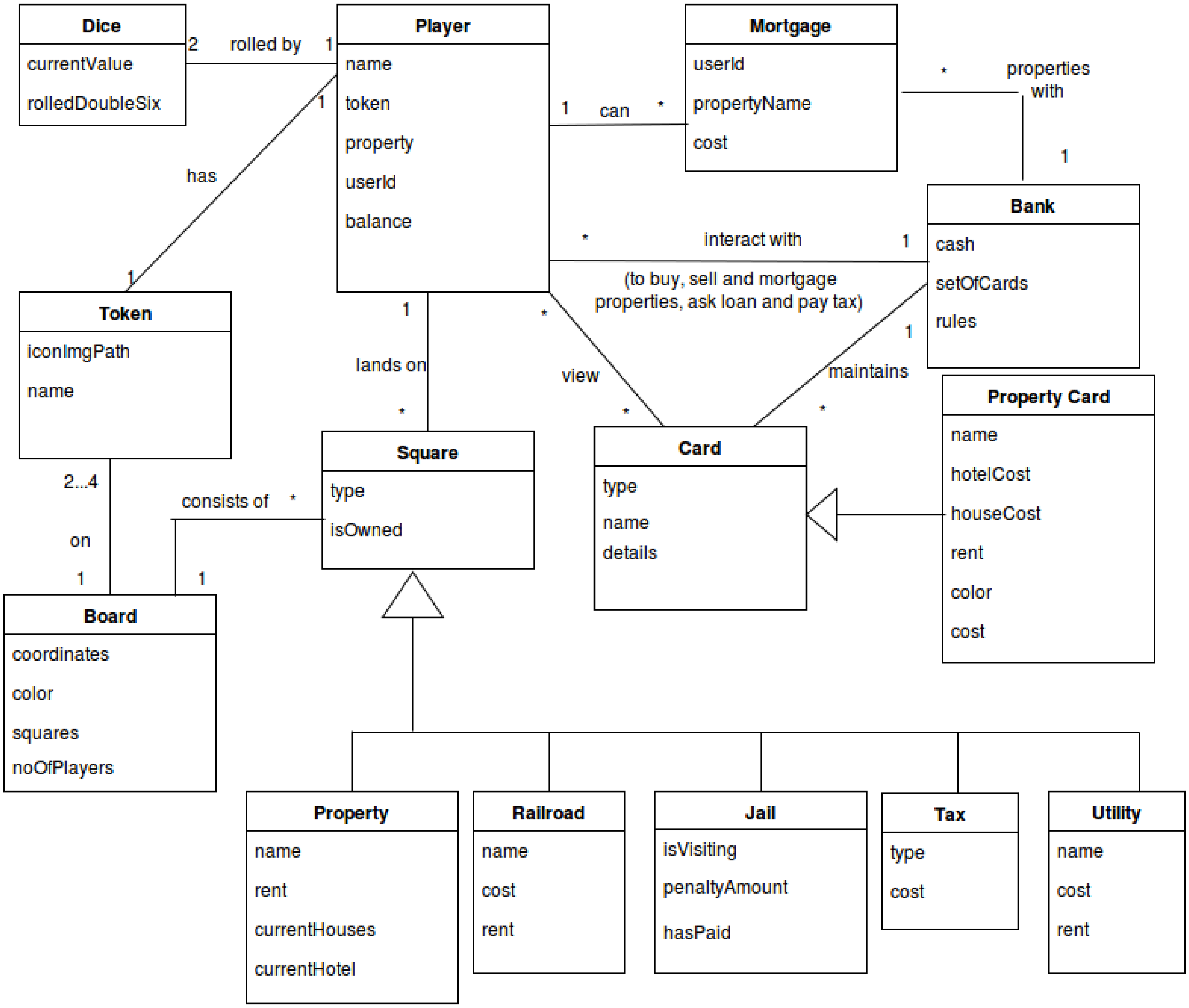
Assignment Submission (A3)
Domain Model for Monopoly Game
Group - CS414J

Team Members (CS414j)-
Ayush Mittal
Dipesh Gogia
Maalvika Bachani
Nimrata Singh

CLASS DIAGRAM



GLOSSARY

CLASS DESCRIPTION

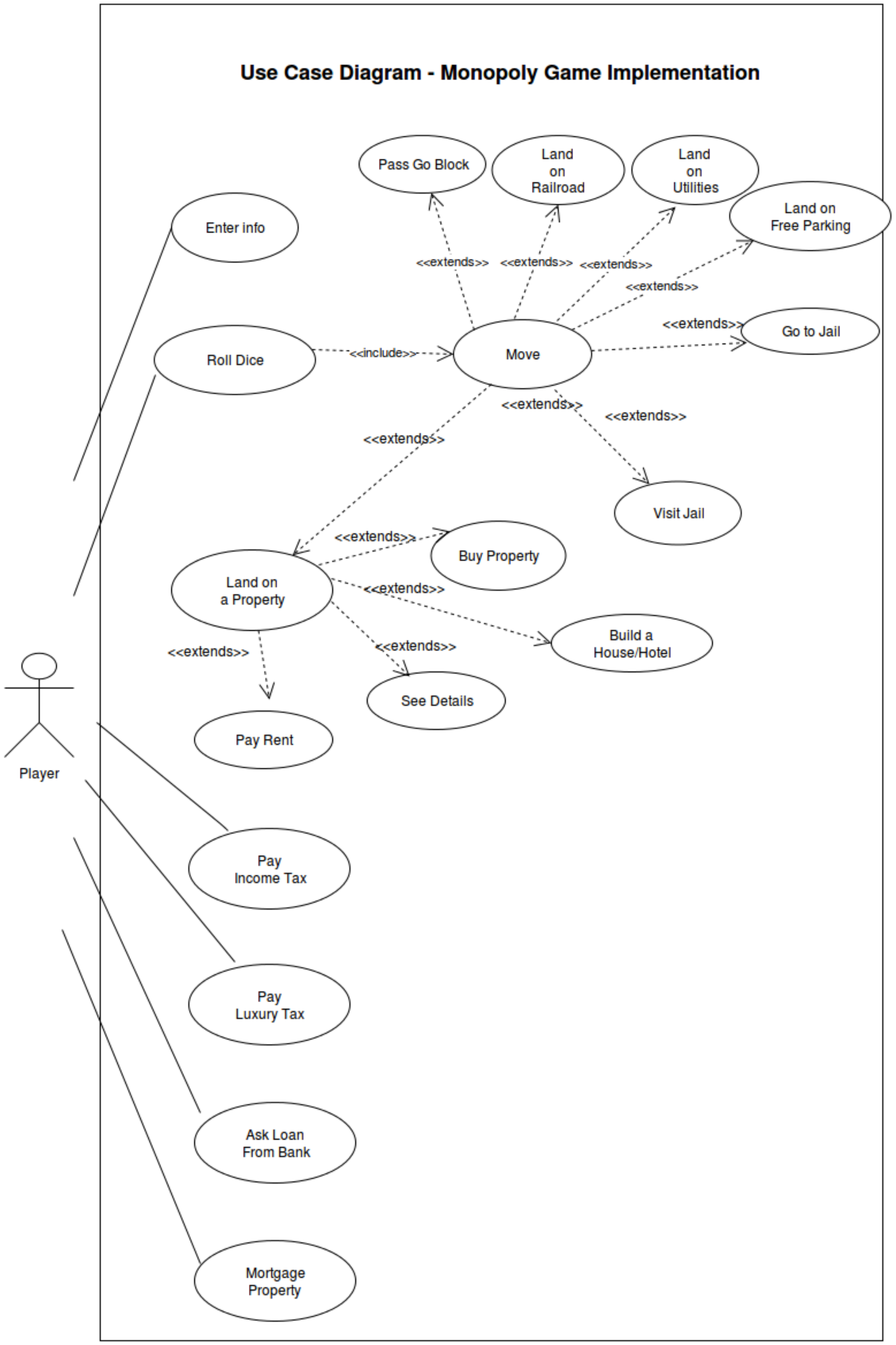
CLASS NAME	ATTRIBUTES	DESCRIPTION
Player	token, name, userId, property, balance	A user who plays the game, there can be multiple instance of it, a maximum of 4 per game. User is provided with an icon which he chooses and have some cash assigned, the cash is used to play game, buy properties and pay rent, tax as per the moves
Bank	setOfCards, cash, rules	Bank is maintained within a system; users interact with bank object as per the requirements during the game. Initially all properties are owned by bank. A user can buy property from the bank and can ask for loans from the bank as per the rules defined in the set.
Board	coordinates, color, squares, noOfPlayers	Board is stored in a matrix, the blocks at the boundary of matrix is represented by different colors and have different property. Each of those blocks at boundary have a unique coordinate. A player moves on these blocks as per the score on dice.
Square	type, isOwned	Any of the blocks in which a player can land on rolling dice.
Tax	type, cost	A type of a square where a player has to pay a certain amount (called Tax) to the bank.
Property	name,currentHouses, currentHotels, rent	A type of a square which a player can own properties and collect rent. A Player has to pay rent if he/she lands on some other player's property
Railroad	name,cost,rent	A type of a square which a player can own properties and collect rent. A Player has to pay rent if he/she lands on some other player's property
Jail	IsVisiting, penaltyAmount, hasPaid	A type of a square where player has to pay a penalty or roll a double to get out.
Utility	name,cost,rent	A type of a square which a player can own utilities and collect rent. A Player has to pay rent if he/she lands on some other player's owned utility. The rent is not fixed,it depends on the dice roll.

Card	name, type, details	It’s an abstract or generic class for different types of card. Card can be mainly of 3 categories ➔ Property Card ➔ Chance (to be implemented later) ➔ Community Chest (to be implemented later) Other cards such as rail road, jail etc., identified from the type attribute
Property Card	name, hotelCost, houseCost, rent, color, cost	It stores all the details for different properties available on the board.
Mortgage Property	userId, propertyName, cost	If a player has insufficient funds, he can mortgage his property to the bank or to the other users.
Dice	rolledDoubleSix, currentValue	It keeps track of the value appearing on each die when a player rolls it.
Token	iconImgPath, name	It keeps details of the all the tokens available in the game. Board will have the list of instances of Token.

ASSOCIATION DESCRIPTION

Name	Source	Destination	Description
has	Player	Token	Each player in the game is identified by a token.
on	Board	Token	The board game contains minimum of 2 tokens and maximum of 4 tokens in each game.
consist of	Board	Square	Board is divided into squares which can be property cell, jail, rail road etc.
Maintains	Banker	Cards	Banker maintains the card for property, community chest, etc. If user has to draw a card request goes to the banker.
Rolled by	Player	Dice	When a player has a turn, he rolls the dice and the number is recorded in the dice class.
lands on	Player	Square	Based on the number on dice, the player's token is placed on a square.
views	Player	Cards	When a player lands on a property or cell, he can request banker to view details of the corresponding property.
interacts with (Mortgage Property or Ask for Loan or pay tax or buy property)	Player	Banker	Any buying, selling or financial operation are authenticated and initiated by the bank. So a player can request the bank.
can	Player	Mortgage	In case of insufficient fund, a player can mortgage his property to other players too.
properties with	Mortgage	Bank	It indicates that the mortgage properties are maintained with bank

USE CASE DIAGRAM



UC1: ROLL DICE

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interest: Other Players waiting for their turns.

Preconditions:

1. The player information should be entered successfully.

2. The game should be in start mode.

3. If this is the first time the roll of dice is being done make sure that all the entered players have $1500 with them.

4. Roll a dice should be displayed.

5. The respective player name should be displayed along with the roll a dice option.

Post conditions:

1. The screen should display the numbers on dice.

2. A sum of both dice should be displayed on the screen.

3. A command should be issued for the token to move on the board a per the number displayed on the screen.

4. The token moves on the board.

Basic Flow:

1. The screen displays the name of player and roll dice option is available to him.

2. The user (player) clicks on the roll a dice option.

3. Random number between 1 to 6 are show and sum of both dice is displayed.

4. The token moves successfully on the board according to the sum presented in the previous step.

Alternative Flows:

a. A player throws a double once

1. He should be given one more chance to throw the dice and accordingly moves the token.

2. One more chance to throw the dice in granted.

b. If the player throws two times consecutively double.

1. The token moves accordingly.

2. One more chance to throw dice is granted.

c. If the player throws three times consecutively double.

1. In that case Send to jail command should be issued and player token should be moved to jail.

d. At any time, the system can crash, reboot or the game may end unexpectedly or even crash.

1. A log should be maintained of all the moves for all the players.

2. When the game restarts, with the help of logs the game can be reconstructed and players can resume the game.

e. The game hangs after clicking the roll a dice.

1. The player will do a force shut down of the game and restart it.

2. With the help of log files, the game will be reconstructed and players can resume the game where they left it.

Special Requirements: The response from roll a dice should come in 2 seconds.

Technology and Data Variation List:

1. Players can play this game remotely via Internet.

2. Players can play the game against the computer.

Frequency of operations: Continuously, after every move. Very Frequently

Open Issues: If game is played on internet, what is the maximum response time acceptable from a remote user, if he does not response in a given time, he should be dropped out of game.

UC2: ENTER PLAYER INFO

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interest: All the players in game

Preconditions:

1. The player should click on the option play the game on the screen.

Post conditions:

1. All the names of the players are entered successfully and the screen should display start the game option.

2. Not more than 8 players should be allowed to play in one game.

Basic Flow:

1. After clicking play in game options, screen should display the box where user can enter the name of player.

2. User clicks on the option of add a player and user will enter more names.

3. Repeat step 1 and 2 until all the name of players are entered.

4. User clicks on start the game option and the game starts.

Alternative Flows:

Case 1:

1. User try to enter more than 8 players in a single session of game.

2. The game should display a warning and provides an option of starting game with 8 players only.

Case 2:

1. User enters the names and after reaching the limit of 8 players, still try to add more by clicking on add a player option.

2. In that case warning should be displayed, if the user consecutively clicks 6 times on add players even after reaching the limit of players, the game should terminate.

Case 3:

1. If the user tries to click on the start option with only one player, a warning should be displayed showing that “need minimum 2 players to play” or choose to play with computer.

2. User selects either on one option, i.e. either user enters the name of another player or choose to play against computer.

3. If chosen against the computer, the game gives a start option on the screen.

Case 4:

1. If the previous scenario(C) occur 6 times consecutively, and player is not choosing any of the given option. The game should terminate.

Special Requirements:

After clicking play the game the screen with options of entering the name of players should appear within 20 seconds 99% time in the case if game is hosted on an online server. If game is played on a local server among different players, the response time should be in 10 seconds 99% time. The screen should have a good visual appearance.

Technology and Data Variation List: A log file should start storing the information of game so it can recover the state if game or system crashes. Only numbers and alphabets are accepted in the field of player name, no special characters are allowed.

Frequency of operations: Frequently, if game is played online or on a local server.

Open Issues: How to manage different sessions of a game on a single server if game is hosted on a local server or an online server.

UC3: ASK FOR LOAN

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interest:

1. Other Players

2. Bank

Preconditions:

1. The player asking for loan should not have sufficient fund (liquid asset) with him to pay the rent, fees or tax.

2. The player should have fixed assets like land, utility, house or hotel to ask for a loan.

Success guarantee (Post conditions):

1. The loan is granted to the player and the player pays of his rent, fee or tax accordingly.

2. No loan is granted to the player since, player does not have any fixed asset with him and the player is declared bank corrupted.

Main Success scenarios (or basic Flow):

1. The player lands on a property where he need to pay tax, fee or rent.

2. Player does not have sufficient funds with him.

3. Player mortgage a property and take a loan from the bank.

4. Player pays off his debt from the money he received from the bank.

Extensions (or Alternative Flows):

Case 1:

1. The player lands on a property where he need to pay tax, fee or rent.

2. Player does not have sufficient funds with him.

3. Player ask for loan from an another player by selling his property or assets.

4. Other player accepts the deal and the player receives cash from other player.

5. The player pays off his debt.

Case 2:

1. The player lands on a property where he need to pay tax, fee or rent.

2. Player does not have sufficient funds with him.

3. Player does not have any property or asset left out to mortgage.

4. The player is declared bank corrupted.

Case 3:

1. The player lands on a property where he need to pay tax, fee or rent.

2. Player does not have sufficient funds with him.

3. Player ask for a loan from another player and offers a deal to him.

4. The other player rejects the deals.

5. Than the player requests the bank for loan by mortgaging the property.

6. Bank grant the loan.

7. Player pay off his debts by receiving the money from the bank.

Special Requirements: None

Technology and Data Variation List: None

Frequency: Depends on the players and game, may be rare or frequent.

Open Issues: What if player request for loan even if sufficient funds are available with him. How to handle this type of request?

UC4: VISIT JAIL

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholder and Interest: None

Preconditions:

1. Player information is entered

2. Player has rolled the dice

Post Conditions: None

Basic Flow: Based on a players turn; he will roll the die. The number that appears on the die, according to that the player's token will be moved to new position. If player land on the corner of the board, they'll land on Just Visiting Jail.

Alternative Flow: None

Special Requirements: None

Technology and Data Variation List: None

Frequency of Occurrence: Not Frequent

Open Issues: None

UC5: GO TO JAIL

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholder and Interest: None

Preconditions:

Player information is entered Player has rolled the dice

Post Conditions:

1. Player has to pay fine of $50 to get out of the Jail.

2. Player don’t get normal chance.

Basic Flow: Based on a players turn; he will roll the die. The number that appears on the die, according to that the player’s token will be moved. The token lands to the “Go to Jail” block.
Alternative Flow:

1. Player throws 3 consecutive six on both dice.

2. If using the optional speed die, rolling a triple and choosing to move to "Go to Jail"

3. If using the optional speed die, rolling a bus and using a combination of the other two dice that results in landing on "Go to Jail".

Special Requirements: None

Technology and Data Variation List: None

Frequency of Occurrence: Not frequent. Occasional

Open Issues: Drawing “Go to Jail” card from community chest

UC6: GET OUT OF JAIL

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interest:

1. Player

2. Other players

Preconditions:

1. The player should be in the jail.

2. The player should have at least $50 cash with him or a get out of jail card or the player should throw a doubles.

Success guarantee (Post conditions): The player is out from the jail and resumes playing.

Main Success scenarios (or basic Flow):

1. The player is in the jail.

2. Player pays $50 fine and gets out of the jail.

Extensions (or Alternative Flows):

Case 1:

1. The player is in the jail.

2. Player throws a double in his next turn.

3. Player gets out of the jail.

Case 2:

1. The player is in the jail

2. Player uses get out of jail card.

3. Player gets out of jail and start playing the game.

Case 3:

1. The player is in the jail

2. Player is not able to get doubles by rolling the dice.

3. The above condition has appeared three times consecutively.

4. The player is forced to pay $50 fine to bank.

5. Player gets out of the jail.

Case 4:

1. The player is in the jail

2. Player is not able to get doubles by rolling the dice.

3. The above condition has appeared three times consecutively.

4. Player does not have sufficient funds to pay off the fine.

5. Player ask for loan from either bank or another player.

6. Player receives funds if the loan is approved and gets out of jail by paying $50 fine.

Special Requirements: None

Technology and Data Variation List: None

Frequency: It varies or as per demand from players.

Open Issues:

1. What if the player lands in jail, does not have sufficient funds?

2. Player asks for loan from bank or another player.

3. The loan request is rejected, in that case the player is declared bank corrupted, then what is need to be done next?

UC7: MOVE TOKEN

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholders and Interests: Player – Wants to be at correct position and take actions according to the position he lands on

Preconditions:

1. Player information is entered

2. Player has rolled the dice

Post Conditions: None

Basic Flow: Based on a players turn; he will roll the die. The number that appears on the die, according to that the player's token will be moved to new position.

Alternative Flow: If a player gets six on both dice after rolling, instead of moving he has to roll again. If thrice he gets six, his token can't move.

Special Requirements: None
Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC8: FREE PARKING

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholders and Interests: None

Preconditions:

1. Player information is entered

2. Player has rolled the dice.

Post Conditions: None

Basic Flow: Based on a players turn; he will roll the die. The number that appears on the die, according to that the player’s token will be moved. The token lands to the “Free Parking” block. Nothing happens when player land here, they move from their space in next chance.

Alternative Flow: None

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: Drawing “Go to Jail” card from community chest

UC9: BUY A PROPERTY

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholders and Interests: Player – Buys a property if he has sufficient funds

Preconditions:

1. Player has rolled the dice.

2. Lands on a property.

3. View Details

4. Have sufficient balance

Post Conditions: The cost of property is deducted.

Basic Flow:

Based on a players turn; he will roll the die. The number that appears on the die, according to that the player's token will be moved. The token lands on a property. Player views information of a property. He pays the money to bank and takes the property card in return.

Alternative Flow:

1. If player doesn't have sufficient funds, he can take loan from bank.

2. If player doesn't have sufficient funds, he can mortgage his property.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC10: Pay Luxury Tax

Scope: System Level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interests: Player: Wants to pay the luxury tax.

Preconditions:

1. Player’s turn to play.

2. Player has rolled the dice.

3. Token landed on a Luxury Tax square.

Success Guarantee (or Post conditions):

1. Player has paid the luxury tax to the bank.

2. Player is bankrupt.

Main Success Scenario:

1. Player views the indicated tax amount.

2. Players pays the tax amount.

3. System updates the player’s balance.

Extensions:

1. If the player doesn’t have sufficient money, the player has to find some alternative means to pay.

2. If the player declares bankruptcy, all the player’s assets are transferred to the bank.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: How to manage different sessions of a game on a single server if game is hosted on a local server or an online server.

UC11: PAY INCOME TAX

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interests: Player: Wants to pay the income tax.

Preconditions:

1. Player's turn to play.

2. Player has rolled the dice.

3. Token landed on Income Tax square.

Success Guarantee (or Post conditions):

1. Player has paid the income tax or 10% of the player's current assets to the bank.

2. Player is bankrupt.

Main Success Scenario:

1. Player views the indicated tax amount

2. Players pays the tax amount.

3. System updates the player's balance.

Extensions:

1. If the player doesn't have sufficient money, the player has to find some alternative means to pay.

2. If the player declares bankruptcy, all the player's assets are transferred to the bank.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Not frequent

Open Issues: None

UC12: MORTGAGE PROPERTY

Scope: System level

Level: user goal-level

Primary Actor: Player

Stakeholders and Interests: Player: Wants to mortgage property.

Preconditions:

1. Player's turn to play.

2. Player has rolled the dice.

3. Player owns the property.

Success Guarantee (or Post conditions):

1. Property has been mortgaged.

2. Player has received the mortgage amount.

Main Success Scenario:

1. Player selects the property to mortgage.

2. Player collects the mortgage amount.

3. System updates the player's balance.

Extensions:

1. Player doesn't have a property to mortgage.

2. If the player declares bankruptcy, all the player's assets are transferred to the bank.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC13: PASS GO BLOCK

Scope: System level

Level: Sub function-level

Primary Actor: Player

Stakeholders and Interests: Player: Wants to collect the Go Block amount from the bank.

Preconditions:

1. Player's turn to play

2. Player has rolled the dice.

3. Token landed on or passed the Pass Go Block.

Success Guarantee (or Post conditions):

1. Player has collected the amount from the bank.

Main Success Scenario:

1. Player collects the amount.

2. System updates the player's balance.

Extensions:

1. If the player passes the Pass Go Block because the player has been sent to jail, then the money can't be collected.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC14: LAND ON RAILROAD

Scope: System level

Level: User goal-level

Primary Actor: Player

Stakeholders and Interests: Player: Wants to buy station or collect rent.

Preconditions:

1. Player's turn to play.

2. Player has rolled the dice.

3. Token landed on a Railroad square.

Success Guarantee (or Post conditions):

1. Player has bought a station.

2. Player has collected the rent.

Main Success Scenario:

1. Player builds a station if a station is already not present.

2. Player collects $25 rent if they own one station; $50 for two; $100 for three; $200 for all four.

3. System updates the player's balance.

Extensions:

1. If the player doesn't have sufficient balance to buy a station, nothing happens.

2. If the player lands on Rail Road and does not have enough funds to pay fee, ask for loan.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC15: LAND ON PROPERTY

Scope: System level

Level: Sub function-level

Primary Actor: Player(s)

Stakeholders and Interests: Player: who rolls the dice and and lands on a property.

Other player(s): who may already own the property and ask for rent.

Preconditions:

1. Player has entered his info.

2. Player has rolled the dice.

Success Guarantee (or Post conditions):

1. The player views information about the unsold property, decides whether to buy it.

2. If property is already owned by another player the player pays rent to that player.

Main Success Scenario:

1. After rolling the dice and moving the number of spaces as the number on dice, the player lands on a property and follows by seeing details about the property cell.

Extensions:

1. If the player doesn't have sufficient balance to buy a station, nothing happens.

2. If the player lands on Rail Road and does not have enough funds to pay fee, ask for loan.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Continuous

Open Issues: None

UC16: PAY RENT

Scope: System level

Level: Sub function-level

Primary Actor: Player(s)

Stakeholders and Interests:

Player1 – who rolls the dice, lands on a pre-owned property cell and pays the rent.

Player2 – who owns the property and will receive the rent.

Preconditions:

1. Player has entered his info.

2. Player's turn to roll the dice.

3. Player landed on a property owned by another player.

Success Guarantee (or Post conditions):

1. The rent amount is added into the property owner's account.

2. The rent amount is subtracted from the player landing on another player's property.

Main Success Scenario:

1. The rent is pre defined and different for each property.

2. If the player lands on a utility cell owned by another player who has just the one utility, the dice is rolled again and the rent is 4 times the amount shown on dice and if both the utilities are owned by the same player, the one landing on it will pay 10 times the amount shown on dice.

3. If the player lands on a railroad, then rent will be $50 * 2 n-1. Extensions: If the player does not have enough money to pay the rent, he can mortgage one or multiple of his properties to bank and take mortgage value. The player can mortgage as many properties as much funds he wants to raise.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Could be Continuous

Open Issues: None

UC17: SEE DETAILS

Scope: System level

Level: Sub function-level

Primary Actor: Player(s)

Stakeholders and Interests: Player – who lands on a property.

Preconditions:

1. Player has entered his info and player turns to roll the dice.

2. Player landed on a property cell.

Success Guarantee (or Post conditions):

1. Player decides to pay the price and buy that property.

Main Success Scenario:

1. The prices are pre defined and different for each property. On landing on a property the player can see the details which include price and rent of a property, it is up to the player to buy it or not.

2. The player buys the property he landed on from bank.

Extensions:

1. The player might not buy the property. In that case, it is next player's turn to roll the dice.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Could be Continuous

Open Issues: None

UC18: BUILD A HOUSE OR HOTEL

Scope: System level

Level: Sub function-level

Primary Actor: Player(s)

Stakeholders and Interests:

Player 1 – who lands on his own property again and will pay the bank the amount to buy a house or hotel.

Player 2 – who lands on player 1's property with a building on it and has to pay more rent.

Preconditions:

1. Player's turn to roll the dice.

2. Player landed on his own property cell.

Success Guarantee (or Post conditions):

1. Player decides to buy a house each time he returns to his property.

2. Player can decide to build a hotel on returning for the third time on his own property provided he already built two houses there.

Main Success Scenario:

1. The prices for buying a house on a property is pre defined.

2. The player who lands on his property again after buying it may choose to build a house there by paying the required amount to bank.

3. After building two houses if the player returns for the third time, he might choose to build a hotel there by paying a required amount to the bank.

Extensions:

1. The player might not have funds to buy a house or hotel, thus may not choose to do so.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Could be Continuous

Open Issues: None

UC19: LAND ON UTILITY

Scope: System level

Level: Sub function-level

Primary Actor: Player(s)

Stakeholders and Interests:

Player1 – who rolls the dice and lands on a utility like electric company or water works.

Player2 – who might own the utility.

Preconditions:

1. Player has entered his info.

2. Player’s turn to roll the dice.

3. Player landed on a utility.

Success Guarantee (or Post conditions):

1. The utility is bought by the player.

2. If utility is already owned by another player, then rent is paid to that player.

Main Success Scenario:

1. The rent and prices are pre defined and different for each property.

2. Utility is bought by the player by paying a specified amount to bank.

Extensions:

1. When the player lands on a utility cell owned by another player who has just the one utility, the dice is rolled again and rent is 4 times the amount shown on dice and if both the utilities are owned by the same player, the one landing on it will pay 10 times the amount shown on dice.

2. If the player does not have enough money to pay rent, he can mortgage one or multiple of his properties to bank and take mortgage value. The player can mortgage as many properties as much funds he wants to raise.

3. The player may choose not to buy a utility; in that case nothing happens.

Special Requirements: None

Technology and Data Variations List: None

Frequency of Occurrence: Could be Continuous

Open Issues: None