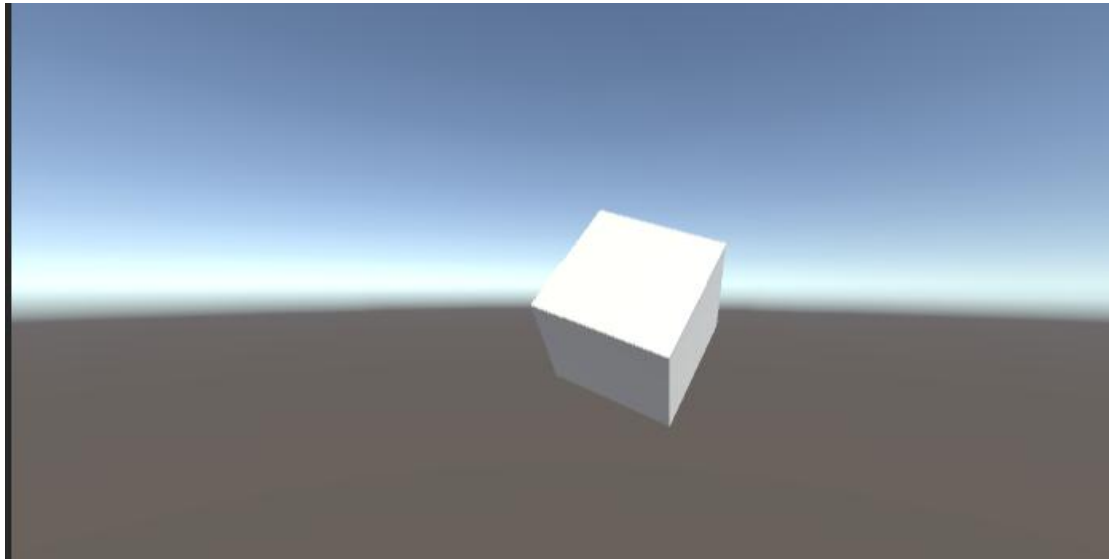


GAME PROGRAMMING LAB-4

- Meghna Sinha, 20BAI1133

BASIC SCRIPTING

We start with our plain scene with only a cube as the game object



Now we will add some scripts to affect the behaviour of this object and the scene

1. KeyCode

These map directly to a physical key on the keyboard

Code :

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

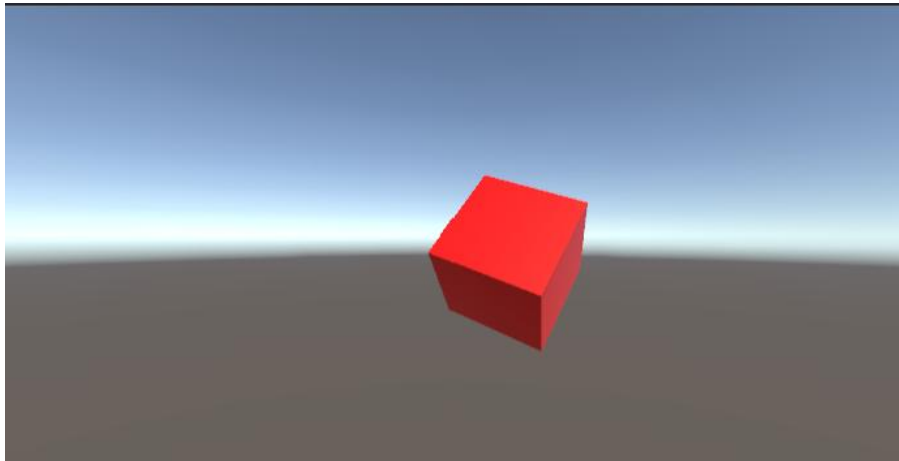
public class ColourBehaviour : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
    }
}
```

```
// Update is called once per frame
void Update()
{
}
```

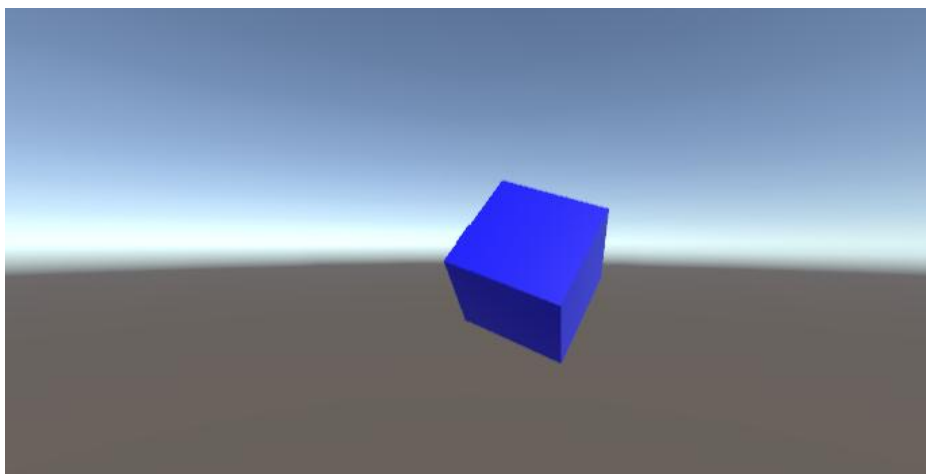
```
    if (Input.GetKeyDown(KeyCode.R))
    {
        GetComponent<Renderer>().material.color = Color.red;
    }
    if (Input.GetKeyDown(KeyCode.G))
    {
        GetComponent<Renderer>().material.color = Color.green;
    }
    if (Input.GetKeyDown(KeyCode.B))
    {
        GetComponent<Renderer>().material.color = Color.blue;
    }
}
}
```

Output:

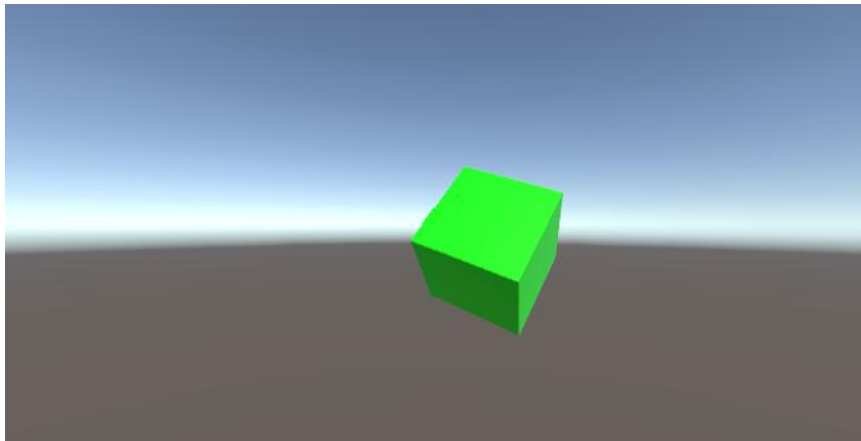
After entering key 'R' (KeyCode R)



After entering key 'B' (KeyCode B)



After entering key 'G' (KeyCode G)



AWAKE and START

Start is called on the frame when a script is enabled just before any of the Update methods are called the first time.

Awake is called when the script object is initialised, regardless of whether or not the script is enabled. Start may not be called on the same frame as Awake if the script is not enabled at initialisation time.

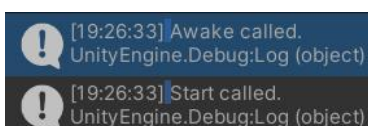
Code :

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

public class StartAwake : MonoBehaviour
{
    void Awake ()
    {
        Debug.Log("Awake called.");
    }

    void Start ()
    {
        Debug.Log("Start called.");
    }
}
```

Output:



```
[19:26:33] Awake called.
UnityEngine.Debug:Log (object)
[19:26:33] Start called.
UnityEngine.Debug:Log (object)
```

UPDATE and FIXED UPDATE

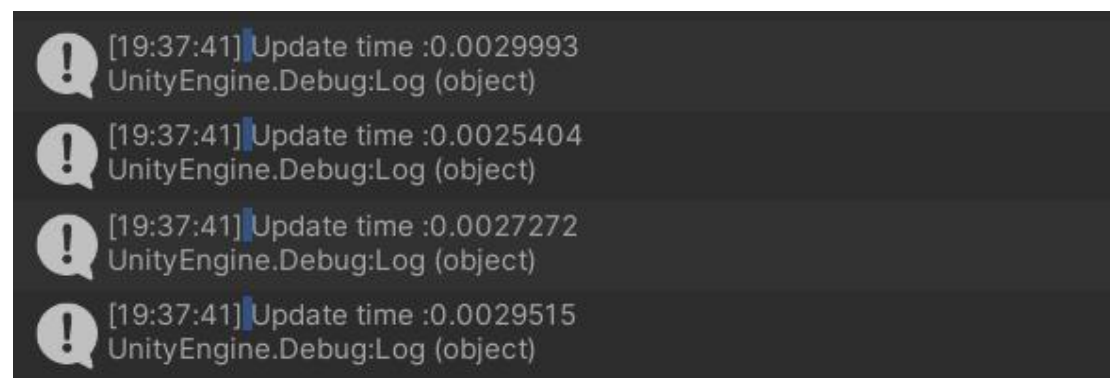
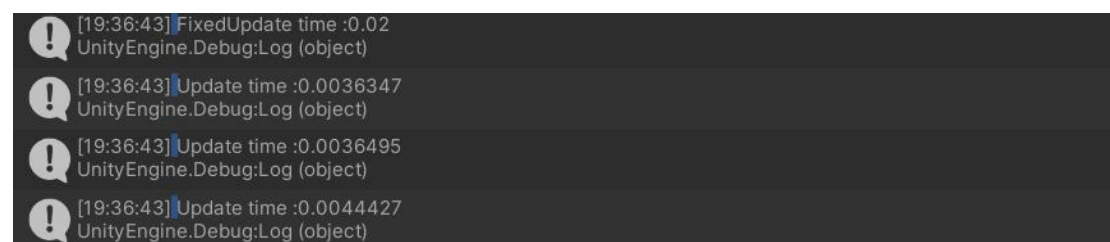
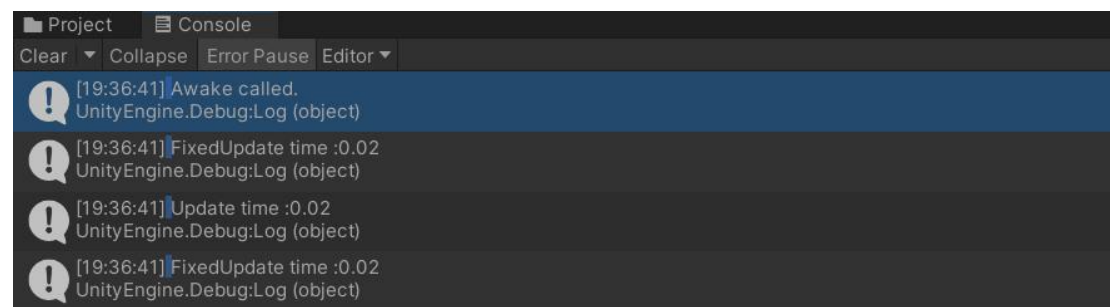
Code :

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

public class UpdateFixedUpdate : MonoBehaviour
{
    void FixedUpdate ()
    {
        Debug.Log("FixedUpdate time :" + Time.deltaTime);
    }

    void Update ()
    {
        Debug.Log("Update time :" + Time.deltaTime);
    }
}
```

Output:



Time.deltaTime()

Time.deltaTime is the completion time in seconds since the last frame. This helps us to make the game frame-independent. That is, regardless of the fps, the game will be executed at the same speed.

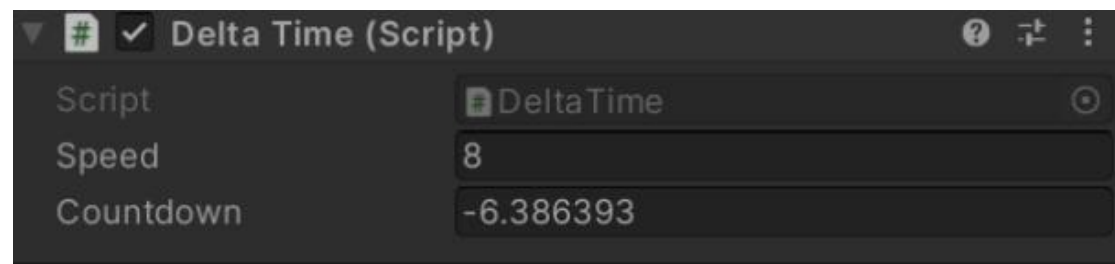
Code :

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

public class Time : MonoBehaviour
{
    public float speed = 8f;
    public float countdown = 3.0f;

    void Update ()
    {
        countdown -= Time.deltaTime;
        if(countdown <= 0.0f)
            GetComponent<Light>().enabled = true;
        if(Input.GetKey(KeyCode.RightArrow))
            transform.position += new Vector3(speed * Time.deltaTime, 0.0f, 0.0f);
    }
}
```

Output:



Debug.log

Logs a message to the Unity Console. We use it to print informational messages that help you debug your application.

Code :

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

public class VariablesAndFunctions : MonoBehaviour
```

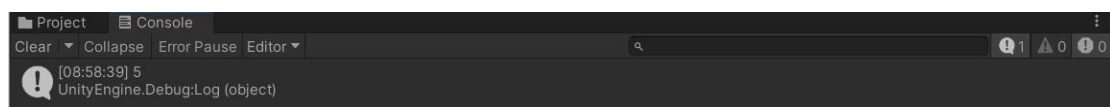
```
{  
    int myInt = 1;
```

```
// Start is called before the first frame update  
void Start()  
{  
    myInt = MultiplyByFive(myInt);  
    Debug.Log (myInt);  
}
```

```
int MultiplyByFive (int number)  
{  
    int result;  
    result = number * 5;  
  
    return result;  
}
```

```
// Update is called once per frame  
void Update()  
{  
  
}  
}
```

Output :



If Conditions

If conditions are used when our program needs to make a decision based on some values. The following code demonstrates if the candy we have is too sweet, too sour or just right based on its taste.

Code :

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class IfStatements : MonoBehaviour  
{  
    float candy = 80.0f;  
    float sweet = 70.0f;  
    float sour = 30.0f;
```

```
// Start is called before the first frame update  
void Start()  
{
```

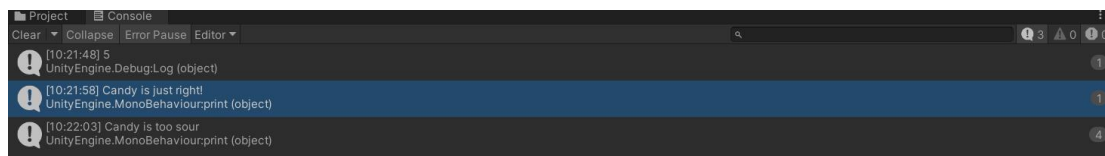
```
}
```

```
// Update is called once per frame
void Update()
{
    if(Input.GetKeyDown(KeyCode.Space))
        TasteTest();

    candy -= Time.deltaTime * 5f;
}
```

```
void TasteTest()
{
    if(candy > sweet)
    {
        print("Candy is too sweet");
    }
    else if(candy < sour)
    {
        print("Candy is too sour");
    }
    else
    {
        print("Candy is just right!");
    }
}
```

Output:



Loops

Loops are used when we wish to repeat a certain instruction 'n' number of times. We have different types of loops mainly - For Loop, While Loop, Do While Loop and For Each Loop

For Loop Code:

```
using UnityEngine;
using System.Collections;

public class ForLoop : MonoBehaviour
{
    int numDish = 3;

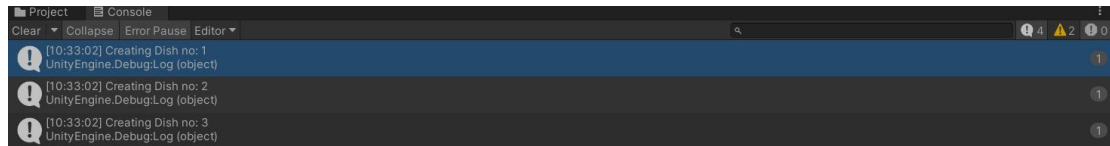
    void Start ()
    {
```

```

        for(int i = 0; i < numDish; i++)
        {
            Debug.Log("Creating Dish no: " + (i+1));
        }
    }
}

```

Output:



While Loop Code:

```

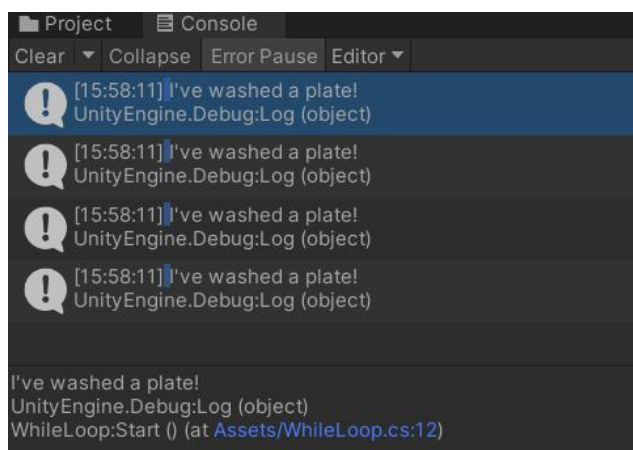
using UnityEngine;
using System.Collections;

public class WhileLoop : MonoBehaviour
{
    int platesInTheSink = 4;

    void Start ()
    {
        while(cupsInTheSink > 0)
        {
            Debug.Log ("I've washed a plate!");
            platesInTheSink--;
        }
    }
}

```

Output:



Do While Loop Code :

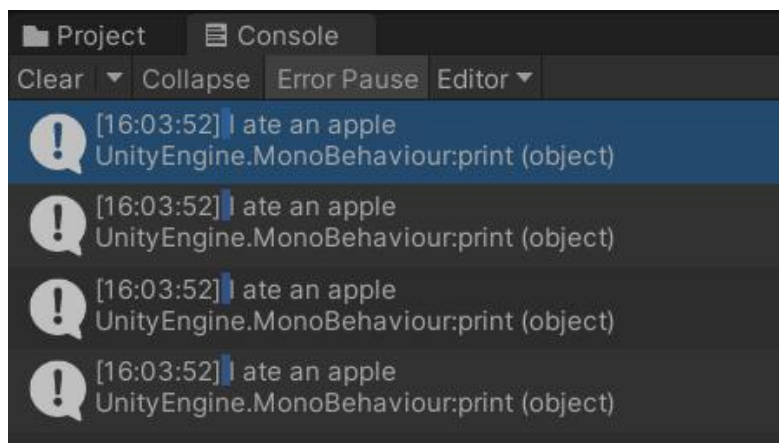
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DoWhile : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        int apples = 4;

        do
        {
            print ("I ate an apple");
            apples--;

        }while(apples>0);
    }
}
```

Output:



For Each Loop Code :

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

public class ForEachLoop : MonoBehaviour
{
    // Start is called before the first frame update
    void Start ()
    {
        string[] strings = new string[3];

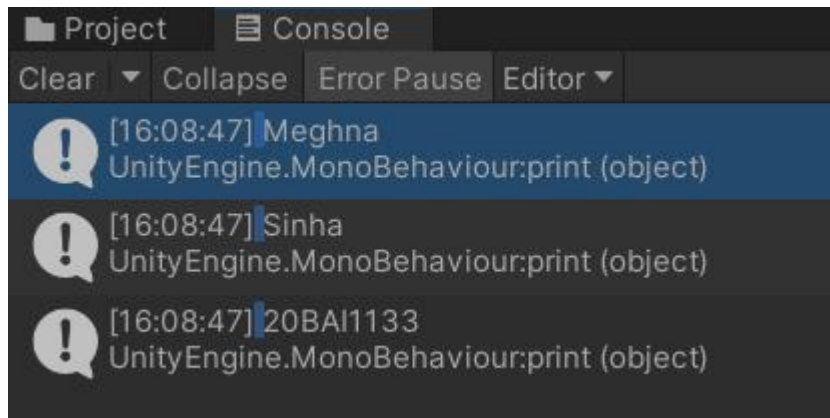
        strings[0] = "Meghna";
        strings[1] = "Sinha";
        strings[2] = "20BAI1133";
    }
}
```

```

        foreach(string item in strings)
        {
            print (item);
        }
    }
}

```

Output:



Enabling and Disabling Components

We can enhance our game object by adding a component which can be enabled or disabled based on user input. To demonstrate that we will be adding a light component to our object.

```

using UnityEngine;
using System.Collections;

public class EnableComponents : MonoBehaviour
{
    private Light myLight;

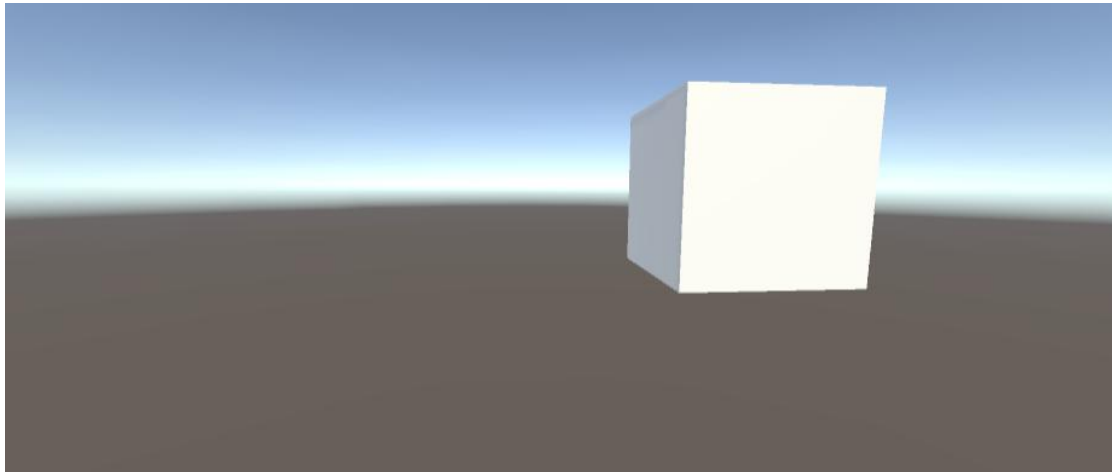
    void Start ()
    {
        myLight = GetComponent<Light>();
    }

    void Update ()
    {
        if(Input.GetKeyUp(KeyCode.Space))
        {
            myLight.enabled = !myLight.enabled;
        }
    }
}

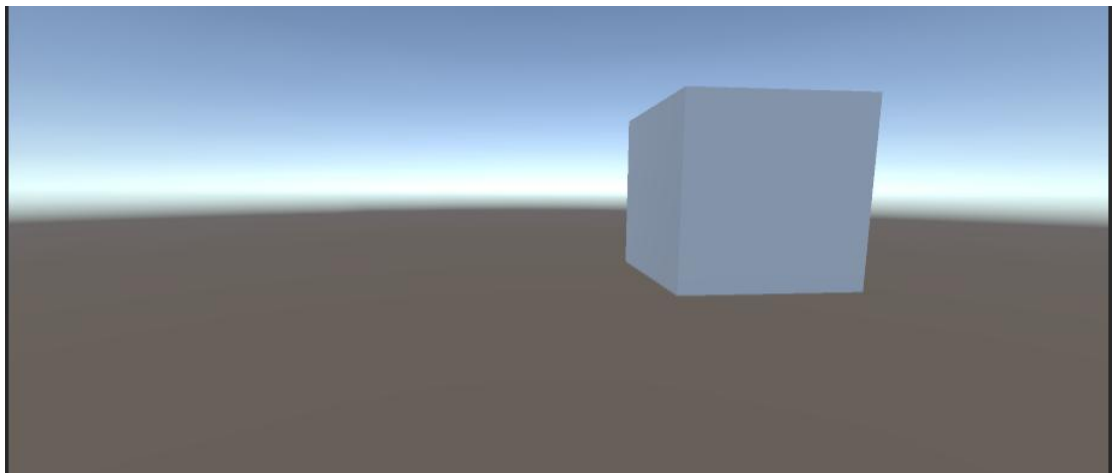
```

Output

Light is On :



Light is Off :



Activating Game Objects

Code:

ActiveObjects -

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

public class ActiveObjects : MonoBehaviour
{
    // Start is called before the first frame update
    void Start ()
    {
```

```
gameObject.SetActive(false);  
}
```

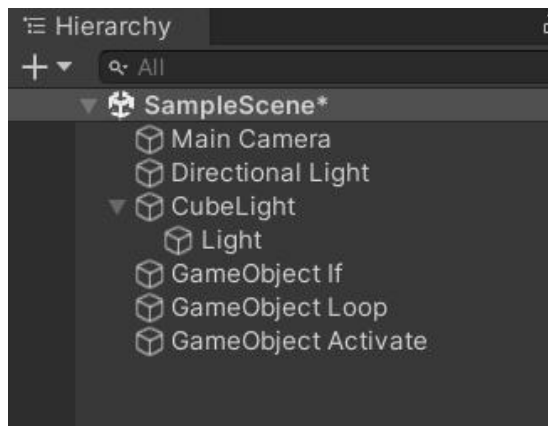
```
// Update is called once per frame  
void Update()  
{  
  
}  
}
```

CheckState -

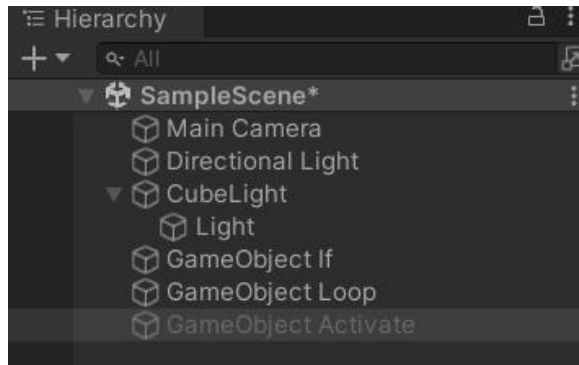
```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
public class CheckState : MonoBehaviour  
{  
    public GameObject myObject;  
  
    void Start ()  
    {  
        Debug.Log("Active Self: " + myObject.activeSelf);  
        Debug.Log("Active in Hierarchy" + myObject.activeInHierarchy);  
    }  
}
```

Output:

Before running script



After running script



Translate & Rotate

Code :

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

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

```
    // Start is called before the first frame update
    void Start()
    {
    }
}
```

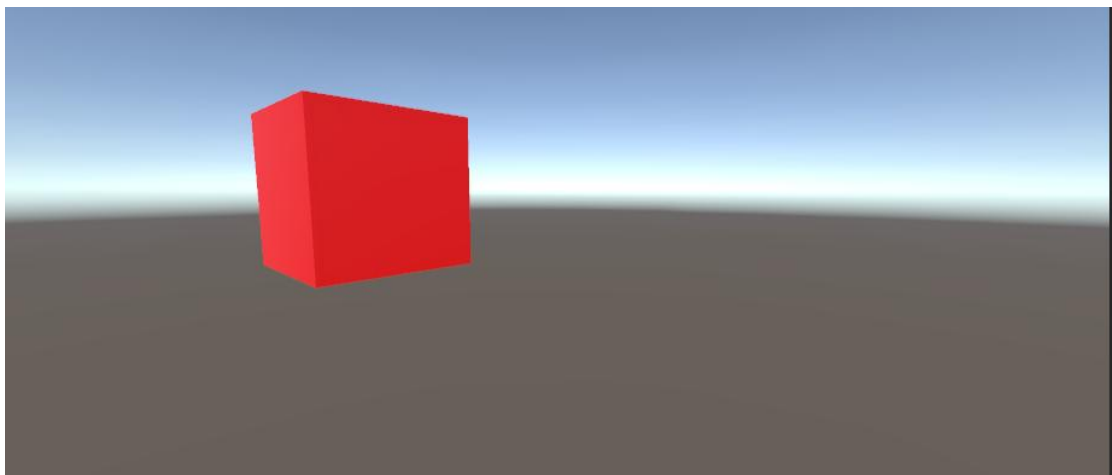
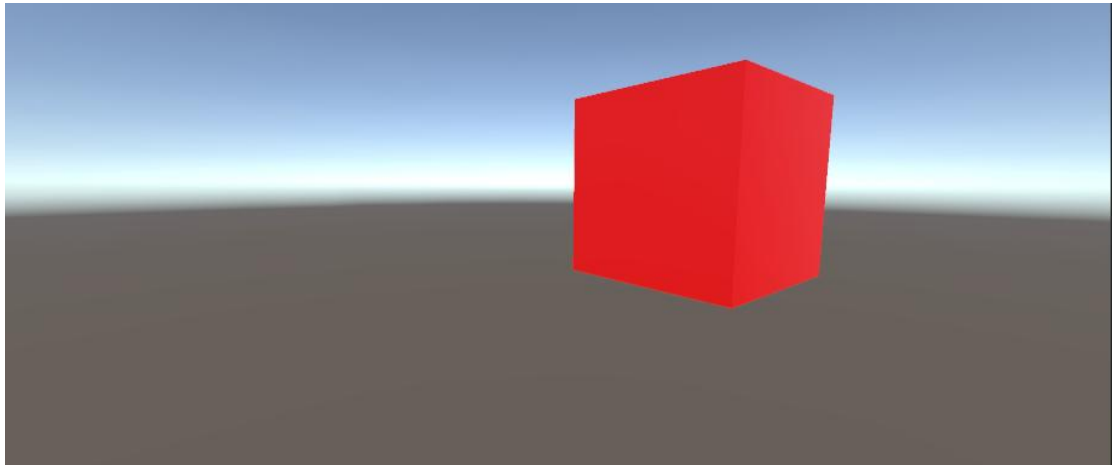
```
    // Update is called once per frame
    void Update()
    {
        if(Input.GetKey(KeyCode.UpArrow))
            transform.Translate(Vector3.forward * moveSpeed * Time.deltaTime);

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

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

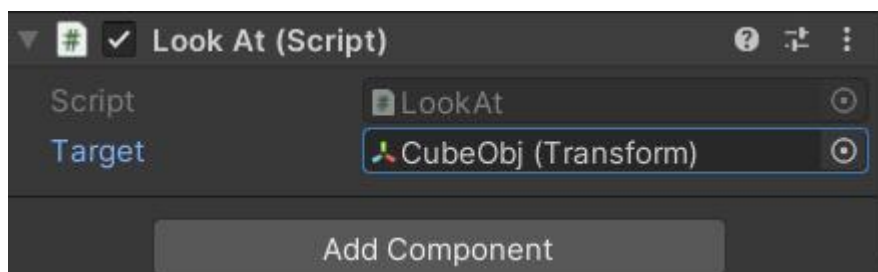
        if(Input.GetKey(KeyCode.RightArrow))
            transform.Rotate(Vector3.up, turnSpeed * Time.deltaTime);
    }
}
```

Output:



LookAt

This script can be applied to the camera to look at a particular target object. In our case it is the Cube Object.



Code:

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

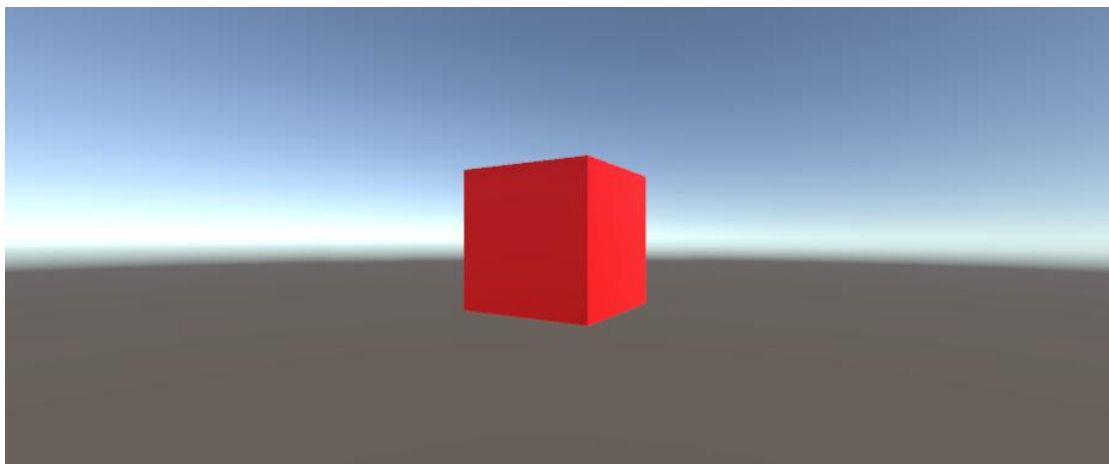
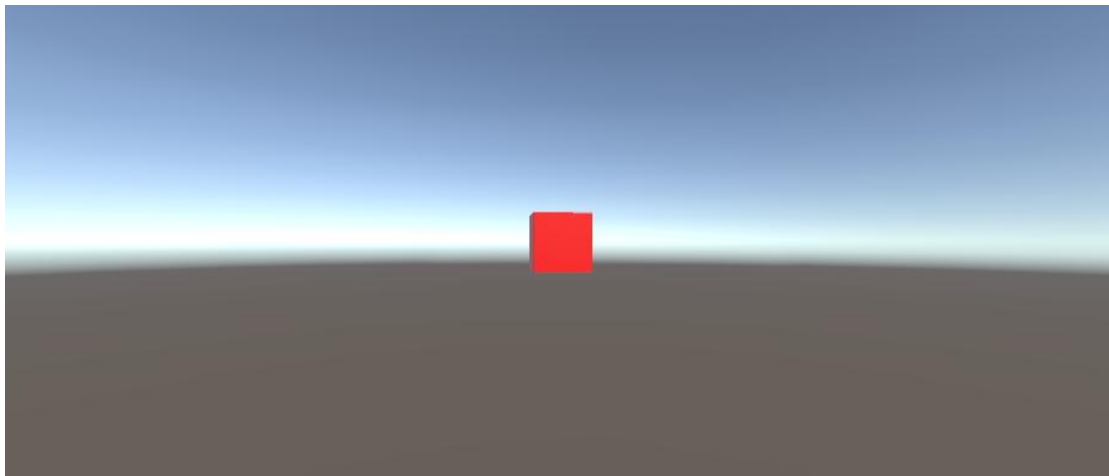
public class LookAt : MonoBehaviour
{
    public Transform target;
```

```
    // Start is called before the first frame update
    void Start()
    {

    }
}
```

```
    // Update is called once per frame
    void Update()
    {
        transform.LookAt(target);
    }
}
```

Output:



As the camera is now focused and is looking at the cube object, it can only move backward and forward and not about x-axis.

Game Video Links :

Time.DeltaTime

https://drive.google.com/file/d/13nwRxJaQrqlaRxz3KoovulqtKlhJG_p/view?usp=share_link

Update and Fixed Update

https://drive.google.com/file/d/1-OBhluGheyaaRJe-ITWhzsoohgGW6cre/view?usp=share_link

Enable

https://drive.google.com/file/d/1FDipT_ya_iJOtmCWOM8vCEn-73vbrPr/view?usp=share_link

Activate

https://drive.google.com/file/d/112DHg7sdtNXaqXzctDbSXTlat4f2B6EQ/view?usp=share_link