

**GNG1103**  
**User and Product Manual**

**Mines Action Canada Killer Robot PSA**

GNG 1103- Engineering Design

December 10th, 2023

Group 8

Megan Bauman

Jadon Xia

Rayyan Faruquee

Ella Macneil

# Table of Contents

[https://drive.google.com/file/d/1T5t1pb5eOMJ5SpQtO5sOz62udljrluad/view?usp=sharing.](https://drive.google.com/file/d/1T5t1pb5eOMJ5SpQtO5sOz62udljrluad/view?usp=sharing)  
1

- Table of Contents..... 2**
- 1 - Abstract..... 3**
- 2 - Relevant Project Information..... 4**
  - 2.1- Background information..... 4
  - 2.2- Design Criteria..... 4
  - 2.3- Storyline..... 4
  - 2.4 - Cautions and Warning..... 4
- 3 - User Accessibility..... 5**
- 4 - Project Materials..... 5**
  - 4.1 - Bill of Materials (BOM)..... 5
  - 4.2 - Audio..... 6
  - 4.3 - Scripts..... 7
    - 4.3.1 - WASD + Camera Movement Code..... 7
    - 4.3.2 - Subtitles..... 9
- 5 - Building the Unity Project..... 10
  - 5.1 - Getting Started..... 10
  - 5.2 - Physical Environment..... 10
    - 5.2.1 - Apartment Room..... 10
  - 5.3 - Programming in Unity..... 12
    - 5.4.1 - Attaching the Audio..... 12
    - 5.4.2 - Attaching the Subtitles & Endcard..... 12
- 6 - Testing Stage..... 12**
  - 6.1 - Prototyping Test Plan..... 13**
- Conclusions..... 16**
- References..... 17**

## 1 - Abstract

It is often necessary for users to thoroughly examine a project to obtain a better understanding of its functions. The purpose of the user manual is to help users navigate the project and include the necessary information to properly recreate and operate its crucial components.

## 2 - Relevant Project Information

### 2.1- Background information

An autonomous killer robot is any weapon that uses sensory data to target and eliminate potential threats without the approval of a human. The purpose of the VR simulation may be summarized in the problem statement: 'to encourage decision-makers in legislatures to preemptively ban autonomous killer robots by designing an immersive and memorable VR simulation that provokes concern while balancing realism and simplicity.' Such information is highly pertinent in understanding how the project is constructed and the reasoning behind many design choices.

### 2.2- Design Criteria

The simulation follows the criteria laid out by GNG1103's contact at Mines Action Canada, wherein the final product must be less than a minute-long VR experience with a focus on the dangers of autonomous killer robots. The VR simulation was designed to convey emotionality to convince decision-makers in legislatures why autonomous robots should be banned. The simulation needed to be simple with the time constraints of the project and memorable with user comprehensibility as a top priority.

### 2.3- Storyline

To meet the design criteria the storyline was designed with user comprehensibility as a main concern. The storyline itself is a simple concept but the features displayed around the room are more complex to add to emotionality. The user will wake up inside the simulation and see that they are in a studio apartment that looks like a normal home but there have been some adaptations made to the home as safety precautions against the robot. The user will see posters designed with information on facial recognition and how to stay under a robot's radar. There are signs of food scarcity and lack of resources due to people being afraid to go outside and be targeted and eliminated by robots. As the user is walking through the room, a news clip is playing from a radio beside the bed that is telling the story of how robots have changed the lives of the people who live in this simulation.

### 2.4 - Cautions and Warning

- Be aware that those prone to motion sickness

### 3 - User Accessibility

The project was designed to be easily run from any VR headset and compatible computer system, from downloading Unity to run the simulation, to the open-source scripts and assets that make up the various aspects of the experience. Of note are the enlarged posters in the simulated space and the addition of subtitles for the hard of hearing and to better facilitate the conveyance of information.

### 4 - Project Materials

This section of the User Manual covers the required materials to create the product in Unity.

#### 4.1 - Bill of Materials (BOM)

PART NUMBER	PART NAME	DESCRIPTION	QTY	UNIT S	UNIT COST	TOTAL AMOUNT
1	unity software	<a href="https://unity.com/success-plans/integrated-success?utm_source=google&amp;utm_medium=cpc&amp;utm_campaign=cc-abm_iss_amer_amer-tl_en_qw_sem-gg_acq_br-pr_2023-05_cc-abm-amer-tl-br_cc3022_ev-br_id:7170000112699686&amp;utm_content=cc_abm_iss_amer_pu_sem_gg_ev-br_pros_x_npd_cpc_kw_sd_all_x_x_courses_id:58700008486337662&amp;utm_term=unity%20tutorial%20course&amp;&amp;&amp;&amp;qad=1&amp;qclid=CiwKCAjwv-2pBhB-FiwAtsQZFPmY5PCIGBarWeAnmJLlArQl6JwCFGMSb6idsw5rJdAnWlVYTG_vkxoCMUMQAvD_BwE&amp;qclsrc=aw.ds">https://unity.com/success-plans/integrated-success?utm_source=google&amp;utm_medium=cpc&amp;utm_campaign=cc-abm_iss_amer_amer-tl_en_qw_sem-gg_acq_br-pr_2023-05_cc-abm-amer-tl-br_cc3022_ev-br_id:7170000112699686&amp;utm_content=cc_abm_iss_amer_pu_sem_gg_ev-br_pros_x_npd_cpc_kw_sd_all_x_x_courses_id:58700008486337662&amp;utm_term=unity%20tutorial%20course&amp;&amp;&amp;&amp;qad=1&amp;qclid=CiwKCAjwv-2pBhB-FiwAtsQZFPmY5PCIGBarWeAnmJLlArQl6JwCFGMSb6idsw5rJdAnWlVYTG_vkxoCMUMQAvD_BwE&amp;qclsrc=aw.ds</a>	1		0\$ (provided by university of Ottawa)	0\$
2	Oculus Quest 2 Headset	(provided by university)	1		0\$ (provided)	0\$
3	Move Camera + WASD Movement Code	<a href="https://gist.github.com/gunderson/d7f096bd07874f31671306318019d996">https://gist.github.com/gunderson/d7f096bd07874f31671306318019d996</a>	1		0\$	0\$
4	Retro Radio	<a href="https://assetstore.unity.com/packages/3d/props/ha-pbr-oid-retro-radio-free-180303">https://assetstore.unity.com/packages/3d/props/ha-pbr-oid-retro-radio-free-180303</a>	1		0\$	0\$
5	Tinned Food	<a href="https://assetstore.unity.com/packages/3d/props/food/tinned-food-89246">https://assetstore.unity.com/packages/3d/props/food/tinned-food-89246</a>	1		0\$	0\$

6	Large Hat	<a href="https://assetstore.unity.com/packages/3d/props/clothing/cowboy-hat-1-200835">https://assetstore.unity.com/packages/3d/props/clothing/cowboy-hat-1-200835</a>	1		0\$	0\$
7	Fridge	<a href="https://assetstore.unity.com/packages/3d/props/electronics/kitchen-appliance-low-poly-180419">https://assetstore.unity.com/packages/3d/props/electronics/kitchen-appliance-low-poly-180419</a>	1		0\$	0\$
8	Padlock	<a href="https://assetstore.unity.com/packages/3d/props/interior/padlock-with-password-22506Z">https://assetstore.unity.com/packages/3d/props/interior/padlock-with-password-22506Z</a>	1		0\$	0\$
9	Wooden Board Textures	<a href="https://assetstore.unity.com/packages/2d/textures-materials/wood/plank-textures-pbr-72318">https://assetstore.unity.com/packages/2d/textures-materials/wood/plank-textures-pbr-72318</a>	1		0\$	0\$
10	Canva (Poster Designing)	<a href="https://www.canva.com/">https://www.canva.com/</a>	1		0\$	0\$
11	DaVinci Resolve Video Editing Software	<a href="https://www.blackmagicdesign.com/products/davinciresolve">https://www.blackmagicdesign.com/products/davinciresolve</a>	1		0\$	0\$
12	Voice Changer	<a href="https://voicechanger.io/">https://voicechanger.io/</a>	1		0\$	0\$
13	Unity Scripting API	<a href="https://docs.unity3d.com/ScriptReference/">https://docs.unity3d.com/ScriptReference/</a>	1			
14	Mask	<a href="https://www.cgtrader.com/free-3d-models/character/clothing/v-for-vendetta-guy-fawkes-anonymous-mask">https://www.cgtrader.com/free-3d-models/character/clothing/v-for-vendetta-guy-fawkes-anonymous-mask</a>	1		0\$	0\$
15	Radiator	<a href="https://assetstore.unity.com/packages/3d/props/old-radiator-185991">https://assetstore.unity.com/packages/3d/props/old-radiator-185991</a>			5.65\$	5.65\$
16	House Interior	<a href="https://assetstore.unity.com/packages/3d/props/interior/free-house-interior-223416">https://assetstore.unity.com/packages/3d/props/interior/free-house-interior-223416</a>	1		0\$	0\$
<b>TOTAL PARTS</b>						<b>TOTAL</b> 6.91\$

## 4.2 - Audio

The newscast audio that is heard during the simulation can be found here and uploaded into a Unity Audio Source component.

[https://drive.google.com/file/d/1\\_R9g2P8Dn9CnkDt4iWX6sWBAAaSDI7Cpy/view?usp=sharing](https://drive.google.com/file/d/1_R9g2P8Dn9CnkDt4iWX6sWBAAaSDI7Cpy/view?usp=sharing)

## 4.3 - Scripts

### 4.3.1 - WASD + Camera Movement Code

```

using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class FlyCamera : MonoBehaviour
{
    /*
    wasd : basic movement
    shift : Makes camera accelerate
    space : Moves camera on X and Z axis only. So camera doesn't gain any height
    */

    float mainSpeed = 1.25f; //regular speed
    float shiftAdd = 2.0f; //multiplied by how long shift is held. Basically running
    float maxShift = 4.00f; //Maximum speed when holding shift
    float camSens = 0.5f; //How sensitive it with mouse
    private Vector3 lastMouse = new Vector3(255, 255, 255); //kind of in the middle of the
screen, rather than at the top (play)
    private float totalRun = 1.0f;

    void Update()
    {
        lastMouse = Input.mousePosition - lastMouse;
        lastMouse = new Vector3(-lastMouse.y * camSens, lastMouse.x * camSens, 0);
        lastMouse = new Vector3(transform.eulerAngles.x + lastMouse.x,
transform.eulerAngles.y + lastMouse.y, 0);
        transform.eulerAngles = lastMouse;
        lastMouse = Input.mousePosition;
        //Mouse camera angle done.

        //Keyboard commands
        Vector3 p = GetBaseInput();
        if (p.sqrMagnitude > 0)
        { // only move while a direction key is pressed
            if (Input.GetKey(KeyCode.LeftShift))
            {
                totalRun += Time.deltaTime;
                p = p * totalRun * shiftAdd;
                p.x = Mathf.Clamp(p.x, -maxShift, maxShift);
                p.y = Mathf.Clamp(p.y, -maxShift, maxShift);
            }
        }
    }
}

```

```

    p.z = Mathf.Clamp(p.z, -maxShift, maxShift);
}
else
{
    totalRun = Mathf.Clamp(totalRun * 0.5f, 1f, 1000f);
    p = p * mainSpeed;
}

p = p * Time.deltaTime;
Vector3 newPosition = transform.position;
if (Input.GetKey(KeyCode.Space))
{ //If player wants to move on X and Z axis only
    transform.Translate(p);
    newPosition.x = transform.position.x;
    newPosition.z = transform.position.z;
    transform.position = newPosition;
}
else
{
    transform.Translate(p);
    newPosition.x = transform.position.x;
    newPosition.z = transform.position.z;
    transform.position = newPosition;
}
}
}

private Vector3 GetBaseInput()
{ //returns the basic values, if it's 0 than it's not active.
    Vector3 p_Velocity = new Vector3();
    if (Input.GetKey(KeyCode.W))
    {
        p_Velocity += new Vector3(0, 0, 1);
    }
    if (Input.GetKey(KeyCode.S))
    {
        p_Velocity += new Vector3(0, 0, -1);
    }
    if (Input.GetKey(KeyCode.A))
    {
        p_Velocity += new Vector3(-1, 0, 0);
    }
    if (Input.GetKey(KeyCode.D))

```



```

    {
        p_Velocity += new Vector3(1, 0, 0);
    }
    return p_Velocity;
}
}

```

#### 4.3.2 - Subtitles

The purpose of subtitles is to aid in the comprehension of the audio, especially for those who are hearing-impaired. The C# script for subtitles works based on time elapsed in the program. There are three variables used. The first is `timeLeft1`, which is the time elapsed BEFORE showing the subtitle. After this time counts down to zero, `timeLeft2` is the time that the subtitle is on the screen. Finally, `gameObject test` is assigned in the Inspector Window in Unity to a Text object that holds the subtitle, which is part of Unity's UI objects. Upon the program's start, the `SetActive()` function makes the subtitle inactive and it does not appear on the Game screen. After `timeLeft1` elapses, `FinalMessage()` is called by `Invoke`, which is part of Unity's Scripting API. `FinalMessage()` causes the subtitle to become active and show up on the Game screen. After `timeLeft2` elapses, `FinalMessage2()` is called and the subtitle is made inactive.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class capt1 : MonoBehaviour
{
    float timeLeft1 = 3.02f; // time before showing subtitle on screen
    float timeLeft2 = 4.08f; // time subtitle on screen before gone
    public GameObject test;

    void Start()
    {
        test.SetActive(false);
        Invoke("FinalMessage", timeLeft1);
    }

    void FinalMessage()
    {
        test.SetActive(true);
        Invoke("FinalMessage2", timeLeft2);
    }
}

```

```
void FinalMessage2()
{
    test.SetActive(false);
}
}
}
}
```

## 5 - Building the Unity Project

### 5.1 - Getting Started

The VR environment for the product was created using Unity Editor 2022.3.11f1. When downloading the editor, it is recommended to download Microsoft Visual Studio as an additional module, as well as Android Build Support and Windows Build Support. These modules are necessary for simplifying programming, building the product onto the Oculus Quest 2 VR headset and building onto a PC, respectively. When creating the project itself, the template should be 3D Core.

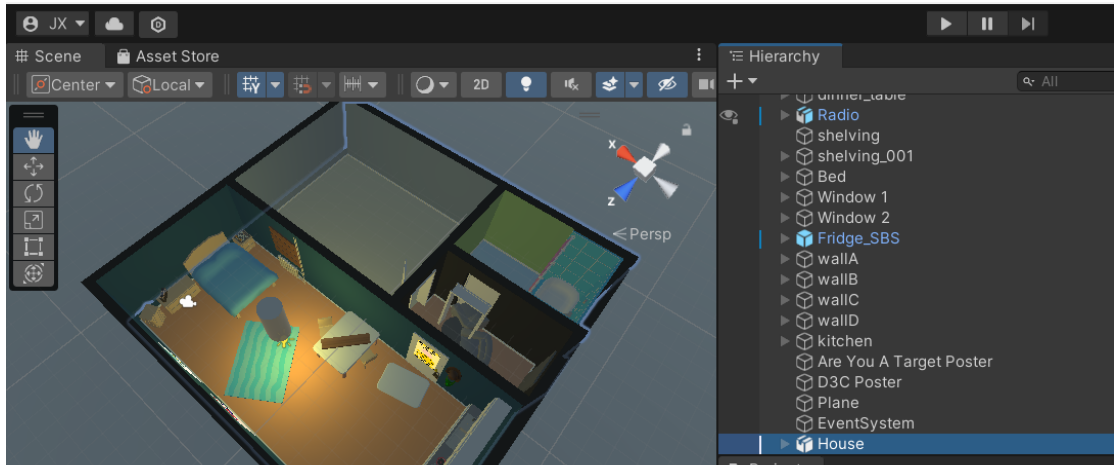
When the Unity project has been created, it may seem overwhelming to navigate Unity's interface. However, Unity offers an excellent and expansive user manual that encompasses its various elements: <https://docs.unity3d.com/560/Documentation/Manual/UnityManual.html>. A good understanding of Unity greatly facilitates the process of building the project.

### 5.2 - Physical Environment

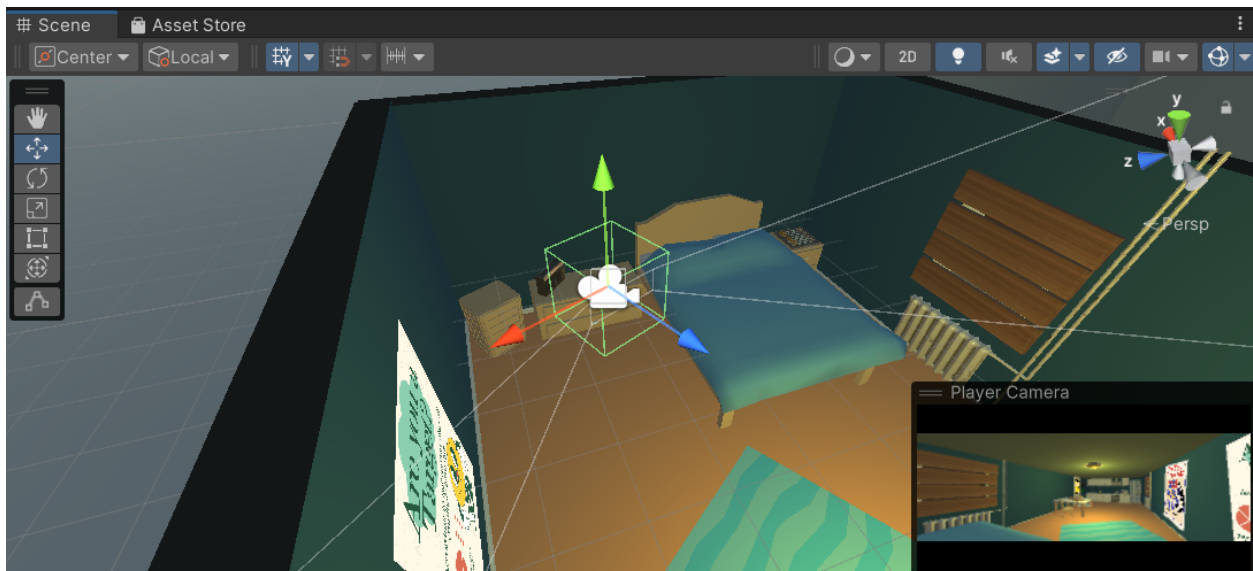
Download and import all relevant assets found in the [Bill of Materials](#) so that they are readily available to add into the environment. This subsection covers the physical aspect of the project and is purely related to the 3D model of the Scene.

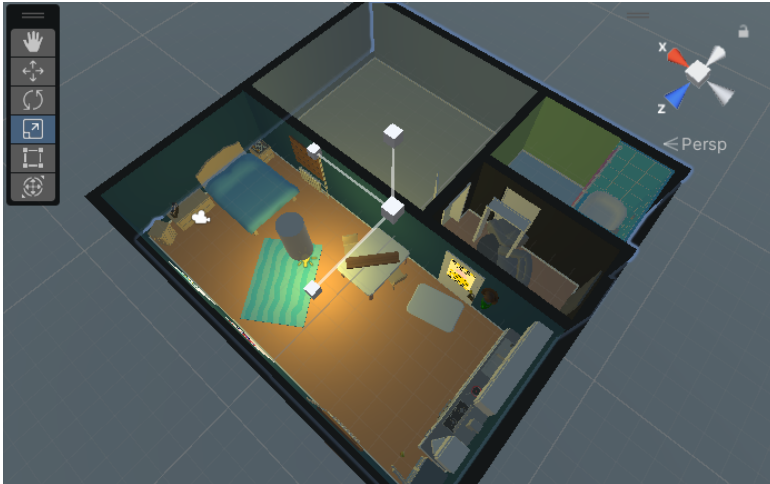
#### 5.2.1 - Apartment Room

In the project file and default scene, the apartment interior was imported from the Unity Asset Store and deposited in the project's asset file. It can be seen in the Unity Hierarchy.



Note that the image shown above contains a different arrangement of furniture than the house asset that you have just imported. The arrangement of furniture will be discussed later. From there, the interior was deployed into the scene and scaled using the Scale Tool until it reached appropriate proportions in relation to the camera's default height. Position the camera using the Move Tool and Rotate Tool so that it resembles the viewpoint of a standing person in the apartment.



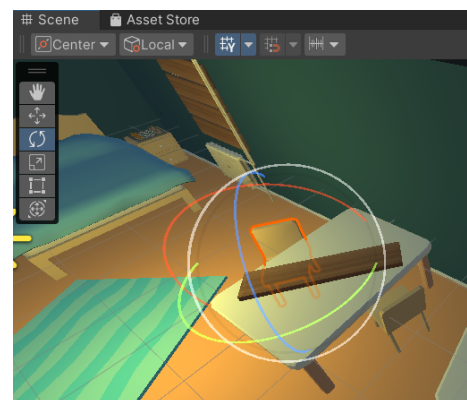


### 5.2.2 - Apartment Floor

The environment contains realistic elements of living in a one-bedroom apartment; the following arrangements are reflective of this. In addition, the arrangement of assets relates to the auditory newscast that will be implemented later, as the newscast helps explain various assets that the user will see. Move the following assets to the living room of the imported house asset (the largest room):



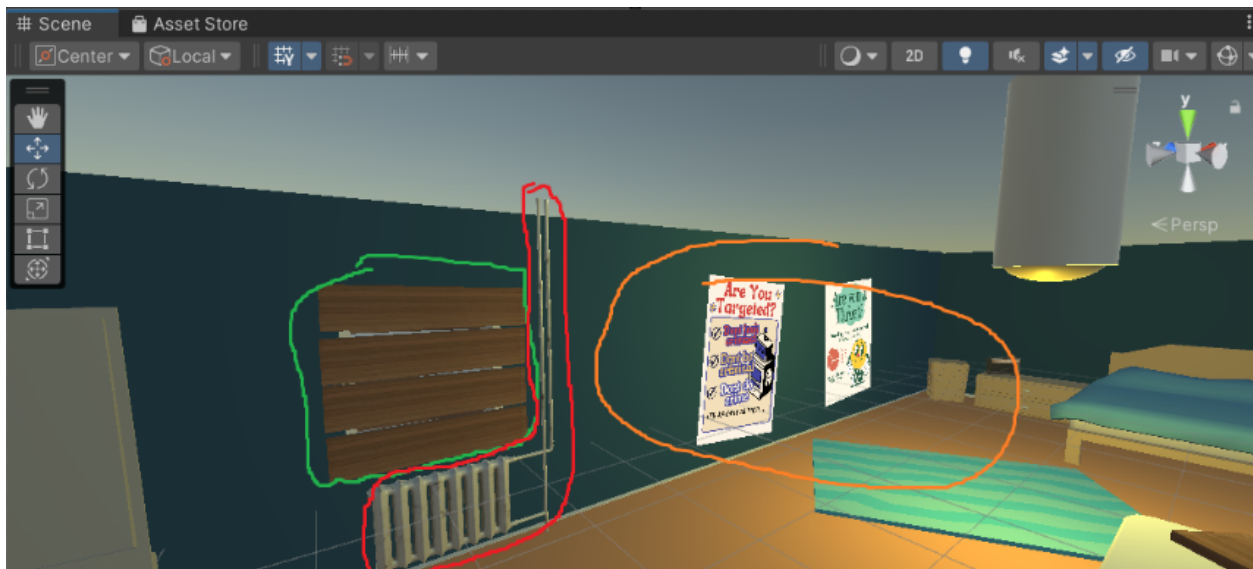
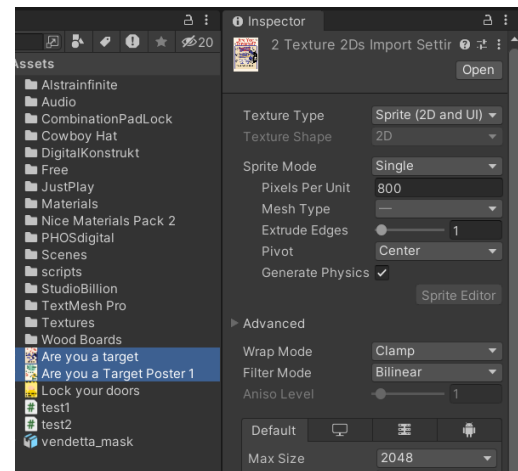
- For assets that are not shown in the above image, move them to the other rooms.
- (red) Move the bed to the living room. This moves both bedside shelvings along into the room. Stretch one of the shelvings using the Scale Tool to resemble a dresser.



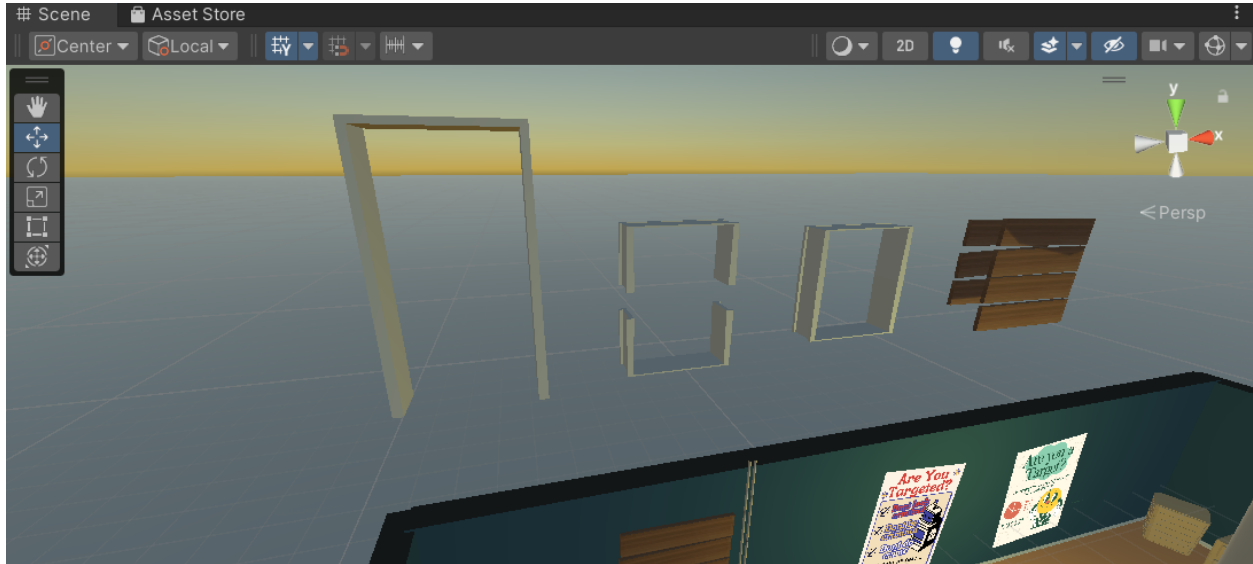
- (yellow) In the assets folder, move the retro radio model onto one of the shelvings.
- (green) Move the laundry basket next to the shelving.
- (pink) Move the two carpets as shown.
- (blue) Move the table as shown. Remove two chairs and use the Rotate Tool to give the chairs a more natural appearance. The wooden board has not yet been implemented. Remove the assets atop of the table.
- (white) Move the kitchen as shown. Remove the red fridge. Go to the assets folders, move the white fridge as shown by the orange circle.

### 5.2.3 - Apartment Walls

The posters, which were created in Canva, were imported into the project file as PNG files. Once in the project's asset folder, their default behaviour was set as "2D Sprite". This allows the posters to be displayed in the scene itself as a 2D object. Drag them into the scene and adjust size accordingly. The posters should be placed on the wall shown below in orange:

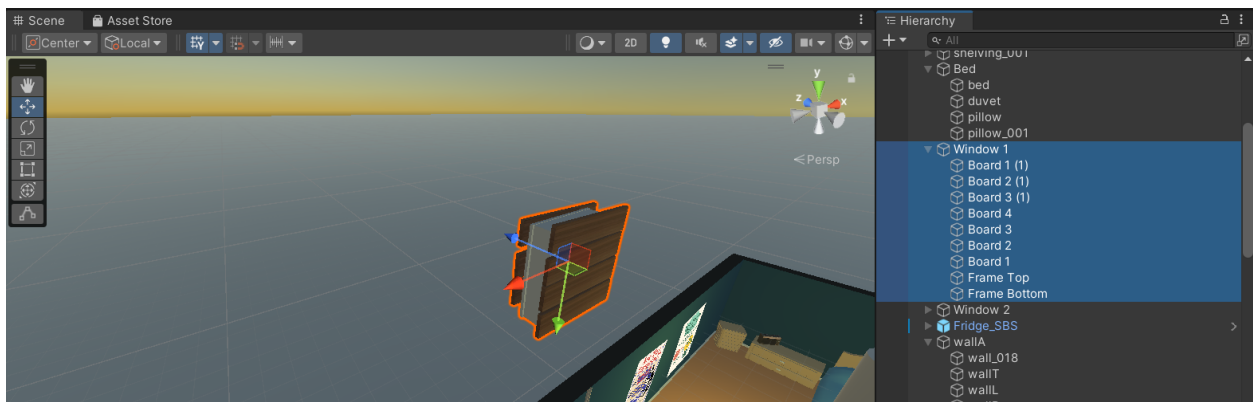


(green) The window is a somewhat complex asset. There are no windows in the imported house asset. Windows may be added either by an external asset or by rearranging an existing asset. In this project, the window frame is actually made up of 2 rearranged doorways stuck together:



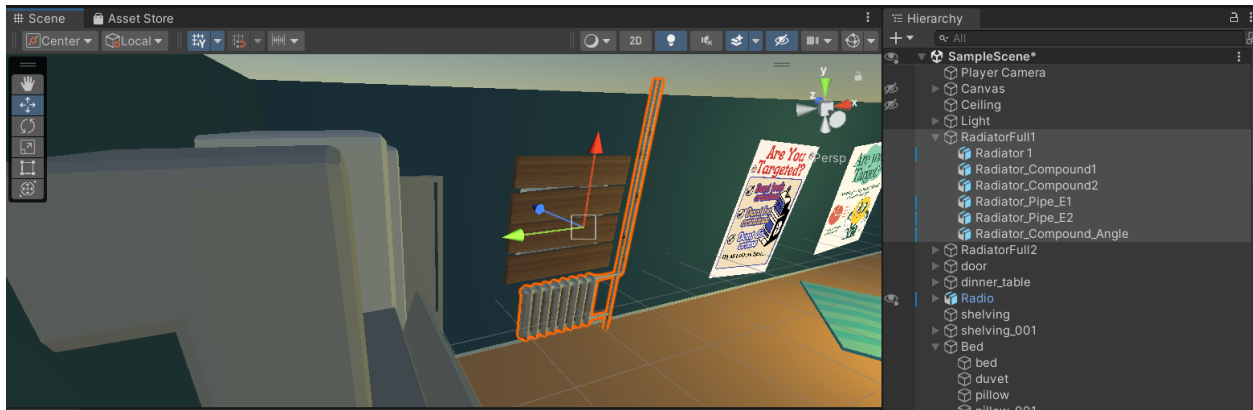
For the boards, follow these instructions:

- In the Hierarchy, click on the '+' symbol > 3D Object > Cube
- In the Assets folder, find the imported wood textures and drag them onto the cube.
- Use the Scale, Move and Rotate Tool to form a wooden board from the cube and place it on the window. Right click the Cube, rename to 'Board #', then duplicate and cover the window opening with wooden boards.
- To facilitate moving the window, which is now made up of multiple components, click on the '+' symbol > Create Empty. A GameObject appears in the Hierarchy. Rename GameObject to 'Window 1', then select all the boards and drag them in the Hierarchy under Window 1.
- Position Window Full on the wall as shown by the green circle.
- A board may now be placed on the table, as shown [earlier](#)

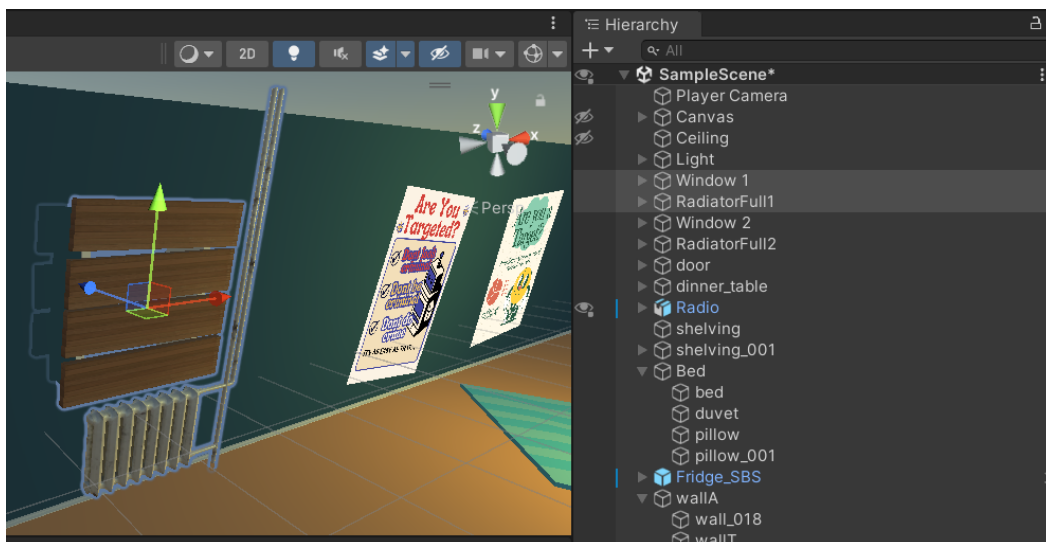


(red) Moving on to the radiator in the red circle: From the Assets folder, move the Radiatormodel onto the Scene. Do not scale it.

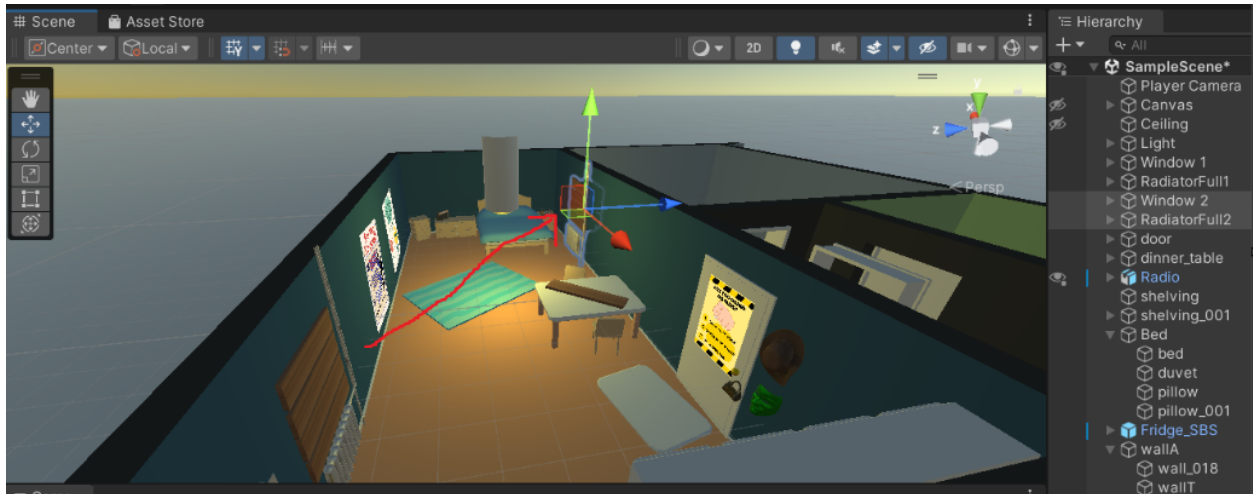
- The two horizontal bars attached to the right are separate assets called Radiator\_Compound.
- The two long vertical bars are separate assets called Radiator\_Pipe\_E.
- The curved connector for the rightmost vertical bar is called Radiator\_Compound\_Angle. Move them as shown.
- In the Hierarchy, click on the '+' symbol > Create Empty. A GameObject appears in the Hierarchy. Rename GameObject to 'Radiator Full', then select all the components and drag them in the Hierarchy under Radiator Full.
- Position Radiator Full as shown and scale accordingly.



Now that this wall is complete, the window and radiator must both be duplicated and placed on the opposite wall. Select both duplicates in the Hierarchy, then use the Move Tool and Rotate Tool to move them.







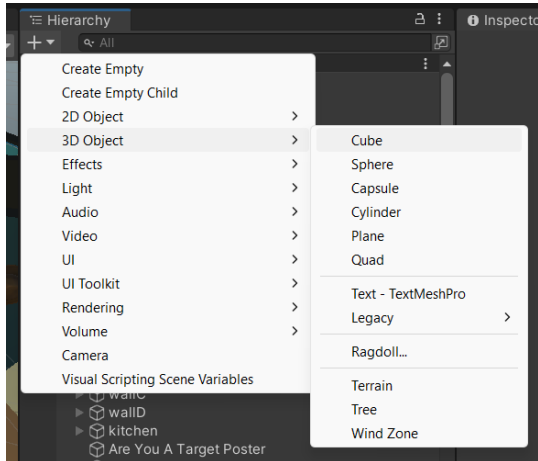
Finally, take the imported padlock from the Assets folder and drag onto the door as shown below. Drag the imported mask and cowboy hat asset and place the two on the wall next to the door, as they represent the user's protection from the outside world. Use the Rotate, Scale and Move tool to adjust all three of these assets.



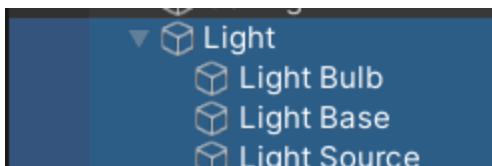
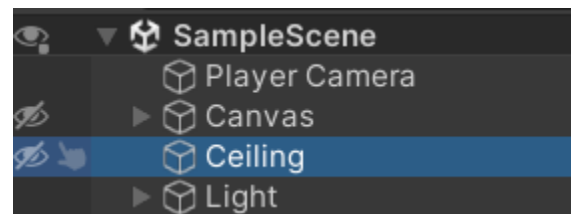


### 5.2.4 Apartment Ceiling

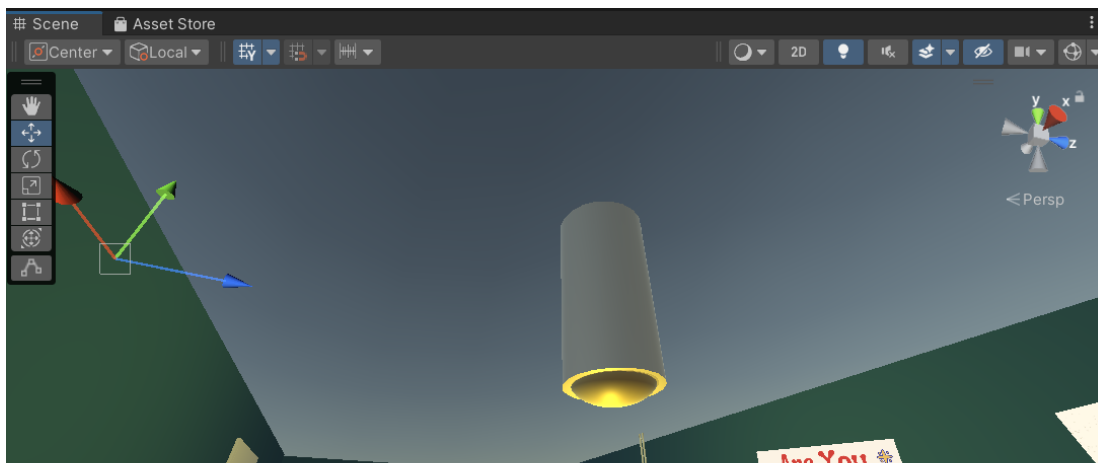
A cube that can fully enclose the top of the apartment room suffices for a ceiling. In the Hierarchy, click '+' > 3D Objects > Cube as shown below. Use the Rotate, Scale and Move Tool to place on top of the room.



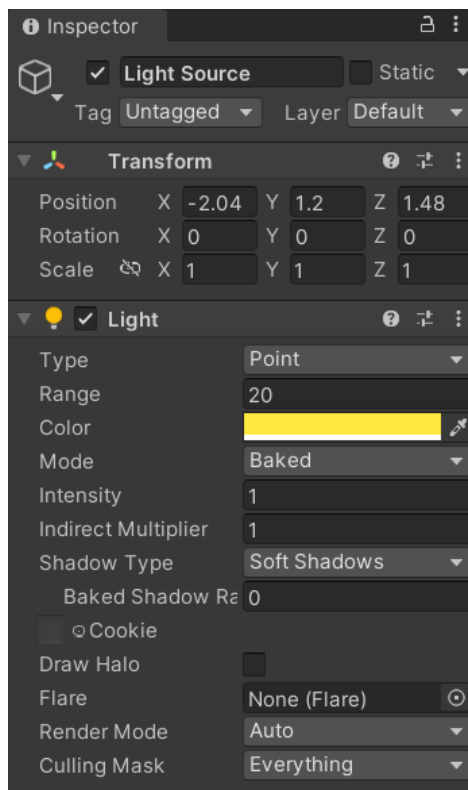
For ease of use, one may use the hide button to make an object invisible (crossed out eye symbol). This is beneficial to the Unity user, as it may obstruct a portion of the environment that must be adjusted or changed.



The only other component for the ceiling is the light. The 3D model itself is made up of a Capsule 3D Object and a Cylinder 3D Object; both can be added from the '+' sign in the Hierarchy.



A third component is the Light Source, which actively simulates light during the simulation. Unity has a wide variety of settings for several types of lights, which are shown below in the Inspector Window:

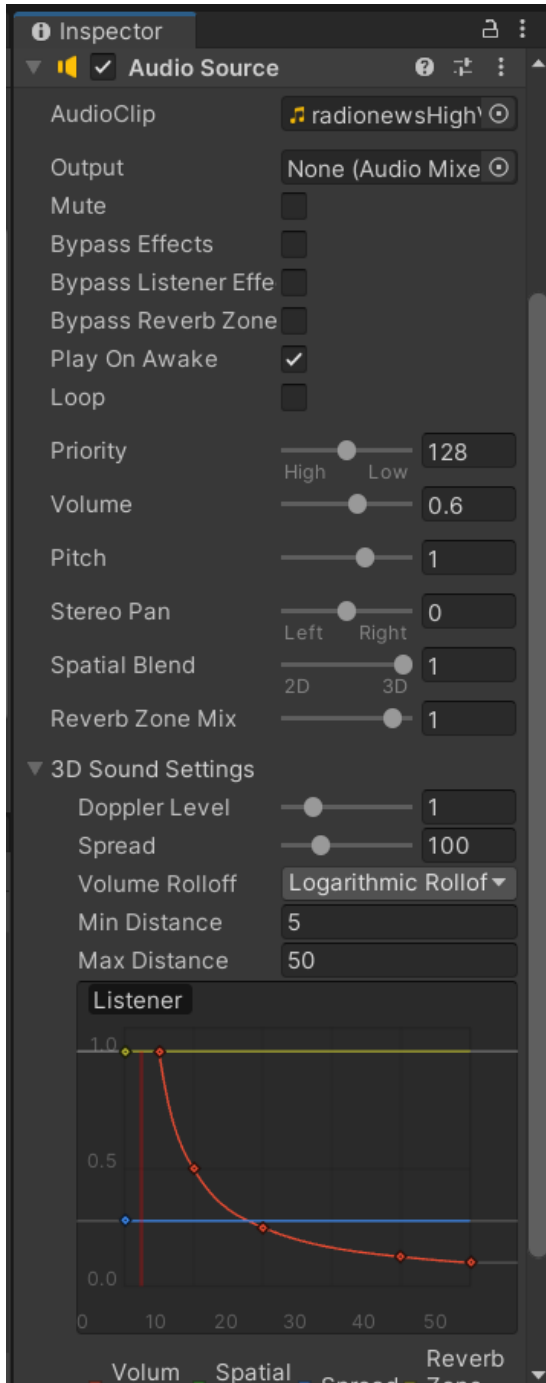


## 5.3 - Programming in Unity

The physical environment establishes the project setting, but it is nothing more than a 3D model. It is required to add scripts and programs with C# and Unity's Scripting API to handle interactive components such as player movement, subtitles, and audio.

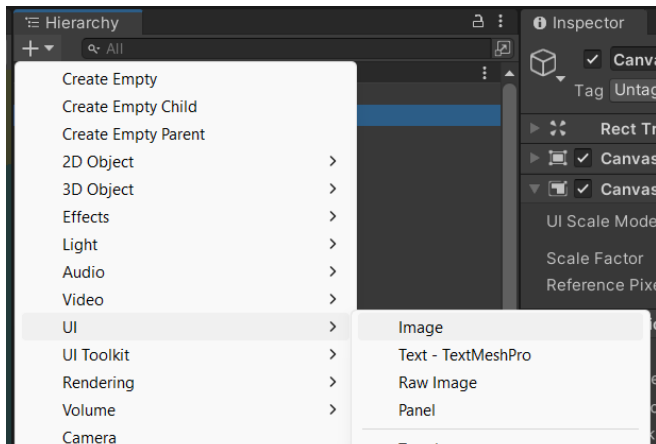
### 5.4.1 - Attaching the Audio

In the project file, the audio source was imported into the project's asset folder. To deploy the audio into the scene and allow it to appear to originate from something in the scene, an Audio Source object has to first be placed in the scene before the mp3 audio clip can be attached. Once in the scene, the audio clip can be fine-tuned for different effects, or even attached to physical objects in the scene. The settings are shown below:

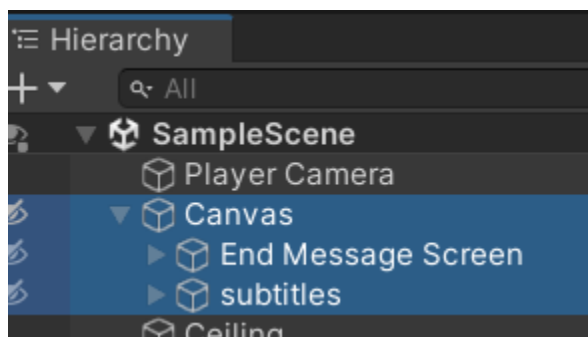


### 5.4.2 - Attaching the End Message Screen

It is necessary to use Unity's UI Elements to create subtitles on the screen. To do this, go to the Hierarchy and press '+' > 'UI' > 'Image'.



In the Hierarchy, an object called 'Canvas' will be created. This serves as the UI for the Player Camera. The image, which is under the Canvas, should be named 'End Message Screen'. Since the End Message Screen causes the screen to go black, it must be a separate image from the subtitles.

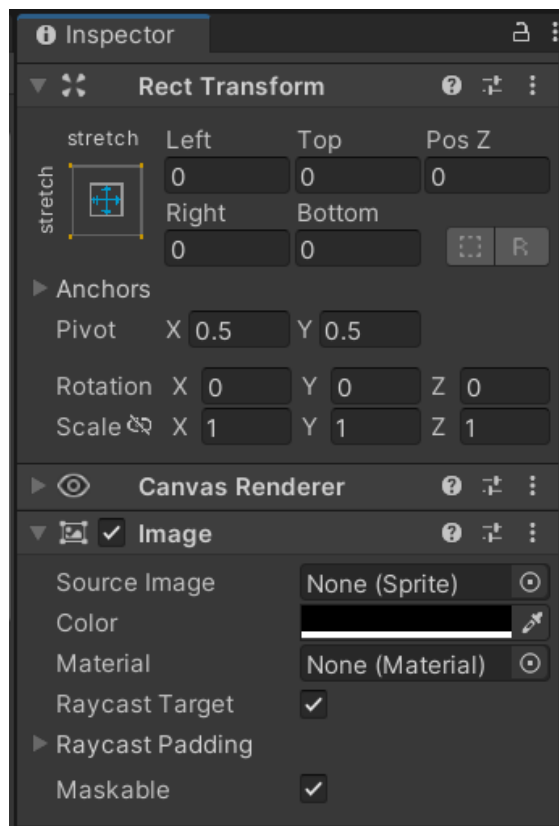


The End Message Screen Image must be adjusted in two ways:

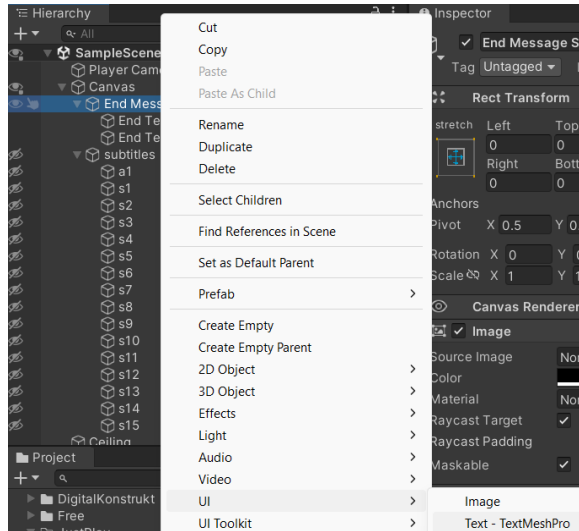
1. under 'Rect Transform', click on the 'stretch/stretch' square, hold Alt, then click the bottom right square. This will stretch the image across the entire screen.



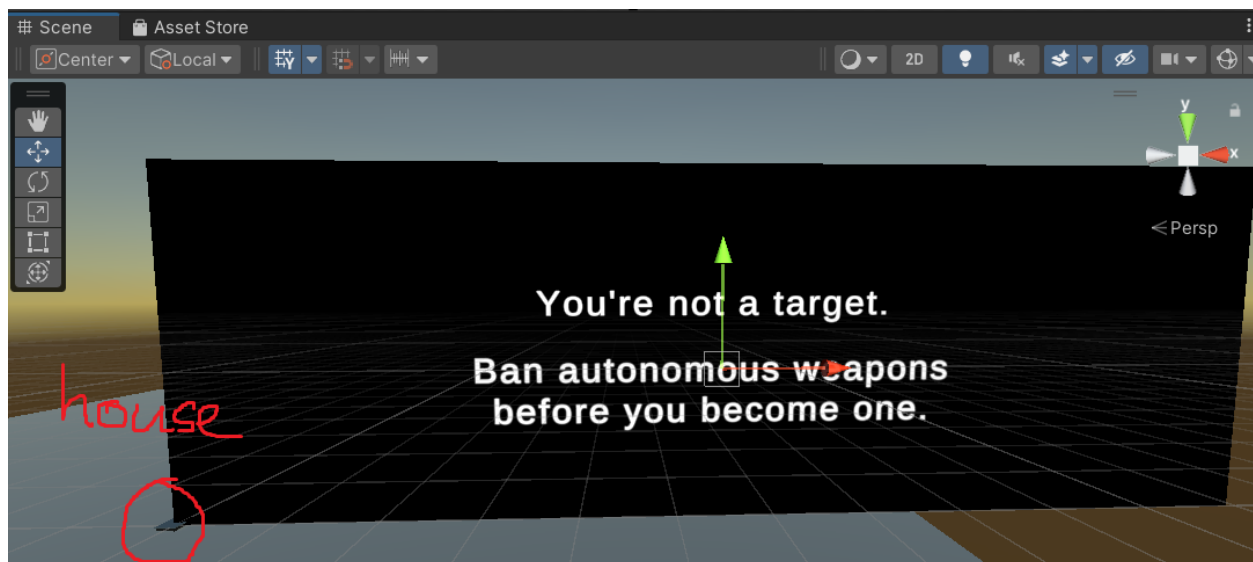
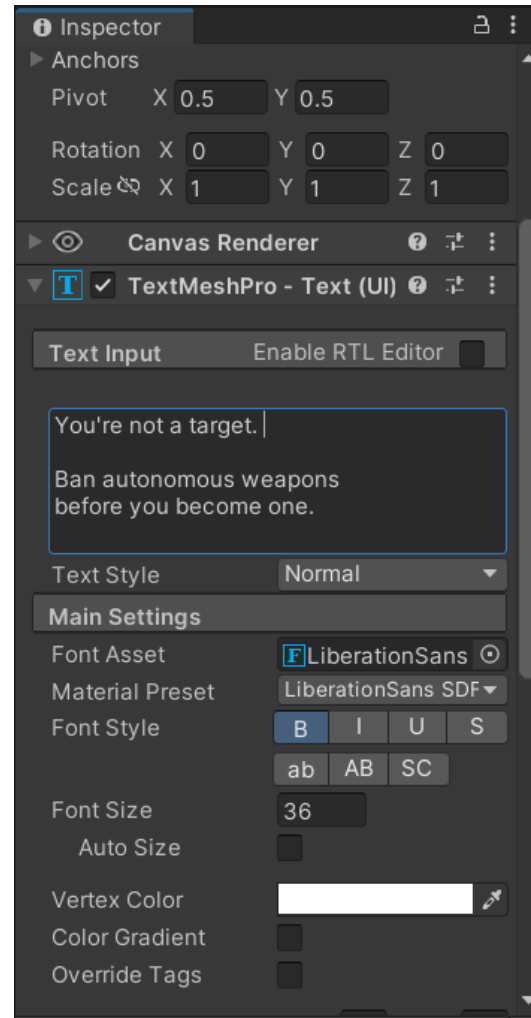
2. under 'Image', set the colour to black. This will cause the Image to turn black.



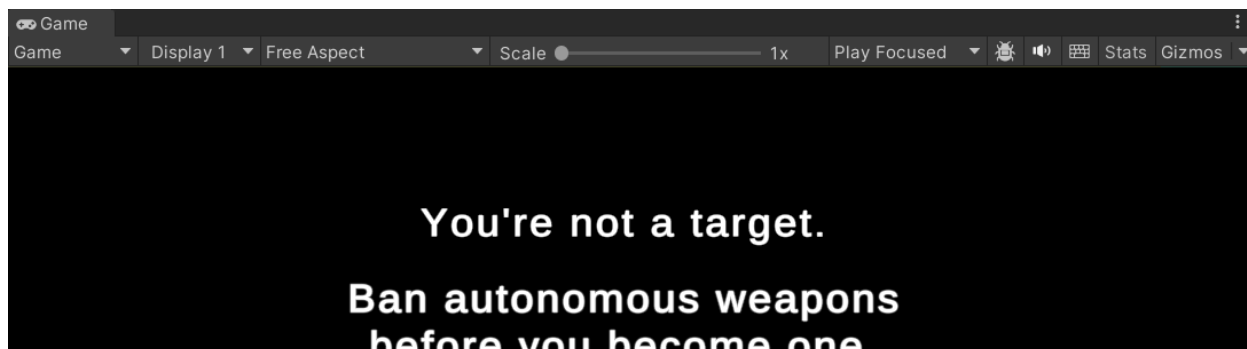
Now, to add the text elements to the UI. Under the newly adjusted End Message Screen Image, right click, then click 'UI' > 'Text'. This will insert a text box onto the UI.



To ensure the best readability, enter the settings to the right, as well as the text for the End Screen. The Material Preset is Drop Shadow. However, the font size might vary, as you will see shortly.

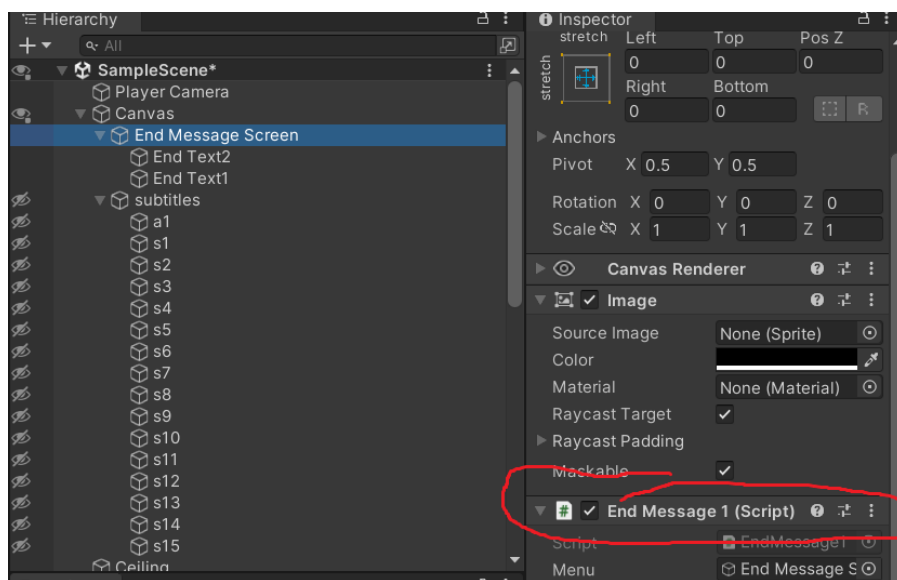


Pictured above is the Scene View of the UI. The text element can be adjusted using the Move, Rotate and Scale Tool.

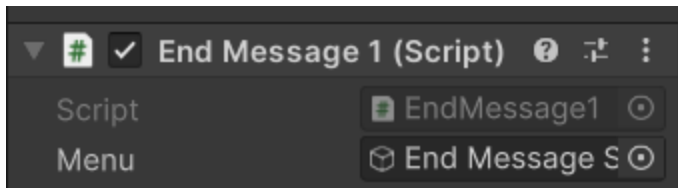


Pictured above is the Game View. You will need to adjust the text elements in the Scene View so that they appear at an appropriate size and legibility in the Game View.

At this point, running the game will simply cause the black screen and text to show up, blocking the Camera's view of the apartment. Now that the UI has been implemented into the game, the [script](#) from earlier must be implemented. On the End Message Screen, click 'Add Component' in the Inspector Window, then choose 'New Script'. Give it an appropriate title. Once it is visible in the Inspector Window, click the 3 dots shown in the red circle and choose 'Edit Script'. This will open Visual Basic or whichever coding software your device defaults to.

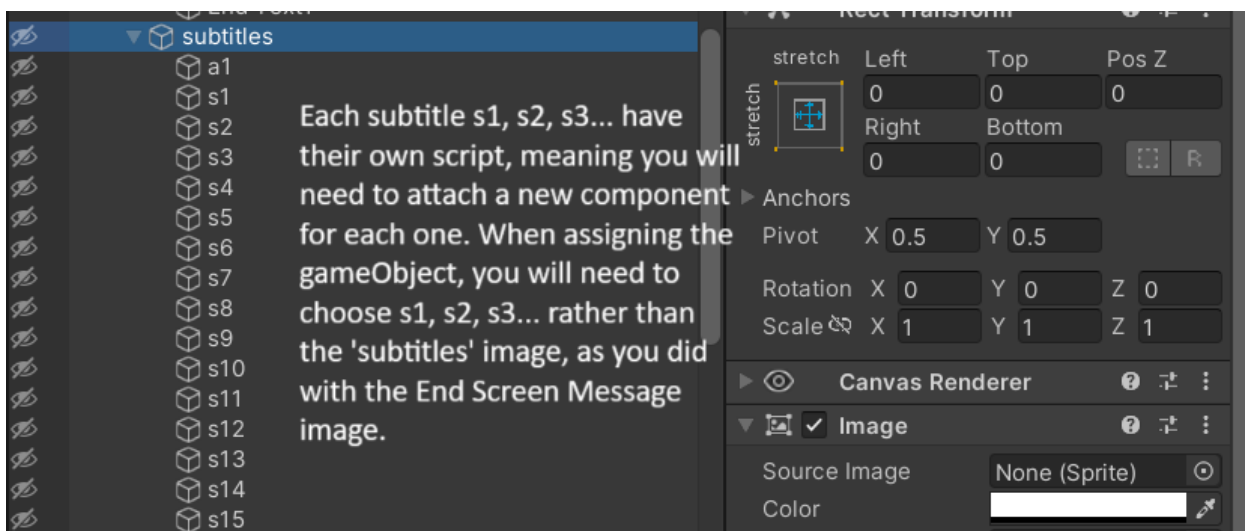


Copy and paste the script in, choosing the appropriate time variables for `timeLeft1` and `timeLeft2`, then save. Go back to the Inspector Window, click on the circle and dot for 'Menu', then assign the script to 'End Message Screen' This will link our code to the 'End Message Screen', which is the gameObject mentioned in the script. When the game runs, it will wait until `timeLeft1` passes, cause the End Message Screen to appear (includes the black image and the text element), then disappears after `timeLeft2` passes.



### 5.4.3 - Attaching the Subtitles

The subtitles follow the same logic, with some slight differences. Create another image under 'Canvas', and name it 'Subtitles'. The only difference is that you will require multiple text elements, and you will need to attach the scripts to each individual text element, rather than the image as a whole.



## 6 - Testing Stage

The testing stage is crucial for checking the feasibility of each of your components and gaining feedback to be sure your result is on the right path and fulfills the user/clients' requirements. Prototypes will be built to test the achievement of previously decided-upon objectives that our team has set out. The prototypes have small targetable objectives with specific tests and measurable results. Below you will find our prototyping test plan and some detailed examples of some of the prototypes our design underwent.

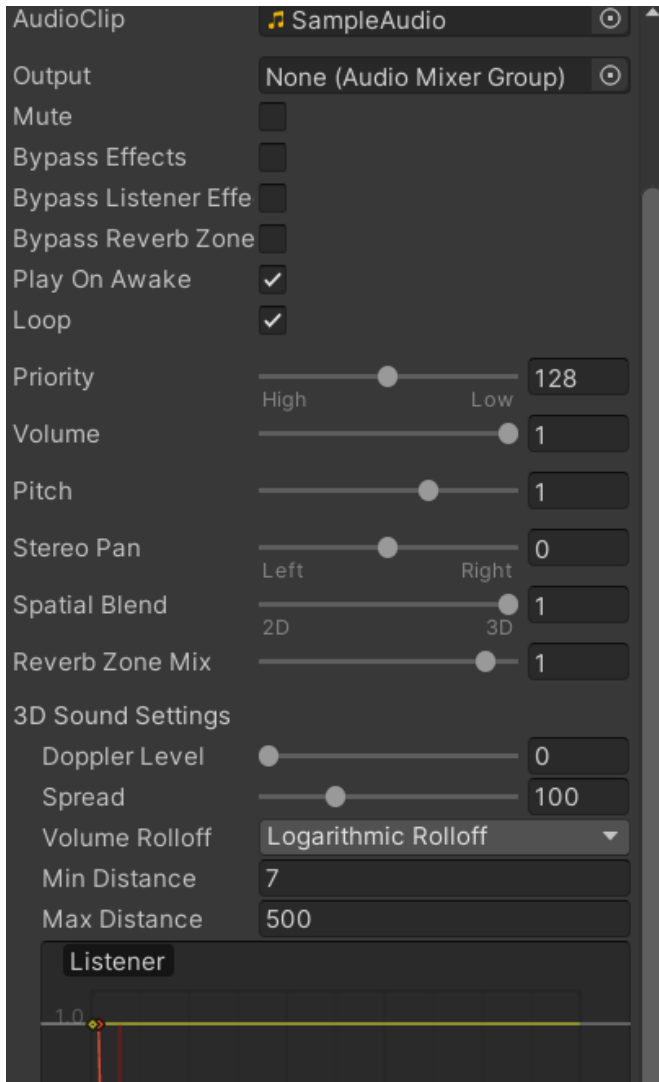
### 6.1 - Prototyping Test Plan

<u>Test #</u>	<u>Test Objective (Why)</u>	<u>Prototype Test Method (What)</u>	<u>Prototype Description (How)</u>	<u>Test Date (When)</u>	<u>Stopping criteria</u>	<u>Progress</u>	<u>Prototype employed</u>
1	Is this product within a specified time limit	Physical, high fidelity prototyping	The length of time it takes to go through our environment will be recorded with a stopwatch	Prototyping test will take 2 hours	Video is within 30-45 seconds		
2	Does the product's movement system function	Physical, low fidelity prototyping	Camera movements and character movement within a generic environment	Prototyping test will take 30	User is able to look up, down, left, right and move forward, back, side to side  Movement is not laggy	Works well (please see below for movement in a box prototype)	<a href="https://drive.google.com/file/d/1UjM8_uIln27GrLPuucJeW6kott3Klxc0/view?usp=sharing">https://drive.google.com/file/d/1UjM8_uIln27GrLPuucJeW6kott3Klxc0/view?usp=sharing</a>

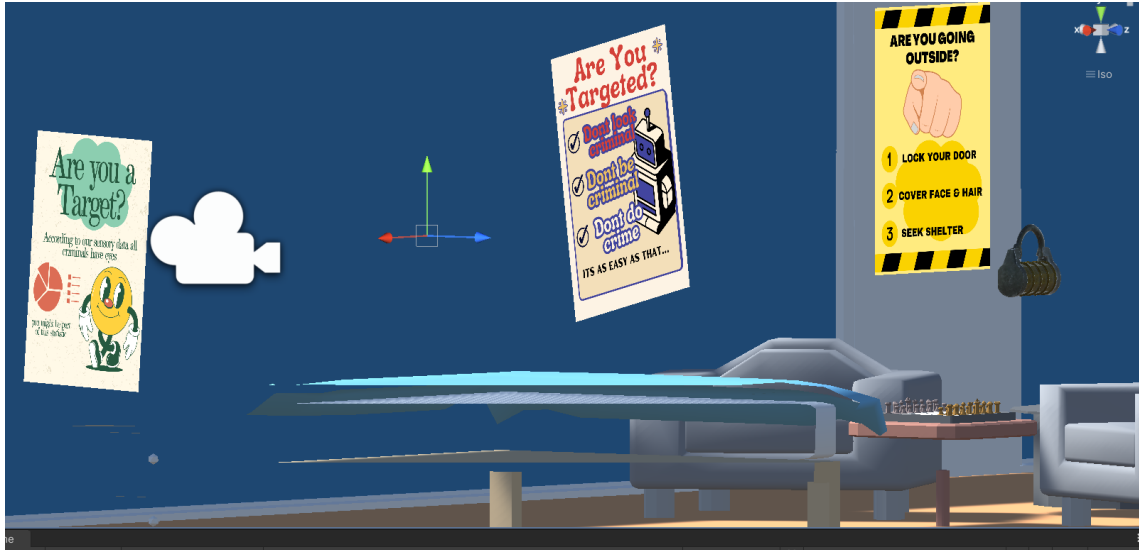


4	User interactivity with environment	Physical low fidelity prototyping	User can see environment clearly  Assets are legible and not blurry	Prototyping test will take an hour	All writing (on posters) is legible  Furniture and assets are recognizable  Image quality is not blurry	Works well (posters yet to be added)	<a href="https://drive.google.com/file/d/1m3QdgWOLgEOWU-D0RKgpU_U2jmyL9DJS/view?usp=sharing">https://drive.google.com/file/d/1m3QdgWOLgEOWU-D0RKgpU_U2jmyL9DJS/view?usp=sharing</a>
5	User accessibility <ul style="list-style-type: none"> <li>subtitles, audio</li> </ul>	physical, high-fidelity	user can clearly hear game and see audio	Prototype test could take	Audio is at least clear to the viewer, subtitles optional.	use C# scripts, to add large and clear subtitles  Check for if 3D spatial audio works correctly and is clear enough to understand  Enlarge Posters and make them high-definition to enhance readability	<a href="https://drive.google.com/file/d/1kZyZhEvHEGvr3at3ISIMHOYoxzKjUASC/view?usp=drive_link">https://drive.google.com/file/d/1kZyZhEvHEGvr3at3ISIMHOYoxzKjUASC/view?usp=drive_link</a>  *please see below for employed audio settings*

\*Prototype of Audio settings\*



\*Test objective #4\*



## Conclusions

Overall this project put the design process into action, beginning with raw data from the client, Mines Action Canada, and turning it into a comprehensive VR prototype that fulfills the discussed design criteria. One benefit of the project that was produced this term was the simplicity of its design, it helped each member of the group complete their work on time so that

there was a feasible product to show the client on Design Day. One thing that would have been done differently

If the project had a longer deadline, simplicity would not have been a high priority. This would have led to a final prototype that was more immersive and complex with features and assets in the simulation. For instance, more time would have been spent making the assets and setting of the simulation interactive and immersive (ie. opening doors, arrows to point users where to go, options users could choose between, etc.) to engage users more in the story. As well, more time would have been allocated towards user testing to receive feedback on the emotionality and comprehensibility of the simulation. Preferably the testing would be done in person to see live user reactions to different parts of the simulation as they have the VR headset on.

Overall the simplicity of the design was a key asset to the functionality of the final prototype on design day, but with more time the final product would have had more complex features and settings for users to navigate.

## References

*Immoral Code*. (2021). *Immoral code - a film by Stop Killer Robots*. Retrieved December 1, 2023, from <https://immoralcode.io/index.html>.

Mines Action Canada. (2023). *Mines Action Canada*. <https://www.minesactioncanada.org/>