Basic Scripting

CS 3540 – Game Programming

1

Graphics Rendering in Unity

2

Unity Graphics Rendering

Meshes

Materials

Shaders

Textures

3

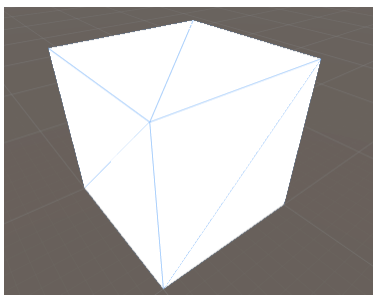
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

Shaders

Textures



4

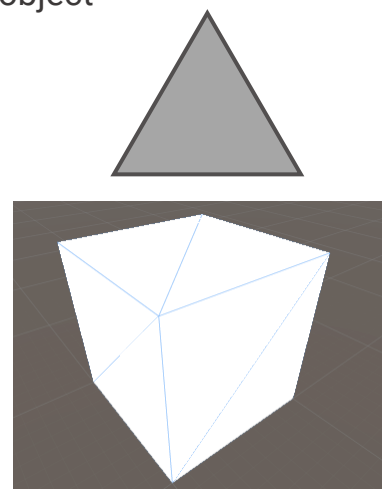
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

Shaders

Textures



5

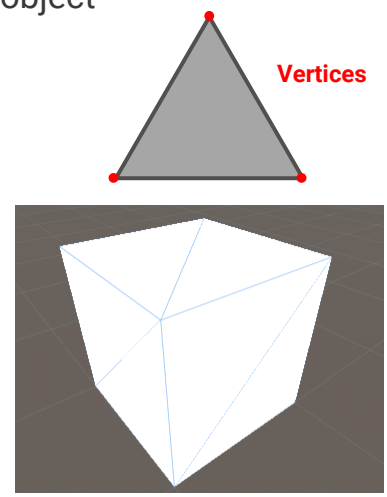
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

Shaders

Textures



6

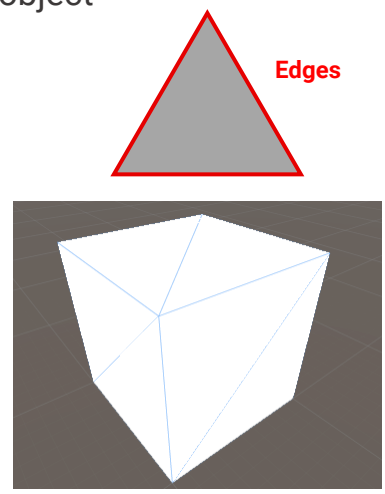
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

Shaders

Textures



7

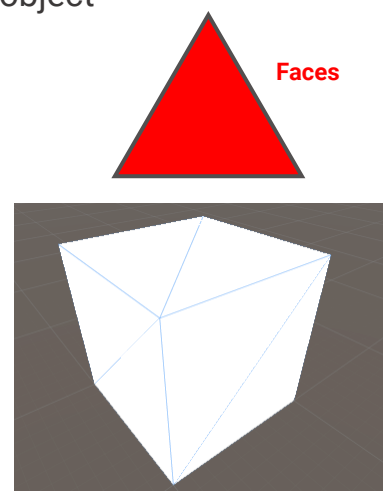
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

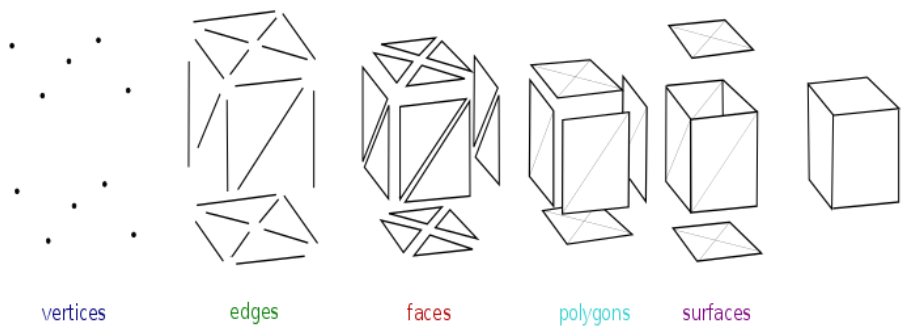
Shaders

Textures



8

Meshes



9

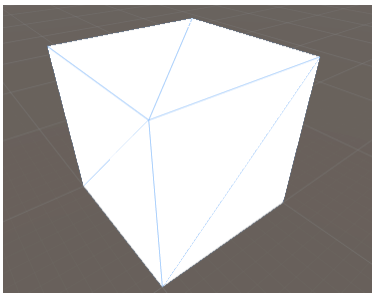
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials

Shaders

Textures



10

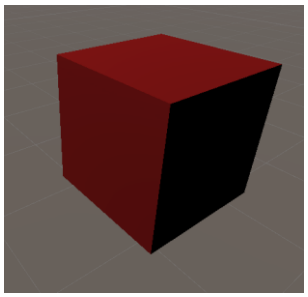
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials
Define how a surface should be rendered

Shaders

Textures



11

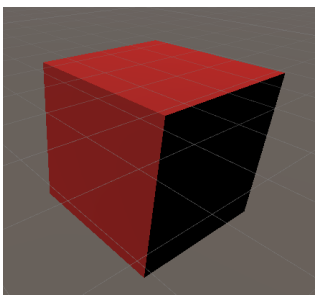
Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials
Define how a surface should be rendered

Shaders
Small scripts that do the lighting calculations for you

Textures



12

Unity Graphics Rendering

Meshes
Main graphics primitive and define the shape of an object

Materials
Define how a surface should be rendered

Shaders
Small scripts that do the lighting calculations for you

Textures
Bitmap images applied over the mesh surface

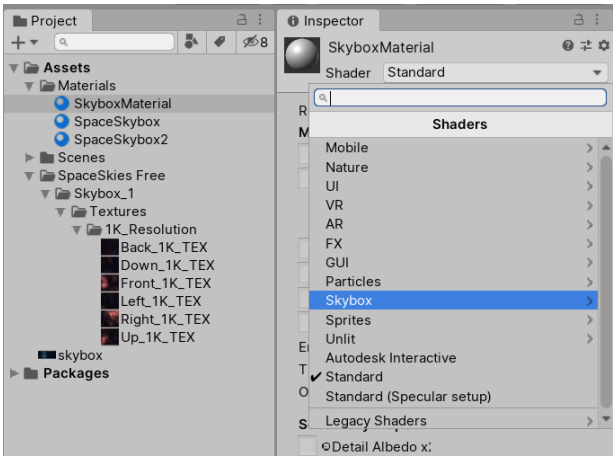
13

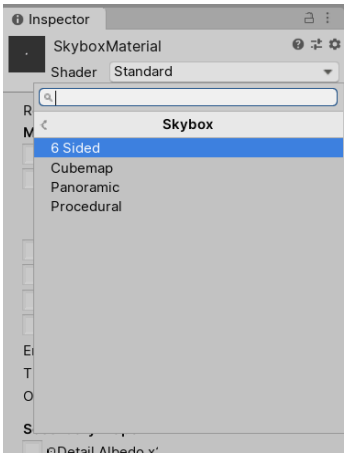
Making a Skybox

14

Making a Skybox

Change the Material's Shader to Skybox > 6 Sided

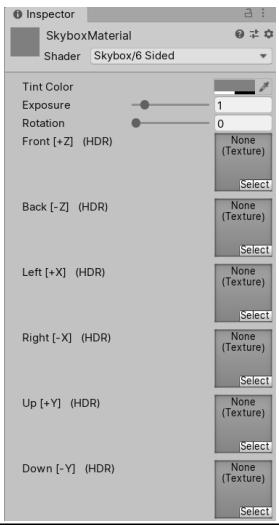




15

Making a Skybox

Specify the textures to be used for each side

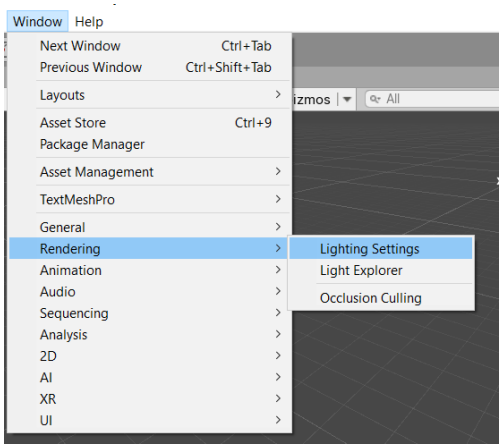


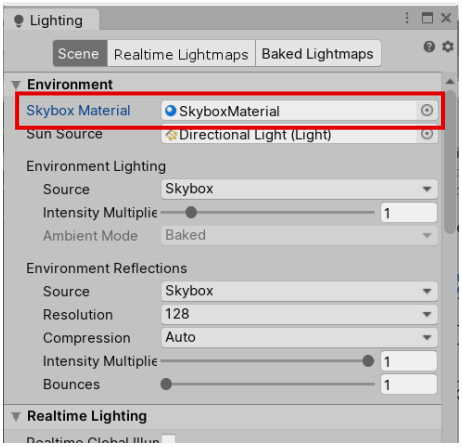
16

Making a Skybox

Need to apply the new material to the default Skybox in Unity

Window > Rendering > Lighting Settings

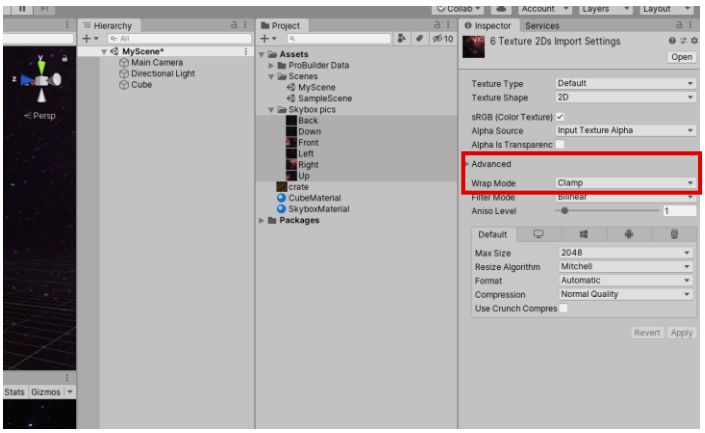




17

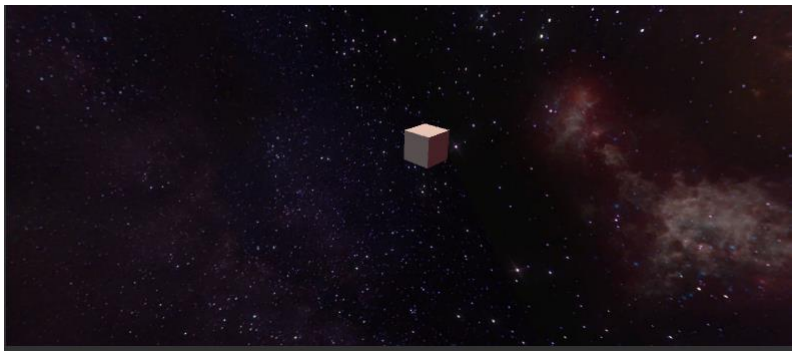
Making a Skybox

Need to change Wrap Mode to Clamp for all textures



18

Making a Skybox



19

Scripting in Unity

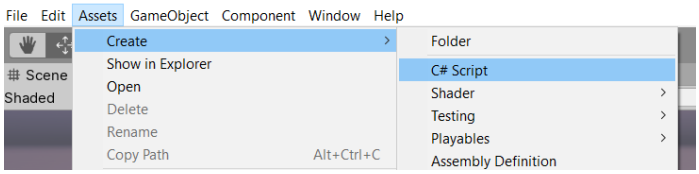
20

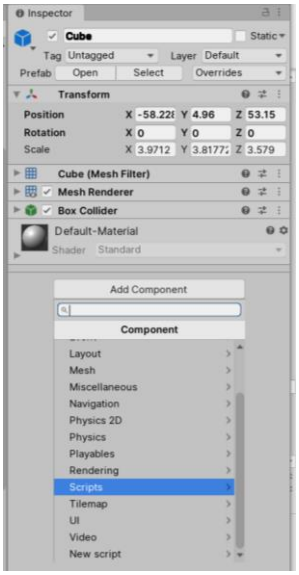
Unity 3D: Scripts

Scripts are Components too!

Allow you to write your own Components

Enable you to customize GameObjects

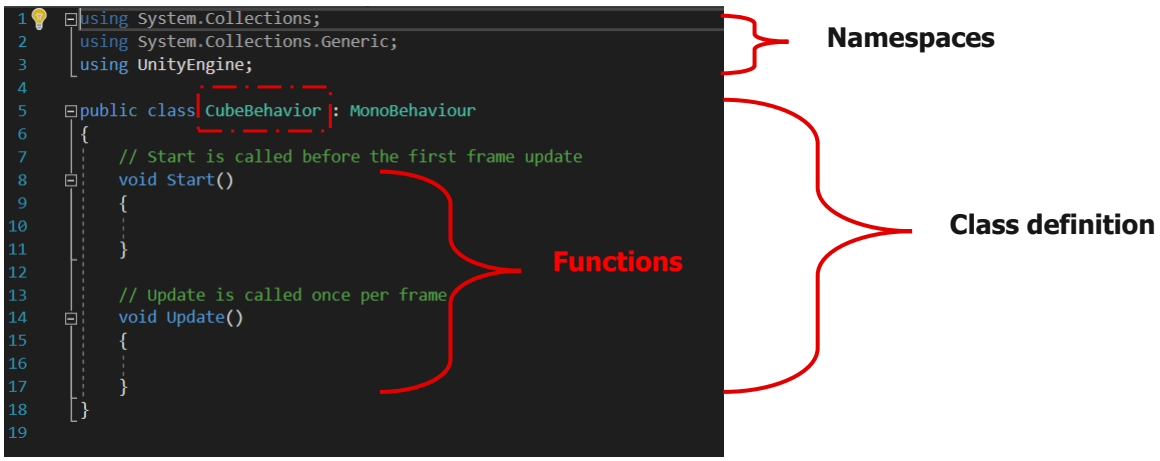




21

Anatomy of a Script File

All scripts, and thus classes, derive from the built-in class MonoBehaviour



22

Start() Function

Called only once on the lifetime of the object

Used for initializing game parameters

Called before the Update() function

23

Update Function

Called every frame

Used for regular updates:
- Moving non-physics objects
- Receiving input
- Simple timers

Update interval times vary

Delete Update() functions from scripts where they are not used

Performance issues

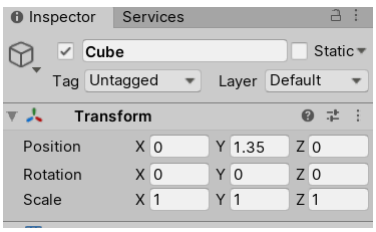
24

C# Primer

25

Rotate Method

Rotates GameObjects in a variety of ways



Can rotate around x, y, and z axes

To rotate the current GameObject

transform.Rotate(x_angle, y_angle, z_angle);

```
void Update()
{

    transform.Rotate(0, -5, 0);

}
```

26

Variables

Public variables are accessible by other classes and code

```
public float someFloat = 1f;
private int someInt = 1;

public string someString = "HelloCube";
public bool someBool = false;

public GameObject someObject;
public Transform someTransform;
public Vector3 someVector = new Vector3();
public Camera someCamera;
```

Private variables can only be accessed within the class they are declared in

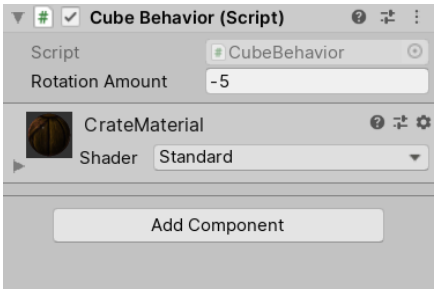
27

Variables in the Inspector

Public variables are accessible from the Inspector

```
public class CubeBehavior : MonoBehaviour
{
    // Start is called before the first frame update

    public float rotationAmount = -5f;
```



28

Basic Keyboard Input

29

Keyboard Input: Input.GetKey

GetKey:

Returns true while the user holds down the key identified by name.

GetKeyUp:

Returns true during the frame the user releases the key

GetKeyDown:

Returns true during the frame the user starts pressing down the key

30

C# Selection Statements

```
if(condition)
{
    //do something
}
```

```
if(condition)
{
    //do something
    //executes when condition is true
}
else
{
    //do something else
    //executes when condition is false
}
```

```
if(condition1)
{
    //do something
    //executes when condition1 is true
}
else if(condition2)
{
    //do another thing
    //executes when condition2 is true
}
else
{
    //do something else
    //executes when none of the conditions is true
}
```

31

Conditional AND (&&) Operator

The **&&** (conditional AND) operator works as follows:

```
if(condition1 && condition2)
{
    //do something
    //executes when both condition1 and condition2 are true
}
```

The combined condition is true if and only if *both* simple conditions are true.

32

Conditional OR (||) Operator

The || (conditional OR) operator works as follows

```
if(condition1 || condition2)
{
    //do something

    //executes when at least
    //one of the conditions is true
}
```

The complex condition evaluates to `true` if either or both of the simple conditions are true

33

Keyboard Input: GetKey

Input.GetKeyDown(Keycode.Space)

Returns true when the user starts pressing down the Space key

```
if(Input.GetKeyDown(KeyCode.Space))
{
    transform.Rotate(0, -5, 0);

}
```

34

Translate Method

Moves the GameObject (transform) in a variety of ways

Can move along x, y, and z axes

To move the current GameObject

transform.Translate(x_amount, y_amount, z_amount);

35

OnMouseDown

Built-in function used as an event trigger

Called when the user has pressed the mouse button while over a GameObject or GUI element

```
private void OnMouseDown()
{
    Debug.Log("GameObject Clicked!");
}
```

36

Destroy

Removes a GameObject, component, or asset

Destroy(gameObject);

gameObject is a reference to the current GameObject to which the script component is attached

Can specify time delay

Destroy(gameObject, 3f);

```
private void OnMouseDown()
{
    Destroy(gameObject);
}
```

37

SetActive

Activates/deactivates the GameObject

Based on the boolean parameter passed in, true or false

gameObject.SetActive(false);

The GameObject is still in Hierarchy, but is disabled in the scene

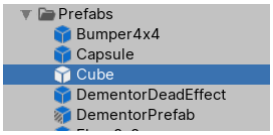
38

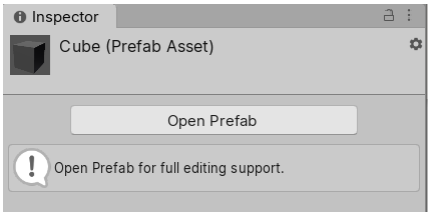
Unity 3D: Prefabs

Allows you to create, configure, and store a GameObject with all its components

Can be thought of as templates

Can use multiple instances of a prefab in your scene





39

Readings

1. https://docs.unity3d.com/Manual/Components.html
2. https://docs.unity3d.com/Manual/Materials.html
3. https://docs.unity3d.com/Manual/ScriptingSection.html
4. https://docs.unity3d.com/Manual/Prefabs.html
5. https://docs.unity3d.com/Manual/HOWTO-UseSkybox.html
6. https://docs.unity3d.com/ScriptReference/MonoBehaviour.Start.html
7. https://docs.unity3d.com/ScriptReference/MonoBehaviour.Update.html
8. https://docs.unity3d.com/ScriptReference/Transform.Rotate.html
9. https://docs.unity3d.com/ScriptReference/Input.GetKey.html
10. https://docs.unity3d.com/ScriptReference/KeyCode.html
11. https://docs.unity3d.com/ScriptReference/Transform.Translate.html
12. https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnMouseDown.html
13. https://docs.unity3d.com/ScriptReference/Object.Destroy.html
14. https://docs.unity3d.com/ScriptReference/GameObject.SetActive.html

40