Crash Course Unity 2D

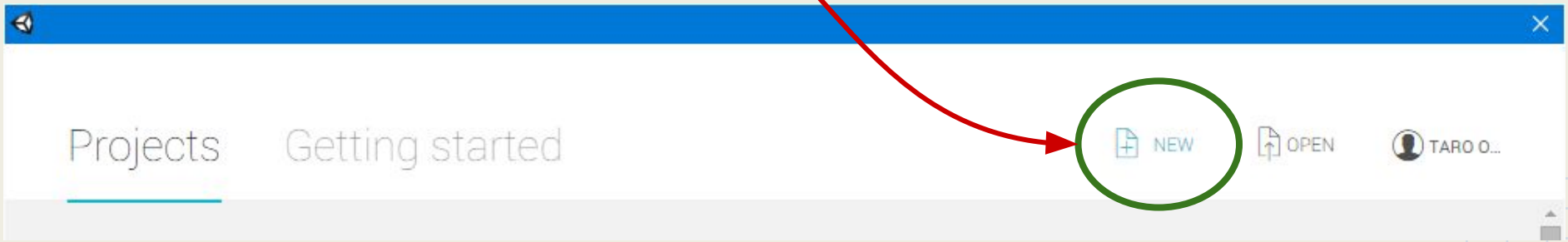
Building 2D platformers in Unity

Goal

- Get comfortable with Unity game editor
- Create an interactive 2D environment
- Learn lots of 2D development terms
- A brief start in C# programming

Starting a new project

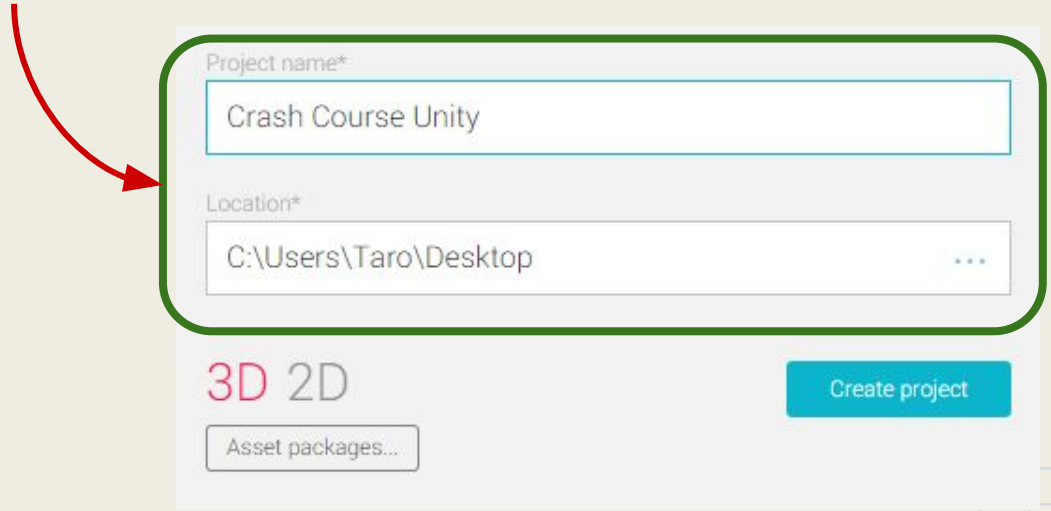
Step 1: Click “New”



Starting a new project

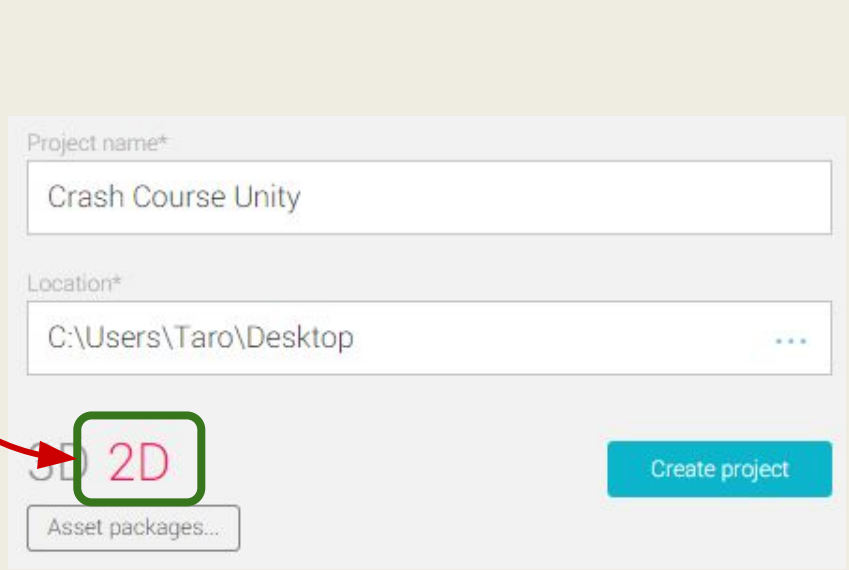
Step 2: Type in a project name and the folder it'll be created in.

Note: Unity will create a new folder with the project's name



Starting a new project

Step 3: Click on “2D” above the “Asset packages…” button

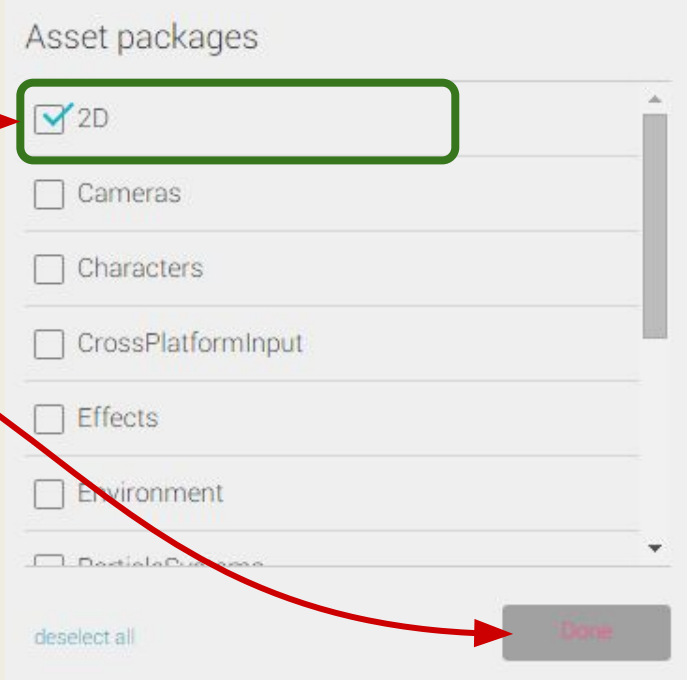


Starting a new project

Step 4: Click “Asset packages…”, and check “2D.”

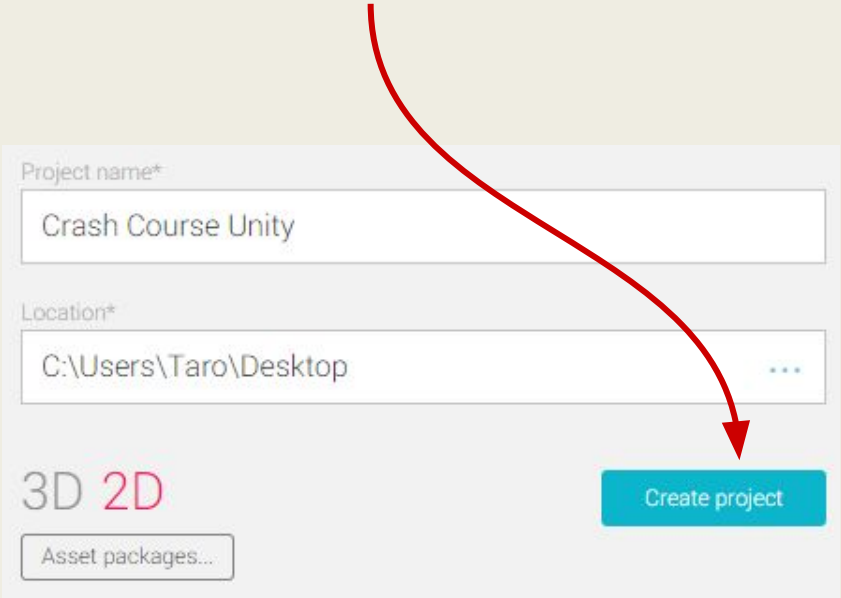
Lastly, click “Done.”

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

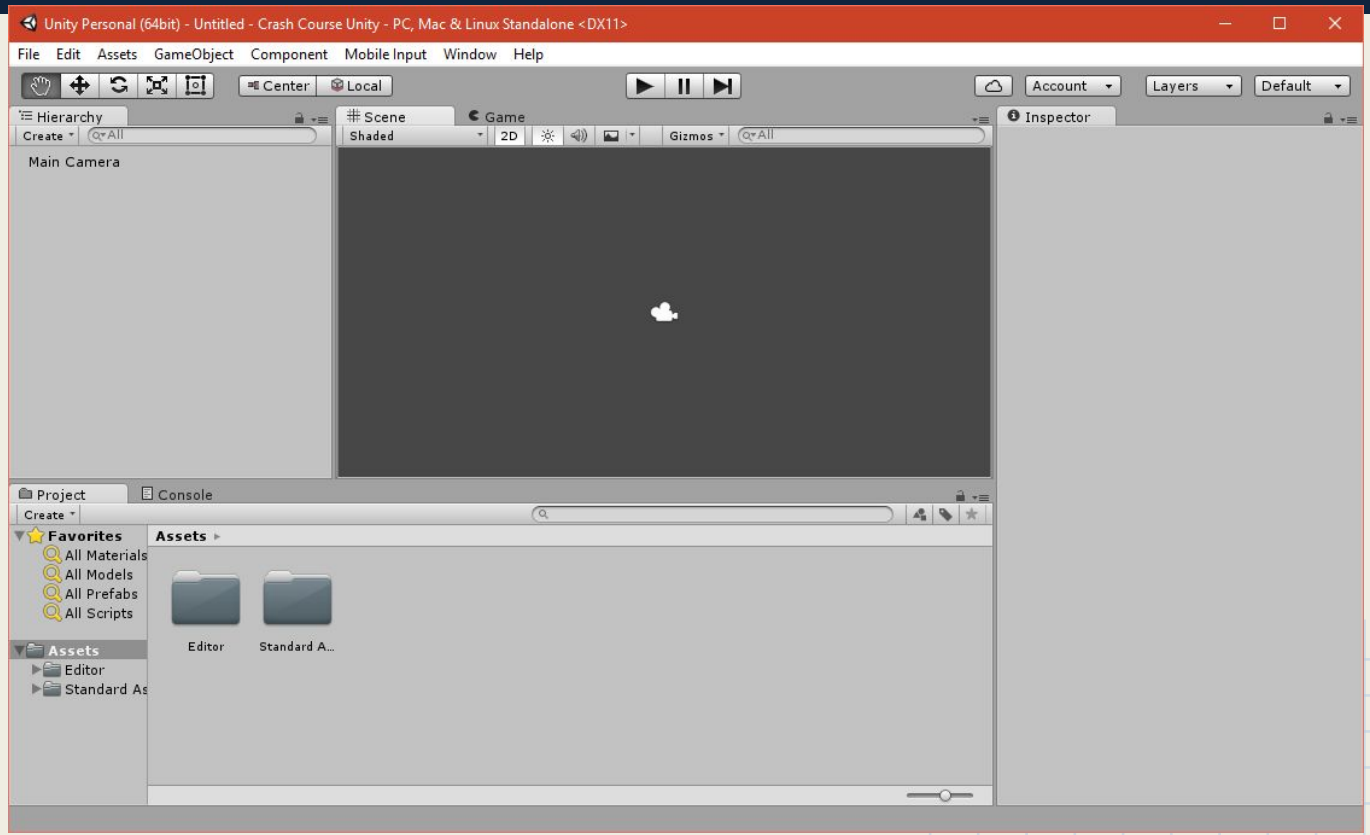


Starting a new project

Step 5: Finally, click “Create project”



About Unity


Unity Personal (64bit) - Untitled - Crash Course Unity - PC, Mac & Linux Standalone <DX11>
File Edit Assets GameObject Component Mobile Input Window Help
Center Local
Account Layers Default
Hierarchy
Create All
Main Camera
Scene Game
Shaded 2D Gizmos All
Inspector
Project Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
Editor
Standard A...
Assets
Editor
Standard As

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, Windows 10, Windows 8, Kindle Fire, PS4, PS3, PS VITA, Xbox One, Xbox 360, Wii U, New 3DS, Ouya, Samsung TV, Tizen

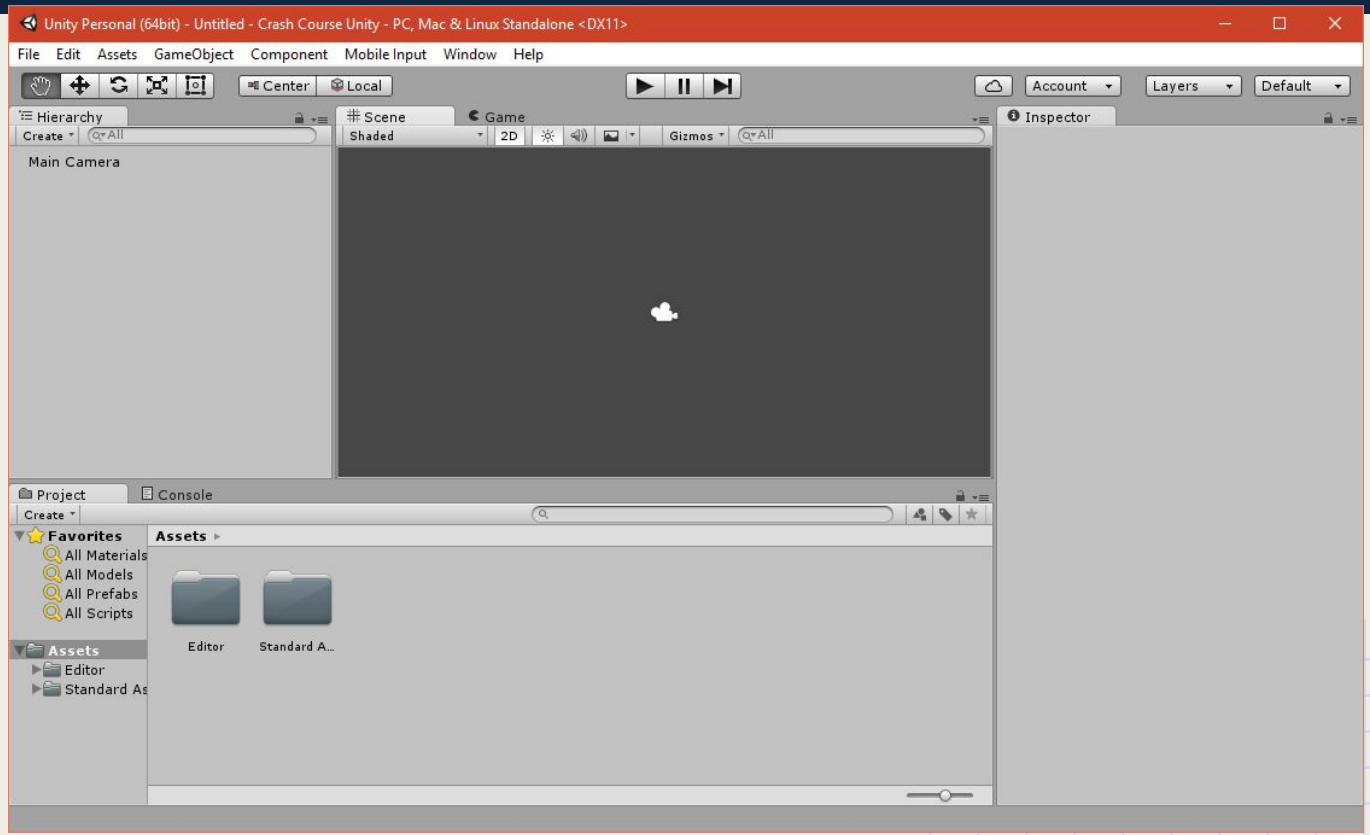
Licenses and Fees

- Personal license (Free!)
  - Build to Windows, Mac, Linux, Webplayer, HTML5 + WebGL, iOS, Android, and Windows 10
  - C# and Javascript(-ish) 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...
- 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 (PS4, Xbox One, Wii U, etc.)
  - Allow customizing splash screen

Licenses and Fees

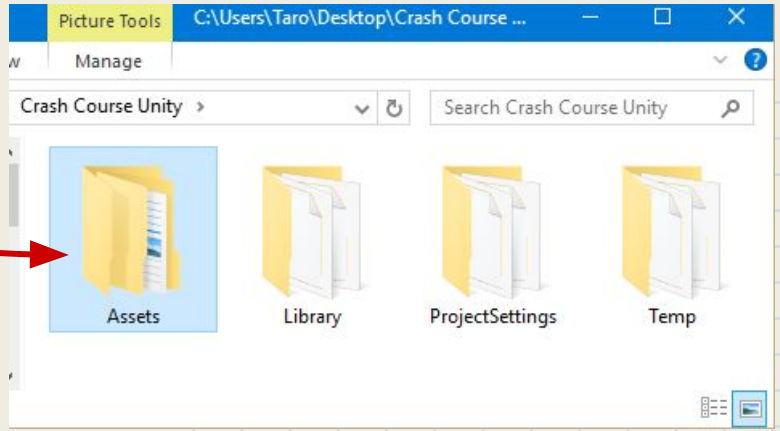
- More about the Pro license ($1,500 or $75/month)
  - Pro license is mainly for access to more online services (team-synchronization, better analytics service, better online multiplayer servers, cheaper asset store prices, etc.)
  - The $1,500 perpetual license does not cover whole-number upgrades (e.g. version 5.3.4 to 6.0.0). While these licenses can be upgraded, it will cost extra.
  - The minimum length of the $75/month subscription is 12 months. These licenses do cover all upgrades.
  - If you do get a Pro license, and plan to release on iOS and/or Android, they *each* will cost an extra $1,500 or $75/month.

Making a game



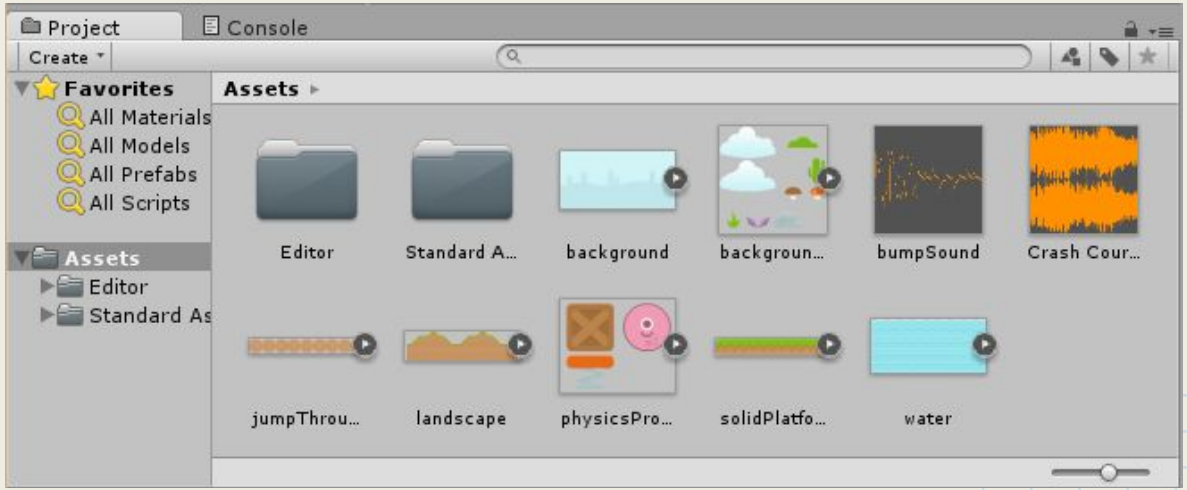
Importing stuff

1. Go to wp.me/a5G4dR-q4, and click the Crash-Course-Unity-2D-Assets link to download a ZIP file.
2. Unzip the file with your favorite file browser.
3. Move **all** of the unzipped files into the project's Assets folder.
4. Switch to Unity.



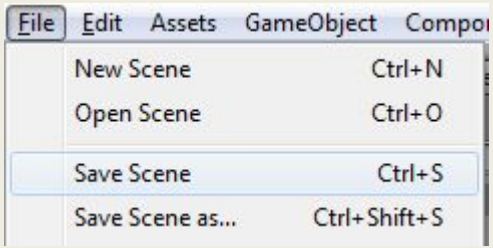
Project Pane

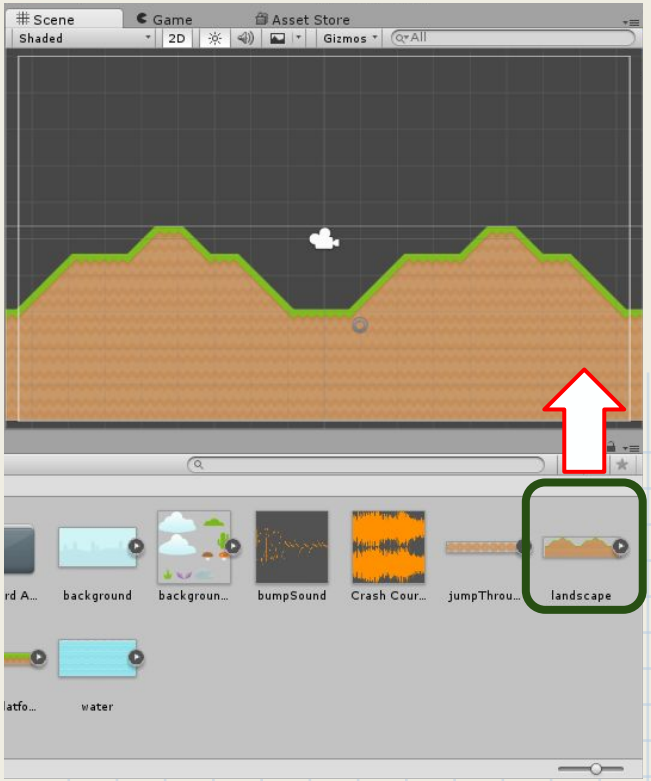
- Displays the contents of the Assets folder.
  - Automatically syncs with the folder if there's any changes
- Has a search bar to make it easier to find assets



Add a sprite to a new scene

1. Select **landscape.png** in the Project pane.
2. Drag-and-drop **landscape.png** 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
- Sprite
  - Industry-wide term for a portion of a 2D image
  - Can representative of a frame in an animation
  - Usually made in a specialized program, e.g. Photoshop, GIMP, Krita, Paint, etc.
- Scene
  - Unity's term for files storing a collection of objects
  - Holds references to assets in the Assets folder
  - Has a *.unity file extension

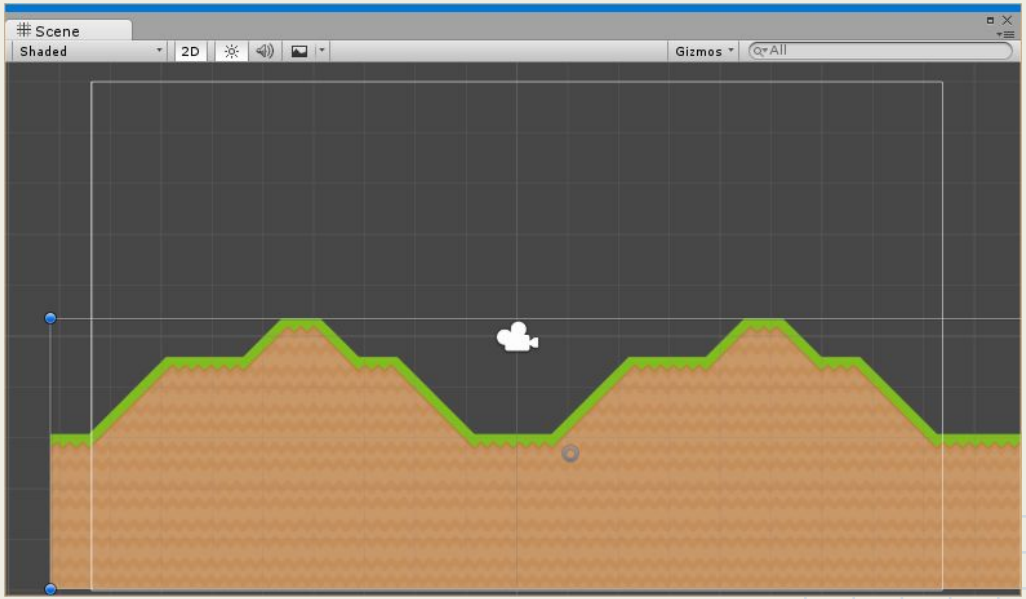
Importing Images

Unity can natively import:

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

Scene Pane

- A 2D & 3D view of a scene where objects can be positioned, rotated, and scaled.



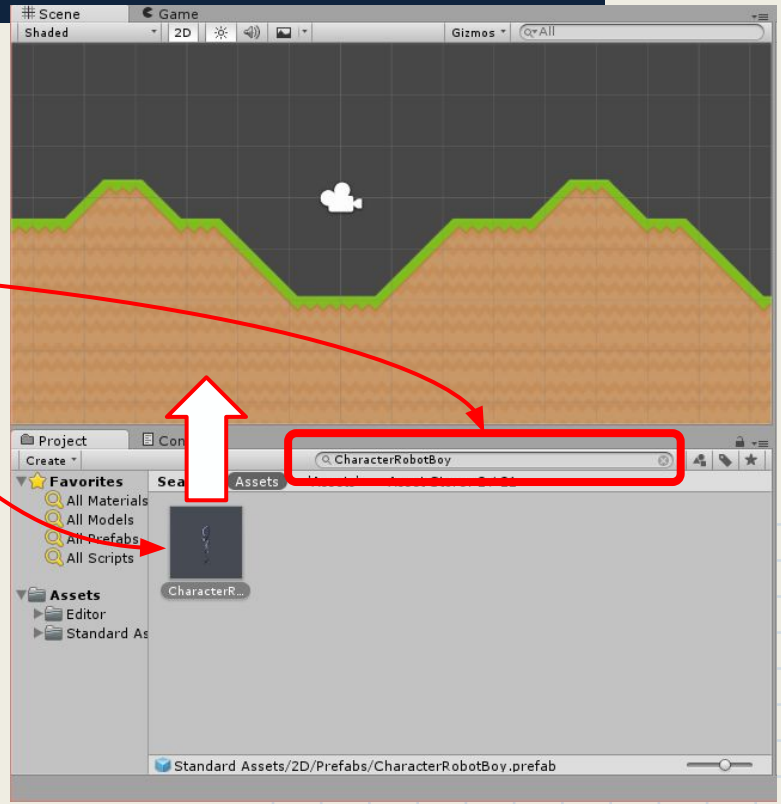
Navigating the scene in 2D

- 2-button mouse:
  - Left-click to select objects
  - Hold the right mouse button to pan
  - Hold alt and right mouse button to zoom in and out
- 3-button mouse:
  - Left-click to select objects
  - Hold the right mouse button to pan
  - Click and hold on the scroll wheel also pans
  - Scroll wheel to zoom in and out
- Hold shift to pan/rotate/zoom faster
- When the mouse is in the Scene pane, tap “F” to focus on the currently selected object

Adding a controllable robot

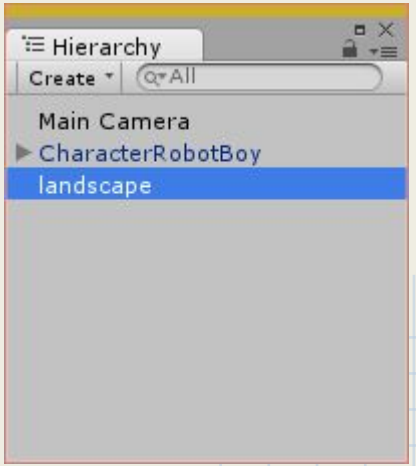
Let's add a prefab (short for prefabricated object) with platformer controls

1. In the Projects pane, search for "CharacterRobotBoy"
2. Drag-and-drop "CharacterRobotBoy.prefab" into the Scene pane.
3. Position the robot above the landscape



Hierarchy Pane

- Displays all the Game Objects (i.e. content) in the scene in a tree hierarchy.
- Game 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
- Prefab
  - Unity's term for prefabricated game objects
  - Files with *.prefab file extension
  - Allows copying a group of game objects from one scene to another
  - Also allows synchronizing data between a group of game objects, regardless of what scene it's in (more on this later)

Playing the Game

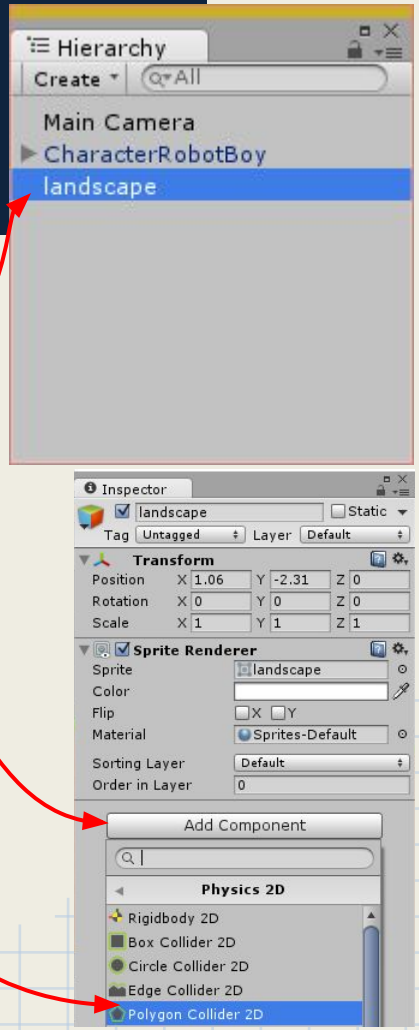
1. Press the play button. 
2. Observe your robot...fall through the floor.
3. Press the play button to stop the game.

4. What are we missing?

Adding a Collider

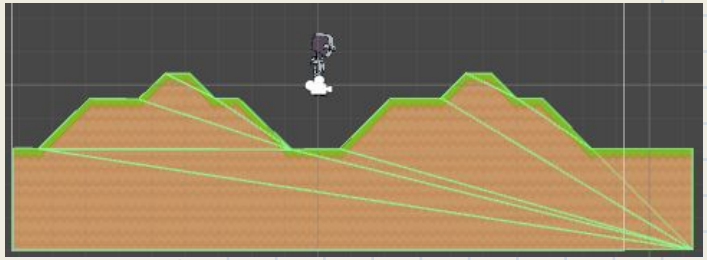
The landscape needs a collider

1. In the Hierarchy pane on the left side of the screen, select **landscape**.
2. In the Inspector pane on the right side of the screen, click "Add Component"
3. Select "Physics 2D -> Polygon Collider 2D"



Editing the Polygon Collider

1. Click the button next to “Edit Collider.”
2. Move the green nodes on the collider to the closest corner of the landscape
3. Add more corners by left-clicking in the middle of a long edge
4. Remove corners by holding ctrl (the node will turn red), then left-click a node
5. When finished, click the button next to “Edit Collider” to turn the collider editor off

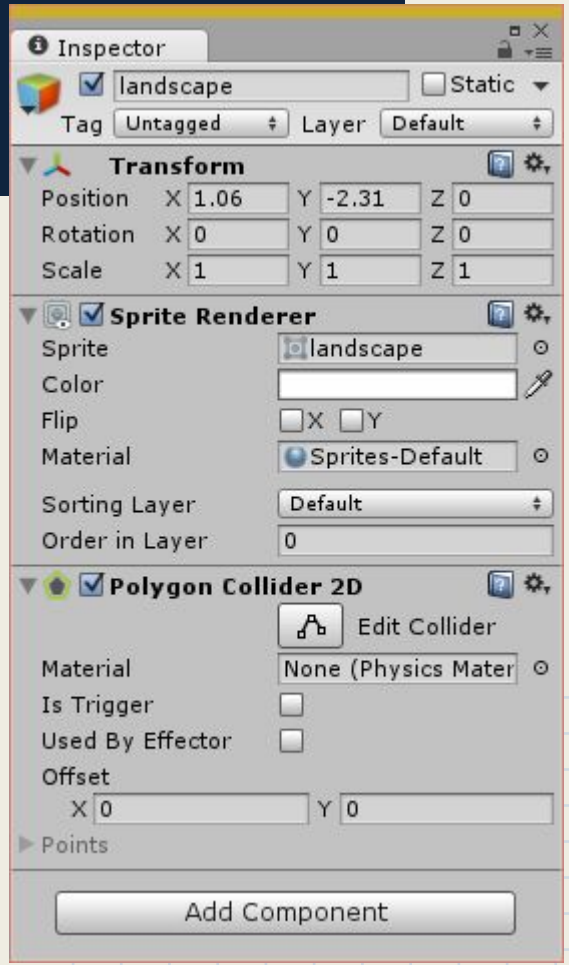


Playing the Game

1. Press the play button. 
2. Use the arrow keys (or WASD) to move, space to jump, and ctrl to crawl.
3. Marvel at your own work.

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.



Quick lexicon review

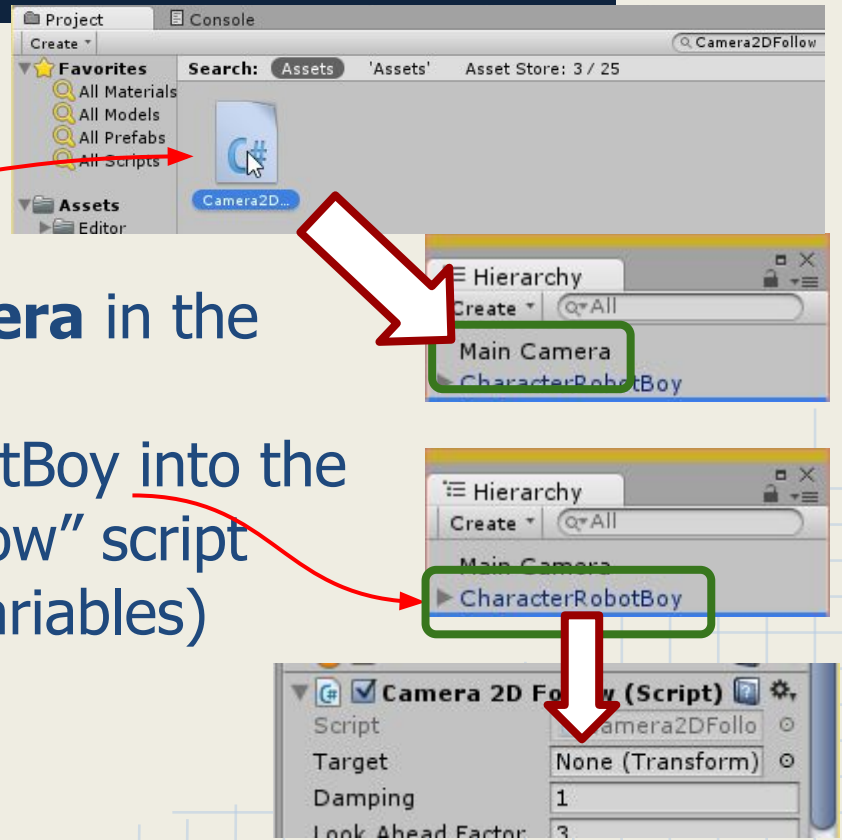
- Components
  - Unity's term for containers with specialized information
  - Game objects retain a list of components (such as Colliders)
  - Can be enabled or disabled
  - Every entry in the Inspector pane is a Component
- Colliders
  - Industry-wide term for shapes representing the boundaries of an object
  - Used by the physics engine to determine where objects collide

Types of colliders

- Polygon Colliders
  - Colliders that can be shaped to any polygon
  - Expensive and inefficient
  - Best for static, non-interactable convex shapes, like buildings
- Edge Colliders
  - Like Polygon Colliders, without any area
  - Best for static, non-interactable concave shapes, like caves
- Box Colliders
  - Rectangle-shaped colliders
  - Very efficient, great for interactable shapes
- Circle Colliders
  - Circle-shaped colliders (no oval support)

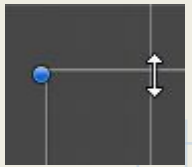
Making the camera move

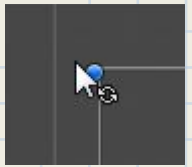
1. In the Project Pane, search for “Camera2DFollow”
2. Drag the C# script from the Project pane to the **Main Camera** in the Hierarchy pane
3. Finally, drag the CharacterRobotBoy into the Target field in “Camera 2D Follow” script (also, play around with other variables)
4. Play the game!



Making the landscape bigger

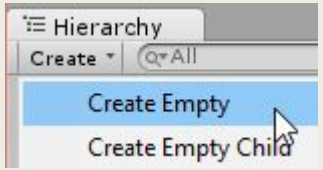
1. If not selected already, click on the right-most radio button on the upper-left hand corner
2. Click on the landscape in the Scene Pane
3. Make the landscape larger by clicking on the corner or edges of the sprite's bounding box
4. Hint: hold shift to make the corner-scaling proportional
5. Play the game!

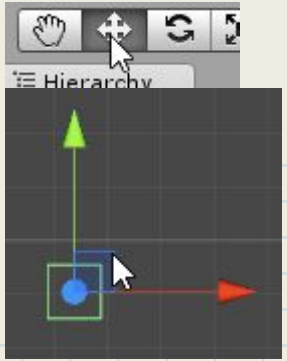




Adding invisible walls

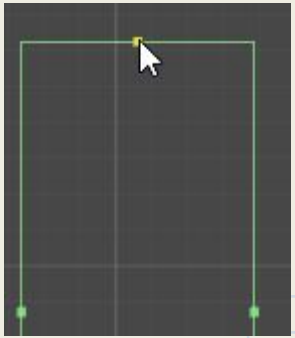
1. Click on **Create -> Create Empty** in the Hierarchy pane
2. In the new object, **GameObject**, add **Physics 2D -> Box Collider 2D** via the inspector
3. Click on the 4-way arrow control, and move the collider to the left of the landscape by clicking the red arrow, and dragging to the left





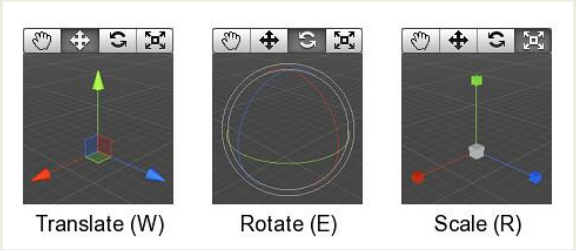
Adding invisible walls

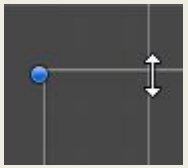
1. In the Inspector, click the **Edit Collider** button
2. Drag the 4 nodes so the collider is as tall as possible. Also make the collider wide to prevent errors.
3. Use the Ctrl+D/Cmd+D shortcut to duplicate the wall and place them to the right of the landscape



Manipulating objects

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





Move 2D Scale 2D Rotate 2D

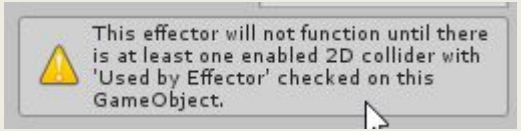
- Controls to toggle object's reference point:

- Play Game controls, from left to right:

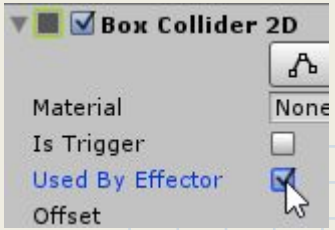
  - Play game (or if already playing, stop game)
  - Pause game (or if already paused, resume game)
  - Move forward one frame

Adding special platforms

1. Drag **jumpThroughPlatform.png** from the Project pane to the scene, and place it above the landscape
2. In the inspector, add **Physics 2D -> Box Collider 2D** & **Physics 2D -> Platform Effector 2D** to the platform
3. Observe warning:



4. As noted by the warning, check the "Used By Effector" field
5. Play the game, and jump through the platform!



About Effectors

Effectors modify the behaviour of colliders affecting Rigidbody 2D

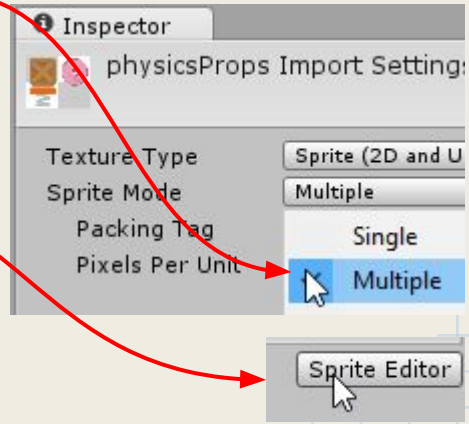
- Platformer Effector 2D
  - Simulates collision only if the Rigidbody 2D hits it at a specific angle
  - Useful for jump-through platforms
- Buoyancy Effector 2D
  - Turns a collider into a water body that lets Rigidbody 2D with low density to float, and higher density to sink slowly
- Area Effector 2D
  - Makes a collider push and/or objects to a specified direction
  - Useful for simulating wind, or creating boost pads

About Effectors

- Point Effector 2D
  - Creates an outward or inward force in a collider’s area
  - Useful for explosions and magnetism
- Surface Effector 2D
  - Makes the surface of a collider push objects in a specified direction
  - Useful for conveyer belts

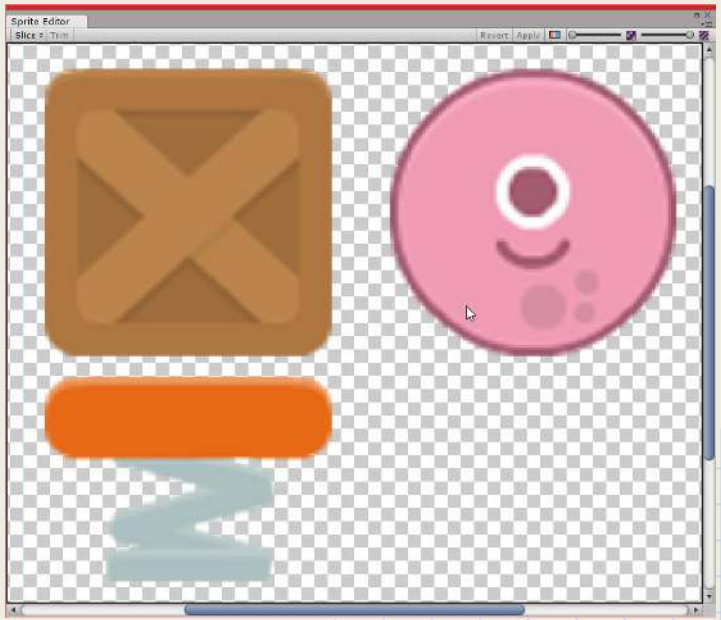
Using Sprite Editor

1. Click on **physicsProps.png** in the Project pane
2. In the Inspector pane, change the "Sprite Mode" to "Multiple"
3. Finally, click the "Sprite Editor" button at the bottom of the inspector
4. If a warning pops-up, click "Apply"



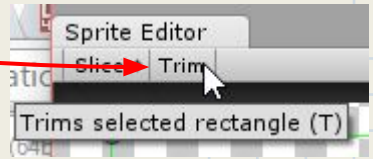
Using Sprite Editor

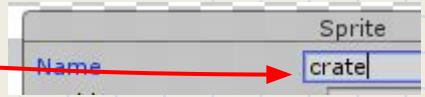
With the Sprite Editor, you can cut-out multiple sprites from a single image file (called sprite sheet). This makes the game more efficient (it loads less files).



Using Sprite Editor

1. Click and drag at the sprite sheet's upper-left hand corner, then down the the lower-right corner of the crate, creating a green square cut-out boundary
2. In the upper-left hand corner of the dialog, click on **Trim** to make the cut-out tighter
3. On the lower-right hand pop-up, change the name to "crate"
4. Repeat these steps for the other elements, with names like "ball" and "spring"





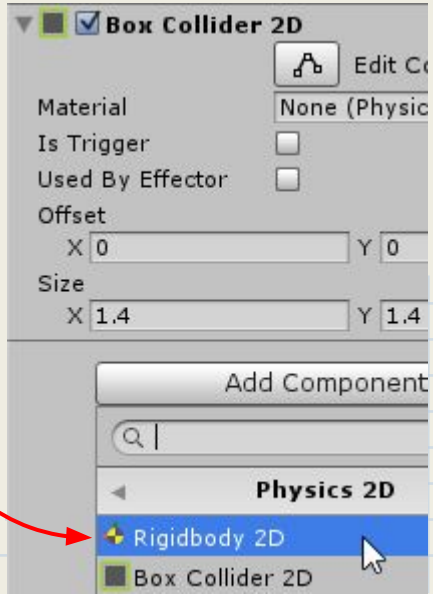
Using Sprite Editor

1. Click the **Apply** button on the upper-right hand corner of the dialog

2. Close the Sprite Editor
3. On the Project Pane, click on the arrow on the **physicsProps.png** to expand all of its sprites

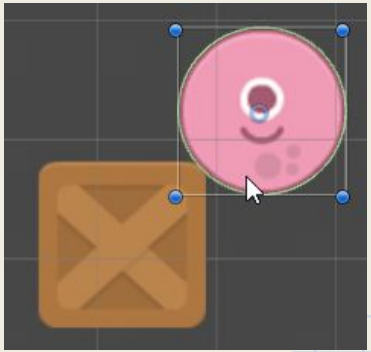
Adding Interactable objects

1. On the Project Pane, drag-and-drop the crate from **physicsProps.png** to the scene
2. As usual, add a **Box Collider 2D** component
3. Using the same method, add "Physics 2D -> Rigidbody 2D"
4. Play the game!



Making weird shapes

1. Drag the ball sprite into the scene
2. In the Scene pane, position the new ball on a corner of the crate
3. Add a **Circle Collider 2D** on the ball
4. In the Hierarchy pane, drag & drop the ball into the crate. The ball should now be a child of crate.
5. Play the game, and push the crate around!

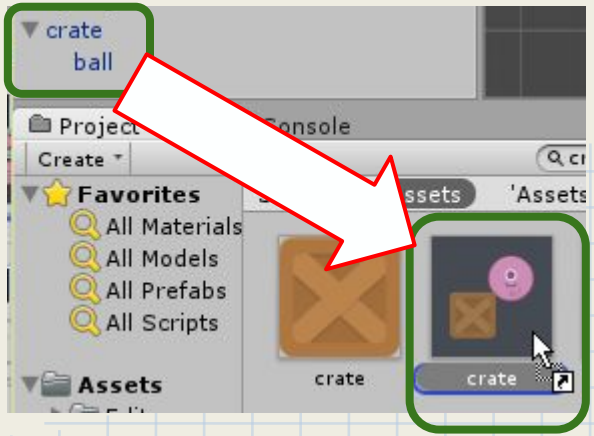


Quick lexicon review

- 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

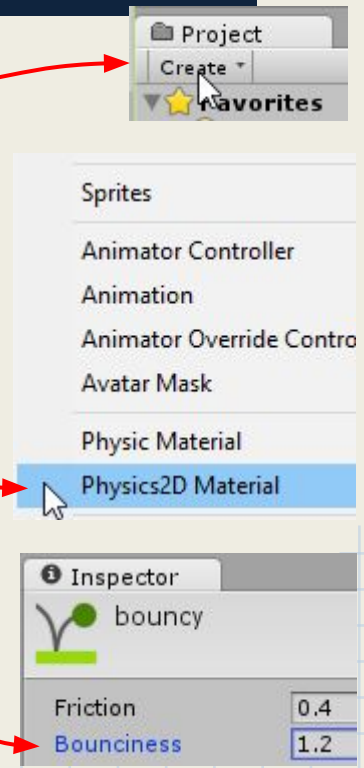
Duplicating the crate

1. Drag & Drop the **crate** from the Hierarchy pane to the Project pane. This creates a new Prefab.
2. Drag & Drop the crate prefab from the Project pane to the Scene pane as many times as you like. This will create many copies of crate.



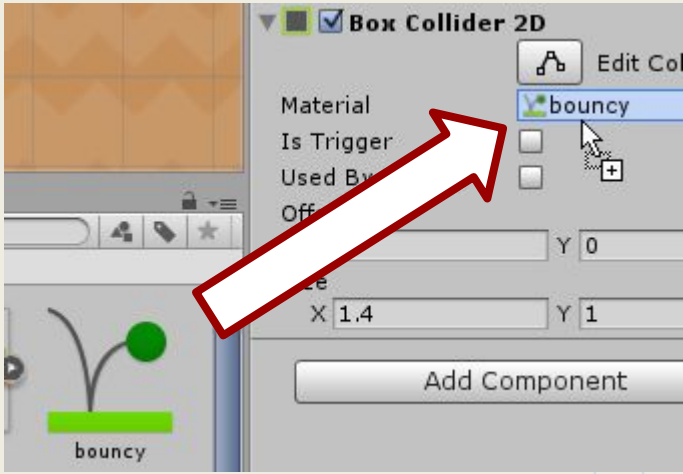
Changing physics

1. Click the "Create" button on the upper-left hand corner of the Project Pane
2. Select "Physics2D Material"
3. Name the new file, "bouncy"
4. In the Inspector pane, change the bounciness to 1.2 (as in, 120%)



Changing physics

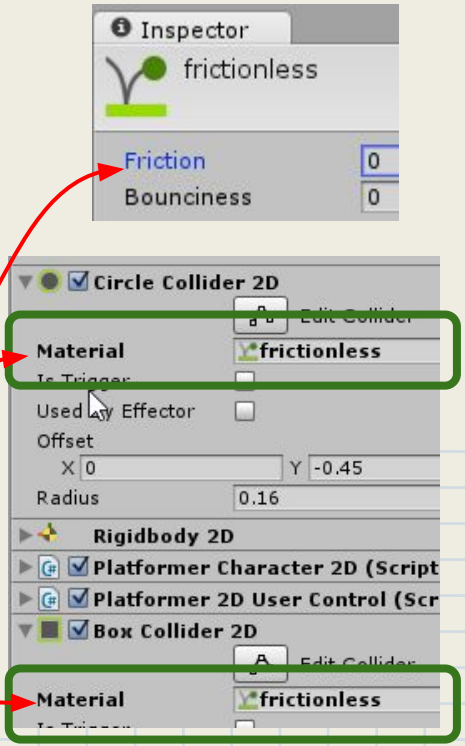
1. Drag the spring sprite to the scene, and place them on top of a flat surface
2. Click on the spring in the scene
3. Add a **Box Collider 2D** via the Inspector
4. Drag-and-drop the **bouncy** physics material
5. Play the game, and jump on top of the spring!



Fixing rough edges

The robot clinging on any vertical edge can be fixed by making the robot frictionless

1. Create a Physics2D Material in the project pane again
2. Name the new file, “frictionless”
3. In the Inspector pane, change the friction to 0 (as in, 0%)
4. Select **CharacterRobotBoy** in the scene
5. Change the Material field in both **Circle Collider 2D** and **Box Collider 2D** to “frictionless”

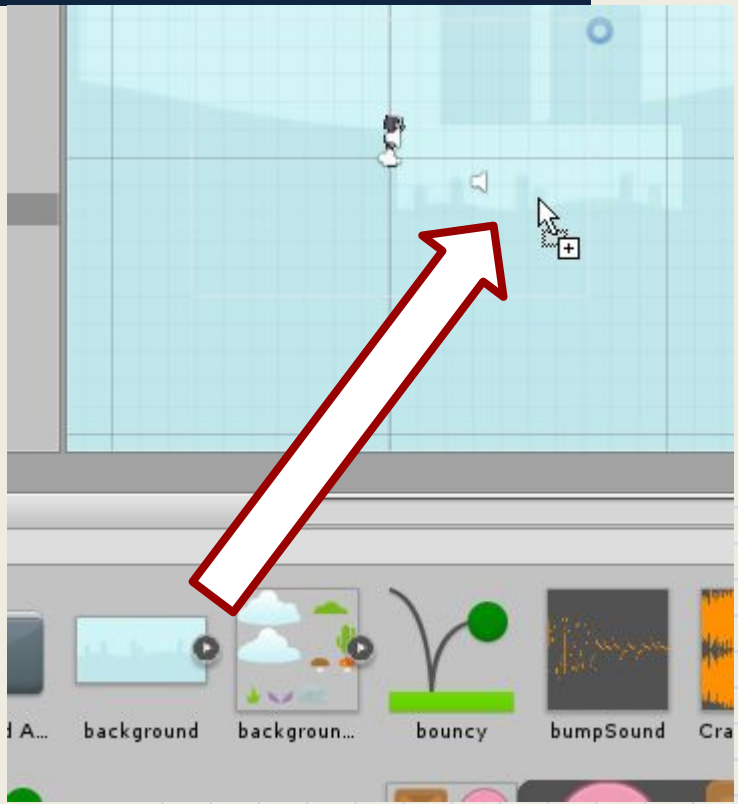


Quick lexicon review

- 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 (*.physicMaterial2D) shared between game objects
  - Changing a physics material’s properties will update all game objects with the same physics material

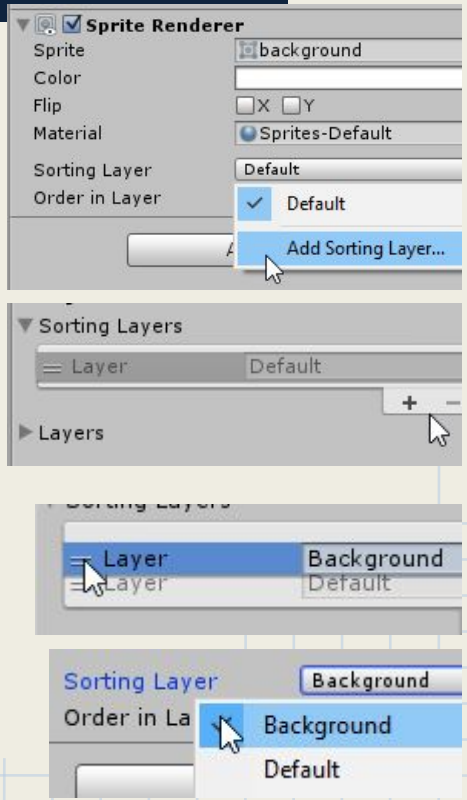
Adding background

1. Drag the background.png from Project pane into the scene
2. Resize the background to take up the entire landscape
3. ...Except it also covers the landscape



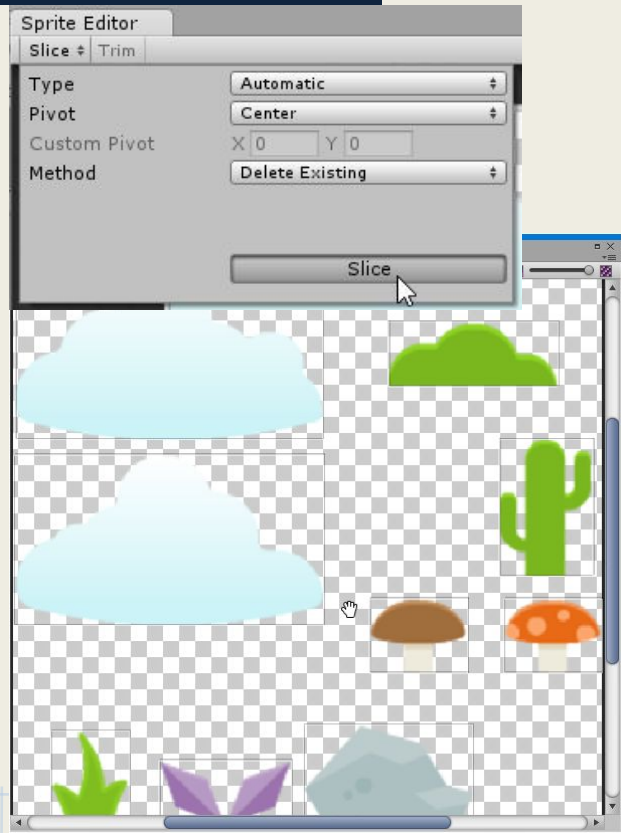
Adding background

1. Click on the “Sorting Layer” property in the **Sprite Renderer** component in the Inspector, and click “Add Sorting Layer…”
2. Click on the “+” sign below Sorting Layer. Name the new layer as “Background”
3. Use the double-lines on the left entry to drag Background above Default. Layers lower on the list are drawn on-top.
4. Click on the background again, and change it’s “Sorting Layer” property to “Background.”



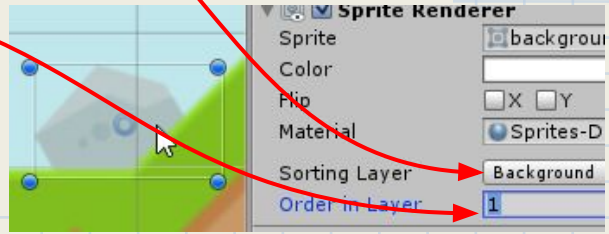
Adding more decorations

1. In the Project pane, click on **backgroundProps.png**
2. In the Inspector pane, change the "Sprite Mode" to "Multiple," and click the "Sprite Editor" button
3. If a warning pops-up, click "Apply"
4. On the Sprite Editor dialog, expand "Slice" at the upper-left hand corner, and click the "Slice" button



Adding more decorations

1. Click “Apply” on the upper-right hand corner of the Sprite Editor dialog, then close the dialog
2. Expand **backgroundProps.png** in the Project Pane, and drag any one of the sprites into the scene
3. Change the sprite’s “Sorting Layer” property to “Background”
4. Change the “Order in Layer” property to 1



About drawing order

- If 2 sprites have the same Sorting Layer and Order in Layer, and are placed at the same location, they will appear to flash.
  - Essentially, both sprites are fighting to be drawn on-top of each other
  - Commonly known as “clipping”
- Sorting Layer broadly groups sprites to be drawn in a specific order (lower in the layer list = drawn on top)
- Order in Layer adds finer controls to the draw order of individual sprites in a layer (larger number = drawn on top)

Adding music

1. Select **CharacterRobotBoy** in scene
2. In the Inspector pane, click "Add Component"
3. Select "Audio -> Audio Source"
4. In the Project pane, drag-and-drop **Crash Course Unity 2D Theme.ogg** into the Audio Source's "Audio Clip" property
5. Check the “Loop” field
6. Play the game!

Importing audio

Unity can natively import:

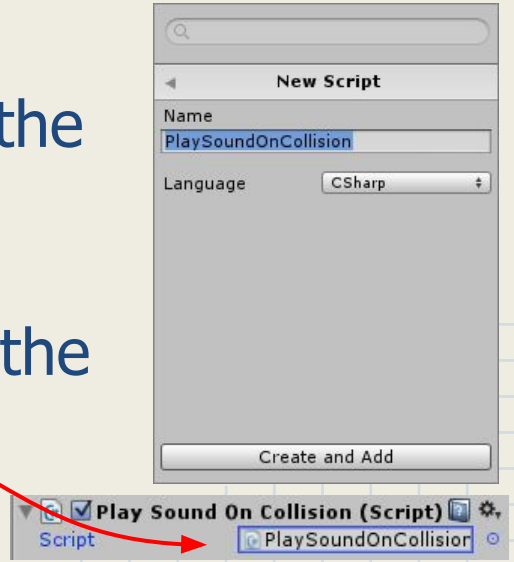
- MP3 (*.mp3)
  - Lossy, compressed. Best for music, especially in mobile devices
- OGG (*.ogg)
  - Lossy, compressed. Best for music, especially PC and consoles
- WAV (*.wav)
  - Lossless, uncompressed. Best for short sound effects
- AIFF (*.aif, *.aiff)
  - Lossless, uncompressed. Best for short sound effects

Adding a sound

1. Select a crate in the Scene pane
2. In the Inspector pane, add an **Audio Source** on the crate
3. In the Project pane, drag-&-drop **bumpSound.wav** into the "Audio Clip" property
4. Play the game!

Adding a Script

1. Select the crate 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

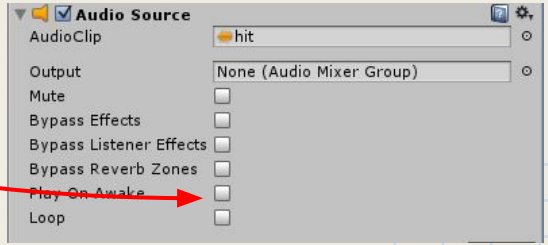


Copy the Following:

```
using UnityEngine;
public class PlaySoundOnCollision : MonoBehaviour {
    AudioSource audioCache;
    void Start () {
        audioCache = GetComponent<AudioSource>();
    }
    void OnCollisionEnter2D(Collision2D info) {
        audioCache.Stop();
        audioCache.Play();
    }
}
```

Finishing the Sound Effect

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



Script Summary

```
AudioSource audioCache;
void Start () {
    audioCache = 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. `audioCache = GetComponent<AudioSource>()` stores the Audio Source Component in a variable, `audioCache`

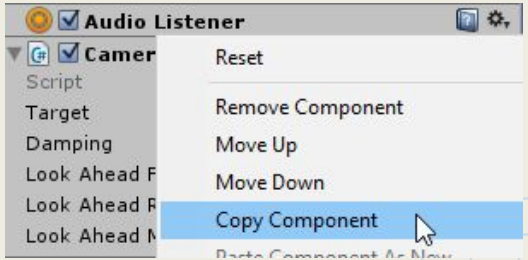
Script Summary

```
void OnCollisionEnter2D(Collision2D info) {
    audioCache.Stop();
    audioCache.Play();
}
```

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

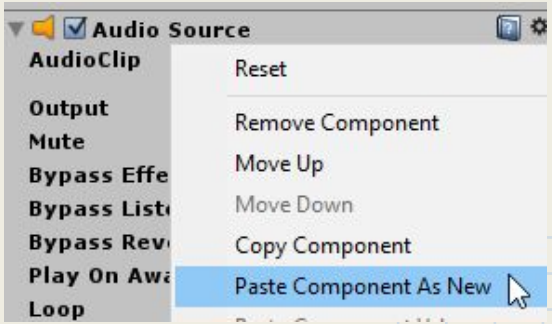
Sound Effect polish

1. Change the Audio Source's Spatial Blend to 1 (full 3D)
2. Play the game. The sound will be quiet due to the "ear" placed on the camera.
3. Click on the **Main Camera** in the Hierarchy
4. In the Inspector, click on the gear at the right of the **Audio Listener**, and select "Copy Component"
5. Click the gear again, and select "Remove Component"



Sound Effect polish

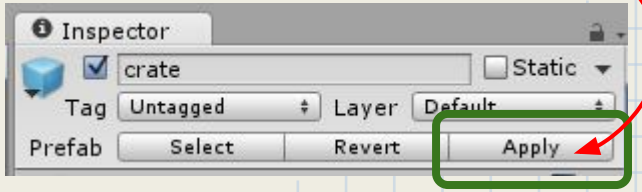
1. Click on the **CharacterRobotBoy** in the Hierarchy
2. In the Inspector, click on the gear on any component, and select “Paste Component As New”
3. Play the game. Notice crates play sounds louder if you’re closer.



Synchronizing Prefabs

You might notice only one crate plays a sound effect. Since it's a prefab, though, we can synchronize its state with the other crates.

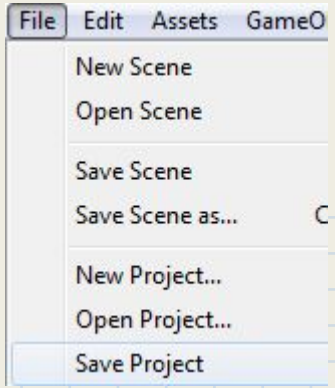
1. Click on the crate with the sound effect
2. At the top of the inspector, click "Apply"
3. Note the bolded fields in the Audio Source turned to normal font style
4. Play the game!



Save project

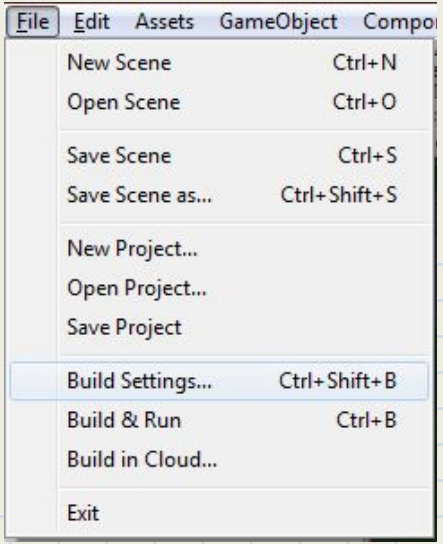
Select “File -> Save Project”

- 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.



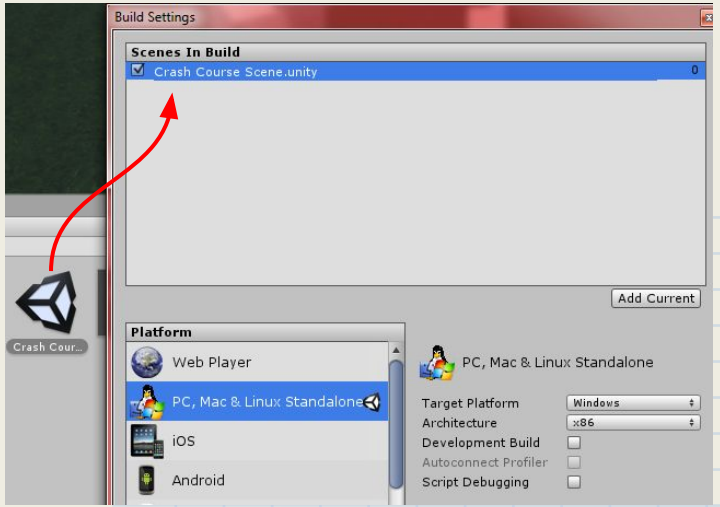
Building an Executable

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



Congratulations!

Any questions?

Supplementary materials

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

Supplementary materials

- docs.unity3d.com/Manual/index.html
  a. Manual for Unity, including scripting documentation
  b. Alternatively, click the “help” icon in the Inspector pane to bring up documentation

Using Version Control?

- Select “Edit -> Project Settings -> Editor”
- In the Inspector, change “Version Control Mode” to “Visible Meta Files”
- Change “Asset Serialization Mode” to “Force Text”
- Select “File -> Save Project”
- Version the project’s “Assets” and “ProjectSettings” folders (the rest can be ignored)