GameDev.tv

Unity 2020.1

Section Intro - Project Boost

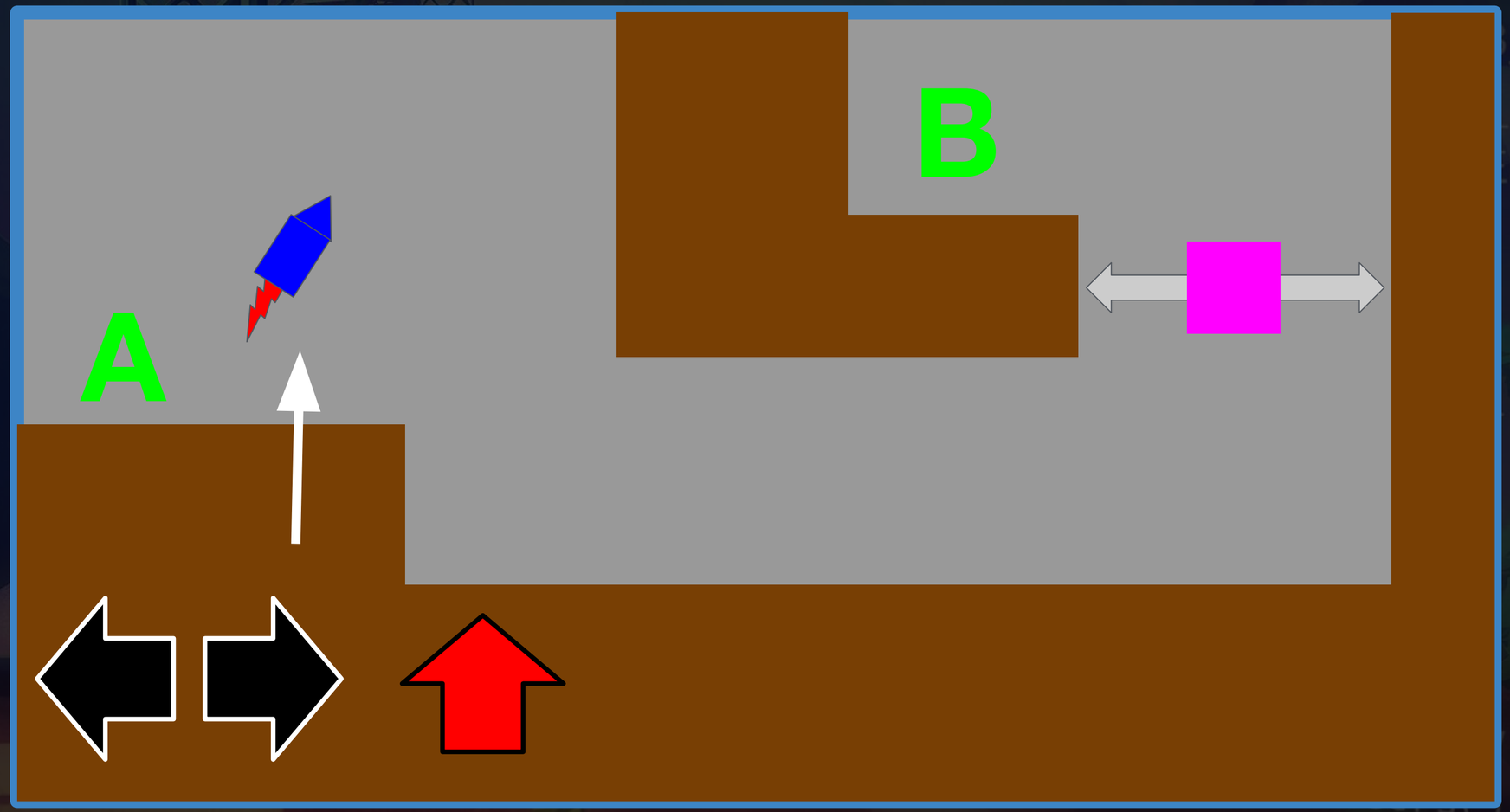
GameDev.tv

Unity 2020.1

Game Design - Project Boost

Rick Davidson

Game Screen



Project Boost Game Design

Player Experience:

Precision. Skillful.

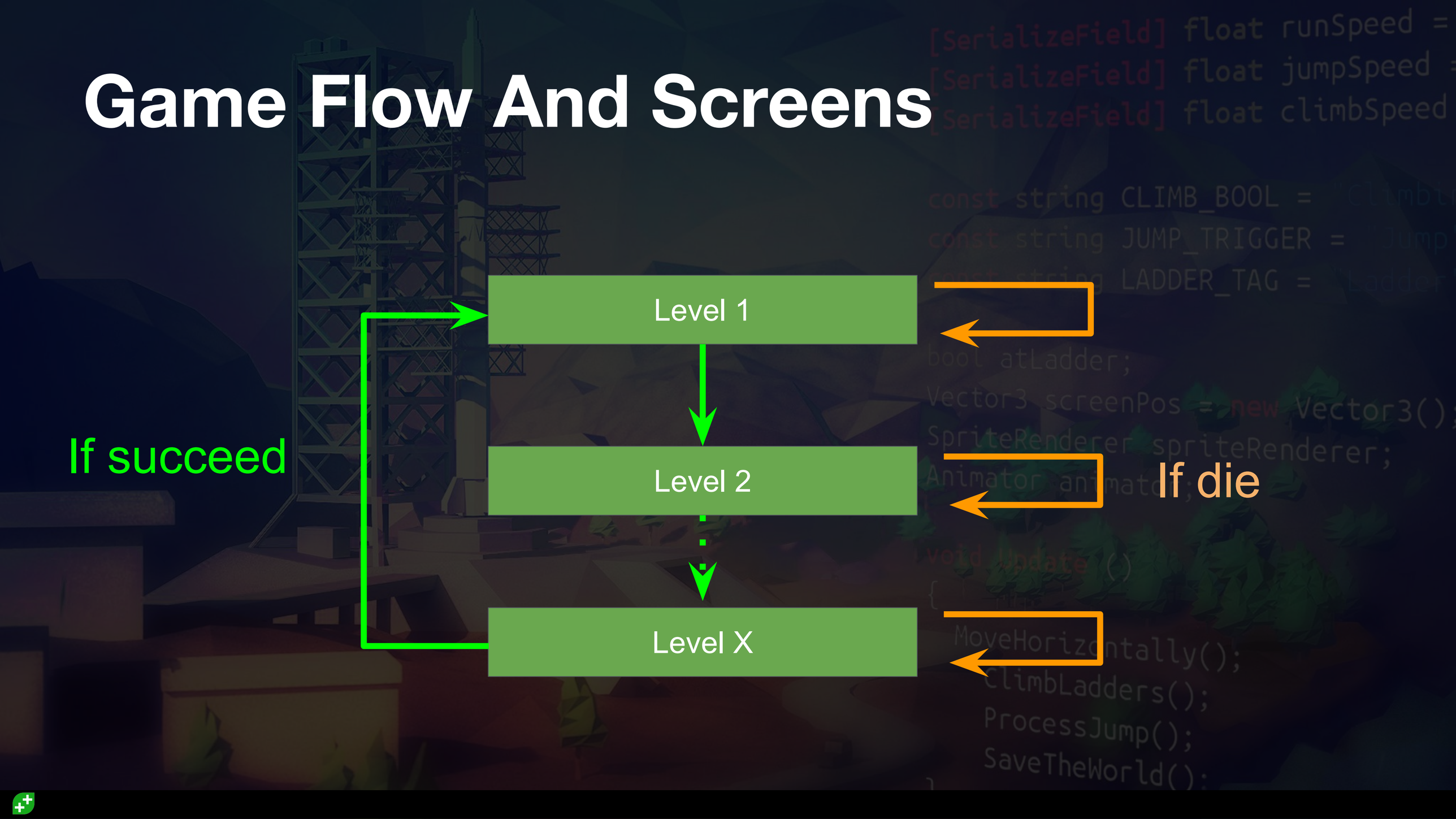
Core Mechanic:

Skillfully fly spaceship and avoid environmental hazards.

Core game loop:

Get from A to B to complete the level, then progress to the next level.

Game Flow And Screens

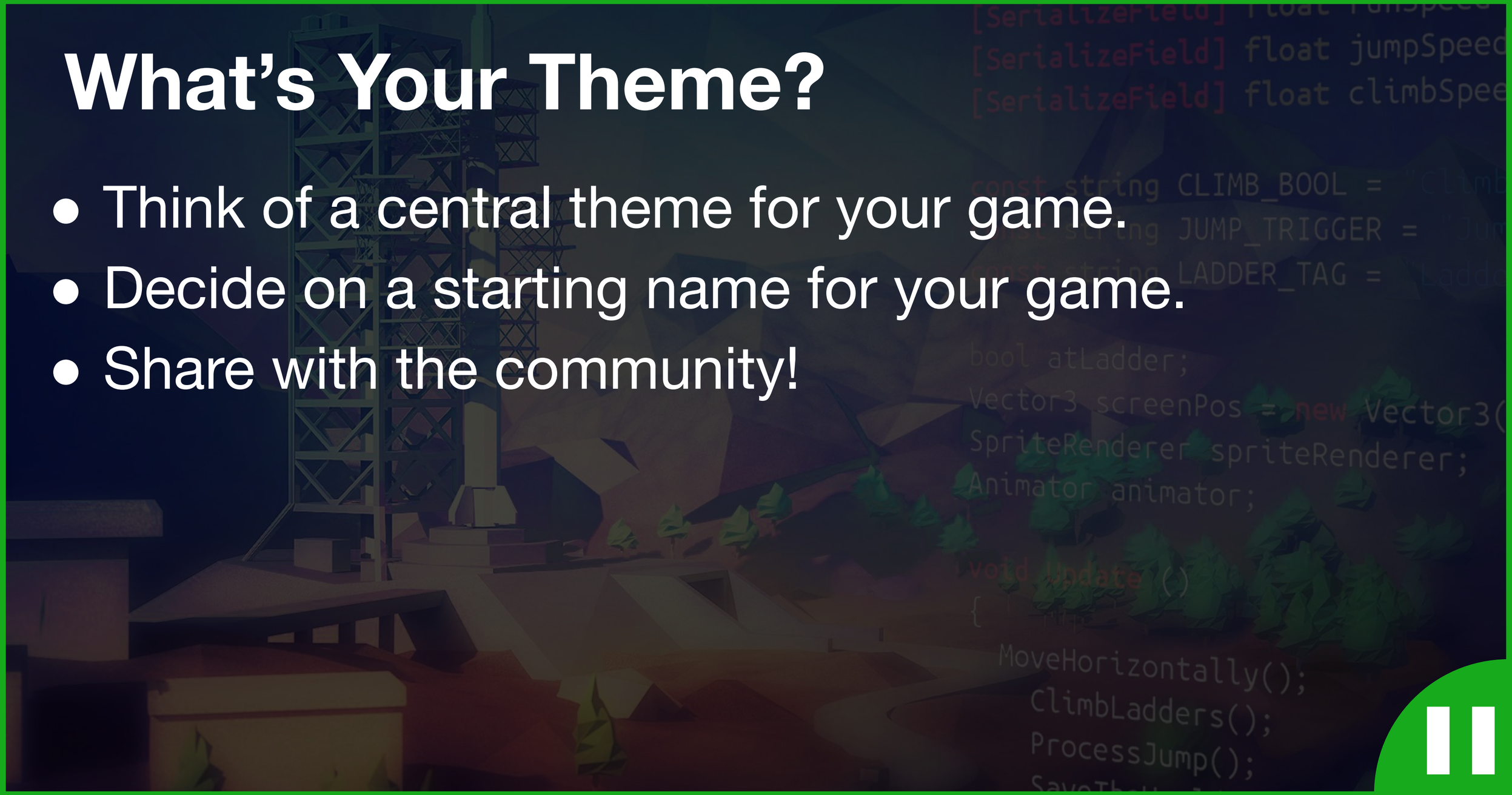


Game Theme (ie. Story & Visuals)

- Experimental early generation spacecraft.
- On an unknown planet, trying to escape.

What's Your Theme?

- Think of a central theme for your game.
- Decide on a starting name for your game.
- Share with the community!



GameDev.tv

Unity 2020.1

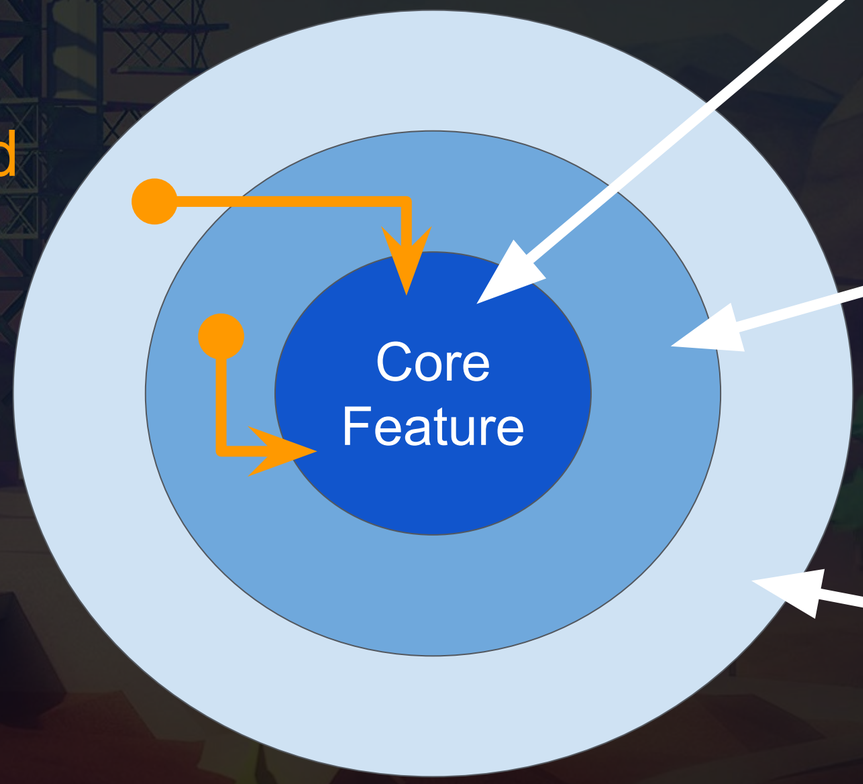
Onion Design

Common Development Questions

- What features should I include in my game?
- Where should I start development?
- What are my priorities?
- What if I run out of time?
- When should I stop?

Onion Design

All features need to feed the core and make it better

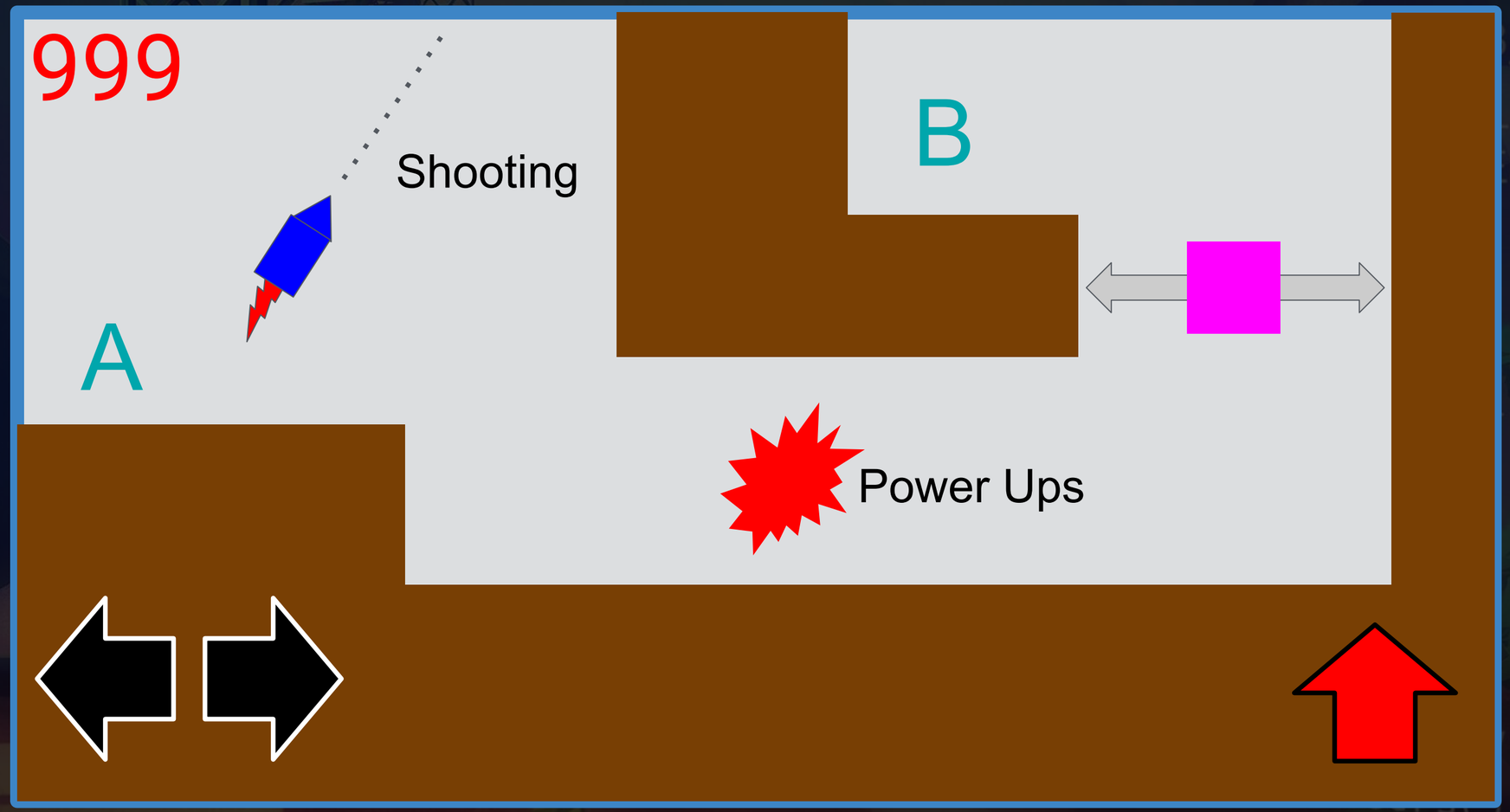


Most important feature

2nd most important feature

3rd most important feature

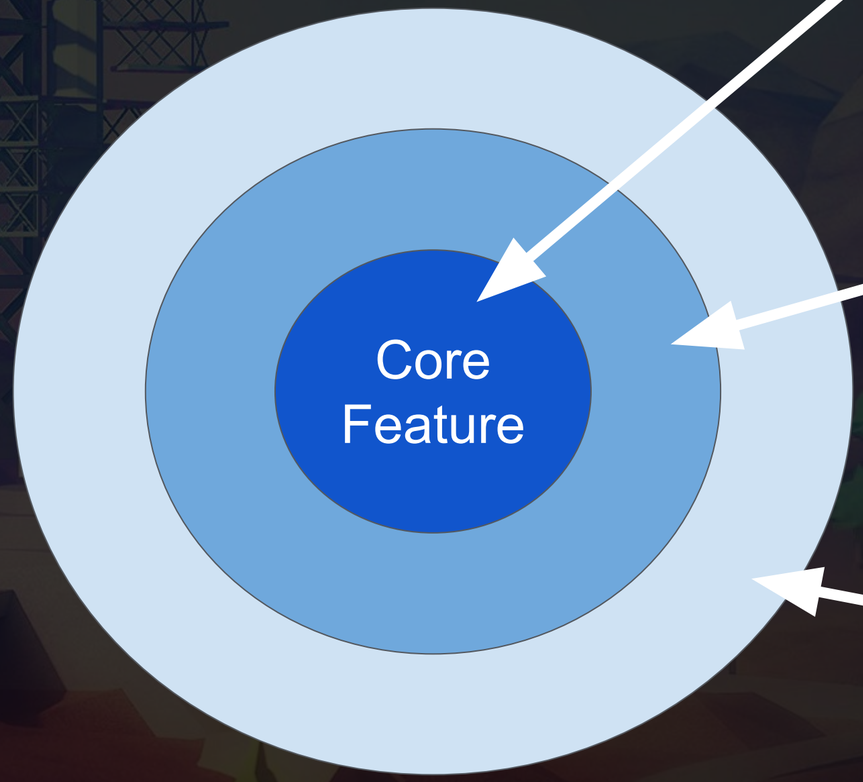
Game Screen


Fuel
999
Shooting
B
A
Moving obstacles
Power Ups
Left
Right
Boost

Sketch Our Onion Design

- What do you believe is the single most important feature of our game?
- What is next most important?
- What is next most important?

Onion Design For Project Boost



1. Movement / flying
2. World collision
3. Level progression

GameDev.tv

Unity 2020.1

Unity Units

Building Our Starting Pieces

Which Axis Is Which...

+x = right

+y = up

+z = forward

Build Our Starting Pieces

- Create the items from just 1 primitive each (later we'll make them more complex if need be):
  - Ground
  - Launch pad
  - Landing pad
  - Rocket

GameDev.tv

Unity 2020.1

Introducing Classes

How C# Is Broadly Organised

```
namespace UnityEngine
    class OurClass
        SomeMethod()
            statementA;
            statementB;
```

What Are Classes?

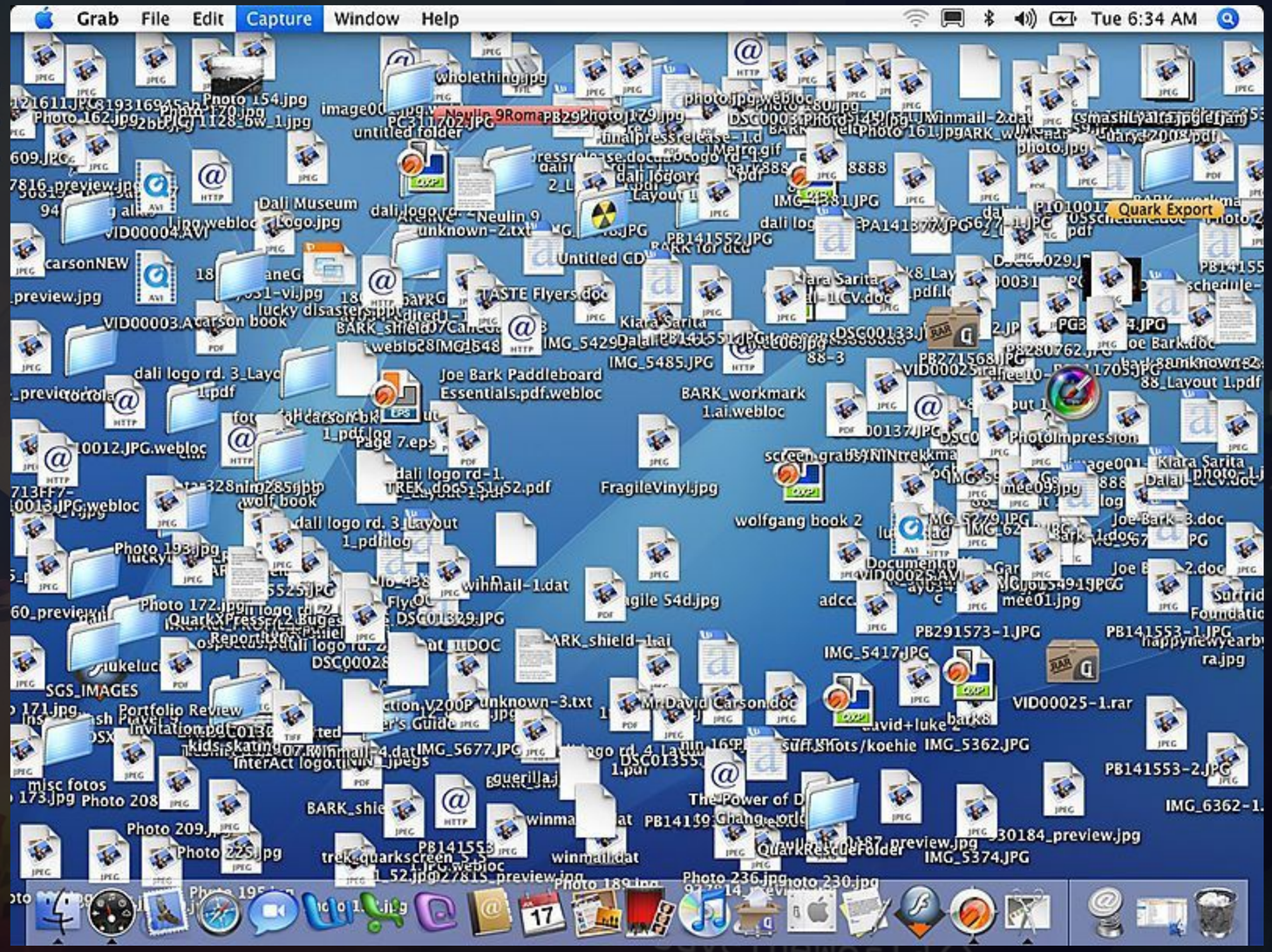
- Classes are used to organise our code
- Classes are “containers” for variables and methods that allow us to group similar things together
- Usually we aim for a class to do one main thing and not multiple things
  - Easier to read our code
  - Easier to fix issues
  - Easier to have multiple people work on a project

Classes We Create

- We tend to create a new class each time we create a new script and have that script responsible for one thing. Eg:
  - Movement
  - CollisionHandler
  - Shooting
  - Score
  - EnemyAI

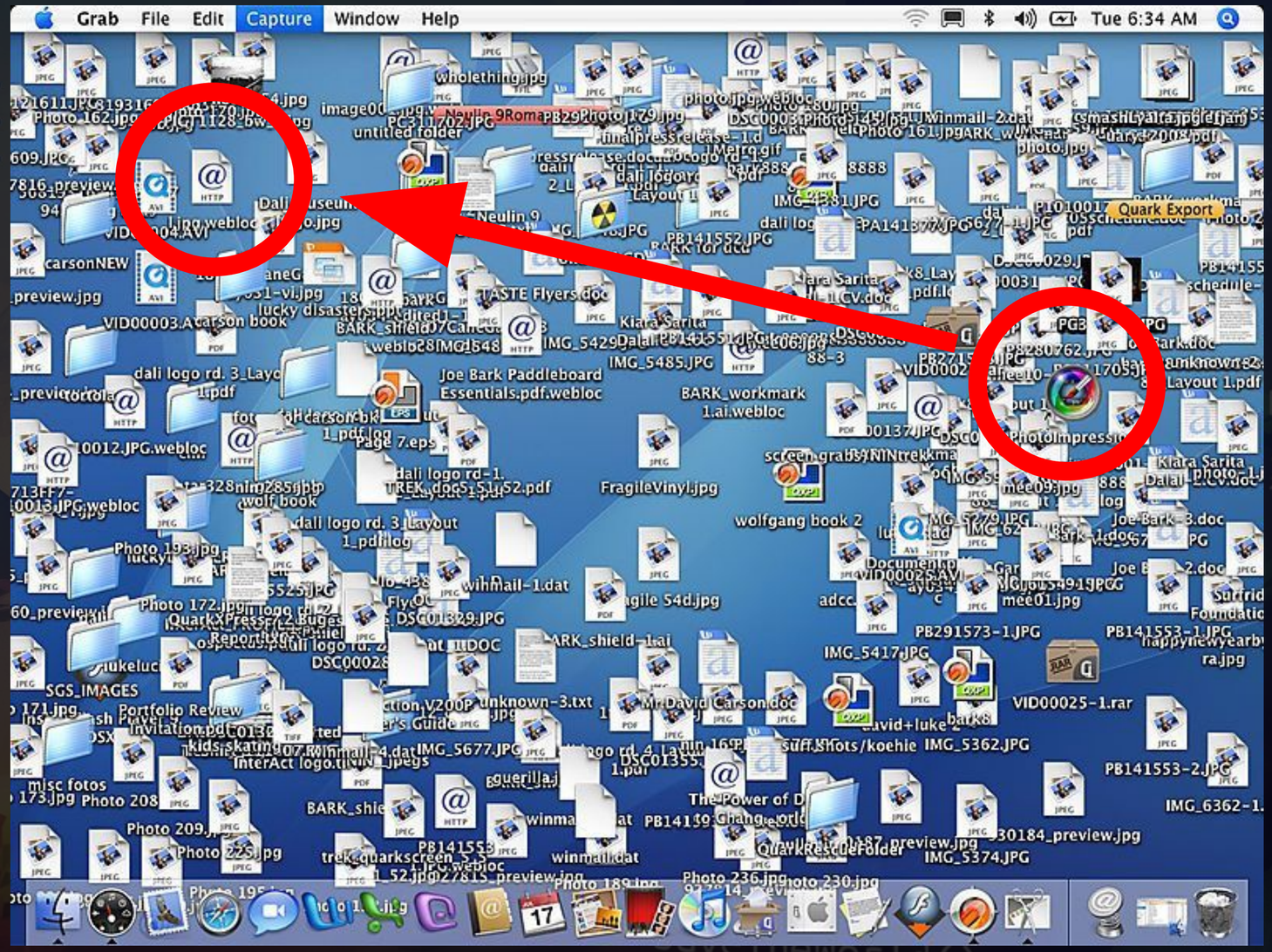
Too Much In One Class Is A Mess

- Cluttered code is like a cluttered desktop



Too Much In One Class Is A Mess

- Also risky because everything can access everything else



Encapsulating Our Code

- Where possible we Encapsulate our code:
  - En-capsule-ating - putting in a capsule.
- Think of the different parts of your code as having a “need to know basis” level of access.
  - Ie. Don’t let everything access everything else.
- For example, only the Methods in our *Movement* Class can alter our player’s movement speed.

Classes Already Built For Us

- Unity has many Classes (containing useful methods and variables) already created for us.
  - By accessing the class, we access its content.
- In our code we write the class name then use the dot operator to access things in the class:

```
ClassName.MethodName()
```

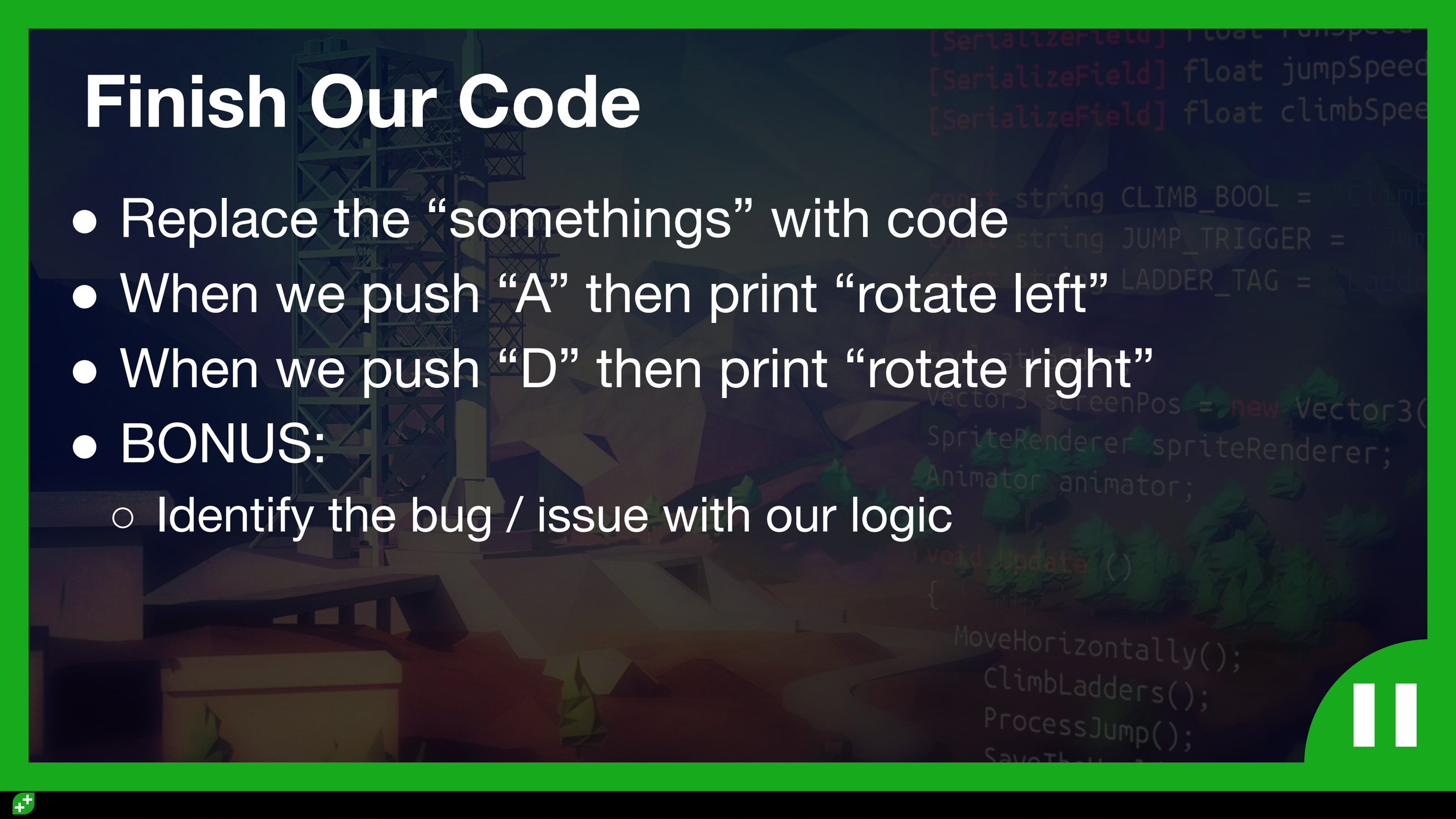
GameDev.tv

Unity 2020.1

Basic Input Binding

Finish Our Code

- Replace the “somethings” with code
- When we push “A” then print “rotate left”
- When we push “D” then print “rotate right”
- BONUS:
  - Identify the bug / issue with our logic



GameDev.tv

Unity 2020.1

Using AddRelativeForce()

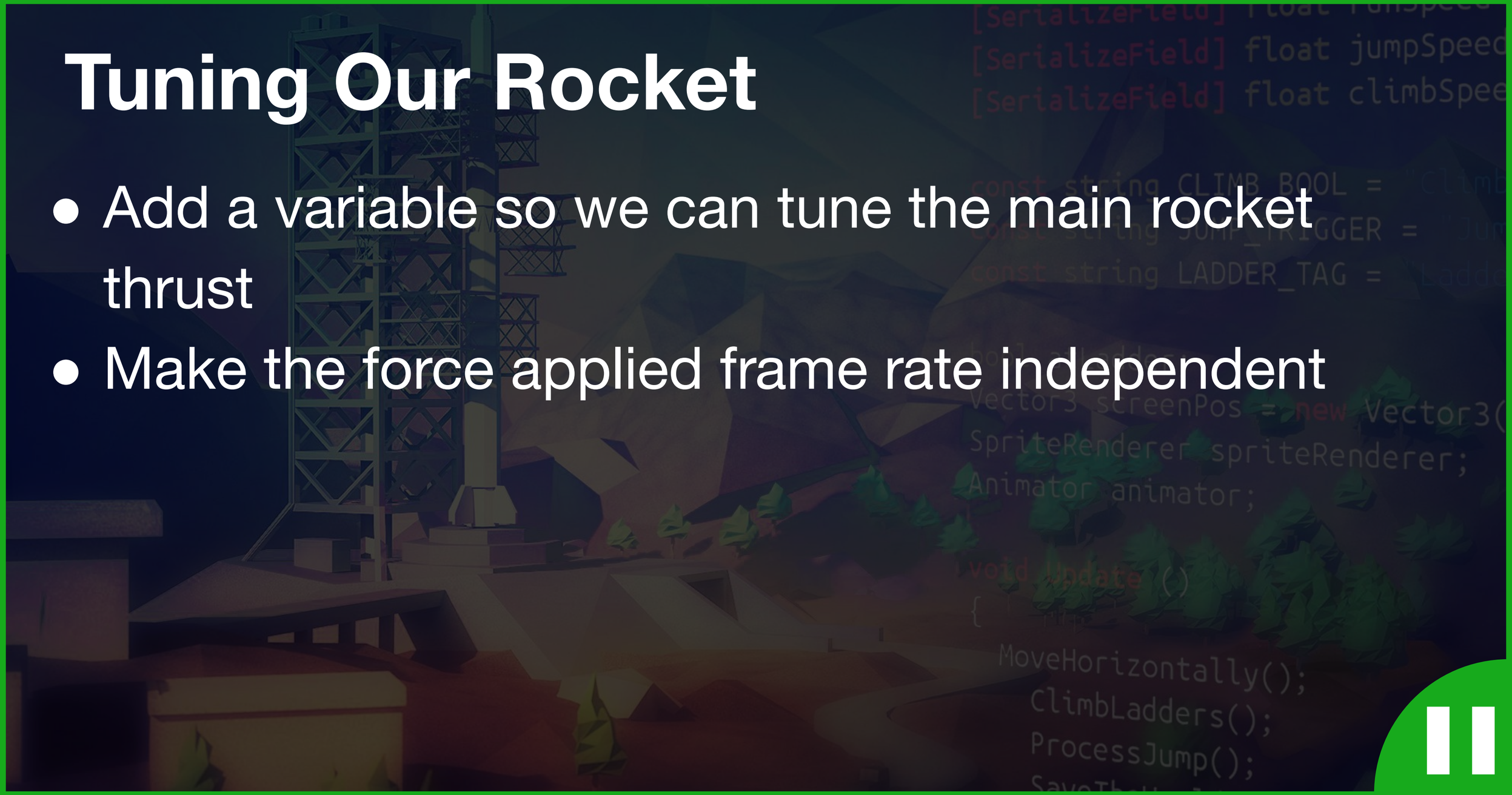
GameDev.tv

Unity 2020.1

Variable For Thrusting

Tuning Our Rocket

- Add a variable so we can tune the main rocket thrust
- Make the force applied frame rate independent



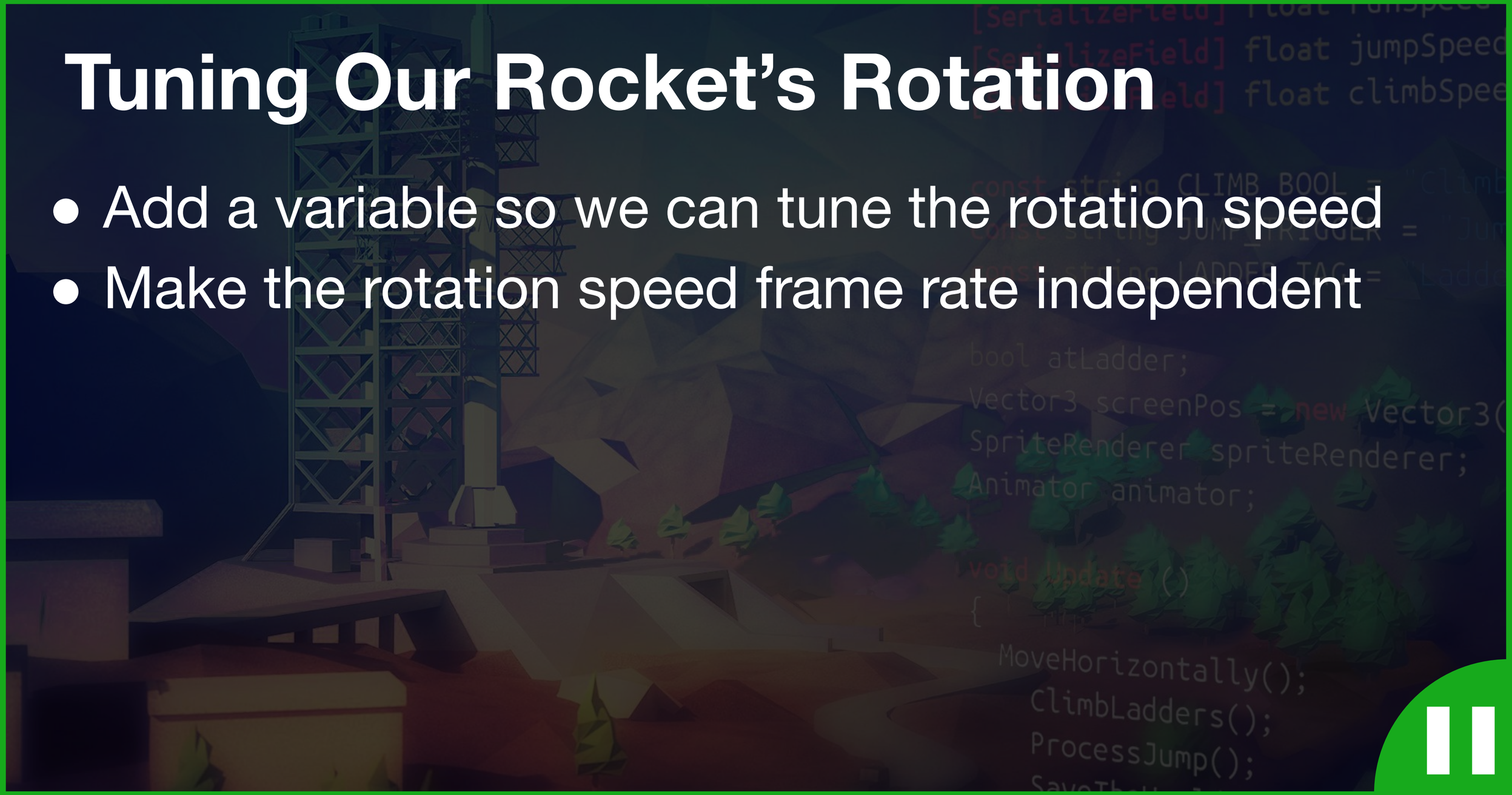
GameDev.tv

Unity 2020.1

Transform.Rotate() Our Rocket

Tuning Our Rocket's Rotation

- Add a variable so we can tune the rotation speed
- Make the rotation speed frame rate independent



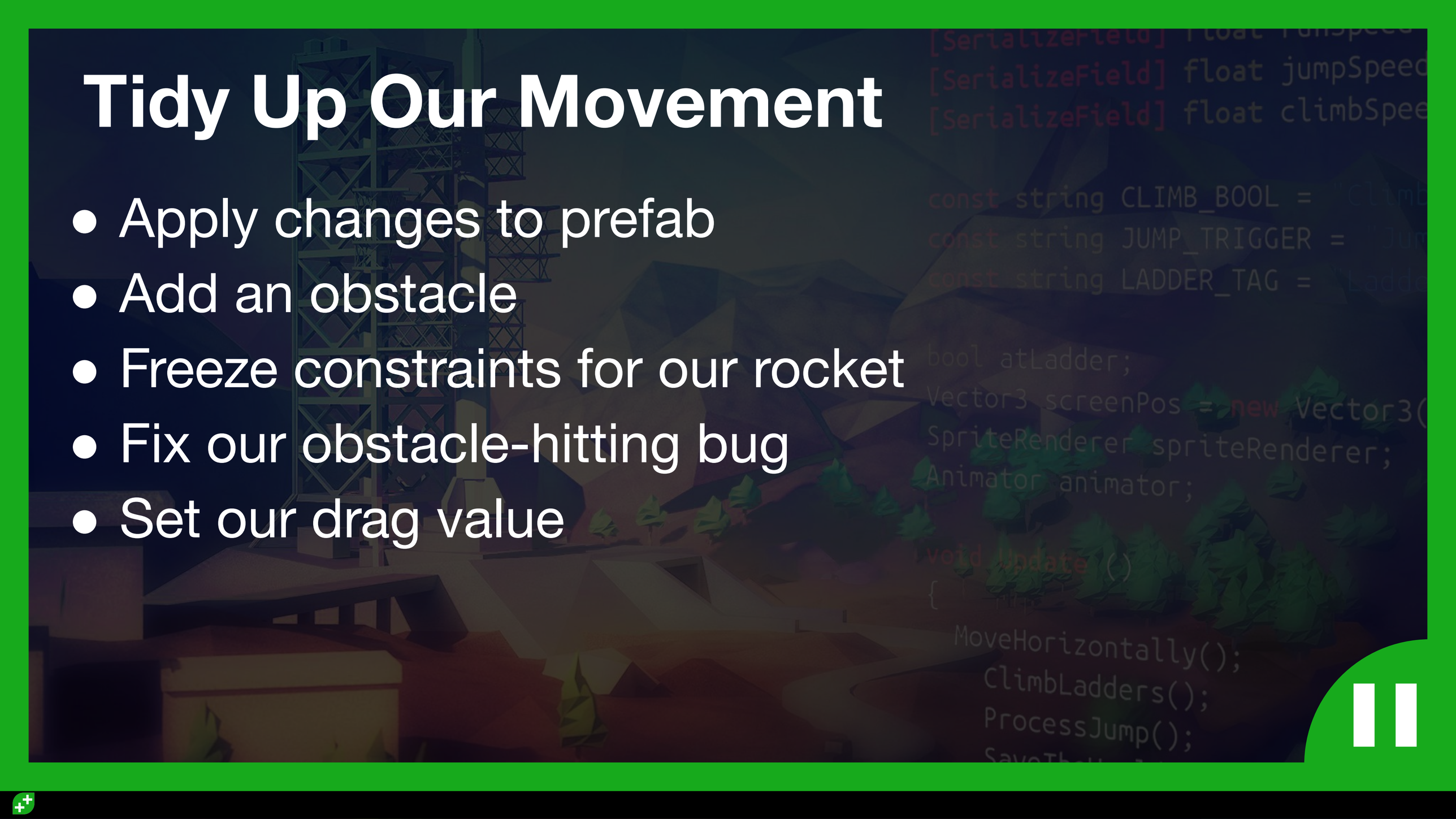
GameDev.tv

Unity 2020.1

Rigidbody Constraints

Tidy Up Our Movement

- Apply changes to prefab
- Add an obstacle
- Freeze constraints for our rocket
- Fix our obstacle-hitting bug
- Set our drag value



GameDev.tv

Unity 2020.1

Our Source Control Repo

Some Terminology

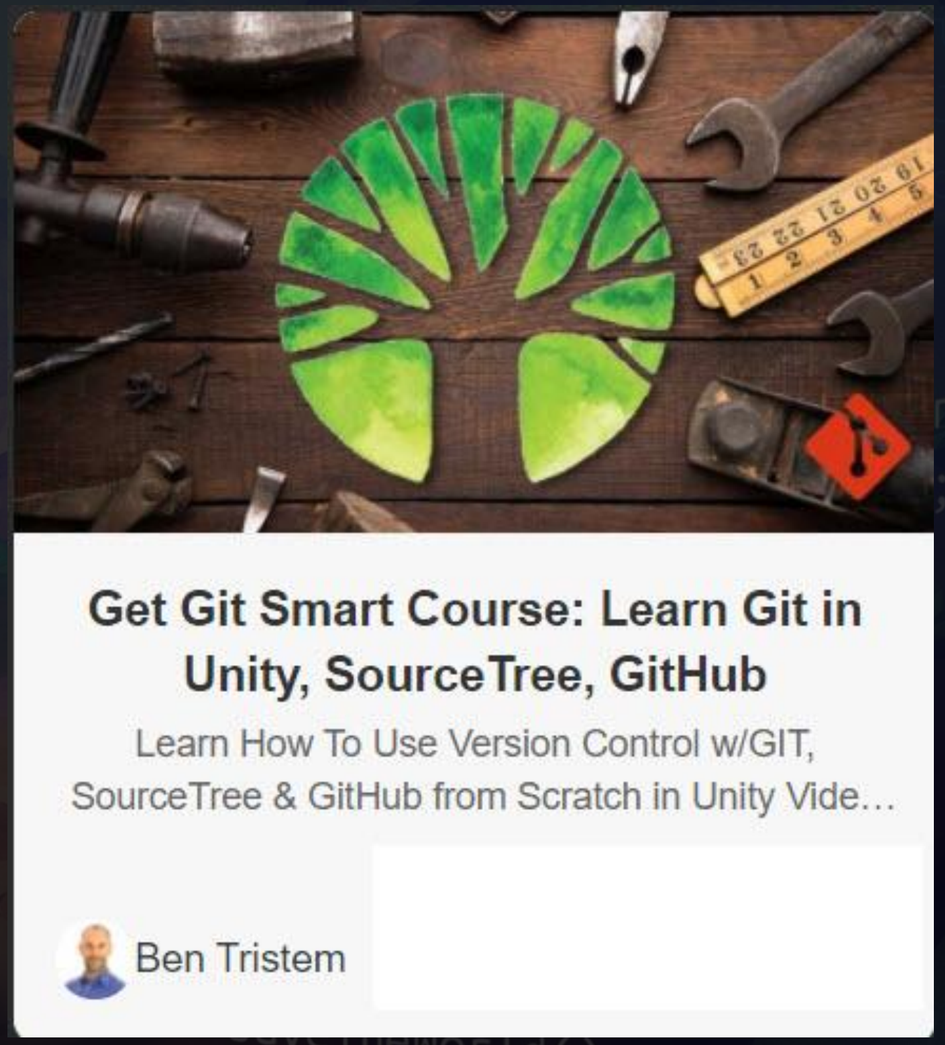
- Source Control: a system for tracking and managing changes to your code and your project
- Git: A type of version control system that tracks changes to files
- Gitlab (also, Github): Repository hosting service
- Repository (Repo): Directory or storage space for your project
- SourceTree: Desktop client to help view your repo

Why Use Source Control?

- As a backup to save your project
- To allow you to safely make changes to your project knowing you can go back to a previous version
- To explore multiple ideas at the same time
- To more easily collaborate on projects

Setting Up Your Own Repo

- Now is a good time to investigate how to set up your own repo
- Google “set up git repo”
- Check out Ben’s *Git Smart* course at GameDev.tv



Accessing My Project Changes

- After every lecture, I commit my project changes to source control
  - Every lecture will have a link to my project changes
- You can easily access my project to see what has changed or compare my code to yours
- You can also download my entire project to explore it

GameDev.tv

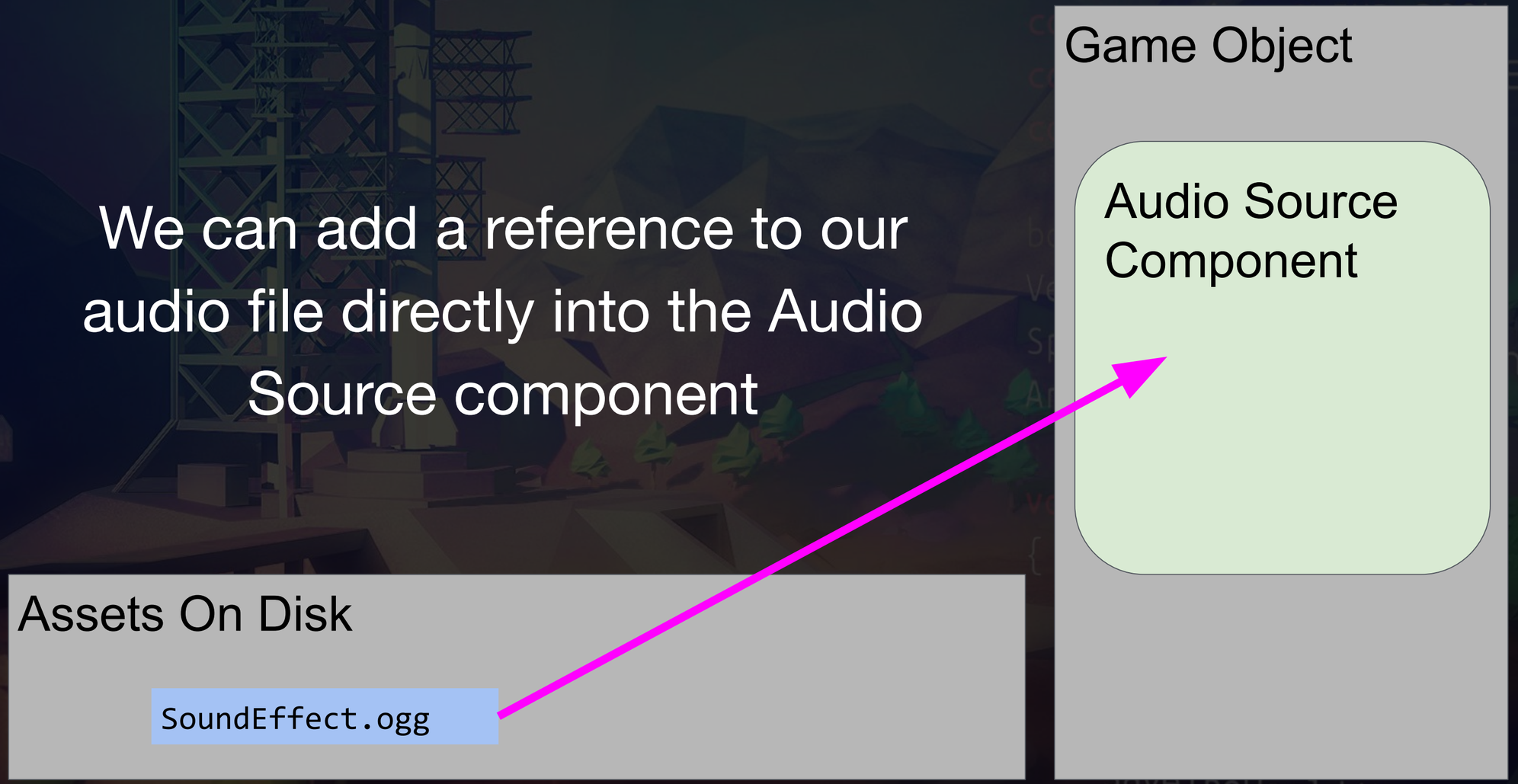
Unity 2020.1

Unity Audio Introduction

3 Main Things We Need

Linking Components To Assets

We can add a reference to our audio file directly into the Audio Source component



GameDev.tv

Unity 2020.1

Play AudioSource SFX

Play SFX When Thrusting

- Cache a reference to AudioSource called `audioSource`
- Use `audioSource.Play()` to play when we are thrusting
- Use `!audioSource.isPlaying` to make sure we only play if we aren't already playing (Note: `!` = not true)
- Use an `else` condition and `audioSource.Stop()` to stop our SFX when we aren't thrusting.

GameDev.tv

Unity 2020.1

Switch Statements

Switch Statements

- Switch Statements are a Conditional like If / Else Statements.
- Allow us to compare a single variable (variables can change or “vary”) to a series of constants (ie. things that don’t change or have a “constant” value).

Switch Syntax

```
switch (variableToCompare)
  {
      case valueA:
          ActionToTake();
          break;
      case valueB:
          OtherAction();
          break;
      default:
          YetAnotherAction();
          break;
  }
```

Print Different Collision Situations

- Using a Switch Statement, print to the console different messages based upon what we collide into.
- Remember that our collision tag returns a “string”.
- Our variable will be: `other.gameObject.tag`
- A reminder of the Switch syntax...

Switch Syntax

```
switch (variableToCompare)
  {
      case valueA:
          ActionToTake();
          break;
      case valueB:
          OtherAction();
          break;
      default:
          YetAnotherAction();
          break;
  }
```

GameDev.tv

Unity 2020.1

Respawn Using SceneManager

Reload Current Scene

- Use `SceneManager.LoadScene()` to load the current scene and therefore respawn our rocket ship when it collides with the ground
- You'll need to use the `UnityEngine.SceneManagement` namespace

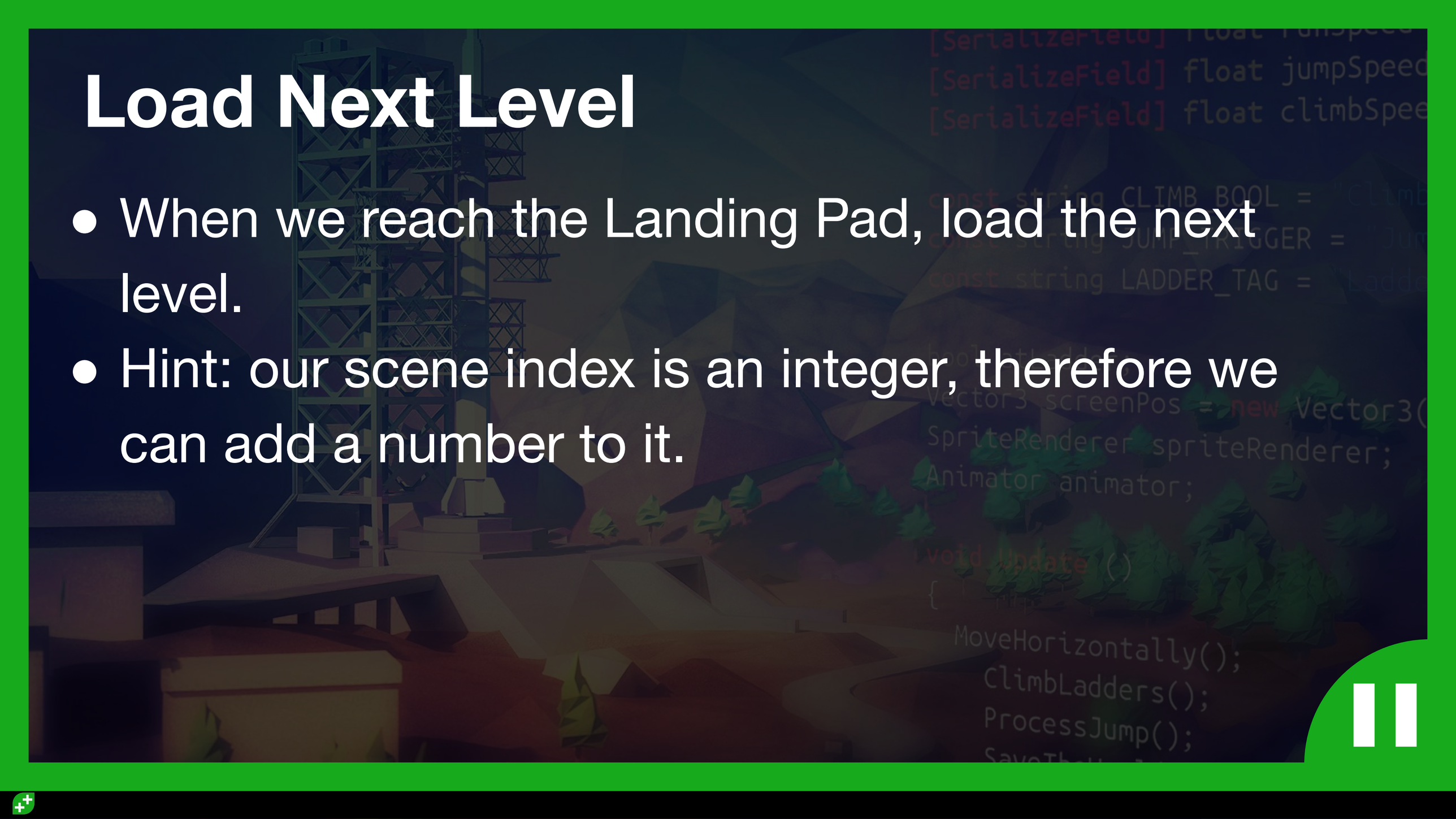
GameDev.tv

Unity 2020.1

Load Next Level

Load Next Level

- When we reach the Landing Pad, load the next level.
- Hint: our scene index is an integer, therefore we can add a number to it.



GameDev.tv

Unity 2020.1

Using Invoke

Using Invoke

- Using `Invoke()` allows us to call a method so it executes after a delay of x seconds.
- Syntax:

```
Invoke(“MethodName”, delayInSeconds);
```

- Pros: Quick and easy to use
- Cons: String reference; not as performant as using a Coroutine

Use Invoke To Delay Next Level

- Parameterise our delay (ie. make a variable which we can tune in the inspector)
- When we reach the Landing Pad, make sure we have a delay before loading the next level
- Stop player controls during the delay

GameDev.tv

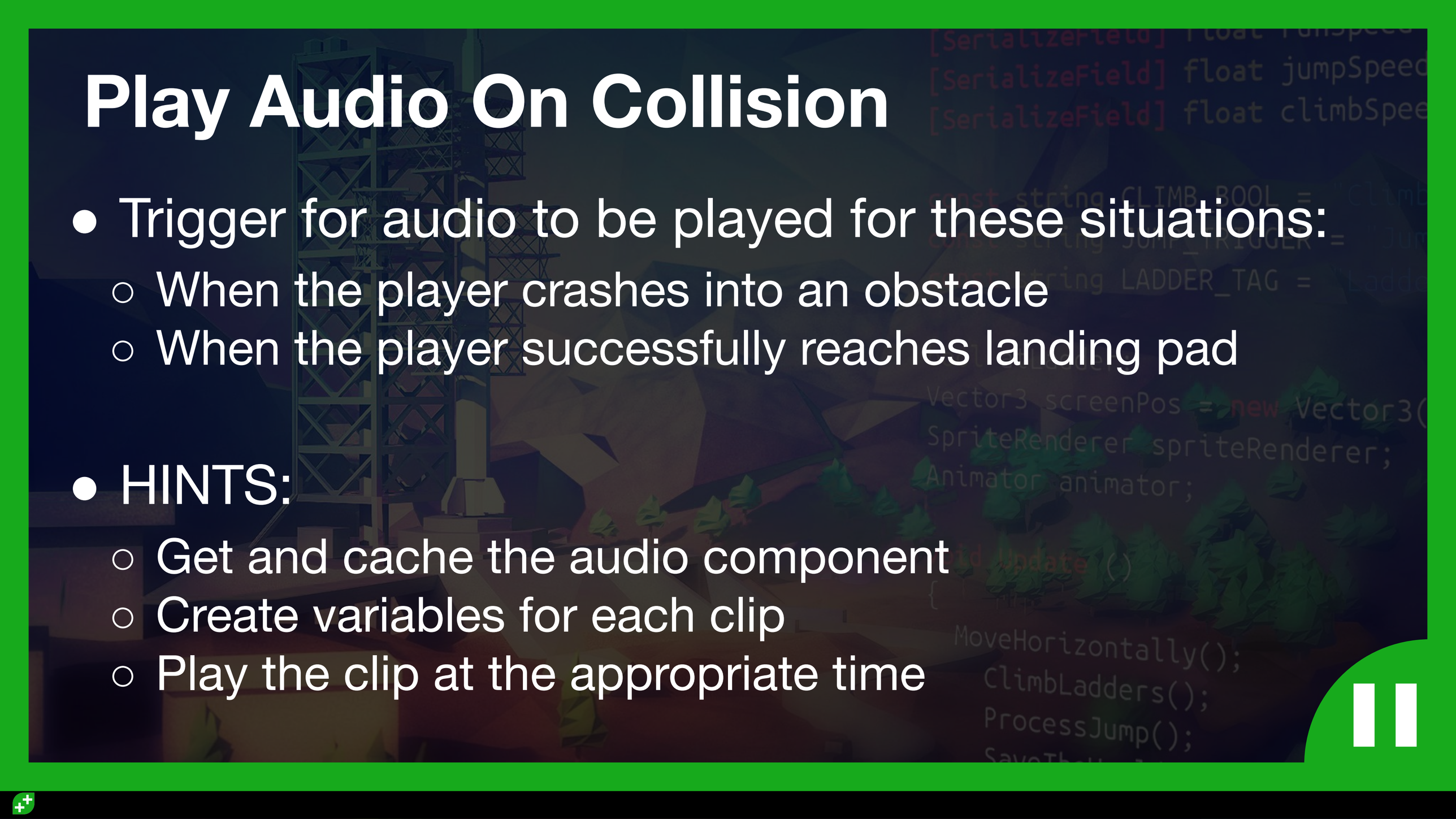
Unity 2020.1

Multiple Audio Clips

Play Audio On Collision

- Trigger for audio to be played for these situations:
  - When the player crashes into an obstacle
  - When the player successfully reaches landing pad

- HINTS:
  - Get and cache the audio component
  - Create variables for each clip
  - Play the clip at the appropriate time



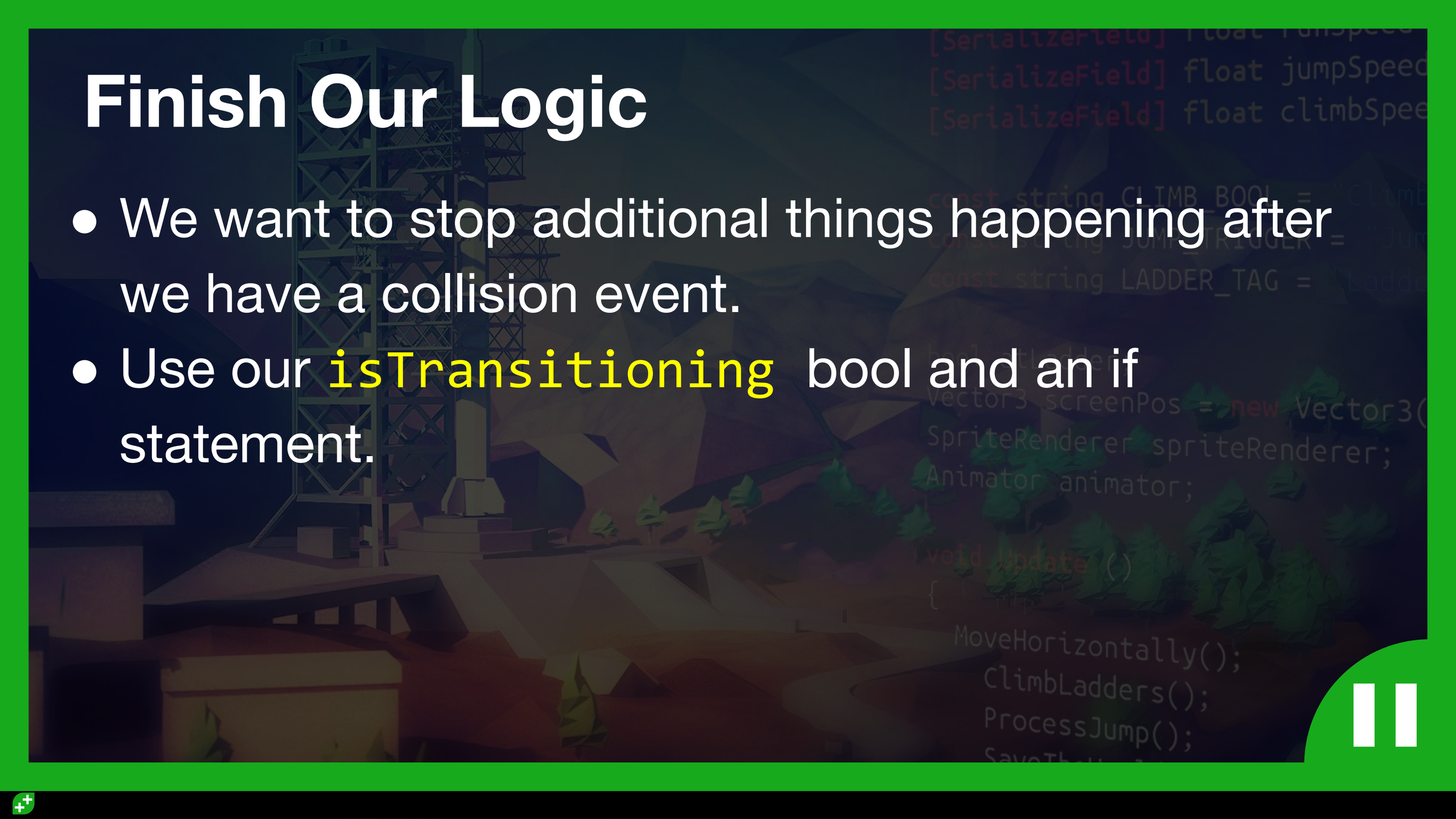
GameDev.tv

Unity 2020.1

Bool Variable For State

Finish Our Logic

- We want to stop additional things happening after we have a collision event.
- Use our `isTransitioning` bool and an if statement.



GameDev.tv

Unity 2020.1

Make Rocket Look Spiffy

Make Your Rocket Look Different

- Using basic shapes, update your rocket prefab to look interesting.
- Add a splash of colour.
- Consider the pivot point so that the rotation feels right.

GameDev.tv

Unity 2020.1

How To Trigger Particles

Particles System Component

Particle System is a Component added to a Game Object

We use Modules for controlling behaviour

Each particle is not a Game Object

Play The Particle Effect

- We use `ParticleSystem.Play()` to play our particle system when triggered...
- When we crash, play the explosion particles
- When we reach landing pad, play success particles

GameDev.tv

Unity 2020.1

Particles For Rocket Boosters

Rick Davidson

Get Ourselves Ready To Trigger

- Set up your rocket so that it has 3 particle systems, ready for us to trigger in code:
  - Main booster
  - Left booster
  - Right booster
- Add them in your prefab and create variables for each

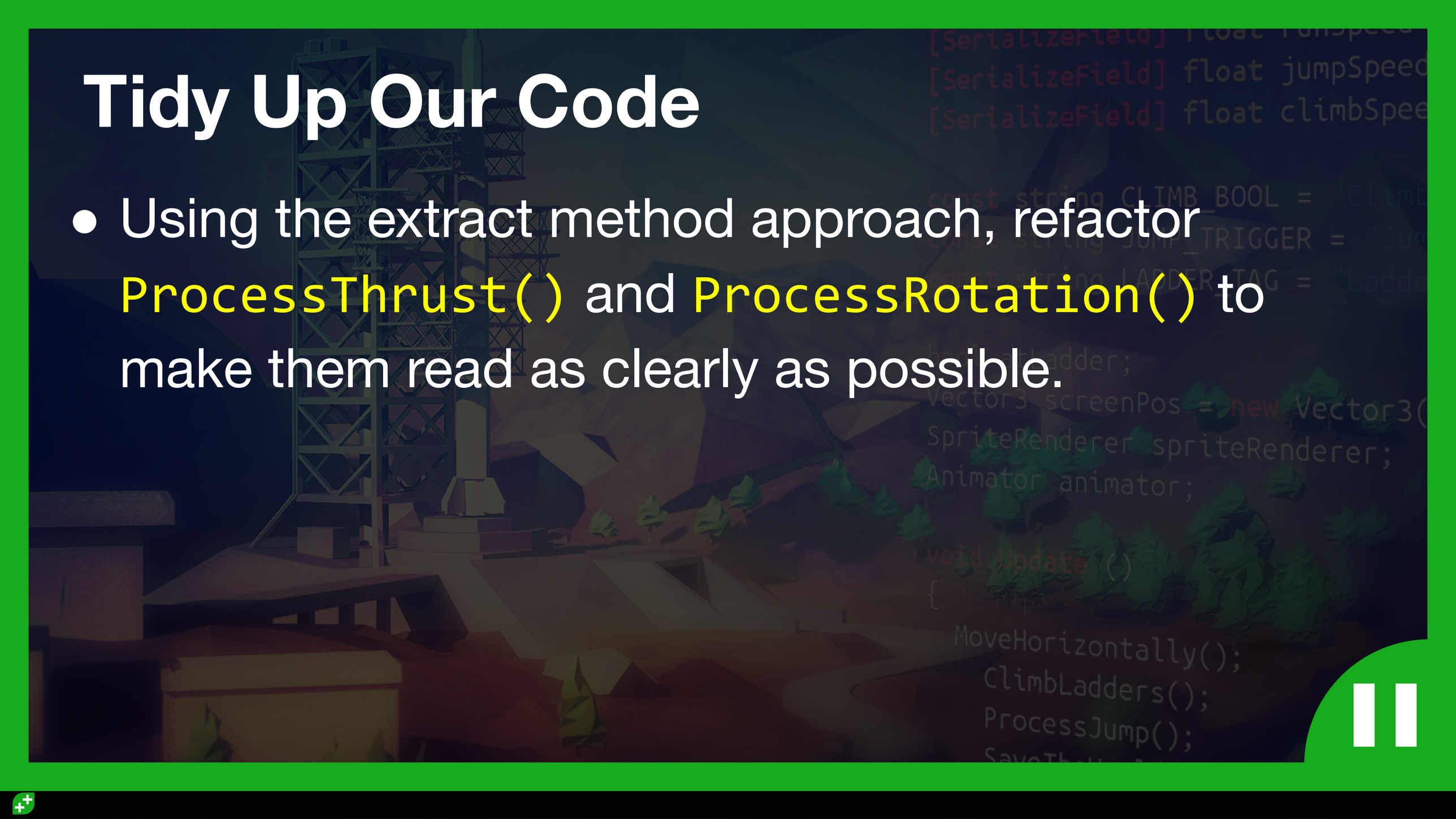
GameDev.tv

Unity 2020.1

Refactor With Extract Method

Tidy Up Our Code

- Using the extract method approach, refactor `ProcessThrust()` and `ProcessRotation()` to make them read as clearly as possible.



GameDev.tv

Unity 2020.1

Add Cheat / Debug Keys

Rick Davidson

Tricky Advanced Challenge!

- When the user pushes the ‘L’ key, load the next level
- When the user pushes the ‘C’ key, disable collisions

GameDev.tv

Unity 2020.1

Make Environment From Cubes

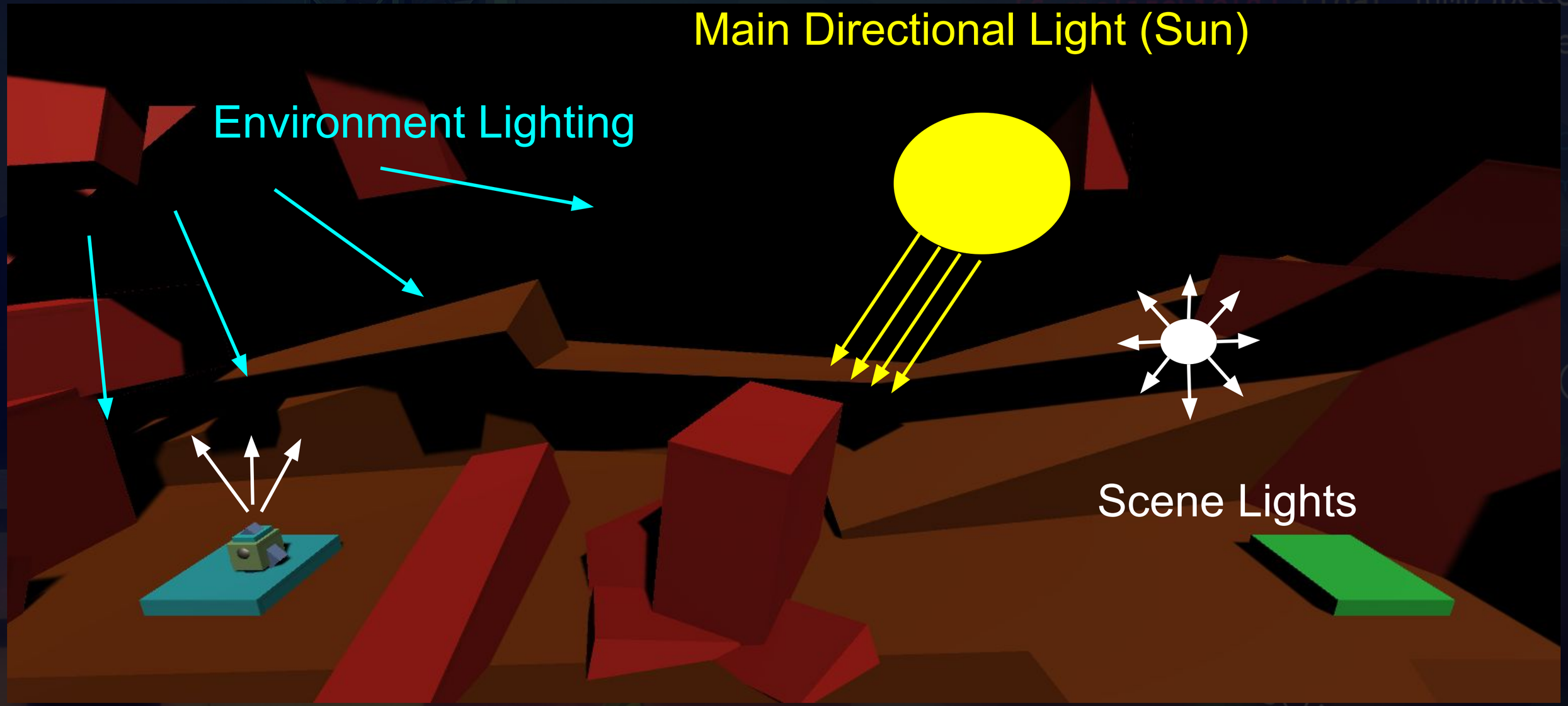
Make Your Levels Look Interesting

- Using just cubes or other primitives, add some depth to the look of your level.
- Try to frame your world so the player can’t fly off where they shouldn’t.

GameDev.tv

Unity 2020.1

How To Add Lights In Unity


Main Directional Light (Sun)
Environment Lighting
Scene Lights

Light Your Scene

- Add at least one point light and one spotlight to your scene.
- Experiment with a lighting setup that you’re happy with.
- Share a screenshot!

GameDev.tv

Unity 2020.1

Move Obstacle With Code

To Move In A Consistent Direction

Starting position

(x, y, z)

Movement Vector

(x, y, z)

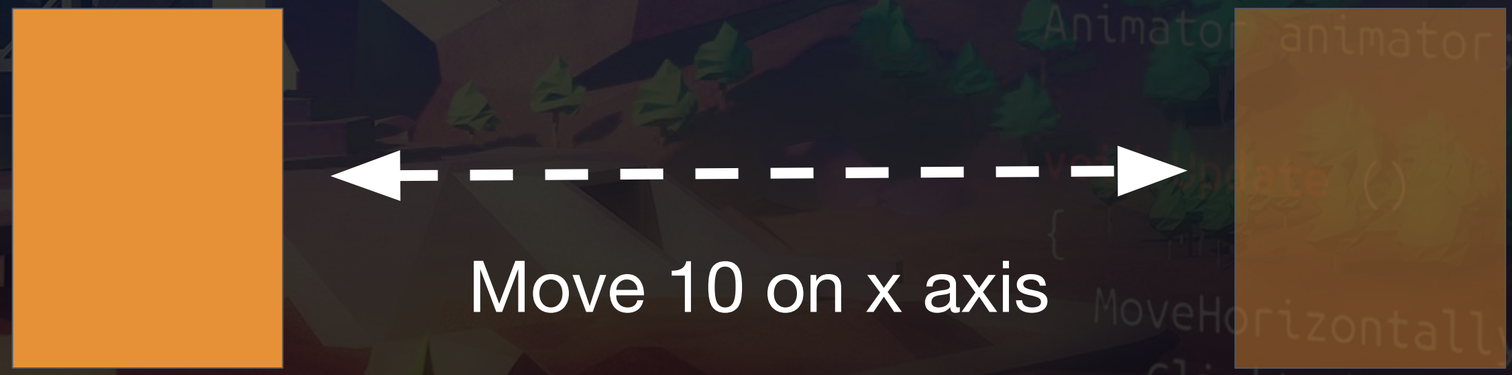
To Move In A Consistent Direction

Starting position

(0, 0, 0)

Movement Vector

(10, 0, 0)



Move 10 on x axis

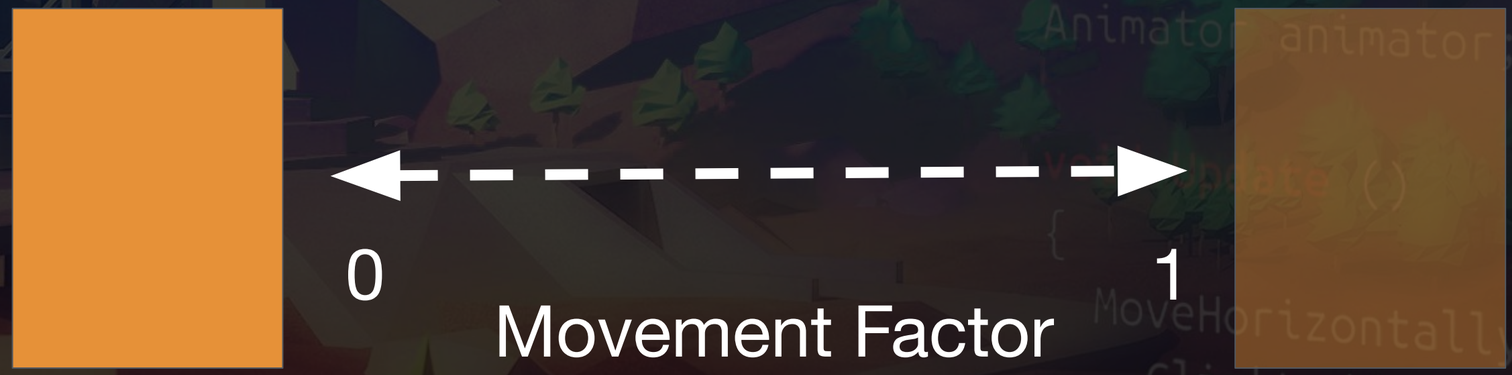
To Oscillate (Back & Forth)

Starting position

(0, 0, 0)

Movement Vector

(10, 0, 0)



0

1

Movement Factor

(between 0 and 1)

To Oscillate (Back & Forth)

Starting position

(0, 0, 0)

Offset =

Movement Vector

(10, 0, 0)

Eg.
Movement Factor
is currently 0.5,
Offset is?...

5, 0, 0

*

0

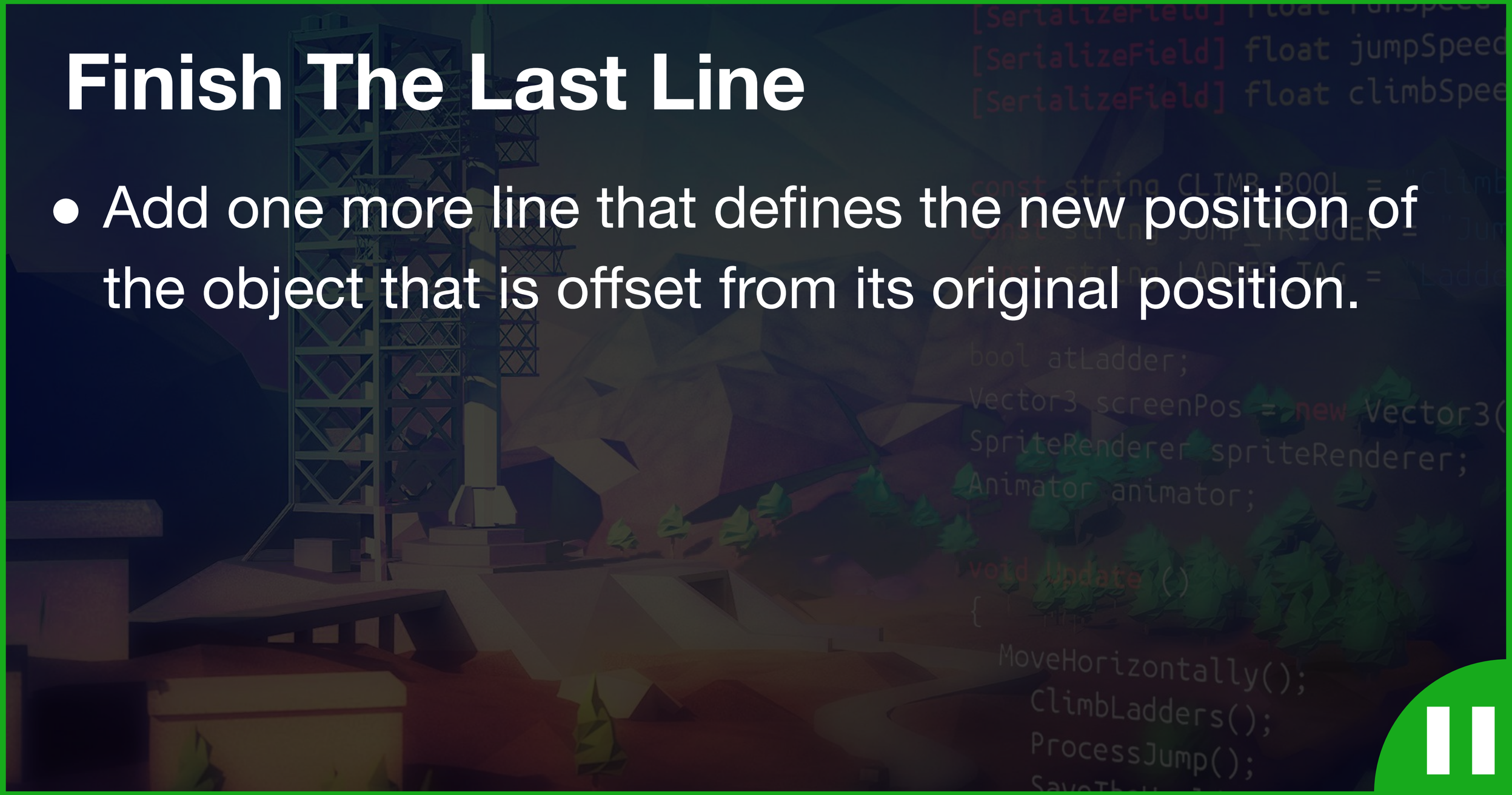
1

Movement Factor

(between 0 and 1)

Finish The Last Line

- Add one more line that defines the new position of the object that is offset from its original position.



GameDev.tv

Unity 2020.1

Mathf.Sin() For Oscillation

Make Something Move

- Oscillate something in your level so that it provides an interesting challenge for your player.

GameDev.tv

Unity 2020.1

Protect Against NaN Error

Try And Protect Zero Period

- Try and put some protection code in.
- The goal is to eliminate the NaN error when period is 0.

Hint: Remember our Discord chat server!

Notes About Comparing `floats`

- Two `floats` can vary by a tiny amount.
- It's unpredictable to use `==` with `floats`.
- Always specify the acceptable difference.
- The smallest float is `Mathf.Epsilon`
- Always compare to this rather than zero.
- For example...

```
if (period <= Mathf.Epsilon) { return; }
```

GameDev.tv

Unity 2020.1

Designing Level Moments

Useful Game Design Approach

- Design “moments” and then expand them into a level. Moments that use the environment:
  - Fly under
  - Fly over
  - Fly through a gap
  - Time your flight through moving obstacle
  - Land on moving platform
  - Fly through narrow tunnel

Useful Game Design Approach

- Moments that use tuning of our existing game:
  - Slower rocket (eg. it got damaged)
  - Faster rocket (eg. got a boost)
  - Darker level
  - Closer camera
  - Bigger rocket (carrying something)
  - Reversed controls
  - And so on...

Level Design Challenge (If Interested)

- Create 5 (or more) different levels, each with a unique game moment
- Add the levels to the build settings so they are playable
- Share a video with us of your one best moment (OBS is a free recording package)

GameDev.tv

Unity 2020.1

Quit Application

Hit Escape To Quit

- Create a new script called `QuitApplication.cs`
- Create logic so that if the player hits escape on their keyboard we call `Application.Quit()`
- Add a debug line to make sure that hitting escape works.

GameDev.tv

Unity 2020.1

How To Build & Publish A Game

Make A WebGL Build

- Make a WebGL build of your game
- Publish to sharemygame.com
- Tell our community about your game and where to find it
- Play at least one other person’s game (good karma)

GameDev.tv

Unity 2020.1

Wrap Up - Project Boost

Rick Davidson


[SerializeField] float runSpeed =
[SerializeField] float jumpSpeed =
[SerializeField] float climbSpeed
const string CLIMB_BOOL = "Climbi
const string JUMP_TRIGGER = "Jump
const string LADDER_TAG = "Ladder
bool atLadder;
Vector3 screenPos = new Vector3();
SpriteRenderer spriteRenderer;
Animator animator;
void Update ()
{
MoveHorizontally();
ClimbLadders();
ProcessJump();
SaveTheWorld();
}

GameDev.tv

Unity 2020.1

The Origin Of Our World

Which Axis Is Which...

+x = right

+y = up

+z = forward

Setup Your World

- Your ground level is at `y = 0`.
- The launchpad is centered on `x = 0, z = 0`.
- You have an initial camera view you like.
- Everything in the Hierarchy is “prefabbed”.
- You have assigned terrain colour.
- You’ve modified the directional light rotation.
- You have shared a screenshot.

GameDev.tv

Unity 2020.1

Placeholder Art From Primitives

Setting-up Compound Objects

- Keep mesh away from top-level so easy to swap.
- Keep top-level object scale close to (1,1,1).
- Beware of Pivot / Centre option (Z key).
- Check object rotates, scales and instantiates ok.

Your Version 1 Ship Is… Shipped

- You have an INITIAL ship you're happy with.
- It's obvious which way is up.
- It has a splash of colour on it.
- It rotates around what looks like it's centre.
- It should have a prefab, and z is into background.
- Drag prefab to Hierarchy puts rocket on launchpad.
- Share a close-up on our community forum.

GameDev.tv

Unity 2020.1

Basic Input Binding

Create Rotation Keys

- Pushing A should repeatedly print “Rotating left”.
- Pushing D should do the same for right.
- You should be able to thrust AND rotate.
- You should not be able to rotate both ways at the same time.

GameDev.tv

Unity 2020.1

Physics and Rigidbodies

Using `GetComponent<>()`

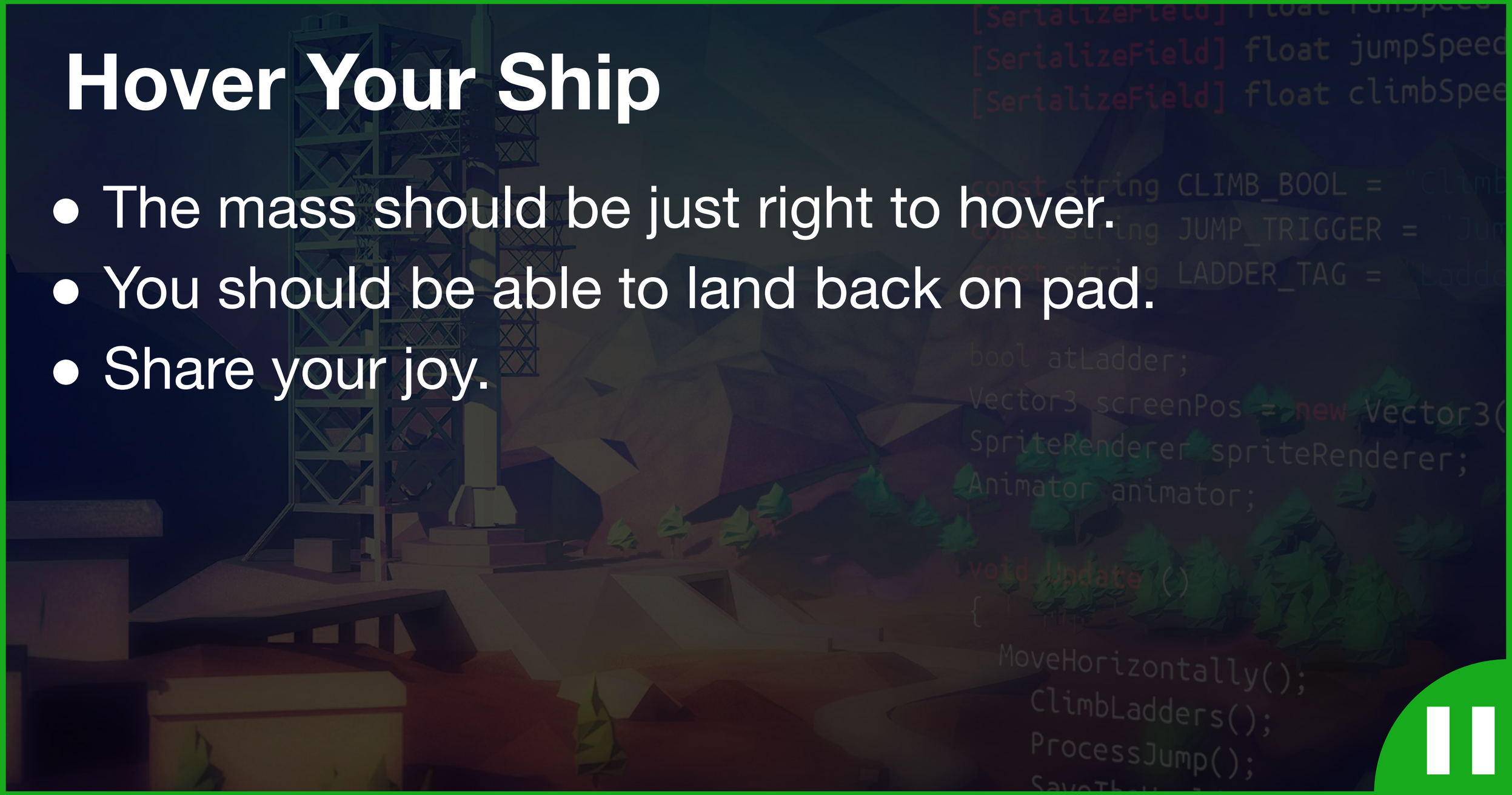
Use the following template to create a `rigidBody` member variable in your code, which allows you to access the rigid body on the same game object…

```
rigidBody = GetComponent<RigidBody>();
```

… pay particular attention to capitalisation.

Hover Your Ship

- The mass should be just right to hover.
- You should be able to land back on pad.
- Share your joy.



GameDev.tv

Unity 2020.1

Coordinate System Handedness

Unity Uses A Left-Handed System

Try Labelling Your Fingers!

- You should understand the difference in the systems between right and left hand.
- Remember to label your fingers in a circular way, but it doesn’t matter where you start.

GameDev.tv

Unity 2020.1

Using Time.deltaTime

Frame-rate Independence

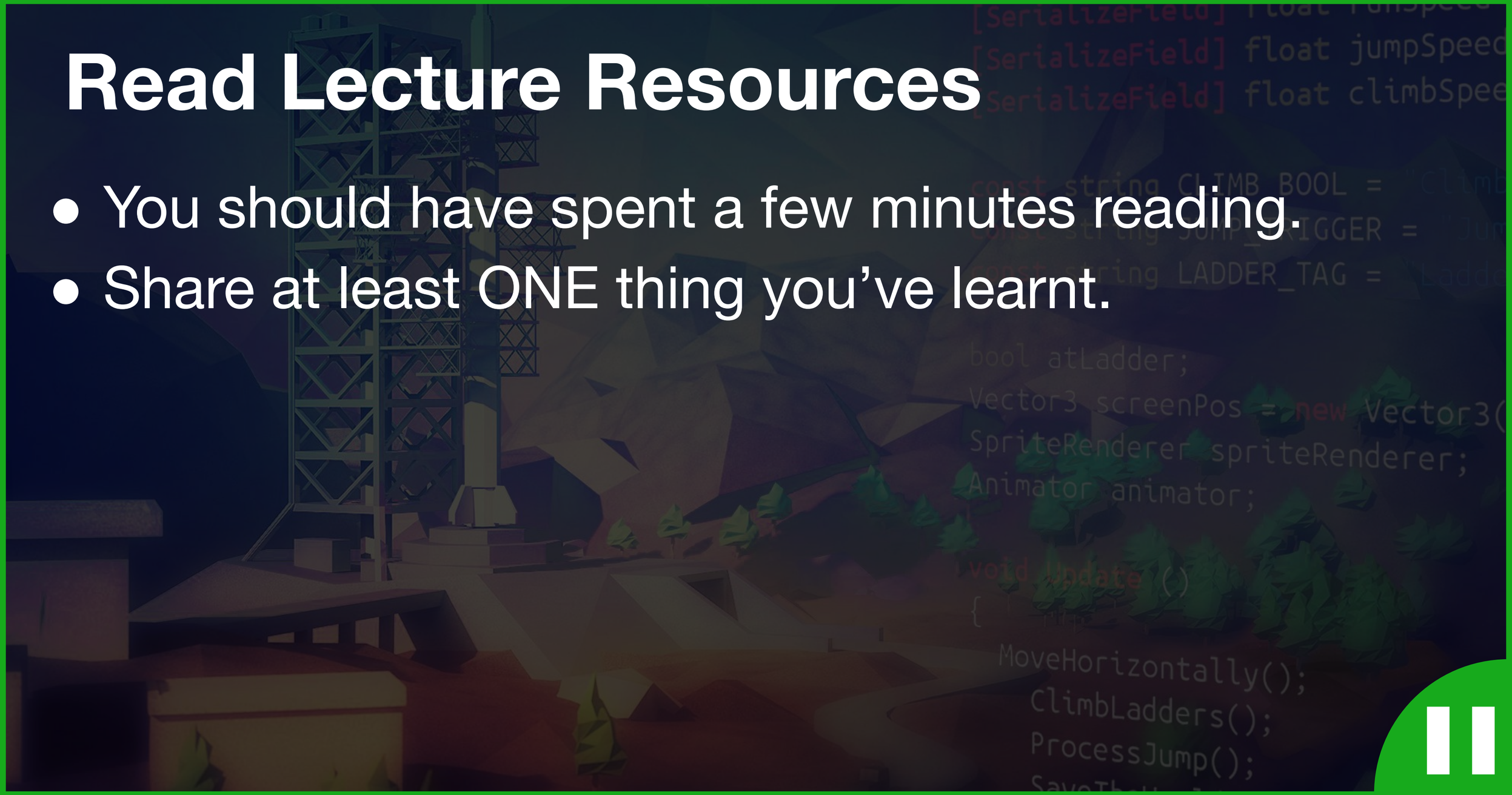
- The time each frame takes can vary wildly.
- `Time.deltaTime` tells you the last frame time.
- This is a good predictor of the current frame time.
- We can use this to adjust our movement.
- Longer frames lead to more movement.
- Shorter frames lead to less movement.
- e.g. `rotation = rcsThrust * Time.deltaTime;`

Words For Parts Of A Transform

	As A Noun	As A Verb	Code Example
Transform	Position	Translate	`transform.Translate();`
	Rotation	Rotate	`transform.Rotate();`
	Scale	Scale	`transform.localScale;`

Read Lecture Resources

- You should have spent a few minutes reading.
- Share at least ONE thing you've learnt.



GameDev.tv

Unity 2020.1

Instructor Hangout 3.1

In This Hangout...

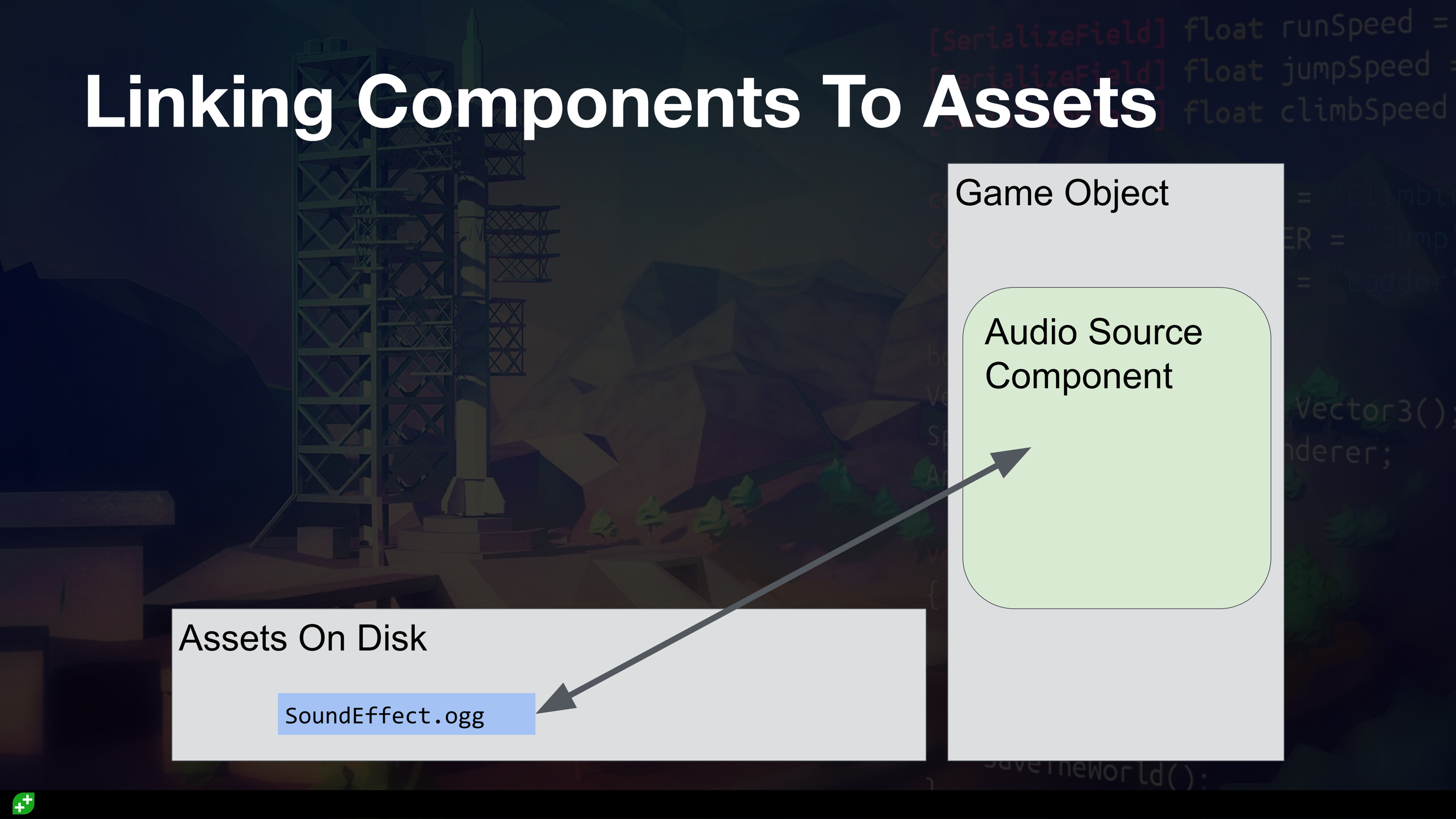
- Why Git rather than Unity Collab (Jason).
- Clarifying the handedness rule finger order.
- Struggling SourceTree on Mac? Forum (Frank).
- How to re-centre pivot point on rocket (Rory).
- Adding box collider to odd shaped rocket (Andy).
- Adding [Prefix] to Q&A question and comments.
- Mad How Disease, and that 1000y old text!

GameDev.tv

Unity 2020.1

Adding A Touch Of Audio

Linking Components To Assets



Start And Stop Audio

- The audio should start when you thrust.
- It should stop immediately you stop thrusting.
- There should be no weird audio artifacts.
- Why not make a little video and share it?

GameDev.tv

Unity 2020.1

Resolving Movement Bugs

“Don’t patch over problems”

Your Movement Is Bug Free

- You can move from one platform to another.
- A platform can induce spin but it stops on thrust.
- You have Drag value you’re happy with.
- Your code is beautiful.
- Share a < 20s video or GIF of your gameplay.

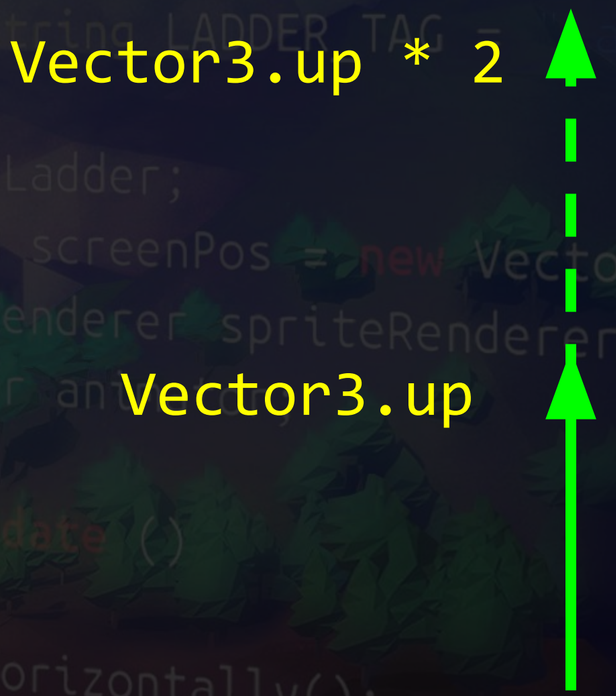
GameDev.tv

Unity 2020.1

Using [SerializeField] vs public

Multiplying Vectors

- Multiply a vector by a float.
- You end-up with a new vector.
- New vector is parallel.
- It’s a different length.
- Works for rotation and translation.



Exposing Member Variables

Modifier	Change In Inspector?	Change From Other Scripts?
`[SerializeField]`	Yes	No
`public`	Yes	Yes

Serialise `mainThrust`

- Main Thrust should be adjustable in the inspector.
- The ship's Rigid Body component should be reset.
- The ship should handle similarly to before.

Enjoy fooling around with your ship!

GameDev.tv

Unity 2020.1

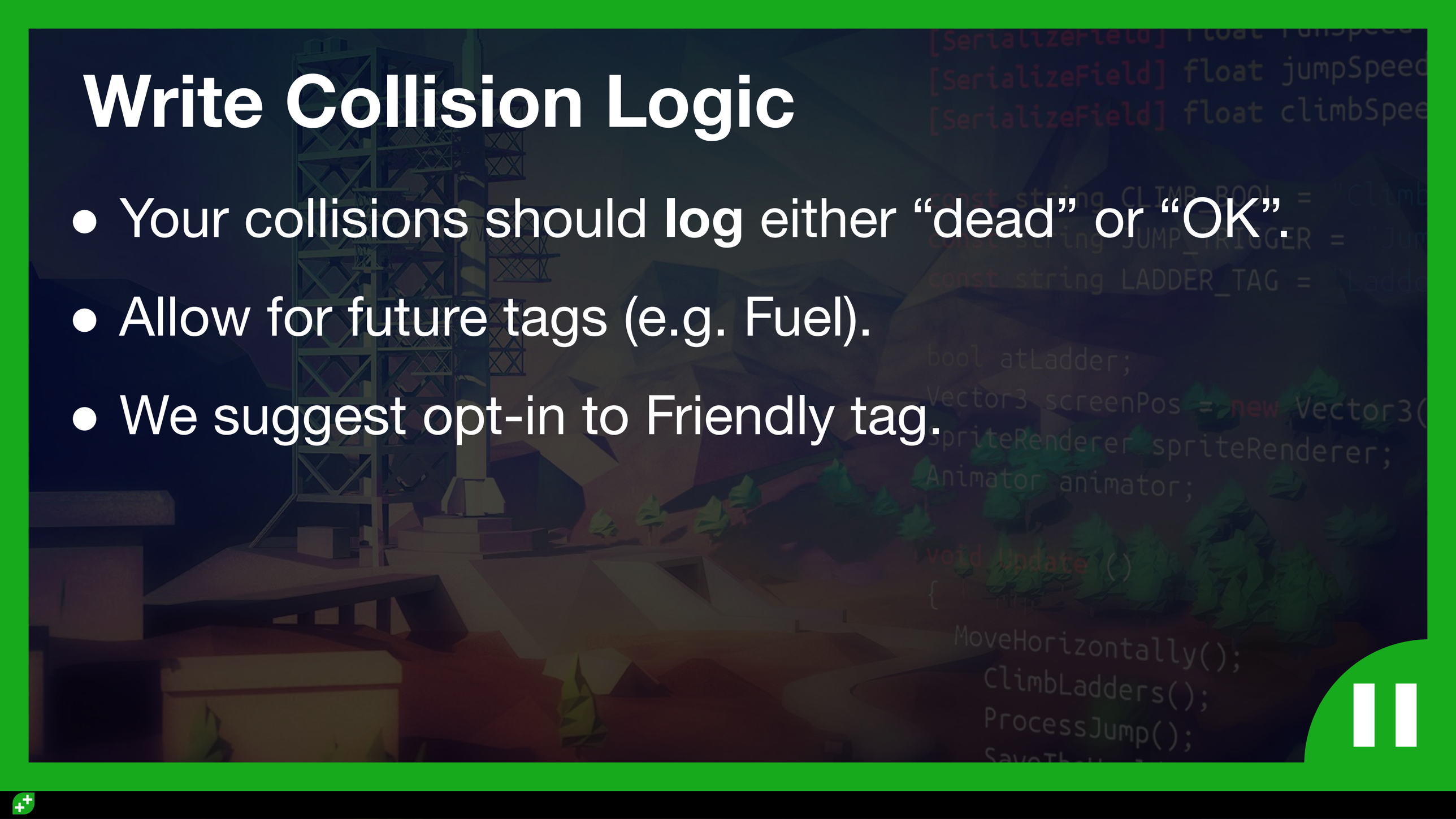
Tagging Game Objects As Friendly

Pros and Cons of Tags

Pros	Cons
Just one per game object	Is based on a string
Very simple to use in Inspector	Have to rename in 2 places
Makes for clear code	Nothing “keeps you honest”

Write Collision Logic

- Your collisions should **log** either “dead” or “OK”.
- Allow for future tags (e.g. Fuel).
- We suggest opt-in to Friendly tag.



GameDev.tv

Unity 2020.1

Basic Level Design

Designing Our First Level

- Levels are a series of interesting moments.
- Always refer back to your game design intention for Player Experience.
  - Our is: “Skilled rocket pilot”
- There is an ongoing tension between gameplay tuning and level tuning.

Create An Interesting Moment

- Refine your camera if need be.
- Place start, finish and obstacles to create an interesting moment.
- Playtesting and refine.
- Share a screenshot with the community.

GameDev.tv

Unity 2020.1

Design Levers And Variation

As Indie Developers...

- Try to use what we have to make new / different / better gameplay before adding features.
- What are our current design levers?
  - Rotation, thrust, gravity, size, level layout, lighting, friendly / unfriendly, camera, goal...
- When prototyping, go to the extremes.

Prototype Something Fresh

- Pull your design levers to create something different to us.
- Remember to commit to your repo beforehand, in case you need to revert!
- Share your idea.

GameDev.tv

Unity 2020.1

Making A Second Level

Some Options For Multiple Levels

1. Create a new Unity scene for each level
2. Place all the levels in one scene and retarget the camera
3. Stitch the levels one after another and use a scrolling camera

Create A Second Level

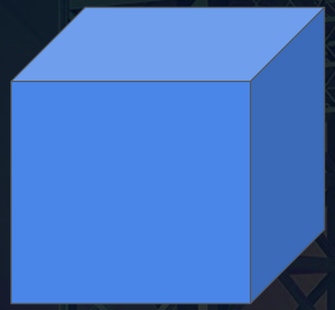
- Duplicate your scene.
- Create a new level based upon a different type of moment.

GameDev.tv

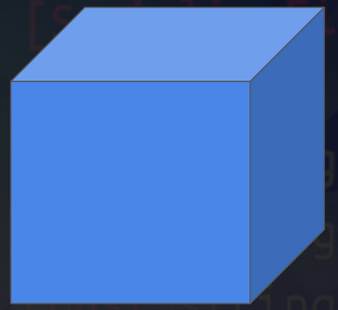
Unity 2020.1

Prefabs In Detail

Create Object



Prefab & Save Scene



Position & Rotation

All other details

`.unity` scene file

Position & Rotation

All other details

New .prefab file

Explore Prefabs

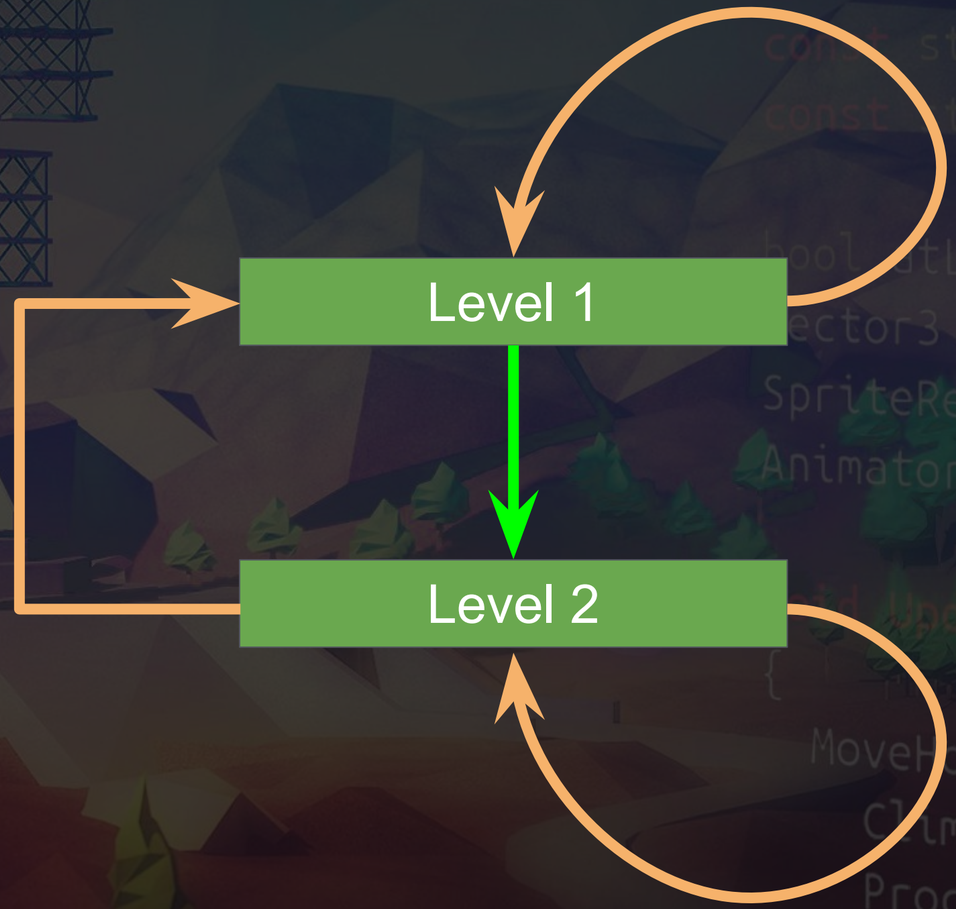
- Create a sandbox scene.
- Explore prefabs until you feel you “get it”.
- Duplicate Launch Pad, prefab it to Landing Pad and tag as “finish”.

GameDev.tv

Unity 2020.1

Level Loading & Scene Management

Your Game Cycles Through 2 Levels

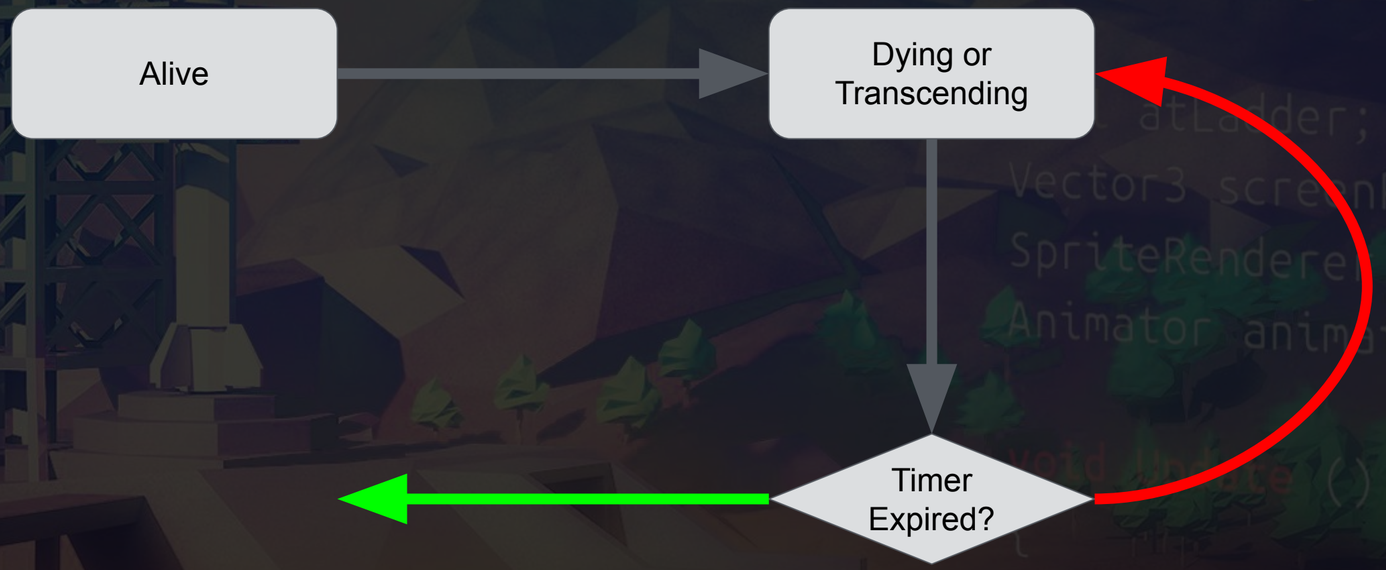


GameDev.tv

Unity 2020.1

Invoke() As A Coroutine Warm-up

Delaying Level Load



Remember other messages still arrive while waiting for timer

Delay Level Load On Death

- The first level should still load when you die.
- There should be a delay before it does so.
- Player controls should be disabled while dying.
- We suggest you create a new method.

GameDev.tv

Unity 2020.1

Instructor Hangout 3.2

In This Hangout...

- Abrupt sound stopping issue (thanks Gregory).
- Care of differences in Debug mode (thanks Jeff).
- Side-effect in FreezeRotation + code reviews (Jeff).
- Well done Morgaine for 1st screen recording!
- Curtis & Robert re “too slick for neophytes”.
- Default values & `[SerializeField]` (Mitchell)
- Frame-rate & `FixedUpdate` (Straesso).

… continued

- Tip about solid background (Manuel).
- Loving the levels on forum (resources).
- Keep engaging even if it’s all clear!

GameDev.tv

Unity 2020.1

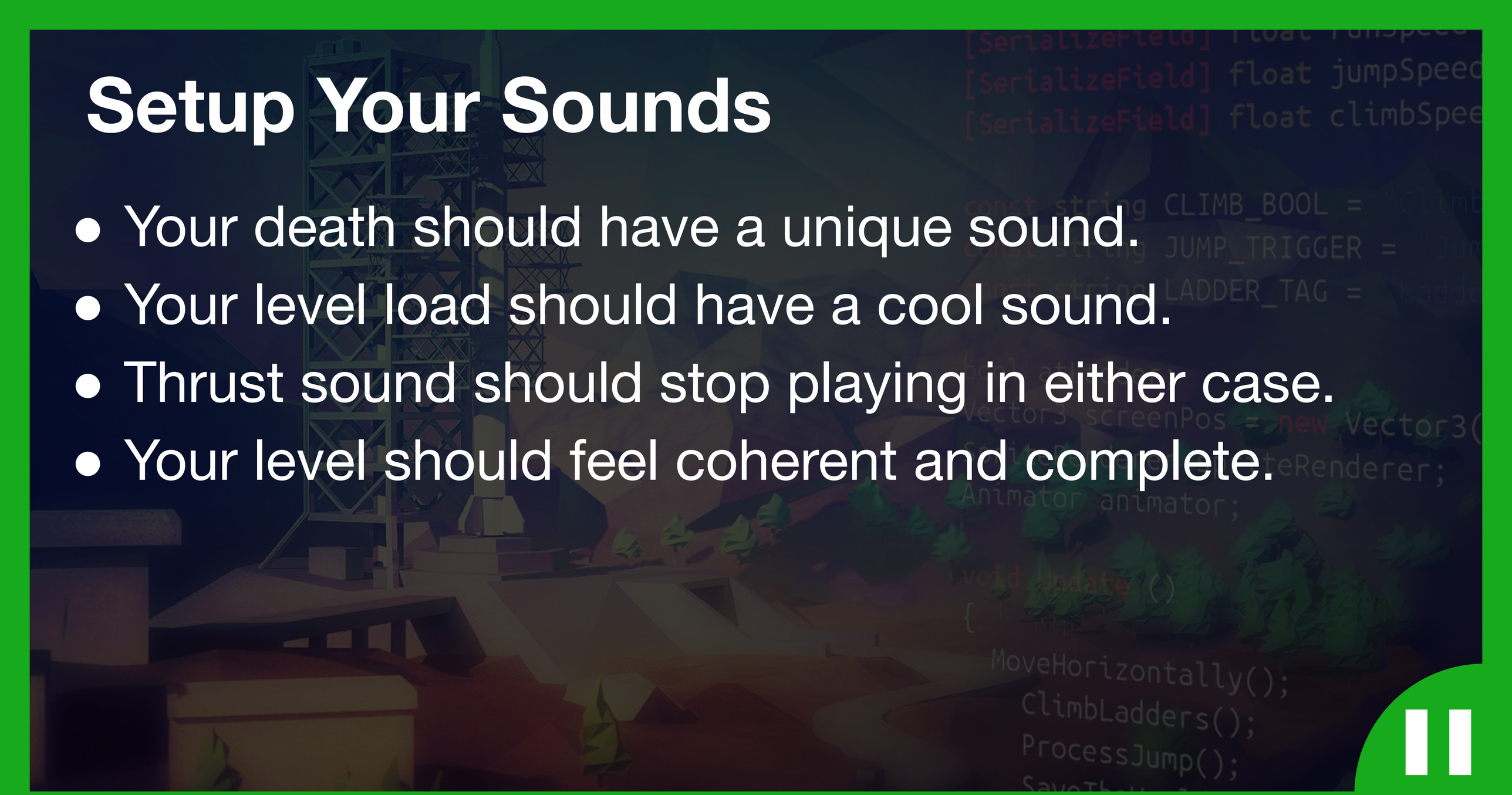
Playing Multiple Audio Clips

An Alternative Way Of Playing Audio

- Still have an audio source.
- No need to have a default clip.
- Specify the clips as `[SerializeField]` “levers”.
- Use `audioSource.PlayOneShot(clipName);`

Setup Your Sounds

- Your death should have a unique sound.
- Your level load should have a cool sound.
- Thrust sound should stop playing in either case.
- Your level should feel coherent and complete.



GameDev.tv

Unity 2020.1

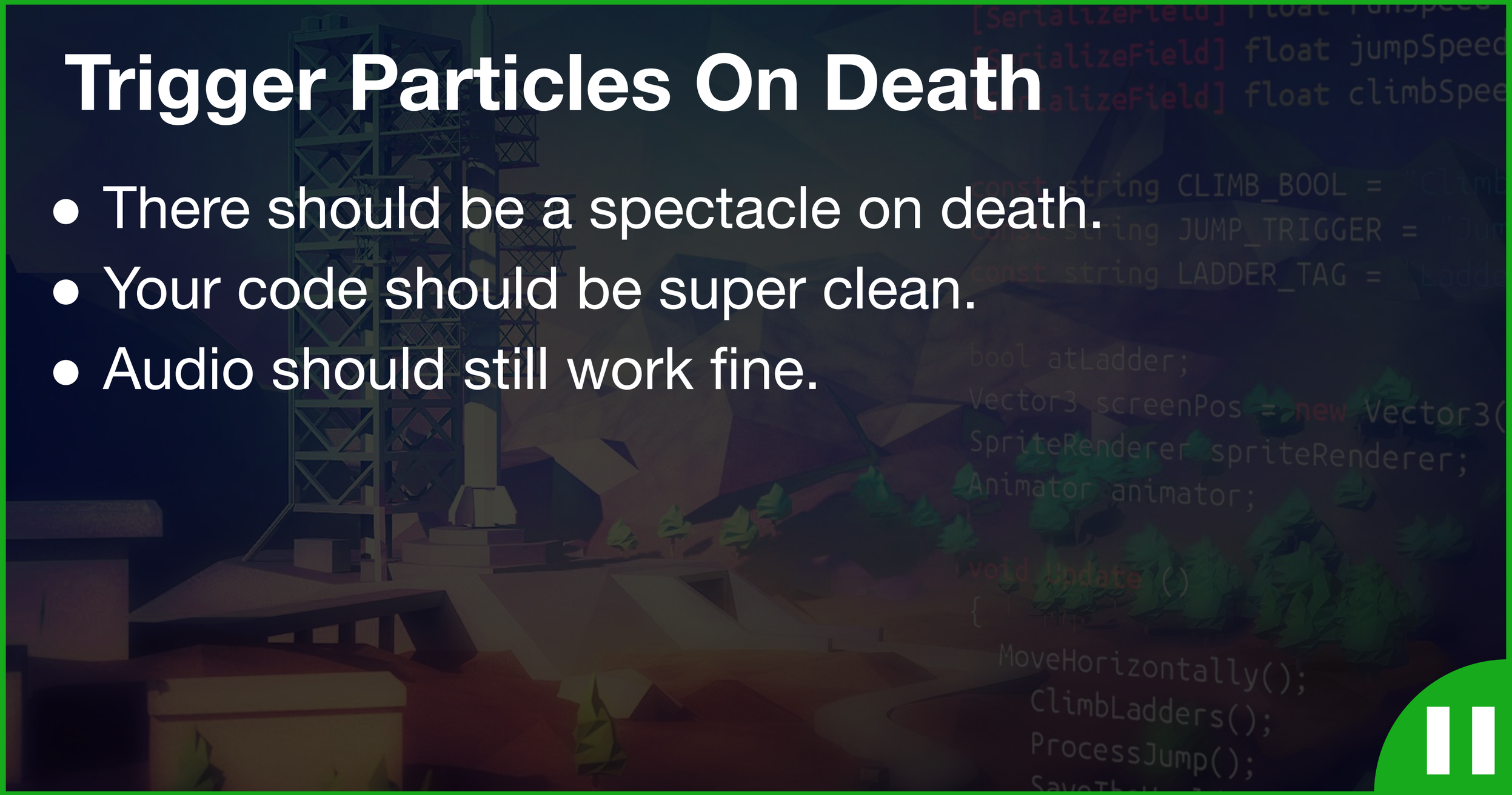
Introducing Particle Effects

Particle Systems Guidelines

- Use a separate game object for particle system.
- Consider disabling “Play On Awake”
- `[SerializeField] ParticleSystem name;`
- Trigger using `name.Play();`
- ENJOY the visual carnage!

Trigger Particles On Death

- There should be a spectacle on death.
- Your code should be super clean.
- Audio should still work fine.



GameDev.tv

Unity 2020.1

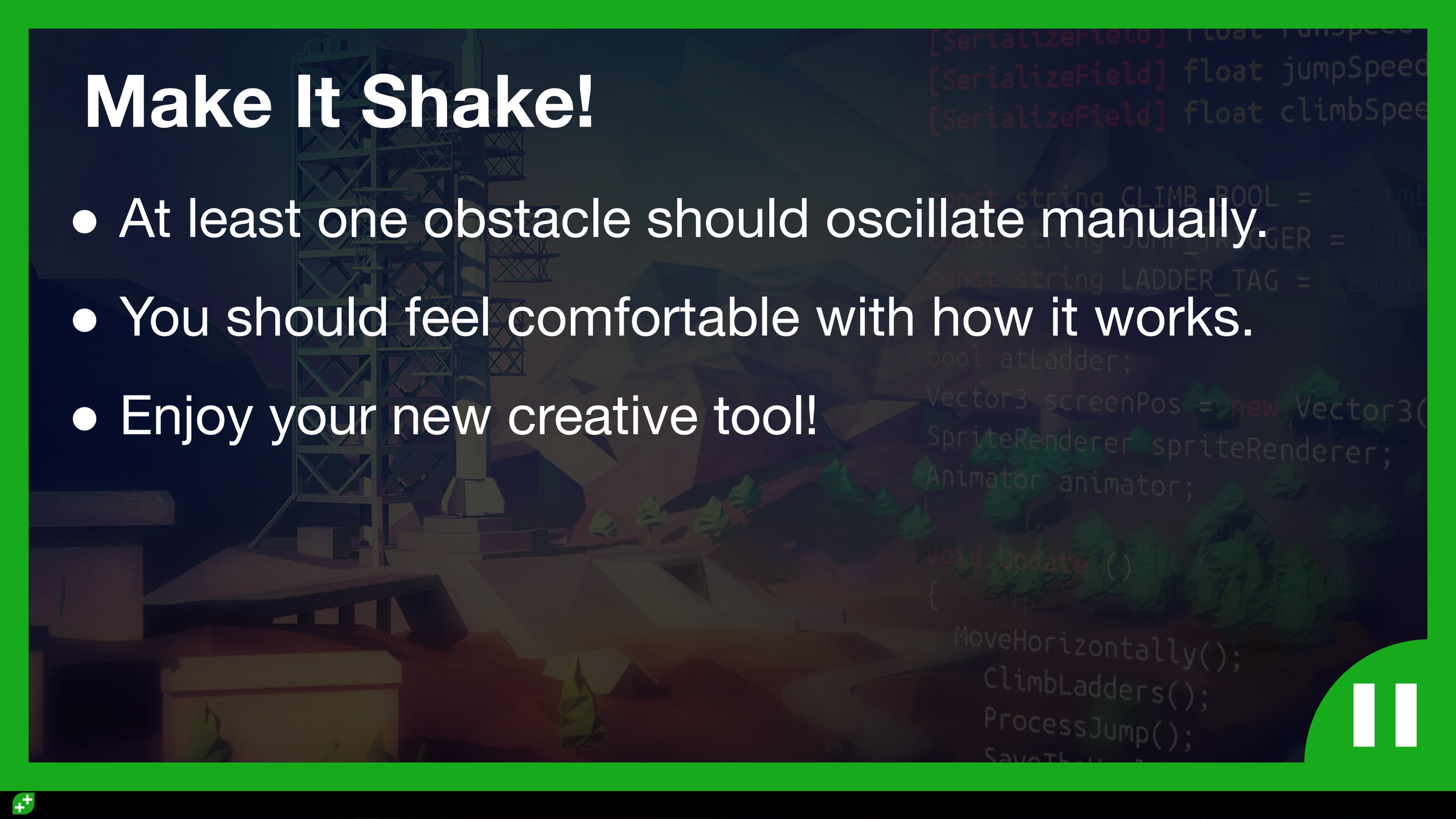
Moving Platform Pattern

Manually Moving Platforms

```
[SerializeField] Vector3 movementVector;
[Range(0, 1)] [SerializeField] float movementFactor;

Vector3 startingPos; // must be stored for absolute movement

void Start()
{
    startingPos = transform.position;
}

void Update()
{
    Vector3 offset = movementVector * movementFactor;
    transform.position = startingPos + offset;
}
```

Make It Shake!

- At least one obstacle should oscillate manually.
- You should feel comfortable with how it works.
- Enjoy your new creative tool!



GameDev.tv

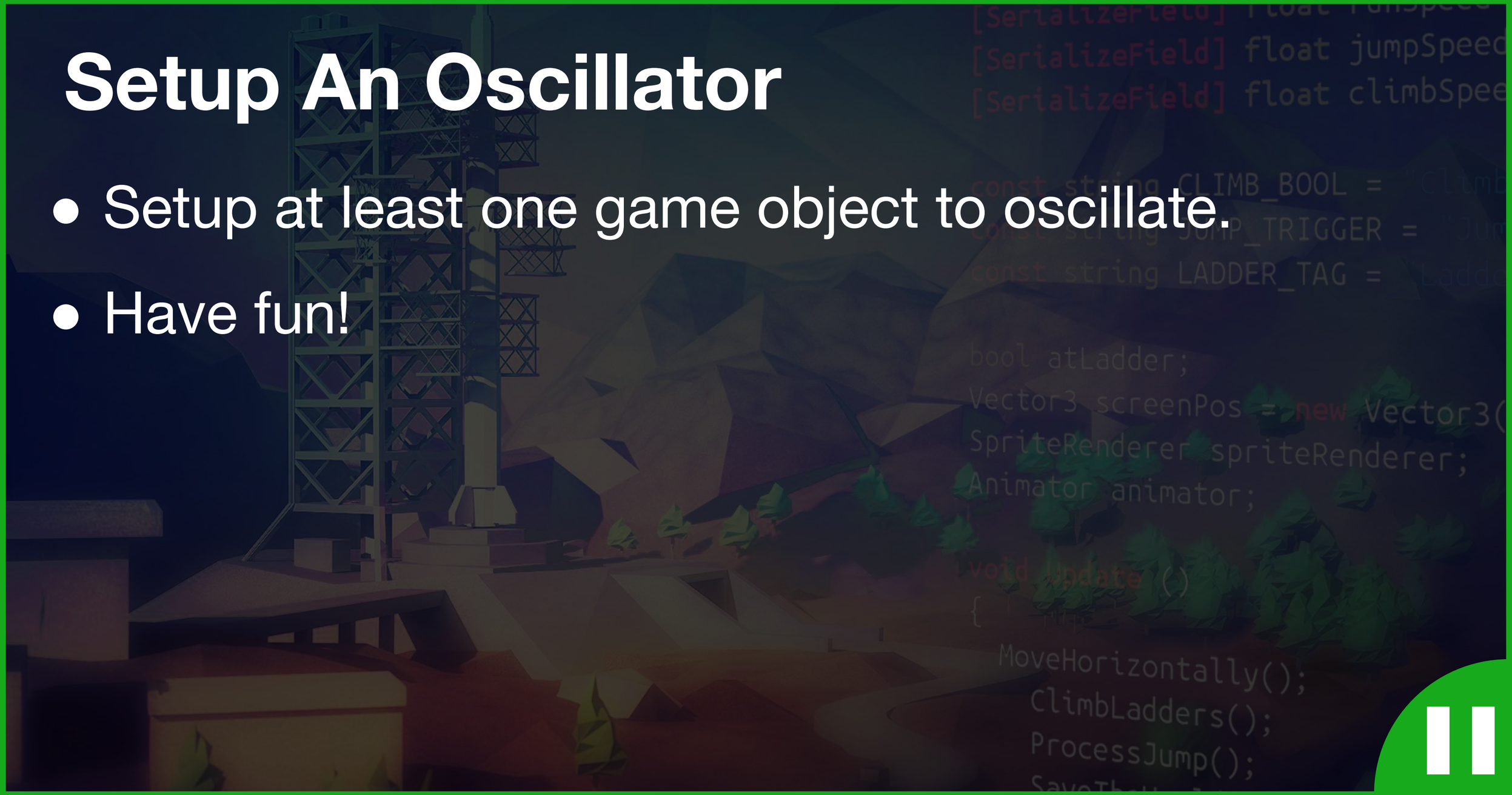
Unity 2020.1

Mathf.Sin() For Movement Cycles


amplitude (m)
period (s)

Setup An Oscillator

- Setup at least one game object to oscillate.
- Have fun!



GameDev.tv

Unity 2020.1

Protecting Against NaN

Try And Protect Zero Period

- Try and put some protection code in.
- Don’t worry if you get a compiler warning.
- Share a 2nd way you could do it in community.

Hint: Remember our Discord chat server!

Notes About Comparing `floats`

- Two `floats` can vary by a tiny amount.
- It's unpredictable to use `==` with `floats`.
- Always specify the acceptable difference.
- The smallest float is `Mathf.Epsilon`
- Always compare to this rather than zero.
- For example...

```
if (period <= Mathf.Epsilon) { return; }
```

GameDev.tv

Unity 2020.1

Organise Your Assets

Organising Your Assets

- Create folders.
- Use search by type.
- Create Favourites for Searches.
- Rearrange window layout / save layouts.

Organise Your Assets

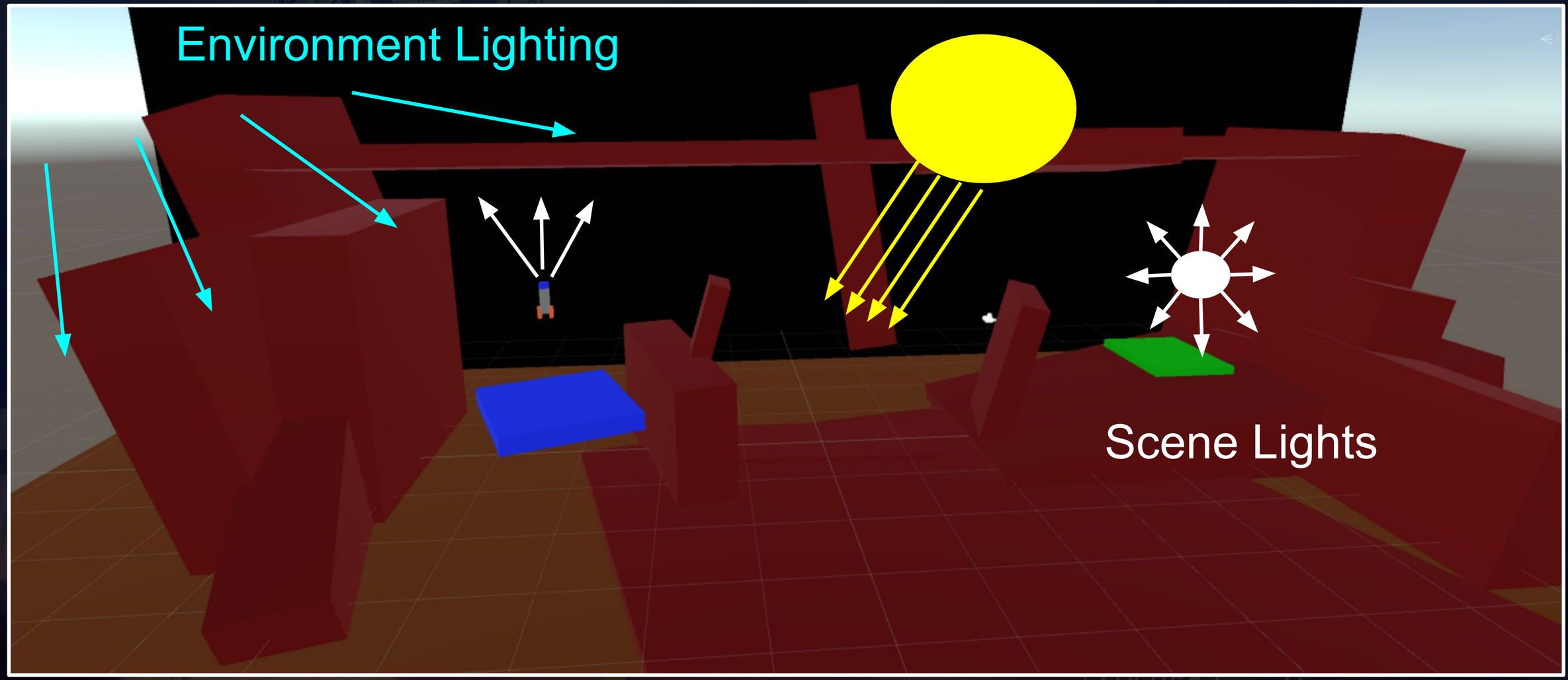
- Create a folder structure that works for you.
- Tidy up your scene assets.
- Make sure your prefabs are correctly linked.

GameDev.tv

Unity 2020.1

Light Your Scene

Main Directional Light (Sun)



Add Scene Lighting

- Alter your scene’s main directional light to make your scene feel different.
- Add at least one scene light.
- Share a screenshot.

GameDev.tv

Unity 2020.1

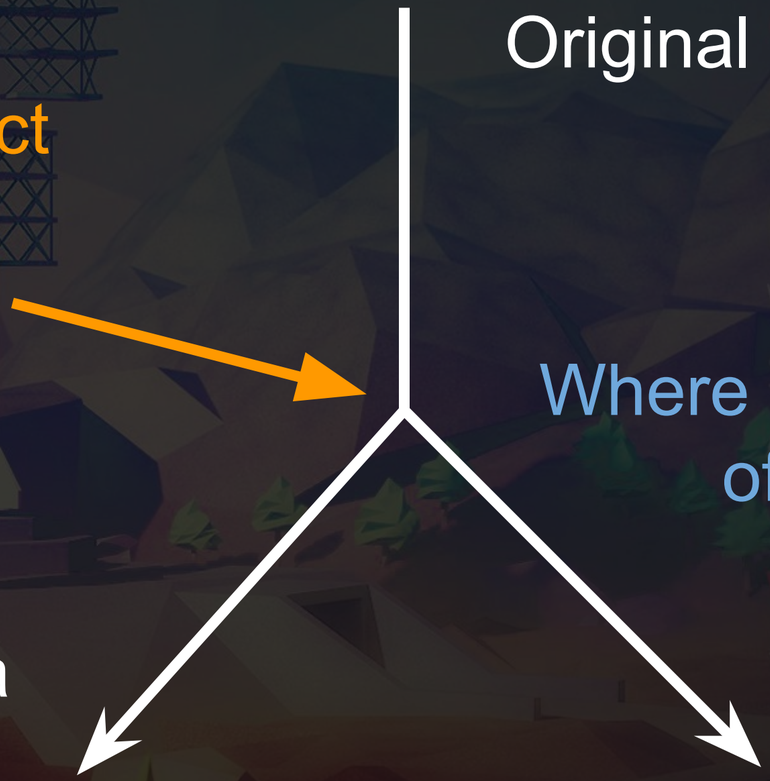
Nested Prefab Joy

Figure Out The Nested Prefab Rules

- A different sort of challenge!
- Using our Rocket Ship as the parent, figure out the relationship / dependency between the child Success Particle Effect and the Success Particle Effect prefab.

Childing A Prefab To A Prefab

The moment we child Particle Effect to Rocket



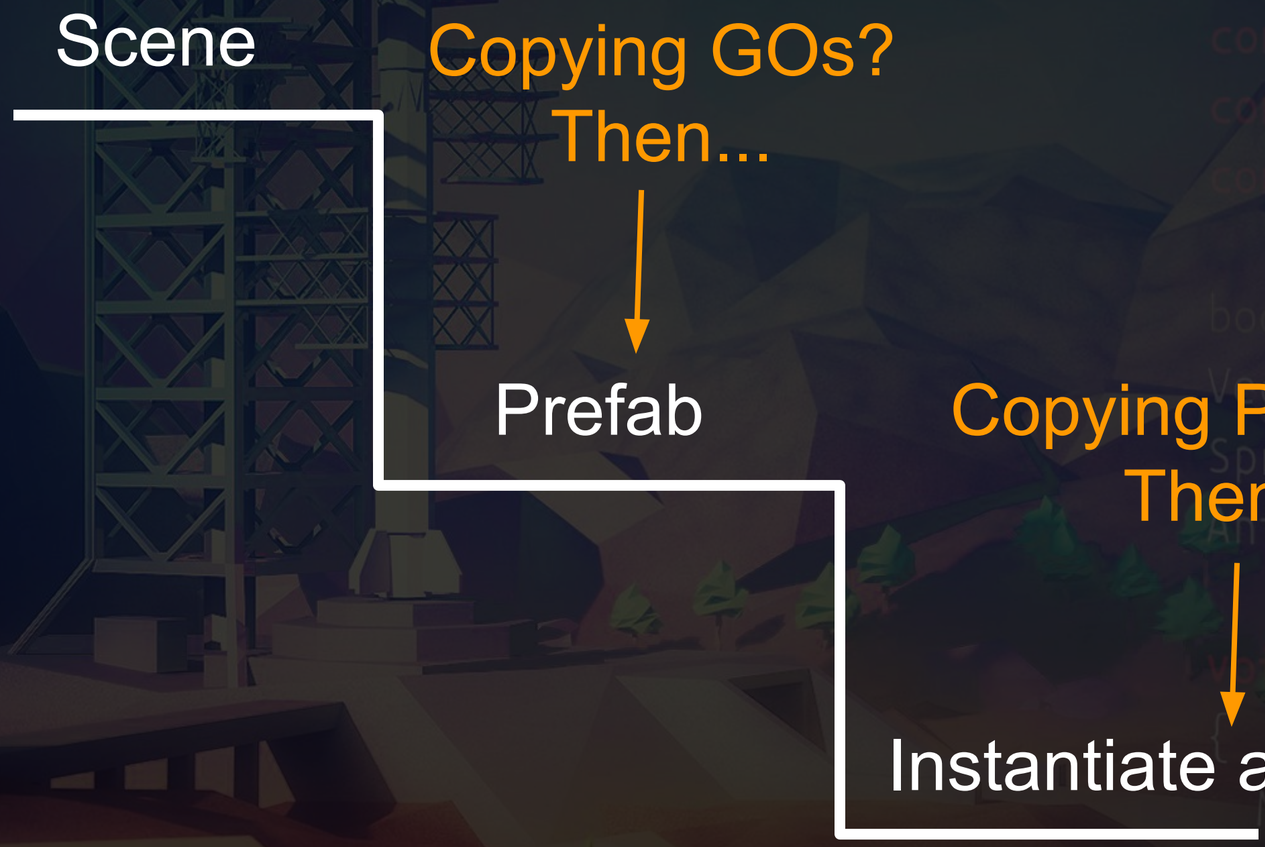
Original Particle Effect

Where is the “Source of Truth”?

Particle Effect Data on Rocket

Particle Effect Data on Prefab

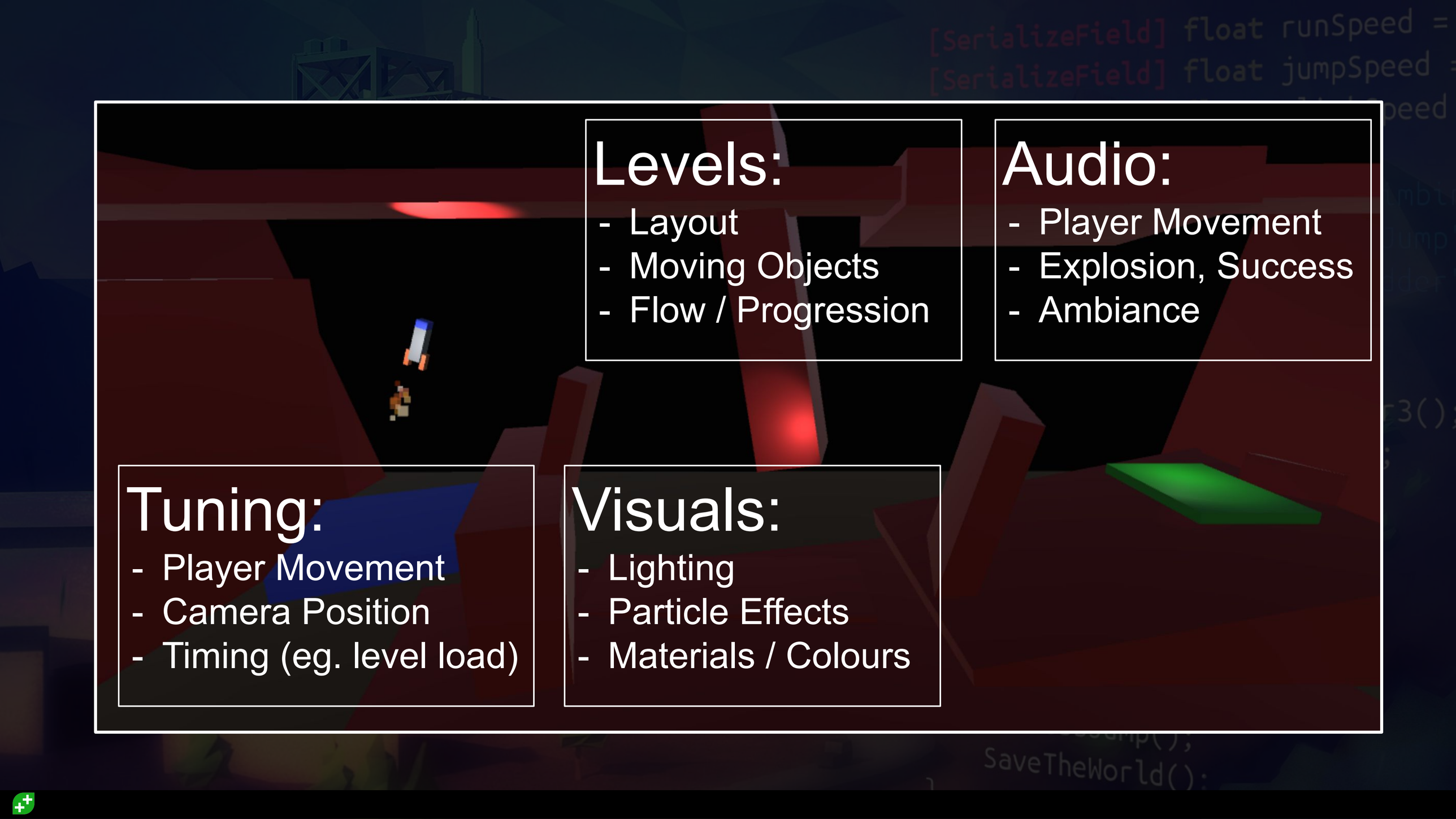
Where Is The Data?



GameDev.tv

Unity 2020.1

Make Game Moments



Levels:
- Layout
- Moving Objects
- Flow / Progression

Audio:
- Player Movement
- Explosion, Success
- Ambiance

Tuning:
- Player Movement
- Camera Position
- Timing (eg. level load)

Visuals:
- Lighting
- Particle Effects
- Materials / Colours

Level Flow And Variety

- We need to keep our players engaged for as long as we can
- Two options with our current game:
  1. Randomised levels of similar challenge level
  2. Sequential levels of increasing challenge level

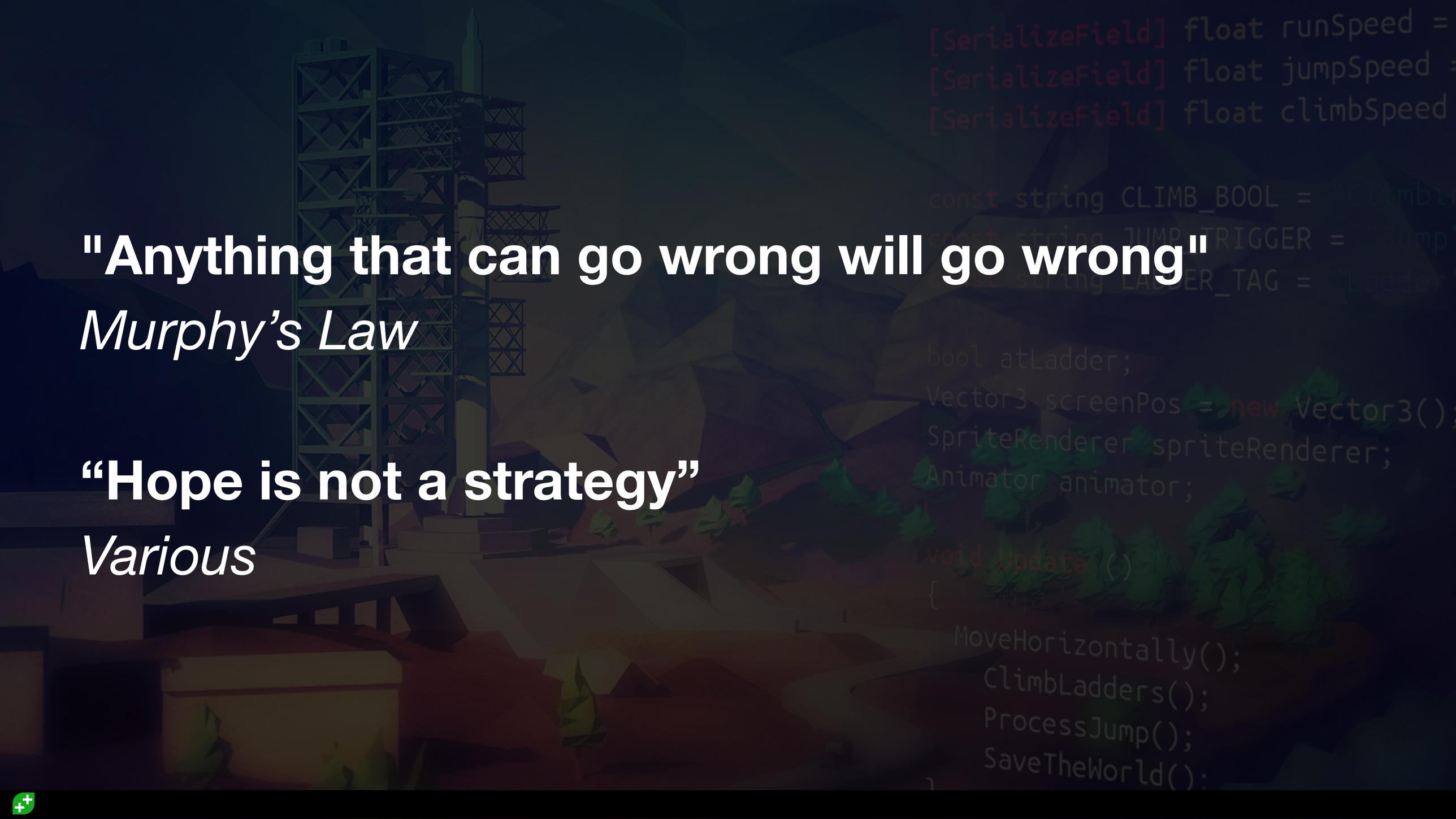
Share Your Best Game Moment

- Refine your Tuning, Visuals, Levels and Audio.
- Create at least 5 levels.
- Find what you think is the best 10-15 second moment.
- Capture video of that moment.
- Share on our Facebook group or Forum.

GameDev.tv

Unity 2020.1

Debug Keys & Builds



"Anything that can go wrong will go wrong"

Murphy's Law

"Hope is not a strategy"

Various

The L Key Advances Level

- Pressing L at any time should immediately load the next level
- Bonus: Pressing the C key toggles collision detection on and off

Note: This only needs to work for one level change for now.

GameDev.tv

Unity 2020.1

Instructor Hangout 3.3

In This Hangout...

- When will we teach mobile inputs? (Ken)
- Important to be good at math? (Adam & Cam)
- One script versus many scripts?
- What to do next with the project?

GameDev.tv

Unity 2020.1

Looping Through Levels

A Temporary Limitation

1. New rocket created on each level load
2. Any member variables are reset
3. So can’t store number of levels won on rocket
4. Simply use a conditional for last level.

We may challenge you to come back and fix this.

Get The Levels Cycling

- Your game should loop around all levels in the current build order.
- When you get to the last level return to the first.

Hint 0: If experienced use % but no need

GameDev.tv

Unity 2020.1

Sharing With Teaser Video

Marketing Your Game 101

- Put your best foot forward
- Make it easy for people
- Ask a question to get them thinking
- Think about what’s in it for them (fun)
- Be prepared for honest feedback.

Build & Share With 20s Video

- There should be a share on our forum showcase
- OR on our FaceBook group
- For live feedback try our Discord chat server
- OR at least with a friend or family
- There should be <= 20s of gameplay footage
- There is a clear link to an online version
- You are asking one simple question.

GameDev.tv

Unity 2020.1

Spit & Polish

In This Video...

- Some level and difficulty adjustments.
- Various minor code clarity improvements.

GameDev.tv

Unity 2020.1

Section 3 Wrap-Up