COMP30820 Main Project

Hassan Albujasim, 15436488

The project consists of 5 classes. A Player class, extended by a VIPPlayer class. A class for each game which contains their rules and choices, finally a main function that combines and test the project together.

First we will describe how the main class ties the project together by following the user flow described below:

The player class is instantiated in the main when the player enters 1.New Player, if they chose the name “boss” they become a VIP.

If they Choose 2.Quit, the game exits.

When player enters 1. The first while true loop is on for them to choose between the 2 games available or enter -1 to quit to leader board. They can return from each game by entering -1 which calls return back to the while true loop, giving them the choice between the games or -1 to Quit and view the leader board.

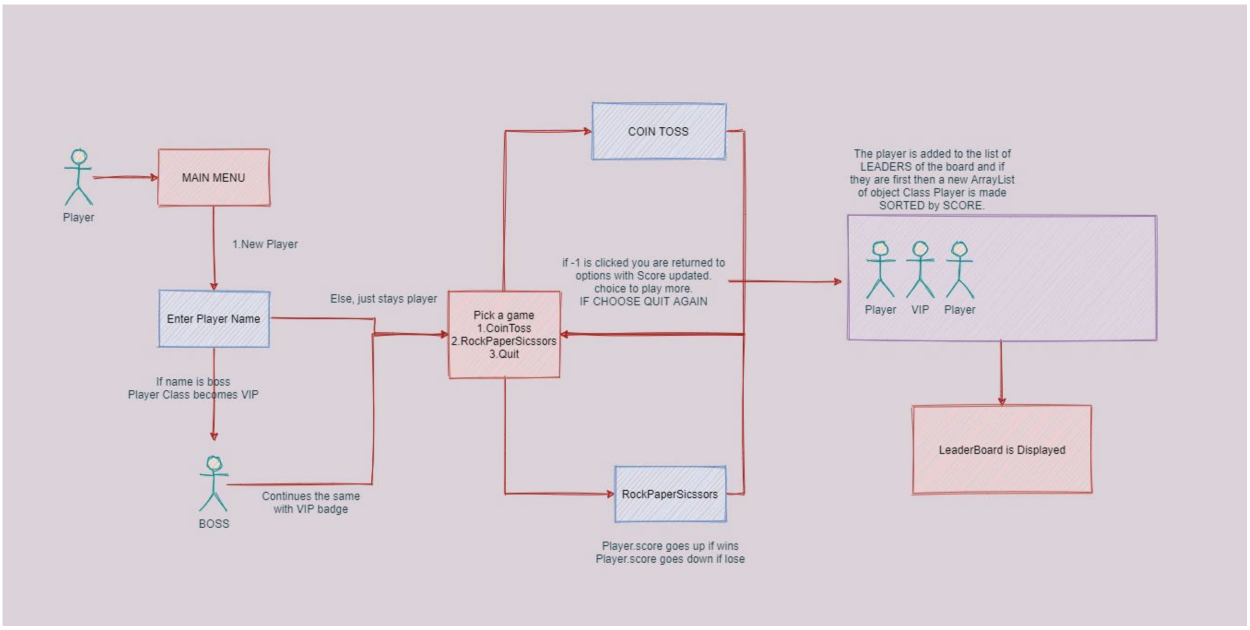
The Main also creates an ArrayList for objects of Player instances and appends them with each iteration of the call.

We create a leaderboard.txt file, we try looking for the file and reading it, adding the objects of player class into our list so that we can print the leader board and sort our persisted list of players and our new list of players.

Then we write each object of the list to the file. If the file does not exist, we write our first object to the file and creating that new file.

In the end we sort the collection of the players in ArrayList leaders and *isplayLeaderBoard*(leaders);

Below is a brief description class components of the project(UML Diagram below):



The Main Class:

The main class is described most above.

The Player Class:

The player class is a basic player with a private name and a protected score set to 0.

The constructor needs a name otherwise it is set by default to “Player_name” with the no arg constructor.

Implements Serializable, in order to save the player object into a file with readable score to allow for sorting based on score.

Player class implements comparable and overrides compareTo() to allow for sorting of the objects based on the scores.

The VIPPlayer Class:

The VIPplayer overwrites the Player class and Adds [VIP] buff for the player who chose the name “boss”.

It inherits the Player and only adds [VIP] to the Name in the main constructor.

The CoinToss Class:

The private TossCoin() returns a random heads or tails as the computer’s ` play!

The player is given an int choice 0 for head, 1 for tails and -1 to return to main menu.

If the player guess is correct, they receive 10 points with this.player.setScore(10);

If they lost the toss, they lose 5 points with this.player.setScore(-5);

The RockPaperScissors Class:

We initialise the three hand gestures, R for ROCK, P for PAPER, and S for SICSSORS. A player object and string name for the game. To create the constructor for the game which we instantiate in the main.

Void startGame() is the main function for the game that deal with the 6 conditions of win or lose for the player +1 for the tie. Awarded again with 10 for a win and -5 for a loss in the same manner as the last game. this.player.setScore().

Along with the players move options there is the -1 option to return to main menu.