<Demon Entertainment>

<Word-Rover>
Project Analysis and Development Plan
Version <1.2>

TEAM MEMBERS

Name	Student ID
Huihua(Angel) GUAN	**4778375**
Francois Gabriel BELLAVANCE	**4481380**
Lifeilai LICHOKCHING	**4633369**
Yo-Zen(Charles) LIU	**4626397**
Wei LI	**5028817**
Hong An ZHANG	**5048400**
Qingzhe(Nick) HUANG	**5037735**
Xiang Qun LIU	**4584414**
Qihui HU	**5047633**

Revision History

Date	Version	Description	Author
26 January 2004	1.0	Document Start	Huihua, lifeilai, Yo-Zen francois, QingZhe
01 February 2004	1.1	Added User Interface graphics	Yo-Zen
03 February 2004	1.2	Combining all parts	Huihua, lifeilai, francois, Yo-Zen, QingZhe,
04 February 2004	1.2	Finalize	lifeilai

Table of Contents

Project Analysis and Development Plan

1. Introduction

The purpose of this document is to collect, analyze, and define the high-level needs and features of Word-Rover, a multi-player game being developed by Demon Entertainment. Word-Rover is a 2 to 4 players game having human interaction opponents and it is a new game, based on the idea of merging Hasbro's Scrabble and on Rummy card game. It will describe the product functions, requirements and constraints of the game.

The details of how the Word-Rover game fulfills these needs are detailed in the use case and more details will be available in the upcoming design phase.

1.1 Purpose

This document describes the specification of the Word-Rover multi-player game, which is in partial fulfillment of the requirements of COMP 354. It will define the high-level requirements of the product functions, user descriptions, assumptions and dependencies, constraints, specific requirements and an analysis model. The analysis model will include use case diagrams, class diagrams, sequence diagrams and state transition diagrams. Furthermore, a detailed project plan will be provided, including the schedule of the upcoming phases. This document is intended for the Demon Entertainment and the project coordinator, Dr. Joey Paquet, as it will serve as a basis for the upcoming phases of the project.

1.2 Scope

This document only addresses the high level requirements of Word-Rover that will be used as a basis for the design phase. The product functions will include a prioritized list of which functions are to be included in the game. The user description will include the target audience and working environment of the game. Specification for the game will be outlined in the specific requirements section and the analysis model will contain UML diagrams. The use case diagrams will give an overview of the functions of Word-Rover and how the users will interact with the game. The class diagrams will show the relationship between different objects in the game and the sequence diagrams will model the flow of logic within the game.

The development plan will outline the three phases of the project with a Work Breakdown Structure (WBS) with a Gantt chart. It includes the number of days to complete each sub-tasks and the project. The resources required for the project are also listed in this section.

1.3 Definitions, Acronyms, and Abbreviations

1.3.1 Definitions

Game-Board
The playing area is made up of squarely 15 cells by 15 cells per side. There are three types of cells on the board. The first type is a normal cell can be played by the player. The second type is bonus cell for doubling letter value, and the third one is bonus cell for doubling word value.

Tile
An alphabet (From A to Z) is written on each Tile and also displays the value of the alphabet. There are 100 Tiles including the two blank Tiles.

Bag
All the letters are placed in a bag, which each player will draw randomly 7 letters for it.

Joker
A Joker is a group of cards (a hand) that can be traded for any letter.

Racks
Each player is assigned a rack to place and hide their Tiles. Each rack can only hold 7 Tiles only.

Bingo
Any word played that uses all seven letters of the rack, earning a bonus of 100 points.

Deck
A deck is a set of 52 cards which are distributed as follows:
4 suits: spade, heart, club and diamond
13 cards in each suit: Ace,2,3,4,5,6,7,8,9,10,J,Q,K

Suite
A Suite is a group of cards with at least 3 consecutive cards or 3 identical cards.

1.3.2 Abbreviations

IEEE: Institute of Electrical and Electronics Engineers, Inc

OS: Operating System

1.4 References

Hasbro, “All about Scrabbles”, Hasbro
http://www.hasbro.com/scrabble/pl/page.rules/dn/home.cfm (20 January 2004)

Template I, Project and Development Plan, Rational Unified Process, Rational Software.
http://newton.cs.concordia.ca/%7epaquet/teaching/354/index354W2004.html

Example of Template I, Montrealopoly, Team Redmon, Fall 2003
http://newton.cs.concordia.ca/%7epaquet/teaching/354/index354W2004.html

Pressman, Roger S. Software Engineering: A Practitioner’s Approach. 5th Ed. Toronto, McGrawHill, 2001.

1.5 Overview

The rest of this document outlines the problem description and the development plan.

The problem description outlines the product methods, user descriptions, assumptions and dependencies, constraints, specification requirements and analysis model.

The development plan describes the project time-lines and project plan.

2. Problem description

2.1 Project Purpose, Scope, and Objectives

The objective of this project is to create a new game by combining Scrabble and Rummy. The new game will be called Word-Rover. It can be played by 2-4 players. The game contains a board, a rack for each player and a scoreboard. It plays mostly by Scrabble's rules with slight variations inspired from Rummy's rules.

2.1.1 *User interfaces*

The first interface that user encounters is the "initial setting interface". It will prompt to ask user to enter the number of players and their respective names.

The "Word-Rover" game interface comes after the initial setting interface. This is the main interface on which players will be playing and interacting with each other. The game interface contains several components:

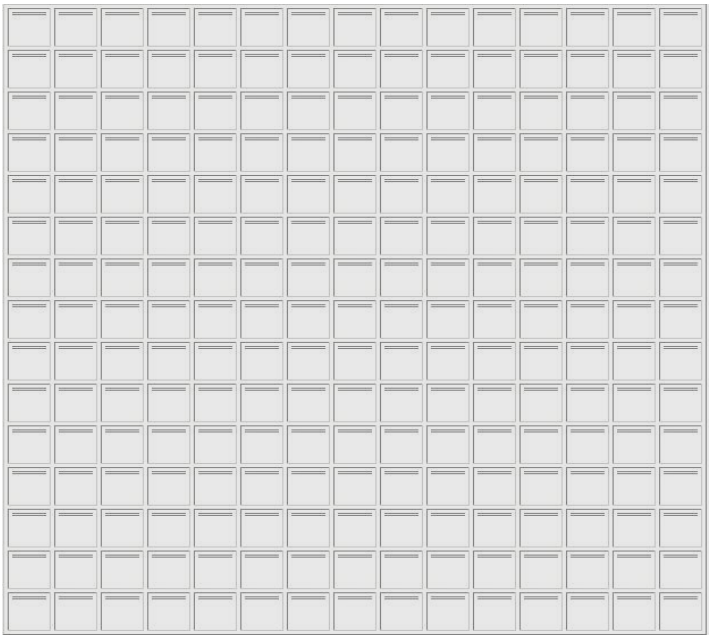
- board
- rack
- letters
- cards
- scoreboard

The winner interface concludes the game by showing the winner's name and the winning score.

2.1.1.1 Initial Setting Interface

The user is asked to enter the number of the players and their respective names before starting the game.

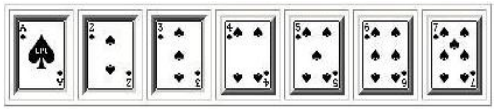
2.1.1.2 Board



The board is used to display the words formed by players. By default, it is 15 by 15 cells. However, it is also customizable upon user's request.

2.1.1.3 Rack

Each player has a rack containing 7 letters and 7 cards.



2.1.1.4 Letters

There are 100 letters in the game and they are distributed as follows:

2 "blank" tiles (scoring 0 points), 9 As (1 point), 2 Bs (3), 2 Cs (3), 4 Ds (2), 12 Es (1), 2 Fs (4), 3 Gs (2), 2 Hs (4), 9 Is (1), 1 J (8), 1 K (5), 4 Ls (1), 2 Ms (3), 6 Ns (1), 8 Os (1), 2 Ps (3), 1 Q (10), 6 Rs (1), 4 Ss (1), 6 Ts (1), 4 Us (1), 2 Vs (4), 2 Ws (4), 1 X (8), 2 Ys (4), and 1 Z (10).

2.1.1.5 Cards

There are 52 cards in the game and they are distributed as follows:

4 suits: spade, heart, club and diamond
13 cards in each suit: Ace,2,3,4,5,6,7,8,9,10,J,Q,K

2.1.1.6 Scoreboard

Snoopy	Gardfield
0	0
0	0
0	0
0	0
0	0
0	0
0	0
Total	Total
0	0

All players will share the same scoreboard.
The scoreboard is used to display the scores in the current turn, the accumulated score from the previous turns and the name of the winner with the highest score at the end of the game.

2.1.1.7 Winner Interface

Congratulations!

Snoopy won the game with a winning score of 300 points!

Click here to restart the game

The winner interface shows the winner's name and the winning score.

2.2 Product Functions

2.2.1 Initial Setting Interface

Input:
- The number of players
- the name of each player

Action:
- An instance of each player is created

Output:
- Confirm the number and the names of players

Validity Check:
- At least 2 players are required to play the game.

2.2.2 Rack

Input:
- Player selects letters to form a word
- Player could also select cards to trade for letters
- When the letters are ready to be posted on the board, the player chooses where (give the coordination) and how (choose from vertical or horizontal) the word will be posted
- Player could discard any number of letters and cards each turn

Action:
- The selected letters are pressed down
- The selected cards are pressed down

Output:
- The selected letters will be displayed as a word on the board by the order they were selected
- The traded letters from the selected cards will be displayed next to the letters rack and shown as pressed down as it is selected by the player
-

Validity Check:
- In the beginning of player's turn, check the number of letters/cards on the player's rack; if any of them is less than 7, distribute new letters/cards
- Check if the selected cards are valid for trade and let user pick a letter

2.2.3 Board

Input:
- Receive the formed word from the rack

Action:
- Ask the other player if the word can be allowed
-

Output:
- Post the word on the board as the user requested (the coordination and the direction of the word on the board)
-

Validity Check:
- Make sure the word to be posted on the board won't exceed the size of the board
-

2.2.4 Scoreboard

Input:
- Receive the posted word from the board
-

Action:
- Calculate the scored points by adding up all the preset letter values, including those traded letters that has the value of the original selected cards.
-

Output:
- Post the score points on the scoreboard
-

Validity Check:
- Check the number of rounds; at the end of round 10, stop the game and pass the winner parameters (winner's name, winner's score) to the winner interface

2.2.5 Winner interface

Input:
- Receive the winner parameters from the scoreboard

Output:
- Congratulate the winner by displaying the winner's name and the winner's score
- Ask if players would like to

2.3 User Description

2.3.1 User Environment

Game Environment
In this section, we will analysis the game tasks of each player in every turn. There are two kinds of tasks, which are immediate tasks and non-immediate tasks. The definition, characteristics, example and explanation of these two different kinds of task will be explained in the following table.

	Immediate task	**Non-immediate task**
Definition	Immediate task is defined as a task that will have immediate feedback right after they are executed.	Non-immediate task is defined as a task whose executing time will be dependent on the player.
Characteristics	• Single player task • Immediate feedback	• Single player task • Non decidable feedback time
Example and Explanation	• Pick up token: Feedback will be given at once whenever a player performs this task. • Discard token: Same as above	• Initialization of game: Each player choose a name adding to the playing list, and is given 14 tokens, each 7 are put on the letter and card rack. • Making a joker: If a player has one or more jokers, he will decide if he needs to use his joker in forming a word. • Forming a word: A player has two ways to form a word, which are either not using a joker or using one or at most 2 jokers.

Users System Environment
The game will run on a PC with Microsoft Windows 9X, 2000 and Xp.

Users Hardware Environment
See the topic 2.6.2 System Requirements

2.3.2 User Profiles

People whose ages are 18 or over can play Word-Rover. In order to play Word-Rover, knowledge of using Microsoft Windows is required.

2.4 Assumptions and Dependencies

The Assumptions of Word-Rover:
(a) Windows Operating Systems are required: Windows 2000, Me or Xp.
(b) The User must know basic computer knowledge.
(c) The Bag in the game has a limited number of words.
(d) The game supports only 2 to 4 players at the same time.

2.5 Constraints

The Word-Rover game is written in the standard C++ and the GUI is written in ASP.NET and HTML.

2.6 Specific Requirements

2.6.1 Applicable Standards

Standard forms of variables are applied:

(a) Attributes names: non-capitalized form of word is used. For example: player
(b) Methods names: First word is capitalized. For example: Player()
(c) Class Names: First word is capitalized.

2.6.2 System Requirements

Hardware Requirements

(a) PC should at least PIII 450MHz Processors
(b) 64MB RAM
(c) Mouse
(d) Video card and Sound Card
(e) Free disk space 500MB is recommended
(f) No internet connection required

2.6.3 Performance Requirements

Reliability: The Word-Rover game is error free, and proper exception-handling methods have been used to handle all kinds of errors.

Accuracy: The Word-Rover game will calculate accurately each player total scores after each round.

2.6.4 Environmental Requirements

Maintainability: The Word-Rover is designed in an incremental form, and Developers can easily update and add new method modules.

Usage Conditions: The Word-Rover game can be installed on all Windows Operating Systems.

Error Handling and Recovery: Proper exception-handling methods have been used to handle errors and each error has a reference number associated with it, which can be checked in the read-me file.

2.7 Analysis model

2.7.1 Use Case Diagram

A Use Case diagram is useful in the analysis phase of software development; it helps communicate the high-level requirements of the system. A Use Case diagram identifies the entities that interact with the system “actors”, the functionalities provided by the system “use cases” and the association between the actors and the use cases. The following Use Case diagram Demon Entertainmentstrates the actors and use cases identified in the system. The use cases “Play Turn” and “End Turn” have been expanded to provide a more detailed high-level representation of the system.

2.7.1.1 The Player Use Case

2.7.2 Use Case Details

2.7.2.1 Use Case 1: Select Game Options

Description	The player changes the game settings (Options).
Actors	Player
Pre-Conditions	The game has not been started.
Flow of Events	
Basic Path	1. The player opens the game options window. 2. The player can change the configuration of the game rules. 3. The player can change the configuration of the game board. 4. The player can change the configuration of the tiles (value and qty). 5. The player can change the configuration of the deck (value and qty).
Alternate Paths	
Post-Conditions	> The game setting (options) changed by the player have been updated.
Related Use Cases	
Include Use Cases	None
Extending Use Cases	None

2.7.2.2 Use Case 2: Start Game

Description	The player starts a game and enters the parameters of the game.
Actors	Player
Pre-Conditions	None
Flow of Events	
Basic Path	1. The player selects the option “Start Game" from the game menu. 2. The player enters the number of players in the game. 3. The player's name is entered for each player. 4. The board is initialized according to the game settings. 5. The tiles are initialized and shuffled according to the game settings. 6. 7 Tiles are assigned to each player rack. 7. The deck of cards is initialized and shuffled according to the game settings. 8. 7 cards are assigned to each players rack. 9. The players' order is randomized. 10. The first player is ready to start.
Alternate Paths	None
Post-Conditions	
Related Use Cases	
Include Use Cases	None
Extending Use Cases	None

2.7.2.3 Use Case 3: Play Turn

Description	It is the player's turn to play.
Actors	Player
Pre-Conditions	It's the players turn to play.
Flow of Events	
Basic Path	1. The player draws some tiles, the tiles picked up are placed on the player's rack. 2. The player draws some cards, the cards picked up are placed on the player's rack. 3. Using the tiles (letters) and cards (suites and groups = Joker) the player can try to form a word and place the tiles of the word on the board. 4. The player ends his turn.
Alternate Paths	Alternative 1: > If the player is not able to form a word step 3 is skipped.
Post-Conditions	> A word has been placed on the board. (If applicable)
Related Use Cases	
Include Use Cases	Draw Tiles, Draw Cards, Form Word
Extending Use Cases	None

2.7.2.4 Use Case 4: End Turn

Description	The player ends his turn.
Actors	Player
Pre-Conditions	It is the player's turn to play.
Flow of Events	
Basic Path	1. If the current player has formed a word, the other players are prompted to challenge the word. 2. If the word is refused the challenger gets some bonus points and the word is removed from the board. 3. If the word is accepted, the player gets some points and the tiles used are removed from the player's rack. 4. The player discards the tiles he does not want in his rack. 5. The player discards the cards he does not want in his rack. 6. Next player's turn.
Alternate Paths	Alternative 1: > The player did not form a word, hence steps 1,2 and 3 are skipped.
Post-Conditions	> The word is accepted. (If applicable) > Next player's turn to play.
Related Use Cases	
Include Use Cases	Discard Tiles, Discard Cards, Challenge Word
Extending Use Cases	None

2.7.2.5 Use Case 5: Display Help

Description	Displays a help window to the player.
Actors	Player
Pre-Conditions	None
Flow of Events	
Basic Path	1. The player selects the "Help" option in the game menu. 2. A help window is displayed on the screen.
Alternate Paths	
Post-Conditions	>The Help window has been displayed.
Related Use Cases	
Include Use Cases	None
Extending Use Cases	None

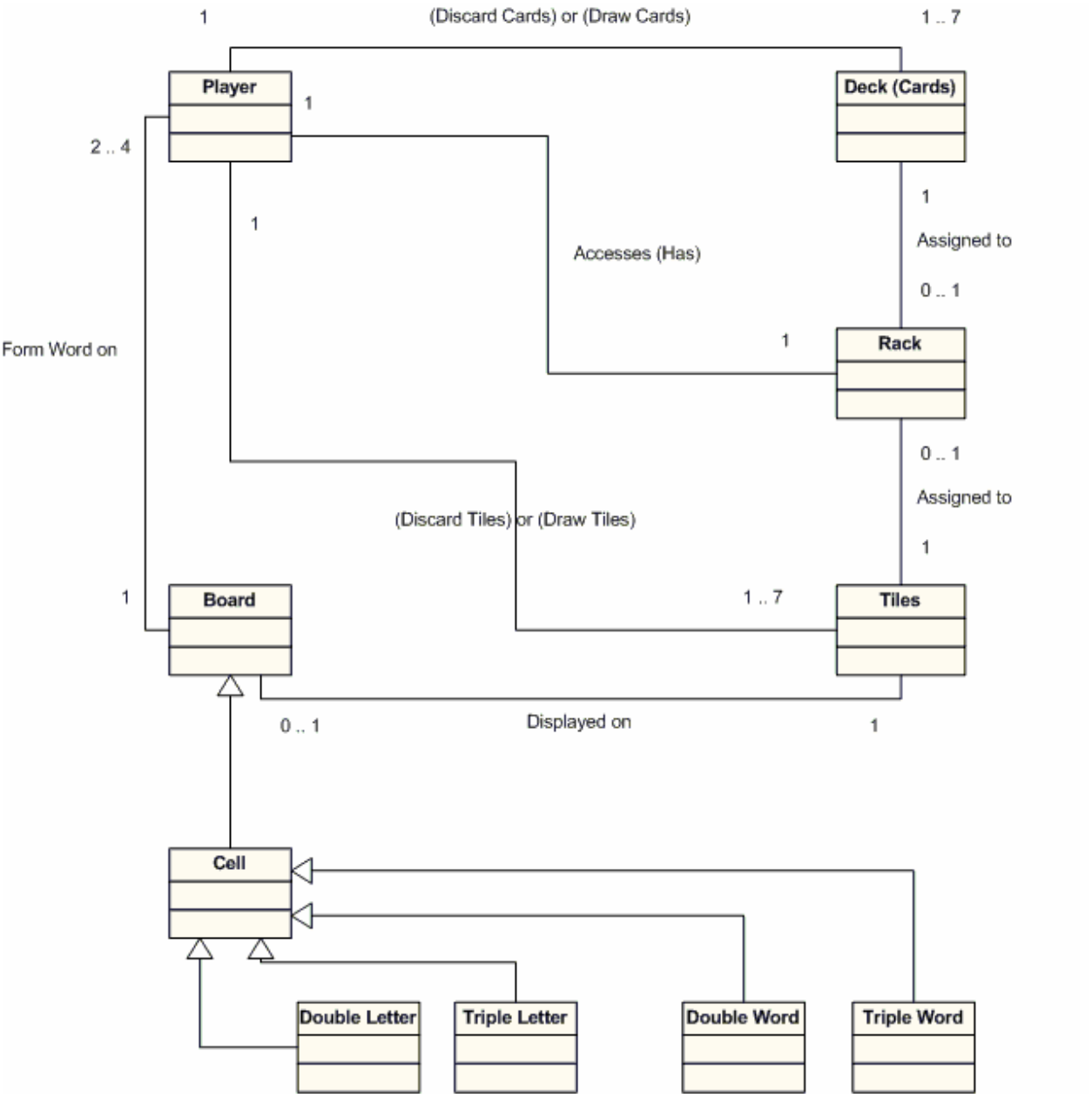
2.7.2.6 Use Case 6: End Game

Description	Player prematurely ends the game before the game is over.
Actors	Player
Pre-Conditions	None
Flow of Events	
Basic Path	1. The player selects the "End Game" Option in the game menu. 2. A dialog box prompts the player to confirm the selection. (Ok/Cancel) 3. The player confirms that he wants to end the game. 4. The total points are assessed for each player. 5. The player(s) with the most points are declared the winner(s).
Alternate Paths	Alternative 1: > In step 2, the player selects cancel, hence the operation of ending the game is cancelled. Steps 4 and 5 are not executed.
Post-Conditions	> The game is over. > The winner(s) has been declared.
Related Use Cases	
Include Use Cases	None
Extending Use Cases	None

2.7.3 Class Diagram

A Class diagram describes the static structure of a system. A Class diagram documents the different classes, interfaces and relationships identified during the analysis. The following Class diagram Demon Entertainmentstrates the problem domain entities. Hence, the following Class diagram is not a design-level diagram consequently the class attributes and functions have been omitted.

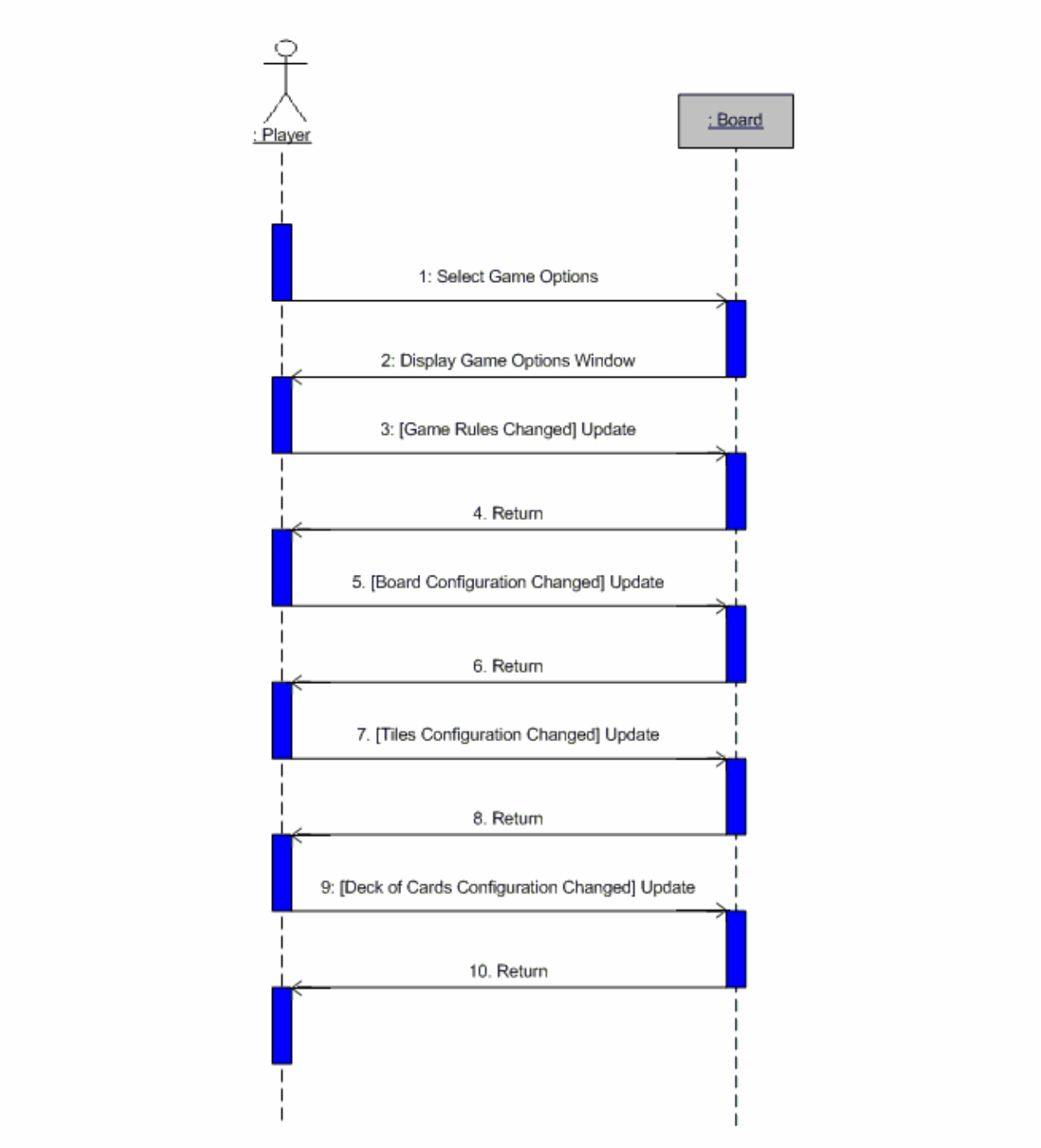
2.7.3.1 Full Class Diagram



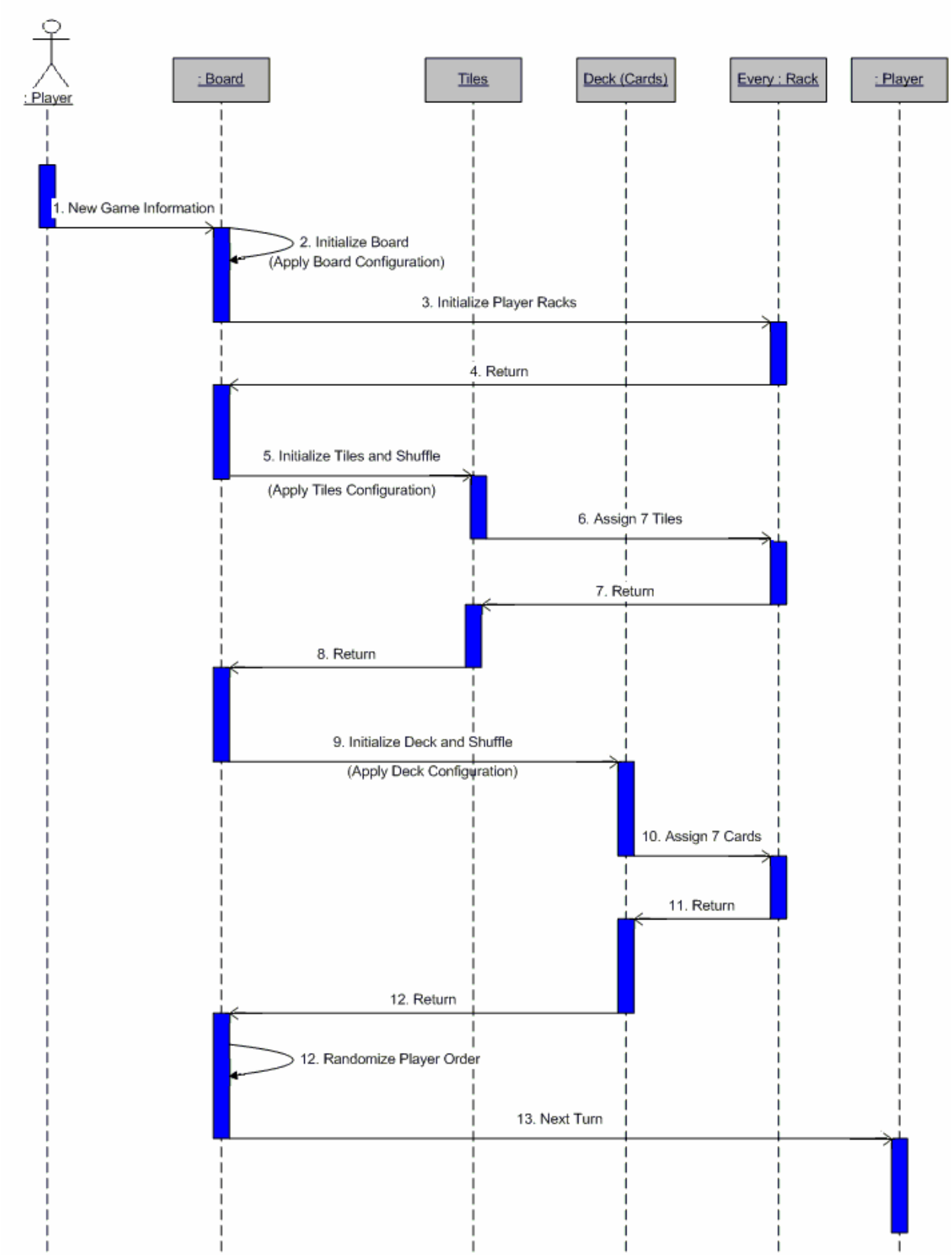
2.7.4 Sequence Diagrams

A Sequence diagram is a useful tool for Demon Entertainmentstrating the dynamic behavior of a system. A Sequence diagram is helpful for documenting use case scenarios, more specifically the interaction between the classes and the sequence of events. The following Sequence diagrams represent the six use case scenarios identified in the 'Player' Use Case diagram in section 2.7.1.1.

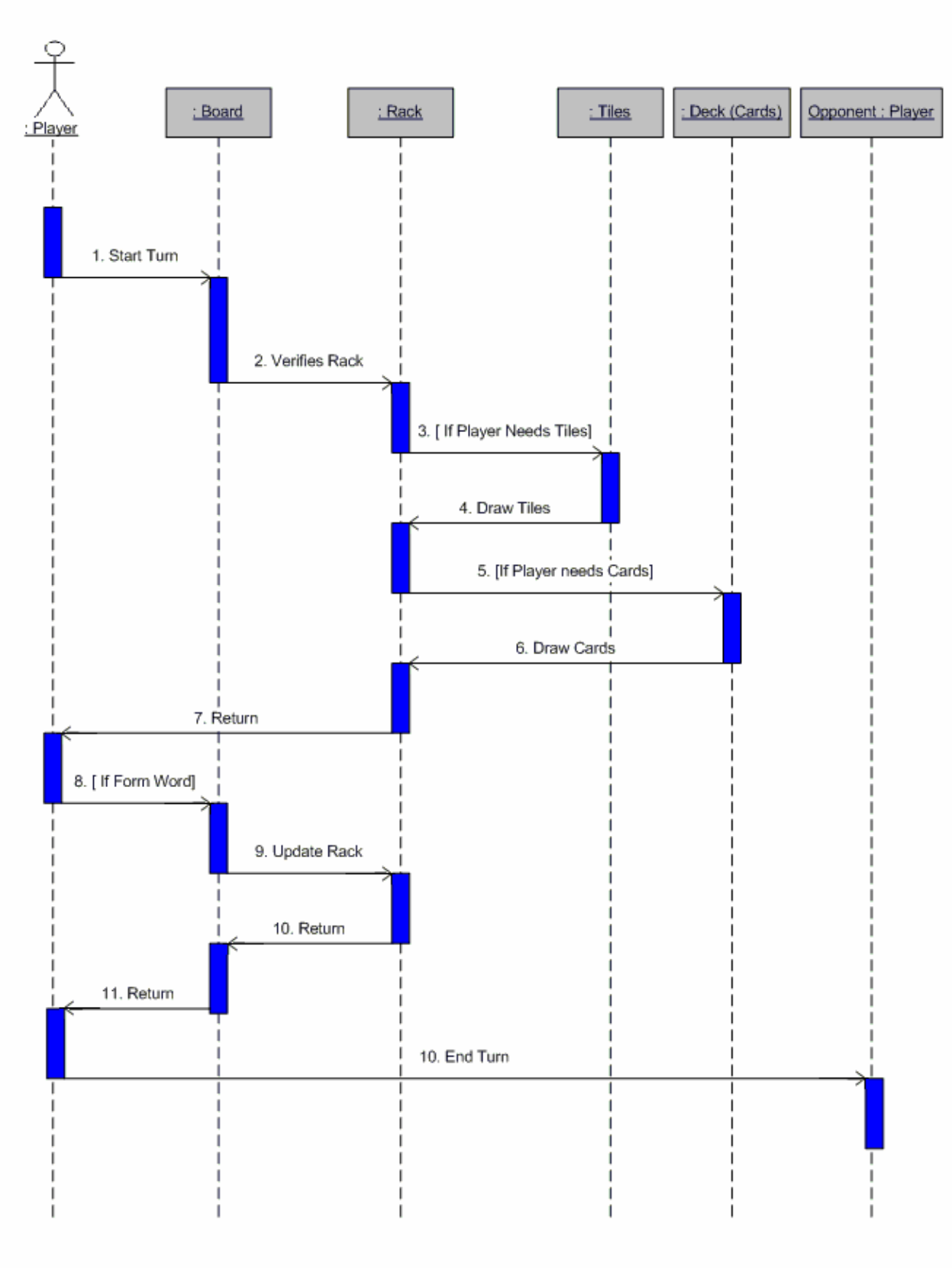
2.7.4.1 Select Game Options (Use Case 1)



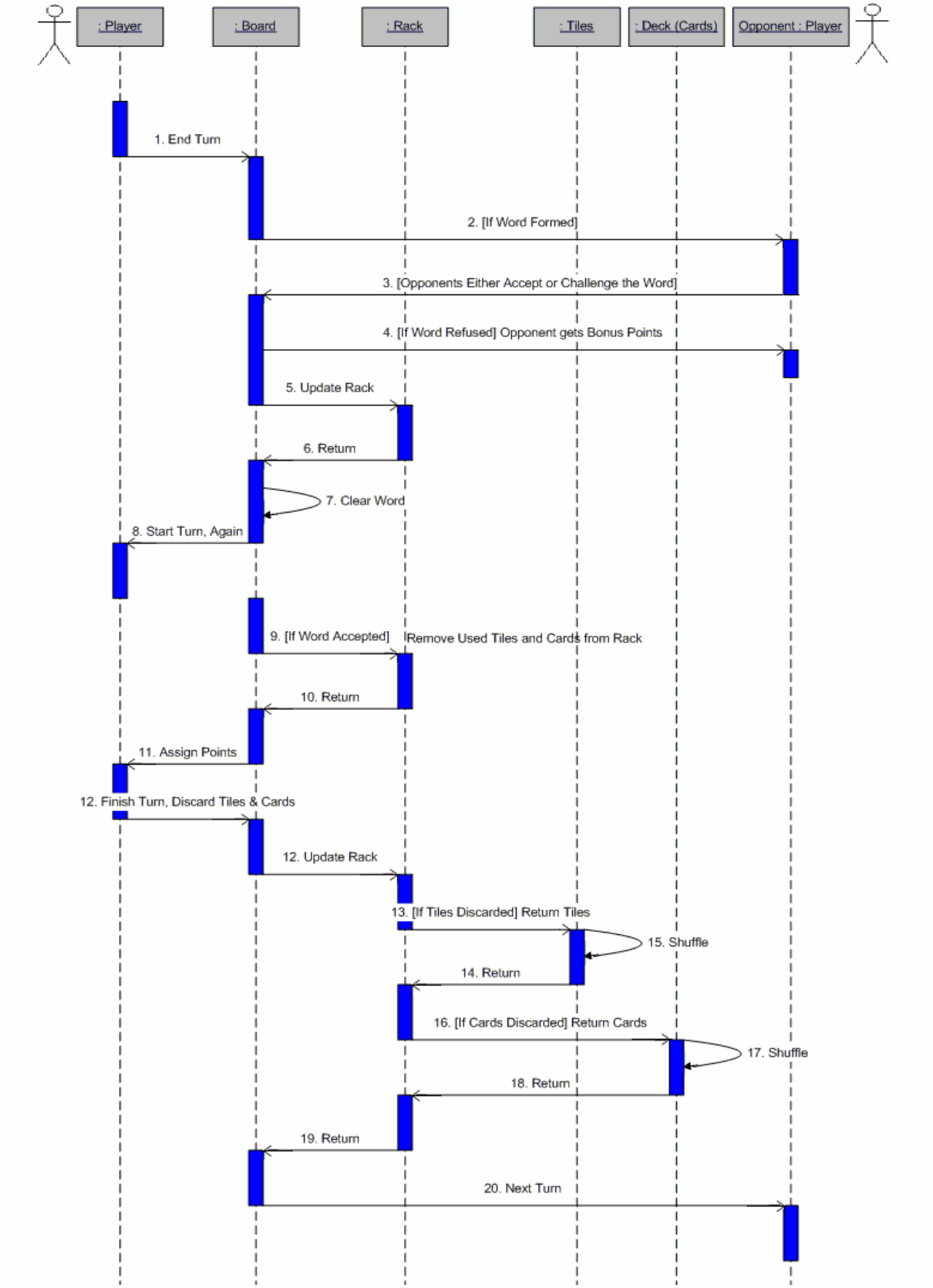
2.7.4.2 Start Game (Use Case 2)



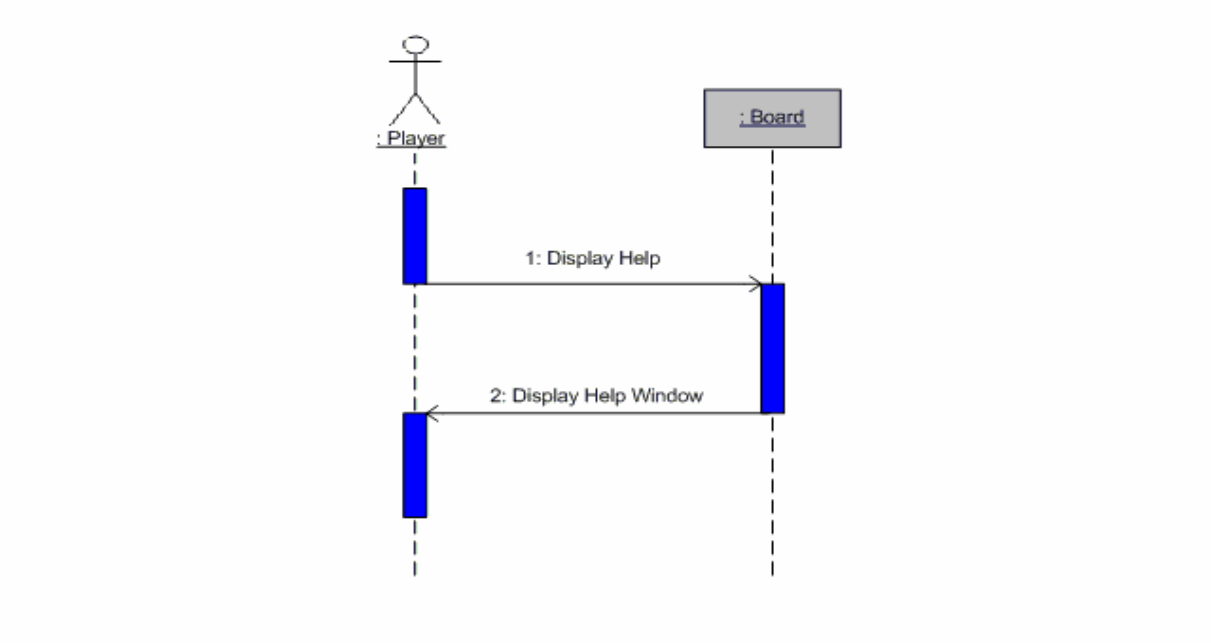
2.7.4.3 Play Turn (Use Case 3)



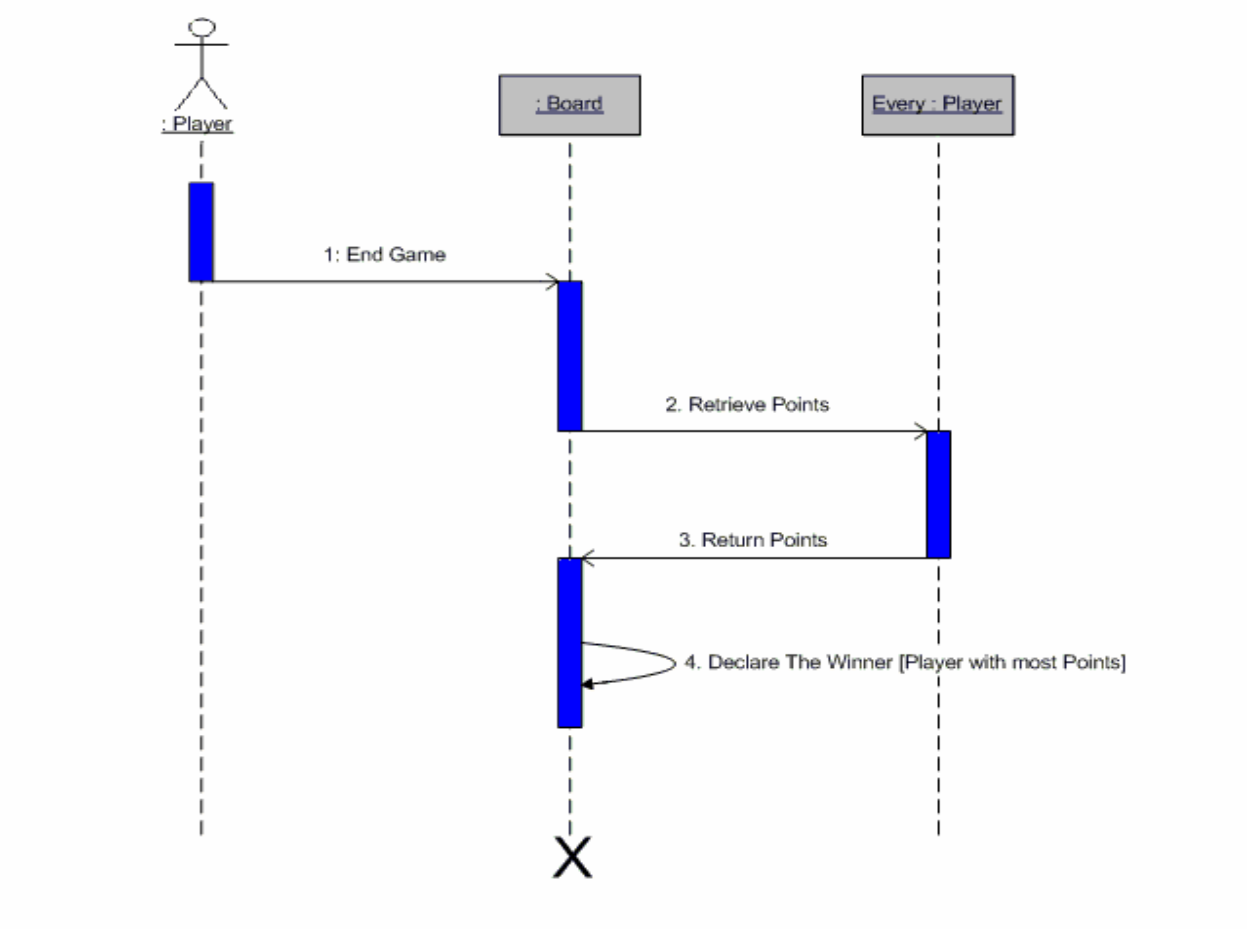
2.7.4.4 End Turn (Use Case 4)



2.7.4.5 Display Help (Use Case 5)



2.7.4.6 End Game (Use Case 6)



3 Development Plan

This section provides the estimated cost and schedule for the project. WBS with a Gantt chart indicates major milestones, their achievement criteria and the exact time of completing them. In addition, human resource is also discussed in this section.

3.1 Project Estimates

This first phase, requirement, took 4 and half weeks to complete in which the average work time per week is 30 hours. As the project schedule indicates, it was finished in the middle of week 5. This amount of time is cumulated by the whole group.

The second phase, design, will take two and half week to finish. This phase includes two parts, which are prototype1 and prototype2. We will do prototype1 from the middle of week5 to week 6, which we estimate to be 25 hours. Then we will revise and optimize the prototype1 into prototype2, which it costs another 16 hours and is completed in week7.

Implementation is the third phase which will start right after first prototype1 is completed. This phase will take six weeks which is from week7 to week12 to complete. The total time is estimated to be 80 hours. As the game's interface, we will prefer to make text interface first. In the middle of implementation phase, we will decide if we can optimize our game' interface into graphic interface according the whole workload of coding and time limit. In the last week of implementation we will test the game and make it optimal.

An overview of each phase is shown in the following project schedule:

Phase \ Week		Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12
1	Requirement												
2	Design												
	Prototype1												
	Prototype2												
3	Implementation												
	Visuals												
	Test												

Project Schedule

3.2 Project Plan

The project plan is made up of a phase plan and a project schedule. The phase plan contains a Work Breakdown Structure Gantt chart, which presents the total number of working hours for each task. It also contains the minor and major milestones in the project.

3.1.1 Phase Plan

Task	Duration(hrs)			Jan								Feb																	Mar											Apr	
				w1	w2		w3	w4				w5						w6				w7				w8			w9			w10	w11			w12			w13	
	# of hrs	# of ppl	total hrs	8	13	15	22	27	29	30	31	3	4	5	6	7	8	10	12	13	14	17	19	20	21	24	26	27	2	4	5	9	11	16	18	19	23	25	26	1	
Phase 1																																									
Requirements																																									
Group Formation/Leader Designation/Team Formation	1	9	9	1																																					
Games Info Collection	1	4	4		1																																				
Analysis Game Info/Define Games Scope	1	9	9			1																																			
Preliminary Design Game Rules/Game	5	9	45			1	4																																		
*Agreement on Game Rules			0				@																																		
*Agreement on Programming Language (C++)			0				@																																		
Analysis/Game Requirements	2	3	6					2																																	
Draft Preliminary Game Specifications and	14.5	4	58						3	5	7																														
Compiling the Project Analysis and Development	1	4	4									1																													
Finalize the Project Analysis and Development Plan	1	9	9										1																												
!Project Deliverable 1 Completed														@																											
Phase 2																																									
Design																																									
Prototype 1																																									
Review preliminary game specifications	6	4	24												2	1	1	2																							
Develop functional specifications	17	4	68													4	4	3	2	2	2																				
Develop prototype based on functional specifications	17	4	68													4	3	2	2	3	3																				
Develop Initial Setting Interface	6	2	12												3		2		1																						
Develop Game Interface: Board, Rack, Letters, Cards,	2	2	4															1	1																						
Develop Winner Interface	3	2	6																	2	1																				
*Completion of individual interfaces																					@																				
Prototype 2																																									
Assemble all the interfaces together	2	5	10																			1	1																		
Review functional specifications	3	3	9																				1	1	1																
Incorporate feedback into functional specifications	2	4	8																					1	1																
*Design complete																																									
!Project Deliverable 2 Completed																														@											
Phase 3:																																									
Implementation																																									
Review functional specifications	12	5	60																							2	2	2		2					2		2				
Identify modular/tiered design parameters	24	3	72																							3	3	3			3		3	3		3		3			
Assign development staff	4	1	4																								1			1			1		1						
Develop code	27	5	135																									3	3	3	3	3	2	2	2	2	2	2			
Developer testing (primary debugging)	11	5	55																												2	2	1	1	1	1	1	1	1		
*Development complete																																							@		
Visuals																																									
Integrate functions to GUI	8	2	16																													1	1	1	1	1	1	1	1		
*Combination of GUI with functions																																									
Test																																									
Develop unit test plans using product specifications	4	4	16																																		3	1			
Develop integration test plans using product specifications	3	4	12																																				3		
*Completion of testings			0																																				@		
!Project Deliverable 3 Completed			0																																					@	
Total hours spent on the entire project	177.5		723																																						

3.2.1.1 Legend

Milestones		Description
Minor (*)	Agreement on Game Rules	A set of rules was defined about how to merge Scrabble with Rummy.
	Agreement on Programming Language (C++)	The game will be programmed with C++.
	Completion of individual interfaces	The game's component appearance takes form.
	Design complete	The entire game's appearance takes form.
	Development complete	Coding is completed.
	Combination of GUI with functions	The game's interfaces can now interact with each other.
	Completion of Testing	Bugs will be discovered and fixed by using several test cases.
Major (!)	Project Deliverable 1 Completed	Collect, analyze, and define high-level needs and features of Word-Rover.
	Project Deliverable 2 Completed	Design prototypes displaying the game interfaces that interact with players and develop functional specifications.
	Project Deliverable 3 Completed	The interfaces will be connected with the logical functions. After the testing and debugging, the final product is delivered to the customer.

3.1.2 Project Schedule

The following table shows the completion dates for the project phases.

Task Description	Completion Date
Phase 1 Deliverable	Feb 5
Milestone: Individual Interface Prototype I	Feb 14
Milestone: Design Completed	Mar 2
Phase 2 Deliverable	Mar 4
Milestone: Development Completed	Mar 26
Milestone: Completion of Testing	Mar 26
Phase 3 Deliverable	Apr 1

3.1.3 Project Resourcing

A group of 9 members divided into 3 teams to complete the project. Team 1-the requirements team, Team 2-the design team, and the team 3-the implementation team, each team is composed of 3 members. Each team is responsible for one phase of the project and will participate in the other phases as well.

Team composition

Member Name	Team
^Huihua (Angel) GUAN	Requirements
Francois Gabriel BELLAVANCE	Requirements
Lifeilai LICHOKCHING	Requirements
^Yo-Zen (Charles) LIU	Design
Wei LI	Design
Hong An ZHANG	Design
*^Qingzhe (Nick) HUANG	Implementation
Xiang Qun LIU	Implementation
Qihui HU	Implementation

*Group Leader
^Team Leader

Since several members of our group are familiar with C++, we have chosen it to implement Word-Rover game.

4 Team Members Log Sheets

4.1 Team Member Name: HuiHua (Angel) GUAN

Date	Task	duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
20-Jan	learn about game rules	3.00
22-Jan	meeting - contribute ideas to the new game	4.00
29-Jan	meeting - assign tasks	2.50
31-Jan	working on Documentation (2.3)	2.00
03-Feb	meeting – draft	1.00
	Total	14.17

4.2 Team Member Name: Yo-Zen (Charles) LIU

Date	Task	duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
24-Jan	working on GUI interface	2.50
25-Jan	working on GUI interface	3.00
29-Jan	meeting - assign tasks	2.00
31-Jan	working on Documentation (2.1,2.2)	4.00
	Total :	19.17

4.3 Team Member Name: Lifeilai LICHOKCHING

Date	Task	Duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
29-Jan	meeting - assign tasks	2.50
31-Jan	Working on Documentation(1.0, 2.4-2.6, 3.0)	3.00
03-Feb	meeting – draft	1.00
04-Feb	Compiling and Drafting Deliverable 1	2.00
	Total	16.67

4.4 Team Member Name: Wei LI

Date	Task	Duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
18-Jan	Learning the template for project	1.00
20-Jan	Design the my version of rules of the game	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
27-Jan	Reading game rules and preparing the suggestion	0.50
28-Jan	Considering the design of the project	1.00
29-Jan	meeting - assign tasks	2.50
01-Feb	Studying draft requirement phase document	1.00
02-Feb	Considering the design of the project	1.00
	Total	13.67

4.5 Team Member Name: QiHui HU

Date	Task	Duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
28-Jan	read game rules	0.50
29-Jan	meeting - assign tasks	2.50
01-Feb	Studying draft requirement phase document	1.00
02-Feb	Considering the design of the project and study draft	2.00
03-Feb	meeting – draft	1.00
	Total :	13.67

4.6 Team Member Name: Hong An ZHANG

Date	Task	Duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
21-Jan	project design	1.50
22-Jan	meeting - contribute ideas to the new game	4.00
27-Jan	reading game rules	0.50
29-Jan	meeting - assign tasks	2.50
02-Feb	read draft requirement	1.00
03-Feb	meeting – draft	1.00
	Total	12.17

4.7 Team Member Name: Xiang Qun LIU

Date	Task	Duration
8-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
27-Jan	Reading game rules	0.50
29-Jan	meeting - assign tasks	2.50
03-Feb	meeting – draft	1.00
	Total :	9.67

4.8 Team Member Name: QingZhe (Nick) HUANG

Date	Task	Duration
08-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
22-Jan	meeting - contribute ideas to the new game	4.00
23-Jan	Setting up coding standard for VC++	0.50
23-Jan	Coding for Dictionary class	6.00
29-Jan	meeting - assign tasks	2.50
29-Jan	Coding for Layout class	2.00
03-Feb	meeting – draft	1.00
	Total :	17.67

4.9 Team Member Name: François Gabriel BELLAVANCE

Date	Task	Duration
8-Jan	meeting - group formation	0.67
15-Jan	meeting - about scrabble and rummy	1.00
16-Jan	Research and learn Visio.	3.00
17-Jan	Researched and learned the rules of Scrabble	3.50
18-Jan	Researched and learned the rules of Rummy.	2.50
19-Jan	Documented ideas and suggestions for the game.	2.50
21-Jan	Analyzed the template of the first deliverable.	1.50
22-Jan	meeting - contribute ideas to the new game	4.00
25-Jan	Documented the framework of our game.	2.00
26-Jan	Researched the topic of UML.	1.00
28-Jan	Researched the topic of UML.	2.50
29-Jan	Group meeting - discuss and finalize game rules.	2.50
01-Feb	Development of UML Diagrams.	8.00
02-Feb	Development of UML Diagrams.	7.00
03-Feb	meeting – draft	1.00
03-Feb	Applying correction to Deliverable 1.	0.50
	Total :	43.17