Extended Project Qualification

2D Platformer Video Game Made with C# And Unity

By Ben MacIver

Contents

Introduction

The objective of this project to create a Single player 2D platformer hybrid video game designed using the Unity game engine and code written with C# can be broken down as follows:

- A single player game means that the gameplay loop (what the player is doing in the game and how the game world is built around that) is all built around one singular player and everything in the game and how its designed is there to accommodate for the one player. This is different to a co-op or multiplayer game where the games design is built around numerous people that interact with each other. This can make the game simpler to design but also makes refinement of the gameplay loop more important since it can lack the variety and unpredictability of a multiplayer game and the interactions with other players that it entails. Without this refinement the game may lack the replayability I desire for the game. The added simplicity of a single player can make it easier introduce more complicated gameplay mechanics that are seen in big Triple-A video games, that I may have struggled to implement if the game supported multiplayer. All of this contributes to making a game that is fun to play but also not too difficult for a novice gamer to try out.

- A 2D platformer hybrid video game means that at its core the gameplay design is that of the platformer genre, where the core gameplay consists of the players character using movement mechanics to navigate across platforms that create the level with the end goal being to get from point a to point b without failing, however this is complimented by additional gameplay mechanics and objectives that are rarely seen in platformers. This should give a unique but familiar feel to the gameplay meaning there isn't a particularly large barrier to entry but whilst still providing a unique experience to the player. This should all help keep the player engaged and want to continue to play the game. Furthermore, the hybrid aspect allows me to venture into the creation of lots of different gameplay systems seen in all kinds of video games and see how they can function together in one game.

- The Unity game engine is an application that allows a programmer to take their programming knowledge and implement it into a suitable software development environment for making video games. It essentially gives the tools for designing video games to the programmer however these tools are rather useless without programming skills and knowledge. C# is one of the programming languages that Unity supports, and it is the one I have the most experience with and therefore have chosen to use. The use of a programming language like C# allows the developer to properly use the tools that a game engine like Unity provides to build video games.

Literature Review

Before beginning this project I recently applied my programming skills in the C# programming language to the unity game engine, learning the fundamentals of how games are put together. This all means I knew what makes a game engaging and fun to play, and I also knew how to program a basic game, so all I needed to do know was hot to apply my programming and game development knowledge into the components of some of the most engaging gameplay systems I have ever interacted with. To do this I would have to learn and research how the developers from some of the worlds most prestigious studios created some of my favourite games.

I focused my research on four main sources and they are as follows:

- Grounded: The Making of The Last Of Us, which I chose since I already had pretty good foundational knowledge about this game, so I thought expanding on this with further research could give me incredibly extensive knowledge on the process of games development.
- God Of War | Raising Kratos | "Making Of" Documentary, which I chose due to how long and extensive the source was, with a full 2 hours of runtime which I felt could have lots of beneficial information that I could use.
- The Details of The Witcher (Music, Combat, Gwent & More),which I chose due to how it is similar to my other sources, and therefore may have similar benefits but is from an outside perspective which I felt would be useful in giving me the whole picture when it comes to game development.
- And finally I chose the book Blood, Sweat and Pixels, which I chose due to how detailed the observations made on the process of game development are despite being an external source and the wide variety of games that are covered at this detail level.

Grounded: The Making of The Last Of Us is an hour and 25 minute documentary published on PlayStation's official YouTube channel and is created by AREA 5 and Naughty Dog, that details the creation of the PlayStation exclusive video game The Last Of Us. Sony, the company behind the PlayStation, is one of the most well recognised companies in the gaming industry, and the company that they own, Sony Interactive Entertainment and the game development studios under it's umbrella, is behind some of the most well recognised video games to be made, with each of them consistently earning very high review scores. One of these games being the Last Of Us which is scored at a 95 on Metacritic, and its sequel The Last Of Us Part II scored a 93 on Metacritic and was won 7 out of the 10 awards it was nominated for at the 2020 game awards, one of which being game of the year. . This high reputation can be seen with The Last Of Us selling 17 million copies by April 2018 and it's sequel selling 4 million copies in it's opening week. The documentary also sits at 2.4 million views on YouTube. The success indicated by these numbers suggests that Naughty Dog ae obviously doing something right when it comes to making good games that the majority of people play. In addition Naughty Dog's large audience means that everything they say would be thought through and likely truthful since otherwise it would be called out by lots and lots of people. It is also worth considering Naughty Dog's vested interests when creating this documentary with video games considerably increasing in scale recently, they would want to make the studio look like an appealing place to work at to increase demand for employment. Furthermore, it has been reported in Jason Schrier's Blood Sweat and Pixels that after the release of Uncharted 4 (2016) Naughty Dog lost nearly 70% of its workforce due to burnout from the strenuous job that is game development. Considering this documentary was released in 2014 one year after the release of the last of us in 2013 it is entirely possible that naughty dog had lost some of its workforce after the games release and this documentary was created to attract new potential employees. This would therefore dictate that the documentary may have a hint of bias, focusing on the positives of game development that make working them seem better and also make the game overall look better to boost sales, considering that the last of us remastered was launching that same year on the new PS4. However, the bias is still going to be moderate since they would've faced extreme criticism due to their very large audience.

God of War | Raising Kratos | "Making Of" Documentary, is an hour and 55-minute documentary published on PlayStation's official YouTube channel and was directed by Brandon Akiaten in Collaboration with SIE Sony Santa Monica Studio. Like naughty dog Sony Santa Monica is also owned

by Sony Interactive Entertainment, which like previously mentioned has been responsible for some of the best and most well-respected video games of recent memory. God Of War itself is sitting at a 94 critic score on Metacritic and was awarded Game Of The Year at the game awards in 2018. On top of that the game also sold 5+ million copies in its first month. Belonging to the PlayStation family of studios also says volume about the quality of the studio. Sony Santa Monica is also very experienced in their own right having developed PlayStation exclusives for a long time, like god of war 1,2 and 3 on previous PlayStation consoles. The Documentary is a very reliable source since it is licensed by PlayStation, which like previously mentioned is very reputable, not only from a business perspective and how much money the company makes of their games, but from a quality perspective with the consistent quality of PlayStation's exclusive titles being celebrated by fans and critics alike which is reflected in their high review scores. This all suggests that the information from this source is reliable and more importantly coming from people who know what they are doing. Source, furthermore. It has been viewed by 1 and a half million people on YouTube so it would be incredibly difficult for Sony Santa Monica to mislead such a large audience without significant backlash. However something that needs to be considered is Sony Santa Monica's vested interests. Considering the documentary was only uploaded 1 year after the games launch it is clear that they would want to make the game look as good as possible and try and hide its weaker areas to encourage people to buy it. They would want to increase sales of the game as much as possible. This financial reward may also be worsened by a want to attract new developers to come and work for them, and therefore they want to make the studio look like a good place to work. This is especially important since games are getting bigger and more expensive and longer to make, so they likely need the extra workforce, which is particularly relevant considering the sequel for the game is right around the corner, with it set to release on the ps5 in 2021. This could all introduce a degree of bias however the room for bias is limited due to the large audience the documentary would have, that have the potential to scrutinise them for everything they choose too include and everything they don't.

The Details of the Witcher (Music, Combat, Gwent & More) is a 26 minute long documentary created by and uploaded to YouTube by a company called. NoClip are a crowdfunded media company who make documentaries that are requested for and funded for by their online fanbase. At first glance it may seem like NoClip may lack some of the reputation as a source, when compared to the gaming giant that is Sony interactive entertainment, a combination of their consistency at which they deliver in depth, detailed and accurate documentaries to a large audience and the calibre and profile of the games studios they have collaborated with in these various documentaries makes up for this. They are a secondary source, with no role in the creation of these games but their documentaries feature in depth interviews with various people who took part in the creation of these games. And as much as NoClip aren't as well-known as some of the games studios I spoke of previously, the polish games studio CD Projekt Red who made The Witcher 3: Wild Hunt have quickly become one of the biggest games studios in the industry over the last decade, which as of April 2020 has sold 28.3 million copies across 4 platforms. The game's influence has been very far reaching with its surge in popularity leading to the creation of a mobile game from CD Projekt Red and a Netflix TV show in 2019 with actor Henry Cavil staring as the protagonist. It is important to remember though, that as much as CD Projekt Red and NoClip have become rather reputable sources their vested interests aren't as simple. As much as NoClip is a crowdfunded company who don't answer to any big publishers in the games industry and will be making their documentaries to please their viewers who they rely on to make a profit CD Projekt Red might have slightly different motivations. CD Projekt Red may very well be wanting to capitalise on their surge of popularity, by making the game look

better and focus on its much talked about strengths to make the development studio look like a better place to work and attract employees for all the different projects they are taking on, like the mobile game previously mentioned but to also build the studios reputation and consequently anticipation for their next triple-A game Cyberpunk 2077. If this was their intention it obviously worked, with the game releasing in December 2020 and garnering 13 million sales at release despite the abundance of negative PR around the release. In addition, it cannot be overlooked that NoClip relies on the cooperation of games studios like CD Projekt Red to make the content that their viewers are funding and if they hurt their relationship with CD Projekt Red in anyway this could make the production of future documentaries incredibly difficult, which they obviously wouldn't want to ensure that they maintain funding from their viewers.

My final source is a slightly different one, with it being Blood Sweat and Pixels, which is a book created by games Journalist Jason Schreier. This book gets its credibility from the numerous interviews from many well-known games studios, the fact Shreier has ensure that all accounts from his sources were corroborated on by at least two other people involved and simply the profile he has as a journalist In the games industry. Some of the news stories he has written have been incredibly talked about have been some of the most talked about amongst gamers and game developers like his piece on development studio Naughty Dogs crunch culture in March which co-president of Naughty Dog even acknowledged and commented on months later. He also has reliably leaked many things, such as details on The Last Of Us Part II's massive story leaks that rocked the internet in April of 2020 a day before Sony itself confirmed the report, in addition in 2018 he successfully revealed a massive change in one of Bethesda Game Studio's biggest Game Franchises Fallout before the company themselves revealed it. All of this proves that Jason Schreier has plenty of credibility and reliability to his words despite only being one man. Rather interestingly Jason Schreier's vested Interests are very different to the 3 previously mentioned sources. Jason Schrier is a website writer, so he doesn't answer to any games studios or publishers, furthermore he also doesn't benefit from the profit the games he is talking about make, unlike some of the previous sources. He isn't worried about the image of the game studio simply pursuing facts from the developers even if he may be inclined to be selective about what he draws attention to in order to ensure he creates a book that has enough content that gets it talked about lots, like many of his news stories or news leaks are talked about, in order to bolster sales of the book. Being a journalist does have the caveat that Schrier himself isn't a game developer and isn't directly involved with the creation of the game, and therefore he may lack some of the expertise of the actual game developers and development studios that helped with the creation of my other sources even if he quotes the developers he has interviewed directly. One thing that shouldn't be worried about with Jason Schrier is potential bias. The fact that Schrier isn't directly involved in the development and sale of the games he is talking about means he is not going to focus on making the studio look as good as possible and may be more likely to look at things more objectively.

Methodology

Relying on pre-existing research in order to guide me through the development of my own game wouldn't be smart, as it is too detached from the specific game I have created. Therefore, following the system, I have outlined below I gathered my own data to help inform the development of my game. Before beginning the project, I questioned a number of individuals who are either

knowledgeable when it comes software and games development or play games themselves and therefore are likely to fall under the target audience for the game I have created.

I began by devising 3 questions that I would ask people, which would help me when trying to make a fun game. This allowed me to ensure the game is something people other than myself would want to play. My plan was to ask these questions before starting the game, and then half way through show them the game and then ask them whether what I have done so far has somewhat met the requirements/ question answers they outlined at the beginning of the project, and I told everyone I questioned that this was my plan before questioning them.

The first one was simply, what makes fun gameplay? This is important as gameplay is at the core of every game, and if it isn't fun people won't play the game for very long.

My second question was how long do you tend to play single player mobile games for at a time. The reason I ask this question is to gauge an idea of for how long I need to keep the player engaged and not bored of the gameplay, however simply asking "how long do you play games for" wouldn't be very useful since play session time varies between single player and multiplayer games, and between smaller scale games like mine and a triple-A game. This is why I specified the game as single player, and as a mobile game, since simply from playing mobile games throughout my life I have found that the scope of a typical mobile game is similar to the scope of my game. It would be unfair to compare an open world game with a 40 hour main story with my much simpler game.

The final question is how difficult should a game ideally be. The purpose of this question is to help me gauge how I should balance difficulty in my game, since if a game is too easy the player will likely get bored to fast, and its too hard the player will just stop playing because the game is too unfair. Many games give multiple difficulty options to solve this problem but I don't really have enough time to create multiple balanced difficulty levels.

In the end I was able to get 3 people who could contribute to this part of the project, one being a knowledgeable software developer who is also a casual gamer, and the other two where a 15 year old and 18 year old who both play lots of different video games on a daily basis.

Discussion

Solution to the Problem

Starting the development of a game can be quite overwhelming, with the first steps not being clear. Therefore, I tried taking a logical approach to the project, taking the project Brief as a problem that needed solving, and therefore I needed to find a solution to that problem. This solution had to meet all the goals described in the project brief and be efficient enough so that I can feasibly complete it in the time I have.

When it comes to developing a game, I've found that there are two main components that need to be considered before you can begin, and that would be how the game Looks and how it works. How the game looks covers stuff like the Assets required for the creation of the level or the characters and items that inhabit the game world, all things that make up the presentation of the game. How the game looks also includes the creation of the games user interface. And then you have how the game works which is essentially the code the user would have to write to make the assets interact with each other and do the things that make the game fun. Without this portion of development the game is simply a collection of images with no intractability and no logic assigned to them. There is of

course many more important aspects to developing a game, especially in bigger projects from studios like Naughty Dog or CD Projekt Red. However, those studios have hundreds of employees all working on one project over multiple years whereas I am just one person with much less time. Therefore, due to the smaller scale of my project many of these important aspects become less significant and can be encapsulated under the 2 main components I previously mentioned, however just because some of the areas of making a game for Naughty Dog or CD Projekt Red aren't all that useful to me, how they break down some of the other areas that do cross over into my project, like combat or level design, is incredibly useful in guiding me how to successfully make these areas fun to play.

Designing the Game's Level

The first thing required to make a level is the terrain, since that is the very thing that the level is made of, and to make a terrain you need assets. So, before I opened the asset store and found a free asset pack for the terrain I had to make a choice. Is there a specific theme for the game? This is important to consider since almost all games, especially those with a focus on their story, have a specific theme to them and the presentation of the game is arguably the most important aspect when it comes to presenting this theme. An example of this would be with The Last of Us. The Last Of Us is a post-apocalyptic game set 20 years after a global pandemic wiped out humanity and left it in ruins, and the levels of the game reflect that theme/setting. As seen in the images below the levels in the game constantly reflect the theme with the player navigating abandoned city buildings that are falling to pieces with vegetation and sewage covering their remains due to the lack of upkeep from the people that once lived there.

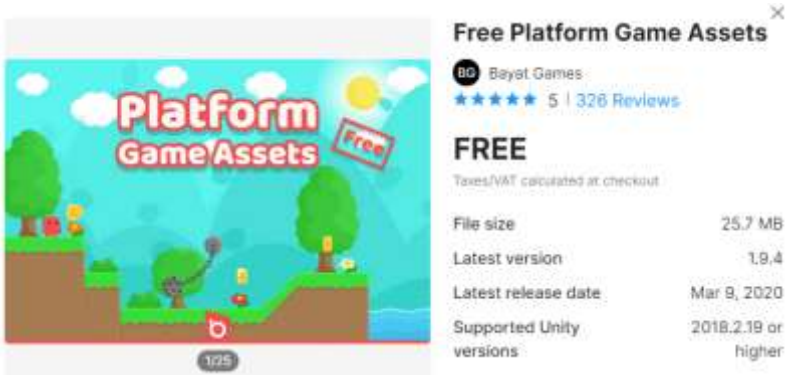
Concept art for The Last of Us

In game screenshot of the finished product

The assets that make up the levels of the last of us have been chosen and designed the way they are because the game has a specific theme it is pursuing, and the levels must reflect and support that theme.

However, my game is much simpler and telling a story is not one of my objectives, simply because that would deviate too much from my project brief, and the reason why I don't make it part of my project brief is because the resources and time required to firstly create a narrative for my game and then secondly to make changes to the game for it to accommodate for that narrative would greatly exceed the resources and time I have at my disposal. Therefore, a specific theme isn't a necessity like it is in the last of us. I do have the option to find an asset package for my level that has a more science fiction feel, or maybe if I chose a western theme I could look for dessert assets for my terrain. However, if I am tied to a specific theme then I am somewhat limiting myself since once a theme has been choosing every single facet of that game must reflect that theme, an example of this being the Witcher 3: Wild Hunt, where the gameplay mechanics the player has at their disposal and the enemies they fighting are all derived from the fantasy setting established by the theme of the game. For example, if I chose a western theme then I would have to ensure every part of the game reflects that theme, which would limit the assets I have at my disposal as they all need to reflect this theme . This would also mean that I could not implement melee combat, which I am familiar with and, instead must spend more time learning how to implement gun combat and shooting mechanics since that is what is most suitable for a Western theme. Needless to say doing this would take lots more time and wouldn't be the most efficient way of me achieving the overall objectives of the project, which doesn't include a specific theme that the game must be tailored to. Therefore, I chose to go for a neutral themed terrain that could accommodate for essentially any theme. This means I am not limiting myself when it comes to finding assets for things like the player character.

Now that I decided the kind of theme, I want for the level I now just need to search a free asset package with some assets for the terrain that fits the neutral look I was looking for. I ended up settling on "Free Platform Game Assets" as seen in the screenshot below.



The assets from this pack that I would use to create the terrain in my level are simple grass assets of varying shape. These are suitable since they could be used with a game of many different themes as they aren't too specific. It also had matching background assets to complete the visuals for the level.

The next important part of constructing the level to consider is how I will go about actually putting it together. I had 2 options at this stage, either create the level by manually dragging and dropping assets into the scene editor and designing the level that way. Unity also supports a method called Tile maps. A tile map is a unity tool which handles the creation of 2d levels. It does this by taking the assets you provide it and allowing you to paint them onto the level using a grid it creates. This is a more efficient way of constructing levels than the manual approach I previously mentioned; however, I haven't used this method before and therefore would have to spend time learning how to get my assets into the supported format for tile mapping and learn how to use tile maps as a whole. While this may not take an overwhelming amount of time, considering the rather small scale of my game the potential efficiency I gain from using this method would likely not make up for the time lost learning how to use tile maps, therefore I elected to use the manual method of creating the method in the scene editor that I previously mentioned since I was more familiar with it, and in the grand scheme of things it is the more efficient approach for this project, which is an important objective.

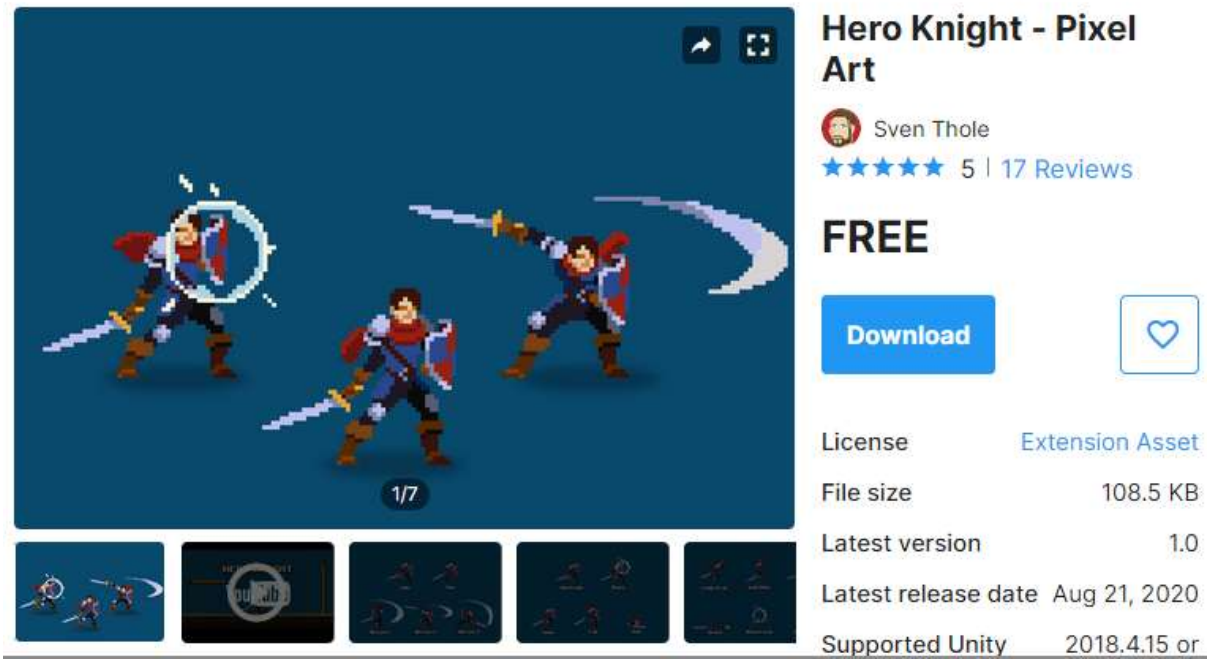
Choosing the rest of the assets

So, besides the level there were three main things I that I knew I needed to consider when it came to how the game looked, outside of the user interface (which I chose to design at a later date once the core gameplay was designed), and they were all assets for the player character, the enemies and finally the collectables.

Now firstly I think it's necessary to explain what I mean by collectables. The idea for collectables was that the level would be littered with small in game objects that the player could go out of the way to pick up, with the reward being, the more collected the more score the player earns. For this I was going to use the following coin asset that can be seen in the image below.

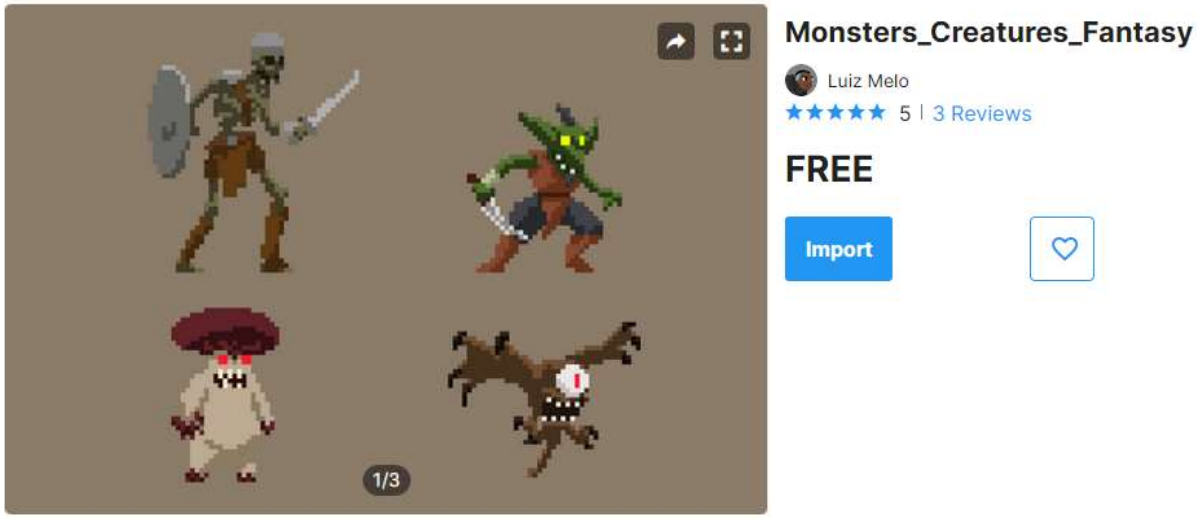
This asset was a part of the Free Platform Game Assets package I previously mentioned so it fit very well with the aesthetic of the overall scene and it had premade animations that brought the image to life, rather than it being a simple static picture. All I had to do was drop it into the scene and then write the code for it to work. However, I ultimately came to the decision to cut collectibles from the game entirely, since I ended up spending a lot of time writing the code for them and trying to get them to work properly but had no luck. It was simply a case of the implementation of coins creating to many problems during development, that the reward of them working wasn't worth the time they were necessitating. I will go into more detail about why they weren't working in a later section, where I talk about designing the code for the game as a whole.

Now that I've explained the planned role of collectables in the game, I think I should address the first of the other 2 game objects that I created and had to find assets for and that would be The Player Character. Any game where the main gameplay loop is designed around the control of one player, the player character is arguably the most important aspect of designing the visuals for the game since it's the very thing that the player would be looking at for their entire time playing the game. So for this I wanted something, that I thought looked good, fit the game but also had the animation library I needed for the gameplay I had in mind. This is because I know how to program, and I know a solid amount about how to use the unity editor to make games but I have not ventured into making my own animations and doing so would take quite a bit of time for something I don't really need to do, since if im going to be getting all the assets that make up the visuals of the game from external sources then I might as well do the same for animations. So after browsing the unity asset store I settled on the following asset package for the player character.



As the image illustrates quite well the "Hero Knight – Pixel Art" asset package by Sven Thole had all the animations I needed from melee combat to jumping animations. This provided the required support I needed to create the desired gameplay systems of player combat and player movement. The assets also have a very high-quality look to them, and due to the rather neutral tone of the existing visuals the hero knight did not look out of place.

The final decision required when it comes to choosing assets for the game was deciding what assets i would choose for the enemy ai. The requirements for the enemy ai were not too difficult to meet as all I needed was an asset package with idle, walking and attacking animations built in and an enemy character that fit with the chosen player character. So, after browsing the unity asset store I found this asset package that fit those requirements.



This asset package had a small selection of enemy characters that all fit the visual style of the game, each had a library of animations that I needed, and most of them didn't look to out of place. However I only need one type of enemy character for the game and decided to go with the skeleton (top left) since I felt it looked the least out of place for my game.

Designing the User Interface

Firstly I should probably outline that a User Interface is simply anything in the game that allows for the game and the player to interact with each other. It's how the player tells the game what it wants to do (eg: quit game, start new game, pause) and how the game tells the player information (eg: time passed, current score).

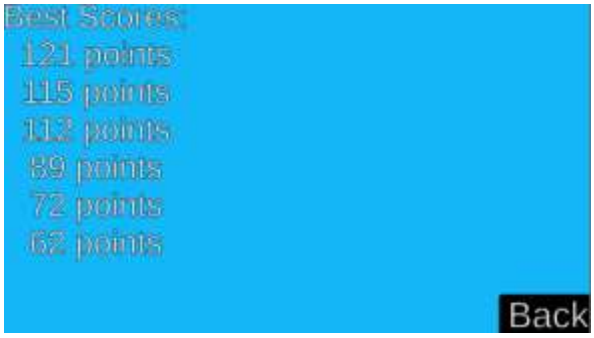
Designing the UI for such a simple game, while important, is not too complicated. It basically just needs to serve its purpose, look presentable and not be too intrusive on the player's experience. The first component of designing a UI would be the heads-up display (HUD), which is everything that appears on screen over the top of the game while it is being played. Now the most important thing about this is that you do not want it to be intrusive, since the player does not want to be looking at UI, they want to be looking at the rest of the game since that is what makes the game fun. So, if I want the HUD not to be intrusive it should simply do what it must do and nothing more, which is tell the player crucial information that they may need at any moment. Of course, this changed as the game was developed since the game at the start and end of development is very different, therefore something that may be crucial information initially may not later. Consequently, I added HUD elements as and when I needed them, deciding how the hud looked second and putting practicality first. All I required from the look of the HUD was it to not look downright bad that it makes the player lose interest in the game faster, and of course not be too big to limit intrusiveness. Below is a screenshot of the heads up display in its final state.

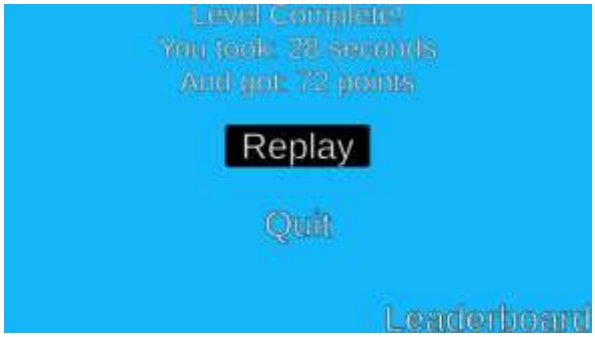
As can be seen, the HUD is all located at the very top of the screen, which is nice and out of the way from where all the important things going on in the game are, ensuring they are not intrusive. Furthermore, they are big enough that they can be quickly and easily read when needed to from the player, but not to big that they get in the way of everything else in the game. All the text in the heads up display also has a black outline to further improve readability.

The next part of UI is the menu's which are important for providing functionality of the game, but outside of that just need to look nice enough that they do not make the game feel like it is of overall lower quality.

For the start screen of the game, I avoided the use of buttons and simply gave a button prompt to tell the player to click Enter to start the game. It is simple and does the job of telling the player that the game has loaded and is ready to begin and allowing for them to get into the game as quickly as possible. Bellow you should be able to see a screenshot of this menu.

For the Menu's look and presentation, I used a light blue background to fairly closely match the background of the main game in order to maintain consistency, and not have the switch from menu to the game feel to jarring, and for the text I stuck with the same font and black outline used in the HUD of the main game. I also added a black highlight over the text that the player currently has selected on all the menu's so that they can successfully navigate the menus with the arrow keys and know what is about to happen when they hit the Enter button. Below you can see some screenshots of the Menu's that exist throughout the game.





Writing the games code

Now that I have outlined the details about how the visuals of the game were developed the next very important of game development that I need to go into is how I made the game work. Making the game work is all about ensuring that it acts as intended whenever the player interacts with the game and that it can be played all the way through without encountering any fatal errors that stop the game from running. Now all of this is done by writing code, which was the backbone of the development process for the game. Managing my time while making this game was done based upon what piece of code I had to write next, and therefore its fairly easy for me to breakdown the process of writing the code for this game, and the stages of development. Firstly, I created the all the algorithms for the Player, then I had to design the algorithms for the collectables (eventually cut from the game), leader boards/timer, enemies and scoring.

Writing the code for the Player's Character

The code for the player was added to throughout development but except for those initial stages of development a lot of the added code is there to serve functionality to other pieces of code. Most of the code for the player character can be found in two C# scripts called PlayerMovement.cs and PlayerStats.cs. Player Movement was the first script I created out of the two and was the one which required the most decision making throughout its development since PlayerStats's purpose is largely just to serve other scripts and aspects of the game, allowing their functions to be executed properly.

When writing the PlayerMovement script (which governs how the user can control the player character, and then how that player character interacts with its surroundings), the main decision I had to make, was the input method required to move the character. I could either use a controller joystick or keyboard presses. The controller joystick is the input method I was more familiar with using; however it is more suitable for 3d games where the player can move in any direction within a 360 degrees circumference. My game however is 2d and the player character can only move left,right or up and therefore the added mobility of a joystick is necessary so I chose to use the arrow keys on a keyboard. This makes the game more accessible since all the user will require is a computer keyboard, leaving out the need for extra peripherals like a controller, that could be a barrier of entry for some. The rest of the time spent developing player movement mechanics was somewhat straightforward when it comes to decision making since the outcome will always be the

same its just a matter of how you get there. You can have all kinds of algorithms for 2d player movement but at the end of the day they will all function relatively similar and only someone with programming knowledge would be able to tell the difference by looking at the code behind it.

Writing the code for collectable items

After having the game in a state where it is inhabited by a player-controlled character with a full range of movement mechanics the next step I pursued was scoring mechanics, since so far the game was just an empty level with a player character that can run around in it. This is where I decided to implement collectables in the form of coins scattered across the level. Collectables are simply things that the player can walk over to and interact with for varying outcomes. The purpose of the collectable coins throughout the level where to be incentive for the player to go out of their way to collect something that may not be on the direct path but can provide the benefit of extra points. However, after spending lots of time trying different approaches to allow for the player to collect these coins and them disappear, I kept on encountering issues. The coin was either ignoring the player was even there trying to pick it up, or even worse blocking the player from moving through the level. Eventually I had to revaluate the situation and recognise I was losing a lot of time trying to get these collectibles to work for a purpose of simply laying the foundation for a scoring mechanic in the game, which is something I can do without collectibles and instead base the scoring mechanics around the speed of the player and the number of enemies defeated in combat.

The approach I ended up settling on for a scoring mechanic did meet the same goal I was aiming with the collectable items; it was just a case of shifting approaches towards something that provided less problems and didn't halt progress of the project for too much time.

Creating the scoring system

Due to the shortcomings, I encountered when implementing collectable items into the game as a foundation for a scoring mechanics I began to pivot my focus towards another method to create a scoring system. This became an important focus since without some sort of scoring system the game quickly becomes much simpler and less engaging and just another basic and generic 2D platformer, failing to meet the project brief. I needed a relatively in-depth scoring mechanic to facilitate extra gameplay mechanics that can tie into the main gameplay loop of a 2D platformer. Without this the game would fail to meet the project brief of "2D Platformer Hybrid". This is where I came to the decision to use the initially planed timer as part of the scoring mechanic.

Firstly, I wrote code to have the amount of time the player has been in the game displayed at the top of the screen as part of the heads-up display. This replaced the previous player goal I was going to establish of getting the most coins, with the goal of completing the level as fast as possible with the timer at the top facilitating this goal. However, I had to make it so that the player actually had an interest in completing this goal and this is where I decided to implement a leader board. For the time being the leader board simply displayed the players time which provided motivation for the player to be as fast as possible, but this was only temporary as once player vs ai combat was implemented I could use that in combination with the player's time to replace the scoring system I planned to base off the collectable coins I removed.

Implementing enemy AI and player vs enemy combat

The final step towards fulfilling the goals set out by the project brief was to implement enemy AI and enemy and player combat. Player combat was already full implemented with animations for swords

swings being triggered as soon as the player hits space bar. After a little work I replicated this mechanic with the enemy skeletons I had littered across the level so and written the necessary code to have the skeletons attack the player when they were close enough. Of course, I also had to have the enemy skeletons walk towards the player which required the use of some sort of ai. At first, I tried using the built-in unity ai systems In my code to force the enemies to follow the player so they can engage in combat however, much like my attempted implementation of collectables I kept on hitting brick wall after brick wall trying to do that. I found that because unity is primarily focused on 3D games their ai systems are also designed for 3D games and trying to implement a system which is quite complicated when compared to the complexity of my game just introduced so many more problems to be solved that I needed to find another solution. All was not lost however, since that simplicity of my game that I mentioned allowed me to just write the code for the enemy ai myself, creating the ai systems from the ground up. Furthermore, this wasn't as long and difficult as it may sound because all the enemy ai had to do was go from following the player to not following the player based on the players position relative to the enemy. All I had to do to get this was write a piece of code to compare the positions of both objects and then have the enemy's position change dependant of that.

With enemy ai and combat functioning I was close to reaching the goals of my project brief, all I had to do was integrate this gameplay system with the scoring mechanics. To do this I updated the HUD with the amount of kills the player has and developed an algorithm that uses a simple equation I created to combine both the players time and their kills to produce a final score. With the scoring system finished the game was fully complete, all it needed now was a little testing for any bugs.

Responding to player feedback

With the creation of my game I was following the answers to the small questionnaire about the fundamentals of a video game that I outlined in my Methodology section. This often informed my approach to designing gameplay systems, for example the need for a game to keep them entertained for at least 20 minutes helped when designing the games level and ensuring it was long enough with enough variety to not become boring within those 20 minutes.

Another example of how I had to use player feedback would be how I used the fact that people wanted easy to learn controls and a moderate level of challenge to keep the game fun for there average play session. Most of the people I asked wanted the game to be challenging, since challenging content has been some of the most repayable content in video games they've played but if a game is too challenging it becomes less fun and they end up putting the game down after not very long. I also saw people explain how if the controls for a game where too convoluted it would be harder for them to begin to enjoy it and would also make it harder to come back to after time away from it. So with this information, outside of the 4 arrow keys for movement there were two other controls, one button for jumping and another for attacking enemies, establishing all the fundamentals of player input while keeping the controls simplistic and easy to learn adjust to. I then spent time balancing the health of the player and the enemies to try and ensure enough difficulty to keep players from getting bored but not too hard that players gave up on the game because it felt unfair. My approach to this was just replaying the game myself until I could manage to complete the level most the time, but not all the time. This is where my experience playing video games was useful, as overtime I have became familiar with when a game is too hard and when it is too easy. However just me playing the game to test this wasn't enough, so I got those who had took my small questionnaire, to play the game a couple times in its current state and give me feedback about what

they had outlined before, which included comments on difficulty. Most people liked the player vs enemy combat and the variety and challenge it provided and approved of the overall challenge of completing the level. I also had a comment from someone who liked the use of a timer as it made them want to go faster and beat their last time. However when playing it, someone did bring up how often, the best way to get the best time, which had become an important part of the gameplay loop was to to just run past all the enemies, so I obviously had to increase the role of enemies in the game, however this was initially difficult as the enemies where already taking up quite a bit of space on the level and dealing quite a bit of damage to the player, so making them harder to fight could risk becoming annoying and too difficult which is where I decided to introduce an "enemies killed" stat, which contributed to the players score. This meant the enemies where still balanced and didn't make the game too hard to play, but running past them was no longer the best strategy as the amount of kills you got affected your score, which helped make the game more challenging but not unfair. The overall feedback I have had of this addition in the finished version has been positive with players liking the difficulty level.

Conclusion

Overall, I believe I have been very successful in achieving my project brief. The description of my game outlined in my project brief fits what I have created, and all boxes it outlines have been ticked. The only part of my project brief which I feel I could've achieved to a better extent is where I outline the game as a "platformer hybrid video game" which while still true of my finished project, I feel that is only by a slim margin and if I was to do the project again, maybe with more time then I think I would've put more time into how I merge the platformer genre with gameplay mechanics seen in other genres, since how I've done it while good, isn't as extensive as I anticipated with only one real game mechanic that uses aspects of the "2D platformer genre" and other genres like action and adventure games and merges them together. Adding some more of those kinds of mechanics and developing the existing ones further would've been nice given more time and maybe slightly improved planning of the project to accommodate for this added time. However, that is something I would do If I had a little more time to better achieve my project brief than I already have, since if I had a considerable amount of extra time then I would take the project much further than the original brief. The first idea that comes to mind would be to replicate what I have created over more game levels with additional gameplay mechanics and situations and then include some form of plot/narrative to the game so I can tell some sort of story through may game, even if it is simple. It would be incredibly interesting to me, to explore the medium of video games to uniquely convey a story to the player and really get them to think a little further about what they are doing rather than just doing it for simply fun and enjoyment. This would be similar to how games like the last of us or god of war use video games to uniquely tell stories just on a smaller scale but to the same effect. This would clearly require lots more time and resources of which I simply didn't have for this project but If I had more of those, that is where I would take my project in the future.

Evaluation

This project as a whole has allowed me to improve my skill as a researcher but has also allowed me to see areas of which I need to improve.

Firstly, I am quite impressed with myself in how I have managed my time and set deadlines for myself in order to not fall behind on the project which is something I have never really had to much in my school life, with many of the deadlines being set for me. This was especially true when completing work in lockdown, when I was at home with much less input from teachers, and only had

one final deadline for the whole project, and I had to take the time that I had between where I was and the deadline and look at what other commitments I had in life and other school subjects and plan out when I would get my work done and when I would get that work done by. Furthermore, due to the aforementioned Lockdown that happened in the beginning of 2021, deadlines for this project had ben changed which required me to adjust but still maintain the deadlines I set for myself, and I feel I did this quite well as I made sure I utilised the extra time I had, but still met deadlines I had set to not fall behind on my workload. In addition to improving my time management skills, I found myself very quickly learning new skills when It comes to research, one of those being how I apply my research. Beforehand I only had moderate experience with research and even less with how to apply that research to a project, and therefore I had to quickly learn how to do that with this EPQ, which I did a pretty good job of. I managed to get lots of research done in a short amount of time, and I managed to use that research to help me approach the creation process of my game and using what I had learned, applying it to my everyday throughout process when working on the project. However, I could've done a slightly better job at gathering a greater range of source types for my research and done a better job of using those resources more equally when developing the project and not favouring one or two of them as much as I think I May have. One area where I could've done things better can be seen with the twelve different objectives I made under the "Activities and timescales" section of my project proposal form. While I did meet all of those objectives a number of those objectives ended up just becoming parts of other bigger objectives of which I managed my time around achieving, so maybe I could've done a batter job of managing my time around the 12 different smaller objectives in order to ensure each got equal time a resources dedicated to them, since I believe that may not have been the case for my project. Another area that I completely overlooked was the need for primary research, which I believe came from a lack of research experience. Going into this project, given the age of technology we live in and how easy it is to obtain extremely useful and extensive information on the internet I found myself relying extremely heavily on secondary research, and due to a lack of research experience I forgot and overlooked the importance to spend an equal amount of time on my own primary research. However this has been beneficial it teaching myself to balance my research in the future, a skill I previously overlooked.

Overall though, despite highlighting some short comings, I feel that as a whole I have achieved my goals rather well and developed valuable research and general academic skills, and where I have had short comings those areas have provided valuable lessons and information about myself as a researcher that I can use in future projects.

Bibliography

Name	Organisation/ Author	URL	Date Accessed
Grounded: The Making of The Last Of Us	Naughty Dog and AREA 5	https://www.youtube.com/watch?v=yH5MgEbBOps	26/09/20
God Of War | Raising Kratos | "Making Of" Documentary	SIE Sony Santa Monica Studio	https://www.youtube.com/watch?v=ra_R-K_IoUc	03/10/20
The Details of The Witcher (Music, Combat, Gwent & More)	NoClip and CD Projekt Red	https://www.youtube.com/watch?v=tDQPScB3bYA&t	10/10/20
Blood, Sweat, and Pixels: The Triumphant, Turbulent Stories Behind How Video Games Are Made	Jason Schreier	https://jasonschreier.com/	15/10/20