Before We Begin

1. Go to wp.me/a5G4dR-1dw, and click the Crash-Course-Unity-2018-Assets link to download the ZIP file, "Crash-Course-Unity-2018-Assets.zip".

Crash Course Unity 3D
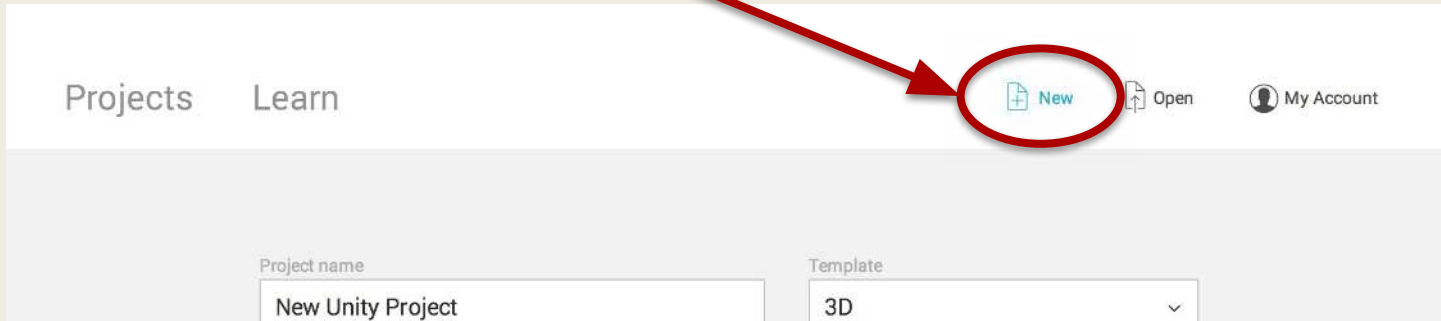
Introduction to how to use Unity

Goal

- Get comfortable with Unity game editor
- Create an interactive 3D environment
- Learn lots of 3D development terms
- A brief start in C# programming

Supplementary materials

- http://unity3d.com/learn/tutorials
  a. Official site full of tutorials on individual Unity feature
  b. Includes in-depth C# programming tutorials!

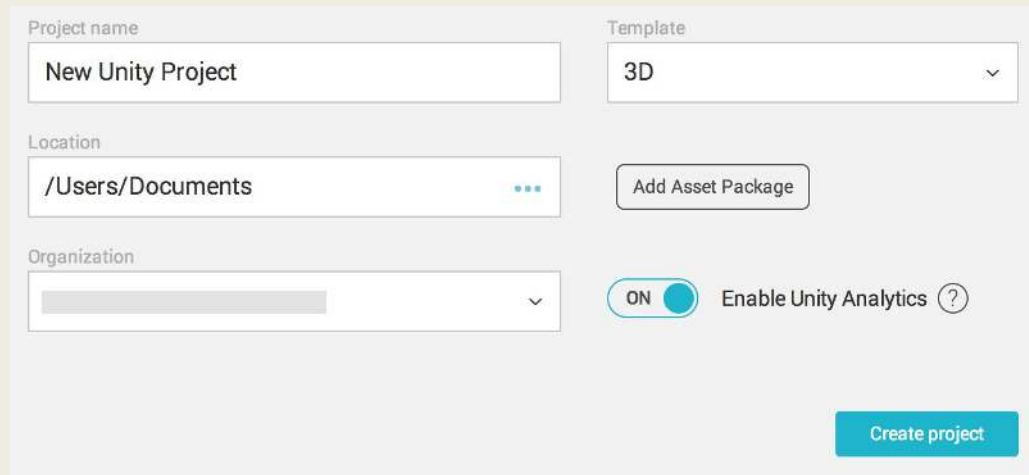# Starting a new project

Step 1: Click "New"

Starting a new project

Step 2: Type in a project name and the folder it'll be created in.
Note: Unity will create a new folder with the project's name
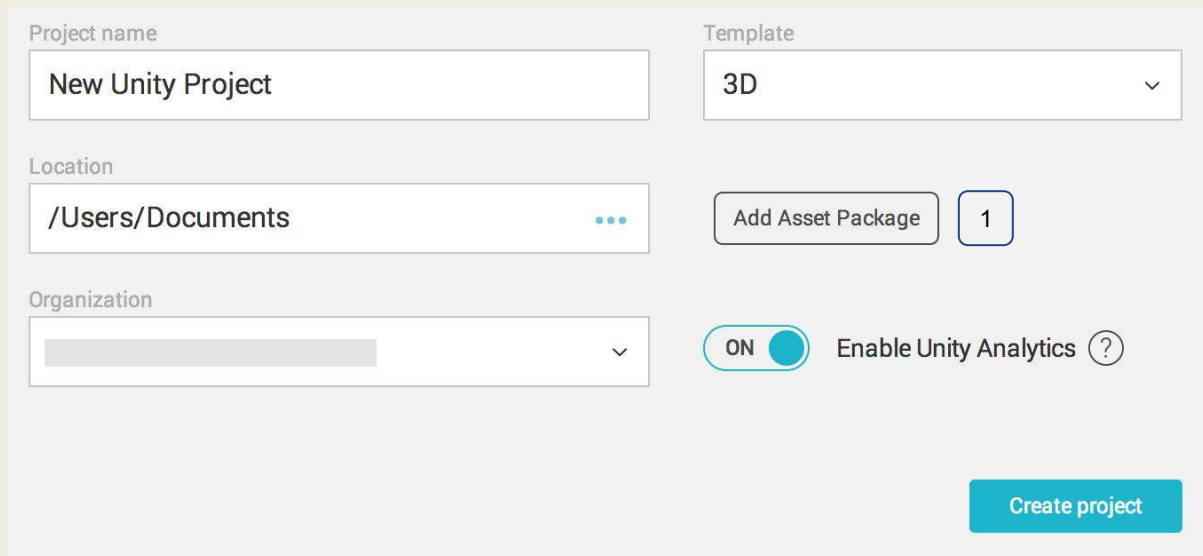
Starting a new project

Step 4: Click "Add Asset Package", and check "Standard Assets." Lastly, click "Done."

Asset packages

Recorder

☐ Shader Forge

☑ Standard Assets

☐ Unity Anima2D

☐ Unity IAP

☐ Unity Particle Pack

☐ Unity Particle Pack

Deselect all   Done
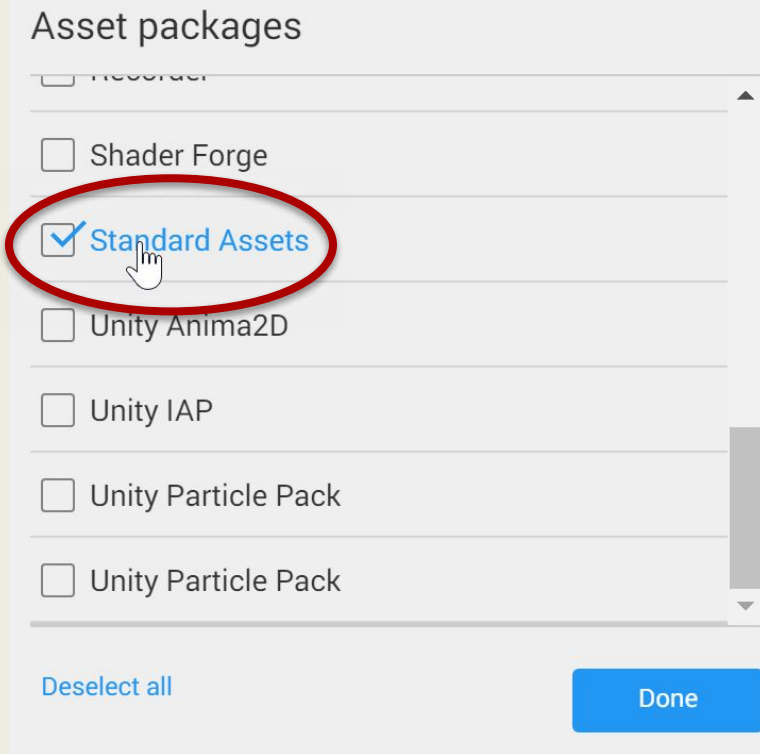
Starting a new project

Step 5: Finally, click "Create project"

Importing stuff

1. Copy all the files in **Crash-Course-Unity-2018-Assets.zip** into the project's Assets folder using your favorite file browser. The files include:
   a. **Level.fbx**
   b. **Grass.png**
   c. **Rock.jpg**
   d. **Hit.wav**
   e. **DragRigidbody.cs**
2. Switch to Unity.

Asset License

Original files obtained from:
http://opengameart.org/content/machu-picchu

- **level.fbx** is a modified version of MPFull.blend from ctdabomb, released under CC-by-sa 3.0
- **grass.png** is from samuncle, released under CC-by-sa 3.0
- **rock.jpg** is from Marianne Gagnon , release under CC-by-sa 3.0

Link to CC-by-sa 3.0 license:

https://creativecommons.org/licenses/by-sa/3.0/legalcode

About Unity
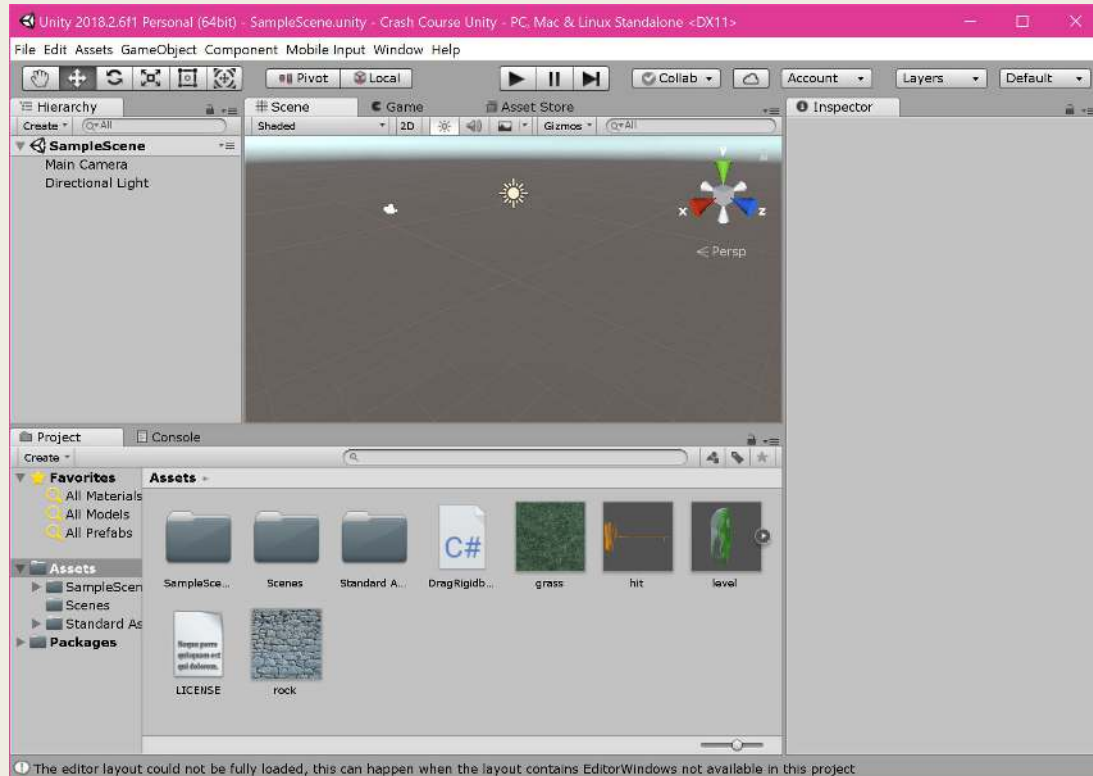
What is Unity?

- A What-You-See-Is-What-You-Get (WYSIWYG) 3D & 2D Game Engine
- Many built-in features
  - Physics, Sound, Scripting, Gamepad support, Plugins, and more!
- Builds to many platforms
  - PC, Mac, Linux, HTML5 + WebGL, iOS (iPhone + iPad), Android, Facebook Gameroom, Windows 10, PS4, Xbox One, Switch, etc.

Licenses and Fees

- Free license (what we're using now!)
  - Build to Windows, Mac, Linux, Facebook Gameroom, HTML5 + WebGL, iOS, Android, and Windows 10
  - C# scripting support
  - Totally OK to sell your game! There's no royalty fees.
    - Exception: if your company's gross revenue/budget exceeds $100,000, you need to purchase...
- If company gross revenue/budget exceeds $100,000...
  - Check out the paid license in the URL below:
  - store.unity.com
  - Plus ($35/month) and Pro ($125/month) license provides cool web-related services, e.g. servers for multiplayer.

Making a game
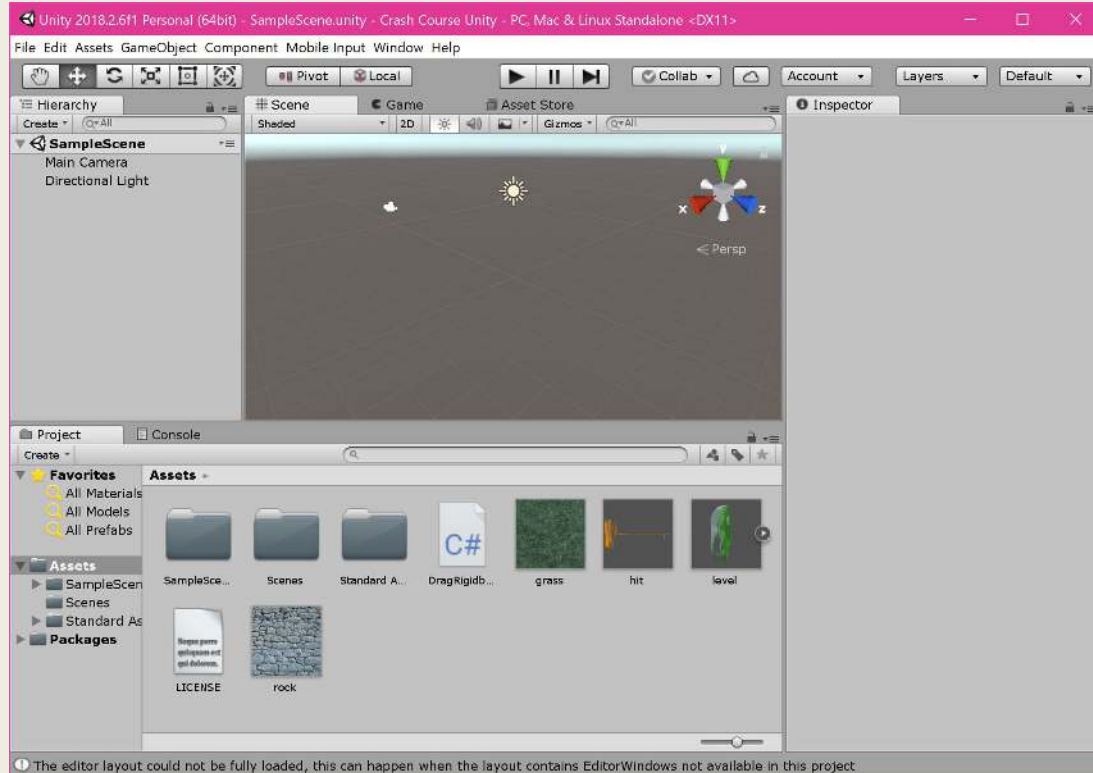
Project Pane

- Displays the contents of the Assets folder.
  - Automatically syncs with the folder if there's any changes
- Has a search bar to make it easier to find assets

Add a model to a new scene

1. Select **level.fbx** in the Project pane.
2. Drag-and-drop **level.fbx** into the Scene pane.
3. In file menu, select "File -> Save Scene as…" to save the scene!

File	Edit	Assets	GameObject	Compo
New Scene				Ctrl+N
Open Scene				Ctrl+O
Save Scene				Ctrl+S
Save Scene as…				Ctrl+Shift+S
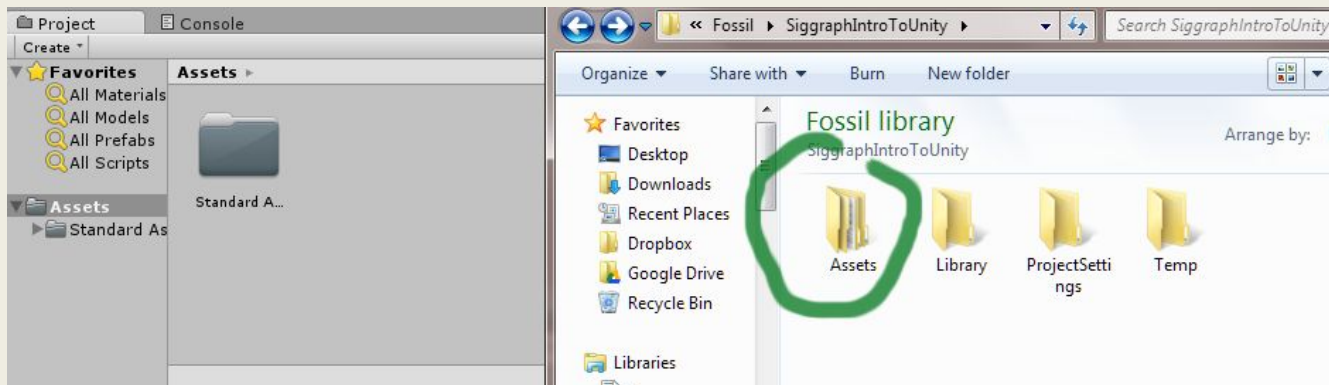
Quick lexicon review

- Assets
  - Industry-wide term for any files used in the game
  - For Unity, that's anything inside the Assets folder
- Models
  - Industry-wide term for 3D sculpture files.
  - Can contain animations.
  - Usually made in a specialized program, e.g. Maya, Blender, etc.

Scene Pane

- A 3D view of a scene where objects can be positioned, rotated, and scaled.
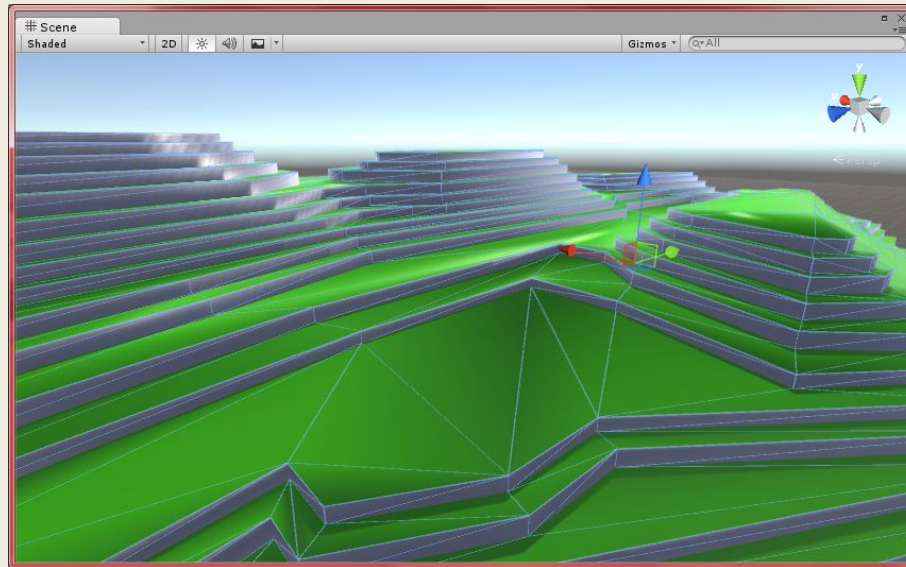
Importing 3D models

Unity can natively import:

- FBX (*.fbx)
- COLLADA (*.dae)
- 3D Studio (*.3ds)
- Wavefront (*.obj)
- Draw Interchange Files (*.dxf)

Importing 3D models

If you have the following software installed on the same computer Unity is, Unity can also import:

- Blender (*.blend)
- Maya (*.mb, *.ma)
- 3D Studio Max (*.max)
- Modo (*.lxo)
- Cinema4D
- Cheetah 3D (*.jas)
- Lightwave

Navigating the scene pane

- 2-button mouse:
  - Left-click to select objects
  - Hold alt and left mouse button to rotate camera around scene origin
  - Hold ctrl, alt, and left mouse button to pan
  - Hold ctrl, alt, and right mouse button to zoom
- 3-button mouse:
  - Left-click to select objects
  - Hold right mouse button to rotate camera around camera position
  - Scroll wheel to zoom in and out
  - Click and hold on the scroll wheel to pan
- Hold shift to pan/rotate/zoom faster

Navigating the scene pane

- **Rotation Lock**
  - Toggles whether rotation is enabled or not.
- **Axis Snap**
  - Click on any cones to snap the view to that axis.
- **Camera Toggle**
  - Toggles between perspective and orthogonal.
  - Orthogonal view creates the isometric view, i.e. the lack of a vanishing point.
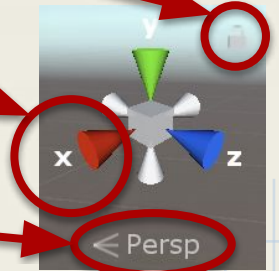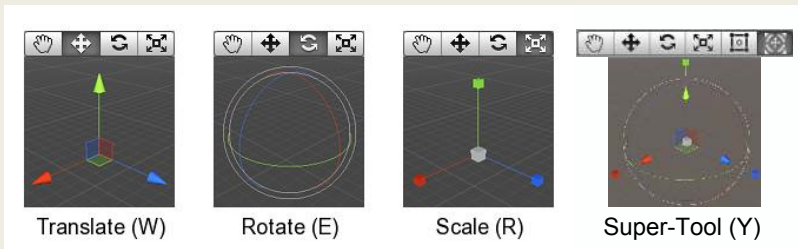
Manipulating game objects

- Object controls, from left to right:
  - Pan View (Q)
  - Translate (W)
  - Rotate (E)
  - Scale (R)
  - 2D Sprite (T)
  - Super-Tool (Y)

Translate (W)   Rotate (E)   Scale (R)   Super-Tool (Y)


- Controls to toggle object's reference point:
- Play Game controls, from left to right:
  - Play game (or if already playing, stop game)
  - Pause game (or if already paused, resume game)
  - Move forward one frame

Hierarchy Pane

- Displays the content of a scene in a tree hierarchy.
- Objects selected in the Hierarchy-pane are also selected in the Scene-pane, and vice versa.
- You can change the order of the objects by dragging them up and down
- Dragging objects into another turns that object into a child (I'll go over this later)
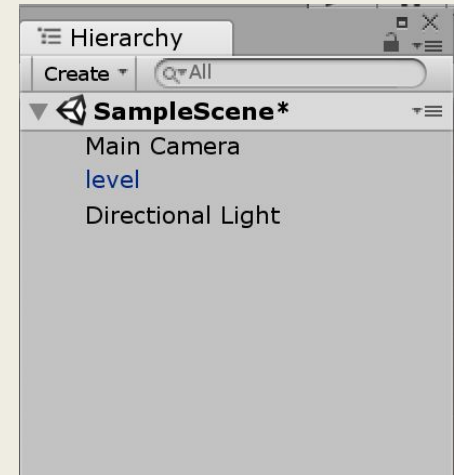
Quick lexicon review

- Game Objects
  - Unity's term for any individual object
  - Can be active or inactive
  - Every entry in the Hierarchy pane is a game object
- Scene
  - Unity's term for files storing a collection of game objects
  - Store references to assets in the Assets folder
  - Has a *.unity file extension
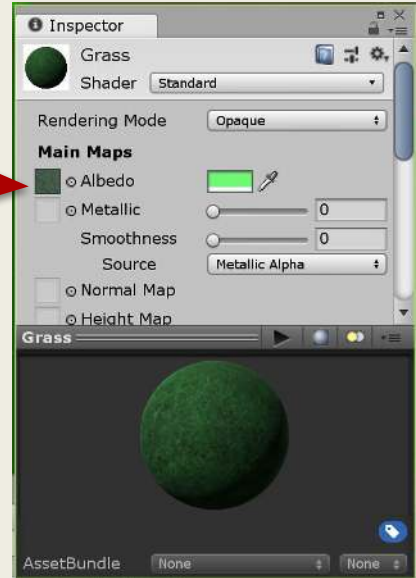
Creating materials

1. Select **Create** in the Project pane.
   a. Note: right-clicking in the Project pane brings up the same context menu.
2. Select **Material**.
3. Change the name of the material to "Grass".
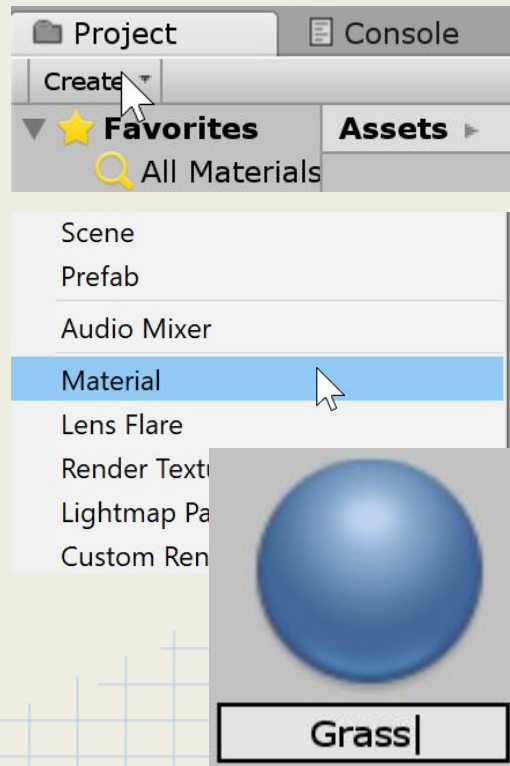
Adding Texture to Material

1. In the Inspector pane, drag-and-drop texture **grass.png** into material **Grass' Albedo field**.
2. Reduce the **Metallic** and **Smoothness field** to **0**.
3. Feel free to adjust the color field next to the **Albedo** field to one's favorite color!

Applying Material to Model

1. From the Project pane, drag-and-drop the **Grass** material to the **level** model in the Scene pane.
2. Create a new material, **Rock**, and repeat the process with texture **rock.jpg**.
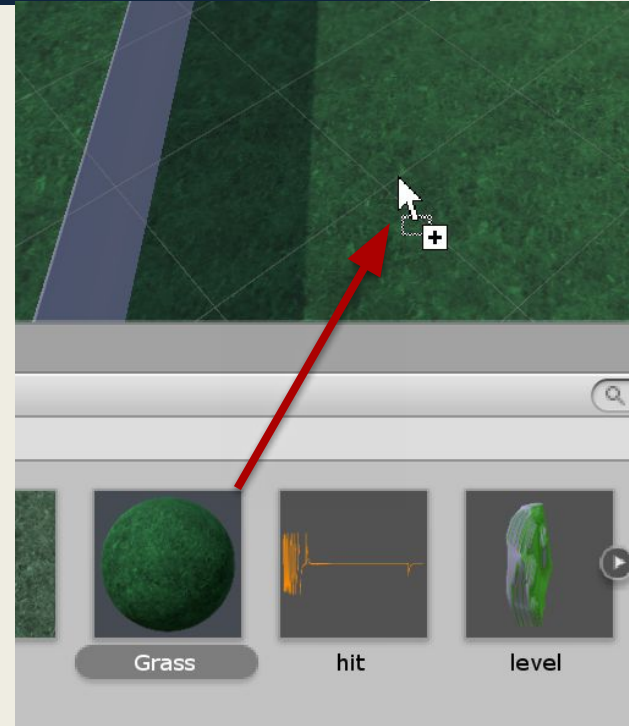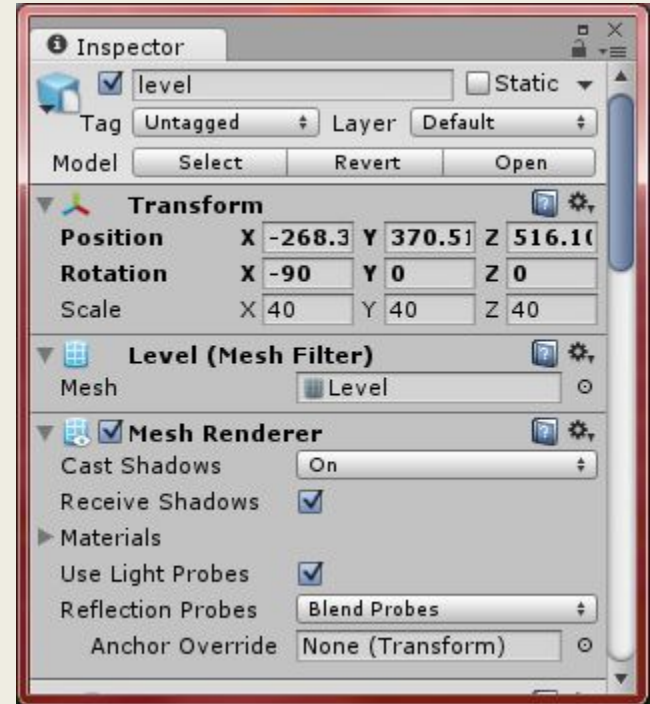
Quick lexicon review

- Material
  - Industry-wide term for what material the surface of a model is supposed to look like (e.g. metal, plastic, non-shiny stuff)
  - In Unity, materials are files (*.mat) shared between game objects
    - Changing a material's properties will update all game objects with the same material
  - Models retain information on how materials are mapped to its surface
  - Normally, you have to create materials by hand
    - Unity conveniently made 2 for us!
  - The Renderer component holds materials
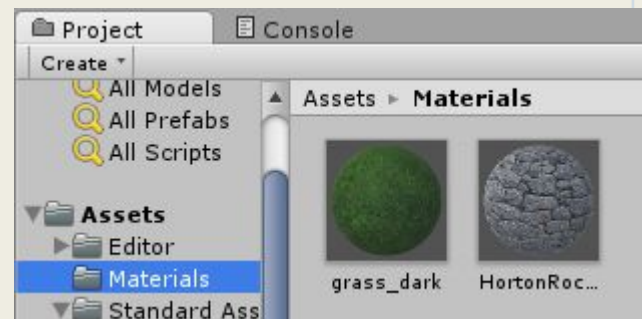
Inspector Pane

- Displays the properties and details on a selected object/file, both in Project pane and Scene pane.
- One can edit the properties of an object here.

Importing Images

Unity can natively import:
- Photoshop (*.psd)
- PNGs (*.png)
- JPEGs (*.jpg, *.jpeg)
- Un-animated GIFs (*.gif)
- Paint (*.bmp)
- TGAs (*.tga)
- and more!

Quick lexicon review

- Textures
  - Industry-wide term for images that represents how a model is supposed to be painted
  - Models usually contains coordinates (called UV) that indicate how a texture is supposed to be mapped on the model
- Components
  - Unity's term for containers with specialized information
  - Game objects retain a list of components (such as Transform)
  - Can be enabled or disabled
  - Every entry in the Inspector pane is a Component

Adding Camera

Let's add a prefab (short for prefabricated object) with first-person controls
1. In the Hierarchy pane, select "Main Camera", and remove it by pressing Delete/Cmd+Backspace.
2. In the Projects pane, search for "FPSController"
3. Drag-and-drop "FPSController" into the Scene pane.
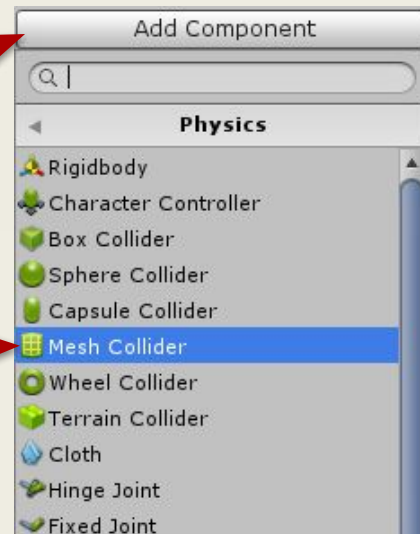4. Position the FPSController above the ground
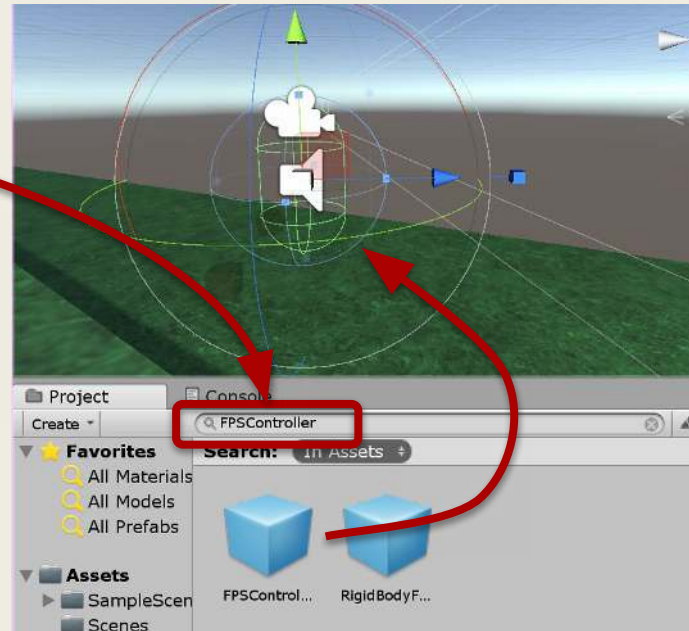
Playing the Game

1. Press the play button.
2. Observe your camera…fall through the floor.
3. Press the play button to stop the game.

4. What are we missing?

Adding a Collider

The ground needs a collider
1. In Scene pane, select **level**.
2. In the Inspector pane, click "Add Component"
3. Select "Physics -> Mesh Collider"

Playing the Game

1. Press the play button.
2. Use the mouse to look around, arrow keys (or WASD) to move, space to jump, and shift to run.
3. Marvel at your own work.

Quick lexicon review

- Prefab
  - Unity's term for prefabricated objects
  - Files with *.prefab file extension
  - Allows copying a group of game objects from one scene to another
- Colliders
  - Industry-wide term for shapes representing the boundaries of an object
  - Used by the physics engine to determine where objects collide

Types of colliders

- Mesh Colliders
  - Collider that is the shape of a model's surface
  - Expensive and inefficient
  - Best for static, non-interactable levels and objects
- Box Colliders
  - Rectangle-shaped colliders
- Sphere Colliders
  - Sphere-shaped colliders (no oval support)
- Capsule Colliders
  - Capsule-shaped colliders (supports height, no oval support)

Adding interactive stuff

1. Select "3D Object -> Cube"
2. In the Scene pane, position the new cube in front of the camera.
3. While leaving the "Cube" selected, click "Add Component" under the Inspector pane.
4. Select "Physics -> Rigidbody"
5. Play the game!
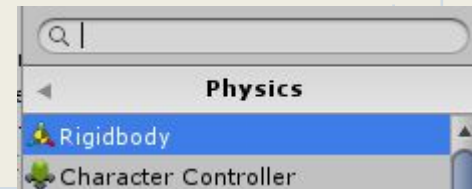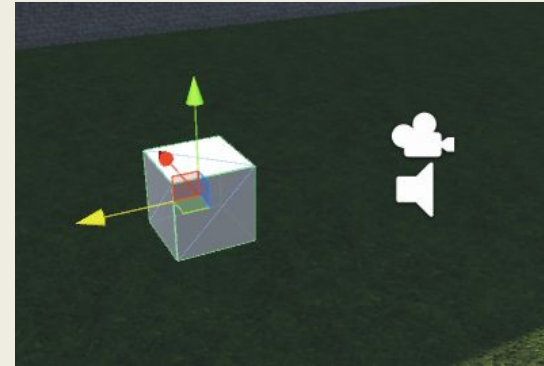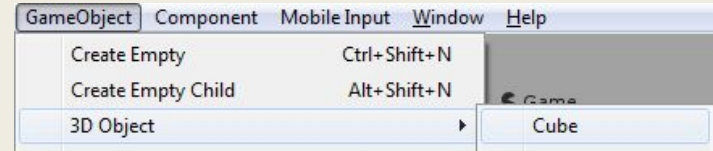
Using a Script

1. In the Project pane, select "DragRigidbody.cs"
2. Drag-and-Drop DragRigidbody.cs onto the "FPSController" under the Hierarchy pane (NOT the Scene pane!).
3. Play the game!
4. Look at the Cube, and click+hold the left mouse button to pick it up.

Making weird shapes

1. Select "3D Object -> Sphere"
2. In the Scene pane, position the new sphere on an edge of a cube.
3. In the Hierarchy pane, drag & drop the Sphere into the Cube.
4. The Sphere should now be a child of Cube.
5. Play the game, and drag around the cube+sphere!
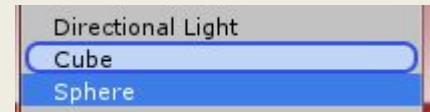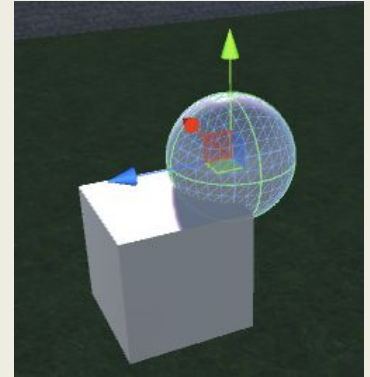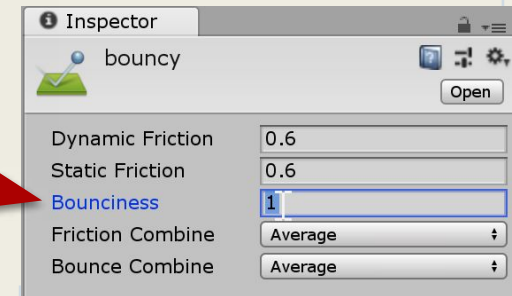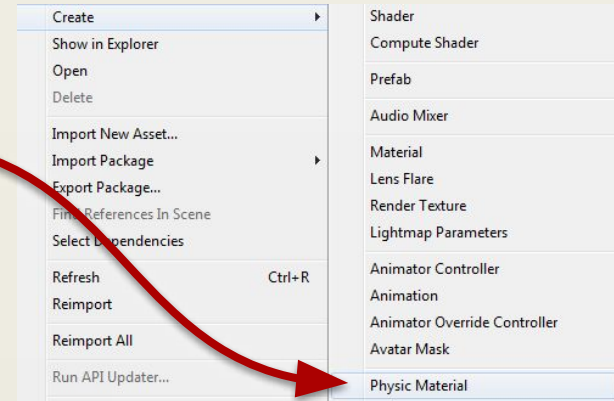
Quick lexicon review

- Rigid Body
  - Industry-wide term for a moving and/or interactive physics objects
  - Contains information such as mass, drag, and center-of-gravity
  - Turns a group of colliders (including those in the children) into a single, interactable shape
- Child
  - Industry-wide term for an object whose position, scale, and rotation follows that of another object: the parent
  - In Unity, they appear as nested entries in the Hierarchy tree view
  - Trivia: this "following parent" calculation process is known as forward kinematics, a term in 3D animations

Change some physics

1. Right-click the Project pane, and select "Create -> Physics Material"
2. Name the file, "bouncy"
3. In the Inspector pane, change the bounciness to 1 (as in, 100%)

Change some physics

1. In the Hierarchy pane, select Cube
2. Drag & drop "bouncy" from the Project pane to Material field under the Box Collider component in the Inspector pane
3. Play the game, and throw the Cube end into the ground
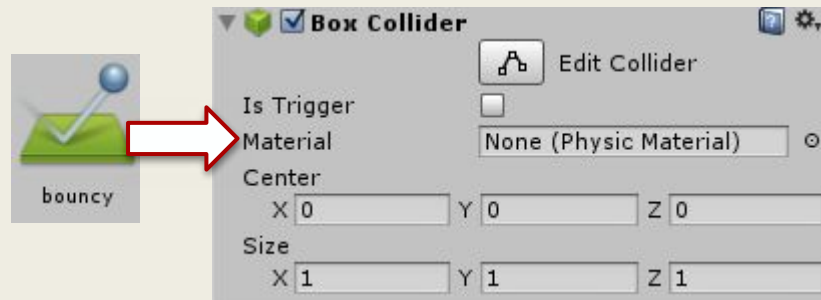
Quick lexicon review

- Physics Material
  - Industry-wide term for how objects are supposed to interact to a collider
    - Adjusts how slippery and bouncy an object can be
  - In Unity, materials are files (*.physicMaterial) shared between game objects
  - Changing a physics material's properties will update all game objects with the same physics material

Adding a Sound

1. Select the Cube in the Scene pane
2. In the Inspector pane, click "Add Component"
3. Select "Audio -> Audio Source"
4. In the Project pane, drag-and-drop **hit.wav** into the Audio Source component's "Audio Clip" property
5. Play the game!
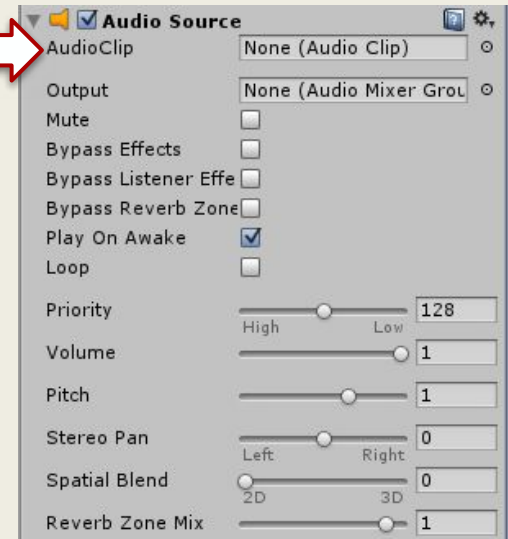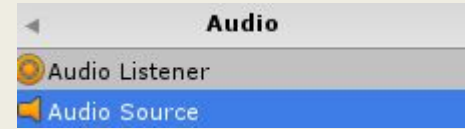
Adding a Script

1. Select the Cube in the Scene pane
2. In the Inspector pane, click "Add Component"
3. Select "New Script"
4. Change the script type to CSharp, and the script name to "PlaySoundOnCollision"
5. Click "Create and Add"
6. Double-click "PlaySoundOnCollision" in the Inspector pane to open Editor
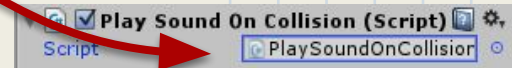
Copy the Following:

```
using UnityEngine;
public class PlaySoundOnCollision : MonoBehaviour {
    AudioSource audio;
    void Start () {
        audio = GetComponent<AudioSource>();
    }
    void OnCollisionEnter(Collision info) {
        audio.Stop();
        audio.Play();
    }
}
```

Finishing the Sound Effect

1. Save the script (under "File" in the menu bar)
2. Switch to Unity
3. Select the cube in the Scene pane
4. Under the Inspector, *uncheck* Audio Source's "Play-On Awake"
5. Play the game!

Script Summary

```
AudioSource audio;
void Start () {
    audio = GetComponent<AudioSource>();
}
```

1. The `Start()` function runs when the game starts
2. `GetComponent<AudioSource>()` gets the Audio Source component from the Game Object this script is attached to
3. `audio = GetComponent<AudioSource>()` stores the Audio Source Component in a variable, `audio`

Script Summary

```
void OnCollisionEnter(Collision info) {
    audio.Stop();
    audio.Play();
}
```

1. The `OnCollisionEnter()` function runs when the Rigidbody collides with a collider
2. `audio.Stop()` makes the sound effect stop, resetting it back from the beginning
3. `audio.Play()` makes the sound effect play again

Importing Sound

Unity can natively import:
- WAV (*.wav)
  - Best for short sound effects
- AIFF (*.aif, *.aiff)
  - Best for short sound effects
- MP3 (*.mp3)
  - Best for music, especially in mobile devices
  - Remember, some mobile devices can only play one MP3 file at once
- OGG (*.ogg)
 - Best for music, especially PC and consoles

Duplicating the Cube

1. Drag & Drop the object, "Cube" from the Hierarchy pane to the Project pane. This creates a new Prefab.
2. Drag & Drop the Cube prefab from the Project pane to the Scene pane as many times as you like. This will create many copies of Cube.
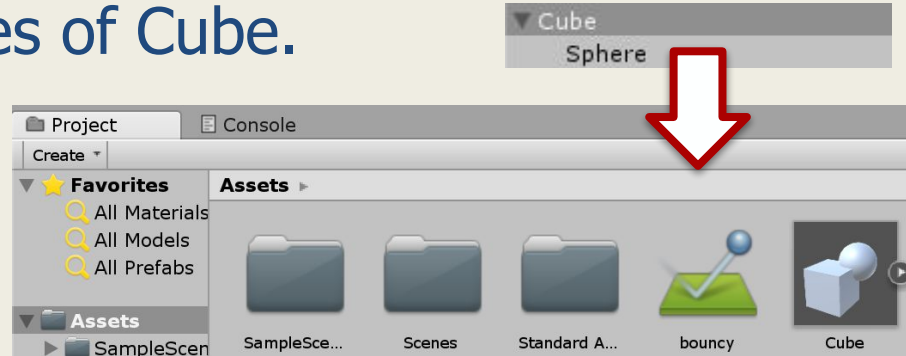
Importing Normal Maps

1. In the Project pane, select **grass.png**, and press Ctrl+D/Cmd+D to duplicate it.
2. Click on **grass 1.png**.
3. In the Inspector, change the Texture Type to "Normal map".
4. Check "Create from Grayscale".
5. Adjust the Bumpiness to a smaller value.
6. Click "Apply".
7. Do the same thing for **rock.jpg**.
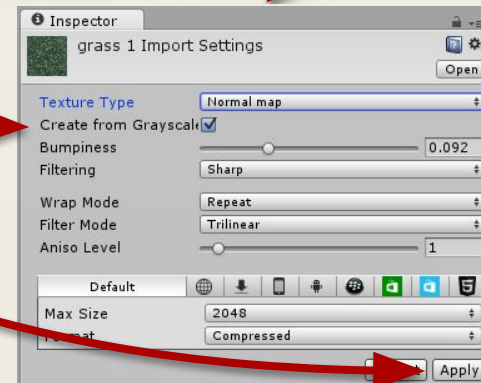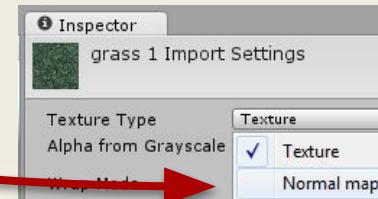
Adding Normal Maps

1. Select **level** in the Hierarchy pane again.
2. In the Inspector pane, drag & drop **rock 1.png** into the **Rock's Normal Map**.
3. Do the same thing for **grass 1.jpg** with **Grass' Normal Map**.
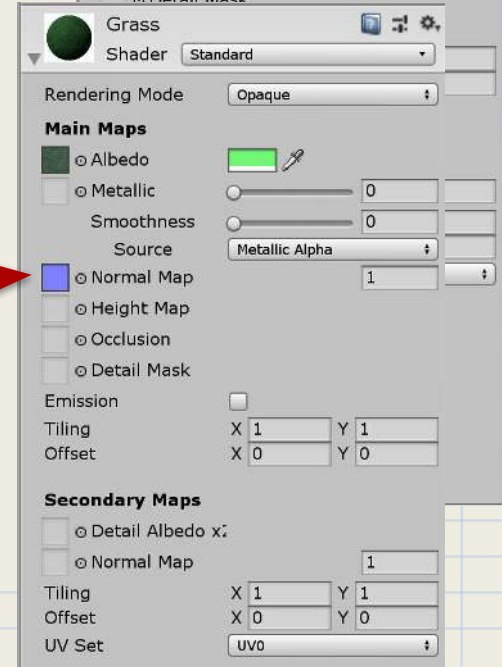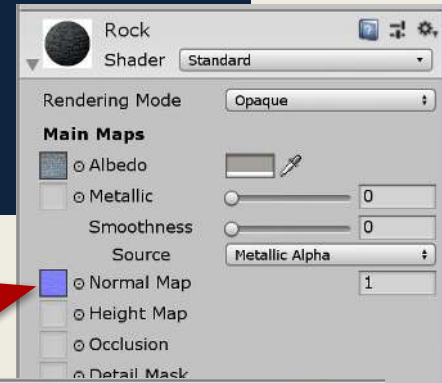
Quick lexicon review

- Normal Map
  - Industry-wide term for images that represents the direction light is supposed to reflect off of a model's surface
  - Model's UV-coordinates indicate how a normal map is supposed to be mapped on the model

Adjust lighting

1. Click on "Directional Light" game object under the Hierarchy pane.
2. Adjust the rotation in the Scene pane.
   a. Quick-tip: hover the mouse on the Scene pane, then tap F to focus on the selected object
3. Change the color and intensity of the light in the Inspector pane.

Note: if the lighting doesn't change in the Scene pane, make sure the lighting button is pressed
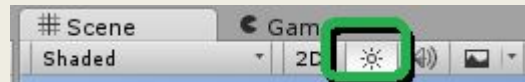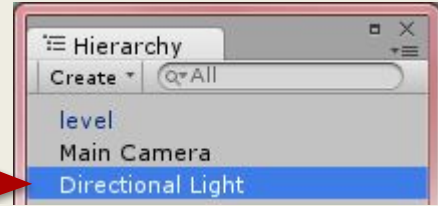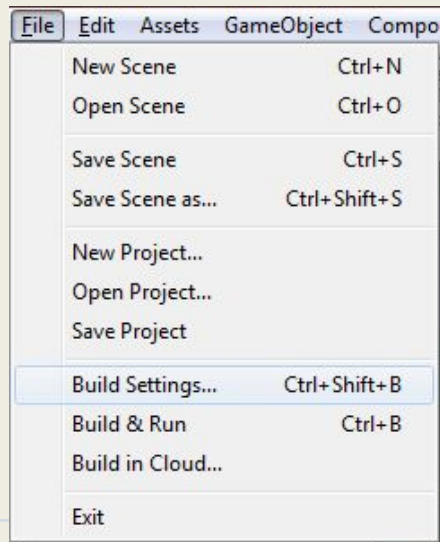
About Lighting

- Create new lights with "GameObject -> Light"

- Directional Light
  - A sunlight emitted in one direction
- Point Light
  - A glow emitting from a single point
- Spotlight
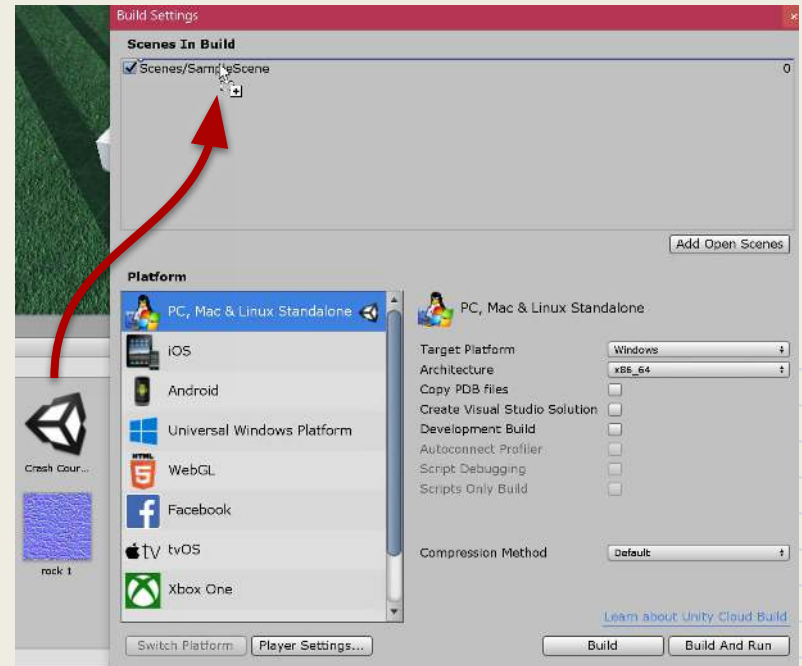 - A cone-shaped light used to simulate flashlights and spotlights
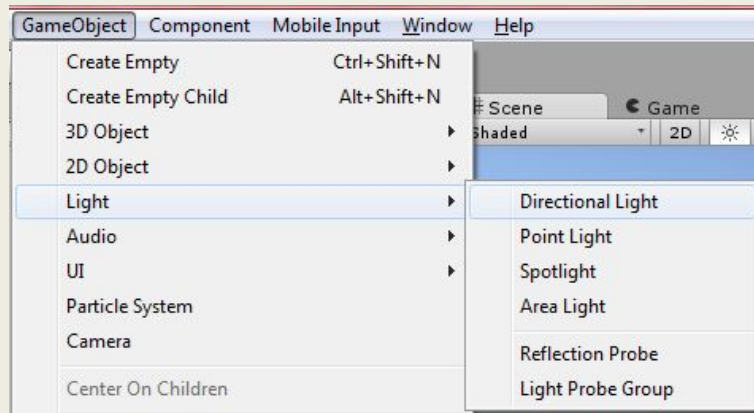
Building an Executable

1. Save the scene with Ctrl+S/Cmd+S.
2. In the file menu, select "File -> Build Settings..."

Building an Executable

1. Drag & drop your scene in the Project pane into the Build dialog
2. Highlight "scenes/SampleScene", and hit the delete key.

Building an Executable

3. Change the Target Platform to your computer's OS
4. Click the "Build" button, and select a folder that *isn't* in your project

Save project

Select "File -> Save Project"
- Saves project settings, such as Build Settings
- Saves anything import settings in the Project Pane
- Saves any Unity files that isn't a scene, such as materials, prefabs, physics materials, etc.
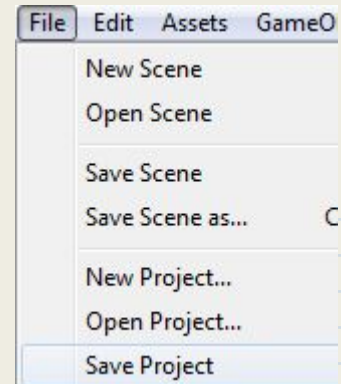
Congratulations!

Any questions?

Supplementary materials

- http://unity3d.com/learn/tutorials
  a. Official site full of tutorials on individual Unity feature
  b. Includes in-depth C# programming tutorials!