Technical Project - Progress Report

ESOF-2918-WA

Christopher Silver - 0877675

Angel Martinez - 0882475

Qianzheng Jiang - 0901431

Shane Davey - 0885534

Supervisor : Dr. Benlamri

February 14th, 2020

Minutes of Previous Meetings

February 13th, 2020 - 12:15PM to 1:00PM

Team members: Christopher Silver Christopher Silver Qianzheng Jiang Qianzheng Jiang

Purpose of meeting: To meet with Dr. Benlamri to review our progress report.

Outcome of meeting: He gave us advice on what to add to our report and how to format reports.

February 6th - 4:00 PM to 5:15 PM

Team members: Christopher Silver Christopher Silver Qianzheng Jiang Qianzheng Jiang

Angel Martinez Angel Martinez Shane Davey S.Davey

Purpose of meeting: Meet with students Sean Dunlop and Ryan Friesen to get advice about virtual reality development.

Outcome of meeting: They gave us advice and tips on how to make our project successful.

February 6th - 2:20 PM to 2:40 PM

Team members: Qianzheng Jiang Qianzheng Jiang

Purpose of meeting: To discuss with Dr. Benlamri regarding the focus of our project.

January 20th, 2020 - 11:00 AM to 11:30 AM

Team members: Christopher Silver Christopher Silver Qianzheng Jiang Qianzheng Jiang

Angel Martinez Angel Martínez Shane Davey S.Davey

Purpose of meeting: To meet with Dr. Benlamri to learn our development environment

Outcome of meeting: We were introduced to our workspace and the Oculus Rift technology.

January 14th, 2020 - 11:30 AM to 12:30 PM

Team members: Christopher Silver Christopher Silver Qianzheng Jiang Qianzheng Jiang

Angel Martinez Angel Martínez Shane Davey S.Davey

Purpose of meeting: To meet with Dr. Benlamri to get ideas for a project

Outcome of meeting: We decided we would create a VR application for the purpose of training.

Table of Contents

Introduction

Virtual reality (VR) is a three-dimensional, computer generated environment which can be explored and interacted with by users. The user wears a headset and the headset displays the virtual environment, which moves as the user moves. The user also has controllers which act as hands. The headset and controllers are tracked using a camera and the camera calculates where you have moved, which way you have turned your head, and where your hands are based on vectors between the sensors on the devices and the camera unit. Since when you move in real life, your headset shows you moving accordingly, it feels like the user is actually in the virtual environment. Taking advantage of the ability to create controllable and repeatable circumstances, virtual reality has been widely used to provide precise training scenarios. Some examples of these scenarios are measuring athletes' performance and cognitive behavior and improving perceptual skills. These factors can benefit from the realistic simulation of VR.[3] When it comes to training jobs that have risks, implementing VR technology in training effectively minimizes the risks in training and therefore on the job. Furthermore, VR systems can also significantly reduce the cost of training as it can be done remotely.

Research ideas and objectives

The current process for training potential employees for the CBSA (Canada Border Services Agency) to search vehicles for illegal substances is expensive and unrealistic. The current routine for training is to send all the employees to Quebec and be trained through real-world simulations by actors hired by the government. This costs the government a lot of money and is potentially inconvenient to employees if they do not want to travel. Also, the simulations have an illegal substance in the vehicle 100% of the time, which is unrealistic and therefore ineffective training for the job. In the current training procedure, there is a limitation as to what you can include in training. The officer's safety is the number one priority, and because of this there is a lack of training in potential but unlikely scenarios, such as finding a bomb.

Our objective is to create a VR application to simulate the action of searching vehicles for illegal substances. We would make a vehicle be a non-moving but interactive object in which you could open vehicle doors, trunks, etc. and there would be items scattered throughout the car. We will create a number of scenarios in which there are different items inside of the car, and it will be the users job to determine whether there is something illegal inside of the car, and if so, the user would have to select the object. We would make 50% of the scenarios contain an illegal substance.

Our solution would give the officer experience in looking for illegal substances without the need of travel or actors hired by the government. The VR system would be portable and could travel from station to station. Since 50% of the scenarios contain an illegal substance, it would be more realistic unlike the current situation, causing the officer to search more attentively. This would give supervisors a better outlook on a potential officer's skills to determine whether they could handle real-world situations. Since we are creating a VR application, we could include unlikely scenarios that are dangerous to officers, such as finding a bomb in a vehicle. Anything that could be hidden in a vehicle could be simulated in our application, meaning officers could get training in all potential situations with our application.

Literature Review

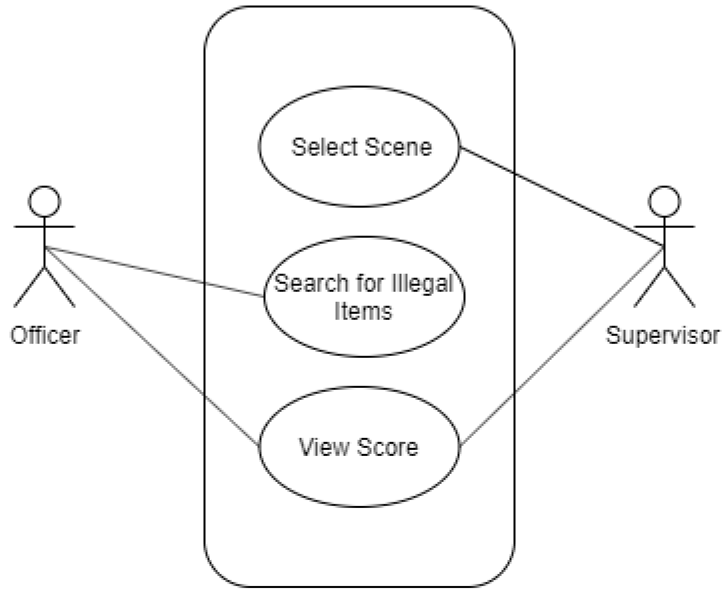
Virtual reality (VR), with the features of immersion, interactivity and imagination, is described as a cutting-edge technology that allows learners to step through the computer screen into a 3-D interactive environment.[4] Although there is indication that using VR for medical purposes is not developing as fast as using VR in other perspectives, it is still an effective method for training. Especially when it comes to surgery. A conference held in Arles, France suggested that surgery is slowly moving from a Halstedian apprenticeship model involving training on live patients, to complementary training using simulation environments.[5] Not only does virtual reality benefit the medical industry, it has positive contributions elsewhere. Training simulations of high-reliability tasks have been utilised by industries worldwide to help learners gain knowledge and experience before actively engaging within a role. Most of these industries have relied on paper, classroom or real-world training exercises to achieve this.[1] As the various implementation of VR for training purposes is examined in other fields, its benefits become obvious. VR gives us readable feedback to analyze the outcomes of certain activities. An example is the use of VR to determine the offensive awareness and decision-making for professional basketball players. The study suggests that participates who used the VR system took less time on the decision of offensive action, which showed the potential of using the proposed system to assist the training of decision-making.[3] In terms of making better decisions, some particular job types require more precise and accurate decisions, especially to those jobs that have a potential for serious risks. An IEEE conference in Hamamatsu, Japan suggested that using VR for simple training purposes is not as effective as training in real life. However, when it comes to some work that has more emphasis on safety rather than working speed, handling dangerous goods such as knives and explosives in VR may be meaningful in terms of both safety improvement and training effect.[2] Based on the outcome of this research, the benefits of using VR to train employees is most beneficial when the employees are Law Enforcement Agencies (LEAs), first responders, etc. Not only is VR training effective in these situations, modeling real life situations would be often expensive to run and require significant resources to plan and manage.[1] Overall, VR in general is an effective new technique that can provide guidance on training purposes and a good platform to study the efficacy of the training in providing guidance for the procedures.[5] So far, there is no such VR system provided for CBSA (Canada Border Services Agency) and other border agencies globally. Our project will conduct the benefits of VR in training for border agencies, whose agents would encounter dangers in searching illegal items. This would reduce the financial pressure for CBSA so that those resources can be allocated in other elements of securing the Canadian border.

System Design and Architecture

Unity is our main game engine where the scenarios will take place. Models are found on turbosquid or blendswap, modified in blender, and then imported into unity which are rearranged and placed in the scene how we want them. We add physics attributes and scripts to these models in unity. Once all of our scripts are completed, we physically attach the VR to the computer through the USB and HDMI cables.. We were given the oculus rift technology by Dr. Benlamri and we used pre-built libraries to connect it in unity. We did not have to make custom codes or scripts to make the VR headset work in unity. We imported libraries that did this for us. We simply had to connect the oculus to unity and then click the play button, and it will launch.

- ### Use case diagram:

  Our use case diagram for our project is shown below. We will have two participants (officer and supervisor), who will have different access to our VR system by entering into different Graphical User Interface(GUI). The supervisor can have the authority to select different scenes for the officer to practice and can notice the outcome of this practice by viewing the scores. The other participant, the officer will need to search illegal items that system placed in a selected scene by supervisor and can review how he or she did by viewing the score.



- ## VR system:

We have been using unity as a game engine, oculus rift as our VR hardware, and blender for creating and fixing existing models. We have found some existing free models on turbosquid and blendswap.

- ## Our High-Level Design and GUI Design:

Our goal is to create a GUI using C# scripts. The GUI will have a menu for the user to select an option. We will have the following options: “Begin”, “Score”, “Select Scene”. If the user selects “Begin”, one of the scenes we created is randomly selected and generated. The user will be placed into the scene. The user must grab the illegal objects (if any) and place them in a box outside of the vehicle. Once the user feels they have found every item, they press a certain button (yet to be decided) on their controllers to exit the scene. Their score will then be calculated, which will just be a percentage out of 100. The formula for score calculation as follows: What oh perfect

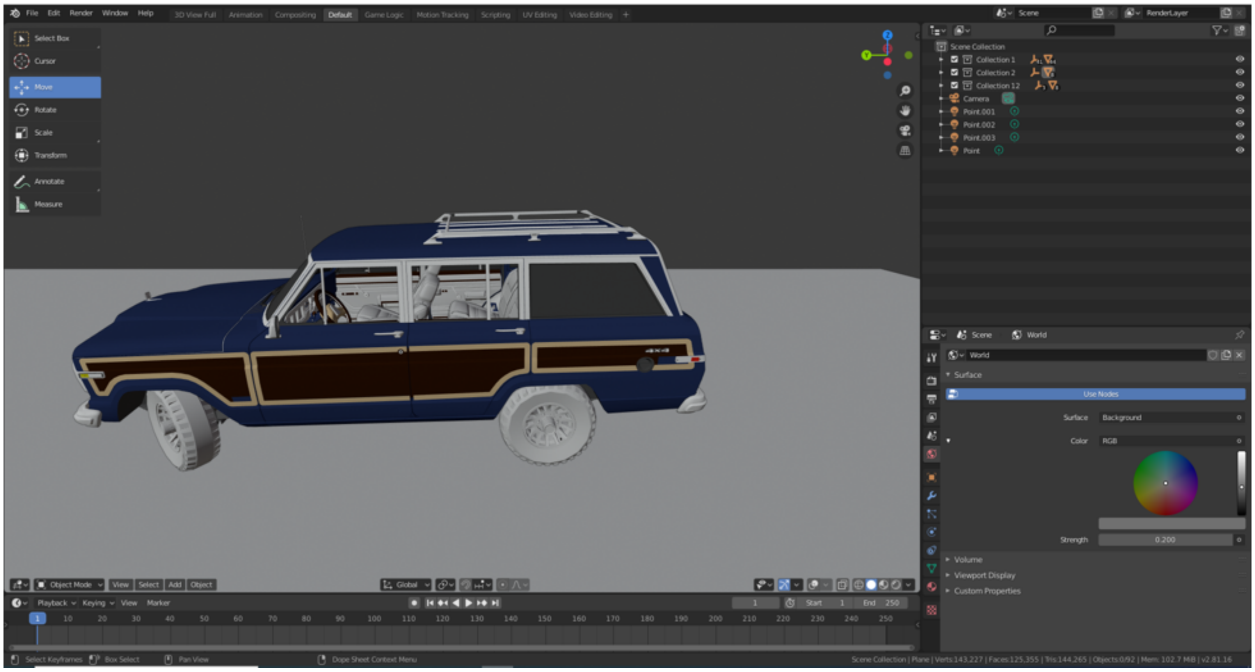
$$Score \ = \ (\frac{Number\ of\ illegal\ items\ found}{Number\ of\ illegal\ items\ in\ car} - \frac{Number\ of\ legal\ items\ wrongly\ selected}{Number\ of\ illegal\ items\ in\ car})\ x\ 100$$

Ideally, you could create your own user and your score would save associated with your user, but considering our time constraints, there will only be one user and there will be an overall score for the user (shown in the “score” tab). This number is an average of the score they have achieved each playthrough.

There should also be a tab for selecting a specific scene that is unique from the other. This way the government could (for example) use scene 1, 3, and 4 for each employee so they all get the same training and it’s fair. They could take statistics on the score of each scene to understand in what scenarios the officers are least effective.

- Details / what we have done:

We first created a scene in unity and created a plane in which we are putting our assets on. We imported the car and realized that we needed to separate the doors and the trunk from the rest of the car. We did so in blender and then re-imported the vehicle. This is the vehicle:



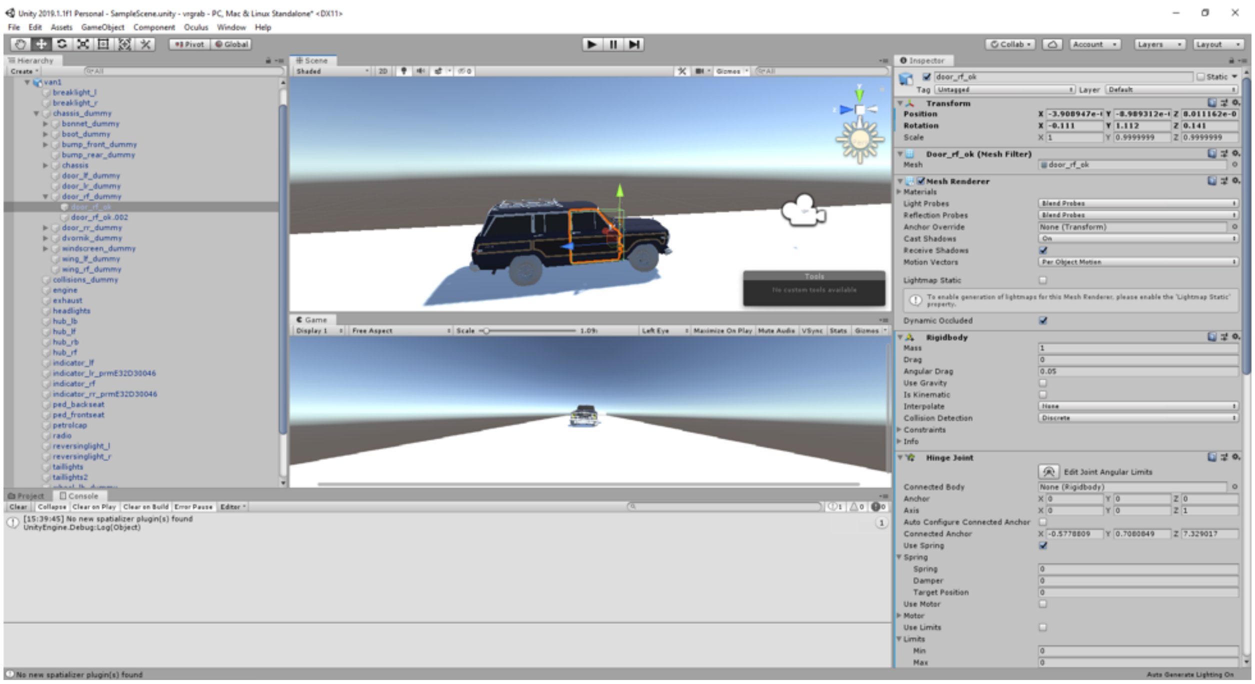
We then decided to work on the VR aspect of the game. We needed to enable the virtual reality and the oculus machine in Edit -> Project Settings -> Player. We downloaded “oculus integration” from the asset store, and searched for the OVRPlayerController which acts as the eyes for the user. Because the user looks through the OVRPlayerController, we deleted the main camera from the scene as it was no longer needed. In order for unity to check for the height of the user and change the OVRPlayerController when the user’s head moves, we had to search for OVRCameraRig in the OVRPlayerController dropdown and change the tracking origin type to “floor level”, and we had to add the “character camera constraint (script)” component to the OVRPlayerController. In this component the “camera rig” must be set to the camera rig seen in the “sample scene” menu, and the dynamic height box must be checked.

We then put “CustomHandLeft” under LeftHandAnchor and “CustomHandRight” under RightHandAnchor. This allows the user to pick things up, move their fingers, and interact with objects.

Once the VR aspect was working, we wanted to control how the doors opened. We clicked on each separate door and added the following attributes: rigidbody with “gravity” off, box collider with “is trigger” on, hinge joint, OVR grabbable script (which allows the object to be grabbed by the user), and a script that sets the settings on the hinge joint that stops the door from opening 360 degrees around the hinge joint. This was required as we want the door to open realistically. Upon testing the doors, we realized there were 3 problems: 1. When “grabbing” the door to open it, you can just pull the door apart from the car and carry it around. 2. To open the door you needed to slam it, and to close it you had to have the motion as if you were opening the door. 3. You can grab the door anywhere and not just the handle. Upon conferring with students who worked with VR last year, we decided that we will add a spring joint to an invisible box that is around the door handle. When you grab the invisible object, the door will come as well. This will solve the issue of grabbing the door off its hinge since the door will no longer be a grabbable object. Since the door is no longer a grabbable object, you can only grab the box around the handle to open the door which solves another problem, and since we will be using a spring, the door should open the desired direction as long as we can stop the spring once it is let go.

We also created a basic background in which the simulation will occur, and started creating a basic GUI which when the user clicks “begin” the scene is launched.

This is what our Unity Interface looks like. This is our work environment:



The right hand panels are the attributes of the model that we can add or edit, the top view is the “developer” view and the bottom view is the “game” or user view. The panel on the left shows all models we can work with or edit.

- What we still have to do:

We have to find and import more assets for the scenes, and then fill the car with these assets. We need to create a basic background in which the simulation will occur, and we need to start creating a basic GUI with a “begin” option. We will create a box in which the illegal items will go and calculate how well the user did. This will be completed by having a script running that detects when an object touches the box. The script will confer to see if the item is illegal, and if so, it will adjust the score accordingly. We are planning to have the first scene completed by the end of February in time for the first presentation. Once that is completed we will have to replicate it with 4 other scenes and put other assets in different places. Once we have our 5 scenes, we will need to randomize when the user selects “begin” so it’s not the same loop of scenes each time. We then will create the score aspect based on the contents in the illegal bin and finish the GUI adding the more options. At that point our prototype will be complete.

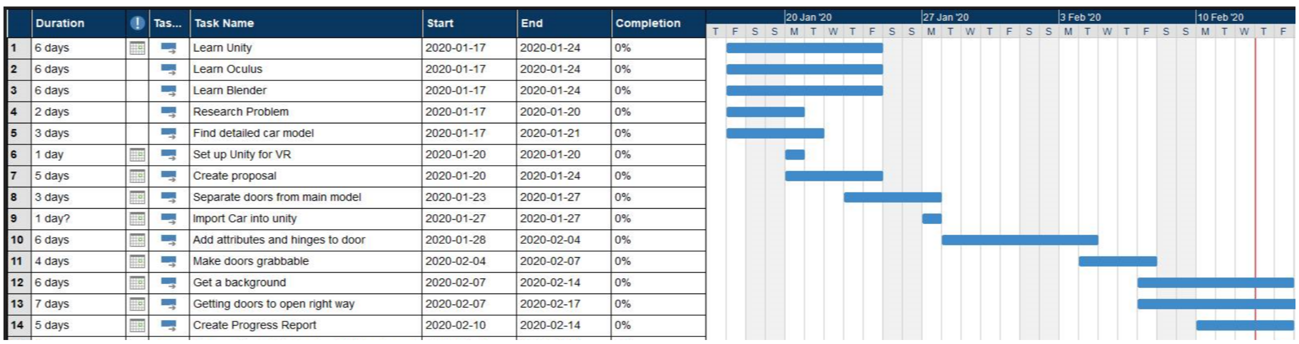
Project Management / Gantt Chart

We have created a gantt chart to track our progress in the project. This way we have a concrete plan and know what we have to do each day / week. Since we have everything planned now, we can plan for potential problems (such as trying to work on the project on weeks when we have lots of midterms) and find a solution now, rather than later on, thus pushing everything behind schedule. Since we have a concrete plan we can all stay accountable and easily see what needs to be worked on. So far, we are on track to finish the project on time. We have allotted 1 week for a buffer time at the end of the project that will be used if we fall behind schedule. While the major learning curves of the technologies have been overcome, the tedious tasks remain in the future that may take more time than we have allotted, so we must ensure we are remaining on schedule and put in extra work when needed.

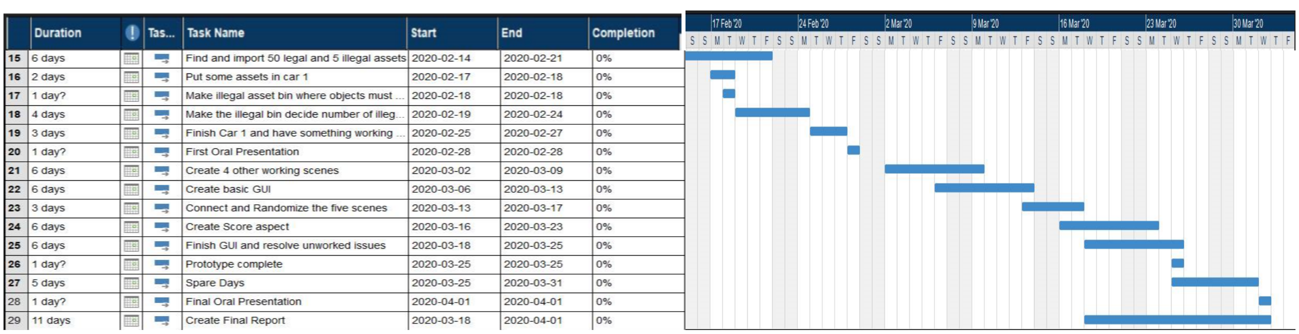
In general we are looking for a 25% split of work for each team member, with some people focusing more on report writing and research, while others are focusing more on the prototype. We have a group chat to keep everyone informed and accountable to do their work. So far, everyone has been working on the prototype, but group members have specialized and worked more on some aspects. Christopher took the lead on the report writing. Qianzheng has focused on research. Angel learned unity and blender and then taught the rest of the group. Shane will be working on the fine details within the prototype.

When we go to the lab in ATAC 4013, we open our project and save it under a new folder, with the current date on the folder. That way we can save multiple versions of the project in case something goes wrong or something is changed that all members do not agree with. We also write a notepad file each day that contains changes from the previous day, in order to backtrack in case of a bug or issue. This is us trying to implement a version control tool, and it also allows us to see who is working on the project and who is not.

This is our gantt chart:



	Duration	!	Tas...	Task Name	Start	End	Completion
1	6 days			Learn Unity	2020-01-17	2020-01-24	0%
2	6 days			Learn Oculus	2020-01-17	2020-01-24	0%
3	6 days			Learn Blender	2020-01-17	2020-01-24	0%
4	2 days			Research Problem	2020-01-17	2020-01-20	0%
5	3 days			Find detailed car model	2020-01-17	2020-01-21	0%
6	1 day			Set up Unity for VR	2020-01-20	2020-01-20	0%
7	5 days			Create proposal	2020-01-20	2020-01-24	0%
8	3 days			Separate doors from main model	2020-01-23	2020-01-27	0%
9	1 day?			Import Car into unity	2020-01-27	2020-01-27	0%
10	6 days			Add attributes and hinges to door	2020-01-28	2020-02-04	0%
11	4 days			Make doors grabbable	2020-02-04	2020-02-07	0%
12	6 days			Get a background	2020-02-07	2020-02-14	0%
13	7 days			Getting doors to open right way	2020-02-07	2020-02-17	0%
14	5 days			Create Progress Report	2020-02-10	2020-02-14	0%



	Duration	!	Tas...	Task Name	Start	End	Completion
15	6 days			Find and import 50 legal and 5 illegal assets	2020-02-14	2020-02-21	0%
16	2 days			Put some assets in car 1	2020-02-17	2020-02-18	0%
17	1 day?			Make illegal asset bin where objects must ...	2020-02-18	2020-02-18	0%
18	4 days			Make the illegal bin decide number of illeg...	2020-02-19	2020-02-24	0%
19	3 days			Finish Car 1 and have something working ...	2020-02-25	2020-02-27	0%
20	1 day?			First Oral Presentation	2020-02-28	2020-02-28	0%
21	6 days			Create 4 other working scenes	2020-03-02	2020-03-09	0%
22	6 days			Create basic GUI	2020-03-06	2020-03-13	0%
23	3 days			Connect and Randomize the five scenes	2020-03-13	2020-03-17	0%
24	6 days			Create Score aspect	2020-03-16	2020-03-23	0%
25	6 days			Finish GUI and resolve unworked issues	2020-03-18	2020-03-25	0%
26	1 day?			Prototype complete	2020-03-25	2020-03-25	0%
27	5 days			Spare Days	2020-03-25	2020-03-31	0%
28	1 day?			Final Oral Presentation	2020-04-01	2020-04-01	0%
29	11 days			Create Final Report	2020-03-18	2020-04-01	0%

Problems we’ve encountered

We had an initial problem in which the hands in VR wouldn’t actually interact with the controllers. The hands would move when the controllers moved, but when we pressed grab or other buttons, the fingers wouldn't move accordingly. To solve this problem we moved “CustomHandLeft” as a child of LeftHandAnchor and moved “CustomHandRight” as a child of RightHandAnchor. These were originally not in the correct hierarchy in the scene, and is why the hands didn’t work as originally expected.

We also had the problem that the windows / glass didn’t follow the doors. When you opened the door, the glass would remain in the same position it was. To fix this we attached the glass to the door in blender and re-imported the car.

Another issue we currently have is the glass is only see-through when looking through the window in 1 direction - when you are outside the car looking in. If you open the doors and then look out the window, it appears as if there is no glass at all. Our proposed solution is to copy the glass model in blender, flip it, and attach it to the door as well.

We had a problem that the pivot points weren’t in the desired location. It was a painstaking task moving the pivot points as their scales were so sensitive.

Considering assets and importing assets, we have encountered a few problems. First, we could not find any free models for some 3D models such as a belt-buckle knife that we were planning to hide. Also, even if we could find some models, some of these models are in the

(.max) format which cannot be used in unity. Unity only accepts assets in 3D object (.fbx), (.dae), (.3ds), (.dxf), and (.obj) format and it is hard and very expensive to convert a (.max) files. Because of these issues, we decided to abandon some of these assets and have researched to find what else is illegal that we can place in the vehicle.

Conclusion and Future Work:

Ideally, we would have many types of vehicles that could be searched other than just the one model we have now. Also, ideally we would have 30 concrete and selectible scenes, but we would also have an option to completely randomize a scene. This means a random car would be selected, random legal and illegal assets are selected and placed in the vehicle semi-randomly. We would also like to have different “views” of each vehicle. For example, the user could select an option for the car to be jacked up so they could look underneath. They could also select an option where the car would be “searched” by dogs who sniff for drugs. The user would see the dog’s reaction in a sort of message box and have to react accordingly. These ideas are quite interesting and would increase the robustness of the program, but we unfortunately do not have time to implement these features so far.

References:

[1]. J. Saunders, S. Davey, P. S. Bayerl and P. Lohrmann, "Validating Virtual Reality as an Effective Training Medium in the Security Domain," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 1908-1911.

[2]. A. Kanazawa and H. Hayashi, "The Analysis of Training Effects with Virtual Reality in Simple Task," 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, 2017, pp. 345-350.

[3]. W. Tsai, L. Su, T. Ko, C. Yang and M. Hu, "Improve the Decision-making Skill of Basketball Players by an Action-aware VR Training System," 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 2019, pp. 1193-1194.

[4]. C. Li, R. Gu and J. Chen, "Research on instructional design model for VR supported skill training," 2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET), Cappadocia, 2010, pp. 232-235.

[5]. S. de Ribaupierre, R. Armstrong, D. Noltie, M. Kramers and R. Eagleson, "VR and AR simulator for neurosurgical training," 2015 IEEE Virtual Reality (VR), Arles, 2015, pp. 147-148.