CE203 Assignment 2 Report- Kai Roper-Blackman (1602999)

The classic game I have decided to develop for this assignment is a Pong. I chose this game because it requires me to demonstrate all the skills I have learnt during this module. Other feasible ideas that I thought of developing were Minesweeper, Snake, and space invaders. My version of Pong has a slightly different scoring system to regular Pong. Your score is derived from the time you managed to play without loosing all 5 lives given to you.

My program consists of a series of classes that represent different parts of the game. The Pong class contains a main method that is initiated at runtime. It creates and instance of JFrame which contains a JPanel at the top of the frame. The JPanel contains one button that starts a new game and a second that opens a new window showing all time high scores and high scores from the previous 24 hours. The Pong class also creates an instance of the class Game in which the main game loop is added. The pong class adds the instance of Game to the centre of the JFrame.

The Game class (extends JPanel Implementing Runnable and Key listener) consists of many methods and a constructor. The constructor creates instances of the paddle, ball, score, and thread classes. It then runs the thread allowing other tasks to happen concurrently. The Game class overrides the paintComponenet method in JComponent allowing shapes to be drawn to it. It draws the instances created in the Game constructor as well as a grid of squares. The player paddle is controlled by the up and down arrow keys. A single press moves the paddle 25 pixels in its respective direction. All movement is implemented by the use of the ActionListener method.

The Game class computes time survived and passes it to the score class. The Game class checks for conditions of other classes throughout the main game loop using a myriad of functions available. For example, the ball class checks for a collision using its checkCollision method. This takes two paddle objects as arguments, gets their y positions and checks whether the balls y position is equal to it whilst near the edge of the screen. If so, the balls xVelocity is reversed so that it travels the other way. If the ball reaches the either edge of the screen, the relevant score in the score class is incremented. The main game loop is broken when either the player or computer score reaches 5. However, the computer is unbeatable, so user will never score a point.

The highScore class's main function is to display high scores on a JFrame. It does this by creating an instance of textParser in the constructor and then calling the returnVector function to retrieve an ordered vector of all scores. The highScore class then adds the first 10 items in a vector to the JFrame. The same is performed for the top scores in the past 24 hours however todays date - timestamp must equal more than one day before an item is added to the scoreboard.

The textParser class connects to the file in which the high scores are kept and adds scores to the file or reads the file and returns an orderd vector. The vector is ordered by collections.sort() with a new comparator.

The buttonHandler class handles button input from the user and either restarts the game or displays the scoreboard using the games reset function or the highScore classes display function.

Comments- I thoroughly enjoyed this task as it challenged me to make use of all the skills I have developed over my time here at the University.