Politechnika Wrocławska

Faculty of Computer Science and Management
Field of study: **COMPUTER SCIENCE**

Bachelor Thesis

Role play game with "Quiz it" Implementation

ALI TAHSIN SARIBUDAK

keywords:
Unity
C# Programming language
Game Development

short summary:
The project contains the design and implementation of a mixed type of game which are role playing games and quiz games. The game is for desktop systems and made with Unity.

Supervisor	Dr. Inz. Marek KOPEL …….. Title/ degree/ name and surname		
The final evaluation of the thesis			
Chairman of the Diploma Examination Committee	..... Title/ degree/ name and surname	..... *grade*	……. *signature*

For the purposes of archival thesis qualified to: *

a) Category A (perpetual files)
b) Category BE 50 (subject to expertise after 50 years)

* Delete as appropriate

stamp of the faculty

Wrocław, 2021

Abstract

The main purpose of the project is to build an RPG game with quiz implementation. I aimed to get information by examining similar games and applications that might be useful for my game. I choose the Unity game engine to program the game. My game is aimed at teaching people some more knowledge while they are playing this game. This work covers the game project based on this knowledge. The documentation begins with an introduction to the subject and research about used technologies and the current game market. The design and implementation chapter has been described by using GDD. The result of this work is a playable alpha version without the quiz implementation. All codes are developed in Visual Studio Code platform and C# programming language.

Streszczenie

Głównym celem projektu jest zbudowanie gry RPG z implementacją quizu. Dążyłem do uzyskania informacji poprzez sprawdzanie podobnych gier i aplikacji które mogą być przydatne dla mojej gry. Do programowania gry wybieram silnik gry Unity. Moja gra ma na celu nauczenie ludzi dodatkowej wiedzy podczas gdy grają w nią. Ta praca obejmuje projekt gry oparty na tej wiedzy. Dokumentacja zaczyna się od wprowadzenia do tematu i poznania stosowanych technologii oraz obecnego rynku gier. Rozdział na temat projektowania i wdrażania został opisany przy użyciu GDD. Wynikiem tej pracy jest grywalna wersja alfa bez implementacji quizu. Cały kod jest utworzony na platformie Visual Studio Code i języku programowania C#.

CONTENTS

1. INTRODUCTION

The increasing number of computer applications is the result of the development of the computer nowadays. From the beginning, the computer industry has witnessed a huge amount of improvement. Computer games history can be traced back to the first game with electronic display, which was built in 1947. In 1962 first game called Spacewar! was created for entertainment purposes and starting from that, the gaming industry kept its improvement in terms of technology up until today's games. Below I would like to talk about one of the most popular genres from nowadays game industry.

1.1 ROLE-PLAYING GAME (RPG)

A role-playing game is a game genre where the player can act as a character in some nicely detailed world. Most of the well-known computer role-playing games are based on their tabletop versions such as Dungeons & Dragons using the same terminology.

Controlling the character, completing quests, hunting monsters, playing as a team with other players in some party, are the basic features in RPGs. One of the greatest attractions of this genre is that the more you play, the character you control or "characterize" let's say gets stronger, increases its level, obtains new skills, or unlocks new quests or worlds to discover.

Nowadays there are many types of RPGs that exist.

- Action RPG
- MMORPG
- Roguelikes
- Tactical RPG
- Sandbox RPG
- FPPB RPG (First-person party-based)
- JRPG (Japanese)
- Monster Tamer

So far many RPG contains an open world known as `Overworld` which means an area within the game where all the levels or locations are being connected.

While I was doing my research about role-playing games I did come across a book which was written by Kevin Oxland saying this;

“Role-playing games are unmistakable. They have evolved from the MUDs (Multi User Dungeon) and text-based MMORPGs (Massive-Multiplayer On-line Role Playing Games) of yesteryear.”[1]

According to Straits Research role-playing game genre was one of the most popular genres in 2020 comparing 2019 was a year for MOBA and battle royale games.[14]

After this small explanation about the role-playing game and its definition, I would like to continue with my goal for this project.

1.2 GOAL OF THE PROJECT

Games are also test objects which is not overpriced. Using games, people can be tested in many areas until the edge of human imagination. They also allows to observe what is done, how it is done, what can be done, maybe using some mathematical algorithm or a computer code for example. Based on the high revenue in the gaming market which will be explained in 2.1 Market Analysis part of the documentation.

The main goal of this thesis is creating a game for those features below

- Creating an effective, interactive, and educating version of role-playing game by combining different features from the different types of applications.
- Creating a game that can be played without the limit of age that can address to everyone.
- Creating a game that people will be interested in going deeper into the storyline to fulfill their curiosity.
- Educate people while they enjoy their time playing the game.

1.3 MOTIVATION

I’ve listed some points which have motivated me to choose this topic. As the first and main motivation, I can say that Improving myself in different professions and the possibility of marketing my own game in the future is a milestone for my career. The idea of improving a type of game a lot of people like by myself, seeing the result of everything that I do.

Even earning some money by making original games or maybe benefit everyone by making educational games. To achieve my dreams and goals I started learning game development and at the same time, I decided to work on the same topic with my bachelor thesis which is working in a depth with a limited subject area. To be able to demonstrate my abilities to formulate business topics, processing the data, work systematically.

2. TECHNOLOGY AND MARKET RESEARCH

I asked myself a question before my research “why do people play video games? Why do I play video games?”. I first began my research looking for an answer to these two questions.

People are looking for a way to distress their selves from their daily life, work or even deal with their own mental fatigue. The most preferred way is to play games individually or collectively. But why?

People have been always very curious about the reasons for playing games and it has always been a subject for someone’s research. In 1981 Thomas Malone defined them as 3 main reasons “ Fantasy, Challenge, Curiosity” and another experiment supported by group of authors such as Alan Amory, Kevin Naicker, Jackie Vincent, and Claudia Adams using educational software in 1998.

According to researches2018 was a record-breaking year that over 164 million adults were playing video games in the US. 75% of all Americans had at least 1 Gamer in their house with the improvement of professional gaming called E-sport.[13]

Among the reasons for playing games, individuals can be given example as forgetting the events happened, happening or will happen at work in near future and people may need to relax themselves psychologically or it can be simply people are willing to compete with each other while playing games and develop their own abilities with some enjoyment.

With the growth of the game industry day by day and the necessary investments made in this sector, games have found an important place in our lives. As an answer to my questions those are most likely to be answers.

- People prefer to play games because they have the opportunity to compete in game and also have chat during the game with their environment they are in.
- Opposite to the previous one, some people are playing games to get away from their environment and relax.
- Often people play games because they feel a need to change their focus on something else.

Or

- They simply play games for fun and they have nothing better to do.

2.1 MARKET ANALYSIS

One of the main reasons to create games is the high profit with a high possibility of success in the industry. Starting from 2012 according to the Newzoo gaming industry has grown rapidly especially mobile gaming.

Fig.1 2012-2021 Global Games Market [12]

After looking a little deeper in Newzoo I've seen that in Eastern Europe Poland was number 2 with 439million $ while the general number for 2016 was 106.5 billion $. Comparing the revenue numbers from 2012 to 2021 I can clearly see that with the improving technology of smartphones and the quality of the mobile games numbers for other platforms going to be keep decreasing.

2020 was a very hard year for everyone but due to COVID-19 all segments saw an increase during the lockdown measures., but the mobile gaming segment was the bigger one. With 13.3% year on year increase clearly made the mobile gaming industry again the biggest segment in 2020 with 77,2 billion $.

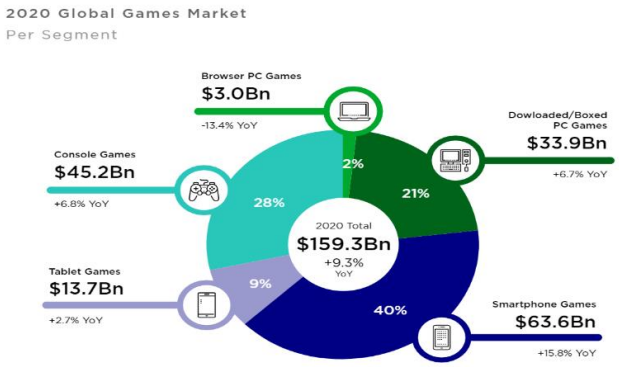

Fig.2 Segments with Year on Year Increase [12]

In 2020 49% of the game market has belonged to the Asia-Pacific region with 78.4 billion $, North America followed them with 25% and 40 billion $ and third-biggest region was of course Europe with 29.6 billion $ and 19% of the market. The reason the Asia-Pacific region to achieve the biggest piece in 2020 was the mobile gaming industry and in 2018 by revenue Role-playing games were the most popular genre in Asia.

After all these high numbers there might be some questions about, Okay I know the growth and revenue for the game market in 2020 but how this number is too great? What makes it so huge?

The answer is total number of players for the regions and how many new players has joined to the gaming world for any kind of segment.

By the end of 2020 all around the world, 2.7 billion people plays video games. With 54% and almost 1.5 billion people Asia-Pacific region is in the first rank but this time second place goes to Europe with 386 million players according to researches made by Newzoo.

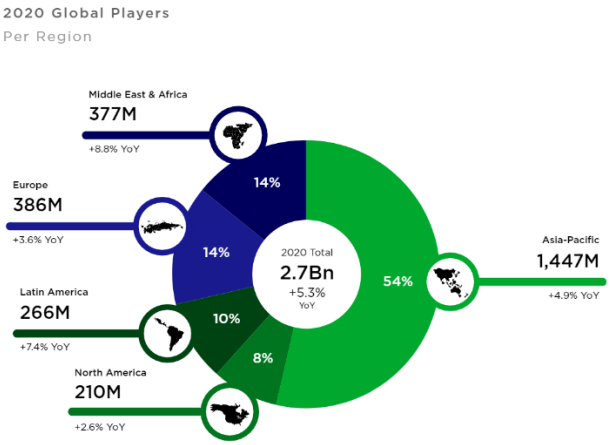

Fig.3 Number Of Players For Regions [12]

2.2 UNITY

Unity is used to create games or other interactive applications (simulation, digital design, etc.), it is an engine programmed by Unity Technologies. There are two versions of Unity which are free version and Pro version. In the free version you can earn money from the games you want and you don’t need to give commission to Unity Technologies. The Pro version has special tools for making higher quality games and is chargeable.

Unity was introduced to the market in 2005, ever since Unity has tried to help everyone creating anything they would like to including computer games, for everyone regardless of technical and/or economical limitations. Soon after the release, in 2006 Unity was chosen for an award at Apple's Worldwide Developer's Conference for being first fully powered game engine. Unity stood largely uncontested on that platform for a couple of years and quickly became a well-known game developing tool among developers.

According to some articles published in 2016 Unity has reported 5.5 million registered users.

Serial number	Feature	Unreal Engine 4	Unity 3D	Notes
1	Programming Language	C++	C#, Java Script	Depends upon the Developer
2	Prototypes	Fast, no line of code,	No line of code but purchase license (extension)	Same like above
3	Assets store	Good	More better, more choices	NA
4	Price	Free but you pay 5% SHARE if revenue from games >$3000 per Quarter	Free until revenue $100000 per year, but not complete version (complicated)	Unreal Engine
5	Cross Platforms	Supports ~6 platforms (Mainly PC+Consoles). Windows PC, Mac OS X, iOS, Android, VR, Linux, SteamOS, HTML5, Xbox One, and PS4.	Supports 21+ platforms (PC, Console, Mobile, Web) Windows PC, Mac OS X, Linux, Web Player, WebGL, VR(including Hololens), SteamOS, iOS, Android, Windows Phone 8, Tizen, Android TV and Samsung SMART TV, as well as Xbox One & 360, PS4, Playstation Vita, and Wii U	Unity
6	Documentation (Data on internet)	Normal documented	Well documented	Unity
7	Graphics	More power full than Unity	Good	Unreal Engine
8		13% of game developers use Unreal Engine 4	47% of game developers use Unity	Unity

[1]Table 1: Comparison between unity and unreal engine(external source)

2.3 BASICS OF UNITY

When you open Unity Hub; You will see projects which were created by user. Then double click that folder. After that interface of Unity will appear on the window.

You can create any folder or game as many possible as you can in here.

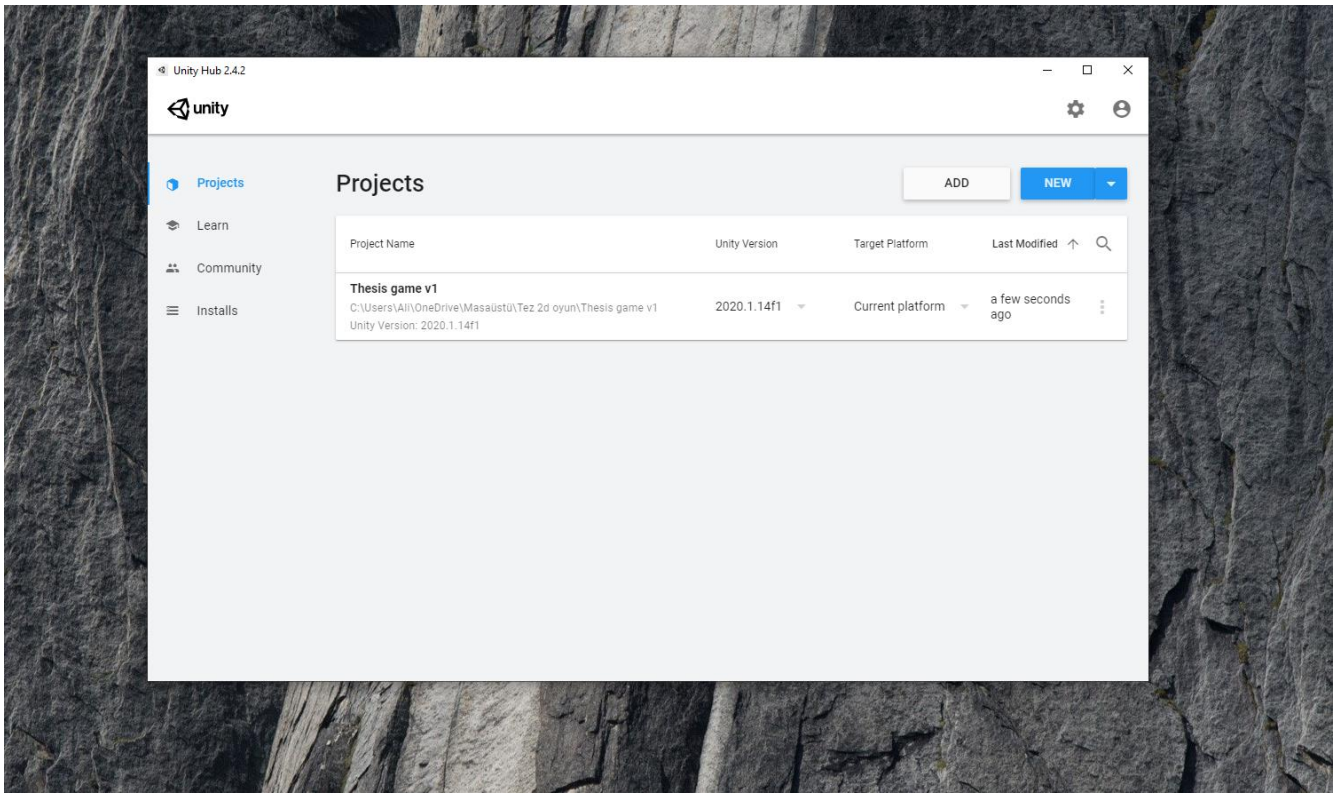

Fig.4 Creating a folder from the Unity Hub(Own source)

The Unity interface consists of 5 panels:

1- Scene Panel: It provides the 2D and 3D vision for the playground. (Scene)

2- Hierarchy Panel: It is a panel that all objects are listed in scene. Objects are called "Game Object" in Unity.

3- Inspector Panel: Detailed information about the selected object are found here.

4- Project Panel: In-game all source that you can use are found here. For example 3D,2D objects photo, sounds folder, fonts etc.

5- Toolbar: Various buttons with your project interaction

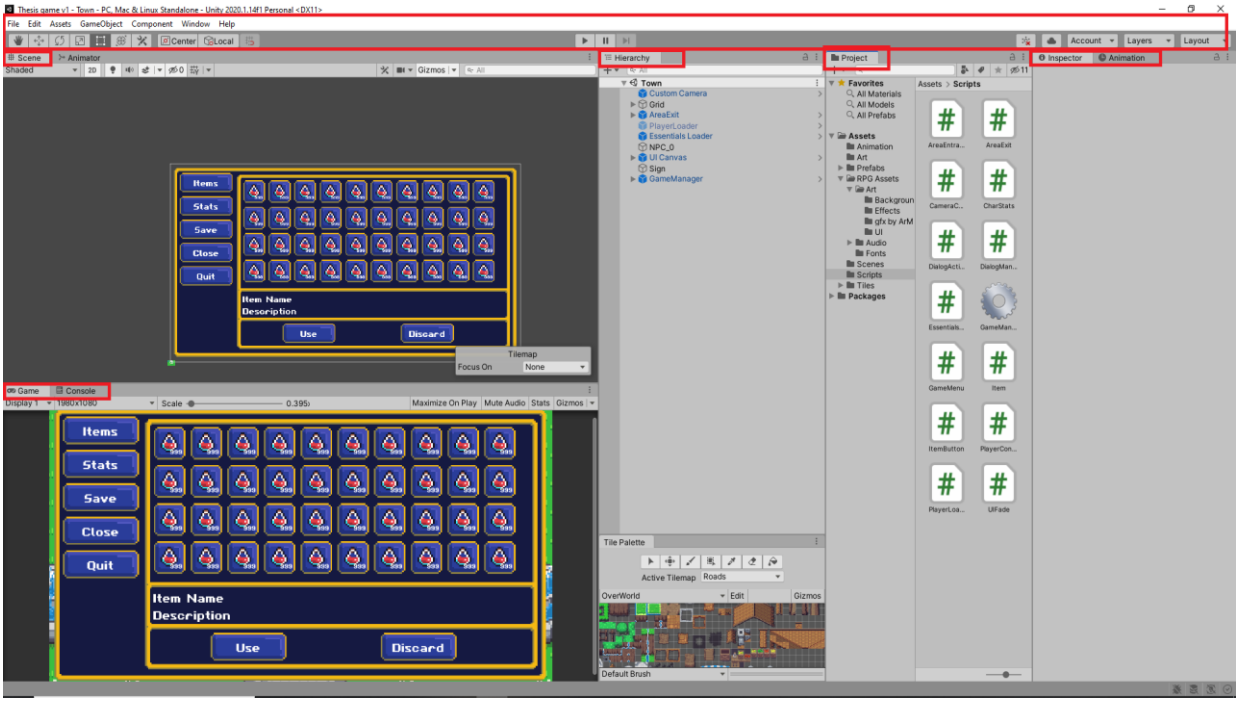

Fig.5 Toolbar, Hierarchy Window, Project Window, Scene View, Inspector Window(Own source)

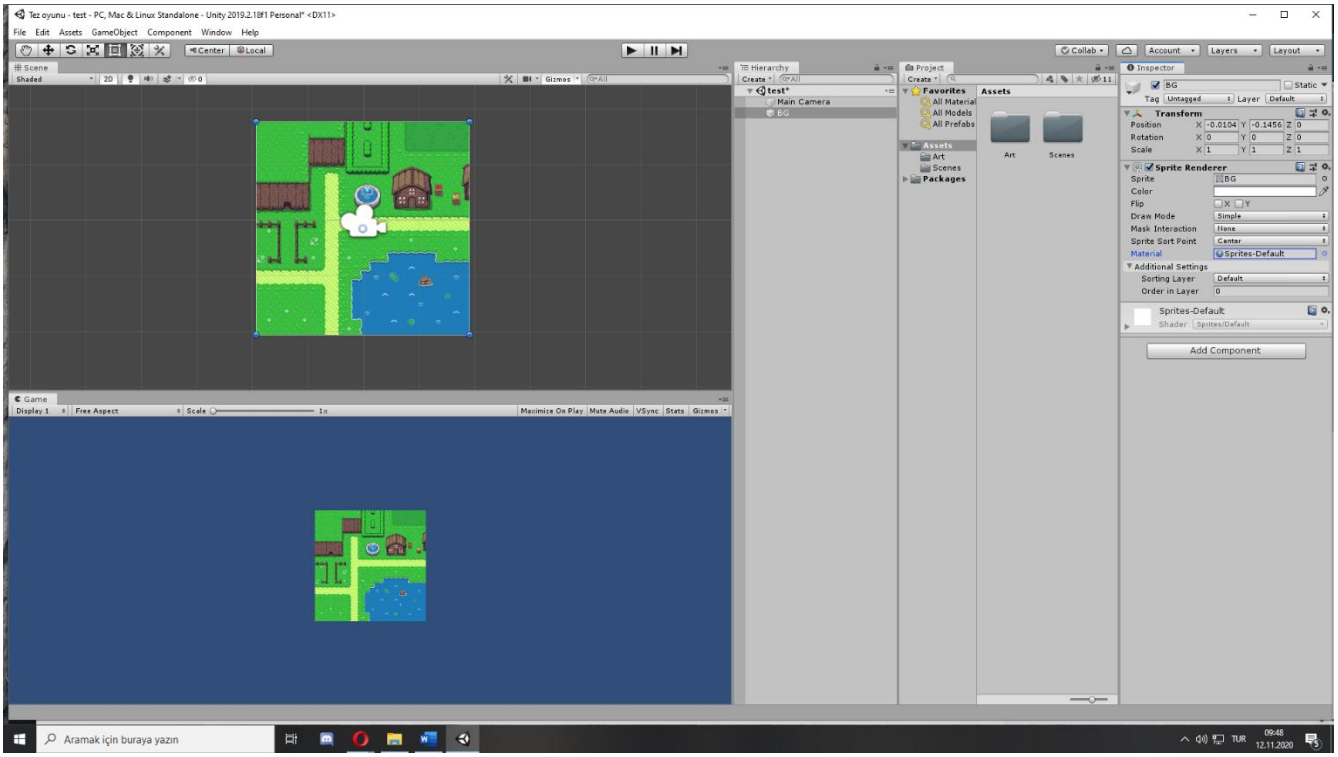

Fig.6 Unity stage design[17] [18]

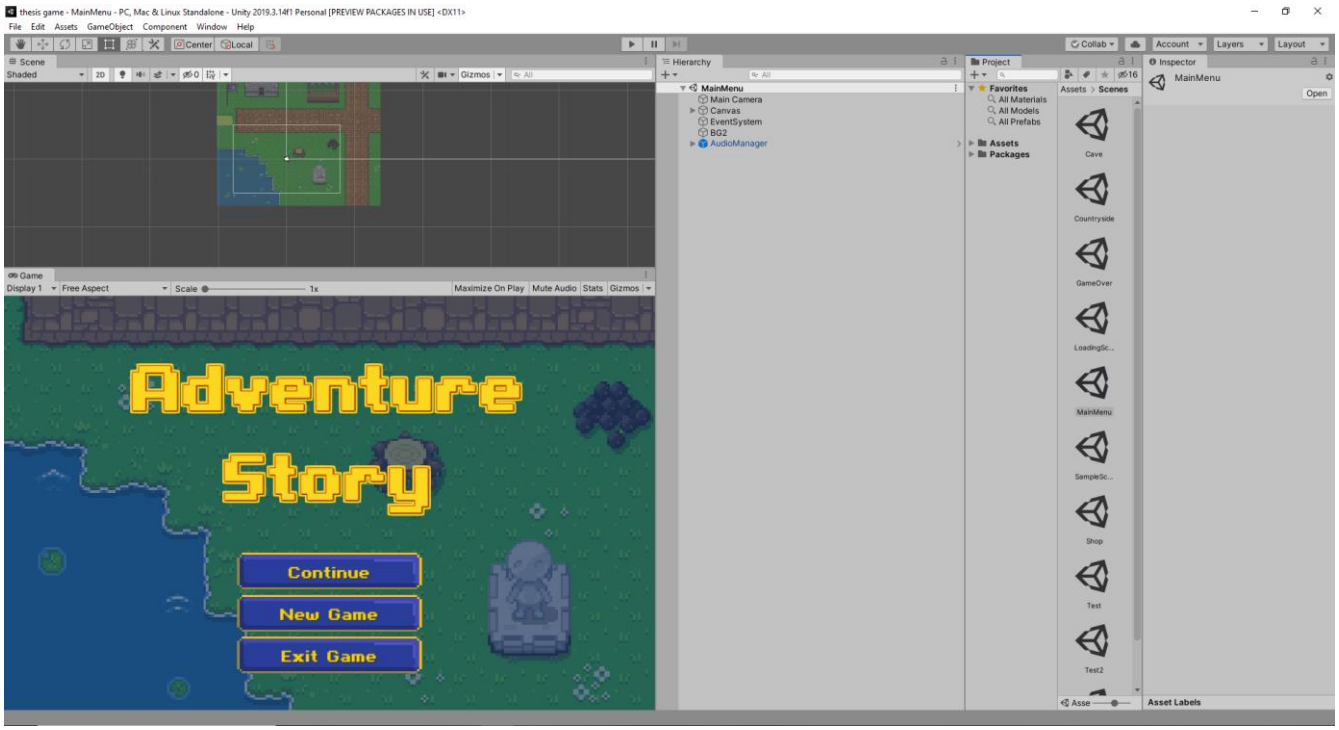

Fig.7 Main screen of the game[17] [18]

2.4 VISUAL STUDIO CODE

When started working on my game I got some problems caused by Visual Studio. Since Unity supports Visual Studio Code so I've decided to move forward using the Visual Studio Code because it's simpler, easier and work just fine for me while I was developing my game.

Visual Studio Code is a powerful and easy to use source code editor. It comes with built-in support for JavaScript, TypeScript, and Node.js and has a huge variety of extensions for other languages (such as C++, C#, Java, Python) and runtimes (such as .NET and Unity).[2]

VS Code is an editor, first and most importantly, it takes very small credit for the work. Unlike traditional IDEs that tend to include everything, in VS code you can modify your installation to the development technologies you are interested in.

VS Code does release a new version every month with new features and bug fixes. Most platforms support auto-updating and you will be directed to install the new release when it becomes available. Another way to update visual studio code is to check updates manually by running "Help > Check for Updates on Linux and Windows" or running "Code > Check for Updates on macOS".

VS Code is a free code editor that can run in the operating systems listed above.

3. GAME DESIGN DOCUMENT

A game design document (also known as GDD) is a highly detailed video game design document for a living software design. The production team generates and edits a GDD and it is used mainly in the video game industry to coordinate efforts within a development team. As a result of cooperation between its designers, artists, and programmers, the development team creates the document as a guiding vision that is used in the game development process. The document must be produced by the development team whenever the game is submitted by the game publisher to the development team, and it is always attached to the publisher-developer agreement; during the game development process, the developer must conform to the GDD. [3]

There are some structures I will be using in my game which are:

- Story
- Character
- Level / Environment design
- Stats and Menu
- Gameplay
- Controls
- Inventory
- Items
- Sound and Music
- Battle.

3.1 STORY

The narrative of a video game is important because it makes the player feel more interested and more immersed in the game. It gives sense to everything in the game and allows the player to realize what they need to do. The benefit of video games is that the player is part of the plot, unlike other forms of storytelling. The player experiences the whole journey and shares the main character's failures and successes.

In video games, storylines are generally told in two ways: through videos and through animations. The narratives are assisted by the interaction and the player's actions. In video games, it is all these things that affect the plot.

In this type of video game in my case its RPG, the story is the main focal point. Players have to solve mysteries and progress through the story. For example in The Legend of Zelda or Fallout integration of the narration to the gameplay was implemented and has been expressed brilliantly.

In my case story is simple.

An ancient dragon is attacking to our village killing their sheep, burning their houses, and so on. Simply torturing those villagers. After some time passing a very strong magician comes and manages to lock this dragon into a dungeon and dies after but leaves a message to villagers.

“ The one who is brave enough to challenge the dragon will have my consent to go in the dungeon. “

After some time our character wakes up with a strange feeling in his/her chest and a message in his hand which directs our hero to our village and our adventure begins.[4]

3.2 CHARACTER

Most of the time, one or more characters automatically come to mind when someone mentions a movie, a series, or even a game. Why? Because we connect the characters with ourselves.

Whenever we watch a film or read a novel, we know instinctively which character we are going to be or which character we feel most relevant or connected to. We even respect and/or condemn other characters and even have a relationship of love/hate often. We value strong character growth above all else.

If we're talking about video games, movies, comic books, anime, who comes to mind when you think about Dragonball? Goku or Vegeta possibly.

It’s the same with games. One of the most important value for players are characterization and character development in games. Because most players will try to find a connection between their character and themselves as they do in movies or comic books. In fact, more often than not, we latch onto one character and use them the most when playing.[5]

Usually, in RPG there are many races such as Elf, Dwarf, Gnome, Dragons, Aliens, and Humans of course.

In my game, our character is from the Human race.

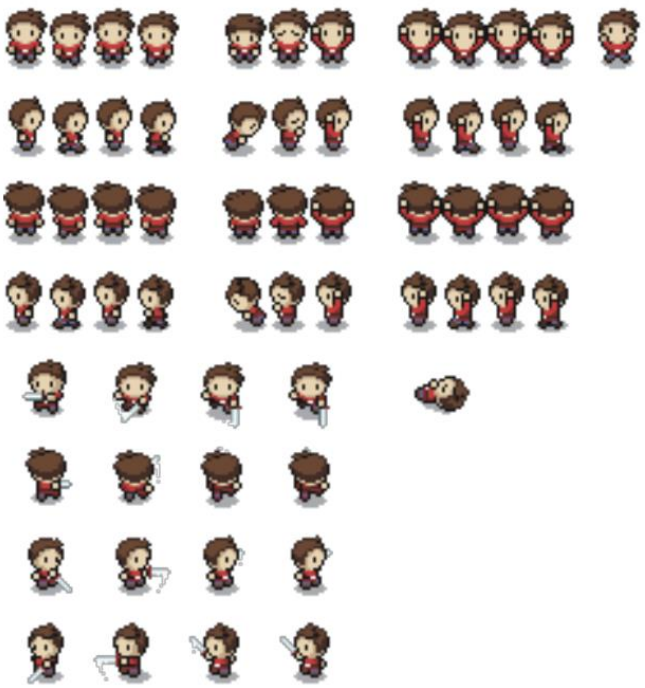

Fig.8 Character sheet[17] [18]

In Fig.5 I've shown the character sheet that I used in my character design.

In order to use this sheet first, I needed to go to sprite editor in unity.

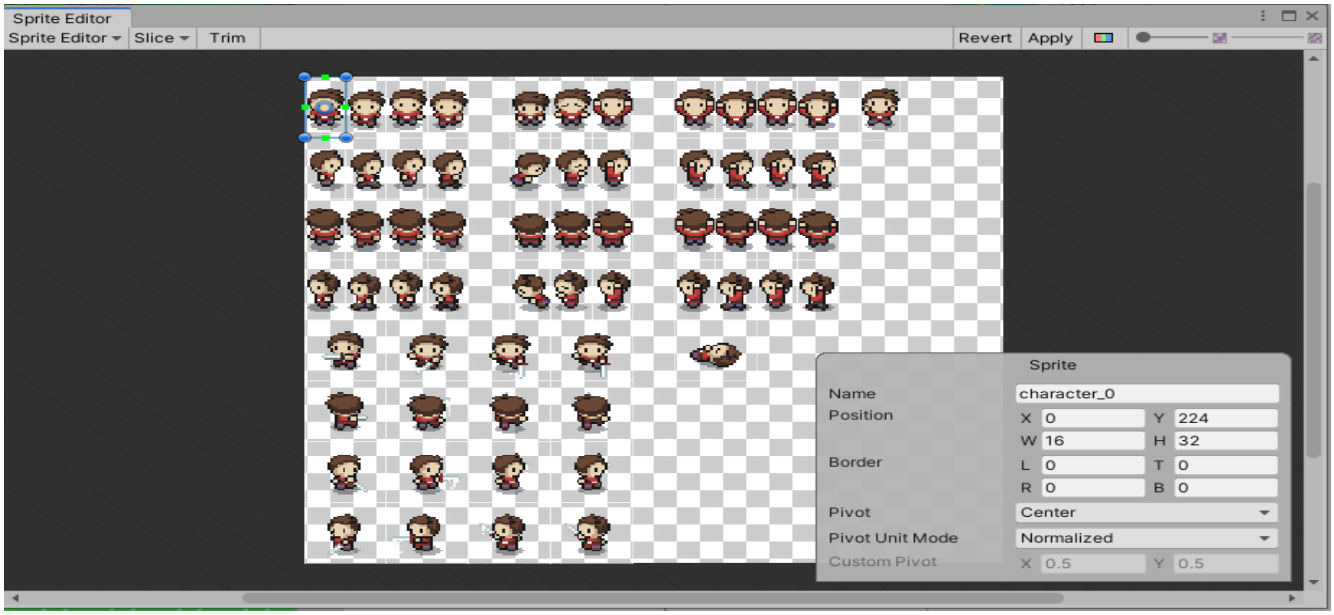

Fig.9 Sprite editor[17] [18]

So before slicing the sheet I had to manage my pixel size and I decided to go for width 16 and height 32.

Of course in order to do that I need to choose "grid by cell size" in the slice menu.

The result is shown in the picture above(Fig.9).

3.3 LEVEL / ENVIRONMENT DESIGN

Creating a world for players to explore in another aspect of game development that can prove daunting. Despite all the games we have played, it can be hard to actually break down what makes a good level or environment.

Two terms "level design" and "environment design" can be considered identical by developers.

Level design or environment design isn't something that can be easily defined in a sentence - it's a far too varied discipline to confine to one area. However, generally, it is the process of creating a game environment in which the game player interacts with the game universe.

In the early 1980s, the level design would have been the process of placing the obstacles, tiles, power-ups, and enemies in for example Mario or Sonic. In the 1990s, it improved with the birth

of 3D gaming, where level design involved creating (essentially) ‘labyrinths’ for the player to explore.[6]

For my project, I used 3 main environments and 1 level which will be shown below.

The first one is the dungeon entrance scene.

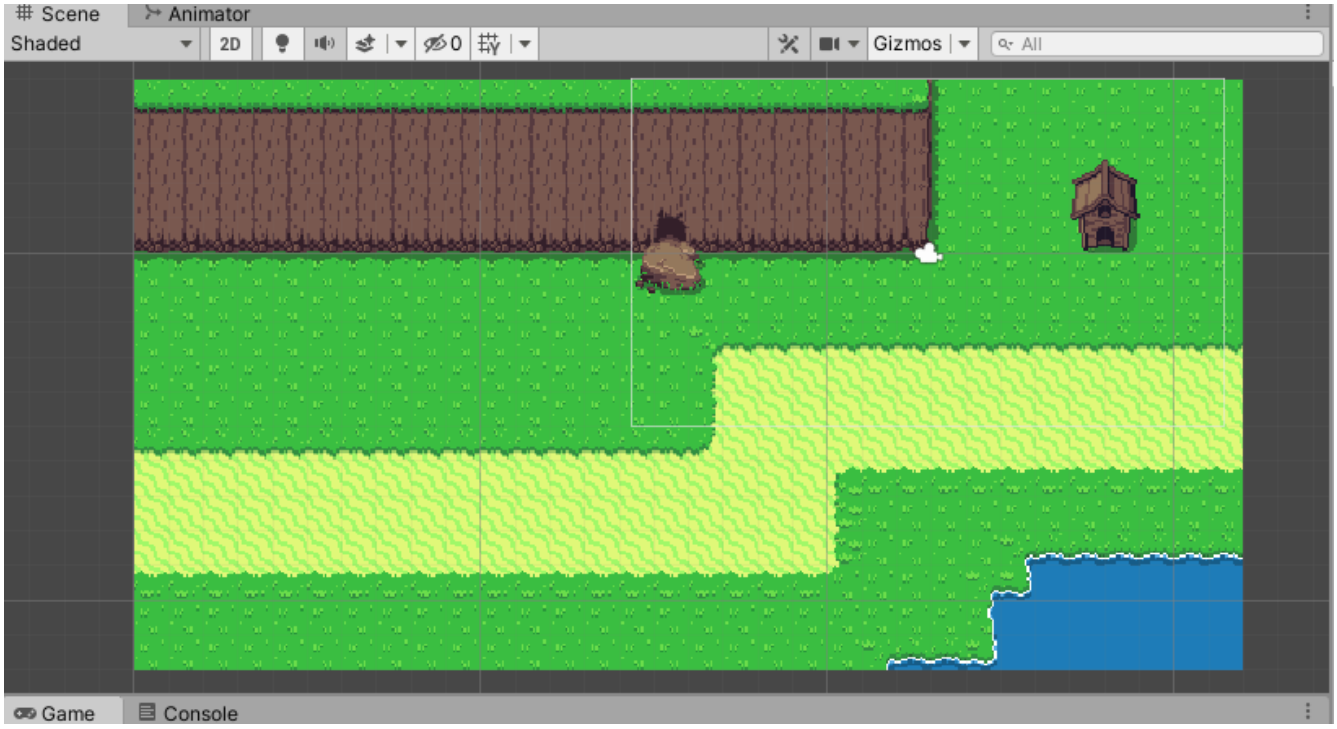

Fig.10 Dungeon Entrance[17] [18]

This is scene shows the town and NPC.

Fig.11 Town[17] [18]

In this picture, you will see the dungeon.

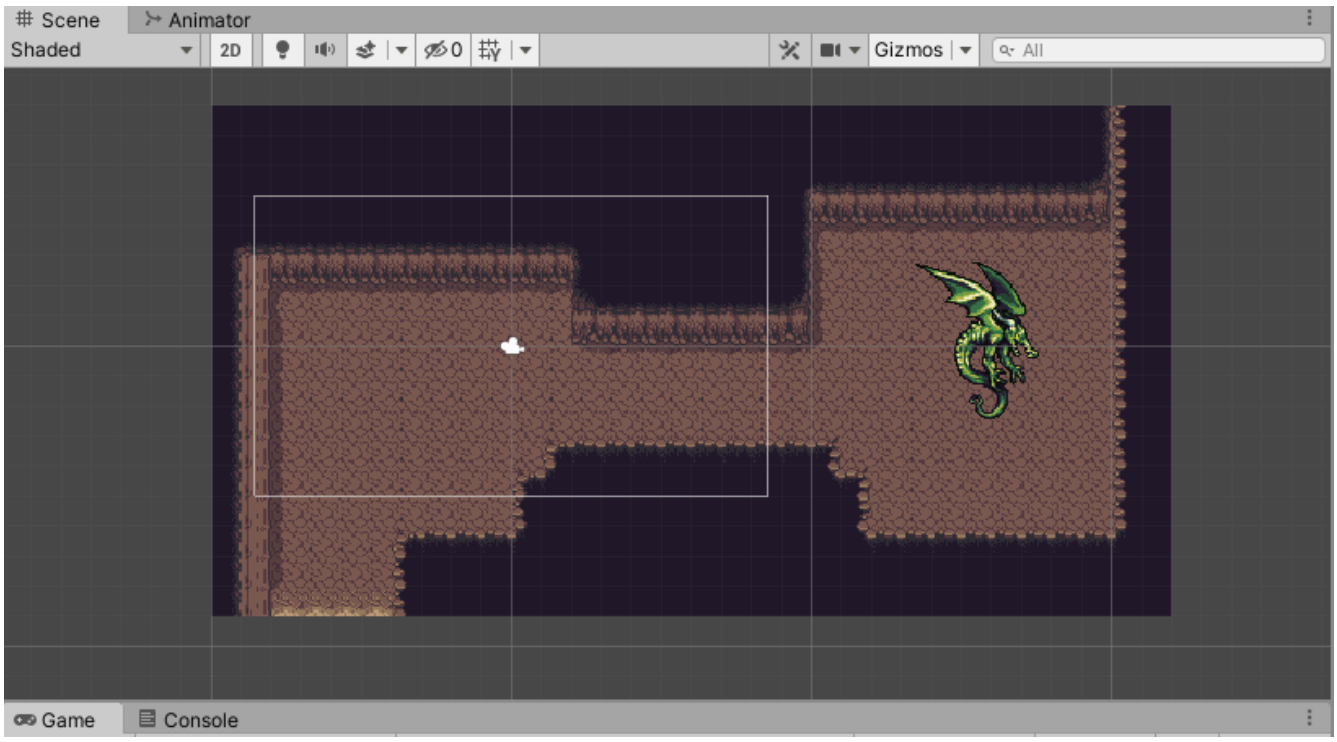

Fig.12 Dungeon and dragon[17] [18]

3.4 STATS AND MENU

What is stat?

Stats are simply a data that presents a particular aspect of a fictional character. They are simply defining how strong an item, character, enemy, the dungeon is. In the role play gaming world everything is up to numbers, even some players believe that in-game with high-enough numbers they can be undefeatable.

In most MMORPG games systems are level and experience-based. The higher level you achieve stronger skills you can use or stronger items you can equip.

I did try to implement this system into the game. So in order to keep a record of player's stats I created a script that is called CharStats.

```
3 references
public class CharStats : MonoBehaviour
{
    1 reference
    public string charName;
    10 references
    public int playerlevel = 1;
    6 references
    public int currentEXP;

    9 references
    public int[] expToNextLevel;
    3 references
    public int maxLevel = 100;

    1 reference
    public int baseEXP = 1000;

    2 references
    public int currentHP;
    4 references
    public int maxHP = 100;
    2 references
    public int currentMP;
    5 references
    public int maxMP = 30;
    1 reference
    public int[] mpLvlBonus;
    1 reference
    public int strenght;
    1 reference
    public int defence;
    0 references
    public int wpnPwr; // weapon power
    0 references
    public int armrPwr; // armor power
```

Listing.1 CharStats[17] [18]

Using those variables I created a levelling and experience system.

```
1 reference
public void AddExp(int expToAdd)
{
    currentEXP += expToAdd;

    if(playerlevel < maxLevel)
    {
            if(currentEXP > expToNextLevel[playerlevel])
            {
                currentEXP -= expToNextLevel[playerlevel];

                playerlevel++;

                //determine wheter to add to  str or def based on odd or even

                if(playerlevel%2 == 0)
                {
                    strenght++;
                    maxMP++;
                }
                else
                {
                    defence++;
                }

                maxHP = Mathf.FloorToInt(maxHP * 1.05f);
                currentHP = maxHP;

                maxMP = maxMP + mpLvlBonus[playerlevel];
                currentMP = maxMP;
            }

    }

        if(playerlevel >= maxLevel)
            {
                currentEXP = 0;
            }

}
```

Listing.2 Levelling system[17] [18]

3.5 GAMEPLAY

I would like to start with the definition of gameplay. This is the specific way in which players interact with a game and in particular with video games. The gameplay is the pattern defined through the game rules, a connection between the player and the game, challenges and overcoming them, plot, and player's connection with it.

In order to keep it interesting, I added some effects and features to game such as fading between scenes or attack effects switching scenes, and many more.

Since I started with fading I would like to explain how fading works in the game and others.

To be able to use fading I needed to learn what canvas is in Unity. For a quick definition of the canvas, I can say that canvas is the area that all UI elements should be. The canvas is a Game Object with canvas component on it and all UI elements must be children of such Canvas.

So, to create such fade effect I simply right-click on the hierarchy choose an image in UI section and canvas will be created automatically in the hierarchy screen but the image I created will be white in order to get black fading effect I need to change the color of the image to black.

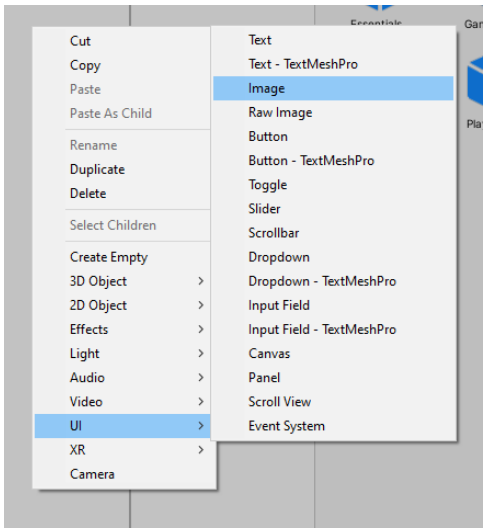

Fig.13 Creating UI Canvas[17] [18]

After creating all necessary elements I need to create new script that will help me control the fading effect which is called UIFade in the game.

In order to call fade I need to attach this script to the canvas to be able to control alpha value in picture using canvas to make this fading effect. While doing that I need to pay attention to one little important thing. Since I am using UI element Unity has a specific section for UI elements so we need to say: **using UnityEngine.UI;**

After that, I can simply create the script as can be seen below.(Listing.3)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

11 references
public class UIFade : MonoBehaviour {

    10 references
    public static UIFade instance;

    12 references
    public Image fadeScreen;
    2 references
    public float fadeSpeed;

    4 references
    public bool shouldFadeToBlack;
    4 references
    public bool shouldFadeFromBlack;

    // Use this for initialization
    0 references
    void Start () {
        instance = this;

        DontDestroyOnLoad(gameObject);

    }

void Update () {

    if (shouldFadeToBlack)
    {
        fadeScreen.color = new Color(fadeScreen.color.r, fadeScreen.color.g, fadeScreen.color.b, Mathf.MoveTowards(fadeScreen.color.a, 1f, fadeSpeed * Time.deltaTime));

        if(fadeScreen.color.a == 1f)
        {
            shouldFadeToBlack = false;
        }
    }

    if (shouldFadeFromBlack)
    {
        fadeScreen.color = new Color(fadeScreen.color.r, fadeScreen.color.g, fadeScreen.color.b, Mathf.MoveTowards(fadeScreen.color.a, 0f, fadeSpeed * Time.deltaTime));

        if (fadeScreen.color.a == 0f)
        {
            shouldFadeFromBlack = false;
        }
    }
}
```

Listing.3 UIFade script[17] [18]

For now, I would like to jump into switching scenes so you can keep it up with me when I explain what I did following in fading between scenes.

In order to use fading effect I should be able to switch from one scene to another scene and to that I created a script called AreaExit. But to use this script I need to have a GameObject that we can add BoxCollider 2D so I can warp our character from the current scene to the one that is connected.

What does my AreaExit script do?

My AreaExit script helps to exit from one and load to another scene using this bit of code shown below in Fig.14

Fig.14 Area To Load[17] [18]

With that, I can simply use this scrip in every scene by making it a prefab.[11]

Since we need to have an object to help us leave the current scene I would like to have another one to help me enter the new scene. Which is created as AreaEntrance in this case. Because If I don't use this our character will simply respawn in the same location that left the previous scene.

So for example, if I left the scene from the right side of the map, this location is going to be our left on the next map in order to proceed in the game. Right at this point the area entrance is playing its role and make our character spawn in the correct position.

To be able to do that I need to name those exits and entrances on the map so my script will know where to go and where the character is coming from in order to pick the correct warp gate that I will show the script that I used for AreaEntrace below.(Listing.4)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;


public class AreaEntrance : MonoBehaviour {


    public string transitionName;

    // Use this for initialization

    void Start () {
        if(transitionName == PlayerController.instance.areaTransitionName)
        {
            PlayerController.instance.transform.position = transform.position;
        }

        UIFade.instance.FadeFromBlack();
        GameManager.instance.fadingBetweenAreas = false;
    }

    // Update is called once per frame

    void Update () {

    }
}
```

Listing.4 Area Entrance Script[17] [18]

So to able to perfect our fading effect I need to have time defined for it because I don't want it to happen so fast as it is called it needs to fade from black to white and white to black. To do that I added some code that can control the time of effect(Listing.5).

```
2 references
public float waitToLoad = 1f;
3 references
private bool shouldLoadAfterFade;
```

```
void Update () {
    if(shouldLoadAfterFade)
    {
        waitToLoad -= Time.deltaTime;
        if(waitToLoad <= 0)
        {
            shouldLoadAfterFade = false;
            SceneManager.LoadScene(areaToLoad);
        }
    }
}
```

Listing.5 Fade timer[17] [18]

And after all, those setting all I have to add UIFade.instance.FadeFromBlack(); to AreaEntrance as seen in Listing.4.

Since we covered fade effects and switching scenes, I would like to continue with character animations such as facing the correct direction as I move the character.

To animate the character, I need to choose animation in the following picture(Fig.15).

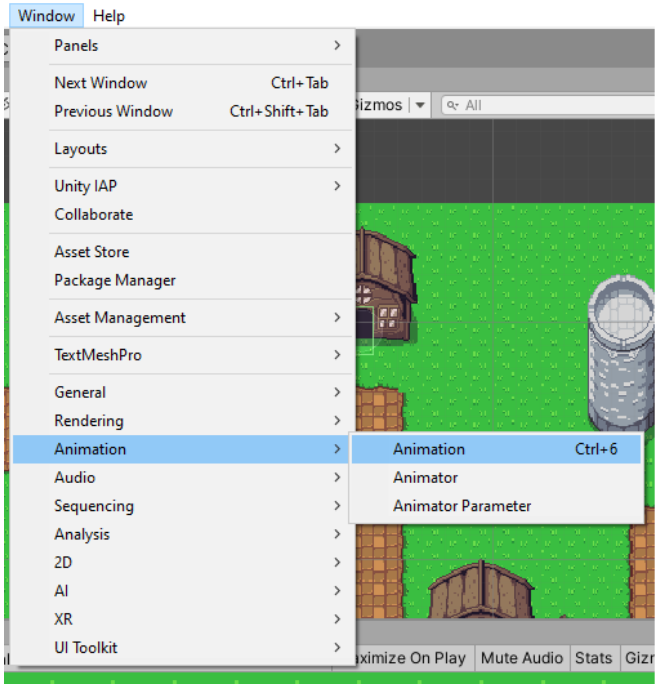

Fig.15 Animation window[17] [18]

After opening the animation window I simply dock it to our unity and next step is creating a new animation in a new folder called animation in-game files.

After creating new animation I am going to use our character sprite that was chopped into smaller pieces when creating our character to give our character moving animation and drag-drop those images according to their movement style in our case I have 4 pictures to make movement animation 1 leg forward - both legs – other leg forward – both legs. But to make movement more smooth I can simply duplicate the last one so between 2 movements animator will be passing smoother than before.

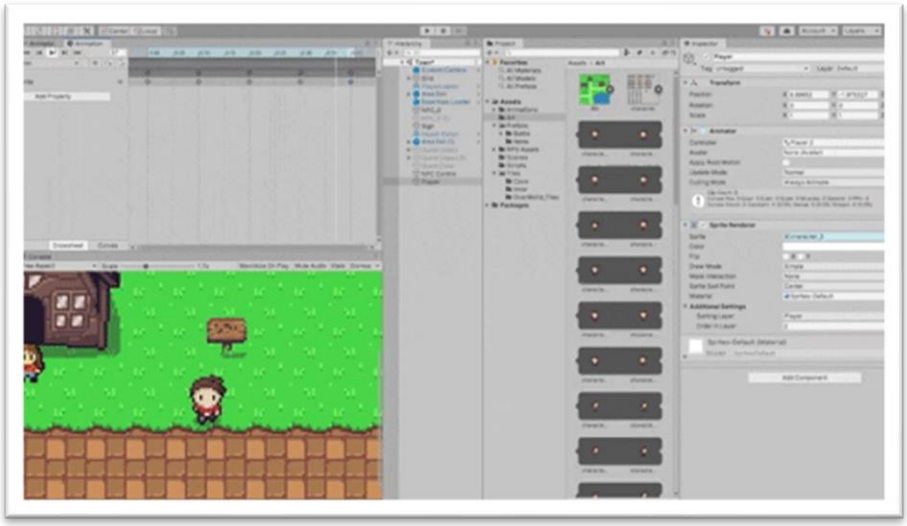

Fig.16 Character animation screen[17] [18]

https://media.giphy.com/media/Z9j4EVqihoAHhlJv0M/giphy.gif

Animated version of character movement in the link above.

After creating animations for all 4 directions I need to simply make a movement for our character for given x and y values in animator. There is an easy way to create this in an animator using a blend tree option.

After creating the blend tree I need to set parameters for our player input for x and y. Clicking to Blend tree we see the parameters section in our inspector by clicking to the plus button I will have two options “Add motion field “ and “New blend tree “ since I have our blend tree already I just need a motion parameters in our tree.

Fig.17 Adding new motion[17] [18]

Clicking the button inside red square can be seen in Fig.17 we can add new motion that we created earlier as animation.

After setting all motions our blend tree will be looking just like in Fig.18

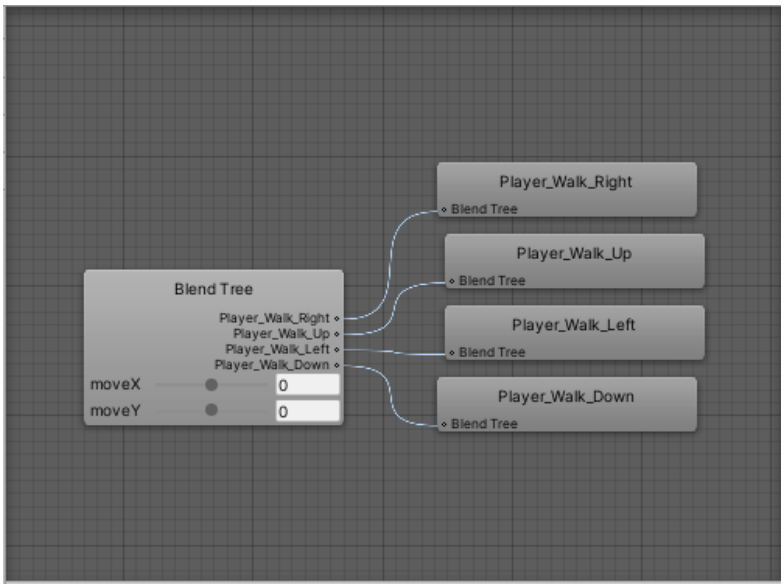

Fig.18 Blend tree[17] [18]

After adding the second motion to our blend tree I will have a directional view generated automatically in Unity and moving this red dot around I can see where our character is looking towards can be easily seen in there.

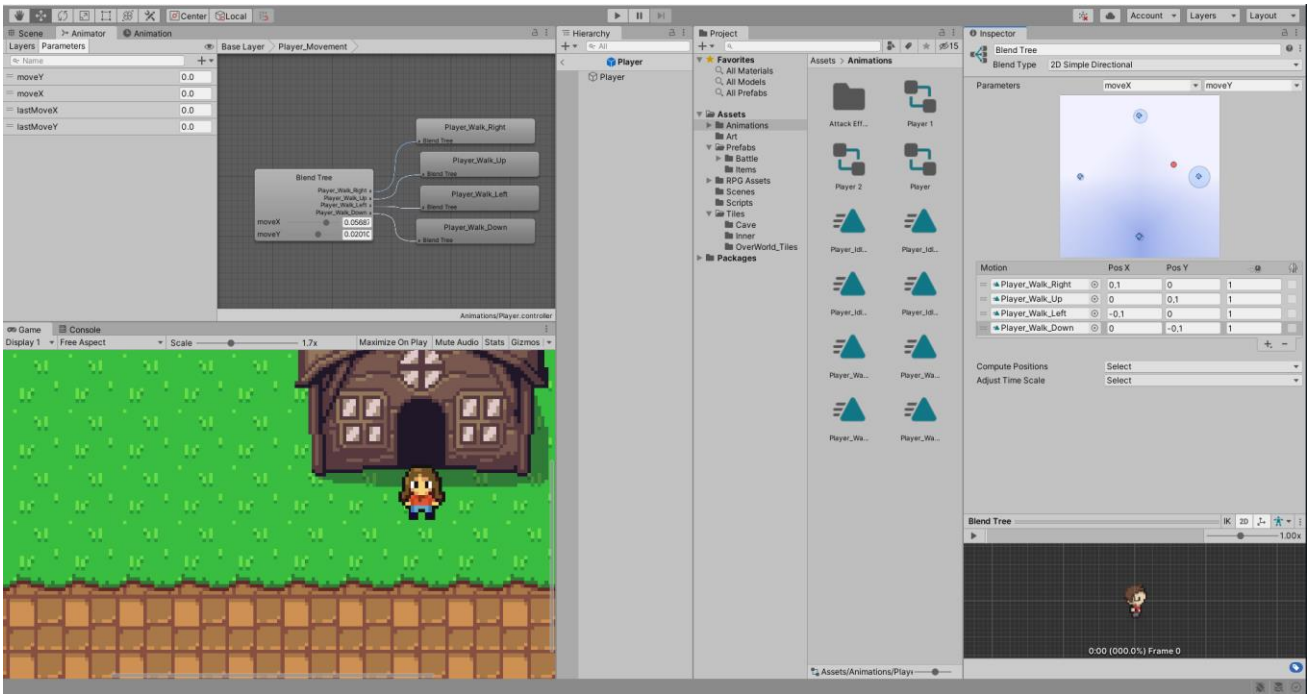

Fig.19 Complete version of blend tree[17] [18]

3.6 CONTROLS

After creating our character I gave circle collider 2D and Rigidbody 2D in unity in order to give my character physics.

Usually, the Unity Editor's Transform component defines how a GameObject (and its child GameObjects) is positioned, rotated and scaled within the Scene. When it is changed, it updates other components, which may update things like where they render or where colliders are positioned. The 2D physics engine is able to move colliders and make them interact with each other, so a method is required for the physics engine to communicate this movement of colliders back to the Transform components. This movement and connection with colliders is what a Rigidbody 2D component is for.[9]

When add circle collider 2D and Rigidbody 2D and I hit the play button my character was falling down from the screen, to solve this issue I removed the gravity scale by setting it to " 0 " in Rigidbody 2D.

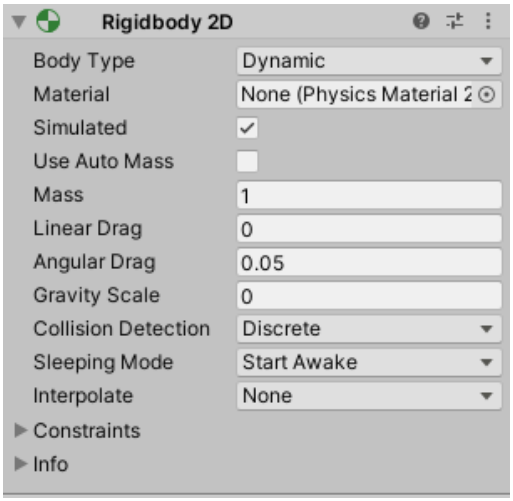

Fig.20 Rigidbody[17] [18]

After those steps in order to make my character, I created a script and named it as PlayerController, and defined my rigidbody2D in my script.

In order to move my character around the world, I should access a particular part of the Rigidbody called velocity.

Another good thing about using Unity we have a defined input list in it. So simply when I am creating my scripts I can use those inputs directly from unity without needing to create or define them from the scratch.

After all my script for moving the character will look like below. (Listing.6)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

4 references
public class PlayerController : MonoBehaviour
{
    3 references
    public Rigidbody2D theRB;
    2 references
    public float moveSpeed;
    // Start is called before the first frame update
    0 references
    void Start()
    {

    }

    // Update is called once per frame
    0 references
    void Update()
    {
        theRb.velocity = new Vector2(Input.GetAxisRaw("Horizontal"), Input.GetAxisRaw("Vertical") * moveSpeed)
    }
}
```

Listing.6 Player Controller Script[17] [18]

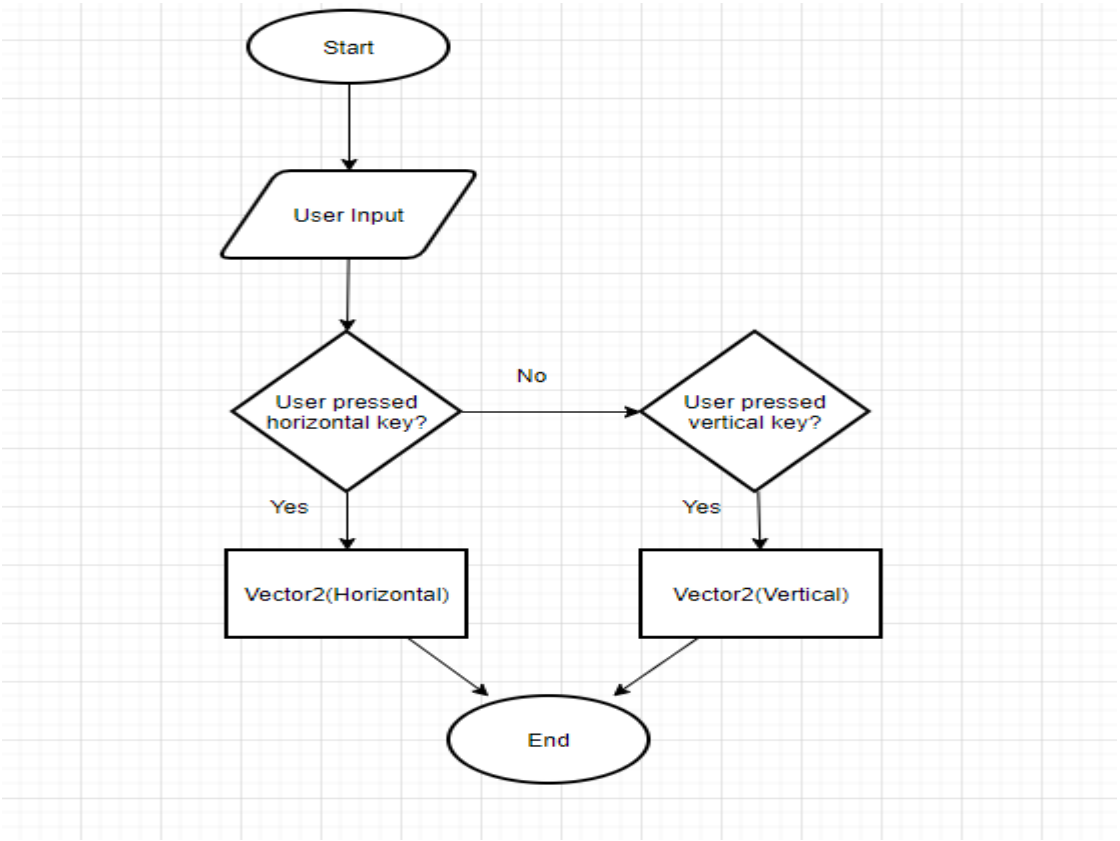

Fig.21 Flow Chart View of Player Movement(own source)

3.7 INVENTORY

In the economy, inventory is a term used to refer to the products that are available for sale and the raw materials used to manufacture goods that are available for sale. But it is slightly different for the game World.

Inventory is a space where game objects, items, consumables such as health potions or mana potions can be stored and carried around the World within the character. Inventory plays a very important role in role-playing games.

After doing some research about inventory systems I found such an image that helps people understand why a different type of inventory systems exist and they need to be used. At their finest, within the scope of your game world, RPG inventory systems are very useful if they are implemented correctly and fitting your World at its best.

At this point of the game “Rule of 99” was the perfect fit for my game so I used this design paradigm. The idea behind this is very simple a player can hold items with only a stash of 99. If you want you can of course increase this number from 99 to 200 or even 1000 by changing the number in the source code.

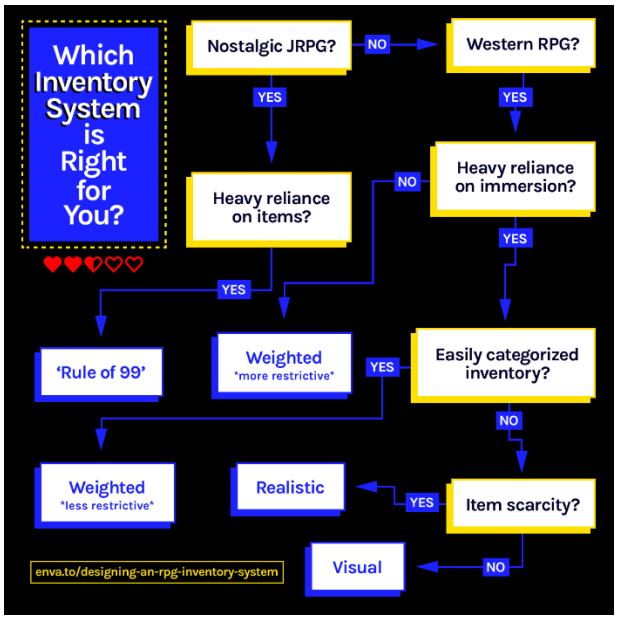

Fig.22 Flow Chart View of Inventory Design [16]

When I implemented the inventory system in the game I used canvas that I created for items. After duplicating this UI object renaming it to item buttons and I needed to make it a child object to the window called item windows. To place items inside the inventory I need to create objects called item buttons that will be child objects for item buttons. To give them a nice look in inventory I used "Grid Layout Group" component that lines the object nicely to have such a view in inventory Picture below.

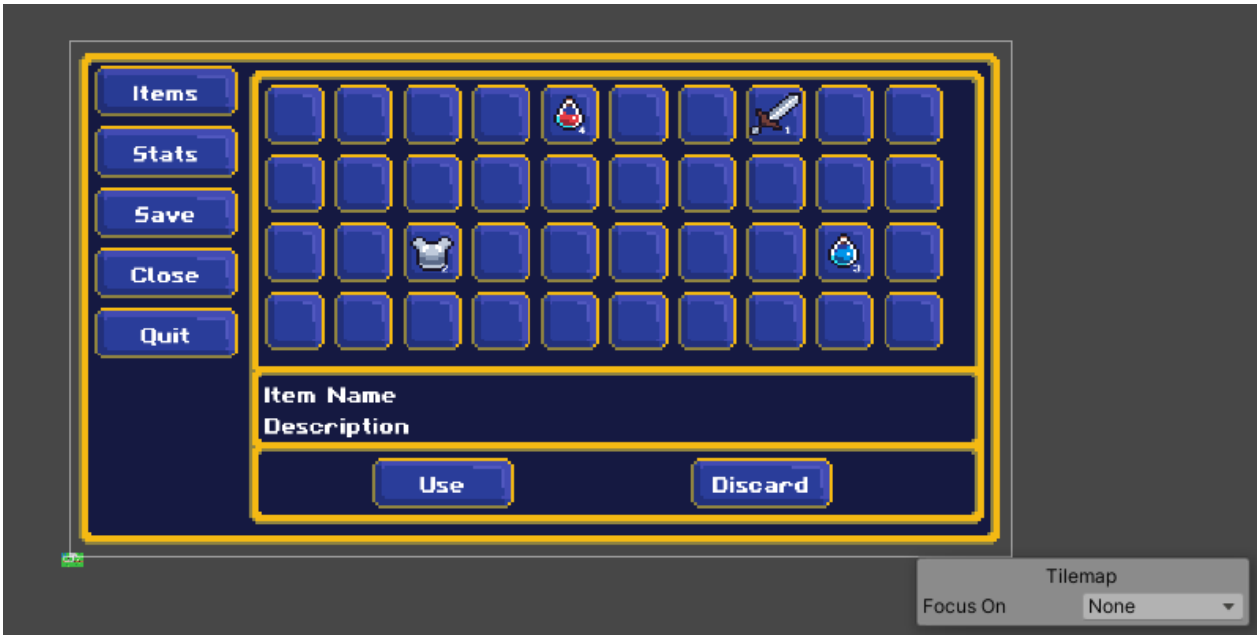

Fig.23 Inventory Design[17] [18]

3.8 ITEMS

For players, items can be very useful objects. Some games contain such items that have curse effects for the enemy player or monsters deals damage over time, some items are just too strong to be defeated or some items may not have any value for the player which are usually called basic items. In role-playing games, items are particularly common, as they are typically required for the completion of quests or to progress through the plot.

Often, such objects can be rare and appear only once at a particular location or after a specific quest has been completed. Other items can appear more often and but those will not provide a big bonus or special effects as explained before. Also in role-playing games, such items with curse effects or which are very strong and can be called `one of its kind`, that can make a fortune in real life in case the owner of this item decides to sell it. Games can vary on how items are used by the player. For example in the Mario and Sonic series, items are automatically used when the player touches them. There are also games, such as Call Of Duty Warzone in this game player has to press a specific button to collect items or else like many role-playing games, items can be collected automatically or by hand pressing the specific button just like explained above, the collected items can be kept inside the inventory and used manually when the player needs it.[7]

As it was said about items I did add some beneficial items to the game that can help to player move forward within the game using those items that I will be shown below with their details. But before that while I was coding the game I did categorize the items as if it “is item” , ”is Weapon” or “ is armor” so I will also use those categorizations here in my documentation.

3.8.1 IS ITEM

In this category, I will explain what are the consumable items. In games, consumable items are 1 use items that will disappear in most cases. In this game, I added 2 items which are listed below.

Health Potion

Mana Potion

3.8.1.1 HEALTH POTION

It is a consumable item that provides 100 health points (100 HP) after the usage of the potion.

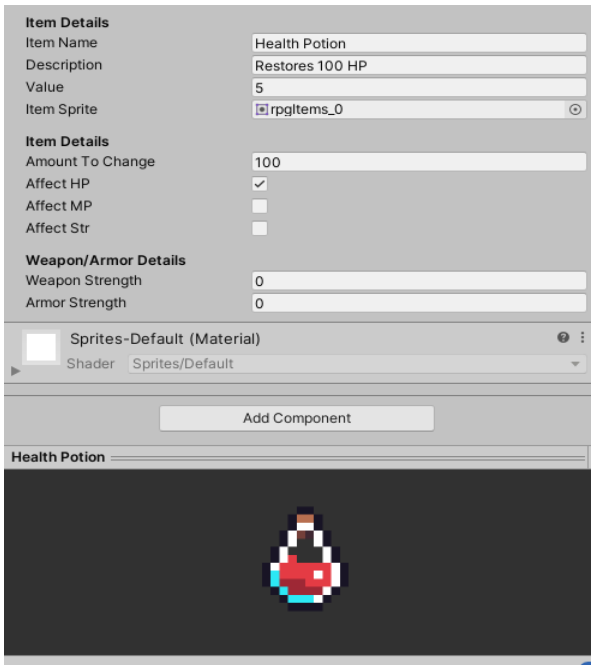

Fig.24 Health Potion[17] [18]

3.8.1.2 MANA POTION

It is a consumable item that provides 50 mana points (100 MP) after the usage of the potion.

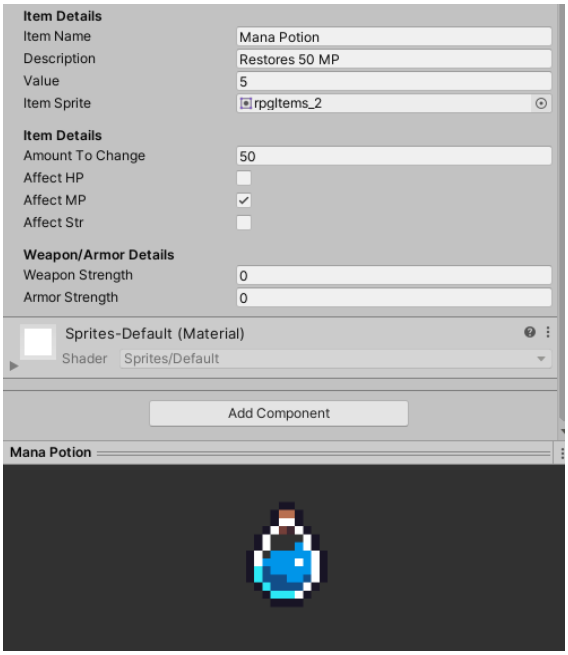

Fig.25 Mana Potion[17] [18]

3.8.2 IS WEAPON

In this category types of weapons in the game will be explained which is a sword in this case. To increase the number of choice in weapon I added 2 different weapons listed below

Iron Sword

Wooden Sword

3.8.2.1 IRON SWORD

It is the strongest weapon in the game that players can use and provides 15 strength to player stats.

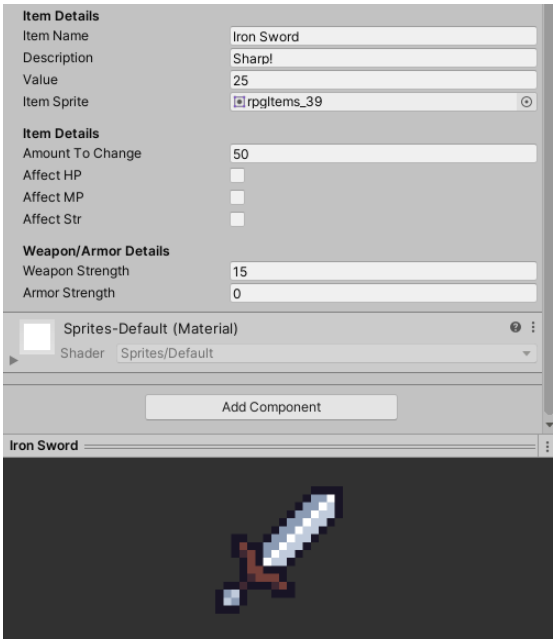

Fig.26 Iron Sword[17] [18]

3.8.2.2 WOODEN SWORD

Basic weapon in the game that player can use and provides only 5 strength to player stats.

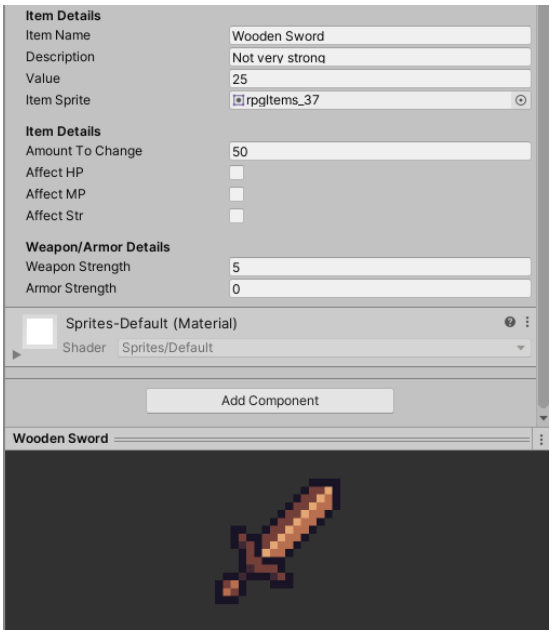

Fig.27 Wooden Sword[17] [18]

3.8.3 IS ARMOR

In this category types of armor will be explained the same as in the weapons section.

Leather Armor

Iron Armor

3.8.3.2 LEATHER ARMOR

Basic armor player can use in the game which provides only 5 armor strength to stats.

Fig.28 Leather Armor[17] [18]

3.8.3.1 IRON ARMOR

Strong armor player can use in the game which provides only 10 armor strength to stats.

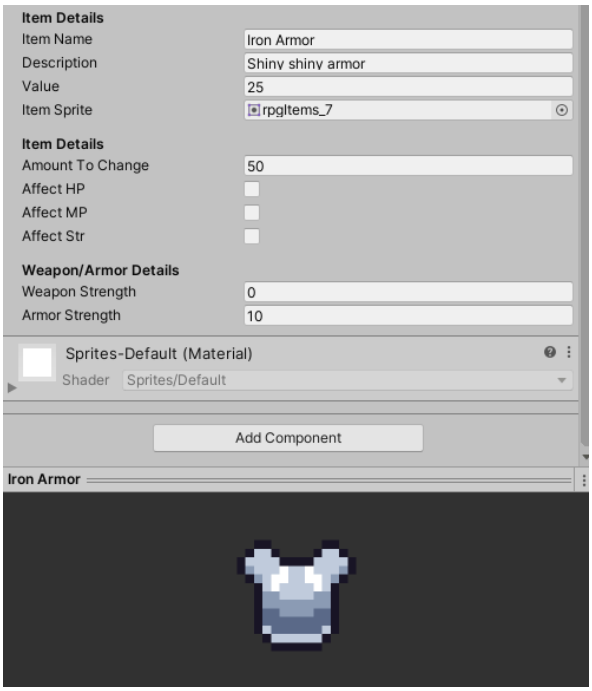

Fig.29 Iron Armor[17] [18]

3.1.9 SOUND AND MUSIC

Nowadays sound effects in video games does make a very big difference since the competitive games are upcoming such as Counter-Strike Global Offensive, Call of Duty League, and many more. Only with a little mistake players can detect enemy player's location only hearing a little sound in those competitions. So after doing some research about the importance of background music in games I have come up with such results which include game developers.

When game developers begin creating games, background music and sound effects are typically important. The ability of music to deliver feelings, a sense of reality, and authentic interactions are crucial for the satisfaction of the player and the game itself. For the overall enjoyment of the game, the storyline, interactions within the game, voiceover actors are all plays a key role. Have you ever wondered, without the iconic theme or outstanding performances of voice actors, what will your favorite game be like?

Whether it's a fast-paced first-person shooter or an educational video, voice actors and background music make an impact, but we are not always paying attention to it. We become immersed in the experience, and we don't notice if something is there when it fits so perfectly into the narration. If a soundtrack is terrible, or a voice actor couldn't perform in a way that can touch the user's soul; your immersion will be gone. Once it is gone you'll be more focused on the setbacks than the story itself, even if the story is fantastic.

A study published by the Journal of Psychology and Psychotherapy says that sound makes an essential difference that was concluded that background music is indispensable in creating a compelling game.[8]

3.1.10 BATTLE SYSTEM

3.1.10.1 WHAT IS TURN-BASED?

Game flow is split into defined parts, called turns, moves, or plays, in turn-based games. Before committing to game action, a player of a turn-based game is given a duration of consideration (often bounded, sometimes unbounded) to ensure a distinction between the game flow and the thought process, which hopefully leads to better choices. This round of play is over after each player has taken his or her turn, and any special mutual processing is completed. This leads our player to the next round of Gameplay.[10]

In order to create a battle system, I needed a couple of things.

First I needed to create a dungeon scene(Fig.12) second I needed a battle scene that will be a layout for a dungeon scene with a battle panel that the player will use it during the battle and a battle characters that I will see what is actually going on in the scene.

When creating the battle scene I simply create a new GameObject simply right-clicking on the hierarchy window and next add a background picture for it from the assets folder after choosing the background image I needed to make it a child object for your battle screen by simply dragging and dropping it on the battle scene.

After creating the battle scene and background I need to create our battle panel for our player. In order to do that I need to create UI Canvas.

After creating scenes for our battle next step is to create a battle script so-called BattleManager. Creating BattleManager needs an instance that I may or use in future steps as a reference in other scripts also to be sure that my DontDestroyOnLoad works fine.

Before start talking about how our battle sequence will work, I need to be sure that every variable I are going to use is defined correctly and there is no active battle scene in the background.

```
public void BattleStart(string[] enemiesToSpawn, bool setCannotFlee)
{
    if(!battleActive)
    {
        cannotFlee = setCannotFlee;

        battleActive = true;

        GameManager.instance.battleActive = true;

        transform.position = new Vector3(Camera.main.transform.position.x, Camera.main.transform.position.y, transform.position.z);
        battleScene.SetActive(true);

        AudioManager.instance.PlayBGM(0);
    }
```

Listing.7 Battle Activation[17] [18]

In order to load players and enemies in the correct spawn positions to do that I used for loop and tell to my script if our players in each position is active? If they are active then I should put them in a correct place with the correct player at its position using the code below (Listing.8)

```
for(int i = 0; i < playerPositions.Length; i++)
{
    if(GameManager.instance.playerStats[i].gameObject.activeInHierarchy)
    {
        for(int j = 0; j < playerPrefabs.Length; j++)
        {
            if(playerPrefabs[j].charName == GameManager.instance.playerStats[i].charName)
            {
                BattleChar newPlayer = Instantiate(playerPrefabs[j], playerPositions[i].position, playerPositions[i].rotation);
                newPlayer.transform.parent = playerPositions[i];
                activeBattlers.Add(newPlayer);


                CharStats thePlayer = GameManager.instance.playerStats[i];
                activeBattlers[i].currentHp = thePlayer.currentHP;
                activeBattlers[i].maxHP = thePlayer.maxHP;
                activeBattlers[i].currentMP = thePlayer.currentMP;
                activeBattlers[i].maxMP = thePlayer.maxMP;
                activeBattlers[i].strength = thePlayer.strength;
                activeBattlers[i].defence = thePlayer.defence;
                activeBattlers[i].wpnPower = thePlayer.wpnPwr;
                activeBattlers[i].armrPower = thePlayer.armrPwr;
            }
        }


    }
}
```

Listing.8 Player Positioning In Battle Scene[17] [18]

3.1.10.2 MAKING TURNS

After getting our battle scene working correctly with our players and enemies appearing in the correct places this is time to start making the turn system.

To do that I need to check the current number of the turns which will help cycle our battlers and time for waiting to end the current turn which can be an input from our player or for the enemy turn to finish.

public int currentTurn;

public bool turnWaiting;

After setting these two I need to check if the battle is active because if the battle is not active I have no reason to check the turns anyway. If the battle is active I simply check if it's the turn for the player move or the enemy move and if it's our turn we should be able to see our battle interface on the screen if not our players turn I don't need those interface buttons so I just hide them as can be seen below. Just to be able to sure it is working I added a little piece of code to check.

```
if(battleActive)
{
    if(turnWaiting)
    {
        if(activeBattlers[currentTurn].isPlayer)
        {
            uiButtonsHolder.SetActive(true);
        } else
        {
            uiButtonsHolder.SetActive(false);

            //enemy should attack
            StartCoroutine(EnemyMoveCo());
        }
    }

    if(Input.GetKeyDown(KeyCode.N))
    {
        NextTurn();
    }
}
```

Listing.9 Testing Turns[17] [18]

While checking if the system works I noticed that the number of turns in inspector was constantly increasing and after some time when the number of active battlers was out of range I was getting an error that ArgumentOutOfRangeException.

To fix this issue I created a new public void so-called NextTurn can be seen below.

```
public void NextTurn()
{
    currentTurn++;
    if(currentTurn >= activeBattlers.Count)
    {
        currentTurn = 0;
    }

    turnWaiting = true;

    UpdateBattle();
    UpdateUIStats();
}
```

Listing.10 Solution for Range Error[17] [18]

4. TESTING

Testing the game or software is critical. To see the results of the product before marketing it developers and owners need a proof that it works as it was supposed to be. For games, testing is very important because games are complex in many ways. Testing games and software are kind of similar most of the features in-game testing is being borrowed from software testing, but of course with some modifications. If I need to compare these two product type games has a big aspect that is called " Fun factor". I mean there is no point in making a game and testing it, if it is not fun to play or does not interact with people to entertain them.

4.1 WHAT IS Debug.Log?

Logs files are keeping information about events that occurs while developing. In those files positive and negative results are kept and as an addition in negative results we can see detailed information about what caused a negative result, what is the error code and it can be accessed by looking at these files.

Using Debug.Log we can print the informational message that will help us to debug our application.[15]

4.2 USER TESTING

So in order to test the game and its user interface(UI) I decided to use the same patterns as I was using in User Interface development laboratory classes. But to do that I need to set some limitations and rules for testing. To complete the testing I asked some of my friends or family to help me and according to their experience give a rating from 1 to 5.

(1 is not satisfied, 5 is totally satisfied)

Rules

1- The player needs to defeat the dragon in order to complete the testing and also the game itself.

2- In order to enter the boss cave player needs to speak with NPC and activate the mission.

3- The player has only 1 life. If the player dies needs to start over and cannot respawn in the location where the player died last time.

4- The player can use some special skills to defeat the monsters and boss.

After setting the limitations and rules for testing, testers will need a scenario that will help them what they exactly need to do and test during the testing.

Scenario

1- Starting the game

When the tester opens the game there will be a screen that allows test players to make a choice. In order to start the Gameplay, my test players were asked to choose the " New Game" option.

2- Controls

As I spoke about controls that were predefined in Unity testers should be able to walk around the world using W, A, S, D buttons.

3- Screen changes / Changing environment

After test players complete previous steps now they will be asked to perform a screen change test and in order to that they will need to move to the locations where they can warp to the next environment.

4- NPC interaction

After testers being able to switch screens the next step is talking with NPC and activation of quest which will help us to enter the boss cave.

5- Boss room

After the NPC interaction test players will return to the previous screen and they can see the rock in front of the cave will be moved to the side as a proof of quest activation was a success.

6- Boss battle

As a last step test players will go and challenge the boss to see if the battle mechanics and turn-based system works correctly and they will try to complete the game.

User ID	Starting The Game	Controls	NPC Interac-tions	Boss Room	Boss Battle
User-1	5	4	5	5	5
User-2	5	4	4	4	4
User-3	5	3	5	5	3
User-4	5	5	3	4	3
User-5	5	4	3	4	4

[2]Table: User Test Feedback(Own source)

After receiving the feedback from my test users I find out their biggest problem was controls and not having a good battle experience with the last boss which caused by me because I just make the boss very weak in order to make the difficulty of the game slightly easier for them to test those mechanics.

About my test users having issues with controls was also my mistake that I haven't informed them before how to play the game and what they need to click to. As an example, in order to interact with NPC, you need to press the left mouse button to activate the conversation window when you are close enough to NPC or the right mouse button to activate the menu to see stats items and others.

5. SUMMARY

As a summary of the work was done so far I can say I did learn a lot of new information, obtained new research skills, and definitely find a different way to express myself. While preparing the game and its stages I considered the goal and motivation. Researched, compared, and analyzed the same type of games and applications. Via analyzing popular games and applications I find combining two different kinds of applications is very hard and it will take more time than I assumed.

5.1 CONCLUSION

In conclusion, after I have begun my research about game development and how to become a game developer I noticed something. As a former smoker, I can say it is very similar to smoking. Once you start you will very much like it and it will start becoming a formidable challenge but also very fun that you will not want to give up on it. That was my main strength when I worked on the game.

I used unity as my game engine because Unity has very effective 2D and 3D game programming support. It is compatible with nowadays technology and operating systems such as iOS, Android, Windows, macOS, Linux, and others. Unity also has easy to learn user interface even if you don't have experience working on unity you can easily earn what is going on inside unity very fast as I did while creating the game and as a very good plus Unity has a very detailed user manual which has examples and many tutorials on the internet.

In a conclusion, while working on the project I was being able to reach a huge amount of information and concepts that helped me to process them and make huge progress to use this information.

Unfortunately, I was not being able to complete the subject of promises with not being able to complete the quiz implementation caused by lack of time. When I started working on this implementation I noticed it will take quite a long time before I finished correctly implementing it. So the product which was prepared for viewers is the alpha version of the role-playing game of alpha version with 1 level and 3 different map designs.

5.2 POSSIBLE FURTHER DEVELOPMENTS

One thing I learned while I prepared my thesis is there always exists a place for improvements in every software product. The desired work from the customer or user needs can shape the applications using user-feedback for example.

In the future I will be adding help sections to games I will create so people like my mother will not have a problem about which buttons do what?

REFERENCES

[1] K. Oxland, (2004). "Definition of RPG". K. Oxland from *'Gameplay and design'* (pp. 27). Pearson Education.

[2] *'Getting Started'* Visual studio code: https://code.visualstudio.com/docs , (2016, April 14), [Last accessed on 10.10.2020]

[3] Wikipedia Community, (2011, February 3), *'Game Design document'*, Wikipedia, Retrieved From: https://en.wikipedia.org/wiki/Game_design_document [Accessed on 01.12.2020]

[4] J. Suárez, (2017, May 10), *'The Importance of Narrative in Video Games'* Retrieved From: https://wibbu.com/importance-narrative-video-games [Last accessed on 17.01.2021]

[5] J. Harris, (2018, March 08),*'The Importance of Character Development in Games'*, 8.03.2018 Retrieved From: http://thenerdstash.com/importance-character-development-games/ [Last accessed on 10.01.2021]

[6] D. Johnston, (2003, January 30). '*What is Level Design?'* Retrieved From: https://www.johnsto.co.uk/design/level-design/ [Accessed on 09.11.2020]

[7] Wikipedia Community, W. (2007, October 29), *Item (game terminology)*, Wikipedia, Retrieved From: https://en.wikipedia.org/wiki/Item_(game_terminology) [Last accessed on 17.01.2021]

[8] Fu, Jiulin Zhang Xiaoqing. *'The influence of background music of video games on immersion.'* Journal of Psychology & Psychotherapy 5.4 (2015).Retrieved From: https://www.longdom.org/open-access/the-influence-of-background-music-of-video-games-on-immersion-2161-0487-1000191.pdf [Last accessed on 17.01.2021]

[9] *'Rigidbody 2D'* Retrieved From: Unity - Manual: Rigidbody 2D (unity3d.com) , (2021, January 11), [Last accessed on 17.11.2020]

[10] Wikipedia Community, (2007, Novermber 7), *'Turns, Rounds and Time-keeping Systems in Games'*, Wikipedia Retrieved From: https://en.wikipedia.org/wiki/Turns,_rounds_and_time-keeping_systems_in_games#cite_note-2 [Last accessed on18.11.2020]

[11] *'Prefabs'* Retrieved From: https://docs.unity3d.com/Manual/Prefabs.html , (2021, January 11), Accessed on 16.10.2020]

[12] Newzoo. (2020). *'Global Games Market Report'*. Retrieved From:https://newzoo.com/products/reports/global-games-market-report [Last accessed on 17.01.2021]

[13] ESA, (2019). *'2019 Essential Facts About the Computer and Video Game Industry'*. Entertainment Software Association, Retrieved From:https://www.theesa.com/esa-research/2019-essential-facts-about-the-computer-and-video-game-industry

[Last accessed on 17.01.2021]

[14] Straits Research. (2020, April 11). *'Top 10 Most Popular Gaming Genres in 2020'*.Retrieved From:https://straitsresearch.com/blog/top-10-most-popular-gaming-genres-in-2020/ [Last accessed on 19.11.2020]

[15] *'Debug.Log'* Retrieved From: https://docs.unity3d.com/ScriptReference/Debug.Log.html , (2021, January 11), [Accessed on 08.01.2021]

[16] R. DellaFave, (2014, February 27). envatotuts+. *'Designing an RPG Inventory System That Fits: Preliminary Steps'*, Retrieved From: https://gamedevelopment.tutsplus.com/articles/designing-an-rpg-inventory-system-that-fits-preliminary-steps--gamedev-14725 [Accessed on 17.01.2021]

[17] J. Doyle, (2018, October 29). Udemy. *'Learn To Create An RPG Game In Unity'*, Retrieved From: https://www.udemy.com/course/unity2drpg/ [Last accessed on 19.01.2021]

[18] J. Doyle, (2020, October 30). Udemy. *'Learn To Create An Action RPG Game In Unity'*, Retrieved From: https://www.udemy.com/course/unityarpg/ [Last accessed on 19.01.2021]

Tables

List of Figures

List Of Listings