***RISK: The Conquest Game***

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

**Concordia University - Winter 2019**

Team Number 24

|  |  |  |
| --- | --- | --- |
| S.No. | Name | Student Id |
| 1 | Chirag Vora | 40091454 |
| 2 | Shashank Rao | 40104247 |
| 3 | Yogesh Nimbhorkar | 40093384 |
| 4 | Pooja Dhir | 40104545 |
| 5 | Neha Dighe | 40093384 |

CODING CONVENTIONS

# Code Layout

**1.1 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 **for**, **while**, or **switch** statement must be indented with respect to its first line and similarly for **if** statements and other nested structures.

**1.2 Blank Lines**:

* One blank line is used in the following circumstances:

- Between function and method definitions.

- Between major sections of a long and complicated function.

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

- Between class declarations.

**1.3 Blank Spaces**:

* Use space between keyword and its condition.

Eg:

if (condition) {

…

}

Separate for statement with its expressions by providing blank space.

Eg: for (expr1; expr2; expr3)

**1.4 Code Structure**: Java source files have the following ordering:

* Beginning comments

/\*

\* <h2>*Classname</h2>*

\* class description

\*

\* @author

\* @version

\*

\*/

* Package and Import statements; for example:

**package** Controller;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.List;

**import** java.util.Scanner;

**1.5 Line Length**:

* Avoid lines longer than 80 characters, since they’re not handled well by many terminals and tools.
* Following are the principles where you have to break the line if expressions don’t fit in a single line.:
* After comma break the line.
* Before an operator break the line.
* While using the last line we can align the new line with beginning of expression.

**2. Naming Conventions**

Naming convention is used to guess the “kind” of name without looking declarations. Generally, ‘Camel Case Style’ is used in naming convention where a single word is formed by multiple words, keeping first letter capital of each word so that it will be easy for user to read it.

**2.1 Package**

* In Package, names are lowercase without having underscore between the words.

**2.2 Class**

* Naming convention for Class name starts with capital letter. Interface also follow the same notation of a noun phrase or adjectives.
* For Test classes we use same name when we tested at starting till the end of the test.

**2.3 Methods**

* For Method names use lowerCamelCase. Eg getRiskPlayer ( ). JUnit Test Methods also follow the same notation of lowerCamelCase and generally start with the word 'test' succeeded by the tested method name.

**2.4 Parameter Names**

* For Parameter Names use lowerCamelCase and for public methods avoid one letter parameter name.

**2.5 Constants**

* Constants are named with all uppercase letters and may include underscores.

# 3. Comments

Application for comments:

* **Single Line Comments:**

According o the length of code, short comments can be used on a single line.

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

The // comment delimiter begins a comment that continues to the newline. To comment out partial or complete line it can be used.

* **Documentation Comments:**

Documentation comments covers Java classes, constructors, interfaces, methods and fields. For documentation comment use /\*\*…\*/ with one comment per API.