

Twin Darkness Technical Report

Student Name: Jamie Mulvaney

Student Number: x12358831

Email address: x12358831@student.ncirl.ie or jamiex304@hotmail.com

Degree Programme Name: BSc (Hons) in Computing

Specialisation: Gaming and Multimedia

National College of Ireland

Project Submission Sheet – 2015/2016

School of Computing

Student Name:

Student ID:

Programme: **Year:**

Module:

Lecturer:

Submission Due

Date:

Project Title:

Word Count:

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:

Date:

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. **Please do not bind projects or place in covers unless specifically requested.**
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Twin Darkness Technical Report

Contents

Executive Summary	4
Introduction to Twin Darkness	5
Initial Game Concept	5
Background & Research	5
Full Game Concept Breakdown	11
Aims	13
Technologies	14
Hardware	14
Software	15
User Requirements Definition	16
Requirements Specification	16
Functional requirements	17
Requirement 1: Play Game/Movement	17
Requirement 2: Save Game	19
Requirement 3: Load Game	21
Requirement 4: Exit Game	23
Requirement 5: Modify In-game Settings	25
Requirement 6: Complete Prologue Chapter	27
Requirement 7: Interact with in-game items/systems	29
Requirement 8: VR Setup/Interaction	31
Non-Functional Requirements	33
Performance/Response time requirement	33
Availability requirement	33
Security requirement	34
Reliability requirement	34
Maintainability requirement	34
Extendibility requirement	34
Reusability requirement	34
Implementation	35
Game Systems Implementation	35
Menu level implementation	35
Main menu widget implementation	36
Chapter Select implementation	37

Twin Darkness Technical Report

<i>Graphical and Performance Options implementation</i>	<i>39</i>
<i>Pause System & Pause Menu implementation</i>	<i>44</i>
<i>Save & Load Backend System implementation</i>	<i>47</i>
<i>In-Game Systems Implementation</i>	<i>49</i>
<i>Doors implementation</i>	<i>49</i>
<i>Readable Objects implementation</i>	<i>50</i>
<i>Interactive Objects implementation.....</i>	<i>51</i>
<i>Artificial Intelligence Implementation.....</i>	<i>52</i>
<i>Aries implementation</i>	<i>53</i>
<i>Main Character Implementation</i>	<i>57</i>
<i>Character Animation Blueprint implementation.....</i>	<i>57</i>
<i>Movement Input Controls & Setup implementation</i>	<i>61</i>
<i>Sprint System implementation</i>	<i>64</i>
<i>Passive Power System implementation</i>	<i>68</i>
<i>Active Power System implementation</i>	<i>70</i>
<i>Death System implementation.....</i>	<i>73</i>
<i>User Features Implementation</i>	<i>75</i>
<i>Virtual Reality implementation</i>	<i>75</i>
<i>Steam Network implementation</i>	<i>77</i>
<i>Steam Achievements implementation</i>	<i>78</i>
<i>Testing.....</i>	<i>79</i>
<i>Overview</i>	<i>79</i>
<i>Usability Testing.....</i>	<i>79</i>
<i>Unit & Integration Testing</i>	<i>82</i>
<i>Customer Testing</i>	<i>85</i>
<i>Graphical User Interface (GUI).....</i>	<i>89</i>
<i>Main Menu.....</i>	<i>89</i>
<i>Options Menu.....</i>	<i>89</i>
<i>Chapter Select Menu</i>	<i>90</i>
<i>Base Chapter Select.....</i>	<i>90</i>
<i>Chosen Chapter Select.....</i>	<i>91</i>
<i>Pause Menu.....</i>	<i>91</i>
<i>Death Screen Menu.....</i>	<i>92</i>
<i>In-Game Objects.....</i>	<i>92</i>

Twin Darkness Technical Report

<i>Readable Object Screen</i>	<i>92</i>
<i>Interactive Object Screen</i>	<i>93</i>
<i>Main Character & Environmental Previews</i>	<i>94</i>
<i>Conclusions.....</i>	<i>96</i>
<i>Milestones and Hurdles.....</i>	<i>96</i>
<i>Further Development & Research</i>	<i>97</i>
<i>Appendix</i>	<i>98</i>
<i>References</i>	<i>98</i>
<i>Reflective Monthly Journals</i>	<i>99</i>
<i>September Journal</i>	<i>99</i>
<i>October Journal</i>	<i>101</i>
<i>November Journal</i>	<i>105</i>
<i>December Journal</i>	<i>108</i>
<i>January Journal</i>	<i>113</i>
<i>February Journal</i>	<i>122</i>
<i>March Journal</i>	<i>126</i>
<i>April Journal</i>	<i>130</i>

Twin Darkness Technical Report

Executive Summary

Twin Darkness, uses Unreal Engine 4, voted #1 best game engine of 2015 to deliver outstanding graphics and performance.

Built from the ground to use Oculus Rift so the player will be fully immersed in this shadowy episodic thriller.

Twin Darkness is a fully dynamic engaging environment, loaded with key features:

- Advanced A.I: developed and implemented decision trees with advanced audio and sight detection branches ensuring realistic responses.
- Dynamic Options Menu: ensuring the game can meet the system for the best performance and graphical output.
- Fully engaged 3D environmental assets.
- Integrated Virtual reality: ensuring a perfect immersed gameplay experience for the player.
- Uniquely designed and developed main character, with advanced passive and active ability options ensuring the player is engaged.
- Gameplay/Story designed around the concept of episodic storytelling.

You the player will take the role of Gemini, a young women with a past that is hidden to everyone including herself. As Gemini, you awaken within a facility called Equinox controlled by someone or something truly dark. From there you must uncover the truth, survive what's been done to you and most importantly escape

"Twin Darkness is a game built both for the present and future of the industry and for a genre that needs an injection of new life"

Introduction to Twin Darkness

Initial Game Concept

Twin Darkness is an immersive first person thriller set in a dystopian future. The world is on the edge of all-out War and new weapons are being built. You will take the role of Gemini, a young women with a past that is hidden to everyone including herself. As Gemini, you awaken within a facility called Equinox controlled by someone or something truly dark. From there you must uncover the truth, survive what's been done to you and most importantly escape

"Twin Darkness a game built both for the present and future of the industry"

Background & Research

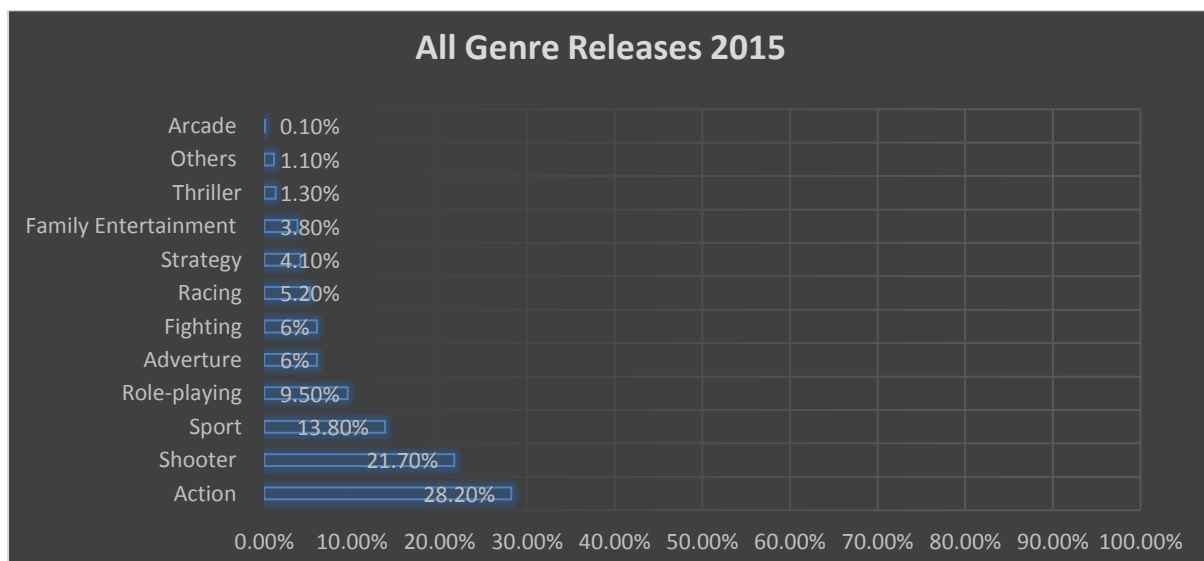
Initial research stage for my project began back in early September, when I began looking into game genres. I started compiling a list of genres from various platforms (*steam, origin and uplay*) from there I used my compiled list and picked a number of genre's that interest me.

The genres I choose for my initial list where:

- ❖ **Role-playing**
- ❖ **Action**
- ❖ **Thriller**
- ❖ **Strategy**
- ❖ **Shooter**

Once I had my list, I moved onto the second stage of research which was investigating the development and releases of these various genres over the past years (*To see which are being neglected and may have an opening in the genre market*). I found a number of sites used to track game statistic such as revenue, releases and projected growth. Using these sites I compiled my first graph which shows game genre releases **across all game platforms** (*PC, Xbox, PS4 etc.*)

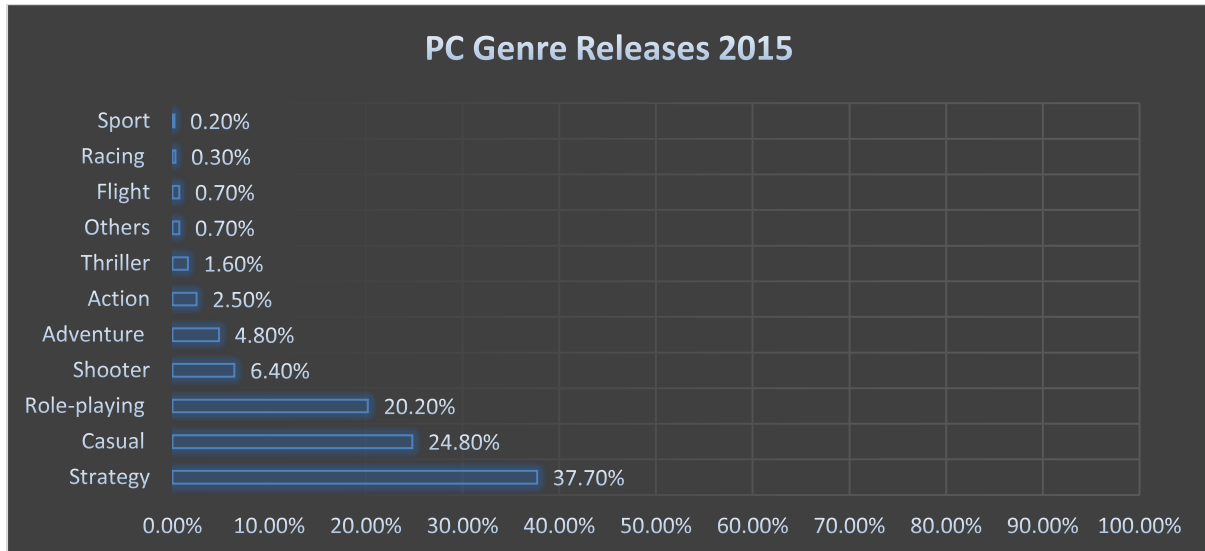
(Graph 1 All Platform Genre Releases)



Twin Darkness Technical Report

Next I moved onto compiling a list of genre releases for just the PC platform (*since it is the main platform I would be developing on*) Using the same set of parameters and the same sites I compiled the second graph which shows the percentage of genre releases for PC.

(Graph 2 PC Platform Genre Releases)



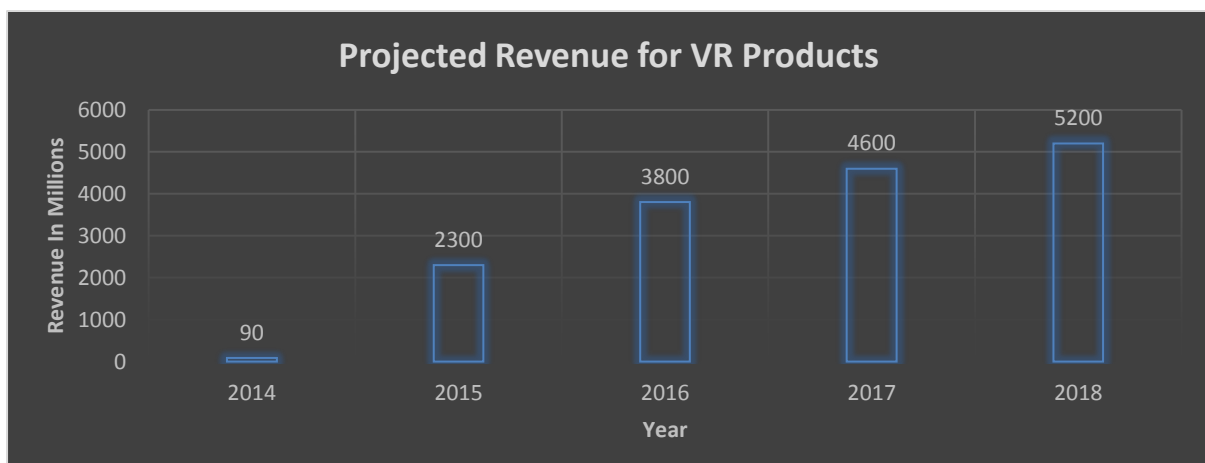
Using graph 1, 2 and my compiled list I narrowed it down based off which genres are currently being flooded with products and have the highest release percentages. After careful examination and study of the graphs I was left with one genre:

➤ **Thriller**

Now that I had my chosen genre it was time to move onto the third stage of research for my game. This stage was all about making *my game separate and unique from other games that are out there*,

I decided to investigate VR technology (Virtual Reality) as it is a technology market set to explode into the market over the next few years, with the first major release of VR technology the oculus rift due out in the first quarter of 2016. Let's start with the projected revenue forecast for VR technology up to 2018.

(Graph 3 Projected Revenue for VR products)



Twin Darkness Technical Report

As can be seen from now until 2018 there is a steady increase in revenue as VR technology hits the markets. Next I want to see the growth in users of VR technology in order to assure that there would be a steady stream of new users over the next few years. Using the same sites/resources as before I compiled a list of projected users of VR up 2018 to align with graph 3.

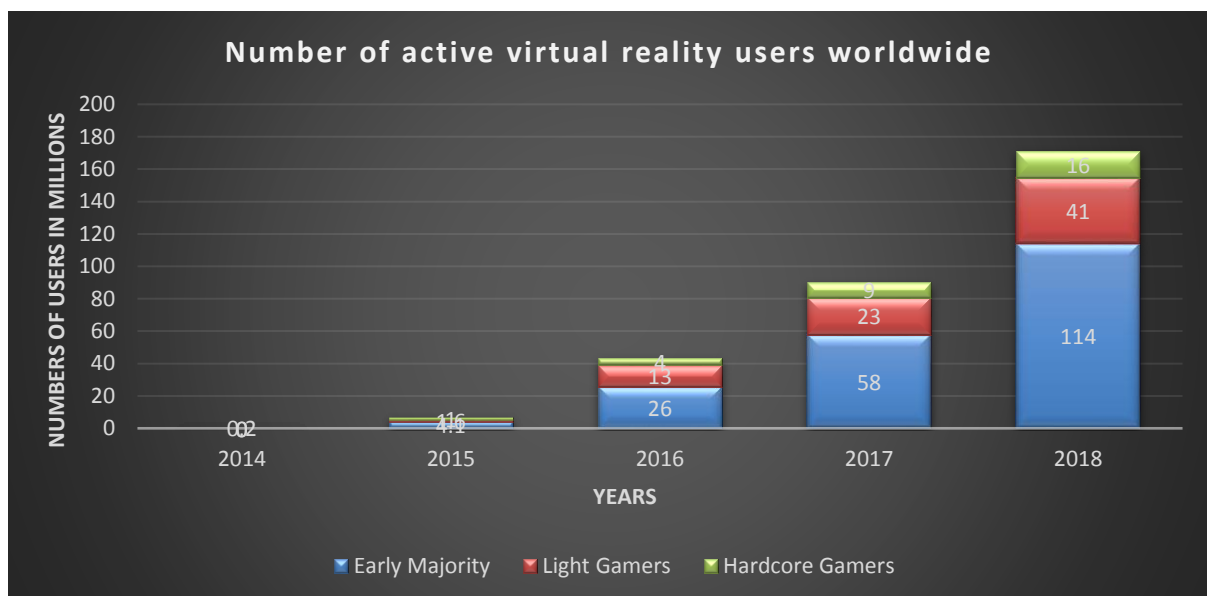
The users are broken down into three test groups

- Early Majority
- Light Gamers
- Hardcore Gamers

(Full Table of VR Users Worldwide)

Year	Early Majority	Light Gamers	Hardcore Gamers
2014	0	0	0.2
2015	4.1	1.6	1
2016	26	13	4
2017	58	23	9
2018	114	41	16

(Graph 4 VR Users Worldwide)



Now that I have broken down that the VR market will be growing for at least the next three years, I now have the standout/innovation side of the project covered.

Now that I have my chosen genre and the inventive technology side covered, I moved onto the fourth stage of research which was game engines.

I examined game engines under four different criteria's

- Code Base Type
- Developer Reviews

Twin Darkness Technical Report

- Industry Used/Industry Standard
- Cost

I started off making a list of the five best engines on the market today: (and began to break them down using the four criteria's listed above)

- Frostbite 3
- Unreal engine 4
- Unity 5
- CryEngine 3
- Source 2

Once I had the information I need for each engine I compiled a list under the all different headings and choose the best one from the list. (*Table 2 Game Engines Broken Down*)

Engine	Code Type	Developer Rating/Review	Industry Used/Industry Standard	Cost	Chosen Result
Frostbite 3	C#, C++	Ranked 5th Best 2015	Shooter systems	Yearly Cost	No
Unreal Engine 4	C+, Visual Code and Blueprints	Ranked 1st Best 2015	Multiple Area's	Free	Yes
Unity 5	C#	Ranked 2nd Best 2015	Multiple Area's	Free	No
CryEngine 3	C#	Ranked 4th Best 2015	Open World system	Yearly Cost	No
Source 2	C++	Ranked 3rd Best 2015	Multiple Area's	Yearly Cost	No

Unreal engine 4 proved to be the winner not only was it ranked the top game engine of 2015 but it was only 1 of 2 that where free to all developers at all levels. It also uses a visual coding style which is the way of the future when it comes to game development, so it was by far the ideal choice for me to begin developing in.

With Unreal chosen I moved onto stage five of my research which was investigating different types of modelling software/asset creation software, I decided to stick with three different types of modelling/asset creation software

- Blender 3D Modelling software
- Mixamo / Maya
- Photoshop

I decided early on that most of the development of 3D assets would be done in Blender as it is free and I have prior experience using it to develop 3D assets

Photoshop would be used for the creation of textures to go over 3D models to bring them to life and make the environment more engaging for the user.

Twin Darkness Technical Report

Maya would be used as a rigging and animation tool if other online sources failed such as the much acclaimed Mixamo offered by Adobe design.

The final stage is deployment of the future game across as many platforms as possible and how it will be released (in what form). Lucky this game will be build and designed for the PC (Windows) thus opens a number of free market place options for deployment:

Option 1: Deployed Website and Social Media formats (2nd Choice Option)

- This would be a site designed and maintained by myself and would have a digital download of the full game for users, the website would be found through social media accounts created for adverting the game and its launch.
- This process would give me full control over distribution but would be highly depended on an active social media campaign in order to spread word of the game.
- Also would offer full control over pricing and payment system (*example would be PayPal*)

Option 2: Digital Distributors (1st Choice Option)

- Using a digital distributor, is the norm for independent game developers there are a number of platforms but the one that is most used is Steam and their greenlight system
- Steam greenlight allows users with active accounts to vote for what go's to market and gets a place in the steam store.
- Greenlight allows users to test beta versions and record a number of statistics (such as: rating, bug reports etc.)
- Greenlight continuously cycles through all active products meaning it will be given an equal chance to be viewed
- Would take some control and rights out of my hands but would mean a much small social media campaign and more control over reviews and testing
- Limited control over pricing

Options 3: Unreal Engine 4 Market (3rd Choice Option)

- Unreal allows for its users to post their content to the market place allowing them to set their own prices
- Limited control as unreal has final say what goes up and for how long.
- No real feedback system

Now that we have our market options, we need to look at in what form the game will be released in, there are a number of key ways:

- **Full Game** Release
- **Episodic** Release
- **Early Access** release to full game release

Full Game: Would not be the right fit in my case since I am a single indie developer meaning it could take years to complete and release a full game and by then there could have been massive changes in the industry as a whole.

Early Access: Would be the same problem as Full Game release, time is the biggest factor

Twin Darkness Technical Report

Episodic: In recent years the new popular way of releasing games is to do it in an episodic form. Which works best in a *story driven* game, hence it is the option I am going with.

A great example of episodic releases come from Telltale games, they release monthly chapters to their audiences. Allowing them to improve and continue to keep audiences engaged with their games.

When it comes to my game it will:

- Allow me to focus on one chapter at a time, which is best to ensure a good experience for the user
- Allow for continuous releases of new content and in-game systems, meaning the audience will not lose interest
- Enable to users to offer feedback on current content and what they would like to see improved and changed

Recap of what options I picked for my project from the research conducted:

- ❖ ***Chosen genre: Thriller***
- ❖ ***Chosen Innovation: VR (Virtual Reality)***
- ❖ ***Chosen Engine: Unreal Engine 4***
- ❖ ***Chosen Tools: Blender, Maya and Photoshop***
- ❖ ***Chosen Launch Platforms: Steam, Deployed Website and Unreal Market***
- ❖ ***Chosen Launch Type: Episodic***

(In the next chapter I move onto the full concept design from the a story view, this will cover the world being created for the user along with the characters and enemies they will meet and interact with)

Full Game Concept Breakdown

Overview of Genre Setting:

- The main genre of the game will be **thriller**
- It will be locked to a **first person** (for a full immersion feel and to highlight the secondary genres)
- Secondary genres that will make up the rest of the game setting will be the following:
(*Survival, Horror and Sci-fi*)

Environment (World) Setting:

- Dystopian future setting
- The Year 2166
- World is on the brink of its End War (Final War)
- New weapons are being developed for the End War

Story Characters:

- **Gemini:** 21 years old, No memory before waking up within the Equinox facility motivation is to understand what is going within the Equinox facility. Has been modified by the leader of the Equinox facility now has 2 powers (active and passive).
 - **Active power** allows Gemini to see in the dark for a limited time the more it is used the more health it will cost, if the player is not careful it will and can kill them
 - **Passive power** allows Gemini to see a highlight around objects and devices she can interact with
- **Libraus System:** Controls the equinox facility and its drones, is the reason the equinox facility is offline, has its own dark past that our main character can uncover. Two different aspects (A.I element and the unknown element)
 - **A.I Element:** The part of the system that is trying to regain control and capture/kill Gemini (Red Screens mean under A.I control)
 - **Unknown Element:** The part of the system that is gone mad (or is corrupted) may help or hinder Gemini (Green Screens mean under Unknown Element control)
 - **Note: Gemini may still interact with either system just results may change**
- **Dr Solomon Thorn:** Head of the equinox facility and lead designer on the new product line *Zodiac*, is the one that modified Gemini
 - **Name will appear on pad reports**
 - **Will be watching Gemini**

Enemies Types:

- **Aries:** Alphas, alphas will be boss type enemies. Random attack pattern and damage, actively hunt the player.
 - **Can Kill Player**
 - **Has Sight and Audio detection abilities**

Setting History

The Year 2166, Earth is on the brink of its End War, resources and natural supplies have been nearly depleted by the three remaining factions

- Alliance of the Pacific Nations
- Hidden Sands Rebellion
- The Kingdom of Europa

The Equinox facility is playing all sides of the growing conflicted promising each side weapons from their new product line Zodiac. Currently there are two known zodiac models Aries and Taurus.

Equinox has promised 12 models and they do not aim to disappoint, they will go to any length and break any rule set by man or nature to make the ultimate weapons of darkness.

But is the Equinox facility ready for what they have unleashed.



Twin Darkness Technical Report

Aims

The aims of Twin Darkness are broken down into two sections functionality and Style to better understand all the aims in a clear way

The Functionality Aims are as followed:

- Create a first person thriller
- Create a complete prologue chapter (this chapter will introduce the user to much of their abilities and the game environment)
- Create the functionality to allow a player to connect a VR headset, hence the reason for the game to be developed in first person maximum impact on the player.
- Allow for the functionality of either keyboard and mouse input and controller input.
- Have a fully build settings and start menu for the game
- Have a full save function
- Have a full load function

The Style Objectives are as followed:

- Create custom assets using Blender. These custom assets include the following:
 - My own main character model
 - My own enemy custom models.
- Create a set of custom visuals using the unreal engine 4 partial system. These visuals will be the main style and theme of the game.
- Add in choice options that the player may pick from that change aspects of the game making each play through different than the one before it.
- Finally have a fully build story that has a beginning, middle and end. To achieve this I will build cut scenes and dialogue.

These Aims will be broken down in detail in the requirements section of this report, each one will be assessed to see if they are functional requirements.

Technologies

Hardware

➤ **Xbox Controller (Standard Wireless and Plugged)**

- The Controller will be used to control and play the game, will be needed if the player is using the VR function.
- If the user has opted out of using the oculus rift, the user may then decide to use either the controller or keyboard and mouse set up as both will be enabled in my project



➤ **The Oculus Rift development kit**

- The oculus rift development kit will be used to build the functions into my game along with testing and maintaining all aspects.
- The development kit has built in functions to work with Unreal engine 4 which will be the engine I use for the development of my game.



➤ **The Oculus Rift headset**

- The oculus rift headset will be used to test out and play the game at its full potential. Along with that it will be used throughout the development process to spot and detect visual and gameplay bugs



Twin Darkness Technical Report

Software

List of software being used:

- ***Unreal engine 4***
- ***Blender***
- ***Photoshop***
- ***Mixamo / Maya***

Brief how the software will be used:

The bulk of the level's and technical functionality will be build using the Unreal engine 4 game system. I will use blender to create custom assets to be imported into the unreal engine 4 and texture them using Photoshop. Mixamo / Maya will be used for skeletal rigging.

Unreal Engine 4

What I will use it for is the following:

- ***Level design***
- ***Mechanics and layout of a game (such as a start menu, save options and cut scenes)***
- ***Animation of assets & and character movement***
- ***Custom In-game systems***
- ***Missions***

Blender

What I will use it for is the following:

- ***Main Character modelling:*** (This will be the character the player will see in all cut scenes and the character they will be controlling) I will use blender to build and model the character. This model will then be imported into unreal for animation setup before being place in the game.
- ***Main enemy modelling:*** This is the same as the process above for the Main character modelling the designs will be made in blender.
- ***Main Item Modelling:*** all in-game assets and props.

Photoshop

What I will use it for is the following:

- ***Environmental textures***
- ***Logo Designs***
- ***Item Textures***
- ***Screen Textures***

Mixamo / Maya

What I will use it for is the following:

- ***Rigging character models for animations and movement***

Twin Darkness Technical Report

User Requirements Definition

The user requirements are as followed:

- The user's machine must meet the minimal requirements of the unreal engine 4, Those requirements are:
 - *Windows 7,8 or 10(Windows 7 preferred choice)*
 - *Minimum of 4GB of RAM installed*
 - *Dual-Core Intel or AMD processor, 2.0 GHz or Faster*
 - *NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*
 - *Desktop PC, Mac or Laptop*
- For the best experience the user's machine should meet the recommended requirements for the unreal engine 4, Those requirements are:
 - *Desktop PC*
 - *Windows 7 64-bit*
 - *Quad-Core Intel or AMD processor, 2.5 GHz or faster*
 - *NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*
 - *8 GB Ram*
- The user must have a keyboard and mouse or a controller
- The user must have an Oculus or another form of VR headset if they wish to use the VR functions
- The user must have a build version of the Game

Requirements Specification

The requirements specification will outline a number of variables that the user will achieve while playing the game, they are as followed:

1. User will become aware of the controls and input required to play the game within 20 minutes of Gameplay
2. User will understand the basics of the story within the first level of gameplay
3. User will be exposed to the multiple choice options and understand there impact with the first levels of gameplay
4. VR enabled users will have a fully immersive gameplay experience if they are using a controller, headset and headphones.

Functional requirements

Requirement 1: Play Game/Movement

Description & Priority

The user must be able to play the game (by play I am mean achieve movement using input controls) thus being able to explore the level of the game, hence the priority on this requirement is large.

Requirement Video

Online Video demonstrating requirement in game: <https://youtu.be/UBSG1s9wHNU>

Use Case

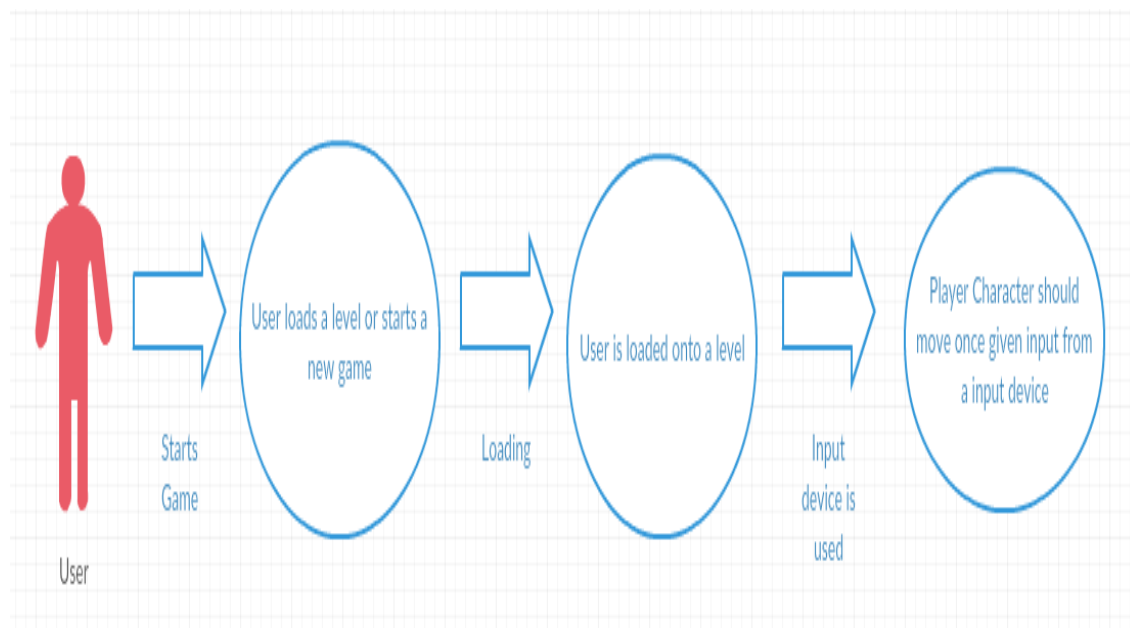
Scope

The scope of this use case is to allow the user to play the game hence moving around the open level of the game

Description

This use case describes the functions of movement that the player must be able to do in order to play and enjoy the experience.

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. A game level should be loaded

Activation

This use case starts when a User/Player uses their input device (keyboard, mouse or controllers) to make the character move in a given direction

Main flow

1. The game loads a level for the user
2. The user sends input using the mouse and keyboard
3. The game reads the input and moves the character
4. Steps 2 and 3 are done at nearly the same time so the user does not have to worry about lag time

Alternate flow

A1: Input from a Controller

1. The game loads a level for the user
2. The user send input using the controller
3. The Use Case continues at position 3 of the main flow

Exceptional flow

E1: Optimal Way of Loading the game

1. The game loads a level for the user
2. The user sends input using the mouse and keyboard
3. The game reads the input and moves the character
4. Steps 2 and 3 are done at nearly the same time so the user does not have to worry about lag time

Termination: Process only stops once the play has stop giving input through a device otherwise it does not terminate

Post condition: (There is no post condition as movement and input from a device will always be needed in some way when playing the game)

Requirement 2: Save Game

Description & Priority

The user must be able to start the game and load into the level from there they must be able to save the game at any point

Requirement Video

Online Video demonstrating requirement in game: https://youtu.be/FwQ_EcjYY2U

Use Case

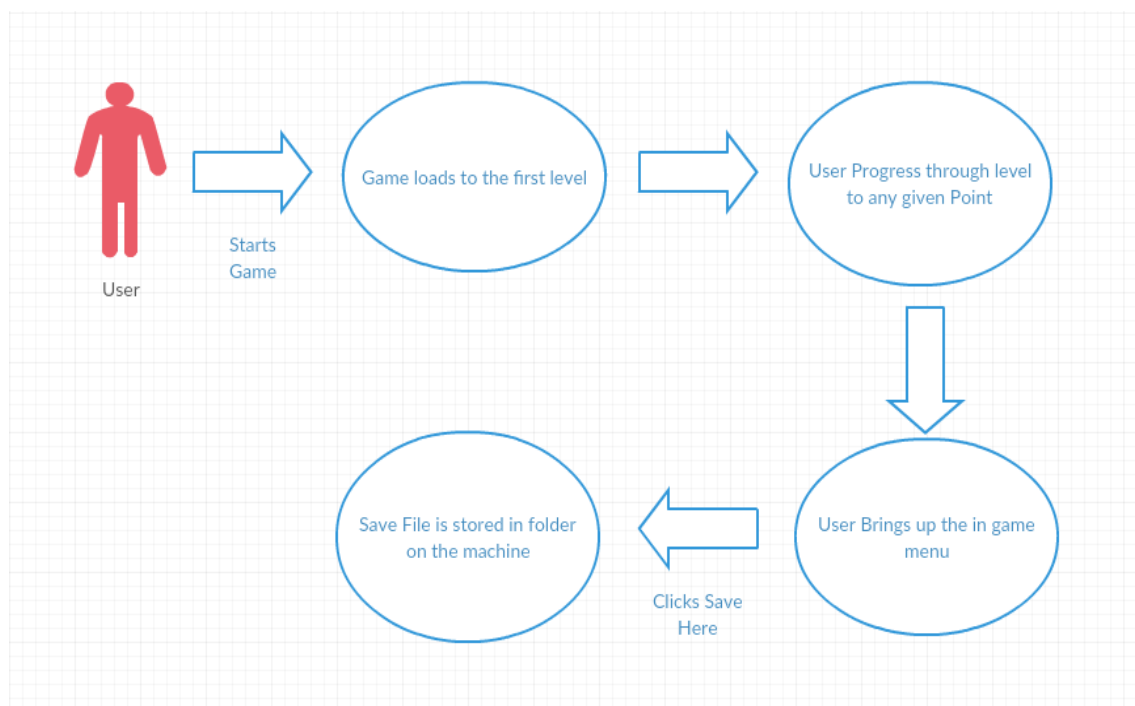
Scope

The scope of this use case is to allow the user to save the game, hence the location and current status on a given level

Description

This use case describes the function of saving an instance of the game, the save will store the users current status (Location etc.) and the Level Status.

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should have also have start a new game or loaded a previous version.

Activation

This use case starts when a User/Player brings up the in game menu after starting the game and loading into a level.

Main flow

1. The game has loaded the level
2. The user can move there character etc. and progress through the level
3. The user brings up the in-game menu
4. The game brings up the in-game menu
5. The user selects Save Here
6. The game creates a save file
7. The game stores the file on the machine

Alternate flow (Does Not Apply)

Exceptional flow

E1: Optimal Way of Saving the game

1. The game has loaded the level
2. The user can move there character etc. and progress through the level
3. The user brings up the in-game menu
4. The game brings up the in-game menu
5. The user selects Save Here
6. The game creates a save file
7. The game stores the file on the machine

Termination: The game saves the file to a target location and returns the user to the game and the level

Post condition: The game system goes into a waiting state, the user may save as much as they want whenever they want

Requirement 3: Load Game

Description & Priority

The user must be able to start the game from the .exe file (see requirement 1 for the number of ways to achieve this), the game should then show the user the main menu. From there a user must be able to click load game, from there they can load a previous saved version of the game and continue on.

Requirement Video

Online Video demonstrating requirement in game: <https://youtu.be/nCyqLjrFpYw>

Use Case

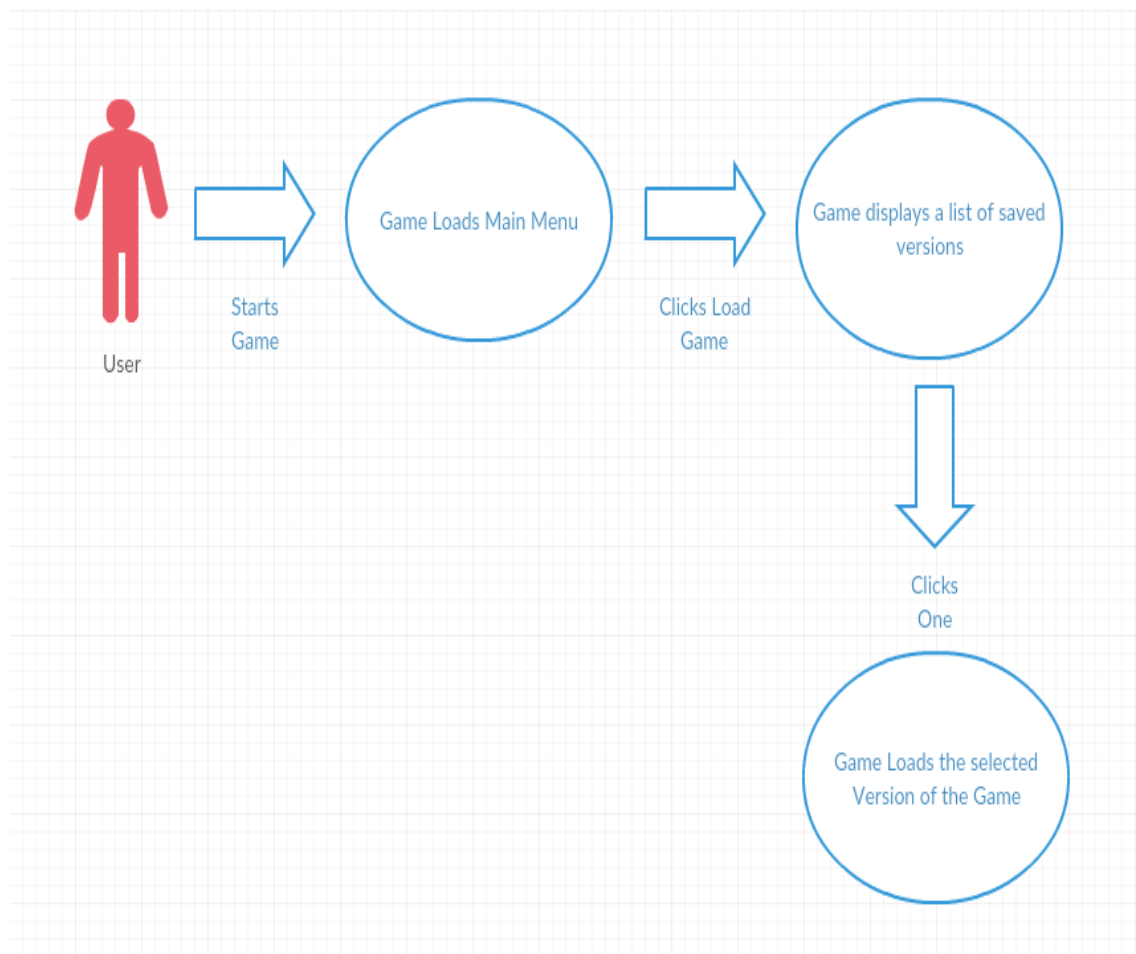
Scope

The scope of this use case is to allow the user to load a saved version of the game, hence continuing on from where they were last

Description

This use case describes the function of loading an instance of the game, the loaded game will allow the user to pick up where they last left off. Insuring they saved the game previously.

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should also have saved the game at least once (see Requirement 2).

Activation

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

Main flow

1. The user starts the game from the .exe file
2. The game loads the main menu
3. The user selects the Load Game button on the main menu
4. The game starts from that saved location

Alternate flow

A1: Load a saved file from an active instance of the game

1. User brings up the in-game menu
2. User selects the Load New Game from the in-game menu
3. The Use Case continues at position 4 of the main flow

Exceptional flow

E1: Optimal Way of Loading the game

1. The user starts the game from the .exe file
2. The game loads the main menu
3. The user selects the Load Game button on the main menu
4. The game starts from that saved location

Termination: The game loads the save file and starts the game from that location.

Post condition: The load game system is inactive during the gameplay

Requirement 4: Exit Game

Description & Priority

The user must be able to exit the game from either the main menu or the in-game menu at any time during the instance of the game.

Requirement Video

Online Video demonstrating requirement in game: <https://youtu.be/5FNqw7CtNbQ>

Use Case

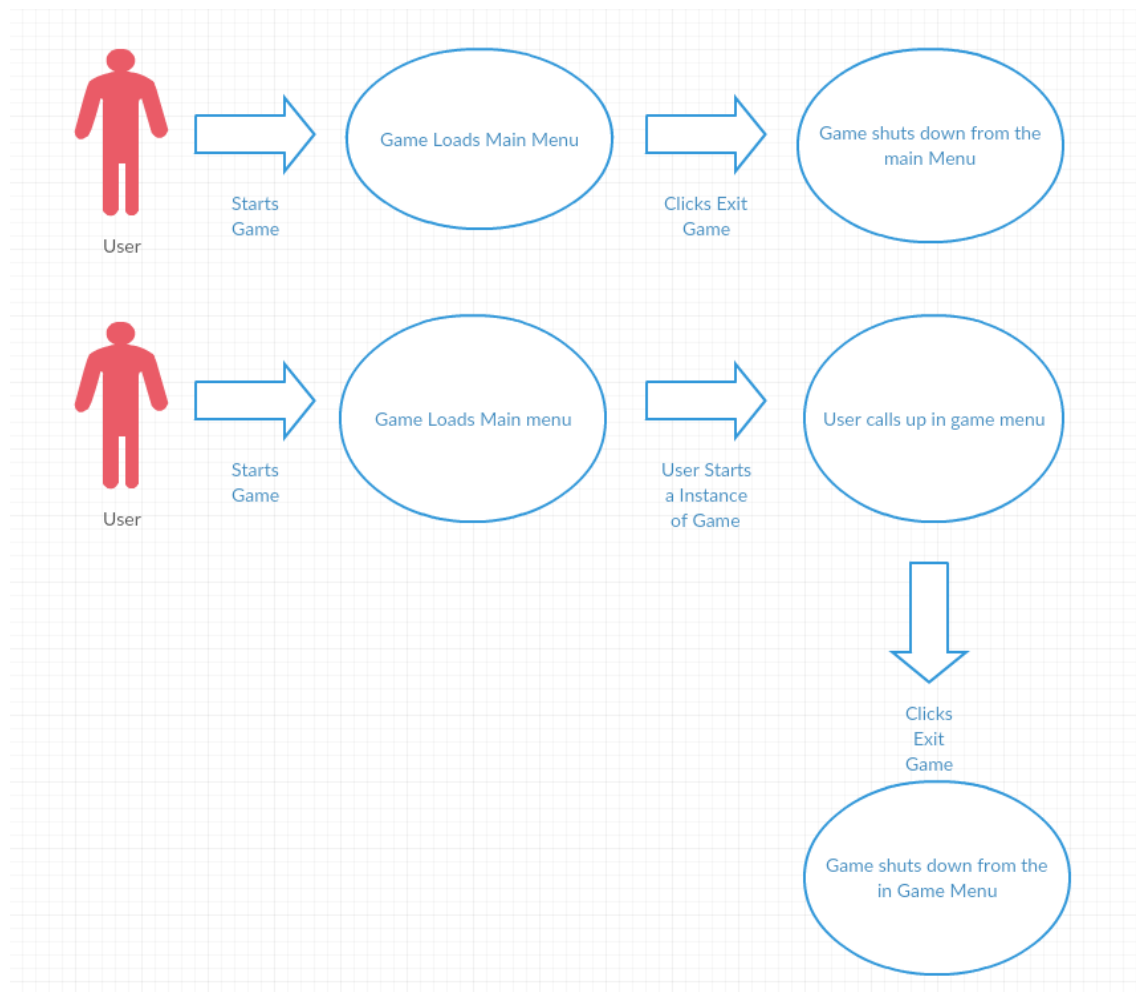
Scope

The scope of this use case is to allow the user to exit the game from either the main menu or the in-game menu.

Description

This use case describes the function of exiting the game from either the main menu or the in-game menu at any time they want, the game will then shut down and the user's desktop will be displayed again and all game functions will have stopped.

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine.
Your desktop or laptop should meet the minimum requirements.

Activation

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

Main flow

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user clicks exit
5. The game shuts down all functions
6. The user is then returned to their desktop screen

Alternate flow

A1: Exit from the in-game menu

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user starts an instance of the game
5. The user then brings up the in-game menu
6. The game bring up the in-game menu
7. The Use Case continues at position 4 of the main flow

Exceptional flow

E1: Optimal Way of Loading the game

1. The user starts the game from the .exe file
2. The game loads
3. The game displays the main menu
4. The user clicks exit

Termination: The game exits correctly and functions are stopped

Post condition: The game has been stopped successfully

Twin Darkness Technical Report

Requirement 5: Modify In-game Settings

Description & Priority

The user must be able to start the game from the .exe file (icon), choose settings to match their machine and start a new instance of the game. This requirement has a massive priority as the game must start for the user.

Requirement Video

Online Video demonstrating requirement in game: <https://youtu.be/yniGE1bbZKo>

Use Case

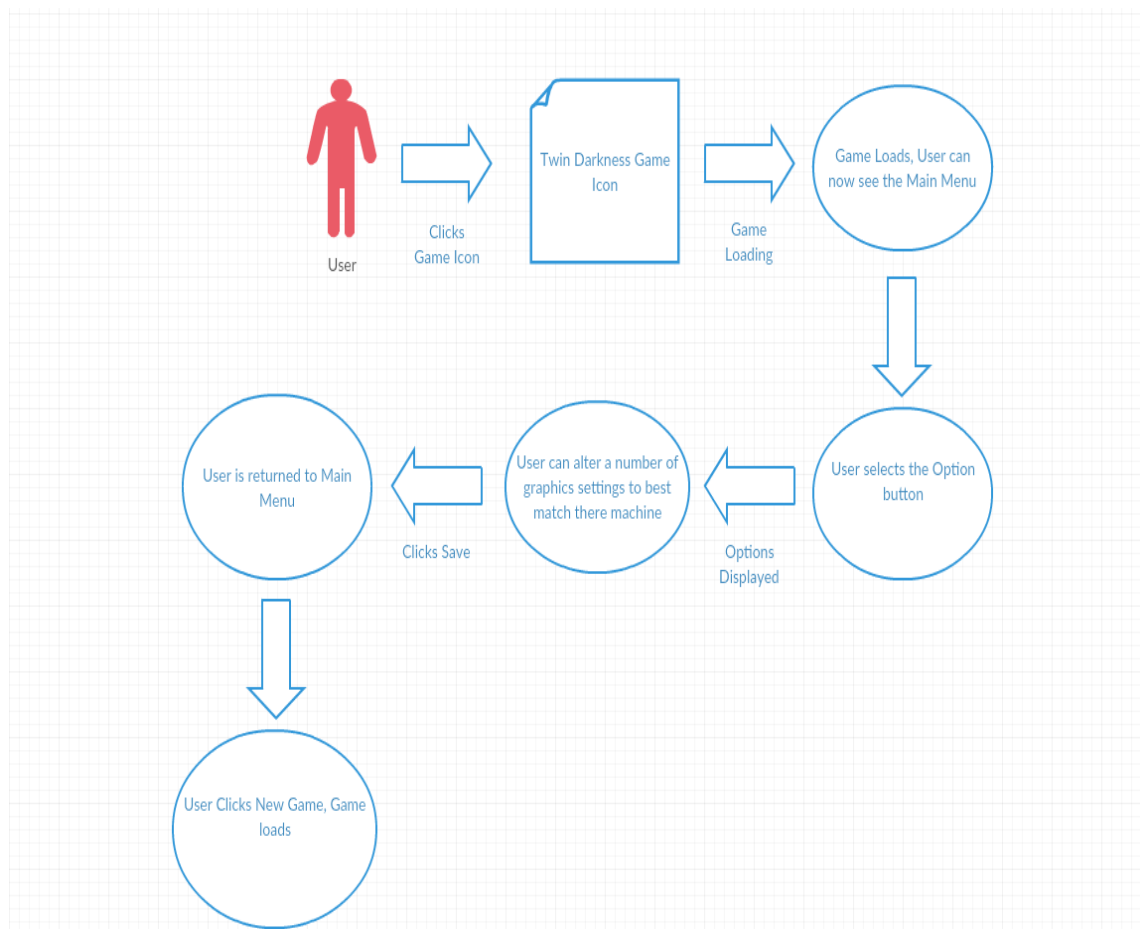
Scope

The scope of this use case is to allow the user to start a new instance of the game and choose settings that match their machines

Description

This use case describes the function of starting a new instance of the game from the .exe file icon of the desktop screen and allowing the user to modify the settings to best match their machines capability for optimal gameplay.

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine.
Your desktop or laptop should meet the minimum requirements.

Activation

This use case starts when a User/Player clicks the desktop icon for the game Twin Darkness, which is a shortcut to the .exe file that was installed when installing the game.

Main flow

1. The User clicks the Twin Darkness game icon on the desktop
2. The Game loads to the Main menu
3. The User clicks the option menu from the main menu display screen
4. The Game loads the default options
5. The User can then modify or leave the options the way there are
6. The Game returns the user to the main menu
7. The User clicks Start New Game
8. The Game loads

Alternate flow

A1: Start Game from the .exe file in the install folder

1. The User finds the install location of the Twin Darkness Game, and locates the .exe file and clicks that instead of the desktop shortcut
2. The Use Case continues at position 2 of the main flow

A2: Start Game from the .exe file in a download zip folder

1. The user downloads the Twin Darkness Zip folder from the site
2. The system downloads
3. The user unzips the folder
4. The user finds the .exe file and clicks it
5. The Use Case continues at position 2 of the main flow

Exceptional flow

E1: Optimal Way of starting the game

1. The User clicks the Twin Darkness game icon on the desktop
2. The Game loads to the Main menu
3. The User clicks Start New Game
4. The Game loads

Termination: The user selects the exit button on the main menu, the system shuts down and returns the user to their desktop.

Post condition: The game system goes into a waiting state on the desktop until started up again by the user

Requirement 6: Complete Prologue Chapter

Description & Priority

The user must be able to start the game and load the main menu, from there they must be able to select prologue chapter for the chapter select screen.

Use Case

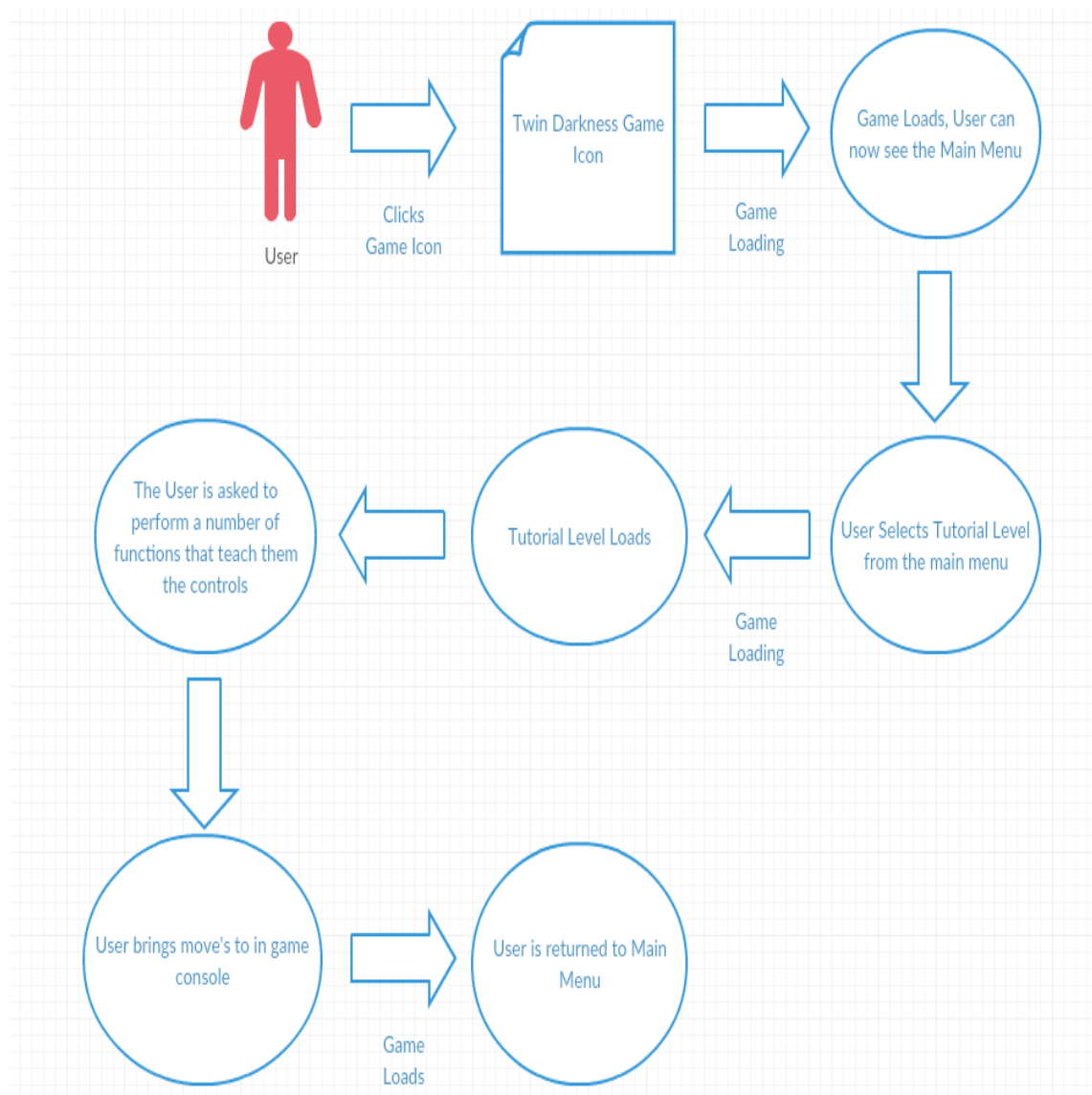
Scope

The scope of this use case is to allow the user to load the prologue chapter, this is the first chapter of the game and will introduce players to the controls and different aspects

Description

This use case describes the function of loading the prologue chapter, so players may get an introduction to the story, controls and the World

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine.
Your desktop or laptop should meet the minimum requirements.

Activation

This use case starts when a User/Player starts the game from the .exe file and the main menu loads.

Main flow

1. The user starts the game from the .exe file
2. The game loads to the main menu
3. The user selects play game
4. The game brings the user to the chapter select screen
5. The user selects the prologue chapter.
6. The prologue chapter loads
7. The user is then asked by the game to complete some simple tasks to show them the controls for the game.
8. The user can then move around and check if there options are ok to play the game
9. The user completes the prologue chapter

Alternate flow (Does Not Apply)

Exceptional flow

E1: Optimal Way of Loading and completing the tutorial level

1. The user starts the game from the .exe file
2. The game loads to the main menu
3. The user selects play game
4. The game brings the user to the chapter select screen
5. The user selects the prologue chapter.
6. The prologue chapter loads
7. The user is then asked by the game to complete some simple tasks to show them the controls for the game.
8. The user can then move around and check if there options are ok to play the game
9. The user completes the prologue chapter

Termination: The game exits back to the chapter select screen so they may continue the story

Post condition: The prologue chapter has been completed

Twin Darkness Technical Report

Requirement 7: Interact with in-game items/systems

Description & Priority

The user must be able to start an instance of the game (hence load or start a level) during the course of the game the user must be able to interact with the in-game items and systems (Examples: unlock/lock doors, use elevators, read computers, use computers to open hatches etc.) These functions have a massive priority as they are key functions of the core game.

Requirement Video

Online Video demonstrating requirement in game: <https://youtu.be/u8TxO0uYG3k>

Use Case

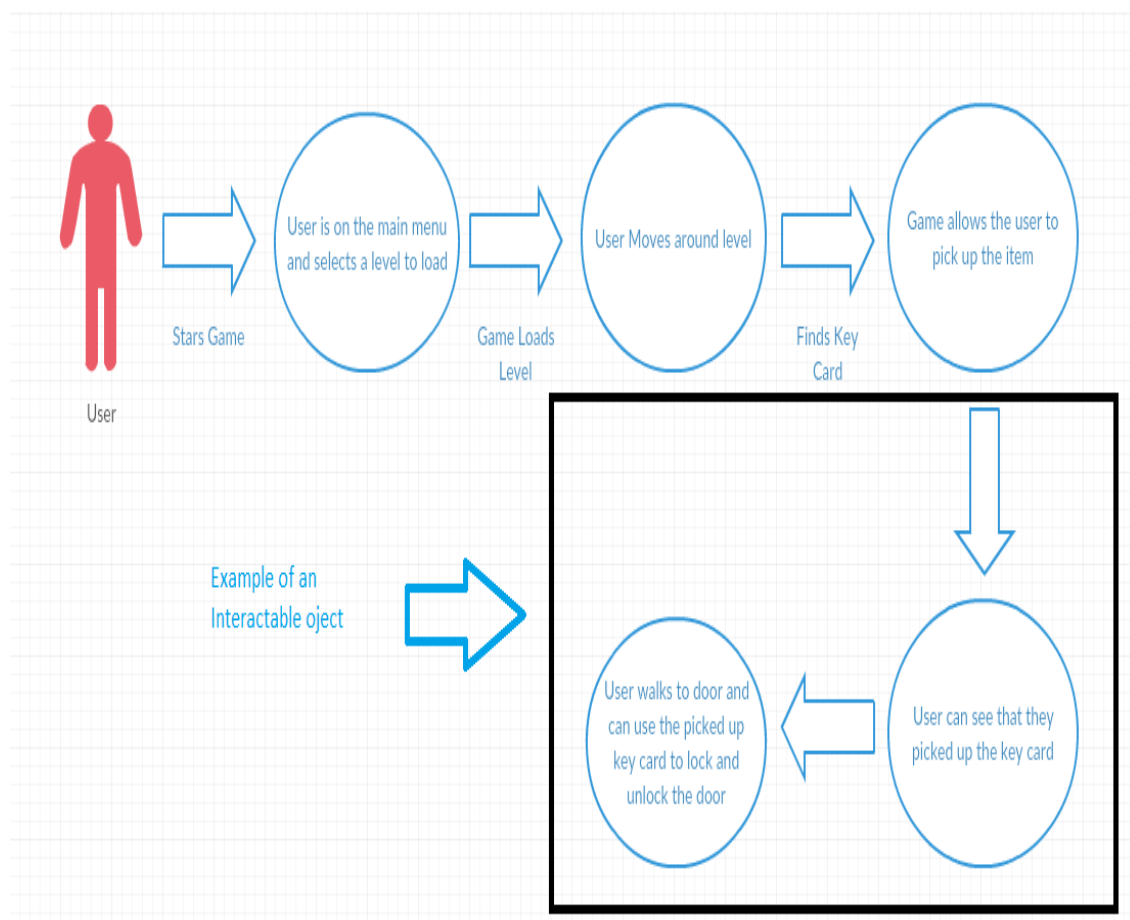
Scope

The scope of this use case is to allow the user to interact with all the different in-game items and systems.

Description

This use case describes the function of the user interacting with the in-game items and systems, Example of such is being able to find a key card and using it to unlock/lock a door. These functions are key to the core gameplay

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. A level is loaded and the user is in game.

Activation

This use case starts when a User/Player starts the game, loads a level and finds an object that can be interacted with in-game.

Main flow

1. The Game loads a level
2. The user moves the character around through the level
3. The user finds an object that can be interacted with.
4. The game allows the user to interact

Alternate flow (Does Not Apply)

Exceptional flow

E1: Optimal Way of Loading and completing the tutorial level

1. The Game loads a level
2. The user moves the character around through the level
3. The user finds an object that can be interacted with.
4. The game allows the user to interact

Termination: The game allows the user to interact

Post condition: The user can interact with given items

Requirement 8: VR Setup/Interaction

Description & Priority

The user must be able to start an instance of the game (hence load or start a level) and enter the VR Mode of the Game using preferred interface Oculus Rift, once the device is connected switch happens automatically, Currently the default setup is Alt Enter to enter the VR mode in the Unreal engine 4. (VR input is optional the game can be played with or without it)

Requirement Videos

Online Video demonstrating requirement part 1 in game: <https://youtu.be/Ksu7POufoDg>

Online Video demonstrating requirement part 2 in game: <https://youtu.be/iDxJDsKZq0s>

Use Case

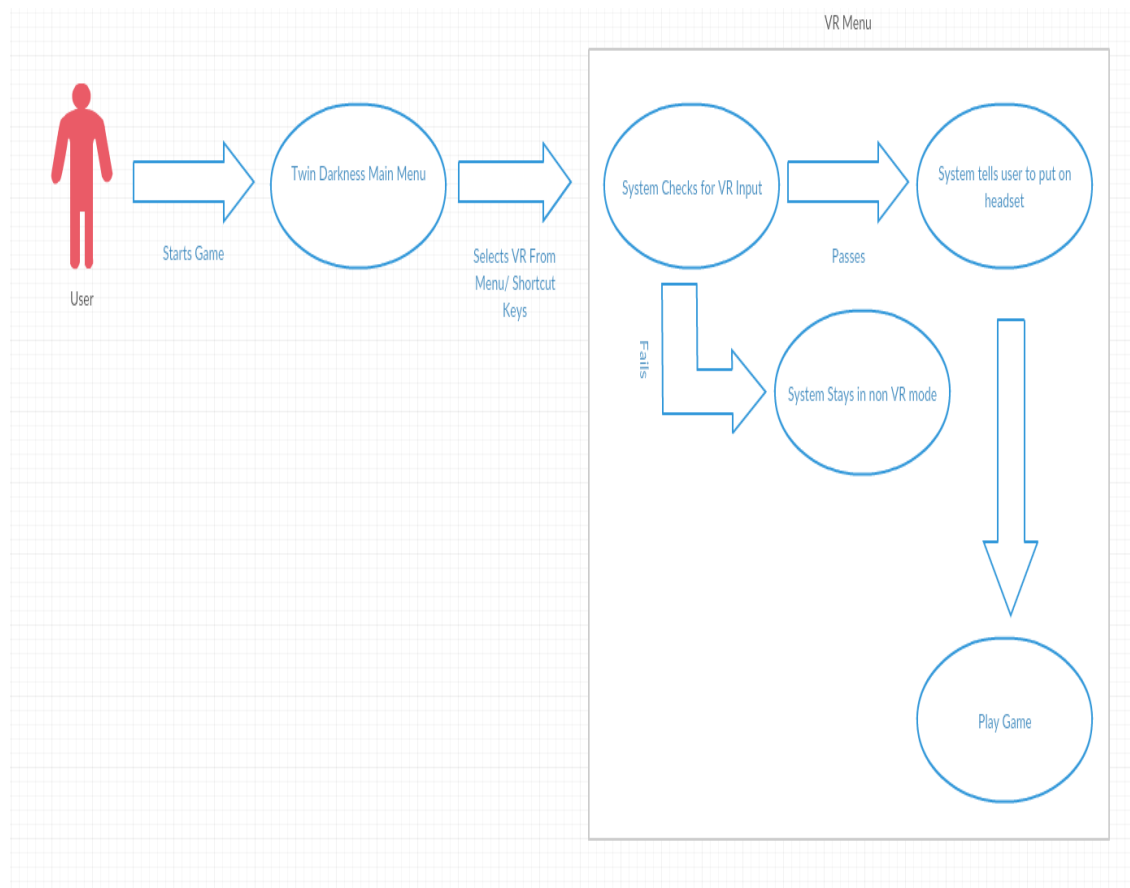
Scope

The scope of this use case is to allow the user to enter the VR mode of the Game automatically when the device is connected.

Description

This use case describes the function of entering the VR mode of the game using the Oculus rift as the preferred input

Use Case Diagram



Flow Description

Precondition

The Game is installed and the Twin Darkness game icon is on the desktop of the machine. Your desktop or laptop should meet the minimum requirements. The user should have all the necessary setup steps for VR input.

- A VR Interface (Oculus Rift Preferred)
- All necessary VR software (Oculus Runtime) information for VR input/setup will be display on Main Menu VR page.

Activation

This use case starts when a User/Player starts the game, and connects their VR device.

Main flow

1. User starts the game
2. System brings the user to the main menu
3. User connects their VR device
4. System checks if there is a suitable VR system installed and ready to go
5. If Pass, VR mode is engaged and the user can put on the headset
6. If Fail, VR mode is not engaged and user is kept in normal mode.
7. User plays the game in what mode is selected.

Alternate flow (Does Not Apply)

Exceptional flow

E1: Optimal Way of Loading and completing the tutorial level

1. User starts the game
2. System brings the user to the main menu
3. User connects their VR device
4. System checks if there is a suitable VR system installed and ready to go
5. If Pass, VR mode is engaged and the user can put on the headset
6. If Fail, VR mode is not engaged and user is kept in normal mode.
7. User plays the game in what mode is selected.

Termination: The game engages into either one of the game modes VR or Normal.

Post condition: The user plays the game in their selected game mode.

Non-Functional Requirements

Performance/Response time requirement

The **performance requirement** is decided by whether the machine being used meets the minimal or recommended settings for Unreal Engine 4

Minimal Settings:

- *Windows 7, 8 or 10, (Windows 7 preferred choice)*
- *Minimum of 4GB of RAM installed*
- *Dual-Core Intel or AMD processor, 2.0 GHz or faster*
- *NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*
- *Desktop PC, Mac or Laptop*

Recommended Settings:

- *Desktop PC*
- *Windows 7 64-bit or Mac OS X 10.9.2 or later*
- *Quad-Core Intel or AMD processor, 2.5 GHz or faster*
- *NVIDIA GeForce 470 GTX or AMD Radeon 6870 HD series Card or Higher*
- *8 GB Ram*

These settings will also determine the **response time** for in-game from boot to in-game actions, for this reason I will be adding a number of options in the options menu when it comes to altering the graphics settings, so players can adjust to meet their machines.

Availability requirement

Option 1: Deployed Website and Social Media formats

- This would be a site designed and maintained by myself and would have a digital download of the full game for users, the website would be found through social media accounts created for advertising the game and its launch.
- This process would give me full control over distribution but would be highly dependent on an active social media campaign in order to spread word of the game.
- Also would offer full control over pricing and payment system (*example would be PayPal*)

Option 2: Digital Distributors

- Using a digital distributor, is the norm for independent game developers there are a number of platforms but the one that is most used is Steam and their greenlight system
- Steam greenlight allows users with active accounts to vote for what goes to market and gets a place in the Steam store.
- Greenlight allows users to test beta versions and record a number of statistics (such as: rating, bug reports etc.)
- Greenlight continuously cycles through all active products meaning it will be given an equal chance to be viewed
- Would take some control and rights out of my hands but would mean a much smaller social media campaign and more control over reviews and testing

- Limited control over pricing

Options 3: Unreal Engine 4 Market

- Unreal allows for its users to post their content to the market place allowing them to set their own prices
- Limited control as unreal has final say what goes up and for how long.
- No real feedback system

Security requirement

The security requirement for the game is light as the game does not require access to the users internet, therefore the users only has to insure that they have the proper security privileges/access on their machine to allow them to install and run the game since the game will only be an .exe file.

Reliability requirement

The reliability requirement is pretty straight forward the game must always be able to launch for the .exe file. Once launched the unreal engine 4 settings take over.

Maintainability requirement

Maintainability of the game will be supported through both the official site and the social media accounts, where users can report bugs and errors. From there I will be able to post fixes and update patches, to insure the game is maintained for the user.

Steam also allows its developer to post and apply updates there are download automatically insuring the user suffers very little downtime. They also have a forum for each of the published games where again users can post errors and bugs.

Extendibility requirement

After the game is released on a number of platforms, it may be built upon with DLC content (downloadable content), this content would include new levels and advancement in story etc. DLC content is always a great way of making performance and bug fixes, so this requirement will be necessary for the long-live of the game.

Reusability requirement

The assets that I will be creating and animation models will be crucial in future DLC content for this game and for future games in the same genre.

Implementation

Each implementation section, will include a detailed breakdown and attached screenshots from within unreal engine 4, documenting the implementation process and its functions.

Game Systems Implementation

This section of the implementation will cover the overall game systems, these systems are crucial to the running of the game and are linked to many of the functional requirement's listed and documented above.

Menu level implementation

Controls, Maintains and Implements:

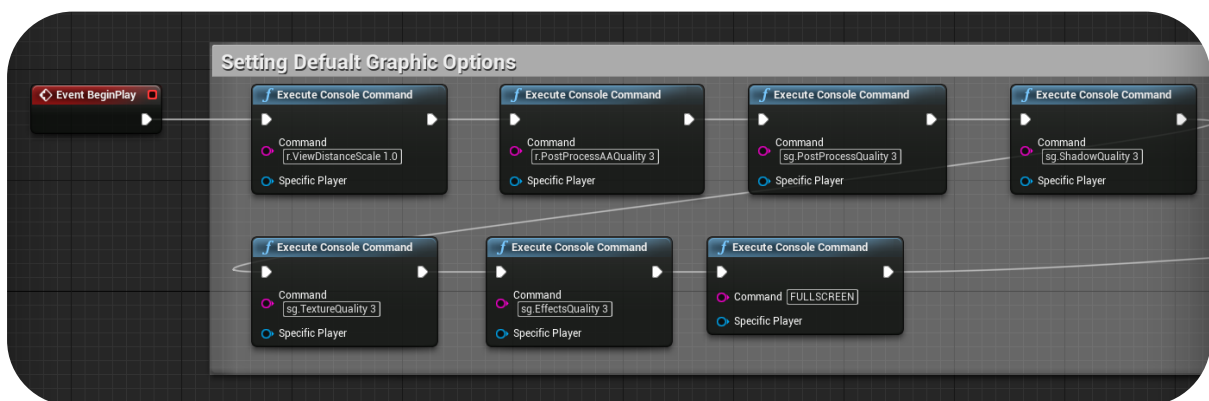
- **Setting the default graphic's:** These settings will be maintained, if the user does not alter them throughout the game
- **Setting screen option to full screen:** This allows the game screen to be set to the users monitor size, can be altered for the options menu throughout the game
- **Enable background animation:** This sets the background animation to playing, enabling it to be used in the menu widgets
- **Creating the main menu:** This is the first widget the user will see, allowing them interaction will all the game options

Implementation & Screenshots:

At menu level creation the blueprint class calls and executes six console commands, these command set the in-game graphical settings.

A seventh command is then executed, setting the screen option to full screen.

These commands are run using the default unreal engine 4 console command system (*command line*)

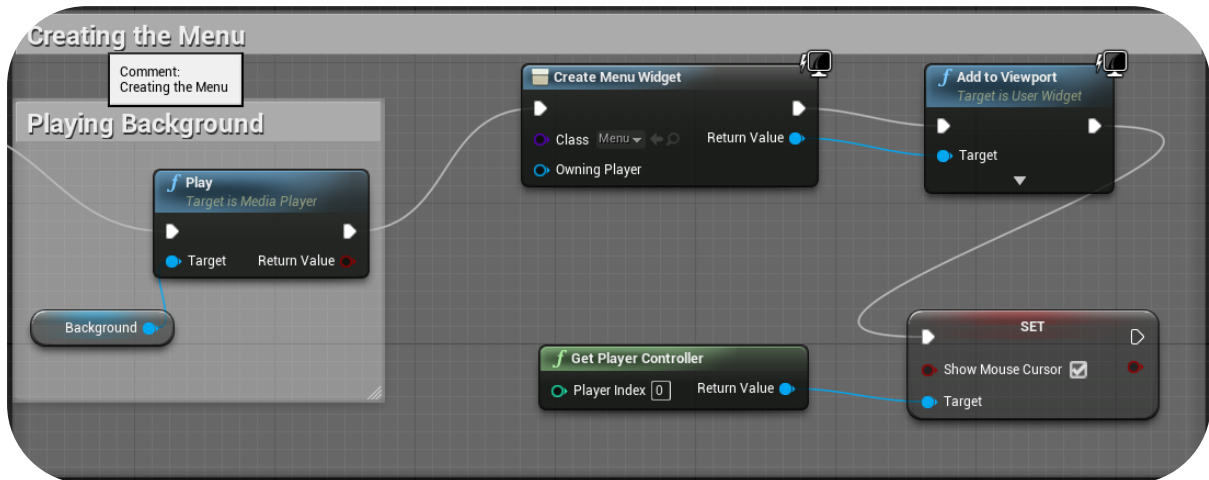


After the console commands are called and set, the background menu animation is enabled, this animation is a playable .wav file imported into unreal engine 4.

The blueprint function: Create Widget is called, this creates the main menu interface for the user. (*Allowing a user to interact*)

Twin Darkness Technical Report

The main menu widget is added to the players screen and the default control option is called and set (default option is mouse)



Main menu widget implementation

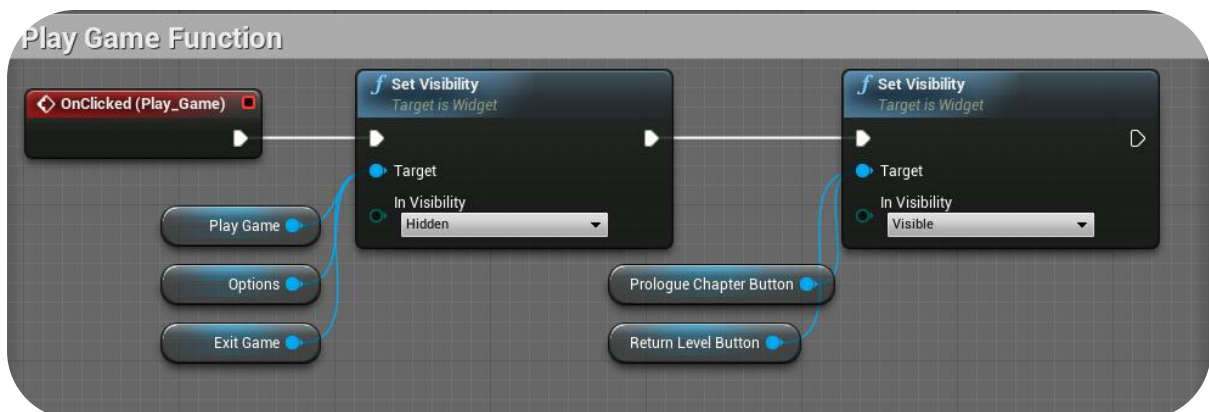
Controls, Maintains and Implements: Large blueprint system some systems will be broken down and detailed in sections, this section will detail first options the user has access to.

- **Play Game button blueprint function:** Allow the user access to the chapter select menu, **this function will detail the setup for changing displays**, while the chapter select menu functions are broken down under their own section, see - [Chapter Select implementation](#)
- **Options button blueprint function:** Allows the user access to the options menu, **this function will detail the setup for changing displays**, while the options menu functions are broken down under their own section, see - [Graphical and Performance Options implementation](#)
- **Exit button blueprint function:** Allows the user to end the game, shutting down all systems and services the game is calling. Returning them to their desktop.

Implementation & Screenshots:

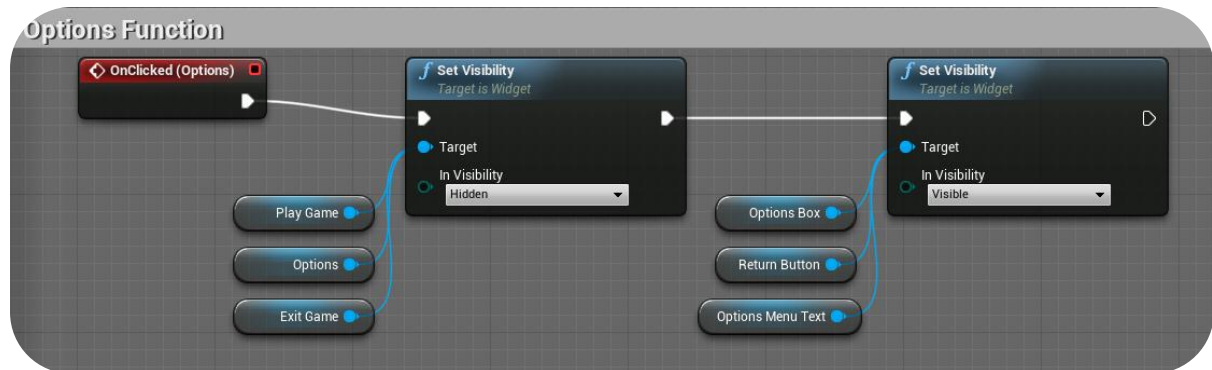
Play game function is an OnClicked event, which sets visibility of the default main menu to hidden (meaning its functions and visuals are no longer visible to the user)

And sets the chapter select side to visible.

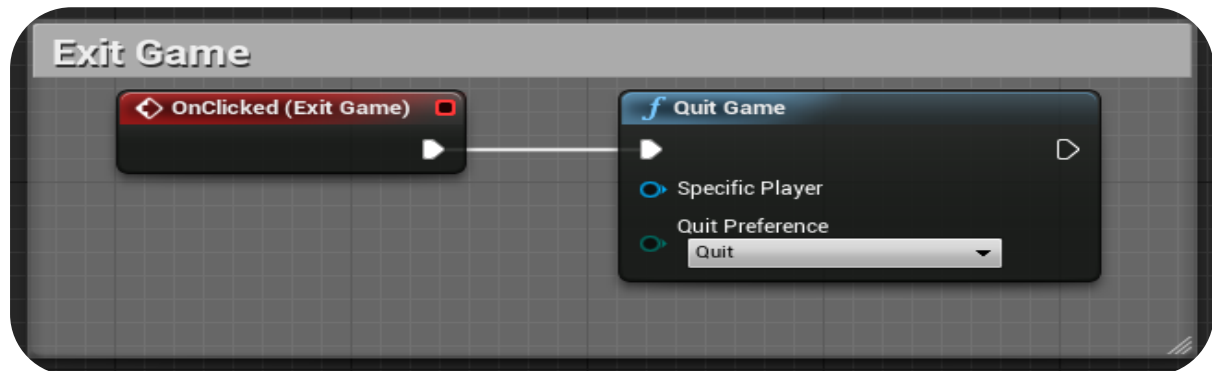


Options function is an OnClicked event, similar function to the above play game function.

Visibility of the default main menu is hidden and the options side of the menu is made visible to the user.



Exit game function is an OnClicked event, which calls the blueprint class quit game, this blueprint class executes and runs all needed shutdown commands, allowing the user to exit to their desktop.



Chapter Select implementation

Controls, Maintains and Implements:

- **Chapter select option (Prologue chapter button blueprint function):** Allows the user to access available chapters (to date only prologue is enabled)
(Chapter select option contains load function, for loading a previous game that the user had saved automatically, for a full breakdown of the backend save and load functions) see - [Save & Load Backend System implementation](#)
- **Return button function:** Allows the user to leave the chapter select and return to the default main menu.

Implementation & Screenshots:

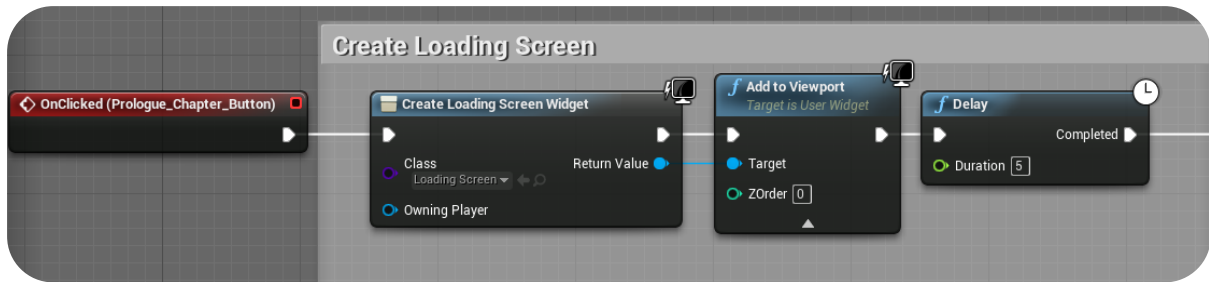
Prologue chapter function is an OnClicked event, its overall function is to transfer the player from the main menu level to the prologue level.

Its first function is the creation of a loading screen between the levels.

Loading screen is created using the Create Widget blueprint function, it is then added to the viewport of the user.

Twin Darkness Technical Report

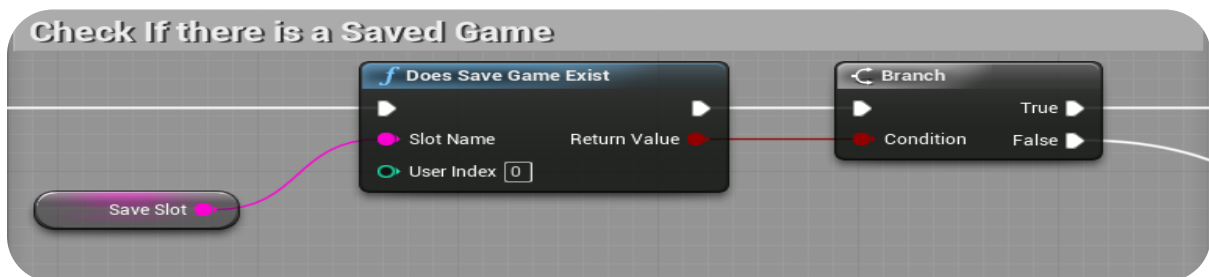
Delay tick of five seconds before the next set of functions are called, insuring adequate load time for the user.



After the delay from the loading screen the blueprint: Does Save Game Exist checks the content of the save game folder, by default unreal saves external files to (*Twin_Darkness\Saved\SaveGames*)

Its checks this folder against the slot name string variable (*TwinDarknessPro*), the result of this check is attached to a branch statement (*True, meaning the saved file exists or False, meaning the saved file does not exist*).

Depending on the result there are two different paths that can be called.

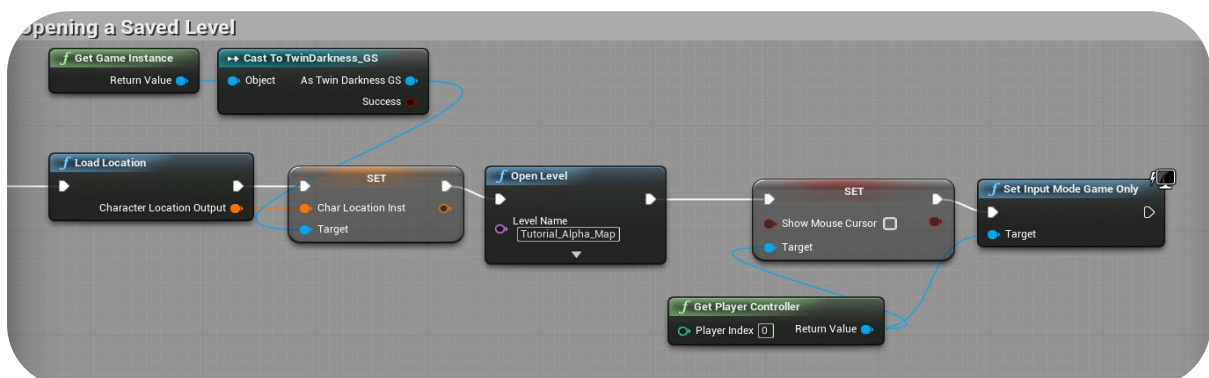


If true: Calls the custom Load location variable, this variable contains a transform output (*transform: contains location, rotation and scale*)

Next the set Char Location Inst is called, this is a custom transform that is called and set with new variables each time a new save is made updating location, rotation (*scale always remains constant*)

Next the game instance is set (*game instance: gets and sets external variables*), this is set so the level can receive the external transform information from the saved file.

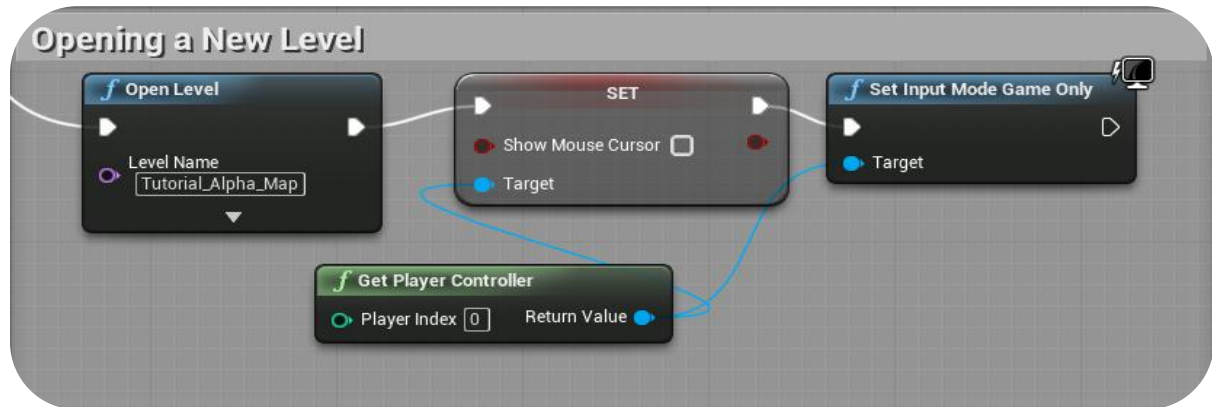
Lastly calls the Open Level blueprint function setting the location to the prologue and we set the in-game input controls and mode (*switching from UI mode to game mode*)



Twin Darkness Technical Report

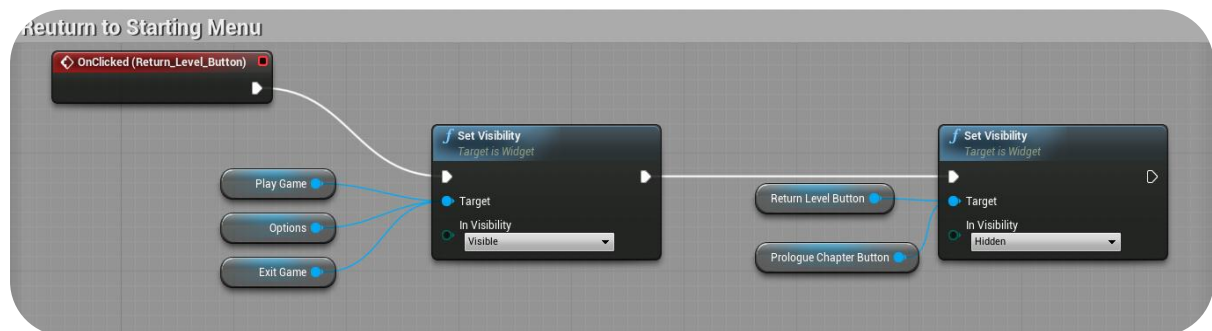
If False: Calls the Open Level blueprint function setting the location to the prologue and we set the in-game input controls and mode (*switching from UI mode to game mode*)

(Does not call any load functions or game instances since there was no saved game detected in the check)



Return button function is an OnClicked event, similar to the other above visibility functions

Visibility of the default main menu is visible and the chapter select menu is set to hidden (**this type of return button function is common on all required widgets**)



Graphical and Performance Options implementation

Outlines the ten different options that the user had access to, ensuring the game can meet the system for the best performance and graphical output.

All options are implemented using **unreal engine 4's console command system**, allowing them to be constant across all levels.

Controls, Maintains and Implements:

- **Resolution Option:** User can set the resolution of the screen
- **Full screen Option:** User can switch from full screen to windowed mode
- **V Sync:** User can set screen refresh to their default monitor
- **View Distance Scale Option:** User can adjust how much of the level loads in front of them.
- **Anti Analising Option:** User can adjust the roughness and smoothness of in-game edges.
- **Post Effect Quality Option:** User can adjust quality of baked in-game effects.
- **Texture Quality Option:** User can adjust quality of in-game textures.
- **Effects Quality Option:** User can adjust quality of rendered in-game effects.

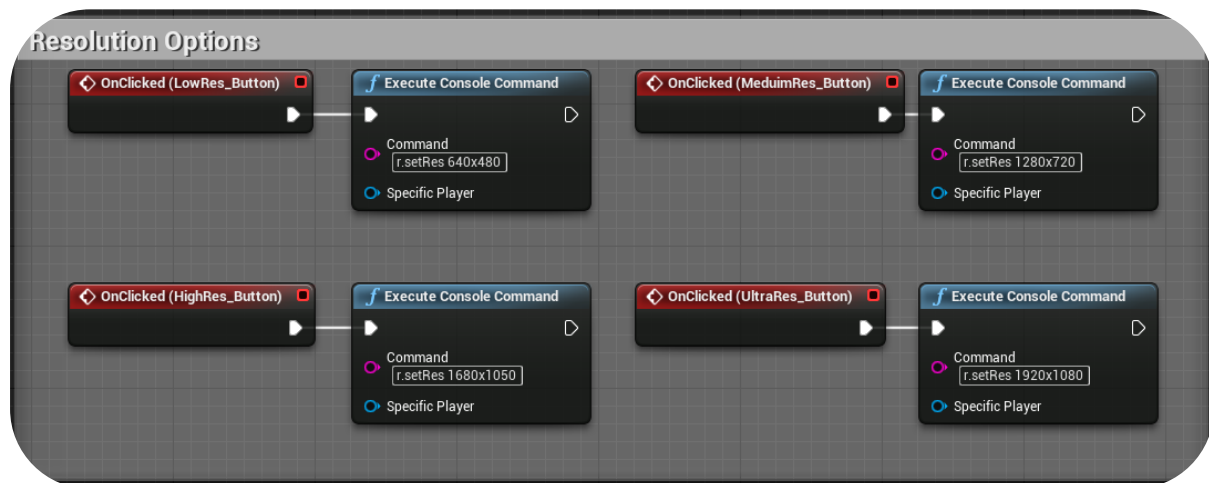
Twin Darkness Technical Report

- **Shadow Quality Option:** User can adjust quality in all in-game shadows and lights.
- **Overall Graphical Pre-set Option:** User can choose pre-set graphical options that will alter all of the above, four pre-set options.

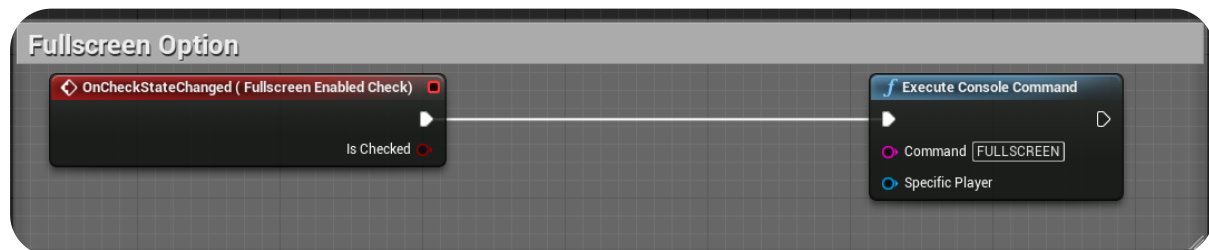
Implementation & Screenshots:

Resolution Option: Resolution is set using one of the four OnClicked button events, the options are *Low, Medium, High and Ultra*. A console command is then executed setting the windowed screen to the appropriate size.

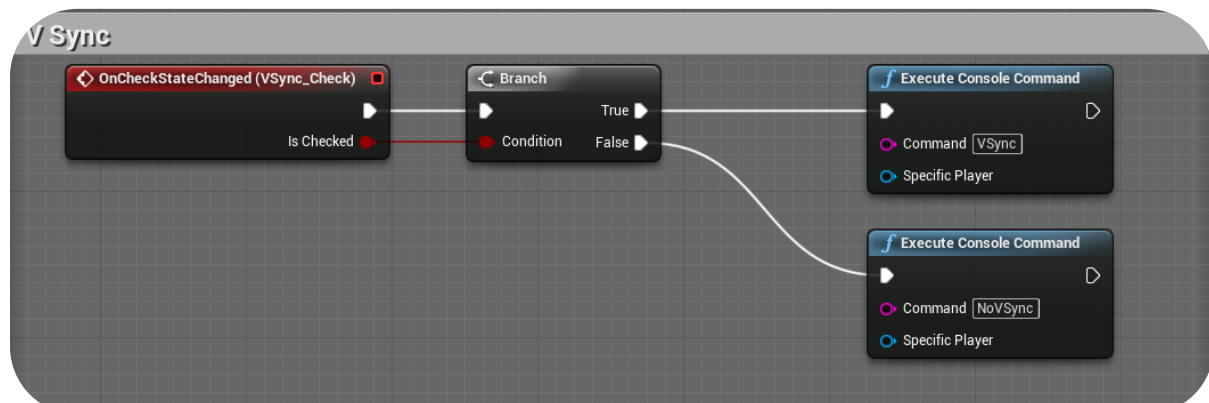
(Only works in Windowed Mode)



Full screen Option: Full screen is set using a single OnCheckStateChanged checkbox event, *Checked to enable full screen, unchecked to enable windowed mode*

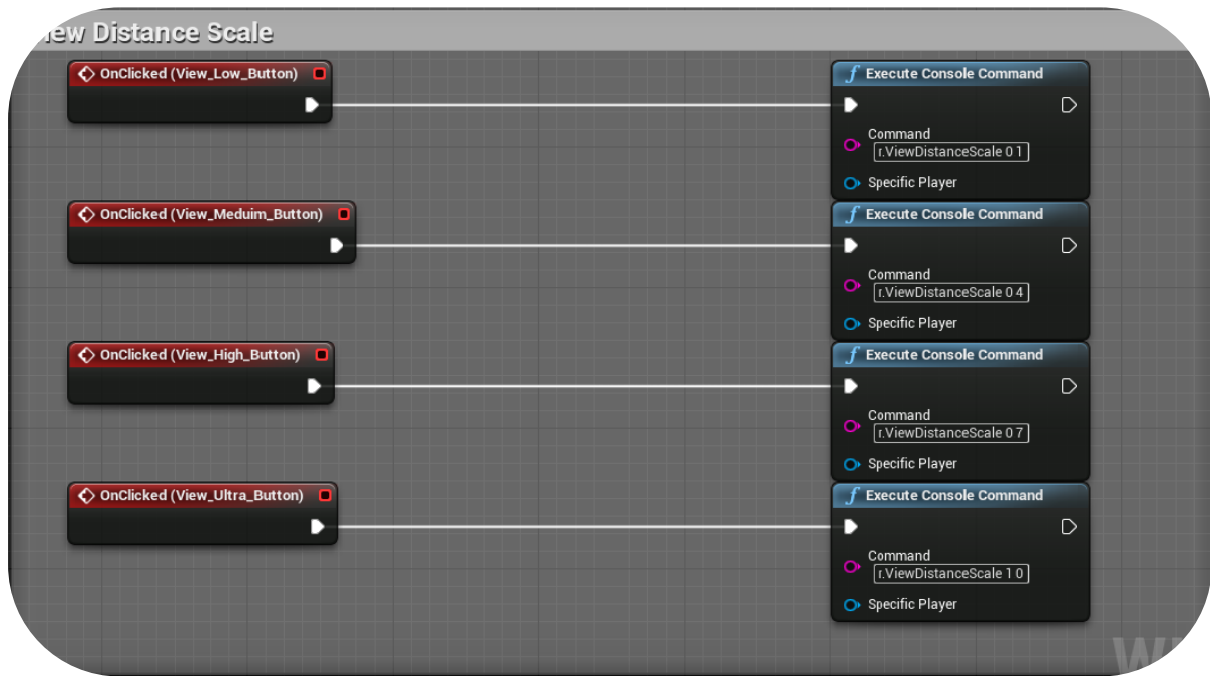


V Sync: V sync is set using a single OnCheckStateChanged checkbox event, with added error handling through a branch statement, *Checked true to enable V sync, unchecked false to disable V sync*



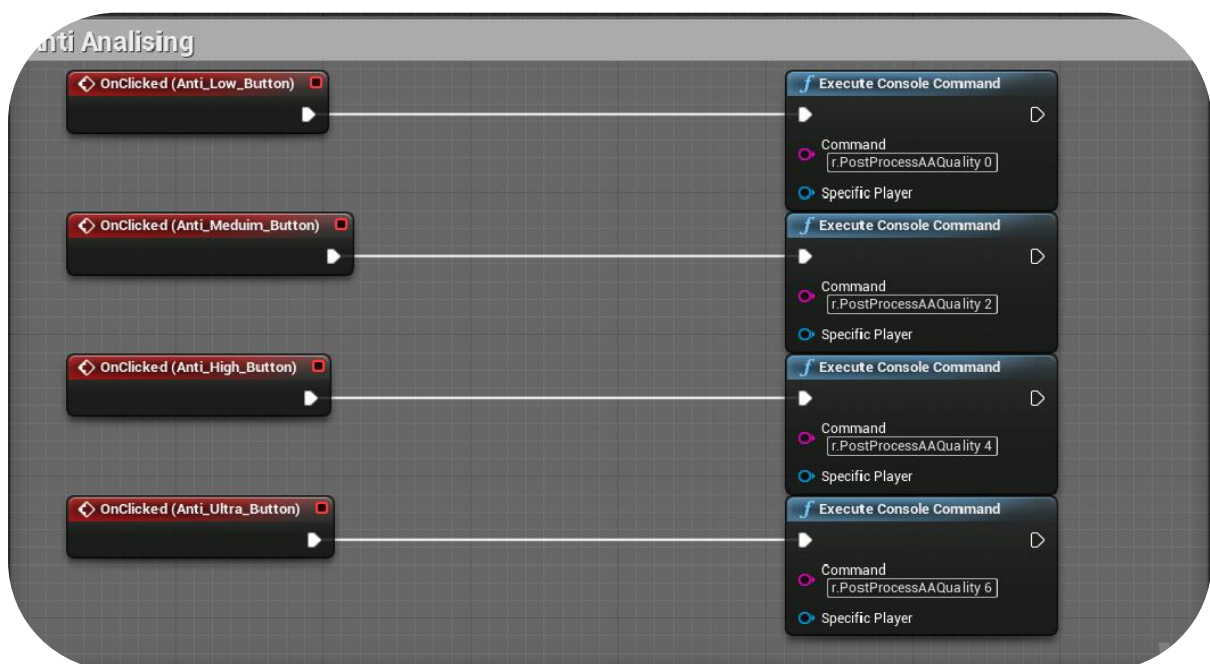
View Distance Scale Option: View distance scale is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed adjusting how much of the level loads in front of them

(Allowing users with low grade or outdated machines to better handle the level and to put less stress on their machines, ensuring a better gameplay experience)



Anti Analising Option: Anti Analising is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed, lower options the more rough the meshes will be, higher options for more smooth meshes

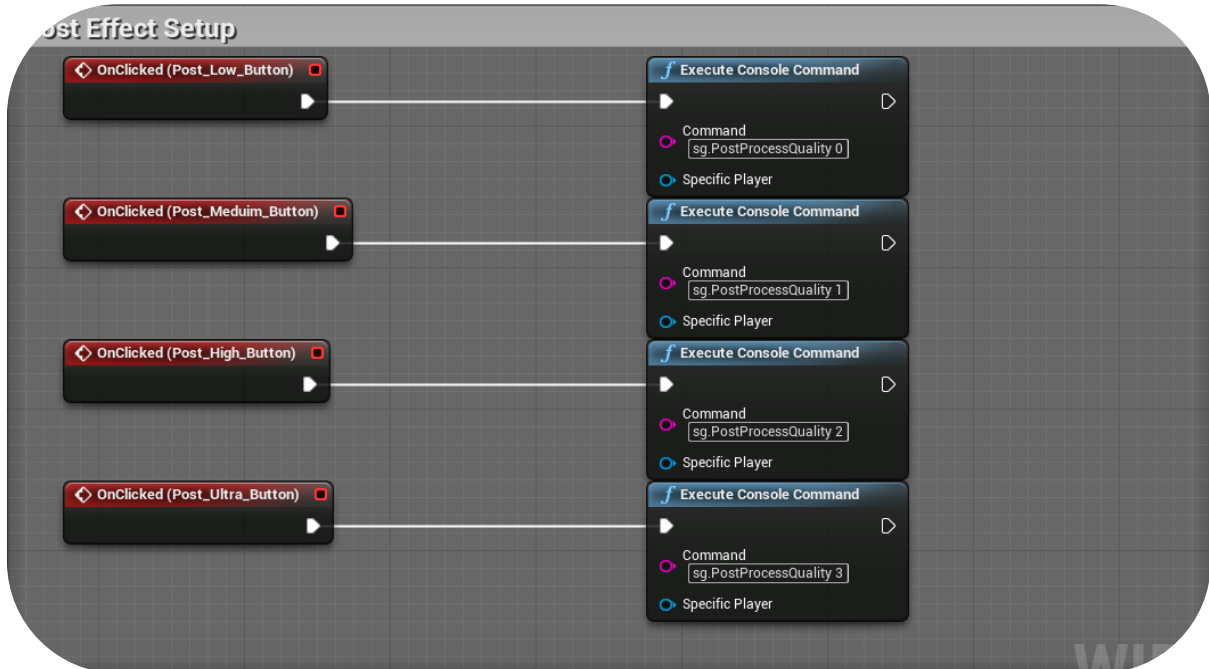
(Not critical on systems, but higher anti analising allows for a better user experience)



Twin Darkness Technical Report

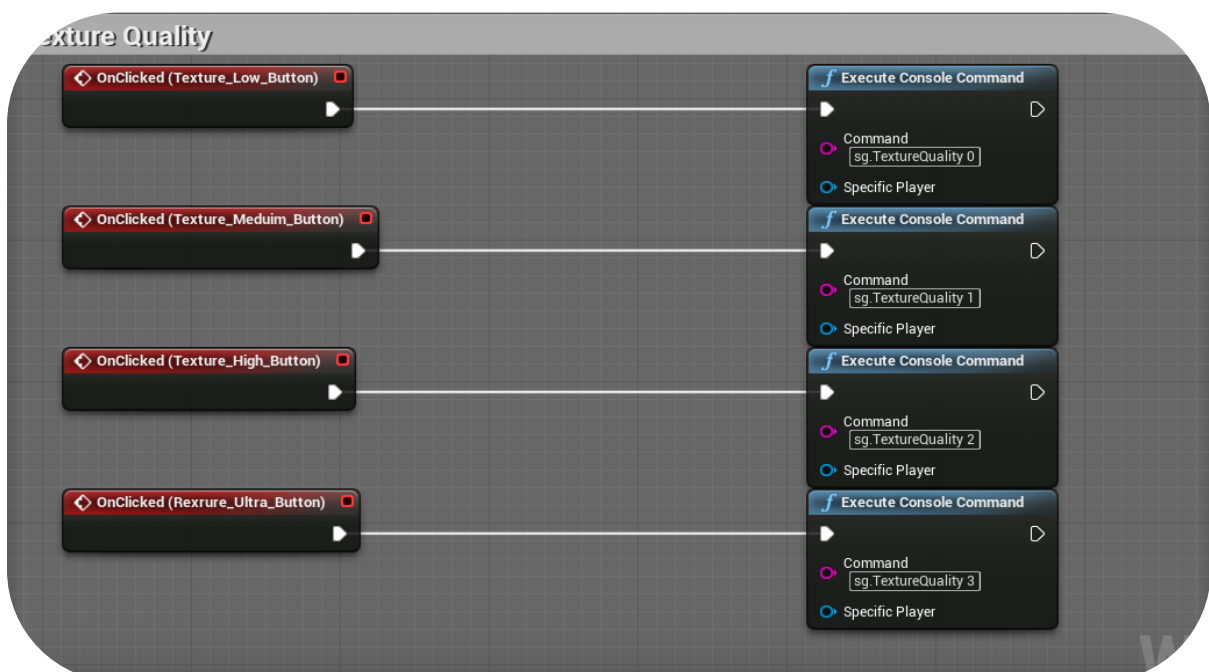
Post Effect Quality Option: Post effect quality is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed, adjusting the quality of baked in-game effects.

(System Need: Not Critical, as these effects are pre-baked into the level)



Texture Quality Option: Texture quality is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed, adjusting the quality of all in-game textures

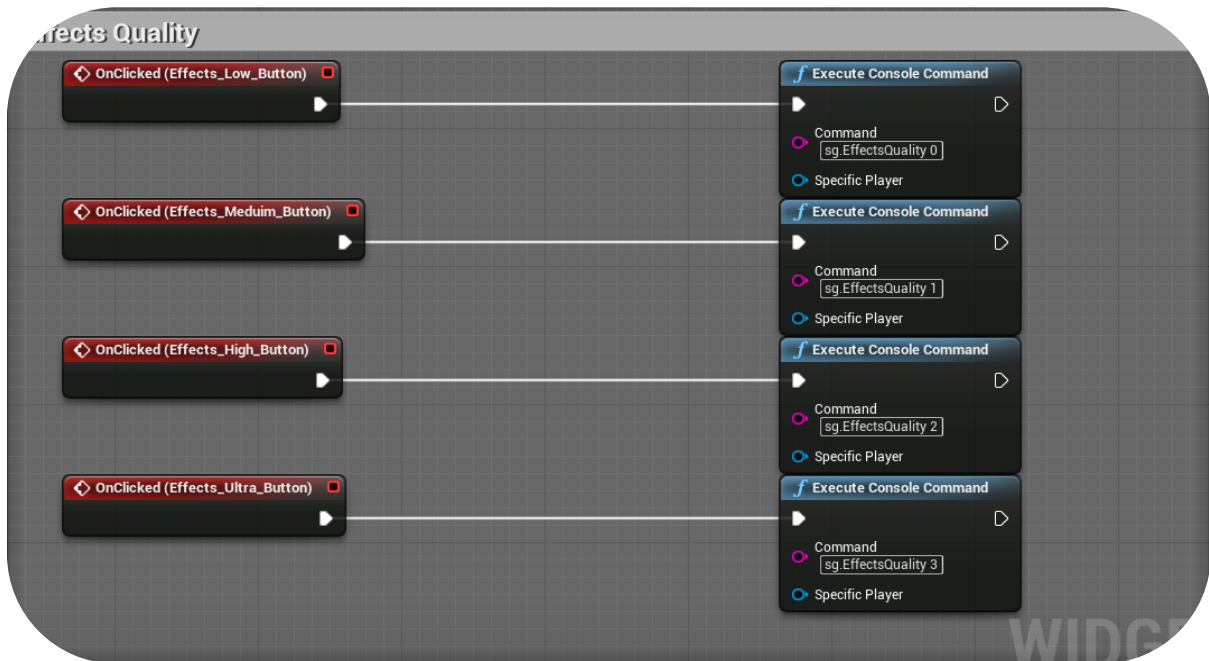
(System Need: Critical, user will low end machines should adjust if they are experiencing performance drops due to the quality of the textures)



Twin Darkness Technical Report

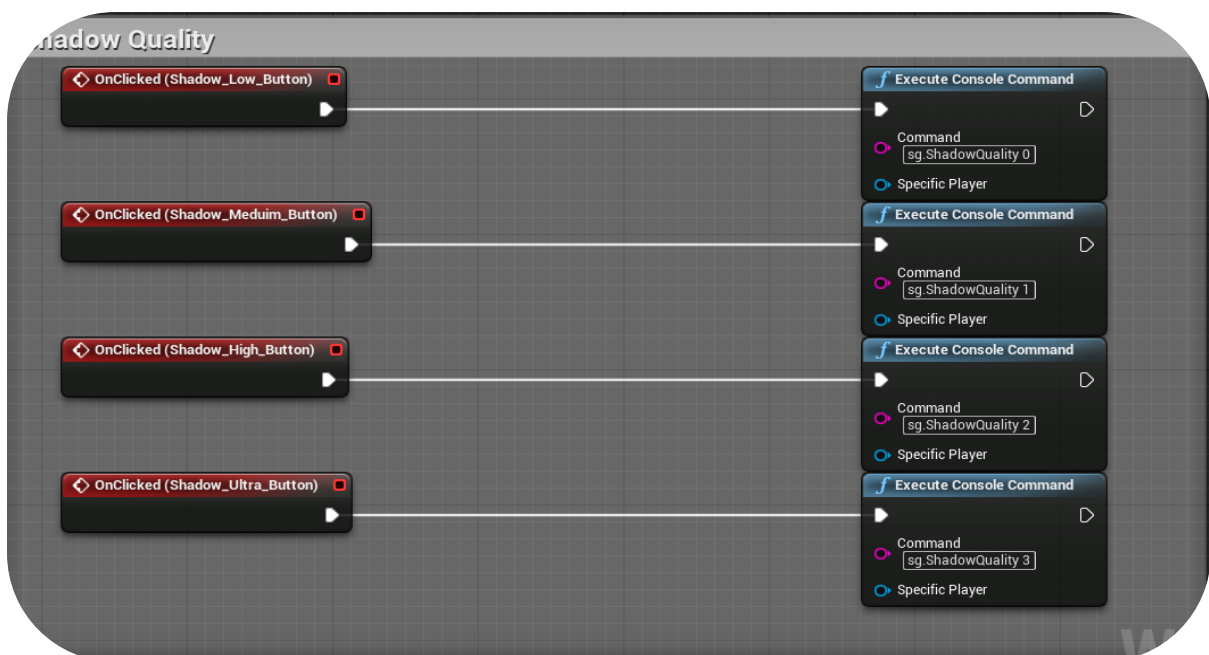
Effects Quality Option: Effects quality is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed, adjusting the quality of rendered in-game effects.

(System Need: Critical, user will low end machines should adjust if they are experiencing performance drops due to some effects being render in too much detail)



Shadow Quality Option: Shadow quality is set using one of the four OnClicked button events, the options are *Low*, *Medium*, *High* and *Ultra*. A console command is then executed, adjusting the quality of all in-game shadows and lighting

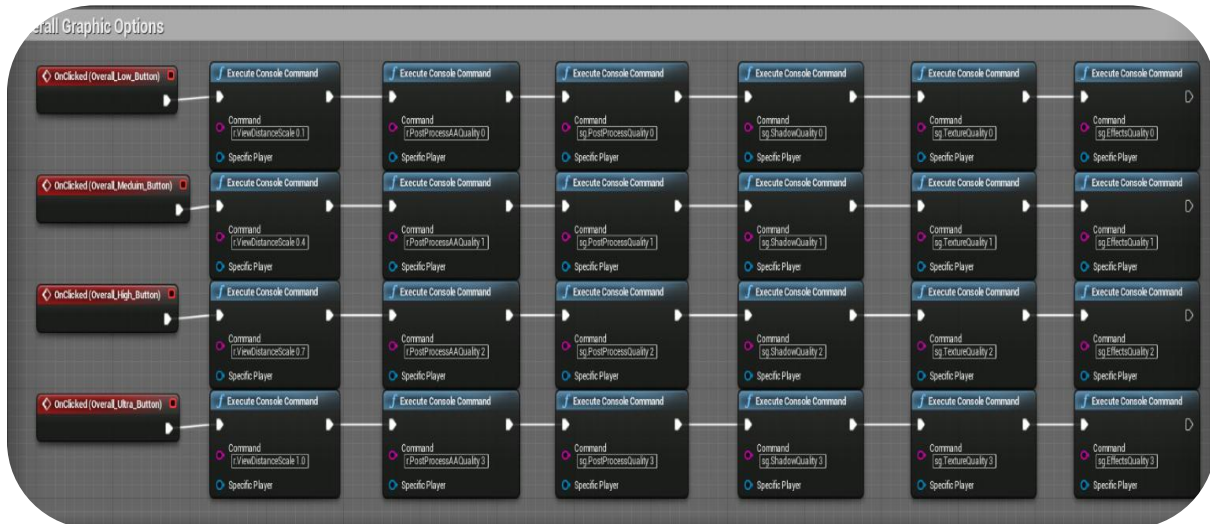
(System Need: Critical, shadows and lighting are the top performance drain on systems, thus this option is necessary)



Twin Darkness Technical Report

Overall Graphical Pre-set Option: Overall graphical options are set using one of the four OnClicked button events, the options are *Low, Medium, High and Ultra*. A set of console commands are executed setting the quality and settings of all the above options to a number of pre-sets

(System Need: Not Critical, User experience tool, allowing them to quickly alter all options are once rather than one at a time)



Pause System & Pause Menu implementation

Outlines how to trigger the in-game pause system, and outlines pause menu functions that will be available

Controls, Maintains and Implements:

- **Pause System:** Allows the user to activate and deactivate the pause system (*which pauses all in-game activities*)
- **Pause Menu:** Widget with options available to the user (*Resume, Options, Save Game, Load Game and Exit Game*) (***Pause menu contains save and load functions for a full breakdown of the backend save and load functions***) see - [Save & Load Backend System implementation](#)

Implementation & Screenshots:

Pause System: Can be activated and deactivated using two set keys, these keys are set and mapped by me to unreal engine 4, allowing a custom pause InputAction to be used in all blueprints

Escape Key press on keyboard and mouse

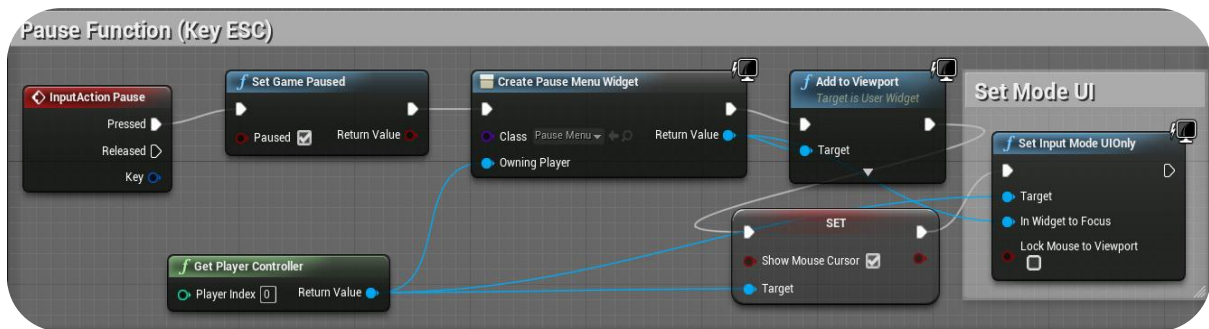
Gamepad special right on controller (Xbox)



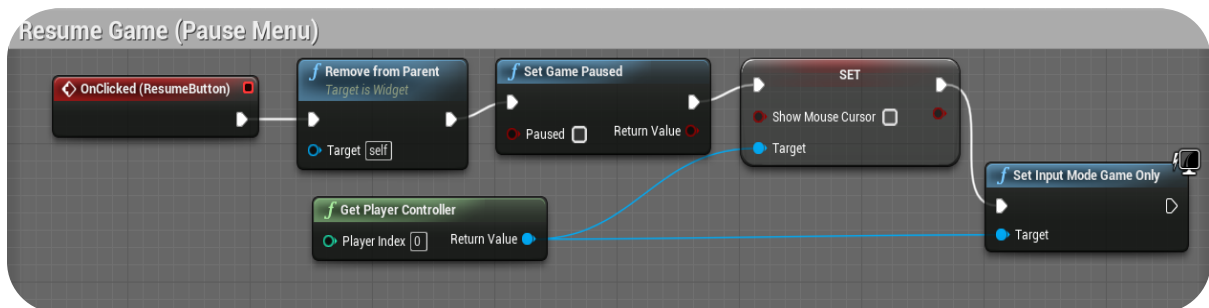
If InputAction Pause is pressed, the blueprint function Set Game Paused is called and is set to true (*this blueprint function will pause all in-game activities*)

Twin Darkness Technical Report

Once paused function is complete, the create widget blueprint is called creating the pause menu, it is then added to the viewers screen and input mode is switched to UI.



Pause Menu – Resume (*Return the user to the game.*): is an OnClicked button event, if pressed the current widget (*Pause Menu*) is removed, Set Game Paused is set to false enabling all in-game activates and we set the in-game input controls and mode (*switching from UI mode to game mode*)



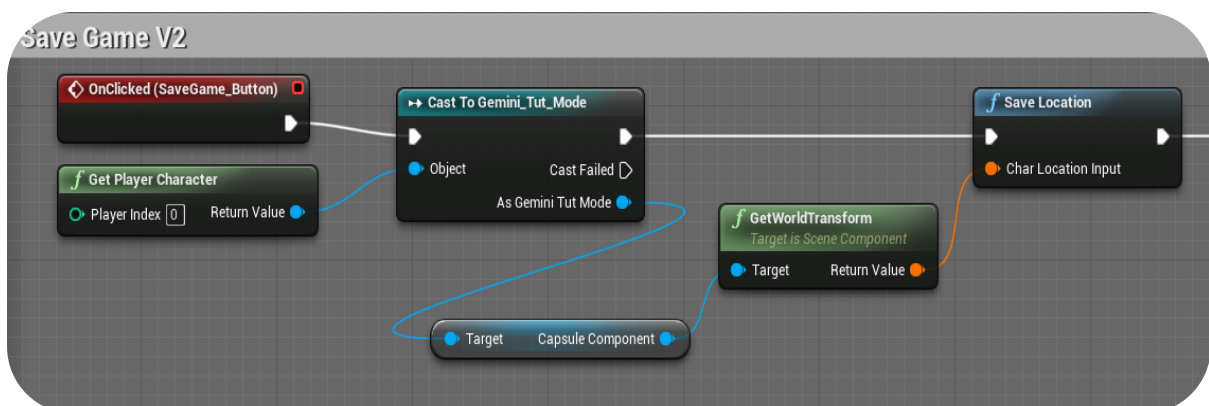
Pause Menu – Options: Same design and implementation as outline here - [Graphical and Performance Options implementation](#)

Pause Menu – Save Game (*Saves the player current location*): is an OnClicked button event, if pressed calls the function Cast to Gemini_Tut_Mode (*Gemini the name of the main character*).

The cast allows us to access variables unique to the character blueprint, in this case we call the capsule component which encircles the character mesh.

We then get the world location (*includes: location, rotation and scale*), that location is then entered into the Save Location variable. (*For use later in load functions*)

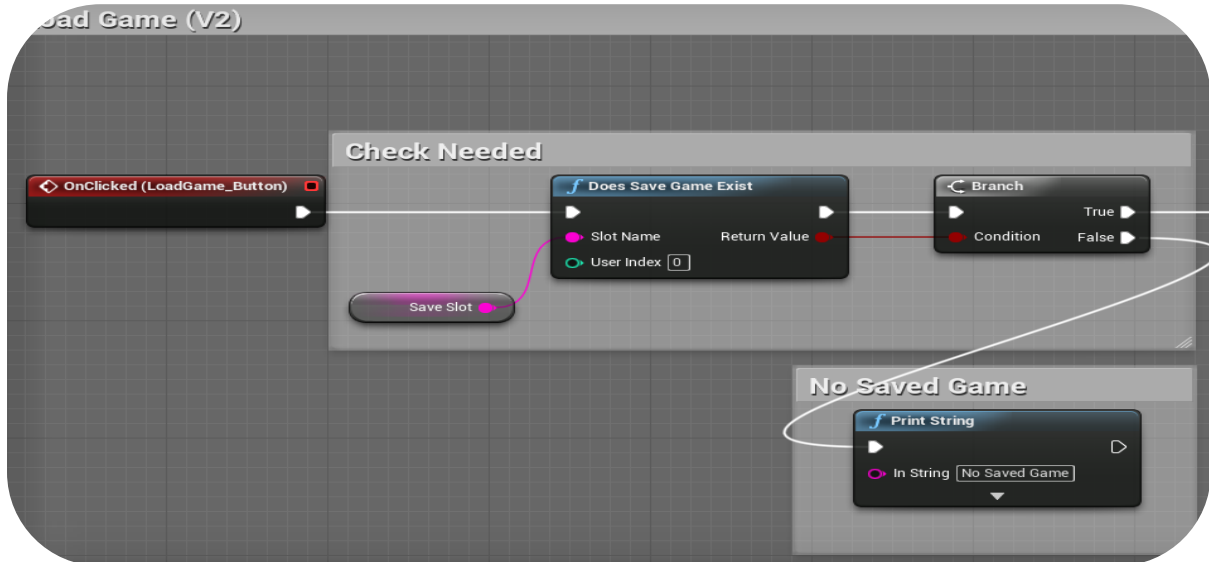
After that a resume game function is called, returning the player to the game



Twin Darkness Technical Report

Pause Menu – Load Game (*Loads the players saved location*): is an OnClicked button event, if pressed calls a check function does Save Game Exist, with a branch statement. Depending on the result one of two branches are called.

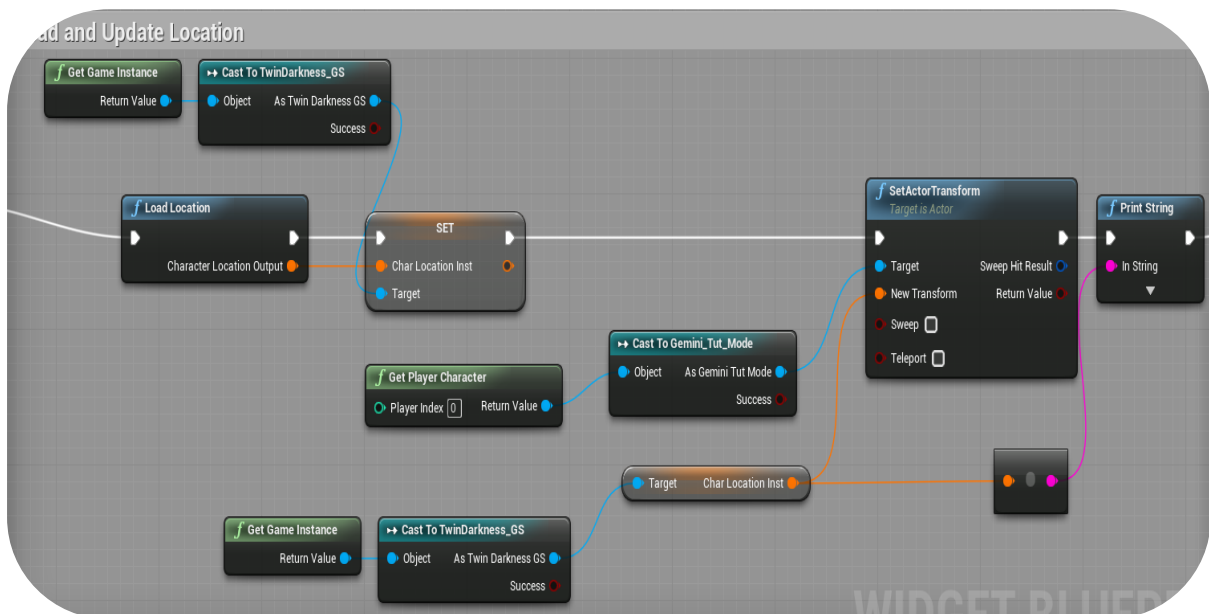
(In this case the false branch call is a print string variable “No Saved Game”)



If the branch statement is true, we call the load function and custom character transform variable, we then call and set the target to the custom game instance (*access to external variables*)

With those variables known, a set transform variable is called, the target for the transform is Gemini (*the character model*)

Its new transform is the custom game instance transform variables. (*Once complete the resume function is called and the user is return to the game at the new location*)



Pause Menu – Exit Game: Same design and implementation as outline under - [Main menu widget implementation](#)

Twin Darkness Technical Report

Save & Load Backend System implementation

Outlines in detail the full save and load system, from the backend perspective of the system. For front end implementation see below for information

Controls, Maintains and Implements:

- **TwinDarkness_Save:** Contains transform variable to store users (location, rotation and scale)
- **Save function:** Allows the user to save their character location (**Save function front-end: detailed under - [Pause System & Pause Menu implementation](#)**)
- **Load function:** Allows the user to load their character location (**Load function front-end: detailed under - [Chapter Select implementation](#) & [Pause System & Pause Menu implementation](#)**)
- **Game Instance:** Accessing external variables and getting/setting variables across level instances.

Implementation & Screenshots:

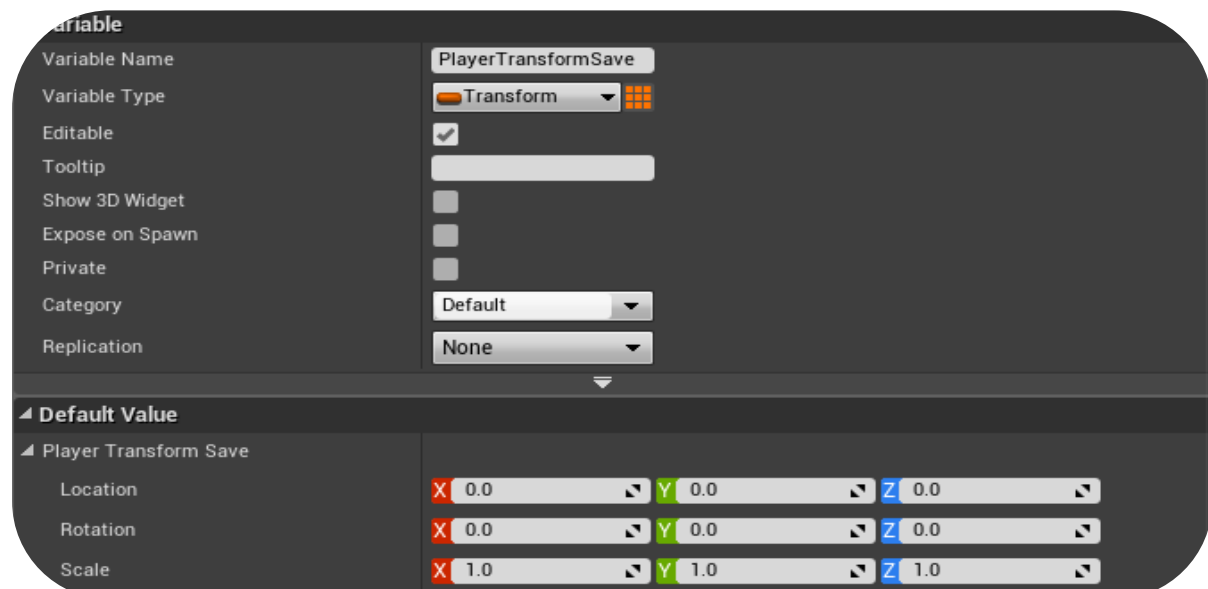
TwinDarkness_Save: Is a custom blueprint created inside unreal engine 4, it may be populated with any type of variable, and this custom blueprint will auto create the external save file.

In this case it is populated with a Transform Variable, which will be used to store the player's location when they save in-game, by default:

Location: x, y and z should equal to zero

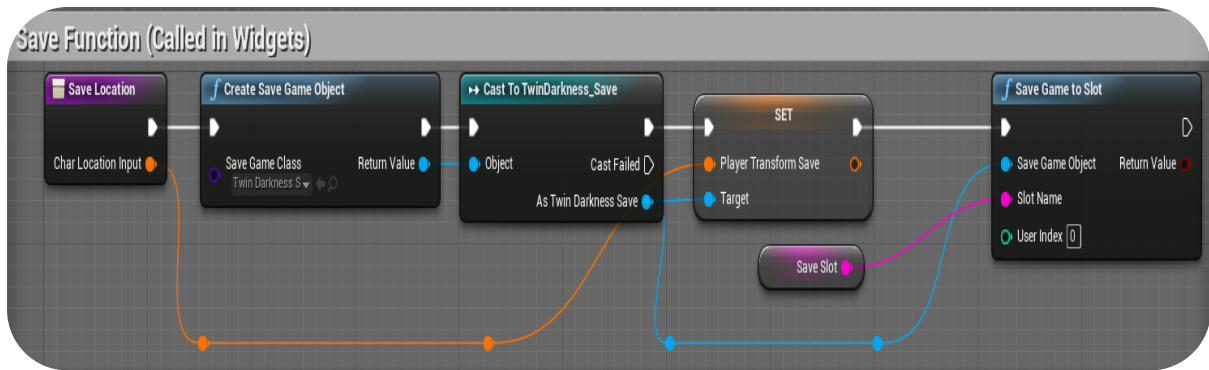
Rotation: x, y and z should equal to zero

Scale: x, y and z should be equal to one (scale will not be altered)



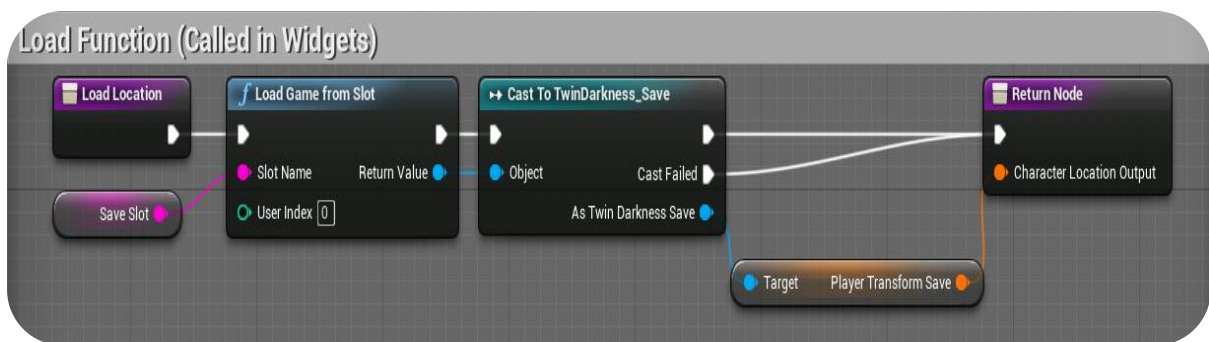
Save function: Custom function called when the user wants to save their current location in-game.

Location, rotation and scale are set inside the TwinDarkness_Save transform variable.



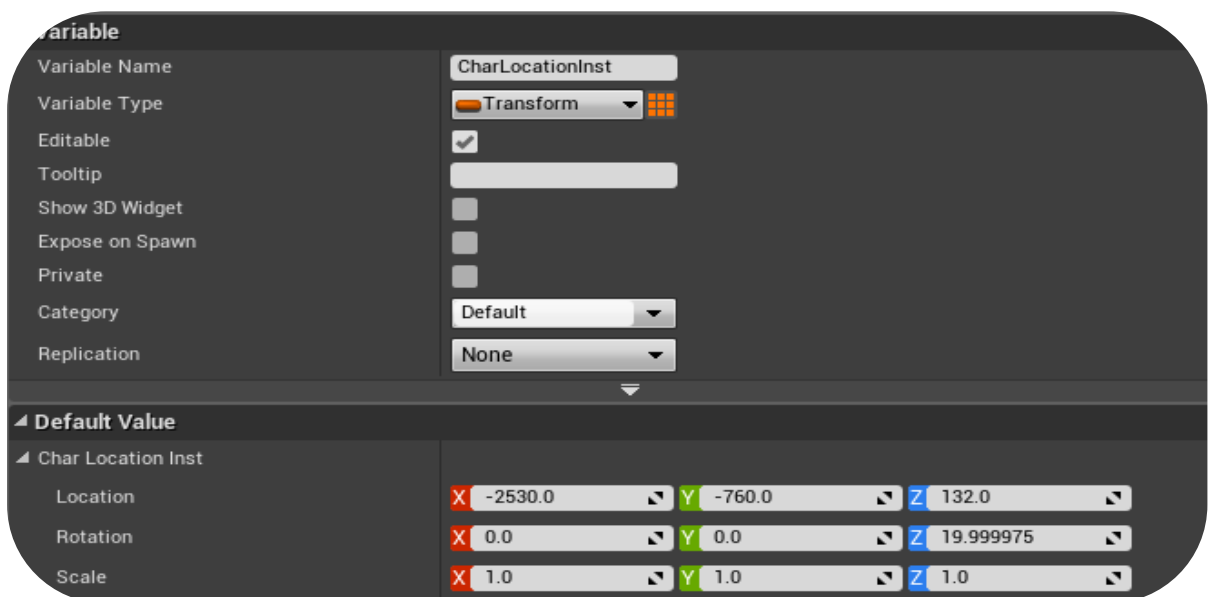
Load function: Custom function called when the user wants to load their current location in-game.

Location, rotation and scale are got from within the TwinDarkness_Save transform variable



Game Instance: Custom game instance blueprint that is updated and used to set the characters in-game location when the player loads

By default - Location, rotation and scale must be set to the characters starting location (*If there is no saved game file being used, the character will be default loaded to the level start location*)



In-Game Systems Implementation

This section of the implementation will cover the in-game systems. These systems are crucial to the gameplay, story and user experience.

Common for all in-game system interactions:

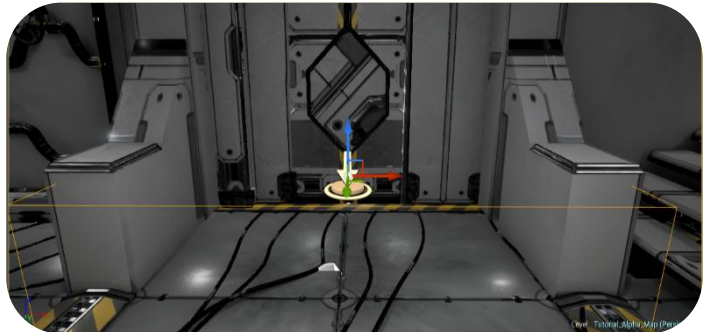
There is the same custom key presses (mapped to unreal engine) to trigger the interaction.

E press when interacting with a keyboard and mouse setup

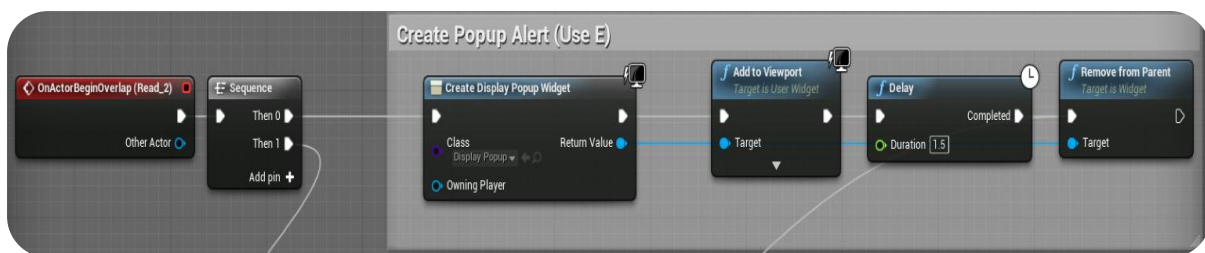
Gamepad Face Button Left when interaction with a Controller (Xbox)



To control what interactions trigger and when, there are unique trigger boxes associated with all in-game systems, see example image (showing the setup of a trigger box around a door)



To inform users that there are objects to interact with, there is an alert popup message that will appear on the users screen, informing them to press the interaction key, this is linked directly to each trigger box zone, see example image.



Doors implementation

Controls, Maintains and Implements:

- **Door controls:** Allows the in-game user to open and close doors (same setup for: double doors, single doors and vent doors)

Implementation & Screenshots:

Door interactions are an InputAction event, triggered by a unique trigger box. The Gate function is used to open and close that interaction (similar to a branch in some ways)

When the user is not overlapping with the trigger box – the gate function is closed not allowing them to interact (pressing the interaction key will do nothing)

Twin Darkness Technical Report

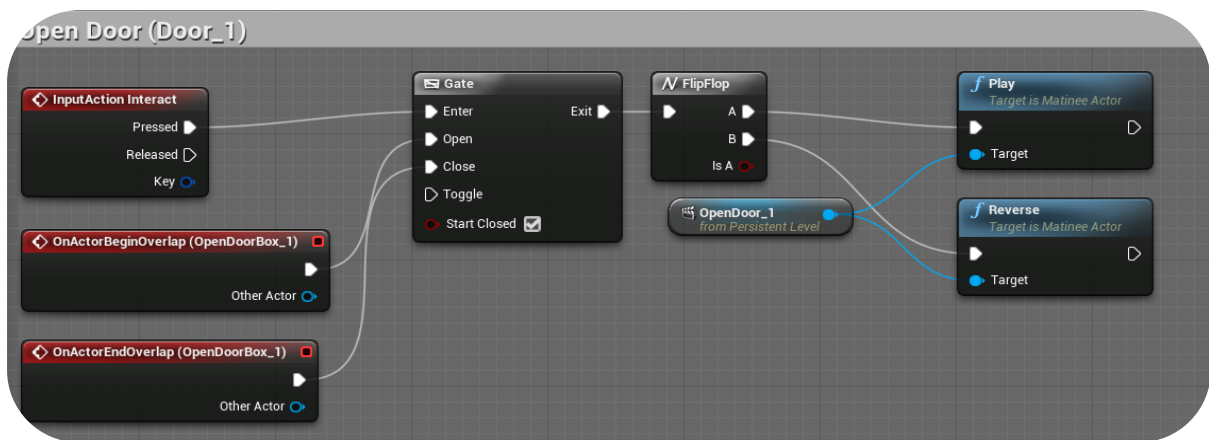
When the user is overlapping with the trigger box – the gate function is open allowing the interaction key to work.

If the gate is open and the interaction key is pressed a flipflop command is called which will trigger A branch first in this case it will play a matinee. (*Opening the door*)

(Matinee – animation tool provides the ability to animate the properties of Actors over time, to create either dynamic gameplay or cinematic in-game sequences.)

If the interaction key is pressed again it will trigger B which will play the same matinee but in reverse (*Closing the door*)

(*Flipflops can be triggered indefinitely*)



Readable Objects implementation

Controls, Maintains and Implements:

- **Readable in-game objects:** Allows the in-game user to open and read files on a number of in-game models (*in-game models: Tablets and Computer Screens*)

Implementation & Screenshots:

Read in-game object interactions are an InputAction event, gate function is used to open and close that interaction (**Identical start-up to the above door implementation system**)

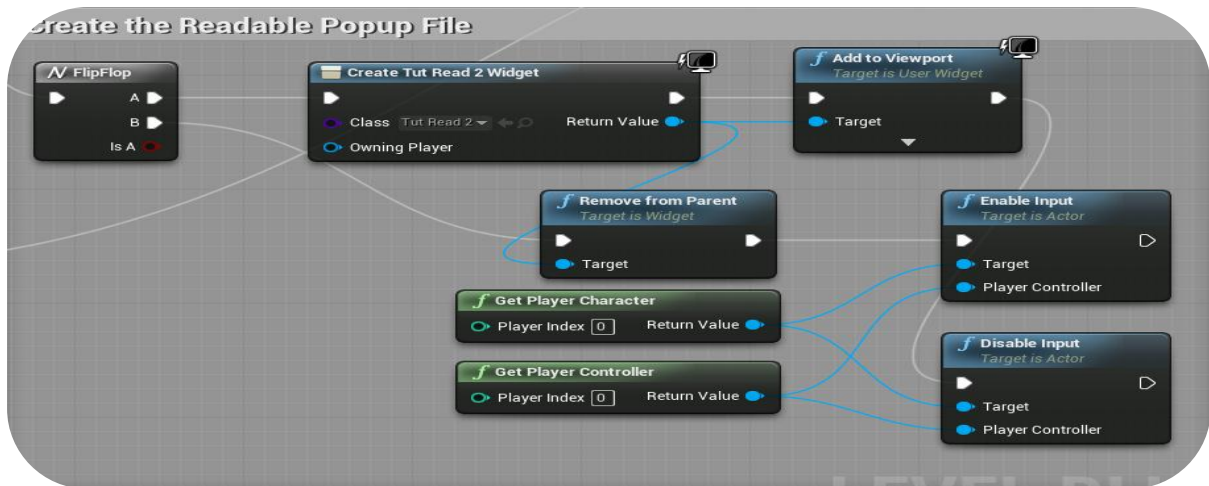
But in the case of readable in-game objects, the output branches have a different setup.

If the gate is open and the interaction key is pressed a flipflop command is called which will trigger A branch first in this case it will create a custom readable widget to the users screen. (*This widget contain story or character information*)

If the user presses the interaction key again the readable widget will close.

(*Key feature: when the readable widget is open on the user screen, movement of the character is disabled, ensuring a realistic setting. Movement is enabled again once the user has closed the widget*)

(*User may still take damage and fail in this mode*)



Interactive Objects implementation

Controls, Maintains and Implements:

- **Interactive in-game objects:** Allows the in-game user to access in-game objects to trigger events around the overall level (such as opening a broken vent door, with a command from the mainframe) (in-game models: Computer consoles)

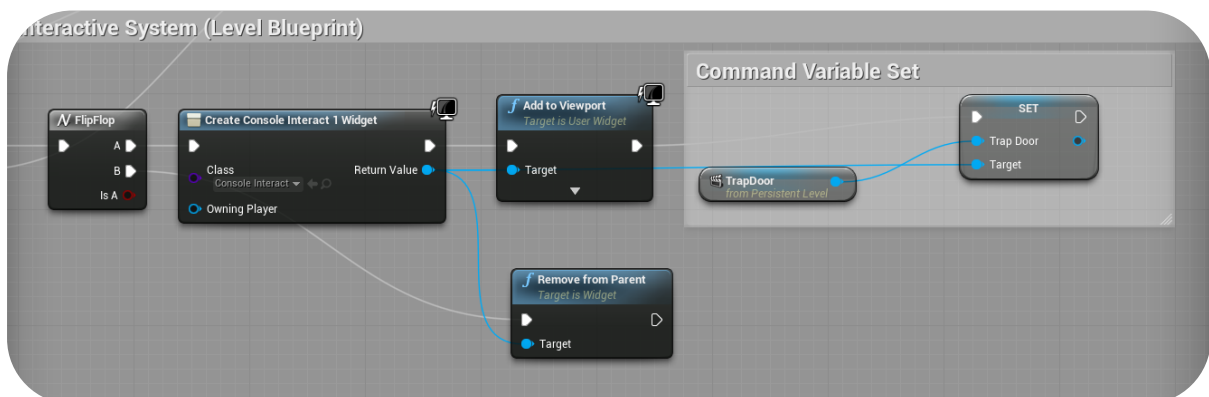
Implementation & Screenshots:

Interactive in-game object interactions are an InputAction event, gate function is used to open and close that interaction (**Identical start-up to the above door implementation system**)

But in order to allow the user to affect conditions of the level from a custom widget we need a two-step process this time.

The first step (*done in the level blueprint*) is: If the gate is open and the interaction key is pressed a flipflop command is called which will trigger A branch first in this case it will create a custom interactive widget to the users screen.

This function handles the front end side of the application (*what the user sees*), it also assigns variables from the level to the custom interactive widget (*in this case a matinee sequence for a trap door/vent door*)

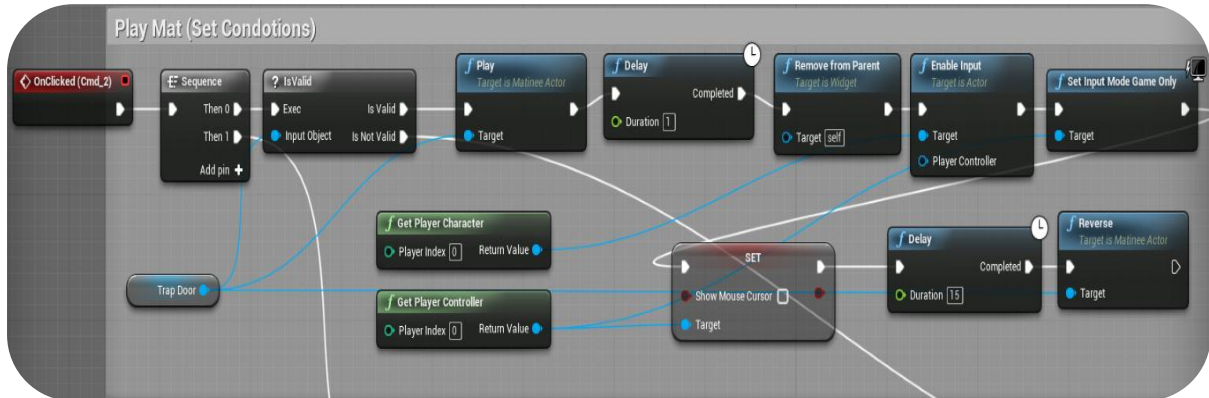


The second step (*done in the custom widget blueprint*) is: the back end setup and commands, once the user has access to the widget, they will be presented with commands that they can execute.

Twin Darkness Technical Report

These are done in the form of OnClicked button events, where we run a valid check on the variables being transferred from the level blueprint to the widget blueprint perform executing the rest of the command.

(In this case playing the matinee sequence to open the trap door/ vent door)

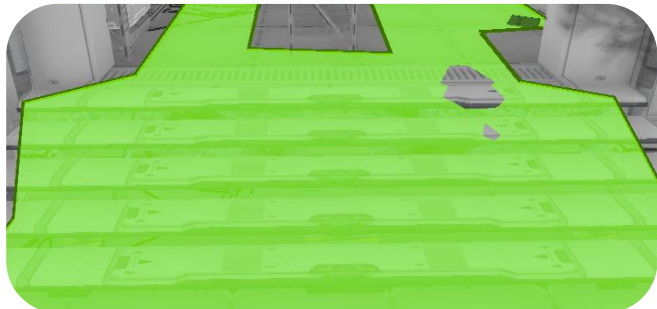


Artificial Intelligence Implementation

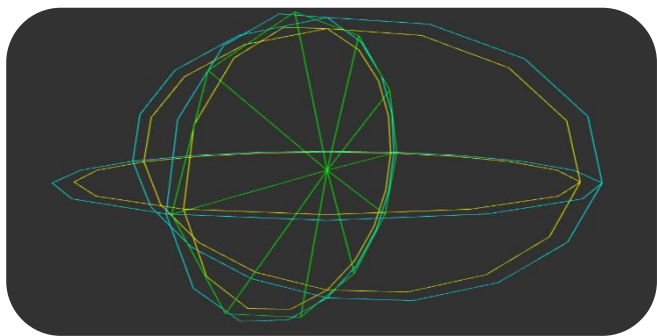
This section of the implementation will cover the A.I design and system. This system is crucial to the gameplay and story.

Unreal Engine 4 A.I Systems used:

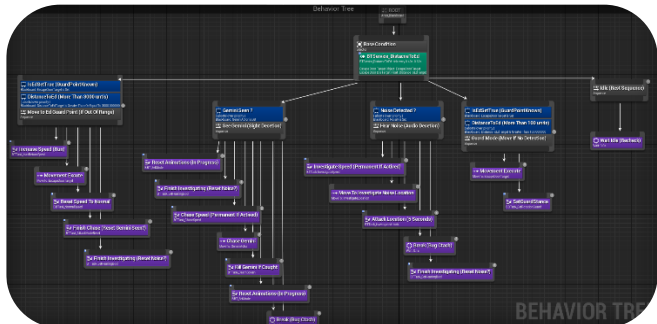
Navigation mesh (NavMesh): is an abstract data structure used in unreal engine 4 to aid agents in pathfinding through complicated spaces. It is placed over the level and its meshes, see example image.



Pawn Sensing: is a data component used in unreal engine 4 to aid agents in sight and audio detection. It is attached to the agents skeletal mesh and updates on regular intervals (My default is 0.2 second), see example image.



Behaviour Tree: is a behaviour support tool that uses a tree-like graph or model of behaviours and their possible consequences, including chance event outcomes, resource costs, and utility. See example image.



Aries implementation

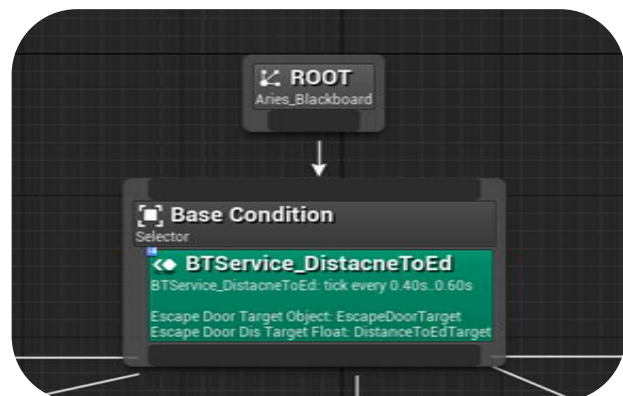
Aries is an A.I guard. Function of Aries is to guard the final door way, investigate sounds and sightings of main character (Gemini). Aries has access to a number of **key abilities (Task Blueprints)** & **conditions (Task and Service Blueprints)** built and developed within its **five branch** behaviour tree.

Abilities and Conditions:

- **Knowledge & Awareness of Guard Location:** Aries is aware of its distance from its guard point and will return if Aries is lured to far from it.
- **Sight Detection:** Aries has line of sight detection that refreshes on 0.2 second intervals. (*Line of sight means that Aries must directly see the main character*), if Aries sees the main character its base speed is increased.
- **Audio Detection:** Aries has volume audio detection that refreshes on 0.5 second intervals. (*The user can trigger unique sounds on the level*). If Aries is in range: it will increase base speed and move to the sound location, investigate the sound location and then return to the guard location.
- **Guard Stance & Idle:** Aries enters a unique guard stance animation when it is at the location of its guard point, if Aries enters the guard stance, the behaviour tree enters an idle mode.

Behaviour tree, branch implementation:

Root & Base condition: There is one base condition, directly under root. This is a service function that continually updates and tracks the distance between the guard point and the Aries A.I agent. The distance is stored within a distance variable for use further down in the behaviour tree, and is refresh on a 0.40 - 0.60 second interval.



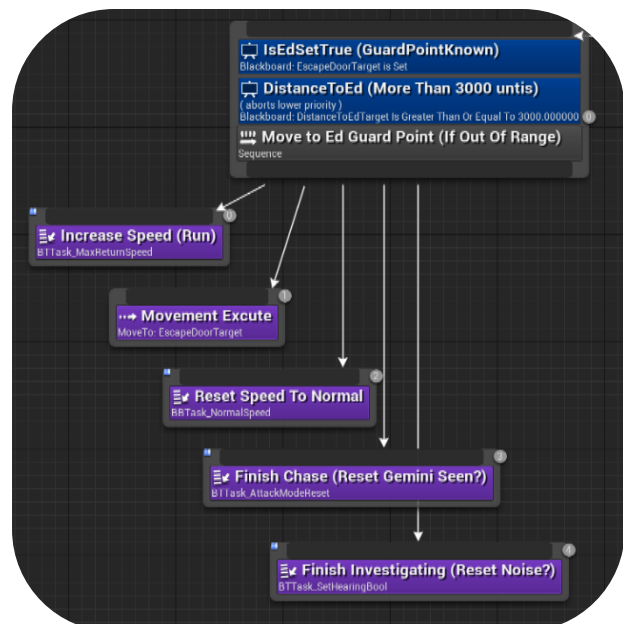
First Branch & Conditions: (*Demonstration video - https://youtu.be/H_qcB6lrIqs*)

Conditions needed for activation:

- If the guard point is set and known to the Aries behaviour tree.
- If the Aries agent, is greater than or equal to 3000 unreal units away from the guard point. (3000 unreal units equal 30 meters)

Activated abilities & conditions:

- Increase movement speed of the Aries agent to maximum (350)



Twin Darkness Technical Report

- Move back to the guard location point, at the newly increased movement speed (350)
- Once at guard location, reset movement speed back to base level (250)
- Reset both sight and audio sensing components (*Needed to prevent infinite loops*)

Impact of overall behaviour: *This branch if activated has **(aborts lower priority)** build into its activation, meaning all other branches and functions are stopped, when this branch it activated.*

Second Branch & Conditions: (*Demonstration video - https://youtu.be/FjNBAOm9_ps*)

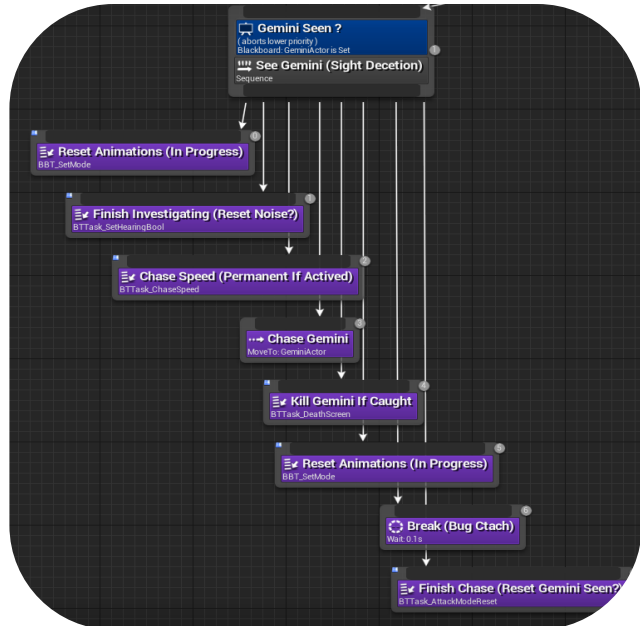
Sight Detection (aborts lower priority branches)

Conditions needed for activation:

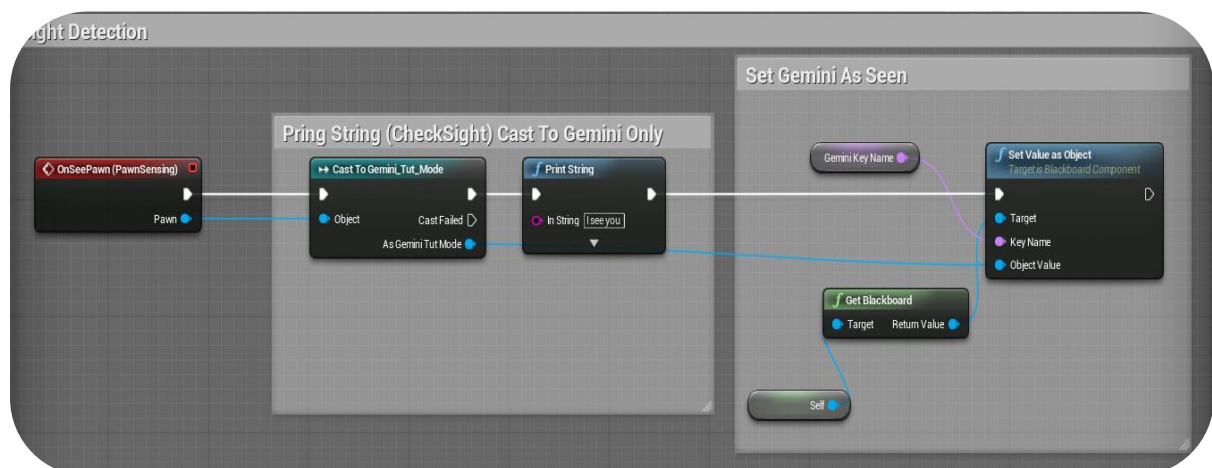
- If the Aries agent detects the main character using line of sight detection system of the pawn sense component.

Activated abilities & conditions:

- Reset all animations currently in progress, to insure a smooth transition.
- Reset audio sensing component. (*Needed to prevent infinite loops*)
- Increase movement speed of the Aries agent to chase speed (325)
- The Aries agent will pursue the main character (*Gemini*) continually across the level, unless the first branch is triggered or the Aries agent comes in contact with the main character.
- *If the Aries agent comes in contact with the main character:* the Aries agent begins an attack animation, and the main character is terminated and begins a death animation. (*Which will trigger a level restart*)
- Reset all variables and animations, once the player is killed



Pawn Sensing: Sight Component: Sets main character (*Gemini*) as the key object, and sets the blackboard value (*Variables used within the behaviour tree*) to that Gemini key.



Third Branch & Conditions: (Demonstration video - <https://youtu.be/Yo1LI7RV8w>)

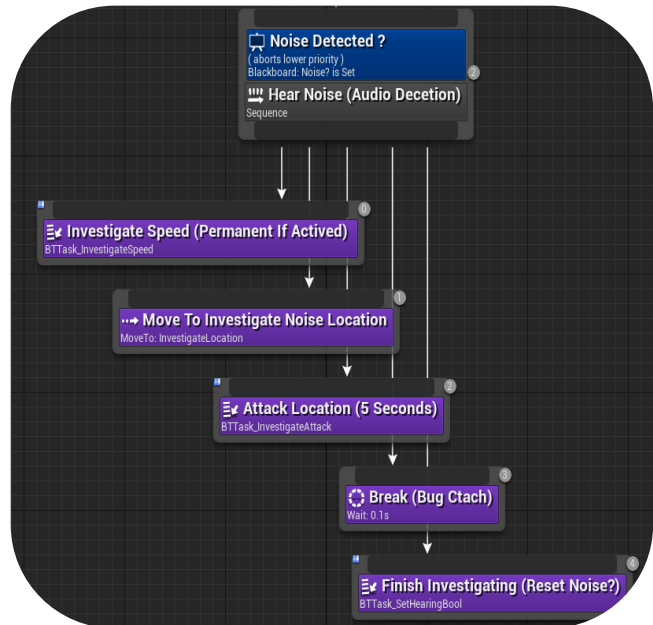
Audio Detection (aborts lower priority branches)

Conditions needed for activation:

- If the Aries agent detects a sound within field of hearing using the audio pawn sensing component.

Activated abilities & conditions:

- Increase movement speed of the Aries agent to investigate speed (300)
- Locate the source point of the sound and move to that location.
- *If the Aries agent is at the sound source:* Begin a custom attack animation at that location for the duration of five seconds.
- Wait 0.2 seconds and then reset the noise variable.

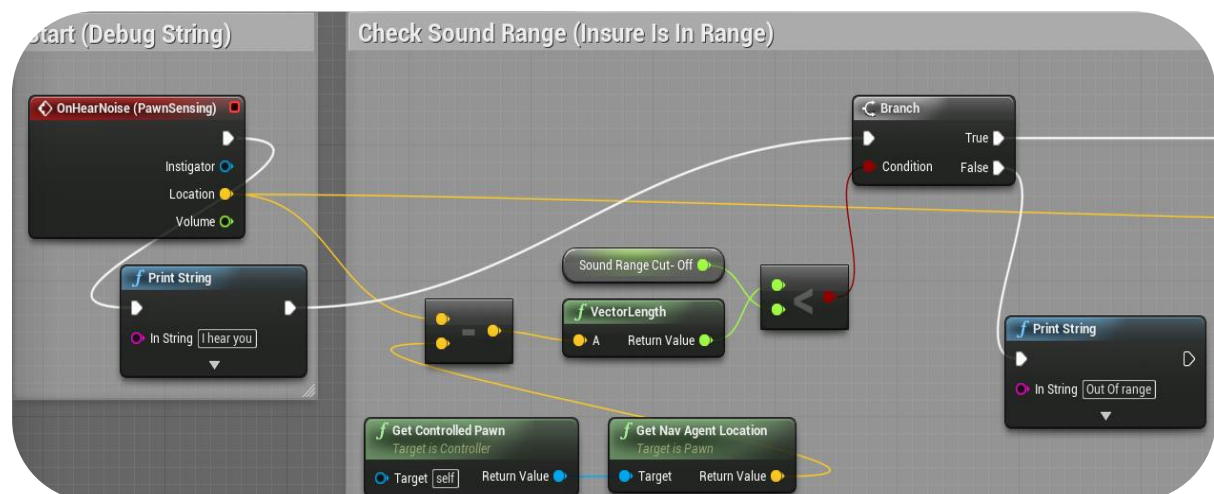


Once the noise variable is reset: Exit the branch, and allow branches four and five to take over, these branches will return the Aries agent to the guard point.

Pawn Sensing: Audio Component: Is broken down into two parts, first part of the audio detection is checking if the sound that was triggered is within the allowed range of the Aries agent.

First we get the location of the sound and compare that with the location of the Aries agent, we then check the distance against a set variable called Sound Range Cut-Off which is always set to 2990. Since branch one will override all functions at 3000.

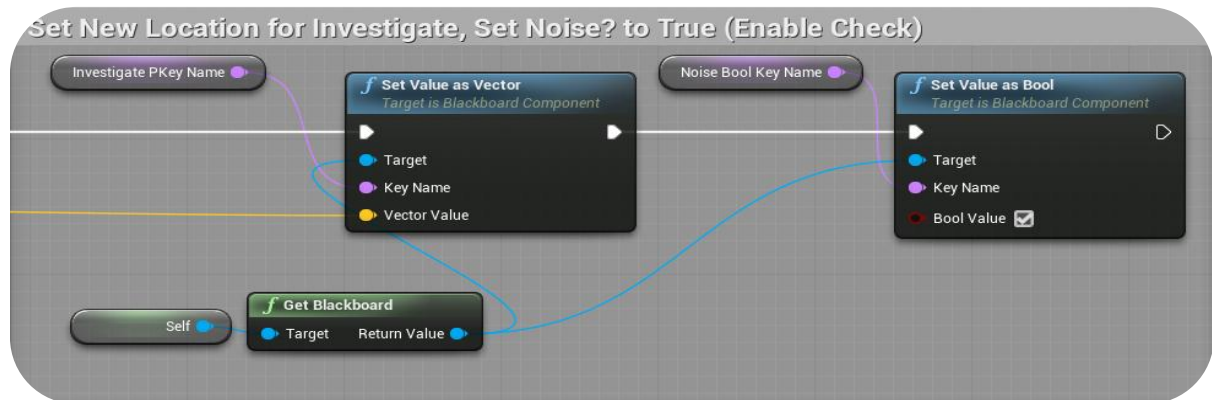
This is then set as a condition in a branch statement (*For false: No Action taken by the Aries agent*)



If true: the investigate location key is set to the location of the triggered sound and the Noise Boolean key is set to true, both these variables are then set to the corresponding blackboard values.

Twin Darkness Technical Report

(Allowing the Aries agent behaviour tree to act or not)



Fourth and Fifth Branch's & Conditions:

(Demonstration video -

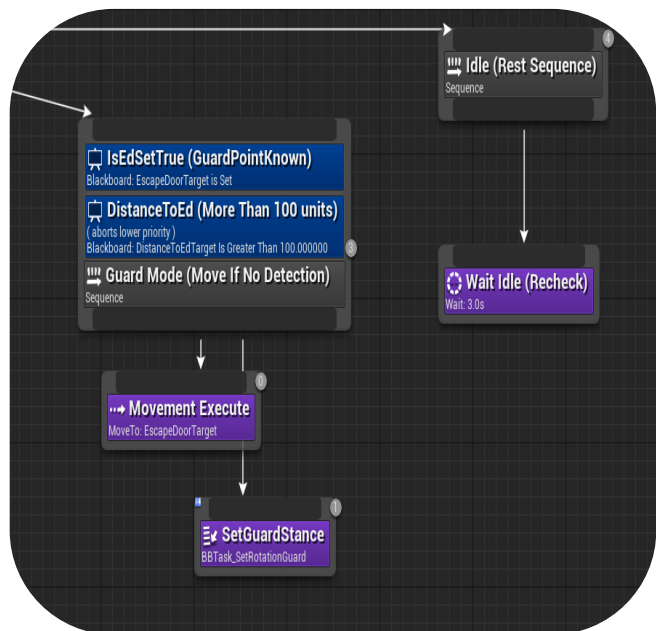
<https://youtu.be/aUKHuYsjxFO>)

Conditions needed for activation:

- If the guard point is set and known to the Aries behaviour tree.
- If the distance of the Aries agent from the guard point is more than 100 unreal units

Activated abilities & conditions:

- Move back to the guard location point at the base movement speed or at an adjusted speed if sight or audio had been triggered before
- *If the Aries agent is at the guard point:* Play and Finish the custom guard stance animation, and correct orientation of the Aries agent.



Once Guard Stand Animation Finishes: Enter the fifth branch, and enter idle mode which refreshes every 3.0 seconds.

Blackboard Variable Description:

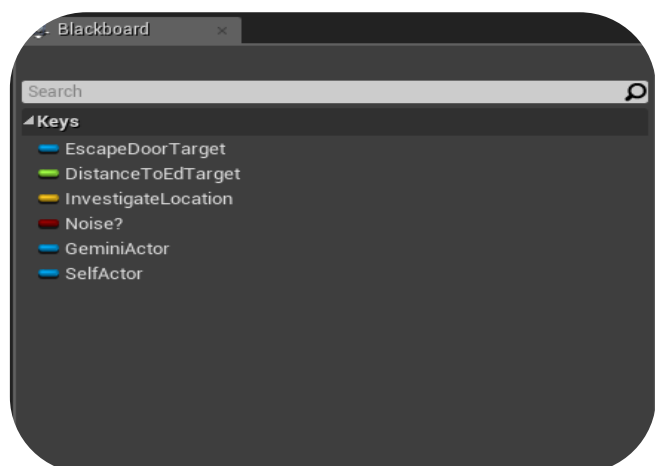
EscapeDoorTarget: Set point on the Map

DistanceToEdTarget: Distance float between A.I and guard point

Investigatelocation: Set point once a noise is detected, vector point x, y and z

Noise? : Boolean variable True or False

GeminiActor: Main Character, set when seen



Main Character Implementation

This section of the implementation will cover all the aspects of Gemini, the main character in twin darkness and the only character the user will play. This is the most important system, as the game would not be playable without a number of these key implementations.

Under this section heading we cover the base character animations and movement controls, along with the unique character abilities that have been developed.

Character Animation Blueprint implementation

Controls, Maintains and Implements:

- **Blend Space:** Enables the blending of animations and a fluid transition between animations.
- **Character animation blueprint:** Assigned to the Gemini character mesh, to enable realistic character animations. *(Without this blueprint, the character mesh would be permanently set to its import pose)*

Implementation & Screenshots:

Blend Space: In the case of twin darkness blend spaces where used to blend the idle, walk and run animations together and the idle crouch and crouch walk animations, so the user can go from one animation to another realistically and fluidly.

Parameters are set on the right hand side, to key parameters are:

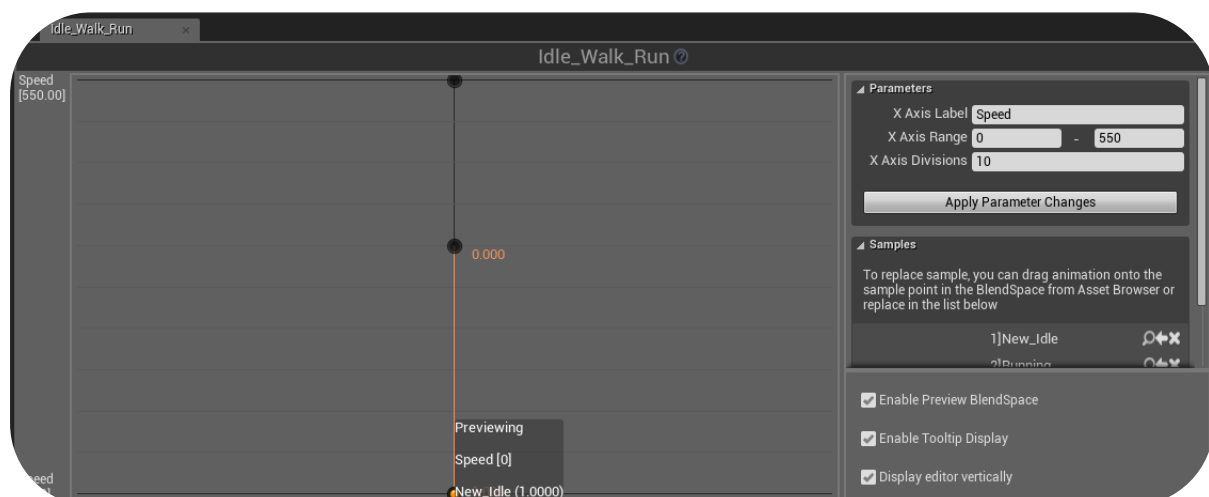
- Range of the graph: Speed value from the start animation (Idle) to the end animation (Run)
- Number of divisions: Number of division lines where animations can be placed.

Idle is placed at speed value 0, meaning if the player is not moving the idle animation is playing)

Walk is placed at speed value 325, meaning if the player begins moving from idle to walk, the animations for idle and walk begin to blend, until only walk is playing)

Run is placed at speed value 550, meaning if player begins moving from walk to run, the animations for walk and run begin to blend, until only run is playing)

(Blend space for idle crouch to crouch walk follows the same setup just with adjusted values)



Twin Darkness Technical Report

Character Animation blueprint has five variables:

- **Speed:** Is a float variable that changes based on the users movement speed. Depending on the speed a different animation blend will play, these blends are setup within blend spaces.
- **IsInAir:** Is a Boolean variable that allows us to tell when the user is on or off the ground. This is used for triggering the jump animation
- **IsCrouched:** Is a Boolean variable that allows us to tell when the user is crouched or uncrouched. This is used for triggering the crouch animation.
- **CrouchSpeed:** Is a float variable that changes based on the users movement speed when crouched. Depending on the crouched speed a different animation blend will play.
- **Dead?:** Is a Boolean variable that allows us to tell when the user has been killed, and trigger the death animation.



(These variables will be used throughout the blueprint)

Character Animation blueprint has two unique sections:

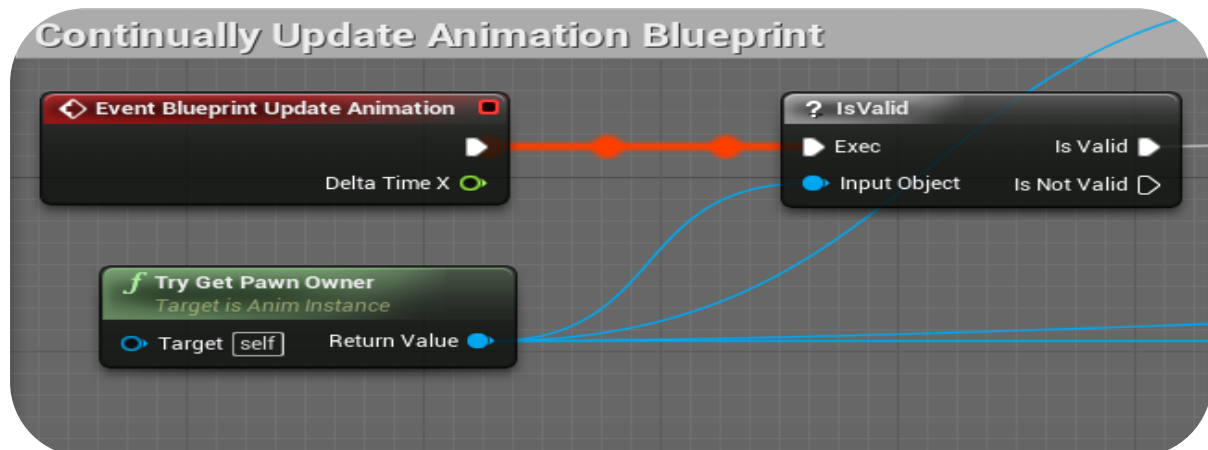
- Event Graph
- AnimGraph (*Animation graph*)

Event Graph (*Handles the getting and setting of all required variable information for the blueprint and for the main character*)

Event blueprint update animation is continually updating the know variables within the event graph each time the user performs an action with the main character.

Which is found using the try get pawn owner function (*All required variables of the character are received through this function*).

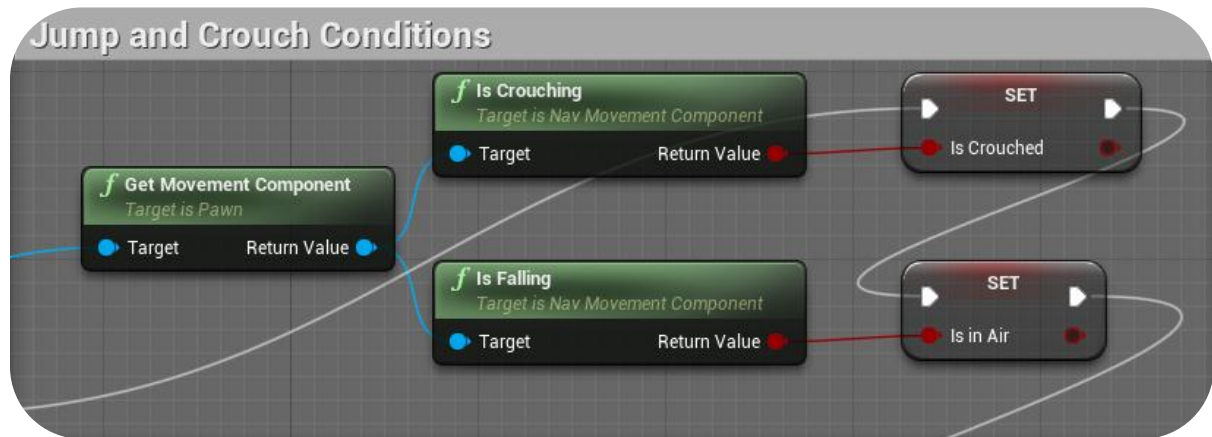
Both of these initial functions are connected to an IsValid as the input and Execute command, allowing only valid updates to be passed to the event graph.



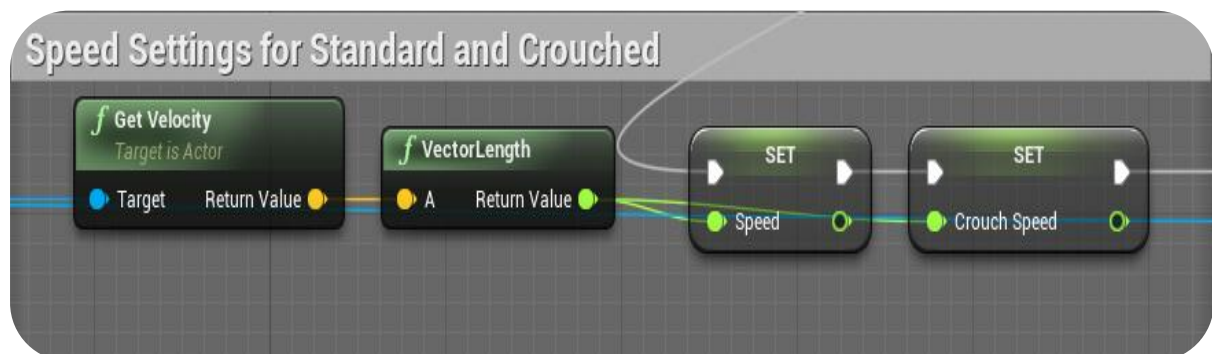
Twin Darkness Technical Report

The first event check is done using the get movement component, this component allow us to check if the character is crouching: If true then the IsCrouched variable is set to true and the animation will be played

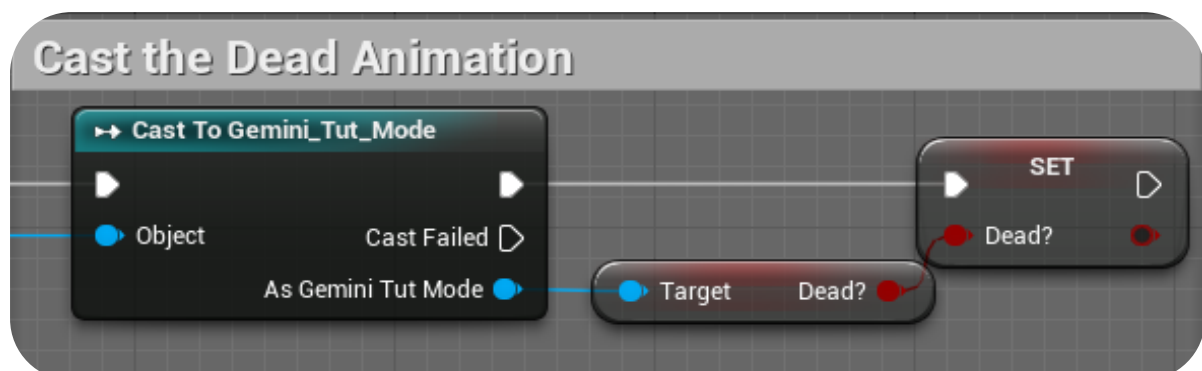
It also checks if the character is falling, hence if they are in the air: If true the IsInAir variable is set to true and the jumping animation will play



The second event check is done using the get velocity, this component allows us to check at what speed if any the character is moving. Depending on that speed a different animation blend is played.



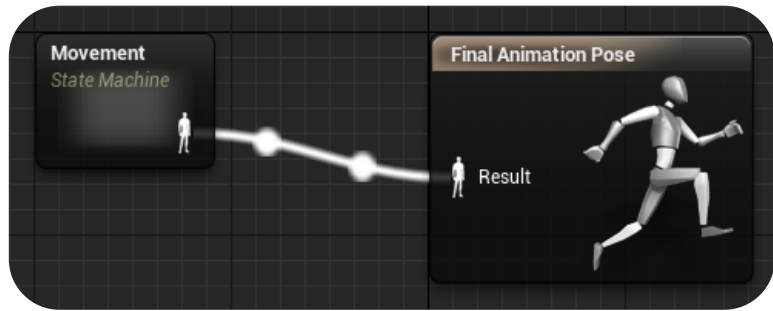
The third and final event check is done using a cast to the main character, this allows us to set conditions with the character blueprint of when and how dead should occur, and this is then set to the Dead? Boolean variable triggering the dead animation.



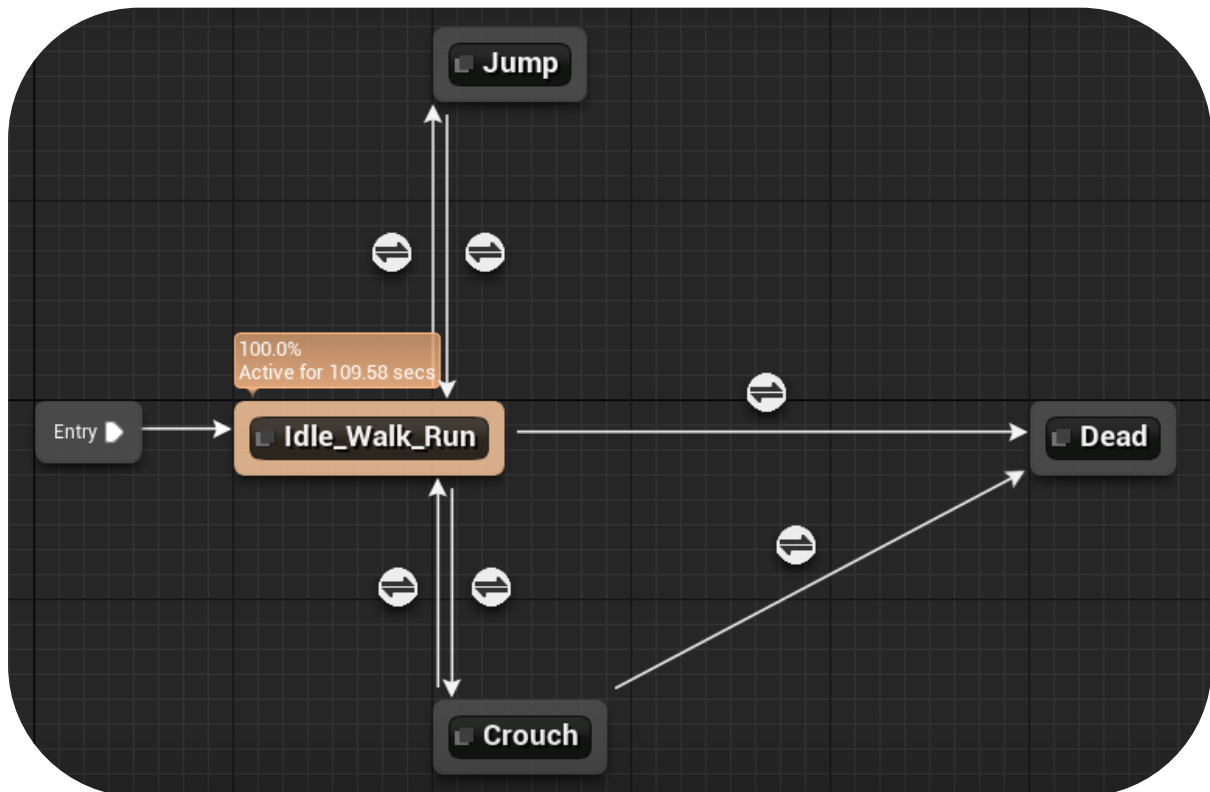
(All these gets and sets are needed within the event graph to allow the right animations to play under the right conditions)

AnimGraph (Evaluate a final pose for the skeletal mesh of the main character for the current frame)

All animations are contained within the movement state machine, then linked to the result of the final animation pose.



The main graph of the movement state machine handles all animations.

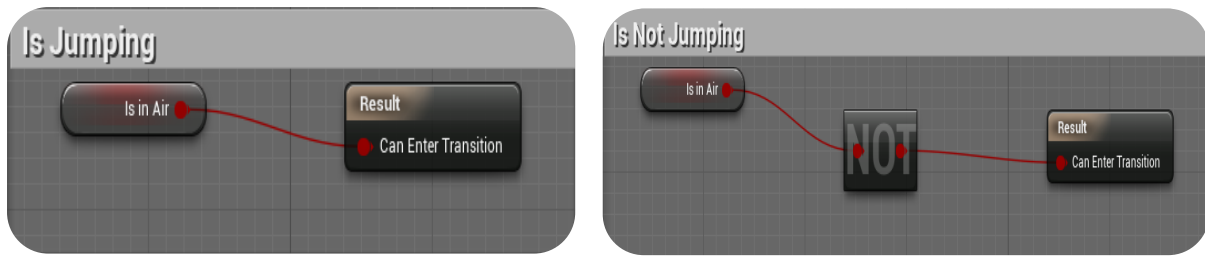


Entry point equals to the *idle walk run* state, by default this state is always active. The state is affected by the speed variable.

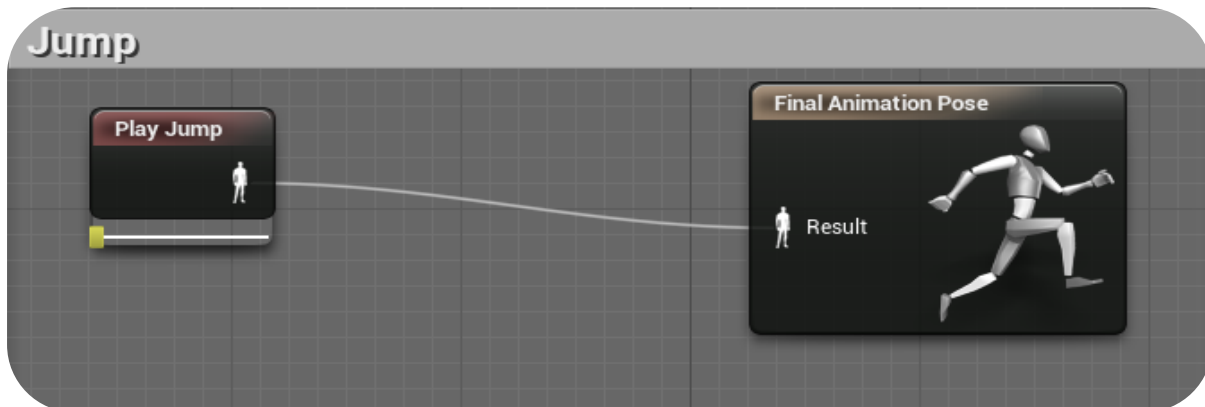


Next there are transition rules between all the animation states, these rules use the Boolean variables to determine if they should play their state animation.

One rule going to the new state and one rule returning to the previous



If one of these rules are met, for example if, IsInAir is true the player is jumping and the jump animation state will trigger.



Crouch combines the above methods one the user must be crouched and then at what speed they are moving will determine the animation blend.

Death only requires one transition rule since the character does not go from dead state to the idle walk run state or crouch state

Movement Input Controls & Setup implementation

This section of the implementation will cover the binding setup for all movement controls both keyboard and controller, and the main character blueprint setup for movement.

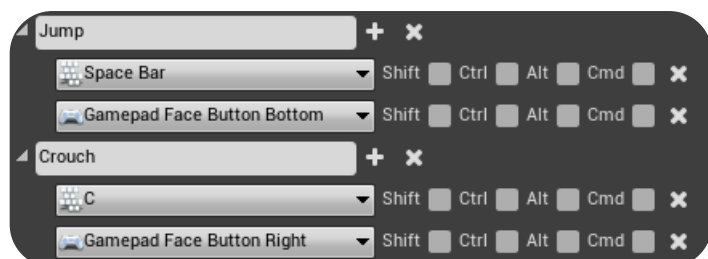
Controls, Maintains and Implements:

- **Movement key bindings:** Sets the keys needed to trigger all of the character movement.
- **Character blueprint movement:** Implementation for the character to move when the correct key is pressed.

Implementation & Screenshots:

Movement key bindings

The first key bindings are for Jump and Crouch these are action mapping keys, meaning they perform a specific action. There is a key for keyboard and a key for controllers (*specifically Xbox 360*).



Twin Darkness Technical Report

The next set of bindings are for moving the character in the forward or backward direction. These are axis mapping keys.

The reason that forward & backward are combined, is because backward movement is forward movement inverted (scale -1)



The next set of bindings are for moving the character left or right. These are axis mapping keys.

The reason the left & right are combined, is because left is right inverted (scale -1)



The next set of bindings are for the characters turn rate of the character camera.

Turn rate is inverted for left (scale -1)



The final set of bindings are for:

- Turning with a mouse
- Look up rate with a controller
- Look up with a mouse.

Mouse's and controllers do not need inverted values



Character blueprint movement

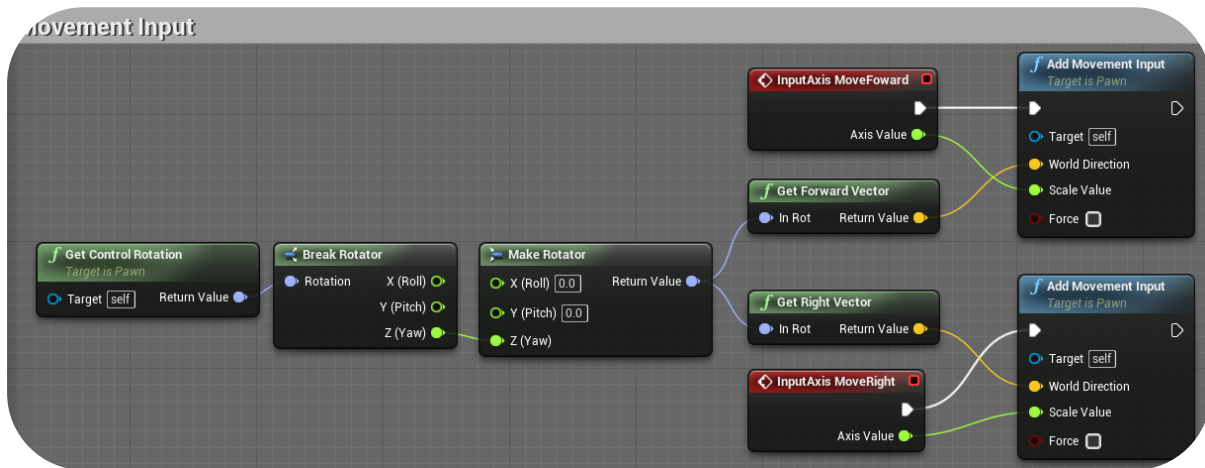
Get control rotation (control can be keyboard, mouse or controller) is called and initially broke, so we can access the z axis value which is (Yaw), we then make a new rotator, only connecting the z axis value.

We then connect the new rotator into both the forward vector and the right vector getters.

Next we call two of our new axis mapping key events Moveforward and MoveRight, these events then call an Add movement input function.

Axis values from the key events are connected to the scale value of the movement input functions.

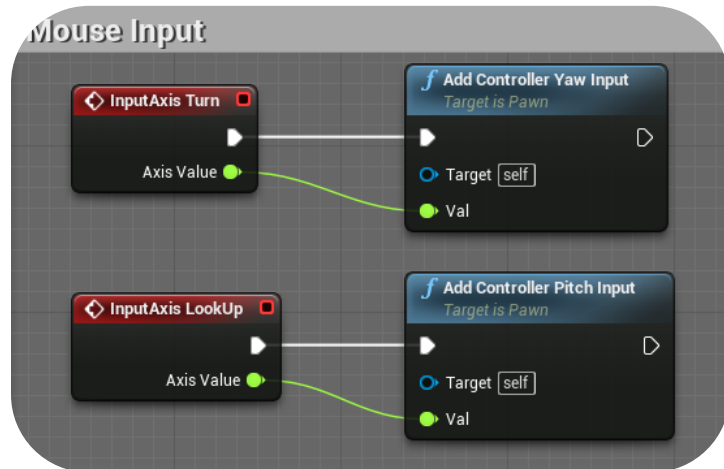
Getters for forward and right vector are connected to the world direction of the movement input functions. (This blueprint setup allows for forward, backward, left and right character movement)



Next we the mouse control setup: We take the input event turn and add controller Yaw input

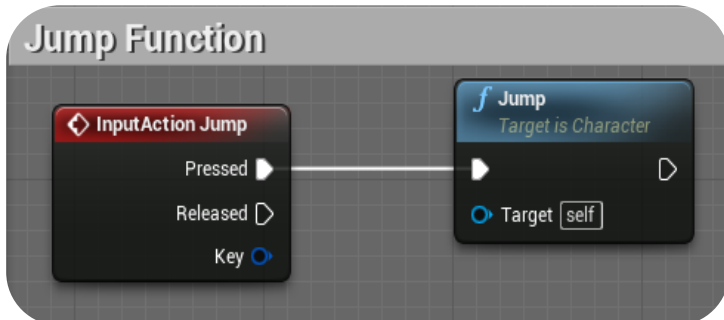
We take the input event lookup and add controller pitch input

(This blueprint setup allows for the left and right movement of the camera and the lookup and lookdown movement of the camera)



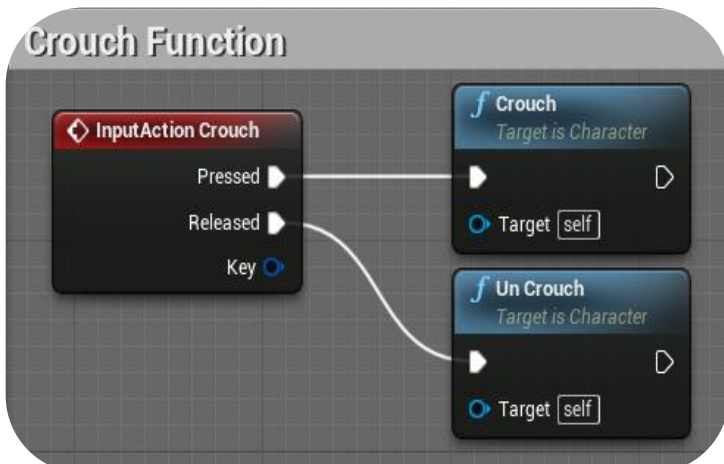
Next we the jump setup: Unreal has default functions for jump, here we just contact the jump event to the jump function

(This blueprint setup allows for the character to jump)



Next we the crouch setup: Unreal has default functions for crouching and un-crouching, here we use the input event crouch connect pressed to crouch (meaning as long as the crouch key is pressed the character will remain crouched)

Released is connected to un-crouch (meaning when the key is released the character will un-crouch)



Twin Darkness Technical Report

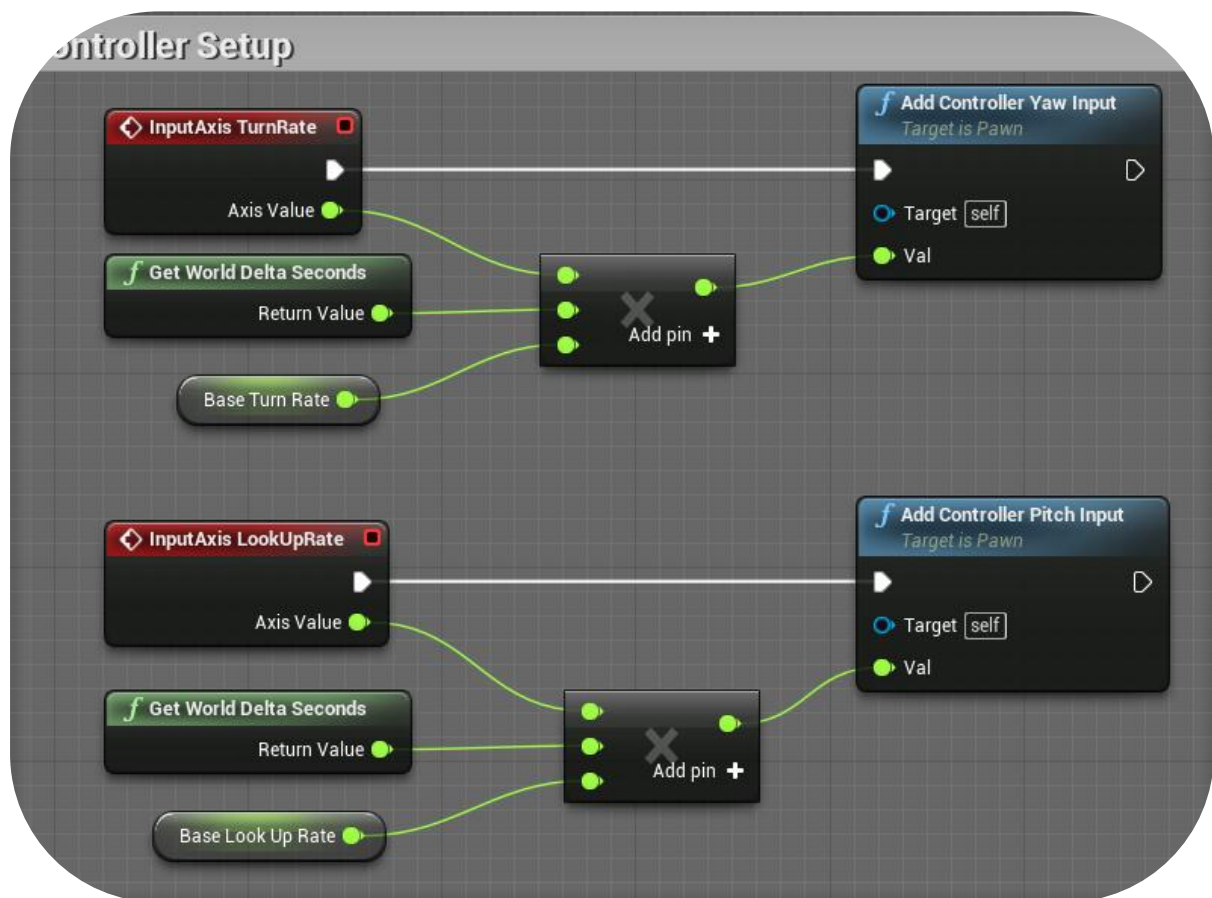
Finally we have the controller setup, similar to the mouse setup in its final function of moving the character camera.

But we need three extra values in order to insure the right movement of the camera.

- Get World Delta Seconds: time tick in game
- Base turn rate: industry standard set by epic games is 150
- Base look up rate: industry standard set by epic games is 150

These values are combined inside a multiplier then connected to the controller yaw and the controller pitch

(Base turn rate and base look up rate, may be adjust depending on specific controllers for a better user experience, from development testing I have left these at the recommended values)



Sprint System implementation

Sprinting is key for the user, it allows them to reach closing doors, evade the Aries A.I if detected and move around the level faster.

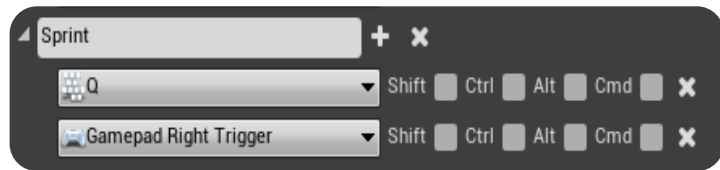
Controls, Maintains and Implements:

- **Sprint System:** Allows the user to press and hold a mapped key that will increase the characters movement speed to max for a limit time. As the user holds the mapped key and moves the sprint bar drains from 100 to 0, at which time the user will be unable to sprint until it recharges.

Twin Darkness Technical Report

Implementation & Screenshots:

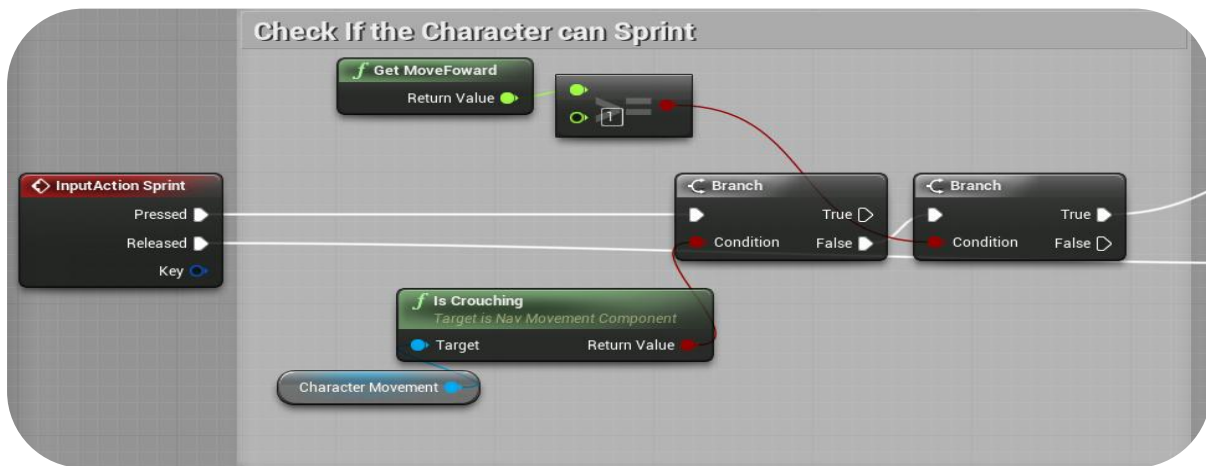
The sprint keys are mapped to unreal. The user will press and hold these keys to trigger the sprinting ability, they are mapped both for keyboard and controller.



Sprint function is an input action event, before the user begins sprinting the system checks if the character is meeting two conditions:

- Character must be moving forward in order to sprint
- Character must not be crouched (*cant sprint while crouched*)

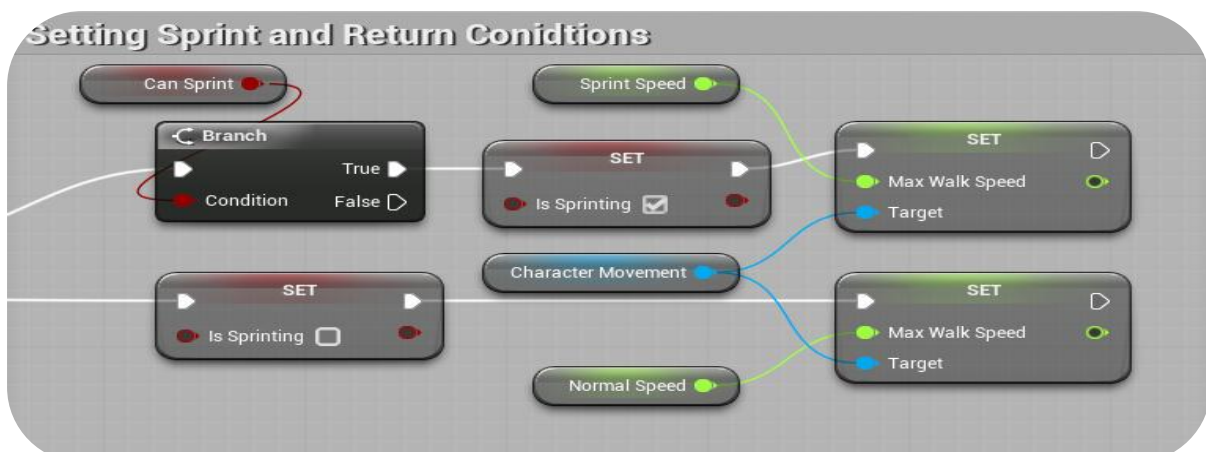
(Character must meet both conditions in order to sprint, otherwise pressing the key will do nothing)



If the character meets both of these conditions, one final branch check is called getting and setting can sprint to true.

Character movement is used to call the set function for max walk speed, the sprint speed variable is connected increasing the walk speed of the character to 550, starting the sprint system

If the user releases the sprint key, is sprinting is set to false, the set max walk speed is returned to the normal speed if 325, stopping the sprint system



Twin Darkness Technical Report

The next part of the sprint system is the stamina bar and regeneration, before the stamina bar can drain or regen a number of branch conditions must be met,

For the stamina to drain the character must be:

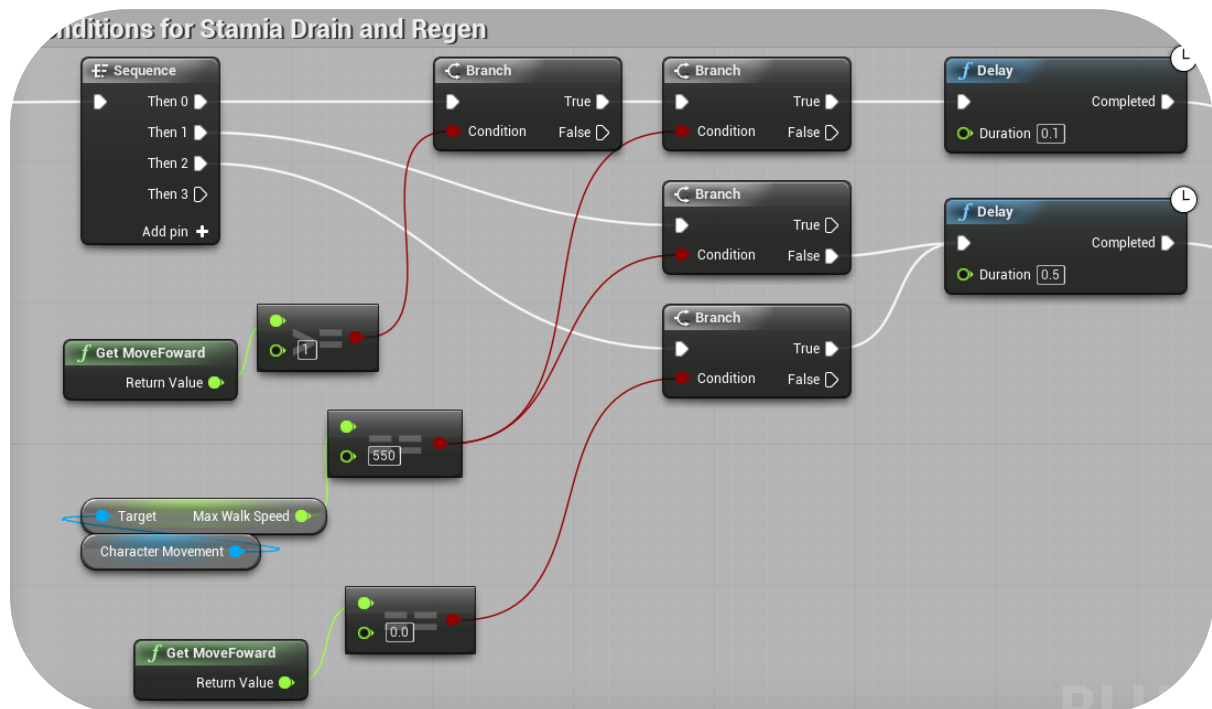
- Must be moving forward
- Must have the character movement speed of 550 (*meaning they are sprinting*)

There is then a 0.1 second delay before the drain begins, note these conditions must remain in order to continue the drain.

For the stamina to regen the character must be:

- Not moving
- Must not have the character movement speed of 550 (*meaning they are not sprinting*)

There is then a 0.5 second delay before the stamina bar will regen, note only one of these conditions must remain in order to continue the regen of the stamina bar.



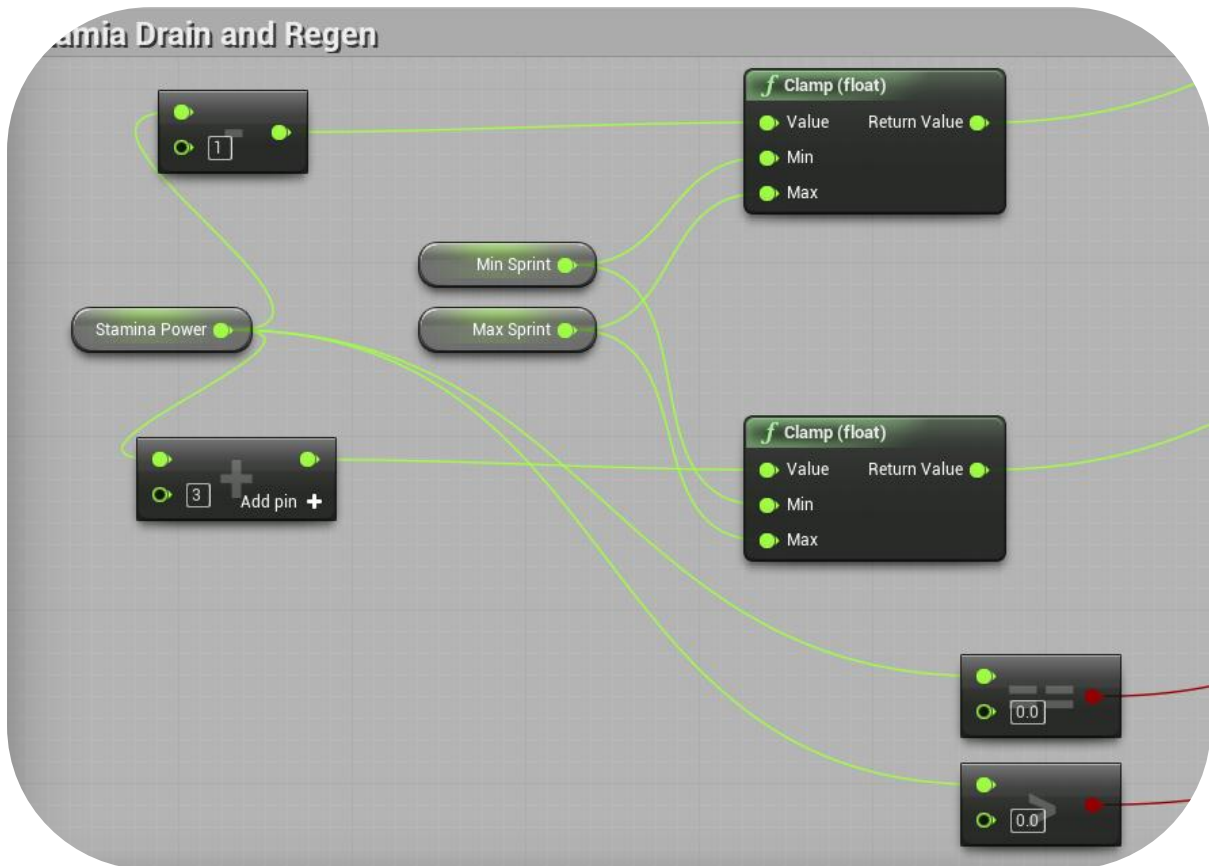
Next we have the base float variables that make up the drain and regen system,

Stamina power: Is set to the value 100, this is the variable that will be drained or regened depending on the above conditions, in essence this is our stamina bar (*Widget UMG bar setup explained further down*)

Min sprint: Is set to the value 0 meaning that it will not drain passed 0

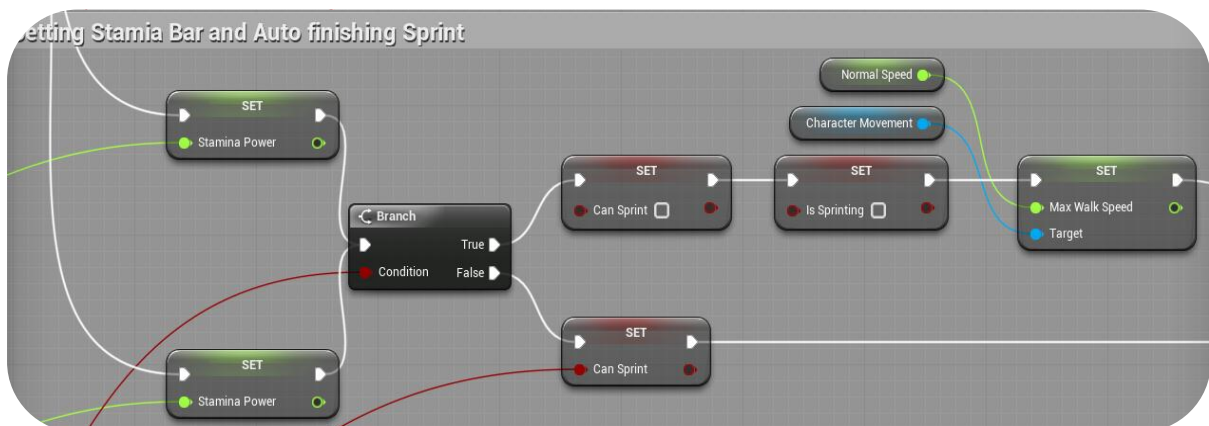
Max sprint: Is set to the value 100 meaning that it will not regen passed 100

To date the current setup is: 1 point of stamina is drained every second, 3 points of stamina is recovered every second.

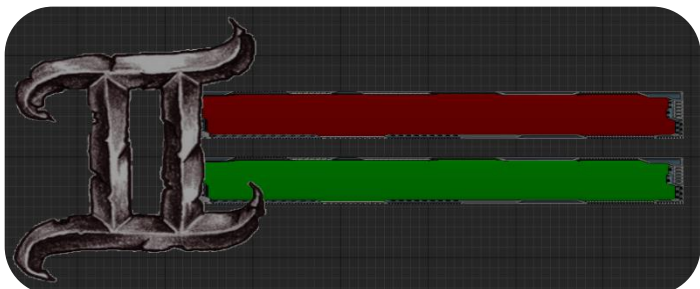


Next we have the controls for shutting off the sprint function if the stamina bar reaches 0, we use the (Can Sprint Boolean) and the (Is sprinting Boolean) to auto end the sprint function based of conditions linked to the stamina power variable.

Once the character is out of stamina the characters movement speed is set back to normal (325)

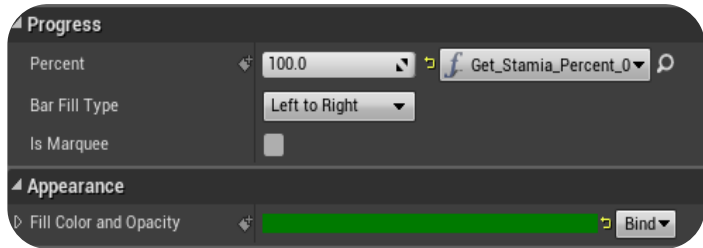


Next we have the display to the users this is setup up using a HUD widget with a progress bar attached, the **green bar** seen in the picture is the stamina bar that will drain and regen.

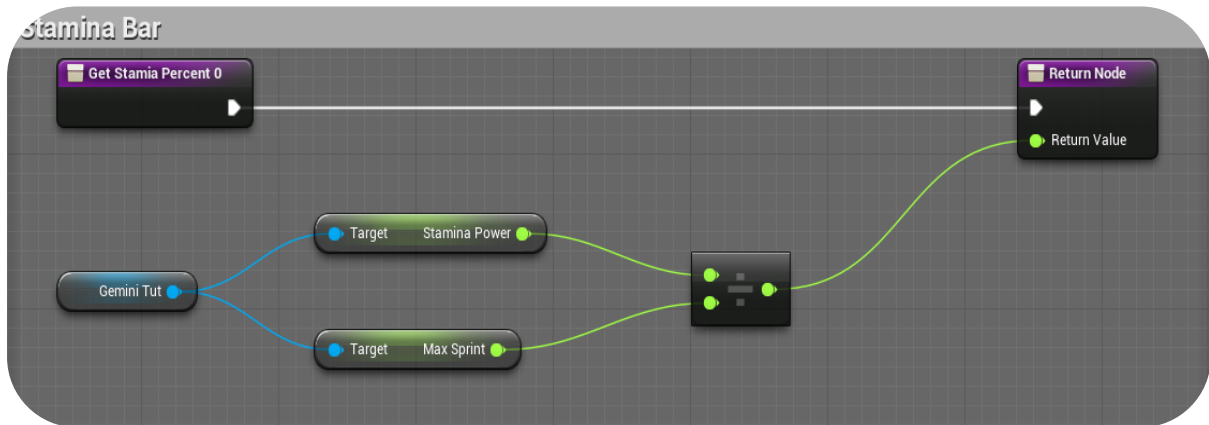


Twin Darkness Technical Report

In order for the UMG process bar to drain it must have the same value as the stamina power value in this case 100.



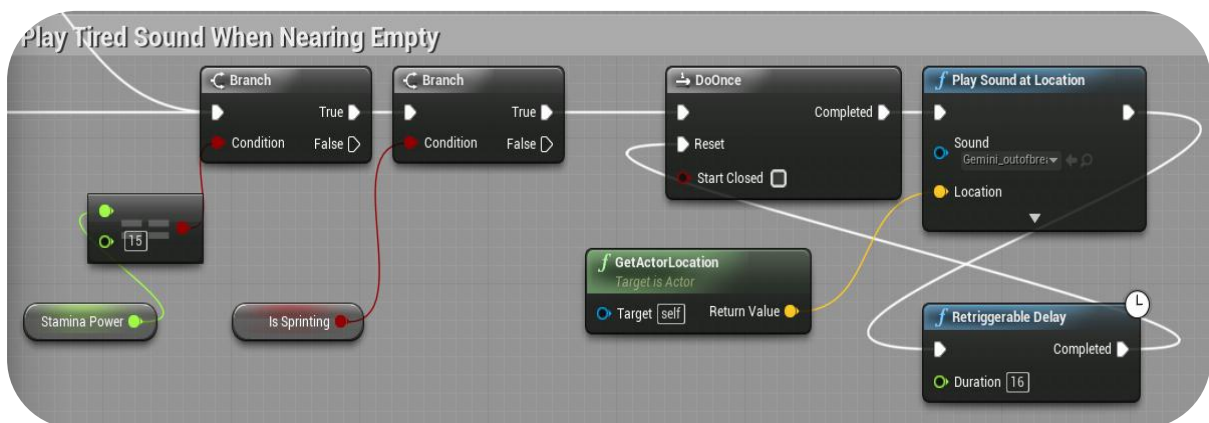
We then attach a function to the process bar itself, we call the character blueprint and export two variables Stamina power and Max sprint we divide the two together and connect the result to the return node.



The final part of the sprint system is alerting the user when they are low on stamina, this is done with a sound trigger and two conditions:

- The characters stamina bar must be at the value of 15
- The character must be sprinting

We then trigger this sound once, to avoid overlapping sound triggers. We play a sound in this case an out of breath sound, finally we retrigger the do once function after 16 seconds, allowing it to be retriggered if the conditions are met again.



Passive Power System implementation

Gemini's passive power is a unique system developed by myself in unreal engine 4, it is essential for the exploration and story aspect of twin darkness, allowing the user to find troves of hidden information all of which will impact the users immersive experience.

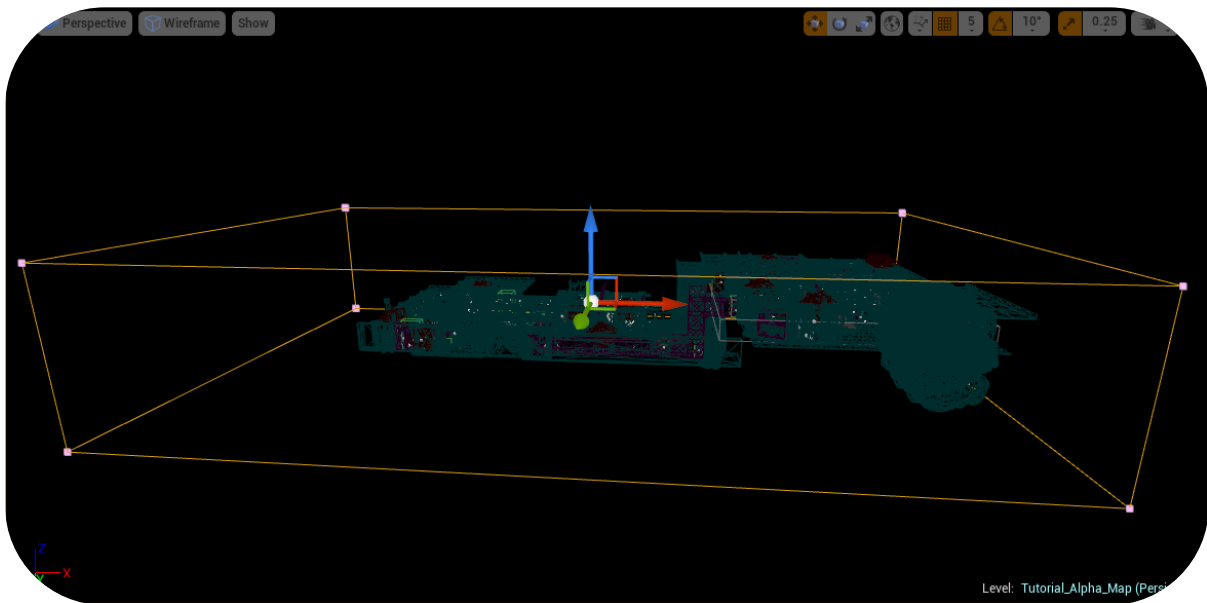
Twin Darkness Technical Report

Controls, Maintains and Implements:

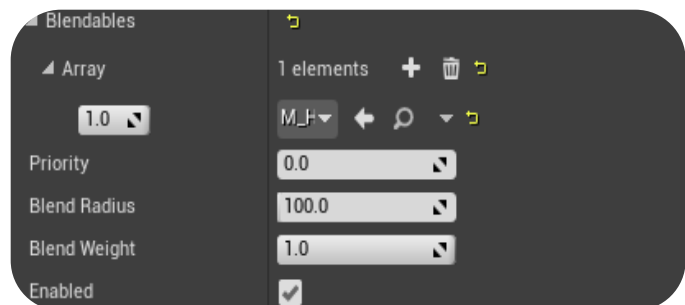
- **Passive Power system:** Allows the user as the move around the map with the main character to locate hidden objectives, points of interest or items with interactive functions. All of these are seen with a uniquely designed white outline that will pulse as the main character approaches them.

Implementation & Screenshots: *The passive power system was achieved in a two-step process*

Step One: Encapsulate the entire level in a PostProcessVolume Box, anything within the box will have a default post process effect.

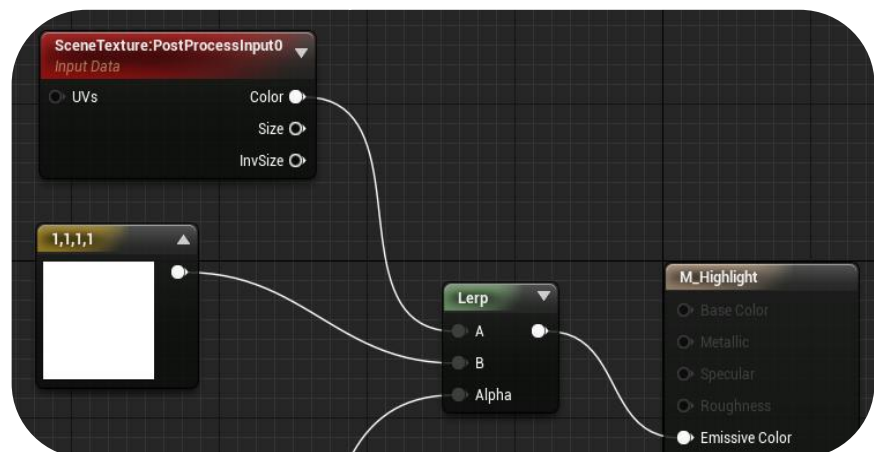


We then assign a unique blendable material under the PostProcessVolume settings, in this case the unique material is M_Highlight. *(This on its own has no effect on the level, as blendables are only called under a unique condition: **Render Custom Depth**)*



For the material we use one of the default's created by unreal and assign two variables:

- SceneTexture: PostProcessInput
- Colour Function (Chosen colour: White)



Twin Darkness Technical Report

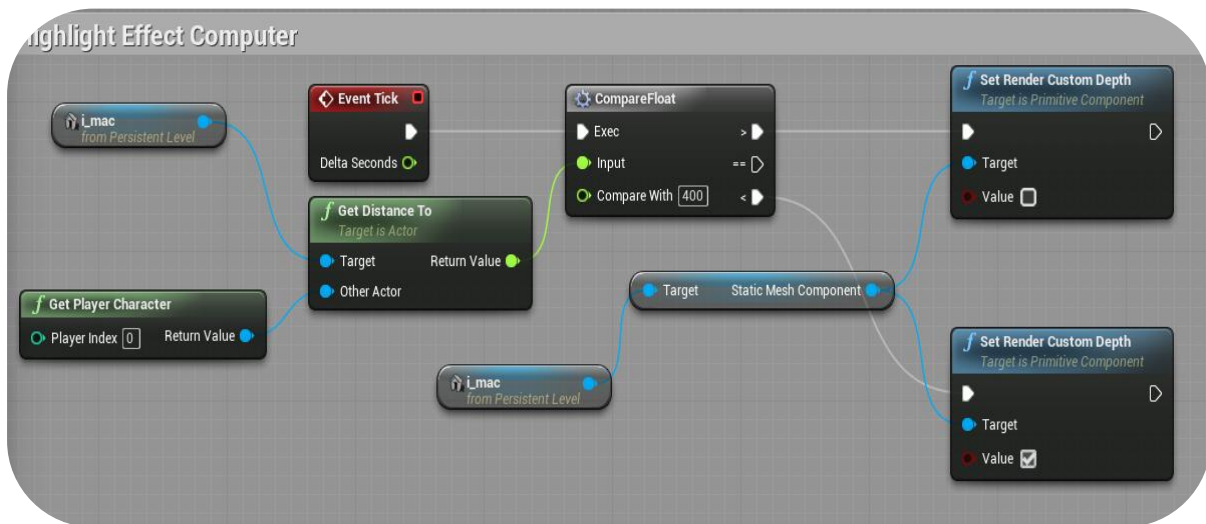
Step Two: Assigning the unique objects and setting up the trigger distance.

Event tick is called (*which is an in-game time event that continually fires at the default time of 1 second*), and attached to a compare float function. Input for the compare float is got from two points:

- The main character
- The unique item

These are then combined within a get distance to function, the input is then compared with a set float value. (*For example the image below is compared against a value of 400, meaning the passive effect will only trigger when the user is 400 units away from the object*)

From the out execute and in execute: we call the blueprint function Set render custom depth, attaching the unique item as the target for both, and setting the value to true for the in execute.



Active Power System implementation

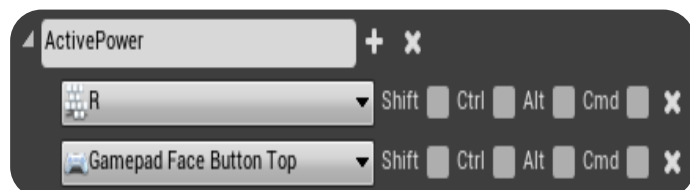
Gemini's active power is essential for the gameplay of twin darkness, allowing the user to see in dark vents and corridors to avoid traps and find the correct route to the end, but it must be used wisely as it has cost on Gemini's health.

Controls, Maintains and Implements:

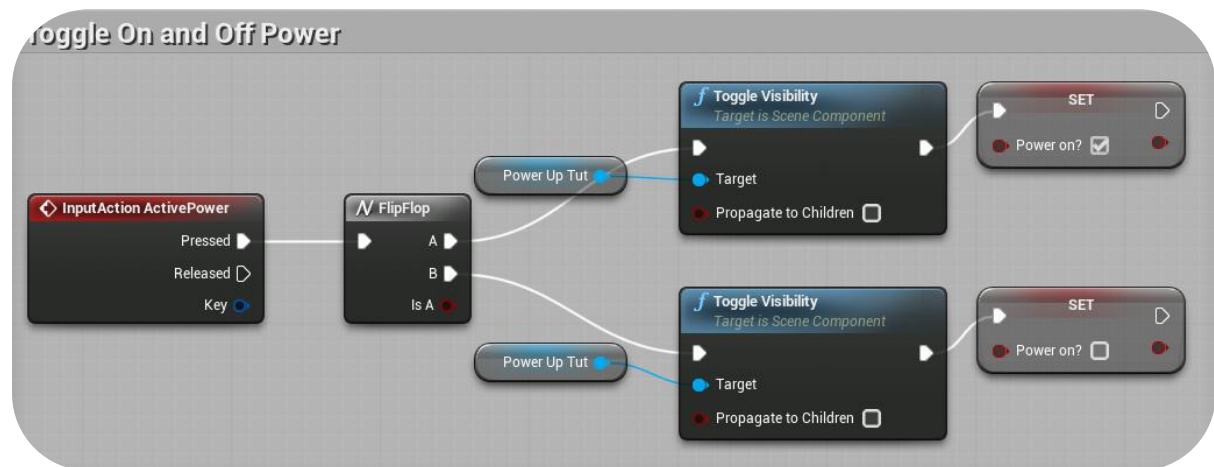
- **Active Power system:** Allows the user to activate an ability that will light the area in a red glow showing everything within a certain distance in front of the main character. As this system is active the health of the main character slowly decays ending in death if the health reaches 0. If the user deactivates the ability, health will regenerate to full.

Implementation & Screenshots:

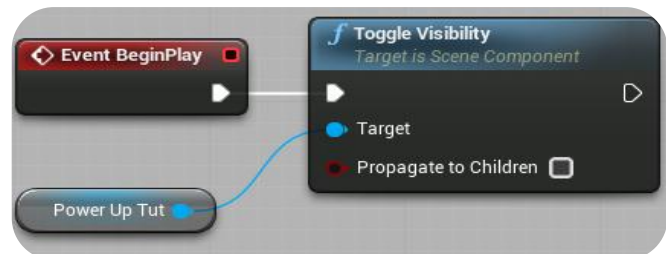
The active power keys are mapped to unreal. The user will press once to trigger the system and press again to deactivate. Mapped for keyboard and controller.



Active Power function is an input action event, which uses a flipflop function to trigger the visibility of the active power (*Power up Tut*), when the active power is on it sets the power on? Boolean to true and when the active power is off it sets the power on? Boolean to false



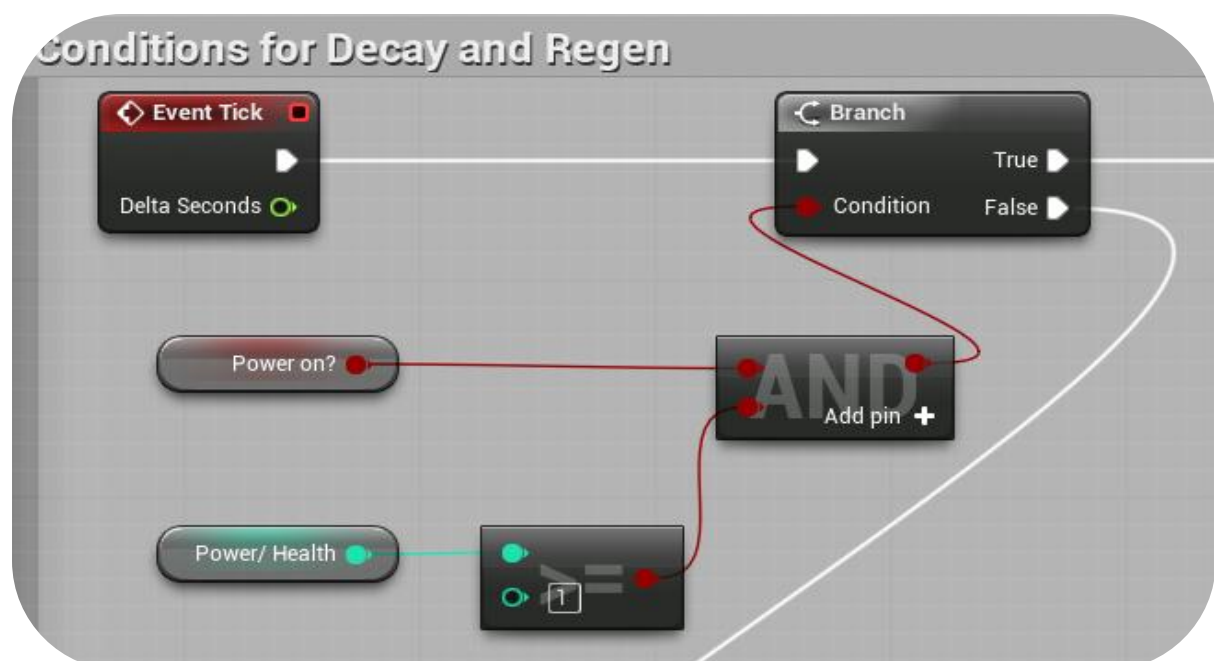
Since the active power system drains health, by default it must be set to off at the level start, this is done with the event begin play function.



Next part of the active power system is the power decay and the power regeneration system. Before the power can decay or regen two conditions must be checked,

The conditions are:

- Power On? Boolean must be set to true
- Power/ Health variable must have a value greater than or equal to one

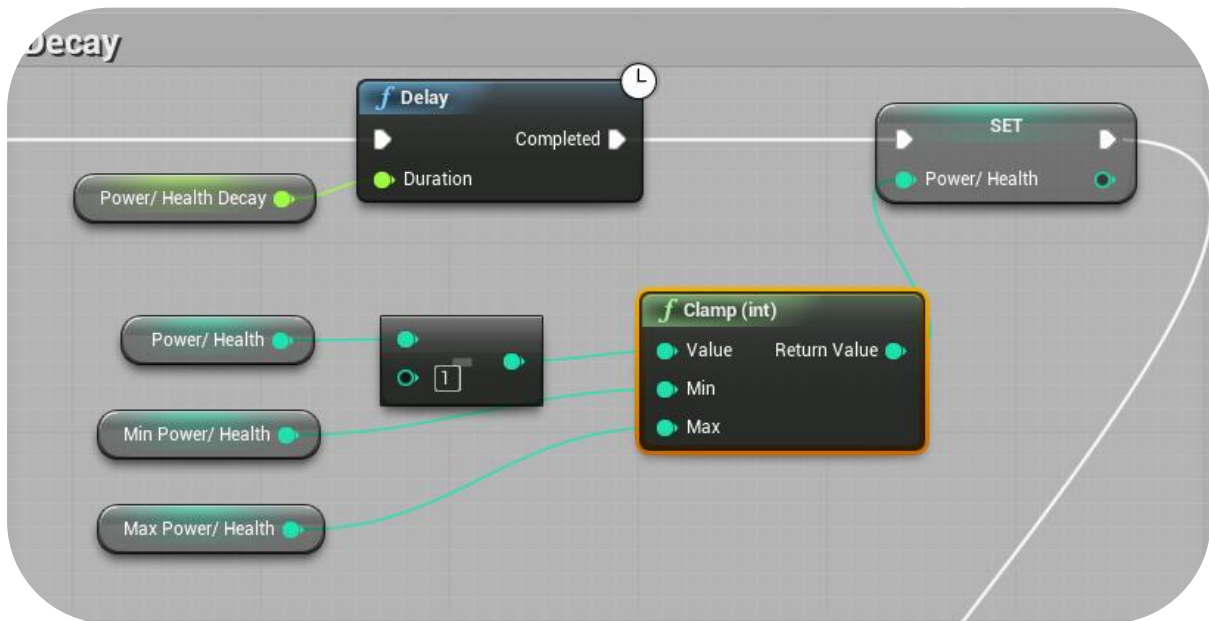


Twin Darkness Technical Report

If True: Power/ Health will begin to decay this is done using a delay function and a float variable with 0.2 set as the decay rate. This is then connected to set Power/Health, where the values for all three variables are set:

- Power/Health set to 100, in essence this is our Health bar (*Widget UMG bar setup explained further down*)
- Min Power/Health set to 0
- Max Power/Health set to 100

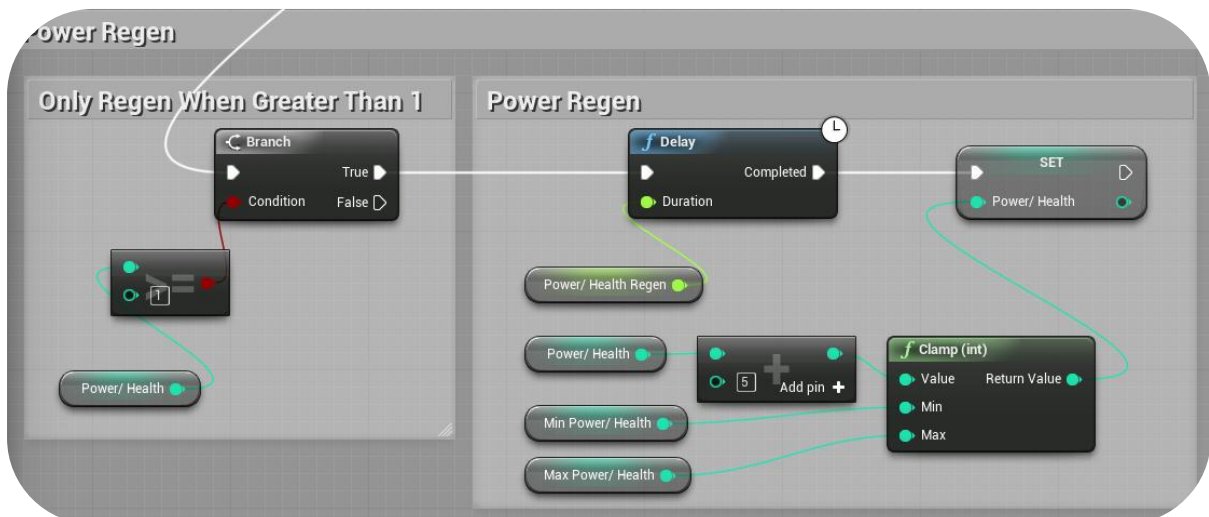
As the delay is triggered the power/ health variable begins to decay by 1



If False: Power/ Health will begin to regen this is done using a delay function and a float variable with 1 set as the regen rate. This is then connected to the set Power/Health, where the three values (*Same values as described above*)

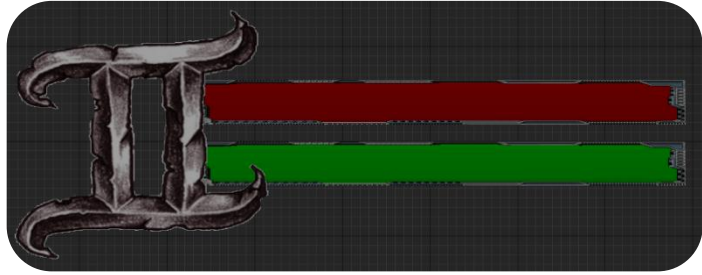
As the delay is triggered the power/health variable begins to regen by 5.

There is also a condition check before, ensuring it will only regen when power/health is more than 1.

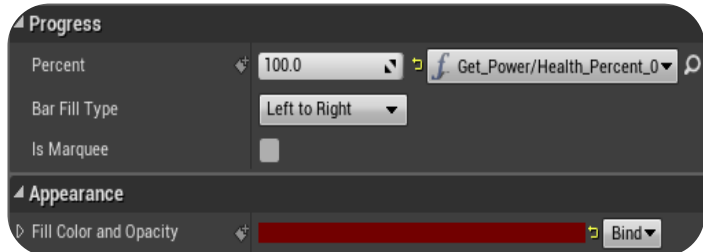


Twin Darkness Technical Report

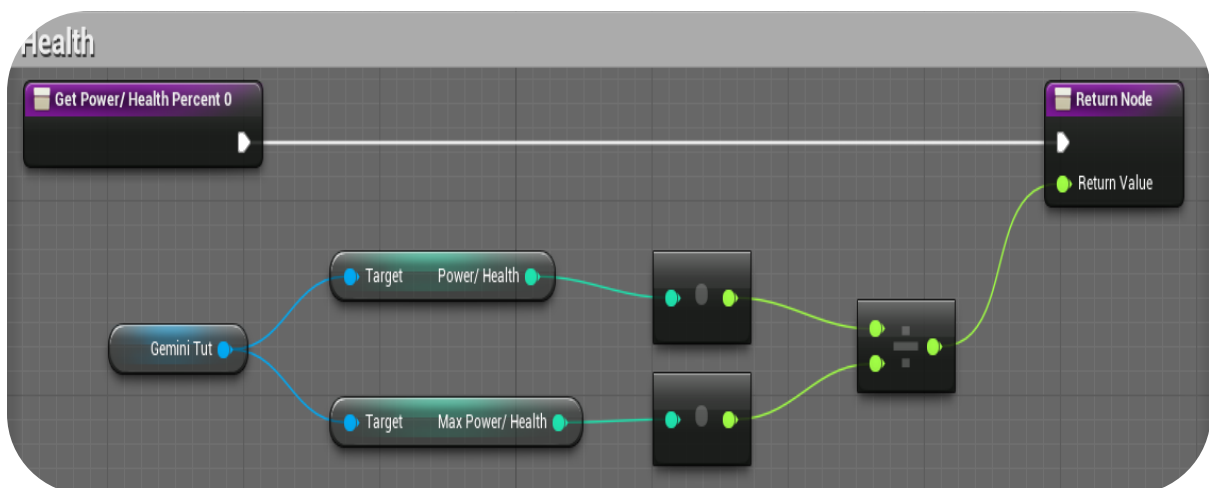
Next we have the display to the users this is setup up using a HUD widget with a progress bar attached, the **red bar** seen in the picture is the Health/Power bar that will decay and regen.



In order for the UMG process bar to drain it must have the same value as the stamina power value in this case 100.



We then attach a function to the process bar itself, we call the character blueprint and export two variables Power/Health and Max Power/Health we then convert them from integer values to floats and divide the two together and connect the result to the return node.



Death System implementation

Death is a way of life and it is no difference in twin darkness, to ensure an immersive experience there are numerous ways in which the user can kill the main character if they are not care full.

Controls, Maintains and Implements:

- **Death System:** User may kill the main character in two scenarios, either scenario will result in the death system triggering (*death sound, death animation and respawn screen being displayed to the user*)

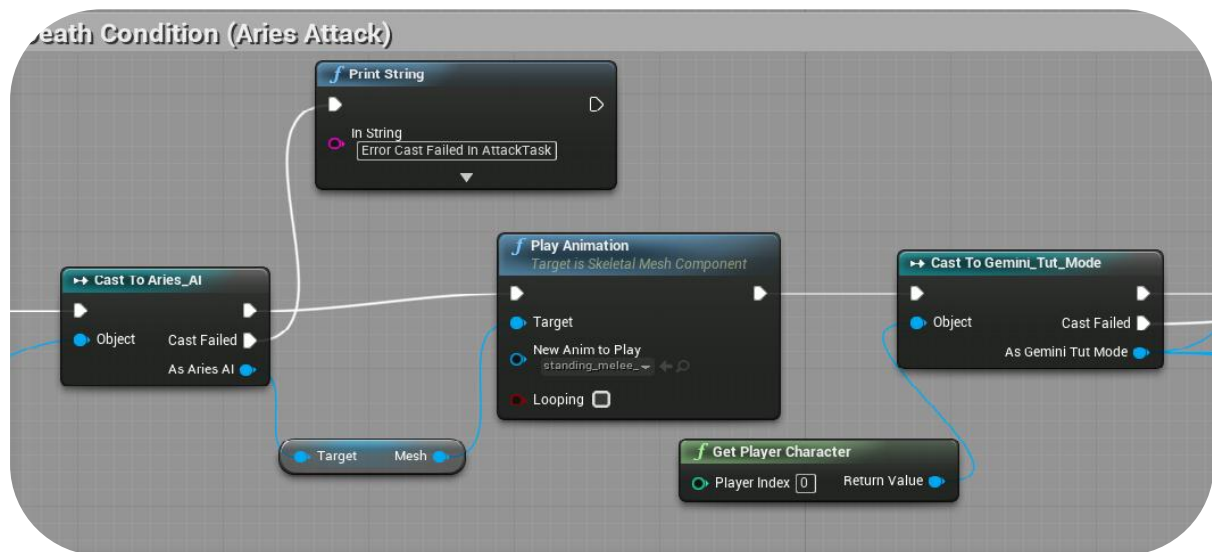
Implementation & Screenshots:

The two scenarios in which the main character can be killed are:

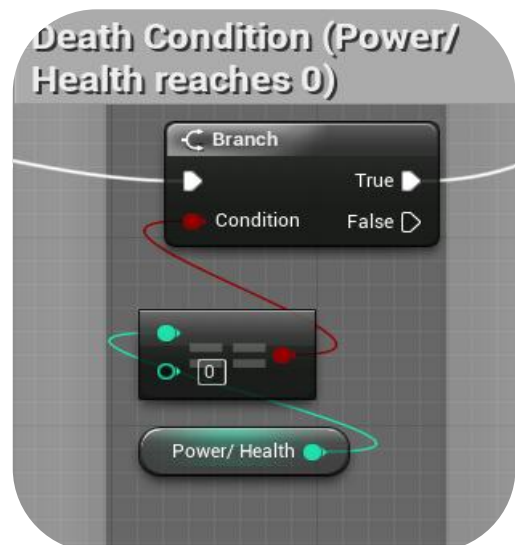
- Be caught and attacked by the Aries agent
- Power/ Health bar decaying to the value of 0

Twin Darkness Technical Report

First Scenario: If the user is chased and caught by the Aries agent, it will trigger an instance kill animation attack using the play animation function. This is then cast to the Gemini model triggering the death system.

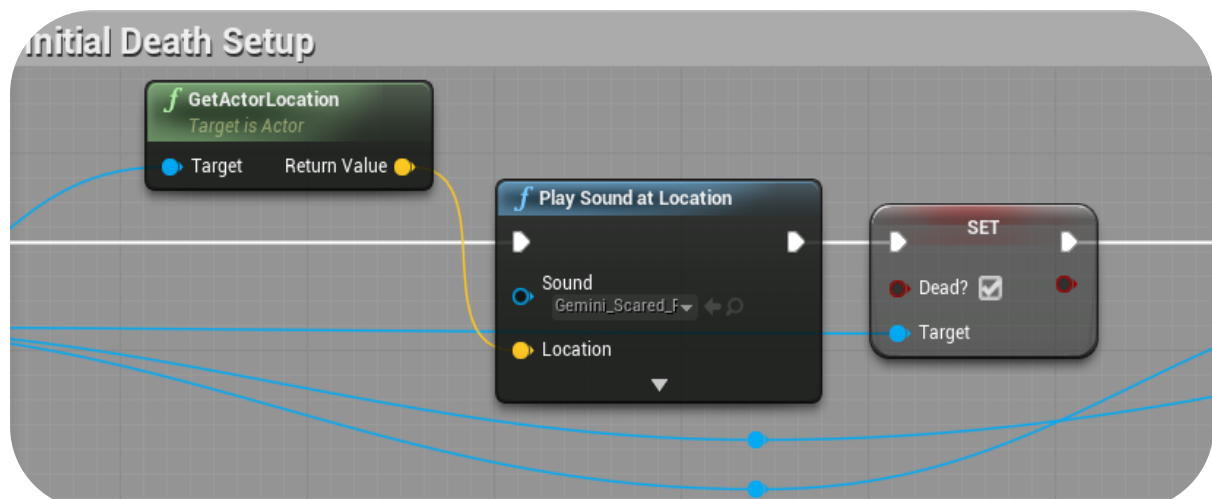


Second Scenario: If the user allows the Power/Health bar to drop to the value of 0, a branch condition set within the character blueprint itself will trigger the death system.



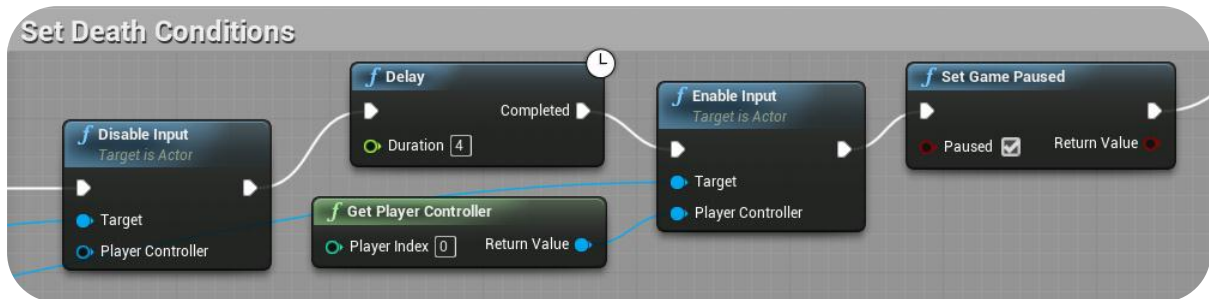
The **Death System** is broken down into three main sections, each section has its own function

First Section: Initial death setup, here the location of the character is got using the get actor location function, then a sound is played (*Dying soundtrack*), finally we said the character animation variable Dead? To true (*triggering the death animation*)



Twin Darkness Technical Report

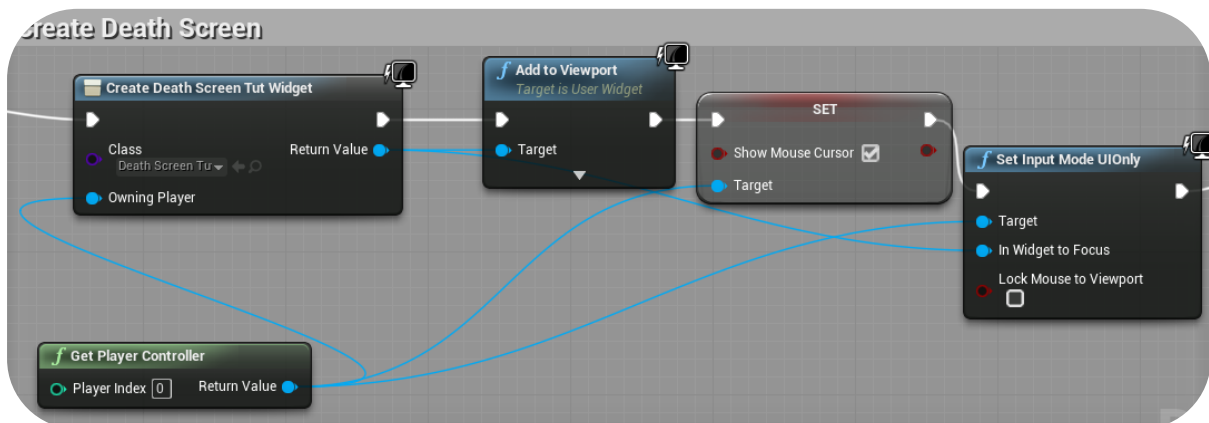
Second Section: Death conditions are set, first the user is disabled for 4 seconds to ensure they can't move while the death animation is being played, the user is then re-enabled while the game is set to a pause mode.



Third Section: The user is then exposed to a death screen widget and their mouse is re-enabled allowing them to choose one of three options:

- Resume Game
- Return to Main Menu
- Exit Game

(The user must choose one of the three options for the game to continue, options follow the same setup as those broken down above in other sections)



User Features Implementation

This section of the implementation will cover the user features developed, integrated and installed for twin darkness. These features are all user optional, the aim of these features is for a better social and gameplay experience. Some features may be limited due to hardware issues or third part licenses

Virtual Reality implementation

Virtual reality is the way of the gaming future and to that point twin darkness has been developed to run on VR devices. VR provides for one of the most immersive gameplay experiences to date.

Controls, Maintains and Implements:

- **Virtual Reality Installed Devices & Library's:** User has a choice of playing the game with two different Virtual reality devices and access to the Oculus Library.

Twin Darkness Technical Report

- **Virtual Reality Level Setup:** Prologue level has two unique functions called at level start-up, these functions aid in providing a clear and clean VR experience.

Implementation & Screenshots:

Installed VR devices:

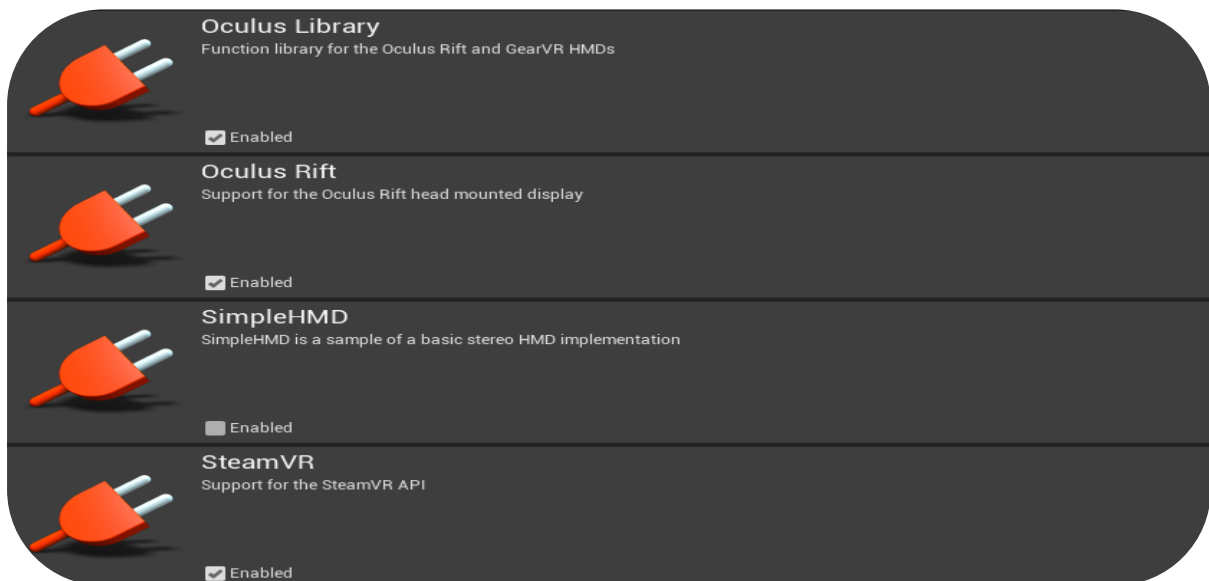
- Oculus Rift, preferred device for twin darkness. Updated to current version for Oculus rift development kit two.
- Steam VR, installed as a backup if oculus integration failed

*Limitations due to hardware accessibility means that **steam VR** was not tested, but has been left installed on the final development version with no problems detected. Removed from the final user version due to no testing ensuring the user only receives a well-tested version.*

Installed VR Library:

- Oculus Library, allows access to the full range of the oculus VR system both at the development level and user level.

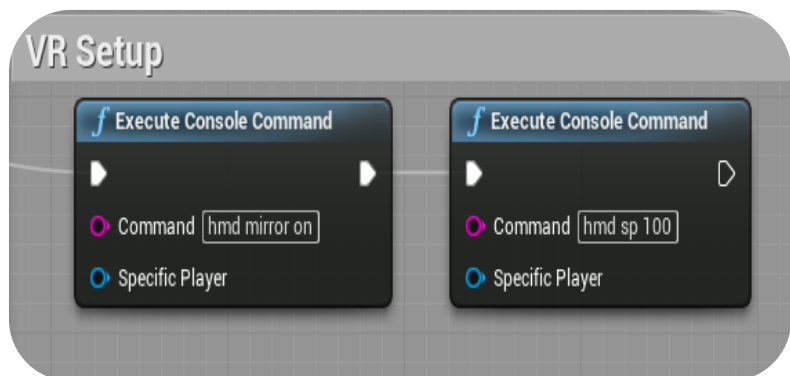
Oculus Framework library allows for plug and play with Oculus rift devices, meaning the user only needs to connect their device and the library will automatic start VR mode.



Level Setup:

Prologue level contains two commands executed on start-up:

- **hmd mirror on:** controls on screen output
- **hmd sp:** Controls quality on the VR device (*Set at 100 always as a good middle point between quality and performance*)



Twin Darkness Technical Report

Steam Network implementation

Steam is the ultimate entertainment platform for PC gaming. Using a developer version of steam, twin darkness allows its users to access their friends, steam achievements and screenshots along with access to the steam community and discussion forums. It also ensures a wider social platform for twin darkness going into its future.

Controls, Maintains and Implements:

- **Steam Network:** If the user has the steam agent installed and is logged in (*logged in by default on computer start-up*), once in game a prompt will alert the user they may access there steam account at any time using the keyboard shortcut (*Alt-Shift*). From there the user has access to the full range of the steam client.

Implementation & Screenshots:

Integrating the steam client with unreal engine 4 was done at the core level, it was **achieved by:**

- Modifying the default engine .INI file, with the below configuration setup using Notepad ++

[/Script/Engine.GameEngine]

```
+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemSteam.SteamNetDriver",DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")
```

[OnlineSubsystem]

```
DefaultPlatformService=Steam
```

[OnlineSubsystemSteam]

```
bEnabled=true  
SteamDevAppId=480  
Achievement_0_Id="ACH_WIN_ONE_GAME"  
Achievement_1_Id="ACH_WIN_100_GAMES"  
Achievement_2_Id="ACH_TRAVEL_FAR_ACCUM"  
Achievement_3_Id="ACH_TRAVEL_SIMPLE"
```

[/Script/OnlineSubsystemSteam.SteamNetDriver]

```
NetConnectionClassName=OnlineSubsystemSteam.SteamNetConnection
```

[URL]

```
GameName=Twin_Darkness
```

[/Script/Engine.UserInterfaceSettings]

```
RenderFocusRule=NavigationOnly  
DefaultCursor=None  
TextEditBeamCursor=None  
CrosshairsCursor=None  
GrabHandCursor=None  
GrabHandClosedCursor=None  
SlashedCircleCursor=None  
ApplicationScale=1.000000  
UIScaleRule=ShortestSide  
CustomScalingRuleClass=None  
UIScaleCurve=(EditorCurveData=(PreInfinityExtrap=RCCE_Constant,PostInfinityExtrap=RCCE_Constant,Keys=((Time=480.000000,Value=0.444000),(Time=720.000000,Value=0.666000),(Time=1080.000000,Value=1.000000),(Time=8640.000000,Value=8.000000))),ExternalCurve=None)
```

Twin Darkness Technical Report

Due to third part nature of steam and its greenlight programme, only the develop access key could be used:

```
[OnlineSubsystemSteam]
```

```
bEnabled=true
```

```
SteamDevAppId=480
```

Even so users still have full access to the steam community, only some developer level functions have been locked.

Steam Achievements implementation

With the Steam network implemented, came the ability to set up a number of limited achievements that the user can unlock as the play the prologue of twin darkness.

Controls, Maintains and Implements:

- **Steam Achievements:** User can complete four achievements with the prologue level of twin darkness. Each achievement is linked to a task that the user will perform when progressing through the prologue level

Due to limitations of developer access four achievements is the max that can be set, achievement names cannot be renamed either.

Implementation & Screenshots:

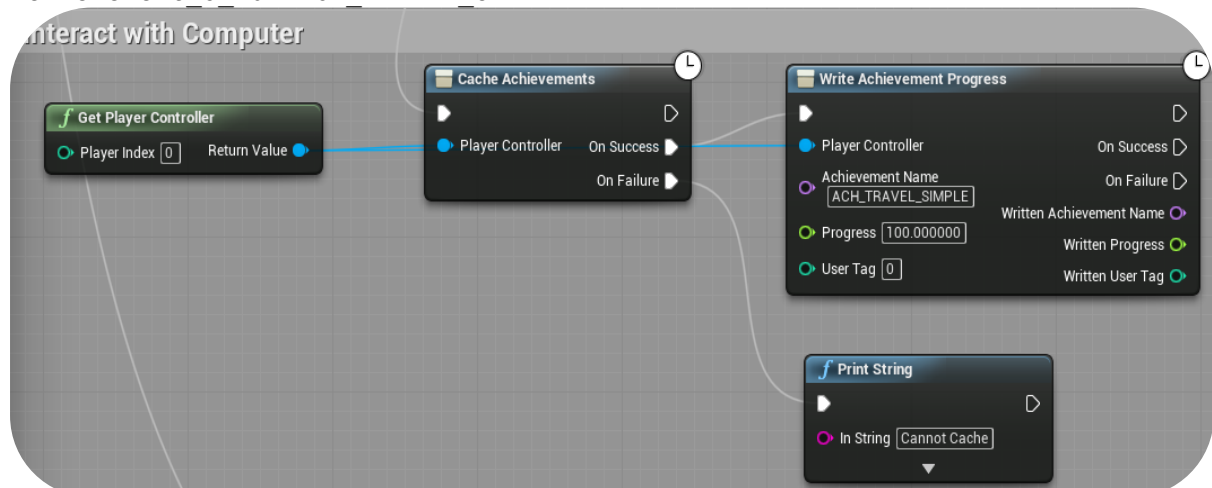
Achievements are linked to other functional blueprints, in the case of the below example, it's linked with interacting with an in-game computer.

To gain the achievement:

- The user must walk over to the computer and hit the interaction key, thus displaying the information to the user. Once the user has finished and walked away the achievement is gained.

This is done by caching the achievement and comparing the name with those declared with the default engine .INI file. If the names declared match the develop account name steam grants the achievement to the user

```
Achievement_3_Id="ACH_TRAVEL_SIMPLE"
```



Testing

Overview

Testing is crucial when it comes to digital games, and it is no different for twin darkness. Three main methods of testing were used throughout the development process. Ensuring that all features and systems presented to a user are functional and meeting the desired requirements.

Tools used in the testing process of twin darkness:

- **Unreal engine 4 Simulation mode:** This mode will simulate in the editor, allowing me the developer to modify level features and systems as the level itself runs (as if playing the game)
- **Unreal engine 4 Test Levels:** Built to solely test minor functions or systems before they are implemented into the main game.
- **Unreal engine 4 Test Projects:** Built to solely test major functions or systems separate from any influence from the overall game project.
- **Unreal engine 4 Twin Darkness Beta:** Latest build version of the full game
- **Online Surveys:** These surveys will auto-generate graphs based on user responses, allowing me the developer to receive feedback to locate bugs or errors and get user impressions on twin darkness.

Methods of testing used in twin darkness:

- **Usability Testing**
- **Unit & Integration Testing**
- **Customer Testing**

Each method of testing is explained under its own heading, tests where participants were used are detailed with the number of active participants and any collected responses from surveys undertaken.

Usability Testing

A Usability study was conducted on twin darkness to help identify problems within the Menu interfaces and in-game control setups. The study also helped stream-line the user experience.

Number of participants: **Five** (age range 18 to 23)

Tools used in Usability study:

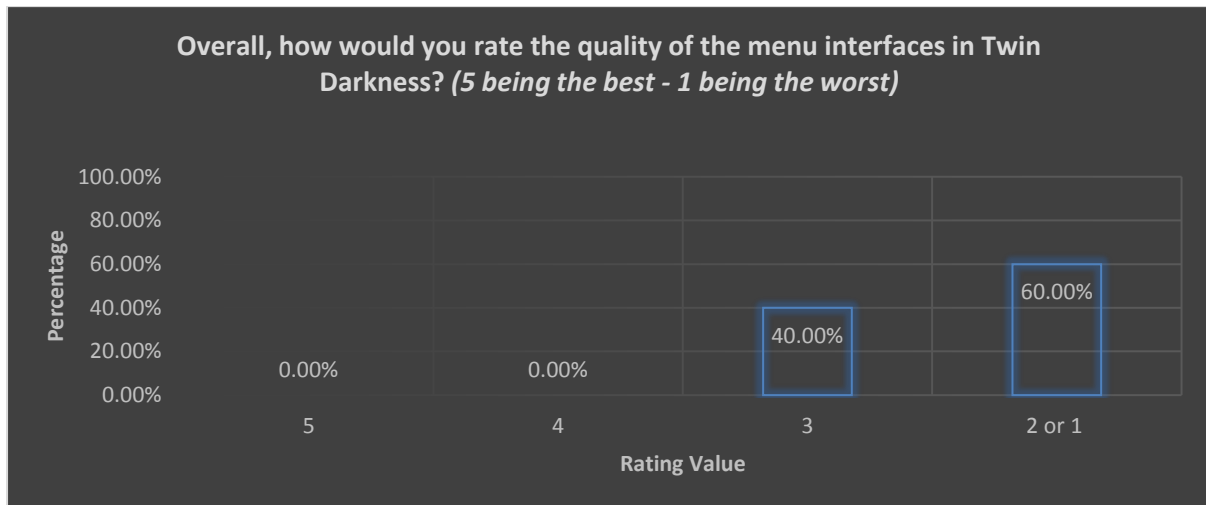
- **Online Survey:** Created a survey based questionnaire, with questions related to improvements, problems and recommendations on the interfaces and the in-game controls
- **Unreal engine 4 Twin Darkness Beta:** Beta version date was 01/04/2016

Usability Study:

All questions are rating based, Ratings below or at 2 are deemed problem areas, users that answer with this rating are asked and required to provide feedback as to what the problem or concern is

Twin Darkness Technical Report

First Question: Overall, how would you rate the quality of the menu interfaces in Twin Darkness? (5 being the best - 1 being the worst)



Collected User Responses from those that answered with a rating of 2 or lower: (Copied directly from survey response)

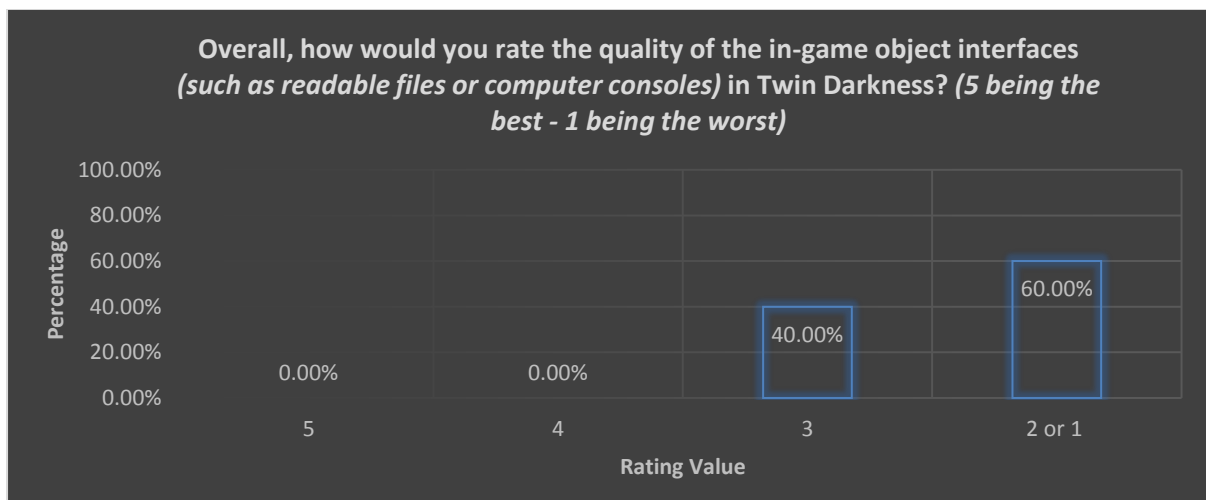
- "Can't tell what button I am pushing or if I have pushed it"
- "Buttons are too small"
- "Hard to tell what button you are pressing on the options page"

Analysis & Solution

From the collected user responses and the ratings, it was clear that the problem with the menu interfaces was the lack of button interaction (There were no hover animations, press animations or sound triggers).

Solution was to create a number of simple button animations using the UMG widget blueprint. These changes should eliminate any problems related to the button visuals on the main menu

Second Question: Overall, how would you rate the quality of the in-game object interfaces (such as readable files or computer consoles) in Twin Darkness? (5 being the best - 1 being the worst)



Twin Darkness Technical Report

Collected User Responses from those that answered with a rating of 2 or lower: *(Copied directly from survey response)*

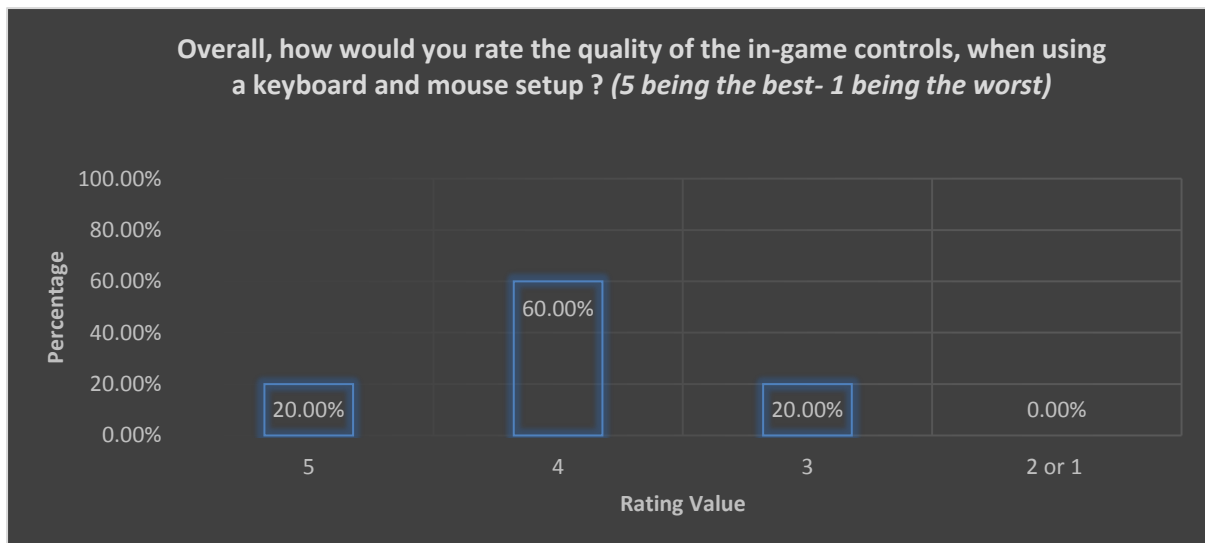
- *"Really hard to see the button on the computer console"*
- *"Cant see writing because of background"*
- *"Background makes it hard to see mouse or buttons"*

Analysis & Solution

From the collected user responses and the ratings, it was clear that the problem with the in-game object interfaces was similar to the first question the lack of button interaction, but there was also a problem deemed with reading text due to the background animation.

Solution for the buttons was the same as the first question create animations for the buttons, the background was fixed by altering the background speed at which it plays. The background speed was cut by 60% in the material blueprint, this made the text placed over it much easier to read and understand

Third Question: *Overall, how would you rate the quality of the in-game controls, when using a keyboard and mouse setup? (5 being the best- 1 being the worst)*



Collected User Responses from those that answered with a rating of 2 or lower:

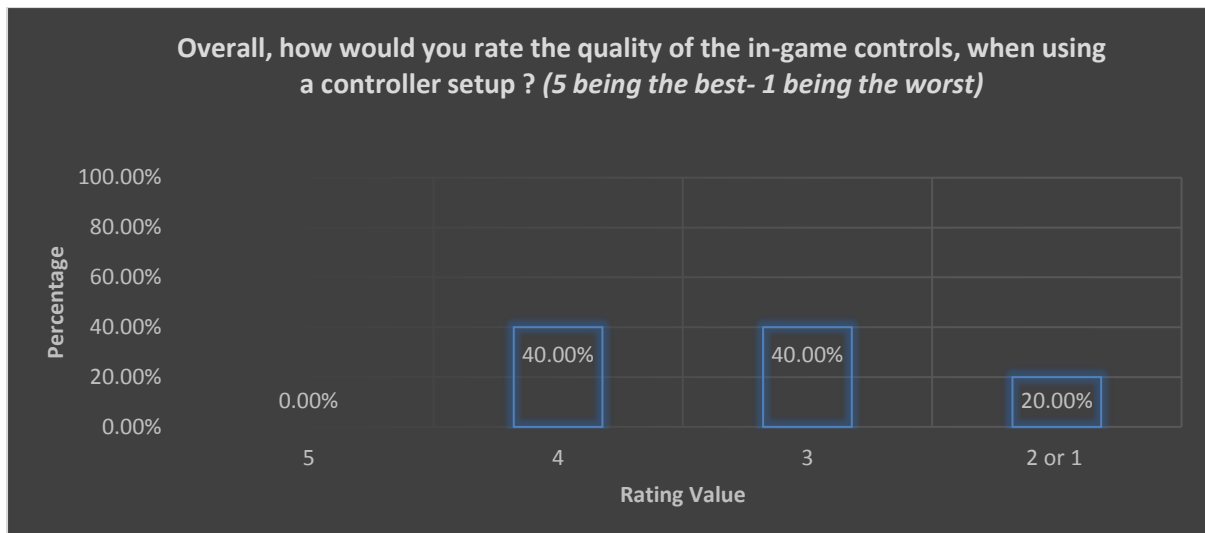
None (No participant answered 2 or lower)

Analysis & Solution

No solution needed, in-game controls for keyboard and mouse deemed acceptable based on participant ratings.

Fourth Question: *Overall, how would you rate the quality of the in-game controls, when using a controller setup? (5 being the best- 1 being the worst)*

All Participants used the same controller setup: Xbox 360 and Wireless sensor



Collected User Reponses from those that answered with a rating of 2 or lower: *(Copied directly from survey response)*

- "Can't use controller on menus"

Analysis & Solution

Known problem that the controller does not work on the menus, unfortunately due to the nature of the problem this cannot be fixed before the project deadline. All participants were informed of this problem, personal feedback after the study, found that it was not a major concern for the participants.

Unit & Integration Testing

Unit testing was carried out throughout the development process of twin darkness, ensuring certain function requirements were developed and built in isolation, as to not effect or be effected by the main project.

Tools used in the Unit Testing process:

- Unreal engine 4 Test Projects
- Unreal engine 4 Test Levels
- Unreal engine 4 Simulation Mode

Unit tests carried out: *(Each Unit test conduct is detailed below)*

Unit Test: Import Protection and Control *(Importing both skeletal and static meshes)*

Tools Used:

- Unreal engine 4 Test Project

Reason for Unit Test: Unfortunately I discovered early on that, importing certain meshes into unreal can be damaging to the project itself and can cause certain crashes if the imported mesh is too large or the in correct format size for unreal

Twin Darkness Technical Report

Process:

- Created an identical starter project to that of the main twin darkness project. Following the same initial creation steps.
- Import select mesh using the drag and drop setup and default import settings
- Inspect the mesh inside unreal engine 4
- If it's a good import, delete within the test project
- If it's a bad import, delete and re-try using different setting options etc. If still no success delete from test project and find alternative mesh and re-test

Unit Test: Save and Load System *(Being able to save the current location of a character and load the save location if one exists)*

Tools Used:

- Unreal engine 4 Test Project
- Unreal engine 4 Simulation Mode

Reason for Unit Test: From conducted research the save and load system would require the alteration of game instances and creation of new functional libraries, these systems may effect or be effected by then current setup of the main project.

Process:

- Created an identical starter project to that of the main twin darkness project. Following the same initial creation steps.
- Developed a basic GUI menu with 4 buttons. Start, Save, Load and Exit and One Default level
- Added basic logic for the Start and Exit Buttons, Tested the functionality
- Began using Unreal engine 4 documentation and researched Save and Load functionality.
- Added developed logic to Save button ,Tested ensuring an external file was created
- Added developed logic to Load button, Tested ensuring the saved location was called.
- Began researching saving and loading between levels
- Added a new game instance class and external library function to achieve loading a saved file between levels.
- Documented each blueprint with comments.

Unit Test: Character Abilities *(The three different abilities the main character has active, passive and sprint)*

Tools Used:

- Unreal engine 4 Test Level

Reason for Unit Test: Didn't want to damage the main character and the animation work already developed within the main game.

Process:

- Created a new default level in within unreal engine 4, with just a basic floor.

- Create a copy of the main character and renamed it to Test Main Character
- All active development of the three ability systems would take place with the Test Main Character blueprint
- All changes to the level blueprint were done to the newly created test level

There were a number of smaller unit tests over the course of the development cycle that used blank test levels to test smaller functionality, changes or improvements.

Integration testing was carried out once a full system or feature had met the required functionality. This type of testing was conducted to track and detect problems with certain features and systems when they are being combined together or being combined within the main project.

Tools used in the Integration Testing process:

- Unreal engine 4 Beta Version

Before any integration test took place a backup copy of the main project was created, in the case of unfixable errors or force system crashes.

Integration tests detailed below are in response to the three main unit tests detailed above. But smaller integration tests were conducted over the course of the development life cycle.

Integration Test: Import Protection and Control

Process:

- Conducted almost straight after a mesh passes a successful unit test import
- Selected mesh is imported into a folder with the main game and inspected

Problems: *(No problems detected with the safe importing of meshes)*

Solution: *(No solution needed)*

Integration Test: Save and Load System

Process:

- Began copying and reproducing the save and load base blueprints over to the main project
- Added buttons to the main menu for the save and load functions
- Copied and reproduced the button functionality
- Ran in-game tests using the simulation mode to ensure no crashes on start-up
- Ran in-game tests ensuring the creation of the save file and the loading functions

Problems: A number of problems occurred due to the change of the game instance from the default one the main project was using to the new one created for the save and load system.

Solution: Delete the default game instance, but ensure that all information with it was transferred to the new game instance created. By default unreal engine 4 will switch to the newly created instance.

Integration Test: Character Abilities

Twin Darkness Technical Report

Process:

- Once a character ability had been developed and met the requirements they would be copied from the test character blueprint to the main character blueprint.
- The ability would then be tested first on the test level created, if no system errors or problems are detected it would then be tested in the prologue level.

Problems: *(No problems detected with the safe importing of meshes)*

Solution: *(No solution needed)*

Backups of the main project were created and stored separately if conducting a major integration or unit test of systems or features. A weekly backup of the main project was also created in case of critical errors or problems.

Customer Testing

A Customer user study was carried out on Twin Darkness to gather feedback, impressions and criticisms on a final version of the game. This study was carried out near the end of the development cycle of the prologue chapter to gather the most accurate results.

The customer user study, will be used to guide the development of future systems and story's in Twin Darkness.

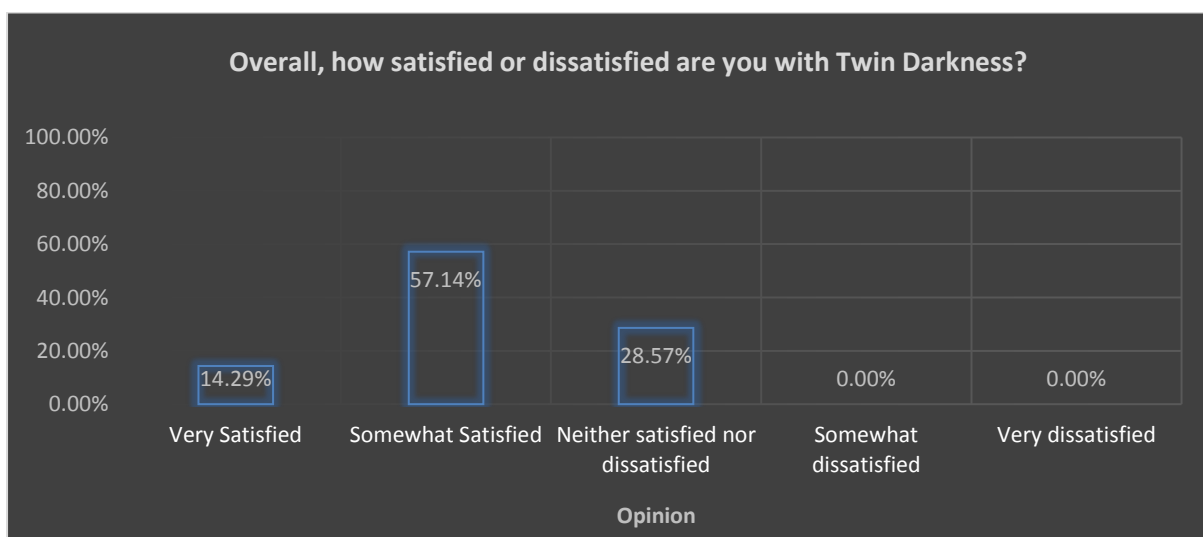
Number of participants: **Seven** *(age range 16 to 23)*

Tools used in Customer User study:

- **Online Survey:** Created a survey based questionnaire, with questions related to feedback and impressions.
- **Unreal engine 4 Twin Darkness Beta:** Beta version date was 25/04/2016

Customer User Study:

First Question: *Overall, how satisfied or dissatisfied are you with Twin Darkness?*



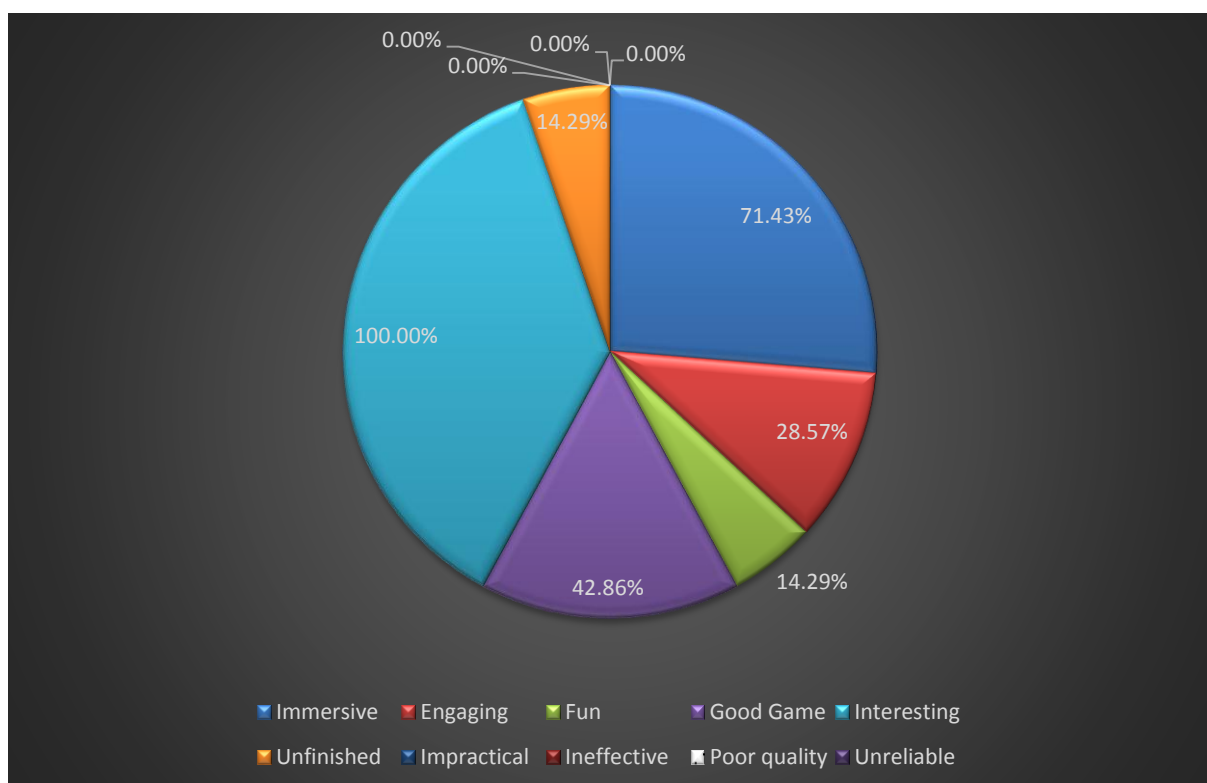
Twin Darkness Technical Report

Analysis & Thoughts

From the feedback on overall satisfaction, it's clear that a majority found Twin Darkness to be somewhat satisfying. In one outlier a participant picked very satisfied which is the best possible answer to be given for the question, there were also two participants that picked the middle ground option of neither satisfied nor dissatisfied.

Overall I believe this to be a clear positive for Twin Darkness, as no participants picked a clearly negative response. And the majority picked the second best answer, indicating that there are features or systems that clearly satisfied them in Twin Darkness.

Second Question: *Which of the following words would you use to describe our products? Select all that apply*



Analysis & Thoughts

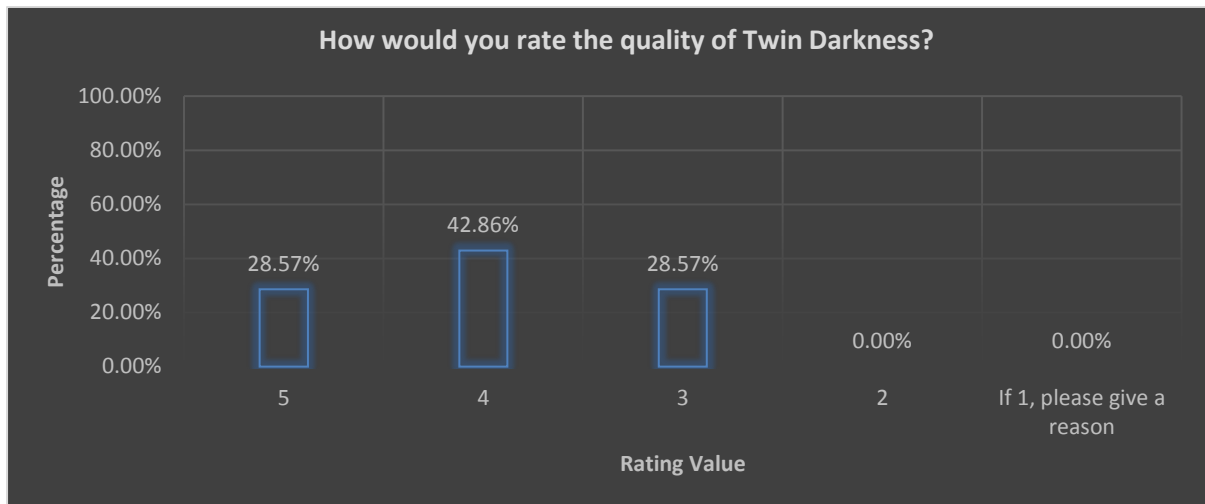
This question was set and asked in such a way to gauge keywords related to twin darkness. All participants could pick as many key words as they wished when answering this question. There were an equal number of positive key words and negative key words. Ensuring a fair distribution.

Overall I am extremely pleased with the key words picked by all participants. The three most picked key words were Interesting with a 100% pick value, Immersive with a 71.43% pick value and Good game with a 42.86% pick value.

One negative key word was picked by just one of the participants which was Unfinished, which was understandable since the version of twin darkness used was still lacking certain minor features. But since this was only picked once, it was clear that it did not affect the other participant's thoughts on twin darkness.

Twin Darkness Technical Report

Third Question: *How would you rate the quality of Twin Darkness?*



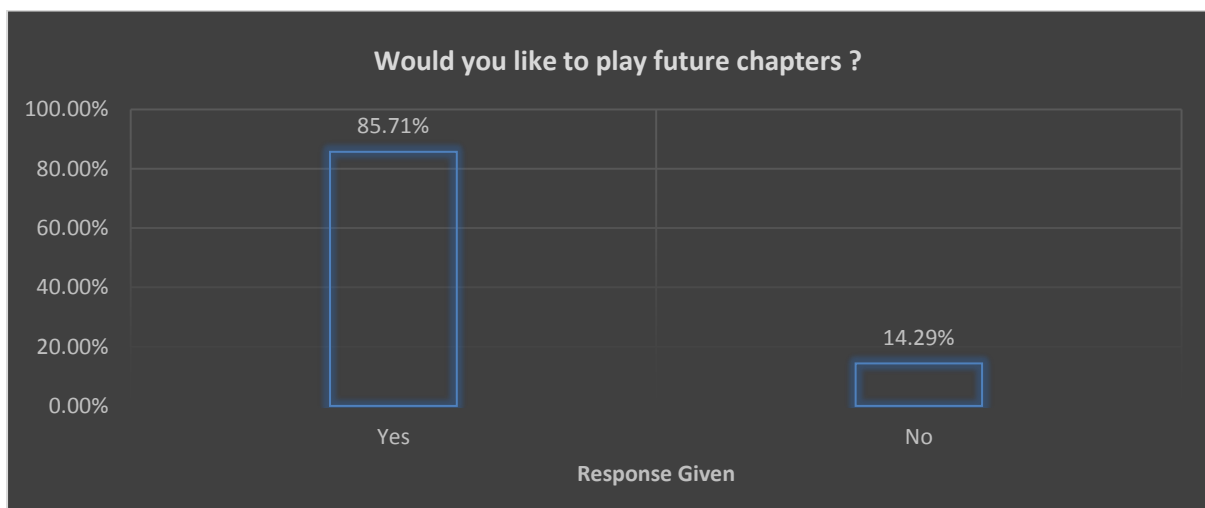
Analysis & Thoughts

This question was asked to gauge a simple rating value for twin darkness, one to five with five being the best possible value and one being the worst possible value. If a participant answered with a value of one, they are asked to give a reason why.

Thankfully not one of the participants picked a rating value lower than three and three was not even the most picked majority. Overall, this is another great result in my opinion. With majority of participants picking a rating value of four and two participants picking the max rating value of five.

The final two questions of the customer user study deal with the future of twin darkness: future chapters and future pricing

Fourth Question: *Would you like to play future chapters?*



Analysis & Thoughts

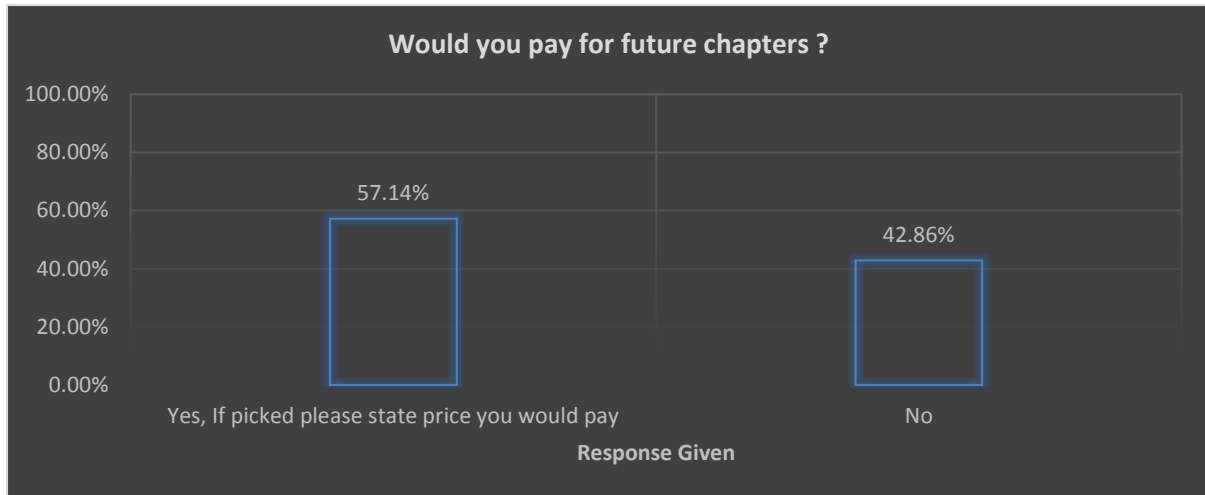
This question was in relation to future chapters of twin darkness, to gauge whether or not to customers would be interested in playing future chapters of twin darkness.

Twin Darkness Technical Report

It also gives myself a clear indication on whether or not future content would be worthwhile developing.

From the participants results it is clear that future content would be worthwhile developing since six out of the seven participants voted, Yes they would like to play future chapters.

Fifth Question: *Would you like to play future chapters?*



Analysis & Thoughts

This last question of the customer user study deals with pricing future content, if it was made. In this question participants that picked yes are asked to state a price they would pay for future chapters.

Taking the last question into consideration it is clear that if future chapters came with a price it would decrease the number of people willing to play.

Overall however four of the seven participants said they would be willing to pay for future content providing estimates of how much they would be willing to pay. A number of the Yes participants entered in multiple value numbers,

- "10 euro"
- "Up to 20 euro"
- "5 - 10 "
- "5.00 to 10.00"

So I have taken two estimate values:

Low estimate value based on Yes response: 10 euro

High estimate value based on Yes response: 12.50 euro

Conclusion of the customer user study

Overall, I am extremely pleased with the feedback and responses given during the study. It helped with gaging the customer views of twin darkness. Along with planning for the future of the game moving forward.

Graphical User Interface (GUI)

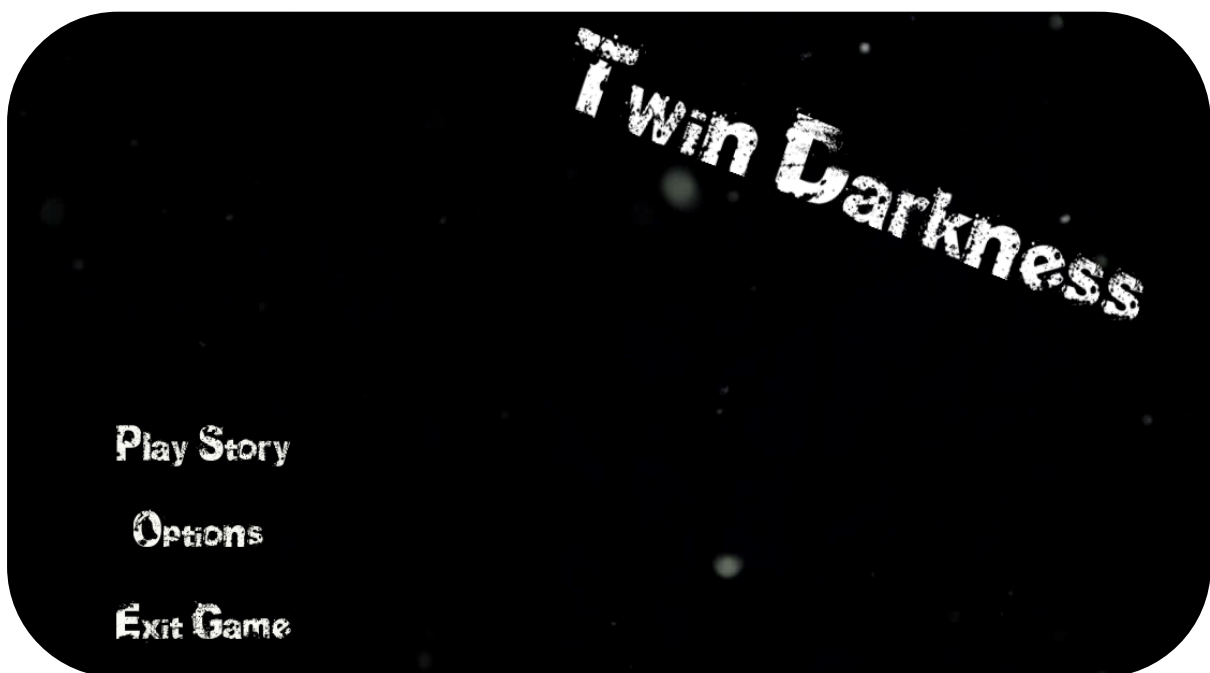
This section of the report will cover all the different GUI menus the user will have access to and need to navigate in order to play twin darkness at its max potential.

By default all menus have animated backgrounds, animations and sound triggers to ensure an optimal user experience.

Main Menu

Description: First interface the user will see and interact with. Has three available button options:

- Play Story, allows the user to go to the chapter select options.
- Options, allows the user to modify all in-game options.
- Exit Game, allows the user to shutdown twin darkness.



Close-up Look: Buttons

- Highlight effect that appears on buttons once the user hovers over the specific option.
- Sound trigger on all button hovers.

Reason: Easier and clearer user experience, for navigating menus



Options Menu

Description: This interface allows the user to modify all in-game settings in order to improve performance on their machine by increasing or decreasing certain options. It also allows users with higher end machines to increase the visual quality of the game.

There are **four pre-set bundle options**:

- Low
- Medium
- High
- Ultra

Nine indiviual options: if the user wishes to modify their performance and graphic ability manually instead of using the four pre-sets.



Chapter Select Menu

Base Chapter Select

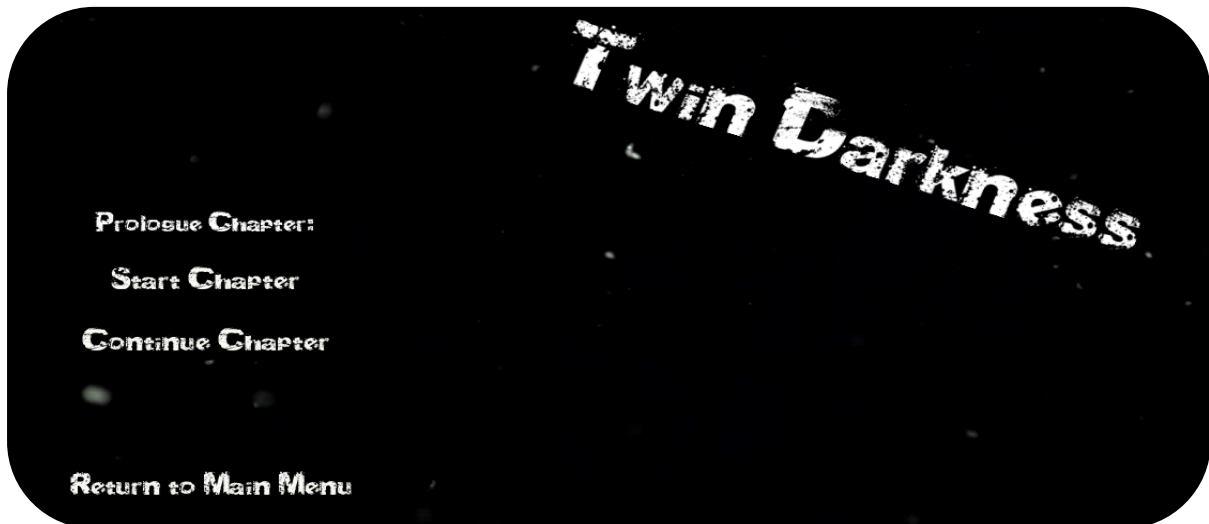
Description: This interface allows user to choose available chapters to play (*to date only prologue level is available*), also allows the user to return to the main menu if they wish to exit the game or modify in-game settings



Twin Darkness Technical Report

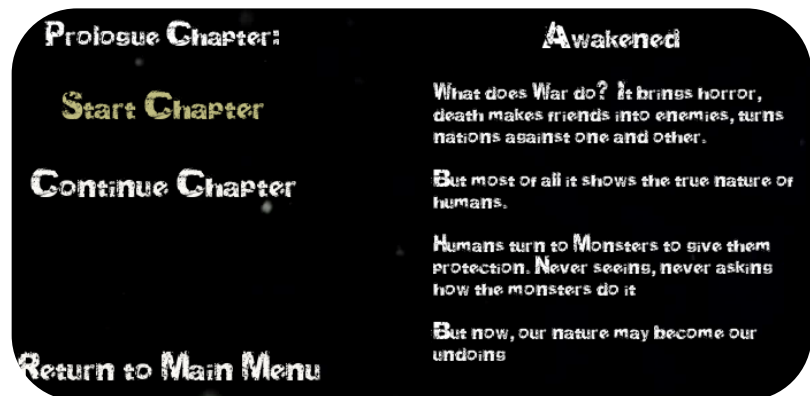
Chosen Chapter Select

Description: This interface allows the user to start the chosen chapter from the beginning if they use the start chapter button, if they have play already and saved they can start from there using the continue chapter button.



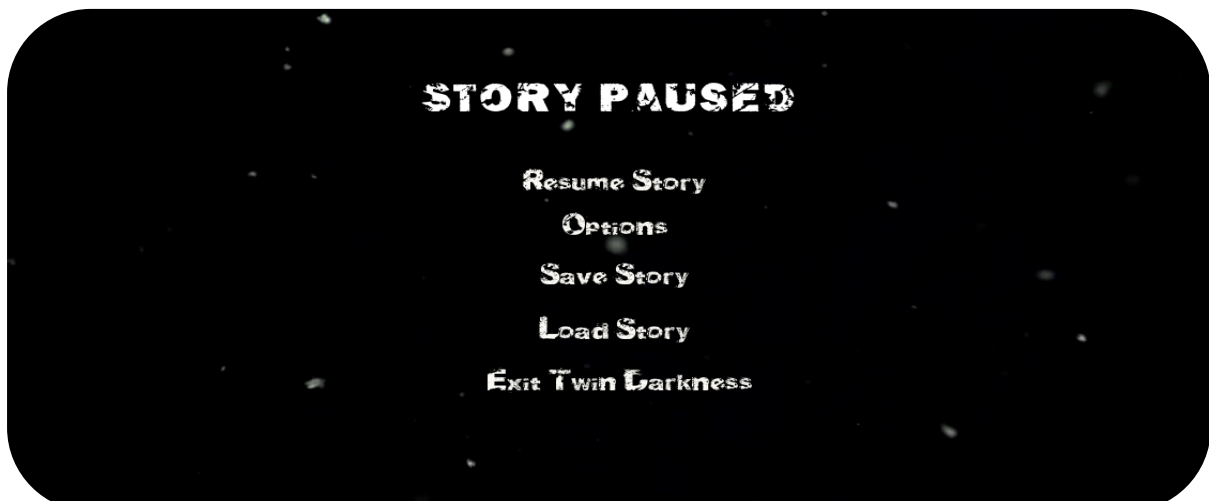
Close-up Look: Buttons

- Highlight effect that appears on buttons once the user hovers over the specific option
- It will display the chapter title along with a brief story



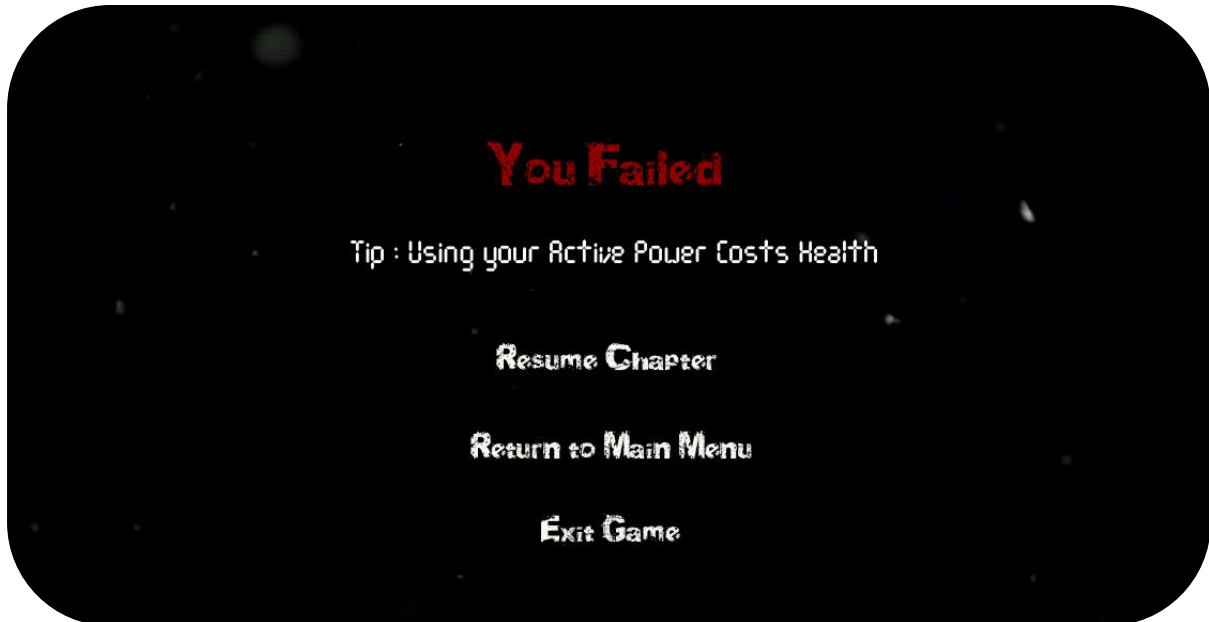
Pause Menu

Description: This interface appears when the user is in game and presses the pause key, this interface **allows the user to:** *resume their game, modify options, save their game, load their game and exit the game.*



Death Screen Menu

Description: This interface appears when the user dies in-game, this interface allows the user to resume their game from their last save or from the start if no save exists. They can also return to the main menu or exit the game



In-Game Objects

Readable Object Screen

Context: Readable Items

Description: Displays story information to the user if they find the item and activate it in-game using the pre-set key. **Types of Items:**

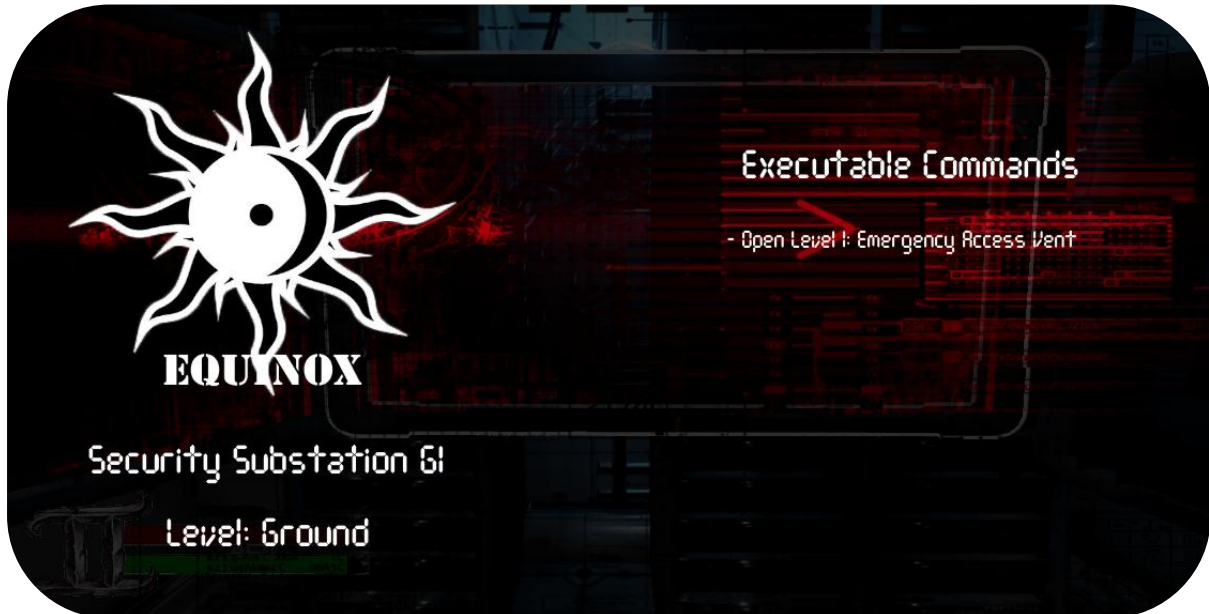
- Tablets
- Computers



Interactive Object Screen

Description: Displays and allows the user to trigger in game events using button commands: this screen example:

- Will open a vent on the other side of the level allowing the user to advance.



Close-up Look: Button Command

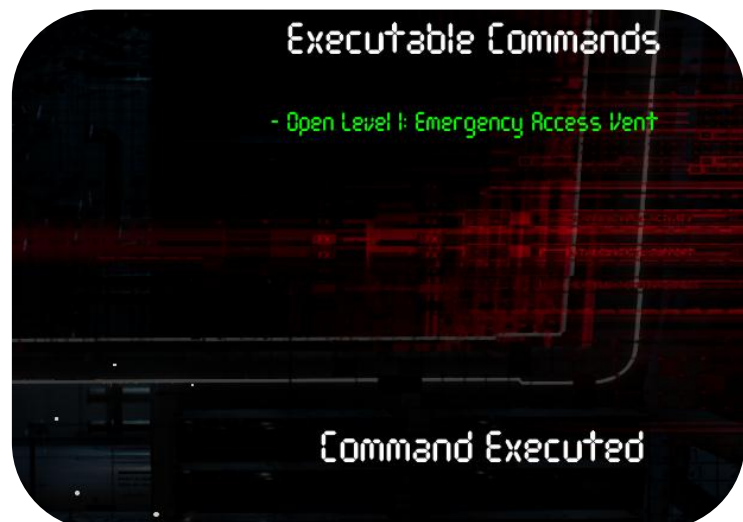
- Highlight feature on the button command when hovered (yellow)



Close-up Look: Command Triggered

- Highlight feature on button press (green)
- Spinning Icon
- Command Executed alert

Last 1.5 seconds after being triggered, the user is then returned to the game.



Twin Darkness Technical Report

Main Character & Environmental Previews

This section of the report will display previews of the main character model and environmental setups.

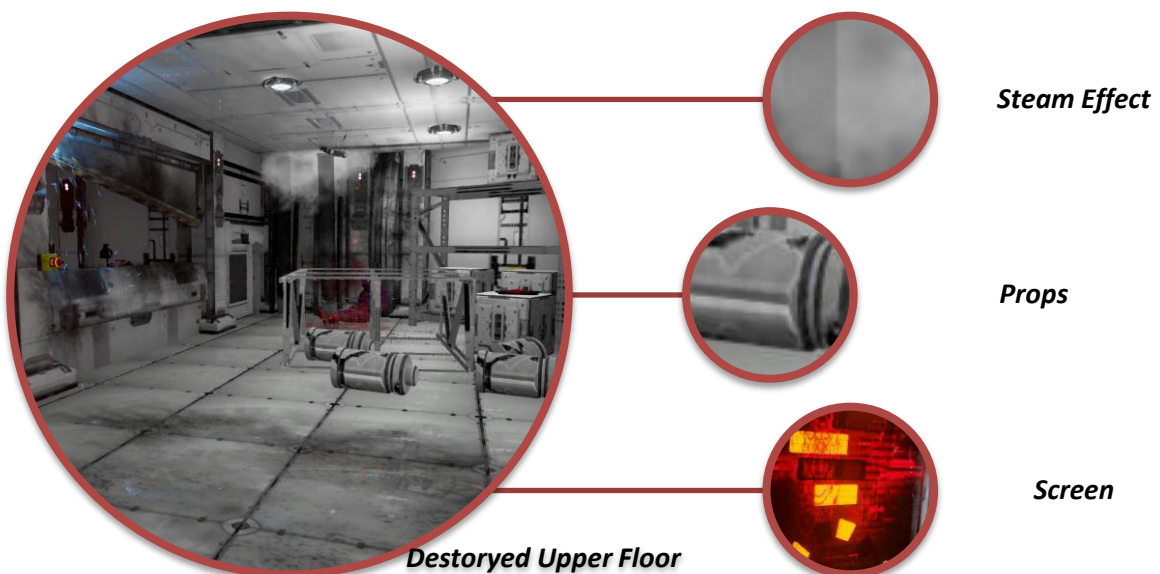
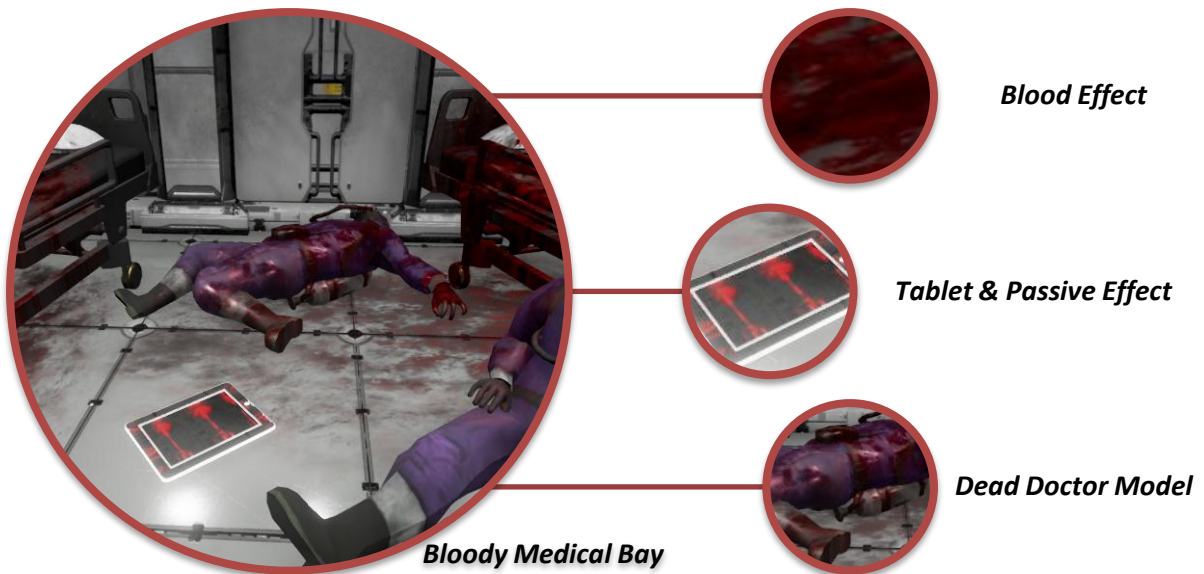
The main character model preview shows the character and how the user see's effects and abilities in game.



The environmental previews show a complete scene taken from in-game, that scene is then broken down highlighting key features and effects.



Twin Darkness Technical Report



Conclusions

Milestones and Hurdles

Milestones were set by myself in two ways:

Self-imposed: These milestones had little to no deadline and were more to do with what I want the end product to be. In terms of visual style, story and overall feel of the game. Examples of these self-imposed milestones are:

- Design the level, with a sci-fi feel in order to better set the tone of the game for users.
- Implement a unique passive and active ability for the main character Gemini, for the purposes of storytelling and user experience.
- Implement a unique A.I agent, with sight and audio sensing abilities.

These milestones may not have deadlines but had major impacts on the game itself and the evolution of twin darkness over the course of the nine month development.

Required: These milestones, unlike those that I self-imposed had deadlines and were more to do with the functional requirements of twin darkness and the project module over all. Examples of these required milestones are:

- Ensuring a daily log was kept on all development work related to twin darkness, this log would be used for the monthly journal submissions that had their date set at the end of each month.
- Ensure a working prototype for the mid-point presentation.
- Complete all eight functional requirements before the final submission deadline

These milestones made up the framework on which the entire twin darkness project was based.

The Main **Hurdles** that were encountered:

Unreal engine 4: Self teaching myself unreal engine 4's blueprint system, unlike any type of coding work done before within the college. It was a challenge understanding the new concept of strings and blueprints. Luckily at times it was a rewarding challenge.

The other challenge of unreal engine 4 was their strict setup on imported meshes, skeletons and materials. Unreal only allows .FBX files and .TGA files meaning a lot of work had to be redone and reimported.

Blender: was originally going to be used for the creation of full skeletal models with full animations that would then be imported into unreal engine 4. Unfortunately after nearly a month of work I found that it is impossible to get the rigged character into unreal engine 4. This was one of the biggest hurdles I faced and it took time to find a solution. But in the end the solution I found which was Mixamo turned out to be better and less time consuming.

Finally the last hurdle was with myself and it involved understanding my limitations and what my realistic goals should be. Thankfully I began to overcome this hurdle early on in the first semester helping to pave real requirements and real deadlines.

Further Development & Research

Further development opportunities for Twin Darkness:

- **New Chapters:** Continue development on new chapters after the completion of the prologue chapter. These new chapters would add new story content, in-game models and characters.
- **Evolving the Main Character:** Develop new ability conditions for the main character, such as increasing the effects already triggered by the active ability. Example: Instead of just increasing the characters sight ability, increase the player's base movement and jump abilities.
- **Improve Twin Darkness:** This can take many forms such as continuing to improve game systems through user testing and feedback.
- **Virtual Reality Improvements:** Continue to increase the VR abilities of twin darkness, by finding and integrating more VR plugins into the system, by continuing to evolve the VR design of twin darkness ensuring a completely immersed experience for the user.
- **Active Release:** Prepare and release a full copy of twin darkness on a number of online platforms, this active release would contain the prologue chapter fully complete ensuring a full user experience.

Research opportunities for Twin Darkness:

- **User Feedback:** Use feedback provided from users to better understand issues or take advice on what should be implemented going forward.
- **Upgrade & Continue Learning:** Upgrade the entire project to a newer version of unreal engine 4, currently twin darkness is developed on version 4.9, the latest version of unreal is 4.11. This new version offers improved systems and new options all which will be self-learned and used to improve and grow twin darkness.

Closing Statement on further development & research for Twin Darkness is that I aim to continue development, this project has become a great hobby, a great interest of mine and increased my knowledge in the areas of gaming, modelling and design.

So much so I have applied, interviewed and accepted a Postgraduate course at D.I.T in digital media, this course offers a specialisation in the second semester called digital games, here students will be taught a number of different engines, software's etc.

I aim to choose this specialisation to grow my knowledge in the area of digital games. It is here I hope to continue growing and evolving twin darkness as a game.

Appendix

References

Docs.unrealengine.com. (2016). Command-Line Arguments. [online] Available at: <https://docs.unrealengine.com/latest/INT/Programming/Basics/CommandLineArguments/index.html> [Accessed 30 Apr. 2016].

Docs.unrealengine.com. (2016). Unreal Engine 4 Documentation. [online] Available at: <https://docs.unrealengine.com/latest/INT/index.html> [Accessed 30 Apr. 2016].

Forums.unrealengine.com. (2016). Animation. [online] Available at: <https://forums.unrealengine.com/forumdisplay.php?66-Animation> [Accessed 30 Apr. 2016].

Forums.unrealengine.com. (2016). Content Creation. [online] Available at: <https://forums.unrealengine.com/forumdisplay.php?5-Content-Creation> [Accessed 30 Apr. 2016].

Forums.unrealengine.com. (2016). Unreal Engine Forums. [online] Available at: <https://forums.unrealengine.com/> [Accessed 30 Apr. 2016].

Oculus. (2015). How to build for Oculus Rift. [online] Available at: <https://forums.oculus.com/community/discussion/25215/how-to-build-for-oculus-rift> [Accessed 1 Mar. 2016].

Statista. (2016). Statista - The Statistics Portal. [online] Available at: <http://www.statista.com/> [Accessed 04 Dec. 2015].

Reflective Monthly Journals

September Journal

Reflective Journal for September

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: September

What I did in September

19/09/2015 (Saturday)

Researched transferring models being built in blender into unreal found that they need to have a skeletal mesh in order to be used as a moveable character. Otherwise they cannot be animated in the game.

Installed Blender on my laptop so I would have it for model and rigging design.

Researched rigging and modelling in blender.

Looked free modelling sites for templates of my characters, decide to go with the following

- <http://tf3dm.com/>

21/09/2015 (Monday)

Came in at 9, to begin the research and testing on my project idea

First order was to test making a first person camera in Unreal Engine 4

After the first person camera was done and tested moved onto to blender testing and rigging for some of the models I am planning to place into my game

Once I was done with the above testing I moved to the first of the documentation that is needed for my project. Which is the project proposal did the 3 following questions

- Background
- Special Resources Required
- Appendices

Later in the project lab ask Eamon a few follow-up questions on the project Proposal so I had an idea of what was needed.

Twin Darkness Technical Report

22/09/2015 (Tuesday)

Attended the Graphics lab at 1 and ask a few questions about what is expected from a game project.

And build a test level where I added light and some of the default Unreal engine 4 props such as tables and chairs, also changed some of the default materials.

24/09/2015 (Thursday)

Found doc's for unreal engine 4 to retarget animations to custom skeletal meshes in unreal engine 4 this will help when using my custom female and monster models

- <https://docs.unrealengine.com/latest/INT/Engine/Animation/RetargetingDifferentSkeletons/index.html>

26/09/2015 (Saturday)

Finished writing up all parts of my project proposal apart from the project plan, Researched multiple developers for both VR and Unreal engine 4 it to my

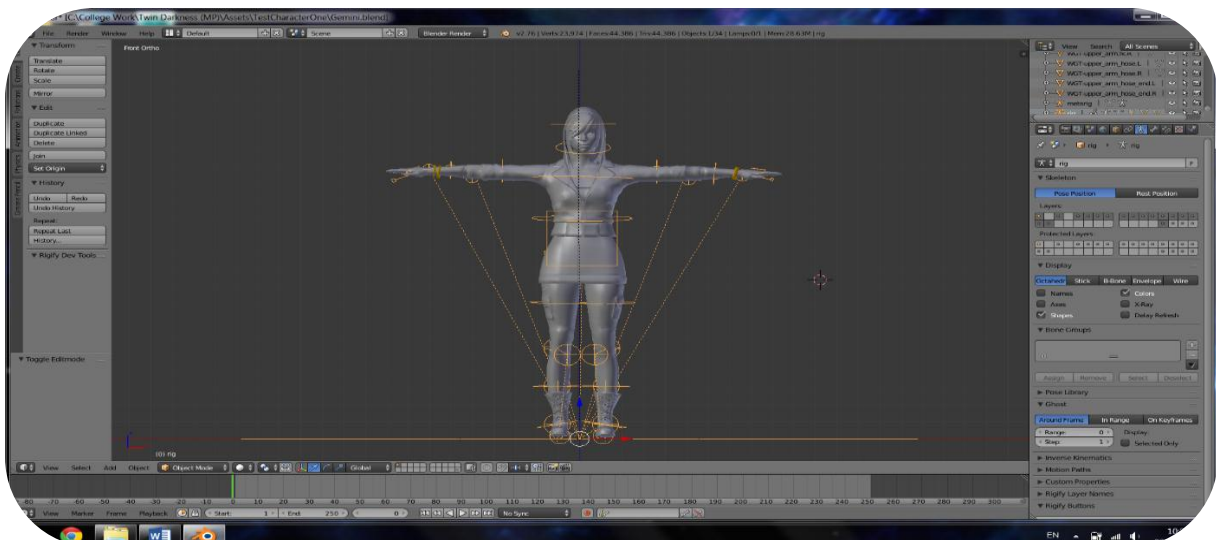
27/09/2015 (Sunday)

Wrote up a full project plan using Microsoft plan including a full gnatt chart and a breakdown of all the known tasks of my project. Listed important upload dates and marked millstones.

28/09/2015 (Monday)

Spend the project class building my main character model this included rigging a full skeletal mesh and then skinning it to my model. I then did some testing in blender to make sure all the joints matched up right and moving my smooth and not clunky.

I had to make a few adjustments to the hands and arms as the mesh was a bit off track. Image of model (Main Character)



29/09/2015 (Tuesday)

Twin Darkness Technical Report

Reviewed my project proposal before the upload on Friday. And attend my weekly Gaming lab with Anu where we finished last week's exercise.

We had to build a small office like structure and add in light and other effects. This will help greatly once I start level design for my project

30/09/2015 (Wednesday)

Came in early to spend a few hours in the morning working on rigging as I am having trouble importing a full rig into unreal engine 4, Did some research and found what I have to do to fix the problem.

02/10/2015 (Friday)

Spend some time writing up my reflective journal for September and submitted my project proposal

October Journal

Reflective Journal for October

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: October

What I did in October

05/10/2015 (Monday)

Came in for the lecture on finding resources using the library systems and the best ways to run searches, also worked a bit more on gathering textures for my main character

06/10/2015 (Tuesday)

Went to my graphics lecture with Anu where we learnt to add landscape and model terrain and add in different features such as water effects

08/10/2015 (Thursday)

Got hold of a tutorial that will help me import and use animations in unreal engine 4 using there retargeting system this will be a massive part of my project and is a big step forward

- <https://www.youtube.com/watch?v=xy9aLbZLdeA&index=34&list=PL1PsUO-UElmbNChC52h7TwcjdKlmoLVr1>

09/10/2015 (Friday)

Twin Darkness Technical Report

Came in on the day off to work on a number of smaller aspects of my projects and do some research

11/10/2015 (Sunday)

Still trying to figure out unreal engine 4 and blender import settings, no point trying to make animations if the import options don't work.

12/10/2015 (Monday)

Was able to import a full rigged model from blender, but have run into problems with movement of the bones etc. may have to redo on a simple skeletal mesh

13/10/2015 (Tuesday)

Found the source of the problem with my rig, it was too big for unreal to understand. So I now have to make a new one that matches unreal engine 4's understanding

14/10/2015 (Wednesday)

May have found the solution to my rigging problem while searching through unreal videos, believe I may now be able to import the default rig for unreal engine 4 into blender and use that as a template for my character. Will have to do some tests to find out.

Test 1 export and reimport after opening and editing in blender

- I exported the default skeleton and mesh out of unreal into blender opened in blender to find some of the bones of the rig not set right, once I fixed them I exported it again using blender to change it to a fbx file. Went through the default import, everything went in fine, I then had to go through the retargeted manager to set up the bones. Finally I retarget one of the animations (Idle) and It work

Next test will be using a custom character

15/10/2015 (Thursday)

First Meeting with my supervisor was on Thursday at 5pm in his office 3.18, what we discussed:

- *The outline of how the final year project is graded and what points I need to hit.*
- *We then broke down the first three stages of my project*
- *Do some market research on games that are out there and then compare them against mine in a table format*
- *Research the way games are build and the processes they follow storyboard out some of my level.*

16/10/2015 (Friday)

Continuing to try and work though my animations and importing problems by testing a number of Skelton's my animation retargeting is getting somewhat closer to what I want

17/10/2015 (Saturday)

Twin Darkness Technical Report

Worked a bit on storyboarding my levels out.

18/10/2015 (Sunday)

Found a new tool for blender that may help with my import problems will begin testing on Monday

19/10/2015 (Monday)

Decided today during the lab for project that it would be best to create my own animations for my characters will begin work soon

20/10/2015 (Tuesday)

Got all my stuff ready for the next upload which is the requirement spec, just need to begin filling it in

21/10/2015 (Wednesday)

Had my CA in graphics where we had to create a full level, definitely will be using some of the techniques in my own project

22/10/2015 (Thursday)

Did some more research into the animation process in blender

23/10/2015 (Friday)

Did the networking lab due on Friday this involved Java code and took up much of my free time. Also spend some time writing up the requirements document due at the start of November

24/10/2015 (Saturday)

Worked on prep for the requirements write up due in November

27/10/2015 (Tuesday)

Visited DIT to talk to a master's director about continuing on with my education after I'm done here also collected a bunch of props such as tables, chairs, monitors and beds to use as assets

28/10/2015 (Wednesday)

Wrote up all the function requirements form my project hand have got have way through the

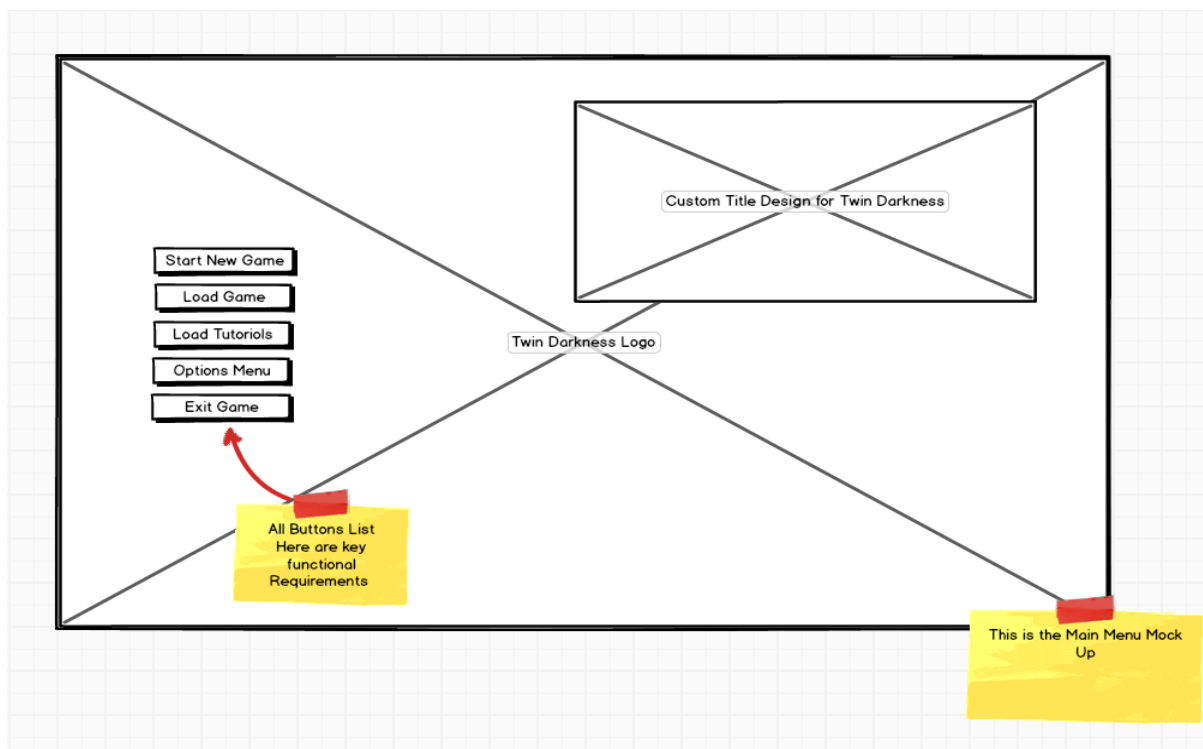
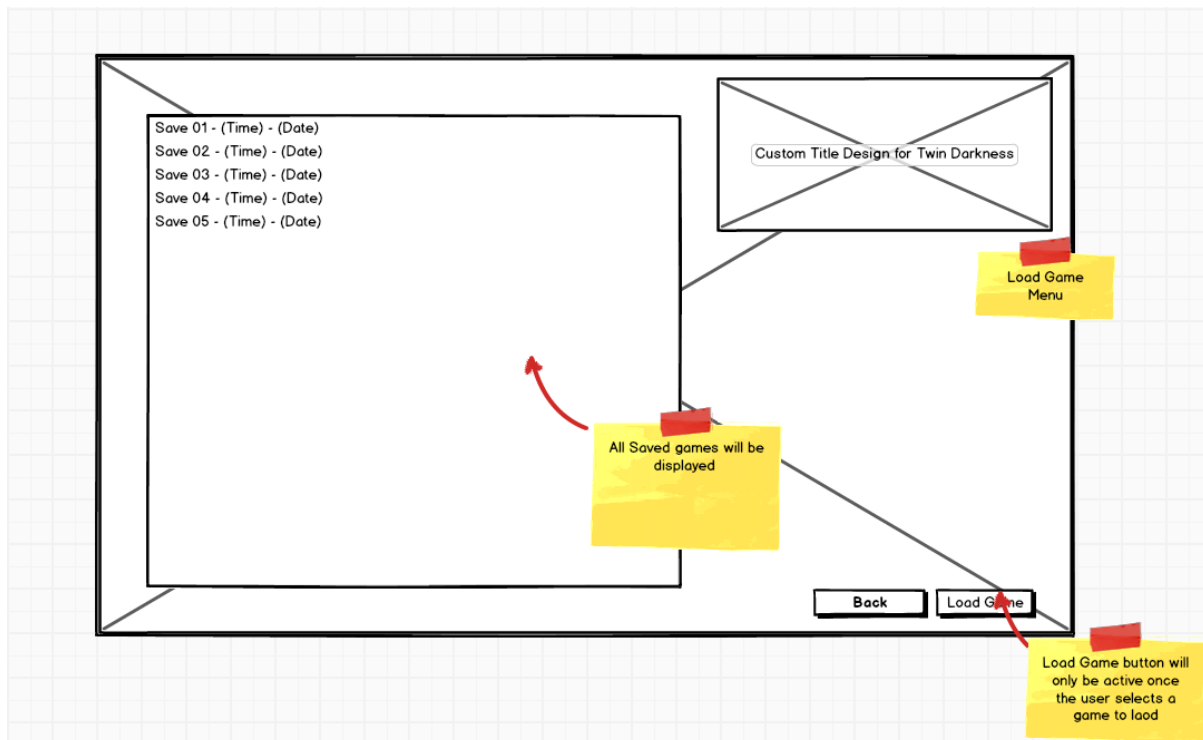
29/10/2015 (Thursday)

Finished all the non-function requirements for the project

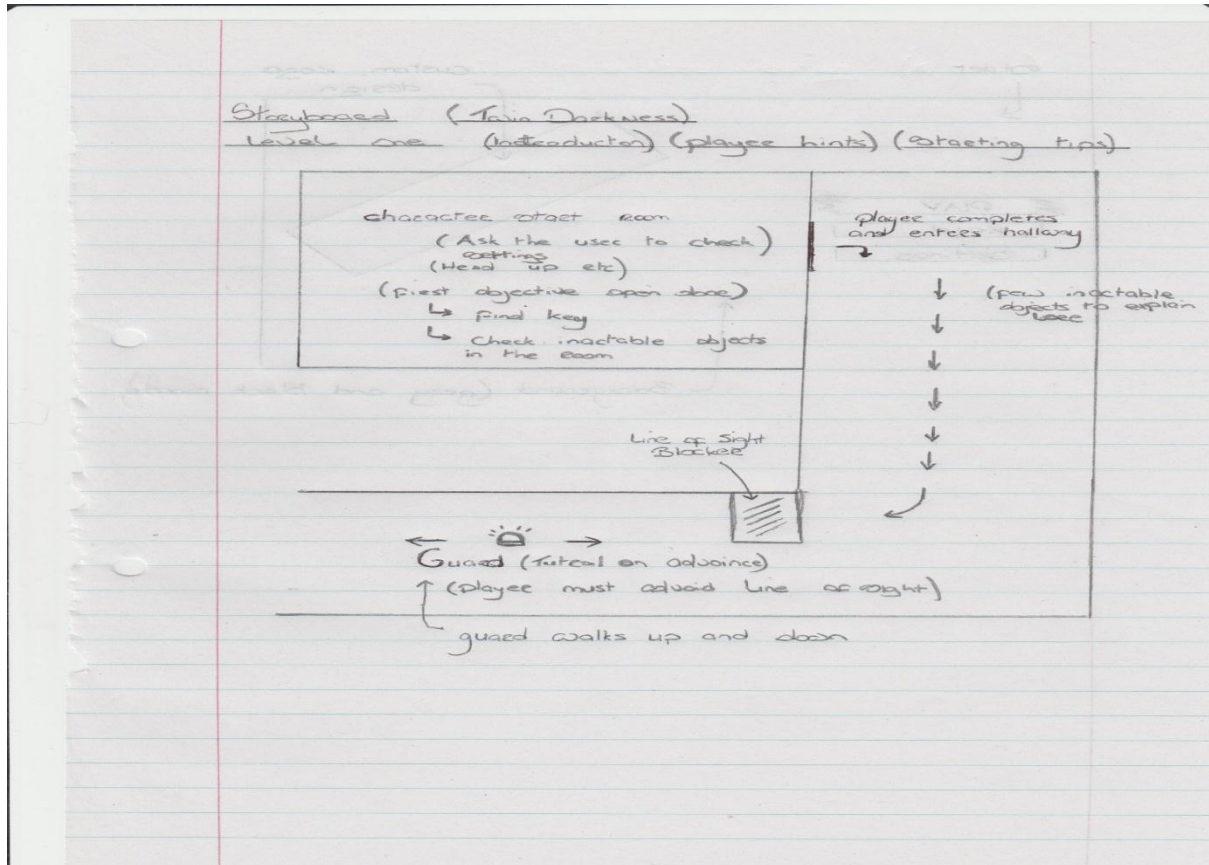
30/10/2015 (Friday)

Completed all the requirement documentation, just finishing up some last minute diagrams to include under the GUI Title

Twin Darkness Technical Report



Twin Darkness Technical Report



31/10/2015 (Saturday)

Submitted my requirement documentation online 6 days before the deadline, also will begin writing up this month's log book

November Journal

Reflective Journal for November

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: November

What I did in November

01/11/2015 (Sunday)

Start to collect images for the HUD design (HUD - Main Menu, Options and Loading Game) of my game. The HUD will be very neatly designed with moving backgrounds and a feel that is in line with the game.

Twin Darkness Technical Report

02/11/2015 (Monday)

Started on Monday by doing a little work on my networking Project which we will be due at the end of the month. Then I moved onto the HUD design for my game I have a bunch of cool images to be used as a background. Hopefully the HUD will be functioning sometime this week.

03/11/2015 (Tuesday)

Began learning Maya, will be looking at this when it comes to make props and items for my game

04/11/2015 (Wednesday)

Looked into adding moving gifs and videos as my main menu background, Unreal doesn't support it directly but I have already found a workaround using flipbooks, I just need to breakdown the gifs into their individual parts.

05/11/2015 (Thursday)

Found a mass breakdown programme that will break gifs down into their individual parts and will convert them to .png so unreal can accept them, I will then need to create a flipbook texture using all the imported images.

06/11/2015 (Friday)

Looked into making blood effects in unreal engine and learnt about unreels decals system which enables wrap able effects meaning you can get the full look of blood splatter on ground, walls and items in a room

Also collected together a bunch of blood texture images

This weekend I am planning on putting together the first draft of my main menu.

07/11/2015 (Saturday)

Did a bit of study on building an in game HUD so I can display out quests and hints to new players

08/11/2015 (Sunday)

Found some useful tutorials on in game HUDs and main menu design hoping to put something together during Monday's class.

09/11/2015 (Monday)

Didn't get around to much in the way of my project development did about an hour of research, before moving on and focusing on my up and coming Network CA which is worth 20% of my grade. This test will be on Tuesday

10/11/2015 (Tuesday)

As was mentioned yesterday, today was just focusing on my networking module and the CA I had. It did not go to well in my opinion. Did find some good links when I was studying at home that my come in handy when I return to development on my project

Twin Darkness Technical Report

11/11/2015 (Wednesday)

Returned to development on my Main Menu and the basic functional requirements that I set out, I have now gathered and made a number of test menus seems like a simple process will persevere further at a later date

12/11/2015 (Thursday)

Decided to return to modelling and level building at mention in an early log I have learnt how to make and build blood textures, I now want to move on to animation for enemy characters and give my main character some much needed time.

13/11/2015 (Friday)

Spend some time developing and model building

14/11/2015 (Saturday)

Continued development using blender and Maya for my custom models

15/11/2015 (Sunday)

Attempted rigging some small models and characters

16/11/2015 (Monday)

Fully rigged my main character and made 8 animations and tested the full import in unreal all looks good will now continue to study the unreal animation graph to set up a full character model

17/11/2015 (Tuesday) to 22/11/2015 (Sunday)

Going to be texturing and creating Materials over the next few days for my characters and finalizing all the final animations that they will be using. After that importing and developing the animation graphs in unreal will come up.

- <<https://www.youtube.com/watch?v=Lg4VhkWmZR4>> - Model Preview and animations preview

Finished all my character texturing and animations will be importing on Monday and developing the animation graph in unreal engine 4.

Also I have a meeting with my manager this week where we will go over my requirement spec and comparison report. My Manager already sent me some feedback through email.

26/11/2015 (Thursday)

Meeting with my manager today we discussed the following:

- We review my requirement spec and he gave me feedback on what needs to be update and add
- Talked about what I could do in order to express my project to a wider group of people
- Talked about developing my comparison report more

- I updated him on what I had done and what my plans are

27/11/2015 (Friday) to 30/11/2015 (Sunday)

Over the next few days I will be developing my monthly log and writing up my final document submission before Christmas which is the Project Analysis and Design document.

The monthly log is due on Monday the 1st of December

The Project Analysis and Design is due on the 4th of December

I may also they and finish small aspects of my project such as setting up a template main menu and some sample blood decals for the in game effects.

December Journal

Reflective Journal for December

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: December

What I did in December

30/11/2015 (Monday)

Did not get a lot done as I have an upcoming submission for another class. Did however find a detail tutorial on building a simple but effective mission system to direct the player through the levels.

Will hopefully try and test this in the coming days.

01/12/2015 (Tuesday) to 04/12/2015 (Friday)

Create a few more textures for my main character over this week and imported them into unreal to make sure everything matched up.

Created 2 videos showing those updated design and textures

- <https://www.youtube.com/watch?v=5YexBOgdm04>
- <https://www.youtube.com/watch?v=NLW1cbbHnRc>

05/12/2015 (Saturday) to 10/12/2015 (Thursday)

Have decided to work on the animation blueprint and final level design over the course of the week.

Will update on the progress in the next post.

Twin Darkness Technical Report

11/12/2015 (Friday)

Managed to find the answers to the big question I had which was "Can I animated my main menu?"

Did research over the past few days and discovered I can using Media Files and a bit of bending unreal rules

Here is the main resource I will be using (it's a YouTube video)

- <https://www.youtube.com/watch?v=qWrFXHxCTIM&list=PL1PsUO-UElmbNChC52h7TwcjdKlmoLVr1&index=68>

Hopefully over the weekend I will be able to put together the first draft of my main menu

12/12/2015 (Saturday)

Spend the vast amount of time researching the UMG and the different ways of coding the actions for the buttons within unreal engine 4

At the end of it all I build a small function menu with a start button and a quit button

13/12/2015 (Sunday)

Went a little bit more in depth with my research into unreal engine 4's UMG side

Research included:

- See how to import and play videos as the background
- Importing custom fonts
- Editing the base UMG

14/12/2015 (Monday)

Moved away from UMG, and decided to finish my character design and animation modelling in unreal engine 4

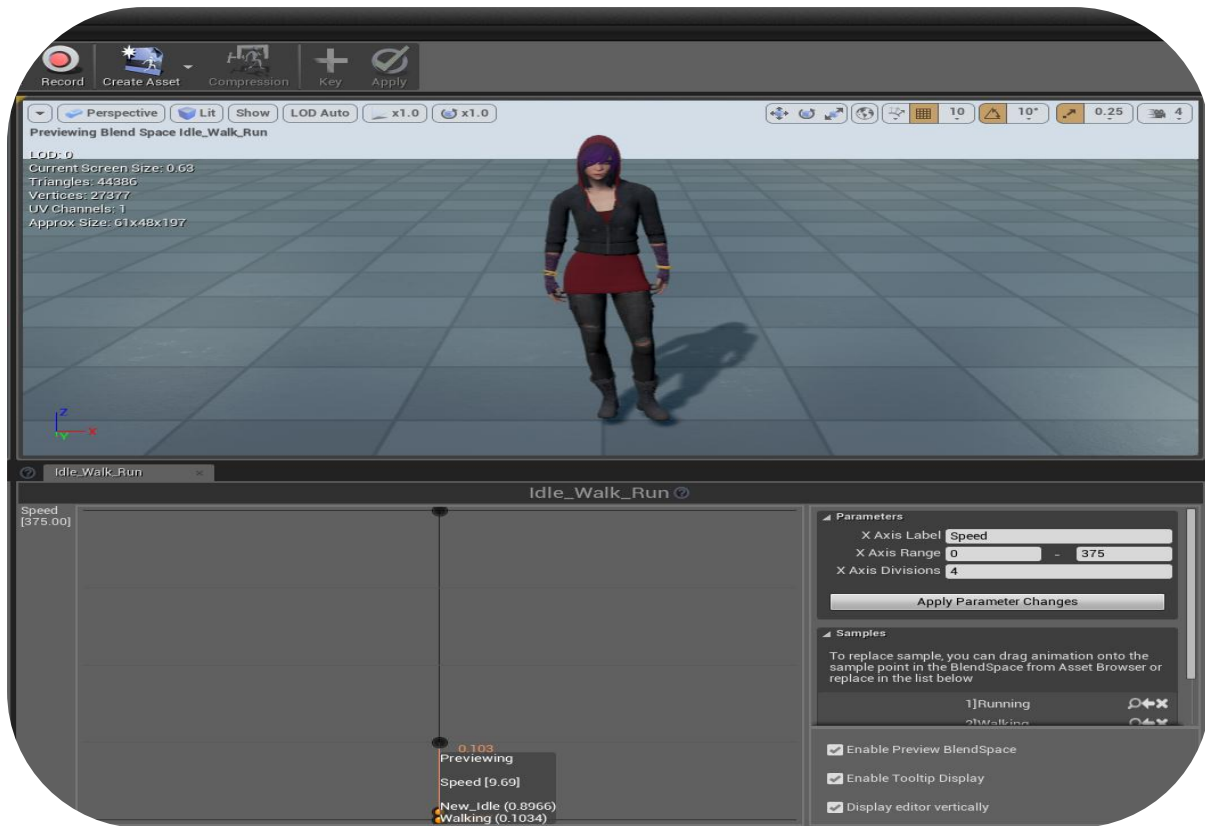
First Step was creating two blend spaces:

- Idle_Walk_Run
- CrouchIdle_Walk

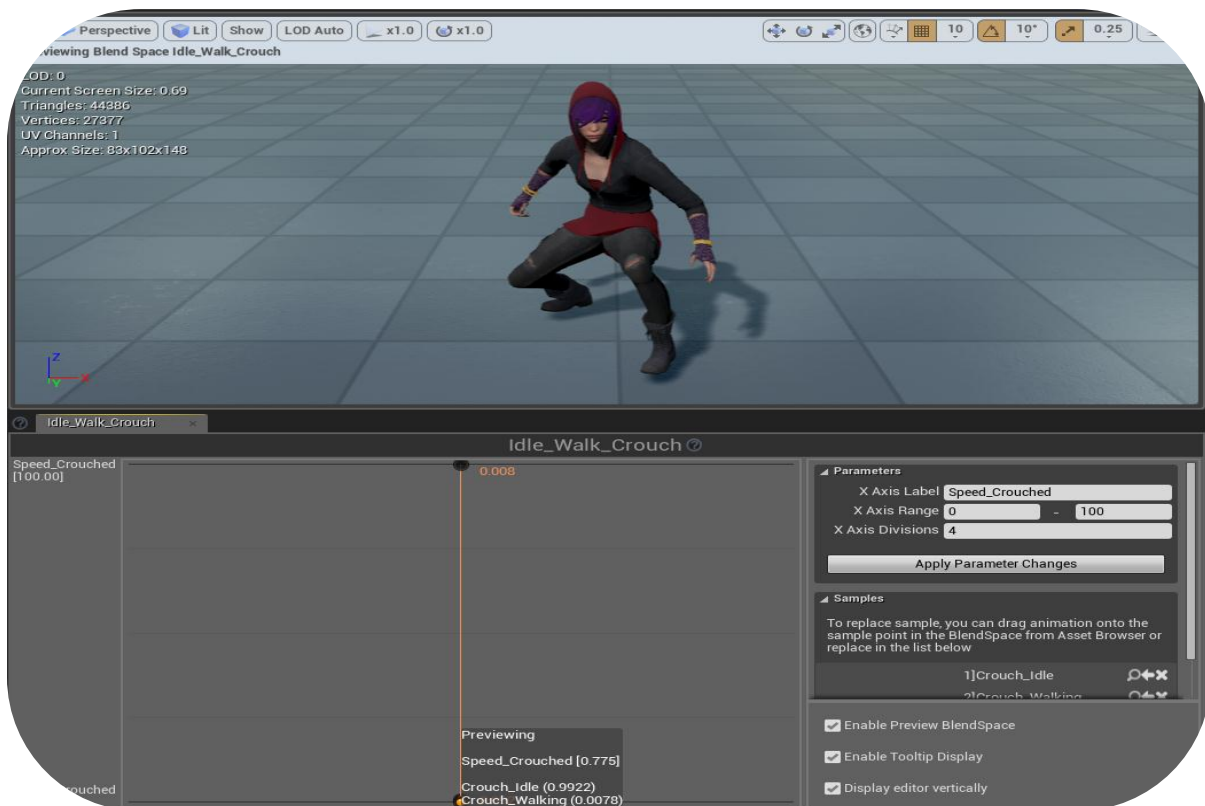
These provide a perfect blend between different animations

Idle_Walk_Run Blend space

Twin Darkness Technical Report

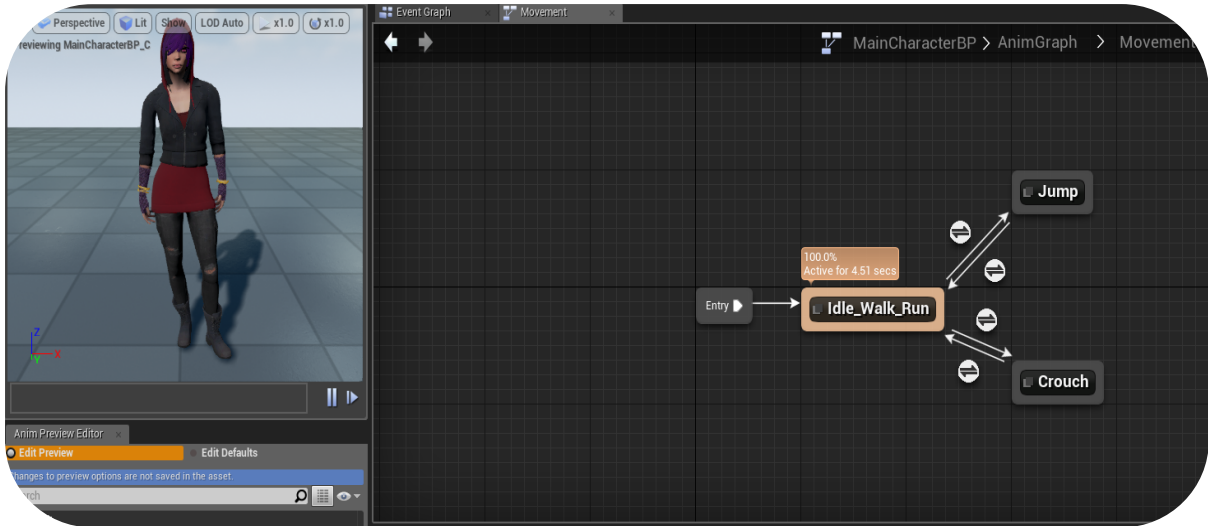


CrouchIdle_Walk Blend space



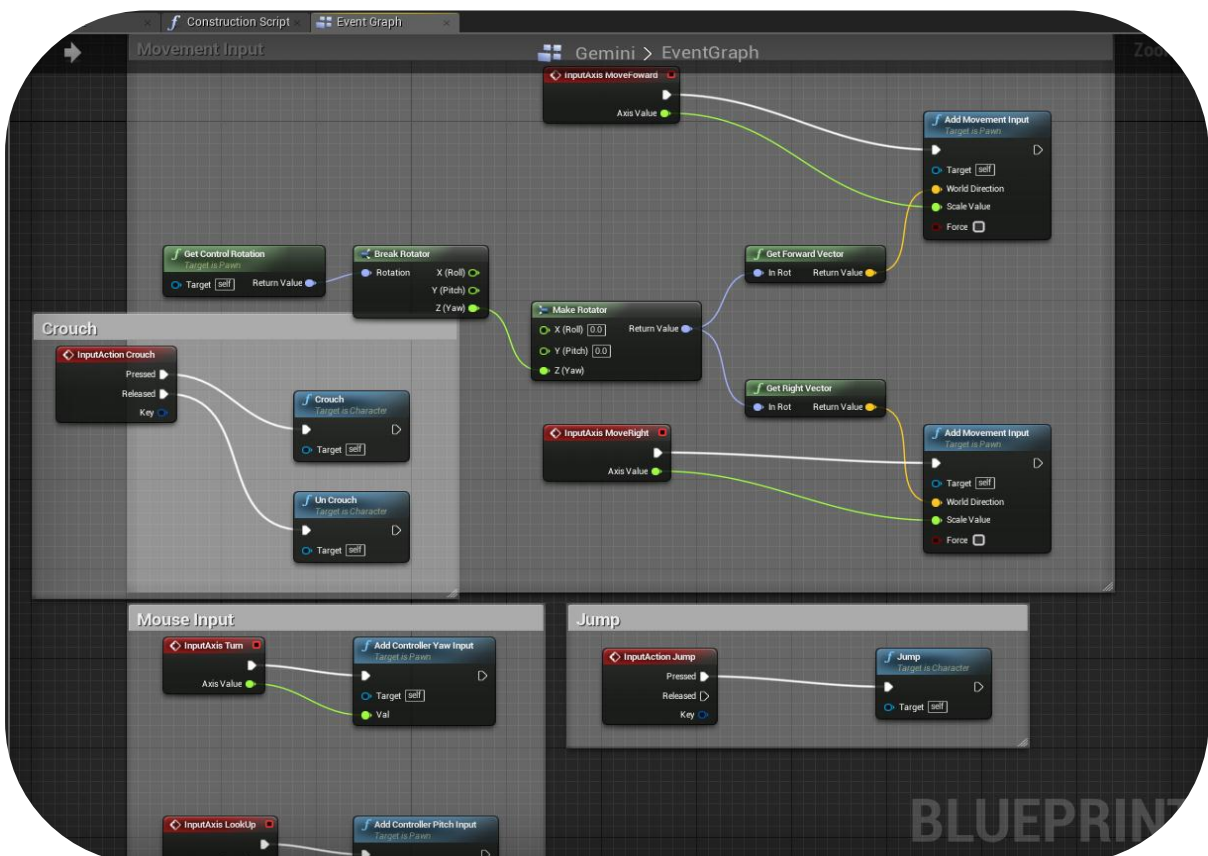
Next I created an animation blueprint for my main character, this handles all the different animations and how they work together

Twin Darkness Technical Report

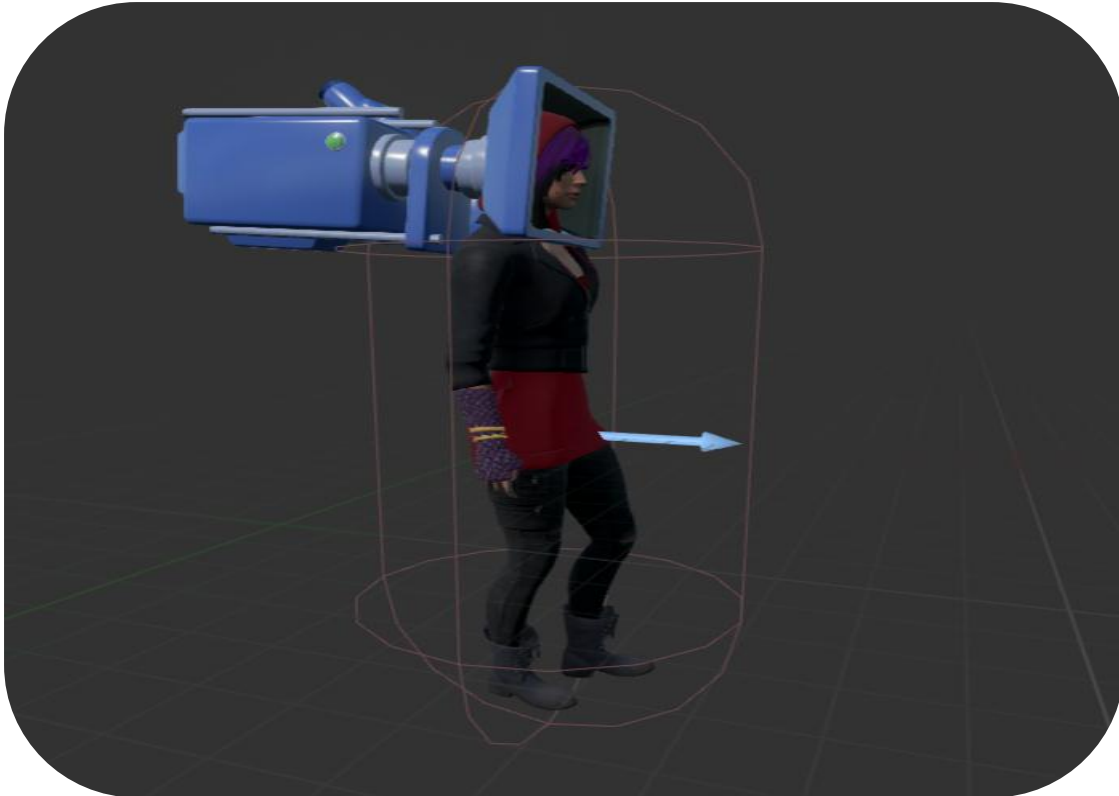


After this I created a custom character chart to handle input:

- Keyboard and mouse
- gamepad
- Touch



Then I modified the camera to make it a True FPS so the player is seeing through the eyes of the character



Testing was a success

15/12/2015 (Tuesday)

Worked a little on the UMG main menu added in a moving background image, after that moved on to the main character blueprint

Added in the following functionality:

- Jumping
- Crouching
- Crouching walking

After that I developed two videos for YouTube to show off the functionality I had created over the past few days

Video showing the full range of the character in-game

- <https://youtu.be/uI5zARE9R9g>

16/12/2015 (Wednesday)

- Updated the main menu with a better background movie
- Designed and build the full pause system UMG and functions into the game
- The user may now pause the game at any time during the level

Video outlining new main menu and pause system in game

- https://youtu.be/dU0U_prb51Q

January Journal

Reflective Journal for January

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: January

What I did in January

17/12/2015 (Thursday) to 18/12/2015 (Friday)

First Order of business was to make a test object and develop the pickup system for it. Started by importing a new object from blender to unreal a Backpack, Then I created the pickup mechanics using unreal engine 4 visual code. Testing was a success I can now create objects and place them in game and the use can pick them up

Next I started work on the Save and Load system, I estimated this system will take me a few days to build as it is very complex and makes up two of my functional requirements

Currently the user can save and load their location in game using the following key presses

G to Save

K to load

Next order of business will be to tie these keys to UMG buttons located in different menus allowing the user to load from the start of the game and to save the game when they are on the pause menu

19/12/2015 (Saturday)

Today I worked more on my save and load system I was able to setup the in-game menu to save and load on button clicks

I am still having massive problems when it comes to the hooking it up to the main menu (will have to research more into this) and hopefully find a solution

But for now I have meet the functional requirements of saving a game and loading a game in clear way for the user

Will try out some more idea's tomorrow for the main menu and move onto the user adjustable settings (which is another functional requirement) and the objective system for the user so they have a clear understand of what to do and where to go

20/12/2015 (Sunday)

Twin Darkness Technical Report

Success, I now have a full save and load system in my game the user can play, then save their location using a quick key press or from the full pause menu

The user can then load from the main menu or when they are in game from the pause menu

Video Highlighting Save and Load Functions

- <http://youtu.be/YOou2nLynXY>

21/12/2015 (Monday)

Worked on the start of my options menu for my game which is another one of my functional requirements

First: I created all the menu setup I would need such as hiding and showing certain aspects of the page only when the buttons are clicked

Second: I build my first set of options to handle resolution and have decided to give the user 4 options low, medium, high and Ultra

Will be adding more options as the days go on

22/12/2015 (Tuesday) - 23/12/2015 (Wednesday)

Discovered a problem with media textures that I was using in regards to the main menus of my game as they were not playing when I switched the gam to standalone or when I would package the game which defeats the purpose

Lucky I found the answer through trial and error and am happy to say it work

I just had to add into the first level blueprint a play function that is automatically run when the game starts up

Next I moved back to the options list and did the following features:

An enable full screen button that lets the user switch in and out of full screen (Resolution is auto to the screen resolution the user is using)

Add in more options on to the main menu they are:

- View Distance
- Anti-Aliasing
- Post Processing
- Shadow Quality

Will add the final view options in the coming days and 4 pre-set configurations to make it easy for the user

24/12/2015 (Thursday)

Finished my options menu with all included functions

Options Menu Highlight Video

➤ <http://youtu.be/tco-KMKaG90>

25/12/2015 (Friday)

Yes I did some work, Just added a pause menu and the options menu to the tutorial level I am creating to allow the user to pause during the tutorial and gave them the options menu so they can match up there settings with their system before playing the full story.

Also added in full range of character controls using a control and cleaned up some other points along the way.

26/12/2015 (Saturday)

Created a number of item assets and began working on allowing the player to interact with items around the world

Was able to place in a computer with a moving monitor when the player gets close the can press E and it will open the monitor and display some information to them.

When they move away from it they can no interact with it again until they move back into range

This is a first step in the interacting items I am developing

Video Highlighting interact able Item

➤ <http://youtu.be/50wC30Dv8PM>

27/12/2015 (Sunday)

Continued on with setting up interact zones and items within my tutorial level completed the HUD design that appears when the player is reading a document from a screen in the game

Made a video outlining this feature and uploaded it to YouTube (Video link is attached above)

Also I made a custom Material and set up a blueprint that when the user gets close to the object the object gets a highlight feature to alert the user that they my interact with it.

28/12/2015 (Monday)

Today I began building in matinee to create an open and close system for doors etc. I was able to create a number of matinee scenes that open the doors, the user has to press E like when they are interacting with the computers or files in game

The middle panel of the door also has a highlight like the computer alerting the user that it can be interacted with

The user can open the door using E and close it using E this will come in handy when running from the enemies in the future

Going to be focusing on this over the next day making it stream looking and clear for the user

Twin Darkness Technical Report

29/12/2015 (Tuesday) to 04/01/2016 (Monday)

Create my first room and tried out the different lighting features in unreal engine 4 to see what it looks like fully rendered.

This room will be part of the tutorial over the next few days I will be building the full tutorial level and the quest system etc. to prepare for the Midpoint presentation.

Prepared half of the initial load hallway and prepared part of the stairs and build more lighting in that area today, also watched a number of videos on good lighting techniques to better prepare the level.

Created nearly all of the upper level that will feed into the vent system that I will be building in this vent system the user will have to use the crouch and enhanced vision techniques of the game

Worked on lighting the level before I go any further with the building of it (it's so annoying)

Continued to work on lighting I also added in the floor to the rest of the textures and build a vent trap door system that blocks out all light this is important for when I am building the vent system, even created a whole new matinee to handle the opening a closing of the vent also decided to place it on a delay so after its active the user only has 10 seconds and the vent shuts behind them.

This will build into the core of the gameplay of running and hiding.

Complete updated development log video and posted it to YouTube showing a walkthrough of the current state of the tutorial level.

Video Highlighting the Current Level development state

➤ <http://youtu.be/fzwvszytBvw>

05/01/2016 (Tuesday)

Exams start tomorrow so development will be slowed for the next day and a half after that development will resume since my other exams are already past modules due to CA's

06/01/2016 (Wednesday)

Finished my exam (should be a passing grade), did a little on the in-game HUD design looks pretty good if I do say so myself

07/01/2016 (Thursday)

Set up a Kill screen template and a death animation system and did some house cleaning

08/01/2016 (Friday)

Created a Full death system and animation with sound and created a better Active power system (better drain and regen setup) also created the First draft of the in game HUD for the tutorial

Currently the HUD is an Active health bar that drains and a non-setup (currently) stamina bar that will allow the user to sprint for a short time until the bar drains

Twin Darkness Technical Report

09/01/2016 (Saturday)

Setup the first stamina draft system currently when the player holds Q they will sprint and the on screen stamina bar drains, when it reaches 0 they are returned to the base walk speed on the character

Also finished the up floor of the game and created a bunch of screen textures in red to make it look like there any system problems and built the full lighting.

Also created Development Log 9 video (video link here) have decide to try and create more videos highlighting the game development!

Tomorrow will add in a few more features to the stamina system like sound when the player is near 0 they sound out of breath and will auto start to slow

Then I will move on to creating the next hallway, room and vent system that will make up the final part of the level

Highlight Video showing Active and Stamina Bar System

➤ <http://youtu.be/CHxDG-1tpy8>

10/01/2016 (Sunday)

Completed the sprint system to what I wanted my do more work in the future, but current system allows the player to speed up and slow down, when the stamina reaches 15/100 the character will begin to sound out of breath.

Also setup the condition that the player must be moving in order to trigger the sprint setup meaning they can't drain there stamina while standing still

Then moved onto the second round of level design and was able to build the next hallway which will be blocked off at the end forcing the user to find

Another way leading to the vents, this hallway will also introduce the sentry A.I, which the user will have to trap or trick, if they want to get out alive.

Should get onto this over the next few days

Highlight Video showing in-game shows

➤ <http://youtu.be/xsReAW1A0SM>

11/01/2016 (Monday)

Only did a small bit, since I have my next exam tomorrow. Managed to setup a few more branch statements to control the sprint condition better such as the user must be moving forward can't be crouched and if they stop the bar auto regens even if they are still holding the sprint key

12/01/2016 (Tuesday)

Twin Darkness Technical Report

Did a bit of work on the medical bay setup in the game just visual design imported a few custom meshes (beds) and build the frame work, still have one more exam on Thursday so Wednesday I will not be doing anything project related as I have to study

Once finished I go back into full production mode and hopefully finish the tutorial over the next week or so before I am back at college.

13/01/2016 (Wednesday)

Didn't have time to do much in terms of design, did however create a new Quick update video and uploaded to YouTube just highlighting parts that are in production

Highlight on current development parts

- <http://youtu.be/LW8VQWMjqAY>

14/01/2016 (Thursday)

Created 2 More Videos, this time highlighting the save and load functions of the game and the checkpoint and auto save functions

Created another video highlighting the options menu

Post above with when I build the systems to make it easy to see in the document

Finished the building of the medical bay made some custom glass textures for it that I will soon cover with blood decals

Highlight Video showing Checkpoint system

- http://youtu.be/1_6_wzirNos

15/01/2016 (Friday)

Finished development on the medical bay added in custom made decals for blood effects, create a custom spark to add the effect of a broken door

Created a video highlighting the finished medical bay

- <http://youtu.be/BFhCaxXAn7Q>

Highlight Video Medical Bay Create a new tablet model that will be used for interact able pickups and did some research into creating dead body's inside unreal engine 4

16/01/2016 (Saturday) to 18/01/2016 (Monday)

Over the weekend I worked a lot on hammering down a clear story that I can build on, along with designing the humanoid enemies in the game

I was able to make and model 2 different base types I then did research and testing into sockets in unreal engine 4 and found I can use the 2 base models as templates for others giving endless possibilities.

Twin Darkness Technical Report

Once I was done with that I went back to the level development since the midpoint will be on in the second week in Feb, I hope to have the full tutorial level build showcasing all the different designs and mechanics I have been able to build

Continued to develop the tutorial level layout also added in character models to make the level seem more alive (Making update video tomorrow - 19/01/2016) to showcase updated level.

19/01/2016 (Tuesday)

Completed work on the second hallway, guard station and destroyed lift.

Made a bulkhead door and created a one way animation that the player can use once to open the bulkhead door to show the broken lift.

Made a video showcasing the current level of the tutorial

Highlighting Video on current level of development.

➤ <http://youtu.be/it5TH03YpG8>

Tomorrow I will move onto enemies and the A.I I will be using and hopefully building the simple quest system in the coming days

20/01/2016 (Wednesday)

Started off by adding sound to a number of the matinee's in the game that control door functions etc.

The moved on to the development of the Sentry A.I, was able to get it to chase and once it catches the player they lose and die

Added a number of in game assets lamps and weapons

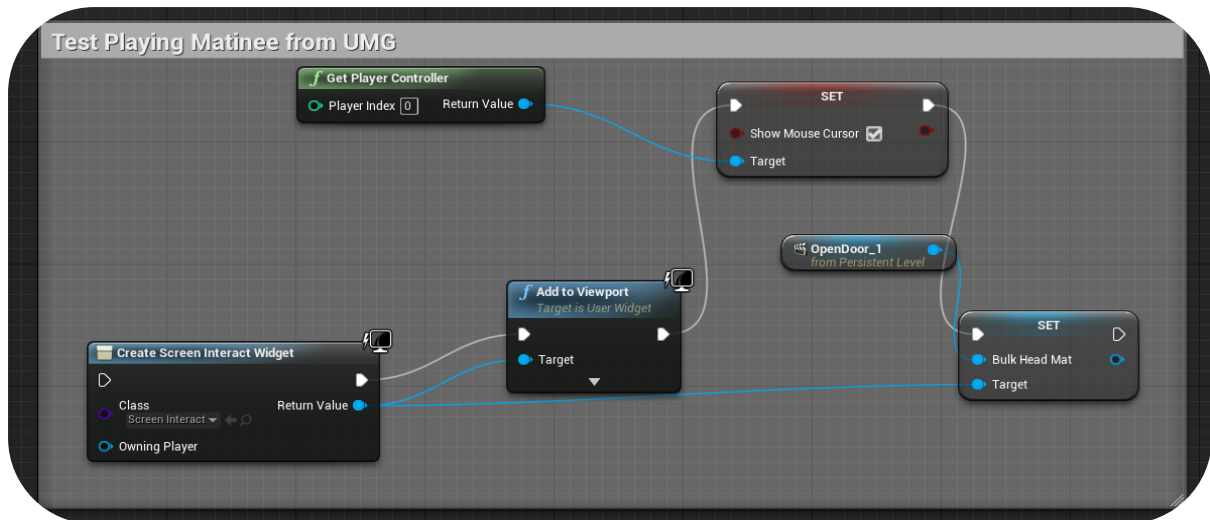
Going to continue to build on the A.I and move onto laying out the final zone in the tutorial over the coming days

21/01/2016 (Thursday)

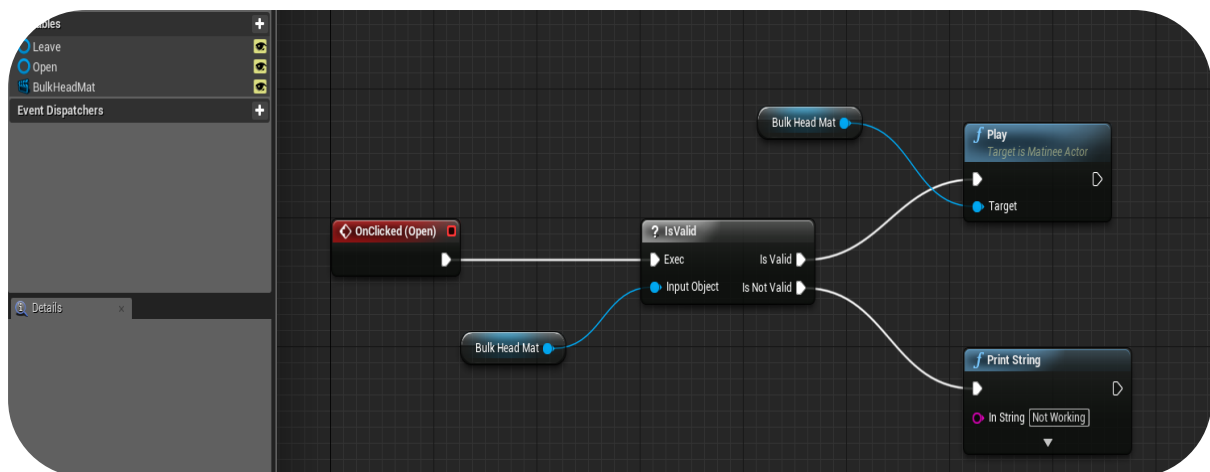
Started off today with a bit of development testing (being able to connect in-game matinee's to UMG widgets)

Testing worked and I then moved onto building a new in-game interaction system

Images of my brilliant solution



Second Image of my brilliant solution



Started creating the upload for January in word and attaching all the fancy development videos.

Build more of the security room and made some new props tables, lamps and a logo for the base using blender and Photoshop.

22/01/2016 (Friday)

Completed work on the security room created a number of new assets including server consoles pipes and wires to give the room the alive look.

Did some touch up work on other assets around the level.

Fixed some physics assets

Tomorrow I will create the new/second interactive system

23/01/2016 (Saturday) to 24/01/2016 (Sunday)

Began work on the new interactive system and did some more house cleaning

Twin Darkness Technical Report

Made the judgment of changing the tutorial to a prologue and moving it into the chapter select part of the menu (I did this because having a separate

Tutorial made no sense from the point of view of the game I am trying to create)

But making it part of the story and making it a prologue genius. It will go towards the telling of the story

25/01/2015 (Monday)

Started back college, did a bit of clean up during my free time in the classes found some bugs such as certain things not working the way they should was able to fix it.

Did up a to-do list for the 4th of February submission so I know what I need to do

Wrote out the headings of my requirements and examined each one and gave each one a %

Current State of My Requirements: (Headings/Requirements may be updated or changed)

- ❖ Requirement 1: Play Game/Movement
 - (Completed 100%)
- ❖ Requirement 2: Save Game
 - (Completed 100%)
- ❖ Requirement 3: Load Game
 - (Completed 100%)
- ❖ Requirement 4: Exit Game
 - (Completed 100%)
- ❖ Requirement 5: Modify Game Settings
 - (Completed 100%)
- ❖ Requirement 6: Load/Complete Tutorial Level (Requirement Name Will Change in Report)
 - (50%)
- ❖ Requirement 7: Interact with in-game items/systems
 - (Completed 100%)
- ❖ Requirement 8: VR Setup/Interaction
 - (25%)

26/01/2015 (Tuesday)

Today began the major work of creating the full technical report both for the mid-point and final submission.

Currently I have completed:

- Executive Summary
- Initial Concept

27/01/2015 (Wednesday)

Still prepping my technical report taking it slow and steady since it will be my final report. Started the research section of the report.

Twin Darkness Technical Report

28/01/2015 (Thursday) to 31/01/2016 (Sunday)

Finished off the research section of my technical document will be moving onto the last 3 sections today and tomorrow, tomorrow is the first independent since I have been back hoping to knock out most of the documentation tomorrow.

Was able to finish most of the needed documentation on Saturday. Including the updated project plan.

Will be moving back to level design.

February Journal

Reflective Journal for February

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: February

What I did in February

01/02/2016 (Monday)

Hoping to get a bit of Oculus testing in this week, finished the report for its submission on Thursday, so it's full steam ahead on development for the mid-point

Today during the break we had between classes I did a level design created a few assets and updated parts of the level

Sent an email to my NEW supervisor attached my report and a link to my YouTube channel where he can see all my videos highlighting milestones in development

Also made the first in a series of videos to be attached to each one of the requirements showing them completed in-game

02/02/2016 (Tuesday)

Did a bit of level work and design

Made 5 new requirement videos:

- Save Function
- Load Function
- Exit Game
- Modify In-Game settings

Twin Darkness Technical Report

- Interact with in-game items and systems

Had to modify my report and remove certain sections as they are not needed for the mid-point emailed Eamon to get clarification on size etc. (waiting on response)

Still have not heard from my new supervisor emailed him on the first

03/02/2016 (Wednesday)

Got a response on my report and was told it was fine so I finished up and added a few final features to the report

Uploaded it for the Thursday deadline

Tomorrow during the free time I have I will hopefully knock out the slide presentation that I need for next Wednesday (thinking about 10 - 14 Max)

Still waiting to hear from my new supervisor

04/02/2016 (Thursday)

Tested out my laptop in the room where I will have my presentation made sure everything worked (Will test again tomorrow to make sure it still works)

Started creating my presentation

05/02/2016 (Friday)

Tested out hooking my laptop to the projector to make sure it still works and Yay it did so I should be covered for Wednesday

06/02/2016 (Saturday) to 07/02/2016 (Sunday)

Spend the weekend putting together my slideshow for my presentation on Wen at 4.20

Have 12 slides ready to go and have a number of options for the prototype part of the presentation

Will more than likely just continue with the level design and try to finish the prologue mission fully in the next few weeks of development

08/02/2016 (Monday)

So it's Monday, been a week since I contacted my new supervisor still no response after 3 emails (at this point couldn't care anymore)

Today did some final testing in 3.04 for my presentation and worked more on the level and feel of the game

Changed the following

- Background movie playing removed the storm replaced with falling white snow on a black background much cooler

Twin Darkness Technical Report

09/02/2016 (Tuesday)

Did some prep work for the demonstration, build out a full version of the project so it would not be depended on unreal engine 4 editor that can eat memory and power.

Printed out copies of my slides for both the examiners and one copy of my mid-point report

10/02/2016 (Wednesday)

Mid-Point presentation day what happened:

- Felt it went over well
- Met Ralph my supervisor for the first
- Didn't really take much away from it because I didn't really receive much in the way of questions or feedback
- Got ask pretty much 0 questions
- Had to give Ralph my copy of my report as he had the wrong one with him, the copy he had was the submission from October.

Overall I think it went well I should get a good grade

11/02/2016 (Thursday)

Got contacted by Ralph who asked me to sign up to a project management site, Asana to handle tracking my project so I did, I found that the tools that the site offer are great and I made a number of tasks about uploading my document's and even adding a link to a playable demo of the game

Will be using it going forward so it was a nice first contact

12/02/2016 (Friday)

Spend the morning using the Asana tool that my supervisor told me about I added all the major things I had done and what I still have to do so my supervisor can review what I have done and what I still have to do

Going to tackle the vent system of the prologue this weekend along with trying to improve the level streaming overall

13/02/2016 (Saturday)

Began work on the vent on the prologue level this will lead to the next area and advance the plot

14/02/2016 (Sunday)

Completed a good portion of the vent system leading to the second floor, also figured out how to block out light complete from a material,

Thus making the vent system complete black meaning it will force the user to use their powers

Over the next week I will be adding effects inside the vent to make it more alive, also will try and knock out the A.I Guard need for the game and for my A.I project as a whole

Twin Darkness Technical Report

15/02/2016 (Monday)

Completed the layout of the vent and where it will end and the new room will begin, placed and test the visual look and feel of props that will be used inside the vents made sure light and caps where fixed.

16/02/2016 (Tuesday)

Did some more work on my final document updating the initial concept description and started work on the implantation side of the document, I got Main Menu and Options done

Finished the full vent layout and prop layout, enabled a matinee that triggers a lift for the user to take while in the lift and finished by building the lighting.

Also modify the movement speed while crouched from 75 to 100 (I felt and got feedback from testers that it was to slow)

Modify the drain on the active power this is a temp setting still need to find the sweet spot between drain and usefulness

17/02/2016 (Wednesday)

Did some clean-up on the recent development in the prologue level

Along with updating some character functions

18/02/2016 (Thursday) to 21/02/2016 (Sunday)

Over the course of the last few days I did clean up and some research into text effects I can use in-game

I created a new update video highlighting the full vent system with all its style and features

➤ <https://youtu.be/fkcclcUjypo>

Began work on the second floor

22/02/2016 (Monday) to Friday (26/02/2016)

Started work on, improving the overall system of the game by including:

- Steam Connection (which could be used in the future if the game is greenlighted)
- Improving level loading

Created a new logo for the main menu and added animations on buttons

So when they are hovered on the get a yellow glow around them to better show the user

Also I add a hover sound to the buttons as well

Finally I created a new blend space for the Enemy animations (idle walk run)

Then created a new blueprint model for said enemy so they can be animated in game. This is the first step in the new enemy AI I am developing for this and for my AI module project

March Journal

Reflective Journal for March

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: March

What I did in March

27/02/2016 (Saturday)

Activated Steam in-game

(Making it ready if the game is greenlighted in the future)

Activating Steam gives access to the wide range of resources including:

Achievements:

- Online
- Steam Community

Video Highlighting Steam Active in game

- <https://youtu.be/AHzcATYOPdw>

28/02/2016 (Sunday) to 04/03/2016 (Friday)

This week I have been working on the prologue level trying to finish the second floor of the level

Along with gather and preparing for the main A.I design and its decision tree

The A.I

- Will be able to hear and see
- Will be able to destroy the player if caught

Also on Thursday

I implemented most of the newly designed Save and Load System why?

Twin Darkness Technical Report

This system is more stable and better built to industry standards based on research done on the unreal engine 4 forums with other developers. It also allows for saves to be loaded across levels.

05/03/2016 (Saturday)

Did mostly level building on the second floor by my estimate it is about 60% to 70% finished

Sorted out the game icon

Tested out some save and load setups for future use

05/03/2016 (Saturday) to 09/02/2016 (Wednesday)

Worked more on the development of the prologue level this included designing the new curved walls and setup

I would say currently the level is 80 to 90% complete

10/03/2016 (Thursday)

Finally got a hold of the VR headset and did the following tests

- Laptop will not run the oculus rift
- Desktop will run with ease oculus rift

Tested the in-game level using the VR headset and all camera controls are set up the right way.

Meaning the game is able to be run with the VR system engaged

11/03/2016 (Friday) to 13/02/2016 (Sunday)

Completed the second floor plan layout

- All walls and doors
- Began roofing

Created a number of new assets

- Medical Bed
- Containment Units
- Number of cloth textures
- Medical Table

In the next day or so I will complete roofing and move to lighting

14/03/2016 (Monday)

Completed the roof build and full interior

Completed Lighting for the second floor

Modified and prepped the Aries A.I. model by getting the correct size in relation to the game

Twin Darkness Technical Report

Created a new highlight video showcasing new second floor and preview of the A.I model

- <https://youtu.be/YgvdvkWkgpw>

15/03/2016 (Tuesday)

Began by adding and assessing, what models I will be using on the second floor such as beds, tables and lights etc.

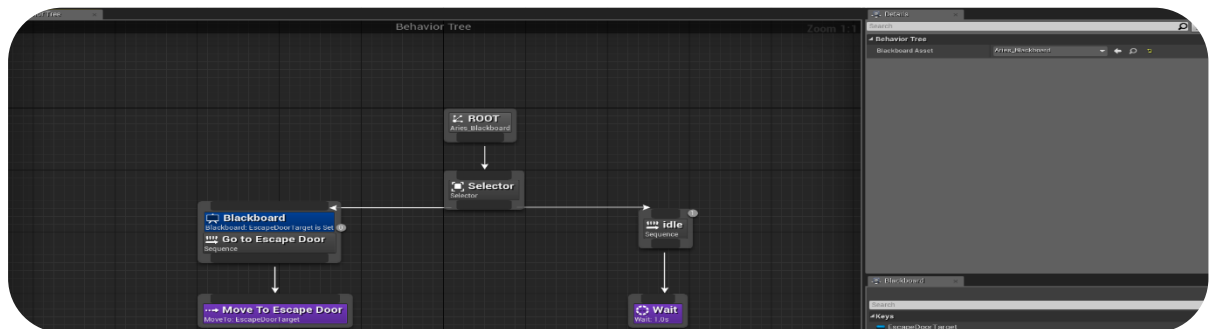
Then I moved onto Aries (My advanced A.I)

Setup a number of conditions and classes

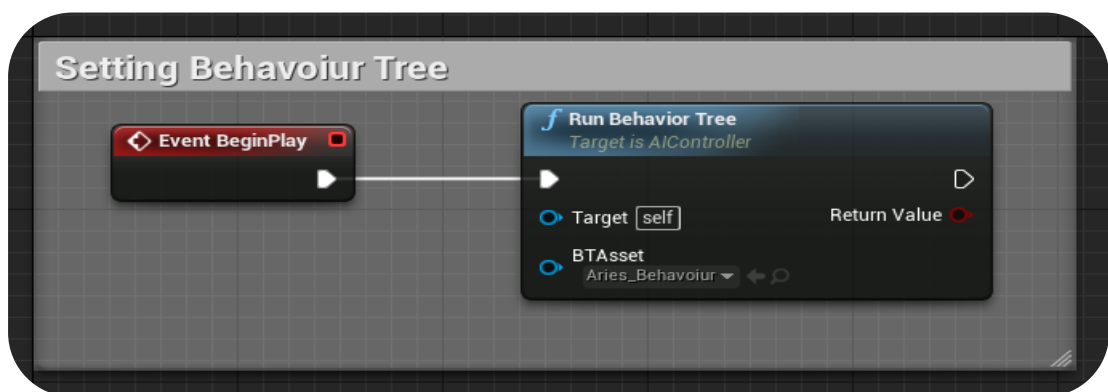
Created:

- NavMesh bounds
- Aries behaviour tree
- Aries blackboard
- Aries controller class

BaseAriesBehaviour



SettingBehaviourTree



Fix movement problems with the Aries model so he now blends correctly through animations

15/03/2016 (Tuesday)

Update 2: Built more into the Aries A.I

Created a number of new branches to control based guard setup

Twin Darkness Technical Report

And began steps for the next branches

- Sight
- Audio

16/03/2016 (Wednesday)

Continued building Aries A.I

Added full functions for Audio detection the Aries system can now detect and investigate sounds

Created a number of conditions such as time to investigate, speed and resets

Next stage will be sight detection of main character

17/03/2016 (Thursday)

Complete all initial work on the Aries system,

The Aries system now has full movement capabilities and attack capabilities

Aries can both use sight and audio detection in game

Next stage will be modifying base values and setting up level conditions (simple value and design changes)

Overall Aries is complete

18/03/2016 (Friday)

Complete work on the A.I and the document needed for my A.I module

Created 4 videos highlighting each of the branches of my tree in action

4 A.I Highlight Videos

- https://youtu.be/H_qcB6Irlgs
- https://youtu.be/FjNBAOm9_ps
- <https://youtu.be/Yol1LI7RV8w>
- <https://youtu.be/aUKHuYsjxF0>

Set up the second level to meet the needed conditions such as setting up distance correctly etc.

19/03/2016 (Saturday)

Moved onto the last real part of my project, developing the story and how it is laid out and modifying/styling the prologue level with extras

20/03/2016 (Sunday)

Continuing level building

Designing new design features and adding effects onto the level

Twin Darkness Technical Report

21/03/2016 (Monday) to 28/03/2016 (Monday)

Ground floor effects and re-style complete

Next step is to redo the intractable console (either remove or find a new way of doing it)

Once done will move to the vent and second floor (should be done by the end of the week)

Managed to develop and set up better decals throughout the level using unreal engine 4 plugins and setting changes

Still working on a new intractable console (may be able to create a keypad system)

Just more effects and detail work

Nearly done the second level

29/03/2016 (Tuesday) to 30/03/2016 (Wednesday)

Got my hands on the VR equipment again to perform final setup and work towards the final presentation

Created two videos highlighting the VR setup on my first level

Uploaded here:

- <https://youtu.be/Ksu7POufoDg>
- <https://youtu.be/iDxJDsKZg0s>

Each video highlights different features and what they look like in VR

Requirement 8 (VR) is complete as of this date

April Journal

Reflective Journal for April

My Details

Student name: Jamie Mulvaney

Student Number: x12358831

Programme: BSc in Computing Gaming and Multimedia

Month: April

What I did in April

01/04/2016 (Friday) to 03/04/2016 (Sunday)

Continuing development on the prologue level to complete that requirement thus finishing the main project

Twin Darkness Technical Report

Then the next step would be the document and poster development.

Started Testing using five volunteers:

They were used to test Usability and asked to answer a question survey on the usability

In the next week or so development on both will be greatly reduced due to up and coming exams

04/04/2016 (Monday) to 16/04/2016 (Saturday)

In full exam mode, over the next two weeks, so development will be light

What was done over this light two weeks is the following:

- Continued to finish off development of the final few rooms of the prologue level
- Started to prepare for the submission date on the 11th of May
- Thinking of poster designs
- Created new button highlight effects based on feedback from users

17/04/2016 (Sunday)

Began clean-up and finishing touches on the prologue level

Began getting files together for the final document

18/04/2016 (Monday)

Began work on the implementation heading under its subheading Game systems

19/04/2016 (Tuesday)

Began work on in-game implementation and AI

20/04/2016 (Wednesday)

Began Work on Main Character implementation headings to go:

- Animation setup
- Movement Control implementation
- Sprint System implementation
- Passive Power System implementation
- Active Power System implementation

21/04/2016 (Thursday)

Finished the write-up on animation setup and movement setup

Began work on the 3 power setups hopefully finished by Friday

22/04/2016 (Friday)

Complete sprint system, passive power and active power implementation

Added one final heading covering the death system still need to finish

Twin Darkness Technical Report

Once done will move onto the last main heading which is User Features Implementation this will cover:

- Steam
- Steam Achievements
- VR

24/04/2016 (Sunday)

Completed all sections of implementation

Will be moving on to testing for the rest of the week

25/04/2016 (Monday)

Did some work on the final document added in all required sections still need to complete them fully.

Did work on testing and created a customer survey for the customer testing heading, this will allow me to enter a number of graphs etc. thanks to survey monkey

Did work on the final level and main menu adding in story and lore.

26/04/2016 (Tuesday)

Began work on the poster for the final deliverable date.

Finished 3 sections in the report, just to get them out of the way since the date is coming up

Milestones and Hurdles

Further development and research

Graphical GUI and In-game effects

One section left is testing which will be written and done this week

27/04/2016 (Wednesday)

Redid GUI section and added in new section more model previews and effects, reason was to make the report clearer and more presentable.

Updated survey for customer testing with some new questions, hoping to conduct customer testing tomorrow.

28/04/2016 (Thursday)

Conducted a customer evaluation test using the survey and latest version of twin darkness

This survey was more of an opinion based survey to gather information on what users thought

Will try to get a few more people tomorrow, if not will write up the customer testing section

Currently writing under the testing heading Unit and Integration testing hope to finish today

29/04/2016 (Friday)

Started the final heading, section which is customer testing

Still trying to collect more data and feedback, while be closing the link tomorrow and finalizing that data

30/04/2016 (Saturday)

Completed the customer testing section with 7 participants including received feedback and graphs, will be going to the printers on Monday, to get the Poster and 2 copies soft bound and printed.

Hoping to have all documents submitted by Friday of Next week.

Final Entry: This journal is being added to the document today

Final State of My Requirements (All Completed)

- ❖ Requirement 1: Play Game/Movement
 - (Completed 100%)
- ❖ Requirement 2: Save Game
 - (Completed 100%)
- ❖ Requirement 3: Load Game
 - (Completed 100%)
- ❖ Requirement 4: Exit Game
 - (Completed 100%)
- ❖ Requirement 5: Modify Game Settings
 - (Completed 100%)
- ❖ Requirement 6: Completed Prologue level
 - (100%)
- ❖ Requirement 7: Interact with in-game items/systems
 - (Completed 100%)
- ❖ Requirement 8: VR Setup/Interaction
 - (100%)

Final tasks for May will be listed on Asana, an online project management site where for the past few months I have been assigning myself weekly or monthly tasks

➤ <https://app.asana.com/0/89556677008611/list>