SherwinB_LoganM_CS302_RD_Report

by Sherwin Bautista

Submission date: 22-Nov-2023 11:23AM (UTC+1300)
Submission ID: 2235561739
File name: LoganMcConnel_SherwinBautista_CS302_RD_Report.docx (8.84M)
Word count: 3016
Character count: 15558

CS302 R&D Report

Sherwin Bautista & Logan McConnel

Department of Software Engineering, Yoobee Colleges, Christchurch Campus

BSE:302.4: Investigative Studio

Professor Ranju Raveendran

30 October 2023

Introduction

In this report, we will cover the development process of the Astro Funk MVP, challenges faced along the way, as well as going over the development schedule and what was completed.

Most of the development was carried out over the course of 7 weeks, with breaks of one to two days for managing mental health being common. On top of this, we focused the first week of the term for planning our work, communicating with our professor, and preparing resources to begin development. This allowed for us to complete the bulk of heavier work in the first few weeks, and use the last two weeks or so as time to go over bug fixing, merging GitHub branches, and writing the report and presentation.

Research Outcomes

Logan

My research question from CS302.1 was partially answered during development. Whilst doing level design, I worked with a friend to test the level, to get a better understanding of what was a good idea, what was a bad idea, what worked and what was unnecessary to player enjoyment and engagement. This fulfilled the core development target of integrated level design, as having someone to test the level and provide feedback allowed for decisions to be made that better unified the games mechanics, and the level.

Integrating user testing, whether that be in the form of internal QA (quality assurance) testing, or having open alpha builds and beta builds, would allow for a more refined process when integrating game mechanics and game world design. Allowing for QA testers with a large body of experience to inform initial decision making, along with a level design team, would likely expedite the initial design phase. From there, open alphas and betas would be ideal to get a larger corpus of knowledge surrounding design decisions, as well as allowing the QA team to focus on bug finding and fixing.

To further extrapolate this to integration of story mechanics and game mechanics, what I have learnt from the level design and testing process has given me an idea of how to better design a narrative to integrate with the game's mechanics. By having narrative designers and programmers working together, they will be able to provide details on what will and wont work, as well as have open and rapid communication with one another to better resolve design conflicts before they get too large to fix before deadlines are reached.

Project Summary

To provide some background on the project, this was originally a third-person project developed in the last term of year two, which was reimagined as a first-person project for our capstone project last semester. The original concept for the game has been in various states of development for almost a decade.

The story concept for the game follows the main character, Zero, across a cyberpunk/dystopian version of Ōtautahi/Christchurch. This setting was chosen as it better allowed for themes to be explored such as personal growth. This setting was also something players often express interest in, which would help to draw players in, and allow them to become immersed in the game.

The core function of the game is movement. The main character, Zero, is hyper-mobile. Able to climb large walls, run extremely fast, jump higher and farther than most, as well as run on vertical surfaces, cover short distances, and change direction with ease. This form of movement, mixed with complex environments would allow for player engagement and expression, as they would be choosing their own path through a level, as well as learning and mastering the skills and mechanics of the game.

Development Schedule

Week 8 – Assigned workloads to team members, and development began.

Logan – Began working on building level, with intentions of testing it with original MVP character movement.

This level design, from a basic standpoint was simply used as a baseline to gauge how players would interact with it, as well as providing a learning opportunity to better design levels that would work

with the player characters abilities.

Sherwin – Began reworking MVP first person mechanics.

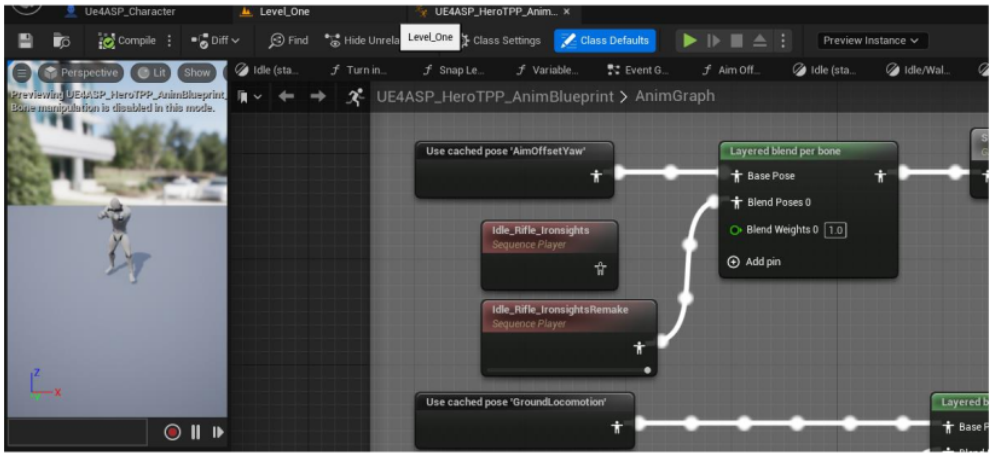
Week 9 – Level design and FPS mechanics redesign continues.

Logan – Level design mostly complete, testers chosen.

Adding in more verticality to the level, to better utilise the main characters movement abilities was identified as a key focus point for level design.

Sherwin – Fixing aiming animations.

There were issues with the starter animation ‘Idle_Rifle_Ironsights’ making character aim unstable and inaccurate. A work around for this problem was to create a new animation ‘Idle-Rifle_IronsightsRemake’ with only 1 keyframe with the pose to replicate the character being frozen in place.

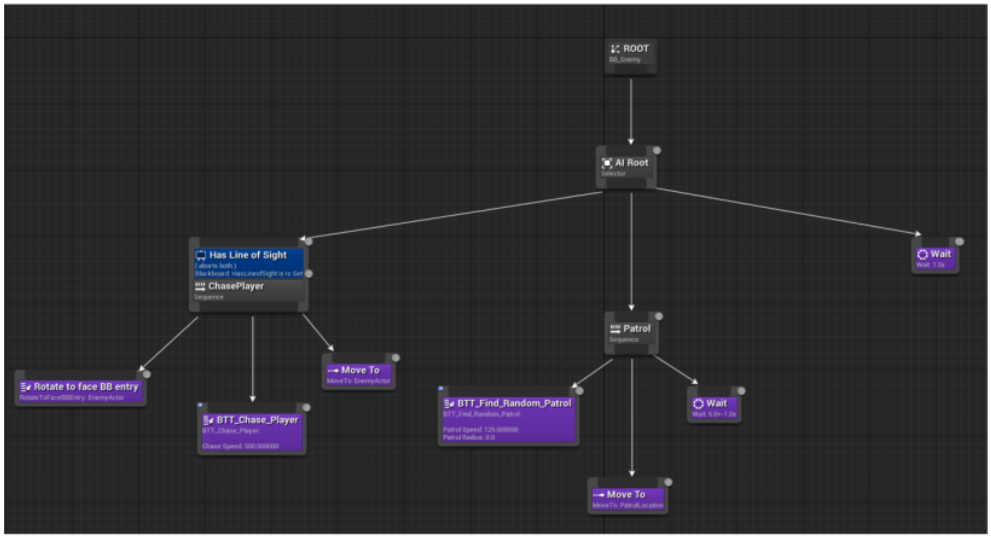


Week 10 – Level testing begins.

Logan – Switched development focus to AI and NPC work after the mid-point of the week.

Sherwin – Research weapon implementation videos for switching weapons. Mental health break.

Logan – Began working on character blueprints and event graphs, creating and naming variables. Some struggles with animations and character meshes.



Here is the first iteration of the NPC (non-player character) behaviour tree. This determines how the NPC will interact with the environment, such as chasing the player once visible, and patrolling the area when the player is not nearby. This was changed later in development. This version of the behaviour tree uses sequence nodes, which operate in a sequence from top to bottom and left to right. This is effective for simpler AI but can be limiting when attempting to implement more complex AI systems.

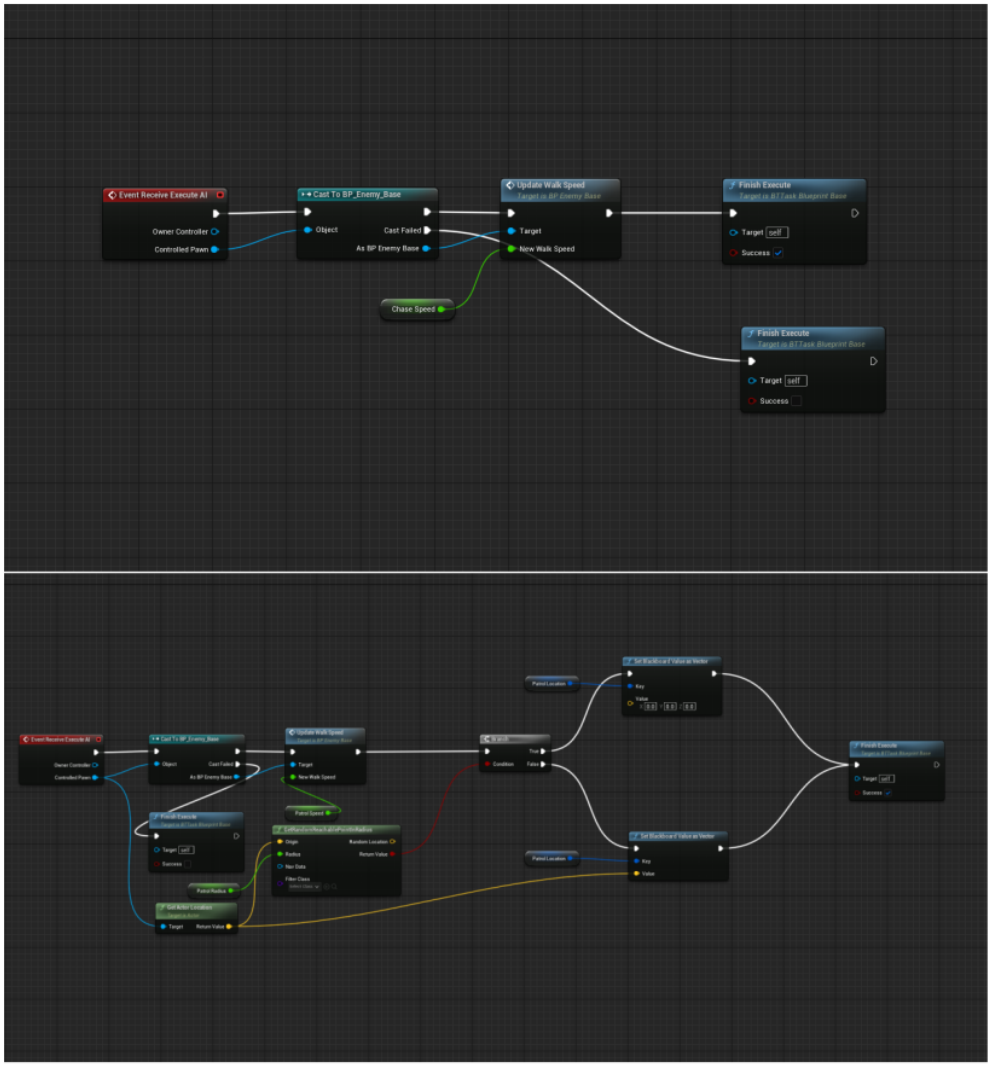
Sherwin – UI and Widget elements added – created crosshair for weapons, ammo count, reserve ammo.

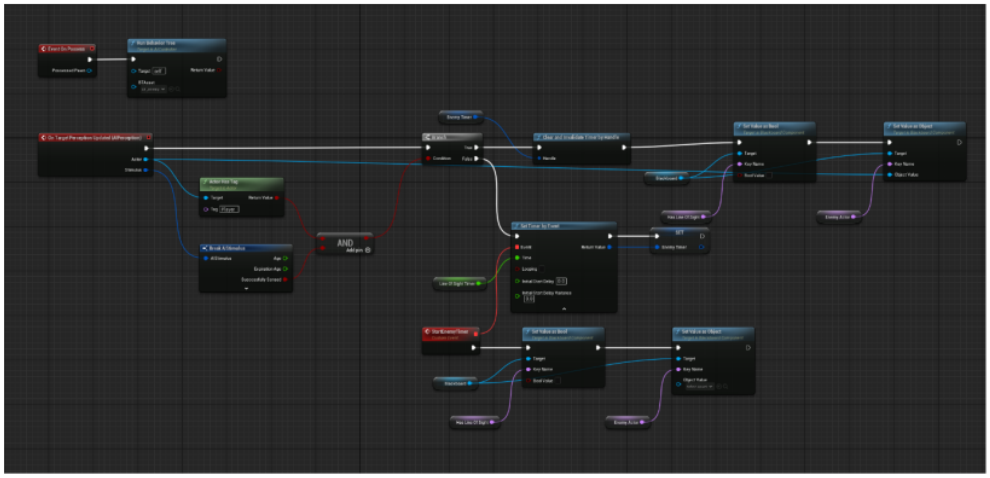
In the image below you can see the ammo count is displayed as a 3d element attached to the weapon. This gives a futuristic look to all the weapons implemented.

Week 12 – Mental health break.

Week 13 – AI Logic implementation completed.

Logan – Blueprints, behaviour trees and event graphs all completed, to begin work on animation.

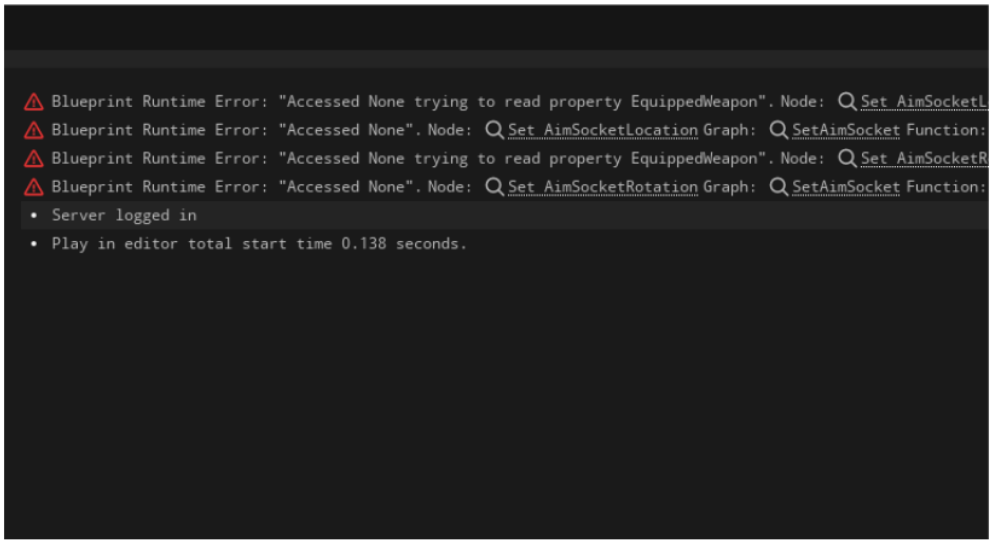

Event Receive Execute AI
Owner Controller
Controlled Pawn
Cast To BP_Enemy_Base
Object
Cast Failed
As BP Enemy Base
Update Walk Speed
Target is BP Enemy Base
Target
New Walk Speed
Chase Speed
Finish Execute
Target is BT Task Blueprint Base
Target self
Success
Finish Execute
Target is BT Task Blueprint Base
Target self
Success



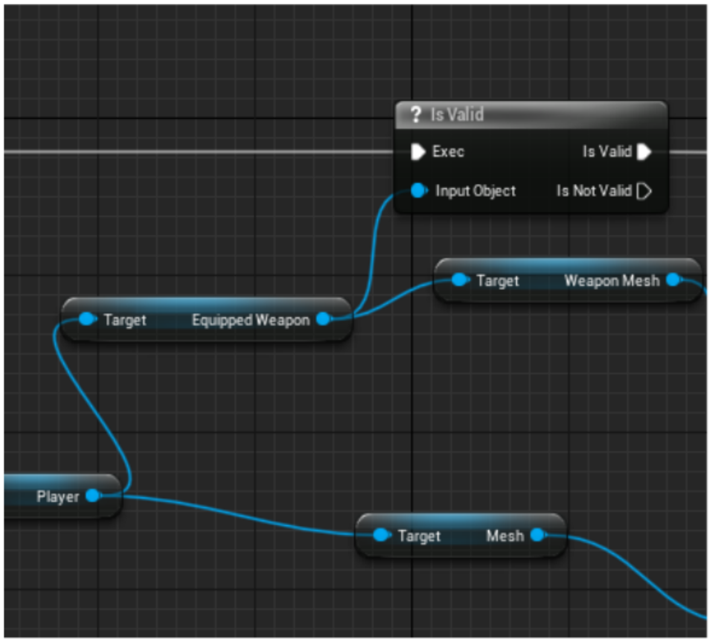
These are the event graphs and the blueprints that are called by the behaviour tree, using selector nodes and sequence nodes, once the game state is active. These determine what senses the NPC has, the detection range of those senses, as well as functions to call the Actor Pawn for the main character to be able to differentiate between other NPCs and the player.

Sherwin – bug fixes for compile error ‘accessed none’ and secondary weapon shell ejection not working.

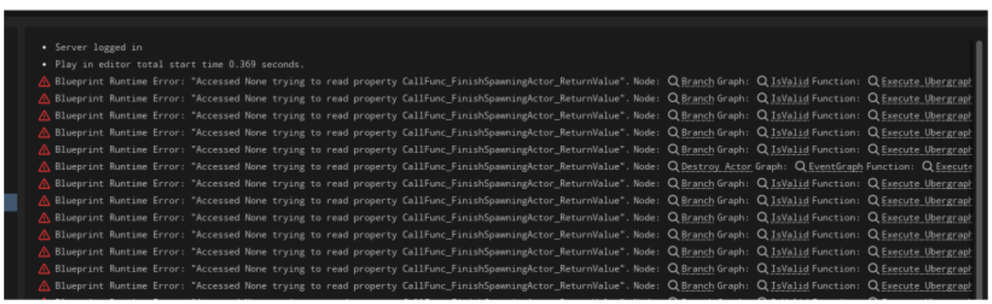
The error says that the variable ‘EquippedWeapon’ has no data inside to read from it. This is caused by executing the function using the variable before any data is assigned to it.



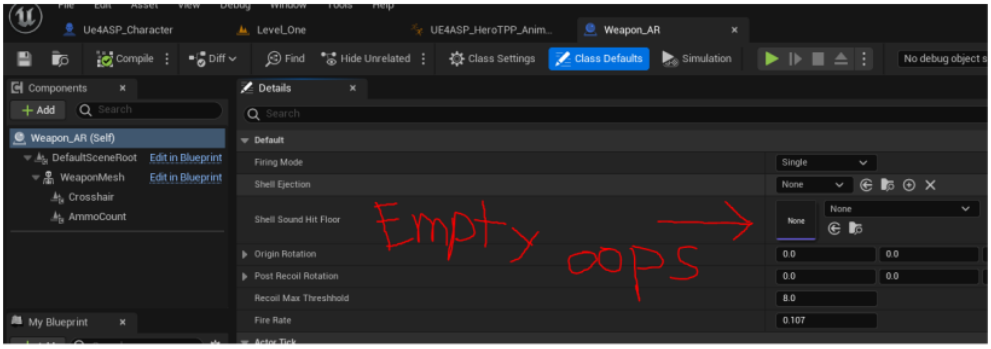
The solution was to pass through an 'is Valid' node to check if there is data in the 'Equipped Weapon' variable before proceeding. This solves the issue of the function executing before there is any data assigned.

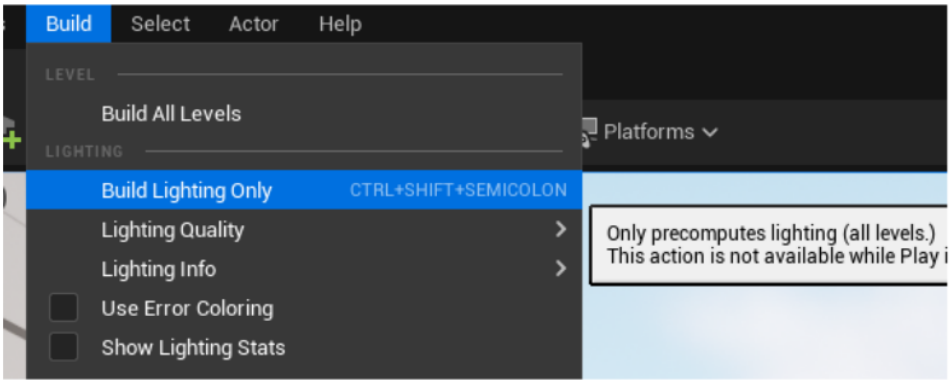


The same issue is seen here with firing the secondary weapon. Calling the 'FinishSpawningActor' returns a value of none.



I have fixed the issue by selecting a shell ejection blueprint for the secondary weapon.

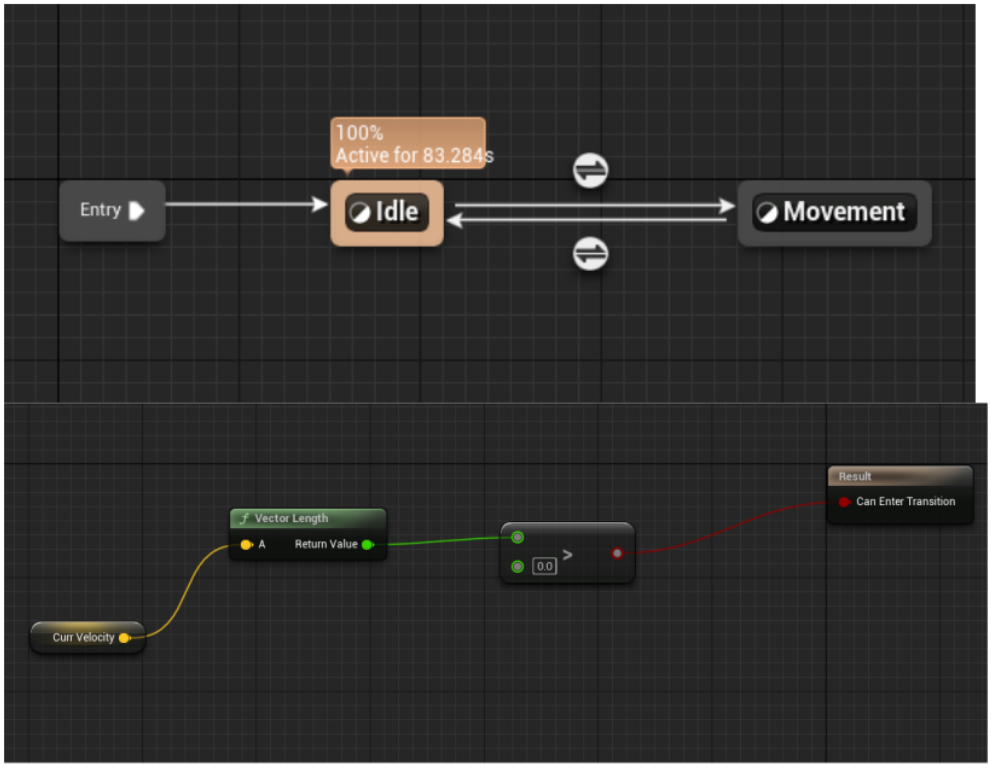




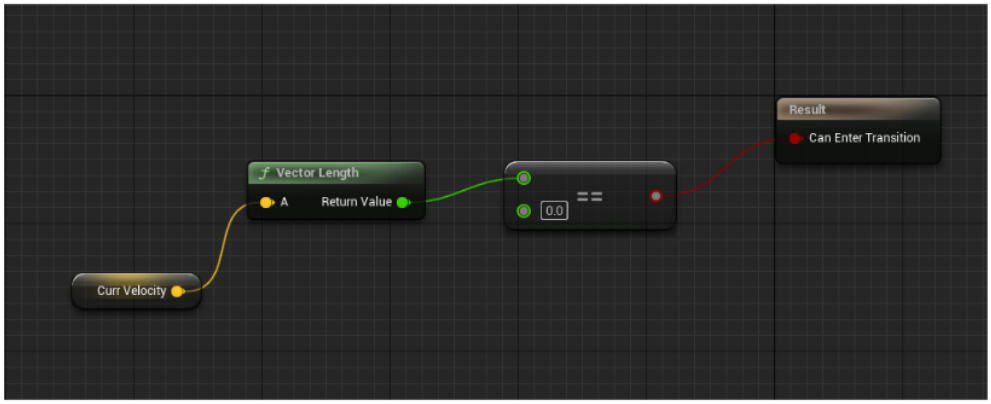
A quick fix for the 'Lighting needs to be rebuilt' error was to go into the build tab and automatically rebuilt all the lighting objects and after that has finished compiling, the error should disappear.

Week 14 – NPC Animation work and AI testing.

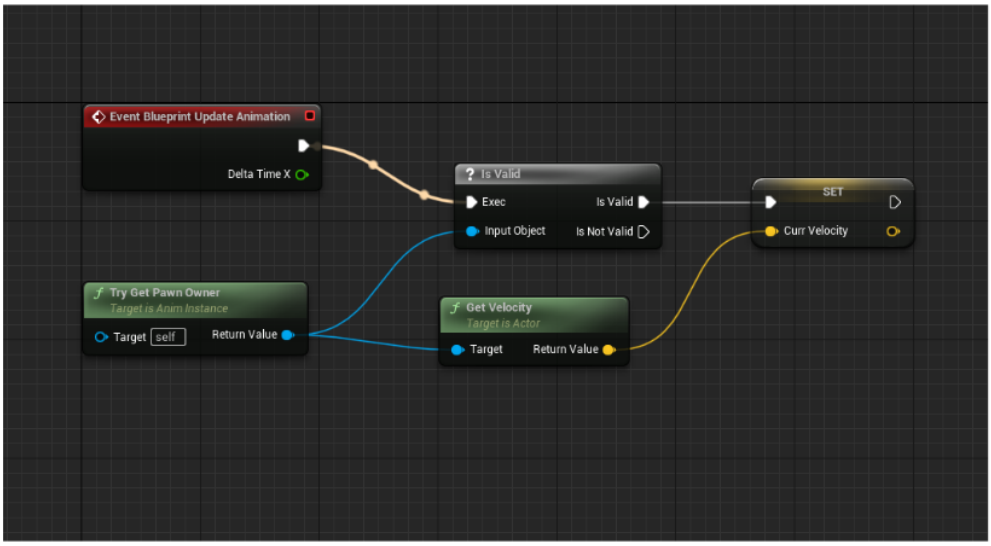
Logan – Worked on animation, using skeleton meshes with animation blueprints to target animations for the NPC. Then created animation graphs and an event graph to call specific animations depending on what the NPC controller pawn was doing.


100%
Active for 83.284s
Entry
Idle
Movement

Result
Can Enter Transition
Vector Length
A
Return Value
0.0
>
Curr Velocity


Result
Can Enter Transition
Vector Length
A
Return Value
0.0
==
Curr Velocity

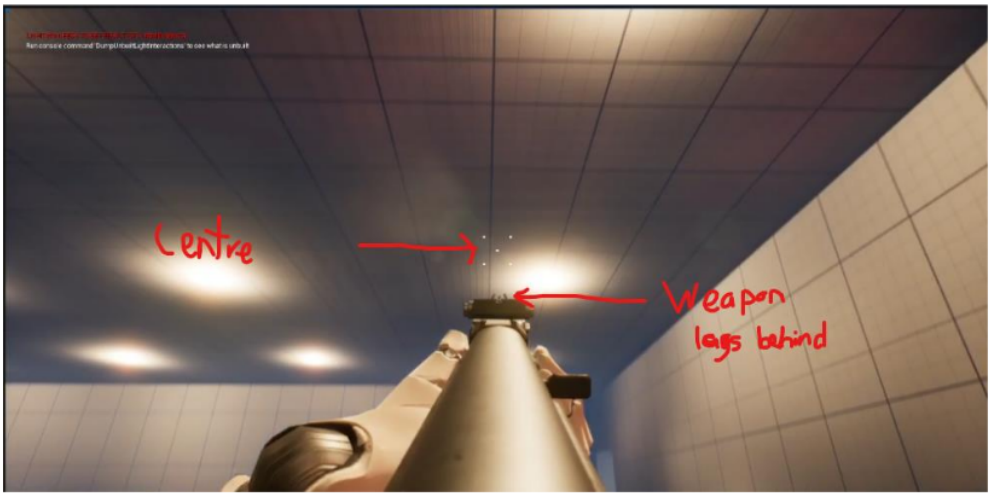
These two blueprints reference the NPCs movement speed, to determine whether the running animation should be played or not.



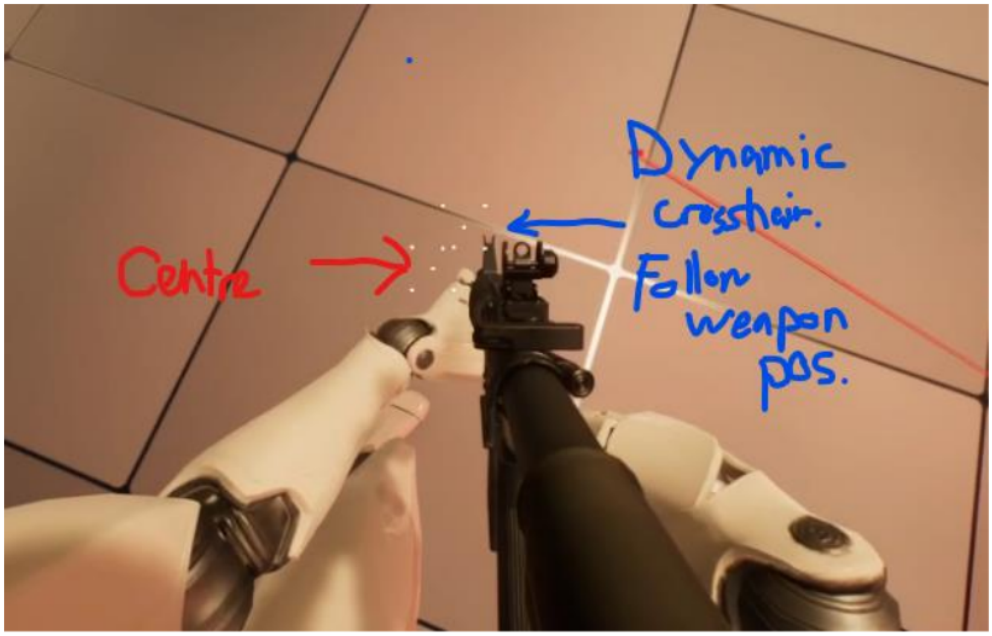
This event graph is checking whether the NPC has seen the player, thus calling for an animation update, using the above blueprints, which will change the character animation to fit the action.

Sherwin - Bug fixes crosshair and aiming accuracy.

Having the crosshair widget fixed to the center of the screen comes with issues when the players yaw is not level. In the image below you can see the weapon lags behind the centre crosshair.

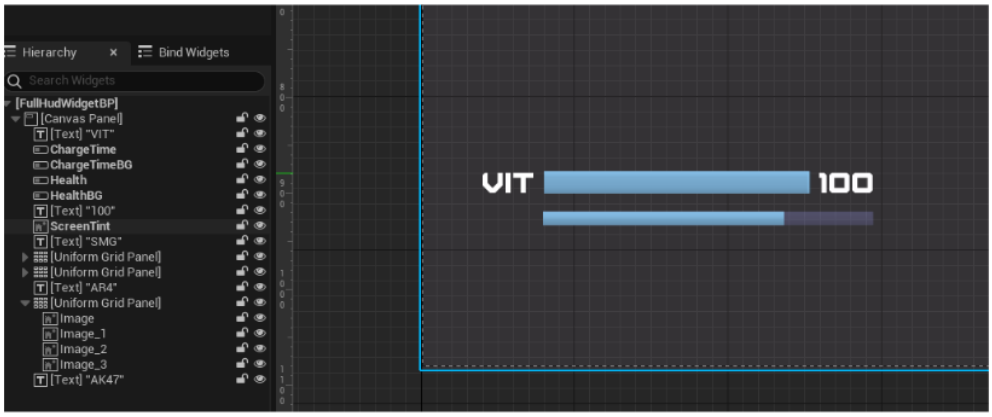


The solution was to attach the crosshair to the gun mesh itself so It does not have the issue with crosshair lagging behind.



The weapon has 3d widgets attached to the gun mesh to fit the theme of futuristic weapon. The Blue widget shows the ammo the gun has, and the red widget is the current firing mode. Another detail is the currently equipped weapon is highlighted on the bottom right of the screen and whenever the player changes weapons it switches the opacity of the text to indicate to the player that it is selected.

The top bar is displaying the health of the player, and the bottom represents the stamina the player has which allows the character to sprint, parkour, and dash.

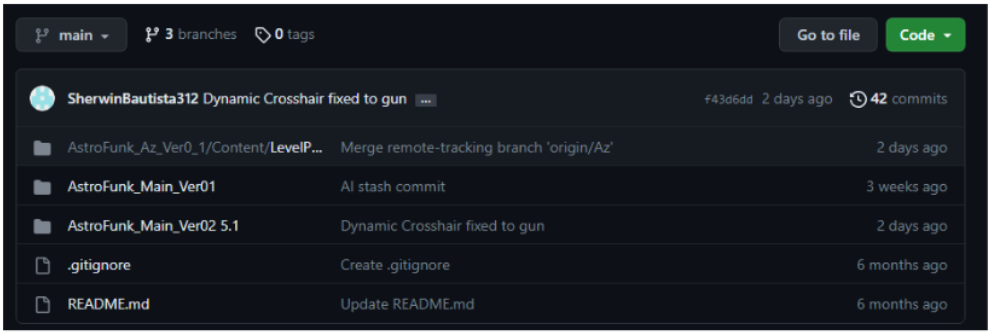


The current full HUD display is shown below with all the widget elements together.

Week 15 – Presentation work begins.

Logan – Began working on presentation, and finalising report, whilst co-ordinating with Sherwin and Ranju on any small issues still needing to be completed in the final weeks of development.

Sherwin – merged branches to main branch.



Week 16 – Final week, presentation is given, and work handed in.

Development Challenges and Successes

Development Successes

Once the right information was found regarding a topic of development, Unreal Engine is designed in such a way that it makes development simple and straightforward.

Logan

Knowledge and Proficiency – This 16-week project has had many hurdles, which have offered great learning opportunities for problem solving in a development cycle. Also learning to swallow my ego and readily accept that questions need to be asked of those with more knowledge has been beneficial to my learning.

Additionally, Unreal and Epic have great online resources in the form of training videos, and an entire website dedicated to documenting how the various versions of Unreal, from 4.27 through to 5.3 work. Being able to open a web doc that had easy to follow information and simple diagrams was much easier to follow than a video, due to being able to move at my own pace.

Player Psychology – Over the course of level design and development, I was able to work with closely with some one as they tested the level I had designed. This was insightful, as it offered a look into how players will interact with the level around them, and how they are likely to use a character's abilities to traverse the environment. This offered an opportunity for note taking, to better allow for level design decisions in the future, as well as developing a work process to test levels with quality assurance in the future.

Modularity – Modularity of the designed blueprints was important. This is in part due to potentially needing to go back later and add more functionality in future, as well as allowing for faster development. The best example is NPC development. Creating basic blueprints and basic animation blueprints allowed for easy AI testing; however, this also meant that any changes in blueprints were

easy to complete, with minimal concern of creating bugs. Furthermore, developing in this way meant creates bases for more complex blueprints to be created from, streamlining the animation and NPC development cycle, since its essentially plug and play, requiring only small changes to create new characters and associated animations.

Sherwin

Knowledge – Throughout the course of this project, a significant milestone was achieved in my development journey: gaining invaluable experience and enhancing my expertise with Unreal Engine 5. While researching methods to implement the core functions, I extensively tapped into YouTube's wealth of tutorials and guides. These resources provided many practical ways to implement the same functionality, which offers a dynamic learning experience.

Moreover, the Unreal Engine forums and threads emerged as invaluable assets in gaining knowledge. Engaging with a vibrant community of developers, troubleshooting challenges, and exchanging insights bolstered my understanding and supported a collaborative environment to work and learn.

This holistic approach to learning significantly contributed to the success achieved in gaining expertise with this cutting-edge platform.

Development Challenges

Logan

Time – The greatest challenge in the development of a game is the time required to complete various functions. AI, whilst it can be developed by a single person, is complex, and anything more than a basic NPC that follows the player once it detects them would likely require a team of people, with more time, and vastly more knowledge, to be able to create AI that functions in such a way that players will see as believable.

Mental Health – A challenge of development was managing mental health. This was brought on by having exacting standards that are likely too high to meet in such a short time frame. Failing to reach to those standards, as well as the pressure from approaching deadlines put stress on my mental health that took a lot of work to manage. Luckily, I have worked in high stress environments before, but developing a game has additional challenges, such as personal development targets that must be set. Failing to reach these goals often has the compounding affect of adding stress, which makes working on catching up or switching to a new development target even more difficult.

From this, I learned that, much like Sherwin, maintaining a healthy balance between working and relaxing is important, so as to be able to maintain a productive schedule without burning out in a few weeks or months.

Sherwin

Burn-out - During the project's course, I encountered a formidable challenge that tested not just my technical skills but also my resilience: grappling with work burn-out and its impact on my mental well-being. As deadlines continue to creep closer and complexities mounted, the pressure to deliver a fully realised project took a toll. Balancing my passion for the project with also needing time for self-care became increasingly difficult.

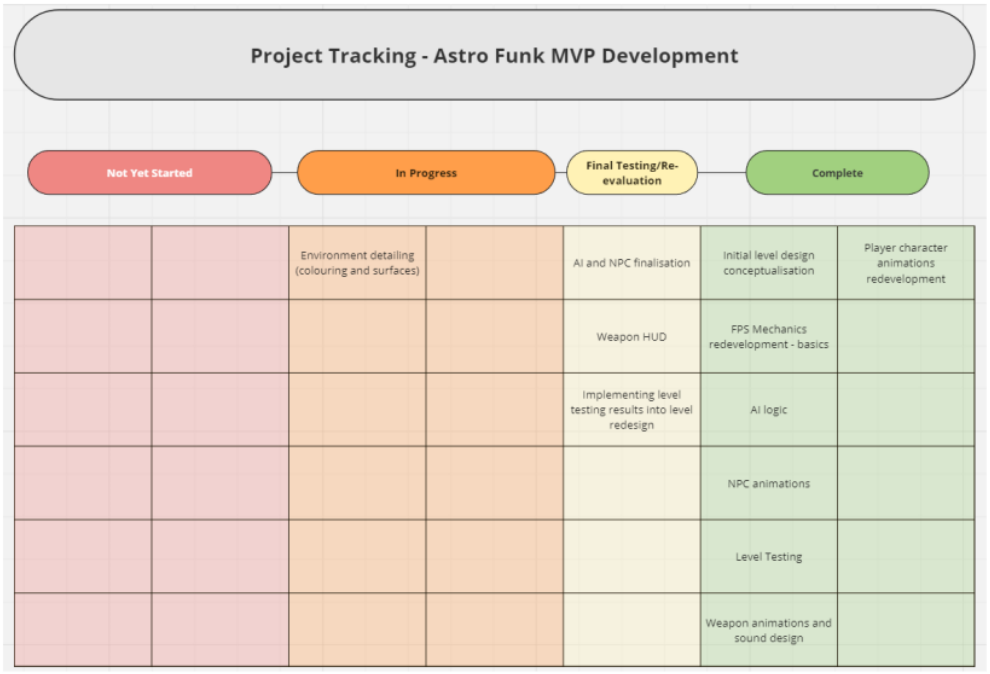
Recognising the importance of mental health, I prioritised self-care strategies, such as regular breaks, exercise, and fostering a supportive environment for both Logan and me. This experience underscored the significance of maintaining a healthy work-life balance and highlighted the imperative nature of self-care in the realm of demanding creative endeavours.

Project Management

For managing the project, we kept constant contact with each other using Discord, a messaging application available on PC, Mac, and on mobile devices. Messages were exchanged regularly, either every day or every other day, with updates on progress, workflow situations and if there were any other commitments that would take our time away from development.

We kept weekly contact with Ranju, our professor, through scrum meetings on Teams, as well as occasional messages on Teams when we got stuck. We also kept attendance of these scrum meetings, as well as contact if one couldn’t attend one.

For project timeline management and tracking, we used Trello. This allowed us to track what we had already worked on, what was still to be completed, as well as tracking progress on development targets, relative to the amount of development completed.



Changes in Development

The projects scope was scaled back after the first couple weeks of development, as the time allowed for the project was not enough to be able to complete the original tasks of development. This meant consigning NPC interaction to a potential development target, rather than essential. Instead, we decided to focus more on the core functions of the game, and better implement them compared to the project we had created in CS301.

Compared to the state of the project at the end of CS301, the first level, main menu, and NPCs have been implemented, as well as the core functionality of the game, being the games base mechanics of movement and weapon usage, were redeveloped and improved upon. This included better animations for the main character when firing, as well as improving the aim. The newly implemented parts of the project were core to moving from simply a proof of concept to an actual MVP that can be used as the basis for a future game, with the help of a team and more time.

Conclusion

Logan

This project has been satisfying to work on and get to a level that can be improved and iterated upon to create a game in future. The improvements to my working practices, such as uploading to GitHub regularly to keep track of work and using a track sheet have improved my ability

to complete work at an efficient pace. On top of this, there is a certain sense of pride from looking at what has been created over the course of this project, and the work that was involved.

In addition to this, being presented with an opportunity to see how a player interacts with a game world, allowed me to use that knowledge to better design an engaging game, that made more effective use of the game's mechanics. It also proved helpful in providing answers to my research question from CS302.1. Allowing a group of players to test a game, each with varying stand points and ideas of how to play games/interact with them, will allow for more information to be gathered, which can be used to better enhance their immersion in later builds of the game.

Sherwin

In navigating the multifaceted landscape of development, the journey with Unreal Engine 5 has been a testament to the intertwined nature of triumphs and challenges. From the great efforts in mastering the engine through diverse resources like YouTube tutorials, forums, and official documentation to the introspective battle against work burn-out and mental well-being, this project unveiled the intricate facets of professional growth. Each success, whether technical expertise gained or the acknowledgment of the vital role of self-care, has contributed to a holistic understanding of the development process. It's not solely about mastering tools but also about honing resilience and embracing the symbiotic relationship between technical prowess and personal well-being. This journey underscores that true success in development lies not just in achieving technical milestones but in embracing a mindset that values both professional growth and individual flourishing.

Date and signature:  22/11/2023	Please check with the marking rubrics to make sure you are meeting all the requirements.

SherwinB_LoganM_CS302_RD_Report

ORIGINALITY REPORT

0%	0%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to North Shore International Academy Student Paper	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		