
Software Requirements Specification

for

Mario

Version <3.0>

Prepared by

Group Name: Five Stars

Jadecrystal Tang Ming Mei
Lee Fui Yee
Nuratiqha Binti Abd Razak
Tan Sheu Yeu
Vernon Chien

36309
36618
38098
39049
39233

jade92crystal@live.com
fannyken@live.com
nuratiqharazumi@gmail.com
sheueyu@gmail.com
vernonchienchien@gmail.com

Instructor: Madam Nurfaeza Binti Jali

Course: TMP 3413 Software Engineering Lab

Lab Section: Friday (2:30 PM – 4:30 PM)

Date: 1st December 2014

Date: 1st December 2014

Contents

LIST OF TABLES, DIAGRAMS AND FIGURES.....	III
REVISIONS.....	IV
1 INTRODUCTION	1
1.1 DOCUMENT PURPOSE	1
1.2 PRODUCT SCOPE.....	1
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW	1
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	2
1.5 DOCUMENT CONVENTIONS	2-3
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	3
2 OVERALL DESCRIPTION	4
2.1 PRODUCT PERSPECTIVE	4
2.2 PRODUCT FUNCTIONALITY	4
2.3 USERS AND CHARACTERISTICS	5
2.4 OPERATING ENVIRONMENT	5
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	6
2.6 USER DOCUMENTATION	6
2.7 ASSUMPTIONS AND DEPENDENCIES	6
3 SPECIFIC REQUIREMENTS	7
3.1 EXTERNAL INTERFACE REQUIREMENTS.....	7-12
3.2 FUNCTIONAL REQUIREMENTS	13
3.3 BEHAVIOUR REQUIREMENTS.....	14-28
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	29
4.1 PERFORMANCE REQUIREMENTS	29
4.2 SAFETY AND SECURITY REQUIREMENTS	29
4.3 SOFTWARE QUALITY ATTRIBUTES.....	29-30
5 OTHER REQUIREMENTS	31
APPENDIX A – DATA DICTIONARY	32-39
APPENDIX B - GROUP LOG.....	40-43

List of Tables, Diagrams and Figures

Category	Number	Title	Page
Table	1.0	Definitions in Mario Game	2
Table	2.0	Abbreviations in Mario Game	2
Table	3.0	Users and Characteristics	5
Table	4.0	Use Case 1: Get Player's Name	15
Table	5.0	Use Case 2: Choose Character	15
Table	6.0	Use Case 3: Play Game	16
Table	7.0	Use Case 4: Mini Game	16
Table	8.0	Use Case 5: Game Over/ Replay	17
Table	9.0	Use Case 6: Reading Instruction	17
Diagram	1.0	Mario Game Flow	4
Diagram	2.0	Use Case Diagram	14
Diagram	3.0	Naming Character	18
Diagram	4.0	Character Selection	19
Diagram	5.0	Reading Instruction	20
Diagram	6.0	Success in Level Game and Mini Game	21
Diagram	7.0	Success in Level Game and Fail in Mini Game (Enough Coins)	22
Diagram	8.0	Success in Level Game and Fail in Mini Game (Not Enough Coins)	23
Diagram	9.0	Fail in Level Game (Have Life)	24
Diagram	10.0	Fail in Level Game (No Life)	25
Diagram	11.0	Mario Game State Diagram	26
Diagram	12.0	Mario Game Activity Diagram	27
Figure	1.0	Mario Game Sample UI01	7
Figure	2.0	Mario Game Sample UI02	8

Figure	3.0	Mario Game Sample UI03	9
Figure	4.0	Mario Game Sample UI04	10
Figure	5.0	Mario Game Sample UI05	11
Figure	6.0	Mario Game Sample UI06	12

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Five Stars	Proposal for Mario Game Application that described on its objectives, scope and expected outcome.	1 st October 2014
2.0	Five Stars	Proposal for Mario Game Application that described on its objectives, scope, expected outcome and project planning.	8 th October 2014
1.0	Five Stars	Software Requirement Specification (SRS) for Mario Game Application that described on the introduction of game application, overall description process, specification requirement either functional or non-functional and other requirements that use in this application.	12 nd November 2014

1 Introduction

This document provides a complete example of a Software Requirements Specification (SRS) document for a small game application (namely Mario) that is meant to improve the popularity and quality of current version of Mario. In the following sections, we specify the purpose of this document, the scope of the end-product, list of abbreviation, its intended audience and all sources used in the production of this document.

1.1 Document Purpose

The product whose software requirements are specified in this document is a small game application (namely Mario) that is meant to improve the popularity and quality of traditional version of Mario. The purpose of this document is to specify and establish the functional and non-functional requirements associated with the Mario game version 2.0. Mario game version 2.0 SRS will provide information for any other documents to be developed for this project in the future. Moreover, it will also provide information for the future software verification and testing by specifying the behaviour requirements of the game in the form of a Use Case Diagram, Class Diagram, Sequence Diagram and State Machines.

1.2 Product Scope

Mario game is a computer based application which is intended to provide different version of Mario game that suitable for everyone regardless of age. Mario game consists of three levels. It allows player to choose a character and name the chosen character before the game is started. Then, player will continue with the level game which is same as the traditional Mario game. Once the player completed the level game, the application will give another task for player before the player can proceed to the next level.

Mario game application will be the game that can improve the popularity and quality of traditional version of Mario game. Furthermore, Mario game will provide some educational purpose to the player. In addition, this Mario game is also suitable for everyone regardless of age. On the other hand, Mario game can also save valuable childhood memory.

1.3 Intended Audience and Document Overview

The SRS document is the result of interview between the game application architect and the client and represents the mutual agreement between these two parties to bind a contract. Therefore, this SRS document should be carefully read and approved by the client, the development team and the Quality Assurance (QA) team. Section 2 of this document provides an overall description of the product by specifying product perspective, high level functionality of the product and other general product related requirements. Section 3 describes the specific requirements of the software including: External Interfaces, Functional Requirements and Behavioural Requirements in the form of use case diagram, class diagram, sequence diagram and state diagram. This section will be the most interest to the project architects, developers and the QA team. Section 4 describes the non-functional requirements of the product and focuses on the different software attributes such as performance, security, safety, usability and reliability.

1.4 Definitions, Acronyms and Abbreviations

Word	Definition
Character Naming System	Function of the game which allow player to name the character that has been selected.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.

Table 1.0: Definitions

Term	Description
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
QA	Quality Assurance
SRS	System Requirements Specification
UI	User Interface

Table 2.0: Abbreviations

1.5 Document Conventions

1.5.1 Formatting Conventions

Several formatting conventions have been followed throughout the entire document:

1. All text contained in this document is 11pt Arial font.
2. Section titles are 18pt Arial font, bold.
3. Subsection titles are 14pt Arial font, bold.
4. Any further subsection breakdown is 12pt Arial font, bold.
5. All sections and subsection are numbered using the X.X.X...format, where X represents numbers.

1.5.2 Naming Conventions

The naming conventions are the means of making the SRS more understandable and easier to follow. They are also used to build a structure for the whole game application. The conventions are used for variables, function names, packages, etc.

The following naming conventions have been used to define the different variables in this SRS document:

1. All constants are CAPITALIZED.
2. All variables representing input are prefixed with i_.
3. All variables representing output are prefixed with o_.

1.6 References and Acknowledgments

The following standards has been applied:

J-STD-016-1995 IEEE/EIA Standard for Information Technology, Software Lifecycle Processes, Software Development, Acquirer-Supplier Agreement

IEEE-STD-P1063 IEEE Standard for Software User Documentation

The following document has been referred:

Morpheus Page (2014). SRS Example – Mini Thermostat.

2 Overall Description

2.1 Product Perspective

The proposed game is Mario game. The game is designed and based from the original 'Super Mario' game. It provides different version of Mario game for game player. Before player starts his or her journey in the game, he or she has to select his or her favorite character and name it. During the journey in the game, player can move forward, jump, collects coins, and shoots the enemies.

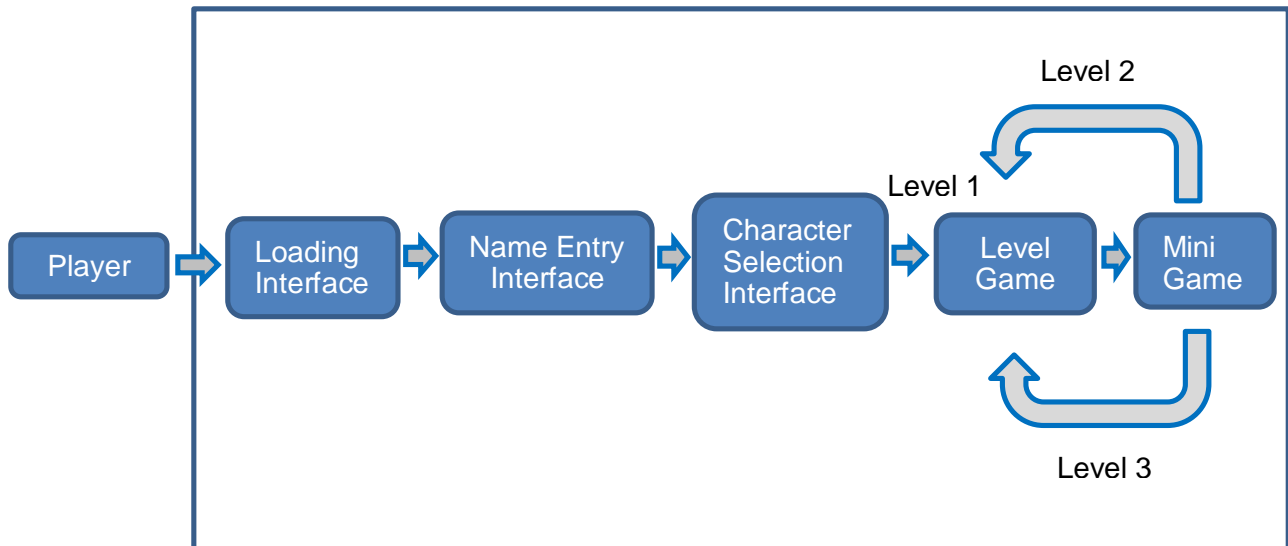


Diagram 1.0: Mario Game Flow

2.2 Product Functionality

Mario game provides the following functions:

- Character naming, means player can name the character chosen at the beginning of the game.
- Allows player to choose his or her favourite character. For example, player can choose either Mario or Luigi before enter the level one journey game.
- Allows player to move around in the 2-D environment. For example, the character that has been chosen can move forward, move backward and jump.
- Allows player to shoot enemies.
- Provides player with the game music.
- Displays coins collected by the player for the level.
- Displays fireballs (bullets) collected by the player for the level.

2.3 Users and Characteristics

No	Name	Role	Area of Expertise
1.	Mdm Nurfaeza Binti Jali	Coordinator	<ul style="list-style-type: none"> Checking whole game.
2.	Mdm Nurfaeza Binti Jali	Lecturer	<ul style="list-style-type: none"> Checking whole game.
3.	Jadecrystal Tang Ming Mei	Project Manager	<ul style="list-style-type: none"> Monitoring the project and ensuring consistent reporting, risk analysis, timeline and cost control.
4.	Nuratiqha Binti Abd Razak	Planning Manager	<ul style="list-style-type: none"> Managing the whole workflow for the project.
5.	Tan Sheu Yeu	Development Manager	<ul style="list-style-type: none"> Leading the team in producing the high-level design, implementing the product and producing the product's user documentation.
6.	Lee Fui Yee	Quality and Process Manager	<ul style="list-style-type: none"> Monitoring and advising the game performance. Ensuring customer requirements and expectations have been accurately that organizations are meeting customer expectations.
7.	Vernon Chien	Support Manager	<ul style="list-style-type: none"> Leading the team in determining its support needs in obtaining needed tools and facilities.
8.	Players	Players	<ul style="list-style-type: none"> Following instructions to play the game.

Table 3.0: Users and Characteristics

2.4 Operating Environment

Mario game can be played in any operating system since it is not platform dependence. It is an offline game. Hence, no internet connection is needed for player to play the game. The game can be installed at any time and at any places as long as there is computer which with Java compiler installed. The game can only be played by a player at a time.

2.5 Design and Implementation Constraints

Mario is written in Java. Its user interface is written with Java programming language, so anyone who wishes to further develop the project has to have adequate knowledge in this programming language. Multilingual support is not available in this game.

2.6 User Documentation

At the present time no user documentation is available for this product. This document should serve as the basis for all user documentation to follow. Each step in the development of the product shall be documented and a detailed user manual shall be compiled during the development process.

2.7 Assumptions and Dependencies

The following is a list of assumed factors that could significantly affect the requirements stated in this document.

- The minimum hardware must be in place.
- A working Java Runtime Environment is necessary.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface shall consist of six main GUI screens. The three interfaces are:

1. Loading Screen – UI01
2. Name Entry Screen – UI02
3. Instruction Screen – UI03
4. Character Selection Screen – UI04
5. Game Over and Replay Screen – UI05
6. Win Game Screen – UI06

3.1.1.1 Loading Screen – UI01

The Loading Screen shall serve as the main UI for the game.

UI01 Screen shall display the following things:

- Loading message: “LOADING...”

Player has to wait until loading is finished and the screen then will switch to Name Entry Screen – UI02.

An example screenshot of UI01 is provided in Figure 1:



Figure 1: Mario game sample UI01

3.1.1.2 Name Entry Screen – UI02

The player should be able to enter the name for the character chosen.

UI02 shall display the following things:

- The welcoming message: “Welcome to Mario”.
- The prompting message: “Enter your name.”.

UI02 shall have the following list of textbox and button in its interface:

- Name textbox: shall allow the player to input i_Name.
- “OK” button: shall allow the Name Entry screen – UI02 switch to Character Selection screen – UI04.
- “How to play” button: shall allow the Name Entry Screen – UI02 switch to Instruction screen – UI03.

An example screenshot of UI02 is provided in Figure 2:



Figure 2: Mario game sample UI02

3.1.1.3 Instruction Screen – UI03

The player should be able to choose the action after they had finished read the instruction.

UI03 shall display the following things:

- The title: “How to play?”.
- The instruction: “Move left”, “Move right”, “Jump”, “Fire”.

UI03 shall have the following list of textbox and button in its interface:

- “Back” button: shall allow the Instruction screen – UI03 switch to Name Entry screen – UI02.

An example screenshot of UI03 is provided in Figure 3:



Figure 3: Mario game sample UI03

3.1.1.4 Character Selection Screen – UI04

The player should be able to choose their favourite character.

UI04 shall display the following things:

- The welcoming message with name given: “Welcome XXX,”, where XXX is the name given in UI02.
- The prompting message: “Please choose one character”.

UI03 shall have the following buttons in its user interface:

- Character 1 button: shall allow the player to choose character 1 (Mario).
- Character 2 button: shall allow the player to choose character 2 (Luigi).

An example screenshot of UI03 is provided in Figure 3:



Figure 4: Mario game sample UI04

3.1.1.5 Game Over and Replay Screen

The player should be able to choose the action when the game is over.

UI05 shall displays the following things:

- Game over message.
- The prompting message: "Do you want to replay?"

UI04 shall have the following buttons in its user interface:

- "Yes" button: Shall allow user to replay the game.
- "No" button: Shall allow user end and exit the game.

An example of screenshot of UI05 is provided in Figure 5:

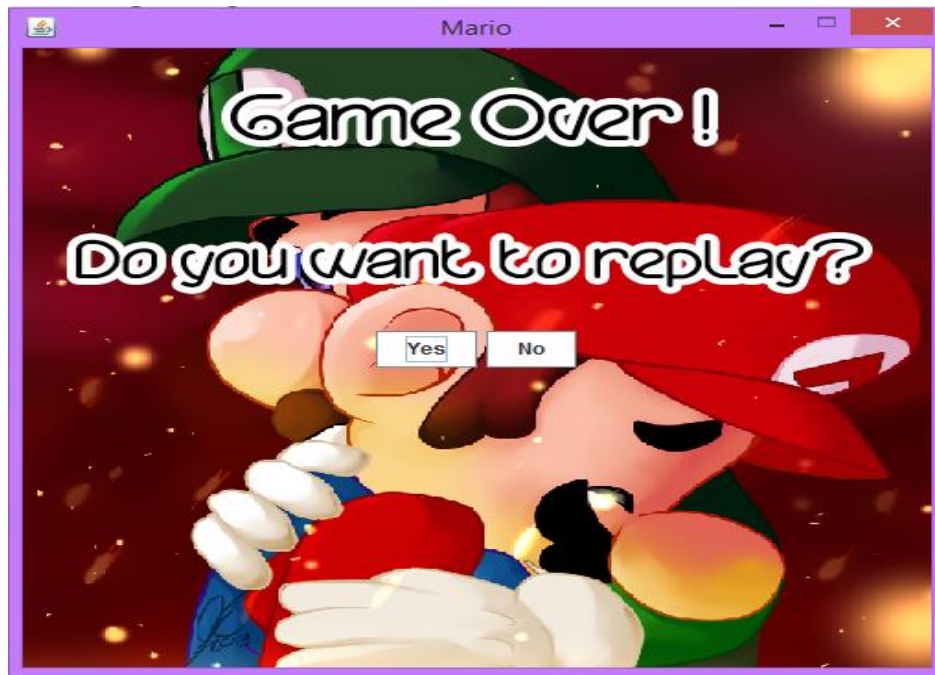


Figure 5: Mario Game Sample UI05

3.1.1.6 Win Game Interface

The player should be able to choose the action when he or she win the whole game.

UI06 shall displays the following things:

- Win whole game message: "Congratulation, You have finish all the levels".
- The prompting message: "Do you want to replay?"

UI06 shall have the following buttons in its user interface:

- "Yes" button: Shall allow player to replay the game.
- "No" button: Shall allow player end and exit the game.

An example of screenshot of UI06 is provided in Figure 6:



Figure 6: Mario Game Sample UI06

3.1.2 Hardware Interfaces

Mario game runs on any computer hardware meeting the following criteria:

- Includes a mouse
- Includes a keyboard

3.1.3 Software Interfaces

Mario game integrates some external software to provide functionality.

Client: Mario game interfaces with the user computer and expect that it have java environment installed.

3.1.4 Communications Interfaces

Mario game is not a web application, so no communications functions and communication standards are required by the game.

3.2 Functional Requirements

3.2.1 Character Naming

- Players shall be able to name the character after finish loading.
- When “OK” button is pressed, current Character Naming screen shall be switched to next screen which is Character Selection screen.

3.2.2 Character Selection

- Players shall be given the option in selecting the character, they can either choose Mario or Luigi.
- When character is selected, current Character Selection screen shall be switched to next screen which is the screen where the level 1 of the game starts.

3.2.3 Move Around

- Players shall be able to move forward when forward key on keyboard is pressed.
- Players shall be able to move backward when backward key on keyboard is pressed.
- Players shall be able to jump when up key on keyboard is pressed.

3.2.4 Shooting

- Players shall be able to shoot the enemies in Mario game when space bar on the keyboard is pressed.

3.2.5 Game Music

- Music shall be played once the players open Mario game.

3.2.6 Recording

- Coins collected by the player shall be recorded and displayed on the top of the windows.
- Fireballs (bullets) collected by the player shall be recorder and displayed on the top of the windows.

3.3 Behaviour Requirements

3.3.1 Use Case View

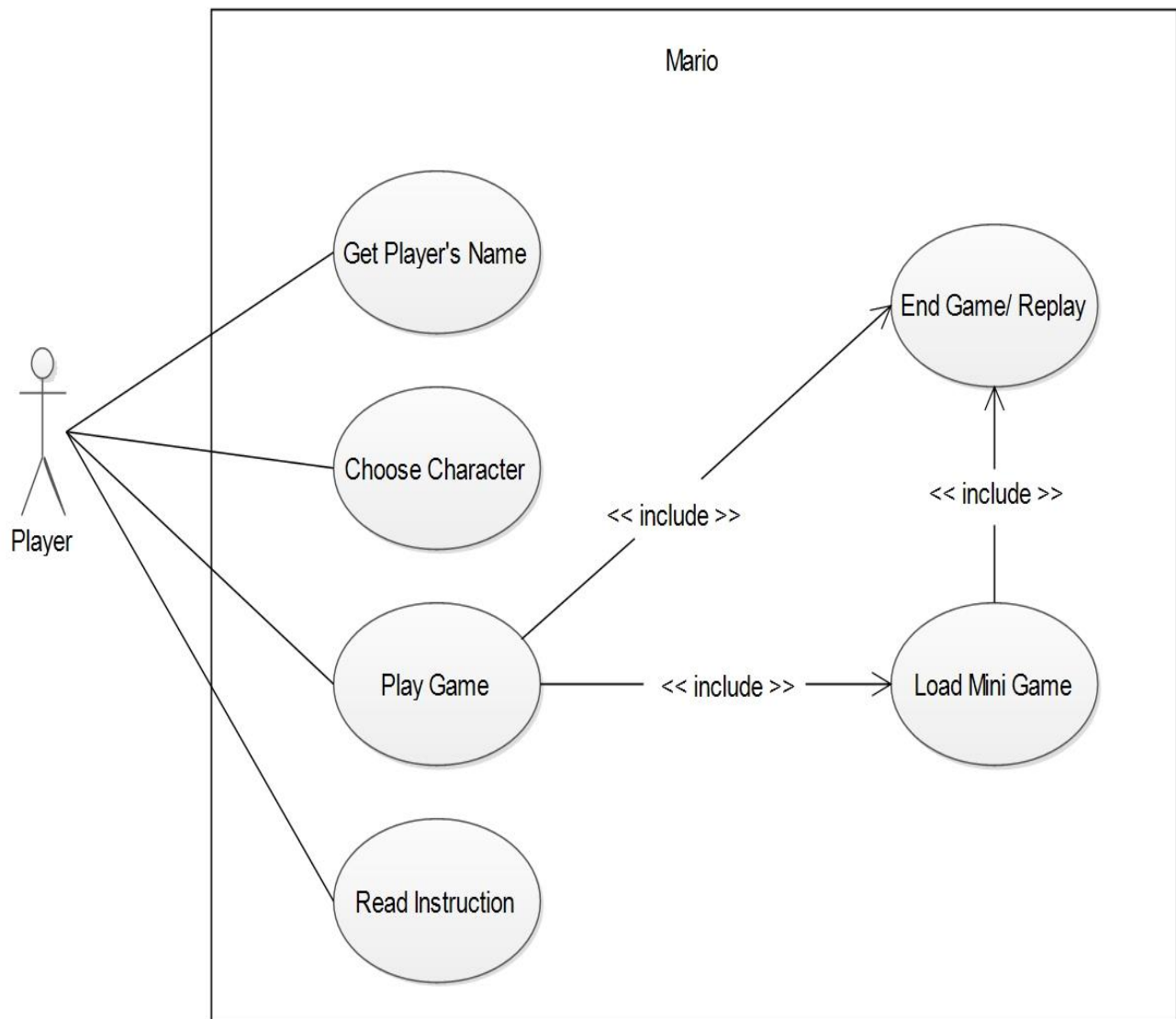


Diagram 2.0: Mario Game Use Case Diagram

Mario: Get player's name	
Actor	Player
Description	The game will provide an interface at every beginning whenever player wants to play the game. The game will prompt player to enter his or her name. After that, the game will continue after player clicks the OK button.
Data	Player's name
Stimulus	User command issued by player
Response	Continue the game
Comments	Player can only key in the name once, the name cannot be changed after the player clicks OK button.

Table 4.0: Use Case 1 Get player's name

Mario: Choose character	
Actor	Player
Description	The game will print out player's name and prompt player to choose the character. Player chooses the character by clicking one of the CHARACTER button. The game will continue after player clicks the CHARACTER button.
Data	Character
Stimulus	User command issued by player
Response	Continue the game with the chosen character
Comments	Player can only click one character. The chosen character will be used throughout the game.

Table 5.0: Use Case 2 Choose Character

Mario: Play game	
Actor	Player
Description	Player starts play the game. The Mario will jump after pressing the UP on the keyboard. Moreover, the Mario will move forward and backward by pressing the RIGHT and LEFT on the keyboard. A fireball will be fired each time when player presses SPACEBAR on the keyboard. Player has to collect fireballs to kill the enemies in the game and collect coins.
Data	Key event from keyboard Action event from mouse
Stimulus	User command issued by player
Response	The game end. The game exit.
Comments	The game will end before the players wins if the Mario is die.

Table 6.0: Use Case 3 Play Game

Mario: Load Mini Game	
Actor	Player
Description	Player starts play the mini game. The player will answer questions with educational purpose by pressing the character or number key on the keyboard. Player has to answer the questions within limited time frame.
Data	Key event from keyboard
Stimulus	User command issued by player
Response	Success: Enter next level game. Fail (with enough coins): Coins will be deducted. Fail (not enough coins): Game over.
Comments	Player can't finish answering the given questions within limited time frame is considered fail.

Table 7.0: Use Case 4 Mini Game

Mario: End Game / Replay	
Actor	Player
Description	The game will provide an interface whenever player fails to play the game, no matter it is level game or mini game. The game will prompt player to select the action to be taken. After that, the game will restart if player clicks the 'yes' button or it will end and exit if player clicks 'no' button.
Data	Action event from mouse
Stimulus	User command issued by player
Response	Yes: Replay the game. No: Exit the game.
Comments	Player can also click 'x' button to exit the game.

Table 8.0: Use Case 5 Game Over/ Replay

Mario: Read Instruction	
Actor	Player
Description	The game will provide an interface at every beginning whenever player wants to play the game. Instruction screen will be shown whenever player click "Instruction" button.
Data	Action event from mouse
Stimulus	User command issued by player
Response	Instruction of the game shown
Comments	This function is optional to player

Table 9.0: Use Case 6 Reading Instruction

3.3.2 Sequence Diagrams

3.3.2.1 Naming Character

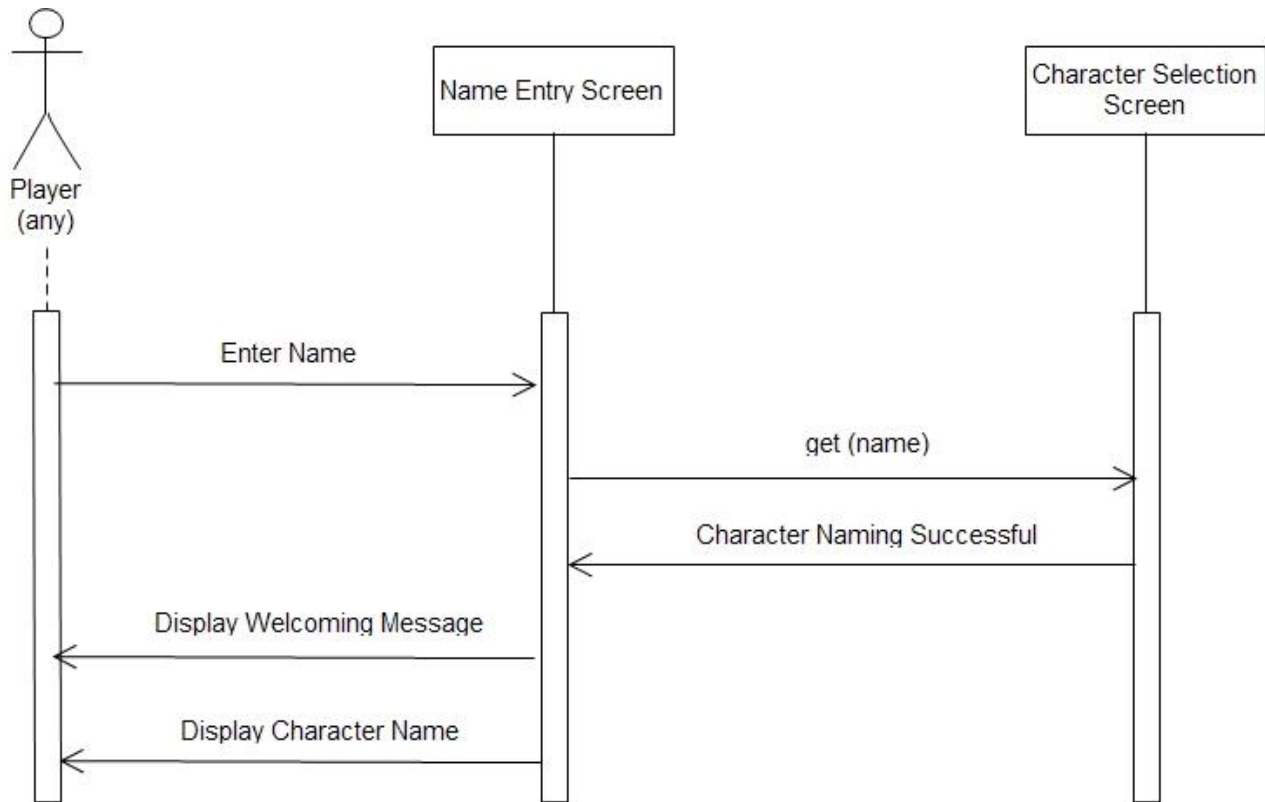


Diagram 3.0: Naming Character

Description:

1. Player enter the name for the character.
2. After the game application get the name, character naming is considered successful.
3. Character's name and welcoming message will be displayed on the screen.

3.3.2.2 Character Selection

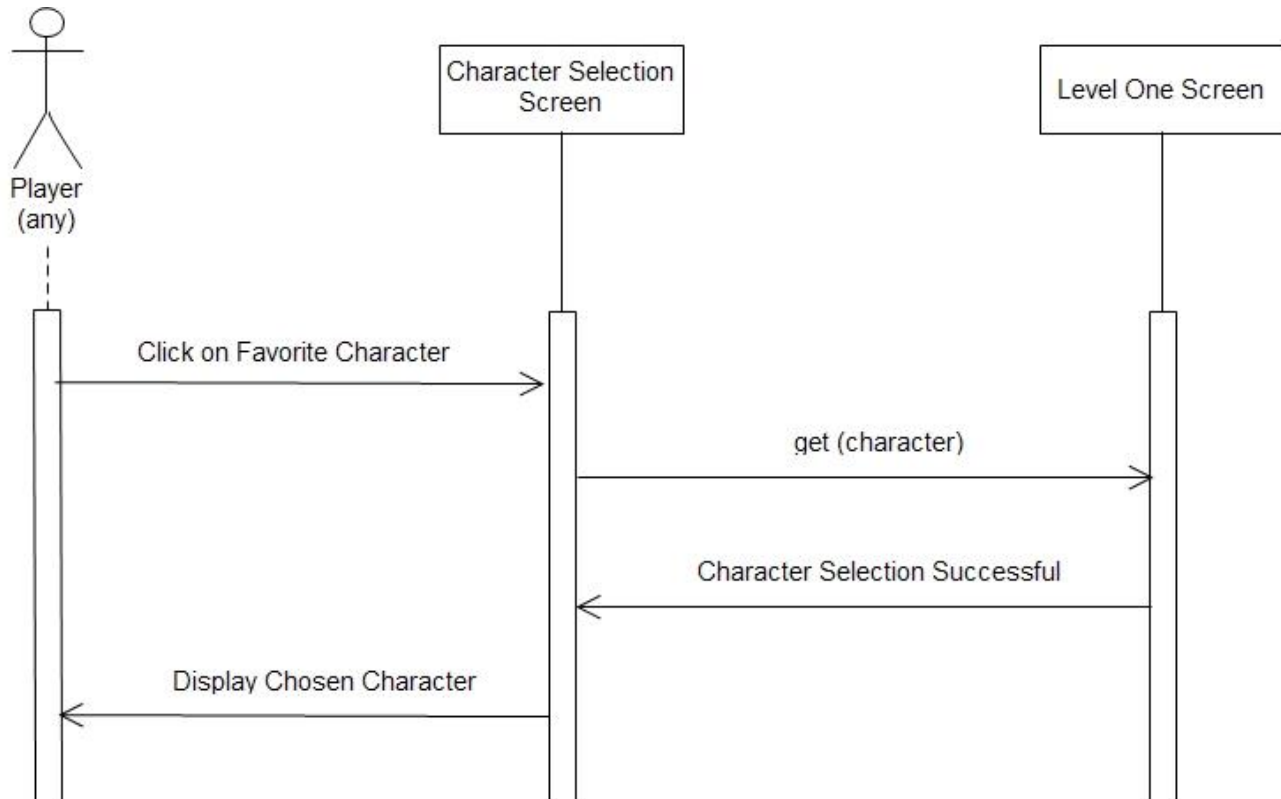


Diagram 4.0: Character Selection

Description:

1. Player clicks on favorite character, that is either Mario or Luigi.
2. After the game application get the selected character, character selection is considered successful.
3. Chosen character will be displayed throughout the game.

3.3.2.3 Reading Instruction

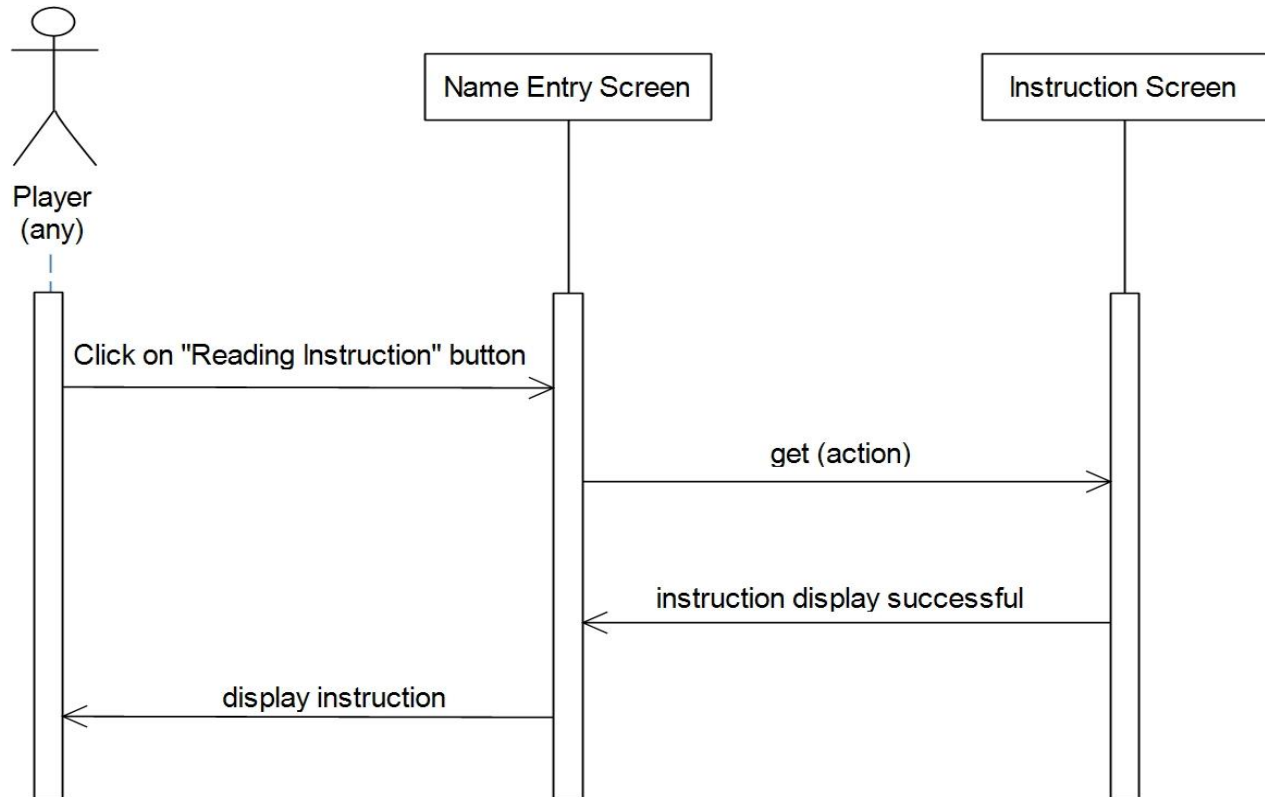


Diagram 5.0: Reading Instruction

Description:

1. Player clicks "Reading Instruction" button.
2. After the game application get the command, instruction display is considered successful.
3. Instruction will be displayed on the screen.

3.3.2.4 Success in Level Game and Mini Game

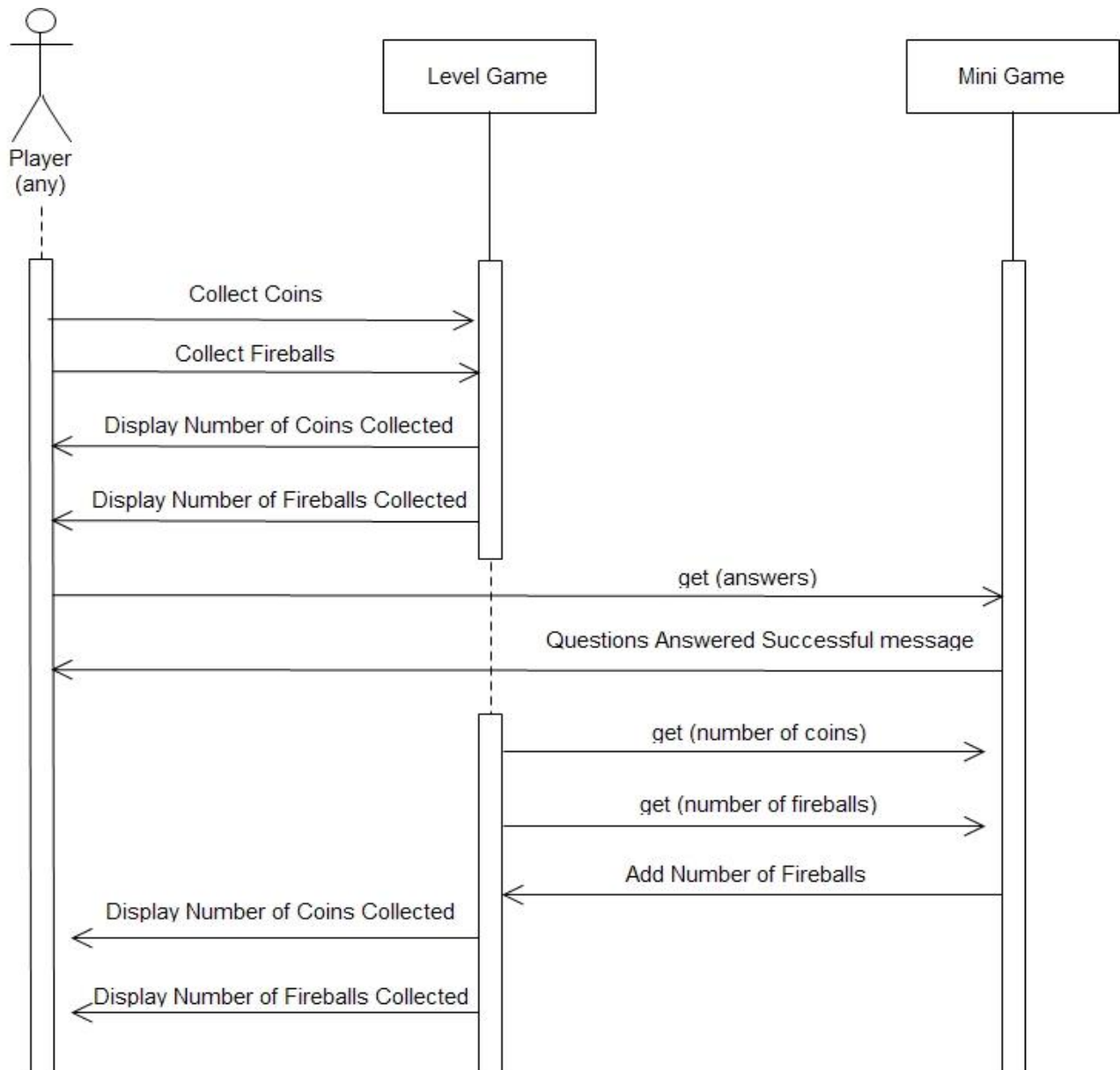


Diagram 6.0: Success in Level Game and Mini Game

Description:

1. Player collects coins and fireballs throughout the level game.
2. The game application displays the number of coins and fireballs collected by the player.

3. After player success in level game, player can continue playing the mini game.
4. Game application gets the answers for mini game from player.
5. Player gets at eight or more questions correct.
6. Success in mini game's message will be displayed.
7. Game application add five fireballs to the player's accumulated fireballs.
8. Number of coins and fireballs will be displayed in the next level game.

3.3.2.5 Success in Level Game and Fail in Mini Game (Enough Coins)

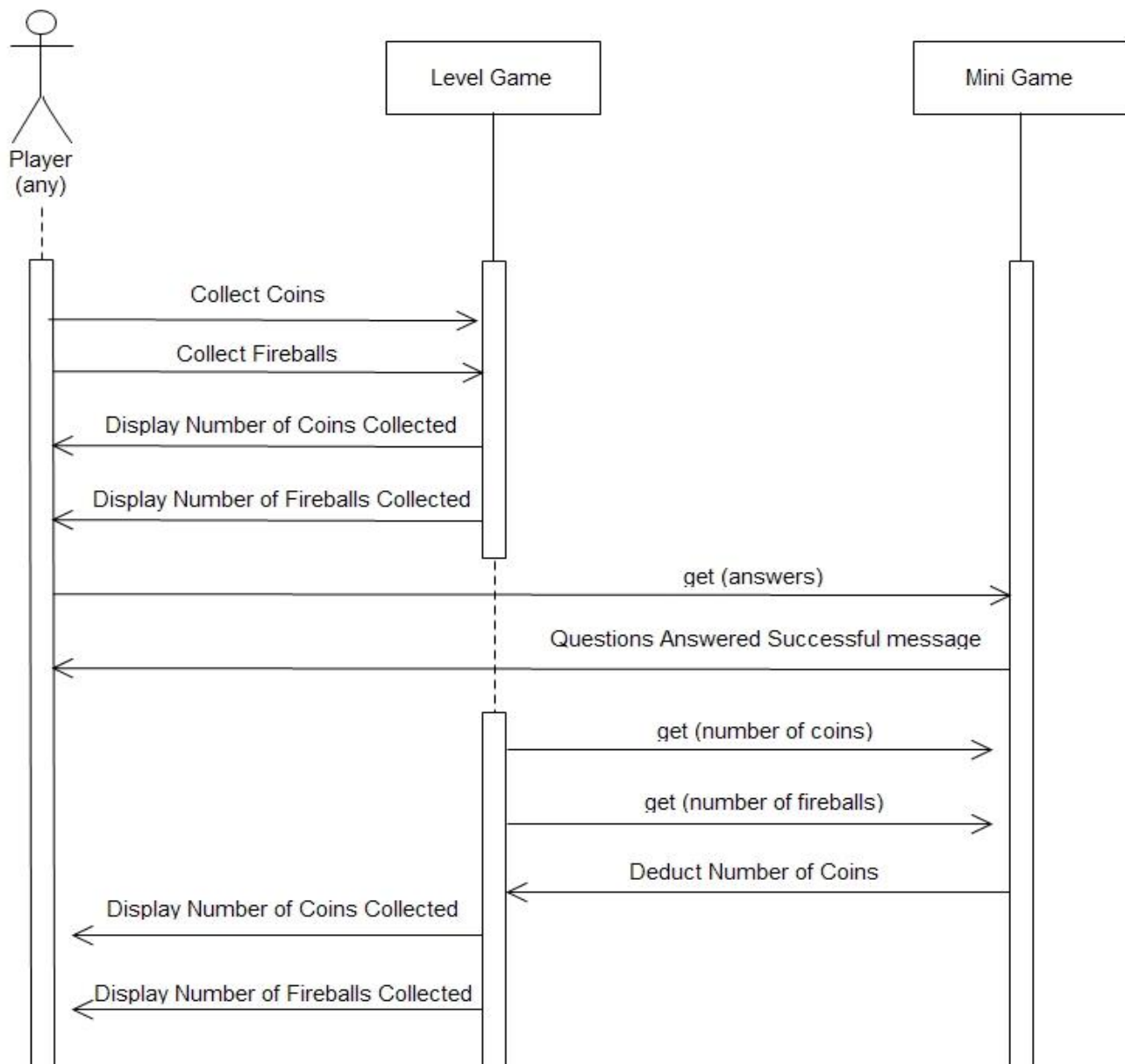


Diagram 7.0: Success in Level Game and Fail in Mini Game (Enough Coins)

Description:

1. Player collects coins and fireballs throughout the level game.
2. The game application displays the number of coins and fireballs collected by the player.
3. After player success in level game, player can continue playing the mini game.
4. Game application gets the answers for mini game from player.
5. Player gets less than eight questions correct.
6. Game application prompts player's decision whether he/ she want to continue the game or not, with the condition 20 coins will be deducted.
7. Player decides to continue the game.
8. Game application deduct 20 coins from the player's accumulated coins.
9. Number of coins and fireballs will be displayed in the next level game.

3.3.2.6 Success in Level Game and Fail in Mini Game (Not Enough Coins)

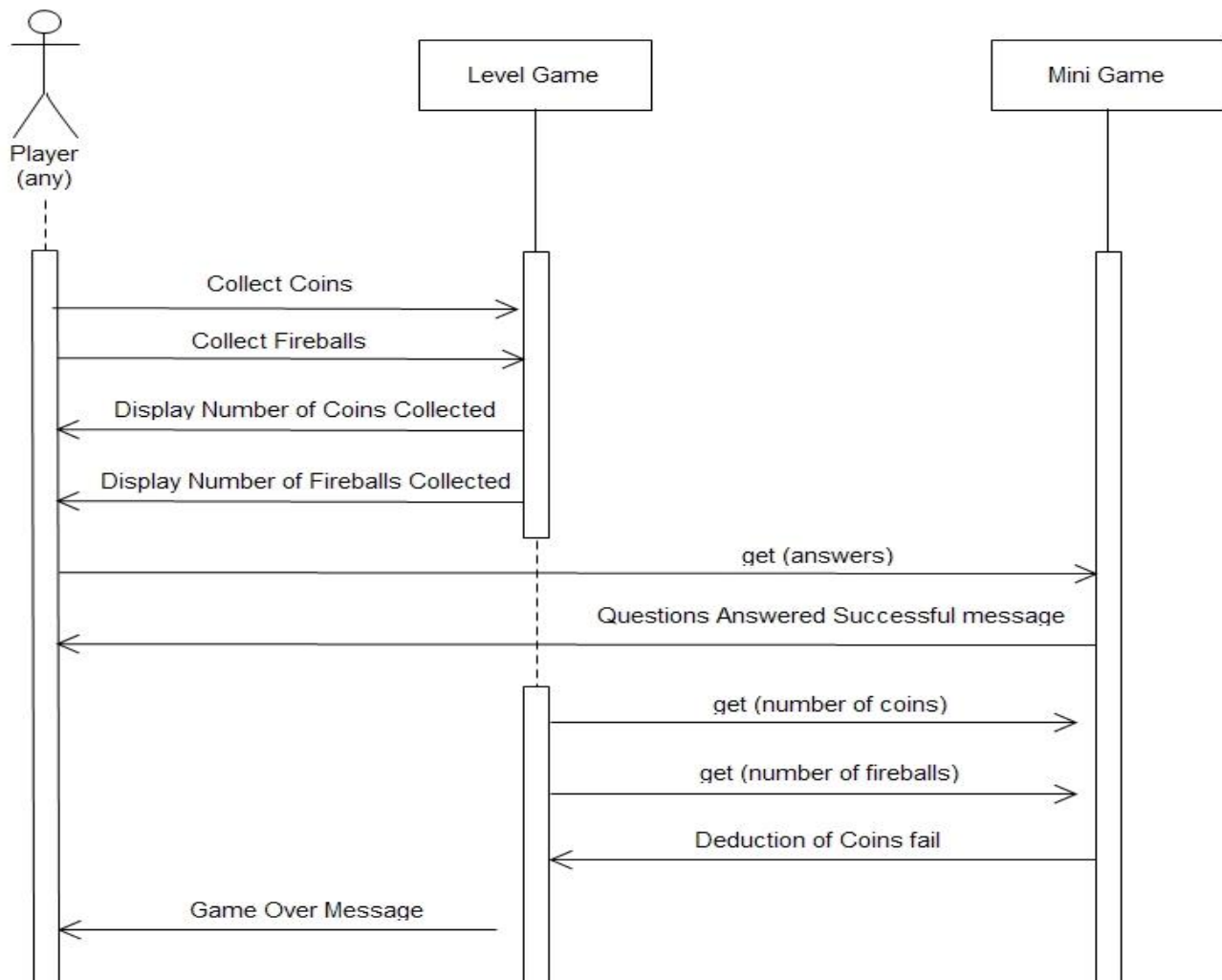


Diagram 8.0: Success in Level Game and Fail in Mini Game (Not Enough Coins)

Description:

1. Player collects coins and fireballs throughout the level game.
2. The game application displays the number of coins and fireballs collected by the player.
3. After player success in level game, player can continue playing the mini game.
4. Game application gets the answers for mini game from player.
5. Player gets less than eight questions correct.
6. Game application prompts player's decision whether he/ she want to continue the game or not, with the condition 20 coins will be deducted.
7. Player decides to continue the game.
8. Player does not have enough coins.
9. Game over message will be displayed on the screen.

3.3.2.7 Fail in Level Game (Have Life)

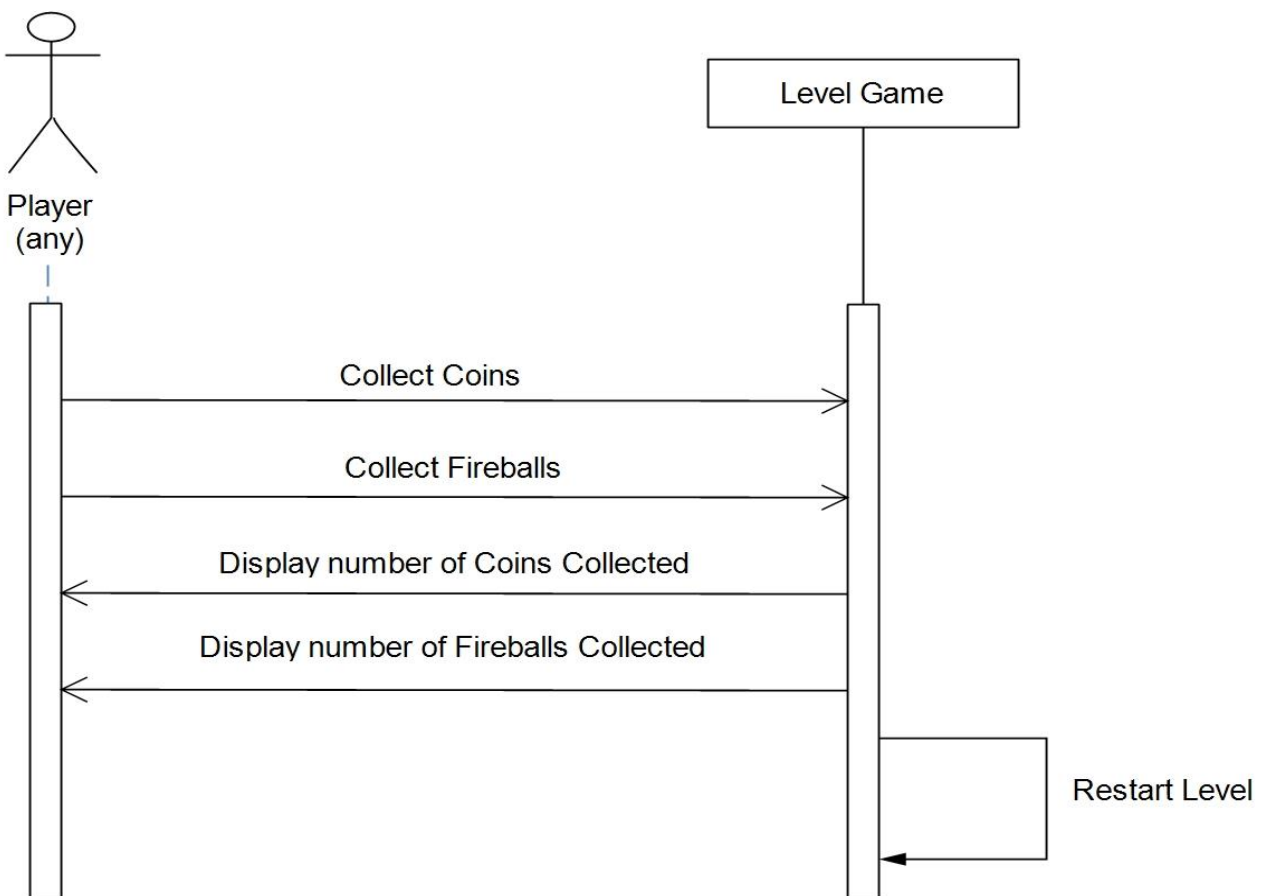


Diagram 9.0: Fail in Level Game (Have Life)

Description:

1. Player collects coins and fireballs throughout the level game.
2. The game application displays the number of coins and fireballs collected by the player.
3. After player fail in level game.
4. player still have life can restart the current level game.

3.3.2.8 Fail in Level Game (No Life)

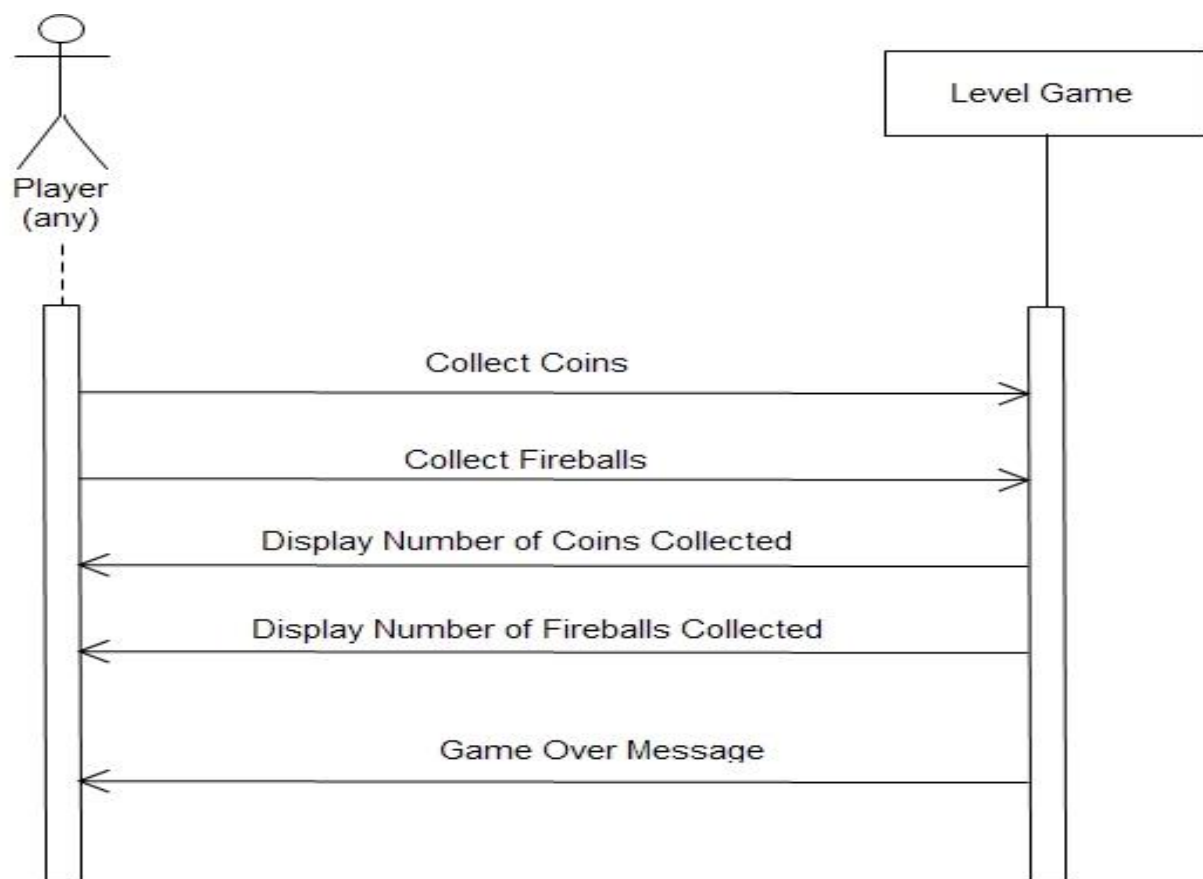


Diagram 10.0: Fail in Level Game (No Life)

Description:

1. Player collects coins and fireballs throughout the level game.
2. The game application displays the number of coins and fireballs collected by the player.
3. After player fail in level game.

4. Player don't have life anymore.
5. Game over message will be displayed on the screen.

3.3.3 State Diagram

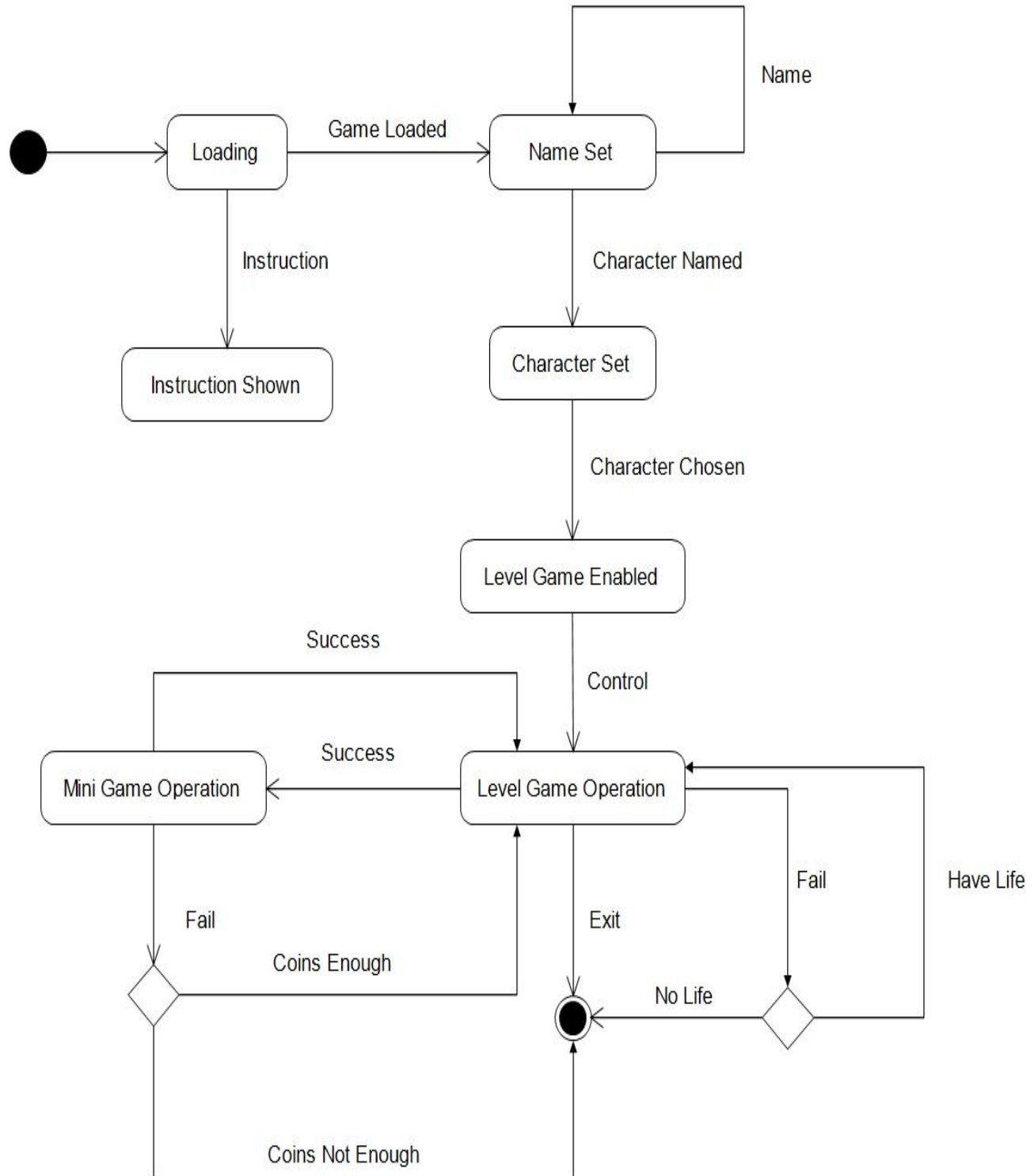


Diagram 11.0: Mario Game State Diagram

3.3.4 Activity Diagram

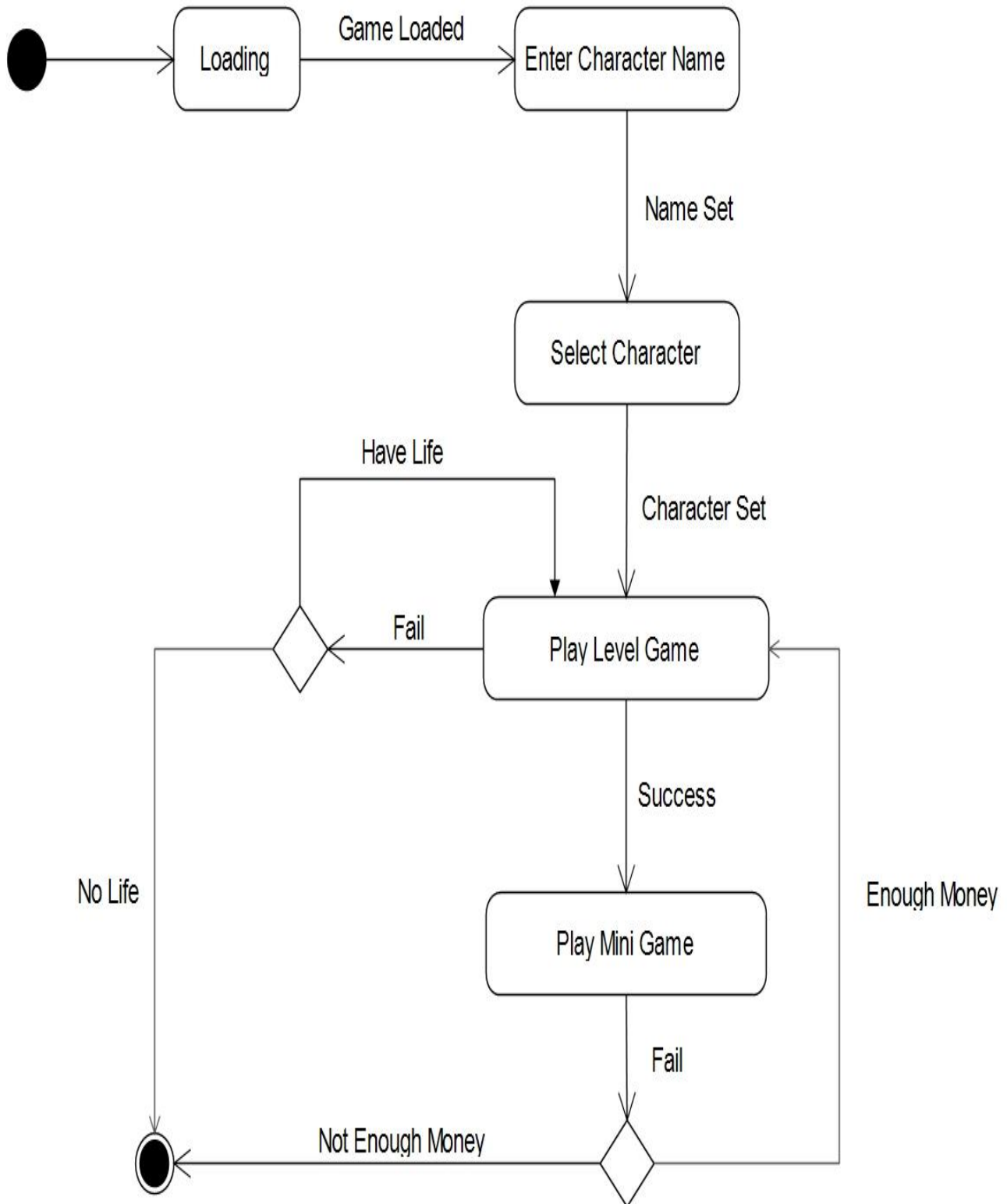


Diagram 12.0: Mario Game Activity Diagram

Description:

1. Game is loading.
2. After the game is loaded, player enters the name for the character.
3. After the character's name is set, player selects his/ her favorite character for the game.
4. After the chosen character is set, player starts playing level game.
5. Player success in level game continue playing mini game.
6. Player fail in level game but still have life will restart the level game.
7. Player fail in level game and don't have life anymore will end the game.
8. Player success in mini game continue playing next level game.
9. Player fail in mini game but still have enough coins to be deducted will be given options that is either continue playing next level game or end the game.
10. Player fail in mini game and don't have enough coins to be deducted will end the game.

4 Other Non-functional Requirements

4.1 Performance Requirements

Mario game consists of the following performance requirements:

- The loading time for Mario game will not take more than 10 seconds.
- The game shall respond to each player input within 3 seconds.
- Only single player gameplay is permitted.
- No internet connection is needed.
- The game must be able to run on all computer platform with java compiler installed.

4.2 Safety and Security Requirements

4.2.1 Safety

There are no safety requirements as the game has only three levels and is safe for children who are susceptible to flashing lights which are absent.

4.2.2 Security

- The game will not access any player data files or programs.
- The game will not alter or replace any system files.

4.3 Software Quality Attributes

4.3.1 Usability

- A new player should be able to play a complete game of Mario in less than 10 minutes.
- A new player should commit less than one error in use of the game every ten minutes.
- A user who is familiar with the traditional Mario game be able to correctly operate the game without any written documentation.

4.3.2 Portability

- The game is Java based and should be compatible to devices with Java implementations.

4.3.3 Reliability

- Since the game is purely for recreation and involves no user data, so reliability is of low importance.

4.3.4 Availability

- The game is Java based game which means it can be installed and played as long as the player has devices with Java implementations.
- The game is not a web application which means it is available to the player without internet connection.

5 Other Requirements

There is no other requirements for Mario game.

Appendix A – Data Dictionary

Entity	Variable Name	Description
Board1	mario	The game character is bring to level 1 game
Board1	e1	Enemy
Board1	e2	Enemy
Board1	e3	Enemy
Board1	e4	Enemy
Board1	e5	Enemy
Board1	e6	Enemy
Board1	e7	Enemy
Board1	e8	Enemy
Board1	e9	Enemy
Board1	e10	Enemy
Board1	mushroom	End of level 1 game
Board1	c1	Coin
Board1	c2	Coin
Board1	c3	Coin
Board1	c4	Coin
Board1	c5	Coin
Board1	c6	Coin
Board1	c7	Mystery box
Board1	c8	Mystery box
Board1	c9	Mystery box
Board1	c10	Mystery box
Board1	c11	Coin
Board1	c12	Coin
Board1	c13	Coin
Board1	c14	Coin
Board1	c15	Coin
Board1	c16	Coin
Board1	c17	Coin
Board1	c18	Mystery box
Board1	c19	Mystery box
Board1	c20	Mystery box
Board1	c21	Mystery box
Board1	c22	Coin
Board1	c23	Coin
Board1	c24	Coin
Board1	c25	Coin
Board1	c26	Coin
Board1	c27	Coin
Board1	c28	Coin
Board1	c29	Mystery box
Board1	c30	Mystery box
Board1	c31	Mystery box
Board1	c32	Coin
Board1	c33	Coin
Board1	c34	Coin

Board1	c35	Coin
Board1	c36	Coin
Board1	c37	Coin
Board1	c38	Coin
Board1	f1	Fireballs award
Board1	f2	Fireballs award
Board1	background	Background of level 1 game
Board1	time	Timer
Board1	animator	Thread
Board1	check	Allow jumping process to begin
Board1	peak	Check whether the game character jump to the peak or not
Board1	done	Check whether the jumping process finish or not
Board1	lost	Check whether the player die in the level 1 game or not
Board1	next	Check whether the player able to reach end of level 1 game or not
Board1	font	Set the font size and pattern to display at the top of the windows
Board1	value	Set the y-coordinate of the game character while the game character is jumping
Board1	frame	Get the frame from the previous windows
Board1	CHARC	Get the game character chose previously
Board2	mario	The game character is bring to level 2 game
Board2	e1	Enemy
Board2	e2	Enemy
Board2	e3	Enemy
Board2	e4	Enemy
Board2	e5	Enemy
Board2	e6	Enemy
Board2	e7	Enemy
Board2	e8	Enemy
Board2	e9	Enemy
Board2	e10	Enemy
Board2	e11	Enemy
Board2	e12	Enemy
Board2	e13	Enemy
Board2	e14	Enemy
Board2	e15	Enemy
Board2	e16	Enemy
Board2	e17	Enemy
Board2	e18	Enemy
Board2	e19	Enemy
Board2	e20	Enemy
Board2	e21	Enemy
Board2	e22	Enemy
Board2	e23	Enemy
Board2	e24	Enemy
Board2	e25	Enemy
Board2	e26	Enemy
Board2	e27	Enemy
Board2	e28	Enemy

Board2	e29	Enemy
Board2	e30	Enemy
Board2	mushroom	End of level 2 game
Board2	c1	Coin
Board2	c2	Coin
Board2	c3	Coin
Board2	c4	Coin
Board2	c5	Mystery box
Board2	c6	Mystery box
Board2	c7	Coin
Board2	c8	Coin
Board2	c9	Coin
Board2	c10	Coin
Board2	c11	Mystery box
Board2	c12	Mystery box
Board2	c13	Mystery box
Board2	c14	Coin
Board2	c15	Coin
Board2	c16	Coin
Board2	c17	Coin
Board2	c18	Coin
Board2	c19	Coin
Board2	c20	Coin
Board2	c21	Coin
Board2	c22	Coin
Board2	c23	Mystery box
Board2	c24	Mystery box
Board2	c25	Mystery box
Board2	c26	Coin
Board2	c27	Coin
Board2	c28	Coin
Board2	c29	Coin
Board2	c30	Coin
Board2	c31	Coin
Board2	c32	Coin
Board2	c33	Mystery box
Board2	c34	Mystery box
Board2	c35	Mystery box
Board2	c36	Mystery box
Board2	c37	Mystery box
Board2	c38	Mystery box
Board2	f1	Fireballs award
Board2	f2	Fireballs award
Board2	t1	Anchor
Board2	t2	Anchor
Board2	t3	Anchor
Board2	t4	Anchor
Board2	t5	Anchor
Board2	t6	Anchor
Board2	t7	Anchor
Board2	t8	Anchor

Board2	t9	Anchor
Board2	image	Background of level 2 game
Board2	time	Timer
Board2	animator	Thread
Board2	check	Allow jumping process to begin
Board2	peak	Check whether the game character jump to the peak or not
Board2	done	Check whether the jumping process finish or not
Board2	lost	Check whether the player die in the level 2 game or not
Board2	next	Check whether the player able to reach end of level 2 game or not
Board2	font	Set the font size and pattern to display at the top of the windows
Board2	value	Set the y-coordinate of the game character while the game character is jumping
Board2	frame	Get the frame from the previous windows
Board2	CHARC	Get the game character chose previously
Board2	ran	Random number
Board3	mario	The game character is bring to level 3 game
Board3	e1	Enemy
Board3	e2	Enemy
Board3	e3	Enemy
Board3	e4	Enemy
Board3	e5	Enemy
Board3	e6	Enemy
Board3	e7	Enemy
Board3	e8	Enemy
Board3	e9	Enemy
Board3	e10	Enemy
Board3	e11	Enemy
Board3	e12	Enemy
Board3	e13	Enemy
Board3	e14	Enemy
Board3	e15	Enemy
Board3	e16	Enemy
Board3	e17	Enemy
Board3	e18	Enemy
Board3	e19	Enemy
Board3	e20	Enemy
Board3	mushroom	End of level 3 game
Board3	c1	Mystery box
Board3	c2	Mystery box
Board3	c3	Mystery box
Board3	c4	Mystery box
Board3	c5	Coin
Board3	c6	Coin
Board3	c7	Coin
Board3	c8	Coin
Board3	c9	Coin
Board3	c10	Coin
Board3	c11	Coin

Board3	c12	Coin
Board3	c13	Mystery box
Board3	c14	Mystery box
Board3	c15	Mystery box
Board3	c16	Mystery box
Board3	c17	Mystery box
Board3	c18	Mystery box
Board3	c19	Mystery box
Board3	c20	Mystery box
Board3	c21	Coin
Board3	c22	Coin
Board3	c23	Coin
Board3	c24	Coin
Board3	c25	Coin
Board3	c26	Coin
Board3	c27	Coin
Board3	c28	Coin
Board3	c29	Coin
Board3	c30	Coin
Board3	c31	Coin
Board3	c32	Coin
Board3	c33	Coin
Board3	c34	Coin
Board3	c35	Mystery box
Board3	c36	Mystery box
Board3	c37	Mystery box
Board3	c38	Mystery box
Board3	f1	Fireballs award
Board3	f2	Fireballs award
Board3	t2	Fire
Board3	t3	Fire
Board3	t4	Fire
Board3	t5	Fire
Board3	t6	Fire
Board3	t7	Fire
Board3	t8	Fire
Board3	t9	Fire
Board3	background	Background of level 3 game
Board3	time	Timer
Board3	animator	Thread
Board3	check	Allow jumping process to begin
Board3	peak	Check whether the game character jump to the peak or not
Board3	done	Check whether the jumping process finish or not
Board3	lost	Check whether the player die in the level 3 game or not
Board3	next	Check whether the player able to reach end of level 3 game or not
Board3	font	Set the font size and pattern to display at the top of the windows
Board3	value	Set the y-coordinate of the game character while the game character is jumping

Board3	frame	Get the frame from the previous windows
Board3	CHARC	Get the game character chose previously
Coin	coin	Coin image
Coin	x	x-coordinate of coin
Coin	y	y-coordinate of coin
Coin	isCoin	Check whether the game character get the coin or not
End	mario	The coins and fireballs collected and life of the game character are bring to the end
End	l1	Display "Game Over!"
End	l2	Display "Do you want to replay?"
End	b1	"Yes" button
End	b2	"No" button
Enemy	enemy	Enemy image
Enemy	x	x-coordinate of enemy
Enemy	y	y-coordinate of enemy
Enemy	isAlive	Check whether the enemy is alive or not
Enemy	imageName	Name of the enemy image
Enter	l0	Display "Welcome"
Enter	l1	Display player's name
Enter	l2	Display "Please choose one character"
Enter	character1	Mario image
Enter	character2	Luigi image
Enter	b1	"Mario" button
Enter	b2	"Luigi" button
EnterName	l1	Display "Welcome to Mario"
EnterName	l2	Display "Enter your name"
EnterName	t1	Get player's name
EnterName	b1	"OK" button
EnterName	b2	"Display how to play" button
EnterName	name	Name of player
Fireball	x	x-coordinate of fireball
Fireball	y	y-coordinate of fireball
Fireball	fireballimg	Fireball image
Fireball	visible	Check whether the fireball is visible or not
Instruction	b1	"Back" button
Loading	l1	Display "Welcome to Mario"
Loading	l2	Display Mario's image
Loading	x	x-coordinate of Mario's image
Loading	y	y-coordinate of Mario's image
Mario	dx	Velocity of x-coordinate
Mario	y	y-coordinate of game character
Mario	character	Game character chose
Mario	turn	Check whether the game character is facing left of not
Mario	still	Game character's image
Mario	image1	Mario still's image
Mario	image2	Mario turn left's image
Mario	image3	Mario jump's image
Mario	image4	Luigi still's image
Mario	image5	Luigi left's image

Mario	image6	Luigi jump's image
Mario	fireballs	Fireballs
Mario	x	x-coordinate of game character
Mario	nx2	Temporary variable for x-coordinate
Mario	nx	Temporary variable for x-coordinate
Mario	left	Default value for x-coordinate of game character
Mario	dy	Velocity of y-coordinate
Mario	fireballNum	Number of fireballs
Mario	coin	Number of coins
Mario	life	Number of lives
MiniGame1	mario	The game character is bring to mini game 1
MiniGame1	countdownTimer	Timer
MiniGame1	timeRemaining	Remaining time
MiniGame1	label1	Display "Mini Game"
MiniGame1	label2	Display remaining time
MiniGame1	l1	Display question 1
MiniGame1	l2	Display question 2
MiniGame1	l3	Display question 3
MiniGame1	l4	Display question 4
MiniGame1	l5	Display question 5
MiniGame1	l6	Display question 6
MiniGame1	l7	Display question 7
MiniGame1	l8	Display question 8
MiniGame1	l9	Display question 9
MiniGame1	l10	Display question 10
MiniGame1	r1	Random number
MiniGame1	r2	Random number
MiniGame1	r3	Random number
MiniGame1	r4	Random number
MiniGame1	r5	Random number
MiniGame1	r6	Random number
MiniGame1	r7	Random number
MiniGame1	r8	Random number
MiniGame1	r9	Random number
MiniGame1	r10	Random number
MiniGame1	r11	Random number
MiniGame1	r12	Random number
MiniGame1	r13	Random number
MiniGame1	r14	Random number
MiniGame1	r15	Random number
MiniGame1	r16	Random number
MiniGame1	r17	Random number
MiniGame1	r18	Random number
MiniGame1	r19	Random number
MiniGame1	r20	Random number
MiniGame1	numWin	Number of correct answers
MiniGame1	a1	Answer for question 1
MiniGame1	a2	Answer for question 2
MiniGame1	a3	Answer for question 3
MiniGame1	a4	Answer for question 4
MiniGame1	a5	Answer for question 5

MiniGame1	a6	Answer for question 6
MiniGame1	a7	Answer for question 7
MiniGame1	a8	Answer for question 8
MiniGame1	a9	Answer for question 9
MiniGame1	a10	Answer for question 10
MiniGame1	submit	"Submit" button
MiniGame1	CHARC	Get the game character chose previously
MiniGame1	t1	Get answer for question 1
MiniGame1	t2	Get answer for question 2
MiniGame1	t3	Get answer for question 3
MiniGame1	t4	Get answer for question 4
MiniGame1	t5	Get answer for question 5
MiniGame1	t6	Get answer for question 6
MiniGame1	t7	Get answer for question 7
MiniGame1	t8	Get answer for question 8
MiniGame1	t9	Get answer for question 9
MiniGame1	t10	Get answer for question 10
MiniGame2	mario	The game character is bring to mini game 2
MiniGame2	CHARC	Get the game character chose previously
MiniGame2	r	r value of RGB
MiniGame2	g	g value of RGB
MiniGame2	b	b value of RGB
MiniGame2	timeRemaining	Remaining time
MiniGame2	countDownTlme	Countdown time
MiniGame2	i	Question's number
MiniGame2	score	Score of the game
MiniGame2	Randword	Word
MiniGame2	rand	Random number
MiniGame2	countdownTimer	Timer
MiniGame2	countdownTimer2	Timer
MiniGame2	pInstr	Panel
MiniGame2	pWord	Panel
MiniGame2	lWord	Display word
MiniGame2	lTitle	Display "Mini Game"
MiniGame2	lTime	Display remaining time
MiniGame2	lScore	Display score
MiniGame2	lAns	Display answer
MiniGame2	click	"Hit me when color matches word!" button
MiniGame2	lLine	Display line
MiniGame2	lLine2	Display line
Mushroom	mushroom	Mushroom image
Mushroom	x	x-coordinate of mushroom
Mushroom	y	y-coordinate of mushroom
Mushroom	isAlive	Check whether the game character able to get the mushroom or not
Win	mario	The game character is bring to win
Win	b1	"Yes" button
Win	b2	"No" button

Appendix B - Group Log

Date	Time	Activity	Attendance
8/ 9/ 2014 – 14/ 9/ 2014, Monday – Friday	-	<ul style="list-style-type: none"> Group project has released. A group member has been formed. Total of 5-6 members. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien
24/ 9/ 2014, Wednesday	(1400 – 1700)	<ul style="list-style-type: none"> Assigned position of every member. Defined the project scope. Discussed about the project, identified the objectives and goals of the project. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien
29/ 9/ 2014 – 3/ 10/ 2014, Monday – Friday	-	<ul style="list-style-type: none"> Prepared the initial proposal. Presented the initial proposal. Initial proposal handed. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien
6/ 10/ 2014, Monday	(1900 – 2230)	<ul style="list-style-type: none"> Defined the target player. Planned project timeline. Distributed each task that need to be corrected before hand-in the edited copy of proposal. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien
10/ 10/ 2014, Friday	(1430 – 1630)	<ul style="list-style-type: none"> Final proposal handed. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien
13/ 10/ 2014, Monday	(1300 – 1600)	<ul style="list-style-type: none"> Discussion for the design of use case diagram, class diagram and sequence diagram of the game. 	<ul style="list-style-type: none"> Jadecrystal Tang Ming Mei Lee Fui Yee Nuratiqha Binti Abd Razak Tan Sheu Yeu Vernon Chien

20/ 10/ 2014	(1300 – 1600)	<ul style="list-style-type: none"> • Use case diagram, class diagram and sequence diagram have improved. • Discussion for the design of activity diagram and project interfaces. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
28/ 10/ 2014, Tuesday	(2000 – 0000)	<ul style="list-style-type: none"> • Activity diagram and project interfaces have improved. • Task distributed for each member for the SRS documentation. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
31/ 10/ 2014, Friday	(1630 – 1930)	<ul style="list-style-type: none"> • A basic interfaces of the game are coded. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
4/ 11/ 2014, Tuesday	(1300 – 1600)	<ul style="list-style-type: none"> • First draft of the SRS documentation completed. • Discussed regarding the documentation and checked for any wrong in the documentation. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
7/ 11/ 2014, Friday	(1630 – 1930)	<ul style="list-style-type: none"> • 50 % of the game has completed. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
10/ 11/ 2014 – 14/ 11/ 2014, Monday – Friday	-	<ul style="list-style-type: none"> • Presented the SRS documentation. • SRS documentation handed. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
17/ 11/ 2014, Monday	(1300 – 1500)	<ul style="list-style-type: none"> • Task distributed for each member for the SDS documentation. 	<ul style="list-style-type: none"> • JadeCrystal Tang Ming Mei • Lee Fui Yee

			<ul style="list-style-type: none"> • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
21/ 11/ 2014, Friday	(1630 – 1830)	<ul style="list-style-type: none"> • 70 % of the game has completed. 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
24/ 11/ 2014, Monday	(1400 – 1600)	<ul style="list-style-type: none"> • Discuss and started the TSPi forms 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
26/ 11/ 2014, Wednesday	(1400 – 1600)	<ul style="list-style-type: none"> • Continue of the TSPi forms 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
28/ 11/ 2014, Friday	(1630 – 1830)	<ul style="list-style-type: none"> • Whole game has completed. • Discussion of the game flow and pointed out some game's problem. 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
1/ 12/ 2014, Monday	(1300 – 1430)	<ul style="list-style-type: none"> • All the problems have solved. • TSPi forms have completed. 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien
5/ 12/ 2014, Friday	(1430 – 1630)	<ul style="list-style-type: none"> • Final meeting for this course. • Any difficulties encountered solved. 	<ul style="list-style-type: none"> • Jadecrystal Tang Ming Mei • Lee Fui Yee • Nuratiqha Binti Abd Razak • Tan Sheu Yeu • Vernon Chien

5/ 12/ 2014, Friday	(1630 – 1800)	<ul style="list-style-type: none">• Compilation of the documentations: SRS, SDS and TSPi forms	<ul style="list-style-type: none">• JadeCrystal Tang Ming Mei• Lee Fui Yee• Nuratiqha Binti Abd Razak• Tan Sheu Yeu• Vernon Chien
------------------------	---------------	--	---