

GAME PROGRAMMING

LAB 3: Beginner Scripting – Part 1

Aim: To practice the following scripts using the required game objects:

<https://learn.unity.com/project/beginner-gameplay-scripting?uv=2019.3>

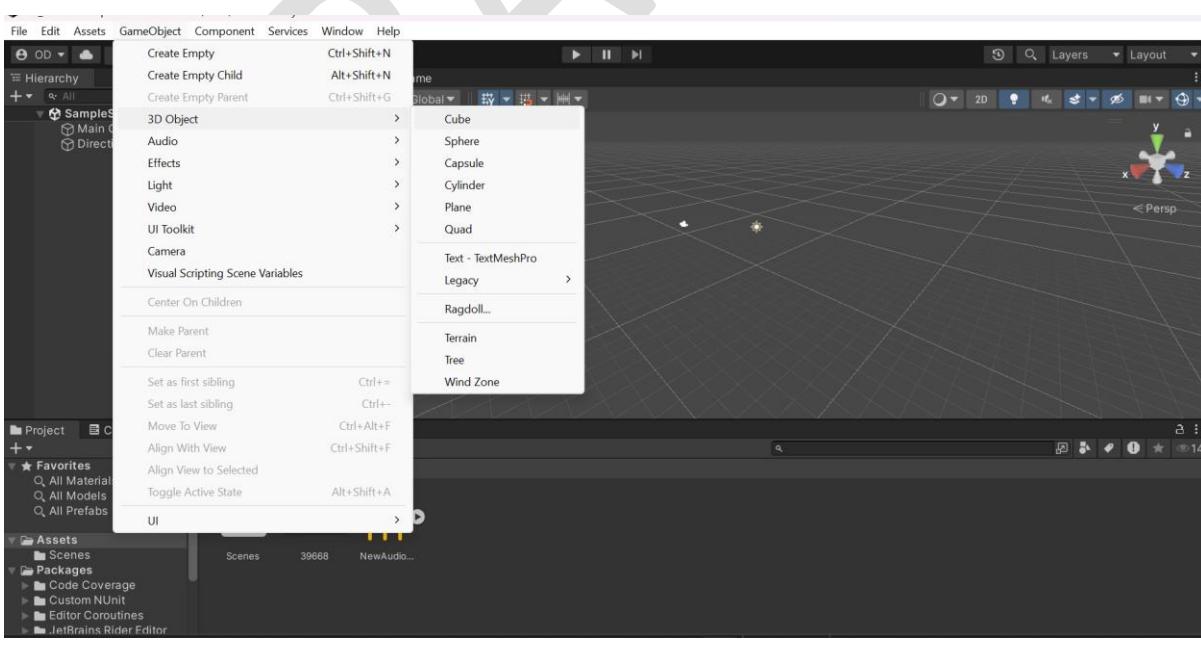
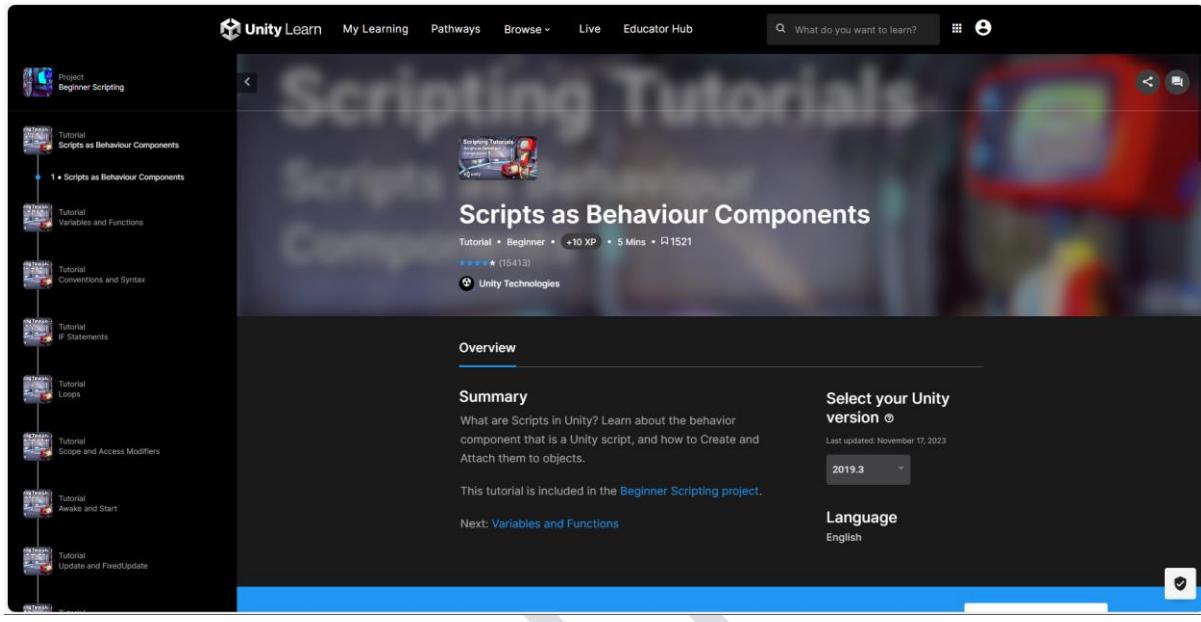
- 1. Scripts as Behaviour Components
- 2. Variables & Functions
- 3. Conventions & Syntax
- 4. If statement
- 5. Loops
- 6. Scope & Access Modifier
- 7. Awake & Start
- 8. Update & Fixed Update
- 9. Vector Maths
- 10. Enabling & Disabling Components
- 11. Activating GameObjects
- 12. Translate & Rotate
- 13. Look At

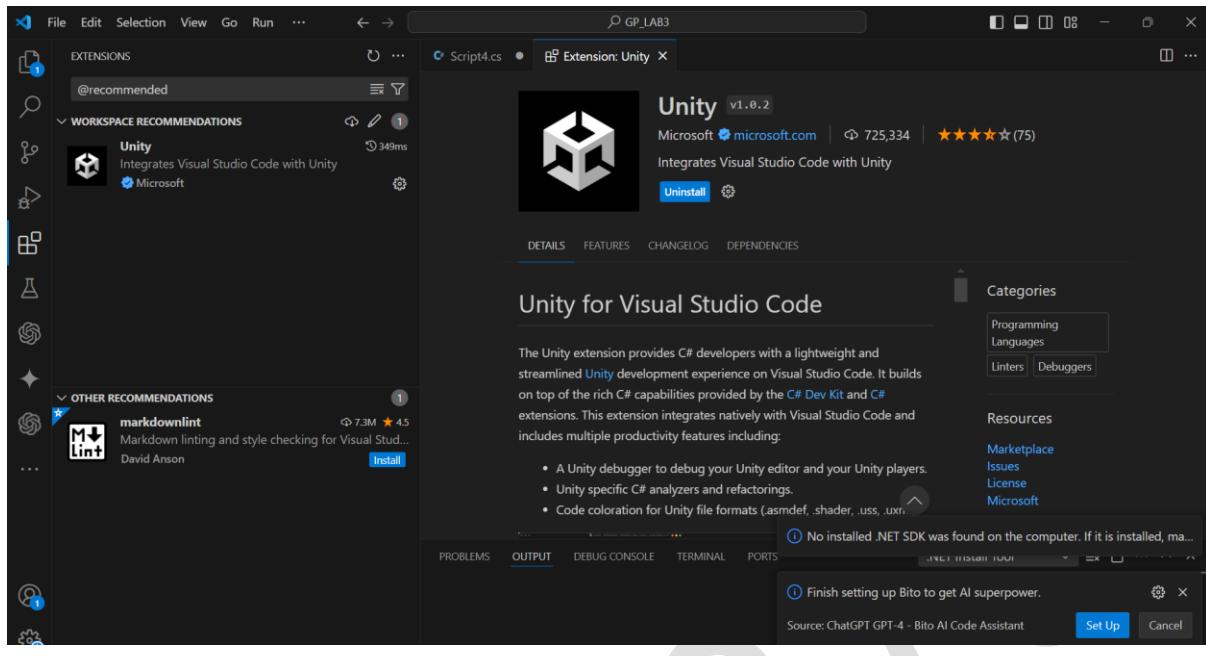
Execute the scripts given in the link for the above listed items and provide appropriate screenshots, gameplay video for giving the clarity of your development. Provide your Register number in the scene using UI objects

NAME : OM SUBRATO DEY

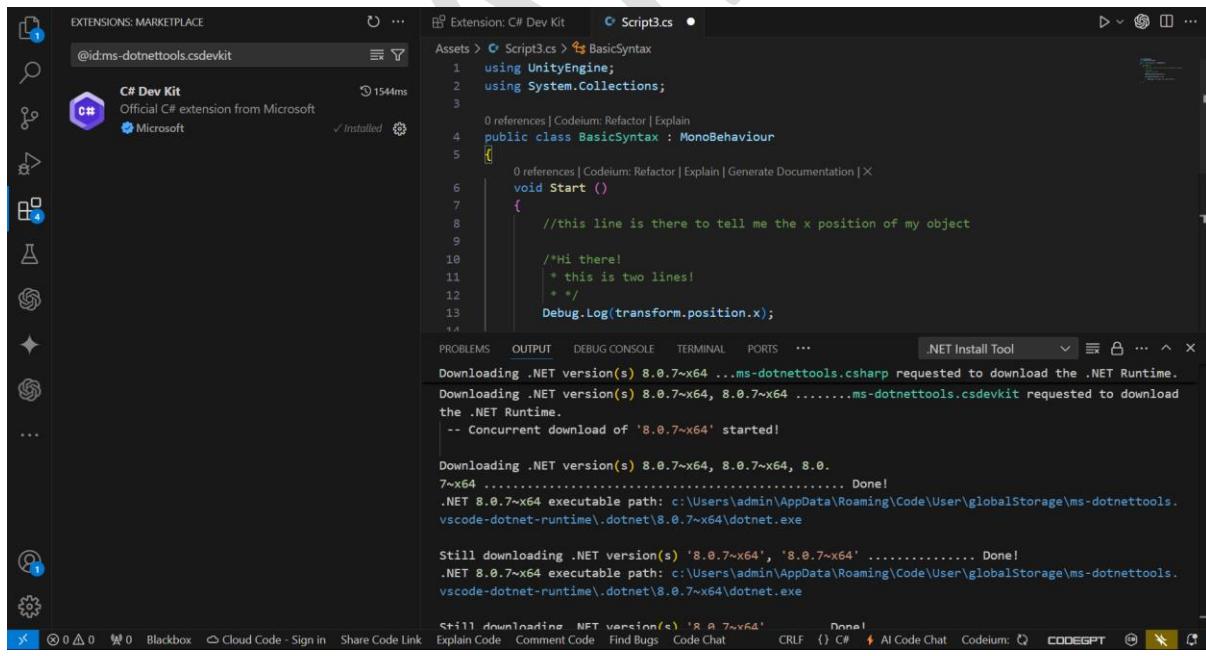
REG NO.: 21BAI1876

Step 0: Initial setup for the environment requirements.

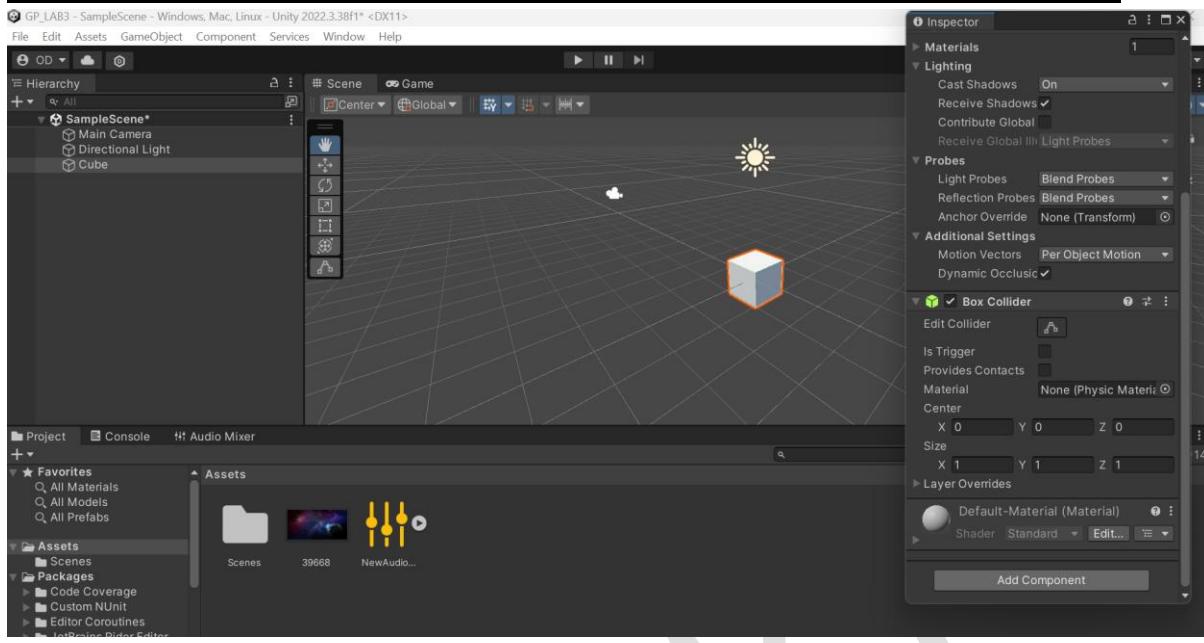


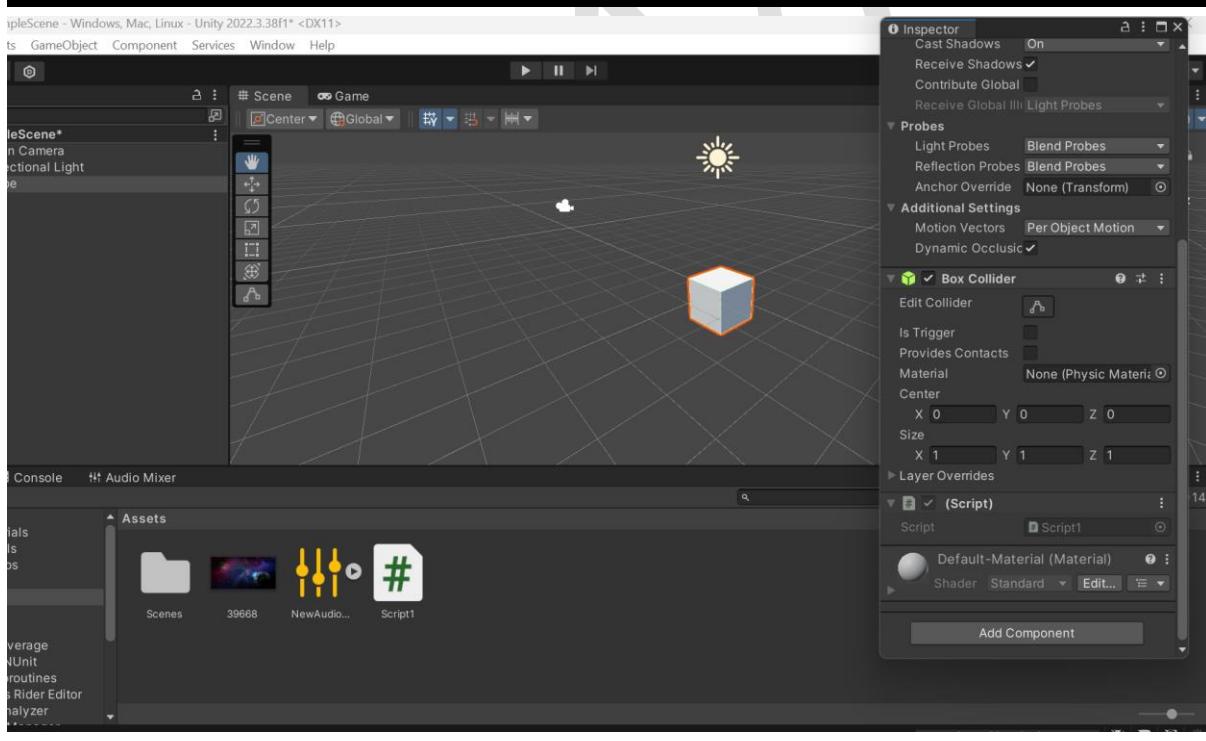
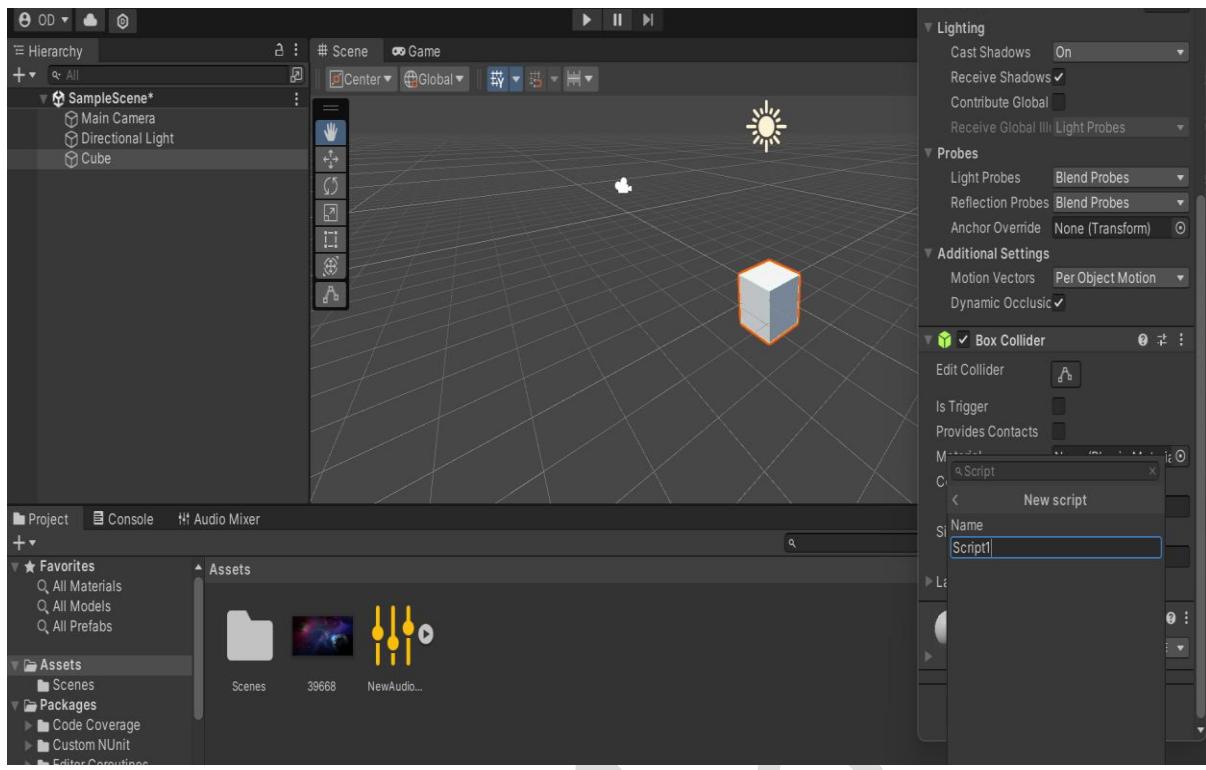


→ Installing C# Dev Kit Extension to execute codes of C sharp in VScode and Unity extensions.

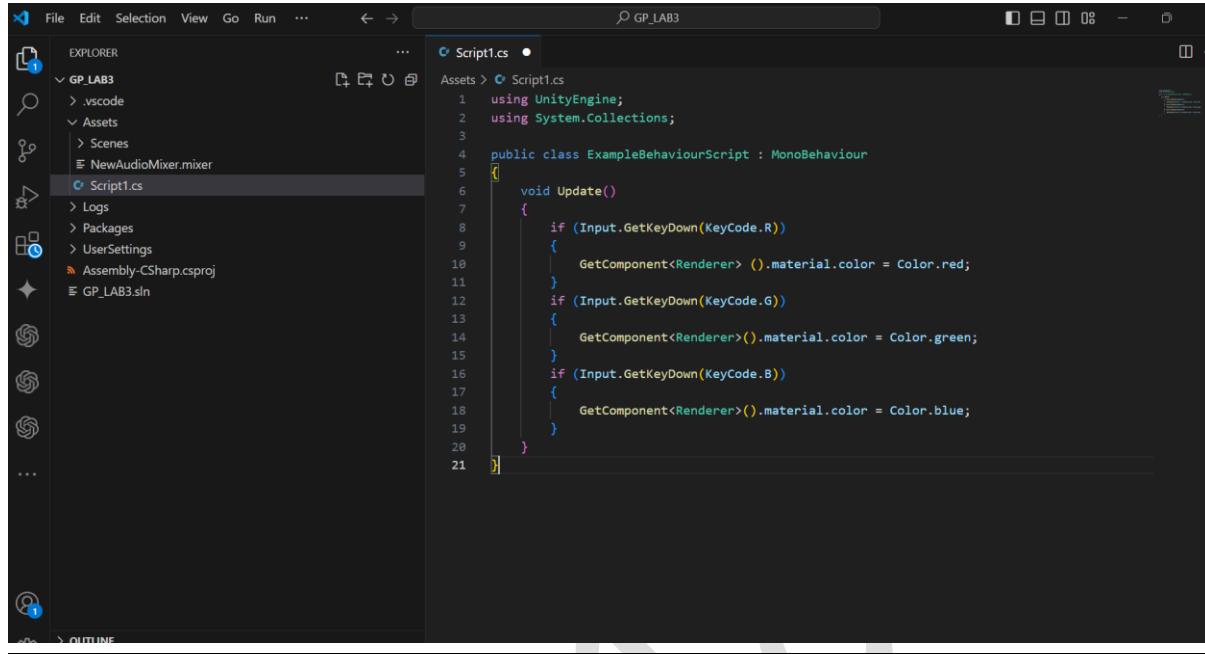


Step 1: Scripting the behaviour component

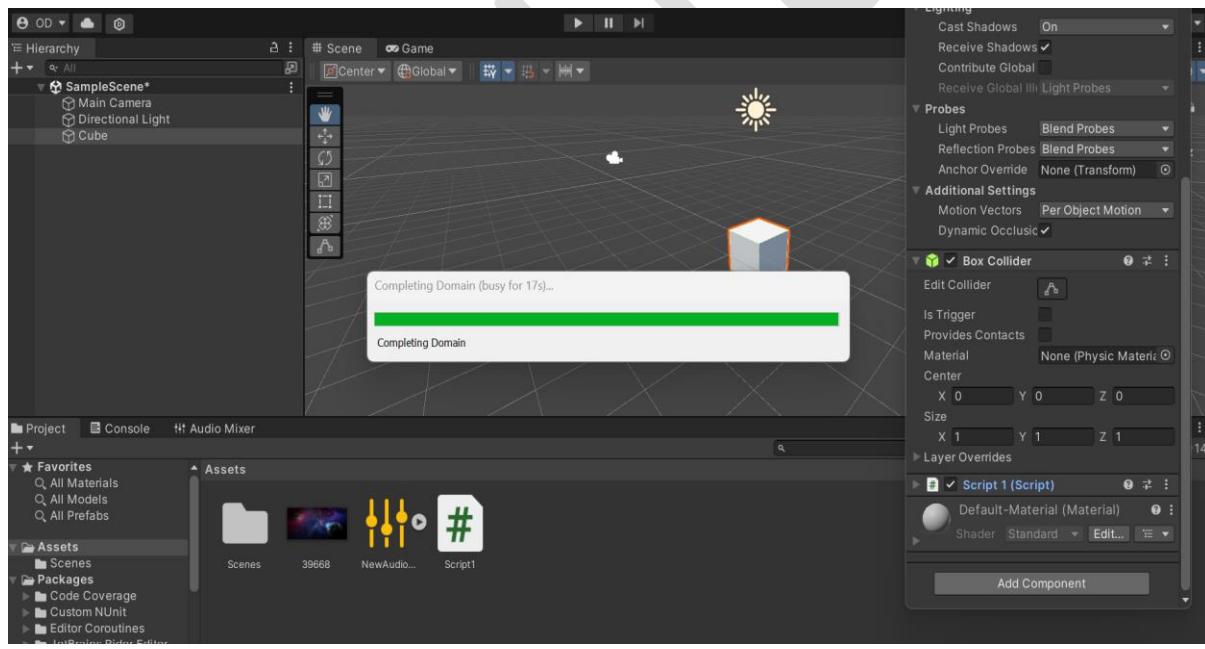




→ Modifying the code:



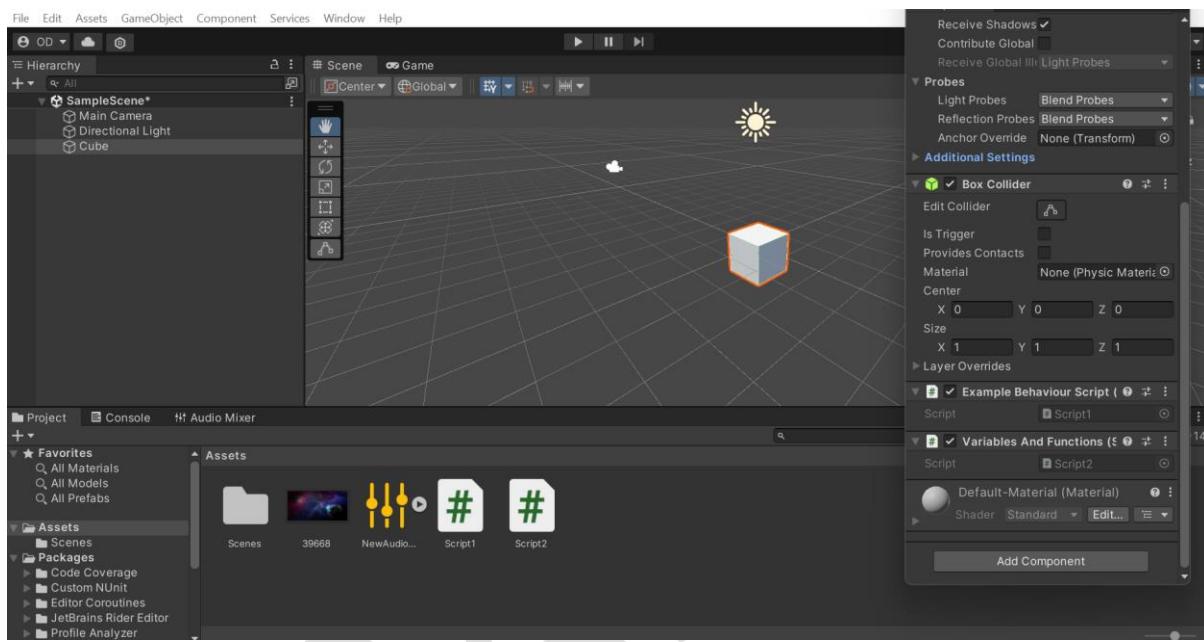
```
Script1.cs
Assets > Script1.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class ExampleBehaviourScript : MonoBehaviour
5  {
6      void Update()
7      {
8          if (Input.GetKeyDown(KeyCode.R))
9          {
10              GetComponent<Renderer>().material.color = Color.red;
11          }
12          if (Input.GetKeyDown(KeyCode.G))
13          {
14              GetComponent<Renderer>().material.color = Color.green;
15          }
16          if (Input.GetKeyDown(KeyCode.B))
17          {
18              GetComponent<Renderer>().material.color = Color.blue;
19          }
20      }
21 }
```



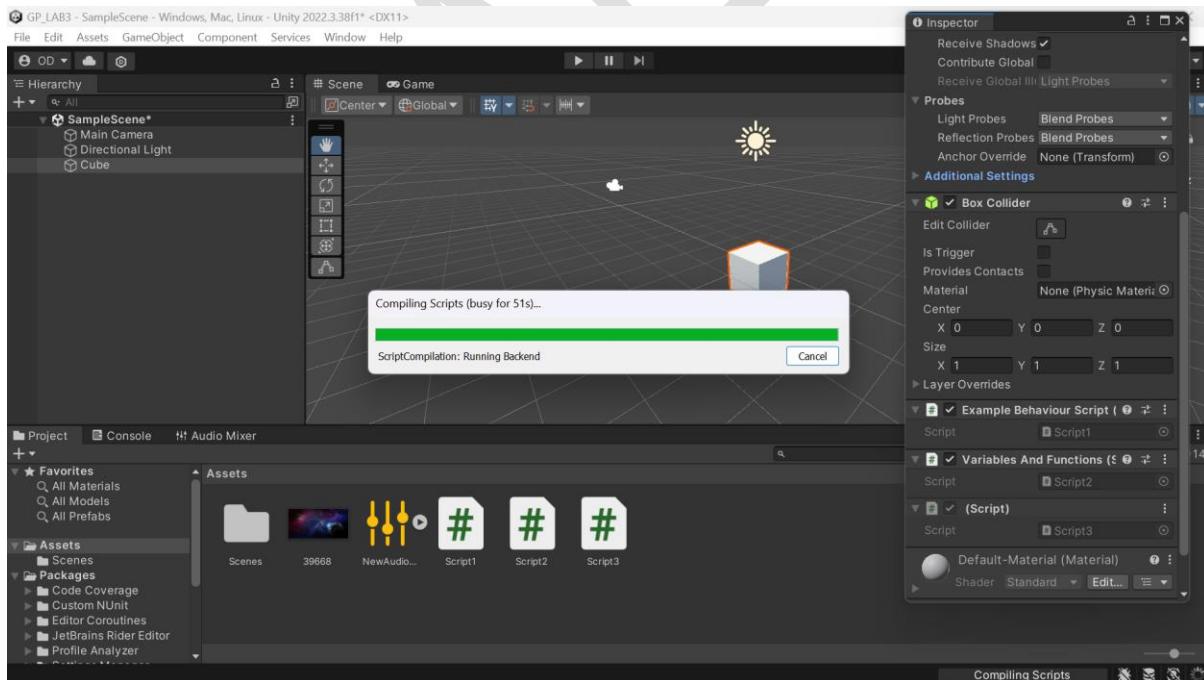
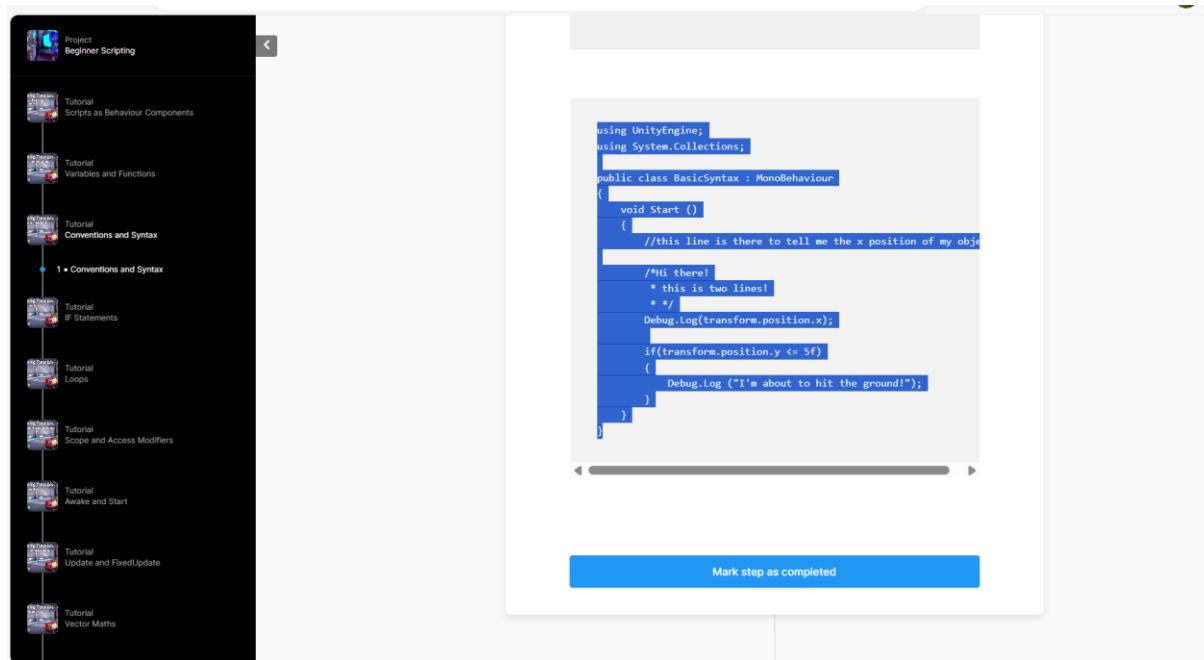
Similarly, we have to add more scripts in the following way.

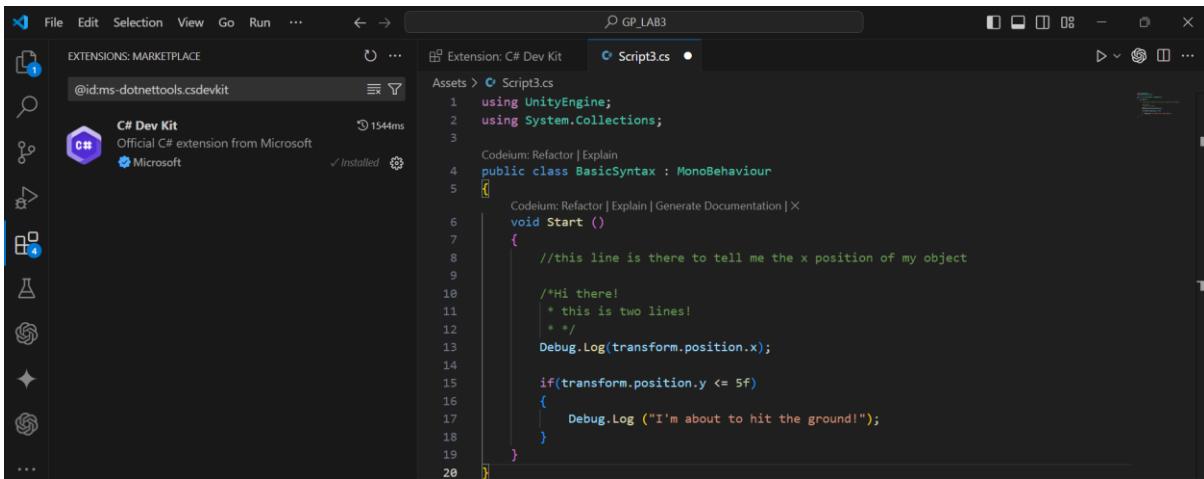
Step 2: For variables and functions

Similarly after modifying the code, we get following scenario.



Step 3: Conventions and syntax





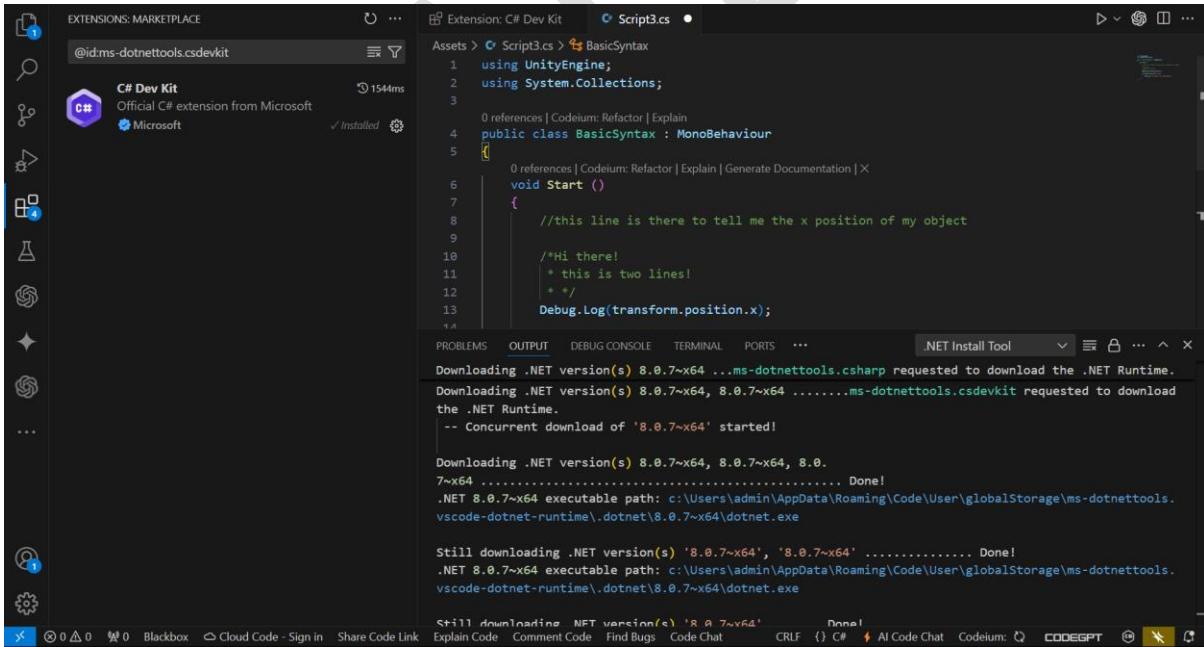
A screenshot of the Visual Studio Code interface. The title bar says "GP_LAB". The left sidebar shows icons for file operations, search, and extensions. The main area displays a C# script named "Script3.cs" under the "Assets > Script3.cs" path. The code uses UnityEngine and System.Collections namespaces. It contains a Start() method with a Debug.Log statement and a conditional check for transform.position.y. The status bar at the bottom shows "1544ms" and other development tools.

```
using UnityEngine;
using System.Collections;

public class BasicSyntax : MonoBehaviour
{
    void Start ()
    {
        //this line is there to tell me the x position of my object
        /*Hi there!
         * this is two lines!
         */
        Debug.Log(transform.position.x);

        if(transform.position.y <= 5f)
        {
            Debug.Log ("I'm about to hit the ground!");
        }
    }
}
```

Code updates as well as extensions extra installations and updates.



A screenshot of the Visual Studio Code interface. The title bar says "GP_LAB". The left sidebar shows icons for file operations, search, and extensions. The main area displays a C# script named "Script3.cs" under the "Assets > Script3.cs > BasicSyntax" path. The code is identical to the previous screenshot. Below the editor, a terminal window titled ".NET Install Tool" shows the process of downloading .NET runtime components. The status bar at the bottom shows "1544ms" and other development tools.

```
using UnityEngine;
using System.Collections;

public class BasicSyntax : MonoBehaviour
{
    void Start ()
    {
        //this line is there to tell me the x position of my object
        /*Hi there!
         * this is two lines!
         */
        Debug.Log(transform.position.x);
    }
}
```

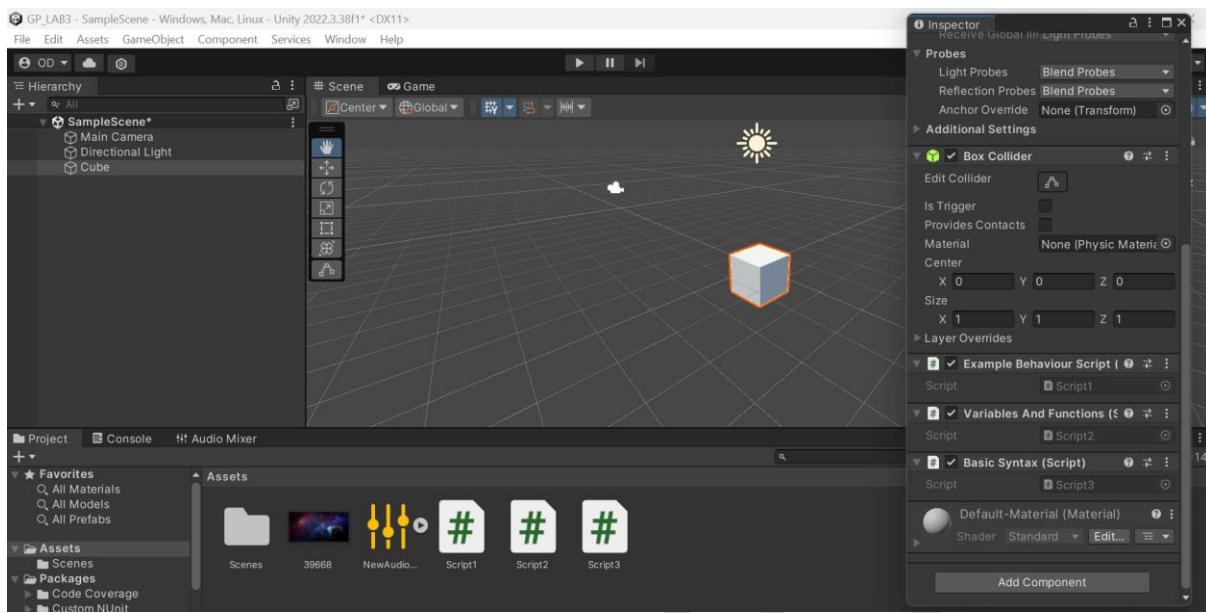
.NET Install Tool

```
Downloading .NET version(s) 8.0.7~x64 ...ms-dotnettools.csharp requested to download the .NET Runtime.
Downloading .NET version(s) 8.0.7~x64 .....ms-dotnettools.csdevkit requested to download the .NET Runtime.
-- Concurrent download of '8.0.7~x64' started!

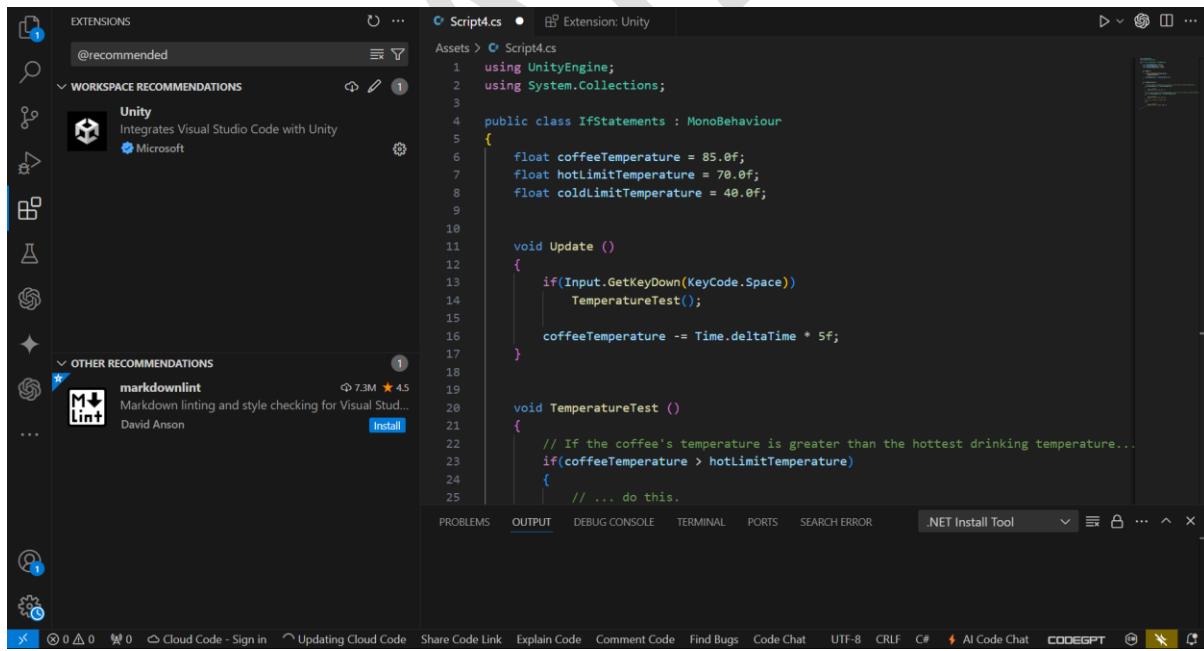
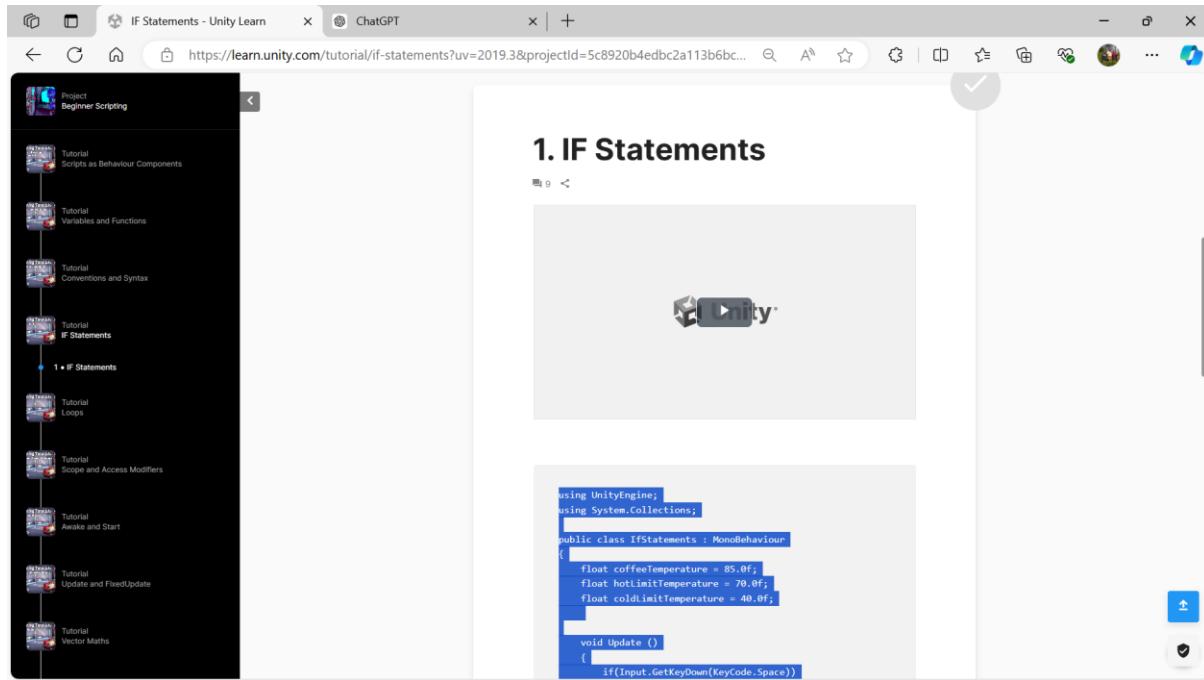
Downloading .NET version(s) 8.0.7~x64, 8.0.7~x64, 8.0.
7~x64 ..... Done!
.NET 8.0.7~x64 executable path: c:\Users\admin\AppData\Roaming\Code\User\globalStorage\ms-dotnettools.vscode-dotnet-runtime\.dotnet\8.0.7~x64\dotnet.exe

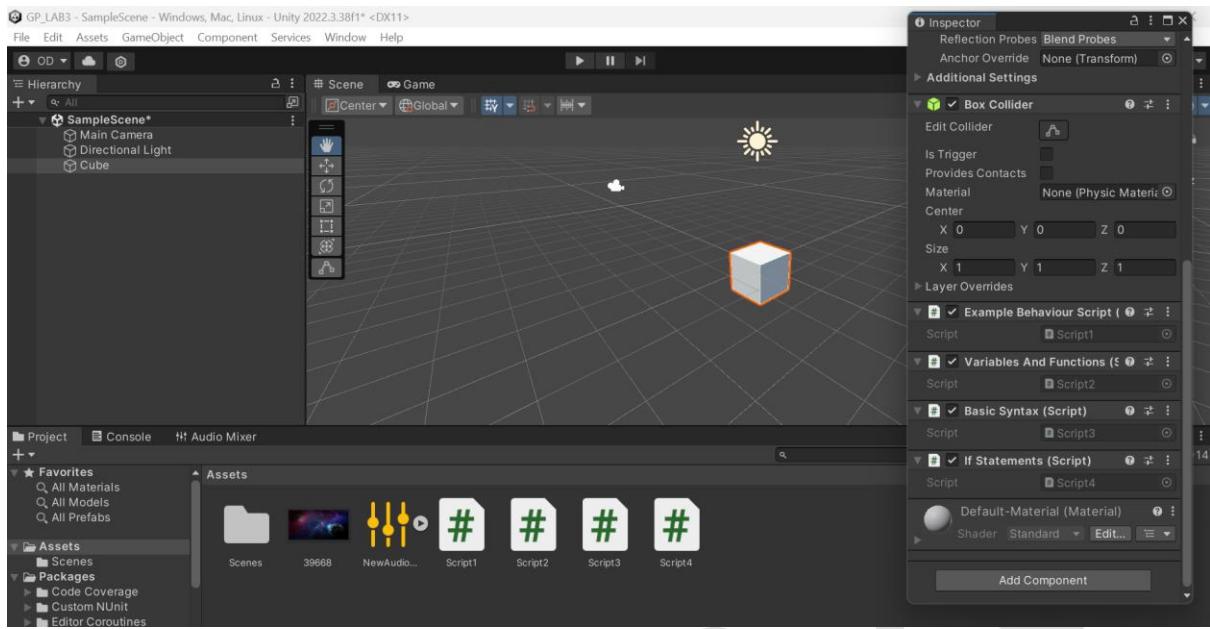
Still downloading .NET version(s) '8.0.7~x64', '8.0.7~x64' .....
.NET 8.0.7~x64 executable path: c:\Users\admin\AppData\Roaming\Code\User\globalStorage\ms-dotnettools.vscode-dotnet-runtime\.dotnet\8.0.7~x64\dotnet.exe

Still downloading .NET version(s) '8.0.7~x64' .....
.NET 8.0.7~x64 executable path: c:\Users\admin\AppData\Roaming\Code\User\globalStorage\ms-dotnettools.vscode-dotnet-runtime\.dotnet\8.0.7~x64\dotnet.exe
```

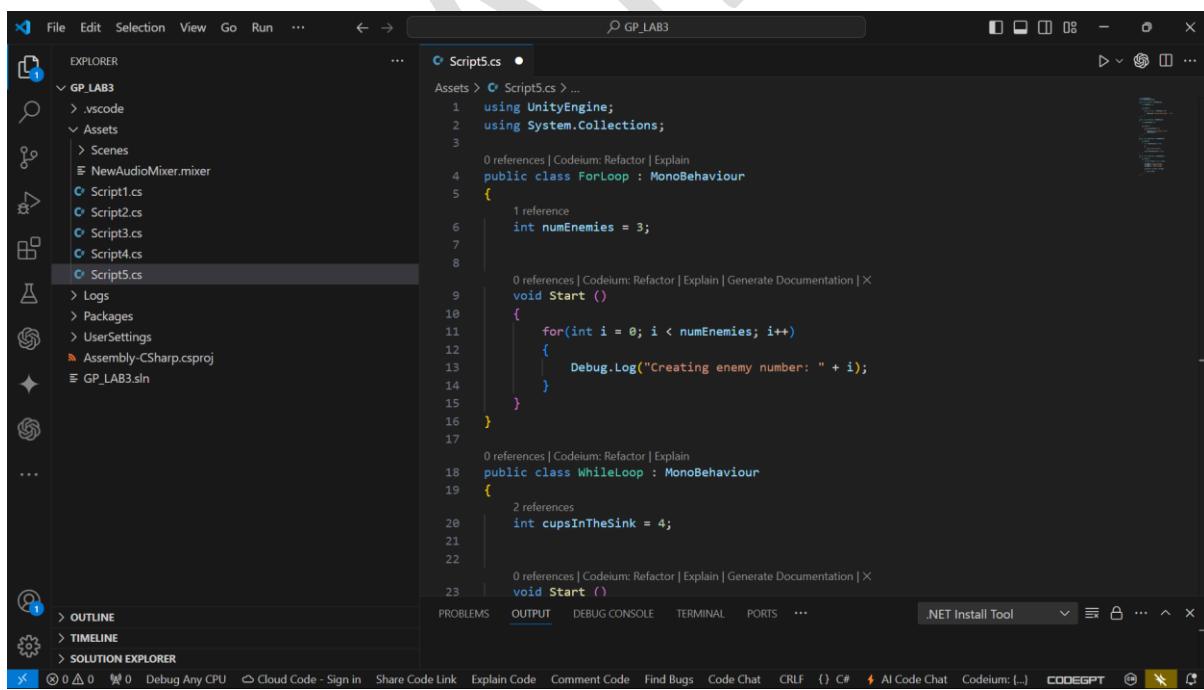
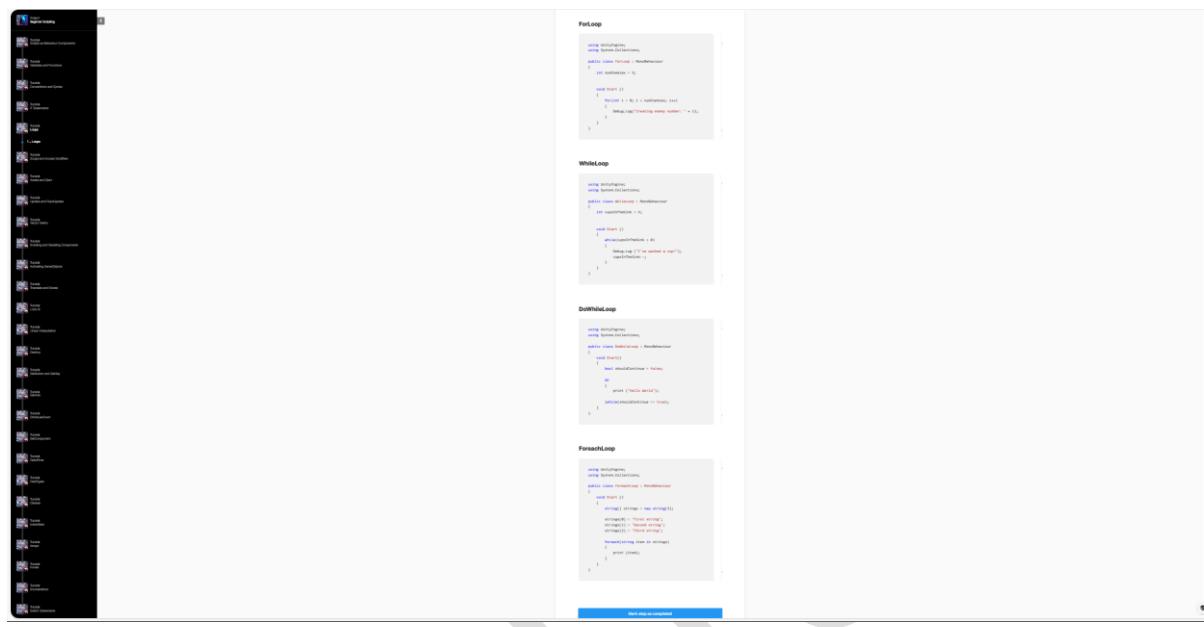


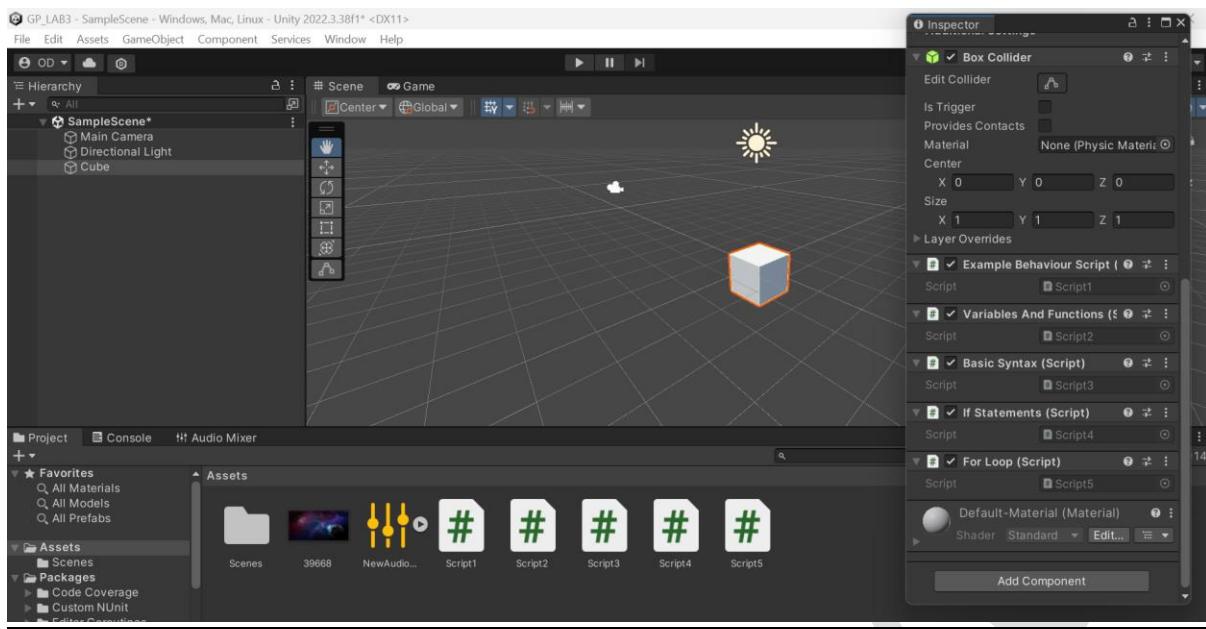
Step 4: If statements





Step 5: Combining all loop code snippets in a single script





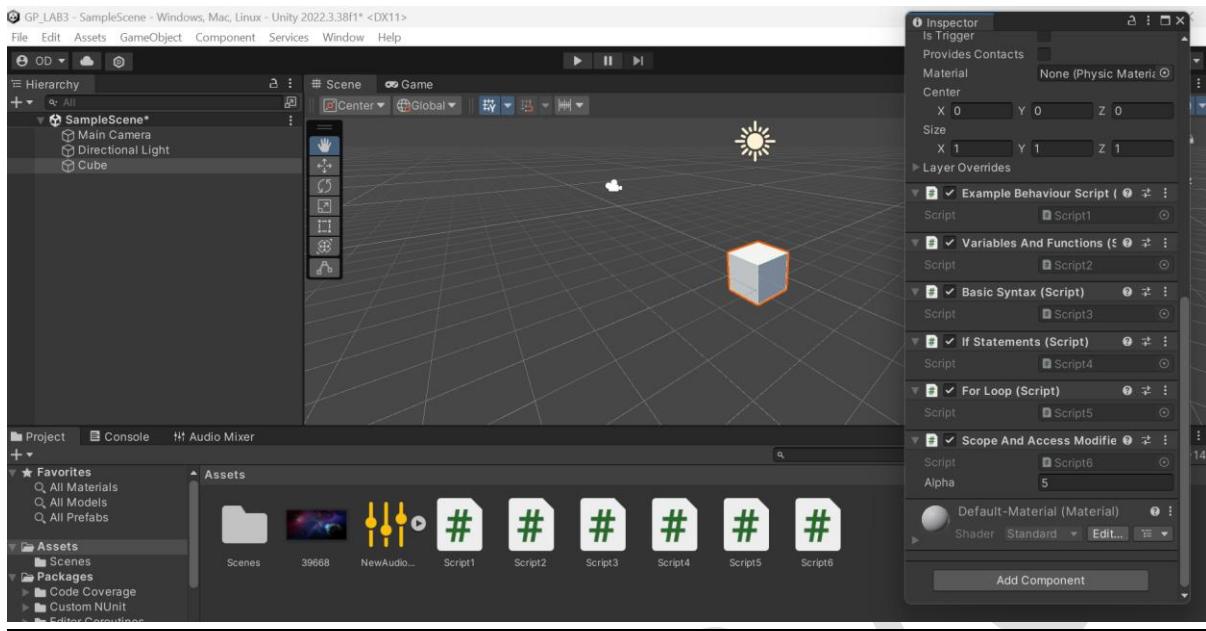
Step 6: Scope and Access Modifiers

The screenshot shows the Unity Learn platform. On the left, a sidebar lists several tutorials under the 'Beginner Scripting' project, with 'Scope and Access Modifiers' highlighted. The main content area displays the details for the 'Scope and Access Modifiers' tutorial, which is rated 4 stars and includes 409 lessons. It features an overview, summary, and a section to select the Unity version (2019.3 is selected). Below the main content, there's a large watermark for 'NOV'.

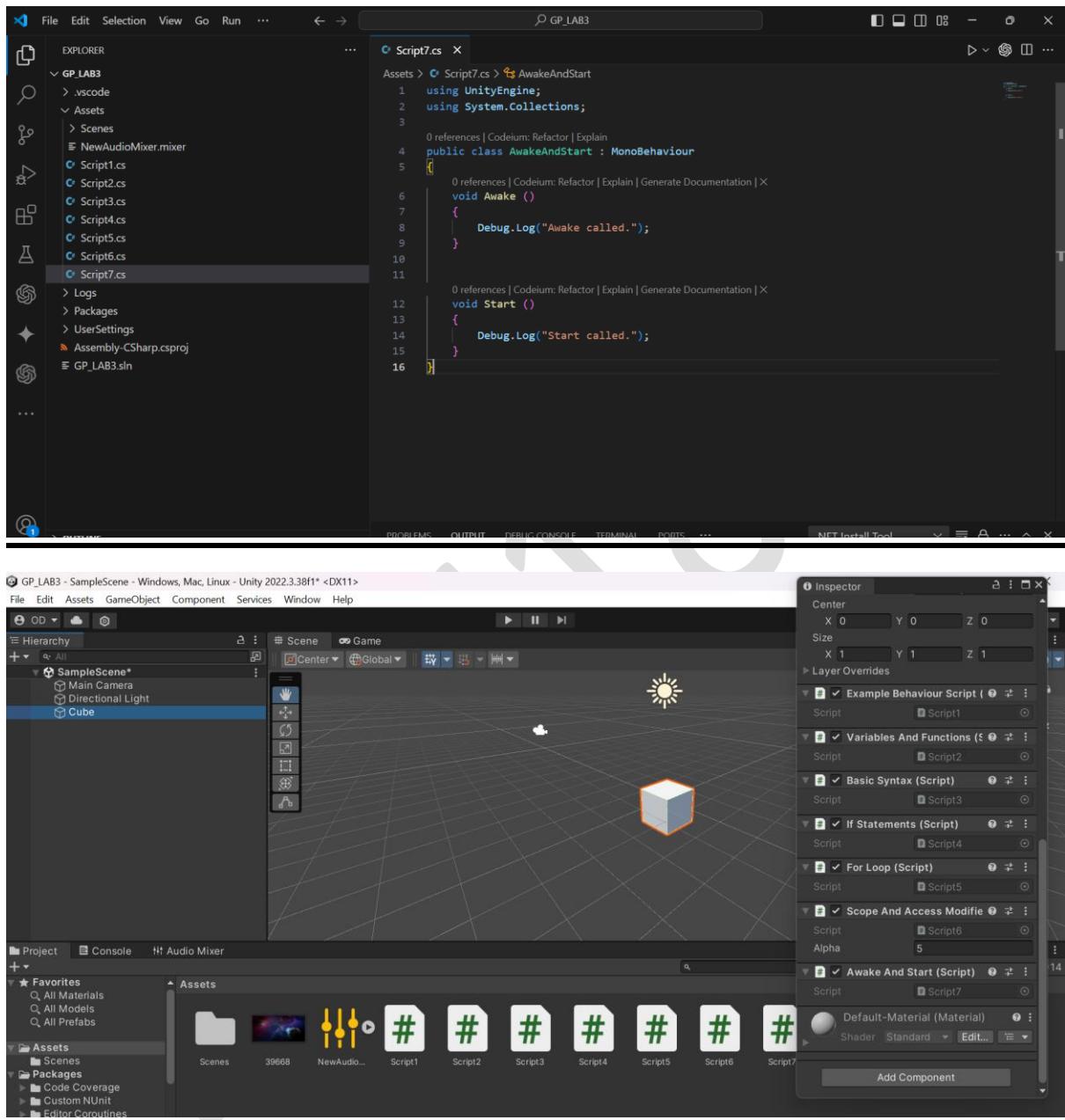
The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left shows a project structure for 'GP_LAB3' containing files like 'Script1.cs' through 'Script6.cs'. The main editor window displays the 'Script6.cs' file with the following C# code:

```
Assets > Script6.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class ScopeAndAccessModifiers : MonoBehaviour
5  {
6      public int alpha = 5;
7
8      private int beta = 0;
9      private int gamma = 5;
10
11     private AnotherClass myOtherClass;
12
13
14     void Start ()
15     {
16         alpha = 29;
17
18         myOtherClass = new AnotherClass();
19         myOtherClass.FruitMachine(alpha, myOtherClass.apples);
20     }
21
22
23
24
25     void Example (int pens, int crayons)
26     {
27         int answer;
28         answer = pens * crayons * alpha;
29     }
30 }
```

The bottom of the screen shows the VS Code status bar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, SEARCH ERROR, and Tasks.



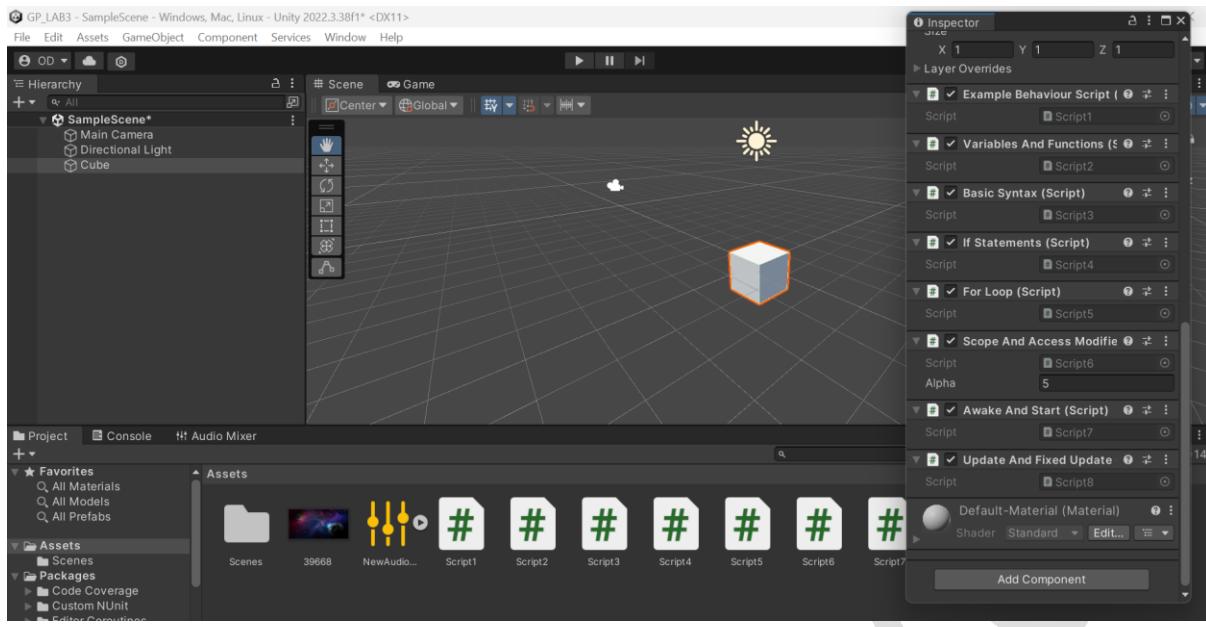
Step 7: Awake and Start



Step 8: Update and FixedUpdate

The screenshot shows a web browser displaying the Unity Learn website at <https://learn.unity.com/tutorial/update-and-fixedupdate?uv=2019.3&projectId=5c8920b4edbc...>. The page title is "Update and FixedUpdate". On the left, there's a sidebar titled "Project Beginner Scripting" with a list of tutorials: "Scripts as Behaviour Components", "Variables and Functions", "Conventions and Syntax", "IF Statements", "Loops", "Scope and Access Modifiers", and "Awake and Start". The main content area shows the "Update and FixedUpdate" tutorial, which is a Beginner level tutorial with 10 XP and 5 minutes duration. It has a rating of 4.5 stars from 3638 reviews. The Unity Technologies logo is present. Below the title, there's an "Overview" section with a brief description: "How to effect changes every frame with the Update and FixedUpdate functions, and their differences." A note states, "This tutorial is included in the Beginner Scripting project." To the right, there's a "Select your Unity version" dropdown set to "2019.3" and a "Last updated: November 17, 2023" message.

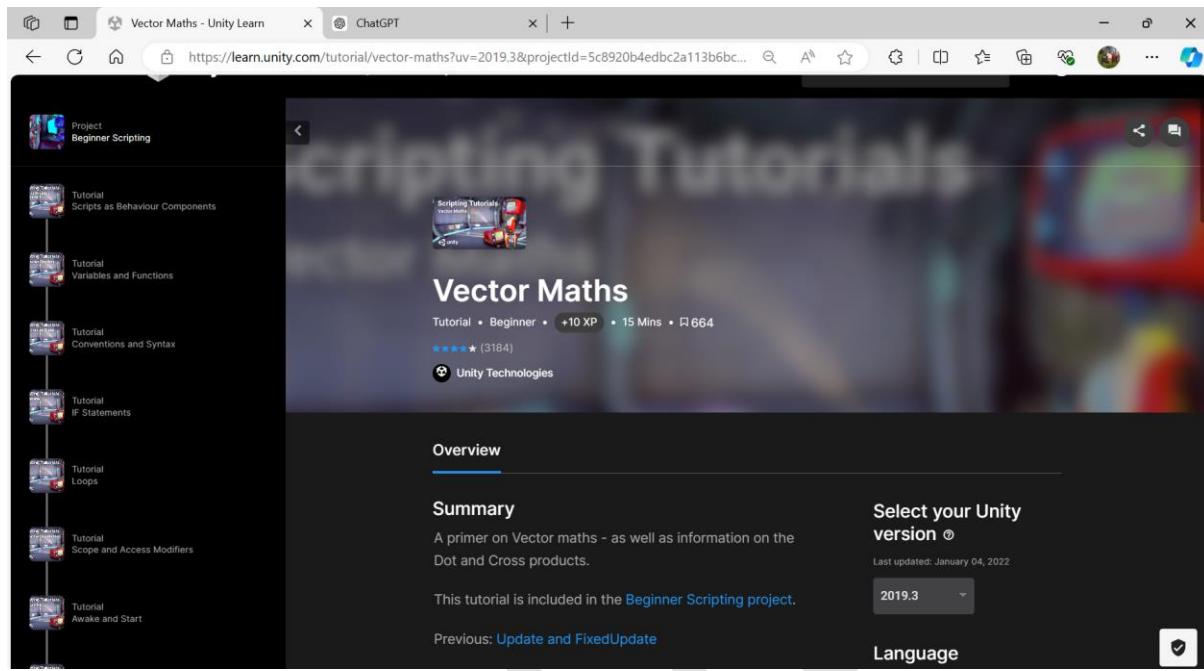
The screenshot shows the Unity Editor interface with the title bar "GP_LAB3 - SampleScene - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>". The menu bar includes File, Edit, Assets, GameObject, Component, Services, Window, and Help. The toolbar has icons for OD, Scene, Game, and various camera and light settings. The Hierarchy panel shows a "SampleScene" with "Main Camera", "Directional Light", and a "Cube". The Scene view shows a 3D grid with a cube and a sun icon. A progress bar at the bottom of the scene view says "Completing Domain (busy for 11s)..." and "Completing Domain". The Project panel shows "Scenes", "39668", "NewAudio...", "Script1", "Script2", "Script3", "Script4", "Script5", "Script6", "Script7", and "Script8". The Assets panel shows "Scenes", "39668", "NewAudio...", "Script1", "Script2", "Script3", "Script4", "Script5", "Script6", "Script7", and "Script8". The Inspector panel on the right lists several scripts attached to the cube: "Example Behaviour Script", "Variables And Functions", "Basic Syntax (Script)", "If Statements (Script)", "For Loop (Script)", "Scope And Access Modifie", "Awake And Start (Script)", and "(Script)". Each script has its name listed under "Script". A "Default-Material (Material)" is also listed in the Inspector.



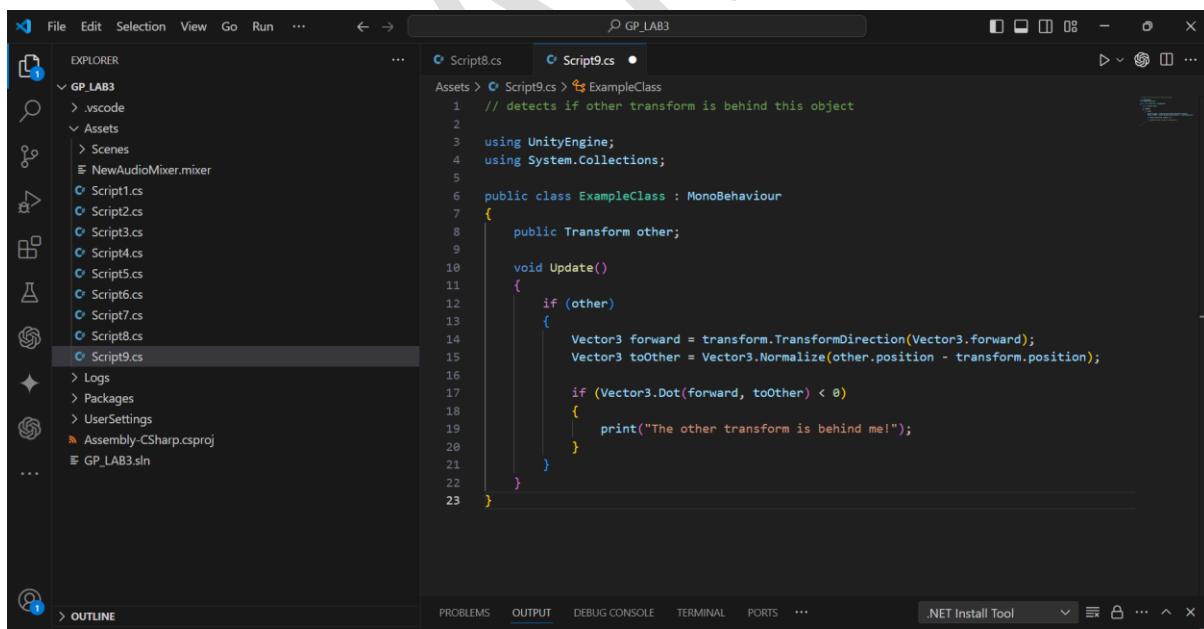
The VS Code interface is shown, displaying the "Script8.cs" file. The code is as follows:

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class UpdateAnd FixedUpdate : MonoBehaviour
5  {
6
7      void FixedUpdate ()
8      {
9          Debug.Log("FixedUpdate time :" + Time.deltaTime);
10     }
11
12     void Update ()
13     {
14         Debug.Log("Update time :" + Time.deltaTime);
15     }
16 }
```

Step 9: Vector Maths



The screenshot shows a web browser window with the URL <https://learn.unity.com/tutorial/vector-maths?uv=2019.3&projectId=5c8920b4edbc2a113b6bc...>. The page displays a "Vector Maths" tutorial from the "Scripting Tutorials" section. On the left, there's a sidebar with a list of other tutorials: "Project Beginner Scripting", "Tutorial Scripts as Behaviour Components", "Tutorial Variables and Functions", "Tutorial Conventions and Syntax", "Tutorial IF Statements", "Tutorial Loops", "Tutorial Scope and Access Modifiers", and "Tutorial Awake and Start". The main content area shows the "Vector Maths" tutorial details, including its rating (4 stars), duration (15 Mins), and points (664). Below this is an "Overview" section with a summary of the tutorial's content and information about its inclusion in the "Beginner Scripting" project. A "Select your Unity version" dropdown is set to "2019.3". At the bottom, there are links for "Previous: Update and FixedUpdate" and "Language".

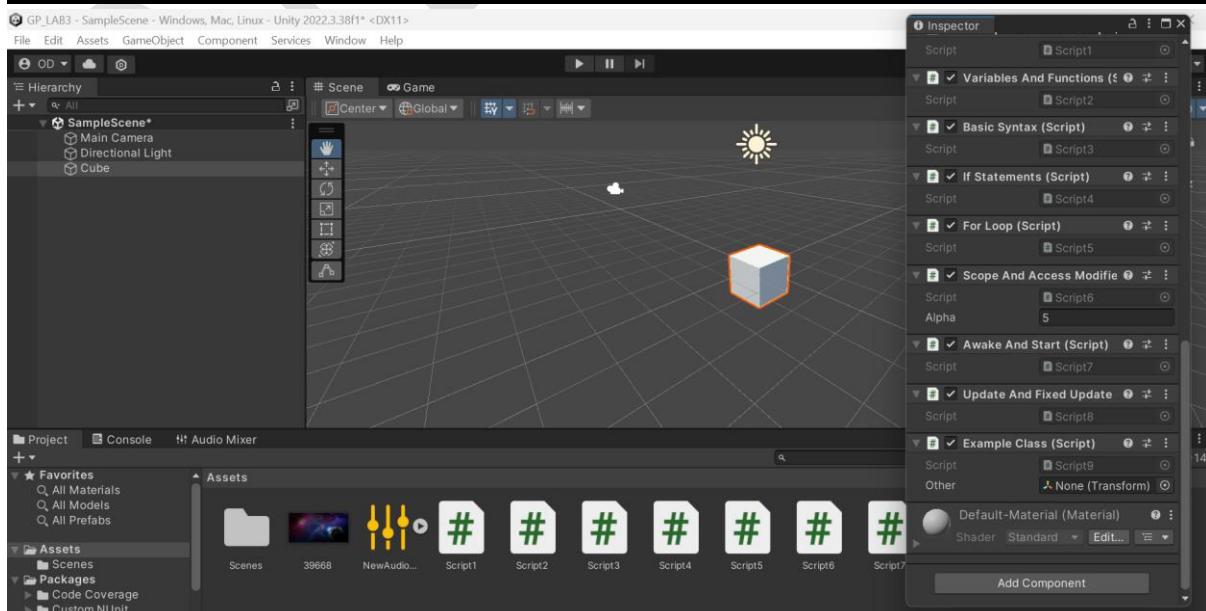
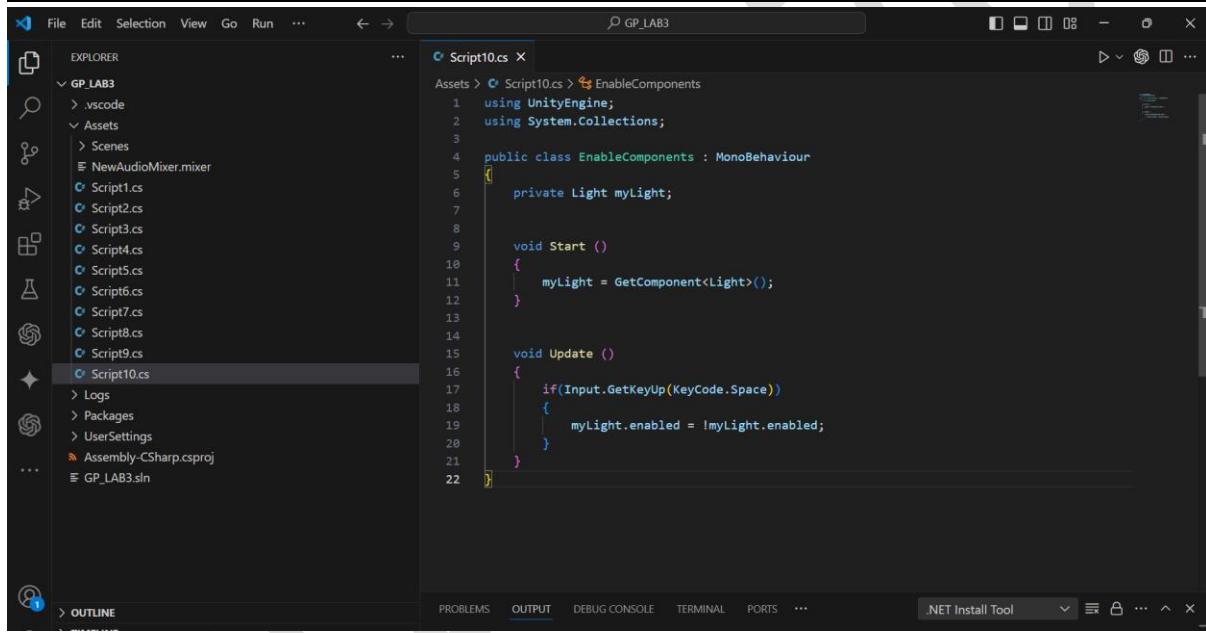
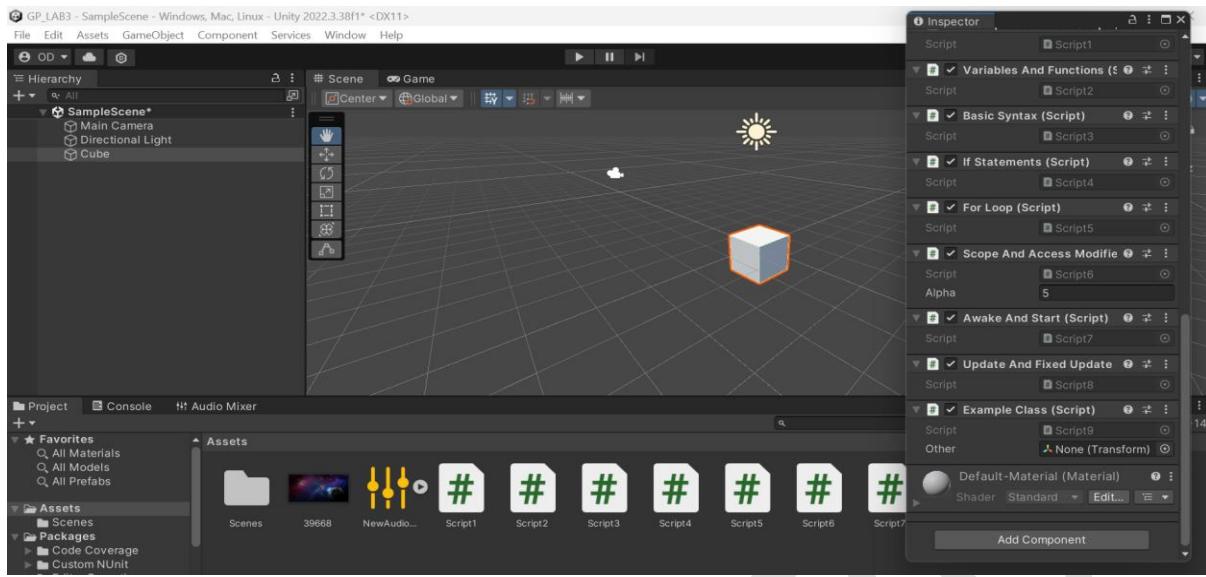


The screenshot shows the Visual Studio Code (VS Code) interface. The title bar says "GP_LAB3". The Explorer sidebar on the left shows a project structure with files like "Script1.cs", "Script2.cs", "Script3.cs", "Script4.cs", "Script5.cs", "Script6.cs", "Script7.cs", "Script8.cs", and "Script9.cs". The main editor pane displays a C# script named "Script9.cs". The code is as follows:

```
Assets > Script9.cs > ExampleClass
1 // detects if other transform is behind this object
2
3 using UnityEngine;
4 using System.Collections;
5
6 public class ExampleClass : MonoBehaviour
7 {
8     public Transform other;
9
10    void Update()
11    {
12        if (other)
13        {
14            Vector3 forward = transform.TransformDirection(Vector3.forward);
15            Vector3 toOther = Vector3.Normalize(other.position - transform.position);
16
17            if (Vector3.Dot(forward, toOther) < 0)
18            {
19                print("The other transform is behind me!");
20            }
21        }
22    }
23 }
```

The bottom of the screen shows the VS Code navigation bar with tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", "PORTS", and "NET Install Tool".

Step 10: Enabling and disabling components



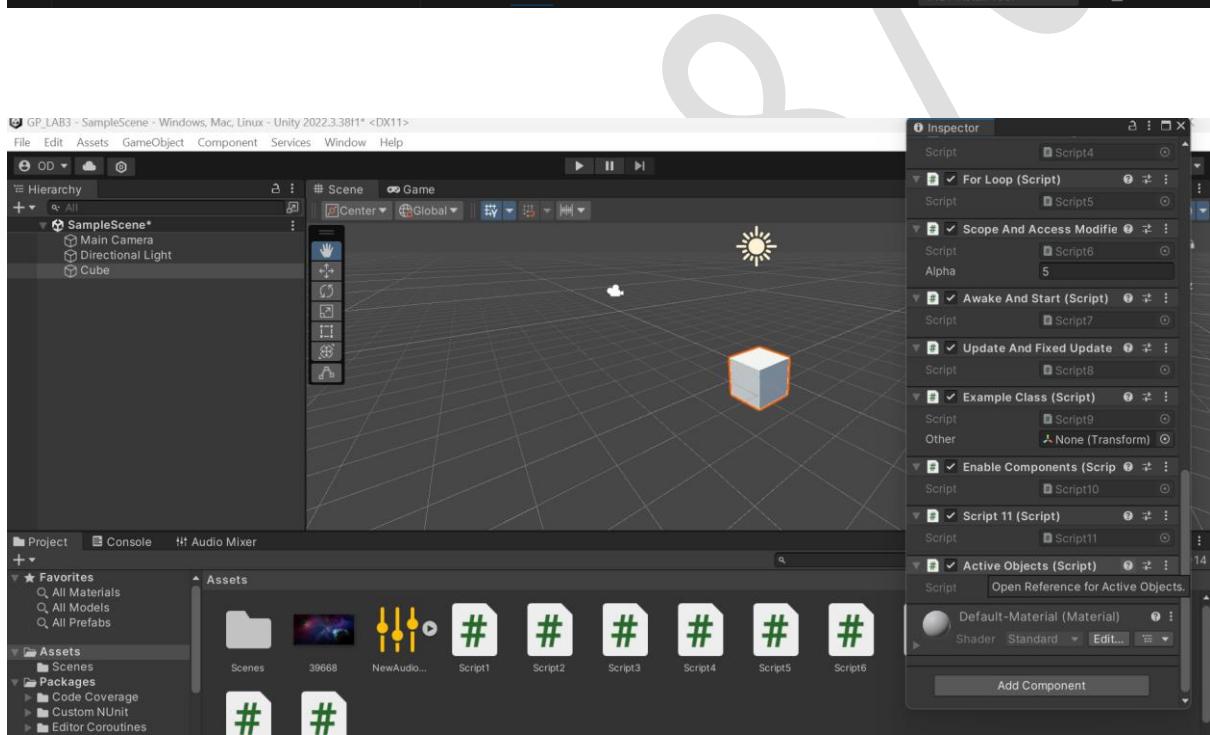
Step 11: Activating GameObjects

The screenshot shows a web browser displaying the Unity Learn website at <https://learn.unity.com/tutorial/activating-gameobjects?uv=2019.3&projectId=5c8920b4edbc2...>. The page title is "Activating GameObjects". On the left, there's a sidebar with a "Project Beginner Scripting" navigation tree. The main content area shows the "Activating GameObjects" tutorial card, which includes a thumbnail, a title, a brief description, a rating of 4.5 stars (2505 reviews), and a "Unity Technologies" logo. Below the card, there's an "Overview" section and a "Summary" section. To the right, there's a "Select your Unity version" dropdown set to "2019.3".

The screenshot shows the Unity Editor interface with the title bar "GP_LAB3 - SampleScene - Windows, Mac, Linux - Unity 2022.3.38f1* <DX11>". The main view displays a 3D scene with a cube, a directional light, and a camera. The Hierarchy panel shows a "SampleScene*" root with "Main Camera", "Directional Light", and "Cube" children. The Project panel shows various assets like "Scenes", "NewAudio...", and multiple "Script" files. The Inspector panel on the right lists several script components attached to objects in the scene, including "If Statements (Script)", "For Loop (Script)", "Scope And Access Modifiers (Script)", "Alpha (float)", "Awake And Start (Script)", "Update And FixedUpdate (Script)", and "Example Class (Script)". A warning message at the bottom of the Inspector panel states: "Assets\Script6.cs(9,17): warning CS0414: The field 'ScopeAndAccessModifiers.beta' is assigned but its value is never used."

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project named "GP_LAB3" containing files like Script1.cs, Script2.cs, Script3.cs, Script4.cs, Script5.cs, Script6.cs, Script7.cs, Script8.cs, Script9.cs, Script10.cs, Script11.cs, and Script12.cs.
- Code Editor:** Displays the content of Script12.cs, which includes code for ActiveObjects and CheckState MonoBehaviour classes.
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, SEARCH ERROR, and .NET Install Tool.



Step 12: Translate and Rotate

Unity Learn

Project Beginner Scripting

Tutorial Scripts as Behaviour Components

Tutorial Variables and Functions

Tutorial Conventions and Syntax

Tutorial IF Statements

Tutorial Loops

Tutorial Scope and Access Modifiers

Tutorial Awake and Start

Scripting Tutorials

Translate and Rotate

Tutorial • Beginner • +10 XP • 5 Mins • 434

4.5 (2594)

Unity Technologies

Overview

Summary

How to use the two transform functions Translate and Rotate to effect a non-rigidbody object's position and rotation.

This tutorial is included in the [Beginner Scripting project](#).

Select your Unity version

Last updated: November 17, 2023

2019.3

File Edit Selection View Go Run ...

EXPLORER

GP_LAB3

.vscode

Assets

Scenes

NewAudioMixer.mixer

Script1.cs

Script2.cs

Script3.cs

Script4.cs

Script5.cs

Script6.cs

Script7.cs

Script8.cs

Script9.cs

Script10.cs

Script11.cs

Script12.cs

Logs

Packages

UserSettings

Assembly-CSharp.csproj

GP_LAB3.sln

Script12.cs

```
using UnityEngine;
using System.Collections;

public class TransformFunctions : MonoBehaviour
{
    public float moveSpeed = 10f;
    public float turnSpeed = 50f;

    void Update ()
    {
        if(Input.GetKey(KeyCode.UpArrow))
            transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);

        if(Input.GetKey(KeyCode.DownArrow))
            transform.Translate(-Vector3.forward * moveSpeed * Time.deltaTime);

        if(Input.GetKey(KeyCode.LeftArrow))
            transform.Rotate(Vector3.up, -turnSpeed * Time.deltaTime);

        if(Input.GetKey(KeyCode.RightArrow))
            transform.Rotate(Vector3.up, turnSpeed * Time.deltaTime);
    }
}
```

File Edit Selection View Go Run ...

EXPLORER

GP_LAB3

Assets

Script4.cs

Script5.cs

Script6.cs

Script7.cs

Script8.cs

Script9.cs

Script10.cs

Script11.cs

Script12.cs

Logs

AssetImportWorker0-prev.log

AssetImportWorker0.log

AssetImportWorker1-prev.log

AssetImportWorker1.log

Packages-Update.log

shadercompiler-AssetImportWorker0.log

shadercompiler-UnityShaderCompiler.exe0.log

shadercompiler-UnityShaderCompiler.exe1.log

shadercompiler-UnityShaderCompiler.exe2.log

shadercompiler-UnityShaderCompiler.exe3.log

shadercompiler-UnityShaderCompiler.exe4.log

shadercompiler-UnityShaderCompiler.exe5.log

shadercompiler-UnityShaderCompiler.exe6.log

shadercompiler-UnityShaderCompiler.exe7.log

Script12.cs

Script12.cs

```
using UnityEngine;
using System.Collections;

public class TransformFunctions : MonoBehaviour
{
    public float moveSpeed = 10f;
    public float turnSpeed = 50f;

    void Update ()
    {
        if(Input.GetKey(KeyCode.UpArrow))
            transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);

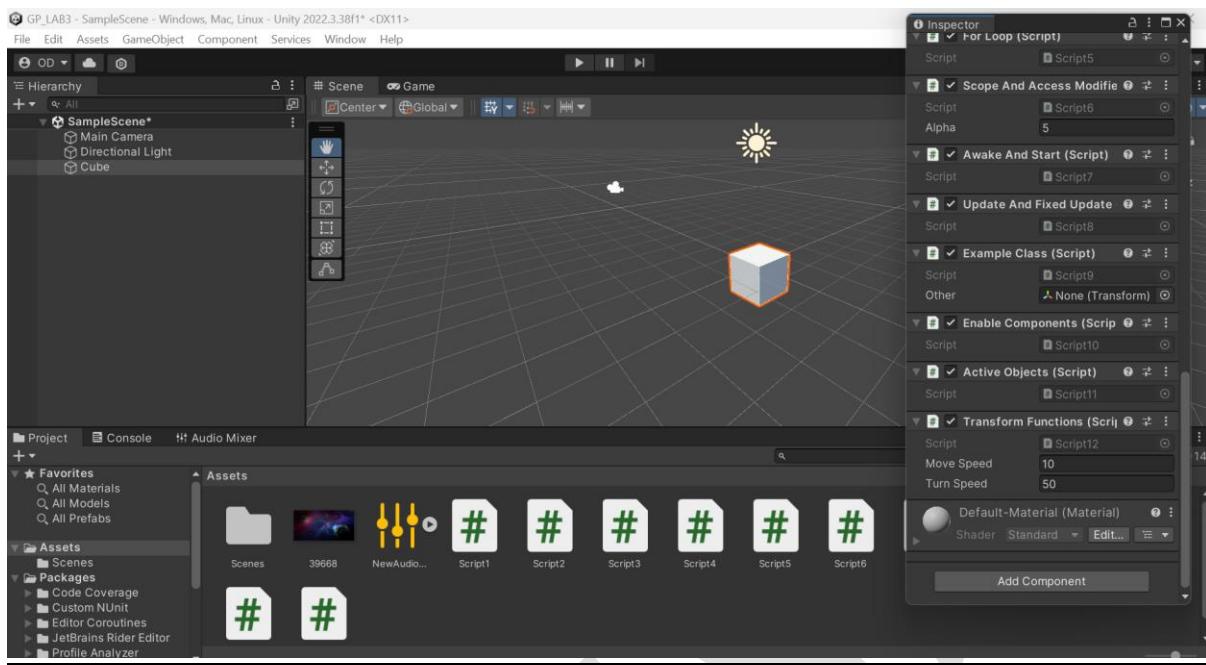
        if(Input.GetKey(KeyCode.DownArrow))
            transform.Translate(-Vector3.forward * moveSpeed * Time.deltaTime);

        if(Input.GetKey(KeyCode.LeftArrow))
            transform.Rotate(Vector3.up, -turnSpeed * Time.deltaTime);

        if(Input.GetKey(KeyCode.RightArrow))
            transform.Rotate(Vector3.up, turnSpeed * Time.deltaTime);
    }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ...

J.NET Install Tool



Step 13: Adding the final component of the beginner script part 1 – Look At

The screenshot shows the Unity Learn website at <https://learn.unity.com/tutorial/look-at-s?uv=2019.3&projectId=5c8920b4edbc2a113b6bc26a>. The page displays the 'Look At' tutorial from the 'Scripting Tutorials' section. On the left, there's a sidebar with a vertical list of other tutorials under 'Project Beginner Scripting'. The main content area shows the title 'Look At' with a rating of 4.5 stars (2336 reviews), created by 'Unity Technologies'. Below the title is an 'Overview' section with a brief description: 'How to make a game object's transform face another's by using the LookAt function.' To the right, there's a 'Select your Unity version' dropdown set to '2019.3'.

The screenshot shows the Unity Editor interface with the project titled 'GP_LAB3 - SampleScene - Windows, Mac, Linux - Unity 2022.3.8f1* <DX11>'. The top menu bar includes File, Edit, Assets, GameObject, Component, Services, Window, and Help. The bottom navigation bar shows tabs for Project, Console, and Audio Mixer. The left side features the Hierarchy panel with objects like Main Camera, Directional Light, and Cube. The center 3D Viewport shows a cube and a directional light. A 'Reloading Domain' progress bar is visible in the center. The right side is the Inspector panel, which is currently expanded to show the 'Scope And Access Modifiers' tab. It lists several components and their scripts, such as 'Awake And Start (Script)', 'Update And FixedUpdate (Script)', 'Example Class (Script)', 'Enable Components (Script)', 'Active Objects (Script)', 'Transform Functions (Script)', and '(Script)'. Scripts like Script6, Script7, Script8, Script9, Script10, Script11, Script12, Script13, and Default-Material (Material) are listed under these components.

```

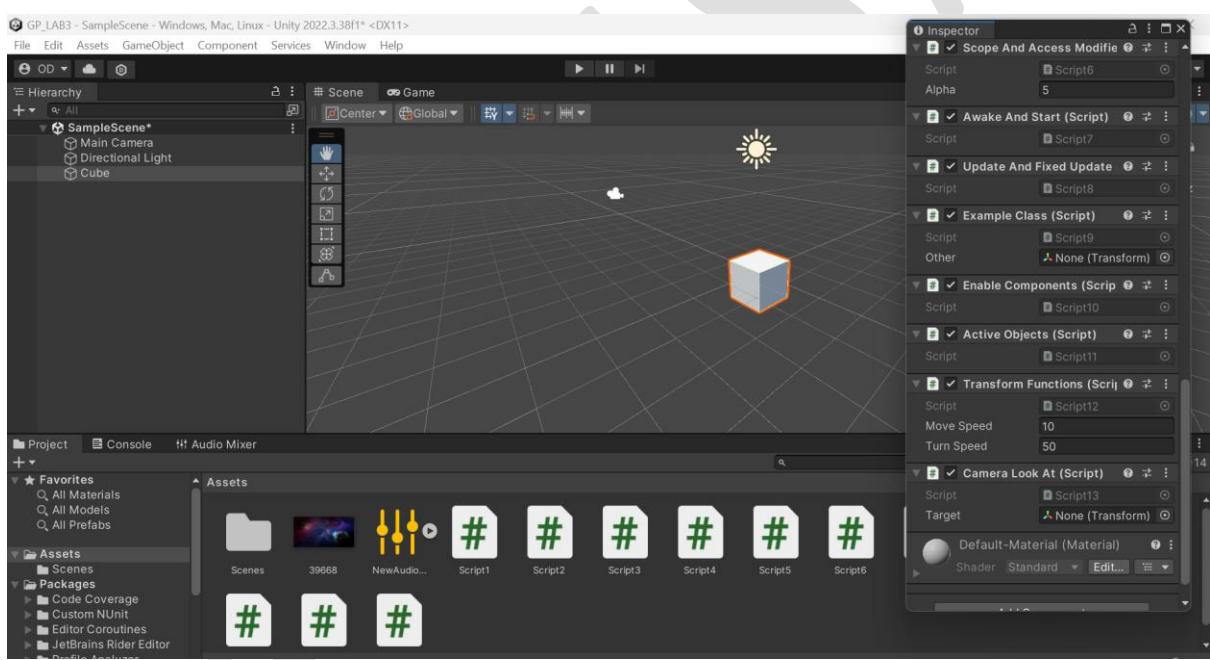
File Edit Selection View Go Run ...
GP_LAB3
Assets
Script1.cs
Script2.cs
Script3.cs
Script4.cs
Script5.cs
Script6.cs
Script7.cs
Script8.cs
Script9.cs
Script10.cs
Script11.cs
Script12.cs
Script13.cs
Logs
manifest.json
packages-lock.json
UserSettings
Layouts
Search.settings
Assembly-CSharp.csproj
GP_LAB3.sln
OUTLINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS .NET Install Tool

```

```

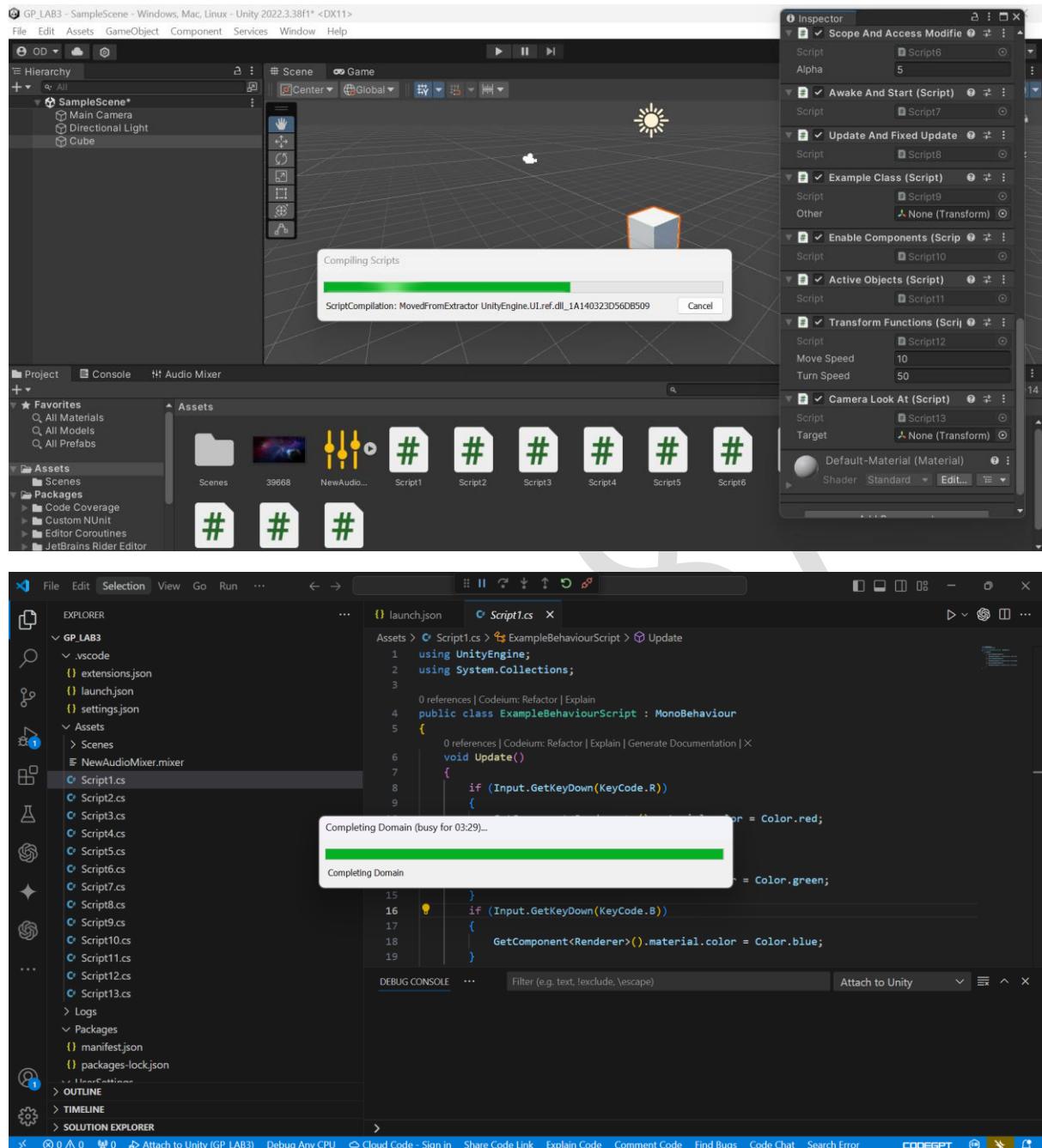
1 using UnityEngine;
2 using System.Collections;
3
4 public class CameraLookAt : MonoBehaviour
5 {
6     public Transform target;
7
8     void Update ()
9     {
10         transform.LookAt(target);
11     }
12 }

```



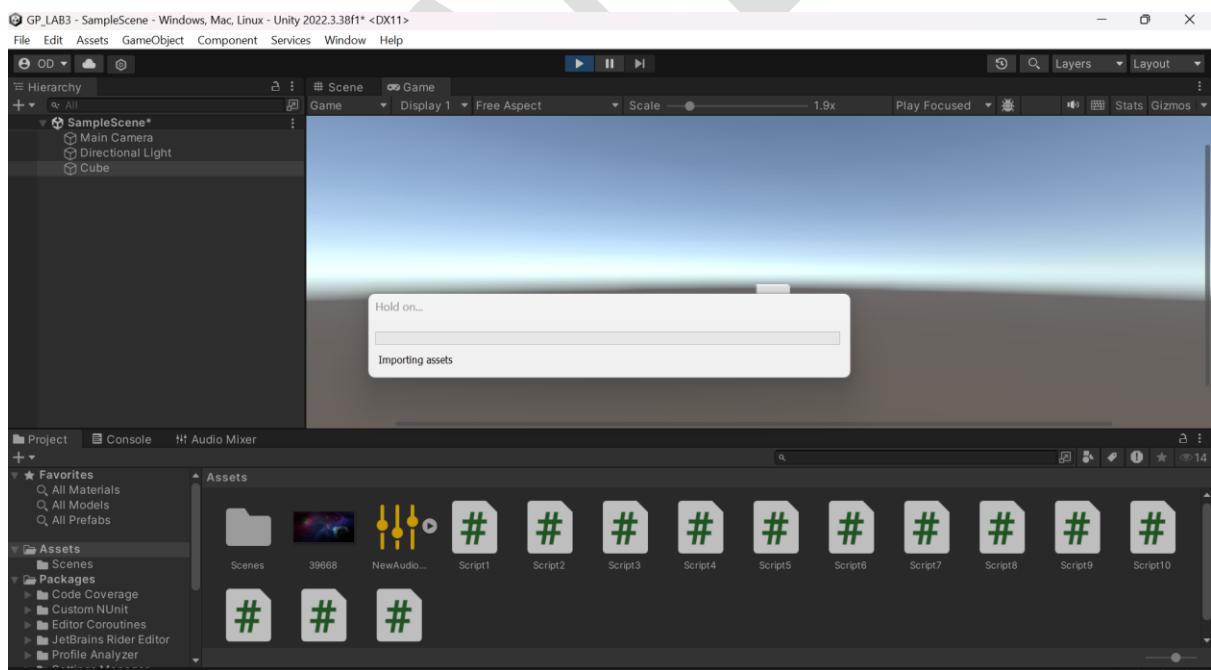
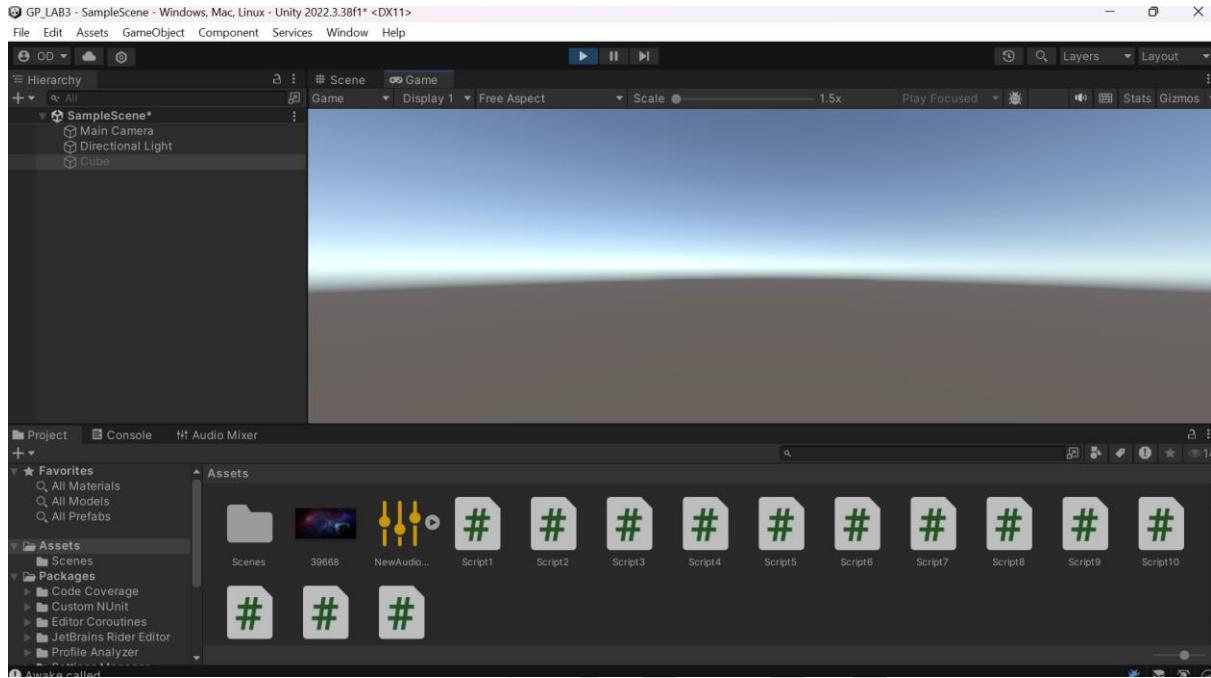
Now we have to execute all the program files in script through the Visual Studio Code using suitable extensions that make it feasible to do so. Before executing the scripts, we need to debug and compile them first and then run those respective files. We will do them in the respective order as per the scripts feeded to the object components.

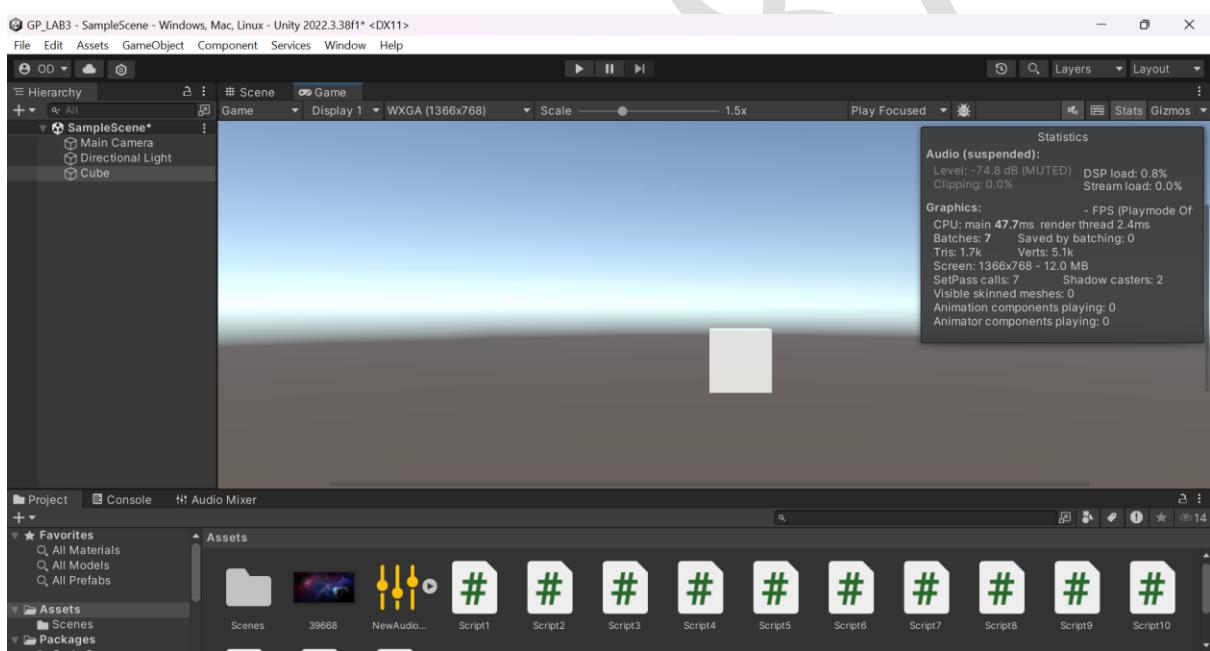
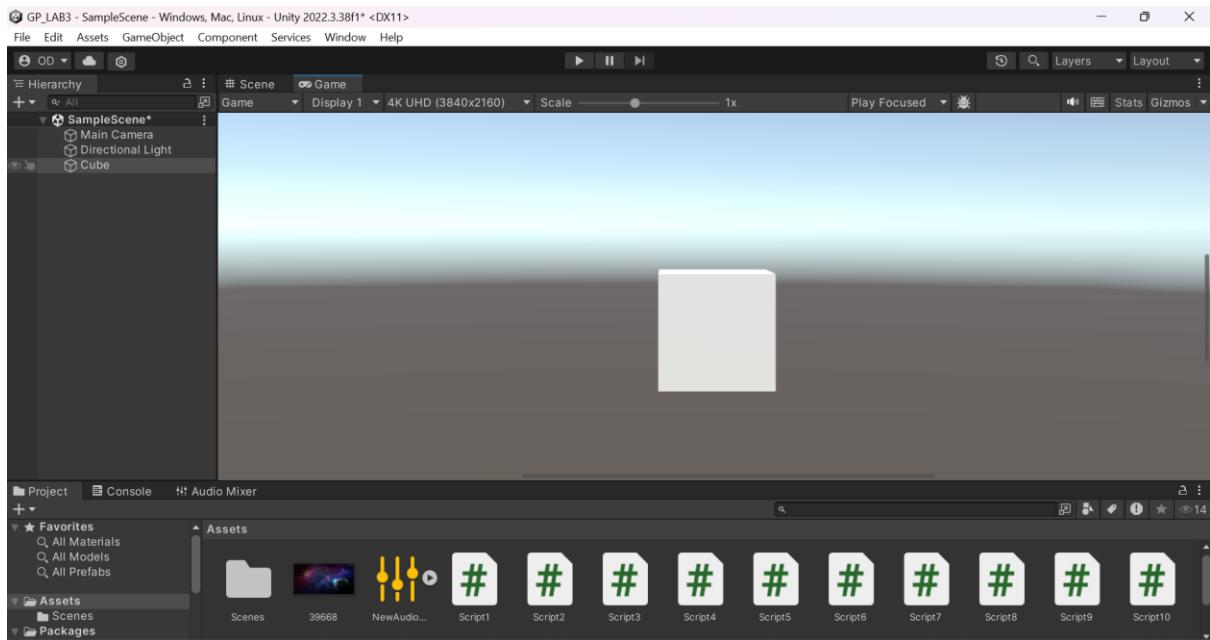
→ Compilation of the scripts:



Similarly followed for other code snippets.

→ Few final stages of Compilation and execution:





21BAI1816

21BAI1816