CS248 Lecture 2

Introduction to Unity

January 11th, 2017

Course Logistics

- Piazza
- Staff Email: cs248-win1617-staff@lists.stanford.edu
- SCPD Grading via Google Hangouts: cs248.winter2017@gmail.com
- Homework 1 due Monday (January 16th)
  - If MLK day does not work for you, find a CA on January 17th.
  - Go to office hours OR schedule a time with another CA in advance.
- Can not make the grading session? Get in touch with the CA's.
- Office hours on website. Any changes will be announced on Piazza.

Overview

- Who Uses Unity?
- Getting Setup with Unity
- Unity Resources
- Unity Overview
- Unity Tutorial
- Live Demo
- Q&A

Who Uses Unity?

MORE THAN JUST
POOR INDIE
DEVELOPERS

Who Uses Unity?

Who Uses Unity?

Who Uses Unity?

Who Uses Unity?

Who Uses Unity?

Who Uses Unity?

- And more! https://en.wikipedia.org/wiki/List_of_Unity_games

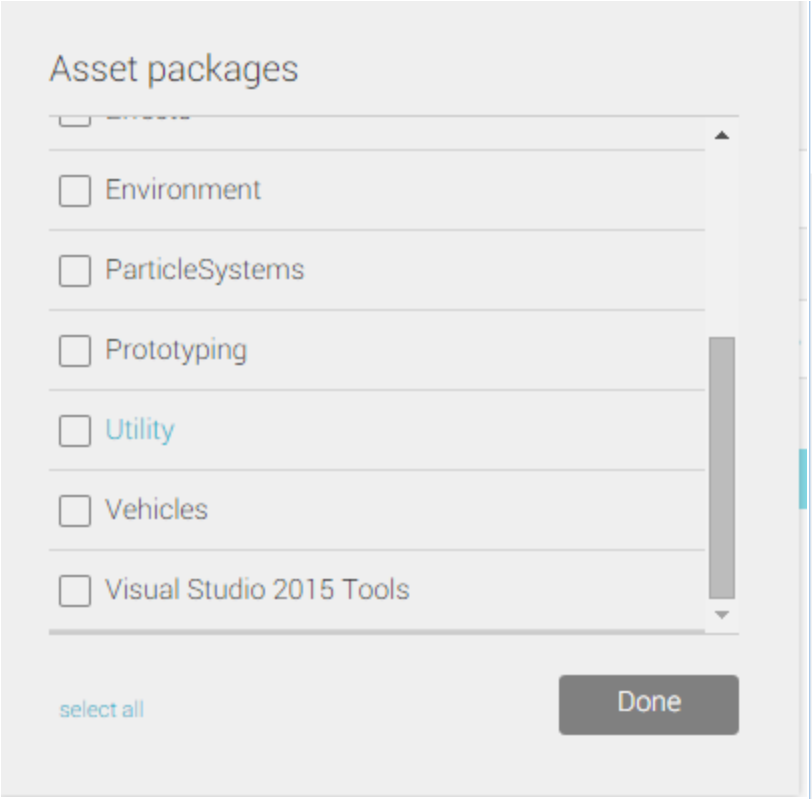
Getting Setup with Unity

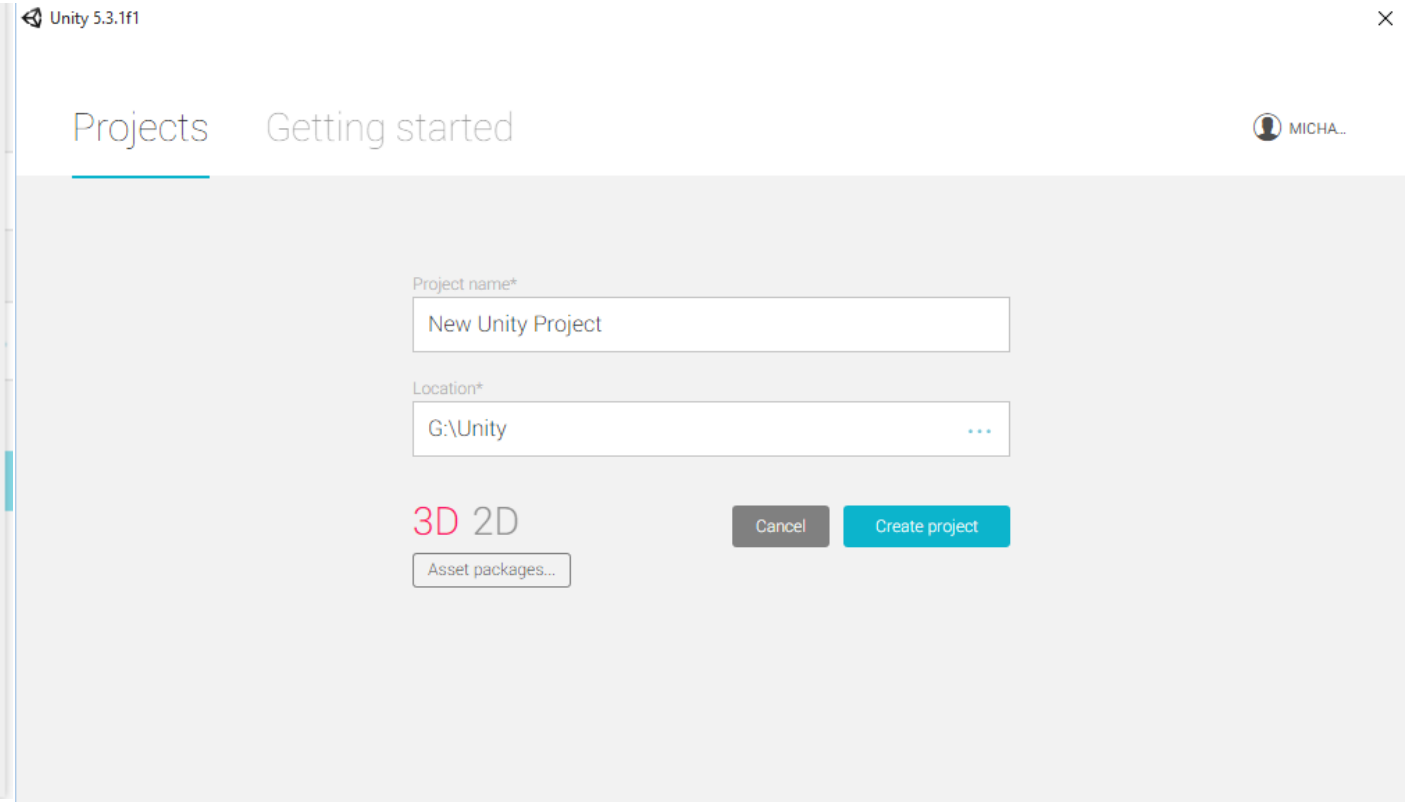
THERE REALLY ISN'T MUCH TO SETUP.

Installation

- Download the ‘Personal Edition’: https://unity3d.com/get-unity
- Select appropriate build targets during installation!
- iOS: Come to my office hours to get added to the Stanford’s Apple Developer account
  - Wait for an announcement on Piazza. Currently some licensing problems that need to be resolved by Stanford’s legal department…may take awhile.

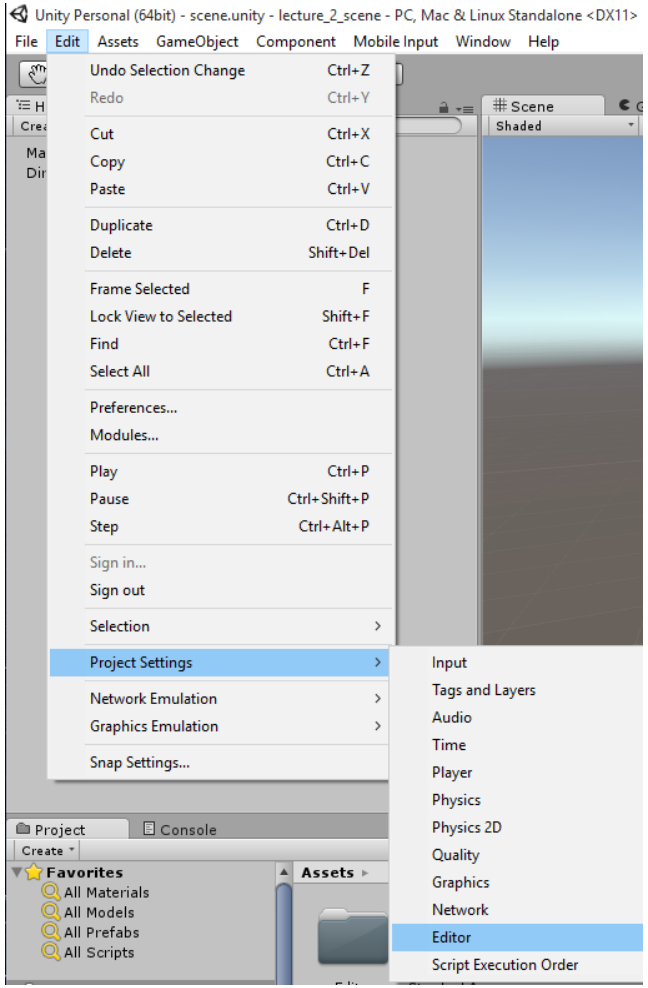
Project Creation

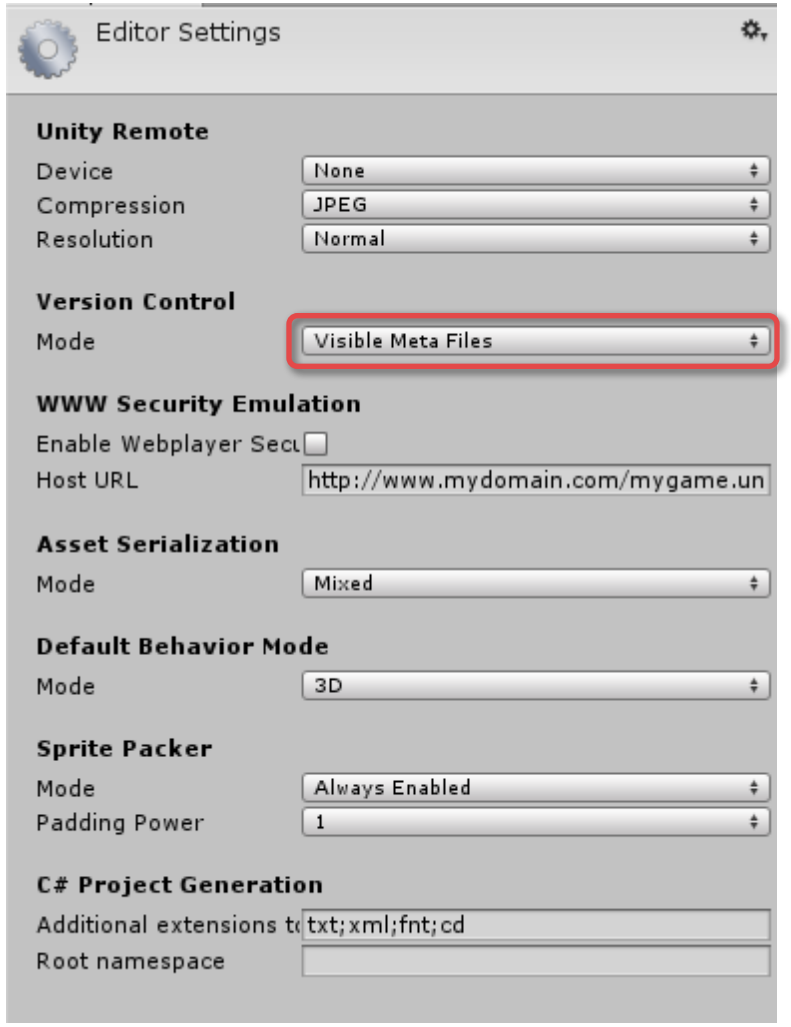




Source Control Setup (Git)

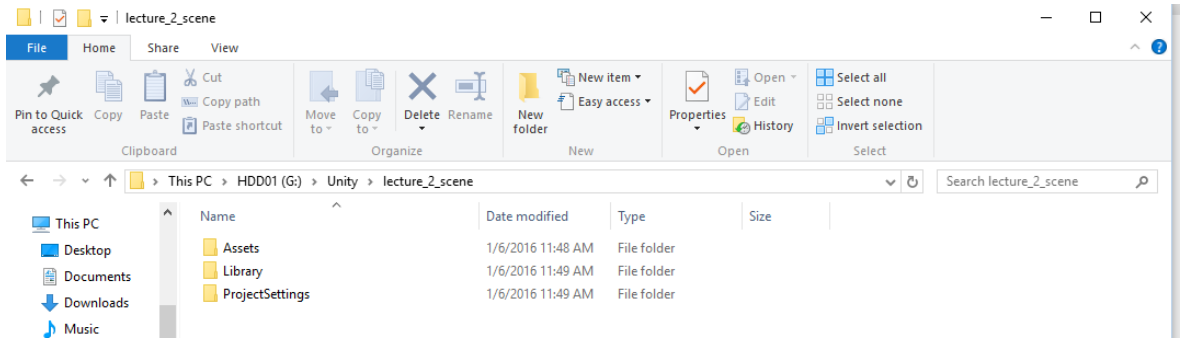
- Note: If you do end up using source control (i.e. Github, Bitbucket), make your repository **private**.





Source Control Setup (Git)

- Assuming you have your repository setup (Github comes with a ".gitignore" for Unity, use it!
- Go to where your project is located (from project setup screen: Location followed by the project name)
- Delete the "Library" folder.
- Point your command line git to the folder and run:
  - git init
  - git remote add origin GIT_URL_HERE
  - git pull origin master
  - git commit -am "COMMIT_MESSAGE"
  - git push origin master
- Unity Docs: http://docs.unity3d.com/Manual/ExternalVersionControlSystemSupport.html
- Smart Merge: http://docs.unity3d.com/Manual/SmartMerge.html
  - Will be useful for collaboration on scenes/prefabs!



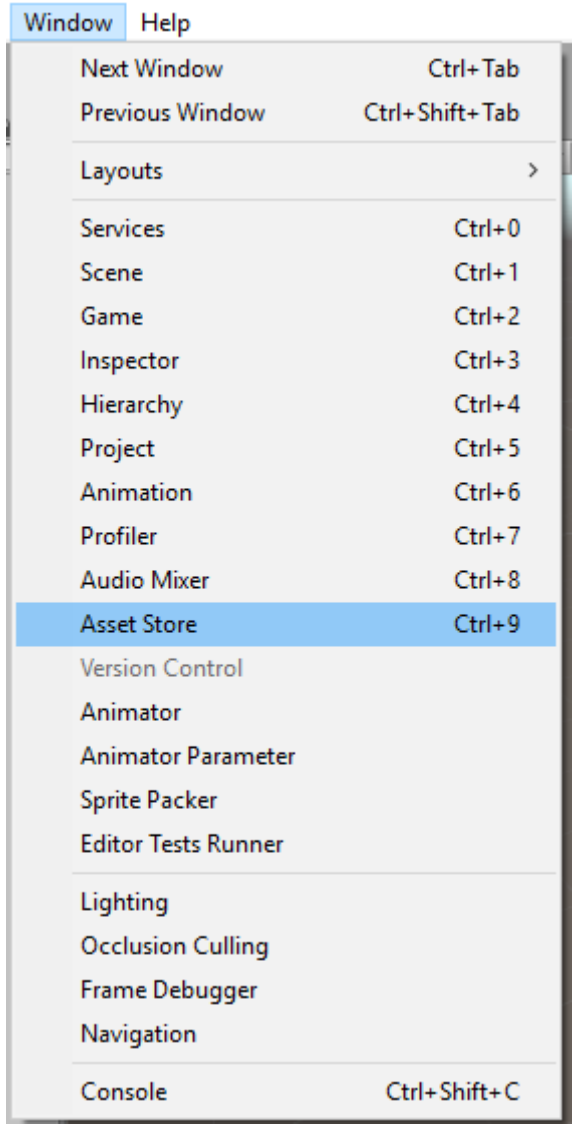
Unity Resources

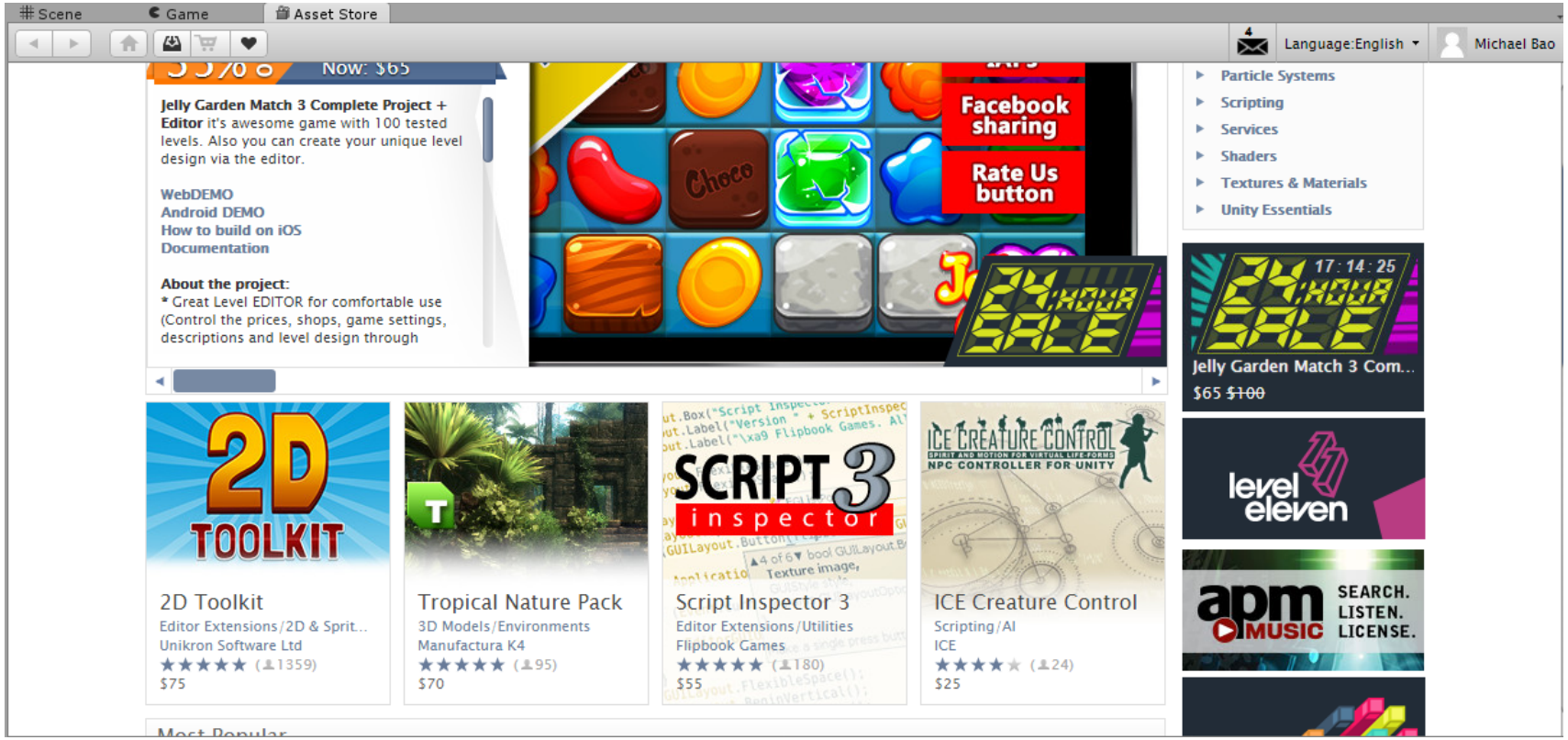
PLEASE READ THE DOCUMENTATION

Documentation

- Manual: http://docs.unity3d.com/Manual/index.html
- Scripting API: http://docs.unity3d.com/ScriptReference/index.html
- These pages should become your best friends.
- Also: http://forum.unity3d.com/

Asset Store


Window
Help
Next Window Ctrl+Tab
Previous Window Ctrl+Shift+Tab
Layouts
Services Ctrl+0
Scene Ctrl+1
Game Ctrl+2
Inspector Ctrl+3
Hierarchy Ctrl+4
Project Ctrl+5
Animation Ctrl+6
Profiler Ctrl+7
Audio Mixer Ctrl+8
Asset Store Ctrl+9
Version Control
Animator
Animator Parameter
Sprite Packer
Editor Tests Runner
Lighting
Occlusion Culling
Frame Debugger
Navigation
Console Ctrl+Shift+C

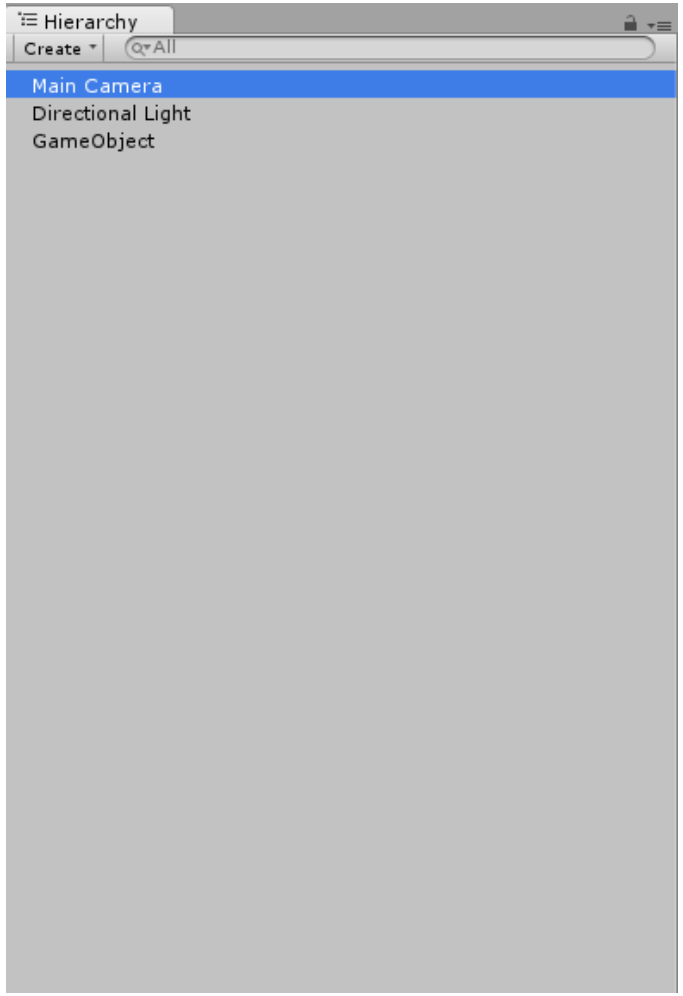

Scene
Game
Asset Store
Language:English
Michael Bao
Now: $65
Jelly Garden Match 3 Complete Project + Editor it's awesome game with 100 tested levels. Also you can create your unique level design via the editor.
WebDEMO
Android DEMO
How to build on iOS
Documentation
About the project:
* Great Level EDITOR for comfortable use (Control the prices, shops, game settings, descriptions and level design through
Facebook sharing
Rate Us button
Particle Systems
Scripting
Services
Shaders
Textures & Materials
Unity Essentials
24 HOUR SALE
17:14:25
Jelly Garden Match 3 Com...
$65 $100
2D TOOLKIT
2D Toolkit
Editor Extensions/2D & Sprit...
Unikron Software Ltd
$75
Tropical Nature Pack
3D Models/Environments
Manufactura K4
$70
SCRIPT 3 inspector
Script Inspector 3
Editor Extensions/Utilities
Flipbook Games
$55
ICE CREATURE CONTROL
NPC CONTROLLER FOR UNITY
ICE Creature Control
Scripting/AI
ICE
$25
level eleven
apm MUSIC
SEARCH. LISTEN. LICENSE.

Unity Overview

NOT A COMPREHENSIVE OVERVIEW

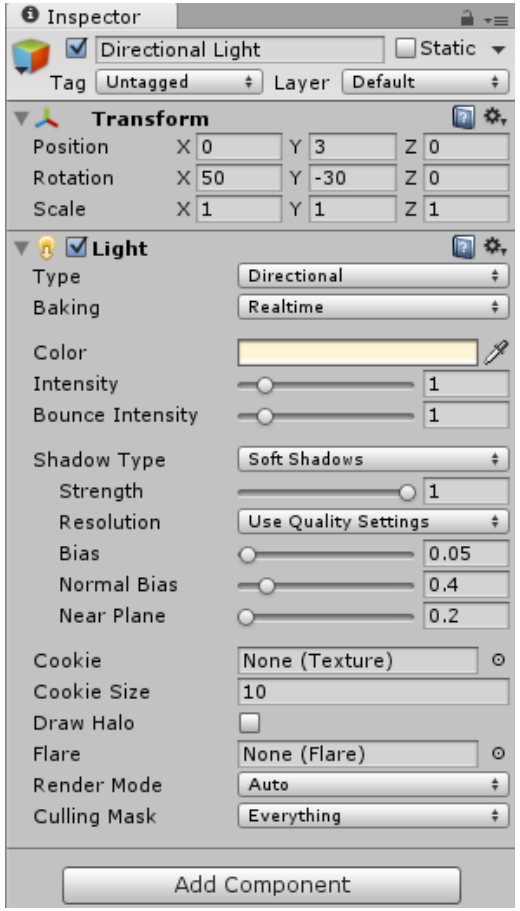
Everything is a “GameObject”

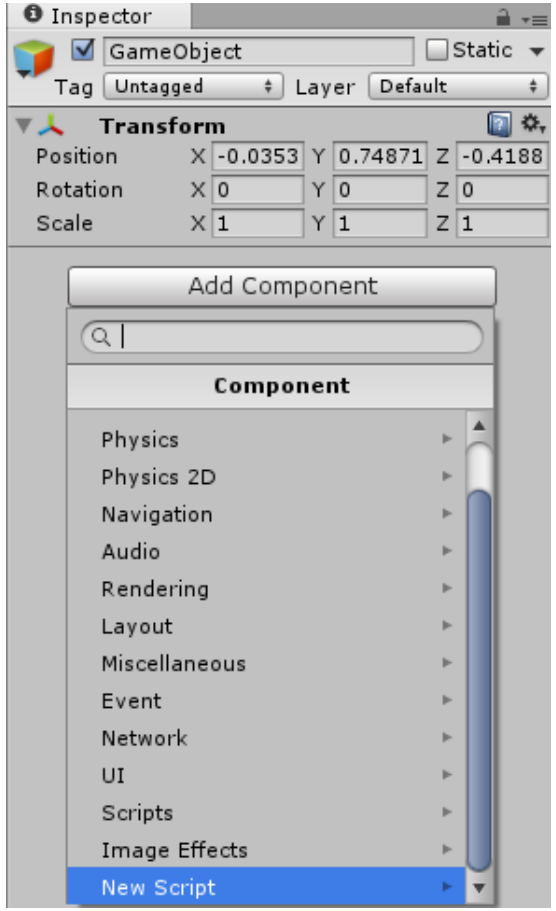
- All objects in your scene hierarchy
  - Cameras
  - Lights
  - Gameplay Logic
  - User Interface
  - Etc.
- Can be destroyed/created on the fly or be made to not get destroyed when the game changes scenes (this is how you get persistent behavior!):
  - http://docs.unity3d.com/ScriptReference/Object.DontDestroyOnLoad.html



Everything is a “GameObject”

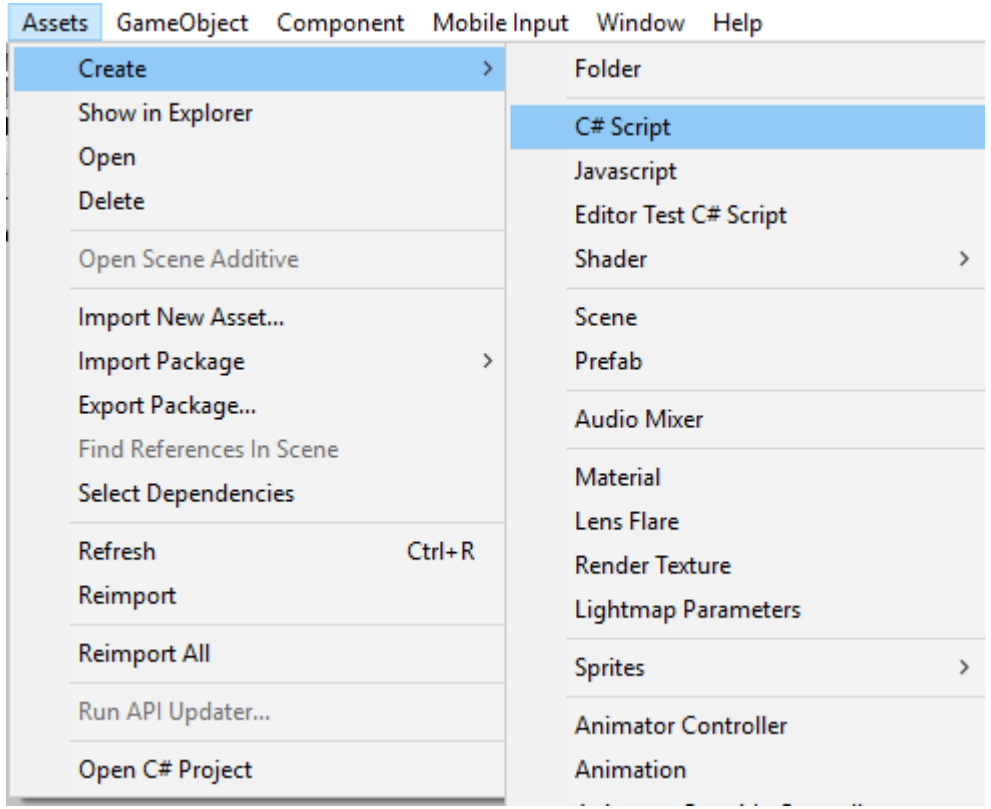
- A “GameObject” does nothing on its own.
- All “GameObject” have a “Transform” component to let it know its position/rotation/scale.
- Must add “Components” to the GameObject to give it some behavior.

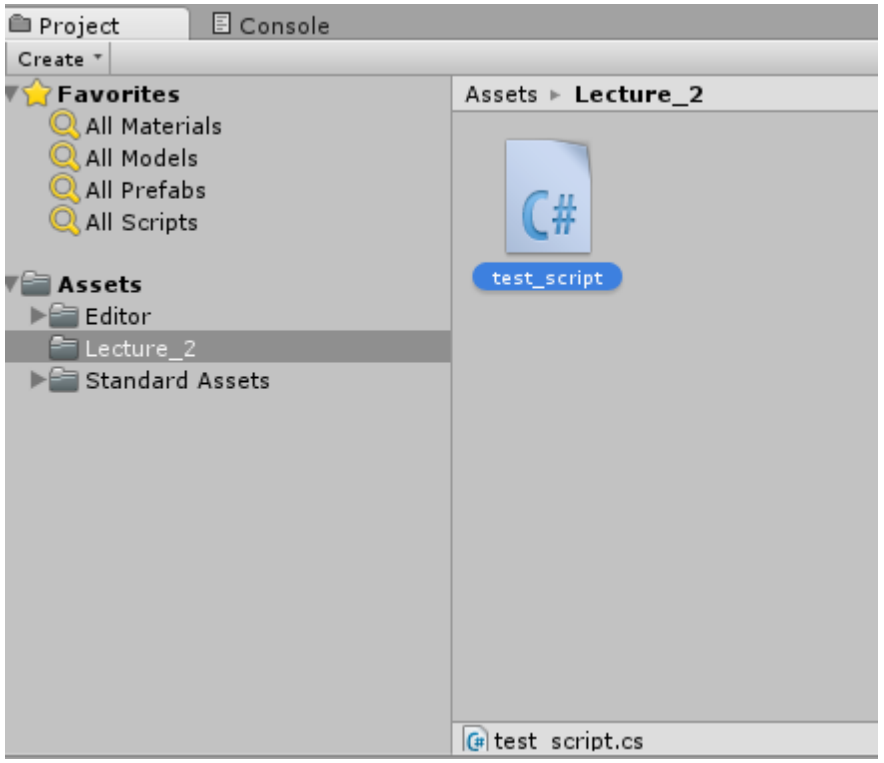




Everything is a “GameObject”

- Many components already exist! But you might want to create your own. These are your scripts that inherit from “MonoBehaviour”.
- Strong Recommendation: Code in C#.





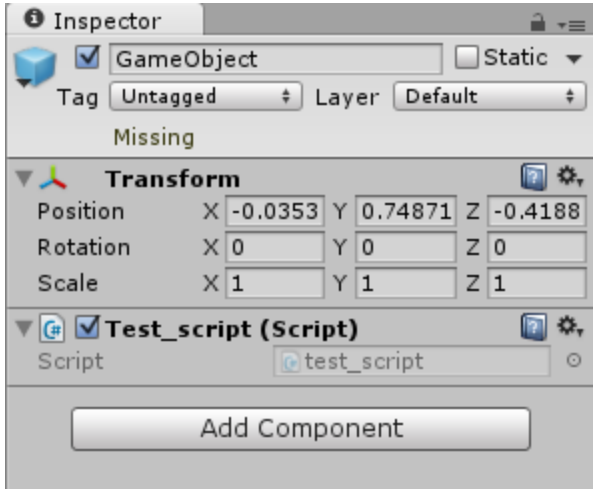
Everything is a “GameObject”

- GameObject: http://docs.unity3d.com/ScriptReference/GameObject.html
- MonoBehaviour: http://docs.unity3d.com/ScriptReference/MonoBehaviour.html
- Drop and Drop Script onto GameObject in the “Inspector” or manually add it by going to Add Component > Scripts > YOUR_SCRIPT_NAME_HERE
- Public variables will show up in the Inspector. A variable that is a Component can also be modified by the inspector! Very useful.

test_script.cs

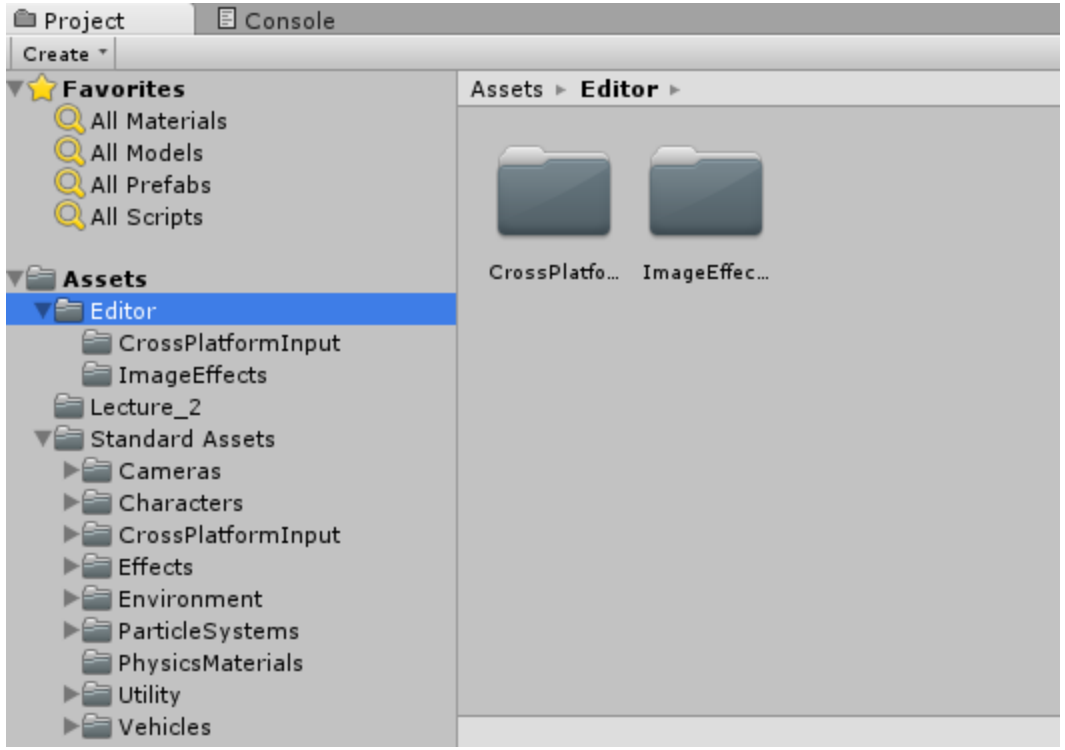
lecture_2_scene.CSharp

```
using UnityEngine;
using System.Collections;

public class test_script : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

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

    }
}
```



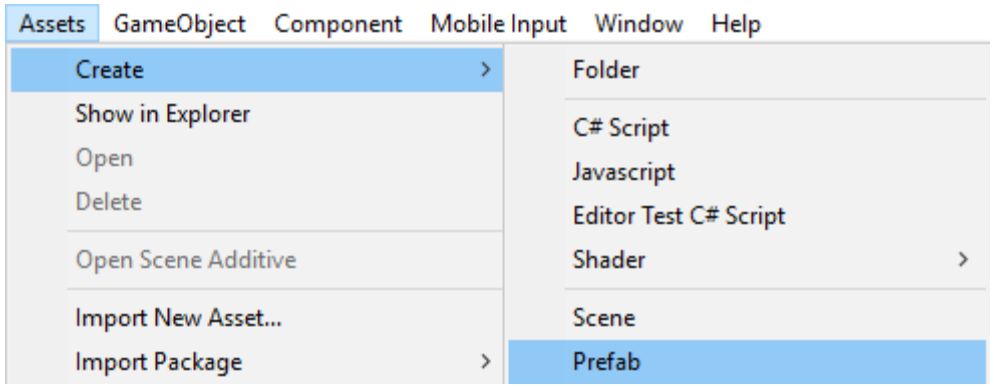
Assets

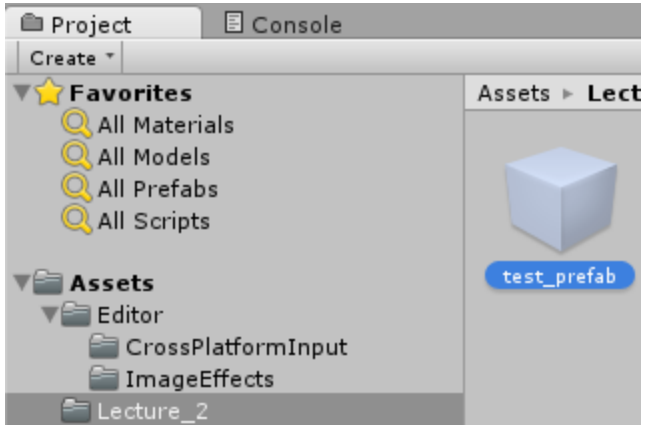
- Scenes
- “Prefabs”
- Scripts
- Textures
- Animations
- Models
- Particles
- Sprites
- Etc.



Prefabs

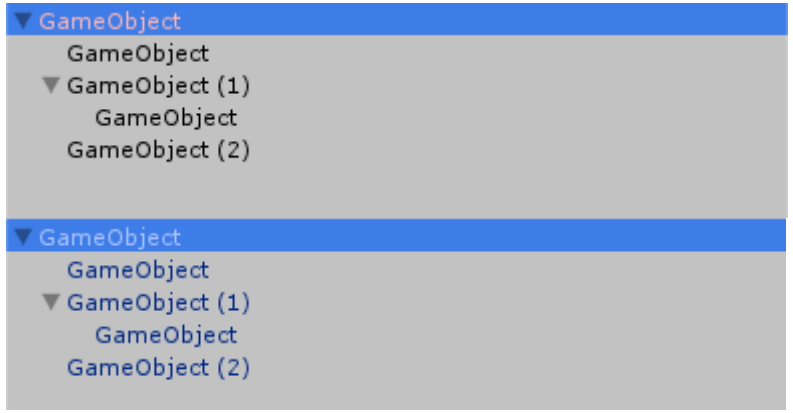
- You created some hierarchy of GameObjects and you want to reuse this hierarchy in multiple places. Use prefabs!
- You can also assign prefabs to scripts which have a public variable of type “GameObject” to be able to programmatically spawn prefabs! Or you could just load it by its path (may be more convenient in certain cases): http://docs.unity3d.com/ScriptReference/Resources.Load.html

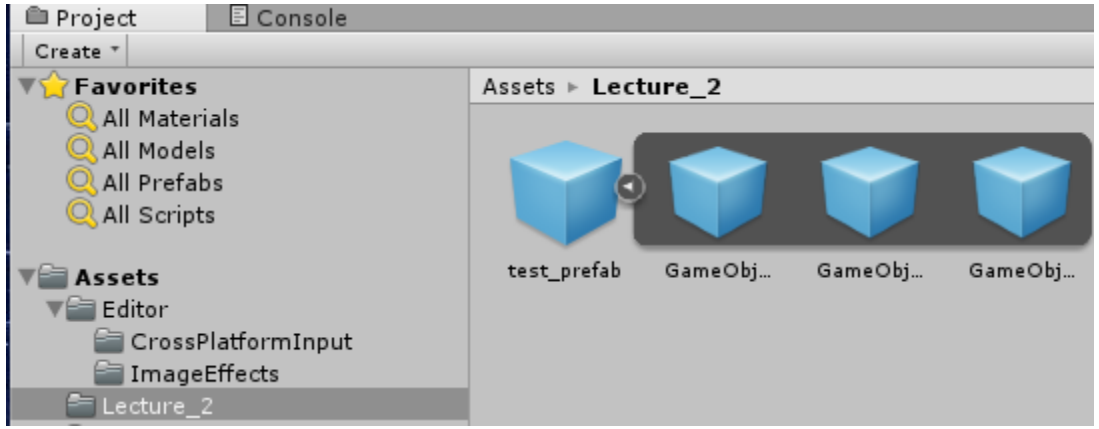




- Create an empty prefab and it will go into your assets folder. A gray prefab is an empty prefab!

Prefabs





- Drag and drop the GameObject onto the gray prefab object and the prefab will turn blue! (and so will the GameObject in the hierarchy). Blue indicates that it is an instance of a prefab and Unity will warn you if you try to change that object!
- Now you can delete the GameObject that exists in the hierarchy, drag and drop the prefab back into the hierarchy and you will see the GameObject show up again!

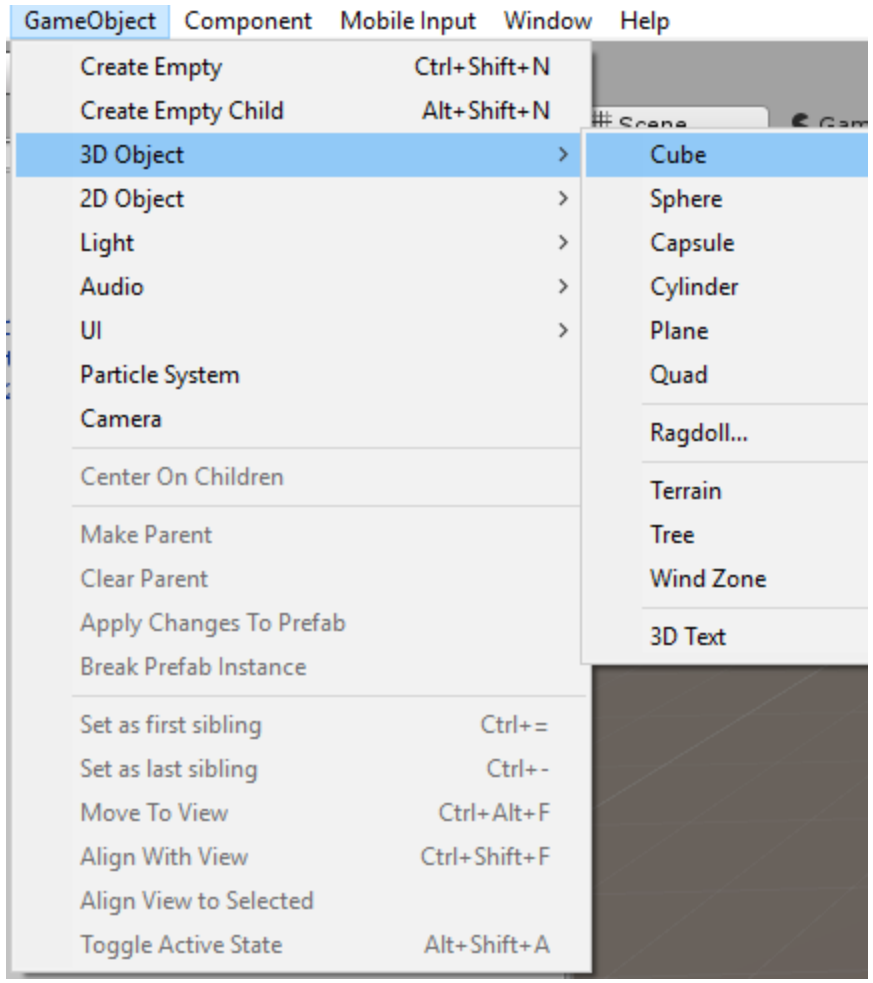
Editor Camera Controls

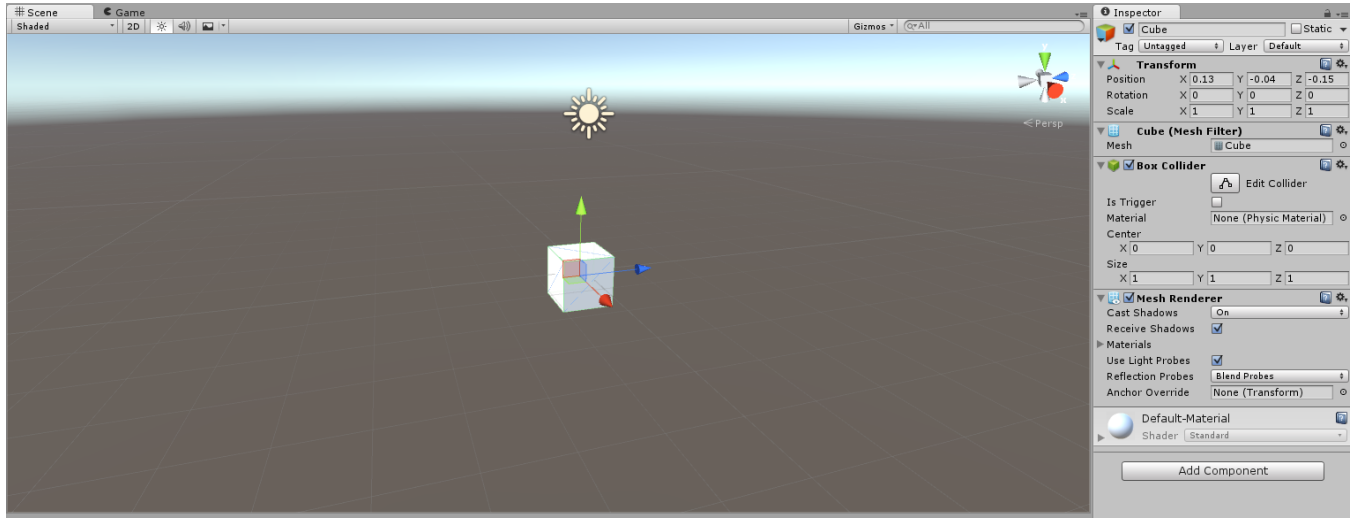
- “Maya”-like Controls:
  - Alt + Left Click & Move: Rotate Camera
  - Alt + Right Click & Move (Or Scroll Up/Down): Move camera back and forth
  - Alt + Middle Click & Move: Move camera up/down or left right
- Flythrough Mode:
  - Click and hold right mouse button and now you can use FPS-like controls to move around through the scene (WASD, Q/E to move up down).
- Unity Documentation: http://docs.unity3d.com/Manual/SceneViewNavigation.html

Editor Object Controls

- “Maya”-like Controls (after selecting an object):
  - W: Activate translation widget.
  - E: Activate rotation widget.
  - R: Activate scale widget.
- Manual Movement:
  - Modify position/rotation/scale in the Inspector.
  - Rotation is in Euler angles. Rotation order: Z, X, Y.
- Unity Documentation: http://docs.unity3d.com/Manual/PositioningGameObjects.html

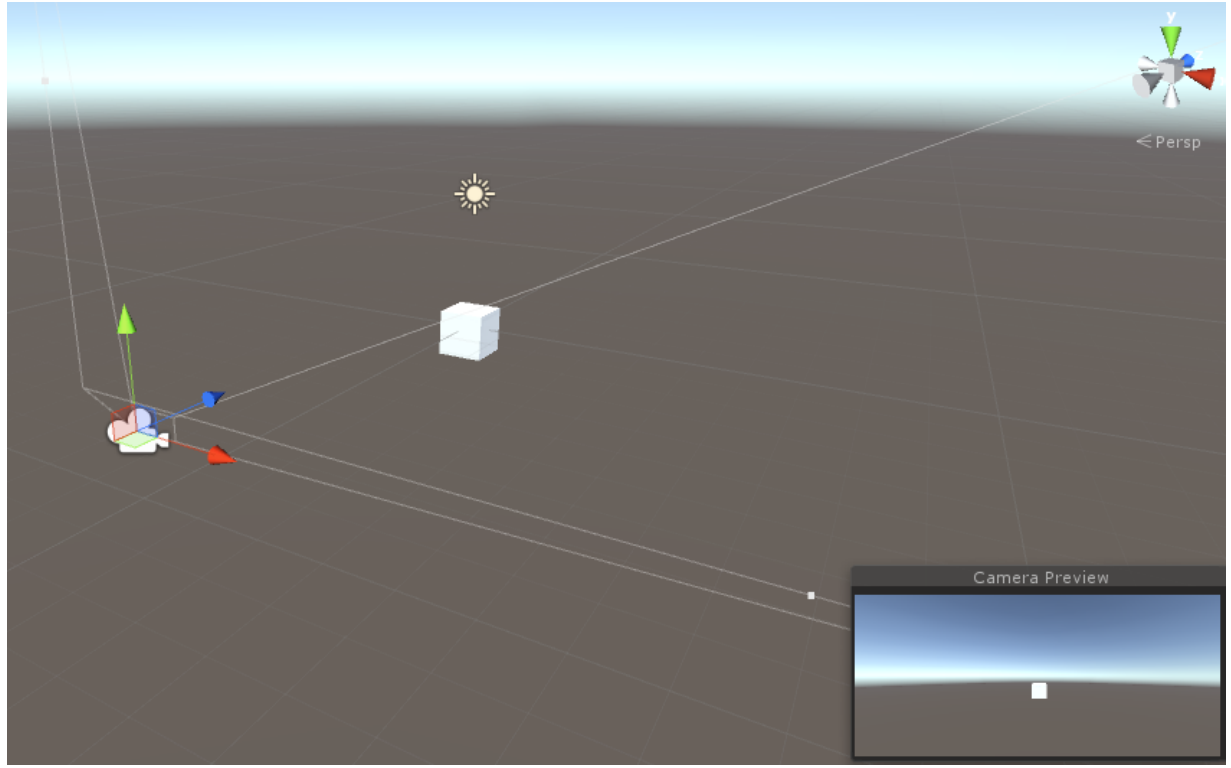
Creating Geometry via the Unity Editor





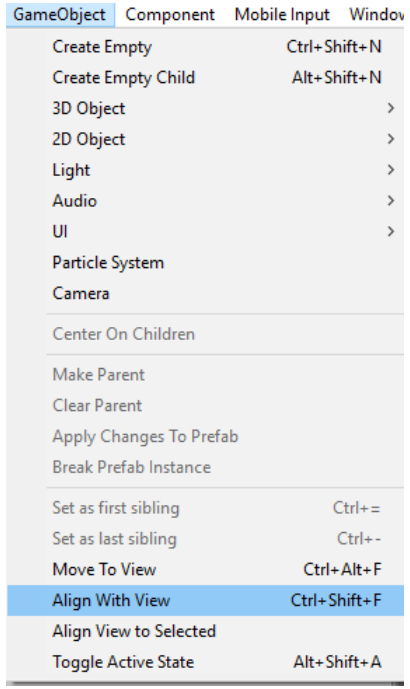
Setting Up The Scene Camera

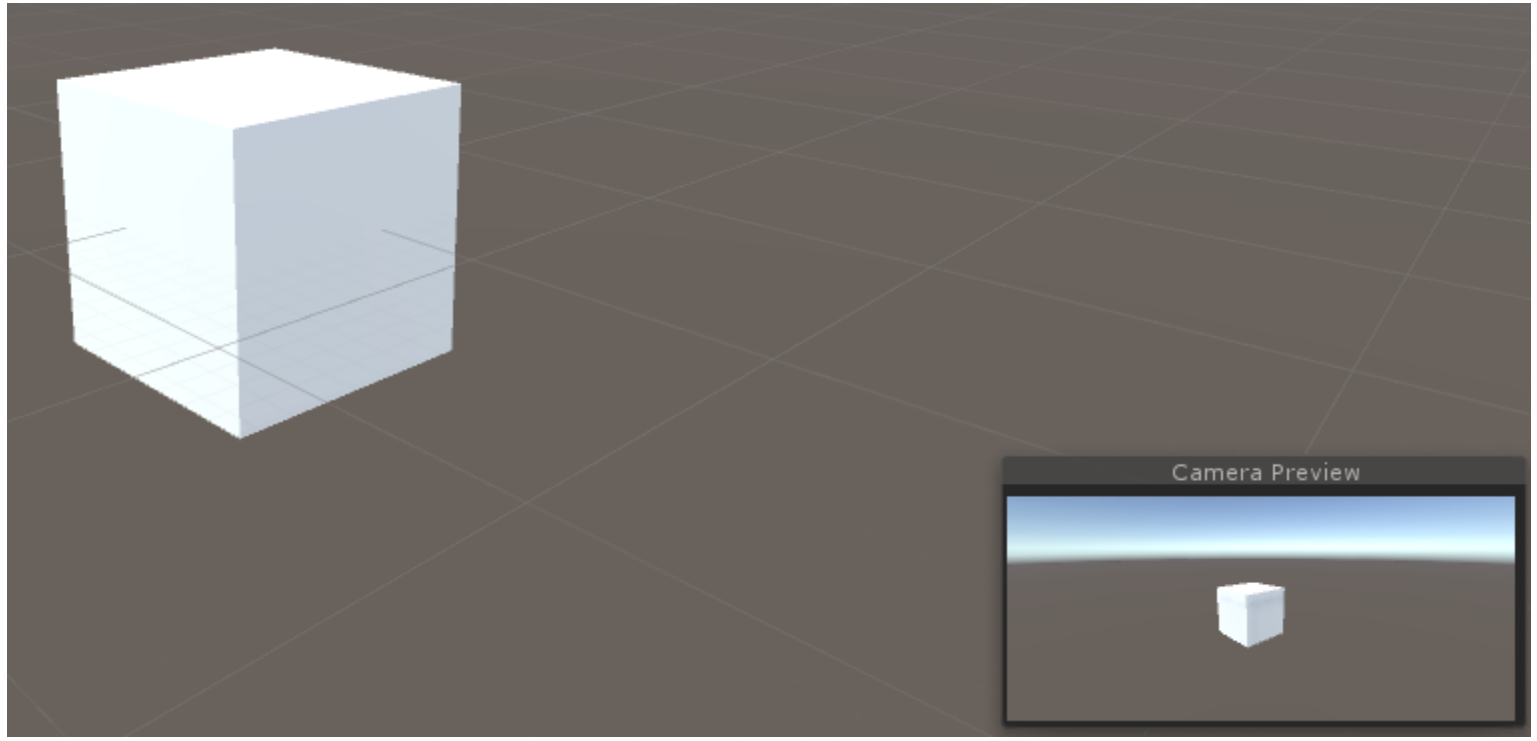
- Do not confuse the scene camera with the editor camera.
- Unity scenes by default come with a “Main Camera.” Notice the tag of “MainCamera” in the inspector, this will be useful for accessing the camera from your scripts.
- “Camera Preview” box is useful to see what your camera can see.
- “Camera Preview” is what you will see when you hit Play!



Setting Up The Scene Camera

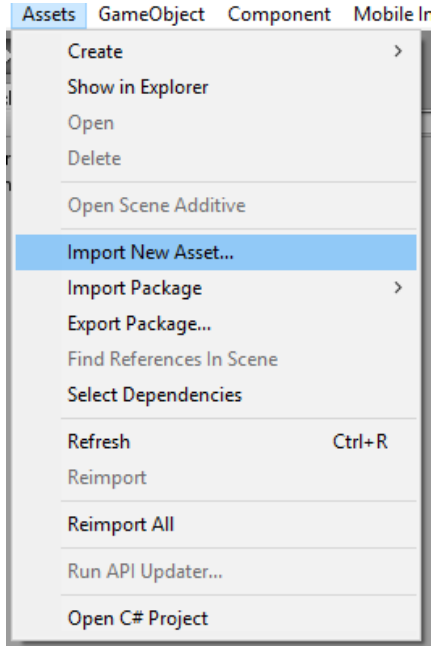
- Moving the scene camera can be done manually by changing the position/rotation/scale in the Inspector.
- Or you can move the editor camera around as mentioned earlier, select the camera and align the camera to the view. Note that your camera should be selected before doing this.
- Hit play and this view will be sufficient for the screenshot in Assignment 1.

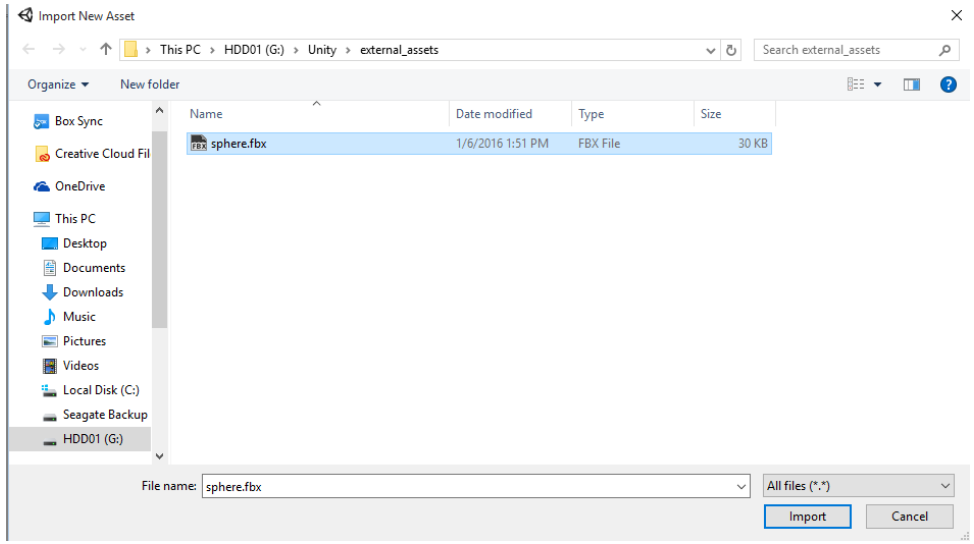




Import External Objects

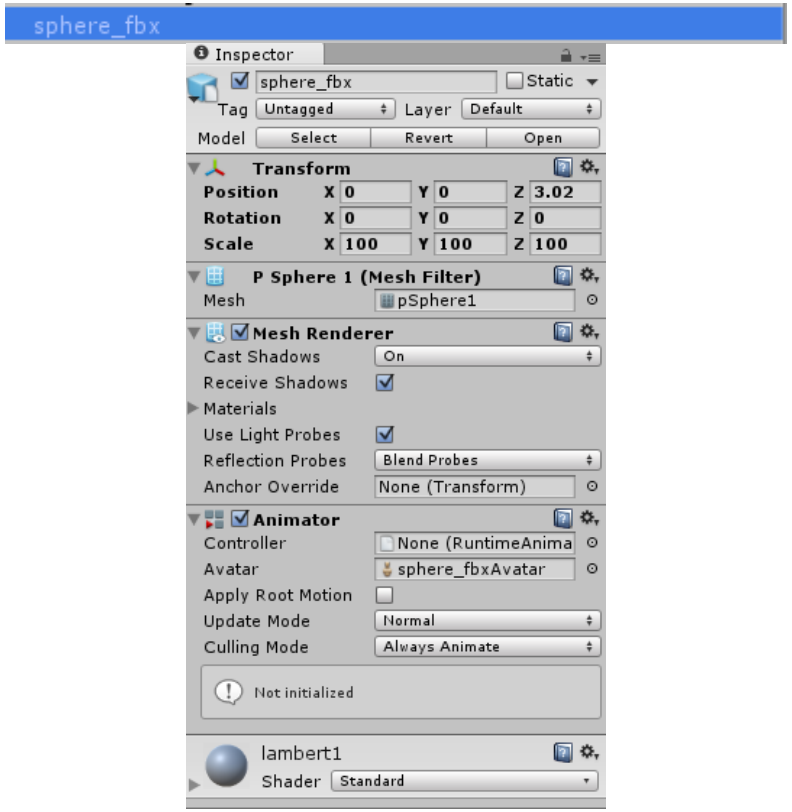
- Create and export an object from Maya/Blender/3ds Max as either an *.OBJ or a *.FBX. You can save this anywhere.
  - Why *.FBX? You will see later when we start talking about character animation.
- Then import this asset into Unity. Unity will take care of everything for you.
- Alternatively, you can just save your *.OBJ or *.FBX inside the “Assets” folder.
  - You will need to right click on the folder it is in and click “Refresh” to get it to show up.

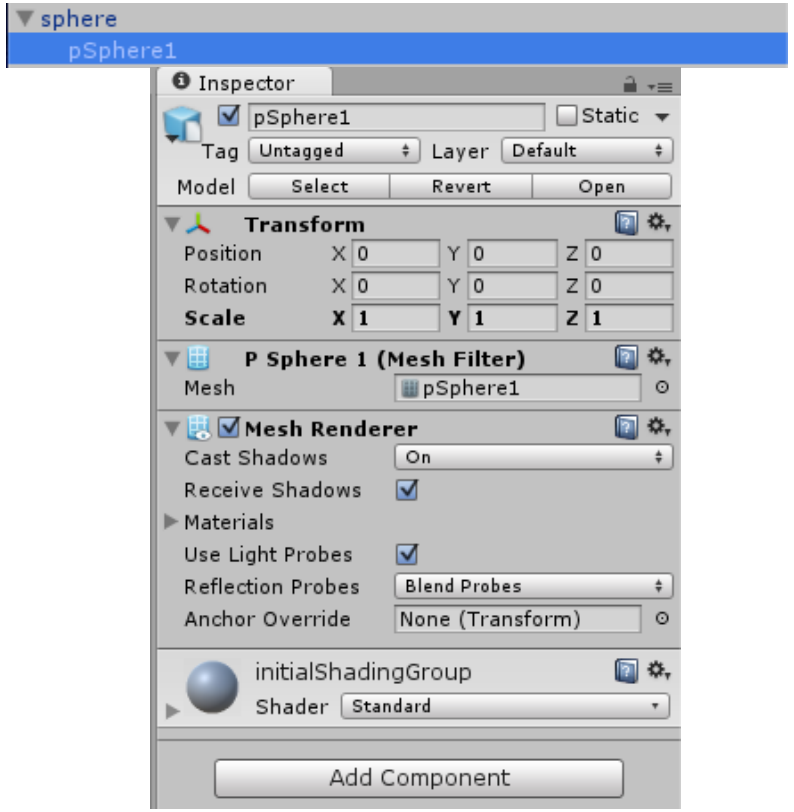




Import External Objects

- Click and drag the object from the assets library into your scene hierarchy and it should now show up! There is a difference in how the GameObject is setup depending on whether you use an FBX or OBJ but that difference is not relevant for now.



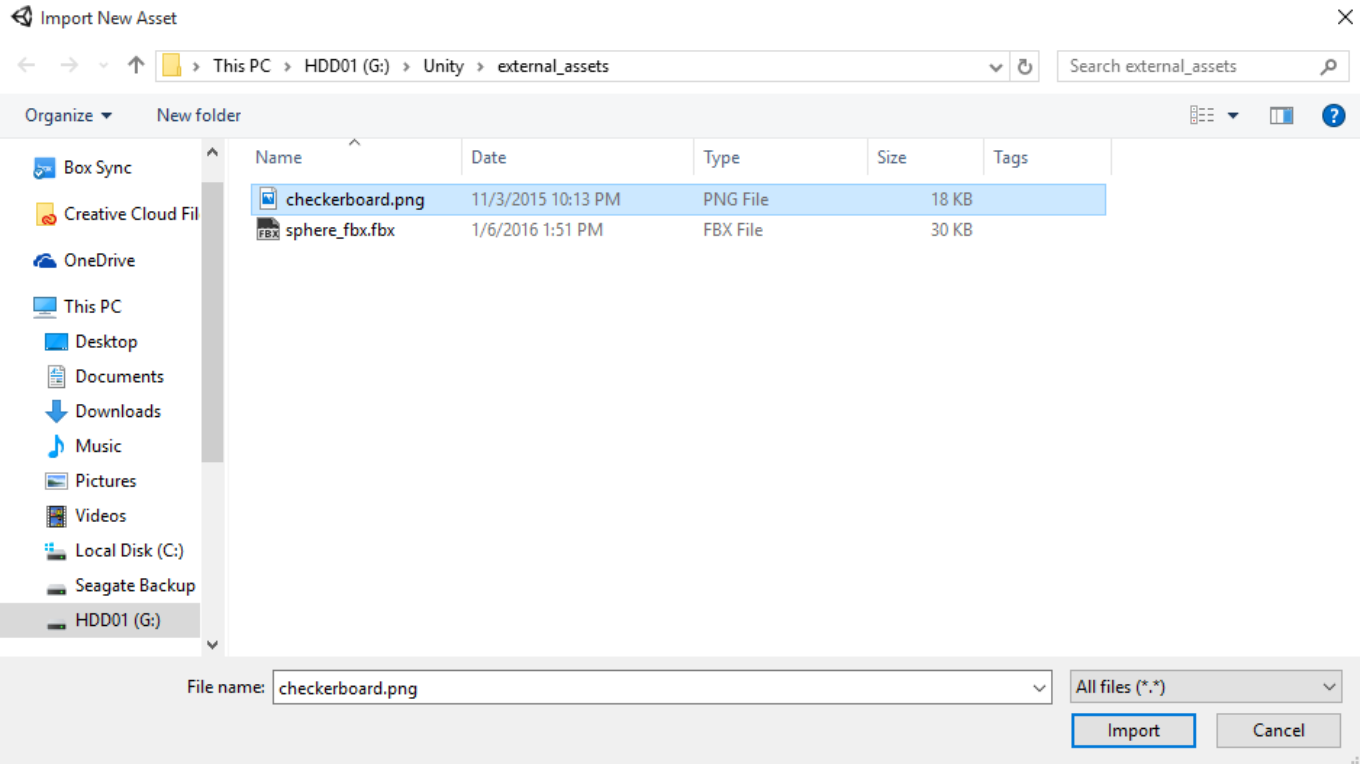


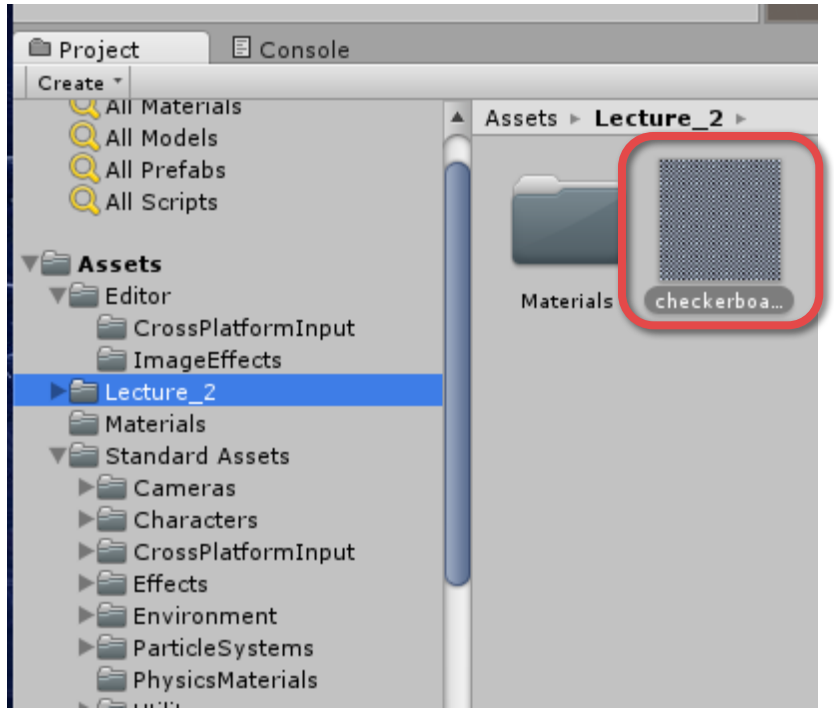
Shading and Materials

- Fairly easy to write your own shader but the Unity Standard shader is very powerful and can probably be used in most situations!
- In most cases you will want to select a shader that comes with Unity and modify the material properties to achieve the look that you want
- Manual Shader Documentation: http://docs.unity3d.com/Manual/ShadersOverview.html
- Standard Shader Documentation: http://docs.unity3d.com/Manual/shader-StandardShader.html
- Materials Documentation: http://docs.unity3d.com/Manual/Materials.html

Importing Textures

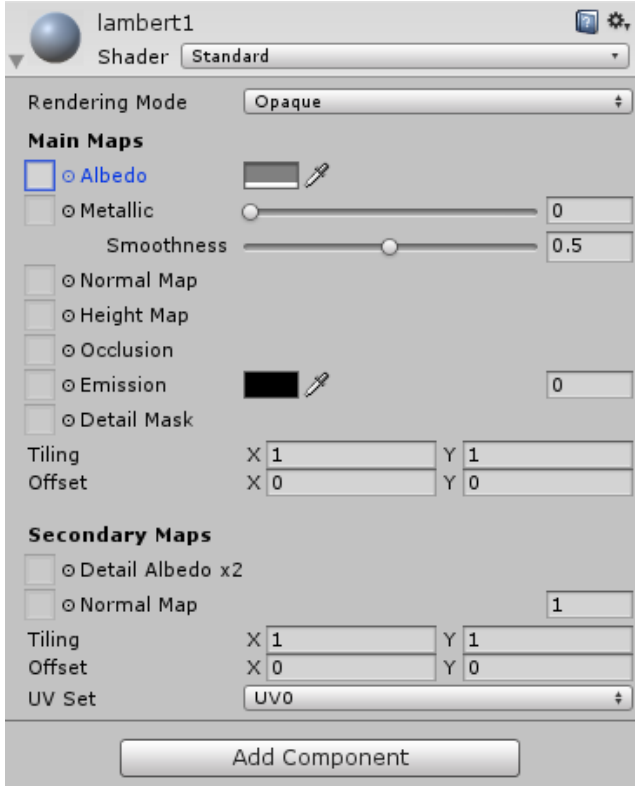
- Process is the same as importing an external object. This time, instead of selecting an *.FBX or *.OBJ, select a *.PNG, *.JPG, etc. You can also place the images inside the Assets folder manually.
- Note that you should make sure your imported object has a UV map before continuing! You can check this in Maya/Blender/3ds Max.





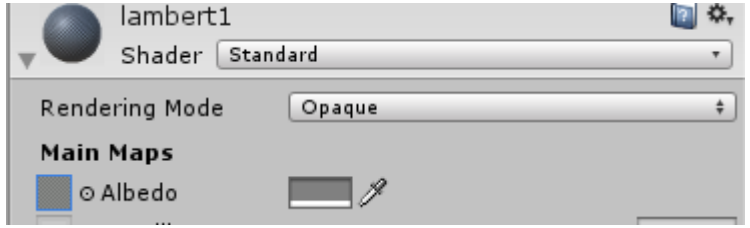
Using Textures

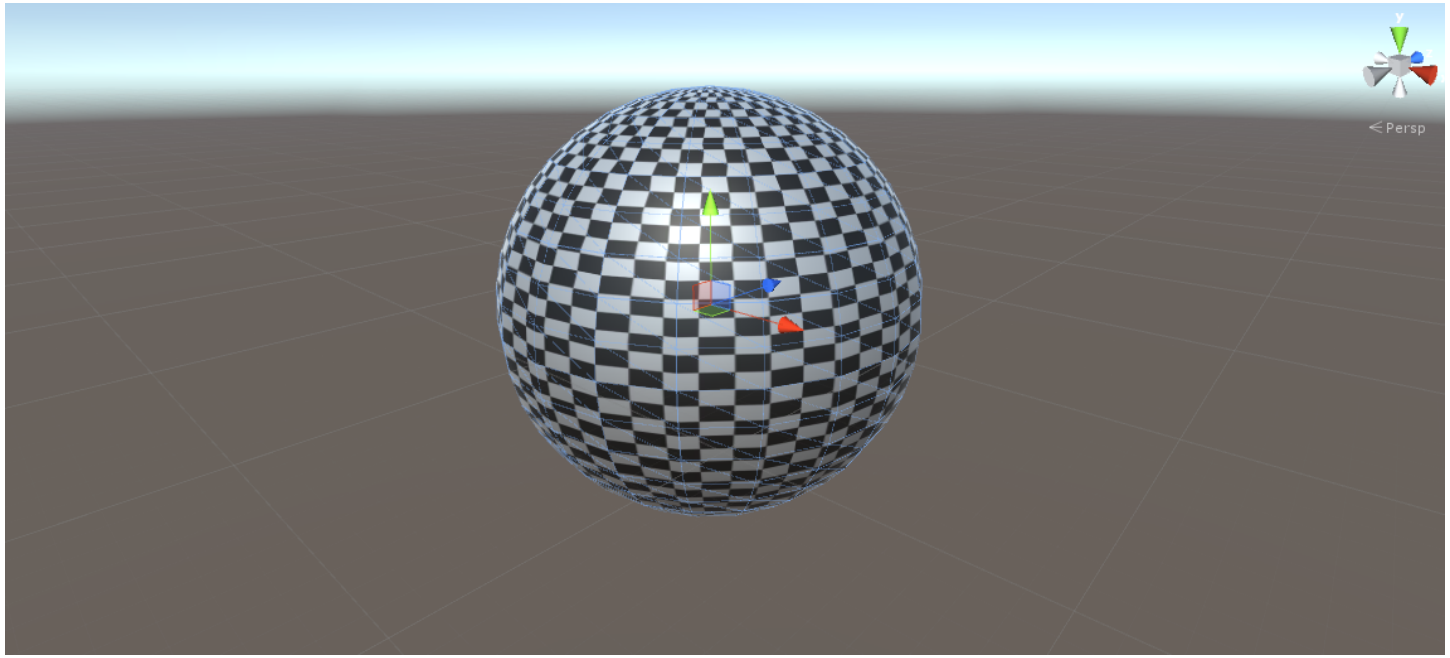
- Click on the object you imported in the scene hierarchy and expand the shader properties in the Inspector.
- The “diffuse” texture that we used in CS148 is now the “Albedo.” There is no distinction between the diffuse and specular color as you saw with the Epic BRDF.
  - Look at CS148 Assignment 4 if you want great (no bias) refresher on what the BRDF/Material does.
  - http://web.stanford.edu/class/cs148/assignments/assignment4.pdf



Using Textures (Option 1)

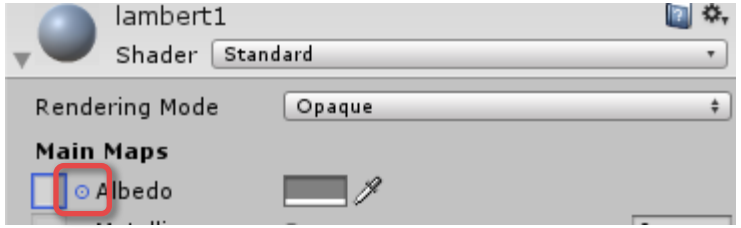
- Click and drag the imported texture onto the square next to “Albedo” and your object should now have a texture on it!

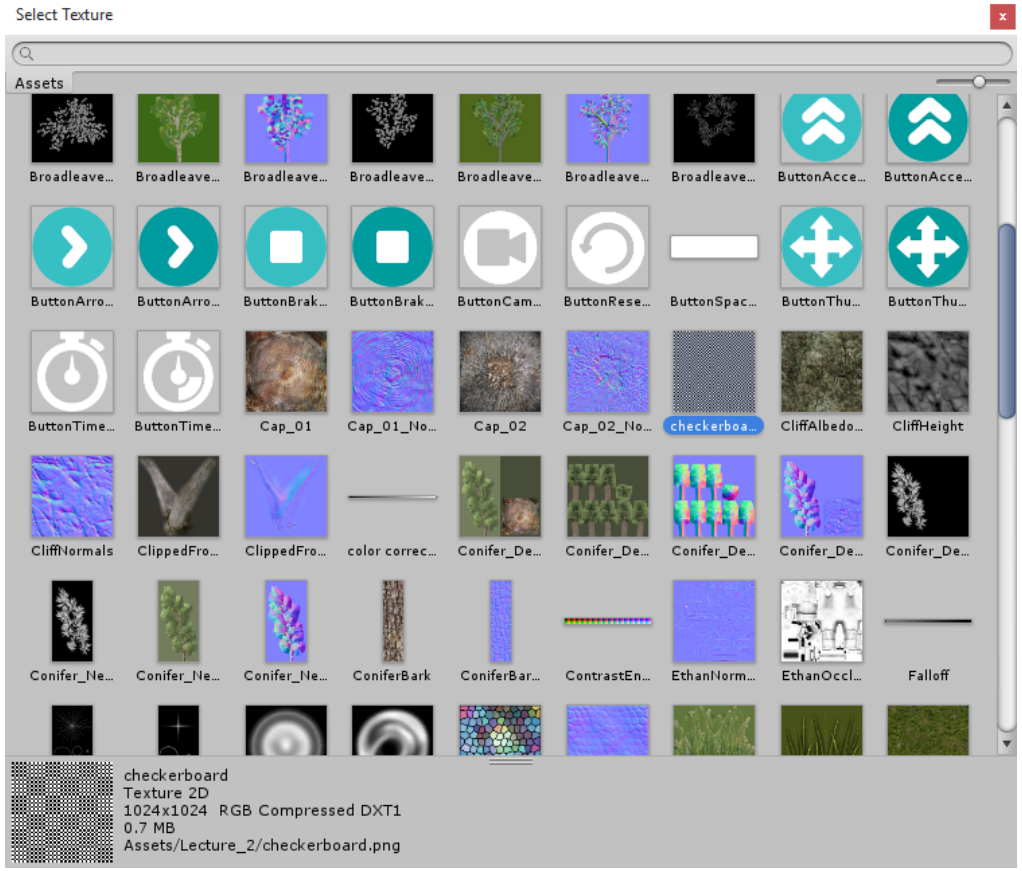




Using Textures (Option 2)

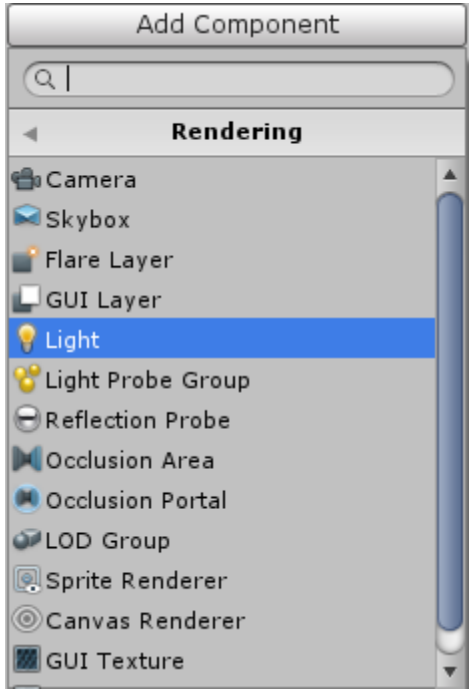
- Click on the circle next to Albedo and select your texture in the dialog box!



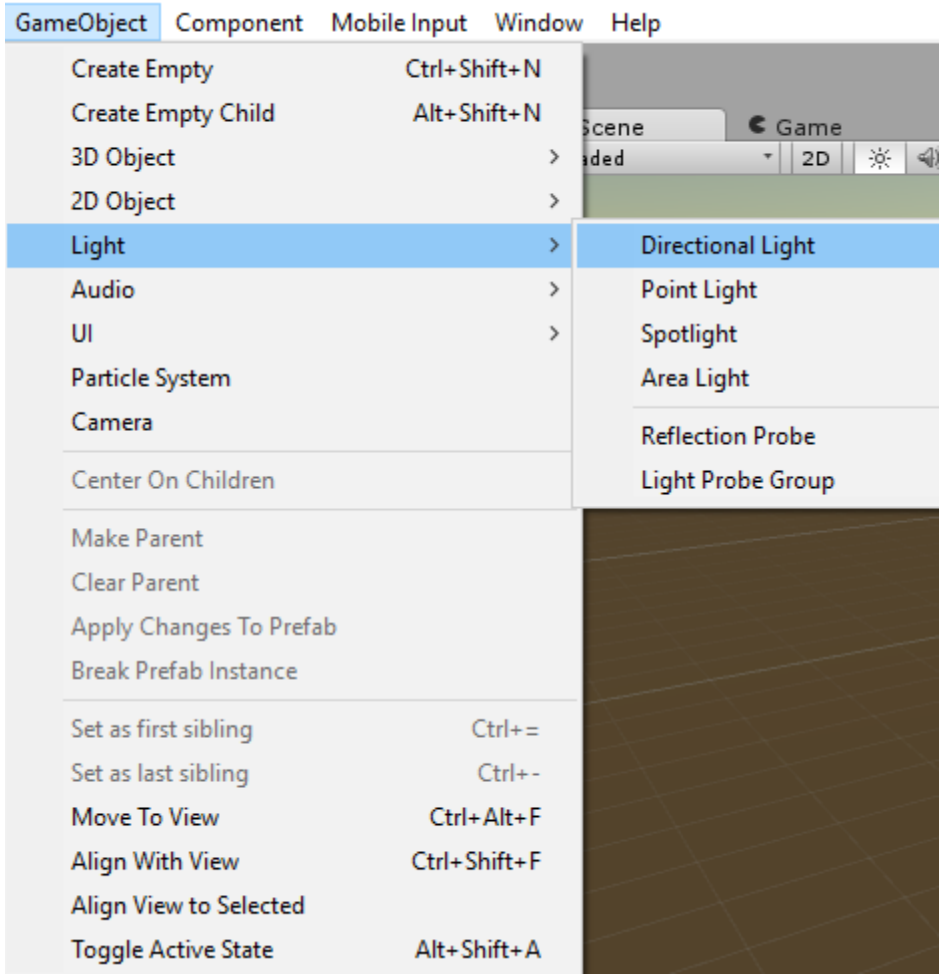


Lighting

- Lighting Documentation: http://docs.unity3d.com/Manual/Lighting.html
- Global Illumination Documentation: http://docs.unity3d.com/Manual/GlobalIllumination.html
- Lighting is accomplished with the “Light” component.

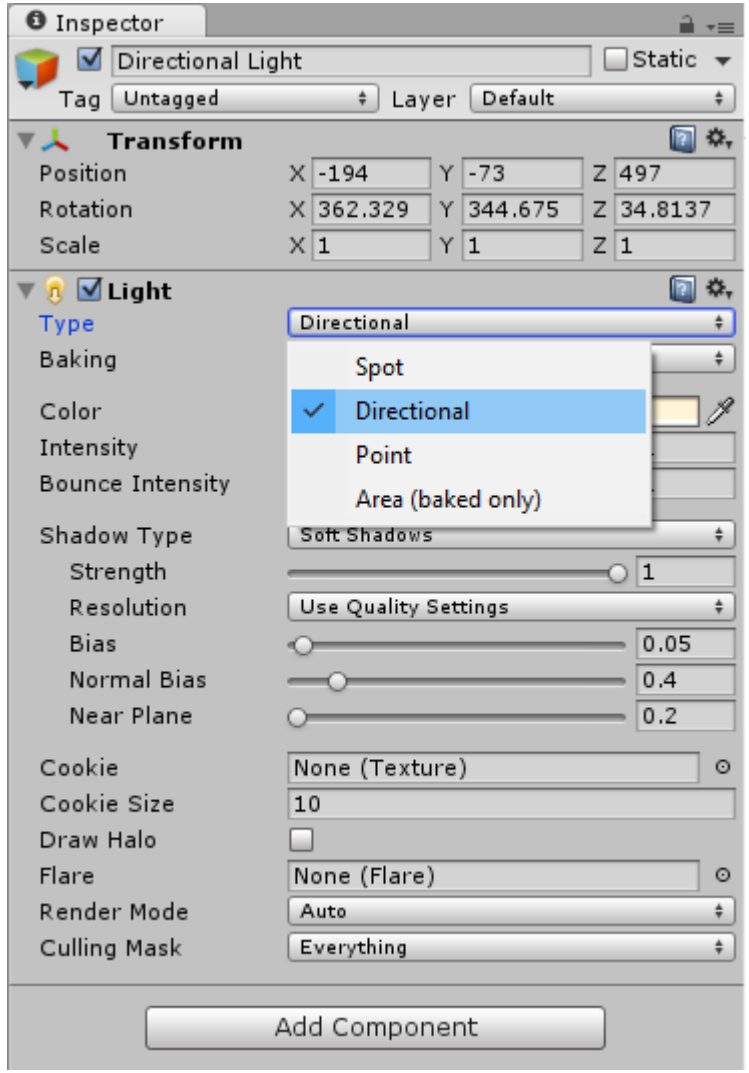


Creating a Light



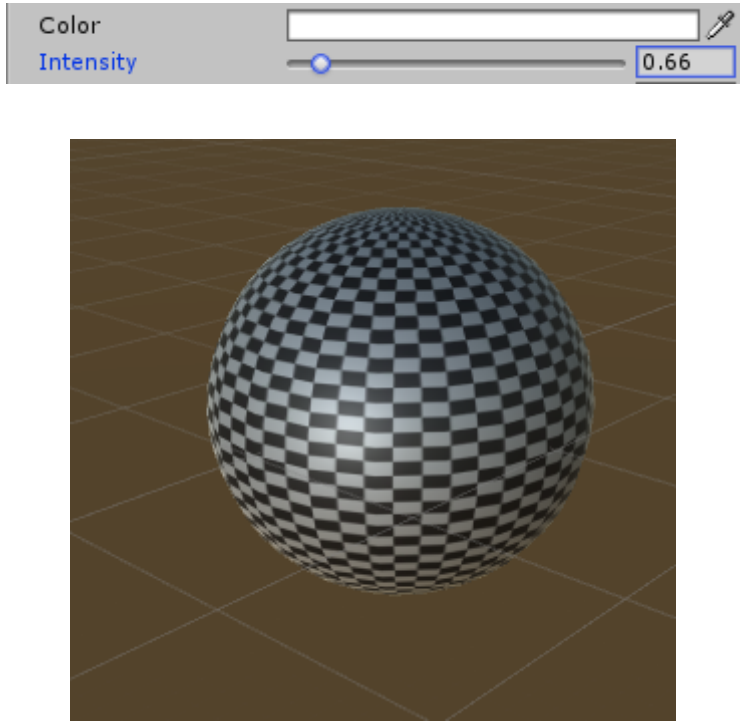
Light Component Properties

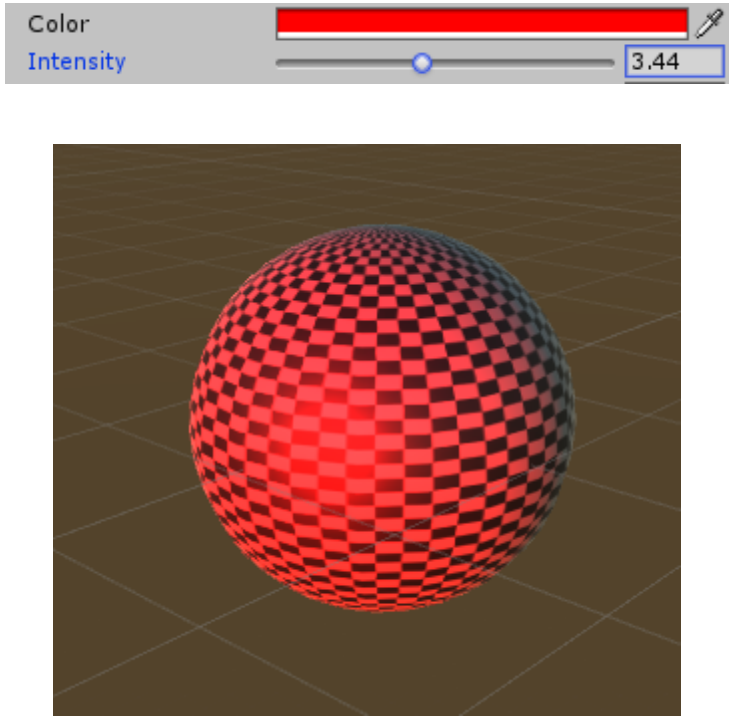
- Light Types:
  - Directional Light
  - Spot Lights
  - Point Lights
  - Area Lights



Light Component Properties

- Color and Intensity





Unity Tutorial

HOW TO DO
ASSIGNMENT 1

Questions?

YOU CAN ALSO ASK ME AFTER CLASS.