



# PORTFOLIO DES310

STUDENT: MIGUEL  
ANASTÁCIO

2002268

# TABLE OF CONTENTS

<b>Development Diary</b>	3
<b>Portfolio</b>	6
• Sprint 1	6
• Sprint 2	7
• Sprint 3	9
• Sprint 4	11
• Sprint 5	13
• Sprint 6	16
• Sprint 7	20
<b>Reflective Essay</b>	21

# DEVELOPMENT DIARY

## SPRINT 1

- Reviewed assigned brief with team
- We chose to work with Unreal engine
- Was assigned the role of UI/Gameplay programmer.
- Researched how to implement systems for an RPG and how UI works in Unreal.
- Got familiarised with some of Unreal Engine Systems

## SPRINT 2

- Set up source control
- Created events UI and connected it to a random events system
- Added planets actors and interaction with player
- Implemented an inventory system and its UI
- Set up UI interactions for the store
- Implemented a quest system and its UI
- Added audio to UI and a more structured planet screen
- Merged Liam's initial procedural route with the other systems.
- Created a basic HUD.
- Learned about delegates in Unreal, so refactored existing code to use delegates.

## SPRINT 3

- Implemented UI screens for route selection and main menu.
- Added UI for when player completes a quest
- Expanded events and its UI
- Incorporated stats made by Owen, made inventory interact with stats.

# DEVELOPMENT DIARY

## **SPRINT 4**

- UI and transition to checkpoint
- Changed the UX and UI of the store based on the feedback from the play party.
- UI animations for panels

## **SPRINT 5**

- Made combat more dynamic and impacted by the stats of the player and the enemy.
- Implemented the character selection screen.
- Made UI for combat and the transitions to and from combat.
- The team had a great meeting with our mentor Gordon where we went over the main problems with the UI/UX and what we should do to turn what we had into a cohesive experience.
- Implemented a Game Over and its respective UI. Imported more events.
- Incorporated Liam's combat abilities into our HUD.
- Separated the Main Menu into its own level.
- Revamped the route selection screen based on feedback.

# DEVELOPMENT DIARY

## **SPRINT 6**

- Changed UX/UI of abilities based on feedback.
- Added a cooldown for events and combat to help with pacing issues in the game.
- Added more information to UI, made routes be distinct in terms of events and combat.
- UI polishing and bug fixing.
- Changed the text to use unreal “Rich Text” so that we could stylize text easily.
- Incorporated Minigame UI into HUD
- Revamped the character selection screen in conjunction with other members of the team.
- Improved upon the minigame developed by Liam.
- Decoupled store from planets
- Polished the Store UX/UI

## **SPRINT 7**

- Worked with Liam and Owen to finalize the Save/Load system.
- Changed combat end transition based on feedback.
- Continued polishing and bug fixing.
- Helped making sure the final build of the game was stable and high quality.

# PORTFOLIO – SPRINT 1

In the first sprint, the direction and flow of our game was not well defined. But we knew that we were making an idle RPG and that we were going to be using the Unreal Engine, since most of our team had experience with it. Also, during this sprint, the workload was divided among the programmers, and besides gameplay I was assigned with implementing the UI.

Initially I started by familiarising myself with the systems and workflow of the Unreal engine. Afterwards, I researched how to implement certain features that we knew would have to be present in our game such as an inventory component and a quest system. I also researched the basics of UMG, the tool for UI inside the Unreal Engine, since I had never used it before.

In terms of development, I started by implementing some basic UI interactions between the player and the world.





# PORTFOLIO – SPRINT 2

We only ended up setting up source control in the beginning of the second sprint. This happened because initially we intended to use Perforce and spent a week setting it up. Just to give up on it because Perforce software was not installed in the university computers. So, we settled instead for Github. I was the one responsible for setting it up.

In this sprint, the idea/direction of our game was more concrete. Thus, I could start implementing the systems mentioned in the previous sprint with more context. We decided that our game would take place in space, I created some basic planet actors, using Unreal's starter content, to test out some of the features.

During this sprint, I implemented an inventory system using an actor component so that it could be used by the player and the store. Another component added to the game was a random events component. This is responsible for making events popup while the player is moving through space. Initially, this actor component was attached to the player. I also created a quest that the player could select and that it would be completed when he reached another planet. A very crude UI was created to support these features and present the necessary information to the player.

To implement the UI for both mechanics I researched about the topic of data driven UI since the contents of the inventory relied on structures of data such as items that the UI would have to support and present the relevant information.

# PORTFOLIO – SPRINT 2

Towards the end of the sprint, Liam and I worked together to merge what we had worked on. This was an important step, since Liam had been working on the first iteration of the route system. This system is a procedurally generated route that acts as the backbone of our game. Hence it was extremely important that all the other features worked well with it.

Additionally, regarding the UI, I created a basic HUD with very little information just to program the behaviour of its visibility when it reached a planet. The first draft of the planet screen was also created using a first draft our UI designer (Iris) so that the quest and store functionality could be tested.

At the end of the sprint, I figured out how to use delegates. This unreal feature, was very useful since it allowed us to link our game loop that was implemented in c++ with the UI that was implemented using blueprints. I used this to refactor the random events system code so that it would be cleaner and more efficient.

The gif on the right shows the systems mentioned in this sprint.





# PORTFOLIO – SPRINT 3

This was the sprint before the first playable, hence I focused on implementing the functionality of menus that the game was lacking such as a main menu and the UI for when the player could select the route.

For this screen I used a delegate since initially we wanted the player to be able to click on the possible paths to see the characteristics of each, the delegate was useful to bind functionality that was already in c++ to the creation of the UI that was implemented in blueprint

Another UI element that was missing was some sort of feedback when the player would complete a quest. This was implemented during this sprint and was just a message that would get displayed in the planet screen when the player had completed a quest by travelling to that planet.

On the route system I collaborated with Liam to address some of its imperfections, namely preventing the same planet actor of repeating itself in the same route and the fact that you could see the other elements of the route when you were orbiting a planet.

# PORTFOLIO – SPRINT 3

At this point, the direction and content of the events system also became clearer. The designers set up events that needed an outcome depending on the option chosen by the player. To accomplish this, I restructured the way events were currently being handled and the data structures behind the events to their final version. Each event is a child class of a UObject and it contains an array of Event Options another child class of UObject. Each Event Option has an outcome and rewards/effects associated with it. Naturally, the UI flow of the event mechanic was also changed, this is represented by the diagram below.



Towards the end of the sprint, I incorporated into the game a stats system developed by Owen. The stats were added to the HUD UI and functionality was added to make the inventory and the stats worked together. To accomplish this, I also expanded upon the stats system to be able to account for the modifications caused by the items. The team envisioned items has applying a multiplier to the base stats of the player. In total a player can have 4 items each of them impacts a different base stat. This base stat is controlled by the level of the player.

# PORTFOLIO – SPRINT 4

At this point the structure and most of the functionality of the route system was almost complete. One of the main features missing from the original design was a checkpoint stage along the route. This would always be present in the long route and it would be a space station instead of a planet. This place was different from the planet, so it needed a different screen. The necessary UI functionality and link to the route system was developed and implemented during this sprint.

At the end of the sprint 3 we had our first play party where we noticed that most people got confused with the UX/UI of the store. So, I made some changes to make it more clear and provide more feedback to the player.

# PORTFOLIO – SPRINT 4

Another UI improvement that I made in collaboration with Damian and Iris was adding slide animation to our events. This was a small change and quite simple to do but we thought it added quite a bit of quality to the overall experience. The two gifs illustrate the difference.



# PORTFOLIO – SPRINT 5

In sprint 5 I focused on implementing a character selection screen since at this point our artist Jake had made different spaceships and we wanted them to be in game for the alpha that was coming up. The structure of the UI was a bit crude, but the most important thing was that the functionality was implemented. This also forced me to adapt the player class since we intended for each spaceship to feel different in terms of gameplay, so they had different stats. The stats component was also adapted to support this. Each spaceship was also associated with a different faction and the UI was supposed to change based on the faction selected by the player, thus I expanded the player class to also account for this.

During this sprint I also separated the main menu into its own level so that Jake could create an aesthetically pleasing scene as the background for our main menu.

At this point our game was missing a significant detail, a game over condition. Therefore, I implemented one and added a game over UI screen to be displayed. Other small things such as, new events or items were also added into the game during this sprint.



# PORTFOLIO – SPRINT 5

The other feature of our game that I focused on was the combat, both in terms of functionality and UI. The combat was originally implemented by Owen, I expanded it by making it more dynamic. To achieve this, I made the enemy actually fight back, and I also incorporated the stats into the combat. 4 stats present in the stats component were used:

- Hull Integrity
- Shield
- Attack Power
- Speed

The hull integrity and the shield defined the HP of the player, the attack power influenced the damage that the player did per attack while the speed increased the chance of the player avoiding taking damage when he was hit by the enemy attack.

To complement this the UI was expanded, health bars were created for the enemy and the player. I also added a UI panel for abilities during combat that were implemented by Liam.

The transition to and from combat felt a bit jarring so I added UI panels to notify the player when combat was about to start and after it ended to congratulate the player. These two screens used the same sort of sliding animation mentioned in the previous sprint.



# PORTFOLIO – SPRINT 5

The last main development of this sprint was a meeting that the team had with our mentor Gordon where he gave us a lot of feedback to help turn our game into an actual usable and cohesive experience. Most of the issues concerned the UX/UI of the game. To help with this we revamped the route selection screen. From now on the player would select a route through a button, instead of clicking on the path itself, and the route selected would be highlighted. To support this, I changed the UI functionality and its connection with the route system.

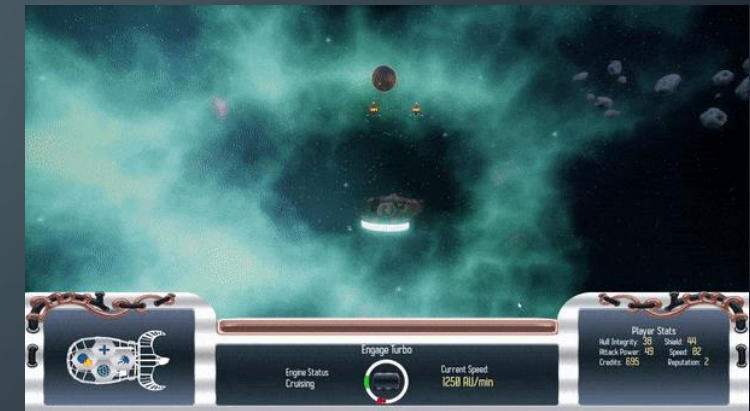
Below you can see a comparison between the old version and the version at this point.



# PORTFOLIO – SPRINT 6

Sprint 6 started after the alpha milestone, so we had a lot of useful feedback to work from both from our clients and from the play party.

Based on this feedback, I changed the UX/UI of the abilities in combat to be more consistent with the rest of the UI and to provide a more pleasant experience. The previous version required a lot of clicking. The gifs on the right show the two versions.



Our game also suffered from pacing issues, the main issue being sometimes combat and events would fire in a very short period of time, so I implemented a cooldown to prevent this from happening.

Another major piece of feedback that we got was the lack of player agency while the spaceship was moving along the route. Thus, the team decided to implement a minigame that would change the player speed. Liam implemented this and I incorporated it into our HUD.

# PORTFOLIO – SPRINT 6

Throughout the sprint I also focused on fixing small existing bugs and polishing the UI. I added more information to the several UI panels existing in our game.

At this point the text blocks present in our game were very hard to read and they did not provide the user with any sort of indication of what information was actually important. To fix this I changed the text to use Unreal's Rich Text feature. This was quite useful and allowed us to maintain a coherent and understandable text style throughout our UI, we used it for titles and to highlight important information in the blocks of text.

The character selection screen was also restructured during this sprint in collaboration with Damian and Iris. Mainly based on the feedback provided by our mentor Gordon.

I also polished significantly the UX/UI of the store during this sprint. I added more feedback to the user through the use of messages and different sound cues when the player bought an item or was unable to due to not having enough credits or already having the same item.



# PORTFOLIO – SPRINT 6

In the spirit of polishing the store, the code structure behind of it was revamped to be more efficient and versatile. Before, each planet was associated with it a vendor actor that held the store inventory. This was set in the beginning but at this stage of development didn't make much sense to keep it this way since it took up more memory and would also require more work for the save/load system that was being implemented by Owen. Thus, I removed the vendor actor from the planets and added a single one to the level. This vendor now had access to all the items in the game and when the player reached a planet the inventory would be randomized with certain restrictions: there had to be between 1-2 items per category.

This allowed Owen to not have to worry about saving the inventory of the planet since in our game the player can only save when he is at a checkpoint.

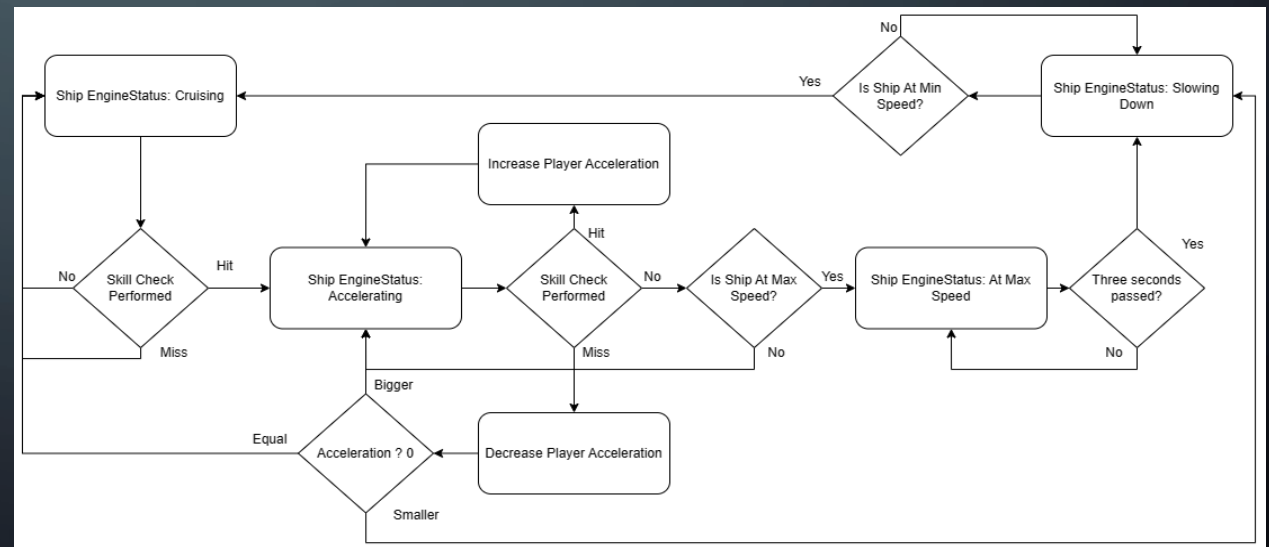
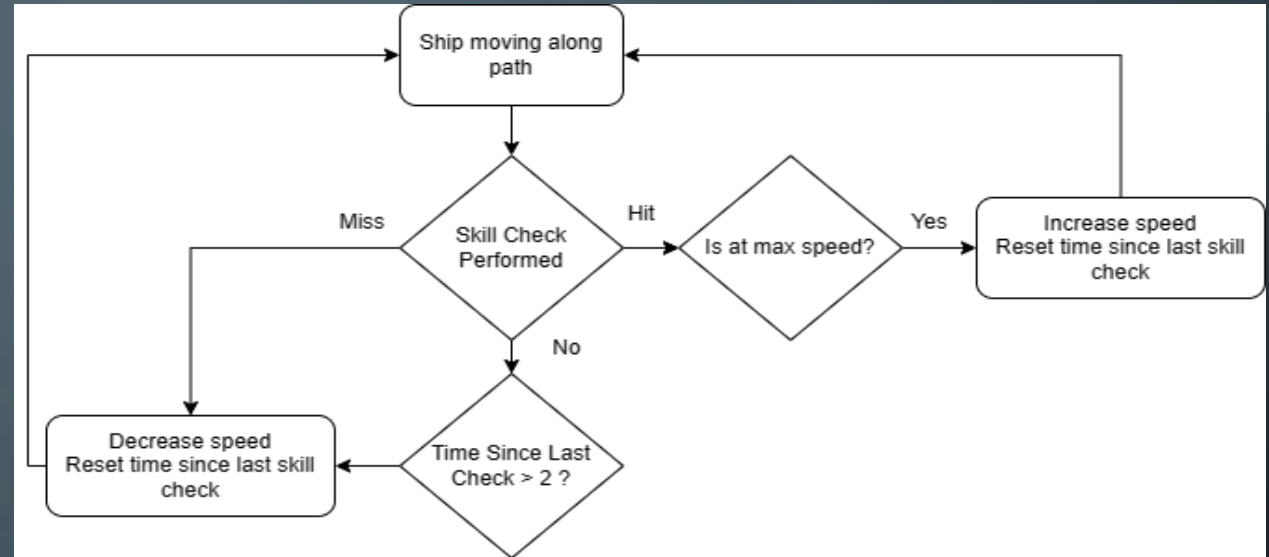
The quest code was also implemented in a similar way to the store, so I refactored it to work like the new store. This was done to simplify the saving/loading system and because it did not make sense for each quest to be associated with a planet in the context of our game. New quest were added as well, and a function to randomize a new quest whenever the player completes one.

# PORTFOLIO – SPRINT 6

At the end of the sprint, I expanded upon the minigame mechanic mentioned before. Liam had set up the minigame to work like the first diagram on the right. I thought that immediately increasing speed looked a bit clunky especially because we were adjusting the field of view based on the speed.

Hence, I changed to instead increasing the acceleration so that the speed increase would occur gradually. My implementation follows the second diagram on the right.

Note, there are 4 ship engine status. In *Cruising* and *AtMaxSpeed* the speed of the spaceship remains constant. In *Accelerating* and *SlowingDown* the speed increases and decreases, respectively.



# PORTFOLIO – SPRINT 7

In the final sprint I focused mostly on bug fixing and polishing. I contributed to finalize the save/load system, by implementing several functions to assist in loading the inventory, quest and the actual player mesh, since our game supports different spaceships. I also helped ensure that the UI and the gameplay experience was consistent when the player would continue from a previous saved game.

The only other relevant gameplay change I made was adjusting the transition when the combat was over, this was done due to feedback from our clients - the transition was too sudden the player needed time to process what had happened. UI wise, I added a simple pop-up panel to notify the player when he levelled up.

In the latter stages of the sprint, I helped the team with the documentation by writing a section of the technical design document, the High-Level Overview of Software Design, and generating and formatting the commit log.



# REFLECTIVE ESSAY

## **Did the project meet the requirements?**

---

Reading the brief back at this point, I would say that the project ended up straying quite a bit from the original idea that the clients had. We respected and kept the original concept and feel behind the brief, but we probably took it in a direction that they were not expecting it. We decided to go with a more sci-fi theme while the original concept was set in a classic medieval fantasy world. This was mainly due to the composition of our team and because we thought that traditional fantasy was a bit boring, sci-fi is also more versatile. For example, it allowed us to incorporate ideas such as Norse and Egyptian mythology into our game.

I think this change in theme was very positive and enabled us to deliver a higher quality project.

Overall, I would say that the project fulfilled the original brief. Most of the features and systems required in the brief were implemented to a high standard and the finished project is a polished and competent prototype.

# REFLECTIVE ESSAY

## What went right?

---

As a team I think we were quite adept at listening to feedback, this was extremely important since it helped us limiting the scope of our game. We were more focused in polishing and improving what we already had than adding sup-par or unnecessary features to our game. Every new feature was subjected to a rigorous scrutiny, it was only added if it would enrich the experience and was possible to implement in the time frame available. For example, the minigame, which was a very late feature, was implemented due to the feedback gathered from the play party and the final version took about a week. It was relatively small feature that really improved gameplay.

Personally, this project allowed me to develop my ability to work with other people from different fields. My ability to communicate with other members of the team was, surprisingly, quite good. Being the UI/Gameplay programmer, I worked in close contact with Damian (lead designer) and Iris (UI designer), we exchanged a lot of feedback and tried to create a good and satisfying user experience. None of us had much experience with UX/UI, I had none, so this was a quite challenging task. However, I think that due to our ability to listen to feedback the overall UX/UI of the game ended up being quite good.

My collaboration with Liam (procedural and gameplay programmer) was also quite fruitful. Since most of our UI was connected in a way with the route system, we always made sure that our code was compatible and worked well together.

# REFLECTIVE ESSAY

## **What could have been better?**

---

At points during the semester, I felt like our team could have been more organised. Everyone knew more or less what they had to do but I think we could have benefited of having a more structured way of working. Namely in terms of prioritization, most of the times I worked on what I decided to even when I was not sure of what was more important at that specific time.

Individually, looking back I believe that I should have researched more before starting development, namely about the UMG (Unreal's engine UI tool). I went in a bit blind just to get things to work so that we could progress with our development. I knew very little and so I made some decisions that ended up leading to more work than if I had just figured out the proper way of doing it from the get-go. However, this is also hindsight, the best way of learning is always to practise.

I also think that I could have done better with the source control. I had no experience with using source control in a project with multiple people so at the beginning everything was a bit "scary", also unreal files do not have a great relationship with Github. My commits could have been better, as well. They improved during the course of the semester, but the first ones were not specific enough and one commit included too many file changes. I noticed this while writing this portfolio.

# REFLECTIVE ESSAY

## Final Thoughts

I learned a lot from the development of this project. Both in my specific field and others such as UX and UI design. Programming wise, this project helped me understand better the benefits and downfalls of the “quick and simple approach”. This is quite effective in the context of game development since it allows us to immediately get a feeling for the feature in the game. However, too much of it and you end up with a codebase that is unmaintainable. A balance is required.

I developed a lot of technical skills, such as my knowledge of Unreal and source control as well as soft skills like my ability to communicate effectively. These skills will certainly be instrumental in my future endeavours.