Your First Program

Tutorial *1*

Dr. Osama Halabi

Project Setup

Following the standard project setup procedure, create a new Project in Unity. For information on the standard project setup procedure

- **Project name:** Hello World
- **Scene name:** (none yet)
- **C# Script names:** (none yet)

Before doing anything else, save your scene by choosing *File* > *Save Scene* from the menu bar. Unity will automatically choose the correct place to save the scene, so just name it *Scene_0* and *click Save.*

> ***Warning***
>
> **Never Change the Name of Your Project Folder While Unity Is Running** If you change the name of the project folder while Unity is running, it will crash in a very ungraceful way. Unity does a lot of file management in the background while it's running, and changing a folder name on it will almost always cause a crash. If you want to change your project folder name, quit Unity, change the folder name, and launch Unity again.

Making a New C# Script

1. Click the *Create* button in the Project pane and choose *Create* > *C# Script*
2. Name this script HelloWorld

3. Double-click the name or the icon of the HelloWorld script to launch MonoDevelop, our C# editor.
4. Type two tabs and the code `print ("Hello World!");`
5. Now, save this script by choosing *File* > *Save* from the MonoDevelop menu bar and switch back to Unity.
6. Click and hold on the name of the HelloWorld script in the Project pane, drag it over on top of *Main Camera* in the scene Hierarchy pane.
7. Click the *Play* button

Start() Versus Update()

1. move the `print()` function call from `Start()` to `Update()`
2. You'll see that *Hello World!* is now printed many, many times in rapid succession

Start() is called once on the first frame that an object exists, whereas Update() is called every frame.

> **Tip**
>
> If you want to see each repeat of the same message only once, you can click the Collapse button of the Console pane and it will ensure that each different message text appears only once.

Making Things More Interesting

Now, we're going to add some Unity style to your first program. In this example, we're going to create many, many copies of a cube.

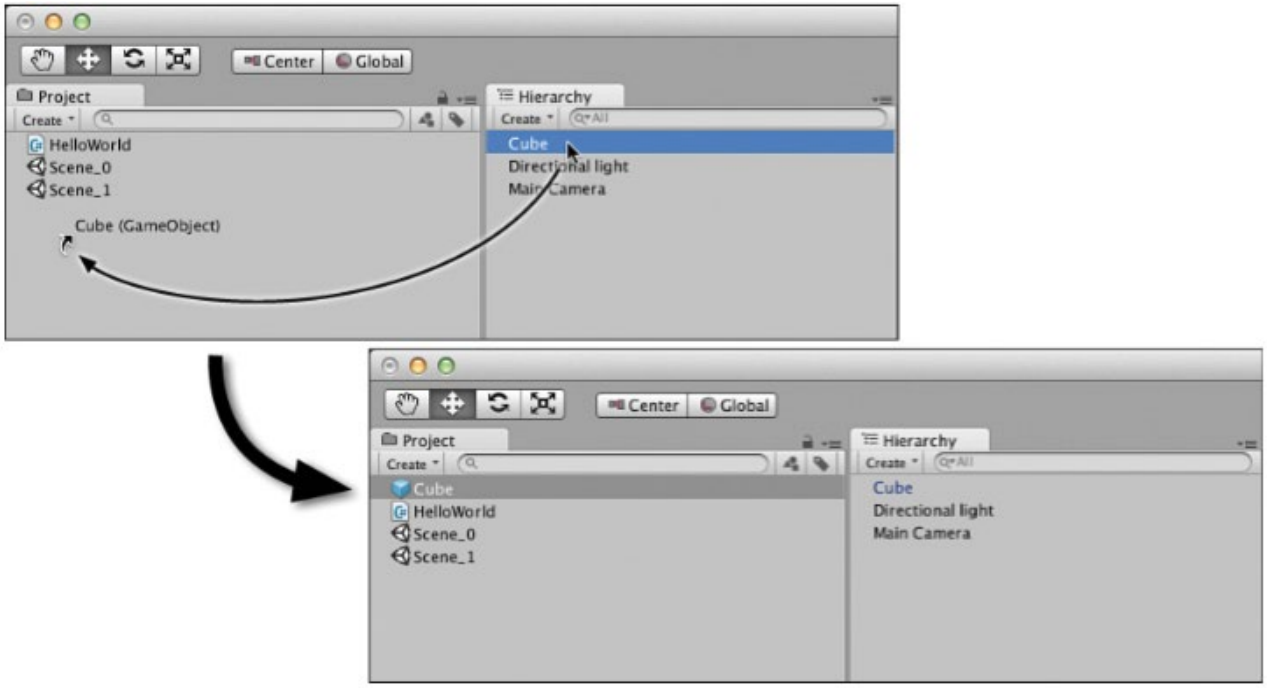
1. create game object cube.
2. add one more component to this GameObject: a *Rigidbody.*
3. Click play button, you'll see the box fall due to gravity

- 1 unit of distance = 1 meter (for example, the units for the position of a transform.
- 1 unit of mass = 1 kilogram (for example, the units of mass of a Rigidbody).
- The default gravity of –9.8 = 9.8 m/s^2 in the downward (negative y) direction.
- An average human character would be about 2 units (2 meters) tall.

Making a Prefab

A prefab is a reusable element in a project that can be instantiated (cloned into existence) any number of times. You can think of a prefab as a mold for a GameObject, and each GameObject made from that prefab is called an *instance* of the prefab (hence the word *instantiate*).

1. To make a prefab, click Cube in the Hierarchy pane, drag it over to the Project pane, and release the mouse button



2. We don't need the cube any more in the scene, so delete the cube in the Heirachcy pane (not the Project panel).

It's time to get our hands dirty with some more code.

3. Create a new script and rename it to ***CubeSpawner***
4. Open the MonoDevelop to exit the script
5. Add the boded code show here, and save it:

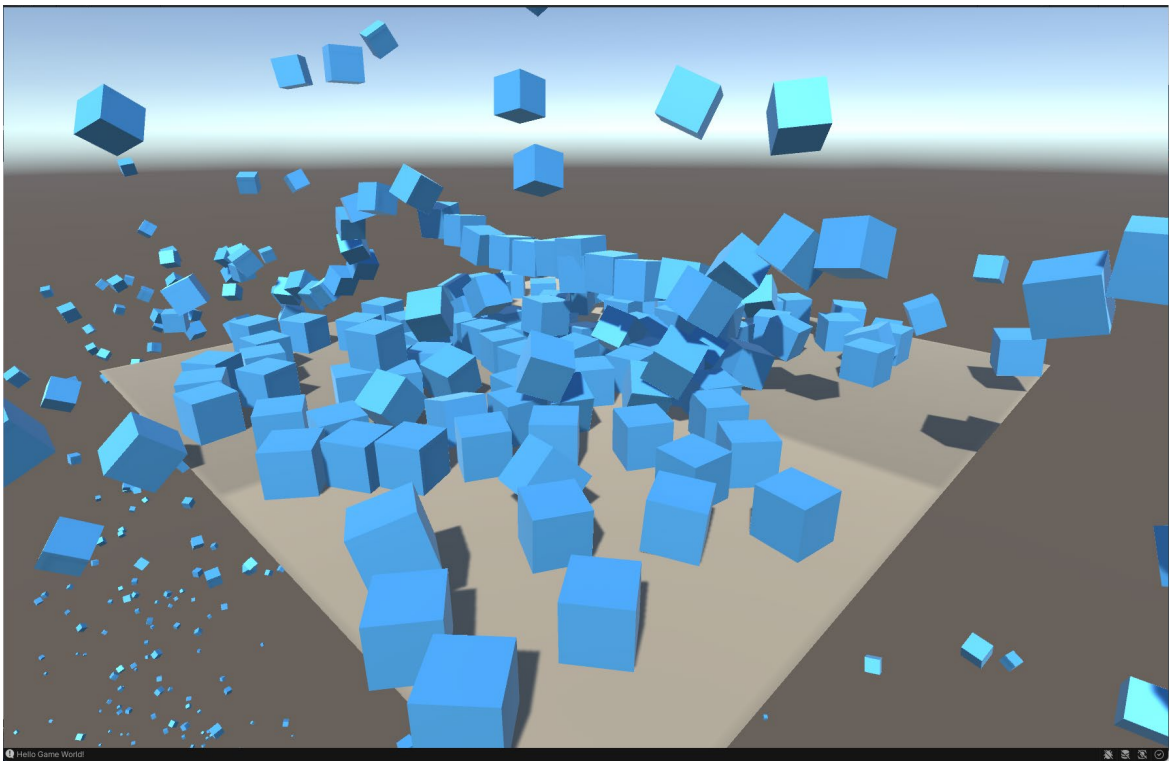
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class cubeSpawner : MonoBehaviour
{
    public GameObject cubePrefabVar;


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

```
        Instantiate(cubePrefabVar);
    }

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

    }
}
```

6. needs to be attached to something to run, so in Unity, drag the CubeSpawner script over to Main Camera.
7. Click the *Play* button.
8. You'll see that a single Cube Prefab (Clone) GameObject is instantiated in the Hierarchy.

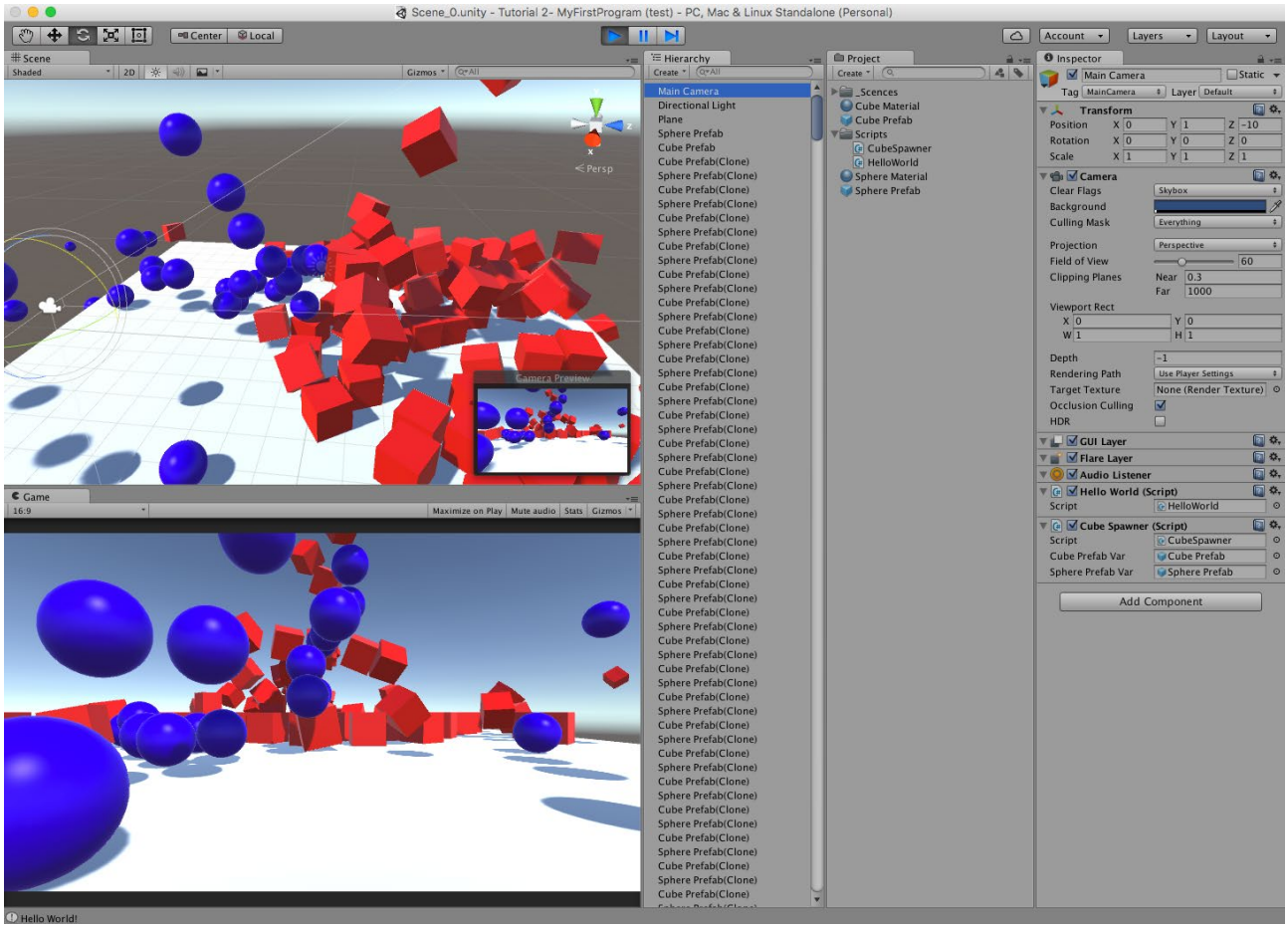
Making Things Even More Interesting

1. Comment out the Instantiate () call in the Start() function
2. Add **`Instantiate( cubePrefabVar );`** statement to **`Update ()`**
3. Click play and see the power of Unity!
4. Add plane game object as the ground.
5. Click the *Play* button.



Exercise

1. Modify the script so that you can achieve the following scene by creating another Ball prefab and generate instances of both prefabs.



2. The speed is very high, can you think of how you can modify the update frame rate to slow down the drawing of the prefabs using the Time.deltaTime to get the time of each frame and use it to draw in 0.25 sec instead of every frame.