Object Oriented Software Design Project – Monopoly

Requirement Analysis Document

Authors: Mertcan Karaca – 150114060,

Mehmet Mum - 150114051,

Erkan Güngör - 150113066

Version: 0.2

Date: 14.10.2017

TABLE OF CONTENTS

1. Vision
2. Scope
3. System Constraints
4. Stakeholders
5. Glossary of Terms
6. Use Cases
7. Vision

We design a Java based Project to simulate well known board game Monopoly so that the user can observe how the game works.

1. Scope

The Monopoly Simulation will provide following functionality:

The ability for a player to move on board

The ability for a player to gain and lose money

The ability for a player to go jail and pay taxes

The ability for observer to see what is currently happening on the game.

1. System Constraints

We will run the Project with a Java Runtime Environment and that is the only needed requirement to execute Monopoly Simulation.

We will not provide a Graphical User Interface, instead of, we will provide a script-based program that does not allow anything to intercept process.

1. Stakeholders

Murat Can Ganiz(Customer)

Mehmet Mum(Programmer/Analyst)

Erkan Güngör(Programmer/Analyst)

Mertcan Karaca(Programmer/Analyst)

1. Glossary of Terms(Unalphabetically Listed)

Player: a code based character based on a real life monopoly player

Board: a code based object based on real life monopoly game board.

Dice: an object that has six faces that contains numbers 1 to 6.

Square: 1 of 40 equal parts of board that contains cities, jail, and taxes

Go Square: starting square of game

Jail: the square that keeps player 3 turns who deserves to go jail.

Go-to jail Square: the square that sends the player who lands on it to jail.

Free Parking Square: the square that nothing acutally happens when you land on it.

Income Tax Square: the square that where you pay %10 of your current assets to bank when you land on it.

Luxury Tax: the square that where you pay 75$ to the bank when you land on it.

Bankrupcy: the event when a player when have nothing left and quits game.

1. Use Cases

User executes the simulation

Actors: User, Monopoly Simulation

Precondition: None

1. User executes simulation

2. Program asks user to enter number of players

2a. If given number of inputs is according to rules, program asks user to give number of iterations to perform simulation.

2b. If not, return to 2.

3. Program creates players, board and dice.

4. The player that whose turn it is now throws dice.

4a. If the player is in jail it waits 3 turns to move unless player has a possibility to move out from jail.

4a1. If player throws a double, moves out from jail,

4a2. If player pays the bail, moves out from jail.

4a3. If three turns passed, player moves out from jail.

5. Player moves it’s piece according to sum of two dice.

6. If it is double, return to 4.

7. If the landed square is a tax square, players pay Money to bank

7a. If it is go to jail square, player moves it’s piece to jail.

7b. If the square is jail square, player is just passing,

7c. If it is GO Square, bank gives 200$ to player.

8. Return to Step 4.

9. If all players played their turns, return to Step 4.

10. If all iterations ended, terminate the game.