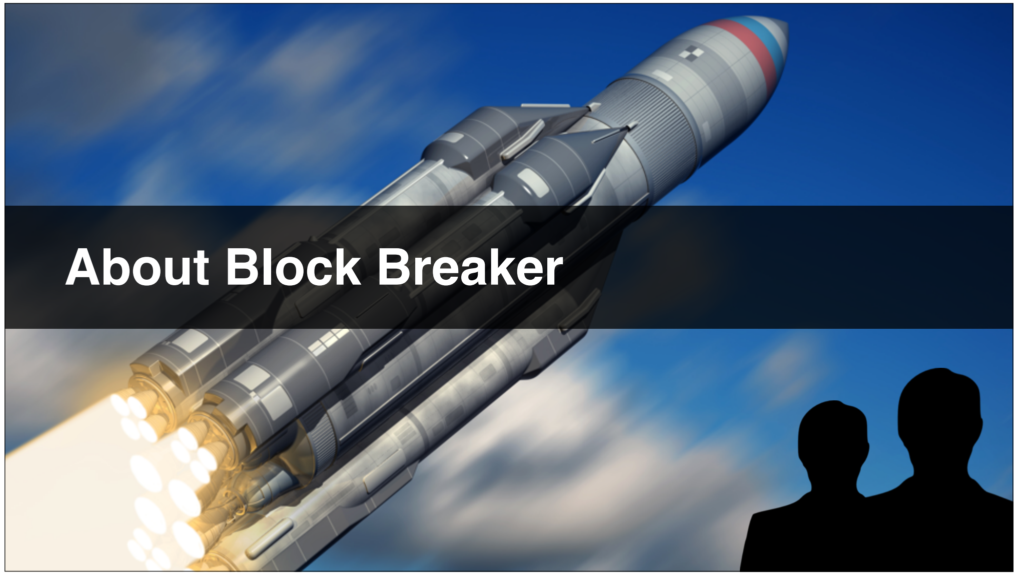
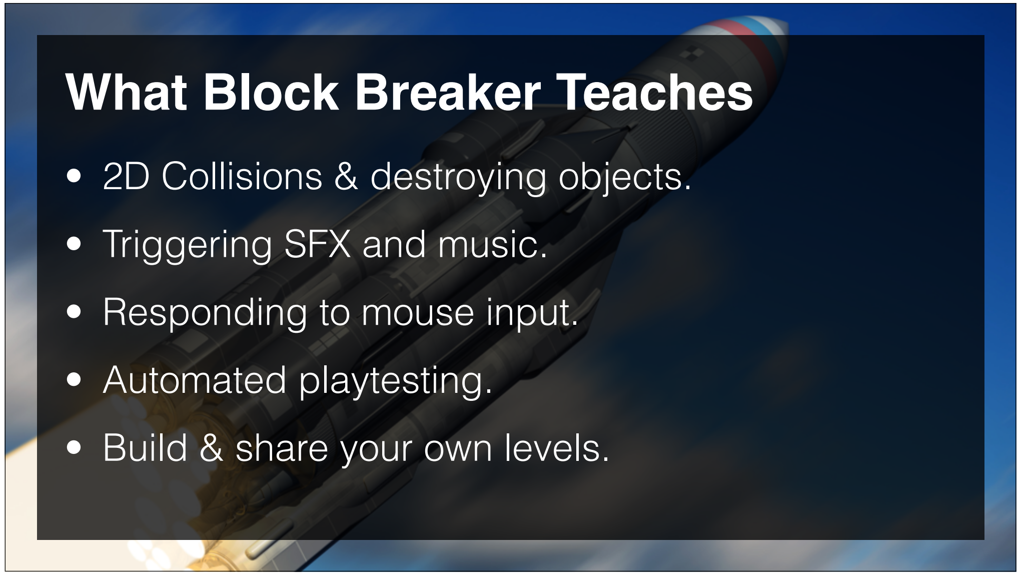
BLOCK BREAKER SLIDE DECK



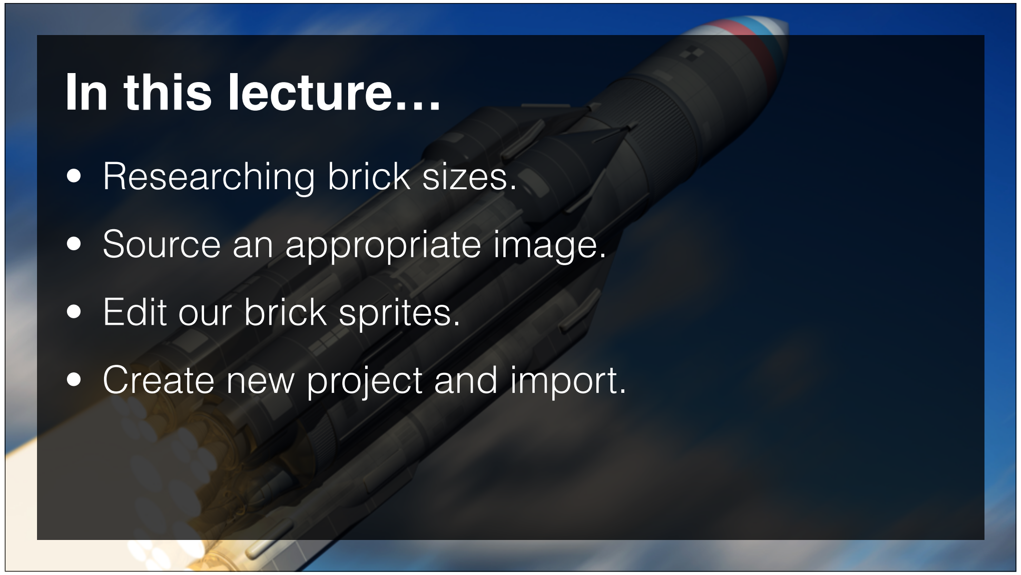
About Block Breaker



What Block Breaker Teaches

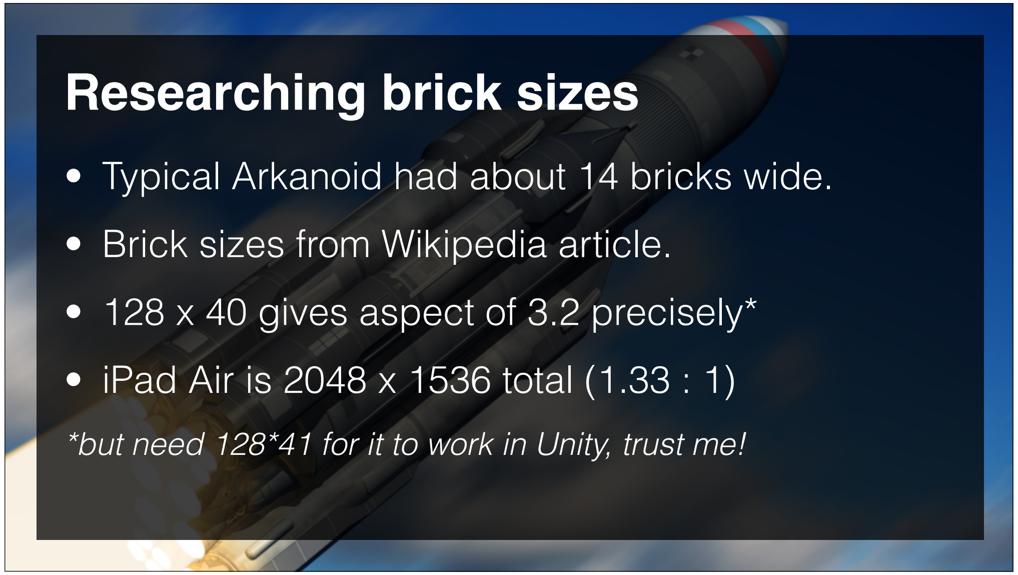
- 2D Collisions & destroying objects.
- Triggering SFX and music.
- Responding to mouse input.
- Automated playtesting.
- Build & share your own levels.

Creating A 2D Paddle Sprite



In this lecture…

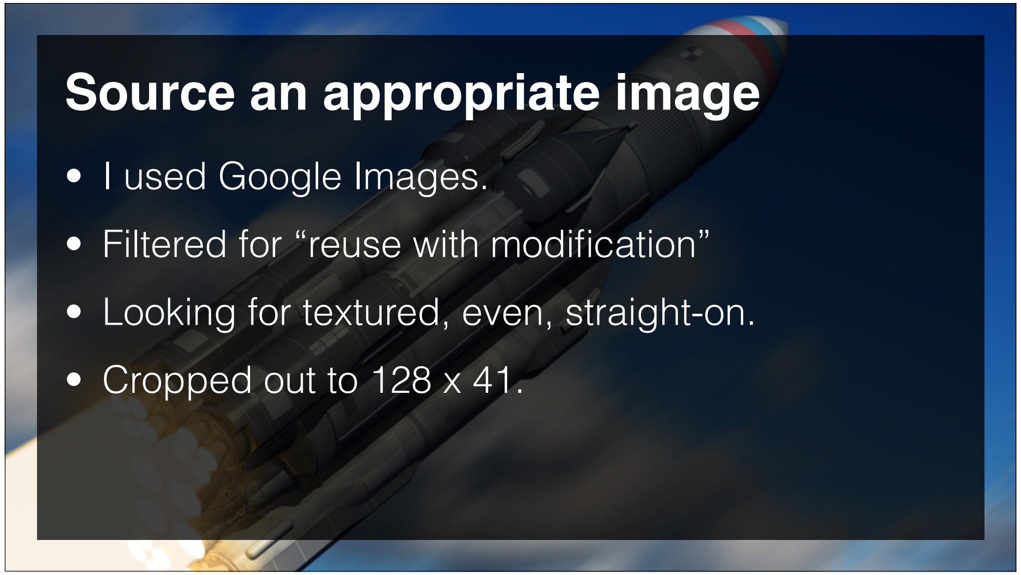
- Researching brick sizes.
- Source an appropriate image.
- Edit our brick sprites.
- Create new project and import.



Researching brick sizes

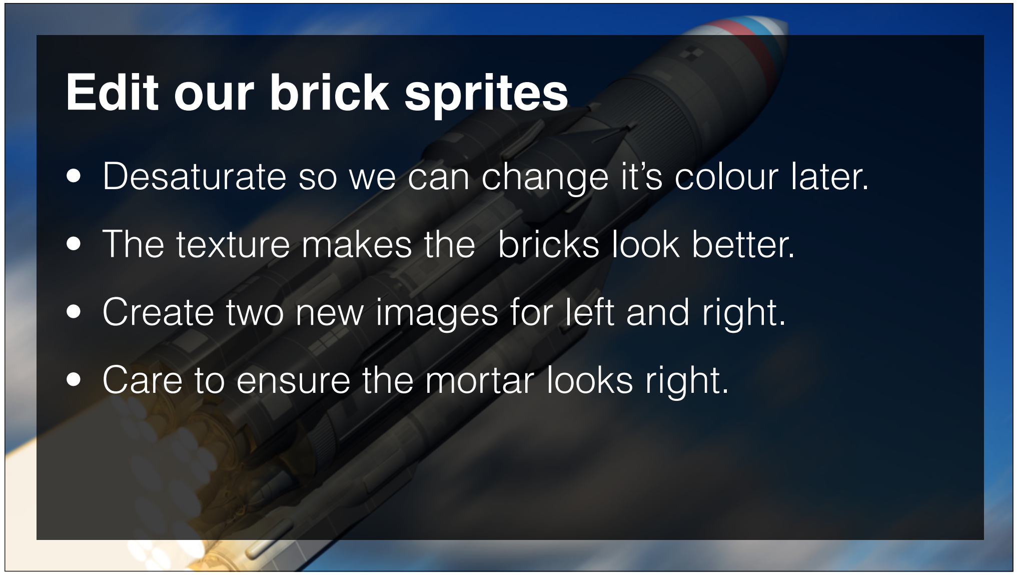
- Typical Arkanoid had about 14 bricks wide.
- Brick sizes from Wikipedia article.
- 128 x 40 gives aspect of 3.2 precisely*
- iPad Air is 2048 x 1536 total (1.33 : 1)

**but need 128*41 for it to work in Unity, trust me!*



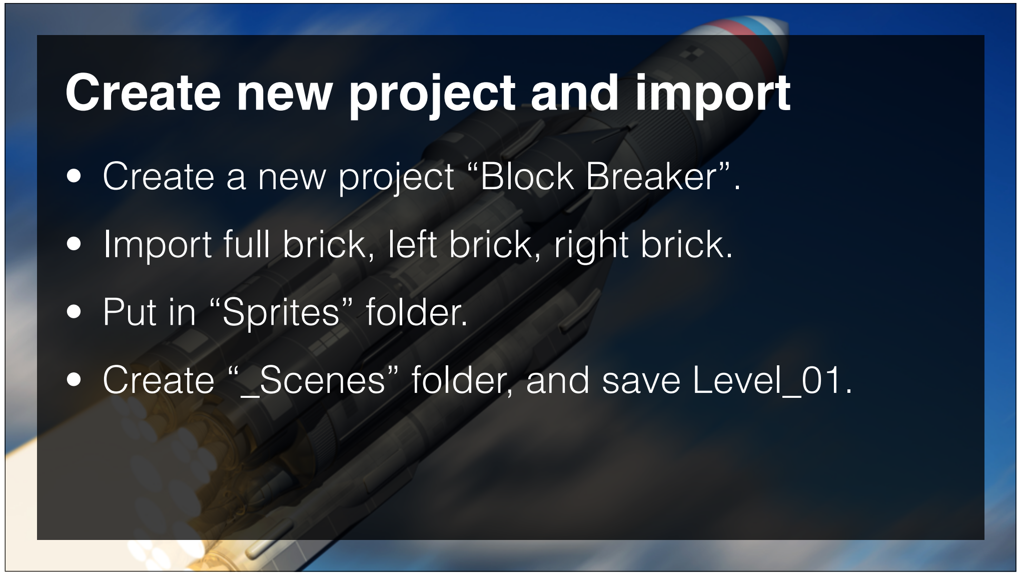
Source an appropriate image

- I used Google Images.
- Filtered for “reuse with modification”
- Looking for textured, even, straight-on.
- Cropped out to 128 x 41.



Edit our brick sprites

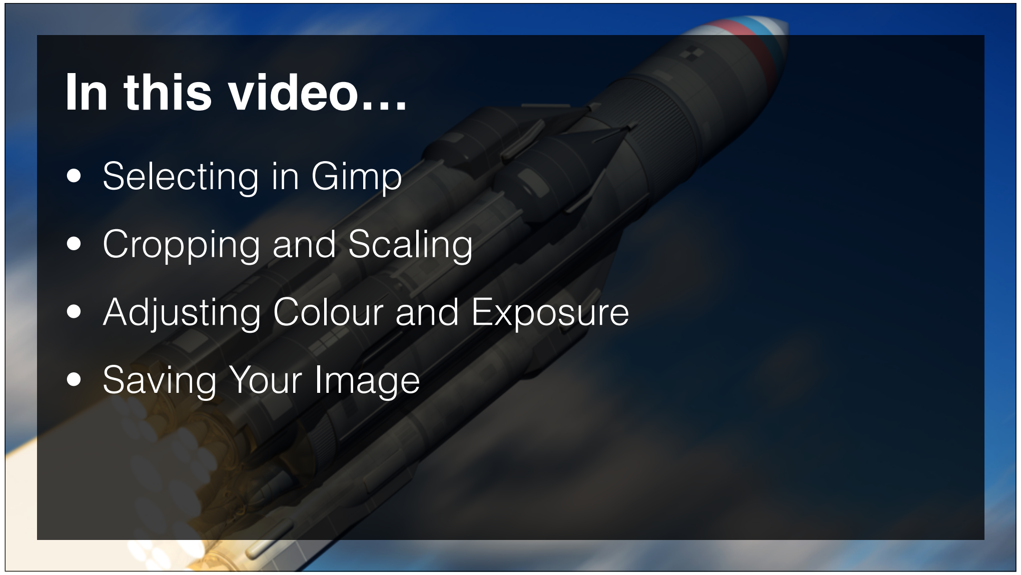
- Desaturate so we can change it’s colour later.
- The texture makes the bricks look better.
- Create two new images for left and right.
- Care to ensure the mortar looks right.



Create new project and import

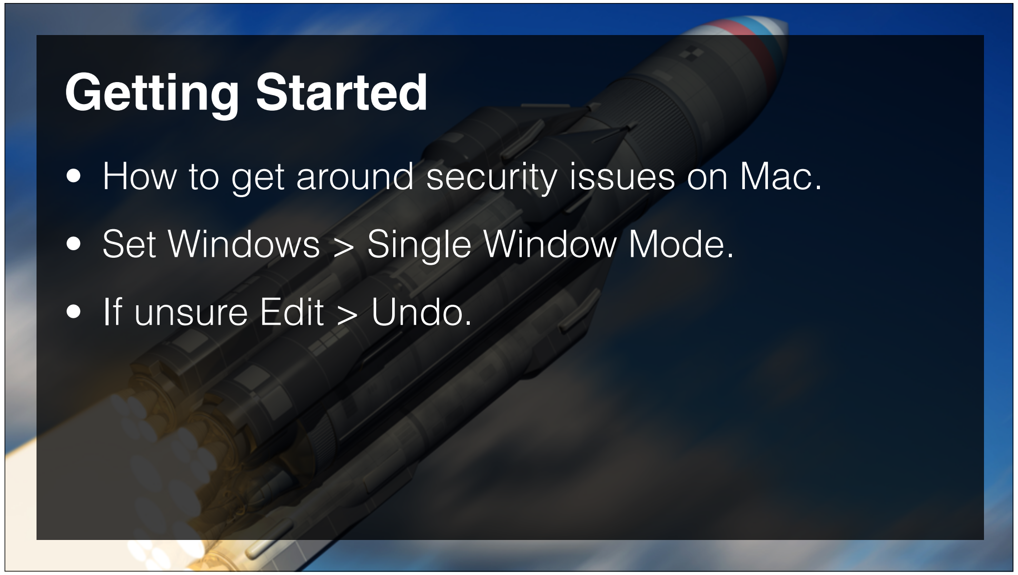
- Create a new project “Block Breaker”.
- Import full brick, left brick, right brick.
- Put in “Sprites” folder.
- Create “_Scenes” folder, and save Level_01.

Gimp Image Editing 101



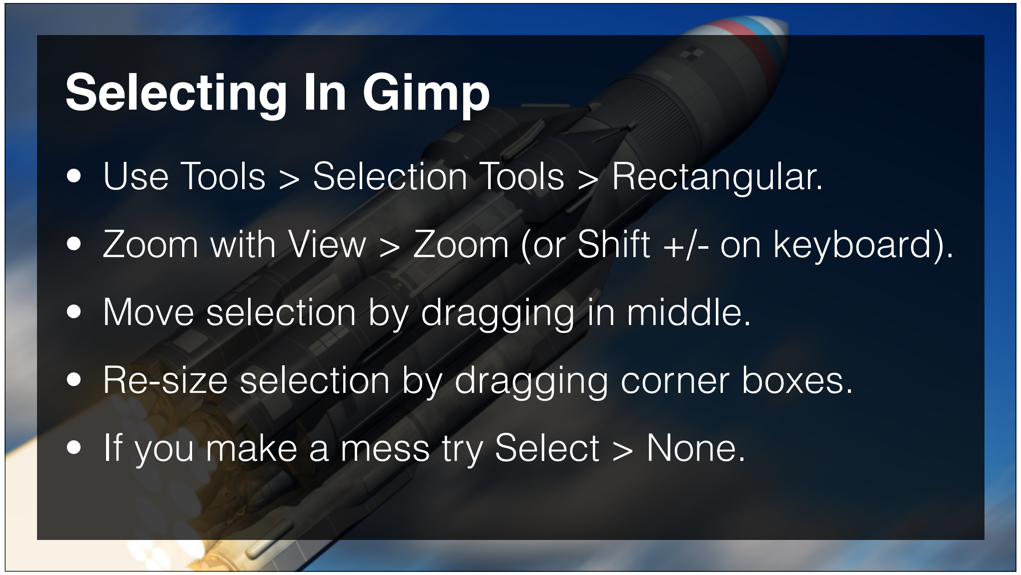
In this video…

- Selecting in Gimp
- Cropping and Scaling
- Adjusting Colour and Exposure
- Saving Your Image



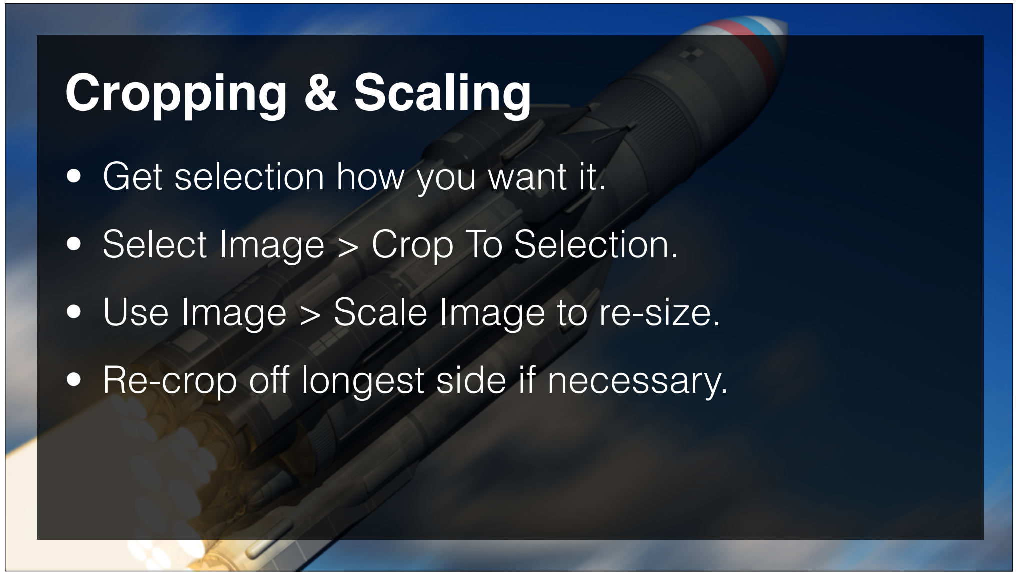
Getting Started

- How to get around security issues on Mac.
- Set Windows > Single Window Mode.
- If unsure Edit > Undo.



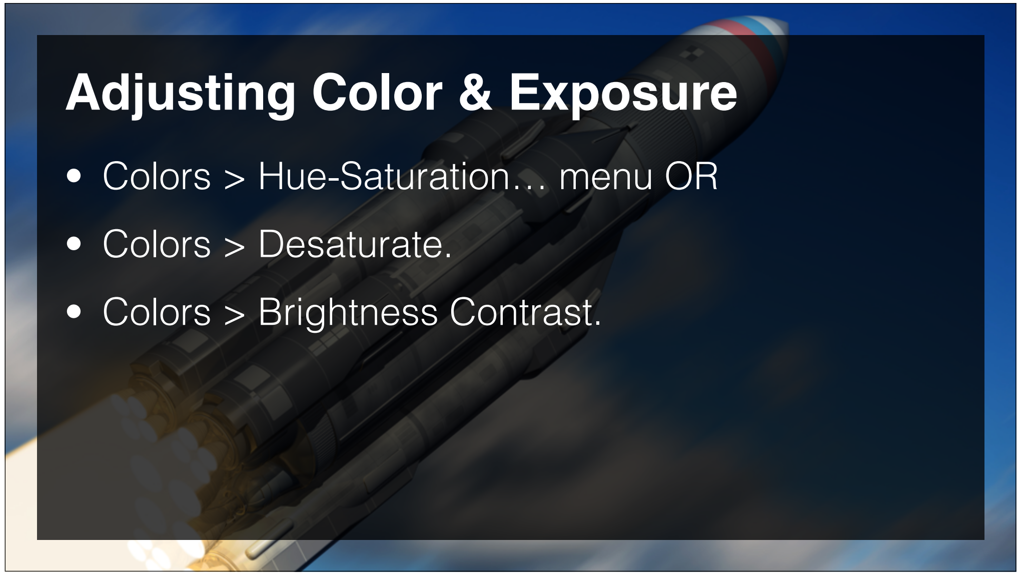
Selecting In Gimp

- Use Tools > Selection Tools > Rectangular.
- Zoom with View > Zoom (or Shift +/- on keyboard).
- Move selection by dragging in middle.
- Re-size selection by dragging corner boxes.
- If you make a mess try Select > None.



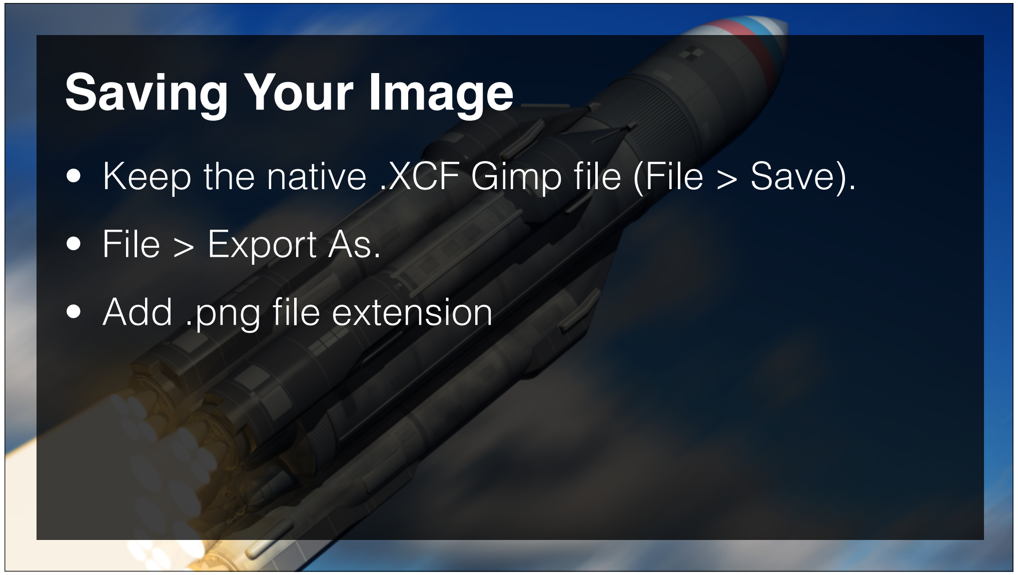
Cropping & Scaling

- Get selection how you want it.
- Select Image > Crop To Selection.
- Use Image > Scale Image to re-size.
- Re-crop off longest side if necessary.



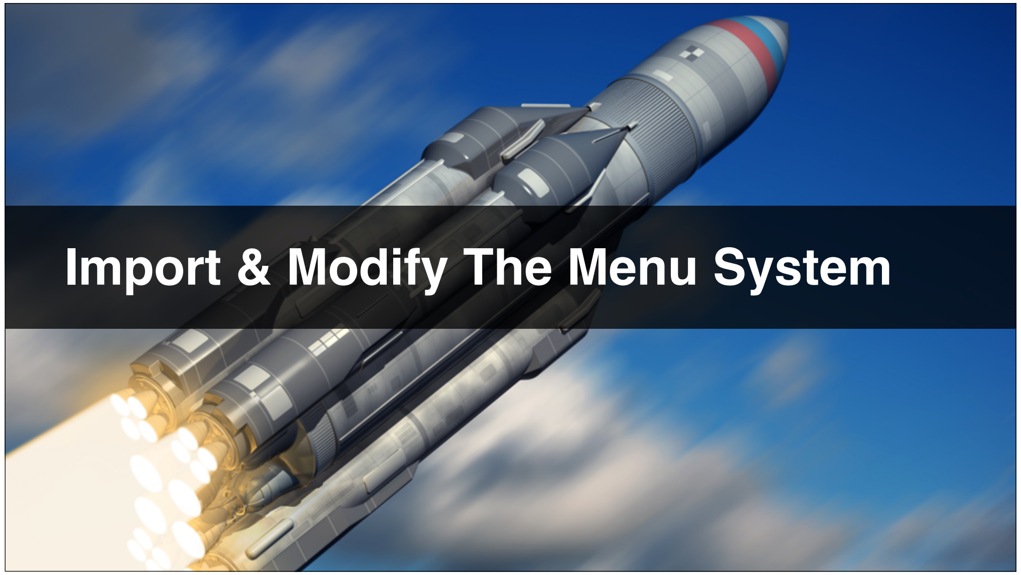
Adjusting Color & Exposure

- Colors > Hue-Saturation… menu OR
- Colors > Desaturate.
- Colors > Brightness Contrast.

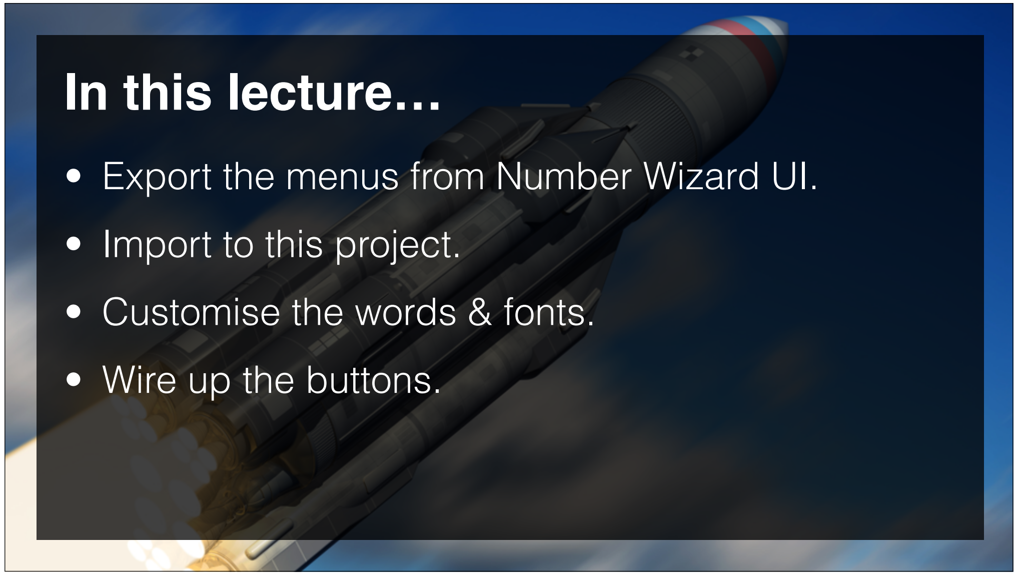


Saving Your Image

- Keep the native .XCF Gimp file (File > Save).
- File > Export As.
- Add .png file extension

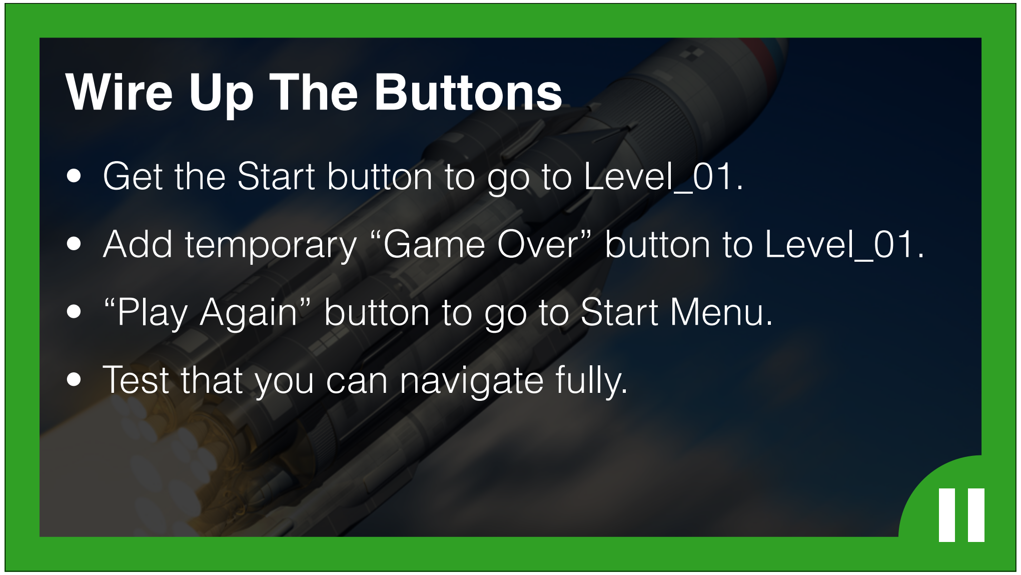


Import & Modify The Menu System



In this lecture…

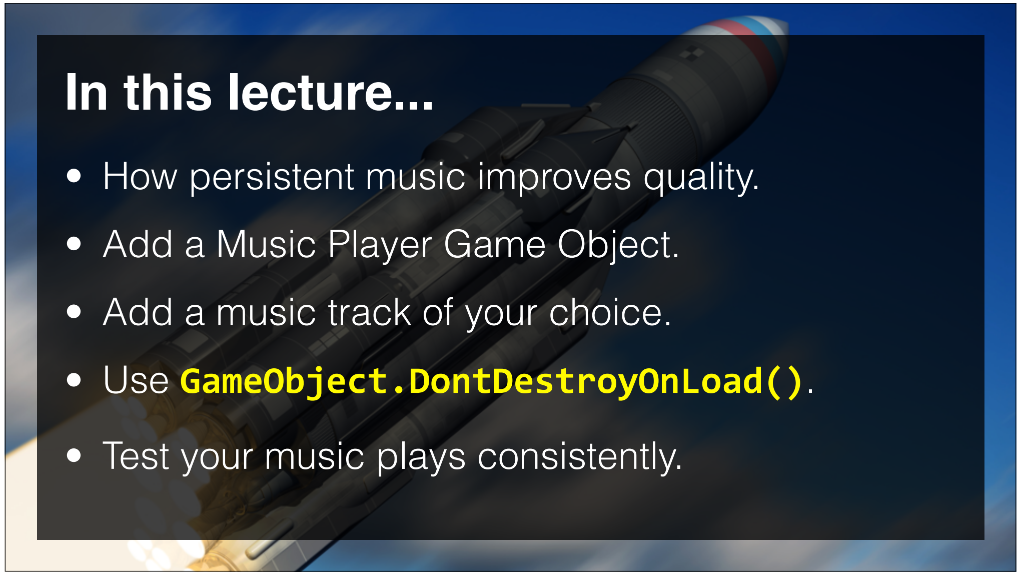
- Export the menus from Number Wizard UI.
- Import to this project.
- Customise the words & fonts.
- Wire up the buttons.



Wire Up The Buttons

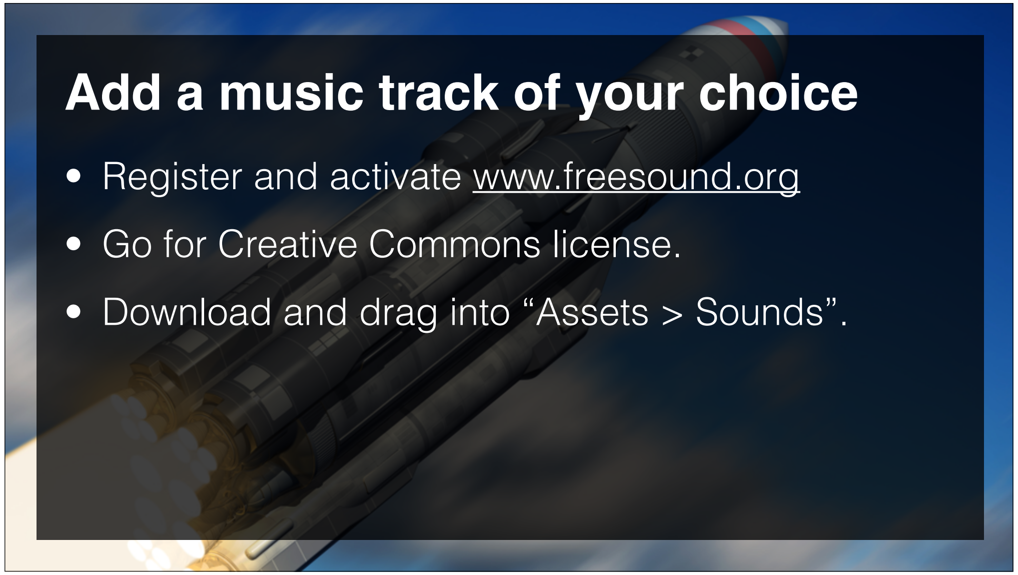
- Get the Start button to go to Level_01.
- Add temporary “Game Over” button to Level_01.
- “Play Again” button to go to Start Menu.
- Test that you can navigate fully.

Playing Background Music



In this lecture…

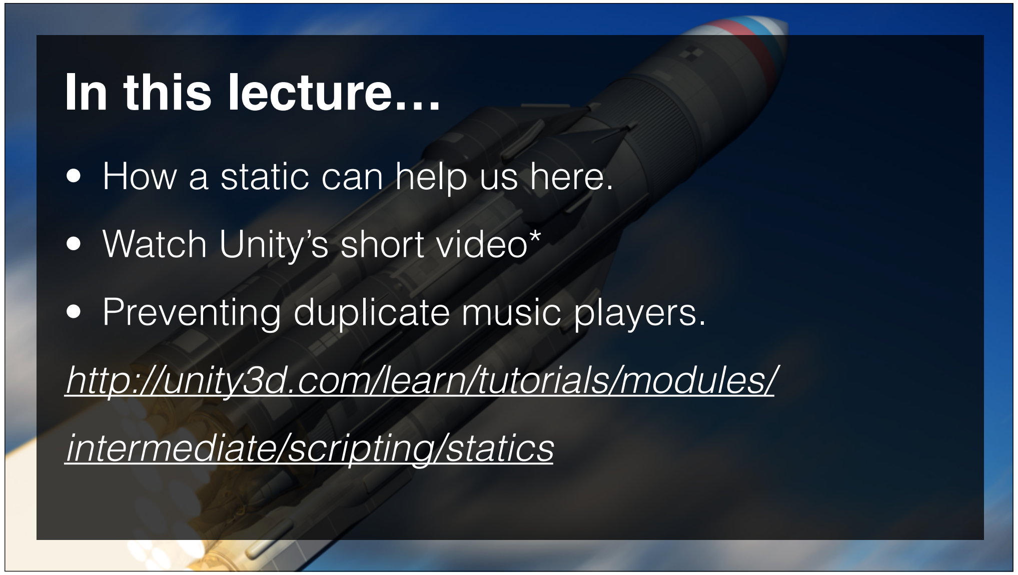
- How persistent music improves quality.
- Add a Music Player Game Object.
- Add a music track of your choice.
- Use `GameObject.DontDestroyOnLoad()`.
- Test your music plays consistently.



Add a music track of your choice

- Register and activate www.freesound.org
- Go for Creative Commons license.
- Download and drag into “Assets > Sounds”.

Introducing Static Variables

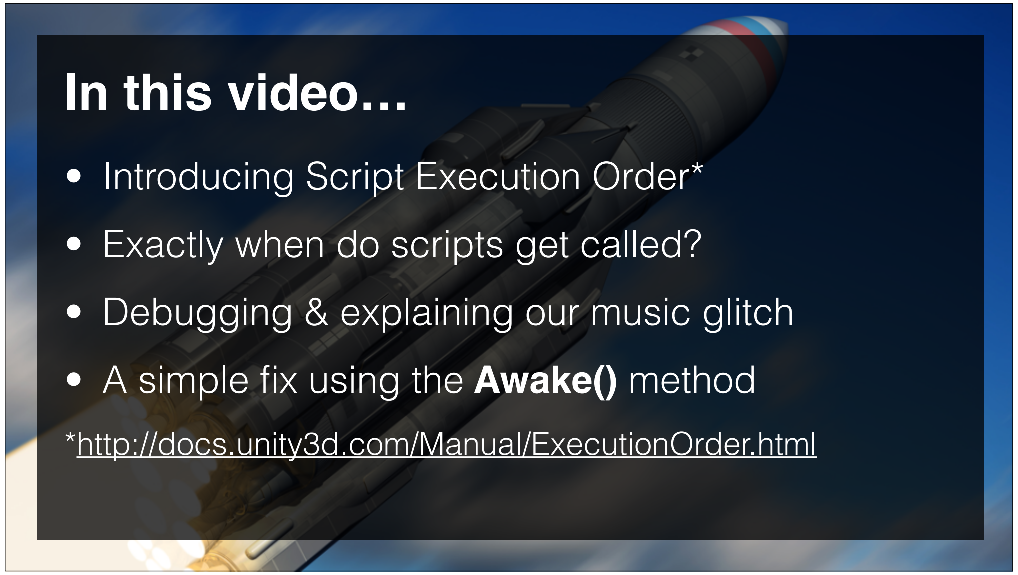


In this lecture…

- How a static can help us here.
- Watch Unity’s short video*
- Preventing duplicate music players.

http://unity3d.com/learn/tutorials/modules/intermediate/scripting/statics

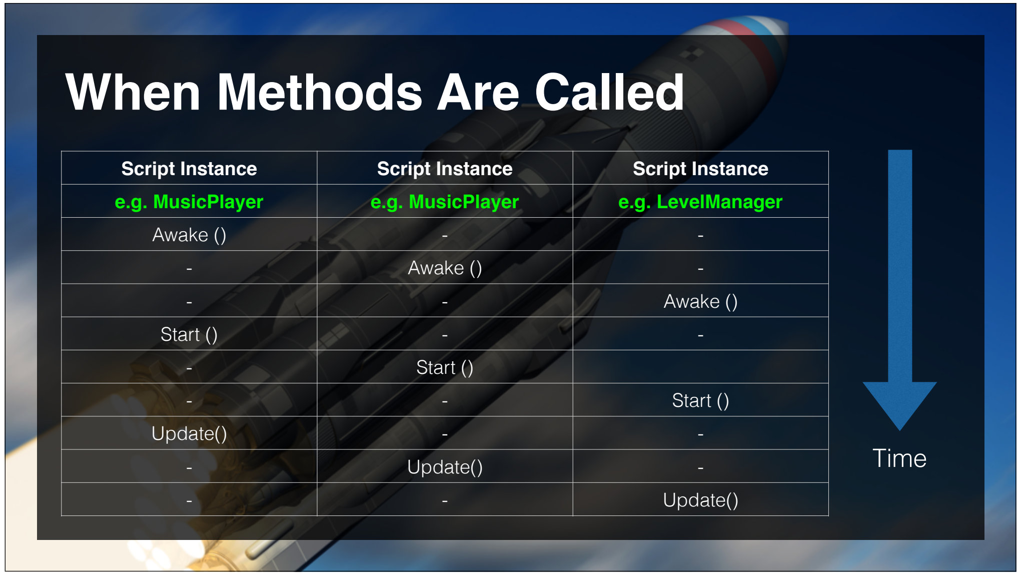
Script Execution Order



In this video…

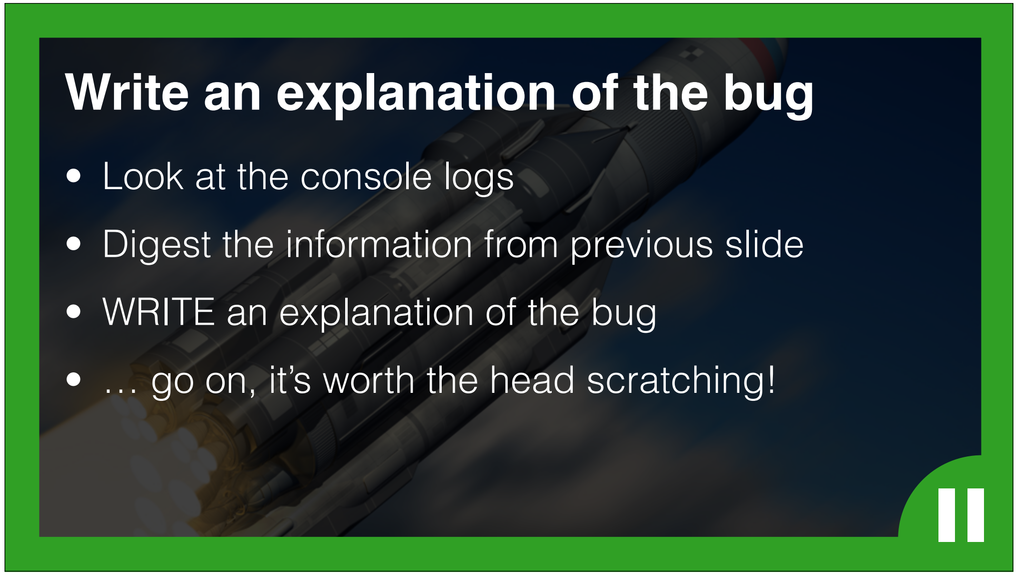
- Introducing Script Execution Order*
- Exactly when do scripts get called?
- Debugging & explaining our music glitch
- A simple fix using the **Awake()** method

*http://docs.unity3d.com/Manual/ExecutionOrder.html



When Methods Are Called

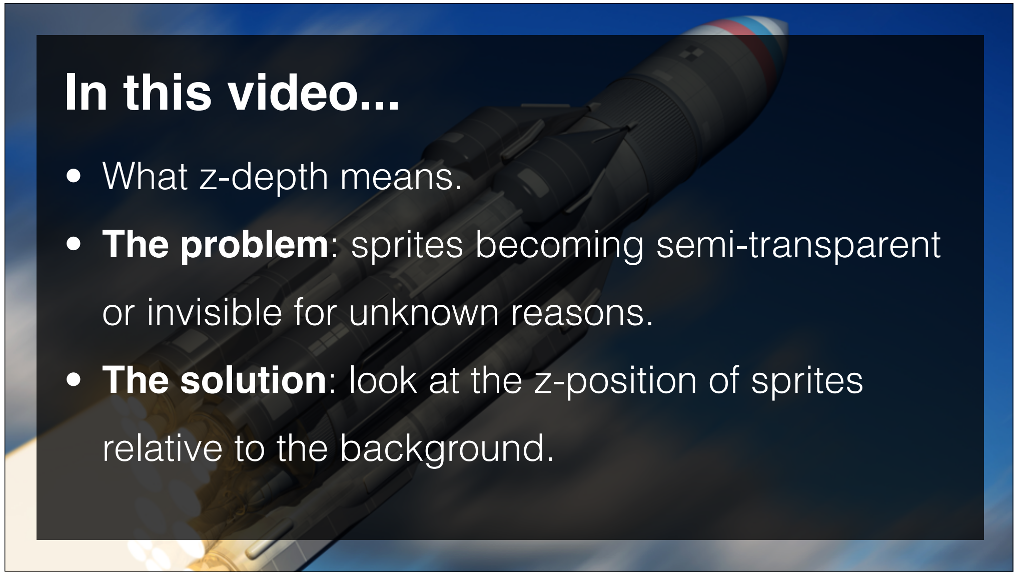
Script Instance	Script Instance	Script Instance
e.g. MusicPlayer	**e.g. MusicPlayer**	**e.g. LevelManager**
Awake ()	-	-
-	Awake ()	-
-	-	Awake ()
Start ()	-	-
-	Start ()	
-	-	Start ()
Update()	-	-
-	Update()	-
-	-	Update()



Write an explanation of the bug

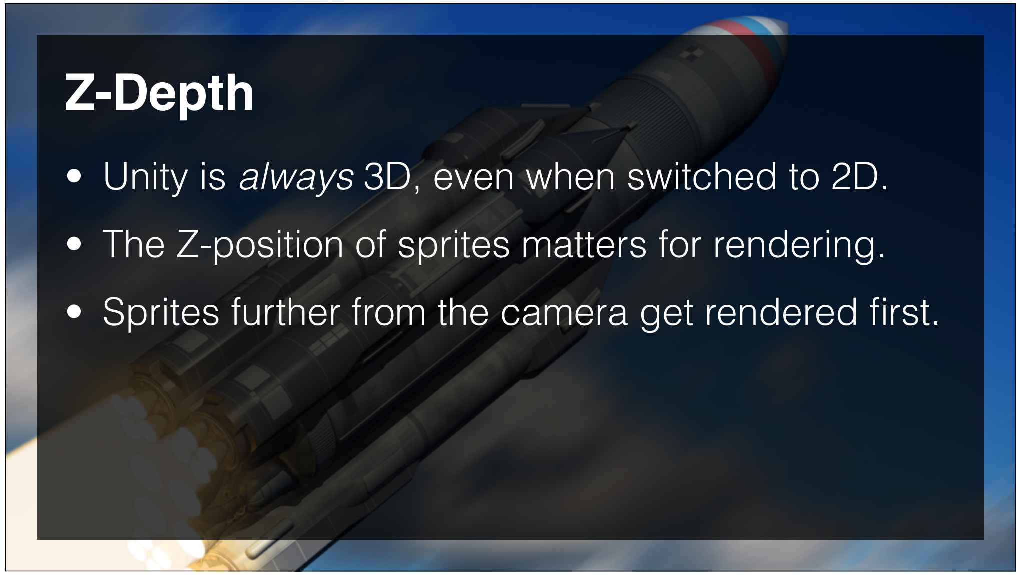
- Look at the console logs
- Digest the information from previous slide
- WRITE an explanation of the bug
- … go on, it's worth the head scratching!

Z-Depth in 2D games



In this video...

- What z-depth means.
- **The problem**: sprites becoming semi-transparent or invisible for unknown reasons.
- **The solution**: look at the z-position of sprites relative to the background.



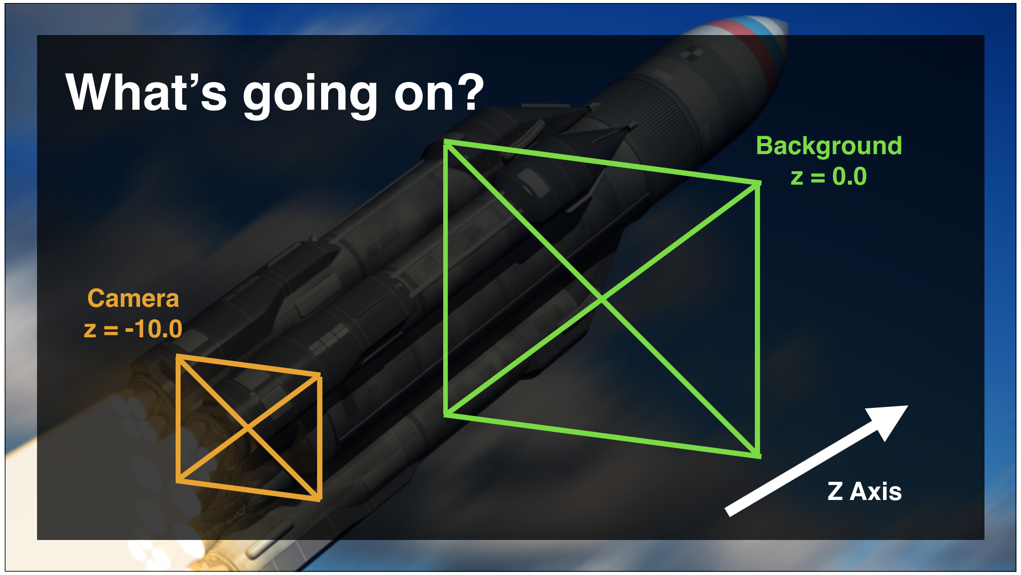
Z-Depth

- Unity is *always* 3D, even when switched to 2D.
- The Z-position of sprites matters for rendering.
- Sprites further from the camera get rendered first.

Typical Issue

“My brick sprites become semi-transparent when the ball hits them.”

“My paddle changes colour when the ball hits it.”

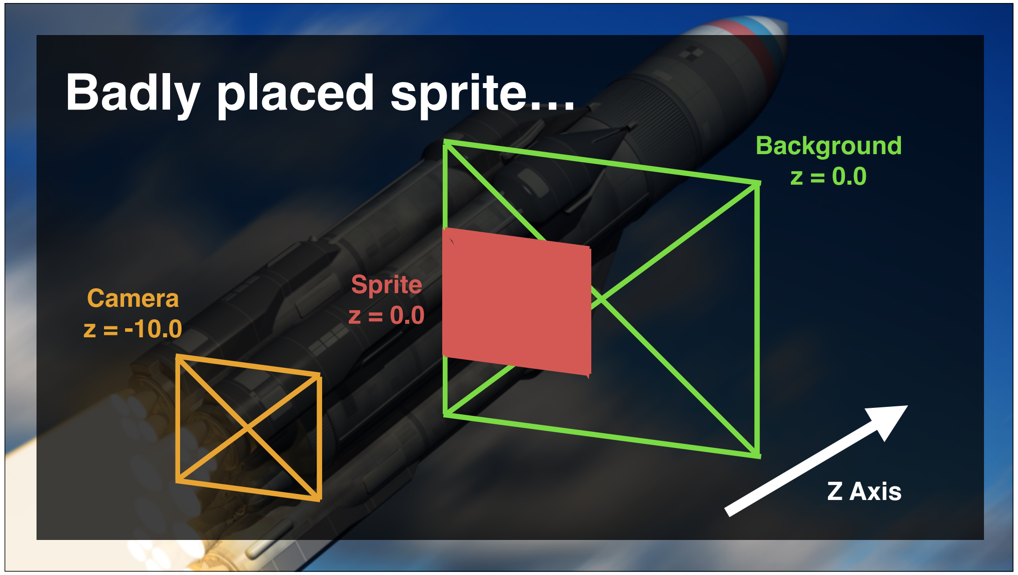


What's going on?

Background
z = 0.0

Camera
z = -10.0

Z Axis



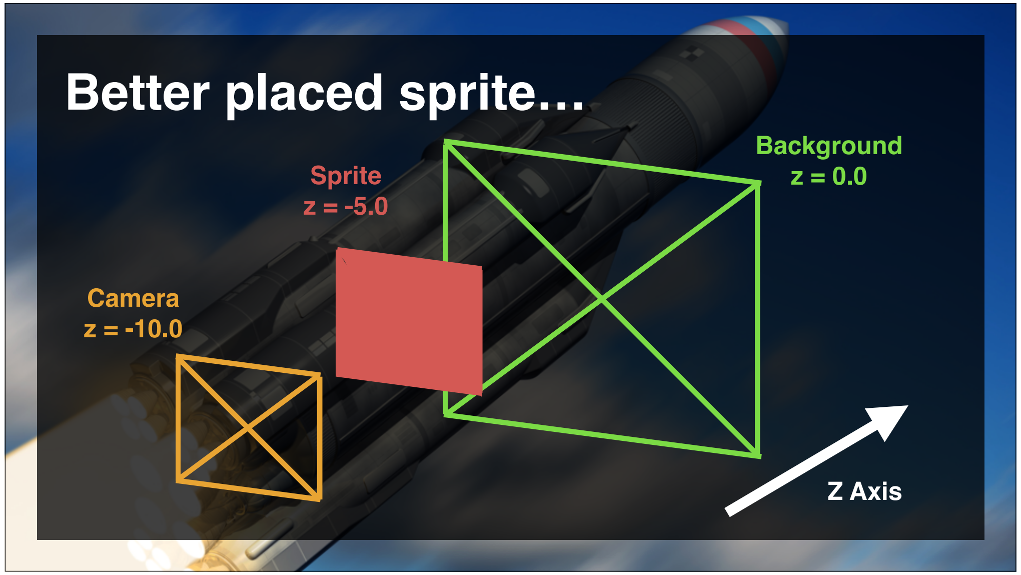
Badly placed sprite…

Background
z = 0.0

Sprite
z = 0.0

Camera
z = -10.0

Z Axis



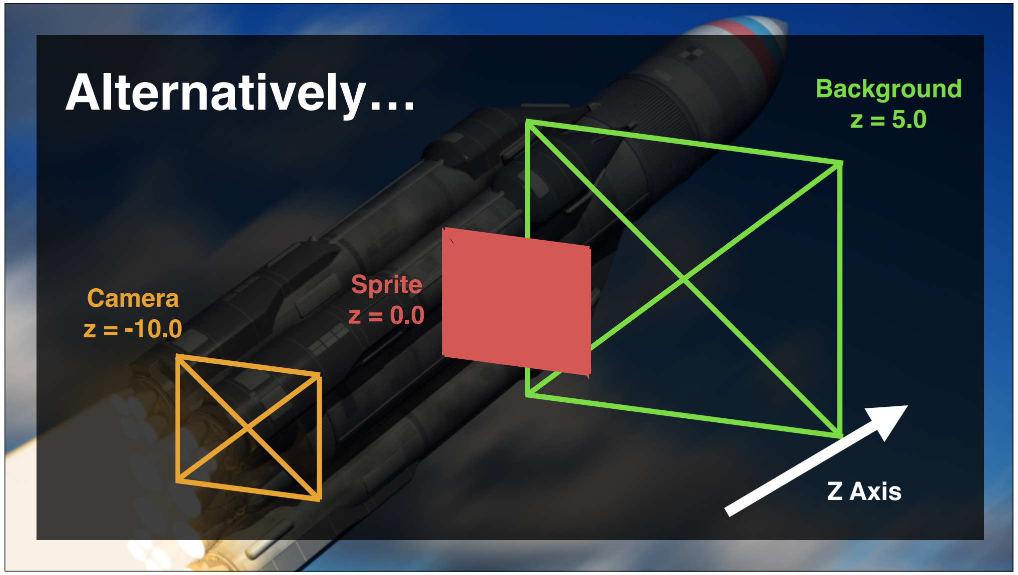
Better placed sprite…

Background
z = 0.0

Sprite
z = -5.0

Camera
z = -10.0

Z Axis



Alternatively…

Background
z = 5.0

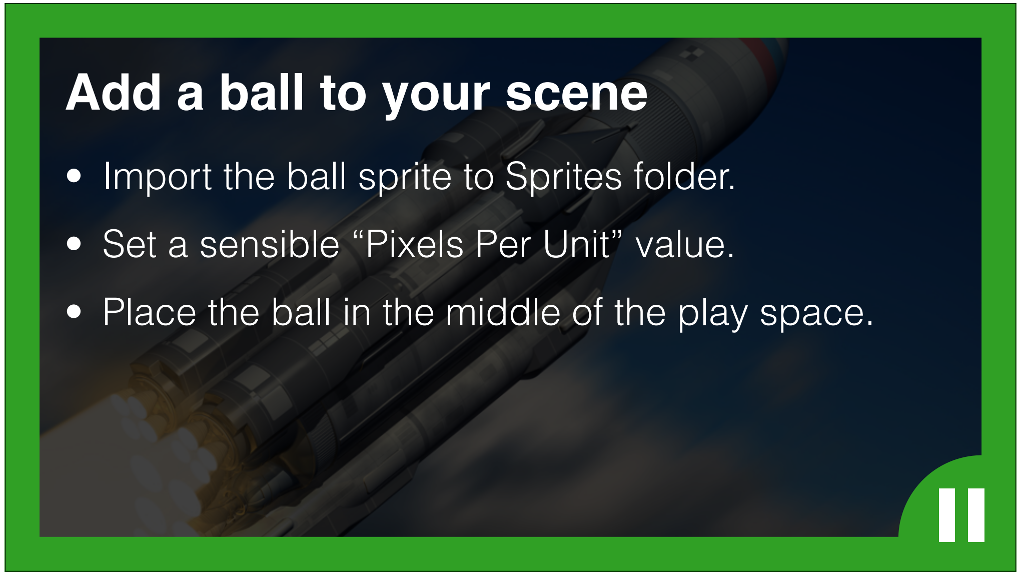
Sprite
z = 0.0

Camera
z = -10.0

Z Axis

Setting Up Your Play Space

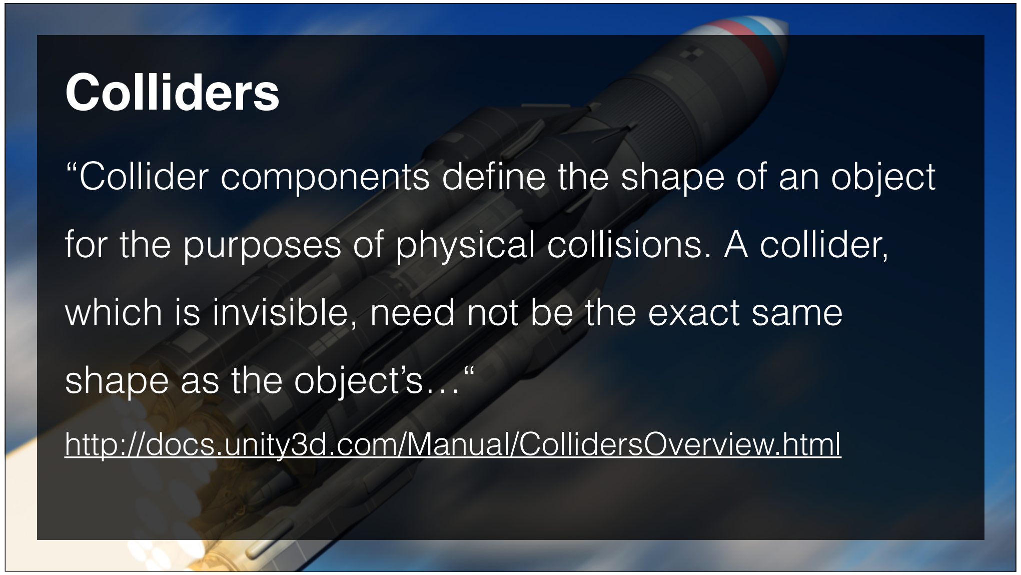
Ball + Gravity + Colliders = Fun



Add a ball to your scene

- Import the ball sprite to Sprites folder.
- Set a sensible “Pixels Per Unit” value.
- Place the ball in the middle of the play space.

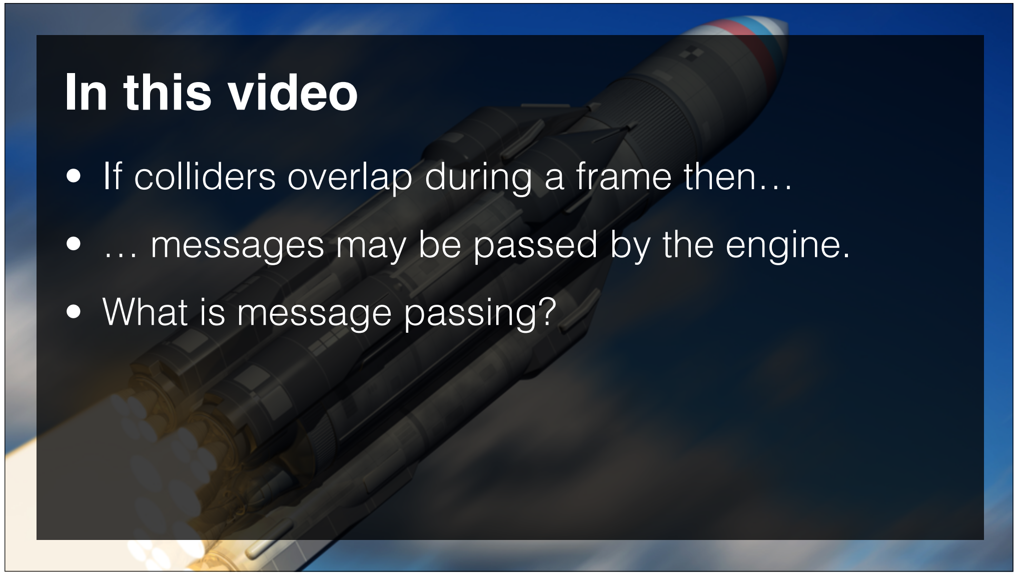
Colliders, Triggers & Collisions in Unity



Colliders

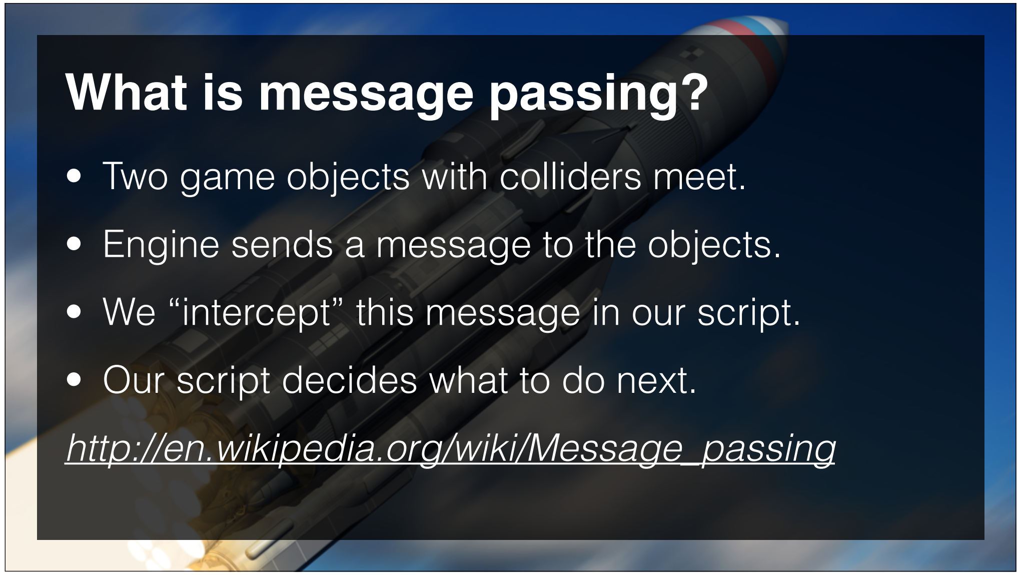
"Collider components define the shape of an object for the purposes of physical collisions. A collider, which is invisible, need not be the exact same shape as the object's…"

http://docs.unity3d.com/Manual/CollidersOverview.html



In this video

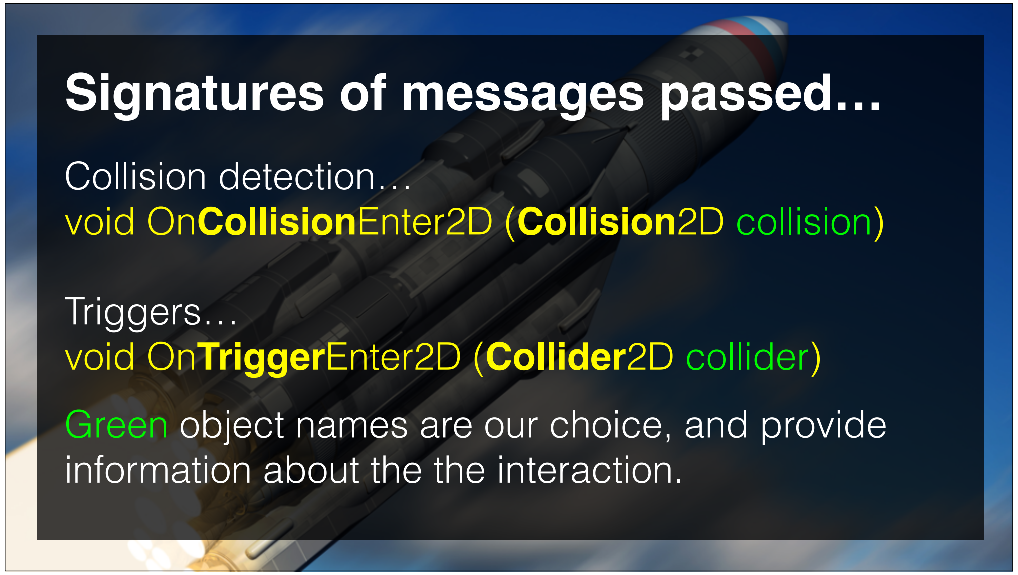
- If colliders overlap during a frame then…
- … messages may be passed by the engine.
- What is message passing?



What is message passing?

- Two game objects with colliders meet.
- Engine sends a message to the objects.
- We "intercept" this message in our script.
- Our script decides what to do next.

http://en.wikipedia.org/wiki/Message_passing

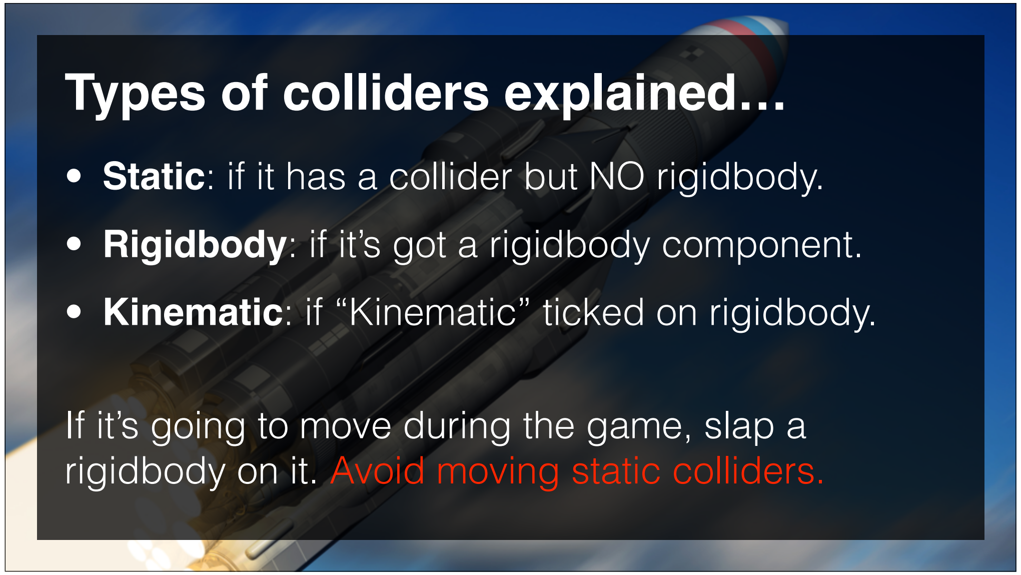


Signatures of messages passed…

Collision detection…
void On**Collision**Enter2D (**Collision**2D collision)

Triggers…
void On**Trigger**Enter2D (**Collider**2D collider)

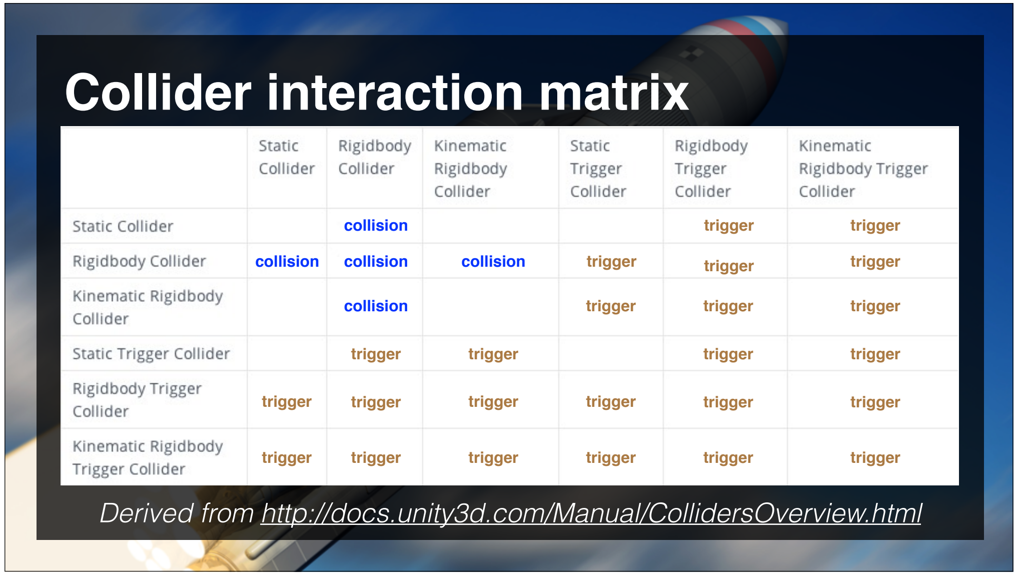
Green object names are our choice, and provide information about the the interaction.



Types of colliders explained…

- **Static**: if it has a collider but NO rigidbody.
- **Rigidbody**: if it's got a rigidbody component.
- **Kinematic**: if "Kinematic" ticked on rigidbody.

If it's going to move during the game, slap a rigidbody on it. Avoid moving static colliders.

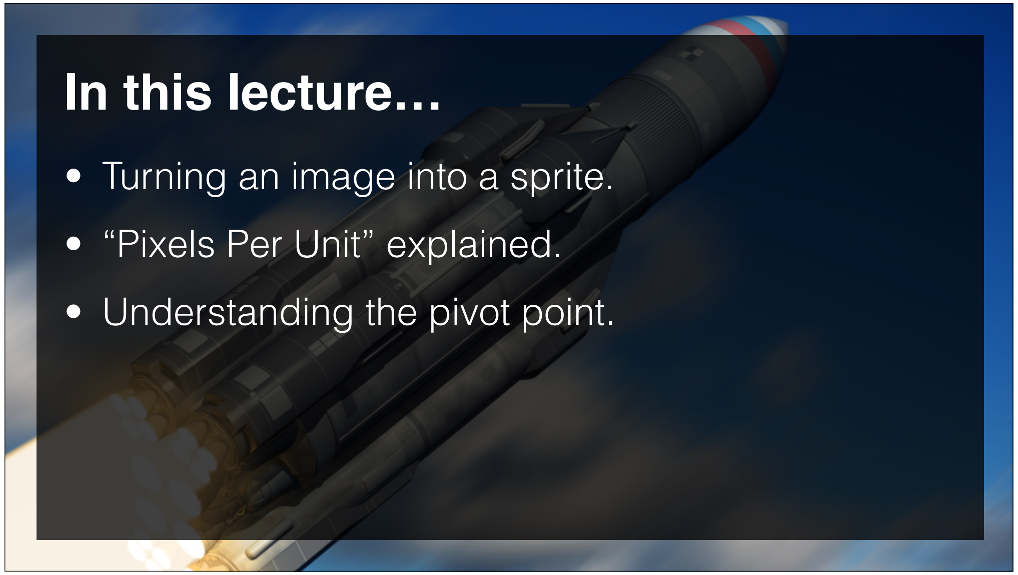


Collider interaction matrix

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		collision			trigger	trigger
Rigidbody Collider	collision	collision	collision	trigger	trigger	trigger
Kinematic Rigidbody Collider		collision		trigger	trigger	trigger
Static Trigger Collider		trigger	trigger		trigger	trigger
Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger
Kinematic Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger

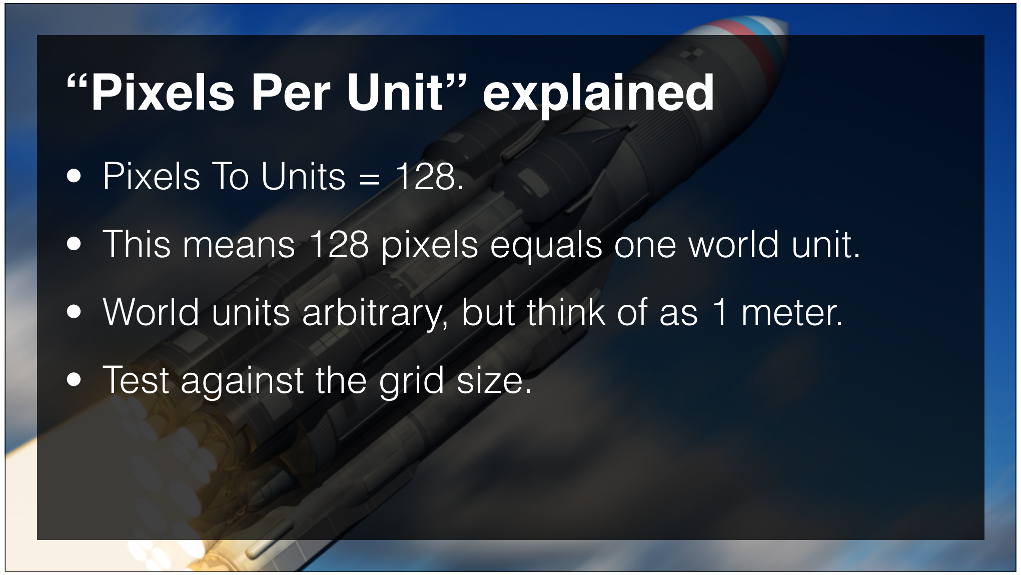
Derived from http://docs.unity3d.com/Manual/CollidersOverview.html

Using Unity's 2D Sprite Editor



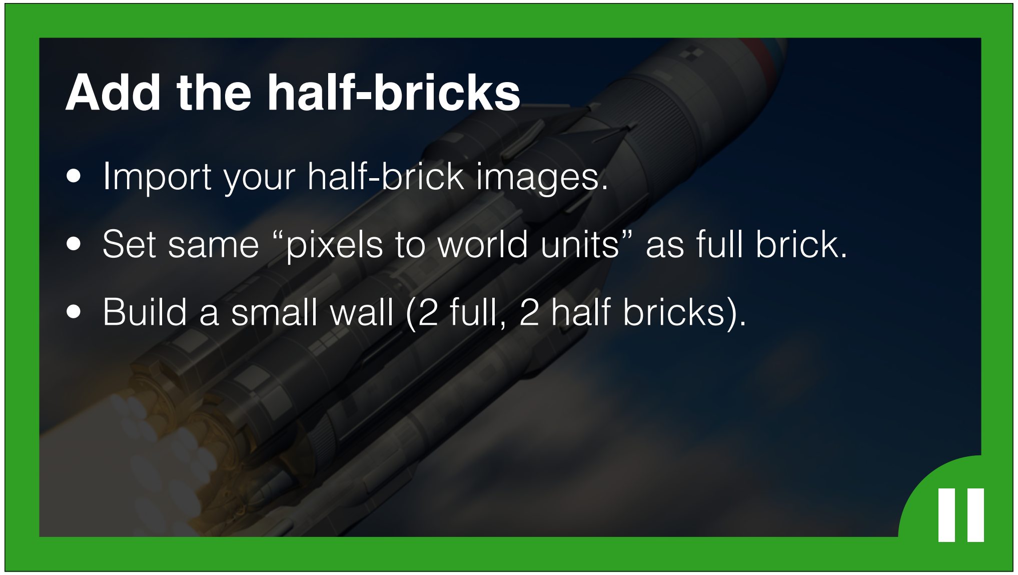
In this lecture…

- Turning an image into a sprite.
- "Pixels Per Unit" explained.
- Understanding the pivot point.



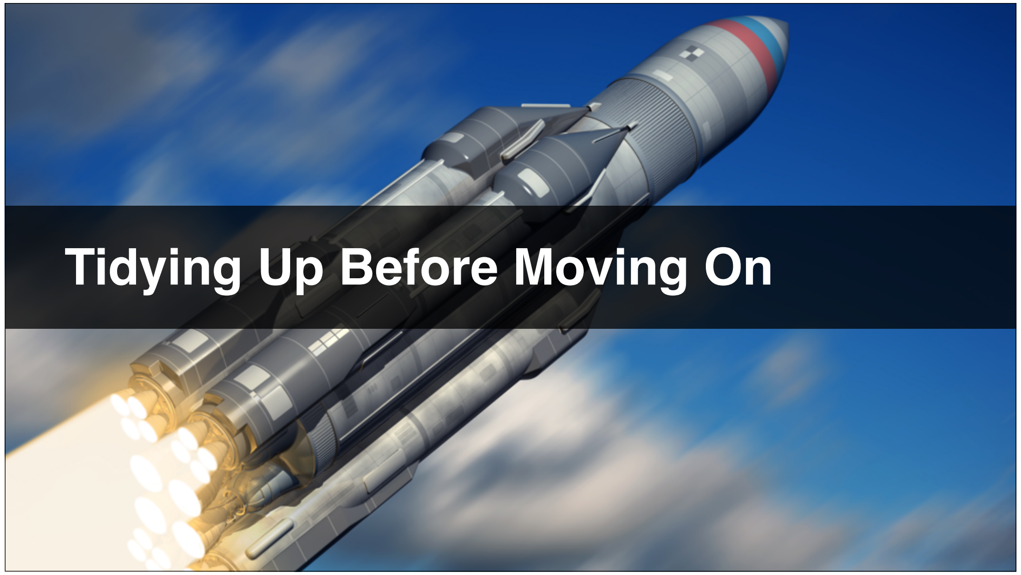
“Pixels Per Unit” explained

- Pixels To Units = 128.
- This means 128 pixels equals one world unit.
- World units arbitrary, but think of as 1 meter.
- Test against the grid size.

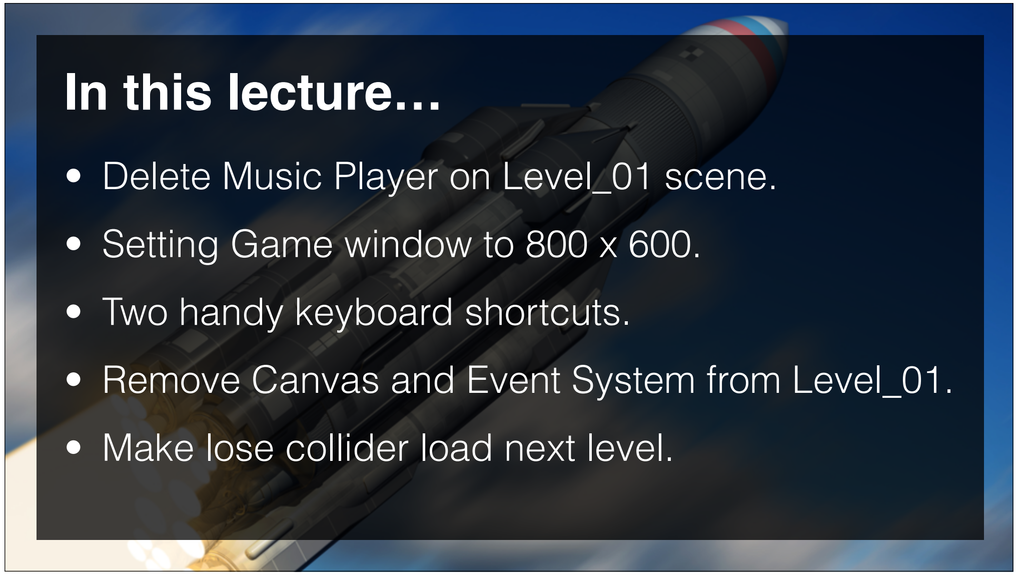


Add the half-bricks

- Import your half-brick images.
- Set same “pixels to world units” as full brick.
- Build a small wall (2 full, 2 half bricks).

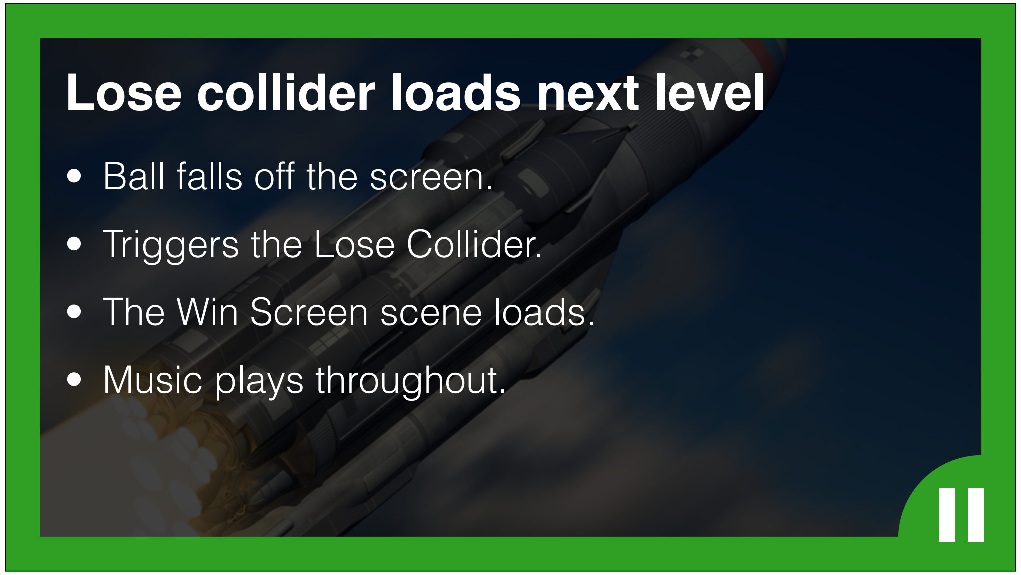


Tidying Up Before Moving On



In this lecture…

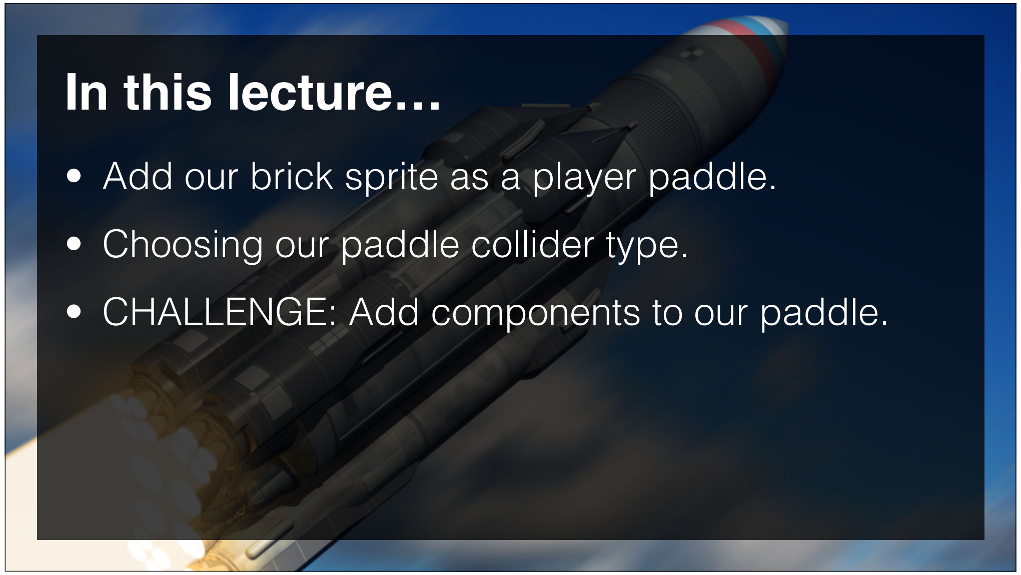
- Delete Music Player on Level_01 scene.
- Setting Game window to 800 x 600.
- Two handy keyboard shortcuts.
- Remove Canvas and Event System from Level_01.
- Make lose collider load next level.



Lose collider loads next level

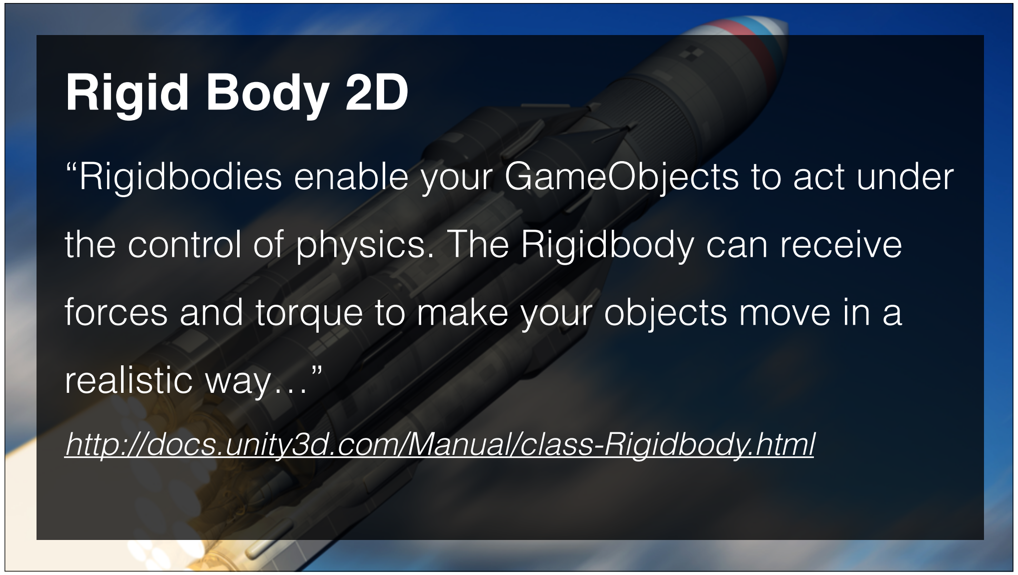
- Ball falls off the screen.
- Triggers the Lose Collider.
- The Win Screen scene loads.
- Music plays throughout.

Choosing The Right Collider



In this lecture…

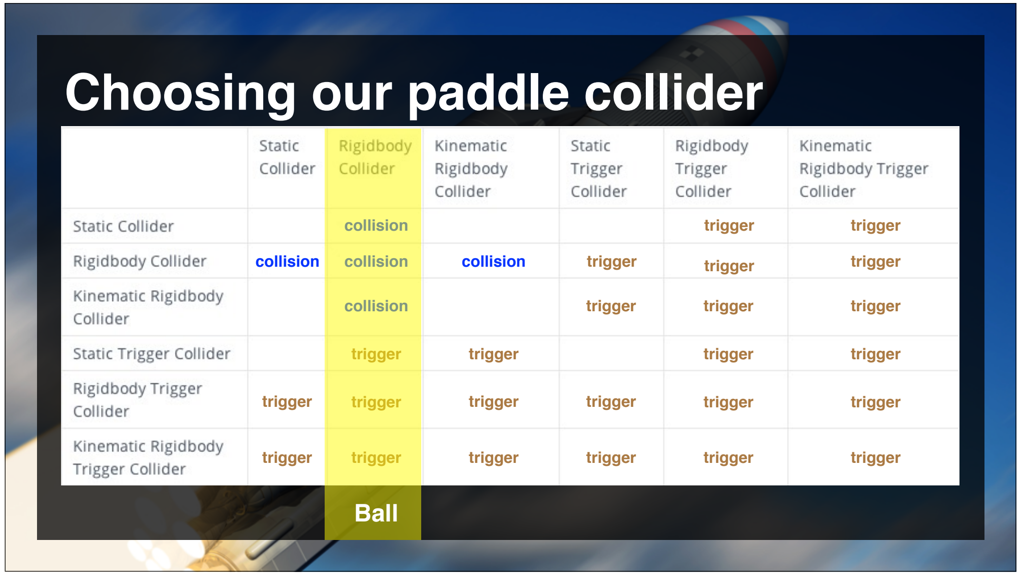
- Add our brick sprite as a player paddle.
- Choosing our paddle collider type.
- CHALLENGE: Add components to our paddle.



Rigid Body 2D

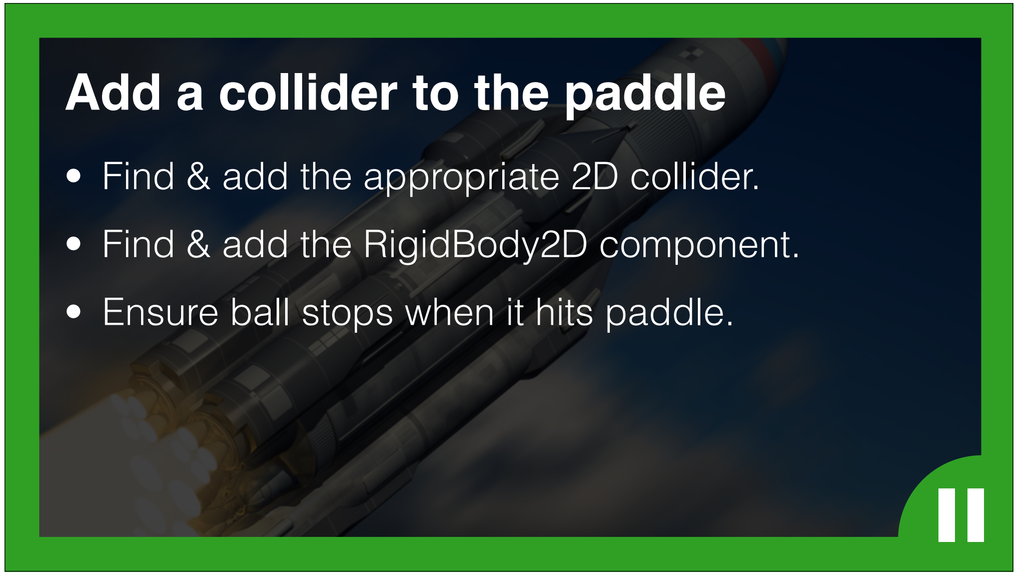
“Rigidbodies enable your GameObjects to act under the control of physics. The Rigidbody can receive forces and torque to make your objects move in a realistic way…”

http://docs.unity3d.com/Manual/class-Rigidbody.html



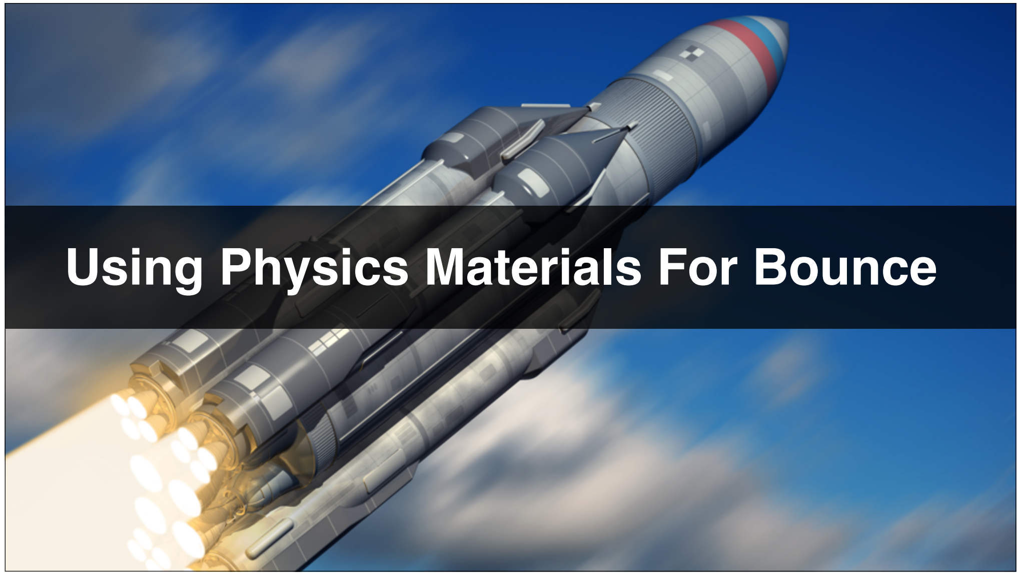
Choosing our paddle collider

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		collision			trigger	trigger
Rigidbody Collider	collision	collision	collision	trigger	trigger	trigger
Kinematic Rigidbody Collider		collision		trigger	trigger	trigger
Static Trigger Collider		trigger	trigger		trigger	trigger
Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger
Kinematic Rigidbody Trigger Collider	trigger	trigger	trigger	trigger	trigger	trigger
		Ball				

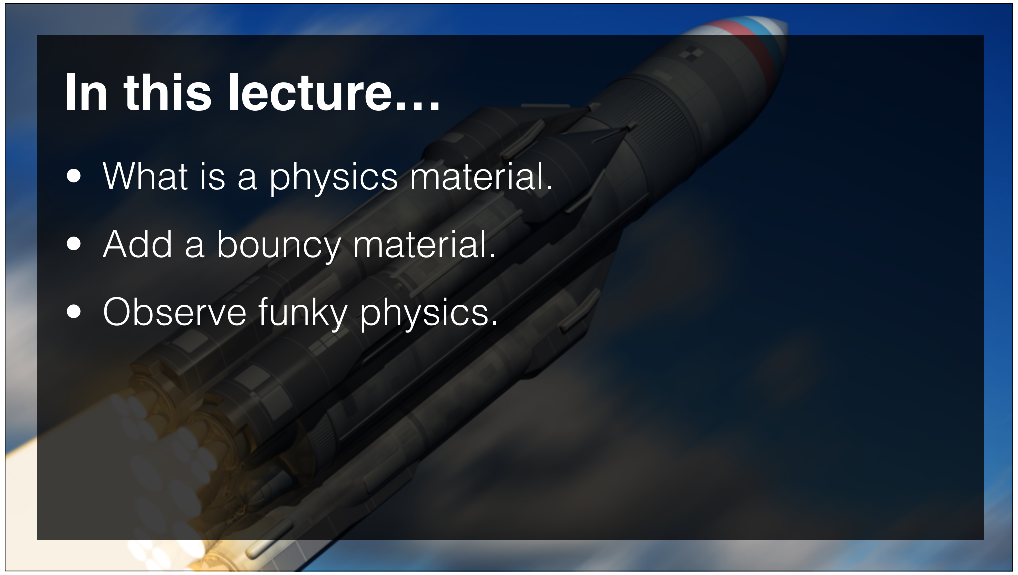


Add a collider to the paddle

- Find & add the appropriate 2D collider.
- Find & add the RigidBody2D component.
- Ensure ball stops when it hits paddle.

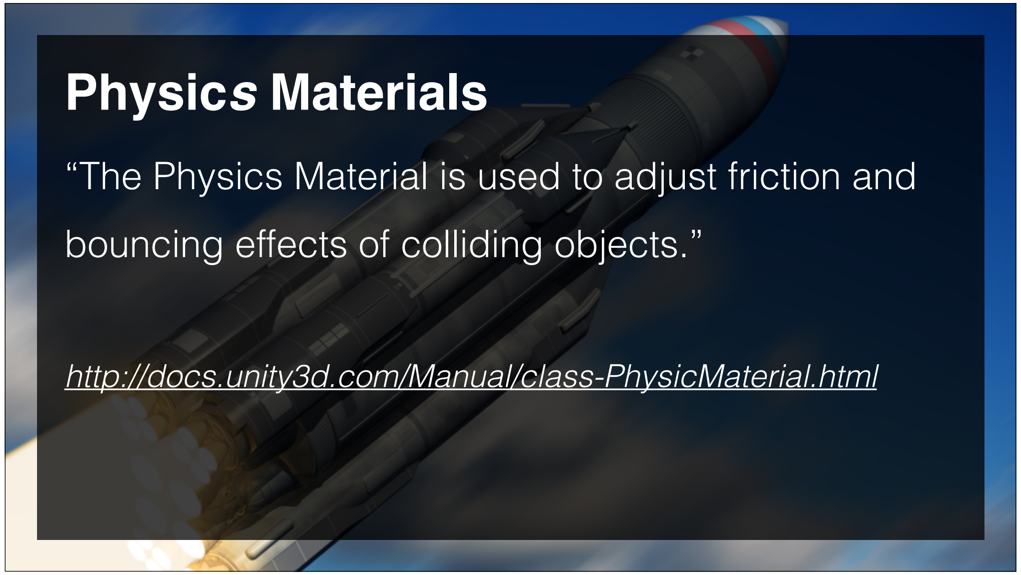


Using Physics Materials For Bounce



In this lecture…

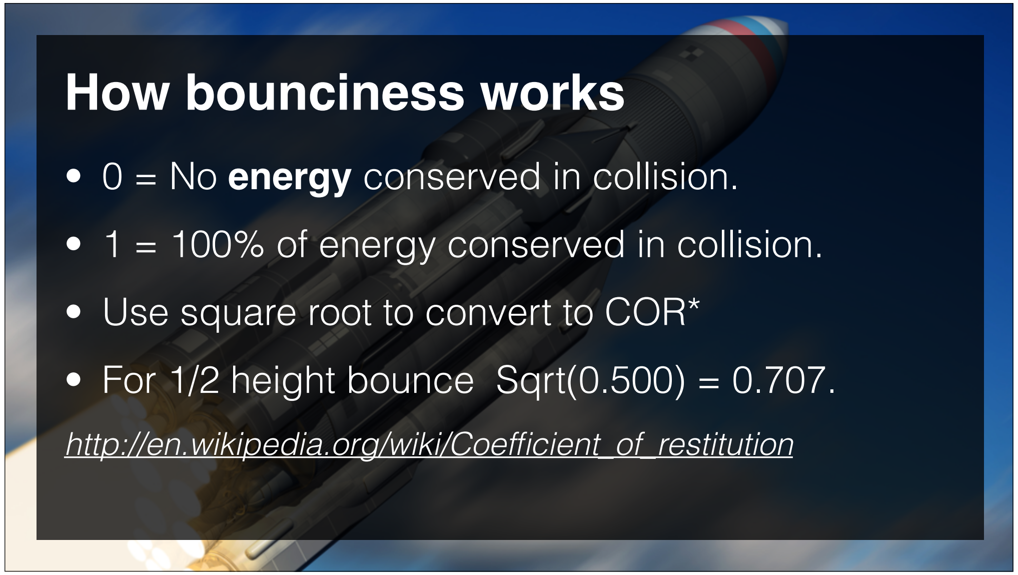
- What is a physics material.
- Add a bouncy material.
- Observe funky physics.



Physics Materials

"The Physics Material is used to adjust friction and bouncing effects of colliding objects."

http://docs.unity3d.com/Manual/class-PhysicMaterial.html

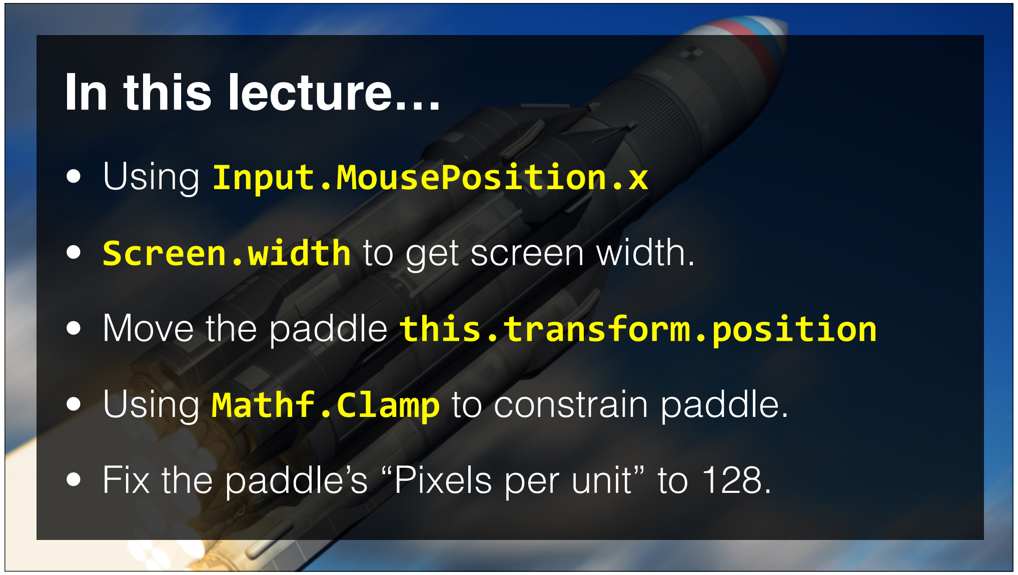


How bounciness works

- 0 = No **energy** conserved in collision.
- 1 = 100% of energy conserved in collision.
- Use square root to convert to COR*
- For 1/2 height bounce Sqrt(0.500) = 0.707.

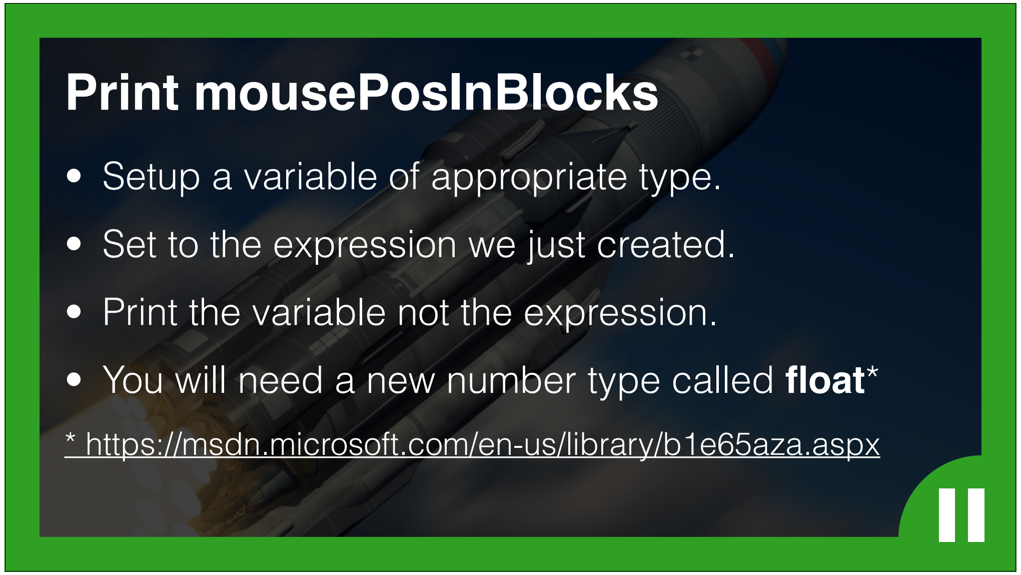
http://en.wikipedia.org/wiki/Coefficient_of_restitution

GameObject Movement By Mouse



In this lecture…

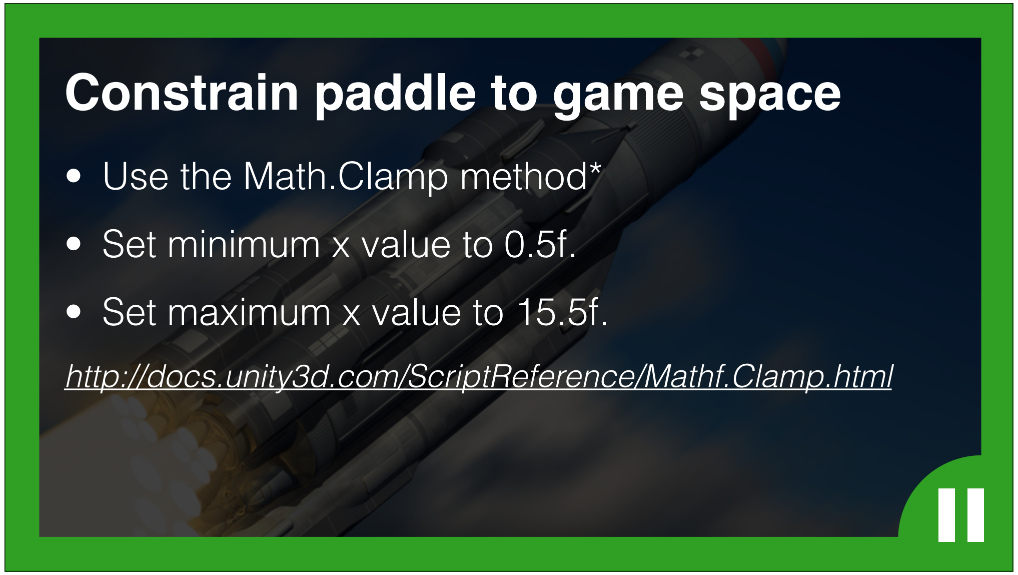
- Using `Input.MousePosition.x`
- `Screen.width` to get screen width.
- Move the paddle `this.transform.position`
- Using `Mathf.Clamp` to constrain paddle.
- Fix the paddle's "Pixels per unit" to 128.



Print mousePosInBlocks

- Setup a variable of appropriate type.
- Set to the expression we just created.
- Print the variable not the expression.
- You will need a new number type called **float***

* https://msdn.microsoft.com/en-us/library/b1e65aza.aspx

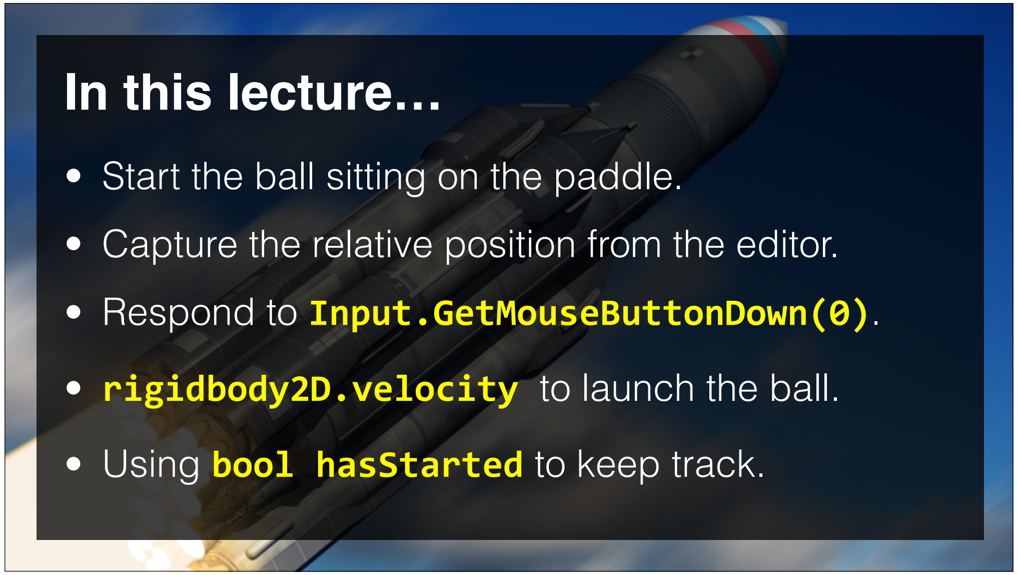


Constrain paddle to game space

- Use the Math.Clamp method*
- Set minimum x value to 0.5f.
- Set maximum x value to 15.5f.

http://docs.unity3d.com/ScriptReference/Mathf.Clamp.html

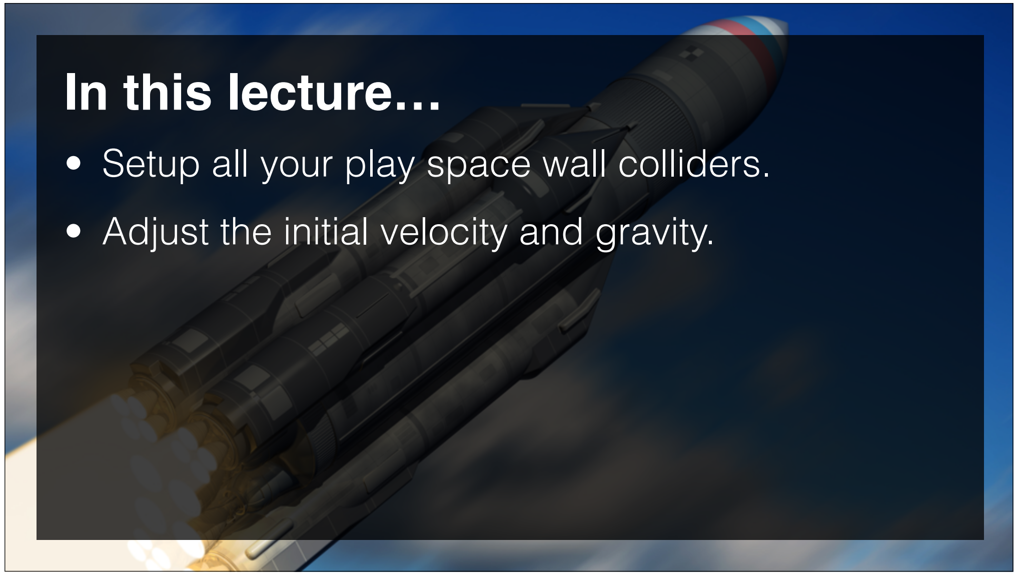
Launching Ball On Mouse Click



In this lecture…

- Start the ball sitting on the paddle.
- Capture the relative position from the editor.
- Respond to `Input.GetMouseButtonDown(0)`.
- `rigidbody2D.velocity` to launch the ball.
- Using `bool hasStarted` to keep track.

Invisible Colliders & Gravity Scale

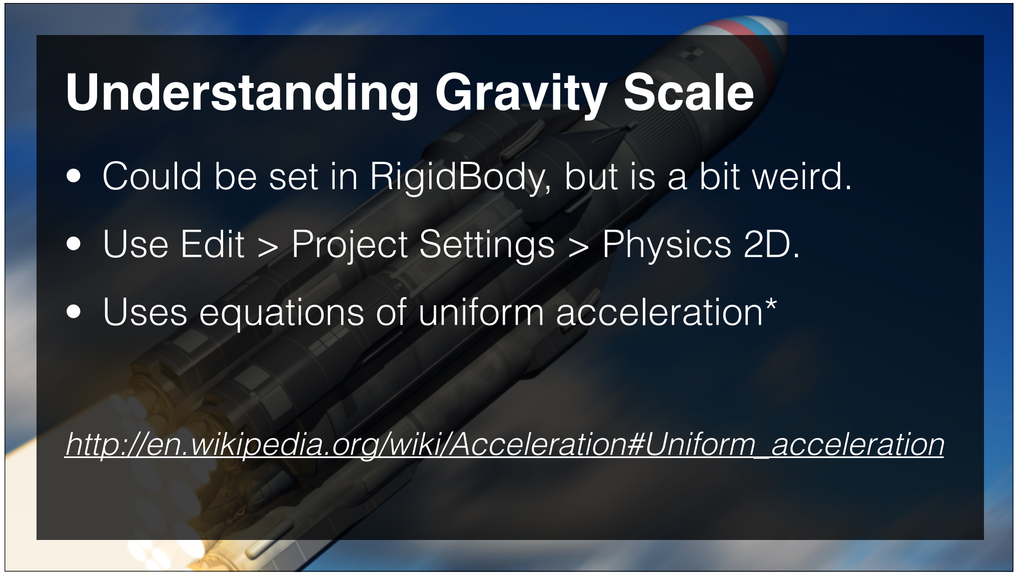


In this lecture…

- Setup all your play space wall colliders.
- Adjust the initial velocity and gravity.

Add Top and Right Colliders

- Add colliders of the same width.
- Ensure there are no spaces.
- Test by playing.

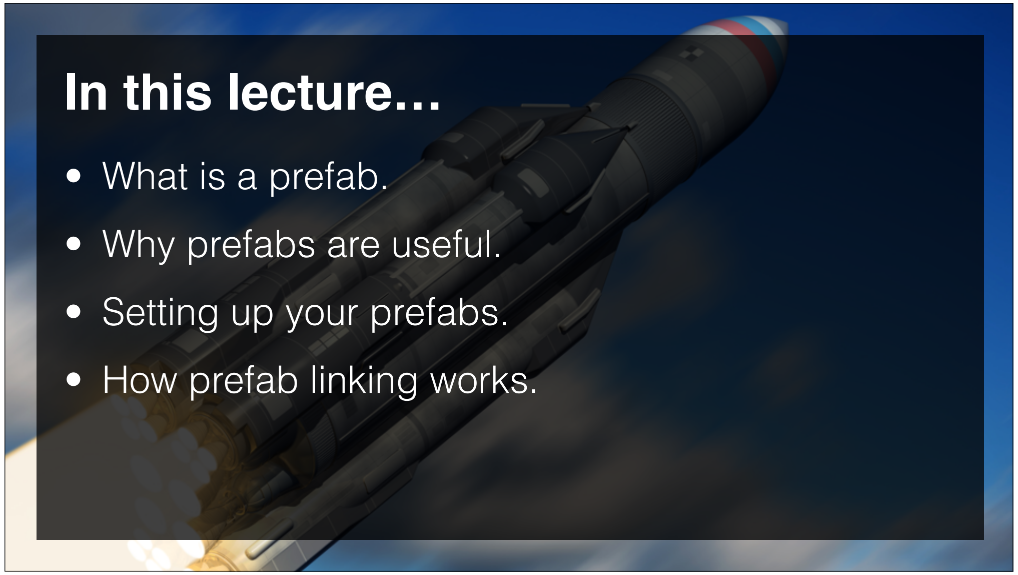


Understanding Gravity Scale

- Could be set in RigidBody, but is a bit weird.
- Use Edit > Project Settings > Physics 2D.
- Uses equations of uniform acceleration*

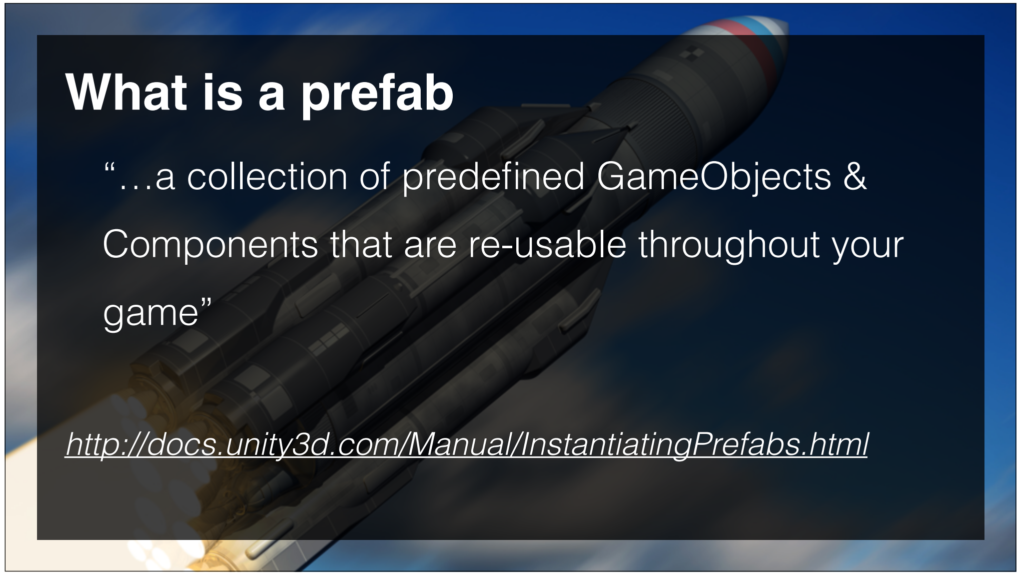
http://en.wikipedia.org/wiki/Acceleration#Uniform_acceleration

Creating & Using Unity Prefabs



In this lecture…

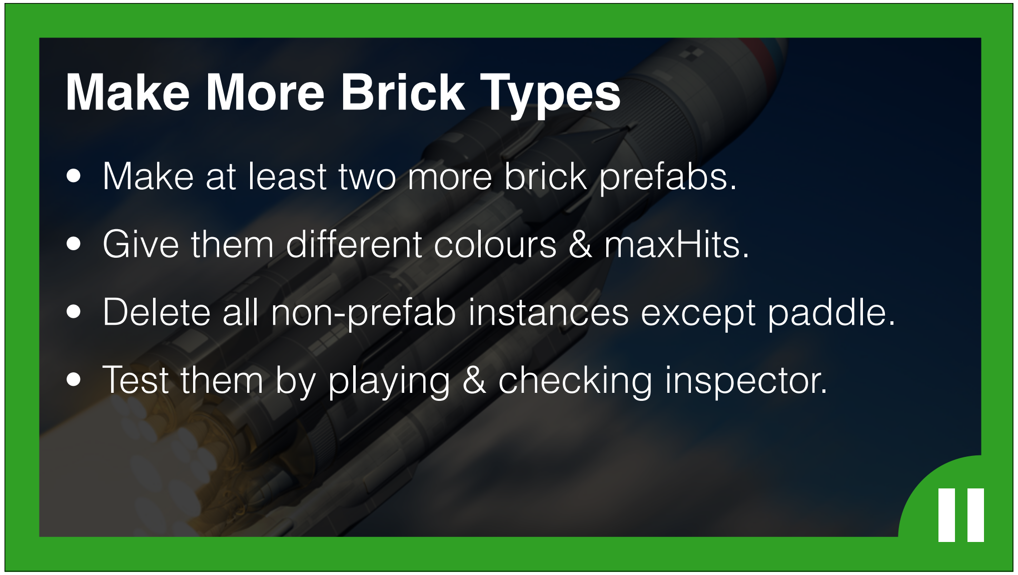
- What is a prefab.
- Why prefabs are useful.
- Setting up your prefabs.
- How prefab linking works.



What is a prefab

> “…a collection of predefined GameObjects & Components that are re-usable throughout your game”

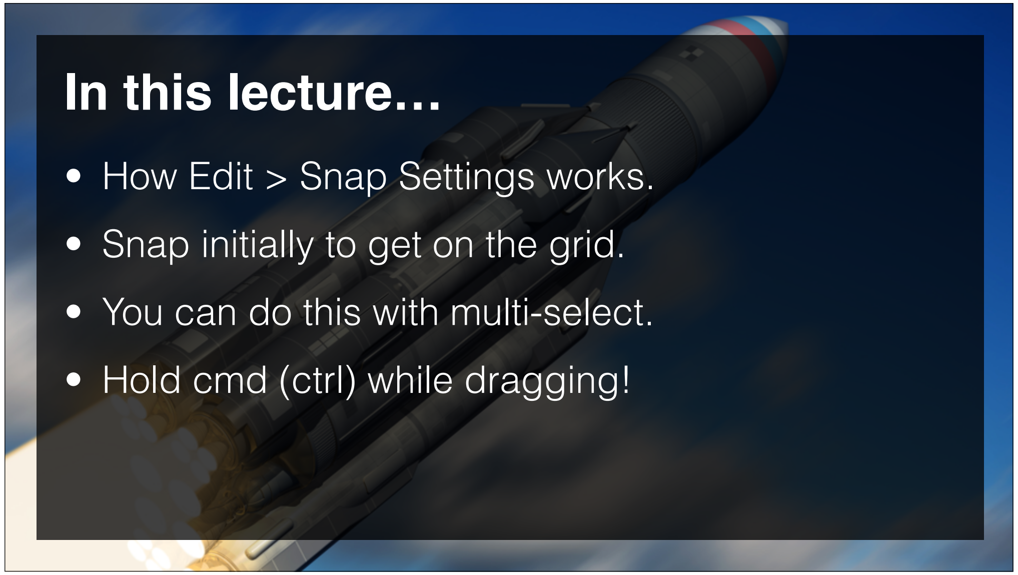
http://docs.unity3d.com/Manual/InstantiatingPrefabs.html



Make More Brick Types

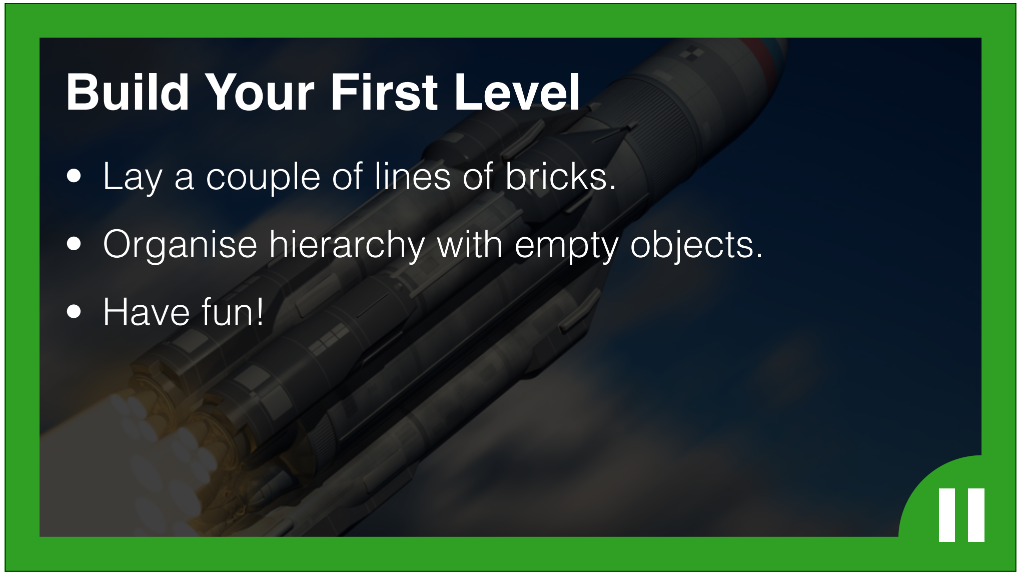
- Make at least two more brick prefabs.
- Give them different colours & maxHits.
- Delete all non-prefab instances except paddle.
- Test them by playing & checking inspector.

Unity Editor Snap To Grid



In this lecture…

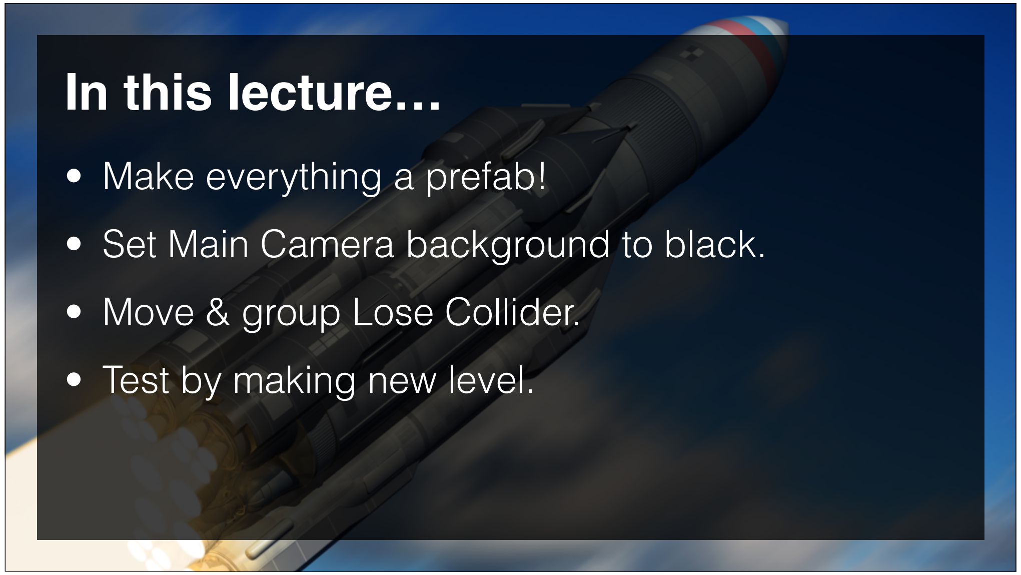
- How Edit > Snap Settings works.
- Snap initially to get on the grid.
- You can do this with multi-select.
- Hold cmd (ctrl) while dragging!



Build Your First Level

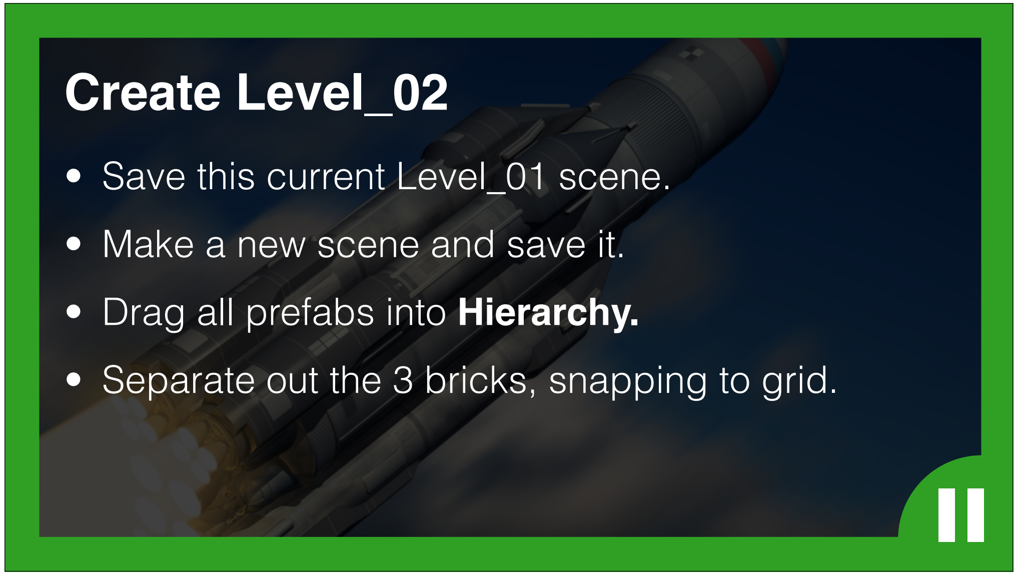
- Lay a couple of lines of bricks.
- Organise hierarchy with empty objects.
- Have fun!

Making Everything A Prefab



In this lecture…

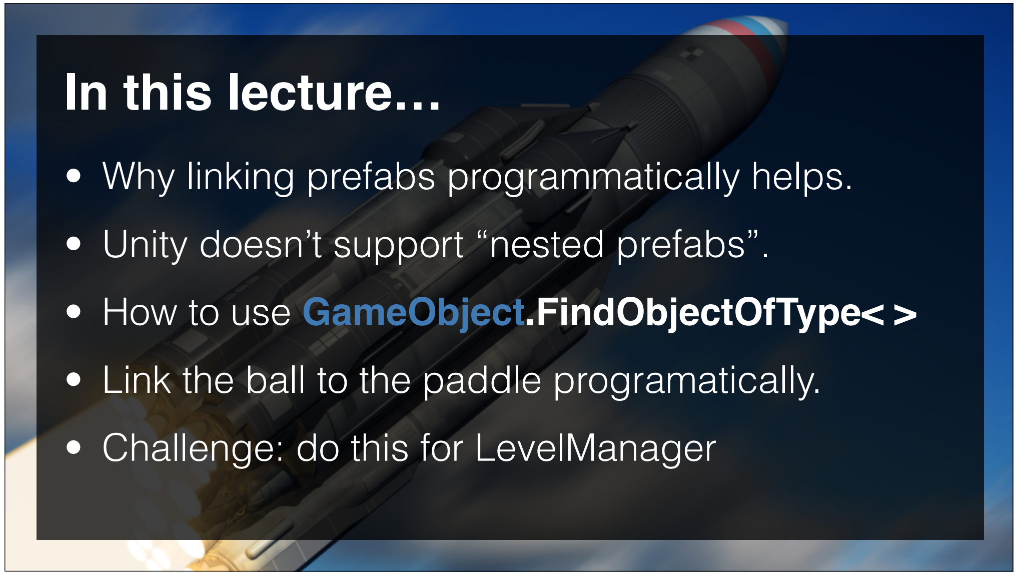
- Make everything a prefab!
- Set Main Camera background to black.
- Move & group Lose Collider.
- Test by making new level.



Create Level_02

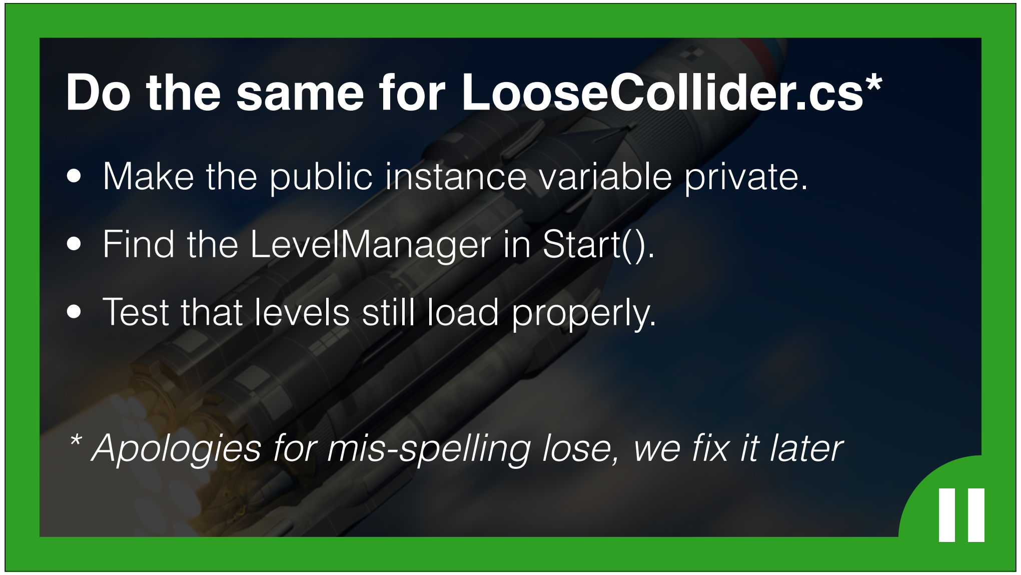
- Save this current Level_01 scene.
- Make a new scene and save it.
- Drag all prefabs into **Hierarchy.**
- Separate out the 3 bricks, snapping to grid.

Using GameObject.FindObjectOfType



In this lecture…

- Why linking prefabs programmatically helps.
- Unity doesn’t support “nested prefabs”.
- How to use **GameObject.FindObjectOfType< >**
- Link the ball to the paddle programatically.
- Challenge: do this for LevelManager

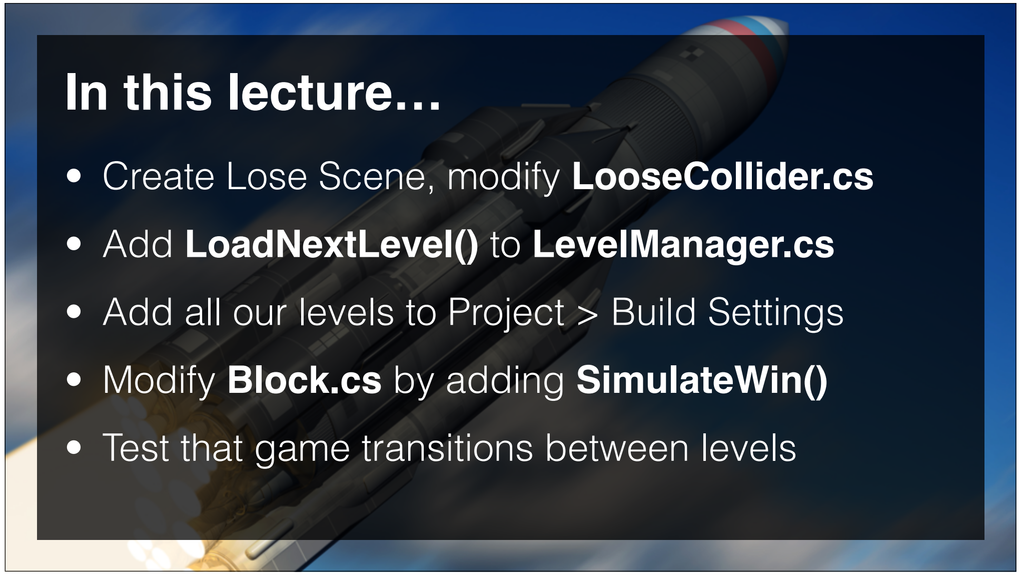


Do the same for LooseCollider.cs*

- Make the public instance variable private.
- Find the LevelManager in Start().
- Test that levels still load properly.

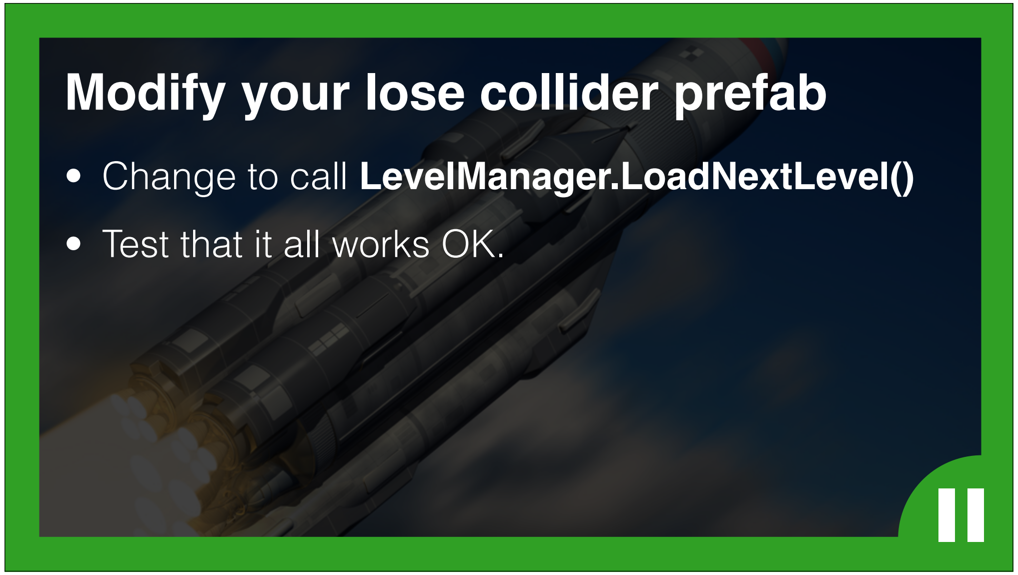
** Apologies for mis-spelling lose, we fix it later*

Level Management & Build Order



In this lecture…

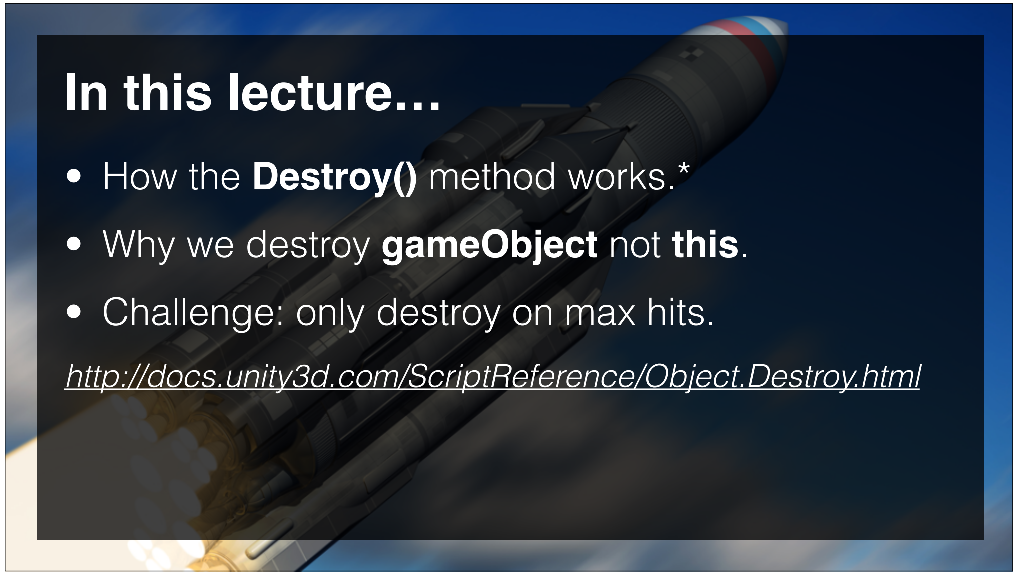
- Create Lose Scene, modify **LooseCollider.cs**
- Add **LoadNextLevel()** to **LevelManager.cs**
- Add all our levels to Project > Build Settings
- Modify **Block.cs** by adding **SimulateWin()**
- Test that game transitions between levels



Modify your lose collider prefab

- Change to call **LevelManager.LoadNextLevel()**
- Test that it all works OK.

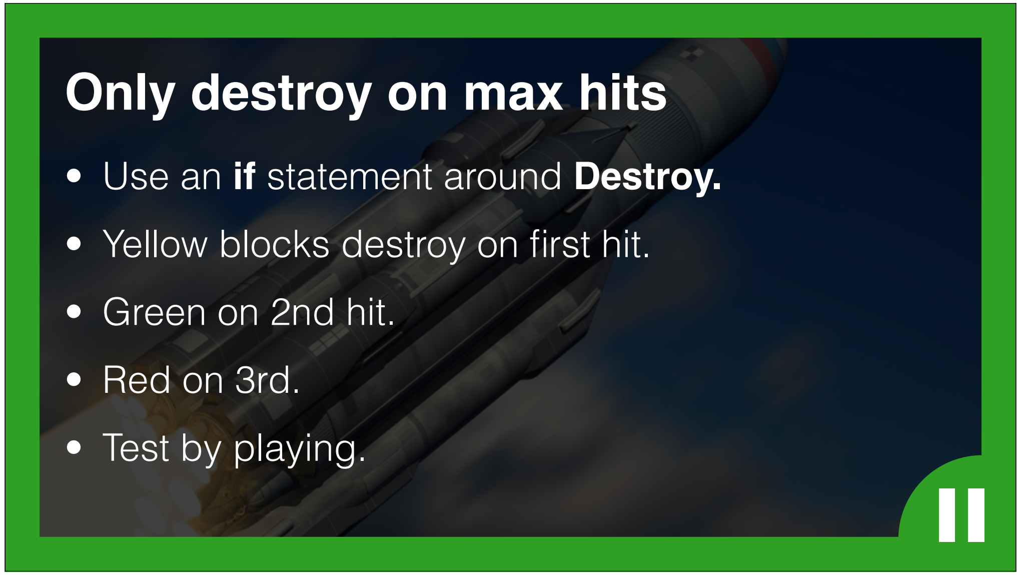
Destroying gameObjects When Hit



In this lecture…

- How the **Destroy()** method works.*
- Why we destroy **gameObject** not **this**.
- Challenge: only destroy on max hits.

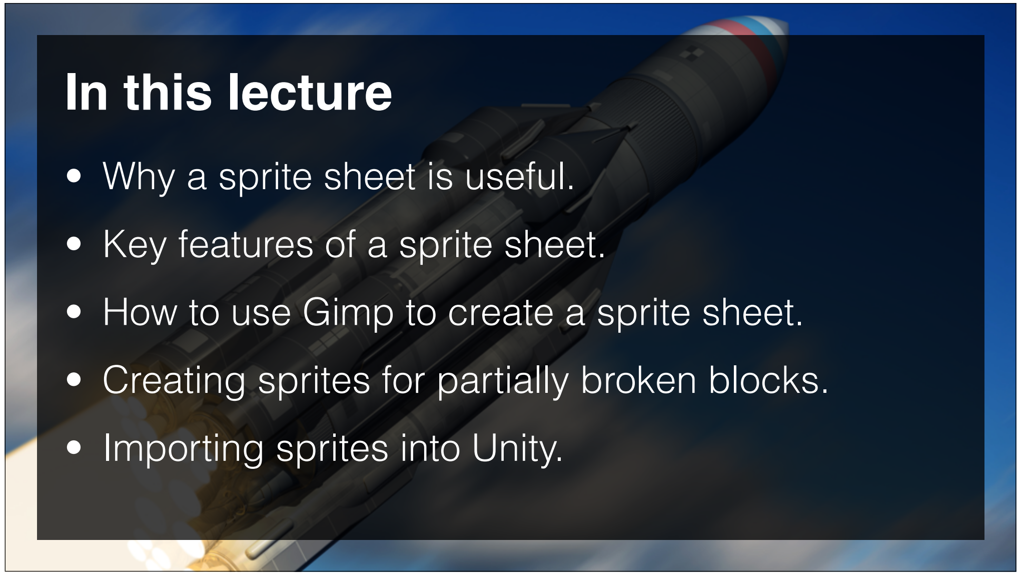
http://docs.unity3d.com/ScriptReference/Object.Destroy.html



Only destroy on max hits

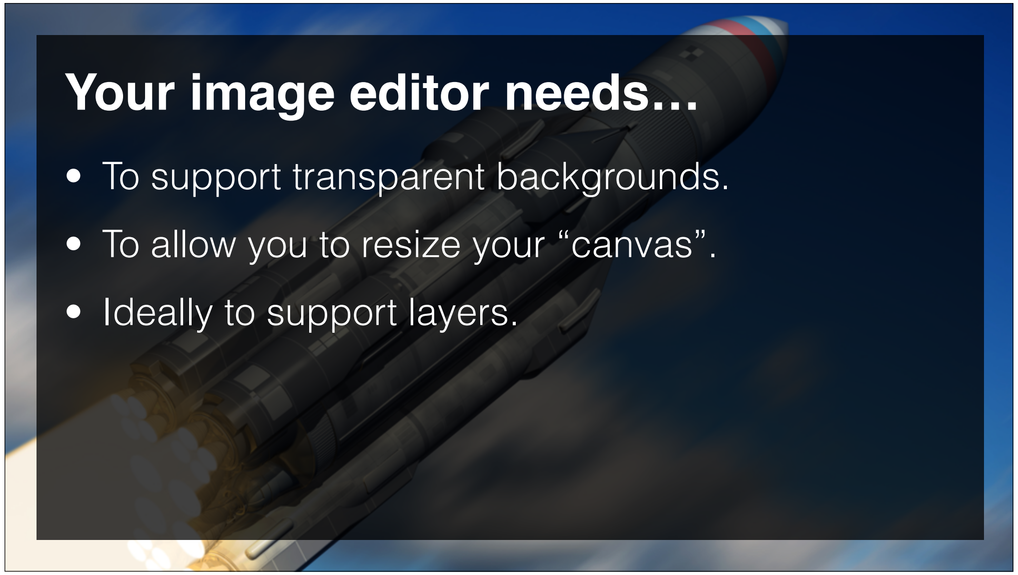
- Use an **if** statement around **Destroy.**
- Yellow blocks destroy on first hit.
- Green on 2nd hit.
- Red on 3rd.
- Test by playing.

Creating & Importing Sprite Sheets



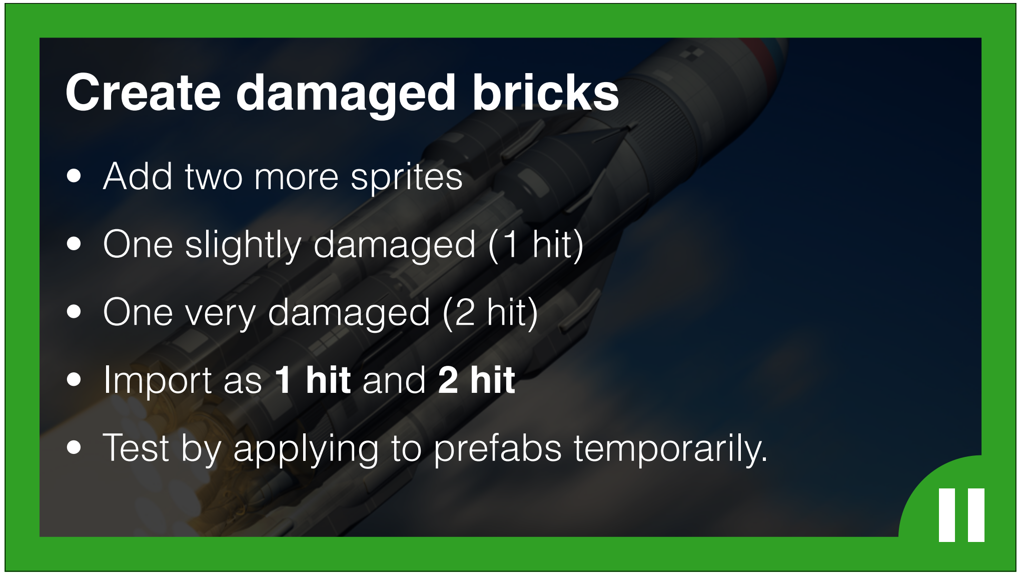
In this lecture

- Why a sprite sheet is useful.
- Key features of a sprite sheet.
- How to use Gimp to create a sprite sheet.
- Creating sprites for partially broken blocks.
- Importing sprites into Unity.



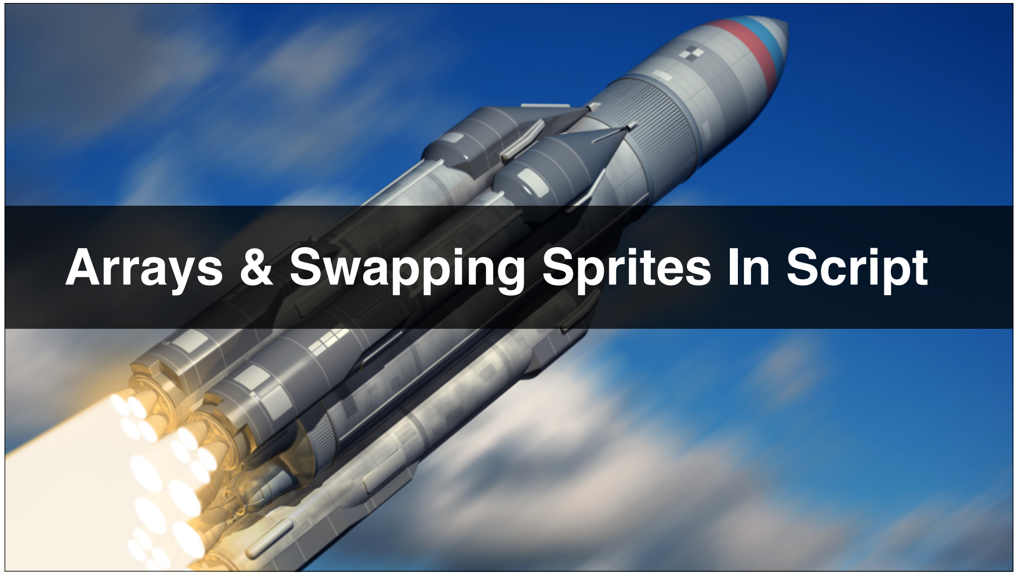
Your image editor needs…

- To support transparent backgrounds.
- To allow you to resize your “canvas”.
- Ideally to support layers.

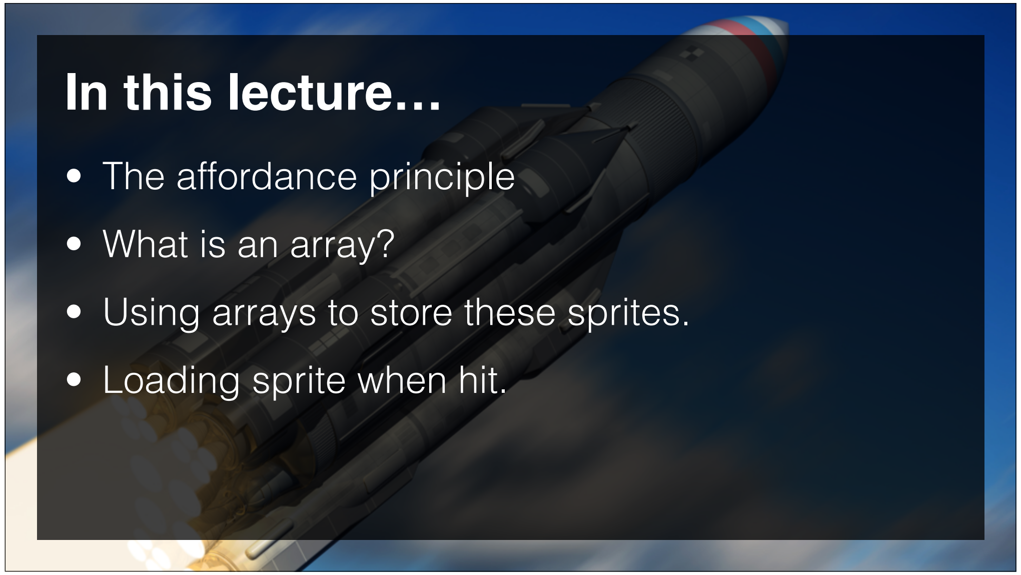


Create damaged bricks

- Add two more sprites
- One slightly damaged (1 hit)
- One very damaged (2 hit)
- Import as **1 hit** and **2 hit**
- Test by applying to prefabs temporarily.

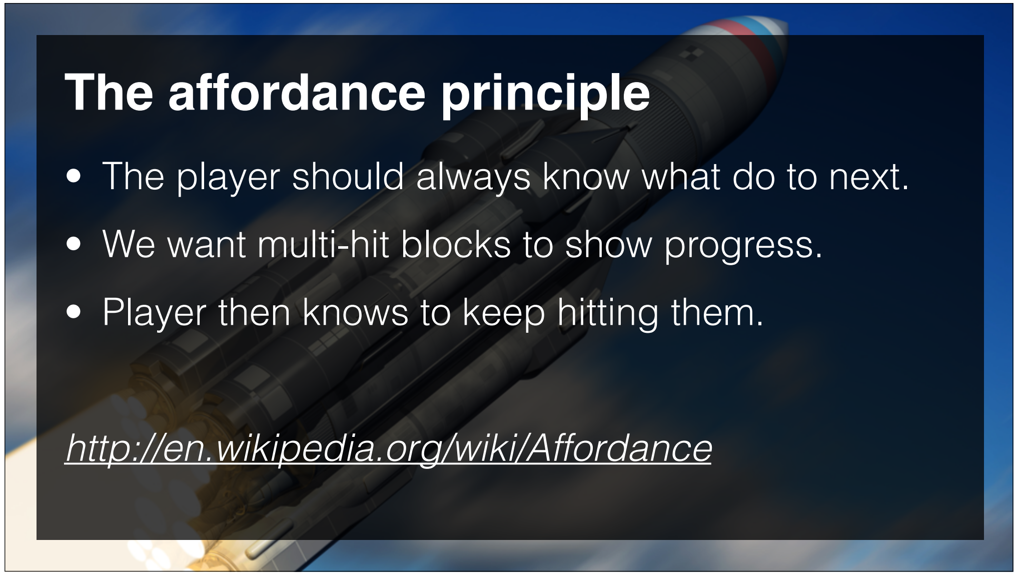


Arrays & Swapping Sprites In Script



In this lecture…

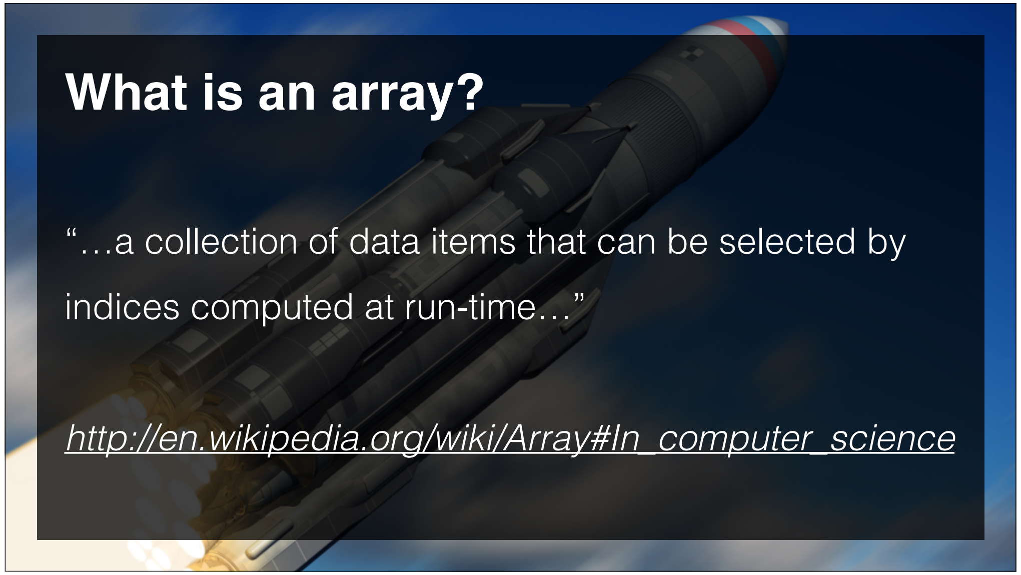
- The affordance principle
- What is an array?
- Using arrays to store these sprites.
- Loading sprite when hit.



The affordance principle

- The player should always know what do to next.
- We want multi-hit blocks to show progress.
- Player then knows to keep hitting them.

http://en.wikipedia.org/wiki/Affordance

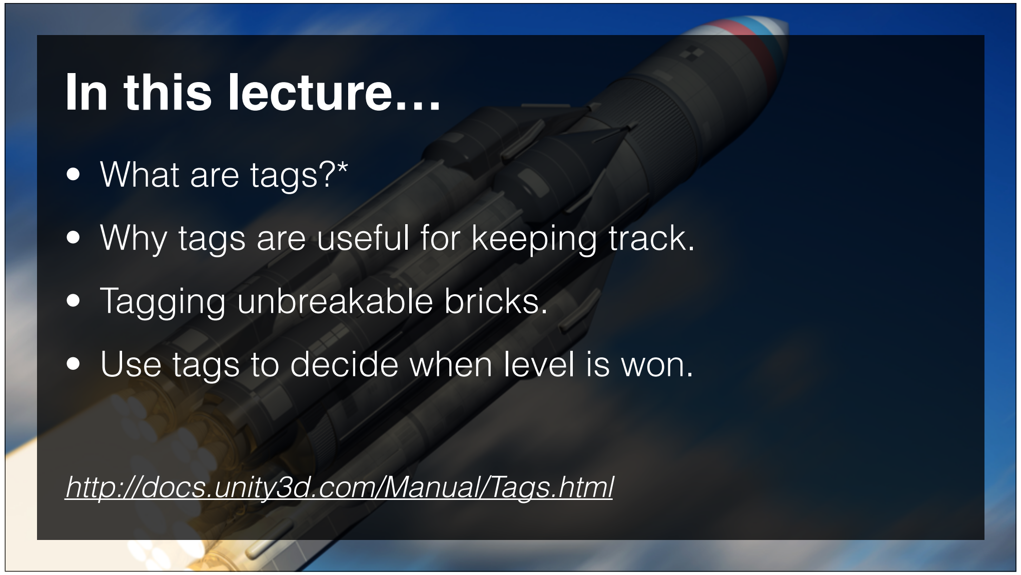


What is an array?

"…a collection of data items that can be selected by indices computed at run-time…"

http://en.wikipedia.org/wiki/Array#In_computer_science

Consolidating Hit Counting

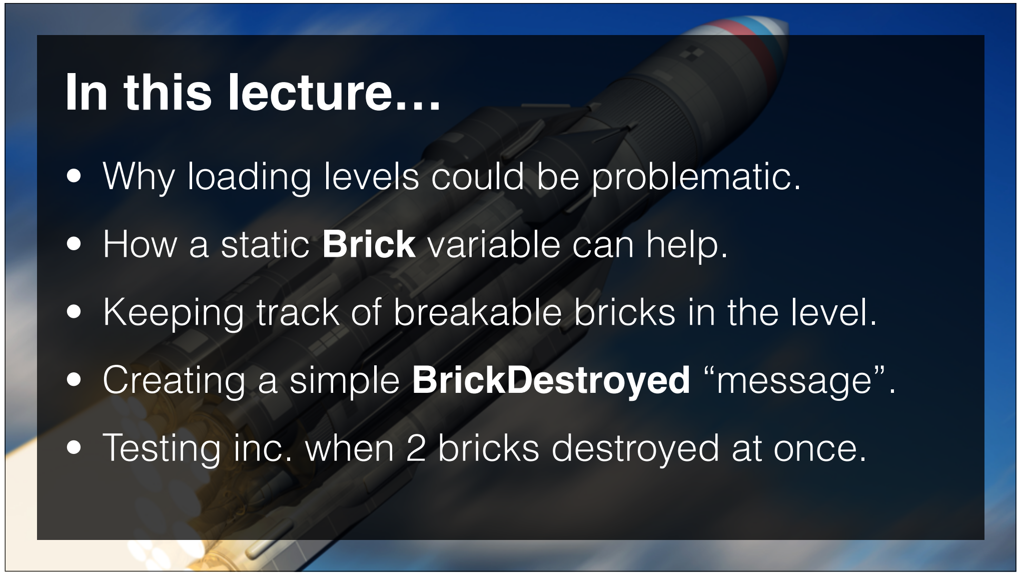


In this lecture…

- What are tags?*
- Why tags are useful for keeping track.
- Tagging unbreakable bricks.
- Use tags to decide when level is won.

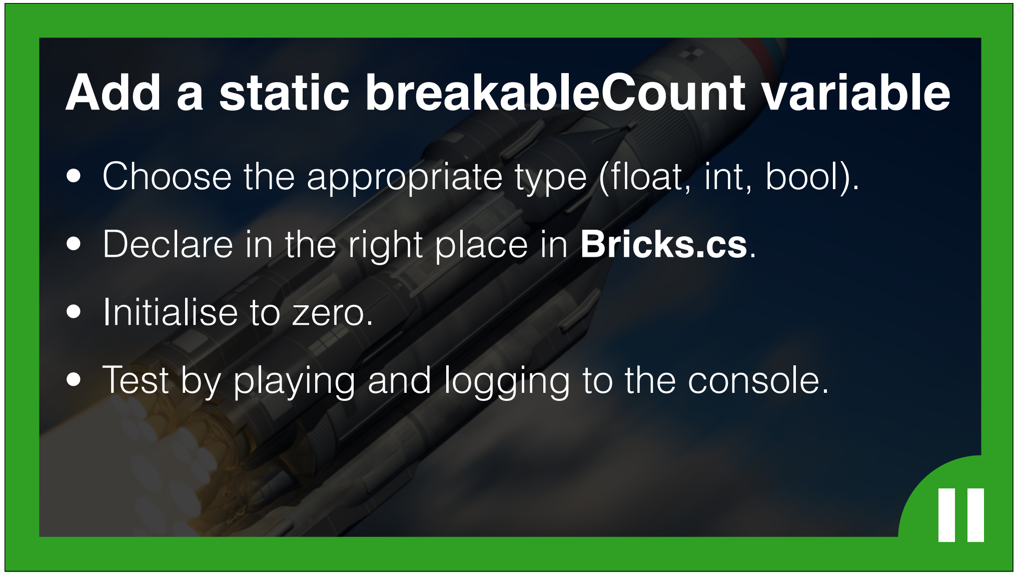
http://docs.unity3d.com/Manual/Tags.html

Statics To Detect Win Condition



In this lecture…

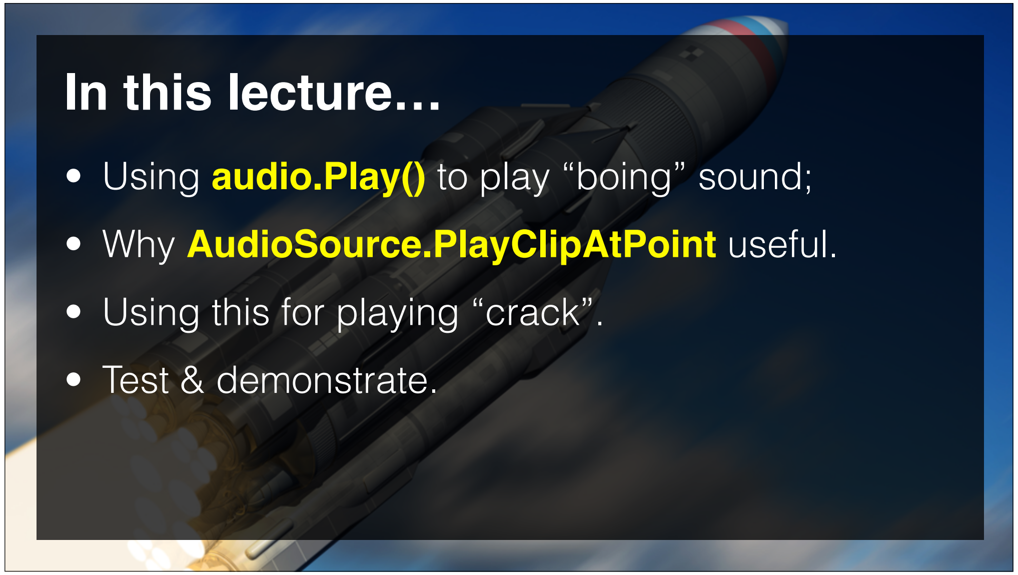
- Why loading levels could be problematic.
- How a static **Brick** variable can help.
- Keeping track of breakable bricks in the level.
- Creating a simple **BrickDestroyed** “message”.
- Testing inc. when 2 bricks destroyed at once.



Add a static breakableCount variable

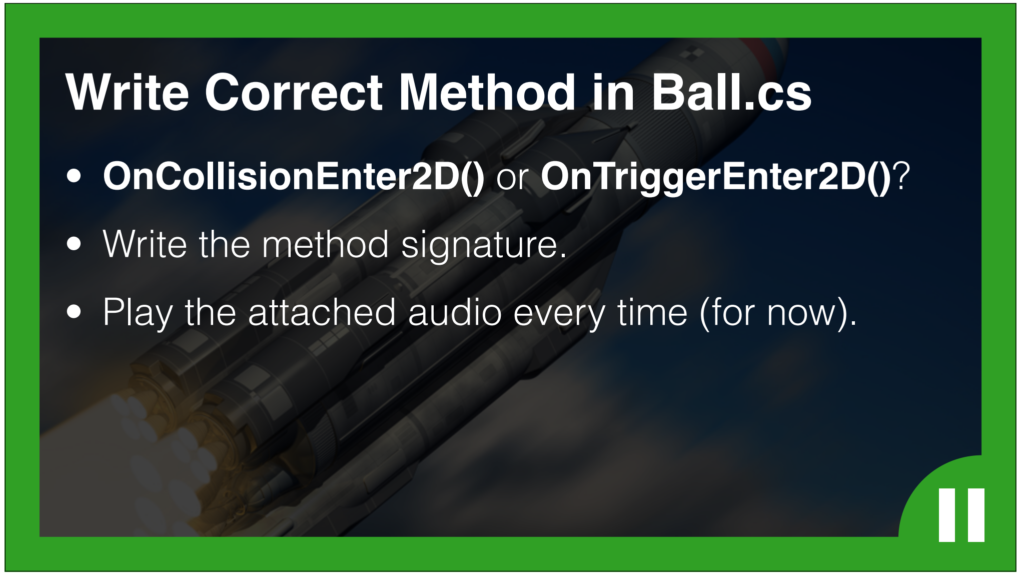
- Choose the appropriate type (float, int, bool).
- Declare in the right place in **Bricks.cs**.
- Initialise to zero.
- Test by playing and logging to the console.

Playing Sound Effects On Impact



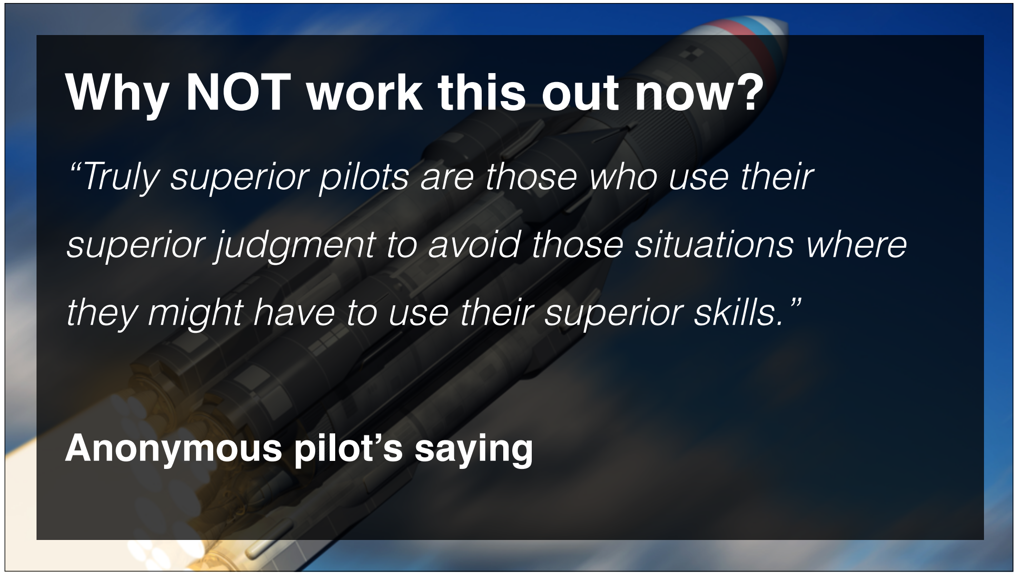
In this lecture…

- Using **audio.Play()** to play “boing” sound;
- Why **AudioSource.PlayClipAtPoint** useful.
- Using this for playing “crack”.
- Test & demonstrate.



Write Correct Method in Ball.cs

- **OnCollisionEnter2D()** or **OnTriggerEnter2D()**?
- Write the method signature.
- Play the attached audio every time (for now).



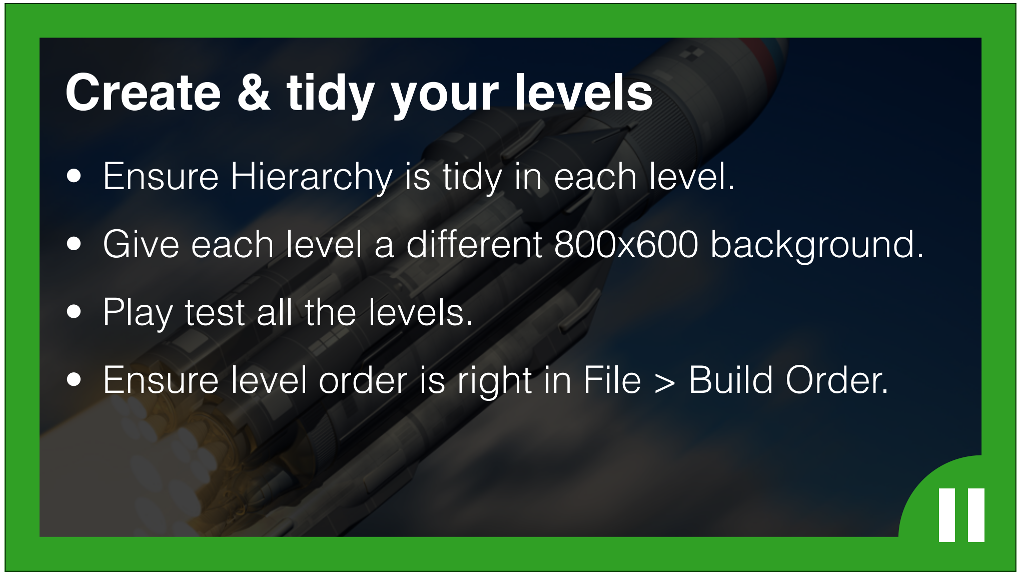
Why NOT work this out now?

“Truly superior pilots are those who use their superior judgment to avoid those situations where they might have to use their superior skills.”

Anonymous pilot’s saying

Stop Boring Loops With Random.Range()

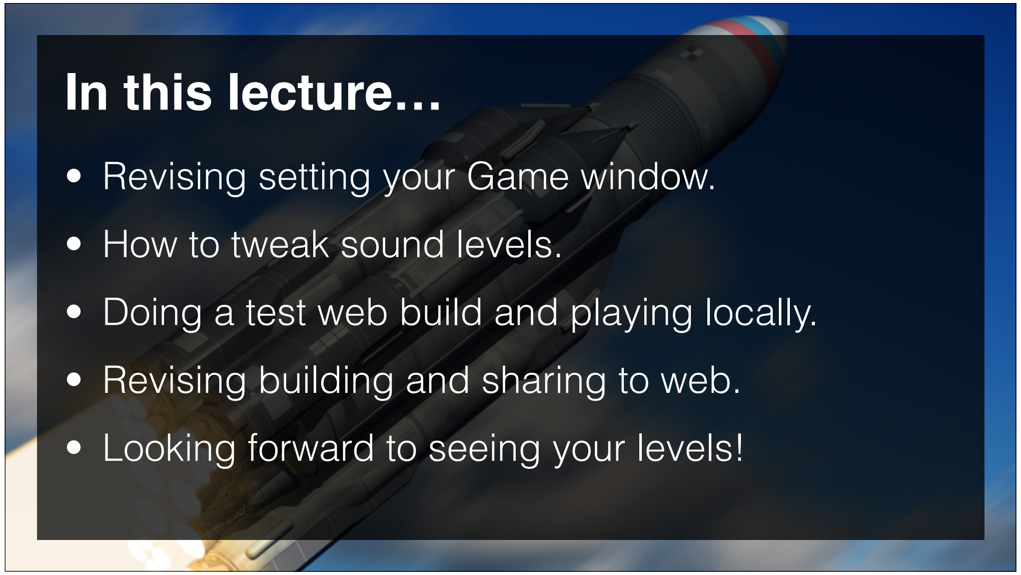
Automated Play Testing



Create & tidy your levels

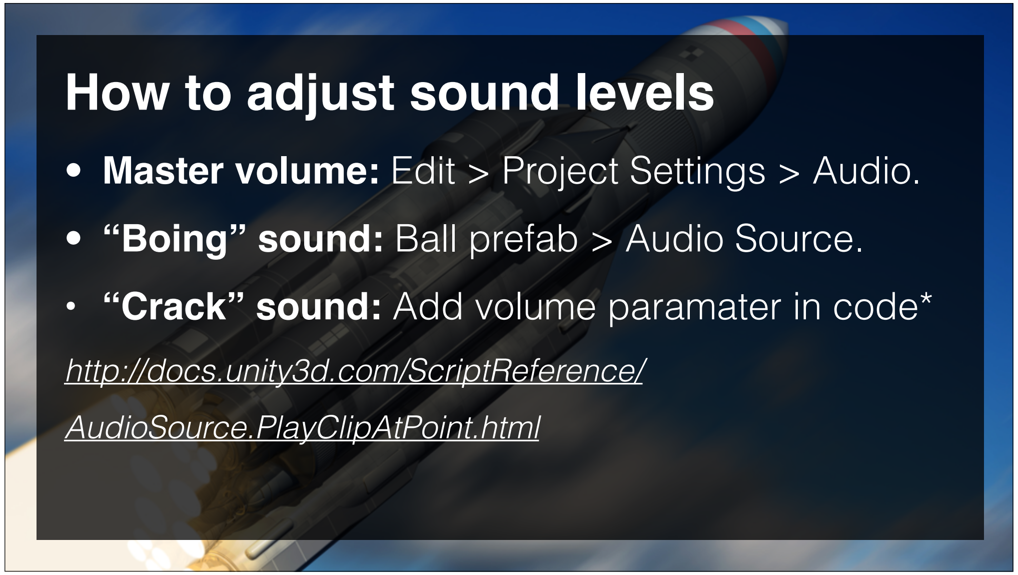
- Ensure Hierarchy is tidy in each level.
- Give each level a different 800x600 background.
- Play test all the levels.
- Ensure level order is right in File > Build Order.

Build & Share On The Web



In this lecture…

- Revising setting your Game window.
- How to tweak sound levels.
- Doing a test web build and playing locally.
- Revising building and sharing to web.
- Looking forward to seeing your levels!

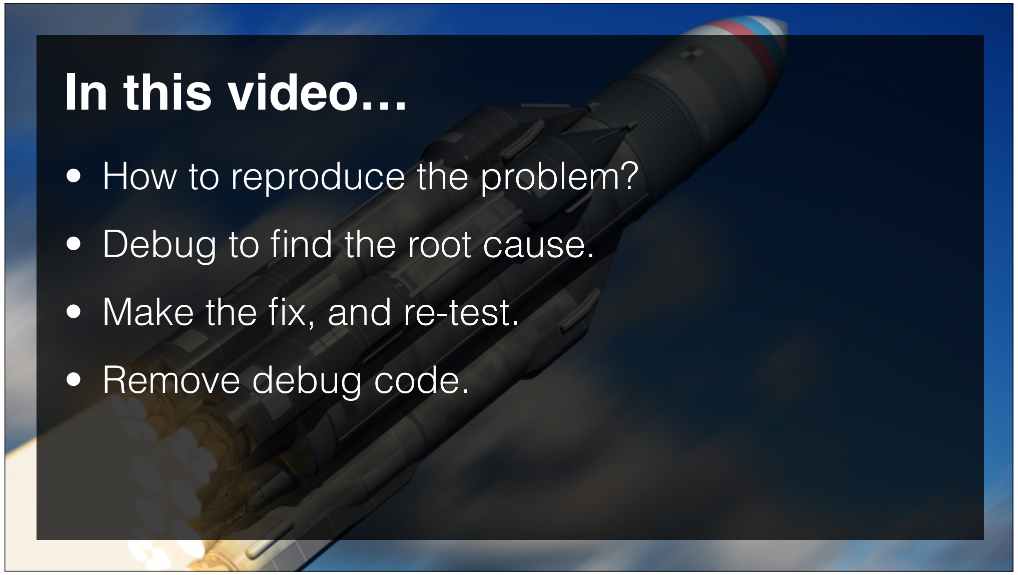


How to adjust sound levels

- **Master volume:** Edit > Project Settings > Audio.
- **"Boing" sound:** Ball prefab > Audio Source.
- **"Crack" sound:** Add volume paramater in code*

http://docs.unity3d.com/ScriptReference/AudioSource.PlayClipAtPoint.html

Fixing User Reported Bugs



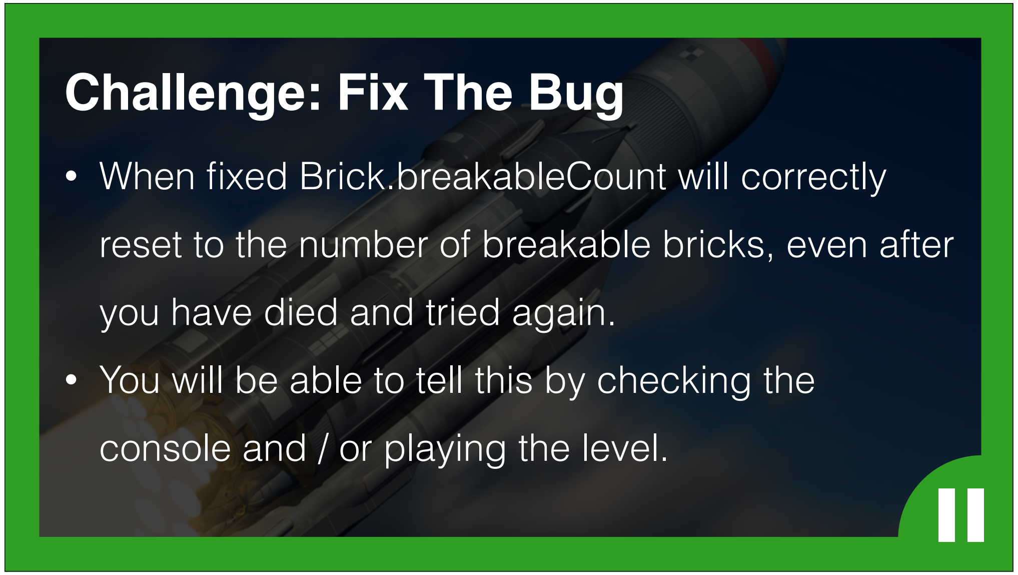
In this video…

- How to reproduce the problem?
- Debug to find the root cause.
- Make the fix, and re-test.
- Remove debug code.

Example Bug Report

"I've noticed an odd behaviour … it works up until I start a new game then it duplicate itself like a loop."

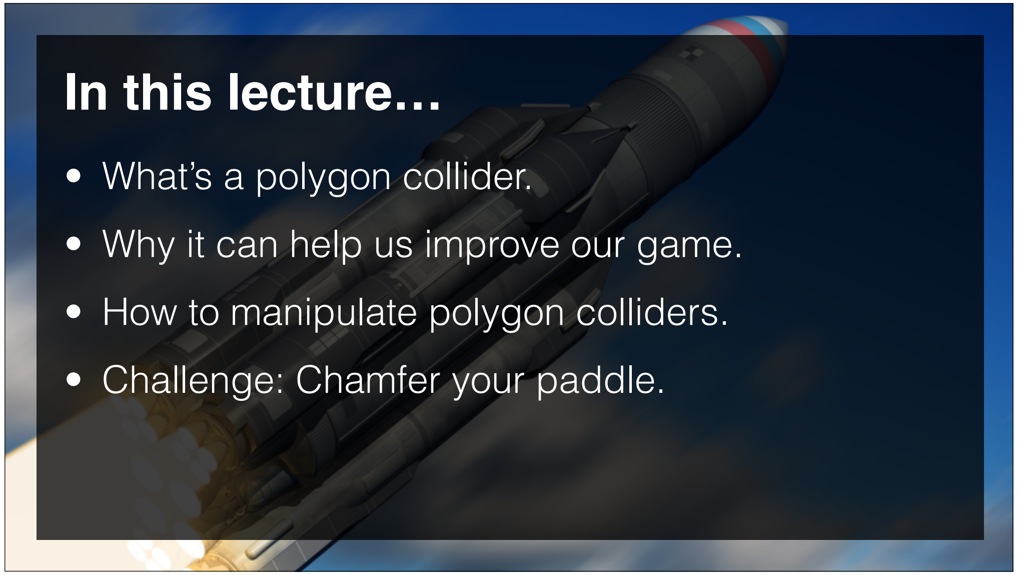
Thanks to Daniel, Nathan and Marko



Challenge: Fix The Bug

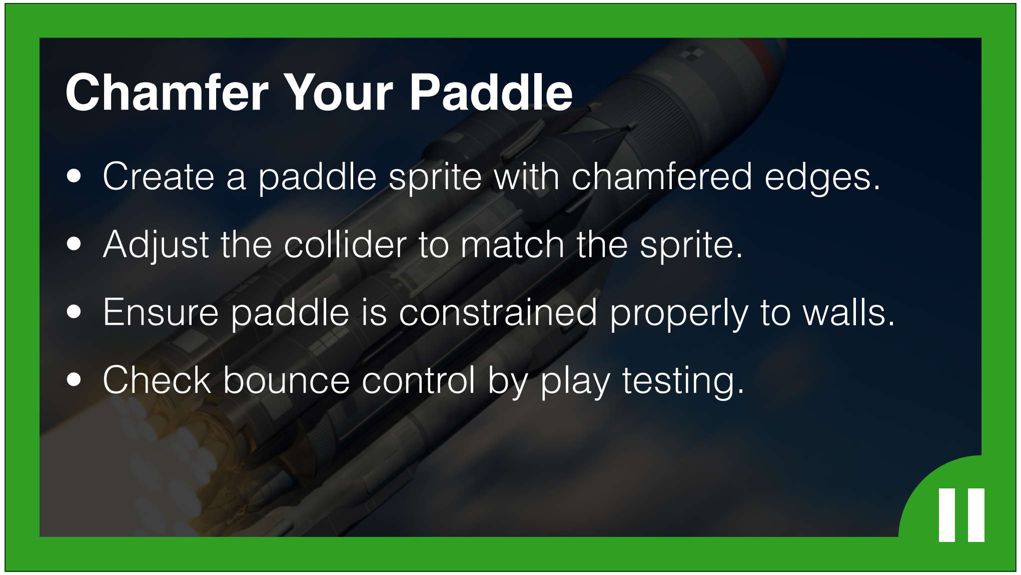
- When fixed Brick.breakableCount will correctly reset to the number of breakable bricks, even after you have died and tried again.
- You will be able to tell this by checking the console and / or playing the level.

More Complex Collider Shapes



In this lecture…

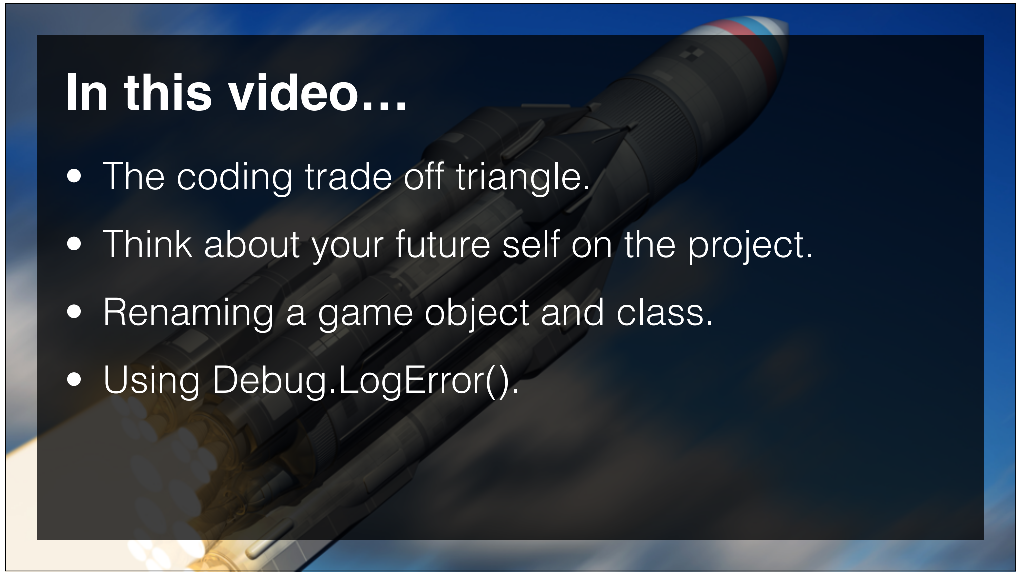
- What's a polygon collider.
- Why it can help us improve our game.
- How to manipulate polygon colliders.
- Challenge: Chamfer your paddle.



Chamfer Your Paddle

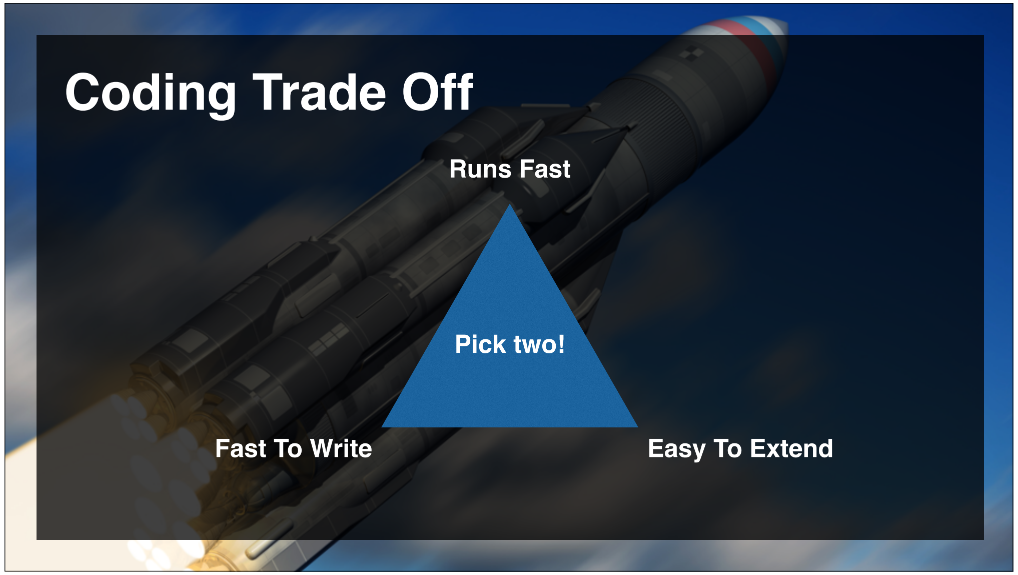
- Create a paddle sprite with chamfered edges.
- Adjust the collider to match the sprite.
- Ensure paddle is constrained properly to walls.
- Check bounce control by play testing.

Making Code Extendable



In this video…

- The coding trade off triangle.
- Think about your future self on the project.
- Renaming a game object and class.
- Using Debug.LogError().



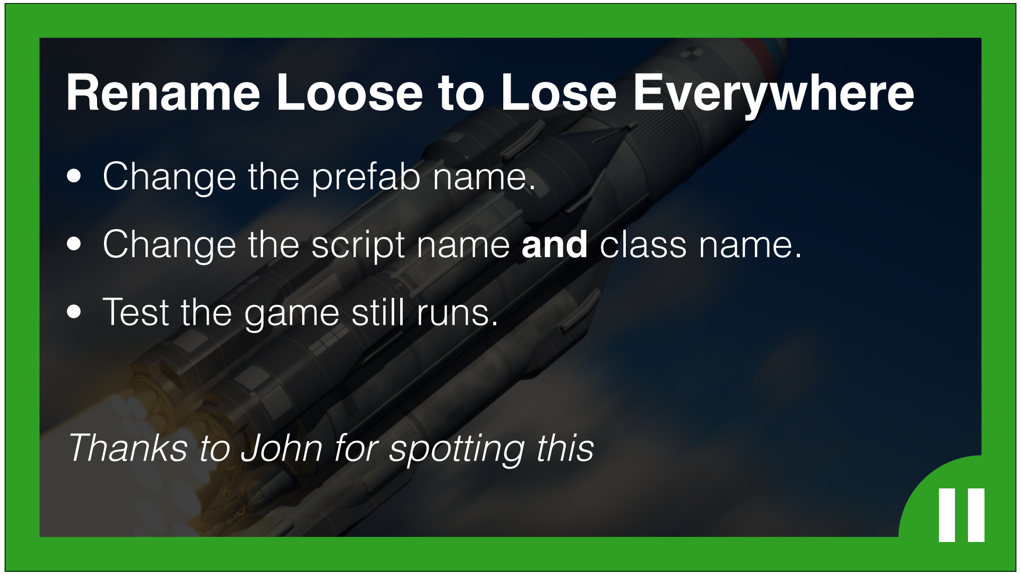
Coding Trade Off

Runs Fast

Pick two!

Fast To Write

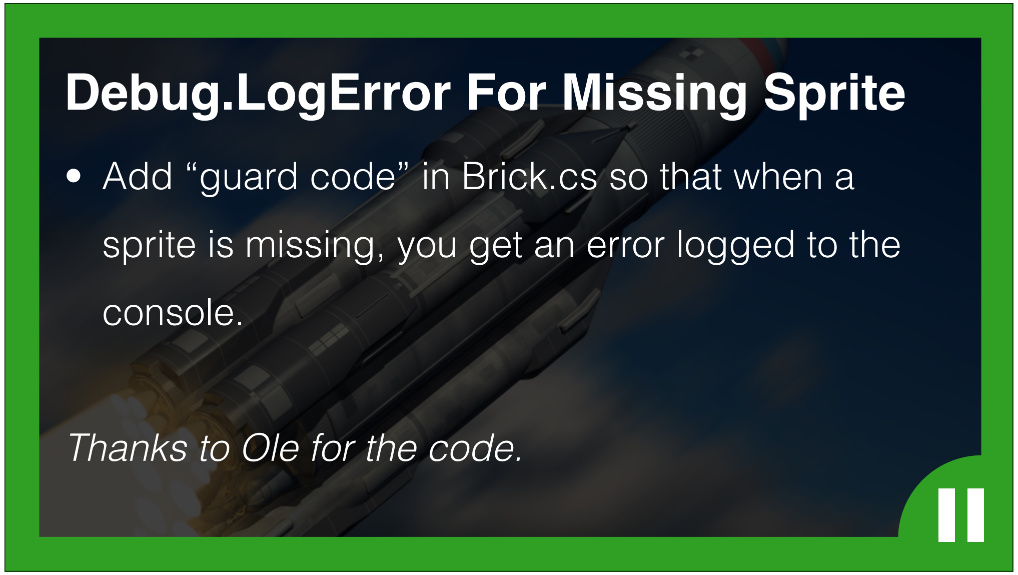
Easy To Extend



Rename Loose to Lose Everywhere

- Change the prefab name.
- Change the script name **and** class name.
- Test the game still runs.

Thanks to John for spotting this

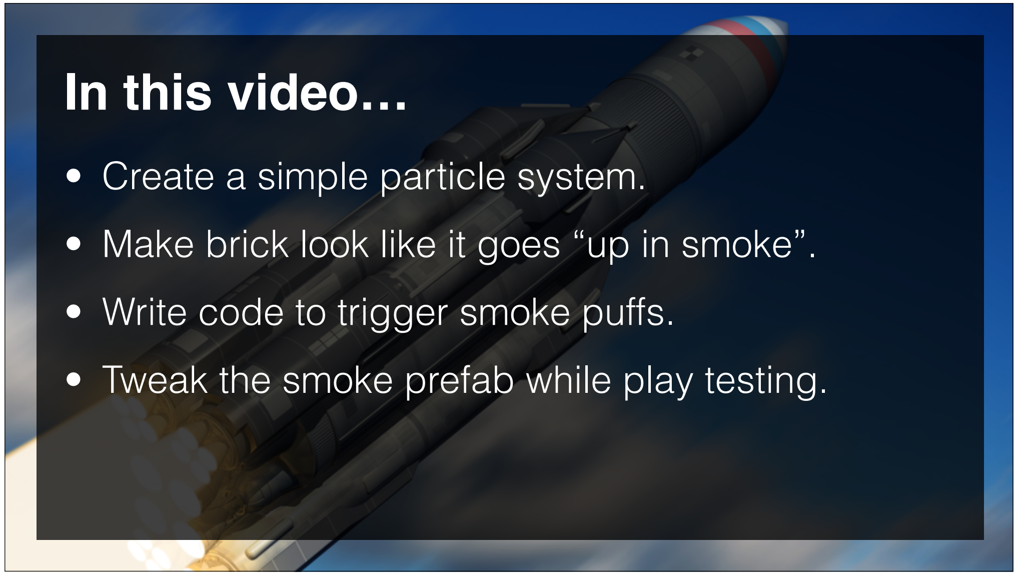


Debug.LogError For Missing Sprite

- Add “guard code” in Brick.cs so that when a sprite is missing, you get an error logged to the console.

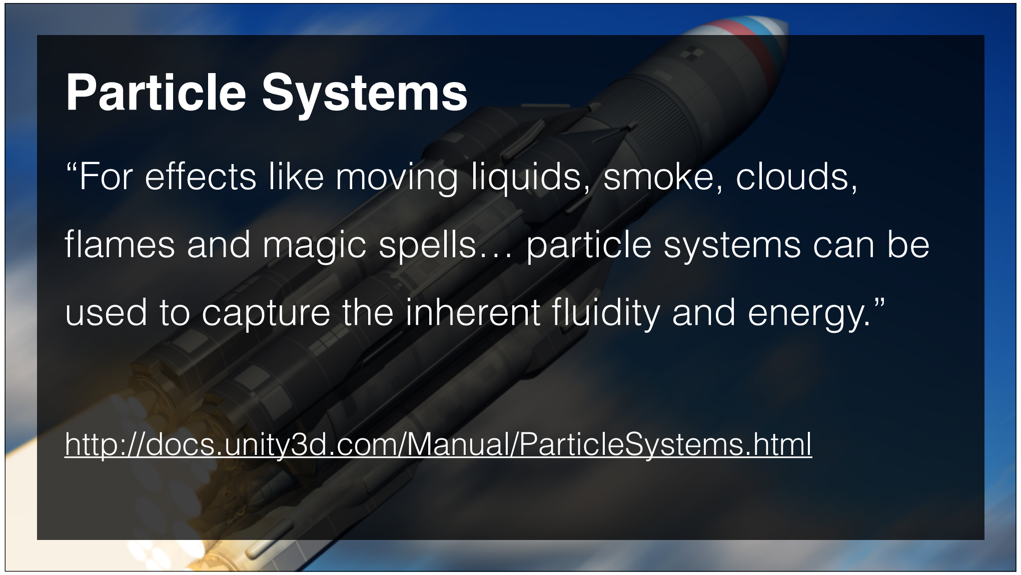
Thanks to Ole for the code.

An Introduction To Particle Systems



In this video…

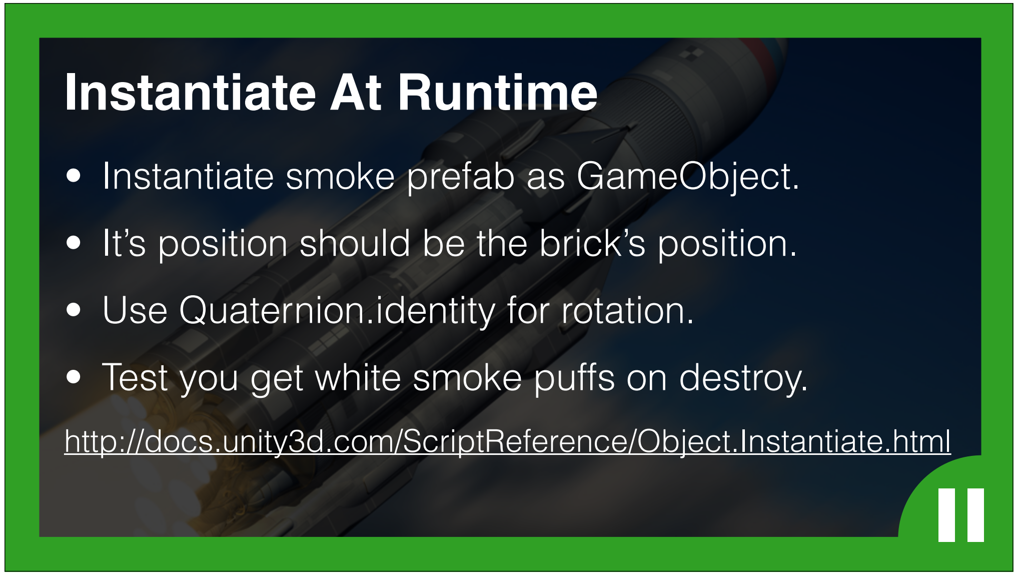
- Create a simple particle system.
- Make brick look like it goes “up in smoke”.
- Write code to trigger smoke puffs.
- Tweak the smoke prefab while play testing.



Particle Systems

“For effects like moving liquids, smoke, clouds, flames and magic spells… particle systems can be used to capture the inherent fluidity and energy.”

http://docs.unity3d.com/Manual/ParticleSystems.html



Instantiate At Runtime

- Instantiate smoke prefab as GameObject.
- It’s position should be the brick’s position.
- Use Quaternion.identity for rotation.
- Test you get white smoke puffs on destroy.

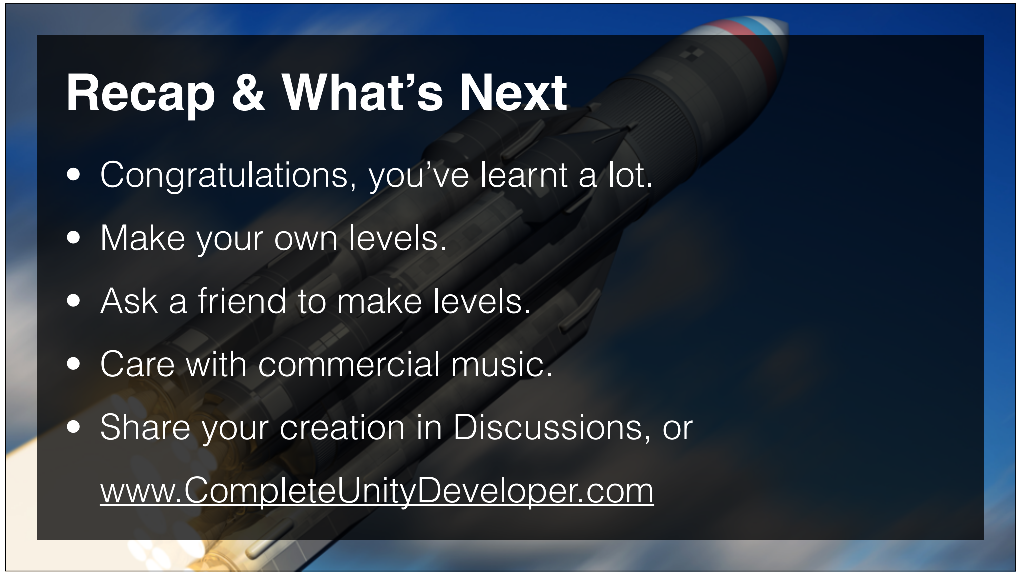
http://docs.unity3d.com/ScriptReference/Object.Instantiate.html

Match Smoke Color To Brick

- Store the puff as a GameObject variable.
- Set smoke’s startColor to the same as brick.
- Test that different bricks make different smoke.

Hint: The color is from the brick’s SpriteRenderer.

Recap & What’s next



Recap & What's Next

- Congratulations, you've learnt a lot.
- Make your own levels.
- Ask a friend to make levels.
- Care with commercial music.
- Share your creation in Discussions, or www.CompleteUnityDeveloper.com