Internship Report

ENVERSED

STUDIOS

Jurrien Brondijk – s1004051

Company Coach: Tim van der Grinten

Teacher Coach: Yaliang Chuang

07-01-2019

Table of Contents

Summary

The past semester I have interned at Enversed Studios in Eindhoven. Enversed is a VR company that makes a variety of VR products, including entertainment, architecture or training programs.

I was employed under the gameplay programming team, who design, program and implement various elements into the products Enversed creates. During this time I have worked with Unreal Engine, making high quality VR products. Besides just programming tasks I also did a number of design tasks, found, created and implemented sound effects, held demos at the Innovation Expo, Dutch Design Week and client visits and more.

I worked on a number of projects, such as the NS free roaming project and BHV training project, which are projects for other stakeholders who hired Enversed to create VR experiences for them, as well as some entertainment projects for Enversed's entertainment center. Through these projects I got to see a number of applications for VR, and different design decisions that come with them.

Overall I believe I've grown a tremendous amount, making huge leaps in especially technology and realization, but other expertise areas as well. I'm super glad that I got the opportunity to intern at this company.

Introduction

In the last few years I have gained interest in game development. After doing a project at industrial design in which I developed a virtual reality game I knew I wanted to further my expertise in this area. As I tend to learn most by practicing I thought an internship would be a fantastic way to develop the skills I wanted, as well as gain insight in what's it's like to work in an experienced, multidisciplinary team in a professional context.
Enversed was a natural fit, having worked with virtual reality in the past I already knew a lot about the technology. Regardless, as a company that specializes in VR, Enversed was still able to show me a lot I did not yet know. It being a smaller company also gave the opportunity to work on a variety of tasks rather than just programming or design.
In this report I will talk about my experiences at the company.

Company Description

Figure 1: Enversed's Entertainment Centre

Enversed is a virtual reality company that provides a variety of VR products. "From 3D visualizations and interactive training applications to large-scale multiplayer Virtual Reality games"[1]. The company has its own entertainment center that it develops games for, as well as a studios division that does projects for other stakeholders such as the NS, MSI or the TU/e. Projects such as recreating the Atlas building in VR so you can get a feeling for what it's going to be like[2] or generating awareness for the effects of sleep deprivation by having people experience them in VR[3].
The company focuses fully on VR and has a lot of expertise in the area as a result. They utilize a variety of different VR devices, such as the Oculus rift, Oculus GO, HTC Vive, Virtuix Omni or Gear VR depending on what best suits the project. They are also on top of new developments such as the Vive Pro and the use of VR backpacks to allow for wireless large scale VR experiences.

The studios department has a few team working on different aspects of the projects. There's management, working on gathering new projects and managing the entertainment center. R&D works on the back-end systems and the more technical side of things, laying the foundation the other teams work on. The art team creates the visuals for the various projects. Anything like 3d models, textures, particles or assembled levels are likely made by them.
I was employed in the gameplay programming team responsible for designing, programming and implementing high quality VR experiences for both the entertainment center and other stakeholders. Most of the ideas and initial designs of games come from Tim van der Grinten, one of the founders and my company coach. He passes them on to the team lead of the programming team, Rutger Meijering, who writes concrete tasks on what work needs to be done to realize the ideas.

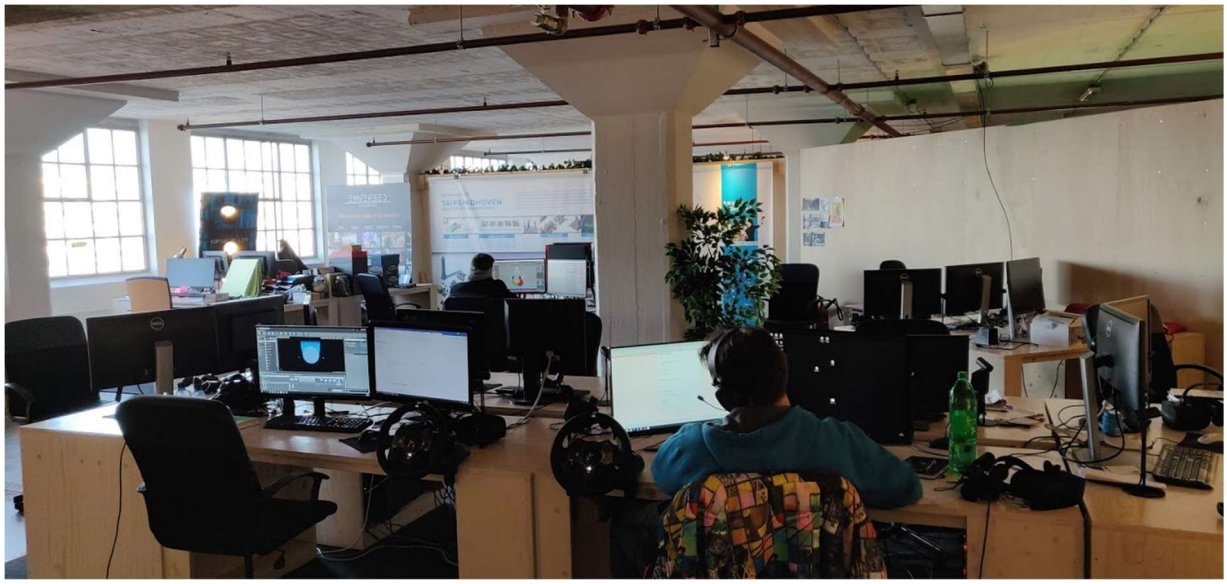


Figure 2: The Studio Office and (in the front) the Gameplay Programming "Island"

Goals

Before starting my internship I had a few goals set out for myself. Of skills I wanted to develop based on past experiences and activities I would be undertaking during my internship. Two of these goals went unmentioned in my PDP, as I constructed them after I had gotten my formal approval. The most important of which was that I wanted to keep a logbook throughout the internship, keeping track of what I did every day and already writing down learning points so that no information gets lost. This Logbook can be found among the appendices.
My other goal was formed after we had gotten an introduction of the company. I was told there would be several opportunities to be involved in the user testing of videogames we were going to develop. As I'm interested in the user experience I was very interested in attending these events and work on my user evaluation skills.

Improve Communicative skills

I had a feeling that poor communication occurred quite often in my projects. People would often not all be on the same page about a project, or unaware on what everyone is working on exactly. Especially in my last project this caused me and my teammate to often have conflicting files and we would have to redo a lot of work quite regularly as a result.
With me starting to work in a team of a larger scale than the ones I had been parts of at ID thus far I wanted to see if I could improve on this. Looking at the measurements the company takes to ensure proper communication.
Another aspect of communication I wanted to work on was that of critique. I have a tendency to keep my thoughts to myself, especially when someone's been working on something quite long I would have trouble asking them to change it. Though in the end if something has to be changed it might as well be made clear immediately. If brought constructively with proper argumentation it will only benefit everyone. This is what I wanted to work on, speaking my mind when necessary.
I planned on discussing progress in this front every month to see if I'm progressing or not, and what I still needed to work on.

Further Technical game development skills

Over the course of my study at ID I had come in touch with a variety of programming languages and programs. During my internship I was going to be to add another one to that list, Unreal Engine. I would also be getting in touch with much higher quality and complexity products than ever before, so it was going to be a great way of improving my technical skills.
Almost all the interns that Enversed usually hire come from game development studies such as the one in Breda at the BUAS. I had a feeling that, compared to them, I was going to be lacking in experience a bit. If I could get to a level similar to theirs I would have considered this goal well achieved.
As I thought I was going to be behind the other interns I had also planned to have a look at Unreal engine over the course of the summer, already getting the basics down by making a simple game.

Learn more about the theoretical background of Game Design

As I was gaining more interest in game design I often found myself analyzing and critiquing the games I was playing. I noticed patterns in for example the different iterations of the combat system in the Assassin's Creed series games. I was making these analyses based purely on personal experience and preferences however. I thought it was about time I was going to learn some more theoretical backgrounds regarding game design by reading "The Art of Game Design" by Jesse Schell[4], as well as some papers regarding game design theories.

Activities

During the summer vacation I already did a few activities in preparation for my internship, mainly starting to read the book I mentioned. One of the biggest eye openers from reading that book was its answer to the question "what skills does a game designer need? In short, it's all of them. Almost anything that you can be good at can become a useful skill for a game designer" [4]. I've always been interested in a large variety of things, and saw it as a weakness, not being able to get really good at one thing, but just decent a few. The book showed me what strength this trait could have. Having a solid understanding of a lot of things allows you to take each of those things into account realistically when designing for them. Video Games combine a huge amount of skill sets. While the products of each of these skills sets can often stand on their own, combining them properly only further strengthens the experience of the end product.

I'm reminded by a paper I read during the course Aesthetics of Interaction, "From Interaction to Trajectories: Designing Coherent Journeys Through User Experiences"[5]. In this paper the designers expressed the importance of designing the transitions between digital and physical aspects of their experiences, making the end experience more uniform. This of course requires knowledge of designing both physical and digital experiences. I believe this extends to an all-digital product like a videogame, where it's important to once again design to unify all aspects of the product, creating one optimal experience. Of course doing so requires knowledge of all these various aspects of the end product. With that in mind I was excited to learn a variety of new skills by doing different kinds of tasks over the course of my internship.

Most of my internship was spent working on projects such as the NS train free roaming and BHV safety training. While sometimes pitching in and working on entertainment products by the names of Snowdown, Skyracer and Skyrunner. In most cases the design and concept phases were already finished and we were mainly tasked developing either the final product or a demo of sorts. Still I was able to learn a ton in the time I spent at the company.

Training

Before we could start working on the actual projects me and the other interns were being trained in the use of Unreal Engine and GitLab. These were the most important programs we would be using throughout the internship. I had a small amount of experience in programming and almost none at all in Unreal, I had not managed to get much practice in over the summer, so I was quite worried I would have trouble keeping up. This wasn't really the case at all, in fact I did very well in understanding more complex systems after getting to work with them for a while. We were trained in a lot of different aspects ranging from basic coding in Unreal, Methods to keep code clean through company rules, learning how the company's own code (the Framework) works, working with version control through GitLab and SourceTree and, what was probably the most difficult, learning how to program Online games.

While coding normally you already have to take into account where certain pieces of information are stored, in an online game the amount of locations increases with every player that gets added. Is the information stored on the server, or on any of the current clients? It adds another bunch of layers to the thought process while you're programming. It was very challenging and I had to keep improving on it as the internship went on, but I really liked the challenge. What helped is that unreal has many different types of objects used for storing data, each existing in different places and having different properties. The table on the right shows the most important ones when programming for multiplayer.

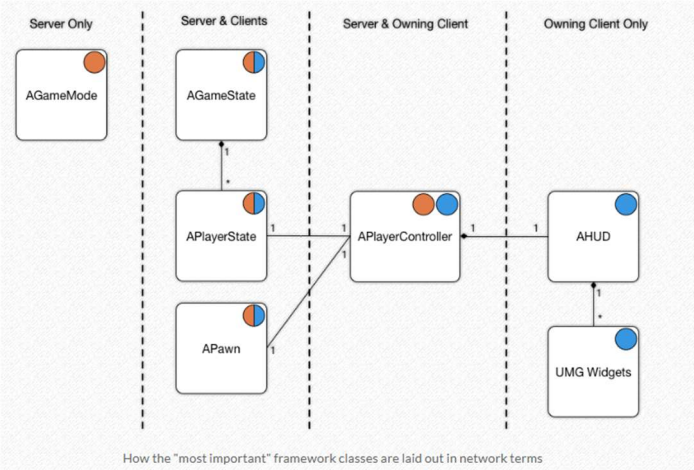


Figure 3: Unreal network/Framework Explaination by Nafonso[6]

Learning about version control was very valuable to me as well, it being the first step in my goal of improving communication. I found out that what I was using in my previous project was a very basic built in version of it that gave us plenty of headaches throughout the project. The reason we were using it was simply that I wasn't aware anything better existed and how it worked exactly. Now that I do I'll be sure to use it whenever I work with big projects like these in the future, as it allows to recover files, reset to older versions or make experimental changes that can simply be discarded without the worry of it ending up in the final version. If there are conflicts, which we had plenty of during my previous project, rather than having to discard conflicting files, these programs allow you to choose which files to discard, and you have an opportunity to back them up so that you can remake the changes in them more easily. What also helped in regards to the conflicts was the open communication we had at the company. Everyone is sitting at the same table, so you can ask who is working on what files and avoid working on the same ones that way. While at first I had to adjust working in a space like this rather than alone at home, I grew used to it and actually feel more productive here now than I do at home.
After 2 weeks of training we were ready to start working on actual projects.

NS Train Free Roaming

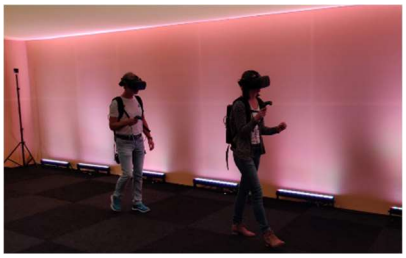


Figure 4: Participants of the Dutch Design Week experiencing the project.

The NS free roaming project was a project Enversed did for the Dutch Railways (NS). The NS was in the process of designing new train models and wanted to showcase their designs. They chose VR as their method of doing so as it allows experience the new train without them having to spend money on resources. The project made use of some of the latest VR technology, the HTC Vive Pro. This new VR headset allows for a play space of up to 10 meters in length, giving you the option to make a large free roaming space that players can experience. The project was made such that 3 people could explore the train together at once, walking around the same room, seeing the same train, and also seeing each other in the virtual world to keep the feeling you are exploring the train together.

I joined this project about half way through its lifetime and was mainly involved in making minor tweaks, implementing a spectator system and assisting in the project demos at both the Innovation Expo in Rotterdam as well as the Dutch Design Week. With the project being largely finished already there was a lot to be learnt from the decisions that had already been made up until that point, especially regarding VR. Among the more interesting ones is the way the company handles menus. In VR you can't put a menu on the player's screen as this is far too close to their eyes. All menus in the project had been given a place in the environment and could be interacted with using the controller, either by touching buttons or pointing at them.

This project was my first change to apply the things I had learnt during training and it became quite clear what I still had to improve on as we started working on it. In the beginning I was very quick to call a task finished without thoroughly testing whether or not it was working as it should. Other people had to then go through my code to find the mistakes. My code itself was also quite messy, in part due to my limited knowledge of the program, I simply did not know how to make it tidier, though partially because my priority was to make things functional rather than tidy. These were both things I kept improving on over the course of my internship.

Another side effect to me being a bit of a beginner was in my confidence, which lead me to have trouble communicating well as well. I often assumed everyone in the company knew better than me and as a result would not question anything they told me regarding code. I would not speak my mind about things not just because of the reasons mentioned in my goals, but because I assumed they knew what they were doing and was afraid of saying something stupid. This would improve over the course of the internship as well. As I gained more programming experience I also gained more confidence in what I created and I started to question the things the other programmers were doing much more often.

The biggest learning points from this project came near the end of it, when I was preparing it for the Dutch Design week, changing things we had learned about during the Innovation Expo. The 2 main problems we saw were that people didn't always know how to select the menus and that people often were confused in the teleportation area of the train. I got the task of fixing these problems and did a bit of a poor job at it. Upon reflection a big part of my mistakes came mainly from mindlessly following orders, not speaking what's on my mind and being afraid to make major changes, especially to something that's not mine to begin with.
In case of the menu selections I was asked to change the menu selection such that touching the menu would be enough for a selection, people no longer had to press a button. Meanwhile, a different change made it so that people had to make a selection in a different menu before entering the train. This menu did require them to press a button. I thought to myself that that might cause confusion, but didn't mention it. Once the Dutch Design week rolled around my theory was correct and I had to revert the change I made. If I had spoken up right away that might not have been necessary. I should mention that there had been instances during the project where I did do this properly, such as when a colleague was making changes to the teleportation system I thought to be counterintuitive. In this instance I ended up asking the team lead on his opinion, who agreed with me and had me do it my way instead.

The teleportation fix I had to work on before the Dutch Design Week was a different case. To provide some context let me briefly explain the system. Since the entire train didn't fit in the play area, the train was cut in half and we'd teleport people between those halves. The teleportation required people to walk towards a certain point once they entered the center of the train (the left of the play area), once they reached that point it would send them to the other train half.
When they entered the teleportation zone we would remove the trains so that all the player could see was the sky and the point they had to walk towards. We lead them to that point using some animated arrows, pointing at where they needed to go. This was not enough feedback for the people as they often ended up confused and asking for help. When I started to improve it I didn't want to change much and simply added more arrows so that wherever you'd look you'd see where to go. The problem was that the environment they were put in was also quite distracting and very open, causing players to wander off, which my solution did not solve.

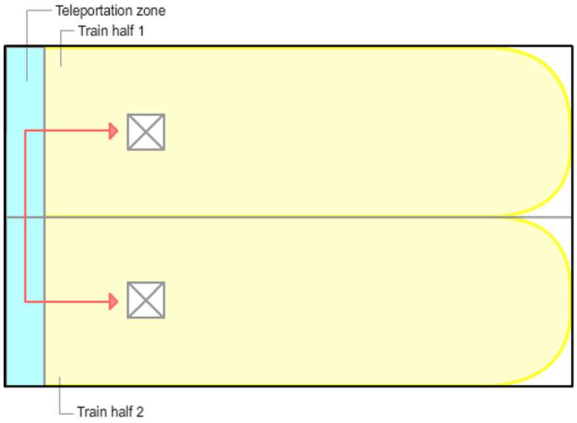


Figure 5: Schematic of the Teleportation System

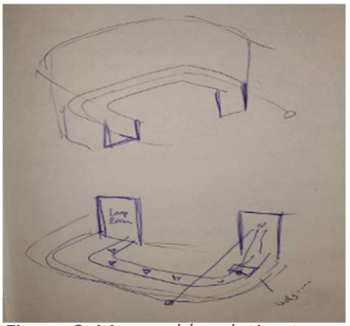


Figure 6: My coach's solution to the teleportation problem.

My company coach took notice of this and helped me design a different environment that does a better job at telling the player where to go and also restrains them more, it basically leads them through a small hallway ending at the teleportation point. It wasn't a difficult change, it simply required taking a step back, away from the computer to think about what the actual problem was and solving that with a new design.

SkyRacer

SkyRacer is Enversed latest entertainment project. It's a multiplayer VR racing game, which in itself is not very groundbreaking, after all racing games have existed almost as long as gaming itself. What was interesting to see is how Enversed designs their game for their target market. Since their center is aimed at a general public they keep their games simple and clear. The games are also made for social events which is why an important requirement was the players being able to see each other throughout the racetrack, so that they are aware of the fact that they are racing together.

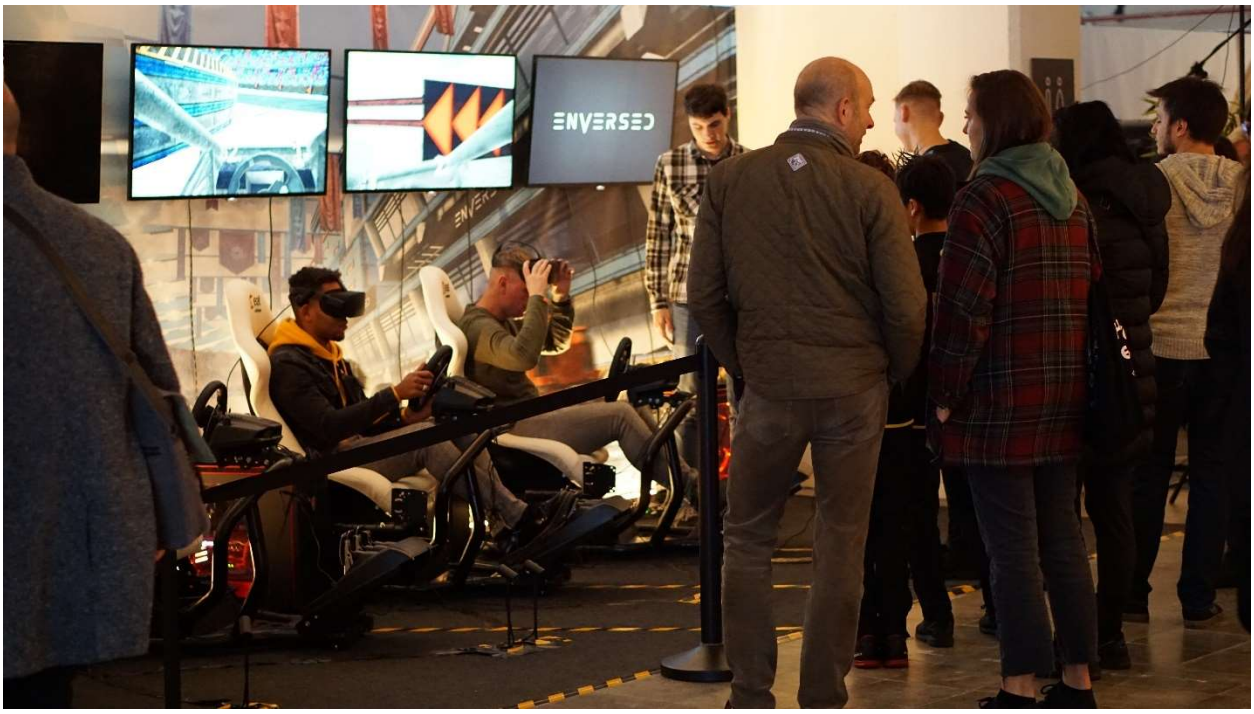


Figure 7: My managing the SkyRacer demo at the Dutch Design Week

We started working on Skyracer in the week I went to the innovation expo. 3 weeks before the Dutch Design Week, at which point we wanted to have a playable demo to be able to show. I took on the task of the force feedback. The players controlled to car using an actual steering wheel. This wheel had several options for force feedback that had to be triggered using code. After figuring out how the force feedback worked exactly, I went through the process of deciding which effects I was going to use at what points, imagining what it would feel like when playing. Implementing the force feedback required a lot of mathematics, calculating at what intensity the effects should play at in certain scenarios. When accelerating the car I wanted the engine revving to be translated into the wheel shaking. I figured this should be a parabolic function so I wrote one with the help of an online tool.

In this function, *b* is the max velocity of the car, *g* a constant used to decide what part of the acceleration receives the effect and *a* the intensity of the effect. The many variables can be changed around for different car types so that each car has a different feel.

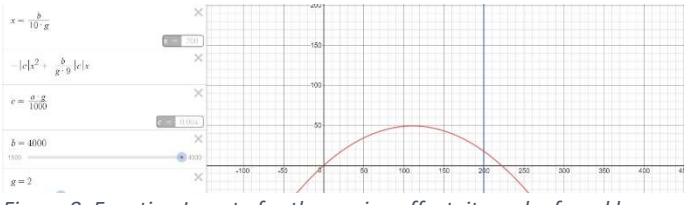


Figure 8: Function I wrote for the engine effect, it can be found here: https://www.desmos.com/calculator/wfi3zvopqe

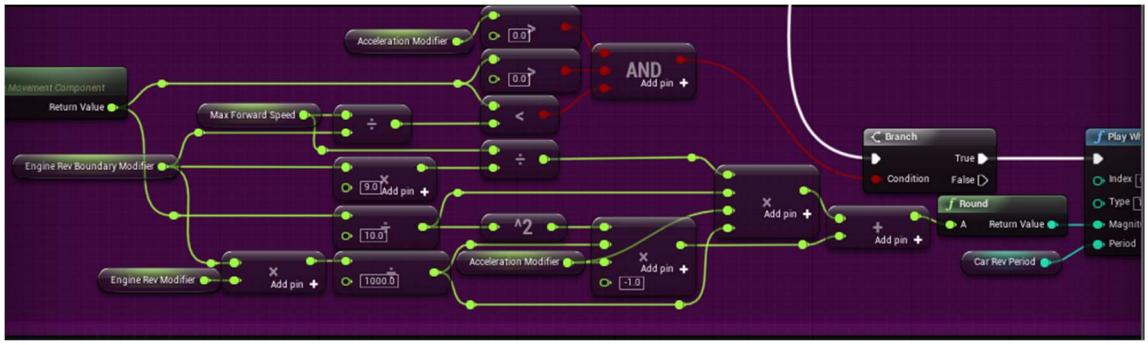


Figure 9: The parabola in Unreal Engine BluePrints

Another part I did which required math was the feedback of the car being hit. The event of an object being hit contains a lot of information, such as the force and location of the hit. Almost all the values came in the form of vectors, so I has to think of how I was going to use those to get the intensities that I needed to play the effects at. In the end, though it took some extra thinking, the vectors actually made everything a lot easier to work with. Using the normal of the hit, I could not only determine the car was hit from the left or right, but the output of the dot product let me know exactly how much the car was hit from the left or right, giving a very natural feel to the effects. I hadn't worked with vectors in a long time, but this definitely opened my eyes to how useful they can be, especially in programming games.

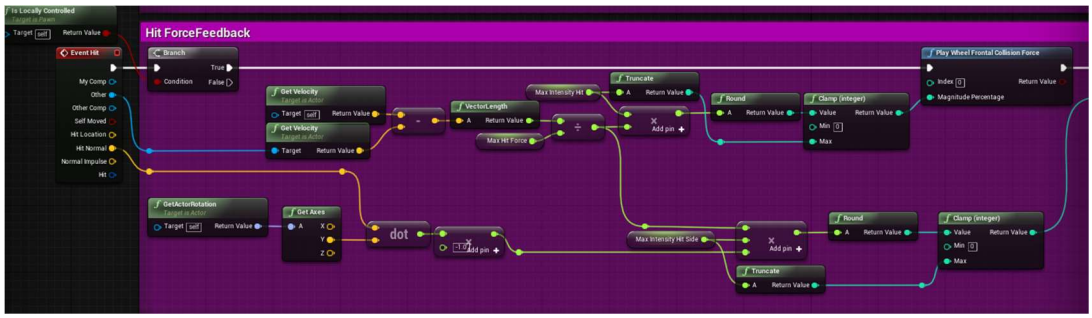


Figure 10: The code for the car being hit

Once I had the code implemented I spent some time tweaking all the variables of intensities I had created until I felt the effects felt natural, also having other employees test the effects and hearing their thoughts. Eventually I considered it finished and was quite satisfied with the end result.

SnowDown and SkyRunner

SkyRunner and Snowdown are already existing entertainment projects at Enversed. SkyRunner being a running/parkouring game and Snowdown a team based shooting game, think laser gaming with snowballs. Throughout my internship I occasionally picked up tasks that involved making changes to these games. Fixing bugs, adding minor gameplay features and the like.
My most notable changes are both for Snowdown. I was tasked with adding turrets to the game to prevent players from entering certain areas, as well as add more feedback to the teleportation mechanics the game has.
I enjoyed working on the turret as it allowed me to create something from the ground up rather than adapt something that was already there. It also allowed me to implement my own ideas as to how the turret should behave when deciding who to shoot at. I was happy with the end result and the fact I programmed it almost entirely by myself. This and the force feedback in SkyRacer were really showing my progress in being able to create systems with little to no supervision.

Figure 11: The turret shooting a snowball

The teleportation feedback was a very different sort of task. The teleportation mechanic has a cooldown, after you use it you won't be able to use it for a short period of time. During that time pressing the teleportation button should give you some feedback. I went for audio and force feedback as it's clear and instant, as well as a visual indicator of how far the teleport has recharged. While trying to find the sound I had in mind I came across a bunch of websites for getting (royalty) free sound effect that I could use for projects of my own. I also already got some experience in editing these sounds. I used a program called Audacity to change the sounds pitch and speed to fit the style the other sound effects in the game had. I then added force feedback in sync with the sound, connecting the two since they go off at the same time, giving it a nicer feeling. Working on this task made me interested in being able to create sounds like this myself. Especially for UI, it's hard to find sound effects that really suit your style. I was already interested in following the sound design course at industrial design, this task made me further realize what value it has to be able to easily find or create sounds like this and I decided to sign up for the course.
Adding a visual indicator is a bit more difficult as in VR you have to give such visuals a place in the world. At first I added a bar above the health bar that was already present on the hand. The bar would empty when you teleport and fill back up, once it was full you could teleport again, at which point the bar would also become a bit brighter, indicating the teleport being ready even better. It was pointed out to me there was an unused part of the hand that had a charging bar built into it. So I removed my bar and hooked my code up to the existing part of the hand. When you teleport you're likely to notice the spot on your hand becoming black, at which point it's easy to figure out what it stands for.

Figure 12: Teleportation charge on the hand

While working on NS and Skyracer we had been trying to reach deadlines and as a result the common way of working for the company had partially been ignored. Now that the Dutch Design Week had passed we would be getting back into the normal way of working for the company, which is assigning yourself to a task, marking the task as finished once you're done so someone can check it and moving on to the next task. Part of this workflow as well is writing new tasks. If we found out about any bugs, or had suggestions for changes we could write a new task and the team lead would add it to the to do list. I found this to be a really good way of doing things. When you pick up a task you can see who wrote it and go to them if you have any questions about it. Having a public to do list like this also helps in keeping everyone aware of what still has to happen to the project before it's finished.
The list of tasks itself was also cleverly organized. On the board of tasks, the more to the right a task was, the closer it was to being finished. We were encouraged to pick tasks closer to the right as it prevented there being a large number of unfinished tasks. I can see myself work in a way like this even on solo-projects. It will allow me to have a clear picture of what tasks need to be done and which have priority over others. I have a tendency to work on unimportant tasks that are more fun, this would prevent that to an extent.

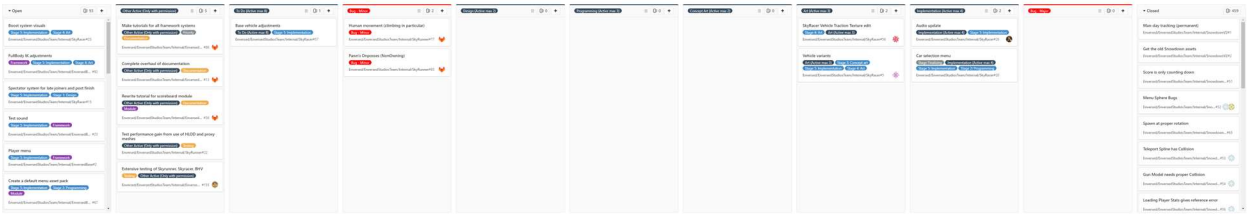


Figure 13: The issue board, we would work from right to left on the various tasks presented here.

Working on these projects I had also gotten to a level of programming where the team lead entrusted me in checking the work of others. This was especially helpful in my goal of giving critique to others. I was pretty good at recognizing patterns, noticing common mistakes that they made. One intern in particular would do a poor job at properly testing his system properly. Either he didn't test them in multiplayer or only in a very specific scenario. I would constructively tell him about this and suggest ways he could improve. Though at times I felt like I was complaining a bit too much, I do have a feeling it helped him see what he needed to work on and try to improve on those.

Another activity I undertook while working on these projects is attending a user testing event. We had the people who work at the entertainment center play SkyRunner and give their feedback. For their first few play sessions I gave them as little information as possible and wrote down their thoughts, as well as things I noticed from just watching them play. Later a colleague joined me and together we started interviewing them a bit more, trying to pull as much information out of them as possible. My colleague reminded me to take into account the type of user we were dealing with. In this case, our buyer/distributor who has plenty of knowledge about the types of products, and the way the end users tend to react to them, what they usually struggle with, etc. So instead of just asking how they experienced it, we also talked about how the end users would and if changes needed to be made for them.

BHV Fire Safety Training

The BHV project was the final project I got to work on during my internship. Enversed is going to sell virtual training exercises to BHV companies, who train people in what to do in the event of, for example, a fire breaking out at your office.

Figure 14: Extinguishing a fire in the BHV project

When I started work on the project, all it consisted of was the basic mechanic of the fires spreading. At this point all that it did was increase a number within the fire object. It had no visuals or anything of the sort. All the other employees and interns were working on other projects while I took on BHV, so nearly all the systems that ended up making it into the final version were either fully developed by myself or I had major involvement in. This included implementing the visuals of the fire, altering the fire to work in multiplayer, the fire extinguishers, openable doors, creating a spectator, implementing sound effects, creating a smoke system and polishing the spectator interface the team lead made. I also attended several demos we held with potential buyers of the training and processed their feedback.

I was able to work on most of the tasks with little to no help, which really shows how far I have come. The smoke system is an especially complicated function which checks if rooms are connected to one another and spreads the smoke in those rooms evenly when doors between them are opened. At first the team lead kept saying he or another employee would write the system but due to him getting sick I eventually did get tasked with it and had to do it on my own with nothing but the vague description he gave me. After a day or two I managed to get it to work and I was very happy with the result.

Figure 15: Smoke building up after a fire

I learned a lot implementing sound into the project. In Snowdown I had created one simple sound effect that had to play at the press of a button, which was not very complicated. This project had no sound effects at all, I needed sounds for the extinguishers, doors, fires and more. I was able to use the websites I had found to get most of these sounds, but I still had to edit them and add them to the project. A lot of sounds needed some extra processing, such as the fires or the extinguishers, which needed audio to loop. I did some research on how to do this properly. At first I would hear popping noises whenever I cut a piece of audio, I later learned this can be avoided by cutting the audio at points where the soundwave crosses the X-axis, so called zero-points.
I looked at how audio was implemented into other projects, such as Snowdown, and used this as reference for putting the audio into this project. This way I learned how to properly implement audio into Unreal, creating sound classes that each have separate settings which can be easily changed for all effects that are a part of that sound class.

By far the biggest learning point this project was in communication. I had the feeling the project needed to be very realistic and high quality and as a result spent a lot of time getting the fire extinguishers to behave realistically, among other things. This actually made us fall behind on schedule a bit. As a team we had a large talk on why this occurred and what needed to be improved. It was made clear to me that for BHV, for the time being, we were making more of a proof of concept to get potential buyers interested and have them fund the project. To an extent it's the company's fault for not making clear to me what they're aiming to do with the project, though it was also wrong of me to assume things instead. I should have asked more often what needed to be finished before the deadline and which tasks had priority and also why. Knowing why it's safer to make assumptions since you can ground them a little. It was a learning point for both me and the company, to have everyone be aware of the goal of a project and the requirements for that goal so that you can give tasks related to these priority.
Another noteworthy occurrence during this project was the introduction of a company Slack. While the open communication at the table was very useful the team lead had a feeling it was often also quite distracting. When you are in the middle of something and someone asks you a question it might take you a few minutes to recover your thought process and continue. This eventually adds up into a lot of lost time. To solve this he wanted to move some of those conversations to Slack, a messaging program. It would allow you to start a conversation in a group chat, relevant to the topic you have a question about, on Slack where everyone could read it and reply when they can. This is less distracting and has a chance to involve more people into the conversation (including people working from home, etc). Another benefit to me was that I became less afraid of asking something. I very well knew how distracting a question could be and as a result often didn't ask my question. With the Slack I don't have to worry about that and can simply ask away. Also, being able to type out the initial question gives you the chance to think about specifically what you want to ask and avoid rambling or putting unnecessary info. After you establish a conversation you can still choose to continue it verbally, now that you're both not working on something.
I was a big fan of this approach and if I ever work in a large team like this again will recommend a system like this.

Results

Over the course of the internship I ended up getting involved with a large number of projects and ended doing a variety of different tasks. The quality of my work increased with each project I undertook and I think it shows in the results.

NS Train Free Roaming

Besides numerous fixes and improvements my largest contribution to the NS project ended up being the spectator system. Looking back, it was a very simple system that did what was required of it, but not in an optimal way. Its basic functionality was that of being able to switch through different cameras in the level, as well as players their cameras. Later it was also required for it to switch levels, reset players their position and more. All these actions were done through button commands which made us the only ones really able to operate it. As a result a developer had to be present during the demonstrations to be able to help operate it.

Figure 16: Various camera angles of the spectator

The project was a way to gather data for the NS about various new designs of chairs in their future trains and we also ended up turning that data into a heatmap for them by making all but the chairs visible and coloring them based on the amount of votes. Sadly I'm unable to show these heatmaps because of the NDA.

SkyRacer

As I mentioned in my activities my main contribution to SkyRacer was in the force feedback system. My main focus when making the system was for it to be easily expandable. I made a system that checks what type of surface each wheel is on, and made sure it would not be extremely difficult to add a new surface type to that list. I made sure all intensities of the effects were settings of the car itself so that when new cars would be added they could be changed to fit the feeling of that car, heavier card probably shaking a lot more for example. The system left some things to be desired in the area of optimization, it makes a lot of checks many times per second, though as it stands it achieved the requirements I had set out for it and I'm satisfied with its end result.

Another minor contribution I made to SkyRacer was the visuals of the car selection menu. I added clear button prompts telling users how to use the steering wheel to select cars and the acceleration pedal to confirm their selection. I also made some adjustments to the controls of the steering wheel in the menu, to improve the overall feel of it.

I Like my work on SkyRacer as it focuses heavily on the feeling I'm trying to convey to the user and I feel I achieved those.

SnowDown

In the case of SnowDown most of my contributions were in small fixes and adjustments. Working on these I did notice the quality of my work increase while the time I spent on tasks was decreasing.
The most notable contributions by me, as mentioned in the activities, were the turret and teleportation feedback. The turret actually has an interesting bit of code in which it finds its most optimal target by looking at all the players in range, their health, their distance and whether or not anything is blocking the turret's view. This way the turrets efficiently take out any enemy team setting foot in their range. Their purpose is to prevent players standing outside the other teams base, killing opponents as they enter the arena, which wouldn't be fun for those players.
I'm also happy with the way the teleportation feedback turned out. In case of the visual feedback, the idea of having the charge bar emit light once it's fully charged was a very nice addition to indicate the bar being at 100% rather than 99% . I was happy with how all my additions fit into the already existing game, and made use of some features and systems that were already present in the project.

SkyRunner

My work on SkyRunner was exclusively in a few minor fixes in some audio aspects as well as the spectator and the play testing event where I gathered feedback for the project.

BHV Fire Safety Training

Of all the projects, the BHV training project the latest project I've gotten to work on and it also is the one I am most responsible for. As I said, nearly all game systems have been entirely or mostly made by myself. It contains

- fires that grow larger and spread across an area
- fire extinguishers the player can hold and use to extinguish the fires
- realistic doors that swing open or closed
- a smoke system that lets rooms fill up with smoke and spreads them evenly across rooms when they are connected
- a much higher quality spectator with a fully functional interface

What's interesting to look at is how much this spectator has improved over the one I created in the NS project. The functionalities it has, starting the training, toggling view mode, selecting and spectator specific players, and restarting the training are now all incorporated in a menu which can be toggled on or off. This is much more user friendly which is important considering the idea is to sell this training to different companies who will have to be able to handle the program themselves.

Figure 17: BHV Spectator Interface

I am very proud of the level of quality I have been able to make this project at, I believe it really shows how far I've come over the course of my internship.

Acknowledgements

Firstly I want to thank both my coach, Tim van der Grinten, and team lead, Rutger Meijering, for the opportunity they gave me by allowing me to intern at their company, despite my educational background not being quite what they are usually looking for. I've had an amazing time at the company and have learned more than I could have possibly imagined. Both have given me very useful insights and tutored me well.

Another very noteworthy mention is the other member of the gameplay programming team, Sjef van Buren. Whenever Rutger wasn't around he would take his role of helping us, which he also did a great job at, I learned a lot from him regarding programming.

Lastly I feel like I should thank the other employees and interns for fun times at the lunch tables and overall being fun to cooperate with, Ramon, Loic, Benjamin, Ian and Colin.

Appendices

Appendix 1: PDP
Appendix 2: Formal Approval
Appendix 3: Logbook

PDP Jurrien Brondijk

Identity

I am very passionate about videogames. I enjoy picking them apart and seeing what parts I think are done well and which aren't. I'm fairly analytical in this way. I'm able to look at a product and consider the implications certain design decisions have on the user experience. I'm very interested in (game) design, how to make things intuitive and how to convey an idea such that it results in an enjoyable user experience.

I have a wide skill set and a broad interest. I enjoy coming up with ideas as well as iterating on existing ones. I often take inspiration from existing sources, capturing what is good about a certain product and reimplementing it in a way that suits mine.
I also consider myself to be a bit of a perfectionist at times. When working on a project I easily get caught up in small details that I feel would enhance the experience, even if only by a small margin.

Vision

Videogames are a very special type of product, in which many different art forms come together to create a unique interactive experience. Each of these art forms has their own impact on the experience and ,if incorporated well, further amplify each other's effects. This is why I think it's important to design not only how each separate area is implemented, but also the way they interact with each other. I think one of the best existing examples of this is the game Journey by Thatgamecompany,, where music, art direction, sound design and even the controls of the game have a certain fit. The game is taking you on a journey and many elements represent this.

Videogames have grown a lot over the years and I am certain they will continue to do so in the future. With computers getting more powerful and new, innovative technologies being developed, game designers are constantly pushing the envelope of how we interact with their digital worlds. A trend that I do tend to see is that many developers are pushing for realism in their games. Both in making games look as realistic as possible, or virtual reality becoming more and more realistic of an experience.
Personally, I would much rather see how creative we can get, and how we can add to the experience. While a level of virtual realism like in the recent movie Ready Player One could be impressive, we should always wonder what it all really adds to the experience. Is VR the next step in gaming, or just a gimmick, similar to the Xbox Kinect?

Goals

The following semester I play to intern at Enversed, a virtual reality company.
I will be part of their design team and help make the games they offer people to play in their VR rooms. I will be working in both the design and programming teams, somewhat acting as a bridge between the two. Translating the ideas of the design team to something the programming team can work with while also putting in ideas and criticisms of my own.
To make sure I further my expertise in the various expertise areas and personal interests I have a few goals set for before and during the internship.

Improve communicative skills

Something that I keep running into with many of my projects is poor communication. During my internship I'll be working in a rather large, multidisciplinary team and I think communication will be even more important than it has been so far. This is why I want to be working on my communicative skills. mainly in getting ideas from one group of people to another in a way they understand. Like writing a basis of the code, or making a small prototype showing the idea. My sketching skills could proof useful here as well.

A communicative skill I want to improve on as well is critique. I find it easily to come up with criticisms, but enforcing them and asking people to change things proofs challenging to me at times. If someone worked hard on something I don't want to ask them to change it and throw out a lot of work, though if it would allow for a better overall result I should put myself to at least sharing my thoughts.

To make sure I improve in these aspects I want to evaluate my progress every so often. This is why I aim to take a look at how I've developed at least every month, preferably together with my company coach, so I can get a view on how they think I'm doing in these regards.

Further my technical game development skills

Over the past 2 years I've learnt a variety of programming languages and have used these to make a few games. During my internship I want to bring my technical skills to an even higher level, since I'll be working with much higher quality products than the ones I have made so far.
I will be working with Unreal Engine, which is an engine I have not worked with this before. I would like to get acquainted with it during the summer holidays before I start my internship. This way I can get straight to work there and won't have to spend time learning the basics of the program there. It will help me be much more useful even early on. Over the course of the internship I will get be working with more in depth mechanics of the program, like learning how to program multiplayer games.

To make sure I understand the basics of the program I want to try to make a simple 2d platformer prototype game in Unreal engine before the end of the summer. This should at least familiarize me with the tools of the program and the language.

Learn some theoretical background on Game Design

For a while I have been analyzing games from personal experience and I have been noticing a few patterns here and there. Though I have gotten useful insights from this I feel it would be a good idea to have a look at some theories by professionals, and try to apply these.
I found a book called “The Art of Game Design” by Jesse Schell which seems to offer an interesting perspective on game design. The book answers a large amount of design questions and leads the reader through the game design process.
I want to read this book over the course of the summer, so I can apply it in my internship. While reading the book I also want to apply some of the theories covered by analyzing at least 2 games I’m playing at that time, with the theories in the book. This can help me train my critical eye. And help me identify things the theory talks about, as well as give some context to the theory.

As well as reading the book I also want to see if I can find some scientifical papers about game design. Since there are probably lots of them out there I want to be looking specifically at intuitiveness in games. I wonder how interaction frogger can be applied to games, for example. I aim to read at least 5 papers, the things I pick up from them I can then also apply on my analyses and my internship.

Formal approval

Internship

Student Jurrien Brondijk **Date** 10/08/2018
Teacher coach Yaliang Chuang
Period activity ☒ September / December ☐ February / June

Personal Development Plan for formal approval	Does the choice of the learning activity align with the Professional Identity and Vision development of the student and are his/her choices well-argued?	Yes The internship aligned Jurrien's interest in game design and could help to shape his vision from the practical experience.
	Does the learning activity contribute to the development of the student?	Yes
	Does the chosen learning activity contribute to a balanced development in the Bachelor program of Industrial Design?	Okay It would facilitate Jurrien to apply the academic knowledge to the practice. He aims to reflect at least every month to improve his development and understanding.
	Are the goals well formulated?	Yes Jurrien formulated three primary goals of the internship. He also specified several activities for preparing beforehand in the summer holiday.
	Complete the aspects for the chosen learning activity:	
Internship	Does the company profile align with the requirements for internships?	Okay
	❖ *Doing an internship at one-man businesses is not allowed; unless the company owner is currently teaching at the Department of Industrial Design, Eindhoven University of Technology.*	
	❖ *The company must support development in expertise areas.*	
	Does the company coach align with the guidelines for internships?	Okay
	❖ *The company coach must hold a MSc. degree in (Industrial) Design or has at least 10 years of professional experience as a designer.*	
	Can the student work on a clearly framed design project or tasks?	Okay
	Positive aspects company*:	Negative aspects company*:
	❖ The company is very open towards my own learning goals. They have also offered to help me with my FBP if the internship goes well.	The company creates small (30 minute) experiences, which possibly have a bit less room for creativity then a more lenghty game would.
	❖ The company uses a programming language I am unfamiliar with (Unreal Engine), allowing me to learn a new language. I will also be working with Multiplayer games and learning how to program those.	The internship will focus heavily on Technology and Realization, and less on other expertise areas.
	❖ I have worked with VR before and have some experience in the technology.	I will mainly be working on existing projects and not help create something new.
	**Discuss these positive and negative points in the teacher coach meeting and discuss how the student can develop expertise areas that might not be covered within the internship.*	
Exchange	Exchange University and program	[Description of the program.]
Minor	Minor at Department of Industrial Design at University of Twente; or at the Department Industrial Design Engineering at Delft University of Technology. (No other departments at these Universities or other Universities in the Netherlands are allowed without permission of the BoE.)	[Description of the minor.]
	Minor at a University elsewhere in the Netherlands	Next to receiving formal approval from the teacher coach, the student also needs to file a request to the Board of Examiners.
Electives	What are the chosen electives?	[Elective], [Elective], [Elective], [Elective], [Elective]

More information: ID.internshipcoordinator@tue.nl
This form needs to be completed, signed and forwarded to the student. In case of an internship as activity, the student has to add their personal development plan plus this form signed by the teacher coach to the appendix of their internship report. In case of an exchange, (a copy of) the form needs to be handed over to the International Office at the Department of ID.

Version 2.0 – 2018/04/01

TU/e Technische Universiteit Eindhoven University of Technology

	In case a student chooses to do more than 15 ECTS worth of electives outside of the Department of Industrial Design, the student needs, next to the formal approval of the teacher coach, file a request to the Board of Examiners.	[Generations before 2015-2016 choose 6 electives, later generations choose 5 electives]
Approval	The personal development plan and chosen learning activity are approved by the teacher coach**: ***Provided the request is granted/supported by the BoE.*	Okay

Teacher Coach Signature

Logbook Internship Jurrien Brondijk

03/9/2018 mon
Today me and the other interns received an introduction to some of the games developed at Enversed. We got told about the people in the company. The company goal is mainly to develop games for the VR experiences they offer but they also do projects for other stakeholders as well.
I got assigned a desk and we started to learn about the workflow of the company. A combination of GitHub and Sourcetree makes for good communication which is teaching me a lot about communication, great for the goal I set.
We also started working with Unreal Engine, it's blueprint based programming takes a lot to get used to. I prefer regular coding but it's good to learn about other methods as well.

4/9/2018 tue
Worked more with unreal. Getting into more details about using the document in a way other people can read it as well. Very important for working on a game in a team.
Internship seems heavily programming focus, going to have to ask if I can get some tasks regarding design as well.
On Wednesdays they should have evenings where they test out demos, maybe I can apply those design criticism skills there.

5/9/2018 wed
More unreal, worked on our platformer making scripts to spawn objects. These scripts (interfaces) communicate in a different way making it easier to transport the game objects to a different project. This way of doing things is great for what they're doing here because they often want to put certain objects in different projects.
We also got an explanation of how Online Networking works in Unreal and on how to package and export our game so it can be tested.

6/9/2018 thu
We got quizzed on the online Networking, to my surprise I remembered more than the other two interns who had more experience in making games than me.
After this we started working on a new test game involving a button. We had to make a pressable button. For this we made the character able to check whether or not it's looking at the button and then press it. The challenge after that was to make the button function for multiple players. The button would remain pressed for 2 seconds before returning. During that time it wasn't allowed to be pressed again. This required our knowledge on Online Networking. Who do you send what information? What does the server need to know, what do the clients need to know? It was quite challenging but we got it to work eventually.

7/9/2018 fri
Continued work on the button and started on a traffic light. Within the traffic light we learned more about changing material properties of objects. I learnt quite a few tricks.
Also online networking was very important once again, the light needs to be accurate for all players. I'm having a bit of trouble keeping up with the programming. It's a lot to take in after all.

10/9/2018 mon
Today we finished the traffic light assignment. I actually caught up to the other interns who are more experienced than I am.
I also installed an Oculus rift on my desk, for testing the games we'll be making.

11/9/2018 tue
Today we got an explanation of the company's framework. It involves a bunch of systems they use in each of their games. Every game inherits aspects of the framework. They do this because they have a lot of projects that require the same systems. With the framework they don't need to make each system multiple times for every project, or start from scratch when making a new project. When something is changed in the framework it is changed for all the projects, so they make sure everything works before rolling it out to the projects.
After the explanation we got to apply the system and created a character in VR that we can take control of.

12/9/2018 wed
Today we had some issues with the public transport so some members of the team couldn't be there. Because of that their planning didn't work out very well and we were given some substitute exercises with the framework. We made some updates to the movement in VR and started working with items.

13/9/2018 thu
Today everyone was present again. We told the missing intern from yesterday about what we had done and continued where we left off yesterday. Today we were adding more proper movement and guns which each have different shooting behavior.

14/9/2018 fri
Today we worked with menus through the framework. This also introduced the respawn mechanics to us.
Afterwards we had our first feedback moment.
The overall feedback was very positive.
We'd gotten to a level where we're almost ready to work on real projects. Within this a possible task for me might be working on the steering wheel of their racing game. The wheel has a lot of options for feedback and I could see what feels the best for the user. This aligns with my goals of looking at not only programming but also user experience.
What also helps with this is the wednesday evening test moments. Once the project is up and running we'll have feedback sessions with the team every monday, playing the game and providing feedback to the teams on what can be improved. Every wednesday evening there are feedback sessions with people from outside the company as well. I volunteered to be present at these sessions so that I can process the feedback from these sessions. This allows me to apply my skills in usertesting on a more professional level which I am very excited about.
Within the project there will be a lot of different tasks to work on. We're not limited to programming tasks which is something I feared when we were being trained in programming so much. There's design and art tasks we can take upon ourselves as well if we so desire (provided we discuss this with the leaders). We're also always welcome to suggest things to be added or tweaked about the games or even framework.

17/9/2018 mon
Today we arrived at work and things were a bit hectic. The studio had a bunch of deadlines coming up and we had to help them get some things done. I was assigned to work on a project for the NS together with another intern. The project still required a few features and there were some bugs in the code that caused systems to fail. It was very interested to get to work on a nearly completed project like this, looking at how their code was created, and adding to it.
We got a list of tasks we tried to work through that day. We added a possibility of switching spectator cameras as well as switching levels. We turned VR off for the spectator. We added cameras the spectator can look through. And we fixed some bugs that caused VR not to work. We ran into a few issues but got most tasks done, or so we thought. We hadn't thoroughly tested our systems.

18/9/2018 tue
Today we were told most our systems weren't working entirely as they should and that we should have tested them more. At this point they taught us a few methods of testing code. We already let the code print text at certain points in order to see if that code was being ran properly. Now we were taught how to use Break Points. Break points allow you to run the code piece by piece instead of real time. This way you can look if each individual piece is being run properly. Very helpful when determining if certain values are being set properly, etc.
So we continued working on the NS project, not doing more testing as we worked on it. We fixed our systems and added more functionality, like the name of the player or camera you were spectating being visible on screen. We ran into a lot of issues and constantly had to change minor things hoping they would be fixes. It was actually quite satisfying finding bugs in the code and finding fixes for them. But at a certain point I got a bit tired of it. Once you don't know what causes an issue it can quickly become very frustrating trying to fix it.

19/9/2018 wed
Today we continued working on the NS project. Still trying to fix issues.

20/9/2018 thu
Today felt quite sick but still went to work. When I arrived however, I noticed I didn't have the energy to focus properly. I soon went back home to rest.

21/9/2018 fri
I was still sick today.

24/9/2018 mon
I was still sick today but decided to go to work anyway.
While I was away my fellow interns made good progress on the train project. I wanted to load the project onto my computer but couldn't because I was missing some files. I had to install these first which took me about 2 hours. When that was finished I made a list of the controls the spectator has in the project.
In the afternoon we attempted to fix some of the last major issues we encountered with the project. It took us some time to find them all. I made a few suggestions but had a feeling I wasn't really being listened to. The fixes we found weren't exactly great either. They worked, but they our solutions weren't very intuitive. That last part is equally important in a large project like this. Since your code has to be expanded upon by other team members you have to make sure it's

understandable for them. The code shouldn't just work, it should make sense just by looking at it.
Understanding someone else their code is also something we're learning while fixing these bugs. Even though I don't enjoy the bug fixing very much it is an important skill to have. We were constantly prying into the problem, trying to see what happens in the code, and didn't put enough effort into trying to read the code and understanding exactly what it does. We found a solution but weren't sure why it was the solution, at that point you know you don't have the ideal solution.
I have a pretty good sense of knowing when something is wrong, or should be done differently. I just don't always know what exactly in what way. I think that is something I need to work on, because if I'm only pointing at issues without providing any solutions it's just gonna feel like I'm complaining a bunch.
We ended our day using sourcetree wrongly and losing a lot of our progres from today. This is also a lesson to us, constantly saving and pushing your files to the server. Working on the right issues in their corresponding branches. And most importantly, discussing what you're working on with other members so that you aren't editing the same files which will later conflict with each other.

25/9/2018 tue
Today I helped set up a VR area large enough to test the train project. This proved very difficult as the HTC Vive isn't exactly designed for such a large play area. Also we realised we had a lot of interference from sunlight so moved to a dark room. This one wasn't as spacious but allowed us to test the project to an extent.
While I was doing this the other interns recovered the progress we lost yesterday. We continued testing and fixing issues. This time also testing the project in an environment close to what it's going to be used in later on. So we actually got to walk around in the trains for once.
There was a major issue regarding the chairs in the trains which we found a workaround for yesterday, but fixed properly today by removing excess chairs.
I was asked to find a way to read data from the file we're writing to, but found no way to do this as of yet.

26/9/2018 wed
When we got to work we had a talk about finishing the NS project as well as our next project and the Dutch Design Week which we needed a schedule for.
The upcoming project (skyracers) is going to focus heavily on Polish, making sure the driving mechanics feel good.
It will be showcased during the Dutch Design week so we have to make a working version of it on quite a short notice.
After the talk we were assigned some regular issues again. I worked on expanding the spectator system with some new features. This allowed me to tinker with a few aspects of unreal I hadn't worked with yet. And I actually got most of the issue done myself. When things got a bit cluttered I made a sketch on paper of what the code had to do, this really helped me write it and I will continue doing so as I get more of these kinds of tasks assigned. I also got some pointers on how to make the code more efficient and easier to read for others. (boolean is NOT true, select node).
This week we have to finish these last few issues and then do some large scale tests to make sure the whole project works as it should.

27/9/2018 thu

Today we continued fixing issues with the train project. We started the day with ideating a solution to some of the problems we found yesterday. Mainly the fact that teleporting is quite disorientating. We decided we could make a visual that both shows the player where they will be teleported as well as help them find the teleportation pad. I made a few sketches of what I imagined it would look like. After this was decided we went back to work. I fixed the last few issues with the system I made yesterday and picked up a new task, adding a teleportation system to one of the maps. This system was already present in a different map so all I had to do was copy it over. This wasn't entirely possible however, as the script was referencing objects within the level. I had to recreate these objects and reference them accordingly in the script. It was important that I looked at exactly how the other script was built up. Once I copied everything it didn't work still, because it turned out I overlooked a boolean that had to be set to "true" by default.

With the system now working I tested it extensively and found out the way in which the teleport worked caused issues. The play area would shift around ever so slightly when using the teleport. This is dangerous because as users use the teleport they might get out of sync with one another. This would cause you to see a person 3 m. away from you in VR but they could actually be right in front of you which would cause people walking into each other.

While working on this level figured the spectator cameras we placed a few weeks ago weren't placed correctly for this level so I moved them around. Placing them at various angles giving a nice overview of the train.

I'm starting to get a hang of the workflow within the team, using gitlab to work on issues is actually really nice. You can see exactly what everyone is working on and what still needs to be done. We are also very communicative as a team. If you have a question you can always ask and get help quickly. We also tell eachother which of files in the game we are currently working on so we don't get conflicting files (if two people work on the same file this causes issues). This is actually something that went wrong often in my previous project working on VR at ID.

It still happens sometimes that progress gets lost but with all the measurements taken here it happens much less often and is also easier to deal with.

I'm really surprised at how much I've learnt already. I'm able to contribute to programming tasks within the team quite well, finding and solving issues. I'm enjoying working on these problems as well. Seeing the project come together in a near final state is very exciting as well.

28/9/2018 fri

Today was all about testing the finalized project. One of the team leads was fixing the last problems with the project while we set up the testing area. We discussed that I would be joining the project at the Innovation Expo in Rotterdam next week. We put all necessary resources for the expo down and made a list of all of them. Then we put them in bags to make sure we'd be able to bring them all. We then set up the testing area in a new location, a larger one this time. This is when all problems should have been fixed and we could test out in an area as close as possible to the area at the expo. We found a few more issues with the teleport which should be an easy fix.

01/10/2018 mon

Today we were going to start with a new project, Skyracers. I was instead assigned to finish the NS project with Sjef. We spent the day ironing out the last few bugs and testing them. We also had a talk with the person who does UI and visuals about how we need to make the visuals look to make them clearer and more in line with the company's other projects. We decided on a Thumb Up or Down for the voting menu instead of the spheres that come up currently.

We also still need to add a visual that makes the teleportation a bit clearer.
As we were testing we were approached by other company staff about the fact that the prime minister, Mark Rutte, would be coming over on wednesday. He would also be having a look at the project. The problem is that by the time of his arrival we would have departed for Rotterdam already. So we had to quickly change plans and make sure we have enough material on both locations (the expo and the center).
I'm looking forward to the expo as I'll be able to have a look at how users will handle the system, etc. I've been told people often have a hard time adjusting to VR. 2 weeks ago I missed our tour of the center due to me being sick. I find this a bit of a shame as I would have gotten a look at the game that currently serves as a tutorial for VR, making the users accustomed to the controller, etc.
I've been thinking of potential projects I could do with this company during my FBP and one that I find very interesting is perhaps exploring several types of tutorials or methods to give users an understanding of the controls quickly. I already have a few ideas (video tutorial, pictures of the controllers and settings around the centre, a physical dummy controller that tells you what buttons do when you press them, before starting a game you get a controller hovering in front of you that shows what each button does).

02/10/2018 tue
Today me and Sjef were making the Final Final bugfixes for the NS Project. I also prepared some of the VR backpacks for use.
For a few days I had been telling Sjef that the solution he was working on to make the teleportation in one of the levels was counterintuitive and messy. I advised him to make it exactly like it was in another level. Sjef kept telling it was fine and told me not to change it because it worked so whatever. Normally if someone tells me something like that I easily give in and just let it be but for once I thought I wouldn't. I stood up and asked a superior for his opinion and he actually agreed with me and told me to change everything. This was a bit complicated but I got it to work soon after. It was a bit of a personal victory to bring up the courage to stand up against another employee and tell them they're wrong, but it's important to do so that nothing weirds goes into the project.

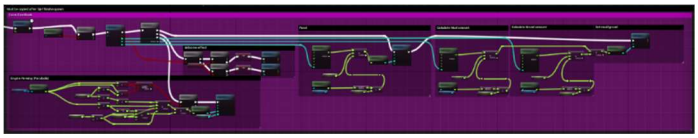
03/10/2018 wed
Today was the day that the project had to be 100% finished and ready for the expo. It wasn't completely yet. Apparently after I left yesterday there was a lot of criticism on some aspects and more changes had to be made. When I arrived I had to prepare the backpacks for the expo by color coding them using tape. This tape had been talked about all week and was finally bought last night. All of these last minute fixes and stress really reminds me of the Demodays at industrial design, everything breaks down moments before. It becomes especially stressful when the program breaks down less than an hour before departure with seemingly no way to fix it. It's all very familiar and kind of funny to see a business like this have the same problems we run into as students. I guess times never change. There were some major problems, but nothing I could prove very helpful in solving (the project wouldn't export), so I was told to pack everything up and get ready to leave.
When we arrived in Rotterdam the stand wasn't ready for us to set up yet. We helped the other workers where we could, and once our area was ready we set up the VR space. It went pretty effortlessly, though it did take a while.

04/10/2018 thu
Today was the day we would head to Rotterdam for the Innovation Expo.
There was a lot of traffic so I arrived about 45 minutes later than I had planned. And so did my colleagues I found out when I arrived. Once at the expo it turned out someone had moved the VR sensors overnight and we had to redo the setup, which took us about an hour. So we started a bit later than planned but everything was good to go. We could tell the people from the NS how the VR experience worked and they could mostly take it from there. We were just around to provide technical support if the game broke in any way.
It was a very interesting day overall. Seeing end users interact with the product I helped develop. It was very similar to my last Demoday, where i also presented the VR game I had developed that year.
There were still a few Issues I saw with the experience, especially since it was made for shorter play sessions.
At the end of the day we sat down with the people from the NS and discussed what could be improved upon for the DDW.
I also pitched in a few ideas regarding the things I saw people struggle with, like the fact that the doors should have *push* and *pull* on them. And perhaps that we should not feature the teleportation mechanic but instead add more levels to cover the entire train.
There's also the issue with the controller. People don't know the controller and have a hard time figuring it out. A lot of people try to point at menus and select things, so I suggested making this work anyway.

05/10/2018 fri
Today I was exhausted from the past two days. We started working on Skyracer, and I took on the task of working with the Force Feedback system of the steering wheel. I aim to make it feel very satisfying to drive the car. I started to think about various events that required force feedback, having a small brainstorm session with myself. I used some of the techniques learned at ID in this. I made a sketch, thought of scenarios. I pictured myself in a car and thought of all the feedback that would give me. Eventually I had a list I could start to work with.
I had to figure out how the force feedback worked and it turned out my options were quite limited to a few preset force feedback effects. With this in mind I adjusted my list and started linking the events I came up with with the effects provided. After that I could start working on the implementation.
I had to come up with a system that keeps track of the materials underneath the car's wheels, so it knows what feedback to give. I couldn't find many tutorials online to help me with this and struggled a bit. The team lead is on vacation and the other employee seemed pretty occupied so I spent most of my time trying things out and looking things up on google, with little result.

08/10/2018 mon
Today I tackled the issues I ran into yesterday and got to work on the force feedback system. First step was to create the system that tracks the materials underneath the car. Turned out I was thinking in the wrong way last friday and had to come up with a different method. I wanted to let the wheels overlap and run code based on that overlap, but I was advised to use line traces to check what material was underneath each wheel.
It took my a while to have it work the way I wanted it to but eventually I worked it out.
Once that was done I could get to the actual feedback part. For this I needed to know how many wheels was touching any of the materials we were using (mud/gravel/asphalt/air). I adjusted my code that checks for materials and had it send values to a script that counted the amount of wheels per material. This worked perfectly and I could move on to the next step, playing the

force feedback effects. I had a lot of parameters that I had already worked out how I wanted to use them, so now I had to write code that used the parameters in the ways I specified.
I ran into a few issues getting the effects to do what I wanted.



Design

- [x] Make a list of all the events which require FF (think terrain types, collisions, airtime etc), write this in the implementation stage as tasks, where needed add the different adjustments that are needed (mud car might have less mud FF) and the priority each FF effect has (does 1 tire on gravel cause gravel FF?)

Programming

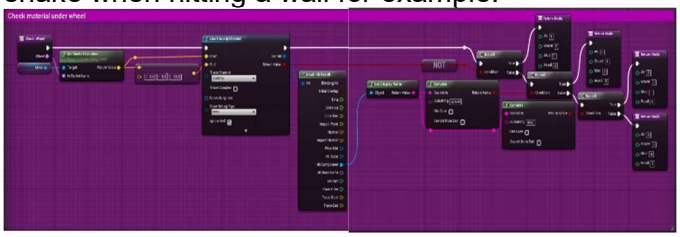
- [x] Create system which keeps track of the terrain type under each wheel

Implementation

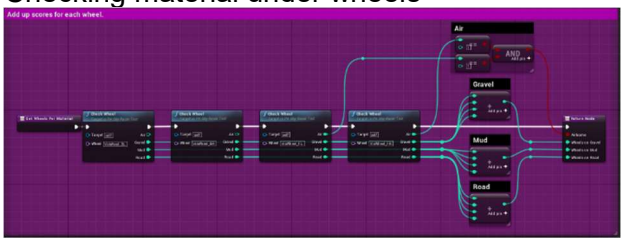
- [x] **Driving on Terrain:** Mud uses slippery road effect. Gravel uses dirt road effect. Asphalt uses surface effect (experiment with parameters to make it feel right, should not be too intrusive). Magnitude based on car type, speed (when speed is 0 magnitude should be 0 as well) and percentage of wheels on terrain type (3/4 wheels touching this terrain means 75% intensity compared to 4/4 wheels, this needs to be tested to see if it feels right).
- [ ] **Impact (general)** uses frontal collision force. Magnitude based on difference in speed between the two colliding actors (the higher the difference the higher the magnitude, take absolute value of difference).
- [ ] **Impact (side)** uses side collision force. Magnitude based on difference in speed between the two colliding actors. Magnitude is a positive number when hit from the right and a negative number when hit from the left. Magnitude can also differ based on car type (Tankier cars getting less magnitude than a lightweight ones).
- [x] **Airtime** uses airborn effect. Activates when both front wheels are off the ground (back wheels don't influence steering). Landing should cause impact (test this and maybe increase the magnitude when coming out of airborn if this feels right).
- [x] **Engine revving** (when accelerating) uses surface effect (experiment with parameters). Magnitude based on acceleration (higher acceleration = higher magnitude). Type, period and magnitude based on car type (ATV would have a lower period and higher magnitude than a formula 1 car).

09/10/2018 tue

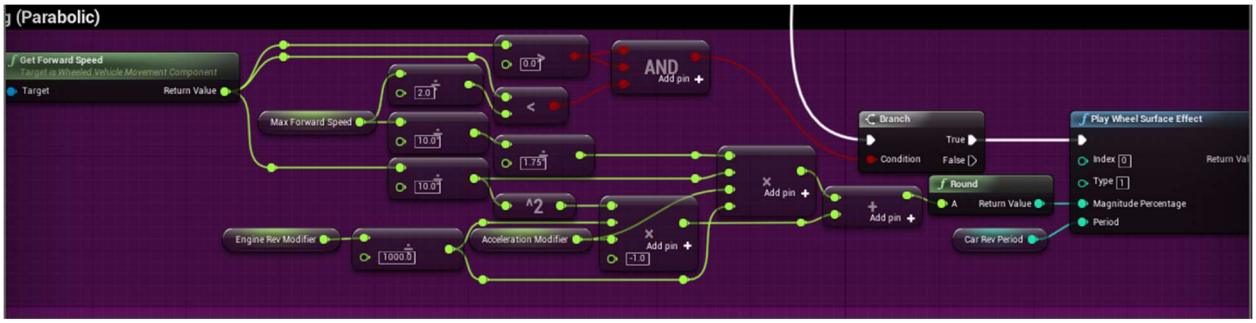
Today I was able to get the force feedback to work. Overnight I came up with some reasons as to why certain things didn't work and tackled them today. One of the effects made the steering wheel very loose, so I had to change the way the effect worked. I also made many adjustments to the code, especially in making it look neater and more readable.
One of the major things I worked on today was the engine revving effect. It involved a lot of tweaking to get right. And a quite complex code. I wanted the intensity of the effect to follow a certain graph. To get it to work I had to do some math. So I actually used math, data and computing. The formula I ended up making was for a parabolic function, that increases and decreases between x=0 and a point I would specify myself (the maximum speed of the car).
It was hard to program this neatly in blueprints. Calculations get messy very quickly.
At the end of the day it all worked and I got to work on collision feedback. The wheel should shake when hitting a wall for example.



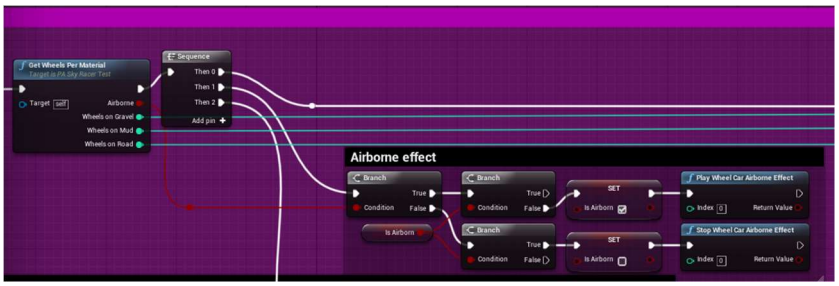
Checking material under wheels



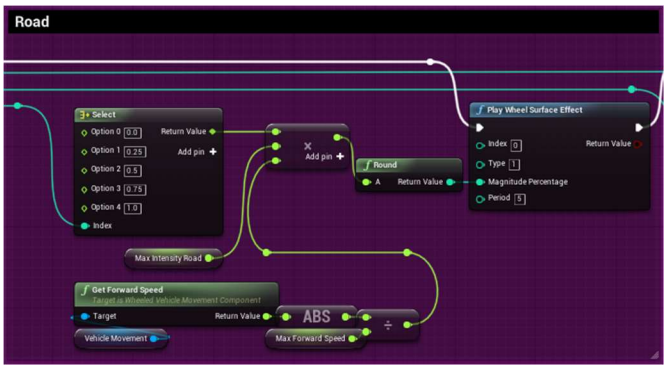
Counting amount of wheels per material.



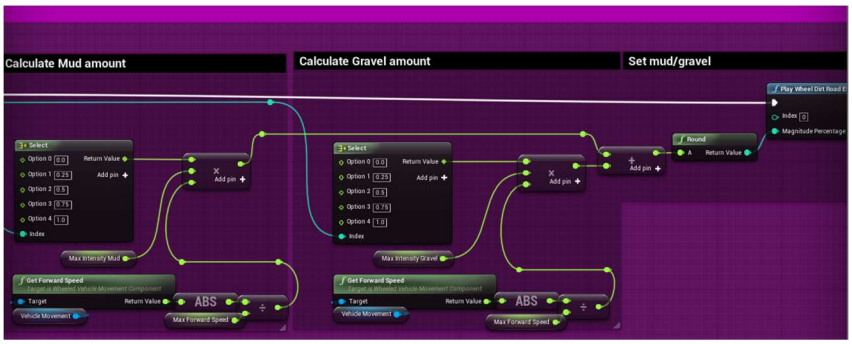
Engine revving.



Applying airborne effect



Applying road surface effect

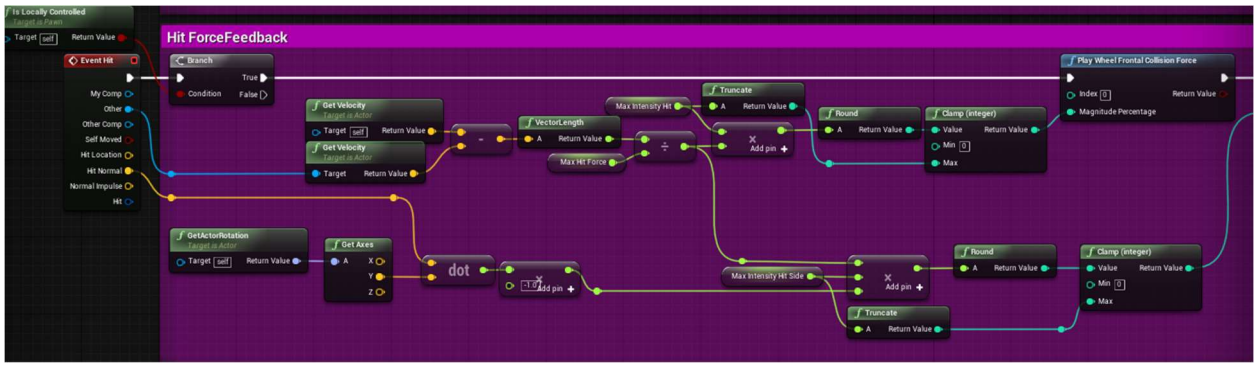


Applying the mud/gravel effect.
This had to be done in one function as they use the same effect and you can only have one of each of the effects active at once.

10/10/2018 wed

Today I continued on the hit feedback. To make it work the way I wanted it to I had to work with vectors. I wanted the collision effect to play at a different magnitude depending on the speed of the car and the object it hit. The speed came as a vector, not a value as I first imagined. This actually made it easier to work with. I could subtract the vector of the two objects colliding from each other and take the length of that vector as the impact force. This force changes the intensity of the effect.

For collisions on the side I had an effect that pushes the steering wheel left or right. I had to detect whether the impact was to the left of the right of the car. The easiest way to do this was with vectors again. Using the dot product with the side axis of the car and the impact vector I got a value between 1 and -1. This was perfect as it not only gave me info on whether the impact was on left or right, but even how much to either direction it was. So I could base the intensity on this value as well. So now I full on side collision plays the effect much stronger than a frontal collision would.

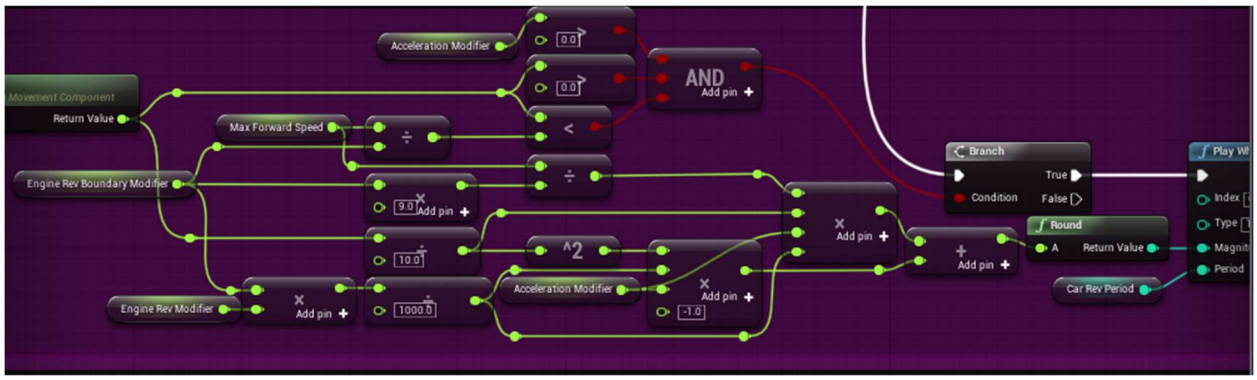


After I finished everything surrounding collision I revisited all of my code and cleaned it up some more. Especially the parabolic graph I made yesterday was still very unclear to me. Looking at the code I had to retrace it entirely to figure out how it worked again.

I used an online graphing tool to give me more insight on how exactly it works.

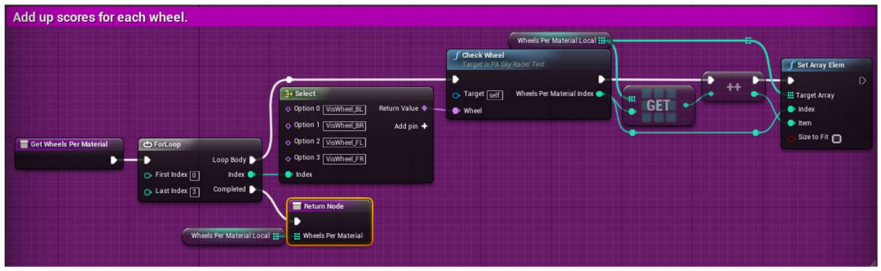
https://www.desmos.com/calculator/wfi3zvopqe

This really helped in coding the graph. The most important change compared to yesterday is that the graph scales along with itself depending on the max speed and some other input variables. To change the way the graph behaves you only have to change some settings variables rather then the code itself.

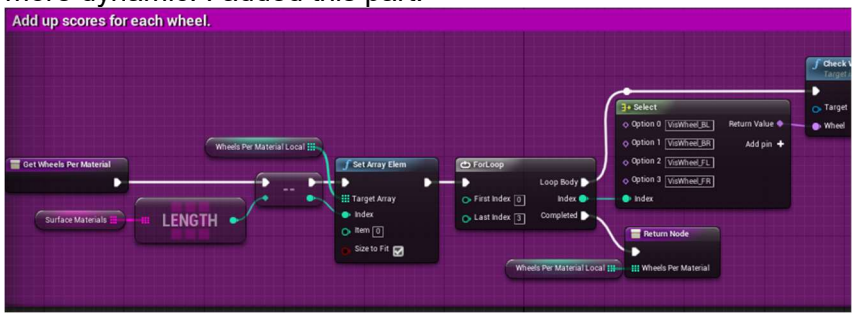


Now all that there's left to do is wait for my colleague to finish his changes to the vehicle so that I can tweak the feel of the force feedback according to his car. He changed the speed variables so that might affect the way my code behaves.

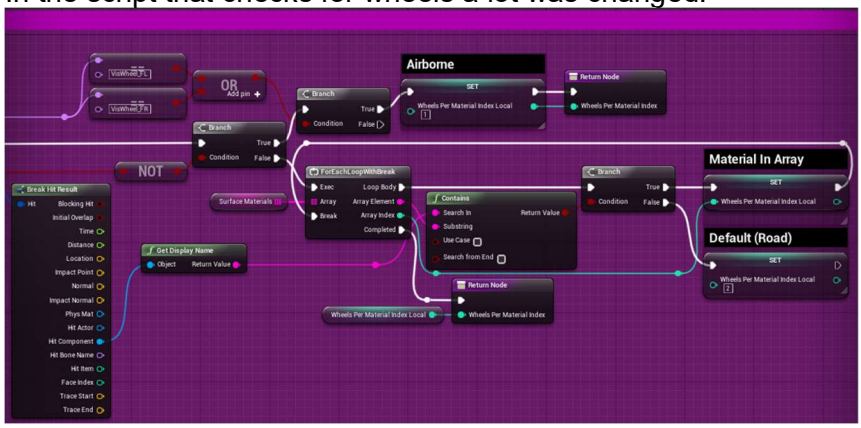
11/10/2018 thu
Today I drastically changed the way my code calculates the amount of wheels per surface.



Instead of sending separate scores it sends them in an array (a list).
This looks much cleaner and is easier to expand upon once more terrain types need to be added.
After doing this I wanted to move on but there was currently no work to be done. So I decided to further improve my code. With the addition of more terrains in mind I went to work on making this even easier, this time in the check wheel script. This script contained a bunch of if statements that would have to be expanded upon every time a material was added. I changed it so that you now have an array in which you can write down all the materials and the rest is done for you. This also meant that the script that counts the amount of wheels per surface had to be more dynamic. I added this part:



Which makes the array of wheels per material the correct length. (the same length as the array of materials).
In the script that checks for wheels a lot was changed.



After determining whether or not the car is airborne (at the top) the script enters a for each loop. This goes through the array of materials and checks if any of the materials correspond with the material the wheel is driving on. If so, it sends the index of that material. If not, it sends the default material index (road). If more materials get added to the array this script will continue to work. All that needs to be done is take the amount of wheels on that material from the array and apply an effect to it.

12/10/2018 fri
Today I had to finish the force feedback by putting it into the game properly. Up until this point I was making it in a test project rather than the actual project.
Sjef had a look at my code and commented on some aspects of it, mainly my method of determining the material, looking at the name of the object we're driving on. This worked in my test build but isn't a guaranteed working system. I changed it so that it actually looks at the physics material of the object it's driving on. This was a minor change to the if statement (branch) in the for loop. Instead of a list of material names it now has a list of materials it goes through. Why I didn't do it this was from the start was because I couldn't find the physics material. It returned NONE. It took both me and Sjef some testing around to find why this is. Once we fixed that everything worked again. Although Sjef did also comment on the way I set the effects, noting I should have sent the material each wheel is on, rather the amount of wheels per materials. I remember wanting to do this but not getting it to work. I'm going to have rewrite some of the code to get it to work that way. It's more efficient than what I have currently. But since my system does work, it's what's going to be used for the DDW for now.
In the mean time I've been put back on the NS project to make some adjustments before the DDW. The first task on my list was to change the menu so that people no longer had to press the menu buttons, simply touching them would be enough.
I struggled with getting this to work properly at first but eventually came up with a decent solution. The only problem was that the menu would also activate when you'd touch it with your body, not just the hands. That was all I was able to get done that day.

This has been a really messy week with our team lead being on holiday. Sjef did not have enough time to attend to all of us, especially when he got a lot of extra work to do himself as well.
A lot of my mistakes were things I wasn't sure about when doing them but couldn't ask for help with because everyone was too busy. I didn't want to waste time doing nothing so continued working and this caused a lot of changes needed to be made later on. Since my code had a bit of a poor foundation.

15/10/2018 mon
This week our team lead was back from his week off.
I continued work on the NS project, but it was pointed out to me that the problem of being able to touch the menu with your body was not acceptable, so I started to change it. I had to find a way to make sure that a hand touched the menu. I spent all day trying to get this to work with little result. At night I had a figured out a way to make my system work.

16/10/2018 tue
Today I tried my solution to my problem, but as I was saving it the team lead told me a much better way to accomplish it. I tested my solution and it worked, but started to work on my team lead's idea anyway. Instead of looking at the overlap of the menu, it now looks specifically at the button component of the menu. This event gives much more information, such as the overlapped component of the object that touched you. I was able to check whether or not that component was a hand and it worked.
Now I could move on with the rest of my fixes. Making the play space fit better and making it more intuitive was next on the list. I adjusted the hitboxes of the teleportation spaces, added extra visual feedback for each level and thought it would be fine. I regret doing this, this was a great moment to take a step back from the programming and brainstorm some ideas for the

transition level. Instead I just added more of what we already had. Which would have been fine, but I could have done much better and really showed my potential there.
I moved on and went to work on my final fix. Having the Archetype menu follow the player. This menu was often ignored, while having players pick an archetype provides a lot of valuable data to the NS. I tried a bunch of things with just code, but couldn't get anything to work. A fellow intern sat down with me and had some ideas of his own but we kept running into the same issues. The system worked, but we wanted to make it so that if the player looked only a little bit to the right or left, the menu wouldn't move yet. Once again, as I was going to bed I had an idea.

17/10/2018 wed
I tried the idea I had last night. Rather than trying to set the entire menu location using code, already adjust the position of the menu so I would only need to rotate it based on where the player is looking. Position was dealt with separately.
This worked but there was a slight issue, at a certain angle the menu would move in the wrong direction. I figured out that the problem was that if the player had a rotation of 0 and the menu of 360, the menu would try to move towards the player rotation, so rotating from 360 to 0, which is the wrong way around. To solve this I used vectors instead of rotations, since these don't work on such a scale.
Tim, my coach and one of the founders of the company had a look at the transition stage and figured out it wasn't enough. So the two of us discussed a better solution, although he came with most of the ideas and I didn't get much input. This is especially why I think I should have thought of a better solution earlier.
Tim made a 3d model of a room which would become the transition room. I was actually impressed by the speed in which he made it using sketchup, I can tell he does it a lot.
With the 3d model in I could make some adjustments to the transition stage and I could get to testing. I set up the VR space for the train project in a room and tested the project. Functionality wise everything seemed to work. All I had to do was make minor adjustments.
At the end of the day all that was left to do was was remove a piece of text I put on the player's screen.

18/10/2018 thu
I started off the day by removing that piece of text and doing some more thorough testing.
I actually found quite a few issues I needed to fix. And took a bit of a bad approach to it. Rather than trying to fix each thing individually I started fixing everything at once.
There were 3 main issues I wanted to fix.

-The archetype menu was visible to the spectator
-You could vote on menus without opening them first
-You were sometimes teleported twice when going through the transition stage.

I overheard we would be setting up the project at 2pm so I tried to finish before then. This made it very stressful. Especially when my solutions weren't working. Also halfway through the day Tim asked if I could add a logo to the spectator screen. This was a small task but just put another thing on my list of things I was doing all at once. It became very messy and stressful. In the future I'll definitely try taking on one task at once instead. It's hard to think straight when doing it the way I did.
Eventually everything was fixed, with some help of my coaches. Sjef pointed out a mistake in my code which solved it. Now everything was fixed. I was really proud of the end result. Even though some solutions were a bit messy (I ended up just hiding the archetype menu from the spectator, rather than actually remove it.
We ended the day by setting up everything at the NS' stand for the dutch design week.
I saw a small issue with the hitboxes which I would fix tomorrow, it was only on the software side of things anyway.

19/10/2018 fri
Today we set up things for the Dutch Design week. It became a very long day (ended up staying until 9:30PM) because some games still had issues that needed to be addressed and other simply took long to set up. Though this was to no fault of my own (probably just some poor planning) I decided to stay and help where I could. It's similar to how things usually go wrong before a Demo Day, funny to see this still happens in a company like this.

20/10/2018 - 27/10/2018
This week was the Dutch Design Week. The entire week ended up being me standing at one of our projects showcasing it to people, letting people interact with it and answering questions when needed. I was mainly stationed at the NS project because I was knowledgeable in it. Much like with the Innovation Expo a few weeks ago, it was very interesting seeing the end users interact with our products. These people also gave a lot of feedback and useful insights, especially on the racing game. This game is going to be added upon in the following weeks, so it would be nice to take people's feedback into account when doing so.

30/10/2018 - tue
Today we helped clean up the space a bit after the Dutch Design week. We also started to work on some minor fixes and changes to the various projects, basically cleaning up the todo list before continuing with the racing game. One of the tasks as well was cleaning up the racing game a bit. Because of the short deadline things were done a bit messy and we had to make sure all was cleared up a little.

31/10/2018 - wed
Today we did more minor fixes to things.

01/11/2018 - thu
Today we were almost alone at the company, our team lead had a day off and the other employee had to go to another expo to showcase the NS project. We were left alone with a list of tasks to do. My tasks weren't very challenging so I worked through them quite fast. I even ended up doing most of the tasks assigned to another intern who was sick.
I did notice I was a bit easier distracted with noone to keep me focussed.
My fellow intern had trouble with his tasks, but I didn't really know how to help him.

02/11/2018 - fri
Today we had to make preparations for the "Major Play Day" we were going to have next monday. Now that the DDW is over the company is getting back into their regular workflow. The major play day is part of that. Every monday we test the games we've worked on the last week. See if the fixes work and if the game feels good. Based on that new tasks are made which can be worked on the week after.
During the day I was asked to set up the NS project so it could be done with some guests on mondays as well. As a test for future projects, I had to set up the space in a 10x10 area.
This took me a little while, setting up such a large space is quite finicky.
After I finished I finished helping setting up the play day. The tasks weren't very clear. It was difficult because it required knowledge of all the projects and how to export them. Eventually we ran out of time and decided to finish things on monday.

05/11/2018 - mon
Today there were some things left to do for the play day. We were also told that we needed to test Skyrunner and Snowdown despite there being no changes.
My fellow intern was a bit upset at the reactions he had gotten to some of his questions last week, and was starting to become afraid to ask questions I had noticed. Both him and I were called over to discuss the things we ran into last week and I brought this up. This led into a few discussions regarding this topic.
While I had a bit of a personal meeting with my coach about this we discussed my progress as well. They think I am doing very well regarding programming. While I have some progress to make in being more proactive, they feel like I do have the correct thought process when it comes to programming and that with enough practice I can be a valuable asset.

06/11/2018 - tue
Today we started work on the issues assembled at the playday. Snowdown got priority as it had a deadline the following week. I fixed a few bugs, such as the game being black and white in some cases, which was caused by the effect not being turned off when the player was destroyed.
I spent the rest of the day restructuring the folders of SkyRacer. During the dutch design week it had gotten very messy and it was up to me to bring it to our default structure. What makes this difficult is that the game references the items in these folders so movin them causes trouble. You have to fix up redirectors every time you move something. This was a pretty tedious process and it ended up failing a few times.

07/11/2018 - wed
Today I continued restructuring the folders. Ater two more attempts I got it to work and could move on. I started fixing the final problems in Skyracers, a few warnings the program gave when trying to build the game. All minor things, and eventually the game had no more problems and we could focus on continuing development.
I then started on a slightly bigger task. There was an issue in Snowdown where players would wait outside the enemy base and kill them as they walked out. To prevent this they had come up with a solution which I had to implement. The solution was a turret that would kill players of the opposing team around the base.
I was able to get the basics down and had a turret that shot bullets at a position in the world. All that was left was for me to do tomorrow was have it shoot players.

08/11/2018 - thu
Today I made the turret target players. It was a long iterative process but I eventually got it to where I wanted. Having the turret shoot at a target in range is easy, what made it more difficult was the fact there can be multiple targets in range and it has to shoot the best one. There were a few things the turret had to take in to account when picking a target. Line of sight, health and distance. I used all these to score targets in range and the person with the highest score would get shot at. The closer the target, the higher the score. The lower the health, the higher the score. If the turret has no line of sight you get a much lower score. This way the turret will take out the easiest or the most threatening targets first.
It was a lot of fun to put this together and see it come to life. I really enjoyed working on this turret.

09/11/2018 - fri
The team lead had assigned me a design task to work on today. The teleport has a cooldown but no feedback on it. I had to make some visual feedback of the teleport charging and audible/sensory feedback for your trying to teleport while it's on cooldown (the teleport failing). I started working on the audio feedback. I figured it needed an error sound like you'd have in a program or something and I started to search for such sounds online. I found a few sounds I was happy with and edited them to sound more like other sound effects in the game. I don't have a lot of experience editing sounds so it took me a while to make it sound good. Eventually I ended on something I was quite satisfied with and implemented it into the game. I made it so that the force feedback triggered along with the sound, so that if would feel as if the sound and feedback were linked. Then there was the matter of visual feedback. The problem with VR games is that you can't make a UI on the screen. The screen is attached to the player and it would feel as if something is stuck on your eye. This is why all visual feedback needs a place in the world. I started with a prototype visual feedback on the hands. Adding a blue bar above the health bar that represented teleportation charge. The implementation wasn't very difficult, I was able to find the variables I needed quite fast. During lunch my colleague told me there was a place on the hand the visual might fit better. There was already a slot in the hand used for indicating things, though it was currently not used for anything. I hooked it up to my code and it worked right away. The only downside being that the slot gets covered up by any gun you grab. I hope that people notice the visual before they get a gun, so that they have a feeling for the cooldown.
Today I also worked on a bug in the game regarding force feedback. While teleporting the controller vibrates, indicating you are charging the teleport. However when you died or the round ended while you were charging, the vibrating wouldn't stop. I assumed this was similar to the problem with the black and white screen. It turned out it was and the fix was also very similar. There was a small difference in execution that I had to figure out but it wasn't anything major.

12/11/2018 - mon
Test day
Left early for moving house.

13/11/2018 - tue
Worked on Snowdown bugs

14/11/2018 - wed
Absent due to moving house

15/11/2018 - thu
Enversedbase bugs

16/11/2018 - fri
EnversedBase bugs
Found out my commits were not as clear as I pictured them being. Depending on the program used to view them they were listed under the issue they were a part of. Without that context a lot of the things I wrote give very little information.
Today I worked together with a fellow intern on an improvement in the framework. The guns were using a lot of replicated variables, which is heavy on the server. It causes the server to have to send a lot of data to all of the clients. It was quite difficult for me to work on this. Partially because I was very tired from moving house this week, but also because I stepped into the task about half way through. My colleague had already done some things and I failed to understand the full picture. It's difficult enough to understand a piece of code, now I had to make sense of something half finished. My colleague also didn't do a very good job at explaining what he had done and what he was trying to accomplish. Still, I could have done a better job here in understanding the problem and thinking of solutions. I'll have to look into it better in the future.

19/11/2018 - mon
Interview for canvas assignment.
Nice insights about my progress

20/11/2018 - tue
Worked on Ns data visualization.

21/11/2018 - wed
Reverting to 4.19 starting on BHV

22/11/2018 - thu
More BHV, getting the fire extinguisher to work.

23/11/2018 - fri
More BHV, getting the fire extinguisher to work work

26/11/2018 - mon
More BHV, particles of the fire extinguisher, start work on fire particles.
Starting to interact more with the art team which greatly improves communication on my part.

27/11/2018 - tue
More BHV, fire particles completely new system. Thought of overnight. Worked with Maps (kind of array) and made a system I'm really proud of.
Worked on doors

27/11/2018 - tue
More BHV, getting ready for demo. Made spectator system, was quite difficult because there's a lot of info being spread around. I was copying from a different project, skipping some steps I thought weren't necessary. It turned out they were and it was an important lesson. I understand the workings of multiplayer and networking though it remains a bit difficult at times.
It really takes some extra steps to realize that the data you're trying to obtain from the server is only present on the client and that you have to send it to the server before you can utilize it there.

28/11/2018 - wed
Final demo preparations

29/11/2018 - thu
Day of the demo in zevenbergen. Went well. Good to meet the stakeholders and nice to see people interact with a product I contributed so much to.

30/11/2018 - fri
Fixing physics of fire extinguisher

03/12/2018 - mon
Fixes + preparing for second demo tomorrow. Started working on the Smoke System, something quite complex.

04/12/2018 - tue
Second demoo, don't promise client too much if you might not be able to deliver.
Teacher coach visit. Main point is to talk about not just the skills I've developed but also make note of academic progress, things I've learnt (for example how to tackle UI in VR).
Team lead gave me some ideas for the Smoke System, said we'd probably start making it together as it's quite complex.

05/12/2018 - wed
Fixed a bug! We though the player spawner was doing something weird but I found out it was the spectator, who also spawns on the player spawner. Which also explained why the bug only occurred occasionally. I figured out it occurred whenever we had a spectator present.
Major fixes on the fire extinguishers, finally worked out how to get the physics to work the way I want them to, they look a lot more professional now and I am very proud.
Made some parts of the smoke system, my team lead told me to let it be for now until he's able to help me do the complicated parts. Until then I should focus on some smaller tasks like issues encountered at the demo.

06/12/2018 - thu
Worked on the Smoke System, getting the basis down so that me and my team lead can continue working on it. I started doing some tweaks to the fire extinguishers and fire system but my team lead told me it wasn't that important to polish it that much as we're just making a demo for the company. He told me to work on some mechanics that make the doors a bit more advanced.

07/12/2018 - fri
Long meeting, working a bit too slow focussing on the wrong things, not knowing deadlines, improving communication.

10/12/2018 - mon
Working on finishing up a task I picked up last week before continuing the smoke system. Was working on implementing the new maps made by the art team. Had a lot of problems in them that are now fixed.
Didn't get to work on the smoke system a lot because I had to leave early.

11/12/2018 - tue
Project had to be cleaned up a lot. A big part of this was because it wasn't clear from the start what the purpose of the program would be. It turns out we need a base game with a bunch of systems. Companies should then be able to pick which systems they want. So we need to make each system modular and independent. Currently a lot of the systems rely on each other and are written in such a way they can't be taken apart. This al has to be rewritten such that they can.
Started working on adding sound effects to the project. First I made a list of all necessary sounds, before I went looking for them. Finding the right sounds was quite difficult but at the end of the day I had all the sounds I was looking for.
I learned a lot about editing these sounds, for instance how to make them loop well. Cutting a piece of audio just anywhere can result in popping noises. Its it's important to cut at a point in the audio where the amplitude is 0. Audacity has a function that finds you the zero points closest to your selection so you can cut freely. I wasn't sure how to add the sounds to unreal yet but left a bit early because I was feeling sick.

12/12/2018 - wed
Stayed sick at home today.

13/12/2018 - thu
Worked on implementing the sounds. I had never properly implemented audio into an unreal project, so I had to get a feeling for it at first. I looked into a different project (Snowdown) to see how it was done there. Based on what I saw I added the raw wav files to the project and turned them into Sound ques. In the sound ques I could give the sounds proper attenuation settings. This allows for realistic drop off of the volume the sounds make.
As I was finishing up the issue I was assigned to implement the Supervisor Interface into the project. The team lead has laid the programming foundation to this and I was meant to add more functionality to it such as starting the match and spectating players among other things. It took some figuring out how the UI worked but I got a hang of it and could get to work.

14/12/2018 - fri

Worked on the interface some more. First tackled all the simple tasks before trying to set the spectator camera properly. This got me a good feeling of how the interface worked and before I did anything really complex. The way I was trying to do the camera did not seem to be working. I had a few ideas as to how I was going to solve it.

1. Have a camera on the player that moved towards the players actual camera (this takes up less bandwidth) but this didn't seem to work because the two cameras were conflicting each other when setting the spectator.
2. Set the camera to that of the player (would create a stuttery view as we had experienced in the NS project)
3. Create a camera actor that follows the player and set the spectator view to that camera actor. (would be easy to reset view to where ever you started spectating)
4. Just have the spectator actor follow the player. (when you stop spectating you'd be at the position of the player, you might still have control over the camera)

I went with the camera actor as at the time it seemed like the best solution to me. I have a tendency to get hung up in a method once I pick it. I was told to do it the first way, when that didn't work I decided to do it in a different way that was still similar, rather than taking a complete step back and seeing what the best way would be.

With my new method I still couldn't get the code to work. I asked the team lead for help and he noticed a completely different problem. The names were not being set, instead they were empty. When no name is given the game is supposed to give players a random name from a list stored in the game. He decided to solve it on his own but couldn't manage to do it before he had to go to a meeting so tasked me with fixing it.

At first I was a bit intimidated, if the team lead can't even solve it how should I?

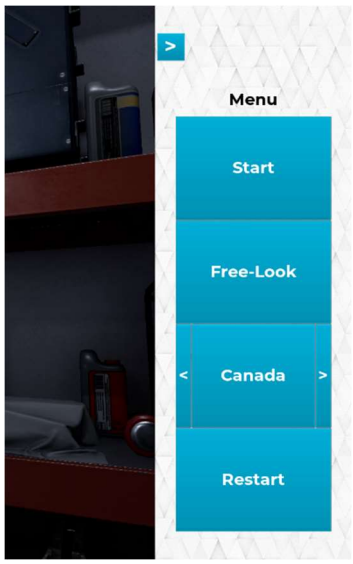
I put some prints into the code that seemed to be misbehaving and actually found out what was wrong. There was an outdated piece of code that looks for a username in a text file. It creates a Map (a type of array) with strings bound to strings (the keys and values of the map). By default, be key exists but it has no value. The script that assigns the names checks if there is a username based on the value existing, so it set the username to its value, which was no value. That's why there were no usernames.

The team lead was very happy I was able to find the problem all on my own. And we ended up solving it by removing the value from the map if it was empty.

12/17/2018 - mon

I continued trying to get the spectator camera to work. I tried a bunch of different methods but nothing seemed to really work. So I dug deeper and found out that the code the team lead had already prepared for me before I started working on this was not working. He was trying to get a reference to the other player locally, but other players' controllers don't exist on your local client. So what I had to do was store the information elsewhere. Once I did this I could retrieve it and figure out how to set the spectator's position. I discussed the pros and cons of my methods with the team and they figured my 4th method would be the most preferable. It would have saved me a lot of time if I had discussed that sooner.

With the basis working and my method decided I started to implement it. At the end of the day I almost had it working. There were 2 minor issues with it that I would work on the day after.



12/18/2018 - tue
I implemented fixes for the issues I had left yesterday, one was being able to see the spectated player's body the other that the camera would keep the rotation after you'd stop spectating. Both were easy fixes on the event that set the spectator target.
I had gotten some new tasks regarding the interface which I had to complete before calling it quits. Currently the spectator would not work in a VR headset was connected to the computer spectating. This had to be fixed by disabling the HMD (head mounted display) at the start of the game, if the player is a spectator.
Another was setting the resolution properly according to the size of the screen, wor which most of the code was provided already.
Now what was left was the visuals of the interface. I made them match the style of Enversed. I took inspiration from the posters around the office and actually found the background image in the marketing folders, so I made it a texture and applied it to the menu.
Now there were just a few minor fixes needed to be done and the menu was finished.

With the menu finished I started work on the Room Actor.
This was a system discussed a few days back. We decided that we needed a general system that kept track of each room. Currently information of rooms was being stored in various locations, mainly the gamemode, and it was getting quite messy. Especially since the future plan is to have different modules of the training, allowing companies to pick and choose between them for what they think they need so it's important systems don't rely on each other too much, but can stand alone as well.
Creating the room actor was a lot of work, but it was not difficult. I had to take pieces of code and move them to this new actor, making sure all the systems referencing each other still referenced correctly and that all systems continue to work. It required an understanding of all the systems, so it's good I was the one working on it, since I made almost all of them or was at least involved with them. This all went very well, on my first test everything worked, none of my changes had seem to have broken any systems. Now all that was left was setting all of the room actors to the objects that required them, in each of the 3 levels.

12/19/2018 - wed
Today I was tweaking values of the feedback on the doors. I started off taking the largest temperature a door could have and set that as the most intense feedback. I realised soon after this didn't work for larger rooms, who wouldn't get to a temperature that high. The largest room in the building hardly gave any feedback, so I had to assure that room gave enough feedback when a large enough fire was present in that room.

12/20/2018 - thu
Working on smoke system, took a while to figure out but eventually had an idea on how to make it work

12/21/2018 - fri
Actually got the function to work. It's quite complicated but seems to work very well. Super happy with the result.

01/02/2019 - wed
Back after the Christmas break, worked on some small tasks. Skyracer menu had to fix all of ben's stuff first.

01/03/2019 - thu
More work on the menu. Changed the way the menu loads the server. Bugfixing helped a lot in learning how to read code and figure out how things work. This is a helpful skill in regular programming as I could come up with solutions making use of existing code efficiently, not only because I knew where to look for the code because I had gone through it when fixing previous bugs but also because I'm better at interpreting it in general.

01/04/2019 - fri
Today I mainly did bug testing to see if the projects are working appropriately. Both Skyrunner and BHV should be finished this week so that we can start on a new project next week while SkyRacer gets pushed further out.

References

1. Enversed Studios: 'home page'. Accessed on January 3 2019, <https://enversedstudios.com/>
2. VRApp and Tu/e: 'Atlas'. Accessed on January 5 2019, <https://vrapp.co/tue/atlas-vr>
3. Enversed Studios and Philips: 'Sleep VR'. Accessed on January 3 2019, <https://enversedstudios.com/portfolio/philips-sleep-vr-360-virtual-reality-tour/>
4. Jesse Schell: 'The Art of Game Design'. Pittsburgh: Taylor & Francis Group, 2015
5. Steve Benford, Boriana Kolevea, Tom Rodden and Gabriella Giannachi: 'From Interaction to Trajectories: Designing Coherent Journeys Through User Experiences'. In Proceedings of the SIGCHI Convference on Human Factors in Computing Systems (CHI '09), New York, ACM, 2009, 709-718.
6. Nuno Afonso, 'Unreal Framework & Network' (April 2015). Accessed on December 12 2018, < http://www.nafonso.com/>