Requirement Analysis Document

Monopoly Game Project

Berre Ergün – 150117507

Gülce Şirvancı – 150116047

Melisa Dönmez – 150116030

Mustafa Furkan Aras – 150116019

Requirement Specification

Vision

The purpose of this Project is to create a Monopoly Board Game with Java in accordance with object-oriented programming rules. Game can contain up to 8 players. The amount of money that players should initially have is determined by the users. Also, total number of tax squares and tax amounts will be entered by the user. The game will start after the required information is received by the user. The player who runs out of Money will be losed. The last player on the board will be the winner.

Scope

Monopoly game includes

40 square with properties; taxes, visit, park, transportation, chances ect. At

the beginning of game every player gets same amount of Money.

To decide players turn every player rolls dice, from the biggest to smallest face value of dice, players take their turn. Every player matches with specific pieces.

Players roll dice according to their turns and then moves up to the sum of the dice.

They can make some special movement according to the square which they have came. They may loose some Money if they came tax squares or if they take chance cards.

Players have to pay rent of the square to the square owner if they come to that squares.

Players can buy house if they have all colors and rent changes due to that.

The players that ran out of Money loses the game and The last player on the board will be the winner.

System Constraints

Runs on Java based platforms.

Will run as a simulation on the console with any device that has Java Runtime Environment installed, will run without GUI part.

Stakeholders

Murat Can Ganiz(Customer)

Berre Ergün (Programmer, Manager, Analyst)

Gülce Şirvancı (Programmer, Manager, Analyst)

Melisa Dönmez (Programmer, Manager, Analyst)

MustafaFurkanAras(Programmer,Manager,Analyst)

Glossary of Terms(Alphabetically Listed)

Board: Playground with 40 squares.

Die: Items that provides random number between 1-6.

Money: Cash that players own.

Piece: Object that matches with one player.

Player: Who plays the game.

Square: Areas which has unique features

Step 1

- Go Square
- Tax Square

Step 2 // Step 3

- Corporation square
- Community Chest square
- Other Square
- Station square
- Free Parking Square
- Jail Square
- City Square

Core System Functionalities

Step 1

- Number of player, tax square and amount of tax sqare's, name of players taken from user as an input.
- Players decides their turn by rolling diece.
- Players lose money on tax squares and gain money on goSquare.
- At the end players that ran out of Money loses the game and The last player on the board will be the winner.

Step 2

- Number of Jail square and die taken from user as an input.
- Game can be played with more than 2 diece.
- Players can buy City square. If they rolled diece more than 8 of sum and if they come to city square they must buy that square.
- If the square that players came is owned by some other player, they should pay the rent to the owner.
- Every city square has different amount of rent.
- If players roll double for more than 3 times they go to jail.
- There is 2 jail square one of them is for visitors and the other is going to jail square.
- If a player is in jail square she should roll double, if she cant roll double for 3 times she pays 500$ and continues the game.
- There is Chance square and Community Chest square if players come to one of them they pick a card and based on the card they do some actions.
- Players can buy Corporation square and gains rent from other players that come to this square rent changes based on the diece sum that player rolled.
- Players can buy Stations square and gains rent from other players that come to this square.

Step 3

- Rent changes according to which square player came. If the player comes to the city square and the owner of that square has all the square of the same color, owner gets double rent.

- If the player comes to the station square and the owner of that square has more than one station square, rent becomes previous rent times number of station square which owns has.
- If player comes to cooperation square and owner of that square has one cooperation square rent equals 4 times die value. If owner has both rent becomes 10 times die value.
- In city square player can buy house. To buy house player must have all the colors, and must be in that square.
- If player has house in the square, rent becomes default rent times house number +1;
- If a player runs out of money all the belongings will become bank's property.

Use Case

Step 1

User runs the application

1. At the begining user supposed to enter player number, name for every player, number of tax squares and amount of tax for every tax square, the amount of initial Money.
2. Simulation begins and without user's effect game continues automaticly.
3. Each user rolls die, player with the biggest number begins first from go square.
4. Every player rolls diece to move and updates it's location.
5. When they come to Tax Squares they loose certain amount of Money according to the square.
6. When a player run out of Money it loses the game. Last player who is on board wins.
7. In every iteration game prints informations :

   Before Rolling diece:

   - Players turn, current location and type of it, amount of Money

After Rolling dice:

- Dice numbers and sum,
- Players new location and type of it, if they lose Money amount of that, total money,
- If player rans out of money, information that it lost,
- Each cycle number at the beginning of the each iteration
- Finally the name of the winner.

Step 2

User runs the application

1. At the beginning user supposed to enter player number, name for every player, number of tax squares and amount of tax for every tax square, the amount of initial Money, number of jail square and die.
2. Simulation begins and without user's effect, game continues automatically.
3. Each user rolls die, player with the biggest number begins first from go square.
4. Every player rolls dice to move and updates it's location.

5. When they comes to Tax Squares they loose certain amount of Money according to the square.
6. When player comes to City Square either buys it, pays rent or moves on.
7. When player comes to Community Chest or Chance Square they pick a card and based on the card they do some actions.
8. When player comes to Corporation square either buys it, pays rent or moves on.
9. When player comes to Station square either buys it, pays rent or moves on.
10. When a player run out of Money it loses the game. Last player who is on board wins.
11. In every iteration game prints information :

Before Rolling dice:

- Players turn, current location and type of it, amount of Money

After Rolling dice:

- Dice numbers and sum
- Players new location and type of it, if they lose Money amount of that, total Money
- If player ran out of Money, information that it lost

- Each cycle number at the beginning of the each iteration
- Finally the name of the winner.

Step 3

User runs the application

1. At the begining user supposed to enter player number, name for every player, number of tax squares and amount of tax for every tax square, the amount of initial Money, number of jail square and die.
2. Simulation begins and without user's effect, game continues automaticly.
3. Each user rolls die, player with the biggest number begins first from go square.
4. Every player rolls diece to move and updates it's location.

5. When they comes to Tax Squares they loose certain amount of Money according to the square.
6. When player comes to City Square either buys it, pays rent, buys house or moves on.
7. When player comes to Community Chest or Chance Square they pick a card and based on the card they do some actions.
8. When player comes to Corporation square either buys it, pays rent or moves on.
9. When player comes to Station square either buys it, pays rent or moves on.
10. When a player run out of Money it loses the game. Last player who is on board wins.
11. In every iteration game prints informations :

    <u>Before Rolling diece:</u>

    - Players turn, current location and type of it, amount of Money

    <u>After Rolling diece:</u>

    - Diece numbers and sum
    - Players new location and type of it, if they lose Money amount of that, total Money
    - If player ran out of Money, information that it losed

- Each cycle number at the begining of the each iteration

- Finally the name of the winner.

Technologies

- IntellijIDEA
- Gitlab
- Draw.io