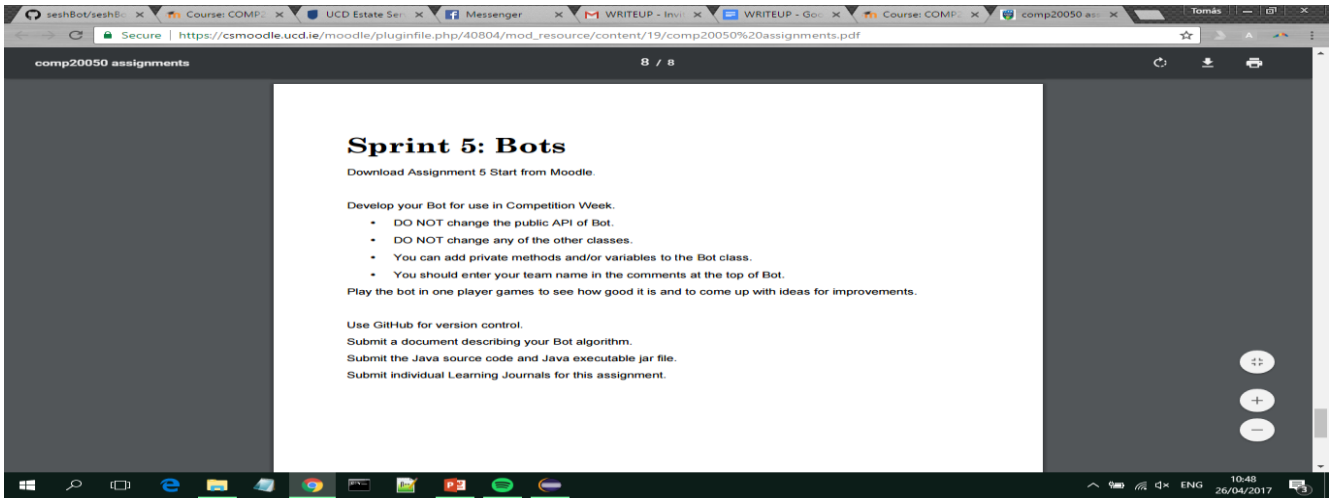
Sprint 5

Student No:
Niall Mc Guinness: 15335796
Tomas Mc Phillips: 15382091
Brian Emmett: 15352991



Strategy:

In Sprint 5 we have been asked to develop a bot to use in competition week with given APIs that we weren't allowed to change. As part of the assignment we were given a set of rules such as we weren't allowed to change the public APIs of the bot or any of the other classes.

It became clear at the start that we had to develop a Winning strategy and implement this into our bot. As a team we met up and went through the different routes, researched all the different and possible strategies that we could implement.

After we looked through and understood the code we had to use, we picked the strategy that we felt was the most effective with the code and rules that we were given. The basics of the strategy is, our bot will constantly roll, buy the property it has landed on if possible and it then mortgages the property. Our bot 'seshBot' will constantly do this until our opponent purchases a property. Once a property is purchased by our opponent our bot will then be able to check what properties have been bought and compares our assists to theirs. The 'seshBot' will constantly loop through this and will then eventually use the 'quit' function once our assets are greater than our opponents.

Our winning strategy or algorithm is simple. We constantly buy property and mortgage it straight away. We then wait until the other bot purchases a site or property and our bot checks the assets of the other player. We continue to move around the board with our bot constantly purchasing, mortgaging and checking the assets of the other player until we have a greater number of assets then our opponent. Once this is true we then use the quit function and we should win at this stage. This occurs incredibly quickly in the game, preventing us from having to implement long term strategy. This complies with the rules of the game and is a winning strategy.

Algorithm:

Our algorithm is simple. By performing checks in sequence we decide on our best options.

- Initially we check if the player is in Jail, if they are we will either use a card if they have one or pay out if their balance is above 500.

- then check if the current square is property and if it is a property we will try and buy it. This is useful before the roll as we may have a higher balance at the beginning of our go. Once a property is bought we set a mortgage on that property to be true and bought is returned.

- Once seshBot is the owner of a property the mortgage value is set to false and the property is mortgaged giving us the asset of the property as well as the mortgage value to add to our overall net worth.

- A method checks to see if the other bot has bought property and will track their assets. A check is performed to see if our total assets are more than another bots and if so we use the quit command, making us the winner. If we are against a bot that just rolls and types done seshBot will quit the game after 100 rolls.

- A binary semaphore is used to switch between roll and done. Done will be returned when increments value is 0 and once called will be changed to 1. Roll will be returned when increments value is 1 and once called will be changed to 0. This makes dealing with doubles and chance and cards easier.

PSEUDOCODE

```
Class seshbot
{
        If (jailcheck()!=”null”)
                Return jailcheck();
        if(buy()!=”null”)
                mortgagepos=propertyposition;
                mortgage=true;
                Return “buy”;
        if(mortgage==true)
                mortgage=false;
                Return “mortgage ”+mortgagepos.shortname();
        if(myassets>otherplayerassets)
                Return quit;
        if(semaphore=0)
                Semaphore++;
                Return done;
        if(semaphore=1)
                Semaphore--;
                Return roll

}
```