

**ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη εφαρμογής βοτανικού κήπου σε περιβάλλον Android
με χρήση σάρωσης QR Code**

**Ιωάννης Χ. Σαλαπίδας
Ερρίκος Γ. Τζαννετάτος**

Εισηγητής: Ιωάννης Έλληνας, Καθηγητής

**ΑΘΗΝΑ
ΙΟΥΝΙΟΣ 2017**

(Κενό φύλλο)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη εφαρμογής βοτανικού κήπου σε περιβάλλον Android με χρήση
σάρωσης QR Code**

**Ιωάννης Χ. Σαλαπίδας
Α.Μ. 41791**

**Ερρίκος Γ. Τζαννετάτος
Α.Μ. 41762**

Εισηγητής:

Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης 5/05/2017

(Κενό φύλλο)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Οι κάτωθι υπογεγραμμένοι Ιωάννης Σαλταπίδας του Χρήστου και Ερρίκος Τζαννετάτος του Γερασίμου με αριθμό μητρώου 41791 και 41762 αντίστοιχα φοιτητές του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβουμε την εκπόνηση της Πτυχιακής Εργασίας μας, δηλώνουμε ότι ενημερωθήκαμε για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε με την άψογη συνεργασία μας τόσο σε ακαδημαϊκό όσο και σε φιλικό επίπεδο ανατρέχοντας σε κάθε είδους πηγές που μας βοήθησαν να ξεπεράσουμε όποιο πρόβλημα προέκυπτε. Η εφαρμογή μας είναι σε περιβάλλον Android Studio και όντας τελειόφοιτοι των Ηλεκτρονικών Υπολογιστικών Συστημάτων είχαμε κάποιες αξιόλογες βάσεις που μας βοήθησαν να ξεδιπλώσουμε τον αχανές αλλά ταυτόχρονα πολύ ενδιαφέρον κόσμο του Android. Θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας Ιωάννη Έλληνα για τις πολύτιμες συμβουλές και καθοδήγησή του σε όποια απορία μας δημιουργήθηκε. Επίσης, προσωπικά σαν Γιάννης, θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε όλα αυτά τα χρόνια οικονομικά αλλά και ψυχικά για να σπουδάσω στο αντικείμενο που πάντα αγαπούσα από μικρή ηλικία. Ωστόσο, προσωπικά σαν Ερρίκος, θα ήθελα να ευχαριστήσω την οικογένεια μου για όλη την οικονομική στήριξη όλα αυτά τα χρόνια όπως και τους συμφοιτητές μου Φαίδωνα Πατερομιχελάκη, Κώστα Φυγετάκη και Ελεάνα Κυτούγια. Αφιερώνω αυτή την πτυχιακή την σε έναν άνθρωπο που δεν πρόλαβε θαυμάσει, στον παππού μου.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία στοχεύει στην δημιουργία μίας ολοκληρωμένης εφαρμογής σε Android κινητά, για τον βοτανικό κήπο στο αγρόκτημα 'Αμφίκαια' στην Αμφίκλεια, όπου με την κάμερα ενός Android κινητού θα σαρώνει εικόνες σε μορφή QR Code με διαφορετική τιμή η κάθε μια. Θα αντιπροσωπεύει το αντίστοιχο φυτό και θα εμφανίζει στον χρήστη τις ανάλογες πληροφορίες. Η υλοποίηση έγινε σε 4 γλώσσες ώστε να είναι προσιτό σε μεγαλύτερο εύρος ανθρώπων και από το εξωτερικό. Ο χρήστης έχει στη διάθεσή του έναν 'ξεναγό' του αγροκτήματος ο οποίος τον ωθεί να τον επισκεφτεί και να γνωρίσει την χλωρίδα της Ελληνικής φύσης. Επιπρόσθετα υπάρχουν στοιχεία και για το αγρόκτημα, στην official σελίδα του στο Internet όπως και το Facebook. Τέλος υπάρχει χάρτης που καθοδηγεί τους επισκέπτες για τον τρόπο με τον οποίο θα μπορέσουν να βρουν την τοποθεσία του αγροκτήματος μέσω Google maps.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Software/Android
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Προγραμματισμός, Android Studio, java, qr-codes

ABSTRACT

This project is about the creation of a complete application for Android smartphone ,for the votanic gardens of Amfikaia , which is located in Amfikaia.With the use of an Android cellphone you can scan images for QR code type with differend price each one.It will represent every plant and appear to the user the proportionate information.The implementation is in four languages that to be accessible in a biggest range of people from abroad.The user has in his possecion a guide of the farm,which provoke him to visit and meet the Greek flora.Furthermore the information of the Amfikaia farm can be found in the official Facebook page.To the end of it all there is a map that can guide every visitor to the Gardens through Google Maps.

SCIENTIFIC AREA: Software/Android

Key Words: Programming, Android Studio, java,qr-codes

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	7
ABSTRACT	8
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	12
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	15
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	16
ΚΕΦΑΛΑΙΟ 1- ΕΙΣΑΓΩΓΗ	17
1.1 Το αντικείμενο της διπλωματικής	17
1.2 Οργάνωση του κειμένου	17
ΚΕΦΑΛΑΙΟ 2- ΘΕΩΡΗΤΙΚΟ ΚΟΜΜΑΤΙ	19
2.1 Ιστορική αναδρομή του Προγραμματισμού	19
2.2 Λειτουργικά συστήματα	24
2.3 Smartphones / Έξυπνα τηλέφωνα	27
2.4 Λειτουργικό σύστημα Android	30
2.4.1 Αναλυτικά οι εκδόσεις του Android	33
2.5 Δομή Λειτουργικού συστήματος	42
2.6 Βασικές αρχές μιας εφαρμογής Android	45
2.7 Συστατικά Android Εφαρμογών	47
2.8 Ασφάλεια στο Android	50
2.9 Γιατί Android	51
ΚΕΦΑΛΑΙΟ 3-ANDROID STUDIO	53
3.1 Τι είναι το Android Studio	53
3.1.2 Χαρακτηριστικά του Android Studio	54
3.1.3 Χαρακτηριστικά του Android Studio IDE	54
3.1.4 Εγκατάσταση του Android Studio	55

3.2	Εργαλεία ανάπτυξης Android Software Java Development	<u>59</u>
3.2.1	Εργαλεία αποσφαλμάτωσης	<u>60</u>
3.2.2	Εικονικές Συσκευές Android (Android Virtual Devices)	<u>63</u>
3.2.3	Εργαλείο καταγραφής συμβάντων (LogCat)	<u>64</u>
3.2.4	Εργαλείο ανάπτυξης (Development Tools)	<u>66</u>
3.2.5	Εργαλεία κατασκευής (Build Tools)	<u>66</u>
3.2.6	Εργαλεία Πλατφόρμας(Platform tools)	<u>67</u>
ΚΕΦΑΛΑΙΟ 4- ANDROID SDK ΚΑΙ ANDROID PROJECT		<u>69</u>
4.1	Android SDK	<u>69</u>
4.1.1	Χαρακτηριστικά του SDK	<u>69</u>
4.1.2	Περιεχόμενο του SDK	<u>70</u>
4.1.3	Εγκατάσταση του Android SDK	<u>71</u>
4.1.4	Πληροφορίες που μας δίνει ο SDK Manager	<u>73</u>
4.2	Android Java Project	<u>77</u>
4.2.1	Δημιουργία Android Project	<u>78</u>
4.2.2	Διαγραφή Android Project	<u>84</u>
ΚΕΦΑΛΑΙΟ 5 – ΑΝΑΛΥΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ		<u>85</u>
5.1	Κλάση MainActivity	<u>85</u>
5.1.1	Αρχείο Layout XML activity_main	<u>91</u>
5.2	Κλάση SplashScreen	<u>95</u>
5.2.1	Αρχείο Layout XML splashscreen	<u>98</u>
5.3	Κλάση Languages	<u>99</u>
5.3.1	Αρχείο Layout XML lang	<u>102</u>
5.4	Κλάση Qr Code Scanner	<u>104</u>
5.5	Κλάση NewURL	<u>109</u>

5.5.1 Αρχείο Layout XML webview	<u>112</u>
5.6 Κλάση InfoURL	<u>113</u>
5.7 Κλάση Info_But	<u>115</u>
5.8 Κλάση FlashbackURL	<u>117</u>
5.9 Κλάση BeezURL	<u>119</u>
5.10 Αρχείο Layout XML AndroidManifest	<u>120</u>
ΚΕΦΑΛΑΙΟ 6 - ΕΠΕΞΗΓΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	<u>126</u>
6.1 Ρόλος Χρήστη	<u>126</u>
6.2 Ρόλος Προγραμματιστή	<u>133</u>
ΚΕΦΑΛΑΙΟ 7 - ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ	<u>135</u>
7.1 Σύνοψη της πτυχιακής εργασίας	<u>135</u>
7.2 Προοπτικές	<u>135</u>
ΒΙΒΛΙΟΓΡΑΦΙΑ	<u>137</u>

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 2.1: Το πρώτο Smartphone	<u>28</u>
Σχήμα 2.2: Apple vs Android	<u>29</u>
Σχήμα 2.3: Λογότυπο πλατφόρμας Android	<u>31</u>
Σχήμα 2.4: Android 1.5, Cupcake	<u>35</u>
Σχήμα 2.5: Android 1.6, Donut	<u>36</u>
Σχήμα 2.6: Android 2.0, Éclair	<u>37</u>
Σχήμα 2.7: Android 2.2, Froyo	<u>37</u>
Σχήμα 2.8: Android 2.3, Gingerbeard	<u>37</u>
Σχήμα 2.9: Android 3.0, Honeybomb	<u>38</u>
Σχήμα 3.0: Android 4.0, Ice Cream Sandwich	<u>39</u>
Σχήμα 3.1: Android 4.1, Jelly Bean	<u>39</u>
Σχήμα 3.2: Android 4.4, KitKat	<u>40</u>
Σχήμα 3.3: Android 5.0, Lollipop	<u>40</u>
Σχήμα 3.4: Android 6.0, Marshmallow	<u>41</u>
Σχήμα 3.5: Android 7.0, Nougat	<u>42</u>
Σχήμα 3.6: Αρχιτεκτονική του Android	<u>42</u>
Σχήμα 3.7: Android Studio	<u>53</u>
Σχήμα 3.8: Καλωσόρισμα στον οδηγό εγκατάστασης	<u>56</u>
Σχήμα 3.9: Επιλογή Χρήστη	<u>57</u>
Σχήμα 4.0: Επιλογή τοποθεσίας εγκατάστασης	<u>58</u>
Σχήμα 4.1: Τέλος Εγκατάστασης	<u>58</u>
Σχήμα 4.2: Android Studio Mac OS X	<u>59</u>
Σχήμα 4.3: Δημιουργία AVD Manager	<u>64</u>

Σχήμα 4.4: Κατέβασμα Εφαρμογής	<u>71</u>
Σχήμα 4.5: Εγκατάσταση Android SDK	<u>72</u>
Σχήμα 4.6: Πίνακας Android πλατφόρμων	<u>73</u>
Σχήμα 4.7: Εμφάνιση λεπτομερειών πακέτων	<u>74</u>
Σχήμα 4.8: SDK Tools	<u>75</u>
Σχήμα 4.9: SDK Tools Εμφάνιση λεπτομερειών	<u>76</u>
Σχήμα 5.0: Ενημερώσεις SDK εργαλείων	<u>77</u>
Σχήμα 5.1: Τρόπος δημιουργίας Android Project	<u>79</u>
Σχήμα 5.2: Δεύτερος τρόπος δημιουργίας Android Project	<u>79</u>
Σχήμα 5.3: Συμπλήρωση πεδίων του Android Project	<u>80</u>
Σχήμα 5.4: Επιλογή μορφής παραγόντων	<u>81</u>
Σχήμα 5.5: Επιλογή τύπου δραστηριότητας	<u>82</u>
Σχήμα 5.6: Προσαρμογή δραστηριότητας	<u>83</u>
Σχήμα 5.7: Περιβάλλον ανάπτυξης	<u>83</u>
Σχήμα 5.8: Κλείσιμο Project	<u>84</u>
Σχήμα 5.9: Διαγραφή Project	<u>84</u>
Σχήμα 6.0: Activity main XML	<u>91</u>
Σχήμα 6.1: Xml SplashScreen	<u>98</u>
Σχήμα 6.2: Xml Lang	<u>102</u>
Σχήμα 6.3: Xml Webview	<u>112</u>
Σχήμα 6.4: App Info	<u>126</u>
Σχήμα 6.5: App Permissions	<u>126</u>
Σχήμα 6.6: Choose Language	<u>127</u>
Σχήμα 6.7: Μενού	<u>128</u>

Σχήμα 6.8: Πληροφορίες Αγροκτήματος	<u>129</u>
Σχήμα 6.9: Gps	<u>130</u>
Σχήμα 7.0: Qr Code	<u>131</u>
Σχήμα 7.1: Φυτά	<u>131</u>
Σχήμα 7.2: Περίληψη Εφαρμογής	<u>132</u>
Σχήμα 7.3: Υπομενού	<u>132</u>

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1.1: Εκδόσεις Android	<u>32</u>
--	------------------

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

- QR** Quick Response
- SDK** Software Development Kit
- APP** Application
- PDA** Personal Digital Assistant
- API** Application Programming Interface
- XML** eXtensible Markup Language
- SMS** Short Message Service
- DDMS** Dalvik Debug Monitor Server
- GPS** Global Positioning System
- GSM** Global System for Mobile Communications
- AVD** Android Virtual Device
- UI** User Interface
- RAM** Random Access Memory
- GIF** Graphics Interchange Format
- ID** IDentity
- SD** Secure Digital
- URI** Uniform Resource Identifier
- URL** Uniform Resource Locator
- XML** eXtensible Markup Language

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Τα **QR Codes** (Quick Response Codes) έχουν ξεπροβάλλει εδώ και καιρό σε διάφορα σημεία του κόσμου, σε περιοδικά, κάρτες ακόμα και σε κτίρια ή μπλουζάκια. Ο κώδικας **QR** είναι ένας γραμμωτός κώδικας (barcode) δύο διαστάσεων, που δημιουργήθηκε από την ιαπωνική εταιρεία Denso-Wave το 1994. Το "QR" προέρχεται από τα αρχικά των λέξεων "Quick Response" (Γρήγορη Ανταπόκριση), γιατί οι δημιουργοί του είχαν ως κύριο σκοπό τα δεδομένα, που περιέχονται στον κώδικα, να αποκωδικοποιούνται με μεγάλη ταχύτητα. Ο Κώδικας QR είναι πολύ διαδεδομένος στην Ιαπωνία, όπου αποτελεί το πιο δημοφιλές είδος κώδικα δύο διαστάσεων.

Η σάρωση ενός κώδικα **QR**, γίνεται με την βοήθεια ενός έξυπνου κινητού. Για να γίνει αυτό, ο χρήστης χρειάζεται μία εφαρμογή η οποία μετατρέπει την κάμερα του κινητού σε σαρωτή κώδικα QR όπως κάνουμε και εμείς στην εφαρμογή μας. Μέσα σε ένα κώδικα QR μπορεί να αποθηκευτεί, π.χ., ένας σύνδεσμος προς μια ιστοσελίδα. Ο χρήστης σαρώνει με το κινητό του τηλέφωνο τον κώδικα QR και πλοηγείται αυτόματα στην ιστοσελίδα. Αυτή η πράξη της σύνδεσης από το φυσικό κόσμο είναι γνωστή ως *hardlink* ή υπερσύνδεση με φυσικό κόσμο.

Οι χρήστες μπορούν, επίσης, να δημιουργήσουν και να εκτυπώσουν τους δικούς τους κώδικες QR με χρήση διάφορων ελεύθερων λογισμικών παραγωγής κώδικα QR που υπάρχουν στο δίκτυο. Μέσα μπορούν να αποθηκεύσουν όποιο μήνυμα θέλουν.

Παρακάτω περιγράφουμε το αντικείμενο της πτυχιακής μας εργασίας αλλά και την οργάνωση του κειμένου σε κάθε κεφάλαιο.

1.1 Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Στην παρούσα πτυχιακή εργασία παρουσιάζεται μια εφαρμογή για Android κινητά που αναπτύχθηκε σε Android Studio περιβάλλον. Η ανάπτυξη και η οπτικοποίηση πραγματοποιήθηκαν με τις γλώσσες προγραμματισμού Java και XML.

Μέσα στην εφαρμογή οι χρήστες που εισέρχονται ως επισκέπτες μπορούν αρχικά να επιλέξουν την γλώσσα στην οποία επιθυμούν να είναι το μενού της εφαρμογής. Έπειτα έχουν την δυνατότητα να περιηγηθούν σε ένα μενού που τους δίνει την επιλογή να γνωρίσουν καλύτερα το αγρόκτημα, την ιστορία του, την τοποθεσία του και κάποιες πολύ χρήσιμες πληροφορίες για την ιδιοσυγκρασία των φυτών του βοτανικού κήπου, την ιστορία τους, κλπ. Επίσης μπορούν να γνωρίσουν την ιστορία των μελισσών, τις φυλές και το πώς παράγεται το μέλι στις εγκαταστάσεις τους. Υπάρχει χάρτης ο οποίος βοηθάει τον επισκέπτη-χρήστη να φτάσει στο αγρόκτημα από όποιο μέρος της Ελλάδος έρχεται ή του εξωτερικού μέσω του Google Maps.

1.2 Οργάνωση του κειμένου

Το κείμενο στο 2^ο κεφάλαιο αρχίζει με την αναδρομή στον προγραμματισμό και συνεχίζει με πληροφορίες σχετικά με τα Android κινητά όπως τις εκδόσεις, τις βασικές δομές του και την ασφάλεια.

Εν συνέχεια, στο 3^ο κεφάλαιο γίνεται αναφορά στο τι είναι το Arduino πως εγκαθιστάται και τα βασικά εργαλεία του.

Επίσης, στο 4^ο κεφάλαιο μιλάμε για τον Android SDK τα χαρακτηριστικά του και βασικά εργαλεία του.

Ακόμα, στο 5^ο κεφάλαιο παρουσιάζεται ο κώδικας της εφαρμογής. Σ' αυτό το κεφάλαιο αναφέρεται και σχολιάζεται κάθε γραμμή κώδικα σχετικά με την λειτουργία και την χρήση του.

Επιπλέον, στο κεφάλαιο 6^ο επεξηγούμε πως λειτουργεί η εφαρμογή αλλά και τον ρόλο του προγραμματιστή-χρήστη.

Τέλος, στο κεφάλαιο 7^ο παρουσιάζονται τα συμπεράσματα και οι προοπτικές που θα μπορούσαν να υπάρξουν στο μέλλον για να την βελτίωση της εφαρμογής.

ΚΕΦΑΛΑΙΟ 2

ΘΕΩΡΗΤΙΚΟ ΚΟΜΜΑΤΙ

2.1 Ιστορική αναδρομή του Προγραμματισμού

Προγραμματισμός υπολογιστών καλείται το σύνολο των διαδικασιών σύνταξης ενός υπολογιστικού προγράμματος, συνήθως ως υλοποίηση κάποιων αλγορίθμων ύστερα από προσεκτική σχεδίαση, για την αυτοματοποιημένη εκτέλεση εργασιών ή επίλυση κάποιου υπολογιστικού προβλήματος από έναν υπολογιστή. Ο προγραμματισμός περιλαμβάνει επίσης τον έλεγχο του προγράμματος για την επαλήθευση της ακρίβειας και της ορθότητάς του (αποσφαλμάτωση), καθώς και την προπαρασκευή των οδηγιών με τις οποίες ένας υπολογιστής θα εκτελέσει τις εντολές που καθορίζονται στις προδιαγραφές του προγράμματος. Θεμελιώδη ρόλο στον υπολογιστικό προγραμματισμό διαδραματίζουν οι χιλιάδες διαφορετικές γλώσσες προγραμματισμού, δηλαδή οι τυποποιημένες τυπικές γλώσσες οι οποίες είναι απαραίτητες για τη σύνθεση ενός προγράμματος. Ο πηγαίος κώδικας του προγράμματος αποτελείται από τις εντολές που έχει γράψει ο προγραμματιστής χρησιμοποιώντας μία γλώσσα προγραμματισμού. Το πηγαίο πρόγραμμα για να εκτελεστεί από την Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ) του Η/Υ -τον επεξεργαστή, πρέπει στη συνέχεια να μεταφραστεί σε γλώσσα μηχανής από το εξειδικευμένο λογισμικό, τους μεταγλωττιστές και τους διερμηνείς .

Με την πάροδο του χρόνου παρουσιάστηκαν ανάγκες για χρήση περαιτέρω γλωσσών προγραμματισμού ανάλογα με το είδος του προβλήματος αλλά και με την προτίμηση του προγραμματιστή, οι οποίες αποτελούνται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που ορίζουν τη δομή και το νόημα, αντίστοιχα, των προτάσεων της γλώσσας.

Ο ψηφιακός υπολογιστής μπορεί να χρησιμοποιήσει μόνο δυαδικούς αριθμούς και να εκτελέσει διαδοχικά εντολές που του δίνονται κι αυτές με μορφή δυαδικού αριθμού. Τα προγράμματα που φτιάχνονται με τέτοιες εντολές λέμε πως φτιάχνονται σε γλώσσα μηχανής. Όταν, για παράδειγμα, θέλουμε να αυξήσουμε το περιεχόμενο κάποιου καταμετρητή N κατά 2, δίνουμε εντολές σε γλώσσα μηχανής, που μοιάζουν κάπως έτσι:

000001000000001100101011

0010000000000000000010000

000010000000001100101011

Αυτές οι μεγάλες σειρές από 0 και 1 ήταν κουραστικές για τον άνθρωπο. Θα βελτιωνόταν κάπως η κατάσταση, αν γράφονταν οι εντολές αυτές με οκταδικούς αριθμούς:

01001453 1000020 02001453

Έτσι οι εντολές διαβάζονταν λίγο πιο εύκολα, πάλι όμως δεν ήταν απλό να δει κανείς αμέσως ποια δουλειά έκαναν αυτές οι εντολές. Επίσης, αν ήθελαν οι προγραμματιστές να κάνουν διορθώσεις, προσθήκες και διαγραφές εντολών σε πρόγραμμα γραμμένο σε γλώσσα μηχανής, αντιμετώπιζαν τεράστιες δυσκολίες σε μια διαδικασία πολύ ευάλωτη από λάθη. Επινοήθηκε λοιπόν μια **συμβολική γλώσσα** για τις εντολές που καταλάβαινε ο υπολογιστής και γράφτηκε ένα συμβολομεταφραστικό πρόγραμμα (assembler), που μετέτρεπε ένα πρόγραμμα συμβολικής γλώσσας σε ένα πρόγραμμα σε γλώσσα μηχανής.

Το προηγούμενο παράδειγμα θα έμοιαζε σε μια υποθετική συμβολική γλώσσα κάπως έτσι:

**LDA N ;N ΕΙΝΑΙ Ο ΜΕΤΡΗΤΗΣ ADD +2 ;ΑΥΞΑΝΕΤΑΙ ΚΑΤΑ 2 ΣΤΑ Ν
;ΑΠΟΘΗΚΕΥΕΤΑΙ Η ΝΕΑ ΤΙΜΗ**

Εδώ, εκτός από τις εντολές προγράμματος στο αριστερό μέρος, υπάρχουν και μερικά επεξηγηματικά σχόλια, (που ο συμβολομεταφραστής τα αγνοεί), τα οποία βοηθάνε πολύ στην φάση συντήρησης του προγράμματος.

Παρ' όλα αυτά όμως, επειδή τα προγράμματα *δεύτερης γενιάς* (αυτά σε συμβολική γλώσσα) είχαν ακριβώς τόσες εντολές όσες είχαν και τα αντίστοιχα προγράμματα σε γλώσσα μηχανής, η ανάπτυξη μεγάλων προγραμμάτων ήταν δύσκολη. Έτσι φτιάχτηκαν οι γλώσσες *τρίτης γενιάς*, όπως είναι η FORTRAN,

όπου ενώ οι προγραμματιστές έδιναν μόνο μια εντολή, (το N να γίνει όσο ήταν συν 2), αναλάμβανε ένα πρόγραμμα - μεταφραστής (compiler) να την αναλύσει σε περισσότερες εντολές σε συμβολική γλώσσα (και σε γλώσσα μηχανής):

N = N + 2

Εκτός από το ότι έτσι γράφονταν λιγότερες εντολές για το ίδιο πρόγραμμα, δημιουργήθηκαν και γλώσσες *ανώτερου επίπεδου* που έμοιαζαν περισσότερο με φυσική γλώσσα. Το ίδιο παράδειγμα σε γλώσσα COBOL θα μπορούσε να γραφτεί έτσι:

ADD 2 TO COUNTER-N .

Το ίδιο παράδειγμα σε γλώσσα C θα μπορούσε να γραφτεί έτσι:

N +=2;

Σήμερα υπάρχουν πολλά είδη γλωσσών γι αυτό τον λόγο μπορούμε να τις κατηγοριοποιήσουμε με διάφορους τρόπους. Οι συνηθέστεροι τρόποι είναι:

- με βάση τον τρόπο οργάνωσης του προγράμματος
- με βάση τον στόχο που έχει η γλώσσα
- με βάση τον τρόπο που περιγράφουν το ζητούμενο αποτέλεσμα

Στην πρώτη περίπτωση προκύπτουν κατηγορίες όπως:

- Διαδικαστικές γλώσσες (procedural) όπου το πρόγραμμα είναι οργανωμένο σε διαδικασίες, που αποτελούνται από σειρές εντολών που περιγράφουν αλγορίθμους. Παραδείγματα γλωσσών που ανήκουν σε αυτή την κατηγορία είναι η Pascal ή η C.
- Αντικειμενοστρεφείς γλώσσες (object-oriented) όπου το πρόγραμμα είναι οργανωμένο σε αντικείμενα. Ένα αντικείμενο είναι μια μονάδα που αποτελείται από την περιγραφή κάποιων δεδομένων και την περιγραφή των αλγορίθμων που τα επεξεργάζονται. Ένα αντικειμενοστρεφές πρόγραμμα αποτελείται από διάφορα αντικείμενα που αλληλεπιδρούν μεταξύ τους. Παραδείγματα αντικειμενοστρεφών γλωσσών είναι η Java ή η C++.
- Συναρτησιακές γλώσσες (functional) όπου οι υπολογισμοί εκφράζονται ως εφαρμογές μαθηματικών συναρτήσεων, σε αντίθεση με τα άλλα είδη

προγραμματισμού όπου οι υπολογισμοί εκφράζονται ως σειρές εντολών, όπου η κάθε μία αλλάζει με κάποιο τρόπο την κατάσταση του συστήματος. Θεωρητικό τους υπόβαθρο είναι ο λ-λογισμός. Χαρακτηριστικές συναρτησιακές γλώσσες είναι η Lisp, η Haskell και η OCaml.

Στην περίπτωση που η κατηγοριοποίηση των γλωσσών προγραμματισμού γίνει με βάση το στόχο που έχει η γλώσσα, υπάρχουν οι παρακάτω κατηγορίες:

- Γλώσσες γενικής χρήσης. Σε αυτήν την κατηγορία ταξινομούνται γλώσσες που δημιουργήθηκαν για τον προγραμματισμό γενικών εφαρμογών, καθώς και πολλές εκπαιδευτικές γλώσσες που αποδείχτηκαν χρήσιμες για την ανάπτυξη γενικών εφαρμογών, όπως η Pascal.
- Γλώσσες προγραμματισμού συστημάτων, που χρησιμοποιούνται συνήθως για τον προγραμματισμό λειτουργικών συστημάτων ή οδηγών (drivers) υλικού, όπου χρειάζεται πολλές φορές ο προγραμματιστής να έχει έλεγχο και γνώση του πως λειτουργεί το υλικό. Η πιο συχνά χρησιμοποιούμενη γλώσσα προγραμματισμού συστημάτων είναι η C.
- Γλώσσες σεναρίων (scripting). Αυτές οι γλώσσες χρησιμοποιούνται συνήθως για τη γρήγορη ανάπτυξη μικρών προγραμμάτων, σε περιπτώσεις που ο χρόνος του προγραμματιστή είναι πιο πολύτιμος από την ταχύτητα εκτέλεσης του προγράμματος, όπως για παράδειγμα συμβαίνει όταν το πρόγραμμα απλά αυτοματοποιεί απλές λειτουργίες. Παραδείγματα γλωσσών σεναρίων (scripting) είναι η Perl, η Python, η Ruby ή τα κελύφη του λειτουργικού συστήματος Unix (shells).
- Γλώσσες ειδικών εφαρμογών. Σε αυτή την κατηγορία ανήκουν γλώσσες που αναπτύχθηκαν ειδικά για μια συγκεκριμένη εφαρμογή. Για παράδειγμα, η γλώσσα PostScript είναι σχεδιασμένη ειδικά για να περιγράφονται με λεπτομέρεια κείμενα προς εκτύπωση, ενώ η γλώσσα Matlab είναι σχεδιασμένη για την επεξεργασία πινάκων από αριθμητικά δεδομένα.
- Παράλληλες ή καταμεμημένες γλώσσες. Στη συγκεκριμένη κατηγορία ταξινομούνται γλώσσες που επιτρέπουν την ανάπτυξη παράλληλων προγραμμάτων, όπου πολλές εντολές εκτελούνται ταυτόχρονα σε πολλούς υπολογιστές, έτσι ώστε το τελικό αποτέλεσμα να προκύψει γρηγορότερα. Οι παράλληλες γλώσσες προσφέρουν συνήθως εύκολους τρόπους επικοινωνίας

μεταξύ των νημάτων που εκτελούνται παράλληλα, καθώς και τρόπους ώστε να δημιουργούνται καινούριες παράλληλες εκτελέσεις. Παραδείγματα γλωσσών που ανήκουν (και) σε αυτή την κατηγορία είναι η Go, η Java, η Erlang, η MultiLisp ή η Cilk.

- Εκπαιδευτικά προγραμματιστικά περιβάλλοντα. Σε αυτή την κατηγορία ανήκουν εκπαιδευτικές γλώσσες προγραμματισμού οι οποίες απευθύνονται σε αρχάριους προγραμματιστές για την κατασκευή μικροεφαρμογών. Είναι κατάλληλες για την εκμάθηση προγραμματισμού σε μικρές ηλικίες. Παραδείγματα τέτοιων γλωσσών είναι η LOGO, το Game Maker και το App Inventor.

Τέλος, στην περίπτωση που η κατηγοριοποίηση γίνεται με βάση τον τρόπο που περιγράφεται το ζητούμενο, υπάρχουν οι παρακάτω κατηγορίες:

- Προστακτικές γλώσσες προγραμματισμού (imperative) είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα κατασκευαστικά, δίνοντας μια σειρά εντολών που όταν εκτελεστούν παράγουν το ζητούμενο αποτέλεσμα. Τέτοιες γλώσσες είναι η C, η Java αλλά και η OCaml.
- Δηλωτικές γλώσσες προγραμματισμού (declarative) είναι οι γλώσσες που περιγράφουν το ζητούμενο αποτέλεσμα χρησιμοποιώντας τις ιδιότητες που έχει, και όχι τον τρόπο με τον οποίο υπολογίζεται. Παραδείγματα δηλωτικών γλωσσών είναι η Haskell, η SQL και η Prolog.

Ως αποτέλεσμα δημιουργήθηκαν εκατομμύρια προγράμματα και συνεχίζουν να δημιουργούνται καθημερινά με σκοπό τη ταχύτερη και εύκολη λύση προβλημάτων του χρήστη. Αυτά τα προγράμματα χρειάζονται βέβαια ένα περιβάλλον ή αλλιώς λειτουργικό σύστημα για να εκτελέσουν εργασίες αλλά επιπλέον ορισμένα αποτελούν το ίδιο το σύνολο του λειτουργικού συστήματος που είναι απαραίτητο στον υπολογιστή για να ξεκινήσει, να επικοινωνήσει με το περιβάλλον του και να συντονίσει την λειτουργία του.

2.2 Λειτουργικά συστήματα

Λειτουργικό σύστημα ή ΛΣ (αγγλικά: Operating System ή OS) ονομάζεται στην επιστήμη της πληροφορικής το λογισμικό του υπολογιστή που είναι υπεύθυνο για τη διαχείριση και τον συντονισμό των εργασιών, καθώς και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει ένα θεμέλιο, ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή. Μια από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση του υλικού, απαλλάσσοντας έτσι το λογισμικό του χρήστη από τον άμεσο και επίπονο χειρισμό του υπολογιστή και καθιστώντας ευκολότερο τον προγραμματισμό τους. Σχεδόν όλοι οι υπολογιστές (παλάμης, επιτραπέζιοι, υπερυπολογιστές, ακόμη και παιχνιδομηχανές) χρησιμοποιούν έναν τύπο λειτουργικού συστήματος. Ορισμένα παλαιότερα μοντέλα ωστόσο βασίζονται σε ένα ενσωματωμένο λειτουργικό σύστημα, το οποίο περιέχεται σε έναν οπτικό δίσκο ή άλλες συσκευές αποθήκευσης δεδομένων.

Στις μέρες μας (2016), τα δημοφιλέστερα λειτουργικά συστήματα στους μικροϋπολογιστές, (συμπεριλαμβανομένων των προσωπικών υπολογιστών), έχουν διαμορφωθεί σε δύο μεγάλες οικογένειες: αυτή των Unix-συμβατών και την οικογένεια των Microsoft Windows. Οι κεντρικοί υπολογιστές και τα ενσωματωμένα συστήματα χρησιμοποιούν μια ποικιλία άλλων λειτουργικών συστημάτων, τα περισσότερα από τα οποία δεν έχουν άμεση συγγένεια με τα Windows ή με το Unix.

Τα *Unix-συμβατά* Λειτουργικά Συστήματα αποτελούν μια πολυποίκιλη ομάδα, με πολλές κύριες υποκατηγορίες συμπεριλαμβανομένων των System V, BSD, και GNU/Linux. Το εμπορικό σήμα Unix χρησιμοποιείται από πολλά ΛΣ που έχουν πολλά κοινά με το αρχικό Unix. Τα Unix Λ.Σ. τρέχουν σε μια μεγάλη γκάμα από αρχιτεκτονικές υπολογιστών. Χρησιμοποιούνται πολύ σαν συστήματα εξυπηρετητές στις επιχειρήσεις και σε σταθμούς εργασίας σε ακαδημαϊκούς και μηχανολογικούς χώρους εργασίας. Παραλλαγές του Unix που διακινούνται ως Ελεύθερο λογισμικό, όπως το GNU/Linux και BSD αυξάνουν σε δημοτικότητα στο χώρο των σταθμών εργασίας και των προσωπικών υπολογιστών. Παραλλαγές που διακινούνται με κλειστές άδειες χρήσης όπως

το HP-UX της Hewlett-Packard, το Irix της Silicon Graphics και το AIX της IBM έχουν σχεδιαστεί να τρέχουν μόνο στο υλικό των συγκεκριμένων εταιρειών ενώ άλλες παραλλαγές μπορούν να τρέξουν και σε προσωπικούς υπολογιστές. Το Solaris της Sun (που άλλαξε άδεια χρήσης σε λογισμικό ανοιχτού κώδικα υπό την CDDL άδεια) είναι ένα τέτοιο πολύμορφο αλλά αληθινό Unix και μπορεί να τρέχει στους σταθμούς εργασίας της Sun αλλά και στον μικρότερο υπολογιστή αρχιτεκτονικής x86. Το ΛΣ Mac OS X της Apple είναι μια παραλλαγή του BSD, και έχει αντικαταστήσει τα προηγούμενα (μη-unix) MacOS ΛΣ της Apple σε μια σχετικά μικρή αλλά αφοσιωμένη αγορά, έχοντας γίνει ένα πολύ δημοφιλές Unix.

Η οικογένεια των *Microsoft Windows* ΛΣ ξεκίνησε σαν ένα επίπεδο γραφικής διασύνδεσης πάνω από το παλιότερο MS-DOS περιβάλλον για τους IBM PC. Τα Windows τρέχουν πάνω σε 32- και 64-bit Intel και AMD υπολογιστές, αν και προηγούμενες εκδόσεις έτρεχαν και σε DEC Alpha, MIPS και PowerPC αρχιτεκτονικές (και υπήρξε και προσπάθεια να μεταφερθεί και σε αρχιτεκτονική SPARC).

Σήμερα, τα Windows είναι το δημοφιλέστερο ΛΣ προσωπικών υπολογιστών απολαμβάνοντας ένα σχεδόν μονοπώλιο του 90% του παγκόσμιου μεριδίου αγοράς των προσωπικών υπολογιστών¹. Επίσης χρησιμοποιείται ευρέως και στους εξυπηρετητές υποστηρίζοντας εφαρμογές όπως Web εξυπηρετητές (Web Servers) και εξυπηρετητές βάσεων δεδομένων (DBMS Servers).

Τα ΛΣ μεγάλων υπολογιστών, όπως της IBM z/OS και ενσωματωμένων ΛΣ όπως QNX, eCos, Symbian και Palm OS, είναι συνήθως άσχετα με το Unix και τα Windows. Τα ΛΣ Windows CE, Windows NT Embedded 4.0 και Windows XP Embedded σχετίζονται με τα Windows.

Παλιότερα ΛΣ που ακόμα χρησιμοποιούνται σε κλειστές αγορές περιλαμβάνουν το παρόμοιο με τα Windows OS/2 της IBM, το VMS της Hewlett-Packard (πρώην DEC), το Mac OS, το όχι-Unix προηγούμενο του Mac OS της Apple X και το AmigaOS, το πρώτο με γραφική διασύνδεση χρήστη ΛΣ με ανεπτυγμένες δυνατότητες πολυμέσων που έγινε διαθέσιμο στο κοινό. Παλαιότερο, επίσης, ΛΣ που χρησιμοποιείται ακόμη σε πολύ περιορισμένες εφαρμογές, είναι ο πρόγονος των Windows στους Προσωπικούς υπολογιστές,

το MS-DOS. Η έρευνα και η ανάπτυξη νέων τύπων ΛΣ συνεχίζεται και αποτελεί ένα ενεργό πεδίο της πληροφορικής.

Στη συνέχεια προέκυψε ένα μεγάλο πρόβλημα στις υπολογιστικές μηχανές που ήταν η μεταφορικότητά τους. Σταδιακά οι επεξεργαστές και όλα τα ζωτικά μέρη ενός υπολογιστή άρχισαν να καλύπτουν λιγότερο χώρο μέσα από χρόνιες έρευνες και δοκιμές με αποτέλεσμα να επέλθει η πρώτη επανάσταση στη τεχνολογία : τα Laptop(φορητοί υπολογιστές) . Ταυτόχρονα τα κινητά τηλέφωνα είχαν εδραιώσει τη θέση τους στη καθημερινότητα κάνοντάς τα όλο και πιο απαραίτητα . Σταδιακά τα πρώτα έξυπνα τηλέφωνα ήταν γεγονός.

Το μερίδιο αγοράς μεταξύ των κατασκευαστών smartphone δεν ισοδυναμεί με το μερίδιο αγοράς των λειτουργικών συστημάτων, δεδομένου ότι ορισμένα λειτουργικά συστήματα είναι διαθέσιμα μόνο προς έναν κατασκευαστή και άλλα έχουν λάβει άδεια από πολλούς κατασκευαστές. Το iPhone της Apple, το Symbian της Nokia και το BlackBerry της RIM είναι προς το παρόν διαθέσιμα μόνο από ενιαίους κατασκευαστές. Το λειτουργικό σύστημα Android της Google και τα λειτουργικά συστήματα για κινητά της Microsoft είναι πλατφόρμες που έχουν λάβει άδεια και χρησιμοποιούνται από διάφορους κατασκευαστές. Ως αποτέλεσμα, οι κατασκευαστές smartphones που χρησιμοποιούν εξουσιοδοτημένα λειτουργικά συστήματα διαμοιράζουν όλοι το συνολικό μερίδιο αγοράς του συγκεκριμένου λειτουργικού συστήματος μεταξύ τους, ενώ το συνολικό μερίδιο για ένα λειτουργικό σύστημα ενός ενιαίου κατασκευαστή κατέχει μόνον ο εν λόγω κατασκευαστής. Το λειτουργικό σύστημα Symbian της Nokia ήταν διαθέσιμο στο παρελθόν από διάφορους κατασκευαστές υπό εξουσιοδοτημένο μοντέλο, αργότερα κατά κύριο λόγο μόνο από τη Nokia.

2.3 Smartphones/Εξυπνα τηλέφωνα

Το smartphone ή με τον ελληνικό όρο έξυπνο τηλέφωνο, είναι ένα κινητό τηλέφωνο βασισμένο σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερο προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα συμβατικό κινητό τηλέφωνο. Τα πρώτα smartphones συνδύαζαν τις λειτουργίες ενός προσωπικού ψηφιακού βοηθού (PDA) και ενός κινητού τηλεφώνου. Σε μεταγενέστερα μοντέλα προστέθηκαν οι λειτουργίες των φορητών media players, low-end compact ψηφιακές φωτογραφικές μηχανές, βιντεοκάμερες τσέπης, καθώς και μονάδες πλοήγησης GPS, με αποτέλεσμα την διαμόρφωση μιας πολυχρηστικής συσκευής. Πολλά σύγχρονα smartphones περιλαμβάνουν επίσης οθόνες αφής υψηλής ανάλυσης και web browsers που εμφανίζουν τυποποιημένες ιστοσελίδες, καθώς και βελτιστοποιημένες ιστοσελίδες για κινητά. Η πρόσβαση σε δεδομένα υψηλής ταχύτητας παρέχεται μέσω Wi-Fi και μέσω κινητών ευρυζωνικών υπηρεσιών. Τα τελευταία χρόνια, η ταχεία ανάπτυξη στην αγορά των εφαρμογών για κινητά και στο εμπόριο κινητών τηλεφώνων έχει γίνει οδηγός για την ευρεία υιοθέτηση των smartphones.

Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων που χρησιμοποιούνται από τα σύγχρονα smartphones περιλαμβάνουν, μεταξύ άλλων, το Android της Google, το iOS της Apple, το Symbian της Nokia, το BlackBerry OS της RIM, το Bada της Samsung, τα Windows Phone της Microsoft, το webOS της Hewlett-Packard, καθώς και ενσωματωμένες διανομές Linux όπως το Maemo και το MeeGo. Τέτοιου είδους λειτουργικά συστήματα μπορούν να εγκατασταθούν σε πολλά διαφορετικά μοντέλα κινητών τηλεφώνων και συνήθως κάθε συσκευή μπορεί να λάβει πολλές ενημερωμένες εκδόσεις λογισμικού λειτουργικού συστήματος κατά τη διάρκεια ζωής της.

Πρόελευση του όρου

Παρότι οι συσκευές που συνδυάζουν τηλεφωνία και πληροφορική είχαν ήδη επινοηθεί από το 1973 και διατέθηκαν προς πώληση το 1994, ο όρος *smartphone* δεν εμφανίστηκε μέχρι το 1997, όταν η Ericsson περιέγραψε το GS 88 "Penelope" ως *Smart Phone (Εξυπνο Τηλέφωνο)*.



Σχήμα 2.1 Το πρώτο Smartphone

Η διάκριση μεταξύ των smartphones και των απλών κινητών τηλεφώνων μπορεί να είναι ασαφής και δεν υπάρχει επίσημος ορισμός για το ποιες είναι οι μεταξύ τους διαφορές. Μία από τις πιο σημαντικές είναι ότι οι προηγμένες διεπαφές προγραμματισμού εφαρμογών (APIs) στα smartphones σχετικά με τη λειτουργία τρίτων εφαρμογών μπορούν να επιτρέψουν σε αυτές τις εφαρμογές και να έχουν καλύτερη ενσωμάτωση στο λειτουργικό σύστημα και στο hardware του τηλεφώνου απ' ό,τι συμβαίνει συνήθως στα απλά κινητά τηλέφωνα. Συγκριτικά, τα συμβατικά κινητά τηλέφωνα τρέχουν συχνότερα σε ιδιόκτητο firmware, με υποστήριξη λογισμικού από τρίτους, μέσα από πλατφόρμες όπως το Java ME ή το BREW. Μια επιπλέον δυσκολία στη διάκριση μεταξύ smartphones και συμβατικών κινητών τηλεφώνων είναι ότι, με την πάροδο του χρόνου, οι δυνατότητες των νέων μοντέλων των απλών κινητών τηλεφώνων μπορούν να υπερβούν εκείνες των τηλεφώνων που είχαν προωθηθεί ως smartphones στο παρελθόν. Ορισμένοι κατασκευαστές και πάροχοι χρησιμοποιούν τον όρο superphone για τα υψηλής απόδοσης τηλέφωνα τους με τις ασυνήθιστα

μεγάλες οθόνες και τα άλλα ακριβά χαρακτηριστικά τους. Άλλοι προτιμούν τον όρο «rphablet», αναγνωρίζοντας τη σύγκλισή τους με τους υπολογιστές tablet χαμηλής απόδοσης. Η κινητή βιομηχανία έφερε στο προσκήνιο πολλά νέα μοντέλα έξυπνων τηλεφώνων όμως επανάσταση έφερε στον τεχνολογικό κόσμο η άφιξη του iPhone. Τον Ιανουάριο του 2007 η Apple ανακοίνωσε ένα νέο μοντέλο κινητού τηλεφώνου το οποίο είχε μεγάλη έγχρωμη οθόνη με χωρητική ψηφιοποίηση και φιλική διεπαφή με τα δάχτυλα του χρήστη (touchscreen). Τον Σεπτέμβριο του 2008, η Google απάντησε στην iOS με το δικό της κινητό λειτουργικό σύστημα, το Android OS. Η εταιρία Partner HTC κατασκεύασε το πρώτο Android-powered smartphone για την T-Mobile, το G1, επίσης γνωστή ως όνειρο. Μέχρι και σήμερα ο ανταγωνισμός των iOS και Android προϊόντων στην αγορά είναι 'σκληρός' κάνοντας τους δυο κολοσσούς να παρουσιάζουν καινοτόμες συσκευές και λογισμικό ανά τακτά χρονικά διαστήματα με στόχο μεγαλύτερη μερίδα της αγοράς.

Κατά το 1ο τρίμηνο του 2012, μετά από 14 χρόνια στην αγορά η Samsung βρέθηκε στην πρώτη θέση με πωλήσεις 44,5 εκατ. smartphones ή 30,6% του μεριδίου αγοράς, ενώ πωλήθηκαν 35,1 εκατ. iPhone ή 24,1% του μεριδίου αγοράς.



Σχήμα 2.2 Apple vs Android

2.4 Λειτουργικό σύστημα Android

Ο όρος **Android** έχει ελληνική προέλευση καθώς προέρχεται από την λέξη ανδρ- που έχει την έννοια του άνδρα ή του ανθρώπου και την κατάληξη –ειδές που χαρακτηρίζει κάποιο είδος. Συνεπώς η έννοια που δίδεται στην λέξη Android είναι το <<Ανδροειδές>> και συμβολίζει το ρομπότ με μορφή ανθρώπου.

Το Android είναι ένα λειτουργικό σύστημα βασισμένο στον πυρήνα του Linux και έχει σχεδιαστεί κυρίως για φορητές συσκευές με οθόνη αφής, όπως smartphones και υπολογιστές tablet. Αρχικά αναπτύχθηκε από την Android Inc., την οποία υποστηρίζει οικονομικά και αργότερα αγόρασε η Google το 2005. Το Android παρουσιάστηκε το 2007 μαζί με την ίδρυση του Open Handset Alliance: μια κοινοπραξία εταιριών υλικού, λογισμικού και τηλεπικοινωνιών που προωθούσαν την εγκαθίδρυση των ανοικτών προτύπων για τις κινητές συσκευές. Το πρώτο δημόσιο διαθέσιμο smartphone που έτρεχε Android, ήταν το HTC Dream, που κυκλοφόρησε στις 22 Οκτωβρίου 2008. Η διεπαφή χρήστη του Android βασίζεται σε άμεσο χειρισμό, με χρήση διάφορων μοτίβων αφής που αντιστοιχούν στον πραγματικό κόσμο σε ενέργειες όπως σύρσιμο, χτύπημα, τσίμπημα και αντίστροφο τσίμπημα για να διαχειριστούν αντικείμενα στην οθόνη. Εσωτερικοί αισθητήρες, όπως επιταχυνσιόμετρα, δέκτες GPS, γυροσκόπια και αισθητήρες εγγύτητας χρησιμοποιούνται από ορισμένες εφαρμογές για να ανταποκριθούν στις πρόσθετες ενέργειες του χρήστη, για παράδειγμα, για την προσαρμογή της οθόνης από κατακόρυφο σε οριζόντιο προσανατολισμό, ανάλογα με το πώς η συσκευή είναι προσανατολισμένη. Το Android επιτρέπει στους χρήστες να προσαρμόσουν τις αρχικές τους οθόνες με τις συντομεύσεις, σε εφαρμογές και widgets, τα οποία επιτρέπουν στους χρήστες να εμφανίζουν ζωντανό περιεχόμενο, όπως μηνύματα ηλεκτρονικού ταχυδρομείου και πληροφορίες για τον καιρό, απευθείας στην αρχική οθόνη. Οι εφαρμογές μπορούν να στείλουν περαιτέρω κοινοποιήσεις προς τον χρήστη για να τον ενημερώσουν για σχετικές πληροφορίες, όπως νέα μηνύματα ηλεκτρονικού ταχυδρομείου και μηνύματα κειμένου (SMS). Ο πηγαίος κώδικας του Android δίνεται από την Google με άδεια χρήσης Apache, μια άδεια που επιτρέπει το λογισμικό να τροποποιηθεί ελεύθερα και θα διανεμηθεί από τους κατασκευαστές εφαρμογών και τους ενθουσιώδης προγραμματιστές. Μέχρι τον Ιούλιο του 2013, το Android

έχει το μεγαλύτερο αριθμό εφαρμογών ("apps") που είναι διαθέσιμα για download στο κατάστημα Google Play, το οποίο περιείχε πάνω από 1 εκατομμύριο εφαρμογές δημοσιευμένες και πάνω από 50 δισεκατομμύρια downloads. Σύμφωνα με μια έρευνα που πραγματοποιήθηκε τον Απρίλιο του 2013 διαπιστώθηκε ότι το Android είναι η πιο διαδεδομένη πλατφόρμα μεταξύ των προγραμματιστών και το χρησιμοποιούν το 71% απο αυτούς.



Σχήμα 2.3 Λογότυπο πλατφόρμας Android

Παρακάτω βλέπουμε την σειρά των λειτουργικών συστημάτων Android τηλεφώνων από την δημιουργία τους μέχρι και σήμερα(Αύξουσα Ταξινόμηση):

Έκδοση :	Κωδική Ονομασία :	Ημερομηνία :	API level :	Διανομή [16] :
7.1	<i>Nougat</i>	4 Οκτωβρίου 2016	25	
7.0	<i>Nougat</i>	22 Αυγούστου 2016	24	0.3%

Έκδοση :	Κωδική Ονομασία :	Ημερομηνία :	API level :	Διανομή [16] :
6.0	<i>Marshmallow</i>	5 Οκτωβρίου 2015	23	24.0%
5.1	<i>Lollipop</i>	9 Μαρτίου 2015	22	22.8%
5.0	<i>Lollipop</i>	3 Νοεμβρίου 2014	21	11.3%
4.4	<i>KitKat</i>	31 Οκτωβρίου 2013	19	25.2%
4.3	<i>Jelly Bean</i>	24 Ιουλίου 2013	18	2.0%
4.2	<i>Jelly Bean</i>	13 Νοεμβρίου 2012	17	6.8%
4.1	<i>Jelly Bean</i>	9 Ιουλίου 2012	16	4.9%
4.0	<i>Ice Cream Sandwich</i>	16 Δεκεμβρίου 2011	15	1.3%
3.2	<i>Honeycomb</i>	15 Ιουλίου 2011	13	0%
2.3.2 +	<i>Gingerbread</i>	9 Φεβρουαρίου 2011	10	1.3%
2.2	<i>Froyo</i>	20 Μαΐου 2010	8	0.1%
2.0	<i>Eclair</i>	26 Οκτωβρίου 2009	7	-

Έκδοση :	Κωδική Ονομασία :	Ημερομηνία :	API level :	Διανομή [16] :
1.6	<i>Donut</i>	15 Σεπτεμβρίου 2009	4	-

Πίνακας 2.4 Εκδόσεις Android

Οι Android εφαρμογές είναι συνήθως γραμμένες σε Java καθώς είναι η πιο διάσημη γλώσσα προγραμματισμού στον κόσμο σήμερα και χαρακτηρίζεται ως αντικειμενοστραφής γλώσσα (object-oriented). Για όσους έχουν προηγούμενη εμπειρία με την φιλοσοφία του object-oriented programming σε άλλες γλώσσες, όπως C#, Objective C ή Ruby τότε δεν θα αντιμετωπίσουν κάποιο ιδιαίτερο πρόβλημα στη μετάβαση σε Java.

Η Google παρέχει το [Native Development Kit \(NDK\)](#) το οποίο επιτρέπει στους developer να γράψουν σημαντικό μέρος των εφαρμογών τους σε native γλώσσες όπως είναι η C και η C++. Αυτό είναι ιδιαίτερα χρήσιμο καθώς μας επιτρέπει να επαναχρησιμοποιήσουμε τον υπάρχων κώδικα (δηλαδή κάποιες γνωστές βιβλιοθήκες) ή να βελτιώσουμε την κατανάλωση πόρων της εφαρμογής που φτιάχνουμε κάνοντας το μέγιστο δυνατό optimization (βελτιστοποίηση) στα σημεία που χρειάζεται.

Υπάρχουν κάποια εργαλεία τα οποία μας επιτρέπουν να γράψουμε εφαρμογές χρησιμοποιώντας και web γλώσσες όπως η HTML, η CSS και η Javascript. Με αυτά μπορούμε να δημιουργήσουμε εφαρμογές που θα τρέχουν μέσα σε ένα WebView, δηλαδή μία πλήρης web page που τρέχει μέσα σε μία εφαρμογή αντί στον browser. Εργαλεία όπως τα [PhoneGap](#) και [Appcelerator Titanium](#) μας βοηθούν να γράψουμε κώδικα ο οποίος μεταγλωττίζεται στο παρασκήνιο σε native code. Υπάρχουν μειονεκτήματα και πλεονεκτήματα σ' αυτή την προσέγγιση και θα πρέπει να έχουμε πάντα υπόψιν μας ότι χρησιμοποιώντας Java παίρνουμε τα καλύτερα αποτελέσματα, αλλά μπορούμε να φτιάξουμε εξίσου δυνατές εφαρμογές με όποιο εργαλείο μας βολεύει. Ένα ακόμη μειονέκτημα των άλλων γλωσσών προγραμματισμού είναι ότι τα 3rd party εργαλεία που χρησιμοποιούμε

θα βρίσκονται πάντα ένα βήμα πίσω από τις τελευταίες εξελίξεις στον χώρο του Android development.

Μια εκ των εφαρμογών αυτών είναι και το Android Studio όπου θα μάθουμε στην επόμενη ενότητα.

2.4.1 Αναλυτικά οι εκδόσεις του Android

Η μακρά ιστορία του **Android** ξεπερνά τα 7 χρόνια... Η αρχή έγινε με το **HTC Dream** και από τότε έχουν κατασκευαστεί χιλιάδες Android συσκευές. Μέσα σε αυτό το χρονικό διάστημα η **Google** έχει διαθέσει **10** ξεχωριστές εκδόσεις, με την κάθε μια να έχει προσφέρει κάτι δικό της στην ανάπτυξη του πιο γρήγορα αναπτυσσόμενου λειτουργικού στο κόσμο. Έχοντας γίνει πλέον ένα ώριμο λειτουργικό, ο ρυθμός με τον οποίο οι νέες εκδόσεις κυκλοφορούν έχει **επιβραδυνθεί**, δίνοντας έτσι χρόνο στους κατασκευαστές, να **προσθέσουν** και δικά τους χαρακτηριστικά αλλά και να **αναπτύξουν** το δικό τους UI για τις συσκευές τους. Φυσικά, το μεγάλο αυτό χρονικό περιθώριο, βοηθάει και την **Google** να εστιάσει σε βαθύτερες αλλαγές, όπως για παράδειγμα το **Material Design** στο Lollipop.

Παρακάτω έχουμε συγκεντρωμένες όλες αυτές τις σημαντικές εκδόσεις του **Android** που εισήγαγαν ένα νέο χαρακτηριστικό, το οποίο συνέβαλε στην ανάπτυξη του διάσημου αυτού λειτουργικού.

Android 1.5 Cupcake

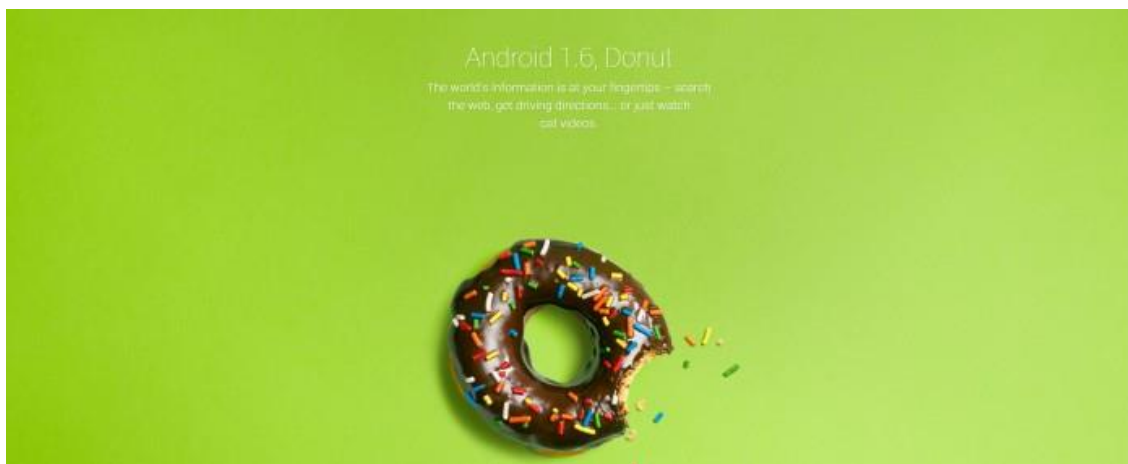
Αρχικά να αναφέρουμε πως θα ήταν ψέμα να πούμε πως το **Android 1.5 Cupcake** ή οποιαδήποτε άλλη έκδοση του Android είχε μια και μοναδική βελτίωση, αλλά αν πρέπει να επιλέξουμε το χαρακτηριστικό που το ανέδειξε, είναι η εισαγωγή ψηφιακού πληκτρολογίου. Ίσως σας φαίνεται περίεργο, αλλά το 2008/2009, τα περισσότερα smartphones είχαν φυσικό πληκτρολόγιο QWERTY.



Σχήμα 2.4 Android 1.5,Cupcake

Android 1.6 Donut

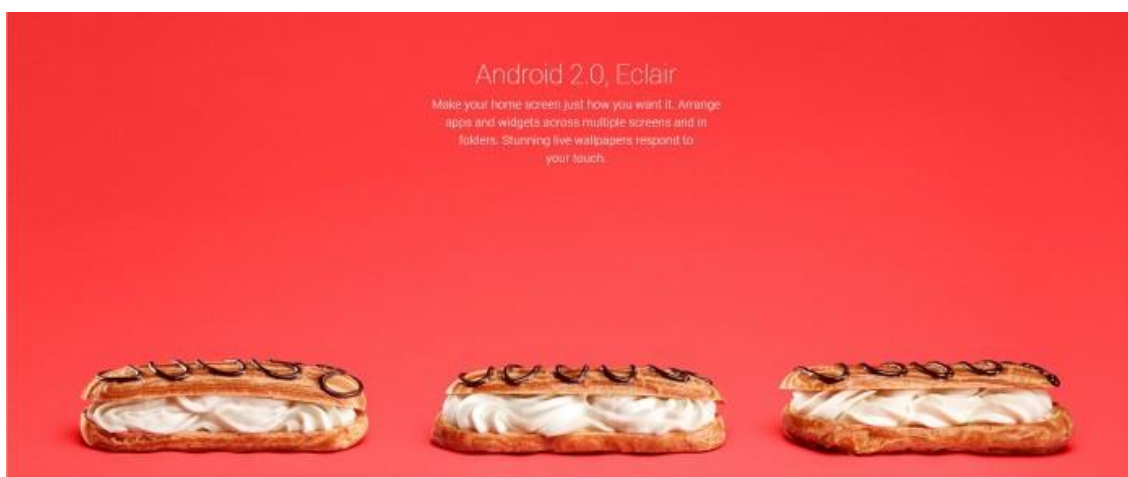
Το **Android 1.6 Donut**, είχε αλλαγές στο εσωτερικό του λειτουργικού και αυτό είχε ως αποτέλεσμα να υποτιμηθεί αρκετά από τους καταναλωτές. Παρόλα αυτά, ήταν πολύ σημαντική αναβάθμιση, καθώς έφερε υποστήριξη διαφορετικών αναλύσεων οθόνης, ανεξάρτητα από την πυκνότητα pixels, κάτι που καθόρισε πραγματικά το μέλλον των Android συσκευών.



Σχήμα 2.5 Android 1.6,Donut

Android 2.0 Eclair

Ξεχωρίζοντας ένα μόνο από τα χαρακτηριστικά του **Eclair**, είναι αρκετά δύσκολο, αλλά αν πρέπει να επιλέξουμε κάτι, αυτό είναι οι βελτιώσεις στην κάμερα. Μέχρι τότε το Android δεν είχε υποστήριξη LED flash στην κάμερα, ούτε δυνατότητες που σήμερα θεωρούμε δεδομένες, όπως χρωματικά εφέ, λειτουργία σκηνών, λειτουργίες εστίασης, ή ακόμη και ρύθμιση της ισορροπίας λευκού.



Σχήμα 2.6 Android 2.0,Eclair

Android 2.2 Froyo

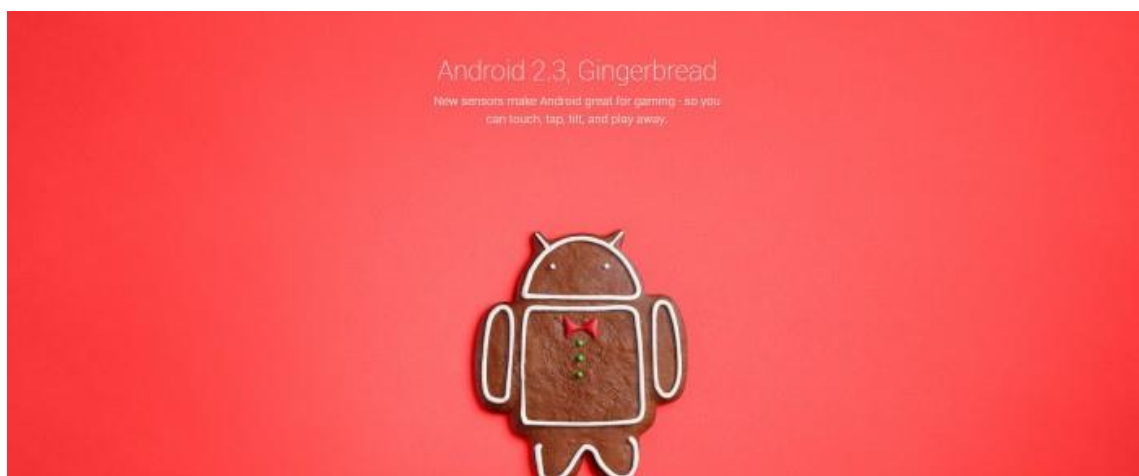
Για το **Froyo** έχουμε να επιλέξουμε ανάμεσα σε δύο χαρακτηριστικά και αυτά είναι η ταχύτητα και υποστήριξη για Wi-Fi hotspots. Αλλά δεδομένου ότι

πρέπει να επιλέξουμε μόνο ένα, θα πρέπει να πάμε με την ταχύτητα, διότι το Wi-Fi Hotspot δεν ήταν διαθέσιμο παγκοσμίως. Στο Froyo λοιπόν, ορίστηκε σαν runtime ο Dalvic, που έφερε σοβαρές βελτιώσεις στην απόδοση της συσκευής μας, που σύμφωνα με την Google γίνεται από 2 έως 5 φορές ταχύτερη.



Σχήμα 2.7: Android 2.2, Froyo

Το **Android 2.3 Gingerbread**, σίγουρα ήταν μία από τις πιο διάσημες εκδόσεις του λειτουργικού και με την προσθήκη υποστήριξης για περισσότερους αισθητήρες, όπως το βαρόμετρο και γυροσκόπιο βοήθησε πραγματικά στην εξέλιξη του Android. Πριν το Gingerbread, με την έλλειψη πολλών αισθητήρων, υπήρχε σοβαρός περιορισμός στο τι μπορούσαμε να κάνουμε με το κινητό μας και ειδικότερα στα παιχνίδια.



Σχήμα 2.8: Android 2.3, Gingerbread

Android 3.0 Honeycomb

Χωρίς αμφιβολία, το **Honeycomb**, ήταν μια από τις λιγότερο δημοφιλείς εκδόσεις του Android, αλλά αυτό σίγουρα δεν έχει να κάνει με την συνεισφορά του στην πλατφόρμα. Το Android, είχε φρική εμπειρία χρήσεως στα tablets, κάτι που άλλαξε με την κυκλοφορία του Honeycomb, όπου ήταν σχεδιασμένο για συσκευές με μεγαλύτερη οθόνη.



Σχήμα 2.9: Android 3.0, Honeycomb

Android 4.0 Ice Cream Sandwich

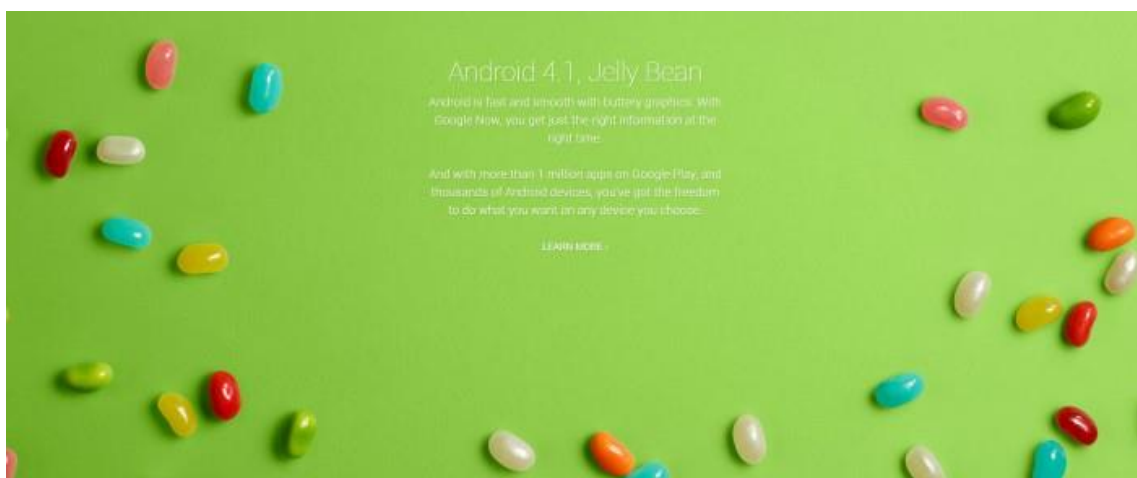
Το **Ice Cream Sandwich**, έφερε πραγματικά τεράστια ανανέωση στον σχεδιασμό του Android με το Holo UI. Το ICS, ήταν η πρώτη έκδοση του Android που έδωσε μοντέρνα χαρακτηριστικά στο λειτουργικό, που ακόμη και σήμερα συναντάμε πολλά στοιχεία του στις Android συσκευές.



Σχήμα 3.0: Android 4.0, Ice Cream Sandwich

Android 4.1 Jelly Bean

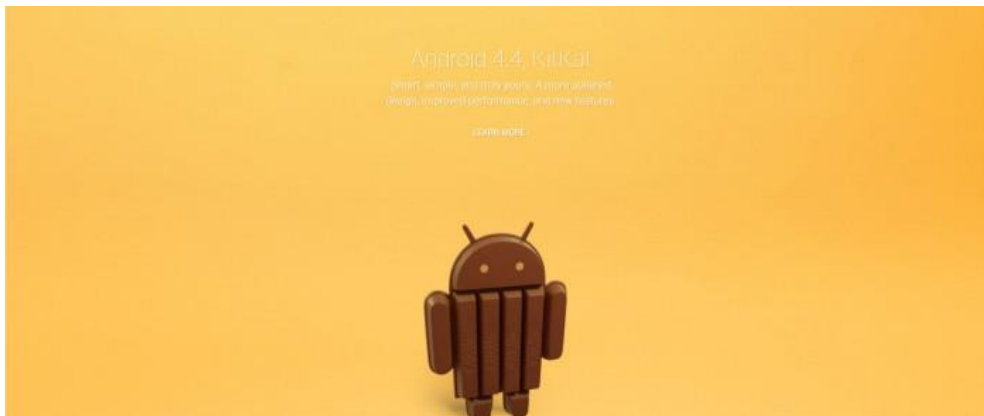
Το **Jelly Bean** είναι μεγάλη υπόθεση για πολλούς λόγους, ιδιαίτερα αν σκεφτούμε ότι κατέχει περίπου το 50% των Android συσκευών. Οι συνεισφορές του στο Android είναι πολλές, αλλά αυτή που ξεχώρισε και αναπτύσσεται μέχρι και σήμερα, είναι το Google Now. Ο ψηφιακός βοηθός της Google όχι μόνο παρέχει γρήγορες απαντήσεις για τα πάντα, αλλά και εμφανίζει νέα και ειδήσεις σύμφωνα με τα ενδιαφέροντα μας. Το Google Now έχει οργανωμένες κάρτες, που μας δίνουν πληροφορίες σχετικές με την ημέρα, τον καιρό αλλά και την περιοχή που βρισκόμαστε.



Σχήμα 3.1: Android 4.1, Jelly Bean

Android 4.4 KitKat

Με το **KitKat**, η Google όχι απλά ενίσχυσε το UI, αλλά και ανέδειξε την δύναμη του λειτουργικού ακόμη και σε φθηνές συσκευές. Χάρη στις πολλές βελτιώσεις επιδόσεων που έγιναν στο KitKat, η Google υποστήριξε πως ακόμη και συσκευές με μόλις 512MB RAM θα είναι σε θέση να τρέξουν το KitKat απροβλημάτιστα. Αυτό ήταν μια αρκετά σοβαρή υπόθεση, δεδομένου ότι οι εταιρείες έδειχναν αδιαφορία στις οικονομικές συσκευές. Έτσι πλέον, δίνεται η δυνατότητα και σε αναπτυσσόμενες χώρες να αποκτήσουν Android συσκευές και κατά συνέπεια να αυξηθούν και οι πωλήσεις φθηνότερων συσκευών.



Σχήμα 3.1: Android 4.4, KitKat

Android 5.0 Lollipop

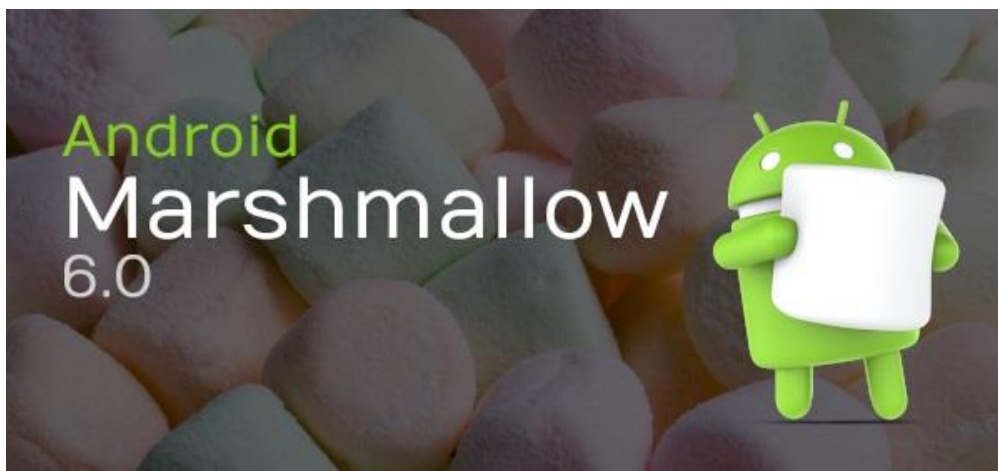
Σειρά έχει το **Lollipop** μία πολύ ελκυστική έκδοση του Android . Με το Material Design της Google, η εμφάνιση του λειτουργικού έγινε επίπεδη, με περισσότερα χρώματα και φυσικά με πάρα πολλά εφέ κίνησης.



Σχήμα 3.2: Android 5.0, Lollipop

Android 6.0 Marshmallow

Σήμερα χρησιμοποιείται ευρέως το Marshmallow με δυνατότητα αναγνώρισης δακτυλικού αποτυπώματος με σκοπό το ξεκλείδωμα της συσκευής ή ακόμα και την πληρωμή λογαριασμών ,μια βοήθεια από τη Google βασισμένη στην ενεργό εφαρμογή (Now On Tap) και τη Παρακολούθηση Συγχρονισμού των εφαρμογών της Συσκευής για εξοικονόμηση ενέργειας.



Σχήμα 3.3: Android 6.0, Marshmallow

Android 7.0 Nougat

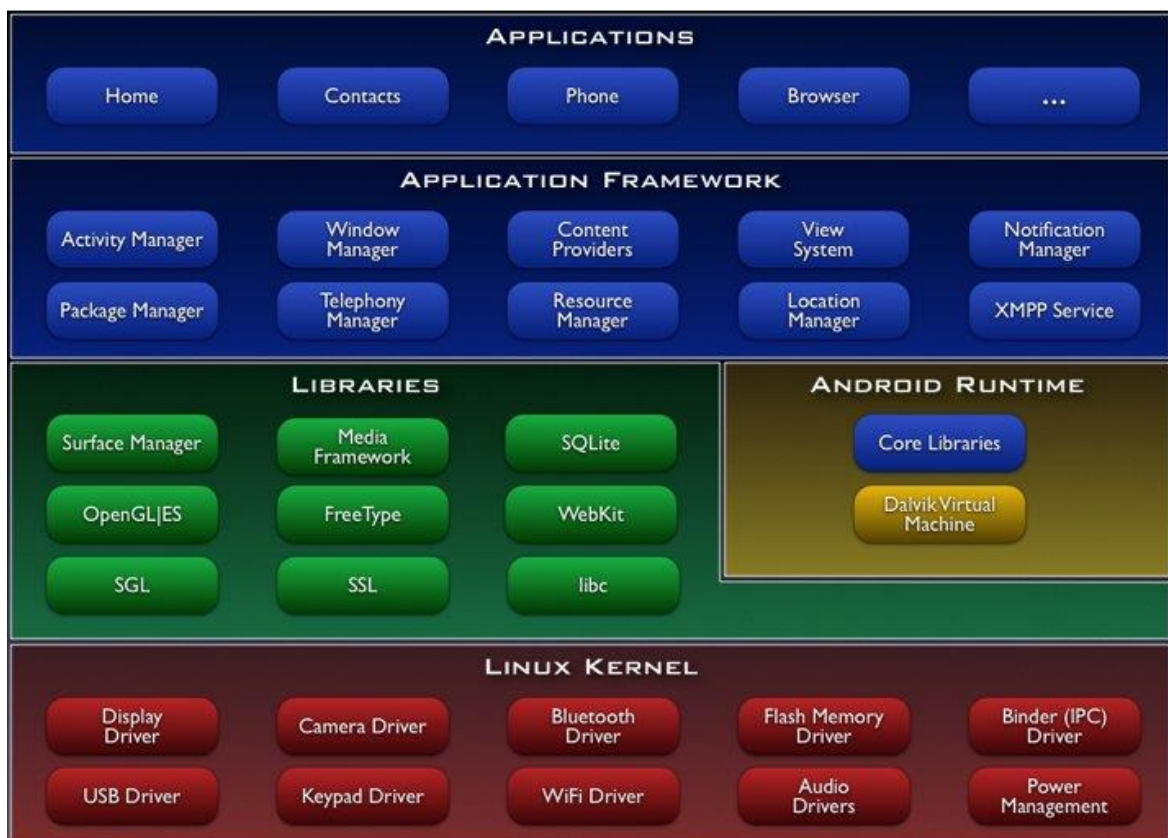
Τέλος φτάνουμε στο Nougat που θα κυκλοφορήσει ολοκληρωμένη έκδοση από τον Γενάρη του 2017 και αναμένεται να καλύψει πολλές από τις απαιτήσεις των χρηστών αλλά και όποιο κενό εμφάνισης και λειτουργικότητας είχε παρουσιαστεί στο Marshmallow .



Σχήμα 3.2: Android 7.0, Nougat

2.5 Δομή Λειτουργικού συστήματος

Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος με τον οποίο είναι δομημένο το λειτουργικό σύστημα Android.



Σχήμα 3.3 Αρχιτεκτονική του Android

Από ότι βλέπουμε λοιπόν η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 5 βασικά επίπεδα.

- ❖ **Τον πυρήνα Linux (Linux Kernel)** - Ο πυρήνας Linux είναι μια ευέλικτη και ασφαλής πλατφόρμα προσαρμοζόμενη σε διαφορετικές αρχιτεκτονικές. Αποτελείται από τους οδηγούς του υλικού, τη διαχείριση της μνήμης και της τροφοδοσίας παρέχοντας διασύνδεση με το υπόλοιπο λειτουργικό. Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει τα παρακάτω:
 - Οδηγό προβολής οθόνης
 - Οδηγό κάμερας
 - Οδηγό Wifi και Bluetooth
- ❖ **Τις βιβλιοθήκες (Libraries)** – Οι ενσωματωμένες βιβλιοθήκες είναι βιβλιοθήκες C/C++ οι οποίες απαιτούνται για τα δύο επόμενα επίπεδα που αφορούν τις εφαρμογές. Αυτές ουσιαστικά αποτελούν τα APIs που είναι διαθέσιμα στους προγραμματιστές για την ανάπτυξη των εφαρμογών. Οι βιβλιοθήκες από μόνες τους δεν αποτελούν εφαρμογές αλλά ενσωματώνονται και χρησιμοποιούνται από τις εφαρμογές για τις διάφορες λειτουργίες που παρέχει η καθεμία από αυτές. Ουσιαστικά αποτελούν ένα από τα δομικά υλικά των εφαρμογών, και άρα είναι αναπόσπαστο κομμάτι τους. Οι βιβλιοθήκες αυτές αφορούν τα WebKit , SWLite, OpenGL, OpenSSL, media. Μερικές από τις κύριες βιβλιοθήκες του Android είναι:
 - System C library – μια ενσωμάτωση της standard βιβλιοθήκης συστήματος της C (libc) τροποποιημένη για κινητές συσκευές βασισμένες στο Linux.
 - SGL – η γνωστή μηχανή δισδιάστατων γραφικών.
 - Βιβλιοθήκες 3D – μια υλοποίηση βασισμένη στα APIs του OpenGL ES. Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια υψηλά βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.

- Βιβλιοθήκες Πολυμέσων – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG.

- ❖ **Η μηχανή εκτέλεσης (Android Runtime)** – Οι βιβλιοθήκες πυρήνα και η εικονική μηχανή Dalvik αποτελούν τη μηχανή εκτέλεσης των εφαρμογών και είναι ειδικά σχεδιασμένα για κινητές συσκευές. Η σχεδίαση τους έγινε λαμβάνοντας υπ’ όψη τους δύο περιορισμούς που ισχύουν στις κινητές συσκευές: την επεξεργαστική ισχύ και την τροφοδοσία ή την αυτονομία. Εκτός από τις βιβλιοθήκες Android περιέχεται ένα υποσύνολο βιβλιοθηκών από την Java Standard Edition και Java Mobile Edition. Η Dalvik VM, σχεδιασμένη από τον Dan Borenstein για την Google, δεν εκτελεί .class αρχεία της Java αλλά .dex (Dalvik EXecutable Files) τα οποία ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση. Έτσι οι εφαρμογές που είναι γραμμένες σε Java, μεταγλωττίζονται σε .class αρχεία και ακολουθώντας σε dex αρχεία χάρη στο “dx” εργαλείο που είναι ενσωματωμένο στο Android SDK. Τέλος το Android είναι από τη φύση του multitasking λειτουργικό σύστημα και για αυτό επιτρέπει στις εφαρμογές του να τρέχουν σε πολλά νήματα ταυτόχρονα και να απασχολούν πολλές διαδικασίες εάν αυτό είναι αναγκαίο. Για να γίνει αυτό εφικτό η μηχανή Dalvik είναι σχεδιασμένη για να έχει ελάχιστο αντίκτυπο στη χρήση της μνήμης. Χάρη στον λιτό της σχεδιασμό, το σύστημα είναι σε θέση να τρέχει πολλές εικονικές μηχανές ταυτόχρονα.

- ❖ **Υποδομή Εφαρμογών (Application Framework)** – Η υποδομή εφαρμογών περιέχει τις κλάσεις με τη χρήση των οποίων είναι δυνατή η δημιουργία εφαρμογών. Το επίπεδο αυτό περιέχει ακόμα διαχειριστές υπεύθυνους για τον έλεγχο του κύκλου ζωής μια εφαρμογής, για την πρόσβαση σε κοινά δεδομένα, την διαχείριση των εγκατεστημένων εφαρμογών και των πόρων της συσκευής. Τα σημαντικότερα δομικά στοιχεία της υποδομής εφαρμογών είναι:
 - Πάροχος Περιεχομένου (Content Provider) – Δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να

ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.

- Σύστημα προβολών (View System) – Αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (ListView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου (TextView), κουμπιά, κλπ.
 - Διαχειριστής Πόρων (Resource Manager) – Παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.
 - Διαχειριστής Ειδοποιήσεων (Notification Manager) – Δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης.
 - Διαχειριστής Δραστηριοτήτων (Activity Manager) – Διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών.
- ❖ Και τέλος οι **εφαρμογές (Application Layer)** – Οι ενσωματωμένες στις συσκευές εφαρμογές ή οι εφαρμογές που αναπτύσσονται από τρίτους. Οι εφαρμογές αυτές χρησιμοποιούν τις βιβλιοθήκες του προηγούμενου επιπέδου.

2.6 Βασικές αρχές μιας εφαρμογής Android

Οι Android εφαρμογές είναι γραμμένες στη γλώσσα προγραμματισμού Java. Τα εργαλεία του Android SDK καταρτίζουν τον κώδικα σας-μαζί με τα δεδομένα και τα αρχεία των πόρων-σε ένα APK: ένα πακέτο Android, το οποίο είναι ένα συμπιεσμένο αρχείο με κατάληξη .apk . Ένα αρχείο APK περιέχει όλα τα

περιεχόμενα μιάς Android εφαρμογής και είναι το αρχείο που χρησιμοποιούν οι Android συσκευές για να εγκαταστήσουν την εφαρμογή. Μόλις εγκατασταθεί σε μια συσκευή, κάθε Android εφαρμογή 'ζει' μέσα στη δική του sandbox ασφάλεια:

- Το λειτουργικό σύστημα του Android είναι ένα multi-user λειτουργικό σύστημα Linux, στο οποίο κάθε εφαρμογή είναι και ένας διαφορετικός χρήστης.
- Από προεπιλογή, το σύστημα εκχωρεί σε κάθε εφαρμογή ένα μοναδικό αναγνωριστικό χρήστη του Linux (το αναγνωριστικό χρησιμοποιείται μόνο από το σύστημα και είναι άγνωστο προς τις εφαρμογές). Το σύστημα ορίζει δικαιώματα για όλα τα αρχεία σε μια εφαρμογή, έτσι ώστε μόνο το αναγνωριστικό χρήστη που έχει εκχωρηθεί σε αυτή την εφαρμογή να μπορεί να έχει πρόσβαση σε αυτά.
- Κάθε διαδικασία έχει τη δική της εικονική μηχανή (VM), έτσι ώστε ο κώδικας του app να τρέχει σε απομόνωση από άλλες εφαρμογές.

Με τον τρόπο αυτό, το σύστημα Android εφαρμόζει την αρχή των ελάχιστων προνομίων. Δηλαδή, κάθε εφαρμογή, από προεπιλογή, έχει πρόσβαση μόνο στα στοιχεία που χρειάζεται για να κάνει τη δουλειά της και όχι περισσότερα. Αυτό δημιουργεί ένα πολύ ασφαλές περιβάλλον, στο οποίο μία εφαρμογή δεν μπορεί να έχει πρόσβαση στα μέρη του συστήματος, για τα οποία δεν έχει δοθεί άδεια. Ωστόσο, υπάρχουν τρόποι για μια εφαρμογή, να ανταλλάξει δεδομένα με άλλες εφαρμογές και για μια εφαρμογή να έχει πρόσβαση στις υπηρεσίες του συστήματος:

- I. Μια εφαρμογή μπορεί να ζητήσει την άδεια για να έχει πρόσβαση στα δεδομένα της συσκευής, όπως επαφές του χρήστη, μηνύματα SMS, αποθηκευτικό χώρο (κάρτα SD), φωτογραφική μηχανή, Bluetooth, και πολλά άλλα. Όλα τα δικαιώματα των εφαρμογών πρέπει να χορηγούνται από το χρήστη κατά τη διάρκεια της εγκατάστασης.

2.7 Συστατικά Android εφαρμογών

Οι εφαρμογές Android αποτελούνται από κάποια συστατικά, που συνδέονται χρησιμοποιώντας ένα XML αρχείο, το `AndroidManifest.xml` του κάθε project, το οποίο περιγράφει κάθε συστατικό και πώς αλληλεπιδρά με τα άλλα. Υπάρχουν έξι συστατικά που παρέχουν τις δομικές μονάδες για τις εφαρμογές:

- A. **Δραστηριότητα (Activity)** - Είναι το επίπεδο παρουσίασης (presentation layer) της εφαρμογής. Μια δραστηριότητα είναι μια απλή οθόνη της εφαρμογής. Κάθε δραστηριότητα υλοποιείται σαν μια κλάση που επεκτείνει (extends) την βασική κλάση Δραστηριότητα (Activity base class). Η συγκεκριμένη κλάση προβάλλει μια διεπαφή χρήστη (user interface) αποτελούμενη από Όψεις (Views) και ανταποκρίνεται σε Συμβάντα (Events). Ένα Activity είναι ισοδύναμο με μια Φόρμα (Form) στην ανάπτυξη desktop εφαρμογών.
- B. **Υπηρεσία (Service)** - Μια Υπηρεσία είναι κώδικας που τρέχει για μεγάλο χρονικό διάστημα και χωρίς διεπαφή χρήστη (UI), ενημερώνοντας τις πηγές δεδομένων και τις ορατές Δραστηριότητες (Activities), ενεργοποιώντας Ειδοποιήσεις (Notifications). Αυτές τρέχουν και επεξεργάζονται δεδομένα ακόμα και όταν οι Δραστηριότητες της εφαρμογής δεν είναι ενεργές ή ορατές.
- C. **Παροχέας Περιεχομένου (Content Provider)** - Οι εφαρμογές μπορούν να σώζουν τα δεδομένα τους σε αρχεία, σε μια βάση δεδομένων SQLite ή με οποιοδήποτε άλλο μηχανισμό μπορούν. Ένας Παροχέας Περιεχομένου, ωστόσο, είναι χρήσιμος ώστε τα δεδομένα μιας εφαρμογής να είναι διαθέσιμα και σε άλλες εφαρμογές. Ένας Παροχέας Περιεχομένου είναι μια κλάση που υλοποιεί μια συγκεκριμένη ομάδα μεθόδων που επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να επανακτούν δεδομένα του τύπου που διαχειρίζεται ο Παροχέας Περιεχομένου. Οι συσκευές Android περιλαμβάνουν διάφορους εγγενείς Παροχείς Περιεχομένου (Content Providers) που εκθέτουν τις χρήσιμες βάσεις δεδομένων, όπως για παράδειγμα των στοιχείων των επαφών του χρήστη.
- D. **Προθέσεις (Intents)** - Το Android χρησιμοποιεί μια ειδική κλάση που λέγεται Πρόθεση (Intent) για να κινείται από οθόνη σε οθόνη. Η Πρόθεση

περιγράφει τι θέλει η εφαρμογή να γίνει στη συνέχεια. Τα δυο πιο σημαντικά μέρη της δομής δεδομένων της Πρόθεσης είναι η Δράση (Action) και τα δεδομένα βάσει των οποίων αυτή θα εκτελεστεί. Τυπικές τιμές για μια δράση είναι η MAIN (η κεντρική είσοδος της εφαρμογής), VIEW, PICK, EDIT κλπ. Τα δεδομένα εκφράζονται ως URI (Uniform Resource Indicator). Υπάρχει μια σχετική κλάση που λέγεται Φίλτρο Πρόθεσης (IntentFilter). Ενώ μια Πρόθεση είναι στην ουσία ένα αίτημα για να γίνει κάτι, το Φίλτρο Πρόθεσης είναι μια περιγραφή του τι είναι δυνατόν να διαχειριστεί ένας Δέκτης Πρόθεσης (Intent Receiver). Μια Δραστηριότητα που είναι σε θέση να προβάλλει πληροφορίες επικοινωνίας για ένα άτομο, θα ανακοίνωνε με ένα Φίλτρο Πρόθεσης (IntentFilter) ότι γνωρίζει πως να διαχειριστεί την VIEW_ACTION όταν τα δεδομένα αντιπροσωπεύουν ένα άτομο. Οι Δραστηριότητες ανακοινώνουν τα Φίλτρα Πρόθεσης (IntentFilters) στο AndroidManifest.xml αρχείο. Η πλοήγηση από οθόνη σε οθόνη πετυχαίνεται με Προθέσεις. Για να πλοηγηθεί κανείς προς τα μπρος, μια Δραστηριότητα καλεί την startActivity(myIntent). Η διαδικασία της υλοποίησης Προθέσεων συμβαίνει κατά τον χρόνο εκτέλεσης της εφαρμογής, όταν δηλαδή καλείται η startActivity, πράγμα που προσφέρει 2 πλεονεκτήματα:

- i. Οι Δραστηριότητες μπορούν να επαναχρησιμοποιούν κάποια λειτουργικότητα από άλλα τμήματα του κώδικα απλά κάνοντας ένα αίτημα υπό την μορφή μιας πρόθεσης.
- ii. Οι δραστηριότητες μπορούν να αντικατασταθούν οποιαδήποτε στιγμή από μια νέα Δραστηριότητα με ένα αντίστοιχο Φίλτρο Πρόθεσης.

Ε. Δέκτες Πρόθεσης (Intent Receiver) - Χρησιμοποιούνται για να εκτελεστεί μια εφαρμογή σε απάντηση ενός εξωτερικού συμβάντος (external event), για παράδειγμα, όταν το τηλέφωνο χτυπά, ή όταν το δίκτυο είναι διαθέσιμο, ή όταν είναι μεσάνυχτα. Οι Δέκτες Πρόθεσης δεν προβάλλουν μια διεπαφή χρήστη (UI), ωστόσο μπορούν να προβάλλουν Ειδοποιήσεις (Notifications) για να ειδοποιήσουν τον χρήστη για κάτι σημαντικό που συνέβη. Οι Δέκτες Πρόθεσης είναι επίσης καταχωρημένοι στο AndroidManifest.xml, αλλά

μπορούν επίσης να καταχωρηθούν από τον κώδικα χρησιμοποιώντας την `Context.registerReceiver()`. Η εφαρμογή δεν χρειάζεται να τρέχει για να κληθούν οι Δέκτες Πρόθεσης που έχει. Το σύστημα θα κινήσει την εφαρμογή, αν χρειαστεί, όταν ένας Δέκτης Πρόθεσης ενεργοποιηθεί. Οι εφαρμογές μπορούν επίσης να στέλνουν τις δικές τους Ανακοινώσεις Πρόθεσης (Intent Broadcasts) σε άλλους με την `Context.broadcastIntent()`.

F. Δέκτες Μετάδοσης (Broadcast Receivers) - Βασική Κλάση για τον κώδικα, που θα λαμβάνει τις Προθέσεις (Intents) που στέλνονται από το `sendBroadcast()`. Μπορεί να καταχωρηθεί είτε δυναμικά μια μέσω της `Context.registerReceiver()`, είτε να καταχωρηθεί ένα στιγμιότυπο αυτής της κλάσης σε μια εφαρμογή μέσω της ετικέτας στο `AndroidManifest.xml`. Οι Δέκτες Μετάδοσης αρχίζουν αυτόματα την εφαρμογή αποκρινόμενοι σε μια εισερχόμενη Πρόθεση, που στην ουσία κάνει την εφαρμογή, μια εφαρμογή οδηγούμενη από τα γεγονότα. Υπάρχουν δύο σημαντικές κατηγορίες Μεταδόσεων (Broadcasts) που μπορούν να παραληφθούν:

- ✚ Οι *Κανονικές Μεταδόσεις* - Normal broadcasts (που στέλνονται με `Context.sendBroadcast`) είναι απολύτως ασύγχρονες. Όλοι οι Δέκτες της Μετάδοσης οργανώνονται σε μια απροσδιόριστη διαταγή, συχνά συγχρόνως. Αυτό είναι αποδοτικότερο, αλλά σημαίνει ότι οι Δέκτες δεν μπορούν να χρησιμοποιήσουν το αποτέλεσμα ή να αποβάλουν APIs που συμπεριλαμβάνονται εδώ.

- ✚ Οι *Μεταδόσεις Διαταγής* - Ordered broadcasts (που στέλνονται με `Context.sendOrderedBroadcast`) παραδίδονται σε έναν Δέκτη τη φορά. Δεδομένου ότι κάθε Δέκτης εκτελείται στη συνέχεια, μπορεί να διαδώσει ένα αποτέλεσμα στον επόμενο Δέκτη, ή μπορεί να αποβάλει τη Μετάδοση έτσι ώστε να μη περάσει σε άλλους δέκτες. Ο χειρισμός της εκτέλεσης των Δεκτών διαταγής μπορεί να γίνει με την ιδιότητα του Φίλτρου Προθέσεων Ταιριάσματος (Matching Intent-Filter). Οι Δέκτες με την ίδια προτεραιότητα τρέχουν με τυχαία σειρά.

G. Ειδοποιήσεις (Notifications) - Είναι ένα πλαίσιο ειδοποιήσεων χρήστη. Οι Ειδοποιήσεις δίνουν την δυνατότητα ειδοποίησης του χρήστη χωρίς να υπάρξει παρεμβολή στην τρέχουσα Δραστηριότητα (Activity) και του παρέχουν τις εξής δυνατότητες:

- ✚ Να δημιουργήσει ένα νέο εικονίδιο status bar.

- ✚ Να ενεργοποιήσει τα φώτα/LEDs.
- ✚ Να δονηθεί το τηλέφωνο.
- ✚ Να ενεργοποιήσει διάφορους ήχους (ringtones).

2.8 Ασφάλεια στο Android

Καθώς μια εφαρμογή εγκαθίσταται στη συσκευή, λειτουργεί αποκλειστικά στη δικιά της εικονική μηχανή, η οποία αποτελεί και το πλαίσιο ασφαλείας (sandbox) της εφαρμογής. Το Android είναι ένα λειτουργικό σύστημα πολλών χρηστών στο οποίο:

- 1) Από προεπιλογή το σύστημα δίνει έναν μοναδικό αριθμό ID ο οποίος είναι άγνωστος στην εφαρμογή. Το σύστημα αναθέτει συγκεκριμένες άδειες χρήσης στα αρχεία της εφαρμογής και μόνο η εφαρμογή με το σωστό ID μπορεί να έχει πρόσβαση σε αυτά.
- 2) Κάθε εφαρμογή τρέχει στην δική της εικονική μηχανή (VM) απομονωμένη από τις υπόλοιπες εφαρμογές. Η κάθε VM εκκινείται μόλις ζητηθεί από το σύστημα και κλείνει είτε επειδή δεν χρησιμοποιείτε πλέον, είτε επειδή το σύστημα θέλει να ελευθερώσει τους πόρους της μνήμης για χρήση από άλλη εφαρμογή. Με αυτό τον τρόπο το Android χρησιμοποιεί την αρχή των ελαχίστων δικαιωμάτων. Η κάθε εφαρμογή έχει πρόσβαση μέσω του AndroidManifest.xml μόνο σε όσους πόρους συστήματος χρειάζεται και κανέναν περισσότερο. Οι πόροι και τα δικαιώματα που απαιτούνται από μία εφαρμογή γίνονται γνωστά στον χρήστη τη στιγμή της εγκατάστασης της, και ο χρήστης μπορεί να επιλέξει να μην εγκαταστήσει μια εφαρμογή εφόσον δεν συμφωνεί να τις παρέχει πρόσβαση στους πόρους που ζητάει.

Η τεχνολογία Android εφαρμόζει ασφάλεια πολλών επιπέδων:

Αποκλεισμός. Οι εφαρμογές στο Google Play σαρώνονται διαρκώς για τον αποκλεισμό επιβλαβών εφαρμογών και παραβάσεων πολιτικών. Μπορείτε επίσης

να επιλέξετε την επαλήθευση των εφαρμογών που έχουν εγκατασταθεί από άλλες πηγές.

Έλεγχος. Τα "περιβάλλοντα δοκιμών" των εφαρμογών παρεμποδίζουν την πρόσβαση των εφαρμογών σε άλλα μέρη του λειτουργικού συστήματος του tablet σας ή μεταξύ τους, εκτός και αν παραχωρήσετε άδεια κατά την εγκατάσταση. Το κλείδωμα και η κρυπτογράφηση της οθόνης συμβάλουν στην παρεμπόδιση της μη εξουσιοδοτημένης πρόσβασης σε ολόκληρο το tablet.

Άμυνα. Μετά τον εντοπισμό τους, είναι δυνατή η απομακρυσμένη κατάργηση των μη εξουσιοδοτημένων εφαρμογών.

2.9 Γιατί Android

Εν τέλει, παρατίθενται εδώ κάποια σημεία και σκέψεις για το Android, σαν μια τελευταία απόδειξη της πρωτοπορίας του στο χώρο και της ισχυρής δυναμικής του. Γιατί Android λοιπόν;

- 1) Η αρχιτεκτονική της είναι βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
- 2) Σύνδεση όλων των εφαρμογών μεταξύ τους και όχι ανεξάρτητη λειτουργία αυτών όπως στο iOS. Για παράδειγμα, θέλω να ανεβάσω μια φωτογραφία στο Dropbox. Στην περίπτωση του Android όπως πλοηγούμαι στην gallery μπορώ με ένα απλό πάτημα του share, να ανεβάσω την φωτογραφία μου στο Dropbox, στο iOS απ' την άλλη θα πρέπει να ανοίξω την εφαρμογή του Dropbox και να κάνω από εκεί upload την φωτογραφία καθώς δεν συνδέονται η gallery με το Dropbox μεταξύ τους εξαιτίας των περιορισμών της Apple. Το ίδιο συμβαίνει σε οποιαδήποτε εφαρμογή αφού είναι θέμα λειτουργικού συστήματος.
- 3) Στο Android μπορώ να βάλω οτιδήποτε σε οποιαδήποτε οθόνη, να μετακινήσω τα apps όπου θέλω, να βγάλω ότι χρειάζομαι στην κεντρική οθόνη και να χρησιμοποιήσω widgets όπου θέλω για να λαμβάνω άμεσα κάθε είδους πληροφορία. Στο iOS από την άλλη δεν έχω Widgets, με αποτέλεσμα να πρέπει να ανοίξω κάθε εφαρμογή ξεχωριστά για να πάρω τις πληροφορίες που χρειάζομαι, ενώ η παραμετροποίηση έγκειται στην

αλλαγή θέσης των εικονιδίων των εφαρμογών αλλά και στην οργάνωση τους με φακέλους. Τέλος, στο Android μπορώ να αλλάξω και εφαρμογές του συστήματος με άλλες όπως για παράδειγμα το πληκτρολόγιο, φυσικά το iOS έχει ένα από τα κορυφαία πληκτρολόγια σε smartphone αλλά υπάρχουν 3d party πολύ καλύτερα και πιο έξυπνα.

- 4) Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλίδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο smartphone.
- 5) Υψηλής ποιότητας γραφικά και ήχος.

ΚΕΦΑΛΑΙΟ 3

ANDROID STUDIO

3.1 Τι είναι το Android Studio

Το **Android Studio** είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα [Android](#). Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο ελεύθερα με την [άδεια Apache License 2.0](#). Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0. Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και [Linux](#), και αντικατέστησε τα Eclipse Android Development Tools (ADT) ως το κύριο IDE της [Google](#) για ανάπτυξη εφαρμογών Android.



Σχήμα 3.4 Android Studio

3.1.2 Χαρακτηριστικά του Android Studio

Τα μέρη που απαρτίζουν το Android είναι :

- Υλικό που υποστηρίζει το λογισμικό.
- Βιβλιοθήκες ανοικτές για την ανάπτυξη εφαρμογών, όπως SQLite, WebKit, OpenGL.
- Πυρήνας Linux που υποστηρίζει τη διασύνδεση με το υλικό, διαχειρίζεται τη μνήμη, παρέχει διεργασίες ελέγχου και έχει βελτιστοποιηθεί για φορητές συσκευές.
- Περιβάλλον εκτέλεσης για εφαρμογές που περιλαμβάνει την εικονική μηχανή Dalvik και τις βιβλιοθήκες πυρήνα
- Υποδομές εγκατάστασης και εκτέλεσης εφαρμογών από χρήστες
- Προεγκατεστημένες εφαρμογές
- Έκθεση διαφόρων υπηρεσιών του λειτουργικού συστήματος στο επίπεδο εφαρμογών, όπως διαχείριση οθόνης, θέσης, τηλεφωνία.
- Υποδοχές ανάπτυξης και εξομοίωσης εφαρμογών χρήστη.

3.1.3 Χαρακτηριστικά του Android Studio IDE

Το Android Studio όπως προαναφέραμε έχει βάση το IntelliJ IDEA πράγμα που του δίνει τη δυνατότητα να κληρονομήσει χαρακτηριστικά του. Ας δούμε μερικά από αυτά:

- Ισχυρή Επεξεργασία Κώδικα (Powerful Code Editing) που περιλαμβάνει: Έξυπνο σύστημα αυτοσυμπλήρωσης κώδικα, ριζική πλοήγηση μεταξύ των αρχείων του έργου (Profound navigation between project files), προχωρημένη και ασφαλή επεξεργασία (π.χ. μετονομασία ενός αρχείου

και εφαρμογή του νέου ονόματος σε όλα τα μέρη του κώδικα μας που αναφέρεται αυτόματα) και προεσκόπηση πόρων.

- Ενσωματωμένα εργαλεία του Android (Built-in Android Tools) που περιλαμβάνει:
 - ❖ Ενσωμάτωση του εργαλείου αποσφαλμάτωσης Dalvik Debug Monitor Server (DDMS).
 - ❖ Περιέχει εκτέλεση και αποσφαλμάτωση για εξομοιώσεις κινητών συσκευών καθώς και πραγματικών συσκευών.
 - ❖ Ισχυρό σχεδιαστή διεπαφής χρήστη με drag-n-drop και υποστήριξη διαφορετικών διατάξεων και μεγεθών οθόνης.
 - ❖ Ενσωμάτωση του συστήματος καταγραφής LogCat με δυνατότητα αναζήτησης και εφαρμογής διάφορων φίλτρων εμφάνισης.
- Ανάλυση κώδικα κατά την επεξεργασία (On-the-fly Code Analysis). Ο συντάκτης επισημαίνει προειδοποιήσεις και σφάλματα στον κώδικα αμέσως καθώς πληκτρολογείτε και επιτρέπει να εφαρμοστεί μια προτεινόμενη γρήγορη λύση αυτόματα.

3.1.4 Εγκατάσταση του Android Studio

Παρακάτω θα δείτε βήμα-βήμα την εγκατάσταση του Android Studio. Μετά τη λήψη, τα ακριβή βήματα για να εγκαταστήσετε το Android Studio διαφέρουν ανάλογα με το λειτουργικό σας σύστημα.

- A. Λήψη της εφαρμογής Android Studio από την παρακάτω διεύθυνση: <http://developer.android.com/sdk/installing/studio.html>. Τρέξτε το αρχείο που κατεβάσατε. Ακολουθήστε τον οδηγό εγκατάστασης για να εγκαταστήσετε το Android Studio .Κλικάρετε το download button εάν ήδη έχει επιλεγεί σωστά το λειτουργικό σας σύστημα (πχ “Download Android Studio for Windows”), διαφορετικά επιλέξτε το “Other Download Options” και επιλέξτε το κατάλληλο package.

Εγκατάσταση σε Windows

Εντοπίστε το εκτελέσιμο αρχείο εγκατάστασης (που ονομάζεται Android-studiobundle .exe) σε ένα παράθυρο του Google Chrome/Mozilla και κάντε διπλό κλικ σε αυτό για να ξεκινήσει η διαδικασία εγκατάστασης. Κάντε κλικ στο κουμπί YES στο παράθυρο διαλόγου, εάν εμφανίζεται. Μόλις εμφανιστεί ο οδηγός εγκατάστασης Android Studio, προχωρήστε μέσα από τις διάφορες οθόνες για να διαμορφώσετε την εγκατάσταση έτσι ώστε να καλύπτει τις απαιτήσεις σας όσον αφορά την τοποθεσία του συστήματος αρχείων του Android Studio όπως βλέπουμε και παρακάτω στις φωτογραφίες.

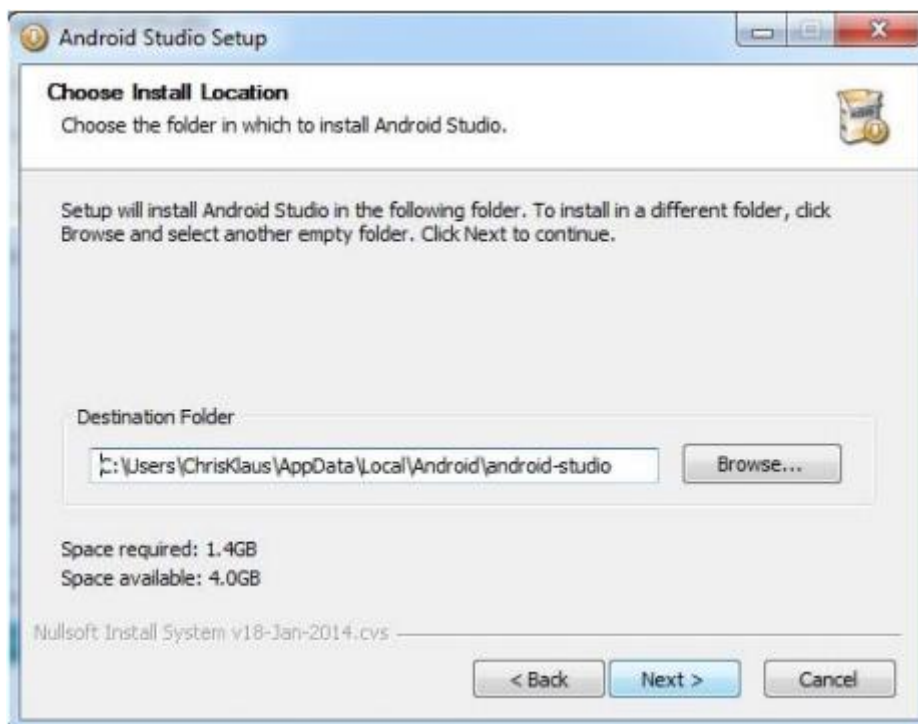


Σχήμα 3.8: Καλωσόρισμα στον οδηγό εγκατάστασης



Σχήμα 3.9: Επιλογή Χρήστη

- Β. Στην παρακάτω εικόνα διαλέγετε που θα εγκατασταθεί το πρόγραμμα. Με την επιλογή Browse διαλέγεται το που θα εγκατασταθεί. Πατήστε μετά Next.



Σχήμα 4.0: Επιλογή τοποθεσίας εγκατάστασης

C. Και εφόσον πραγματοποιήσουμε σωστά τα παραπάνω βήματα θα μας εμφανίσει, όταν τελειώσει η εγκατάσταση, την παρακάτω εικόνα.



Σχήμα 4.1: Τέλος Εγκατάστασης

Εγκατάσταση σε Mac OS X

Το Android Studio για Mac OS X κατεβαίνει σαν ένα disk image (.dmg) αρχείο. Μόλις κατέβει το androidstudioide.dmg αρχείο, εντοπίστε το σε ένα παράθυρο του Finder και κάντε διπλό κλικ για να ανοίξει:



Σχήμα 4.2: Android Studio Mac OS X

Για να εγκαταστήσετε το πακέτο, απλά σύρετε το εικονίδιο Android Studio και αφήστε το πάνω στο φάκελο Εφαρμογές Applications. Το πακέτο του Android Studio στη συνέχεια θα εγκατασταθεί στο φάκελο Εφαρμογές Applications του συστήματος, μια διαδικασία που θα διαρκέσει συνήθως μερικά λεπτά για να ολοκληρωθεί. Για να ξεκινήσει το Android Studio, εντοπίστε το εκτελέσιμο στο φάκελο Εφαρμογές, χρησιμοποιώντας ένα παράθυρο του Finder και κάντε διπλό κλικ πάνω του. Όταν προσπαθείτε να ξεκινήσει το Android Studio, μπορεί να εμφανιστεί ένα παράθυρο διαλόγου σφάλματος που υποδεικνύει ότι η JVM δεν μπορεί να βρεθεί. Εάν παρουσιαστεί αυτό το σφάλμα, θα πρέπει να κατεβάσετε και να εγκαταστήσετε το πακέτο Mac OS X Java (6 ή 7 κτλ) JRE στο σύστημα. Αυτό μπορείτε να το κατεβάσετε από την Apple χρησιμοποιώντας τον παρακάτω σύνδεσμο: <http://support.apple.com/kb/DL1572> .

3.2 Εργαλεία ανάπτυξης Android Software Development SDK

Η ανάπτυξη λογισμικού Android είναι η διαδικασία με την οποία οι νέες εφαρμογές που δημιουργούνται για το λειτουργικό σύστημα Android. Οι εφαρμογές αναπτύσσονται συνήθως στη γλώσσα προγραμματισμού Java χρησιμοποιώντας το kit ανάπτυξης λογισμικού Android (SDK), αλλά και άλλα περιβάλλοντα ανάπτυξης είναι επίσης διαθέσιμα. Ένα ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment, IDE) είναι μία σουίτα λογισμικού που βοηθάει στην ανάπτυξη προγραμμάτων υπολογιστή. Συνήθως ένα IDE περιλαμβάνει κάποιον επεξεργαστή πηγαίου κώδικα, έναν μεταγλωττιστή, εργαλεία αυτόματης παραγωγής κώδικα, αποσφαλματωτή, συνδέτη, σύστημα ελέγχου εκδόσεων και εργαλεία κατασκευής γραφικών διασυνδέσεων χρήστη για τις υπό ανάπτυξη εφαρμογές.

Τα βασικά εργαλεία ανάπτυξης της εφαρμογής ονομάζονται Android SDK[12] (Standard Development Kit) και δίνονται όλα συγκεντρωμένα από την εταιρεία Google. Αυτό περιλαμβάνει τις βιβλιοθήκες Java[4] του Android που θα χρησιμοποιήσει η εφαρμογή, τον emulator για την δοκιμαστική εκτέλεση της εφαρμογής χωρίς να χρειάζεται να υπάρχει συσκευή, τον compiler που παράγει τον κώδικα και μία πληθώρα διαφορετικών βιβλιοθηκών. Το SDK δεν παρέχεται μόνο του, αλλά έρχεται μαζί με ένα περιβάλλον ανάπτυξης Αυτή τη στιγμή υπάρχουν δύο επιλογές, το eclipse και το Android Studio[16]. Το Android Studio είναι ένα περιβάλλον βασισμένο στο IntelliJ IDEA και είναι το περιβάλλον που αναπτύσσεται ενεργά από τη Google. Υποστηρίζει όλες τις δυνατότητες που παρέχει και το eclipse, εκτός από το NDK. Το eclipse ήταν παλιότερα το επίσημο εργαλείο συγγραφής κώδικα για Android, αλλά έχει πλέον εγκαταλειφθεί. Τον Νοέμβριο του 2014 η Google άρχισε να προτείνει επίσημα στους χρήστες την χρήση του, εκτός και αν τους ήταν απαραίτητη κάποια λειτουργία που δεν παρέχει. Η ανάγκη για εργαλεία ελέγχου, καλύπτεται στο έπακρο με μια σειρά εφαρμογών τα οποία δένοντας αρμονικά με το περιβάλλον ανάπτυξης (IDE) μας λύνουν τα χέρια, στο δύσκολο έργο του εντοπισμού σφαλμάτων. Τα εργαλεία αυτά αφορούν τον έλεγχο της εφαρμογής σε εικονικές μηχανές (Android Virtual Devices), καταγραφή σφαλμάτων (logcat), εργαλεία ελέγχου μνήμης και άλλων

λειτουργιών της συσκευής (DDMS), εργαλεία ανάπτυξης (Development Tools) , κατασκευής (Build Tools) και πλατφόρμας (Platform Tools).

3.2.1 Εργαλεία αποσφαλμάτωσης

Αποσφαλματωτής (debugger) ή εργαλείο αποσφαλμάτωσης (debugging tool) ονομάζεται ένα πρόγραμμα υπολογιστή το οποίο χρησιμοποιείται για την εξέταση και αποσφαλμάτωση άλλων προγραμμάτων (του προγράμματος "στόχου"). Ο υπό εξέταση κώδικας θα μπορούσε εναλλακτικά να εκτελείται σε ένα προσομοιωτή συνόλου εντολών (instruction set simulator - ISS), μια τεχνική που καταλογίζει μεγάλη δύναμη στην ικανότητα της να σταματά όταν απαντώνται συγκεκριμένες συνθήκες αλλά τυπικά θα είναι κάπως πιο αργή από την εκτέλεση του κώδικα κατευθείαν στον κατάλληλο (ή στον ίδιο) επεξεργαστή. Μερικοί αποσφαλματωτές προσφέρουν δύο τρόπους λειτουργίας, πλήρης ή μερική προσομοίωση, για να περιορίσουν αυτόν τον αντίκτυπο.

Για την ορθή λειτουργία των παραπάνω εργαλείων χρειάζεται κάποιο είδος προγράμματος client-server που να συνδέει τον υπολογιστή με τις συσκευές μας, εικονικές και μη. Τον ρόλο αυτό αναλαμβάνει το Android Debug Bridge (ADB). Πρόκειται για ένα εργαλείο γραμμής εντολών που έρχεται μαζί με το Android SDK και το οποίο αποτελείται από 3 μέρη:

- 1) Έναν server ο οποίος τρέχει σαν υπηρεσία παρασκηνίου στον υπολογιστή που βρίσκεται το SDK, όπως και ο client. Ο server εξασφαλίζει την επικοινωνία μεταξύ του client και του εργαλείου "δαίμονα" (daemon) που τρέχει στη συσκευή.
- 2) Έναν client ο οποίος τρέχει στον υπολογιστή που έχουμε στήσει το SDK. Μπορούμε είτε να τον εκκινήσουμε χειροκίνητα είτε να χρησιμοποιήσουμε κάποιο εργαλείο το οποίο ξεκινάει αυτόματα δικό του client, όπως το DDMS ή το ADT Plugin.
- 3) Ο "δαίμονας" (daemon) που τρέχει σαν διεργασία παρασκηνίου στην εικονική ή πραγματική συσκευή που χρησιμοποιείτε για εξομοίωση.

Όταν ξεκινάει το ADB, ο client ελέγχει αν υπάρχει κάποια υπάρχουσα διεργασία του server που να εκτελείται ήδη, αλλιώς δημιουργεί μια νέα. Μετά δημιουργεί μια τοπική TCP σύνδεση στην θύρα 5037 και είναι έτοιμος να δεχτεί εντολές. Μετά ελέγχει το εύρος θυρών TCP από 5555 μέχρι 5585, στο οποίο επικοινωνούν οι συσκευές εξομοίωσης, και ελέγχει αν υπάρχουν διαθέσιμες και πόσες είναι αυτές. Αφού εντοπίσει κάποια συσκευή, ελέγχει αν σε αυτή τη συσκευή τρέχει ο “δαίμονας”, και αν ναι δημιουργείται σύνδεση adb με την συσκευή. Αφότου πραγματοποιηθεί σύνδεση μεταξύ συσκευής και client μπορούμε να χρησιμοποιήσουμε όλες τις δυνατότητες που μας παρέχει το ADB για να ασκήσουμε πλήρη έλεγχο της συσκευής μας. Οι εντολές μέσω γραμμής εντολών δίνονται ως εξής:

```
adb [-d|-e|-s <serialNumber>] <command>
```

όπου -d είναι η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη συσκευή, -e η απευθείας εντολή εάν υπάρχει μόνο μία συνδεδεμένη εικονική συσκευής η απευθείας εντολή στην συσκευή που έχει το , και είναι η εντολή που θέλουμε να εκτελεστεί.

Ένα πρόσθετο εργαλείο αποσφαλμάτωσης είναι το DDMS του ADT plugin. Ενώ η εφαρμογή εκτελείται, επιλέγεται το DDMS στο επάνω δεξί μέρος της οθόνης. Η επάνοδος στην αρχική οθόνη γίνεται με την επιλογή της Java.

Μερικές από τις δυνατότητες που προσφέρει το συγκεκριμένο εργαλείο είναι:

- Στο παράθυρο Devices απεικονίζονται οι εξομοιωτές και οι συσκευές που βρίσκονται σε λειτουργία καθώς επίσης και οι διεργασίες που εκτελούνται.
- Η διαχείριση της μνήμης από μια συσκευή Android είναι μια πολύ κρίσιμη λειτουργία. Με τον όρο “Heap Memory” εννοούμε μια ποσότητα μνήμης, μέρη της οποίας δεσμεύονται από το λειτουργικό σύστημα για μια διεργασία, για ένα αντικείμενο, μια μεταβλητή, κλπ. Με την επιλογή “Update Heap” αφού έχουμε ενεργοποιήσει το παράθυρο Heap, μπορούμε να δούμε την χρήση μνήμης πατώντας “Cause GC”, τη χρήση της μνήμης ανά αντικείμενο πατώντας στο αντικείμενο.

Τέλος το DDMS μπορεί να δημιουργήσει εικονικά γεγονότα, όπως κλήσεις, μηνύματα SMS και συντεταγμένες θέσης, μέσω του παραθύρου Emulator Control. Τα δημιουργημένα γεγονότα είναι εισερχόμενα, από το DDMS προς τον εξομοιωτή.

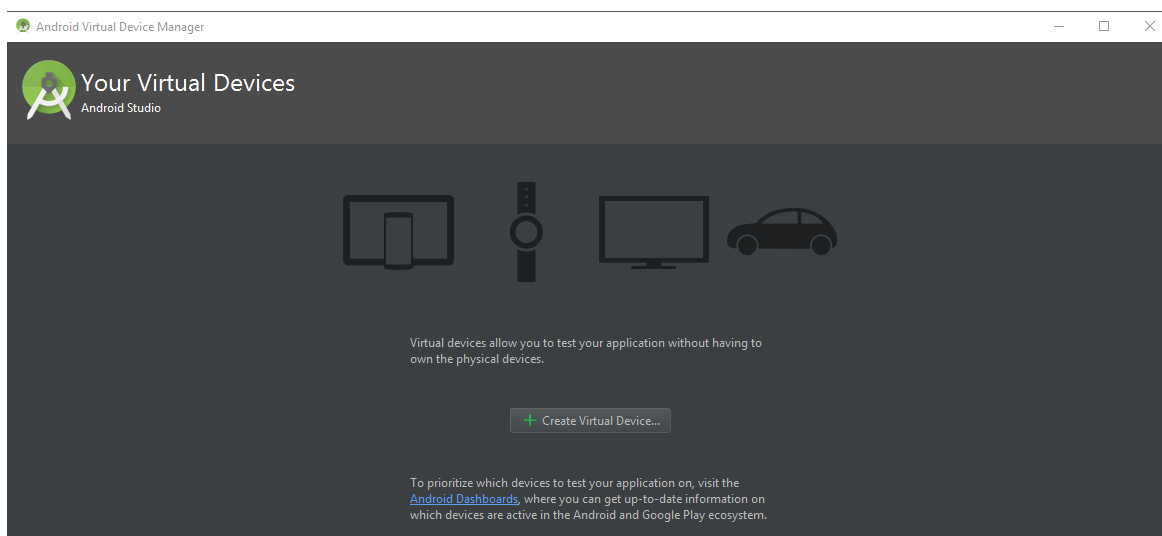
3.2.2 Εικονικές Συσκευές Android (Android Virtual Devices – AVD)

Πρόκειται για μια συσκευή εξομοίωσης η οποία μας επιτρέπει να εξομοιώσουμε την λειτουργία και συμπεριφορά μιας κανονικής συσκευής, ορίζοντας τις επιλογές υλικού και λογισμικού που θέλουμε στον εξομοιωτή του Android. Με αυτό τον τρόπο ο developer μπορεί να ελέγξει την εφαρμογή του σε μια σειρά από πραγματικά σενάρια λειτουργίας και να πάρει γρήγορα και άμεσα feedback για τη λειτουργία της εφαρμογής του. Μια εικονική συσκευή αποτελείται από:

- Την έκδοση του Android: Επιλογή της έκδοσης της πλατφόρμας του Android που θέλουμε να εξομοιώσει η εικονική συσκευή. Μπορούμε επίσης να επιλέξουμε και μεταξύ ειδικών εκδόσεων της πλατφόρμας, μεταξύ των οποίων τις Google TV, και άλλων.
- Το προφίλ του υλικού: Σε αυτό προσδιορίζονται οι ιδιότητες και τα χαρακτηριστικά της εικονικής συσκευής. Μπορούμε παραδείγματος χάρη να ορίσουμε την ανάλυση της οθόνης και την πυκνότητα σε pixels (dpi), το μέγεθος της μνήμης RAM, αν η συσκευή θα έχει κάμερα, υποστήριξη GPS, κλπ.
- Έξτρα χώρος αποθήκευσης: Εδώ αποθηκεύονται όλα τα δεδομένα της εφαρμογής, και επίσης μπορούμε να ορίσουμε μια εικονική κάρτα μνήμης ώστε να επεκτείνουμε τον αποθηκευτικό χώρο, όπως θα κάναμε και σε μια πραγματική συσκευή.

Δημιουργία μιας AVD:

Αρχικά εκκινούμε τον Διαχειριστή AVD (AVD Manager). Έπειτα επιλέγουμε στο Android Studio Tools > Android > AVD Manager.



Σχήμα 4.3: Δημιουργία AVD Manager

Στην κύρια οθόνη του Διαχειριστή AVD κάνουμε κλικ στο Create Virtual Device. Στο παράθυρο Select Hardware, επιλέγουμε κάποια συσκευή και μετά πατάμε Next. Επιλέγουμε την επιθυμητή έκδοση συστήματος για την AVD και πατάμε στο Next. Για την οριστικοποίηση της επιλογής μας πατάμε Finish.

3.2.3 Εργαλείο καταγραφής συμβάντων – LogCat

Το Android διαθέτει ένα μηχανισμό καταγραφής συμβάντων, σκοπός του οποίου είναι η συλλογή και προβολή των αρχείων αποσφαλμάτωση του συστήματος. Τα δεδομένα των διάφορων εφαρμογών αλλά και του λειτουργικού συστήματος συγκεντρώνονται σε μια σειρά από buffers, τους οποίους μετά μπορούμε να προβάλουμε και να φιλτράρουμε με την εντολή "logcat". Στον προγραμματισμό υπάρχουν οι λεγόμενες "εξαιρέσεις" (exceptions), καταστάσεις δηλαδή που προκύπτουν όταν κάτι δεν πάει καλά, κάτι το οποίο έχει σαν αποτέλεσμα την διακοπή λειτουργίας του προγράμματος σε περίπτωση που δεν

έχουμε φροντίσει να “χειριστούμε” την εξαίρεση. Τα σφάλματα λειτουργίας μιας εφαρμογής στο Android, που προκύπτουν από εξαιρέσεις λειτουργίας, συνήθως προκαλούν τον άμεσο τερματισμό της εφαρμογής προβάλλοντας ένα παράθυρο με το όνομα της εφαρμογής που τερματίστηκε, και ένα απλό μήνυμα σφάλματος, δίνοντας μας την “επιλογή” να πατήσουμε “Force Close”. Το LogCat λοιπόν είναι μια εντολή γραμμής εντολών η οποία μπορεί να χρησιμοποιηθεί μέσω του ADB για να δούμε τα debug logs της συσκευής που δουλεύουμε, και άρα ως συνεπακόλουθο, της εφαρμογής που αναπτύσσουμε ώστε να εντοπίσουμε τις πηγές των σφαλμάτων, οι οποίες συνήθως αν όχι πάντα, είναι exceptions στον κώδικα μας. Το eclipse ενσωματώνει μία GUI έκδοση του LogCat για αποτελεσματικότερη αποσφαλμάτωση του κώδικα μας. Φυσικά υπάρχουν και τα σφάλματα στον κώδικα τα οποία δεν προκαλούν αναγκαστικό κλείσιμο της εφαρμογής, αλλά παρόλα αυτά συμβάλουν στην μη σωστή λειτουργία της. Αυτού του είδους τα σφάλματα φυσικά δεν τα πιάνει ο compiler, ούτε εμφανίζονται με την μορφή που εμφανίζονται τα σφάλματα που προκύπτουν από εξαιρέσεις. Το αποτέλεσμα συνήθως αυτών των σφαλμάτων είναι μια κενή λίστα, μια λάθος τοποθετημένη εικόνα, κάποιο λάθος κείμενο, κλπ. Πως αντιμετωπίζουμε λοιπόν ένα σφάλμα το οποίο δεν εμφανίζει κάποιο stack trace με ακριβές σημείο κώδικα προς διόρθωση, όπως συμβαίνει στην περίπτωση των εξαιρέσεων; Για το σκοπό αυτό μπορούμε να χρησιμοποιήσουμε το σύστημα καταγραφής του Android, για να πάρουμε τις πληροφορίες που θέλουμε. Η κλάση Log υπάρχει για αυτό ακριβώς το σκοπό. Περιλαμβάνει έναν αριθμό από διαφορετικές μεθόδους αναλόγως με τον τύπο του σφάλματος που ψάχνουμε. Η πιο συνηθισμένη μέθοδος είναι η Log.d(), όπου το d συμβολίζει τη λέξη Debug.

3.2.4 Εργαλεία ανάπτυξης (Development Tools)

Τα εργαλεία ανάπτυξης είναι τα παρακάτω:

- a) Hierarchy Viewer: Επιτρέπει να κάνετε αποσφαλμάτωση και τη βελτιστοποίηση της διεπαφής χρήστη σας. Παρέχει μια οπτική αναπαράσταση Δείτε ιεραρχία της διάταξης (το Layout View) και μια μεγεθυμένη επιθεωρητή της οθόνης (το Pixel Perfect View).
- b) Ο Διαχειριστής Εργαλείων Ανάπτυξης Εφαρμογών (SDK Manager).
- c) Sqlite3: Από ένα απομακρυσμένο κέλυφος για τη συσκευή σας ή από το μηχάνημα που σας φιλοξενεί, μπορείτε να χρησιμοποιήσετε το sqlite3 πρόγραμμα γραμμής εντολών για τη διαχείριση SQLite βάσεις δεδομένων που δημιουργήθηκαν από εφαρμογές Android. Το εργαλείο sqlite3 περιλαμβάνει πολλές χρήσιμες εντολές, όπως .dump για να εκτυπώσετε τα περιεχόμενα ενός πίνακα και .schema για να εκτυπώσετε το SQL CREATE δήλωση για έναν υπάρχοντα πίνακα.

3.2.5 Εργαλεία κατασκευής (Build Tools)

Τα εργαλεία κατασκευής μπορούν να χρησιμοποιηθούν για την ανάπτυξη και την εργασία από την γραμμή εντολών. Τα εργαλεία κατασκευής είναι τα παρακάτω:

- Το εργαλείο Apksigner μας επιτρέπει την σύνδεση με αρχεία APK και την επιβεβαίωση ότι η υπογραφή ενός APK θα επαληθευτεί με επιτυχία σε όλες τις εκδόσεις της πλατφόρμας Android που υποστηρίζεται από τις APK. Αυτή η σελίδα παρουσιάζει ένα σύντομο οδηγό για τη χρήση του εργαλείου και χρησιμεύει ως σημείο αναφοράς για τις διάφορες επιλογές γραμμής εντολών που υποστηρίζει το εργαλείο.
- Java Opaque Binary Blob (JOB). Το εργαλείο JOB μας επιτρέπει να δημιουργήσουμε κρυπτογραφημένα και μη κρυπτογραφημένα αρχεία επέκτασης APK σε αδιαφανής Binary Blob (OBB) μορφή. Μπορούμε να

κατεβάσουμε και να τοποθετήσουμε αυτά τα αρχεία επέκτασης στην εφαρμογή μας χρησιμοποιώντας StorageManager σε συσκευές με Android 2.3 (API Level 9) ή υψηλότερη. Τα αρχεία OBB χρησιμοποιούνται για την παροχή πρόσθετων περιουσιακών στοιχείων του αρχείου για Android εφαρμογές (όπως γραφικά, ήχους και βίντεο), ξεχωριστά από το αρχείο APK της εφαρμογής.

- Το zipalign το οποίο είναι ένα εργαλείο αρχείο ευθυγράμμισης που παρέχει σημαντική βελτιστοποίηση της εφαρμογής Android αρχεία (APK). Ο σκοπός είναι να διασφαλιστεί ότι όλα τα ασυμπίεστα δεδομένα αρχίζουν με μια συγκεκριμένη ευθυγράμμιση σε σχέση με την αρχή του αρχείου. Συγκεκριμένα, προκαλεί όλα τα ασυμπίεστα δεδομένα εντός του ΔΧΕ, όπως εικόνες ή αρχεία RAW, να ευθυγραμμιστούν με τα όρια 4-byte. Αυτό επιτρέπει σε όλα τα τμήματα να έχουν άμεση πρόσβαση με mmap (), ακόμη και αν περιέχουν δυαδικά δεδομένα με περιορισμούς ευθυγράμμισης. Το όφελος είναι η μείωση στην ποσότητα της μνήμης RAM που καταναλώνονται κατά την εκτέλεση της εφαρμογής. Το εργαλείο αυτό πρέπει πάντα να χρησιμοποιείται για την ευθυγράμμιση αρχείων APK πριν από την διανομή σε τελικούς χρήστες.
- Και τέλος το ProGuard το οποίο συρρικνώνει ,βελτιστοποιεί και συσκοτίζει τον κώδικα με την αφαίρεση αχρησιμοποίητου κώδικα και μετονομασία κατηγοριών, τα πεδία και οι μέθοδοι με σημασιολογικά ονόματα.

3.2.6 Εργαλεία Πλατφόρμας(Platform tools)

Τα εργαλεία της πλατφόρμας συνήθως ενημερώνονται κάθε φορά που θα εγκαταστήσετε μια νέα πλατφόρμα SDK. Κάθε ενημέρωση από τα εργαλεία της πλατφόρμας είναι συμβατή με παλαιότερες πλατφόρμες. Συνήθως, μπορείτε να χρησιμοποιήσετε άμεσα μόνο ένα από τα εργαλεία πλατφόρμας-το Android Debug Bridge (ADB). Το Android Debug Bridge είναι ένα ευέλικτο εργαλείο που σας επιτρέπει να διαχειριστείτε την κατάσταση μιας για παράδειγμα εξομοιωτή ή το Android-powered συσκευή. Μπορείτε επίσης να το χρησιμοποιήσετε για να

εγκαταστήσετε ένα αρχείο εφαρμογής Android (APK) σε μια συσκευή. Τα εργαλεία πλατφόρμας είναι τα παρακάτω:

- I. Android Debug Bridge (ADB) όπως είπαμε και παραπάνω.
- II. LogCat όπως προαναφέραμε σε προηγούμενο υποκεφάλαιο(βλ.3.2.3)
- III. Και τρίτον το BMGR,το οποίο είναι ένα εργαλείο κέλυφος που μπορείτε να χρησιμοποιήσετε για να αλληλεπιδράσει με την Διαχείριση Backup για συσκευές Android έκδοση 2.2 (API Level 8) ή υψηλότερη. Το εργαλείο παρέχει εντολές για την έναρξη δημιουργίας αντιγράφων ασφαλείας και λειτουργίες επαναφοράς, έτσι ώστε να μην χρειάζεται να σκουπίσει επανειλημμένα δεδομένα ή να λάβει παρόμοια παρεμβατικά βήματα για να ελέγξετε τη λειτουργία δημιουργίας αντιγράφων ασφαλείας της εφαρμογής σας. Το εργαλείο bmgr υποστηρίζει τόσο Auto Backup και κλειδιού / τιμής Backup. Μπορείτε να εκτελέσετε bmgr εντολές σε μια συσκευή μέσω ADB κέλυφος και στη συνέχεια παρακολουθεί την έξοδο των εντολών με logcat. Για μια λίστα και την περιγραφή των διαθέσιμων εντολών, εκτελέστε το εργαλείο bmgr χωρίς επιχειρήματα.

ΚΕΦΑΛΑΙΟ 4

ANDROID SDK ΚΑΙ JAVA PROJECT

4.1 Android SDK

Το Android SDK παρέχει τα απαραίτητα εργαλεία για την ανάπτυξη προγραμμάτων χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Παρέχονται APIs για την χρήση προγραμμάτων περιήγησης στο διαδίκτυο, εμφάνιση δισδιάστατων και τρισδιάστατων γραφικών, χρήση των τεχνολογιών GSM, Bluetooth, EDGE, 3G, 4G και WiFi, χρήση συσκευών όπως φωτογραφική μηχανή, GPS, πυξίδα, επιταχυνσιόμετρο, δομημένη αποθήκευση δεδομένων σε βάση δεδομένων, εμφάνιση πολυμεσικού υλικού. Το πακέτο Android SDK συνεργάζεται με το Eclipse και το Android Studio ευνοώντας την ανάπτυξη εφαρμογών και συνεπώς ο προγραμματιστής μπορεί εύκολα και γρήγορα να βλέπει τις αλλαγές του κώδικα στον emulator που του παρέχει το Android SDK, χωρίς να χρειάζεται να εξάγει κάθε φορά την εφαρμογή και να την εγκαθιστά σε κινητό. Επίσης, ο emulator, έχει ακριβώς την ίδια συμπεριφορά με αυτή που θα είχε η εφαρμογή εάν είχε εγκατασταθεί σε ένα κινητό τηλέφωνο Android, κάτι που το καθιστά πολύ αξιόπιστο. Σε αναβαθμίσεις λογισμικού, η εφαρμογή εξακολουθεί να δουλεύει χωρίς την ανάγκη επανασχεδιασμού κάποιων σημαντικών κομματιών του κώδικα, που αφορούν την αλληλεπίδραση της εφαρμογής με τα δομικά μέρη του κινητού τηλεφώνου κάτι που αποτελεί άλλο ένα σημαντικό πλεονέκτημα.

4.1.1 Χαρακτηριστικά του SDK

Το Android SDK αποτελεί ένα πολύ χρήσιμο εργαλείο για την εξερεύνηση του «εσωτερικού» του Android. Παρακάτω, παρουσιάζονται τα χαρακτηριστικά του:

- SQLite για αποθήκευση δεδομένων
- Φωτογραφική μηχανή
- Δωρεάν διανομή και χρήση.
- GPS, πυξίδα, και επιταχυνσιόμετρο
- Εργαλεία για τον εντοπισμό σφαλμάτων
- Υποστήριξη πολυμέσων για αρχεία ήχου, βίντεο και εικόνων (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- Άριστα γραφικά, τα οποία τροφοδοτούνται από μια προσαρμοσμένη βιβλιοθήκη 2D και 3D
- Bluetooth, EDGE, 3G, WiFi (εξαρτώμενα από το hardware)
- GSM
- Ολοκληρωμένο πρόγραμμα περιήγησης
- Υποστήριξη Wi-Fi
- Υποστήριξη εφαρμογών P2P με χρήση του GoogleTalk
- Φυλλομετρητή διαδικτύου (WebKit browser)
- Υποστήριξη ενδοεπικοινωνίας μεταξύ των εφαρμογών (IPC-Inter Process Communication)

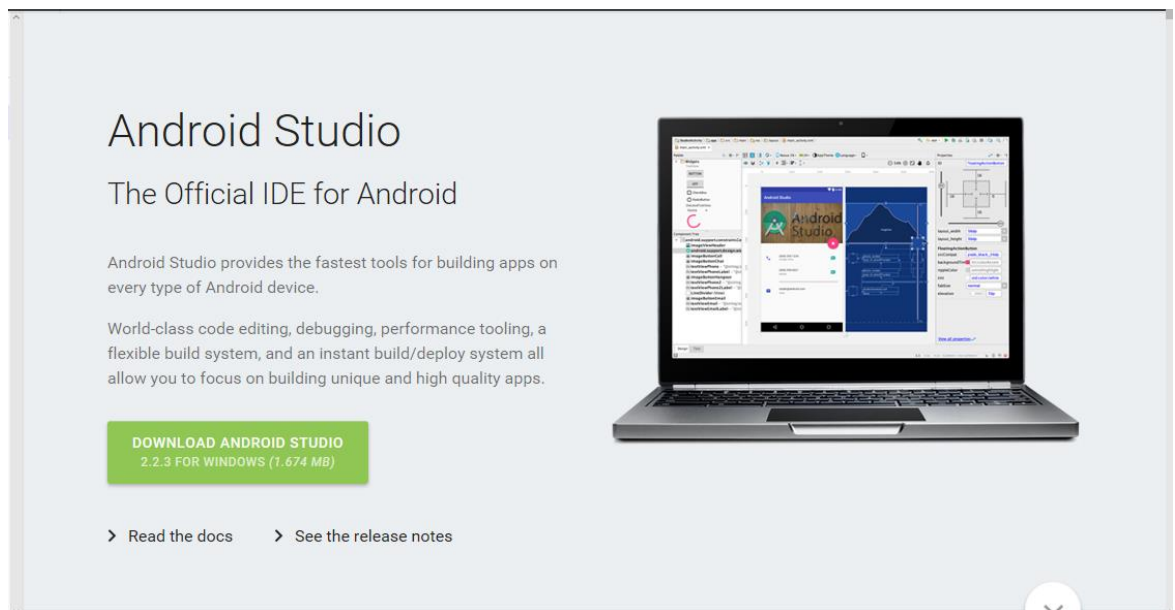
4.1.2 Περιεχόμενο του SDK

Οι εφαρμογές Android είναι γραμμένες σε Java αλλά εκτελούνται στην εικονική μηχανή Dalvik. Κάθε εφαρμογή αποτελεί μια διαφορετική διεργασία η οποία εκτελείται στο περιβάλλον που δημιουργεί το λειτουργικό σύστημα, το οποίο τις διαχειρίζεται με τέτοιο τρόπο ώστε να υπάρχει βέλτιστη διαχείριση των πόρων της συσκευής. Η μηχανή Dalvik και το σύστημα εκτέλεσης των εφαρμογών βρίσκεται πάνω από τον πυρήνα Linux ο οποίος διαχειρίζεται σε χαμηλό επίπεδο το υλικό της συσκευής με τόσους απαιτούμενους οδηγούς. Στο εργαλείο SDK υπάρχουν τα εξής:

- i. **Εργαλεία ανάπτυξης** - Δυνατότητα ανάπτυξης και αποσφαλμάτωσης κώδικα για τη δημιουργία εφαρμογών, οι οποίες μπορούν να εκτελούνται στον εξομοιωτή ή σε πραγματική κινητή συσκευή.
- ii. **Εργαλεία εξομοίωσης** – Δυνατότητα εξομοίωσης κινητών συσκευών με δραστικές δυνατότητες ώστε ο χρήστης να δοκιμάζει τις εφαρμογές του σε

- ένα περιβάλλον που μοιάζει σημαντικό με τις πραγματικές κινητές συσκευές.
- iii. **API** – Βιβλιοθήκες που παρέχουν πρόσβαση σε διαδικασίες του λειτουργικού συστήματος. Με τη βοήθεια αυτών των βιβλιοθηκών έχουν δημιουργηθεί οι ενσωματωμένες εφαρμογές των κινητών συσκευών.
 - iv. **Τεκμηρίωση** – Πλήρης τεκμηρίωση για τη χρήση κλάσεων και μεθόδων αυτών.
 - v. **Κώδικας** – Υποδείγματα εφαρμογών για την κατανόηση και εξοικείωση με τη χρήση βιβλιοθηκών αλλά και τη φιλοσοφία του λειτουργικού συστήματος.
 - vi. **Διαδικτυακή υποστήριξη** – Δυναμική κοινότητα για την ανάπτυξη εφαρμογών.

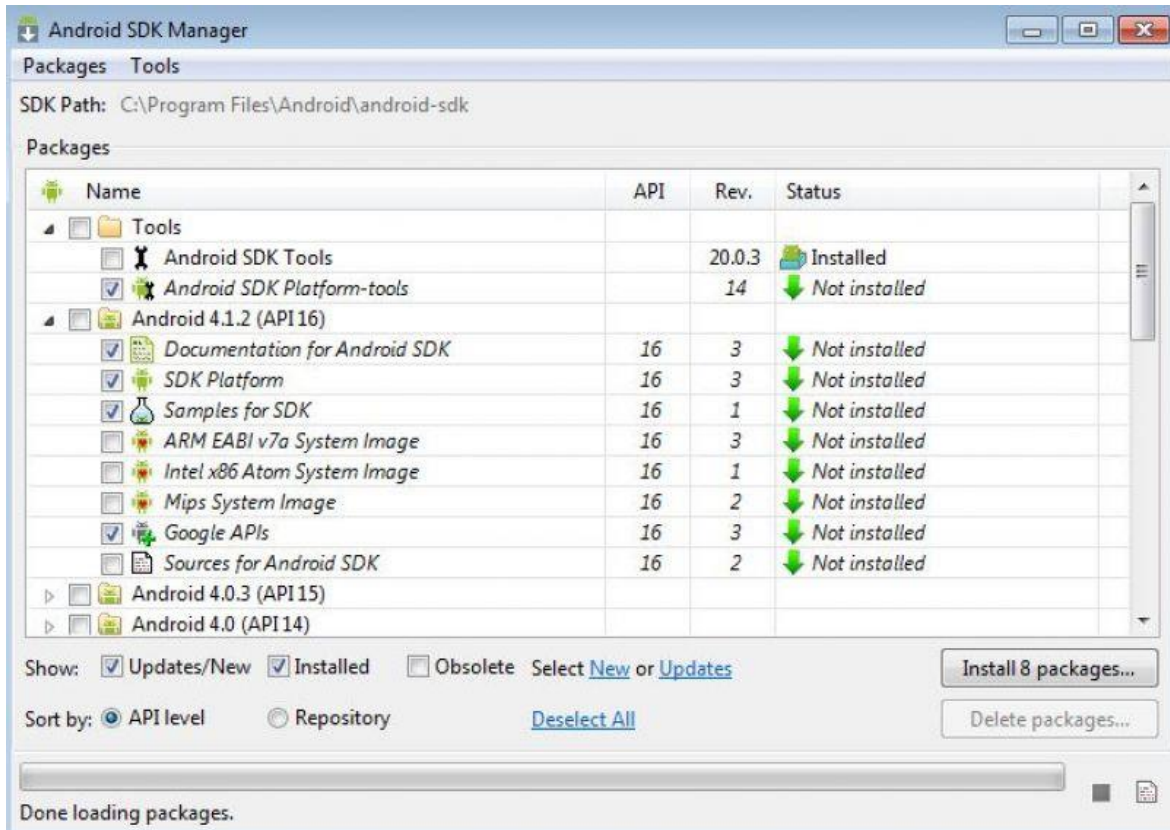
4.1.3 Εγκατάσταση του Android SDK



Σχήμα 4.4: Κατέβασμα Εφαρμογής

- 1) Από τη σελίδα <http://developer.android.com/sdk/index.html> κατεβάζουμε την εφαρμογή.
- 2) Κάνουμε διπλό κλικ στο εκτελέσιμο αρχείο (.exe) για να ξεκινήσει η εγκατάσταση. Σημειώστε το όνομα και τη θέση στην οποία αποθηκεύεται το

SDK διότι θα χρειαστεί να αναφερθείτε στον κατάλογο SDK αργότερα, κατά τη δημιουργία των plugin ADT και κατά τη χρήση των εργαλείων SDK από τη γραμμή εντολών. Μόλις ολοκληρωθεί η εγκατάσταση μπορείτε να ξεκινήσετε την Διαχείριση Android SDK.



Σχήμα 4.5: Εγκατάσταση Android SDK

Android SDK σημαίνει “Android Software Development Kit” και είναι το επίσημο εργαλείο της Google για αυτούς που θέλουν να δημιουργήσουν στο Android.

Παρακάτω θα βλέπουμε ότι είναι το βασικό εργαλείο για:

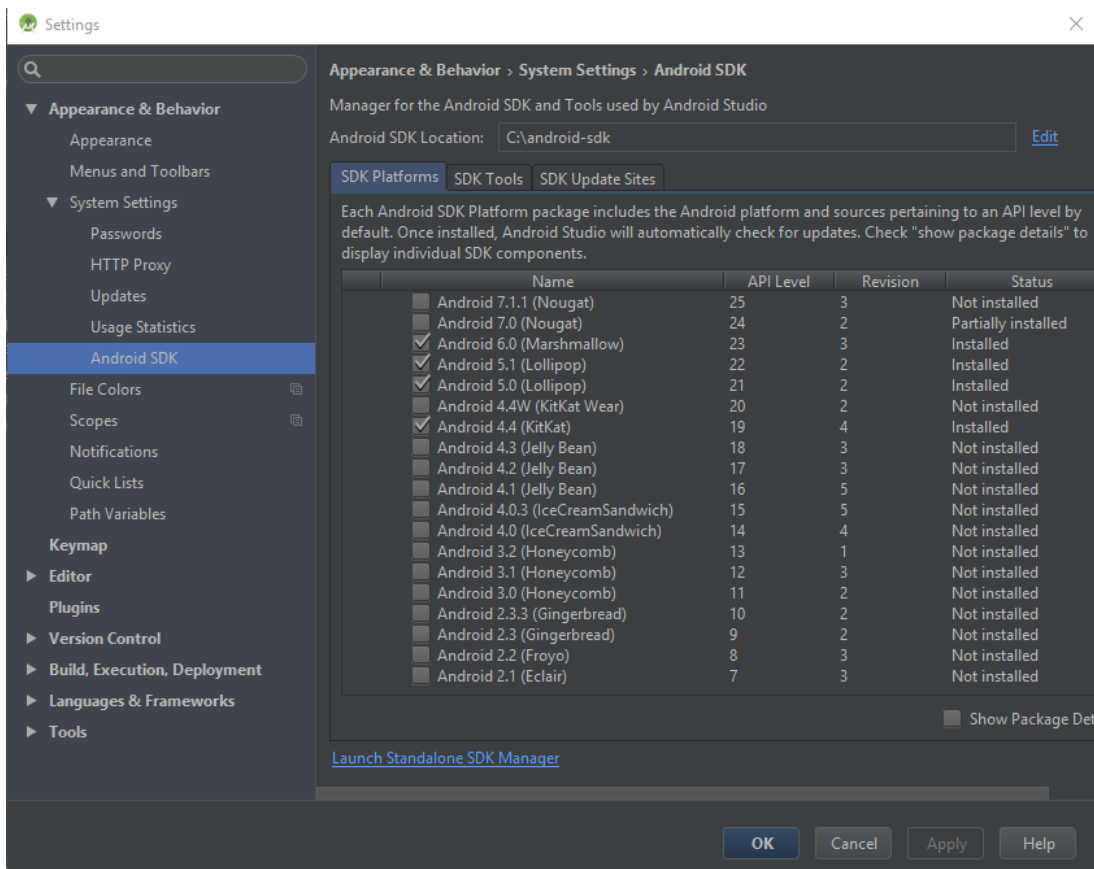
- Δημιουργία εφαρμογών
- Κατασκευή custom ROM
- Μεταγλώττιση πυρήνα για τη συσκευή μας
- και γενικό hacking

Μέσω του **SDK** μπορούμε να χρησιμοποιήσουμε εργαλεία όπως το **ADB**, για να μεταφέρουμε αρχεία σε χώρους που κανονικά δεν επιτρέπεται και το **fastboot** για να εγκαθιστούμε custom recovery εικόνες και να ξεκλειδώνουμε τον bootloader της συσκευής μας, κυρίως όμως μπορούμε να κατασκευάσουμε προγράμματα για το Android αρκεί να έχουμε γνώσεις Java προγραμματισμού για να το καταφέρουμε με επιτυχία.

Το Android Studio πρόκειται για ένα Android περιβάλλον ανάπτυξης που βασίζεται στο IntelliJ IDEA και είναι ειδικά σχεδιασμένο για την ανάπτυξη του Android. Εγκαινιάστηκε στις 16 Μαΐου 2013, κατά τη διάρκεια της ετήσιας εκδήλωσης της Google I/O 2013. Περιέχει όλα τα Android SDK Εργαλεία για το σχεδιασμό, τη δοκιμή, την αποσφαλμάτωση και το προφίλ της εφαρμογής σας.

4.1.5 Πληροφορίες που μας δίνει ο SDK Manager

Στην παρακάτω εικόνα παρατηρούμε τις πλατφόρμες Android που είναι διαθέσιμες ως προς να κατέβασμα, τα ονόματα τους με τα επίπεδα API Level, τις έως τώρα αναθεωρήσεις όπως και το στάτους τους.

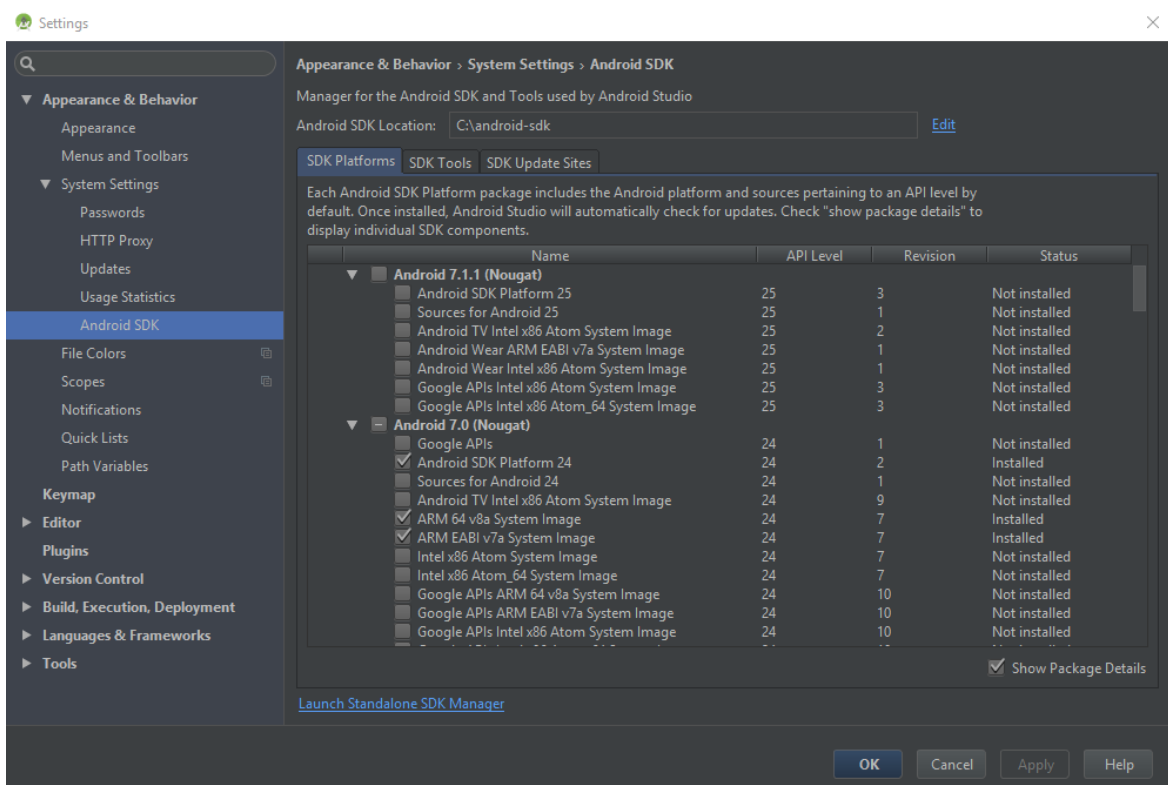


Σχήμα 4.6: Πίνακας Android πλατφόρμων

Ένα πακέτο μπορεί να έχει τρεις καταστάσεις όπως βλέπουμε και στην παραπάνω εικόνα, οι οποίες είναι:

- ✓ Installed – Εγκατεστημένο
- ✓ Update Available – Διαθέσιμη ενημέρωση
- ✓ Not Installed – Μη εγκατεστημένο

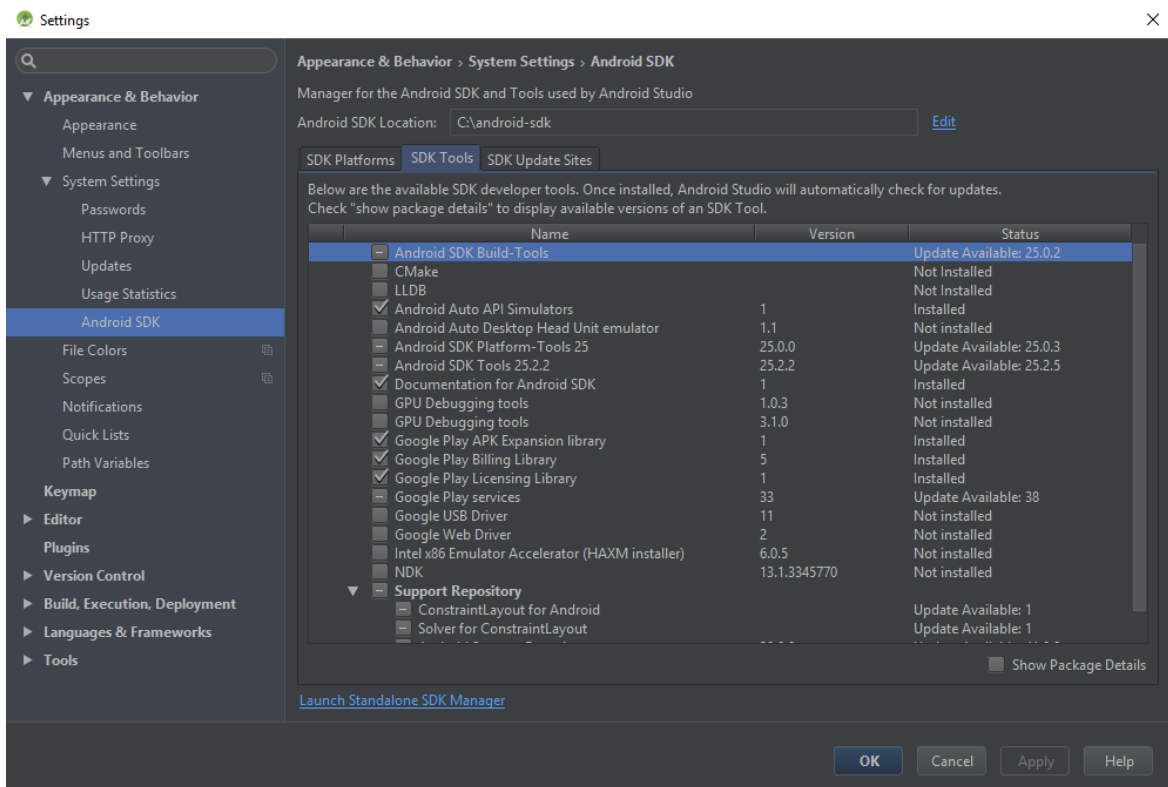
Εάν θέλουμε να δούμε τα συστατικά τα οποία διαθέτει κάθε διαθέσιμο πακέτο (platform) τότε επιλέγουμε την εμφάνιση λεπτομερειών πακέτου (Show Package Details) όπως βλέπουμε και παρακάτω.



Σχήμα 4.7: Εμφάνιση λεπτομερειών πακέτων

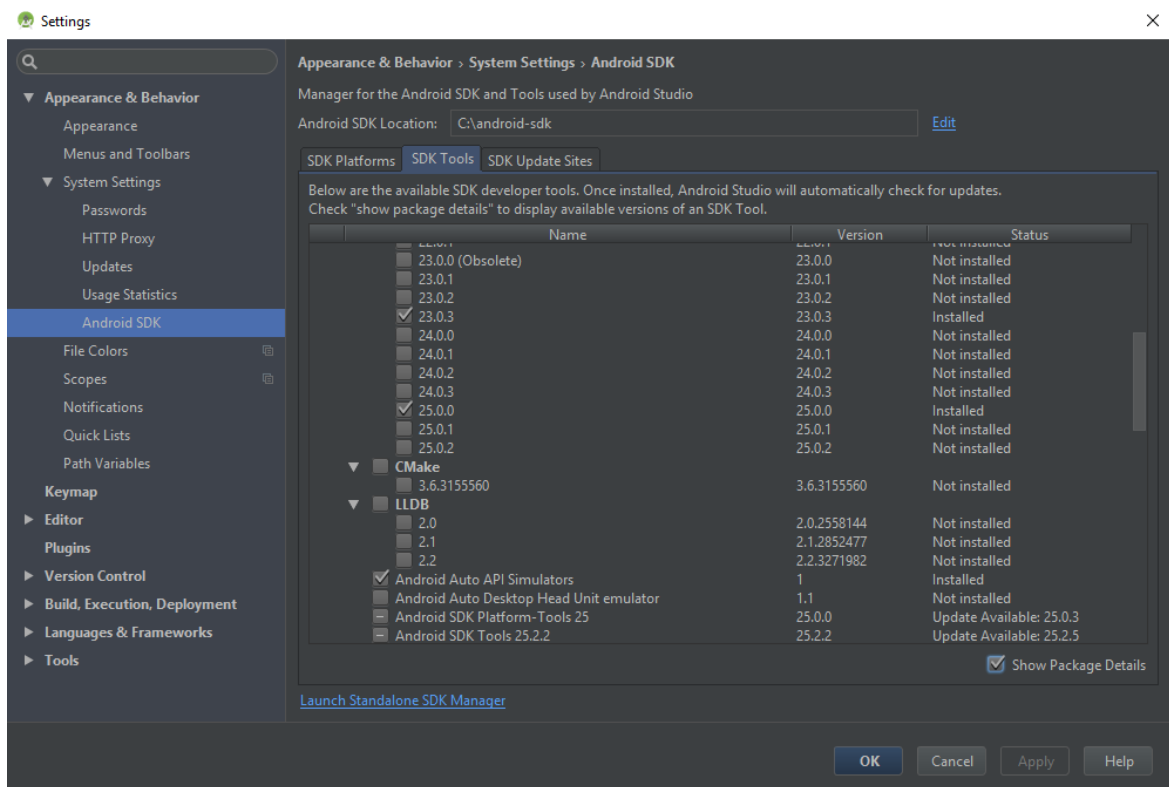
Ανάπτυξη εφαρμογής βοτανικού κήπου σε περιβάλλον Android με χρήση σάρωσης QR Code

Εν συνέχεια αν επιλέξουμε την καρτέλα SDK Tools θα παρατηρήσουμε ότι διαθέτει όλα τα διαθέσιμα εργαλεία ανάπτυξης του SDK.Επίσης όπως και παραπάνω στα πεδία βλέπουμε το όνομα του πακέτου ,την έκδοση και το στάτους του.



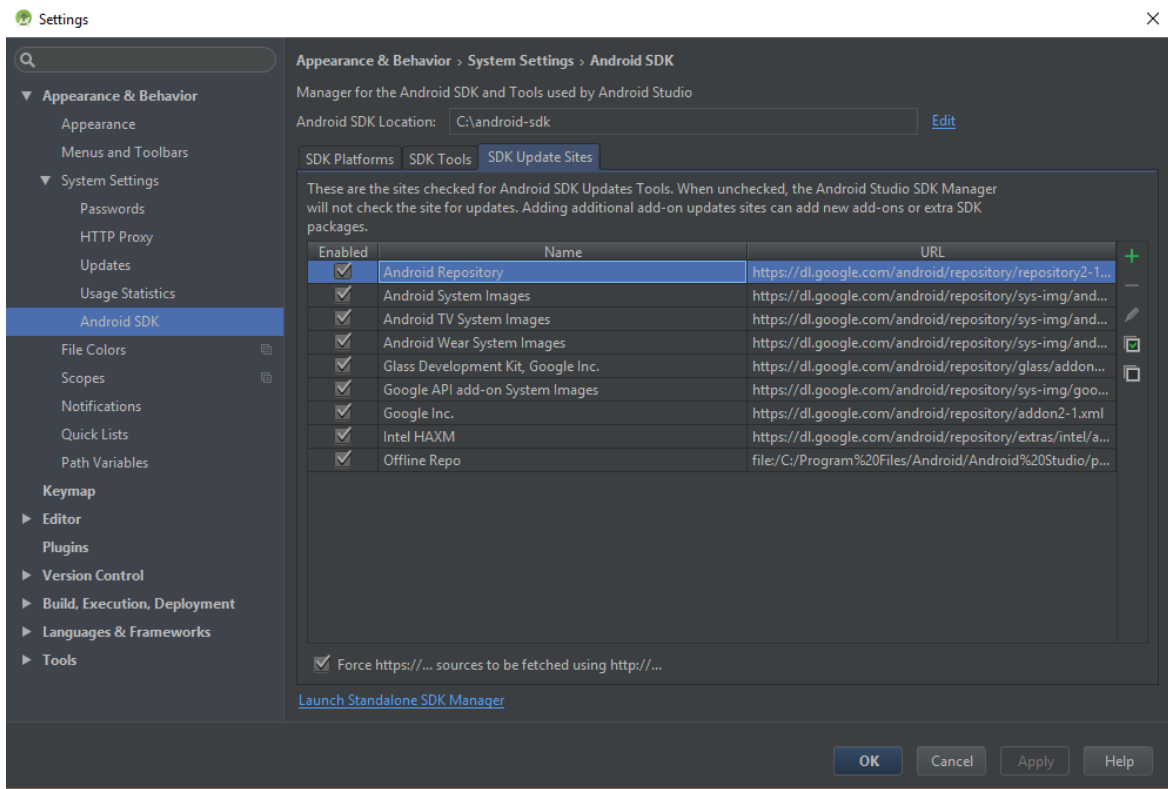
Σχήμα 4.8: SDK Tools

Αν επιλέξουμε ξανά όπως και στην καρτέλα SDK Platforms το κουτάκι Show Package Details θα μας εμφανίσει για κάθε εργαλείο ανάπτυξης τα συστατικά του.



Σχήμα 4.9: SDK Tools εμφάνιση λεπτομερειών

Τέλος, μέσω της τελευταίας καρτέλας –SDK Update Sites – ελέγχουμε τις ενημερώσεις των εργαλείων SDK.



Σχήμα 5.0: Ενημερώσεις SDK εργαλείων

4.2 Android Java Project

Ένα Android Java project περιέχει όλα τα αρχεία που συγκροτούν τον πηγαίο κώδικα της εφαρμογής του Android σας τυλιγμένα σε ένα πακέτο για εύκολη διαχείριση.

Τα σημαντικότερα αρχεία που υπάρχουν σε ένα νέο Android Java Project μετά την δημιουργία του είναι κάποια από τα παρακάτω:

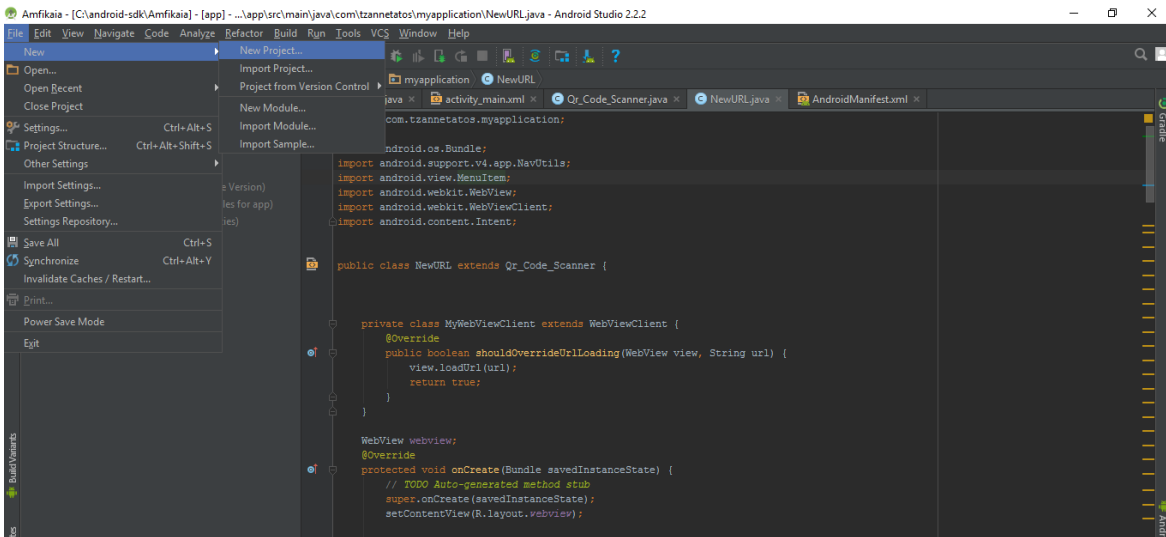
- ❖ **.gradle.** Ο .gradle φάκελος περιέχει διάφορες υπολογισμένες πληροφορίες για την έκδοση gradle
- ❖ **.idea.** Οι ρυθμίσεις του Project αποθηκεύονται με κάθε συγκεκριμένο Project ως ένα σύνολο από XML αρχεία στο φάκελο .idea
- ❖ **App.**
 - **build/** Περιέχει τους φακέλους κατασκευής για τις συγκεκριμένες παραλλαγές κατασκευής.

- **libs/** Περιέχει ιδιωτικές βιβλιοθήκες.
- **src/** Περιέχει το στέλεχος (Stub) αρχείο της Δραστηριότητας (Activity), η οποία αποθηκεύεται στο μονοπάτι `src/main/java/<Όνομα της Δραστηριότητας>.java`.
- **src/main/res** Περιέχει πόρους της εφαρμογής, όπως drawable αρχεία, αρχεία τοπολογίας (Layout) και τιμές αλφαριθμητικών (String Values).
- **src/main/res/anim/** Αφορά XML αρχεία που μεταγλωττίζονται σε αντικείμενα κινουμένων σχεδίων (Animation Objects).
- **src/main/res/color/** Αφορά XML αρχεία που περιγράφουν τα χρώματα.
- **src/main/res/drawable/** Αφορά αρχεία bitmap (.png, .jpeg ή gif), 9-Patch αρχεία εικόνας και XML αρχεία που περιγράφουν σχήματα (Drawable).
- **src/main/res/mipmap/** Αφορά το εικονίδιο του εκκινήτη (Launcher).
- **src/main/res/layout** Αφορά XML αρχεία τα οποία μεταγλωττίζονται σε τοπολογίες ολόκληρης της οθόνης
- **src/main/res/menu/** Αφορά XML αρχεία που ορίζουν τα μενού της εφαρμογής.
- **src/main/res/xml/** Αφορά τα διάφορα αρχεία XML που διαμορφώνουν τα στοιχεία της εφαρμογής.
- **src/main/AndroidManifest.xml** Το αρχείο ελέγχου που περιγράφει τη φύση της εφαρμογής και κάθε ένα από τα συστατικά του. Εκεί δηλώνονται όλα τα δικαιώματα, οι βιβλιοθήκες, τα επίπεδα API υποστηρίζονται κλπ.
- **build.gradle** Περιέχει προσαρμόσιμες ιδιότητες για το σύστημα κατασκευής.

4.2.1 Δημιουργία Android Project

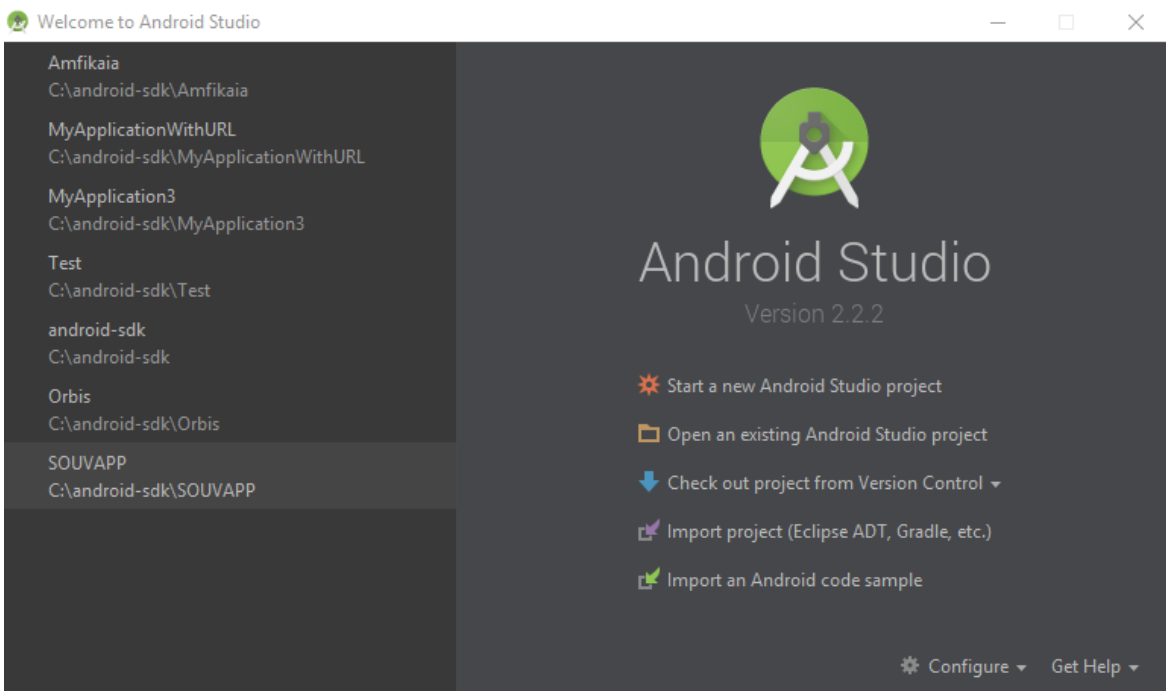
Τα βήματα για την δημιουργία ενός νέου έργου (Project) είναι τα παρακάτω:

1. Στον Package Explorer του Android Studio επιλέγουμε File→New και κάντε κλικ στην επιλογή “New Project”.



Σχήμα 5.1: Τρόπος δημιουργίας Android Project

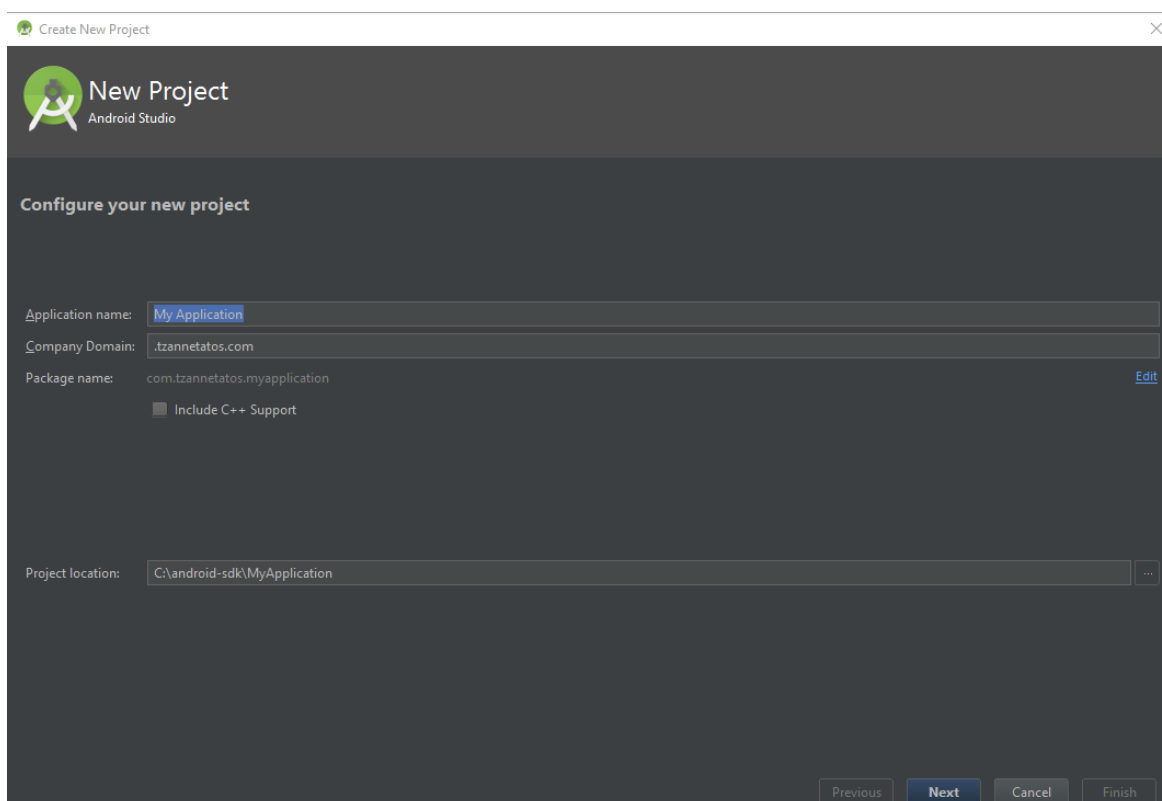
Επίσης υπάρχει και ο τρόπος με την επιλογή “Start a new Android Studio project” στο παράθυρο που εμφανίζεται κατά την εκκίνηση του Android Studio, για να δημιουργήσουμε ένα νέο Android Project όπως βλέπουμε παρακάτω.



Σχήμα 5.2: Δεύτερος τρόπος δημιουργίας Android Project

Μετά την επιλογή του new Android Studio Project θα πρέπει να συμπληρώσουμε τα 3 εξής πεδία που θα μας εμφανιστούν:

- Application name (Το όνομα της εφαρμογής)
- Company Domain (Το Domain Name χρησιμοποιείται από το Android Studio για να δημιουργήσετε ένα όνομα πακέτου. Το όνομα του πακέτου είναι απλά ένα μοναδικό αναγνωριστικό για την εφαρμογή στο Play Store της Google. Μπορεί να είναι οτιδήποτε, αρκεί να είναι μοναδικό.)
- Και τέλος το Project Location το οποίο είναι το όνομα του μοναπατιού που θέλουμε να βρίσκεται το Project που θα δημιουργήσουμε.

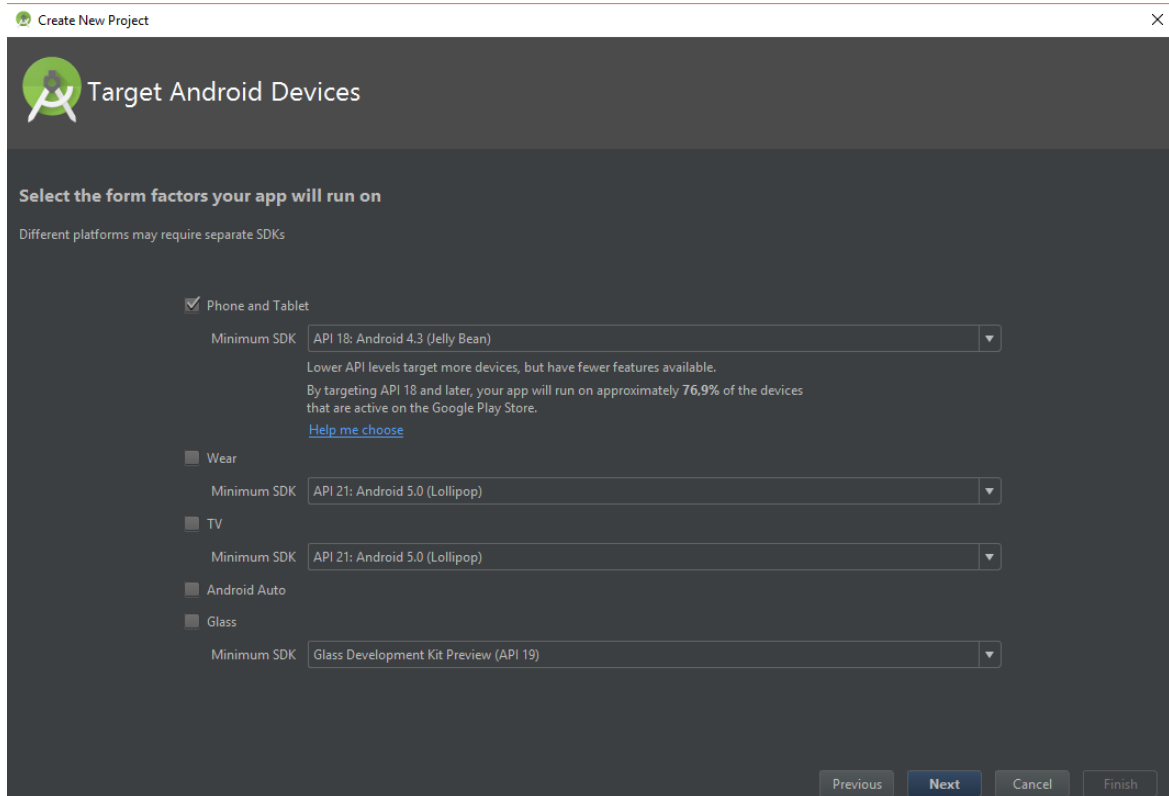


Σχήμα 5.3: Συμπλήρωση πεδίων του Android Project

Αφού συμπληρώσουμε τα πεδία πατάμε Next. Το επόμενο παράθυρο το οποίο μας εμφανίζεται μας ζητάει να επιλέξουμε σε τι τύπους συσκευών επιθυμούμε να

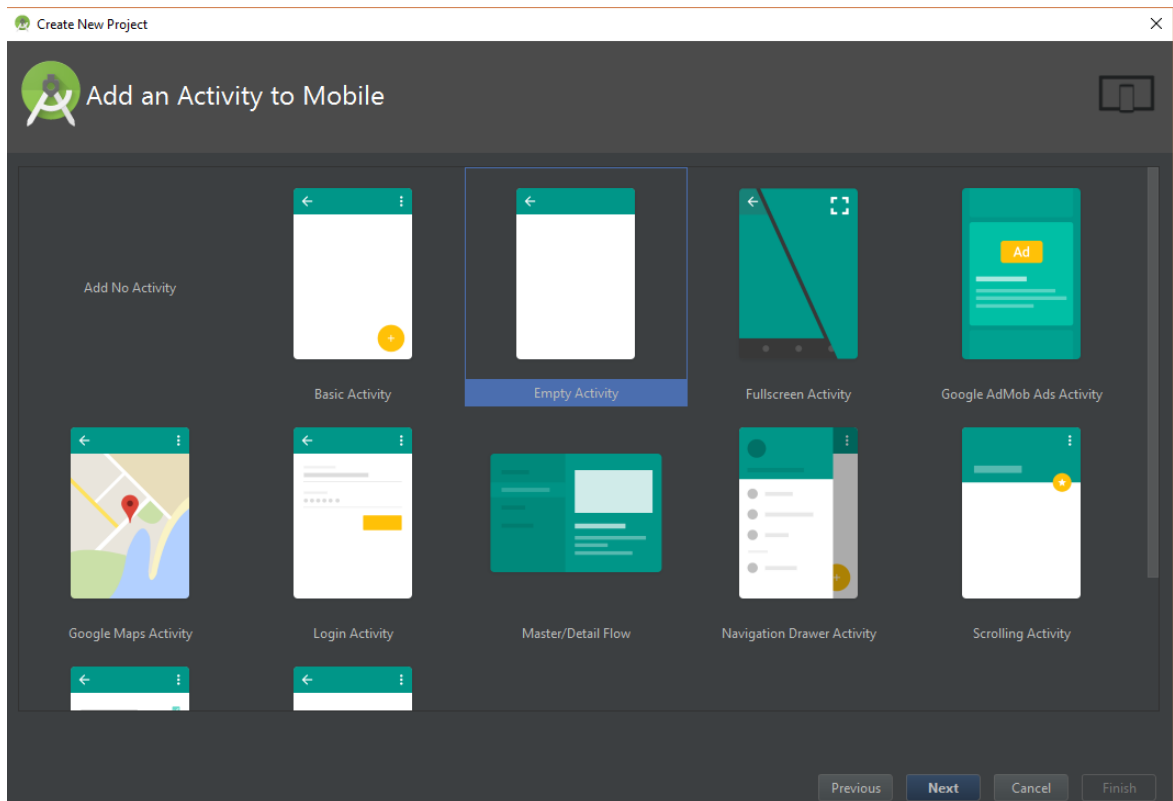
Ανάπτυξη εφαρμογής βοτανικού κήπου σε περιβάλλον Android με χρήση σάρωσης QR Code

είναι συμβατή η εφαρμογή μας, από την ελάχιστη έκδοση έως αυτή που επιθυμούμε. Έπειτα επιλέγουμε Next.



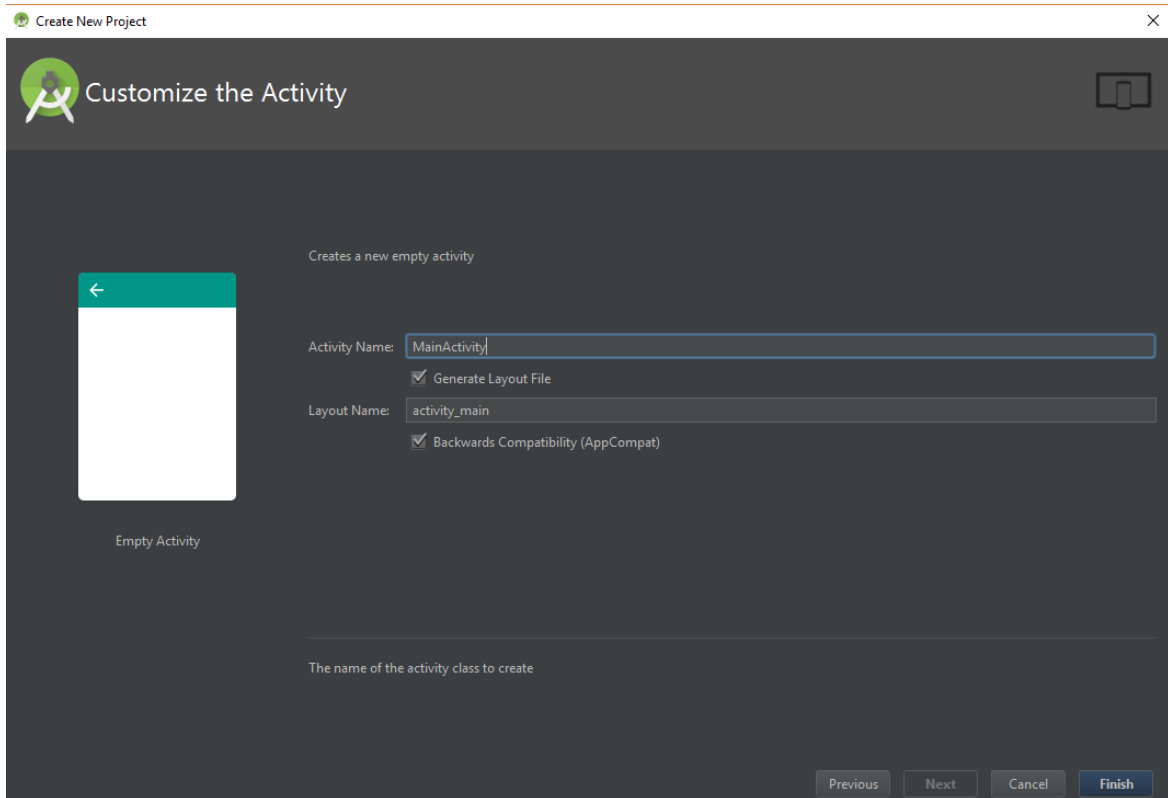
Σχήμα 5.4: Επιλογή μορφής παραγόντων

Αφού πατήσουμε Next το επόμενο και προτελευταίο παράθυρο που μας εμφανίζεται μας ζητάει να καθορίσουμε τον τύπο της δημιουργημένης δραστηριότητας. Συνήθως επιλέγεται η κενή δραστηριότητα (Blank Activity) ,την οποία χρησιμοποιούμε σαν αφετηρία για την ανάπτυξη της εφαρμογής μας.



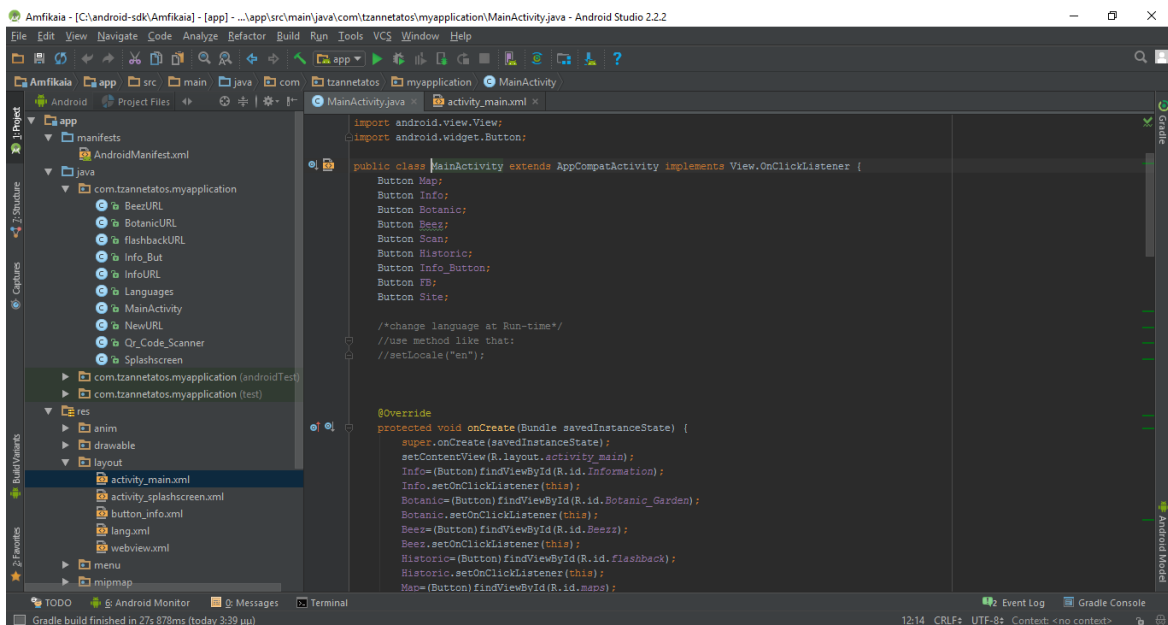
Σχήμα 5.5: Επιλογή τύπου δραστηριότητας

Αφού επιλέξουμε το Next μας εμφανίζει το τελευταίο παράθυρο στο οποίο δίνουμε όνομα στη δημιουργημένη κύρια δραστηριότητα και στο περιγραφικό αρχείο xml το οποίο περιέχει τη διάταξη της οθόνης για το χρήστη.



Σχήμα 5.6: Προσαρμογή δραστηριότητας

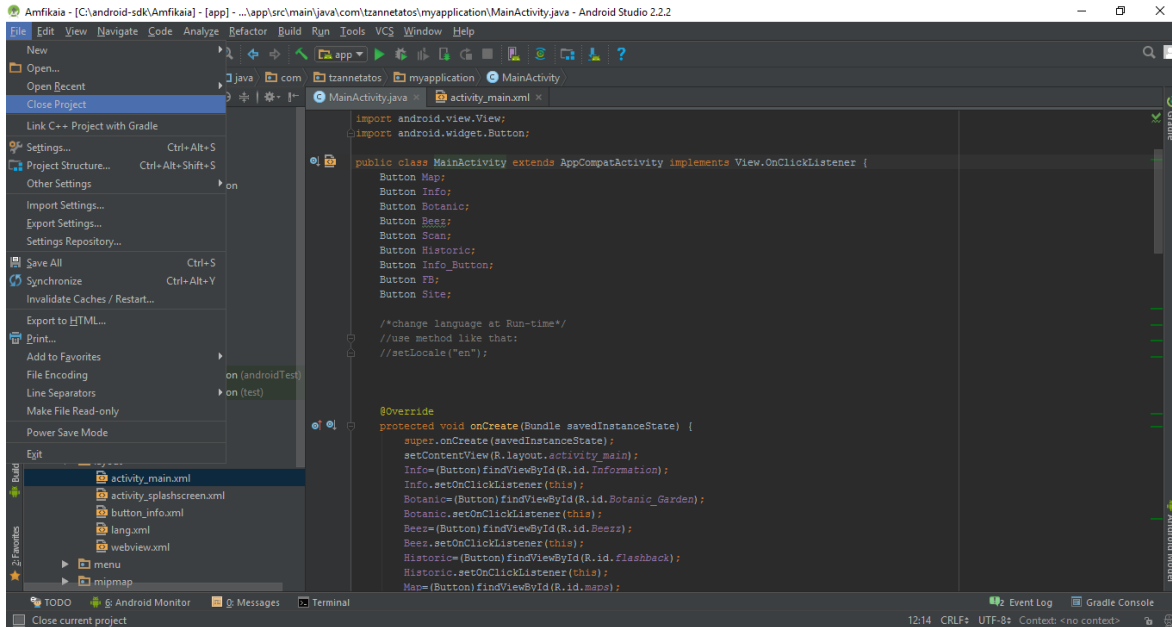
Με την επιλογή Finish τερματίζεται η διαδικασία δημιουργίας νέου έργου και στην οθόνη του υπολογιστή εμφανίζεται το περιβάλλον ανάπτυξης.



Σχήμα 5.7: Περιβάλλον ανάπτυξης

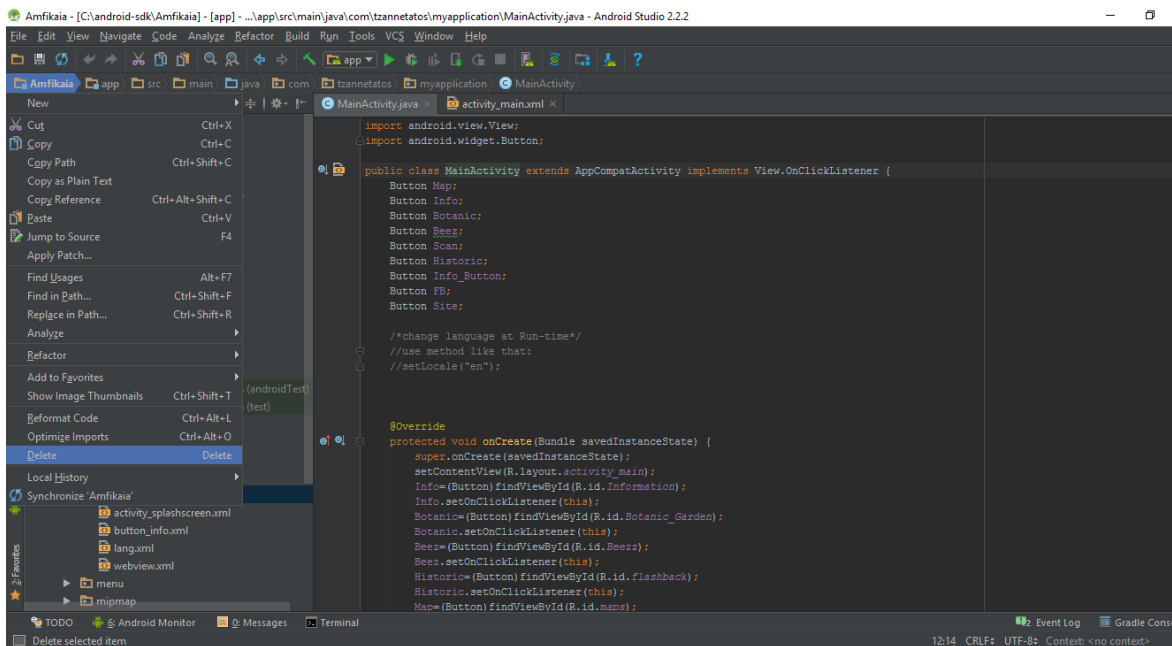
4.2.2 Διαγραφή Android Project

Για την διαγραφή ενός Android Project πρέπει πρώτα να το κλείσουμε, όπως βλέπουμε και στην παρακάτω εικόνα επιλέγουμε File→Close Project.



Σχήμα 5.8: Κλείσιμο Project

Έπειτα κάνουμε δεξί κλικ στην πρώτη καρτέλα με το όνομα του Project μας και επιλέγουμε το Delete.



Σχήμα 5.9: Διαγραφή Project

ΚΕΦΑΛΑΙΟ 5

ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή μας 'ΑΜΦΙΚΑΙΑ' είναι το αποτέλεσμα διαφόρων κλάσεων και των αντίστοιχων XML αντικειμένων τους. Κάθε λειτουργία της , δηλαδή , αποτελείται από τον αλγόριθμο που χρησιμοποιούμε για την αλληλουχία των αντιδράσεων ανάλογα με τις επιλογές του χρήστη αλλά και το σχεδιασμό της εμφάνισης της εφαρμογής σε αντιστοίχιση με την κάθε κλάση. Χρησιμοποιούμε βιβλιοθήκες ενσωματωμένες στην γλώσσα JAVA μαζί με τη χρήση βάσης δεδομένων από το Google Drive και χρήση τοποθεσίας στους χάρτες του Google Maps. Οι δυο αυτές εφαρμογές αποτελούν το λογισμικό Android νέας γενιάς με αποτέλεσμα να είναι προσβάσιμες σε όλους τους χρήστες που έχουν πρόσβαση στο διαδίκτυο.

Επίσης , η εφαρμογή θα αξιοποιεί την αποκωδικοποίηση barcodes/QR codes μέσω της βιβλιοθήκης ZXing και την απεικόνιση link στο διαδίκτυο ή συνδεση στη βάση δεδομένων μας 'ΑΜΦΙΚΑΙΑ' (φάκελος στο Google Drive που έχει γίνει διαμοιρασμός στοιχείων του). Εμφανίζει αποτέλεσμα στην οθονη της εφαρμογής μας (π.χ. με χρήση των βιβλιοθηκών NavUtils και WebView) ως ενσωματωμένος browser (π.χ. Mozilla Firefox App) και FileViewer . Για να γίνουν τα παραπάνω κατανοητά θα γίνει επεξήγηση κάθε κλάσης ξεχωριστά αλλά και σε σχέση μεταξύ των υπολοίπων μαζί με τα αρχεία που αποτελούνται αλλά και σε όλα τα μέρη τα οποία είναι κομβικά στη πορεία της δημιουργίας της εφαρμογής μας στο Android Studio.

ΣΗΜΕΙΩΣΗ : Οι γραμμές κώδικα που επαναλαμβάνονται και σε άλλα σημεία της εφαρμογής μας δεν είναι απαραίτητο να επεξηγούνται ξανά επειδή εννοούνται .

5.1 Κλάση MainActivity

1. `package com.tzannetatos.myapplication; // Το μοναδικό αναγνωριστικό για την εφαρμογή στο Play Store της Google`

2. `import android.content.Intent;` // Ορίζεται η βιβλιοθήκη *Intent* (πρόθεση), χρησιμεύει στην έναρξη των δραστηριοτήτων για την σύνδεση μεταξύ τους.
3. `import android.net.Uri;` // Ορίζεται βιβλιοθήκη για την δημιουργία ενός URL
4. `import android.support.v7.app.AppCompatActivity;` //
5. `import android.os.Bundle;` // Μια χαρτογράφηση από *String* κλειδιά σε διάφορες τιμές
6. `import android.view.Menu;` // Ορίζεται η βιβλιοθήκη για το μενού
7. `import android.view.MenuItem;` // Ορίζεται η βιβλιοθήκη για τα αντικείμενα του μενού
8. `import android.view.View;` // Η κατηγορία αυτή αποτελεί το βασικό δομικό στοιχείο για τα στοιχεία διεπαφής μεταξύ χρήστη-εφαρμογής
9. `import android.widget.Button;` // Αντιπροσωπεύει *push-buttons* που μπορούν να πατηθούν για να εκτελεστεί μια ενέργεια
10. `public class MainActivity extends AppCompatActivity implements View.OnClickListener` { // Η κύρια *Activity* όπου επεκτείνεται με την *AppCompatActivity* και υλοποιείται από την λειτουργία *OnClickListener*
11. `Button Map;` // *Push button* Χάρτη
12. `Button Info;` // *Push button*
13. `Button Botanic;`// *Push button*
14. `Button Beez;` // *Push button*
15. `Button Scan;` // *Push button*
16. `Button Historic;` // *Push button*
17. `Button Info_Button;` // *Push button*
18. `Button FB;` // *Push button*
19. `Button Site;` // *Push button*

20. `@Override` // Είναι ένας υπαινιγμός για τον *compiler* για να παρακάμψει τη μέθοδο της μητρικής κατηγορίας
21. `protected void onCreate(Bundle savedInstanceState)` { // Προστατευόμενη σημαίνει ότι είναι ορατή μέσα από την τάξη και τις τάξεις που κληρονομεί, `void` σημαίνει ότι δεν έχει τιμή επιστροφής

```
22. super.onCreate(savedInstanceState); // Λέμε στον Dalvik VM να τρέξει τον
    κώδικα μας επιπροσθέτως με τον υπάρχοντα κώδικα της onCreate()
23. setContentView(R.layout.activity_main); // Την καλούμε για να επιλέξουμε
    το Design (R=Resource,layout=design,main=xml αρχείο)
24. Info=(Button)findViewById(R.id.Information); // Δηλώνουμε το button
    Information
25. Info.setOnClickListener(this); // Δημιουργεί μια ανώνυμη υποκλάση
    OnClickListener στην παράμετρο(this)
26. Botanic=(Button)findViewById(R.id.Botanic_Garden);
27. Botanic.setOnClickListener(this);
28. Beez=(Button)findViewById(R.id.Beezz);
29. Beez.setOnClickListener(this);
30. Historic=(Button)findViewById(R.id.flashback);
31. Historic.setOnClickListener(this);
32. Map=(Button)findViewById(R.id.maps);
33. Map.setOnClickListener(this);
34. Scan=(Button)findViewById(R.id.Qr_Code);
35. Scan.setOnClickListener(this);
36. Info_Button=(Button)findViewById(R.id.InfoButton);
37. Info_Button.setOnClickListener(this);
38. FB=(Button)findViewById(R.id.fbButton);
39. FB.setOnClickListener(this);
40. Site=(Button)findViewById(R.id.Site_Button);
41. Site.setOnClickListener(this);
42. }
43. public void onClick(View view) { // Δημιουργούμε μια συνάρτηση void που
    αναλόγως την επιλογή δρα ανάλογα
44. switch (view.getId()) { // Ανάλογα το κουμπί που θα πατηθεί επιλέγεται το
    αντικείμενο που θέλουμε το οποίο μας μεταφέρει στην ανάλογη κλάση
45. case R.id.Information:
46. Intent intent = new Intent(MainActivity.this, InfoURL.class); // Στην
    περίπτωση που πατηθεί button Information μας μεταφέρει στην κλάση
    InfoURL
```

```
47. startActivity(intent); // Σύνδεση
48. break; // Διακοπή ροής προγράμματος

49. case R.id.Botanic_Garden:
50. Intent intent1 = new Intent(MainActivity.this, BotanicURL.class);
51. startActivity(intent1);
52. break;

53. case R.id.Beezz:
54. Intent intent2 = new Intent(MainActivity.this, BeezURL.class);
55. startActivity(intent2);
56. break;

57. case R.id.flashback:
58. Intent intent3 = new Intent(MainActivity.this, flashbackURL.class);
59. startActivity(intent3);
60. break;

61. case R.id.maps:
62. double latitude = 38.611575; // Δήλωση γεωγραφικού πλάτους
63. double longitude = 22.641213; // Δήλωση γεωγραφικού μήκους
64. String label = "Amfikaia Hotel"; // Δήλωσε τίτλου ετικέτας
65. String uriBegin = "geo:" + latitude + "," + longitude; // Συγχώνευση και των
    δύο σε μία μεταβλητή
66. String query = latitude + "," + longitude + "(" + label + ")"; // Δήλωση
    τοποθεσίας στο GoogleMaps
67. String encodedQuery = Uri.encode(query); // Κωδικοποιεί τις παραμέτρους
    που του δόθηκαν παραπάνω
68. String uriString = uriBegin + "?q=" + encodedQuery + "&z=16";
69. Uri uri = Uri.parse(uriString); // Ονομάζει Uri την τοποθεσία του
    αγροκτήματος
70. Intent intent7 = new Intent(android.content.Intent.ACTION_VIEW, uri); //
    Μας μεταφέρει στην τοποθεσία με την χρήση GoogleMaps
71. startActivity(intent7);
```



```
72.break;

73.case R.id.Qr_Code:
74.Intent intent5 = new Intent(MainActivity.this, Qr_Code_Scanner.class);
75.startActivity(intent5);
76.break;
77.case R.id.InfoButton:
78.Intent intent6 = new Intent(MainActivity.this, Info_But.class);
79.startActivity(intent6);
80.break;
81.case R.id.fbButton:
82.String YourPageURL =
    "https://www.facebook.com/amfikaia/?ref=ts&fref=ts"; // Δηλώνουμε σε
    μεταβλητή τον ιστότοπο του αγροκτήματος
83.Intent browserIntent = new
    Intent(Intent.ACTION_VIEW,Uri.parse(YourPageURL)); // Βάζουμε σε μια
    μεταβλητή browserIntent τον ιστότοπο που θέλουμε να ανοίξει από την
    default εφαρμογή του Internet
84.startActivity(browserIntent); // Άνοιγμα της σελίδας του Fb
85.break;
86.case R.id.Site_Button:
87.String YourPageURL2 = "http://www.amfikaia.gr/"; // Official site page
88.Intent browserIntent2 = new Intent(Intent.ACTION_VIEW,
89.Uri.parse(YourPageURL2));
90.startActivity(browserIntent2);
91.break;

92.}
93.}

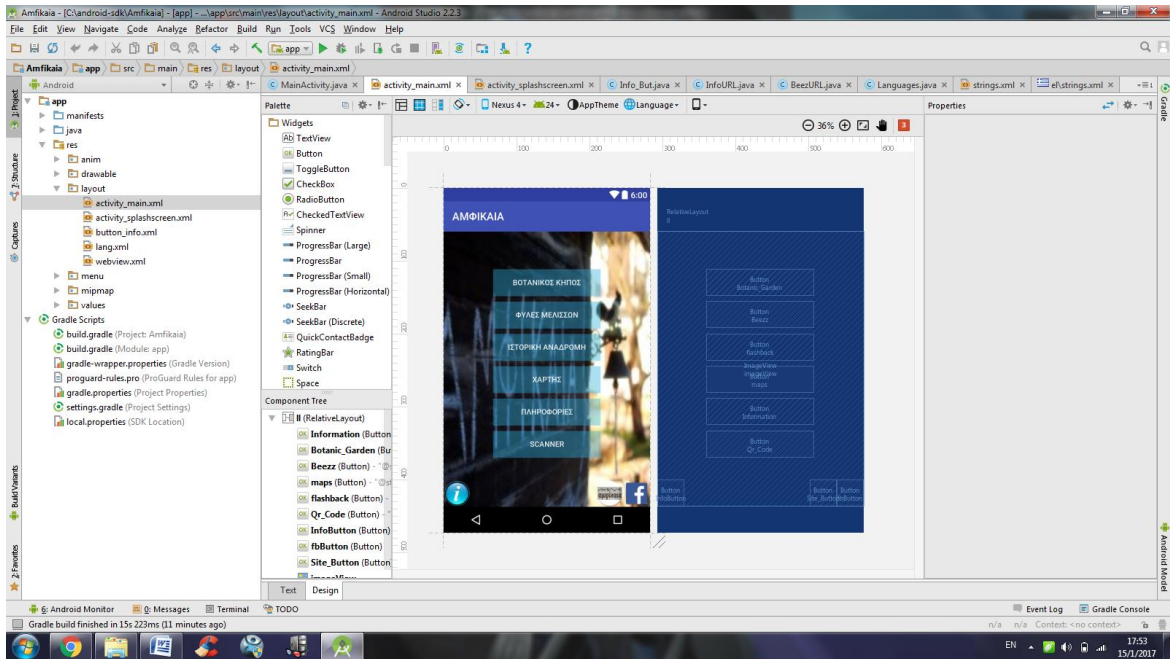
94.@Override //Εξηγούμε στον compiler ότι παρακάμπτουμε την θηγατρική
    διεργασία
95.public boolean onCreateOptionsMenu(Menu menu) { // Δημιουργία του
    Menu. Αυτό προσθέτει στοιχεία στη γραμμή δράσης.
```

```
96. getMenuInflater().inflate(R.menu.main, menu); // Παίρνει τις παραμέτρους
    από το main.xml που υπάρχει στην διαδρομή res/menu
97. return true; // Τερματισμός συνάρτησης
98. }

99. @Override //Εξηγούμε στον compiler ότι παρακάμπτουμε την θηγατρική
    διεργασία
100. public boolean onOptionsItemSelected(MenuItem item) { //Θέτουμε μπάρα
    με μενου δύο επιλογών
101. int id = item.getItemId();//Καταχωρείται σε μεταβλητή η επιλογή που θα
    πατηθεί
102. if (id == R.id.action_settings) {//Πρωτη επιλογή γλώσσες
103. Intent intent8 = new Intent(MainActivity.this, Languages.class); //
    Μεταφερόμαστε από την κύρια κλάση μας στην κλάση languages
104. startActivity(intent8);//Εκκίνηση languages.java
105. return true;
106. }else if (id == R.id.exit) {//Δεύτερη επιλογή έξοδος
107. finishAffinity();//Τερματίζει η κύρια δραστηριότητα και όλες οι συγγενικές
    της
108. // MainActivity.this.finish(); Θα μπορούσε να χρησιμοποιηθει και αυτή η
    συνάρτηση με ορισμενες διαφορές στην λειτουργία της
109. }

110. return super.onOptionsItemSelected(item);
111. }
112. @Override
113. public void onBackPressed() {//Όταν είμαστε στην κύρια κλάση μας και
    πατηθεί το κουμπί "πίσω" το οποίο υπάρχει εργοστασιακά σε κάθε συσκευή
114. finishAffinity();
115. }
116. }
```

5.1.1 Αρχείο Layout XML activity_main



Σχήμα 6.0: Activity main XML

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<RelativeLayout // Το RelativeLayout είναι μια ομάδα που εμφανίζει το παιδί από σχετικές θέσεις. Η θέση της κάθε προβολής μπορεί να καθοριστεί όπως σε σχέση με συγγενικά στοιχεία (όπως ως προς τα αριστερά-του ή κάτω από μια άλλη άποψη) ή σε θέσεις σε σχέση με το γονικό RelativeLayout περιοχή (όπως ευθυγραμμίζονται προς τα κάτω, αριστερά ή κέντρο)`
3. `xmlns:android=http://schemas.android.com/apk/res/android // Αντί της κλήσης του android:id το XML χρησιμοποιεί την παραπάνω διαδρομή σαν μοναδική. Στην πραγματικότητα αυτή η σελίδα δεν υπάρχει (είναι URI και όχι URL) αλλά μερικές φορές είναι μια διεύθυνση URL που εξηγεί το χώρο ονομάτων (namespace) που χρησιμοποιείται`
4. `xmlns:tools=http://schemas.android.com/tools // Με αυτή την εντολή μπορεί να αναγνωρίσει επιπλέον namespace, τα εργαλεία, τα οποία δεν είναι το προεπιλεγμένο namespace, έτσι όταν αναφέροντας στοιχεία ή χαρακτηριστικά που ορίζονται εκεί, θα πρέπει να προσθέσουμε τα εργαλεία`
5. `android:layout_width="match_parent" // Για να διασφαλίσουμε ότι η διάταξη θα είναι ευέλικτη και θα προσαρμόζεται σε διαφορετικά μεγέθη οθόνης θα`

χρειαστεί να χρησιμοποιήσουμε την εντολή “match_parent” ή “wrap_content”, για το πλάτος και το ύψος. Στην συγκεκριμένη περίπτωση το match_parent κάνει το στοιχείο να επεκταθεί για να ταιριάζει με το μέγεθος του γονέα.

6. android:layout_height="match_parent"

7. tools:context="com.tzannetatos.myapplication.MainActivity"

8. android:id="@+id/II"

9. android:clickable="true"> //

10. <Button // Δημιουργούμε ένα νέο Button

11. style="?android:attr/buttonStyleSmall"

12. android:layout_width="wrap_content"

13. android:layout_height="wrap_content"

14. android:text="@string/Information" // Στο αρχείο string έχουμε ήδη δηλώσει το όνομα του button για αυτό γράφουμε την διαδρομή

15. android:id="@+id/Information" // Ορίζουμε το μοναδικό Id με όνομα Information

16. android:layout_below="@+id/maps"

17. android:layout_alignStart="@+id/maps" // Ορίζουμε ότι ξεκινάει από εκεί που ξεκινάει και το button maps

18. android:layout_alignEnd="@+id/maps" // Ορίζουμε τα όρια του να σταματάνε όπου και του button maps

19. android:height="50dp"

20. android:enabled="true"

21. android:background="@color/wallet_highlighted_text_holo_dark"

22. android:layout_marginBottom="10dp" // Ορίζουμε την απόσταση που θα υπάρχει ανάμεσα στα button

23. android:textColor="@color/cast_expanded_controller_progress_text_color"

24. />

25. <Button

26. style="?android:attr/buttonStyleSmall"

27. android:layout_width="wrap_content"

```
28. android:layout_height="wrap_content"
29. android:text="@string/Botanic_Garden" // Στο αρχείο string έχουμε ήδη
δηλώσει το όνομα του button για αυτό γράφουμε την διαδρομή
30. android:id="@+id/Botanic_Garden"
31. android:layout_alignParentTop="true" // Ορίζουμε το button αυτό να είναι στην
κορυφή, δηλαδή να ξεκινάμε από αυτό
32. android:layout_centerHorizontal="true"
33. android:layout_marginTop="70dp" // Ορίζουμε την απόσταση που θα έχει το
button το layout
34. android:height="50dp"
35. android:width="200dp"
36. android:background="@color/wallet_highlighted_text_holo_dark" // Το χρώμα
που θα είναι στο background του button
37. android:layout_marginBottom="10dp" // Ορίζουμε την απόσταση που θα
υπάρχει ανάμεσα στα button
38. android:textColor="@color/cast_expanded_controller_progress_text_color" //
Το χρώμα των γραμμάτων
39. />
40. <Button
41. ...
42. android:text="@string/Beezz"
43. android:id="@+id/Beezz"
44. ...
45. />

54. <Button
55. ...
56. android:text="@string/maps"
57. android:id="@+id/maps"
58. ... />

67. <Button
68. ...
```

69. android:text="@string/flashback"

70. android:id="@+id/flashback"

71. ...

72. />

81. <Button

82. ...

83. android:text="@string/Qr_Code"

84. android:id="@+id/Qr_Code"

85. ...

86. />

94. <Button

95. android:layout_width="50dp"

96. android:layout_height="50dp"

97. android:id="@+id/InfoButton"

98. android:background="@mipmap/information_button" // Ορίζουμε να έχει ως background μια εικόνα που έχουμε ήδη στον φάκελο mipmap με το όνομα που βλέπουμε

99. android:layout_alignParentStart="true"

100. android:layout_alignParentBottom="true" />

101. <Button

102. ...

103. android:id="@+id/fbButton"

104. android:background="@mipmap/fb"

105. ... />

108. <Button

109. ...

110. android:layout_toStartOf="@+id/fbButton"

111. android:id="@+id/Site_Button"

```
112. android:background="@mipmap/site_button" />
```

```
115. <ImageView
```

```
116. android:layout_width="fill_parent"
```

```
117. android:layout_height="fill_parent"
```

```
118. android:id="@+id/imageView"
```

```
119. android:background="@drawable/amfikaia" // Καλούμε από τον φάκελο  
drawable την εικόνα με όνομα amfikaia ώστε να είναι στο background
```

```
120. android:layout_alignParentStart="true" />
```

```
121. </RelativeLayout>
```

5.2 Κλάση SplashScreen

Έχοντας συνθέσει το βασικό μενού της εφαρμογής μας θελήσαμε να προσθεσουμε ένα εικονικό εφέ στην εκκίνισή της έτσι ώστε να κερδίζει το ενδιαφέρον στον χρήστη και να εμπλουτιστεί το περιεχόμενο της. Έχουμε εκχωρίσει εικόνα για αυτόν τον σκοπό (amfikaia.jpg).

1. package com.tzannetatos.myapplication; *//Αναγράφεται σε κάθε κλάση*
2. import android.app.Activity; *//Βιβλιοθήκη που χρησιμοποιεί μεθόδους από την βασική μας κλάση και επέκταση με τη θηγατρική*
3. import android.content.Intent;
4. import android.graphics.PixelFormat; *//Βιβλιοθήκη για χρήση γραφικών στο πρόγραμμα*
5. import android.os.Bundle;
6. import android.view.Window; *//Παρέχει βασικές πολιτικές διεπαφής χρήστη ως προς την εικόνα (background, title area, default key processing)*
7. import android.view.animation.Animation; *//Παρομοίως*
8. import android.view.animation.AnimationUtils;
9. import android.widget.ImageView;
10. import android.widget.LinearLayout;

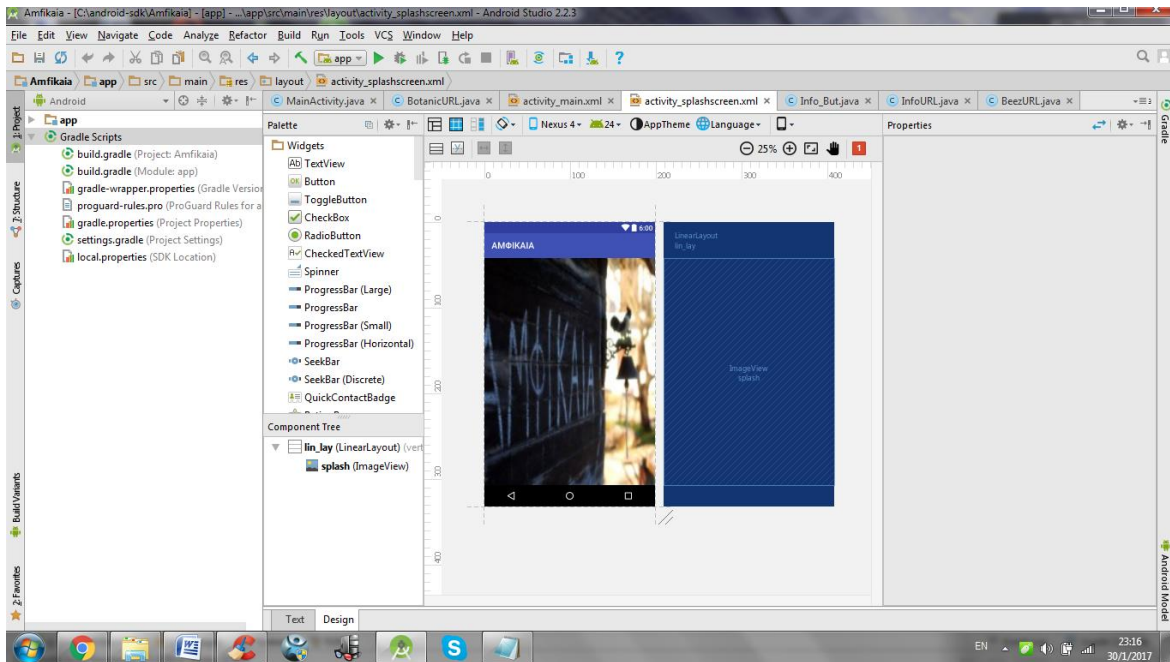
```
11. public class Splashscreen extends Activity {//Επέκταση μεθόδων και  
    συναρτήσεων από την μητρική μας κλάση(MainActivity)  
12. public void onAttachedToWindow() {  
13. super.onAttachedToWindow();  
14. Window window = getWindow();//Συλλέγει την εικόνα της δραστηριότητας  
    σε πραγματικό χρόνο  
15. window.setFormat(PixelFormat.RGBA_8888);//Pixel Format για το δείγμα  
16. }  
17. /** Called when the activity is first created. */  
18. Thread splashTread;
```

```
1. @Override  
2. public void onCreate(Bundle savedInstanceState) {  
3. super.onCreate(savedInstanceState);  
4. setContentView(R.layout.activity_splashscreen);//Επιλέγουμε  
    δραστηριότητα  
5. StartAnimations(); //Σειρά 7-27 σύνθεση εικονικού εφέ και εκκίνηση της  
    δραστηριότητας  
6. }  
7. private void StartAnimations() {  
8. Animation anim = AnimationUtils.loadAnimation(this, R.anim.alpha);  
9. anim.reset();  
10. LinearLayout l=(LinearLayout) findViewById(R.id.lin_lay);  
11. l.clearAnimation();  
12. l.startAnimation(anim);  
13. anim = AnimationUtils.loadAnimation(this, R.anim.translate);  
14. anim.reset();  
15. ImageView iv = (ImageView) findViewById(R.id.splash);  
16. iv.clearAnimation();  
17. iv.startAnimation(anim);  
18. splashTread = new Thread() {  
19. @Override  
  
20. public void run() {
```



```
21. try {
22. int waited = 0; // Splash screen pause time
23. while (waited < 3500) {
24. sleep(100);
25. waited += 100;
26. }
27. Intent intent = new Intent(Splashscreen.this, Languages.class);
    // Μεταφερόμαστε από το εικονικό εφέ στο μενού με τις γλώσσες που έχουμε
    // ως δυνατότητα επιλογής
28. intent.setFlags(Intent.FLAG_ACTIVITY_NO_ANIMATION);
29. startActivity(intent);
30. Splashscreen.this.finish();
31. } catch (InterruptedException e) { // do nothing
32. } finally {
33. Splashscreen.this.finish();
34. }
35. }
36. };
37. splashTread.start();
38. }
39. }
```

5.2.1 Αρχείο Layout XML splashscreen



Σχήμα 6.1: Xml Splashscreen

1. `<?xml version="1.0" encoding="utf-8" ?>`
2. `<LinearLayout` // Είναι ένα view group που ευθυγραμίζει όλα τα παιδιά σε μια κατεύθυνση, οριζόντια ή κάθετη
3. `xmlns:android=http://schemas.android.com/apk/res/android` // Αντί της κλήσης του `android:id` το XML χρησιμοποιεί την παραπάνω διαδρομή σαν μοναδική. Στην πραγματικότητα αυτή η σελίδα δεν υπάρχει (είναι URI και όχι URL) αλλά μερικές φορές είναι μια διεύθυνση URL που εξηγεί το χώρο ονομάτων (*namespace*) που χρησιμοποιείται
4. `android:layout_width="fill_parent"`
// Fill_parent=match_parent (Μετανομάστηκε για >API level 8). Η ρύθμιση της διάταξης ενός widget για να fill_parent θα την αναγκάσει να επεκταθεί για να καταλαμβάνει τόσο χώρο όσο είναι διαθέσιμη εντός του στοιχείου διάταξης
5. `android:layout_height="fill_parent"`
6. `android:id="@+id/lin_lay"` // Δημιουργούμε ένα αναγνωριστικό που ονομάζεται `lin_lay` στο χώρο ονομάτων της εφαρμογής μας
7. `android:gravity="center"` // Η εντολή αυτή "τακτοποιεί" το περιεχόμενο μέσα στην προβολή. Ουσιαστικά κεντράρει οριζοντίως και καθέτως το `ImageView`

8. `android:orientation="vertical">` // Δεν υπάρχει κάποιο όφελος στην συγκεκριμένη εντολή απλώς η στοίβαξει θα γινόταν κάθετα αν είχαμε κάποιο `view` και όχι οριζόντια
9. `<ImageView`
10. `android:layout_width="match_parent"` // Για να διασφαλίσουμε ότι η διάταξη θα είναι ευέλικτη και θα προσαρμόζεται σε διαφορετικά μεγέθη οθόνης θα χρειαστεί να χρησιμοποιήσουμε την εντολή `"match_parent"` ή `"wrap_content"`, για το πλάτος και το ύψος. Στην συγκεκριμένη περίπτωση το `match_parent` κάνει το στοιχείο να επεκταθεί για να ταιριάζει με το μέγεθος του γονέα.
11. `android:layout_height="match_parent"`
12. `android:id="@+id/splash"` // Σημαίνει ότι δημιουργούμε ένα αναγνωριστικό που ονομάζεται `splash` στο χώρο ονομάτων της εφαρμογής μας
13. `android:background="@drawable/amfikaia" />`
14. `</LinearLayout>`

5.3 Κλάση Languages

Σε αυτό το σημείο του προγραμματισμού μας θελήσαμε να δώσουμε έναν παγκόσμιο χαρακτήρα στην εφαρμογή ώστε να είναι κατανοητή από ένα μεγαλύτερο μέρος ανθρώπων προσθέτοντας την επιλογή της γλώσσας. Μετά το εικονικό εφέ της έναρξης ο χρήστης πηγαίνει αυτόματα σε μενού για την επιλογή γλώσσας (Αγγλικά, Γερμανικά, Γαλλικά ή Ελληνικά) αλλά και μετά την πρώτη επιλογή ο χρήστης μπορεί οποιαδήποτε στιγμή να την αλλάξει πάλι με κουμπί τοποθετημένο στη πάνω αριστερή γωνία του παραθύρου μας.

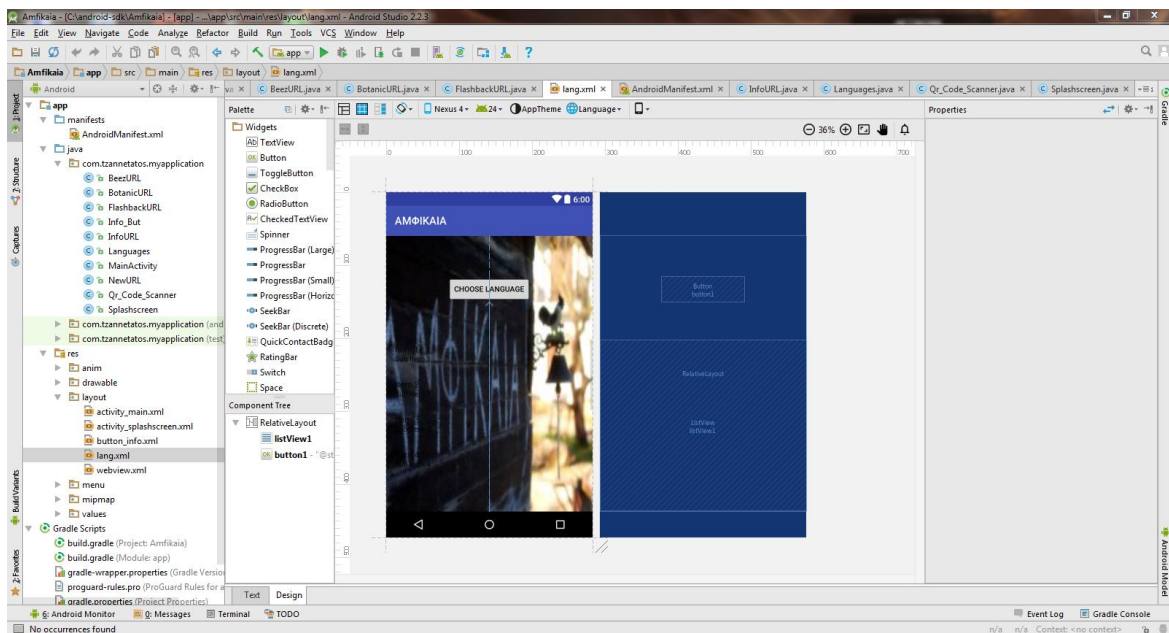
1. `package com.tzannetatos.myapplication;`
2. `import java.util.Locale;`
3. `import android.app.Activity;`
4. `import android.content.Intent;`
5. `import android.content.res.Configuration;`
6. `import android.os.Bundle;`
7. `import android.view.View;`

```
8. import android.widget.AdapterView;
9. import android.widget.ArrayAdapter;
10. import android.widget.ListView;
11. import android.widget.AdapterView.OnItemClickListener;
12. import android.widget.Toast;
13. import static android.R.id.text1;
14. public class Languages extends Activity {
15.     ListView lv1;
16.     String chosenlang="";
17.     int langpos;
18.     @Override
19.     protected void onCreate(Bundle savedInstanceState) {
20.         super.onCreate(savedInstanceState);
21.         setContentView(R.layout.lang);
22.         lv1=(ListView)this.findViewById(R.id.listView1);
23.         String[] langvalues =
            newString[]{"English","German","French","Greek"};
24.         ArrayAdapter<String> adapter = new
            ArrayAdapter<String>(this,android.R.layout.simple_list_item_checked,
            text1, langvalues);
25.         lv1.setAdapter(adapter);
26.         lv1=(ListView)findViewById(R.id.listView1);// By using setAdpater
            method in listview we an add string array in list.
27.         lv1.setOnItemClickListener(new OnItemClickListener() {
28.             public void onItemClick(AdapterView<?> parent, View view, int
                position, long id) {
29.                 chosenlang = (String) lv1.getItemAtPosition(position);
30.                 Toast.makeText(getApplicationContext(), chosenlang,
                    Toast.LENGTH_SHORT).show();
31.                 switch(position) { // Υπάρχουν 4 επιλογές γλώσσας, αν ο
                    χρήστης επιλέξει 0 τότε η γλώσσα στο menu θα είναι Αγγλικά, αν επιλέξει 1
                    τότε θα είναι γερμανικά, αν επιλέξει την επιλογή 2 θα είναι γαλλικά και τέλος
                    ελληνικά στην επιλογή 3
32.                     case 0:
```

```
33.     updateconfig("en");
34.     Intent intent = new Intent(Languages.this, MainActivity.class);
35.     startActivity(intent);
36.     break;
37. case 1:
38.     updateconfig("de");
39.     Intent intent1 = new Intent(Languages.this, MainActivity.class);
40.     startActivity(intent1);
41.     break;
42. case 2:
43.     updateconfig("fr");
44.     Intent intent2 = new Intent(Languages.this, MainActivity.class);
45.     startActivity(intent2);
46.     break;
47. case 3:
48.     updateconfig("el");
49.     Intent intent3 = new Intent(Languages.this, MainActivity.class);
50.     startActivity(intent3);
51.     break;
52. }
53.}
54.});
55.}
56.     public void updateconfig(String s)
57.     {
58.         String languageToLoad = s; // Αλλαγή γλώσσας
59.         Locale locale = new Locale(languageToLoad);
60.         Locale.setDefault(locale);
61.         Configuration config = new Configuration();
62.         config.locale = locale;
63.         getBaseContext().getResources().updateConfiguration(config,
            getBaseContext().getResources().getDisplayMetrics());
64.         Bundle tempBundle = new Bundle();
65.         onCreate(tempBundle);
```

```
66.     setTitle(R.string.app_name);
67.     invalidateOptionsMenu();
68. }
69. @Override
70. public void onBackPressed() {finishAffinity();}
71. }
```

5.3.1 Αρχείο Layout XML lang



Σχήμα 6.2: Xml Lang

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<RelativeLayout`
`xmlns:android="http://schemas.android.com/apk/res/android" // Το`
`RelativeLayout είναι μια ομάδα που εμφανίζει το παιδί από σχετικές θέσεις.`
`Η θέση της κάθε προβολής μπορεί να καθοριστεί όπως σε σχέση με`
`συγγενικά στοιχεία (όπως ως προς τα αριστερά-του ή κάτω από μια άλλη`
`άποψη) ή σε θέσεις σε σχέση με το γονικό RelativeLayout περιοχή (όπως`
`ευθυγραμμίζονται προς τα κάτω, αριστερά ή κέντρο)`
3. `android:layout_width="match_parent" // Για να διασφαλίσουμε ότι η διάταξη`
`θα είναι ευέλικτη και θα προσαρμόζεται σε διαφορετικά μεγέθη οθόνης θα`
`χρημαστεί να χρησιμοποιήσουμε την εντολή "match_parent" ή`
`"wrap_content", για το πλάτος και το ύψος. Στην συγκεκριμένη περίπτωση το`

match_parent κάνει το στοιχείο να επεκταθεί για να ταιριάζει με το μέγεθος του γονέα.

4. `android:layout_height="match_parent"`
5. `android:background="@drawable/amfikaia">` // Αντλεί από τον φάκελο *drawable* την εικόνα με όνομα *amfikaia* ώστε να εμφανίζεται στο *background*
6. `<ListView` // Εμφανίζονται διάφορα στοιχεία σε κάθετη κυλιόμενη λίστα. Τα στοιχεία της λίστας προστίθενται αυτόματα στη λίστα χρησιμοποιώντας έναν προσαρμογέα που τραβά το περιεχόμενο από μια πηγή, όπως έναν πίνακα ή μια βάση δεδομένων ερωτήματος και μετατρέπει κάθε είδους αποτέλεσμα σε μια προβολή που τοποθετείται σε μια λίστα.
7. `android:id="@+id/listView1"` // Δημιουργούμε ένα αναγνωριστικό για την *ListView* που ονομάζεται *listView1* στο χώρο ονομάτων της εφαρμογής μας
8. `android:layout_width="match_parent"`
9. `android:layout_height="wrap_content"`
10. `android:layout_below="@+id/button1"` // Ορίζουμε ένα `button` με αναγνωριστικό `button1` εντός της λίστας
11. `android:layout_centerHorizontal="true"`
12. `android:entries="@array/lang_array"` //
13. `android:layout_marginTop="70dp"`
14. `android:checkMark="?android:attr/listChoiceIndicatorMultiple"`
15. `android:checked="true">`
16. `</ListView>`
17. `<Button`
18. `android:id="@+id/button1"`
19. `android:layout_width="wrap_content"`
20. `android:layout_height="wrap_content"`
21. `android:layout_centerHorizontal="true"`
22. `android:layout_marginTop="75dp"`
23. `android:textColor="@color/black"`
24. `android:text="@string/change_lang" />`
25. `</RelativeLayout>`

5.4 Κλάση Qr_Code_Scanner

Ένα απο το πιο σημαντικά εργαλεία που χρησιμοποιήθηκαν για την πραγματοποίηση της εφαρμογής μας είναι ο σαρωτής QR Code. Είναι μια αυτόνομη κλάση που σκανάρει κώδικες QR (και άλλους γραμμικούς συμβολισμούς 2D barcode) με χρήση ZXing και αφού τους μεταγλωττίσει σε ανθρώπινη γλώσσα (στη περίπτωση μας Ελληνικά) μεταφέρει την λέξη στη κλάση NewURL.java . Για την δημιουργία των 21 Qr Codes, λόγω του ότι μας δώθηκαν 21 διαφορετικά είδη φυτών για επεξεργασία , επιλέξαμε μια δωρεάν διαδικτυακή εφαρμογή το QR Code Generator.

1. package com.tzannetatos.myapplication;
2. import android.app.AlertDialog;*//Δηλώνουμε τον τρόπο που το μήνυμα μπορεί εμφανιστεί μετά από κάποια ενέργεια*
3. import android.content.Intent;
4. import android.net.Uri; *//Είναι αμετάβλητο άρα και ασφαλές για τα νήματα*
5. import android.os.Bundle;
6. import android.support.v4.app.NavUtils; *// Παρέχει βοήθεια στη λειτουργικότητα για εφαρμογές που έχουν πρότυπα πλοήγησης Android UI.*
7. import android.support.v7.app.AppCompatActivity;*//Μπορεί να μην δηλωθεί αφού επεκτείνουμε τη κλάση μας απευθείας με τη βασική*
8. import android.util.Log; *// API για την αποστολή εξόδου καταγραφής*
9. import android.view.MenuItem;
10. import com.google.android.gms.appindexing.Action;
11. import com.google.android.gms.appindexing.AppIndex;
12. import com.google.android.gms.common.api.GoogleApiClient;*//10,11,12 χρήση διαδικτυακών εργαλείων και βιβλιοθήκης*
13. import com.google.zxing.Result;*// Μια βιβλιοθήκη η οποία υποστηρίζει αποκωδικοποίηση και παραγωγής των barcodes (όπως QR Code, PDF 417, EAN, UPC, Αζτέκων, Data Matrix, Codabar) μέσα σε εικόνες.*
14. import me.dm7.barcodescanner.zxing.ZXingScannerView;*//Βιβλιοθήκη για εμφάνιση σκάνερ σε συγχρονισμό με τη κάμερα*


```
15. public class Qr_Code_Scanner extends MainActivity implements
    ZXingScannerView.ResultHandler { //Επέκταση της κλάσης μας με τη
    θηγατρική-βασική και ενσωμάτωση της βιβλιοθήκης ZXing για χρήση της
16. private ZXingScannerView mScannerView;
17. /**
18. ATTENTION: This was auto-generated to implement the App Indexing API.
19. See https://g.co/AppIndexing/AndroidStudio for more information.
20. */
21. private GoogleApiClient client;
22. @Override
23. protected void onCreate(Bundle savedInstanceState) {
24. super.onCreate(savedInstanceState);
25. setContentView(R.layout.activity_main); //Αφορά την βασική μας κλάση
    MainActivity αφού θέλουμε να μην έχουμε ξεχωριστό αρχείο xml για τη
    Qr_Code_Scanner
26. // ATTENTION: This was auto-generated to implement the App Indexing
    API.
27. // See https://g.co/AppIndexing/AndroidStudio for more information.
28. client = new GoogleApiClient.Builder(this).addApi(AppIndex.API).build();
29. mScannerView = new ZXingScannerView(this);
30. setContentView(mScannerView);
31. mScannerView.setResultHandler(this);
32. mScannerView.startCamera(); //28-32 εκκίνηση κάμερας και συγκέντρωση
    φακού προσθετοντας στοιχεία του σκάνερ μέσα από αυτή
33. }
34. @Override
35. protected void onPause() {
36. super.onPause();
37. mScannerView.stopCamera();
38. } //35-38 χρήση κάμερας
39. @Override
40. public void handleResult(Result rawResult) {//Αποτέλεσμα μετά τη
    σάρωση και τη μεταγλώττιση. Δεν κάνει κάτι με το αποτέλεσμα
41. Log.e("handler", rawResult.getText()); // Κρατάει στο buffer την τιμή
```

```
42. Log.e("handler", rawResult.getBarcodeFormat().toString());
43. AlertDialog.Builder builder = new AlertDialog.Builder(this);
44. //1
45. if (rawResult.getText().equals("ΔΑΦΝΗ")) {//Συγκρίνει την τιμή που σκάναρε
    αν συμπίπτει με τη λέξη String "ΔΑΦΝΗ"
46. Intent intent = new Intent(Qr_Code_Scanner.this, NewURL.class);//Αν
    συμπίπτει τότε θα ξεκινήσει η κλάση NewURL.java
47. Qr_Code_Scanner.this.startActivity(intent); //Εκκίνηση NewURL
48. Intent apotelesma = new Intent(getApplicationContext(),
    NewURL.class);//Δηλώνουμε τη μεταβλητή apotelesma
49. apotelesma.putExtra("apotelesma",rawResult.getText());//Εκχώριση τιμής
    σαρωσης στη μεταβλητή apotelesma
50. startActivity(apotelesma);//Εκκίνηση εκχώρησης
51. } else if(rawResult.getText().equals("ΔΕΝΔΡΟΛΙΒΑΝΟ")){//Αλλιώς γίνονται
    οι αντίστοιχες ενέργειες για το "ΔΕΝΔΡΟΛΙΒΑΝΟ"
52. Intent intent = new Intent(Qr_Code_Scanner.this, NewURL.class);
53. Qr_Code_Scanner.this.startActivity(intent);
54. Intent apotelesma = new Intent(getApplicationContext(), NewURL.class);
55. apotelesma.putExtra("apotelesma",rawResult.getText());
56. startActivity(apotelesma);
57. } else if(rawResult.getText().equals("ΘΡΟΥΜΠΙ")){
58. ...//Όπως παραπάνω είναι παρόμοιες οι συνθήκες για το κάθε είδος φυτού

59. } else{ //45-59 Αν και οι 21 εναλλακτικές δεν συμπέσουν τότε η κλάση μας
    λειτουργεί σαν αναγνώστης QR Code και άλλων γραμμικών Bar Code σε
    δισδιάστατο πεδίο
60. builder.setTitle("Scan result :");//Ετικέτα εμφανιζόμενου αποτελέσματος
61. builder.setMessage(rawResult.getText());//Διαχείριση αποτελέσματος
62. AlertDialog alert1 = builder.create();//Δημιουργία αντικειμένου-συναγερμού
    alert1
63. alert1.show();//Εμφάνιση στην οθόνη
64. }
```

```
65.// If you would like to resume scanning, call this method below:
66.mScannerView.resumeCameraPreview(this);//Επανάληψη συγκέντρωσης
    κάμερας
67.}

68. @Override
69. public void onStart() {
70. super.onStart();//Εκκίνηση κάμερας σαρωτή

71. client.connect();//Σύνδεση διακομιστή κάμερας
72. Action viewAction = Action.newAction(
73. Action.TYPE_VIEW, // TODO: choose an action type.
74. "Qr_Code_Scanner Page", // TODO: Define a title for the content shown.
75. Uri.parse("http://host/path"),
76. // TODO: Make sure this auto-generated app URL is correct.
77. Uri.parse("android-app://com.tzannetatos.myapplication/http/host/path")
78.);
79. AppIndex.AppIndexApi.start(client, viewAction);
80.}

81. @Override
82. public void onStop() {
83. super.onStop();

84. Action viewAction = Action.newAction(
85. Action.TYPE_VIEW, // TODO: choose an action type.
86. "Qr_Code_Scanner Page", // TODO: Define a title for the content shown.
87. // Otherwise, set the URL to null.
88. Uri.parse("http://host/path"),
89. // TODO: Make sure this auto-generated app URL is correct.
90. Uri.parse("android-app://com.tzannetatos.myapplication/http/host/path")
91.);
92. AppIndex.AppIndexApi.end(client, viewAction);
93. client.disconnect();
```

```
94. //189-221 Αυτόματα ενσωματώνει κώδικα το Android Studio για να  
υλοποιηθεί το ευρετήριο εφαρμογών του API  
95.  
96. public boolean onOptionsItemSelected(MenuItem item) {  
  
97. //handle presses on the action bar items  
98. switch (item.getItemId()) {  
  
99. case android.R.id.home:  
100.     NavUtils.navigateUpFromSameTask(this);//Μεταφερόμαστε στη  
προηγούμενη δραστηριότητα δηλαδή στη MainActivity (βασική) όταν  
πατηθεί το δηλωμένο μας κουμπί  
101.     return true;  
102.     }  
103.     return super.onOptionsItemSelected(item);  
104.     }  
105.     @Override  
106.     public void onBackPressed() {  
107.         Intent intent5 = new  
108.         Intent(Qr_Code_Scanner.this,MainActivity.class);  
109.         //Επαναφορά στη βασική δραστηριότητα μετά το πάτημα του  
υπάρχοντος πίσω κουμπιού του κινητού  
110.         startActivity(intent5);  
111.         return;  
112.     }  
113.     }
```

5.5 Κλάση NewURL

Σε αυτό το σημείο υπήρξε αναγκαίο να βρούμε ένα εργαλείο ώστε να ανοίγουν οι ιστοτόποι και τα αρχεία ,τα οποία είναι αποθηκευμένα στο προσωπικό μας Google Drive , μέσα στην εφαρμογή . Επιλέξαμε να αποφύγουμε τη χρήση του ενσωματωμένου browser των Android συσκευών για το άνοιγμα των αρχείων

επειδή θα ήταν μη φιλικό προς τον χρήστη και επίπονο για τη μνήμη RAM της συσκευής. Επίσης γλυτώνουμε τον περιττό χώρο που θα κάλυπτε η εφαρμογή στο σκληρό δίσκο κάθε κινητού αφού τα περίπου 20Mb ως μέγεθος θα μπορούσαν να ξεπεραστούν , λόγω όγκου αρχείων, φτάνοντας ακόμα και το 1Gb. Έτσι , στη κλάση αυτή ανάλογα τη τιμή που κρατήσαμε από τη σάρωση ανοίγει μέσω διαδικτύου ανάλογο ισότοπο (πληροφορίες από Wikipedia και άλλα sites) εντός εφαρμογής "Amfikaia".

1. package com.tzannetatos.myapplication;
2. import android.os.Bundle;
3. import android.support.v4.app.NavUtils;
4. import android.webkit.WebView; *//Βιβλιοθήκη για βάση πάνω στην οποία μπορούμε να ανοίξουμε το δικό μας πρόγραμμα περιήγησης στο Web ή απλά να εμφανιστεί κάποιο διαδικτυακό περιεχόμενο μέσα στη δραστηριότητας μας.*
5. import android.webkit.WebViewClient; *//Κατασκευαστής webview*
6. import android.content.Intent;
7. public class NewURL extends Qr_Code_Scanner { *//Η κλάση NewURL έχει ως πατέρα της την Qr_Code_Scanner ώστε σύμφωνα με ποια τιμή σαρώθηκε να ξεκινάει ανάλογο link. Θέλουμε ο χρήστης να επιστρέφει πάλι στον scanner όταν τελειώνει με την ανάγνωση της σελίδας που άνοιξε ώστε να επαναλάβει πολλές φορές την ίδια διαδικασία με διαφορετικό QR code(διαφορετικό φυτό)*
8. private class MyWebViewClient extends WebViewClient {
9. @Override
10. public boolean shouldOverrideUrlLoading(WebView view, String url)
{//Δήλωση για άνοιγμα νέα σελίδας χωρίς τη χρήση του default browser
11. view.loadUrl(url);*//Άνοιγμα σελίδας και αναμονή για link*
12. return true;
13. }
14. }
15. WebView webview;
16. @Override

```
17. protected void onCreate(Bundle savedInstanceState) {
18. // TODO Auto-generated method stub
19. super.onCreate(savedInstanceState);
20. setContentView(R.layout.webview); //Θέτουμε πλαίσιο στην οθόνη που θα
    βρισκείται ο προβολέας μας
21. webView=(WebView)findViewById(R.id.webview1); //Δήλωση προβολέα μας
    webView τον οποίο έχουμε δηλώσει ως webView1
22. webView.setWebViewClient(new MyWebViewClient()); //Δήλωση
    κατασκευαστή
23. openURL(); //Εκκίνηση προβολέα άρα και συνάρτησης openURL
24. }

25. /** Opens the URL in a browser */
26. private void openURL() {

27. Bundle extras = getIntent().getExtras(); //Καλούμε τη συνάρτηση getIntent
    για να συνδέσουμε τις δύο συναρτήσεις και για να πάρουμε τη μεταβλητή
    που έχουμε κρατήσει καλούμε τη getExtras.
28. String i = extras.getString("apotelesma"); //Επειδή η τιμή της μεταβλητής
    extras μένει στην μνήμη για λίγο (buffer memory) την εκχωρούμε σε νέα
    μεταβλητή για πρακτικούς λόγους. Η τιμή μιας μεταβλητής δεν μεταφέρεται
    σε άλλη δραστηριότητα αλλιώς, παρά μόνο με αυτόν τον τρόπο (Γίνεται
    Global Variable)

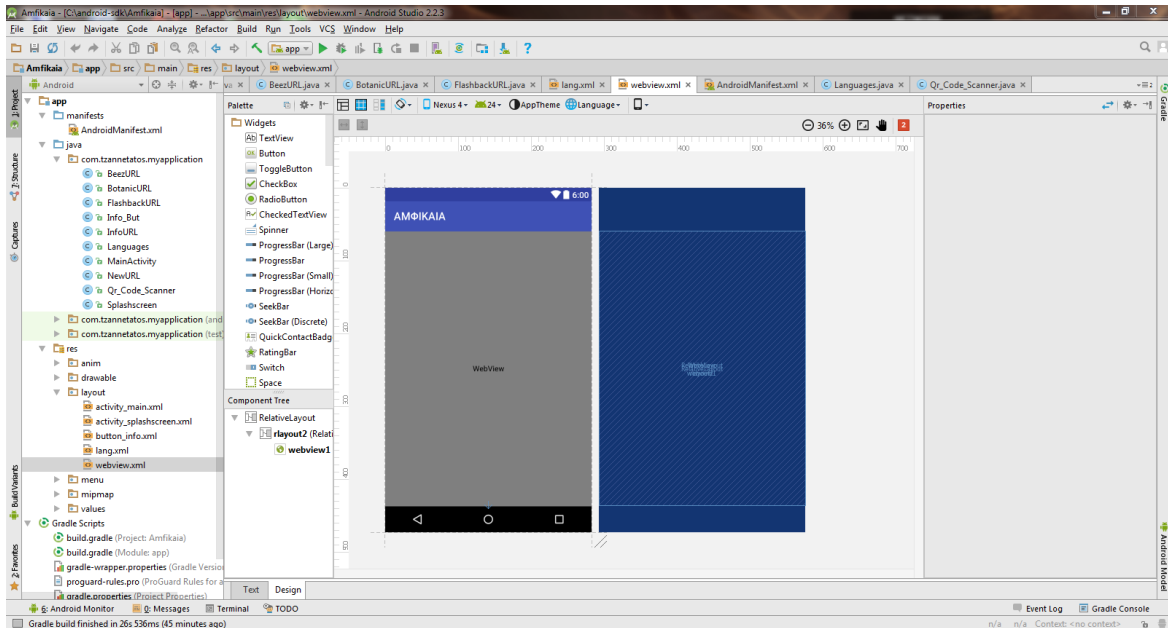
29. if (i.equals("ΔΑΦΝΗ")) { //Εάν το "i" είναι ίσο με τη λέξη "ΔΑΦΝΗ" ισχύει η
    συνθήκη
30. webView.loadUrl("https://el.wikipedia.org/wiki/%CE%94%CE%AC%CF%86
    %CE%BD%CE%B7_(%CF%86%CF%85%CF%84%CF%8C)"); //Φορτώνει
    συγκεκριμένο ιστότοπο και τον ανοίγει εντός του προβολέα της
    δραστηριότητάς μας
31. webView.getSettings().setJavaScriptEnabled(true); //Αντικαθιστά τον
    τρέχοντα χειριστή Java Script. Είναι μια εφαρμογή του WebViewClient.
    Χωρίς αυτή η σελίδα δεν προβάλλεται σωστά και παρουσιάζονται "Bugs"
```

```
32.}
33.else if(i.equals("ΔΕΝΔΡΟΛΙΒΑΝΟ")){
34.webview.loadUrl("https://el.wikipedia.org/wiki/%CE%94%CE%B5%CE%BD
    %CE%B4%CF%81%CE%BF%CE%BB%CE%AF%CE%B2%CE%B1%CE
    %BD%CE%BF");
35.webview.getSettings().setJavaScriptEnabled(true);

36.}
37.else if(i.equals("ΘΡΟΥΜΠΙ")){
38....
39.} //29-38 Εικοσιένα εναλλακτικές με διαφορετικό link η κάθε μία

40.@Override
41.public void onBackPressed() { //Στη περίπτωση που πατηθεί από τον
    χρήστη το πίσω κουμπί που υπάρχει εργοστασιακά στη συσκευή τότε
    επιστρέφουμε στην βασική δραστηριότητα μας
42.Intent intent5 = new Intent(NewURL.this,
    Qr_Code_Scanner.class); //Δηλώνουμε μια μεταβλητή intent5 όπου μας
    μεταφέρει από τη κλάση NewURL.java στη Qr_Code_Scanner
43.startActivity(intent5); //Εκκίνηση δραστηριότητας
44.return; //Τέλος λειτουργιών συνάρτησης
45.}
46.public boolean onOptionsItemSelected(MenuItem item) {
47.//handle presses on the action bar items
48.switch (item.getItemId()) {
49.case android.R.id.home:
50.NavUtils.navigateUpFromSameTask(this);
51.return true;
52.}
53.return super.onOptionsItemSelected(item);
54.} //121-129 Επαναφορά στη βασική μας δραστηριότητα πατώντας κουμπί
    που έχουμε δηλώσει (πάνω-αριστερά)
55.}
```

5.5.1 Αρχείο Layout XML webview



Σχήμα 6.3: Xml Webview

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<RelativeLayout`
`xmlns:android="http://schemas.android.com/apk/res/android"`
3. `android:orientation="vertical" android:layout_width="fill_parent"`
4. `android:layout_height="fill_parent" >`
5. `<RelativeLayout android:layout_width="fill_parent"`
6. `android:layout_height="fill_parent"`
7. `android:id="@+id/rlayout2"`
8. `android:layout_alignParentBottom="true"`
9. `android:layout_alignParentStart="true">`
10. `<WebView android:id="@+id/webview1" android:layout_width="fill_parent"`
 - a. `android:layout_height="fill_parent"`
 - b. `android:layout_weight="1.0" /> // Η κλάση WebView είναι μια επέκταση κάποιων παραπάνω κλάσεων διότι μας επιτρέπει να εμφανίσεζουμε ιστοσελίδες ως μέρος της activity layout μας. Δεν περιλαμβάνει οποιεσδήποτε δυνατότητες από ένα πλήρως ανεπτυγμένο πρόγραμμα περιήγησης, όπως στοιχεία ελέγχου`

πλοήγησης ή έναν φραγμό διευθύνσεων. Όλα αυτά που κάνει η WebView, από προεπιλογή, είναι η εμφάνιση μιας ιστοσελίδας.

11. </RelativeLayout>

12. </RelativeLayout>

5.6 Κλάση InfoURL

Στην InfoURL εδραιώσαμε μία πολλή ενδιαφέρουσα λειτουργία η οποία αφορά την ανάγνωση και προβολή ορισμένων αρχείων από το Google Drive. Η διαδικασία διαμοιρασμού ενός αρχείου στο Drive είναι η εξής:

1. Κάνουμε upload το αρχείο στο Google Drive.
2. Επιλέγουμε το αρχείο και πατάμε το εικονίδιο κοινής χρήσης.
3. Στην επιλογή «ιδιωτικό» πατάμε «Αλλαγή» και επιλέγουμε το «Οποιοσδήποτε διαθέτει το σύνδεσμο» ⇒ Αποθήκευση.
4. Ο σύνδεσμος είναι έτοιμος για κοινή χρήση . Η μορφή του είναι:

https://drive.google.com/file/d/0Bzi0N1W_eYVZaXhjM1NFZHRzR0U/edit?usp=sharing

//"**0Bzi0N1W_eYVZaXhjM1NFZHRzR0U**" τυχαία κωδικοποίηση αρχείου από το Google Drive ως παράδειγμα

Έτσι υπάρχει η δυνατότητα να αλλάξουμε το αρχείο όσο θέλουμε και όποτε θέλουμε χωρίς να επηρεάζεται η σωστή λειτουργία της "AMFIKLIA"

1. package com.tzannetatos.myapplication;
2. import android.content.Intent; //Δήλωση βιβλιοθήκης για νέα δραστηριότητα
3. import android.os.Bundle;
4. import android.support.v4.app.NavUtils; //Βιβλιοθήκη για τη χρήση πίσω κουμπιού
5. import android.view.MenuItem;
6. import android.webkit.WebView;

```
7. import android.webkit.WebViewClient;
8. import android.widget.Button;

9. public class InfoURL extends MainActivity {//Επεκτείνουμε την MainActivity
    ως πατρική της InfoURL
10. private class MyWebViewClient extends WebViewClient {
11. @Override
12. public boolean shouldOverrideUrlLoading(WebView view, String url) {
13. view.loadUrl(url);
14. return true;
15. }
16. }
17. WebView webview;
18. @Override
19. protected void onCreate(Bundle savedInstanceState) {
20. // TODO Auto-generated method stub
21. super.onCreate(savedInstanceState);
22. setContentView(R.layout.webview);

23. webview=(WebView)findViewById(R.id.webview1);
24. webview.setWebViewClient(new MyWebViewClient());
25. openURL();
26. }

27. /** Opens the URL in a browser */
28. private void openURL() {
29. webview.loadUrl("https://drive.google.com/file/d/0B_UfFGbZIOPTRGc3cW
    9zMFdYZ0U/view?usp=sharing");//Το συγκεκριμένο link αφορά ένα αρχείο
    που έχουμε αποθηκεύσει στο Google Drive. Αλλάζοντας link προβάλουμε
    ανάλογο αρχείο
30. webview.getSettings().setJavaScriptEnabled(true);
31. }
32. public boolean onOptionsItemSelected(MenuItem item) {//handle presses
    on the action bar items
```

```
33. switch (item.getItemId()) {  
  
34. case android.R.id.home:  
35. NavUtils.navigateUpFromSameTask(this);  
36. return true;  
37. }  
38. return super.onOptionsItemSelected(item);  
39. }  
40. @Override  
41. public void onBackPressed() {  
42. Intent intent5 = new Intent(InfoURL.this, MainActivity.class);  
43. startActivity(intent5);  
44. return;  
45. }  
46. // 9-47 Έχουμε παρόμοια λειτουργία με αυτήν της κλάσης NewURL αλλά το  
    αρχείο που ανοίγουμε είναι μοναδικό
```

5.7 Κλάση Info_But

Η κλάση αυτή είναι ένα ξεχωριστό κουμπί από κεντρικό μενού μας όπου αφού πατηθεί παραπέμπει τον χρήστη σε μία εικόνα με κείμενο που αναφέρουμε λίγα λόγια για την εφαρμογή και τους συντελεστές της. Το σημαντικότερο κομμάτι στη σύνταξη και εμφάνιση αυτής της κλάσης γίνεται στην `button_info.xml` .

```
1. package com.tzannetatos.myapplication;  
2. import android.content.Intent;  
3. import android.os.Bundle;  
4. import android.support.v4.app.NavUtils;  
5. import android.view.MenuItem;  
  
6. public class Info_But extends MainActivity {
```

```
7. public void onCreate(Bundle savedInstanceState) {//Δήλωση έναρξης δραστηριότητας αφού πατηθεί το κουμπί "i" κάτω δεξιά στο βασικό μας μενού
8. super.onCreate(savedInstanceState);//Εκκίνηση δραστηριότητας
9. setContentView(R.layout.button_info);//Προβολή κειμένου xml
10.}

11. public boolean onOptionsItemSelected(MenuItem item) {
    //handle presses on the action bar items
12. switch (item.getItemId()) {

13. case android.R.id.home:
14. NavUtils.navigateUpFromSameTask(this);
15. return true;
16.}
17. return super.onOptionsItemSelected(item);
18.}
19. @Override
20. public void onBackPressed() {
21. Intent intent5 = new Intent(Info_But.this, MainActivity.class);
22. startActivity(intent5);
23. return;
24.}
25.}
```

5.8 Κλάση FlashbackURL

Ένα ακόμα κουμπί στο κεντρικό μενού της βασικής μας δραστηριότητας όπου εμφανίζεται η ίδια λειτουργία με τη συνάρτηση OpenURL() της κλάσης NewURL με μοναδικό link το "<http://www.amfikaia.gr/about-us/>".

```
1. package com.tzannetatos.myapplication;

2. import android.content.Intent;
3. import android.os.Bundle;
4. import android.support.v4.app.NavUtils;
5. import android.view.MenuItem;
6. import android.webkit.WebView;
7. import android.webkit.WebViewClient;
8. import android.widget.Button;

9. public class FlashbackURL extends MainActivity {
10. private class MyWebViewClient extends WebViewClient {
11. @Override
12. public boolean shouldOverrideUrlLoading(WebView view, String url) {
13. view.loadUrl(url);
14. return true;
15. }
16. }
17. WebView webview;
18. @Override
19. protected void onCreate(Bundle savedInstanceState) {
20. // TODO Auto-generated method stub
21. super.onCreate(savedInstanceState);
22. setContentView(R.layout.webview);

23. webview=(WebView)findViewById(R.id.webview1);
24. webview.setWebViewClient(new MyWebViewClient());
25. openURL();
26. }

27. /** Opens the URL in a browser */
28. private void openURL() {
29. webview.loadUrl("http://www.amfikaia.gr/about-us/");
```

```
30.webview.getSettings().setJavaScriptEnabled(true);
31.}
32.public boolean onOptionsItemSelected(MenuItem item) {

33.//handle presses on the action bar items
34.switch (item.getItemId()) {

35.case android.R.id.home:
36.NavUtils.navigateUpFromSameTask(this);
37.return true;
38.}
39.return super.onOptionsItemSelected(item);
40.}
41.@Override
42.public void onBackPressed() {
43.Intent intent5 = new Intent(FlashbackURL.this, MainActivity.class);
44.startActivity(intent5);
45.return;
46.}
47.}
```

5.9 Κλάση BeezURL

Εμφάνιση αρχείου αναφορικά με τις φυλές μελισσών από το Google Drive .
Χρήση του ίδιου πηγαίου κώδικα της InfoURL με διαφορετικό link ως προς άνοιγμα.

```
1. package com.tzannetatos.myapplication;

2. import android.content.Intent;
3. import android.os.Bundle;
4. import android.support.v4.app.NavUtils;
5. import android.view.MenuItem;
```

```
6. import android.webkit.WebView;
7. import android.webkit.WebViewClient;
8. import android.widget.Button;

9. public class BeezURL extends MainActivity {
10. private class MyWebViewClient extends WebViewClient {
11. @Override
12. public boolean shouldOverrideUrlLoading(WebView view, String url) {
13. view.loadUrl(url);
14. return true;
15. }
16. }
17. Button btnBack;
18. WebView webview;
19. @Override
20. protected void onCreate(Bundle savedInstanceState) {
21. // TODO Auto-generated method stub
22. super.onCreate(savedInstanceState);
23. setContentView(R.layout.webview);

24. webview=(WebView)findViewById(R.id.webview1);
25. webview.setWebViewClient(new MyWebViewClient());
26. openURL();
27. }

28. /** Opens the URL in a browser */
29. private void openURL() {
30. webview.loadUrl("https://drive.google.com/file/d/0B_UfFGbZIOPTTzg5Zjlm
    ZzNNamc/view?usp=sharing");//Ορίζουμε link που μας παραπέμπει στο
    Google Drive.Ανοίγει αρχείο με τις φυλές μελισσών
31. webview.getSettings().setJavaScriptEnabled(true);
32. }
33. public boolean onOptionsItemSelected(MenuItem item) {
```

```
34. //handle presses on the action bar items
35. switch (item.getItemId()) {

36. case android.R.id.home:
37. NavUtils.navigateUpFromSameTask(this);
38. return true;
39. }
40. return super.onOptionsItemSelected(item);
41. }
42. @Override
43. public void onBackPressed() {
44. Intent intent5 = new Intent(BeezURL.this, MainActivity.class);
45. startActivity(intent5);
46. return;
47. }
48. }
```


5.10 Αρχείο Layout XML AndroidManifest

1. `<?xml version="1.0" encoding="utf-8"?>`
2. `<manifest xmlns:android="http://schemas.android.com/apk/res/android"`
3. `package="com.tzannetatos.myapplication">`
// Ζητά ένα δικαίωμα που πρέπει να χορηγείται η εφαρμογή προκειμένου να λειτουργήσει σωστά. Τα δικαιώματα εκχωρούνται από τον χρήστη όταν εγκαθίσταται η εφαρμογή. Στις γραμμές 5-15 παρατηρούμε ότι ζητάμε από τον χρήστη κατά την εγκατάσταση να δώσει δικαίωμα στα παρακάτω για να λειτουργήσει η εφαρμογή.
4. `<uses-permission`
`android:name="android.permission.ACCESS_COARSE_LOCATION" />`
5. `<uses-permission android:name="android.permission.CAMERA" />`
6. `<uses-permission android:name="android.permission.INTERNET" />`
7. `<uses-permission`
`android:name="android.permission.ACCESS_NETWORK_STATE" />`
8. `<uses-permission`
`android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`
9. `<uses-permission`
`android:name="android.permission.WRITE_INTERNAL_STORAGE" />`
10. `<uses-permission`
`android:name="android.permission.ACCESS_FINE_LOCATION"/>`
11. `<uses-permission android:name="myapp.permission.MAPS_RECEIVE"/>`
12. `<uses-permission`
`android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>`
13. `<uses-feature android:name="android.hardware.location.gps" />`
14. `<uses-permission`
`android:name="android.permission.CHANGE_CONFIGURATION" />`
15. `<uses-feature android:name="android.hardware.camera" /> // Ο σκοπός της δήλωσης <uses-feature>είναι να ενημερώσει κάθε εξωτερική οντότητα του συνόλου των δυνατοτήτες υλικού και λογισμικού από τα οποία εξαρτάται η εφαρμογή σας. Το στοιχείο προσφέρει ένα απαιτούμενο χαρακτηριστικό`

που σας επιτρέπει να καθορίσετε αν η εφαρμογή σας απαιτεί και δεν μπορεί να λειτουργήσει χωρίς τη δηλωμένη δυνατότητα, ή εάν προτιμά να έχουν τη δυνατότητα, αλλά μπορεί να λειτουργήσει χωρίς αυτήν. Για αυτό δηλώνουμε και την 16,17,15 εντολή διότι η εφαρμογή μας απαιτεί φωτογραφική κάμερα.

16. <uses-feature android:name="android.hardware.camera.autofocus" />

17. <meta-data

18. android:name="com.google.android.gms.version"

19. android:value="@integer/google_play_services_version" />

20. <meta-data

21. android:name="com.google.android.maps.v2.API_KEY"

22. android:value="AlzaSyAnSRUgjk8k7u3TuGorWeBESOMsqM8jyc" />

23. <application

24. android:icon="@drawable/ic_launcher" // Βάζουμε σαν εικόνα της εφαρμογής στο κινητό την εικόνα με όνομα ic_launcher από τον φάκελο drawable

25. android:allowBackup="true"

26. android:label="@string/app_name" // Αντλούμε από το αρχείο string.xml το όνομα της εφαρμογής που είναι ήδη δηλωμένο και θα φαίνεται σαν label κάτω από το εικονίδιο

27. android:supportsRtl="true" // Δηλώνει αν η εφαρμογή μας είναι πρόθυμη να υποστηρίξει δεξιά προς τα αριστερά (RTL) σχεδιαγράμματα.

28. android:theme="@style/AppTheme">

29. <activity android:name=".Splashscreen"> // Δηλώνουμε την κλάση με όνομα Splashscreen

30. <intent-filter>

31. <action android:name="android.intent.action.MAIN" /> // Η εντολή αυτή σημαίνει ότι αυτή η δραστηριότητα είναι το σημείο εισόδου της

- εφαρμογής, δηλαδή, όταν θα ξεκινήσει η εφαρμογή, η δραστηριότητα αυτή δημιουργείται.*
32. `<category android:name="android.intent.category.LAUNCHER" />`
//Με την εντολή αυτή η κλάση Splashscreen θα πρέπει να εμφανίζεται στην εκκίνηση του ανώτατου επιπέδου.
33. `</intent-filter>`
34. `</activity>`
35. `<activity android:name=".MainActivity" >`
36. `<intent-filter>`
37. `<action android:name="android.intent.action.MAIN"/>` *// Η εντολή αυτή σημαίνει ότι αυτή η δραστηριότητα είναι το σημείο εισόδου της εφαρμογής, δηλαδή, όταν θα ξεκινήσει η εφαρμογή, η δραστηριότητα αυτή δημιουργείται.*
38. `<category android:name="android.intent.category.DEFAULT" />` *//*
39. `</intent-filter>`
40. `</activity>`
41. `<activity android:name=".Languages" android:noHistory="true" android:theme="@style/ListFont" android:choiceMode="multipleChoice">`
42. `<intent-filter>`
43. `<action android:name="android.intent.action.MAIN" />`
44. `<category android:name="android.intent.category.DEFAULT" />`
45. `</intent-filter>`
46. `</activity>`
47. `<activity android:name=".Qr_Code_Scanner" android:parentActivityName=".MainActivity" >`
48. `<meta-data`
49. `android:name="android.support.PARENT_ACTIVITY"`
50. `android:value=".MainActivity" />`
51. `</activity>`
52. `<activity android:name=".NewURL"`

```
53. android:parentActivityName=".Qr_Code_Scanner" >

54. <!-- Parent activity meta-data to support 4.0 and lower -->
55. <meta-data
56. android:name="android.support.PARENT_ACTIVITY"
57. android:value=".Qr_Code_Scanner" />
58. </activity>
59. <activity android:name=".Info_But"
60. android:parentActivityName=".MainActivity" >

61. <!-- Parent activity meta-data to support 4.0 and lower -->
62. <meta-data
63. android:name="android.support.PARENT_ACTIVITY"
64. android:value=".MainActivity" />
65. </activity>
66. <activity android:name=".BeezURL"
67. android:parentActivityName=".MainActivity" >

68. <!-- Parent activity meta-data to support 4.0 and lower -->
69. <meta-data
70. android:name="android.support.PARENT_ACTIVITY"
71. android:value=".MainActivity" />
72. </activity>
73. <activity android:name=".FlashbackURL"
74. android:parentActivityName=".MainActivity" >

75. <!-- Parent activity meta-data to support 4.0 and lower -->
76. <meta-data
77. android:name="android.support.PARENT_ACTIVITY"
78. android:value=".MainActivity" />
79. </activity>
80. <activity android:name=".BotanicURL"
81. android:parentActivityName=".MainActivity" >
```

```
82. <!-- Parent activity meta-data to support 4.0 and lower -->
83. <meta-data
84. android:name="android.support.PARENT_ACTIVITY"
85. android:value=".MainActivity" />
86. </activity>
87. <activity
88. android:name=".InfoURL"
89. android:parentActivityName=".MainActivity" >

90. <!-- Parent activity meta-data to support 4.0 and lower -->
91. <meta-data
92. android:name="android.support.PARENT_ACTIVITY"
93. android:value=".MainActivity" />
94. </activity>
95. </application>
96. </manifest>
```

ΚΕΦΑΛΑΙΟ 6

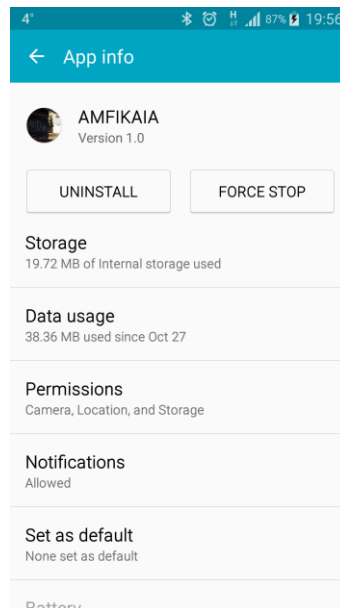
ΕΠΕΞΗΓΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Αφού ολοκληρώσαμε το καθαρά προγραμματιστικό κομμάτι της "ΑΜΦΙΚΑΙΑ" καλούμαστε να εξηγήσουμε τον τρόπο με τον οποίο θα χρησιμοποιήσει κάποιος την εφαρμογή μας αλλά και πως μπορεί ο υπεύθυνος του αγροκτήματος να συμβάλλει στην βελτίωσή της . Τέλος, εμείς ως προγραμματιστές έχουμε περιθώρια για αλλαγές και αναβαθμίσεις ανάλογα με τις ανάγκες των χρηστών και του διαχειριστή. Για αυτό τον λόγο , προκύπτουν οι παρακάτω ρόλοι που χωρίζονται τα καθήκοντα στη χρήση της εφαρμογής μας.

6.1 Ρόλος χρήστη

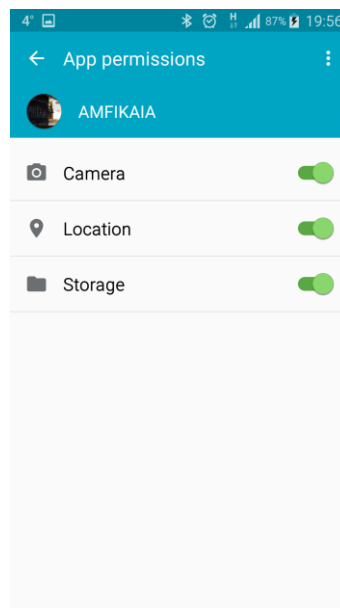
Αρχικά , πρέπει να σημειωθεί ότι αφού εγκατασταθεί σωστά η εφαρμογή μπορεί να χρειαστεί να ενεργοποιηθούν χειροκίνητα οι άδειες("permissions") της κάμερας της τοποθεσίας και του αποθηκευτικού χώρου. Αυτό μπορεί να συμβαίνει λόγω συσκευής ή λόγω έκδοσης λογισμικού της. Μπορεί ο χρήστης να επέμβει σε αυτό ακολουθώντας συγκεκριμένα βήματα:

Ρυθμίσεις(Settings) > Διαχειριστής Εφαρμογών (Application Manager) > "ΑΜΦΙΚΑΙΑ"



Σχήμα 6.4: App Info

> Άδειες(Permissions)



Σχήμα 6.5: App Permissions

Τσεκάρισμα καθενός.

Βασική προϋπόθεση είναι η συσκευή να είναι συνδεδεμένη με Wi-Fi ή σε κινητό δίκτυο με πρόσβαση στο διαδίκτυο.

Τώρα η εφαρμογή μας είναι έτοιμη προς χρήση. Πατώντας το εικονίδιο ξεκινάει το αρχικό "animation" η αλλιώς κινούμενη εικόνα μας και ανοίγει το μενού με τις γλώσσες προς επιλογή.



Σχήμα 6.6: Choose Language

Πατώντας την επιθυμητή γλώσσα του ο χρήστης, η εφαρμογή μας παραπέμπει στο κύριο μενού. Στο παρακάτω μενού έχει πατηθεί η γλώσσα "Greek"



Σχήμα 6.7: Μενού

Σε αυτό το σημείο ο χρήστης έχει τη δυνατότητα να επιλέξει να πατήσει μία από τις έξι βασικές κατηγορίες για το αγρόκτημα . Πατώντας το κουμπί "ΒΟΤΑΝΙΚΟΣ ΚΗΠΟΣ" ανοίγει σελίδα με ένα αρχείο με λίγες πληροφορίες για το αγρόκτημα.



Σχήμα 6.8: Πληροφορίες Αγροκτήματος

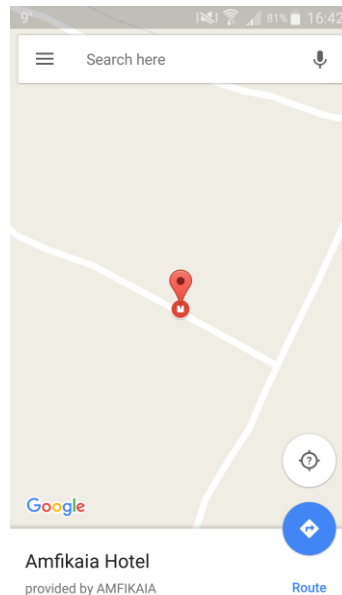
Μπορούμε να σύρουμε την εικόνα πάνω ή κάτω ανάλογα τη προτίμησή μας και να μεγεθύνουμε την εικόνα βάζοντας ταυτόχρονα δύο δάκτυλα και ανοίγοντάς τα μαζί. Πατώντας το βελάκι πάνω-αριστερά ή το ενσωματωμένο πίσω κουμπί της touch συσκευής μας πάμε στο βασικό μενού μας.

Παρομοίως στο "ΦΥΛΛΕΣ ΜΕΛΙΣΣΩΝ" παρουσιάζουμε τις δύο ράτσες μελισσών που υπάρχουν στα μελίσσια του αγροκτήματος με ένα αρχείο.

Στο κουμπί "ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ" έχουμε προσθέσει κομμάτι του επίσημου ιστότοπου της Αμφίκιας με λίγα λόγια για το πώς γεννήθηκε αυτός ο τόπος.

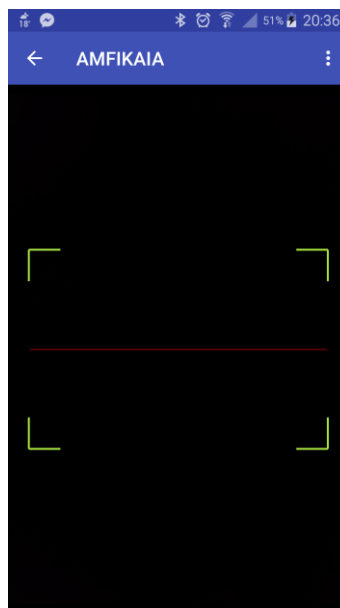
Ενδιαφέρον παρουσιάζει το κουμπί "ΧΑΡΤΗΣ" αφού όταν το πατήσουμε ανοίγει σελίδα σε χάρτη Google Maps που δείχνει την τοποθεσία του αγροκτήματος σε στίγμα ώστε να βοηθάει κάποιον που θέλει να το βρει οδηγώντας ή περπατώντας.

Παρατήρηση : Για να βγάλει διαδρομή με προορισμό την Αμφίκαϊα χρειάζεται η ενεργοποίηση τοποθεσίας GPS. Πατώντας το ενσωματωμένο πίσω κουμπί της συσκευής επιστρέφουμε στο αρχικό μενού.



Σχήμα 6.9 : Gps

Τέλος , η σημαντικότερή μας λειτουργία το "SCANNER" ανοίγει παράθυρο με κάμερα που εντοπίζει QR Codes και αφού σαρώσει κάποιον ανοίγει σελίδα με ανάλογες πληροφορίες για το αντίστοιχο φυτό. Αν βρει είδος barcode μη καταχωρημένο στην εφαρμογή εμφανίζει μία λευκή ετικέτα με την λέξη που αποκωδικοποιήθηκε.



Σχήμα 7.0 : Qr Code

Αν για παράδειγμα σαρώσουμε το QR Code "ΔΑΦΝΗ" τότε ανοίγει η παρακάτω σελίδα με δυνατότητα να σύρει η χρήστης το κείμενο (scrolling).



Σχήμα 7.1: Φυτά

Επιπλέον , έχουμε δημιουργήσει 3 ακόμα κουμπιά βοηθητικά για το μενού της εφαρμογής μας όπου το ένα που βρίσκεται κάτω αριστερά στη βασική μας δραστηριότητα με σύμβολο "i" , παρουσιάζει περιληπτικά την εφαρμογή και τους δημιουργούς της



Σχήμα 7.2: Περίληψη εφαρμογής

Και 2 ακόμα κουμπιά κάτω-δεξιά όπου το ένα ("αμφίκαια") ανοίγει το επίσημο site της Αμφίκαιας και το άλλο την επίσημη σελίδα της στο Facebook ("F")

Σε όλη τη διάρκεια χρήσης της εφαρμογής και σε κάθε σελίδα ο χρήστης μπορεί να πατήσει το κουμπί πάνω-δεξιά (τρεις τελείες) αν θέλει να τερματίσει την εφαρμογή ή να αλλάξει γλώσσα.



Σχήμα 7.3: Υπομενού

Πατώντας "Γλώσσες" πηγαίνουμε πάλι στο μενού γλωσσών.

6.2 Ρόλος Προγραμματιστή

Έχοντας αναλύσει τον τρόπο με τον οποίο η εφαρμογή μας ανταποκρίνεται στις επιλογές που κάνει ο χρήστης και στα κουμπιά που πατάει θέλουμε σε αυτό το σημείο να εντοπίσουμε τι μπορεί ο προγραμματιστής να βελτιώσει πάνω σε αυτή. Αν κάποια στιγμή δημοσιεύσουμε την "ΑΜΦΙΚΑΙΑ" στο Play Store θα θέλαμε να αναβαθμίζεται σε τακτά χρονικά διαστήματα σύμφωνα με τις ανάγκες των χρηστών αλλά και σε συνεργασία με τα αιτήματα του ιδιοκτήτη του αγροκτήματος . Ευελπιστούμε σε μια συνεργασία μακροχρόνια και υγιής ώστε η δουλεία μας να έχει συνέχεια και το καλύτερο δυνατό αποτέλεσμα .

Καταρχήν , τα 21 φυτά που έχουμε εκχωρήσει ως συνθήκη για το τι πληροφορίες θα εμφανιστούν στον χρήστη μπορούν να αυξηθούν σε πολλά περισσότερα . Εφόσον ο βοτανικός κήπος το επιτρέψει σύμφωνα με το κλίμα και την καρποφορία, προσθέτοντας λίγες γραμμές κώδικα στην NewURL.java γίνεται να έχουμε περισσότερα αποτελέσματα φυτών.

Ένας άλλος τρόπος λειτουργίας των QR Codes θα ήταν να εκχωρηθεί το λινκ του ιστότοπου στην κάθε QR Code εικόνα μας ώστε μετά τη σάρωση να ανοίγει η αντίστοιχη σελίδα κατευθείαν μικραίνοντας έτσι την έκταση της NewURL.java. Δεν την προτιμήσαμε αυτή την εκδοχή λόγω του ότι υπάρχουν πολλές εφαρμογές, ήδη δημοσιευμένες, που κάνουν αυτή ακριβώς τη λειτουργία . Προτιμήσαμε κάτι πιο χαρακτηριστικό και απόλυτο.

Επίσης, οι πληροφορίες που δίνονται για το κάθε φυτό μπορούν να παρέχονται όλες μέσω αρχείου αναρτημένο στο Google Drive αντί για λινκ από το διαδίκτυο. Δηλαδή, δίνοντάς μας ορισμένα αρχεία ο ιδιοκτήτης εγγεγραμμένα από αυτόν με πληροφορίες και εικόνες σε μορφή word ή pdf που θα κάλυπταν τις προσωπικές του προσδοκίες. Μία απλή λειτουργία που θα μπορούσε να εφαρμοστεί είναι ένα είδος μόνιμου προβολέα σε κάποιο σημείο της εφαρμογής που να δείχνει νέα του αγροκτήματος και της Αμφίκλιας, ανακοινώσεις προς

τους χρήστες και τους ένοικους των σπιτιών για εκδηλώσεις και πολλά άλλα. Επιπρόσθετα, η αρχιτεκτονική της εμφάνισης της εφαρμογής μας μπορεί να βελτιωθεί προσθέτοντας πιο πολλά γραφικά στα κουμπιά και στις αλλαγές μεταξύ των δραστηριοτήτων όπως κινούμενες εικόνες και ήχους. Τέλος, αλλάζοντας κάποια κομμάτια κώδικα θα μπορούσαμε να κάνουμε πιο γρήγορη το άνοιγμα και την λειτουργία της αλλά την χωρητικότητα που καταλαμβάνει στο εσωτερικό δίσκο του κινητού.

ΚΕΦΑΛΑΙΟ 7

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΡΟΟΠΤΙΚΕΣ

7.1 Σύνοψη της διπλωματικής εργασίας

Κάθε εφαρμογή στο Android πρέπει να δομείται με διαφορετικό τρόπο, ώστε να ανταποκρίνεται στις ανάγκες του χρήστη-διαχειριστή όπως και των κανόνων που πρέπει να τηρηθούν για να δημοσιευθεί στο Play Store. Για αυτό τον λόγο έχουν μεγάλη σημασία το στάδιο της μελέτης, το στάδιο της ανάλυσης των απαιτήσεων και της δημιουργίας του κώδικα που θα εμφανίζει στον χρήστη πληροφορίες για κάθε φυτό αλλά και θα έχει τον ρόλο της επίσημης εφαρμογής του αγροκτήματος 'Αμφίκαια' .

Στην διπλωματική αυτή εργασία, ύστερα από μελέτη και ανάλυση των απαιτήσεων που έγινε, με βάση τις ανάγκες της εφαρμογής και των στόχων που είχαν τεθεί, συντάχθηκε ο κατάλληλος κώδικας για την δημιουργία μιας εφαρμογής φιλικής προς το χρήστη. Κατά την συγγραφή της, παρουσιάστηκε όσο το δυνατό καλύτερα ο σχεδιασμός, η σύνταξη , η υλοποίηση αλλά και η λειτουργία μιας εφαρμογής σάρωσης Qr Codes που βασίζεται στον προγραμματισμό Android Studio.

7.2 Προοπτικές

Όπως ισχύει για όλες τις εφαρμογές, έτσι και για την εφαρμογή μας η ανάπτυξη της δεν σταματά εδώ, αλλά πάντα θα υπάρχουν περιθώρια εξέλιξης λόγω του ότι η τεχνολογία αναπτύσσεται με γοργούς ρυθμούς. Η εφαρμογή που αναπτύχθηκε θα μπορούσε να περιλαμβάνει και άλλες δυνατότητες όπως αυτές που αναφέρονται παρακάτω:

- I. Καλύτερο interface με καλύτερα γραφικά στοιχεία, με ήχους και πιο εμπλουτισμένο menu.

- II. Προσθήκη quiz ή mini games που θα ψυχαγωγούν αλλά ταυτόχρονα θα βοηθάνε το χρήστη να αφομιώσει τις πληροφορίες με ένα πιο ευχάριστο τρόπο.
- III. Ενσωμάτωση περισσότερων κατηγοριών σχετικά με τις παροχές και την πανίδα του αγροκτήματος πχ στάβλος, είδη αλόγων
- IV. Δημιουργία εσωτερικού χάρτη ο οποίος θα παρουσιάζει όλα τα μέρη του αγροκτήματος και θα κατευθύνει τον χρήστη στο επιθυμητό σημείο αναφοράς μέσω εικόνας Drone από μικρό υψόμετρο που καλύπτει όλο το αγρόκτημα.
- V. Δυνατότητα δημιουργίας λογαριασμού του κάθε χρήστη ώστε να πραγματοποιεί κράτηση δωματίου αλλά και προπληρομής με χρήση κάρτας-PayPal.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. <http://stackoverflow.com>
2. <http://youtube.com>
3. <https://developer.android.com>
4. https://el.wikipedia.org/wiki/Android_Studio
5. <https://source.android.com/security/>
6. http://el.wikipedia.org/wiki/%CE%A0%CF%85%CF%81%CE%AE%CE%BD%CE%B1%CF%82_Linux
7. <https://developer.android.com/studio/index.html>
8. <https://source.android.com/devices/camera/index.html>
9. Εισαγωγή στον προγραμματισμό Ν.Ι.ΕΛΛΗΝΑΣ ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ
10. Android για προγραμματιστές, συλλογικό έργο Deitel Paul J., Deitel Harvey M., Deitel Abbey, Morgano Michael, εκδόσεις Γκιούρδας.
11. <https://developer.android.com/studio/command-line/adb.html>
12. <https://developer.android.com/guide/topics/location/index.html>
13. <https://developer.android.com/studio/intro/update.html>

