BCIT Technical Test

Task	Status
Hyperlink creation (Itch.io)	✓
3D model spawn dynamically	✓
Middle mouse button hold functionality	✓
Right mouse button rotate functionality	✓
Left mouse button functionality	✓
A UI list – lists all the parts (Machine gun model)	✓
Reset button (Resets the application)	✓
X-Ray button (Gives an x-ray view of the model)	✓
Transparent button (Gives a “clear” view of the model)	✓
GitHub utilization	✓
Commented code (For better preview and future edits)	✓

Workings:

1. **3D model:** The model provided with the project comprises of over 130 substructures. However, none of the parts had any colliders on them. To combat this, I have added mesh colliders (mesh colliders are more dynamic to shapes compared to box colliders but loose out to efficiency) on them for future use. A script called “SpawnObject” is employed to spawn the model (as a prefab) in the center of the screen.

2. **3D interactions:** Major 3D interactions are handled by two scripts – “GameManager” and “SelectObject”. Interactions such as rotation of the model, selecting an individual part of the model, moving individual parts and many more are handled by the said scripts in a dynamic fashion.

3. **UI elements:** Various UI elements such as buttons and lists are displayed on the screen in a dynamic fashion. The resolution of the screen will not alter the style/position of the UI elements in any way. Therefore, the elements are as dynamic as possible. Also, a floating text box to show the element’s name.

4. **X-Ray & Transparent modes:** The two buttons at the top left corner are responsible for changing the mode of the model. The default mode is in its regular color, while x-ray mode will activate a more wireframe mesh mode imitating an x-ray showing the inner workings of the model. The transparent mode will make the model transparent owing to the near zero value of the alpha channel.

Environment:

1. The camera background color has been changed to black for a more application effect.
2. The model is converted into a prefab for future use and is stored inside the prefabs folder.
3. A red material is created for the usage of highlighting certain game object.
4. A wireframe material is created to use during x-ray mode. (Custom VR shader)
5. A transparent material is used for the glass like effect. (Shader with alpha turned down)
6. A UI canvas to display all the UI elements such as buttons, text boxes and dropdown menu.

Directions of use:

WebGL:

Open the link and press the play button. Upon play, the model will be spawned in the middle of the screen.

Left mouse button: Hover over an object to highlight it (red color) or press and hold it on the object to change its position.

Middle mouse button: Press and hold the middle mouse button at any point on the screen to change the model's position to that point.

Right mouse button: Press and hold the right mouse button to change the orientation (rotation) of the model along the x and y axis.

X-Ray button: Press the x-ray button to change the model from being a gun to a blueprint of the gun (due to a custom shader, the materials for this effect could not be compiled for WebGL).

Transparent button: Press the button to change the transparency of the model.

Reset button: Press the button to reset everything to its original place.

TreeUI: A list displaying all the components of the model (machine gun).

Desktop:

Download the project either using the GitHub link or the google drive link and open the project in Unity 2020.3.12f1.

Left mouse button: Hover over an object to highlight it (red color) or press and hold it on the object to change its position.

Middle mouse button: Press and hold the middle mouse button at any point on the screen to change the model's position to that point.

Right mouse button: Press and hold the right mouse button to change the orientation (rotation) of the model along the x and y axis. The amount of rotation can be altered using a slider in the inspector menu.

X-Ray button: Press the x-ray button to change the model from being a gun to a blueprint of the gun (the materials for this effect do work in Unity).

Transparent button: Press the button to change the transparency of the model.

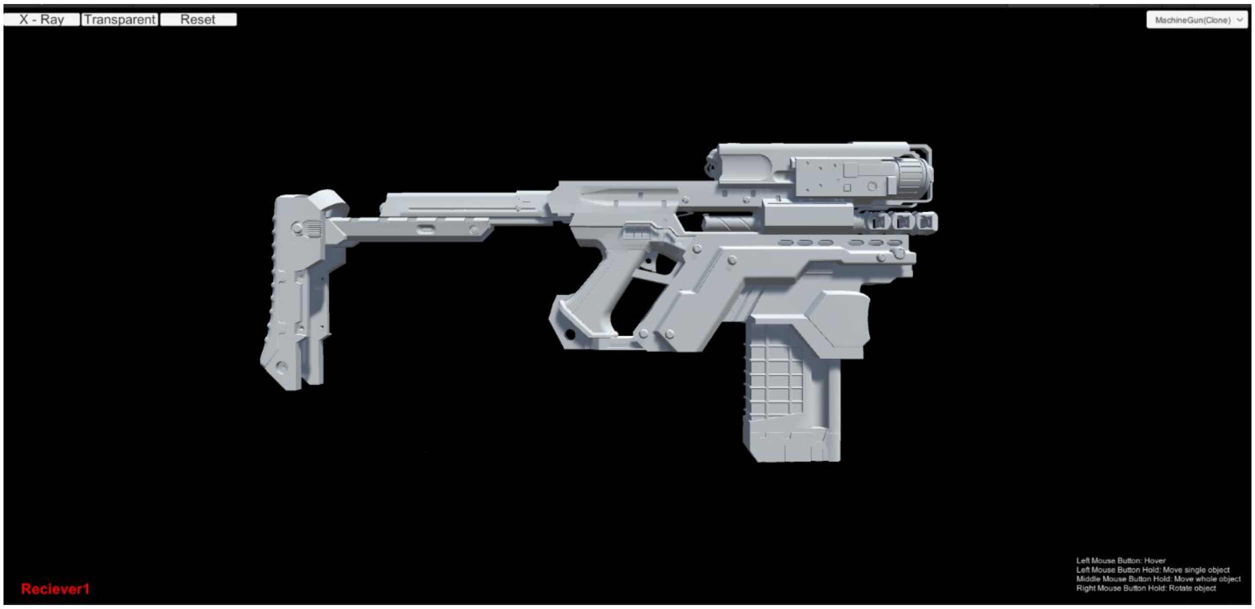
Reset button: Press the button to reset everything to its original place.

TreeUI: A list displaying all the components of the model (machine gun). When an object is selected, the name of the selected object is displayed in the console window.

Technical Challenges and innovations:

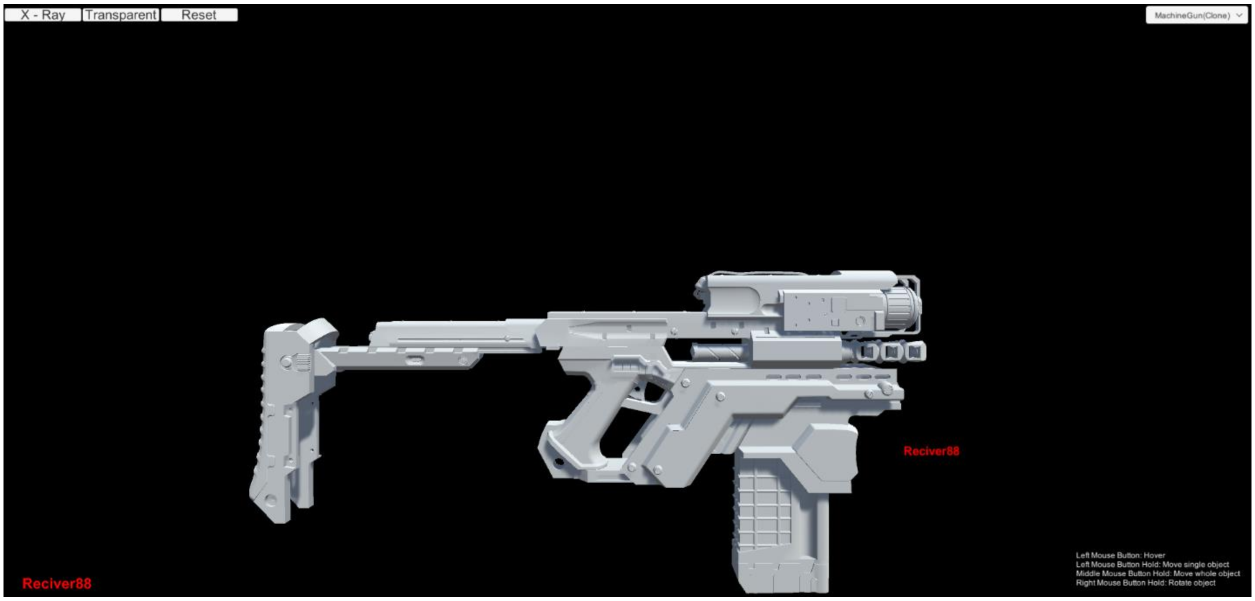
1. The UI elements in the scene will remain in their respective positions irrespective of the screen resolution. *(Dynamic scaling)*
2. If more than one object is to be spawned, simply drag and drop the object into the list of the spawner game object and the code will work. *(Dynamic spawning)*
3. The 3D interactions will work irrespective of the model. *(Dynamic interactions)*
4. The materials used for the project will always be compiled to avoid any errors. (Project settings -> Graphics -> add the shaders)
5. The project consistently hits over *300 fps* when played inside of Unity with no visible lag. (Tested on Core i7, 16GB Ram and 1660Ti graphics card)
6. The code has been written in a way that follows the camelCase as well as the SOLID principles.
7. Little to no use of the Update method in the program to avoid bottleneck.
8. Due to minor error in the conversion between Vector3 and Transform values, the real-world mouse coordinates do not get properly converted to screen values, causing trouble when moving a selected object. (The object has an offset)

Screen Captures:



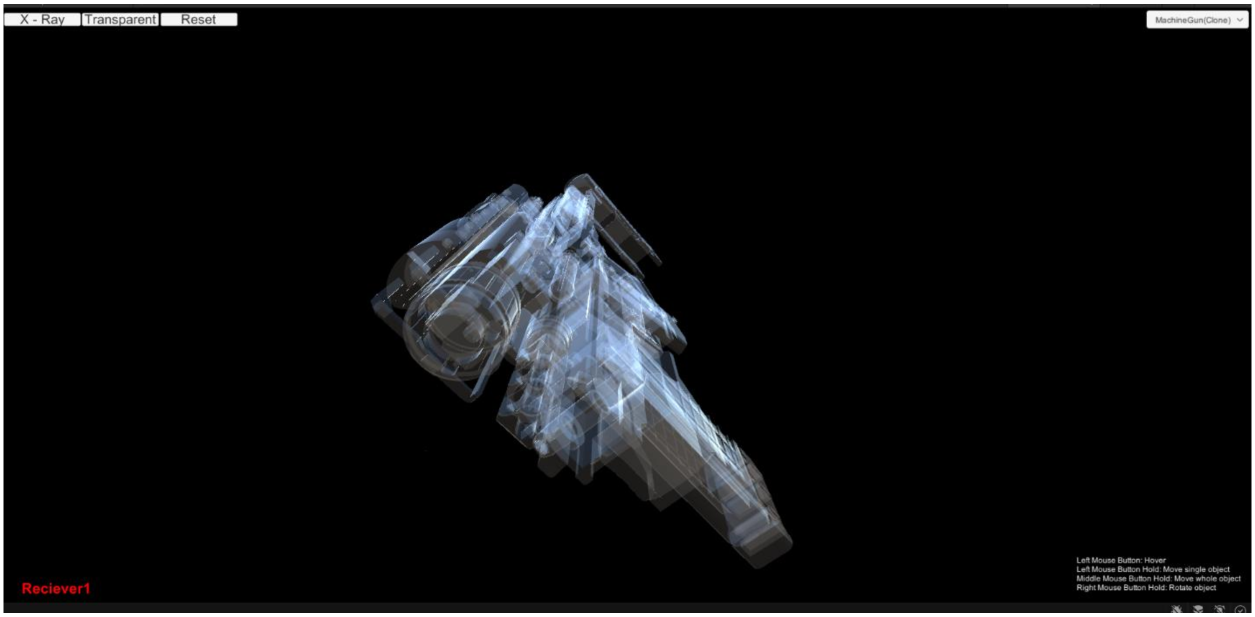
Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 1 The UI



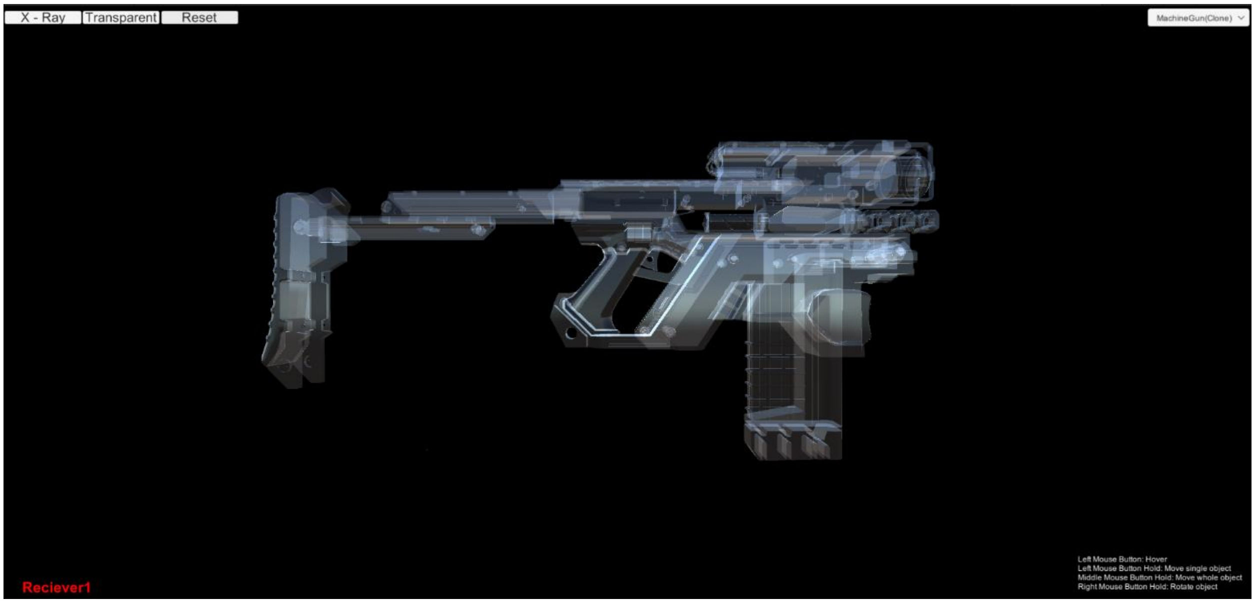
Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 2 Drag the game object



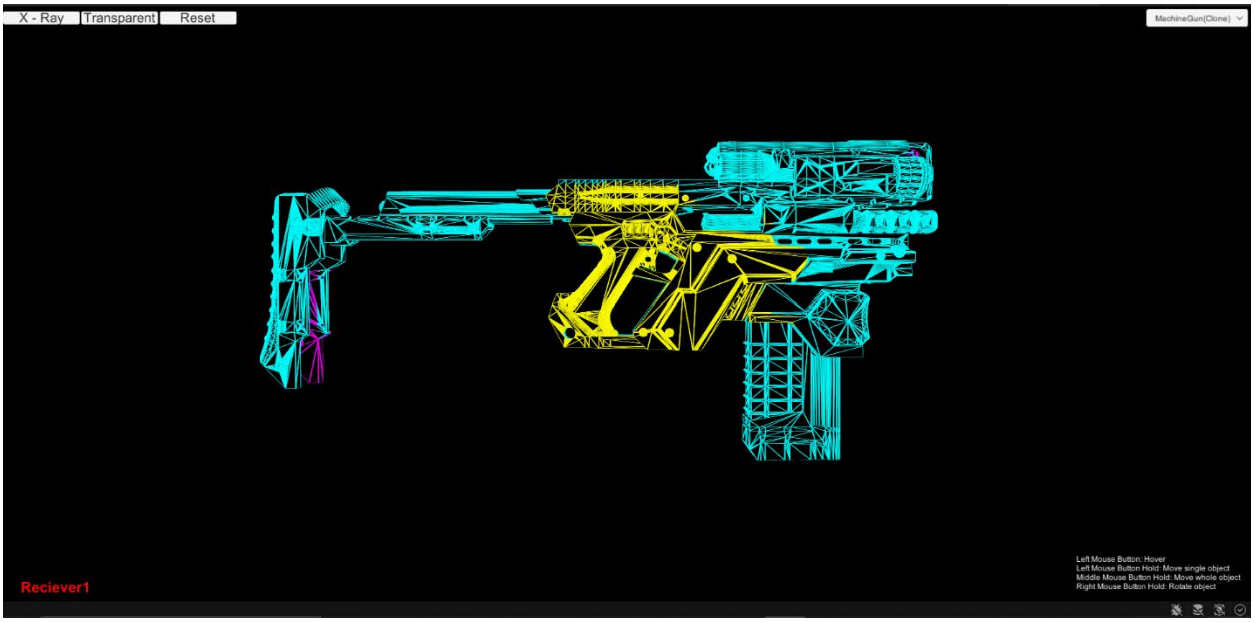
Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 3 Rotate the game object



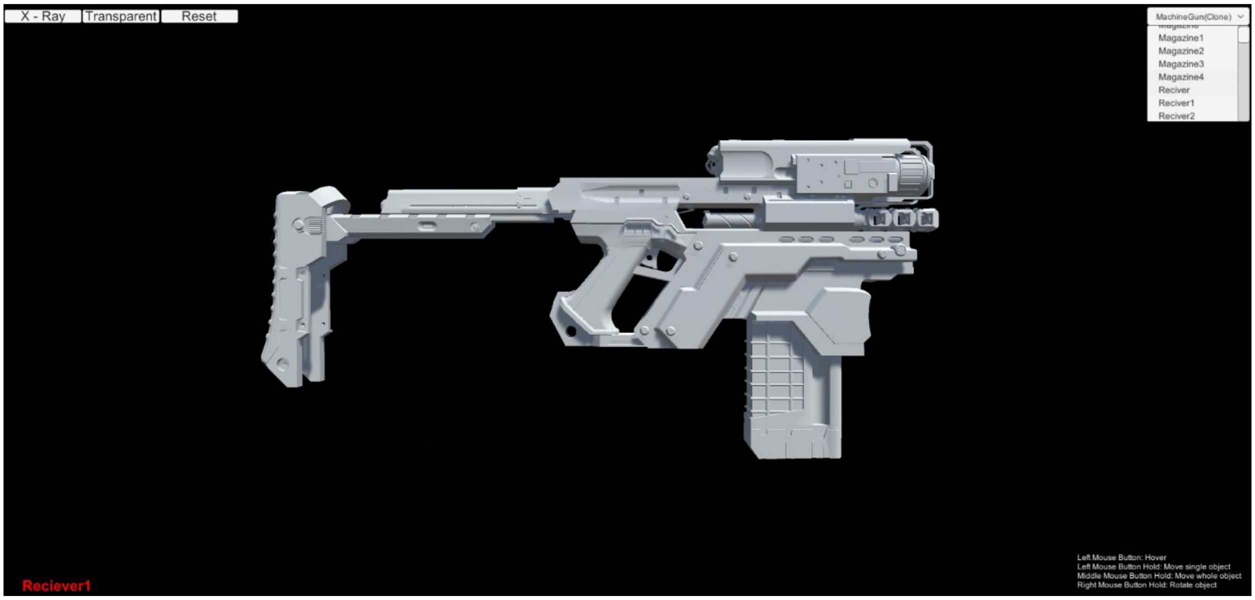
Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 4 Transparent mode



Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 5 X-Ray mode



Left Mouse Button: Hover
Left Mouse Button Hold: Move single object
Middle Mouse Button Hold: Move whole object
Right Mouse Button Hold: Rotate object

Figure 6 UI Tree

Shortcomings:

1. In WebGL window, the X-Ray does not work as the shader is a custom shader and is not compatible with WebGL. (Solution tried: added the shader in the project settings to always be compiled, no errors while building the application)

2. The mouse drag for a ray-cast object does not work perfectly. The object when dragged has an offset on the X and Y axis compared to the mouse position. (Solution tried: The code works for the whole prefab but falters for individual objects, there is a slight error in the conversion from Vector3 to Transform)