Technical Project - Final Report

ESOF-2918-WA

Christopher Silver - 0877675

Angel Martinez - 0882475

Qianzheng Jiang - 0901431

Shane Davey - 0885534

Supervisor : Dr. Benlamri

April 3rd, 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. In the recent year, the developments of VR in both hardware and software aspects are incredibly significant. There is no denying that more and more VR is being implemented for various purposes suggesting the benefits of VR exceed original expectations. A lot of traditional procedures that require large amounts of resources allocation can be optimized by developing proper VR systems.

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 was to create a VR application to simulate the action of searching vehicles for illegal substances. We planned to 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 for 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. We also wanted to have an option to create infinitely many scenes from the pool of vehicles and assets. This way the game would be unique. While these randomly generated scenes would be present, we also wanted to create 5 scenes that were always the same, so one officer's results could be compared against others, and so the supervisor could know, prior to the scene being run, how many illegal items there are and where they are. This would allow the supervisor to better analyze the skill of employees.

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] As the development of technology increases exponentially, there are more and more fields of study that are using or considering introducing VR into real practice. VR systems can provide low-cost, realistic training for intracardiac techniques[9]. The most common usages of VR are in the medical field, training purposes, and assistance on teaching. In this particular real-life problem, we tended to emphasize our research process on how VR benefits in a training procedure that could be dangerous to participants.

The main purpose of using VR in the medical aspect is that the simulated environment surrounds an individual's perceptual field such that the user feels psychologically present in the digital world, rather than in their physical reality[6]. The reduction of potential risk that we can reduce is one of the major benefits we can acquire via VR simulation. 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. 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.[6] Currently in schools, the way we learn is through strictly fact retention, consuming a large amount of information and focusing on remembering as much as possible. Learning this way can be difficult and overwhelming for many students. While some people learn well by memorization, many people benefit more from hands-on experiences. This is where VR plays a significant role. It can give students learning experiences by providing them with a real hands-on experience that they otherwise may not have been able to do. Various fields of work can benefit from this. Examples of this would range from performing medical procedures to realistic flight simulators, and the great thing about learning in virtual reality, is that there is no actual risk while still being a realistic experience. [1]

From the aspect of training using VR implementation, we should consider the trade off when using such technology as it may not be as effective as we expected. Most of the VR developments in present time are specifically targeted to a certain project or environment. As stated in a Japan IEEE conference, In the practice of collaborative training of severe emergencies that occur too rarely within regular curricular training programs, multi-user virtual reality and serious game technologies can be used to provide collaborative training in dynamic settings. However, actual training effects seem to depend on a high presence and supportive usability[10]. The support that we need to enhance the accuracy for VR simulation is a challenge for non-professionals as the configuration of the entire VR system needs to adjust case by case. As the various implementations of VR for training purposes are examined in other fields, it's 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 participants who used the VR system took less time on decisions 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 jobs require more precise and accurate decisions, especially 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]

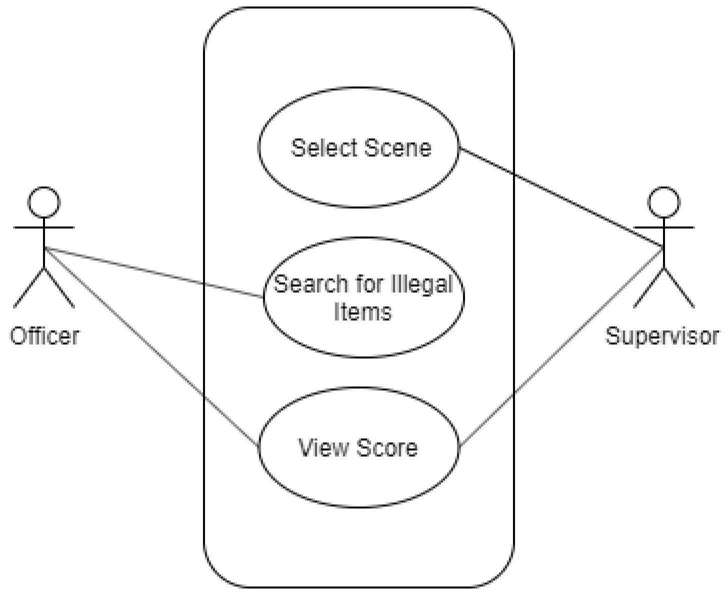
Overall, VR in general is an effective new technique that can provide guidance on training purposes and a good platform to study the efficacy of the training in providing guidance for the procedures.[10] Virtual reality has evolved in a short period of time, the field has been growing at an exponential rate. Examples are: greatly increasing the resolution in which the user sees as well as developing photorealistic avatars that can even reflect subtle facial features. As for the hardware, experts are working on making the device more comfortable, lightweight and removing the need for a seperate pc connection making it portable. They are also working to eliminate the need for physical controllers by creating hand and finger detection so the headset can detect what your hands are doing and therefore can act as the controllers themselves. However, there is an issue when it comes to using VR in training. A Landing Signal Officers (LSOs) device located in Oceana, VA, has been operating for several decades. The problem of this device, and it may also appear on all other VR simulators, is that the amount of time they can spend using this training device is undeniably too short. The need to provide LSOs with an unlimited number of training opportunities unrestricted by location and time, mixed with recent advancements in commercial off the shelf (COTS) immersive technologies, provided an ideal platform to create a lightweight training solution that would fill those gaps and extend beyond the capabilities currently offered in the 2H111 simulator[11]. This highlighted a realistic problem found in VR simulators. The development of VR hardware ideally should keep up with the software development. The lack of time spending on a VR simulator can significantly cut back the efficiency of VR systems. Nevertheless, the possibilities for a technology like this are endless and will continue to be developed at an exponential rate. The better the technology gets, the better it can be used for things like training and education. [5] Since VR was only introduced in the recent years, there is no such VR system provided for CBSA (Canada Border Services Agency) and other border agencies globally. Our project will take advantage of the benefits of VR in training for border agencies, whose agents could encounter dangers in searching illegal items. This would protect officers and also 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 hardware 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.

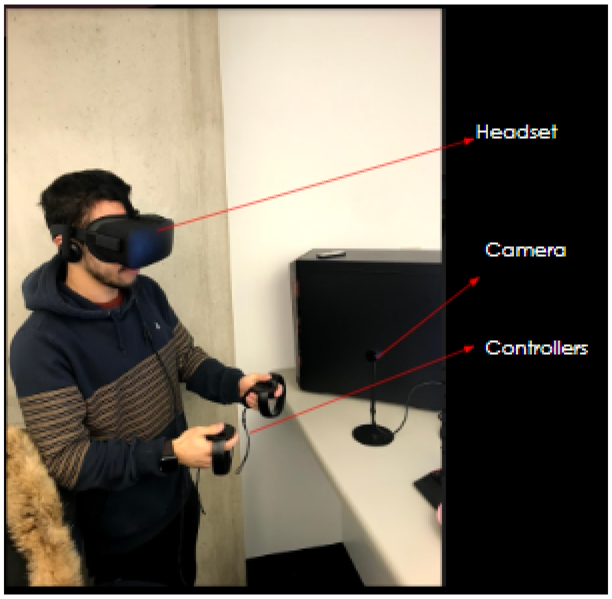
- ### Use case diagram:

  Our use case diagram for our project is shown below. We wanted to have two participants (officer and supervisor), who will have different access to our VR system by entering into different Graphical User Interfaces(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 for illegal items that are placed in a selected scene by the supervisor who can review how the officer did by viewing the score.

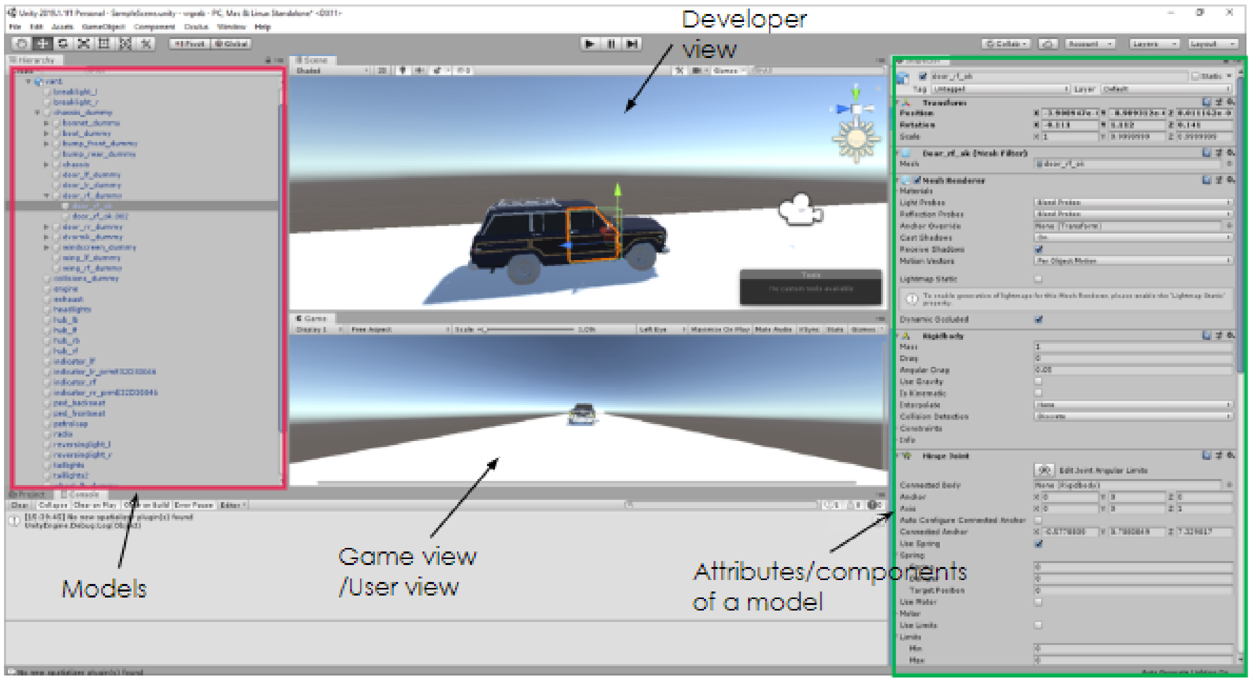


- VR system / Work Environment:

We have been using unity as a game engine, oculus rift as our VR hardware, and blender for fixing existing models found on turbosquid and blendswap. Below is a picture of a group member wearing the device, testing a scene. The cameras and headset connect to your computer, the camera detects your motion based on the position of the headset and the hand controllers. The buttons of the back of the controllers are used to open doors and grab assets.



Below 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.

● Our High-Level Design and GUI Design:

Our goal was to create a GUI using C# scripts. The GUI was envisioned to have a menu for the user to select an option. We will have the following options: “Play Random Game”, “Select Scene”, “View Score”, “Quit”. If the user selects “Play Random Game”, a scene with one of the cars is randomly generated, and a bunch of assets are randomly placed in the vehicle. 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 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:

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

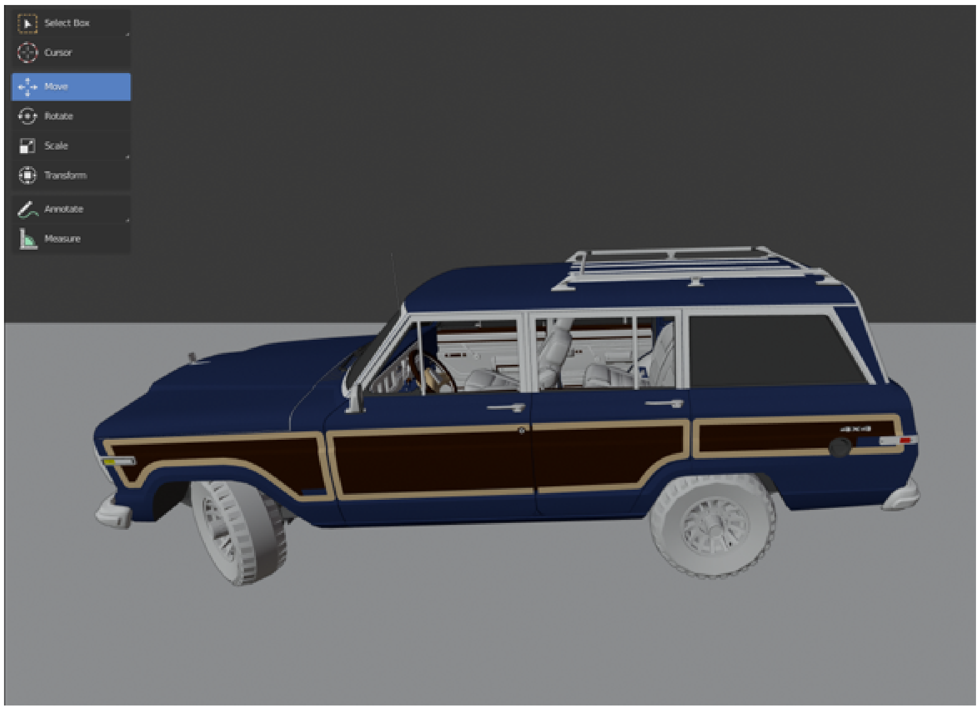
We would also ensure that any negative score received (such as selecting 0 illegal objects and 1 legal object) would be rounded up to 0%.

Ideally, you could create your own user and your score would be saved associated with your user, but considering our original time constraints, we decided from the beginning that 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. We planned to create 5 overall scenes that had different cars and different items in each. This way the government could (for example) use scene 1, 3, and 4 for each employee so they all get the same training. They could take statistics on the score of each scene to understand in what scenarios the officers are least effective, and they could isolate certain scores to ascertain which employees are struggling with searching vehicles and may require more training.

- ## Details / what we have done:

We first created a scene in unity and created a plane in which we put our assets on. We imported our first 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:



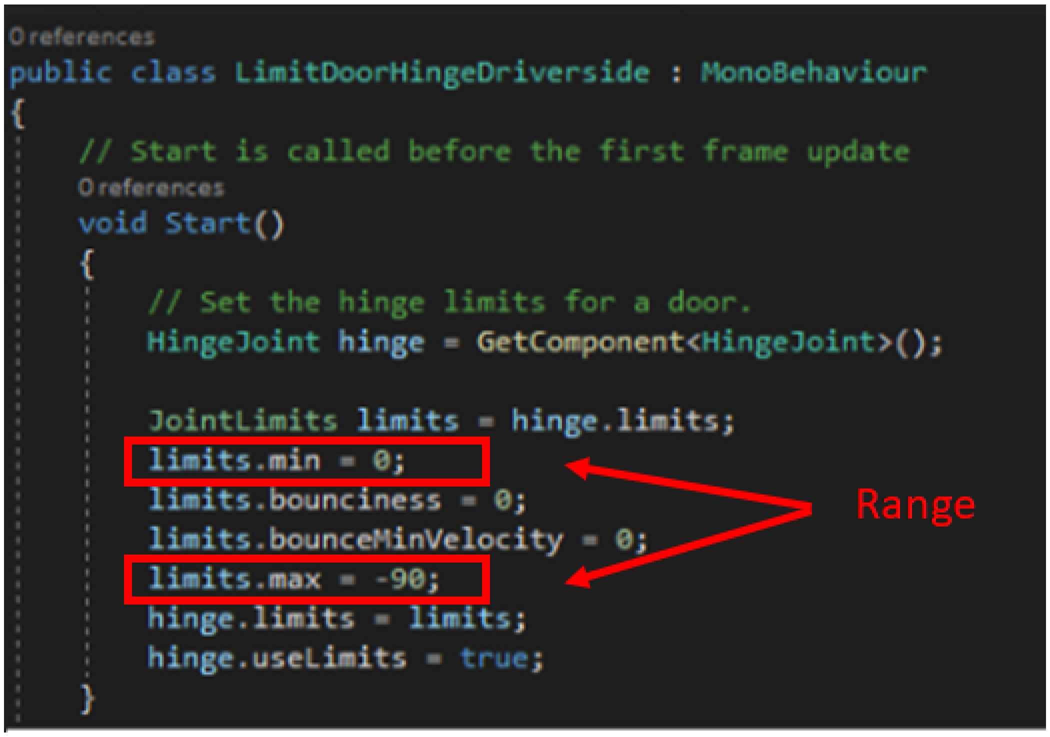
Implementing Virtual Reality

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.

Procedure to “Open Doors”

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. Here is the script for this situation:



```
0 references
public class LimitDoorHingeDriverside : MonoBehaviour
{
    // Start is called before the first frame update
    0 references
    void Start()
    {
        // Set the hinge limits for a door.
        HingeJoint hinge = GetComponent<HingeJoint>();

        JointLimits limits = hinge.limits;
        limits.min = 0;
        limits.bounciness = 0;
        limits.bounceMinVelocity = 0;
        limits.max = -90;
        hinge.limits = limits;
        hinge.useLimits = true;
    }
```

If you examine the code it can be observed that this code simply edits properties. We get the component “HingeJoint” and then access it’s JointLimits in an object called “limits” from here we could directly set each property of the HingeJoint. When limiting how much the door can actually open, we used “limits.min =0” and “limits.max= -90” which means you can only open the door in that range around the hinge joint.

All of the above procedures were 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.

Gathering Assets

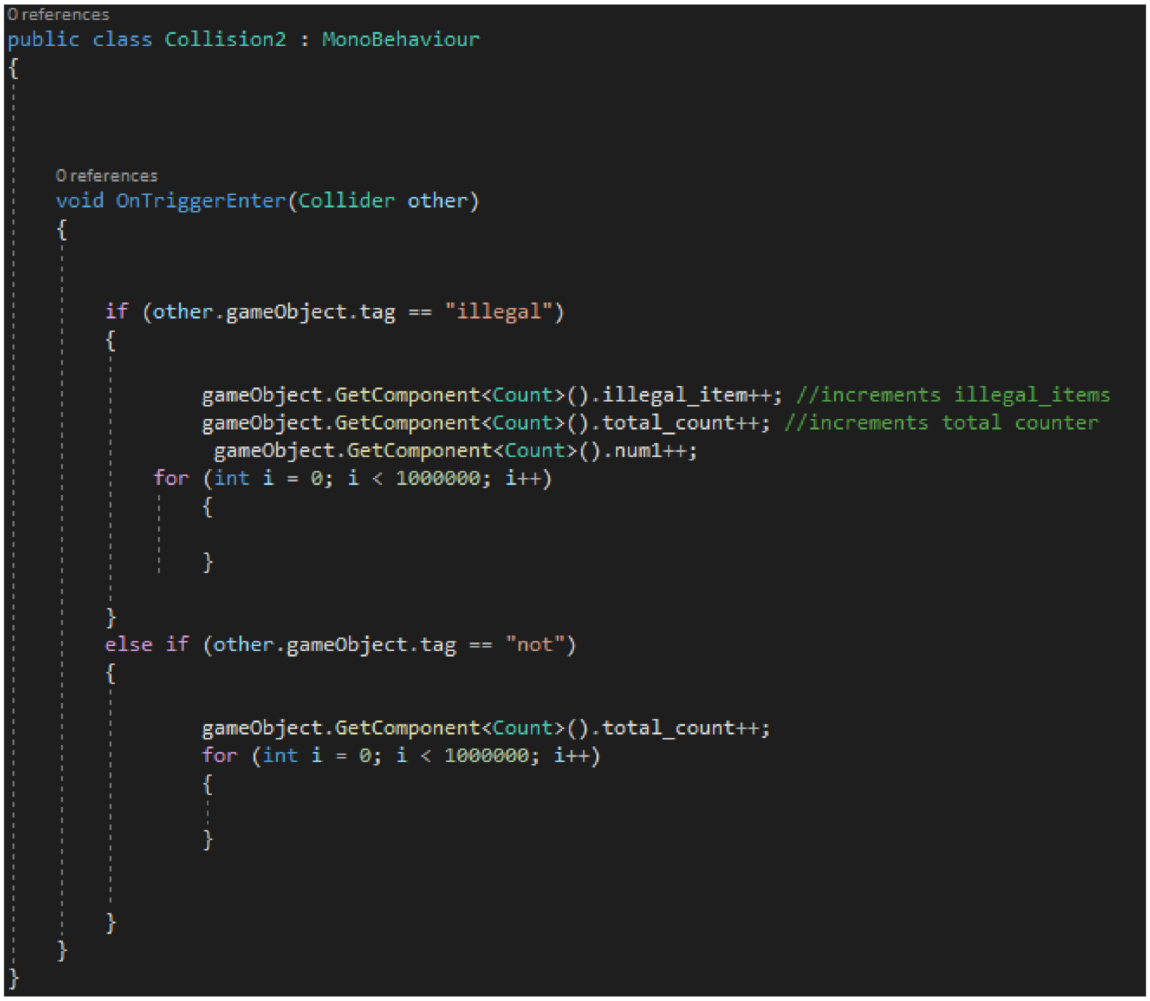
We realized that the heart of our game was quality assets, and these assets needed to be recognizable and easily distinguishable between illegal and legal, so we went hunting for assets. We searched on websites “turbosquid” and “blendswap” for free (.obj) and (.fbx) files. We found about 50 assets, 10 of which were illegal. The illegal items ranged from weapons to drugs. We also found 40 assets which were mostly everyday common objects.

Preparing for Presentation 1

At this point it was almost time for our first presentation. We needed a working version to show. We created a basic background in which the simulation will occur, which is just a brick wall on all sides. After this we placed planes on the seats and floors of the vehicle. We did this because the assets would actually fall until they contacted a plane. We needed something in the car to interact with the assets and stop them from falling. The planes achieved this. We filled our first car with 2 illegal assets and 5 legal assets. These were scattered along the floor, seats, trunk, and under the hood of the vehicle.

Creating Box to Drop Items Into

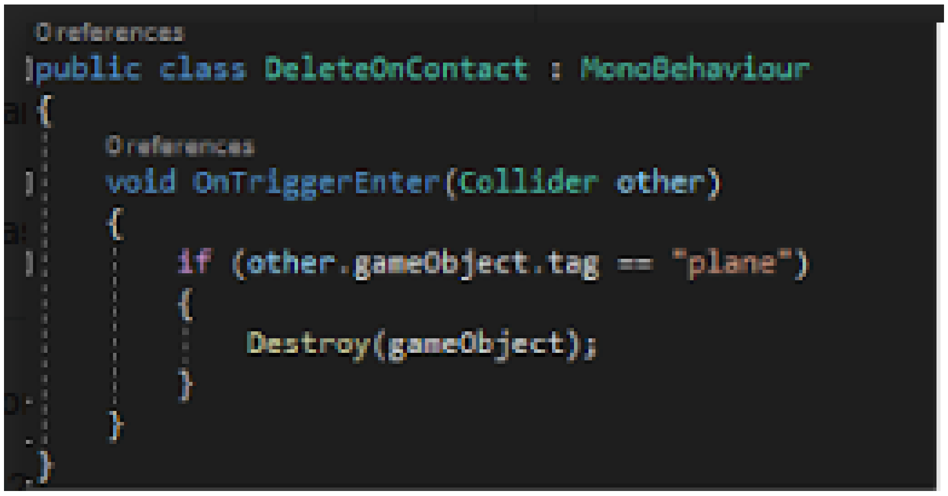
We also needed to create a box for users to drop their illegal assets into. We found an asset for a box and placed it in unity, near the vehicle. We placed a plane at the bottom of the box and added a script to the plane. Basically the script ran when there was a collision with an asset and the plane. There were 2 count variables: 1 for illegal items and 1 for total items. When a collision occurred between an item and the plane at the bottom of the box, it would increment the “total items” count every time, but it would only increment the “illegal items” count if the **tag** on the **asset** was set to “illegal”. Tags are basically things that describe each object. We took advantage of this and set the tag of all illegal items to “illegal” and set the tag of all legal items to “not”. The code for these count variables being incremented based on their tag is below. The empty for loops were used for debugging and should have been eliminated for the report, unfortunately, due to the COVID-19 pandemic, we were not able to access the ATAC machine and take screenshots of the most recent code.



```
0 references
public class Collision2 : MonoBehaviour
{



    0 references
    void OnTriggerEnter(Collider other)
    {


        if (other.gameObject.tag == "illegal")
        {

                gameObject.GetComponent<Count>().illegal_item++; //increments illegal_items
                gameObject.GetComponent<Count>().total_count++; //increments total counter
                 gameObject.GetComponent<Count>().num1++;
            for (int i = 0; i < 1000000; i++)
                {

                }

        }
        else if (other.gameObject.tag == "not")
        {

                gameObject.GetComponent<Count>().total_count++;
                for (int i = 0; i < 1000000; i++)
                {

                }


        }
    }
}
```

Also, the plane at the bottom of the box had a tag set to “plane”. This is significant for the following.

Although the breakthrough on discovering we could use tag to increment the count variables was monumental to us, we had more issues. When the assets are dropped onto the plane they would cause the plane to tip and fly around based on where it was hit from. Also, the object may bounce and incorrectly get counted many times. We fixed these issues by making the mass of the plane very large (so a small weight wouldn’t move it) and we created a delete on contact script, which deleted the asset from the scene once the count variables were updated.

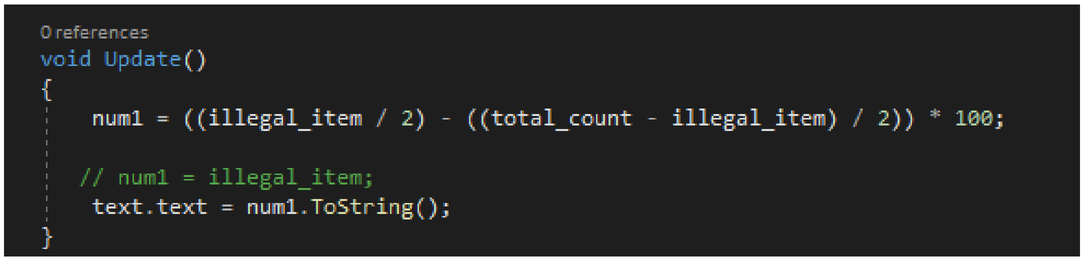


```
0 references
public class DeleteOnContact : MonoBehaviour
{
    0 references
    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "plane")
        {
            Destroy(gameObject);
        }
    }
}
```

Above is the code for the delete on contact script. This script is placed on each asset in order to check the tag of the object it comes into contact with. If the tag = “plane” it will delete. This does not mean the object will delete whenever it touches a plane, it just means it will delete when it touches something that has a tag of “plane”. The tags of the planes you walk on were null and the tags of the planes in the vehicle were null, and therefore wouldn’t delete when this contact occurs.

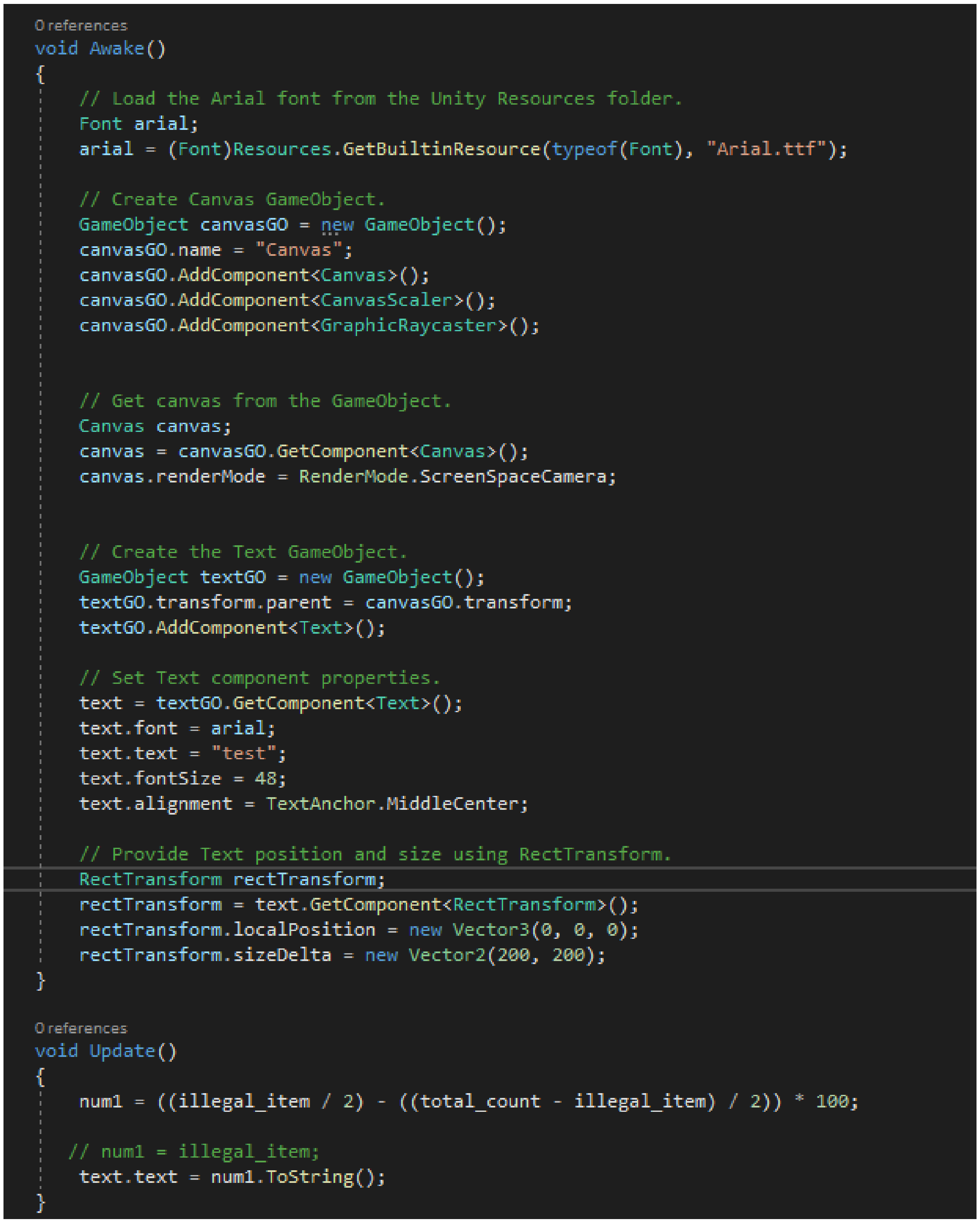
Calculate Score & Completing First Scene

Based on our 2 count variables and knowing that the number of illegal items in this first scene was 2, we could calculate the score. The following screenshot shows us doing so:



```
0 references
void Update()
{
    num1 = ((illegal_item / 2) - ((total_count - illegal_item) / 2)) * 100;

   // num1 = illegal_item;
    text.text = num1.ToString();
}
```

The “text” variable that gets updated is shown below. For the progress report we simply displayed the score on the **monitor** at all times. The score was not visible inside the VR headset though, which is exactly what we were aiming for in a demo. We were planning to have a final screen that is a summary and displays the user’s overall score for that round, but we were unable to complete this due to the COVID-19 pandemic. Below is the code for displaying the score as well as calculating the new score as well.



```
0 references
void Awake()
{
    // Load the Arial font from the Unity Resources folder.
    Font arial;
    arial = (Font)Resources.GetBuiltinResource(typeof(Font), "Arial.ttf");

    // Create Canvas GameObject.
    GameObject canvasGO = new GameObject();
    canvasGO.name = "Canvas";
    canvasGO.AddComponent<Canvas>();
    canvasGO.AddComponent<CanvasScaler>();
    canvasGO.AddComponent<GraphicRaycaster>();


    // Get canvas from the GameObject.
    Canvas canvas;
    canvas = canvasGO.GetComponent<Canvas>();
    canvas.renderMode = RenderMode.ScreenSpaceCamera;


    // Create the Text GameObject.
    GameObject textGO = new GameObject();
    textGO.transform.parent = canvasGO.transform;
    textGO.AddComponent<Text>();

    // Set Text component properties.
    text = textGO.GetComponent<Text>();
    text.font = arial;
    text.text = "test";
    text.fontSize = 48;
    text.alignment = TextAnchor.MiddleCenter;

    // Provide Text position and size using RectTransform.
    RectTransform rectTransform;
    rectTransform = text.GetComponent<RectTransform>();
    rectTransform.localPosition = new Vector3(0, 0, 0);
    rectTransform.sizeDelta = new Vector2(200, 200);
}

0 references
void Update()
{
    num1 = ((illegal_item / 2) - ((total_count - illegal_item) / 2)) * 100;

   // num1 = illegal_item;
    text.text = num1.ToString();
}
```

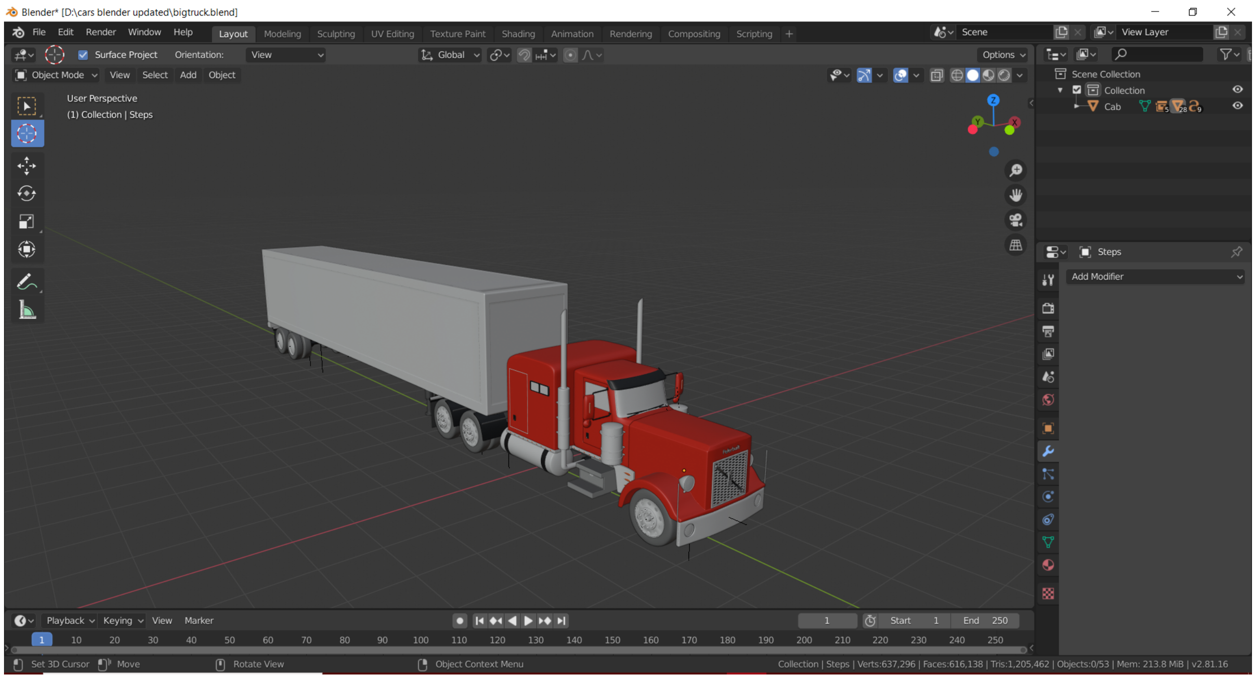
With everything in order for our first scene, we filmed two videos. The first was demonstrating us opening all the doors, trunk, and hood of the vehicle. It also showed our background. The second video showed us grabbing objects and dropping them in the box to be calculated towards the total score. We first drop a shotgun in the box, which is illegal, and causes the score to go up to 50%. We then dropped brass knuckles in the box, which is illegal, causing the score to go up to 100% since both illegal items were found. After that we dropped 5 straight legal items: mug, shoe, toy, syringe with needle, and fidget spinner. We noticed the score drop by 50 each time, from 100 to 50, 0, -50, -100, -150. This isn't realistic since you cannot have a negative score, we just wanted to demonstrate that the box and the score calculation was working as intended. In reality we would round a negative score up to 0 if they ended with a negative score. We would indeed keep the negative number throughout the game though, since if they first dropped a legal item in the box and then 2 illegal objects, the score would go from -50 to 0 to 50. However if you rounded all negative numbers to 0 you would get a score of -50 (rounded to 0) 0 to 50 and 50 to 100% even though you incorrectly selected a legal asset. Links to the videos are now shown:

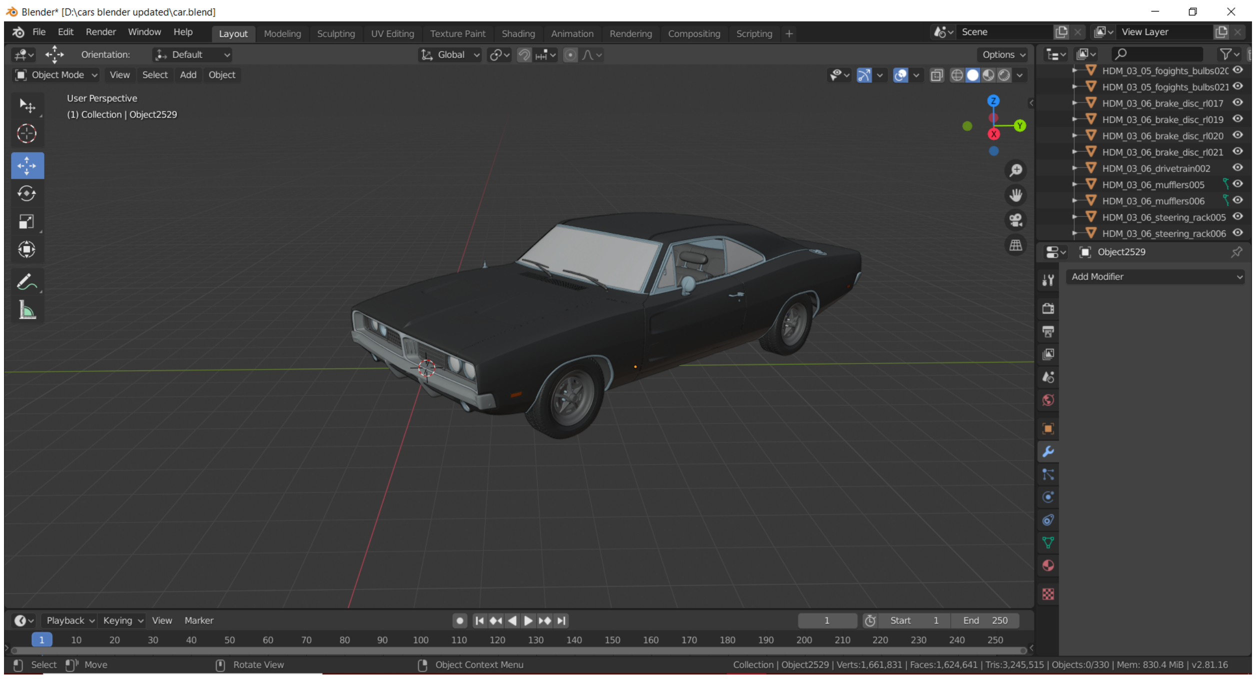
Video 1 link:https://www.youtube.com/watch?v=c6Au5u-GCRc

Video 2 link:https://www.youtube.com/watch?v=ujVQO-VOhPE

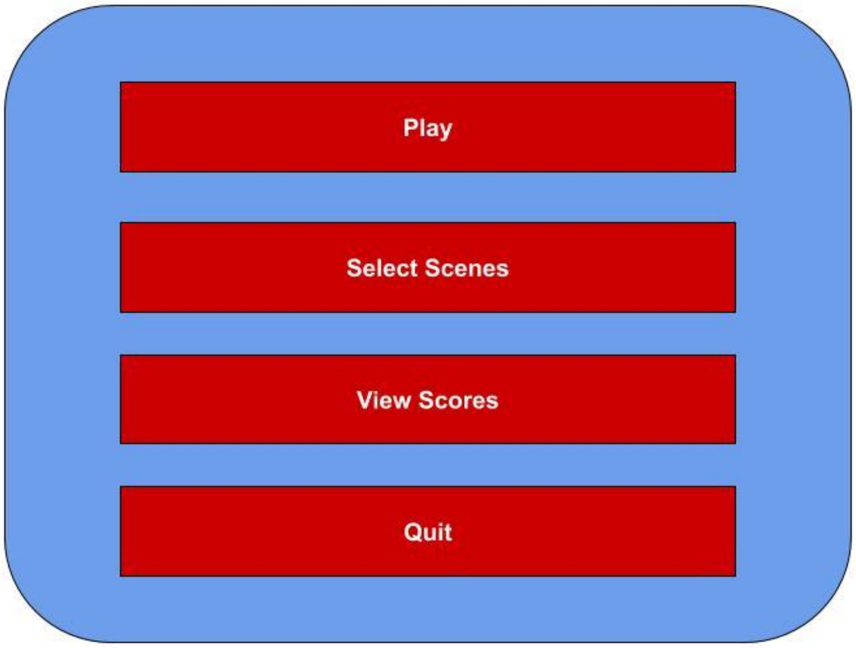
Creating 4 Other Static Scenes & Starting GUI

After the progress report we continued to work. Our original goal was to have 5 static scenes that never changed. We had this first scene completed, so we set out to create 4 more scenes. We originally copied the same scene 4 times and started placing different legal and illegal assets in different portions of the vehicles, but we decided (due to the suggestion by Dr. Benlamri) to have a different vehicle for each scene. Below are images of 2 of the vehicles we were going to use. One was a semi truck and one was a car. This way the scenes wouldn't be boring and there would be unique locations to hide items for each vehicle.


Blender* [D:\cars blender updated\bigtruck.blend]
User Perspective
(1) Collection | Steps
Scene Collection
Collection
Cab
Steps
Add Modifier
Collection | Steps | Verts:637,296 | Faces:616,138 | Tris:1,205,462 | Objects:0/53 | Mem: 213.8 MiB | v2.81.16

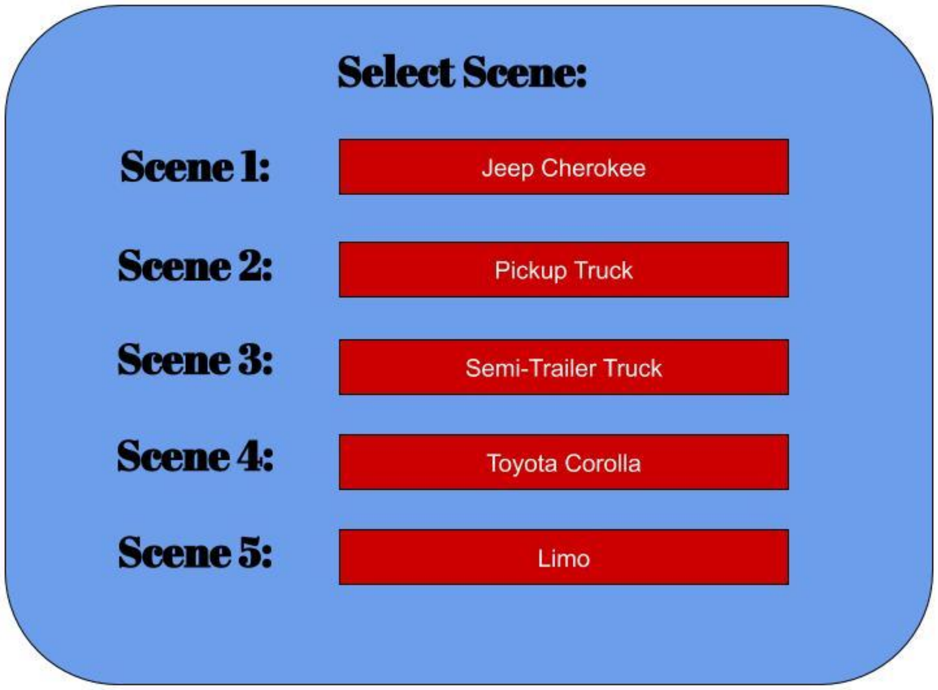

Blender* [D:\cars blender updated\car.blend]
User Perspective
(1) Collection | Object2529
HDM_03_05_foglights_bulbs020
HDM_03_05_foglights_bulbs021
HDM_03_06_brake_disc_rl017
HDM_03_06_brake_disc_rl019
HDM_03_06_brake_disc_rl020
HDM_03_06_brake_disc_rl021
HDM_03_06_drivetrain002
HDM_03_06_mufflers005
HDM_03_06_mufflers006
HDM_03_06_steering_rack005
HDM_03_06_steering_rack006
Object2529
Add Modifier
Collection | Object2529 | Verts:1,661,831 | Faces:1,624,641 | Tris:3,245,515 | Objects:0/330 | Mem: 830.4 MiB | v2.81.16

We were in the process of separating the doors, hood, and trunk from the car and combining all other components when we were interrupted by the COVID-19 pandemic. We had also begun working on a GUI that a user would encounter when they first started the game. We had the following vision for the GUI:



When you select play, we originally wanted it to generate a completely random level. When you select “quit” the game is exited. When you select “View Scores” a list of previous games along with the associated date, time, and score would be shown.

When you select “Select Scenes” the following menu would appear:



We tried to have a variety of vehicles so the user could be trained in all types of scenarios. We tried to research the most common vehicles to cross the Canada/USA border but we could not find any information so we made the vehicles diverse. We have 1 car, 1 truck, 1 semi-trailer truck, and 2 speciality vehicles - a limo and a jeep cherokee which is a hybrid of a van and a truck.

Although we did create the 5 unique scenes, we were unable to access the machine located in the ATAC building to capture new screenshots and videos.

What we did not have the Opportunity to Finish:

Due to the COVID-19 pandemic, we were unable to finish our project as some members went home to quarantine. The members who remained in Thunder Bay could not complete the project since the computer with all of our files and codes, as well as the hardware, was all located in the ATAC building which was shut down. If we had the adequate time to complete the project, we wanted to work on the randomization feature of the program. Currently, the 5 unique concrete scenes would display when you click “select scenes” from the first menu. We wanted to add a feature when the user clicked “play” to create a scene with a random vehicle asset we had, with random amounts of legal and illegal assets scattered in the vehicle. This way there could always be a unique experience every time the game is played. We also needed to finish the UI and create the “View Scores” tab. We discovered we were missing one piece of functionality. In our progress videos we had no way for the user to signify that they were finished and would like the final score to be calculated. To correct this, we needed to assign one of the buttons on the controllers to be signified as a “finish” button. We were also going to implement a final screen after the user selects the finish button. This screen would display their results, which includes their score, the number of illegal items in the vehicle they should have found, and the number of illegal items in the vehicle that they actually found.

Our Ideal Simulation:

Since we were unable to finish our program we cannot physically show our ideal simulation in video format, however I can explain it. The user would begin the game and the main GUI menu would pop up. The user would choose “select scene” and select whichever scene they wanted. They would drop whichever items they thought were illegal in the bin, and then they would select the finish button on their controller. Their score would then be displayed to them and recorded to be viewed later either by supervisor or officer in the “view scores” tab of the main GUI menu. The user would then be brought back to the main GUI menu to select to either play another scene, view their scores, or quit the game.

Ideas we changed as the program progressed:

We mentioned when outlining our goals for the project in the “use case diagram” section of the report that we wanted two GUI menus. One for the user and one for the supervisor. Later we found that idea redundant and hard to implement. We also wanted to create an average score based on a combination of all past games played, as outlined in the “system design” section of this report. Both of these issues would be solved if each officer had their own account where they can view a list of all the games they have played and a score for each individual game. However, the supervisor would be given privilege to view a list of all games all officers they supervise have played. This is more robust and extendable, although we were unable to implement it in a short amount of time.

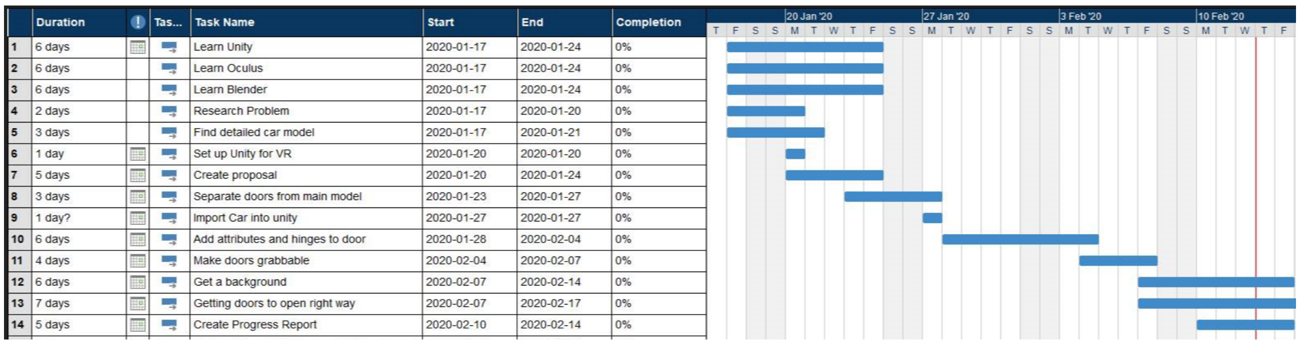
Project Management / Gantt Chart

We created a gantt chart to track our progress in the project. This way we had a concrete plan and knew what we needed to do each day / week. Since we had everything planned at the beginning, we could 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 had a concrete plan we could all stay accountable and easily see what needs to be worked on. Up until we were no longer able to work on the project due to the COVID-19 pandemic, we were on track to finish the project on time. Just in case we had run out of time, we allotted 1 week for a buffer time at the end of the project that was going to be used if we fell behind schedule.

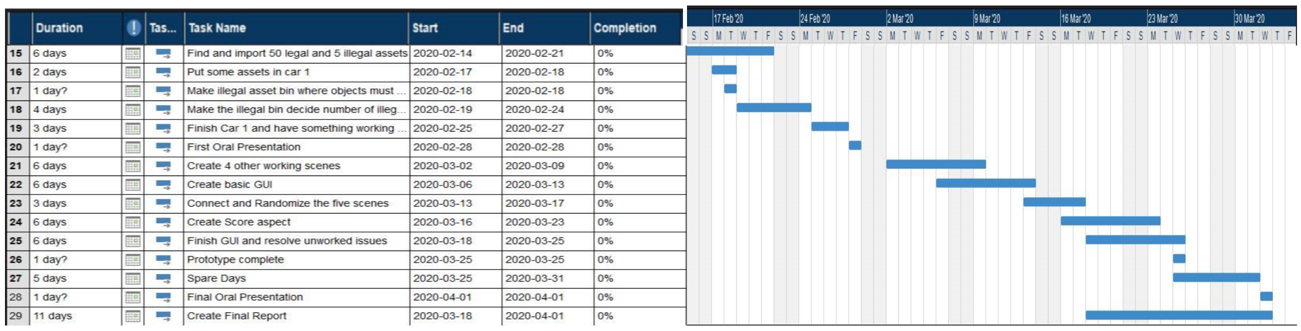
In general we are looking for a 25% split of work for each team member, with some people focusing more on report writing and research, while others are focusing more on the prototype. We have a group chat to keep everyone informed and accountable to do their work. So far, everyone has been working on the prototype, but group members have specialized and worked more on some aspects. Christopher took the lead on the report writing and the prototype. 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 went to the lab in ATAC 4013, we opened our project and saved 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%

We were on track to complete our project when we were halted by the COVID-19 pandemic.

Although we had these time-management aids, some group members did much more work than others. It was hard to motivate some group members to do any work, which was quite frustrating.

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 did not have time to fix this problem.

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.

We had a couple of problems when we interacted with the doors. First, when grabbing the door to open it, we can just pull the door apart from the car and carry it around. We came up with a solution. Instead of making the entire door object grabbable, we could add a small invisible grabbable 3D object on the top of the door handle and configure the door as ungrabbable object. As the user tries to open the door with the door handle(what people normally would do to open a door), the user is actually interacting with that particular invisible object. The door will remain attached with the vehicle since the door is not a grabbable object. Second problem that we had is that the user can grab the door anywhere and not just the door handle. This issue can be fixed as the same as the previous problem by adding an invisible grabbable 3D object on the top of the door handle. Those two issues will no longer occur once we decide to not to allow users to grab the door directly. Another issue that we encounter is a little bit weird. Sometimes, 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. We were trying to understand why this was happening. Because the door opening and closing mechanism function correctly occasionally. As we went through more testing, we found out that if the user approaches the door from a realistic angle and grabs the side of the door where it would open and close normally. This issue may be the internal issue that came with the game engine(Unity) that we used.

There was also an issue with the planes inside the vehicles that assets are placed on. If a user approached these planes they would actually climb on top and be inside the vehicle. To solve this issue we were going to place an invisible plane at head level above the vehicle. This would stop the user from climbing onto the car when approaching these planes. We were unable to fix this either although it would be an easy fix.

There was an issue based on the hitbox of some assets. For example, if you watch the second video linked in this report, you will see that when the user grabs the shotgun, they are actually grabbing the air. That is because the hitbox for the entire object was in this random position. We fixed this issue by placing the script that makes the object grabbable to the shotgun component itself instead of the entire object.

Future Work / Ways we would Extend the Project:

Not only would we have liked to include the components of the project we did not have the opportunity to finish, there are ways we would extend the project past our initial vision given even more time on the project. We would 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. This idea is quite interesting and would increase the robustness of the program, but we unfortunately do not have time to implement these features so far.

Conclusion:

Due to the particularity of VR itself, our project has some limitations. One of the limitations is that this VR system still lacks flexibility. Unless we have enough algorithms to randomize everything, our scene is limited to whatever we created. The users of this simulation system (CBSA agents) can only practice a limited number of scenes, which doesn’t match the original purpose of making the training process realistic. The other limitation we have is that since this project is specifically targeted for the border security training process, it can not promote reusability for other fields. Although we could say that a complete VR system for border training can technically cover for all countries. Laws and regulations vary in different countries. The adjustments still need to be done.

As far as we are concerned, a VR system is still a comprehensive development process from the aspect of software. As we work on this project, we find out that we need to have more knowledge of other fields as software developers. For example, one of the biggest challenges we had at the early phase was figuring out how to make the doors open and close correctly. It required us to have at least a general sense of physics, and how the colliders and joints work. It reminds us that VR software of all purposes depends on the specific situation. Sometimes, this development process becomes an interdisciplinary challenge for us.

VR systems are meant to recreate visible and repeatable environments to replace some of the training processes or to enhance analysis strategies. The qualities of such systems are not only depending on how real it may look but also relying on the developer to foresee all possible scenarios that may occur.

In our project, most 3D objects we imported into our gaming engine are simple as we don't have the resources nor knowledge to come up with sophisticated 3D objects. We used whatever we can find on the Internet, which limited our project from being more realistic. In the future, should we or anyone else who would proceed with this project, a significant amount of resources and manpower need to be allocated.

In general, we feel we adapted to challenges that were presented to us and we were on track to complete most of the aspects we outlined in our suggested solution to the problem. We learned a great deal of diverse knowledge, and really enjoyed this project overall. It was unfortunate we did not have the time to bring the entire project to fruition because we would have liked to have a complete working prototype.

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.

[6]. Babich, N. September 19th, 2019. “How VR In Education Will Change How We Learn And Teach”. Retrieved from Adobe Xd ideas. https://xd.adobe.com/ideas/principles/emerging-technology/virtual-reality-will-change-learn-teach/

[7] Mbryonic Ltd. 2019. “What’s Next for Virtual Reality”. Retrieved from https://mbryonic.com/advances-in-virtual-reality/

[8]. Persky, S., & Lewis, M. (2019). Advancing science and practice using immersive virtual reality: what behavioral medicine has to offer. Translational Behavioral Medicine, 9(6), 1040–1046. https://doi.org/10.1093/tbm/ibz068

[9]. P. Chiang, J. Zheng, Y. Yu, K. H. Mak, C. K. Chui and Y. Cai, "A VR Simulator for Intracardiac Intervention," in IEEE Computer Graphics and Applications, vol. 33, no. 1, pp. 44-57, Jan.-Feb. 2013.

[10].J. Schild *et al*., "EPICSAVE Lifesaving Decisions – a Collaborative VR Training Game Sketch for Paramedics," *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Osaka, Japan, 2019, pp. 1389-1389.

[11]. L. Greunke and A. Sadagic, "Taking Immersive VR Leap in Training of Landing Signal Officers," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 4, pp. 1482-1491, 21 April 2016.