



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**

**“Slayer! The Web Base Game”**

**A PROJECT REPORT**

**Submitted to**  
**Department of Computer Application Kathmandu**  
**BernHardt College**  
**Bafal, Kathmandu Nepal**

*In partial fulfilment of requirements for the Bachelors in Computer Application*

**Submitted by:**

<u>Name</u>	<u>Roll No.</u>
Prashanna Pun Magar	4610536

**August,2022**

**Under the supervision of**  
**Kumar Prasun**

## ABSTRACT

Entertainment has always been a big part of life for everyone, young or old everyone needs entertainment to enjoy their life and for their mental wellbeing. Slayer is web app game which is meant to provide entertainment. Slayer! The Web Base Game is an RPG game where a player gets to experience a thriller and excitement. RPG games has been with us since the ancient times, war general use to play these types of games to improve their tactics and for entertainment purpose.

Keywords: RPG, browser game, Slayer

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Project Supervisor **Mr. Kumar Parson**, who continuously helped and guided me through different problem faced during this whole Project titled “Slayer”.

I would like to thank Co-Ordinator **Mr. Rambabu Mahato** for assisting in the different scope of the programs. I would like to thank Kathmandu Bernhardt College for providing all the resources and ideal environment for the project developing process and encourage us to develop such a project.

I would also like to share our gratitude to Tribhuvan University for providing the opportunity to evaluate our programming skills through this project.

Prashanna Pun Magar

4610536

## Table of Contents

<b>ABSTRACT.....</b>	<b>iii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iv</b>
<b>LIST OF ABBREVIATIONS .....</b>	<b>vii</b>
<b>LIST OF FIGURES .....</b>	<b>viii</b>
<b>LIST OF TABLES .....</b>	<b>ix</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
<b>1.1 Introduction .....</b>	<b>1</b>
<b>1.2 Problem Statements.....</b>	<b>2</b>
<b>1.3 Objectives .....</b>	<b>2</b>
<b>1.4 Scope and limitation .....</b>	<b>2</b>
<b>1.4.1 Scope .....</b>	<b>2</b>
<b>1.4.2 Limitation.....</b>	<b>2</b>
<b>1.5 Development Methodology .....</b>	<b>2</b>
<b>1.6 Report Organization .....</b>	<b>3</b>
<b>CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW .....</b>	<b>5</b>
<b>2.1 Background Study .....</b>	<b>5</b>
<b>2.2 Literature Review .....</b>	<b>5</b>
<b>CHAPTER 3: SYSTEM ANALYSIS AND DESIGN .....</b>	<b>7</b>
<b>3.1 System Analysis .....</b>	<b>7</b>
<b>3.1.1 Requirement analysis .....</b>	<b>7</b>
<b>I. Functional requirements.....</b>	<b>7</b>
<b>II. Non-functional requirements .....</b>	<b>8</b>
<b>3.1.2 Feasibility Analysis.....</b>	<b>8</b>
a) Technical feasibility .....	8
b) Economic feasibility.....	8
c) Operational feasibility .....	9
<b>3.1.3 Object Modelling: Class Diagram.....</b>	<b>9</b>
<b>3.1.4 Dynamic modelling: state and sequence diagram .....</b>	<b>10</b>
<b>3.1.4.1 Sequence diagram.....</b>	<b>10</b>
<b>3.1.4.2 State diagram.....</b>	<b>10</b>
<b>3.1.5 Process modelling: Activity diagram .....</b>	<b>11</b>
<b>3.2.1 Object Modelling: Object and Class Diagram.....</b>	<b>14</b>
<b>3.2.2 Component diagram.....</b>	<b>15</b>
<b>3.2.3 Deployment diagram .....</b>	<b>15</b>

3.3 Algorithms .....	16
<b>CHAPTER 4: IMPLEMENTATION AND TESTING .....</b>	<b>17</b>
4.1 Implementation.....	17
4.1.1 Tools Used .....	17
4.1.2 Implementation Details of modules.....	18
4.2 Testing .....	19
4.2.1 Taste cases for unit testing.....	19
4.2.2 Test cases for System testing.....	20
<b>CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION.....</b>	<b>22</b>
5.1 Lesson learnt/ outcomes .....	22
5.2 Conclusion.....	23
5.3 Future recommendations.....	23
<b>REFERENCES</b>	

## **LIST OF ABBREVIATIONS**

CSS- Cascading Style Sheet

HTML- Hyper Text Markup Language

JSON – Java Script Object Notation

JS- Java Script

OOP – Object Oriented Program

RPG – Role-Playing Games

UML- Unified Modeling Language

## LIST OF FIGURES

Figure 1 Waterfall Model.....	3
Figure 2 Use Case diagram .....	8
Figure 3 Class Diagram .....	9
Figure 4 Sequence Diagram.....	10
Figure 5 State diagram .....	11
Figure 6 Action Activity .....	12
Figure 7 Battle Activity .....	13
Figure 8 Physical Class Diagram.....	14
Figure 9 Component Diagram .....	15
Figure 10 Deployment Diagram .....	15
Figure 11 Main Menu .....	22
Figure 12 Battle.....	22
Figure 13 Over World.....	23

## LIST OF TABLES

Table 1: Test cases for Unit Testing .....	19
Table 2: Test case for System Testing.....	20



# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

RPG also known as Role-Playing Games, is a game in which players assume the role of characters in a frictional setting. Player takes responsibility for acting out these roles within a narrative, either literal acting or through a process of structured decision-making regarding character development.[1] Actions taken within many games succeed or fail according to a formal system of rules and guideline.

Slayer! The Web Base Game is a Role-Playing Games (RPG). The game revolves around an only pupil of a famous pizza chef, who has to hunt down the cults who stole the secret scroll. You battle different opponents to become strong. This means you increase your pizza's health max as the level increases. The story revolves around a hero, following him through his journey to get back the secret recipe which was stolen by a cult. When battle takes place the game shifts into battle screen view where both the player's and enemy's pizza are seen on the screen. Both hero and enemy can use maximum of two pizza to battle.

So, when creating a Role-Playing Games, the key is making sure that player sticks around. The story should be immersive and interesting for the player to get hooked on. Video Game is specially played to gain the sense of adventure. With role-playing games, this is especially true, the adventure you take on as a player, has its highs and lows, ups and down and is highly dependent on the role you are playing. Role-playing game give the gamers a break from stress of life, allowing us to focus in on a world that may be better than our own. Essential powers chosen by the player helps them to become the best version possible through gaming.[2]

According to Wikipedia there are several forms of role-playing games. The original form, sometimes called the tabletop role-playing games (TRPG), is conducted through discussion, whereas in live action role-playing (LARP), players physically perform their characters actions. In both of these forms, an arranger called a game master (GM) usually decides on the rules and setting to be used, and each of the other players takes on the role of a single character.[3]

## **1.2 Problem Statements**

Just about all of the most popular video games have some requirement for problem- solving and / or critical thinking? This promotes adaptability and cognitive flexibility. These are really important skills to have in any kind of problem-solving task.

Studies have indicated that compared to non-gamers, experienced gamer are better at: Task objects; keeping track of several objects simultaneously; filtering out irrelevant information; switching from task to task; detecting changes in visual layouts; and 3D mental rotation.

## **1.3 Objectives**

The major objectives of the project are:

- To make offline web app game
- To make people realize that literature cannot be only found on books
- To improve cognitive skills

## **1.4 Scope and limitation**

### **1.4.1 Scope**

When people talk about video games development, they often read that that rate of project abandonment is very high and it's true; it's fun, attractive, with the right functionality, and with enough marketing to be seen among the many that are published every day.

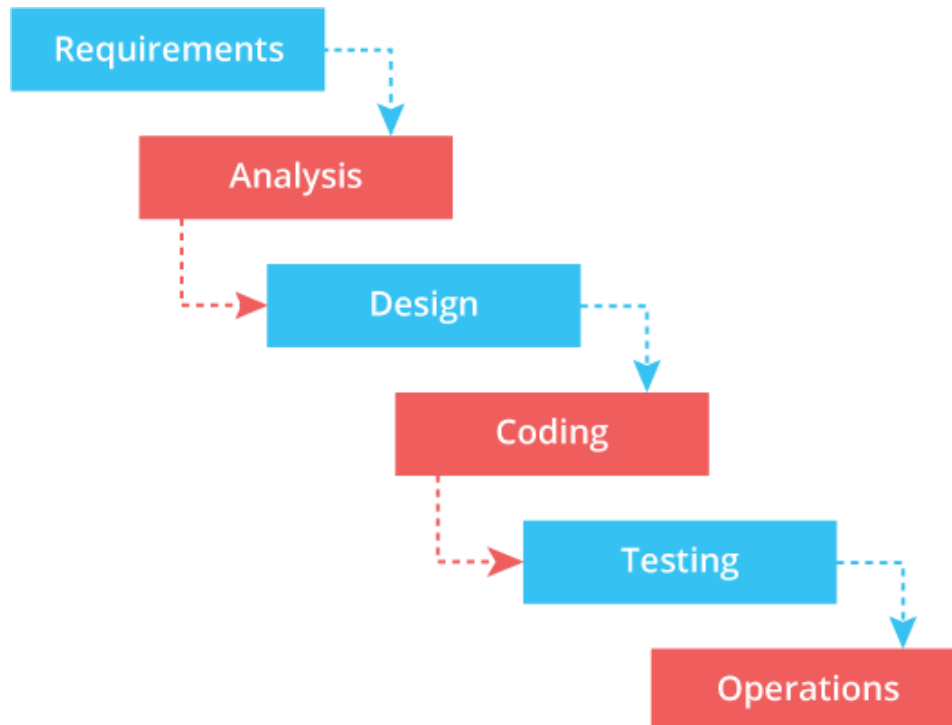
### **1.4.2 Limitation**

- Only works on JS version 1.5 or above
- May cause epilepsy
- Does not support color blind mode

## **1.5 Development Methodology**

Waterfall model has been used to develop the web app game "Slayer". The software development activity is divided into different phase and each phase consist of a series of tasks and has different objectives. It is divided into phases and output of one phase becomes

the input of the next phase. It is mandatory for a phase to be completed before the next phase starts. There is no overlapping in the waterfall model.



**Figure 1 Waterfall Model**

(Source: <https://www.oxagile.com/article/the-waterfall-model/>)

Water fall model was used to during the making of my project because the objective was clear during the start of my project.

## **1.6 Report Organization**

This report consists of five chapters, which starts with the introductory chapter that discusses the need of the game along with the problem statement and objectives of the project. Here, the tools that are used while completing the project are also introduced. Chapter 2 analyses the existing system along with background study and literature review of other systems. Chapter 3 summarizes the system design along with the requirement analysis and feasibility analysis. Chapter 4 explain the tools that are used on our project. The modules and the development model used are also explained in this part. Chapter 5 discusses the conclusion of how the project is accomplished, its findings and many more.

Recommendation for future enhancements of the project is also discussed here. In conclusion, this chapter overview's purpose of doing this project including its scopes and objectives.

## CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

### 2.1 Background Study

The history of Role-Playing Games begins with an earlier trading of role-playing, which combined with the rulesets of fantasy wargames in the 1970 to give rise to the modern role-playing games.[4] RPG is a type of game which the participants assume the roles of characters and determine the actions of their characters based on their characterization and the action succeed or fail according to a system of rules and guidelines. Within the rules, the participants may improvise freely; their choices shape the direction and outcome of the games.

RPG are substantially different from competitive games such as ball games and card games. This has led to confusion among some non-players about the nature of fantasy gaming. The game *Dungeons & Dragons* was a subject of controversy in the 1980 when well-publicized opponents claimed it caused negative spiritual and psychological effects. Nowadays educators support RPG as a healthy way to hone reading and arithmetic skills.[5]

### 2.2 Literature Review

I had to review several games to make my project. From different browser games to different Role-Playing games, some of them are:

**Pokémon** originally known as *Pocket Monster* is a series of RPG developed by Game Freak and published by Nintendo and The Pokémon Company. It is about humans known as Pokémon trainers collecting different monsters and battling other trainers for sports. Over the years, a number of spin-off games based on the series have also been developed by multiple companies. While the main series consists of RPGs, spin-off games encompass other genres, such as action role-playing, puzzle, fighting and digital pet games. Pokémon was made by Satoshi Tajiri in around 1989, which generated from this hobby as a child where he used to collect beetles and battle with his friends. [6]

**Digimon** also known as *Digital Monster* is a Japanese franchise encompassing virtual pet toys, anime, manga, video games and films. The franchise focuses on the digital creatures, who inhabit a “Digital World”, a parallel universe that originated from Earth’s various

communication networks. There are many that call Digimon as Pokémon clone but the truth is Digimon was inspired for Tamagotchi. Tamagotchi is a virtual pet which the real person has to take care or they might die. The franchise gained momentum with early RPG called Digimon World, which was based on both video game and digital pet in 1999. [7]

**Nexomon** is a Role-Playing Game that takes place in fantasy world inhabited by humans and filled with powerful creatures known as Nexomon. Humans who tame these monsters are called Nexomon Tamers. The Nexolord rules over all Tamers and all other humans. The player takes control of the protagonist who can be male or female depending on the player's choice. [8]

**Coromon** is a pixel world filled with challenging puzzles, interesting creatures and titanic bosses. The story follows an aspiring researcher joining the technologically advance organization of Lux Soils, who is assigned to be a specialized team called the Titan Taskforce, and are tasked with investigating the six Titans known in the region of Velua. But soon finds extraterrestrial forces bringing a mysterious elemental type to the planet, causing the fundamental of the world slowly collapse. [9]

**Browser game** is a video game that is played via the internet using a web browser. They are mostly free to play and can be single player or multiplayer. Some browser games are also available as mobile apps. The advantage of the browser version is not having to install the game. The front end of the browser game is what runs in user's browser. it is implemented with the standard web technologies of HTML, CSS, JS and WebEssembly. In addition, WebGL enables more sophisticated graphics. On the back end, numerous server technologies are used. When internet first became widely available and initial web browser with basic HTML support were released, the earliest browser games were text based. The browser game evolved as the support for browser plug-ins and JS was introduced. In 1999 Tom Fulp kickstarted the flash games scene with the release of Pico's School on his site. Different Social media sites like Facebook, hi5 etc. drove more player to Brower games. [10]

## **CHAPTER 3: SYSTEM ANALYSIS AND DESIGN**

### **3.1 System Analysis**

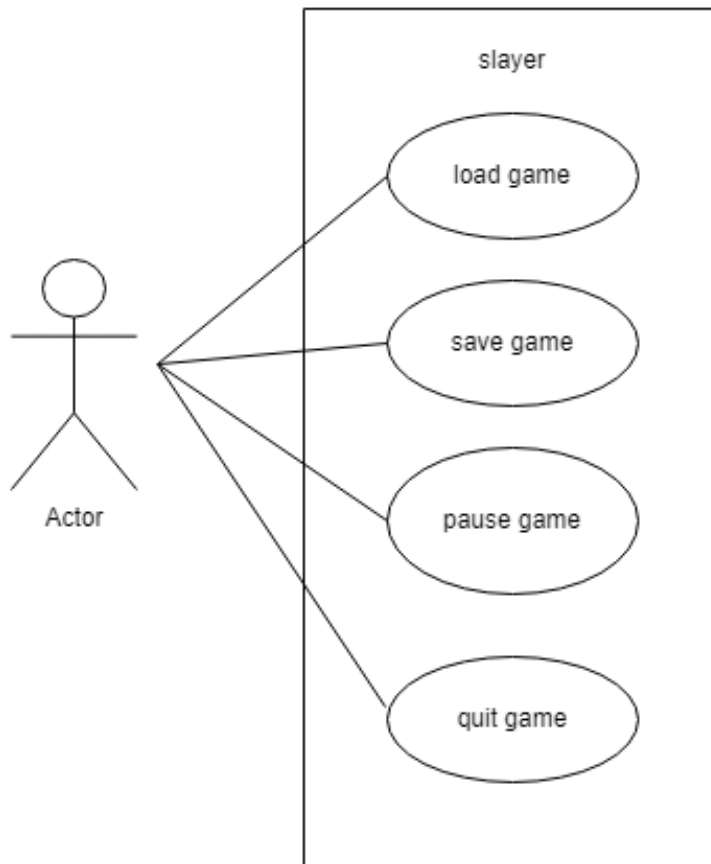
System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Employee Leave Management System. It is done to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. It is an important phase of any system development process. The system is studied to the minutest detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system is identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified

#### **3.1.1 Requirement analysis**

Requirement analysis results in the specification of operational characteristics of software: indicates interface of software with other system elements and establishes constraints the software must meet. The requirement analysis is mainly categorized into two types functional and non-functional:

##### **I. Functional requirements**

- They system should be running windows 7 or above



**Figure 2 Use Case diagram**

## **II. Non-functional requirements**

- No personal information required to gain access of the game.
- No special permission for admin is required

### **3.1.2 Feasibility Analysis**

#### **a) Technical feasibility**

Technical feasibility assesses the current resources (hardware and software) and technologies, which are required to accomplish user requirements. It requires a computer with a computer with web browser.

#### **b) Economic feasibility**

It is economically feasible because the project is created using free tools and application like VS-code.



### c) Operational feasibility

It is operational feasibility since the system/project fulfills the needs or requirements it need to.

### 3.1.3 Object Modelling: Class Diagram

Class diagram represents the static view of an application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. It shows a collection of classes, interfaces, associations, collaborations and constraints.

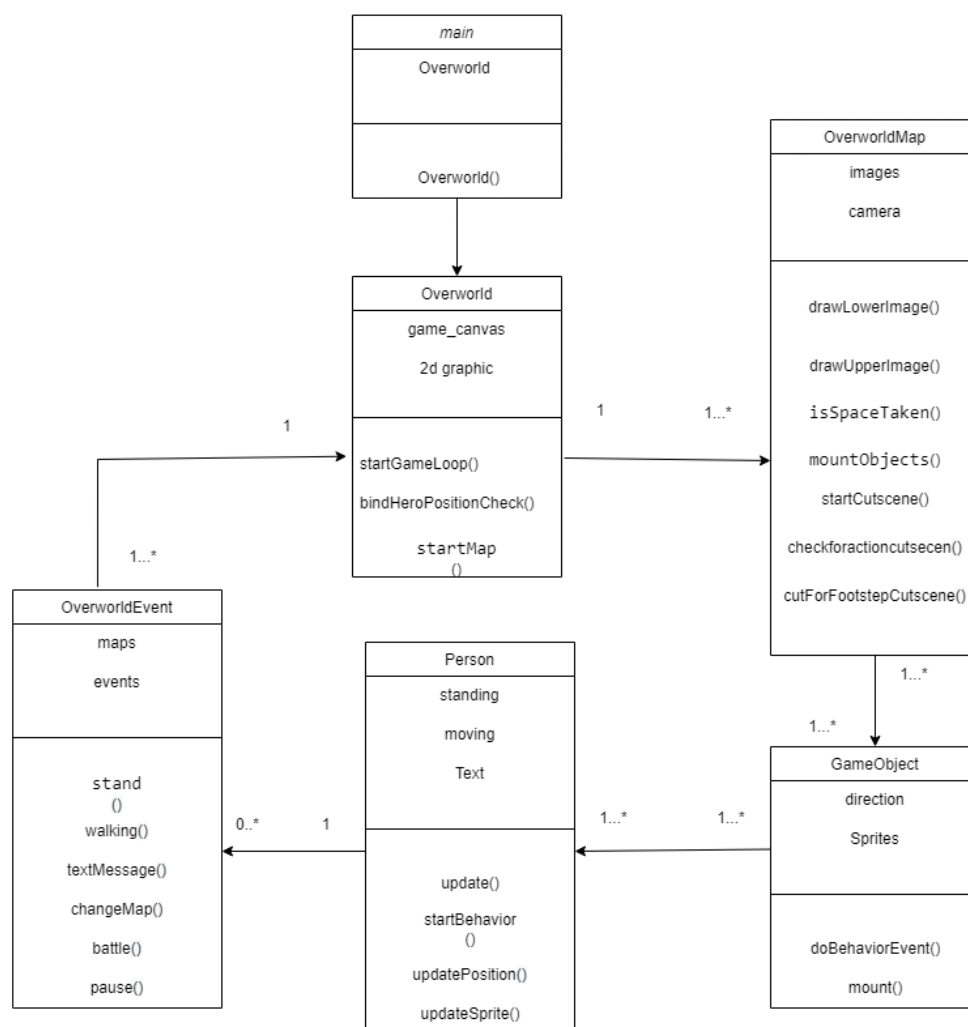


Figure 3 Class Diagram

### 3.1.4 Dynamic modelling: state and sequence diagram

#### 3.1.4.1 Sequence diagram

Sequence diagrams are interaction diagram that detail how operations are carried out. It describes interactions among classes in terms of an exchange of message over time.

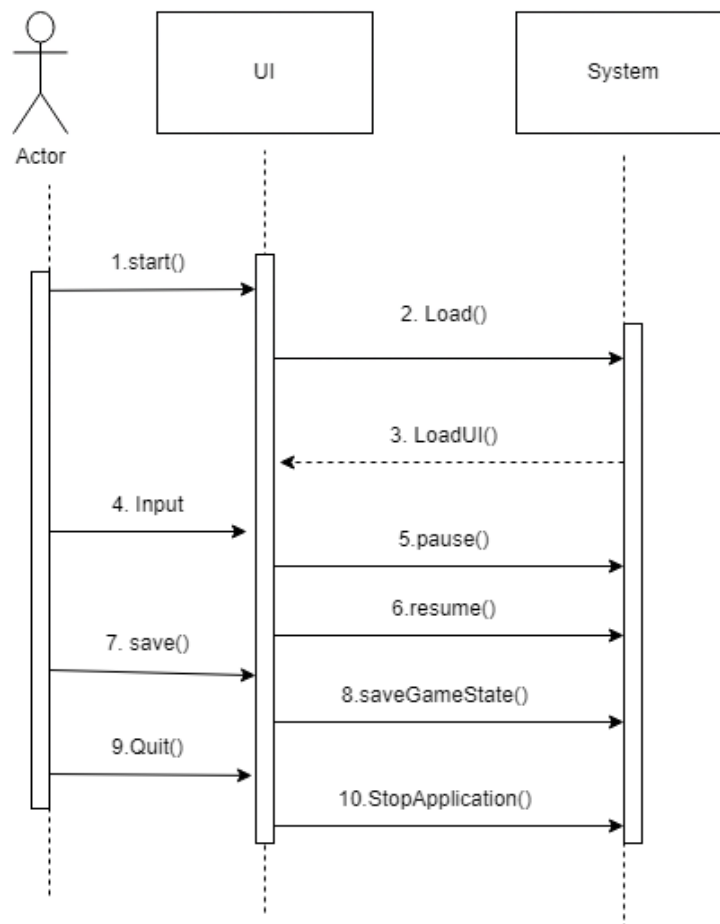


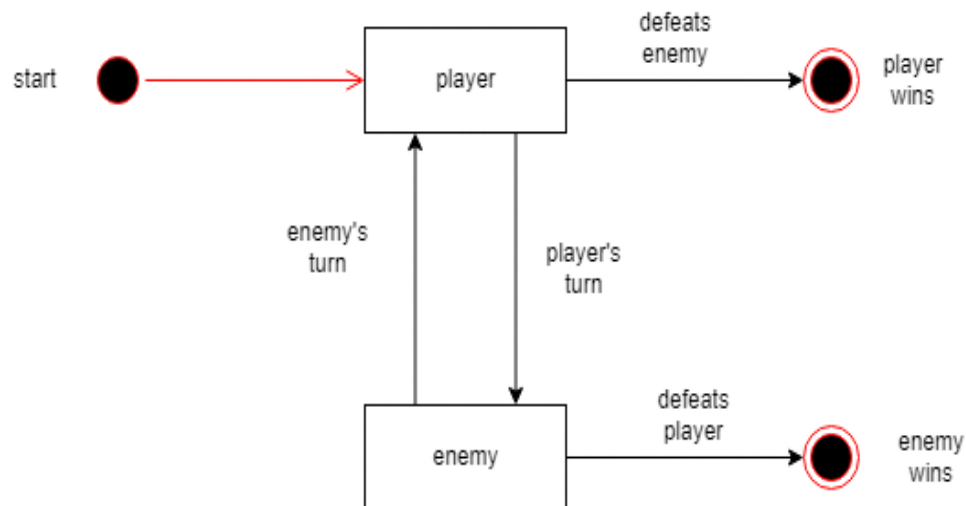
Figure 4 Sequence Diagram

Actor starts/ loads the game and the system load the UI. The actor can give input to move the hero, they can pause, resume, save and quit the game, with the help of different UI.

#### 3.1.4.2 State diagram

State diagram is used to design the dynamic aspect of the system. It defines the state of the components and state changes triggered by an event. It is used to identify the events

causing the changes in the state and it is also used for forward and reverse engineering.

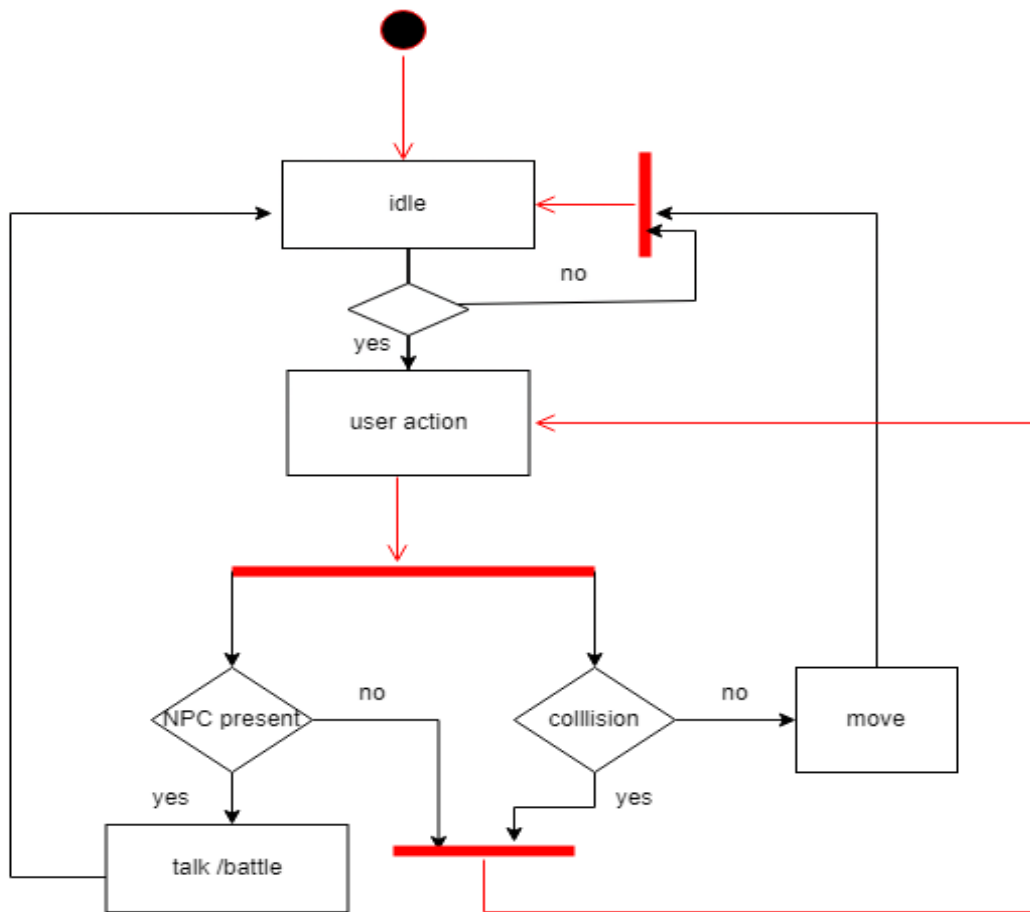


**Figure 5 State diagram**

### **3.1.5 Process modelling: Activity diagram**

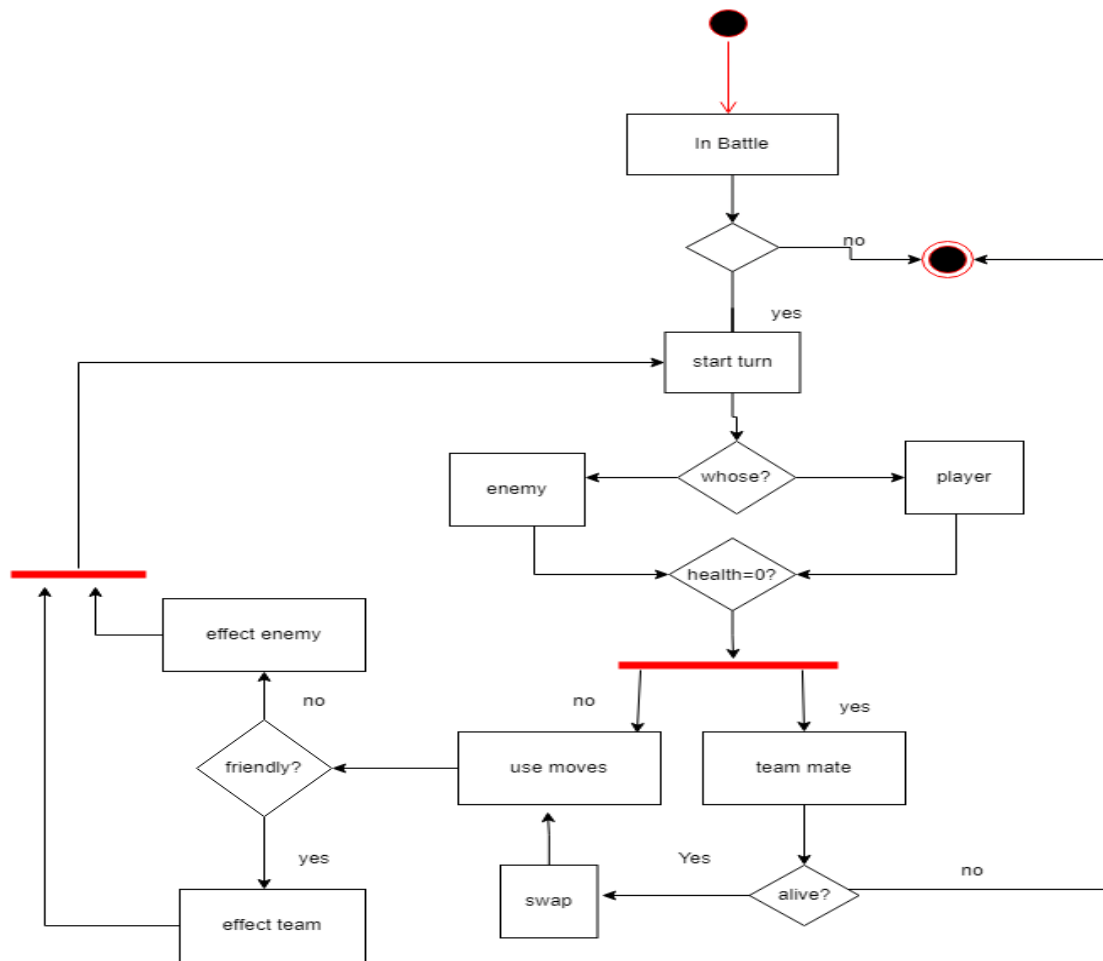
The activity diagram is a flow chart to represent the flow of control among the activities in a system. It is another important diagram in UML to describe the dynamic aspects of the system. The activity can be described as an operation of the system.

It consists of activities that are made up of smaller actions. It is an advancement of a flowchart that contains some unique capabilities.



**Figure 6 Action Activity**

In the above figure, for the user to provide different action to the hero, the game checks if the hero is idle or not and if the hero is idle the player can assign different action like walking, talking or battling. For walking activity, the game checks if there is any collision or not and if there is no collision the hero moves, but if there is a collision the hero does not move and the player can assign another action. For talking activity, the game checks if there is any NPC in the direction the hero is facing. If the NPC is present the hero talks with them and battle if needed. But if there is no one the user can assign another action to the hero. If the action provided to the hero is successful the game check if the hero is idle or not and the player can provide another action to the hero.



**Figure 7 Battle Activity**

In the above figure, for the user to assign what move to use, the game first checks if the hero is in battle or not. If the player is not in battle, then the activity ends, but if the player are in battle the game checks whose turn it is at the start and after deciding the turn it also check if the health is zero or not. If the health is more than zero then the move is chosen. After the move is chosen it is checked if the move is friendly or not. If move is friendly, it effects the friendly team and if the move is not friendly it attacks the enemy. After the attack is made it checks whose turn it is. When the health is found to be zero the system checks if there is extra team mate or not. If there is no team mate then the battle ends but it there is then the team it checks if it has health or not. If the team mate is not alive then the battle ends but if the team mate is alive then it swaps with the fainted team mate and use move.

### 3.2.1 Object Modelling: Object and Class Diagram

Generalization is the refinement of a class into more refined classes. It allows the developer to model objects into hierarchical structure based on their similarities. the class being refined is classed super-class and the refined versions of is are called sub-classes. Each sub-class inherits the attributes and operations from their super-class. Methods and attributes can then be refined and the sub-class specific attributes and operations. A discriminator is a variable of enumeration type, which indicate which property of an object is being abstracted. The most important use of inheritance is the conceptual simplification it makes trough the reduction of independent features in the system. Generalization refers to the relationship among classes and inheritance refers to the method of reusing attributes and operations in the hierarchic. By definition a feature with the same name and subclass can override a superclass feature.

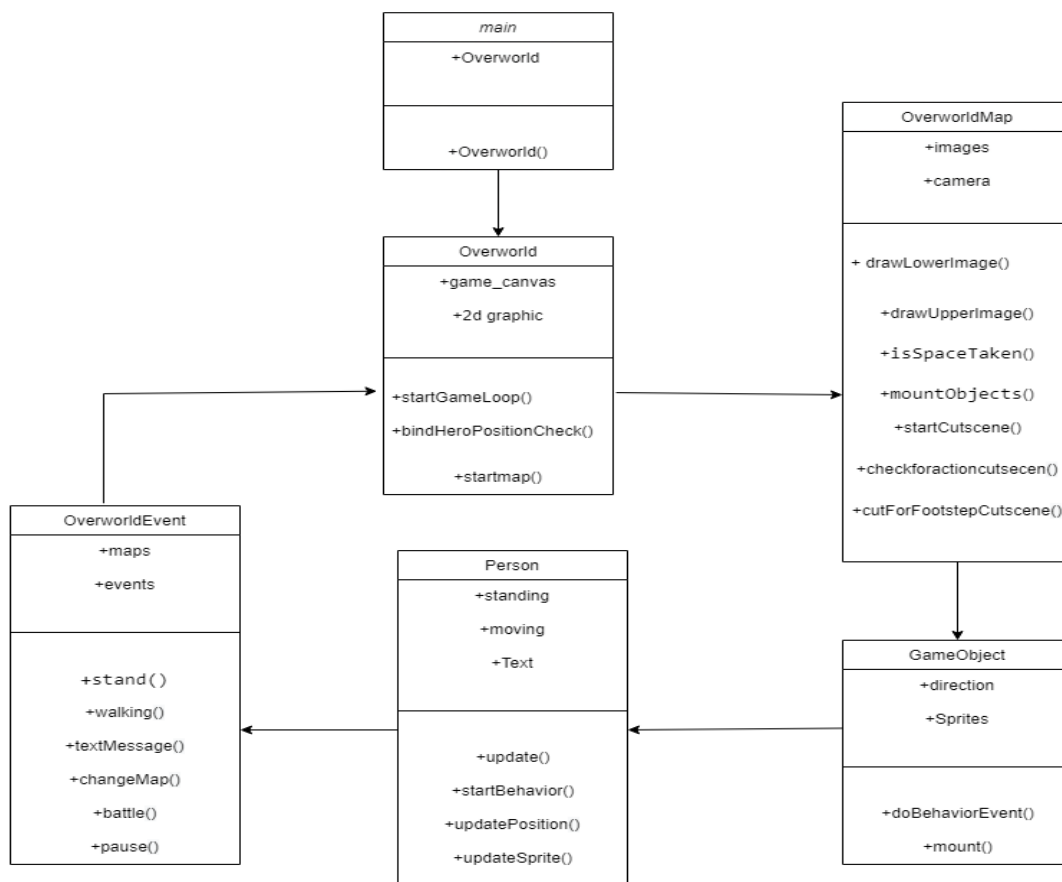
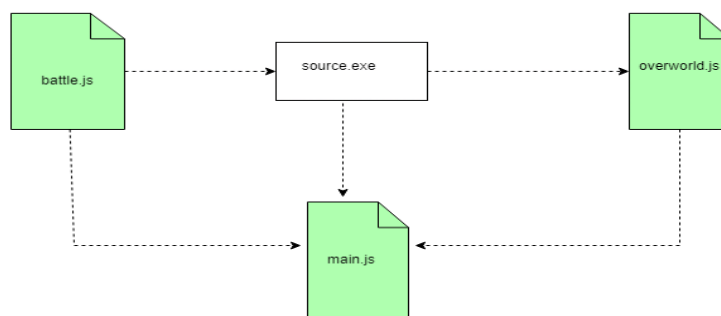


Figure 8 Physical Class Diagram

### 3.2.2 Component diagram

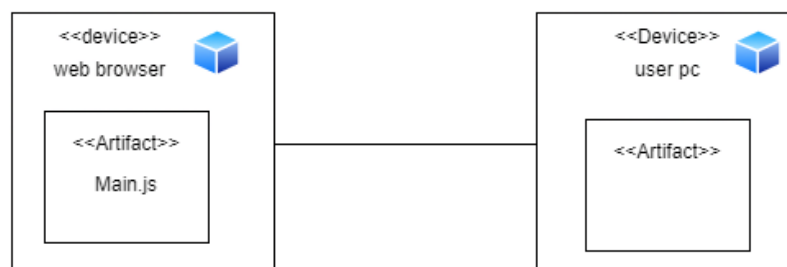
Component diagram can be described as a special kind of UML as it shows static implementation view of a system. They are used to visualize the organization and relationships among components in a system. The purpose of the component diagram is to construct executables by using forward and reverse engineering. And, they are used during the implementation phase of an application.



**Figure 9 Component Diagram**

### 3.2.3 Deployment diagram

In the context of Unified modeling language, A deployment diagram is a diagram that shows the execution architecture of a system, including nodes such as hardware or software execution environment, and the middleware connecting them. Deployment diagrams are typically used to visualize hardware and software of a system. It consists of nodes and their relationships.



**Figure 10 Deployment Diagram**

### 3.3 Algorithms

- Scaling

Scaling refers to a mathematical rule applied to change the size of the image.

If the value of scaling factor is negative, the size is decreased and when the value is positive the size is increased.

Suppose the point with coordinates A (x, y) is to be scaled by a factor of (a, b) then,

Final scaled value is

$$x_2 = x + a \text{ and } y_2 = y + b.$$

- Translation

Translation refers to a technique in which a point is shift from one place to another, whose distance is known.

Consider a point A (x, y) be shifted to another point B (x1, y2). Now we only know the shifting distance (a, b) then,

the final translated value is

$$x_1 = x + a \text{ and } y_2 = y + b.$$

- 2D collision detection (Circle Collision)

This algorithm is used to detect collision in 2D games. It depends on the type of shapes that can collide. Generally, you will have a simple generic shape that covers the entity known as "hitbox" so even through the collision may not be pixel perfect, it will look good enough and be performant across multiple entities



## CHAPTER 4: IMPLEMENTATION AND TESTING

### 4.1 Implementation

#### 4.1.1 Tools Used

1. Visual Studio Code

Microsoft Visual Studio is an integrated development environment (IDE). It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. It can produce both native code and managed code.

2. HTML

HTML or hypertext markup language is a language to make web pages. Canvas tag has been used for this project. Everything that happens in the project takes place inside this tag.

3. CSS

CSS or cascading style sheet is used to display how HTML elements are to be displayed on screen, paper or in other media. CSS is used for menu, color for health bar as well as exp bar and for the different color coating used during battle.

4. JS

JS or java script is programming language of the web. It for used almost everything for this project. From assigning the movements for NPC to manually moving the main character.

5. JSON

JSON or java script object notation is a text format for storing and transporting data. JSON was used in the project to save the game and use the saved data to continue the game from where it was left off.

### 4.1.2 Implementation Details of modules

#### a) Movement Controller

This controller is used to move all the elements in the game. Every surface walkable comprises of a map which is used to detect collision as well as detect the location and position of the character. Whenever user inputs any movement keys (w, a, s, d, enter, arrow key, space and esc) movement controller is responsible to handle the movement.

#### b) Camera Controller

This module is responsible for all of the camera movements in the game. There can be multiple cameras in game to show various parts of the game window. The camera is locked on to the hero, following him so the player can only focus on him and follow his story.

#### c) Character State

This module is responsible to handle all the characters in the game. Character features such as Health, Damage, Movement Speed, all are managed by this module. Character State determines the state of the characters in the current frame in the game.

#### d) Implementation of Algorithm

Scaling is used to scale the small images so it fits appropriately to the screen, we can properly see the images and the text present while playing the game. Translation 3D is used to move the pizza from one corner of the screen to another while attacking in battle. Rotation is used to rotate the pizza in 45 degrees to give the pizza the spinning look while it is translating through the screen.

E.g.:

At the start translation (x, y, z) are (0,0,0) because it is at starting position and rotate is at 0 deg and is scaled by 2

```
0%{transform: translate3d(0,0,0) rotate(0deg) scale(2)}
```

Then when it reaches 25% of the total distance to be traveled translation (x, y, z) is at ( -130px, 45px, 0) and rotate is at 45deg and is scaled by 2

```
25%{transform: translate3d(-130px,45px,0) rotate(45deg) scale(2)}
```

Then finally when it reaches to its destination the translation (x, y, z) is at (0, 0, 0) and rotate is at 0deg and is scaled by 2

```
100%{transform: translate3d(0,0,0) rotate(0deg) scale(2)}
```

Above are the steps for translation to so the pizza spin and move toward the enemy and vice-versa during attack animation.

2D collision detection is used during battle. Mainly circle collision is used for the battle as the pizzas are circle. During battle when the attack is performed and the collision between the two is detected by the circle collision, the pizza which is taking damage starts to blink and the HP decreases as the response to the collision.

## 4.2 Testing

### 4.2.1 Taste cases for unit testing

**Table 1: Unit Testing**

S.no	Function	Input	Expected outcome	Result	Status
1.	Move character	Arrow keys and a, s, d, w	Character movement	Character moved successfully	true
2.	Character interaction	Space and enter key	Interact with other characters	Talk or initiate battle	true
3.	pause	ESC key	pause	Pause the game and open menu	true
4.	resume	ESC key	resume	Close the menu and resume the game	true

5.	Swap pizza position		Swap the pizza position if there is more than one	Swap place with other pizza	true
6.	Save		Save the game	The game saves	true
7.	New game		New game starts	New game can be played even if there is already saved data	true
8.	continue		Continue the game	Game continues in the map we saved and with the changes to the pizza	true

#### 4.2.2 Test cases for System testing

**Table 2: System Testing**

S.no	function	Input	Expected outcome	Result	Status
1.	Initiate battle		Starts battle	Battle starts when talking to the enemy	True
2.	Trigger movement animations	Arrow keys and W, A, S, D	Walking animation	Animation triggers successfully	true
3.	Take damage on hit		Take damage	Health reduces	true
4.	Die when health 0		die	success	true
5.	Gain exp on enemy defeated		Increase exp gain	Exp bar increases	true

6.	Level up when exp reaches 100		Gains level by 1	Level and HP both increases	true
7.	Use item		Can use item on self	Can use item to heal and clear debuff	true
8.	3 types of action during battle		Self-buff, debuff enemy and damage enemy	Can buff self, debuff enemy as well as damage enemy with different moves	true
9	sounds		Play sounds during battle	Sounds should play while taking damage	false

## CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATION

### 5.1 Lesson learnt/ outcomes

There have been several improvements in our programming language and writing skill as well as our time management skills while doing this project. I conclude that this project has helped me gain more knowledge about the topic that I researched while doing my project. A lot was learned about time management as the project had to be submitted before the deadline along with the documentation. Due to time constraints, I could now add more facilities to it. Although it is expectedly good, some new features to this system could be added in the upcoming days to make it more user friendly and efficient.

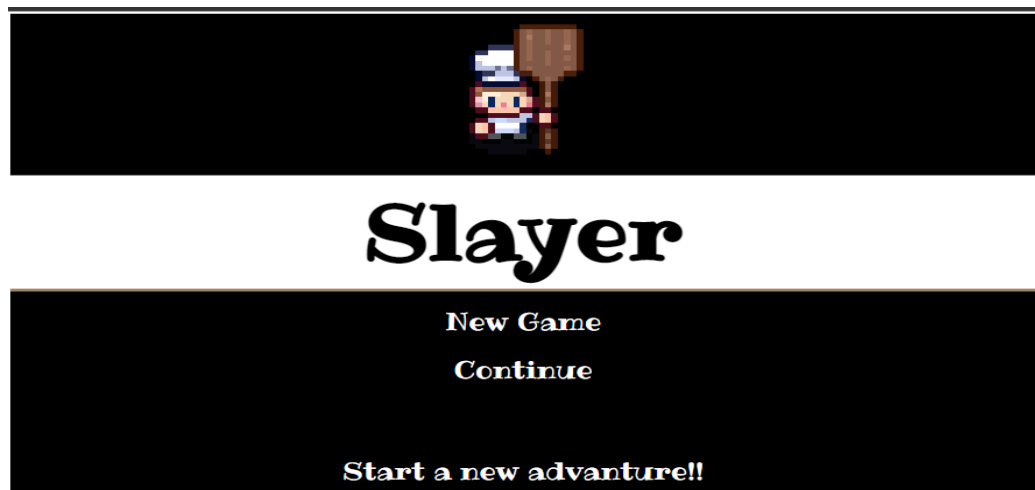


Figure 11 Main Menu



Figure 12 Battle



Figure 13 Over World

## 5.2 Conclusion

This project is done as Efficient as possible. The description of the background and the context of the project was thoroughly researched by the author. The purpose, scope, applicability, and requirement specifications of the system have been accurately explained. The author has included features and operations in detail including screen layouts and the limitations on which the project is being developed. Finally, the system is implemented and tested according to test cases. After the development of the system finally, it was tested and the views about results were exchanged. After testing, the limitations of the existing system were discussed. In conclusion, tools like Photoshop, VS code and programming language like JS is used in the development of my system. I would be glad to enhance and promote this project if given chance and help ourselves and society in the near future.

## 5.3 Future recommendations

The applicability and usage of this project can be increased by enhancing the system based on the limitations. Entertainment is now a basic need. Every Home has their own personal Computer. There are many people who need entertainment. So, with the help of this game

we can deliver a good service to customer who wants some entertainment into. This helps to uplift the mental wellbeing of people. New effective modules can be also added from time to time.

In future, new game modes can be added to the system where player can interact with even more objects. More interesting events can be added to the game with improved animation and graphics. Multiplayer can be introduced so people can battle and come out with awesome strategists.



## REFERENCES

- [1] Grouling, Jennifer (2010). *The Creation of Narrative in Tabletop Role-Playing Games*. McFarland & Company.
- [2] <https://www.gamersdecide.com/pc-game-news/7-reasons-why-we-love-rpg-games>
- [3] Kim, John. "'Narrative' or 'Tabletop' RPGs". Archived from [the original](#) on 29 August 2008. Retrieved 9 September 2008.
- [4] "Where we've been and where we're going". Archived from [the original](#) on 2011-06-29.: "Generation 1" games
- [5] Gwendolyn F.M. Kestrel. "Working Hard at Play".: An educator's opinion of role-playing games
- [6] <https://en.wikipedia.org/wiki/Pokemon>
- [7] <https://en.wikipedia.org/wiki/Digimon>
- [8] [https://nexomon.fandom.com/wiki/Nexomon\\_Wiki](https://nexomon.fandom.com/wiki/Nexomon_Wiki)
- [9] [https://coromon.wiki.gg/wiki/Coromon\\_Wiki](https://coromon.wiki.gg/wiki/Coromon_Wiki)
- [10] [https://en.wikipedia.org/wiki/Browser\\_game](https://en.wikipedia.org/wiki/Browser_game)