HÖGSKOLAN VÄST

IMC600 24/25

Individual portfolio report

Immersive computing 2024/2025

Name: Maria Anderson

HÖGSKOLAN VÄST

IMC600 24/25

Contents

Exercises Extras

To make playing these exercises run faster and smoother, I implemented them in a new level (I did not name the level, but it is under MyExercises.). This proved a lot less time consuming than implementing the exercises in the original MyExercises level where I had imported the twin motion town square.

In this new level, I have matched the base requirements and attempted some extras for the following:

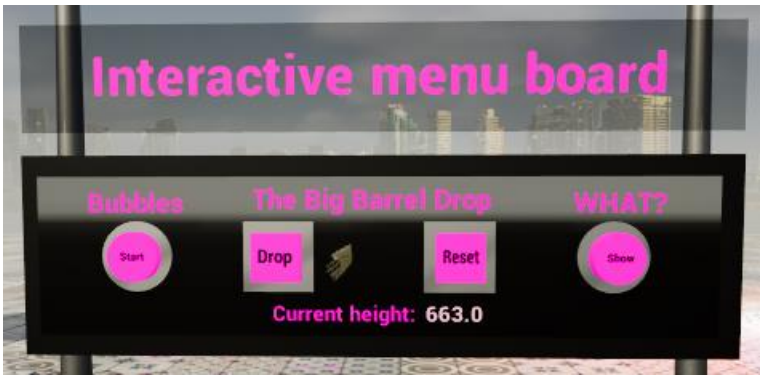
Extras Exercise 3 - Dynamic Virtual Worlds

Create a more complex chain reaction, involving more objects and objects of different types. This entails the first chain reaction as far as the pile of little barrels at the end. It took some experimenting to figure out how to make the whole (hopefully) pile of barrels fall as the big ball hits them and then hide them shortly after so that they would not litter the area. The mini-barrels are separate blueprints, and on collision physics activate and gravity makes them fall to the ground. It does not seem to be working 100% of the times thoug, and the impact of the ball could be adjusted further.

In the second chain reaction that is triggered by entering a Trigger box, when the ball falls down, there is a collision event in the level blueprint that makes the board tilt forwards to send the ball rolling down towards a cylinder, that is then sent rolling up a leaning board (I had to make adjustments here to make it actually roll up and not stop) to hit a barrel. The barrel then hits a trigger box that reveals blocking walls at the end so the cylinder and barrel will not roll away. The walls then lower a little to not block the view as much. The big barrel is based on the same blueprint as the mini barrels, so it becomes hidden with a slight delay after being hit.

Use a Timeline to animate the opening and closing of the door, as in the video by Katie. This was an easy thing to implement. I also included a text render in my door blueprint that slides along with the door.

Trigger some other change in the level with some collision/overlap event, handled in a blueprint actor, e.g., toggling a light, changing its color, show/hide objects, etc. Here I have implemented quite a few things on buttons on the menus.

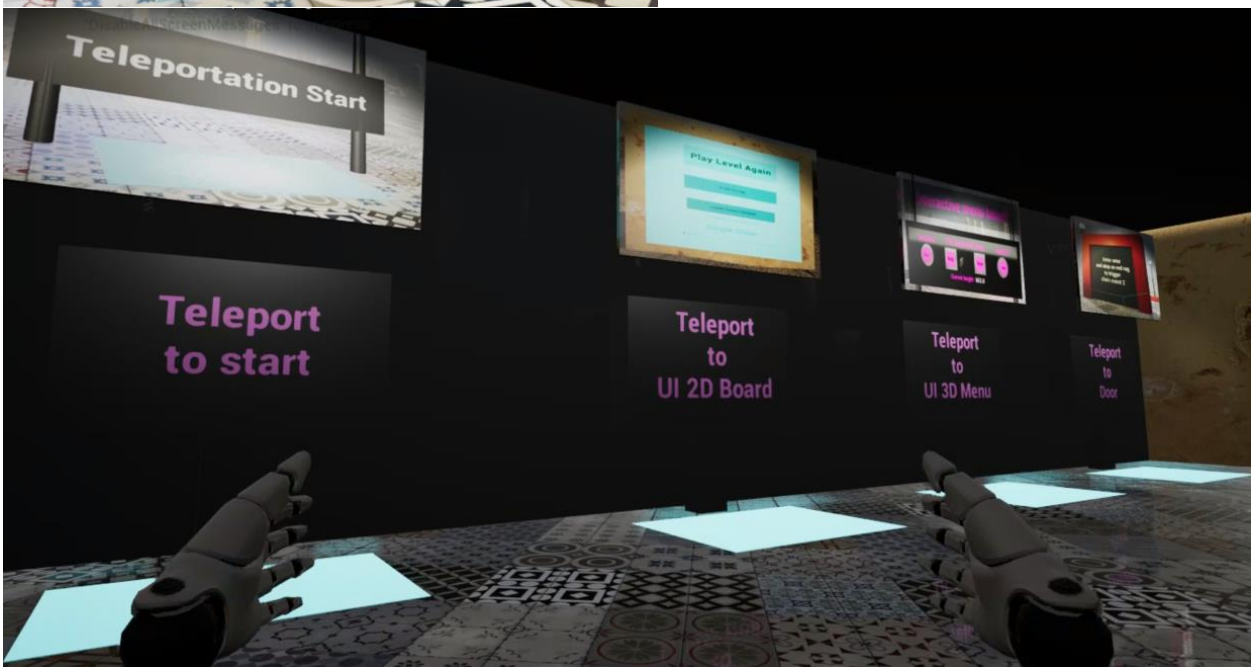


Bubbles come rotating up from the ground as you push one of the buttons on the 3d menu. This took 2 separate timelines and a few discussions with GPT for me before I understood what the different nodes did and how to connect them to accomplish the desired effect. To create the materials of the bubbles I adjusted the settings to make it translucent and then adjusted the opacity and the color. For the middle buttons I took the easy way and adjusted the Variables in the existing blueprint. But on the left side I show and hide three mannequins with lights of different colors shining on them, and the text on the button changes between show/hide. I also animated the mannequins with the existing idle animation. Then I experimented with a pull-lever mockup blueprint that hides and shows a media plate playing an old mp4 video I made in a previous course (*Media för Webben*). The pull lever is not grab activated, but instead runs an animation of the lever moving out and in as you overlap a trigger cylinder in it.

Include a collision event in a chain reaction, triggering a change in the level as part of the chain. This was implemented in the second chain reaction twice. Once by tilting the board as the falling ball hits it,

and once by the barrel colliding with the trigger box that raises the walls at the end of its track. Also. A light turns on over the chain reaction as the ball drops and turns back off after a delay as the barrel hits the trigger box.

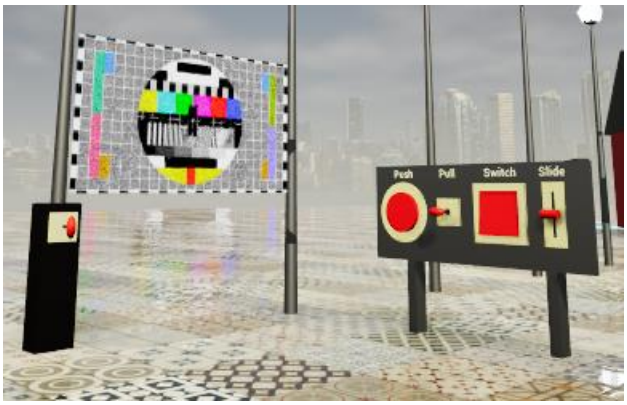
Create a "teleporting hub" with teleporters for at least 4 different locations in your level. Make sure they are labeled. Here I made rectangular opaque floor-pads to represent the teleporters. I used the original blueprint from exercise4 to make the teleportation function properly. The first point of teleportation is labeled with Teleportation Start and it has a pulsating light shining on it. The following are labeled with text renders, and you can move through them in sequence with a **"hub"** by the fourth location, to give you alternative travel destinations. In the hub the teleportation destinations are marked with both text renders and images.



Extras Exercise 4 - Basic VR interaction

Create a significantly larger control panel, or several control panels, controlling several things in the world with a sensible layout for interaction in VR (with hands).

a. Including additional mockups of sliders etc Here I was a bit uncertain what was required, but I did make a mockup menu board of different possible interactions in the style of the second chain reaction in the level. I created something that could be made to function as a slider , a knob that could be made to turn, a pull-lever that could be made to move out and back in, and a flipping switch, meant to work like a real life light switch in changing its angle to flip up and down with a flip/flop node as it is touched. Those are all non-functional though.



I did, however, use the pull lever by the mediaplate next to the mockup menu-board. I and animated it to move as you touch it and to show and hide the media plate with an overlap event. I also made a separate slider that I made into a blueprint, located next to the first door. The slider if not functioning properly though.

Create a new Blueprint actor that can be picked up and dropped with the provided (simulated) VR hands, via the GrabComponent, but that also has a light component that turns on when grabbed and turns off when dropped. Here I used the teapot I scanned with RealityScan. Grab my teapot and it turns on a spotlight in the bottom. I tweaked the center of gravity some so that it is more likely to land bottom down when you drop it again and the light turns off. It is located on the ground next to the 2D UI menu board.

HÖGSKOLAN VÄST

IMC600 24/25

Connect the 2D UI to your world.

a. Make something change in the world when pressing a button in your UI. I added three buttons and a slider to the 2D UI.

The first button restarts the level from the beginning. The second button and the third button are made as separate blueprints that I then added to the 2D UI. One prints a string and **turns a light on** over the board. The other one was adjusted to count the numbers of times pressed, *like in the original example*. I did create my UI from scratch though. The slider functions in sliding fourth and back along the track, but is not connected to the world.

Train ride

a. Create a platform or cart that moves along a path with you in/on it. Movement can be triggered by a button or by simply entering it. I created a traveling cart with opaque walls on three sides *(hoping to decrease the sense of VR sickness by somewhat narrowing the area of vision)*, a roof and a pink pulsating neon light below the floor that gives it a hoovering visual effect. It can be started either by pushing a button next to it and then stepping in to it *(You have 3 seconds to get in, or maybe send one of the grabbable items for a ride?)*, or you can just step in and make sure to touch the button at the back. It is guided from turn-point to turn point around the center square by a vector track timeline, meant to give you a fairly smooth ride. It also rotates at the turning points with the help of another float track on the timeline. Unfortunately, I have not yet had a chance to test it in VR for real though. **It is guaranteed to give you VR-sickness**.

Theoretical reflections

Based on the literature seminars and course project seminars.

Literature seminars

Question:

What are the consequences of using avatars in a learning situation in VR? How do Avatars function and what do they make possible in educational environments? Explain examples from the articles as well as other examples that you come up with.

According to the article “Investigating educational affordances of virtual reality” (Ke, Pachman & Dai, 2020) from the first seminar that in caught my interest. The presence of an avatar can improve the learning curve as it enhances engagement of students (Grivokostopoulou, F., et.al, 2020). Ke, Pachman & Dai’s (2020) paper also points to the fact that the knowledge acquired in the presence of an avatar is more extensive. One reason for that could be the increased immersion in a learning environment close to reality. This allows for putting theoretical and often abstract knowledge into practice, aided by the pedagogic virtual agents. One example from the article was how the agent followed the students and

guided them through different steps working with technical issues, where the student appreciated the sense of presence. I would imagine this could be compared to having someone by your side that can be of help if you get stuck, even in real life.

With AR developing quickly, alongside VR, the thought of a possible AR agent, and the possibilities it would open up for are intriguing.

With an embodied presence readily available as an Agent, I can see possibilities also for psychiatry as a lot of ailments appear to be caused to some extent by loneliness. Agents could then potentially be used as a virtual social company, leading to increased virtual social interaction.

Question:
How could applying flickering affect a learning curve?

During the second seminar I stumbled upon an article about flicker augmentations for visual guidance in Augmented reality (Sutton, et.al., 2024). They researched how flickers could guide users to look at the right thing to interact with. With this guidance my assumption was that the times would decrease for how long it would take to perform a task or learn something, but according to this article, the flicker did not seem to make a significant difference in the time spent on performing a certain task, although they did find from registering eye movement, that the users located the objects faster. I am still inclined to think that in a different setting, like one in combination with an agent/avatar, it could prove beneficial since they did register the faster focus of attention accomplished by the flicker. This to me, still seems promising.

Although they also mentioned that rapid flickers could be obtrusive and risk hiding things, my conclusion is that it should be possible to catch someone's attention with a more subtle, opaque, pulsating flicker. This made me think about the pedagogic agent's/avatars presence and how it could be combined with this flicker somehow.

Course project seminars

From the literature we have studied throughout the course, and my previous interest in ways of implementing VR and AR in practical work environments, so people will get good use of this modern immersive technology. I was driven to a project to work with that could possibly contribute to current research with ideas or even practical implementations through a prototype of some sort.

Being that the presence of a pedagogic avatar has potential to make students learn faster (Ke, Pachman & Dai, 2020), it got me playing with the thought of building a prototype that somehow incorporated that.

Sutton, et.al. (2024) mentioned rapid flicker could be obtrusive so my conclusion was that it should be possible to catch someone's attention with a more subtle, pulsating flicker. This made me think about the pedagogic agent's/avatars presence and how it could be combined with this flicker at a subtle and

less intrusive way somehow and still reach the result of a possibly faster learning curve in training simulations.

When attending the PhD Defense (Andersson, 2024), it conveyed a sense that the users of that prototype had not been quite clear as to what they should do and how from the start, like turning on the radio for example. Here the guidance of a present pedagogical agent/avatar that could guide the user could have been of help. Maybe at a first session, watching the scene play, with the avatar doing the actual tasks, followed by a second session where the Agent is present and points you in the right direction by movement or spoken words followed by a subtle pulsating flicker. In a third session the flicker could disappear and then in the fourth they would be on their own. Then I started thinking about what other scenarios this could be applied to.

Also, I found an article by Tubis, et.al, (2023) when locating articles for our first project seminar. It was about training train personnel/Conductors in using a lift for wheelchairs, so they could get on the train. It caught my attention, as I was searching for articles related to Training in VR and Conductors. Here they had like a poster, or a sign, filled with a series of instructions on an in-game floating screen. It looked cluttered, and the user was supposed to refer to this to see what the following instruction was. This is another example of when an Agent/Avatar and a flicker that would clarify the specific item to interact with, could have contributed to a clearer learning scenario and might even shorten the learning curve. This article also confirmed there is a need for training certain scenarios in VR as those train wagons with the lift are constantly on the go.

We wanted to do something that could in some way contribute to this project and possibly be applied in a real-world scenario, if developed further. Knowing about the ongoing pedagogic avatar-project, including amongst others, SJ and Tenstar Simulation AB, I contacted Amir, who recently researched, and wrote a paper about the use of VR for training train personnel in handling fire-emergencies (Haj-Bolouri, et.al 2024). He recommended we meet with Darmin at Högskolan Väst, to find out more about that project.

After speaking with Darmin, the thought of how our prototype could look started visualizing.

Presenting our project concept at the last project seminar, also meeting a representative(Daniel) from Tenstar, along with Darmin, Zacharias and Our mentor Daniel, provided us with some of the missing pieces and also confirmed that we were on the right track.

Course project

The Unreal Engine project is on Perforce

Tobot, the friendly pedagogical agent/avatar, guiding it’s apprentice with instructions enhanced by pulsating lights or flickering overlays aiming to decrease the learning curve through a first time training-session has taken up all of my time last few weeks.

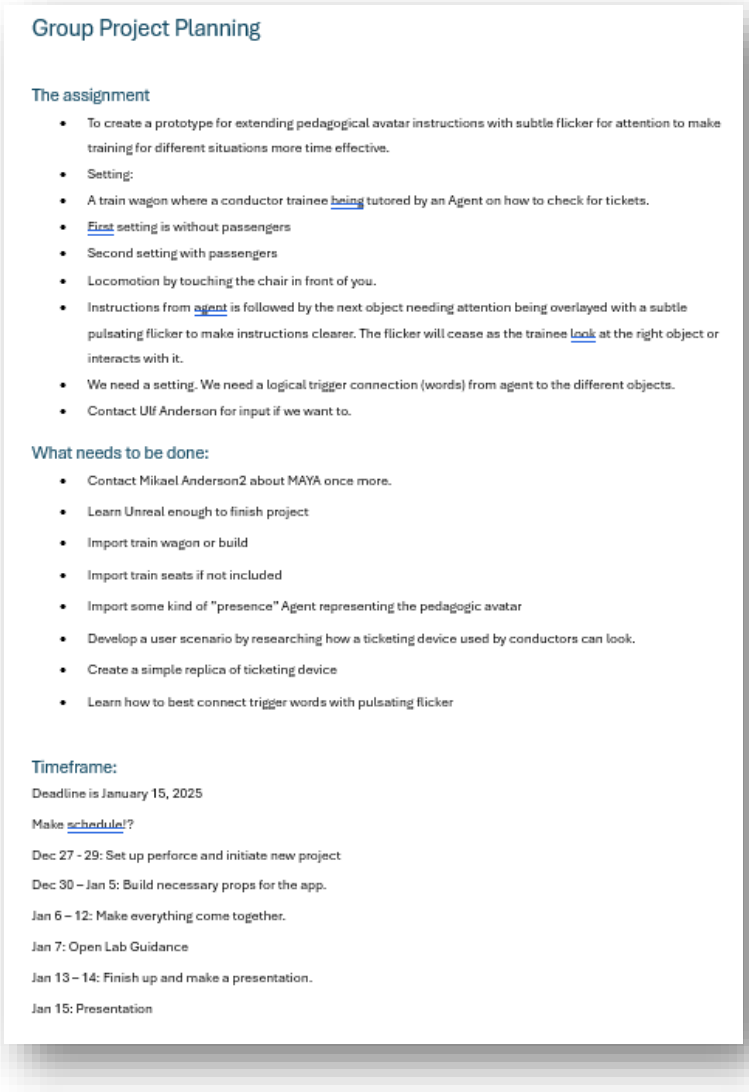
Individual Contribution

Our group project grew, from when we first started discussing it, to when we were supposed to have the prototype finished. At first it was more a vague idea that grew out of articles we had read and presented during the seminars.

The work

Working on developing this prototype was delayed by holidays and illness from my side, But about a week late, intensive work started in learning how to both use perforce and how to develop basic scripts in Unreal engine 5.4.4, since I had not used either before starting this course.

First though, I started by trying to make a time estimate with a list of things that needed to be done.



Group Project Planning

The assignment

- To create a prototype for extending pedagogical avatar instructions with subtle flicker for attention to make training for different situations more time effective.
- Setting:
- A train wagon where a conductor trainee being tutored by an Agent on how to check for tickets.
- First setting is without passengers
- Second setting with passengers
- Locomotion by touching the chair in front of you.
- Instructions from agent is followed by the next object needing attention being overlayed with a subtle pulsating flicker to make instructions clearer. The flicker will cease as the trainee look at the right object or interacts with it.
- We need a setting. We need a logical trigger connection (words) from agent to the different objects.
- Contact Ulf Anderson for input if we want to.

What needs to be done:

- Contact Mikael Anderson2 about MAYA once more.
- Learn Unreal enough to finish project
- Import train wagon or build
- Import train seats if not included
- Import some kind of "presence" Agent representing the pedagogic avatar
- Develop a user scenario by researching how a ticketing device used by conductors can look.
- Create a simple replica of ticketing device
- Learn how to best connect trigger words with pulsating flicker

Timeframe:

Deadline is January 15, 2025

Make schedule!?

Dec 27 - 29: Set up perforce and initiate new project

Dec 30 – Jan 5: Build necessary props for the app.

Jan 6 – 12: Make everything come together.

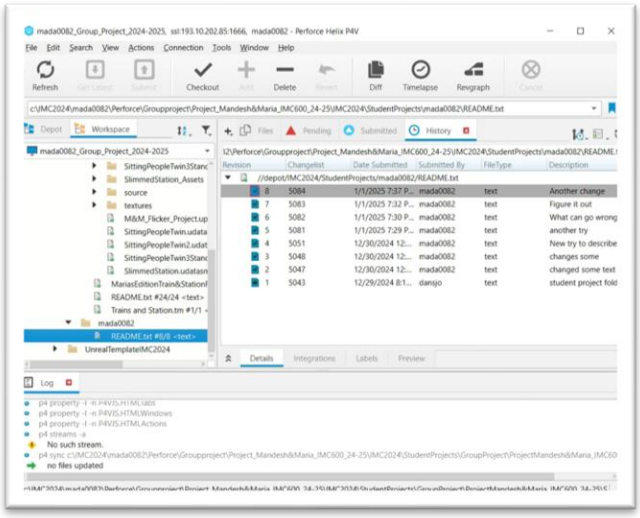
Jan 7: Open Lab Guidance

Jan 13 – 14: Finish up and make a presentation.

Jan 15: Presentation

Perforce

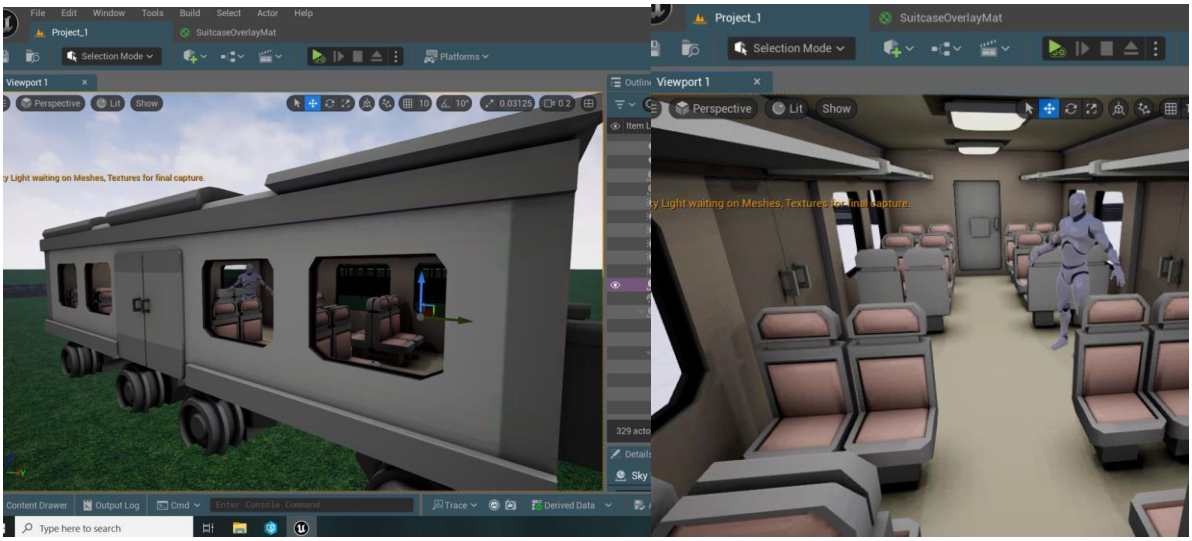
Perforce showed to be a challenge when two people who has never used it was going to work on it together. I am happy to have learnt how to use it good enough to debug some of the more common erors and even help my course collegue to solve some of them too after I learnt how to.



Making sure you are in the correct directory to start with, since my file path mistakenly had become very long, I had an issue with that at one time. Also learnt not to submit the 2000 Saved and Intermediate files at every submit and get them back when you get latest. This was something both of us did wrong a couple of times.

Starting our project

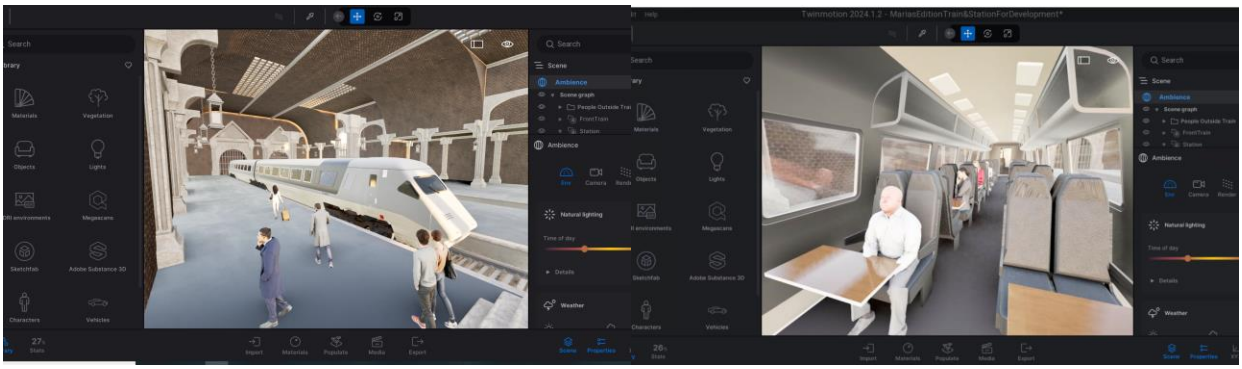
When first looking for train models, we aimed for something realistic that would seem believable as a Swedish train. First, we searched the Epic store for free assets and found a train each, but no luck. We both then managed to find a few in Scetchfab that could have been ok. They came in pieces that had to be puzzled together. Although they did not either quite give the right atmosphere we aimed for, since the cooperation was with SJ in thought as our user, we wanted a more Swedish atmosphere.



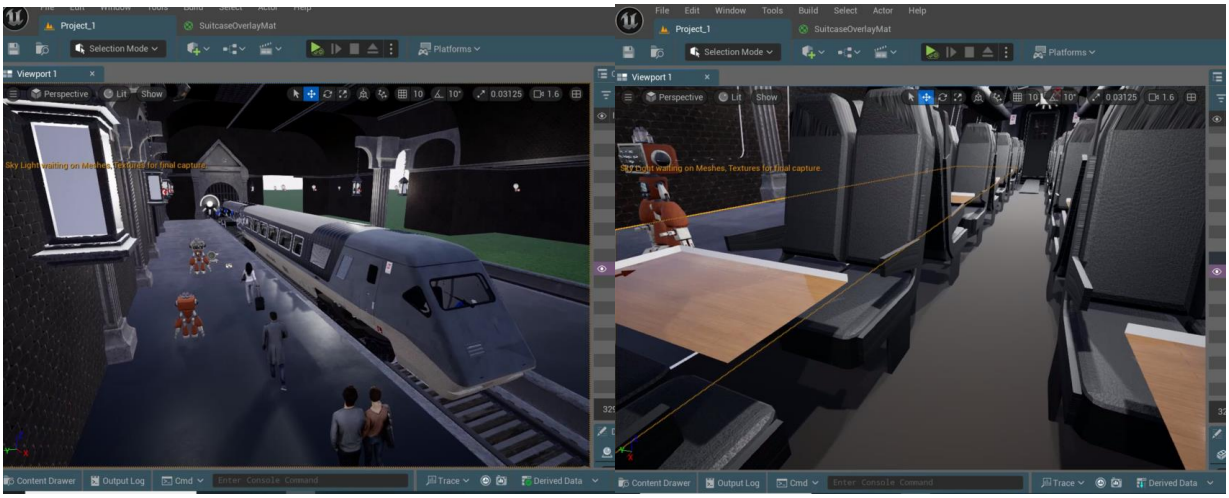
HÖGSKOLAN VÄST

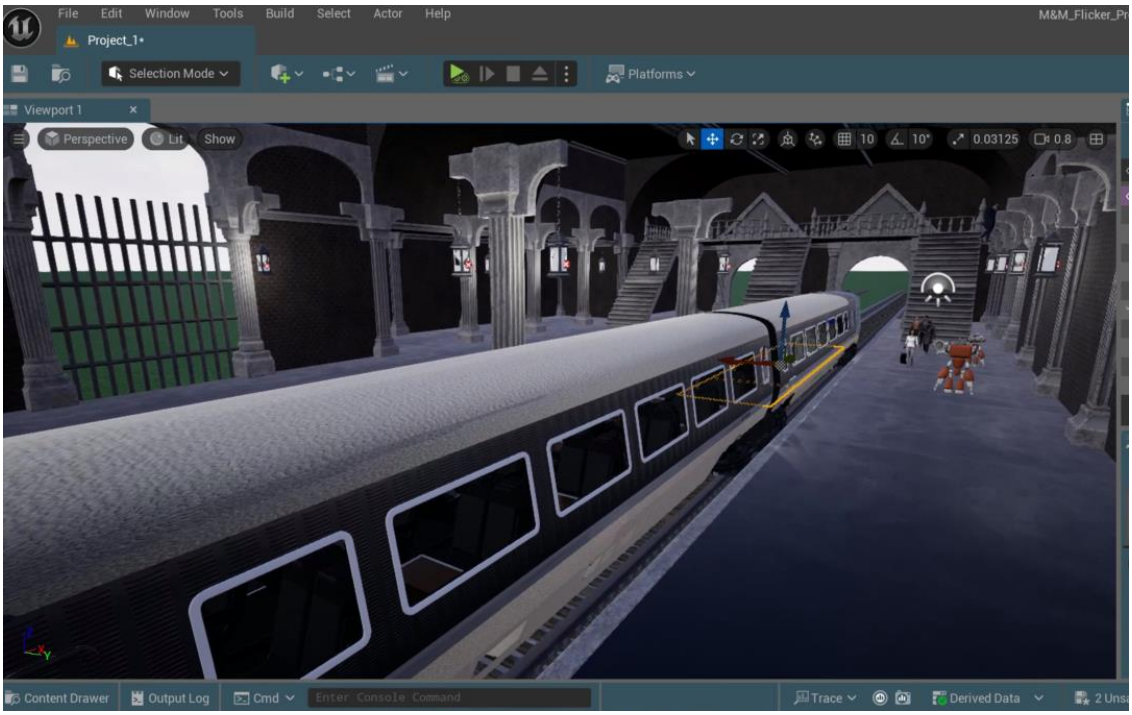
IMC600 24/25

We decided that my course colleague would try to build or find trains in Twin Motion that we could export and then import as a datasmith file.

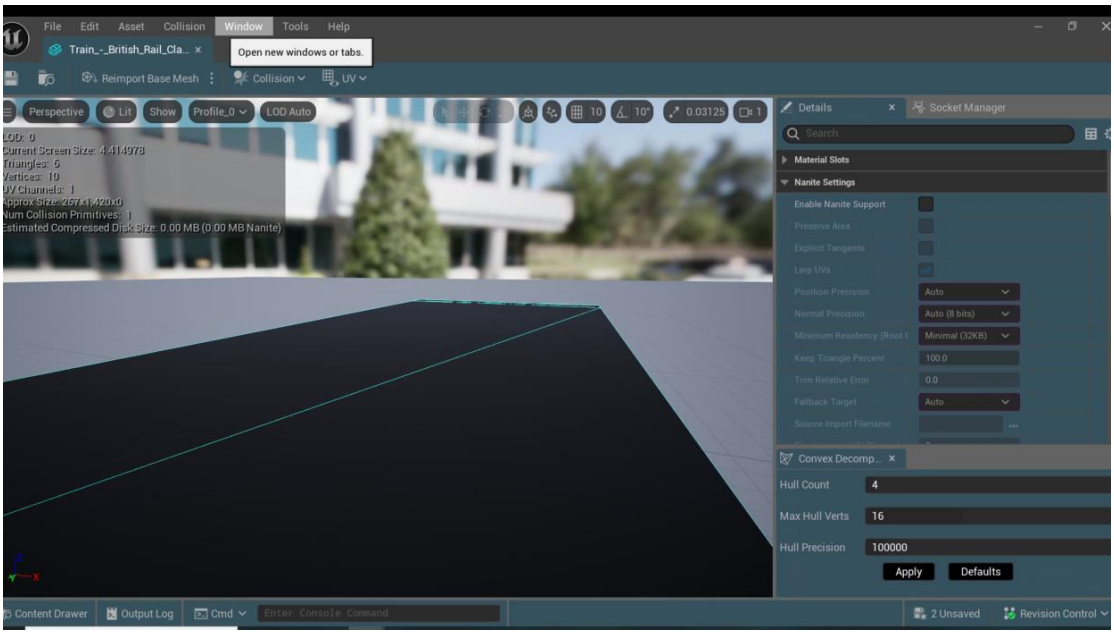


It became problematic importing the datasmith file into our project though, as colors went missing and UV-maps overlapped. Resulting in a very dark impression and one-way transparent walls, plus a lot of wasted time trying to fix this or somehow working around it.



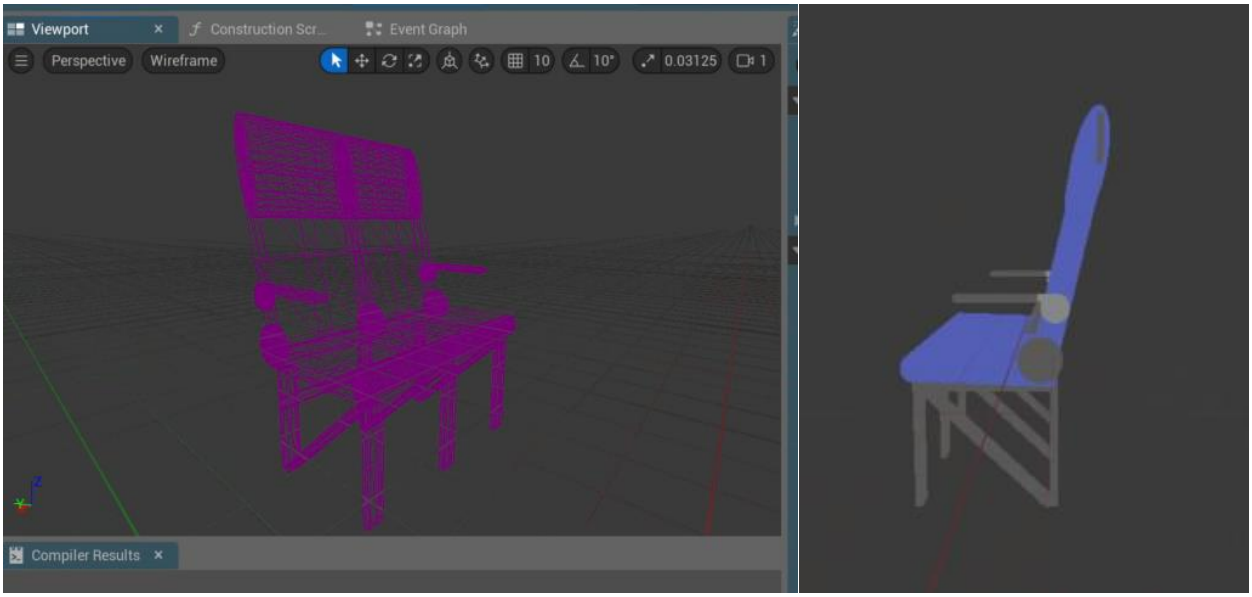


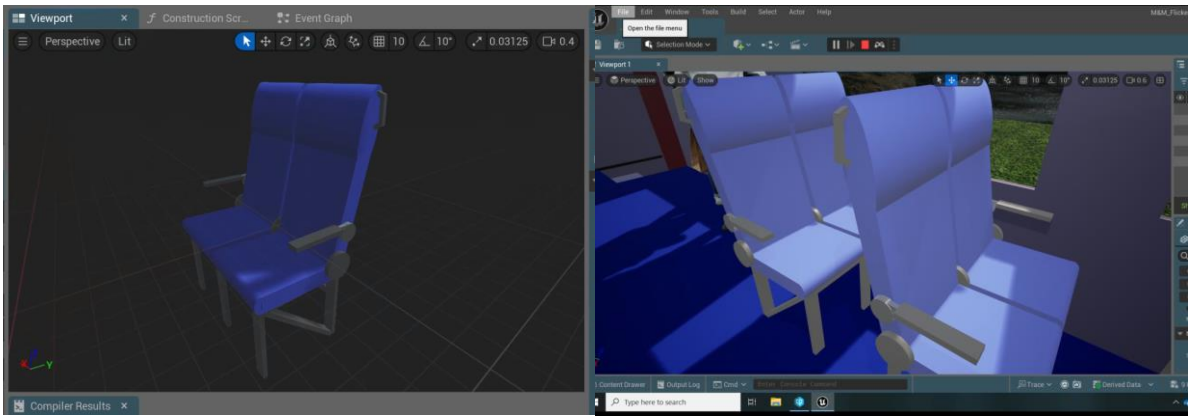
Also, the collision was missing on all the meshes imported from Twin Motion so I added them manually where they were needed, which also took quite some time.



My own special contribution

Considering the time spent on trying to find and import models of trains and the train station with people, and it still not working well, I decided to start a new level where I could just build a train out of simple meshes in Unreal Engine. I started with building the chairs. I worked on this new, simpler Train setting for a few hours, a couple of days in a row, before I was satisfied.

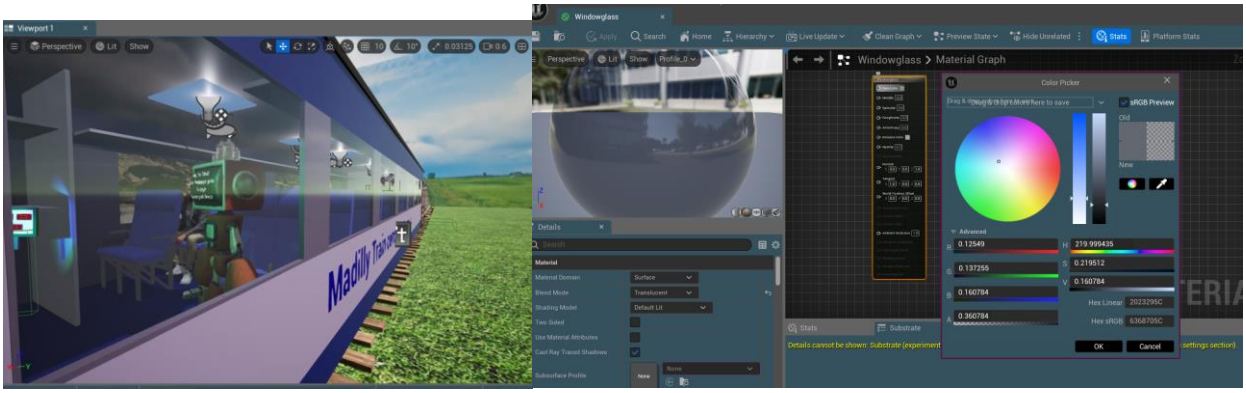




I then multiplied my newly created chair in pairs after grouping them together and made double rows of chairs and a pair of tables. I made new materials to get colors I thought worked well together, but using blue shades was a deliberate choice, originating in the assumption of blue projecting a sense of calm and trustworthiness. It is also a color that will make red details stand out a little more, considering the overlays in red.

Then I built the shell of the train wagon

The walls, floor roof ceilings, wheels and doors. Followed by Interior luggage racks and windowpanes along the sides of the train. The tricky part was to get all things to snap to where it was supposed to. Maybee I could have fine-tuned the controls a little better. For the landscape I just wanted a quick fix to give it more life and remembered how I learnt to make leaning panels with images giving an illusion of landscape, when I took a course making games in Unity a while back. I showed My course colleague how this could be accomplished, with the first mesh boxes with the image of the railroad track, then left the rest to him. My aim was to make a realistic enough environment to where you would feel naturally immersed as a user.

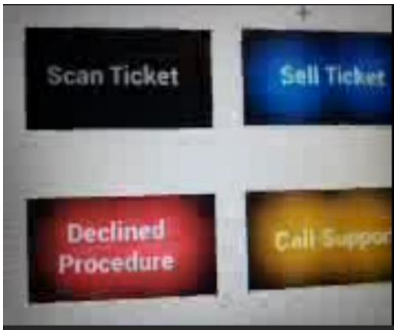


Making the window glass meant making a material translucent and then opaque, while adjusting the color and metallic to look just right. I changed the settings after watching a video and consulting chat GPT. I also used Chat GPT as a *"Bollplank"* and asked for eli5 explanations of what the different settings meant and how they could be used.

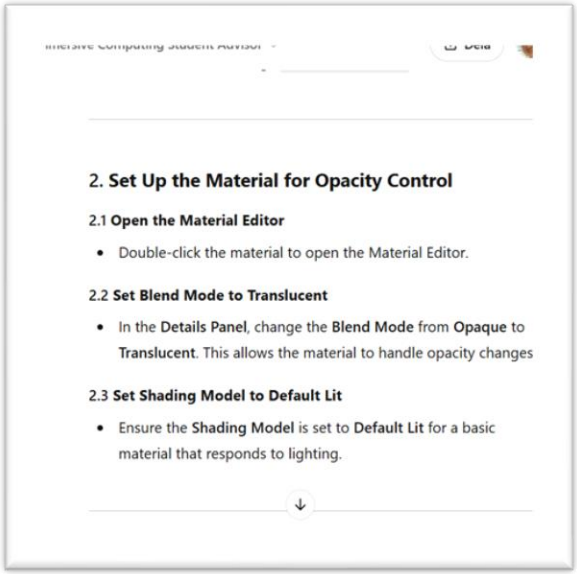
Pulsating Flicker

While starting out the development of the pulsating flicker that was going to be used, I was recalling what Sutton, et.al. (2024) had written regarding the flicker. It can catch attention fast if it it a fast flicker although a more subtle flicker could feel more appealing for the user and not be as obtrusive in the environment and therefor not breaking the immersive experience more than necessary. I first

implemented the flickering by shining a spotlight, that was attached to a timeline adjusting the brightness of the light. I also tried them in different colors discussing with my course colleague what would be the best color to use. After considering green, white, red, yellow and blue I chose a more neon-red color.



What I really wanted was to create a pulsating overlay though, so once more I consulted chat GPT. Often the replies seemed short and sweet but *also taking for granted you already have previous knowledge* of some kind, so I would typically ask follow-up questions regarding the step skipped, or the directions would be for a different version of Unreal Engine. You can never just take for granted you will get a correct and working answer even if you think you have written a detailed and clear prompt. And it also comes down to being familiar with the actual layout and language used in Unreal Engines blueprints.



2. Set Up the Material for Opacity Control

2.1 **Open the Material Editor**

- Double-click the material to open the Material Editor.

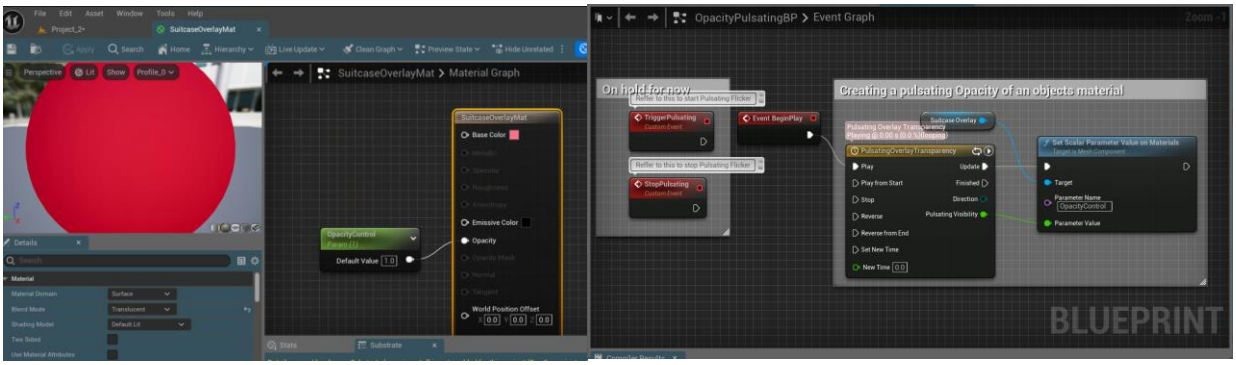
2.2 **Set Blend Mode to Translucent**

- In the **Details Panel**, change the **Blend Mode** from **Opaque** to **Translucent**. This allows the material to handle opacity changes

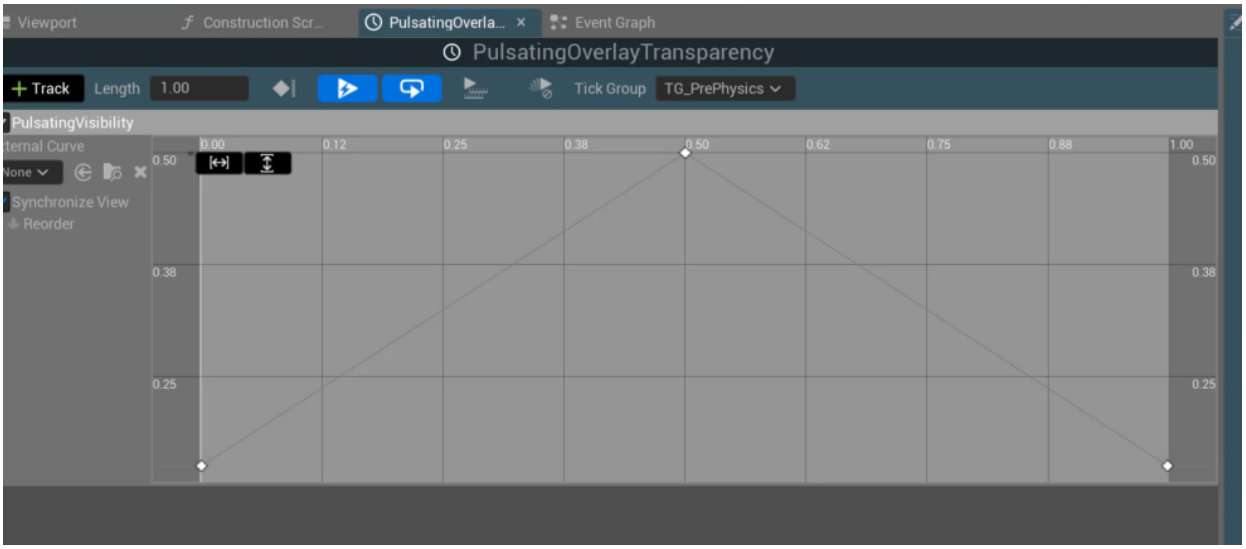
2.3 **Set Shading Model to Default Lit**

- Ensure the **Shading Model** is set to **Default Lit** for a basic material that responds to lighting.

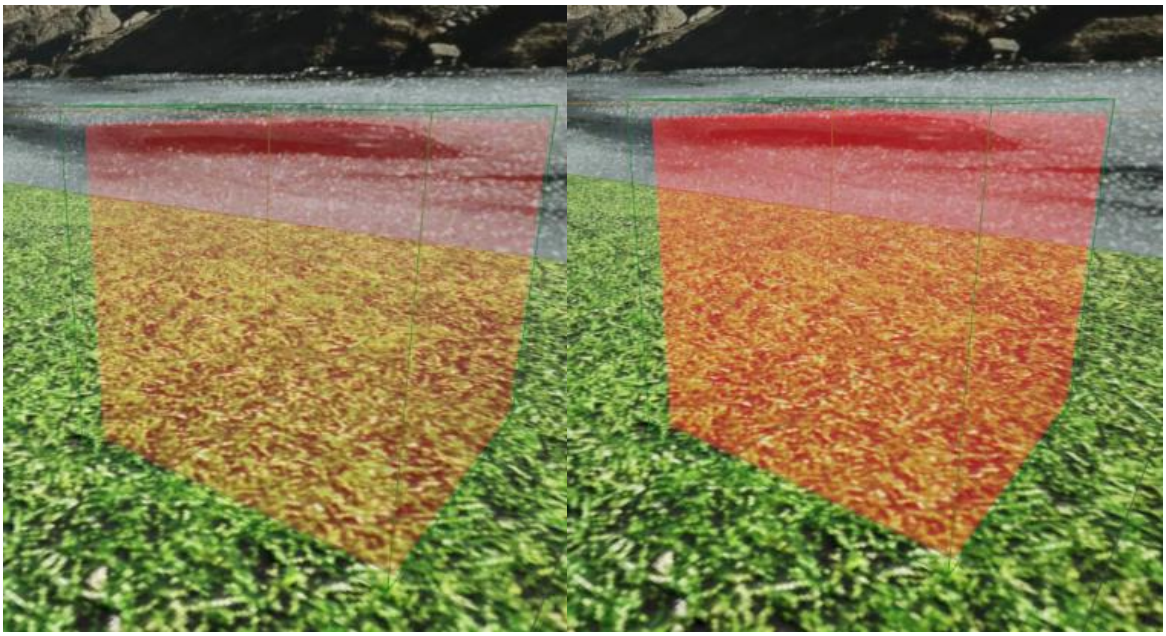
After learning how to do, the following became the material for my first pulsating overlay.



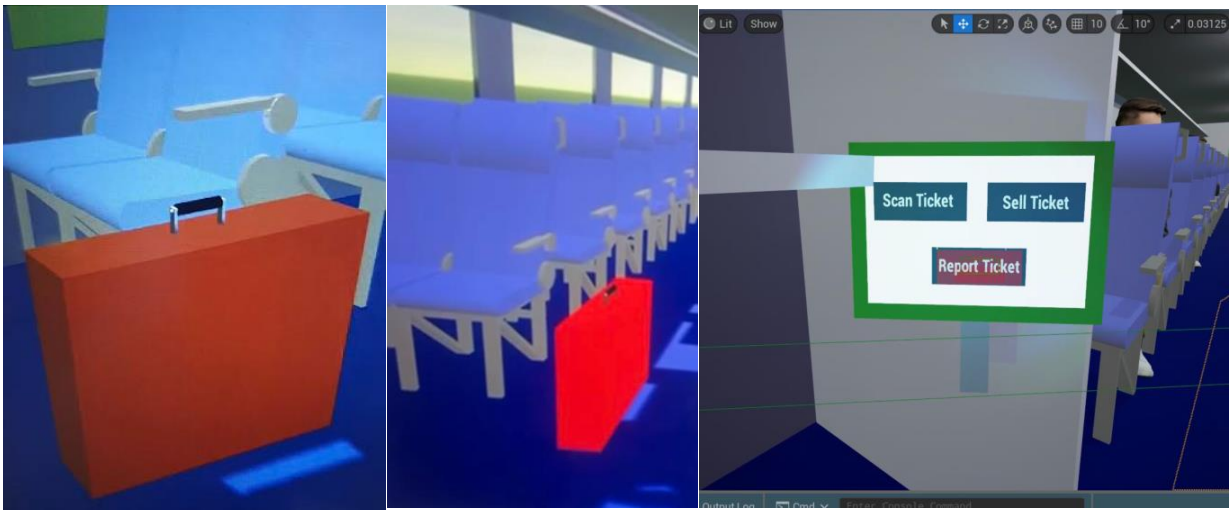
The timeline was a one second long, repeated loop, starting on the lowest level as to not seem to suddenly appear out of nowhere.



The material then had to be combined with a mesh to set the opacity and make it an object in the scene, resulting in something close to the following:



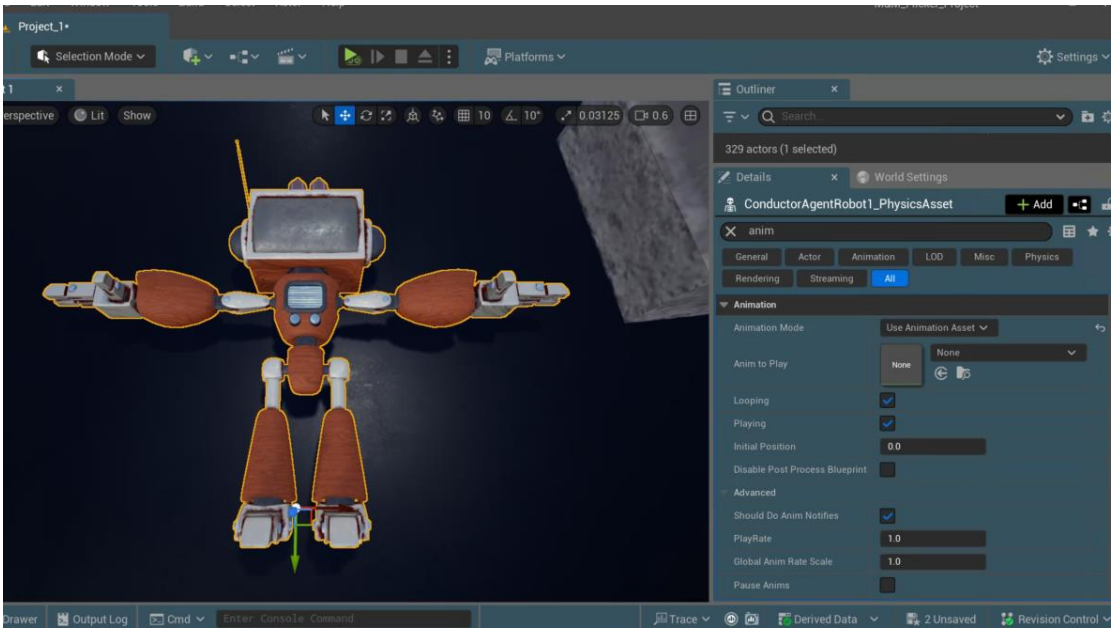
After making the overlays I started developing the overlay for buttons on Ticketing device, card, tickets, phone and suitcase, starting with the suitcase. The first thought was that if the player touches the surrounding color, then it will become hidden again, so it will stop pulsating. Another thought I had was to get another event to turn it off with another opacity timeline letting it fade away as you get closer.



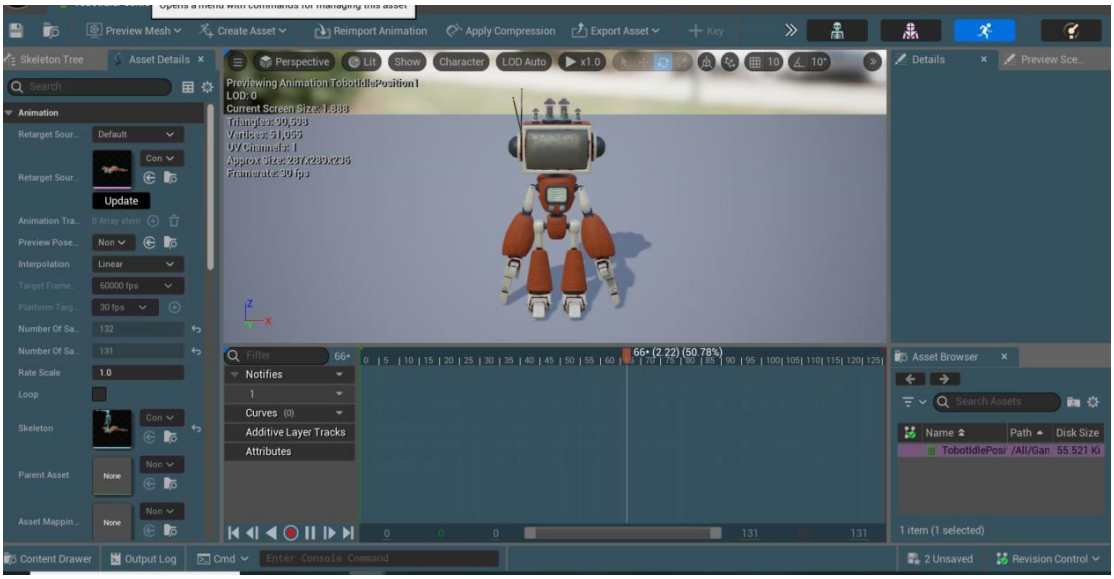
Tobot the Pedagogical Precence

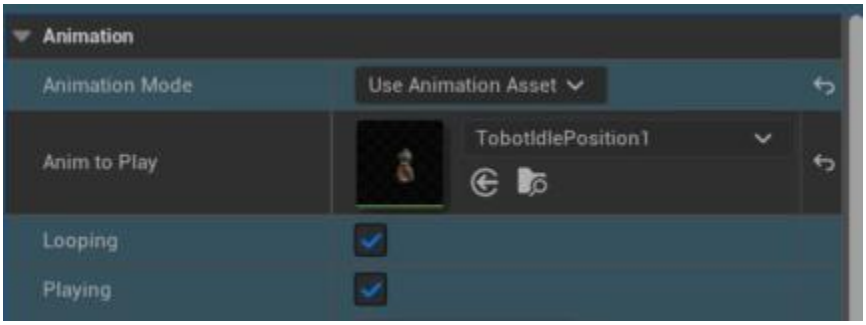
Considering the article by Ke, Pachman & Dai, (2020) we wanted to create the sense of a pedagogic avatar being present. Since this was also something discussed at the last project seminar it was in our initial plan. The need for a pedagogic agent/avatar to give directions made us consider options like meta humans, different humanoids and other life forms. I then once again stumble upon a solution when

browsing the Epic store for free assets. It was a robot with an old retro tv for a head. I wrote my course colleague and happened to misplace my finger, so Robot turned into “Tobot”, and that became his name from there on.

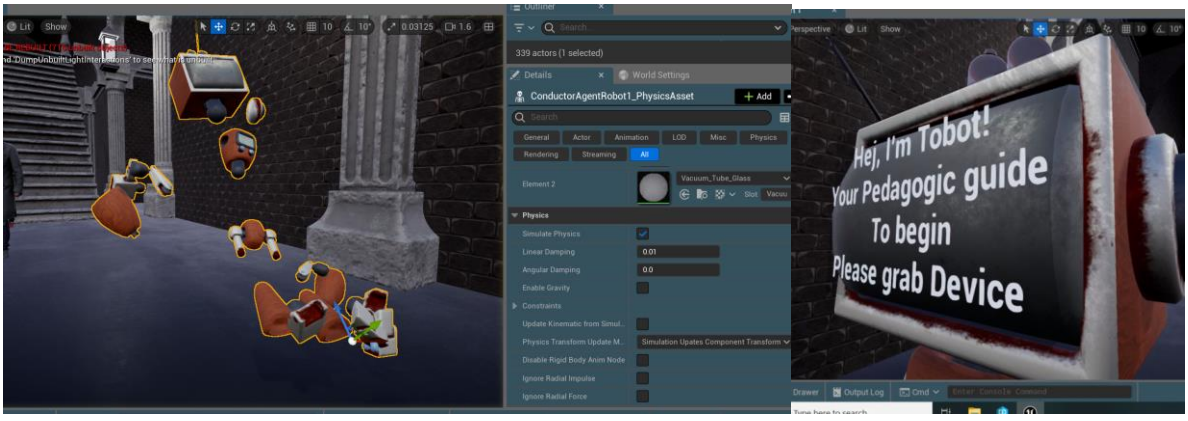


Tobot, comes fully animatable, has a skeleton and comes as a character lying flat on the ground. To make it stand in a more suitable position I googled and found out that I could animate a sequence of the robot standing still after having positioned it the desired way. Positioning was hard though, since all parts had one common pivot they rotated around, but after some time I managed to get it to look ok. I need to mention that I have never animated anything like this in my life. The closest to it is little “sequences” in the corner of my math book that made a stick figure look like he was bouncing a ball when you flipped the pages fast.





Another issue we ran into with Tobot was that he fell apart at play, but un-clicking simulate physics, that issue was solved. Tobot became the perfect representation of a pedagogic agent/avatar for our project, since Meta humans would not work for us and we did not have a soundcard on the cloud computer to be able to work with somehow making recorded instructions, the display was perfect for visualizing the sequence of messages.



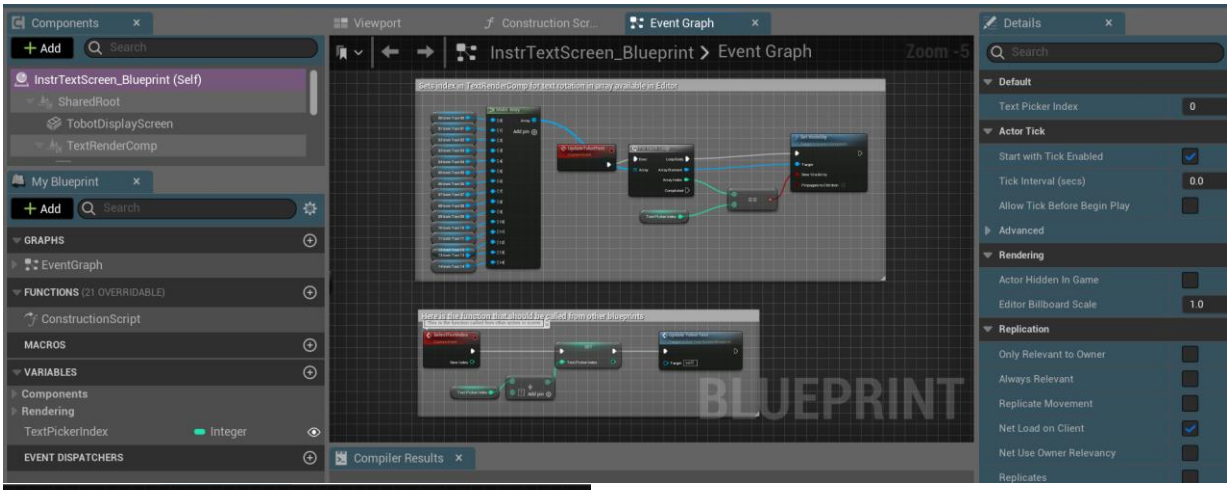
HÖGSKOLAN VÄST

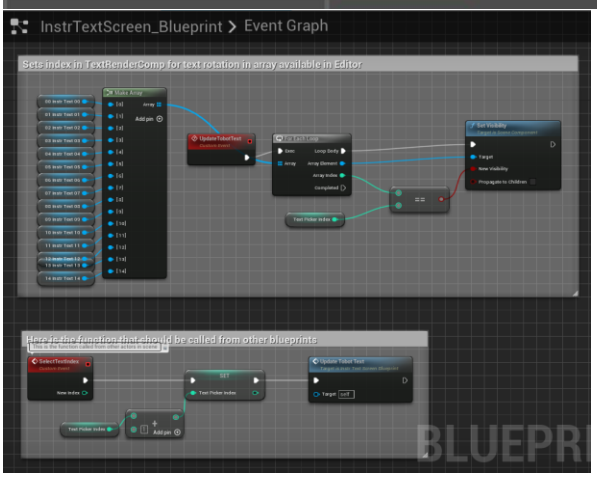
IMC600 24/25

Tobot's Communication

Making a sequence of 13 messages and making them into a blueprint together with an opaque background turned out well for our needs.

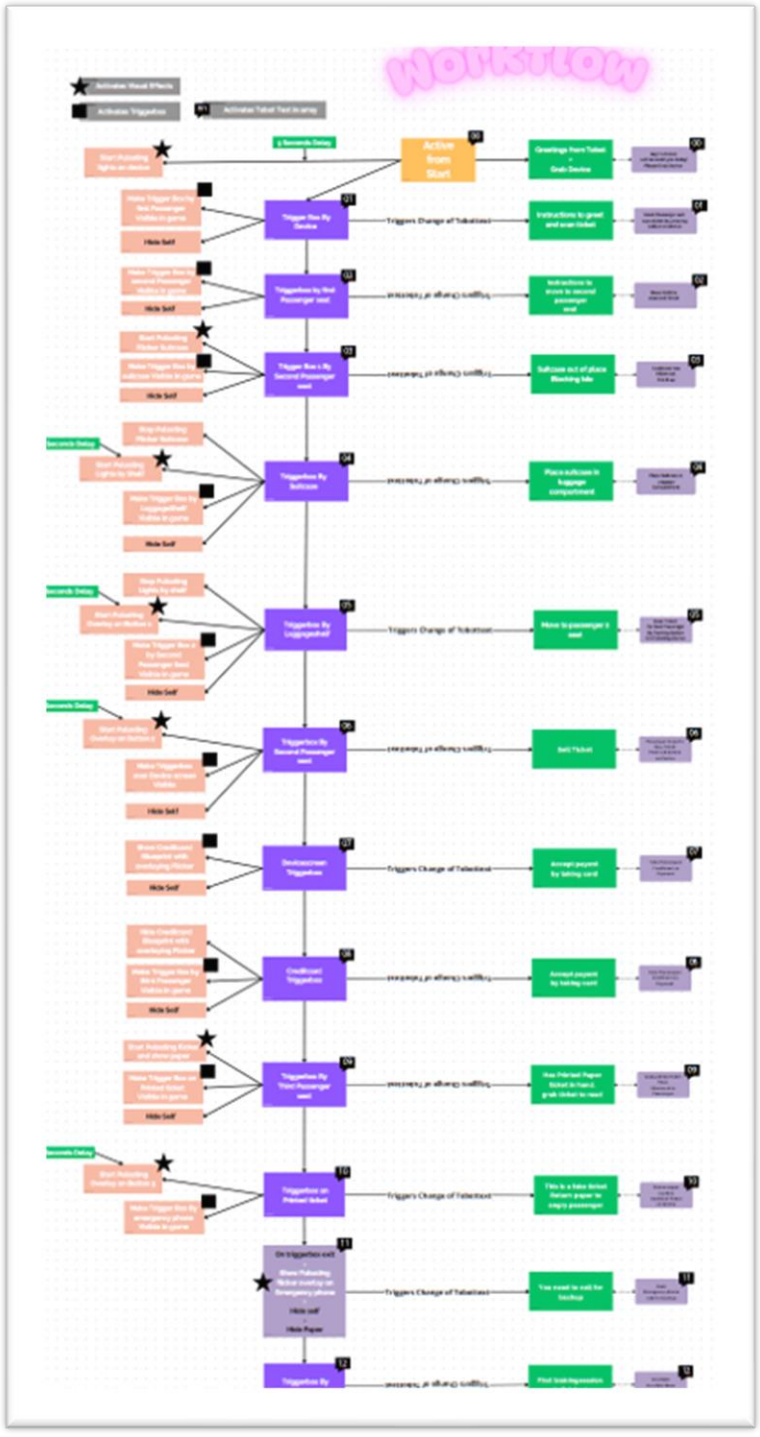
When working in the blueprint though, the text renders would not be in the right order on the left sidebar no matter how I tried. I got some help from one of my proficient teachers, to help me figure out how to make the text-renders loop, but still not showing in the order I wanted them as they looped. Next, I had another long chat with Chat GPT regarding how to make an array in Unreal Engine 5.4. made it eli5 everything… even what had already been eli5:ed. Eventually I managed to combine some knowledge of my own with my teacher's knowledge and some of what I had learnt from Chat GPT, to make an array that worked well. At least until I realized I wanted some more text and had forgotten where the details were that I added the extra index spaces for the array list after pulling the text renders into the array in the blueprint.





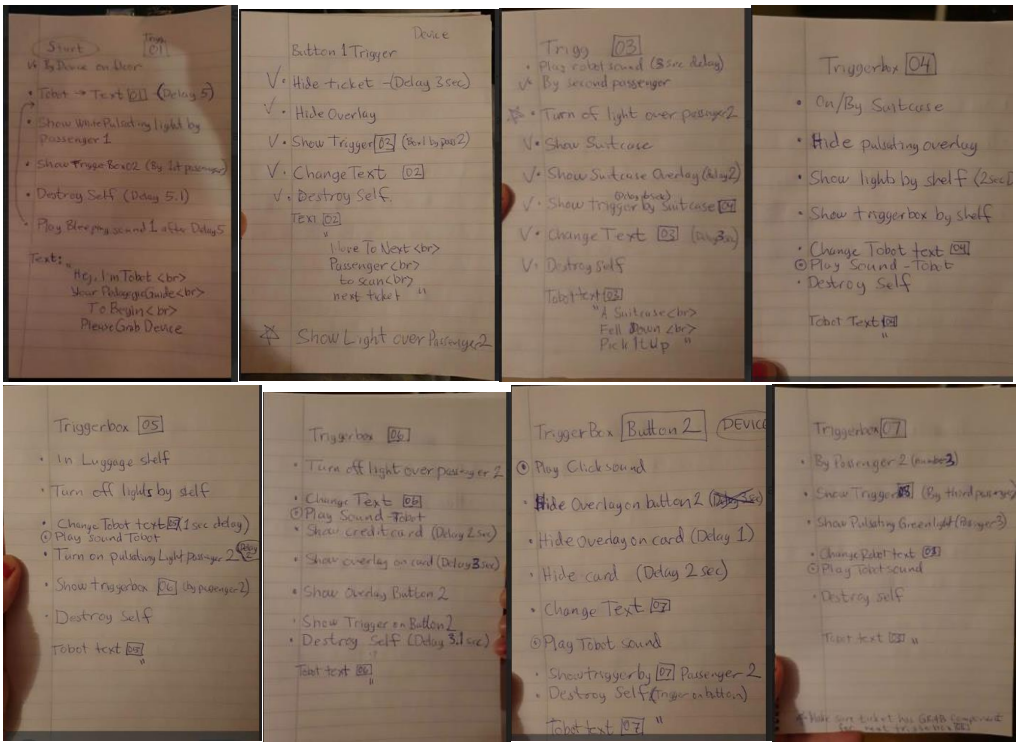
Storyline Planning

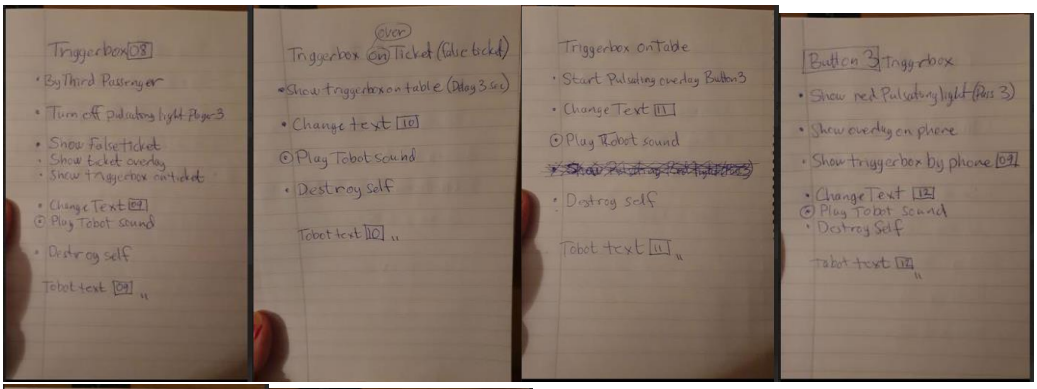
At about this time, I realized there was no way to do this without getting more into depth on exactly what needed to be done, where and write it down so both my course colleague and I could both access the information and edit it. We needed a storyline to make the prototype around. Now, in hindsight, I wish we would have tried making a simple storyline in the beginning of starting our project out, since it could have saved us, or mainly me, a lot of time on the blueprint programming. We (97% by me) used Canva for preliminary visualizing the workflow. I have gotten a bigger understanding why a lot of developing firms have these huge whiteboards to collaborate their ideas and input on. That would have been nice.



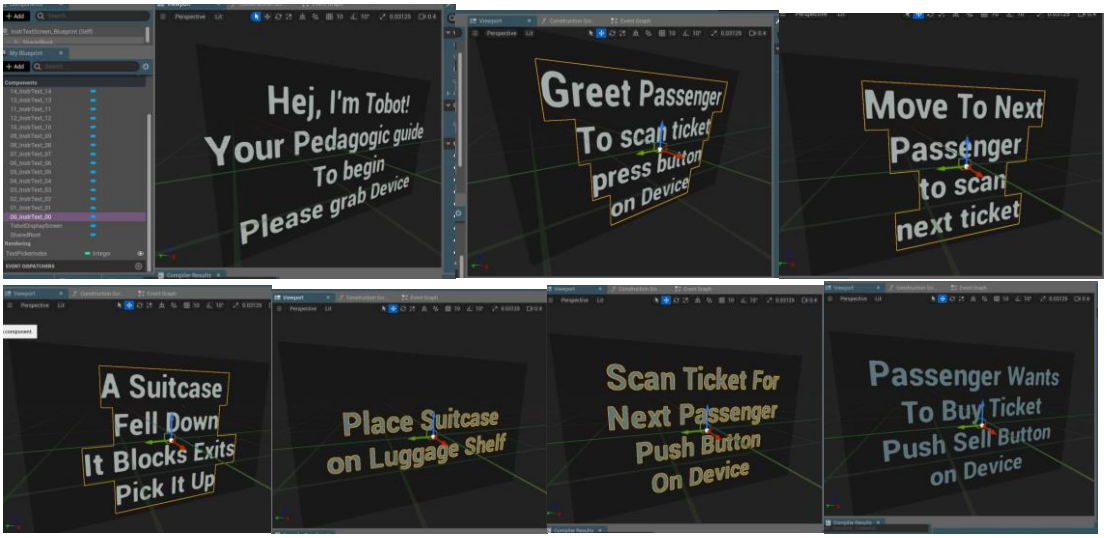
- Tobot, the Pedagogic agent/avatar, with screen blueprint of 13 Text Renders
- Device:
  - Three buttons with overlays and triggerboxes
- Environment
- Extra Triggerboxes:
  - By device
  - by first Passenger
  - no1 by second passenger
  - by suitcase
  - By luggage Shelf
  - no 2 by second passenger
  - Over device screen
  - By creditcard
  - by third passenger
  - on printed ticket
  - by emergency phone
- Blueprints to connect for functionallity
- Components
  - Suitcase with overlay
  - Creditcard with overlay
  - Printed ticket with overlay
  - Emergency phone with overlay
  - Flickering/pulsating lights
  - Flickering/Pulsating overlay

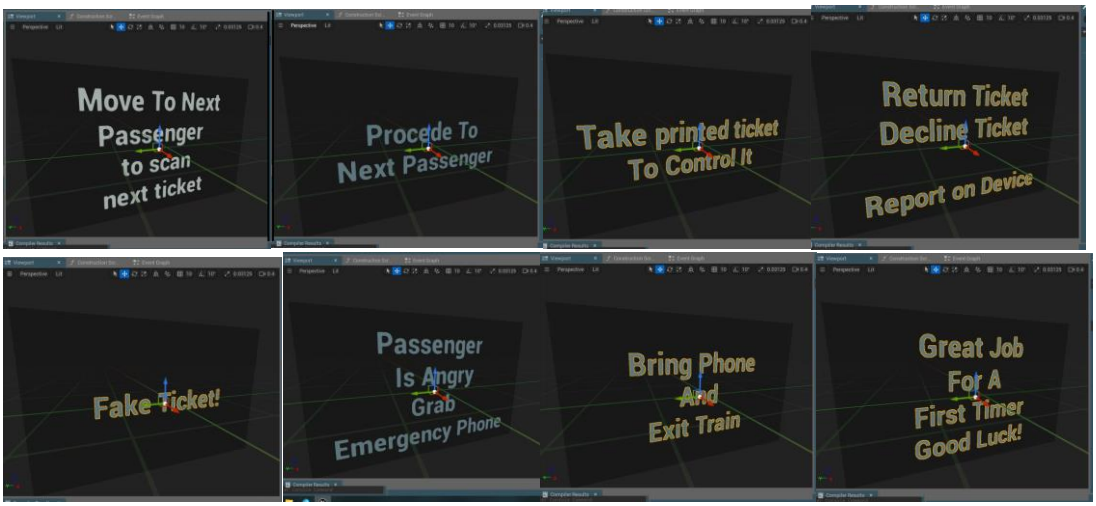
Then I realized I had to break it down even further to be able to organize the role of each trigger box along the way. So, I sat for half a day working on dividing it up in trigger boxes, that was going to start a series of events each. Most of them also update Tobots text including changing Tobots text messages.



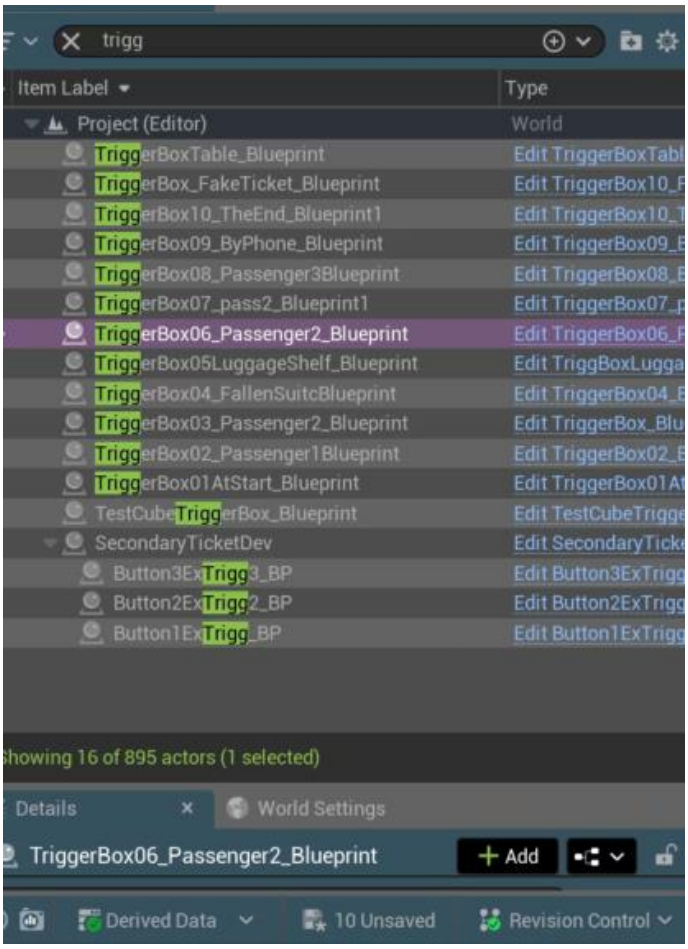


I shared these with my course colleague, putting him in charge of sound effects since he lives closer to school and could go to OpenLab and actually hear the sounds and test play the prototype during development in one of the VR HMDs. Then I finished up writing the text renders.

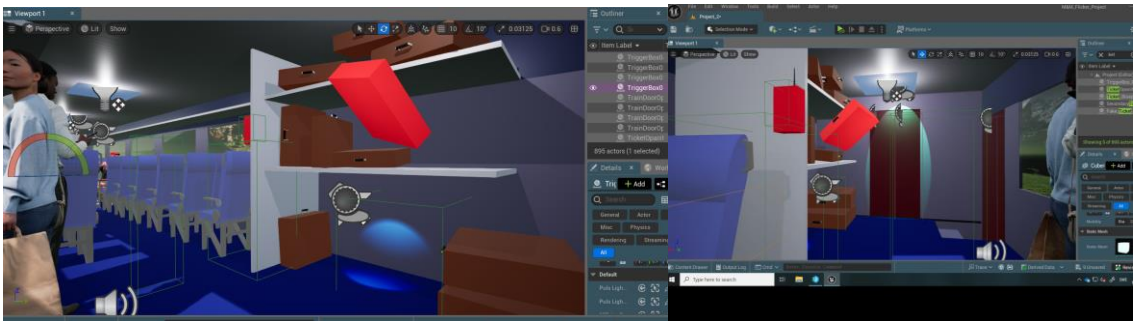




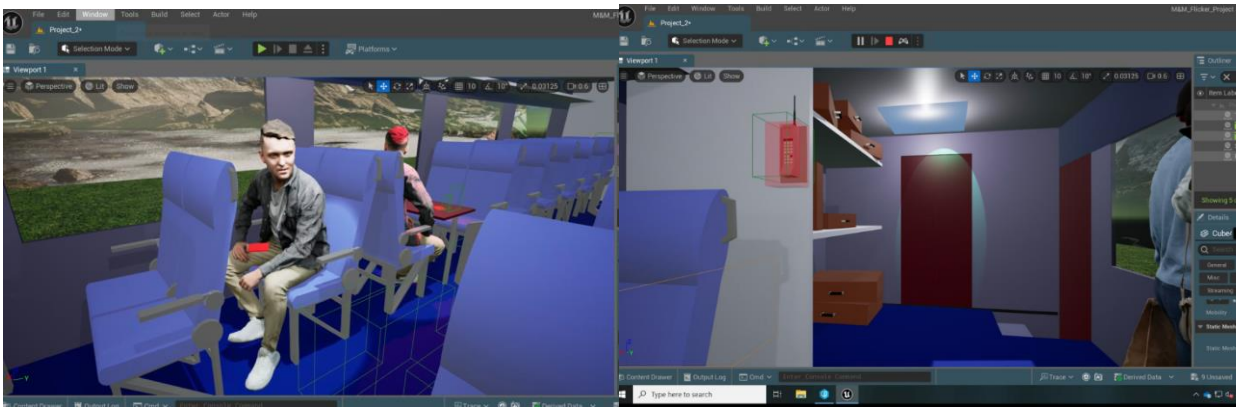
Next step was setting up all of the trigger boxes and making the blueprints communicate with each other when triggered.



The suitcase was strategically placed, and set hidden in game, to be able to fall after user collides with a trigger. The pulsating overlay does not show until it has landed.



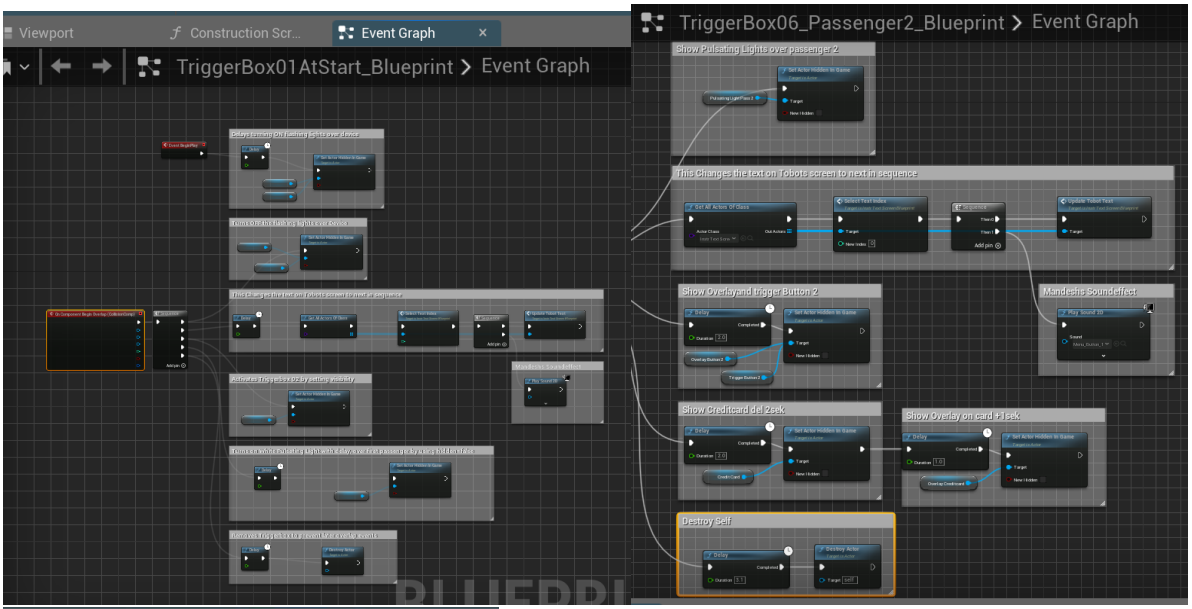
The light pulsating/flashing on/over the next passenger to interact with triggered by the trigger box ahead of it in the sequence.

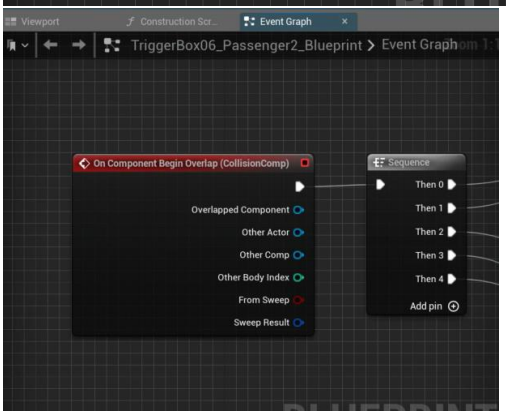


The overlay on the phone and the green light on the door are to be triggered one after the other. In all there were 15 triggers placed in the level, including the ones on the ticketing device. They all contained

several sequence events/things that would occur, either simultaneously, after each other, or with a time delay.

Examples of the blueprints I have made:





I prefer to make things look nice and tidy and line up my code, so I am thankful for the comment boxes that make it easy to keep things together a mark them up. Some of these blueprints looked no better than a ball of yarn that the cat got a hold of, before cleaning it up with these boxes and comments.

One thing I was considering was to have only player being able to trigger a box, after running into trouble with ghost like behaviors where the suitcase would fall at start and Tobot text would keep changing and only stopped for time delays… Then there were other things that started happening too. But setting a delay on a trigger box that is meant to trigger a set of events after being activated, is probably not the best solution in a scenario like this where space is kind of tight and you risk hitting the same trigger box more than once. One way to fix the Tobot text changing ahead of time would be to set

an index number for each trigger box text changing. I started doing that but unfortunately ran out of time. I also started setting the trigger boxes to destroy themselves after all the events where triggered, but the time delay in some of them still made it possible to collide with them more than once. Then I tested leaving collision turned off until it is turned on by the preceding trigger box but being short on time, I did not get a chance to test this out fully.

Individual Analysis

I am proud to have learnt as much as I have, from literature, seminars and during classes, as well as about Unity, its blueprints, Perforce, prompting Chat GPT and working with people from other cultures, not only in this course, but so far, throughout the Master's program.

In hindsight maybe I should have communicated more with Darmin and for example asked for some feedback on the intensity, color and the speed with which the overlays and lights were pulsating/flashing. According to Sutton, et.al (2024) a flicker is most effective when it is not too obvious. They also came to the conclusion that a flicker could make users dependent upon seeing the flicker to follow the instruction. It would have been good for the project to discuss this further while connecting it to the agent/avatar instructions, as it would have given insight from another point of view. According to Sutton, et.al. (2024), A high intensity fast flickering is fast at catching someone's attention but can also be too intrusive and take focus from other things that might require attention simultaneously. The balance between Flicker and Pulsating, intensity and making sure a visual overlay or light is visible, is even more crucial when considering a prototype that could possibly work even in AR,

I am feeling pride in many of the things in our project, since I did create most of it. I wish there could have been more course mates and several could have worked on the same project. I made the mistake of aiming to far and visualizing something that I ,being new to Unreal, have not been proficient enough to create, fast enough. There is a good foundation in the storyline, and the materials are basically in place, so someone with more programming skills could likely fix the issues, without spending several weeks trying to find answers and experimenting their way to the correct solution.

Finding Tobot and creating the train, together with creating a storyline and coming up with this concept, that I am hoping someone will continue to research, has been a great deal of fun. I am always full of ideas and trying to find solutions for every issue. I keep seeing more and more possible areas of use, seeing all the possibilities for implementing this idea.

Starting to learn to program with blueprints, as far as simple collisions, with some help from chat GPT, has made me see a little bit of what great potential there is in learning Unreal Engine. Although it has also been frustrating and tiring, with sleepless nights and being stuck with code errors taking focus time from everything else.

One thing I really wished for was access to MAYA for making models, but I never got a response regarding the license. Making those chairs and the train wagon from basic shapes did not take much time either though. There is also a plugin for modeling in Unreal, that unfortunately is not yet stable. If I

had used that I might have been able to make a seat with a back and bottom cushion as one piece, But I did not want to take chances.

Stumbling my way through this course, as well as more or less stumbling and falling on the finish line, I must still reflect on how incredibly much I have learnt. I will raise back up again, wiser and stronger.

References

Andersson, Ulf R. (2024). Sense and Storyfication: From virtual worlds to storyworlds, PhD Thesis: University West ; 73, Trollhättan: University West. https://hv.diva-portal.org/smash/record.jsf?pid=diva2%3A1910308&dswid=-1855

Grivokostopoulou, F., Kovas, K., & Perikos, I. (2020). The Effectiveness of Embodied Pedagogical Agents and Their Impact on Students Learning in Virtual Worlds. *Applied Sciences*, 10(5), 1739. http://dx.doi.org/10.3390/app10051739

Haj-Bolouri, A., Katende, J. O., Rossi, M. (2024). Gamified immersive safety training in virtual reality: a mixed methods approach. *Journal of Workplace Learning*, Vol. 36, nr 7, s. 516-538. https:// doi/10.1108/jwl-01-2024-0008/full/html

Ke, F., Pachman, M., & Dai, Z. (2020). Investigating educational affordances of virtual reality for simulation-based teaching training with graduate teaching assistants. *Journal of Computing in Higher Education*, 1-21. https://doi.org/10.1007/s12528-020-09249-9

Sutton, J., Langholtz, T., Plopski, A., Hornbaek, K. (2024). Flicker Augmentations: Rapid Brightness Modulation for Real-World Visual Guidance using Augmented Realit *CHI '24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* Article No.: 752, Pages 1 – 19. https://doi.org/10.1145/3613904.3642085

Tubis, A.A, Restel, F., Jodejko-Pietrruzuk, A. (2023). Development of an Virtual Reality Tool for Train Crew Training. *Appl. Sci.* , *13*(20). https://doi.org/10.3390/app132011415