Software Project Lab 2

Thakurmar Jhuli: An Action Adventure Game

Software Requirement Analysis

Syed Ahmedul Kavi - BSSE 1017
Atkia Akila Karim Sobah - BSSE 1015

Supervised by:
Nadia Nahar,
Lecturer,
Institute of Information Technology,
University of Dhaka

Contents

1. INTRODUCTION

This chapter is a part of our software requirement specification for the project “Thakurmar Jhuli: An Action Adventure Game”. In this chapter we will focus on the intended audience for this project.

1.1 PURPOSE

This document briefly describes the Software Requirement Analysis of Thakurmar Jhuli: An Action Adventure Game. It contains the functional, non-functional and the supporting requirements and establishes a requirement’s baseline for the development of the system. The requirements contained in the SRS are independent, uniquely numbered and organized by topics. The SRS serves as an official means of communicating user requirements to the developer and provides a common reference point for both the developer team and the stakeholder community. The SRS will evolve over time as users and developers work together to validate, clarify and expand its contents.

1.2 INTENDED AUDIENCE

This SRS report is intended for several audiences including the customers as well as the project managers, designers, developers, and testers. The customer will use this SRS to verify that the developer team has created a product that is acceptable to the customer. The project managers of the developer team will use this SRS to plan milestones and a delivery date, and ensure that the developing team is on track during development of the system. The designers will use this SRS as a basis for creating the system’s design. The designers will continually refer back to this SRS to ensure that the system they are designing will fulfill the customer’s needs. The developers will use this SRS as a basis for developing the system’s functionality. The developers will link the requirements defined in this SRS to the software they create to ensure that they have created a software that will fulfill all of the customer’s documented requirements. The testers will use this SRS to derive test plans and test cases for each documented requirement. When portions of the software are complete, the testers will run their tests on that software to ensure that the software fulfills the requirements documented in this SRS. The testers will again run their tests on the entire system when it is complete and ensure that all requirements documented in this SRS have been fulfilled.

1.3 CONCLUSION

This analysis of the audience helped us to focus on the users who will be using our analysis. This overall document will help each and every person related to this project to have a better idea about the project.

2. INCEPTION OF Thakurmar Jhuli: An Action Adventure Game

In this chapter, the Inception part of the SRS will be discussed briefly.

2.1 INTRODUCTION

An action-adventure game can be defined as a game with a mix of elements from an action game and an adventure game, especially crucial elements like puzzles. Action-adventures require many of the same physical skills as action games, but also offer a storyline, numerous characters, an inventory system, dialogue, and other features of adventure games. They are faster-paced than pure adventure games, because they include both physical and conceptual challenges.Action-adventure games normally include a combination of complex story elements, which are displayed for players using audio and video. The story is heavily reliant upon the player character's movement, which triggers story events and thus affects the flow of the game. Some examples of action-adventure games include *The Legend of Zelda*, *God of War*, and *Tomb Raider* series.(Source: Wikipedia) Our game will use the mechanics of Action-adventure games and will incorporate the story and environment from Thakurmar Jhuli.

2.2 INCEPTION PROCEDURE

At the beginning of our project, we entered the inception stage. This stage includes how the project will be started and their scope and limitations . The main goal of this phase is toidentify the requirements ,demand and establish some sort of mutual understanding between the software team and the stakeholders of Thakurmar Jhuli: An Action Adventure Game. In order to make this phase effective, we followed some steps namely:

a. Identifying the client of our project

b. Icebreaking

c. Identifying the stakeholders of this game

d. Identifying the multiple viewpoints of stakeholders

2.2.1 Identifying the client of our project

At first, we identified the location from where we will start our expedition. Normally 6+ children will act as a stakeholder. But there are other things related to this as well. So we have to go through a systematic approach in order to identify all stakeholders.We have analyzed our requirements with the consent of them.

2.2.2 ICEBREAKING

Icebreaking refers to the fact that to diminish the communication barrier between you and the other person. It is a crucial part since it denotes the acceptance of our proposal. We started this phase by talking with the children and adults with context free languages. The children informally expressed their expectations.They are really excited.The behavior of children and adults were positive and they want this game.

2.2.3 IDENTIFYING THE STAKEHOLDERS OF A COURSE

Stakeholder refers to any person or group who will be affected directly or indirectly by the system. Stakeholders include end-users who interact with the system and everyone else in an organization who may be affected by its installation.We have few stakeholders only.The stakeholders of our system are given below:

- Player
- Developers
- Project Managers
- Sponsors

2.2.4 IDENTIFYING THE MULTIPLE VIEWPOINTS OF THE STAKEHOLDER

Different stakeholders expect different benefits from the system as every person has his own point of view. So, we have to recognize the requirements from multiple viewpoints. Different viewpoints of the stakeholders about the expected software are given below:

- Player View Point

  ➢ Smartphone (Android) based game.

  ➢ Fighting

  ➢ Level up

  ➢ Coin gathering

  ➢ Enjoy an immersive and interesting story

2.3 Conclusion

Our primary goal is to design a game which will revamp our childhood memory of Thakumar Jhuli stories. At the same time, where players can enjoy this adventurous game. As children are our main concern, the game will be designed in such a way that it won’t be complicated to play. The game will have simple but interesting features so that children can easily immerse into it.

Otherwise, it will not meet its goal.To make this software project successful, collaboration with the stakeholders i.e. the player is a main priority that what they want, how the software will work, how it can be more effective for all etc.

3. ELICITATION OF Thakurmar Jhuli: An Action Adventure Game

So far we have discussed the Inception phase of our project. Now we need to focus on the Elicitation phase. So this chapter specifies the Elicitation phase.

3.1 INTRODUCTION

Elicitation is a part of requirements engineering that is. We have faced many difficulties, like understanding the problems, making questions for players, getting appointments from them in spite of their busy schedule, making them understand the game. Despite not being easy to gather requirements within a very short time, we have overcome all problems in a systematic manner. We have done several meetings and finalized the requirements for our software.

3.2 ELICITING REQUIREMENTS

The Elicitation phase is mainly combining the elements for problem solving, elaboration, negotiation and specification. Without the collaboration of the stakeholder eliciting would have been really hard. We have finished the following tasks for eliciting requirements-

- Collaborative Solution
- Quality Function Deployment
- Usage Scenarios
- Elicitation work products

3.2.1 COLLABORATIVE SOLUTION

We have met with many players in the Inception phase . These meetings created an indecisive state and we could identify what the real problem is. To solve this problem, we have met with the stakeholders several times and came up with the solution which eventually helped us to elicit the requirements.

3.2.2 QUALITY FUNCTION DEPLOYMENT

QFD, Quality Function Deployment, is a technique that translates the needs of the customer into technical requirements for software. QFD mainly translates subjective quality criteria into objective ones that can be quantified and measured and which can then be used to design and manufacture the product. It is a methodology concentrating on maximizing customer satisfaction from the software engineering process. So, we have followed this methodology to identify the requirements for the project. The requirements identified successfully by the QFD are given below

3.2.2.1 NORMAL REQUIREMENTS

The requirements which are normally stated by the customer in various meetings are the normal requirements of a system, In general sense the objectives and goals that are stated for a product or system during meetings with the customer are normal requirements. The presence of these requirements fulfills customers' satisfaction. The normal requirements for our project are given below

- ➢ Simple and user-friendly interface
- ➢ Smartphone based application
- ➢ Easily accessible for all
- ➢ Having combat, platforming and adventure mechanism

3.2.2.2 EXPECTED REQUIREMENTS

These requirements are intrinsic to the system and may be so elementary that the customer does not explicitly state them. But the absence of these requirements would result in dissatisfaction. The expected requirements are given below

- ➢ Difficulty suitable for children and teen
- ➢ Adequate tutorials
- ➢ Option to pick difficulty

3.2.2.3 EXCITING REQUIREMENTS

These requirements are for features that go beyond the customer's expectations and prove to be very satisfying. These are the exciting features of our project

- ➢ Based on the stories of Thakurmar Jhuli
- ➢ Amazing graphics from fabled tales
- ➢ Magics and abilities gained through the gameplay

3.2.3 USAGE SCENARIO

The game revamps our childhood memory of Thakurmar Jhuli stories with combat, platforming, strategy and adventure. The game is mainly intended for 6+ children. It will be available for Android and Windows Operating Systems. The game features characters from thakurmar jhuli like Nilkomol, Shakchunni, Duorani etc.

Plot

Once there was a kingdom named Indrapur. The king had two queens, Suorani and Duorani. Suorani was a kindhearted woman while Duorani used to conspire against many people. Suorani had a son named Nilkomol and Duorani had a son named Lalkomol. Our story follows Nilkomol.

One day Duorani invited Nilkomol to her palace and spoke to him about an enchanted forest where fairies and monsters used to live. The forest was beautiful but required courage for anyone to venture into it. In that forest there was a specific location where a horrific monster named Shakchunni lived. She was hideous and scary and was known by the name Reaper of the forest. She guarded a precious magical necklace. If a hero courageous enough could bring the necklace to this kingdom, it would bring blessing for the whole kingdom. So Nilkomol starts his quest for the magical necklace in this mysterious woods known as Rajendra Forest.

Level 1(Rajendra Forest): Nilkomol enters the forest and encounters several of its residents. Bangoma and Bengomi, two talking birds guide him towards his goal. Nilkomol gathers coins left out by fallen warriors and also finds a merchant among these woods who will sell him various items including potions, arrows etc.

As the sun sets down, Nilkomol finds himself in the darker parts of the forest where he encounters more ferocious monsters. Finally he reaches Shakchunni's lair and slays her. Nilkomol retrieves the magical necklace.

But instead of blessing, a catastrophic result occurs. The forest is set on fire as if by magical force. Nilkomol runs in forward direction to find shelter from this fire. In this tense state, he finds a pond and collapses beneath it.

A fairy lived in the pond. She appears before Nilkomol as he regains consciousness. The fairy grants him power to breathe under the pond. She also hints that Nilkomol may find the solution to forest fire which extended towards his kingdom under that pond. Nilkomol dives in it.

Level 2(The underwater kingdom): Beneath the magical pond, Nilkomol finds a vast underwater kingdom. He reaches the palace fighting underwater monsters. In front of the entrance of the palace, Nilkomol finds Lalkomol, his step brother who has come to slay him by the king's order since the king thinks the whole kingdom is under ruin due to his mistake persuaded by Duorani's conspiracy.

Nilkomol is confused and is forced to fight. After sometimes being wounded by Nilkomol's weapon, Lalkomol shows his true form. He is a descendant of Rakshasha. Nilkomol is able to slay his demonic brother.

Nilkomol enters the palace and finds a lone princess with some of her subordinates and a wise sage. Nilkomol learns that a witch killed all of her family by magic. Now the witch resides as a queen of Indrapur. Nilkomol learns of a special sword to kill this mighty witch from the wise sage.

Level 3(Burning Rajendra Forest): Nilkomol finds the sword hidden in the burning Rajendra forest. The fairy guarding the sword shows him the way to a beautiful oasis. A horse was drinking water there. It was no ordinary horse. It was Pangkhiraj, the horse with wings. Nilkomol approaches the horse and tames it. Nilkomol rides Pangkhiraj and rushes in the sky towards his endangered kingdom.

Level 4(Ruined Indrapur): He finds a nightmarish Indrapur, swarming with devilspawns. Some houses were on fire. Nilkomol enters the ruined palace and encounters Duorani. She is slain by the magical sword and the curse lifts. Demons leave Indrapur.

Nilkomol becomes the king and rebuilds his kingdom. He marries the princess from an underwater palace. They live happily ever after.

Gameplay

User perspective: There will be a main menu with 3 buttons: Play Game, Options and Quit. The player can start the game by clicking the Play Game button. The game will have cutscenes to convey the story. When the game starts the player will be in control of Nilkomol. Bengoma and Bengomi, two talking birds will initiate a tutorial by guiding the player about the controls. After learning the controls, the game will actually start and the player will be able to face enemies, collect treasures, visit merchants etc. The player will be able to talk with NPCs (Non Playable Characters) and gain various information and items from them. Gradually through the course of the game player will learn new abilities which will allow Nilkomol to progress. A Player Rank or High score system will be present which will be calculated based on playtime, coin collected, enemy killed and damage taken.

Developer perspective: 3 bars control player's status: Red, Green, and Blue. The red bar governs health which is decreased by taking damage and replenished by heal magic. Green bar governs stamina which is drained through each hit (attack) and replenished automatically. The blue bar governs focus which can be used to cast magic. It is accumulated by attacking and taking hits.

The player can move horizontally, jump, attack, block. Some magical powers can be gained and used. Player earns experience and coins by defeating enemies. Experience accumulates to gain skill points which can be used to learn skills.

Coins can be traded with merchants for various items.

The game contains fighting and platforming elements. Fighting extends to block, magic, dodge etc. and platforming extends jumping at correct time and various minigames.The game can be saved and save data is produced based on position of character, number of remaining enemies and current profile of the player.

4. SCENARIO BASED MODELING OF Thakurmar Jhuli: An Action Adventure Game

This chapter describes the Scenario Based Model for the "Thakurmar Jhuli: An Action Adventure Game "

4.1 INTRODUCTION

Although the success of a computer-based system or product is measured in many ways, user satisfaction resides at the top of the list. If we understand how end users (and other actors) want to interact with a system, our software team will be better able to properly characterize requirements and build meaningful analysis and design models. Hence, requirements modeling begins with the creation of scenarios in the form of Use Cases, activity diagrams and swim lane diagrams.

4.2 DEFINITION OF USE CASE

Use case defines the stylized story about how an end user interacts with the system under a specific set of circumstances. A Use Case diagram simply describes a story using corresponding actors who perform important roles in the story and makes the story understandable for the users. The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function

and behavior that is to be described. Actors represent the roles that people play as the system operators.

Primary Actors:The system has one primary actor: player.

Secondary Actor: The system also acts as an internal actor in some cases. Therefore, it is the secondary actor of our system.

4.3.1 LEVEL-0 USE CASE DIAGRAM - Thakurmar Jhuli: An Action Adventure Game

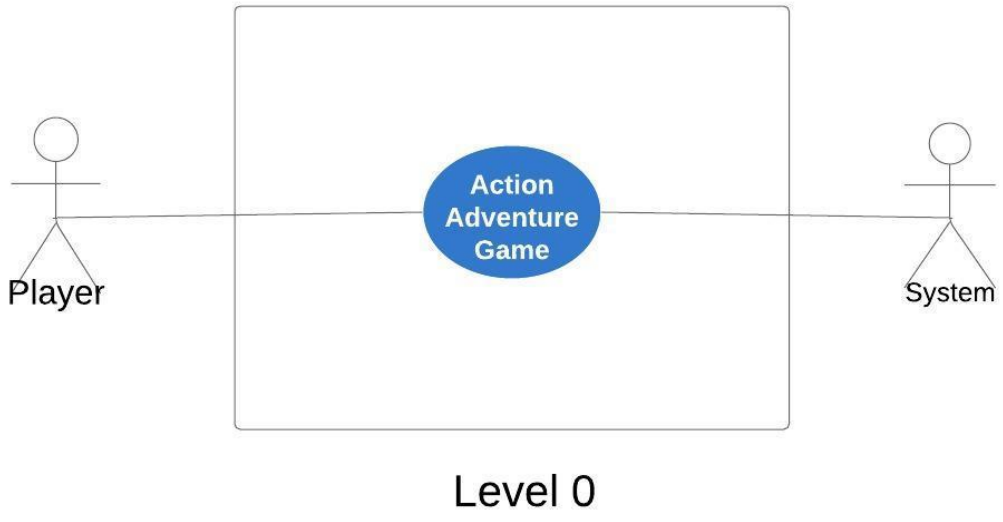
Level: 0

Use case Name: Thakurmar Jhuli: An Action Adventure Game

Use case Id: 0

Primary actor:player

Secondary actor:System



Level 0

Fig 1: level 0

Description: This use case shows the low level interaction between system and actors.

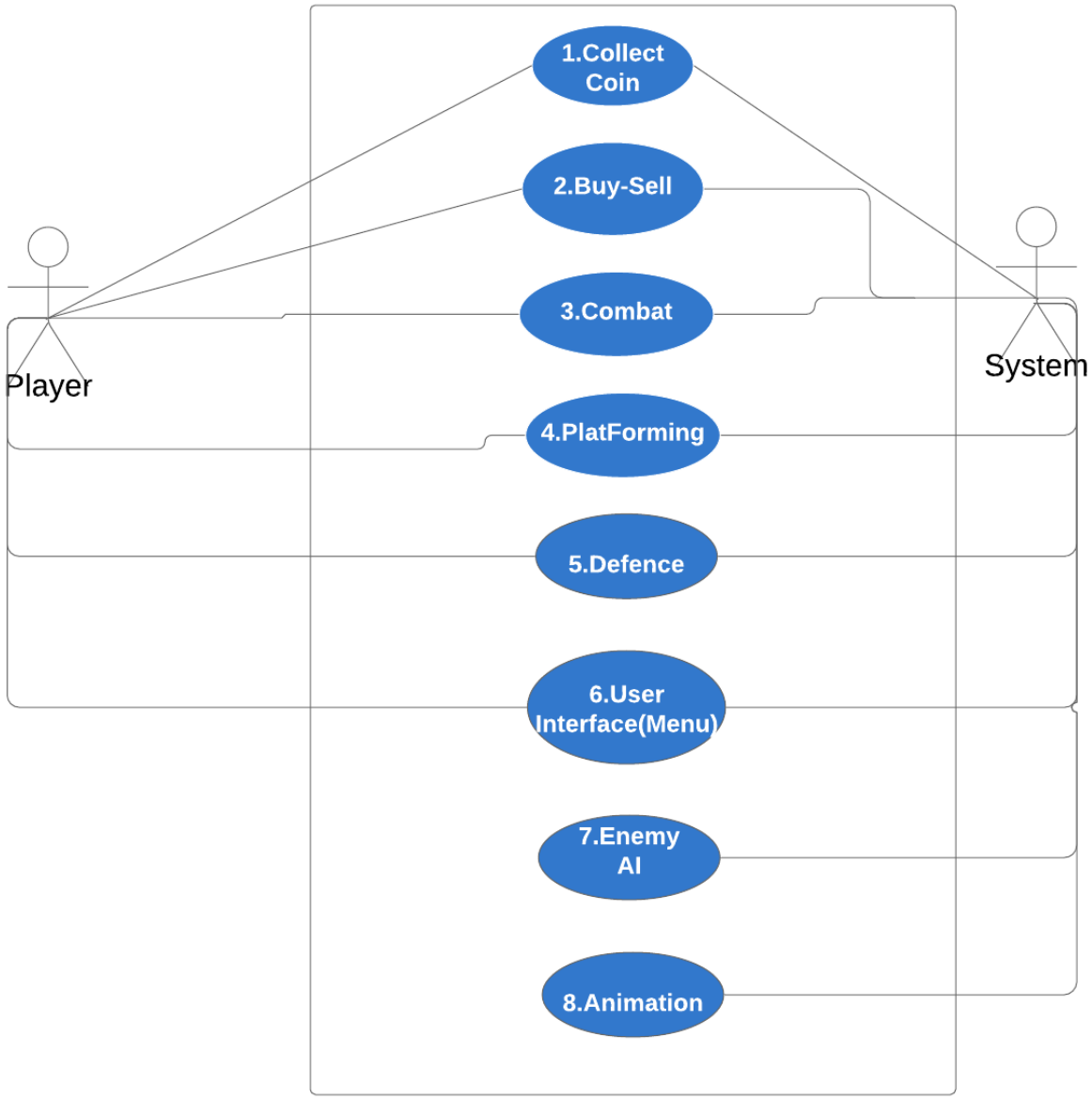
4.3.2 LEVEL-1 USE CASE DIAGRAM- SUBSYSTEM

Level: 1:

Use case Name: SUB SYSTEM

Use case Id: 1

Primary actor: player



Level 1

Fig 2: level 1

Description: From this level all the subsystems of the proposed main system and connectivity of those subsystems through actors has been explicit. From this level interaction between actors and subsystems will be clearer. Here, the whole system is divided into eight subsystems.

4.3.2.1 LEVEL-1.1 USE CASE DIAGRAM- Collect Coin

Use case Name: Collect Coin

Primary actor: player,system

Action: Player will collect coin

Reply: System will store coin

4.3.2.2 LEVEL-1.2 USE CASE DIAGRAM- Buy-Sell

Use case Name: Buy-Sell

Primary actor: player,system

Action: Player will buy using coin/sell items to get coin

Reply: System will track the number of coins

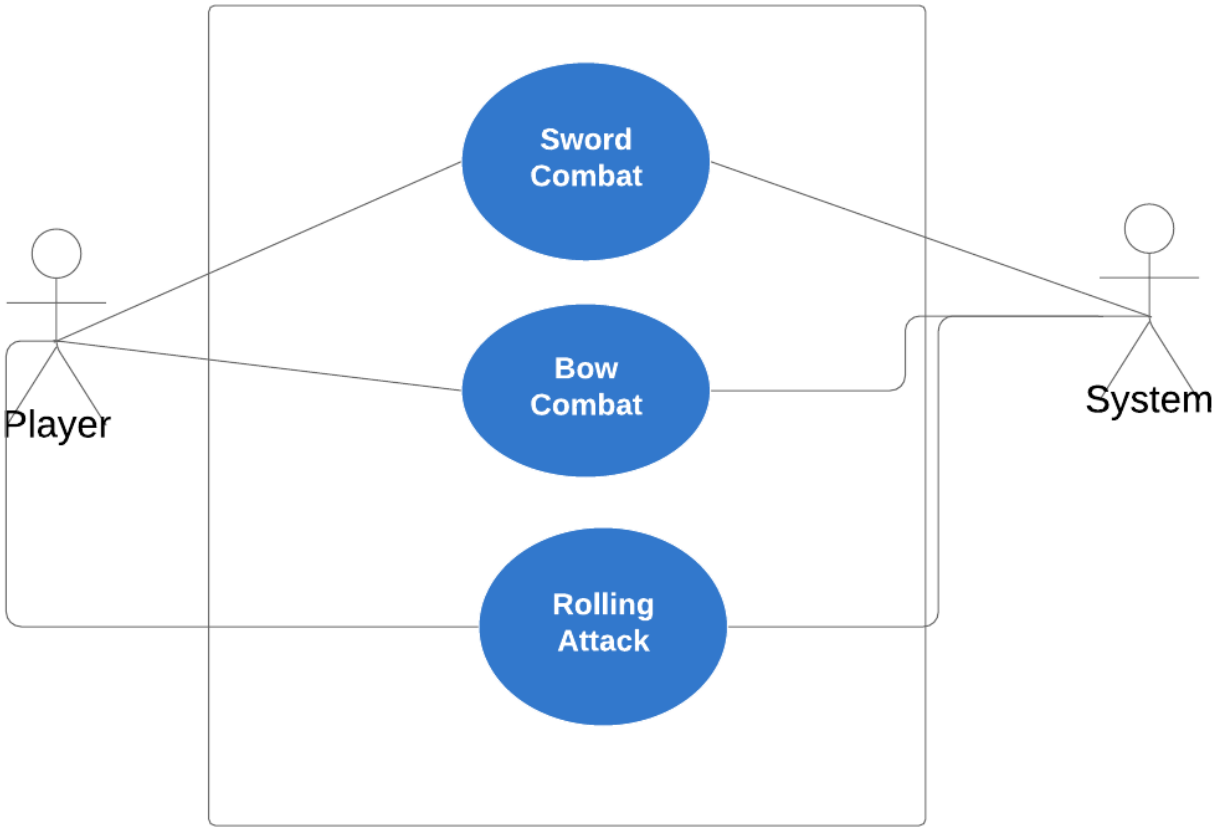
4.3.2.3 LEVEL-1.3 USE CASE DIAGRAM- Combat

Level: 1.3:

Use case Name: Combat

Use case Id: 1.3

Primary actor:player,system



Level 1.3

A: Player presses combat button with sword equipped
R: The player will do attack animation, enemy will do hurt animation, enemy will take fixed damage

A: Player presses combat button with bow equipped
R: Arrow goes based on trajectory, and hits enemy if trajectory is accurate, enemy will play hurt animation

A: Player presses attack just after roll
R: Rolling attack executes with fixed multiplier

Fig 3: level 1.3

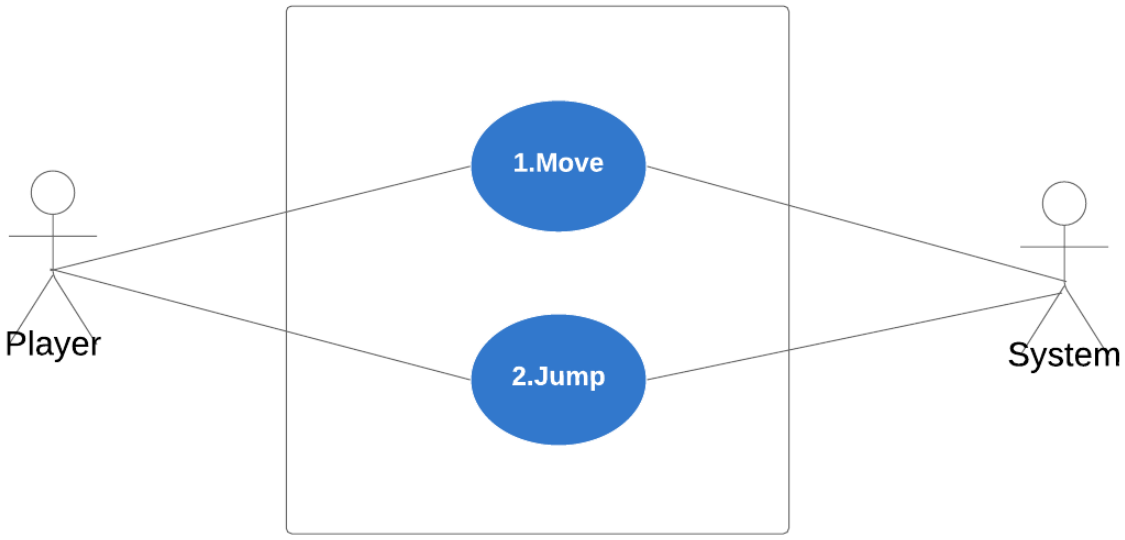
4.3.2.4 LEVEL-1.4 USE CASE DIAGRAM- Platforming

Level: 1.4:

Use case Name: Platforming

Use case Id: 1.4

Primary actor:player,system



Level 1.4

Fig 3: level 1.4

A: Player tilts touch screen joypad

R: Player moves

A: Player presses jump button

R: Player character jumps

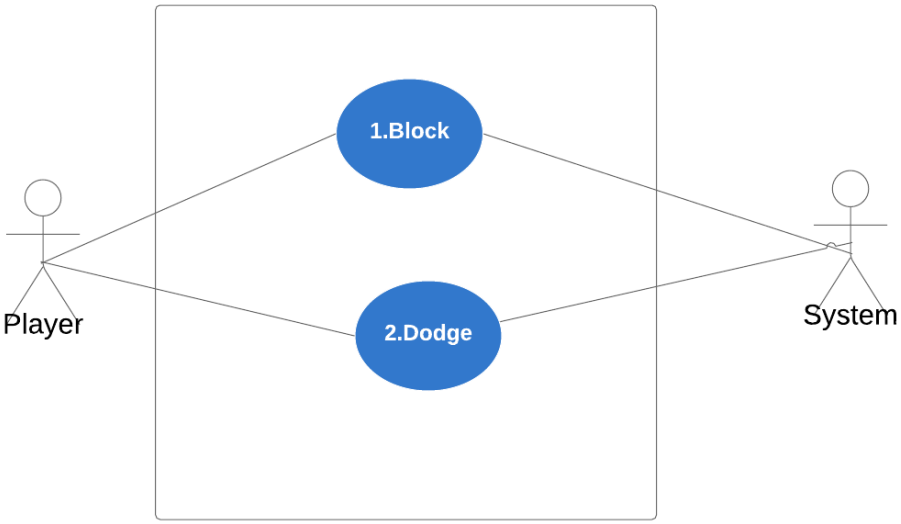
4.3.2.5 LEVEL-1.5 USE CASE DIAGRAM- Defence

Level: 1.5:

Use case Name: Defence

Use case Id: 1.5

Primary actor:player,system



Level 1.5

A: Player presses on screen block button
R: Hero goes into block state. Any damage from enemy is blocked, consumes stamina

A: Player presses on screen dodge button
R: Hero rolls and avoids any damage from enemy, consumes stamina

Fig 4: level 1.5

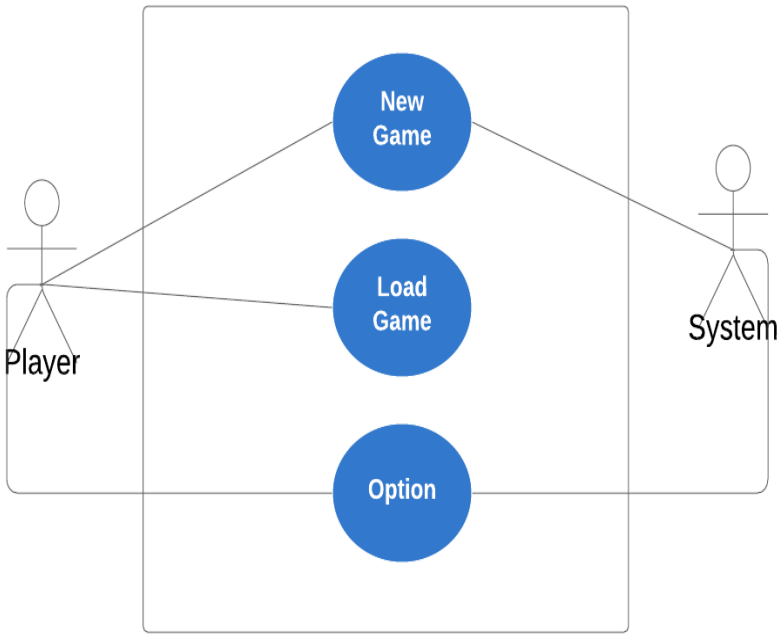
4.3.2.6 LEVEL-1.6 USE CASE DIAGRAM- User Interface

Level: 1.6:

Use case Name: UI

Use case Id: 1.6

Primary actor:player,system



Level 1.6

A: Player clicks the New Game button from Menu User Interface
R: A new game is started where story starts from beginning

A: Player clicks Load Game button from Menu User Interface
R: Loads a previously saves game

A: Player clicks the Option button from Menu User Interface
R: Brings player to a configuration window where the player can configure settings of the game according to his/her ease

Fig 6: level 1.6

4.3.2.7 LEVEL-1.7 USE CASE DIAGRAM- Enemy AI

Level: 1.7:

Use case Name: Enemy AI

Use case Id: 1.7

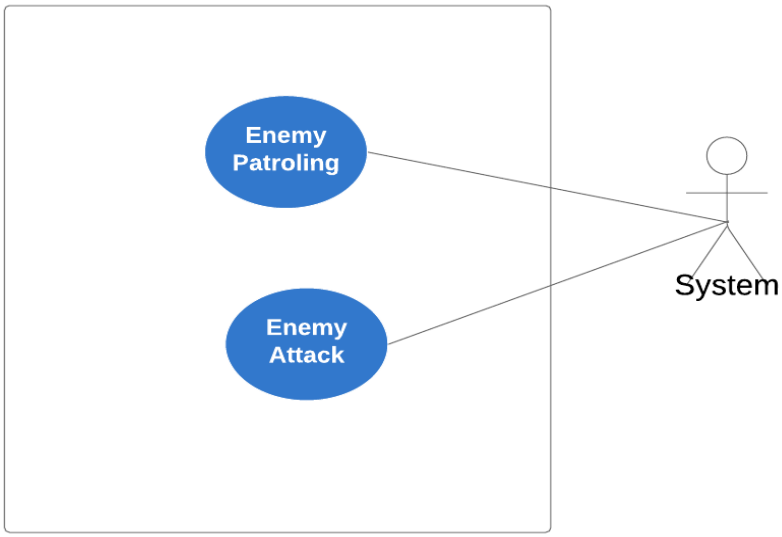
Primary actor:system

Fig 6: level 1.7

4.3.2.8 LEVEL-1.8 USE CASE DIAGRAM- Animation

Level: 1.8:

Use case Name: Animation



Level 1.7

A: Player is not within enemy range
R: Enemy will patrol within a given radius

A: Player comes within enemy range
R: Enemy attacks player at a constant rate and damage, enemy attack animation and player hurt animation plays, player recieves damage if not blocked

Use case Id: 1.8

Primary actor:system

A: Certain activity plays out

R: Appropriate animation plays

4.4 Activity Diagram Of Thakurmar Jhuli: An Action Adventure Game

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

4.4.1 Authentication Activity name: Collect Coin

Activity id: 4.4.1

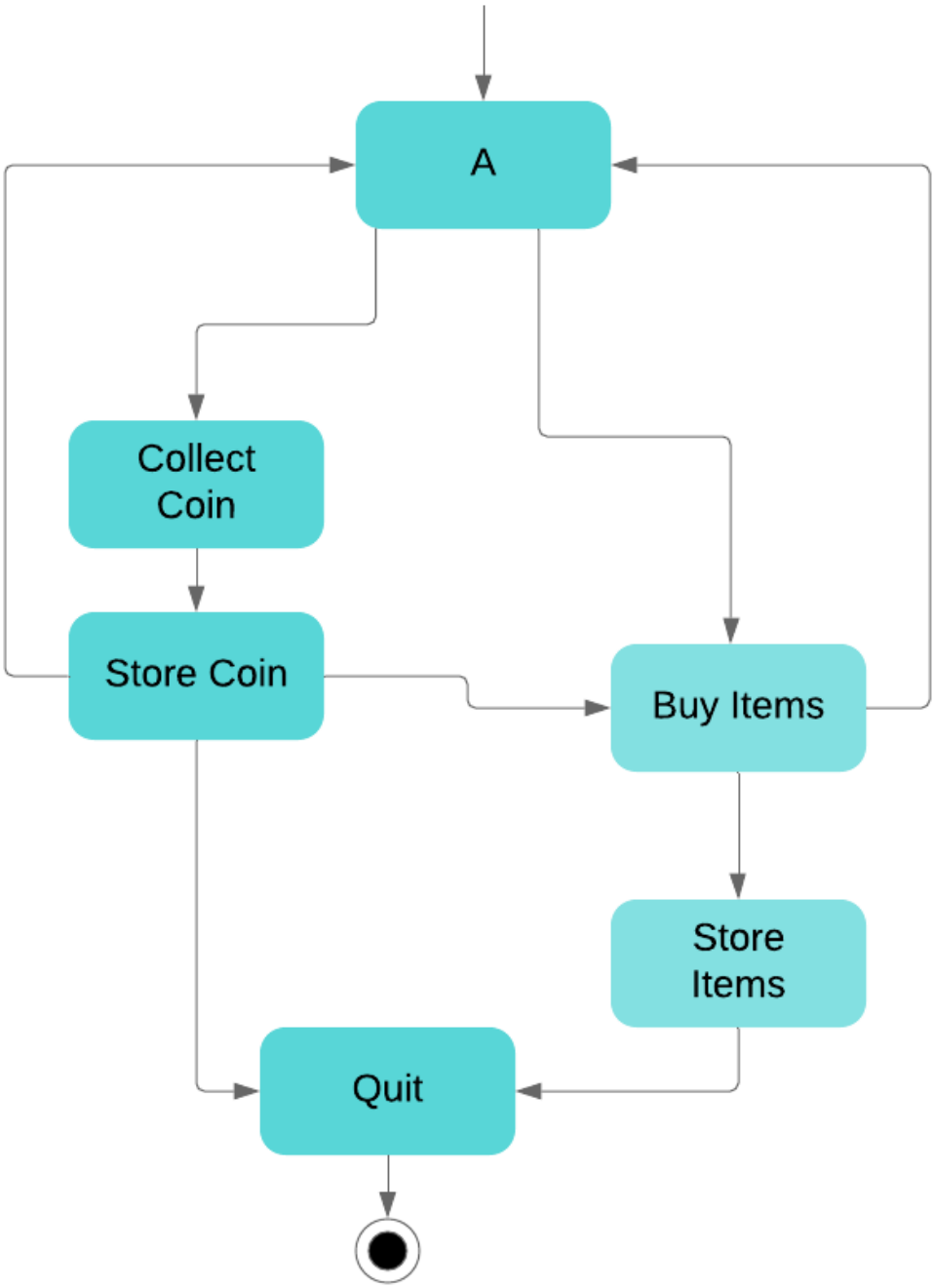


Fig 7:Collect Coin

4.4.2 Authentication Activity name: Combat

Activity id: 4.4.2

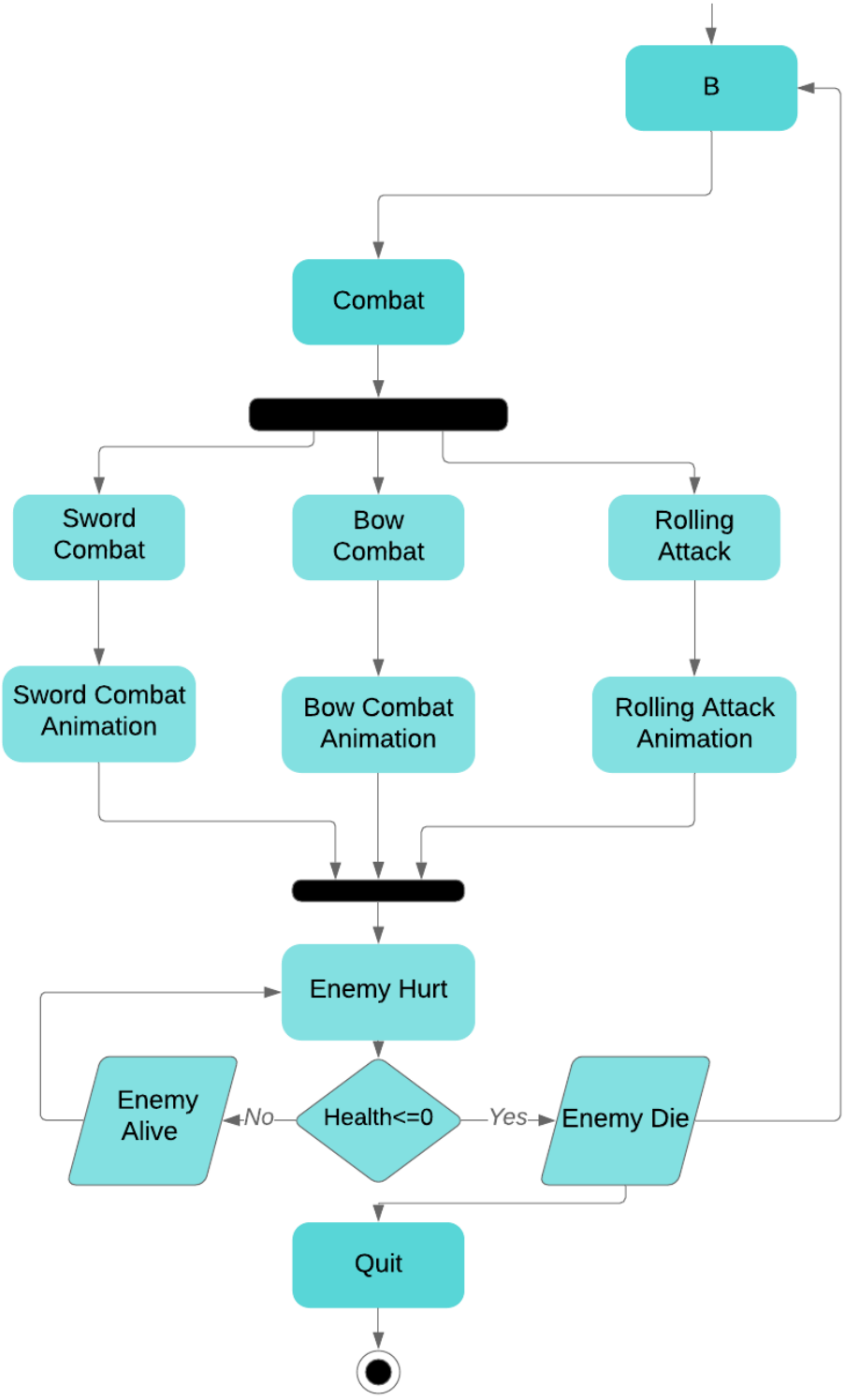


Fig 8:Combat

4.4.3 Authentication Activity name: Platforming

Activity id: 4.4.3

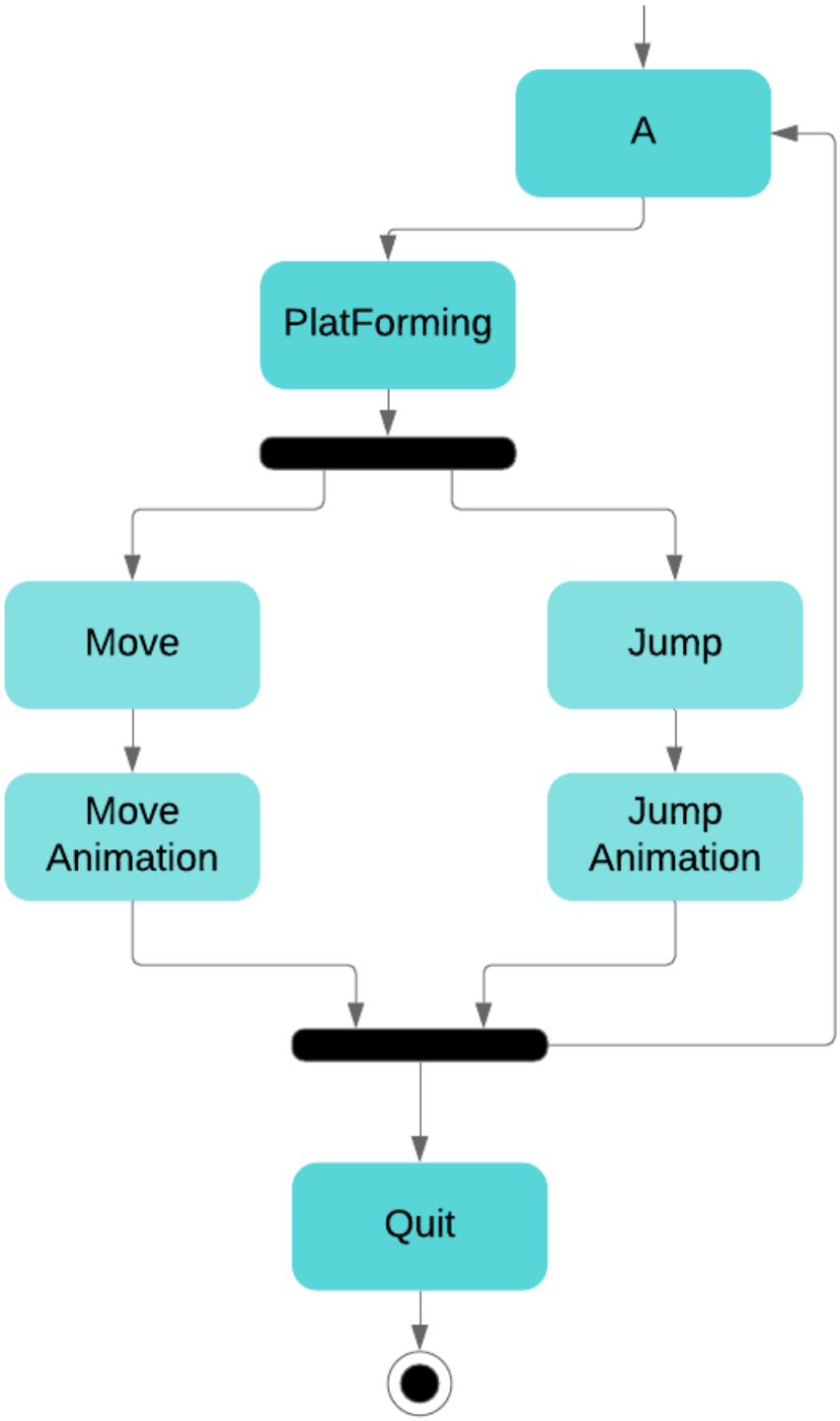
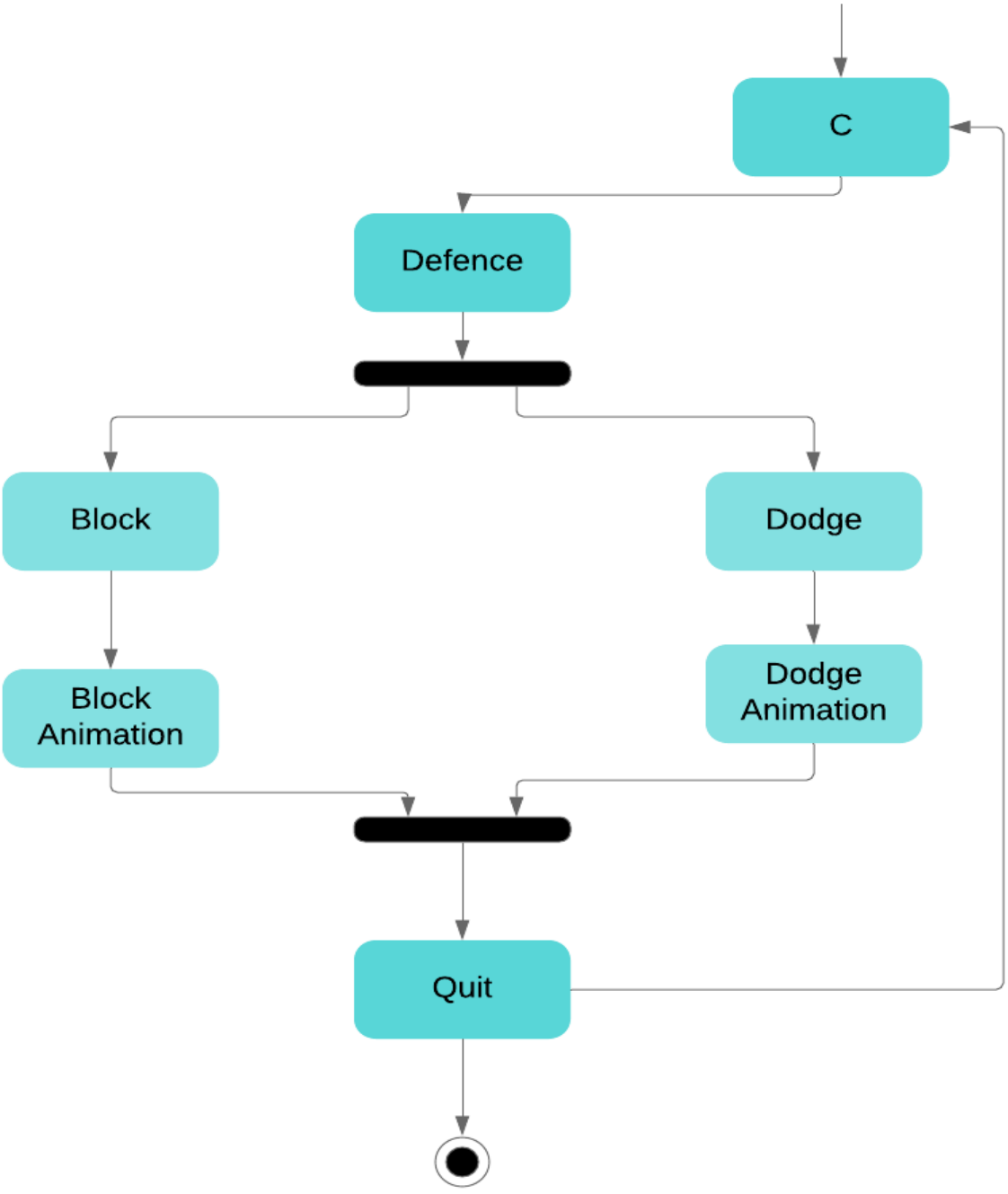


Fig 9:Platforming

4.4.4 Authentication Activity name: Defence

Activity id: 4.4.4



Flg 10:Defence

4.4.5 Authentication Activity name: User Interface

Activity id: 4.4.5

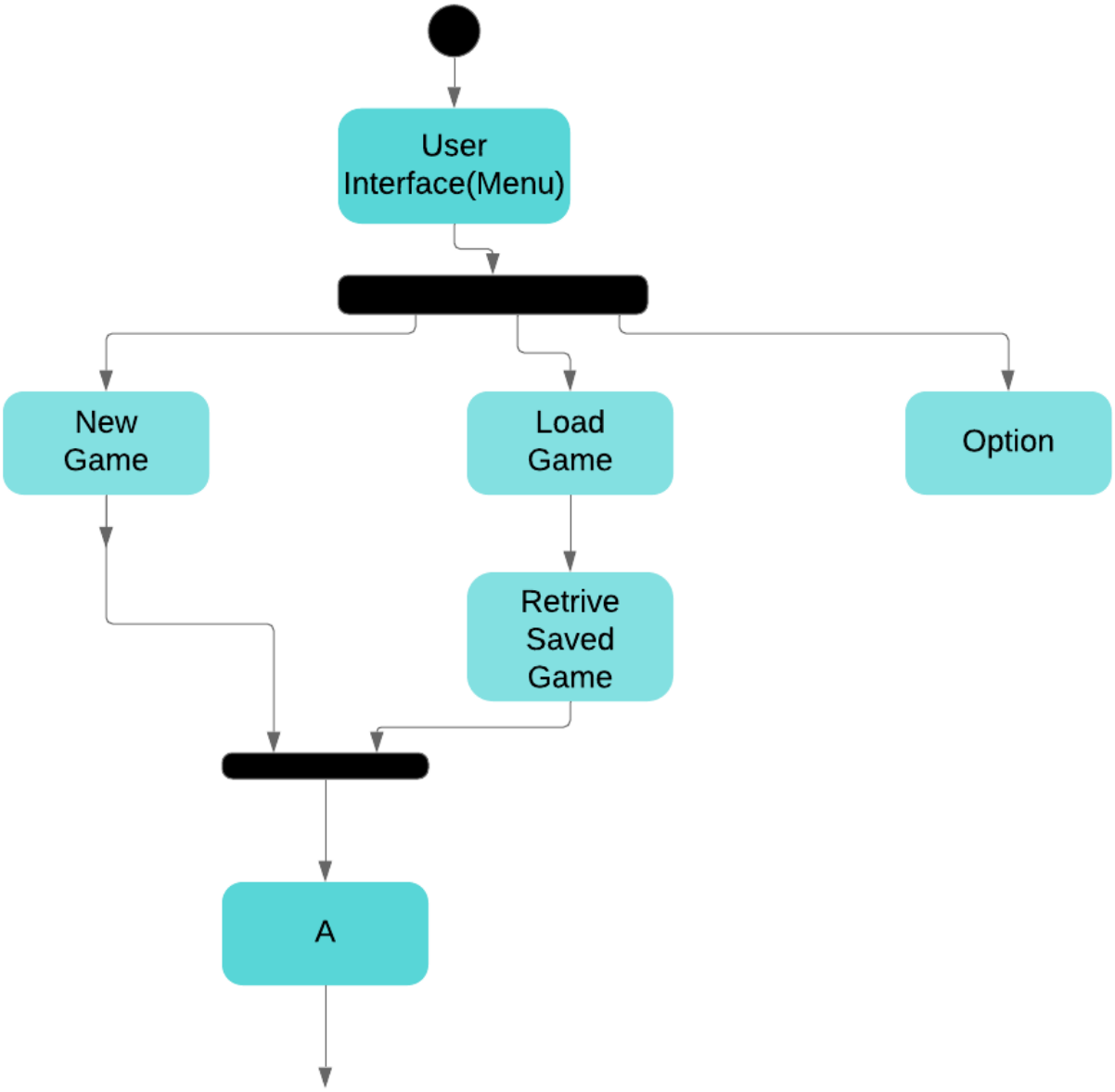


Fig 11:User Interface

4.4.6 Authentication Activity name: Enemy AI

Activity id: 4.4.6

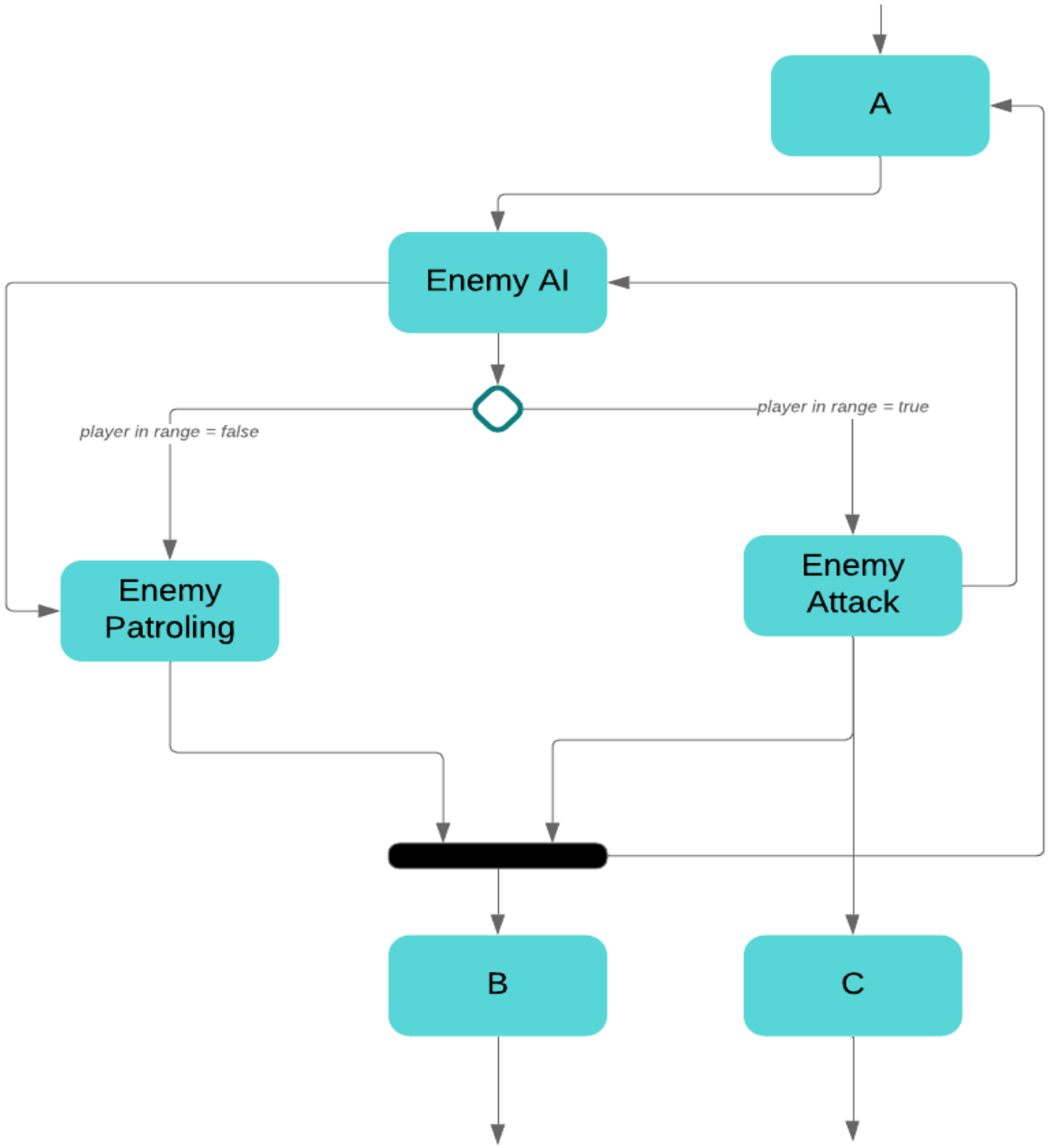


Fig 12:Enemy AI

4.5 Swimlane Diagram of Thakurmar Jhuli: An Action Adventure Game

Collect Coin

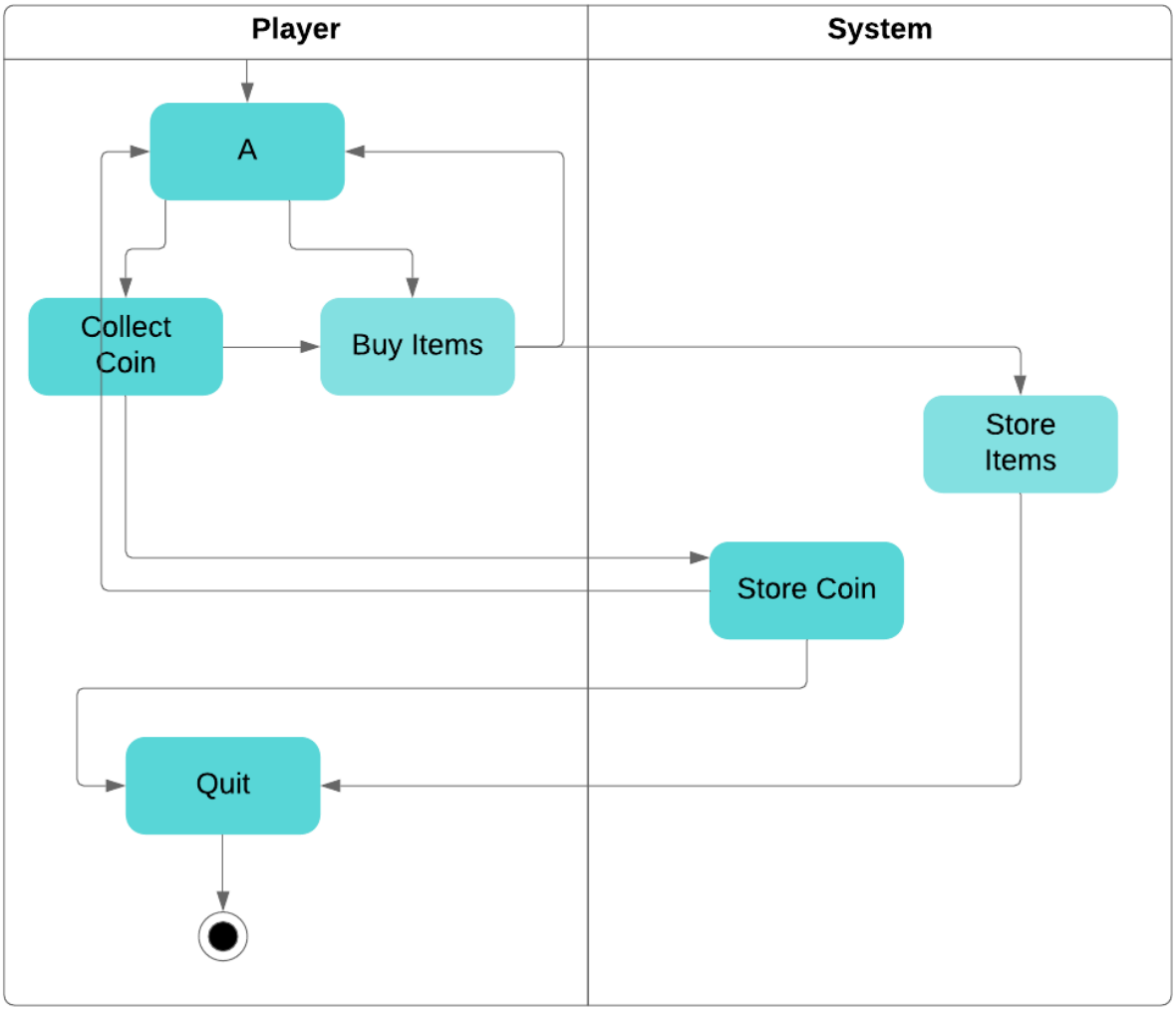


Fig 13: Swimlane Diagram of Collect Co**in**

Description: Figure 13 shows the swimlane diagram of Collecting coin.

Combat

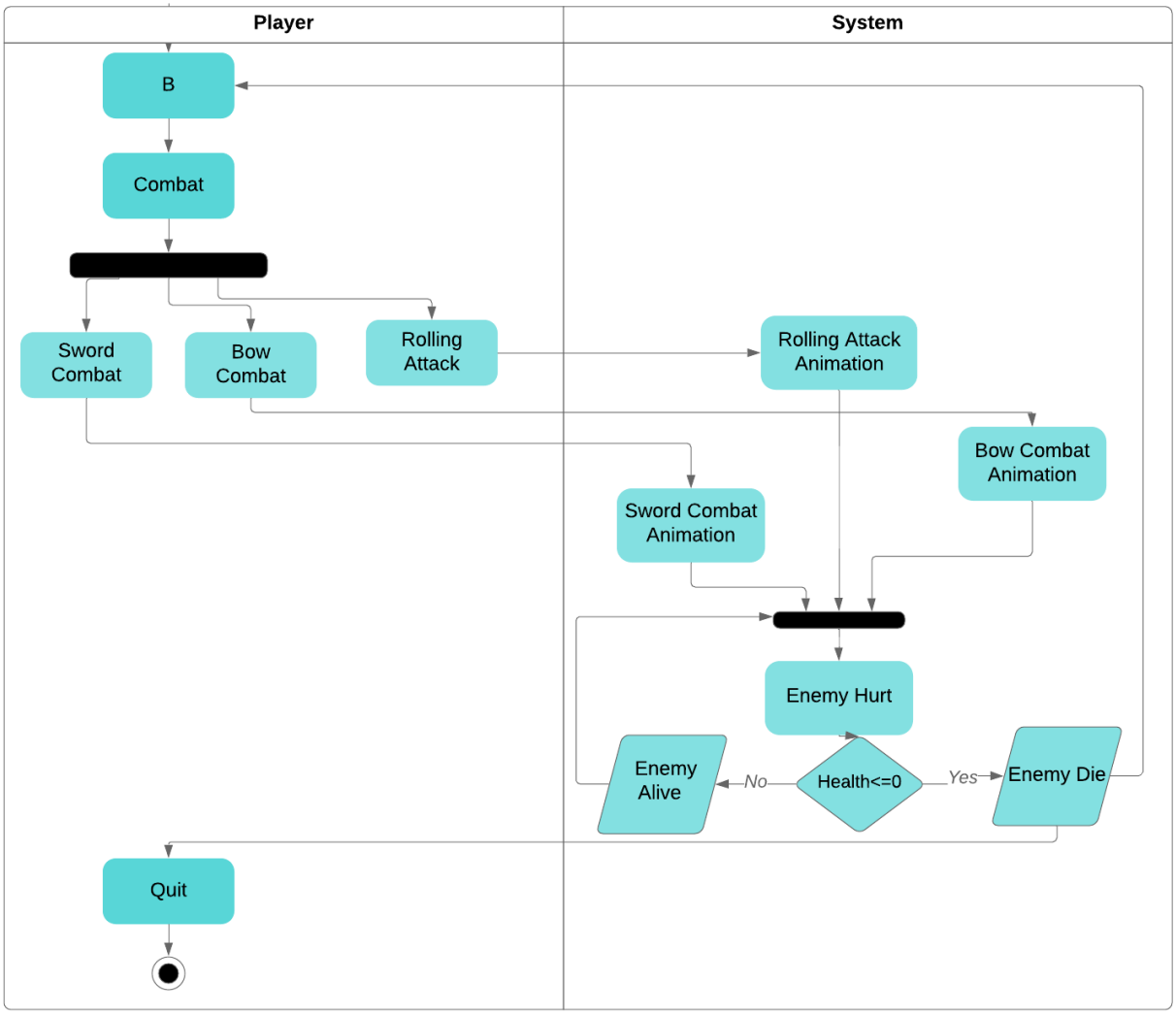


Fig 14: Swimlane diagram of Combat

Description: Figure 14 shows the swimlane diagram of Combat.

Defence

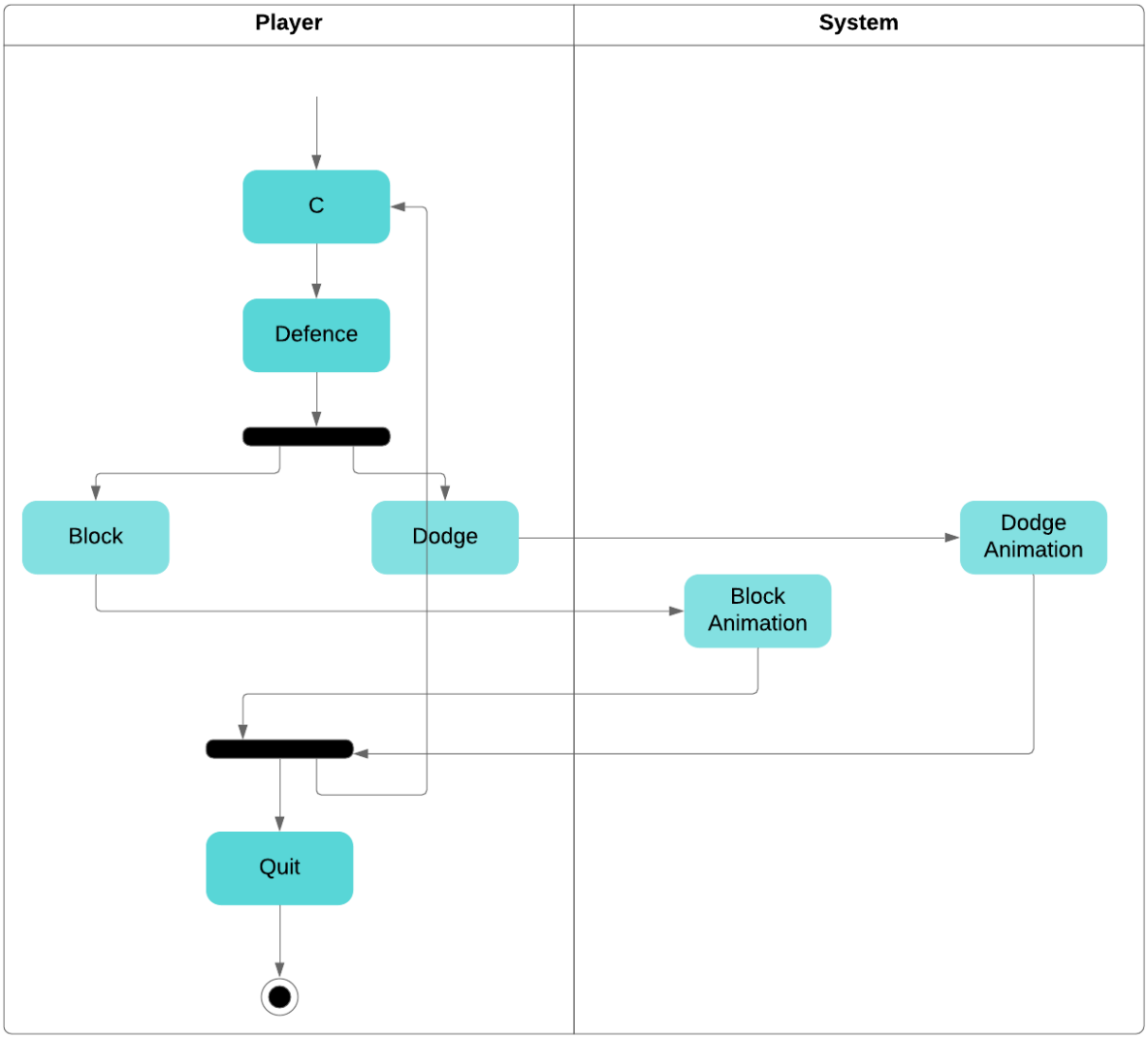


Fig 15: Swimlane Diagram of Defence

Description: Figure 15 shows the swimlane diagram of Defence.

Platforming

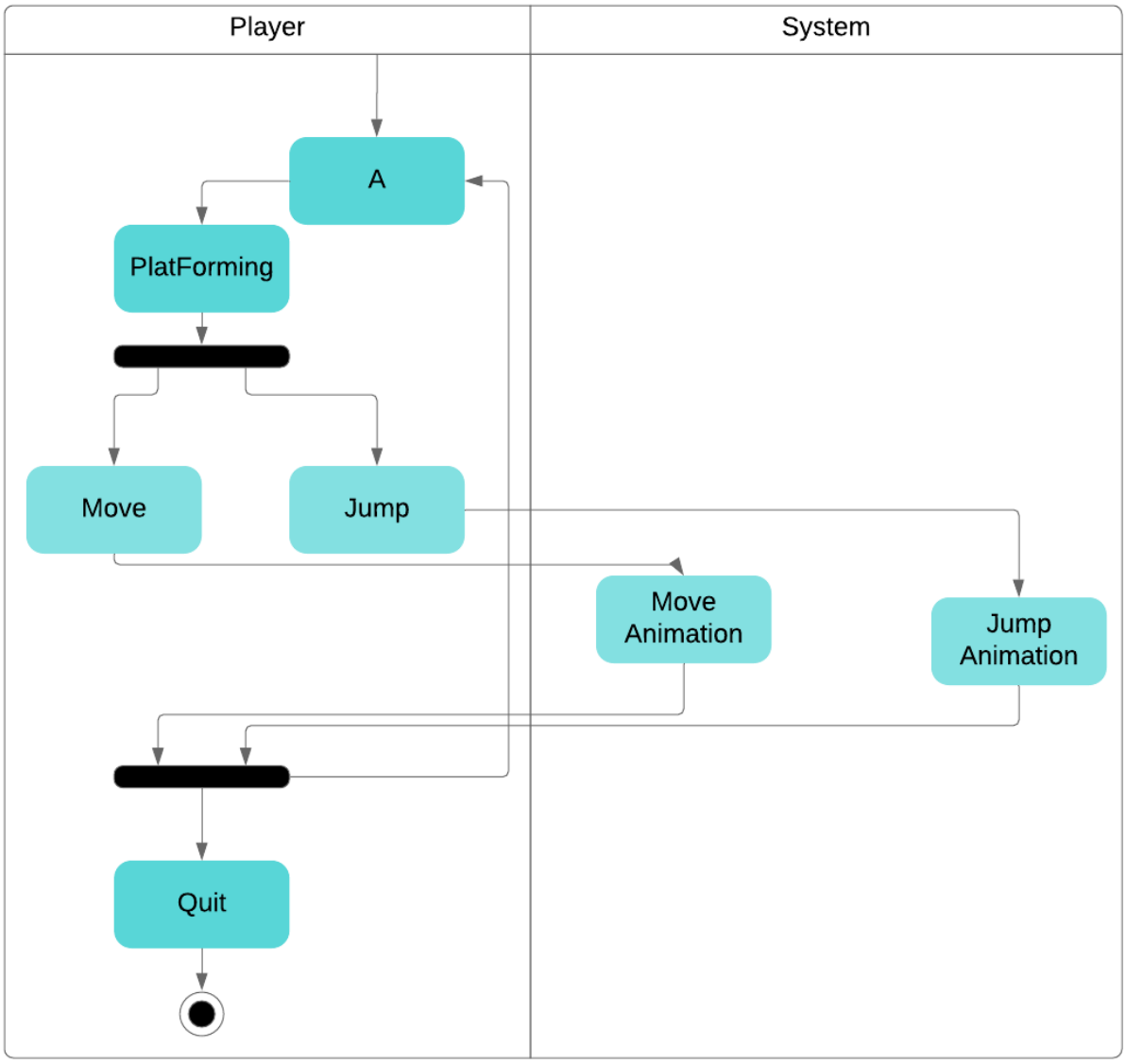


Fig 16: Swimlane Diagram of Platforming

Description: Figure 16 shows the swimlane diagram of Platforming.

User Interface

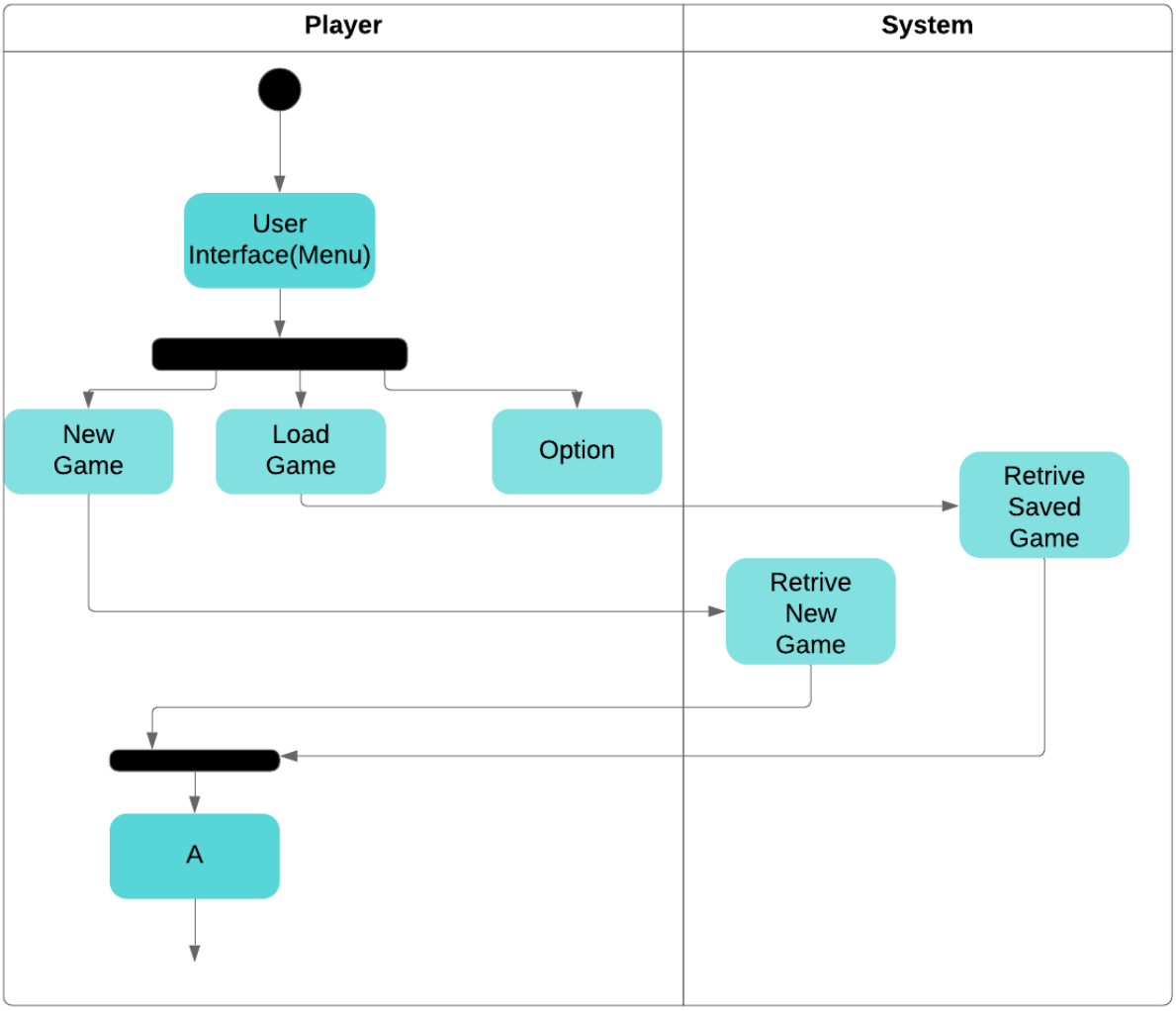


Fig 17: Swimlane Diagram of User Interface

Description: Figure 17 shows the swimlane diagram of User Interface.

Enemy AI

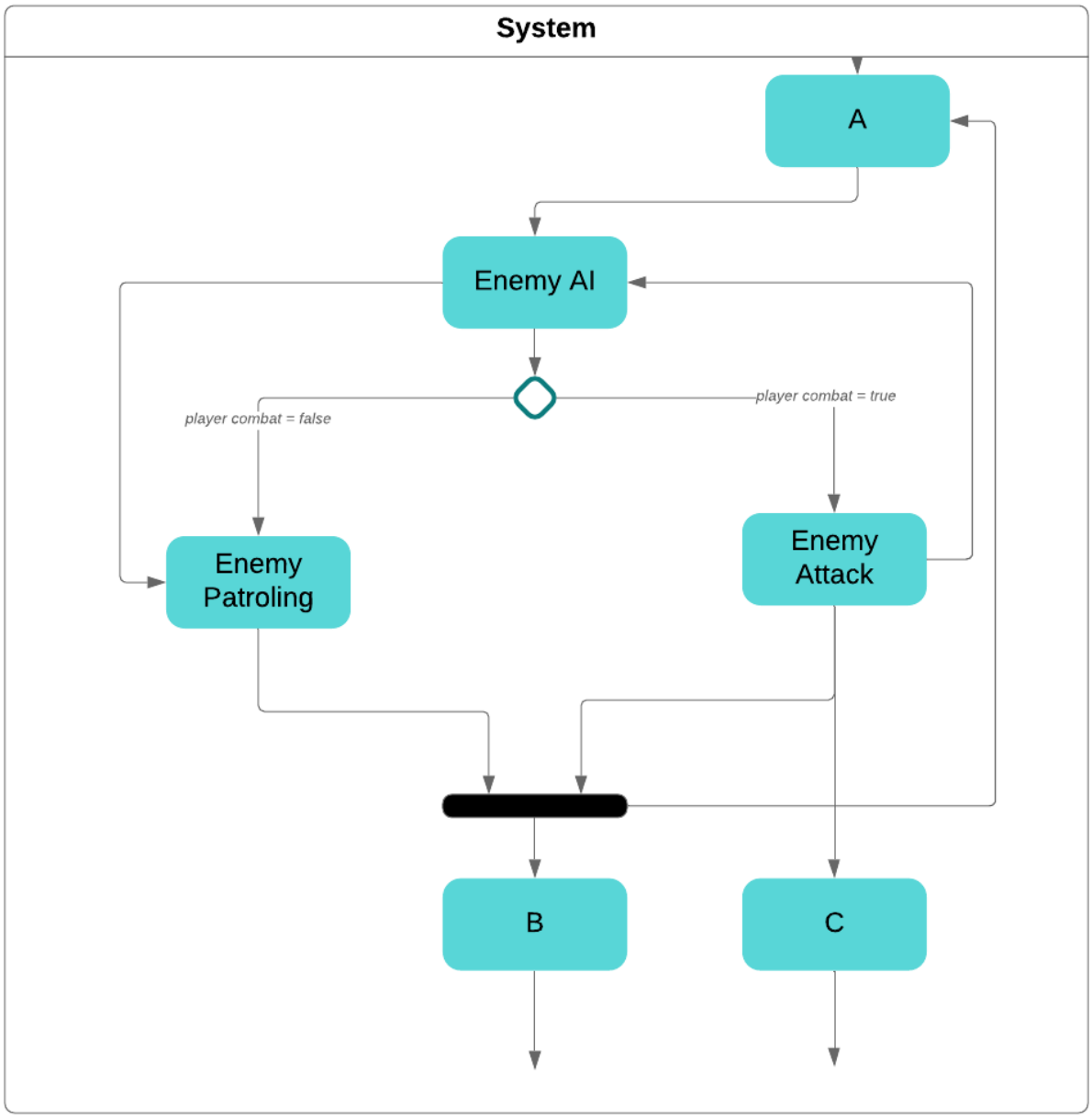


Fig 18: Swimlane Diagram of Enemy AI

Description: Figure 18 shows the swimlane diagram of Enemy AI.

5. CLASS-BASED MODELING FOR Thakurmar Jhuli: An Action Adventure Game

In this part we have discussed the classes along with their attributes and methods of our Thakurmar Jhuli: An Action Adventure Game**.**

5.1 CLASS BASED MODELING CONCEPT

Class-based Modeling represents the object. The system manipulates the operations. The elements of the class based model consist of classes and objects, attributes, operations, class – responsibility - collaborator (CRS) models.

5.2 GENERAL CLASSIFICATION

To identify the potential classes, we have first selected the nouns from the solution space of the story. We have used seven general characteristics for this. The seven general characteristics are as follows :

1. External entities
2. Things
3. Events
4. Roles
5. Organizational units
6. Places
7. Structures

Serial no.	Name(Noun)	General Classification
1.	Shakchunni	*
2.	Bangoma	*

3.	Bangomi	*
4.	Coin	2
5.	Merchant	4
6.	Potions	2
7.	Arrows	2
8.	Nilkomol	2,3,4
9.	Monster	4
10.	Megical Necklace	2
11.	Fairy	2,4
12.	Lalkomol	*
13.	Wise Sage	*
14.	Raksha	*
15.	Magic	2
16.	Rajendra Forest	*
17.	Pongkhiraj	*
18.	Player Attack	2,3
19.	Enemy	2,3,4
20.	Duorani	*

21.	Indrapur	*
22.	Treasure	2
23.	NPC	2,3,6
24.	Red Bar	2
25.	Green Bar	2
26.	Blue Bar	2
27.	Taking Damage	2
28.	Healing magic	2
29.	Casting magic	2
30.	Jumping	2
31.	Blocking	2
32.	Magical Power	2
33.	Platforming	2,3
34.	Player Combat	2,3
35.	Player Health Bar	2,3,6
36.	Player Health	2,3,6
37.	Enemy Health Bar	2,3,6

5.3 SELECTION CRITERIA

The potential classes were then selected as classes by six Selection Criteria. A potential class becomes a class when it fulfills all six characteristics.

1. Retain information

2. Needed services

3. Multiple attributes

4. Common attributes

5. Common operations

6. Essential requirements.

No	Noun	Selection Criteria
1.	Nilkomol	1,2,3,4,5
2.	Enemy	1,2,3,4,5
3.	Player Movement	1,2,3
4.	Player Combat	1,2,3,4,5
5.	Enemy Combat	1,2,3
6.	Enemy Movement	1,2,3
7.	Player HealthBar	1,2,3
8.	Player Health	1,2,3

9.	Enemy HealthBar	1,2,3
10.	NPC	1,2,3,5
11.	Merchant	1,2,3,5
12.	Magic Power	1,3,4,5
13.	Coin	1
14.	Potions	1
15.	Fairy	1

5.4 Associate Noun and Verb identification

We will now identify the nouns and verbs associated with the potential classes to find out the attributes and methods of each class :

No	Potential Class	Noun	Verb
1.	Player	Player	
2.	Enemy	Enemy	
3.	Player Movement	player	Move,Jump
4.	Player Combat	Sword,bow,Arrow	Attack,Take Damage
5.	Enemy Combat	player,	Attack,Take Damage
6.	Enemy Movement	Enemy	Move
7.	Player HealthBar	Player	HandleHealth Change,Awake

8.	Enemy HealthBar	Enemy	SetHealth
9.	NPC	Fairy,Bangoma,Bangomi	Offer item, Talk
10.	Merchant	Coin,Potion	Trading
11.	Magic	Potion,Healing Magic,Casting Magic	Heal,Cast

5.5 Identify Attributes

We have identified our potential classes, So, now we have identified the classes

No	Class	Attribute
1.	Player	Player_healthBar, Player_Movement, Player_Combat
2.	Enemy	Enemy_healthBar, Enemy_Movement, Enemy_Combat,
3.	Player Movement	Horizontal_move,jump, runSpeed
4.	Player Attack	attackRange,attackDamage,attackPoint
5.	Enemy Combat	PlayerHealth,nextTime,attackTimer
6.	Enemy Movement	Radius,PatrolSpeed,relativeX,atLeft
7.	Player HealthBar	Player HealthBar,Max Health,CurrentHealth
8.	Enemy HealthBar	Enemy HealthBar,Max Health,CurrentHealth
9.	NPC	Item,dialogue
10.	Merchant	Item_Price_Mapping
11.	Magic Power	Focus_cost, damage

5.6 Identify Methods of potential Classes

Now , we are going to identify the methods of our potential classes

No	Class	Method
1.	Player	Player()
2.	Enemy	Enemy()
3.	Player Movement	Move(), Jump()
4.	Player Attack	Attack()
5.	Enemy Combat	Attack(), Detect()
6.	Enemy Movement	Move()
7.	Player HealthBar	Health(), SetHealth(), SetMaxHealth(), Take_Damage(), Die()
8.	Enemy HealthBar	Health(), SetHealth(), SetMaxHealth(), Take_Damage(), Die()
9.	NPC	Talk(), Give_item()
10.	Marchent	Sell()
11.	Magic Power	Damage()

5.7 Class cards

After identifying our final classes we have generated the following class cards :

Player	
Attributes	**Methods**
● Player_health, ● Player_Movement, ● Player_Combat	● Player()

Responsibility	Collaborator
1.Constructor 2.Controlled by user	Player, Player_health, Player_Movement, Player_combat

Enemy	
Attributes	**Methods**
● **Enemy_health,** ● **Enemy_Movement,** ● **Enemy_Combat**	● **Enemy()**
Responsibility	**Collaborator**
1.Constructor **2.Controlled by system**	**Enemy,** **Enemy_health,** **Enemy_Movement,** **Enemy_combat**

Player Movement	
Attributes	**Methods**
● **Horizontal_move,** ● **jump,** ● **runSpeed**	● **Move(),** ● **Jump()**
Responsibility	**Collaborator**
1. Moving **2. Jumping** **3. Controlled by user**	**Player Movement,** **Player**

Player Attack	
Attributes	**Methods**
● **attackRange,**	● **Attack()**

• **attackDamage,** • **attackPoint**	
Responsibility	**Collaborator**
1. **Attacking**	**Player attack,** **Player,** **Enemy,** **Enemy Combat**

Player HealthBar	
Attributes	**Methods**
• **Max Health,** • **CurrentHealth**	• **Health(),** • **Take_Damage(),** • **Die()** • **SetHealth(),** • **SetMaxHealth()**
Responsibility	**Collaborator**
1. **Setting Health** 2. **Alive or die check** 3. **Damage measure**	**Player,** **Player Health,** **Enemy Combat**

Enemy Movement	
Attributes	**Methods**
• **Radius,** • **PatrolSpeed,** • **relativeX,** • **atLeft**	• **Move(),**
Responsibility	**Collaborator**
1. **Moving**	**Enemy Movement,** **Enemy**

Enemy Combat	

Attributes	Methods
• PlayerHealth, • nextTime, • attackTimer	• Attack()
Responsibility	**Collaborator**
2. Attacking	Player attack, Player, Enemy, Enemy Combat

Enemy HealthBar	
Attributes	**Methods**
• Max Health, • CurrentHealth	• Health(), • Take_Damage(), • Die() • SetHealth(), • SetMaxHealth()
Responsibility	**Collaborator**
1. Setting Health 2. Alive or die check 3. Damage measure	Enemy, Enemy Health, Player Attack

Merchant	
Attributes	**Methods**
• Item_Price_Mapping	
Responsibility	**Collaborator**
1. Trading with Player	Player

NPC	

Attributes	**Methods**
• **Item** • **Dialogue**	• **Talk(),** • **Give_item()**
Responsibility	**Collaborator**
1.Taking with Player **2.Giving item**	**Player**

Magic Power	
Attributes	**Methods**
• **Focus_cost** • **Damage**	• **Damage()**
Responsibility	**Collaborator**
1.Damaging	**Player**

6. DATA MODELLING OF Thakurmar Jhuli: An Action Adventure Game

6.1 Data Modelling Concept

If software requirements include the necessity to create, extend or interact with a database or complex data structures need to be constructed and manipulated, then the software team chooses to create data models as part of overall requirements modeling. The entity-relationship diagram (ERD) identifies all data objects that are processed within the system, the relationships

between the data objects and the information about how the data objects are entered, stored, transformed and produced within the system.

6.1.1 Data Objects

A data object is a representation of composite information that must be understood by the software. Here, composite information means information that has a number of different properties or attributes. A data object can be an external entity, a thing, an occurrence, a role, an organizational unit, a place or a structure.

6.1.1.1 NOUN IDENTIFICATION

We identified all the nouns whether they are in problem space or in solution space from our story:

Table 1

Serial Number	Noun	Problem/Solution space	Attributes
1	Nilkomol	s	10,11,12,13,21

2	Suorani	p	
3	Duorani	s	11,21
4	Shakchunni	s	11,21
5	Necklace	p	
6	Rajendra Forest	p	
7	Bangoma	p	
8	Bengomi	p	
9	Pangkhiraj	p	
10	Coin	s	
11	Health	s	
12	Stamina	s	
13	Focus	s	
14	Merchant	p	
15	NPC	p	
16	Player_Rank	s	1,17,22
17	Playtime	s	
18	Monster	s	11,21
19	Save_Data	s	1,11,20
20	Player_Position	s	
21	Damage	s	
22	Enemies_Killed	s	

6.1.1.2 Potential Data objects:

Table 2

Serial no	Data Object	Attributes
1	Nilkomol	10,11,12,13,21
2	Duorani	11,21
3	Shakchunni	11,21
4	Monster	11,21
5	Player_Rank	1,17,22
6	Save_Data	1,20,18

6.1.1.3 Analysis for finalizing Data Objects

- ❏ Nilkomol being the playable character, all his attributes can be stored in the Player_Profile data object which will more generally refer to the player.
- ❏ Duorani, Shakchunni and Monster, all fall in under the category Enemy. So only the Enemy data object is enough to represent all enemies to defeat.
- ❏ Player_Rank data object requires Player_Profile, Enemies defeated, Playtime etc. So it has a relation with Player_Profile and Enemy.
- ❏ Save_Data similarly like Player_Rank has relation with Player_Profile, Enemies and Player_Position.
- ❏ IDs are added to each Data object as primary keys.
- ❏ For keeping multiple enemies in Save_Data, Save_Game_Enemy table is created

6.1.1.4 Final Data Objects

In the following table we finalize the data objects with their attributes Most of the attributes of the data objects are selected from the usage scenario and some of the attributes are selected to complete the system which are not in the usage scenario but important for the data objects .

Table 3

Serial No.	Data Object	Attributes
1	Player_Profile	Coin, Health, Stamina, Focus, Damage, Player_ID
2	Player_Rank	Player_Profile, Enemy_Killed, Playtime, Player_ID
3	Save_Data	Player_Profile, Player_Position, Save_ID
4	Enemy	Health, Damage, Enemy_ID
5	Save_Game_Enemy	Save_ID, Enemy_ID, Quantity

6.2 DATA OBJECT RELATIONSHIPS

Data objects are connected to one another in different ways.

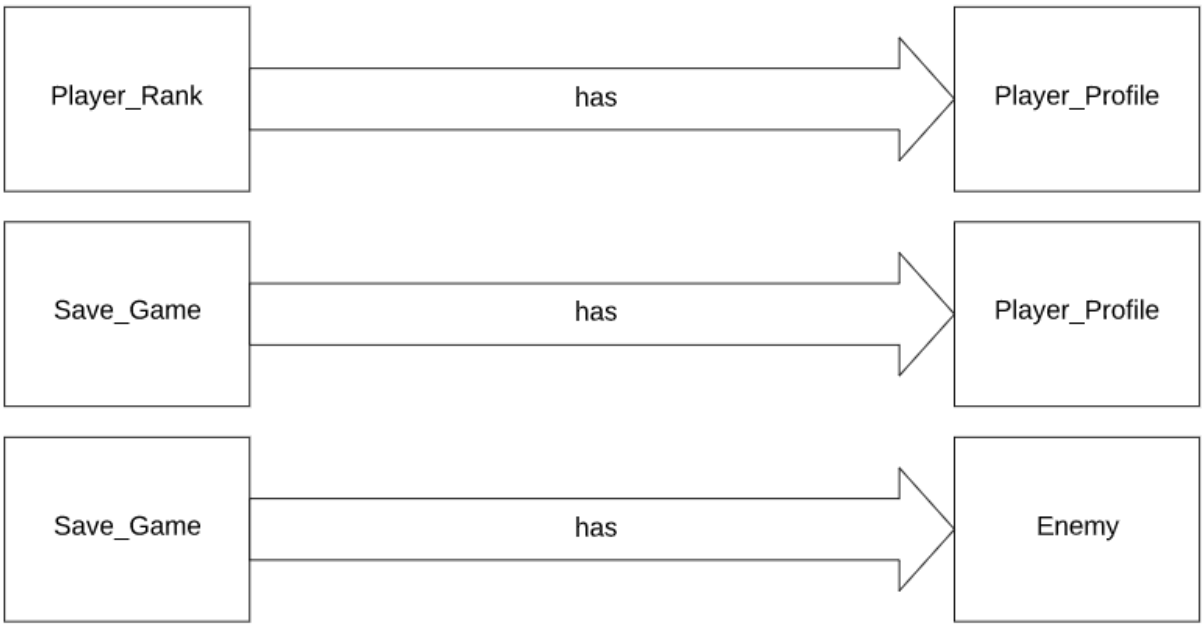


Fig: Data Object Relationship Diagram

6.3 ENTITY RELATIONSHIP DIAGRAM

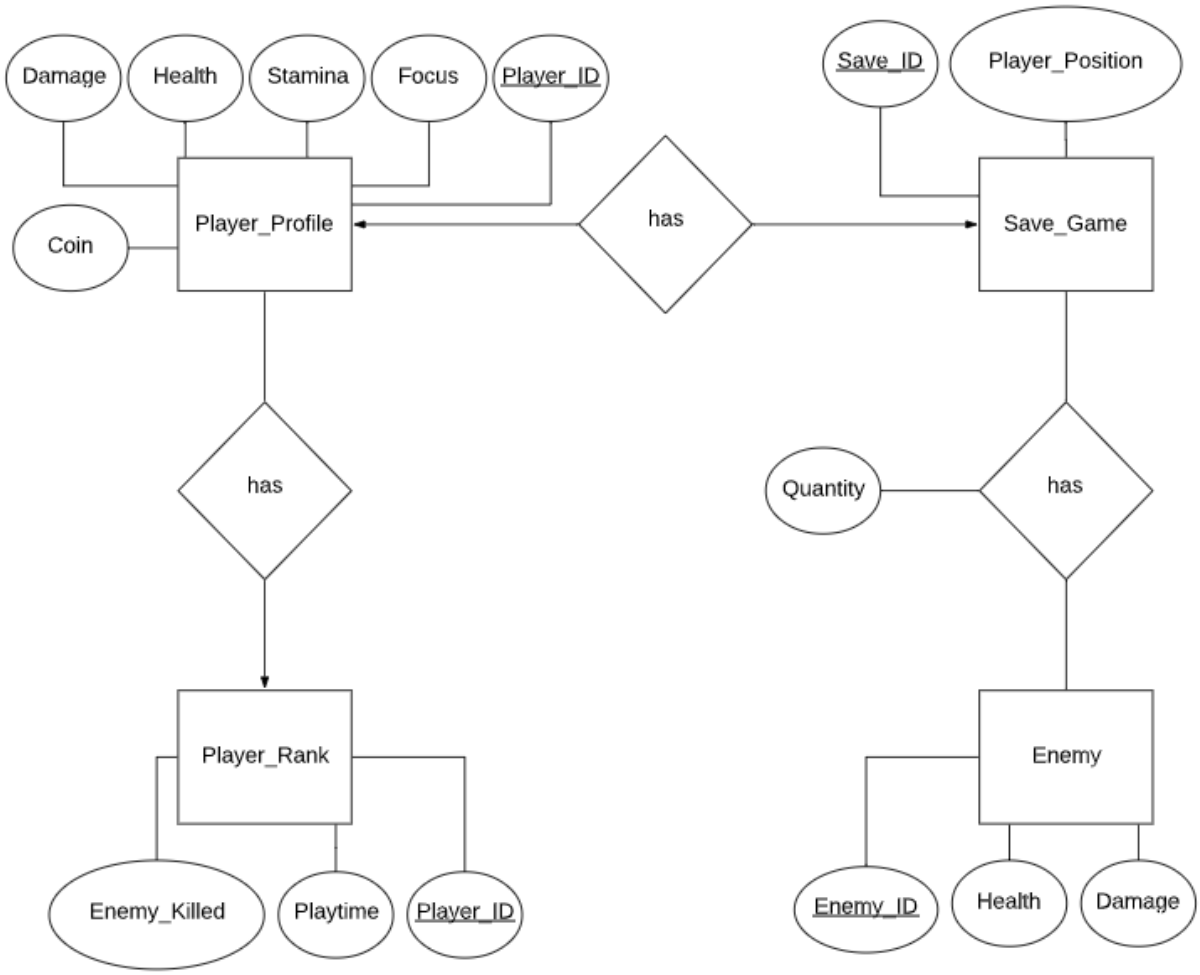


Fig: Entity Relationship Diagram

6.4 SCHEMA DIAGRAM

Table 3

Player_Profile

Attribute	Type	Size
Health	Integer	5
Stamina	Integer	5
Focus	Integer	5
Damage	Integer	5
Coin	Integer	10
Player_ID	Varchar	50

Table 4

Player_Rank

Attribute	Type	Size
Player_ID	Varchar	50
Player_Profile	Object	150
PlayTime	Float	20
Enemy_Killed	Integer	10

Table 5

Save_Game

Attribute	Type	Size
Save_ID	Varchar	50
Player_Position	Float	20
Player_Profile	Object	150

Table 6

Enemy

Attribute	Type	Size
Enemy_ID	Varchar	50
Health	Integer	5
Damage	Integer	5

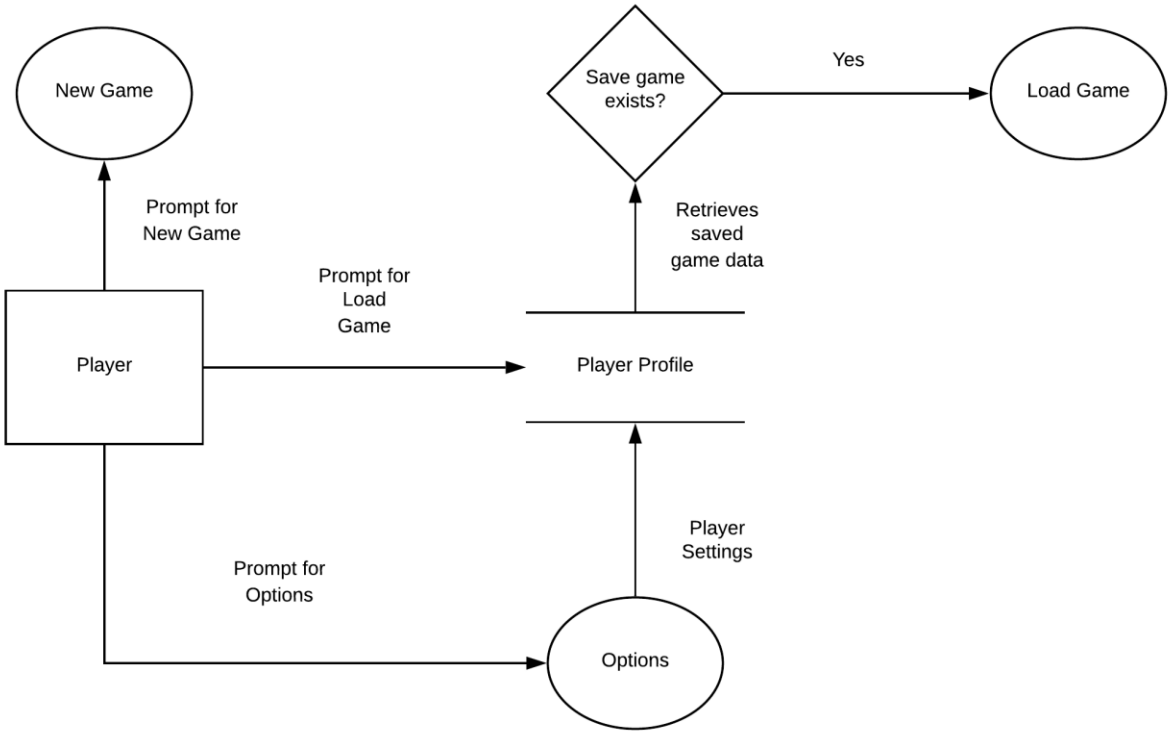
Table 7

Save_Game_Enemy

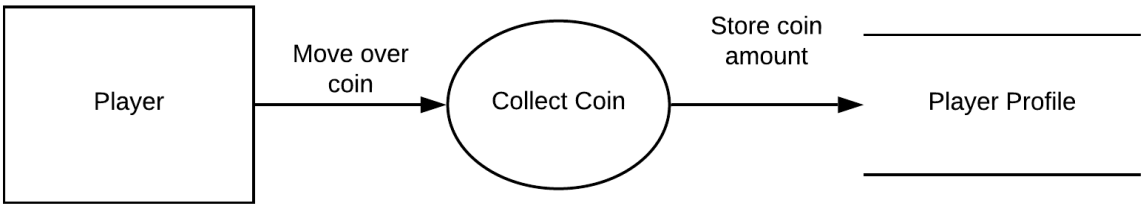
Attribute	Type	Size
Save_ID	Varchar	50
Enemy_ID	Varchar	50
Quantity	Integer	5

7. Data Flow Diagram

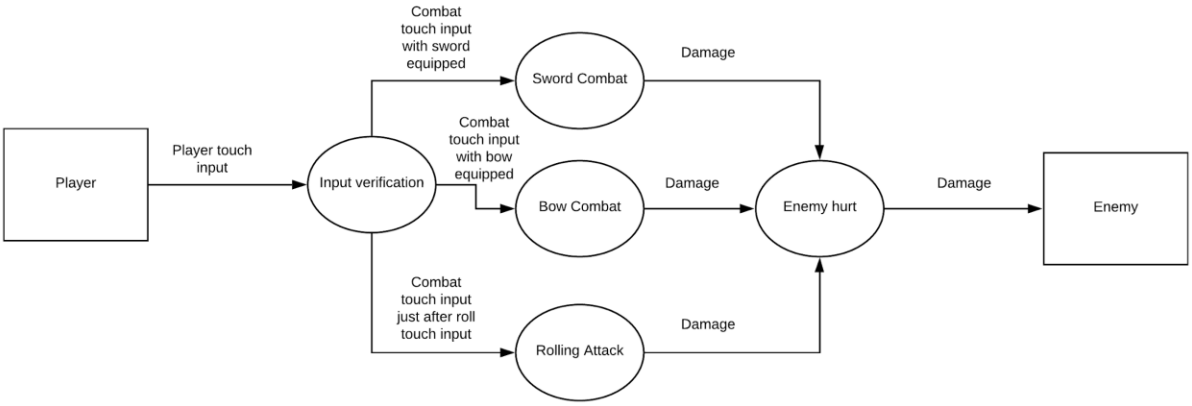
7.1. Menu Data Flow



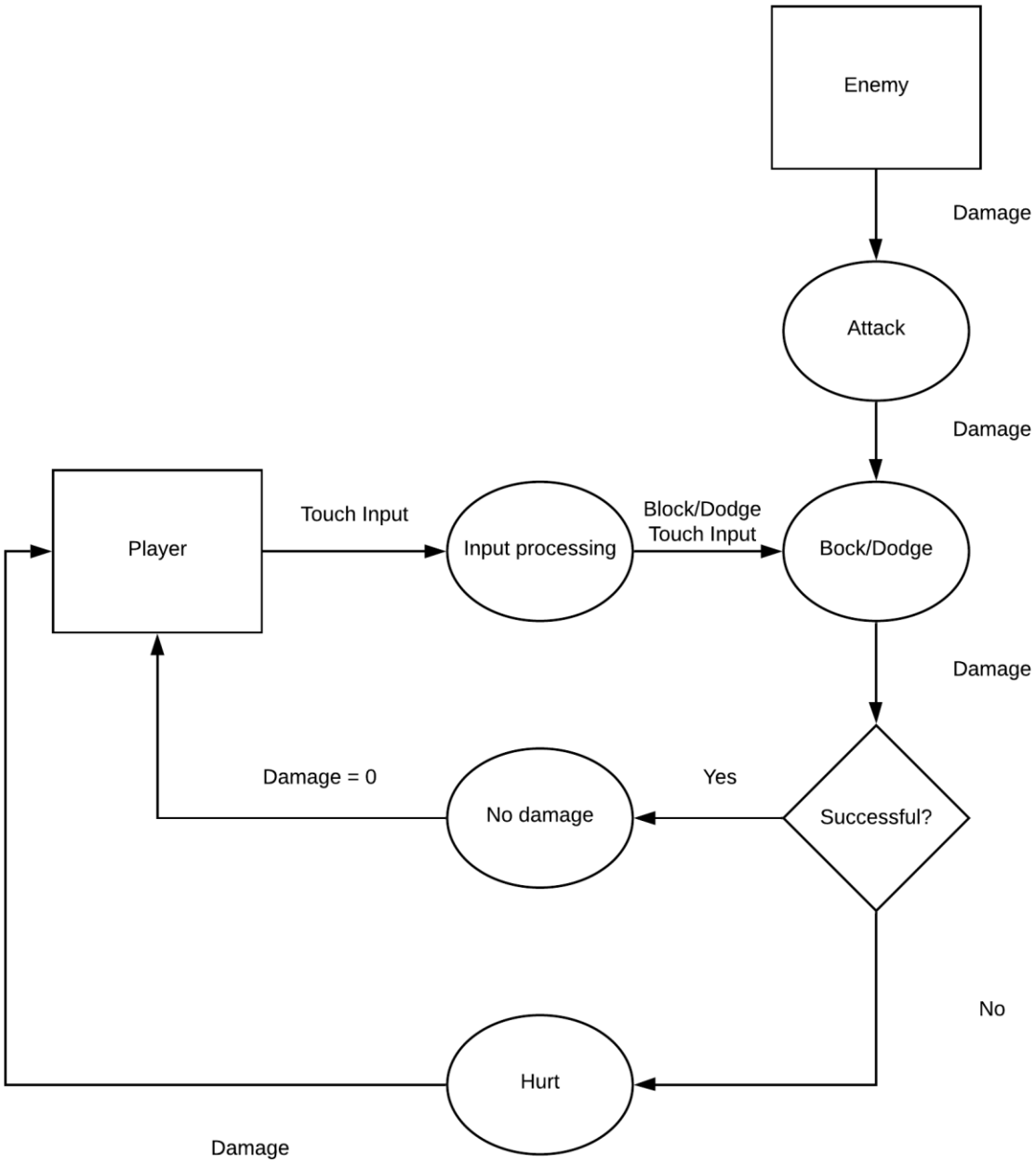
7.2..Coin Collection Data Flow



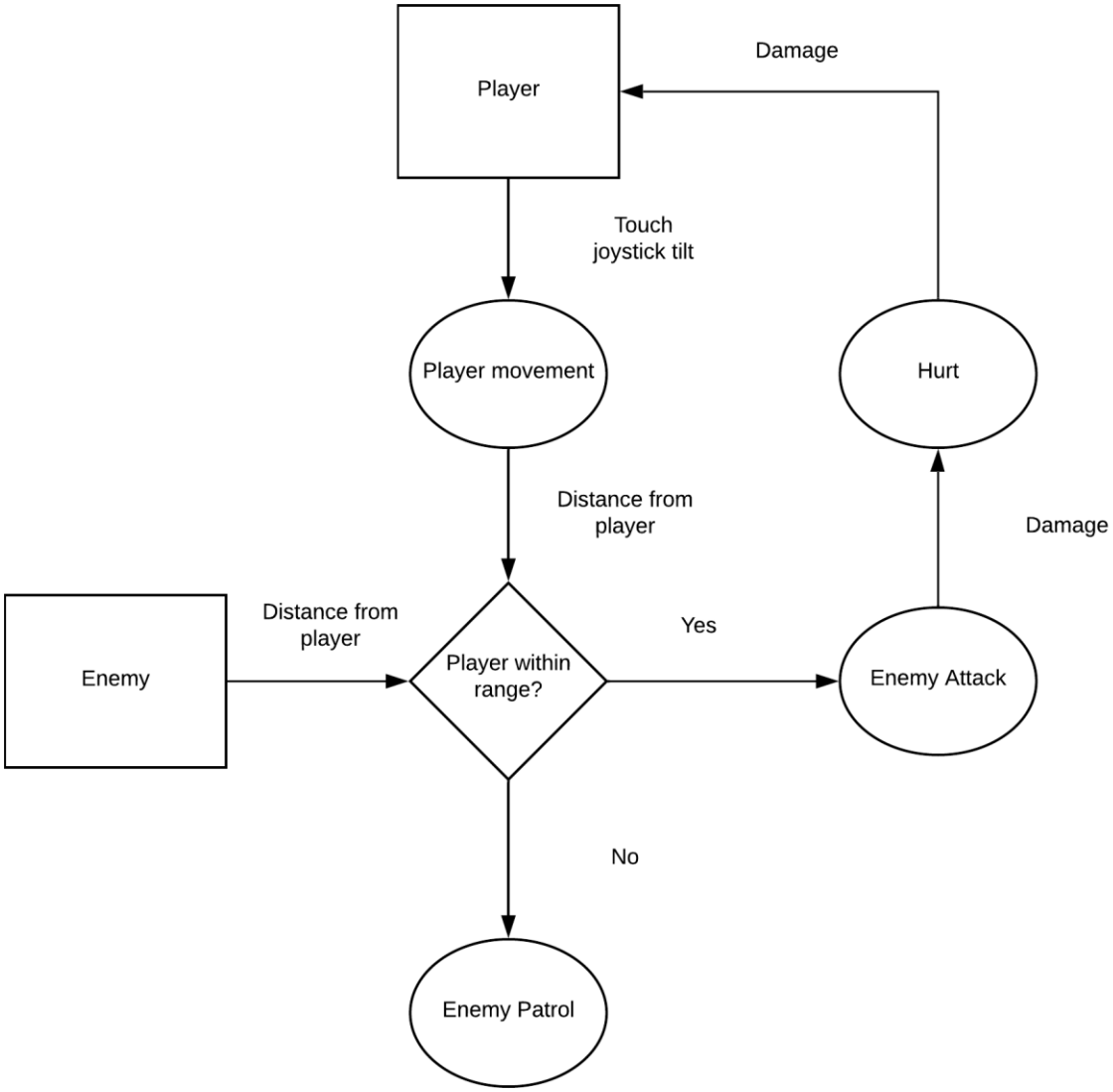
7.3. Combat Data Flow


Player
Player touch input
Input verification
Combat touch input with sword equipped
Sword Combat
Damage
Combat touch input with bow equipped
Bow Combat
Damage
Enemy hurt
Damage
Enemy
Combat touch input just after roll touch input
Rolling Attack
Damage

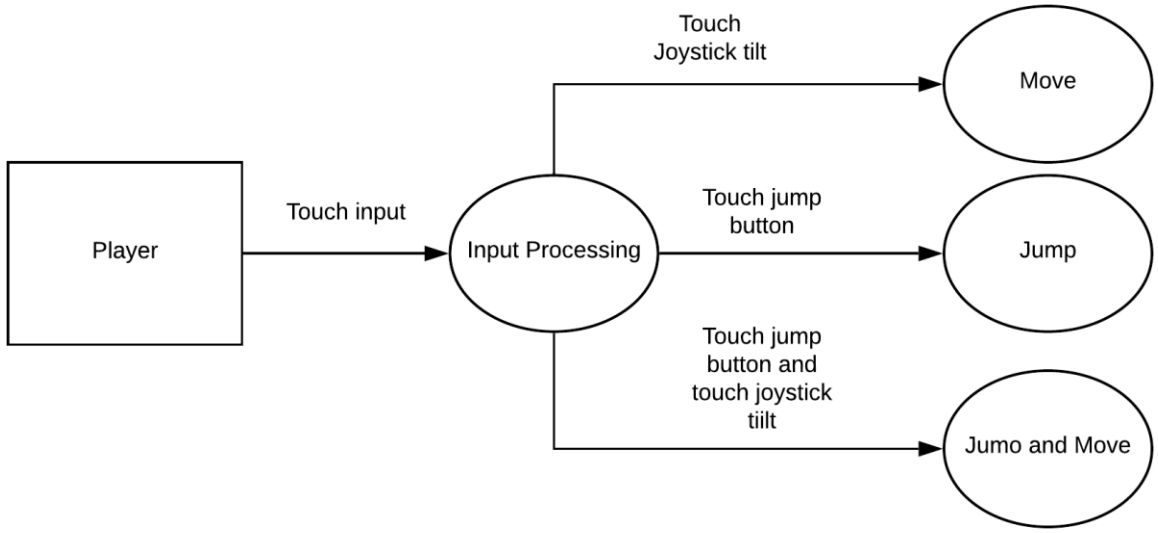
7.4. Defence Data Flow



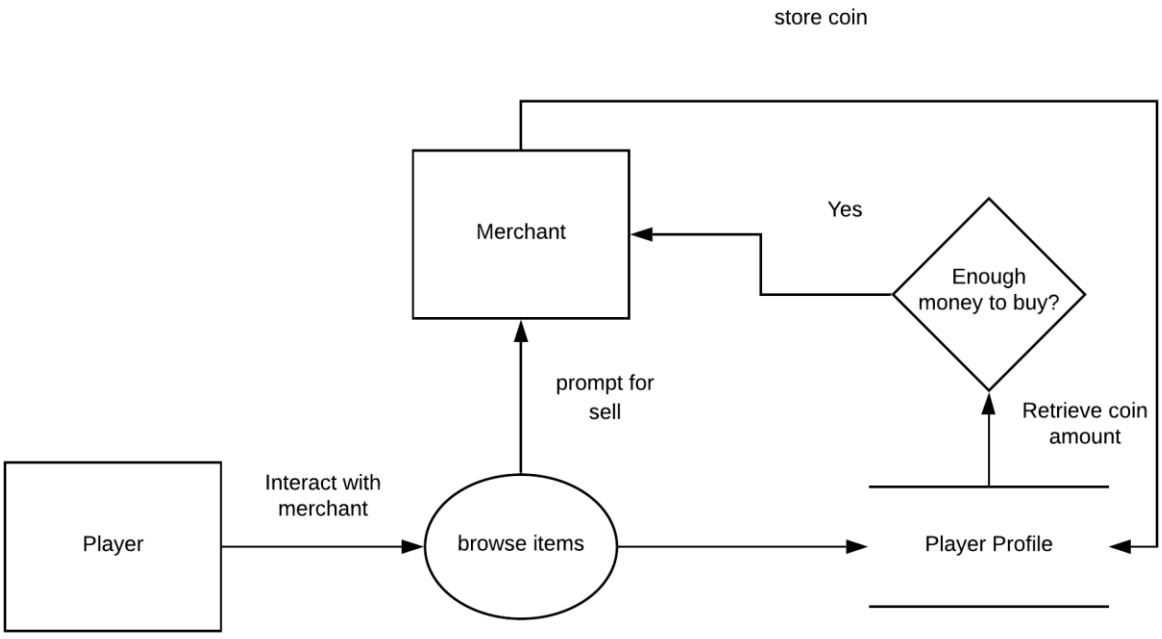
7.5. Enemy AI Data Flow



7.6. Platforming Data Flow



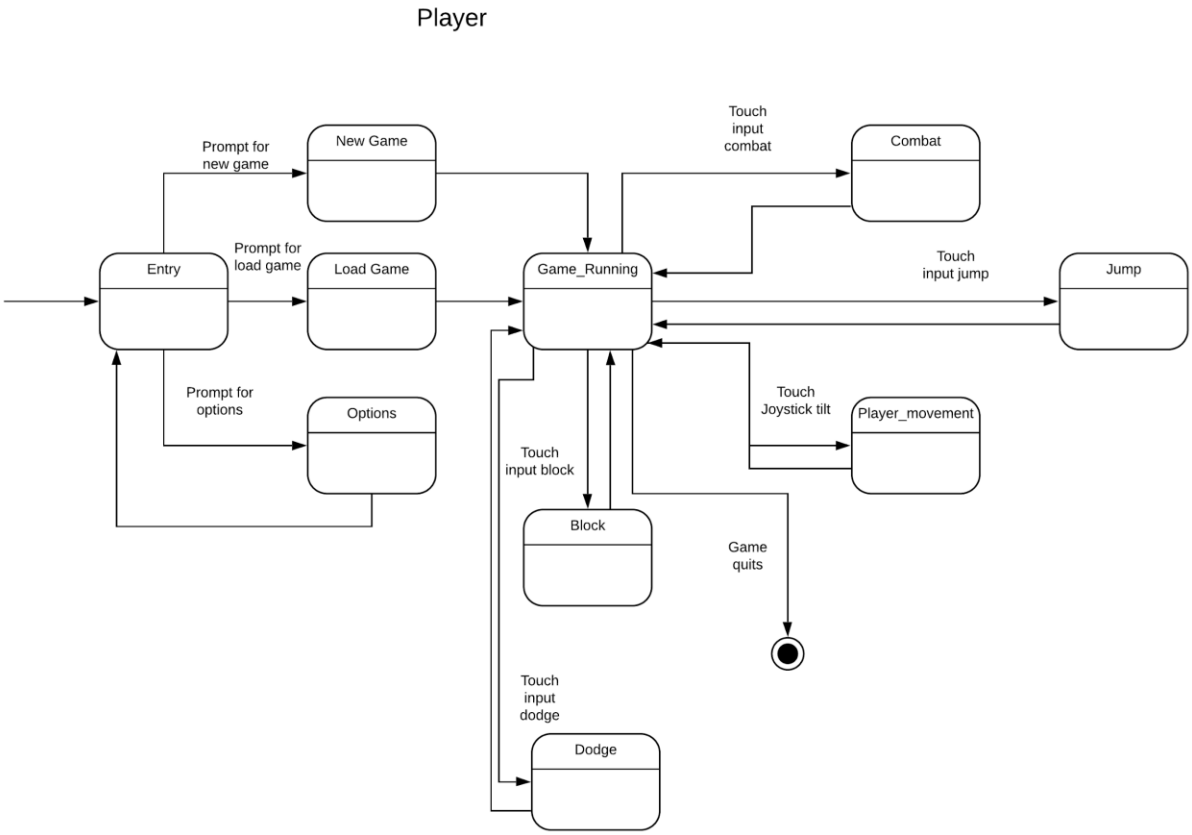
7.7. Buy-Sell Data Flow



8. Behavioral Modelling

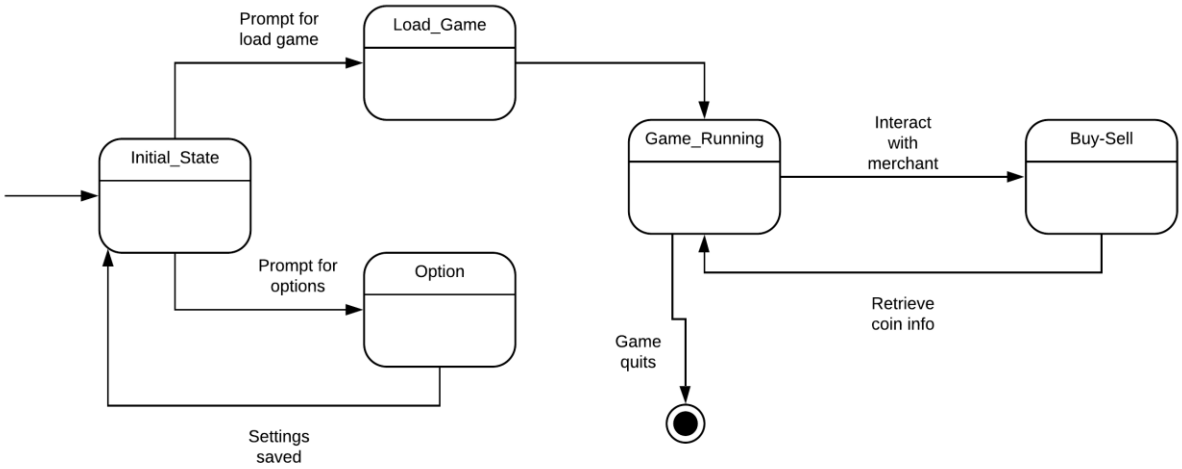
8.1. State Transition Diagram

8.1.1.Player State Transition

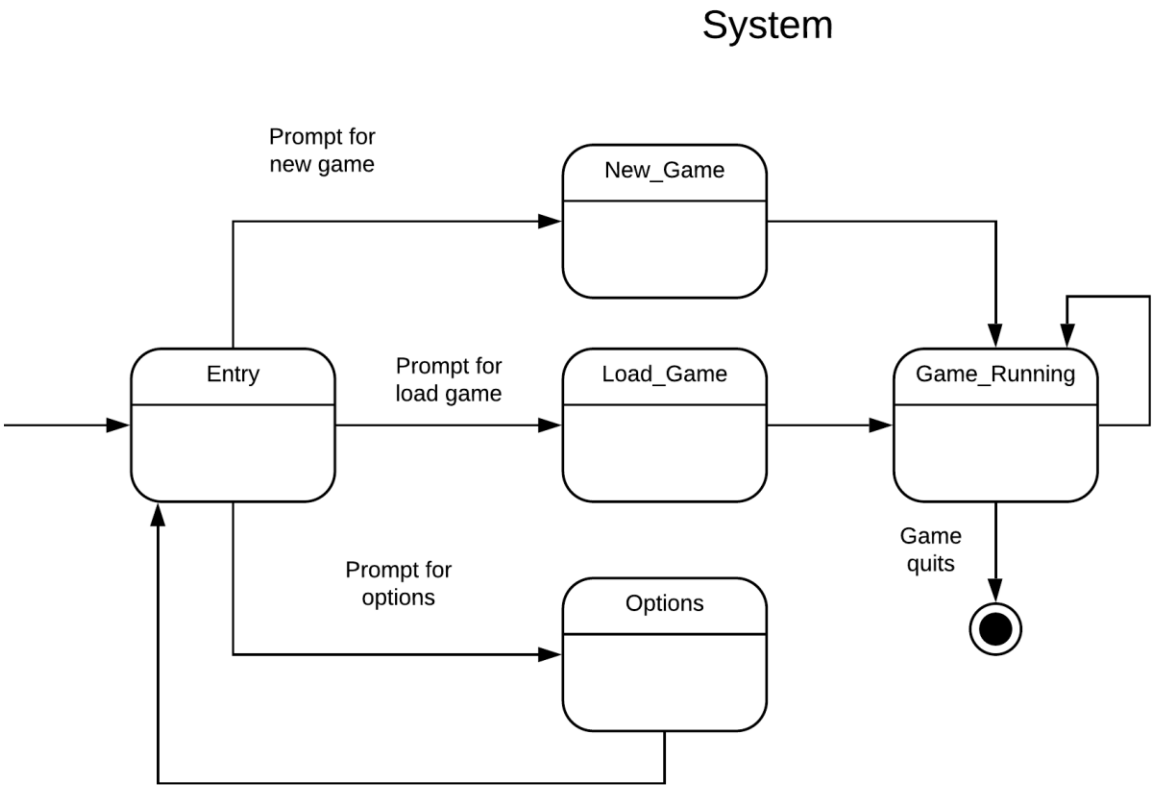


8.1.2. Player Profile State Transition

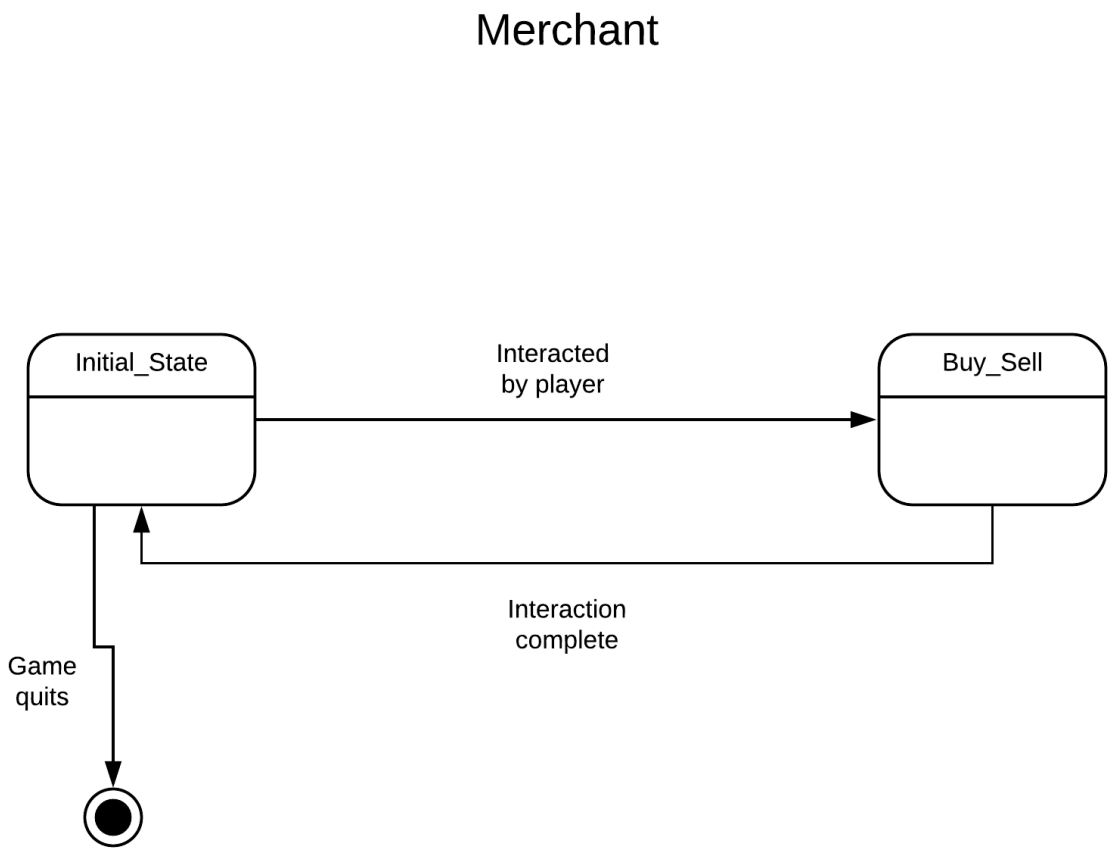
Player_Profile



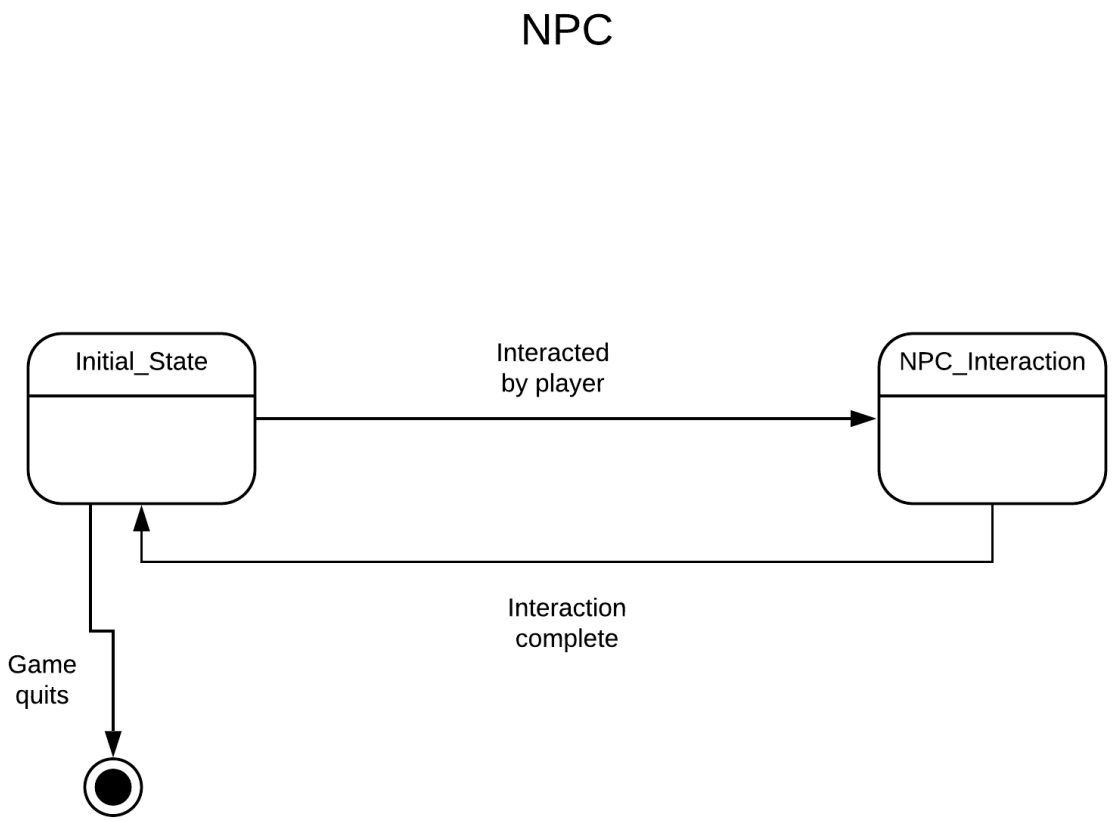
8.1.3. System State Transition



8.1.4. Merchant State Transition



8.1.5. NPC State Transition



8.2. Sequence Diagram