



FlashBuddy Flash Card Style Guide

Version 1.0

Changelog

Date	User	Change
03/01/14	JLeidel	v.1.0: Initial Printing

Contents

1	INTRODUCTION.....	4
1.1	OVERVIEW	4
1.2	DOCUMENT ASSUMPTIONS	4
1.3	SOFTWARE REQUIREMENTS.....	4
1.4	TYPOGRAPHICAL CONVENTIONS.....	4
2	STYLE RULES.....	5
2.1	ANDROID STYLE RULES.....	5
2.2	ANDROID JAVADOC RULES.....	6

1 Introduction

1.1 Overview

The following document provides a concise introduction to the FlashBuddy programming style regulations. Any code insertions and/or patches are required to adhere to the following style and format specifications.

1.2 Document Assumptions

This document assumes basic knowledge of the Java programming language, the Eclipse Integrated Development Environment and the Android build and execution environment.

1.3 Software Requirements

This document relies upon the latest version of the Android Development Toolkit [ADT]. We rely upon the ADT version with the bundled Eclipse IDE. This ensures that the build paths and emulation environment are correctly installed and initialized.

One may obtain the latest version of the Google ADT from the following website for a variety of platforms:

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

1.4 Typographical Conventions

This document contains the following typographical conventions:

Bold Fixed Width Font is used for system-generated output and source code examples.

2 Style Rules

2.1 Android Style Rules

The following rules are subsets of the overall Android programming and style guide. This guide can be found here: <https://source.android.com/source/code-style.html>

- Do not catch generic exceptions
- Do not ignore exceptions
- Do not use finalizers
- Fully qualify imports
- Statements shall not share single lines other statements.
- Compound statements such as `a+=b;` are permissible
- Static single assignment is preferred. [SSA]
- Do not omit braces on if-then-else statements with a single target [operations under mask]
- All comparisons shall be enclosed in compound parenthesis for clarity
- All new classes must have descriptor names and simple constructors
- All new modules or functions within classes must have descriptive names
- All layout items must have descriptive names. [this includes string declarations in `strings.xml`]
- All colors for text must be predefined in the layout XML specification
- All predefined asset files must be manually cached in the device working directory

2.2 Android Javadoc Rules

Every file shall include a simple file header as follows. It is not necessary to include the license details on every file as this is included in the package deliverable.

```
/**
 * FlashBuddy Main Activity
 *
 * FlashBuddy Main Activity Class
 *
 * @author John Leidel
 * @author Zack Falgout
 * @author Chase Baker
 * @version 1.0
 */
```

Developers must also document every publicly accessible method using standard Javadoc formatting. In this manner, we may automatically generate developer-style documentation. An example of doing so is as follows:

Constructors:

```
/**
 * FlashBuddyDeck Constructor Overloaded to build new decks
 * @param title title is a String representing a new title
 * @param subject subject is a String representing a new subject
 */
    public FlashBuddyDeck( String title, String subject ){
        this.title = title;
        this.subject = subject;
        this.numCards = 0;
        this.cards = null;
    }
```

Methods:

```
/**
 * setTitle : sets the title of the target FlashBuddy deck
 * @param title String variable for the title
 * @return returns true on success, false otherwise
 */
    public Boolean setTitle( String title ){

        if( title.length() == 0 ){
            return false;
        }

        this.title = title;
        return true;
    }
```

