Gülşah Yılmaz   Akram Mustafa   Hale Şahin

Requirement Analysis for Monopoly Game Part 3

We were doing a monopoly game which play itself autamatically when the user enters the number of players. Lastly, we have added square properties like jail square(square 11 on the board), go to jail square(square 31), free parking, income tax square(square 5), and luxury tax square(square 39). Now, we are wanted to add much more specific squares. These new squares are "Lots", "Railroads" and "Utility" squares. If a player lands on these squares, player can buy the square and own the property of that square, but if some other player didn't buy the square before him/her. If so, then new player who landed an owned property, then must pay rent to the owner. When a player buy the square nothing happens.

For the Lots squares there will be no specific information like color or name. And we can use the square number as a name for that square and for the prompt to the user. The information about the squares like their position, price, rent will be read from a comma separated text file as the example file that we recieved from the supervisor. Utility squares are as Electric Utility and Water Utility, which are placed at 13 and 29. Those two squares like the Lots squares, a player who landed on that places can buy the utility, and some other player if landed that square must pay rent to the owner and once they are bought, then noone can ever buy them again. The price of utility square is 150. And the rent price will calculated by the face values of the dice, total face values times 10 is the rent. For the railroad squares, there four of them, and placed at the board in 6,16,26,36. The price of them is 200. If the square is owned by a player and some other player lands on it, then the landed player must pay the rent to the owner. And the price of rent is 5 times of the sum values of face values of thre rolled dice plus 25.

Because of the player is simulated not a real person, then the player must have decide that if he/she wants to buy a property or not. To decide that, we are wanted to check if the square is already owned. If not, then make one roll a dice, and if the number comes out is higher than 4 and the player has enough cash, than player buys that square.

If a player cannot afford to pay a rent or tax, then that player goes bankrupt and out of the game. If all players go bankrupt and leaves only one player left, then the simulation, game ends.

The number of players and the initial cash values are asked to the user at command line. And the output of the game iterations must be write down in the txt file.

Akram Mohammed Mustafa	Partial UML, Full UML
Gülşah Yılmaz	Sequence Diagram, Requirement Analysis
Hale Şahin	Implementation