Software Development PBL

Morra Game

John Frazer - x16138015

Marcos Figueira - x16135041

Benjamin Adeline - x16135083

Table of Contents

[Project Report 2](#_Toc480625394)

[Roles 2](#_Toc480625395)

[Input, Processing, Output 2](#_Toc480625396)

[Java Classes 2](#_Toc480625397)

[MorraApp.java 2](#_Toc480625398)

[MorraGameLogic.java 3](#_Toc480625399)

[MorraPlayer.java 3](#_Toc480625400)

[MorraCPUPlayer.java 3](#_Toc480625401)

# Project Report

## Roles

John took on the role of tech lead, taking the brief and building an API framework which the team would then be able to fill in.

## Input, Processing, Output



IPO Morra Game Project.xlsx is also included in the upload zip.

## Java Classes

In all Java classes, we have included all code written by the team members (commented out where appropriate to avoid duplicate methods). All code is noted with the JavaDoc @author tags to show who wrote what.

### MorraApp.java

This is the main class of the application, containing the main method. In this method, the game flow is written. Where needed, this code calls out to the MorraGameLogic, MorraPlayer, and MorraCPUPlayer instantiable classes.

### MorraGameLogic.java

This class contains the individual logic units for the game. Each method in this class does some function for the MorraApp class, such as request the player enter the number of fingers, or calculate and update the score.

### MorraPlayer.java

The player instantiable class. This contains the information about the player, and the getter/setter methods for these. In a lot of cases, rather than giving the methods getX or setX names, they have something more informative, such as checkScore() (this is the equivalent of getPlayerScore()).

### MorraCPUPlayer.java

The MorraCPUPlayer is an extension of the MorraPlayer class. MorraCPUPlayer adds an extra method, which generates a pseudo-random number between 1 and 10 (inclusive).