεπόρτερ σφαλμάτων σε πραγματικό χρόνο που βοηθά τους προγραμματιστές να παρακολουθούν, να δίνουν προτεραιότητα και να διορθώνουν ζητήματα σταθερότητας που μειώνουν την ποιότητα των εφαρμογών τους. [44]
 - ✓ Συμπεριλαμβάνεται στην Canvas.
- **Performance** - Η υπηρεσία παρακολούθησης απόδοσης Firebase δίνει στους προγραμματιστές μια εικόνα για τα χαρακτηριστικά απόδοσης των εφαρμογών τους iOS και Android για να τους βοηθήσει να καθορίσουν πού και πότε μπορεί να βελτιωθεί η απόδοση των εφαρμογών τους. [44]
 - ✓ Συμπεριλαμβάνεται στην Canvas.
- **Test lab** - Οι προγραμματιστές μπορούν να δοκιμάσουν τις εφαρμογές τους iOS ή Android σε διάφορες συσκευές και διαμορφώσεις συσκευών. Μπορούν να δουν τα αποτελέσματα, συμπεριλαμβανομένων βίντεο, screenshot και screen recordings, στην κονσόλα Firebase. [44]
 - ✓ Η Canvas έχει δεχτεί τα συγκεκριμένα test με σκοπό την βελτίωση της.

3.3.4 - ViewModel

Όπως αναφέρθηκε στο, εάν ένα Activity αποκρύπτεται εντελώς από ένα άλλο Activity, το παράθυρό του είναι κρυφό και συχνά θα σκοτώνεται από το σύστημα όταν χρειάζεται μνήμη αλλού. Σε αυτή τη περίπτωση όμως, και σε πολλές άλλες παρόμοιες, η διεργασία της εφαρμογής «σκοτώνεται» και η εκτέλεση του Activity σταματά. Όταν λοιπόν ο χρήστης επιλέξει να γυρίσει πίσω στην εφαρμογή, πολλά δεδομένα τα οποία είχαν παραχθεί κατά την προηγούμενη εκτέλεση, θα έχουν χαθεί. Αυτά τα δεδομένα θα μπορούσαν να είναι λίστες αντικειμένων, εικόνες, κείμενο, και πολλά ακόμη τα οποία είναι λογικό να είναι αναγκαία. Να σημειωθεί εδώ ότι με κάτι τόσο απλό όσο μια περιστροφή οθόνης, χρειάζεται να επανεκκινηθεί το τρέχων Activity, πάλι προκαλώντας το ίδιο πρόβλημα. [46]

Εδώ έρχεται η κλάση ViewModel να δώσει λύση. Αξιοποιώντας το γεγονός ότι ακόμα και αν η εφαρμογή κλείσει, η κλάση διατηρεί τα αποθηκευμένα δεδομένα της, ο προγραμματιστής μπορεί να αποφύγει το παραπάνω πρόβλημα. Η εικόνα 3.9 δείχνει τη διάρκεια ζωής του ViewModel δίπλα στον σχετικό κύκλο ζωής ενός Activity το οποίο υποβάλλεται πρώτα σε περιστροφή και έπειτα σε τερματισμό με την εντολή *finish()*.



Εικόνα 3.9 – ViewModel Lifecycle [47]

3.3.5 - Navigation UI

Το Navigation (πλοήγηση) αναφέρεται σε αλληλεπιδράσεις που επιτρέπουν στους χρήστες να περιηγηθούν στα διάφορα κομμάτια περιεχομένου μιας εφαρμογής και να επιστρέψουν από αυτά. [48]

Στην Canvas, το Navigation UI χρησιμοποιήθηκε στο Main Menu, για την πλοήγηση μεταξύ των επιλογών Gallery, New Canvas και About. Το συγκεκριμένο στοιχείο πλοήγησης ονομάζεται «Navigation Drawer», ένα κλασικό παράδειγμα στοιχείου του Navigation.

3.4 Η γλώσσα Kotlin

Η Kotlin είναι μια γλώσσα προγραμματισμού επόμενης γενιάς που αναπτύχθηκε από την JetBrains. Συνδυάζει τόσο αντικειμενοστραφή όσο και λειτουργικά χαρακτηριστικά. Επικεντρώνεται στη διαλειτουργικότητα, την ασφάλεια, τη σαφήνεια και την υποστήριξη εργαλείων. Μπορεί να χρησιμοποιηθεί σε εφαρμογές για server, desktop ή κινητών. [49]

```
fun main() {  
    println("Hello World!")  
}
```

Συνάρτηση:

```
fun max(a: Int, b: Int) = if (a > b) a else b
```

Κλάση:

```
class Person(val firstName: String, val lastName: String, var age: Int) {  
    // ...  
}
```

Τα κύρια χαρακτηριστικά που κάνουν την Kotlin ξεχωριστή: [50]

- Δεν είναι απαραίτητη η χρήση του ερωτηματικού «;» στο τέλος κάθε εντολής.

- Διαλειτουργική με Java: Ο κώδικας Kotlin μπορεί να κληθεί από κώδικα Java και αντίστροφα. Για παράδειγμα, μέσα σε μια Java κλάση μπορεί να δηλωθεί ένα αντικείμενο κλάσης Kotlin, ακριβώς όπως θα το δηλώναμε αν ήταν κλάσης Java.
- Data class: Τύπος κλάσης που απαλλάσσει τον προγραμματιστή από τους πολυάριθμους getters και setters της Java, καθώς τους δημιουργεί έμμεσα.
- Smart Cast: Ένα αντικείμενο του οποίου ο τύπος δεν έχει δηλωθεί, μπορεί να πάρει αυτόματα τον επιθυμητό τύπο, μόλις αρχικοποιηθεί.
- Null Safety: Τα αντικείμενα της Kotlin, από προεπιλογή δεν μπορούν να πάρουν την τιμή null, ώστε να αποφεύγονται τα null exceptions. Μπορούν να πάρουν null μόνο εάν κατά την δήλωση τους προστεθεί ο χαρακτήρας « ? » δίπλα στον τύπο τους.
- Named Arguments: Όταν καλείται μια συνάρτηση, ο προγραμματιστής μπορεί να ονομάσει μια παράμετρο και έπειτα να της θέσει τιμή χρησιμοποιώντας το σύμβολο του ίσον « = ». Με αυτόν τον τρόπο, ο κώδικας είναι πιο ευανάγνωστος και η σειρά των επιχειρημάτων μπορεί να αλλάξει.
- Default Arguments: Μπορεί να προκαθοριστεί η τιμή για μια παράμετρο μιας συνάρτησης, στην δήλωση της. Έτσι, εάν κληθεί αυτή η συνάρτηση χωρίς την παράμετρο, η συνάρτηση θα χρησιμοποιήσει την default τιμή της.
- Extension Function: Πρόσθεση επιπλέον λειτουργικότητας σε ένα υπάρχον στοιχείο / κλάση.
- High Order Functions: Οι High Order Functions μπορούν να δεχτούν συναρτήσεις ως παραμέτρους ή και να επιστρέφουν συναρτήσεις.
- Επιστροφή πολλαπλών τιμών: Μια συνάρτηση μπορεί να επιστρέψει πολλές τιμές μαζί και να θέσει τόσες μεταβλητές.



Εικόνα 3.10 – Kotlin Logo [51]

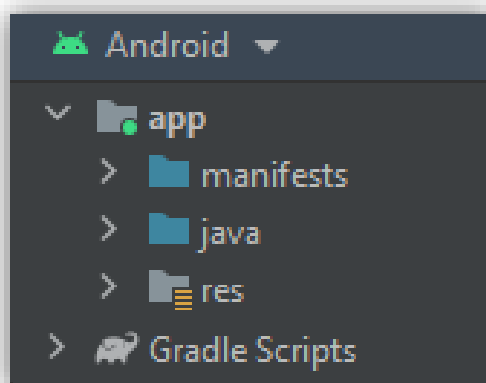
ΚΕΦΑΛΑΙΟ 4

ΛΕΠΤΟΜΕΡΗΣ ΤΕΧΝΙΚΗ ΠΕΡΙΓΡΑΦΗ

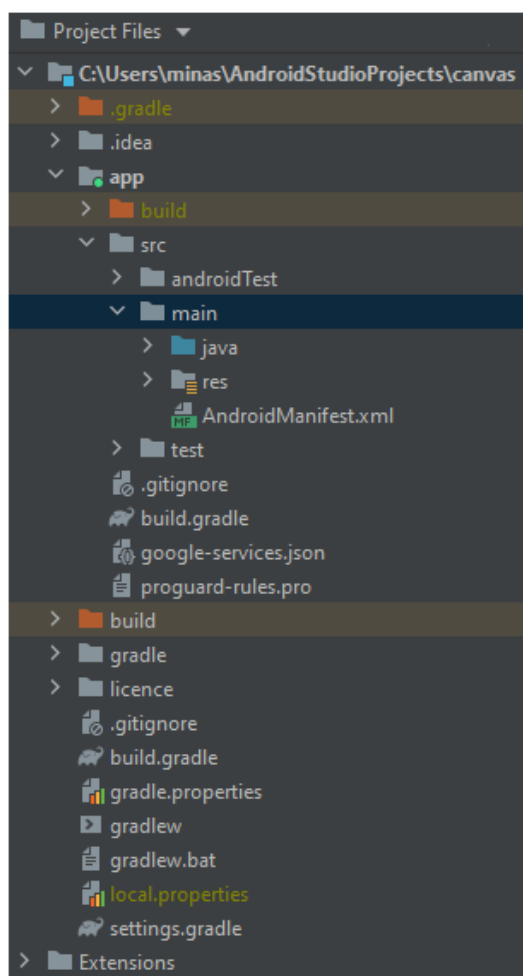
Η αρχιτεκτονική και οργάνωση του κώδικα παίζουν καθοριστικό ρόλο σε ένα προγραμματιστικό project. Είναι άλλο ένα παράδειγμα πετυχημένης εφαρμογής του «διαίρει και βασίλευε», καθώς με την τακτοποίηση των πόρων σε φακέλους –όπου στην Kotlin ονομάζονται «πακέτα» -, επιτυγχάνεται η εύκολη κατανόηση των περιεχομένων του project. Κατανόηση όχι μόνο από άλλους, αλλά και από τον ίδιο τον δημιουργό, καθώς ένας μεγάλος αριθμός αρχείων αποδεικνύεται συχνά χαοτικός χωρίς τη σωστή οργάνωση.

Στο παρόν κεφάλαιο θα αναλυθούν όλες οι κλάσεις και οι πόροι που χρησιμοποιούνται από την Canvas, καθώς και το πως όλα μαζί συνδυάζονται για να δουλέψει. Πρώτα όμως, χρειάζεται η κατανόηση της βασικής δομής ενός Android Studio project.

Κάθε project στο Android Studio περιέχει το μανιφέστο του, τον κώδικα, τα resources και τα Gradle Scripts. Στην εικόνα 4.1, φαίνεται πως αναπαριστά το Android Studio αυτά τα 4 στοιχεία (αναπαράσταση «Android»), παρόλο που στο file system, τα πράγματα είναι λίγο πιο περίπλοκα (αναπαράσταση «Project Files», εικόνα 4.2). Για ευκολία λοιπόν, οι προγραμματιστές συνήθως επιλέγουν την «Android» αναπαράσταση κατά την δημιουργία της εφαρμογής.



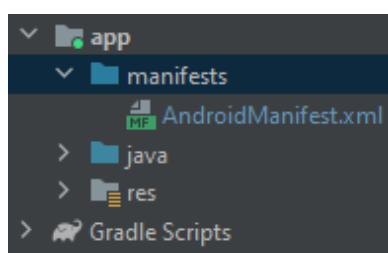
Εικόνα 4.1 – Η «Android» αναπαράσταση ενός project (ο κώδικας βρίσκεται στον φάκελο «java»).



Εικόνα 4.2 – Η «Project Files» αναπαράσταση του Canvas project.

4.1 Android Manifest

Ο φάκελος «manifests» περιέχει το αρχείο AndroidManifest.xml. Αυτό περιέχει βασικές πληροφορίες για την εφαρμογή οι οποίες χρησιμεύουν στην εκτέλεση της και στην διαχείριση της από το λειτουργικό και το Google Play.



Εικόνα 4.3 – Φάκελος manifests.

Συγκεκριμένα, το αρχείο αυτό πρέπει να περιέχει τα εξής:

1. Το όνομα πακέτου της εφαρμογής, έτσι ώστε κατά την εκτέλεση, τα εργαλεία build του Android να γνωρίζουν ακριβώς τη θέση κάθε αρχείου κώδικα. Κατά την δημιουργία του APK αρχείου, τα αρχεία Gradle αντικαθιστούν αυτό το όνομα με ένα αναγνωριστικό (application ID), το οποίο έπειτα χρησιμοποιείται ως μοναδικό αναγνωριστικό στο σύστημα και στο Google Play.

Το όνομα πακέτου της Canvas είναι το «com.mirapadakis.canvas».

2. Τα δομικά στοιχεία που χρησιμοποιούνται από την εφαρμογή, τα οποία, όπως ειπώθηκαν στην ενότητα 3.1, περιλαμβάνουν τα activities, services, broadcast receivers και content providers. Κάθε στοιχείο πρέπει να περιλαμβάνει λεπτομέρειες που περιγράφουν τον σκοπό και τον τρόπο που χρησιμοποιείται.

Για παράδειγμα, στην Canvas τα δομικά στοιχεία που έχουν δηλωθεί στο μανιφέστο είναι δυο activities (τα οποία θα παρουσιαστούν μετέπειτα), και τέλος ένας content provider ο οποίος επιτρέπει την κοινωποίηση εικόνων.

Μάλιστα, το ένα από τα δυο activity, περιλαμβάνει ειδικά φίλτρα («intent filters») τα οποία είναι απαραίτητα για την χρήση των CANVAS αρχείων.

3. Τα δικαιώματα (permissions) τα οποία χρειάζεται η εφαρμογή για να αποκτήσει πρόσβαση σε συγκεκριμένες λειτουργίες του συστήματος ή άλλων εφαρμογών. Δηλώνονται επίσης οι προϋποθέσεις υπό τις οποίες μπορεί μια άλλη εφαρμογή να αποκτήσει πρόσβαση σε περιεχόμενο από αυτήν την εφαρμογή.

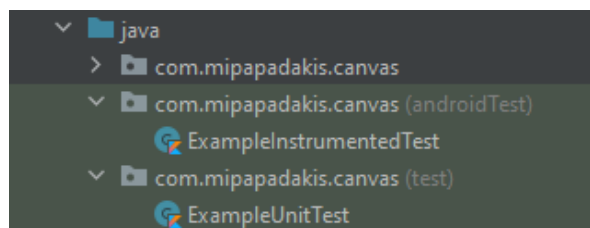
Παραδείγματος χάριν, το μανιφέστο της Canvas, δηλώνει ότι χρησιμοποιείται το δικαίωμα READ_EXTERNAL_STORAGE, το οποίο ζητείται στην περίπτωση που ο χρήστης επιθυμεί να εισάγει φωτογραφία στον καμβά.

4. Τα hardware και software χαρακτηριστικά που απαιτεί η εφαρμογή ώστε να μπορεί να χρησιμοποιηθεί. Αυτά επηρεάζουν ποιες συσκευές μπορούν να εγκαταστήσουν την εφαρμογή από το Google Play.

Η Canvas δεν έχει καμία τέτοια απαίτηση.

4.2 Πακέτα κώδικα

Όλος ο κώδικας του project, είτε σε Kotlin είτε σε Java, αποθηκεύεται μέσα στον φάκελο «java», οργανωμένος σε πακέτα. Επιπρόσθετα, σημειώνεται ότι ο φάκελος αυτός, πέρα από τα αρχεία του κώδικα περιέχει και όλα τα «test files», αρχεία τα οποία μπορούν να τρέξουν διαφορετικές εκδοχές του κώδικα, αλλά και να ακολουθήσουν μια σειρά ενεργειών πάνω στο UI της εφαρμογής ενώ αυτή τρέχει. Τα αποτελέσματα των συγκεκριμένων τεστ μπορούν να βοηθήσουν τον προγραμματιστή να εντοπίσει τυχόν θέματα που έχει η εφαρμογή όσον αφορά την απόδοση και τα λάθη («bugs») που ενδεχομένως και να προκαλούν crash. Ο προγραμματιστής μπορεί να επιλέξει να παραχθούν όλα τα αρχεία ελέγχου αυτόματα, από το Android Studio.



Εικόνα 4.4 – Περιεχόμενα φακέλου java.

Ο πρώτος υπο-φάκελος περιλαμβάνει τον κώδικα, και οι δυο επόμενοι τα test files.

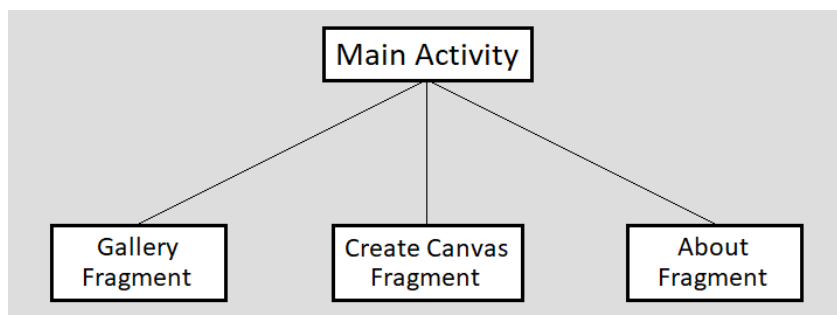
Αυτό που ενδιαφέρει όμως πέρα από τα αρχεία ελέγχου, είναι τα αρχεία κώδικα, τα οποία περιέχουν τις κλάσεις Activity και όποιες άλλες έχουν δημιουργηθεί από τον προγραμματιστή. Με την δημιουργία οποιουδήποτε νέου project με Kotlin, το Android Studio δημιουργεί από προεπιλογή το αρχείο της κλάσης MainActivity.kt, δηλαδή της κλάσης του κύριου Activity, του οποίου ο κώδικας εκτελείται μόλις ανοίξει η εφαρμογή. Μέσα στο αρχείο αυτό, όπως και σε κάθε τέτοια κλάση, φαίνεται στην πράξη ο κύκλος ζωής ενός Activity, καθώς σε αυτό υλοποιούνται οι βασικές μέθοδοι *onCreate()*, *onStart()*, *onResume()*, *onRestart()*, *onPause()*, *onStop()* και *onDestroy()*. Από αυτές, η μέθοδος η οποία περιέχει συνήθως τον περισσότερο κώδικα είναι η *onCreate()*, η οποία αρχικά αναθέτει ένα layout στο Activity. Έπειτα, είναι στην ευθύνη του προγραμματιστή να καθορίσει τι συμβαίνει όταν ο χρήστης αλληλοεπιδράσει με κάποιο στοιχείο UI του συγκεκριμένου layout. Το πιο αντιπροσωπευτικό παράδειγμα για αυτό, παρατίθεται στο παράρτημα Β.

4.2.1 - Γνωστές Κλάσεις

Πέρα από την κλάση του Activity, συνίσταται ο κώδικας να διαμοιράζεται και σε άλλες κλάσεις. Υπάρχει μεγάλος αριθμός από έτοιμες κλάσεις τις οποίες μπορεί να κληρονομήσει μια κλάση του project – δηλαδή να κληρονομήσει κάποια χαρακτηριστικά τους. Αυτές, έχουν δημιουργηθεί ώστε να παρέχουν συγκεκριμένες και εστιασμένες λειτουργίες και πολλές από αυτές σπάνια θα λείπουν από μια Android εφαρμογή.

Γνωστά παραδείγματα τέτοιων κλάσεων, οι οποίες μάλιστα χρησιμοποιούνται εκτενώς στην Canvas, είναι:

- Η προαναφερθείσα κλάση ViewModel.
- Fragment: αντιπροσωπεύει ένα επαναχρησιμοποιήσιμο τμήμα του UI της εφαρμογής. Ένα Fragment καθορίζει και διαχειρίζεται το δικό του layout, έχει τον δικό του κύκλο ζωής όπως και το Activity, και υλοποιεί μια σειρά αντίστοιχων μεθόδων όπως η *onCreate()* και *onDestroy()*. Σημειώνεται δε ότι δεν μπορεί να ζήσει από μόνο του – πρέπει να φιλοξενείται από ένα Activity ή άλλο Fragment. Στην περίπτωση της Canvas, η κλάση Fragment χρησιμοποιείται για τις επιλογές *Gallery*, *New Canvas* και *About* του Κυρίως Μενού (εικόνα 4.5).



Εικόνα 4.5 – Διάταξη των Fragment στην Canvas. [52]

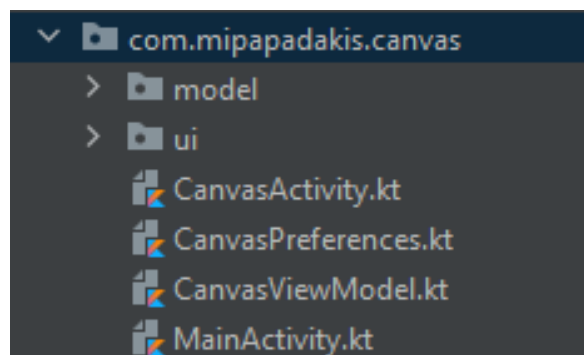
- View: Είναι η βασική κλάση για γραφικά στοιχεία, τα οποία χρησιμοποιούνται για τη δημιουργία διαδραστικών στοιχείων UI (κουμπιά, πεδία κειμένου κ.α.). Καταλαμβάνει μια ορθογώνια περιοχή στην οθόνη και είναι υπεύθυνη για τον σχεδιασμό και το χειρισμό συμβάντων.

Η υποκατηγορία (ή αλλιώς «παιδί») της κλάσης View που ενδιαφέρει την συγκεκριμένη εργασία είναι η κλάση *ImageView*, η οποία είναι υπεύθυνη για την προβολή μιας εικόνας στην οθόνη. Ουσιαστικά, ο καμβάς που βλέπει ο χρήστης στην Canvas βασίζεται σε αυτήν την κλάση, με αρκετές λειτουργικές προσθήκες οι οποίες θα επεξηγηθούν περεταίρω στην επόμενη υπο-ενότητα.

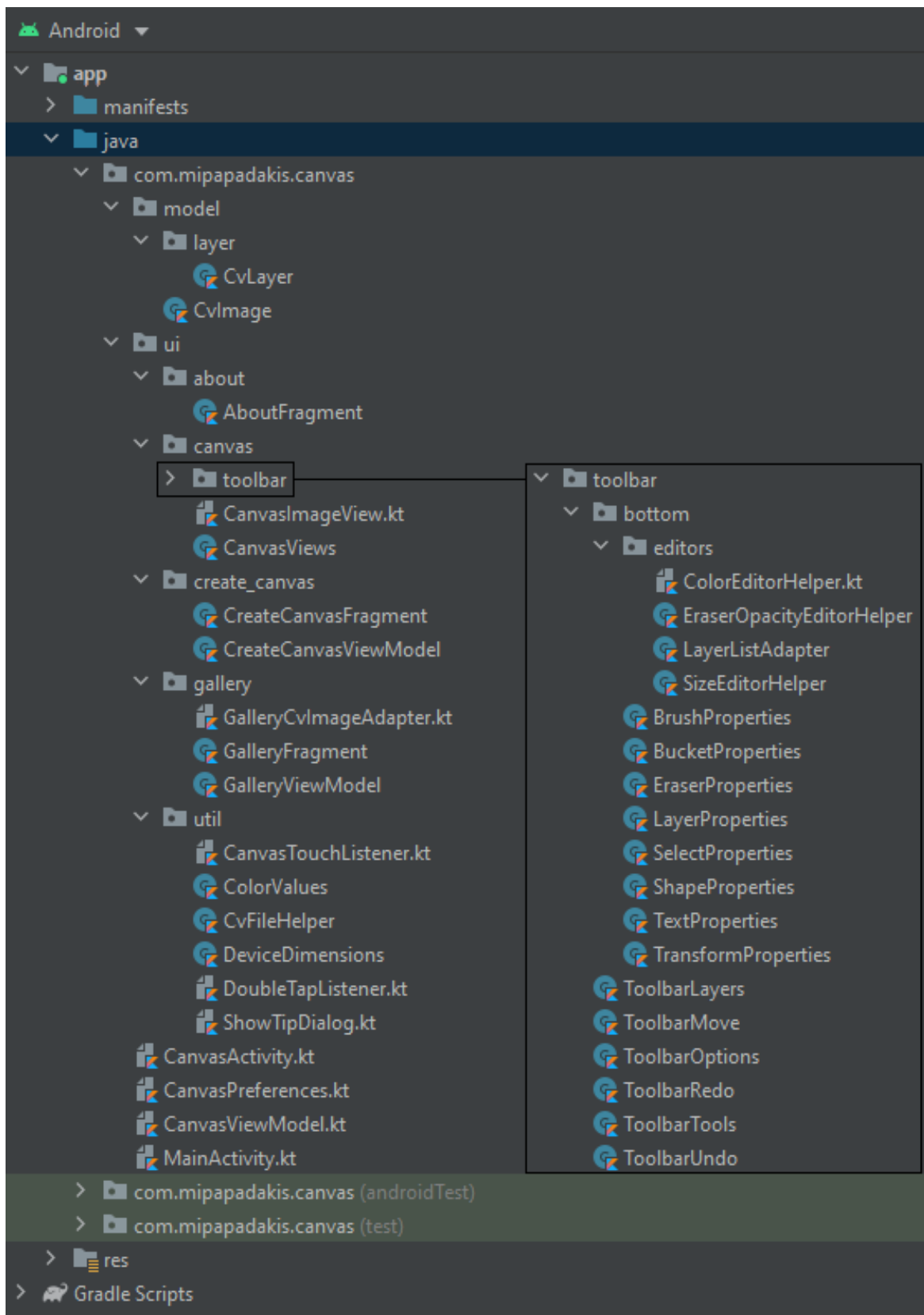
4.2.2 – Ο κώδικας της Canvas

Ο κώδικας της εφαρμογής αποτελείται από 38 κλάσεις (εικόνα 4.7), οι οποίες έχουν χωριστεί σε 3 μέρη: κλάσεις μοντέλου, κλάσεις UI, και «top-level» κλάσεις.

- Οι κλάσεις μοντέλου αναφέρονται στο μοντέλο του καμβά, δηλαδή κρατούν όλα τα δεδομένα που τον περιγράφουν, όπως ο τίτλος του, οι διαστάσεις του, και φυσικά τα layers από τα οποία αποτελείται.
- Οι κλάσεις UI καθορίζουν την συμπεριφορά κάθε στοιχείου UI της εφαρμογής.
- Οι top-level κλάσεις είναι τέσσερις: η *MainActivity.kt*, η *CanvasActivity.kt*, η *CanvasViewModel.kt* και η *CanvasPreferences.kt*. Αυτές δεν αναφέρονται μόνο στον καμβά αλλά στην εφαρμογή ως σύνολο. Ουσιαστικά «συνδέουν» όλες τις κλάσεις των δυο παραπάνω κατηγοριών, και τις τροφοδοτούν με τα κατάλληλα δεδομένα.



Εικόνα 4.6 – Ο φάκελος με τον κώδικα της Canvas.



Εικόνα 4.7 – Όλα τα αρχεία κώδικα της εφαρμογής.

MainActivity.kt

Το αρχείο MainActivity.kt περιέχει την κλάση Activity η οποία καλείται όταν ανοίγει η εφαρμογή. Σκοπός τότε είναι η δημιουργία του Κυρίως Μενού και των fragment κάθε επιλογής αυτού, και έπειτα η συλλογή δεδομένων που χρησιμεύουν για την δημιουργία του καμβά. Η συλλογή γίνεται είτε μέσω επιλογής εικόνας του GalleryFragment, είτε μέσω αρχικοποίησης καμβά στο CreateCanvasFragment. Αφού αποκτήσει τα δεδομένα του καμβά, τα τροφοδοτεί στο δεύτερο Activity της εφαρμογής, το CanvasActivity.

Όταν από την άλλη ο χρήστης επιλέξει να ανοίξει ένα οποιοδήποτε αρχείο εικόνας της συσκευής με την εφαρμογή Canvas, η MainActivity λαμβάνει το αρχείο αυτό, και το τροφοδοτεί κατευθείαν στο CanvasActivity.

CanvasActivity.kt

Η κλάση CanvasActivity δέχεται τον αρχικοποιημένο καμβά από την MainActivity και δημιουργεί το Toolbar, το Bottom Toolbar και το View το οποίο τον αναπαριστά. Το τελευταίο, έχει ονομαστεί CanvasImageView και είναι μια custom κλάση ImageView η οποία θα εξηγηθεί στο User Interface.

CanvasViewModel.kt

Η κλάση αυτή συγκρατεί σημαντικές πληροφορίες οι οποίες δημιουργούνται κατά την ζωγραφική στον καμβά, όπως ο ίδιος ο καμβάς, το χρώμα και το εργαλείο που χρησιμοποιούνται, οι επιλεγμένες ιδιότητες του εργαλείου, και το ιστορικό ενεργειών που ακολούθησε ο χρήστης, το οποίο χρησιμεύει στα *Undo* και *Redo*. Κατά την εκκίνηση του CanvasActivity, δημιουργείται το αντικείμενο της κλάσης CanvasViewModel και αρχικοποιείται κατάλληλα, ενώ ανανεώνεται ανάλογα με το user input.

Μια δευτερεύουσα χρήση της κλάσης είναι η έμμεση επικοινωνία του MainActivity και του CanvasActivity: στην περίπτωση όπου ο χρήστης επιλέξει να ανοίξει οποιαδήποτε εικόνα της συσκευής με την εφαρμογή Canvas, η MainActivity «στέλνει» την εικόνα στην CanvasActivity μέσω μιας *static* μεταβλητής της CanvasViewModel.

CanvasPreferences.kt

Η κλάση αυτή αποθηκεύει τοπικά στην συσκευή βασικές ρυθμίσεις, οι οποίες ισχύουν όχι μόνο κατά την εκτέλεση, αλλά και κάθε φορά που εκτελείται η Canvas. Για παράδειγμα, σε αυτήν αποθηκεύεται με ποιο αρχικό χρώμα θα αρχικοποιείται ένας κενός καμβάς. Επίσης, αποθηκεύονται δεδομένα τα οποία καθορίζουν εάν η εφαρμογή τρέχει για πρώτη φορά, επομένως να μην εμφανιστεί το μήνυμα «Welcome To Canvas» δεύτερη φορά.

Η κλάση που χρησιμοποιήθηκε για αυτόν τον σκοπό είναι η *SharedPreferences*, μια αρκετά διαδεδομένη κλάση για τοπική αποθήκευση δεδομένων από Android εφαρμογές.

Model

Η χρήση μοντέλου είναι διαδεδομένη για κάθε λογής εφαρμογές και προγράμματα. Γενικά, είναι μια κλάση η οποία αντιπροσωπεύει διάφορες οντότητες που χρησιμοποιούνται επαναλαμβανόμενα στις εφαρμογές αυτές, και ουσιαστικά αποθηκεύει τα βασικά δεδομένα τους. Στην περίπτωση της Canvas, υπάρχουν δυο μοντέλα, το *CvImage* και το *CvLayer*. Το πρώτο, αντιπροσωπεύει ολόκληρο τον καμβά, ενώ το δεύτερο ένα μεμονωμένο layer. Εμβαθύνοντας σε αυτά τα δυο μοντέλα:

- Το *CvLayer* καταγράφει τον τίτλο, την εικόνα, την αδιαφάνεια, τις διαστάσεις και την ορατότητα του layer που αντιπροσωπεύει.

```
class CvLayer(var title: String, private var bitmap: Bitmap){  
    private var opacity = 255  
    val width = bitmap.width  
    val height = bitmap.height  
    var visible = true  
    ...  
}
```

Εικόνα 4.8 – Η δήλωση της κλάσης *CvLayer*, με όλες τις ιδιότητες της.

- Το *CvImage* καταγράφει για τον καμβά τον τίτλο, τις διαστάσεις, και την λίστα των layers που τον απαρτίζουν.

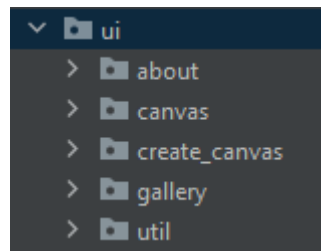
```
class CvImage(var title: String, var width: Int, var height: Int): ArrayList<CvLayer>() {...}
```

Εικόνα 4.9 – Η δήλωση της κλάσης *CvImage*, με όλες τις ιδιότητες της.

Μια κοινή λειτουργία που έχουν και οι δυο αυτές κλάσεις είναι η δυνατότητα να μετατραπούν σε Serializable κλάσεις, και το ανάποδο. Οι Serializable κλάσεις επιτρέπουν την αποθήκευση τους σαν αρχείο και την ανάκτηση τους ακριβώς όπως ήταν πριν. Επομένως, με αυτόν τον τρόπο, η Canvas μπορεί να αποθηκεύει ολόκληρα τα μοντέλα του και να τα ανακτά με ασφάλεια και ευκολία.

User Interface

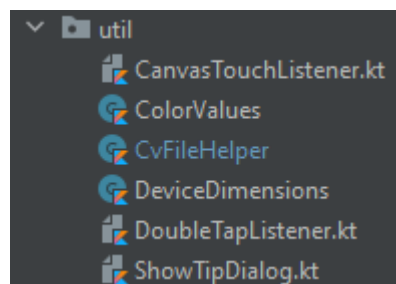
Οι περισσότερες κλάσεις του project βρίσκονται στον υπο-φάκελο *ui*. Είναι αυτές που αναλαμβάνουν την προβολή του UI της εφαρμογής, και καθορίζουν τι συμβαίνει ανάλογα το user input.



Εικόνα 4.10 – Περιεχόμενα φακέλου *ui*.

Οι φάκελοι *about*, *create_canvas* και *gallery* αφορούν το MainActivity και περιέχουν τα αντίστοιχα fragments, και ενδεχομένως κάποιες βοηθητικές κλάσεις. Οι δυο τελευταίοι, περιέχουν και τα αντίστοιχα δικά τους ViewModels για την αποθήκευση χρήσιμων δεδομένων.

Στον φάκελο *util*, υπάρχουν κάποιες κλάσεις-εργαλεία, οι οποίες χρησιμοποιούνται από τις υπόλοιπες κλάσεις του project για συγκεκριμένες λειτουργίες. Σημαντική είναι η κλάση *CvFileHelper*, υπεύθυνη για την αποθήκευση και ανάκτηση των αρχείων εικόνων.



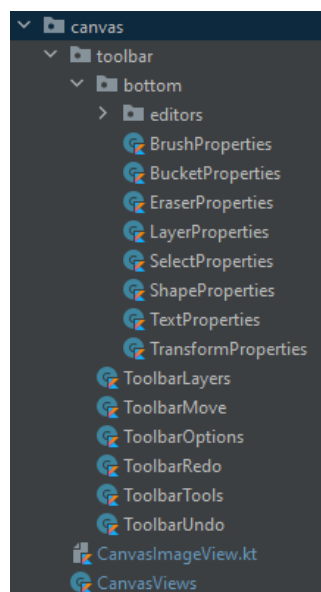
Εικόνα 4.11 – Περιεχόμενα φακέλου *ui* -> *util*.

Ο φάκελος *canvas* περιέχει κλάσεις που αφορούν το *CanvasActivity*. Περιέχει δυο πολύ σημαντικές κλάσεις: την *CanvasImageView* και την *CanvasViews*.

Η πρώτη, αντιπροσωπεύει τον καμβά, δηλαδή αναλαμβάνει να προβάλλει όλα τα *layers* του *CnImage* μοντέλου στην οθόνη. Είναι υπεύθυνη να «ακούει» τις χειρονομίες του χρήστη πάνω σε αυτήν και είτε να αλλάζει την μορφή της στην περίπτωση του αγγίγματος με δυο ή παραπάνω δάχτυλα, είτε να ζωγραφίζει στο *top-layer*, αναλόγως το επιλεγμένο εργαλείο. Δουλειά της επίσης αποτελεί να αποθηκεύει την κάθε ενέργεια που συμβαίνει στο ιστορικό του *CanvasViewModel*.

Η *CanvasViews*, από την άλλη, είναι υπεύθυνη για τα *Toolbars*. Συντονίζει την μετατόπιση του *Toolbar*, την ορατότητα του *Bottom Toolbar*, καθώς και τα περιεχόμενα αυτών. Καθορίζει τι συμβαίνει με κάθε πάτημα κουμπιού, με κάθε παρατεταμένο άγγιγμα ενός στοιχείου κ.α.

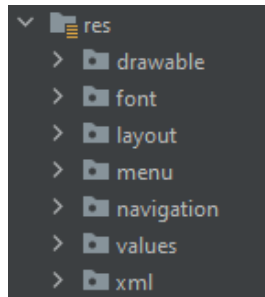
Οι λειτουργίες αυτές έχουν διαμοιραστεί σε πολλά αρχεία (εικόνα 4.12), και κάθε ένα αναλαμβάνει ένα υποσύνολο της δουλειάς. Τα αρχεία του φακέλου *toolbar* αφορούν τα κουμπιά του *toolbar*, ενώ τα αρχεία του *toolbar->bottom* αφορούν τις ιδιότητες των κουμπιών αυτών. Τέλος, ο φάκελος *toolbar->bottom->editors* περιλαμβάνει βοηθητικές κλάσεις για τον καθορισμό χρώματος, μεγέθους ή αδιαφάνειας των εργαλείων.



Εικόνα 4.12 – Περιεχόμενα φακέλου *ui -> canvas*.

4.3 Resources

Όλοι οι πόροι όπως εικονίδια, γραμματοσειρές, ορισμοί layouts και themes, βρίσκονται στον φάκελο `res`, οι υπο-φάκελοι του οποίου φαίνονται στην εικόνα 4.13:



Εικόνα 4.13 – Περιεχόμενα φακέλου `res`.

Κάθε ένας από τους υπο-φάκελους περιέχει αρχεία στατικού περιεχομένου τα οποία χρησιμοποιούνται ενδεχομένως και πάνω από μια φορά στην εφαρμογή. Αυτό αποτελεί και ένα σημαντικό λόγο ώστε να δημιουργηθεί η ανάγκη αποθήκευσης των αρχείων αυτών σε ξεχωριστό σημείο του project, ανεξάρτητο από τον κώδικα.

Ένας άλλος λόγος της ανεξαρτητοποίησης των resources από τον κώδικα της εφαρμογής είναι η εύκολη πρόσβαση σε αυτούς ως σύνολο και συμπερασματικά η εύκολη επεξεργασία τους.

Πλεονέκτημα αυτής της ανεξαρτητοποίησης είναι επίσης ότι το σύστημα του Android μπορεί να επιλέξει διαφορετικά resource αρχεία αναλόγως την διαμόρφωση της συσκευής, π.χ. εάν βρίσκεται σε κάθετη ή οριζόντια θέση, εάν είναι σε dark mode κ.α. Ο προγραμματιστής μπορεί να ορίσει συγκεκριμένα αρχεία για κάθε τέτοια διαμόρφωση, εξασφαλίζοντας την καθολικότητα και την αποφυγή λαθών σε διαφορετικές συσκευές και περιπτώσεις.

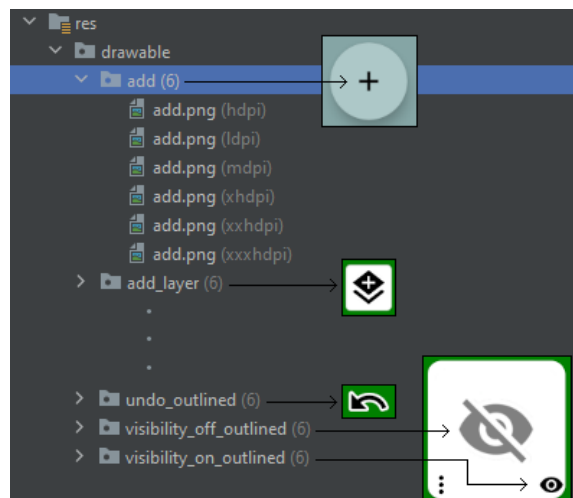
Όσον αφορά την πρόσβαση των αρχείων αυτών από τον κώδικα της εφαρμογής, επιτυγχάνεται μέσω αναγνωριστικών *id*, τα οποία ορίζονται ξεχωριστά για κάθε resource στοιχείο και καταγράφονται στην κλάση *R* – η οποία δημιουργείται αυτόματα.

Στην συνέχεια ακολουθεί η επεξήγηση κάθε υπο-φάκελου του `res`.

4.3.1 - drawable

Ένα drawable αναπαριστά μια εικόνα η οποία μπορεί να εμφανιστεί στην εφαρμογή. Μπορεί να εφαρμοστεί κατευθείαν σε ένα άλλο resource ή να ανακτηθεί από τον κώδικα με API όπως το *getDrawable(id)*.

Για κάθε drawable, μπορούν να υπάρχουν πολλά αρχεία εικόνας ίδιου περιεχομένου αλλά διαφορετικού μεγέθους ή ποιότητας (εικόνα 4.14: «add»), έτσι ώστε το σύστημα να λαμβάνει αυτόματα όποια από αυτές το βολεύει σε κάθε περίπτωση.

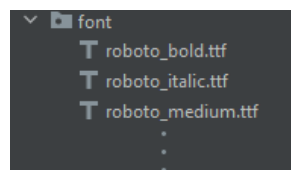


Εικόνα 4.14 – Περιεχόμενα φακέλου drawable.

Στην Canvas, χρησιμοποιούνται τουλάχιστον 70 drawables, τα οποία είτε δημιουργήθηκαν, είτε αποκτήθηκαν από γνωστό ιστότοπο της Google ο οποίος τα παρέχει δωρεάν. [53]

4.3.2 - font

Περιέχει όλες τις γραμματοσειρές πέρα από αυτές που παρέχονται από το Android Studio. Και αυτές, αποκτήθηκαν από γνωστό ιστότοπο της Google δωρεάν. [54]

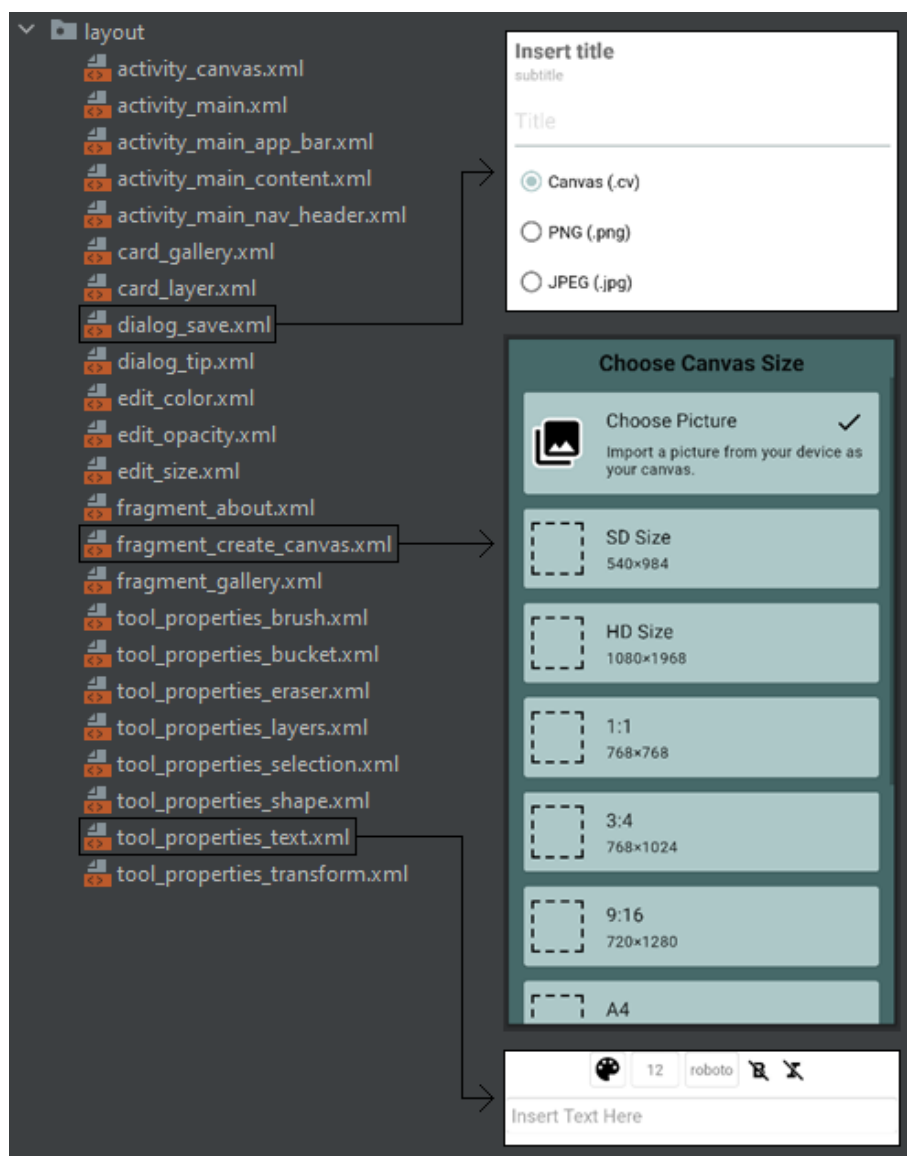


Εικόνα 4.15 – Περιεχόμενα φακέλου font.

4.3.3 - layout

Η κλάση *ViewGroup*, υποκατηγορία της κλάσης *View*, είναι η βασική κλάση για τα *Layouts* – ή αλλιώς, διαμορφώσεις οθόνης – τα οποία ουσιαστικά είναι αόρατα «κοντέινερ» που συγκρατούν άλλα *Views* ή *ViewGroups*, και ορίζουν τις ιδιότητες διάταξης αυτών. Κάθε αρχείο layout βρίσκεται στον φάκελο *res -> layout* και καθορίζει τα χαρακτηριστικά των στοιχείων που περιλαμβάνει, χρησιμοποιώντας κώδικα XML (βλ. παράρτημα Γ).

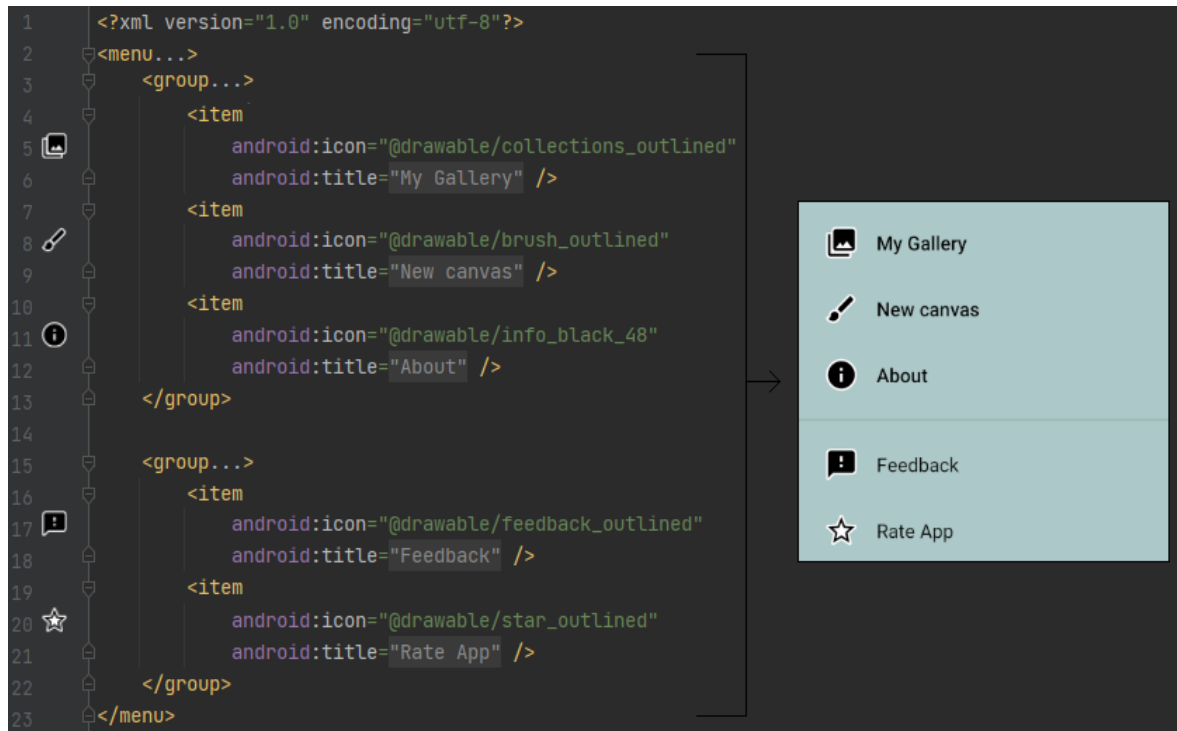
Στην *Canvas* χρησιμοποιούνται πάνω από 20 layouts. Αυτά είναι φτιαγμένα είτε για να καλύπτουν ολόκληρη την οθόνη της συσκευής, είτε ένα μέρος αυτής.



Εικόνα 4.16 – Περιεχόμενα φακέλου layers, και προεπισκόπηση κάποιων από αυτά.

4.3.4 - menu

Κάθε αρχείο μενού ορίζει με κώδικα XML ένα μενού της εφαρμογής, δηλαδή τις επιλογές που περιέχει, τις πιθανές ομαδοποιήσεις αυτών, το εικονίδιο κάθε επιλογής κ.α. Παράδειγμα τέτοιου μενού αποτελεί το Κυρίως Μενού της Canvas.



Εικόνα 4.17 – Κώδικας και προεπισκόπηση ενός αρχείου μενού.

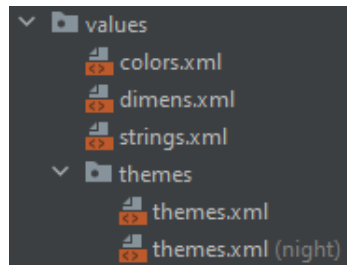
4.3.5 - navigation

Περιλαμβάνει αρχεία XML με εντολές ή γραφήματα πλοήγησης μεταξύ διαφορετικών περιοχών της εφαρμογής. Αντιπροσωπεύει όλες τις «διαδρομές» που μπορεί να ακολουθήσει ένας χρήστης με μια σειρά από ενέργειες, και ορίζει ποιος θα είναι ο προορισμός.

Για παράδειγμα στην Canvas, οι επιλογές *Gallery*, *New Canvas* και *About* του Κυρίως Μενού χρησιμοποιούν ένα navigation αρχείο ώστε να διευκρινιστεί σε ποιο Fragment θα οδηγηθεί ο χρήστης μετά από την πραγματοποίηση μιας επιλογής στο μενού.

4.3.6 - values

Ο φάκελος αυτός περιέχει συλλογές από τιμές πολλών ειδών, όπως κωδικούς χρωμάτων, διαστάσεις στοιχείων, strings, themes κ.α. σε κώδικα XML.



Εικόνα 4.18 – Περιεχόμενα φακέλου values.

Όλες αυτές οι τιμές μπορούν να χρησιμοποιηθούν σε κώδικα ή σε άλλα resource αρχεία (μέσω αναγνωριστικών id), επιτρέποντας έτσι στην ομοιομορφία και οργάνωση του project. Η επεξεργασία ενός value, θα εφαρμοστεί σε κάθε σημείο που αυτό εμφανίζεται, χωρίς να χρειάζεται ο προγραμματιστής να αλλάζει κάθε ένα από αυτά τα σημεία ξεχωριστά.

Μεταξύ άλλων, τα συνηθέστερα αρχεία values είναι τα εξής:

- **colors.xml**: Συλλογή από χρώματα σε δεκαεξαδικό σύστημα. Το Android Studio παρέχει και color editor, για την δημιουργία custom χρωμάτων.
- **dimens.xml**: Συλλογή από διαστάσεις, συνήθως για τα διάφορα στοιχεία που ορίζονται στα layouts της εφαρμογής.
- **strings.xml**: Συλλογή από όλα τα strings – δηλαδή λέξεις, φράσεις ή κείμενα – τα οποία εμφανίζονται στην εφαρμογή.

Μια αξιοσημείωτη χρήση του *strings.xml* είναι ότι αποθηκεύοντας σε αυτό όλα αυτά τα strings, διευκολύνεται η μετάφραση της εφαρμογής σε άλλες γλώσσες, καθώς ο μεταφραστής θα χρειαστεί να αλλάξει μόνο αυτό το αρχείο. Το γεγονός όμως ότι το αρχείο καταγράφει όλα αυτά τα strings με μεθοδικό τρόπο, επιτρέπει την μετάφραση όχι μόνο από άνθρωπο, αλλά ενδεχομένως και από λογισμικό.

- **themes**: Περιλαμβάνει XML αρχεία τα οποία καθορίζουν χρώματα και συμπεριφορές του UI σε διαφορετικές πλατφόρμες, συνθήκες ή διαμορφώσεις.

Για παράδειγμα, συχνά υπάρχουν τουλάχιστον δυο θέματα, ένα για Light και ένα για Dark Mode. Όταν όμως ο προγραμματιστής επιθυμεί να χρησιμοποιήσει στο project μια λειτουργία ή εργαλείο το οποίο δεν υπήρχε σε παλιότερες εκδόσεις Android, μπορεί να φτιάξει ειδικό theme για τα παλιότερα versions και ειδικό για τα νεότερα, το οποίο να χρησιμοποιεί αυτή την λειτουργία.

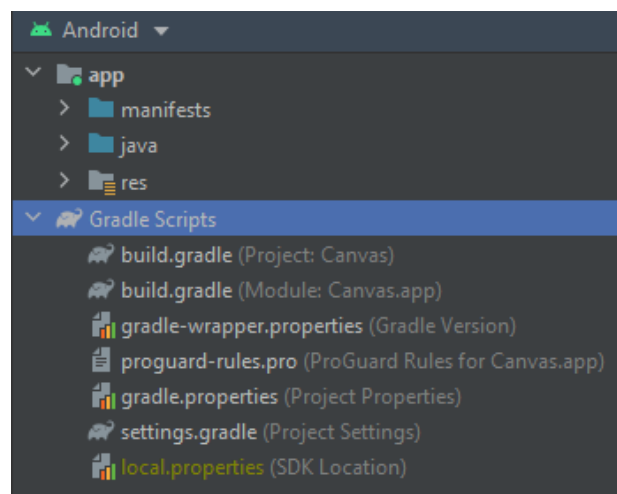
4.3.7 - xml

Αποθήκευση αυθαίρετων αρχείων XML για οποιαδήποτε χρήση. Στην Canvas, χρησιμοποιείται ένα τέτοιο αρχείο για την παροχή του path της εφαρμογής στον Content Provider ο οποίος αναλαμβάνει την κοινοποίηση των εικόνων.

4.4 Gradle Scripts

Κάθε φορά που δημιουργείται ένα project στο Android Studio, δημιουργούνται αυτόματα όλα τα απαραίτητα αρχεία Gradle. Αυτά τα αρχεία αλληλοεπιδρούν με την επέκταση Android Gradle και καθορίζουν την δημιουργία (build) της εφαρμογής.

Η επέκταση στις χρήσεις όλων αυτών τα αρχείων ξεφεύγει από τους σκοπούς της εργασίας, εκτός των δυο πρώτων: `build.gradle(project)` και `build.gradle(module)`, τα οποία είναι τα μόνα που συνήθως θα χρειαστεί να τροποποιηθούν από τον προγραμματιστή.



Εικόνα 4.19 – Gradle Scripts.

Πριν την επεξήγηση αυτών των δυο script, χρειάζεται αν διευκρινιστεί ότι τα project του Android Studio αποτελούνται από ένα ή και περισσότερα *modules*. Το καθένα από αυτά, είναι διαφορετικές «εκδοχές» της εφαρμογής οι οποίες μπορούν να κάνουν build, να τρέξουν, και να επιδιορθωθούν ξεχωριστά η μία από την άλλη. [55] Για αυτόν τον λόγο, το Android Studio δημιουργεί αυτούς τους δυο διαφορετικούς τύπους build.gradle αρχείων:

- build.gradle(project): Γνωστό και ως top-level build αρχείο, περιέχει επιλογές διαμόρφωσης οι οποίες είναι κοινές για όλα τα modules του project. [55]
- build.gradle(module): Γνωστό και ως module-level build αρχείο, διαθέτει ρυθμίσεις για την κατασκευή ενός συγκεκριμένου module. Σε αυτό το αρχείο καταγράφονται σημαντικές πληροφορίες και ρυθμίσεις, όπως:
 - Το όνομα πακέτου της εφαρμογής.
 - Η έκδοση της εφαρμογής.
 - Οι εκδόσεις Android στις οποίες μπορεί να λειτουργήσει η εφαρμογή.
 - Plugins, τα οποία αντιπροσωπεύουν ένα σύνολο από διεργασίες που πρέπει να γίνουν κατά την κατασκευή του project. Αυτές μπορεί να είναι μεταγλώττιση κλάσεων, η δημοσίευση ορισμένων αρχείων κ.α. [56]
 - Εξαρτήσεις) της εφαρμογής, οι οποίες θα μπορούσαν να χαρακτηριστούν και βιβλιοθήκες, και χρησιμεύουν κατά την μεταγλώττιση του κώδικα της.

Τα υπόλοιπα gradle script πέρα από τα δυο αρχεία build.gradle, τις περισσότερες φορές δεν θα χρειαστεί να τροποποιηθούν από τον προγραμματιστή, καθώς ενημερώνονται αυτόματα όταν γίνονται οποιεσδήποτε σχετικές αλλαγές στο project. [55]

ΚΕΦΑΛΑΙΟ 5

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Το παρόν κεφάλαιο είναι το τελευταίο και αναλύει γενικά συμπεράσματα και πιθανές μελλοντικές επεκτάσεις της Canvas.

5.1 Συμπεράσματα

Η εφαρμογή της διπλωματικής αυτής, χρησιμοποιεί τις τελευταίες τεχνολογίες για mobile programming, οι οποίες παρεμπιπτόντως εξελίσσονται κάθε μέρα. Σκοπός λοιπόν της εργασίας δεν είναι τόσο η επίδειξη πρότυπων τεχνικών λύσεων για την δημιουργία μιας εφαρμογής αυτής της κατηγορίας, όσο η παροχή ενός γενικότερου σκεπτικού. Ότι δηλαδή ένας προγραμματιστής θα πρέπει να αξιοποιεί τα πιο πρόσφατα εργαλεία με κατάλληλο τρόπο ώστε να επιτευχθεί η καλύτερη απόδοση της εφαρμογής, αλλά και μια πιο ευχάριστη εμπειρία για τον χρήστη. Εξάλλου, για αυτόν ακριβώς τον λόγο εξελίσσονται και αυτά τα εργαλεία, από άτομα και εταιρείες με μακράν μεγαλύτερη εμπειρία πάνω σε αυτό το θέμα.

Προσωπικά, επιτεύχθηκε μεγάλη εξέλιξη μέσω αυτής της εργασίας, τόσο σε γνώσεις και εμπειρία όσο και στην εφαρμογή του παραπάνω σκεπτικού. Είναι μια εργασία ιδιαίτερα εκτενής προγραμματιστικά, η οποία χρησιμοποιεί ένα μεγάλο εύρος εργαλείων, τα οποία μπορούν να φανούν χρήσιμα για κάθε τύπου εφαρμογή. Αυτός ήταν άλλωστε και ένας από τους λόγους επιλογής του συγκεκριμένου θέματος. Φυσικά, οι αποκτημένες γνώσεις δύνανται να θέσουν τα θεμέλια για την μελλοντική δημιουργία άλλων, διαφορετικών εφαρμογών, εμπορικών ή μη.

5.2 Μελλοντικές επεκτάσεις

Η εφαρμογή της διπλωματικής αυτής, παρέχει πολλά βασικά εργαλεία για ζωγραφική, αλλά υπάρχουν μεγάλα περιθώρια για εξέλιξη. Οι δυσκολίες που συναντήθηκαν κατά τον προγραμματισμό της Canvas δεν επέμειναν για πάνω από δυο με τρεις μέρες πριν την επίλυση τους. Παρόλα αυτά, ήταν μια γενικά χρονοβόρα διαδικασία, με αποτέλεσμα να ήταν ανέφικτη η παροχή παραπάνω λειτουργιών μέσα στο χρονικό περιθώριο του ενός εξαμήνου. Και πάλι, όμως, αρκετοί αρχικοί στόχοι όχι μόνο επιτεύχθηκαν αλλά και ξεπεράστηκαν.

5.2.1 - Bugs

Τα σημεία τα οποία χρίζουν κάποιας βελτίωσης στην εφαρμογή είναι τα εξής:

- Η εισαγωγή μεγάλης εικόνας ως φόντο στον καμβά, πολλές φορές έχει αποτέλεσμα να κολλάει η εφαρμογή.
- Ο αριθμός των layers όπου μπορεί να δημιουργήσει ο χρήστης μπορεί να προκαλέσει crash λόγω έλλειψης μνήμης εάν γίνει πολύ μεγάλος. Γι' αυτόν τον λόγο, έχει τεθεί όριο των 15 layers ως προσωρινή λύση.
- Το ιστορικό ενεργειών για το Undo και Redo δεν έχει προς το παρόν κάποιο όριο. Εάν λοιπόν προστεθεί πολύ μεγάλος αριθμός από ενέργειες, είναι πιθανό να μην χωράνε στην μνήμη και να προκληθεί crash.

Σημειώνεται ότι ο τρόπος λειτουργίας του Undo είναι η αποθήκευση του μοντέλου κάθε στιγμή όπου συμβαίνει μια ενέργεια, έτσι ώστε όταν ο χρήστης κάνει αναίρεση να επανέλθουμε στο προηγούμενο μοντέλο. Αυτό όμως συνεπάγεται ότι στο ιστορικό καταγράφονται ολόκληρες εικόνες οι οποίες περιέχονται στο μοντέλο. Μια πιθανή βελτίωση θα μπορούσε να ήταν να αποθηκεύονται μόνο οι αλλαγές, ξεχωριστά. Δηλαδή εάν ο χρήστης ζωγραφίσει ένα τετράγωνο, να αποθηκευτούν μόνο οι συντεταγμένες του και όχι ολόκληρη η εικόνα.

- Αλγόριθμος Flood Fill: ο αλγόριθμος αυτός εφαρμόζεται στο Bucket Tool και αναλαμβάνει να χρωματίσει την κατάλληλη περιοχή. Όταν όμως η περιοχή αυτή είναι σχετικά μεγάλη, η εφαρμογή αργεί κάποια κλάσματα του δευτερολέπτου να εφαρμόσει την αλλαγή από την στιγμή που δοθεί η εντολή από τον χρήστη. Όσο μεγαλύτερη περιοχή, τόσο μεγαλώνει αυτή η απόκλιση.
Σημειώνεται ότι το πρόβλημα παραμένει, παρόλο που η Canvas εφαρμόζει μια βελτιωμένη μορφή του αλγορίθμου. Ο αυθεντικός αλγόριθμος Flood Fill με αναδρομή, είχε πολύ χειρότερες αποδόσεις.

Τα παραπάνω προβλήματα είναι γνωστά στους προγραμματιστές και ως «bugs».

5.2.2 - Προσθήκη έξτρα λειτουργιών

Οι λειτουργίες οι οποίες μπορούν να προτεθούν στην εφαρμογή είναι πάρα πολλές. Απόδειξη για αυτό είναι πολλές γνωστές εφαρμογές στην αγορά οι οποίες έχουν πολλές περισσότερες παροχές από αυτές της Canvas. Επομένως δεν έχει νόημα η απαρίθμηση των λειτουργιών που της λείπουν, αλλά μπορεί τουλάχιστον να γίνει μια σύντομη αναφορά σε κάποιους στόχους οι οποίοι υπήρχαν αρχικά και δεν πρόλαβαν να υλοποιηθούν:

- Select Tool. Το επόμενο βήμα εάν υπήρχε περισσότερος χρόνος θα ήταν η δημιουργία του εργαλείου Επιλογής. Αυτό επιτρέπει την επιλογή μιας περιοχής του καμβά και έπειτα την επεξεργασία της. Παράμετροι:
 - Selection Type: Ορθογώνια, κυκλική, ελεύθερη, «μαγική» επιλογή κ.α.
 - Selection Method: Νέα επιλογή, πρόσθεση στη προηγούμενη, αφαίρεση από την προηγούμενη, τομή με την προηγούμενη.
 - Ενέργειες Επιλογής: Copy/Cut, Paste, Deselect, Inverse, set Opacity κ.α.
- Crop Canvas στις επιλογές του Transform.

- Shape: Όπως και στο Photoshop, κάθε σχήμα να αποθηκεύεται ως αντικείμενο «Shape» και να μπορεί να τροποποιηθεί ακόμα και μετά την εισαγωγή τους τον καμβά.
Επίσης, ο χρήστης να μπορεί να αποθηκεύει ένα selection ως Custom Shape.
- Εισαγωγή εικόνας ως layer σε έναν υπάρχων καμβά.
- Settings για τον καθορισμό γενικών ρυθμίσεων της εφαρμογής.
- Ruler Tool, για την μέτρηση αποστάσεων και συμμετρίας στον καμβά.
- Μεταμόρφωση ενός selection ή shape (σαν τις επιλογές *Scale, Rotate, Flip, Skew, Distort, Perspective* και *Wrap* του Photoshop)
- Έξτρα εργαλεία, όπως τα *Smudge, Blur, Sharpen, Smoothen, Shrink-Expand* και *Liquify*, όπως στην γνωστή εφαρμογή IbisPaint)
- Log in και Upload σε Cloud
- Φίλτρα: *Brightness, Contrast, Saturation*, κ.α.

BIBΛΙΟΓΡΑΦΙΑ

- [1] <https://theartofeducation.edu/2019/05/29/digital-vs-traditional-art-is-one-better-than-the-other/>
- [2] <https://www.creativeblog.com/advice/the-best-software-for-digital-artists>
- [3] <https://www.lifewire.com/best-digital-art-software-4705458>
- [4] <https://www.techradar.com/best/best-drawing-and-painting-software>
- [5] <https://conceptartempire.com/digital-painting-software/>
- [6] <https://wethegeek.com/best-digital-art-software/>
- [7] <https://www.digitalartsonline.co.uk/features/creative-software/best-painting-software-for-artists/>
- [8] <https://www.clipstudio.net/how-to-draw/archives/155465>
- [9] <https://www.tomsguide.com/round-up/best-drawing-apps>
- [10] <https://play.google.com/store/apps/details?id=com.adsk.sketchbook>
- [11] <https://reviews.financesonline.com/p/autodesk-sketchbook/>
- [12] https://play-lh.googleusercontent.com/NIhaPLVMI69snWk7IOWADeBwU-8FG5VL1q41_fmd6TVa4FEIqXQw3Ar3BovrXvza2Q=s180-rw
- [13] <https://play.google.com/store/apps/details?id=jp.co.celsys.clipstudiopaint.googleplay&hl=en&gl=US>
- [14] [https://www.lifewire.com/thmb/F2r7tNZyM9JTKSr3D10-ide8u2E=/1000x0/filters:no_upscale\(\):max_bytes\(150000\):strip_icc\(\):format\(webp\)/Clip_Studio_Paint-96c554726adb4f0f81e830e338971310.jpg](https://www.lifewire.com/thmb/F2r7tNZyM9JTKSr3D10-ide8u2E=/1000x0/filters:no_upscale():max_bytes(150000):strip_icc():format(webp)/Clip_Studio_Paint-96c554726adb4f0f81e830e338971310.jpg)

- [15] <https://play.google.com/store/apps/details?id=jp.ne.ibis.ibispaint.app&hl=en&gl=US>
- [16] https://play-lh.googleusercontent.com/s8moWkCF9wE-ynJgNyg8k3uhhVlbQLdphqTYJWkrsLRxkFZxx9FvykHmwXYmTl_h0l8=s180-rw
- [17] https://storage.googleapis.com/gweb-uniblog-publish-prod/images/HeroArticlePage_1500x850.max-1000x1000.jpg
- [18] [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
- [19] https://en.wikipedia.org/wiki/HTC_Dream
- [20] <https://www.android.com/gms/>
- [21] <https://www.appbrain.com/stats/number-of-android-apps>
- [22] <https://developer.android.com/guide/components/fundamentals>
- [23] <https://developer.android.com/reference/android/app/Activity>
- [24] https://developer.android.com/guide/components/images/activity_lifecycle.png
- [25] <https://developer.android.com/studio/index.html>
- [26] https://commons.wikimedia.org/wiki/File:Android_Studio_Trademark.svg
- [27] <https://developer.android.com/studio/releases#older-releases>
- [28] <https://venturebeat.com/2014/12/08/google-releases-android-studio-1-0-the-first-stable-version-of-its-ide/>
- [29] <http://social.techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development/>
- [30] <https://www.androidauthority.com/develop-android-apps-languages-learn-391008/>

- [31] [https://en.wikipedia.org/wiki/Library_\(computing\)](https://en.wikipedia.org/wiki/Library_(computing))
- [32] <https://en.wikipedia.org/wiki/GitHub>
- [33] <https://github.com/>
- [34] <https://github.com/logos>
- [35] <https://gradle.org/>
- [36] https://en.wikipedia.org/wiki/Android_application_package
- [37] https://developer.android.com/images/tools/studio/build-process_2x.png
- [38] <https://en.wikipedia.org/wiki/Software>
- [39] <https://en.wikipedia.org/wiki/JetBrains>
- [40] <https://www.jetbrains.com/company>
- [41] https://www.jetbrains.com/company/brand/img/jetbrains_logo.png
- [42] <https://material.io/design>
- [43] https://upload.wikimedia.org/wikipedia/commons/thumb/c/c7/Google_Material_Design_Logo.svg/1024px-Google_Material_Design_Logo.svg.png
- [44] <https://searchmobilecomputing.techtarget.com/definition/Google-Firebase>
- [45] https://upload.wikimedia.org/wikipedia/commons/thumb/3/37/Firebase_Logo.svg/1280px-Firebase_Logo.svg.png
- [46] <https://developer.android.com/topic/libraries/architecture/viewmodel>
- [47] <https://developer.android.com/images/topic/libraries/architecture/viewmodel-lifecycle.png>
- [48] <https://developer.android.com/guide/navigation>

- [49] <https://devopedia.org/kotlin-language>
- [50] <https://medium.com/android-news/kotlin-best-features-a3facac4d6fd>
- [51] https://upload.wikimedia.org/wikipedia/commons/thumb/d/d4/Kotlin_logo.svg/768px-Kotlin_logo.svg.png
- [52] https://google-developer-training.github.io/android-developer-advanced-course-concepts/images/1-2-c-fragment-lifecycle-and-communications/fragment_communications.png
- [53] <https://fonts.google.com/icons?selected=Material+Icons:info>
- [54] <https://fonts.google.com/?selected=Material+Icons:info>
- [55] <https://developer.android.com/studio/build>
- [56] https://docs.gradle.org/current/userguide/tutorial_using_tasks.html#tutorial_using_tasks
- [57] https://www.google.com/books/edition/API_Design_for_C++/IY29LyIT85wC
- [58] https://en.wikipedia.org/wiki/Integrated_development_environment
- [59] <https://en.wikipedia.org/wiki/XML>

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α

Συντομογραφίες

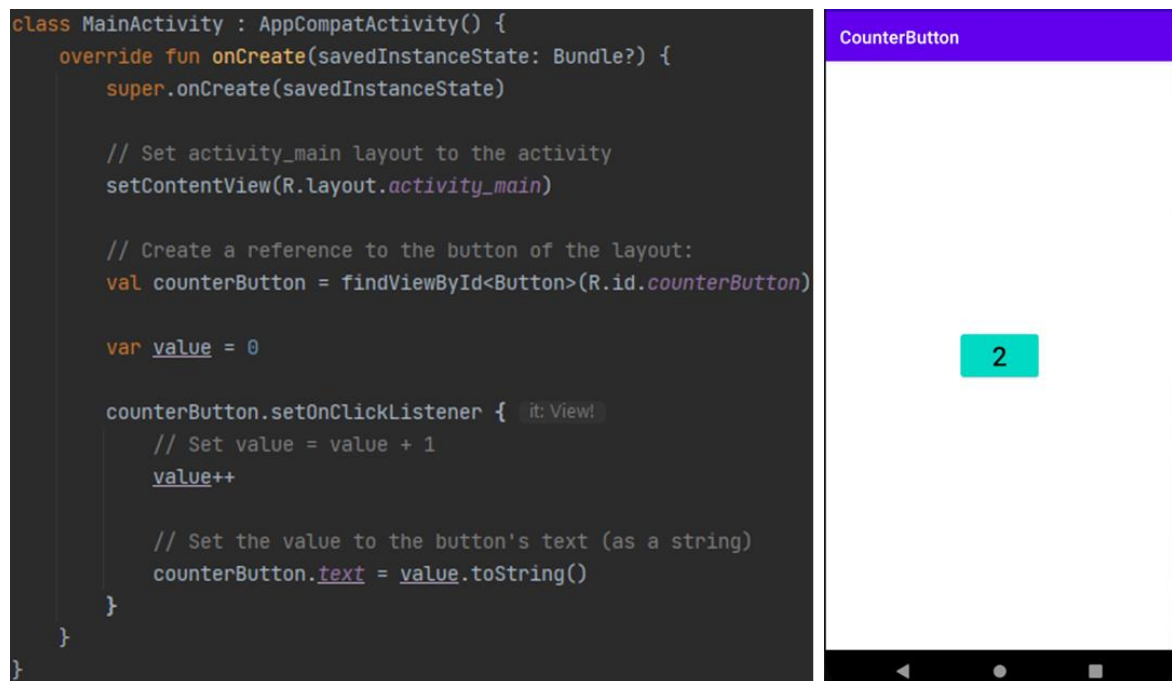
- Api – Application Interface: Μια σύνδεση μεταξύ υπολογιστών ή μεταξύ προγραμμάτων υπολογιστών. Είναι ένας τύπος διεπαφής λογισμικού, που προσφέρει μια υπηρεσία σε άλλα κομμάτια λογισμικού [57].
- APK – Android Package
- ARGB – Alpha, Red, Green, Blue
- GPS – Global Positioning System
- HD – High Definition
- IDE – Integrated Development Environment: Είναι μια εφαρμογή λογισμικού που παρέχει ολοκληρωμένες εγκαταστάσεις στους προγραμματιστές υπολογιστών για την ανάπτυξη λογισμικού. Ένα IDE κανονικά αποτελείται από τουλάχιστον έναν επεξεργαστή πηγαίου κώδικα, εργαλεία αυτοματοποίησης κατασκευής και ένα πρόγραμμα εντοπισμού σφαλμάτων.[58]
- JPEG – Joint Photographic Experts Group
- PNG – Portable Network Graphics
- PSD – Photoshop Document
- SD – Standard Definition
- UI - User Interface
- XML - Extensible Markup Language: Είναι μια γλώσσα σήμανσης που ορίζει ένα σύνολο κανόνων για την κωδικοποίηση εγγράφων σε μορφή τόσο αναγνώσιμη από τον άνθρωπο όσο και από μηχανή. [59]

ΠΑΡΑΡΤΗΜΑ Β

Υλοποίηση ενός `OnClickListener`

Το πιο αντιπροσωπευτικό παράδειγμα αξιοποίησης του user input για την επίτευξη ενός σκοπού είναι η ανάθεση ενός «`OnClickListener`» σε ένα συγκεκριμένο κουμπί, δηλαδή ανάθεση μιας μεθόδου η οποία θα εκτελείται μόλις ο χρήστης πατήσει το κουμπί.

Στο παρακάτω παράδειγμα (εικόνα Β.1), αναθέτουμε μια τέτοια μέθοδο στο κουμπί «`counterButton`» με την χρήση της μεθόδου `setOnClickListener{...}`. Το κουμπί αυτό (το οποίο περιέχεται στο layout «`activity_main`»), δείχνει ένα αριθμό ο οποίος είναι αρχικά μηδέν, και με κάθε κλικ, αυξάνεται κατά 1.



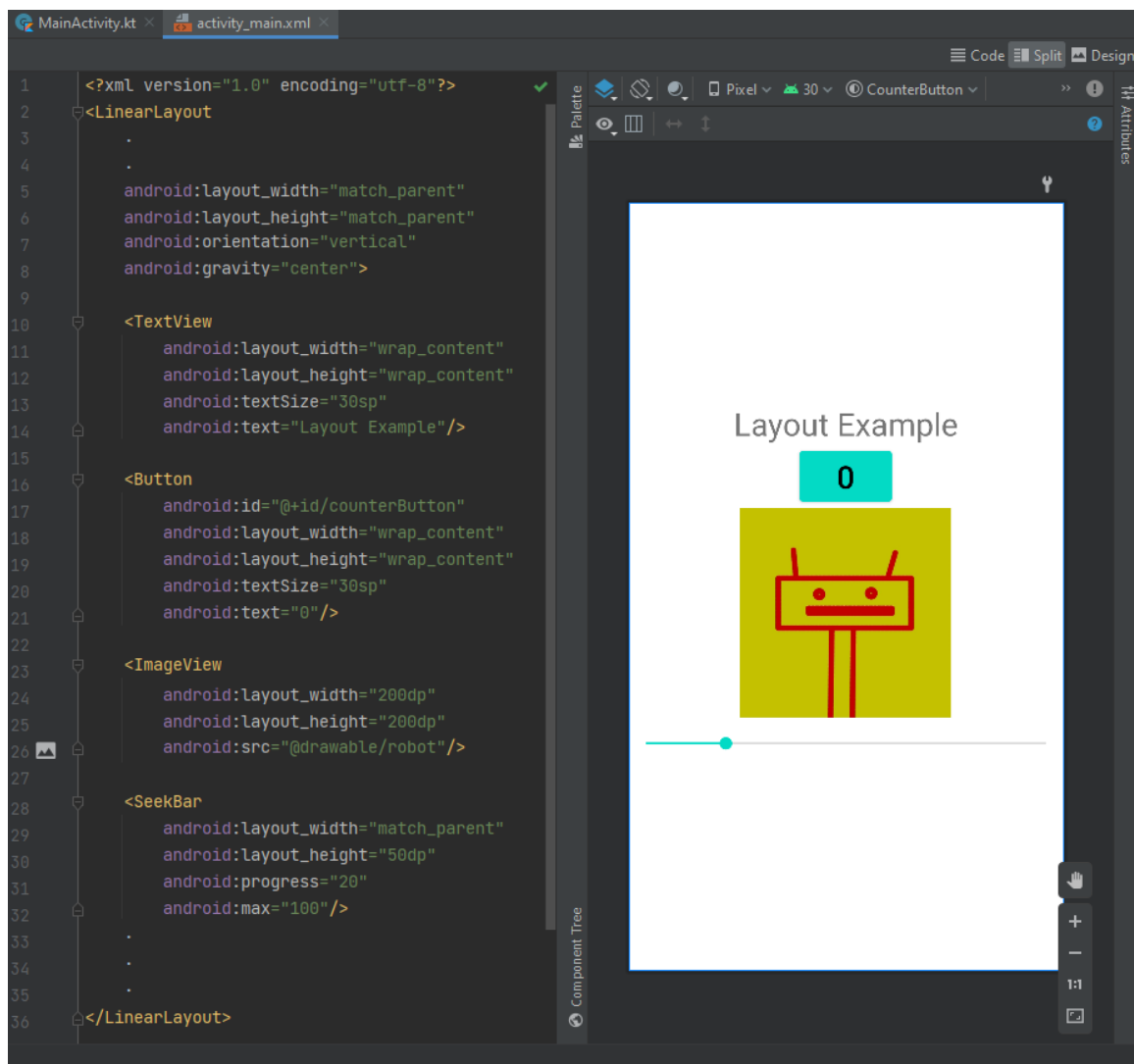
Εικόνα Β.1 – Υλοποίηση click listener (αριστερά), και αντίστοιχο στιγμιότυπο μετά από δυο κλικ (δεξιά).

ΠΑΡΑΡΤΗΜΑ Γ

Παράδειγμα ενός layout

Η δημιουργία των layouts είναι ένα από τα πρώτα βήματα για την δημιουργία μιας εφαρμογής. Η δουλειά τους είναι να καθορίζουν τι θα βλέπει ο χρήστης στην οθόνη του.

Στην παρακάτω εικόνα (εικόνα Γ.1) παρουσιάζεται ο τρόπος που απλές εντολές XML μπορούν να μεταφραστούν σε μια συγκεκριμένη διαμόρφωση οθόνης. Το View το οποίο είναι «γονέας» όλων των υπόλοιπων είναι το LinearLayout το οποίο είναι ένα ViewGroup όπου εμφανίζει τα περιεχόμενα του το ένα μετά το άλλο, είτε οριζόντια είτε κάθετα.



Εικόνα Γ.1 – Παράδειγμα διάταξης διαφόρων στοιχείων View σε layout (κώδικας & προεπισκόπηση).

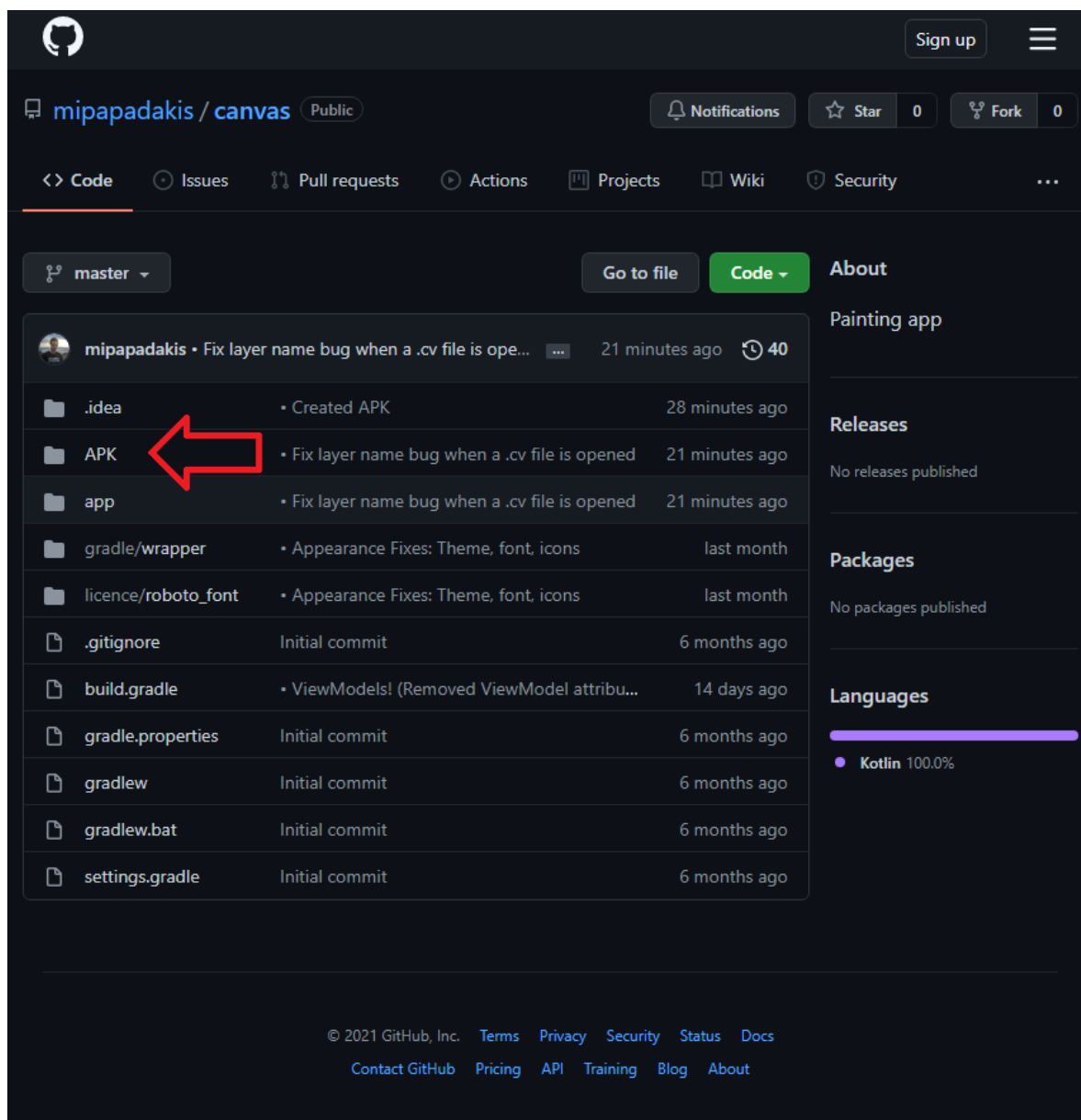
ΠΑΡΑΡΤΗΜΑ Δ

Οδηγίες εγκατάστασης της εφαρμογής

Ο πηγαίος κώδικας της Canvas βρίσκεται στο Github, στον παρακάτω σύνδεσμο:

<https://github.com/mipapadakis/canvas>

Εκεί, στον φάκελο *APK*, υπάρχει το αρχείο *canvas.apk*, το οποίο μπορεί κάποιος να το κατεβάσει, και όταν το ανοίξει, η εφαρμογή να εγκατασταθεί (σε συσκευή Android).



Εικόνα Δ.1 – Παράδειγμα διάταξης διαφόρων στοιχείων View σε layout (κώδικας & προεπισκόπηση).