Monopoly Requirements Specification

Version 2.0

August 22, 2005

Project Team:

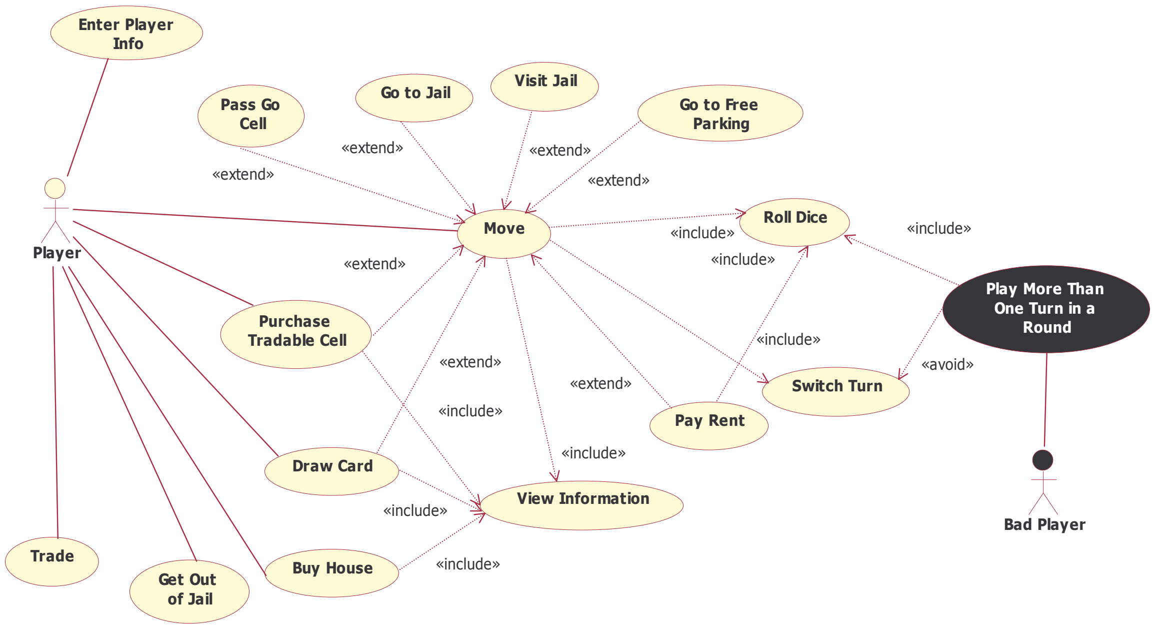
Chih-wei Ho, Team Lead
Hema Srikanth, Quality Assurance Manager
Nachi Nagappan, Requirements Analyst
Lucas Layman, Project Manager
Mark Sherriff, Development Manager

Document Author(s):
Nachi Nagappan, Requirements Analyst

Customer Representative(s):
Michael Gegick

I. Introduction

The goal of this project is to create a Java-version of Monopoly board game. This game provides several features we can see in the board game version. This document describes the requirements of this program.



II. Use Cases

UC1	Enter Player Info
UC2	Move
UC3	Pass Go Cell
UC4	Go to Jail
UC5	Visit Jail
UC6	Go to Free Parking
UC7	Purchase Tradable Cell
UC8	Buy House
UC9	Pay Rent
UC10	Draw Card
UC11	Roll Dice
UC12	Switch Turn
UC13	View Information
UC14	Get Out of Jail
UC15	Trade
UC16	Play More Than One Turn in a Round

UC1 Flow of Events for the *Enter Player Info* Use Case

1.1 Preconditions:

None.

1.2 Main Flow:

Right after the game gets started, the Player Information dialog will show to prompt the players enter the number of players for the game [E1] and the name of each player [E2] [E3].

1.3 Subflows:

None.

1.4 Alternative Flows:

[E1] The number of players is a whole number between 2 and 8. If the players do not enter a whole number, or the number is not between 2 and 8, the game asks the player to retype the number of players again.
[E2] The name cannot be an empty string. If a player enters an empty string, the game asks the player to retype his/her name.
[E3] When the Cancel button is pressed, the Player Information dialog closes and the game ends.

UC2 Flow of Events for the *Move* Use Case

2.1 Preconditions:

The players have entered their information in the Player Information dialog.

2.2 Main Flow:

The game is turn based. The first player's turn starts when the players' information is entered. The movement is based on the player's dice roll [UC11]. If the dice roll is 2, the player moves forward 2 steps; if the dice roll is 3, the player moves forward 3 steps; etc. What happens to the player depends on the cell the player lands on [S1] and whether the movement passes the Go cell [UC3]. The new position and information of the player is displayed on the game board [UC13]. The turn ends when the player hits the End Turn button [UC12]. Then the next player's turn begins.

2.3 Subflows:

[S1] After the player moves to a new cell, based on the type of the cell, he or she may stop at the Go cell [UC3]; proceed to the Jail cell [UC4]; stop at the Jail cell [UC5]; stop at Free Parking [UC6]; pay rent to the cell owner [UC9]; draw a card from Community Chest or Chance [UC10]; or purchase an available tradable cell [UC7].

2.4 Alternative Flows:

None.

UC3 Flow of Events for the *Pass Go Cell* Use Case

3.1 Preconditions:

1. It is the player's turn.
2. The player has rolled the dice.

3.2 Main Flow:

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 [E1].

3.3 Subflows:

None.

3.4 Alternative Flows:

[E1] If the player passes the Go cell because he or she is sent to Jail, the player cannot collect the money. A player can be sent to Jail either because he or she draws a Go to Jail card, or because he or she lands on the Go to Jail cell.

UC4 Flow of Events for the *Go to Jail* Use Case

4.1 Preconditions:

1. It is the player’s turn.
2. The player has rolled the dice
3. The player lands on the Go to Jail cell.

4.2 Main Flow:

The player is sent to the Jail cell directly. When a player is sent to the Jail cell, he or she is said to be *in jail*.

4.3 Subflows:

None.

4.4 Alternative Flows:

None.

UC5 Flow of Events for the *Visit Jail* Use Case

5.1 Preconditions:

1. It is the player’s turn.
2. The player has rolled the dice.
3. The player lands on the Jail cell.

5.2 Main Flow:

The player visits the Jail. Nothing happens to the Jail visitors.

5.3 Subflows:

None.

5.4 Alternative Flows:

None.

UC6 Flow of Events for the *Go to Free Parking* Use Case

6.1 Preconditions:

1. It is the player’s turn.
2. The player has rolled the dice.
3. The player lands on the Free Parking.

6.2 Main Flow:

Nothing happens to a player landing on the Free Parking cell.

6.3 Subflows:

None.

6.4 Alternative Flows:

None.

UC7 Flow of Events for the *Purchase Tradable Cell* Use Case

7.1 Preconditions:

1. It is the player’s turn.
2. The player has rolled the dice.
3. The player lands on an available tradable cell.

7.2 Main Flow:

There are three types of tradable cells in this game: property cells, railroad cells, and utility cells. A tradable cell is available if it has no owner. When a player lands on an available tradable cell, he or she may buy the cell by clicking the Purchase button [E1]. The price the player needs to pay is the land value of the tradable cell [E2]. Player's information displayed on the game board is refreshed to show the cells and the amount of money a player owns [UC13].

7.3 Subflows:

None.

7.4 Alternative Flows:

[E1] Nothing happens if the player does not have enough money for buying the cell.

[E2] The price for a railroad cell or a utility cell is fixed. Railroad cells all cost the same. So do utility cells.

UC8 Flow of Events for the *Buy House* Use Case

8.1 Preconditions:

1. It is the player's turn.
2. The player has not rolled the dice.
3. The player has monopoly on one or more color groups.

8.2 Main Flow:

When a player has all the tradable cells in a color group, this player is said to have monopoly on the color group. A player may build house(s) in the property cells in the color groups the player has monopoly on by pressing the Buy House button before he or she rolls the dice [S1] [E1 – E2]. The price of the house is determined by the cell. After buying the house(s), the status of the player is updated and displayed on the game board [UC13].

8.3 Subflows:

[S1] When the Buy House button is clicked, the Buy House dialog shows up. The player selects the monopoly color group and the number of houses from that dialog. After clicking on OK in the dialog box, the player pays the fee, and the houses are created. All the property cells in the selected color group have the same number of houses.

8.4 Alternative Flows:

[E1] Nothing happens if the player does not have enough money.

[E2] The player can build at most five houses in a cell.

UC9 Flow of Events for the *Pay Rent* Use Case

9.1 Preconditions:

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.

9.2 Main Flow:

The player pay rent to the owner of the cell. The rate of the rent depends on the type of cell the player lands on [S1 – S3] [E1].

9.3 Subflows:

[S1] The rent of a property cell is defined in the property attribute. Each cell may have different rent rate. If the cell is in the owner's monopoly color group, the rent doubles.

[S2] If the cell is a utility cell, the player rolls the dice again [UC11]. If the owner owns one utility cell, the player pays three times the dice roll; if the owner owns two utility cells, the player pays ten times the dice roll.

[S3] If the cell is a railroad cell, and the owner owns *N* railroad cells, the amount of rent the player needs to pay is \$50 * 2 $^{N\text{-}1}$.

9.4 Alternative Flows:

[E1] 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.

UC10Flow of Events for the *Draw Card* Use Case

10.1 Preconditions:

1. It is the player's turn.
2. The player has rolled the dice.
3. The player lands on a card cell.

10.2 Main Flow:

There are two types of card cells in this game: Community Chest and Chance. Each type of card cell is associated with a pile of cards. When the player lands on a card cell, he or she draws a card by clicking the Draw Card button. A card is drawn from the top of the Community Chest card pile or the Chance card pile, depending on the type of cell the player lands on. The player performs the actions described on the card [S1 – S4]. After that, the card is put back to the bottom of the card pile, and the status of the player is updated and displayed [UC13].

10.3 Subflows:

[S1] If the card says the player can collect some certain amount of money, that amount of money is given to the player.

[S2] If the card says the player loses some certain amount of money, that money is subtracted from the player [E1].

[S3] If the card says the player goes to jail, the player is sent to the Jail cell immediately.

[S4] If the card says the player goes to some cell, the player is sent to that cell immediately.

10.4 Alternative Flows:

[E1] If the player does not have enough money, he or she is bankrupt. He or she needs to give up all his / her money, and all the tradable cells he / she owns become available. The player is out of the game.

UC11Flow of Events for the *Roll Dice* Use Case

11.1 Preconditions:

It is the player's turn.

11.2 Main Flow:

The player rolls the dice by clicking on the Role Dice button. The Dice Roll dialog pops up to indicate the value of the dice roll. In this game, there are two six-faced dice.

11.3 Subflows:

None.

11.4 Alternative Flows:

None.

UC12Flow of Events for the *Switch Turn* Use Case

12.1 Preconditions:

1. It is the player’s turn.
2. The player has rolled the dice, and moved to the new cell.

12.2 Main Flow:

The player’s turn ends when he or she clicks on the End Turn button.

12.3 Subflows:

None.

12.4 Alternative Flows:

None.

UC13Flow of Events for the *View Information* Use Case

13.1 Preconditions:

None.

13.2 Main Flow:

The players can see their status, including their names, money, and properties, on the game board. The attributes of the cells, including the names, the owners, the number of houses, and the price, is displayed on the game board, too.

13.3 Subflows:

None.

13.4 Alternative Flows:

None.

UC14Flow of Events for the *Get Out of Jail* Use Case

14.1 Preconditions:

1. It is the player’s turn.
2. The player has not rolled the dice.
3. The player is in jail.

14.2 Main Flow:

Before the player can roll the dice, he or she needs to click on Get Out of Jail button. Upon clicking the button, the player pays $50, and is no longer in jail [E1].

14.3 Subflows:

None.

14.4 Alternative Flows:

[E1] If the player does not have enough money, he or she is bankrupt. He or she needs to give up all his / her money, and all the tradable cells he / she owns become available. The player is out of the game.

UC15Flow of Events for the *Trade* Use Case

15.1 Preconditions:

1. It is the player’s turn.
2. The player has not rolled the dice.

15.2 Main Flow:

The player may ask another player to sell his or her tradable cells. If the player wants to trade with another player, he or she clicks on the Trade button. The Trade Property dialog pops up and the player enters the player (the seller) he or she wishes to trade with, the cell he or she wishes to buy, and the amount of money he or she wish to pay [E1 – E2]. Then another dialog box shows up to ask the seller if the seller agrees with the deal. The seller clicks on Yes in the dialog box, and the cell is sold to the player for that amount of money [E3].

15.3 Subflows:

None.

15.4 Alternative Flows:

[E1] If the player clicks on Cancel button, the dialog closes and the deal is cancelled.

[E2] If the player does not have enough money, the deal is cancelled.

[E3] If the seller says no to this deal, the deal is cancelled.

III. Misuse Cases

UC16Flow of Events for the *Play More Than One Turn in a Round* Use Case

16.1 Preconditions:

The player has completed moving, except for clicking on the End Turn button.

16.2 Main Flow:

Instead of the End Turn button, the player clicked on the Roll Dice button so that he or she can play another turn in the same round [E1].

16.3 Sub-flows:

None.

16.4 Alternative Flows:

[E1] The Roll Dice button is disabled after the player rolls the dice. The player cannot click on it until the next turn.

IV. Nonfunctional Requirements

NR1. Performance

The system shall wait for all user inputs, and execute only the necessary functions given a user input to the system. All functions shall be completed quickly.

NR1.1. User response

The system shall respond to any user input within 0.01 seconds.
Origin:

Priority: 3

Implementation Completed Date: July 9, 2004.

NR1.2. Update user data

The system shall update user data within 0.01 seconds.

Origin:

Priority: 3

Implementation Completed Date: July 9, 2004.

NR2. Usability

A user shall be able to determine quickly what player options they have to perform.

NR2.1. Player options

A user shall only have access to functionality that is allowed to them at a given time.

Origin:

Priority: 3

Implementation Completed Date: July 9, 2004.

NR2.2. User Interface

The system shall allow a user to interface with it through mouse events on buttons and drop down boxes and keyboard events on text fields. The amount of user keyboard input shall be minimized by the system to include only entering the number of players, player names, and a trade price.

Origin:

Priority: 3

Implementation Completed Date: July 9, 2004.

NR2.3. User Errors

The system shall catch improper input from all text fields in the system.

Origin:

Priority: 3

Implementation Completed Date: July 9, 2004.

V. Constraints

All code development shall be done with the Java programming language.
All testing shall be done using JUnit and FIT.

VI. Requirements Dependency Traceability Table

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16	NR1.1	NR1.2	NR2.1	NR2.2	NR2.3
UC1																					
UC2											X	X	X								
UC3		X																			
UC4		X																			
UC5		X																			
UC6		X																			
UC7		X											X								
UC8							X						X								
UC9		X					X				X										
UC10		X											X								
UC11	X																				
UC12	X																				
UC13	X																				
UC14				X						X											
UC15							X														
UC16											X	X									
NR1.1																					
NR1.2																					
NR2.1																					
NR2.2																					
NR2.3																					

VII. Development and Target Platforms

1. Windows XP Operating System
2. Intel Pentium IV processors
3. Eclipse IDE

VIII. Project Glossary

IX. Document Revision History

Version	2.0
Name(s)	Sarah Smith
Date	August 22, 2005
Change Description	Updated UC 11 to be Roll Dice

Version	1.0
Name(s)	Dright Ho and Sarah Smith
Date	July 19, 2004.
Change Description	Original creation of the SRS.