LONDON METROPOLITAN UNIVERSITY

Islington college

(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6P05NI Final Year Project

Assessment Weightage & Type

25% FYP Interim Report

Semester

2021 Autumn

Project Title: King's Guard (Game Development)

Student Name:

London Met ID:

College ID:

Internal Supervisor: Subin Chitrakar

External Supervisor: Sumit Bajracharya

Assignment Due Date: 15th December 2021

Assignment Submission Date: 15th December 2021

Word Count (Where Required): 5035

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

This report is an overview of the progress that has been made till date on the project "King's Guard". This report provides information on why this project was chosen and the need to make a computer game in Nepal. This project aims to be one the first original platformer game that is developed in Nepal. The report also highlights the background of game industry and current situation of Nepal's gaming industry. In the end, the report shows all the future work that is left in the development process.

Table of contents

Table of Tables

Table of Figures

1. Introduction

Game is a structured form of play. Every game has its own rules, obstacles and a goal or a predefined condition for the game to end. Games can be competitive but the main incentive to play a game is to have fun. With the increasing availability of computers, the number of people playing video games has also been rising. It was estimated that there were around 2.69 billion active gamers around the world in 2020. This number is expected to cross 3 billion by 2023. (Clement, 2021) More and more people are playing games every day which means the video game industry is growing rapidly. The recent pandemic has made a lot of people seek video games as a form of entertainment. This surge in video game popularity can also be seen in Nepal. According to a recent survey (Survey 5.3) of 43 Nepalese, 90% of them said that they played video games and majority of the people also stated that computers is their preferred platform to play games on. Understanding this, I have decided to make a computer game for my final year project.

King's guard is a 2D platformer game with a slight influence of Metroid Vania genre. The protagonist is sent by the king to investigate a distant area because people are going missing. The protagonist starts in a village and must find out what is causing this. Anybody can play this game, but the targeted audience will be people aged 18 to 34 years old.

1.1. Problem Scenario

Even though the game industry generated $155 billion in revenue last year, (Beattie, 2021) Nepal has not done anything to take a piece of the billion-dollar pie. As stated above, 90% of the surveyed (Survey 5.3) people play video games but only 15% of them have played games that were developed in Nepal. When asked for games that were made by a Nepali developer, there was no mention of a computer game. This shows that there is an abundance of Nepali gamers but a lack of Nepali games.

There has not been a significant improvement in Nepal's gaming industry. Almost all the games that are developed in Nepal are mobile games which are a recreation of board games so there is a lack of games made for computers. This scarcity might be due to various reasons, one of them being that developing a big video game requires a lot of investment and manpower. AAA games cost around $60 to $80 million on average to make. (Veresockaya, 2020) With a budget this big, it is impossible to find an investor in Nepal. Nepal's gaming industry is still in its infancy stage, so it is not economically viable to spend millions of dollars on developing a big-budget game.

Although a lot of Nepalese are interested in video games, they lack a powerful computer which is required to run almost all modern games. In the same survey mentioned above, only 38.5% said that they had a high-end computer/laptop. Most people buy a laptop for work/college, so they do not have the latest and greatest graphics card in it so to play most games, people must spend extra money to get a better computer which not everyone can afford, especially in Nepal.

1.2. Project as Solution

This project is going to be an indie game that is going to be developed just by me. This will ensure that the project will be affordable as the only resource it requires will be my time. Although the industry is still in its nascent stage there is a lot we can do besides recreating board games. There are a lot of Nepali gamers and very few Nepali games which means that almost everyone is playing games that are made by foreign companies. 72% of the surveyed people stated that they have spent money on video games so this means that a lot of capital we could be earning is going out of our country. By creating a game on our own, we could stop the money from going out and develop our gaming industry. Since the majority of people do not have high-end computers, this game will have low system requirements. This will ensure that more people can run the game without having problems.

1.3. Aims and Objectives

The aim of this project is to create a fun, complete, 2D platformer game where the character will have abilities and upgrades. The game will have different enemies, NPC, and a boss battle.

The objectives of creating this project are:

- To learn about game design and level design by studying existing platformer games and applying it to this project within 2 months.
- To learn how to make a game using Unity by following online tutorials and creating this project in unity within 5 months.
- To learn how to use C# to make games by watching online tutorials and using it to write code for the game within 5 months.
- To make a fun, playable 2D platformer in 5 months. A survey can be conducted to ensure this objective has been achieved.

2. Structure of the report

Introduction

An introduction to the project where the project is explained in brief. Problems that exist and ways this project helps to reduce the problems is also explained. Finally, the aims and objectives of this project are stated.

Background study

This section provides information on how the gaming industry started and how it is today. This section also has information on the current state of Nepal's gaming industry. A literature review is also done along with a comparison between 3 similar games and this project.

Progress

Since the proposal was not even a month ago there has not been any significant progress in the development of the project. This section contains the system requirement specification, use case diagrams, survey that was conducted and the assets that are chosen.

Future work

Here, all the development work that is left is explained. Questions like "Is the project going as per the Gantt Chart?" are answered and all the work that is left to complete this project is outlined.

3. Background Study

3.1. Background on video games

Long before the gaming industry was a multibillion-dollar industry, video games were being made in science labs with no intention of being sold to the public. "Cathode-ray tube amusement device" is one of the first interactive video game which was patented in 1948 (Soffe, 2014). This game used a cathode ray tube hooked to an oscilloscope display and it challenged players to fire a gun at a target. Video games like tic-tac-toe, blackjack, checkers, chess and pong were also being created in the 50's. None of these early video games were sold to the public as they were too expensive or too big to be sold. The very first computer-based video game was developed in 1962 named "Spacewar!". It was developed by a MIT student, Steve Russell. (The Strong , 2021) Personal computers were not as popular back then as it is today, so naturally, computer games were also not popular.

Ralph Baer, also known as the father of Video Games, wanted to get video games out of the science lab and onto the living room which led to the creation of a gaming console named Odyssey. Odyssey was released in 1972 and one of the games was an inspiration for the classic arcade game "Pong" made by Atari. (History, 2019) The late 1970's to mid-1980's is considered to be the "golden age" for arcade games. (Northfield, 2018) Games like space invaders, PAC-Man and Microsoft's first flight simulator were being released at this time. Nintendo’s Donkey Kong, the game that introduced the world to Mario and the first game to have a storyline, was also released at this time and it is recognized as the first platformer game. By the mid 90's the popularity of arcade games was coming to an end whereas computer and console games were gaining popularity.

3.2. Background study and Literature review of Platform games

Donkey Kong marks the genesis of the platformer genre in video games. After releasing Donkey Kong, the next platformer Nintendo created was "Mario bros" (1983) and perhaps the most influential and recognizable game "Super Mario Bros" (1985). Super Mario Bros had side-scrolling 2D level design which means the camera followed the players movement and this was new at the time. It also had powerups, multiple levels and a boss battle. Other platformer game series like Mega Man, Kirby and Sonic the hedgehog helped to solidify platformers as a game genre. Platformer has spawned countless games and given birth to many subgenres like Metroidvania and infinite runner and as computers has gotten more powerful it has expanded into third dimension too.

A platform game or more commonly known as platformer is a video game genre which is typically 2D and allows players to jump between platforms. The early platform games used to have a single screen level design where the screen would change once the player completed the level. The cameras in these games were static and the player could only move from one edge of the screen to the other. After side scrolling platformer games started to be developed, players could now move vertically as well as horizontally. This meant that the camera could no longer be stationary and had to move and follow the player. (Master Class, 2021)

The name platformer comes from the players jumping on platforms so the core elements a game needs to have for it to be platformer is a jumping mechanic and platforms. Besides these some of the common mechanisms a platformer has are different player abilities, enemies and obstacles the player can overcome through the given abilities, items the player can get and use, a main objective/destination and puzzles the player can solve along the way.

3.3. Scenario of game industry in Nepal

It has been less than a decade since Nepali developers started making games, so it is safe to say that the video game industry in Nepal is still in its infancy stage. AR Cube, Semantic Creations, Yarsa Games, Reizon Studios and Red Tail Studios are some of the Nepali companies that are developing video games.

According to Uttam Adhikari (co-founder and CEO of Red Tail Studio), there is a lack of game developers, designers, artists and game project managers in Nepal to attempt big game development projects. (Rana, 2020) Due to this all the games that are being developed by the Nepali studios are simple mobile games. Most of them are either endless runners or recreation of board or card games. There was an attempt to make an open world 3D game with a compelling story called "Chronicles of the Himalayans", but the scope of the project was too big and ambitious for Nepal's gaming industry, so the game never got released.

There is still hope for the gaming industry in Nepal as the number of players has been rising and people like Pawan Thakuri and Sandesh Tamang, commonly known by their YouTube name "4K Gaming Nepal" and "2B Gamer", have made a living just by playing video games. There is an abundance of gamers in Nepal, so it is the developer's job to create new and innovative games.

3.4. 3 similar system

Super Mario bros.

The original Super Mario Bros is the first side scrolling game that started the platformer game genre. It has a storyline and a very good character control for a game that was made in 1985. It was an inspiration and a guideline for a lot of platformer games to come throughout the years.

Super Mario bros was made by Nintendo for their 8-bit game console Nintendo Entertainment System (NES). The protagonist is named "Mario" and he can move both directions as well as jump but once the player has gone far into the right, the previous area is inaccessible to the player. There are 15 different types of enemies along with one boss battle. The player can also pick up mushrooms that grants the player abilities like invisibility, using fireball, etc. Along with powerups the player can also gain coins, gain points, and go to secret areas. Super Mario Bros has sold over 40 million copies as of 2020. (Clement, 2021)

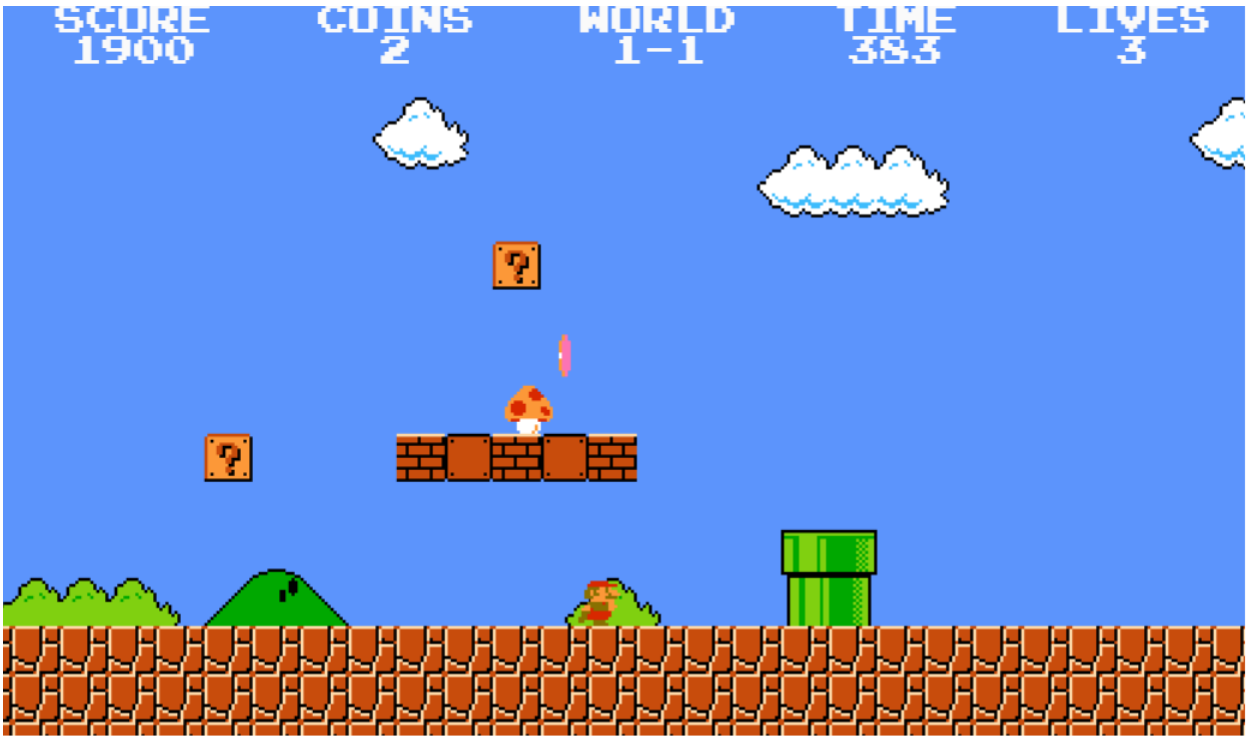


Figure 1: Collectables in Mario

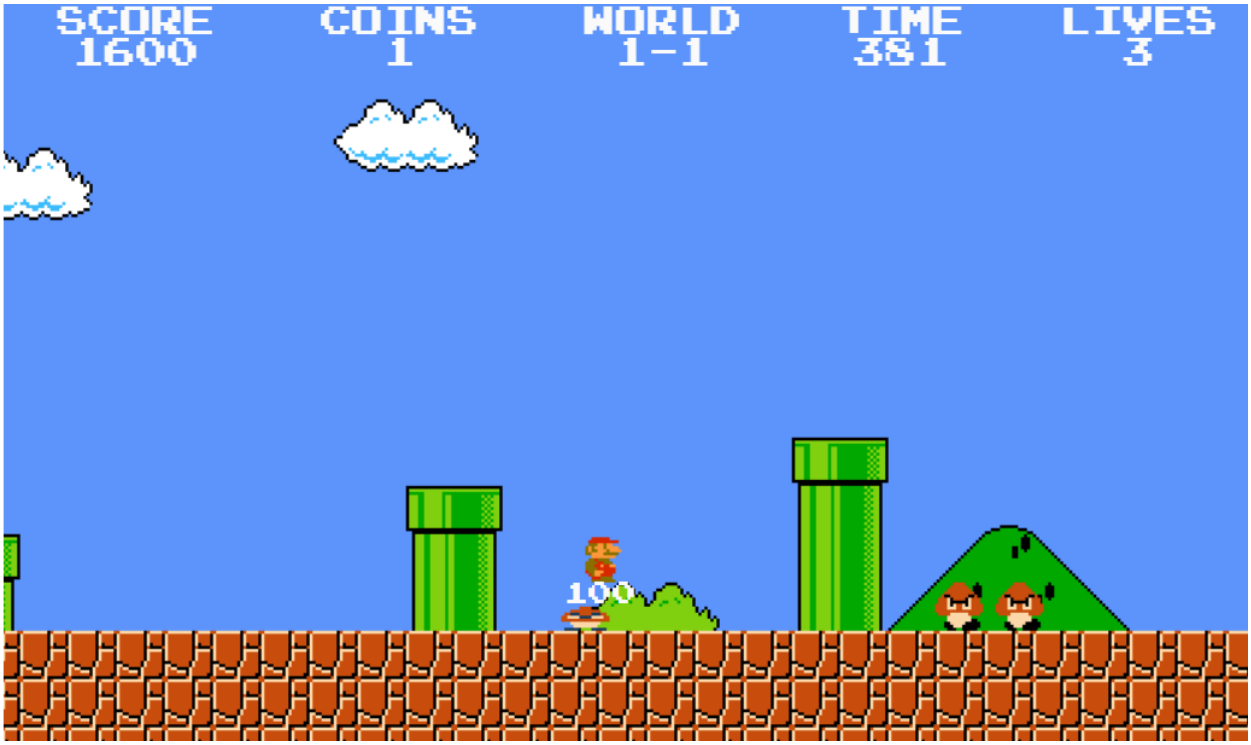


Figure 2: Attacking in Super Mario

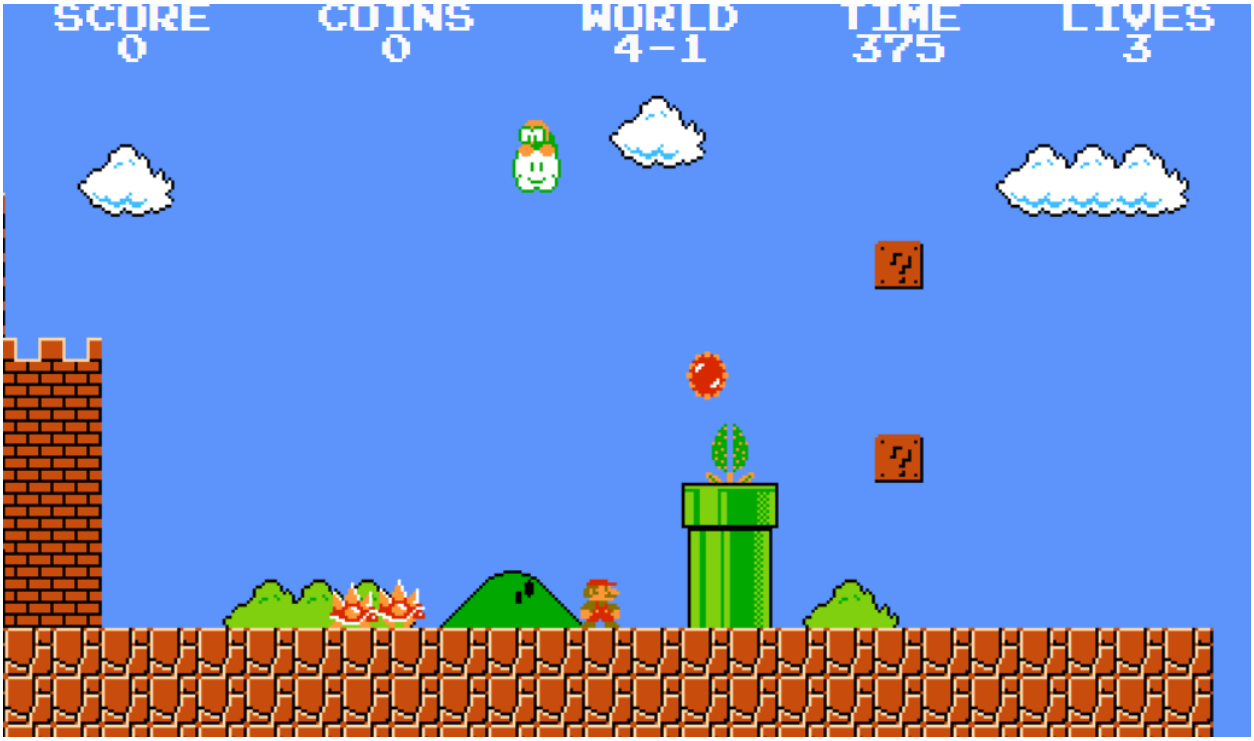


Figure 3: Types of enemies

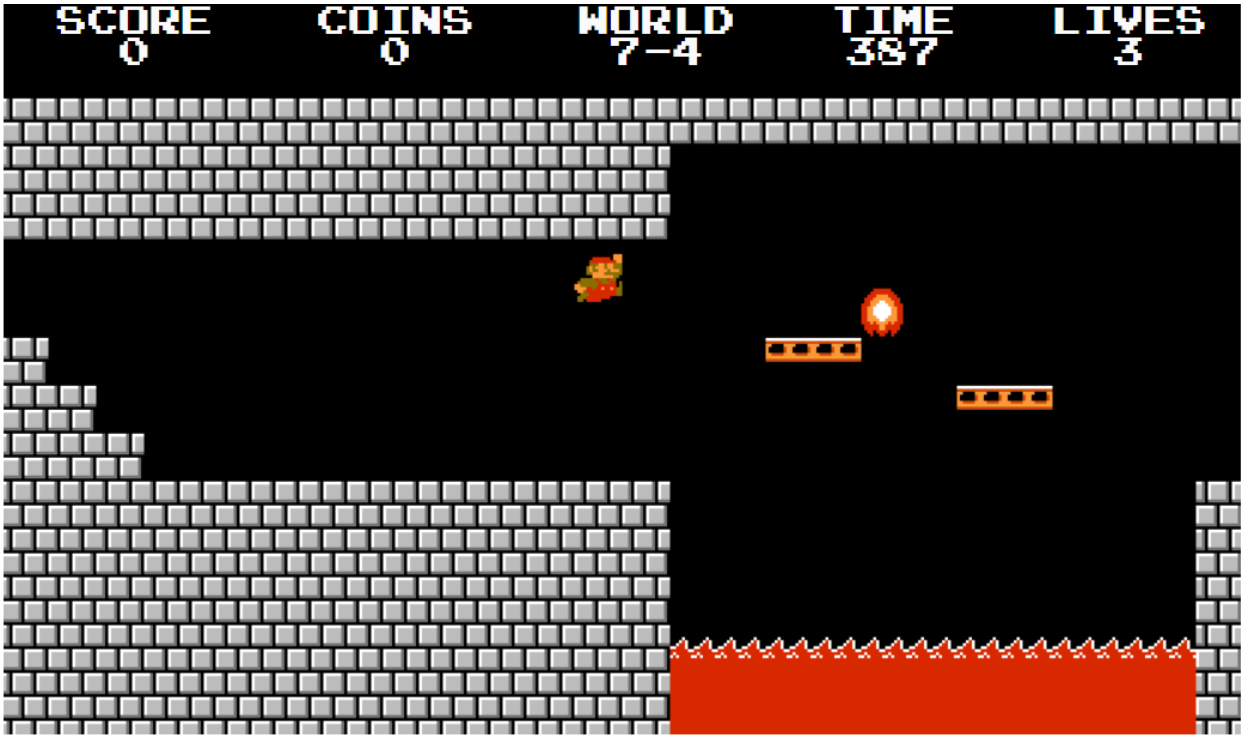


Figure 4: Different levels and obstacles (Mario)

Inside

Inside is a 2D puzzle-platformer game developed and published by a Danish game company in 2016. The player controls a little boy who navigates through different areas while guards try to catch him.

The player can run, walk, swim, climb, push, and pull objects to solve puzzles and jump. The game does not have a life system so whenever the player dies the game starts from the last saved point. There are a number of ways the boy can die like by getting caught by the guards, being drowned, getting shot, etc. It is a dark game with a compelling story. There are two different endings to the game the player can experience. Inside has sold more than a million copies till date.

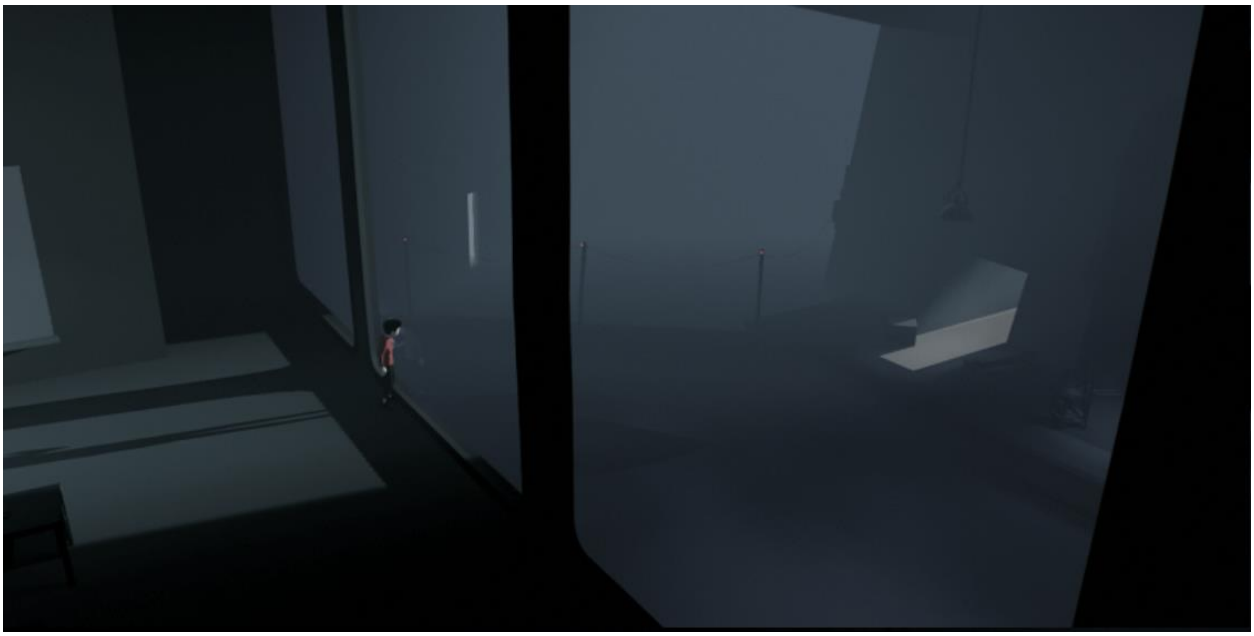


Figure 5: "Inside" game

Figure 6: Player interaction with the game environment (Inside)

Figure 7: Jumping in Inside

Celeste

Celeste is a platformer game that was developed and produced by "Extremely OK Games" in 2018. It was nominated for game of the year award in the same year. In this game the player plays as a young woman named Madeline who has to find her way up Mount Celeste.

The player can run, jump, and climb walls for a certain time. The main game mechanic of this game is the character's ability to perform a mid-air dash in 8 different directions. The gameplay revolves around this single feature, but this mechanic is utilised to the fullest. The character can only use the dash once and has to touch the ground or hit a floating crystal to replenish it. There are items the player can collect while playing like cassette tapes and strawberries that unlock additional game content. Celeste sold nearly a million copies by the end of 2020. (NintendoSoup, 2019) Due to updates and discounted sales the fanbase of Celeste is still growing.

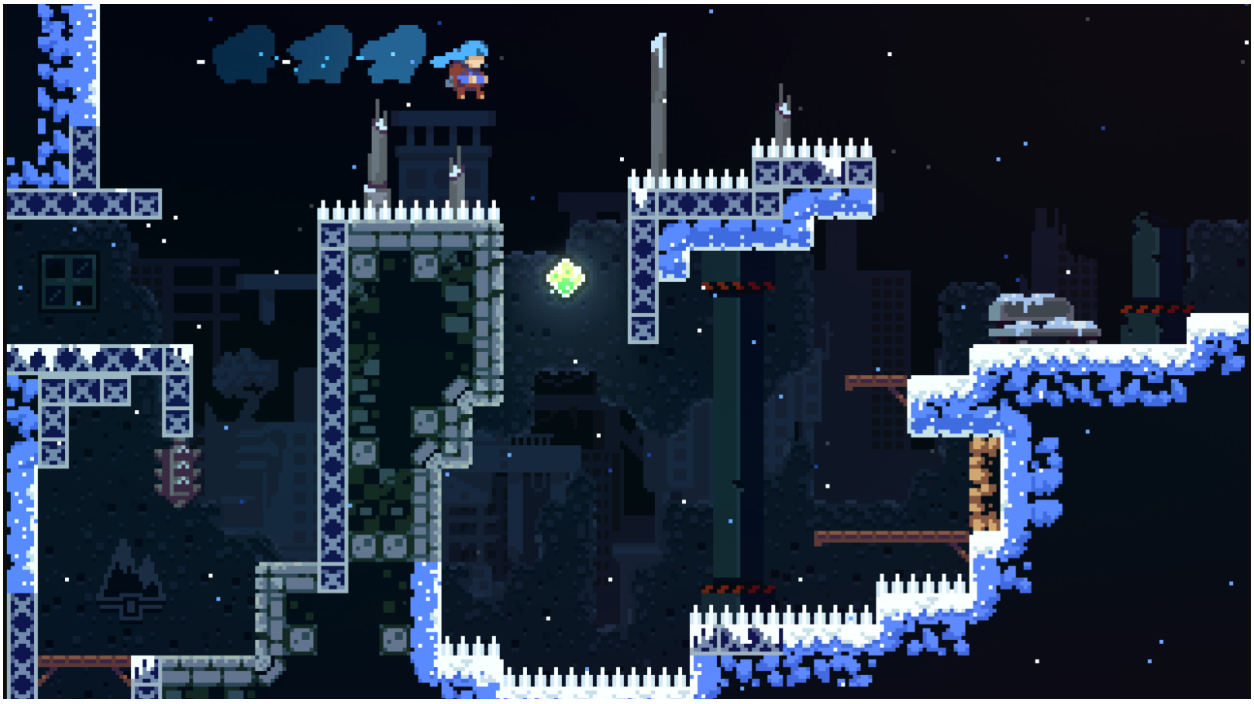


Figure 8: Collectables in Celeste

Figure 9: Player interaction with NPC (Celeste)

Figure 10: Jumping in Celeste

Comparison between similar games:

	Super Mario Bros.	Inside	Celeste	King's Guard
Can jump	Yes	Yes	Yes	Yes
Can attack enemies	Yes	No	No	Yes
Powerups	Yes	No	Yes	Yes
Programming language	Assembly language	C#	C#	C#
Game engine	None	Unity	XNA	Unity
Level type	Side-scroller	Side-scroller	Side-scroller	Side-scroller
Point of view	3rd person	3rd person	3rd person	3rd person
Has spoken dialog	No	No	No	No
Game type	Linear	Linear	Linear	Non-linear
Cost	$25 (When it was released)	$9.99	$19.99	Free
Interconnected level design	No	No	No	Yes
Made in Nepal	No	No	No	Yes

Table 1: Similar games comparison.

Analysis of the comparison:

From the table above, it is clear that being able to jump is important in a platformer game but having enemies and being able to attack is not so important. Still, the character in King's Guard will be able to both jump and attract enemies. Most of the characters in platformer games will have powerups they can get like using fireballs in Mario. King's guard will also have power ups the player can get and use. Most of the modern games are made in Unity using C# so, King's Guard will also follow the same trend. All modern platformer games need to have a side-scrolling level design and most of them are from 3rd person perspective. Same is being done in King's Guard. Having dialogs is not important in a platformer game so King's Guard will also not feature this. Mario, celeste and inside all have linear game design which means that the player is given a predefined way to achieve the games' goal. King's Guard will be an open world 2D platformer in a sense that the player can reach the same destination through multiple ways. This makes it different than other platformer games. It also has interconnected level design which means the players are free to go to whichever part of the game they want. They can visit

the start of the game anytime which cannot be done in other platformer games. There is no intention of creating this game as of now so it will be free for anyone that wants to play.

4. Progress

4.1. System Requirements Specification

4.1.1. Introduction

a. Purpose

The purpose of this document is to create a detailed description of the 2D platformer game, King's Guard. It explains features and the interface of the system.

b. Intended Audience

This document is made for the developers of this system to be used as a reference while developing. Testers can read this document to know the vision of the game and others are free to read it from front to back to understand this project.

c. Project scope

This project will be a 2D platformer game with influence of Metroidvania genre. It will have a unique story line, an ending, and many challenges throughout the gameplay. The game will be made in a way that people with low system specification can also play the game without any trouble.

4.1.2. Overall Description

a. Product perspective

This project is being made with the intention of creating a fun and unique game in a market (Nepal's game market) that is oversaturated with uninspired mobile games.

b. Product features

The product will have all the features a platformer game has like a character that can run and jump but along with it the player can gain and use different abilities, fight enemies, and interact with NPC. The levels will be interconnected so the player can move back and forth unlike most platformer games. The players can save and load the game as well. There will be a story in the game to ensure that the players are engaged. The players can control the character by using the

keyboard and the controls will be very simple. The game can be stopped and exited anytime the player wants.

c. User classes and characteristics

There will be only one user in the system, and it is the player that plays the game. The player will have access to all the features the game has to offer.

d. Operating environment

The project will be made in Unity as it is one of the most popular and powerful game engines while still being easy to use for beginners. C# is the programming language that will be used as it is the only language supported by unity. This system will run on a windows computer.

e. Design and implementation constraints

The system is going to be made for windows computers so it is unlikely that the game can be played on a console or a mobile device. The game will be made available for other platforms in the future, but it is out of the scope of this project. The system will be tested on multiple windows computers to ensure that it works perfectly.

4.1.3. System Features

1. Jumping:

1.1. Description:

It is not a platformer game if the character cannot jump so this is one of the most important features of this system.

1.2. Functionality requirement:

Req-1: The character should jump when the player presses a designated button on the keyboard.

Req-2: The character should jump higher or lower based on how long the player holds the jump button.

Req-3: Level should be designed according to the character's jump height.

2. Double jump:

2.1. Description:

Double jump is one of the abilities the player can gain while playing the game. This mechanic will be introduced to the player after they have gotten used to the basic controls and will allow the player to make further progress in the game.

2.2. Functionality requirement:

Req-1: The character should gain the ability before they can use it.

Req-2: To perform a double jump the player needs to press the jump button while in air.

Req-3: Certain areas should be designed in such a way that the player cannot get access to it until they have double jump ability.

3. Projectiles:

3.1. Description:

The character will gain the ability to use projectiles at later stages of the game. This ability will allow player to do more damage to the enemies and make the game easier.

3.2. Functionality requirement:

Req-1: Players should be able to use this ability with a press of a designated button.

Req-2: It should make the game easier to ensure that players use it.

4. Enemies:

4.1. Description:

Enemies are necessary for games as they are a source of challenge, and challenges are crucial to make any game engaging and fun.

4.2. Functionality requirement:

Req-1: The system needs to have a type of enemy that actively tries to kill the player.

Req-2: The system also needs enemies that don’t actively try to kill the player but will deal damage if the player collides with them.

Req-3: The enemies should get harder to kill as the game progresses.

Req-4: There should be a boss enemy that the player needs to defeat at the end of the game.

5. **Obstacles:**
   - **5.1. Description:**

     Obstacles are another way the game can be made challenging, and they are important for a platformer game.
   - **5.2. Functionality requirement:**

     Req-1: Gap between platforms where the character dies if they fall into it.

     Req-2: Spikes and traps throughout the level.

     Req-3: Platforms that crumble if the character stands in it for too long.
6. **Title Screen:**
   - **6.1. Description:**

     Title screen is the starting screen the players see when starting the game. It helps players to navigate through the game, so it is an important function.
   - **6.2. Functionality requirement:**

     Req-1: The title screen must appear every time the game is launched.

     Req-2: If the player presses the exit button the game must stop running.

     Req-3: If the player presses the start button the game starts or loads from the last saved point.

     Req-4: If the player exits the game while playing or completes the whole game, they must return to the title screen.
7. **Pause Menu:**
   - **7.1. Description:**

     A player might want to take a break, pause, or stop playing the game for a number of reasons. The pause menu allows the players to do so.
   - **7.2. Functionality requirement:**

     Req-1: The game must resume if the player hits the resume button.

     Req-2: The player must return to the title screen if they press the "Exit to menu" button

     Req-3: The game must close if they press the "Exit" button.

4.1.4. External Interface Requirements

- A computer with Unity installed for coding.
- Internet connection.
- Adobe Illustrator and photoshop for creating assets or manipulating existing assets.
- Visual Studio for writing scripts of the game.

4.1.5. Non-Functional requirements

a. Performance requirement:

King's guard will not have a high system requirement and will run on most of the modern computers. The game will use assets that do not require a lot of processing power to render to all laptops and computers should run the game with no problem.

b. Security requirement:

This game will not ask permissions for personal information therefore there is no risk of compromise of sensitive information. The players and testers can directly download the game without any verification process so other people might be able to play the game if they have access to the players computer.

4.1.6. Software Architecture:

The process of playing this game is very simple. The player downloads the game to their device and starts playing. The game data will be stored in the player's computer itself so there is no need for servers. The characters will be controlled by keyboard.

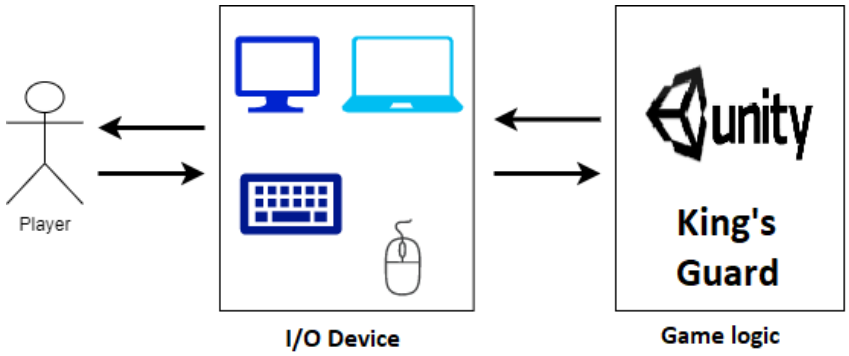


Figure 11: Game architecture

4.2. Use Case

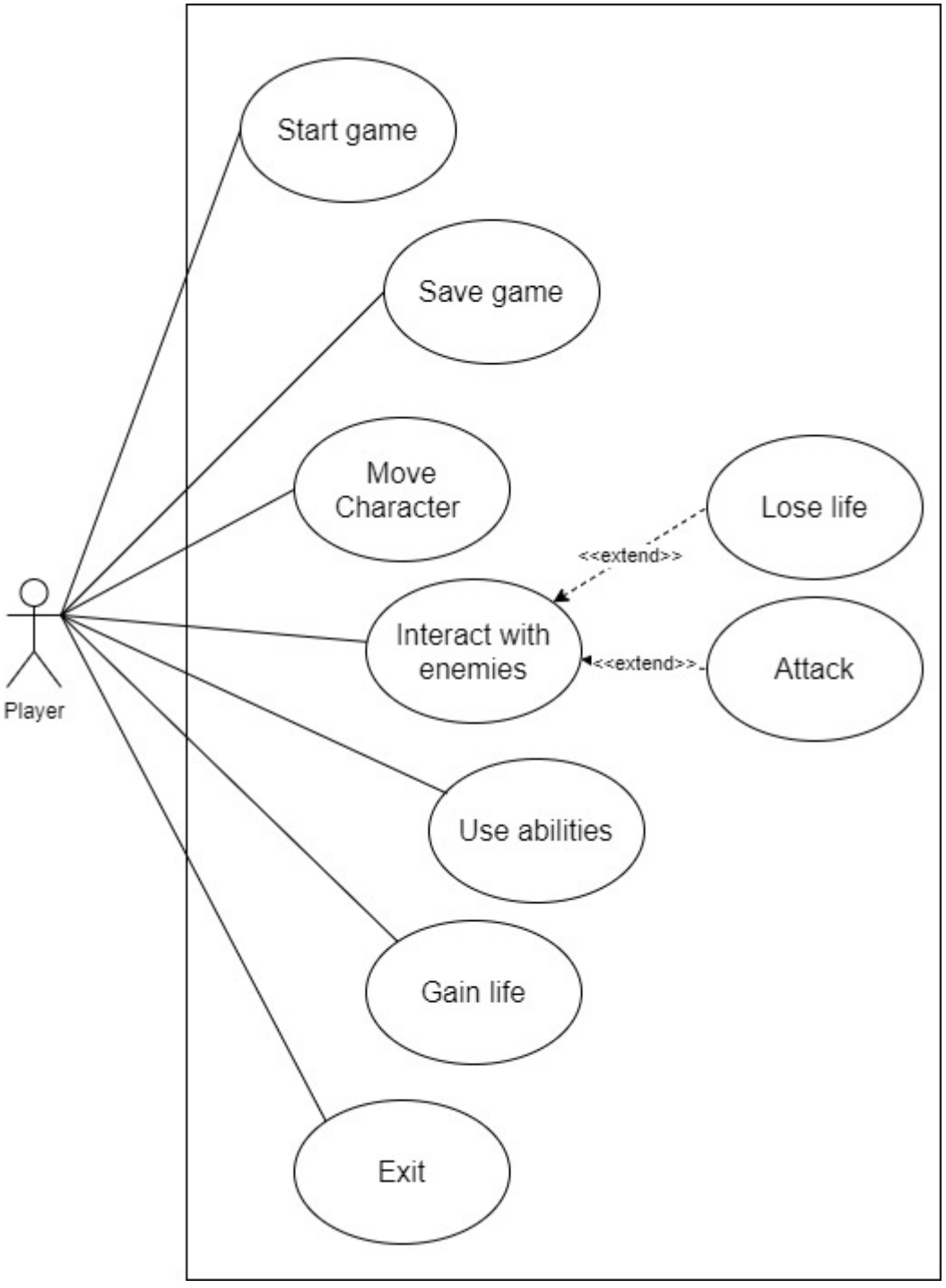


Figure 12: Use case diagram

Expanded use case diagram is in the appendices section (1. Use Case)

4.3. Activity diagram

Activity diagram for each use case is given bellow:

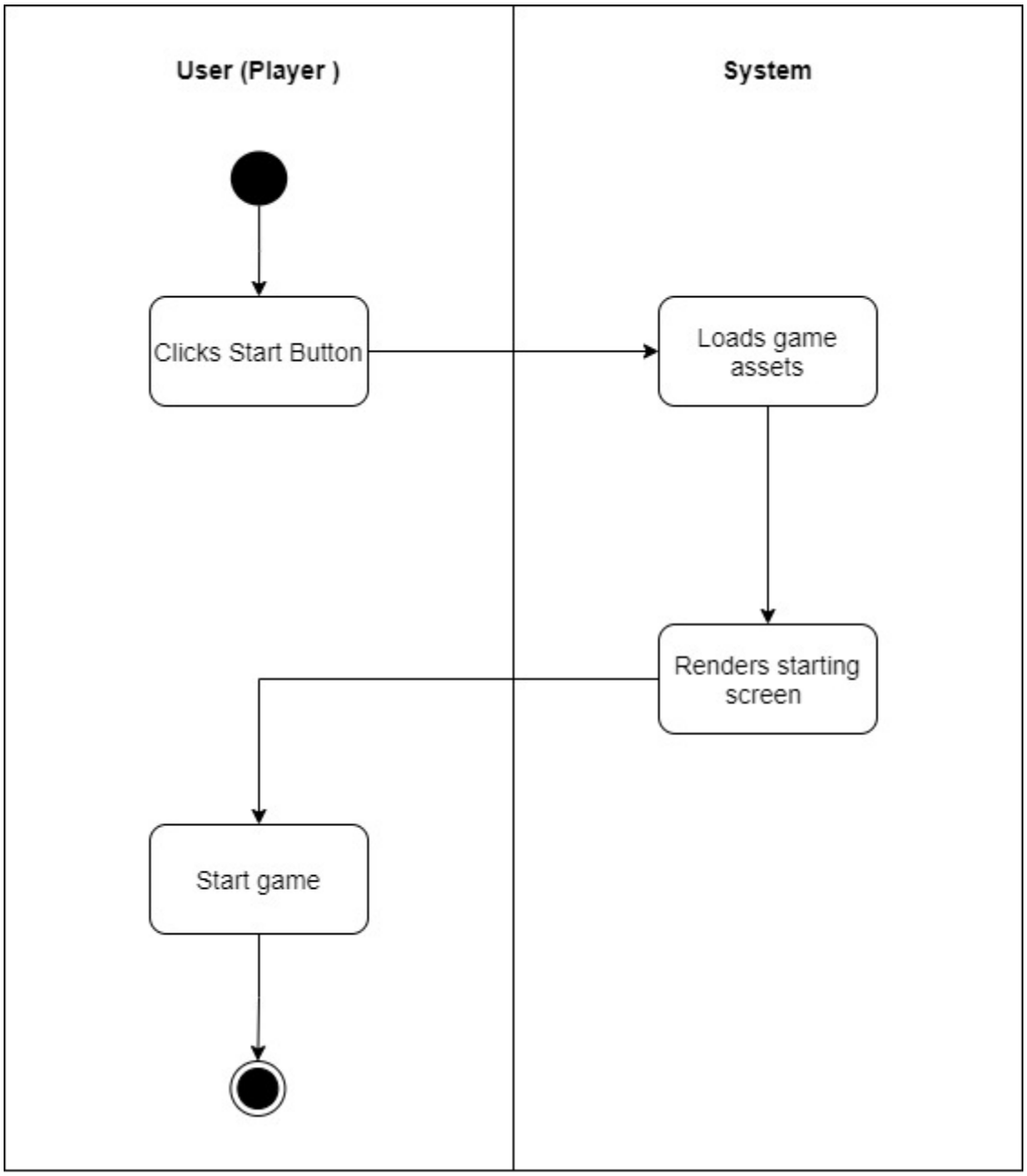


Figure 13: Activity diagram for start game

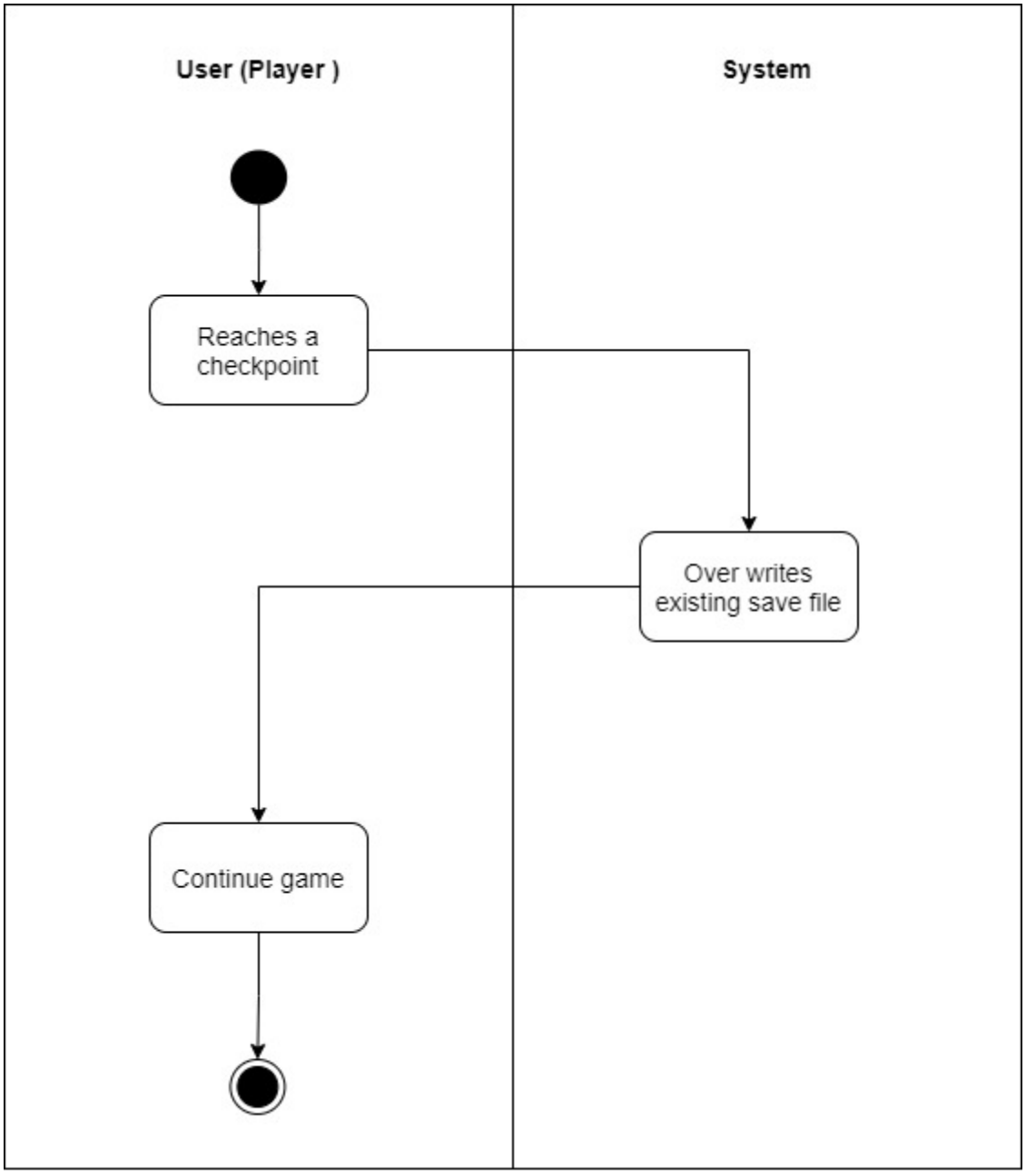


Figure 14: Activity diagram for save game

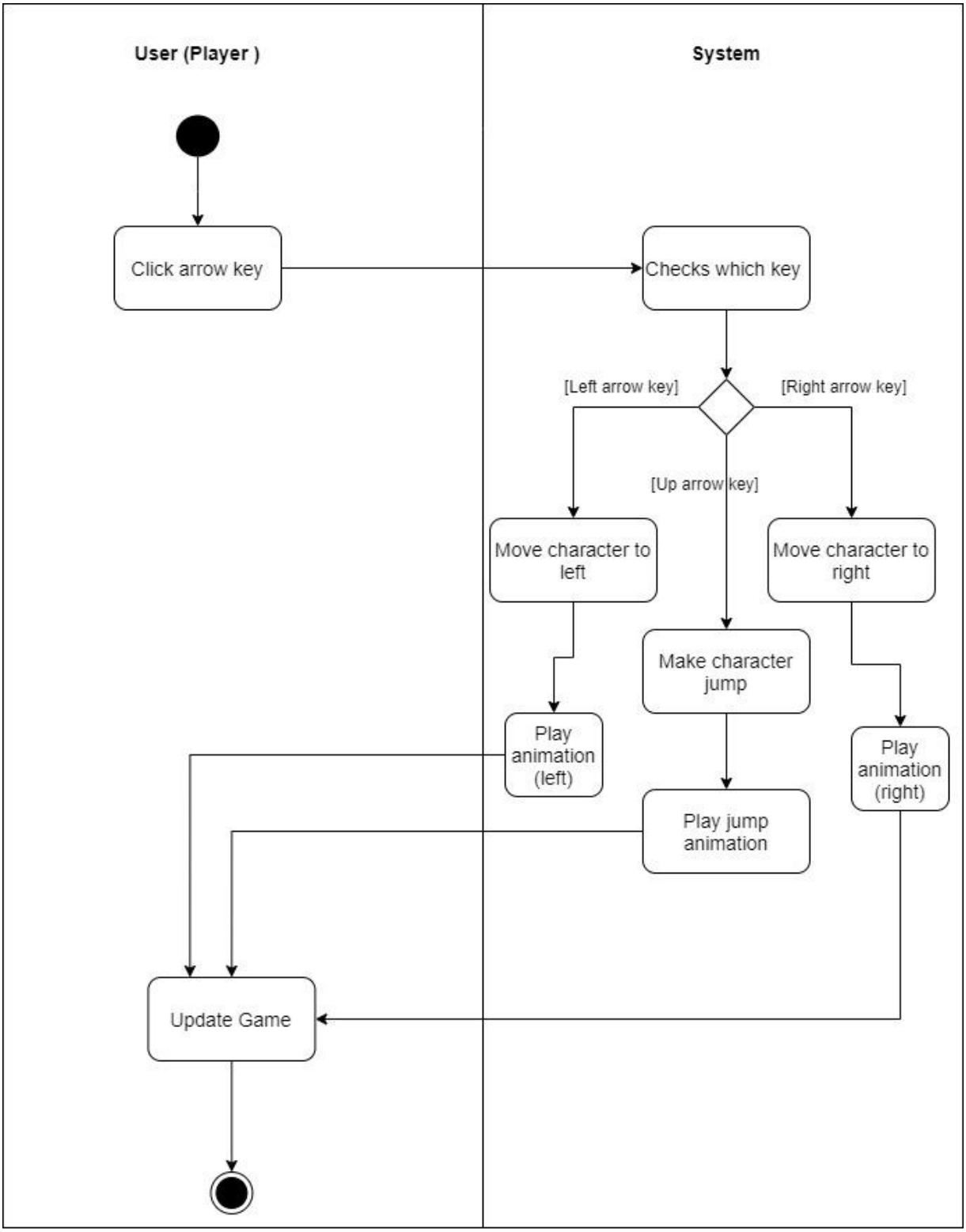


Figure 15: Activity diagram for character movement

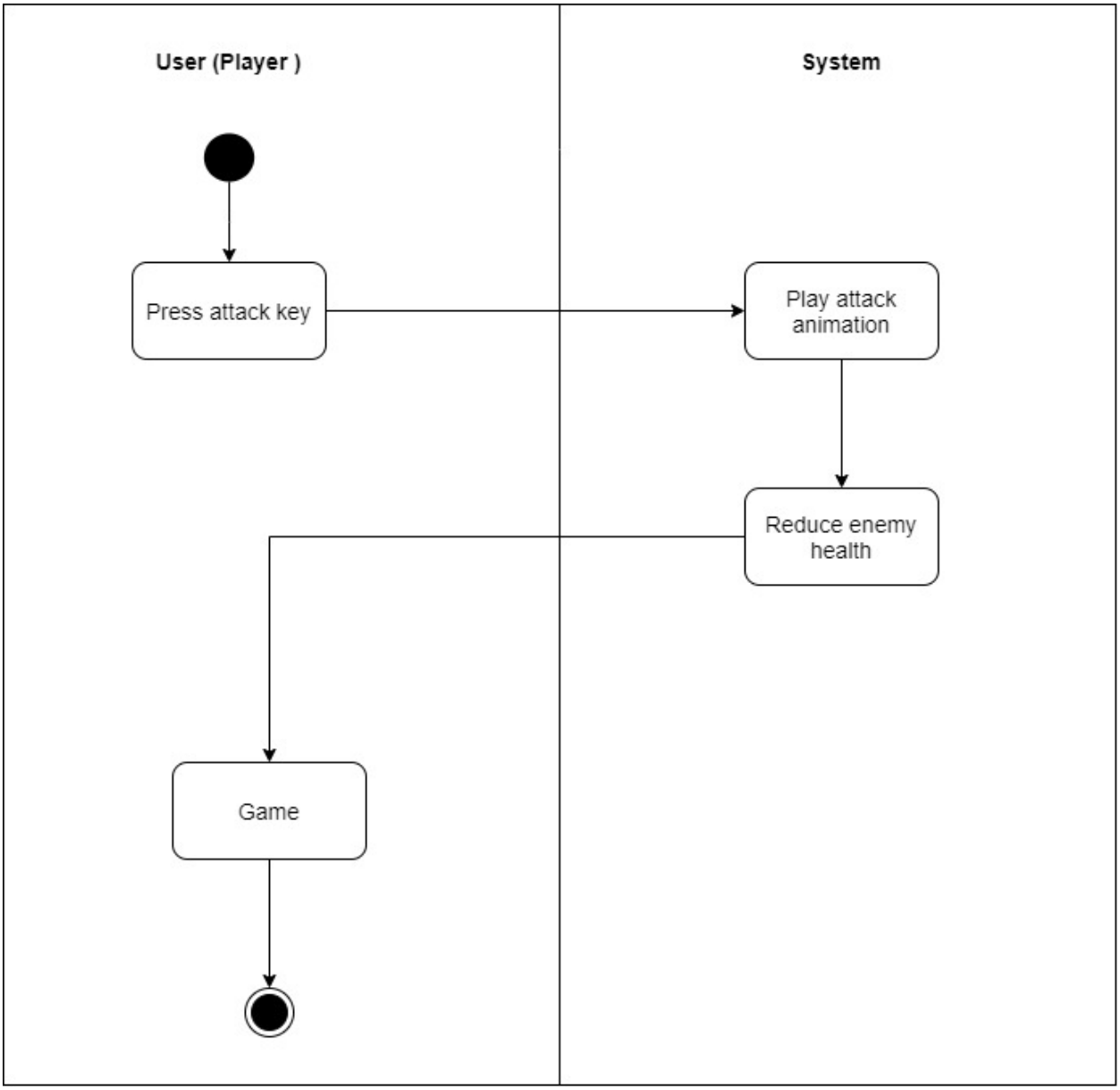


Figure 16: Activity diagram for attack enemies

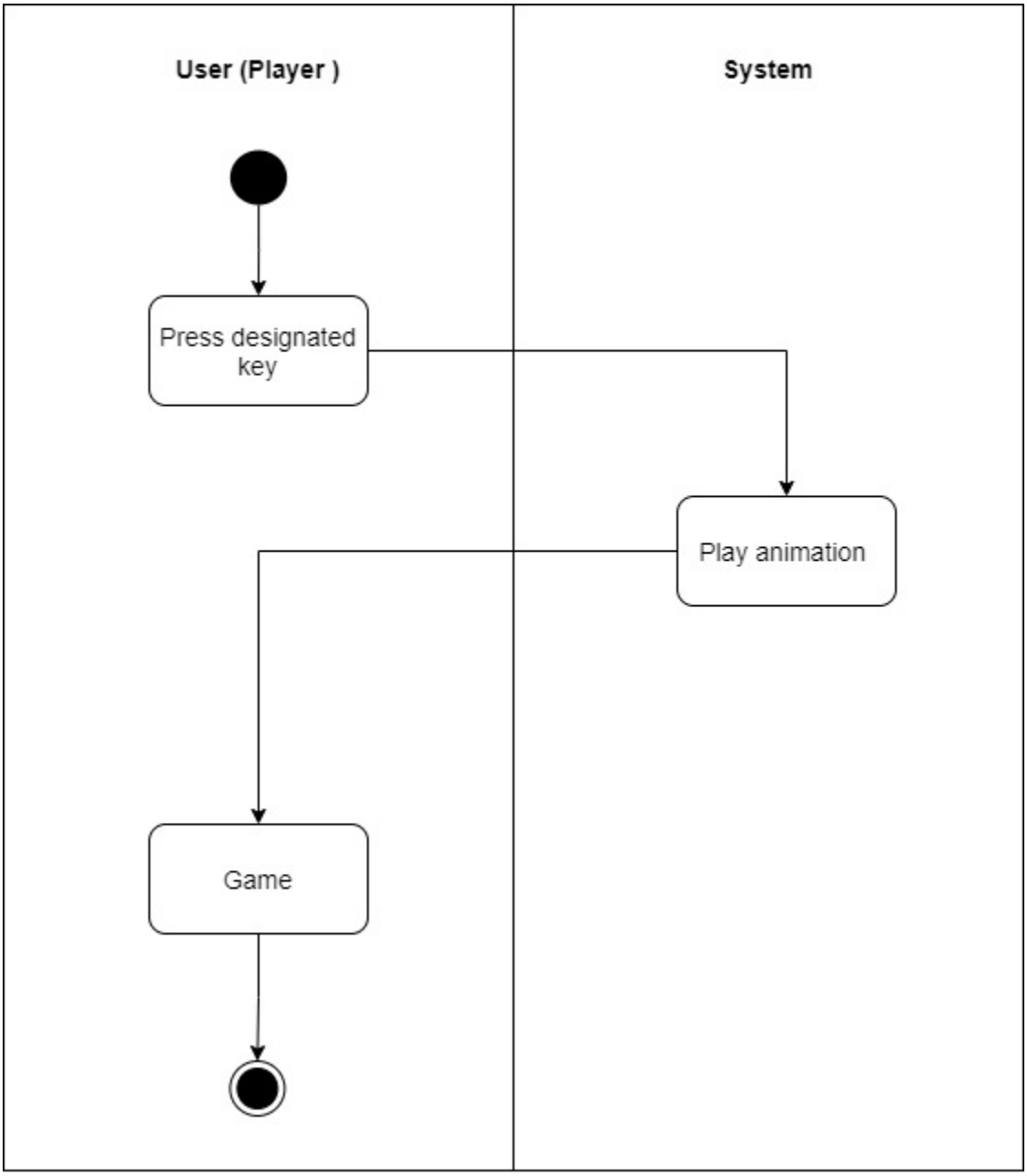


Figure 17: Activity diagram for using ability

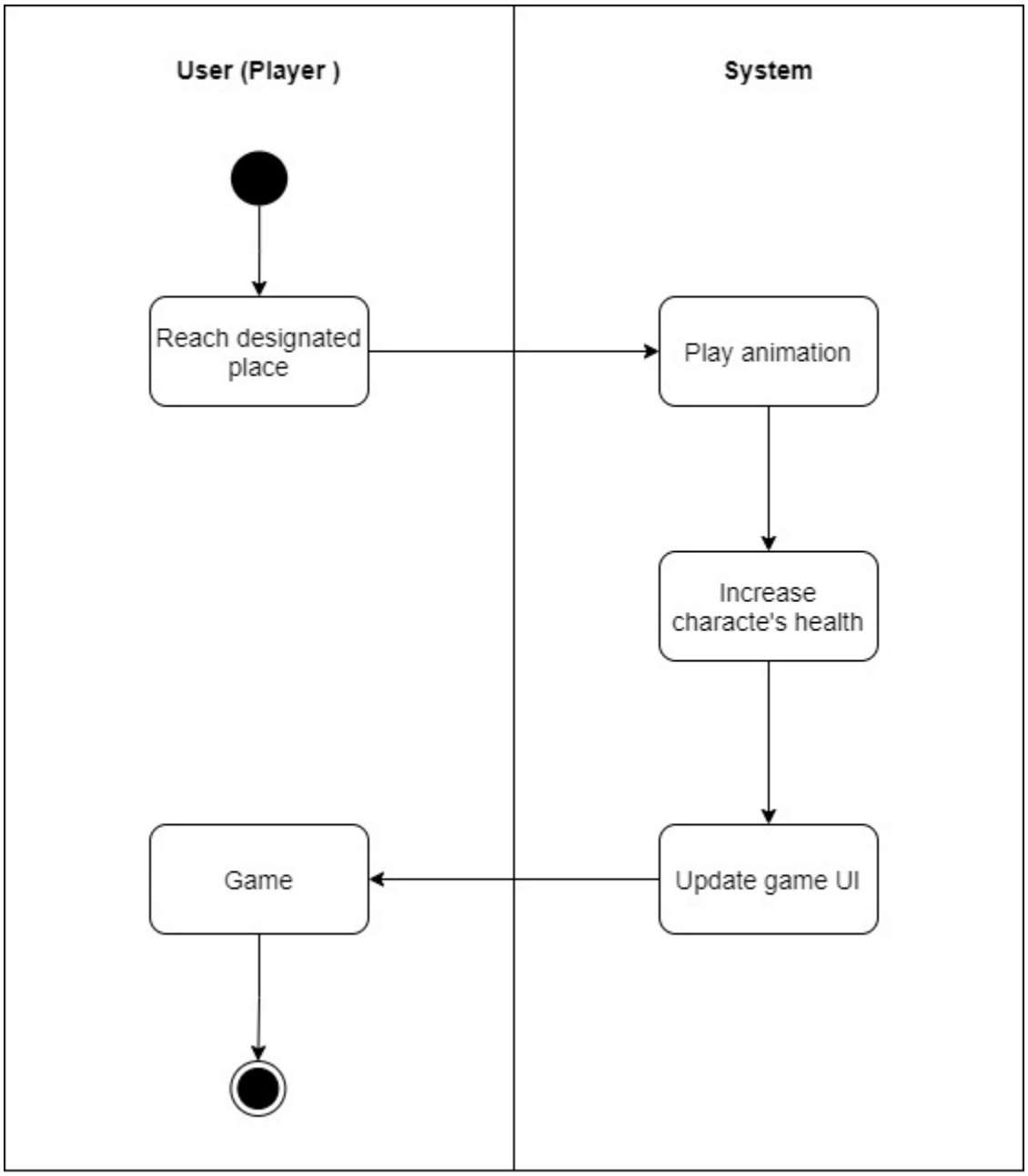


Figure 18: Activity diagram for gaining health

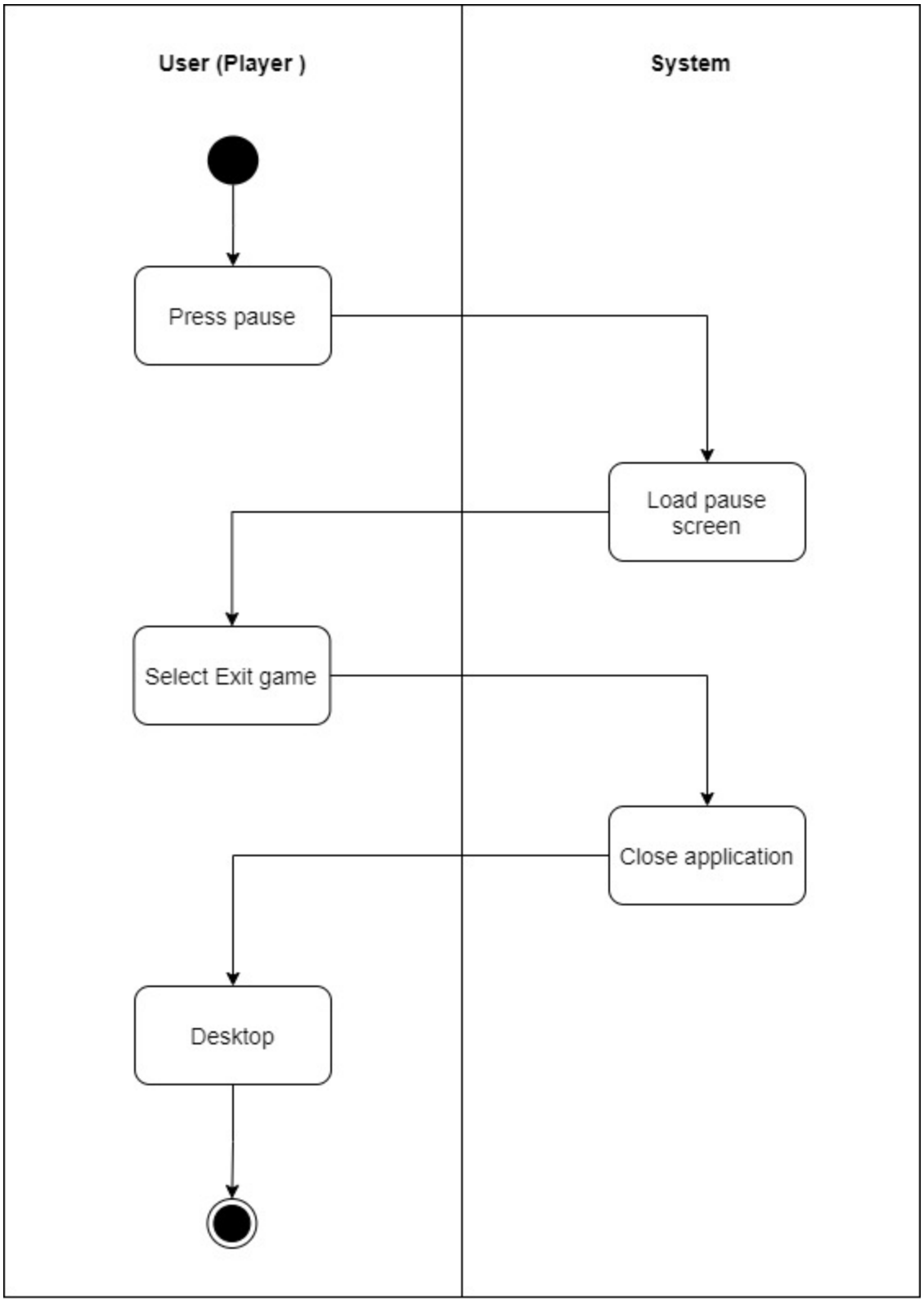


Figure 19: Activity diagram for Exit game

4.4. Collaboration diagram

Collaboration diagram for two of the use cases, Start game and gain life, has been given below:

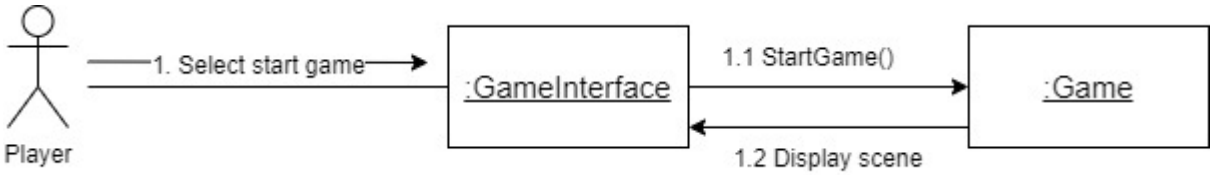


Figure 20: Collaboration diagram for start game

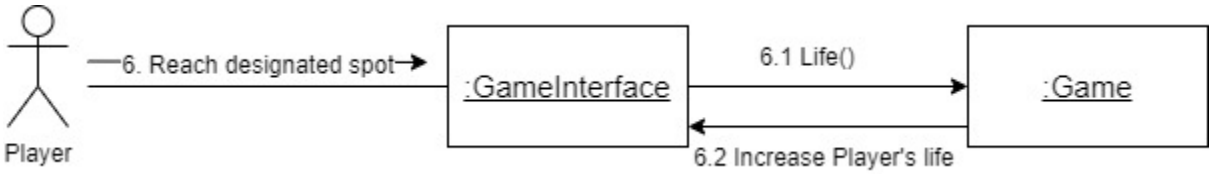


Figure 21: Collaboration diagram for "gain life" use case

Rest of the collaboration diagram can be found in Appendix section (2. Collaboration diagram)

4.5. Sequence Diagram

Sequence diagram for starting game, gaining life and character movement is given below:

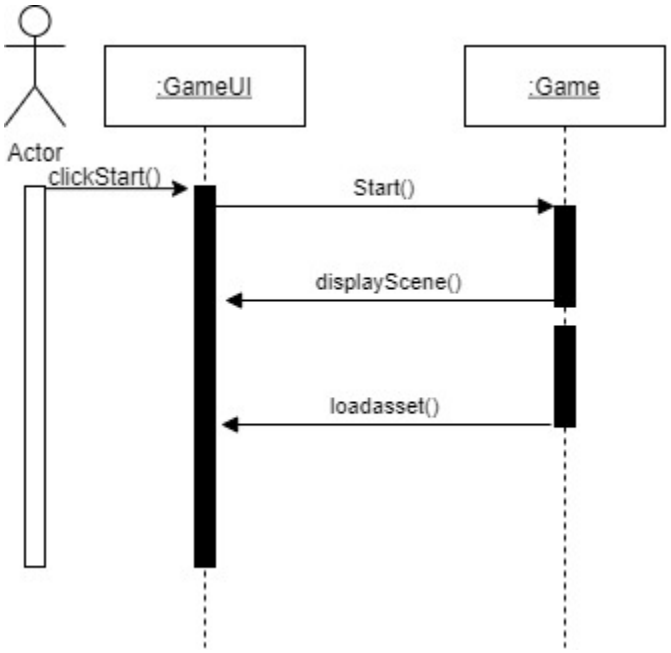


Figure 22: Sequence diagram for start game

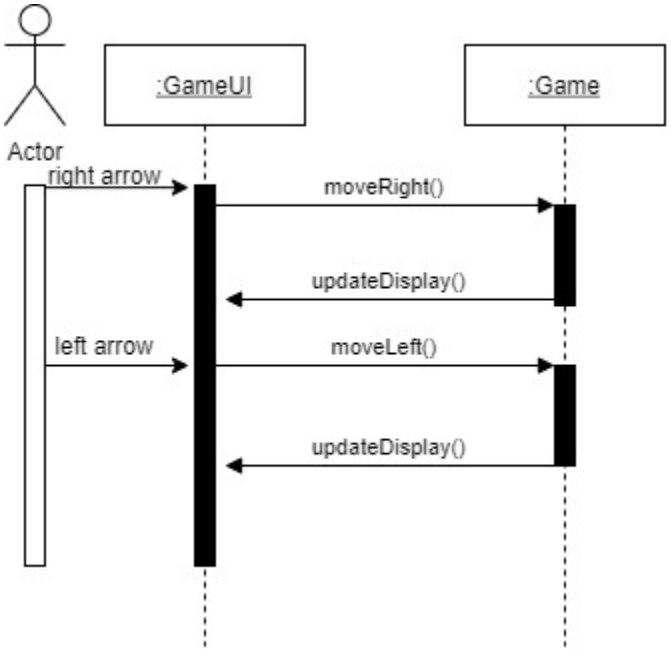


Figure 23: Sequence diagram for character movement

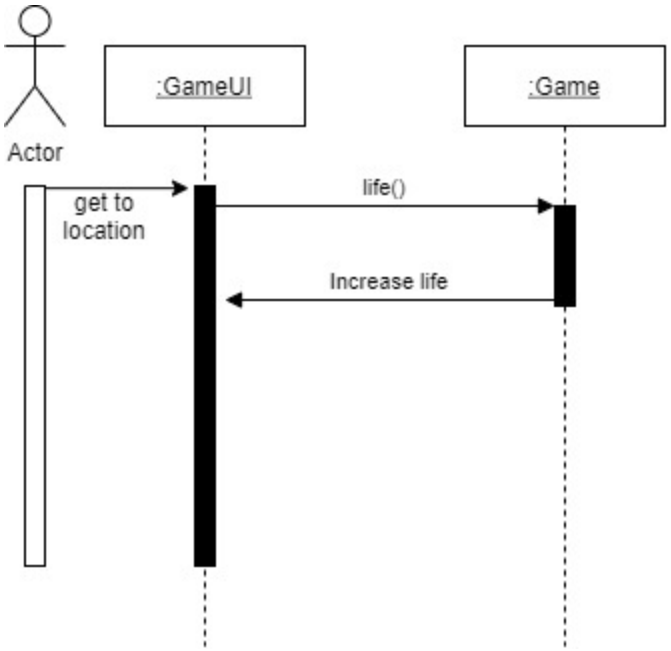


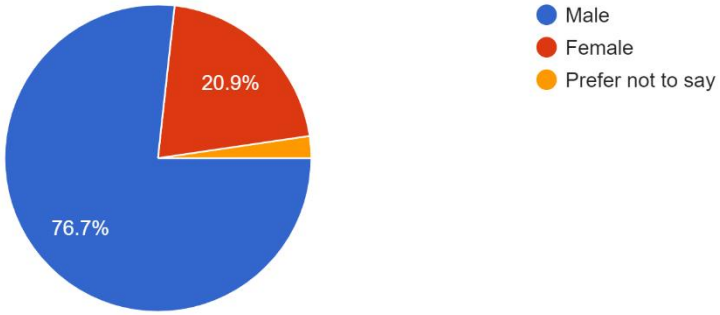
Figure 24: Sequence diagram for gaining life

Rest of the sequence diagram is in appendix (3. Sequence diagram)

4.6. Survey

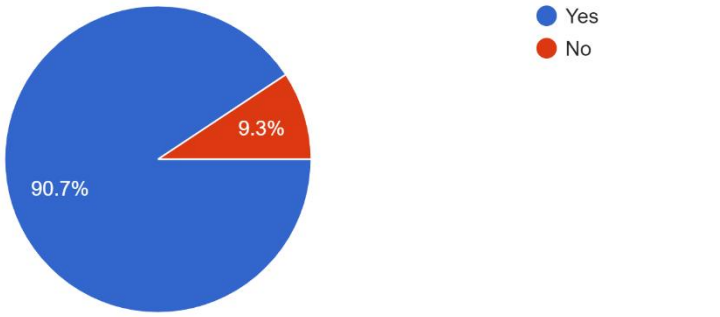
What is your gender?

43 responses



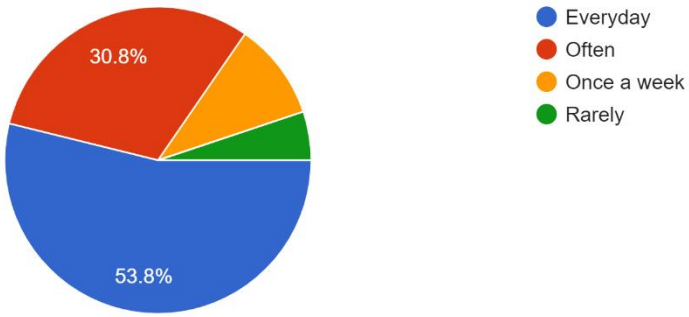
Have you ever played any video games?

43 responses



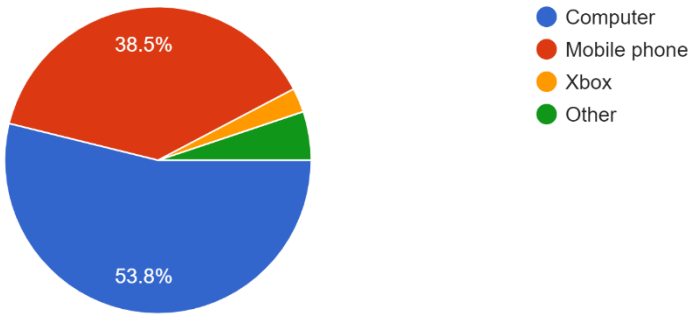
How often do you play games?

39 responses



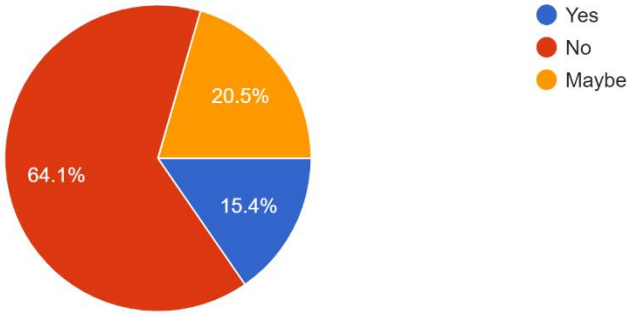
Which platform do you prefer the most to play video games in?

39 responses



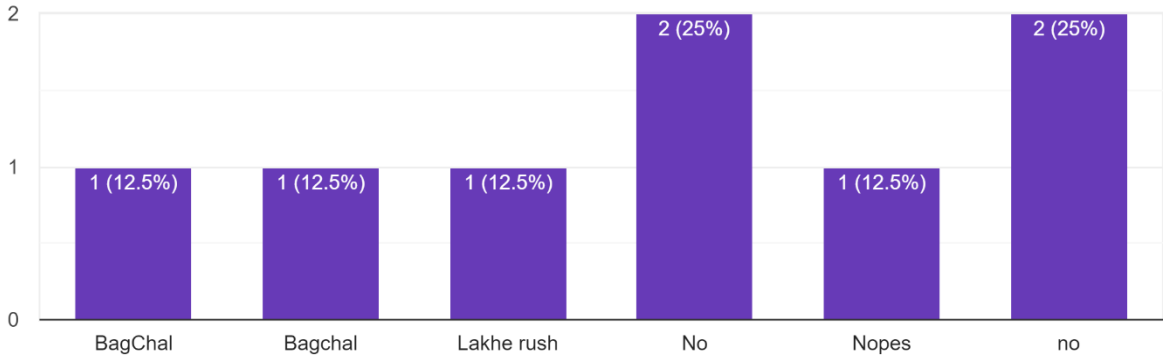
Have you played any computer games that were developed in Nepal?

39 responses



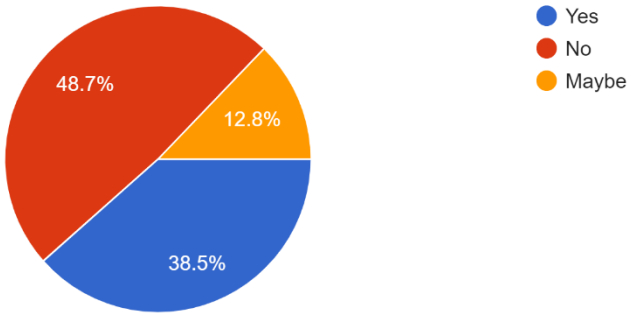
Do you know any computer games that were developed in Nepal? (If yes, mention the name below.)

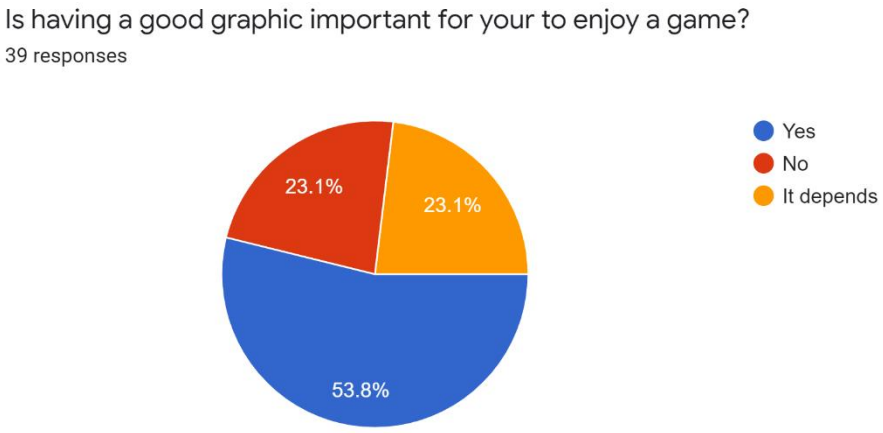
8 responses

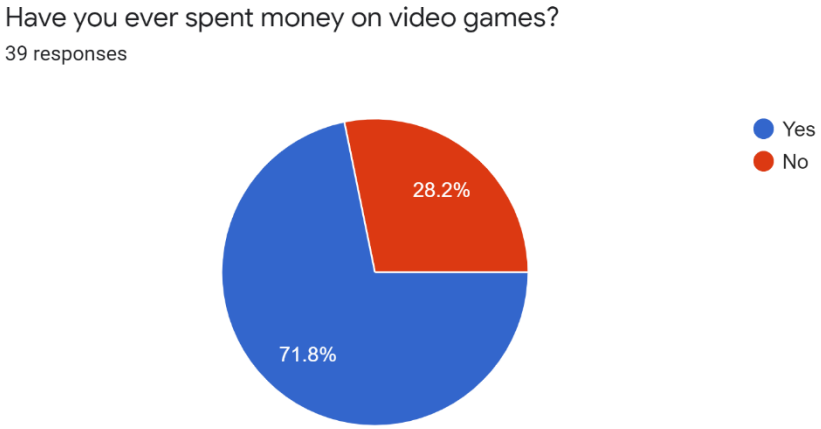


Do you own a high-end gaming pc or laptop?

39 responses







4.7. Assets

An asset in Unity and in game development is anything that goes into a video game. This might be character, music, background images, etc. Unity store provides a lot of free assets that a game developer can easily implement in their game. This is very helpful for a beginner developer who cannot afford to pay for games assets or for developers that are just trying out game development. For King's Guard, most of the assets will be the free assets that are available in the unity store. Free assets are being used for this game for 2 primary reasons:

i. Creating my own assets for the whole game will take a lot of time specially because I do not have artistic skills. So, using the readily available assets will make the development process a lot faster.

ii. As a student buying premium assets or asking artists to make assets that I want is not an option.

As of now I have chosen two different asset packs:

i. "Platformer Set" by Szadi art. (Unity, 2019)

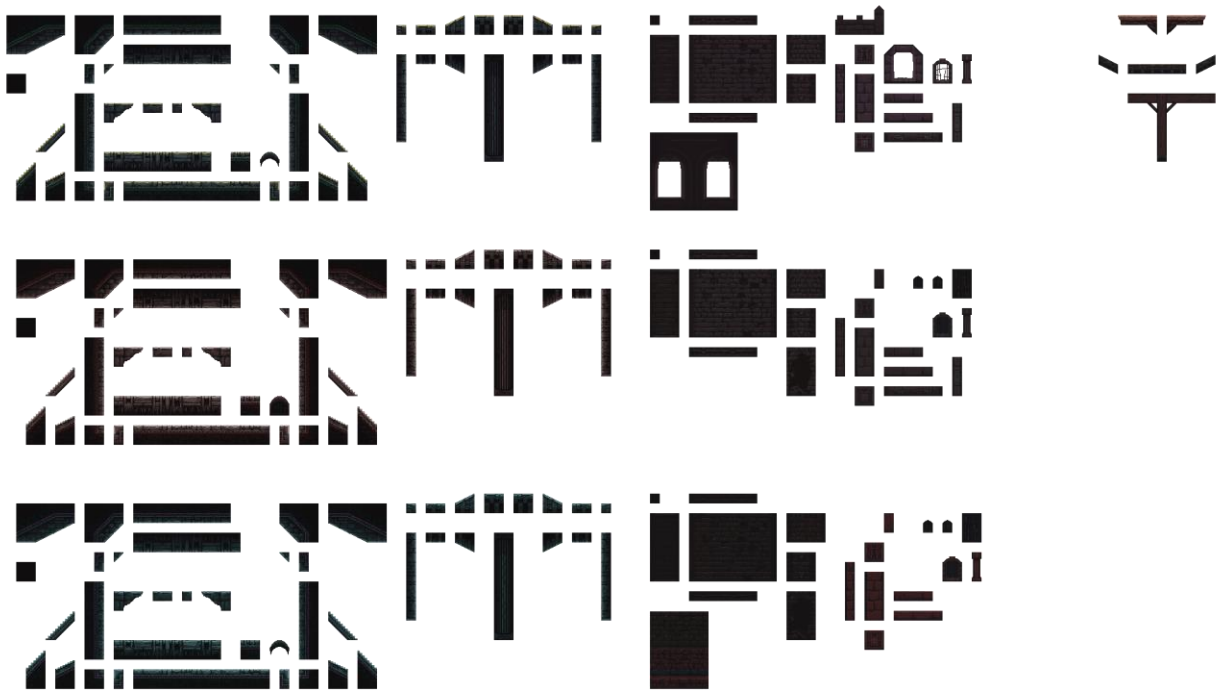


Figure 25: Platformer set asset pack

ii. "Hero and Opponents" by Szadi art. (Unity , 2019)

Figure 26: Hero and opponent asset pack

5. Future work

The first milestone of the project was reached on time and the second milestone for the project was expected to be completed by January 1, 2022. The project might be slightly behind schedule but nothing it cannot recover from. Pre-production and production phase of the first iteration is nearly completed so the first milestone will be completed on time.

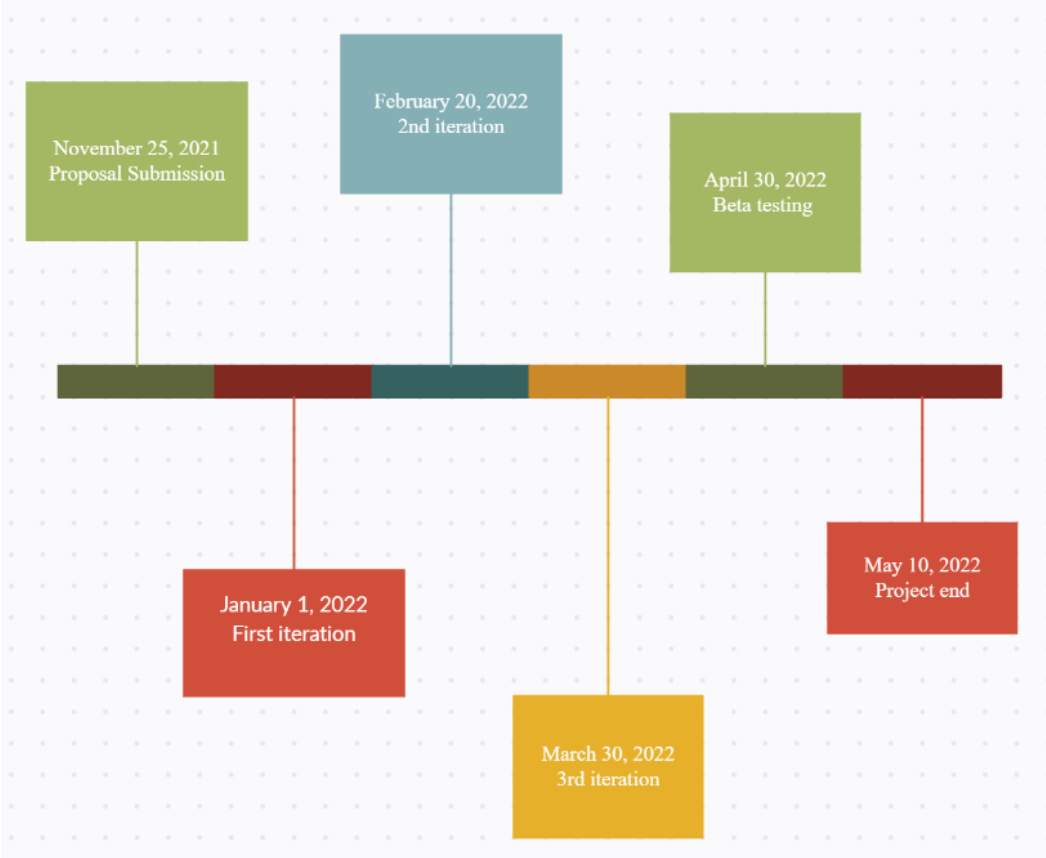


Figure 27: Project Milestones

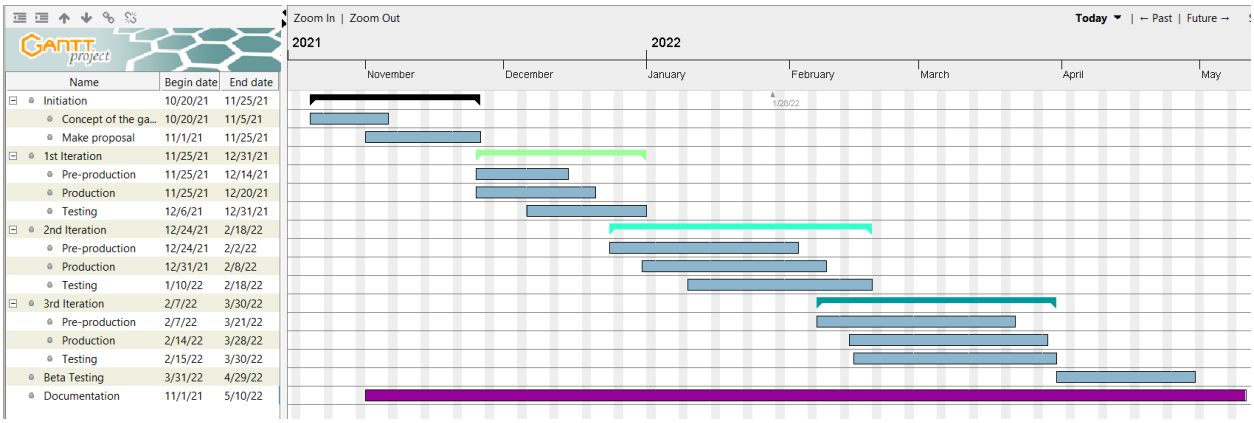


Figure 28: Gantt Chart

5.1. First iteration

The concept and story for the game has been finalised and assets for the game have also been selected. Sprite creation has also been completed. Creating the user interface and adding movement to the character has not been done yet. After this a functionality testing also needs to be done. The first iteration will be completed after that.

5.2. Second iteration

The third milestone set for this project is completing the second iteration by February 20, 2022. In the second iteration, a prototype for the game will be created with all the game's core mechanics using Unity. Unity is being used because it is a free and easy tool to use which is good for a beginner game developer. Core mechanics for the game includes the character's ability to double jump, interact with other game objects, move between levels, enemies, and the ability to attack the enemies. The character losing life when hit by an enemy and the player being able to hit and kill the enemy is also added here. Colliders and physics will be added to the necessary game objects. This prototype will be tested for bugs and to see if it’s a fun game or not. Necessary changes will be made, and bugs will be fixed. All the script of the game will be written in C# using Visual Studio. Unity only uses C# so there are no other options.

5.3. Third iteration

Third iteration is expected to end by March 30, 2022. Rest of the game mechanics like interacting with NPC is added in this stage. The progression of the game will be decided here and whether the game is balanced is also evaluated. Balancing the game means that the player should not feel too powerless or overpowered throughout the game. Making the character too powerful will result in lack of challenges which will make the game less fun. Making the character feel powerless for a long time will result in the game being frustrating to play and players to quit playing. So, finding a balance is important in order for the game to be enjoyable and for the players to keep playing. Audio and sounds will also be added at this stage. Free sound assets will be used, or custom sounds will be created. Tools like sound forge or adobe audition will be used to record or edit sounds. At the end the game is tested again to ensure there is no bug and all the functions are working as intended.

5.4. Beta testing

In this stage the game will be given to people that want to beta test the game. This will help to make sure the game is compatible in other devices as well and the more people play test it the more bugs and missing functionality can be found. The game will either be sent directly to the beta testers, or a simple website will be created which will allow people that want to beta test to directly download it. After testing, they will be asked questions regarding the game and bugs, or missing functionality will be identified. Bugs will be fixed and changes that can be made will be made. The development of the project will end after this stage.

5.5. Documentation

Reporting for a project is very important so, the documentation of this project will be done side by side with the development of the game. This document will record every aspect of the game development process in detail. Microsoft Word will be used to write the report and it will later be converted to PDF. The project is completed when the development and documentation process is completed.

Bibliography

Beattie, A., 2021. *Investopedia.* [Online]
Available at: https://www.investopedia.com/articles/investing/053115/how-video-game-industry-changing.asp
[Accessed 8 November 2021].

Clement, J., 2021. *statista.* [Online]
Available at: https://www.statista.com/statistics/985770/mario-unit-sales/
[Accessed 13 Decemebr 2021].

Clement, J., 2021. *statista.* [Online]
Available at: https://www.statista.com/statistics/748044/number-video-gamers-world/
[Accessed 8 November 2021].

History, 2019. *History.* [Online]
Available at: https://www.history.com/topics/inventions/history-of-video-games
[Accessed 10 December 2021].

Master Class, 2021. *Master Class.* [Online]
Available at: https://www.masterclass.com/articles/platform-game-explained#what-is-a-platform-game
[Accessed 10 December 2021].

NintendoSoup, 2019. *NintendoSoup.* [Online]
Available at: https://nintendosoup.com/celeste-has-nearly-sold-a-million-copies-no-plans-for-a-sequel-why-chapter-9-is-free/
[Accessed 13 December 2021].

Northfield, R., 2018. *Engineering and Technology.* [Online]
Available at: https://eandt.theiet.org/content/articles/2018/07/gaming-s-golden-age-top-10-retro-vintage-arcade-classics/
[Accessed 9 December 2021].

Rana, P. S., 2020. *The Kathmandu Post.* [Online]
Available at: https://kathmandupost.com/science-technology/2020/10/06/starting-small-

aiming-big-game-development-in-nepal
[Accessed 12 December 2021].

Soffe, E., 2014. *TEDED.* [Online]
Available at: https://blog.ed.ted.com/2014/09/11/the-past-present-and-future-of-video-games/
[Accessed 7 December 2021].

The Strong , 2021. *The Strong.* [Online]
Available at: https://www.museumofplay.org/video_games/
[Accessed 9 December 2021].

Unity , 2019. *Unity asset store.* [Online]
Available at: https://assetstore.unity.com/packages/2d/characters/hero-nad-opponents-animation-140776
[Accessed 14 December 2021].

Unity, 2019. *Unity asset store.* [Online]
Available at: https://assetstore.unity.com/packages/2d/environments/platformer-set-150023
[Accessed 14 December 2021].

Veresockaya, S., 2020. *kevuru games.* [Online]
Available at: https://kevurugames.com/blog/why-are-aaa-games-so-expensive-everything-you-need-to-know-about-the-budget/
[Accessed 12 November 2021].

Appendix

1. Expanded Use Case diagram

a. Start game

Use case name: Start game

Scenario: The player is in the title screen and is about to start the game by pressing the Start button.

Actors: Player

Preconditions: The game has already been open and loaded.

Post Conditions: The game starts to run, and the player can control the character.

Flow of activities:

Player	Game
	The game loads the title screen and displays the start button.
The player clicks the start button.	
	The game starts the first section of the game.

Table 2: Expanded use case of start game

b. Save game

Use case name: Save Game

Scenario: After the player has made progress in the game and reaches a checkpoint the game saves the player's progress.

Actors: Player

Preconditions: The player has reached a checkpoint.

Post Conditions: The game progress is saved.

Flow of activities:

Player	Game
The player reaches a checkpoint	
	The game automatically saves the player's progress.

Table 3: Save game (Expanded use case)

c. Move Character

Use case name: Move Character

Scenario: After the game has been started the player is allowed to move and jump around.

Actors: Player

Preconditions: The game has started.

Post Conditions: The player can make progress in the game.

Flow of activities:

Player	**Game**
The player presses the left arrow key.	
	The game moves the character to the left.
The player presses the right arrow key.	
	The game moves the character to the right.
The player presses the jump key.	
	The character jumps.

Table 4: Move character (Expanded use case)

d. Interact with Enemies

Use case name: Interact with enemies

Scenario: After moving forward the player meets enemies.

Actors: Player

Preconditions: The game has started, and the player moves forward.

Post Conditions: The player makes progression in the game.

Related Use cases: Lose life, Attack

Flow of activities:

Player	**Game**
The player presses the attack button.	

	The game lowers the enemies hit point.
The player takes damage from the enemy.	
	The game lowers the character's hit point.
The player kills the enemy.	
	The game plays an animation of the enemy's death.

Table 5: Interact with enemies (Expanded use case)

e. Use abilities

Use case name: Use abilities

Scenario: The character has made progress in the game and has access to the abilities.

Actors: Player

Preconditions: The player has access to the abilities.

Post Conditions: The player can use the ability.

Flow of activities:

Player	**Game**
The player presses the designated button to use ability.	
	The game plays an animation for the ability.

Table 6: Use abilities (Expanded use case)

f. Gain life

Use case name: Gain Life

Scenario: After the player has taken damage, they can go to a designated spot to gain their lost life back.

Actors: Player

Preconditions: The player has lost life.

Post Conditions: The player has full health.

Flow of activities:

Player	Game
The player goes to a designated place to rest.	
	The game replenishes the players health to the max

Table 7: Gain life (Expanded use case)

g. Exit

Use case name: Exit

Scenario: After the player is done playing and wants to exit, they can press the exit button to quit the game.

Actors: Player

Preconditions: The game is running.

Post Conditions: The game closes.

Flow of activities:

Player	Game
The player presses the pause button.	
	The game pauses and displays the pause menu.
The player selects the "exit" option on the pause menu.	
	The game closes.

Table 8: Exit (Expanded use case)

2. Collaboration diagram

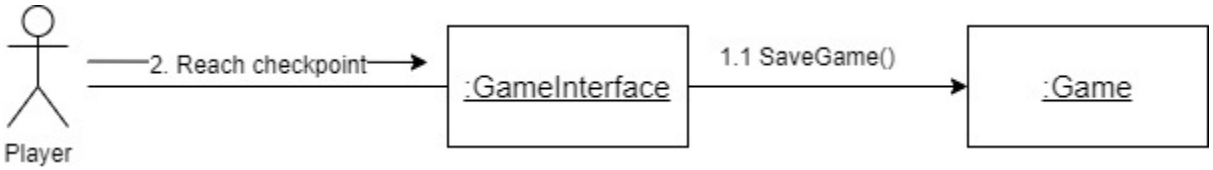


Figure 29: Collaboration diagram for save game

Figure 30: Collaboration diagram for Move character

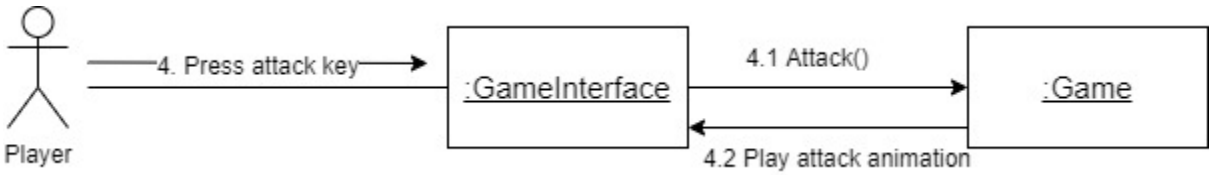


Figure 31: Collaboration diagram for attack enemy

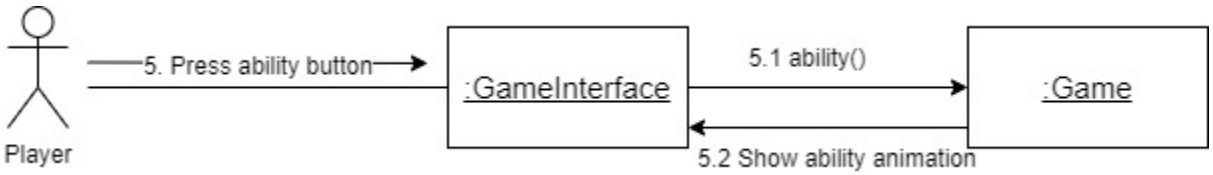


Figure 32: Collaboration diagram for use ability

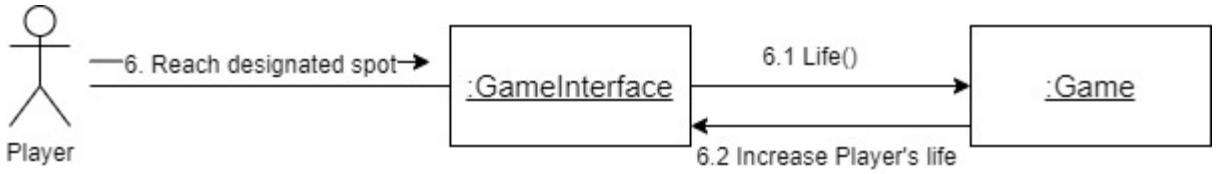


Figure 33: Collaboration diagram for gain life

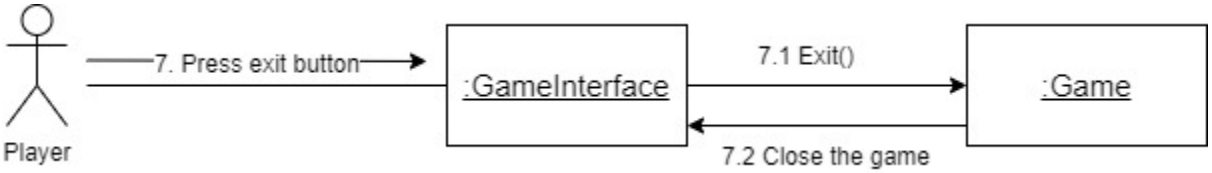


Figure 34: Collaboration diagram for Exit game

3. Sequence diagram

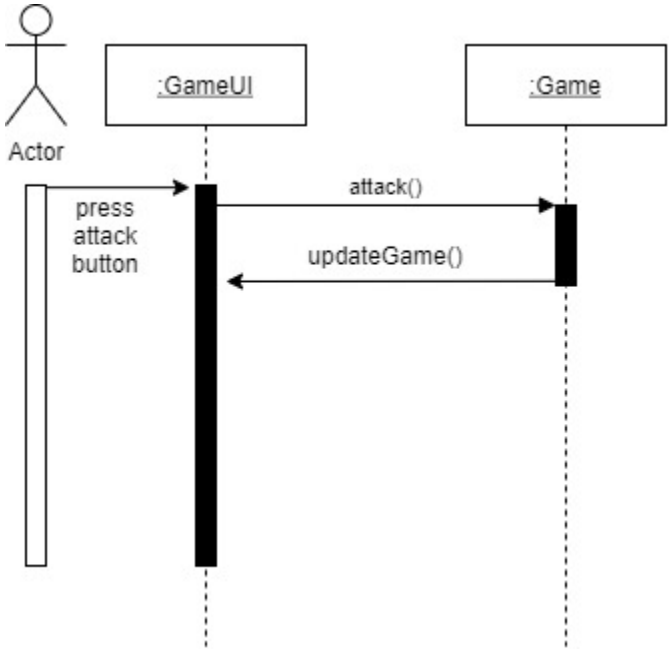


Figure 35: Sequence diagram for attacking enemy

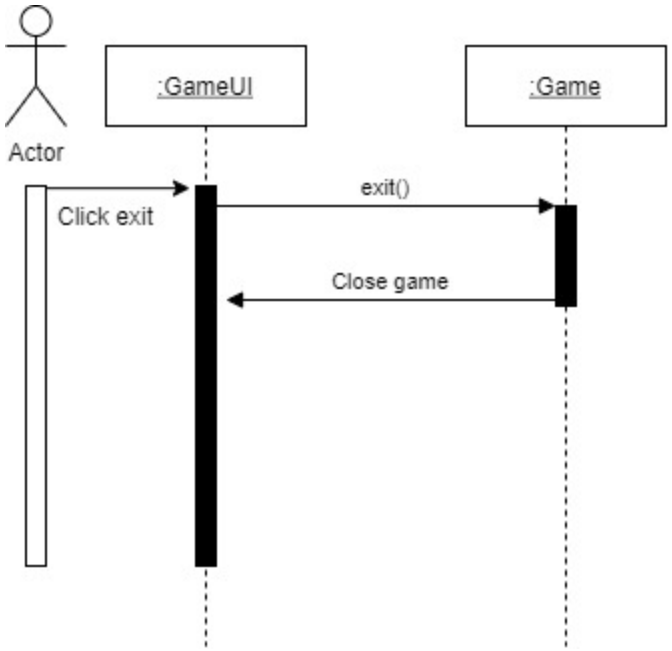


Figure 36: Sequence diagram for exiting game