



WHAT IS UNITY?

Game engine – system designed to help create video games

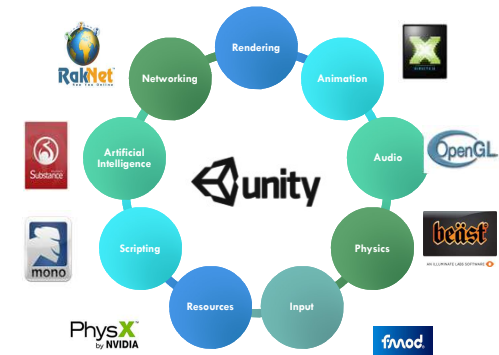
- Easier & Faster

Visual editor – see changes in real-time

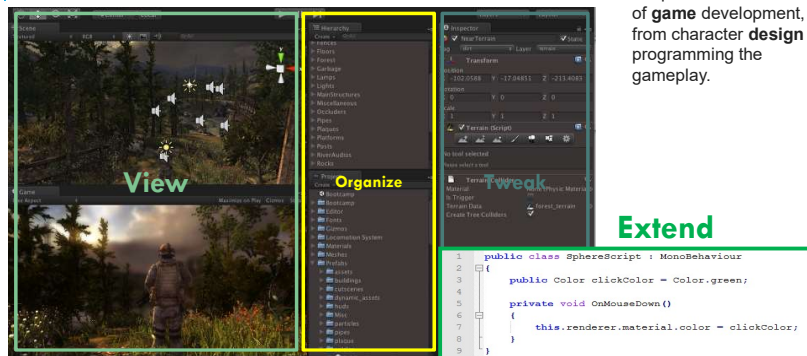
- Interactive & Rapid prototyping

Component-based – functionality built out of smaller pieces

- Modular & Extensible



WHAT DOES UNITY LOOK LIKE?



Unity has standardized the process of **game development**, from character **design** to programming the gameplay.

UNITY GAMES RUN EVERYWHERE



Unity understands you



MULTIPLE PROGRAMMING LANGUAGES

JavaScript

```
var explosion : Transform;

function OnCollisionEnter() {
    Destroy(gameObject);
    Instantiate(explosion, transform.position, transform.rotation);
}
```

C#

```
using UnityEngine;
using System.Collections;

public class Example : MonoBehaviour {
    public Transform explosion;

    void OnCollisionEnter() {
        Destroy(gameObject);
        Instantiate(explosion, transform.position, transform.rotation);
    }
}
```



Boo

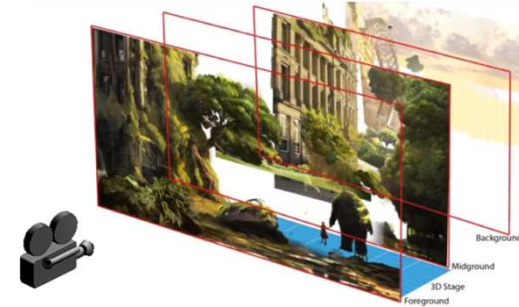
```
import UnityEngine
import System.Collections

class Example(MonoBehaviour):

    public explosion as Transform

    def OnCollisionEnter():
        Destroy(gameObject)
        Instantiate(explosion, transform.position, transform.rotation)
```

WHAT ABOUT 2D GAMES?



REALM

<http://www.therealmgame.com/>

WHAT ABOUT 2D GAMES?



REALM

<http://www.therealmgame.com/>

GAMES CREATED WITH UNITY



Beat Sneak Bandit



Bad Piggies



The Room



Temple Run 2



Scrolls



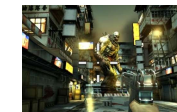
Roachard



Thomas was Alone



CSR Racing



Dead Trigger 2

HOW TO GET UNITY?

Unity Basic

Has every essential features such as graphics, audio, physics, animation, networking, input, and scripting

Free (with splashscreen)

Unity Pro

Advanced graphics, audio, animation, and artificial Intelligence

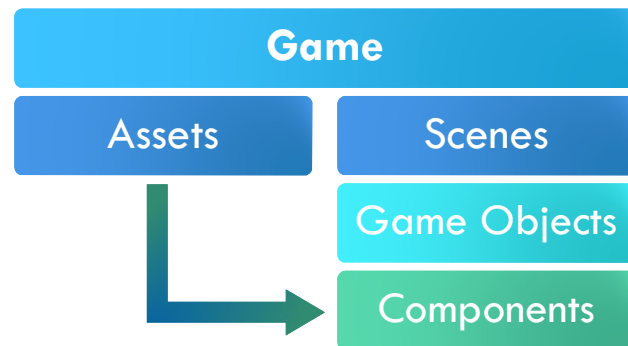
\$1,500+

Download from <http://unity3d.com>

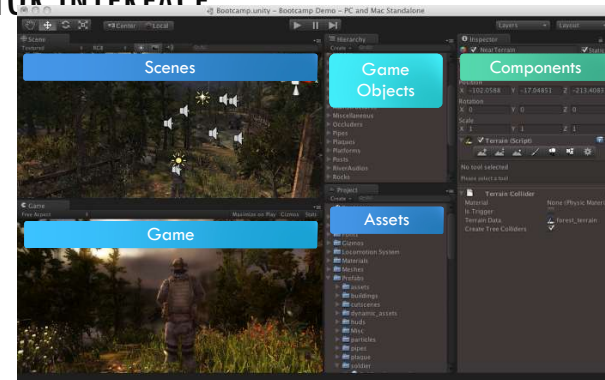
CONCEPTS AND WORKFLOW

Part 2

ARCHITECTURE



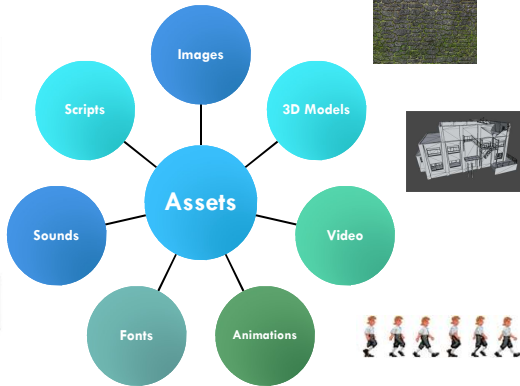
EDITOR INTERFACE



ASSETS

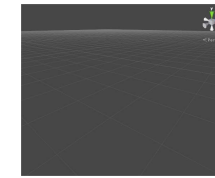
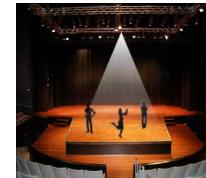
```
public class Spawncube : MonoBehaviour
{
    public Color color = Color.green;

    private void OnMouseDown()
    {
        this.renderer.material.color = color;
    }
}
```



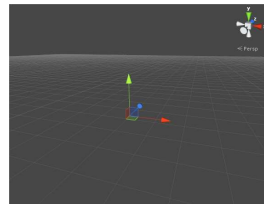
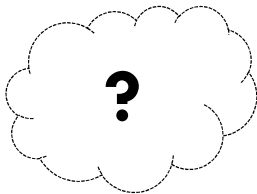
SCENE

- Unity games are divided into **scenes**
- **Scenes** are empty spaces...
- ...that can be filled with **game objects**



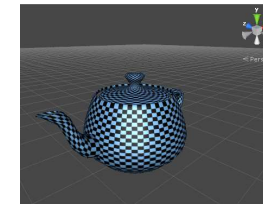
GAME OBJECT

- Everything inside a **scene** is a **game object**
- **Game objects** also start out empty and do nothing...



GAME OBJECT

- ...but by adding **components** to them they can become anything!



COMPONENT

Each **component** adds a piece of functionality to the **game object**

The combination of all **components** defines what the **game object** is

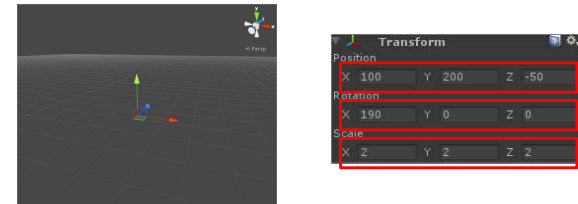
Let's see some examples!

THE TRANSFORM COMPONENT

Where?

Which direction?

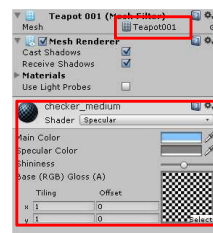
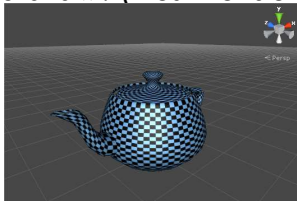
How large?



RENDERING COMPONENTS

What to draw? (mesh filter)

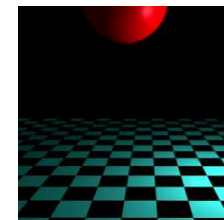
How to draw? (mesh renderer)



PHYSICS COMPONENTS

Is solid? (collider)

Moves? (rigid body)



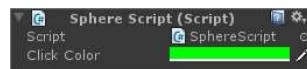
THE SCRIPT COMPONENT

Adds custom behavior

```

1 public class SphereScript : MonoBehaviour
2 {
3     public Color clickColor = Color.green;
4
5     private void OnMouseDown ()
6     {
7         this.renderer.material.color = clickColor;
8     }
9 }

```



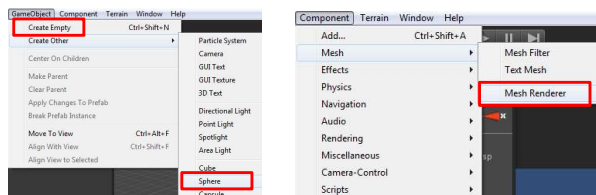
OTHER COMPONENTS

Light
Camera
Text mesh
Audio listener & source
Particle system
Skybox
...and many more.

HOW TO CREATE GAME OBJECTS

Create an empty **game object** and manually add **components** to it

Choose one of the default **game objects**



GAME

