**Non-Functional Requirements & Use Cases**

Needs modifiability for future updates and enhancements.

Modifying the game is a relatively simple process.

The gameplay and game concepts must be easily understood after reading the rules, taking 2-3 times max to fully understand the game concepts.

Display of game is laid out in a manner than even novice users can use with ease.

System elements are easily tested.

The player has the option to do certain things regardless if it is their go, e.g. checking cash held.

The game is not restricted to a single platform.

The game can easily be moved to other platforms.

The game should be able to run efficiently on any computer that fulfils the hardware and software interface requirements.

Within one second of finishing rolling the dice, user piece should be in the new position on the game board.

The game shall be able to accurately keep track of each user's money, properties, position, and turns.

The game should not crash under normal conditions.

There shall not be any ways to cheat in the game.

|  |  |
| --- | --- |
| **Flow of Events for the *Get Out of Jail* Use Case** | |
| **Objective** | **Player gets out of Jail** |
| **Precondition** | 1. It is the player’s turn.  2. The player has not rolled the dice.  3. The player is in jail. |
| **Main Flow** | 1. A player is in Jail. 2. The player clicks the “Get out of Jail” button. 3. £50 is decremented from their money. 4. The player can then roll the dice and continue with the game. |
| **Alternative Flows** | 1. A player is in Jail. 2. The player clicks the “Get out of Jail” button. 3. The player has less than £50. 4. The player becomes bankrupt and all the tradable cells he or she owns becomes available in the game. 5. The player is out of the game. |
| 1. A player is in jail 2. The player uses a Chance or Community Chest Get Out of Jail Free card |
| **Post-Condition** | Player is out of jail |

|  |  |
| --- | --- |
| **Flow of Events for the *Go to Jail* Use Case** | |
| **Objective** | **Player is sent to Jail** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice  3. The player will land in jail if they land on "Go to Jail", throw three straight doubles in a turn, or draw a Community Chest or Chance Card saying "Go to Jail". |
| **Main Flow** | 1. The player is sent to the Jail cell directly. 2. The player may not buy property directly as are not able to land on the properties. However, they may sell, trade and mortgage properties, purchase buildings, participate in auctions, and collect rents. |
| **Alternative Flows** | - |
| **Post-Condition** | Player is in jail |

|  |  |
| --- | --- |
| **Flow of Events for the *Enter Player Details* Use Case** | |
| **Objective** | **Player’s details are recorded** |
| **Precondition** | - |
| **Main Flow** | 1. At beginning of game, a Player Information dialog box will appear asking the players enter the number of players for the game and the name of each player |
| **Alternative Flows** | 1. The number of players must be a whole number between 2 and 8. If the players insert anything different, they will be prompted to retype the number of players again. 2. The name field cannot be empty. If a player fails to enter a string, the game asks the player to retype his/her name. 3. When the Exit button is pressed, the Player Information dialog box closes and the game ends. |
| **Post-Condition** | Player details are displayed |

|  |  |
| --- | --- |
| **Flow of Events for the *Pass Go* Use Case** | |
| **Objective** | **Player has passed Go** |
| **Precondition** | 1. It is the player’s turn. 2. The player has rolled the dice. |
| **Main Flow** | 1. If the player passes the Go cell during the movement, or if the player lands on the Go cell after the movement, the player gains £200 |
| **Alternative Flows** | 1. If the player passes the Go cell because he or she is sent to Jail, the player cannot collect the money. |
| **Post-Condition** | Player has passed Go and collected £200 |

|  |  |
| --- | --- |
| **Flow of Events for the *Visit Jail* Use Case** | |
| **Objective** | **Player is visiting jail** |
| **Precondition** | 1. It is the player’s turn. 2. The player has rolled the dice. 3. The player lands on the Jail cell. |
| **Main Flow** | 1. The player visits Jail. 2. Nothing happens. |
| **Alternative Flows** | - |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Roll Dice* Use Case** | |
| **Objective** | **Player takes turn and rolls the dice** |
| **Precondition** | 1. It is the player’s turn. |
| **Main Flow** | 1. The player rolls the dice by clicking on the Role Dice button. 2. There are two dice 3. The Dice Roll dialog pops up to indicate the value of the dice roll. |
| **Alternative Flows** | 1. The player rolls a double and can take another roll. 2. Upon rolling three doubles in a row, player is sent to jail. |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Buy House* Use Case** | |
| **Objective** | **Player buys a house** |
| **Precondition** | 1. It is the player’s turn.  2. The player has not rolled the dice.  3. The player has monopoly on one or more colour groups. |
| **Main Flow** | 1. A player can choose to build houses in the property cells in the colour groups that they have monopoly on by pressing the Buy House button before rolling the dice. 2. The price of the house is determined by the cell. 3. After buying the house, the status of the player is updated. |
| **Alternative Flows** | 1. Nothing happens if the player does not have enough money. 2. The player can build at most five houses in a cell. |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Go to Free Parking* Use Case** | |
| **Objective** | **Player lands on Free Parking** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice.  3. The player lands on the Free Parking. |
| **Main Flow** | Nothing happens to a player on the Free Parking cell. |
| **Alternative Flows** | - |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Trade Property* Use Case** | |
| **Objective** | **Player wants to trade property** |
| **Precondition** | 1. It is the player’s turn.  2. The player has not rolled the dice. |
| **Main Flow** | 1. The player may ask another player to sell their tradable cells. 2. If the player wants to trade with another player, clicks on the Trade button. 3. The Trade Property dialog box appears and the player enters the player with whom they wish to trade with, the cell they want to buy, and the amount of money willing to pay. 4. Dialog box shows up to ask the seller if the seller agrees with the deal. 5. The seller clicks on Yes and the cell is sold. |
| **Alternative Flows** | 1. If the player clicks on Cancel button, the dialog closes and the deal is cancelled. 2. If the player does not have enough money, the deal is cancelled. |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Move* Use Case** | |
| **Objective** | **Player takes turn and moves position** |
| **Precondition** | Player has rolled the dice or given movements from a drawn card |
| **Main Flow** | 1. The movement is based on the player’s dice roll: whatever number is rolled, player moves the same amount of places 2. The new position and information of the player is displayed 3. The turn ends when the player hits the End Turn button |
| **Alternative Flows** | Possible actions due to movement: at the Go cell, go to the Jail, visit Jail, stop at Free Parking, pay rent, draw a card from Community Chest or Chance or buy a tradable cell |
| **Post-Condition** | - |

|  |  |
| --- | --- |
| **Flow of Events for the *Pay Rent* Use Case** | |
| **Objective** | **Player pays rent** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice.  3. The player lands on a tradable cell that is owned by another player. |
| **Main Flow** | 1. The player pays rent to the owner of the cell. The rate of the rent depends on the type of cell the player lands on 2. Each cell may have a different rent rate. If the cell is in the owner’s monopoly colour group, the rent doubles. |
| **Alternative Flows** | 1. If the player does not have enough money to pay the rent, the player is bankrupt. He or she needs to give all the tradable cells to the owner, and is out of the game. |
| **Post-Condition** | Player has paid rent on properties |

|  |  |
| --- | --- |
| **Flow of Events for the *Draw Card* Use Case** | |
| **Objective** | **Player has to draw a Community Chest and Chance card** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice.  3. The player lands on a card cell. |
| **Main Flow** | 1. There are Community Chest cards and Chance cards. 2. When the player lands on a card cell, they draw a card by clicking the Draw Card button. 3. A card is drawn from the top of the Community Chest card pile or the Chance card pile, depending on the cell they landed on. 4. The player performs the actions on the card – collect money, lose money, go to jail, get out of jail free token, go to a particular cell. |
| **Alternative Flows** | 1. If the player does not have enough money to pay the fee, the player is bankrupt. If the fee is to pay another player, they need to give all the tradable cells to the owner, the player is now out of the game. |
| **Post-Condition** | Player has followed instructions on the chosen card |

|  |  |
| --- | --- |
| **Flow of Events for the *Pay Tax* Use Case** | |
| **Objective** | **Player pays tax** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice.  3. The player lands on a tax fee cell. |
| **Main Flow** | 1. The player pays tax to the bank. 2. The fee amount may vary depending on the cell. |
| **Alternative Flows** | If the player does not have enough money to pay the tax, the player is bankrupt. He or she needs to give all the tradable cells to the bank, and is out of the game. |
| **Post-Condition** | Player has paid tax |

|  |  |
| --- | --- |
| **Flow of Events for the *Buy Property* Use Case** | |
| **Objective** | **Player buys a property cell** |
| **Precondition** | 1. It is the player’s turn.  2. The player has rolled the dice.  3. The player lands on an unowned property cell. |
| **Main Flow** | 1. A player can choose to buy the property if they land on it and it is not currently owned. 2. The price of the property is determined by the cell. 3. After buying the property, the status of the player is updated. |
| **Alternative Flows** | 1. Nothing happens if the player does not have enough money. 2. The player now owns all cells of this type and can upgrade housing. |
| **Post-Condition** | Player now owns the property or couldn't afford it |

|  |  |
| --- | --- |
| **Flow of Events for the *Sell Property* Use Case** | |
| **Objective** | **Player sells a property cell** |
| **Precondition** | 1. The player owns the cell |
| **Main Flow** | 1. A player can choose to sell a currently owned property. 2. The player selects the sell property option. 3. The Sell Property dialog box appears and the player enters the cell they wish to sell. 4. Dialog box shows up displaying the value the bank offers. 5. The seller clicks on Yes and the cell is sold. |
| **Alternative Flows** | 1. If the player clicks on Cancel button, the dialog closes and the deal is cancelled. 2. The player has sold the property via another player trading them.   \*Trading property use case |
| **Post-Condition** | Player has sold the property to the bank or another player |

**Use Case Diagram**

