Gülşah Yılmaz *Akram Mohammed Mustafa* *Hale Şahin*

Requirement Analysis for Monopoly Game

In the second iteration, we wanted add some additional properties in the squares. We will use polymorphism to make that happen. We already made a square class and we can add new specific squares inherited from the square. Firstly, we need to add a square called Go square it is the first square on the board. It means for starting, each player will take $200. Then if any player goes a whole turn at board and again comes to Go square, let say it is current position become 40 then it has to land on Go square and current position of that player becomes 0 again. And player gets $200 money for each landing in Go square. If a player goes through the Go square without landing say it landed one next to the Go square, Normally, in this situation player should take money again, but in our project, Our supevisor didnt want it from us, So we make taking money only in the landing issue.

Again, for a specific square we wanted to create a jail square. If a person comes here normally it means just visiting. But if person comes Jail Square with the command of the Go Jail Square or rolling dice double sequencially three times, than it means he is in Jail, we can make a boolean function that specifies is in Jail or just visiting. If it is just visiting, then no action, on the next turn player can move forward. But if player goes into jail, then he cannot move until he can go out. When player comes back from ‘Go Jail Square(30)’ to the ‘Jail Square (10)’ player cant get $200 moneybecause it goes directly not stopping by any squares. Player can get out from Jail by rolling double before waiting three turns or can pay bank to get out. After that player still in jail for three turns he/she goes out autamatically, if game didn’t end. If player is in jail and didn’t pay to get out, player can try to roll double but for each rolling player gives $50 money to bank. The turn of player gets out from jail, he/she cannot move until the next iteration.

There will be a free parking square at the 20th square of the board, that means no money loosing and no gaining, just normal square to pass.

There will be income tax square at the 4th square of the board, it means players has to pay 10% of their total assets which means total cash on hand.

Again adding a new square called luxury tax square which means player has to pay $75 money.

And we are gonna run the code some specific number as we defined iteration number on previous iteration. It will determines the iteration, we use 20 iteration. That means each player will play 20 times by sequencially. And each iteration is gonna prompt the user for iteration number, current player, first die value, secend die value, total die face Value, current position of current player and new position of current player, some information about the squares. A player will go bankruptcy if all of her/his money goes blow the 0 and player will removed from the game. If all players go bankruptcy than game omverş

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