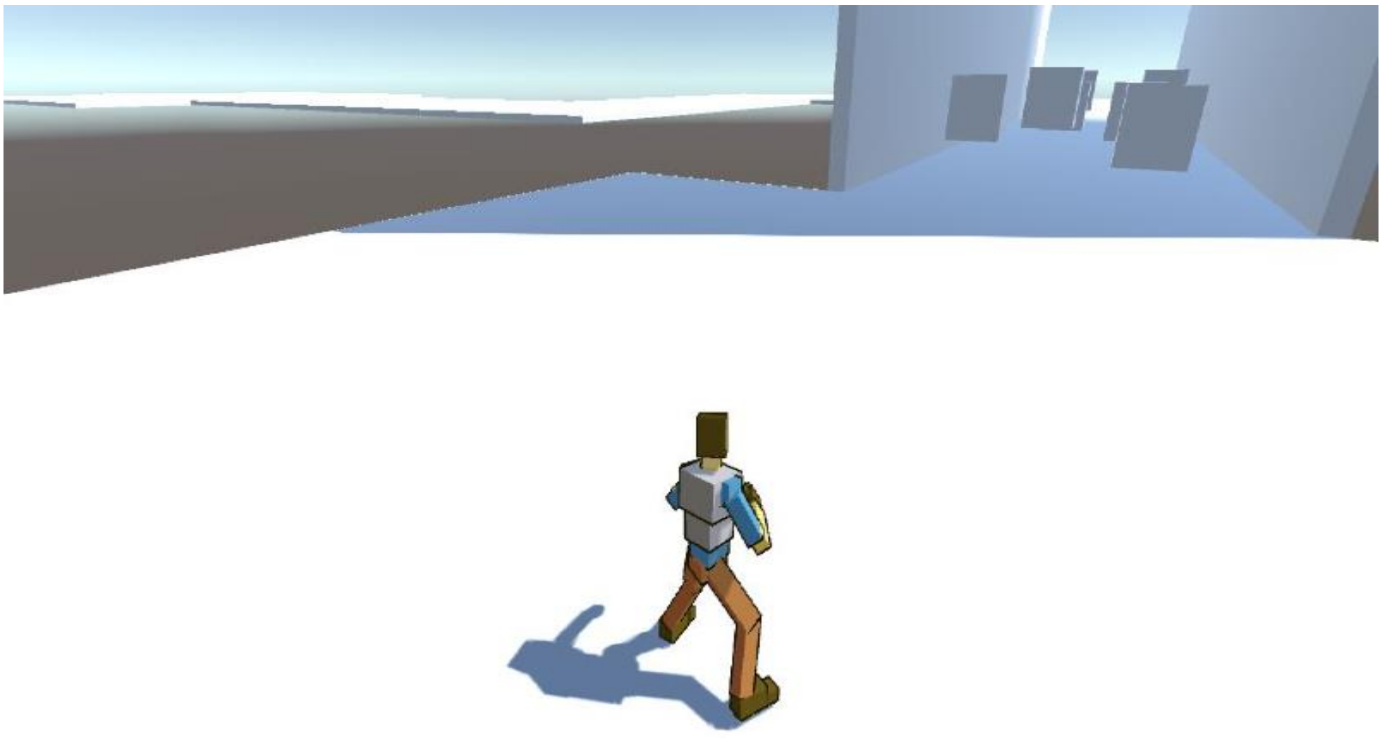
Creating A Robust And Dynamic Camera System.

Ryan Manning
Ryan2.manning@live.uwe.ac.uk
Supervisor: James Huxtable

Department of Computer Science and Creative Technology
University of the West of England
Coldharbour Lane
Bristol BS16 1QY



Abstract

This project is to create a camera system that can transisiton into diferent states such as first person, third person, top down and on rails. As well as this the camera has a cutscene system that allows a user to create points and the camera will move between them. This system is for developers so that they don't need to create a scripted camera in game and can start using this one.

Keywords: Camera, Lerp, Cutscene

Brief biography

I am Ryan Manning I chose this project as I believe that cameras can make games great and if a poor camera is used in a game it can ruin the players immersion. This project will give me the skills to be able to understand what is needed in making a good game level as I was insipred by the god of war camera and how well made that was. Here is a link to my website. https://ryanmanning3.wixsite.com/gamesdev

To access the project please go to this link - https://github.com/RyanManning24/CCTPCameraProject

1. Introduction

The project explores a camera in games and different camera types. The main goal of the project was to create a camera script system that would allow a developer to have access to multiple different camera behaviors with the ability to switch to them at any time. As well as this the camera would have collision and detect when the camera cannot see the target and will move so that the target is in view. The project is important as it will simplify development as creating a camera for games will not be necessary as it will already be completed. Also, it will allow projects that use the camera to decide which camera types they need and to use and test them straight away in a scene. This project arose by looking at different cameras in games and seeing how much of an impact it had on the game and the gameplay. The main inspiration for this was the camera that was in God Of War(Sony Interactive Entertainment,2018). As this camera was implemented well and allowed for the gameplay to shine through.

My main deliverables for this project were:

- Create a third person camera
- Create a first person camera
- Create a top down camera
- Create a on rails camera
- Create a system for cutscenes
- A collision system for the camera

My main objectives for this system were:

- Too be as easy as possible to use for the player and developer.
- How does the camera handle small spaces?
- How well does the camera handle transition into different camera states.

These objectives will be used to test my level against so that I can iterate on the system so that by the end of the project the system is as good as it can be. The project used sprints and milestones so that the system was always able to constantly keep evolving and being iterated on.

2. Literature review

I started by researching previous games cameras. One of the main cameras that this project was inspired by was the camera from God Of War(Sony Interactive Entertainment, 2018). The camera in this game is one of the better third person cameras in games. As this camera has very good collision detection which allows the player to be able to fight and play the game no matter what environment they are put in. Also, the camera transitions seamlessly through cutscenes this allows the game to flow and to highlight important parts of the games story design. This camera should be set as an example as to what a good camera system is. For my project, this camera is the ideal third person camera.

As a base for my project to be based off the next camera that was looked at was Super Mario 64(Ninetendo,1996) camera. This camera for the time the game came out was revolutionary. This was because most games before its time only had a fixed viewpoint for the player. This was one of the first games to really give the player the freedom of camera movement. However, this does not mean that the camera system was any good as compared any 3D platformer the camera is more smooth and easier to control. This camera is a good one to use as it shows the start of 3D platforming cameras and gives a good base to compare the newer cameras of. The reason for looking at this camera was to see the comparison in cameras over time and to see what was needed in creating a basic follow camera that worked so well.

Next, I looked at some books and papers to find what they believed a perfect camera looked like. In his book Rogers ,S(2014) he states some requirements and challenges when creating different camera types. Focusing on the first-person camera and the third person camera as these are the two cameras important camera types to my project. Firstly, the first-person camera advantages are how it allows the player to be able to aim easily. This was considered while creating this camera type as otherwise the camera type will be poor as it cannot even do what it was intended to do. Also, a first-person camera will allow the player to be able to see the gun or world objects easier. Something to consider for my system is a field of view slider for the designer as this is something that can change the feel of a game and changes what the player sees. Next the third person camera Rogers ,S(2014) states that it's important for a third person camera has collision as to not disrupt gameplay and to lose the players immersion in the game. These points that were outlined from Rogers, S(2014) helped me in creating my camera system.

In his book Haigh-Hutchinson(2009) outlines some of the design principles that are involved in creating a good camera in games. These are something that I considered whilst creating my system. As well as this Haigh-Hutchinson(2009) also goes into detail in and compares the differences between real world cameras and game cameras which gives some perspective as to how to position the camera in game. This book was extremely useful in trying to pin down what a perfect camera looks for as it showed some real insight into how to make a camera in a game.

One of the main camera types that was going to require the most attention was the on rails camera system. Therefore, finding a good camera that already exists was important as it provided a good base to base my system off. For this I chose the camera in Pokémon Sword and Shield(Nintendo, 2019). In Pokémon Sword and Shield (Nintendo, 2019) the camera is mostly locked to a specific path. For my system this needs to be an option for the designer to toggle camera control like Pokémon Sword and Shield (Nintendo, 2019) Also it needs to allow the designer to set these positions manually. This should then look like the screenshot from figure 1. As through the game you do not even need to think about positioning the camera as everything u need to see is already in view.

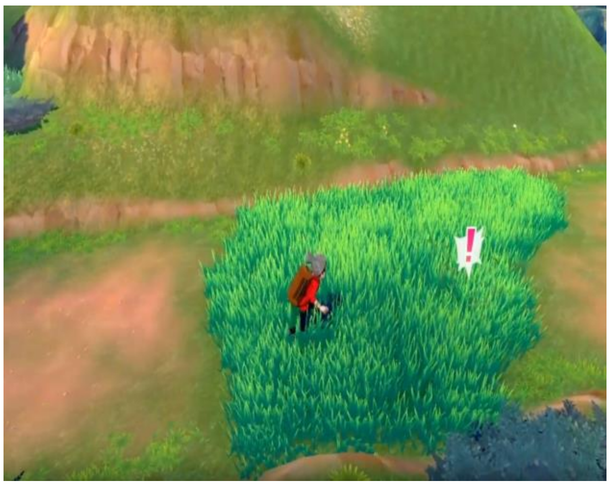


Fig 1- Pokémon Sword and Shield On Rails Camera

Another camera state that was going to be difficult to implement was adding cutscenes. To research this I looked back at one of the first games to implement a cutscene in a game which was Half Life (Valve Corportaion,1998). The cutscenes starts at the start of the game where the player starts off on a moving platform that takes the player though and starts to introduce them into the game's universe. This was considered genius at the time the game was released as cutscenes like this did not exists. This was something to keep in mind as this cutscene did not use any camera movement or lock the player off and it still had a lasting effect on many players. One of the main questions that was raised from this was how to make a cutscene system have the impact that this does.

Another example of a good cutscene system in a game is the one that is used in God Of War(Sony Interactive Entertainment,2018). What makes this games cutscenes great is that it transitions perfectly into actual gameplay. This is so good because it allows the game to not break the immersion at all meaning that the player will be fully immersed through the whole game. This is something that could be done in my project. This would be a goal as this would be hard to achieve it. However, a simple implementation of this could work for my project.

Finally, I found a previous project Schramm, J(2013) in which the project investigates third person cameras and their features. This report details cameras features such camera shake, bloom, FOV (Field Of View) and lens flare. These features would be something to look to add features for in the future. This would upgrade my system as it would give the developers more ways to use and create with my system.

3. Research questions 250w

Some of the research that was conducted allowed me to form some questions based on the research such as whether a system with multiple cameras states would work well? The camera system could be hard to use and therefore this system would not be useful to a developer and would only hinder them. If this is not the case, then the next question of the system would be doing each individual camera system work and allow a developer to use them well? This question is raised as so the systems usefulness can be evaluated.

As well as this another question that arose was how to make extend this system past what was initially intended? One way this could be answered would be to add in more different camera types as this would then give the system more variety. Also, another way this could be achieved was by adding more unique behaviours.

Another research question that was found was will the cutscene camera be able to transition between gameplay and cutscene well? This question is based off of the research into the God Of War(Sony Interactive Entertainment,2018) cutscene where the transition between cutscene and gameplay is close and its hard to tell the difference between them.

A final research question is will it be possible to make a cutscene system that can give the player an impactful experience? As stated in the research question this was raised due to the impact that the cutscenes had in Half Life(Valve Corporation, 1998). This question is hard to answer without either releasing a game from this or having this system user tested by a lot of people which is not feasible.

4. Research methods

One of the main research methods that I used was primary research. This was used to collect research about different cameras in games. Some of the games that were researched using primary research was Pokémon Sword and Shield(Ninetendo, 2019), Apex Legends(Respawn Entertainment, 2019) and Horizon Zero Dawn(Guerrilla Games, 2017) as well as some other games. This was used as the main bulk of my research as this was used to gather information on camera types that were used in the professional fields. This was useful as it allowed me to build up examples of camera types so that once my system was built I could cross reference these cameras across and then use them to pick holes in my system.

As well as this I used secondary research. This was used to find out and build up criteria that makes a perfect camera system. These criteria were then used to check against my system so that I could then improve my system. Some of the research that used secondary research was Rogers ,S(2014) book, Schramm, J(2013) project and Haigh-Hutchinson, M, (2009) book. These all allowed me to learn more about cameras in a real-world sense and move that knowledge into the virtual world.

Also, user research was not used in this project as I felt that with the research, I had done on different cameras in a professional game that this would not give me any new results that would not be found using the research that was outlined previously. As well as this the project is intended for players and developers therefore there would need to give the developers enough time to use and understand the system to fully be able to get useful feedback. The players that would have tested the system would of tested the base camera and would have mainly found bugs of the different camera types. However, looking back now this may have provided results that varied from my own as I could be biased towards the project.

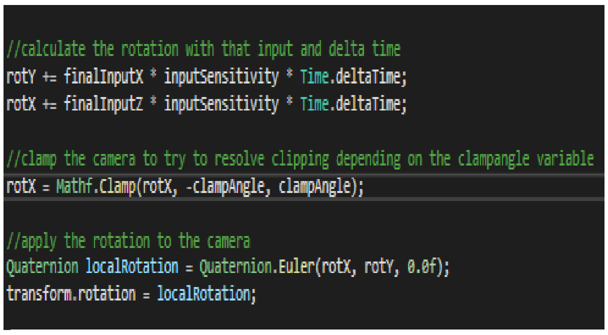
5. Ethical and professional principles

For this I will not be using human participants as I am using my research findings as a guide to base the camera types of.

6. Research findings

The importance of my research has allowed me to identify different camera examples that can be used against my camera types. This will be used to judge how good my camera is and will allow me to iterate and improve the cameras. As well as this it has given some insight in the requirements that go into creating a good camera. Also, in Rogers ,S(2014) book he outlines some features that make a good third person camera such as collision as to not break the immersion of the player. As well as this the book Rogers ,S(2014) also states features that makes a good first person camera as I outlined in the review. This allows me to test these features in my system as to make sure that they work and do not make the camera types worse.

As well as this the camera features given in Haigh-Hutchinson, M, (2009) project is something that could be done to improve my system after all the basics are completed. This could be adding something like a way to adjust field of view on camera types such as the first-person camera.



```
//calculate the rotation with that input and delta time
rotY += finalInputX * inputSensitivity * Time.deltaTime;
rotX += finalInputZ * inputSensitivity * Time.deltaTime;

//clamp the camera to try to resolve clipping depending on the clampangle variable
rotX = Mathf.Clamp(rotX, -clampAngle, clampAngle);

//apply the rotation to the camera
Quaternion localRotation = Quaternion.Euler(rotX, rotY, 0.0f);
transform.rotation = localRotation;
```

Fig 2- Code used to calculate the rotation of the third person camera.

7. Practice

For this tool to come to fruition the base camera script needed to be created along with the collision. First was creating a third person follow camera script that could be easily adjusted by a designer. This was done by setting up input axis for the mouse to use so that the camera will be able to follow the mouses movements. From this data I could the calculate the rotation that the camera needed to rotate based on the input. Next was clamping the camera so that I could minimize the risk of the camera clipping through the different terrain during gameplay. This code can be seen in figure 2. Next the collision of the camera was to first decide which was the desired position that the camera wanted to go. Once this was determined I can then line cast to check to see if the camera is going to collide with a game object. If this returns true, then clamp the distance value to then better be able to move the camera out of the way of the game object. Otherwise, it sets the distance to the max distance as there is nothing in the way of the camera, so it is free to move as intended.

Due to how well the third person camera script behaves, this script ended up being the base script that all the other pre-set positions use apart from the cutscene and the on rails camera. This is because all the other camera pre-sets such as first person and top down all work in the same way and the script I have created allows these cameras to behave exactly as they should.

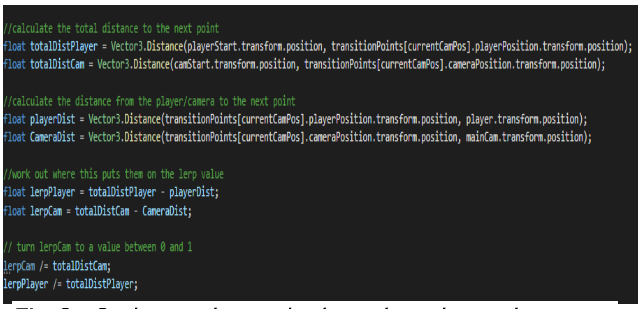
Therefore, to make using the system as easy as possible to use, (Unity Lerping Documentation, 2021) Lerps was used along with empty game objects meaning that all anyone must do is move the game object in the scene and then set the transitions to have the camera positioned anywhere they want. Setting up different transitions into the different states is done through a script that manages the state of the camera. This is done through different Booleans that dictate which state the camera is in. For the camera to transition a cam must be selected such as third person or first person and then the transition bool must be check. This was added so that designers could dictate when they would like a transition to occur rather than making them code this in.

This felt like a good point in the project to review the code I have already written and adjust some of the numbers in certain scripts. The focus of this ended up being the base follow camera script and the collision script. For the cutscene camera and some of the other scripts mouse movement and rotating the camera is not needed. Therefore, the follow script got split up into functions that only ran based off what camera was being used at that point in time. As for the collision script this was mainly reviewing if any position I either broke the script or caused undefined behaviour. One situation that was found was that if the camera were in a small space the camera would clip through the wall as it was trying to maintain the distance away from the player. This was fixed by just adjusting some of the values to do with the clamping of the distance.

After this the next camera state that needed to be implemented was the top down camera. For this camera type to work the camera needed to be positioned far away from the player and look down at the player or a target. This was done through moving the camera to the respective game objects position. Then using the Look At function to focus the camera on the appropriate target. This was simple to implement however adjusting the base camera position to make sure it was in good position. As after it was first positioned it was hard to see where the player was moving but moving the camera back a bit and higher up fixed this issue and now its easier to see where the player is moving.

The next focus was the implementing of the cutscene. This needed to allow for as many points as a designer wanted and be consistent in how it moved between each point. The start of this was understanding the lerp function in Unity(Unity Lerping Documentation, 2021)and how to manipulate this overtime as previously the lerp was being set to its final position to transition into the correct camera position. There is a boolean in the cutscene script that starts the cutscene this is added as it again allows delays in when someone wants to activate the cutscene. This was added as so if someone wanted to lock the player before a specific cutscene then this can be done and then play the cutscene when it is appropriate to do so. Something that was considered for this system was adding smoothing to the start and the end of the lerp for the cutscene. This is because the lerp function naturally adds speed at the start and end of movements however this addition was not a big priority as the camera can still work without this feature. On this website (Lerp Like A Pro, 2014) there was a post that goes through and tries to fix this problem.

As well as this the cutscene camera needed something to allow a developer to slow down the cutscene and apply different rates of speed to each cutscene. This was an easy addition to the script however was a very necessary one as otherwise the cutscene would only run at one speed which would ruin cutscenes that is supposed to focus on the background of a level. Also, another addition that was add was the ability to repeat cutscenes this then allows for multiple cutscenes to occur once per level which is vital in any game. A problem that occurred during creating this was after ever point was done lerping the camera would reset to the position it originally started in and then carry on to the next point. This issue was being caused due to one of the if statements that I was using to try to check the percentage that the camera was through the lerp. Rewriting this was enough to fix it and carry on with implementation.



```
//calculate the total distance to the next point
float totalDistPlayer = Vector3.Distance(playerStart.transform.position, transitionPoints[currentCamPos].playerPosition.transform.position);
float totalDistCam = Vector3.Distance(camStart.transform.position, transitionPoints[currentCamPos].cameraPosition.transform.position);

//calculate the distance from the player/camera to the next point
float playerDist = Vector3.Distance(transitionPoints[currentCamPos].playerPosition.transform.position, player.transform.position);
float CameraDist = Vector3.Distance(transitionPoints[currentCamPos].cameraPosition.transform.position, mainCam.transform.position);

//work out where this puts them on the lerp value
float lerpPlayer = totalDistPlayer - playerDist;
float lerpCam = totalDistCam - CameraDist;

// turn lerpCam to a value between 0 and 1
lerpCam /= totalDistCam;
lerpPlayer /= totalDistPlayer;
```

Fig 3- Code used to calculate the where the player is on the line of the on rails camera.

next and this camera acted similarly to the cutscene camera did as it would be a lerp that depending on the players movements moved the camera forward towards the next position or back along towards the last position just in case the player wants to walk back along this section. The implementation that I chose was to have two arrays of points one for the camera's points and one for the player to move too. The reason this was chosen was because it mimics the type of camera that I researched which was the camera in sections of Pokémon sword and shield (Nintendo, 2019). This was achieved by first calculating the distance the player and camera was from the next point. This creates a total distance that can be used to then calculate where the player is and then the camera can be updated to relate to this position. Once the total distance and the players distance to the next point has been calculated then through a simple calculation the players position on this line can be calculated. This can be seen in Figure 3. From then it is just turning that number into a value between zero and one and using that value in the lerp. The on rails camera state also has some strange behaviours sometimes which cannot be pinpointed now this is something to do with the transitioning of the camera to different states. However, all this is the cameras position lerping to the correct position after the on rails section.

As well as this I added a function that checks how many positions are left for the camera to move to this then exits the camera out of the on rails camera. This can be a problem as if the script exits the camera, then the cameras settings in the transition script is still that the rails camera is active when it is not. If the player does not hit a collider or a switch to change the camera then the camera can get stuck here. To fix this adding a favoured state into the transitions script so that when the camera comes out of this section the script knows to switch the camera up allows for this issue to be resolved. In addition to this I wanted to add some replay ability to this like was done for the cutscene camera. However, this does not work as if the current camera position gets set to zero and the player walks back through that section then the camera will move to the first point rather than to the last point. If this wanted to be added, then a developer could code this when it is no longer possible for the player to go to backwards. The camera also allows for the player to walk backwards and the on rails camera will move and update its position. For example, if the camera is on point one and the player walks back to the start of this section then the camera will minus one to the current camera position and go back to the state previously that was being used. This means that a player can freely explore during these sections and won't get punished for accidently missing something.

To see how the system would work and to test it in an appropriate way by creating a level that transitioned between each camera. This allowed me to find some big issues in the system such as there was currently no way to reset states so changing camera states was very unintuitive. Therefore, a function was added that allowed a

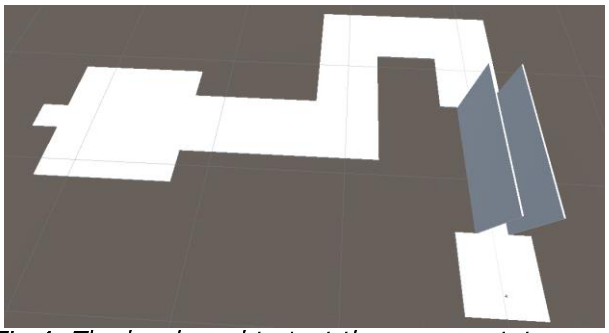


Fig 4- The level used to test the camera states

designer to reset the cameras states this then allowed someone to program the states of the camera easier as they can reset the fields of the camera back to default. As well as this because in some of the camera states the mouse gets locked when transitioning from certain cameras such as the on rails camera to the third person camera the rotation on the camera breaks and stops looking at the player. This was a big problem with my system and so without the creation of the level this may not have been found.

The level that was implemented has a section for each camera state in which the level ends with a custscene. The camera starts out on the third person camera. In this section the player walks through a walled off area with boxes in the way. This is to test the collision for the camera and how the camera would deal with these objects that could get in the way of the player. The next section is the on rails section. This section just shows how the camera works by moving around in some corners. This then transitions into the first person camera state this is only a small section as its just to test if the camera can transition into that state and see if it works. The top down camera gives the player a bigger space to walk around. Finally once the player reaches the end of the level the cutscene will play which will fly the camera from the end to the start giving the player an overview of the level.

The issue that was causing this was in the transition script I was setting the rotation of this camera to zero and this was happening every frame. Removing this code then fixes this issue however the reason for this line of code was to fix the rotation of the camera after it exits the other camera states. To fix this issue I added an empty gameobject above the players head and using the LookAt() function the cameras rotation can be set accurately as to not allow the cameras target to be out of focus.

This felt like another good point to review code and start to clean up the project. This was doing things such as fixing small bugs like when walking back on the on rails camera section it did not work because the value was too low. This then allowed me to start adding new features like allowing the cutscene camera to repeat during a level. These features were little quality of life features that just allowed some of the camera states to work better. Moreover, adding to the example level and making sure that every camera worked with all the new changes was important. Some of the smaller issues with the level was that when entering the on rails section it was hard to actually carry on walking towards the correct way as the controls aren't for this type of camera. However, by moving around some of the points this fixed this issue. Something that needed addressing was some of the codes variables such as all the bools needed for the transition scripts could be made into structs. This made the scripts easier to read and manage as all the variables were now together. Doing this for the cutscene camera and the on rails camera as well allowed for the systems to be more manageable. Also making sure that all the variables are private if they do not need to be used in the inspector. This just cleans up the look of the script.

8. Discussion of outcomes

The project has answered all the research questions and has also opened new questions. The project works and is a solid camera system that allows a developer to code in transitions into different camera states and into cutscenes. The project works well as each camera state works as intended and allows the user to perform the basic tasks. However, looking back at this some of the camera states could be improved upon. For example, the first-person camera is a very basic implementation and sometimes can be a bit difficult to control. This is something that was stated in the research as Rogers ,S(2014) stated that some of the advantages of the first person camera was aiming weapons. This is something that my first-person camera does not do well however it is possible. This is because the rotation on the camera is strange however this is playable. If this were to be done again, I would try to improve this camera.

The rest of the camera states I believe are well implemented. All the cameras other than the first-person camera has relatively no bugs or issues. The third person camera which has ended up being the base for all the cameras is good as it is easy to control and use. However this camera isn't as good as the camera in God Of War(Sony Interactive Entertainment,2018). This is nothing bad as from analysing the God of war camera it is one of the best cameras in games. However, I believe that my third person camera has all the basics of the God of war camera as it has good collision that does not allow anything to block the cameras view of the target. The improvements that could be made to this camera type would be accounting for different movements such as rolls and more complex character movements. In theory this should all be fine with my implementation however this is not a guarantee therefore to improve this camera adding more movement to the player could break the camera.

The top down cameras implementation was basic however this is all this camera needed to do as all the games that use this camera rely on different movement controls to take full use of the camera. The main inspiration for this camera type was from league of legends(Riot Games, 2009). This camera type only really needs to worry about being static and just locking on to a target. To improve this camera a feature that could be add would be to allow someone to dynamically add in a new target for the camera to focus on this could be useful to then use this camera for spectating games that could have multiple points of focus.

The on rails camera was a something that required the most attention as it had a lot of issues with it. However once the camera started to work it is a good implementation and works similarly to the example that I used which was Pokémon Sword and Shield(Nintendo, 2019). However, the way that this was approached helped as it allowed me to set out and plan a method to do this. Without this planning the on rails camera would have taken longer to implement. However, a limitation of the camera is that it does not move on a curve as it is a direct lerp. Spirals can still be achieved as they the designer can create multiple points around the spiral that the camera and player can follow. This is one of the main limitations of the system as it can be tedious to continuously place different points around a sphere. However, this can be mitigated by turning the spiral into a hexagonal shape as so then the spiral would still work with my system.

Finally, was the cutscene camera. This system works very well as it allows a user to create points in the world using game objects and then the camera loops through and moves to those points. This cameras inspiration came from God of War(Sony Interactive Entertainment, 2018) as in this the camera transitions from a third person state into a cutscene seamlessly. This is something that my cutscenes do as well as they move to all the points set seamlessly. Also the camera has a function that allows a developer to pick a target similar to how in God of War(Sony Interactive Entertainment, 2018) the focus of the camera isn’t always the main character.

Reflecting on the project management side of the project and the use of sprints. The use of sprints was good in theory as it would allow me to work on the project consistently. However, what happened was that the approach I took was more of an agile approach where once the systems were done then id review and iterate on the whole system. This was also a valid approach and allowed me to also balance time between other work. If the project were done again then switching to the intended sprint approach would be better as it would allow me to create a smaller system and then improve it. This would allow me to improve the projects quality quicker as a whole.

Looking back at the impact the level had this allowed me to test and refine the camera states that were implemented. The level that was created was simple, but it allowed for me to test each state work and proved the transitioning into state through collision boxes worked. The main goal of this was to use this a bug testing ground. Which ultimately is what it became as it allowed me to find bugs and flaws within that cameras that would affect a user that wants to create a level with my system. This would have been bad as one of the main reasons for this project is to make it easier for designers to start creating games as the cameras behaviours and states are already coded. However, this would have made it harder as they would have had to bug fix a system that is not their code or would just not use the project. With the level implemented it allowed for the project to be improved upon. One of the main problems with the level was that the on rails trigger was hard to activate and stay activated however this was fixed by instead of using on trigger enter using on trigger exit instead to change the camera state after the player leaves the collider. Without the level this interaction would not have been found.

One of the research questions was to see if each camera state worked. This research question turned out to be true as all camera states work to some level and allow the player to control them and a designer to manipulate them. This gives me the base to continue improving on this system in the future.

The next research question was how well the cameras work with a developer using them. To test this, I created a simple level that takes advantage of each camera by using colliders to transition between each state. Mostly the cameras are easy to use however setting some of the cameras up can be hard such as the on rails camera as this camera needs two sets of points one for the camera and one for the player as explained in the section above. As well as this the positioning the third person camera at the start of using the system can be hard. As because of some of the scripts code it can force the camera into an offset that was not intended. But if this is understood then the camera is easy to use and set up.

Another research question that was answered was how would this system be able to be extended? Some ways this project could be extended would be adding different cameras like mentioned in the research questions section. As well as this the project itself has been set up so that it should be easily extended this was a conscious thought whilst making the system. Due to not being abled to fully implement some features this then allows me to improve and attempt them in the future.

The next research question was one that was based off of the cutscene in the God Of War(Sony Interactive Entertainment,2018) game. This question was is it possible to transition between gameplay and cutscene well? Like is done well in God Of War(Sony Interactive Entertainment,2018). This was a hard to achieve as the cutscene camera was never playing a video to the player which could have a different quality to that of the in game world. Therefore, the answer to this question is yes because of the implementation of the cutscene camera where everything it is showing must be achieved in the engine at real time. This then will create the sense that the world is not being broken and the player will always experience actions the same was as they would experience them in game the only difference is they do not have control over them.

The final research question was is it possible to make a cutscene system that has high impact to a player. As stated, before this was hard to gauge however it is not impossible as in modern times the best way to do this is to add meaningful narrative to the cutscene. This in the project is very hard to achieve. However, looking at more modern games that have similar impactful cutscenes such as God Of War(Sony Interactive Entertainment, 2018) and Horizon Zero Dawn(Guerrilla Games, 2017). This backs up that narrative makes impactful cutscenes. The best way as a designer to do this is to try and give the developer as many tools as possible to achieve this. The impact this has on my system is that adding extra functionality such as multiple camera targets that allows somebody to switch between the targets or just a single target for the cutscene as well. This would allow my system to be used and to hopefully be able to make a difference.

This project has allowed me to improve my knowledge of cameras in the real world and more importantly in the game world. As well as this my knowledge of lerping and camera movement and rotation has improved. However, from this project no new information was found regarding cameras in games. I believe that this is due to how cameras are already master's in games and very few games come out now that do not have very good cameras. But overall, this project has been a success as all the camera systems work and can be improved upon still.

9. Conclusion and recommendations

As stated above this project was a success as I have succeeded in creating a camera system that allows for transitions and multiple states. To improve and extend the project creating more camera states for the system to go off would allow the system to be more dynamic. Also simplifying the cameras use by adding more way for a designer to use and manipulate the system would be a good starting point to further the project. This project could also be carried on using academic journals to report on the progress. This project if these systems are improved upon would give someone using it an easy time in being able to jump into developing a game as all the camera behaviours would be already covered. Also, a recommendation for the continuation of the project would be to give the system to multiple game developers and each ask them to create a level using all the camera types as well as documenting their approach and issues. This would allow for the camera scripts to be able to be improved and allow for the actual system to be improved and iterated on. As well as this smoothing on the cutscene camera was mentioned in the outline this is a quality of life feature that would make the cutscenes be more professional.

Overall, the project was a success as it achieves everything that I set out to achieve with it and has been set up for further improvements.

10. References

God Of War(2018) developer [SIE Santa Monica Studio] , Publisher[Sony Interactive Entertainment].

Super Mario 64(1996)developer[Nintendo], Publisher[Nintendo].

Rogers ,S(2014) Level Up: A Guide To Great Video Game Design[Online]. 2nd edition VLeBooks.[Accessed 28/04/2021]

Haigh-Hutchinson, M., (2009). Real time cameras: A guide for game designers and developers.[Online] CRC Press [Accessed on 28/04/2021].

Pokémon Sword and Shield developer[GameFreak], publisher [Nintendo].

Figure 1 – Wispmetas(2019) Pokémon Sword and Shield Fixed Camera[Screenshot] in: resetera fourms[Online] Available from: https://www.resetera.com/threads/pok%C3%A9mon-sword-shield-gameplay-reveal-trailer-raid-battles-new-dynamax-mechanic-overworld-pok%C3%A9mon-encounters-new-pok%C3%A9mon-out-nov-15-2019.121044/page-7#post-21449106 [Accessed 18/01/21]

Schramm, J(2013)Analysis of Third Person Cameras in Current Generation Action Games[Online] Gotland University, School of Game Design, Technology and Learning Processes. [Accessed 28/04/21]

Apex Legends (2019)developer[Respawn Entertainment], Publisher[Electronic Arts]

League Of Legends (2009)developer[Riot Games], Publisher [Riot Games].

Half Life(1998) developer[Valve Corporation], Publisher [Valve Corporation]
Horizon Zero Dawn(2017)developer[Guerrilla Games], Publisher[Sony Interactive Entertainment]

ChicoUnity3D(2014) How To Lerp Like A Pro Available From: https://chicounity3d.wordpress.com/2014/05/23/how-to-lerp-like-a-pro/ [Accessed 30/04/21]

Unity(2021) Unity Lerping documentation Avaliable from: https://docs.unity3d.com/ScriptReference/Vector3.Lerp.html [Accessed 29/04/2021]

11. Bibliography

God Of War(2018) developer [SIE Santa Monica Studio] , Publisher[Sony Interactive Entertainment].

Rogers ,S(2014) Level Up: A Guide To Great Video Game Design[Online]. 2nd edition VLeBooks.[Accessed 28/04/2021]

Heussner, T.H; Toiya, F.K.T; Helper, J.B.H; Lemay, AL(2015) The Game Narrative Toolbox[Online]. First Edition O'Reilly Publisher Routledge [Accessed 06/01/2021]

Pokémon Sword and Shield developer[GameFreak], publisher [Nintendo].

Strachan, D.S(2020)Screen Transitions for Computer Games and Films Available from: http://www.davetech.co.uk/screentransitions [Accessed 06/01/21]

Kelly, T(2011)Camera Comes First[Game Design] Available from: https://www.whatgamesare.com/2011/10/camera-comes-first-game-design.html [Accessed 11/11/20]

Super Mario 64(1996)developer[Nintendo], Publisher[Nintendo].

Horizon Zero Dawn(2017)developer[Guerrilla Games], Publisher[Sony Interactive Entertainment]

Apex Legends(2019)developer[Respawn Entertainment], Publisher[Electronic Arts]

Kuchera,B(2018) God of War's camera was a huge risk that paid off Available from: https://www.polygon.com/2018/4/23/17263016/god-of-war-playstation-4-camera-single-shot [Accessed 18/01/2021]

Call Of Duty: Black Ops Cold War developer[Treyarch, Raven Software], publisher [Activision]

Haigh-Hutchinson, M., (2009). Real time cameras: A guide for game designers and developers.(Online) CRC Press [Accessed on 28/04/2021].

Unity(2021) Unity Lerping documentation Avaliable from: https://docs.unity3d.com/ScriptReference/Vector3.Lerp.html [Accessed 29/04/2021]

Appendix A: Project Log

Date	Improvements	Things To Do	Additional Comments
20/10/20	N/A	Improve Proposal.	Improve proposal and make it more interesting and detailed.
20/10/20	Filled out the proposal a bit added some more research(mario) and detail.	What engine is this being created in? Add in the book about camera systems	1000 word cap
23/10/20	Had meeting with supervisor	Use new template and add in the engine I'm using	
03/11/20	Wrote 2nd draft	Get draft check by supervisor	
10/11/20	Feedback got back from supervisor improve draft based off feedback	Check over draft	
11/11/20	Submitted proposal		
24/11/20	Start work on base third person follow camera	Complete follow cam Add collision to the camera	Will need to revisit this and make sure that this is efficient and optimized.

1/12/20	Finished base third person follow camera and collision	Next add in options for a first person camera during runtime and for designers to choose between.	Ensure the transition between is good and works both ways. Make sure to create this with more scripts in mind.
10/12/20	Start work on first person camera	Finish implementation of the fps camera	
29/12/20	Started some basic research for the research documentation	Finish the research document and get to a point for the project video	
4/01/21	Finished a draft of the research doc	Finish references and appendix	
07/01/21	Edited the doc to make the research section more specific and better	Finish references and appendix	
13/01/21	Added some more references and bibliography and appendix	Check through and make sure everything is good	
18/01/21	Used feedback from supervisor to improve research document	Check through new draft	
22/01/21	Added in top down camera base and basic transition	-Rotate camera to look at the player -some issues with the fps camera but this can be done later -Some issues with transitions when multiple are true	Remember to improve the fps camera. Maybe add in some more advanced functionality into all the scripts
24/01/21	Finished basic implementation of top down camera	-still issues with fps cam and when Multiple bools are true nothing major now -start the on Rails cam	
25/01/21	Put in the base logic of the on rails camera	-finish the implementation -see if it's worth making it in its own script or in the transitions script.	
26/01/21	Ran in some troubles with the base camera's rotation. Fixed these issues need to re consider the top downs cameras rotation issues and moving out of a top down camera.	Fix the top down camera	
27/01/21	Fixed the previous issue.	-Work on a to automatically move states.(stop bug) -carry on with the on rails cam	
29/01/21	Continued to make progress on the system for the on rails cam. Some of the same parts of this will also be used for the cutscene.	-finish this system and start on the cutscene creator	
02/02/21`	Finished WIP video	Check with supervisor to see for feedback	

04/02/21	Improved WIP video based on feedback	Carry on with the implementation of the system.	
15/02/21	Started work on onRails cam	Carry on with on rails implementation	
16/02/21	-Cam is not moving to points -Points are resetting when game runs -movement and camera target works	Fix the two issues and finish the implementation then start on the cutscene creator	
22/02/21	-fixed some issues with the top down camera -struggling with getting the on Rails cam moving not sure how to detect if the player is moving in the correct way or not -the positions for the gameobjects were wiping on run this is now fixed	-Carry on with this system and maybe start debugging the rest -also start work for the cutscene system	
28/02/21	-Basic implementation of the on rails cam a lot of bugs atm that need clearing up but the basic system Is there	-carry on with fixing and refining this system then start on the cutscene	
01/03/21	-fixed some of the previous clipping issues -started implementing cutscene system	-need to fix/ carry on with the on rails cam -continue fixing bugs - consider optimizing existing systems	
02/03/21	-cutscene works but only for one object now trying to sort out smoothing out the transition	-carry on with the fixing the cutscene and on rails camera	-might want to consider making a separate function in transition script to move the camera to the different objects using delta time
08/03/21	-Started work on the practice outline	-carry on with the fixing the cutscene and on rails camera	
15/03/21	-Outlined the design of the final level to show off -need to sort out cutscene and on Rails cam -Cutscene now works a tiny bit jittery need to add a way to now go through each of the different gameobjects that the user wants to -camera position resets after lerp	-want to block out a “level” that shows off the different features of and cycles through all the scripts I have created - continue fixing the on rails and cutscene script	

16/03/21	-Added in some functions to look at something during a cutscene(This needs to have a rotation reset after the camera is done)	want to block out a "level" that shows off the different features of and cycles through all the scripts I have created - continue fixing the onrails and cutscene script	
30/03/21	Started work on my practice outline		
31/03/21	Continued work on my practice outline		
07/04/21	Still having issues with the lerping for the cutscene. The issue is that the lerp resets after exiting a if loop.	Fix this asap Start work on the polish for the whole thing	
08/04/21	Fixed the issues and now cutscenes work as intended. Need to add in a favored state for the designer to return the player into after the cutscene	Finish the OnRailsCam Add the level in Then review each system	
14/04/21	On Rails Cam base finished	-Transitioning out of onRailsCam breaks the base camera. -walking back on the on Rails Cam does not really work consistently	Optimal space for the on Rails Cam is close levels that do not allow the players to move around too much
15/04/21	Finished base example level	Fix bugs with camera Finish implementation of level (colliders and camera switches)	
16/04/21	Added the scripts and colliders for camera switches in the level	-fix bugs -test the level works properly -write final report	May need to look at disabling animations for the first person camera or changing this in some way
18/04/21	Testing the basic level for bugs/issues only problem so far is that the FPS cam sometimes has rotation issues.	-fix bugs -test the level works properly -write final report	
23/05/21	Tried to fix some bugs. Tested level and mostly works as intended	-fix bugs -write final report -fix slight bugs in level	
24/05/21	Fixed a major bug with FPS camera	-may need to rewrite the transition script as this script is messy and is causing issues with other cameras.	

		-write final report -fix slight bugs in level	
25/05/21	Fixed some of the bugs in the level	-may need to rewrite the transition script as this script is messy and is causing issues with other cameras. -write final report	
26/04/21	Found some more bugs that is caused by disabling input with some of other cameras reinforces that the transition script needs to be re written. -added to report	-may need to rewrite the transition script as this script is messy and is causing issues with other cameras. -write final report -the On rails cam and top down cam breaks the third person camera	-small bug FPS Camera tilts slightly I believe it something to do with the the player as the player tilts.
27/04/21	-Found the issue with the rotation of the third person camera not working after certain situation was setting the rotation in a loop in update. -no longer need to rewrite the transition script however this script could use a clean up -another issue with the transition script lerps is always applying this leads to strange camera movement -wrote part of the final report	-fix the issues found -clean the projects scripts out -finish report	Need to finish the report as soon as possible as to get feedback Maybe look at adding something to the system if time permits.
29/04/21	-Fixed the previous issues outlined with the camera transitioning to the third person script -Issue with the lerps applying all the time is not fixed however the behavior no longer causes glitchy behavior -continued work on	-Finish report	
30/04/21	-Continuing writing report -added cutscene camera now can repeat	-Finish Report	
03/04/21	-Finished Report		

Appendix B: Project Timeline

October		x days x days
November	Hand In - Final proposal to be submitted by 12/11/20	2

	Sprint 1- More Research on different games Base camera implementation third person camera **Sprint 2-** follow script Add In first person camera Debug any bugs with the system	14 14 (Throughout the Project)
December	**Sprint 3-** Add In top down camera **Sprint 4-** Write up research Add camera collision	14 14
January	**Sprint 5-** Hand In - Documentation via blackboard by 14/01/21	14
February	**Sprint 6-** Hand In - Prototype demos by 05/02/21 Add in spline behaviour for third person camera	14
March	**Sprint 7-** Add in transitions between different cameras **Sprint 8-** Add Cutscenes into the system	14 14
April	**Sprint 9-** Optimising the camera Hand In - Practice outline by 15/04/21 **Sprint 10-** Clean up the system Optimise the system	14 14
May	**Sprint 11-** Hand In - Final report and artifact by 06/05/21 Hand In - Final Video by 18/05/21 Hand In - Viva by 25/05/21	14

Appendix C: Assets used in the Project
This is a list of project assets: all source materials used in the project. Clearly state which were produced by yourself and which were not. If not produced by yourself, include their reference, and status with regard to copyright/ creative commons licensing.

RPG Character Mecanim Animation Pack FREE. These assets camera from the unity asset store and can be found at this link: https://assetstore.unity.com/packages/3d/animations/rpg-character-mecanim-animation-pack-free-65284. The assets that were used from this was the player character model and animations as well as the materials.