BUCKSTAR Coffee Shop

Game Development Guide

Contents

[1. Introduction 2](#_Toc308283719)

[2. Toolchain 3](#_Toc308283720)

[2.1 General 3](#_Toc308283721)

[2.2 Programming 3](#_Toc308283722)

[2.3 Graphics 3](#_Toc308283723)

[2.4 Music & Sound 3](#_Toc308283724)

[3. Naming Conventions 4](#_Toc308283725)

[4. Versioning Scheme 5](#_Toc308283726)

[5. Code Documentation 6](#_Toc308283727)

[5.1 Source File 6](#_Toc308283728)

[5.2 Class 6](#_Toc308283729)

[5.3 Enum 6](#_Toc308283730)

[5.4 Global 6](#_Toc308283731)

[5.5 ToDo 6](#_Toc308283732)

[6. Android OS 7](#_Toc308283733)

[6.1 Android Fundamentals 7](#_Toc308283734)

[6.2 Android Tutorials 7](#_Toc308283735)

[6.3 UI GuideLines 7](#_Toc308283736)

[7. Graphics 8](#_Toc308283737)

[7.1 Supporting Multiple Screens 8](#_Toc308283738)

# Introduction

This document outlines some of the key topics required for development. It serves as a guide ranging in topics from the setup of the development environment, to some of the tools, libraries, and additional software that will be used to create the game.

# Toolchain

## General

* Microsoft Office Suite
* Project Management: Github (Overhed/buckstar)

## Programming

* Language: JAVA
* Integrated Development Environment: Eclipse with Android Plugin
* Target Android API Level: 2.2
* Data Storage: SQL Lite
* Source Control: Git (GitHub)
* Graphics Engine: Android 2D Engine
* UML: Star UML

## Graphics

* Adobe Suite

## Music & Sound

* TBD

# Code Conventions

## 3.1 General Naming Conventions

The source code created is to follow standard Java/Eclipse naming conventions. See details below.

* File Names: MyFile.java
* Classes/Enumerations: MyClass
* Methods: myMethod
* Contants: MY\_CONSTANT
* Variables: myVar

## 3.2 Source Code Conventions

The source code will follow standard Java/Eclipse conventions. See samples below

* Java Files

package com.example.helloandroid;  
  
import android.app.Activity;  
import android.os.Bundle;  
**import android.widget.TextView;**  
  
public class HelloAndroid extends Activity {  
   /\*\* Called when the activity is first created. \*/  
   @Override  
   public void onCreate(Bundle savedInstanceState) {  
       super.onCreate(savedInstanceState);  
       **TextView tv = new TextView(this);  
       tv.setText("Hello, Android");  
       setContentView(tv);**  
   }  
}

* XML Files

<?xml version="1.0" encoding="utf-8"?>  
<TextView xmlns:android="http://schemas.android.com/apk/res/android"  
  android:id="@+id/textview"  
  android:layout\_width="fill\_parent"  
  android:layout\_height="fill\_parent"  
  android:text="@string/hello"/>

# Versioning Scheme

The version number consists of three sub-numbers and two dots: maj.min.sub.

**maj** is the major version number. It is 0 for all development releases and becomes 1 with the gold release.

**min** is the minor version number. It is increased for every release beginning with the alpha version 0.1.0. It is reset to 0 for every increase of **maj**.

**sub** is the subminor version number. It is increased for every developer release or milestone and reset to 0 with every change of **min**.

0.9.0 is the beta release. It must be in a state that no further releases are necessary. Only minor changes/bugfixes are added until 1.0.0 which is gold. There are no special version numbers for release candidates. They are just codenamed RCx, where x is the candidate number. Releases have a codename, no conventions are made for it.

# Code Documentation

## Source File

Preamble for source code file:

/\*\*

\* @fileName.java

\* @authors Author1, Author2 (List of developers who actually touched the file)

\* @createDate MM/DD/YYYY

\* @brief A short description of the contents.

\*/

## Classes/Enums

Every class/enum is preceded by comment block with the following:

/\*\*

\* @brief <short description>

\*

\* <full description>

\*/

All other comments in a class should be based on need, such as before obscure-looking code.

It is important that comments explain the “Why”, not so much the what, as the source code should do this.

## To Do

To Do comments can be a single line, in the following format:

// @todo Brief explanation

# Android OS

## Android Fundamentals

<http://developer.android.com/guide/topics/fundamentals.html>

## Android Tutorials

<http://developer.android.com/resources/browser.html?tag=tutorial>

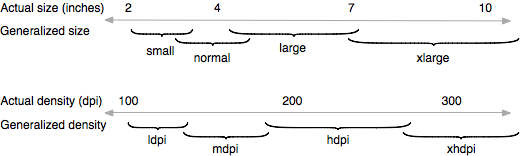
## UI GuideLines

<http://developer.android.com/guide/practices/ui_guidelines/index.html>

# Graphics

## Supporting Multiple Screens

* When defining UI layout, resolution should be expressed in units of Density-independent pixels (dp). A dp is a virtual pixel unit that is equivalent to one physical pixel on a 160 dpi (dots-per-inch) screen. 160 dpi is the baseline screen density assumed by the system for a “medium” density screen. “At runtime, the system transparently handles any scaling of the dp units, as necessary, based on the actual density of the screen in use.”
* Screen Size Ranges (measured diagonally) & Screen Density (dots-per-inch)



*xlarge* screens are at least 960dp x 720dp

*large* screens are at least 640dp x 480dp

*normal* screens are at least 470dp x 320dp

*small* screens are at least 426dp x 320dp

* “To optimize your application's UI for the different screen sizes and densities, you can provide[*alternative resources*](http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources)for any of the generalized sizes and densities. Typically, you should provide alternative layouts for some of the different screen sizes and alternative bitmap images for different screen densities. At runtime, the system uses the appropriate resources for your application, based on the generalized size or density of the current device screen.”
* “You do not need to provide alternative resources for every combination of screen size and density. The system provides robust compatibility features that can handle most of the work of rendering your application on any device screen, provided that you've implemented your UI using techniques that allow it to gracefully resize (as described in the[*Best Practices*](http://developer.android.com/guide/practices/screens_support.html#screen-independence), below).”