***RISK: The Conquest Game***

**Advance Programming Practise Project (SOEN 6441)**

**Concordia University - Fall 2018**

|  |  |
| --- | --- |
| **Team 29** | **Submitted to:** |
| Jatin Rupeja (40084383) | Dr. Joey Paquet |
| Aanand Kacha () |  |
| Gunvansh Bhatia () |  |
| Roohani Naik () |  |
| Kunal Ghai () |  |

**Coding Conventions For Risk Game**

## Code Layout:

1. **Indentation**:

* Tab with four spaces is used for indentation.
* The body of a function/method must be indented with respect to its function header; the body of a **for**, **while**, or **switch** statement must be indented with respect to its first line; and similarly for **if** statements and other nested structures.

1. **Blank Lines**:

* One blank line is used in the following circumstances:

Between methods.

Between the local variables in a method and its first statement.

Between major sections of a long and complicated function.

Between public, protected, and private sections of a class declaration.

Between class declarations in a file.

1. **Blank Spaces**:

* A keyword followed by a parenthesis should be separated by a space.

Ex:

while (true) {

..

}

The expressions in a for statement should be separated by blank spaces.

Ex: for (expr1; expr2; expr3)

1. **Code Structure**: Java source files have the following ordering:

* Beginning comments

/\*

\* *Classname*

\*

\* *Version info*

\*

\* *Copyright notice*

\*/

* Package and Import statements; for example:

**package** Controller;

**import** java.awt.Color;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.image.BufferedImage;

**import** java.io.File;

* Class and interface declarations and logic implementation with methods.

1. **Line Length**:

* Avoid lines longer than 80 characters, since they’re not handled well by many terminals and tools.
* When an expression will not fit on a single line, break it according to these general principles:

Break after a comma.

Break before an operator.

Prefer higher-level breaks to lower-level breaks.

Align the new line with the beginning of the expression at the same level on the previous

line.

If the above rules lead to confusing code or to code that’s squished up against the right

margin, just indent 8 spaces instead.

Here are some examples of breaking method calls:

function(longExpression1, longExpression2, longExpression3,

longExpression4, longExpression5);

var = function1(longExpression1,

function2(longExpression2,

longExpression3));

1. **Blank Spaces**:

* A keyword followed by a parenthesis should be separated by a space.

Ex:

while (true) {

..

}

The expressions in a for statement should be separated by blank spaces.

Ex: for (expr1; expr2; expr3)

## Naming conventions:

* **Classes**:

Class names should be nouns, in mixed case with the first letter of the first word capitalized and every first letter of the internal word capitalized. Try to keep class names simple and descriptive. Use whole words—avoid acronyms and abbreviations.

* **Methods**:

Methods in mixed case with the first letter lowercase, with the first letter of each internal word capitalized.

* **Variables:**

Variable names should be short yet meaningful. The choice of a variable name should be mnemonic— that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary “throwaway” variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

First letter of internal word to be capitalized.

* **Constants:**

The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by underscores (“\_”).

# Comments

Implementation of comments:

* **Single-Line Comments:**

Short comments can appear on a single line indented to the level of the code that follows.

* **End-Of-Line Comments:**

The // comment delimiter begins a comment that continues to the newline. It can comment out a complete line or only a partial line.

* **Documentation Comments:**

Doc comments describe Java classes, interfaces, constructors, methods, and fields. Each doc comment is set inside the comment delimiters /\*\*...\*/, with one comment per API. This comment should appear just before the declaration:

/\*\*

\* The Example class provides ...

\*/

class Example { …}

Note: The coding conventions have been picked selectively from the below reference.

Reference: https://www.oracle.com/technetwork/java/codeconventions-150003.pdf