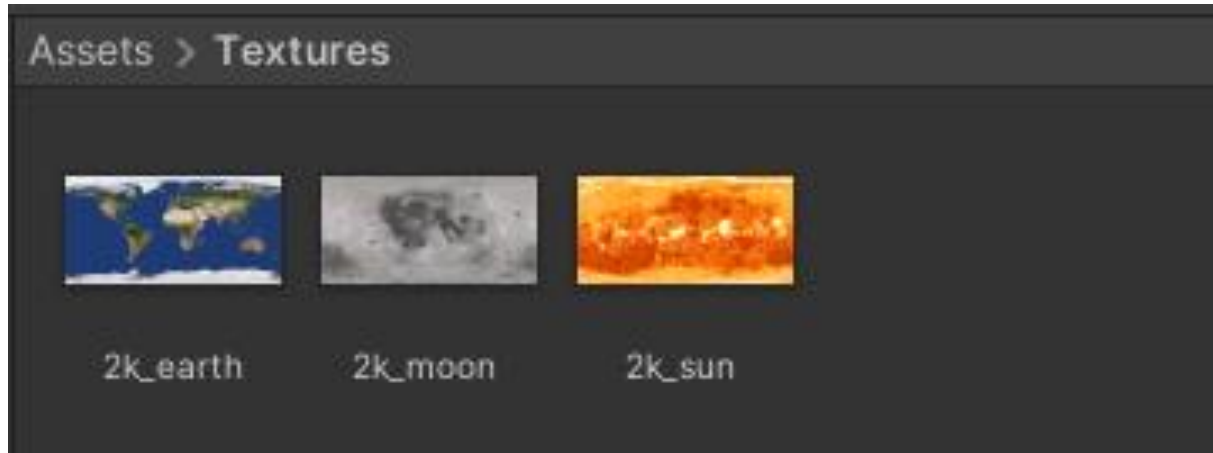


Computer Graphics

- [Unity] Rotation & Revolution

Sung Soo Hwang

Practice: Rotation & Revolution

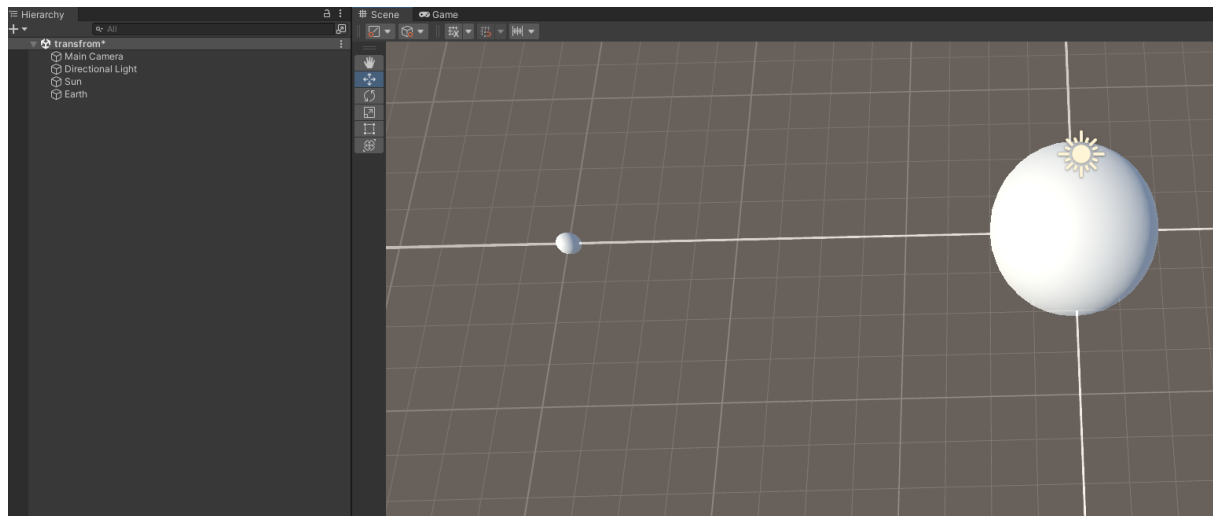


- Files to download
 - Texture images for Sun, Earth and Moon

<https://www.solarsystemscope.com/textures/>

Practice: Rotation & Revolution

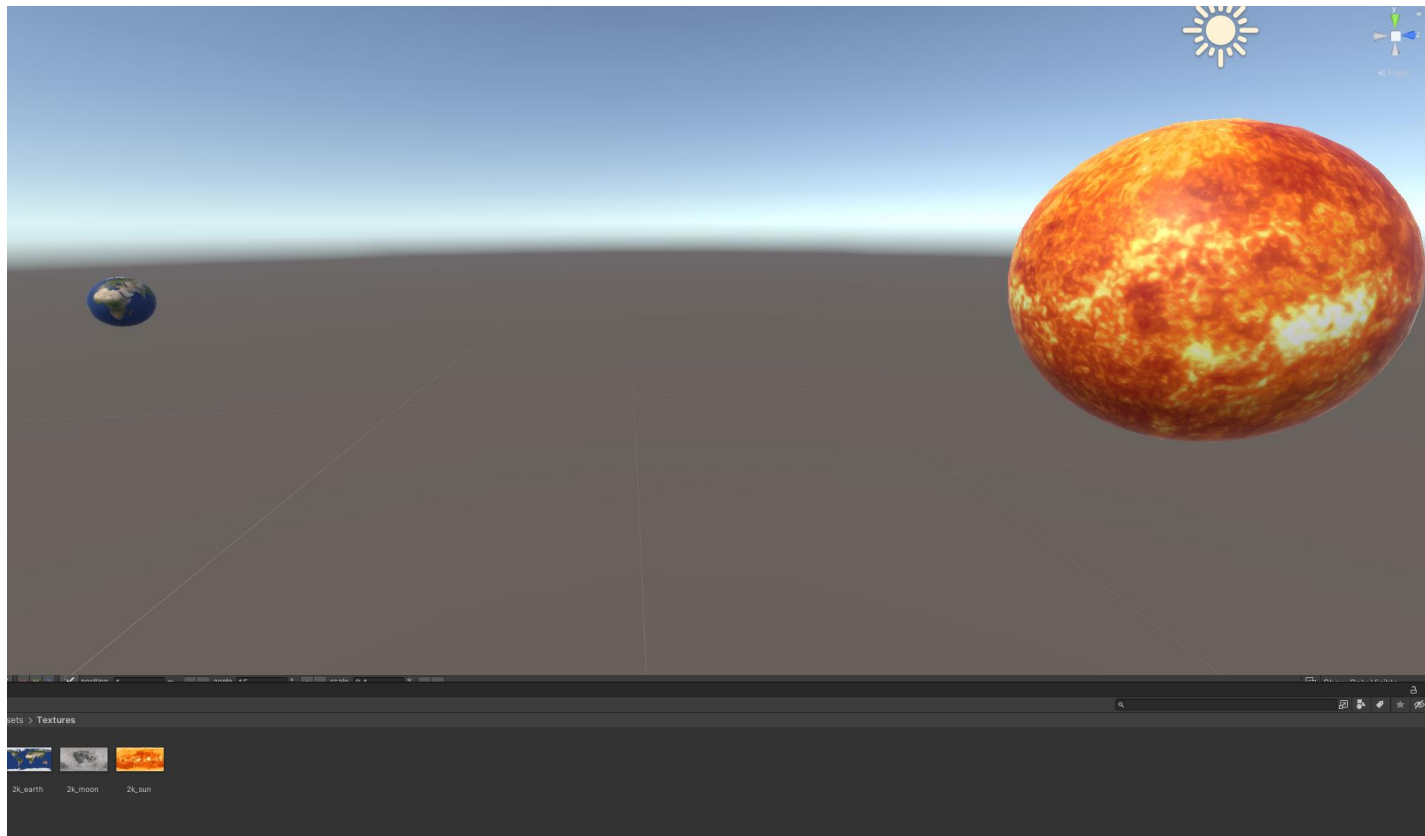
- Make spheres and change their names and sizes according to the table on the right. Place the sun at the origin and align the other sphere along the negative z direction.



Sphere Name	Scale
Sun	4.0
Earth	0.5

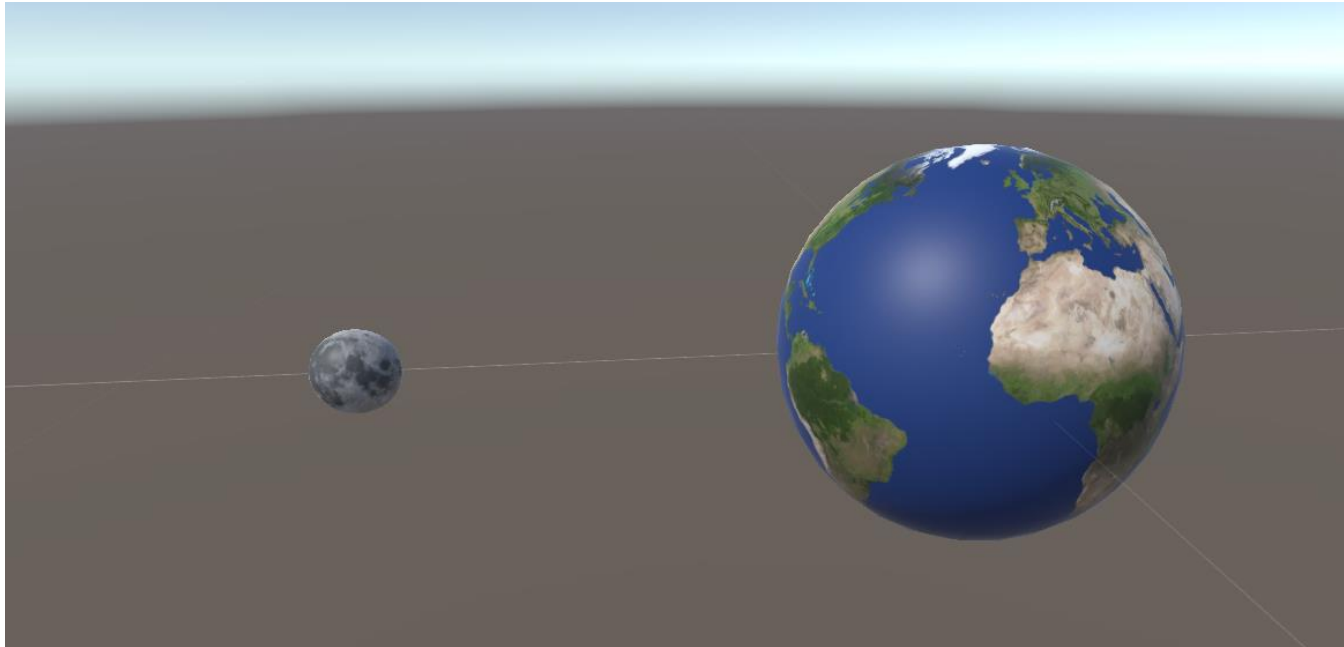
Practice: Rotation & Revolution

- Assigning textures
 - Make "Textures" folder in the project and import all texture files into that folder. Then, assign textures to their respective spheres.



Practice: Rotation & Revolution

- Making satellites
 - Make a small sphere and rename it, Moon. Assign the moon's texture to it and place it around planet Earth.



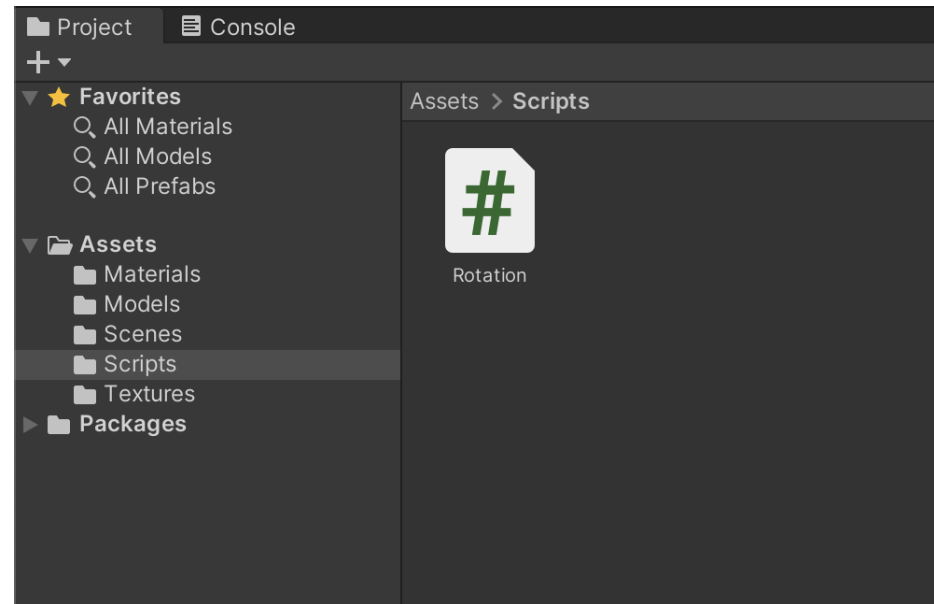
Practice: Rotation & Revolution

- Constructing a solar system hierarchy
 - Construct a hierarchy of game objects as shown below:



Practice: Rotation & Revolution

- Make a new folder and rename it Scripts. Create a script file, Rotation.cs, in the Scripts folder. The script will be used to produce the self-rotation and orbital revolution of the objects in the solar system



Practice: Rotation & Revolution

- Open “Rotation.cs”

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Rotation : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11      }
12
13      // Update is called once per frame
14      void Update()
15      {
16
17      }
18  }
19
```

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

Update is called every frame.

Practice: Rotation & Revolution



■ Whole Code

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

public class Rotation : MonoBehaviour
{
    public float obital_speed;
    public float selrot_speed;
    private Transform parent;

    // Start is called before the first frame update
    void Start()
    {
        obital_speed = Random.Range(1.0f, 50.0f);
        selrot_speed = Random.Range(1.0f, 50.0f);
        parent = transform.parent;
    }

    // Update is called once per frame
    void Update()
    {
        transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

        if(parent)
        {
            transform.RotateAround(parent.transform.position, Vector3.up, obital_speed * Time.deltaTime);
        }
    }
}
```

Practice: Rotation & Revolution

- What the script actually does?

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

Important necessary modules
for the program

```
public class Rotation : MonoBehaviour
```

```
{
```

```
    public float orbital_speed;
```

```
    public float selrot_speed;
```

```
    private Transform parent;
```

A member variable for orbital speed and
self-rotational speed of the current object
to which the script was added.

```
    // Start is called before the first frame update
```

```
    void Start()
```

```
    {
```

```
        orbital_speed = Random.Range(1.0f, 50.0f);
```

```
        selrot_speed = Random.Range(1.0f, 50.0f);
```

```
        parent = transform.parent;
```

```
    }
```

```
    // Update is called once per frame
```

```
    void Update()
```

```
    {
```

```
        transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);
```

```
        if(parent)
```

```
        {
```

```
            transform.RotateAround(parent.transform.position, Vector3.up, orbital_speed * Time.deltaTime);
```

```
        }
```

```
    }
```

```
}
```

Practice: Rotation & Revolution

- What the script actually does?

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

public class Rotation : MonoBehaviour
{
    public float orbital_speed;
    public float selrot_speed;
    private Transform parent;

    // Start is called before the first frame update
    void Start()
    {
        orbital_speed = Random.Range(1.0f, 50.0f);
        selrot_speed = Random.Range(1.0f, 50.0f);
        parent = transform.parent;
    }

    // Update is called once per frame
    void Update()
    {
        transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

        if(parent)
        {
            transform.RotateAround(parent.transform.position, Vector3.up, orbital_speed * Time.deltaTime);
        }
    }
}
```

Member variable to store the transformation of the parent object.

Practice: Rotation & Revolution

- What the script actually does?

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

public class Rotation : MonoBehaviour
{
    public float orbital_speed;
    public float selrot_speed;
    private Transform parent;

    // Start is called before the first frame update
    void Start()
    {
        orbital_speed = Random.Range(1.0f, 50.0f);
        selrot_speed = Random.Range(1.0f, 50.0f);
        parent = transform.parent;
    }

    // Update is called once per frame
    void Update()
    {
        transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

        if(parent)
        {
            transform.RotateAround(parent.transform.position, Vector3.up, orbital_speed * Time.deltaTime);
        }
    }
}
```

Randomly assign the speed of revolution and rotation. And get the transformation of the parent object

Practice: Rotation & Revolution

- What the script actually does?

```
// Update is called once per frame
void Update()
{
    transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

    if(parent)
    {
        transform.RotateAround(parent.transform.position, Vector3.up, obital_speed * Time.deltaTime);
    }
}
```

It will generate the self-rotation.

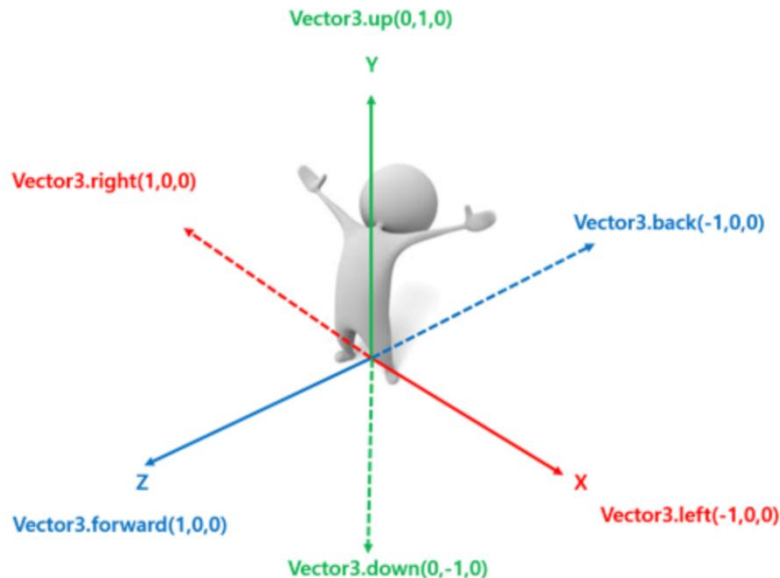


Practice: Rotation & Revolution

- What the script actually does?

```
// Update is called once per frame
void Update()
{
    // Rotational axis    Angle to rotate for the elapsed time
    transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

    if(parent)
    {
        transform.RotateAround(parent.transform.position, Vector3.up, orbital_speed * Time.deltaTime);
    }
}
```

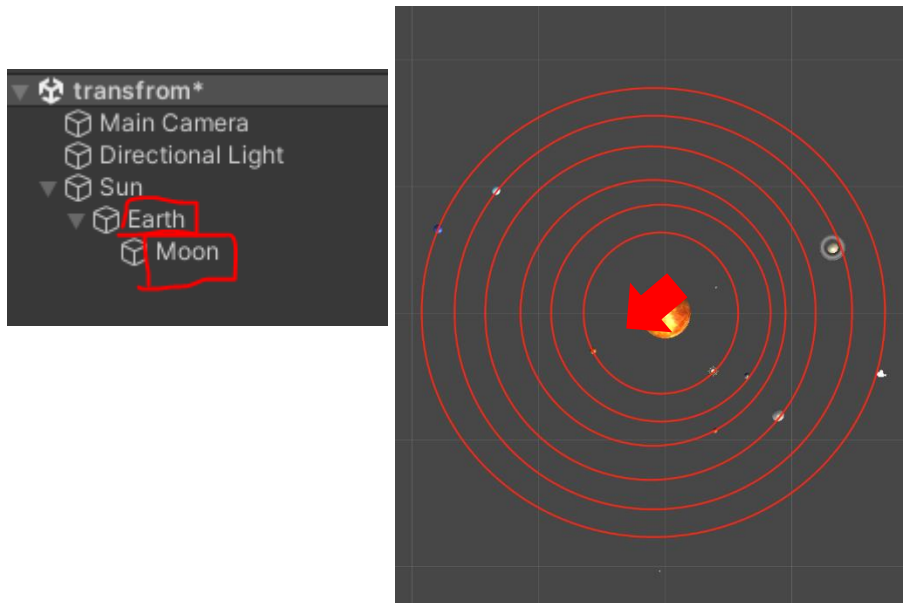


Practice: Rotation & Revolution

- What the script actually does?

```
// Update is called once per frame
void Update()
{
    transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);

    if(parent)    If the current object has its parent object (e.g., sun), it rotates around
    {              the parent object.,
        transform.RotateAround(parent.transform.position, Vector3.up, obital_speed * Time.deltaTime);
    }
}
```



Practice: Rotation & Revolution

- What the script actually does?

```
// Update is called once per frame
```

```
void Update()
```

```
{
```

```
    transform.Rotate(Vector3.up, selrot_speed * Time.deltaTime);
```

```
    if(parent)
```

```
    {
```

```
        transform.RotateAround(parent.transform.position, Vector3.up, orbital speed * Time.deltaTime);
```

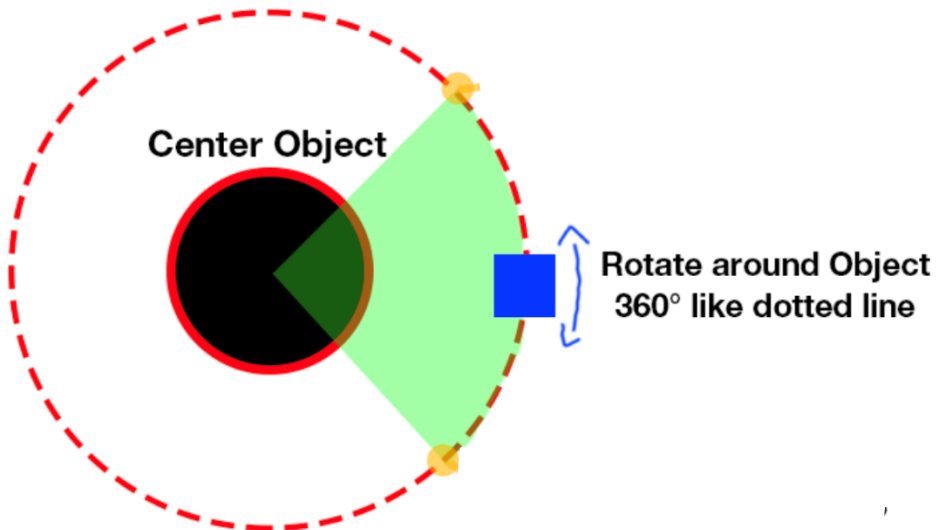
```
    }
```

```
}
```

Center Object for the rotation

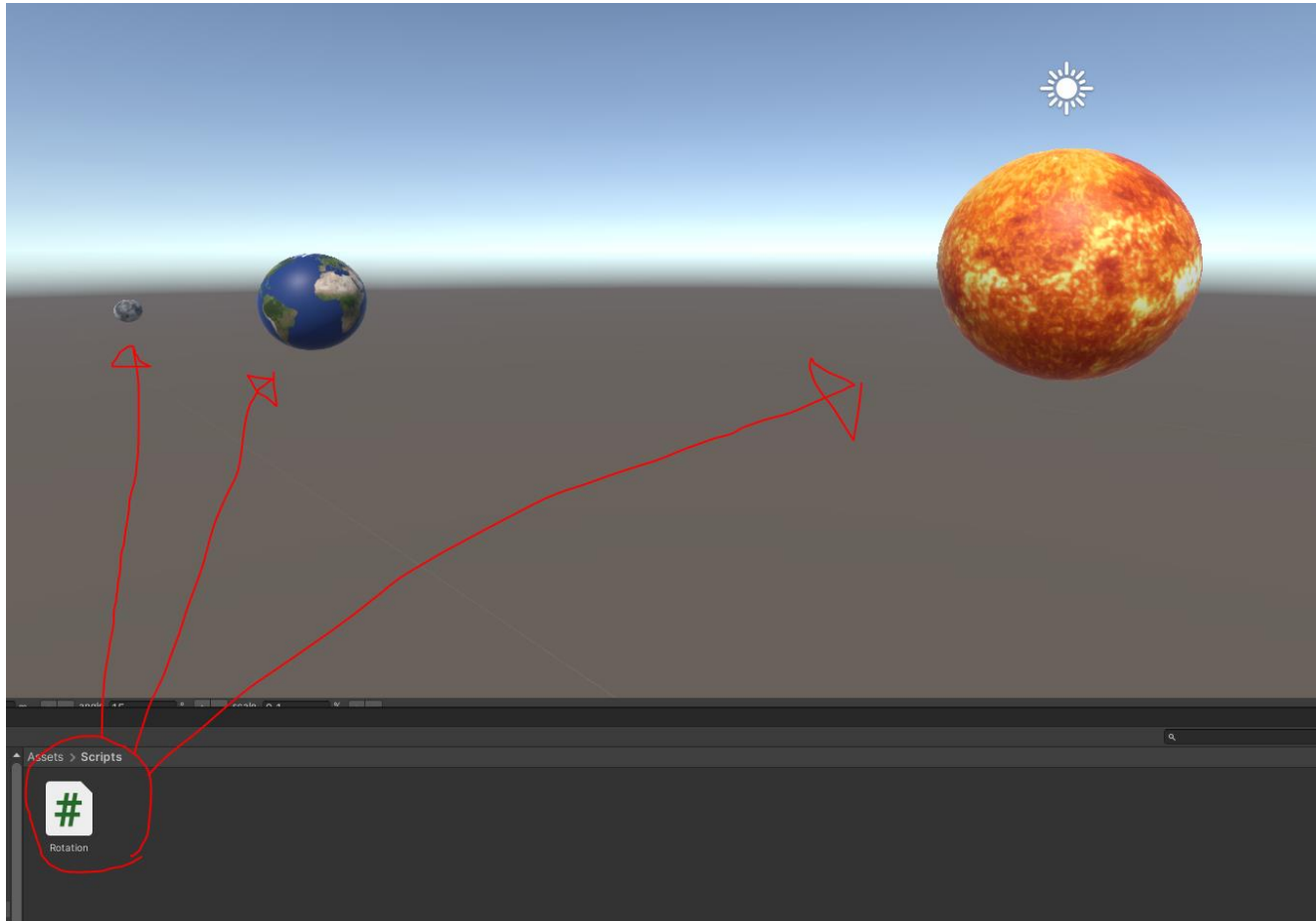
Angle to rotate for the elapsed time

Rotational axis



Practice: Rotation & Revolution

- Drag & Drop the given script file.



Practice: Rotation & Revolution

- Rendering result

