Udacity Android Developer Nanodegree - Git Style Guide

Introduction

This style guide acts as the official guide to follow in your projects. Udacity reviewers will use this guide to grade your projects. In order to reduce the confusion on what style students should follow during the course of their projects, we urge all students to refer to this style guide for their projects.

Commit Messages

Message Structure

A commit messages consists of three distinct parts separated by a blank line: the title, an optional body and an optional footer. The layout looks like this:

```
type: subject

body

footer
```

The title consists of the type of the message and subject.

The Type

The type is contained within the title and can be one of these types:

- **feat:** a new feature
- **fix:** a bug fix
- **docs:** changes to documentation
- **style:** formatting, missing semi colons, etc; no code change
- **refactor:** refactoring production code
- **test:** adding tests, refactoring test; no production code change
- **chore:** updating build tasks, package manager configs, etc; no production code change

The Subject

Subjects should be no greater than 50 characters, should begin with a capital letter and do not end with a period.

Use an imperative tone to describe what a commit does, rather than what it did. For example, use **change**; not changed or changes.

The Body

Not all commits are complex enough to warrant a body, therefore it is optional and only used when a commit requires a bit of explanation and context. Use the body to explain the **what** and **why** of a commit, not the how.

When writing a body, the blank line between the title and the body is required and you should limit the length of each line to no more than 72 characters.

The Footer

The footer is optional and is used to reference issue tracker IDs.

Example Commit Message

```
feat: Summarize changes in around 50 characters or less
```

```
More detailed explanatory text, if necessary. Wrap it to about 72
characters or so. In some contexts, the first line is treated as t
subject of the commit and the rest of the text as the body. The
blank line separating the summary from the body is critical (unles
you omit the body entirely); various tools like `log`, `shortlog`
and `rebase` can get confused if you run the two together.

Explain the problem that this commit is solving. Focus on why you
are making this change as opposed to how (the code explains that).
Are there side effects or other unintuitive consequenses of this
change? Here's the place to explain them.

Further paragraphs come after blank lines.

 - Bullet points are okay, too

 - Typically a hyphen or asterisk is used for the bullet, preceded
   by a single space, with blank lines in between, but conventions
   vary here

If you use an issue tracker, put references to them at the bottom,
like this:

Resolves: #123
See also: #456, #789
```