

Gesticulator (Gesture Based Calculator): An Android Application

Mohit Sardesai, Sanjay Yadav, Rohit Srinivasan, Mrs Bharti Joshi

Dept. of Computer Engineering, Saraswati College of Engineering, Mumbai University, India.

Abstract: We are going to make an application for Android OS. It is a gesture based calculator (Gesticulator). It will be the first app of its kind Android being touch operated opens doors for a variety of gestures. Gestures are one of the most intuitive ways of interacting with an app. Keeping this in mind we aim to reduce the number of keystrokes the user needs to make. This calculator will help us save numerous keystrokes and ultimately time. The app will basically support 4 gestures. These gestures represent the 4 main logical operations viz. Addition, Subtraction, Multiplication and Division.

Keywords: Android, Android Tools, Android SDK, Calculator, Gestures, Eclipse, and ADT plug-in

I. INTRODUCTION

We are trying to make an application that can operate on gesture performed by user. That is the calculation operations are performed by touching the touchscreen of phone in a specific direction.

A calculator is one of the most used programs in any device. It comes in perfectly handy for small quick calculations. People don't like to spend a lot of time on their calculators. Quick, Glance and Go Application. When discussing over it, we asked ourselves: Can this experience be made any better? The answer was born in the form of a Gesture based Calculator.

In order to reduce those keystrokes we omitted the operator buttons along with the (=) key altogether. In their place, Gestures were used. Android being a touch friendly device supports an array of gesture functions. Gestures allow us to depict and express something complex very easily. 4 primary gestures are used for the four arithmetic operators. The use of the (=) key is redundant since the result of computation will be displayed automatically.

TABLE I

Gesture	Operation
Left to right	+ Addition
Right to left	- Subtraction
Top to bottom	* Multiplication
Bottom to top	/ Division

Various Gestures

We found this idea from a Windows phone application which had some gesture based controls. We couldn't find any technical literature on gesture based mobile application as well as calculator. Then we studied some java applet and swing programs for Arithmetic Calculator for its GUI and basic applications from links given in references.

So we decided to make an application in Android OS where we can include gestures for operations and make it simpler to use.

The Android Architecture layers are as follows:

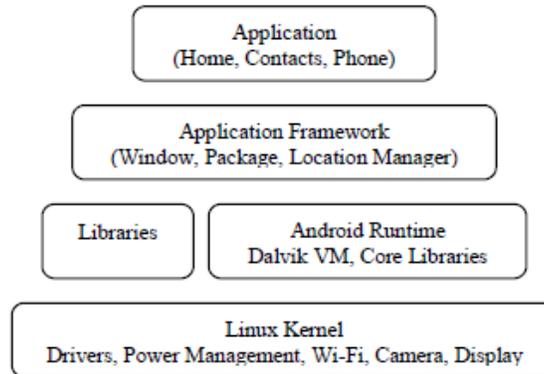


Fig 1: Android Architecture

Application Framework

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user. Underlying all applications is a set of services and systems, including:

- A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even a web browser.
- Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.
- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.
- A Notification Manager that enables all applications to display custom alerts in the status bar.
- An ActivityManager that manages the lifecycle of applications and provides a common navigation back stack.

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

- **System C library** - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices.
- **Media Libraries** - based on Packet Video's OpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- **Surface Manager** - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.
- **LibWebCore** - a modern web browser engine which powers both the Android browser and an embeddable web view
- **SGL** - the underlying 2D graphics engine.
- **3D libraries** - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer.
- **FreeType** - bitmap and vector font rendering.
- **SQLite** - a powerful and lightweight relational database engine available to all applications.

Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

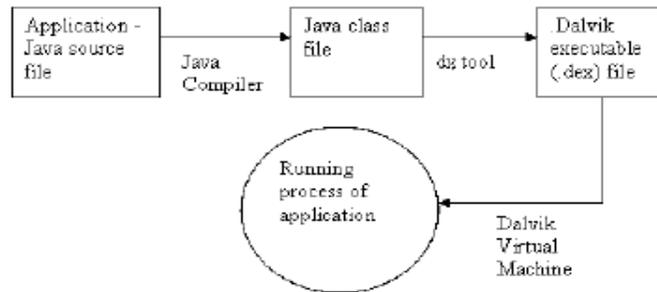
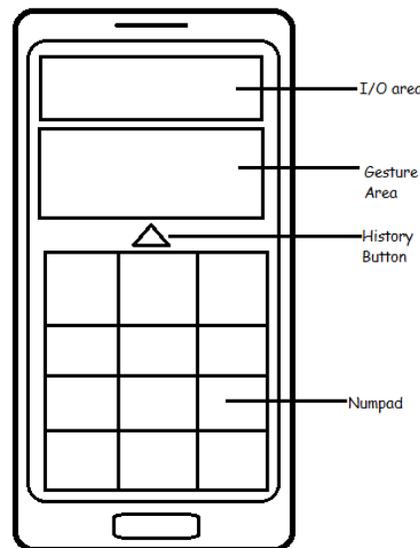


Fig 2: Dalvik Virtual Machine

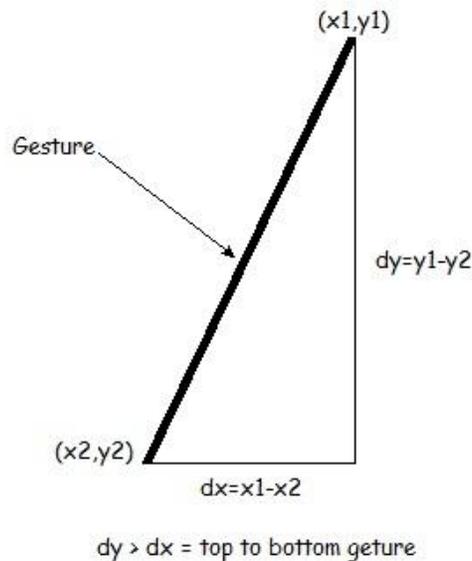
The Dalvik virtual machine is simple Java interpreter machine, completely optimized for Android platform and which is developed to run on low-end memory mobile devices. One of the prominent aspects in Dalvik is its capability to run along an application compilation enhancing the runtime performance of the applications. DVM delivers a good performance over various stages of an application runtime environment, conserving more battery-power during long run of an application. A small drawback with Dalvik is its non-compatibility with Java SE libraries, Java Applet, Java ME class libraries and Swing Java libraries. It uses its own Apache Harmony Java implementation libraries.

II.INTERFACE



Proposed System

II. GESTURE RECOGNITION



When the user tries to initiate a gesture, the start and end positions of the single gesture are recorded. D_x is the distance along the X axis. D_y is the distance along the Y axis. If $dy > dx$, a top to bottom gesture has been initiated. Similarly if $dy < dx$, a bottom to top gesture has been initiated. This technique is similarly applied to determine the direction of sideways gestures. This thus enables us to set gestures to perform logical operations.

III. ANALYSIS

Advantages

The application GUI is easy to use. The gesture makes operations very easy. Use of java programming string parsing techniques for fast calculations. First application of its kind on Google Play.

Disadvantages

As it is only Android OS based application hence it is platform dependent. As it's only a calculator so scope is narrow i.e. we cannot add more functionalities than a calculator

IV. FUTURE SCOPE

This is first application of its kind hence there is a lot scope. After its first version by considering the mistakes, flaws and deficiencies in project, newer and better versions can be made. This is the first version of this application so in the next version we will add some more functions e.g. scientific calculations. We are trying to make it more user friendly by improving the interface.

V. CONCLUSIONS

This Application provides fast, robust and extremely high Quality performance. Hence we believe we have successfully made basic calculation operations much easier to do by making the interface more intuitive. We believe this way shall save considerable time and also make the process seem effortless.

V. ACKNOWLEDGEMENT

We wish to thank everyone who have helped us and encouraged us for making this application.

REFERENCES

1. <http://jacobian.web.id/2010/03/25/creating-calculator-using-java-applet/>
2. <http://www.roseindia.net/java/example/java/swing/calculator-in-swing.shtml>
3. <http://stackoverflow.com/questions/12016391/designing-a-calculator-in-java>
4. <http://developer.android.com/index.html>

Tools References

- Download the Android SDK.
- Get the latest Eclipse IDE.
- Android source-> <http://source.android.com/>
- Android developers site (<http://developer.android.com/index.html>). It has all the latest of what's going on Android