REQUIREMENT ANALYSIS

VISION

This project is a monopoly game simulation. It takes json file as input. The file includes game parameters and, the game is built according to these parameters. In the beginning of the game each player rolls dices and they are sorted in descending order according to the sum of face values of dices they rolled. They play the game according to this order. When a player takes turn, the player will roll the dices, and move on the board based on the face values of dices. If player stops on the tax square, the player pays tax in amount tax square payment. When a player lands on go square it gains money.

There are some property squares in the which can be a lot square, railroad square or utility square. When the players land on these squares, they can buy them if there is no owner. If there is an owner of these squares, players need to pay rent to the owner.

Lot squares are placed on the board in colored lot groups. If a player has all the lots in a lot group, they can build(upgrade) houses or upgrade houses to a hotel. For each upgrade rent goes pricier for the other players.

There are jail square and go to jail square on the board. If a player stops on the go to jail square, he/she is sent to the jail square immediately if even the player passes from the go square cannot take money. Other condition to go to jail is, rolling double dices three times in a row.

To get out of jail player can pay money or try to roll dices twice for three rounds.

If a player rolls dice and stops on the jail square he/she is in the visitor part of the jail square. If a player is in the jail, he/she cannot play the games. The player just takes turn to try to get out from the jail. If he/she cannot it means the player is bankrupt.

The player who lost all money that have, leaves the game. Finally, the winner is announced when single player stays in the game.

SCOPE

The game has 40 squares on the board. Tax squares, and property squares are placed randomly according to config. In the staring of the program, each user has staring balance. Each user recieves money coming through rents from other players if they have properties and when they land or pass Go Square. Players are trying to bankrupt each other by trying to have monopoly on all the properties on the board.

SYSTEM CONSTRAINTS

The program starts to run with reading a json file which contains a tax square count, starting balance, players name, tax payment, salary of go square, and to arrange the speed of the simulation it takes times for sleep after turn in seconds. After reading json file the game starts. If simulation cannot start, then simulation terminates.

First, to include the libraries the steps given below is needed to follow:

File -> Project Structure -> Modules -> Dependencies -> Export -> JARs or directories -> Lib -> OK

If libraries are ready, you may need to set the SDK for the project. Following the steps:

File -> Project Structure -> Project Settings -> Project -> Project SDK -> Set a specific SDK -> OK

PROJECT MEMBERS

- Anıl ALTAY
- Ayten BİNAY
- Merve YAYIN

TERMINOLOGY

- **Go Square:** Go square is the first square on the board when players passed this square, they get salary amount of go salary.
- **Go Salary:** Go salary is that how much money a player gains when the player passes from the go square.
- **Tax Square:** Tax square is the square that when the players passed pay tax amount of tax square payment.
- **Tax Square Payment:** Tax square payment is that how much money a player pays when the player passes from the tax square.
- **Lot Square:** Lot square is one of the property squares.
- **Lot Group:** Lot group, groups lots within a colored group.
- **Railroad Square:** Railroad square is one of the property squares.
- **Utility Square:** Utility square is one of the property squares.
- **Jail Square:** This square is divided into two sections they are jail and visitor. Players can not play game in the jail part.
- **Go to Jail Square:** Go to Jail Square is the square when the players landed on go to jail.
- **Piece:** Pieces represent player of the game. Every player is represented by a unique piece and movement on the board occurs with the changing on the current location of the piece.
- **Current Location:** Current location is that index of the square that piece is on.
- **Dice Value:** Dice value is a number in between 1 and 6 which is the face value of the dice after the dice is rolled.
- **Balance of Player:** Balance of player is the amount of the money that a player has.
- **Tax Square Count:** Tax square count is that how many tax squares are on the board.
- **Starting Balance:** Starting balance is that how much money a player has when the game starting.
- **Sleep After Turn:** The time period in seconds after every single player takes turn.
- **Sleep After Piece Move:** The time period in seconds after every single piece move.
- **Rolling Double Dice:** After a player rolled the dices facing values of the dices are the same.
- **Bankrupt:** If a player loses all the money that he/she have then the player is bankrupt.

USE CASE

- When the Monopoly simulation runs it takes some parameters from the json file.
- Game starts with rolling dices of every player to takes turn; they are ordered by the face values of the dices in descending order.
- According to this order every player takes its turn.
- The player who takes turn rolls the two dices and according to the sum of the face values of dices, the player moves.
- When a player lands on a tax square, the player pays tax.
- When a player both lands and passes on a go square, gets the salary specified.
- If a player who takes turn rolls double dices, he/she takes another turn.

- If a player stops on the lot square, he/she can buy this square if it’s already bought. If there is an owner of the square, a player who stops on this square need to pay certain amount of rent. Some of lot squares are related to each other and they have some color. If a player can have all squares in the same color, he/she get double rent.
- After the end of every turn players can build on/upgrade their lots in the lot groups they own. Players allocates 40% of their money to spend on upgrading their lots with buildings and pick the lots based on random number generation.
- If a player has buildings on a lot of theirs, other players pays rent according to the increasing building count coefficient. With every building added to the lot coefficient will increase, so the rent.
- If a player having problems with paying the rent or penalty, they can downgrade the lots they have by selling buildings or they can sell whole lot groups they own.
- If a player stops on the railroad square, he/she can buy this square if it’s already bought. If there is an owner of the square, a player who stops on this square need to pay certain amount of money.
- If a player stops on the utility square, he/she can buy this square if it’s already bought. If there is an owner of the square, a player who stops on this square need to pay money to the owner. If the owner has not all the utility squares, he/she get the amount of money that is 10,000 times of the total of the facing value of the dices. If the owner has all the utility squares this amount is 40,000 times.
- If a player stops on the go to jail square, he/she is sent to the jail square immediately.
- If a player rolls dice and stops on the jail square he/she is in the visitor part of the jail square.
- If a player rolls double dices three times in a row, he/she go to the jail.
- To get out of jail player can pay money or try to roll dices twice for three rounds. If the player cannot pay the money after three turns, he/she is bankrupt must leave the game.
- In any case if a player cannot pay the money in cash, he/she can pay this money by selling his/her properties.
- When a player lost all their money or can’t pay the rent or jail penalty even by selling all his properties, they have, player is bankrupt, and they leave the game.
- When there is one player left who is not bankrupted, winner is set and game ends.