

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

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Δημιουργία Android εφαρμογής οδηγού πόλης για σπουδαστές για smartphones

**Καγιάννης Πασχάλης
Σκιαδιώτη Ευγενία**

Εισηγητής: Κουκουλέτσος Κωνσταντίνος

**ΑΘΗΝΑ
ΜΑΡΤΙΟΣ 2016**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Δημιουργία Android εφαρμογής οδηγού πόλης για σπουδαστές για smartphones

**Καγιάννης Πασχάλης
Α.Μ. 34428**

**Σκιαδότη Ευγενία
Α.Μ. 35868**

Εισηγητής:

Κουκουλέτσος Κωνσταντίνος

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης

Ευχαριστίες

Αρχικά, θα θέλαμε να ευχαριστήσουμε τον υπεύθυνο καθηγητή της πτυχιακής μας εργασίας κ. Κουκουλέτσο Κώστα, ο οποίος ανέλαβε την επίβλεψη της εργασίας μας και μας παρείχε κάθε δυνατή διευκόλυνση. Ένα μεγάλο ευχαριστώ χρωστάμε επίσης στον συνάδελφο μας Αλέξανδρο Πουρσανίδη, για την καθοριστική συμβολή του στην ολοκλήρωση της εργασίας μας.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως αντικείμενο την ανάπτυξη μιας εφαρμογής για κινητά τηλέφωνα που χρησιμοποιούν το λογισμικό Android της Google. Η επιλογή του θέματος αυτού έγινε λόγω της ραγδαίας ανάπτυξης της αγοράς smartphones (έξυπνων τηλεφώνων), που είχε σαν αποτέλεσμα τη μετεξέλιξη των κινητών σε υπολογιστές με ενσωματωμένο σύστημα κινητής τηλεφωνίας και όχι σε απλά κινητά τηλέφωνα. Η εποχή την οποία διανύουμε είναι εποχή όπου η χρήση υπολογιστών για μια σειρά από λειτουργίες θα αντικατασταθεί από την χρήση smartphones και tablets καθώς οι γρήγοροι ρυθμοί της καθημερινότητας κάνουν τον χρήστη να αναζητάει ολοένα και πιο εύκολες λύσεις. Οι λειτουργίες και οι δυνατότητες που μας παρέχουν καλύπτουν ένα μεγάλο μέρος των καθημερινών μας αναγκών όπως ψυχαγωγία, διασκέδαση, εργασία, άμεση πρόσβαση σε δεδομένα, ικανότητα σύνδεσης στο διαδίκτυο κ.α.

Η εφαρμογή μας έχει σκοπό την παροχή πληροφοριών στους χρήστες - φοιτητές, που αναζητούν διάφορα σημεία ενδιαφέροντος στην Αθήνα (πχ εργασία, φαγητό, διασκέδαση). Για την επίτευξη του στόχου χρησιμοποιήθηκε το λογισμικό Eclipse ένα πρόγραμμα ανοιχτού κώδικα που σε συνεργασία με τα δωρεάν εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google (Android Development Tools) αποτελεί ένα πολύ ισχυρό εργαλείο ανάπτυξης εφαρμογών Android. Επίσης γίνεται περιγραφή και ανάλυση της εγκατάστασης του προγραμματιστικού περιβάλλοντος που χρησιμοποιήθηκε και κάποιων τεχνολογιών που σχετίζονται άμεσα με το android.

Abstract

This project is intended to develop an application for mobile phones using the Google Android software. The choice of this theme was because of the rapid growth of smartphone market, which resulted in the evolution of mobile phones in computers with integrated mobile phone system and not just cell phones. The era in which we live is a time when the use of computers for a number of activities will be replaced by the use of smartphones and tablets as well as the fast pace of everyday life make the user search of increasingly easy solutions. The functions and features that provide us cover a large part of our daily needs such as entertainment, work, immediate access to data, internet connectivity, etc. Our application is intended to provide information to users - students, looking for a point of interest in Athens (eg work, food, entertainment). To achieve the goal, we used the Eclipse software, an open source program along with with the free Android software development tools offered by Google (Android Development Tools), which is a powerful development tool for Android apps. Also there is a description and analysis of the installation of the programming environment used and some technologies directly related to the android.

Περιεχόμενα

Ευχαριστίες	1
Περίληψη	3
Abstract	4
Περιεχόμενα	5
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή	7
1.1 Εισαγωγή	8
1.2 Smartphones	8
1.2.1 Λειτουργικό σύστημα και εφαρμογές	10
1.3 Λειτουργικά Συστήματα	12
1.3.1 Android	13
1.3.2 iOS	13
1.3.3 Windows Phone	16
1.3.4 Άλλα λειτουργικά συστήματα	18
1.4 Open Source	20
1.4.1 Πλεονεκτήματα και μειονεκτήματα Open Source	22
ΚΕΦΑΛΑΙΟ 2: Android - Ιστορικά & Εκδόσεις	24
2.1 Σύντομη ιστορική αναδρομή	25
2.1.2 Γιατί Android;	26
2.2 Εξέλιξη του Android	28
2.2.1 Οι εκδόσεις	28
2.2.2 Αρχιτεκτονική του Android	37
2.2.3 Ασφάλεια στο Android	43
ΚΕΦΑΛΑΙΟ 3: Ανάπτυξη Εφαρμογής – Λέξεις κλειδιά	45
3.1 Κύκλος Ανάπτυξης Εφαρμογής	46
3.1.1 Εγκατάσταση Λογισμικού	46
3.1.2 Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής	46
3.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής	47
3.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό	48
3.2 Βασικές έννοιες - Λέξεις κλειδιά	48
3.2.1 Android SDK	48
3.2.2 Integrated Development Environment	49
3.2.3 Χρήσιμες έννοιες	50
3.3 Java Keywords	52
3.3.1 Προσδιοριστές πρόσβασης	52
3.3.2 Κλάσεις	53
3.3.3 Πακέτα και διεπαφές	53
3.3.4 Δομές επιλογής & κληρονομικότητα	54

3.3.5 Λέξεις – κλειδιά “πολλαπλών ρόλων”	54
3.4 Package Explorer	55
ΚΕΦΑΛΑΙΟ 4: Δομή “Student Guide” – Εγκατάσταση Eclipse	58
4.1 Δομή εφαρμογής	59
4.1.2 Δενδροειδής δομή εφαρμογής	61
4.2 Λειτουργικά συστήματα - Απαιτήσεις συστήματος	61
4.2.1 Εγκατάσταση Java Development Kit	62
4.2.2 Διαδικασία εγκατάστασης Eclipse	63
4.2.3 Εγκατάσταση Android SDK manager	64
4.3 Δημιουργία και διαμόρφωση Android Virtual Device	66
ΚΕΦΑΛΑΙΟ 5: Ανάλυση & υλοποίηση εφαρμογής	68
5.1 Δημιουργία Android application project	69
5.1.1 Εισαγωγή εικόνων στο project	71
5.1.2 Εισαγωγή icon	72
5.1.3 Δημιουργία αρχείου xml	73
5.2 Κεντρική οθόνη - Main.xml	74
5.2.1 Αρχεία Java	78
5.2.2 MainActivity.java	78
5.3 Οθόνη home - Portrait_xml	82
5.3.1 PortraitActivity.java	83
5.3.2 Manifest.xml	84
5.3.3 Τα υπόλοιπα xml αρχεία	84
5.4 Java Αρχεία – κώδικας	87
Βιβλιογραφία	89

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Εισαγωγή



Εικόνα 1

Στο πλαίσιο αυτής της πτυχιακής εργασίας, μελετήθηκαν διάφορες τεχνολογίες οι οποίες θα μπορούσαν να χρησιμοποιηθούν για το σχεδιασμό και την υλοποίηση μιας εφαρμογής για κινητές συσκευές τύπου smartphones. Μετά από σύγκριση που κάναμε με άλλες πλατφόρμες καταλήξαμε στην πλατφόρμα της Google Android.

1.2 Smartphones



Εικόνα 2

Τα smartphones, αποτελούν την φυσική εξέλιξη των κλασικών συσκευών κινητής τηλεφωνίας. Είναι συσκευές που συνδυάζουν ένα κινητό τηλέφωνο με ένα PDA (Personal Digital Assistant) ή έναν υπολογιστή. Τα smartphones είναι αποτέλεσμα της

σύζευξης των κλασικών κινητών τηλεφώνων με τα Personal Digital Assistants (PDAs), τα οποία ήταν στην πράξη ηλεκτρονικές φορητές ατζέντες, που μπορούσαν να επικοινωνήσουν με τον υπολογιστή για ανταλλαγή στοιχείων. Το πρώτο κινητό τηλέφωνο με ενσωματωμένα χαρακτηριστικά PDA παρουσιάστηκε το 1992 στο συνέδριο Comdex. Δύο χρόνια αργότερα τον Αύγουστο του 1994 κυκλοφόρησε στην αγορά από την εταιρεία BellSouth η πρώτη συσκευή τύπου Smartphone με την επωνυμία Simon Personal Communicator. Η συσκευή αυτή με την οθόνη αφής εκτός του ότι μπορούσε να πραγματοποιεί και να δέχεται τηλεφωνικές κλήσεις, ήταν επίσης σε θέση να στέλνει-λαμβάνει φαξ και e-mail καθώς επίσης περιελάμβανε ποικιλία εφαρμογών όπως βιβλίο διευθύνσεων, ημερολόγιο, ατζέντα, παιχνίδια, αριθμομηχανή, ηλεκτρονικό σημειωματάριο, χειρόγραφες σημειώσεις και πληκτρολόγιο αφής. Η επόμενη κυκλοφορία κινητού Smartphone έγινε από την εταιρεία Nokia το 1996 με την ονομασία Nokia9000 και ανήκε στη σειρά Communicator Nokia. Η συσκευή αυτή ήταν τύπου clamshell και ήταν δύο συσκευές σε μία, καθώς λειτουργούσε ξεχωριστά σαν απλό κινητό τηλέφωνο και όταν το άνοιγες σαν PDA κατασκευασμένο από την HP με οθόνη 640 x 200 στο πάνω μέρος και πληκτρολόγιο QWERTY στο κάτω μέρος. Η αποστολή e-mail και η περιήγηση στο διαδίκτυο με τη μορφή απλού κειμένου γινόταν με βάση το λειτουργικό σύστημα GEOS V3.0. Υπάρχουν και άλλα λειτουργικά συστήματα όπως Palm OS, BlackBerry OS και Windows CE/Pocket PC, των οποίων οι πρώτες εκδόσεις έτρεχαν στα PDA, αλλά αργότερα εξελίχθηκαν και άρχισαν να αποτελούν τα λειτουργικά συστήματα των Smartphones. Το 2001 η Palm, Inc. κυκλοφόρησε το Kyocera 6035, την πρώτη συσκευή Smartphone που μπορούσε να λειτουργήσει ταυτόχρονα και ως PDA και ως κινητό τηλέφωνο, είχε περιορισμένη περιήγηση στο διαδίκτυο και λειτουργούσε στο δίκτυο Verizon Wireless. Φτάνοντας στο σήμερα και συγκρίνοντας με το παρελθόν, τα smartphones εξελίχθηκαν με τέτοιο τρόπο, ώστε κάθε μέρα να γίνονται όλο και πιο χρήσιμα και αναγκαία στη ζωή του σύγχρονου ανθρώπου. Κυρίως γιατί η λειτουργικότητα μιας τέτοιας συσκευής προσφέρει σε κάθε χρήστη, επαγγελματία η μη, τη δυνατότητα ανά πάσα στιγμή να έχει ορισμένες από τις ιδιότητες του υπολογιστή του μέσα στο κινητό του. Η επικράτηση των Social Media (Facebook, Twitter, Google) στην καθημερινότητα μας, η ανάγκη για άμεση πληροφόρηση στο διαδίκτυο και η ψυχαγωγία που προσφέρουν,

έχουν κάνει τα Smartphones διαδεδομένα καθώς μια κύρια λειτουργία τους είναι η σύνδεση στο διαδίκτυο. Η ευκολία χειρισμού και η σχετικά χαμηλές τιμές των συσκευών αυτών κάνουν τα Smartphones προσιτά σε όλες τις ηλικίες. Τα κύρια κριτήρια επιλογής ενός Smartphone είναι το μέγεθος και η ανάλυση της οθόνης, η ισχύς του επεξεργαστή, η μνήμη RAM, η διαθέσιμη μνήμη αποθήκευσης, η ύπαρξη και ακρίβεια του GPS και το σημαντικότερο, το λειτουργικό σύστημα από το οποίο αποτελείται. Στην αγορά κυκλοφορούν κινητές συσκευές με λειτουργικά συστήματα όπως BlackBerry, Symbian, Bada και τα πιο διαδεδομένα Android, iOS και Windows Phone. Στην ιστορία των smartphones, η εταιρεία Research in Motion (RIM) κατέχει μια ιδιαίτερη θέση, αφού από το 2002 διαθέτει στην αγορά το BlackBerry, όνομα το οποίο για αρκετά χρόνια αποτελούσε τον ορισμό του «smartphone». Το εργονομικό «qwerty» πληκτρολόγιο και η δυνατότητα ασφαλούς λήψης και αποστολής email κατευθείαν μέσω του δικτύου κινητής τηλεφωνίας, κατέστησε το BlackBerry το κατεξοχήν μοντέλο κινητού το οποίο μπορούσε να ανταποκριθεί στις απαιτήσεις των ατόμων που χρειάζεται να δουλεύουν εν κινήσει.

1.2.1 Λειτουργικό σύστημα και εφαρμογές

Ίσως το κυριότερο χαρακτηριστικό που ξεχωρίζει τα smartphones, πέραν της εμφάνισής τους, είναι το λειτουργικό σύστημα που χρησιμοποιούν. Ίσως να φαίνεται λίγο περίεργο για τους χρήστες οι οποίοι δεν είχαν συνηθίσει να επιλέγουν κάποιο κινητό με κριτήριο το λογισμικό, εντούτοις αποτελεί μια σημαντική παράμετρο, η οποία προδιαγράφει τις πραγματικές δυνατότητες της συσκευής. Το λειτουργικό σύστημα μπορεί να παρέχεται από εταιρεία διαφορετική της κατασκευάστριας της συσκευής. Μέσω αυτού, ο χρήστης μπορεί να εγκαταστήσει στην συσκευή του εφαρμογές οι οποίες θα είναι συμβατές με το λειτουργικό. Αυτό σίγουρα είναι μια βελτίωση από άλλες τεχνολογίες στα κινητά, όπου για παράδειγμα κάποιες εφαρμογές σε Java μπορεί να μην έτρεχαν σε διαφορετικά μοντέλα κινητών. Μέσω του λειτουργικού συστήματος λοιπόν, μπορούν οι εφαρμογές να εκμεταλλευτούν την τεχνολογία που χρησιμοποιεί η συσκευή smartphone. Κάποιες κατασκευάστριες εταιρείες όπως η Apple και RIM, χρησιμοποιούν τα δικά τους «κλειστά» λειτουργικά, με αποτέλεσμα οι εφαρμογές που μπορούν να εγκατασταθούν να έχουν περιορισμούς. Αντίθετα, άλλες εταιρείες που

χρησιμοποιούν λειτουργικά ανοικτού κώδικα όπως είναι το Android, Linux (το Symbian για ένα χρονικό διάστημα), μπορούν να δώσουν περισσότερη δύναμη στον προγραμματιστή εφαρμογών για το smartphone. Ανάμεσα σε άλλες, οι κυριότερες δυνατότητες των smartphones είναι η αποστολή και λήψη email, η πρόσβαση στο διαδίκτυο, η λήψη φωτογραφιών ή και βίντεο υψηλής ευκρίνειας, η οθόνη αφής, η ύπαρξη πλήρους «qwerty» πληκτρολογίου (με πλήκτρα ή δυνητικό στην οθόνη), καθώς και η δυνατότητα ανάγνωσης κειμένων. Εξίσου σημαντικά χαρακτηριστικά είναι η δυνατότητα αναπαραγωγής μουσικής και η εμφάνιση φωτογραφιών και βίντεο στην οθόνη. Επιπρόσθετες δυνατότητες αποτελεί η ύπαρξη συστήματος εντοπισμού θέσης (A-GPS). Πρόσφατα διάφορα smartphones παρέχονται με εσωτερικό επιταχυνσιόμετρο, προσφέροντας την δυνατότητα για πιο εξελιγμένο και φυσικό χειρισμό της συσκευής. Επιλέγοντας κάποιο smartphone, οι παράμετροι που μπορεί να επηρεάσουν την απόφαση είναι το υλικό μέρος, στο οποίο περιλαμβάνεται η αισθητική και ο εργονομικός σχεδιασμός, η ποιότητα της οθόνης, η ευκρίνεια της κάμερας, καθώς και ο χρόνος αυτονομίας. Μια ακόμη σημαντική παράμετρος είναι η ύπαρξη εφαρμογών για το λειτουργικό σύστημα, από παιχνίδια μέχρι εξειδικευμένες υπηρεσίες. Μάλιστα πολλές εταιρείες ξεκίνησαν να δημιουργούν ηλεκτρονικά καταστήματα εφαρμογών, όπως το App Store για τα κινητά που τρέχουν το iPhone Operating System της Apple και το Ovi Store για κινητά που τρέχουν το Symbian OS, το οποίο υποστηρίζεται από την Nokia.

1.3 Λειτουργικά Συστήματα



Εικόνα 3

Όπως είναι γνωστό, σχεδόν όλες οι ηλεκτρονικές συσκευές αποτελούνται από δύο βασικά τμήματα, το hardware και το software. Όσο καλοσχεδιασμένο και αν είναι το hardware, μια συσκευή είναι αδύνατο να λειτουργήσει χωρίς το software, το οποίο αποτελείται κατά ένα μέρος και από το λειτουργικό σύστημα. Έτσι οι συσκευές Smartphones χωρίς κάποιο λειτουργικό σύστημα δε θα μπορούσαν καν να λειτουργήσουν. Το λειτουργικό σύστημα το οποίο είναι προεγκατεστημένο στα Smartphones από τις κατασκευαστικές τους εταιρείες, δε γίνεται να αντικατασταθεί, γι' αυτό και υπάρχουν κατηγορίες ανθρώπων που τα επιλέγουν σύμφωνα με αυτό. Κάποιες από αυτές είναι οι εξειδικευμένοι χρήστες ή οι επαγγελματίες που διαλέγουν με βάση πόσο χρήσιμο θεωρούν ότι τους είναι και οι προγραμματιστές που επιλέγουν με βάση τις δυνατότητες που τους προσφέρει το καθένα. Στην αγορά σήμερα υπάρχει ποικιλία λειτουργικών συστημάτων αλλά τα επικρατέστερα είναι το iOS, το Windows Phone και το Android. Στη συνέχεια, θα αναφερθούμε ξεχωριστά στην εξέλιξη του καθενός καθώς επίσης και στα αρνητικά και θετικά χαρακτηριστικά τους.

1.3.1 Android



Εικόνα 4

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιρειών, εταιρειών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Μέσω της δικιά του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), επιτρέπει στους κατασκευαστές να δημιουργούν πρωτοποριακές εφαρμογές. Είναι ελεύθερο (open source), έχει ισότιμες εφαρμογές, επεκτείνει τις δυνατότητες των εφαρμογών και παρέχει γρήγορη, εύκολη και δωρεάν ανάπτυξη. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA.

1.3.2 iOS



Εικόνα 5

Το 2007 η AppleInc. κυκλοφόρησε και παρουσίασε για πρώτη φορά το iPhone του οποίου οι πωλήσεις μέσα σε ένα χρόνο ξεπέρασαν το ένα εκατομμύριο. Η εταιρεία ανέφερε απλά ότι η συσκευή έτρεχε με λειτουργικό σύστημα OS X και οι εφαρμογές τρίτων ακόμα δεν υποστηρίζονταν. Την ίδια χρονιά ανακοινώθηκε από την εταιρεία ότι το πρώτο Software Development Kit (SDK) βρισκόταν υπό ανάπτυξη και μισό σχεδόν χρόνο μετά κυκλοφόρησε η πρώτη beta έκδοση. Εκείνη τη χρονιά παρουσιάστηκε και το νέο όνομα του λειτουργικού συστήματος που ήταν το iPhone OS. Τον Ιούνιο του 2010, το λειτουργικό σύστημα πήρε τη σημερινή του ονομασία που είναι το iOS και η Apple κυκλοφόρησε το iPad και το iPod touch, νέες συσκευές που είχαν το λειτουργικό αυτό. Το iOS είναι βασισμένο στο Mac OS, επομένως είναι ένα Unix λειτουργικό σύστημα, όπου η ανάπτυξη εφαρμογών γίνεται σε Objective-C, C και C++ με χρήση του Cocoa Touch API. Με κάθε μικρή ή μεγάλη ενημέρωση της έκδοσης του λειτουργικού, η Apple ανακοινώνει κάθε φορά και ένα καινούριο SDK. Μερικές iOS SDK beta εκδόσεις κυκλοφορούν πριν από την επίσημη ενημέρωση της έκδοσης του λειτουργικού, έτσι ώστε οι προγραμματιστές να μπορούν να δοκιμάζουν τη συμβατότητα των εφαρμογών τους και να προσθέτουν τα καινούρια χαρακτηριστικά που υπάρχουν. Αν και το iOS SDK μπορεί κάποιος να το κατεβάσει δωρεάν, για να το χρησιμοποιήσει θα πρέπει πρώτα να εγγραφεί στο iPhone Developer Program, ένα βήμα που εκτός από την έγκριση της Apple απαιτεί και μια ετήσια συνδρομή του ποσού των 99\$. Ένα από τα μειονεκτήματα για την ανάπτυξη εφαρμογών σε iOS είναι ότι το SDK μπορεί κάποιος να το εγκαταστήσει μόνο σε λειτουργικό Mac OS X και η τελευταία έκδοση απαιτεί το OS X 10.7.4 "Lion" ή και τα πιο πρόσφατα, "Mountain Lion" ή "Mavericks" .

Η Apple κάθε χρόνο σχεδόν κυκλοφορεί και μια νέα έκδοση του λειτουργικού της. Έτσι το 2008 κυκλοφόρησε η δεύτερη έκδοση του iPhone OS, οι αναβαθμίσεις της οποίας έφτασαν μέχρι και το 2.2.1. Ένα χρόνο αργότερα, τον Ιούνιο του 2009 έκανε την εμφάνιση της η έκδοση iPhone OS 3.0 και ενημερώθηκε μέχρι και την έκδοση 3.1.3. Το iPad πρωτοκυκλοφόρησε με την έκδοση iPhone OS 3.2, μια έκδοση που δημιουργήθηκε αποκλειστικά για αυτά. Η επόμενη έκδοση κυκλοφόρησε τον Ιούνιο του 2010 με την νέα ονομασία iOS 4.0. Η έκδοση αυτή είχε πάνω από 1500 νέα API's για τους προγραμματιστές με σημαντικότερο το πολυαναμενόμενο multitasking

χαρακτηριστικό. Παρ' όλα αυτά πολλές λειτουργίες δεν ήταν ακόμα διαθέσιμες και η έκδοση διατέθηκε μόνο για το iPod Touch, το iPhone και το iPhone CDMA. Η αναβάθμιση της έκδοσης σε iOS 4.2 έφερε το iPhone με το iPad σε μεγάλη ισότητα χαρακτηριστικών. Η τελευταία αναβάθμιση για το iPod Touch και το iPhone ήταν η iOS 4.2.1, ενώ για τα άλλα προϊόντα κυκλοφόρησε η 4.3. Τον Ιούνιο του 2011, η Apple παρουσίασε το iOS 5.0 και το iOS 5 beta SDK. Η έκδοση αυτή του iOS έχει πάνω από 200 νέα χαρακτηριστικά όπου τα σημαντικότερα είναι το iMessage που επιτρέπει την συνομιλία σε όλες τις συσκευές που τρέχουν αυτή την έκδοση, η πλήρη ενσωμάτωση του iCloud, οι βελτιώσεις του AirPlay κ.α.. Το iOS 5.0 υποστηρίζει όλα τα μοντέλα του iPad, το iPhone 3GS, το iPhone 4 GSM και CDMA, καθώς και το iPod Touch (3ης και 4ης γενιάς).

Πέρα από όλες τις εκδόσεις που αναφέρθηκαν παραπάνω και κυκλοφόρησαν στην αγορά, υπήρξαν και μερικές ακυκλοφόρητες. Μέχρι τον Οκτώβριο του 2010, δύο εκδόσεις του iOS ποτέ δε δημοσιεύτηκαν. Μετά την πρώτη beta της έκδοσης iOS 1.2, η δεύτερη beta ονομάστηκε 2.0b2 αντί για 1.2b2. Η άλλη έκδοση που δε δημοσιεύτηκε ήταν η iOS 4.2 η οποία αντικαταστάθηκε με την 4.2.1 εξαιτίας ενός προβλήματος που παρουσίασε η 4.2 beta 3 στο δίκτυο Wi-Fi.

Το καλό είναι ότι η Apple κυκλοφορεί ταυτόχρονα τις ίδιες εκδόσεις iOS για τα μοντέλα των συσκευών iPod Touch και iPhone που έχουν κυκλοφορήσει την ίδια χρονική περίοδο, με τη μόνη διαφορά ότι οι χρήστες του iPhone λαμβάνουν όλες τις ενημερώσεις λογισμικού δωρεάν ενώ οι χρήστες του iPod Touch έπρεπε να πληρώσουν για τις σημαντικές αναβαθμίσεις 2.0 και 3.0. Μετά την έκδοση του iOS 4.0 όμως, οι ενημερώσεις για τους χρήστες του iPod Touch παρέχονταν πλέον δωρεάν. Το αρνητικό της όλης υπόθεσης, είναι ότι οι συσκευές της Apple μπορούν να στέλνουν αρχεία μόνο αναμεταξύ τους και για να στείλουν σε κάποια άλλη συσκευή θα χρειαστεί η εγκατάσταση κάποιας ειδικής εφαρμογής. Επίσης, σε αντίθεση με τα άλλα λειτουργικά συστήματα οι χρήστες του iOS δε μπορούν να παραμετροποιήσουν τις συσκευές τους και έχουν την άδεια να εγκαταστήσουν εφαρμογές μόνο από το App Store.

1.3.3 Windows Phone



Εικόνα 6

Σαν μεγάλη εταιρεία η Microsoft, η οποία έχει διαδοθεί μέσα από το software που παράγει και αποτελεί το μόνο λειτουργικό σύστημα σε μεγάλη πληθώρα Η/Υ, θα ήταν αδύνατο να μη δημιουργήσει ένα λειτουργικό σύστημα που θα αναφερόταν στα “έξυπνα τηλέφωνα”. Αρχικά, η ονομασία του λειτουργικού συστήματος ήταν το Windows Mobile και στόχευε περισσότερο σε πωλήσεις στον επιχειρησιακό χώρο. Αργότερα, το 2004, έγινε η πρώτη προσπάθεια ενημέρωσης του Windows Mobile με την ονομασία Photon αλλά το σχέδιο κινήθηκε αργά και εν τέλει ακυρώθηκε. Τέσσερα χρόνια μετά, η Microsoft αναδιοργάνωσε την ομάδα του Windows Mobile και ξεκίνησε τις εργασίες για ένα νέο λειτουργικό σύστημα που θα απευθυνόταν στα κινητά και θα έδινε περισσότερη βάση στους καταναλωτές. Το προϊόν ήταν να κυκλοφορήσει το 2009 με την ονομασία Windows Phone αλλά αρκετές καθυστερήσεις ανάγκασαν τη Microsoft να αναπτύξει το Windows Mobile 6.5 σαν μια προσωρινή κυκλοφορία. Τελικά, το Φεβρουάριο του 2010, η Microsoft παρουσίασε στο Mobile World Congress, το νέο λειτουργικό σύστημα με την ονομασία Windows Phone 7 series. Επειδή όμως το όνομα της φάνηκε πολύπλοκο και “φλύαρο”, τον Απρίλη του 2010 το έκοψε επίσημα σε Windows Phone 7. Μεγάλες εταιρείες κατασκευής κινητής τηλεφωνίας όπως HTC, Dell, Samsung και LG άρχισαν να χρησιμοποιούν το νέο αυτό λειτουργικό σύστημα στα καινούρια μοντέλα που κατασκεύαζαν με αποτέλεσμα την εκτόξευση των πωλήσεων τους. Σχεδόν ένα χρόνο αργότερα, κυκλοφόρησε το Windows phone 7.5 με την κωδική ονομασία Mango. Πριν την επίσημη ενημέρωση του λειτουργικού

συστήματος στη Νέα Υόρκη στις 24 Μαΐου του 2011, ο Steve Ballmer αποκάλυψε ότι θα υπάρχουν πάνω από 500 νέα χαρακτηριστικά. Παρ' όλο που η ενημέρωση αυτή στην ουσία ήταν η 7.1, διατέθηκε στην αγορά ως 7.5. Με τη κυκλοφορία της έκδοσης αυτής, νέες κατασκευαστικές εταιρείες όπως: Acer, Fujitsu και ZTE έγιναν συνεργάτες της Microsoft. Το 2012, μια μικρή ενημέρωση κυκλοφόρησε με την ονομασία Tango η οποία πέρα από κάποιες διορθώσεις σε κάποια προβλήματα που είχαν δημιουργηθεί, μείωσε και τις απαιτήσεις υλικού για να επιτρέψει σε συσκευές κινητής τηλεφωνίας με επεξεργαστή 800MHz και 256MB RAM, να τρέχουν Windows Phone. Κάποια από τα σημαντικότερα χαρακτηριστικά της ενημέρωσης αυτής ήταν τα καλύτερα μηνύματα πολυμέσων με την δυνατότητα αποστολής ringtones, εξαγωγή και διαχείριση των επαφών της κάρτας SIM, πιο αξιόπιστες ειδοποιήσεις, εγκατάσταση και συμβατότητα της εφαρμογής Skype και λήψη συνημμένου αρχείου μέσα από το server Microsoft Exchange 2003. Επίσης, η ενημέρωση αυτή επιβεβαιώθηκε σε αγορές 23 επιπλέον χωρών και τα πρώτα κινητά που την έτρεξαν ήταν Nokia Lumia 610 και το ZTE Orbit. Τα Windows Phone 8 με κωδική ονομασία Apollo ήταν η επόμενη μεγάλη ενημέρωση και δικαιολογημένα σηματοδότησε την επόμενη γενιά των Windows Phone, καθώς αποτελείται από πολλά νέα χαρακτηριστικά βασισμένα σε καινούριες τεχνολογίες. Κάποια από τα σημαντικότερα είναι:

- 1) Η δυνατότητα μετάβασης σε βασικά στοιχεία από τα Windows 8 όπως: τον πυρήνα, το σύστημα αρχείων, τους οδηγούς (drivers), τα εξαρτήματα ασφαλείας και την υποστήριξη γραφικών.
- 2) Η εγκατεστημένη εφαρμογή του Internet Explorer 10.
- 3) Το ιστορικό multitasking.
- 4) Η υποστήριξη πολυπύρηνων επεξεργαστών έως και τεσσάρων πυρήνων.
- 5) Η υποστήριξη για κάρτες MicroSD.
- 6) Η υποστήριξη για ανάλυση WXGA (1280x720, 1280x768).
- 7) Η υποστήριξη NFC που συμπεριλαμβάνει την ανταλλαγή πληρωμών και τη κοινή χρήση του περιεχομένου (Η NFC υποστηριζόταν μερικώς και στην έκδοση Tango).
- 8) Τη συμβατότητα των εφαρμογών Windows 8 με Windows Phone 8.

- 9) Τους χάρτες της Nokia (Navteq χάρτες που μπορούν να εμφανιστούν ακόμα και όταν η συσκευή βρίσκεται εκτός λειτουργίας).
- 10) Την απομακρυσμένη διαχείριση συσκευών των Windows Phone παρόμοια με την διαχείριση των Windows PCs.
- 11) Voip και βίντεο συζητήσεις για κάθε εφαρμογή που χειριζόταν τέτοιες λειτουργίες.
- 12) Ελάχιστη 18μηνη υποστήριξη των Windows Phone ενημερώσεων που υποστήριζαν Windows Phone 8.
- 13) Επιλογή στη φωτογραφική μηχανή να βγάζει έναν αριθμό επαναλαμβανόμενων φωτογραφιών για να μπορεί ο χρήστης να επιλέξει την καλύτερη.
- 14) Τη δυνατότητα λήψης πανοραμικής φωτογραφίας.
- 15) Τη δυνατότητα λήψης στιγμιότυπων οθόνης.

Τα Windows Phone 8.1 περιλαμβάνουν ορισμένες αλλαγές σε σχέση με τον προκάτοχό τους, που αφορούν στο πλήθος των live tiles που μπορεί να δεχτεί, ορισμένες λειτουργίες στις ρυθμίσεις, η προσθήκη notification center που ήταν σημαντική έλλειψη στις προηγούμενες εκδόσεις. Επίσης, έχουν γίνει αλλαγές στο γραφικό περιβάλλον των κλήσεων, της κάμερας, των άλμπουμ φωτογραφιών και βίντεο, της μουσικής και στην οθόνη κλειδώματος.

Τέλος, όσον αφορά τα Windows Phone, η Microsoft προσφέρει ένα νέο γραφικό περιβάλλον με μια γλώσσα σχεδιασμού που ονόμασε "Metro". Θετικό στοιχείο του λογισμικού αυτού, είναι ότι επιτρέπει την ενσωμάτωση εφαρμογών από τρίτους πέρα της Microsoft και ζητάει τις ελάχιστες απαιτήσεις από την συσκευή στην οποία τρέχει. Δυστυχώς, τα Windows Phone δεν είναι τόσο διαδεδομένα και αυτό οφείλεται στο μικρό μερίδιο των κινητών στην αγορά που τρέχουν το λειτουργικό σύστημα και ο μικρός αριθμός των εφαρμογών που υπάρχουν στο Windows Phone Store.

1.3.4 Άλλα λειτουργικά συστήματα

Σήμερα στην αγορά πέρα από τα τρία βασικά λειτουργικά συστήματα που αναφέραμε πιο πάνω υπάρχουν και άλλα τα οποία όμως είναι λιγότερο διαδεδομένα

και με λιγότερη απήχηση στον κόσμο. Κυριότεροι λόγοι είναι ότι αυτά τα λειτουργικά συστήματα τρέχουν μόνο σε σειρές Smartphone συγκεκριμένων εταιρειών και έχουν λίγες εφαρμογές διαθέσιμες στα καταστήματα εφαρμογών τους (App stores). Τέτοια λειτουργικά συστήματα είναι:

1) Το *Symbian*, το οποίο αποτελούσε τη δημοφιλέστερη πλατφόρμα smartphone στην παγκόσμια αγορά μέχρι και το τέλος του 2010. Αρχικά, αναπτύχθηκε από το Symbian Ltd σε γλώσσα προγραμματισμού C++, υποστήριζε τρεις διαφορετικές διεπαφές χρήστη και λειτουργούσε αποκλειστικά με ARM επεξεργαστές. Πολλές γνωστές εταιρίες όπως: Fujitsu, Nokia, Samsung, Sharp και Sony Ericsson προτίμησαν αυτό το λειτουργικό για τις συσκευές τους. Το 2009, το Symbian αποτέλεσε πλατφόρμα ανοιχτού λογισμικού η οποία αναπτύχθηκε από το Symbian Foundation (διάδοχος του αρχικού λειτουργικού συστήματος). Η τελευταία έκδοση, κυκλοφόρησε τέλη του 2010 με το Nokia N8 να τη χρησιμοποιεί πρώτο και το τελευταίο κινητό με λειτουργικό Symbian κυκλοφόρησε τον Ιούνιο του 2012 και ήταν το Nokia 808 PureView. Σήμερα, το λειτουργικό συντηρείται από την Accenture και σύμφωνα με μια συμφωνία που είχε με τη Nokia, θα της παρέχει υπηρεσίες υποστήριξης, ανάπτυξης και ενημέρωσης λογισμικού μέχρι και το 2016.

2) Το *BlackBerry*, το οποίο είναι ένα ιδιόκτητο λειτουργικό σύστημα έξυπνων κινητών τηλεφώνων και έχει αναπτυχθεί από τη Research In Motion σε γλώσσα προγραμματισμού C++. Η πλατφόρμα του είναι ίσως η πιο γνωστή όσον αναφορά το κομμάτι της υποστήριξης του εταιρικού ηλεκτρονικού ταχυδρομείου μέσω του MIDP 1.0 και τώρα πλέον και του MIDP 2.0, το οποίο επιτρέπει την πλήρη ενεργοποίηση και τον ασύρματο συγχρονισμό με ημερολόγια, επαφές, σημειώσεις κ.α.. Εξαιτίας της εύκολης λειτουργίας του αρχικά είχε σχεδιαστεί αποκλειστικά για επιχειρήσεις, σήμερα όμως είναι διαθέσιμη και για τους απλούς χρήστες. Οι προγραμματιστές μπορούν εύκολα να αναπτύξουν εφαρμογές σε γλώσσα προγραμματισμού Java αλλά για την δημοσίευση εφαρμογών που τρέχουν ορισμένες λειτουργίες απαιτείται πρώτα η ψηφιακή υπογραφή. Σήμερα, το κατάστημα εφαρμογών της BlackBerry (Blackberry App World) έχει πάνω από 50.000 ενεργές εφαρμογές.

3) Το *Bada*, το οποίο αναπτύχθηκε από τη Samsung Electronics. Το όνομα της προέρχεται από μια κορεάτικη λέξη που σημαίνει ωκεανός ή θάλασσα. Το πρώτο

κινητό που έτρεξε το λειτουργικό αυτό ήταν το Samsung Wave S8500, το οποίο είχε οθόνη αφής και κυκλοφόρησε τον Ιούνιο του 2010. Μαζί με το κινητό κυκλοφόρησε και το αποκλειστικό κατάστημα εφαρμογών της Samsung, το Samsung Apps, που σήμερα διαθέτει γύρω στις 3000 εφαρμογές. Η Samsung ισχυρίζεται ότι δε θεωρεί το Bada σαν λειτουργικό σύστημα έξυπνων τηλεφώνων αλλά ως ένα λειτουργικό σύστημα με μια αρχιτεκτονική ρύθμιση του πυρήνα. Παρ' όλα αυτά το χρησιμοποιεί για τα Smartphones που κατασκευάζει ταυτόχρονα με τα λειτουργικά συστήματα του Android και των Windows Phone.

4) Το Palm OS (επίσης γνωστό ως Garnet OS OS) είναι ένα κινητό λειτουργικό σύστημα το οποίο αναπτύχθηκε αρχικά από την Palm, Inc, για προσωπικούς ψηφιακούς βοηθούς (PDAs) το 1996. Το Palm OS έχει σχεδιαστεί για ευκολία στην χρήση με οθόνη αφής που βασίζεται στη γραφική διεπαφή χρήστη. Είναι εφοδιασμένο με μια σειρά από βασικές εφαρμογές για την διαχείριση προσωπικών πληροφοριών.

1.4 Open Source



Εικόνα 7

Το open source είναι λογισμικό το οποίο διατίθεται ελεύθερα και κάποιος μπορεί να το αντιγράψει, να το μελετήσει & να το τροποποιήσει ελεύθερα. Τα παραπάνω ισχύουν τόσο για τον πηγαίο κώδικα όσο και για τα εκτελέσιμα αρχεία του

προγράμματος. Οι υπολογιστές, αποτελούνται από το hardware και το software. Το software γράφεται με τη βοήθεια των γλωσσών προγραμματισμού, οι οποίες είναι ενδιάμεσες γλώσσες που τις “καταλαβαίνουν” τόσο οι άνθρωποι όσο και τα computers, όπως η C, Fortran κλπ. Ο κώδικας αυτός λέγεται πηγαίος κώδικας (source code). Όπως οι άνθρωποι έχουν την κατ’ εξοχήν γλώσσα τους που την προτιμάνε, έτσι και τα computers έχουν τη δική τους, τη γλώσσα μηχανής, το δυαδικό σύστημα δηλαδή “0-1”. Οι υπολογιστές εμπεριέχουν ένα ειδικό πρόγραμμα, τον μεταγλωττιστή που μεταφράζει τη γλώσσα προγραμματισμού σε γλώσσα μηχανής. Το compilation από γλώσσα προγραμματισμού σε γλώσσα μηχανής, έχει ένα κακό όμως. Επειδή υπεραναλύει τη διαδικασία σε πολύ μικρά κομματάκια, “κρύβει” τις λεπτομέρειες υλοποίησης του προγράμματος από τους άλλους. Έτσι, από τη στιγμή που γίνεται compile ένα πρόγραμμα, δεν υπάρχει τρόπος να καταλάβει κανείς το πώς ο προγραμματιστής το έφτιαξε, ειδικά όταν μιλάμε για περίπλοκα προγράμματα. Έτσι οι εταιρείες και οι προγραμματιστές έδιναν στους πελάτες τους μόνο τα εκτελέσιμα αρχεία (την γλώσσα μηχανής), ούτως ώστε οι ανταγωνιστές τους να βλέπουν μόνο το αποτέλεσμα και όχι την υλοποίησή τους.

Ο Richard Stallman, ο οποίος κατάλαβε το πρόβλημα των closed-source λογισμικών (ή προγραμμάτων) έθεσε τις βάσεις για την ανάπτυξη ενός νέου είδους προγραμμάτων, των “ελεύθερων προγραμμάτων”. Με την απόκτηση τέτοιων προγραμμάτων ο χρήστης έχει τη δυνατότητα να δει και να επεξεργαστεί τον κώδικά τους. Το open source και το free software υποστηρίζουν αυτό τον τρόπο ανάπτυξης προγραμμάτων. Σύμφωνα με το Ίδρυμα Ελεύθερου Λογισμικού, οι άδειες χρήσης πρέπει να περιλαμβάνουν τις εξής ελευθερίες:

- 1) Ελευθερία χρήσης του προγράμματος για οποιονδήποτε σκοπό.
- 2) Ελευθερία μελέτης και τροποποίησης του προγράμματος.
- 3) Ελευθερία αντιγραφής του προγράμματος.
- 4) Ελευθερία βελτίωσης του προγράμματος και επανέκδοσης του, προς το συμφέρον της κοινότητας των χρηστών.

Οι ελευθερίες 1 και 3 προϋποθέτουν την πρόσβαση των χρηστών στον πηγαίο κώδικα του λογισμικού.

1.4.1 Πλεονεκτήματα και μειονεκτήματα Open Source

Ας αρχίσουμε με τα **πλεονεκτήματα** που στην ουσία μας “φανερώνουν” την χρησιμότητά του open source στο Android:

- 1) Η λογική της ανάπτυξης του λογισμικού του ανοιχτού κώδικα είναι τέτοια η οποία δίνει τη δυνατότητα για περαιτέρω έρευνα και επεξεργασία των προγραμμάτων.
- 2) Είναι δυνατή η παραμετροποίηση των προγραμμάτων αυτών, ώστε να προσαρμοστούν σε κάθε ανάγκη της βιβλιοθήκης ή οποιουδήποτε οργανισμού.
- 3) Μπορεί να αποτελέσει σημαντικό εκπαιδευτικό εργαλείο ή εργαλείο για απόκτηση προγραμματιστικής εμπειρίας απ’ αυτούς που αναπτύσσουν κώδικα.
- 4) Το κόστος των προγραμμάτων ανοιχτού κώδικα είναι τις περισσότερες φορές μηδενικό.
- 5) Τέλος, η χρήση ανοιχτού κώδικα δεν περιορίζει τον οργανισμό σε μια σχέση εξάρτησης από εταιρείες

Ας αναφέρουμε όμως και ορισμένα **μειονεκτήματα**:

- 1) Η υποστήριξη των προγραμμάτων ανοιχτού κώδικα πολλές φορές είναι δύσκολη, καθώς δεν υπάρχει κάποια επίσημη εταιρεία με καταρτισμένους τεχνικούς.
- 2) Παρατηρείται πολλές φορές ελλιπής τεκμηρίωση στα προγράμματα ανοιχτού κώδικα.
- 3) Τα αντίστοιχα εμπορικά προγράμματα προσφέρουν κατά κύριο λόγο καλύτερη υποστήριξη, τεκμηρίωση και συνεργασία, ενώ οι δυνατότητές τους συχνά είναι υπέρτερες.
- 4) Αν ένα πρόγραμμα “εγκαταλειφθεί” απ’ τους δημιουργούς του, ο οργανισμός μένει δίχως υποστήριξη και μελλοντικές προοπτικές.
- 5) Τέλος μια βιβλιοθήκη, με την εγκατάσταση ενός λογισμικού ανοιχτού κώδικα χρειάζεται να έχει εξαιρετικά καταρτισμένο τμήμα πληροφορικής ώστε να

Δημιουργία Android εφαρμογής οδηγού πόλης για σπουδαστές για smartphones

αντιμετωπίζει τις όποιες δυσκολίες που παρουσιάζονται, με αποτέλεσμα την επιβάρυνση του προϋπολογισμού.

ΚΕΦΑΛΑΙΟ 2: Android - Ιστορικά & Εκδόσεις

2.1 Σύντομη ιστορική αναδρομή

Το Android δημιουργήθηκε το 2003 στην Καλιφόρνια από τους Rubin, Miner, Sears και White. Αρχικός τους σκοπός ήταν να δημιουργήσουν ένα λειτουργικό για “έξυπνα” κινητά τα οποία θα γνωρίζουν που βρίσκεται ο ιδιοκτήτης τους ανά πάσα στιγμή. Αρχικά η ομάδα του Android λειτουργούσε μυστικά, αλλά η μεγάλη ανάπτυξη στο λειτουργικό ξεκίνησε μετά την εξαγορά του από την Google το 2005. Από τη χρονιά εκείνη μέχρι και το 2007 η Google δούλευε σιωπηλά πάνω στο Android κατοχυρώνοντας πατέντες και ψάχνοντας συνεργάτες.

Το 2007 δημιουργήθηκε η Open Handset Alliance, ένας συνεταιρισμός που αποτελούνταν από τη Google, την HTC, την Samsung, την Qualcomm (κατασκευαστές επεξεργαστών) και άλλους, με σκοπό να κάνουν το Android ένα ανοικτό λογισμικό (open source) και να κυκλοφορήσουν τα πρώτα smartphones με

Android. Πράγματι, το πρώτο smartphone με λειτουργικό Android κυκλοφόρησε ένα χρόνο μετά από την HTC και ονομαζόταν HTC Dream. Από εκεί και μετά η ιστορία είναι λίγο πολύ γνωστή. Με πρώτη τη Samsung, δεκάδες εταιρείες υιοθέτησαν το Android ως λειτουργικό και κατάφεραν να το κάνουν το κυρίαρχο λειτουργικό αυτή τη στιγμή στην αγορά. Η αναφορά της Samsung δεν είναι τυχαία, αφού είναι η πρώτη εταιρεία στις πωλήσεις κινητών παγκοσμίως και η εξάπλωση του Android οφείλεται κατά μεγάλο βαθμό σε αυτήν.

2.1.2 Γιατί Android;



Εικόνα 8

Παραπάνω αναλύσαμε τα σημαντικότερα λειτουργικά συστήματα. Εμείς επιλέξαμε το Android γιατί σαν πλατφόρμα είναι η πιο γρήγορα αναπτυσσόμενη σήμερα και χρησιμοποιείται από πολλούς κατασκευαστές κινητών τηλεφώνων, ενώ στην αγορά υπάρχουν πολλές διαθέσιμες συσκευές σε όλες τις κατηγορίες τιμών. Πιο αναλυτικά, το Android έχει τα εξής πλεονεκτήματα:

- 1) Υπάρχει μια Android συσκευή για όλα τα γούστα και κατηγορίες τιμών από διάφορους κατασκευαστές όπως Samsung, HTC, Motorola, Sony Ericsson, LG και Vodafone.
- 2) Μπορείς να επιλέξεις πάνω από 1.600.000 εφαρμογές (δωρεάν ή με πληρωμή). Ο αριθμός αυτών των εφαρμογών καθημερινά αυξάνεται.
- 3) Γύρω στο 60% των android εφαρμογών στο android market είναι δωρεάν.
- 4) Είναι εύκολο στη χρήση του και στην προσωποποίηση ώστε να τα ρυθμίσεις όπως σε βολεύει.
- 5) Συγχρονίζεται εύκολα με υπηρεσίες τις Google, όπως το Gmail, Contacts & Google Maps και άλλα.

- 6) Είναι μια πραγματικά ανοιχτή ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux.
- 7) Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
- 8) Πάμπολλες ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες.
- 9) 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.
- 10) Υψηλής ποιότητας γραφικά και ήχος.
- 11) Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik*. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίξουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης.

Το Android έχει μια μεγάλη κοινότητα προγραμματιστών που γράφουν εφαρμογές, οι οποίες επεκτείνουν τη λειτουργικότητα των συσκευών. Οι εφαρμογές γράφονται σε μια προσαρμοσμένη έκδοση της JAVA και μπορεί κάποιος να κατεβάσει από το online κατάστημα Google Play (πρώην Android Market) της Google όπως και από άλλα sites. Μέχρι τον Ιούλιο του 2015 περισσότερες από 1.600.000 εφαρμογές ήταν διαθέσιμες για Android ενώ εκτιμάται ότι ο αριθμός των downloads από το Android Market μέχρι τον Ιούλιο του 2013 είχε υπερβεί τα 50 δισεκατομμύρια.

2.2 Εξέλιξη του Android



Εικόνα 9

Όπως αναφέραμε παραπάνω, το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα. Η εξέλιξη του λόγω της open source φύσης του είναι ραγδαία και αυτό αντικατοπτρίζεται στο γεγονός ότι οι 7 κύριες εκδόσεις του έχουν κυκλοφορήσει σε διάστημα 2.5 ετών, από τον Απρίλη του 2009 μέχρι τον Νοέμβριο του 2011. Στην πληροφορική συνηθίζεται τα προϊόντα hardware και software να κυκλοφορούν εκτός από τον αριθμό έκδοσης τους, και με μία κωδική ονομασία. Η κωδική ονομασία μπορεί να είναι για παράδειγμα ονόματα πόλεων (Windows Vienna, Chicago) ή ονόματα ζώων (OSX Leopard, Lion). Στην περίπτωση όμως του Android, η κωδική ονομασία έρχεται στη μορφή επιδόρπιου.

2.2.1 Οι εκδόσεις

Από το 2008 που βγήκε το πρώτο Android Smartphone μέχρι σήμερα, έχουν βγει αρκετές εκδόσεις οι οποίες πάντα παίρνουν το όνομά τους από γλυκά και επιδόρπια. Η ονομασία των εκδόσεων αυτών ακολουθεί αλφαβητική σειρά, π.χ Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, κλπ. Ας δούμε με χρονολογική σειρά τις κυριότερες εκδόσεις Android που κυκλοφόρησαν από το 2008 μέχρι σήμερα:



Έκδοση 1.0

Ημερομηνία Κυκλοφορίας: 23 Σεπτεμβρίου 2008



Έκδοση 1.1

Ημερομηνία Κυκλοφορίας: 9 Φεβρουαρίου 2009



Έκδοση 1.5 (CupCake)

Ημερομηνία κυκλοφορίας : 30 Απριλίου 2009

Αλλαγές:

1. Δυνατότητα εγγραφής και παρακολούθησης video στο CamCorder mode.
2. Ανέβασμα video στο youtube, και εικόνες στο Picasa, απευθείας από το τηλέφωνο.
3. Εικονικό πληκτρολόγιο με πρόβλεψη λέξεων.
4. Υποστήριξη Bluetooth A2DP και AVRCP.
5. Δυνατότητα αυτόματης σύνδεσης ακουστικών headset σε συγκεκριμένη απόσταση.

6. Νέα Widgets, και δυνατότητα προσθήκης φακέλων στην αρχική οθόνη.
7. Κινούμενα Screen Transitions.



Έκδοση 1.6 (Donut)

Ημερομηνία κυκλοφορίας : 15 Σεπτεμβρίου 2009

Αλλαγές:

1. Βελτιωμένο Android Market.
2. Δυνατότητα πολλαπλής επιλογής φωτογραφιών για επεξεργασία από τη Gallery.
3. Αναβαθμισμένο voice Search.
4. Αναβαθμισμένες δυνατότητες αναζήτησης από την κεντρική οθόνη που πλέον περιλαμβάνει και bookmarks, ιστορικό, επαφές κ.α
5. Υποστήριξη CDMA/EVDO, 802.1x, VPNs, και text-to-speech.
6. Υποστήριξη οθόνες αναλύσεων WVGA.
7. Βελτίωση ταχύτητας στις εφαρμογές αναζήτησης και την κάμερα.
8. Gesture framework και εργαλείο ανάπτυξης GestureBuilder.
9. Δωρεάν turn-by-turn πλοήγηση από τη Google.



Έκδοση 2.0, 2.1 (Eclair)

Ημερομηνία κυκλοφορίας : 26 Οκτωβρίου 2009(2.0), 3 Δεκεμβρίου 2009(2.0.1),
12 Ιανουαρίου 2010(2.1)

Αλλαγές:

1. Βελτιστοποίηση ταχύτητας hardware.
2. Υποστήριξη για μεγαλύτερες αναλύσεις και μεγέθη οθονών.
3. Ανανεωμένο UI.
4. Νέο Browser UI και υποστήριξη HTML5.
5. Νέα λίστα επαφών.
6. Καλύτερο Contrast Ratio για τα Backgrounds.
7. Google Maps 3.1.2
8. Microsoft Exchange Server, για υποστήριξη ActiveSync 2.5
9. Ενσωματωμένη υποστήριξη για Flash στην κάμερα.
10. Ψηφιακό ζούμι.
11. Δυνατότητα αντίληψης Multitouch
12. Βελτιωμένο εικονικό πληκτρολόγιο.
13. Bluetooth 2.1.
14. Live Wallpapers.



Έκδοση 2.2 (Froyo)

Ημερομηνία κυκλοφορίας : 20 Μαΐου 2010

Το Froyo είναι εγκατεστημένο στο μεγαλύτερο μέρος των συσκευών συγκεντρώνοντας το 65.9% αυτών.

Αλλαγές:

1. Βελτιστοποίηση στην ταχύτητα του OS, της διαχείριση μνήμης και την γενική απόδοση.

2. Ενσωμάτωση του Chrome V8 JavaScript στα Browsers applications.
3. Αναβαθμισμένη υποστήριξη Microsoft Exchange
4. Βελτιωμένος Application Launcher με συντόμευσεις για τις εφαρμογές τηλεφώνου και Browser.
5. USB Tethering και λειτουργία WiFi Hotspot.
6. Επιλογή απενεργοποίησης για δεδομένα μέσω κινητών δικτύων (Data Access over Mobile Network).
7. Αναβαθμισμένο Market με δυνατότητα αυτόματων updates.
8. Γρήγορη μετάβαση ανάμεσα στις γλώσσες του πληκτρολογίου.
9. Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κωδικούς.
10. Επιλογή εγκατάστασης εφαρμογών στην κάρτα μνήμης.
- 11.Υποστήριξη Adobe Flash 10.1.
- 12.Υποστήριξη για οθόνες με μεγάλο dpi όπως οθόνες 4 ιντσών 720p.



Έκδοση 2.3(GingerBread)

Ημερομηνία Κυκλοφορίας 6 Δεκεμβρίου 2010

Αλλαγές:

1. Ενημερωμένο UI Design.
2. Υποστήριξη για πολύ μεγάλα μεγέθη οθονών και αναλύσεων(WXGA και μεγαλύτερες).
3. Προεγκατεστημένη υποστήριξη για VoIP telephony.
4. Υποστήριξη για WebM/VP8 video playback αλλά και AAC audio encoding.
5. Νέα ηχητικά εφέ όπως τα reverb, equalization, headphone virtualization, και bass boost.

6. Επανασχεδιασμένο Multi-touch πληκτρολόγιο.
7. Λειτουργίες Copy-paste σε όλο το λειτουργικό.
8. Αυξημένη υποστήριξη για development.
9. Βελτιώσεις ήχου και γραφικών για τους devs παιχνιδιών.
10. Προεγκατεστημένη υποστήριξη για περισσότερους αισθητήρες (όπως γυροσκόπιο και βαρόμετρο).
11. Download manager για κατέβασμα μεγάλων αρχείων.
12. Βελτιωμένη διαχείριση ενέργειας και έλεγχος των εφαρμογών.
13. Προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες.
14. Μετατόπιση από το YAFFS σε ext4 filesystem.



Έκδοση 3.0, 3.1, 3.2 (honeycomb)

Είναι η έκδοση αποκλειστικά για tablets. Έφερε αλλαγές κυρίως στο γραφικό περιβάλλον και πρόσθεσε υποστήριξη πολλαπλών πυρήνων μαζί με βελτιωμένα γραφικά



Έκδοση Ice cream Sandwich (4.0)

Το Android Ice Cream Sandwich θα φέρει μαζί του “ολογραφικό” 3D UI, πλουσιότερα widgets, ενισχυμένο multitasking, νέα APIs και φυσικά θα είναι πλήρως

open source. Για να πάρουμε μια καλή γεύση του Android Ice Cream Sandwich, η Google παρουσίασε τη λειτουργία face tracking, η οποία δεν περιορίζεται μόνο στις κινήσεις του κεφαλιού, αλλά ακολουθεί τα μάτια, τη μύτη

και το στόμα. Στην περίπτωση που πραγματοποιείται video κλήση, “ζουμάρει” αυτόματα προς αυτόν που μιλά εκείνη τη στιγμή.



Έκδοση Jelly Bean 4.1- 4.2.

Κυκλοφόρησε τον Ιούνιο του 2012 και αποτελεί την καλύτερη έκδοση του Android μέχρι σήμερα. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορα και καλοφτιαγμένα από ποτέ ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην χρήση φωνής (υπαγόρευση κειμένου) και στην κάμερα. Η πρώτη συσκευή που είχε το Jelly Bean ήταν το tablet Google Nexus 7 ενώ η έκδοση Android 4.2 πρωτοεμφανίστηκε στα Nexus 4 και Nexus 10.

- Το Android 4.2, Jelly Bean βελτιώνει την ταχύτητα και την απλότητα του Android 4.1 και περιλαμβάνει όλα τα νέα χαρακτηριστικά, Photo Sphere, μια εντελώς επανασχεδιασμένη εφαρμογή για την κάμερα, νέο πληκτρολόγιο Gesture Typing, Google Now και πολλά άλλα.
- Τα πάντα στο Jelly Bean γίνονται γρήγορα, ευέλικτα και ομαλά. Η μετακίνηση μεταξύ των οθονών home και η εναλλαγή μεταξύ των εφαρμογών είναι εύκολη, σαν να γυρίζουμε τις σελίδες ενός βιβλίου.
- Το Jelly Bean βελτιώνει την απόδοση σε όλο το σύστημα, συμπεριλαμβανομένης της ταχύτερης αλλαγής προσανατολισμού, της ταχύτερης εναλλαγής μεταξύ πρόσφατων εφαρμογών, και ομαλότερη και πιο συνεπή απόδοση σε όλο το σύστημα μέσω vsync και τριπλού buffering.

- Το Jelly Bean έχει πιο αντιδραστική και ομοιόμορφη ανταπόκριση στην αφή, και κάνει την Android συσκευή σας να ανταποκρίνεται καλύτερα με την ενίσχυση της CPU αμέσως μόλις αγγίζετε την οθόνη, ενώ βελτιώνει τη διάρκεια ζωής της μπαταρίας.



Έκδοση Android 5.0 “Lollipop”

Κυκλοφόρησε τον Ιούνιο του 2014 και εισάγει ορισμένες σημαντικές καινοτομίες όπως:

- Αυξημένο έλεγχο της συσκευής από τον χρήστη. Ο χρήστης μιας συσκευής με Lollipop μπορεί να είναι επιλεκτικός και να ορίσει ποιες κλήσεις ή ειδοποιήσεις θα δέχεται, ανάλογα με το πρόγραμμά του.
- Διακηρυγμένη επέκταση της αυτονομίας της μπαταρίας έως και 90 λεπτά, υπό την προϋπόθεση ότι η μπαταρία του smartphone με Android Lollipop είναι επαρκώς εξαντλημένη.
- Δυνατότητα ορισμού διαφορετικών λογαριασμών χρηστών -χρήσιμο ιδιαίτερα στα tablet. Οι επισκέπτες μπορούν να χρησιμοποιούν την συσκευή σε Guest Mode, ώστε να μην έχουν πρόσβαση σε προσωπικό υλικό.
- Νέοι τρόποι κλειδώματος από το καθιερωμένο PIN, το password, ή ένα μοτίβο αλλά και το Smart Lock με το οποίο η συσκευή θα ξεκλειδώνεται μόνο όταν βρεθεί σε ζεύξη με μια άλλη ορισμένη ως έμπιστη συσκευή, δηλαδή το έξυπνο ρολόι ή το αυτοκίνητο με τον σχετικό εξοπλισμό.

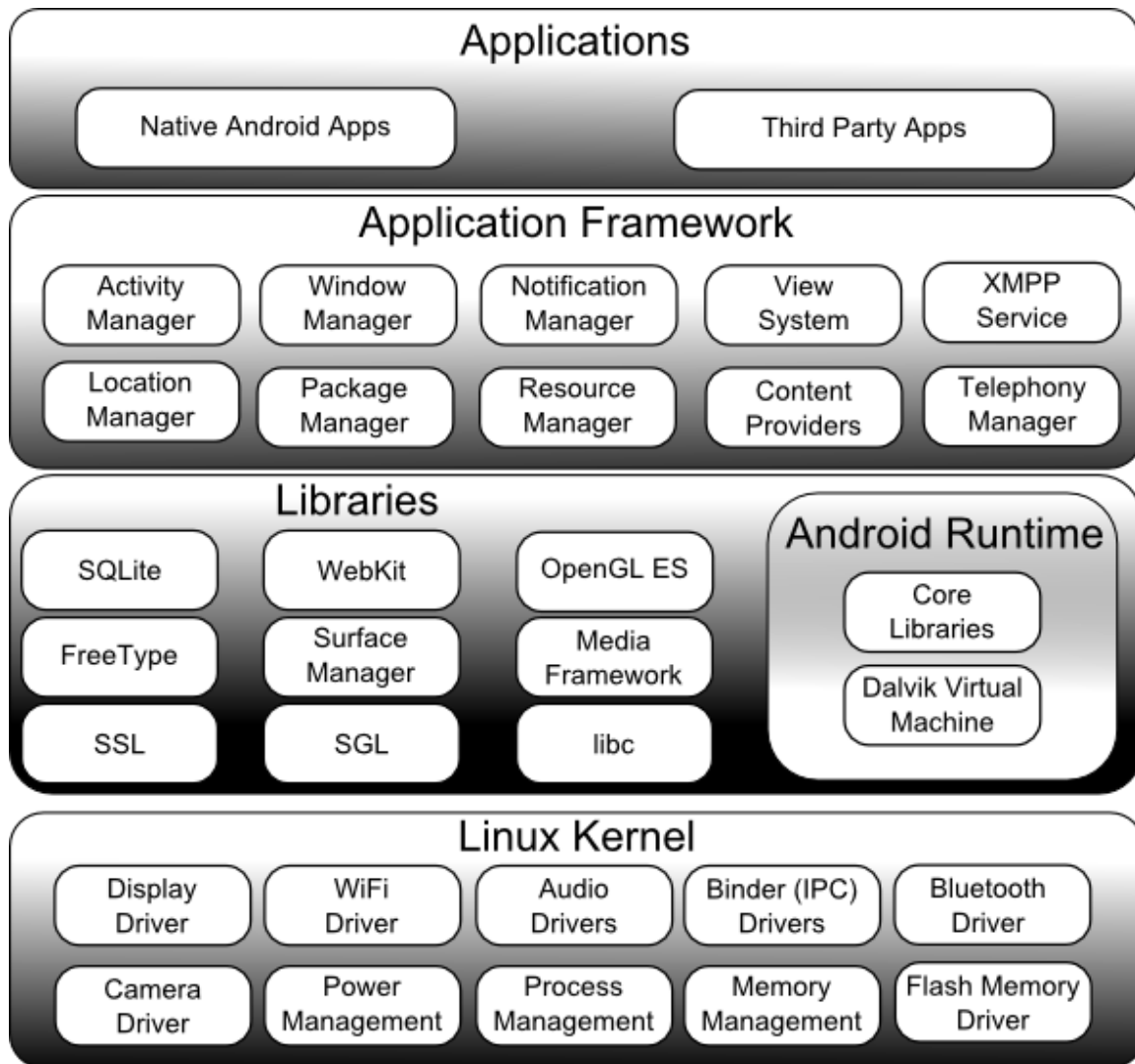


Έκδοση Android 6.0 “Marshmallow”

Κυκλοφόρησε τον Οκτώβριο του 2015 με την προοπτική να διορθώσει ορισμένα κακώς κείμενα των προκατόχων του, όπως:

- Αναγνώριση δακτυλικού αποτυπώματος.
- Οι εφαρμογές ζητούν άδεια από το χρήστη πριν αποκτήσουν πρόσβαση σε διάφορα δεδομένα.

2.2.2 Αρχιτεκτονική του Android



Εικόνα 10

Το Android δεν είναι μόνο ένα λειτουργικό σύστημα. Είναι μια στοίβα λογισμικού η οποία αποτελείται από το λειτουργικό σύστημα, τις υπηρεσίες διασύνδεσης με τις εφαρμογές (middleware) και τέλος από τις κύριες (core) εφαρμογές, μεταξύ αυτών, ενός email client, μιας εφαρμογής διαχείρισης SMS, ενός ημερολογίου, ενός browser, εφαρμογή διαχείρισης επαφών, και άλλες οι οποίες έρχονται δεμένες με την υπόλοιπη στοιβάδα λογισμικού του Android. Η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 4 βασικά επίπεδα:

- Τον πυρήνα Linux (Linux Kernel)

- Τις εγγενείς και τις προηγμένες βιβλιοθήκες (Libraries)
 - Την εικονική μηχανή Dalvik (Dalvik VM) & τον χρόνο εκτέλεσης (Android Runtime).
 - Το πλαίσιο εφαρμογής (Application Framework)
- Ας δούμε το κάθε ένα αναλυτικά:

1^ο επίπεδο: Πυρήνας Linux (Linux Kernel)

Η βάση της στοίβας λογισμικού του Android είναι ο πυρήνας Linux. Ο τροποποιημένος πυρήνας του συστήματος βασίζεται στην έκδοση 2.6 (και στην έκδοση 3.0.1 για το Android 4.0) του Linux Kernel, η οποία υποστηρίζει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers). Οι οδηγοί αυτοί είναι υπεύθυνοι για την επικοινωνία του software με το hardware της συσκευής. Ενδεικτικά ο πυρήνας του Android περιέχει:

- Οδηγό προβολής οθόνης.
- Οδηγό Wifi και Bluetooth.
- Οδηγό κάμερας.

Ο πυρήνας του Android μπορεί να βασίζεται στον πυρήνα του Linux, αλλά διαφέρει αρκετά από αυτόν. Ο λόγος είναι οι αλλαγές στην αρχιτεκτονική που έχει κάνει η Google για να είναι ελαφρύτερος και βελτιστοποιημένος για χρήση σε κινητές συσκευές. Αυτό σημαίνει ότι παρότι το Android είναι κατά βάση Linux, επί της ουσίας είναι αρκετά δύσκολο να τρέξουν εφαρμογές ή να χρησιμοποιηθούν βιβλιοθήκες από τη μία πλατφόρμα στην άλλη. Ο Linus Torvalds έχει αναφέρει ότι τελικά στο μέλλον το Android και το Linux θα μοιράζονται έναν κοινό πυρήνα, αλλά αυτό θα αργήσει 4-5 χρόνια ακόμα.

2ο επίπεδο: Εγγενείς βιβλιοθήκες

Στο δεύτερο επίπεδο της στοίβας έχουμε τις βιβλιοθήκες του Android. 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 δυνατότητες των βιβλιοθηκών του Android γίνονται εμφανείς στους προγραμματιστές στην στοίβα του πλαισίου εφαρμογής. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. 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.
- **Βιβλιοθήκες Πολυμέσων** – Υποστηρίζει αναπαραγωγή και εγγραφή πολλών δημοφιλών μέσων ήχου και εικόνας, όπως: MPEG4, H.264, MP3, AAC, AMR, JPG, και PNG
- **Surface Manager** – Διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές.
- **LibWebCore** – Μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (browser engine) η οποία χρησιμοποιείται και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές.
- **SGL** – Η γνωστή μηχανή δισδιάστατων γραφικών.
- **Βιβλιοθήκες 3D** – Μια υλοποίηση βασισμένη στα APIs του OpenGL ES 1. 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.
- **FreeType** – Παρέχει ευκρίνεια γραφικών στα bitmaps και τις γραμματοσειρές των εφαρμογών του συστήματος.
- **SQLite** – Μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων.

Η εικονική μηχανή Dalvik & Χρόνος Εκτέλεσης Εφαρμογής (Android Runtime).



Εικόνα 11

Dalvik Virtual Machine: Όπως προαναφέραμε το Android αποτελείται από 4 επίπεδα και 5 ομάδες. Μία απ' αυτές τις ομάδες είναι η Dalvik Virtual Machine. Η “μηχανή” αυτή είναι υπεύθυνη για την δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να “τρέξει” το λειτουργικό σύστημα. Κάθε εφαρμογή του Android είναι γραμμένη σε γλώσσα Java, την οποία το λειτουργικό σύστημα δεν την αντιλαμβάνεται απευθείας. Για τον λόγο αυτό η Dalvik Virtual Machine αναλαμβάνει τη δημιουργία των εκτελέσιμων αρχείων .dex (Dalvik Executable) τα οποία εκτελούνται από το λειτουργικό σύστημα. Η Dalvik δεν υποστηρίζει τον κώδικα bytecode, αντί αυτού οι κλάσεις της Java γίνονται compile σε αρχεία .dex ώστε να τρέξουν στην VM. Τα αρχεία dex ουσιαστικά αποτελούν συμπιεσμένα δεδομένα για εξοικονόμηση χώρου κατά την εκτέλεση. 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. Κάθε εκτελέσιμο πρόγραμμα εκτελείται από την δική του Virtual Machine, ακόμα και όταν εκτελούνται παράλληλα, με αποτέλεσμα τα διαφορετικά προγράμματα να μην επηρεάζουν το ένα το άλλο και σε περίπτωση που προκύψει κάποιο σφάλμα σε ένα απ' αυτά, να μην προκαλέσει προβλήματα στα υπόλοιπα. 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.

Android Runtime: Ο χρόνος εκτέλεσης των εφαρμογών του Android, βρίσκεται στο ίδιο επίπεδο με τις κύριες βιβλιοθήκες και την μηχανή Dalvik. Εδώ βρίσκουμε το κοινό σημείο επαφής μεταξύ των δυνατοτήτων που παρέχουν οι βιβλιοθήκες και του

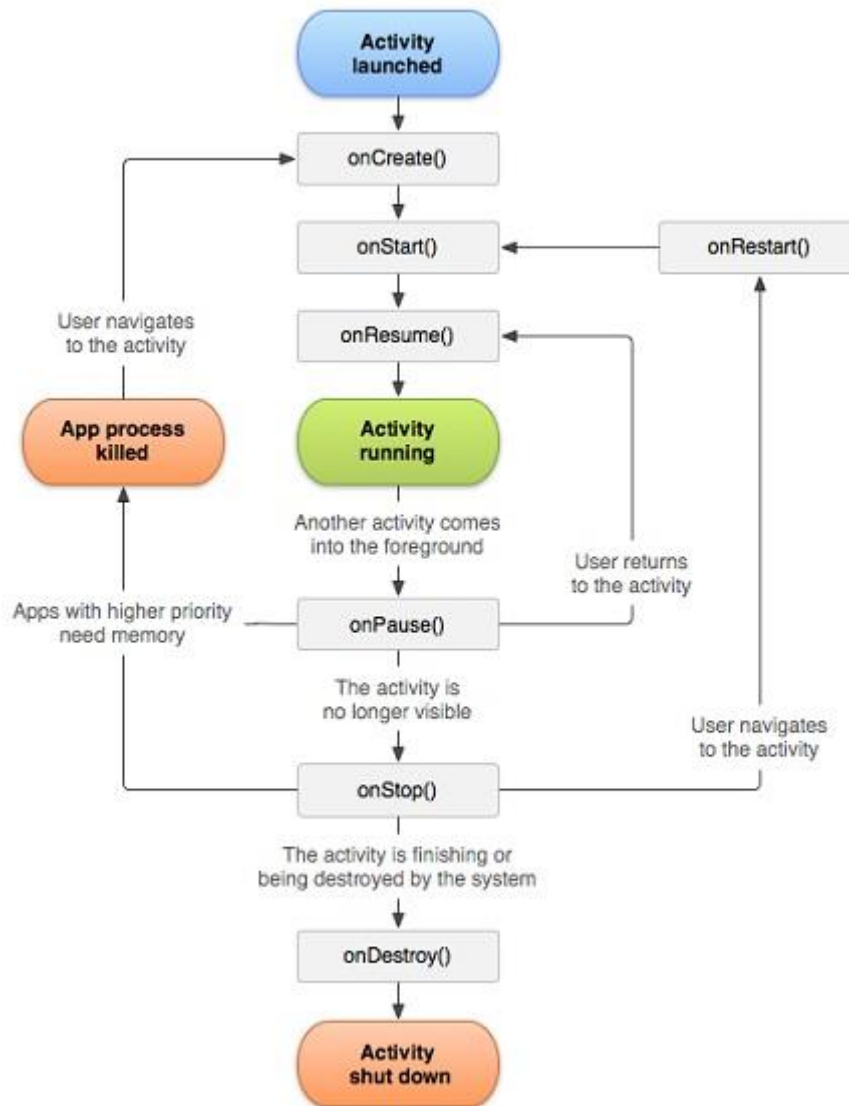
χρόνου εκτέλεσης της εικονικής μηχανής Dalvik τις λειτουργίες τις οποίες, περιγράψαμε παραπάνω.

3^ο επίπεδο: Πλαίσιο Εφαρμογής (Application Framework)

Στο επόμενο επίπεδο της αρχιτεκτονικής του Android λοιπόν, συναντάμε το πλαίσιο των εφαρμογών. Οι developers έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι:

- **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, κουμπιά, κλπ.
- **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.
- **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, κλπ.

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.



Εικόνα 12(Κύκλος ζωής μιας δραστηριότητας (Activity lifecycle)

4ο επίπεδο: Εφαρμογές και Widgets

Στο υψηλότερο επίπεδο της στοίβας Android, φιγουράρουν οι εφαρμογές και τα widgets. Αυτό είναι που βλέπουν οι χρήστες χωρίς να γνωρίζουν την όλη από κάτω διαδικασία. Αυτές είναι εφαρμογές που γράφουν οι κατασκευαστές λογισμικού, εφαρμογές που ήδη είναι εγκατεστημένες στο τηλέφωνο ή που ο χρήστης παίρνει από το Android Market. Οι εφαρμογές είναι προγράμματα που καταλαμβάνουν ολόκληρη την οθόνη και αλληλεπιδρούν με το χρήστη. Από την άλλη τα widgets λειτουργούν σε μικρά τετράγωνα μέσα στην αρχική οθόνη – εφαρμογή.

2.2.3 Ασφάλεια στο Android



Εικόνα 13

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

- Η κάθε εφαρμογή αντιμετωπίζεται σαν διαφορετικός χρήστης.
- 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.
- Κάθε εφαρμογή τρέχει στη 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 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 της παρέχει πρόσβαση στους πόρους που ζητάει.

ΚΕΦΑΛΑΙΟ 3: Ανάπτυξη Εφαρμογής – Λέξεις κλειδιά

3.1 Κύκλος Ανάπτυξης Εφαρμογής

Η ανάπτυξη εφαρμογών στο Android είναι μια σύνθετη και χρονοβόρα διαδικασία η οποία συνοψίζεται σε 4 βασικά στάδια, αλλά και αρκετά επί μέρους, τα οποία θα σχολιαστούν μεταξύ των βασικών.

3.1.1 Εγκατάσταση Λογισμικού

Στο πρώτο στάδιο της ανάπτυξης ο προγραμματιστής καλείται να στήσει το περιβάλλον εργασίας στο οποίο θα γίνει ο σχεδιασμός, η ανάπτυξη, ο έλεγχος, και η λειτουργία των εφαρμογών. Μπορεί να επιλέξει όποιο περιβάλλον ανάπτυξης (IDE) τον εξυπηρετεί καλύτερα και να χρησιμοποιήσει όλα τα εργαλεία του Android SDK μηδενός εξαιρουμένου. Στη συνέχεια θα πρέπει να δημιουργήσει έναν αριθμό από εικονικές συσκευές στην διαχείριση εικονικών συσκευών (AVD) για να δοκιμάσει την λειτουργία της εφαρμογής σε διαφορετικές πραγματικές συνθήκες λειτουργίας. Ιδανικά ο developer θα διαθέτει έναν αριθμό διαφορετικών φυσικών συσκευών ώστε να δοκιμάσει ο ίδιος πως συμπεριφέρεται η εφαρμογή του σε κάθε περίπτωση, όμως αυτή η πρακτική μπορεί να αποδειχθεί πολυδάπανη και χρονοβόρα. Εδώ αναλαμβάνουν δράση η ευελιξία των AVDs, για τις οποίες θα γράψουμε περισσότερα παρακάτω.

3.1.2 Ανάπτυξη Πηγαίου Κώδικα Εφαρμογής

Πρόκειται αν μη τι άλλο για τη πιο χρονοβόρα και πολύπλοκη διαδικασία. Σε αυτό το στάδιο ο προγραμματιστής πρέπει να αποφασίσει για τις δυνατότητες και το περιεχόμενο που θα περιλαμβάνει η εφαρμογή, να εντοπίσει ποιες από αυτές τις δυνατότητες είναι εφικτές και ποιες θέλουν παραπάνω έρευνα για να προστεθούν στο μέλλον, να σχεδιάσει το layout με γνώμονα την λειτουργικότητα και να αποφύγει υπερβολές στο σχεδιασμό. Τέλος να δέσει αρμονικά τον κώδικα με το layout για να φέρει το τελικό αποτέλεσμα. Η διαδικασία ξεκινάει με ένα νέο Project το οποίο θα περιέχει τον πηγαίο κώδικα, τις εικόνες, τα κείμενα και γενικά ό,τι χρειάζεται η εφαρμογή για να τρέξει ως οφείλει. Στο project του ο developer θα πρέπει να φροντίσει

ώστε το υλικό του να είναι τακτοποιημένο και ο κώδικας του ευανάγνωστος ώστε να ακολουθήσει η διαδικασία του Debugging.

3.1.3 Αποσφαλμάτωση (Debugging) και Δοκιμαστική Φάση Εφαρμογής

Η διαδικασία του debugging είναι εξίσου κρίσιμη και μερικές φορές και εξίσου χρονοβόρα με την διαδικασία ανάπτυξης του πηγαίου κώδικα της εφαρμογής. Αποτελείται από αρκετά επί μέρους στάδια τα οποία αναλύονται παρακάτω. Το πρώτο στάδιο αφορά το αρχικό χτίσιμο της εφαρμογής και η λειτουργία αυτής σε debug mode. Για να γίνει το compile της εφαρμογής φυσικά τα περισσότερα περιβάλλοντα ανάπτυξης (IDE) προϋποθέτουν ότι ο κώδικας δεν έχει κανένα συντακτικό λάθος, αλλιώς ειδοποιούν τον χρήστη να τα διορθώσει. Αφού γίνει το compile η εφαρμογή μπορεί να δοκιμαστεί είτε σε εικονική συσκευή μέσω του AVD Manager, είτε απευθείας σε φυσική συσκευή μέσω ADB push εντολής. Για το ADB θα μιλήσουμε εκτενώς παρακάτω.

Στο δεύτερο στάδιο ο προγραμματιστής καλείται να αντιμετωπίσει τα λειτουργικά και αισθητικά προβλήματα της εφαρμογής του, πρώτα εντοπίζοντάς τα στην λειτουργία της συσκευής και μετά διορθώνοντας τα κομμάτια του κώδικα που δημιουργούν τα σφάλματα. Το κύριο εργαλείο που κάνει αυτή τη διαδικασία εφικτή είναι το “LogCat” το οποίο μας επιστρέφει το stack trace του κώδικα στο σημείο εκείνο που συνέβη το σφάλμα. Υπάρχουν φυσικά και άλλα εργαλεία τα οποία θα αναλυθούν εκτενώς παρακάτω.

Στο τρίτο στάδιο ο προγραμματιστής αφού έχει τελειώσει την αποσφαλμάτωση (debugging) επιστρέφει στο βήμα ένα, δηλαδή στο compile και τη δοκιμή της εφαρμογής σε εικονική ή φυσική συσκευή ώστε να διαπιστώσει τα αποτελέσματα του 2^{ου} βήματος, της αποσφαλμάτωσης.

Η διαδικασία του debugging είναι σαν ένα βρόγχος (loop) που επαναλαμβάνεται συνέχεια μέχρι να εντοπιστούν και να διορθωθούν όλα τα σφάλματα της εφαρμογής, και για αυτό το λόγο μπορεί να αποδειχθεί εξαιρετικά χρονοβόρα.

3.1.4 Τελική έκδοση και δημοσίευση της εφαρμογής στο κοινό

Στο τέταρτο και τελευταίο στάδιο της ανάπτυξης, ο προγραμματιστής έχει να κάνει μερικές τελευταίες κινήσεις. Πρώτον πρέπει να έχει διορθώσει όλα τα σφάλματα που προέκυψαν κατά τη διαδικασία αποσφαλμάτωσης, να κάνει τις τελευταίες ρυθμίσεις και tweaks της εφαρμογής, και να κάνει το τελικό compile της εφαρμογής σε κανονική λειτουργία αυτή τη φορά και όχι debug.

Στη συνέχεια ακολουθεί η διάθεση της εφαρμογής με το μέσο της επιλογής του developer. Μπορεί να την διαθέσει στο Google Play, αφού πρώτα κάνει λογαριασμό developer, ή να την διαθέσει σε κάποιο εναλλακτικό market όπως το marketplace της Amazon. Μπορεί κατά τη δημοσίευση σε οποιοδήποτε μέσο να ορίσει τιμή πώλησης ή να διαθέσει την εφαρμογή δωρεάν.

3.2 Βασικές έννοιες - Λέξεις κλειδιά

Παρακάτω παραθέτουμε ορισμένες από τις βασικές έννοιες που χρειάζεται να γνωρίζουμε προτού προχωρήσουμε στην ανάπτυξη μίας εφαρμογής με java σε περιβάλλον eclipse.

3.2.1 Android SDK

Το Android SDK (Software Developers Kit) αποτελεί μια συλλογή εργαλείων και βιβλιοθηκών που καθιστούν εφικτή την ανάπτυξη εφαρμογών σε Android. Το λογισμικό ανάπτυξης περιλαμβάνει μια μεγάλη λίστα με εργαλεία ανάπτυξης. Σε αυτά περιλαμβάνονται:

- Εργαλεία Debugging των εφαρμογών
- Βιβλιοθήκες
- Εξομοιωτής συσκευών (Android Virtual Machines)
- Documentation
- Δείγματα Κώδικα
- Tutorials

Το SDK υποστηρίζει πολλά δημοφιλή λειτουργικά συστήματα συμπεριλαμβανομένων όλων των σύγχρονων διανομών Linux, το MAC OS X 10.4.9 και μεταγενέστερα, και τα Windows XP και τις μεταγενέστερες εκδόσεις. Το λογισμικό ανάπτυξης αποτελείται από πακέτα τα οποία βρίσκονται αποθηκευμένα σε ένα επίσημο γερο της Google, και ο προγραμματιστής μπορεί να κατεβάσει πέραν των βασικών πακέτων, και άλλα τα οποία υποστηρίζουν παλαιότερες εκδόσεις του Android, ή άλλες συσκευές εκτός κινητών συσκευών (πχ Google TV Addon). Όσον αφορά την υποστήριξη παλαιότερων εκδόσεων του Android, το SDK κάνει εφικτή την υποστήριξη σε αυτές δίνοντας στον προγραμματιστή την δυνατότητα να στοχεύσει αυτός σε πια APIs θα απευθύνεται η εφαρμογή του. Αυτό είναι αναγκαίο λόγω του ότι πολλοί χρήστες έχουν παλαιότερες λειτουργικές συσκευές οι οποίες κυκλοφορήσαν με παλαιότερες εκδόσεις του Android (πχ 1.6 ή 2.1), και ο κατασκευαστής της συσκευής δεν έχει ή δεν πρόκειται να βγάλει αναβάθμιση για την συσκευή τους. 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 SDK παρέχει ελεύθερα και δωρεάν τα απαραίτητα εργαλεία και APIs για να αναπτύξετε προγράμματα χρησιμοποιώντας την γλώσσα προγραμματισμού Java. Η ανάπτυξη λογισμικού που γίνεται με ένα plugin της Google για το Eclipse, καθώς και ο emulator τρέχουν σε Windows, Mac και σε Linux.

3.2.2 Integrated Development Environment

IDE (Integrated Development Environment). Το ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (IDE) είναι ένα γραφικό περιβάλλον το οποίο έχει στόχο να βοηθήσει το χρήστη του να γράψει προγράμματα για μια συγκεκριμένη γλώσσα προγραμματισμού (ή καμιά φορά και για περισσότερες από μία). Ένα IDE συνήθως περιλαμβάνει έναν επεξεργαστή κειμένου ο οποίος δείχνει με συγκεκριμένο τρόπο (π.χ. χρωματίζοντας κάποιες λέξεις της γλώσσας προγραμματισμού με συγκεκριμένο χρώμα) τον πηγαίο κώδικα βοηθώντας τον προγραμματιστή να γράψει σωστά

προγράμματα. Επίσης, μπορεί να περιλαμβάνει εργαλεία αποσφαλμάτωσης του κώδικα, γραφικά εργαλεία για την μοντελοποίηση του λογισμικού και για την σχεδίαση της διεπαφής με τον χρήστη, όπως και αυτόματη παραγωγή κώδικα από αυτά. Πολύ γνωστό IDE είναι το eclipse για την γλώσσα Java ή την C/C++ και είναι το περιβάλλον στο οποίο θα δουλέψουμε εμείς.

3.2.3 Χρήσιμες έννοιες

Java Η Java είναι μια σύγχρονη γλώσσα αντικειμενοστραφούς προγραμματισμού η οποία δημιουργήθηκε το 1995 από την Sun Microsystems. Η μεγάλη επιτυχία που γνώρισε αμέσως οφείλεται στην ικανότητά της να είναι ανεξάρτητη πλατφόρμα εκτέλεσης (το ίδιο αρχείο τρέχει και σε Linux και σε Windows), στην ευκολία εκμάθησης, στην ιδιότητά της να μην ξεφεύγει από τον χώρο μνήμης που είναι διαθέσιμος και την ομοιότητά της με την πασίγνωστη C++.

XML Η XML είναι μια γλώσσα που χρησιμοποιείται για την περιγραφή μιας σημειακής γλώσσας. Ο τεχνικός όρος μιας τέτοιας γλώσσας είναι **μετα-γλώσσα**. Τα αρχικά XML προέρχονται από το Extensible Markup Language που μεταφράζεται ως "επεκτάσιμη γλώσσα σήμανσης" και μοιάζει αρκετά με τη γλώσσα HTML. Αποτελεί πλέον το πρότυπο για την αναπαράσταση και αποθήκευση των δεδομένων που διακινούνται μέσω του Internet.

OSS - Open source software (λογισμικό ανοιχτού πηγαίου κώδικα)OSS είναι το λογισμικό εκείνο το οποίο αναπτύσσεται, ελέγχεται για την ορθότητά του και βελτιώνεται μέσω της συνεργασίας της παγκόσμιας κοινότητας προγραμματιστών. Ο όρος γεννήθηκε το 1998 και τυπικά ορίζεται μέσα από τις τέσσερις ελευθερίες: α) καθένας μπορεί να το αποκτήσει, β) να το χρησιμοποιήσει, γ) να το αντιγράψει και να το διανείμει και δ) να το τροποποιήσει και να διανείμει νέες εκδόσεις. Το γνωστότερο OSS είναι το λειτουργικό σύστημα Linux.

API Ένα λειτουργικό σύστημα ή μια εφαρμογή μπορούν να διαθέτουν ένα API (application program interface) ώστε να δώσουν την δυνατότητα σε προγραμματιστές άλλων εφαρμογών να χρησιμοποιήσουν τις λειτουργίες τους. Έτσι, για παράδειγμα με τη χρήση του API του Microsoft Word κάποιος μπορεί από το πρόγραμμά του

αυτόματα να ανοίξει ένα έγγραφο του Word και να το στείλει για εκτύπωση (όπως θα μπορούσε να κάνει αν άνοιγε την ίδια την εφαρμογή Word).

DDMS (Dalvik Debug Monitor Service) Είναι μια εφαρμογή αποσφαλμάτωσης που συμπεριλαμβάνεται στο SDK και εφόσον προγραμματίζουμε σε eclipse και χρησιμοποιούμε το ADT plugin, ενσωματώνεται αυτόματα στο περιβάλλον ανάπτυξης.

Customization Το **customization** είναι η προσαρμοστικότητα μιας συσκευής και αποτελεί μία απ'τις μεγαλύτερες δυνατότητες του Android. Επιτρέπει στους χρήστες και σε διάφορους προγραμματιστές να επεξεργάζονται το interface της συσκευής τους, με επιλογές που ουσιαστικά είναι άπειρες. Αν για παράδειγμα βρει ο χρήστης κάτι που δεν του αρέσει στη συσκευή του, μπορεί να το τροποποιήσει. Είτε πρόκειται για το πιο απλό, όπως η “ταπετσαρία” της αρχικής οθόνης είτε ακόμα και για το ίδιο το λειτουργικό. Το να προσαρμόσει ο χρήστης τη συσκευή του στα δικά του δεδομένα είναι πολύ εύκολο και μπορεί να επιτευχθεί στην ουσία με τρεις κύριους τρόπους μέσω των οποίων επιτυγχάνονται απεριόριστες λύσεις και προτάσεις. Ο πρώτος τρόπος είναι με την χρήση ποικίλων εφαρμογών που υπάρχουν στο Google Play και γενικά στο διαδίκτυο. Ο δεύτερος, είναι με διάφορα θέματα (themes) που κυκλοφορούν και ο τρίτος τρόπος είναι με αλλαγή του λειτουργικού, ώστε να βρει αυτό που καλύπτει τις δικές του προδιαγραφές, είτε “χειροκίνητα”, είτε προγραμματιστικά.

ADB (Android Debug Bridge) Είναι η “Γέφυρα Εντοπισμού Σφαλμάτων του Android” και είναι ένα πρόγραμμα που περιέχεται μέσα στο Android SDK. Με αυτό το εργαλείο μπορούμε να συνδεθούμε μέσω υπολογιστή στην Android συσκευή μας και να της στείλουμε εντολές μέσω τερματικού για την εκτέλεση εργασιών όπως: 1) Την απεγκατάσταση προεγκατεστημένων εφαρμογών. 2) Το αυτόματο backup και restore των εφαρμογών μας. 3) Τη μαζική εγκατάσταση εφαρμογών είτε στην εσωτερική είτε στην εξωτερική μνήμη. 4) Να επανεκκινήσουμε τη συσκευή μας σε fastboot ή recovery mode και γενικά να μεταφέρουμε, να τροποποιήσουμε, να διαγράψουμε αρχεία σε οποιονδήποτε φάκελο και κατάμηση της συσκευής μας.

ADT (Android Developer Tools) Είναι μια σειρά από εργαλεία τα οποία ενσωματώνονται στο Eclipse IDE και παρέχουν πρόσβαση σε πολλές λειτουργίες που βοηθούν στη γρήγορη ανάπτυξη Android εφαρμογών.

Υπηρεσίες (Services) Πρόκειται για λειτουργίες της εφαρμογής οι οποίες είναι σχεδιασμένες να τρέχουν στο παρασκήνιο και να επιστρέφουν αποτελέσματα ακόμη και όταν η εφαρμογή δεν είναι στο προσκήνιο. Πχ μια εφαρμογή media player μπορεί μέσω μιας υπηρεσίας να συνεχίσει να παίζει μουσική ακόμη και όταν το κύριο παράθυρο της εφαρμογής δεν βρίσκεται στο προσκήνιο.

Πάροχος Περιεχομένου (Content Providers) Η ανταλλαγή δεδομένων από μια εφαρμογή στην άλλη όπως είπαμε παραπάνω μπορεί να γίνει μέσω ενός Intent. Ένας πάροχος περιεχομένου όμως έχει πιο σύνθετη λειτουργία. Οι content providers μιας εφαρμογής διαχειρίζονται συγκεκριμένα δεδομένα της εφαρμογής τα οποία έχει ορίσει ο προγραμματιστής κατά την κατασκευή του. Συνηθισμένα δεδομένα τα οποία μοιράζονται μέσω Content Providers, είναι οι βάσεις δεδομένων SQLite μιας εφαρμογής, και οι επαφές του χρήστη.

Δέκτες Μετάδοσης (Broadcast Receivers) Πρόκειται για ένα είδος υπηρεσιών οι οποίες αντιλαμβάνεται κάποια γεγονότα του συστήματος και αναλαμβάνουν να ενημερώσουν το σύστημα ή τις υπόλοιπες εφαρμογές. Ο σκοπός τους είναι διπλός καθώς μπορούν και να ενημερωθούν για κάποιο συμβάν από άλλες εφαρμογές, αλλά και να ειδοποιήσουν τις υπόλοιπες εφαρμογές και το σύστημα για κάποιο συμβάν που τις ενεργοποίησε. Δεν έχουν γραφικό περιβάλλον αλλά μπορούν να προβάλουν ειδοποίηση στον χρήστη μέσω της μπάρας ειδοποιήσεων. Συνήθως χρησιμοποιούνται ως διαμεσολαβητές μεταξύ των Activities και των Services μιας εφαρμογής.

3.3 Java Keywords

3.3.1 Προσδιοριστές πρόσβασης

Οι προσδιοριστές πρόσβασης είναι λέξεις οι οποίες μπαίνουν μπροστά από μεταβλητές, μεθόδους, και κλάσεις και δείχνουν αν στα εν λόγω μέλη μπορούμε να έχουμε πρόσβαση από άλλη κλάση ή από άλλο πακέτο. Είναι οι εξής:

public: Αυτός ο προσδιοριστής δεν απαγορεύει την πρόσβαση από πουθενά. Όποιο μέλος δηλωθεί ως public μπορεί να προσπελαστεί από όλες τις κλάσεις, σε οποιοδήποτε πακέτο.

private: Ο προσδιοριστής private είναι αυτός που μας περιορίζει περισσότερο απ' όλους: Όσα μέλη έχουν δηλωθεί ως private, μπορούν να προσπελαστούν ΜΟΝΟ από την κλάση στην οποία έχουν δηλωθεί.

protected: Όσα μέλη δηλώνονται protected, μπορούμε να τα προσπελάσουμε από όλες τις κλάσεις του ίδιου πακέτου (default πρόσβαση), καθώς και από τις κλάσεις οποιουδήποτε πακέτου που κληρονομούν την κλάση που έχει το εν λόγω protected μέλος.

3.3.2 Κλάσεις

class: Καθορίζει ότι ο κώδικας που ακολουθεί στα άγκιστρα ανήκει σε μια κλάση:
`class myclass{<σώμα της κλάσης>}`

static: Αν ένα μέλος μιας κλάσης δηλωθεί ως static τότε το μέλος αυτό παύει να είναι στιγμιότυπο και για να το καλέσουμε δε χρειάζεται η δημιουργία αντικειμένου.

3.3.3 Πακέτα και διεπαφές

package: Με τη λέξη αυτή καθορίζουμε ένα πακέτο. Όλες οι κλάσεις που ακολουθούν της δήλωσης αυτής, η οποία πρέπει να είναι στην αρχή του προγράμματος, συμπεριλαμβάνονται στο εν λόγω πακέτο. Εδώ πρέπει να σημειώσουμε ότι το αρχείο .class που θα δημιουργηθεί θα βρίσκεται σ' ένα κατάλογο με το όνομα που δώσαμε στο πακέτο. Αν ο κατάλογος δεν υπάρχει τότε η JAVA θα δημιουργήσει έναν.

import: Σε περίπτωση που θέλουμε να εισάγουμε τις κλάσεις ενός άλλου πακέτου στο πρόγραμμά μας, τότε χρησιμοποιούμε τη λέξη import και δίπλα το όνομα της κλάσης.

interface: Η λέξη αυτή χρησιμοποιείται αντί της class όταν θέλουμε να δηλώσουμε ότι ο κώδικας στο επόμενο block είναι μια διεπαφή.

3.3.4 Δομές επιλογής & κληρονομικότητα

switch: Η switch είναι μια δομή επιλογής εναλλακτική της σκάλας if-else-if.
switch(<μεταβλητή>) {<περιπτώσεις>}

case: Η λέξη κλειδί case καθορίζει τις περιπτώσεις. Δηλαδή αν για κάποια τιμή της μεταβλητής στη switch θέλουμε να συμβεί κάτι θα γράψουμε: case <τιμή>: <κώδικας>

Κληρονομικότητα

extends: Αν θέλουμε να δηλώσουμε ότι η κλάση η οποία συντάσσουμε θα κληρονομήσει μια άλλη κλάση τότε ακριβώς δίπλα από τον ορισμό της προσθέτουμε τη λέξη extends και δίπλα το όνομα της κλάσης που θέλουμε να κληρονομηθεί.

super: Η λέξη κλειδί super κάνει δύο δουλειές. Πρώτα – πρώτα καλεί το δομητή της υπερκλάσης. Αυτό στην περίπτωση που συντάσσεται ως μέθοδος (super()). Επιπλέον, αν τη συντάξουμε ως κλάση (super.<όνομα μέλους>), μπορούμε να καλέσουμε τη μέθοδο ή τη μεταβλητή η οποία υπερβαίνεται ή «σκιάζεται» αντίστοιχα.

3.3.5 Λέξεις – κλειδιά “πολλαπλών ρόλων”

break: Με τη λέξη κλειδί break μπορούμε να κάνουμε τις εξής δουλειές:

1) Κατ’ αρχήν η break είναι βασικό συστατικό της switch. Είναι απαραίτητη για την έξοδο από ένα case της switch. Χωρίς αυτή, αν κάποιο case συμβεί, τότε θα γίνουν και όλα τα παρακάτω μέχρι να τελειώσει το block της switch.

2) Μπορεί επίσης να τερματίσει ανά πάσα στιγμή έναν οποιοδήποτε βρόχο. Αν έχουμε ένα loop και με κάποια συνθήκη που θα επαληθευτεί, εκτελεστεί ένα break, τότε ο βρόχος θα σταματήσει χωρίς να γίνει καμία απολύτως ενέργεια μέσα σ’ αυτόν. Στην περίπτωση που ο βρόχος στον οποίο θα εκτελεστεί η break, είναι εμφωλευμένος, τότε σπάει μόνο αυτός και οι υπόλοιποι εκτελούνται κανονικά.

3) Τέλος η break μπορεί να χρησιμοποιηθεί και ως μια εναλλακτική μορφή της goto, με κάποιους περιορισμούς. Αυτό μπορεί να γίνει βάζοντας μία ετικέτα σ’ ένα block (label1:{...code...}). Το μόνο που απομένει είναι να πούμε στο πρόγραμμα να πάει στην ετικέτα (break label1;). Πρέπει όμως να είμαστε προσεκτικοί στη χρήση της γιατί αν από ένα block, ζητήσουμε στο πρόγραμμα να πάει σε κάποιο άλλο, θα δημιουργηθεί

σφάλμα χρόνου μεταγλώττισης. Αυτό συμβαίνει για να μοιάζει η όλη υπόθεση με την go to, και όχι να είναι η go to με άλλο προσωπείο.

this: Η λέξη κλειδί **this** αποτελεί μία αναφορά στο τρέχον αντικείμενο.

void: Το **void** είναι τύπος επιστροφής μεθόδου και δηλώνει ότι μια μέθοδος δε θα επιστρέψει τίποτα.

3.4 Package Explorer

Θα δούμε περιληπτικά τι περιλαμβάνει το package explorer το layout του eclipse, ουσιαστικά δηλαδή τα περιεχόμενα της εφαρμογής μας τα οποία δουλεύουμε και επεξεργαζόμαστε και τα οποία βρίσκονται στην αριστερή στήλη στο περιβάλλον του eclipse.

Οι φάκελοι src & res: Στο φάκελο **src** (source) περιέχονται τα αρχεία κλάσης της Java όλων των Activities, Services, Content Providers, βοηθητικά αρχεία, κλπ. Ο φάκελος περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται τα αρχεία του κώδικα μας.

Ο φάκελος **res** (resource) περιέχει όλα τα αρχεία εικόνας, κειμένου, xml, layout, κλπ τα οποία χρησιμοποιούνται από τις Activities που βρίσκονται στον φάκελο **src**. Φυσικά δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος **drawable** οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή μας. Σε αυτόν αποθηκεύουμε όλες τις εικόνες που θα χρησιμοποιήσουμε.

Φάκελος gen: Στο φάκελο αυτό περιέχονται τα αρχεία που παράγονται από το eclipse αυτόματα και προσθέτουν από μόνα τους τα στοιχεία που χρειάζονται και είναι καλό να μη τα επεξεργαζόμαστε. Ενημερώνονται όσο προχωράμε στον προγραμματισμό και προσθέτουμε πράγματα το eclipse.

Layout: Είναι ο φάκελος ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή.

Values: Είναι ο φάκελος στον οποίο αποθηκεύονται όλοι οι πόροι κειμένου που χρησιμοποιούνται στην εφαρμογή.

Φάκελος assets: Ο φάκελος assets αρχικά είναι άδειος. Εδώ μπορούμε να αποθηκεύσουμε αρχεία και καταλόγους της εφαρμογής τους οποίους όμως δε θέλουμε να διαχειριστούμε σαν πόρους της εφαρμογής. Παραδείγματος χάρη, στο φάκελο αυτό μπορούμε να δημιουργήσουμε ένα άλλο φάκελο με το όνομα Fonts και να τοποθετήσουμε μέσα αρχεία γραμματοσειρών που θα χρησιμοποιήσουμε στην εφαρμογή μας.

Libs: Είναι ο φάκελος με τις διαθέσιμες βιβλιοθήκες του project μας και περιέχει αρχεία με τη κατάληξη .jar δηλαδή αρχεία Java. Τα αρχεία αυτά ορίζουν ένα μορφότυπο συνάθροισης πολλών αρχείων σε ένα και χρησιμοποιούνται για την διανομή κλάσεων Java.

apk file: Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού γίνουν compiled μέσω του Android SDK μας δίνουν το αρχείο .apk. Το αρχείο αυτό αποτελεί την εφαρμογή και μπορούμε να το εγκαταστήσουμε στις συσκευές μας.

dex file: (dalvik executable): Είναι τα εκτελέσιμα αρχεία που προκύπτουν από την μεταγλώττιση μιας εφαρμογής μέσω του Dalvik Virtual Machine και εκτελούνται από το λειτουργικό σύστημα.

Δραστηριότητες (Activities): Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Δραστηριότητα είναι μια οθόνη διεπαφής χρήστη (GUI) και προβολής πληροφοριών. Κάθε εφαρμογή έχει τόσες Activities όσες και οι διαφορετικές οθόνες οι οποίες εμφανίζονται στον χρήστη. Όλες οι δραστηριότητες συνεργάζονται μεταξύ τους για να δώσουν στον χρήστη μια συνολική εμπειρία χρήσης της εφαρμογής.

Intents: Οι δραστηριότητες επικοινωνούν και εναλλάσσουν την λειτουργία τους μέσω των Intents. Ουσιαστικά τα Intents εξασφαλίζουν την μετάβαση από την μία δραστηριότητα σε μια άλλη και επίσης χρησιμοποιούνται για ανταλλαγή δεδομένων. Η ανταλλαγή δεδομένων, μπορεί να γίνει είτε μεταξύ των Activities μιας εφαρμογής, είτε από τη μία εφαρμογή στην άλλη. Παραδείγματος χάρη μπορούμε μέσω ενός Intent να εκκινήσουμε έναν browser ώστε να μας ανοίξει απευθείας ένα url το οποίο έχουμε παρέχει εμείς μέσω ενός Intent.

Και τέλος:

Το αρχείο AndroidManifest.xml: Κάθε project εφαρμογής περιέχει ένα αρχείο στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της εφαρμογής, και το αρχείο αυτό ονομάζεται AndroidManifest.xml. Πρόκειται όπως λέει και το όνομα του για ένα αρχείο xml μέσα στο οποίο ο προγραμματιστής καταχωρεί τις σημαντικότερες πληροφορίες της εφαρμογής για χρήση από το λειτουργικό σύστημα. Κάποιες από αυτές τις πληροφορίες είναι:

- Το όνομα του πακέτου της εφαρμογής
- Το κανονικό της όνομα που φαίνεται στον χρήστη
- Η έκδοση των APIs που χρησιμοποιούνται
- Ο αριθμός έκδοσης της εφαρμογής
- Οι άδειες χρήσης που ζητάει η εφαρμογή
- Όλες οι δραστηριότητες, πάροχοι περιεχομένου, υπηρεσίες, κλπ, που περιέχει και χρησιμοποιεί η εφαρμογή.

Όπως αντιλαμβανόμαστε πρόκειται για πολύ σημαντικό αρχείο και αποτελεί κύριο

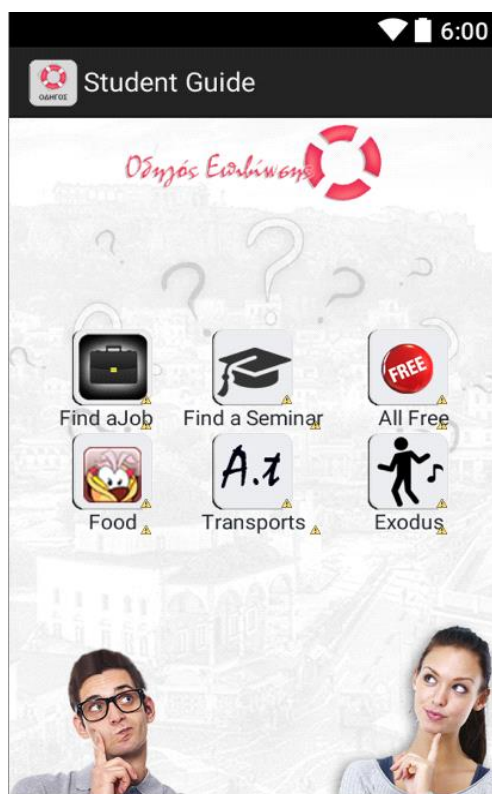
συστατικό κάθε εφαρμογής. Από αυτό το αρχείο διαβάζει το Android λειτουργικό, όταν τρέχουμε την εφαρμογή για να ξέρει πώς να την εκτελέσει με όλα τα στοιχεία που την αποτελούν. Επιπλέον το AndroidManifest προσδιορίζει τα δικαιώματα τα οποία πρέπει να έχει η τρέχον εφαρμογή για να μπορεί να αλληλοεπιδρά με άλλες εφαρμογές και ορίζει τα δικαιώματα που άλλες εφαρμογές πρέπει να έχουν για να χρησιμοποιήσουν την εφαρμογή μας.

ΚΕΦΑΛΑΙΟ 4:

Δομή “Student Guide” – Εγκατάσταση Eclipse


4.1 Δομή εφαρμογής

Για να αρχίσουμε να στήνουμε την εφαρμογή μας θα χρειαστούμε καταρχήν οχτώ αρχικές οθόνες:

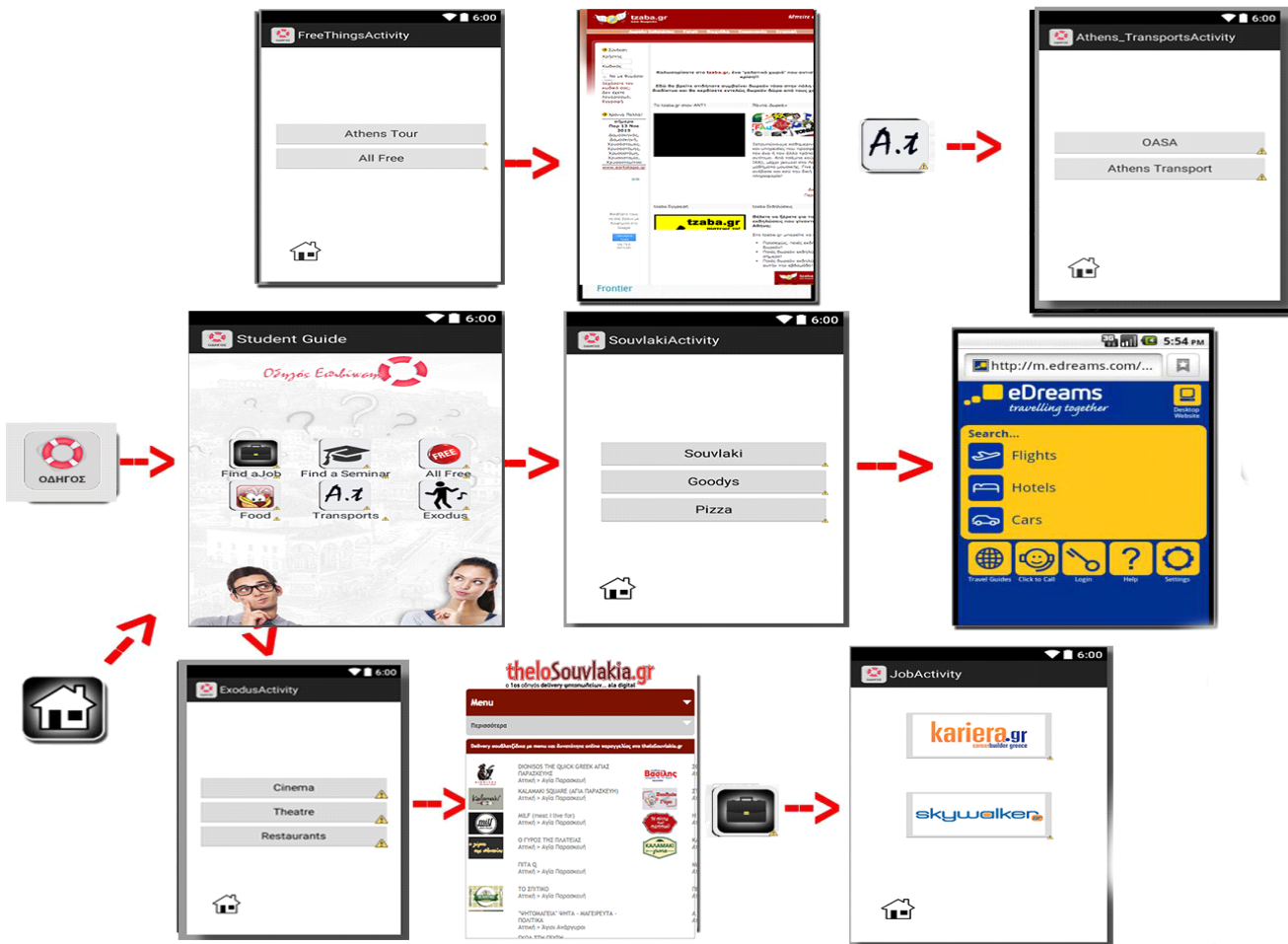


Εικόνα 14

- Οθόνη εκκίνησης Startup screen
- Οθόνη αρχική Home
- Οθόνη Find a Seminar
- Οθόνη Find a Job
- Οθόνη All Free
- Οθόνη Food
- Οθόνη Exodus
- Οθόνη Transports

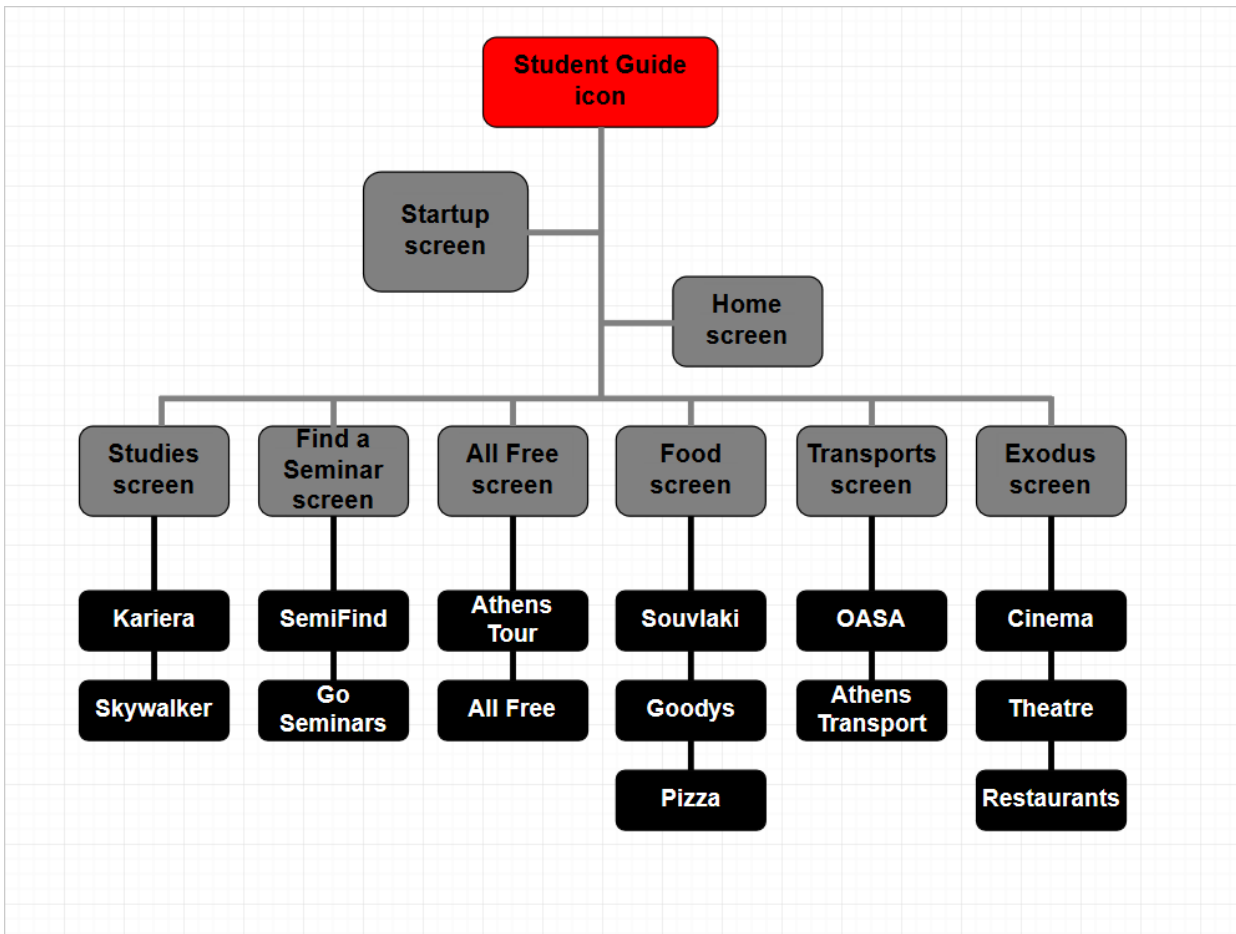
Αυτή θα είναι η αρχική σελίδα. Κάθε κουμπί στο εσωτερικό του θα έχει στο κάτω αριστερό μέρος το εικονίδιο home  που θα σε μεταφέρει ακριβώς σε αυτήν τη σελίδα. Το κάθε ένα κουμπί αντίστοιχα θα σε μεταφέρει στη δική του οθόνη. Η κάθε οθόνη θα περιέχει ένα μενού από κουμπιά, τα οποία θα συνδέονται αυτόματα με hyperlinks που θα σε μεταφέρουν σε σελίδες σχετικού περιεχομένου.

Παρακάτω παραθέτουμε ένα σχεδιάγραμμα για το πώς θα λειτουργεί η εφαρμογή το οποίο το έχουμε σχεδιάσει στο Photoshop. Το κάθε κουμπί συνδέεται με μία βάση δεδομένων αντίστοιχου περιεχομένου, όπου μπορεί ο χρήστης να αναζητήσει τις πληροφορίες που θέλει.



Εικόνα 15

4.1.2 Δενδροειδής δομή εφαρμογής



Εικόνα 16

4.2 Λειτουργικά συστήματα - Απαιτήσεις συστήματος

- Windows XP (32-bit), Vista (32- ή 64-bit), ή Windows 7 (32- ή 64-bit)
- Mac OS X 10.5.8 ή επόμενη (x86 μόνο)
- Linux (δοκιμασμένο στο Ubuntu Linux, Lucid Lynx)
- GNU C Library (glibc) 2.7 ή επόμενη έκδοση.
- Ubuntu Linux, 8.04 έκδοση ή επόμενη.

Για να μεταγλωττίσουμε και να τρέξουμε τα προγράμματά μας χρειαζόμαστε ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment IDE). Το περιβάλλον ανάπτυξης Android εφαρμογών στηρίζεται σε τρία βασικά εργαλεία, τα

οποία και τα τρία είναι δωρεάν: Java, Eclipse ή κάποιο άλλο IDE και Android. Τα συγκεκριμένα εργαλεία πρέπει να εγκατασταθούν με την σειρά με την οποία τα αναφέραμε γιατί το κάθε ένα έχει σαν προϋπόθεση ότι το προηγούμενο έχει ήδη εγκατασταθεί στο σύστημα. Στην εφαρμογή μας εμείς θα χρησιμοποιήσουμε το eclipse. Το eclipse είναι ένα IDE ελεύθερο πολυ-γλωσσικό προγραμματιστικό περιβάλλον ανάπτυξης λογισμικού για παραγωγή κώδικα σε java και κατασκευή web και mobile εφαρμογών.

Επιπλέον θα χρειαστούμε:

- Το πακέτο android SDK μία εργαλειοθήκη ανάπτυξης συστήματος λογισμικού (Software Development Kit SDK).
- Το JDT (Java Development Tools) το οποίο αποτελείται από μία σειρά από plug-ins που προσθέτουν τις δυνατότητες και τα χαρακτηριστικά ενός πλήρως εξοπλισμένου Java IDE στην πλατφόρμα eclipse. Το JDT θα το βρείτε ενσωματωμένο στις περισσότερες νέες εκδόσεις του Eclipse.
- Το JDK 6 (Java Development Kit) το οποίο είναι ένα περιβάλλον ανάπτυξης για τη δημιουργία εφαρμογών με τη χρήση της γλώσσας προγραμματισμού Java. Όλη η ανάπτυξη Android εφαρμογών στηρίζεται στην στάνταρντ έκδοση (Java Platform Standard Edition – Java SE) γλώσσα προγραμματισμού Java.

4.2.1 Εγκατάσταση Java Development Kit

Χρησιμοποιούμε το εξής link:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> και ανάλογα με το λειτουργικό μας κατεβάζουμε το κατάλληλο προϊόν, όπως στην εικόνα.

Java SE Development Kit 8u74		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	154.74 MB	jdk-8u74-linux-i586.rpm
Linux x86	174.92 MB	jdk-8u74-linux-i586.tar.gz
Linux x64	152.74 MB	jdk-8u74-linux-x64.rpm
Linux x64	172.9 MB	jdk-8u74-linux-x64.tar.gz
Mac OS X x64	227.27 MB	jdk-8u74-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.72 MB	jdk-8u74-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.09 MB	jdk-8u74-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.02 MB	jdk-8u74-solaris-x64.tar.Z
Solaris x64	96.19 MB	jdk-8u74-solaris-x64.tar.gz
Windows x86	182.01 MB	jdk-8u74-windows-i586.exe
Windows x64	187.31 MB	jdk-8u74-windows-x64.exe

Εικόνα 17

Στη συνέχεια τρέχουμε το εκτελέσιμο αρχείο και εγκαθιστούμε το Java Development Kit.

4.2.2 Διαδικασία εγκατάστασης Eclipse

Χρησιμοποιούμε το εξής link:

<https://www.eclipse.org/downloads/>

και κατεβάζουμε το πρώτο link Eclipse IDE for Java Developers επιλέγοντας πάντα το κατάλληλο προϊόν για το λειτουργικό μας.



Εικόνα 18

Το αρχείο που κατεβάσαμε είναι σε μορφή .zip. Το Eclipse δεν χρειάζεται εγκατάσταση. Απλά αποσυμπιέζουμε το αρχείο που μόλις κατεβάσαμε στον τοπικό δίσκο και κάνουμε διπλό κλικ επάνω στο εικονίδιο του Eclipse για να τρέξει. Επιπλέον θα χρειαστούμε όπως αναφέραμε και πιο πάνω το πακέτο android SDK μία εργαλειοθήκη ανάπτυξης συστήματος λογισμικού (Software Development Kit).

4.2.3 Εγκατάσταση Android SDK manager

Χρησιμοποιούμε το εξής link:

<http://developer.android.com/sdk/index.html#top>

και κατεβάζουμε το zip αρχείο.

Platform	Package	Size	SHA-1 Checksum
Windows	installer_r24.4.1-windows.exe (Recommended)	151659917 bytes	f9b59d72413649d31e633207e31f456443e7ea0b
	android-sdk_r24.4.1-windows.zip	199701062 bytes	66b6a6433053c152b22bf8cab19c0f3fef4eba49
Mac OS X	android-sdk_r24.4.1-macosx.zip	102781947 bytes	85a9cccb0b1f9e6f1f616335c5f07107553840cd
Linux	android-sdk_r24.4.1-linux.tgz	326412652 bytes	725bb360f0f7d04eaccff5a2d57abdd49061326d

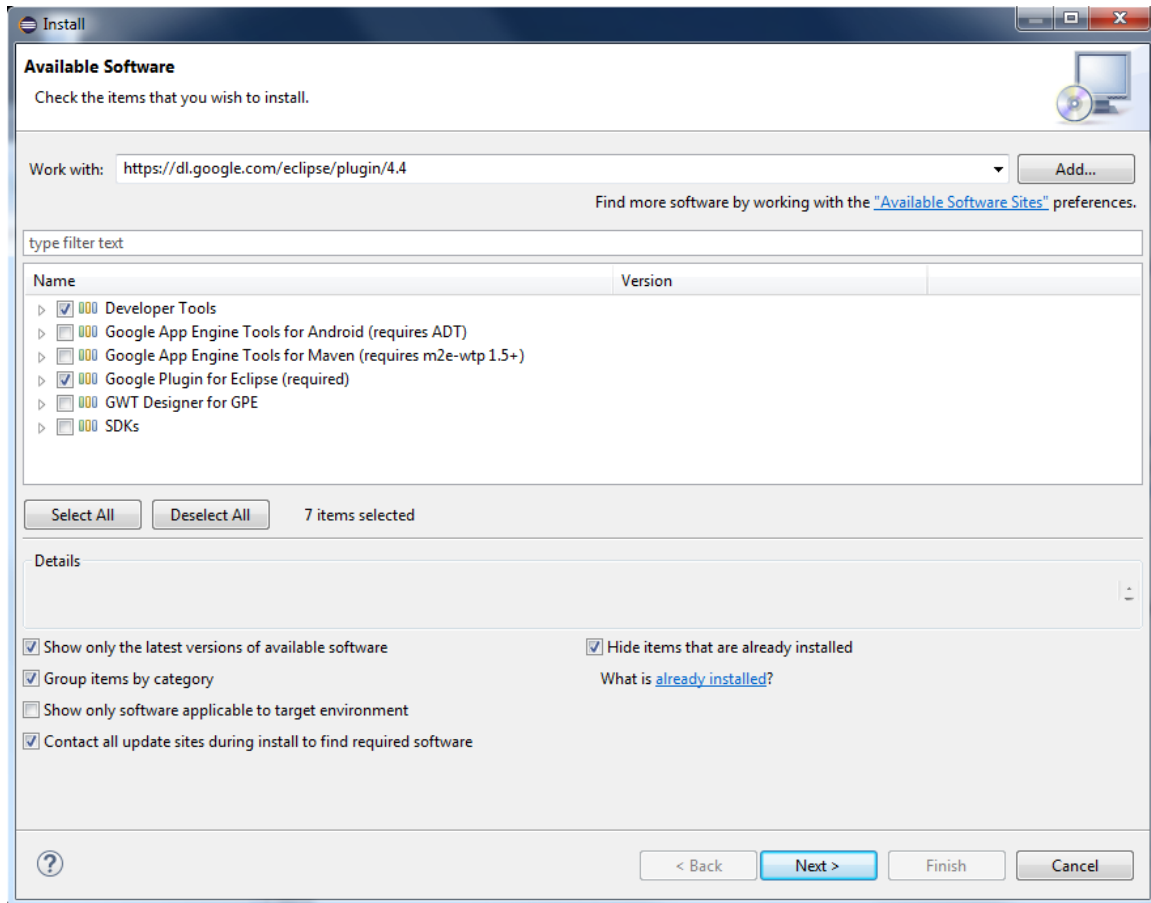
Εικόνα 19

Αφού το κατεβάσουμε κάνουμε ό,τι ακριβώς κάναμε και με το eclipse, δηλαδή αποσυμπίεση του φακέλου στον τοπικό μας δίσκο.

Στη συνέχεια τρέχουμε το εκτελέσιμο αρχείο του eclipse για να ανοίξει το πρόγραμμα. Στο menu επιλέγουμε:

Help > Install New Software και στο **Work with** χρησιμοποιούμε το εξής link:
<https://dl.google.com/eclipse/plugin/4.4>

και επιλέγουμε **Add**.



Εγκαθιστούμε τα Developer Tools και το Google Plugin for Eclipse.

Θα χρειαστεί να κάνουμε επανεκκίνηση το eclipse όπως μας ζητείται. Στη συνέχεια πρέπει να τρέξουμε το **SDK manager** και να εγκαταστήσουμε όλες τις διαθέσιμες πλατφόρμες που μπορούμε να χρησιμοποιήσουμε για τις android εφαρμογές μας.

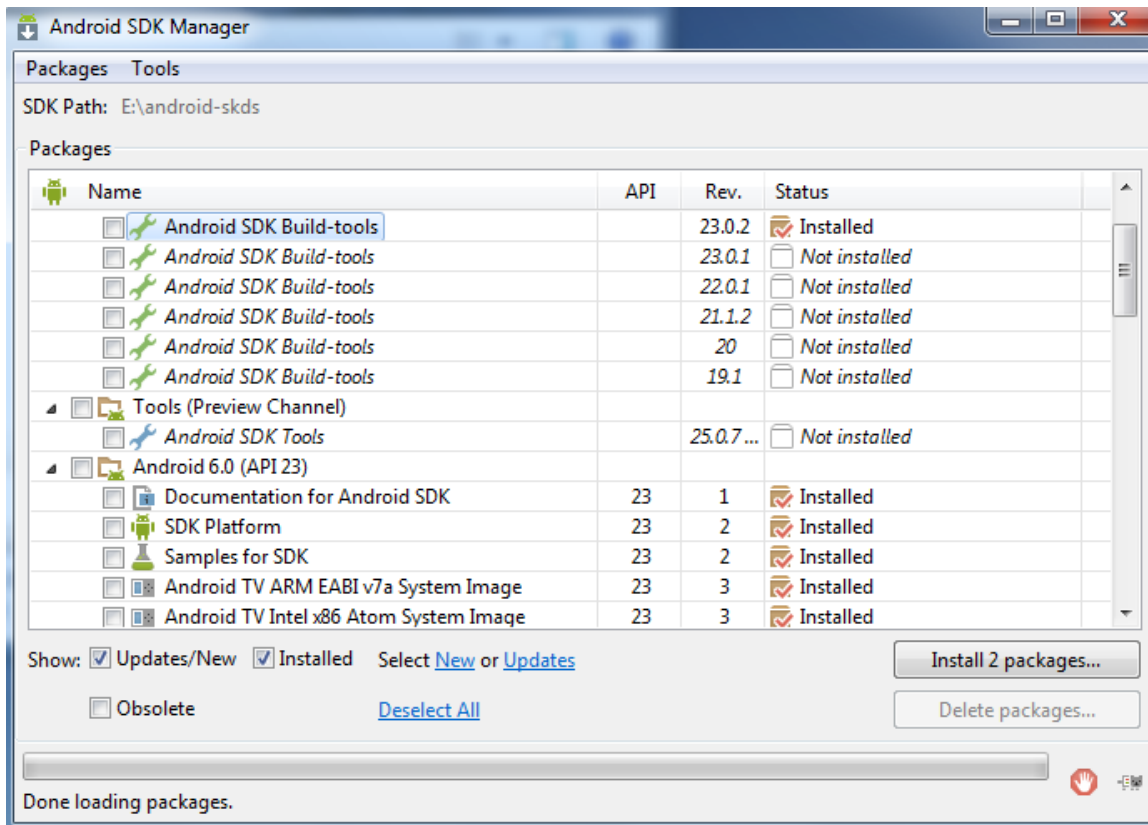
Στη συνέχεια θα μας εμφανιστεί μια λίστα με όλες τις διαθέσιμες προγραμματιστικές βιβλιοθήκες για όλες τις εκδόσεις του Android. Εμάς μας ενδιαφέρουν κυρίως 3 από αυτές τις επιλογές:

1) Τα Android SDK Platform-tools.

2) Το Android 2.2 API. Επιλέγουμε αυτή την έκδοση του Android για να είμαστε σίγουροι ότι οι εφαρμογές μας θα τρέχουν στις περισσότερες συσκευές. Μπορούμε όμως να επιλέξουμε να προγραμματίσουμε σε πιο τελευταία έκδοση API.

3) Το Google USB Driver package το οποίο μας επιτρέπει να μεταφέρουμε τις εφαρμογές μας στην κινητή συσκευή μας με την χρήση της USB θύρας του υπολογιστή.

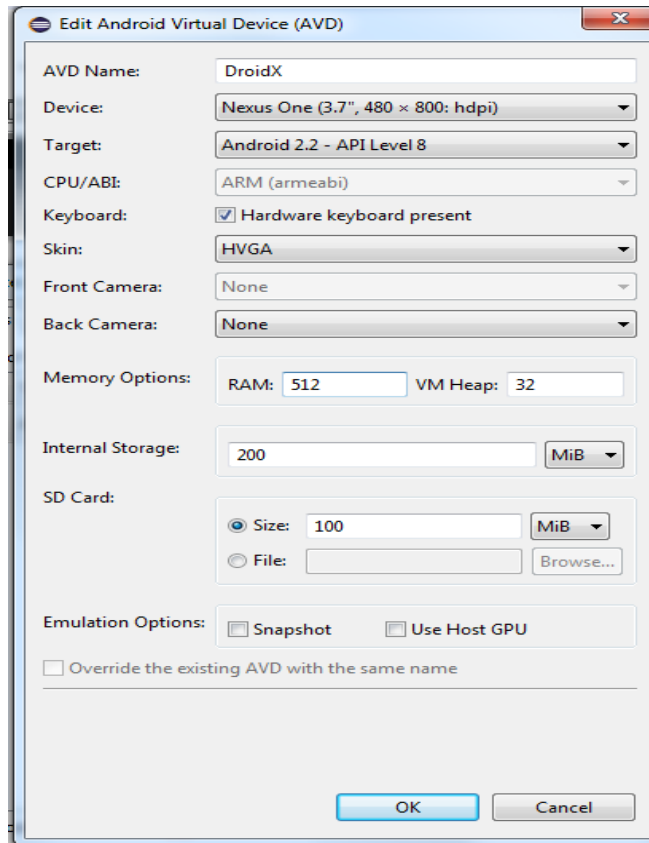
Στο παράθυρο που μας εμφανίζεται επιλέγουμε Accept all και install.



Εικόνα 20

4.3 Δημιουργία και διαμόρφωση Android Virtual Device

Στο μενού επιλέγουμε **Window>Android Virtual Device Manager** και επιλέγουμε **create**. Χρησιμοποιήσαμε τις ρυθμίσεις που φαίνονται στην εικόνα.

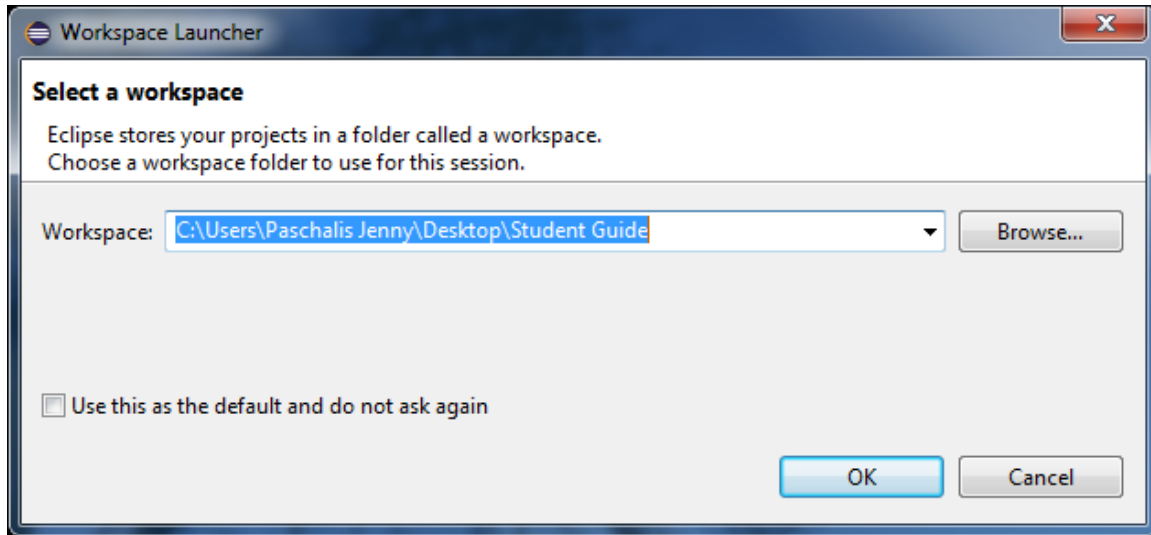


Εικόνα 21

ΚΕΦΑΛΑΙΟ 5: Ανάλυση & υλοποίηση εφαρμογής

5.1 Δημιουργία Android application project

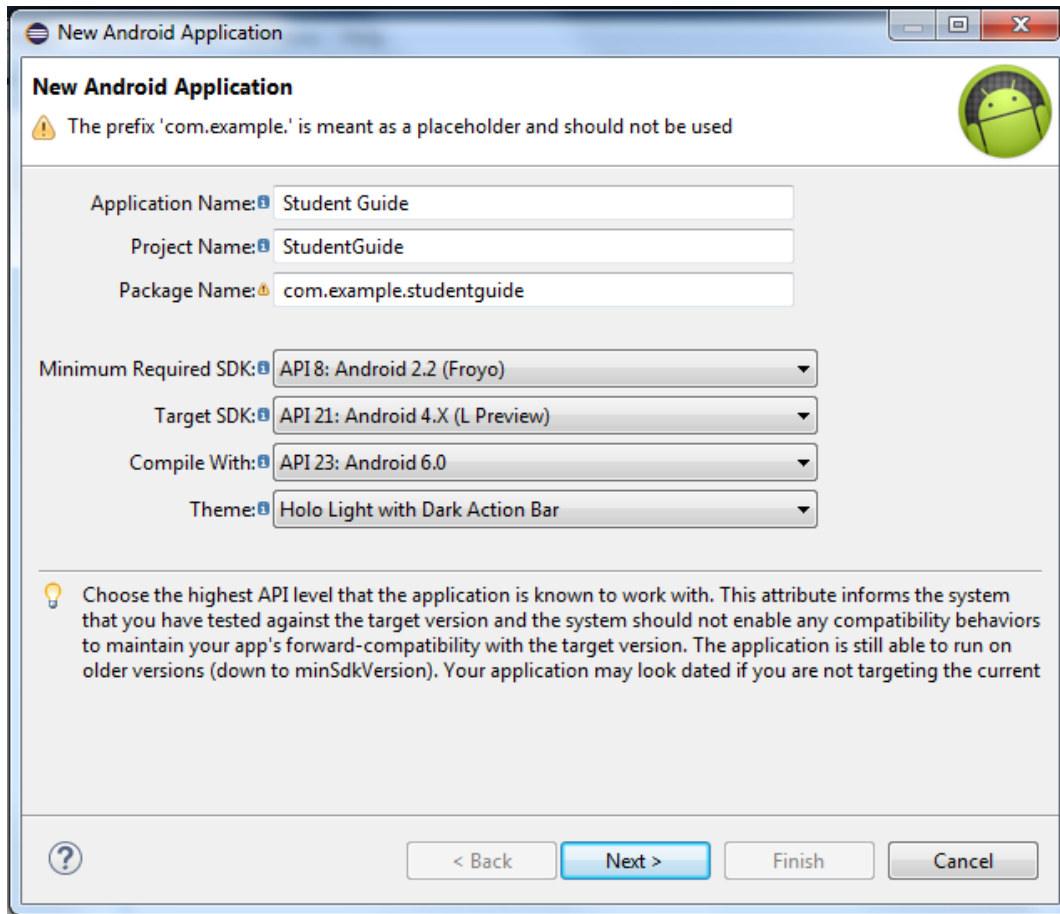
Για να δημιουργήσουμε ένα νέο project πατάμε αρχικά διπλό κλικ στο εικονίδιο του eclipse. Μας εμφανίζεται το εξής μήνυμα στο οποίο και μας ζητείται να ορίσουμε το workspace που θα δουλέψουμε.



Εικόνα 22

Εμείς έχουμε φτιάξει ένα φάκελο στην επιφάνεια εργασίας με το όνομα **Student Guide**. Για την δημιουργία νέου project επιλέγουμε **File> New>Android Application Project**.

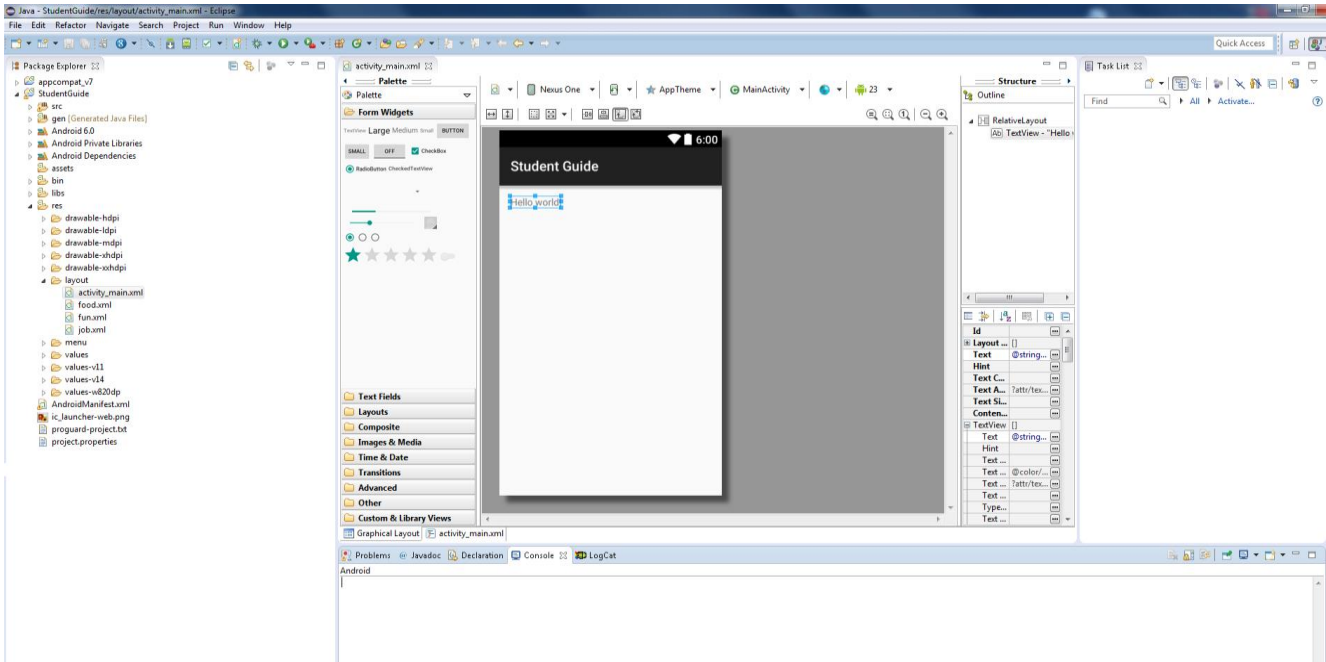
Στο παράθυρο που θα εμφανιστεί καταχωρούμε το όνομα που θα δώσουμε στην εφαρμογή, το όνομα του Project και το όνομα του πακέτου (package name) που χρησιμοποιείται για τη μοναδική αναγνώριση της εφαρμογής μας. Το Android χρησιμοποιεί το όνομα του πακέτου για να καθορίσει αν μία εφαρμογή έχει εγκατασταθεί ή όχι. Η γενική ονομασία είναι: com.companyname.applicationname. Προσοχή στο όνομα διότι θα πρέπει να είναι μοναδικό στην αγορά.



Εικόνα 23

Επιλέγουμε API 8: Android 2.2 (Froyo) για να είμαστε σίγουροι ότι οι εφαρμογές μας θα τρέχουν στις περισσότερες συσκευές.

Στις επιλογές που εμφανίζονται στη συνέχεια δεν κάνουμε κάποια αλλαγή. Κρατάμε το όνομα του activity ως έχει (MainActivity) και πατάμε finish. Αυτό είναι το περιβάλλον εργασίας του project μας (Student Guide) πάνω στο οποίο θα δουλέψουμε:



Εικόνα 24

5.1.1 Εισαγωγή εικόνων στο project

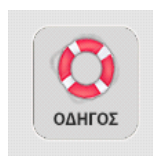
Για να μπορούμε γενικότερα να χρησιμοποιούμε εικόνες στην εφαρμογή μας θα πρέπει αρχικά να τις αποθηκεύσουμε στον κατάλληλο φάκελο μέσα στην εφαρμογή για να μπορεί να τις διαβάζει και ύστερα εμείς να τις επεξεργαζόμαστε και να τις διαμορφώνουμε. Για να γίνει λοιπόν αυτό θα πρέπει σαν πρώτο βήμα να αποθηκεύσουμε όλες τις εικόνες που θα χρειαστούμε για την εφαρμογή στον φάκελο **drawable-mdpi**, σε μορφή .png, .jpg ή .gif. Ο φάκελος αυτός έχει δημιουργηθεί αυτόματα με την δημιουργία ενός νέου android application project. Στην δική μας εφαρμογή τα αρχεία εικόνων που θα χρησιμοποιήσουμε για κουμπιά έχουν μορφή .png και μέγεθος 80x80 px. Πηγαίνω στο workspace, από εκεί στο φάκελο που αποθηκεύτηκε το project μας, και στη συνέχεια **res** και **drawable-mdpi**. Όπως παρατηρούμε υπάρχουν και τα αντίστοιχα **drawable-hdpi**, **drawable-ldpi**, **drawable-xhdpi** για εικόνες υψηλής ή χαμηλής ανάλυσης. Εμείς επιλέξαμε το ενδιάμεσο για να είμαστε καλυμμένοι.



Εικόνα 25

5.1.2 Εισαγωγή icon

Αφού λοιπόν έχουμε δημιουργήσει στο Photoshop το λογότυπο της εφαρμογής, μας το αποθηκεύουμε σε μορφή PNG και διαστάσεις 48x48 px στον φάκελο που προαναφέραμε. Το icon μας είναι έτοιμο και μοιάζει κάπως έτσι:



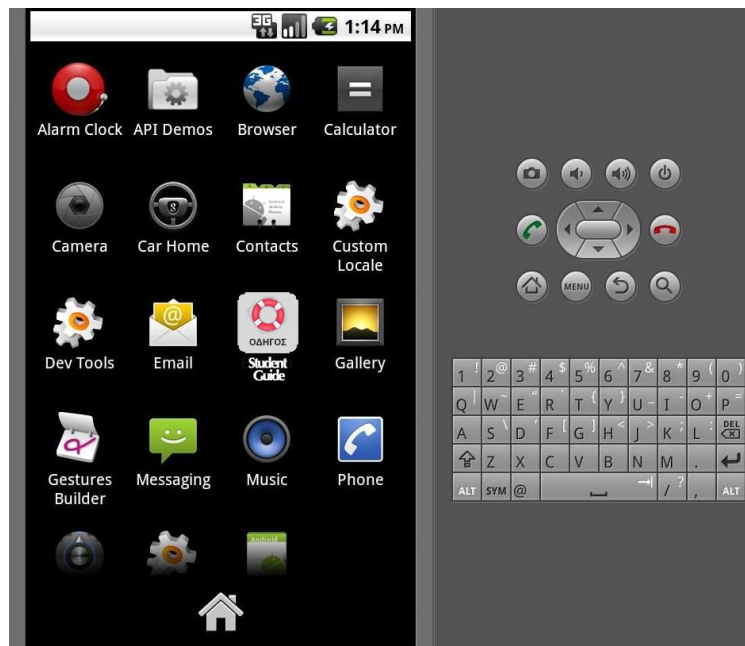
Για να εισάγουμε το icon στην εφαρμογή ώστε να εμφανίζεται στο μενού του κινητού μαζί με τα υπόλοιπα εικονίδια των άλλων εφαρμογών θα πρέπει να πάμε στο **AndroidManifest.xml** (αριστερά στο workspace) και στο android icon να γράψουμε: "**@drawable/icon**".

```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
```

Εικόνα 26

Όπου "*ic_launcher*" είναι το όνομα που έχουμε δώσει στην εικόνα που χρησιμοποιούμε για λογότυπο.

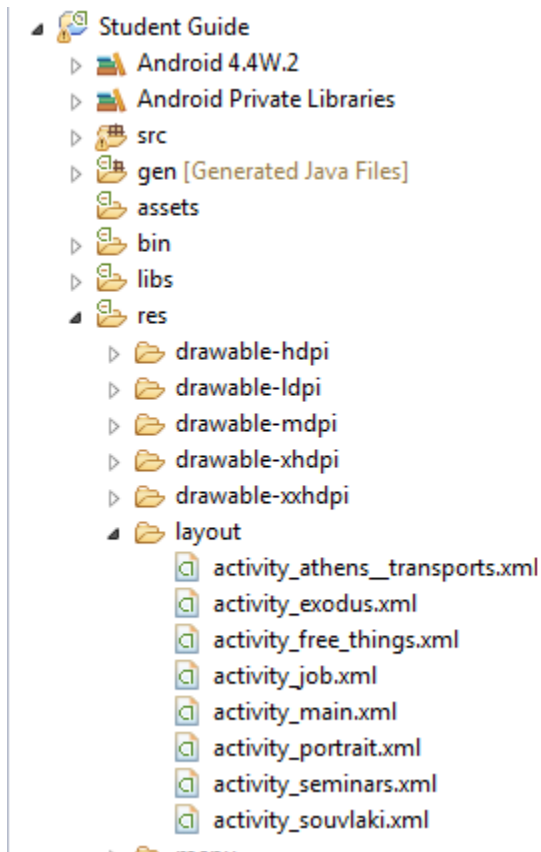
Κάνουμε δεξί κλικ στο project **Student Guide**>**Run as**>**Android Application** και το αποτέλεσμα είναι κάπως έτσι:



5.1.3 Δημιουργία αρχείου xml

Εκτός από το **activity_main.xml** που δημιουργείται αυτόματα, θα χρειαστεί να δημιουργήσουμε 6 επιπλέον xml αρχεία, ένα για το κάθε κουμπί, καθώς και τα αντίστοιχα java αρχεία τους για να γίνουν λειτουργικά. Για τα xml αρχεία επιλέγουμε: **res>layout>new>Android xml file** και επιλέγουμε τον τύπο του root element (LinearLayout). Πατάμε Finish για να δημιουργηθεί το νέο μας layout. Επιλέξαμε το LinearLayout και σαν root γιατί είναι πιο εύκολο στην διαχείριση και ενδείκνυται να

χρησιμοποιείτε ως root στα περισσότερα layouts. Τέλος δίνουμε το αντίστοιχο όνομα για το κάθε ένα από αυτά:



Εικόνα 27

5.2 Κεντρική οθόνη - Main.xml

(Κώδικας XML του Layout της κεντρικής οθόνης)

Στο κεντρικό λοιπόν xml **activity_main.xml** θα πρέπει αρχικά να ορίσουμε το background (φόντο) της εφαρμογής μας. Σώζουμε μία εικόνα σε μορφή .png με το όνομα **background** μέσα στον φάκελο **drawable-mdpi** (**ΠΡΟΣΟΧΗ ΟΧΙ ΜΕ ΚΕΦΑΛΑΙΑ**). Στην συνέχεια καλούμε την εικόνα μας μέσα από τον εξής κώδικα: `android:background="@drawable/background"` που δηλώνουμε μέσα στο RelativeLayout που εμφανίζεται αυτόματα με την δημιουργία του xml μας. Διαγράφουμε όλο TextView που είναι το παράδειγμα "Hello_world" που εμφανίζει το

eclipse και θα εισάγουμε το background όπως αναφέραμε. Το αποτέλεσμα θα είναι το εξής:



Εικόνα 28

Στην συνέχεια θα δηλώσουμε μέσα στο Layout τα κουμπιά που θα εμφανίζονται στην αρχική σελίδα καθώς και όλες τις παραμέτρους: το όνομά, το μέγεθος και τη διάταξή τους στο χώρο. Για να τοποθετήσουμε δηλαδή το κουμπί που αντιπροσωπεύει το “**All Free**” στο πάνω δεξί άκρο και το κείμενο ακριβώς από κάτω χρησιμοποιήσαμε τον εξής κώδικα:

```
<ImageButton
```

```
    android:id="@+id/imageButtonFree"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_marginRight="15dp"
    android:layout_marginTop="126dp"
    android:background="@null"
    android:src="@drawable/free" />
<TextView
    android:id="@+id/tv_free"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/imageButtonSouvlaki"
    android:layout_alignRight="@+id/imageButtonFree"
    android:gravity="center"
    android:text="All Free"
    android:textColor="@color/black" />
```

Ας δούμε αναλυτικά τι κάνει κάθε μια από αυτές τις εντολές:

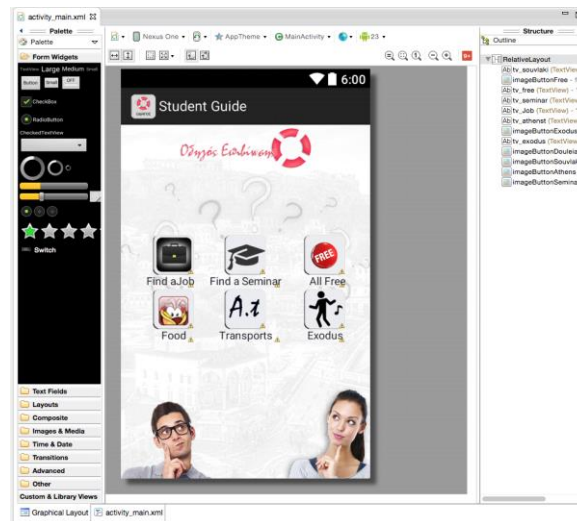
Με την ετικέτα `<ImageButton` δηλώνουμε ότι θέλουμε να εισάγουμε κουμπί στην εφαρμογή σε μορφή εικόνας. Με την εντολή `android:id="@+id/imageButtonFree"` προσδιορίζουμε μοναδικά την ταυτότητα του image button μας και ταυτόχρονα δηλώνουμε ότι πρέπει να δημιουργηθεί ένα νέο resource name μέσα στο R.java αρχείο. Με την εντολή `android:layout_width="wrap_content"` καθορίζουμε τι ποσοστό από το διατιθέμενο πλάτος της οθόνης θα καταλαμβάνει το Image Button. Σε αυτή την περίπτωση θέλουμε να καταλαμβάνει μόνο το περιεχόμενο του κουμπιού κάτι που ρυθμίζεται από το "wrap_content". Σε περίπτωση που θα θέλαμε να καταλαμβάνει όλη την οθόνη θα χρησιμοποιούσαμε το "fill_parent". Αντίστοιχα με το `android:layout_height="wrap_content"`. Με τις ακόλουθες εντολές: `android:layout_alignParentRight="true",android:layout_marginRight="15dp",android:layout_marginTop="126dp"` δίνουμε τις κατάλληλες παραμέτρους ώστε να τοποθετήσουμε το κουμπί μας στην πάνω δεξιά γωνία της οθόνης, σε απόσταση 126

dp από την κορυφή και 15 dp από το αριστερό περιθώριο. Με την εντολή `android:background="@null"` ουσιαστικά απομακρύνουμε το background του πρότυπου κουμπιού που έχει το eclipse. Τέλος με την εντολή `android:src="@drawable/free"` καλούμε την εικόνα *free* από τα resources η οποία θα πρέπει να έχει το ίδιο ακριβώς όνομα με το αρχείο μέσα στο φάκελο **drawable-mdpi**.

Όσον αφορά το κείμενο προκειμένου να το εισάγουμε στην εφαρμογή χρησιμοποιούμε την ετικέτα `<TextView`.

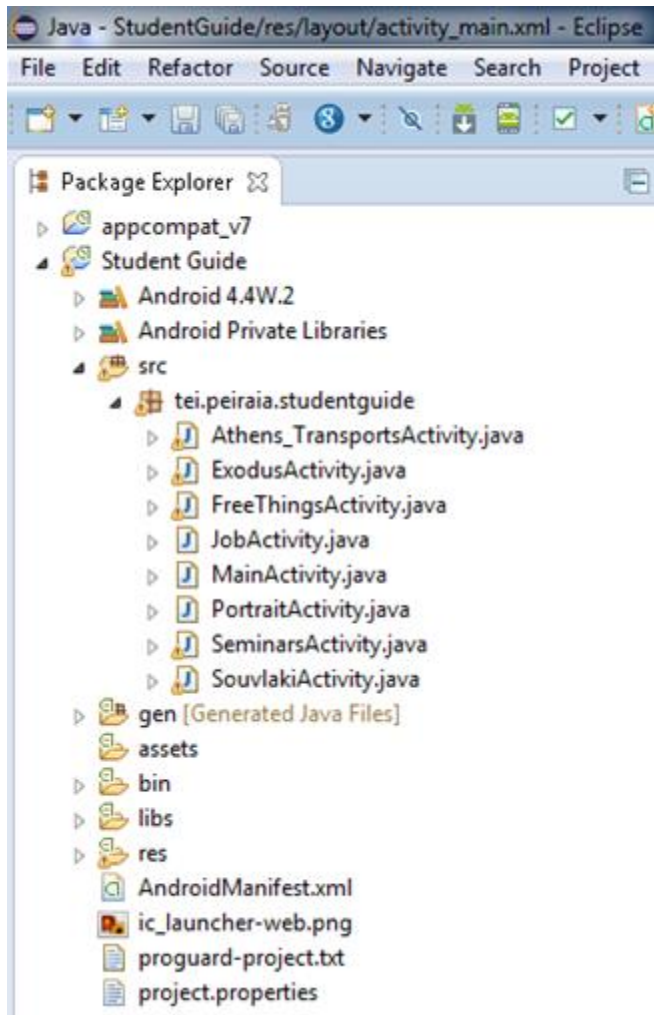
Με την εντολή `android:layout_above="@+id/imageButtonSouvlaki"` στοιχίζουμε το κείμενο που θέλουμε σε σχέση με το Image Button μας. Με το `android:gravity="center"` το στοιχίζουμε στο κέντρο στο κάτω μέρος του κουμπιού. Με το `android:text="All Free"` θέτουμε το κείμενο που θα περιλαμβάνει το TextView. Και τέλος με το `android:textColor="@color/black"` ορίζουμε το χρώμα του κειμένου σε μαύρο.

Για να διατάξουμε και τα υπόλοιπα κουμπιά και τα κείμενα στην main οθόνη όπως φαίνεται στην παρακάτω εικόνα χρησιμοποιήσαμε τον παρακάτω κώδικα με την ίδια ακριβώς λογική:



Εικόνα 29

5.2.1 Αρχεία Java



Εικόνα 30

Εκτός από το **MainActivity.java** που δημιουργείται αυτόματα, θα χρειαστεί να δημιουργήσουμε 7 επιπλέον **java** αρχεία, για να τα ενώσουμε με τα image buttons, ώστε να γίνουν λειτουργικά. Για τη δημιουργία java αρχείων (activities) επιλέγουμε: **crs>new>other>Android activity** και δίνουμε το αντίστοιχο όνομα για το κάθε ένα από αυτά:

5.2.2 MainActivity.java

Ας δούμε αναλυτικά τι κάνουν ορισμένες από τις βασικές εντολές που χρησιμοποιήσαμε:

```

package tei.peiraia.studentguide;

import android.content.Intent;

public class MainActivity extends PortraitActivity {

    private final OnClickListener imgButtonListener = new OnClickListener() {
        public void onClick(final View v) {
            final int id = v.getId();
            if (id == R.id.imageButtonDouleia) {
                final Intent mainToJob= new Intent(MainActivity.this,
                    JobActivity.class);
                MainActivity.this.startActivity(mainToJob);
            } else if (id == R.id.imageButtonSeminars) {
                final Intent mainToSeminars = new Intent(MainActivity.this,SeminarsActivity.class);
                MainActivity.this.startActivity(mainToSeminars);
            } else if (id == R.id.imageButtonAthens) {
                final Intent mainToAthensTrans = new Intent(MainActivity.this,
                    Athens_TransportsActivity.class);
                MainActivity.this.startActivity(mainToAthensTrans);
            } else if (id == R.id.imageButtonSouvlaki) {
                final Intent mainToSouvlaki = new Intent(MainActivity.this,
                    SouvlakiActivity.class);
                MainActivity.this.startActivity(mainToSouvlaki);
            } else if (id == R.id.imageButtonFree) {
                final Intent mainToFree = new Intent(MainActivity.this,
                    FreeThingsActivity.class);
                MainActivity.this.startActivity(mainToFree);
            } else if (id == R.id.imageButtonExodus) {
                final Intent mainToExodus = new Intent(MainActivity.this,
                    ExodusActivity.class);
                MainActivity.this.startActivity(mainToExodus);
            }
        }
    };
}

```

Εικόνα 31

Το package name δηλώνει το μοναδικό όνομα που έχουμε δώσει στην εφαρμογή μας. **package** tei.peiraia.studentguide;. Η **public class** MainActivity **extends** PortraitActivity {}, είναι η κύρια κλάση μας όπως έχει δημιουργηθεί αυτόματα και η οποία περιέχει τη συνάρτηση **onCreate()**. Η συνάρτηση αυτή όπως φαίνεται καλείται με την δημιουργία της κλάσης και αφού αυτή αποτελεί την αρχική μας κλάση καλείται στην αρχή του προγράμματος. Στην δική μας περίπτωση η κύρια κλάση καλεί το **PortraitActivity**, το οποίο είναι στην ουσία ένα νέο activity που δημιουργήσαμε, διότι κάποια σημεία του κώδικα επαναλαμβάνονταν μέσα στα διάφορα activities, και το portrait περιλαμβάνει τα σημεία αυτά, ώστε να κληρονομούνται από τις αρχικές μας activities. Θα εξηγήσουμε εκτενέστερα την λειτουργία του portrait παρακάτω. Στην συνέχεια υλοποιούμε την **onClick()** η οποία καλείται όταν πατήσει κλικ ο χρήστης. Έπειτα με την χρήση του switch ελέγχουμε την τιμή (**v.getId()**) αν ισούται με κάθε ένα

από τα **if** και για κάθε κλικ παίρνουμε το **View** του με βάση το **id** και ανάλογα με αυτό, εκτελούμε τι θέλουμε να κάνουμε. Για να αποφύγουμε την επανάληψη του κώδικα και να βελτιώσουμε την αναγνωσιμότητα του δημιουργήσαμε ένα κοινό **clickListener** για όλα τα **image Buttons**.

Γενικά στο Android ο προγραμματιστής δεν έχει έλεγχο της κατάστασης του συστήματος και του **activity** που εκτελείται. Η κατάσταση ελέγχεται από το σύστημα. Γι' αυτό το λόγο όταν γίνεται αλλαγή κατάστασης καλούνται κάποιες συναρτήσεις οι οποίες πρέπει να υπερφορτωθούν αν θέλουμε να εκτελεστεί κάτι διαφορετικό. Για κάθε ένα πλήκτρο που έχει επιλεγεί δημιουργούμε ένα αντικείμενο τύπου **Intent** που το αρχικοποιούμε στην κλάση που θέλουμε να το προωθήσουμε και ξεκινάμε τη νέα ενέργεια που θέλουμε με την **startActivity()**; **Intents** είναι μηχανισμοί που περιγράφουν μία συγκεκριμένη ενέργεια. Γενικά στο προγραμματισμό με Android χρησιμοποιούνται πολύ συχνά. Όποτε χρειάζεται μια νέα λειτουργία χρησιμοποιούμε **intent** και μπορούμε να υπερφορτώσουμε κάποια ήδη υπάρχοντα όπως και στα **activities**. Για παράδειγμα στην περίπτωση: **if** (**id** == **R.id.imageButtonDouleia**) καλούμε τον συγκεκριμένο μηχανισμό **intent**, που στην περίπτωση μας κάνοντας κλικ στο **Find a Job** σε μεταφέρει στην **JobActivity**. Σε περίπτωση που το **id** δεν είναι αυτό που ζητάει, εκτελείται το επόμενο **else if** μέχρι να βρει το **id** που θέλει για να τρέξει.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final ImageButton imageButtonDouleia = (ImageButton) findViewById(R.id.imageButtonDouleia);
    imageButtonDouleia.setOnClickListener(imgButtonListener);

    final ImageButton imageButtonSeminars = (ImageButton) findViewById(R.id.imageButtonSeminars);
    imageButtonSeminars.setOnClickListener(imgButtonListener);

    final ImageButton imageButtonFree = (ImageButton) findViewById(R.id.imageButtonFree);
    imageButtonFree.setOnClickListener(imgButtonListener);

    final ImageButton imageButtonExodus = (ImageButton) findViewById(R.id.imageButtonExodus);
    imageButtonExodus.setOnClickListener(imgButtonListener);

    final ImageButton imageButtonAthens = (ImageButton) findViewById(R.id.imageButtonAthens);
    imageButtonAthens.setOnClickListener(imgButtonListener);

    final ImageButton imageButtonSouvlaki = (ImageButton) findViewById(R.id.imageButtonSouvlaki);
    imageButtonSouvlaki.setOnClickListener(imgButtonListener);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}

```

Εικόνα 32

Το @Override είναι ένα annotation της Java, δηλαδή ένα επιπλέον στοιχείο που εισάγεται από την έκδοση J2SE 5 και που επιτρέπει στους προγραμματιστές να ενσωματώνουν πρόσθετες πληροφορίες που ονομάζονται **metadata** σε ένα αρχείο πηγαίου κώδικα της Java. Συγκεκριμένα το method overriding είναι ένα χαρακτηριστικό του αντικειμενοστραφούς προγραμματισμού που επιτρέπει σε μία υπερκλάση (superclass, parent class) να κληροδοτήσει σε μία και πάνω υποκλάσεις (subclasses) μια συγκεκριμένη εφαρμογή της μεθόδου αυτής. Οι μέθοδοι αυτοί θα έχουν τα ίδια χαρακτηριστικά: όνομα, υπογραφή, παραμέτρους. Μπορούμε να εκμεταλλευτούμε

αυτό το χαρακτηριστικό και στον μεταγλωττιστικό έλεγχο και επιπλέον καθιστά τον κώδικα πιο ευανάγνωστο και κατανοητό.

Το **public** δηλώνει ότι η μέθοδος είναι ορατή και μπορεί να καλεστεί από οπουδήποτε ακόμα και από κλάσεις άλλων packages και το **void** ότι η μέθοδος δεν επιστρέφει μεταβλητή. Εάν επέστρεφε θα έπρεπε να γράψουμε int αντί για void. Ο λόγος που χρησιμοποιούμε την **onCreate(Bundle savedInstanceState)** είναι γιατί στην ουσία εάν δεν την χρησιμοποιήσουμε δεν κάνει την μεταγλώττιση ο compiler και δεν τρέχει ο κώδικας. Στην συνέχεια καλούμε τη **setContentView()**, η οποία θέτει το layout της activity, περιλαμβάνει κουμπιά, **textviews** κλπ, το περιεχόμενο δηλαδή που ορίσαμε στο activity_main.xml και θα εμφανιστεί. Τέλος για να γίνουν λειτουργικά τα κουμπιά που χρησιμοποιήσαμε στην main.xml δηλώνουμε το κάθε ένα από αυτά στην main.java όπως φαίνεται στον παραπάνω κώδικα.

5.3 Οθόνη home - Portrait.xml

Το portrait.xml περιλαμβάνει μόνο το home button, το οποίο εισάγουμε με τον ακόλουθο κώδικα:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="tei.peiraia.studentguide.PortraitActivity" >

    <ImageButton
        android:id="@+id/imageButtonHome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:background="@null"
        android:cropToPadding="true"
        android:src="@drawable/home" />

</RelativeLayout>
```

Εικόνα 33

5.3.1 PortraitActivity.java

```
package tei.peiraia.studentguide;

import android.app.Activity;

public class PortraitActivity extends Activity implements OnClickListener{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
    }
}
```

Εικόνα 34

Δηλώνουμε με τον παραπάνω κώδικα ότι ο προσανατολισμός της οθόνης θα είναι **portrait** σε όλα τα java αρχεία. Αρχικά είχαμε δηλώσει τον ίδιο κώδικα σε κάθε java αρχείο μέσα στο onCreate για να μην αλλάζει ο προσανατολισμός, στην συνέχεια εφόσον ο κώδικας επαναλαμβανόταν τον σβήσαμε και τον βάλουμε στην superclass portrait που δημιουργήσαμε.

```
@Override
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
}
```

Εικόνα 35

Σε όλα τα .java υπήρχε ο παραπάνω κώδικας, τον σβήσαμε και τον βάλουμε στην superclass portrait που δημιουργήσαμε, η οποία περιλαμβάνει τα σημεία που επαναλαμβάνονται σε όλα τα activities. Το onConfigurationChanged δηλώνει ότι οποιαδήποτε αλλαγή και να γίνεται παραμένει σε portrait.

```
public void onClick(View v) {
    if (v.getId()==R.id.imageButtonHome)
        finish();
}
```

Εικόνα 36

Επιπλέον παρατηρούμε ότι ο **onClickListener** που αντιστοιχεί στο home button είναι κοινός σε όλα τα activities οπότε το μεταφέρουμε στο superclass **portrait**. Τέλος το finish σκοτώνει το τρέχον activity.

5.3.2 Manifest.xml

Στην ουσία στο manifest δηλώνονται αυτόματα όλα τα activities που δημιουργούμε. Εμείς απλά προσθέσαμε αυτές τις δύο γραμμές κώδικα `android:configChanges="keyboardHidden|orientation"`, `android:screenOrientation="portrait"` > σε κάθε activity για να κρύψουμε το πληκτρολόγιο του android που βγαίνει αυτόματα και να θέσουμε το screenOrientation σε **portrait** χωρίς να αλλάζει κάθε φορά που θα γυρνάς την mobile συσκευή σου.

5.3.3 Τα υπόλοιπα xml αρχεία

Ο κώδικας που χρησιμοποιήσαμε είναι ο ακόλουθος:


```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="tei.peiraia.studentguide.JobActivity" >

    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/include1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="25dp"
        android:src="@drawable/kariera" />

    <ImageButton
        android:id="@+id/imageButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:src="@drawable/skywalker" />

    <include layout="@layout/activity_portrait"/>

</RelativeLayout>
```

Με το:

`<include layout="@layout/portrait_activity" />` καλούμε την `portrait_activity` μέσω της μοναδικής ταυτότητας της. Στην συνέχεια εισάγουμε τα κουμπιά με την εντολή `button` και δίνουμε τις απαραίτητες παραμέτρους για το κάθε κουμπί σχετικά με το όνομα, το μέγεθος και την στοίχιση τους.

Τα ίδια ισχύουν και για τα υπόλοιπα xml αρχεία. Το αποτέλεσμα θα έχει ως εξής:



Εικόνα 37

5.4 Java Αρχεία – κώδικας

```

package tei.peiraia.studentguide;

import android.content.Intent;

public class JobActivity extends PortraitActivity implements OnClickListener {
    private OnClickListener imgButtonListener = new OnClickListener() {

        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.imageButton1:
                    Intent Kariera = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.kariera.gr/"));
                    startActivity(Kariera);
                    break;
                case R.id.imageButton2:
                    Intent Skywalker = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.skywalker.gr/"));
                    startActivity(Skywalker);
                    break;
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_job);

        ImageButton imageButtonHome = (ImageButton) findViewById(R.id.imageButtonHome);
        imageButtonHome.setOnClickListener(this);

        ImageButton imageButton1 = (ImageButton) findViewById(R.id.imageButton1);
        imageButton1.setOnClickListener(imgButtonListener);

        ImageButton imageButton2 = (ImageButton) findViewById(R.id.imageButton2);
        imageButton2.setOnClickListener(imgButtonListener);
    }
}

```

Εικόνα 38

Η λογική του παραπάνω κώδικα λειτουργεί με τον ίδιο τρόπο όπως εξηγήσαμε και στο **mainActivity.java**, υλοποιούμε δηλαδή τον **onclickListener**, με το **switch** ελέγχουμε τα cases μας βάσει του id. Σ' αυτήν την περίπτωση η μόνη διαφορά είναι ότι ο **Intent** που χρησιμοποιήσαμε δεν κάνει την ίδια ενέργεια, να εμφανίζει δηλαδή μια activity.xml του project. Αντιθέτως χρησιμοποιεί ένα url που σε μεταφέρει σε μία ιστοσελίδα:

```

Intent Kariera = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://www.kariera.gr/"));

```

Τέλος, όπως και στο **mainActivity.java** για να γίνουν λειτουργικά τα κουμπιά που χρησιμοποιήσαμε στα xml δηλώνουμε το κάθε ένα στο αντίστοιχο **.java** αρχείο

όπως φαίνεται στον παραπάνω κώδικα. Τα ίδια ισχύουν και για τα υπόλοιπα java αρχεία.

Βιβλιογραφία

- [1] Java μέρος I: Εισαγωγή στον αντικειμενοστραφή προγραμματισμό, Χρήστος Δ. Κυταγιάς MSc, Κώστα Δ. Κυταγιάς MSc, Δρ. Δημήτριος Χ. Κυταγιάς, Δρ. Γεώργιος Πρεζεράκος, εκδόσεις Σύγχρονη Εκδοτική.
- [2] Android για προγραμματιστές, συλλογικό έργο Deitel Paul J., Deitel Harvey M., Deitel Abbey, Morgano Michael, εκδόσεις Γκιούρδας.
- [3] Android Apps with Eclipse, συγγραφέας Cinar Onur, εκδότης John Wiley & Sons Inc.
- [4] Beginning android programming, συγγραφέας Lee Wei-Meng, εκδότης APress.
- [5] Ανάλυση της αρχιτεκτονικής του Android.
<http://developer.android.com/index.html>
- [6] Ανάλυση των βασικών συστατικών μιας εφαρμογής του Android, και ο συσχετισμός τους με την αρχιτεκτονική του συστήματος.
<http://developer.android.com/index.html>
- [7] Πληροφορίες σχετικά με τη δομή και τη χρήση του του αρχείου AndroidManifest.xml
<http://developer.android.com/guide/topics/manifest/manifest-intro.html>
- [8] Η επίσημη ιστοσελίδα του Eclipse IDE, το οποίο χρησιμοποιήθηκε κατά την υλοποίηση του project και είναι και προτεινόμενο από την Google.
<http://www.eclipse.org/>
- [9] Πόροι και πληροφορίες για την εγκατάσταση της java.
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [10] Πόροι και πληροφορίες για την εγκατάσταση του Android SDK manager.
<http://developer.android.com/sdk/index.html#top>
- [11] Wikipedia, the free encyclopedia.
http://en.wikipedia.org/wiki/Main_Page