

**Submitted By:**Ashna Wasif (2791)  
**Class**: BSCS 7B (Morning)

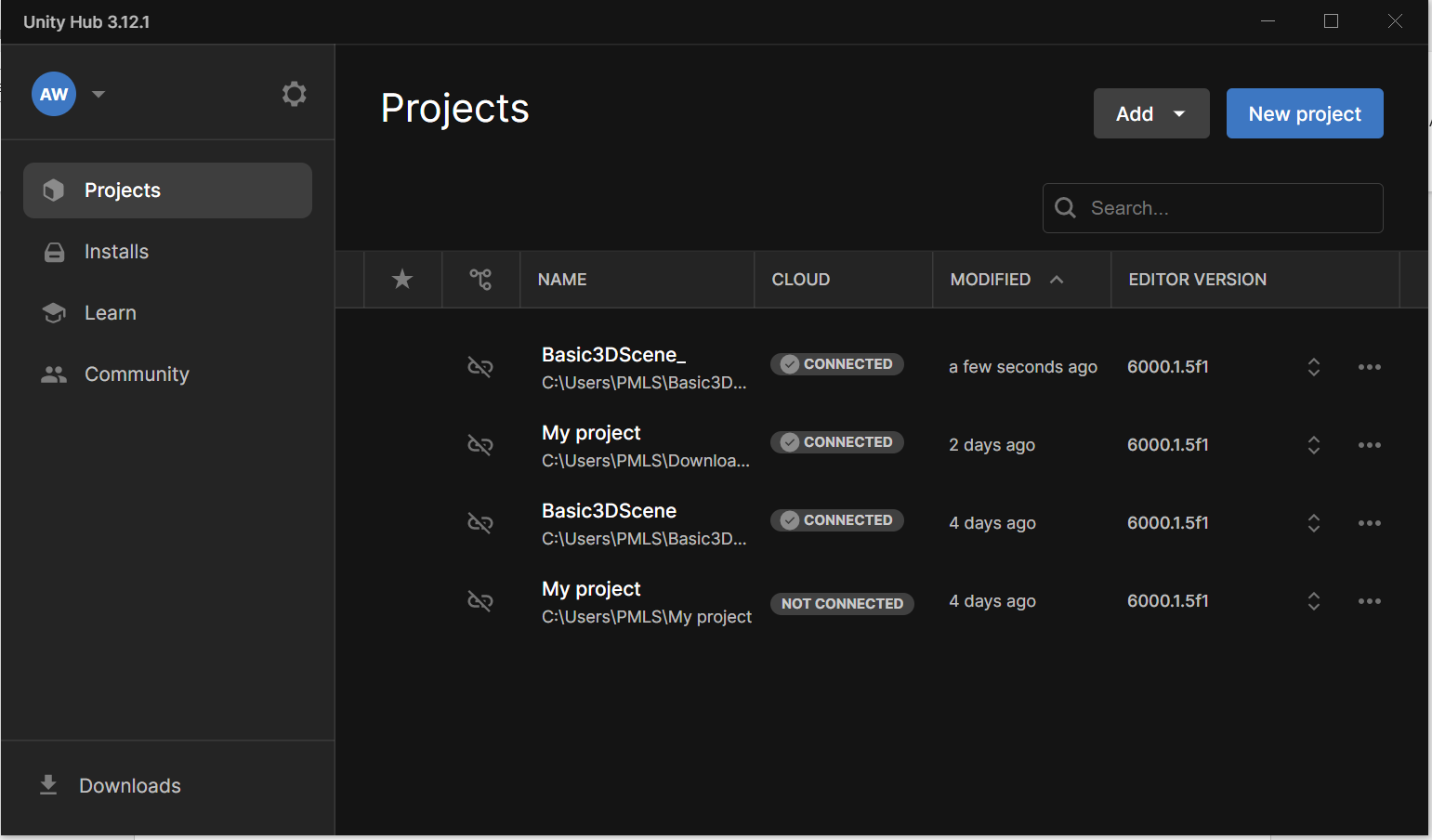
**Subject**: Computer Graphics

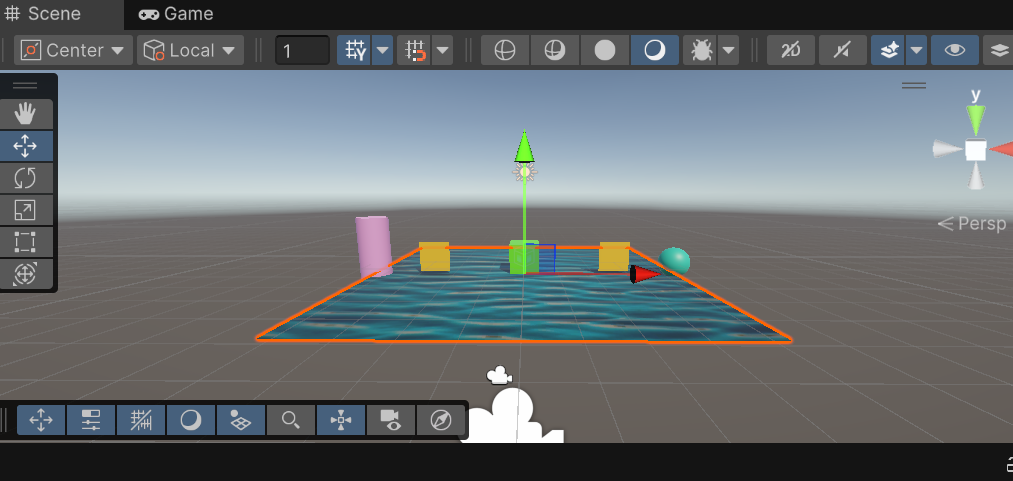
**Submitted To:** Mam Sabina Irum

**Date:** 3rd June, 2025

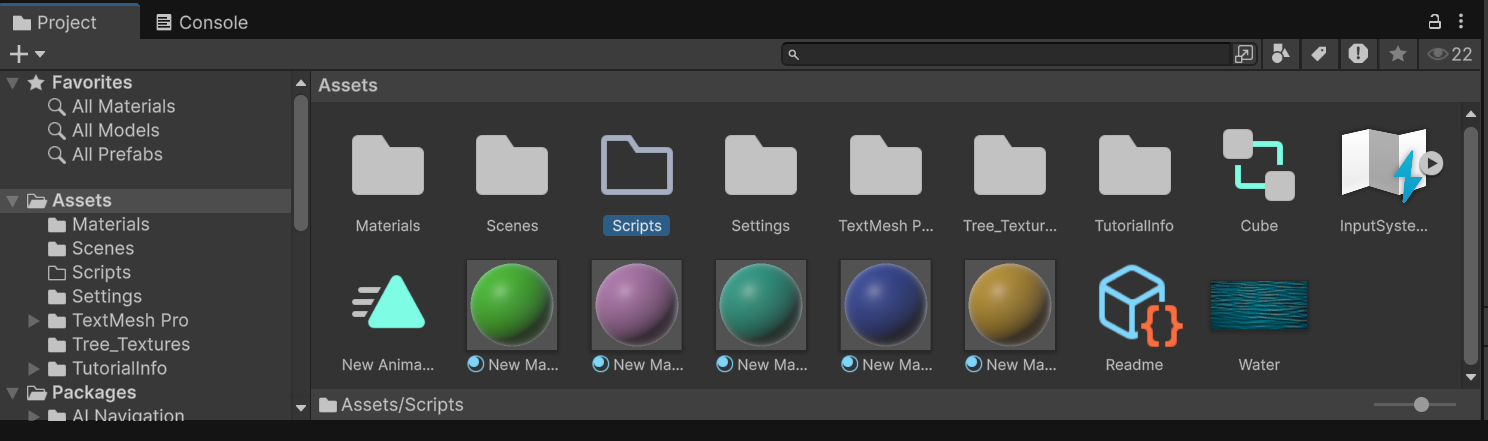
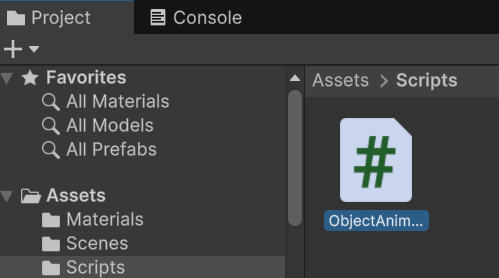
**UNITY 3D LAB TASK 3**

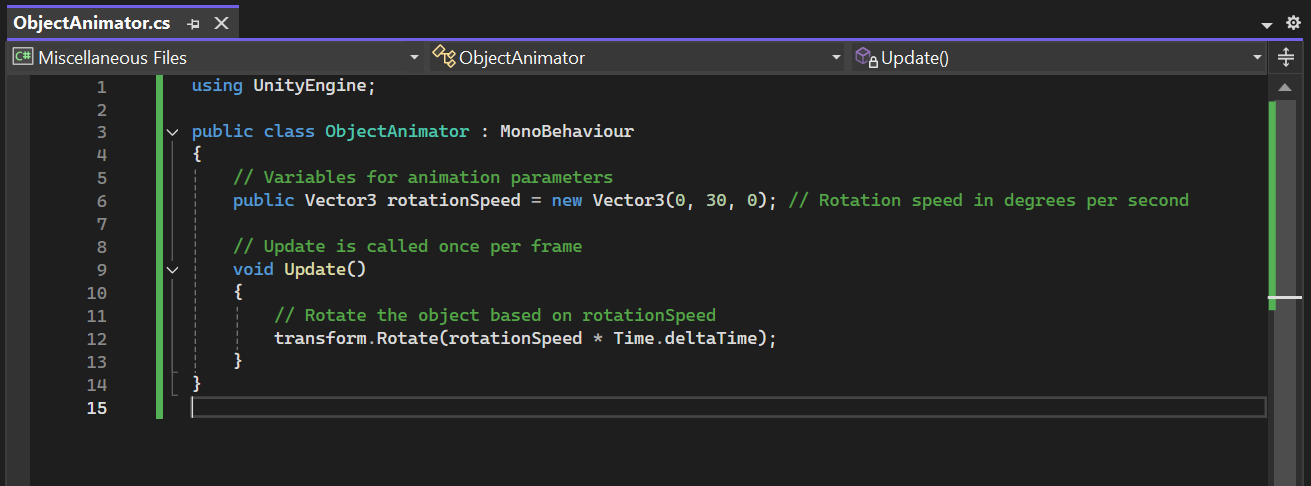
1. **Setting Up the Project:**

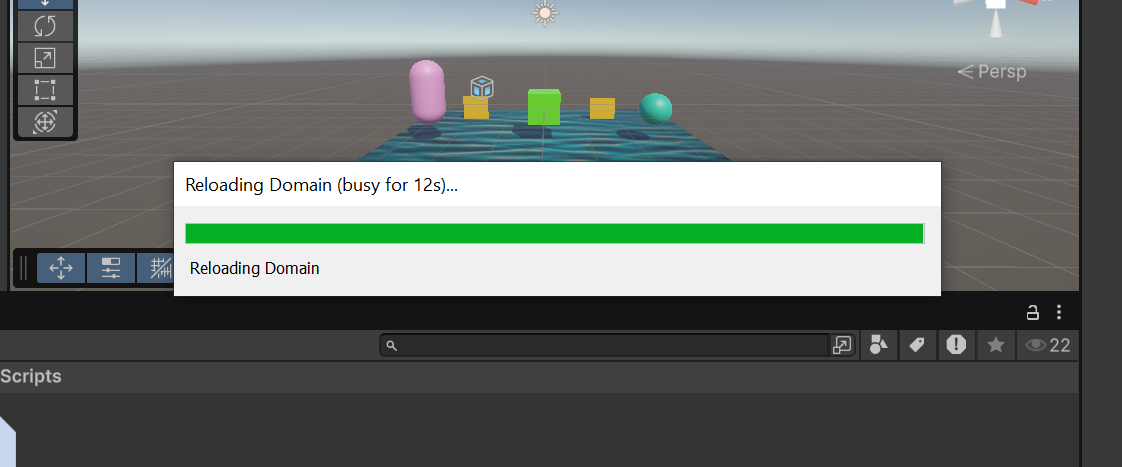
* Open Unity Hub and load the "Basic3DScene" project.

It will open our old project which is:

* Create a new C# script named "ObjectAnimator" and add it to the project's Scripts folder.



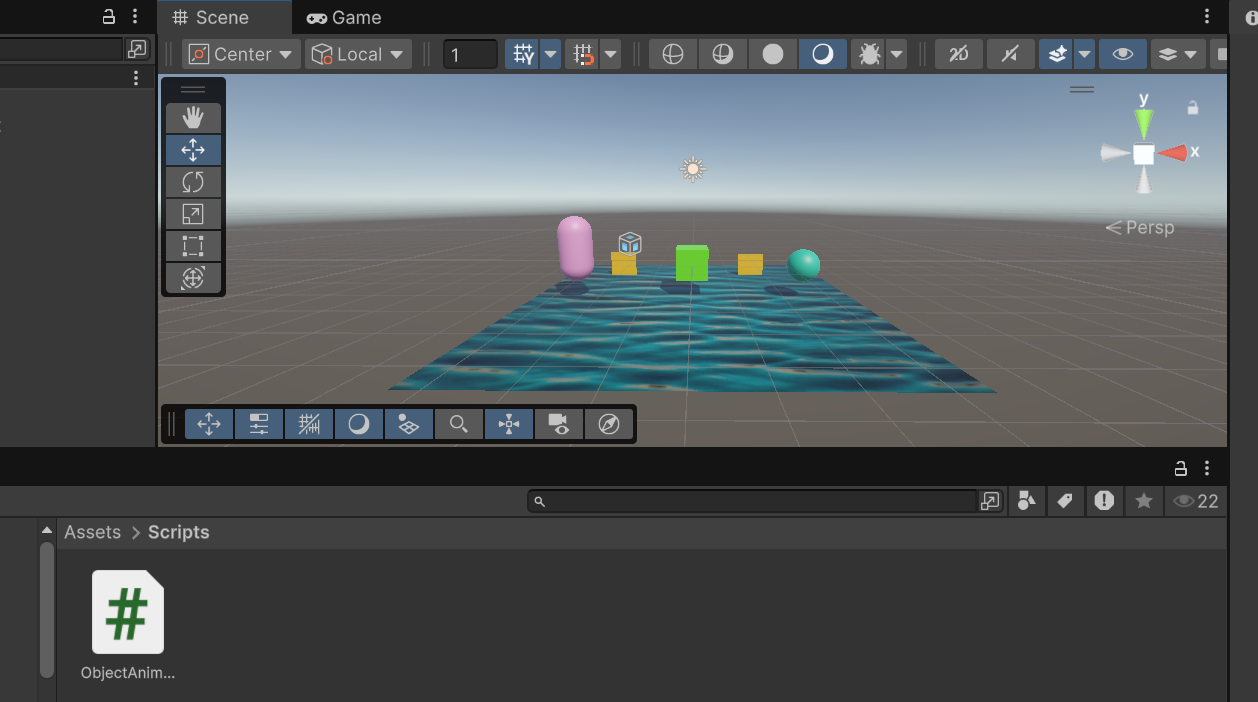
Double click on it, this will redirect you to visual Studio. Write code there.

Then come back to Unity. It will show:

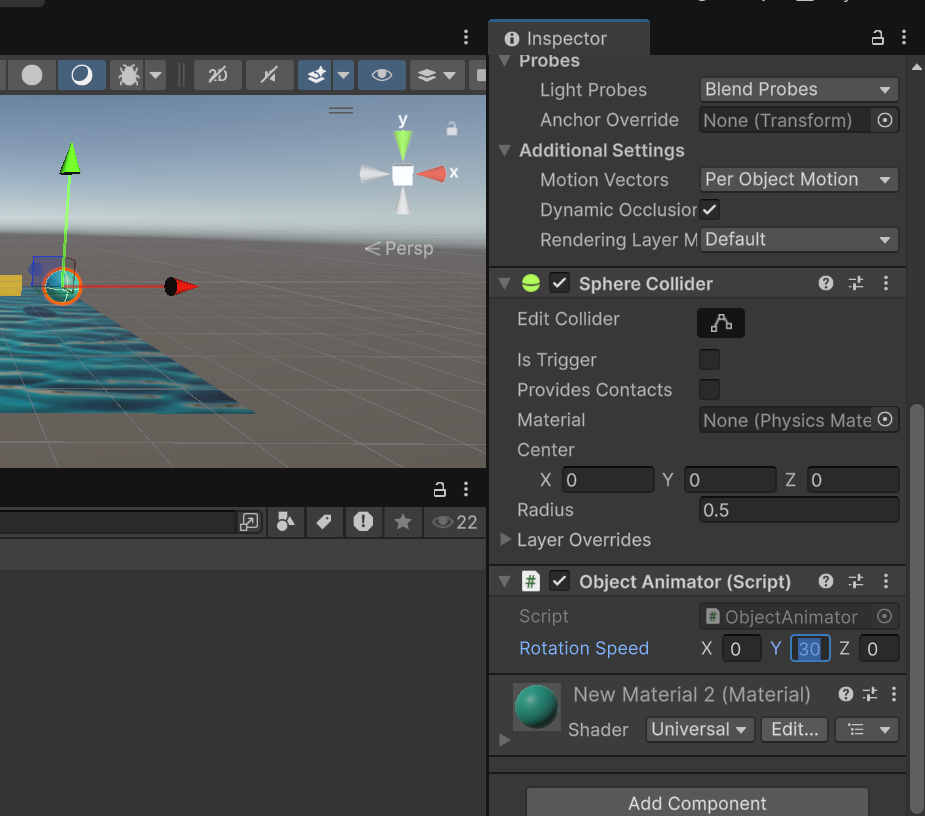
It means unity will detect all changes on its own. Now dragging the script onto a any game Object in the Hierarchy. I’ve dropped it into Sphere. On running it, We can clearly see the cube slowly rotate around the Y-axis.

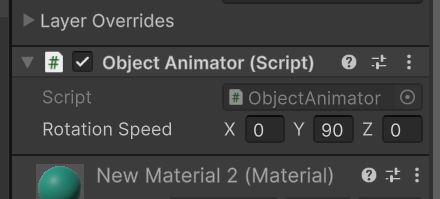
1. **Adding More Objects:**

* Attach the ObjectAnimator script to additional 3D objects like cylinders, cones, or capsules in the scene.

I’ve dropped the ObjectAnimation script to all other 3D shapes as well:

* Adjust the rotationSpeed variable in the script component of each object to control their rotation speed.

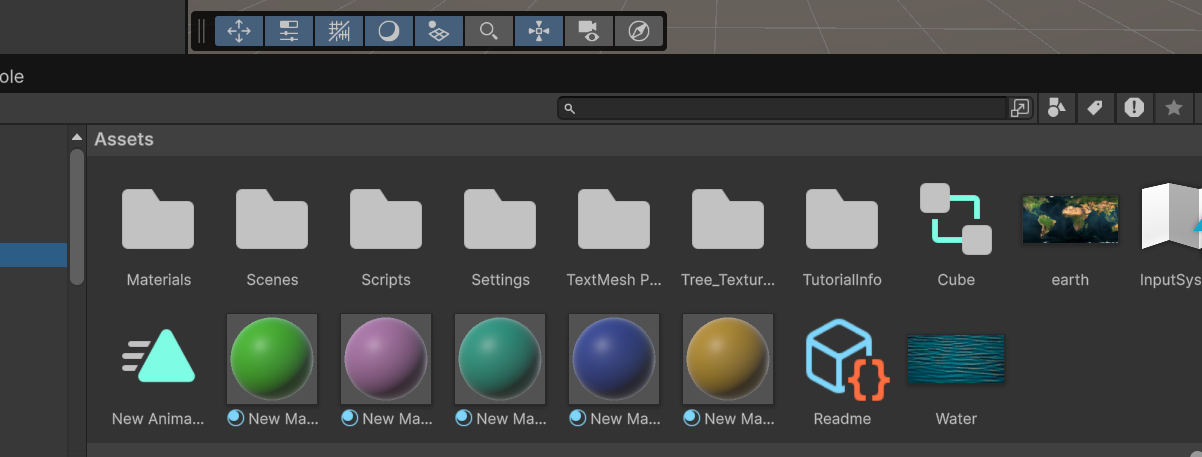
Select the Object in the Hierarchy that has ObjectAnimator script attached (e.g., a Sphere). In the Inspector panel, find the ObjectAnimator (Script) component. You will see a field named “Rotation Speed” with X, Y, Z values.

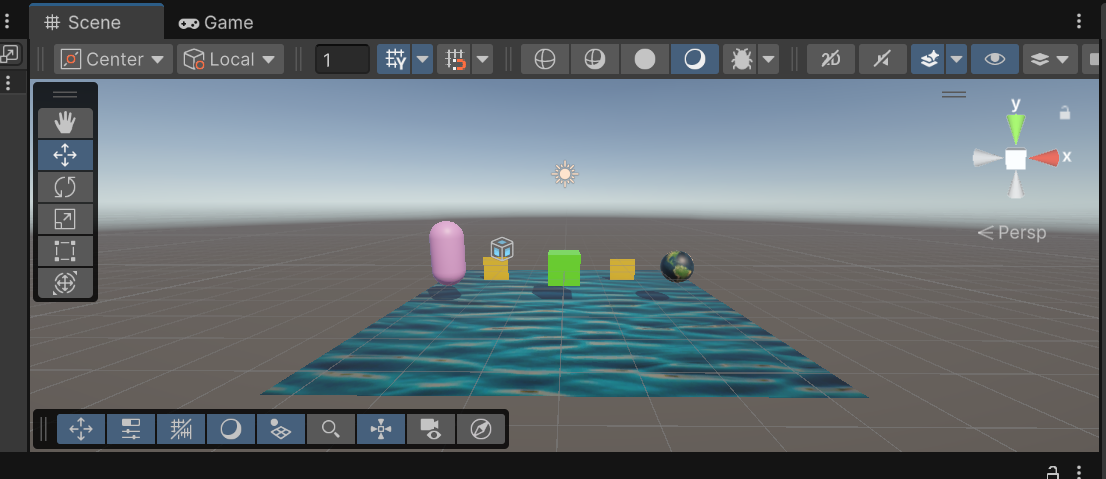
On changing speed:

1. **Applying Textures and Materials:**

* Ensure that the objects with the ObjectAnimator script have appropriate materials applied.
* Experiment with different materials and textures to enhance the visual effects of the rotating objects.

Added another texture:

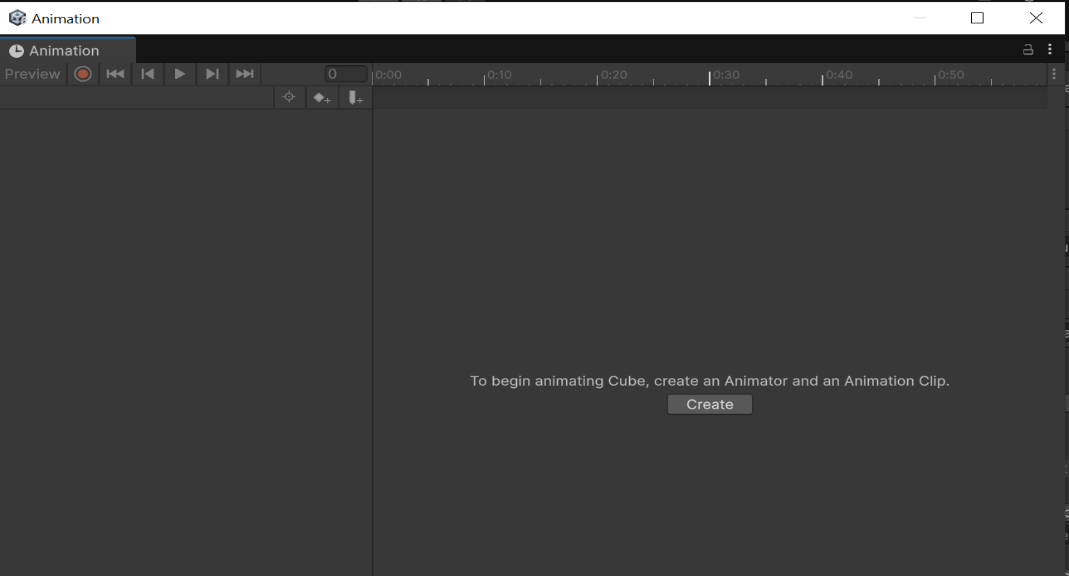
Applying it on Sphere:

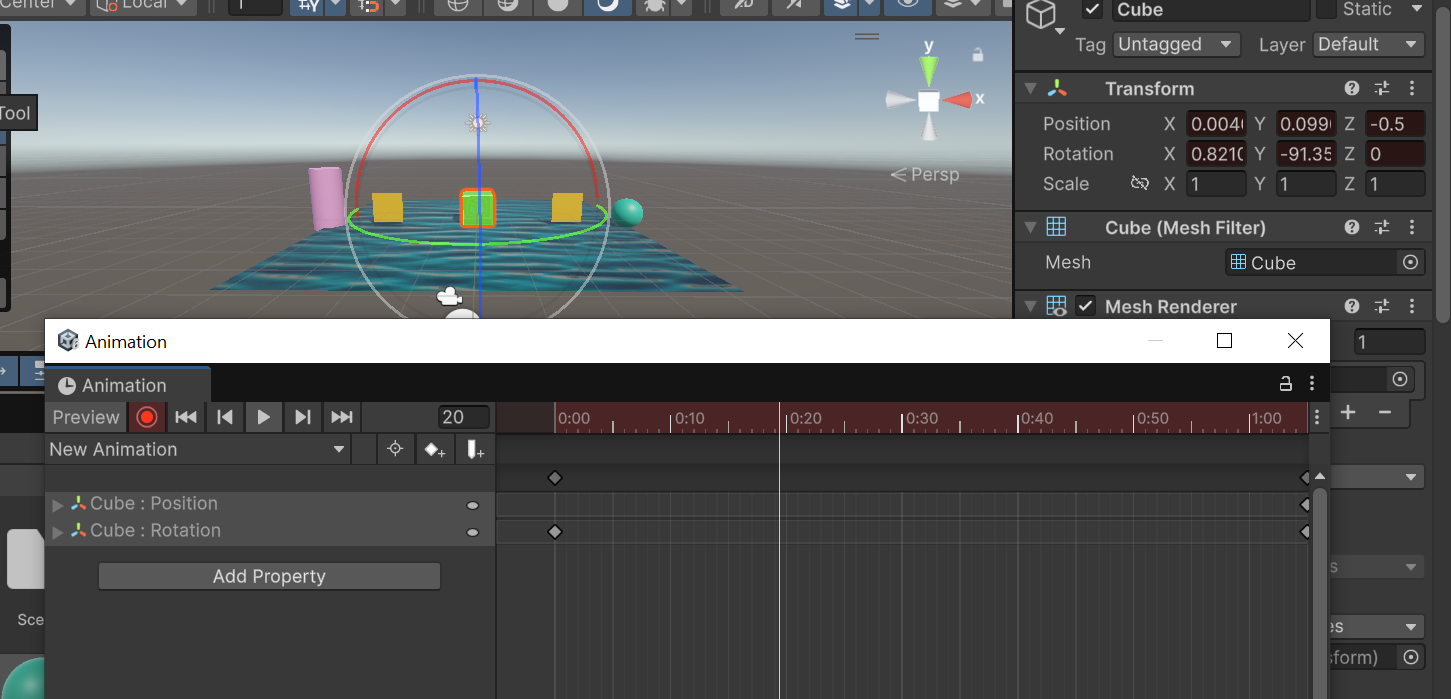


On running, we can now clearly see the rotation.

1. **Implementing Animation:**

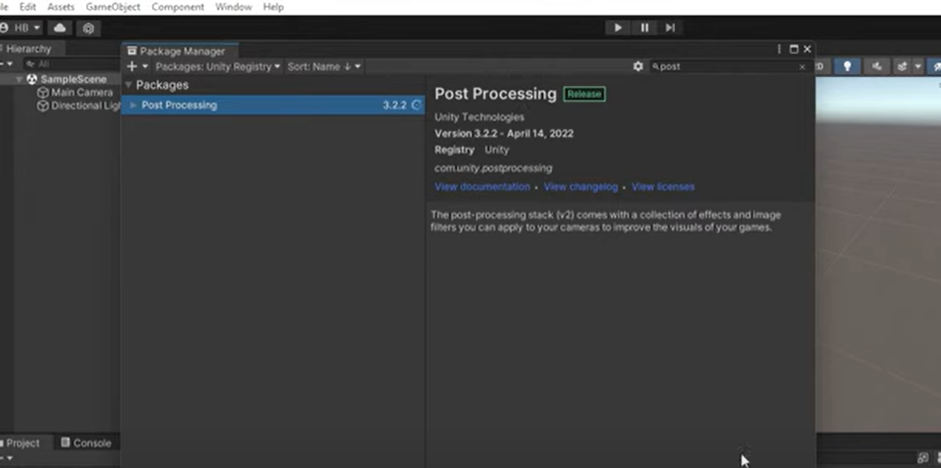
* The ObjectAnimator script automatically animates the rotation of objects.
* Use Unity's Animation window to create additional animations for other object properties such as position or scale.

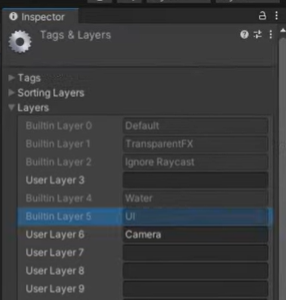
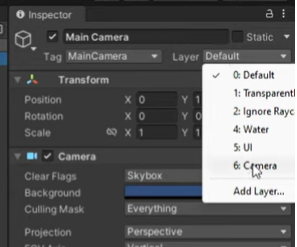
For this, we’ll be opening Animation window:

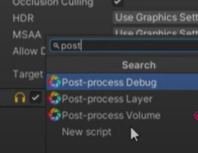
Start recording and animate object:

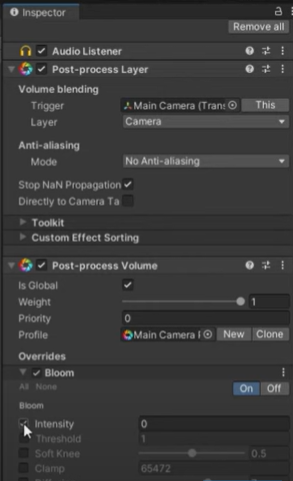
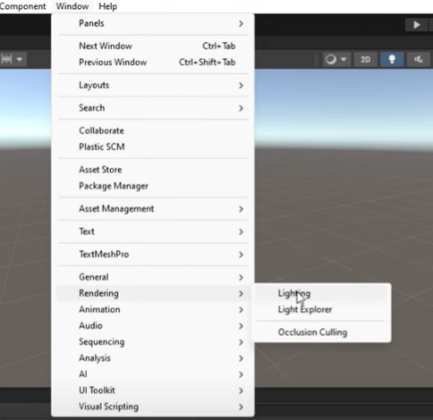
1. **Adding Particle Systems:**

* Objects with the ObjectAnimator script can emit particles using Unity's built-in particle systems.
* Add a ParticleSystem component to the objects and adjust parameters such as emission rate and color.

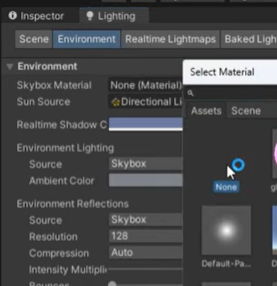
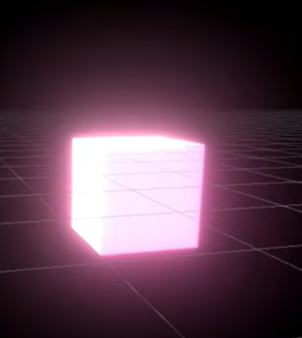
For this, click windows >> Package management >> Pacakge Manager. You’ll see this window. Search Post processing:

Install this package. Once installed, Add layer in main camera:

Then, adding audio and setting intensity:



This will add lightning in our cube. It will make our cube glow:



1. **Implementing Audio:**

Already implemented in Step 5.

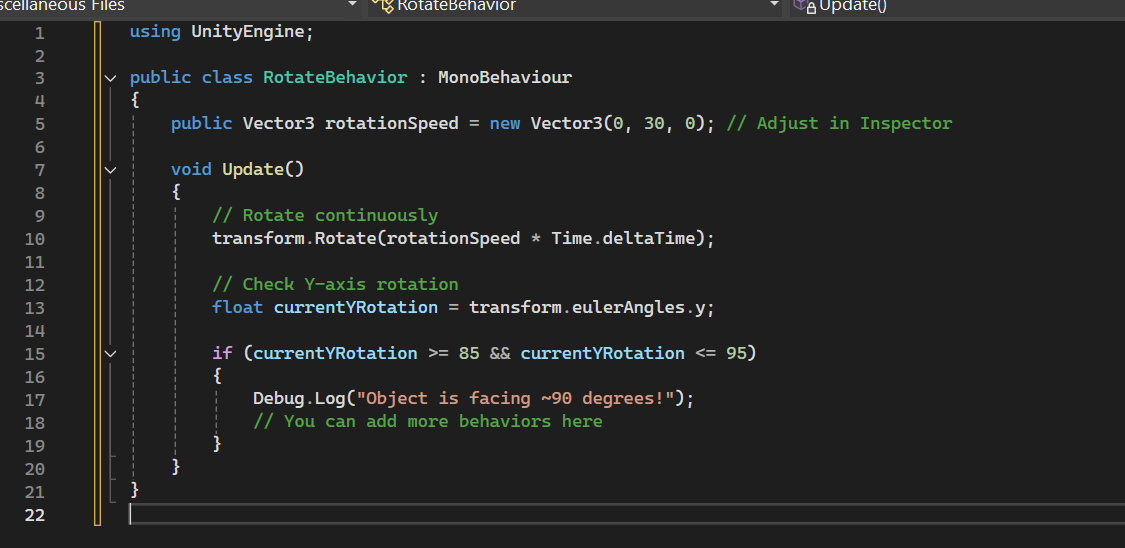
1. **Creating Interactive Elements:**

* Objects with the ObjectAnimator script can respond to user interactions through scripting.

Yes, on testing we have observed that it worked perfectly fine.

* Write additional scripts to handle user inputs and control the behavior of interactive elements based on the object's rotation.

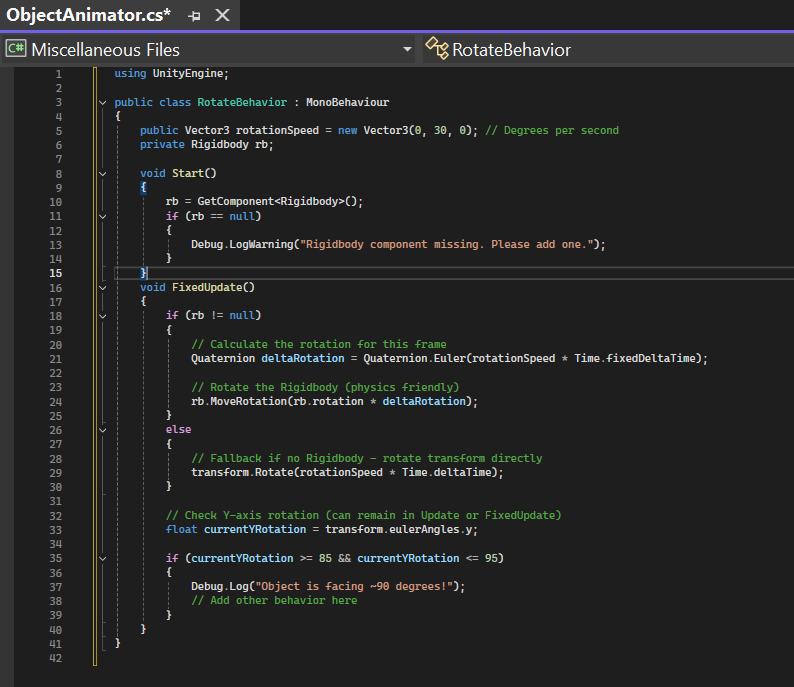
I’ve added script that Rotates the object smoothly. When its Y rotation is around 90°, it does something (like logs or triggers a function).



1. **Enhancing Physics Simulation:**

* The ObjectAnimator script does not directly interact with Unity's physics system.
* However, you can combine it with physics-based interactions by adding Rigidbody components to the objects and adjusting their properties accordingly.

We have added a Rigidbody variable and get it in Start(). Secondly, used rb.MoveRotation() inside FixedUpdate() to rotate in a physics-friendly way. If no Rigidbody is attached, it falls back to the original direct transform rotation. Rotation logic and angle check remains basically the same.



* **Testing, Debugging, and Optimization:**
* Test the scene to ensure that objects with the ObjectAnimator script rotate as expected.
* Debug any issues related to rotation speed or unexpected behavior by inspecting the script's variables and debugging output.

Everything’s working as expected.

1. **Documentation and Reflection:**

* Document the use of the ObjectAnimator script in the project's documentation.
* Reflect on how the script enhances the scene by providing dynamic animation to objects and facilitating interactive experiences for users.

The script enhances the scene by adding dynamic rotation to objects, making them visually engaging. It creates a more immersive and interactive experience by allowing users to see animated objects in their real-world environment. This movement draws attention and adds life to the scene.