Test Plan

for

Project 1 – Chess Game

Version 1.0 approved

Prepared by Madison Kuyawa, William Widmer, Bunrith Seng, Robert (James) Castleberry, Scott Flolid

CS 3398

March 01, 2018

1. Introduction

This document contains the details of our test plan for the Chess Game application. Our test plan includes plans to test code (unit tests) and also overall functionality.

1. Tests
2. Test Player Versus Player (PvP)

We will test the PvP feature. First, the player will launch the game. Next, the player will choose the PvP option in the Settings Panel. Next, the game should begin and the Settings Panel will switch to the Move History Panel. We will have the main player and a second player then begin playing the game to test for any inconsistencies.

1. Test Player Versus AI (PvAI)

We will test the PvAI feature. First, the player will launch the game. Next, the player will choose the PvAI option in the Settings Panel. Next, the game should begin and the Settings Panel will switch to the Move History Panel. We will have the player the begin playing the game to test for any inconsistencies.

1. Test Legal Moves of Each Piece

We will test the legal moves of each of the six chess pieces. We have a ‘Piece Class’ and then six classes (Pawn, Rook, Bishop, Knight, King, and Queen) that each have an inheritance relationship to the ‘Piece Class’. Each of the 6 piece classes should have their own rules in place to allow the ‘Piece Class’ to use it’s function: getLegalMoves().

1. Test Check Mate Scenarios

We will use input to create a check mate (or win) scenario to ensure the correct results/output. We will achieve this test by moving pieces to simulate a real game being played.

One expected result will be the Icon Capture Feature: when a user captures an opponent’s piece, it will be displayed under the player’s side panel. Another expected result is the move history log: which will keep track of every move each player has made.