**Project Report**

**Group 10**

**COMP2021 Object-Oriented Programming (Fall 2018)**

**Cheung Ka Ho (17047931d)**

**Ngan Ting Cheuk (17061434d)**

**Yan Ho Wang (17057122d)**

**Yau Yuet Man (17056093d)**

**1 Introduction**

This document describes the design and implementation of the Jungle game by group 10. The project is part of the course COMP2021 Object-Oriented Programming at PolyU. The following sections describe the requirements that were implemented and the design decisions taken. The last section describes the available commands in the game. **(To be completed…….)**

**2 The Jungle Game**

Give a short description of what the game is. **(To be completed…)**

**2.1) Requirements (To be completed..)**

**REQ01:** When the program is launched, a user should be able to choose between starting a new game and opening a saved game;

**Supporting software elements:**

Class name:

Methods:

**REQ02:** At the beginning of a new game, the two players X and Y should be prompted to input their names. Then the initial board should be printed and player X should be prompted to input a command;

**Supporting software elements:**

Class name:

Methods:

**REQ03:** A command can be a save command, an open command, or a move command:

* save [filePath]: To save the current game into file at [filePath].
* open [filePath]: To load a saved game from [filePath]. If the current game is not saved yet, prompt the player to save the current game first.
* move [fromPosition] [toPosition]: To move the piece at [fromPosition] to [toPosition]. For example, move C7 C3 means to move the piece at position C7 to position C3. You may refer to Figure 1 in Appendix A for the encoding of positions on the game board.

**Supporting software elements:**

Class name:

Methods:

**REQ04:** If a command is valid, in the sense that it can be executed successfully, the command is executed. An invalid command should not affect the game state. Continuing the previous example, if the player making the move has no piece at location C7 or the piece is not allowed to directly jump to C3, the move is invalid and will not change the state of the game;

**Supporting software elements:**

Class name:

Methods:

**REQ05:** After each valid move, the updated game board should be printed, and the game checks if a player has achieved the goal (Appendix A.5): If yes, the game is over and the program should exit after printing the name of the winning player; Otherwise, the current player's turn is terminated and the other player should be prompted to input the next command;

**Supporting software elements:**

Class name:

Methods:

**REQ06:** Upon an invalid command, an error message should be shown, the same player should be prompted to input another move. If the invalid command is a move command, the current game board should also be printed;

**Supporting software elements:**

Class name:

Methods:

**Bonus REQ01:** The game should have a full-fleged GUI mode.

**Bonus REQ02:** Two players should be able to play the game with each other on different computers.

**2.2) Design (To be completed..)**

Use a class diagram to give an overview of the game design and describe in general terms how

different components together. Feel free to elaborate on design patterns used (except for the

MVC pattern) or anything else that might help others understand the design.

**2.3) Quick Start Guide (To be completed..)**

Describe here all the commands that players can use.