



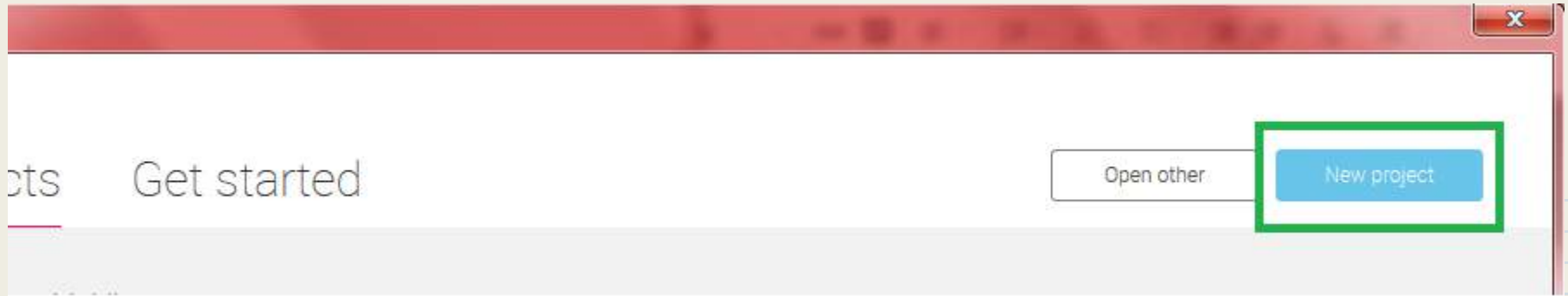
# Crash Course Unity 5

Introduction to how to use Unity 5



# Starting a new project

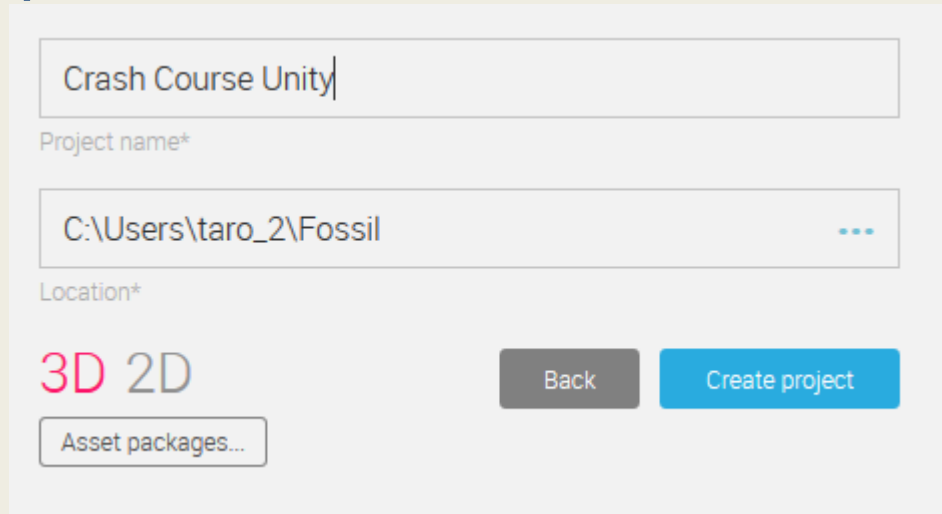
Step 1: Click "New Project"



# 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*



The screenshot shows the Unity 'Create New Project' dialog box. It features two input fields: 'Project name\*' with the text 'Crash Course Unity' and 'Location\*' with the path 'C:\Users\taro\_2\Fossil'. Below the fields are radio buttons for '3D' (selected) and '2D'. At the bottom, there is an 'Asset packages...' button, a 'Back' button, and a 'Create project' button.

Crash Course Unity

Project name\*

C:\Users\taro\_2\Fossil

Location\*

3D 2D

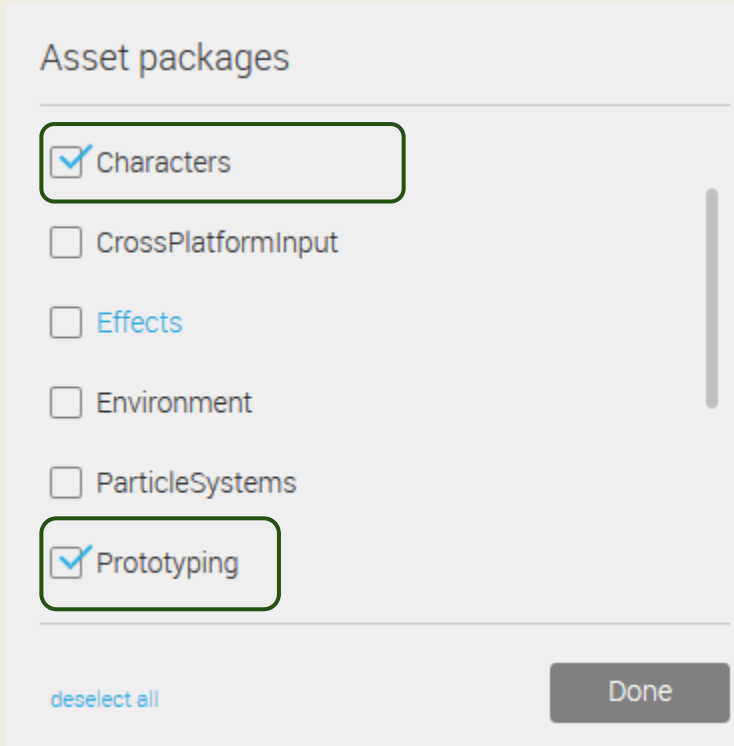
Asset packages...

Back Create project

# Starting a new project

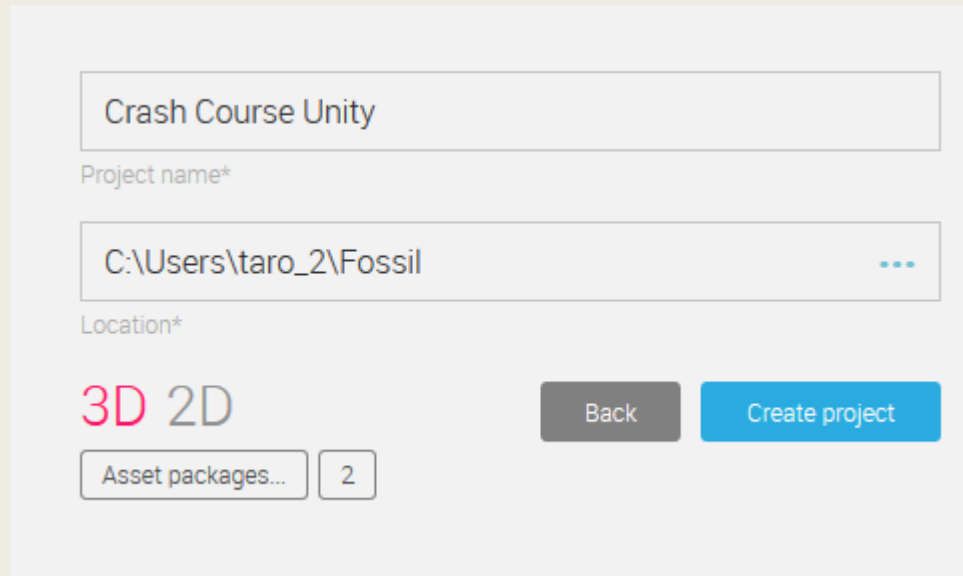
Step 4: Click “Asset packages...”, and check “Character” and “Prototyping.” Lastly, click “Done.”

*Note: the rest of the packages can be imported at any time!*



# Starting a new project

Step 5: Finally, click “Create project”



Crash Course Unity

Project name\*

C:\Users\taro\_2\Fossil

Location\*

3D 2D

Asset packages... 2

Back Create project

The screenshot shows a project creation interface. At the top, there is a text input field containing "Crash Course Unity". Below it is the label "Project name\*". The next field is a file path "C:\Users\taro\_2\Fossil" with a blue ellipsis icon to its right. Below this is the label "Location\*". There are two radio buttons, "3D" (which is selected and highlighted in red) and "2D". At the bottom left, there is a text input field "Asset packages..." followed by a small box containing the number "2". On the right side, there are two buttons: a grey "Back" button and a blue "Create project" button.

# Importing stuff

1. Go to <http://wp.me/a5G4dR-8Z>, and download "Crash Course Unity 5 Assets.zip"
2. Unzip the file.
3. Open your favorite file browser.
4. From the unzipped files, move the 3D model **level.fbx**, textures (images) **grass.png** and **rock.jpg**, sound effect **hit.wav**, and script **DragRigidbody.js** into the project's Assets folder.
5. Switch to Unity.

# Asset License

Original files obtained from:

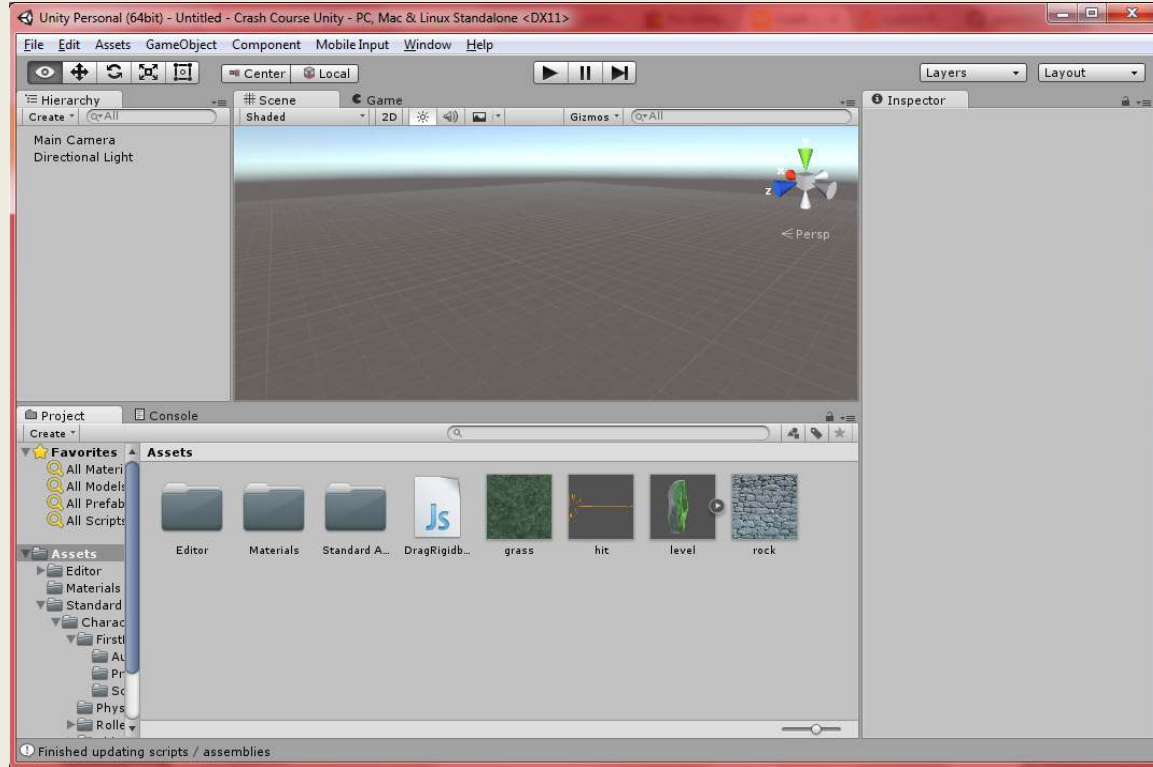
<http://opengameart.org/content/machu-picchu>

- **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





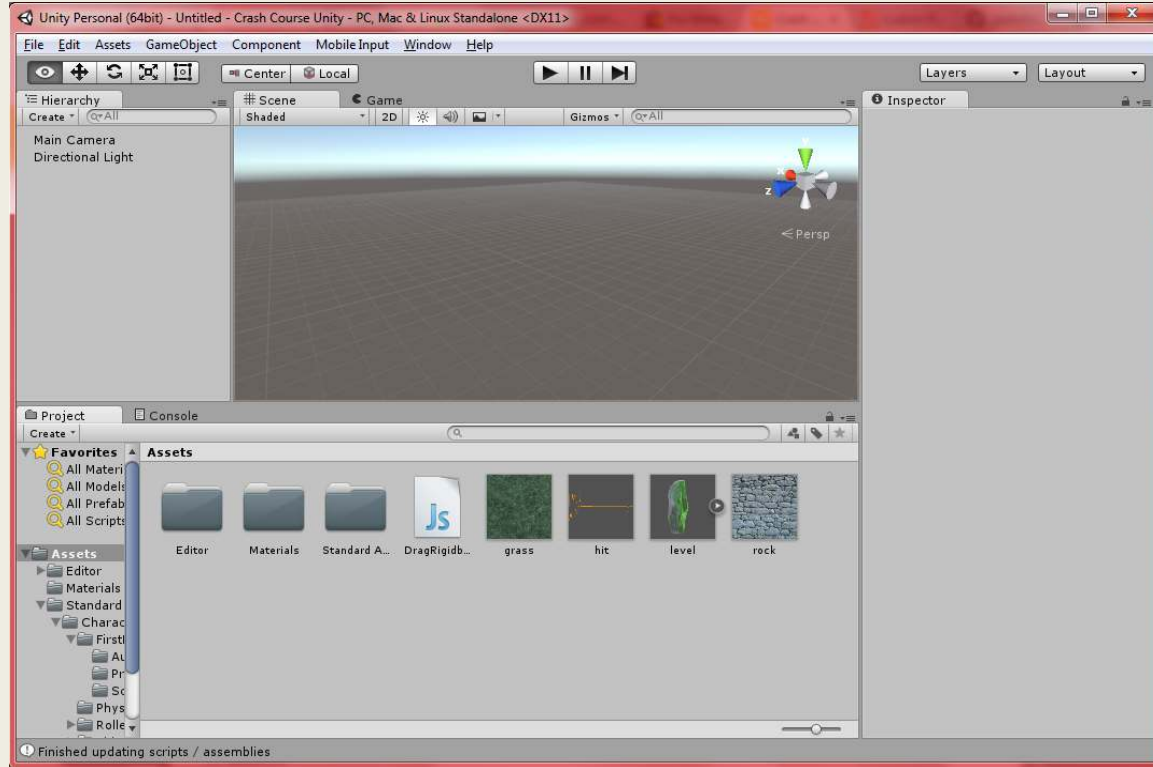
# What is Unity?

- 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, Webplayer, HTML5, iPhone, iPod, iPad, Android, Kindle Fire, Barnes & Noble Nook, Windows 8, Blackberry, Wii U, New 3DS, PS4, PS3, PS VITA, Xbox One, Xbox 360, Ouya, Samsung TV

# Licenses and Fees

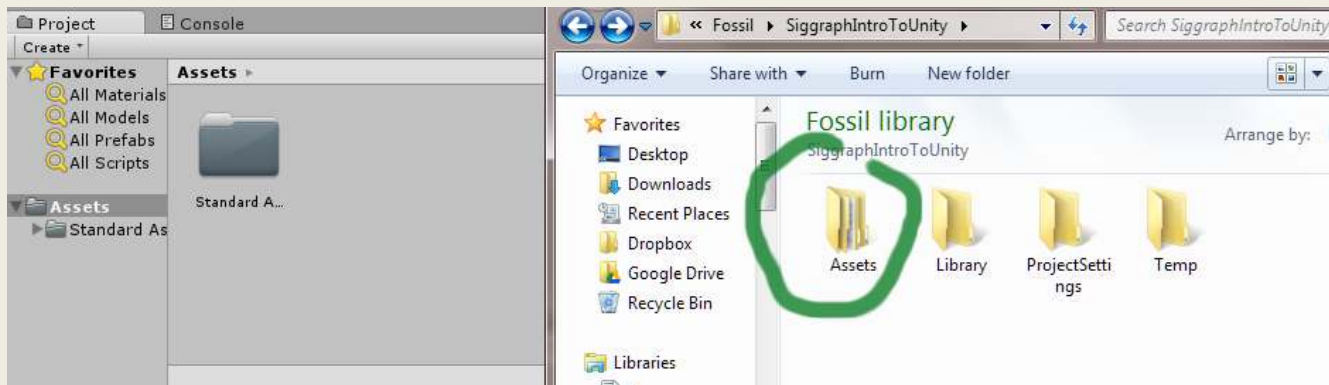
- Free license (\$0) (what we're using now!)
  - Build to Windows, Mac, Linux, Webplayer, HTML5, iOS, Android, Blackberry, Windows App (Metro), and Windows 8 Phone
  - C# and Javascript(-ish) scripting support
  - Totally OK to sell your game! There's no royalty fees.
    - Unless your company's gross revenue/budget exceeds \$100,000, in which you need to purchase...
- Pro license (\$1,500 or \$75/month)
  - Required if company gross revenue/budget exceeds \$100,000
  - Supports up to 2 computers per license
  - Access to more platforms (Wii U, PS4, Xbox One, etc.)
  - Allow customizing splash screen

# Making a game



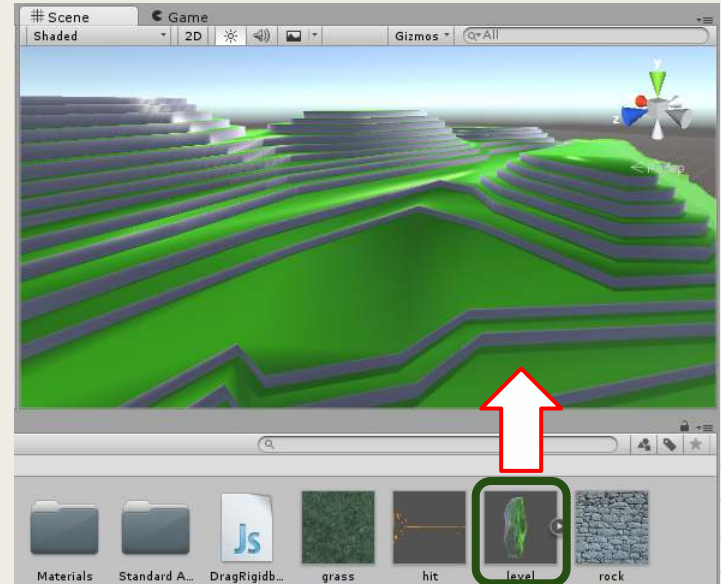
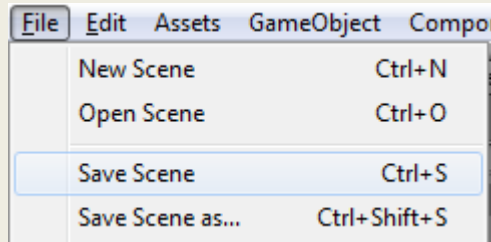
# Project Pane

- 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. Press Ctrl+S/Cmd+S to save the scene (or "File->Save Scene")



# Quick lexicon review

- **Assets**

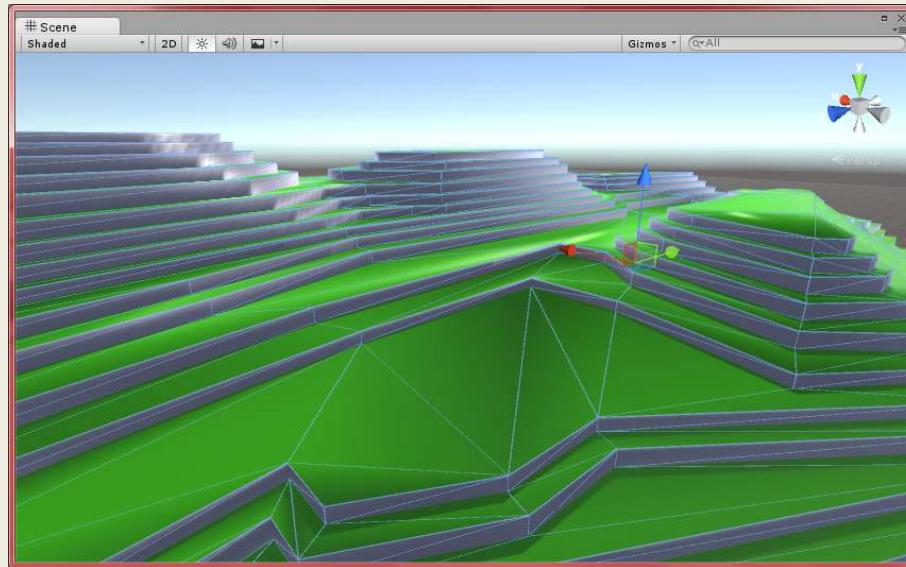
- Industry-wide term for any files used in the game
- For Unity, that's anything inside the Assets folder

- **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:

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

# Navigating the scene pane

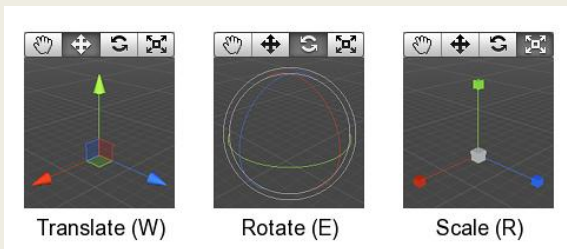
- **3-button mouse:**
  - Left-click to select objects
  - Hold right mouse button to rotate camera around camera position
  - Scroll wheel to zoom in and out
  - Click and hold on the scroll wheel to pan
- **2-button mouse:**
  - Left-click to select objects
  - Hold alt and left mouse button to rotate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

# Manipulating game objects

- Object controls, from left to right:



- Pan View (Q)
- Translate (W)
- Rotate (E)
- Scale (R)
- 2D Sprite (T)



- 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)

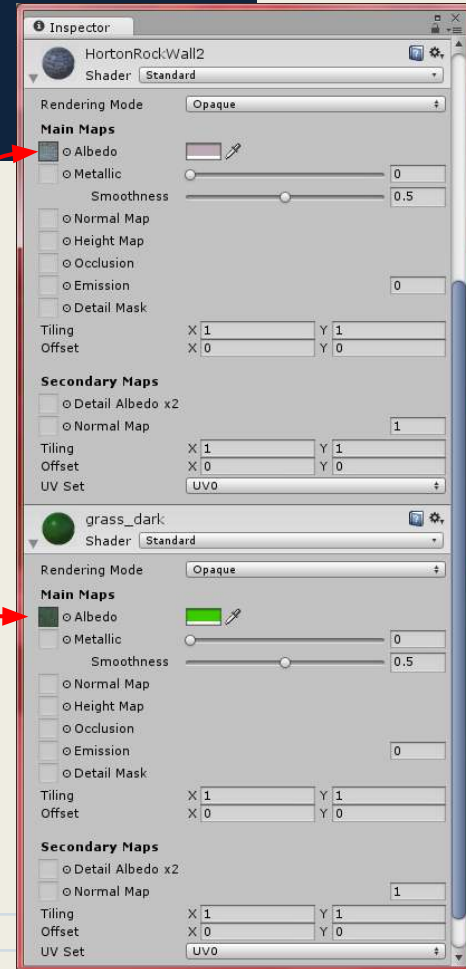


# Quick lexicon review

- **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

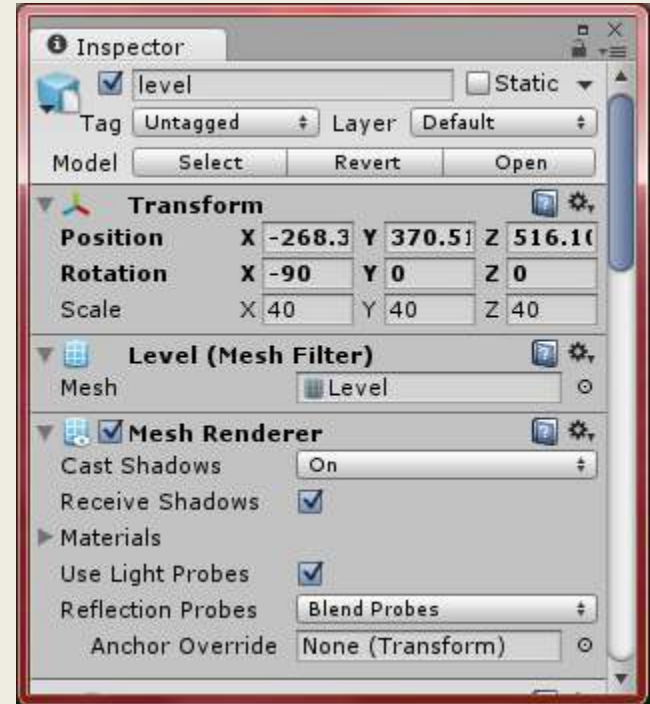
# Adding Texture to Model

1. Select **level** in the Hierarchy pane
2. In the Inspector pane, drag-and-drop texture **rock.jpg** into material **HortonRockWall2's Albedo field**, and **grass.png** into material **grass\_dark's Albedo field**.



# 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!



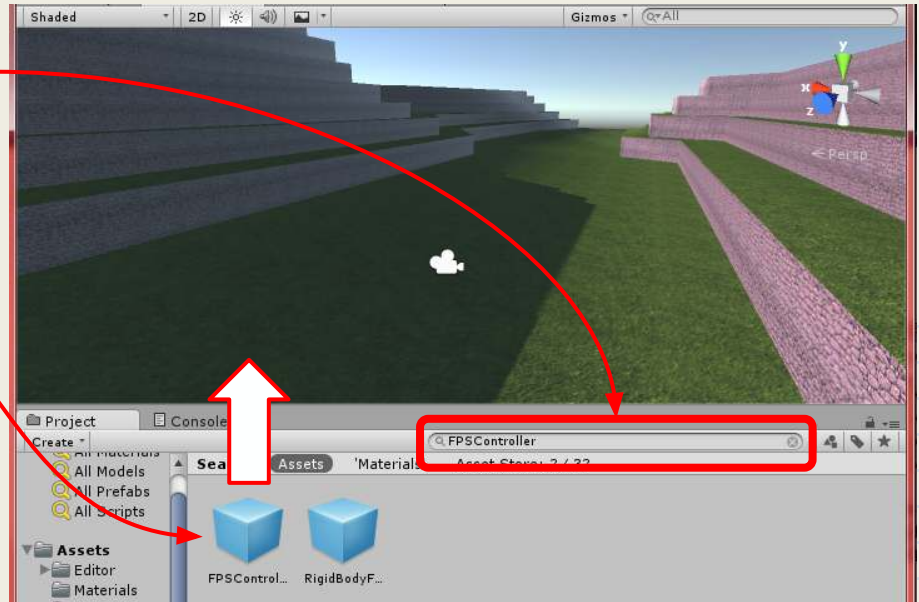
# Quick lexicon review

- 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

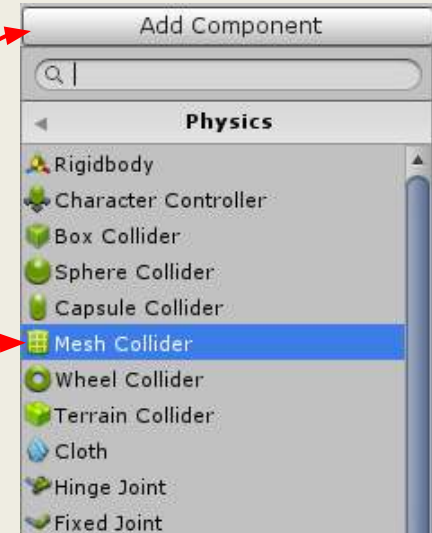
1. Press the play button.
2. Observe your camera...fall through the floor.
3. Press the play button to stop the game.
4. What are we missing?



# 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

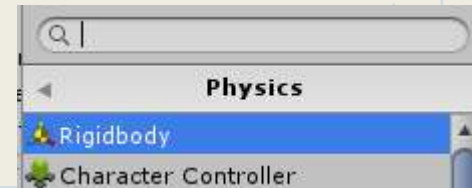
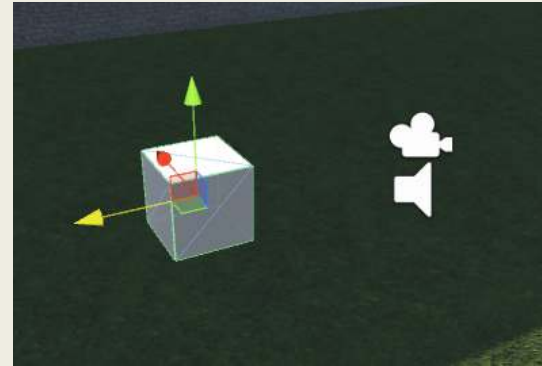
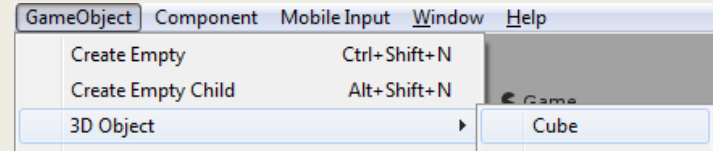
- Prefab
  - Unity's term for prefabricated objects
  - Files with \*.prefab file extension
  - Allows copying a group of game objects from one scene to another
- Colliders
  - Industry-wide term for shapes representing the boundaries of an object
  - Used by the physics engine to determine where objects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



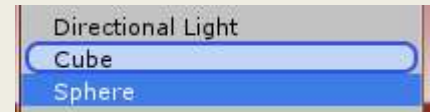
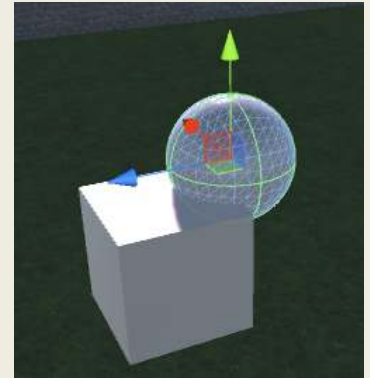


# Using a Script

1. In the Project pane, select "DragRigidbody.js"
2. Drag-and-Drop DragRigidbody.js onto the "FPSController" under the Hierarchy pane (NOT the Scene pane!).
3. Play the game!
4. Move the cursor to the Cube, and click+hold on it 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!

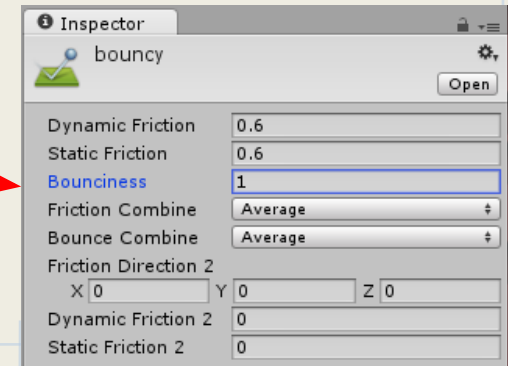
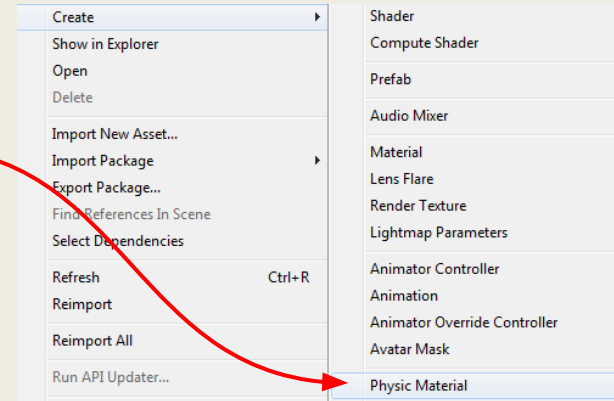


# Quick lexicon review

- Rigid Body
  - Industry-wide term for an 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's tree view

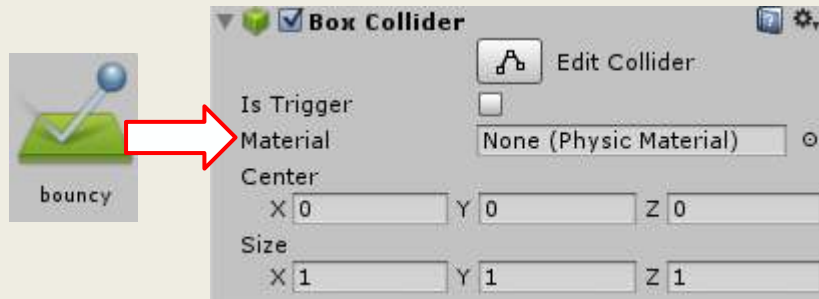
# Change some physics

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

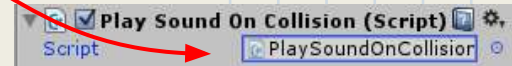
# Adding a Sound

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 (Ctrl + S)
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

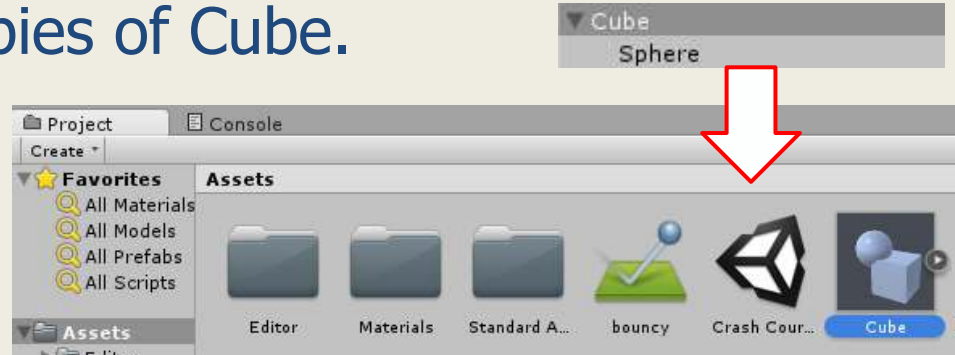
# Importing Sound

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

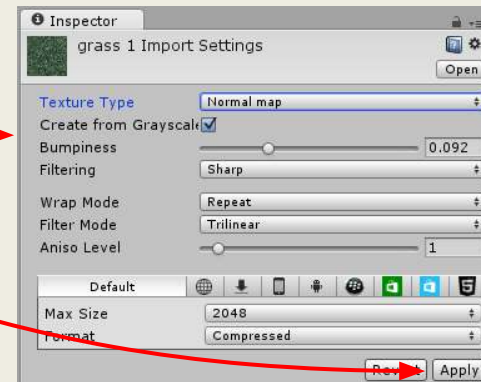
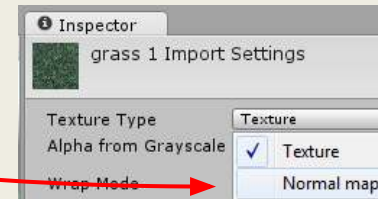
# Duplicating the Cube

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.



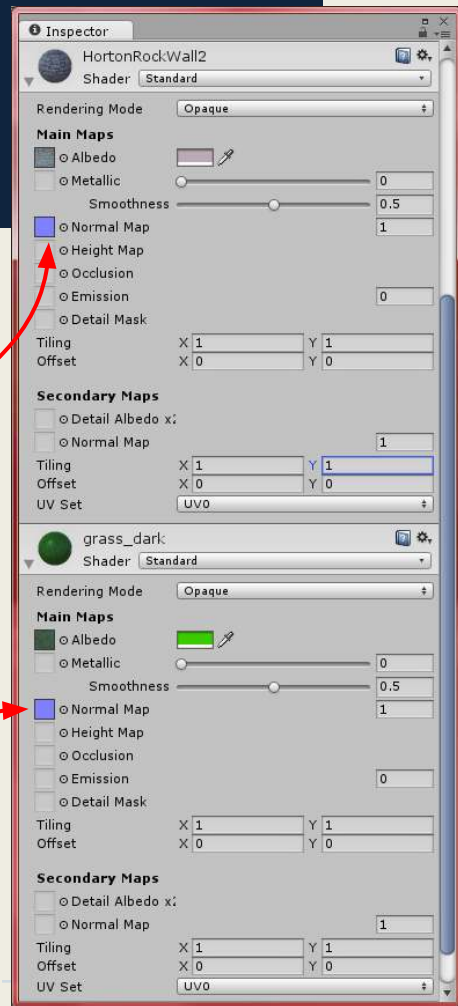
# Importing Normal Maps

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

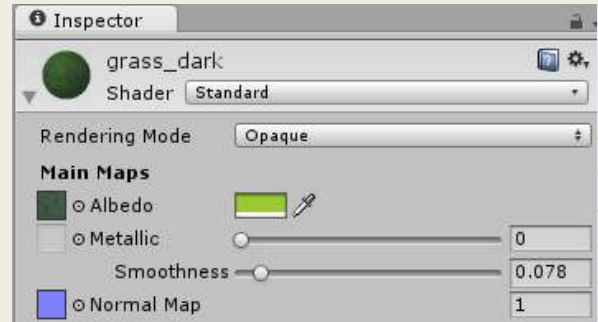
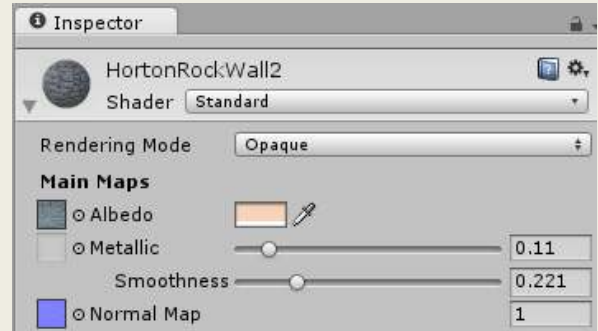
1. Select **level** in the Hierarchy pane again.
2. In the Inspector pane, drag & drop **rock 1.png** into the **HortonRockWall2's Normal Map**.
3. Do the same thing for **grass 1.jpg** with **grass\_dark's Normal Map**.





# Adjust materials

- Adjust the **Metallic** (reflectiveness) and **Smoothness** (shininess) values under the **HortonRockWall2** and **grass\_dark**.
- Experiment with the **color** (next to Albedo) by clicking on it, and selecting a color in the color picker



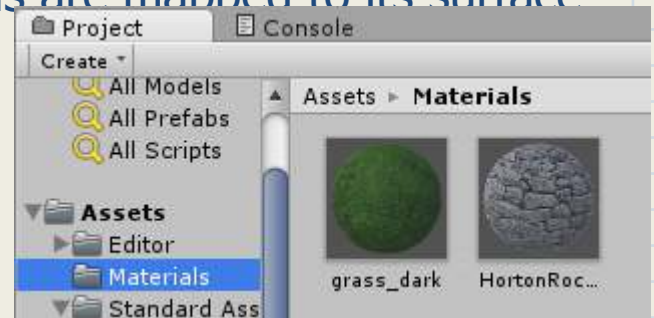
# Quick lexicon review

- 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

# Quick lexicon review

- Material

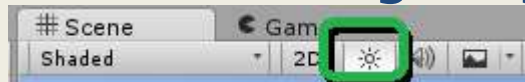
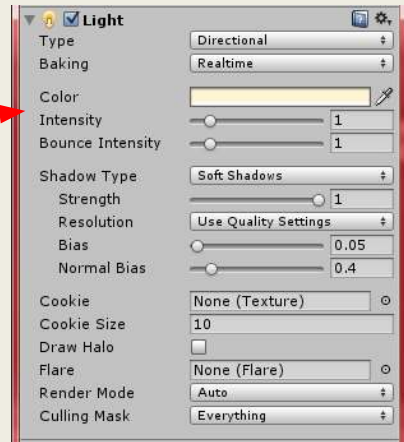
- Industry-wide term for what material the surface of a model is supposed to look like (e.g. metal, plastic, non-shiny stuff)
- In Unity, materials are files (\*.mat) shared between game objects
  - Changing a material's properties will update all game objects with the same material
- Models retain information 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



# Adjust lighting

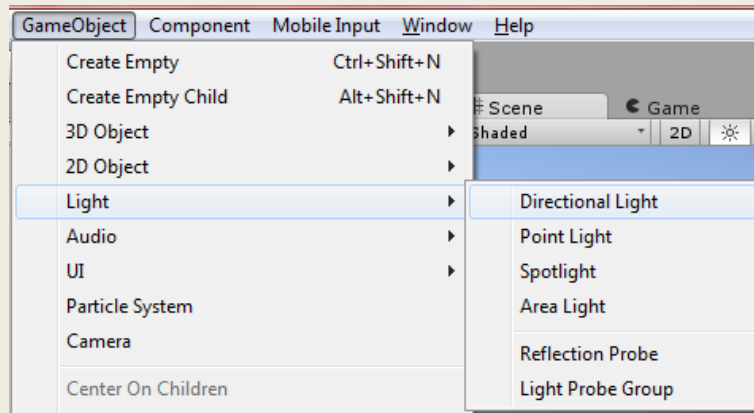
1. Click on "Directional Light" game object under the Hierarchy pane.
2. Adjust the rotation in the Scene pane.
  - a. Quick-tip: hover the mouse on the Scene pane, then tap F to focus on the selected object
3. Change the color and intensity of the light in the Inspector pane.

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



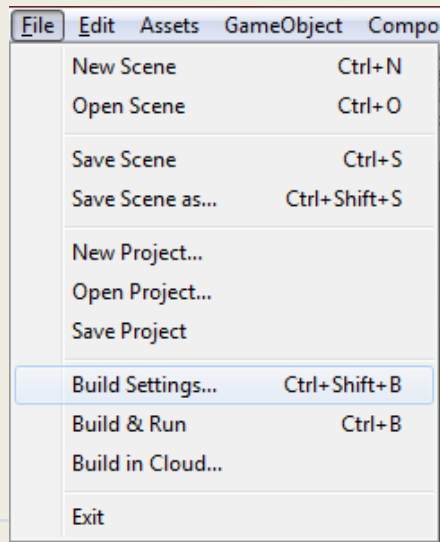
# About Lighting

- 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



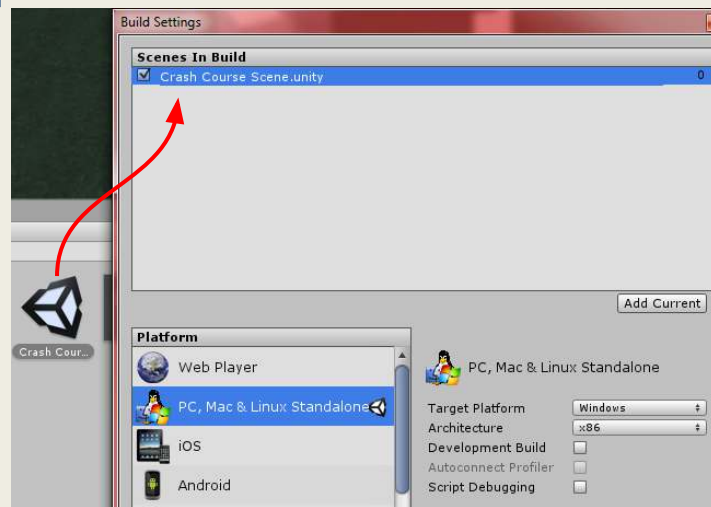
# Building an Executable

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



# Building an Executable

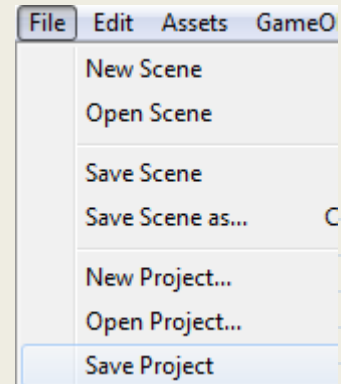
1. Drag & drop your scene in the Project pane into the Build dialog
2. Change the Target Platform to your computer's OS
3. Click the "Build" button, and select a folder that *isn't* in your project



# 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?

