

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ



ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΣΥΣΚΕΥΕΣ ANDROID ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ IDE BASIC4ANDROID

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΡΗΓΑΚΗΣ ΗΡΑΚΛΗΣ

ΦΟΙΤΗΤΕΣ

ΓΟΥΓΟΥΛΗ ΜΑΡΙΑ

ΧΑΣΑΠΗ ΑΝΤΩΝΙΑ

Ευχαριστίες

Με την εκπόνηση της διπλωματικής μας εργασίας θα θέλαμε να ευχαριστήσουμε τον καθηγητή μας, κ. Ρηγάκη Ηρακλή για την πολύτιμη βοήθεια του και την υποστήριξη του.

Επίσης οφείλουμε ένα μεγάλο ευχαριστώ στους φίλους μας Φραγκίσκο και Μελίνα για την απίστευτη υπομονή που επιδείξανε όλο αυτό το χρονικό διάστημα, την υπερπολύτιμη βοήθεια τους, την αμέριστη συμπαράσταση τους και την εμπύχωση τους.

Περίληψη

Η ανάγκη του ανθρώπου για επικοινωνία οδήγησε στην ανάπτυξη των τηλεπικοινωνιακών υπηρεσιών με ένα από τα τελευταία επιτεύγματα της, την κινητή τηλεφωνία. Τα τελευταία χρόνια η κινητή τηλεφωνία εξελίχθηκε αρκετά, ώστε το κινητό να αποτελεί τόσο μέσο επικοινωνίας, όσο και μέσο συγκρότησης διάφορων εργασιών, ενημέρωσης και διασκέδασης.

Στην παρούσα πτυχιακή εργασία θα ασχοληθούμε με την ανάπτυξη μίας εφαρμογής για έξυπνα κινητά τηλέφωνα (smart phones) με λειτουργικό σύστημα Android. Η εφαρμογή αυτή έχει ως σκοπό την ενημέρωση του χρήστη για τις καιρικές συνθήκες που επικρατούν εκείνη την στιγμή σε κάποιες πόλεις της Ελλάδος, που έχουμε αρχικά ορίσει. Συγκεκριμένα, στην Θεσσαλονίκη, την Αθήνα, τα Χανιά και το Ηράκλειο. Επίσης, σε αυτήν την εργασία αναλύουμε ένα κομμάτι από την Basic4Android, το περιβάλλον με το οποίο υλοποιήσαμε την εφαρμογή μας.

Λέξεις – κλειδιά : Android, Basic4Android, IDE, εφαρμογή, καιρός, κινητή τηλεφωνία

Abstract

The human need for communication has led to the development of telecommunication services, and to their latest achievement, mobile telephony. In recent years, mobile telecommunications have evolved enough so that your mobile phone is both a means of communication, and means of setting up various tasks, information and entertainment.

This thesis will deal with the development of an application for smart mobile phones (smart phones) which use Android as an operating system. This application will inform the user about the weather conditions prevailing at that time in various cities of Greece, which we originally set, specifically, in Thessaloniki, Athens, Chania and Heraklion. Also, in this thesis, we analyse a piece of Basic4Android, the environment in which we developed our application.

Key-words : Android, Basic4Android, IDE, application, weather, mobile telephony

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	5
ΠΕΡΙΛΗΨΗ	6
ABSTRACT	6
ΠΕΡΙΕΧΟΜΕΝΑ	7
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ	10
1.1 Αντικείμενο της διπλωματικής	10
1.2 Οργάνωση κειμένου	11
ΚΕΦΑΛΑΙΟ 2 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΚΙΝΗΤΗ ΤΗΛΕΦΩΝΙΑ	13
2.1 Ιστορία κινητής τηλεφωνίας	13
2.2 Smart phones και Λειτουργικά συστήματα	14
2.2.1 iOS	15
2.2.2 Windows Phone	16
2.2.3 Android	17
2.3 Γιατί επιλέξαμε Android	17
ΚΕΦΑΛΑΙΟ 3 ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ ANDROID	19
3.1 Εισαγωγή	19
3.2 Ιστορικά στοιχεία	19
3.3 Εκδόσεις	20
3.4 Google Play	26
3.5 Αρχιτεκτονική του Android	27
3.5.1 Πυρήνας Linux	28
3.5.2 Εγγενής βιβλιοθήκες	28
3.5.3 Χρόνος εκτέλεσης	29

3.5.4 Πλαίσιο εφαρμογών	30
3.5.5 Εφαρμογές & Widgets	31
ΚΕΦΑΛΑΙΟ 4 ΠΡΟΓΡΑΜΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ BASIC4ANDROID	33
4.1 Γενικά	33
4.2 Ορισμός	33
4.3 Χαρακτηριστικά και δυνατότητες	34
ΚΕΦΑΛΑΙΟ 5 ΒΑΣΙΚΑ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	37
5.1 Γενικά	37
5.2 JAVA JDK	37
5.3 ANDROID SDK	39
5.4 Εγκατάσταση και διαμόρφωση Basic4Android	41
5.5 Android virtual device	43
5.6 Basic4Android – Bridge	47
ΚΕΦΑΛΑΙΟ 6 ΑΝΑΛΥΣΗ ΒΑΣΙΚΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ BASIC4ANDROID	50
6.1 Εξοικείωση με το IDE	50
6.1.1 Πρώτη γνωριμία	50
6.1.2 Προκαθορισμένος κώδικας – Default code	52
6.1.3 Σχεδιαστής – Designer	54
6.1.4 Παράδειγμα	57
6.2 GPS	66
6.3 HTTP – Υπηρεσίες Web	70
6.4 Camera	75
6.5 Android Serial Port (Bluetooth)	78

ΚΕΦΑΛΑΙΟ 7 ΑΝΑΛΥΣΗ ΕΦΑΡΜΟΓΗΣ ΚΑΙΡΟΥ	83
7.1 Εισαγωγή και περιγραφή εφαρμογής	83
7.2 Αρχεία εικόνων και αρχείο τιμών καιρού	84
7.3 Ανάλυση κώδικα	85
ΣΥΝΟΨΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	92
ΒΙΒΛΙΟΓΡΑΦΙΑ	93
ΠΑΡΑΡΤΗΜΑ	95

1

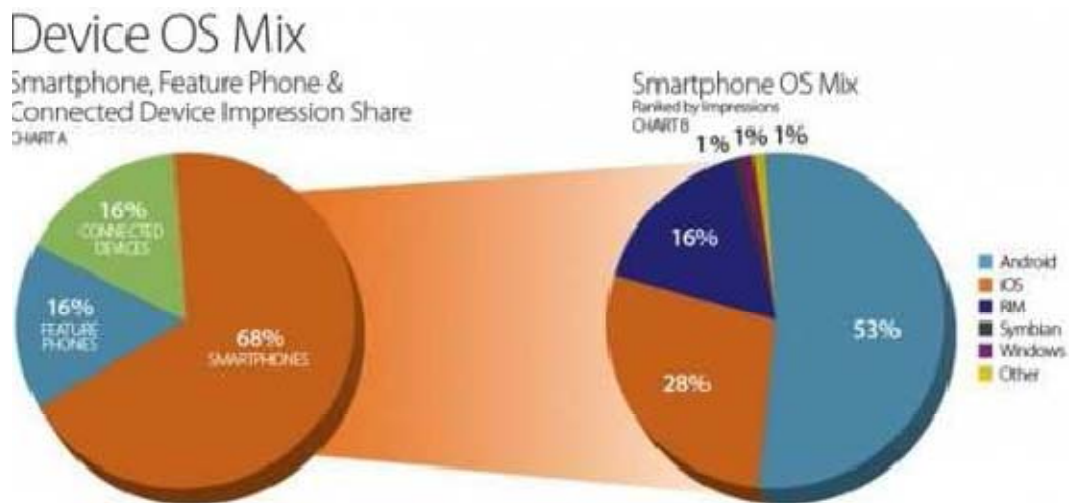
Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Είναι γεγονός ότι τα τελευταία χρόνια βιώνουμε την ραγδαία εξέλιξη των νέων τεχνολογιών που προσπαθούν να ανταπεξέλθουν όσο το δυνατόν καλύτερα στους γρήγορους ρυθμούς ζωής και στις σύγχρονες ανάγκες του κάθε ανθρώπου.

Τα έξυπνα κινητά τηλέφωνα, αναμφίβολα, είναι ένα από τα καλύτερα παραδείγματα αυτής της εξέλιξης. Περιλαμβάνουν ολοκληρωμένο λειτουργικό σύστημα, ισχυρό επεξεργαστή ικανό για εκτέλεση πολύπλοκων και χρονοβόρων υπολογισμών, δυνατότητα σύνδεσης στο διαδίκτυο είτε μέσω ασύρματων δικτύων (wifi), είτε δικτύων τρίτης και τέταρτης γενιάς (3G & 4G networks), κάμερες υψηλής ανάλυσης, συστήματα εντοπισμού θέσης (GPS), έναν αριθμό από αισθητήρες κ.α.

Τα πιο γνωστά λειτουργικά συστήματα κινητών τηλεφώνων είναι το Android της Google, το iOS της Apple και το Windows Phone της Microsoft. Την κυρίαρχη θέση στην αγορά αυτή τη στιγμή, κατέχει το Android και σύμφωνα με έρευνες που έχουν γίνει, θα επικρατήσει στην πρώτη θέση και στο μέλλον.



Στατιστικά πωλήσεων λειτουργικών συστημάτων smartphone

Στα πλαίσια της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μίας εφαρμογής, που προορίζεται για τα κινητά τηλέφωνα που διαθέτουν λειτουργικό σύστημα Android. Με την εφαρμογή αυτή, ο χρήστης θα μπορεί να ενημερώνεται για τις τελευταίες καιρικές συνθήκες σε κάθε πόλη. Σκοπός μας είναι η μελέτη, η κατανόηση και η ανάπτυξη κάποιων από τις παραμέτρους και τις δυνατότητες του εργαλείου ανάπτυξης εφαρμογών, Basic4Android.

1.2 Οργάνωση κειμένου

Η διπλωματική αυτή εργασία αναλύεται συνολικά σε 7 κεφάλαια.

Στο **1^ο κεφάλαιο** ασχολούμαστε με την εισαγωγή στο αντικείμενο που πραγματεύεται η εργασία. Εισάγουμε τον αναγνώστη στις νέες τεχνολογίες όσον αφορά τα κινητά τηλέφωνα.

Στο **2^ο κεφάλαιο** θα αναφερθούμε με λίγα λόγια στην ιστορία της κινητής τηλεφωνίας. Ποιος ο δημιουργός της πρώτης συσκευής κινητού τηλεφώνου και πως εξελίχθηκε έως σήμερα. Θα μιλήσουμε για τα έξυπνα κινητά τηλέφωνα και τις δυνατότητες τους και θα αναφερθούμε στα λειτουργικά συστήματά τους. Τέλος θα αναφέρουμε και τους λόγους που επιλέξαμε το λειτουργικό σύστημα Android.

Στο **3^ο κεφάλαιο** περιγράφεται το λειτουργικό σύστημα Android. Εισάγουμε τον αναγνώστη στην ιστορία του λειτουργικού συστήματος Android, τις εκδόσεις του, στο Google play αλλά και στην αρχιτεκτονική του συστήματος.

Στο **4ο κεφάλαιο** δίνουμε τον ορισμό στο περιβάλλον ανάπτυξης εφαρμογών Basic4Android και βλέπουμε τα χαρακτηριστικά του και τις δυνατότητες που έχει στην δημιουργία εφαρμογών.

Στο **5ο κεφάλαιο** θα δούμε τα βασικά εργαλεία ανάπτυξης που απαιτούνται για την ανάπτυξη εφαρμογών και πως πρέπει αυτά να εγκατασταθούν, ώστε να έχουμε το σωστό αποτέλεσμα.

Στο **6ο κεφάλαιο** θα αναλύσουμε μερικά βασικά χαρακτηριστικά του προγράμματος και θα δώσουμε παραδείγματα για την καλύτερη κατανόηση τους.

Στο **7ο κεφάλαιο** παρουσιάζεται αναλυτικά η αρχιτεκτονική της εφαρμογής και οι επιλογές που έχει ο χρήστης αλληλεπιδρώντας με αυτήν. Περιγράφεται κάθε διαφορετική οθόνη της εφαρμογής καθώς και ο τρόπος υλοποίησης των παραπάνω λειτουργιών στο προγραμματιστικό περιβάλλον.

Τέλος, περιγράφονται τα συμπεράσματα που προκύπτουν από την εκπόνηση της διπλωματικής εργασίας, καθώς και οι μελλοντικές εξελίξεις της.

Στο Παράρτημα της εργασίας, παρατίθεται αναλυτικά ο κώδικας της εφαρμογής που αναπτύχθηκε.

Η Βιβλιογραφία που χρησιμοποιήθηκε για την συγγραφή του παρόντος κειμένου καθώς και όλες οι ιστοσελίδες που βοήθησαν στην ανάπτυξη της εφαρμογής, βρίσκονται στο τέλος της εργασίας μας.

2

Εισαγωγή στην κινητή τηλεφωνία

2.1 Ιστορία κινητής τηλεφωνίας

Η περιπέτεια της κινητής τηλεφωνίας ξεκίνησε μετά τον 2^ο Παγκόσμιο Πόλεμο, με τις πρώτες προσπάθειες των Σουηδών, Φιλανδών και Αμερικανών.

Όμως ως ληξιαρχική πράξη γέννησης της θεωρείται η 3^η Απριλίου 1973. Ένα μουντό ανοιξιάτικο πρωινό ο Dr Martin Cooper της Motorola, περπατώντας σ' ένα δρόμο της Νέας Υόρκης γνώριζε ότι γράφει ιστορία. Στα χέρια του κρατούσε μία συσκευή που έμοιαζε με φορητό ασύρματο τηλέφωνο, το οποίο είχε ύψος 25 εκατοστά και βάρος 900 γραμμάρια. Ήταν το πρώτο σύγχρονο κινητό τηλέφωνο, το Motorola Dynatac. Σχημάτισε τον αριθμό του βασικού ανταγωνιστή του, Joel Engel, που δούλευε για λογαριασμό της Bell Labs, « Γεια σου Τζο, σου μιλάω από ένα αληθινό κινητό τηλέφωνο», του είπε.



Ο Μ. Cooper με την πρώτη συσκευή κινητού

Η Bell πήρε την ρεβάνς το 1978 κατασκευάζοντας το πρώτο δοκιμαστικό δίκτυο κινητής τηλεφωνίας, που ήταν αναγκαίο για την εξέλιξη και την εμπορική εκμετάλλευση του κινητού. Το πρώτο αυτοματοποιημένο δίκτυο κινητής τηλεφωνίας λειτούργησε στις αρχές τις δεκαετίας του '80 στην Σκανδιναβία. Μέχρι τα τέλη της δεκαετίας αυτής, τα κινητά τηλέφωνα ήταν ογκώδη για να μεταφέρονται στην τσέπη κι έτσι ήταν εγκατεστημένα στα αυτοκίνητα. Το πρώτο κινητό που έλαβε άδεια έγκρισης ήταν το μοντέλο της Motorola, το Dynatac 8000x.

Αυτό υπήρξε και η ναυαρχίδα των λεγόμενων κινητών τηλεφώνων της πρώτης γενιάς (1G), το 1983.

Στις αρχές της δεκαετίας του '90 άρχισε η απογείωση των κινητών με την ψηφιοποίηση δικτύων (GSM) και συσκευών. Τα κινητά έγιναν μικρότερα, χωρούσαν στην παλάμη και έμπαιναν έστω και με δυσκολία στην τσέπη του χρήστη. Πέρασαμε έτσι στην 2^η γενιά (2G) κινητών τηλεφώνων, τα οποία παρείχαν και άλλες ευκολίες, όπως την αποστολή σύντομων γραπτών μηνυμάτων (SMS) και αργότερα την λήψη φωτογραφιών και άλλες δυνατότητες που εξελίχθηκαν με τον καιρό.

Στις αρχές του 21^{ου} αιώνα ήλθαν τα κινητά τρίτης γενιάς (3G), με τις απεριόριστες δυνατότητες των πολυμέσων, πρόσβαση στο διαδίκτυο, email, ραδιόφωνο, τηλεόραση, εικονοκλήσεις. Τα κινητά με λειτουργικό σύστημα και τα λεγόμενα smart phones άρχισαν να κάνουν τα πρώτα βήματα τους στην αγορά και εν συνεχεία να κατακτήσουν το μεγαλύτερο μέρος του καταναλωτικού κοινού.

2.2 Smart phones και λειτουργικά συστήματα

Οι συσκευές κινητής τηλεφωνίας, πλέον, μπορούν να χαρακτηριστούν καλύτερα ως υπολογιστές με ενσωματωμένο σύστημα κινητής τηλεφωνίας παρά ως απλές συσκευές κινητών τηλεφώνων. Οι λειτουργίες και οι δυνατότητες που μας παρέχουν, καλύπτουν ένα μεγάλο μέρος των καθημερινών αναγκών μας, όπως ψυχαγωγία, ενημέρωση, εργασία, άμεση πρόσβαση σε δεδομένα, ικανότητα σύνδεσης στο διαδίκτυο και σχεδόν κάθε είδους επικοινωνίας.

Οι συσκευές αυτές, τα επονομαζόμενα smart phones, περιέχουν το κάθε ένα από αυτά κάποιο λειτουργικό σύστημα. Το λειτουργικό σύστημα είναι προεγκατεστημένο στα smart phones από τις κατασκευαστικές εταιρίες και δεν γίνεται να αντικατασταθεί. Σύμφωνα με αυτό επιλέγονται οι συσκευές από τις κατηγορίες του αγοραστικού κοινού.

Στην αγορά σήμερα υπάρχει ποικιλία λειτουργικών συστημάτων. Τα επικρατέστερα από αυτά είναι α) το iOS, β) το Windows Phone, γ) το Android. Στις παρακάτω παραγράφους θα αναφερθούμε ξεχωριστά στα βασικά χαρακτηριστικά του καθενός.

2.2.1 iOS

Το λειτουργικό σύστημα iOS είναι μία καινοτομία στον χώρο της τεχνολογίας. Το iOS σχεδιάστηκε από την εταιρία Apple Inc και χρησιμοποιείται αποκλειστικά στις συσκευές iPhone, iPad και iPod touch, οι οποίες είναι σχεδιασμένες από την ίδια εταιρία. Το πρώτο iPhone κυκλοφόρησε στην αγορά των ΗΠΑ στις 29 Ιουνίου 2007.



Λογότυπα της Apple

Το περιβάλλον χρήσης του, εξ' αρχής ήταν βασισμένο στην άμεση αλληλεπίδραση του χρήστη με την οθόνη αφής της συσκευής. Αυτό εξίταρε το καταναλωτικό κοινό, ειδικά την περίοδο που παρουσιάστηκε η πρώτη συσκευή της εταιρίας. Μ' αυτόν τον τρόπο ο χειρισμός γίνεται πιο ευχάριστος, γρήγορος αλλά και απλός για τον χρήστη, αφού μπορεί να αλληλεπιδρά με φυσικότητα με τα αντικείμενα που προβάλλονται στην οθόνη.

Το λειτουργικό σύστημα iOS είναι βασισμένο στο MAC OS X, επομένως είναι ένα Unix λειτουργικό σύστημα. Το λειτουργικό σύστημα χρησιμοποιεί περίπου 500 MB μνήμης από τον αποθηκευτικό χώρο της συσκευής. Το iOS για την ανάπτυξη εφαρμογών στο περιβάλλον του χρησιμοποιεί το λογισμικό ανάπτυξης εφαρμογών iOS SDK, το οποίο αναπτύχθηκε από την Apple και δόθηκε στους προγραμματιστές τον Φεβρουάριο του 2008. Η ανάπτυξη των εφαρμογών γίνεται σε Objective C, C, C++ με χρήση του Cocoa Touch API.

Αν και στην πρώτη έκδοση οι εφαρμογές που διατίθενται είναι ελάχιστες, η Apple σχεδόν κάθε χρόνο λάνσαρε μία καινούργια έκδοση. Σε κάθε νέα έκδοση προσθέτονται νέες εφαρμογές, όπως στην δεύτερη έκδοση (iOS 2.0) που πρωτοεμφανίστηκε το App Store, το ηλεκτρονικό κατάστημα εφαρμογών της Apple. Το App Store και το iOS 2.0 υποστήριζε εφαρμογές τρίτων και παρείχε πρόσβαση σε χιλιάδες εφαρμογές που δημιουργήθηκαν από διάφορους προγραμματιστές. Σε επόμενη έκδοση, το iOS 4.0, που διατέθηκε στην αγορά το 2010, έκανε την εμφάνιση του το πολυαναμενόμενο multitasking. Χαρακτηριστικό που έδινε την δυνατότητα στους χρήστες να μεταπηδούν μεταξύ των εφαρμογών χωρίς να χάνουν τη δουλειά τους. Η τελευταία έκδοση, το iOS 7 κυκλοφόρησε στην αγορά στις 18 Σεπτεμβρίου του 2013. Η έκδοση αυτή περιλαμβάνει ένα επανασχεδιασμένο περιβάλλον χρήσης με πολλές αλλαγές στην λειτουργικότητα του. Πιο ευκρινή και λιτά εικονίδια, ένα περιβάλλον εργασίας που δείχνει ομαλό και χαρούμενο και νέες δυνατότητες που εκμεταλλεύονται στο έπακρο την τεχνολογία της συσκευής.

Αξιόλογο είναι το γεγονός ότι σχεδόν κάθε συσκευή iPhone έχει τη δυνατότητα και είναι πολύ εύκολο να αναβαθμιστεί με κάποια καινούργια έκδοση.

2.2.2 Windows Phone

Το Windows Phone είναι το λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα που δημιούργησε η Microsoft, το οποίο αντικατέστησε το παλαιότερο λειτουργικό σύστημα της, το Windows Mobile. Η τελευταία έκδοση του Windows Mobile ήταν η 6.5 και η πρώτη του Windows Phone ήταν η 7, η οποία εμφανίστηκε στα τέλη του 2010. Παρόλο που το Windows Phone είναι τελείως διαφορετικό από το προγενέστερο του, η εταιρία αποφάσισε να συνεχίσει την αρίθμηση του. Προφανώς, λειτούργησε κατά αυτόν τον τρόπο ώστε να στηρίξει την απόφαση αυτή στο γεγονός ότι το νέο λειτουργικό βασίζεται στον πυρήνα Windows CE που στηριζόταν και το Windows Mobile.



Λογότυπο της Windows Phone

Το νέο λειτουργικό σύστημα περιλαμβάνει ένα εντελώς νέο περιβάλλον χρήσης το οποίο έχει δημιουργηθεί με μία γλώσσα σχεδίασης της ίδιας εταιρίας, που ονομάζεται Metro. Η γλώσσα αυτή μετονομάστηκε Modern UI με την έλευση της αμέσως επόμενης και τελευταίας έκδοσης, το Windows 8 OS. Η έκδοση Windows 8 βασίζεται στον πυρήνα Windows NT βοηθώντας έτσι στην εύκολη “μεταφορά” των εφαρμογών από το λειτουργικό για υπολογιστές στο λειτουργικό για smart phones.

Τα Windows 8 έδωσαν μία πρωτοτυπία στον σχεδιασμό των smart phones. Η οθόνη έναρξης προσφέρει την δυνατότητα στον χρήστη να τη διαμορφώσει όπως αυτός επιθυμεί, με τα γνωστά Tiles και Live Tiles. Αυτά είναι εικονίδια εφαρμογών και live widgets, μέσα σε τετράγωνα εικονίδια, τα οποία μπορεί να τα προσαρμόσει στις διαστάσεις που επιθυμεί, αλλά και τα χρώματα που του αρέσουν.

Τα Windows Phone είναι μία σειρά από λειτουργικά συστήματα smart phone κλειστού λογισμικού. Αυτό έχει ως απόρροια το γεγονός ότι δεν μπορεί ένας προγραμματιστής να το εκμεταλλευτεί πλήρως για να δημιουργήσει διάφορες εφαρμογές που να τρέχουν σε αυτό. Δυστυχώς η Microsoft παρέχει ορισμένα και λίγα APIs στους προγραμματιστές και αυτό έχει ως αποτέλεσμα τον μικρό αριθμό εφαρμογών στο Marketplace σε σχέση με τους ανταγωνιστές του, καθώς επίσης και την μέτρια ως χαμηλή ποιότητα τους.

2.2.3 Android

Το Android είναι και αυτό ένα λειτουργικό σύστημα για έξυπνα κινητά τηλέφωνα, το οποίο δημιουργήθηκε από την Google. Αναμφίβολα κατέχει τη μερίδα του λέοντος σε παγκόσμιο επίπεδο. Το μερίδιο του Android οφείλεται στο ότι είναι ένα ανοιχτό και “απελευθερωμένο” λειτουργικό σύστημα, το οποίο μπορεί να εγκατασταθεί σε οποιαδήποτε συσκευή, και δεν αναφερόμαστε μόνο στα smart phone, αλλά σε αυτόνομες μικρές συσκευές, τηλεοράσεις κ.α.

Επίσης η μεγάλη αυτή απήχηση του στο καταναλωτικό κοινό οφείλεται και στο γεγονός ότι οι τιμές των συσκευών του ξεκινάνε από πολύ χαμηλά, με αποτέλεσμα να είναι πιο προσιτό, σε αντίθεση με τους ανταγωνιστές του.

Το Android είναι δωρεάν, διαθέτει πληρέστατο κατάστημα εφαρμογών και επιπλέον μπορεί να δεχθεί εκτενέστατη τροποποίηση με μοναδικό όριο τις τεχνικές γνώσεις και το γούστο του χρήστη και του κατασκευαστή. Μπορεί να έχει την αρχική μορφή που του έχει δώσει η Google ή να αποτελέσει το υπόστρωμα ενός περιβάλλοντος χρήσης, όπου τα γραφικά του να μην θυμίζουν σε τίποτα Android.



Λογότυπο του Android

2.3 Γιατί επιλέξαμε Android

Το Android είναι μία μοναδική πλατφόρμα που επιτρέπει την ανάπτυξη εφαρμογών λογισμικού, το οποίο εκμεταλλεύεται πλήρως τις δυνατότητες μίας συμβατής συσκευής. Για παράδειγμα, οι προγραμματιστές εφαρμογών είναι ελεύθεροι να δημιουργήσουν εφαρμογές που χρησιμοποιούν οποιαδήποτε από τις βασικές λειτουργίες του τηλεφώνου, όπως η αποστολή μηνυμάτων, οι τηλεφωνικές κλήσεις, η λήψη φωτογραφιών, το GPS, κτλ. Έτσι, διευκολύνονται στην ανάπτυξη πιο περίπλοκου και πιο πλούσιου λειτουργικού λογισμικού. Αυτό το λειτουργικό σύστημα κινητών τηλεφώνων ή άλλων μικρών φορητών συσκευών που συνδέονται στο διαδίκτυο, στηρίζεται στον ελεύθερο πυρήνα του Linux. Επιπλέον, η

πλατφόρμα ανάπτυξης Android είναι μία πλατφόρμα multitasking, πράγμα που σημαίνει ότι κάθε εφαρμογή μπορεί να τρέξει στο τηλέφωνο ταυτόχρονα με κάποια άλλη χωρίς να επηρεαστεί η απόδοση του και αυτό είναι καλύτερο από το να περιορίζεται κάθε φορά σε μία εφαρμογή. Το Android είναι μία πλατφόρμα ανοιχτού κώδικα, το οποίο σημαίνει ότι πολύ εύκολα μπορεί να επεκταθεί και να τροποποιηθεί για να συμβαδίζει και να υιοθετεί τις τελευταίες τεχνολογικές εξελίξεις. Το γεγονός ότι και η πηγή της πλατφόρμας είναι ανοιχτή διασφαλίζει ότι η ανάπτυξη του Android θα έχει συνεχή πρόοδο και θα εξελίσσεται αφού ένας μεγάλος αριθμός ικανών Android προγραμματιστών εργάζεται για την δημιουργία ελεύθερων για χρήση προηγμένων εργαλείων λογισμικού.

Το Android είναι το πλέον ευέλικτο λειτουργικό σύστημα. Αυτός ήταν και ο πρωταρχικός ρόλος της επιλογής μας γι' αυτήν την εργασία. Παράλληλα η αναζήτηση για κάτι καινούργιο στον προγραμματισμό και την δημιουργία εφαρμογών μας οδήγησε στην χρήση του προγράμματος ανάπτυξης εφαρμογών, την Basic4Android. Το συγκεκριμένο IDE (Integrated Development Environment) ειδικεύεται στην πλατφόρμα του Android.

3

Λειτουργικό σύστημα Android

3.1 Εισαγωγή

Το Android είναι η πρώτη ολοκληρωμένη ανοιχτή και ελεύθερη πλατφόρμα που περιλαμβάνει ένα λειτουργικό σύστημα, το απαραίτητο ενδιάμεσο λογισμικό, βιβλιοθήκες και βασικές εφαρμογές. Αρχικά δημιουργήθηκε από την Google Inc και πλέον αναπτύσσεται από την Open Handset Alliance, μία κοινοπραξία από 48 τηλεπικοινωνιακές εταιρίες, εταιρίες λογισμικού καθώς και κατασκευής hardware. Η δημιουργία της πλατφόρμας αυτής στηρίχθηκε στον πυρήνα του λειτουργικού Linux. Αξίζει, επίσης να σημειωθεί ότι είναι ένα ελεύθερο λογισμικού ανοιχτού κώδικα (free and open source software), πράγμα που επιτρέπει στους χρήστες την δωρεάν ανάπτυξη κάθε είδους εφαρμογής.



Λογότυπο του Android

3.2 Ιστορικά στοιχεία

Η ιστορία ξεκίνησε όταν ο ευφυής Andy Rubin θέλησε την άνοιξη του 2005 να χρησιμοποιήσει την Google ως κατ' εξοχήν μηχανή αναζήτησης για το T-Mobile Sidekick, την οποία είχε αναπτύξει με την ομάδα συνεργατών του. Εν συνεχεία ζήτησε να συναντηθεί με τον Larry Page, έναν από τους δύο ιδρυτές της Google. Σ' αυτήν την συνάντηση ο Rubin παρουσίασε το Android ως ένα εν δυνάμει παγκόσμιο ανοιχτό λειτουργικό σύστημα που θα άλλαζε για πάντα τον τρόπο που διαντιδρούνε οι χρήστες με το κινητό τους. Ταυτόχρονα,

τόνισε τη σταθερή υπεροχή που παρατηρείται στις συνήθειες του αγοραστικού κοινού των κινητών τηλεφώνων σε αντιδιαστολή με τις πωλήσεις ηλεκτρονικών υπολογιστών.

Την ίδια στιγμή ο Page δεν ήθελε να γίνει απλώς ο υποστηρικτής του Android, ήθελε να γίνει ο ιδιοκτήτης του. Εκείνη την περίοδο, εμφανίστηκε στο προσκήνιο ένας ισχυρός παίχτης με ένα καινούργιο προϊόν, το iPhone. Έθεσε, έτσι, τους όρους του ανταγωνισμού σε άλλο επίπεδο και αυτό έδωσε ένα προβάδισμα στον Andy Rubin.

Ο επιχειρηματικός – τεχνολογικός κόσμος περίμενε πως η Google θα απαντούσε με ένα gPhone, αλλά αυτό δεν έγινε. Εν αντιθέσει έγινε κάτι σημαντικότερο. Το φθινόπωρο του 2005 ανακοινώνεται ότι 34 εταιρίες, όπως η Texas Instruments, η Intel, η T-Mobile και η Sprint Nextel, ενώνουν τις δυνάμεις τους με την Google για την δημιουργία μίας πλατφόρμας ανοιχτού κώδικα που θα έχει ενσωματωμένο το λογισμικό Linux και θα εκπροσωπείται από μία κοινοπραξία εταιριών που θα καλείται Open Handset Alliance. Αργότερα, στην συστάδα αυτή των εταιριών προστέθηκαν και άλλες εταιρίες, όπως η HTC, η Motorola και η LG, ανακοινώνοντας την πρόθεση τους να δώσουν προς πώληση στην αγορά smart phone με λειτουργικό σύστημα Android.

Η πρώτη παρουσίαση της πλατφόρμας έγινε στις 5 Νοεμβρίου του 2007, με την κυκλοφορία του Android 1.0 beta και το αντίστοιχο SDK (Software Development Kit). Η πρώτη εμπορική έκδοση, Android 1.0 ανακοινώθηκε τον Σεπτέμβριο του 2008. Από τον Απρίλιο του 2009, μέχρι και σήμερα έχουν δημοσιευτεί αρκετές ενημερώσεις για αυτό το λειτουργικό σύστημα.

3.3 Εκδόσεις

Η κάθε έκδοση του Android έχει πάρει την κωδική ονομασία της από το όνομα ενός επιδόρπιου και κυκλοφόρησαν με αλφαβητική σειρά: Cupcake (1.5), Donut (1.6), Éclair (2.0, 2.1), Froyo (2.2), Gingerbread (2.3.x), Honeycomb (3.x.x), Ice cream sandwich (4.0.x), Jelly Bean (4.1.x) και Kit Kat (4.4). Παρακάτω θα δούμε όλες τις υπάρχουσες εκδόσεις και θα αναφέρουμε επιγραμματικά τις εξελίξεις της κάθε μίας.

- Η πρώτη εμπορική έκδοση του λογισμικού, το Android 1.0 κυκλοφόρησε στις 23 Σεπτεμβρίου 2008. Η πρώτη συσκευή, το HTC Dream ενσωματώνει τα ακόλουθα χαρακτηριστικά:
 - Την εφαρμογή Android Market όπου ο χρήστης μπορεί να κατεβάσει νέες εφαρμογές και να αναβαθμίσει αυτές που ήδη έχει.
 - Browser για τη διαχείριση ιστοσελίδων

- Υποστήριξη κάμερας
 - Πρόσβαση σε e-mail servers, υποστηρίζοντας τα πρωτόκολλα POP-3, IMAP4, SMTP.
 - Ενσωμάτωσε τα Google Contacts, Calendar, Maps, Search, Talk
 - Υποστήριξη Wifi και Bluetooth
- Στις 9 Φεβρουαρίου του 2009, κυκλοφόρησε το Android 1.1. Αρχικά, κυκλοφόρησε μόνο για το T-Mobile G. Η ενημέρωση έλυσε σφάλματα, άλλαξε το API και πρόσθεσε μία σειρά από άλλα χαρακτηριστικά:
- Δυνατότητα να αποθηκεύονται τα συνημμένα σε μηνύματα
 - Λεπτομέρειες και κριτικές όταν ένας χρήστης κάνει αναζήτηση για επιχειρήσεις στο Google Maps
 - Προστέθηκε υποστήριξη για μαρκίζα σε διατάξεις του συστήματος
- Τον Απρίλιο του 2009 κυκλοφόρησε η αναβάθμιση 1.5 με κωδική ονομασία “cupcake” βασισμένη στο Linux Kernel 2.6.27. Η έκδοση αυτή, πρόσθεσε τα ακόλουθα χαρακτηριστικά:
- Υποστήριξη για widgets – μικροσκοπική θεώρηση της εφαρμογής που μπορεί να ενσωματωθεί σε άλλες εφαρμογές (όπως στην αρχική οθόνη) και να λαμβάνει περιοδικές ενημερώσεις
 - Προστίθενται χαρακτηριστικά αντιγραφής και επικόλλησης στο πρόγραμμα περιήγησης στο web
 - Δυνατότητα να ανεβάσετε βίντεο στο YouTube
 - Δυνατότητα να ανεβάσετε φωτογραφίες στο Picasa
 - Αυτόματη αντιστοίχιση (pairing) και υποστήριξη στέρεο για το Bluetooth
- Στις 15 Σεπτεμβρίου 2009 δημοσιεύθηκε το Android 1.6, που ονομάστηκε Donut, με βάση το Linux Kernel 6.2.29 και περιελάμβανε πολλά νέα χαρακτηριστικά:
- Η αναζήτηση με φωνή και κείμενο ενισχύθηκε με το ιστορικό του σελιδοδείκτη, της επαφής και το διαδίκτυο
 - Ευκολότερη αναζήτηση και την ικανότητα να δει κανείς στιγμιότυπα εφαρμογών στο Android Market



Android 1.5 Cupcake



Android 1.6 Donut

- Δυνατότητα για τους προγραμματιστές να συμπεριλάβουν το περιεχόμενο τους στα αποτελέσματα αναζήτησης
- Δυνατότητα στους χρήστες να επιλέξουν πολλές φωτογραφίες μαζί για διαγραφή
- Υποστήριξη ανάλυσης οθόνης WVGA
- Βελτίωση στην ταχύτητα όσον αφορά εφαρμογές σχετικές με την αναζήτηση και την φωτογραφική μηχανή

➤ Στις 26 Οκτωβρίου του 2009 δημοσιεύθηκε το Android 2.0/2.1 με την κωδική ονομασία Éclair, με βάση του Linux Kernel 2.6.29. Ενσωματώθηκαν οι παρακάτω αλλαγές:

- Διευρυμένη δυνατότητα συγχρονισμού λογαριασμών, επιτρέποντας στους χρήστες να προσθέσουν πολλούς λογαριασμούς σε μία συσκευή για το e-mail και τον συγχρονισμό των επαφών
- Υποστήριξη Bluetooth 2.1
- Δυνατότητα να αξιοποιήσει ο χρήστης μία φωτογραφία από τις επαφές και να επιλέξει να καλέσει αυτήν την επαφή, να στείλει γραπτό μήνυμα ή e-mail
- Βελτιωμένη ταχύτητα δακτυλογράφησης σε εικονικό πληκτρολόγιο, με εξυπνότερο λεξικό που μαθαίνει από την χρήση των λέξεων και περιλαμβάνει τα ονόματα των επαφών καθώς και διάφορες προτάσεις
- Ανανεωμένο UI του Browser με μικρογραφίες σελιδοδείκτη και υποστήριξη για HTML 5
- Βελτιστοποίηση της ταχύτητας του hardware και ανανεωμένο UI
- Βελτιωμένη έκδοση Google Maps 3.1.2



Android 2.0/2.1 Éclair

➤ Στις 20 Μαΐου του 2010 κοινοποιήθηκε το Android 2.2, το επωνομαζόμενο Froyo (Frozen Yogurt) με βάση το Linux Kernel 2.6.32. Περιείχε τις εξής αλλαγές:

- Βελτιστοποιήσεις στην ταχύτητα, τη μνήμη και στην επίδοση
- Λειτουργικότητα πρόσδεσης (tethering) USB και Wifi Hotspot
- Προστέθηκε μία επιλογή για απενεργοποίηση της πρόσβασης σε δεδομένα μέσω δικτύου κινητής τηλεφωνίας



Android 2.2 Froyo

- Φωνητική κλήση και ανταλλαγή επικοινωνίας μέσω Bluetooth
- Υποστήριξη για αριθμητικούς και αλφαριθμητικούς κώδικες πρόσβασης
- Υποστήριξη για την εγκατάσταση εφαρμογών στην επεκτάσιμη μνήμη
- Υποστήριξη Adobe Flash
- Βελτιωμένη εκκίνηση εφαρμογών με συντομεύσεις στις εφαρμογές του τηλεφώνου και στις εφαρμογές περιήγησης

➤ Το Android 2.3 ή Gingerbread ανακοινώθηκε στις 6 Δεκεμβρίου 2010 βασισμένο στο Linux Kernel 6.2.35. Οι αλλαγές που περιλαμβάνονται είναι:

- Ενημέρωση στον σχεδιασμό διεπαφής του χρήστη (user interface) κάνοντας το πιο απλό και πιο γρήγορο
- Υποστήριξη για εξαιρετικά μεγάλα μεγέθη οθόνης και μεγάλες αναλύσεις
- Εγγενής υποστήριξη για VoIP τηλεφωνία
- Ενισχυμένη λειτουργικότητα για αντιγραφή/ επικόλληση επιτρέποντας στους χρήστες να επιλέξουν μία λέξη πατώντας και διαλέγοντας αντιγραφή και επικόλληση
- Νέος διαχειριστής λήψεων, δίνοντας στους χρήστες εύκολη πρόσβαση σε κάθε αρχείο που κατεβάζει από τον browser, e-mail ή άλλη εφαρμογή
- Ενισχυμένη υποστήριξη για την ανάπτυξη εγγενή κώδικα
- Βελτιώσεις σε ήχο και γραφικά για τους προγραμματιστές παιχνιδιών
- Υποστήριξη συνομιλίας με φωνή ή βίντεο χρησιμοποιώντας το Google Talk
- Βελτιωμένη απόδοση μπαταρίας



Android 2.3 Gingerbread

➤ Η έκδοση 3.0, γνωστή ως Honeycomb κυκλοφόρησε στις 22 Φεβρουαρίου 2011. Είναι η πρώτη έκδοση που υποστηρίζει tablets με βάση τον πυρήνα του Linux 2.6.36. Οι αλλαγές που περιλαμβάνονται είναι:

- Βελτιστοποιημένη υποστήριξη για tablets με ένα νέο εικονικό και “ολογραφματικό” user interface
- Προστέθηκε μία μπάρα συστήματος, που διαθέτει γρήγορη πρόσβαση στις ειδοποιήσεις, η εμφάνιση της κατάστασης της συσκευής και τα μαλακά πλήκτρα πλοήγησης διαθέσιμα στο κάτω μέρος της οθόνης



Android 3.0 Honeycomb

- Απλοποιήθηκε το multitasking, με την μπάρα συστήματος ο χρήστης μπορεί να δει όλες τις εφαρμογές που τρέχουν και να μεταβεί από τη μία στην άλλη
- Επιτάχυνση του hardware
- Υποστήριξη για multi-core επεξεργαστές
- Δυνατότητα για την κρυπτογράφηση όλων των δεδομένων του χρήστη
- Συνδεσιμότητα για αξεσουάρ USB
- Υποστήριξη για χειριστήρια παιχνιδιών
- Υψηλή απόδοση για την διατήρηση της σύνδεσης μέσω Wifi όταν η οθόνη της συσκευής είναι απενεργοποιημένη
- Υποστήριξη HTTP proxy για κάθε συνδεδεμένο σημείο Wifi
- Βελτιωμένη υποστήριξη Adobe Flash στον Browser

- Στις 19 Οκτωβρίου του 2011 δημοσιοποιήθηκε το Android 4.0.1, το οποίο βασίζεται στο Linux Kernel 3.0.1. Ο υπάλληλος της Google, Gabe Cohen δήλωσε ότι το Android 4.0 είναι “θεωρητικά συμβατό” με οποιαδήποτε 2.3.x Android συσκευή που βρίσκεται στην παραγωγή εκείνη την εποχή. Ο πηγαίος κώδικας για το Android 4.0 έγινε διαθέσιμος στις 14 Νοεμβρίου 2011. Αυτή η έκδοση παρουσιάζει πολλά νέα χαρακτηριστικά, όπως:



Android 4.0.1 Ice cream sandwich

- Εικονικά κουμπιά στο περιβάλλον εργασίας αντί για φυσικά κουμπιά
- Βελτιωμένη διόρθωση λάθους στο πληκτρολόγιο
- Δυνατότητα πρόσβασης σε εφαρμογές απευθείας από την οθόνη κλειδώματος
- Βελτιωμένη λειτουργικότητα αντιγραφής και επικόλλησης
- Νέος web browser που επιτρέπει έως και 16 καρτέλες
- Ενότητα Χρήσης Δεδομένων στις ρυθμίσεις, που επιτρέπει τους χρήστες να ορίσουν προειδοποιήσεις όταν φθάσουν σε ένα συγκεκριμένο όριο χρήσης και να απενεργοποιήσουν τη χρησιμοποίηση δεδομένων όταν υπερβούν αυτό το όριο
- Δυνατότητα να κλείσει ο χρήστης εφαρμογές που χρησιμοποιούν δεδομένα στο παρασκήνιο

- Η έκδοση Android 4.1 με την κωδική ονομασία Jelly Bean ανακοινώθηκε στις 27 Ιουνίου του 2012. Η έκδοση αυτή είναι βασισμένη στο Linux 3.0.31. Τα σημαντικότερα χαρακτηριστικά αυτής της έκδοσης είναι:

- Βελτίωση της λειτουργικότητας και της απόδοσης



Android 4.1 Jelly Bean

της διεπαφής του χρήστη με κύρια χαρακτηριστικά που τη πιστοποιούν να είναι η πρόβλεψη θέσης δαχτύλων, triple buffering, παρατεταμένο σύστημα χρονισμού vsync, και άλλα

- Αυτόματη προσαρμογή widgets σε κάθε οθόνη.
- Νέος τρόπος λειτουργίας πληκτρολογίου με δυνατότητα πρόβλεψης και πρότασης της επόμενης λέξης.
- Περισσότερες επιλογές για άτομα με ειδικές ικανότητες
- Φωνητική πληκτρολόγηση που πλέον δουλεύει και χωρίς σύνδεση internet.
- Νέο γραφικό περιβάλλον στο αποτέλεσμα μίας αναζήτησης.
- Τεχνολογία Google Now, η οποία εμφανίζει αποτελέσματα αναζήτησης ανάλογα με την τοποθεσία του χρήστη και τα προσωπικά στοιχεία που έχει δηλώσει στο ημερολόγιο.

To Jelly Bean έβγαλε δύο ακόμα αναβαθμίσεις το Android 4.2 και το Android 4.3. Το Android 4.2 είναι βασισμένο στο Linux Kernel 3.4.0 και έκανε το ντεμπούτο της στις συσκευές Google Nexus 4 και το Nexus 10, οι οποίες βγήκαν στην αγορά στις 13 Ιουνίου 2012. Η Android 4.3 ανακοινώθηκε από την Google στις 24 Ιουνίου 2013. Στην Τρίτη γενιά του Jelly Bean παρατηρούνται βελτιώσεις στις τεχνολογίες που είδαμε για πρώτη φορά στο 4.1 κάνοντας τις συσκευές ακόμη πιο γρήγορες και επιπλέον υπήρξαν βελτιώσεις και στην ασφάλεια.

- Η Google δημοσίευσε την έκδοση Android 4.4 Kit Kat στις 3 Σεπτεμβρίου 2013. Η έκδοση Kit Kat έκανε το ντεμπούτο της στις συσκευές Nexus 5, στις 31 Οκτωβρίου του 2013 και έχει βελτιστοποιηθεί ώστε να τρέχει σε μεγαλύτερη γκάμα συσκευών από τις προηγούμενες εκδόσεις. Μερικά βασικά χαρακτηριστικά της έκδοσης αυτής είναι τα εξής:

- Σχεδιάστηκε με τέτοιο τρόπο ώστε να μπορεί να προσαρμόζεται σε συσκευές με μνήμη RAM μόλις 512 Mb.
- Ταχύτερο multitasking
- Ανανεωμένο Google Now με swipe στην αριστερή πλευρά της οθόνης.
- Built in Google Cloud Print και υποστήριξη νέων εκτυπωτών HP
- Η κάμερα λειτουργεί σαν HDR+. Με την λήψη της φωτογραφίας και η κάμερα επιλέγει τον καλύτερο συνδυασμό χρωμάτων και φτιάχνει το καλύτερο δυνατό



Android 4.4 Kit Kat

αποτέλεσμα. Αν και αυτή η λειτουργία είναι διαθέσιμη προς το παρόν μόνο στο Google Nexus 5.

- Ανανεωμένο Hangouts και πλέον υποστηρίζει και τα μηνύματα
- Διαφανής μπάρα εργαλείων και διαφανής Notification bar.

3.4 Google Play

Το Google play, πρώην Android Market, είναι η ψηφιακή πλατφόρμα διανομής εφαρμογών για το λειτουργικό σύστημα Android και είναι το online κατάστημα που λειτουργεί από την Google.

Η υπηρεσία επιτρέπει στους χρήστες να περιηγηθούν και να κατεβάσουν εφαρμογές που έχουν αναπτυχθεί με το Android SDK και δημοσιεύονται μέσω της Google, καθώς και μουσική, περιοδικά, βιβλία, ταινίες και πρόγραμμα τηλεόρασης. Οι χρήστες μπορούν επίσης να αγοράσουν hardware υλικό, όπως Chromebooks, συσκευές Google Nexus, Chromecasts και αξεσουάρ.

Οι εφαρμογές είναι διαθέσιμες στο Google Play είτε δωρεάν ή με κάποιο κόστος. Μπορούν να κατέβουν απευθείας σε μία συσκευή ή Google TV μέσω της συγκεκριμένης εφαρμογής (Google Play) ή ξεκινώντας μία εφαρμογή σε μία συσκευή μέσω της ιστοσελίδας του Google Play.

Πολλές εφαρμογές μπορεί να απευθύνονται σε συγκεκριμένους χρήστες βασισμένο σε κάποιο χαρακτηριστικό της συσκευής τους, όπως ο αισθητήρας κίνησης (για παιχνίδια που εξαρτώνται από την κίνηση) ή μία κάμερα στο μπροστινό μέρος (για κλήσεις video).

Στις 6 Μαρτίου του 2012 με την συγχώνευση του Android Market και του Google Music, η υπηρεσία μετονομάστηκε με το σημερινό της όνομα ώστε να συμβαδίζει με την μετονομασία της στρατηγικής για την ψηφιακή διανομή της Google.

Τον Ιούλιο του 2013 το online κατάστημα έχει επίσημα δημοσιεύσει πάνω από 1 εκατομμύριο εφαρμογές και τα downloads έχουν ξεπεράσει τα 50 δισεκατομμύρια.



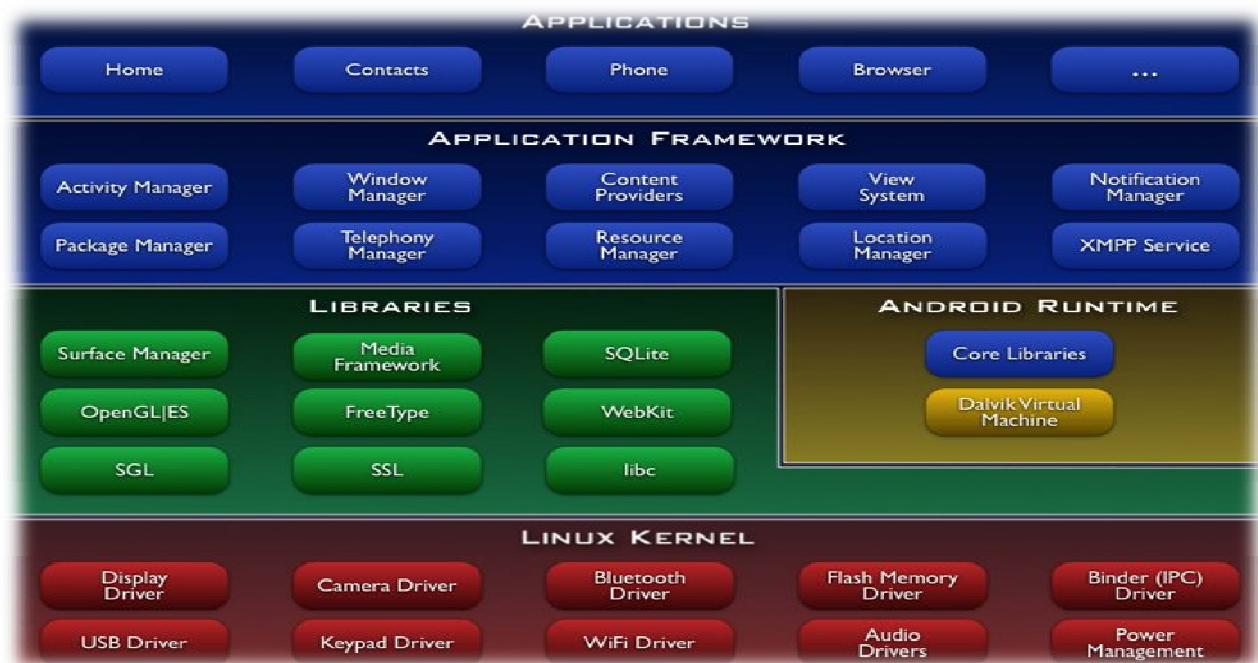
Λογότυπο Google Play

3.5 Αρχιτεκτονική του Android

Το Android μπορεί να θεωρηθεί ως μία στοίβα λογισμικού από διαφορετικές στοιβάδες. Η κάθε στοιβάδα διαθέτει μία ή δύο ομάδες προγραμμάτων για την ομαλή και σωστή λειτουργία του συστήματος, το λειτουργικό σύστημα, το ενδιάμεσο λογισμικό και τις σημαντικές εφαρμογές.

Κάθε στοιβάδα της αρχιτεκτονικής παρέχει διαφορετικές υπηρεσίες στην στοιβάδα που βρίσκεται ακριβώς επάνω από αυτήν. Επιγραμματικά, τα επίπεδα του Android αναφέρονται παρακάτω:

1. Ο πυρήνας του Linux, που περιλαμβάνει οδηγούς (drivers) για να τρέξει το σύστημα
2. Οι εγγενείς βιβλιοθήκες και η ομάδα του χρόνου εκτέλεσης
3. Το πλαίσιο εφαρμογών
4. Οι εφαρμογές και τα widgets, οι οποίες είναι γραμμένες σε Java και εκτελούνται στην εικονική μηχανή Dalvik.



Αρχιτεκτονική του Android

3.5.1 Πυρήνας Linux

Η βάση του Android αποτελείται από τον πυρήνα της Linux των εκδόσεων 2.6.x και 3.x τροποποιημένες ώστε να λειτουργούν αποτελεσματικά στις συσκευές κινητής τηλεφωνίας. Το επίπεδο αυτό χειρίζεται υπηρεσίες συστήματος, όπως την διαχείριση των οδηγών (drivers) του υλικού του συστήματος (hardware), τη διαχείριση ασφάλειας, τη διαχείριση μνήμης, τη διαχείριση διαδικασιών, τη διαχείριση ενέργειας και τη διαχείριση δικτύωσης δηλαδή τη διαχείριση των διεπαφών δικτύου που διαθέτει κάθε συσκευή.

Ο πυρήνας Linux αποτελεί μία δοκιμασμένη πλατφόρμα, η οποία προσφέρει πολλές δυνατότητες. Το γεγονός ότι είναι μία πλατφόρμα ανοιχτού κώδικα, ελευθερώνει τις δυνατότητες των προγραμματιστών και των κατασκευαστών κινητών τηλεφώνων να προχωρήσουν και να αναβαθμίσουν την πλατφόρμα.

Ένα άλλο πλεονέκτημα του πυρήνα Linux είναι η απαιτούμενη αξιοπιστία που ζητάει ο χρήστης άλλα και η καλή απόδοση της συσκευής. Επιπλέον, ο πυρήνας λειτουργεί ως ένα ενδιάμεσο αφηρημένο επίπεδο (abstraction layer) μεταξύ του υλικού και του υπόλοιπου λογισμικού της στοίβας. Αυτό σημαίνει ότι τα ανώτερα επίπεδα δεν χρειάζεται να υποστούν αλλαγές, αν αλλάξει το hardware. Έτσι, καθώς εμφανίζονται νέες συσκευές και εξαρτήματα από πλευράς υλισμικού, οι drivers μπορούν να γράφονται πάνω στον πυρήνα για να παρέχουν υποστήριξη ακριβώς όπως σε άλλες πλατφόρμες Linux.

3.5.2 Εγγενής Βιβλιοθήκες

Το αμέσως επόμενο επίπεδο είναι οι εγγενής βιβλιοθήκες (native libraries) του Android. Οι βιβλιοθήκες είναι γραμμένες σε γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το τηλέφωνο. Οι εγγενής βιβλιοθήκες γράφτηκαν με σκοπό την κλήση τους από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και έπειτα, δόθηκε η δυνατότητα σε όλους τους προγραμματιστές να γράφουν τις δικές τους βιβλιοθήκες με την βοήθεια της εργαλειοθήκης NDK (Native Development Kit).

Μερικές από τις σημαντικότερες βιβλιοθήκες είναι:

- ‘System C Library’ ή ‘Bionic Library’, είναι η υλοποίηση της πρότυπης βιβλιοθήκης συστήματος C (libc), συντονισμένη για συσκευές βασισμένες σε Linux. Επειδή η συγκεκριμένη βιβλιοθήκη έπρεπε να φορτώνεται σε κάθε διεργασία, απαιτούνταν να έχει μικρό μέγεθος. Έτσι δημιουργήθηκε η Bionic και είχε μέγεθος περίπου

200Kb, σχεδόν το μισό μέγεθος της libc, εξαιτίας αυτού είναι και πιο γρήγορη. Χρησιμοποιείται για υπηρεσίες του Android όπως ιδιότητες συστήματος και αρχεία καταγραφής. Δεν υποστηρίζει ορισμένα χαρακτηριστικά POSIX, όπως εξαιρέσεις σε C++ και wide chars, τα οποία δεν χρειάζονται στο Android. Επίσης, δεν είναι συμβατή με την GNU libc. Όλος ο εγγενής κώδικας μεταγλωττίζεται πάνω στην bionic και όχι στην libc.

- ‘Media Libraries’, μία βιβλιοθήκη πολυμέσων που περιέχει αποκωδικοποιητές (codecs) για αναπαραγωγή και εγγραφή πολυμέσων διάφορων τύπων όπως MPEG, MP3, JPG, κλπ.
- ‘Surface Manager’, μία βιβλιοθήκη για την διαχείριση των παραθύρων όπως διάφανα παράθυρα και κομψές μεταβάσεις μεταξύ αυτών. Καθώς και την σύνδεση δισδιάστατων (2D) και τρισδιάστατων (3D) γραφικών.
- ‘SQLite’, μία βιβλιοθήκη για την υποστήριξη της βάσης SQL. Κρατάει επίσης αποθηκευτικό χώρο για οποιαδήποτε εφαρμογή.
- ‘SSL’, είναι ένα πρωτόκολλο για την ασφάλεια της επικοινωνίας μεταξύ του web browser και του web server μέσω ενός πιστοποιητικού.
- ‘WebKit’ για την μηχανή αναζήτησης ιστού και την ασφάλεια του διαδικτύου. Είναι η βιβλιοθήκη που υποστηρίζει τον Google Chrome και το Safari.

3.5.3 Χρόνος Εκτέλεσης

Το περιβάλλον χρόνου εκτέλεσης του Android αποτελείται από ένα σύνολο βασικών βιβλιοθηκών και την εικονική μηχανή Dalvik. Το Android runtime βρίσκεται στο ίδιο επίπεδο με τις εγγενής βιβλιοθήκες, αλλά σε διαφορετική ομάδα. Οι βιβλιοθήκες πυρήνα (Core Libraries) παρέχουν αρκετές από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της γλώσσας Java. Κάποιες άλλες δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα έχουν προστεθεί και άλλες (Android – specific libraries) προσαρμοσμένες στις δικές του ανάγκες.

Στις συμβατικές συσκευές υπολογιστών, το λογισμικό τρέχει απευθείας στον πυρήνα του λειτουργικού συστήματος, αλλά μία εφαρμογή Android τρέχει στην δική της διαδικασία με την υπόδειξη της εικονικής μηχανής Dalvik. Η μηχανή αυτή δημιουργήθηκε έτσι ώστε μία κινητή συσκευή να μπορεί να τρέξει πολλά στιγμιότυπα με αποτελεσματικό τρόπο. Η Dalvik δεν είναι η τυπική Java Virtual Machine (JVM) αλλά στην ουσία είναι μία βελτιστοποιημένη υλοποίηση μίας εικονικής μηχανής Java για φορητές συσκευές. Η Dalvik δημιουργεί εκτελέσιμα αρχεία (.dex αρχεία), τα οποία είναι bytes codes που προέρχονται από αρχεία .class και .jar, προκειμένου να τα τρέξει το λειτουργικό σύστημα. Εν αντιθέσει όμως με τα

.class αρχεία, τα .dex αρχεία είναι πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη, επεξεργαστική ισχύ και μπαταρία.

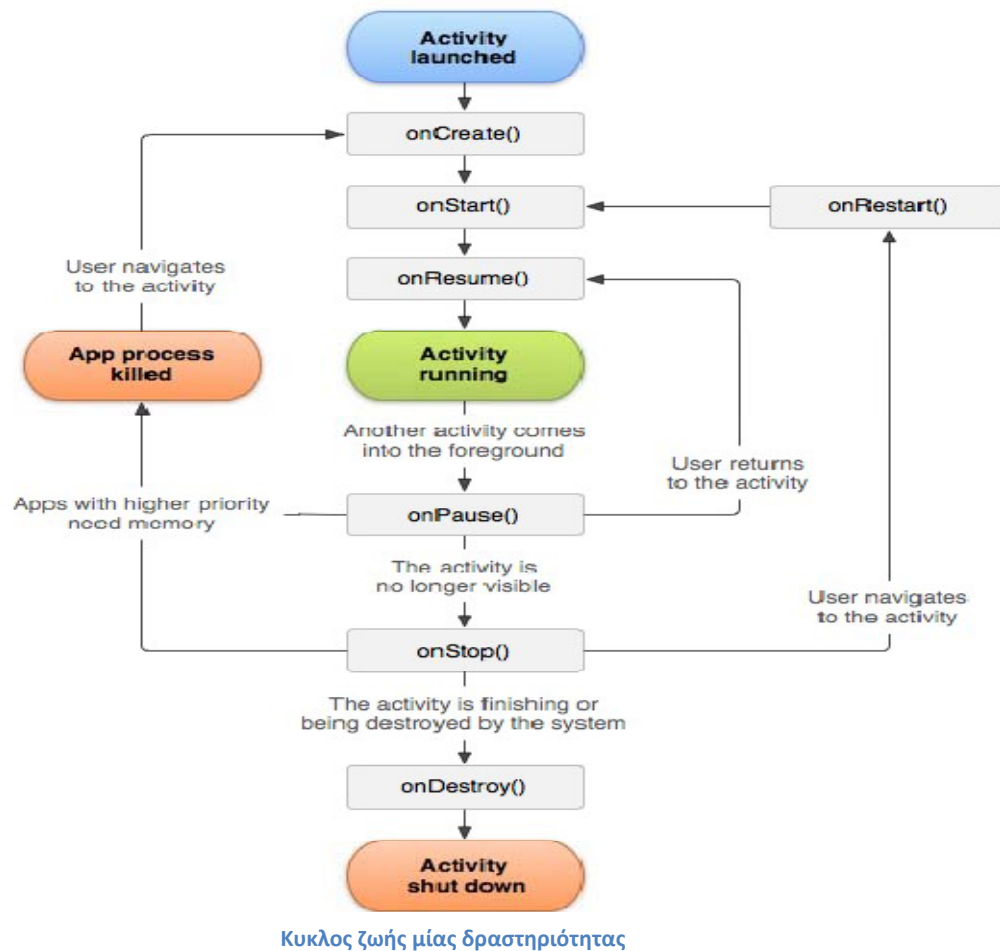
Με την έκδοση 2.2 ενσωματώθηκε ένας Just-in-time (JIT) compiler στη μηχανή Dalvik, ο οποίος μεταφράζει τον κώδικα της εφαρμογής σε μία μορφή που τρέχει πιο γρήγορα ενώ η εφαρμογή συνεχίζει να τρέχει. Οι βελτιώσεις που επιτυγχάνονται είναι από διπλάσιες έως πενταπλάσιες σε σύγκριση με την προηγούμενη Dalvik. Η μηχανή Dalvik βασίζεται στον πυρήνα Linux για λειτουργίες όπως threading και χαμηλού επιπέδου διαχείρισης μνήμης.

3.5.4 Πλαίσιο εφαρμογής

Στην επόμενη στοιβάδα βρίσκεται το πλαίσιο εφαρμογής (application framework). Αυτό το επίπεδο προσφέρει στους προγραμματιστές την πλήρη πρόσβαση στις έτοιμες δομικές μονάδες (API Interface) που χρησιμοποιούνται για την δημιουργία εφαρμογών. Η δομή των εφαρμογών έχει σχεδιαστεί έτσι ώστε να ευνοείται η επαναχρησιμοποίηση δομικών συστατικών και επίσης επιτρέπεται η χρήση των δυνατοτήτων μίας εφαρμογής από άλλες εφαρμογές. Όλα αυτά, βέβαια, υπόκεινται σε περιορισμούς ασφαλείας που επιβάλλονται από το πλαίσιο εφαρμογής. Τα σημαντικότερα δομικά στοιχεία του πλαισίου είναι:

- Σύστημα προβολών (View System), αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI, τα οποία μπορούν να χρησιμοποιηθούν κατά τον σχεδιασμό μίας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (list view), το πλέγμα (grid view), πεδία εισαγωγής κειμένου, κουμπιά, κτλ.
- Πάροχος Περιεχομένου (Content Provider), ο οποίος δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μίας συγκεκριμένης μορφής που 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. Παραδείγματα αυτών είναι οι εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ.
- Διαχειριστής ειδοποιήσεων (Notification Manager), που 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. Στο παρακάτω σχεδιάγραμμα φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας.



3.5.5 Εφαρμογές & Widgets

Οι εφαρμογές και τα widgets αποτελούν το υψηλότερο στρώμα της αρχιτεκτονικής του Android. Αυτό το κομμάτι είναι που βλέπουν και χρησιμοποιούν οι χρήστες με διαφάνεια ως προς το τι συμβαίνει πίσω από αυτές ή τι απαιτείται για την εκτέλεση τους από το λειτουργικό σύστημα.. Μπορεί να είναι εφαρμογές που είναι ήδη εγκατεστημένες στη συσκευή, καθώς επίσης και όλες αυτές που εγκαθιστά ο χρήστης αργότερα μέσω του Google Play.

Αν και τα στοιχεία του λειτουργικού είναι γραμμένα σε C και C++, οι εφαρμογές είναι γραμμένες σε Java, όπως έχουμε ήδη αναφέρει. Το επίπεδο αυτό μπορεί να θεωρηθεί ως το πιο σημαντικό, διότι οι συσκευές χωρίς τις βασικές εφαρμογές τους, π.χ. οι επαφές, η εφαρμογή για κλήση αριθμού, τα μηνύματα, δεν εξυπηρετούν κανέναν απολύτως σκοπό.

Κάθε εφαρμογή αποτελείται από ένα σύνολο αρχείων και φακέλων δομημένα σε μορφή project, τα οποία αφού μεταγλωττιστούν μέσω του Android SDK (Software Development Kit) μας δίνουν το αρχείο .apk. Όλες οι εφαρμογές πρέπει να έχουν ένα μοναδικό όνομα πακέτου (package name), το οποίο χρησιμοποιείται από το λειτουργικό σύστημα για αναγνώριση της εφαρμογής. Μία εφαρμογή μπορεί να αποτελείται από πολλά υποπακέτα, εφόσον αυτό είναι απαραίτητο λόγω της πολυπλοκότητας της εφαρμογής, αλλά μόνο από ένα κύριο. Το αρχείο αυτό είναι το AndroidManifest.xml, στο οποίο βρίσκονται αποθηκευμένες όλες οι σημαντικές πληροφορίες της εφαρμογής. 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.

4

Πρόγραμμα ανάπτυξης εφαρμογών Basic4Android

4.1 Γενικά

Η Google έχει ορίσει το Eclipse ως το επίσημα υποστηριζόμενο περιβάλλον ανάπτυξης εφαρμογών για συσκευές. Παρ' όλα αυτά υπάρχουν στην αγορά και άλλα εργαλεία ανάπτυξης, όπως το Basic4Android. Στην εργασία αυτή θα αναπτύξουμε και θα παρουσιάσουμε μία απλή εφαρμογή μέσω του Basic4Android. Πρώτα από όλα, όμως, ας δούμε μερικά βασικά στοιχεία για το συγκεκριμένο περιβάλλον.

4.2 Ορισμός

Η Basic4Android είναι ένα ολοκληρωμένο ψηφιακό εργαλείο που αναπτύχθηκε ώστε να βοηθήσει στη δημιουργία εφαρμογών για συσκευές Android. Είναι ένα από τα γρηγορότερα αναπτυσσόμενα εργαλεία ανάπτυξης εφαρμογών (Rapid Application Development – RAD) που υπάρχουν στην αγορά. Ένα εμπορικό προϊόν, το οποίο είναι εμπνευσμένο από την γλώσσα προγραμματισμού Microsoft Visual Basic και το Microsoft Visual Studio.



Η Basic4Android διατίθεται από την εταιρία Anywhere Software και οι δημιουργοί της είναι ο Erel Uziel και ο Klaus Christl. Σκοπός της εταιρίας είναι η προώθηση εργαλείων ανάπτυξης εφαρμογών για πλατφόρμες κινητών τηλεφώνων, tablets και τελευταία

προστέθηκε και η δυνατότητα ανάπτυξης εφαρμογών για desktop ανεξαρτήτως λειτουργικού συστήματος. Η ιστορία της εταιρίας Anywhere Software ξεκινάει το 2005 με την Basic4ppc, ένα ολοκληρωμένο εργαλείο ανάπτυξης εφαρμογών για συσκευές Windows Mobile. Το 2010, με την αύξηση της δημοτικότητας του Android, η εταιρία λάνσαρε ένα νέο εργαλείο, το Basic4Android για συσκευές Android όπως λέει και το όνομα του. Το εργαλείο αυτό είναι πλέον και η ναυαρχίδα της εταιρίας.

Το περιβάλλον της Basic4Android είναι μία ολοκληρωμένη πρόταση, φιλική προς τον κάθε χρήστη, με κύριο χαρακτηριστικό την εργαλειοθήκη ταχείας ανάπτυξης που επιτρέπει τόσο στους έμπειρους, όσο και στους αρχάριους προγραμματιστές να αναπτύξουν εφαρμογές υψηλής απόδοσης.

Ήδη σημαντικό, επίσης είναι να αναφέρουμε ότι υπάρχει μία συνεχώς αναπτυσσόμενη και ενεργή διαδικτυακή κοινότητα, που αυτή τη στιγμή αριθμεί πάνω από 40000 προγραμματιστές. Οι προγραμματιστές αυτοί είναι αφιερωμένοι στην βοήθεια του κάθε χρήστη για την σχεδίαση και την εκτέλεση των εφαρμογών πραγματικού χρόνου για κινητά τηλέφωνα.

Το εργαλείο αυτό μπορεί κάποιος να το κατεβάσει από την ιστοσελίδα της Anywhere Software. Υπάρχει διαθέσιμη μία δωρεάν έκδοση για 30 ημέρες. Ωστόσο, η δοκιμαστική έκδοση δεν υποστηρίζει εξωτερικές μονάδες και όλες τις βιβλιοθήκες. Έτσι, για να δημιουργηθούν πιο προηγμένες εφαρμογές θα χρειαστεί να αγοραστεί η πλήρης άδεια. Αυτή κοστίζει μόλις 49 δολάρια και παρέχει την δυνατότητα της πλήρους πρόσβασης στην διαδικτυακή κοινότητα και επίσης να κατεβάσετε κομμάτια κώδικα ή βιβλιοθήκες από το φόρουμ. Σε περίπτωση που η χρήση του προγράμματος αφορά ακαδημαϊκούς σκοπούς, τότε υπάρχει έκπτωση έως 50% στην αρχική τιμή.

4.3 Χαρακτηριστικά και δυνατότητες

Η Basic4Android είναι ένα ισχυρό εργαλείο ανάπτυξης εφαρμογών πολύ καλά δομημένο και τακτοποιημένο. Ένα ολοκληρωμένο εργαλείο που στοχεύει 100% στην ανάπτυξη του Android. Ένα βασικό χαρακτηριστικό της είναι ότι δεν απαιτεί βιβλιοθήκες χρόνου εκτέλεσης, διότι τα .apk αρχεία της είναι πανομοιότυπα με εκείνα που δημιουργούνται με την Java και το Eclipse.

Ο οπτικός σχεδιαστής που παρέχει μπορεί να χρησιμοποιηθεί για την δημιουργία διασύνδεσης των εφαρμογών. Επιτρέπει να προστεθούν ενεργά εικονίδια όπως πλήκτρα (buttons), πλαίσια ελέγχου (checkbox), ετικέτες (labels), edit text, list view, panels και πολλά

άλλα. Για κάθε ένα στοιχείο που προστίθεται, δίνετε η δυνατότητα στον χρήστη να ρυθμίσει τις ιδιότητες του, ώστε να ταιριάζουν απόλυτα σ' αυτό που χρειάζεται. Ο σχεδιαστής που διατίθεται είναι βασισμένος στο WYSIWYG (What You See Is What You Get). Επίσης, ο οπτικός συντάκτης υποστηρίζει την δημιουργία πολλαπλών διατάξεων ώστε να ταιριάζουν σε διαφορετικού μεγέθους οθόνες. Για να ενισχυθεί αυτή η λειτουργία και να προστεθεί στην ευελιξία της Basic4Android, μπορεί να χρησιμοποιηθεί ένα χαρακτηριστικό που ονομάζεται 'Designer Scripts'. Το χαρακτηριστικό αυτό αναπτύχθηκε ώστε να βοηθήσει στην προσαρμογή των λεπτομερειών της κάθε διάταξης, για να μπορούν εύκολα να ενταχθούν σε διαφορετικές οθόνες.

Το πρόγραμμα επεξεργασίας κώδικα, εκτός του ότι έχει πολύ ωραίο περιβάλλον, έρχεται με την ιδιότητα της αυτόματης συμπλήρωσης που επιτρέπει την εισαγωγή προκαθορισμένων εντολών. Μπορεί, επίσης, ο χρήστης να λάβει οποιαδήποτε συμβουλή μέσω του εσωτερικού ευρετηρίου με ενσωματωμένη τεκμηρίωση, οι οποίες θα τον βοηθήσουν να δημιουργήσει μία σταθερή και αποτελεσματική εργασία.

Είναι πολύ σημαντικό να αναφέρουμε ότι η γλώσσα είναι object oriented (OOP), όπως κάθε σύγχρονη Basic. Τα αντικείμενα υπάρχουν για να έχει πρόσβαση ο χρήστης στο διαδίκτυο, σε βάσεις δεδομένων SQL, να σχεδιάζει διαγράμματα και πολλά άλλα.

Η χρήση της Basic4Android είναι αρκετά εύκολη, αφού η σύνταξη του κώδικα είναι παρόμοια μ' αυτή της Visual Basic. Ακόμα, μπορούμε να πούμε ότι δεν χρειάζονται γνώσεις προγραμματισμού XML. Οι δυνατότητες της μπορούν να ενισχυθούν με την χρήση προσαρμοσμένων βιβλιοθηκών της Java.

Αξιόλογο χαρακτηριστικό είναι και το πρόγραμμα εντοπισμού σφαλμάτων που υποστηρίζει γρήγορη ανάπτυξη και hot code swapping. Ένα χαρακτηριστικό που δεν υποστηρίζεται από κανένα άλλο εργαλείο ανάπτυξης εφαρμογών Android.

Τέλος, οι λόγοι που μπορεί να επιλέξει κάποιος προγραμματιστής την Basic4Android είναι ότι πέρα από την ευκολία και την ταχύτητα που μας προσφέρει, μπορεί και υποστηρίζει όλα τα βασικά χαρακτηριστικά του Android. Μερικά από αυτά αναφέρονται παρακάτω:

- Σύνδεση με βάση δεδομένων SQL
- GPS
- Widgets & Live wallpapers
- Υπηρεσίες διαδικτύου
- Υπηρεσίες HTTP (βασισμένο στο Jetty)
- Ειδοποιήσεις PUSH (GCM & CZDM)
- Δικτύωση (TCP, UDP, FTP, SFTP, SMTP, POP3 & SMB/CIFS, Wifi Direct)

- Κωδικοποίηση JSON, XML, XLS (MS EXCELL), MDB (MS ACCESS) και CSV format
- Υπηρεσίες παρασκηνίου (Multitasking) και broadcast listeners
- Μετατροπή κειμένου σε ομιλία και αναγνώριση φωνής
- Διαφημίσεις AdMob και άλλες πλατφόρμες διαφημίσεων
- Αισθητήρες
- Ενσωμάτωση με υπηρεσίες όπως Google, Facebook, Twitter και άλλες
- Υπηρεσίες αγοράς σε μορφή εφαρμογής (Google Play)
- SIP/ VoIP υπηρεσίες
- Bluetooth
- Χαρακτηριστικά τηλεφώνου όπως κλήσεις, μηνύματα, επαφές
- USB Host / accessory modes και MTB
- Κάμερα
- Multitouch gestures
- Views Animation
- NFC
- Video and audio playback/ recording
- OpenGL, libGPX
- Graphics & Charts
- Files I/O
- Object serialization
- Device administration Apps
- Andruino ADK
- Και άλλα.....

Στο επόμενο κεφάλαιο κάποια από αυτά θα τα δούμε λίγο πιο αναλυτικά.

5

Βασικά εργαλεία ανάπτυξης

5.1 Γενικά

Εν αρχή όλων θα πρέπει να ξεκινήσουμε με τις απαραίτητες διαδικασίες και εγκαταστάσεις των βασικών εργαλείων για την ανάπτυξη των εφαρμογών. Τα εργαλεία αυτά βοηθάνε τον προγραμματιστή σε όλη την διαδικασία της ανάπτυξης, από την μεταγλώττιση του πηγαίου κώδικα ως την ψηφιακή υπογραφή της εφαρμογής για την διανομή της. Για την δημιουργία μίας εφαρμογής συνήθως ακολουθείται μία συγκεκριμένη πορεία ανάπτυξης, η οποία περιλαμβάνει τέσσερα βήματα όπως φαίνεται στο παρακάτω σχήμα.

5.2 Java JDK

Ως γνωστόν, όλη η ανάπτυξη εφαρμογών Android στηρίζεται στην στάνταρ έκδοση γλώσσας προγραμματισμού Java (Java Platform Standard Edition – Java SE) και για αυτό θα πρέπει πρώτα από όλα να εγκαταστήσουμε την τελευταία έκδοση του JDK (Java Development Kit). Μπορούμε να κατεβάσουμε το Java 7 JDK απ' ευθείας από την ιστοσελίδα της Oracle, που είναι και η εταιρία που το διανέμει

(<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>)

ή μέσω του συνδέσμου που υπάρχει στη σελίδα της Anywhere Software

(<http://www.basic4ppc.com/android/downloads.html>)

Επιλέγουμε την κατάλληλη πλατφόρμα που αντιστοιχεί στο λειτουργικό μας σύστημα και κατεβάζουμε το αρχείο.

Java SE Development Kit 7 x

www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Αγγλοελληνικό Λεξι... Design | Android De... Basic4android (Basic... Tutorial - Android P... Basic4Android: Hell... video.cloudfront.pc... video.cloudfront.pc... » Άλλοι σελιδοδείκτες

ORACLE® Sign In/Register Help Country Communities I am a... I want to... Search

Products Solutions Downloads Store Support Training Partners About OTN

Oracle Technology Network Java Java SE Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

Looking for JavaFX SDK?
JavaFX SDK is now included in JDK 7 for Windows, Mac OS X, and Linux i686/x64.

See also:

- Java Developer Newsletter (click the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK MD5 Checksum

Looking for JDK on ARM?
JDK 7 for ARM downloads have moved to the JDK 7 for ARM download page.

Java SE Development Kit 7u60

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux i686	119.67 MB	jdk-7u60-linux-i686.rpm
Linux i686	136.95 MB	jdk-7u60-linux-i686.tar.gz
Linux i64	120.97 MB	jdk-7u60-linux-i64.rpm
Linux i64	133.77 MB	jdk-7u60-linux-i64.tar.gz
Mac OS X i64	166.94 MB	jdk-7u60-macosx-i64.dmg
Solaris i686 (SVR4 package)	139.43 MB	jdk-7u60-solaris-i686.tar.Z
Solaris i686	95.5 MB	jdk-7u60-solaris-i686.tar.gz
Solaris i64 (SVR4 package)	24.64 MB	jdk-7u60-solaris-i64.tar.Z
Solaris i64	16.33 MB	jdk-7u60-solaris-i64.tar.gz
Solaris SPARC (SVR4 package)	138.73 MB	jdk-7u60-solaris-sparc.tar.Z
Solaris SPARC	98.67 MB	jdk-7u60-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24.04 MB	jdk-7u60-solaris-sparc64.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u60-solaris-sparc64.tar.gz
Windows i686	127.91 MB	jdk-7u60-windows-i686.exe
Windows i64	129.65 MB	jdk-7u60-windows-i64.exe

Java SE Development Kit 7u60 Demos and Samples Downloads

Java SDKs and Tools

- Java SE
- Java EE and GlassFish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

Java magazine Get it now for FREE! Subscribe Today

Webcast Introducing Java 8 Watch Now

Αφού αποθηκεύσουμε το αρχείο, το εκτελούμε και το εγκαθιστούμε δεχόμενοι τις προκαθορισμένες ρυθμίσεις σε κάθε παράθυρο.



Εγκατάσταση Java JDK

5.3 Android SDK

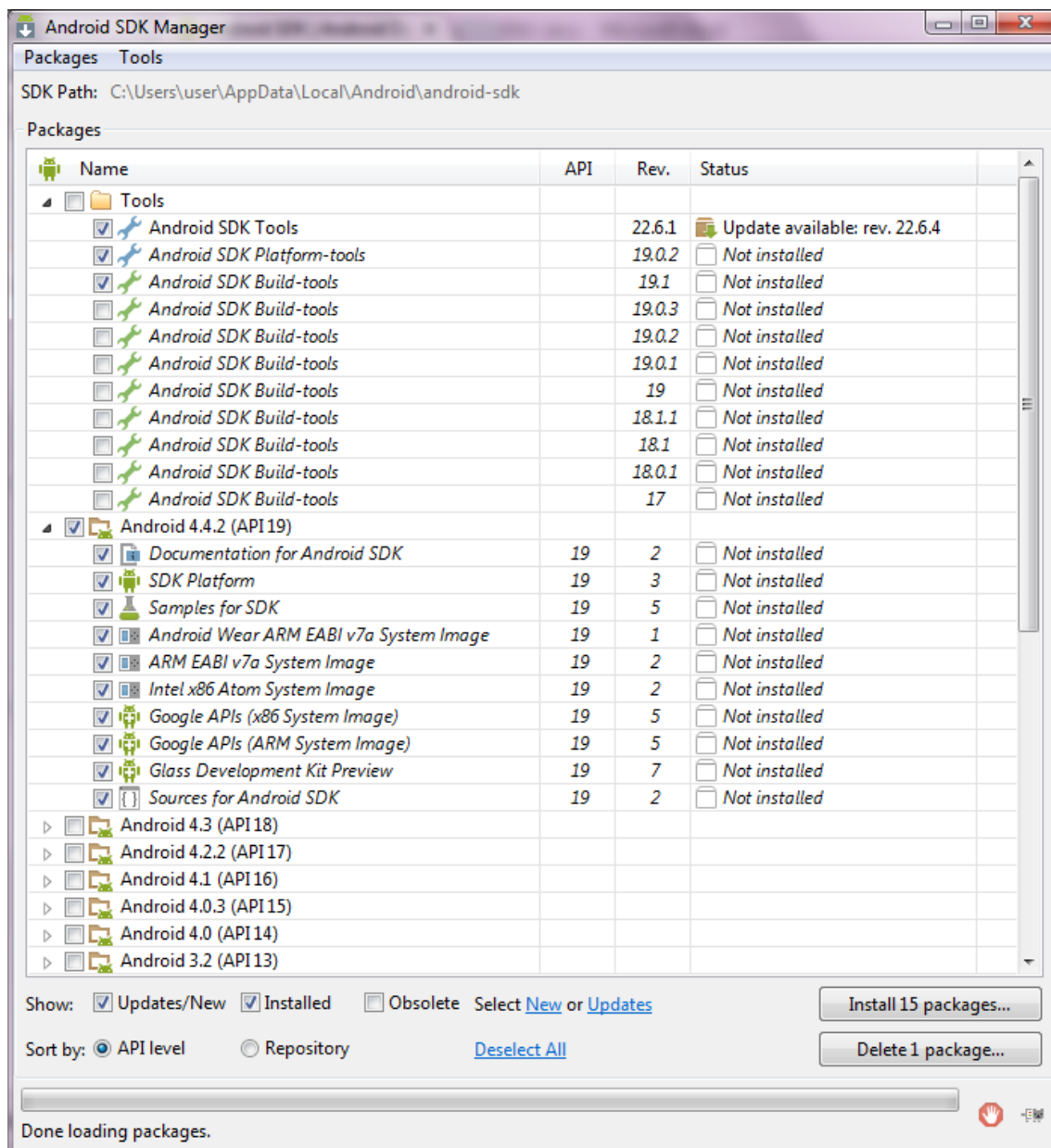
To Android SDK (Android Software Development Kit) είναι η επίσημη εργαλειοθήκη της Google για την δημιουργία εφαρμογών για Android συσκευές. Το Android SDK παρέχει την δυνατότητα της κατασκευής, του ελέγχου και του εντοπισμού σφαλμάτων. Έτσι, λοιπόν, μπορούμε να πούμε ότι κάποια βασικά εργαλεία που παρέχει το Android SDK είναι τα εξής:

- Εργαλεία debugging εφαρμογών
- Βιβλιοθήκες API
- Εξομοιωτής συσκευών (emulator)
- Documentation
- Δείγματα κωδικών
- Tutorials

Το Android SDK υποστηρίζει όλα τα λειτουργικά συστήματα και πολλά περιβάλλοντα ανάπτυξης κώδικα (IDE), συμπεριλαμβανομένου και την Basic4Android.

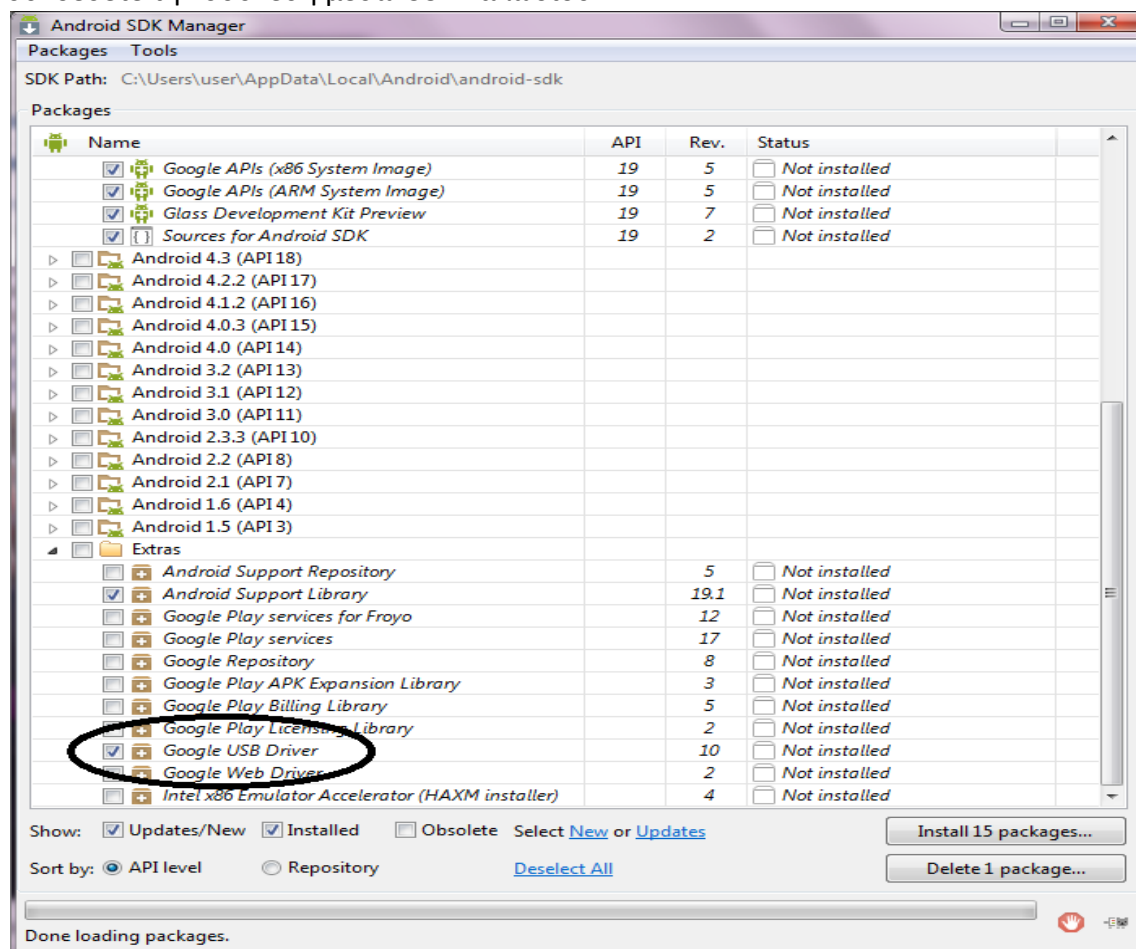
Η μεταφόρτωση του SDK μπορεί να γίνει είτε μέσω της επίσημης ιστοσελίδας του Android (<https://developer.android.com/sdk/installing/index.html>) ή για την πιο εύκολη και γρήγορη διευθέτηση του μέσω του συνδέσμου στην ιστοσελίδα της Anywhere Software (<http://www.basic4ppc.com/android/downloads.html>).

Για την Basic4Android πρέπει να εγκαταστήσουμε την πλατφόρμα με τα εργαλεία και τουλάχιστον μία πλατφόρμα με εικόνες. Μπορεί να χρησιμοποιηθεί το API 8 ή μεταγενέστερο.



Εγκατάσταση Android SDK Manager

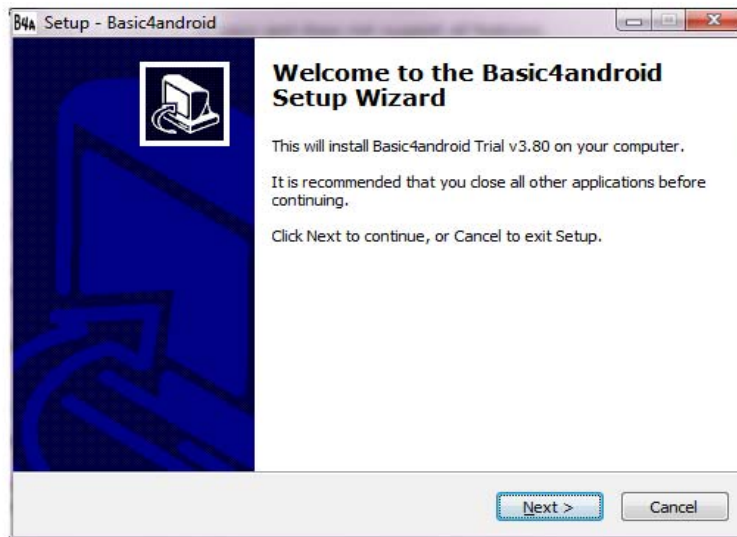
Επίσης, μπορούμε να εγκαταστήσουμε το Google USB driver, εφόσον θέλετε να συνδέσετε την συσκευή μέσω USB καλωδίου.



Εγκατάσταση Google USB Driver

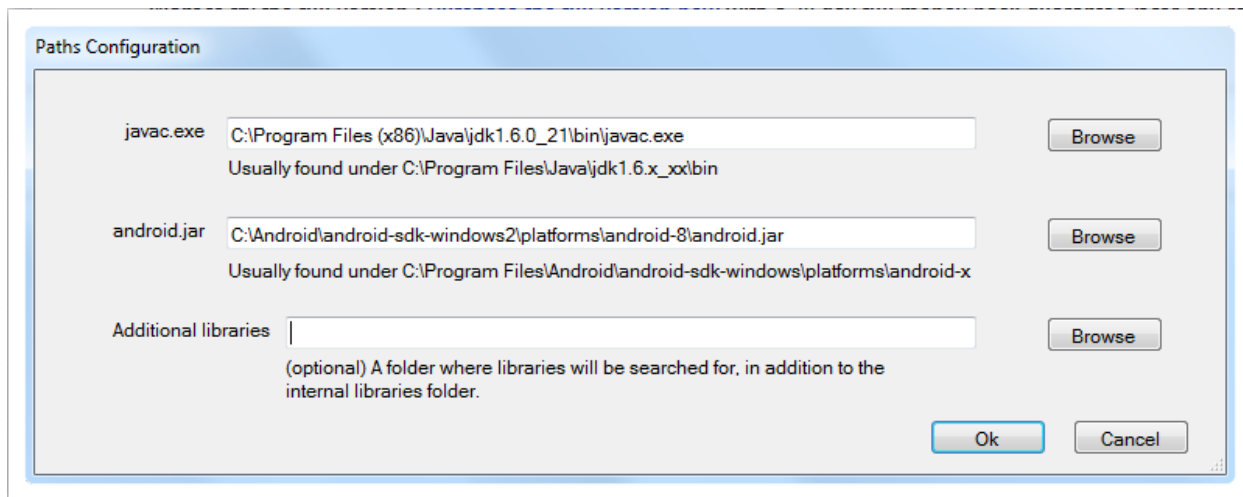
5.4 Basic4Android – Εγκατάσταση και διαμόρφωση

Με την ολοκλήρωση της εγκατάστασης των προαπαιτούμενων προγραμμάτων, θα πρέπει να κατεβάσουμε και να εγκαταστήσουμε το βασικό εργαλείο μας, το Basic4Android. Στην ιστοσελίδα της Anywhere Software δίνετε η δυνατότητα να κατεβάσει κάποιος την δοκιμαστική έκδοση ή να αγοράσει την ολοκληρωμένη έκδοση, η οποία μπορεί να προσφέρει οτιδήποτε χρειάζεται για την πλήρη ανάπτυξη μίας εφαρμογής και την πρόσβαση στη διαδικτυακή κοινότητα.



Εγκατάσταση Basic4Android

Μόλις αποκτήσουμε το IDE, απλά το “τρέχουμε” και στη συνέχεια θα πρέπει να ορίσουμε τα Paths. Ορίζουμε τα δύο πρώτα, μιας και το τρίτο είναι προαιρετικό.



Ορισμός Paths

Επιλέγουμε Configure Paths μέσα από το μενού Tools. Χρησιμοποιούμε το Browse, ώστε να τοποθετήσουμε τα αρχεία <javac.exe> και <android.jar>. Το αρχείο <javac.exe>

αποθηκεύεται στον φάκελο <android_sdk_windows\platforms\android_x>. Αυτό ήταν και το τελευταίο βήμα με τις εγκαταστάσεις και πλέον μπορούμε να ξεκινήσουμε με την κατασκευή μίας εφαρμογής.

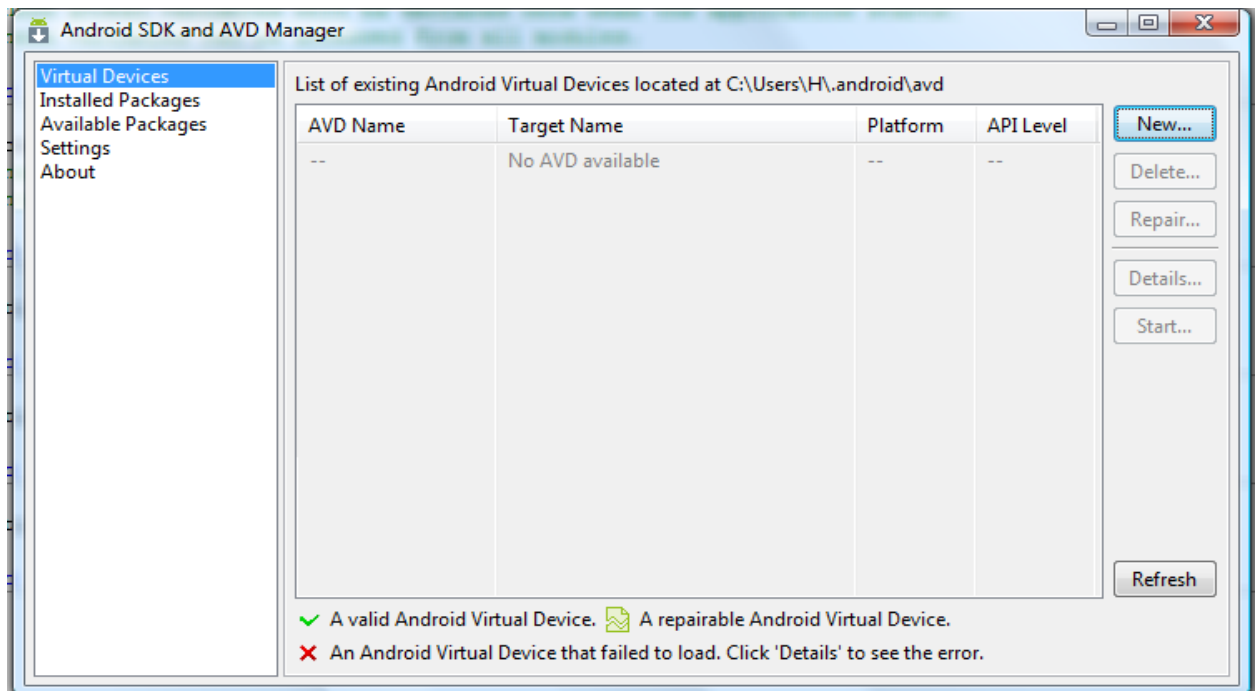
5.5 Android Virtual Device

Ένα θεμελιώδες κομμάτι της δημιουργίας μίας εφαρμογής είναι ο έλεγχος της λειτουργίας της. Ο έλεγχος του κώδικα και της εφαρμογής σε μία αληθινή συσκευή είναι πολύ σημαντικός, αλλά με την εξομοίωση κάθε πιθανού σεναρίου έχεις μεγαλύτερη ευκολία και παράλληλα προστατεύεις την συσκευή από πιθανές δυσλειτουργίες.

Το Android Virtual Device (AVD) είναι μία διάταξη εξομοίωσης, το οποίο σου επιτρέπει να διαμορφώσεις μία εικονική συσκευή με τον καθορισμό του υλικοτεχνικού εξοπλισμού και του λογισμικού ώστε να προσομοιάζουν σε μία αληθινή συσκευή. Ο εξομοιωτής παρέχεται μέσα στο Android SDK.

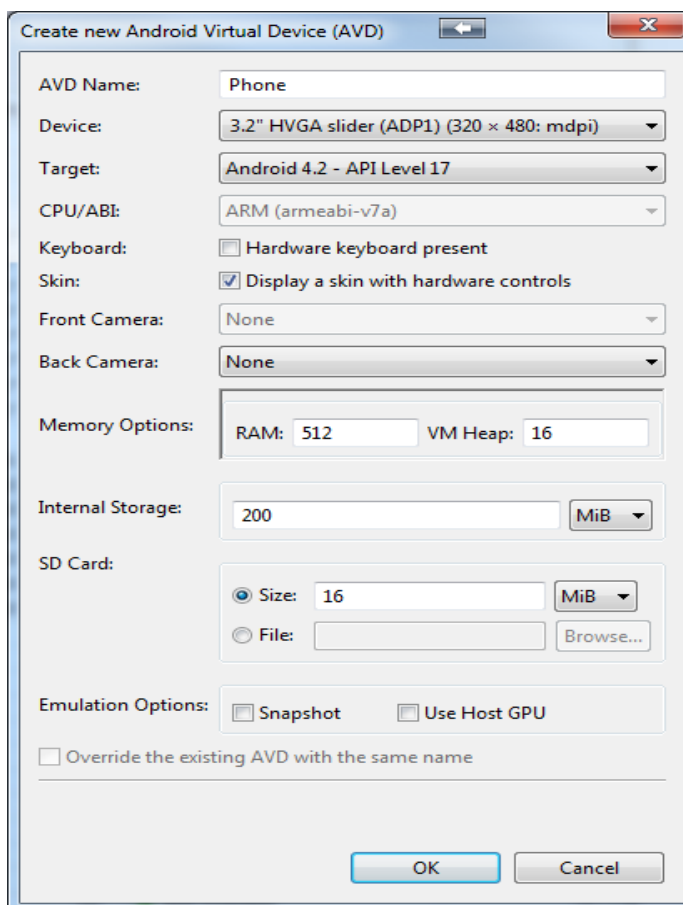
Για την δημιουργία ενός εξομοιωτή πρέπει να ακολουθήσουμε τα παρακάτω βήματα:

- a) Θέτουμε σε λειτουργία τη Basic4Android
- b) Από το μενού tools, επιλέγουμε το Run AVD Manager
- c) Στο παράθυρο Android SDK Manager, που αναδύεται, από το μενού tools επιλέγουμε το Manage AVDs
- d) Το AVD Manage θα εμφανίσει το παρακάτω παράθυρο



Δημιουργία εξομοιωτή

Επιλέγουμε “New” και συμπληρώνουμε το όνομα που θέλουμε να δώσουμε. Στο επόμενο πεδίο, στο Target ορίζουμε την πλατφόρμα που θέλουμε να δουλέψουμε.

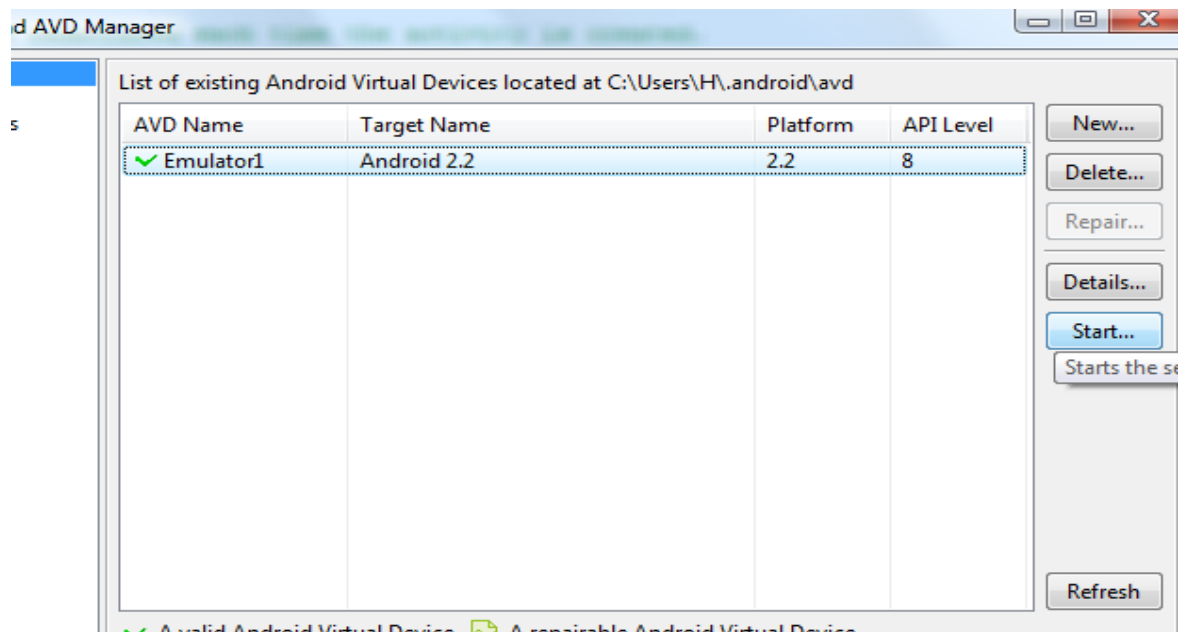


Δημιουργία εξομοιωτή

Επιλέγουμε, επίσης, και αν η συσκευή μας διαθέτει κάρτα μνήμης και την χωρητικότητα της.

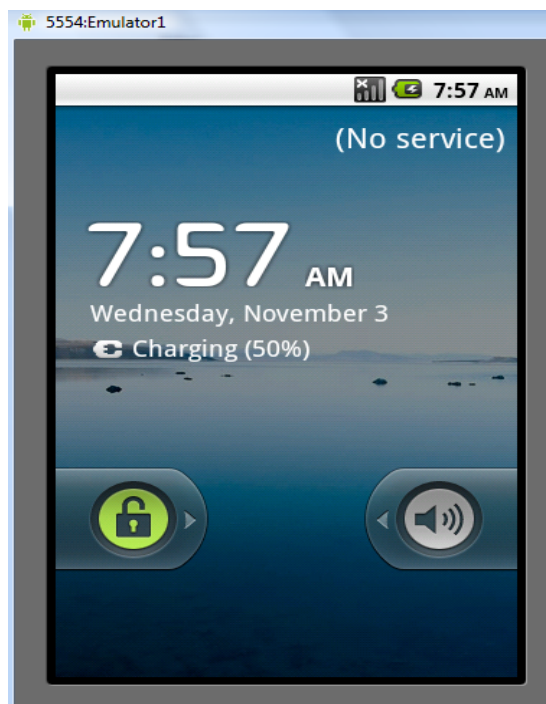
Με την επιλογή ενεργοποίησης του snapshot, έχουμε την δυνατότητα να αποθηκεύσουμε και να επαναφέρουμε ένα στιγμιότυπο μίας κατάστασης για κάθε ένα AVD, με σκοπό να αποφύγουμε την εκκίνηση όταν ξεκινάμε τον προσομοιωτή.

Στην περιοχή skin μπορούμε να επιλέξουμε προκαθορισμένες διαστάσεις οθόνης ή να ορίσουμε τα χαρακτηριστικά που θέλουμε να έχει η συσκευή μας, όπως π.χ. η μνήμη Ram, αν υποστηρίζει κάρτα μνήμης, αν διαθέτει κάμερα, αν μπορεί να αναπαράγει και να ηχογραφεί ήχους κ.α.



Απεικόνιση διαθέσιμων εξομοιωτών

Η συσκευή εξομοίωσης είναι έτοιμη. Με τον ίδιο τρόπο μπορούμε να δημιουργήσουμε επιπλέον AVD, οι οποίες μπορούν να έχουν διαφορετική ανάλυση οθόνης και χαρακτηριστικά. Για να θέσουμε σε λειτουργία κάποιον εξομοιωτή κλικάρουμε στο όνομα και μετά πατάμε start. Μετά την εκκίνηση του θα εμφανιστεί ένα παράθυρο, όπως η οθόνη μίας πραγματικής συσκευής.



Αρχική εικόνα μίας virtual device

5.6 Basic4Android – Bridge

Υπάρχουν πολλές επιλογές για την προώθηση μίας εφαρμογής όταν αυτή αναπτύσσεται μέσω της Basic4Android, όπως:

- 1) Ο εξομοιωτής
- 2) Η σύνδεση σε μία πραγματική συσκευή μέσω της λειτουργίας USB Debugging
- 3) Την σύνδεση σε μία πραγματική συσκευή μέσω της Basic4Android – Bridge

Πιο αναλυτικά:

Ο εξομοιωτής είναι πολύ αργός συγκριτικά με μία αληθινή συσκευή. Συνήθως η καλύτερη επιλογή είναι η σύνδεση μέσω USB Debugging. Για την σύνδεση αυτού του τύπου, θα πρέπει πρώτα να γίνει διαμόρφωση μίας πραγματικής συσκευής ώστε να υποστηρίζει τη λειτουργία USB Debugging. Επιπλέον, πρέπει να εγκατασταθεί το OEM USB Driver στη συσκευή. Δυστυχώς, όμως, πολλές συσκευές δεν υποστηρίζουν τη λειτουργία αυτή. Λύση έρχεται να μας δώσει η τρίτη επιλογή, η σύνδεση μέσω του Basic4Android – Bridge.

Το Basic4Android – Bridge είναι ένα εργαλείο για προγραμματιστές που χρησιμοποιούν την Basic4Android και επιτρέπει την σύνδεση του IDE με φυσικές συσκευές μέσω ασύρματου τοπικού δικτύου ή με μία σύνδεση Bluetooth. Το εργαλείο αυτό σχεδιάστηκε με την Basic4Android και ο πηγαίος κώδικας του είναι διαθέσιμος στην ενεργή κοινότητα στην ιστοσελίδα της εταιρίας. Το εργαλείο αυτό είναι διαθέσιμο στο Google Play και είναι δωρεάν.

Το Basic4Android – Bridge αποτελείται από δύο στοιχεία. Το ένα στοιχείο εκτελείται στη συσκευή και επιτρέπει στο δεύτερο, το οποίο αποτελεί μέρος του IDE για σύνδεση και επικοινωνία με τη συσκευή.

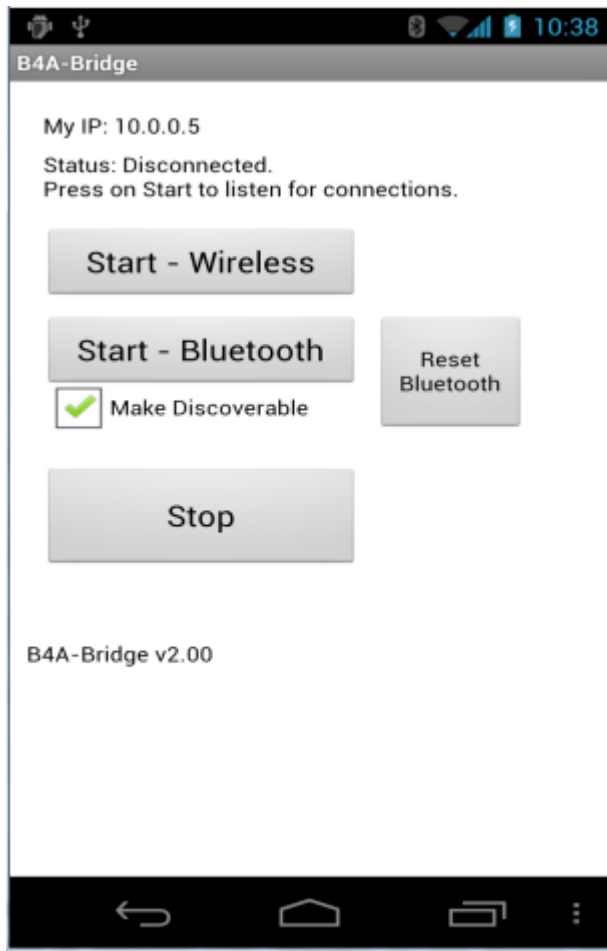
Εφόσον συνδεθεί το Basic4Android – Bridge υποστηρίζει όλα τα χαρακτηριστικά του IDE, που περιλαμβάνουν:

- Εγκατάσταση εφαρμογών
- Δυνατότητα να βλέπεις τα αρχεία καταγραφής (Logs)
- Debugging
- Τον οπτικό σχεδιαστή (Visual Designer)

Ξεκινώντας, εγκαθιστούμε το Basic4Android – Bridge στη συσκευή είτε μέσω του Google Play ή μεταφορτώνοντας το από την ιστοσελίδα

<http://basic4ppc.com/android/downloads.html> . Για να ολοκληρωθεί η διαδικασία

εγκατάστασης του, πρέπει να ενεργοποιηθεί στη συσκευή η δυνατότητα εγκατάστασης εφαρμογών από “άγνωστες πηγές” και η μόνη απαίτηση που έχει, είναι να υπάρχει διαθέσιμος εγγράψιμος χώρος στην κάρτα μνήμης. Με την λήψη της εφαρμογής και την εκκίνηση της, θα εμφανιστεί στην οθόνη το παρακάτω:



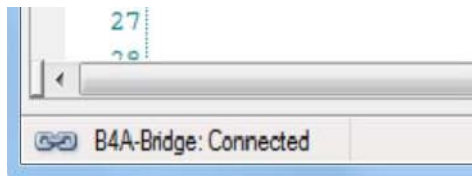
Εικόνα έναρξης της B4A-Bridge

Επιλέγουμε τον τρόπο με τον οποίο θα συνδέσουμε την συσκευή με το IDE. Οι επιλογές που εμφανίζονται είναι : “start wireless” και “start Bluetooth”. Το checkbox “make discoverable” θα επιτρέψει τη συσκευή σας να είναι ανιχνεύσιμη για πέντε (5) λεπτά, το οποίο χρειάζεται μόνο αν η συσκευή δεν έχει ξανασυνδεθεί με τον υπολογιστή. Από το IDE, πάμε στο μενού tools και επιλέγουμε είτε το “Connect wireless” ή το “Connect Bluetooth”, αντίστοιχα με την επιλογή στη συσκευή.

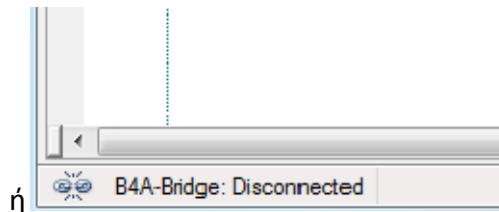
Αν επιλέξουμε την ασύρματη σύνδεση, στον Η/Υ θα εμφανιστεί ένα παράθυρο, όπου θα μας ζητηθεί να εισάγουμε την IP της συσκευής. Αυτή εμφανίζεται στην οθόνη της εφαρμογής ως My IP. Υπάρχει πιθανότητα να εμφανίσει τη διεύθυνση του δικτύου κινητής

τηλεφωνίας. Σε μία τέτοια περίπτωση μπορούμε να βρούμε την διεύθυνση του τοπικού δικτύου από τις προηγμένες ρυθμίσεις του δικτύου.

Αν επιλέξουμε σύνδεση μέσω Bluetooth, θα πρέπει πρώτα να κάνουμε αναζήτηση συσκευών στο παράθυρο που αναδύεται στην οθόνη αμέσως μετά την επιλογή μας, Connect Bluetooth από το μενού Tools – Basic4Android Bridge. Εμφανίζει μία λίστα συσκευών, οι οποίες είναι σε ανιχνεύσιμη κατάσταση. Από τη στιγμή που θα ολοκληρωθεί η σύνδεση θα εμφανιστεί στο κάτω μέρος της οθόνης, η τρέχουσα κατάσταση όπως στην παρακάτω εικόνα.



Απεικόνιση κατάστασης της B4A-Bridge



Και στους δύο τρόπους συνδεσμολογίας, η Basic4Android – Bridge μόλις συνδεθεί, ελέγχει αν υπάρχει κάποια αναβάθμιση στον designer και την πραγματοποιεί. Από την στιγμή που θα ενεργοποιηθεί, θα παραμείνει ανοιχτή μέχρι να ζητήσουμε την διακοπή της, πατώντας το stop. Για να δούμε την κατάσταση, μπορούμε να ανοίξουμε την οθόνη ειδοποιήσεων και πατώντας επάνω της, ανοίγει η κύρια οθόνη του εργαλείου.

6

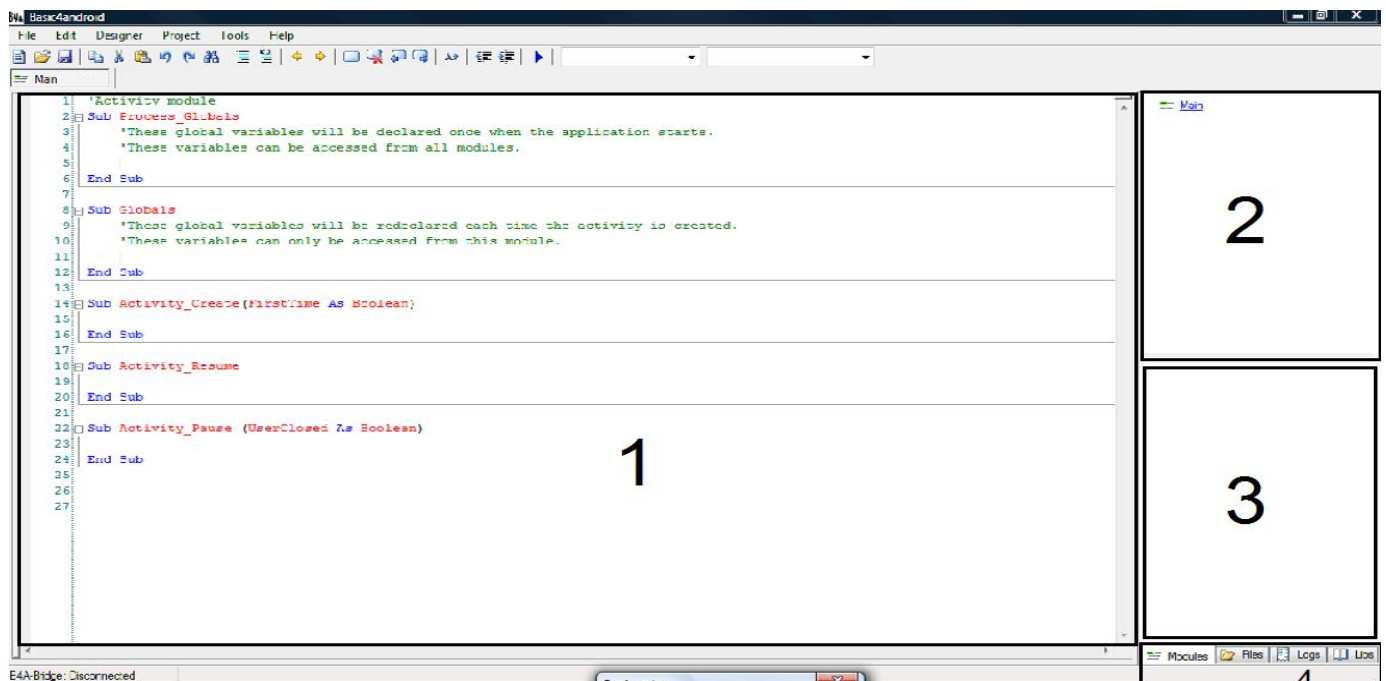
ΑΝΑΛΥΣΗ ΒΑΣΙΚΩΝ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ BASIC4ANDROID

6.1 Εξοικείωση με τη Basic4Android

Πρώτα απ' όλα, καλό θα είναι να γνωρίσουμε και να εξοικειωθούμε με το εργαλείο της Basic4Android.

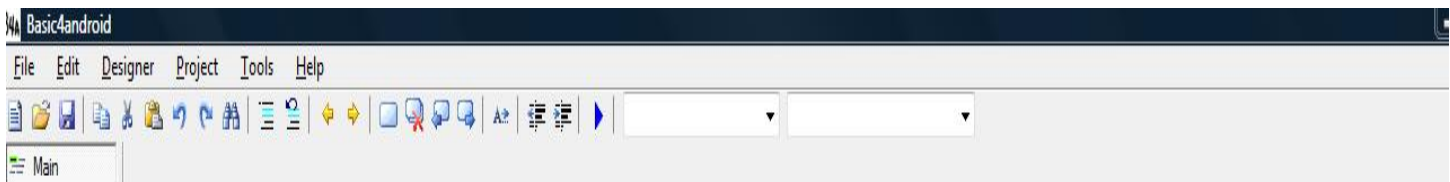
6.1.1 Πρώτη γνωριμία

Όταν ανοίγουμε το IDE, παίρνουμε μία μορφή όπως στην παρακάτω εικόνα. Υπάρχουν τέσσερις κύριες περιοχές:




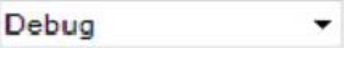

- Η περιοχή του κώδικα (code area). Είναι η περιοχή όπου συντάσσουμε τον κώδικα μας.
- Η λίστα των υπομονάδων (module list). Είναι μία λίστα, η οποία περιέχει τα ονόματα όλων των υπομονάδων.
- Η λίστα των υπορουτίνων (subroutine list). Η λίστα αυτή περιέχει όλες τις ρουτίνες στην υπομονάδα που έχει επιλεγεί.
- Οι καρτέλες (tabs). Είναι καρτέλες για διάφορες ρυθμίσεις, όπως η λίστα των βιβλιοθηκών, τα αρχεία καταγραφής, η λίστα των υπομονάδων και των υπορουτίνων και η λίστα των αρχείων που έχουν προστεθεί στο πρόγραμμα.

Επιπρόσθετα, στο πάνω μέρος της οθόνης όπως σε κάθε προγραμματιστικό εργαλείο υπάρχει το μενού που μας δίνει τις δυνατότητες, που μπορούμε να έχουμε αναπτύσσονται μία εφαρμογή. Καθώς επίσης υπάρχει η γραμμή εργαλείων για την πιο εύκολη και γρήγορη χρήση κάποιων βασικών στοιχείων και εντολών.

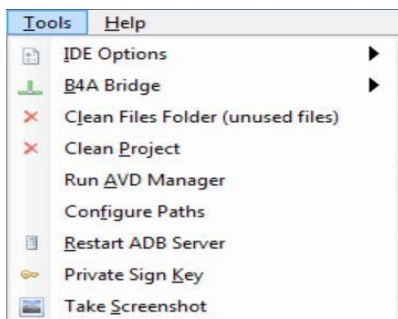


Γραμμή εργαλείων της Basic4Android

Μερικά βασικά κουμπιά είναι:

- Το  με το οποίο τρέχουμε τον compiler
- Το , όπου μας παρέχεται μία λίστα με τις επιλογές του compiler και του προγράμματος εντοπισμού σφαλμάτων (debugger)
- Το  είναι η λίστα με τις ρουτίνες

Αξίζει, επίσης, να σημειώσουμε ότι στο μενού tools, δίνονται οι επιλογές για τη σύνδεση με το Basic4Android – Bridge και την εκκίνηση του AVD Manager.



6.1.2 Προκαθορισμένος κώδικας (Default code)

Μόλις ανοίγουμε το πρόγραμμα διαπιστώνουμε ότι υπάρχει ήδη καταχωρημένος κάποιος κώδικας. Αυτός είναι ο προκαθορισμένος πίνακας κωδικών του Basic4Android καθώς, επίσης και το μέρος όπου θα καταχωρήσουμε τον κώδικα μας. Πιο αναλυτικά:

Sub Process_Globals

Η υπορουτίνα process_globals είναι μία προκαθορισμένη λειτουργία της Basic4Android, όπου δηλώνονται οι γενικές μεταβλητές. Οι μεταβλητές αυτές είναι ισχύουσες καθ' όλη τη διάρκεια της επεξεργασίας και προσβάσιμες οπουδήποτε στο πρόγραμμα.

```
1 'Activity module
2 Sub Process_Globals
3     'These global variables will be declared once when the application starts.
4     'These variables can be accessed from all modules.
5
6 End Sub
```

Sub Process_Globals

Sub Globals

Η ρουτίνα αυτή αφοσιώνεται στη δήλωση των μεταβλητών διαδικασίας. Οι μεταβλητές αυτές είναι ισχύουσες μόνο κατά την διάρκεια του συγκεκριμένου Activity και δεν είναι προσιτές από τις υπόλοιπες Activity του προγράμματος.

```
7
8 Sub Globals
9     'These global variables will be redeclared each time the activity is created.
10    'These variables can only be accessed from this module.
11
12 End Sub
13
```

Sub Globals

Sub Activity_Create (FirstTime As Boolean)

Στη συγκεκριμένη υπορουτίνα γράφουμε τον κώδικα που θέλουμε να εκτελείται πρώτος όταν “τρέχουμε” το πρόγραμμα μας. Στην περιοχή αυτή, επίσης, γίνονται και οι αρχικοποιήσεις των μεταβλητών, όπως αυτών του Process_Globals. Η παράμετρος της FirstTime, η οποία είναι αρχικοποιημένη ως Boolean είναι καθοριστική αν η Activity δημιουργείται για πρώτη φορά. Η παράμετρος FirstTime είναι χρήσιμη όταν γίνεται η αρχικοποίηση των μεταβλητών της διεργασίας.

```
13  
14 Sub Activity_Create(FirstTime As Boolean)  
15  
16 End Sub  
17
```

Sub Activity_Create

Sub Activity_Resume

Αυτή η ρουτίνα καλείται κάθε φορά που μία δραστηριότητα ξαναρχίζει, ενεργοποιείται εκ νέου. Σ’ αυτήν την περιοχή μπορούμε να ορίσουμε ενδεχόμενες παραμέτρους ρύθμισης για την νέα τρέχουσα δραστηριότητα.

```
17  
18 Sub Activity_Resume  
19  
20 End Sub  
21
```

Sub Activity_Resume

Sub Activity_Pause (UserClosed As Boolean)

Όταν μία δραστηριότητα ετοιμάζεται να παύσει τη λειτουργία της, τότε καλείται η ρουτίνα Activity_Pause. Σ’ αυτήν την περιοχή χρειάζεται να αποθηκευτούν οι παράμετροι δραστηριότητας που θα απαιτηθούν να ανακτηθούν όταν η δραστηριότητα ενεργοποιηθεί εκ νέου.

Ένας από τους λόγους που μπορεί μία δραστηριότητα να σταματήσει, είναι η αλλαγή κατεύθυνσης οθόνης. Σ’ αυτήν την περίπτωση, η τρέχουσα δραστηριότητα παύει και συνεχίζει με την νέα κατεύθυνση μέσω της ρουτίνας Activity_Resume. Συνήθως όμως συνεχίζεται μέσω των ρουτινών Globals, Activity_Create και Activity_Resume.

```

21
22 Sub Activity_Pause (UserClosed As Boolean)
23
24 End Sub
25
26

```

Sub Activity_Pause

Η μεταβλητή UserClosed υπάρχει για να διακρίνει εάν η δραστηριότητα σταμάτησε εξαιτίας του συστήματος, π.χ. αλλαγή κατεύθυνσης ή διακοπή από τον χρήστη, π.χ. κλικ στο κουμπί επιστροφής.

6.1.3 Ο Σχεδιαστής (Designer)

Η Basic4Android, όπως έχουμε αναφέρει, διαθέτει έναν οπτικό σχεδιαστή. Η χρήση του σχεδιαστή βοηθάει τον προγραμματιστή στη δημιουργία των διατάξεων του, έχοντας στη διάθεση του, τη δυνατότητα να ελέγχει κάθε φορά πως θα φαίνεται η διάταξη του σε μία αληθινή συσκευή με τη βοήθεια του Abstract Designer.



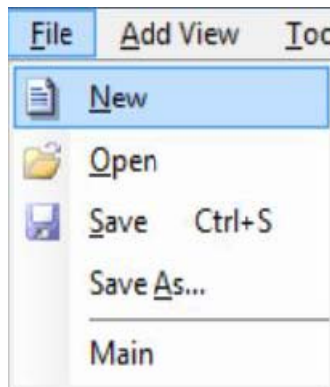
Απεικόνιση εφαρμογής στον εξομοιωτή και τον Abstract Designer

Θα ήταν πρόπον να δούμε λίγο πιο αναλυτικά πως μπορούμε να χρησιμοποιήσουμε το εργαλείο αυτό αλλά και ποιες δυνατότητες μπορεί να μας προσφέρει για την καλύτερη διαχείριση της εφαρμογής μας.

Ξεκινώντας, για να ανοίξουμε τον οπτικό σχεδιαστή, αρκεί από το αρχικό μενού της Basic4Android να κλικάρουμε την καρτέλα του Designer. Ανοίγουν δύο παράθυρα, ένα του Designer και ένα του Abstract Designer.

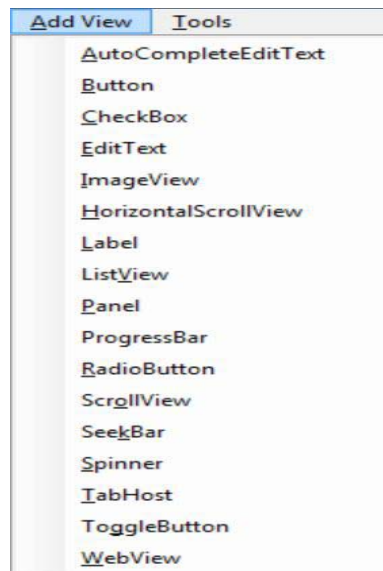
Στο μενού του Designer υπάρχουν τρεις επιλογές:

- 1) File menu. Από αυτό το μενού μπορούμε να δημιουργήσουμε, να ανοίξουμε, να



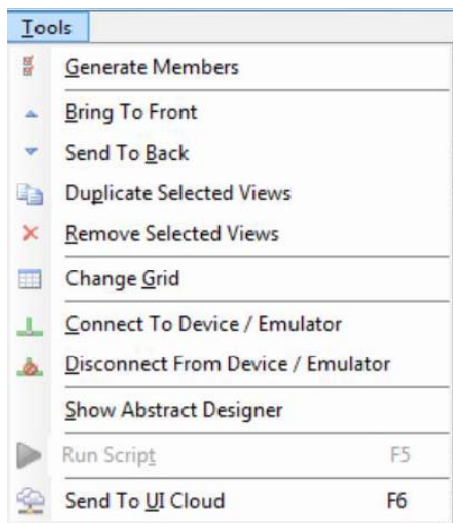
αποθηκεύσουμε μία διάταξη.

- 2) Add view menu. Αυτό το μενού μας επιτρέπει να επιλέξουμε το εικονίδιο που θέλουμε

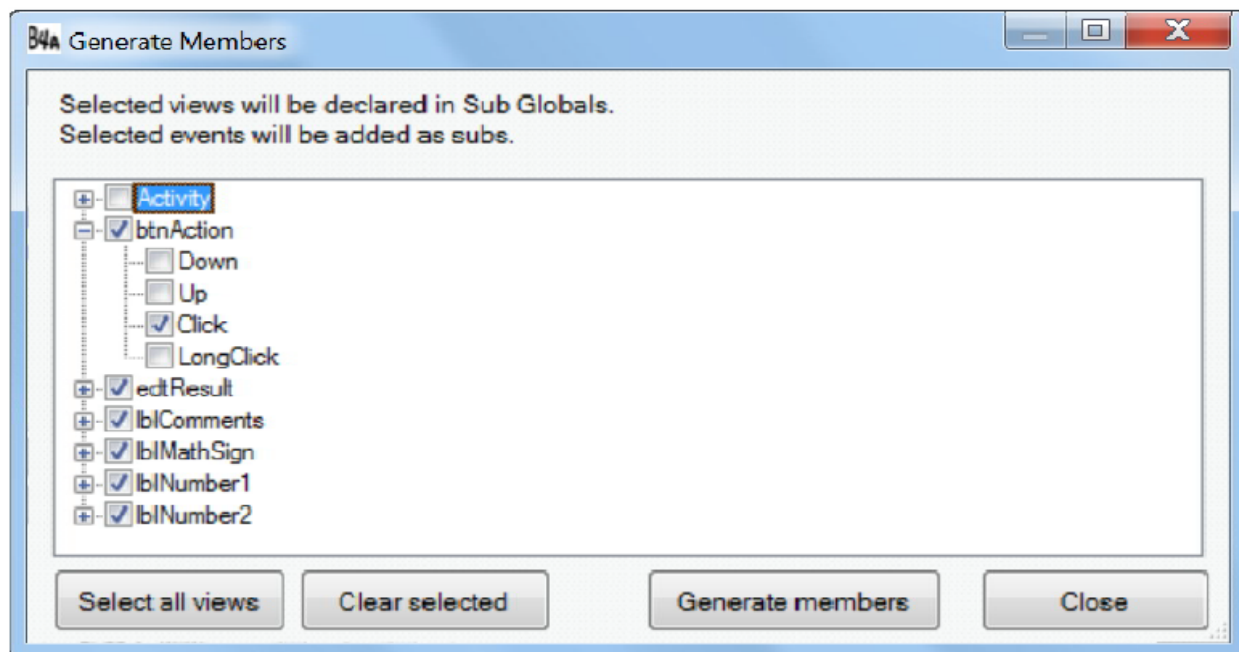


να προσθέσουμε στην τρέχουσα διάταξη.

3) Tools menu. Εδώ μας δίνονται όλες οι δυνατότητες για την ανάπτυξη των διατάξεων.



Το εργαλείο Generate Members, που βρίσκεται σ' αυτό το μενού επιτρέπει την αυτόματη δήλωση των στοιχείων που έχουμε επιλέξει και την δημιουργία των πλαισίων των υπορουτίνων. Κλικάροντας για να ανοίξει, λαμβάνουμε την παρακάτω εικόνα.



Εικόνα Generate Members

Στο σημείο αυτό εμφανίζονται όλα τα ενεργά εικονίδια (views) που έχουμε προσθέσει στη συγκεκριμένη διάταξη. Επιλέγοντας κάποιο από τα εικονίδια (views) δημιουργείται η

αναφορά του στην ρουτίνα Globals. Αυτό χρειάζεται για να γίνει το εικονίδιο αναγνωρίσιμο και να του επιτρέπεται η λειτουργία αυτόματης συμπλήρωσης.

```
Dim btnAction, btn0 As Button  
Dim lblResult As Label  
Dim lblComments As Label  
Dim lblMathSign As Label  
Dim lblNumber1 As Label  
Dim lblNumber2 As Label
```

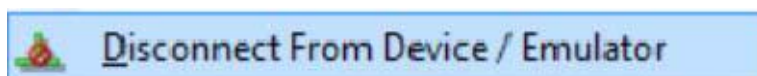
Αντίστοιχα, επιλέγοντας ένα εικονίδιο event, τότε δημιουργείται το πλαίσιο της υπορουτίνας του event.

```
Sub btnAction_Click  
  
End Sub
```

Στο μενού tools, επίσης, μας δίνεται η δυνατότητα να περάσουμε τη διάταξη μας σε μία συσκευή ή στον εξομοιωτή για να δούμε την πραγματική εικόνα του



Και αντίστοιχα να αποσυνδεθούμε.



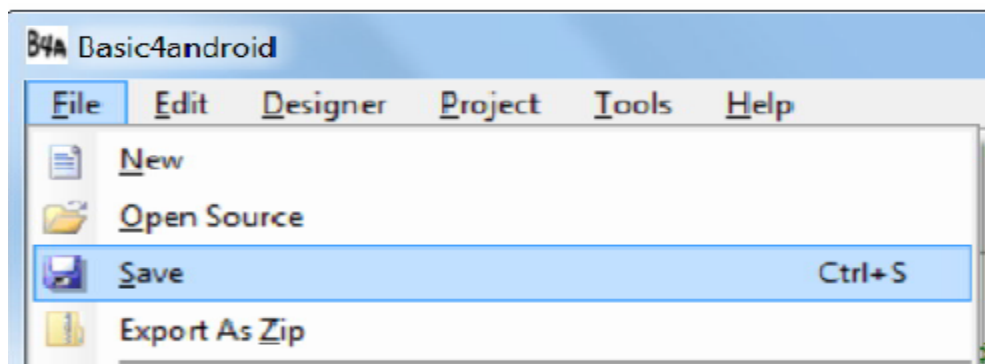
Επιπροσθέτως, έχουμε και την δυνατότητα της επιλογής του Sent To UI Cloud.

Όταν σχεδιάζουμε μία διάταξη στο Designer Scripts μπορούμε να στείλουμε το αρχείο της διάταξης στην ιστοσελίδα της Basic4Android και μας επιστρέφει μία σελίδα που δείχνει τη διάταξη μας σε διαφορετικές συσκευές με διαφορετικές οθόνες. Είναι ένα πολύ χρήσιμο εργαλείο για να τσεκάρουμε τη διάταξη μας χωρίς να χρειάζεται να έχουμε φυσικές συσκευές.

6.1.4 Παράδειγμα

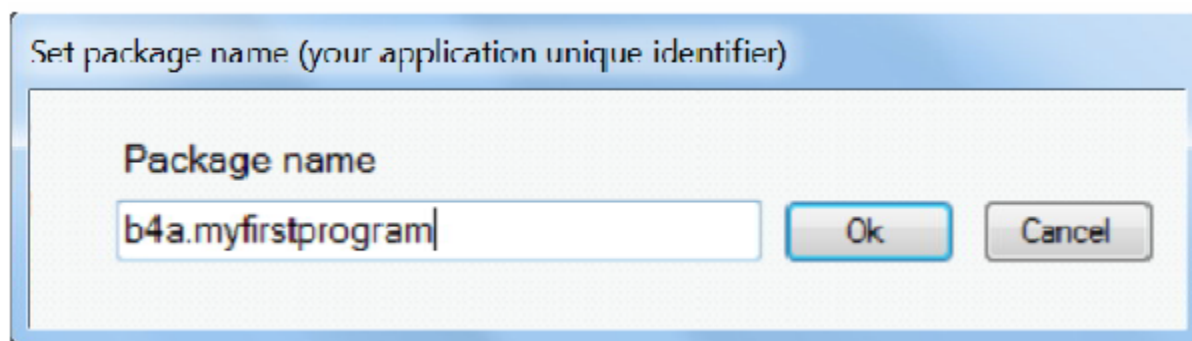
Για την κατανόηση των βασικών χαρακτηριστικών και των εργαλείων της Basic4Android, θα προβούμε στην δημιουργία μίας απλής εφαρμογής. Στη διάταξη αυτή θα επιλεγόνται τυχαία από το σύστημα, δύο αριθμοί και θα ζητείται να γράψουμε το αποτέλεσμα της πρόσθεσης τους. Μετά την εισαγωγή του αποτελέσματος στο αντίστοιχο εικονίδιο και την επιβεβαίωση της έγκυρης εισαγωγής από τον χρήστη, θα εμφανίζεται ένα μήνυμα στην οθόνη που θα μας ενημερώνει αν είναι η απάντηση σωστή ή λάθος. Ας ξεκινήσουμε λοιπόν:

Αρχικά ανοίγουμε το IDE και σώζουμε το project ώστε να μπορούμε να ανοίξουμε τον Designer.



Αποθήκευση project

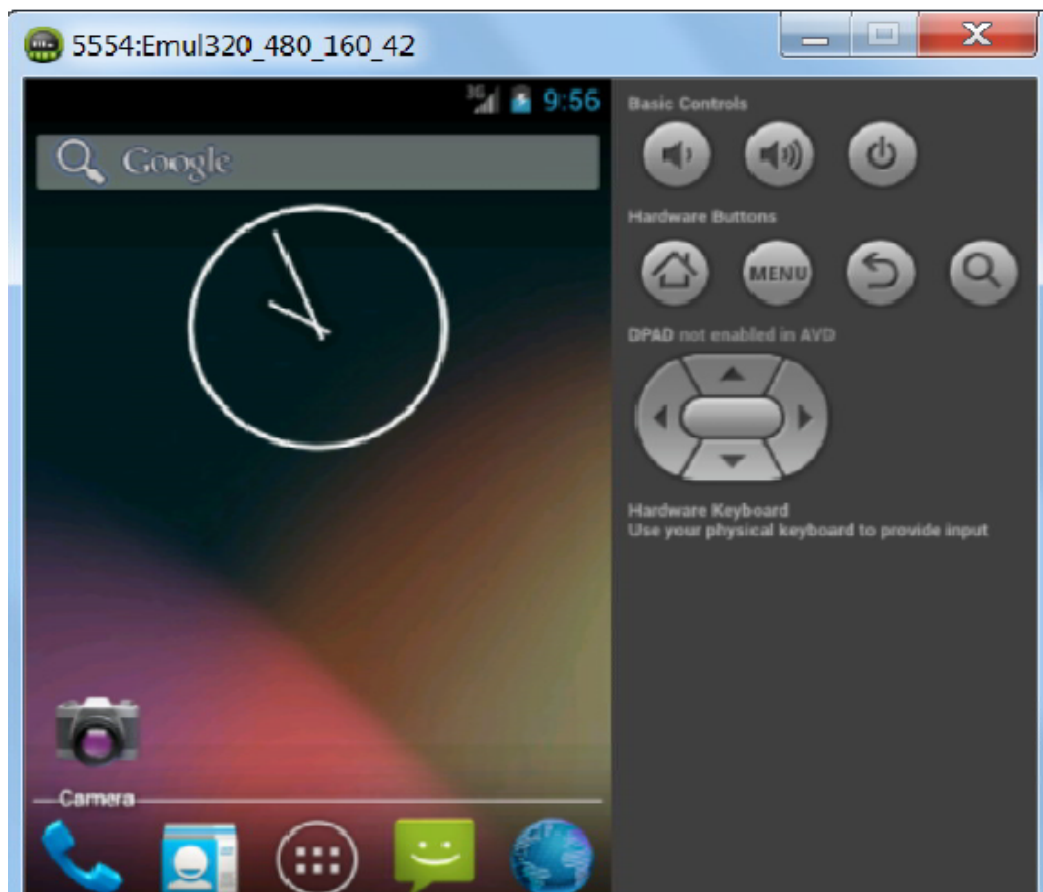
Το project το σώζουμε με το όνομα myfirstprogram. Ορίζουμε το package name του project από το μενού Project – Package name, μας εμφανίζει το παρακάτω παράθυρο



Ορίζουμε το package name του project

Αλλάζουμε το προκαθορισμένο όνομα με b4a.myfirstprogram. Στη συνέχεια ορίζουμε το Application label. Το Application label είναι το όνομα του προγράμματος, το οποίο θα εμφανίζεται στη συσκευή. Στο μενού Project – Application label γράφουμε το όνομα της εφαρμογής μας. Επίσης, από το μενού Project – Choose icon μπορούμε να ορίσουμε και την εικόνα που θα φαίνεται στην εφαρμογή μας.

Η επόμενη ενέργεια μας είναι να τρέξουμε τον AVD Manager από το μενού Tools, επιλέγουμε από την λίστα την εικονική συσκευή και πατάμε Start. Περιμένουμε λίγο μέχρι ο εξομοιωτής να είναι έτοιμος και να δούμε την παρακάτω εικόνα

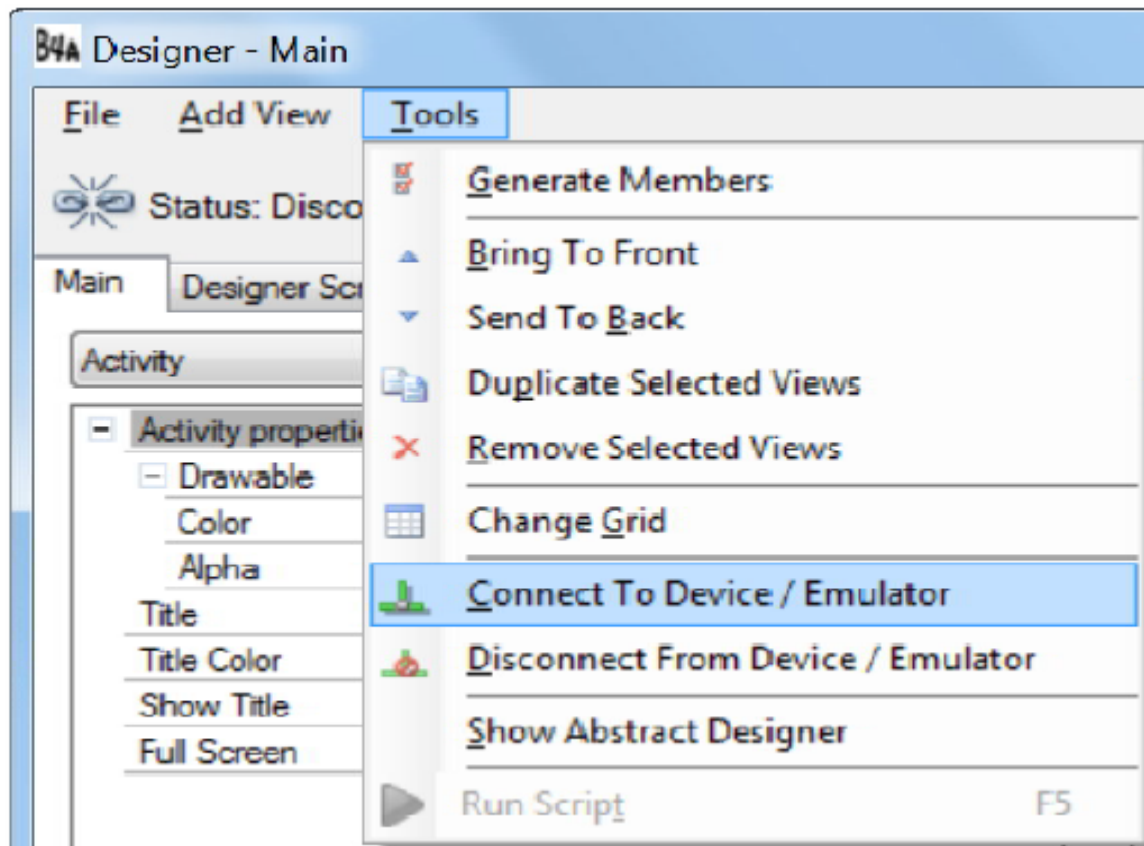


Οθόνη εξομοιωτή

Από το IDE ανοίγουμε τον Designer.

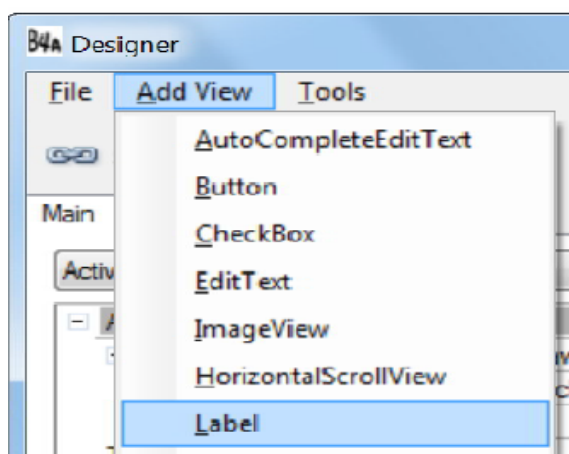


Συνδέουμε τον Designer με τον εξομοιωτή.

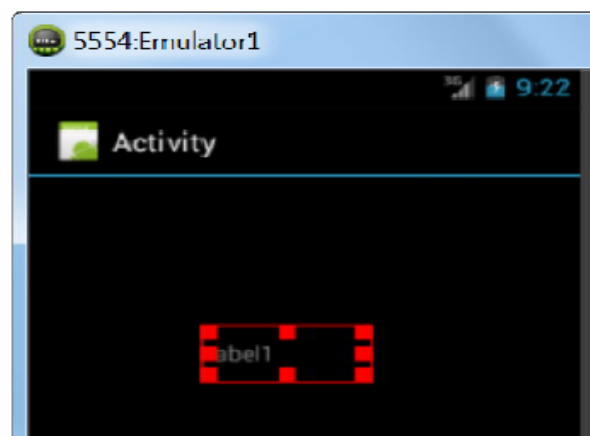


Σύνδεση Designer με εξομοιωτή

Πλέον είμαστε έτοιμοι να ξεκινήσουμε την σχεδίαση της διάταξης. Από το μενού Addview προσθέτουμε ένα label για τους αριθμούς, που θα προστίθενται.



Πρόσθεση label από τον Designer



Εμφάνιση label στον εξομοιωτή

Μέσα από το υπομενού που εμφανίζεται στον Designer για το label, αλλάζουμε την ονομασία του και τις διαστάσεις του. Το τελευταίο μπορούμε και έχουμε την δυνατότητα να το αλλάξουμε και μέσω του εξομοιωτή. Στον εξομοιωτή μπορούμε να αλλάξουμε και την θέση του εικονιδίου στην οθόνη.

Στη συνέχεια, χρειάζεται να προσθέσουμε ένα δεύτερο label πανομοιότυπο με το πρώτο, για τον δεύτερο αριθμό. Αντί να προσθέσουμε ένα καινούργιο μπορούμε να αντιγράψουμε το πρώτο. Αυτό θα το καταφέρουμε με την εντολή Duplicate Selected Views. Στο νέο εικονίδιο αλλάζουμε το όνομα από τις ιδιότητες του και την θέση του στον εξομοιωτή.

Με την ίδια λογική, δημιουργούμε και ένα τρίτο label, που αντιπροσωπεύει το μαθηματικό σύμβολο. Στο συγκεκριμένο label θα αλλάξουμε την ιδιότητα Text και θα προσθέσουμε το σύμβολο +.

Το εικονίδιο για το αποτέλεσμα θα το ορίσουμε με ένα EditText, το οποίο θα το προσθέσουμε και αυτό από το μενού Addview. Στο συγκεκριμένο εικονίδιο θα τροποποιήσουμε την επιλογή Input Type σε NUMBERS και στο Hint Text θα γράψουμε ENTER RESULT. Θέτοντας το Input Types σε NUMBERS, επιτρέπουμε στον χρήστη να εισάγει μόνο αριθμούς. Το Hint Text αντιπροσωπεύει το κείμενο που θα εμφανίζεται στο EditText, όσο ο χρήστης δεν εισάγει κάποιον αριθμό.

Εν συνεχεία, προσθέτουμε ένα button. Όταν ο χρήστης πατήσει το πλήκτρο αυτό είτε θα εκτελείτε ο έλεγχος του αποτελέσματος ή θα δημιουργεί ένα νέο μαθηματικό πρόβλημα. Στις ιδιότητες του button, πέρα από το όνομα θα αλλάξουμε και το Text με την λέξη OK.

Τέλος, θα προσθέσουμε ένα ακόμα label, το οποίο θα είναι για τα σχόλια που θα μας επιστρέφει το πρόγραμμα.

Σώζουμε τη διάταξη μας σε ένα αρχείο με την ονομασία main. Για να μπορέσουμε να γράψουμε τώρα τον κώδικα μας, πρέπει να δημιουργήσουμε τις αναφορές των ενεργών εικονιδίων που προσθέσαμε με τον σχεδιαστή. Αυτό γίνεται με την εντολή Generate Members. Τσεκάρουμε όλα τα εικονίδια και στο btnAction (Button) τσεκάρουμε το Click. Με αυτήν την επιλογή δημιουργείται η υπορουτίνα αυτού του γεγονότος.

Στην υπορουτίνα Globals, εμφανίζονται όλα τα δηλωμένα εικονίδια.

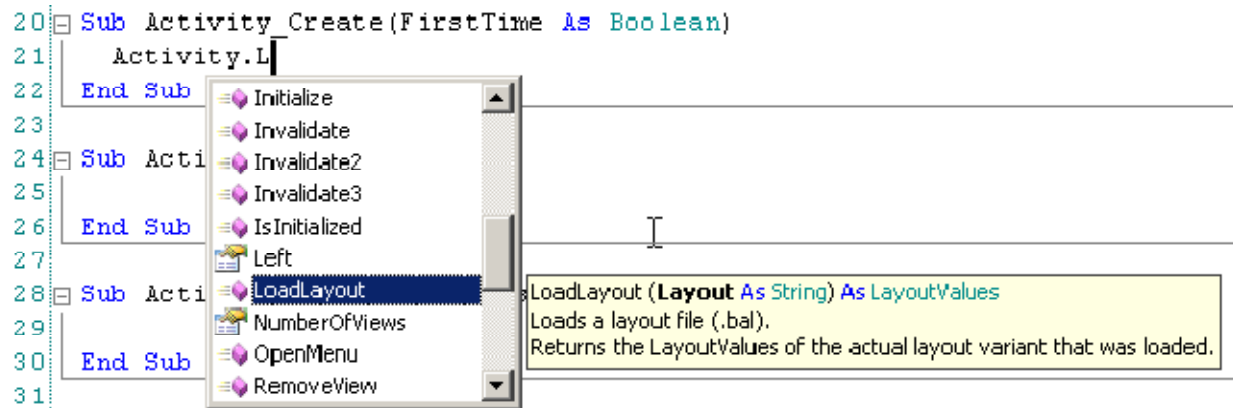
```
Dim btnAction As Button
Dim edtResult As EditText
Dim lblComments As Label
Dim lblMathSign As Label
Dim lblNumber1 As Label
Dim lblNumber2 As Label
```

Επίσης, έχει προστεθεί και μία ακόμα υπορουτίνα, αυτή του btn_Click.

```
Sub btnAction_Click  
  
End Sub
```

Είμαστε έτοιμοι να γράψουμε το πρόγραμμα μας.

Αρχικά, πρέπει να φορτώσουμε το αρχείο της διάταξης μας στο Sub Activity_Create.
Πληκτρολογούμε Activity.LoadLayout ("main")



Φόρτωση του αρχείου διάταξης

Με την λειτουργία της αυτόματης συμπλήρωσης εμφανίζεται η βοήθεια για τη μέθοδο της σύνταξης αλλά και πληροφορίες της κάθε εντολής. Στην ίδια υπορουτίνα και ακριβώς κάτω από την εντολή για την φόρτωση της διάταξης, γράφουμε New.

```
Sub Activity_Create(FirstTime As Boolean)  
'Do not forget to load the layout file created with the visual ...  
Activity.LoadLayout("Main")  
New  
End Sub
```

Μ' αυτόν τον τρόπο καλούμε μία άλλη υπορουτίνα, γράφοντας απλά το όνομα της. Στη νέα αυτή υπορουτίνα (Sub New) θα δημιουργήσουμε δύο αριθμούς που θα λαμβάνουν τιμές από το 1 έως το 10 (Number1, Number2). Η υπορουτίνα έχει ως εξής:

```
Sub New  
    Number1 = Rnd(1, 10) ' Generates a random number between 1 and 9  
    Number2 = Rnd(1, 10) ' Generates a random number between 1 and 9  
    lblNumber1.Text = Number1 ' Displays Number1 in label lblNumber1
```

```

lblNumber2.Text = Number2 ' Displays Number2 in label lblNumber2
lblComments.Text = "Enter the result" & CRLF & "and click on OK"
edtResult.Text = "" ' Sets edtResult.Text to empty
End Sub

```

Οι αριθμοί θα εμφανίζονται μέσα στα lblNumber1 και lblNumber2. Προϋπόθεση για να είναι αναγνωρίσιμοι αυτοί οι αριθμοί από το πρόγραμμα, θα πρέπει να τους δηλώσουμε ως μεταβλητές στο Sub Globals, με τον παρακάτω τρόπο:

```

Dim Number1, Number2 As Int

```

To lblCommentsText = " &CRLF & " αποτυπώνει στην οθόνη το σχόλιο " Enter result and click on OK", το οποίο βρίσκεται μέσα στο lblComment που δημιουργήσαμε στο Designer. Το CRLF είναι χαρακτήρας τροφοδότησης γραμμής.

Συνεχίζουμε με την δημιουργία του Button Click event, την υπορουτίνα Sub BtnAction_Click. Εδώ, έχουμε δύο περιπτώσεις:

- a) Όταν το κείμενο του Button είναι ισότιμο με το 'OK', αυτό σημαίνει ότι ένα νέο πρόβλημα προτείνεται και το πρόγραμμα περιμένει τον χρήστη να εισάγει ένα νέο αποτέλεσμα και στη συνέχεια να πατήσει το κουμπί.
- b) Όταν το κείμενο του είναι ισότιμο με το 'New', που σημαίνει ότι ο χρήστης έχει εισάγει την σωστή απάντηση και ο χρήστης μπορεί να πατήσει το κουμπί για να εισάγει ένα νέο πρόβλημα.

```

Sub btnAction_Click
    If btnAction.Text = "O K" Then
        If edtResult.Text="" Then
            MsgBox("No result entered", "E R R O R")
        Else
            CheckResult
        End If
    Else
        New
        btnAction.Text = "O K"
    End If
End Sub

```


Σ' αυτό το σημείο, αξίζει να δούμε, την εντολή if btnAction.text = 'OK' then ... με την οποία τσεκάρουμε αν το button είναι ίσο με το 'OK'. Εάν είναι ίσο, τότε τσεκάρουμε εάν το edtResult (EditText) είναι άδειο (" "). Εάν είναι τότε θα εμφανιστεί στην οθόνη ένα παράθυρο (MessageBox), το οποίο θα πληροφορεί τον χρήστη ότι δεν έχει εισαχθεί αποτέλεσμα στο εικονίδιο EditText. Εάν δεν είναι άδειο το EditText, τότε ελέγχουμε εάν το αποτέλεσμα είναι σωστό ή λάθος. Εάν το Button δεν είναι ίσο με το 'OK', τότε δημιουργούμε

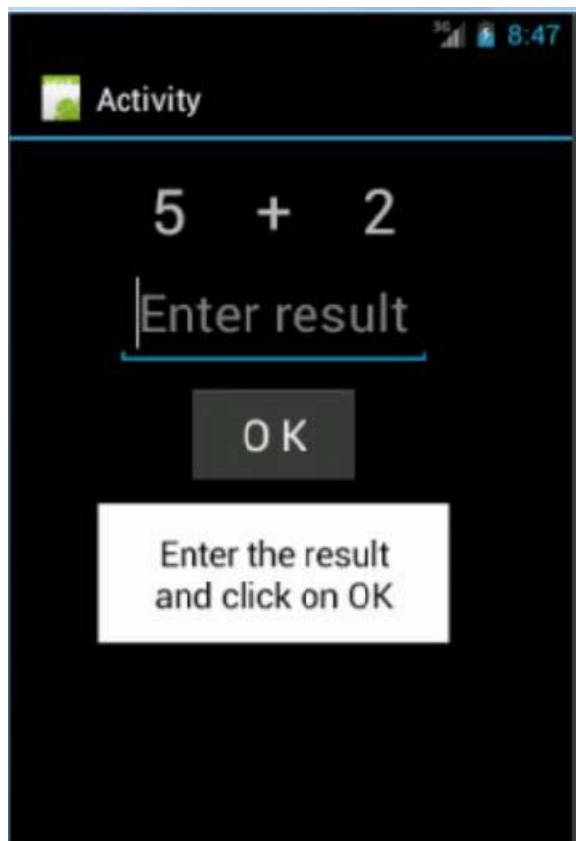
ένα νέο πρόβλημα κλικάροντας το κουμπί, που αναγράφει 'OK' και καθαρίζουμε το εικονίδιο EditText.

Η τελευταία ρουτίνα, που δημιουργούμε είναι για να ελέγξουμε το αποτέλεσμα, η Sub CheckResult.

```
Sub CheckResult
    If edtResult.Text = Number1 + Number2 Then
        lblComments.Text = "G O O D result" & CRLF & "Click on NEW"
        btnAction.Text = "N E W"
    Else
        lblComments.Text = "W R O N G result" & CRLF & "Enter a new
        result" & CRLF & "and click OK"
    End If
End Sub
```

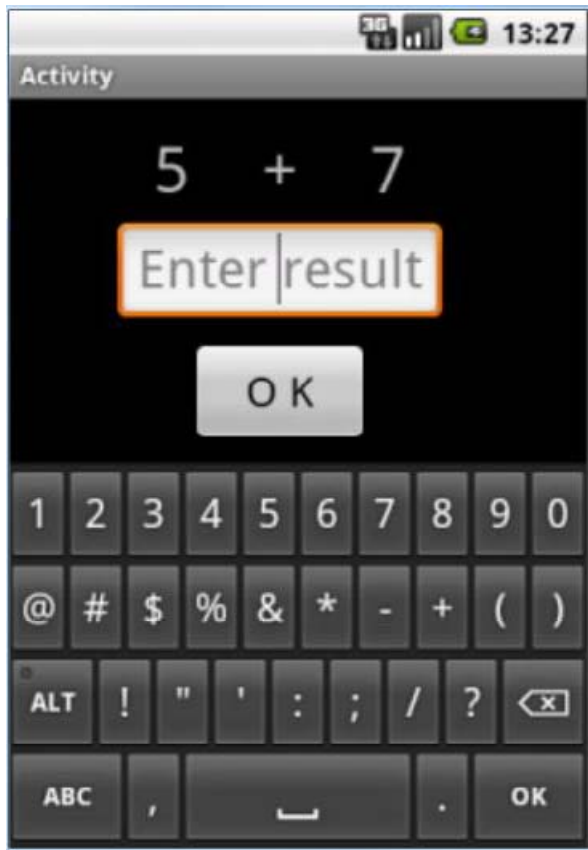
Με την εντολή if edtResult.Text = Number1 + Number2 then ... ελέγχουμε εάν το αποτέλεσμα είναι σωστό. Εάν είναι, τότε εμφανίζεται στην οθόνη, στο εικονίδιο lblComments το κείμενο "GOOD RESULT" & "Click on NEW" και το κείμενο του button αλλάζει σε "NEW". Εάν το αποτέλεσμα δεν είναι σωστό, τότε στο lblComments εμφανίζεται το μήνυμα "WRONG RESULT" & "Enter a new result" & "and click OK".

Με την ρουτίνα αυτή τελειώσαμε το πρόγραμμα. Τώρα θα το μεταγλωττίσουμε και θα το μεταφέρουμε στον εξομοιωτή για να το ελέγξουμε. Στο IDE πατάμε το []. Όταν ολοκληρωθεί η μεταγλώττιση θα έχουμε την παρακάτω εικόνα:



Εικόνα εξομοιωτή

Όταν κλικάρουμε στο εικονίδιο EditText, με το κείμενο “ Enter Result “ θα εμφανιστεί το εικονικό πληκτρολόγιο με τους αριθμούς.



Εικόνα εξομοιωτή

Το παράδειγμα δόθηκε ώστε να γίνουν κατανοητά τα βασικά χαρακτηριστικά της Basic4Android και του Designer. Στις επόμενες παραγράφους θα δούμε αναλυτικά μερικά χαρακτηριστικά που διαθέτουν οι συσκευές Android και μπορούν να χρησιμοποιηθούν σε εφαρμογές που δημιουργούνται με την Basic4Android.

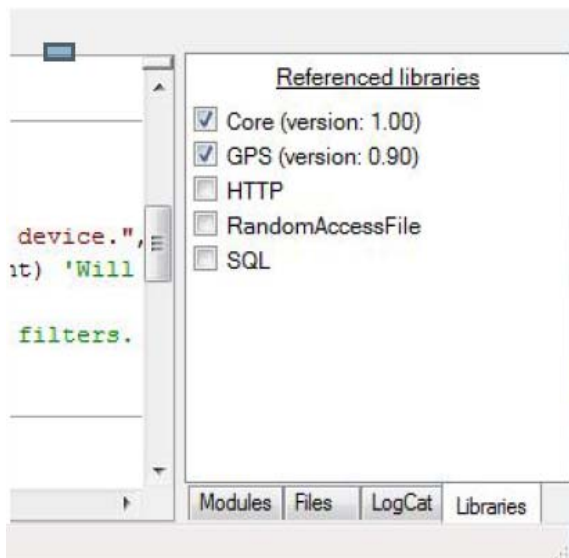
6.2 GPS

Το GPS είναι ένα πολύ σημαντικό χαρακτηριστικό για τις περισσότερες συσκευές. Σ' αυτήν την παράγραφο θα καλύψουμε μερικές βασικές λειτουργίες του GPS μέσω ενός απλού προγράμματος. Η εφαρμογή αυτή δείχνει την τρέχουσα θέση καθώς και την κατάσταση των δορυφόρων.

Αρχικά, πρέπει να αναφέρουμε ότι η λειτουργικότητα του GPS συμπεριλαμβάνεται στην βιβλιοθήκη του GPS, η οποία είναι μέρος της βασικής γλώσσας της Basic4Android. Στην βιβλιοθήκη αυτή υπάρχουν τρεις τύποι του σχετικού αντικειμένου:

- GPS. Το GPS διαχειρίζεται τη σύνδεση και τα γεγονότα (Events).
- Location. Το Location είναι η δομή που κρατάει τα στοιχεία διαθέσιμα που σχετίζονται με μία συγκεκριμένη 'θέση'. Αυτά τα στοιχεία περιλαμβάνουν τις συντεταγμένες του γεωγραφικού πλάτους και μήκους της συγκεκριμένης 'θέσης' και άλλες πληροφορίες, όπως τον καθορισμό της θέσης, το υψόμετρο και άλλα.
Συνήθως με τα αντικείμενα του Location εργαζόμαστε σε events GPS_LocationChanged. Ωστόσο, μπορούμε να αρχικοποιήσουμε τα αντικείμενα αυτά μόνοι μας. Αυτό μας είναι αρκετά χρήσιμο για τον υπολογισμό της απόστασης και τον καθορισμό μίας θέσης.
- GPSSatellite. Το GPSSatellite είναι μία δομή που συγκρατεί διάφορες πληροφορίες που αφορούν τους τρέχοντες γνωστούς δορυφόρους.

Ας ξεκινήσουμε με το πρόγραμμα μας. Το πρώτο πράγμα που έχουμε να κάνουμε είναι να προσθέσουμε τη βιβλιοθήκη του GPS στη λίστα με τις βιβλιοθήκες που χρησιμοποιούμε.



Προσθήκη βιβλιοθήκης GPS

Στον κώδικα μας ξεκινάμε την δήλωση του αντικειμένου το GPS στη ρουτίνα Process_Globals. Εν εναντία περιπτώσει, νέα τεκμήρια θα λαμβάνουν χώρα κάθε φορά που η δραστηριότητα θα αναδημιουργείται.

```
Sub Process_Globals
    Dim GPS1 As GPS
End Sub
```

Εν συνεχεία, αρχικοποιούμε το GPS με την παράμετρο του eventName του (Initialize (eventName As String))

```
Sub Activity_Create (FirstTime As Boolean)
    If FirstTime Then
        GPS1.Initialize ("GPS")
    End If
    Activity.LoadLayout ("1")
End Sub
```

Επόμενο βήμα είναι να δώσουμε το έναυσμα να ξεκινήσει το GPS να λαμβάνει δεδομένα. Αυτό συνίσταται να γίνει στη ρουτίνα Activity_Resume. Λόγω του ότι το GPS καταναλώνει αρκετή μπαταρία, συνίσταται να σταματάμε την χρήση του όταν δεν είναι απαραίτητο. Ως εκ τούτου μπορούμε να σταματήσουμε το GPS στη ρουτίνα Activity_Pause.

```
Sub Activity_Pause (UserClosed As Boolean)
    GPS1.Stop
End Sub
```

Το GPS δεν μπορεί να ενεργοποιηθεί προγραμματιστικά. Το Android OS, λόγω της ιδιωτικότητας του κάθε χρήστη δεν το επιτρέπει. Το καλύτερο που μπορούμε να κάνουμε είναι να ζητάμε από τον χρήστη να ενεργοποιεί τη συσκευή του GPS. Στο παρακάτω κομμάτι κώδικα εμφανίζεται ένα μήνυμα που ζητάμε από τον χρήστη να ενεργοποιήσει το GPS εάν αυτό δεν είναι ήδη ενεργοποιημένο. Επίσης, ανοίγει τον πίνακα ελέγχου του GPS, έτσι ώστε ο χρήστης να χρειάζεται μόνο να επιβεβαιώσει την επιλογή αυτή.

```
Sub Activity_Resume
    If GPS1.GPSEnabled = False Then
        ToastMessageShow ("Please enable the GPS device.", True)
        StartActivity (GPS1.LocationSettingsIntent) 'Will open the relevant settings screen.
    Else
        GPS1.Start (0, 0) 'Listen to GPS with no filters.
    End If
End Sub
```

Εάν το GPS είναι ενεργό, τότε ξεκινάει να λαμβάνει δεδομένα. Το Start λαμβάνει δύο τιμές (Start (Minimum Time As Long, Minimum Distance As Float)). Η πρώτη τιμή αφορά την μικρότερη περίοδο μεταξύ των γεγονότων (μετράει σε millisecond) και η δεύτερη τιμή αφορά την μικρότερη μεταβολή της απόστασης στην οποία προκλήθηκαν τα γεγονότα (μετράνε σε μέτρα). Με την τιμή 0 και στα δύο επωφελούμαστε την υψηλότερη συχνότητα και ως εκ τούτου θα λάβουμε όλα τα γεγονότα των θέσεων

Το GPS υποκινεί τρία γεγονότα:

- GPS_LocationChanged (Location1 As Location). Αυτό είναι και το κύριο event. Το Location1 κρατά τα στοιχεία της νέας θέσης.
- GPS_GpsStatus (Satellites As List). Αυτό το γεγονός επιτρέπει να εμφανίσουμε πληροφορίες σχετικά με τους διαθέσιμους δορυφόρους τη δεδομένη στιγμή.

Σημειωτέο είναι το γεγονός ότι δεν χρησιμοποιούνται όλοι οι δορυφόροι της λίστας γι' αυτόν υπολογισμό της τελευταίας θέσης. Σύμφωνα με αυτό είναι δυνατόν να περιέχονται αρκετοί δορυφόροι αλλά η λήψη δεν θα είναι αρκετά καλή για τον εντοπισμό της θέσης.

- `GPS_UserEnabled (Enabled As Boolean)`. Αυτό το γεγονός προκαλείτε κάθε φορά που ο χρήστης αλλάζει την κατάσταση της συσκευής του GPS. Επίσης, προκαλείται αμέσως μετά το κάλεσμα της εντολής "Start".

Παρακάτω βρίσκεται ο κώδικας του παραδείγματος, όπου εμφανίζονται τα γεγονότα και οι ρουτίνες που αναφέραμε παραπάνω

```
Sub Process_Globals
    Dim GPS1 As GPS
End Sub

Sub Globals
    Dim lblLon As Label
    Dim lblLat As Label
    Dim lblSpeed As Label
    Dim lblSatellites As Label
End Sub

Sub Activity_Create(FirstTime As Boolean)
    If FirstTime Then
        GPS1.Initialize("GPS")
    End If
    Activity.LoadLayout("1")
End Sub

Sub Activity_Resume
    If GPS1.GPSEnabled = False Then
        ToastMessageShow("Please enable the GPS device.", True)
        StartActivity(GPS1.LocationSettingsIntent) 'Will open the relevant settings screen.
    Else
        GPS1.Start(0, 0) 'Listen to GPS with no filters.
    End If
End Sub

Sub Activity_Pause (UserClosed As Boolean)
    GPS1.Stop
End Sub

Sub GPS_LocationChanged (Location1 As Location)
    lblLat.Text = "Lat = " & Location1.ConvertToMinutes(Location1.Latitude)
    lblLon.Text = "Lon = " & Location1.ConvertToMinutes(Location1.Longitude)
```

```

        lblSpeed.Text = "Speed = " & Location1.Speed
End Sub

Sub GPS_UserEnabled (Enabled As Boolean)
    ToastMessageShow("GPS device enabled = " & Enabled, True)
End Sub

Sub GPS_GpsStatus (Satellites As List)
    lblSatellites.Text = "Satellites:" & CRLF
    For i = 0 To Satellites.Size - 1
        Dim Satellite As GPSSatellite
        Satellite = Satellites.Get(i)
        lblSatellites.Text = lblSatellites.Text & CRLF & Satellite.Prn & _
            " " & Satellite.Snr & " " & Satellite.UsedInFix & " " & Satellite
            .Azimuth _
            & " " & Satellite.Elevation
    Next
End Sub

```

6.3 HTTP – Υπηρεσίες Web

Σ' αυτήν την παράγραφο θα δούμε μερικά στοιχεία για την σύνδεση μίας διαδικτυακής υπηρεσίας με μία εφαρμογή Android. Ο καλύτερος τρόπος κατανόησης είναι να δώσουμε και εδώ ένα παράδειγμα. Το παράδειγμα μας είναι μία εφαρμογή που κάνει μετατροπή νομισμάτων μέσω μίας υπηρεσίας web. Συγκεκριμένα μέσω της υπηρεσίας <http://www.webserviceX.net/currencyconvertor/>

Ολόκληρος ο κώδικας της εφαρμογής βρίσκεται παρακάτω:

```

'Activity module
Sub Process_Globals
    Dim countries As List
    Dim URL As String
    URL = "http://www.webserviceX.net/CurrencyConvertor.asmx/ConversionRate?FromCurrency="
    Dim HttpClient1 As HttpClient
    Type StateType (TextUp As String, TextDown As String, _
        IndexUp As Int, IndexDown As Int)
    Dim State As StateType 'This must be a process variable as it stores the
        state
                                'and should not be released when the activity is
        destroyed.
End Sub

Sub Globals
    Dim txtUp, txtDown As EditText
    Dim spnrUp, spnrDown As Spinner
    Dim btnUp, btnDown As Button
    Type MyTag (FromValue As EditText, ToValue As EditText, _

```

```

        FromCurrency As Spinner, ToCurrency As Spinner)
    Dim CurrentTask As MyTag
End Sub
Sub ResetState
    'set the starting state
    State.TextUp = 1
    State.TextDown = ""
    State.IndexUp = 0 'USD
    State.IndexDown = 43 'Euro
End Sub

Sub Activity_Create(FirstTime As Boolean)
    If FirstTime Then
        Log("*****") 'load the list of countries
        countries = File.ReadList(File.DirAssets, "CountryCodes.txt")
        'initialize the HttpClient object which is responsible for all commun
ication.
        HttpClient1.Initialize("HttpClient1")
        ResetState
    End If

    Activity.LoadLayout("layout1")
    spnrUp.AddAll(countries)
    spnrDown.AddAll(countries)

    Dim t1 As MyTag
    t1.FromValue = txtUp
    t1.ToValue = txtDown
    t1.FromCurrency = spnrUp
    t1.ToCurrency = spnrDown
    btnDown.Tag = t1

    Dim t2 As MyTag
    t2.FromValue = txtDown
    t2.ToValue = txtUp
    t2.FromCurrency = spnrDown
    t2.ToCurrency = spnrUp
    btnUp.Tag = t2
End Sub

Sub Activity_Resume
    txtUp.Text = State.TextUp
    txtDown.Text = State.TextDown
    spnrUp.SelectedIndex = State.IndexUp
    spnrDown.SelectedIndex = State.IndexDown
End Sub

Sub Activity_Pause (UserClosed As Boolean)
    If UserClosed Then
        ResetState 'reset the state to the initial settings.
    Else
        State.TextUp = txtUp.Text
        State.TextDown = txtDown.Text
        State.IndexUp = spnrUp.SelectedIndex
        state.IndexDown = spnrDown.SelectedIndex
    End If
End Sub

```

```

End Sub
Sub btn_Click
    Dim btn As Button
    btn = Sender 'Fetch the actual button that raised this event.
    CurrentTask = btn.Tag 'Take the object from its Tag property.
    Dim fromCountry, toCountry As String
    fromCountry = CurrentTask.FromCurrency.SelectedItem.SubString2(0, 3) 'get
the currency code
    toCountry = CurrentTask.ToCurrency.SelectedItem.SubString2(0, 3)

    Dim request As HttpRequest
    request.InitializeGet(URL & fromCountry & "&ToCurrency=" & toCountry)
    request.Timeout = 10000 'set timeout to 10 seconds
    If HttpClient1.Execute(request, 1) = False Then Return 'Will be false if
their is already a running task (with the same id).
    ProgressDialogShow("Calling server...")
End Sub
Sub HttpClient1_ResponseSuccess (Response As HttpResponse, TaskId As Int)
    Log("ResponseSuccess")
    ProgressDialogHide
    Dim result As String
    result = Response.GetString("UTF8") 'Convert the response to a string
    Log(result)
    Dim rate As Double
    'Parse the result
    i = result.IndexOf(".NET/")
    If i = -1 Then
        MsgBox("Invalid response.", "Error")
        Return
    End If
    i2 = result.IndexOf2("<", i + 1)
    rate = result.substring2(i + 7, i2)
    Log("Rate = " & rate)
    If IsNumber(CurrentTask.FromValue.Text) = False Then
        MsgBox("Please enter a valid number.", "Error")
        Return
    End If
    'Set the answer

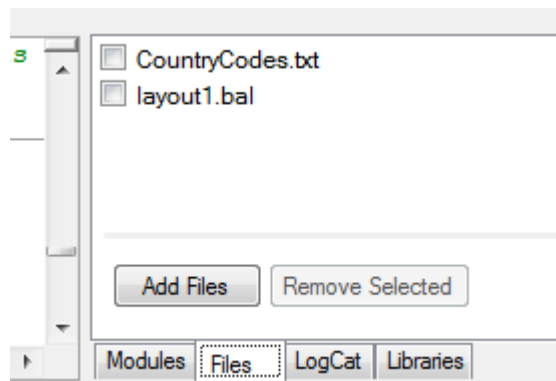
    CurrentTask.ToValue.Text = Round2(CurrentTask.FromValue.Text * rate, 2)
End Sub
Sub HttpClient1_ResponseError (Reason As String, StatusCode As Int, TaskId As
Int)
    Log(Reason)
    Log(StatusCode)
    ProgressDialogHide
    msg = "Error connecting to server."
    If reason <> Null Then msg = msg & CRLF & Reason
    ToastMessageShow (msg, True)
End Sub

```

Ας αναλύσουμε τον κώδικα και να δούμε αναλυτικά τις πτυχές που αφορούν τα σημεία του τρέχοντος θέματος. Εξ' αρχής, οφείλουμε να αναφερθούμε στην βιβλιοθήκη HTTP. Η βιβλιοθήκη HTTP περιλαμβάνει τρία αντικείμενα, με κυριότερο το HttpClient. Το HttpClient είναι το αντικείμενο που εκτελεί και διαχειρίζεται τα αιτήματα και τις απαντήσεις. Το

HttpClient μπορεί να εκτελέσει πολλαπλά αιτήματα, ταυτόχρονα. Διαχειρίζεται αιτήματα που εκτελούνται στο παρασκήνιο. Το αντικείμενο αυτό το δηλώνουμε στην υπορουτίνα Process_Globals.

Στο παράδειγμα μας προσθέτουμε δύο αρχεία. Το CountryCodes.txt, το οποίο είναι ένα αρχείο κειμένου που περιέχει τη λίστα με τα διάφορα νομίσματα. Το δεύτερο αρχείο είναι το layout.bal, είναι το αρχείο της διάταξης και δημιουργήθηκε με τον σχεδιαστή. Στο αρχείο της διάταξης περιέχονται άλλα δύο αρχεία εικόνων και τα κουμπιά με το σχήμα βέλους



Αρχεία διάταξης και κωδικών χωρών

Ο κώδικας διαβάζει το αρχείο κειμένου και το αποθηκεύει σε μία λίστα, στην υπορουτίνα Activity_Create.

```
If FirstTime Then  
    countries = File.ReadList(File.DirAssets, "CountryCodes.txt")
```

Το File.ReadList είναι μία μέθοδος που ανοίγει ένα αρχείο και προσθέτει όλες τις γραμμές σε μία λίστα.

Επόμενο βήμα είναι να δούμε τον τρόπο με τον οποίο γίνεται η διασύνδεση. Η επικοινωνία γίνεται σε δύο στάδια. Πρώτα επαληθεύεται η σύνδεση με την αποστολή ενός αντικειμένου HttpRequest και στη συνέχεια η απάντηση διαβάζεται από τον διακομιστή. Το παρακάτω κομμάτι κώδικα δημιουργεί και στέλνει την αίτηση GET:

```
Dim request As HttpRequest  
request.InitializeGet(URL & fromCountry & "&ToCurrency=" & toCountry)  
request.Timeout = 10000 'set timeout to 10 seconds  
If HttpClient1.Execute(request, 1) = False Then Return 'Will be false if  
their is already a running task (with the same id).
```

Επειδή η σύνδεση μπορεί να καθυστερήσει αρκετά, θέτουμε ένα χρονικό όριο στα 10 δευτερόλεπτα. Αυτό το χρονικό όριο στα είναι αρκετά μικρό, έτσι αν υπάρξει αποτυχία της σύνδεσης λόγω αστάθειας της διαδικτυακής υπηρεσίας να ολοκληρωθεί γρήγορα.

Επίσης στο κομμάτι αυτό παρατηρούμε και την μέθοδο HttpClient.Execute, η οποία λαμβάνει δύο παραμέτρους. Η πρώτη είναι το αντικείμενο request και η δεύτερη είναι το Task ID. Αυτή η τιμή θα περάσει στα γεγονότα ResponceSuccess ή ResponceError. Η μέθοδος HttpClient.Execute θα επιστρέψει false εάν υπάρχει ήδη σε λειτουργία κάποια εργασία με το ίδιο ID. Μόλις η απάντηση είναι έτοιμη, τότε θα προκληθούν τα γεγονότα ResponceSuccess ή ResponceError. Εάν έχουμε θετικό αποτέλεσμα, τότε θα μπορούμε να διαβάσουμε και να εμφανίσουμε την απάντηση. Διαφορετικά, θα εμφανιστεί ένα “toast” μήνυμα που θα εμφανίσει το μήνυμα λάθους.

Την πρώτη φορά που θα θέσουμε σε λειτουργία την εφαρμογή θα πρέπει να ορίσουμε τις τιμές με τις προκαθορισμένες τιμές που θέλουμε. Αυτό γίνεται στην υπορουτίνα Sub ResetState.

```
Sub ResetState
    'set the starting state
    State.TextUp = 1
    State.TextDown = ""
    State.IndexUp = 0 'USD
    State.IndexDown = 43 'Euro
End Sub
```

Το State είναι μία μεταβλητή, η οποία αποθηκεύει την κατάσταση της εφαρμογής και δεν αλλοιώνεται σε περίπτωση καταστροφής της activity.. Η μεταβλητή αυτή είναι τύπου Type, επονομαζόμενο StateType και τον έχουμε δημιουργήσει μόνοι μας με την λέξη – κλειδί Type. Το State αποτελείται από τις τιμές στα πλαίσια κειμένου (text boxes) και το τρέχον επιλεγμένο νόμισμα και δηλώνεται στο Sub Process_Globals.

```
Type StateType (TextUp As String, TextDown As String, _
    IndexUp As Int, IndexDown As Int)
    Dim State As StateType 'This must be a process variable as it stores the
state
                                'and should not be released when the activity is
destroyed.
```

Στη συνέχεια αποθηκεύουμε και διαβάζουμε τη μεταβλητή State, όπως απαιτείται κάθε φορά στις υπορουτίνες Activity_Resume και Activity_Pause. Συγκεκριμένα στην υπορουτίνα Sub Activity_Resume διαβάζουμε τις τιμές και ορίζουμε τα απαιτούμενα ενεργά εικονίδια.


```

Sub Activity_Resume
    txtUp.Text = State.TextUp
    txtDown.Text = State.TextDown
    spnrUp.SelectedIndex = State.IndexUp
    spnrDown.SelectedIndex = State.IndexDown
End Sub

```

Στο Sub Activity_Pause αποθηκεύουμε τις τιμές στην μεταβλητή State. Στην υπορουτίνα αυτή ορίζουμε αν ο χρήστης πατήσει το κουμπί επιστροφής, η εφαρμογή θα κλείσει και θα επιστρέψουμε στην αρχική κατάσταση.

```

Sub Activity_Pause (UserClosed As Boolean)
    If UserClosed Then
        ResetState 'reset the state to the initial settings.
    Else
        State.TextUp = txtUp.Text
        State.TextDown = txtDown.Text
        State.IndexUp = spnrUp.SelectedIndex
        state.IndexDown = spnrDown.SelectedIndex
    End If
End Sub

```

Με το συγκεκριμένο παράδειγμα δώσαμε μερικά βασικά στοιχεία που αφορούν την σύνδεση μίας εφαρμογής με μία διαδικτυακή υπηρεσία. Οι δυνατότητες της Basic4Android, όμως, δεν σταματούν εδώ.

6.4 Λήψη εικόνων με την κάμερα

Η Basic4Android μας δίνει την δυνατότητα να μπορούμε να χρησιμοποιούμε την φωτογραφική μηχανή μίας συσκευής σε εφαρμογές. Αυτό γίνεται με την χρήση της βιβλιοθήκης Camera και το νέο CameraEX Class, το οποίο επεκτείνει την λειτουργικότητα της κάμερας και των παραμέτρων της.

Για να μπορέσουμε να τραβήξουμε μία εικόνα με την φωτογραφική μηχανή, απαιτούνται τα παρακάτω στάδια:

- Πρώτα πρέπει να δηλώσουμε και να αρχικοποιήσουμε ένα αντικείμενο Camera. Η μεθοδολογία Initialize αναμένει ένα Panel. Οι εικόνες προεπισκόπησης θα εμφανίζονται μέσα σ' αυτό το Panel.
- Μόλις η φωτογραφική μηχανή είναι έτοιμη, προκαλείται το γεγονός Camera_Ready. Στην υπορουτίνα του γεγονότος Ready, καλούμε το Camera.StartPreview για να δούμε την προεπισκόπηση της εικόνας. Αμέσως μετά καλούμε το TakePicture ώστε να τραβήξουμε μία εικόνα. Όπως φαίνεται και παρακάτω στο συνημμένο κώδικα, το κουμπί TakePicture (btnTakePicture) ενεργοποιείται μόνο όταν η κάμερα είναι έτοιμη.

- Το γεγονός PictureTaken θα προκληθεί με την εικόνα να περνάει ως μία σειρά από bytes (σε μορφή jpeg) και να αποθηκεύεται στην εξωτερική μνήμη της συσκευής.
- Τέλος, καλούμε ξανά το StartPreview για να επανεκκινήσουμε την δυνατότητα της προεπισκόπησης των εικόνων.

```

Sub Process_Globals
    Private frontCamera As Boolean = False
End Sub

Sub Globals
    Private Panel1 As Panel
    Private camEx As CameraExClass
End Sub

Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("1")
End Sub

Sub Activity_Resume
    InitializeCamera
End Sub

Private Sub InitializeCamera
    camEx.Initialize(Panel1, frontCamera, Me, "Camera1")
    frontCamera = camEx.Front
End Sub

Sub Activity_Pause (UserClosed As Boolean)
    camEx.Release
End Sub

Sub Camera1_Ready (Success As Boolean)
    If Success Then
        camEx.SetJpegQuality(90)
        camEx.CommitParameters
        camEx.StartPreview
        Log(camEx.GetPreviewSize)
    Else
        ToastMessageShow("Cannot open camera.", True)
    End If
End Sub

Sub btnTakePicture_Click
    camEx.TakePicture
End Sub

Sub btnFocus_Click
    camEx.FocusAndTakePicture
End Sub

Sub Camera1_PictureTaken (Data() As Byte)
    Dim filename As String = "1.jpg"
    Dim dir As String = File.DirRootExternal

```

```

        camEx.SavePictureToFile(Data, dir, filename)
        camEx.StartPreview 'restart preview

        'send a broadcast intent to the media scanner to force it to scan
        the saved file.
        Dim Phone As Phone
        Dim i As Intent
        i.Initialize("android.intent.action.MEDIA_SCANNER_SCAN_FILE", _
            "file://" & File.Combine(dir, filename))
        Phone.SendBroadcastIntent(i)
        ToastMessageShow("Picture saved." & CRLF & "File size: " &
File.Size(dir, filename), True)
End Sub

Sub ChangeCamera_Click
    camEx.Release
    frontCamera = Not(frontCamera)
    InitializeCamera
End Sub

Sub btnEffect_Click
    Dim effects As List = camEx.GetSupportedColorEffects
    If effects.IsInitialized = False Then
        ToastMessageShow("Effects not supported.", False)
        Return
    End If
    Dim effect As String =
effects.Get((effects.IndexOf(camEx.GetColorEffect) + 1) Mod
effects.Size)
    camEx.SetColorEffect(effect)
    ToastMessageShow(effect, False)
    camEx.CommitParameters
End Sub

Sub btnFlash_Click
    Dim f() As Float = camEx.GetFocusDistances
    Log(f(0) & ", " & f(1) & ", " & f(2))
    Dim flashModes As List = camEx.GetSupportedFlashModes
    If flashModes.IsInitialized = False Then
        ToastMessageShow("Flash not supported.", False)
        Return
    End If
    Dim flash As String =
flashModes.Get((flashModes.IndexOf(camEx.GetFlashMode) + 1) Mod
flashModes.Size)
    camEx.SetFlashMode(flash)
    ToastMessageShow(flash, False)
    camEx.CommitParameters
End Sub

Sub btnPictureSize_Click
    Dim pictureSizes() As CameraSize =
camEx.GetSupportedPicturesSizes
    Dim current As CameraSize = camEx.GetPictureSize
    For i = 0 To pictureSizes.Length - 1

```

```

    If pictureSizes(i).Width = current.Width AND
pictureSizes(i).Height = current.Height Then Exit
    Next
    Dim ps As CameraSize = pictureSizes((i + 1) Mod
pictureSizes.Length)
    camEx.SetPictureSize(ps.Width, ps.Height)
    ToastMessageShow(ps.Width & "x" & ps.Height, False)
    camEx.CommitParameters
End Sub

```

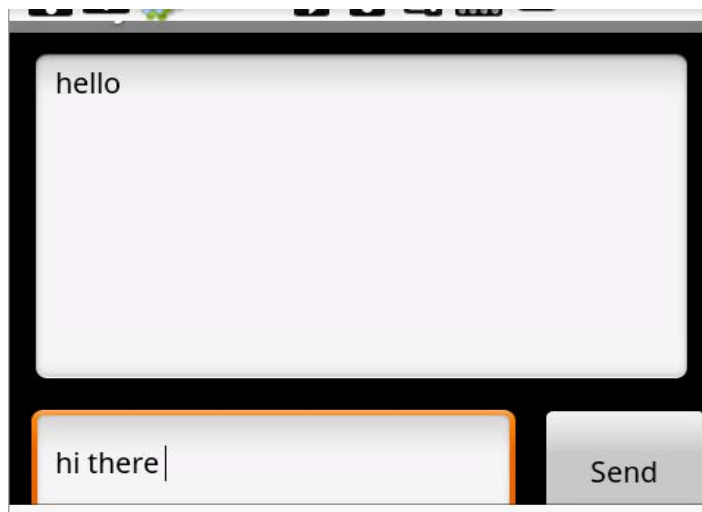
Με την έκδοση της βιβλιοθήκης Camera v2.20 + έχουμε την δυνατότητα να χρησιμοποιήσουμε το CameraEX Class, το οποίο επεκτείνει την λειτουργικότητα της κάμερας.

Μερικά βασικά χαρακτηριστικά του CameraEX Class είναι:

- Εύκολο άνοιγμα της μπροστινής ή της πίσω κάμερας.
- Υποστηρίζονται όλοι οι προσανατολισμοί στις εικόνες προεπισκόπησης αλλά και στις αποθηκευμένες εικόνες.
- Δίνει πρόσβαση στις παραμέτρους της Camera (flash mode, picture size, effects, και άλλα)

6.5 Android Serial (Bluetooth)

Η βιβλιοθήκη Serial επιτρέπει τη σύνδεση συσκευών μέσω Bluetooth χρησιμοποιώντας τις εικονικές σειριακές θύρες (RFCOMM). Σ' αυτήν τη παράγραφο θα καλύψουμε ένα μεγάλο κομμάτι της σειριακής επικοινωνίας δίνοντας μία εφαρμογή απλής γραπτής συνομιλίας μέσω Bluetooth.



Αρχικά δημιουργούμε ένα αντικείμενο, που ονομάζεται Serial1 και είναι τύπου Serial. Το Serial1 δηλώνεται στο Sub Process_Globals. Το αντικείμενο Serial, το αρχικοποιούμε στην υπορουτίνα Sub Activity_Create όταν η πρώτη φορά είναι αληθής (FirstTime As Boolean). Μ' αυτόν τον τρόπο τα process globals αντικείμενα αρχικοποιούνται μόνο μία φορά.

```
Sub Activity_Create(FirstTime As Boolean)
    If FirstTime Then
        Serial1.Initialize("Serial1")
        Timer1.Initialize("Timer1", 200)
    End If
    Activity.LoadLayout("1")
    Activity.AddMenuItem("Connect", "mnuConnect")
    Activity.AddMenuItem("Disconnect", "mnuDisconnect")
End Sub
```

Στην υπορουτίνα Sub Activity_Resume ελέγχουμε εάν η συσκευή Bluetooth είναι ενεργοποιημένη. Εάν δεν είναι, τότε ζητάμε από τον χρήστη να την ενεργοποιήσει. Αυτό το κομμάτι του κώδικα το τοποθετούμε στο Sub Activity_Resume αντί στο Sub Activity_Create και δίνει τη δυνατότητα να γίνεται αυτός ο έλεγχος κάθε φορά που η δραστηριότητα θα συνεχίζεται μετά από οποιαδήποτε διακοπή. Έτσι, εάν ο χρήστης μεταβεί στην οθόνη των ρυθμίσεων, ενεργοποιήσει το Bluetooth και μετά επιστρέψει στην εφαρμογή μας, θα ξέρουμε ότι το Bluetooth έχει ενεργοποιηθεί. Διαφορετικά, εάν το Bluetooth είναι ενεργοποιημένο, τότε ξεκινάμε να ακούμε τις εισερχόμενες συνδέσεις. Αυτό επιτρέπει στις άλλες συσκευές να συνδεθούν με την δικιά μας.

```
Sub Activity_Resume
    If Serial1.IsEnabled = False Then
        MsgBox("Please enable Bluetooth.", "")
    Else
        Serial1.Listen 'listen for incoming connections
    End If
End Sub
```

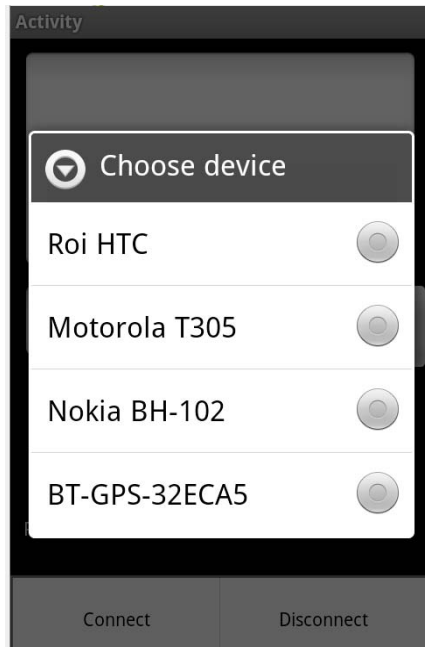
Όταν ο χρήστης πατήσει στο Connect Menu, εμφανίζεται στην οθόνη μία λίστα με τις γνωστές αξιόπιστες συσκευές. Μόλις ο χρήστης επιλέξει το όνομα μίας συσκευής, θα εξάγουμε την διεύθυνση MAC από τον χάρτη και θα συνδεθούμε σ' αυτήν.

```
Sub mnuConnect_Click
    Dim PairedDevices As Map
    PairedDevices = Serial1.GetPairedDevices
    Dim l As List
    l.Initialize
    For i = 0 To PairedDevices.Size - 1
        l.Add(PairedDevices.GetKeyAt(i)) 'add the friendly name to the list
    Next
```

```

Dim res As Int
res = InputList(1, "Choose device", -1) 'show list with paired devices
If res <> DialogResult.CANCEL Then
    Serial1.Connect(PairedDevices.Get(1.Get(res))) 'convert the name to m
ac address
End If
End Sub

```



Οθόνη διαθέσιμων συσκευών για σύνδεση

Η σύνδεση δεν εδραιώνεται αμέσως, η διαδικασία αυτή ολοκληρώνεται στο παρασκήνιο. Όταν η σύνδεση εδραιωθεί, θα προκληθεί το γεγονός Connected. Εάν η σύνδεση εδραιωθεί επιτυχώς, τότε μπορούμε να ξεκινήσουμε τη μεταφορά των δεδομένων. Τα TextReader1 και TextWriter1 είναι αντικείμενα process globals και τα αρχικοποιούμε χρησιμοποιώντας το serial stream. Αυτό θα μας επιτρέψει να στέλνουμε και να λαμβάνουμε κείμενο μέσω της σύνδεσης που έχουμε δημιουργήσει. Ο timer1 χρησιμοποιείται για να εξετάσουμε κατά πόσο υφίστανται εισερχόμενα δεδομένα.

```

Sub Serial1_Connected (Success As Boolean)
    If Success Then
        ToastMessageShow("Connected successfully", False)
        TextReader1.Initialize(Serial1.InputStream)
        TextWriter1.Initialize(Serial1.OutputStream)
        timer1.Enabled = True
        connected = True
    Else
        connected = False
        Timer1.Enabled = False
        MsgBox(LastException.Message, "Error connecting.")
    End If
End Sub

```

```
End If  
End Sub
```

Όταν ο χρήστης πατήσει το κουμπί Send, στέλνουμε το κείμενο. Χρησιμοποιούμε την μεταβλητή connected, που χρησιμοποιούμε για να γνωρίζουμε εάν είμαστε συνδεδεμένοι στην τρέχουσα κατάσταση. Με την κλήση του Flush επιβεβαιώνουμε ότι το κείμενο δεν θα κάνει buffering και ότι θα σταλεί αμέσως.

```
Sub btnSend_Click  
    If connected Then  
        TextWriter1.WriteLine(txtSend.Text)  
        TextWriter1.Flush  
        txtSend.Text = ""  
    End If  
End Sub
```

Στην λήψη μηνυμάτων γίνεται έλεγχος αν υπάρχουν δεδομένα κάθε φορά που στο χρονόμετρο (timer1) θα έχουμε ένα χτύπο. Αν υπάρχουν θα τα διαβάσει και θα τα προσθέτει στο μεγάλο EditText.

```
Sub Timer1_Tick  
    If connected Then  
        If TextReader1.Ready Then 'check if there is any data waiting to be read  
            txtLog.Text = txtLog.Text & TextReader1.ReadLine & CRLF  
            txtLog.SelectionStart = txtLog.Text.Length  
        End If  
    End If  
End Sub
```

Το TextReader.Readline είναι μία κλήση με την οποία περιμένουμε να διαβαστεί μέχρι και ο τελευταίος απλός χαρακτήρας. Ως εκ τούτου πρέπει να ελέγχουμε το TextReader.Ready εάν δεν θέλουμε να μπλοκάρει η εφαρμογή μας.

Αξίζει να σημειώσουμε ότι η συγκεκριμένη εφαρμογή μπορεί να χρησιμοποιηθεί για να συνδέσουμε την συσκευή μας με μία συσκευή που μπορεί να μην είναι Android, όπως για παράδειγμα με ένα εξωτερικό GPS. Απλά το GPS θα στέλνει τα δεδομένα του ως κείμενο.

Ολόκληρος ο κώδικας της εφαρμογής εμφανίζεται παρακάτω:

```

Sub Process_Globals
    Dim Serial1 As Serial
    Dim TextReader1 As TextReader
    Dim TextWriter1 As TextWriter
    Dim Timer1 As Timer
    Dim connected As Boolean
End Sub

Sub Globals
    Dim btnSend As Button
    Dim txtLog As EditText
    Dim txtSend As EditText
End Sub

Sub Activity_Create(FirstTime As Boolean)
    If FirstTime Then
        Serial1.Initialize("Serial1")
        Timer1.Initialize("Timer1", 200)
    End If
    Activity.LoadLayout("1")
    Activity.AddMenuItem("Connect", "mnuConnect")
    Activity.AddMenuItem("Disconnect", "mnuDisconnect")
End Sub

Sub Activity_Resume
    If Serial1.IsEnabled = False Then
        MsgBox("Please enable Bluetooth.", "")
    Else
        Serial1.Listen 'listen for incoming connections
    End If
End Sub

Sub mnuConnect_Click
    Dim PairedDevices As Map
    PairedDevices = Serial1.GetPairedDevices
    Dim l As List
    l.Initialize
    For i = 0 To PairedDevices.Size - 1
        l.Add(PairedDevices.GetKeyAt(i))
    Next
    Dim res As Int
    res = InputList(l, "Choose device", -1) 'show list with paired devices
    If res <> DialogResponse.CANCEL Then
        Serial1.Connect(PairedDevices.Get(l.Get(res))) 'convert the name
to mac address
    End If
End Sub

Sub Serial1_Connected (Success As Boolean)
    If Success Then
        ToastMessageShow("Connected successfully", False)
        TextReader1.Initialize(Serial1.InputStream)
        TextWriter1.Initialize(Serial1.OutputStream)
        Timer1.Enabled = True
        connected = True
    Else
        connected = False
        Timer1.Enabled = False
    End If
End Sub

```



```

        MsgBox(LastException.Message, "Error connecting.")
    End If
End Sub
Sub mnuDisconnect_Click
    Serial1.Disconnect
    connected = False
End Sub

Sub Activity_Pause (UserClosed As Boolean)

End Sub

Sub btnSend_Click
    If connected Then
        TextWriter1.WriteLine(txtSend.Text)
        TextWriter1.Flush
        txtSend.Text = ""
    End If
End Sub
Sub Timer1_Tick
    If connected Then
        If TextReader1.Ready Then 'check if there is any data waiting to
be read
            txtLog.Text = txtLog.Text & TextReader1.ReadLine & CRLF
            txtLog.SelectionStart = txtLog.Text.Length
        End If
    End If
End Sub

```

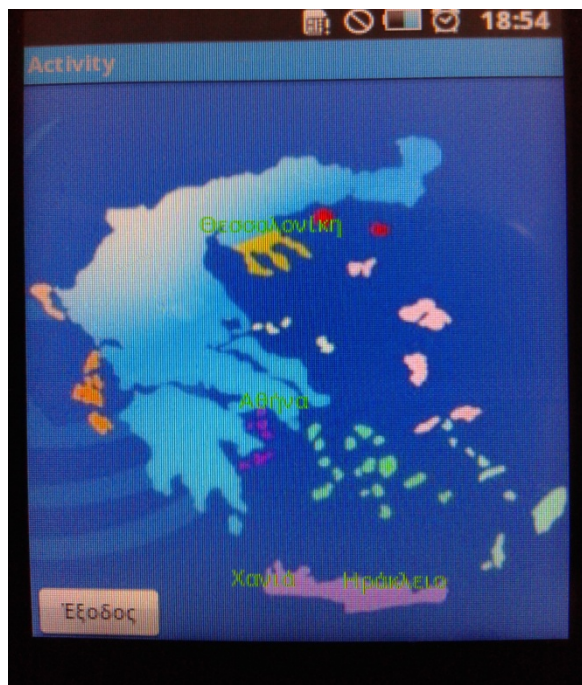
7

Ανάλυση εφαρμογής καιρού

7.1 Εισαγωγή και περιγραφή εφαρμογής

Στο κεφάλαιο αυτό θα δούμε την εφαρμογή που δημιουργήσαμε για την ενημέρωση του καιρού. Η εφαρμογή μας εμφανίζει τον καιρό στις τέσσερις πόλεις, Θεσσαλονίκη, Αθήνα, Χανιά, Ηράκλειο την τρέχουσα στιγμή. Τα αρχεία των εικόνων και το αρχείο των τιμών που χρησιμοποιήσαμε το λαμβάνουμε από έναν FTP Server. Η εφαρμογή μας είναι αρκετά απλή

και δημιουργήθηκε ώστε να δούμε μερικά ακόμη στοιχεία που μπορεί να μας προσφέρει η Basic4Android.



7.2 Αρχεία εικόνων και τιμών καιρού

Όπως προαναφέραμε, στην εφαρμογή μας εμφανίζονται κάποιες εικόνες. Οι εικόνες αυτές χρησιμοποιούνται για να απεικονίσουν την ηλιοφάνεια στην εκάστοτε περιοχή, την κατεύθυνση του ανέμου, αλλά και μία χαρακτηριστική εικόνα της πόλης, της οποίας ανοίγουμε την καρτέλα. Το αρχείο αυτών των εικόνων βρίσκεται σε έναν φάκελο στον FTP Server με τον οποίο συνδέεται η εφαρμογή μας. Αυτό μας δίνει τη δυνατότητα να αλλάξουμε πολύ εύκολα και ανά πάσα στιγμή τις εικόνες αυτές. Σε αντίθετη περίπτωση, αν ήταν σεταρισμένες στην συσκευή του κινητού τηλεφώνου δεν θα είχαμε αυτή τη δυνατότητα. Όπως, επίσης, θα έπρεπε κάθε φορά που θέλουμε να τρέξουμε την εφαρμογή μας σε μία άλλη συσκευή, θα χρειαζόταν να σετάρουμε τις εικόνες στην νέα πλέον συσκευή.

Το αρχείο με τις τιμές των θερμοκρασιών και την ταχύτητα του ανέμου είναι ένα αρχείο CSV και το βρίσκουμε και αυτό στον FTP Server που χρησιμοποιούμε. Επιλέξαμε το CSV ως μία πολύ απλή και κατανοητή μορφή αρχείου. Το αρχείο αυτό μας προσφέρει δύο χαρακτηριστικά που μας εξυπηρετούν.

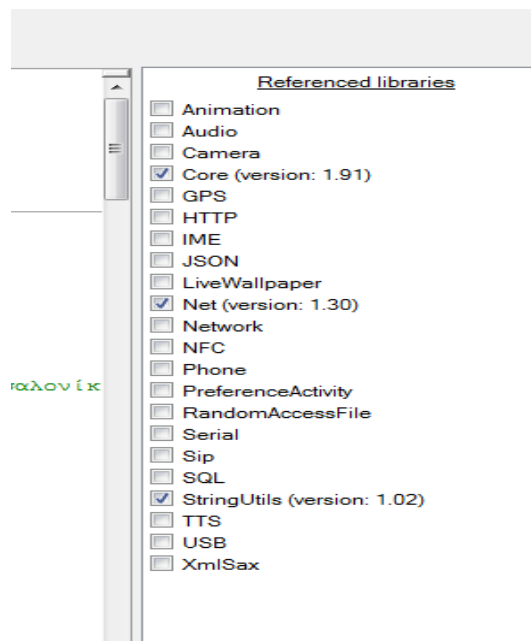
- Είναι πολύ εύκολο να διαβαστεί από το Android χωρίς να απαιτείται κάποια επιπλέον βιβλιοθήκη και τα δεδομένα που περιέχει να μεταφερθούν άμεσα σε πίνακα.
- Μπορούμε να επεξεργαστούμε το αρχείο CSV πολύ εύκολα με το Excel.

7.3 Ανάλυση εφαρμογής

Προϋπόθεση όλων είναι να δηλώσουμε όλες τις απαραίτητες μεταβλητές, τις οποίες θα χρειαστούμε στο πρόγραμμα μας. Οι βασικότερες μεταβλητές που έχουμε να δηλώσουμε είναι τα αντικείμενα FTP για την σύνδεση της εφαρμογής μας με τον Server. Αυτά τα αντικείμενα δηλώνονται στο Sub Process_Globals ώστε να μπορούν να προσπελαστούν από όλες τις μονάδες.

```
Sub Process_Globals
    'These global variables will be declared once when the application
    starts.
    'These variables can be accessed from all modules.
    Dim FTP As FTP
    Dim FTP1 As FTP
End Sub
```

Το αντικείμενο FTP είναι μέρος της βιβλιοθήκης NET. Οπότε, θα πρέπει να αναφέρουμε την συγκεκριμένη βιβλιοθήκη στη λίστα των βιβλιοθηκών που χρησιμοποιούμε στον κώδικα μας.



Το FTP πρωτόκολλο υποστηρίζει μία εργασία κάθε φορά, έτσι οι εργασίες υποβάλλονται σε επεξεργασία σειριακά. Γενικά, η βιβλιοθήκη FTP είναι αρκετά απλή στη χρήση της.

Μετά την δήλωση τους, πρέπει να αρχικοποιήσουμε τα αντικείμενα FTP. Η αρχικοποίηση τους γίνεται στο Sub Activity_Create όταν είναι η πρώτη φορά αληθής (FirstTime As Boolean).

```
FTP.Initialize("FTP","data-expert.dyndns.info","21","maria11992","m1n8k4a5")  
    'Αρχικοποιώ το FTP για το Connection
```

Αξίζει να σημειώσουμε ότι η αρχικοποίηση του αντικειμένου FTP δεν προκαλεί τη σύνδεση με τον Server. Η σύνδεση θα εδραιωθεί έμμεσα μαζί με την επόμενη εργασία.

Στην εφαρμογή που εμείς κατασκευάσαμε, έχουμε χρησιμοποιήσει δύο αντικείμενα FTP. Ο λόγος που προβήκαμε σε αυτήν την ενέργεια είναι για να μπορέσουμε να εισάγουμε, σχεδόν παράλληλα τις τιμές του καιρού από το CSV αρχείο με την σύνδεση του αντικειμένου FTP και τις εικόνες με την σύνδεση του αντικειμένου FTP1.

Εν συνεχεία, ορίζουμε και τα δύο αντικείμενα FTP να δουλεύουν σε παθητική λειτουργία.

```
FTP.PassiveMode = True
```

Το επόμενο βήμα είναι να επιβεβαιώσουμε ότι έχει αρχικοποιηθεί η σύνδεση μας, ώστε να μπορέσουμε να ζητήσουμε από τον Server την λίστα των αρχείων. Μ' αυτόν τον τρόπο ελέγχουμε παράλληλα αν έχει εδραιωθεί η σύνδεση της εφαρμογής με τον Server, διότι η διαδικασία αυτή δεν γίνεται στο προσκήνιο αλλά στο παρασκήνιο.

Αν σε περίπτωση δεν εδραιωθεί η σύνδεση, εμφανίζεται ένα μήνυμα στην οθόνη της συσκευής ότι η σύνδεση με τον διακομιστή δεν είναι εφικτή.

```
FTP.Initialize("FTP","data-expert.dyndns.info","21","maria11992","m1n8k4a5")  
    'Αρχικοποιώ το FTP για το Connection  
    FTP.PassiveMode = True  
  
    If FTP.IsInitialized = True Then  
        'Ζητάω την λίστα αρχείων και δείχνω μήνυμα αν δεν μπορώ να συνδεθώ  
        FTP.List("/android")  
        'Το αρχείο το διαβάζω από τη μνήμη μόνο αν δεν μπορώ α συνδεθώ  
    Else  
        MsgBox ("Η σύνδεση με τον διακομιστή δεν είναι εφικτή.", "FTP  
Connection")  
    End If  
    DoEvents  
    'Αφήνω την εφαρμογή λίγο ελεύθερη  
    Lost_Picture = False  
    'Ορίζω ότι μου λείπουν αρχεία από εικόνες καιρού και ψάχνω να δω αν μου  
    λείπει καμία  
    For i = 0 To 7  
        'και επειδή οι 3 πίνακας  
        έχουν λιγότερες θέσεις, σταματάω στο 4 και το 3. Προσοχή, θα κατεβάσει αρχεία  
        έστω και ένα αν λείπει
```

```

        If i < 6 AND File.Exists(File.DirInternal, hliofaneia(i)) = False
Then Lost_Picture = True
        If i < 4 AND File.Exists(File.DirInternal, Town_Name(i)) = False
Then Lost_Picture = True
        If File.Exists(File.DirInternal, anemos(i)) = False Then
Lost_Picture = True
        Next
        If Lost_Picture = True Then
            'Αν τελικά λείπει...
            FTP1.Initialize("FTP", "data-
expert.dyndns.info", "21", "maria11992", "m1n8k4a5") 'Αρχικοποιώ το
FTP για το Connection
            FTP1.PassiveMode = True
            If FTP1.IsInitialized = True Then
                'Ζητάω την λίστα αρχείων και δείχνω μήνυμα αν δεν μπορώ να συνδεθώ
                FTP1.List("/android/images")
            Else
                MsgBox ("Εντοπίστηκε πρόβλημα στην εφαρμογή", "Internal
Error")
            End If
        End If
    End Sub
End Sub

```

Αξίζει να σημειωθεί ότι σε περίπτωση που δεν είναι εφικτή η σύνδεση, η εφαρμογή μας θα εμφανίζει τα στοιχεία του καιρού που είχαμε λάβει από την τελευταία σύνδεση που είχε γίνει με τον διακομιστή.

Προχωρώντας στο πρόγραμμα αρχικοποιούμε και το δεύτερο αντικείμενο FTP (FTP1) και ορίζουμε και σε αυτό τον τρόπο λειτουργίας του, ο οποίος είναι επίσης παθητικός. Στη συνέχεια ελέγχουμε αν έχει ολοκληρωθεί η αρχικοποίηση της σύνδεσης και αν έχει γίνει ζητάμε από τον διακομιστή να προσκομίσει τη λίστα των αρχείων που βρίσκονται στην καθορισμένη διαδρομή. Τα αρχεία που ζητάμε από αυτήν την σύνδεση είναι οι εικόνες. Αντιθέτως αν δεν πραγματοποιηθεί η αρχικοποίηση μας εμφανίζει ένα μήνυμα που μας ενημερώνει ότι εντοπίστηκε πρόβλημα στην εφαρμογή.

Όλη η διαδικασία που αφορά το FTP1 γίνεται υπό την προϋπόθεση ότι λείπει κάποιο αρχείο εικόνας από την συσκευή, οπότε και του ζητάμε να μας τα κατεβάσει στη μνήμη της συσκευής.

```

Lost_Picture = False
'Ορίζω ότι μου λείπουν αρχεία από εικόνες καιρού και ψάχνω να δω αν μου
λείπει καμία
For i = 0 To 7 'και επειδή οι 3 πίνακες
έχουν λιγότερες θέσεις, σταματώ στο 4 και το 3. Προσοχή, θα κατεβάσει αρχεία
έστω και ένα αν λείπει
    If i < 6 AND File.Exists(File.DirInternal, hliofaneia(i)) = False
Then Lost_Picture = True
    If i < 4 AND File.Exists(File.DirInternal, Town_Name(i)) = False
Then Lost_Picture = True

```

```

        If File.Exists(File.DirInternal, anemos(i)) = False Then
Lost_Picture = True
        Next
        If Lost_Picture = True Then 'Αν τελικά λείπει...
            FTP1.Initialize("FTP", "data-
expert.dyndns.info", "21", "maria11992", "m1n8k4a5") 'Αρχειοποιώ το
FTP για το Connection

```

Στον κώδικα μας, αμέσως πριν καλέσουμε το [FTP1.Initialize](#) έχει προηγηθεί ένας έλεγχος για να επιβεβαιώσουμε αν όλες οι εικόνες υπάρχουν στην εσωτερική μνήμη της συσκευής και μπορούν να διαβαστούν.

```

Lost_Picture = False
'Ορίζω ότι μου λείπουν αρχεία από εικόνες καιρού και ψάχνω να δω αν μου
λείπει καμία
For i = 0 To 7 'και επειδή οι 3 πίνακες
έχουν λιγότερες θέσεις, σταματώ στο 4 και το 3. Προσοχή, θα κατεβάσει αρχεία
έστω και ένα αν λείπει
    If i < 6 AND File.Exists(File.DirInternal, hliofaneia(i)) = False
Then Lost_Picture = True
    If i < 4 AND File.Exists(File.DirInternal, Town_Name(i)) = False
Then Lost_Picture = True
    If File.Exists(File.DirInternal, anemos(i)) = False Then
Lost_Picture = True
    Next

```

Τα γεγονότα [FTP.ListCompleted](#) προκαλούνται μετά το αίτημα που έχει στείλει η [FTP.List](#) και όταν τα δεδομένα είναι πλέον διαθέσιμα.

Στο [FTP.ListCompleted](#), ελέγχουμε αν η εντολή List του FTP έχει ολοκληρωθεί επιτυχώς. Αν ο παραπάνω έλεγχος μας επιφέρει θετικά αποτελέσματα θα κατεβάσει το αρχείο με τις νέες τιμές καιρού. Διαφορετικά αν δεν είναι επιτυχής ο έλεγχος, θα εμφανίσει μήνυμα που θα μας ενημερώνει ότι δεν έγινε η φόρτωση των νέων τιμών καιρού και θα διαβάσει τις τελευταίες αποθηκευμένες, στη συσκευή, τιμές.

```

Sub FTP_ListCompleted (ServerPath As String, Success As Boolean, Folders() As
FTPEntry, Files() As FTPEntry)
    If Success = False Then
        'Ελέγχει αν η εντολή listindex του FTP έχει
ολοκληρωθεί επιτυχώς
        MsgBox ("Σφάλμα φόρτωσης νέων τιμών καιρού.", "Σφάλμα Σύνδεσης")
        'Αν όχι, βγάζει σφάλμα
        If File.Exists(File.DirInternal, "temp.csv") Then
            Dim cells() As String
            'Ορίζω τα κελιά για διάβασμα αρχείου CSV
            Dim su As StringUtils
            Dim list1 As List
            'Ορίζω μια λίστα για να αποθηκεύσω τα κελιά
            list1.Initialize

```

```

        list1 = su.LoadCSV(File.DirInternal, "temp.csv", ";")
        'Αποθηκεύω το αρχείο στην λίστα
        For i = 0 To list1.Size - 2
            'Διαβάζω μια-μια τις γραμμές τις λίστας και αποθηκεύω στον πίνακα
            καιρού
                cells = list1.get(i+1)
                temp(0,i) = cells(0)
                temp(1,i) = cells(1)
                temp(2,i) = cells(2)
                temp(3,i) = cells(3)
            Next
        End If
    Else
        For i = 0 To Files.Length - 1
            'Αν ναι, ελέγχει τα ονόματα των αρχείων στον φάκελο του
            ListIndex και τα κατεβάζει
            FTP.DownloadFile("/android/" & Files(i).Name, False,
            File.DirInternal, Files(i).Name)
        Next
    End If
End Sub

```

Στο συγκεκριμένο μέρος του κώδικα, αξίζει να τονίσουμε ότι δηλώνουμε την μεταβλητή su ως StringUtils. Η μεταβλητή αυτή χρησιμοποιείται ώστε να γίνει η φόρτωση του αρχείου CSV και να το αποθηκεύσει στην λίστα (list1). Απαιτείται η δήλωση της ομώνυμης βιβλιοθήκης στην αντίστοιχη λίστα βιβλιοθηκών, ώστε να αναγνωρίζεται από το πρόγραμμα η μεταβλητή αυτή.

Αντίστοιχη περίπου διαδικασία γίνεται και στο γεγονός [FTP1.ListCompleted](#). Αν είναι επιτυχής η σύνδεση του FTP1 , ζητάμε να κατεβάσει τα αρχεία των εικόνων και να τα αποθηκεύσει στην εσωτερική μνήμη του τηλεφώνου. Αντίθετα, αν υπάρχει αποτυχία, τότε θα εμφανίσει μόνο ένα μήνυμα ότι δεν έγινε η φόρτωση τιμών.

```

Sub FTP1_ListCompleted (ServerPath1 As String, Success1 As Boolean,
Folders1() As FTPEntry, Files1() As FTPEntry)
    If Success1 = False Then
        'Ελέγχει αν η εντολή listindex του FTP έχει
        ολοκληρωθεί επιτυχώς
        MsgBox ("Σφάλμα φόρτωσης εικόνων.", "Σφάλμα Σύνδεσης") 'Αν όχι,
        βγάζει σφάλμα
    Else
        For i = 0 To Files1.Length - 1
            'Αν ναι, ελέγχει τα ονόματα των αρχείων στον φάκελο
            του ListIndex και τα κατεβάζει
            FTP1.DownloadFile("/android/images" & Files1(i).Name,
            False, File.DirInternal, Files1(i).Name)
        Next
        DoEvents
    Next

```

```
End If
End Sub
```

Με την επιτυχή ολοκλήρωση των γεγονότων [FTP.ListCompleted](#) και αφού ολοκληρωθούν και τα γεγονότα [FTP.DownloadFile](#) προκαλούνται αντίστοιχα δύο νέα γεγονότα, το [FTP.DownloadCompleted](#) και το [FTP1.DownloadCompleted](#) .

Στο FTP_DownloadCompleted έχουμε ένα σημείο ελέγχου, το οποίο εξετάζει την ορθότητα της κατάστασης. Αν είναι θετικό το αποτέλεσμα, τότε διαβάζει και γεμίζει τον πίνακα των τιμών καιρού. Η διαδικασία αυτή γίνεται με την βοήθεια του StringUtils, που όπως έχουμε προαναφέρει, χρησιμοποιείται για να φορτώσει το αρχείο CSV. Τα δεδομένα αποθηκεύονται στις θέσεις του πίνακα temp. Ειδάλλως, θα μας εμφανίσει μήνυμα στην οθόνη που θα μας ενημερώνει ότι δεν ολοκληρώθηκε η λήψη των δεδομένων.

```
Sub FTP_DownloadCompleted (ServerPath As String, Success As Boolean)
    If Success = False Then
        MsgBox ("Η λήψη στοιχείων καιρού δεν είναι εφικτή", "Error
Message") 'Ελέγχει αν έγινε το download στο αρχείο καιρού
    Else
        'Αν όχι, βγάζει μήνυμα
        Dim cells() As String
        'Αν ναι, διαβάζει και γεμίζει τον πίνακα τιμών
        καιρού
        Dim su As StringUtils
        Dim list1 As List
        list1.Initialize
        If File.Exists(File.DirInternal, "temp.csv") Then
            list1 = su.LoadCSV(File.DirInternal, "temp.csv", ";")
            For i = 0 To list1.Size - 2
                cells = list1.get(i+1)
                temp(0,i) = cells(0)
                temp(1,i) = cells(1)
                temp(2,i) = cells(2)
                temp(3,i) = cells(3)
            Next
        End If
    End If
End Sub
```

Αντίστοιχα στην υπορουτίνα [FTP1_DownloadCompleted](#) αν δεν είναι επιτυχής η φόρτωση των αρχείων εικόνων θα μας εμφανίζει μήνυμα ότι η λήψη των εικόνων δεν είναι εφικτή.

```
Sub FTP1_DownloadCompleted (ServerPath1 As String, Success1 As Boolean)
    If Success1 = False Then MsgBox ("Η λήψη αρχείων εικόνας δεν είναι
εφικτή", "Error Message") 'Ελέγχει αν έπρεπε να κατεβάσει εικόνες και τα
κατάφερε
End Sub
```


Οι παραπάνω υπορουτίνες αφορούν την σύνδεση της εφαρμογής μας με τον διακομιστή και όλες οι διαδικασίες που απαιτούνται για να λάβουμε στην συσκευή μας τα αρχεία που χρειαζόμαστε.

Από τη στιγμή που έχουμε ορίσει όλες τις προϋποθέσεις για να φορτώσουμε τα δεδομένα μας στη συσκευή, προχωράμε στο επόμενο στάδιο. Αυτό είναι η χρήση των αρχείων στην εφαρμογή μας αλλά και την δημιουργία και τοποθέτηση των ελεγκτηρίων που θα χρησιμοποιηθούν σ' αυτήν.

Στην υπορουτίνα Sub Activity_Touch λαμβάνει μέρος το γεγονός Touch, το οποίο χρησιμοποιείται για να χειριστεί τις ενέργειες της αλληλεπίδρασης του χρήστη με την οθόνη της εφαρμογής.

Εξ' αρχής ορίζουμε την μεταβλητή Town_Select σε μία τιμή όπου δεν έχουμε θέσει κάποια από τις πόλεις. Αυτό το κάνουμε ώστε η εφαρμογή να αντιλαμβάνεται το άγγιγμα του χρήστη στις αντίστοιχες θέσεις των πόλεων και να μας εμφανίζει την καρτέλα της εκάστοτε πόλης. Οπότε αν ο χρήστης αγγίζει την οθόνη σε μία από τις συγκεκριμένες θέσεις, τότε γίνεται ορατό και έρχεται σε πρώτο πλάνο ένα πάνελ που περιέχονται όλα τα στοιχεία του καιρού και της πόλης.

```
Sub Activity_Touch (Action As Int, X As Float, Y As Float)
    Town_Select = 4
    'Label1.text = "X= " & X & CRLF & "Y= " & Y
    For i = 0 To 3
        If (X > town_xy(i,0) AND X < town_xy(i,1)) AND (Y > town_xy(i,2)
AND Y < town_xy(i,3)) Then
            Town_Select = i
        End If
    Next
    If Town_Select > -1 AND Town_Select < 4 Then
        apotelesma.Visible = True
        apotelesma.BringToFront
        'Label1.Text = Label1.Text & CRLF & Town_Select & " " &
temp(Town_Select,0) & "," & temp(Town_Select,1) & "," & temp(Town_Select,2) &
"," & temp(Town_Select,3)
        If File.Exists(File.DirInternal, Town_Name(Town_Select)) Then
            labp1.Initialize("thermokrasia")
            labp2.Initialize("mpofor")
            keim1.Initialize("text1")
            keim2.Initialize("text2")
            labp1.Text = temp(Town_Select,0) & " C"
            labp2.Text = temp(Town_Select,2) & " B"
            labp1.textColor = Colors.Green
            'Ορίζω το χρώμα για κάθε Label πόλης
            labp2.textColor = Colors.Green
            keim1.text = "Η θερμοκρασία αυτή τη στιγμή είναι..."
            keim2.text = "Η ένταση του ανέμου είναι..."
            keim1.textColor = Colors.Red
            keim2.textColor = Colors.Red
        End If
    End If
End Sub
```

```

        keim1.TextSize = 10
        keim2.TextSize = 10
        For j = apotelesma.NumberOfViews - 1 To 0 Step -1
            apotelesma.GetView(j).RemoveView
        Next
        town_img.Initialize("eikona_polhs")
        town_img.Bitmap = LoadBitmap(File.DirInternal,
Town_Name(Town_Select))
        Img_t1.Initialize("Fwto_Hlios")
        Img_t1.Bitmap = LoadBitmap(File.DirInternal,
hliofaneia(temp(Town_Select,1)))
        Img_t2.Initialize("Fwto_anemos")
        Img_t2.Bitmap = LoadBitmap(File.DirInternal,
anemos(temp(Town_Select,3)))
        apotelesma.AddView(Img_t1, 0,1,100,100)
        apotelesma.AddView(keim1, 100,35,85,30)
        apotelesma.AddView(labp1, 130,70,130,30)
        apotelesma.AddView(Img_t2, 0,70,100,100)
        apotelesma.AddView(keim2, 100,105,100,30)
        apotelesma.AddView(labp2, 130,120,100,30)
        apotelesma.AddView(town_img, 0,128,195,160)
    End If
Else
    apotelesma.Visible = False
End If
End Sub

```

Τέλος έχουμε προσθέσει στην εφαρμογή μας και ένα πλήκτρο, το οποίο κλείνει την εφαρμογή και δεν τρέχει στο παρασκήνιο, όπως θα συνέβαινε αν ο χρήστης πατούσε απλά το πλήκτρο επιστροφής της συσκευής. Αυτό το γεγονός μας βοηθάει ώστε να μην καταναλώνει ενέργεια και χώρο από την μνήμη της συσκευής η εφαρμογή μας.

```

Sub Button1_Click
ExitApplication
End Sub

```

Με αυτό το κομμάτι ολοκληρώσαμε την ανάλυση του κώδικα της εφαρμογής μας. Στο παράρτημα μπορείτε να βρείτε ολόκληρο τον κώδικα. Επίσης, στο παράρτημα υπάρχουν και οι αντίστοιχες εικόνες με τις τιμές του καιρού που εμφανίζονται σε κάθε πόλη.

Σύνοψη και μελλοντικές επεκτάσεις

Εν κατακλείδι, θα μπορούσαμε να πούμε ότι η ανάπτυξη εφαρμογών Android με το ολοκληρωμένο αυτό εργαλείο Basic4Android είναι αρκετά απλή, ευχάριστη και πολύ ενδιαφέρουσα. Φυσικά, έχοντας υπόψη μας ότι αυτά που έχουμε αναφέρει για το συγκεκριμένο περιβάλλον ανάπτυξης εφαρμογών είναι ένα μικρό κομμάτι της πίτας αυτής.

Από τα παραπάνω κεφάλαια καταλήγουμε ότι το λειτουργικό σύστημα Android παρέχει πολλές δυνατότητες στους χρήστες του αλλά ακόμα περισσότερες δυνατότητες στους προγραμματιστές, για να υλοποιούν και να διαθέτουν στις αγορές του Android ενδιαφέρον και ουσιώδεις εφαρμογές που μπορούν να λύσουν τα χέρια πολλών αλλά και να τους διασκεδάσουν.

Στην συγκεκριμένη εφαρμογή θα μπορούσαμε μελλοντικά να την αναπτύξουμε, προσθέτοντας πολλές ακόμη δυνατότητες, όπως για παράδειγμα θα μπορούσαμε να προσθέσουμε και την δυνατότητα ακουστικών οδηγιών για κάποιους που έχουν προβλήματα με την όραση τους.

Στον συγκεκριμένο κώδικα μπορούμε να κάνουμε και αρκετές βελτιστοποιήσεις ώστε να μπορούμε να έχουμε καλύτερα και, ίσως, πιο γρήγορα αποτελέσματα στην οθόνη του κινητού του χρήστη.

Βιβλιογραφία

<http://www.sansimera.gr/articles/241>

<http://el.wikipedia.org/>

<http://translate.google.gr/translate?hl=el&langpair=en%7Cel&u=http://apple-history.com/iphone&ei=XOjIUOvSKuHm4QTf9oH4Dw>

<http://tnt.pblogs.gr/2007/09/101141.html>

<http://www.iphoner.gr/>

<http://www.iphonehellas.gr/forum/topic/64280-%CE%B7-%CE%B9%CF%83%CF%84%CE%BF%CF%81%CE%AF%CE%B1-%CF%84%CE%BF%CF%85-iphone-4-%CE%BC%CE%AD%CF%87%CF%81%CE%B9-%CF%83%CE%AE%CE%BC%CE%B5%CF%81%CE%B1/>

<http://en.wikipedia.org/wiki/IOS>

<http://www.cultofmac.com/191340/the-evolution-of-ios-from-iphone-os-to-ios-6-gallery/>

<http://www.allaboutandroid.gr/?p=6362>

<http://www.myphone.gr/>

<http://el.wikipedia.org/wiki/Android>

<http://www.adds.gr/>

[http://en.wikipedia.org/wiki/Android software development](http://en.wikipedia.org/wiki/Android_software_development)

<http://developer.android.com/>

www.doctorandroid.gr

<http://www.greeceandroid.gr/>

<http://ghz.gr/2012/06/20/%CE%B7-microsoft-%CE%B1%CE%BD%CE%B1%CE%BA%CE%BF%CE%AF%CE%BD%CF%89%CF%83%CE%B5-%CF%84%CE%BF-%CE%BB%CE%B5%CE%B9%CF%84%CE%BF%CF%85%CF%81%CE%B3%CE%B9%CE%BA%CF%8C-%CF%83%CF%8D%CF%83%CF%84%CE%B7%CE%BC%CE%B1-windows-phone-8>

<http://www.windowsphone.com/el-gr/phones>

<http://www.sourcecodester.com/>

<http://blogs.lessthandot.com/>

<http://myandroidgr.blogspot.gr/>

<http://www.pcpro.co.uk/>

<http://blog.g4ilo.com/2013/01/basic-android-programming.html>

<http://code.tutsplus.com/tutorials/android-virtual-device-creation--mobile-12350>

<http://www.basic4ppc.com/android/wiki/index.php/AVD>

<http://www.softpedia.com/get/Programming/Coding-languages-Compilers/Basic4android.shtml>

<http://en.wikipedia.org/wiki/Basic4ppc>

<http://www.doctorandroid.gr/2012/03/android-basics-android-sdk.html>

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

<http://www.basic4ppc.com/>

B4A- Tutorials.pdf

Παράρτημα

Στο Παράρτημα παρατίθεται αναλυτικά ο κώδικας της εφαρμογής, ο οποίος για διευκόλυνση της αναγνωσιμότητας του κειμένου δεν παρουσιάστηκε ολοκληρωμένος στα παραπάνω κεφάλαια. Επίσης, σ' αυτό το σημείο παρατίθενται και στιγμιότυπα από την χρήση της εφαρμογής.

```
'Activity module
Sub Process_Globals
    'These global variables will be declared once when the application
    starts.
    'These variables can be accessed from all modules.
    Dim FTP As FTP
    Dim FTP1 As FTP
End Sub

Sub Globals
    Dim picture1 As ImageView                'Η εικόνα του χάρτη που
    χρησιμοποιούμε
    Dim labelp1 As Label                    'Δήλωση για τα Labels που θα μούν
    για τις πόλεις (Θεσσαλονίκη)
    Dim labelp2 As Label                    'Δήλωση για τα Labels που θα μούν
    για τις πόλεις (Αθήνα)
    Dim labelp3 As Label                    'Δήλωση για τα Labels που θα μούν
    για τις πόλεις (Χανιά)
    Dim labelp4 As Label                    'Δήλωση για τα Labels που θα μούν
    για τις πόλεις (Ηράκλειο)
    Dim town_xy(5,4) As Int                 'Δήλωση πίνακα με συντεταγμένες
    πόλεων
    Dim Town_Select As Int                  'Αριθμός επιλεγμένης πόλης που μας
    δείχνει και την στήλη του πίνακα (0=Θεσσαλονίκη, 1=Αθήνα, 2=Χανιά,
    3=Ηράκλειο)
    Dim hliofaneia(6) As String             'Πίνακας μεταγλώττισης τιμών σε
    εικόνες καιρού
    Dim anemos(8) As String                 'Πίνακας μεταγλώττισης τιμών σε
    εικόνες κατεύθυνσης ανέμου
    Dim Town_Name(4) As String              'Πίνακας ονομάτων για τις εικόνες
    πόλεων
    Dim File_Names As String                'Όνομα αρχείου για Download από τον
    FTP
    Dim temp(5,4) As Int                    'Πίνακας αποθήκευσης καιρού από
    αρχείο
```

```

        Dim Lost_Picture As Boolean           'Μεταβλητή όπου αποθηκεύω
αποτέλεσμα ελέγχου απώλειας αρχείων φωτογραφιών
        Dim labp1 As Label                   'Label όπου θα προβάλλω το
αποτέλεσμα θερμοκρασίας
        Dim labp2 As Label                   'Label όπου θα προβάλλω τα
μποφορ
        Dim town_img As ImageView
        Dim Img_t1 As ImageView
        Dim Img_t2 As ImageView
        Dim keim1 As Label
        Dim keim2 As Label
        Dim Panel1 As Panel
        Dim Label1 As Label
        Dim apotelesma As Panel
        Dim Button1 As Button
End Sub

Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("dokimh_xy")
    'Ενεργοποίηση του Layout
    picture1.Bitmap = LoadBitmap(File.DirAssets, "xarths.jpg")  'Φόρτωση
χάρτη ως εικόνα πάνω σε ImageView του Designer
    labelp1.Initialize("thessaloniki")
    'Ορίζω τα Labels των πόλεων
    labelp2.Initialize("athens")
    labelp3.Initialize("chania")
    labelp4.Initialize("irakleion")
    labelp1.text = "Θεσσαλονίκη"
    'Ορίζω το κείμενο για κάθε Label πόλης
    labelp2.text = "Αθήνα"
    labelp3.text = "Χανιά"
    labelp4.text = "Ηράκλειο"
    labelp1.textColor = Colors.Green
    'Ορίζω το χρώμα για κάθε Label πόλης
    labelp2.textColor = Colors.Green
    labelp3.textColor = Colors.Green
    labelp4.textColor = Colors.Green
    Activity.AddView(labelp1, 78, 64, 150, 30)
    'Προσθέτω το View και ορίζω θέση τοποθέτησης
    Activity.AddView(labelp2, 95, 153, 100, 30)
    Activity.AddView(labelp3, 91, 242, 100, 30)
    Activity.AddView(labelp4, 140, 243, 150, 30)
    town_xy(0, 0) = 74
    'Ορίζω τις συντεταγμένες για κάθε πόλη
    town_xy(0, 1) = 150
    'Οι γραμμές είναι x1, x2, y1, y2
    town_xy(0, 2) = 60
    'Οι στήλες είναι Θεσσαλονίκη, Αθήνα, Χανιά, Ηράκλειο
    town_xy(0, 3) = 84
    town_xy(1, 0) = 91
    town_xy(1, 1) = 134
    town_xy(1, 2) = 149
    town_xy(1, 3) = 171
    town_xy(2, 0) = 87
    town_xy(2, 1) = 122
    town_xy(2, 2) = 238

```

```

town_xy(2,3) = 263
town_xy(3,0) = 136
town_xy(3,1) = 195
town_xy(3,2) = 239
town_xy(3,3) = 264
hliofaneia(0) = "hlios.jpg"
'Mεταγλώττιση τιμών σε ηλιοφάνεια
hliofaneia(1) = "hlios_syn.jpg"
hliofaneia(2) = "synnefa.jpg"
hliofaneia(3) = "broxh.jpg"
hliofaneia(4) = "xioni.jpg"
hliofaneia(5) = "kategida.jpg"
anemos(0) = "a.jpg"
'Mεταγλώττιση τιμών σε κατεύθυνση ανέμου
anemos(1) = "ba.jpg"
anemos(2) = "b.jpg"
anemos(3) = "bd.jpg"
anemos(4) = "d.jpg"
anemos(5) = "nd.jpg"
anemos(6) = "n.jpg"
anemos(7) = "na.jpg"
Town_Name(0) = "thessaloniki.jpg"
Town_Name(1) = "athina.jpg"
Town_Name(2) = "xania.jpg"
Town_Name(3) = "hrakleio.jpg"
Town_Select = 4
'Αρχικοποιώ τη μεταβλητή σε τιμή που δεν υπάρχει πόλη
FTP.Initialize("FTP","data-
expert.dyndns.info","21","maria11992","m1n8k4a5") 'Αρχικοποιώ το
FTP για το Connection
FTP.PassiveMode = True

If FTP.IsInitialized = True Then
'Ζητώ την λίστα αρχείων και δείχνω μήνυμα αν δεν μπορώ να συνδεθώ
FTP.List("/android")
'Tο αρχείο το διαβάζω από τη μνήμη μόνο αν δεν μπορώ α συνδεθώ
Else
Msgbox ("Η σύνδεση με τον διακομιστή δεν είναι εφικτή.", "FTP
Connection")
End If
DoEvents
'Αφήνω την εφαρμογή λίγο ελεύθερη
Lost Picture = False
'Ορίζω ότι μου λείπουν αρχεία από εικόνες καιρού και ψάχνω να δω αν μου
λείπει καμία
For i = 0 To 7 'και επειδή οι 3 πίνακας
έχουν λιγότερες θέσεις, σταματώ στο 4 και το 3. Προσοχή, θα κατεβάσει αρχεία
έστω και ένα αν λείπει
If i < 6 AND File.Exists(File.DirInternal, hliofaneia(i)) = False
Then Lost_Picture = True
If i < 4 AND File.Exists(File.DirInternal, Town_Name(i)) = False
Then Lost_Picture = True
If File.Exists(File.DirInternal, anemos(i)) = False Then
Lost_Picture = True
Next

```

```

        If Lost_Picture = True Then
            'Αν τελικά λείπει...
            FTP1.Initialize("FTP","data-
expert.dyndns.info","21","maria11992","m1n8k4a5")           'Αρχειοποιώ το
FTP για το Connection
            FTP1.PassiveMode = True
            If FTP1.IsInitialized = True Then
                'Ζητάω την λίστα αρχείων και δείχνω μήνυμα αν δεν μπορώ να συνδεθώ
                FTP1.List("/android/images")
            Else
                MsgBox ("Εντοπίστηκε πρόβλημα στην εφαρμογή", "Internal
Error")
            End If
        End If
    End Sub

Sub FTP_ListCompleted (ServerPath As String, Success As Boolean, Folders() As
FTPEntry, Files() As FTPEntry)
    If Success = False Then
        'Ελέγχει αν η εντολή listindex του FTP έχει
ολοκληρωθεί επιτυχώς
        MsgBox ("Σφάλμα φόρτωσης νέων τιμών καιρού.", "Σφάλμα Σύνδεσης")
        'Αν όχι, βγάζει σφάλμα
        If File.Exists(File.DirInternal, "temp.csv") Then
            Dim cells() As String
            'Ορίζω τα κελιά για διάβασμα αρχείου CSV
            Dim su As StringUtils
            Dim list1 As List
            'Ορίζω μια λίστα για να αποθηκεύσω τα κελιά
            list1.Initialize
            list1 = su.LoadCSV(File.DirInternal, "temp.csv", ";")
            'Αποθηκεύω το αρχείο στην λίστα
            For i = 0 To list1.Size - 2
                'Διαβάζω μια-μια τις γραμμές της λίστας και αποθηκεύω στον πίνακα
καιρού
                cells = list1.get(i+1)
                temp(0,i) = cells(0)
                temp(1,i) = cells(1)
                temp(2,i) = cells(2)
                temp(3,i) = cells(3)
            Next
        End If
    Else
        For i = 0 To Files.Length - 1
            'Αν ναι, ελέγχει τα ονόματα των αρχείων στον φάκελο του
ListIndex και τα κατεβάζει
            FTP.DownloadFile("/android/" & Files(i).Name, False,
File.DirInternal, Files(i).Name)
        Next
    End If
End Sub

Sub FTP1_ListCompleted (ServerPath1 As String, Success1 As Boolean,
Folders1() As FTPEntry, Files1() As FTPEntry)

```



```

        If Success1 = False Then
            'Ελέγχει αν η εντολή listindex του FTP έχει
            ολοκληρωθεί επιτυχώς
            MsgBox ("Σφάλμα φόρτωσης εικόνων.", "Σφάλμα Σύνδεσης") 'Αν
            όχι, βγάζει σφάλμα
        Else
            For i = 0 To Files1.Length - 1
                'Αν ναι, ελέγχει τα ονόματα των αρχείων στον φάκελο
                του ListIndex και τα κατεβάζει
                FTP1.DownloadFile("/android/images" & Files1(i).Name,
                False, File.DirInternal, Files1(i).Name)
                DoEvents
            Next
        End If
    End Sub

Sub FTP_DownloadCompleted (ServerPath As String, Success As Boolean)
    If Success = False Then
        MsgBox ("Η λήψη στοιχείων καιρού δεν είναι εφικτή", "Error
        Message") 'Ελέγχει αν έγινε το download στο αρχείο καιρού
    Else
        'Αν όχι, βγάζει μήνυμα
        Dim cells() As String
        'Αν ναι, διαβάζει και γεμίζει τον πίνακα τιμών
        καιρού

        Dim su As StringUtils
        Dim list1 As List
        list1.Initialize
        If File.Exists(File.DirInternal, "temp.csv") Then
            list1 = su.LoadCSV(File.DirInternal, "temp.csv", ";")
            For i = 0 To list1.Size - 2
                cells = list1.get(i+1)
                temp(0,i) = cells(0)
                temp(1,i) = cells(1)
                temp(2,i) = cells(2)
                temp(3,i) = cells(3)
            Next
        End If
    End Sub

Sub FTP1_DownloadCompleted (ServerPath1 As String, Success1 As Boolean)
    If Success1 = False Then MsgBox ("Η λήψη αρχείων εικόνας δεν είναι
    εφικτή", "Error Message") 'Ελέγχει αν έπρεπε να κατεβάσει εικόνες και τα
    κατάφερε
End Sub

Sub Activity_Resume

End Sub

Sub Activity_Pause (UserClosed As Boolean)

End Sub

Sub Activity_Touch (Action As Int, X As Float, Y As Float)

```

```

Town_Select = 4
'Labell1.text = "X= " & X & CRLF & "Y= " & Y
For i = 0 To 3
    If (X > town_xy(i,0) AND X < town_xy(i,1)) AND (Y > town_xy(i,2)
AND Y < town_xy(i,3)) Then
        Town_Select = i
    End If
Next
If Town_Select > -1 AND Town_Select < 4 Then
    apotelesma.Visible = True
    apotelesma.BringToFront
    'Labell1.Text = Labell1.Text & CRLF & Town_Select & " " &
temp(Town_Select,0) & "," & temp(Town_Select,1) & "," & temp(Town_Select,2) &
"," & temp(Town_Select,3)
    If File.Exists(File.DirInternal, Town_Name(Town_Select)) Then
        labp1.Initialize("thermokrasia")
        labp2.Initialize("mpofor")
        keim1.Initialize("text1")
        keim2.Initialize("text2")
        labp1.Text = temp(Town_Select,0) & " C"
        labp2.Text = temp(Town_Select,2) & " B"
        labp1.textColor = Colors.Green
        'Ορίζω το χρώμα για κάθε Label πόλης
        labp2.textColor = Colors.Green
        keim1.text = "Η θερμοκρασία αυτή τη στιγμή είναι..."
        keim2.text = "Η ένταση του ανέμου είναι..."
        keim1.textColor = Colors.Red
        keim2.textColor = Colors.Red
        keim1.TextSize = 10
        keim2.TextSize = 10
        For j = apotelesma.NumberOfViews -1 To 0 Step -1
            apotelesma.GetView(j).RemoveView
        Next
        town_img.Initialize("eikona_polhs")
        town_img.Bitmap = LoadBitmap(File.DirInternal,
Town_Name(Town_Select))
        Img_t1.Initialize ("Fwto_Hlios")
        Img_t1.Bitmap = LoadBitmap(File.DirInternal,
hliofaneaia(temp(Town_Select,1)))
        Img_t2.Initialize ("Fwto_anemos")
        Img_t2.Bitmap = LoadBitmap(File.DirInternal,
anemos(temp(Town_Select,3)))
        apotelesma.AddView(Img_t1, 0,1,100,100)
        apotelesma.AddView(keim1, 100,35,85,30)
        apotelesma.AddView(labp1, 130,70,130,30)
        apotelesma.AddView(Img_t2, 0,70,100,100)
        apotelesma.AddView(keim2, 100,105,100,30)
        apotelesma.AddView(labp2, 130,120,100,30)
        apotelesma.AddView(town_img, 0,128,195,160)
    End If
Else
    apotelesma.Visible = False
End If
End Sub

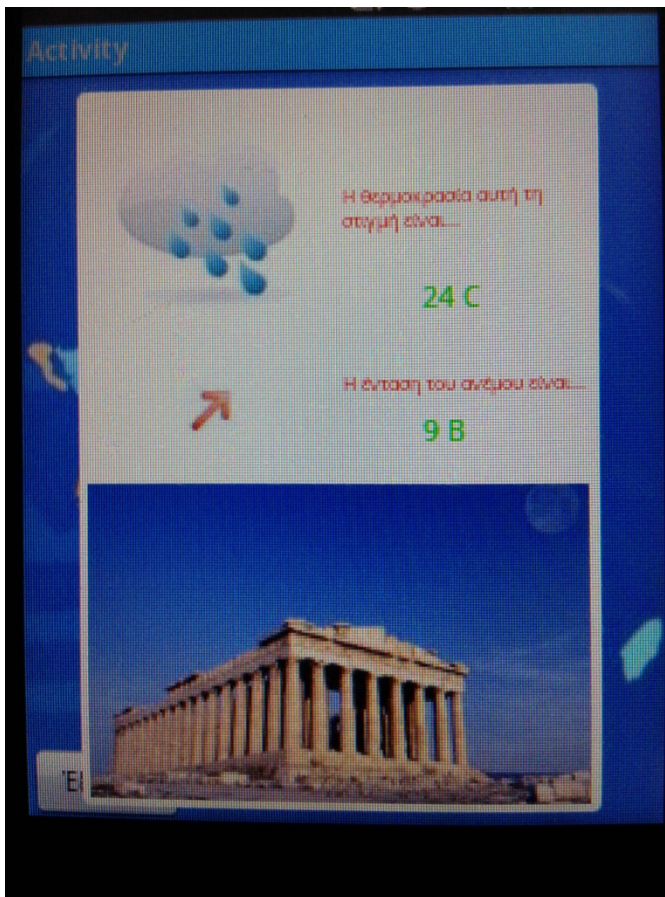
Sub Button1_Click

```

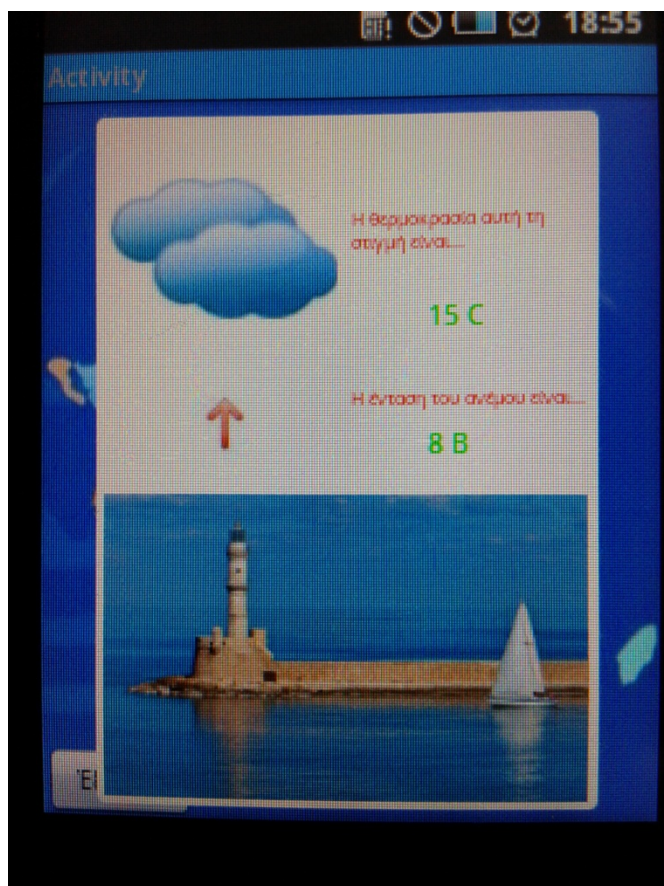
```
ExitApplication  
End Sub
```



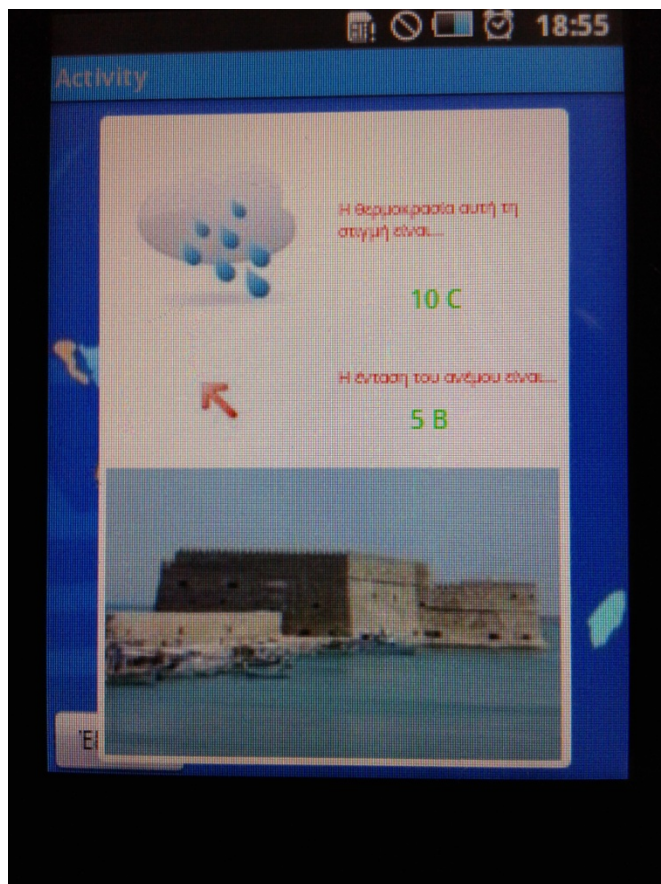
Στιγμιότυπο της οθόνης για την πόλη της Θεσσαλονίκης



Στιγμιότυπο της οθόνης για την πόλη της Αθήνας



Στιγμιότυπο της οθόνης για την πόλη των Χανίων



Στιγμιότυπο της οθόνης για την πόλη του Ηρακλείου