

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ



Τμήμα Μηχανικών Η/Υ, Τηλεπικοινωνιών & Δικτύων



## Εφαρμογή της μεθόδου VAS σε περιβάλλον Android.

**Διπλωματική Εργασία**

**Μπερκάκης Αντώνης**

Υπεύθυνος καθηγητής : Αλκιβιάδης Γ. Ακρίτας  
Αναπληρωτής Καθηγητής

Επιβλέπων Καθηγήτρια: Ασπασία Δασκαλοπούλου  
Επίκουρη καθηγήτρια

**Βόλος, 2013**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ  
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

## Εφαρμογή της μεθόδου VAS σε περιβάλλον Android.

Διπλωματική Εργασία

Μπερκάκης Αντώνης

Υπεύθυνος καθηγητής : Αλκιβιάδης Γ. Ακρίτας

Αναπληρωτής Καθηγητής

Επιβλέπων Καθηγήτρια: Ασπασία Δασκαλοπούλου

Επίκουρη καθηγήτρια

Εγκρίθηκε από την Διμελή Εξεταστική επιτροπή τον Φεβρουάριο του 2013.

Υπογραφή

Υπογραφή

(.....)

(.....)



## Ευχαριστίες

---

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Αλκιβιάδη Ακρίτα για την αμέριστη συμπαράσταση, υποστήριξη και εμπιστοσύνη που μου έδειξε κατά την διάρκεια εκπόνησης της διπλωματικής μου εργασίας.

Ευχαριστώ πολύ την καθηγήτρια Ασπασία Δασκαλοπούλου που δέχτηκε να αναλάβει την επίβλεψη της διπλωματικής εργασίας μου.

Ευχαριστώ τον δημιουργό της βιβλιοθήκης Giac , Bernard Parisse του οποίου η βοήθεια διευκόλυνε την διπλωματική μου εργασία.

Τέλος , θα ήθελα να ευχαριστήσω την οικογένεια μου για την συμπαράσταση και υποστήριξη που μου προσφέρει σε κάθε βήμα μου.

Μπερκάκης Αντώνης

Βολος,2013

## Περιεχόμενα

<b>ΕΙΣΑΓΩΓΗ ΣΤΟ ANDROID .....</b>	<b>6</b>
ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ .....	6
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΕΚΔΟΣΕΙΣ ANDROID.....	8
<b>ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΓΙΑ ANDROID.....</b>	<b>11</b>
ANDROID SDK .....	11
ANDROID NDK.....	12
ANDROID EMULATOR .....	14
QT NECESSITAS FOR ANDROID .....	15
MINISTRO .....	17
<b>Ο ΑΛΓΟΡΙΘΜΟΣ VAS.....</b>	<b>20</b>
ΕΙΣΑΓΩΓΗ .....	20
ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ & ΙΣΤΟΡΙΚΗ ΑΝΑΣΚΟΠΗΣΗ .....	20
<b>QT NECESSITAS FOR ANDROID .....</b>	<b>22</b>
ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΕΠΕΞΗΓΗΣΗ QT.....	22
<b>ΕΦΑΡΜΟΓΗ REAL ROOTS .....</b>	<b>26</b>
ΠΡΩΤΗ ΕΠΑΦΗ ΜΕ ΤΙΣ ΒΙΒΛΙΟΘΗΚΕΣ ΚΑΙ ΤΟ PROJECT ΤΟΥ QT .....	26
ΜΕΤΑΓΛΩΤΤΙΣΗ ΚΑΙ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΜΑΣ .....	34
<i>Android Emulator &amp; Physical Device</i> .....	36
ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ REALROOTS .....	38
ΛΕΙΤΟΥΡΓΙΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ REALROOTS .....	38
ΔΙΕΠΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ REALROOTS .....	39
<b>ΑΝΕΒΑΣΜΑ ΕΦΑΡΜΟΓΗΣ ΣΤΟ GOOGLE PLAY .....</b>	<b>45</b>
ΜΕΤΑΠΟΙΗΣΗ ΤΟΥ ANDROID MANIFEST .....	45
ΥΠΟΓΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ ΜΕ RSA ΚΛΕΙΔΙ .....	46
GOOGLE PLAY .....	50

## Ευρετήριο Εικόνων

Εικόνα 1. HTC G1.....	6
Εικόνα 2. Google Nexus One.....	7
Εικόνα 3. Android Market και Google Play.....	7
Εικόνα 4. Διαγράμμα της αρχιτεκτονικής του Android.....	9
Εικόνα 5. developer.android.com , η σελίδα διανομής του SDK.....	11
Εικόνα 6. <a href="http://developer.android.com/tools/sdk/ndk/index.html">http://developer.android.com/tools/sdk/ndk/index.html</a> .....	13
Εικόνα 7. Android Emulator το οποίο “τρέχει” την έκδοση ice cream.....	14
Εικόνα 8. <a href="http://qt.nokia.com/products/">http://qt.nokia.com/products/</a> .....	15
Εικόνα 9. <a href="http://labs.qt.nokia.com/2011/02/28/necessitas/">http://labs.qt.nokia.com/2011/02/28/necessitas/</a> .....	16
Εικόνα 10. μήνυμα για εγκατάσταση Ministro.....	17
Εικόνα 11. Γραφικό Περιβάλλον για την επιλογή των απαιτούμενων βιβλιοθηκών.....	19
Εικόνα 12. οδηγίες εγκατάστασης σε περιβάλλον Linux.....	22
Εικόνα 13. Necessitas Qt Installer.....	23
Εικόνα 14. Qt Necessitas installer options .....	24
Εικόνα 15. Necessitas Qt Creator.....	25
Εικόνα 16. directory Android Qt project.....	26
Εικόνα 17. ενδεικτικός κώδικας για εισαγωγή νέου Layout .....	27
Εικόνα 18. εισαγωγή Qt Designer Form .....	27
Εικόνα 19. Qt Designer Form Class.....	28
Εικόνα 20. Class Name .....	29
Εικόνα 21. my_name.ui.....	29
Εικόνα 22. διαφορές διαθέσιμες επιλογές για το my_name.ui.....	30
Εικόνα 23. ui as xml.....	31
Εικόνα 24. Push Button Layout.....	32
Εικόνα 25. push Button .....	32
Εικόνα 26. Connecting a button using ( Go to Slot).....	33
Εικόνα 27. On_pushButton_clicked().....	34
Εικόνα 28. deploy configurations.....	35
Εικόνα 29. Android Emulator .....	36
Εικόνα 30. Creating an emulator .....	37
Εικόνα 31. Amdroid Emulator .....	37
Εικόνα 32. Αρχική οθόνη εφαρμογής .....	40
Εικόνα 33. More Function Layout .....	41
Εικόνα 34. Advanced .....	42
Εικόνα 35. Εφαρμογή της μεθόδου VAS.....	43
Εικόνα 36. Approximation VAS Roots.....	44
Εικόνα 37. AndroidManifest.xml.....	45
Εικόνα 38. Package configuration.....	47
Εικόνα 39. Keystore .....	48
Εικόνα 40. Τελικό στάδιο Package configurations .....	49
Εικόνα 41. Upload applications instructions.....	50

# Εισαγωγή στο Android

## Ιστορική Ανάδρομη

Τον Ιούλιο του 2005 η Google αγόρασε μια μικρή εταιρία με έδρα το Παλο Αλτο (Καλιφόρνια) που ονομάζετε Android INC.

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 τη δημιουργία μιας πλατφόρμας ανοιχτού κώδικα που θα έχει ενσωματωμένο το λογισμικό Linux και θα εκπροσωπείται από μια νέα συστάδα εταιριών που θα καλείται Open Handset Alliance.

Η πρώτη συσκευή που βγήκε στην αγορά με το Android ήταν το HTC G1 (android 1.0) που προωθήθηκε από την T-mobile USA τον Οκτώβριο του 2008. Επίσης προωθήθηκε μια τροποποιημένη έκδοση με πρόσβαση Root\* για προγραμματιστές με την ονομασία Dev Phone-1.



Εικόνα 1. HTC G1

\*Root= όπως το super user στο λειτουργικό linux , λογαριασμός με πλήρη δικαιώματα.

Στις 5 Ιανουαρίου 2010 η Google δημιούργησε το δικό της τηλέφωνο με την ονομασία Google Nexus One(Android 2.1).



Εικόνα 2. Google Nexus One

Στις 28 Αύγουστου 2008 η Google ανακοινώσε το Android Market ( όπου από 6 Μάρτιου 2012 ονομάζετε Google Play ) και σταδιακά άρχισε να υποστηρίζει εφαρμογές επί πληρωμή σε διαφορές χώρες ξεχωριστά.



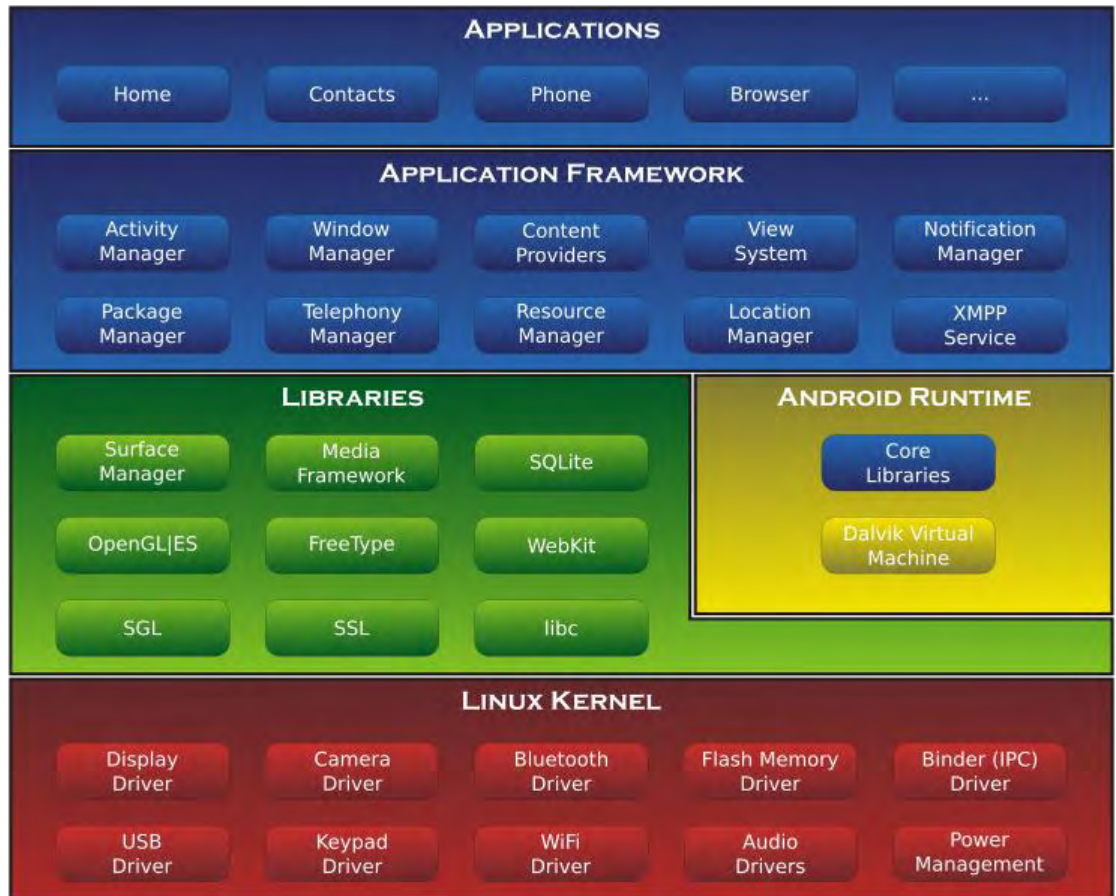
Εικόνα 3. Android Market και Google Play



## Αρχιτεκτονική και εκδόσεις Android

### Αρχιτεκτονική

- Ο πυρήνας Linux:  
Πρόκειται για το χαμηλότερο επίπεδο του συστήματος. Είναι υπεύθυνο για την αλληλεπίδραση με το hardware. Περιέχει όλους τους απαραίτητους drivers γιατί. Βασίζετε στο πυρήνα GNU Linux 2.6 στον οποίο η ομάδα της Google έχει προσθέσει ορισμένες τροποποιήσεις.
- Βιβλιοθήκες:  
Πάνω από τον πυρήνα βρίσκονται οι βιβλιοθήκες που περιλαμβάνει και το σύστημα. Συνήθως είναι προγραμματισμένες σε C/C++ και κανονικά δεν έχουμε πρόσβαση σε αυτές τις βιβλιοθήκες άμεσα, καθώς χρησιμοποιούμε αυτές του ανώτερου επιπέδου (το πλαίσιο) .
- Εικονικό μηχανήμα (Dalvik):  
Τόσο για να αποφευχθούν θέματα αδειοδότησης , όσο και για να προσπαθήσουμε να βελτιστοποιήσουμε ή να προσπαθήσουμε να καταπολεμήσουμε τον υπάρχοντα κατακερματισμό σε JAVA ME, η Google κατέληξε στο δικό της εικονικό μηχανήμα.  
  
Ο σχεδιασμός του δίνει την δυνατότητα ταυτόχρονης λειτουργίας πολλών εικονικών μηχανημάτων. Έτσι , κάθε εφαρμογή τρέχει στο δικό της εικονικό μηχανήμα (κάτι που προσθέτει άλλο ένα επίπεδο ασφάλειας ).
- Πλαίσιο εφαρμογών:  
Παρέχει ένα πλαίσιο ανάπτυξης που τυποποιεί και διευκολύνει τον προγραμματισμό των εφαρμογών Android. Αυτές οι βιβλιοθήκες είναι γραμμένες σε Java.  
Ο σχεδιασμός του πλαισίου επικεντρώνετε στην επαναχρησιμοποίηση των στοιχείων. Εφόσον τηρούνται οι κανόνες ασφάλειας που περιλαμβάνει , οποιαδήποτε εφαρμογή μπορεί να δημοσιεύσει τις δυνατότητες της, έτσι ώστε οποιαδήποτε άλλη εφαρμογή να μπορεί να τις χρησιμοποιεί αργότερα.
- Εφαρμογές:  
Το Android , από προεπιλογή , διαθέτει μια σειρά προεγκαταστημένων εφαρμογών. Π.χ. κατάλογο επαφών ,ημερολόγιο, διαχειριστή SMS κλπ. Ο χρήστης μπορεί να εγκαταστήσει νέες εφαρμογές χάρη στο Google Play και αλλά παρόμοιες προγράμματα.



Εικόνα 4. Διαγραμμα της αρχιτεκτονικής του Android.

## Εκδόσεις του Android

Κάθε νέα έκδοση του Android περιλαμβάνει νέα χαρακτηριστικά και επιλύει προβλήματα των προηγούμενων. Κάθε έκδοση σχετίζεται με ένα αριθμητικό API Level που δίνει την δυνατότητα αναγνώρισης της. Έτσι, το σύστημα μπορεί να γνωρίζει αν είναι σε θέση να τρέξει μια συγκεκριμένη εφαρμογή.

Μερικές από τις αρχικές εκδόσεις μπορούμε να δούμε παρακάτω.

- Android 1.1
- Ημερομηνία :Φεβρουάριος 2009.
- API Level:2

- Android 1.5 (Cupcake)
- Ημερομηνία :Μάιος 2009.
- API Level:3

- Android 1.6 ( Donut )
- Ημερομηνία : Οκτώβριος 2009.
- API Level:4

- Android 2.0/2.1 ( Éclair )
- Ημερομηνία :Ιανουάριος 2010.
- API Level:7

- Android 2.2 (Froyo)
- Ημερομηνία :Ιανουάριος 2010.
- API Level:8

- Android 2.3 ( Gingerbread)
- Ημερομηνία :Νοέμβριος 2010.
- API Level:9

Τελευταίο στην σειρά με πρόσφατη κυκλοφορία( Αύγουστος 2012)

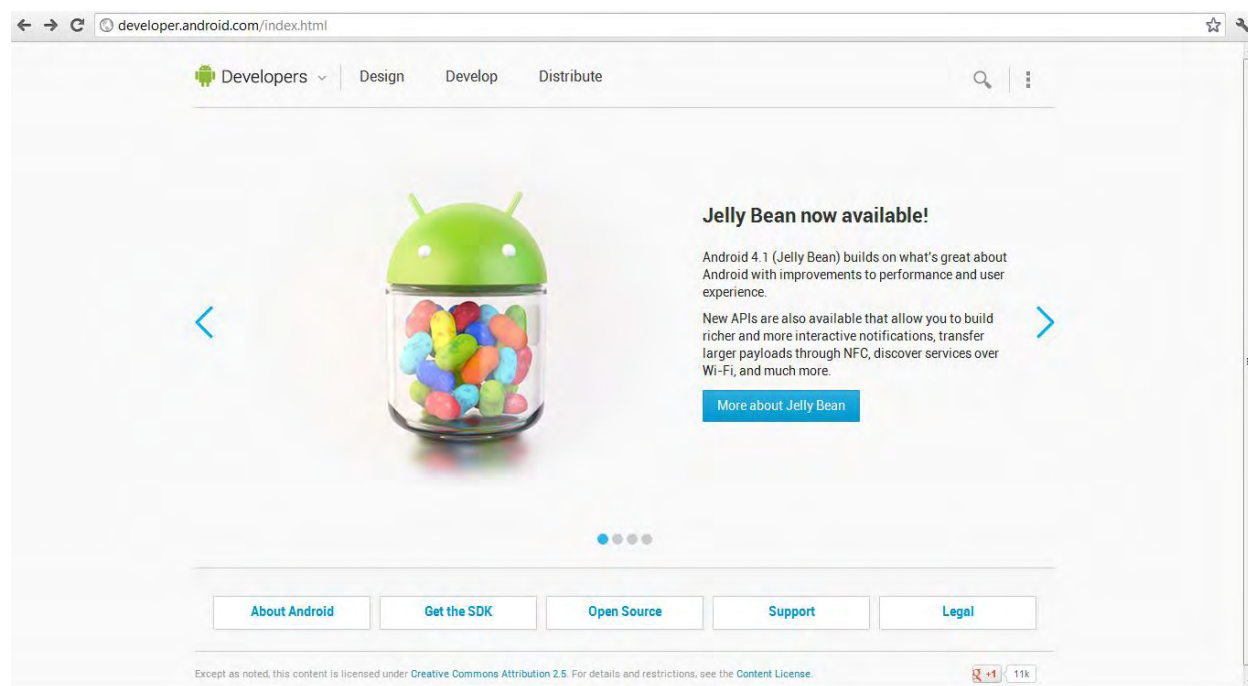
- Android 4.1 (Jelly Bean )
- API Level:16

# Περιβάλλον Προγραμματισμού για Android

## Android SDK

Το Android SDK (Software development kit) περιλαμβάνει μια σειρά από εργαλεία για την ανάπτυξη εφαρμογών για Android. Πιο συγκεκριμένα περιλαμβάνει debugger, διάφορες βιβλιοθήκες για βασικές λειτουργίες (όπως χειρισμό διαφόρων αισθητήρων της κάθε συσκευής). Η γλώσσα προγραμματισμού του Android είναι η Java αν και θα δούμε παρακάτω ότι η ίδια η εταιρία έχει εισάγει ένα εργαλείο με το οποίο μπορούμε να προγραμματίσουμε σε C/C++ για τον κύριο λόγο ώστε να έχουμε την δυνατότητα να χρησιμοποιήσουμε διάφορες βιβλιοθήκες που υπάρχουν ήδη σε C/C++ χωρίς να υπάρχει λόγος να ξαναγραφτούν σε Java.

Οι πλατφόρμες στις οποίες υποστηρίζετε το SDK είναι οι τρεις πιο δημοφιλείς, δηλαδή windows XP και μετά, MAC OS X 10.5.8 και μετά όπως και κάθε σύγχρονη διανομή Linux.



Εικόνα 5. developer.android.com, η σελίδα διανομής του SDK

Το SDK χρειάζεται κάποιο IDE (integrated development environment) στο οποίο μπορεί να ενσωματωθεί και να λειτουργήσει για την ανάπτυξη εφαρμογών.

Τέτοια IDE είναι το Eclipse (<http://www.eclipse.org/>), το Netbeans (<http://netbeans.org/>) όπως και το Necessitas (<http://labs.qt.nokia.com/2011/02/28/necessitas/>) το οποίο είναι ένα open source IDE τροποποιημένο για ανάπτυξη Android και βρίσκεται σε alpha στάδιο.

Αυτήν την στιγμή (Αύγουστος 2012) το Android SDK βρίσκεται στην έκδοση 20.0.1.

## Android NDK

Το Android NDK (native development kit) όπως αναφέρθηκε και προηγουμένως είναι ένα πακέτο από εργαλεία που επιτρέπει στον κάθε προγραμματιστή να ενσωματώσει διάφορα κομμάτια της εφαρμογής του τα όποια είναι σε C/C++ .

Ο κύριος λόγος που προτείνετε από την ίδια σελίδα του NDK της Google είναι για την επαναχρησιμοποίηση διαφόρων βιβλιοθηκών που ήδη έχουν γραφεί σε C/C++ και πιθανώς είναι και βελτιστοποιούμενες στην συγκεκριμένη γλώσσα. Το ίδιο το πακέτο NDK περιέχει από μόνο του αρκετές γνώστες βιβλιοθήκες. Π.χ. η “Math.h”. Αυτός είναι και ο κύριος λόγος που θα το χρειαστούμε και εμείς καθώς θα χρησιμοποιήσουμε διάφορα πακέτα open source όπως “Tommath” και “Giac”.

Επίσης το ίδιο το NDK περιέχει διαφόρους compilers ώστε να μπορείς να κάνεις μεταγλώττιση των βιβλιοθηκών στην κάθε αρχιτεκτονική.

Developers ▾ | Design | **Develop** | Distribute

Android Training | API Guides | Reference | **Tools**

---

Developer Tools

- Download ▾
- Installing the SDK ▾
- Exploring the SDK
- NDK**
- Workflow ▾
- Tools Help ▾
- Revisions ▾
- Extras ▾
- Samples
- ADK ▾

## Android NDK

The NDK is a toolset that allows you to implement parts of your app using native-code languages such as C and C++. For certain types of apps, this can be helpful so that you may reuse existing code libraries written in these languages and possibly increased performance.

Before downloading the NDK, you should understand that **the NDK will not benefit most apps**. As a developer, you need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not result in a noticeable performance improvement, but it always increases your app complexity. In general, you should only use the NDK if it is essential to your app – never because you simply prefer to program in C/C++.

Typical good candidates for the NDK are self-contained, CPU-intensive operations that don't allocate much memory, such as signal processing, physics simulation, and so on. When examining whether or not you should develop in native code, think about your requirements and see if the Android framework APIs provide the functionality that you need.

**IN THIS DOCUMENT**

- [Downloads](#)
- [Revisions](#)
- [System and Software Requirements](#)
- [Installing the NDK](#)
- [Getting Started with the NDK](#)
- [Using the NDK](#)
- [Contents of the NDK](#)
- [Development tools](#)
- [Documentation](#)
- [Sample apps](#)

### Downloads

Platform	Package	Size	MD5 Checksum
Windows	<a href="#">android-ndk-r8b-windows.zip</a>	188724991 bytes	6d290d4f2729ef2063c5ae5b1e335622
Mac OS X (intel)	<a href="#">android-ndk-r8b-darwin-x86.tar.bz2</a>	181255568 bytes	94fe392194ea41f8a70cfce0dee3870f
Linux 32/64-bit (x86)	<a href="#">android-ndk-r8b-linux-x86.tar.bz2</a>	160466240 bytes	6694ccc04d543500f0661a75f6c46526

### Revisions

The sections below provide information and notes about successive releases of the NDK, as denoted by revision number.

- ▾ [Android NDK, Revision 8b \(July 2012\)](#)
  - The main features of this release are a new GNU Compiler Collection (GCC) 4.6 toolchain and GNU Debugger (GDB) 7.3.x which adds debugging support for the Android 4.1 (API Level 16) system image.

Εικόνα 6. <http://developer.android.com/tools/sdk/ndk/index.html>

Αυτήν την στιγμή (Αύγουστος 2012) το Android NDK βρίσκεται στην έκδοση 8b.

## Android Emulator

Το ίδιο το Android SDK περιλαμβάνει ένα προσομοιωτή το οποίο δίνει την δυνατότητα για τον χρήστη να κάνει έλεγχο της εφαρμογής του χωρίς να χρησιμοποιήσει μια φυσική συσκευή.

Επίσης δίνει την δυνατότητα σε διαφορετικές εκδόσεις emulator , αναλύσεις , δυνατοτήτων συσκευών όπως και για το αν πρόκειται τα κινητό η tablet.

Περισσότερα για το Android emulator μπορείτε να βρείτε στην παρακάτω σελίδα

<http://developer.android.com/tools/devices/emulator.html>



Εικόνα 7. Android Emulator το οποίο “τρέχει” την έκδοση ice cream.

## Qt Necessitas for Android

Το QT Necessitas for Android είναι το IDE (integrated development environment) το οποίο επιλέχτηκε για την ανάπτυξη της εφαρμογής μας. Ο λόγος που επιλέχτηκε το συγκεκριμένο είναι ότι μπορεί να υποστηρίξει αρκετά καλά τις βιβλιοθήκες που μας είναι απαραίτητες για την εφαρμογή. Σε παρακάτω παράγραφο θα αναπτυχθούν επακριβώς οι απαιτούμενες βιβλιοθήκες.

Αρχικά το Qt SDK (<http://qt.nokia.com/products/>) το οποίο περιέχει και ένα framework για την ανάπτυξη εφαρμογών, έχει σχεδιαστεί ώστε να στοχεύει σε συσκευές Symbian OS όπως τα περισσότερα Nokia smartphones. Επίσης το Qt μπορεί να παράγει και εφαρμογές για όλα τα γνωστά λειτουργικά των Υπολογιστών (PC), Microsoft Windows, Mac OS X και Linux.



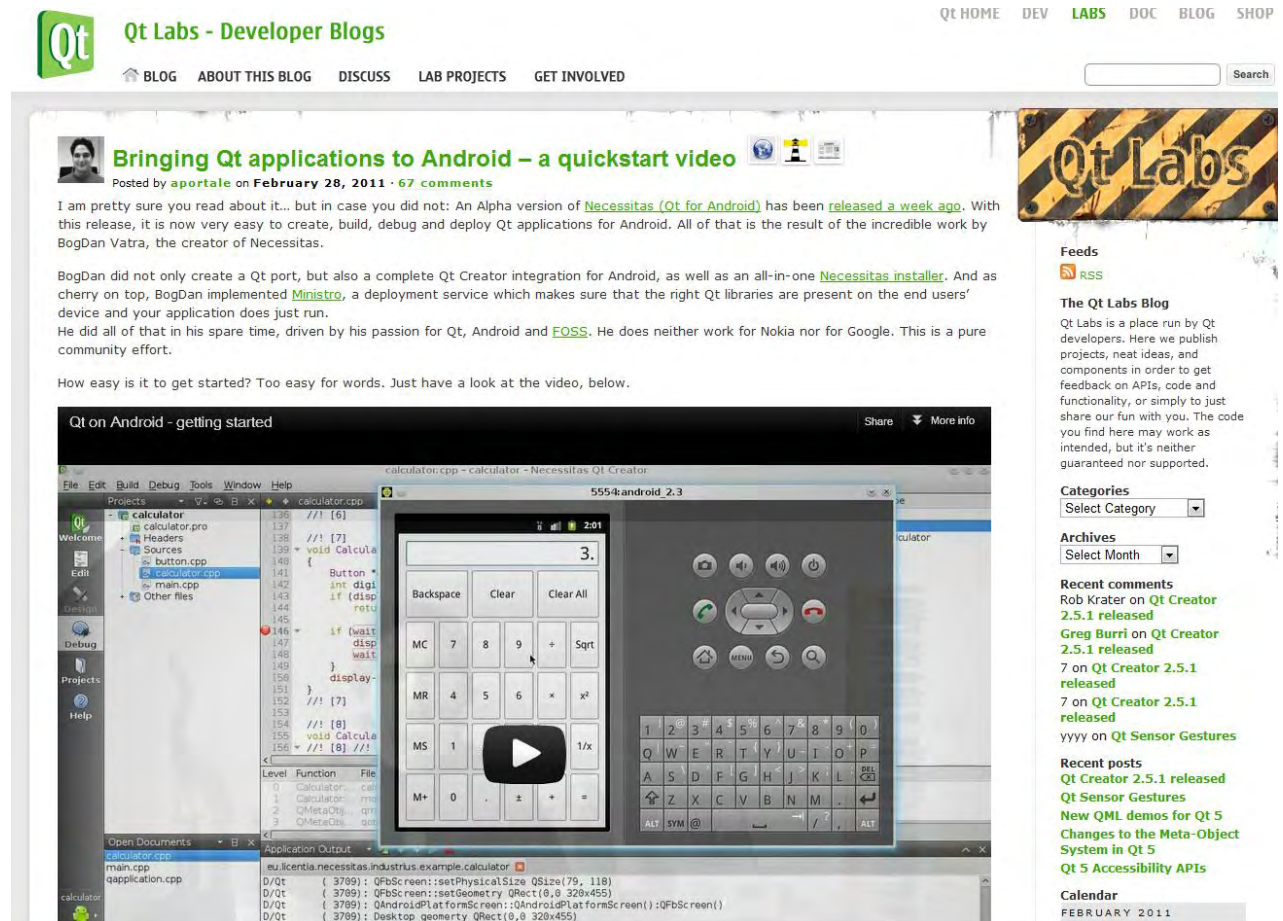
The image shows a screenshot of the Qt website homepage. At the top, there is a navigation menu with links for HOME, PRODUCTS, LEARN Qt, Qt IN USE, PARTNERS, DOWNLOADS, SUPPORT, and ABOUT US. There is also a search bar with the text "Google Custom Search" and a "Search" button. The main content area is divided into several sections:

- Qt**: A section describing the Qt SDK, which combines the Qt framework with tools for streamlining application creation for Symbian phones and Nokia N9 smartphones, as well as desktop platforms like Windows, Mac OS X, and Linux. It lists key features: Qt framework (intuitive APIs for C++ and CSS/JavaScript-like programming with Qt Quick for rapid UI creation), Qt Creator IDE (powerful cross-platform IDE with UI designer tools and on-device debugging), and Tools and toolchains (simulator, local and remote compilers, internationalization support, etc.).
- Download the Qt SDK**: A section with a "Download" button and links to Qt 4.7.4, Qt Creator 2.4, and Qt Quick.
- Get started, learn Qt**: A section with a "Learn More" button and links to Documentation, Qt Training, and Qt Courses.
- Embedded development**: A section with a "Go Mobile" button and links to Nokia devices, Embedded Linux, and Windows CE.
- Qt framework**: A section describing Qt as a cross-platform application and UI framework with APIs for C++ programming and Qt Quick for rapid UI creation. It lists features like intuitive class libraries, ease of use, and high runtime performance.
- Qt Creator IDE and tools**: A section describing Qt Creator IDE as a cross-platform IDE for the Qt SDK. It lists features like support for desktop and mobile targets, C++ and JavaScript code editor, integrated UI designer, project and build management tools, and debuggers.
- Qt Quick**: A section describing the Qt UI Creation Kit, which lets you create rich, touch-enabled UIs and apps quickly. It lists features like QML (CSS & JavaScript like language), support by Qt Creator IDE, and Qt Quick UI runtime.

Εικόνα 8. <http://qt.nokia.com/products/>



Όσον αναφορά το Qt Necessitas είναι μια open source παραλλαγή του αρχικού Qt το οποίο εκτός των προαναφερθέντων μπορεί να δημιουργήσει και εφαρμογές για Android.



Εικόνα 9. <http://labs.qt.nokia.com/2011/02/28/necessitas/>

Η παραλλαγή έχει γίνει από τον BogDan Vatra ο οποίος το βελτιώνει συνεχώς καθώς είναι σε πρώιμο στάδιο το συγκεκριμένο IDE.

Αν και αυτήν την στιγμή βρίσκεται στην έκδοση Alpha 4 , η οποία όπως αναφέρετε μάλλον θα είναι και η πρώτη beta, υπάρχει μια μεγάλη κοινότητα γύρω από το συγκεκριμένο προϊόν.

Ένα google group στο οποία έχουν πολλές συζητήσεις πάνω στο συγκεκριμένο θέμα.

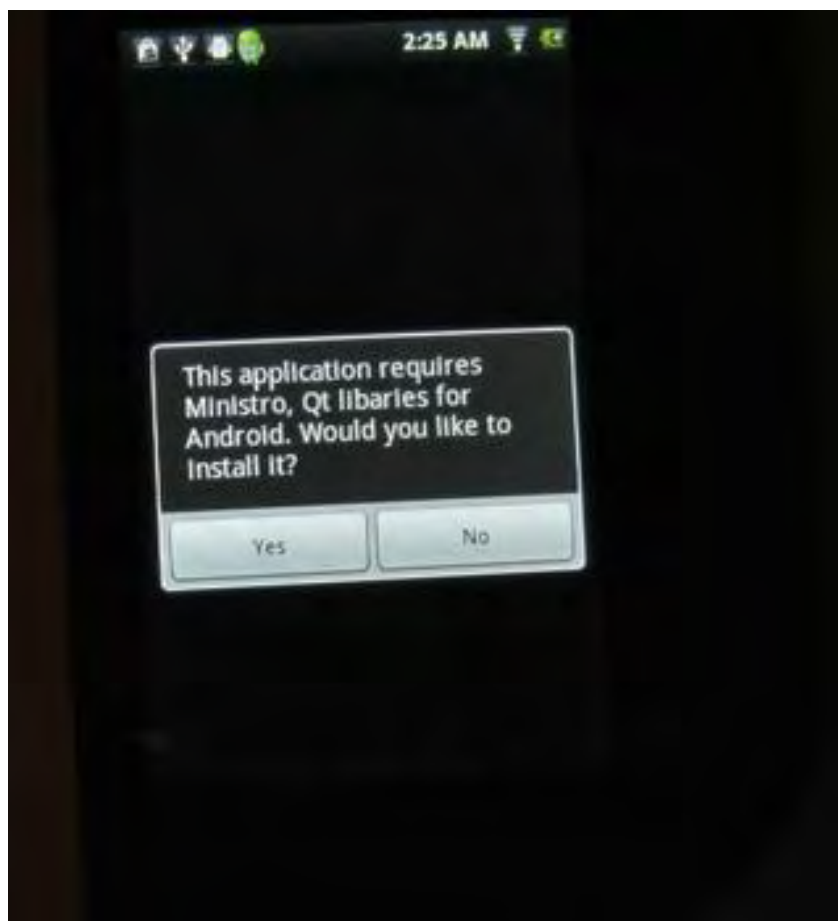
<https://groups.google.com/forum/?fromgroups#!forum/android-qt>

## Ministro

Για να μπορεί να λειτουργήσει κάθε Qt Εφαρμογή στην συσκευή Android εκτός τις απαιτούμενες βιβλιοθήκες του android χρειάζεται πλέον και τις βιβλιοθήκες των Qt εφαρμογών.

Για να εγκατασταθούν οι βιβλιοθήκες στην συσκευή την πρώτη φορά που θα «ανοίξει» μια οποιαδήποτε εφαρμογή Qt και δεν υπάρχουν οι βιβλιοθήκες θα εμφανιστεί το παρακάτω μήνυμα. Το οποίο αυτόματα προωθεί τον χρήστη στο Google Play για την εγκατάσταση της background εφαρμογής Ministro, την οποία έχει δημιουργήσει επίσης ο BogDan Vatra.

Όταν τελειώσει η εγκατάσταση πλέον η συσκευή έχει εγκατεστημένες όλες τις απαραίτητες βιβλιοθήκες και έτσι μπορεί να τρέξει οποιαδήποτε Qt εφαρμογή.



Εικόνα 10. μήνυμα για εγκατάσταση Ministro.

Η προηγούμενη διαδικασία γίνεται μόνο μια φορά στην συσκευή.

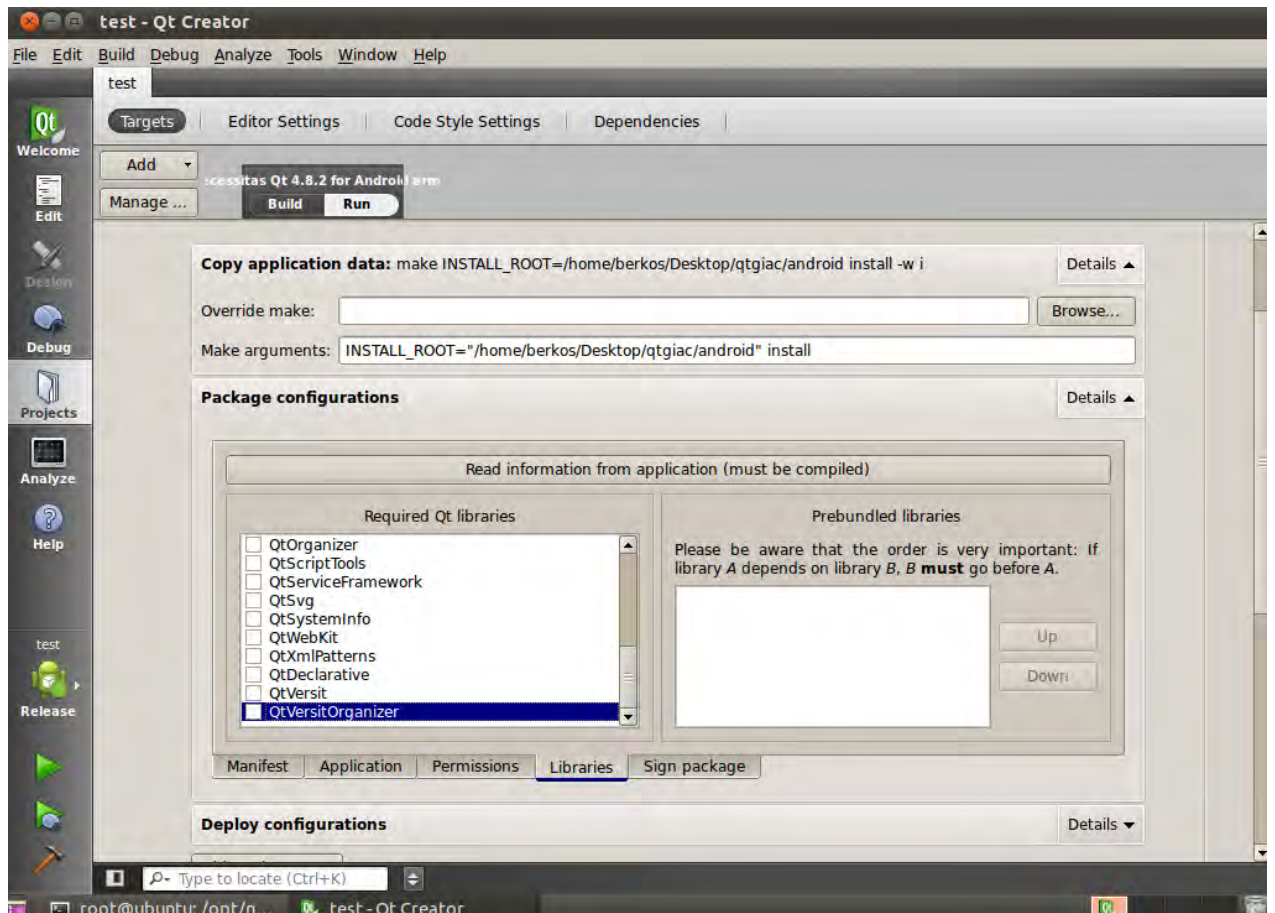
Ανάλογα με την εφαρμογή που συγγραφούμε χρειαζόμαστε και διαφορές QT βιβλιοθήκες οι κύριες βιβλιοθήκες είναι οι QtCore, QtGui.

Ενώ οι διαθέσιμες είναι :

QtFeedBack  
QtGallery  
QtNetwork  
QtSript  
QtSensors  
QtSql  
QtTest  
QtXml  
QtBearer  
QtConnectivity  
QtContacts  
QtLocations  
QtOpenGL  
QtOrganizer  
QtScriptTools  
QtServiceFrameWork  
QtSvg  
QtSystemInfo  
QtWebkit  
QtXmlpatterns  
QtDeclarative  
QtVersit  
QtVersitOrganizer

Όπως βλέπουμε μπορούμε να χρησιμοποιήσουμε το Qt ώστε να παράγουμε οποιαδήποτε android εφαρμογή η οποία είτε να χρησιμοποιεί σενσορες, είτε υπηρεσίες τοποθεσίας κ.α.

όποτε με βάση αυτές που θα επιλέξουμε στην εφαρμογή μας αυτές θα κατεβάσει το Ministro. Σε ενδεχόμενη εφαρμογή με περισσότερες απαιτούμενες βιβλιοθήκες θα παρέμβει και πάλι το Ministry το οποίο θα μας ζητήσει άδεια να κατεβάσει αυτές που λείπουν από την συσκευή.



Εικόνα 11. Γραφικό Περιβάλλον για την επιλογή των απαιτούμενων βιβλιοθηκών.

# Ο Αλγόριθμος VAS

## Εισαγωγή

Θα αναφερθούμε περιληπτικά στον αλγόριθμο VAS (Vincent-Akritas-Strzebonski) , τις βασικές έννοιες και την ιστορική του ανασκόπηση καθώς έχει γίνει ήδη λεπτομερής αναφορά από την Διπλωματική Εργασία του Σπύρου Κεχαγιά στο Τμήμα μηχανικών Υπολογιστών , Τηλεπικοινωνιών και δικτύων του Πανεπιστήμιου Θεσσαλίας στην όποια και μπορεί ο ενδιαφερόμενος να ανατρέξει για παραπάνω πληροφορίες.

## Βασικές έννοιες & ιστορική Ανασκόπηση

Απομόνωση (isolation) των πραγματικών ριζών μιας πολυωνμικής εξίσωσης είναι η διαδικασία εύρεσης πραγματικών διαστημάτων , μη τεμνόμενων μεταξύ τους , έτσι ώστε κάθε διάστημα να περιέχει ακριβώς μια πραγματική ρίζα , και κάθε πραγματική ρίζα να περιέχεται σε κάποιο διάστημα. Προσέγγιση (approximation), από την άλλη μεριά είναι η διαδικασία σμίκρυνσης των διαστημάτων απομόνωσης των ριζών τόσο, όσο να προσεγγίσουν αυτές στην επιθυμητή ακρίβεια.

Συμφώνα με τη “ Γαλλική μαθηματική Σχολή ” του 19<sup>ου</sup> αιώνα τα προβλήματα της απομόνωσης και προσέγγισης ριζών αποτελούν υποπροβλήματα του γενικού ζητήματος της επίλυσης πολυωνμικών εξισώσεων με ακέραιους συντελεστές . Επομένως , η επίλυση του γενικότερου προβλήματος ανάγεται πλέον στην επίλυση των δυο επιμέρους υποπροβλημάτων. Το πρώτο πρόβλημα, η απομόνωση ριζών είναι και το σημαντικότερο και έτσι άρχισαν οι προσπάθειες των μαθηματικών της εποχής εκείνης για την επίλυση του.

Το 1637 ο Γάλλος Rene Descartes παρουσίασε – γενικεύοντας τα συμπεράσματα του Ιταλού Μαθηματικού Gerolamo Gardano – τον “κανόνα των πρόσημων” (Descartes’ rule of signs), που μας δίνει ένα άνω όριο στο πλήθος των θετικών ριζών ενός πολυώνυμου. Στις αρχές του 19<sup>ου</sup> αιώνα εμφανιστήκαν τα δυο θεώρηματα των Γάλλων F.D. Budan (1807) και J.B.J. Fourier(1820). Τα θεώρηματα αυτά , αν και διαφορετικά είναι ισοδύναμα και μας δίνουν ένα άνω φράγμα στο πλήθος των πραγματικών ριζών μια πολυωνμικής εξίσωσης σε ένα δεδομένο ανοιχτό διάστημα.

Το 1829 , ο Γάλλος μαθηματικός C.Sturm βασιζόμενος στο θεώρημα του Fourier , παρουσίασε τις πραγματικές ρίζες ενός πολυώνυμου χρησιμοποιώντας την μέθοδο της διχοτόμησης. Ο Sturm ήταν ο πρώτος που έλυσε το πρόβλημα της απομόνωσης των πραγματικών ριζών και η μέθοδος του χρησιμοποιήθηκε μέχρι τα τέλη της δεκαετίας του 1980, όταν και αναπτύχθηκε από τον Αλκιβιάδη Γ. Ακρίτα η μέθοδος συνεχών κλασμάτων VAS.

Το 1836 παρουσιάστηκε το θεώρημα Vincent. το οποίο εξαρτάται από τον κανόνα των πρόσημων του Descartes , ενώ κάνεις μπορεί να εξακριβώσει και τη συνάφεια του με το θεώρημα του Budan μελετώντας και συγκρίνοντας τα δυο θεώρηματα. Το θεώρημα αυτό αποτελεί τη βάση δυο μεθόδων απομόνωσης των πραγματικών ριζών πολυωνυμίων

εξισώσεων με συνεχή κλάσματα . Η πρώτη από αυτές τις μεθόδους ανήκει στον ίδιο το Vincent (1836) και έχει εκθετικό χρόνο υπολογισμού. Η δεύτερη αναπτύχθηκε το 1978 από τον Αλκιβιάδη Γ. Ακρίτα , βελτιώθηκε με τον Strzebonski (VAS) το 1994 και έχει πολυωνυμικό χρόνο υπολογισμού. Χρησιμοποιείτε στο σύστημα υπολογιστικής άλγεβρας Mathematica.

Το 1975-1976 αναπτύχθηκε η μέθοδος Vincent-Collins-Akritas (VCA) από τους Collins και Ακρίτα η οποία στηρίζεται σε μια ολική τροποποίηση του θεωρήματος του Vincent. Η μέθοδος αυτή χρησιμοποιεί διχοτόμηση για την απομόνωση των πραγματικών ριζών , γιατί και είναι γνωστή ως VCA-bisection method. Χρησιμοποιείται στο σύστημα υπολογιστικής άλγεβρας Maple. Η ταχύτερη υλοποίηση αναπτύχθηκε από τους Rouillier και Zimmerman. Μια δεύτερη μέθοδος διχοτόμησης βασισμένη στο θεώρημα του Vincent αναπτύχθηκε το 2000 από τους Alesina-Galuzzi ( Vincent-Alesina-Galuzzi bisection method).

Περισσότερες πληροφορίες για τον κανόνα πρόσημων του Descartes, το θεώρημα του Vincent (1836) , απομόνωση ριζών με το παραπάνω θεώρημα όπως και περιγραφή της μεθόδου VAS-CF μπορεί να βρεθούν στην διπλωματική του Σπύρου Κεχαγιά όπως αναφέρθηκε και πιο πάνω.

# Qt Necessitas for Android

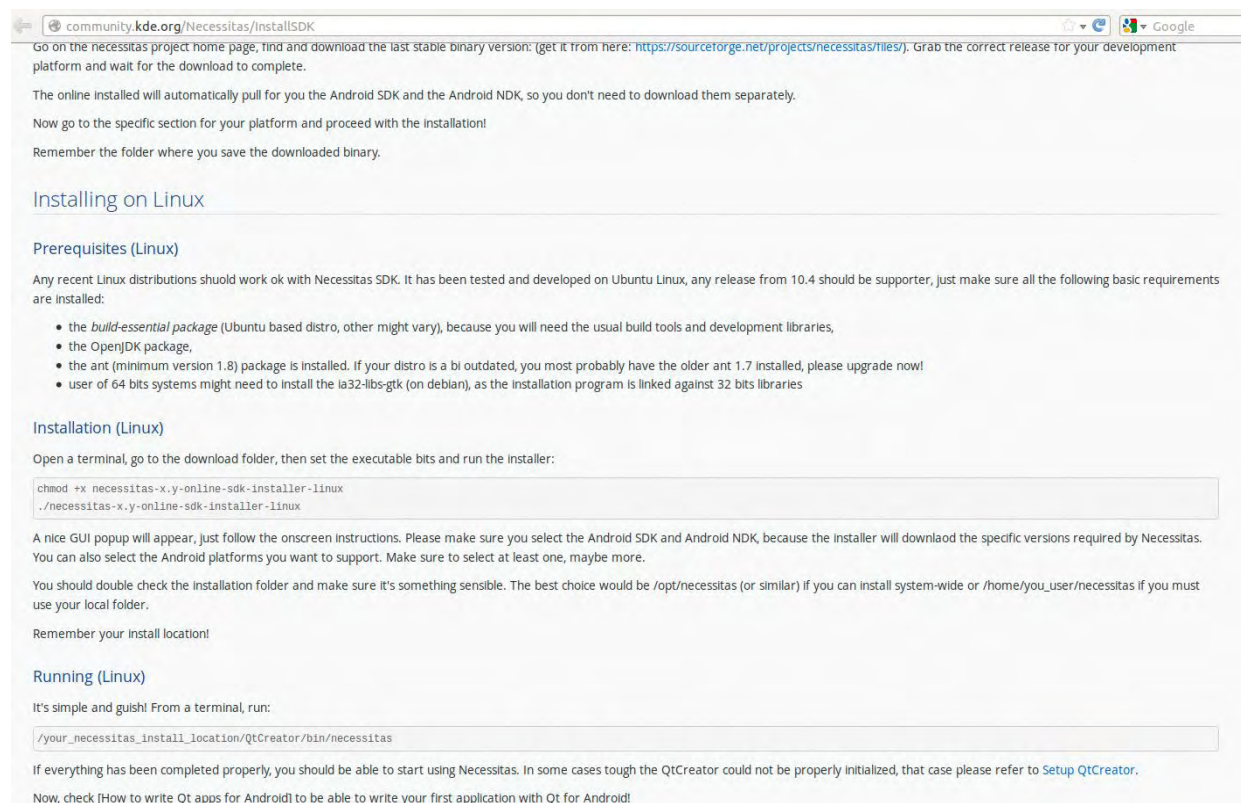
## Εγκατάσταση και επεξήγηση Qt

Στην επομένη παράγραφο θα περιγράψουμε βήμα βήμα τον τρόπο με τον οποίο γίνετε η εγκατάσταση του Qt Necessitas σε Linux συστήματα στο οποίο και χρησιμοποιήσαμε για την συγγραφή της εφαρμογής.

Αν και το Qt Necessitas προσφέρει και για 3 βασικά λειτουργικά συστήματα Windows, Linux, Mac έχουν αναφερθεί διάφορα προβλήματα συμβατότητας κυρίως στα windows, όποτε και γιαυτό προτιμήσαμε το προτεινόμενο λειτουργικό σύστημα, το Linux. Για την ακρίβεια Ubuntu 11.04

Επίσης λόγω διαφορών “bugs” του IDE καθώς πρόκειται ακόμα για alpha έκδοση θα ήταν προτεινόμενο η εγκατάσταση του και η χρήση του να γίνει σαν root[] χρήστης.

Εδώ <http://community.kde.org/Necessitas/InstallSDK> μπορείτε να βρείτε λεπτομερείς οδηγίες για την προτεινόμενη εγκατάσταση.



The screenshot shows a web browser window displaying the installation instructions for Qt Necessitas on Linux. The page title is "community.kde.org/Necessitas/InstallSDK". The content includes instructions on how to download the SDK, prerequisites for Linux (such as build-essential, OpenJDK, and ant), and the steps for installation and running the application. The installation instructions include a terminal command to make the installer executable and another to run it. The prerequisites section lists several requirements, including the use of 64-bit systems and the installation of specific packages. The running section provides a terminal command to start the application.

community.kde.org/Necessitas/InstallSDK

Go on the necessitas project home page, find and download the last stable binary version: (get it from here: <https://sourceforge.net/projects/necessitas/files/>). Grab the correct release for your development platform and wait for the download to complete.

The online installer will automatically pull for you the Android SDK and the Android NDK, so you don't need to download them separately.

Now go to the specific section for your platform and proceed with the installation!

Remember the folder where you save the downloaded binary.

### Installing on Linux

#### Prerequisites (Linux)

Any recent Linux distributions should work ok with Necessitas SDK. It has been tested and developed on Ubuntu Linux, any release from 10.4 should be supported, just make sure all the following basic requirements are installed:

- the *build-essential* package (Ubuntu based distro, other might vary), because you will need the usual build tools and development libraries,
- the OpenJDK package,
- the ant (minimum version 1.8) package is installed. If your distro is a bit outdated, you most probably have the older ant 1.7 installed, please upgrade now!
- user of 64 bits systems might need to install the ia32-libs-gtk (on debian), as the installation program is linked against 32 bits libraries

#### Installation (Linux)

Open a terminal, go to the download folder, then set the executable bits and run the installer:

```
chmod +x necessitas-x.y-online-sdk-installer-linux
./necessitas-x.y-online-sdk-installer-linux
```

A nice GUI popup will appear, just follow the onscreen instructions. Please make sure you select the Android SDK and Android NDK, because the installer will download the specific versions required by Necessitas. You can also select the Android platforms you want to support. Make sure to select at least one, maybe more.

You should double check the installation folder and make sure it's something sensible. The best choice would be /opt/necessitas (or similar) if you can install system-wide or /home/you\_user/necessitas if you must use your local folder.

Remember your install location!

#### Running (Linux)

It's simple and guish! From a terminal, run:

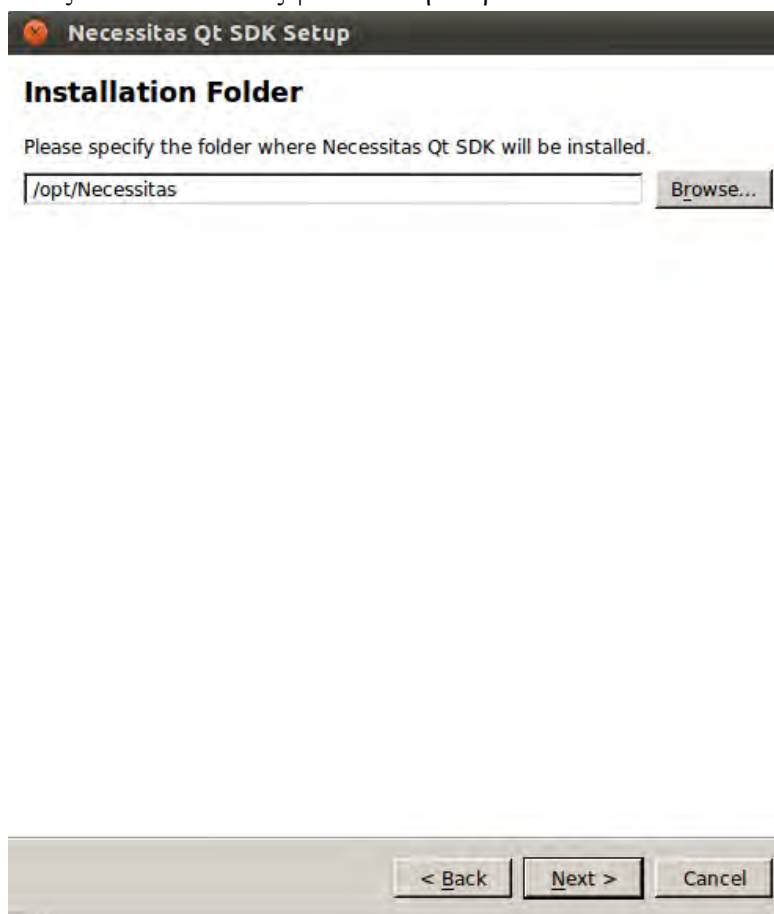
```
/your_necessitas_install_location/QtCreator/bin/necessitas
```

If everything has been completed properly, you should be able to start using Necessitas. In some cases though the QtCreator could not be properly initialized, that case please refer to [Setup QtCreator](#).

Now, check [How to write Qt apps for Android](#) to be able to write your first application with Qt for Android!

Εικόνα 12. οδηγίες εγκατάστασης σε περιβάλλον Linux

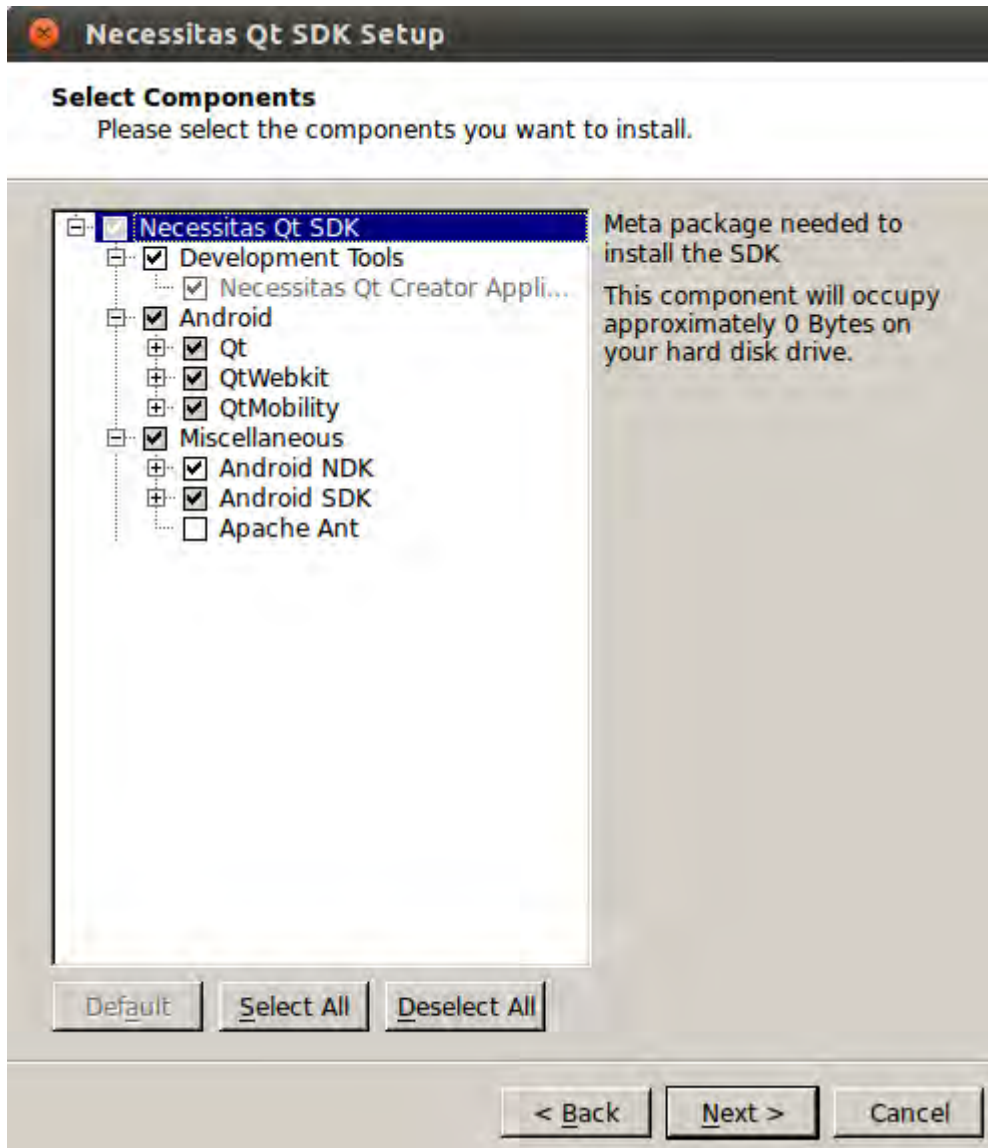
- Αρχικά μπορούμε να κατεβάσουμε το installer από το site <http://necessitas.kde.org/> είτε από το [sourceforge.net/p/necessitas/home/](http://sourceforge.net/p/necessitas/home/) καθώς όπως έχουμε αναφέρει και παραπάνω πρόκειται για ένα open-source IDE.
- Έπειτα σαν επόμενο βήμα είναι να ανοίξουμε ένα terminal και να πληκτρολογήσουμε `“ sudo su cd “path” ”` όπου path ο φάκελος στον οποίο κατέβηκε ο installer
- Επομένη εντολή εφόσον βρισκόμαστε στον σωστό φάκελο είναι `“ chmod +x necessitas-0.3-online-sdk-installer-linux ”` αυτό γίνεται για να αλλάξουμε τα δικαιώματα του installer που μόλις κατεβάσαμε.
- Μετά `“ ./necessitas-0.3-online-sdk-installer-linux ”` και έτσι στο επόμενο βήμα θα ανοίξει ο installer όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 13. Necessitas Qt Installer

- Στην παρακάτω εικόνα αφήνουμε τις προεπιλεγμένες επιλογές, μπορούμε να παρατηρήσουμε ότι το Necessitas θα κατεβάσει και θα εγκαταστήσει αυτόματα το Android SDK και το Android NDK το οποίο είναι απαραίτητο για να μπορέσουμε να γράψουμε σε C++ το πρόγραμμά μας.

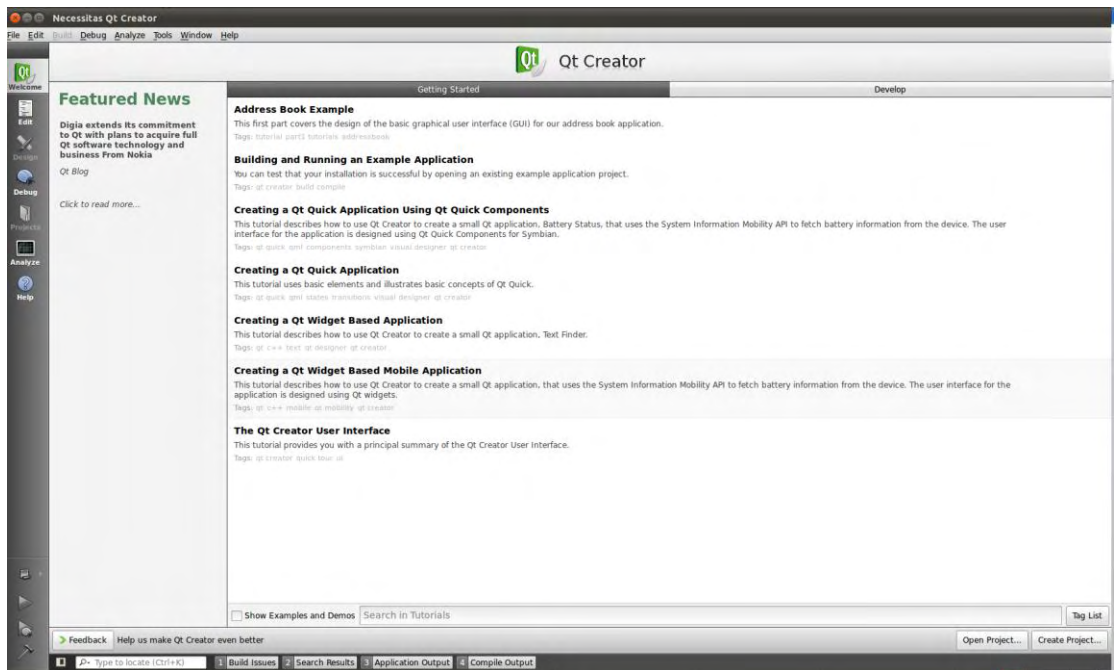




Εικόνα 14. Qt Necessitas installer options

- Στο επόμενο βήμα , μόλις τελειώσει η εγκατάσταση δεν τρέχουμε αμέσως το πρόγραμμα όπως έχει προεπιλογή. Απλά κλείνουμε τον installer και έπειτα ανατρέχουμε στο terminal πάλι.  
σημείωση: το terminal να είναι με δικαιώματα superuser.
- Έπειτα πληκτρολογούμε `"cd /opt/necessitas/QtCreator/bin/"` όπου opt/necessitas ήταν το path που επιλέξαμε για την εγκατάσταση.
- Τελική εντολή για να ανοίξει το IDE `"/necessitas"`

Έπειτα από αυτά τα βήματα θα ανοίξει το IDE όπως φαίνεται στην παρακάτω εικόνα.



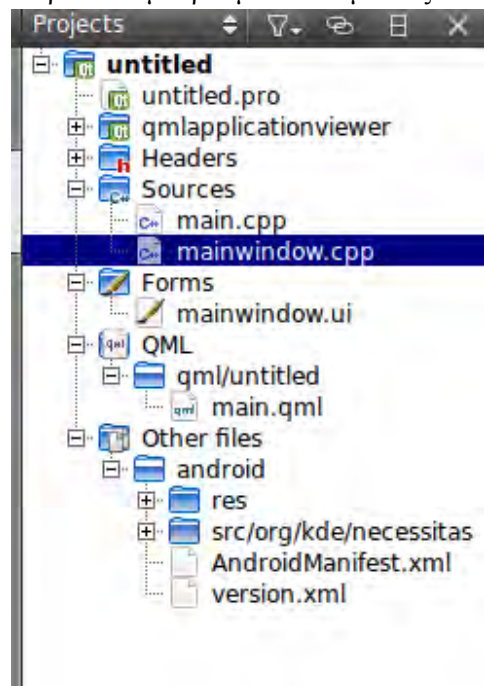
Εικόνα 15. Necessitas Qt Creator

# Εφαρμογή Real Roots

## Πρώτη επαφή με τις βιβλιοθήκες και το project του Qt.

Όπως τα περισσότερα IDE και το Qt δουλεύει με project δηλαδή υπάρχει ένας φάκελος με όλα τα απαραίτητα αρχεία για να ξεκινήσει κάποιος την συγγραφή κώδικα.

Παρακάτω μπορούμε να δούμε πως είναι ένα τέτοιο directory.



Εικόνα 16. directory Android Qt project

Όπως βλέπουμε παραπάνω το κάθε directory αποτελείται από διάφορα αρχεία όπως το \*.pro το οποίο είναι το αρχείο που επιλεγεί ο χρήστης ώστε αυτό με την σειρά του να ανοίξει το προτζεκτ ολόκληρο. (File -> Open File or Project )

Το main.cpp είναι το βασικό αρχείο όπου από εκεί θα αναφέρουμε ποιο θα είναι το πρώτο layout που θα τρέξει στην εφαρμογή μας.

Layout είναι η κάθε οθόνη που βλέπουμε στην εφαρμογή μας η οποία συνοδεύεται από το λογικό μοντέλο κώδικα που υλοποιεί την λειτουργικότητα της εφαρμογής.

Έπειτα βλέπουμε ότι κάτω από ένα φάκελο android βρίσκετε τα resources της εφαρμογής και το Android Manifest που κατά κύριο λόγο ότι βρίσκεται κάτω από αυτό τον φάκελο δεν θα χρειαστεί να το τροποποιήσουμε, καθώς αυτό είναι και ο σκοπός του Necessitas Qt να μπορούμε να γράψουμε native C++ σε λειτουργικό που γράφετε σε JAVA (Android).

Επίσης παρακάτω βλέπουμε και τρεις βασικούς υποφακελους του προτζεκτ. "Headers", "Sources", "Forms".

Όσον αναφορά τα layout που πρέπει να δημιουργηθούν για την εφαρμογή μας υπάρχουν 2 τρόποι.

- a) είτε να δημιουργηθούν μέσα στο αρχείο όπου γράφετε η λογική του προγράμματος, δηλαδή να γραφτούν χειρόγραφα με κώδικα.

Παρακάτω μπορείτε να δείτε ένα παράδειγμα τέτοιου κώδικα

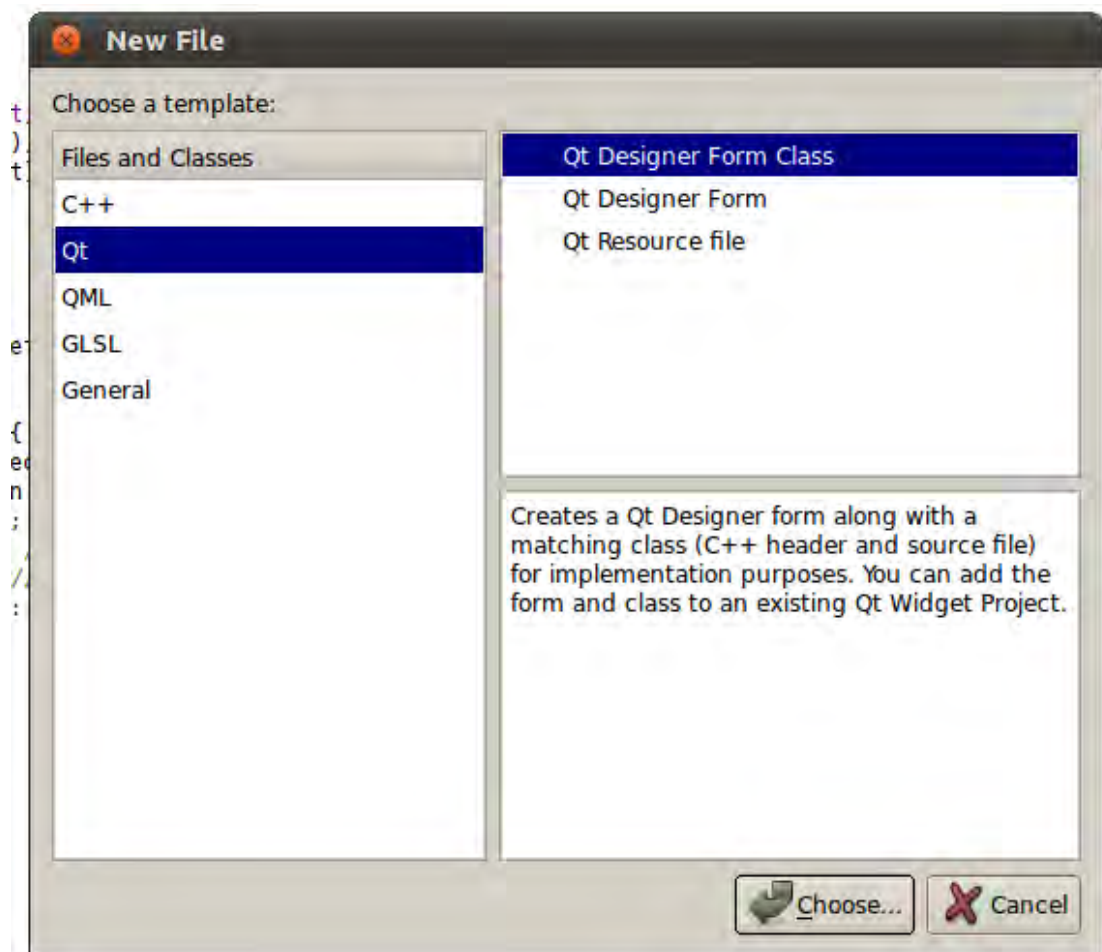
```
m_Input = new QLineEdit;
m_Output = new QLineEdit;

m_Layout = new QVBoxLayout;
m_Layout->addWidget(input);
m_Layout->addWidget(output);

Fen = new QWidget;
Fen->setLayout(m_Layout);
setCentralWidget(Fen);
```

Εικόνα 17. ενδεικτικός κώδικας για εισαγωγή νέου Layout

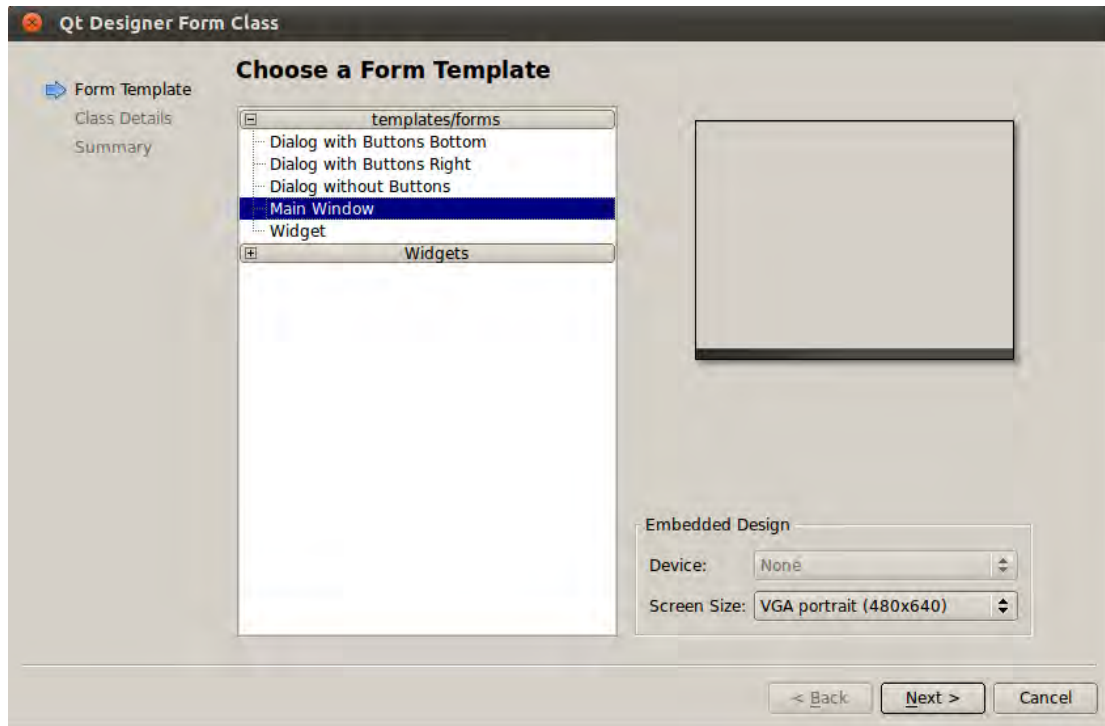
- b) είτε σαν δεύτερος και πιο εύκολος τρόπος για τον άπειρο χρήστη είναι να ορίσουμε ένα Qt Designer Form class όπως βλέπετε παρακάτω.



Εικόνα 18. εισαγωγή Qt Designer Form

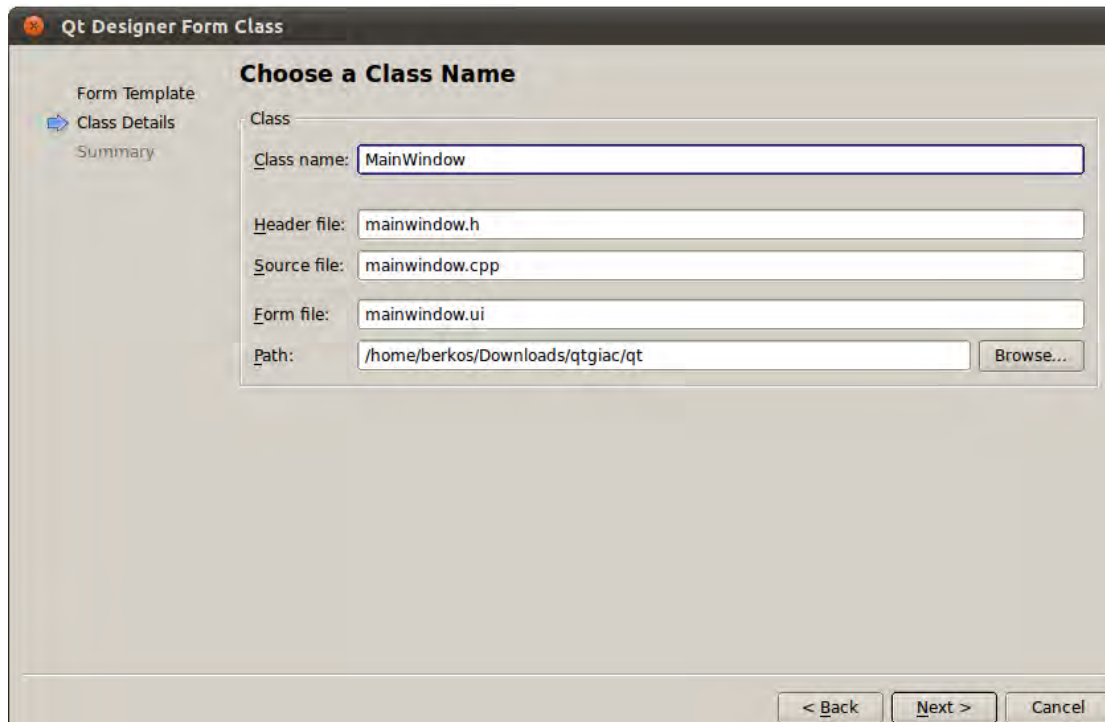
Όπως λει και η περιγράφει δημιουργεί 3 αρχεία , ένα my\_name.cpp ένα my\_name.h και ένα my\_name.ui , τα οποία συνδέονται μεταξύ τους και κάνουν αρκετά ευκολότερη την διαδικασία συγγραφής κώδικα.

Σαν επόμενο βήμα από την πάνω οθόνη είναι να επιλέξουμε τι ακριβώς γραφικό περιβάλλον θέλουμε για αυτήν την κλάση καθώς και το μέγεθος οθόνης το οποίο προορίζετε.



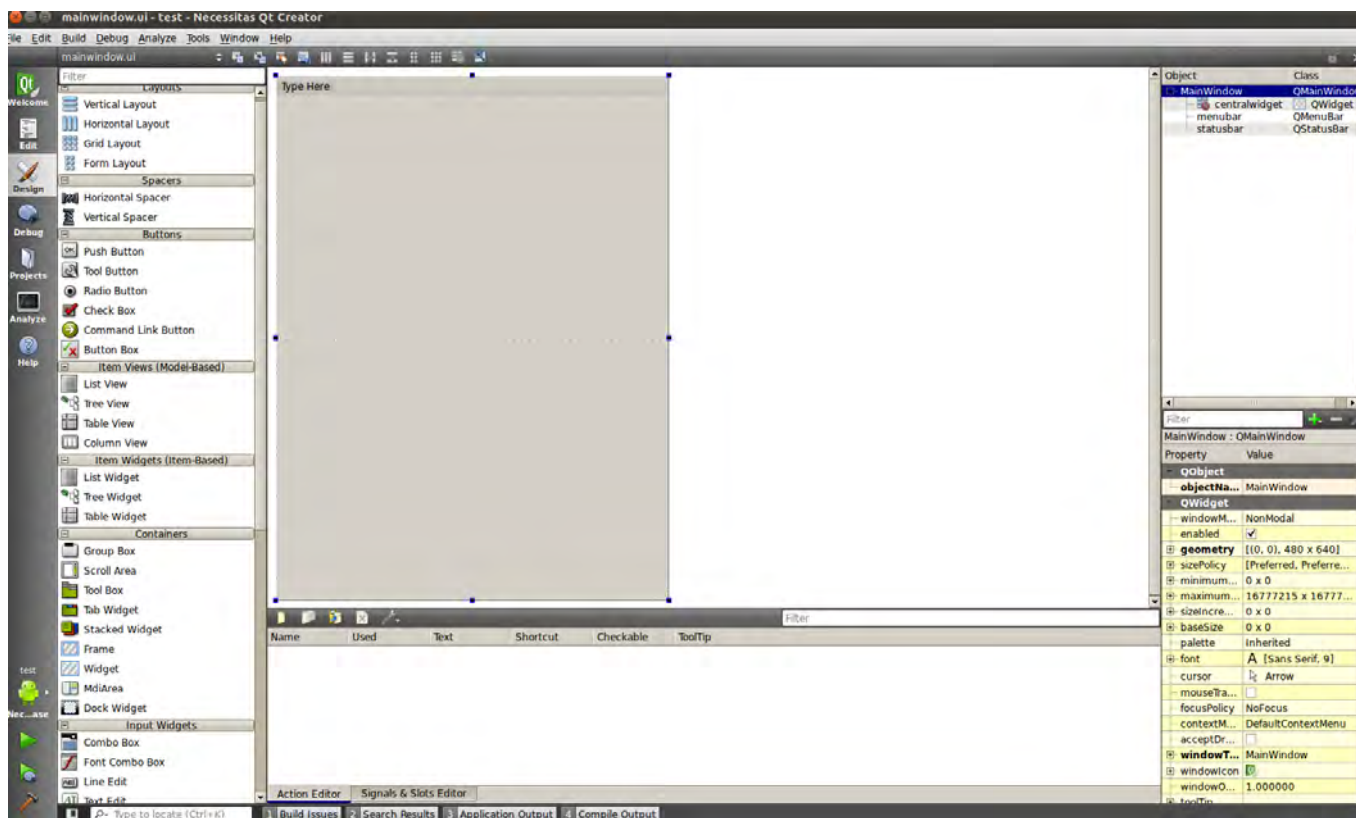
Εικόνα 19. Qt Designer Form Class

Παρακάτω φαίνεται πως συσχετίζει τα 3 αρχεία με το ίδιο όνομα που εμείς του δίνουμε και σε ποιο μονοπάτι θα τα αποθηκεύσει.



Εικόνα 20. Class Name

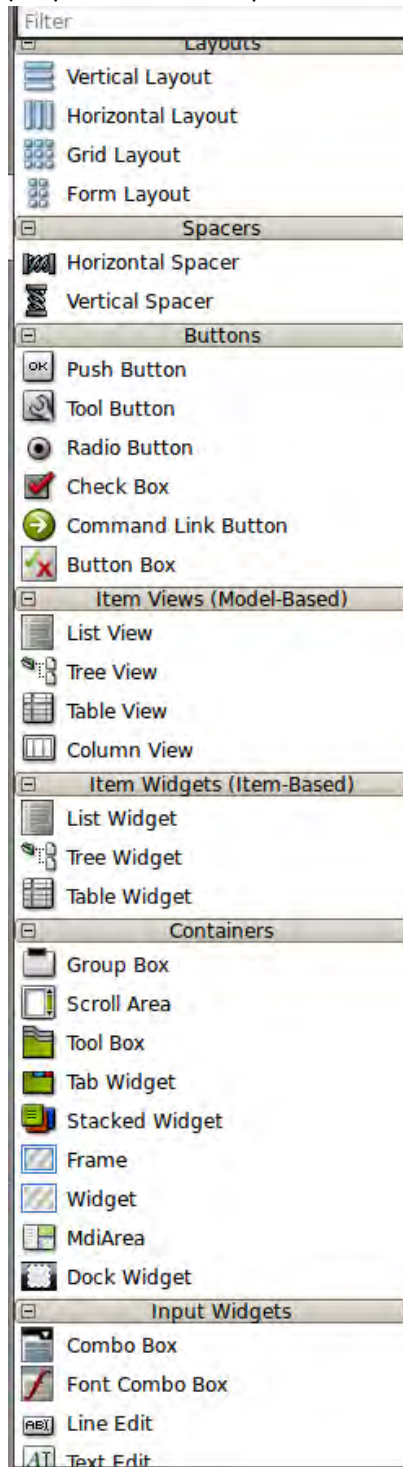
Μόλις γίνει η αποθήκευση η επομένη εικόνα είναι



Εικόνα 21. my\_name.ui

Όπου σε αυτήν μπορούμε να δούμε ότι μας άνοιξε το my\_name.ui και ένα γραφικό περιβάλλον στο οποίο μπορούμε να εισάγουμε από τα αριστερά διάφορα widwet όπως

μπορείτε να δείτε παρακάτω.

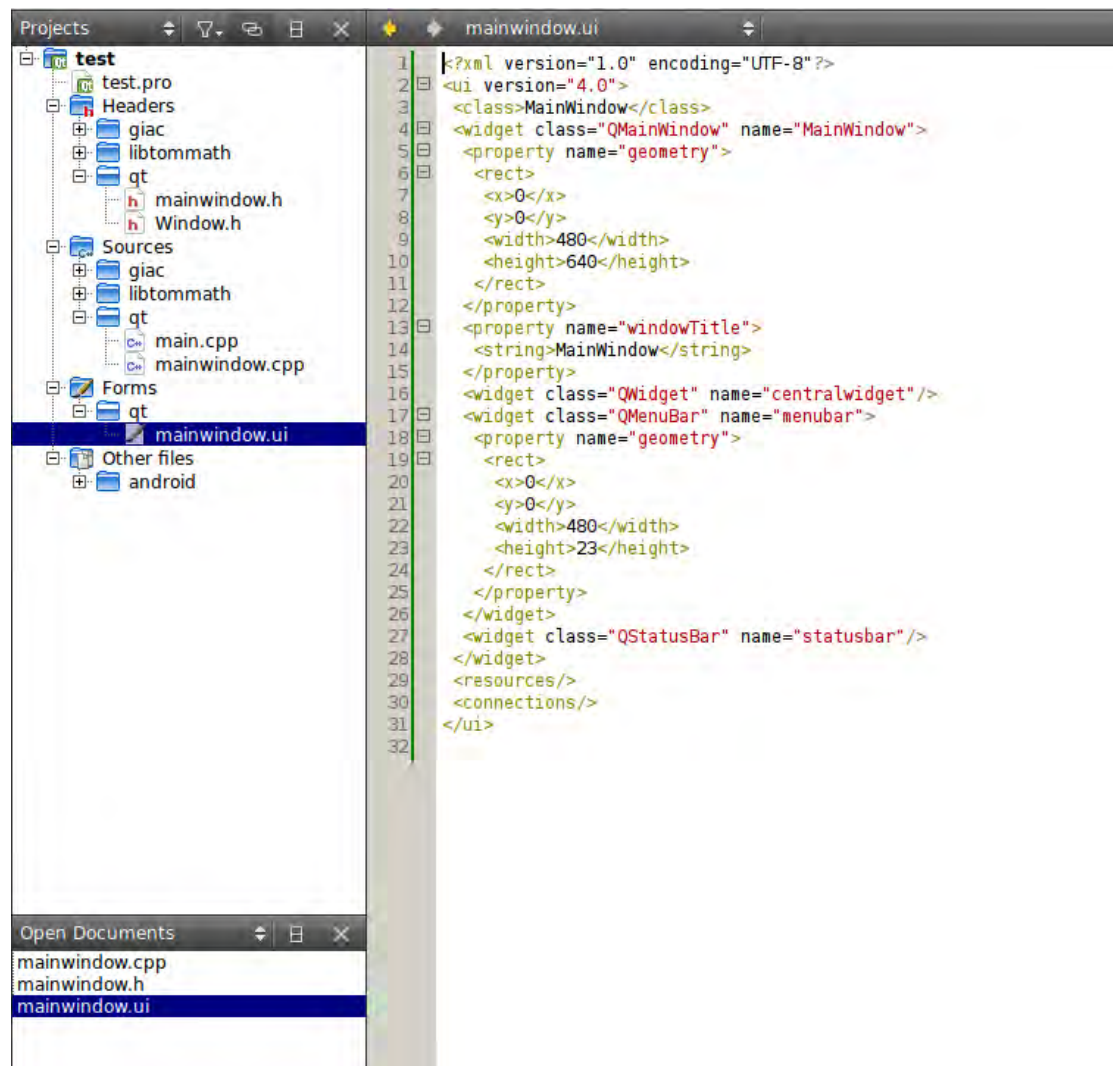


Εικόνα 22. διαφορές διαθέσιμες επιλογές για το my\_name.ui

Πατώντας πάνω αριστερά το εικονίδιο edit.

Μπορούμε να δούμε ότι το αρχείο .ui δεν είναι κάτι άλλο παρά ένα αρχείο περιγραφής σε μορφή xml.

Επίσης μπορούμε να παρατηρήσουμε ότι και τα ανάλογα αρχεία ίδιου ονόματος που έχουν δημιουργηθεί κάτω από τους φάκελους Headers , Sources , Forms.



Εικόνα 23. ui as xml

Τώρα πλέον μπορούμε να επιλέξουμε ένα κουμπί από το αριστερό μενού και εφόσον το τοποθετήσουμε στο Layout όπως φαίνεται παρακάτω





Εικόνα 24. Push Button Layout

Μπορούμε να έχουμε πρόσβαση σε αυτό και τις διαφορές μεθόδους του από το \*.cpp με τον απλό τρόπο όπως

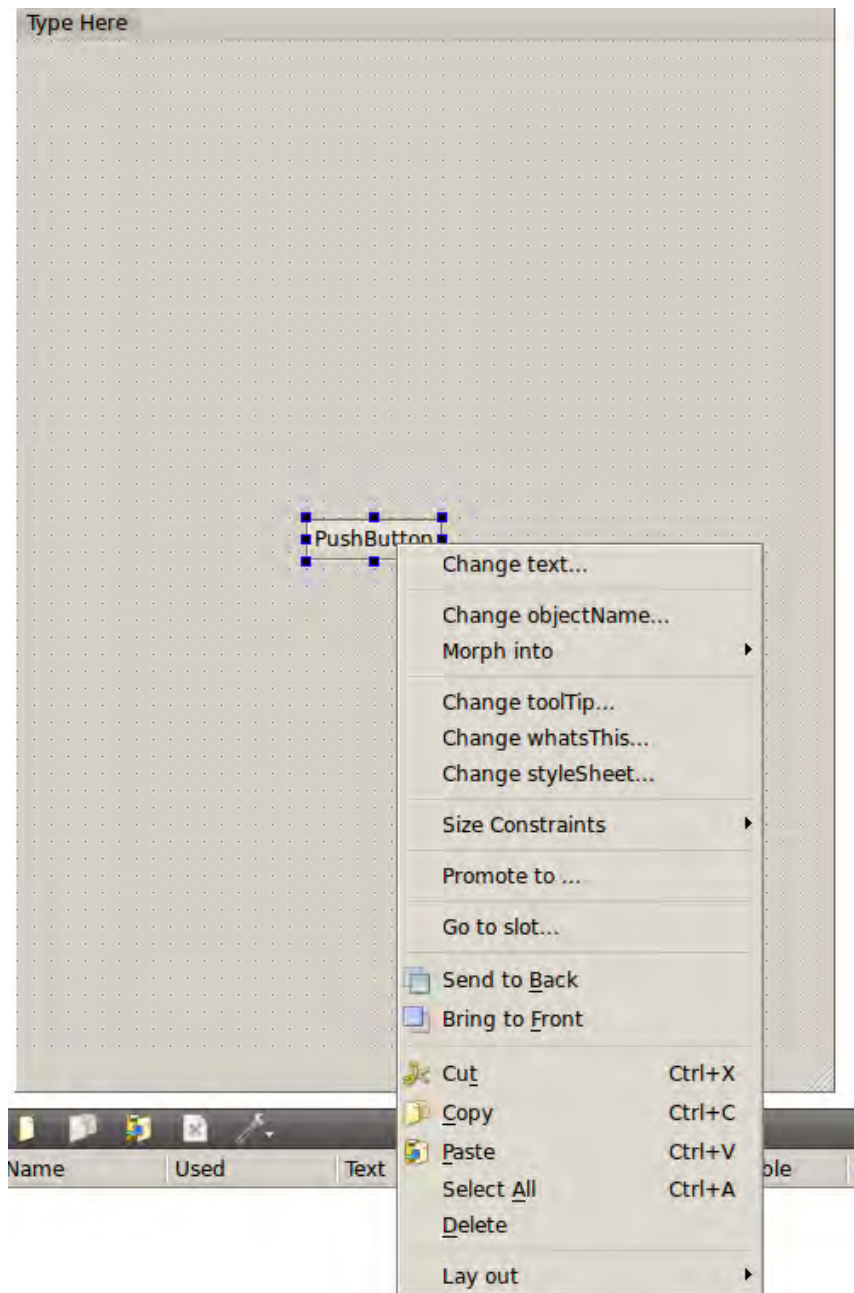
```
mainwindow.cpp* MainWindow::MainWindow(QWidget *)
1 #include "mainwindow.h"
2 #include "ui_mainwindow.h"
3
4 MainWindow::MainWindow(QWidget *parent) :
5     QMainWindow(parent),
6     ui(new Ui::MainWindow)
7 {
8     ui->setupUi(this);
9
10    ui->pushButton->
11 }
12
13 MainWindow::~MainWindow()
14 {
15     delete ui;
16 }
17
```

Εικόνα 25. push Button

Όπως φαίνεται για να γίνει αυτό απλά να υπάρχει το #include “ui\_mainwindow.h” στην γραμμή 2 του πηγαίου κώδικα, το οποίο με τον τρόπο που αναφέραμε γίνεται αυτόματα χωρίς να χρειαστεί να κάνουμε εμείς κάτι παραπάνω.

Κατά κύριο λόγο τα κουμπιά και τα περισσότερα από τα υπόλοιπα widgets δουλεύουν σαν Slots τα οποία είτε εκτελούν μια συνάρτηση ( διαδικασία ) του προγράμματος είτε μας πάνε σε κάποιο άλλο Layout.

Για να εισάγει ο χρήστης κάποια σύνδεση με κάποια συνάρτηση, η οποία είτε μπορεί να εκτελεί κάποια εργασία είτε να μας στέλνει σε άλλο Layout, μπορεί να το κάνει χειροκίνητα με κώδικα είτε αλλιώς με τον παρακάτω τρόπο.



Εικόνα 26. Connecting a button using ( Go to Slot)

Με αυτόν τον τρόπο θα κάνει όλες τις απαιτητές συνδέσεις και θα μας στείλει αυτόματα στο `my_name.cpp` το οποίο θα έχει δημιουργήσει μια νέα συνάρτηση που θα εκτελείτε με το πάτημα του κουμπιού.

```
mainwindow2.cpp* MainWindow2::on_pushButton_clicked()
1  #include "mainwindow2.h"
2  #include "ui_mainwindow2.h"
3
4  MainWindow2::MainWindow2(QWidget *parent) :
5      QMainWindow(parent),
6      ui(new Ui::MainWindow2)
7  {
8      ui->setupUi(this);
9  }
10
11 MainWindow2::~MainWindow2()
12 {
13     delete ui;
14 }
15
16 void MainWindow2::on_pushButton_clicked()
17 {
18     |
19 }
20
```

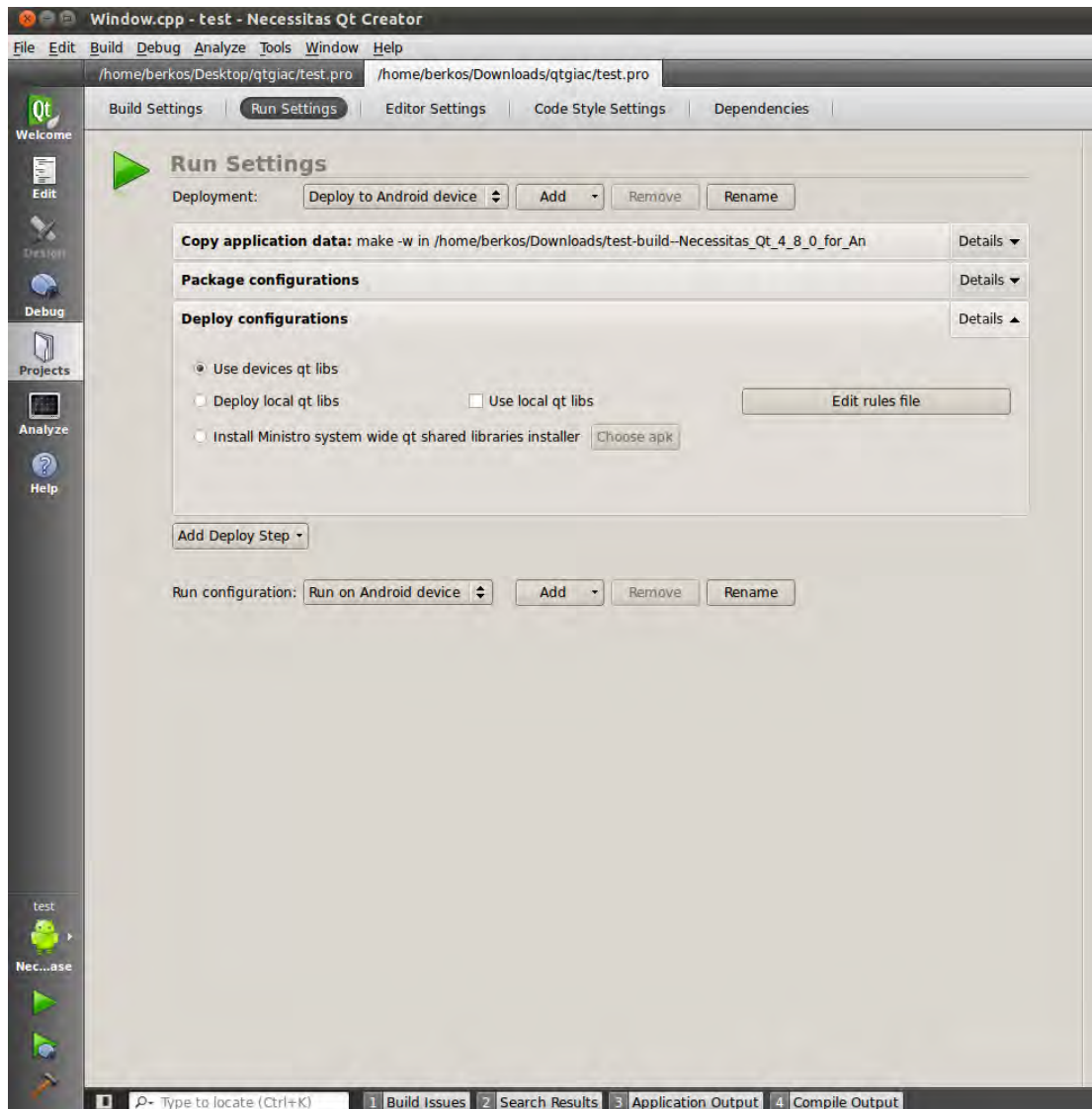
Εικόνα 27.On\_pushButton\_clicked()

## Μεταγλώττιση και εκτέλεση του προγράμματος μας

Κοιτώντας την “εικόνα 6” μπορούμε να δούμε ότι η μεταγλώττιση μπορεί να γίνει απλά πατώντας το πράσινο κουμπί “play”.

Όμως την πρώτη φορά , πριν γίνει αυτό θα πρέπει να γίνουν μερικές ρυθμίσεις που αφορά το εκάστου πρόγραμμα που δημιουργείτε.

Αρχικά πρέπει να πατήσουμε το κουμπί projects στα αριστερά και στα run settings αρκεί να πατήσουμε το deploy configurations.



**Εικόνα 28. deploy configurations**

Για να γίνει εγκατάσταση είτε στον emulator είτε σε κάποια φυσική συσκευή μπορούμε να επιλέξουμε την τρίτη επιλογή “Install Ministro system wide qt shared libraries installer” στο “choose apk” θα πρέπει να κατεβάσουμε από το αρχικό site του Necessitas που έχουμε αναφέρει πιο πάνω το MinistroII.apk το οποίο και θα χρησιμοποιηθεί από τον emulator του android για να γίνει εγκατάσταση των απαραίτητων βιβλιοθηκών.

Όσον αφορά την φυσική συσκευή που μπορεί να υπάρχει τότε μόλις ανοίξει η εφαρμογή μας αν δεν υπάρχει εγκατεστημένο το Ministro θα μας στείλει η ίδια η εφαρμογή στο Google play για εγκατάσταση του Ministro.

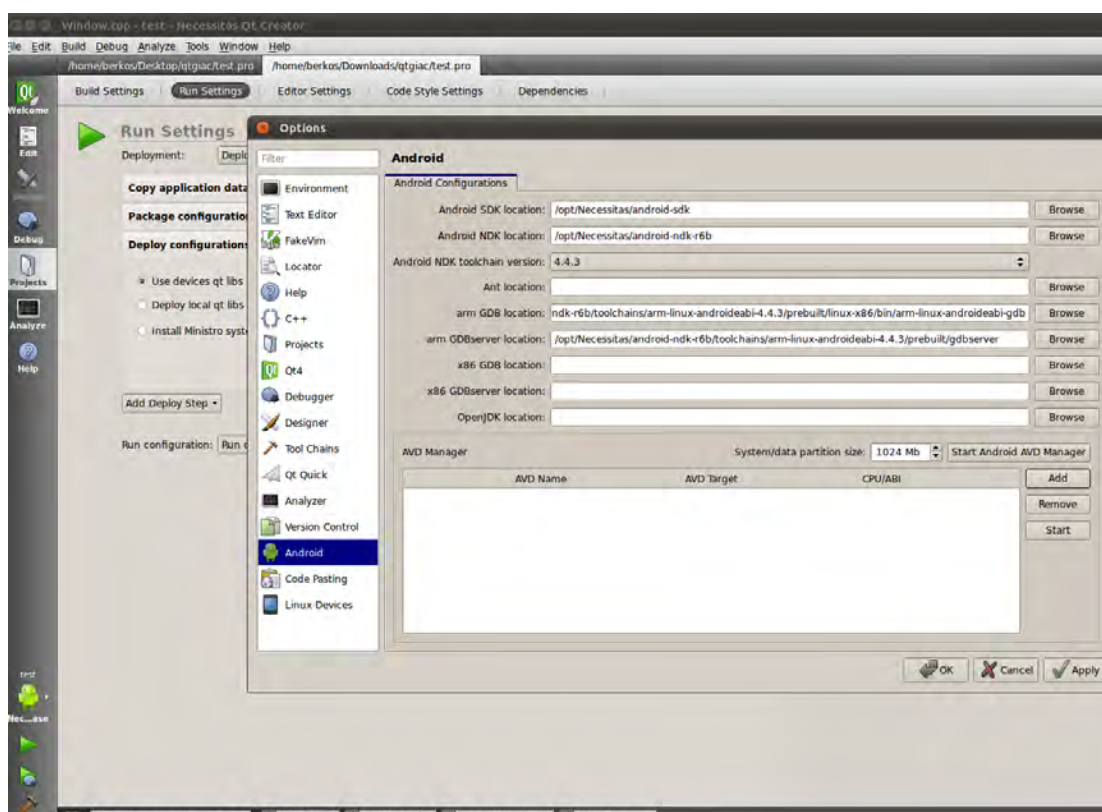
## Android Emulator & Physical Device

Όπως αναφέραμε η εφαρμογή μας μπορεί να τρέξει και να δοκιμαστεί και στις 2 παραπάνω συσκευές.

- Όσον αφορά την φυσική συσκευή το μόνο που έχει ο χρήστης να κάνει είναι να την συνδέσει στο σύστημα και απλά να ενεργοποιήσει από την συσκευή τον Εντοπισμό Σφαλμάτων η αλλιώς από τις ρυθμίσεις να ενεργοποιήσει το Debugging Mode.
- Αν τώρα επιθυμούμε τον Android Emulator τότε θα πρέπει να δημιουργήσουμε έναν από το ανάλογο μενού.  
Το πακέτο με τον διαχειριστή των Android Emulators άρχετε μαζί με το Android SDK και έτσι πάντα θα είναι διαθέσιμο.

Tools->Options->Android

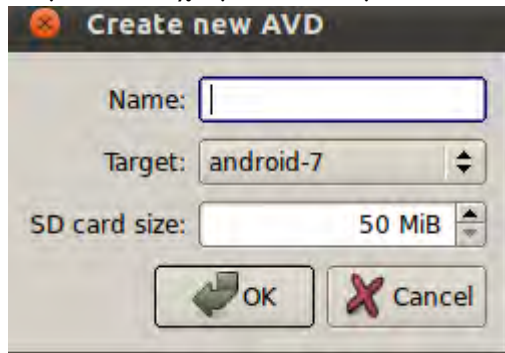
Και έτσι θα έχουμε την παρακάτω εικόνα.



Εικόνα 29. Android Emulator

Όπου σε αυτό το σημείο μπορούμε να δούμε που βρίσκετε το Android SDK και το Android NDK .

Το μόνο που έχουμε να κάνουμε είναι να πατήσουμε στο add.



Εικόνα 30. Creating an emulator

Εδώ απλά έχουμε να ορίσουμε το όνομα του και το κυριότερο τι API να υποστηρίζει. Δηλαδή ποια να είναι η έκδοση του Android emulator που θα δημιουργηθεί.



Εικόνα 31. Android Emulator

## Ανάπτυξη της Εφαρμογής RealRoots

Η εφαρμογή RealRoots χρησιμοποιεί δυο βασικές βιβλιοθήκες.

- Την tommath library η οποία χρησιμοποιείτε από την Giac ώστε να μπορούμε να κάνουμε πράξεις με αρκετά μεγάλους συντελεστές πολυώνυμων.
- Την βιβλιοθήκη Giac η όποια είναι μια μαθηματική βιβλιοθήκη γραμμένη σε C++ που υπόκειται στην φιλοσοφία του ανοιχτού κώδικα. Αποτελεί τον πυρήνα του υπολογιστικού συστήματος Xcas ,το όποιο είναι ένα μαθηματικό εργαλείο γενικού σκοπού. Δημιουργός του Cass και της Giac είναι ο Bernard Parisse. Περισσότερες πληροφορίες για τον ίδιο και για την βιβλιοθήκη μπορείτε να βρείτε ανατρέχοντας στην σελίδα του xcas.

Η εφαρμογή μας αναπτύχθηκε ξεκινώντας από ένα βασικό πακέτο που περιλάμβανε τις βιβλιοθήκες tommath και Giac μεταγλωττισμένες για Android συσκευές με την χρήση του Qt Necessitas.

Χάρη στην βοήθεια του ιδίου του Bernard Parisse είχαμε αυτό το πακέτο για να ξεκινήσει η ανάπτυξη της εφαρμογής.

## Λειτουργίες της εφαρμογής RealRoots

Οι λειτουργίες της εφαρμογής χωρίζονται σε 2 κατηγορίες.

- Μαθηματικές λειτουργίες πάνω σε πολυώνυμα
- Προηγμένη Λειτουργία (Advanced Mode)

### Μαθηματικές λειτουργίες πάνω σε πολυώνυμα

1. Απομόνωση Πραγματικών ριζών πολυώνυμων με θετικούς συντελεστές κάνοντας χρήση της μεθόδου συνέχων κλασμάτων Vincent-Akritas-Strzbonski (VAS) . Επιστρέφεται μια λίστα των διαστημάτων απομόνωσης μαζί με τις πολλαπλότητες των αντιστοιχών ριζών. Αν βρεθεί ακριβής ρίζα  $m$ , τότε το διάστημα απομόνωσης που επιστρεφτό θα είναι της μορφής  $[m,m]$ .
2. Απομόνωση πραγματικών ριζών πολυώνυμων με θετικούς συντελεστές κάνοντας την χρήση της μεθόδου Sturm. Επιστρέφεται μια λίστα διαστημάτων απομόνωσης μαζί με τις πολλαπλότητες των αντιστοιχών ριζών. Και σε αυτήν την περίπτωση αν βρεθεί η ακριβής ρίζα  $m$ , τότε το διάστημα απομόνωσης που επιστρέφετε θα είναι της μορφής  $[m,m]$ .
3. Προσέγγιση πραγματικών ριζών πολυώνυμων ...
4. Εύρεση κάτω ορίων στις θετικές ρίζες πολυώνυμων.
5. Εύρεση άνω ορίων στις θετικές ρίζες πολυώνυμων.
6. Square-free factorization (διάσπαση πολυωνύμου σε παράγοντες ελεύθερους από τετράγωνα)

7. Εύρεση των ρητών ριζών πολυώνυμων.
8. Απλή παραγοντοποίηση πολυώνυμων.

### Προηγμένη Λειτουργία(Advanced Mode)

Αυτή η λειτουργία αναφέρετε σε ήδη χρηστές του xcas. Σε αυτήν την επιλογή η εφαρμογή RealRoots μπορεί να δεχτεί σαν input οποιαδήποτε εντολή του Xcas(εκτός τις εντολές για σχεδιασμό γραφικών παραστάσεων) και να παράγει το ανάλογο output.

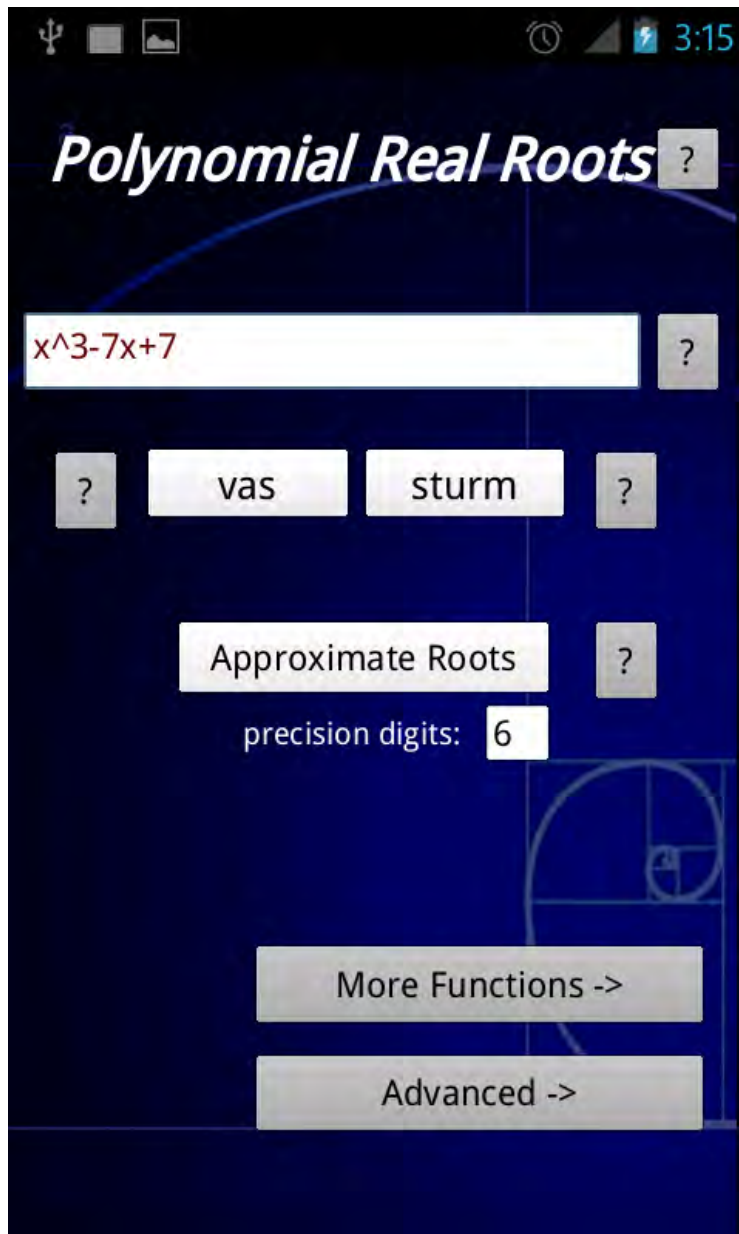
Όλες οι λειτουργίες της RealRoots συνοδεύονται από επεξηγηματικά σχόλια για την λειτουργία του κάθε κουμπιού και που μπορείτε να ανατρέξετε άμεσα, μέσω του πλοηγού του κινητού για περισσότερες πληροφορίες.

### **Διεπαφή της εφαρμογής RealRoots**

Μόλις γίνει η εγκατάσταση όλων των βιβλιοθηκών η εφαρμογή μας είναι έτοιμη προς εκτέλεση.

Σαν πρώτη οθόνη της εφαρμογής βλέπουμε την παρακάτω εικόνα.

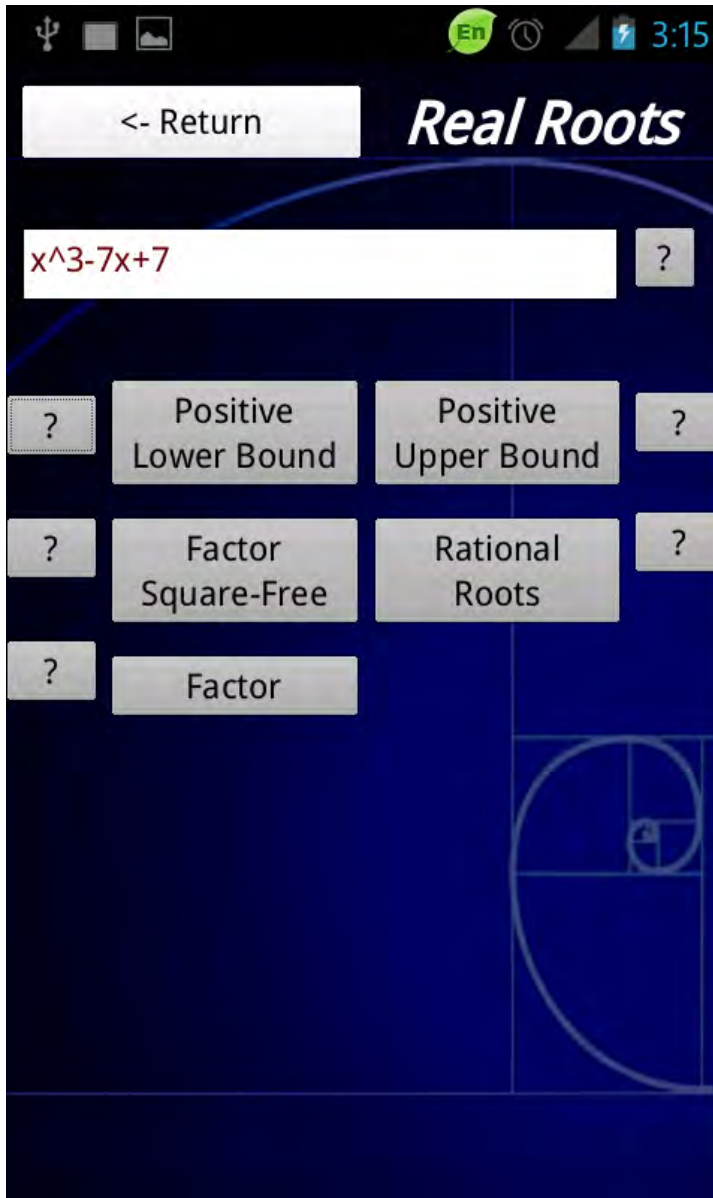




Εικόνα 32. Αρχική οθόνη εφαρμογής

- Στην παραπάνω οθόνη μπορούμε να δούμε τον τίτλο της εφαρμογής καθώς και ένα ερωτηματικό το οποίο προσφέρει κάποιες πληροφορίες για την συγγραφή της εφαρμογής και τι περιλαμβάνει.
- Έπειτα υπάρχει μια γραμμή εισαγωγής πολυώνυμου το οποίο συνοδεύεται από ένα ερωτηματικό το οποίο περιγράφει τον τρόπο με τον οποίο μπορεί ο χρήστης να εισάγει το πολυώνυμο που επιθυμεί.
- στην συνέχεια έχουμε τα 2 κουμπιά για τις δυο βασικές μεθόδους VAS και Sturm όπου αριστερά και δεξιά από την κάθε μια επιλογή αντίστοιχα υπάρχουν κουμπιά (ερωτηματικά) που επεξηγούν το σκοπό τους.

- Το επόμενο κουμπί Approximate Roots συνοδεύετε με την επιλογή precision digits όπου ο χρήστης μπορεί να επιλέξει την προσέγγιση των ριζών με όσα ψηφία ακριβείας θέλει από 2 έως 12.
- Τέλος μένουν τα 2 τελευταία κουμπιά της πρώτης οθόνης. Το “More Functions” μας στέλνει στην εικόνα 33 ενώ το Advanced μας στέλνει στην εικόνα 34. Παρακατω θα αναλυθούν περισσότερο.



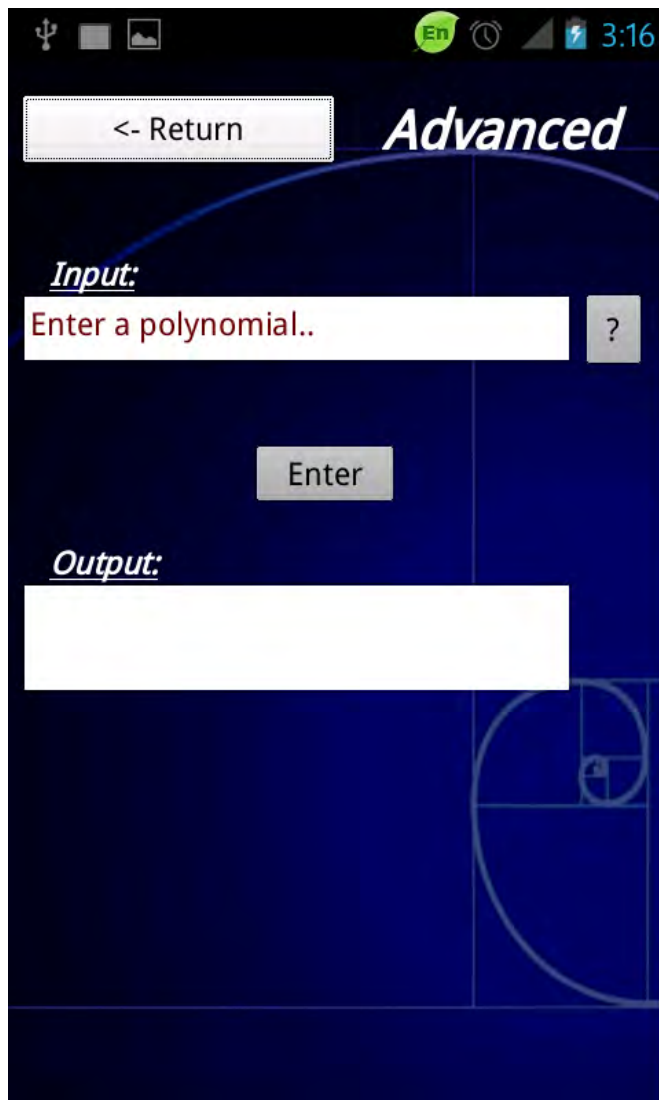
**Εικόνα 33. More Function Layout**

Το παραπάνω Layout εμφανίζετε αν ο χρήστης πατήσει το κουμπί More Functions που υπάρχει στην αρχική εικόνα.

- Στο πάνω μέρος της οθόνης υπάρχει το κουμπί για την επιστροφή στην αρχική οθόνη.
- Όπως και στην πρώτη οθόνη και εδώ υπάρχει η είσοδος του πολυώνυμου μαζί με την ενημερωτική πληροφορία για τον τύπο εισαγωγής ενός πολυώνυμου.

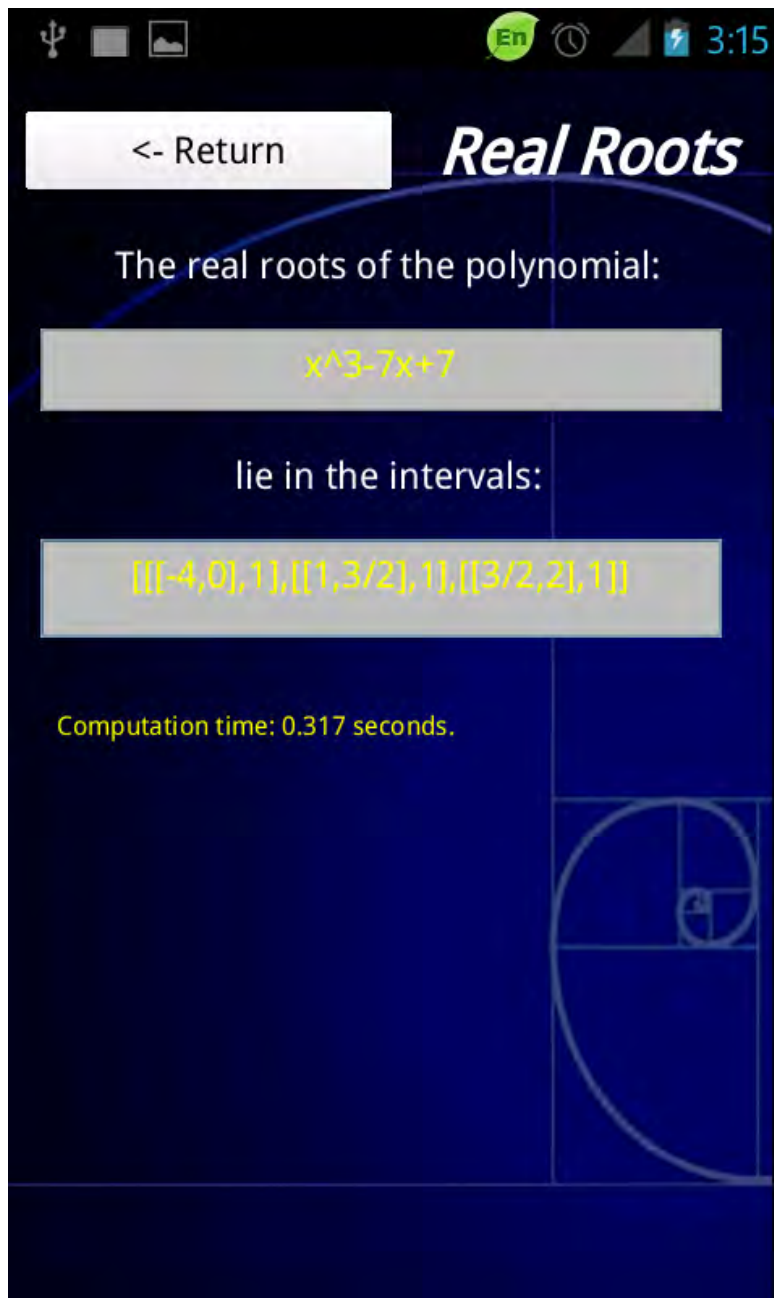
- Στην συνέχεια υπάρχουν 5 κουμπιά μαζί με την επεξήγηση τους , τα όποια είναι οι πράξεις πολυώνυμων που είναι διαθέσιμες στο χρήστη .
  - Positive Lower Bound:  
Εύρεση κάτω ορίων στις θετικές ρίζες πολυώνυμων.
  - Positive Upper Bound:  
Εύρεση άνω ορίων στις θετικές ρίζες πολυώνυμων.
  - Factor Square-Free:  
διάσπαση πολυωνύμου σε παράγοντες ελευθέρους από τετράγωνα.
  - Rational Roots:  
Εύρεση των ρητών ριζών πολυώνυμων.
  - Factor:  
Απλή παραγοντοποίηση πολυώνυμων.

Στην επομένη εικόνα μπορούμε να δούμε την λειτουργία Advanced στην όποιο ο χρήστης μπορεί να εισάγει εντολές του Xcas.

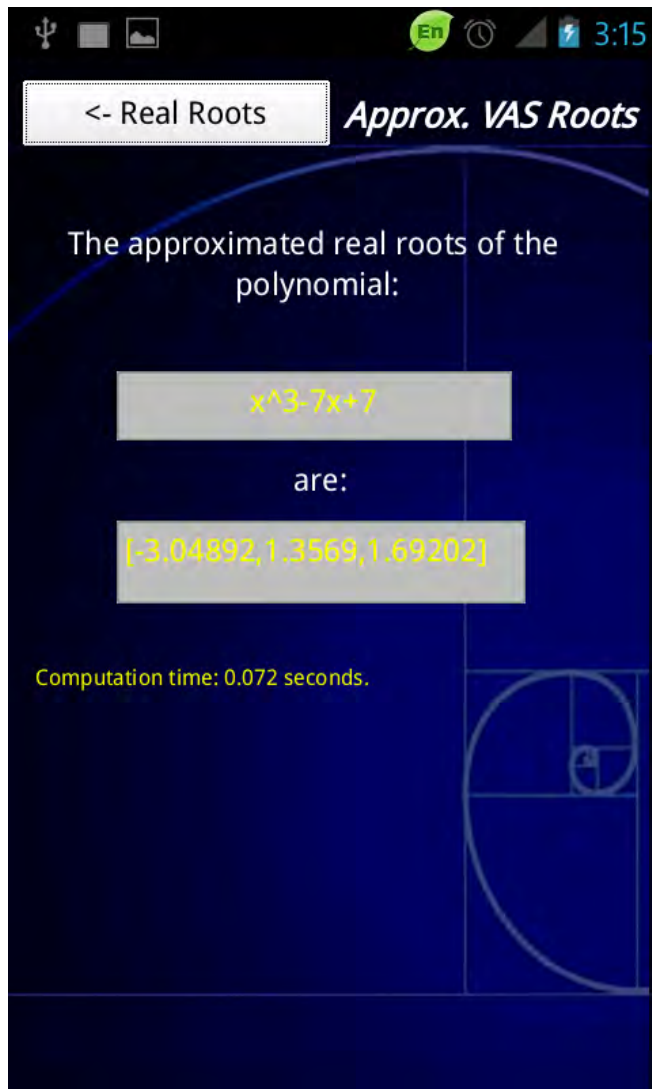


Εικόνα 34. Advanced

Έπειτα στις επόμενες εικόνες παραθέτουμε διάφορα screenshot από την εφαρμογή μετά από εκτέλεση διάφορων λειτουργιών της.



Εικόνα 35. Εφαρμογή της μεθόδου VAS



Εικόνα 36. Approximation VAS Roots



## Υπογραφή της εφαρμογής μας με RSA κλειδί

Εφόσον έχουμε έτοιμη την εφαρμογή μας για διανομή το πρώτο πράγμα που θα πρέπει να γίνει στην εφαρμογή μας είναι να την υπογράψουμε με κάποιο RSA κλειδί το οποίο θα πρέπει να υπάρχει πάντα στον υπολογιστή μας και με αυτό να υπογραφούμε κάθε αναβάθμιση της εφαρμογής που θα θέλουμε να ανεβάσουμε στο Google Play.

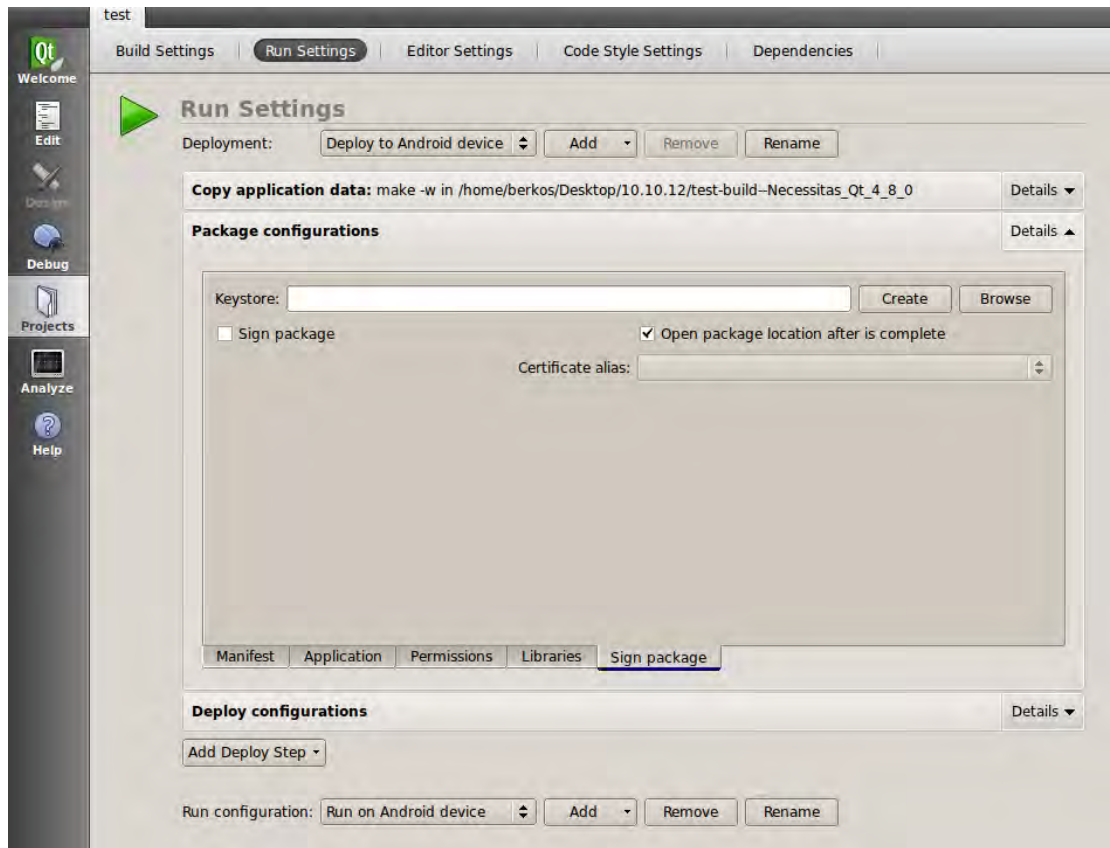
Στον φάκελο του προτζεκτ κάτω από το path 'project'/android/bin/ υπάρχει το.apk της εφαρμογής μας το οποίο μπορούμε να παρατηρήσουμε ότι έχει όνομα του τύπου “ “my\_app”-debug-unsigned.apk” το οποίο και θέλουμε να κάνουμε signed για να το δεχτεί το Google play.

Όσον αφορά το unsigned application αυτό μπορεί να διανεμηθεί σε κάποιους χρήστες χειροκίνητα ( δηλαδή εκτός Google Play) για testing με την προϋπόθεση η συσκευή τους να μπορεί να τρέξει μη υπογεγραμμένες εφαρμογές. Φυσικά κατά κύριο λόγο ο emulator τρέχει unsigned εφαρμογές της οποίες και δοκιμάζουμε.

Για να γίνει η εφαρμογή μας signed υπάρχει είτε ο κλασσικός τρόπος μέσω κονσόλας για την δημιουργία ενός RSA κλειδιού και την υπογραφή αυτού . Εναλλακτικά το Qt Necessitas μας δίνει ένα πολύ εύχρηστο γραφικό περιβάλλον με το οποίο μπορούμε πολύ απλά να υπογράψουμε την εφαρμογή μας.

Τρόπος υπογραφής μέσω QtNecessitas:

Αρχικά πατάμε την επιλογή από την αριστερή μπάρα “Projects” και έπειτα στο Run Settings μπορούμε να δούμε την μπάρα Package configurations, μόλις πατήσουμε διπλά σε αυτήν την επιλογή Details θα μας δώσει την παρακάτω εικόνα εφόσον πατήσουμε Sign Package.



Εικόνα 38. Package configuration

Πλέον επόμενο βήμα είναι να δημιουργήσουμε ένα νέο keystone πατώντας create και να εισάγουμε αληθινά στοιχεία στην παρακάτω καρτέλα καθώς αυτά θα είναι στην διάθεση της



Google.

**Create a keystore and a certificate**

**Keystore**

Password:

Retype password:

Show password **Password is too short**

**Certificate**

Alias name:

Keysize: 2048

Validity (days): 10000

Password:

Retype password:

Show password **Password is too short**

**Certificate Distinguished Names**

First and last name:

Organizational unit (e.g. Necessitas):

Organization (e.g. KDE):

City or Locality:

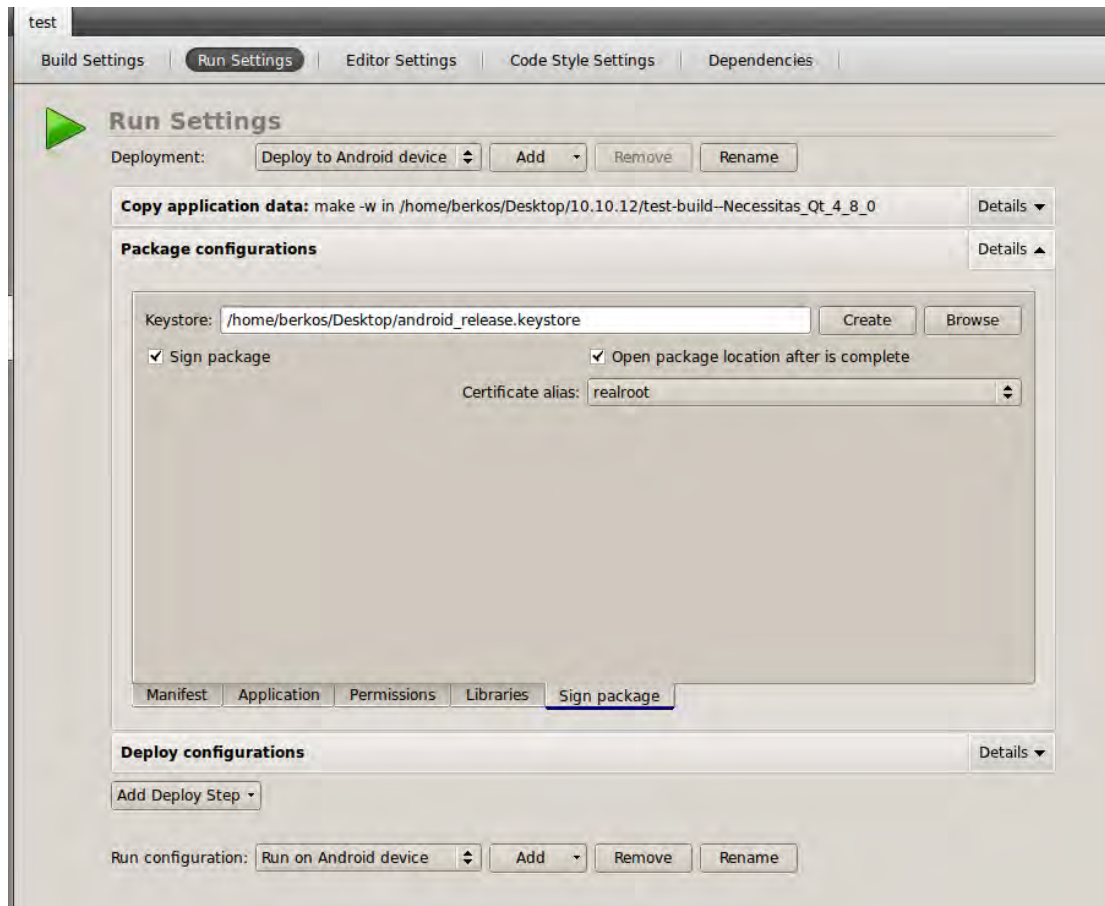
State or Province:

Two-letter country code for this unit (e.g. RO):

Save Close

Εικόνα 39. Keystore

Εφόσον δημιουργήσουμε ένα keystore και το αποθηκεύσουμε στον υπολογιστή μας απλά μένει να επιλέξουμε το sign package όπως στην παρακάτω εικόνα.



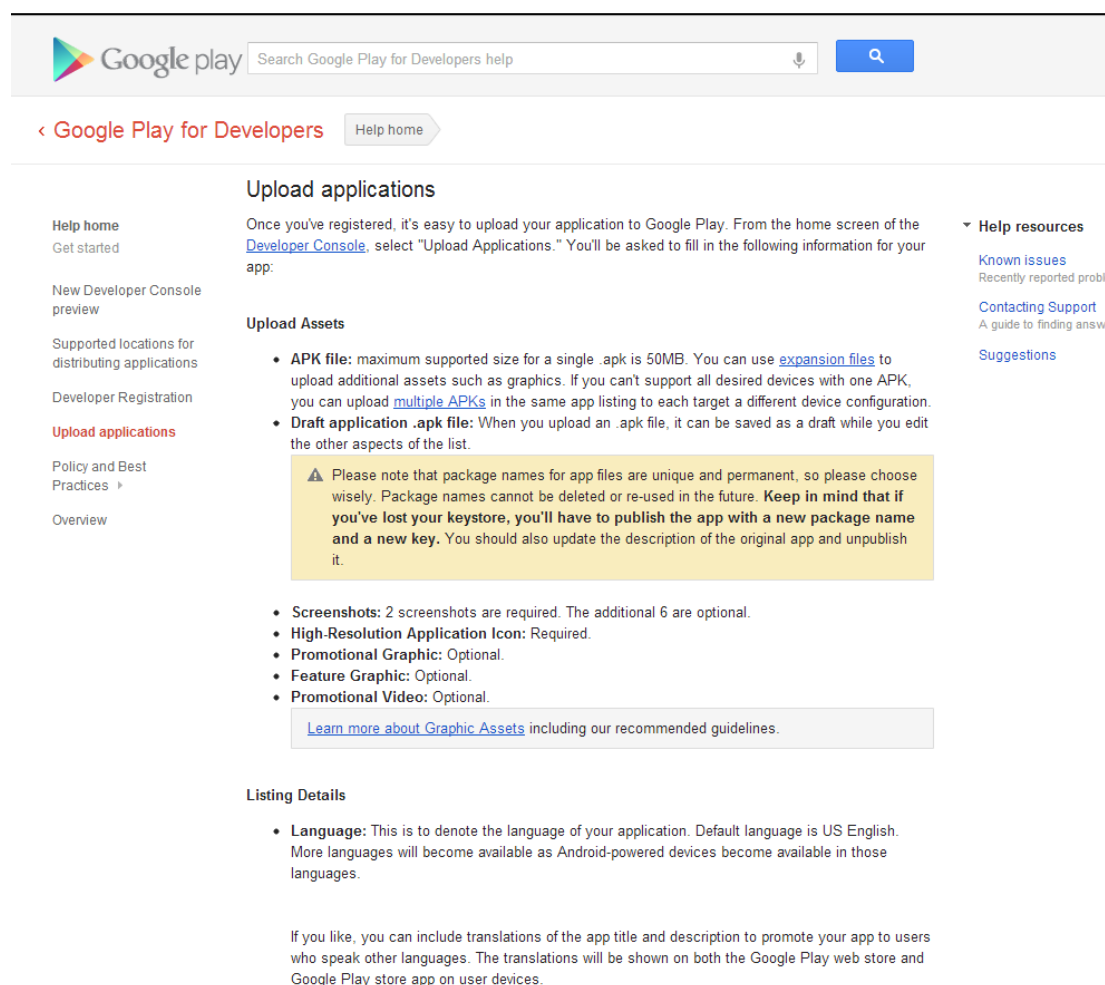
Εικόνα 40.Τελικο στάδιο Package configurations

Πλέον είμαστε έτοιμοι να πατήσουμε πάλι το κουμπί “play” στην αριστερή μπάρα του Qt Necessitas. Τώρα εκτός ότι θα τρέξει πάλι την εφαρμογή μας θα μας ανοίξει αυτόματα και τον φάκελο στον οποίο θα βρίσκετε το my\_app\_signed.apk το οποίο και θα ανεβάσουμε στο Google Play.

## Google Play

Για να ανεβάσει κάποιος την εφαρμογή στο Google Play , αρχικά θα πρέπει να εγγράφει σαν Developer στο Google Play και να έχει κάποιο λογαριασμό.

Έπειτα μπορεί να ακολουθήσει τις οδηγίες όπως αυτές υπάρχουν στο παρακάτω σύνδεσμο <http://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=113469>



The screenshot shows the 'Upload applications' page in the Google Play Developer Console. The page has a header with the Google Play logo and a search bar. Below the header, there are navigation links for 'Google Play for Developers' and 'Help home'. The main content area is titled 'Upload applications' and contains the following information:

- Help home**: Get started
- New Developer Console preview**
- Supported locations for distributing applications**
- Developer Registration**
- Upload applications** (highlighted)
- Policy and Best Practices**
- Overview**

**Upload applications**

Once you've registered, it's easy to upload your application to Google Play. From the home screen of the [Developer Console](#), select "Upload Applications." You'll be asked to fill in the following information for your app:

**Upload Assets**

- **APK file**: maximum supported size for a single .apk is 50MB. You can use [expansion files](#) to upload additional assets such as graphics. If you can't support all desired devices with one APK, you can upload [multiple APKs](#) in the same app listing to each target a different device configuration.
- **Draft application .apk file**: When you upload an .apk file, it can be saved as a draft while you edit the other aspects of the list.

**Warning:** Please note that package names for app files are unique and permanent, so please choose wisely. Package names cannot be deleted or re-used in the future. **Keep in mind that if you've lost your keystore, you'll have to publish the app with a new package name and a new key.** You should also update the description of the original app and unpublish it.

- **Screenshots**: 2 screenshots are required. The additional 6 are optional.
- **High-Resolution Application Icon**: Required.
- **Promotional Graphic**: Optional.
- **Feature Graphic**: Optional.
- **Promotional Video**: Optional.

[Learn more about Graphic Assets](#) including our recommended guidelines.

**Listing Details**

- **Language**: This is to denote the language of your application. Default language is US English. More languages will become available as Android-powered devices become available in those languages.

If you like, you can include translations of the app title and description to promote your app to users who speak other languages. The translations will be shown on both the Google Play web store and Google Play store app on user devices.

**Help resources**

- [Known issues](#)
- [Recently reported problem](#)
- [Contacting Support](#)
- [A guide to finding answer](#)
- [Suggestions](#)

### Εικόνα 41. Upload applications instructions

Τα απαιτούμενα για την εφαρμογή από το Google Play είναι

- Δυο screenshot από την εφαρμογή .
- Ένα εικονίδιο της εφαρμογής σε ανάλυση 512\*512
- Και την ίδια εφαρμογή υπογεγραμμένη.

Όταν ανέβουν αυτά πάνω στον λογαριασμό μας και ορίσουμε σε ποιες χώρες θα γίνει η διανομή κ.λ.π το μόνο που μένει είναι να πατήσουμε “Publish”.

Ο χρόνος για να ανέβει η εφαρμογή στο Google Play είναι περίπου από μίση ώρα μέχρι 12 ώρες ανάλογα με τον φόρτο εργασίας των server τους.