

Games Development

Unit 47

Matthew Crosby | 10170652

Contents

[Task 1 2](#_Toc7558913)

[The Game Type 2](#_Toc7558914)

[Story 2](#_Toc7558915)

[Characters 3](#_Toc7558916)

[Magma 3](#_Toc7558917)

[Warchief 3](#_Toc7558918)

[Common Enemy 4](#_Toc7558919)

[Droghunter (Boss) 4](#_Toc7558920)

[Levels 4](#_Toc7558921)

[Gameplay 5](#_Toc7558922)

[Art 7](#_Toc7558923)

[Sound 8](#_Toc7558924)

[User Interface (UI) 9](#_Toc7558925)

[Controls 10](#_Toc7558926)

[Task 2 10](#_Toc7558927)

[Methodologies 12](#_Toc7558928)

[Waterfall 12](#_Toc7558929)

[Agile 13](#_Toc7558930)

[Scrum 16](#_Toc7558931)

[The Model I am Using 18](#_Toc7558932)

[Conclusion 21](#_Toc7558933)

[References/Bibliography 22](#_Toc7558934)

# Task 1

# The Game Type

The game type I will use will be a combination using different concepts, but the main genre/game type will be hack and slash, this is a game genre that emphasises on combat, usually melee combat. Hack and slash games first started as board games such as Dungeons and Dragons, which is still popular today. Now, most hack and slash games are either considered Role Playing Games, such as Diablo, or Action games such as Devils May Cry 5. These are games where, as previously stated, emphasise on combat, it makes it a necessary component to the game. With other games attacking can be very basic with simple swings and hits but in a hack and slash game, it makes the player know that they have hit a target, this could be by slowing down time on a hit, or pausing the sword swing animation for a second before continuing to make the player feel as if they have hit the target.

The reason why games have a game type is for many reasons, one of the main ones is to be able to categorise what game it is which is useful in online stores such as steam, this allows for players to pick what game type they like and then get displayed similar games based on what they selected. Another reason why the game type is important is because it gives the game structure, if the game didn’t have a type then it would be all over the place with features that may not work well together. By having this structure, it allows for the game to follow a template in a way, not so much with the game itself but with the concepts of how the world and characters will act. If you take first person shooters for example the enemies will always shoot back, and in more advanced games they will take cover. This is based of the concept of a first-person shooter.

# Story

Within the world of Lost Origins there is a hidden island, on this island there are two clans, Doomford and Ravenforge, who are currently at war to determine which clan will lead the rest of the island, but it was not always like this, at one point the clans got along with trading different materials, food, and ores. Back in the first era a Chieftain called Gristomp had a son gifted to him by the gods, this son was granted the power of the elements to be able to conjure magic and increase overall strength. When the son was young, he trained like any other young warrior on the island to be able to defend it from threats but as the son got older, he was guided by his father to go down his own path. This path allowed him to harness his abilities to his max potential as only Gristomp knew the importance of his son gaining control of the power he was born with. Gristomp foretold a time where the chieftain of Ravenforge would abuse stolen power from the gods to rain terror on the rest of the island and only the power of the son may defeat this evil. In the 5th Era the island was at war between the two clans, at this point the son learned a small portion of the abilities he could do with his power. More recently a group of members from Ravenforge came to the village of Doomford in the middle of the night and killed the father, who also happened to be the chieftain of the clan. Filled with rage the son goes to the village of Ravenforge and take matters into his own hands. With the guidance of the war chief the son learned how to channel his rage into abilities that aid them in their conquest for revenge. Once enough rage had built up, he can unleash this to deal more damage and ferocious attack on their enemies. His main objective is to draw out the chieftain of Ravenforge and put an end to his questionable methods of domination. The personality of the son depends on the playstyle of the player, making them silent protagonists.

The reason that a story is important is because it also gives the developers a path to follow when creating the game. An example is a popular game series called The Witcher, this game series was based off events that had occurred in The Witcher books, this allows for the player to connect to the character and understand what has happened, but as well as this it helps link both the player and in game character to the same motive, for example the story could be some family member being killed at the start of the game, this would then build emotions in the player and the in-game character to want revenge. This also links in with the games narrative side. This is important as it helps the player to feel more involved and immersed within the game. It also gives meaning to the game and helps include the player as part of the story.

# Characters

## Magma

Magma is the son of Gristomp, he channels his energy from the auras around him to fuel his primary abilities such as Healing or Blinking. Magma was raised by his father to protect and guard the earth of the island. Gristomp was lucky to be granted a child with such abilities as a child like this is only born once every 500 years with a 30% chance of survival. Magma is a skilled swordsman and a skilled mage training in both field for years, and with the help of his abilities the time needed to master the skills of a swordsman and mage was significantly decreased.

## Warchief

The war chief was the main adviser about war and defence for the father. Now his main role is to guide Magma into finishing the war that they didn’t start. The war chief in the game does not play a big part in the game, instead he is intended to talk to the player for guidance.

## Common Enemy

The common enemies will have an ability to attack and follow the player, when one is killed the player will earn a skill point, from here the player can use these to upgrade their abilities through the war chief.

## Droghunter (Boss)

After a number of enemies are killed a meter will build, this meter being the rage meter of Droghunter, after this meter is full a portal will spawn, and the player will be able to teleport to the boss. The boss will have some abilities that the common enemies won’t. As well as this the boss will deal more damage than the common enemy.

Having different characters within the game allow for the player to form bonds with specific characters. If every character in the game was the same, then the player wouldn’t feel any type of connection to any of the characters so by having different characters with different personality’s it allows for the player to form these bonds and be more immersed. An example is in The Witcher 3 Wild Hunt, in this game there are many characters, some characters with little importance to the game share the same model with different clothing whilst important characters get their own mesh and voice, as well as this in The Witcher you can choose a side to who you wish to be with, this is completely down to the player based on who they prefer.

# Levels

For the levels I plan to have more than 2, this will be in the risk assessment and if I can’t complete all the levels, I will have a minimum of 2. There will be two types of levels, one will be the levels where the enemies will spawn to build up skill points and health. This type of level will have constant waves of enemies until the boss’s rage meter is full, when this occurs the player will be able to go to the next level. The second type of level will be the boss level, this will be where only one enemy will spawn to start, the boss, at certain stages the boss will become invincible and enemies will spawn, when they are killed the boss fight will continue. If the player does not upgrade their skills, the player will find it difficult to defeat the boss. Once the boss is defeated the player can chose to start again to try and beat their best time.

Levels are important within a game as they can also help build a picture of previous events based on what the level looks like, for example a level that looks destroyed can relate to events in the games story. Another reason levels are important is because they allow for the game to all work together by adding a playing field for the player and AI to interact with each other, without a level then the game simply wouldn’t be able to exist as it is a main component.

# Gameplay

The gameplay of my game is going to be within the hack and slash genre. As the players spawns, they will be greeted by the war chief who will give an explanation. Next, the player will have to fight enemies that will spawn, for each enemy that is killed a percentage of the bar to summon the portal will fill, and when this is filled a portal will spawn that the player can use to go to the next level. Once the player has entered this portal they cannot return, and this will be the main boss fight of the game. The boss will use a series of attacks to try and kill the player so the player must use their abilities to dodge these attacks. Once the boss has been defeated the player will be displayed with their time which they can try to beat when playing again. The player will have the choice of abilities they wish to use including being able to block, heal, blink and throw projectiles as well as swinging a sword.

Gameplay is important within a game as it is how the player is playing the game and what they are doing during the game. If the gameplay was not very good, then no one would want the play the game thus losing the company money. Good gameplay would make the play feel in control of what they are doing and help to make the characters actions immersive, even if the gameplay involves something that is not real such as conjuration magic for example, the gameplay helps to give an idea what it would be like if it was real. Gameplay is a combination of all other aspects of the game as for example the story helps to build a basis of what the characters should do and what they need to be able to do.

When creating my game, I would need to introduce aspects of the game which would make it more competitive for example or more appealing for example. Within fantasy games today there is usually the option to allow for the player to be able to play how they want to play, an example is being able to pick whether you would like to play as a sword or magic wielder, this adds the element of immersion and realism by not limiting the player to how they can play.

Another element of gameplay is the challenge of the game, developers do this by setting a goal or objective by having win and lose states such as collecting all the coins and not dying. A game without a challenge such as the Spore Demo, can be fun but can also get boring and old fast, the reason for this is because there is no overall goal, in the demo of Spore you just create a creature and walk it round in a small circle, it can be fun to create so funny looking monsters but overall it can be very repetitive. By adding goals and objectives it gives you something to work towards such as completing the main story of a game, this makes you want to continue to play the game for hours until the main goal is complete. A technique used by developers now is by adding downloadable content to increase the number of goals after the main goal has been completed.

Another main element of a game is what I mentioned earlier about picking either a sword wielder or a mage, this element is choices. Choices will affect the gameplay and story itself so the player will be able to determine the overall outcome of the game making it feel like it is their own game world. Without choices in a game it would become an interactive movie where no matter what the player does the overall outcome will be the same.

When designing a game however the most important element is to fail faster, the idea of this is that no idea is perfect so even if the plan you have isn’t very good you can correct it each iteration until something good comes out of it. With feedback people can outline what’s good and what’s bad to try and eliminate the bad stuff. The faster and more often the developer fails the better the game will be as correction and learning will occur every iteration.

The quality of the game itself is also an aspect of development, good games will have a balance of everything needed in order to make it good, for example, games have rules, goals, choices, conflict and an uncertainty to what will happen next. All these play a big part to the player in order to keep them immersed, and to keep the enjoyment of the game.

Another main element of game development is how many players will be able to play in one session. This can vary from 1 to 4, maybe even up to 100+, different games depend on different amounts of players, for example an MMO would require there to be lots of players as this is what they are revolved around but a heavy story telling RPG would be single player as its idea and concept is to immerse the player in a world that is dependant on the choices they make.

Within my game I will have many different aspects and components. Firstly, my game will include a timer, this is not to rush the player through the game the firs time around but it is there to add a competitive side to the game so players can compare who will get the fastest time to complete the game. Another element I will include choices, this will be simple, and some may not notice it but the choice the player will be able to make is what weapon they wish to use, whether this be magic or a sword. The gameplay will also be difficult to add challenge and not make it too easy for the player to complete as this will lead to an unsuccessful and boring game.

Since my game is based around a story, I think it will be best suited to being a single player game where If I was to have a team working on it a whole world and progressive story could be built that will be guided by the players decisions. By failing often in my game, I am quickly realising what ideas are bad and what are good, this helps me to improve on areas where I and my peers think are good, this is required in order to make a successful game.

# Art

The art of the game will be based on free assets from the Epic Games Store. These come with some animations and meshes to allow for some customisation. Some other art such as some textures will be created using programs like Photoshop. As well as this I will be creating my own assets and sourcing assets that I find suitable for my game, these will be inspired by games such as The Witcher which is a role-playing game about hunting monsters. Below I shall include an image of an inspired art style.



<https://www.iamag.co/the-art-of-the-witcher-3-wild-hunt/>

This artwork is inspiring as it shows 3 different enemies but also 3 different play styles if this was a player character, the first being a character with more health than the others, the middle being an all-round sword wielder and the last being a magic wielder. This also inspired my game to be set in a medieval style era. Art is an important aspect of game development as it can help visualise what characters will look like and the story that has gone into them, as well as this it can also help to represent what a level will look like.

In game development artwork is used for the purpose previously mentioned but also to give the rest pf the team ideas and inspiration. Artwork is very important as it is not only visuals to the game but also what the consumer and players will see before buying the game so artwork will need to represent what the game will look like and play like. Artwork is also very quick as most of the time they are just drawings, this allows for an artist to express themselves and be able to show many different versions of the same idea so that the team can see which would best suit the game they are creating. Almost every game has pages of artwork that are never used but shows what the game could look like if this style of model for example was introduced.

# Sound

The sound will also be based from the asset store as the packs come with sound files. If The sound files included do not match what I want in a certain situation I will source from the internet to find free sound files and from my own recording to be able to create sounds such as breaking branches for example.

|  |  |
| --- | --- |
| Sound | Description of sound |
| Sound of player getting attacked (eg. Scream) | This sound will be played when the player is attacked, this sound could be anything but more commonly it is a sound of pain that the character will make to indicate that damage has been taken. |
| Sword swing | A sound of a sword swinging is the sound of splitting air quickly, this sound is not as important but adds realism into the game as in real life a sword swing would create a sound. |
| Sound of enemy getting attacked | This can be a combination of sounds, one being the sword connecting and then making a sound and the other being the enemy making a sound such as a scream to show damage has been taken, this is to show the player that the enemy has been hit and has taken damage, this also adds realism into the game. |
| Sound of fire burning | This sound can be played for destruction within a level or for spells the player can cast, this sound will sound like a fire crackling and will indicate the player that they are near/using fire. |
| Background music | There will be different types of background music, one for enemies and one for the boss, these will be to drown out silence and also motivate the player. In games today background sounds vary from where the player is and whether they are in combat or not. |
| Blink sound | This sound will be almost like a whoosh and will ply when the player blinks/teleports, this is an audio cue to let the player know that they have moved position. |
| Dodging | Dodging will cause an audio cue which will play a sound of a roll, this is similar to the blink sound as the player will see that it was been successful but will add realism and allow the player to use one of their other senses to sense what has happened. |

Sounds in games are just as important as graphics as they help to increase the realism of the game and increase the players immersion. A game can use sound effects to immerse the player by reflecting what is happening, for example if the player was outside then bird and nature sounds would play and if the player was in a cave the player would hear sounds such as water dropping and eerie sounds. Sound is also very important as it can show the player that something is behind them for example, if there was no sound the player wouldn’t know but with sound an enemy’s idle sound would play making the player aware that something is there. The implementation of music also helps to give emotion to the player and help the player to understand what is happening for example in a happy game scene happy music will play. As well as this the theme music to the game also helps players in recognising the game without even seeing it. A popular game that does this is Skyrim, most people can recognise the game just by hearing the sound and from here the person will get a visual of what the game looks like in their mind. By having bad implementation of sounds the overall gaming experience will be limited, and the player will not be aware of what’s happening. These are the reasons to why sound is so important in game development.

# User Interface (UI)

The user interface will involve a few different sections, one will be the health bar with text above showing the number of health left out of a set number. This is similar to the mana bar which also has a number next to it, this number being default to 200 and health being 500. Above this will be the abilities bar which will use progress bars to track the cooldown of abilities. I have planned to put 6 abilities, 5 being normal mana-based abilities and the 6th being an ultimate ability that costs rage which is built up by defeating enemies. This rage ability will slow time when a successful hit is made and will cause more damage. The final planned features are to add a timer to give a competitive view on how fast a player can complete the game and the boss bar which shows the progress of each level.

The user interface of the game isn’t as important as the other components within a game but is still important, the reason I say it isn’t as important is because some players enjoy a challenge and chose to turn the UI off in order to make it harder and seem more realistic. The user interface allows for the play to see their health and other important pieces of information such as what abilities they can currently use. Although this isn’t an important aspect of a game, it is still useful to have so players can chose whether they wish to use it or not.

# Controls

The controls of the character will be similar to other games. The W key will cause the player to walk forward, the S key will cause the player to walk backwards, A and D will cause the player to move left and right. The next controls will be for the abilities, Q and E will be for the first two abilities the player has and then the 1, 2 and 3 keys will be for the other 3 main abilities. The Z key will be for activating the ultimate ability when it is available. When the player presses the shift key the camera will change position and the player will run at a higher speed, once it is released the player will revert back to its original state. The player will also be able to use the F key to interact with objects such as talk to no hostile characters and The C key will be used to change between combat mode and normal mode, combat mode being where the player rotation is dependant on the mouse movement.

Controls are very important within a game as they allow for a player to interact with the world. Almost every game now allows for the player to customise the controls to their liking as everyone is different and some people may find it harder to hit specific keys. By allowing for customisation you are allows for a wider range of players to enjoy and feel comfortable playing the game. A newish controller for the Xbox called the Xbox Adaptive Controller allows for players who have a disability to customise the controller itself as some people may not be able to reach like others. This is intended for the disabled but can be used by anyone. This is very important as games can give people joy and this is the opportunity some people needed to be able to share what others feel from these games.

# Task 2

Within game development there are many fears about the development of the next generation of games as with increased hardware capabilities we can now create more expansive and more immersive games. As well as this other areas of game development are growing such as the size of the team working on games, back in the 1980s game development was a lot simpler as games were simple 2D pixel graphic games that a small team of people could create. Coming forward to today teams are a lot bigger, for example instead of just one company creating a game we now have the studios that create the game, as some have more than one, and the publisher of the game. These game studios and publishers can have hundreds of people working in them allowing for more diverse and immersive games to be created.

Like previously stated in the 1980s a game development studio could be very small and have very little resources devoted to it due to the limits on games. One team could create all the sound, art and graphics, an example of game is Pac-Man. Pac-Man is a simple 2D game that allows the player to collect dots whist avoiding the ghosts. In the 1980s Pac-Man was the popular game due to the limit’s studios had. Another game example is Pong, released in 1972, this game was even simpler with two lines only having the ability to move up and down and a dot which bounces from one side of the screen to the other until the player/line misses, then the opponent scores a point. This type of game has monaural sound which is a single channel of sound which is created by one speaker.

Today we have a wider variety of games, that are more cinematic and interactive, and a wider variety of tools allowing for us to create these games. Back in the 1980s characters were 2D sprites that changed the image to look as if the character was animated, now we use 3D modelled characters that have bones/rigged so that we can move them and animate them. As well as this, levels have also drastically changed, instead of being a 2D image with invisible walls to prevent the player going through them we now have fully 3D environments that have been sculpted. This allows for 3D houses, trees and walls that would all be individually modelled and places into a level. This gives a lot more control as an object can easily be moved if needed. Something else that has changed is the gameplay aspect itself, before story telling was a simple dialog box that would appear to show a character talking to the player, now we have cinematic cutscenes that can either be from a third or first person perspective showing characters in the environment talking face to face with the player making it feel more immersive. Unlike the 1980s, teams today have many areas focusing on different parts of a game. For example, a studio would be split into different teams including: Artists that would use a different set of software such as Photoshop to create the art work for the game, or some artists create the game world so in this case they would use the game engine. The advantage of using separate software is that it allows for a person/team to be more creative in what they want to do. An example of this is in the Borderlands series created in Unreal. Borderlands has a unique comic style shader that allows for bold black outlines of objects which help the payer to better understand features, previously the developers have had trouble due to the limits on the engine so these black outlines would scale in size with the object as that is scaled. More recently, in the Borderlands 3 development video from 2016 the developers explain how the new technology in unreal allow for the shader to outline an objects features. Another area would be the animators who would use software such as Blender to rig and animate characters, blender would also be used by the 3D modelers to create the models for buildings, enemies and characters. Another area is programmers who would use tools such as Visual Studio to program the features of the game. All these areas would have multiple people specialised in their own way to create a more diverse and immersive game.

Traditional game development methodologies such as the waterfall model spend a lot of time in the front0end of a game to define the functionality and important elements such as the mechanics and levels. The waterfall model is very similar to an assembly lines as everything that is put onto the assembly line travels down having things added to it until the end of the line, but whatever is placed on the assembly line cannot go back so developers will never be able to truly have a feel for their game to see if the mechanics were right.

A newer model which is used is referred to as the Agile Model allows for multiple demonstratable iterations that can go into production. This method also helps the team to organise the game as it allows for the game to be split into slices to allow for one part of the game to be created first, for example the first iteration could simply just be the main mechanics of the game. The problems that a game development studio has can be varied as the different teams, such as programmer, art and sound designers will all have their own problems. With the new technology we have game development for games can be a very long time, short games can be anywhere from 2 years in development and longer games can be around 5 years in development due to the problems a team may face and the content the team puts into the game.

Another model that can be used today is the spiral model. The spiral model however takes more time as it prevents risks by analysing how to prevent them. This takes more time to create a game so when compared to agile after each iteration a bug fix can be implemented to fix any problems the game may have.

# Methodologies

## Waterfall

The waterfall model is an approach that moves in one direction and cannot go back, this would be from the project’s requirements to the actual design and implementation of the game. With the method there is very little evaluation on what has been done meaning any changes cannot easily be reverted.

In game development with the waterfall model, a team will create a design document for the game which is where all the mechanics and features will be outlines. From here the document is split into chunks which is where developers extract the required assets, mechanics and functionality of the game. This also where the number of teams involved is decided.

From here the waterfall method can begin as the requirements pass down through animation, programming, character art, level art, sound and effects. The waterfall method can continue if the previous team is done with their feature, from here it can be passed onto another team. If we take a character for example, it would begin with the design document and passed along to the director of the game, from here it would be broken into the components of the character such as its animations, mesh, texture and mechanics. Each team will focus on a component and then work till something like the requirement has been created.

Next it is passed back to the director to see if what the team has created matches the design document and then is passed along to testing, from there it is passed back to different departments for bug fixing. With this methodology the mechanics of a game are created from the ground up over a period of time.

Advantages of the waterfall model include: using a clear structure, when comparing to the other methodologies the waterfall model focuses on a clear and defined set of steps. These being, outlining the requirements and creating the necessary documents, designing the game, for example the characters, levels, story. Then the implementation of what’s been designed, this is where the characters and levels are created, after this testing is done to spot any bugs before the product is released. Next is delivery of the finished game to the market and then finally maintenance is done to the game. The teams must complete the previous step before moving on so if any problems occur in the step the team will know straight away. Another advantage is that the end goals of the project are outlined early so there is less chance to get lost in the details. If we compare this to SCRUM which divides the project into sprints, the waterfall model keeps the team focused on the end goal all the time. Another advantage is that information is transferred well between each step, as each step contains a team of different people all the information from the previous step is documented so that the new team can catch up to what’s happening very fast. Back in the 1980s this would be the best model to use as there was no downloadable content or constant updates to games so what was delivered would be what was staying.

Disadvantages of the waterfall model include: it is difficult to make changes to the previous step as the steps are always moving forward. Traditionally with this method there is no room for unexpected changes. Another disadvantage is that it excludes the client as this method is an internal process where the team will create what has been outlined on the requirements. This would be very bad now for game publishers as the market is constantly changing so games may change half way through development to meet the user’s requirements, this would especially apply for games that span over 5 years of development. Another disadvantage is that testing is delayed until after the project is completed meaning any game breaking errors such as the core mechanics which everything else relies on could essentially cause the team to have to start development again.

## Agile

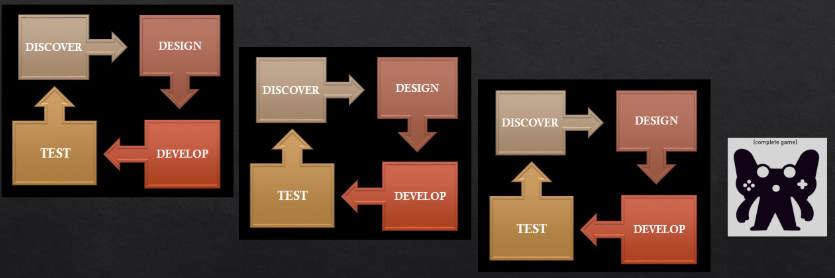
Agile is another method when developing games, in this method we use short iterations. The main idea of Agile is to not make the whole game, unlike waterfall where its build from the ground up, but instead create small features of the game in sprints (short period of time) so that whatever results are created can be used to alter the design document and plan of the game.

With Agile each iteration is a small project by using inspect and adapt practices to be able to adjust the goals of the project and be able to measure progress. In figure 1 there are four phases to each iteration, these being discover, design, test, develop. These can vary bases on who is using the model but they all come down to the same idea, that being: Initiation, where the game ideas are brought up, the feasibility of the game is discussed to see if it can actually be done, the style of the game using focus groups to see what is currently popular. Pre-production, where the designing of the game takes place, planning of what will happen in production, story ideas so the team will have an idea of what levels, or characters will need to look like or act like and what the objective/goal of the game. Production, where the first level is built for example or the character, essentially a part of the game in isolation to be able to test the features and mechanics. Testing and Reviewing, this is where the part of the game that has been developed is tested to see if it works as intended and for the developers to get a feel for the character.

If something is wrong, for example it isn’t competitive enough or action packed enough then the next iteration can fix the issue. This part of the game is the Alpha stage where anything is subject to change at any time based on what the developers think best. Once the developers are happy with what they have created the game will go into a stage called closed beta, this is where the game is released to specific groups of people to test. The implementation of a closed beta has many reasons, but some include being able to help to prevent leaks at that stage and limit the amount of people that can play.

After this stage the developers will release the game into open beta where anyone can play the game in its current state, the reason for this is so that the game can be released to players that have an unbiased opinion of certain aspects of the game that the developers may have. As well as this the open beta stage helps to spot any bugs as with popular games millions of people will be playing the open beta. If a team of 100 develop a game and 1,000,000 play the beta this gives a 10000 time more chance to find bugs. As well as this the open beta allows for the developers to see how the game will perform on different hardware and systems allowing for them to optimise it more, as with the developers, they will have very powerful systems needed to be able to create these games.

After the beta the game can go back into production to fix any bugs and optimise the game. The idea of a beta is to gain feedback and improve the game. Iterative models such as the agile model allow for a lot more creativity when compared to the waterfall model as each iteration that the agile model goes through the developers can add and remove objects and features that they see fit. Examples could be after gaining feedback from players they could find out that a level in their game isn’t competitive enough. This gives the developers chance to change the level based on what they see fit for the game thus giving them total creativity each iteration.

Figure 1

In traditional planning a game developer would go though the four stages listed above and then follow the original plan going step by step. But the problem with this is that no one may play the game as the plans weren’t based on what the players want so the developers would be forced to change to change the plans. With agile planning the developers get a small set of goals set over short periods of time or sprints and forces the developers to go through the 4 step process each iteration. At the end of each iteration the developers can step back and conclude the results. By using these results the initial plan can change, and new goals can be created. This occurs over and over until the deadline is achieved.

Agile development is best suited when the size of the project is going to be big as with small iterations/sprints a small team is required as that sprint will be small in size. Some advantages of the agile model include: The delivery of software is constant, adding new updates/features to keep on top of the players needs. The players are going to be satisfied as each sprint there is a working version of the game. Another advantage is that the players can give feedback to change the next sprint of development. The main advantage however is that changes in the requirements are accepted in the later stages of development meaning that the team will be able to change even the core mechanics of the game easily. This model is more suited to software and game development as most new games release with room to add more later down the roadmap of the game.

Some disadvantages of the agile method are that: documentation is less so the next team will have less of an idea about what has been done previously, as well as this the requirements are not always clear, so the overall result is unclear. Another disadvantage is that some risks may appear as agile is unlike the spiral model which outlines all the risks in order to prevent them, the only problem with that is that it takes longer.

Now to fully compare agile and waterfall. The points I am about to list will be short with more detailed comparisons after that. The waterfall model is sequential where there are defined requirements and being able to deliver high quality products in a rigid process. Agile on the other hand is flexible that allows for change, this model also delivers high quality products and is constantly evolving. The waterfall model is a structured process where the next step can’t be started until the previous step is completed whilst with agile it is flexible so you can move through the steps as you like. With the waterfall model being sequential this means it is linear whilst with agile it does not enforce a linear process of development. The Waterfall model also has defined requirements before the project is started meaning it is hard to change them when development has begun but with Agile the requirements can change half way through development which won’t affect the whole development process. Like this the waterfall model doesn’t allow for change so once something has been done then you can’t go back to change it whilst with Agile development you can change aspects of the game in future iterations.

The Waterfall model should be used if changes in the project scope aren’t expected to change, and the project is simple to do. As well as this the requirements are well known so the customer will know what to expect before development is started. Agile on the other hand should be used if the final project scope isn’t clear and rapid deployment of software is required. The reasons to use agile is similar to the points listed below when comparing agile to SCRUM.

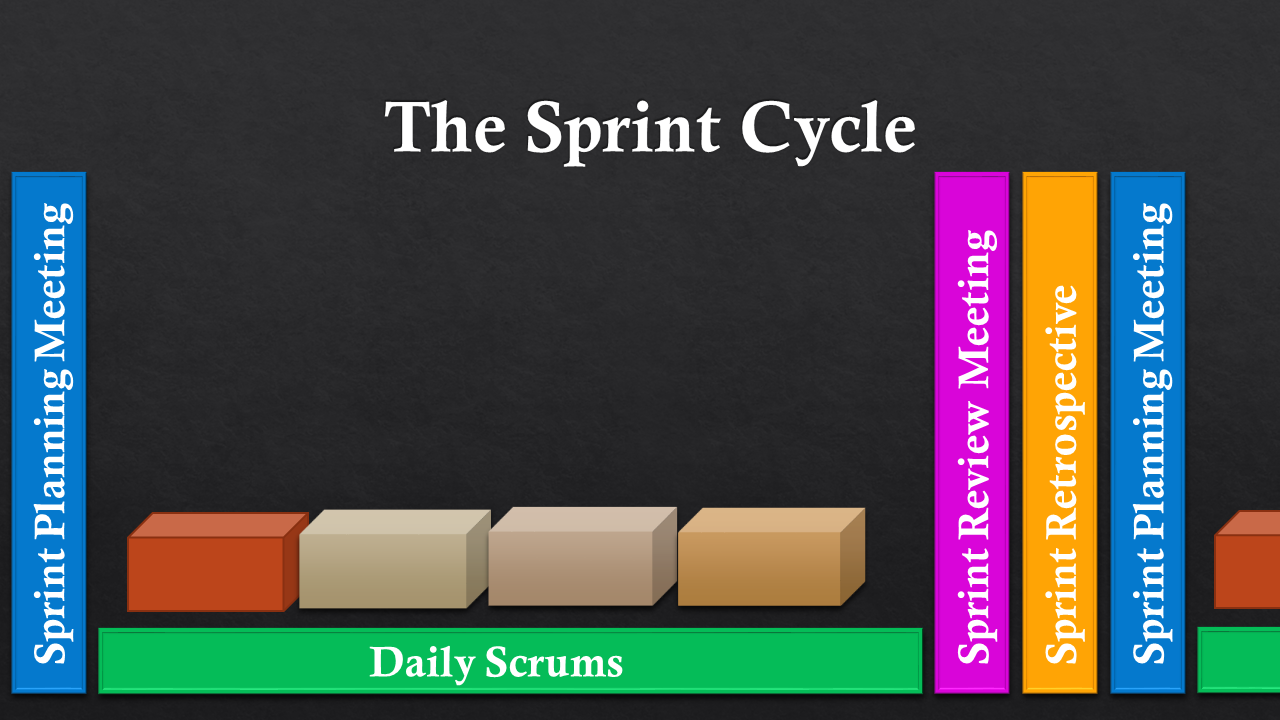
## Scrum

SCRUM is a form of the agile model, with the agile model it sits at the time as principles that need to be followed but with scrum it is a concrete implementation under agile. SCRUM produces software that is demonstratable every 2-4 weeks and based on previous projects that have been created using SCRUM it is very successful. A product backlog is another part which is used in SCRUM, it contains many features, but with the SCRUM process it helps to create some goals in the sprints of an iteration. Once the sprint goal is defined then a team can undergo several tasks to be able to achieve the goal, after the sprint the developers can release a game that is slowly improving with content. This allows for feedback each iteration so bugs and problems can be fixed but as well as this the feedback given can help outline what the users want and then implement it to be able to make the game successful.

There are 3 roles within SCRUM, these being:

* The product owner, who is responsible for profit, prioritise features, can change features, and can accept or reject what has been done.
* SCRUM master, who ensures the team is functional, ensures that each process is followed and participates in meetings and reviews.
* The development team, who are responsible for selecting the sprint backlog, demoing the results to the product owner.

In SCRUM there will be meetings to outline priorities and commit teams to certain areas, as well as this there will be daily SCRUMs which will outline what has been done yesterday, what will be done today and what are the problems. Then after this a meeting will be held that anyone can attend to review what was accomplished by demonstrating it.



SCRUM is best used when the success of the game is based on what the player requests, such as any reworks, as well as this SCRUM is best used if requirements change frequently so the team won’t know the overall goal. Some advantages of SCRUM include: ensuring the best and most efficient use of time and money, this is because SCRUM allows for the ability to manage tasks in an organized way and have the necessary planning needed to be able to reach these tasks. Another advantage is that large projects are split into small sprints making them easier to manage, this allows for a team to work on a specific part of the game and make sure that that part is functional and up to expectation meaning that the team will have the time to be able to create their character for example exactly how it was in concept. Another advantage is that each sprint is tested to make sure it is working correctly allowing for easier implementation of more future features, this relates to the advantage above as it allows for the team to ensure that they have a working part of the game that is up to expectation, without testing regularly then the game would encounter errors and problems when new features are implemented. SCRUM is also good for fast moving projects that will constantly change as especially with game development the project will change based on what consumers want and whether the planned feature will actually fit with the game so far, SCRUM allows for the ability to easily change features as you can go back and alter the character for example as its own iteration. Being able to adapt to feedback given is another advantage as like stated in the previous advantage the character for example can be altered in its own iteration, this means that when feedback has been gained on a character from the player base, it can be changed based on what the developers think is healthy for the game. For example, a character could have too much health compared to enemies. Finally, SCRUM allows for the team to get a clear vision of what’s happening through meetings, this allows for the whole development team, this being made up of an art team, sound team, programmers, and other areas of game development to be clear on what other areas of the game are working on. This helps the art team for example to design art based on what other areas of the development team are working on.

There are some disadvantages of using SCRUM however, for example: A game design document is no longer needed as a backlog spreadsheet replaces it, this may not be considered a disadvantage to some but it does cut back on actual documentation of the game which can be used to guide other areas of the team when they have to create a part of the game which will need to be related. Meetings will interrupt peoples work as they are frequent to evaluate what’s been done, within SCRUM meetings are common to see what all areas are working on but the problem with them being often is that it can interrupt someone’s work, if the developers were like me, this being having an idea and having to write it down, then being pulled away from this idea can cause the developer to forget or lose the clear picture of what was planned. Another disadvantage of SCRUM is being able to adopt SCRUM is challenging for a team, and some developers make it mandatory, SCRUM is slightly different to the traditional agile model so adapting to it can be difficult if the team is used to using a different method. If someone leaves it can have an impact on the others as a small team usually dedicate their own roles to complete the sprint and finally quality is hard to implement until testing has been done. Defining deadlines is also different as work is shifting all the time, this can make it hard to pin point when something will be finished. Finally, developers can get bored easily as each iteration will be based around the same principles of planning, developing and testing. The SCRUM master will have the role of trying to keep the team engaged to prevent the downfall of work ethic.

Now, both SCRUM and agile follow the same system with some slight differences. Agile “describes a set of principles in the Agile Manifesto for building software through iterative development” (smartsheet.com 29/04/2019) and SCRUM is a set of rules which will be followed when practicing software development with Agile. Agile is the philosophy of development and SCRUM is the methodology. SCRUM is an implementation of Agile so both share similarities such as being an iterative model and releasing bits of software often. SCRUM would be best suited if the requirements will change over time, if feedback will be required, if there is no commitment to a fixed deadline and software will have to be delivered regularity. Agile on the other hand would be best suited if the final project is not defined, if the clients will be able to change the scope of the project, of changes need to be made during the entire process and if optimization is required for fast development.

# The Model I am Using

For my game I will be using the Agile Model, this is because I am focusing on one feature, such as the character, and creating a working version of it before moving onto something else, this allows for me to reduce the amount of errors in other parts of the game related to the character. The reason I am using agile over SCRUM is because SCRUM is more suited to teams as they will need to have regular meetings that will help update the rest of the team which will help them to create parts of the game related to what has already been done, as well as this SCRUM doesn’t use a game design document, instead it uses a backlog which will list what needs doing, this can be a downfall as more documentation gives a basic guideline to follow, from there the developers can steer off the line to change the game as they see fit. The downside to using agile when compared to waterfall is similar to agile and SCRUM, and that is that there isn’t as much documentation to be able to follow and use if a presentation was required but the upside to agile is that it gives me the opportunity to easily change features in the next iteration if they no longer suit the rest of the game as it changes through development, this is a huge upside to iterative models such as agile because I can change anything about the game in the next iteration thus eliminating errors and improving the overall game. Within the project the presentation is an example of part of an iterative model, once I present the presentation to my peers, I will gain feedback on what I have already done and what I could improve on. This is similar to real projects where feedback gained from the current build can be implemented in the next cycle, so after this feedback I can implement new/improved features in the next iteration. Although agile is suited to big projects, with my game it is essentially a prototype, or a part of the game that is playable, meaning that if it was for a real game each iteration more and more would be added over 3 years of development for example. This also allows for my risk assessment to be up to date as I will be constantly altering it depending on any risks that I may come across over the cycle of development.

The reasons I chose the agile model are, the ability to make quick decisions, if I think that part of my game isn’t working as it should or this is what the feedback states, I can quickly decide whether to fix this issue quickly or after other features have been completed. Also change is going to play a huge part within my game as some features I like now may not be fitting in the future so by using Agile I have gained the ability to easily change features based on what I and my peers see fit. Another reason is that I can quickly create part of a game to represent what the full game would look like and then add to it from there, if I was to use the waterfall model, I couldn’t be able to do this and instead would have to create the full game. Another reason I chose this model is based on the reason above as I can ensure that the section of the game I will create will be at a high level of quality, instead of being rushed to get the whole game complete. Agile also helps me to reduce risks as each sprint will be created and tested to work with the previous sprint, this helps to eliminate complete project failure. In a real development environment Agile would also give me the tools to be able to complete a full game from my concept. With this assignment being only part of what I would like to create I cannot implement all the features I would like but with agile I can slowly introduce new features even after the projects initial release date. By using the agile model, I can easily adapt to any problems that may occur down the line of development, this is especially handy as I shouldn’t meet any game breaking error that may cause me to have to start again, instead I can just change what is broken via regular testing.

# Conclusion

In conclusion, this is my game design document outlining all the features within my game such as the story, characters, art, sound, UI and controls which gives me a bases to work on when creating the game. It also outlines different methodologies, this being agile, waterfall and SCRUM, the advantages of them, the disadvantages of them and the situation when each would be best suited in. Finally, after I outlined the methodologies I have explained and justified which method I will use for my game references the assignment brief as an iterative model.

# References/Bibliography

Cprime. (2019). When to Use Scrum? - Cprime. [online] Available at: https://www.cprime.com/2012/09/when-to-use-scrum/ [Accessed 8 Apr. 2019].

Gamasutra.com. (2019). Paper Burns: Game Design With Agile Methodologies. [online] Available at: http://www.gamasutra.com/view/feature/131151/paper\_burns\_game\_design\_with\_.php [Accessed 7 Apr. 2019].

Gamasutra.com. (2019). Top 10 Pitfalls Using Scrum Methodology for Video Game Development. [online] Available at: http://www.gamasutra.com/view/feature/132121/top\_10\_pitfalls\_using\_scrum\_.php?page=1 [Accessed 8 Apr. 2019].

Lucidchart.com. (2019). The Pros and Cons of Waterfall Methodology. [online] Available at: https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology [Accessed 7 Apr. 2019].

Marionette Studio. (2019). Agile Game Development - A Quick Overview - Marionette Studio. [online] Available at: https://marionettestudio.com/agile-game-development-quick-overview/ [Accessed 7 Apr. 2019].

Simplilearn.com. (2019). Scrum Project Management Pros and Cons | CSM Training. [online] Available at: https://www.simplilearn.com/scrum-project-management-article [Accessed 8 Apr. 2019].

Smallbusiness.chron.com. (2019). Advantages and Disadvantages of the Scrum Project Management Methodology. [online] Available at: https://smallbusiness.chron.com/advantages-disadvantages-scrum-project-management-methodology-36099.html [Accessed 8 Apr. 2019].

Tryqa.com. (2019). What is Agile methodology? Examples, when to use, advantages & disadvantages. [online] Available at: http://tryqa.com/what-is-agile-methodology-examples-when-to-use-it-advantages-and-disadvantages/ [Accessed 7 Apr. 2019].