

Puzzle Platformer



Computer Games Technology BSc Hons

Supervisor: Silvester Czanner

It will apply to the current project protocol and timeframe stated. If
to review the ethical consideration(s) and this will include completion

Disclaimer

No part of this project is
qualification at this or
citations to the work of

agree or
object containing

Signed

[Redacted Signature]

Date: 25/10/16 (DD/MM/YY)

Project is (please check the appropriate box):

ues requiring further consideration and the project can

s requiring further consideration and will refer the project
Officer.

_____ Date: _____ (DD/MM/YY)

_____ Position: _____
r UG and PG Taught/ PGRs RD1 Scrutiniser/

Abstract

This is a computer games technology project, looking to create a playable game combining two alternative genres from the computer games industry. The project is intending to create a platforming game that uses puzzles to challenge the player and test their in game ability. The hopes of the project are to create a working level of a game that contains a variety of puzzles and a reliable player controller that is found enjoyable by intended players of the game upon completion. There are similar pieces of work on the market that can be used as references of what players find enjoyable and different styles of game that have worked in the past, as well as what the players tend to expect from such a game. The project will be made using game engine software that is found idea for its requirements and the main priorities are ensuring player enjoyment, lack of bug and glitches and thorough testing to ensure that these goals have been achieved.

Acknowledgments

I would like to thank Silvester for offering his support throughout the entire project

Contents

Disclaimer	1
Abstract	2
Acknowledgments	2
Project Introduction – Section 1	6
Background 1.1	6
Approach 1.2	6
Aim and Objectives 1.3	6
Aim 1.3.1	6
Objectives 1.3.2	6
Report Organisation 1.4	7
Literature Review – Section 2	8
Game Engine 2.1	8
CryEngine 2.1.1	8
Unreal Engine 2.1.2	8
Unity 2.1.3	8
Platformer Games and mechanics 2.2	8
Designing a Platforming game 2.2.1	9
Puzzles in video games 2.3	10
Puzzles used in video games 2.3.1	10
Designing a Puzzle game 2.3.2	11
Games Design 2.4	11
Summary 2.5	12
Product Design – Section 3	13
Design Introduction 3.1	13
Concept Design Requirements 3.2	13
Game Layout 3.4	13
Map Design 3.5	14
Puzzle Design 3.6	14
Moving platforms 3.6.1	14
Falling Platforms 3.6.2	14
Alternating locks 3.6.3	14
Jumping Puzzle 3.6.4	15
Key cards 3.6.5	15
Laser Grids 3.6.6	15
Invisible Bridge 3.6.7	15

Moving Walls 3.6.8.....	16
Shrinking Puzzle 3.6.9	16
Implementation – Section 4	17
Game Engine Decision 4.1	17
Programming Language 4.2.....	17
Player Controller Development 4.3	17
Double Jump 4.3.1	17
Dash 4.3.2	18
Shrink 4.3.3.....	19
Line Trace Gun 4.3.4	20
Timer 4.3.5.....	20
Puzzle Development 4.4	21
Moving platforms 4.4.1.....	21
Falling Platforms 4.4.2.....	21
Alternating locks 4.4.3	22
Key cards 4.4.4	23
Laser Grids (Moving) 4.4.5.....	25
Laser Grid (Timed) 4.4.6	26
Moving Walls 4.4.7.....	26
Testing and Evaluation – Section 5.....	28
Introduction 5.1	28
Alpha Testing 5.2.....	28
Process of Testing 5.2.1	28
Testers Opinion 5.2.2	28
Beta Testing 5.3	28
Process of Beta Testing 5.3.1	28
Directed Testing Script	28
Questionnaire	29
Beta Testing	31
Beta Results Commentary.....	35
Research Evaluation.....	36
Design Evaluation.....	36
Implementation Evaluation	36
Time Management Evaluation	37
Conclusion – Section 6	38
Closing Remarks	38

Appendix.....	39
Terms of Reference.....	39
Ethics Form.....	42
Bibliography	45

Figure 1	14
Figure 2	15
Figure 3	16
Figure 4	16
Figure 5	18
Figure 6	18
Figure 7	19
Figure 8	19
Figure 9	20
Figure 10	20
Figure 11	21
Figure 12	21
Figure 13	22
Figure 14	22
Figure 15	23
Figure 16	23
Figure 17	23
Figure 18	24
Figure 19	24
Figure 20	24
Figure 21	25
Figure 22	25
Figure 23	26
Figure 24	27
Figure 25	31
Figure 26	31
Figure 27	32
Figure 28	32
Figure 29	33
Figure 30	33
Figure 31	34
Figure 32	34
Figure 33	35
Figure 34	35

Project Introduction – Section 1

Background 1.1

There are already many platforming games on the market with a variety of objectives and have been for a very long period of time in both 2d and 3d. A few examples of long running popular platforming games would be the Mario (1985) and Sonic the Hedgehog (1991) series which have created games of both forms. The general theme of the two is to keep moving forwards and collect items which is where the intention for the game changes from the ‘normal’ structure as it is to be intended for the player character to keep moving upwards in an extended jumping puzzle with other puzzles added to increase the challenge. In addition to there being many platformer games there are also a large variation of puzzle games, with one of the most well-known being the Portal (2007) series which involves using a ‘portal’ to manipulate the areas and reach the goal.

The inherent challenge in the creation of a puzzle platformer game comes from the difficulty of its puzzles and how to make them challenging yet enjoyable for the player. There also arises the complication of designing a varied number of puzzles and fitting them into the design. Other challenges are likely to arise from the development of the game itself including the coding used and the addition of any potential extras such as 3D virtual reality implementation. Overall there will need to be plenty of research into puzzles that would be able to fit with the games theme and a few tests to see if the users are able to complete any puzzles that are presented.

Approach 1.2

The game will need to use specific software intended for the production of games and it will need to be able to provide an ideal user experience both during their time playing the game and the overall accessibility of the game. The game will be driven using previous experience with a variety of game engines and different programming languages, as well as self-taught understanding of different aspects of game development and online support and documentation. Upon completion of the game there will be a focus on thorough testing of individual features in varied environments and conditions to ensure that the final product ends up running as intended in all situations.

Aim and Objectives 1.3

Aim 1.3.1

To create a 3D game where the player is required to navigate a series of obstacles such as jumping puzzles and mazes to reach a specific goal.

Objectives 1.3.2

The game will need to have a few objectives which will determine separate stages of its design and development and will be used to review the progress that has so far been made.

1. Game Engine Review – Look into a variety of different game development software and compare their inherent strengths and weaknesses with the needs involved with developing a puzzle platformer.
2. Product Design – Use information gathered from other attempts at similar projects and design a game that is both different from what has come before but does not decide to reinvent the wheel.
3. Development of the Game – Use the designs of what the game is intended to be to make a game that has as few bugs as possible in the time provided through careful time management and clear set goals for completion

4. Thorough Testing Period – Ensure that the game goes through multiple separate stages of testing to get the best possible results for the players who will be using the game at the end to enjoy without any problems they are likely to encounter or suffer through
5. Resolution of Problems – Use the information gathered in the testing period to go back over the implementation process and improve any features that may be lacking and removing anything that is unnecessary or should not be in the final version.

Report Organisation 1.4

The project report will be structured and organised in the following manner:

Section 2 – Literature Review

This segment will focus on the work of other people that are similar to the project aims to provide an understanding of what is usually expected of such a game and how these expectations have been met in the past.

Section 3 – Project Design

This segment will contain the designs for each individual section of the game and will outline what each feature of the game is expected to do and how it is expected to do it. It will include plans for the overall layout of the level map and specifications for how the final game should look and feel for the players, highlighting what should be considered the highest priority based upon the research completed in the literature review.

Section 4 – Implementation

This segment will cover how each feature from the design has been implemented into the game and how each different feature is expected to work in practice, as well as definitively explaining what software is being used and why it was considered the better option.

Section 5 – Testing and Evaluation

This segment will include the testing stages of the game and the opinions of the testers in both structured and recorded observations and free play, recording the players feeling as they continue through the level. It will then also feature numerous personal review of individual stages of the development process and where improvements could have been made at any point.

Section 6 – Conclusion

Will tie together the project and provide a personal assessment of the final product and the process it went through to get there.

Literature Review – Section 2

Game Engine 2.1

In order to create the game there will need to be a game engine that will be both ideal for the game mechanics and features as well as providing sufficient usability that it does not become a taxing experience on its user. Of all the game engines, three of the main engines would likely be ideal for the production of the game and should have their strengths and weaknesses checked for consideration. First of all CryEngine, which has a strong reputation for being the best looking engine and has a strong inclination towards PC and console game production however also requires a strong learning curve to truly utilise it. Unreal Engine also has a reputation for producing good-looking games however it comes with the added benefit of providing a simpler user experience in the development process of a game and no stark weaknesses aside from its comparative lack of user and developer support guides for the production of different game styles. Finally there is Unity which is one of the most widely used game engines and has the most by far in terms of support from both its users and its developer which means it can be used to make almost any game after only a relatively shorter period of research. Unity's weaknesses come from its strong inclination towards 2D games and its lack of in-engine designing, making mock-ups of a level harder than other engines such as Unreal; however, it has proven to be a capable and reliable engine for game production.

CryEngine 2.1.1

CryEngine is one of the most powerful engines available to game developers and has been used to produce games that have graphical capacities surpassing Unity and largely Unreal as well as a larger variety of lighting, a more reliable physics engine and highly advanced animation systems. The price for this is that there is a much steeper learning curve for its users. It is ideal for the production of games that want high-end graphics but otherwise it is recommended to look to other engines.

Unreal Engine 2.1.2

Unreal engine provides a graphical capability almost on par with CryEngine but is much simpler for a newer user to use. It also possesses a system of BSP's which allow the user to design and plan an area before producing a game. The downside to using Unreal over other engines is that whilst it is easier to use than CryEngine does not possess the sheer quantity of support material that Unity has, yet it is still one of the best potential engines for the production of a first person game.

Unity 2.1.3

Unity is one of the most supported game engines around and can be used to produce a large variety of games; it is easy for a user to grasp and provides guides online for the production of almost any game template. The downsides to using Unity are that it has a strong inclination towards development for mobile devices and 2D games which is not ideal for the development of a 3D platforming game and it also has a lower graphical potential than the other engines.

Overall when considering the inherent strengths and weaknesses of each engine CryEngine is not ideal for the production of a platforming game due to its unnecessary complexity in relation to the task and between Unreal and Unity, Unreal's allowance for a mock-up using BSP means it is much more useful for the production of a puzzle game which may undergo frequent level changes.

Platformer Games and mechanics 2.2

"Platform games are one of gaming's most enduring staples" [Towell 16]

Platforming games are video games that revolve around playing as a character who runs around and jumps up and down floors, onto ledges and climbs stairs and other objects, and is most frequently described as a sub-genre of action game. The first platforming games were developed in the early

1980's which makes the genre one of the oldest to currently exist. The Genre is viewed to have truly begun with the release of the games 'Space Panic' and the later release of 'Donkey Kong' and the two games have displayed a clear influence on all those that followed them in the level design and concept. As a genre, platformers have also shown a strong ability to incorporate mechanics and features from other genres quite comfortably including the levelling and abilities features of role-playing games. [Klappenbach 16]

The article by Klappenbach provides an overview of the pedigree of platforming games as well as a brief summary of what they are at their core, then moving onto the varieties of platforming games that exist. The first sub-type mentioned are single screen platformers that involve the player only using a single screen that will likely contain obstacles that are to be avoided in order to complete the mission. He goes on by providing an example of this type of platformer game such as Donkey Kong 1981 where the player character uses steel platforms and ladders to continue moving forwards while dodging barrels that have been thrown down.

He also goes on to describe another sub-type which is 'Side and Vertical Scrolling Platformers' which he defines as games that can be identified by its scrolling game screen and a background that moves along with the player as they approach the edge of the screen. He also states that in such games the player is likely to travel across the screen collecting items, defeating enemies and completing a variety of objectives as they progress until the level is complete before moving onto the next, more difficult level. It is not uncommon for such games to end each of their levels in a form of boss fight, which must be defeated before advancing to the next level. Klappenbach also goes on to provide a few examples of such games, including 'Super Mario Bros', 'Castlevania' and 'Sonic the Hedgehog'.

Klappenbach finishes his article describing the decline and subsequent resurgence of platforming games as a genre. To open he claims that the rise of high end graphics and the increasing complexity of most video games is to be blamed for the decline considerably since the late 1990's. However he then goes on to describe the return of the genres popularity, crediting the popularity of mobile gaming.

Throughout his article, Klappenbach describes many different types of platformer game yet each one shares strong similarities. Each game is likely to involve a jumping character moving from platform to platform and avoiding hazards with changes mainly to the camera angle used in the game

"The platforming genre is arguably the most important and successful genre in the history of gaming" [Racketboy 16]

Designing a Platforming game 2.2.1

Racketboy's article also mentions the early history of platforming games however does not go into detail on the different types of overall gameplay and focuses more on the individual games and their mechanics. He does go on to describe the 'conventions' used in the platforming genre consistently over the years as well as critical gameplay elements that have been used in order to make these games. The apparent conventions that Racketboy has stated include 'parkour and freerunning', physical activities based upon fast paced and efficient manoeuvring around obstacles and opponents yet showing a strong focus on style and flare as well. He states that these mechanics can best be seen more in modern platformers such as 'Mirror's Edge', 'Tomb Raider' and 'Prince of Persia'. Another convention is the design of level based upon a specific element, such as fire, ice and water, with fire levels often requiring specific timing to avoid fireballs or rising lava and ice levels having the character slip and slide around instead of stopping comfortably before dangerous traps. Water is also often used to slow the player and to constantly rise until the player has drowned. Another relevant convention involved is the concept of 'speed running', which is the player focus on the fastest completion of the level possible to achieve the higher rewards their speed will provide.

Racketboy then goes on to mention critical elements in a platforming games design based upon its basic gameplay elements. The first element is character movement, the basics controls of the game influence the difficulty and potential enjoyment of a platformer. The influence of the controls is highly affected by how high the character can jump, how much their moving speed affects their jumping distance and in air manoeuvrability. There are also other features involved in controls, such as having the character sliding when landing and left, right forwards and back responsiveness and speed. Racketboy states that the controls have a vital effect on the 'feel' of a game, which subsequently is a key element in player enjoyment. Another vital component of a platforming game to consider in the design is its physics. Racketboy states that given the strong emphasis on movement in platforming games the physics it uses are a key element, referring to games such as portal and sonic the hedgehog to show how physics can be applied in different ways. In both he refers to the use of momentum, from gravity in portal helping to clear large jumps and speed in sonic to complete a loop-de-loop.

Other notable gameplay mechanics to consider in a platformers design are the environmental interactions that the player must make. This includes knowing which environmental objects can be used to traverse the terrain, such as hanging vines and ledges being climbable objects, swinging on ropes and sliding down a pole. Racketboy points out that in a platformer these are highly noticeable as in any other genres they are backdrop as opposed to vital mechanics. The camera used in the game is also stated to be vital in the design as it can be one of the most infuriating aspects of a platformer game if done wrong, such as not being able to accurately gauge jump angles or distances. Racketboy claims that games should have smart cameras that adjust themselves to the right position instead of players having to manually address it constantly. Other staples of platformer design include environmental hazards, traps and obstacles such as lava pits, spikes and arrow traps as in many platforming games the greatest opponent is the environment, with many historic games of the genre having infamously difficult traps to stop the player. Collectible are also frequently strewn throughout the level with the aim to grant the player a boost such as invulnerability or a score increase, maybe If the player collects enough it will grant them bonuses such as secret rooms or extra lives. These collectible have not always been obvious either with some being hidden throughout levels needing the player to hunt them down. The final design consideration Racketboy highlights is the double jump mechanic, a feature that allows the player a second jump whilst in the air in order to give them extra elevation or change jump direction. As a feature, it is notable for reducing frustration in the gameplay.

Puzzles in video games 2.3

Puzzles used in video games 2.3.1

TVTropes is a website that uses input from multiple people to gather large stores of information and it has collected information on many of the different puzzles that have been used in video games. They open by saying that puzzles are an element that many game developers want to add as they give the player time to 'relax their fingers and stretch their brain', even if they don't make real sense it is often credited to it being a video game after all. Some puzzles used are far more common than others, some used in almost every adventure game made to date, largely due to their effectiveness. Some examples of 'stock puzzles' can vary between a simple block puzzle which requires the player to push a block into place to cover a switch or create a path that can be walked across or jumped onto. Another example is the 'control room puzzle' where the player is given a series of switches and only one specific setup will allow the player to continue. Other puzzles include keypads, where the player is required to input an answer they have received elsewhere or a lock and key where they need to find an item instead of a code to open the door. There are also common puzzles that involve the player characters movement being altered; frictionless ice puzzles make the character slide around in one direction; Conveyor belt puzzles with forced movement buttons push the character in one direction

and can force them back to the start of a puzzle and block entry and the stepping stone puzzle which involves creating a series of floating platforms in order to safely cross a chasm.

Other puzzles often involve altering the environment, such examples would be reflecting light with mirrors, designed to be positioned in such a way that the light is forced to shine on a specific object and open a door. There are also trick shot puzzles where the player has to aim a projectile in such a way it trigger the release of a door mechanism or releases a ladder and weighted switch puzzle where the player has to place an object on a switch and leave it there to keep the door open. [Puzzle 16]

Designing a Puzzle game 2.3.2

Tulleken's article goes over the history of puzzles in general, from the labyrinths to angry birds and how the addition of computers has made such dramatic changes to how puzzle games are designed. He states that there are five categories of puzzle, Mathematical, Physics based, Visual Field, Language and Common sense/ situational. Mathematical puzzles include Sudoku, chess, minesweeper and mazes, in short puzzles that the player needs to use logic to solve. Physics based puzzles include bridge builder and angry birds as they require the player to account for gravity and physical impacts to complete their objectives. Visual field puzzles are things that can be done by eye alone such as spot the difference and hidden object puzzles. Language puzzles use text and include games like hangman and crosswords where a grasp of a specific language is necessary. Common sense / situational puzzles require general knowledge but not always logic, such as riddles and treasure hunts. Each of these categories can be described as broad and Tulleken goes on to say that each category can be divided even further by the type of play they permit.

Tulleken follows his classification of the different types of puzzles with classification by their design elements. One of the first he mentions is the use of procedural puzzle games where the designer plans the core mechanics involved and then allows the computer to put it all together to challenge the player. The most common examples of procedural puzzles are Tetris and Minesweeper where the difficulty is controlled by limiting how much the computer may use and by scaling the puzzle. Procedural puzzle games have no level design. At the other end of the spectrum, there are heterogeneous puzzle games, a process where the designer hand creates each puzzle used. This is found most frequently in adventure games and require a combination of common sense and lateral thinking to solve. The difficulty of each puzzle can be increased by obscuring the solution or combining puzzles into more complicated systems. Tulleken then goes on to say that, the other design method is between these two extremes, calling them combinatorial puzzles games, where the core mechanics are reused often and in a variety of combinations. Levels are often carefully hand designed in this method and a few strong examples are games like Portal and Braid. The difficulty can be increased when the designer introduced more mechanics, forces the player to use the mechanics in a smarter manner or combines more mechanics into each puzzle.

Tulleken finally describes ways in which puzzle design elements can be applied to other games, claiming that it helps to focus on the games mechanics, level designs, teaching mechanics to the player, revealing of information, complexity of combining mechanics, difficulty, progression and how a players perception shifts.[Tulleken 16]

Games Design 2.4

"Which is better, having the best darts game available or having a middling darts game and a bad game of pool?" [Koster 16]

Koster has written an article based upon the comparison of Fun vs Features involved in designing a game based upon the time constraints that the creator is likely to be under. The article uses a game of darts as a comparison to the production of a game and points out how the addition of a larger number of features can have a negative effect on the enjoyment of the end result. In his analogy, he uses the darts game in a bar to state that the creator will have a specific amount of time. He then goes on to say

how that time could be applied, it could be used to focus specifically on one game, in this case the darts to make it the best darts game it can be in the limited time provided or how they could add a pool table, something completely different with a whole new set of commands and features. This new added feature means less time can be spent making the darts game brilliant and due to the time already spent it is also unlikely that the new pool game will be any good. The message he tries to get across with his analogy is that despite how the extra features may mean there is more to do and a better game, if they can't be completed in the amount of time the creator has they only reduce user enjoyment and make the whole game worse, implying that in the process of designing a game it is better to focus on fewer features and ensure they are completed thoroughly than to try and add an unnecessary amount. This comes simply from the fact that whilst each new feature offers the strong potential of fun for the user, the fun requires tuning and balance from the creator and doesn't just appear from the feature's existence. He then goes on to say that if such a feature is going to be added, complete the original features first then focus on the extras, promising to add them later if necessary, so long as they are each fully fleshed out and enjoyable when they are released and don't detract from the final solution.

Koster finishes his article describing his own method to game design, he states that he always leans towards 'elegant systems', meaning they possess as few rules and variables as possible. He claims that populating a game with many of these ends up leading to 'emergent behaviour' which can be quite fun.

Summary 2.5

The platforming game to be produced needs to take into account what is ideal for its user and must be enjoyable for them to play. It will need to be made in a system that is accessible to the users and provides the best possible experience of play. Whilst the game itself is intended to be unique and different to what is already out there, it is not the intention to reinvent the wheel, so should strongly consider building on concepts that have been used before and taking them in a different direction.

Product Design – Section 3

Design Introduction 3.1

This section of the project is intended to lay out what features are intended to be added to the level and the basic design principles that need to be considered when designing a puzzle platformer based on the work of previous developers in similar genres and what people have come to expect as a result of these past examples.

Concept Design Requirements 3.2

The game will be a platformer so the main focuses will need to be the player characters mobility as they move around and jump through the air as they will need to be in the air jumping onto and between platforms and obstacles for large amounts of the game. Towards this end there should be priority for allowing the player to control character movement whilst airborne instead of locking them in one speed and direction. Another technique would be to use ‘abilities’ designed to improve the characters mobility, such as sprinting, dashes or double jumps that allow the player more variation and control of how they move around the area. It is important that the player character movement is smooth and comfortable as any janky or unnatural controls could disrupt the player and their experience of the gameplay and bring the overall quality down.

The second design focus for a platforming game is the map design. It will need to provide a varied number of obstacles that are designed to challenge the player as they try to complete the level in the fastest time they can. The most common examples of platforming puzzles are usually floating platforms that become unstable when the player jumps onto them and makes them fall, which can be added to a variety of situations in a game. Another common platformer feature would be moving platforms that the player need to time correctly to land on and continue on their path over large pits that they can’t travel any other way.

The game is also going to include puzzles so it will need to have a series of different riddles or challenges for the player to resolve and these need to be thoroughly designed so they can be solved but are not too easy. A few common examples of in game puzzles include mazes that the player will need to navigate to find the correct exit in order to continue along their way, which is always something that can be added both within the map design and as the overall map design. Other examples of puzzles that might often be included into a game would involve a large variety such as buttons in the floor opening different areas, key card puzzles that need to be found to open areas of the game, logic puzzles that need the player to get a correct sequence to open a door and maybe mathematic puzzles that need the player to work out a problem to proceed.

Production on the 3D puzzle platforming game will involve a focus on designing a single stage of the ‘final product’. The stage will have a series of objects that the player is capable of jumping onto and between, some taking them on paths in the wrong direction. The stage will originally be built up using the unreal engine geometry objects with the intention to test how far the player character is capable of jumping and playing around with different setups to see what is possible and try to see just how challenging the game can be made. There will also be a series of maze designs tested to see which provides the most reasonable level of challenge as well as other puzzle designs that are intended to be added.

Game Layout 3.4

The game will start with a menu that would in a final product display controls to the player, levels to choose from and settings such as volume and screen resolution. The menu would be stored into 4 separate areas labelled, Play (plays the game from the start), Levels (Displays completed levels and the time achieved), Settings (Lets the player browse and alter settings about sound, graphics and controls) and Quit (Exits the game).

Map Design 3.5

The overall 'map' of the area that the player will be in will ultimately be a large maze that they are required to navigate in order to reach the end goal. The maze will be filled with pits and walls that prevent the player continuing through means other than the puzzles that the game will be filled with. Any large rooms in the design will be filled with some form of puzzle as well as many of the longer corridors that should challenge players as they try to progress.

Figure 1 shows the initial drawn out designs of the platformer level map

Puzzle Design 3.6

Moving platforms 3.6.1

A series of platforms will move left and right constantly and the player will need to time their jump so they can land on the next platform along. This will be used to cover gaps that the player would be unable to jump across normally and to allow progress further into the game

Falling Platforms 3.6.2

Falling platforms are platforms that will fall after a short period once the player has jumped onto them and after another short period will move back to their original location so the player will be able to try again in the event that they do not complete the puzzle the first time. This puzzle will be used as a way to cover gaps in the same way as moving platforms. Falling platform will also be used to ascend to higher areas by jumping between them before they fall.

Alternating locks 3.6.3

Alternating locks are used by placing a series of coloured buttons on the floor, red, green and blue which will open doors of their corresponding colour. When a red button is pressed, it will lock the blue and green doors and when touching blue it will lock red and green and so on, leaving only one

colour of door open at any time so the player has to use the changes to navigate the puzzle. The puzzle will be used to hide a key card that the player needs to progress and grants them new abilities.



Figure 2 shows the initial drawn out designs and path of the alternating locks

Jumping Puzzle 3.6.4

Jumping puzzles will involve a series of platforms, they may be static, moving or falling and may involve other elements but the basic idea is that the player will have to jump between objects to reach the end. Jumping puzzles will be used heavily throughout the game and will be used at almost every gap or higher ledge as a means for the player character to progress to new areas.

Key cards 3.6.5

Key cards will open doors that prevent the player accessing the rest of the game until they have been found. There will be a series of coloured key cards, similar to the alternating doors and as each key card is picked up it will turn off the door of other colours. There will be in the designed level red, blue and green key cards that will open their corresponding doors. Each key card will also have abilities tied to them and once the player picks them up they will be granted those abilities. They will be hidden at key points around the level and the player's main objective is to collect them and open the way to the end.

Laser Grids 3.6.6

There will be two types of laser in the game, the type that moves up and down, left and right and the type that turns on and off at regular intervals. The moving lasers will need the player to run past, jump over and avoid the laser as if it hits them it will send them back to the start of the level with the timer still running. There is a similar catch with the on and off lasers as they each turn on and off for a certain amount of time allowing the player to pass by without hitting any of the beams. They are used to challenge the players timing skills and evasion in game.

Invisible Bridge 3.6.7

The invisible bridge puzzle is intended to force the player to keep an eye both on where they are stepping to keep on the bridge and on the timing of the lasers that will send them back to the start if they touch them. It is used to cross a gap and challenge the player's awareness and timing abilities

Figure 3 shows the initial drawn out designs of the Invisible Bridge

Moving Walls 3.6.8

The moving walls puzzles involve the player firing a line trace at a wall and triggering it to extend in some fashion that will allow them to progress with the level. The player is supposed to fire at a glowing wall and the wall will extend to create a bridge, raise them upwards or lower them to a specific height. It can also be used to reduce walls which may be blocking the players path and allow them to progress. It will test the player problem solving abilities as they decide which walls will progress them onwards and how they use those walls to do this until they leave the room.

Shrinking Puzzle 3.6.9

The shrinking puzzle reduced the player to a much smaller size so they can pass through holes they would not be able to normally. This can be used in a way, which means they will have to time their jumping as a smaller character to get through gaps in the walls.

Figure 4 shows the initial drawn out designs of the Shrinking Puzzle

Implementation – Section 4

Game Engine Decision 4.1

As originally mentioned in the literary review of the project, Unreal Engine 4 has been viewed as the most ideal game engine for the creation of the puzzle platformer in spite of the strengths of Cryengine and unity. The reasons for this choice include its ease of use and the amount of experience that exists for making a 3D game in unreal as well as the intuitive nature of its inherent programming language, blueprint. This is because it allows for constant development and error warnings perfectly in sync with the games development. The most compelling reasons for using Unreal Engine 4 over others came down to its easy to use level building tools, with BSP or geometry tools allowing the design of large areas in engine with ease and no need to import 3d assets from other sources. Another benefit that made Unreal Engine 4 the best choice for the platformer is the overall familiarity with the controls of the engine and making assets within it that ensured its use for the game.

Programming Language 4.2

The game will be using the Unreal Engine 4 blueprint as its main language, due to its strong functionality with the engine and intuitive use during development. Blueprint provides an effective method of producing a platforming game due to it being largely structured around individual actors within the game and the level itself. These are ideal for a platforming game as it allows code to be easily triggered by overlap of Collision boxes tied to objects as blueprint actors and recognising the ‘player controller’ or ‘player character’.

Player Controller Development 4.3

Having decided that the player character is vital to the success and ‘feel’ of a platforming game it is key to ensure that every movement function works smoothly and allow the player the most possible control in game. The player character blueprint is based on the standard ‘first person example’ which provides basic movement functions and a gun asset that is largely unnecessary so needed altering for the final result. The first priority is the increase of the main character mobility in game and as such the base movement speed of the player character is increased by around two hundred which allows them to travel around at a much faster and thus more enjoyable rate when exploring the level maze.

Double Jump 4.3.1

The second priority is to implement the player characters ability to double jump (Figure 5). ‘Double Jump’ will become available to the player after they have collected the ‘Green Attunement’ which will be mentioned later and as such makes an initial check to see if the player has yet collected it. After this check if the player is found to not have the attunement they will perform a normal jump and releasing the jump key, in this case space bar, will cut their jump shorter slightly. Once they have collected the attunement releasing the space bar will do nothing but pressing it again will cause the player to launch themselves upwards again to reach greater heights and cover greater distances in the air.

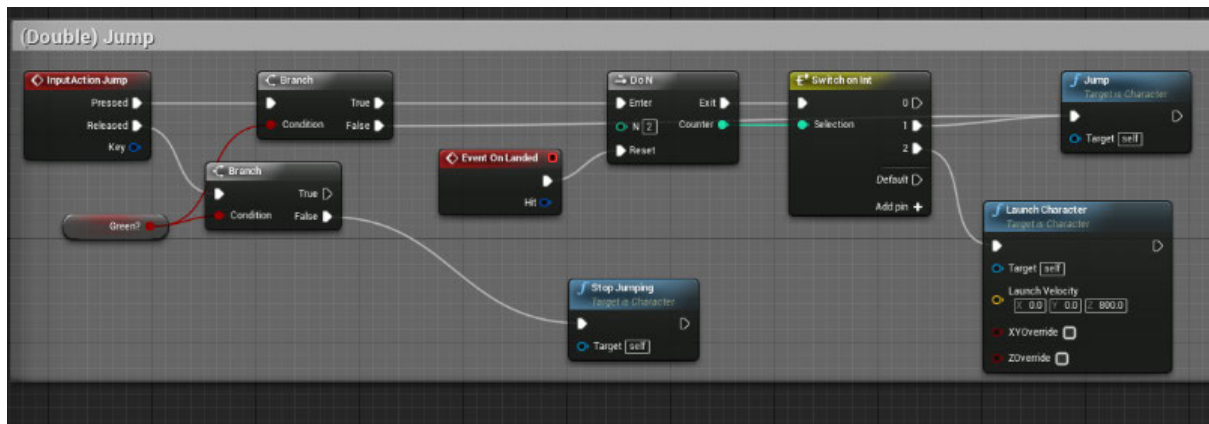


Figure 5

Dash 4.3.2

After implementing the double jump mechanic into the game there still needed to be more ability to control the player character when they are moving around. To this end they can now dash in the direction they are facing when the shift key is pressed and after they have collected the 'Green Attunement' just like with 'Double Jump'. (Figure 6) The dash functions by checking when the shift key is pressed and then checking to see if the green attunement has been picked up and it is not on dash cooldown. After these two checks have been proven true it launched the player character forward in the direction they are facing for 0.03 seconds then stopping them dead. After a 1 second delay the player character is then allowed to dash again. The idea behind the implementation of dash is to allow the player the ability to travel faster both whilst running and in the air as well as allowing different puzzle mechanics that require the longer jumping movement to complete.

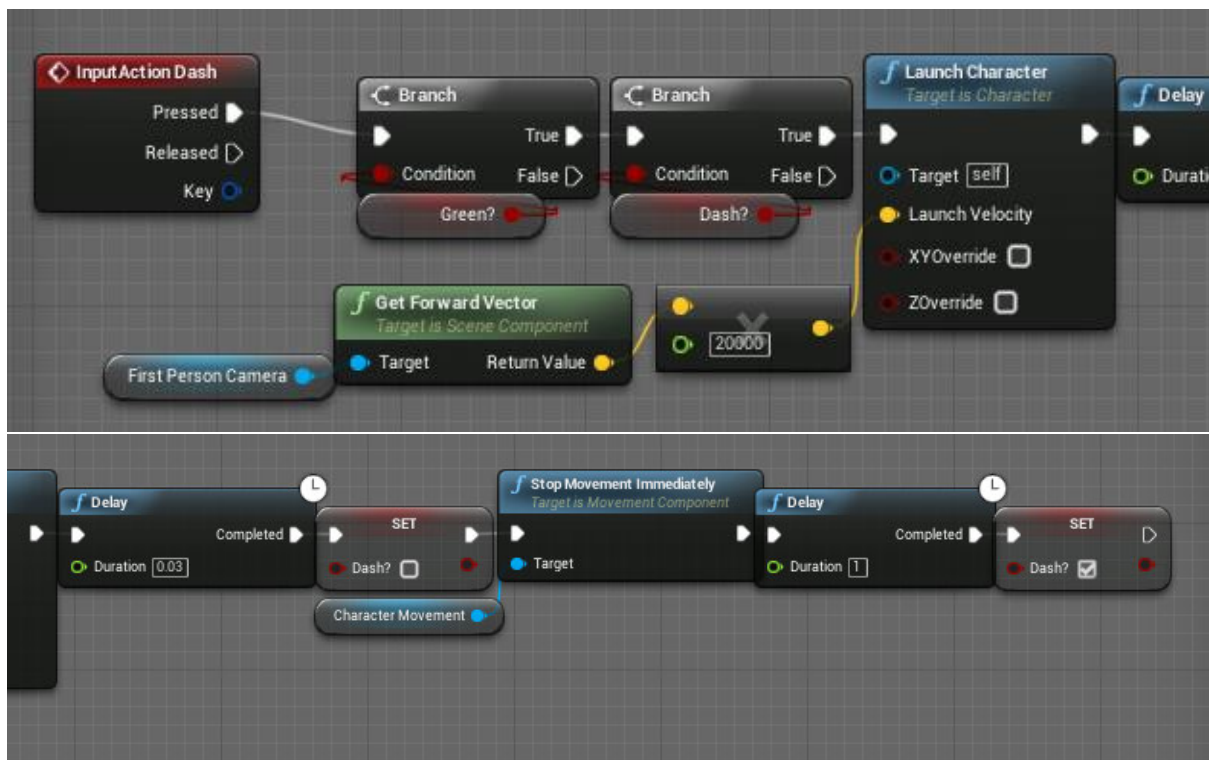


Figure 6

Shrink 4.3.3

After improving the player character mobility in the game it was time to include the rest of the character abilities. First came the ‘Shrink’ feature which when right mouse was pressed will reduce the size of the player to 1/10th the size. The feature is intended to be used so the player can pass through smaller obstacles in the game that they would not normally be able to progress through however it is also supposed to slow the player down whilst being used to prevent constant use. The ‘Shrink’ feature works by checking to see if the player has collected the ‘Blue Attunement’(Figure 7) this time which is found elsewhere in the game to the ‘Green Attunement’. Once it has checked for the blue attunement it reduces the scale of the player to 0.1 in all axes then sets a variable called ‘scale’ to 0.2. This variable is then used as a means of slowing the player movement in the movement blueprint (Figure 8). Once the player releases the right mouse button they return to the original size and reverts to their original speed.

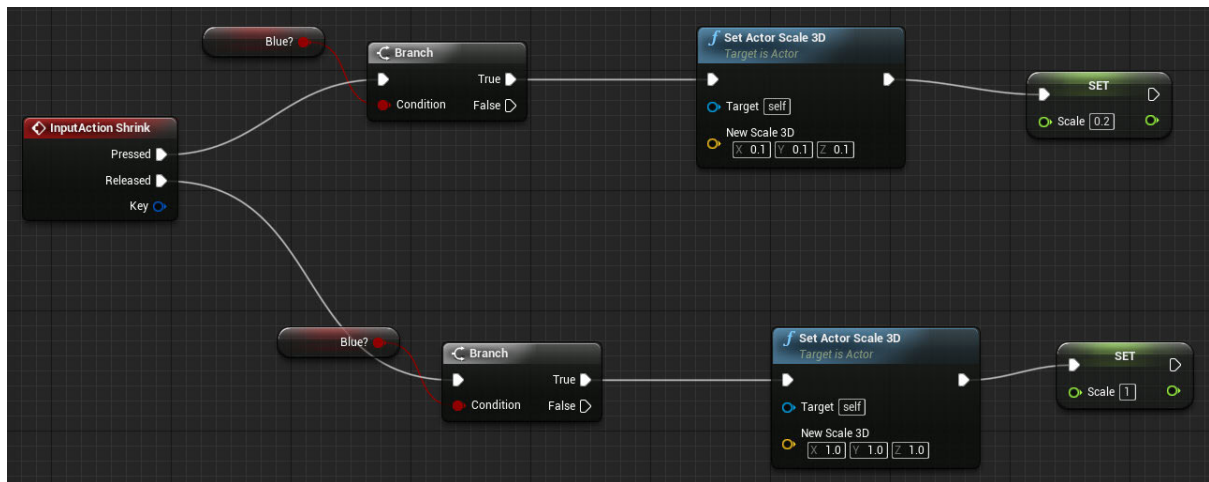


Figure 7

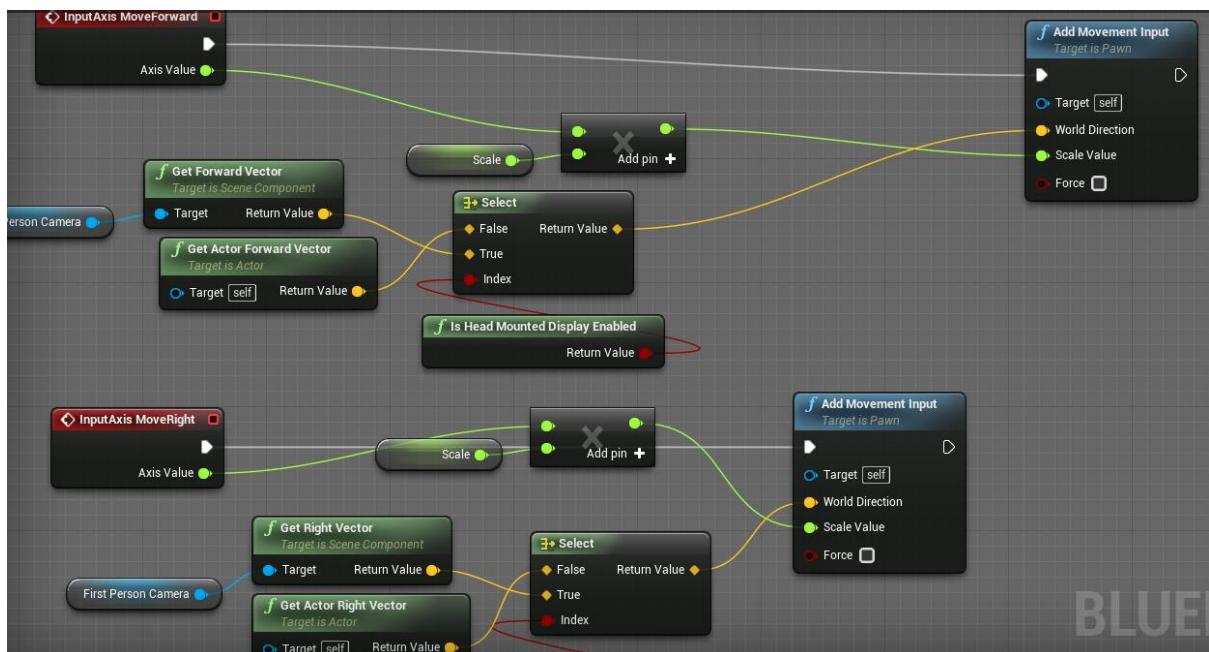


Figure 8

Line Trace Gun 4.3.4

The 'LineTrace Gun' is a feature that finds a point directly ahead of the player where there is an obstacle and returns that point. In this game it is supposed to function as a cause for specific objects at range and as such needs to act as a trigger for other specific blueprints. This ability functions when the player has obtained the 'Red Attunement' and when the left mouse button is pressed (Figure 9). It plays a short animation of recoil on the gun to show something has happened and then triggers the 'Trace' function.

The 'Trace' function revolves around a Line Trace command which gets the position of the players gun and a point far into the distance in order to see if it collides with any object along the way. After it has done this check it then it triggers an 'On Interact' interface blueprint on an actor it hits. If this actor also has an 'On Interface' event trigger then it will also run that actors blueprint (Figure 10).

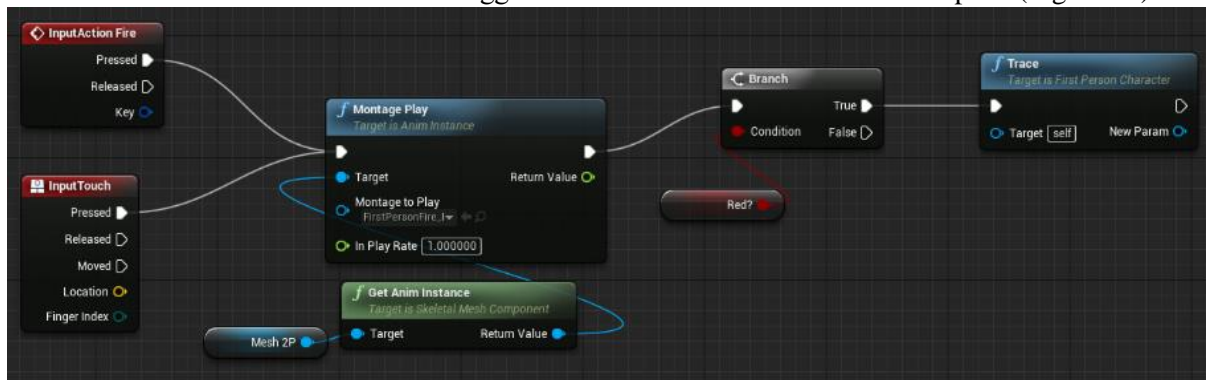


Figure 9

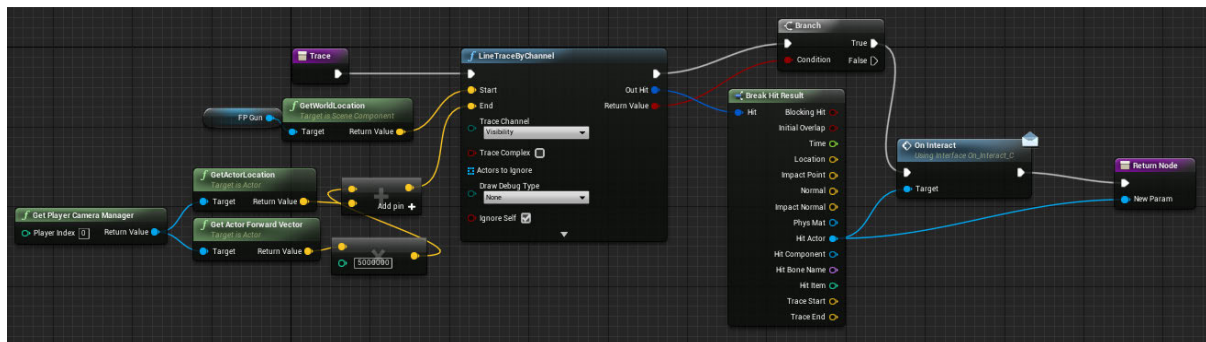


Figure 10

Timer 4.3.5

The games level timer is tied to the player character so it can be read by the UI at the top of the screen. The timer functions by adding one to the second's integer every second and once seconds reaches 60 adding one to the minute's integer and resetting the second's integer. These integers are saved in the character and the User Interface Widget uses them to display the time in minutes and seconds at the top of the screen (Figure 11).

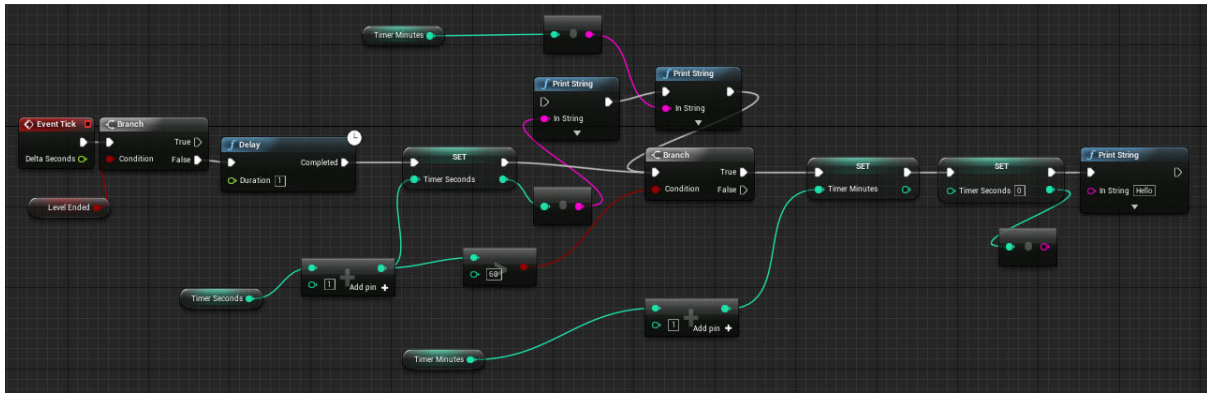


Figure 11

Puzzle Development 4.4

Moving platforms 4.4.1

The moving platforms work by having a set maximum and minimum value they can move between on an axis. When the game begins the system chooses a random value, either 2 or -2 and uses this value to determine the speed of the platform movement and the direction it travels in. The platform works off an event tick which is performed every frame by getting the current location of the platform and moving it either 2 or -2 in the assigned axis every frame until it reaches the maximum or minimum value. Upon reaching its maximum or minimum value it multiplies its directional modifier by -1, reversing the direction of the platform and making go the other way until it reaches the other maximum/ minimum and flips direction again. The platforms are set to loop this way forever and will always uses their current location as a reference never their original position. The reason the direction of the platforms has been made random is to ensure they each space themselves out more than they would if they all started facing the same direction at once and allows all the blueprint to be stored in one actor(Figure 12).

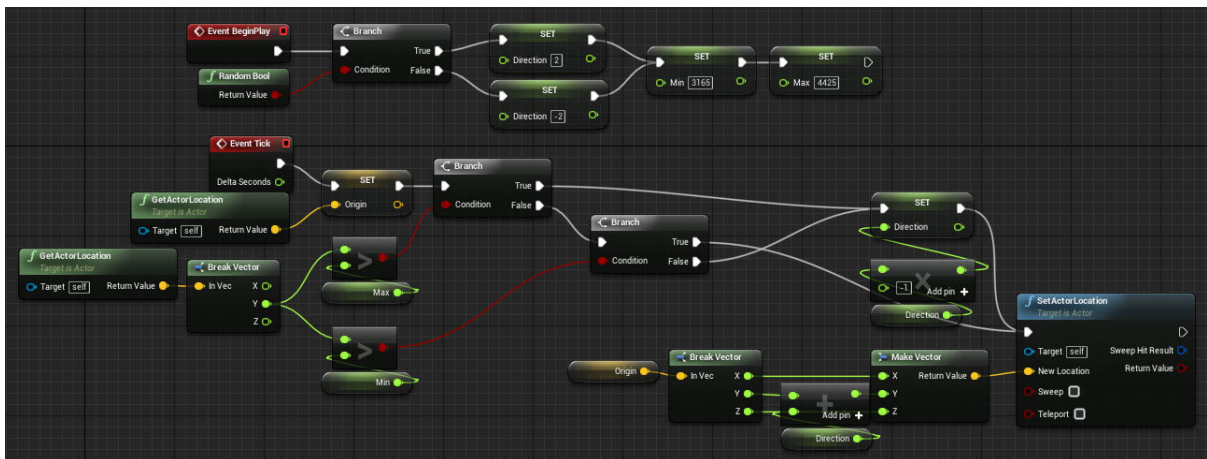


Figure 12

Falling Platforms 4.4.2

The falling platforms blueprint revolves around storing the actor's original location upon the game start and waiting for the player character to overlap with its collision boxes to trigger the rest of the blueprint. When the falling platform blueprint detects overlap it will wait for 1 second before falling by enabling physics on the cube for a second. It will then wait for another 5 seconds before reverting to its original position and allowing the player to jump onto it again, repeating the process. The blueprint stores the actor's original location and rotation once the game begins and resets it to that every time(Figure 13).

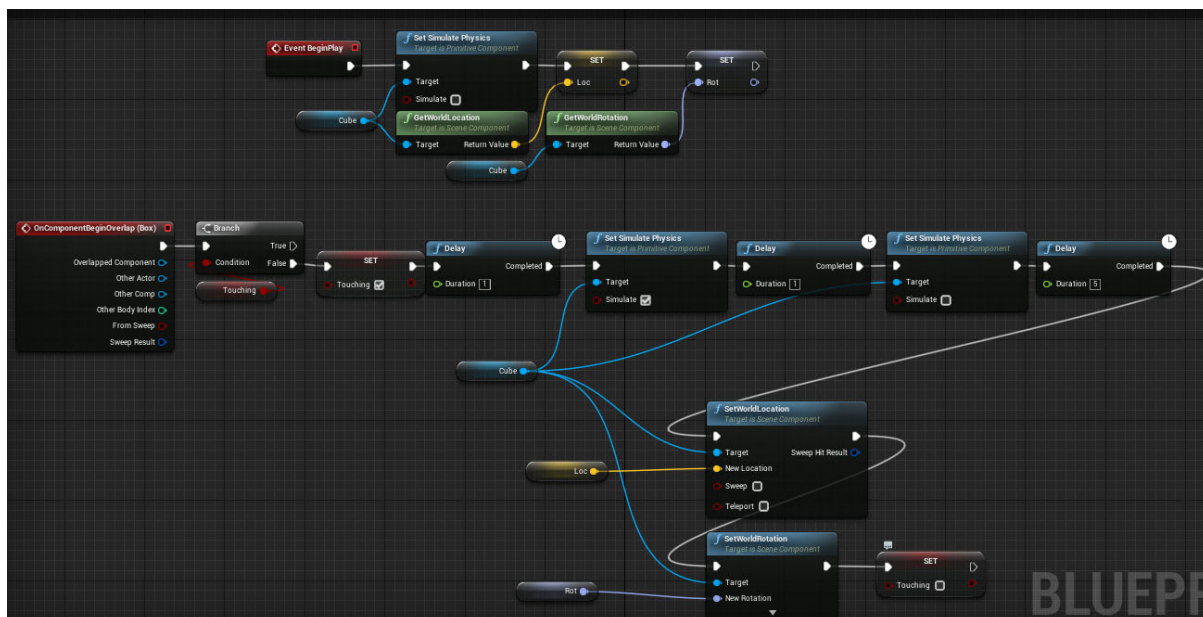


Figure 13

Alternating locks 4.4.3

The (Figure 14) blueprint is contained in the 'button' actors, in this case specifically for the blue button. Its function is to detect when the player is standing on the button and it 'presses itself in' as well as calls three 'event dispatchers', OpenBlue, CloseGreen and CloseRed. The event dispatchers are then picked up by the game's level blueprint (Figure 15). In this blueprint the level receives the dispatcher from a specific button and runs as an event. In the case shown it will open the blue door by setting the variable 'Blue?' to positive and then loop through each 'BlueDoor' blueprint actor and run the 'SetBlue' (Figure 16) function and run the 'ButtonPress' (Figure 17) event. The 'SetBlue' function exists for the sole purpose of setting the 'BlueDoor' variable 'ButtonBlue' to true which then is used in a branch to see if the door should be on or off, in this case it would turn off the door by setting its visibility to false and removing all forms of collision. When another button is pressed and 'BlueButton' becomes false, the blueprint will then make the door visible again and reactivate its collision.

Each different colour of door actor has a similar blueprint which is changed by the event dispatch triggers of the button the player stands on.

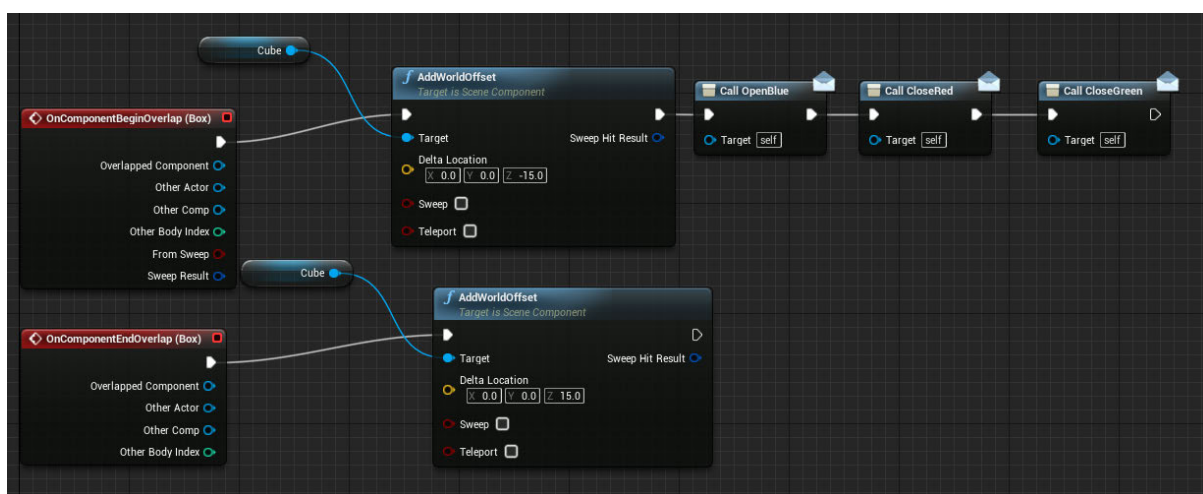


Figure 14

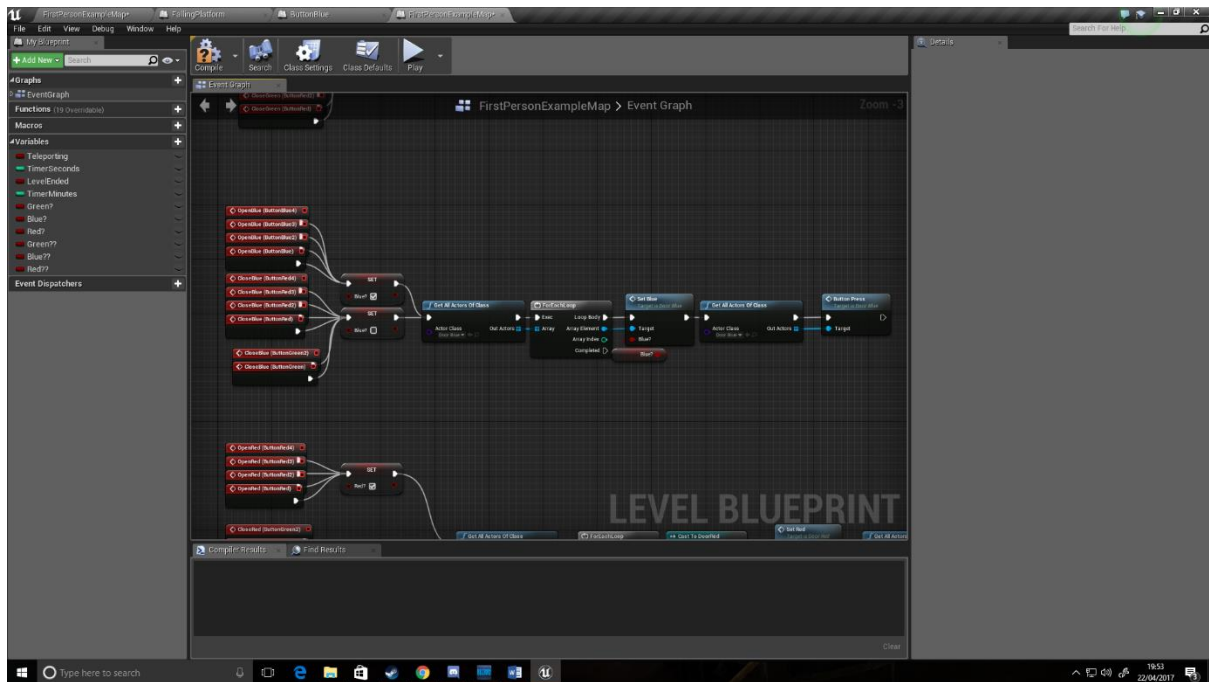


Figure 15



Figure 16

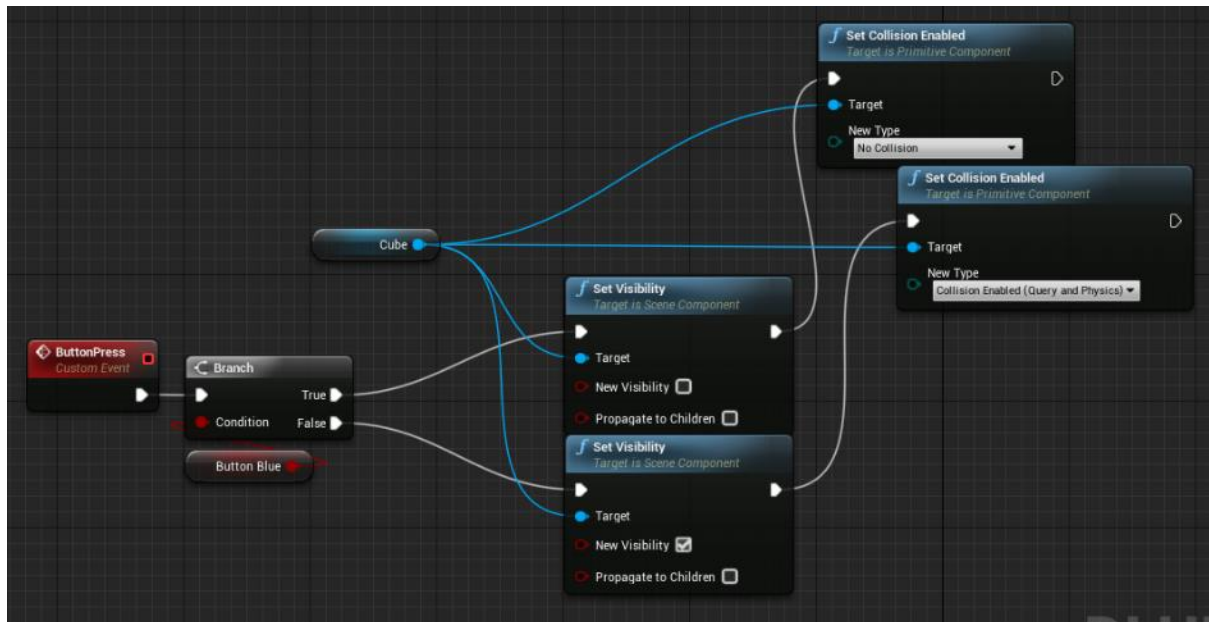


Figure 17

Key cards 4.4.4

The key cards in the game are referred to as 'Blue Attunement', 'Green Attunement' and 'Red Attunement' and when each is collected it locks the other doors in a similar manner to the alternating locks previous, however each 'Attunement' grants the player an ability as well as the ability to open doors, these abilities have been mentioned above. The Key card or 'Attunement' blueprint (Figure 18) works by checking to see if the player is overlapping its collision box and then setting a Boolean

variable called Touching to true. Once this variable is true if the player presses the interact key, in the case 'E' then there will be a check to see if the key has been pressed and the player is touching the object. If this check is true then the blueprint will launch its event dispatchers similar to the alternating locks and will be read in the level blueprint(Figure 19).

The level blueprint receives the dispatchers and then sets the Green?, Red? And Blue? Variables in the player character depending on the 'Attunement' collected. After this it sets the variable Green??. Red?? Or Blue?? In the level blueprint which determines which doors can be opened when the player walks near them. It loops the SetGreenLock(Function 20) function on each 'LockedDoor' actor in the level. This function sets the variable Keycard? to true or false and when the player overlaps with the collision box performs a check on Keycard? and either removes the door or leaves it there until they get the right 'Attunement'(Figure 21).

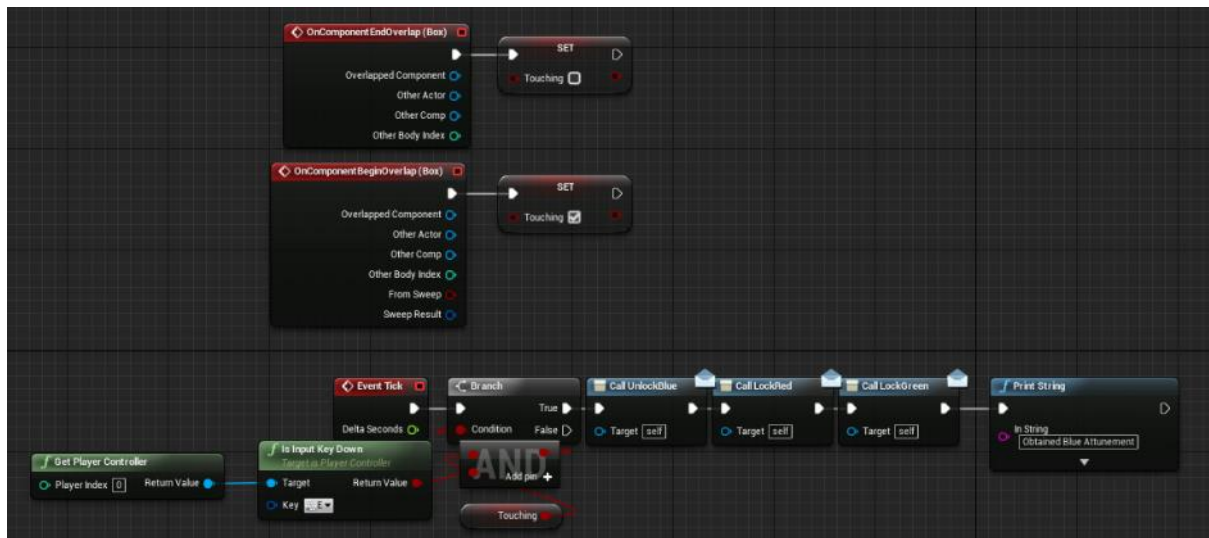


Figure 18

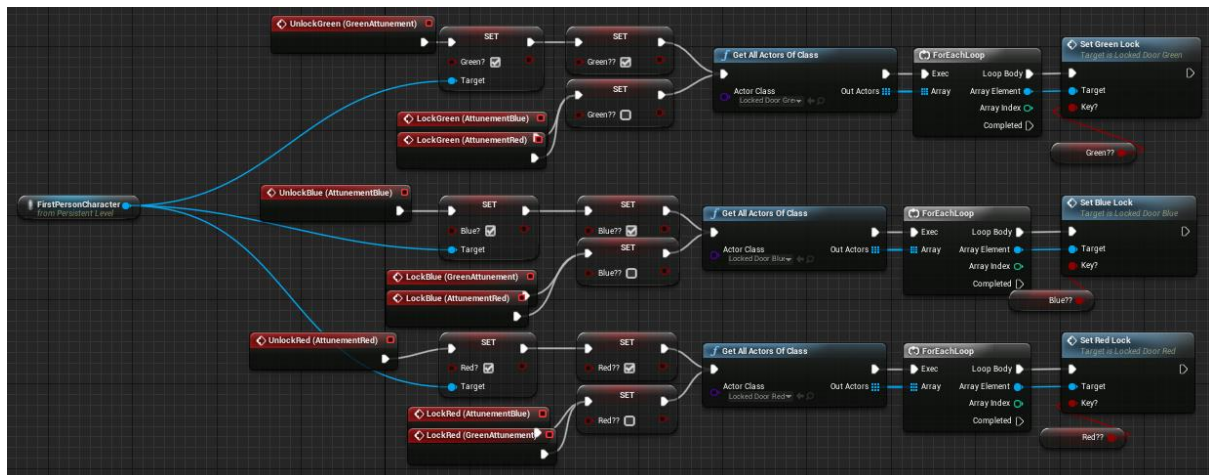


Figure 19



Figure 20

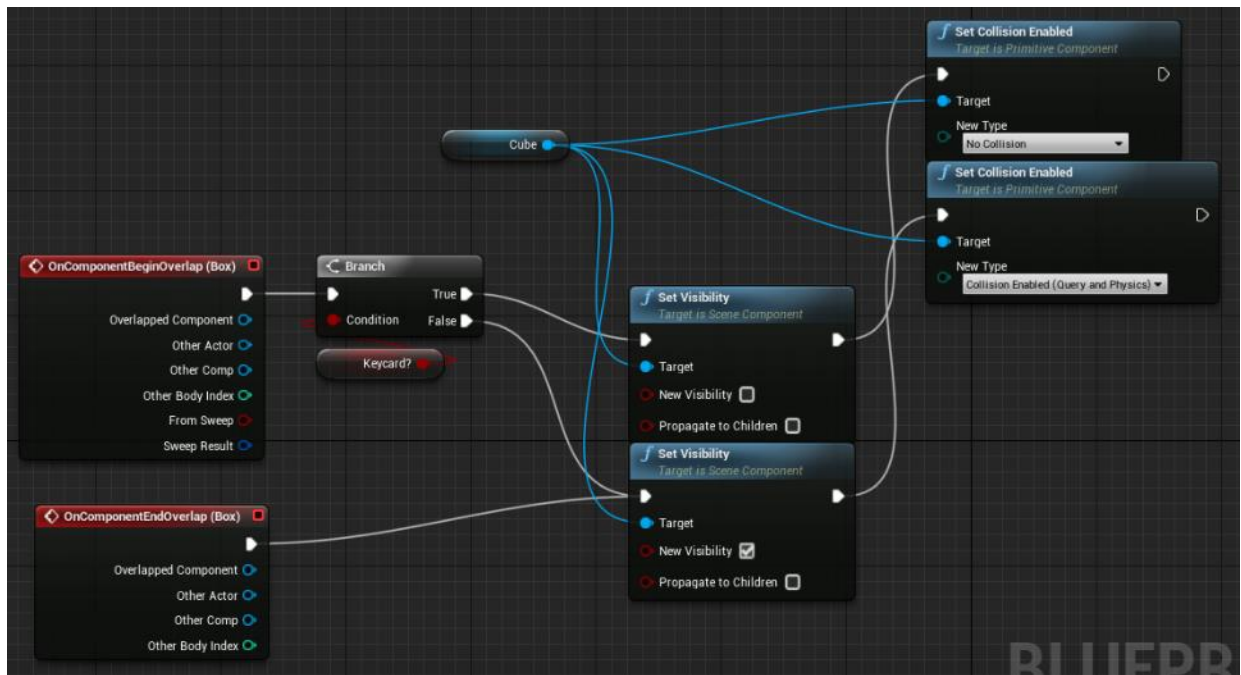


Figure 21

Laser Grids (Moving) 4.4.5

The laser grids use similar movement characteristics to the moving platforms with a set max and minimum value that the laser will travel between based upon its 'direction' value. The direction value for the moving lasers differs from the moving platform as the movement speed will vary between lasers instead of being the same speed just in different starting directions. Beyond this the main key difference to the moving platforms is the 'reset' feature which triggers on player overlap with the actors hit boxes. This works by teleporting the player back to the coordinates they started at should they walk into or be hit by any of the lasers in the grid (Figure 22).

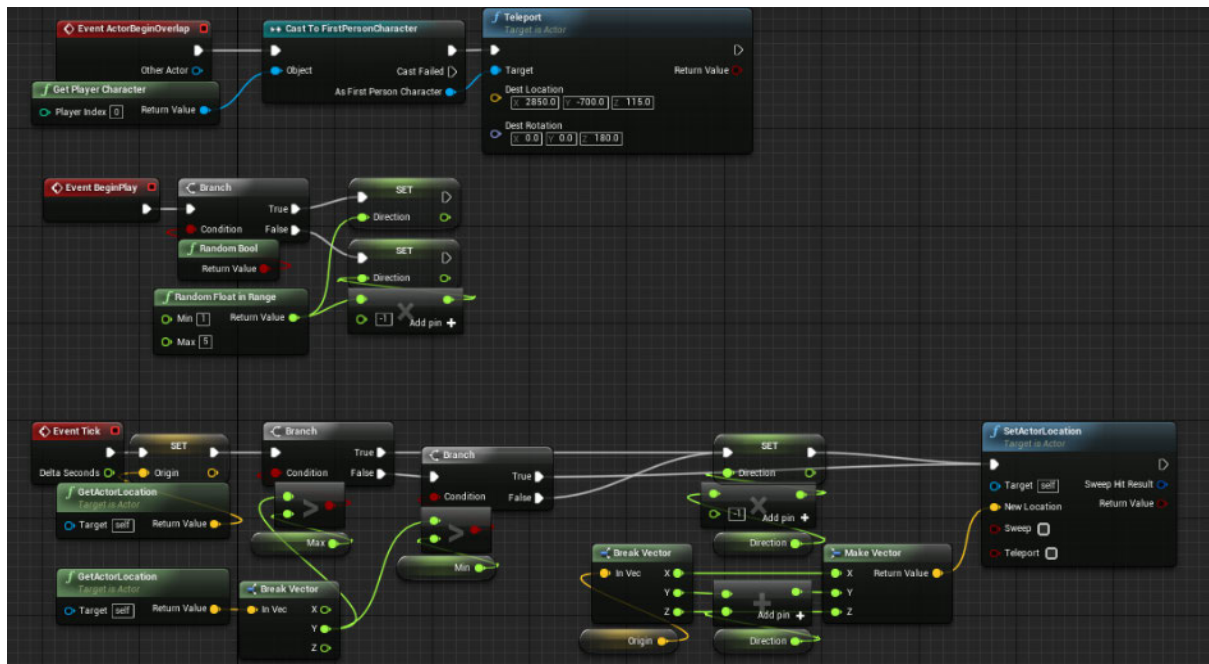


Figure 22

Laser Grid (Timed) 4.4.6

The timed laser adds extra functionality to the moving laser however it removes all the movement blueprint. Instead a new variable has been added called 'Trigger' which determines if the laser will trigger the 'Reset' on the player when they overlap with the blueprint actor. The blueprint functions by turning the laser on and off at a random interval, and then turning it back on. It begins by checking to see if the blueprint is already running preventing any sudden off and on of the laser as the player tries to pass through. If the blueprint is not already running it sets the 'Running' variable to true before the laser is turned on for between 2 and 6 seconds at a time and then off for 3 seconds. After this delay the blueprint makes the visibility of the lasers set to false, hiding them, as well as multiplying the trigger by a negative number which mean the 'Reset' check will not trigger and allows the player through. After the three seconds the laser becomes visible again and the 'Running' variable is set to false, allowing the laser to stay on for the random time again (Figure 23).

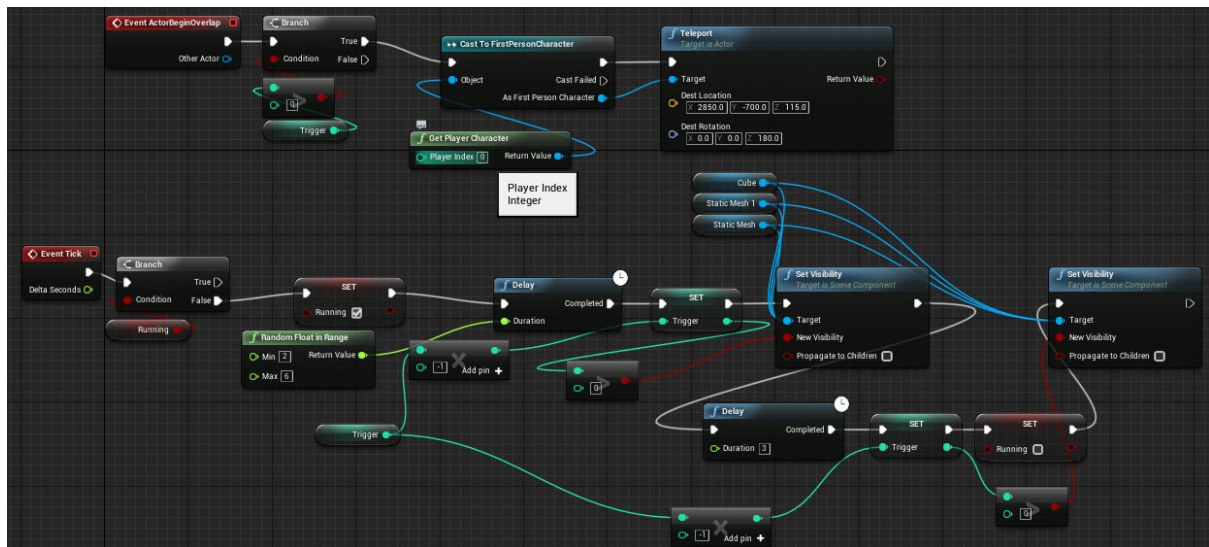


Figure 23

Moving Walls 4.4.7

The moving walls and floor are set to be triggered by the player characters line trace shown above. When the game begins the blueprint ensures the wall is in the correct location, waiting for the player to hit it using the line trace gun. The event 'On Interact' is when the actor is hit by the line trace and it goes into a check to see if the platform is raised already. If it is not raised it plays a timeline function that raises the platform to a specific height over 0.5 seconds and if it was already raised it plays the same timeline in reverse, lowering the platform over 0.5 seconds. (Figure 24)

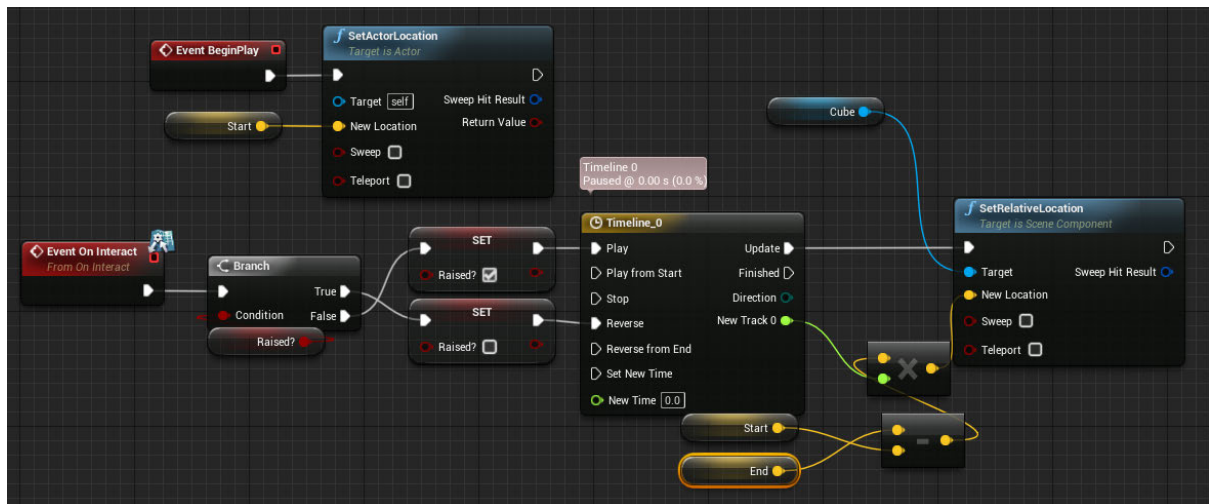


Figure 24

Testing and Evaluation – Section 5

Introduction 5.1

Upon each stage of the games completion it is necessary to preform tests on each of its features, both in terms of the game itself not contain any bugs or glitches within it but also in terms of user enjoyment and opinion of the game itself. Common methods of games design often overlap with more conventional testing methods such as ‘Code Based Testing’ which is structurally similar to white box testing and Component based testing which can vary between both white and black box testing [Redavid and Faird 2015].

Alpha Testing 5.2

Process of Testing 5.2.1

For the alpha testing of the platforming game a few select people were asked to test an early build of the game and express their opinions of the gameplay as they progressed. They were each given an undirected experience of the game to figure out controls and features for themselves and comment on anything they thought could be made better or that was unnecessary.

Testers Opinion 5.2.2

The most common commentary that was noticed as the testers played through the game was in relation to the games current jumping mechanics. The main comment was that they felt as if their character floated for too long in the air with too little player control which they felt impacted the difficulty and feel whilst playing the game.

Another comment that was made was about the lengthy distances between each of the games puzzles. They believed that the travel times were too large and the distances should either be reduced or the player character should be sped up to travel faster.

Overall the play testers I asked seem to think the game was going in the right direction and with a few tweaks and additions made it is looking to be a promising prospect.

Beta Testing 5.3

Process of Beta Testing 5.3.1

For the beta testing a larger number of people were given a directed test of the near completed game to see if they could think of any issues that could be removed or thing that would need to be tweaked. They also served to test if there were any bugs inherent in the system and ensure that the game worked as intended on a series of different machines. After each of the testers had played a run through of the level they were each asked to fill out a questionnaire that would express their feeling of the play through and allow them to comment on anything they thought they needed to. After the initial wave of testing there will be a period of fixing any bugs in the game and then letting tester play the game again and ensure that all of the problems had been removed.

Directed Testing Script 5.3.2

- First open the game
- Press the play button
- Familiarise yourself with the movement controls, left, right, forwards, backwards and jump
- Move towards the first platforming puzzle
- Complete the puzzle and turn left towards the glowing yellow platforms
- Complete the platforming puzzle
- Each door is opened using its corresponding colour button, however that button will close the other doors, completed the puzzle and find the green box
- Touch the green box and press interact

- Test your double jump and dash functions, space twice for double jump and shift for dash
- Return from where you came and pass through the large green wall
- Approach the moving red lasers, pass through at your own pace
- Upon completion turn left and walk into the black box
- You will now need to complete the jumping puzzle using your double jump and dash functions, complete at your own time and collect the blue box at the end
- Touch the blue box and press interact
- Test your new 'Shrink' function, familiarise yourself with using it
- Return through the black box and pass through the blue wall at the far end then into the next black box
- There are small holes in each of the wall, use your new shrink feature to pass through them and reach the end of the maze
- Upon completion of the maze there will be a slightly visible bridge and flashing lasers, familiarise yourself with the patterns and cross in your own time
- Continue onward and turn left onto the moving platforms, cross at your own pace
- There is a red box on a platform in the centre of the room, collect it
- You will now be able to move wall platforms, return to the beginning of the moving platforms and turn left
- In each of the walls, floor and ceiling there are a yellow boxes, shoot one
- Use this new ability to travel through the area and reach the top
- You now have an understanding of each of the games features, complete the last of the game at your own pace.

Questionnaire 5.3.3

1. How professional is the game?
 - Extremely professional
 - Very professional
 - Moderately professional
 - Slightly professional
 - Not at all professional
2. How convenient is the game to use?
 - Extremely convenient
 - Very convenient
 - Moderately convenient
 - Slightly convenient
 - Not at all convenient
3. Did you encounter any issues in the game, jamming, bugs, glitches...?
 - Over 20
 - Between 15 and 19
 - Between 10 and 14
 - Between 5 and 9
 - Below 4
 - None
4. How easy or difficult was the game to use?
 - Impossible
 - Hard

- Okay
 - Easy
5. Do you like the graphics used by the game?
 - A lot
 - Yes
 - They were okay
 - a little
 - No
 6. Were the games 'mechanics' efficient?
 - Always
 - Mostly
 - Sometimes
 - Occasionally
 - Rarely
 7. Overall, are you satisfied with the controls of the game, neither satisfied nor dissatisfied with them, or dissatisfied with them?
 - Extremely satisfied
 - Moderately satisfied
 - Slightly satisfied
 - Neither satisfied nor dissatisfied
 - Slightly dissatisfied
 - Moderately dissatisfied
 - Extremely dissatisfied
 8. Do you like the game, neither like nor dislike it, or dislike it?
 - Like a great deal
 - Like a moderate amount
 - Like a little
 - Neither like nor dislike
 - Dislike a little
 - Dislike a moderate amount
 - Dislike a great deal
 9. Would you recommend this game to someone else?
 - To everyone
 - Yes
 - Maybe
 - A few people
 - No
 10. What changes or improvements would you recommend for the game?

Beta Testing 5.3.4

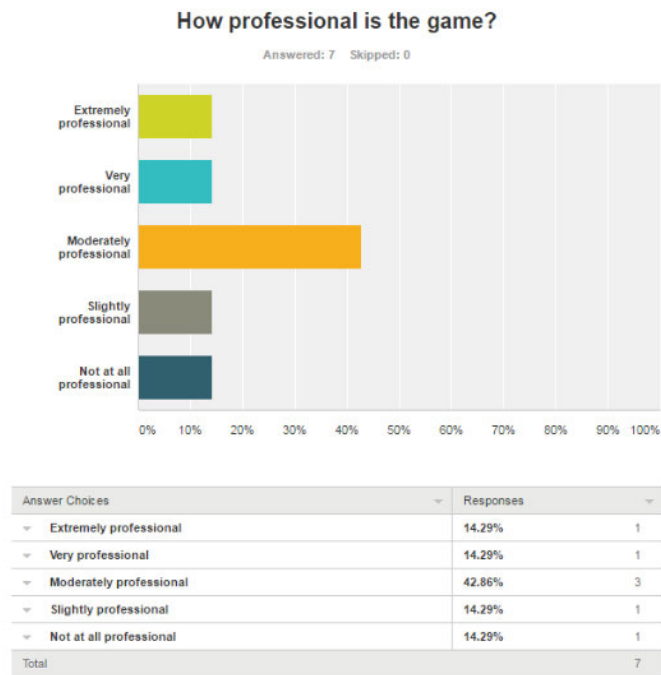


Figure 25

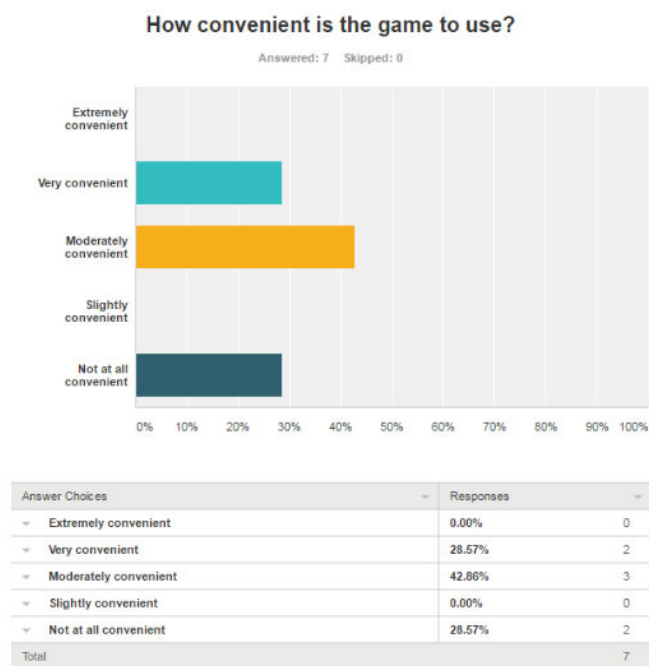


Figure 26

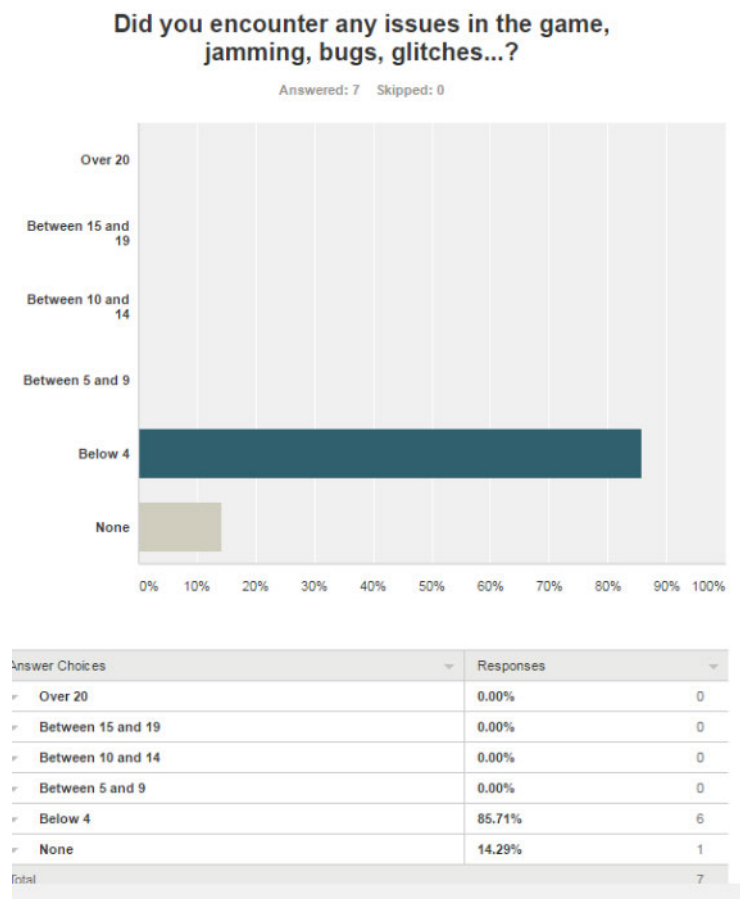


Figure 27

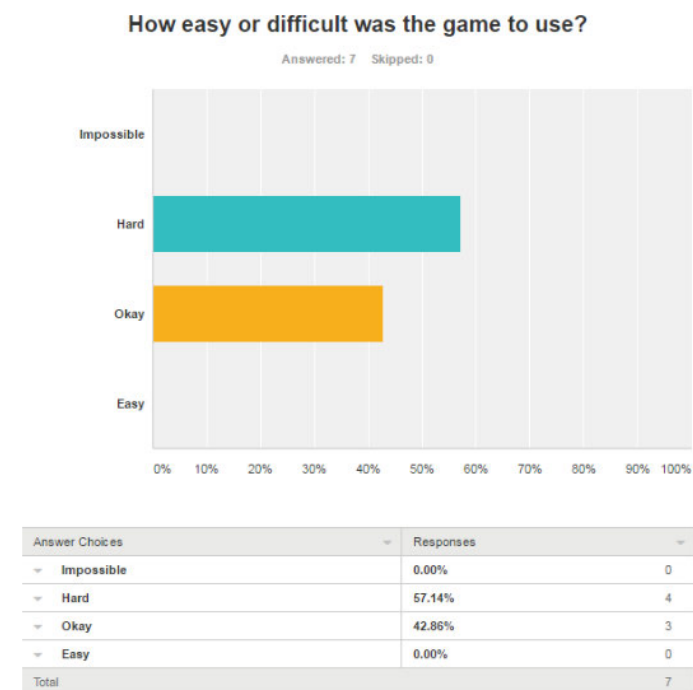


Figure 28

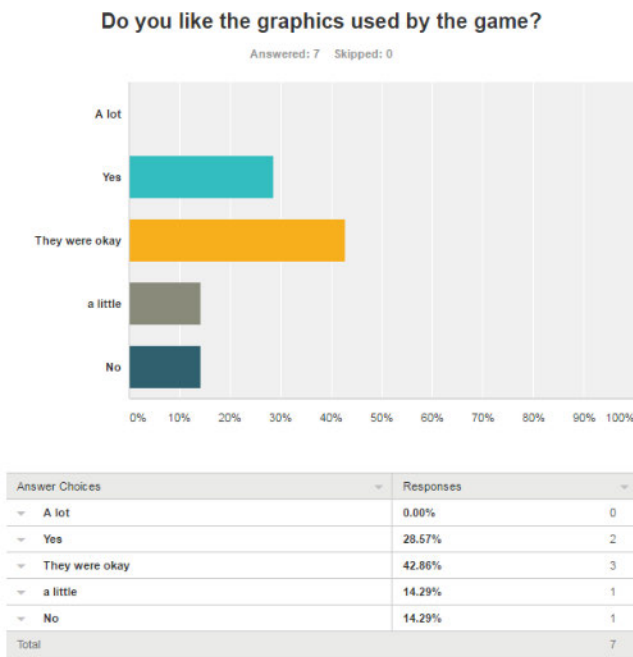


Figure 29

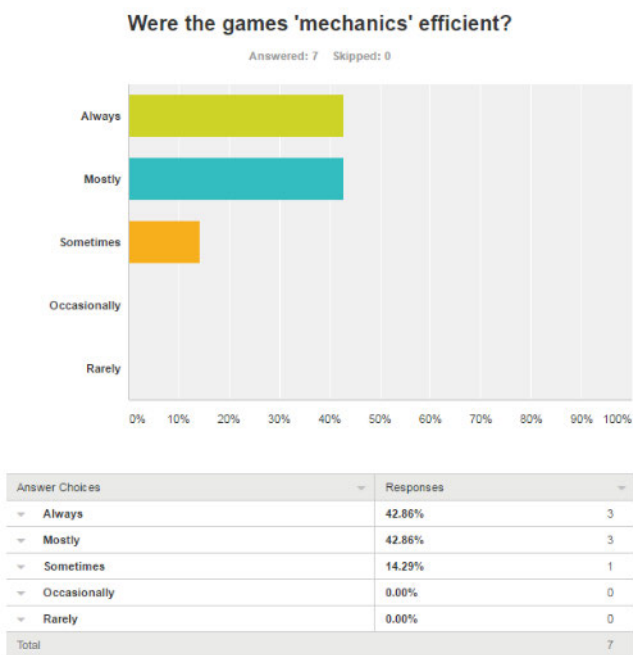
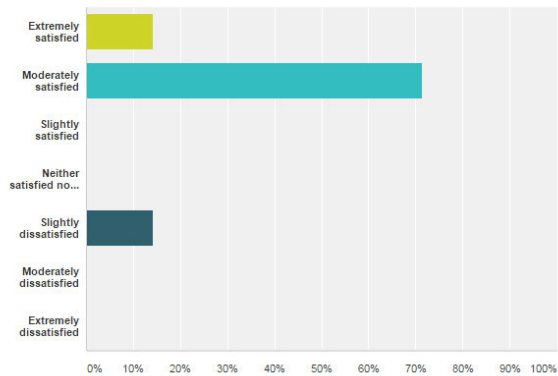


Figure 30

Overall, are you satisfied with the controls of the game, neither satisfied nor dissatisfied with them, or dissatisfied with them?

Answered: 7 Skipped: 0

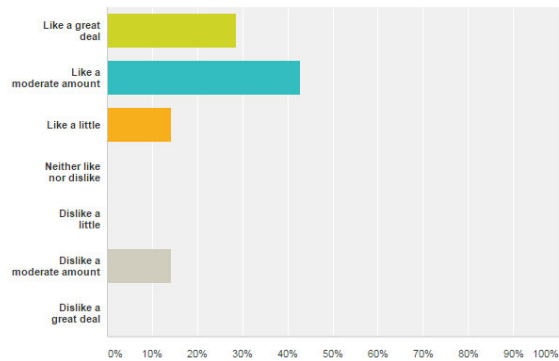


Answer Choices	Responses
Extremely satisfied	14.29% 1
Moderately satisfied	71.43% 5
Slightly satisfied	0.00% 0
Neither satisfied nor dissatisfied	0.00% 0
Slightly dissatisfied	14.29% 1
Moderately dissatisfied	0.00% 0
Extremely dissatisfied	0.00% 0
Total	7

Figure 31

Do you like the game, neither like nor dislike it, or dislike it?

Answered: 7 Skipped: 0



Answer Choices	Responses
Like a great deal	28.57% 2
Like a moderate amount	42.86% 3
Like a little	14.29% 1
Neither like nor dislike	0.00% 0
Dislike a little	0.00% 0
Dislike a moderate amount	14.29% 1
Dislike a great deal	0.00% 0
Total	7

Figure 32

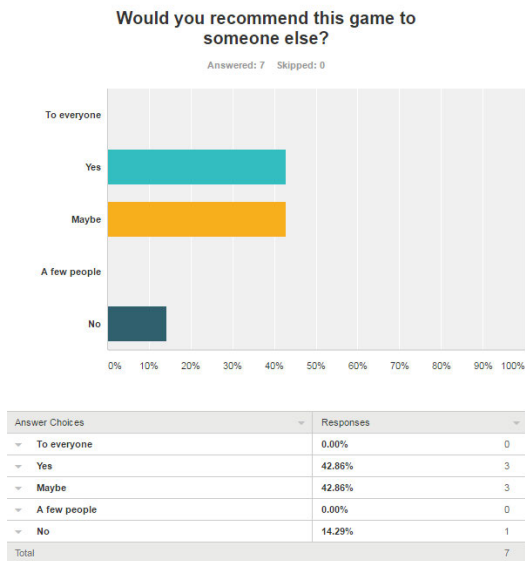


Figure 33

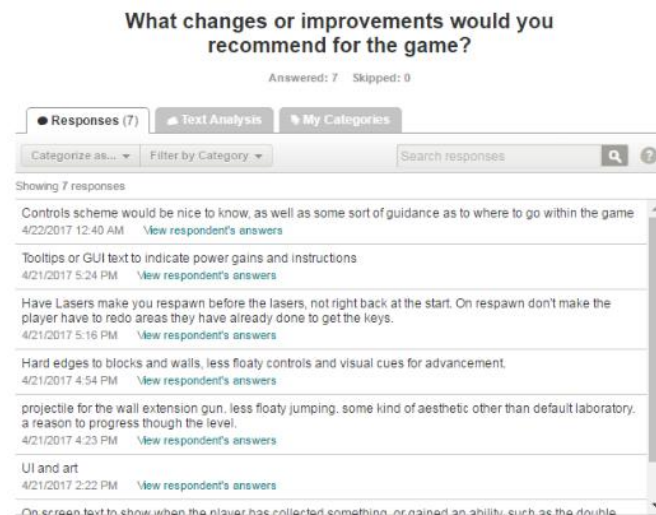


Figure 34

- Controls scheme would be nice to know, as well as some sort of guidance as to where to go within the game
- Tooltips or GUI text to indicate power gains and instructions
- Have Lasers make you respawn before the lasers, not right back at the start. On respawn don't make the player have to redo areas they have already done to get the keys.
- Hard edges to blocks and walls, less floaty controls and visual cues for advancement.
- Projectile for the wall extension gun. Less floaty jumping. Some kind of aesthetic other than default laboratory. A reason to progress through the level.
- UI and art
- On screen text to show when the player has collected something, or gained an ability, such as the double jump.

Beta Results Commentary 5.3.5

The initial results of the beta testing revealed a few problems with some blueprints in the game. The first problem that players would encounter involved the yellow falling platforms respawning at higher

elevations and therefore at a height they could not jump onto, in some cases platforms would also not respawn at all after they had fallen. The second problem involved the roof of one of the building in the game, as they player could use their double jump to pass through the roof of this one building and travel around on top of the building or fall off and fail the game. They also found a problem where the elevators that are used to recover the player if they fall would not function at all sometime but act normally at others. Finally they were unable to finish the level because some of the objects were not spawning that they needed to use in order to progress. These problems were then looked into and changed were made to their blueprint that proved effective at stopping the issues in development testing but would need to go through further user testing to be sure.

Beyond any bugs or glitches they players found they were also asked questions about the look and feel of the game as well as ways in which they felt it could be improved in further development. The most common comments involved that addition of a user interface that would tell them when items had been picked up, when abilities have been gained and how to use them and provided hints on how to progress through the level. There was also comments about the overall 'floaty' feel of the player character as they progressed and varied comments about the textures used in the game.

Research Evaluation 5.4

The research I performed on works similar to my overall aim revealed a lot of information about how players wanted platformers to feel and common mechanics that have been used in platformers and puzzle games alike in the past. The research it performed didn't get into too much specific information of the design of such puzzles or ways to implement balance in the creation of a platformer or and individual puzzle. This lack of research likely hindered future design and implementation as most trials of the game showed that people felt it was very hard, one describing it to me as 'The 'Souls' of platformers'.

Design Evaluation 5.5

The design stages of my project greatly influenced the end product, which is largely as expected, however it also resulted in areas that testing found 'boring' in the initial stages due to lack of prior planning or consideration to add the portals used in the final product as a method of removing long dull periods of inactivity. Beyond this the designs were very helpful in planning the layout of the game and meant when I started implementation on the project the development went quickly and efficiently whilst following the intended plans.

Implementation Evaluation 5.6

Looking back on the development of the project I strongly believe that completing my project in Unreal Engine 4 was the best choice, as it played towards my strengths in regards to my familiarity with the basic use of blueprint. This became something which greatly aided me in developing new knowledge of how different process work in blueprint and being able to create new things and new functionality within this game and future ones I may develop, as well as allowing me to work with and implement a more varied number of features for the game.

The flaws with the implementation of my game are largely related to my initial more cumbersome approaches to individual problems that would result in there being a greater number of separate blueprint actor types used instead of allowing in level alterations to each individual instance of a blueprint actor. There were also initial problems with some of the puzzles involved, involving the platforms that never returned and the elevators not moving as discovered during beta testing, but also other quality of life issues related to laser patterns and the player's lack of control in the air.

However the main reason Unreal Engine 4 proved to be the better choice of game engine for creating a puzzle platformer is due to the use of 'Geometry Objects' for every size shape and situation. The versatility of the basic design tools allowed for a much more rapid development of the level map and permitted much earlier testing to ensure the player character moved around at a desired pace and with

and ideal amount of mobility. It also meant the early testing highlighted issues with the maps design, mainly the long distances between each intended puzzle, which was resolved easily and cleanly by deleting only a few walls and holes in the walls. The Engine also makes altering a singular blueprint easier to view as a whole with its user interface and the intuitive structure of the blueprint itself making reading what it is doing and comparing it with the basic principle of what it should be doing far simpler for me.

Time Management Evaluation 5.7

The time management of the project largely worked fine for the implementation of the final product. This meant that post completion of the level map blueprint and the addition of each separate puzzle there was plenty of time for the testing stages of the game and bug fixes that the initial beta testing revealed in both observations and through questionnaires. The timing was not perfect however as there was little time to do any major quality of life changes that were suggested by the second stage of beta testing, meaning the final product did not end up as suitable for the players as it could have been. On the other hand over the course of the project each of the objectives have been achieved and the end result is a fully functional and according to the beta testing results enjoyable 3D platforming game with puzzles that need to be solved by the players.

Conclusion – Section 6

Closing Remarks 6.1

The project has produced a working prototype level for a puzzle platforming game that managed to both accomplish all the objective of the project set out at the start and also proved what can be combined together to produce a working game.

Overall, my experience of working on the puzzle platforming game has been both enjoyable and educational. My work in the unreal engine and with blueprint has helped me to understand how to better use the software and given me a broader perspective on how to approach problems within it that I may encounter in the future. I have also gained a better understanding of the features within the engine and the manner in which they are used or how then can be used to resolve a large variety of issues. It has been fun to steadily build up a full game from scratch and see it at the end both working and enjoyed by a majority of the testers who played it.

Appendix

Terms of Reference

Puzzle Platformer

1.1. Course-Specific Learning Outcomes

- To study the history of computer games, game genres, game structures and game design principles and to use the skills acquired to specify and evaluate new game applications;
- To become knowledgeable in the use and development of computer graphics software/game middleware tools and to be able to apply this knowledge to the implementation of real-time interactive systems;
- To learn structured approaches to computer programming;
- To learn how the theories and techniques of behavioural systems can be used to enhance the playability and sophistication of computer games;
- To study a range of mathematical tools and techniques and to be able to use these, in particular, to solve problems in the area of game modelling and animation; and
- To gain an appreciation of the multidisciplinary environment in which commercial games are designed and produced and to acquire skills in project management and team working.

1.2. Project Background

There are already many platforming games on the market with a variety of objectives and have been for a very long period of time in both 2d and 3d. A few examples of long running popular platforming games would be the Mario (1985) and Sonic the Hedgehog (1991) series which have created games of both forms. The general theme of the two is to keep moving forwards and collect items which is where the intention for the game changes from the 'normal' structure as it is to be intended for the player character to keep moving upwards in an extended jumping puzzle with other puzzles added to increase the challenge. In addition to there being many platformer games there are also a large variation of puzzle games, with one of the most well-known being the Portal (2007) series which involves using a 'portal' to manipulate the areas and reach the goal.

The inherent challenge in the creation of a puzzle platformer game comes from the difficulty of its puzzles and how to make them challenging yet enjoyable for the player. There also arises the complication of designing a varied number of puzzles and fitting them into the design. Other challenges are likely to arise from the development of the game itself including the coding used and the addition of any potential extras such as 3D virtual reality implementation. Overall there will need to be plenty of research into puzzles that would be able to fit with the games theme and a few tests to see if the users are able to complete any puzzles that are presented.

1.3. Aim

To create a 3D game where the player is required to navigate a series of obstacles such as jumping puzzles and mazes to reach a specific goal.

1.4. Objectives

The game will need to have a few objectives which will determine separate stages of its design and development and will be used to review the progress that has so far been made.

1.4.1. Game Engine Review

Practice development of the project in different game engines to determine which is the best suited for the project. This will result in less risk of the entire project being wasted if there is a problem with the game engine later into development that is vitally necessary to resolve.

1.4.2. Product Design

Produce a complete collection of potential designs for the games layout and aesthetic and review them in detail with possible public recommendations and alterations. Ensure each aspect of the game has been planned ahead to reduce delays in the system development

1.4.3. Development of the Game

The game and its features shall be implemented from the design segment to involve a moving character and the ability to traverse the puzzles involved.

1.4.4. Thorough Testing Period

Complete a thorough test of the games controls and boundaries to ensure that there will be no issues for the player and the game runs without crashing. Ensure that each of the games core functions as well as its inputs and boundaries work as intended.

1.4.5. Resolution of Problems

Use the games testing data to fix problems that will cause issues for the player or result in serious crashes during use. Retest each of the altered segments as well as the rest of the game after each change to ensure the issues have been resolved.

1.4. Problems

The main problems in the development of the game are likely to arise from the work on the game itself, as there may be problems with the code used that will need to be reviewed to ensure any errors are removed, in addition the implementation of other technology such as 3D virtual reality equipment will add another layer of complexity that will be a highly involved process in the games development and basic mechanics. There will also be delays involved in the designing of each puzzle and area as each section may become too difficult for the player or alternatively provide no challenge.

1.5 Timetable and deliverables

Week	Date of Monday	Activity
0	19/09/16	Terms of reference
1	26/09/16	Terms of reference
2	03/10/16	Terms of reference
3	10/10/16	Terms of reference
4	17/10/16	Terms of reference
5	24/10/16	Ethics form
6	31/10/16	Literature Survey
7	07/11/16	Literature Survey
8	14/11/16	Literature Survey
9	21/11/16	Literature Survey
1	28/11/16	Evaluation Design
11	05/12/16	Evaluation Design
12	12/12/16	Evaluation Design
13	09/01/17	Evaluation Design
14	16/01/17	Evaluation Design
15	23/01/17	Evaluation Design
16	30/01/17	Evaluation Design
17	06/02/17	Report Outline
18	13/02/17	Report Outline
19	20/02/17	Report Outline
20	27/02/17	Draft Slides
21	06/03/17	Draft Slides
22	13/03/17	Practice Presentation
23	20/03/17	
24	27/03/17	
25	24/04/17	Report and Product submitted

1.6 Required Resources

The game will require the use of a computer with sufficiently high enough specifications to run one of the ideal game engines, to be chosen from unity, unreal and cry-engine. The game will also need any other additions that may be included with the final product such as 3D virtual reality apparatus, however this software will largely be unnecessary for the development and more for the testing

Ethics Form

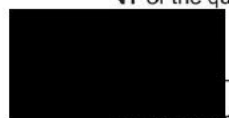


ed to be warned of pain or mild discomfort resulting from the		
or repetitive testing or does it include a	<input type="checkbox"/>	<input checked="" type="checkbox"/>
s a physical intervention and here to read how any prolonged d for participant wellbeing and safety		
e study without their knowledge and include a justification.	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ere research may be carried out without informed consent		
than reasonable expenses and d to participants?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
nt for participants		
etween the researcher(s) and the nsidered? For instance, a lecturer i manager interviewing her/his staff?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
disting power relationships need to be dealt with in research		
sments for each of the procedures that	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ing place outside of the UK?	<input type="checkbox"/>	<input checked="" type="checkbox"/>
the following security sensitive	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ecurity call rity clearances groups nsitive Information Form	<input type="checkbox"/>	<input checked="" type="checkbox"/>

al will apply to the current project protocol and timeframe stated. If
to review the ethical consideration(s) and this will include completion

st

NY or the questions 5a – 17 then they must complete

 Date: 25/10/16 (DD/MM/YY)

Project is (please check the appropriate box):

ues requiring further consideration and the project can

s requiring further consideration and will refer the project
Officer.

_____ Date: _____ (DD/MM/YY)

_____ Position:
r UG and PG Taught/ PGRs RD1 Scrutiniser/

Bibliography

[Masters 16] <http://blog.digitaltutors.com/unity-udk-cryengine-game-engine-choose/>

(Seen on 16/11/16)

[Towel 16] <http://www.gamesradar.com/best-platform-games-ever-arent-mario/>

(Seen on 20/11/16)

[Klappenbach 16] <https://www.lifewire.com/what-is-a-platform-game-812371>

(Seen on 20/11/16)

[Racketboy 16] <http://www.racketboy.com/retro/platformers/platforming-games-101-all-you-need-to-know>

(Seen on 20/11/16)

[Tulleken 16] <http://devmag.org.za/2011/04/16/how-are-puzzle-games-designed-introduction/>

(Seen on 23/11/16)

[Puzzle 16] <http://tvtropes.org/pmwiki/pmwiki.php/Main/StockVideoGamePuzzle>

(Seen on 23/11/16)

[Koster 16] <http://www.raphkoster.com/2012/01/25/fun-vs-features/>

(Seen on 24/11/16)

[Redavid and Faird 2015]. *An Overview of Game Testing Techniques*. Paper available at:

http://www.idt.mdh.se/kurser/ct3340/ht11/MINICONFERENCE/FinalPapers/ircse11_submission_15.pdf

Nintendo R&D4. (1985) Super Mario Bros. Video Game. Family Computer. Nintendo

Sonic Team. (1991) Sonic the Hedgehog. Video Game. Sega Megadrive. Sega

Valve Corporation. (2007) Portal. Video Game. Microsoft Windows and Xbox. Valve Corporation