A PROJECT REPORT ON

MONOPOLY BOARD GAME

TEAM

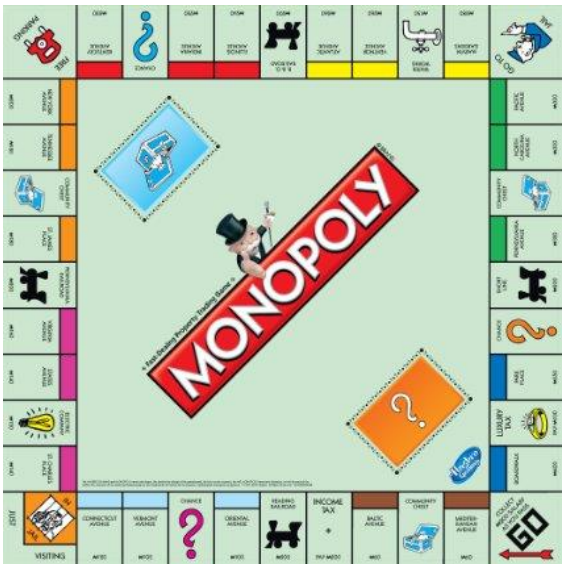
Rümeysa Eliöz (github/rumeysaelioz)

Mert Kelkit (github/mertKelkit)

Ferhat Özkan (github/ferhatozkan)

Third Step#3

About the Monopoly Board Game



Monopoly is a board game where players roll two six-sided dice to move around the game-board buying and trading properties, and develop them with houses and hotels. Players collect rent from their opponents, with the goal being to drive them into bankruptcy. Money can also be gained or lost through Chance and Community Chest cards, and tax squares; players can end up in jail, which they cannot move from until they have met one of several conditions. The game has numerous house rules and hundreds of different editions exist, in addition to many spin-offs and related media; Monopoly has become a part of international popular culture, having been locally licensed in more than 103 countries and printed in more than thirty-seven language

About the Project

Requirement Specification Vision and Scope

The aim of the "Monopoly Game Project" is to create a Java based object oriented implementation of the Monopoly Board Game. The game will run as a simulation where necessary parameters like number of players and player names are taken from the observer.

This project is developed by a team of three members. Members are as follows:

- Mert Kelkit - 150115013
- Ferhat Özkan - 150115009
- Rümeysa Eliöz - 150114016

The Project will have three steps. It will change and develop in each step depending on the demands and feedback of the customer. Requirements and

feedbacks will be listed in each step. Feedback will be analyzed after every step and changes will applied to the project.

System constraints

- Will run on any Java based platform.
- Will run as a simulation on the console with any device that has Java Runtime Environment installed.

Stakeholders

- Murat Can Ganiz (Customer)
- Berna Altınel (Customer)

Glossary of Terms

- Monopoly Game – a game which is played on the board with two dies and 2-8 players.
- Board – a playground which has 40 squares
- Die – an object which creates random values for the player.
- Piece – an object which defines every players visual identity and their location
- Player – a gamer who plays the game
- Square – every single spot which has a unique specialty
  -Income Tax Square
  -Luxury Tax Square
  -Go To Jail Square
  -Jail Square
  -Free Parking Square
  -Regular Square
  -Purchasable Square
  -Lot Square
  -Utility Square
  -Railroad Square
- Cash – amount of cash a player owns

Core System Functionalities

Step1

- Construct a Monopoly board game simulation.

- Number of iterations and number of players will be taken from user as a input
- Observer will be informed about each players turn over the console screen.
- Information will include player name, owned piece, current location, rolled dies, die sum and location after player piece moved.
- The simulation will stop after given iterations.
- Observer may choose between simulation run with a delay between each turn or the option where he/she has to press a key after each turn to observe next turn.

Step2

- Several type of squares like go square, jail square, free parking square, income tax square and luxury tax square must be implemented with their corresponding actions.
- Players land on Go Square which is the first square on the board will get $200 cash.
- At the beginning of the game each player starts from Go square and gets $200 cash.
- Passing Go square without directly landing on it wont give players cash.
- Players land on ‘GO TO JAIL’ square and players rolled doubles three times in succesion in one move will be moved to the jail which is located right across the ‘GO TO JAİL’ square.
- Players sent to jail wont collect $200 salary regardless of where he/she piece is or the path of the board. Players piece will be moved directly into the jail. After sending the player to the jail his/her turn will end.
- If the player is in jail his turn is suspended until either the player rolls a double or pays to get out.
- If a player is just visiting the Jail space is considered a ‘safe’ space, where nothing will happen.
- Rolling doubles on any of that playes’s next three turns in Jail. If a player succeeds in doing this, he or she immediately will move forward the number of spaces shown by the thow. Even if doubles are rolled, the player wont take another turn. Paying a $50 fine to the Bank before throwing the dice for either the first turn or the second turn in Jail.

- If a player lands on the income tax square he/she will pay %10 of his/her total assests.
- The free parking square is a square where nothing will happen.
- A player will go bankruptcy and will be removed from the game if his/her cash is reduced to 0 or below because of tax squares. If all players go bankruptcy before the predefined number of game iterations the game will halt.

Step3

- New type of square Purchasable Square will be implemented. Players will buy them if they can if not they will just pass them. If players pass a already bought purchasable square they will have to pay rent to the owner. Purchasable Squares will have a fixed cost taken by the .csv file.
- Purchasable Squares are specified as Lot squares, Utility squares and Railroad squares.
- The simulation will end if there is only one player left in the game (the winner).
- There will no longer be a number of iteration taken from the user.
- If a player has not enough money to pay the owner rent he/she will go bankruptcy.
- Initial cash for players will be taken as an input from the user.
- Consol output will be printed on console and monopoly-output.txt simultaneously.
- If a player goes bankruptcy his/her owned squares will be purchasable again.

Technologies & Control Mechanisms

- IntellijIDEA
- JUnit
- Github
- Trello

Project Plan & Deadlines

Step#1 - Thu Oct 26

Step#2 – Thu Nov 12

Step#3 – Thu Nov 26

Resources

www.0wikipedia.com

www.docs.oracle.com/javase/8/docs/api/

www.stackoverflow.com