CODING CONVENTIONS USED BY (TEAM-7)

Naming

Package Names;

Are all *lowercase*, with consecutive words simply concatenated together (no underscores). For example, *mainmenu*, not *mainMenu* or *main menu*.

Class Names;

Class names are written in ***UpperCamelCase***.

Class names are typically nouns or noun phrases. For example, *SideBar* or *GameLoop*.

Test classes are named starting with the name of the class they are testing, and ending with Test. For example, *SideBarTest*.

Method Names;

Method names are written in ***lowerCamelCase***.

Method names are typically verbs or verb phrases. For example, *createScenery* or *draw*.

Same rules apply for our Tests method.

Constant Names;

Constant names use all *uppercase* letters, with words separated by underscores. But what *is* a constant, exactly? Every constant is a static final field, but not all static final fields are constants.

For example:

```
public static final Image NOMONEY_TILE_IMAGE= new Image("/assets/images/NoMoney.png");
```

Parameter Names;

Parameter names are written in ***lowerCamelCase***.

One-character parameter names is not acceptable, e.g. *mousePosition*.

Local Variable Names;

Local variable names are written in ***lowerCamelCase***, and can be abbreviated more liberally than other types of names.

Declarations

Variable Names;

One variable per declaration style. That is, every variable declaration (field or local) declares only one variable: declarations such as *int width, height* are not used.

Local variables are **not** habitually declared at the start of their containing block or block-like construct. Instead, local variables are declared close to the point they are first used, to minimize their scope. Local variable declarations typically have initializers, or are initialized immediately after declaration.

Indentation

Braces;

Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement.

Non Empty blocks: K & R Style;

Braces follow the Kernighan and Ritchie for *nonempty* blocks and block-like constructs:

No line break before the opening brace.

Line break after the opening brace.

Line break before the closing brace.

Line break after the closing brace if that brace terminates a statement or the body of a method, constructor or *named* class. For example, there is *no* line break after the brace if it is followed by else.

Empty blocks: may be concise;

An empty block or block-like construct *may* be closed immediately after it is opened, with no characters or line break in between ({}), unless it is part of a *multi-block statement* (one that directly contains multiple blocks: if/else-if/else or try/catch/finally).

Block Indentation: +4 spaces;

Each time a new block or block-like construct is opened, the indent increases by four spaces. When the block ends, the indent returns to the previous indent level. The indent level applies to both code and comments throughout the block.

One Statement per line;

Each statement should be followed by a line-break.

Comments

Javadoc Comment Style;

All classes including its states and behaviors must be commented using the Javadoc API standard. This style uses the form /** . . . */, Also single line comment style is also allowed using the form //. . . .