Unity 2017.1.1

Welcome To Section 3

Project Boost

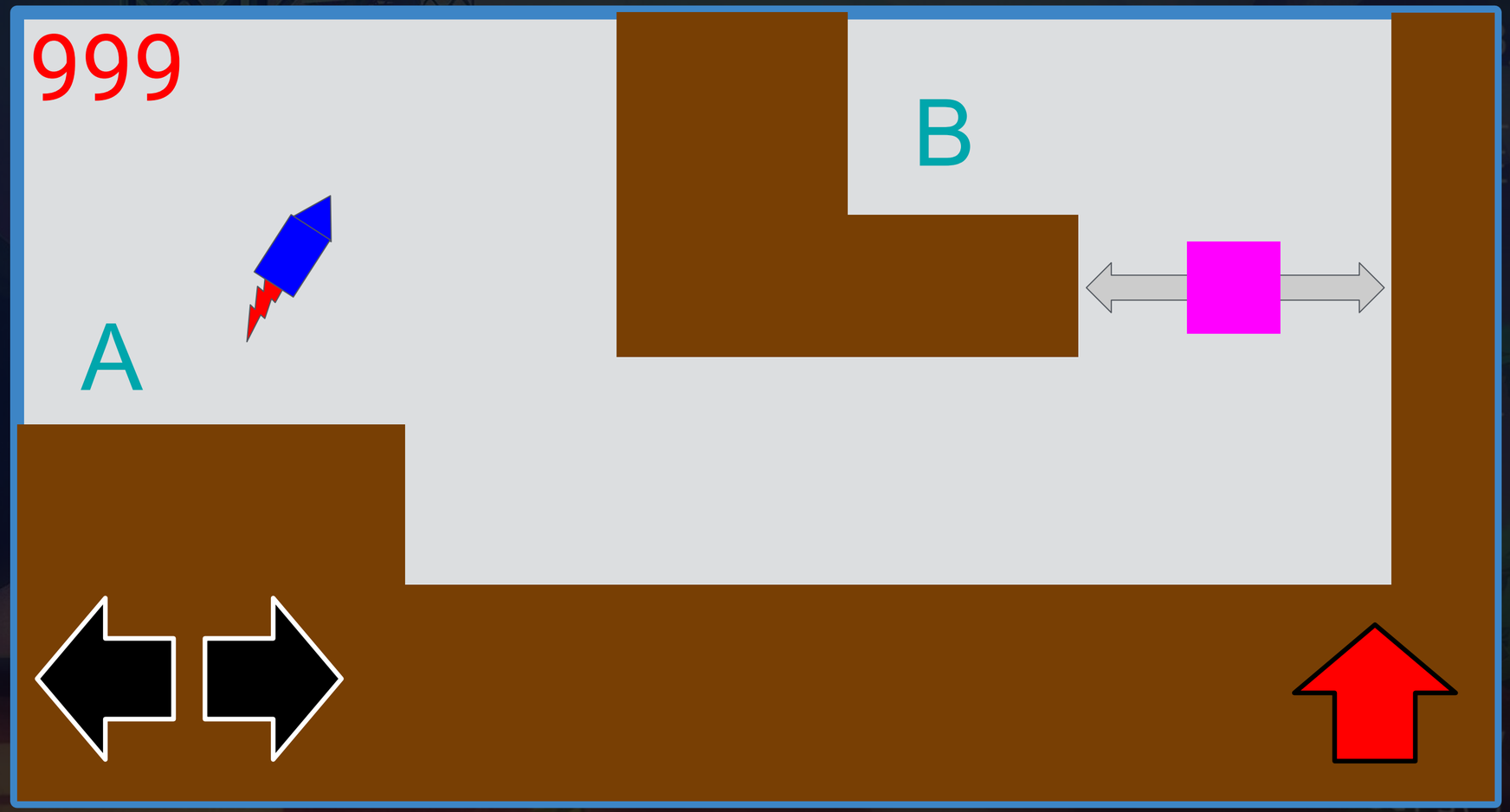
This is a 2.5D space piloting game inspired by the BBC Micro classic Thrust.

This is the third section of the Complete Unity Developer 2.0 course.

Unity 2017.1.1

Project Boost Game Design

Game Screen



Project Boost Game Design

Player Experience:

Skilled rocket pilot

Core Mechanic:

Skillfully fly spaceship and avoid environment without running out of fuel.

Core game loop:

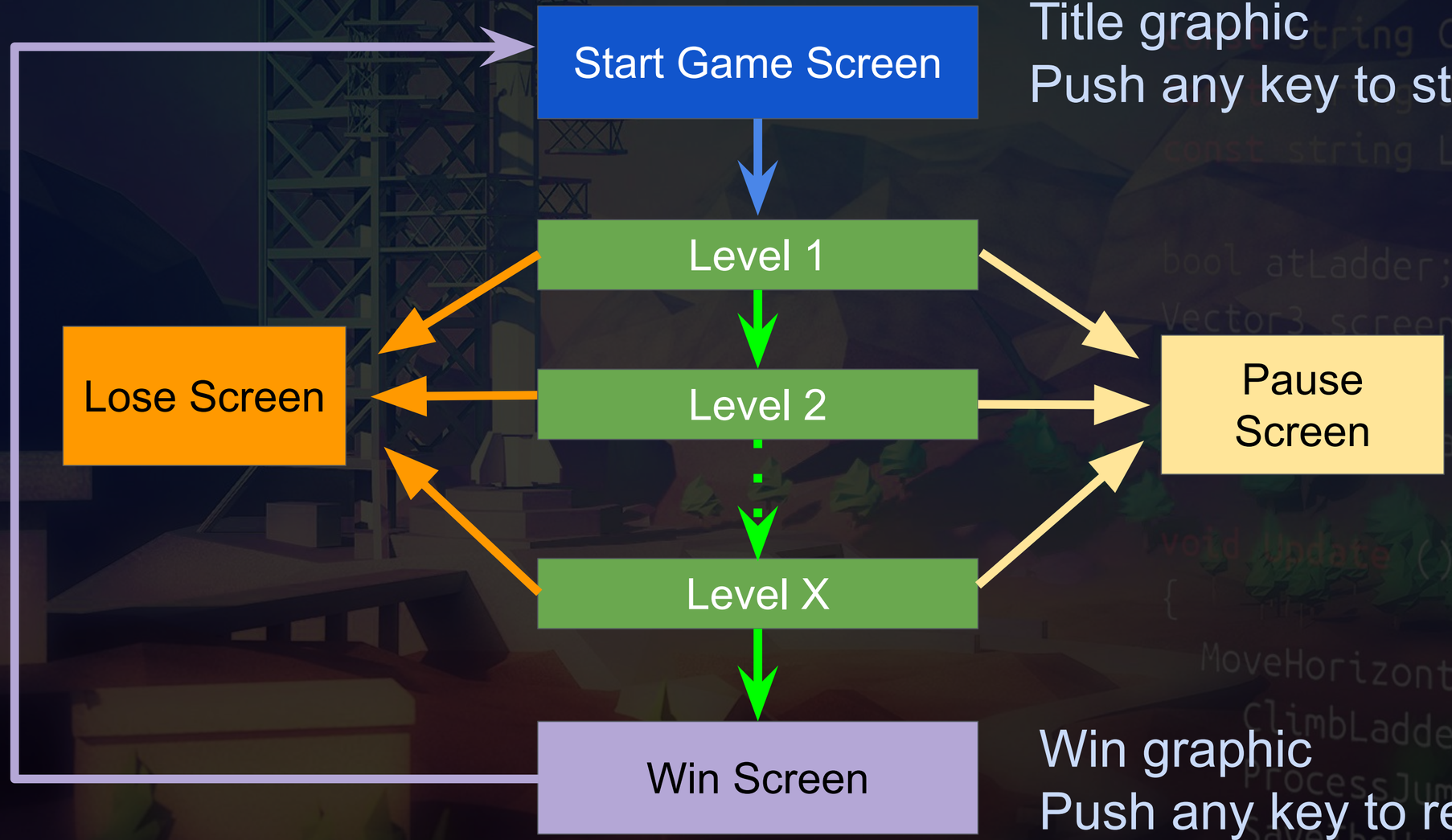
Get from A to B to complete the level, then progress to the next level.

Tech Specs

- 1920x1080 16:9 aspect ratio.
- Build to web, iOS & Android.
- Mono audio including SFX and background music.

hint

Game Flow And Screens



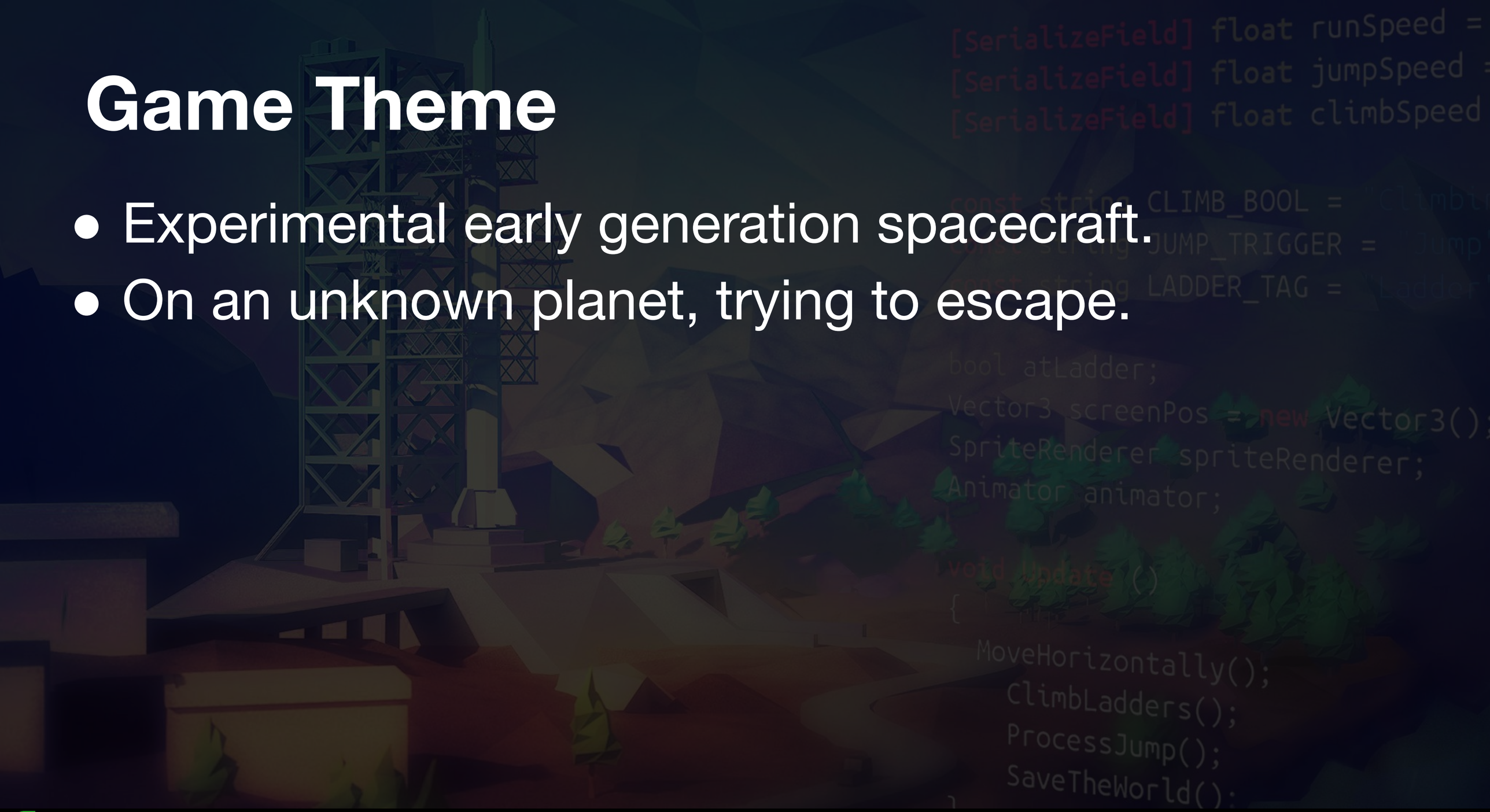
Title graphic
Push any key to start

Win graphic
Push any key to return to menu

Game Flow And Screens

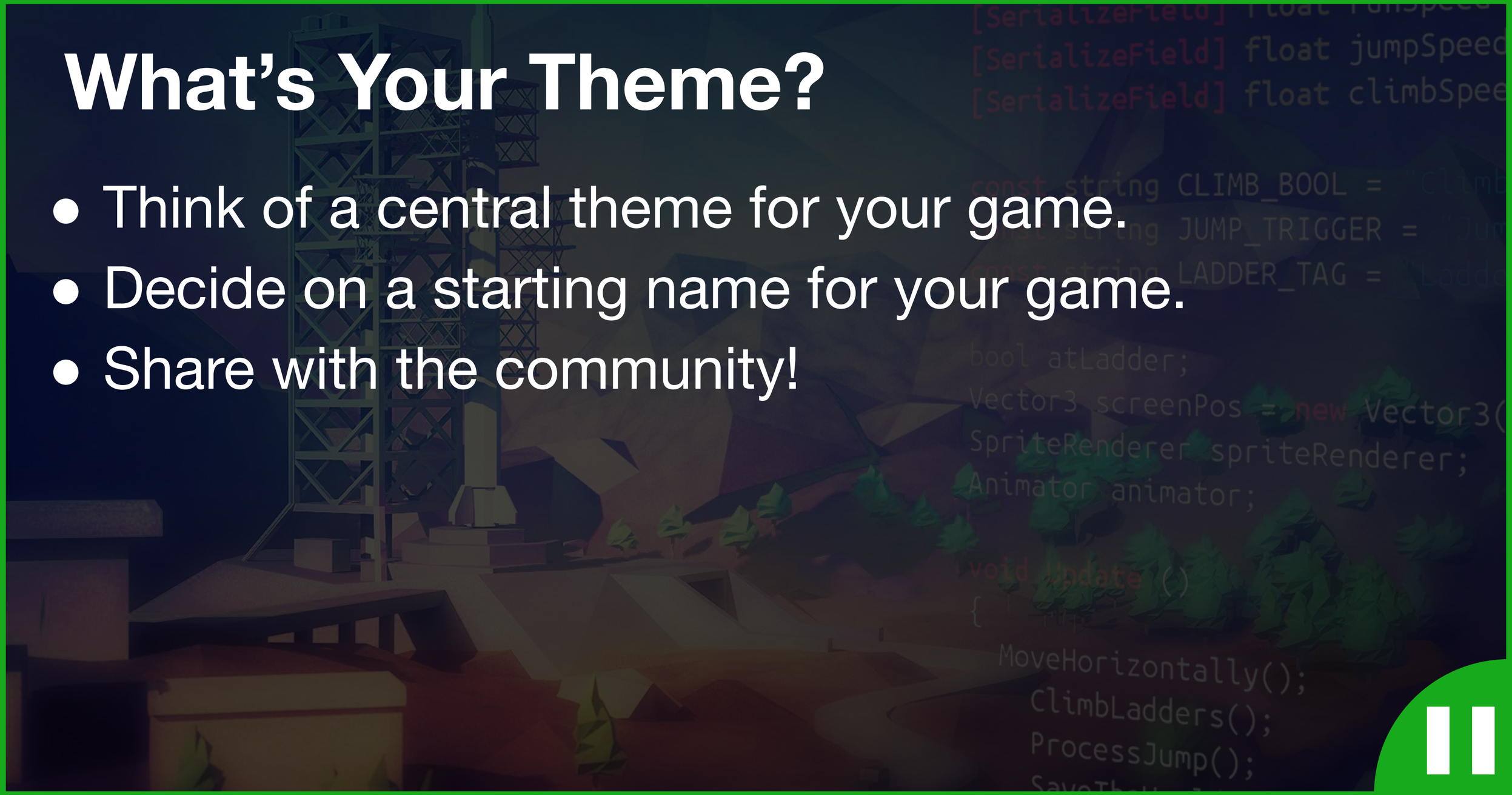
Game Theme

- 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!



Unity 2017.1.1

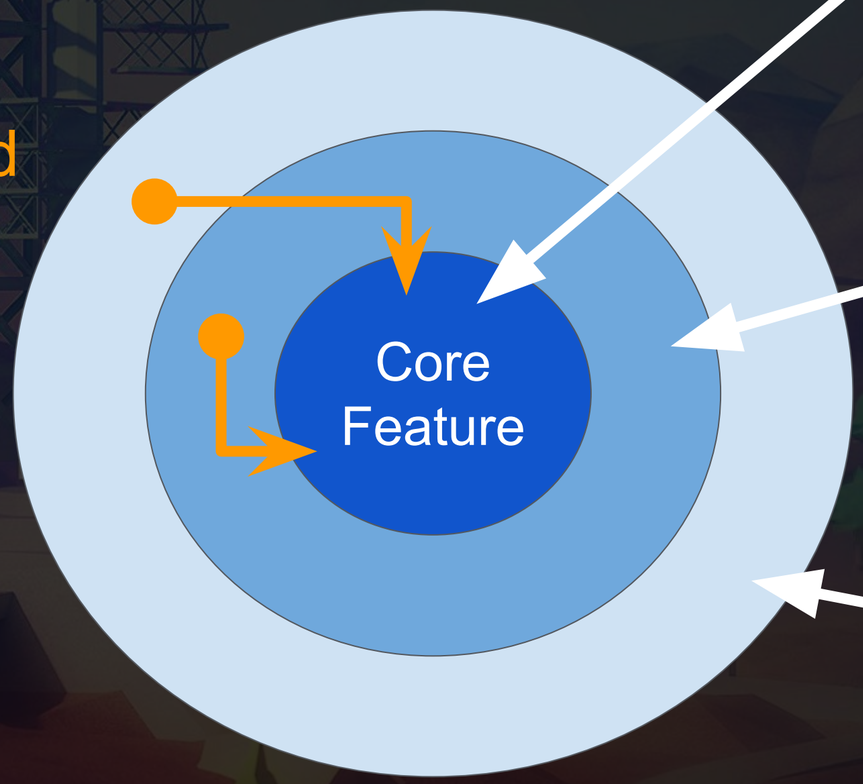
Onion Design

Common Design Challenges

- What features should I include?
- Where should I start?
- 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

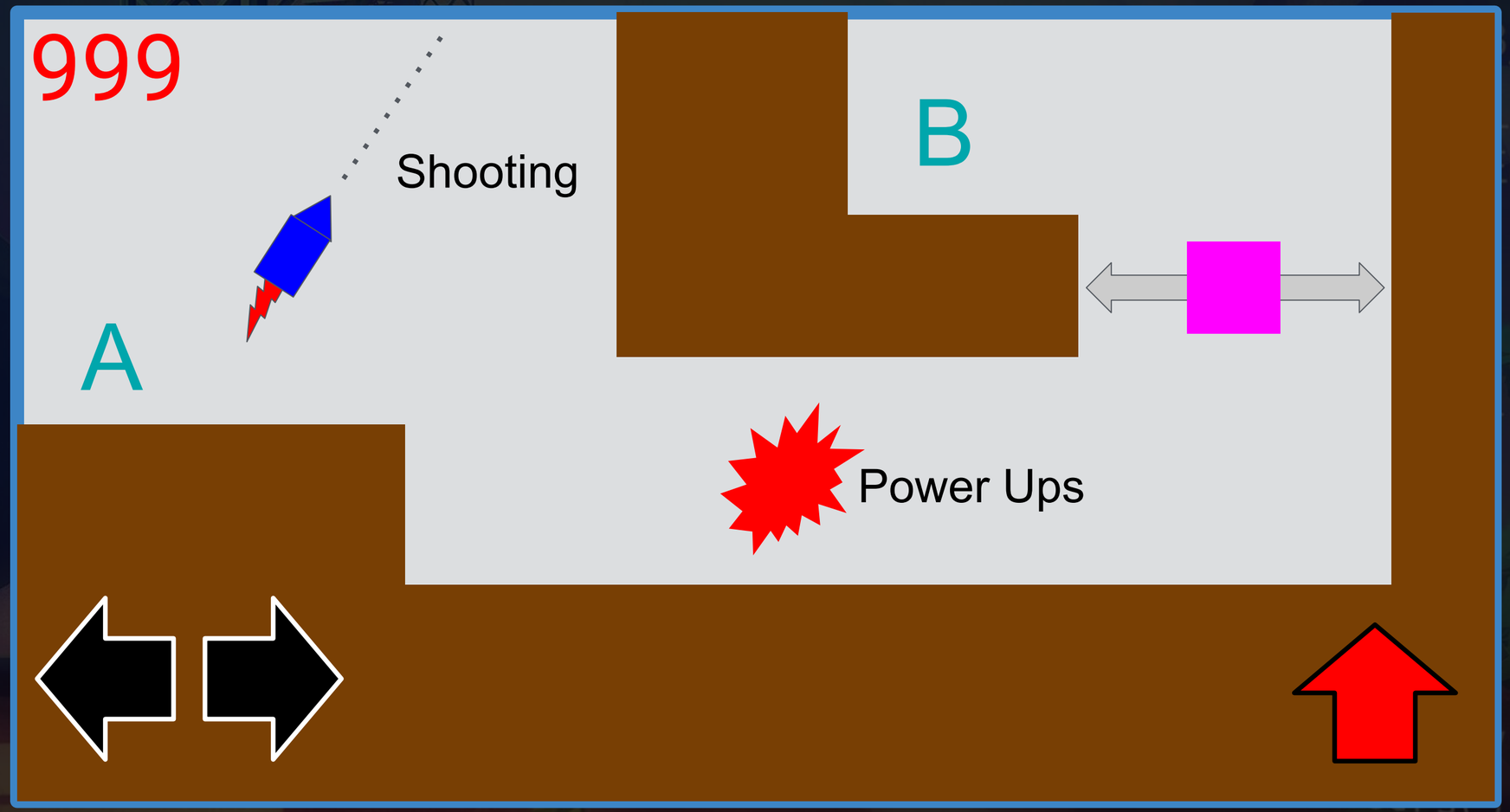


Most important feature

2nd most important feature

3rd most important feature

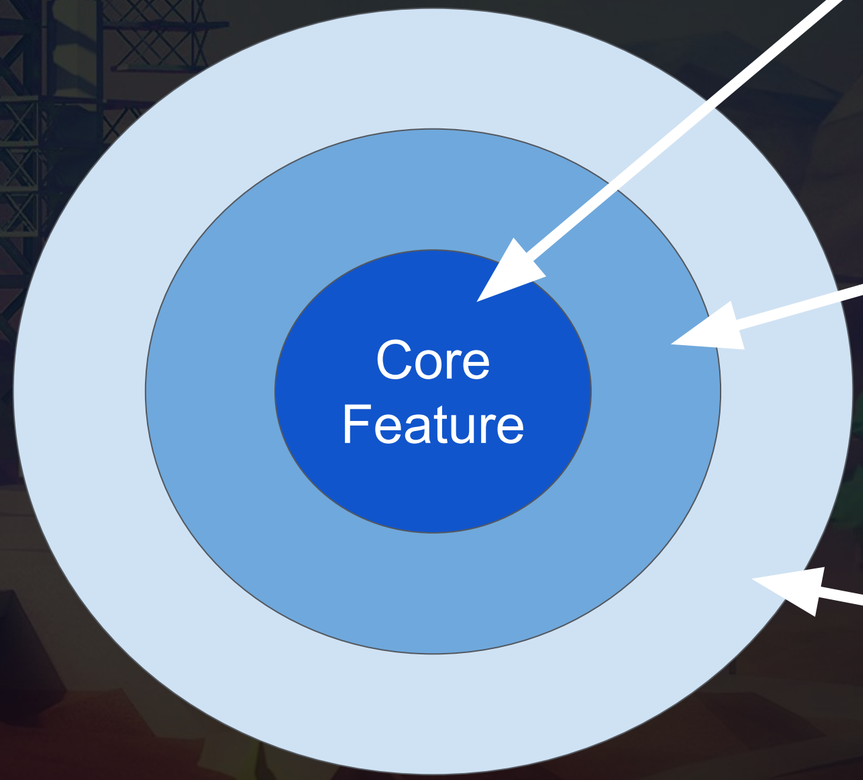
Game Screen


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. Fuel mechanic

Unity 2017.1.1

Introducing Version Control

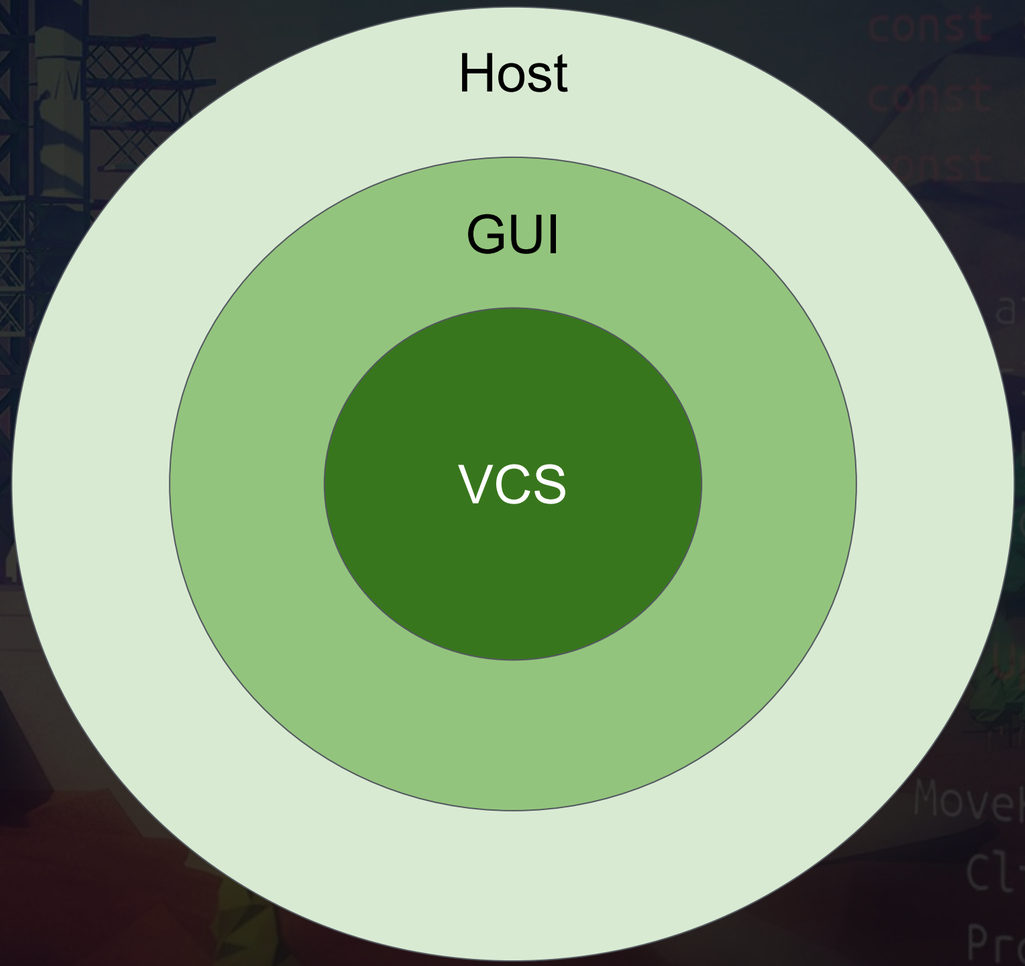
What > Why > Who > How

WHAT: Version Control System. Records history.

WHY: Backup, versioning, integrity.

WHO: Atlassian SourceTree, GitHub Desktop, GitKraken. OR go commando with the terminal in which case Stack Overflow!

Version Control Onion!



Some Version Control Jargon

- Git is a Version Control System (VCS)
- Your project folder contains a repository
- You commit your changes to your repo
- SourceTree is a Graphical User Interface (GUI)
- GitHub / BitBucket host repos remotely online

URICK: How To Setup Your VCS

0. **U**nity: Create Unity project as usual.
1. **R**epository: Create using SourceTree GUI.
2. **I**gnore: Add your `.gitignore` file.
3. **C**ommit: Your initial project state.
4. **K**eep: a useful history.

Unity 2017.1.1

Add Unity .gitignore Easily

An Easy Way To Setup .gitignore

Can be a pain because it’s hidden.

1. Ignore the Library folder from the GUI.
2. Open the `.gitignore` file it creates.
3. Paste in from a trusted source, e.g. GitHub.
4. Commit the `.gitignore` file.

Play With Version Control

- Do something else really simple in Unity.
- Save or close the editor.
- See what’s changed in version control.
- Make a commit with a useful name.
- Share a screenshot of your early history.
- Make a mess, have fun!

Unity 2017.1.1

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.

Unity 2017.1.1

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.

Unity 2017.1.1

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.

Unity 2017.1.1

Physics and Rigidbodies

Using `GetComponent<>()`

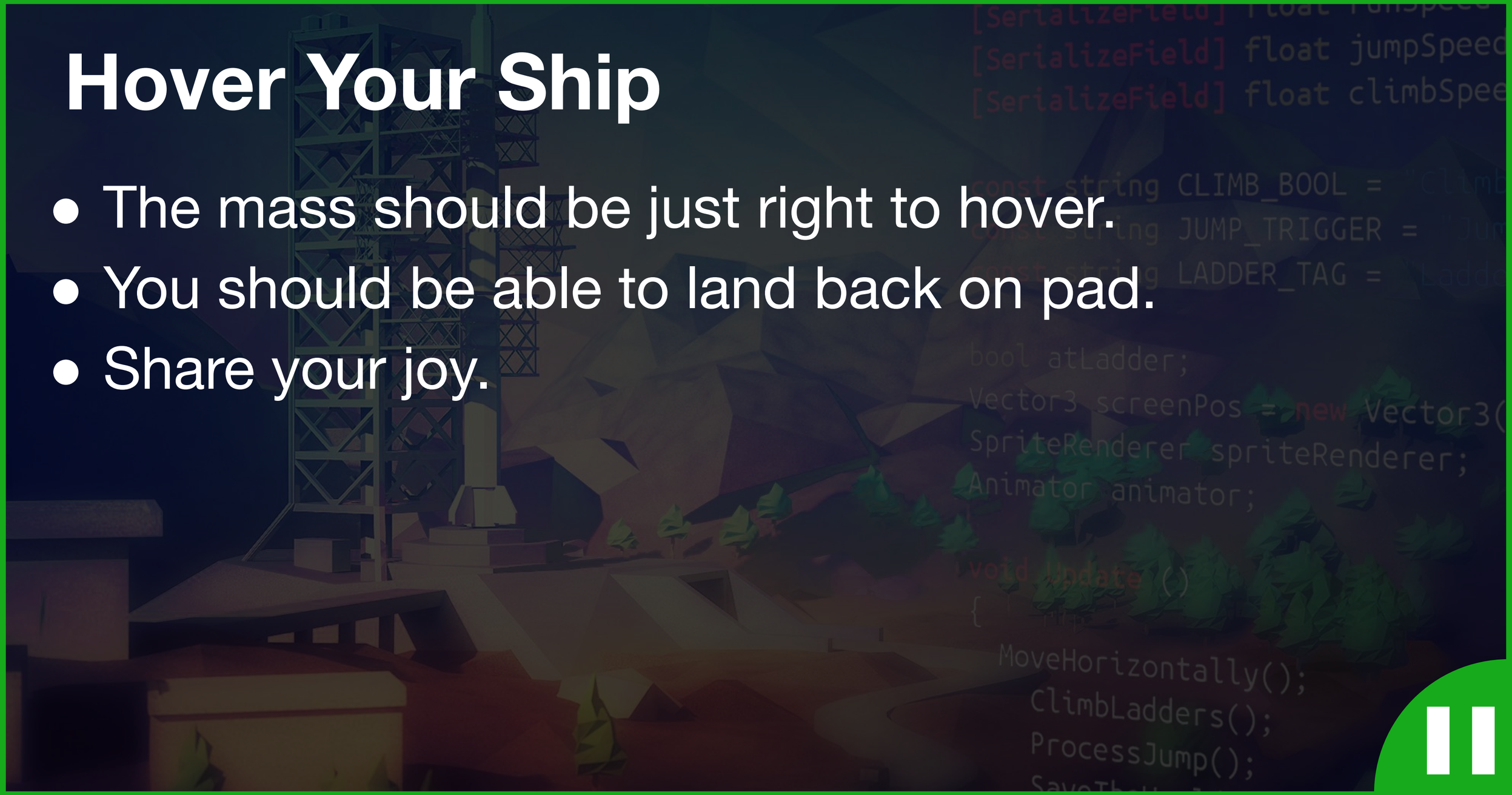
Use the following template to create a `rigidBody` member variable in your code, which allows you to access the rigid body on the same game object…

```
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.



Unity 2017.1.1

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.

Unity 2017.1.1

Using Time.deltaTime

Frame-rate Independence

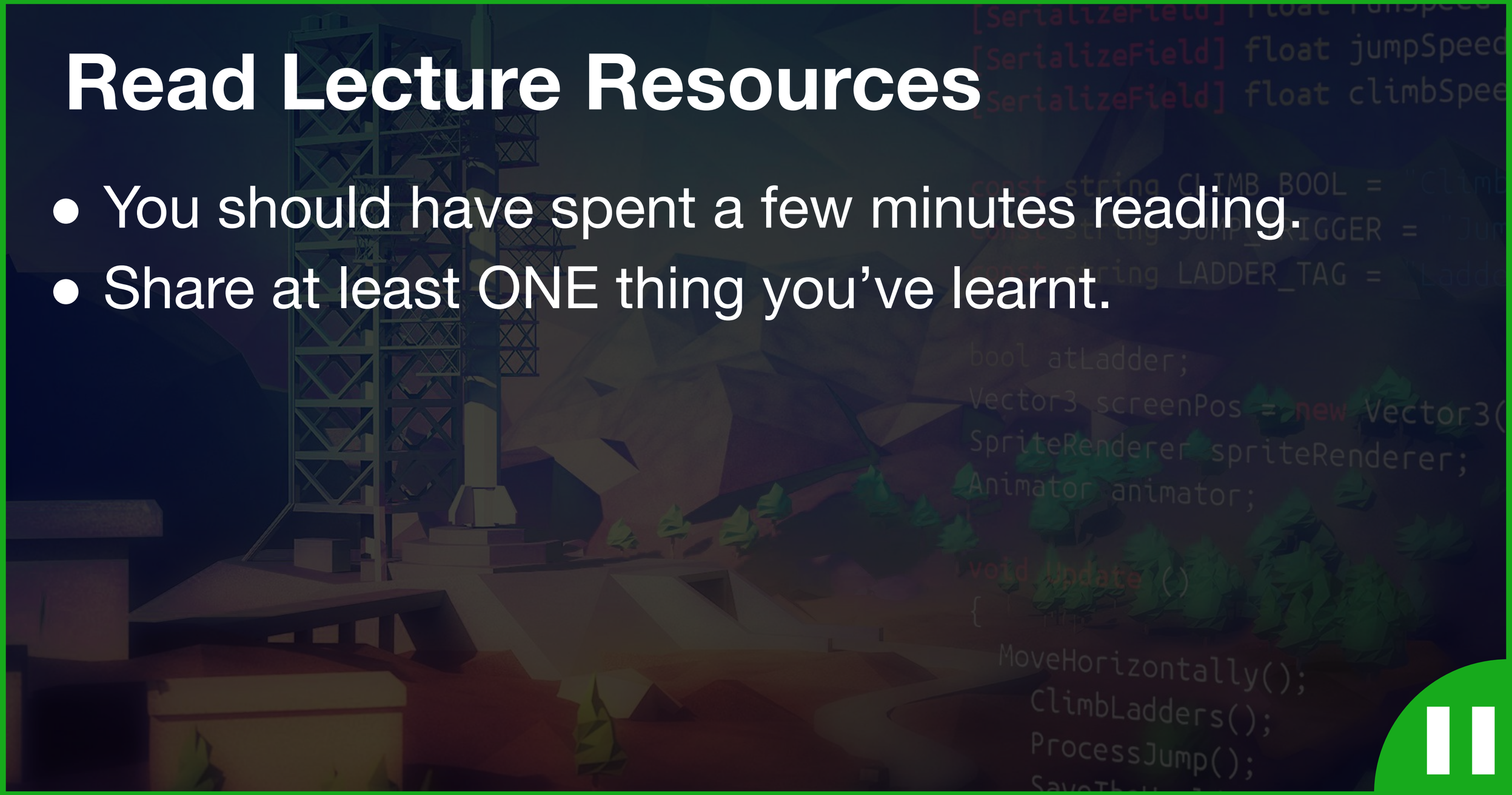
- The time each frame takes can vary wildly.
- `Time.deltaTime` tells you the last frame time.
- This is a good predictor of the current frame time.
- We can use this to adjust our movement.
- Longer frames lead to more movement.
- Shorter frames lead to less movement.
- e.g. `rotation = rcsThrust *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



Unity 2017.1.1

Instructor Hangout 3.1

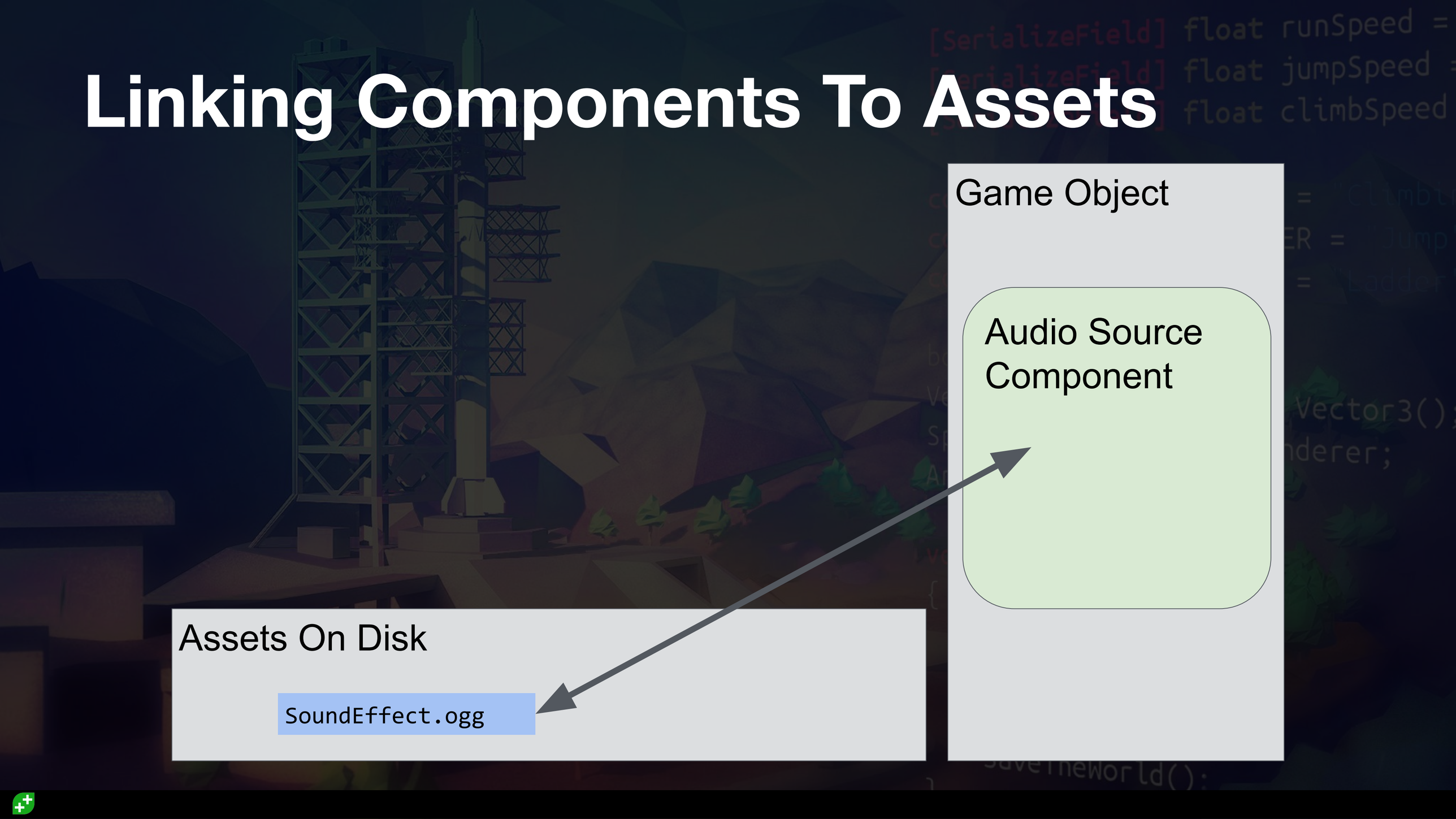
In This Hangout...

- 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!

Unity 2017.1.1

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?

Unity 2017.1.1

Resolving Movement Bugs

“Don’t patch over problems”

Your Movement Is Bug Free

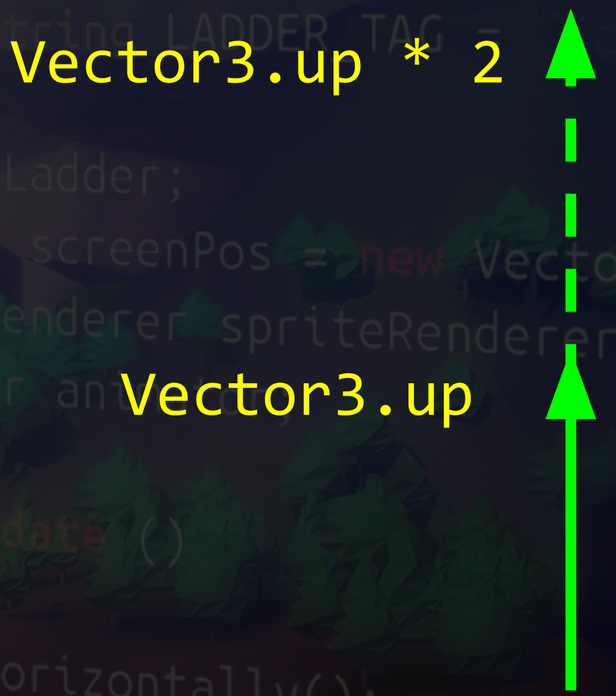
- You can move from one platform to another.
- A platform can induce spin but it stops on thrust.
- You have Drag value you’re happy with.
- Your code is beautiful.
- Share a < 20s video or GIF of your gameplay.

Unity 2017.1.1

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!

Unity 2017.1.1

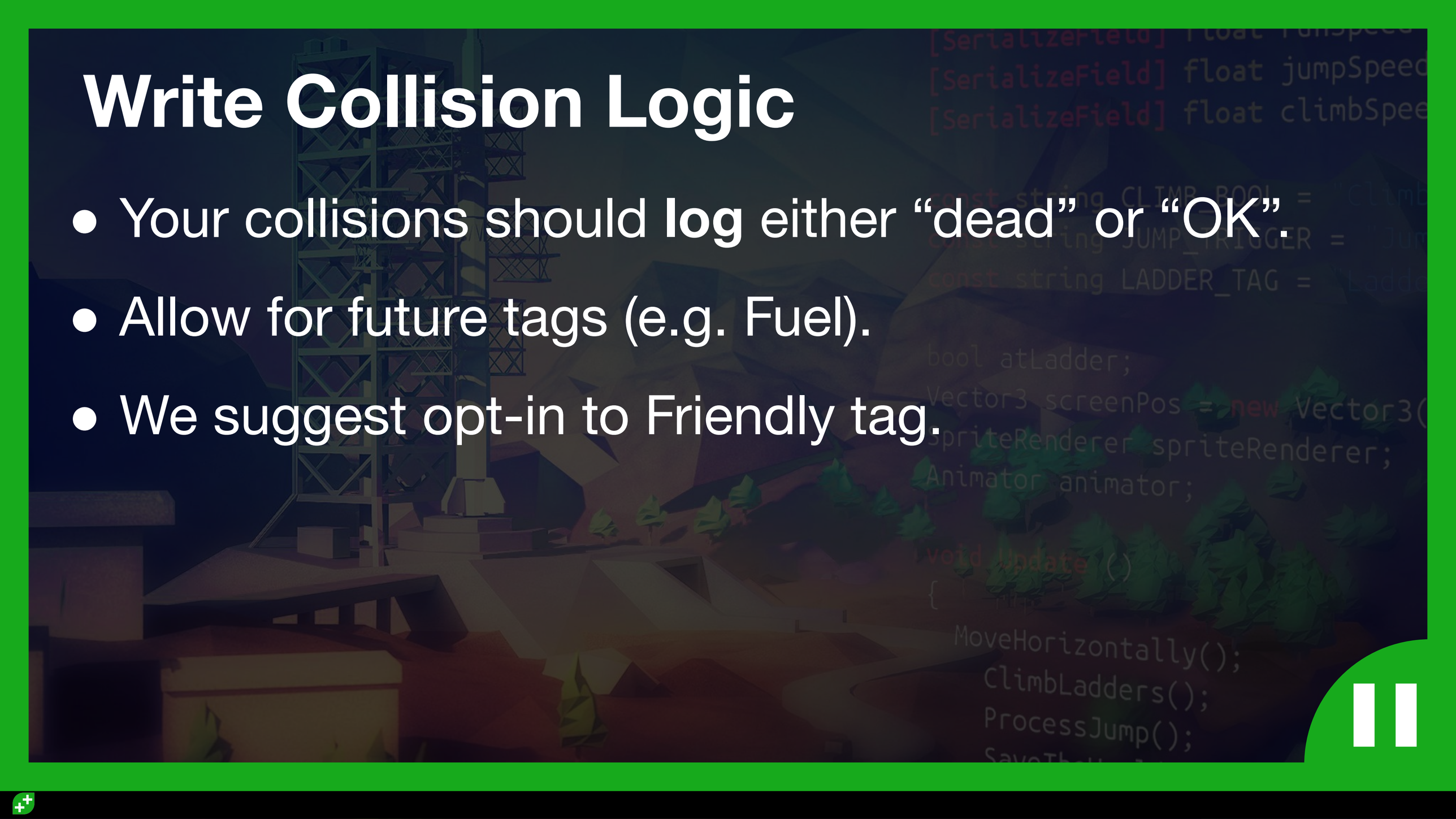
Tagging Game Objects As Friendly

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.



Unity 2017.1.1

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.

Unity 2017.1.1

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.

Unity 2017.1.1

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

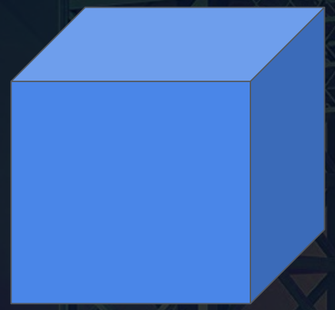
Create A Second Level

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

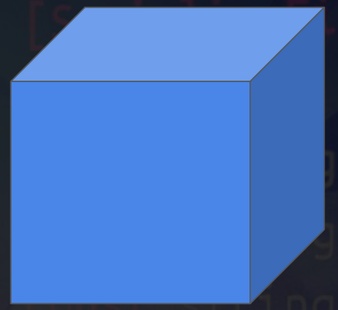
Unity 2017.1.1

Prefabs In Detail

Create Object



Prefab & Save Scene



Position & Rotation

All other details

`.unity` scene file

Position & Rotation

All other details

New .prefab file

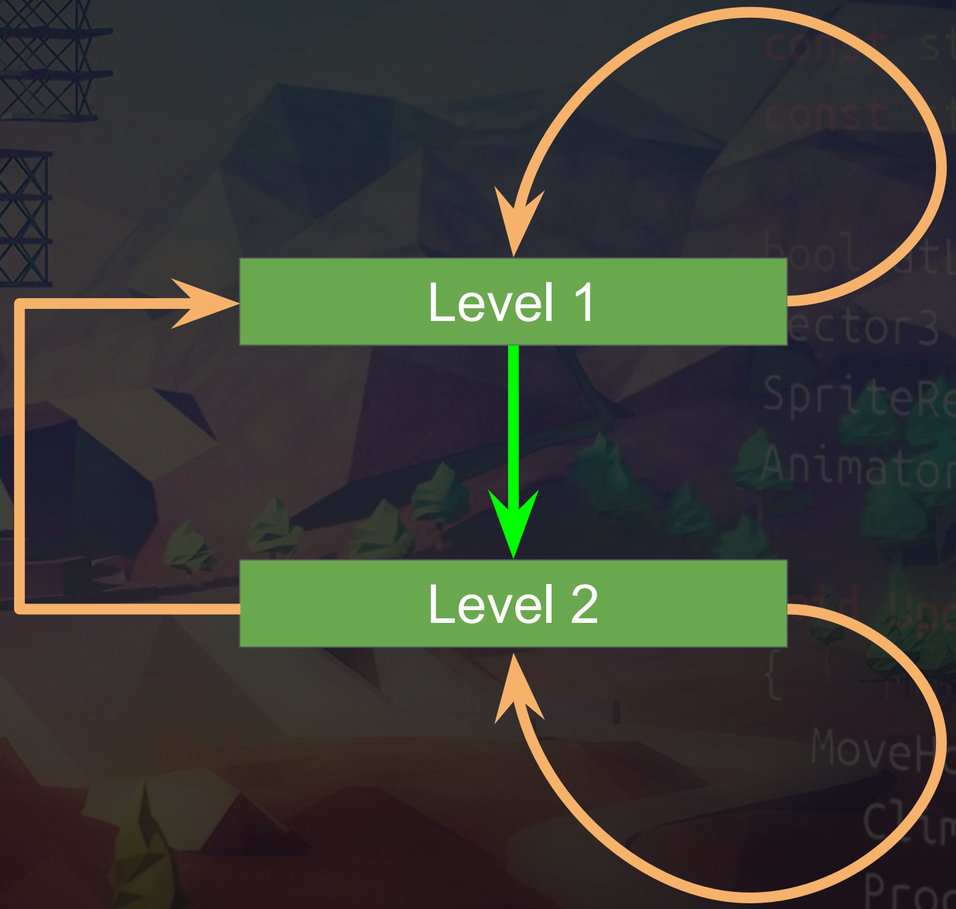
Explore Prefabs

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

Unity 2017.1.1

Level Loading & Scene Management

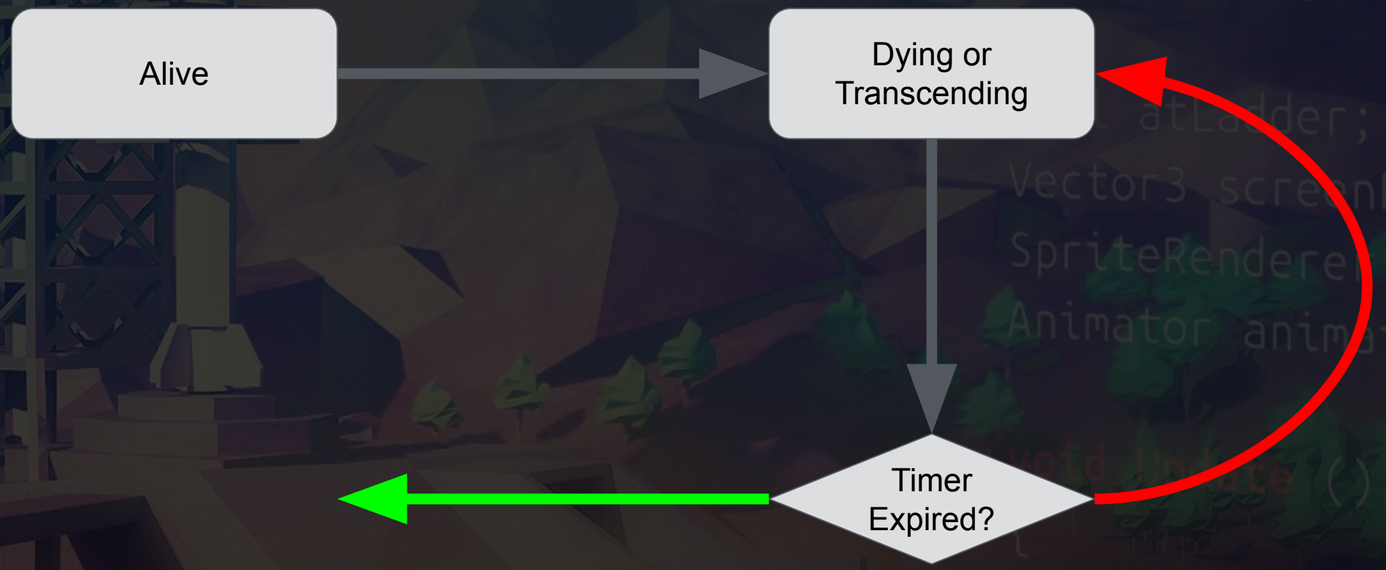
Your Game Cycles Through 2 Levels



Unity 2017.1.1

`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.

Unity 2017.1.1

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!

Unity 2017.1.1

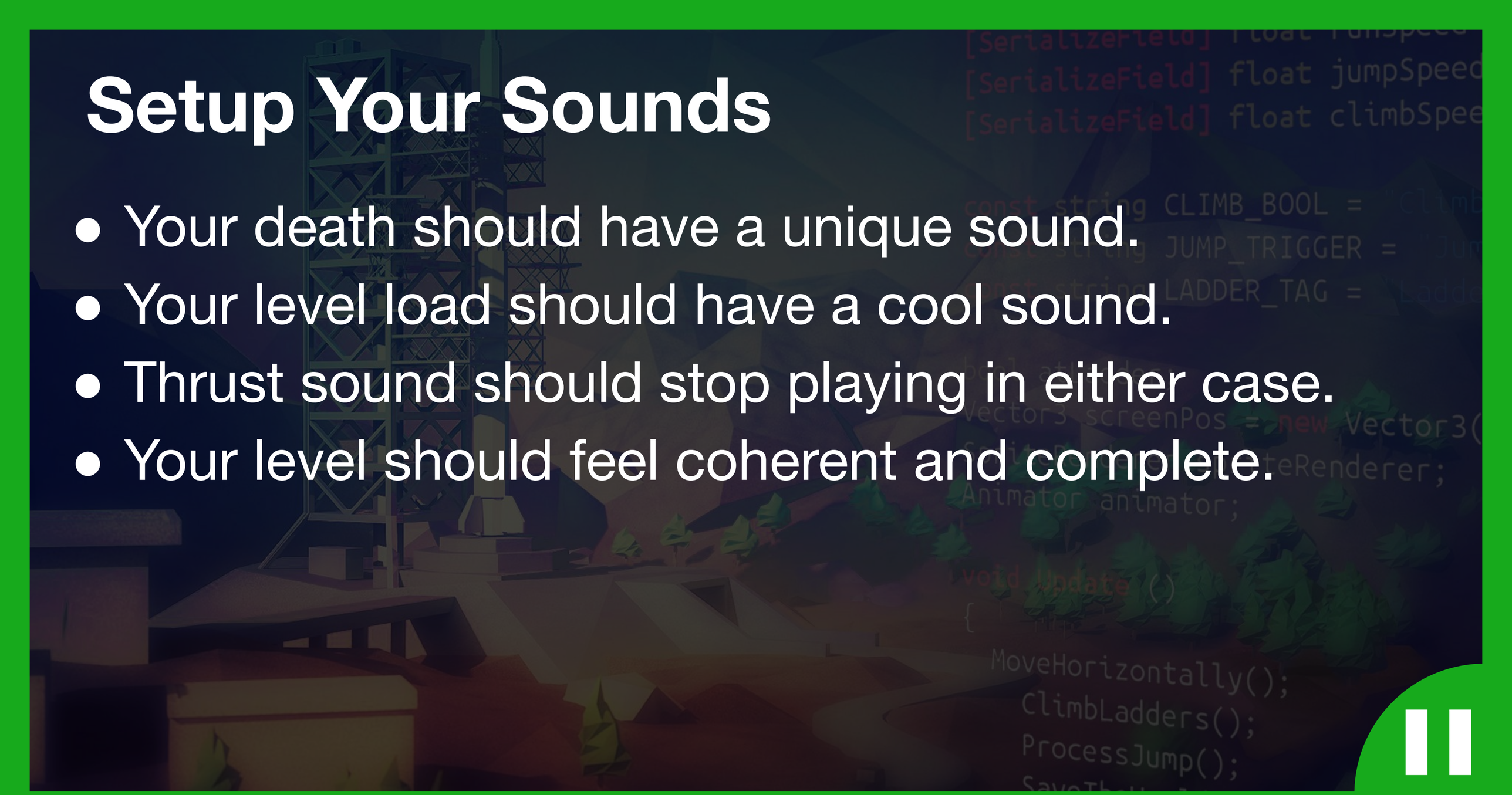
Playing Multiple Audio Clips

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.



Unity 2017.1.1

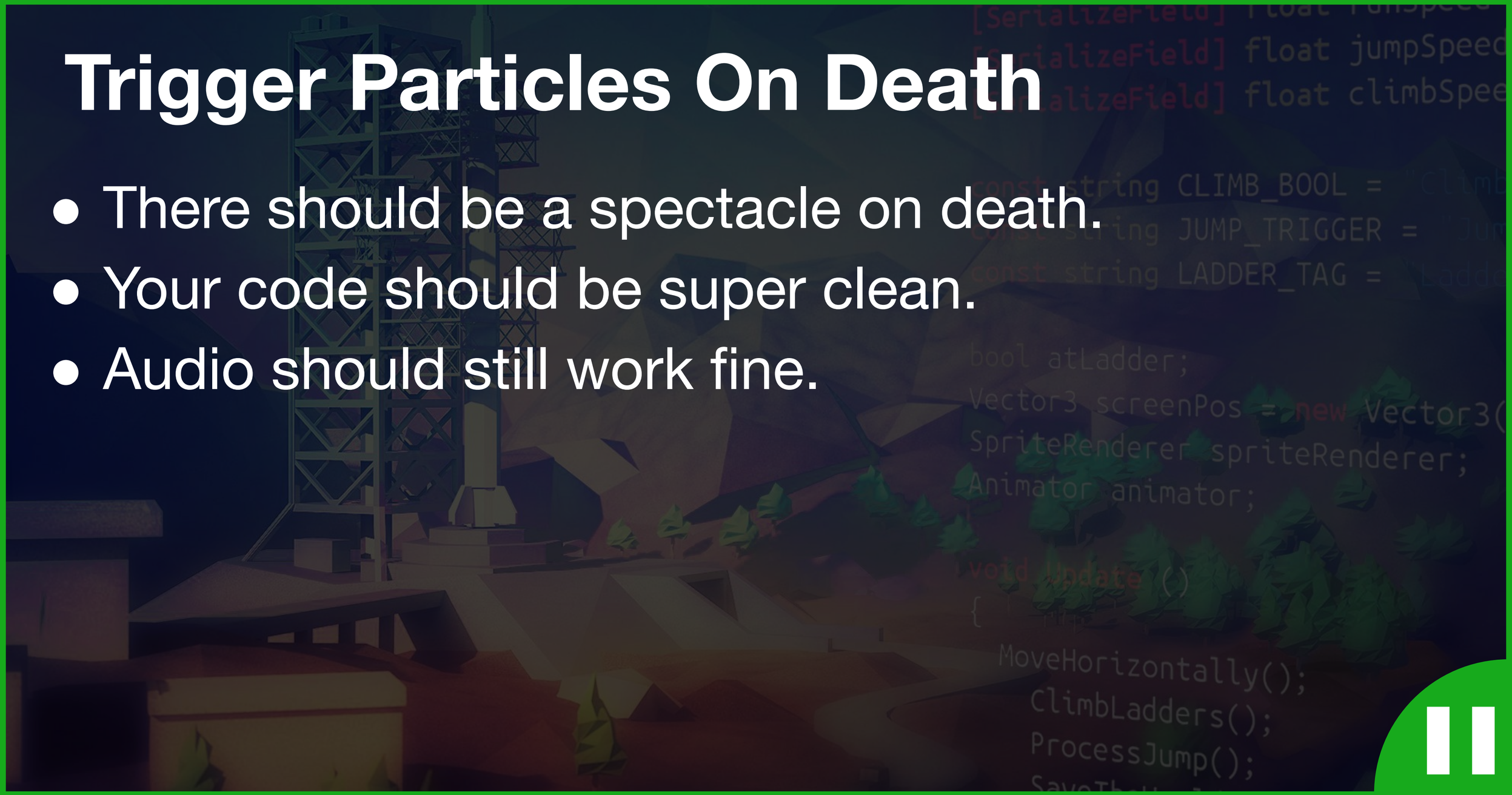
Introducing Particle Effects

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.



Unity 2017.1.1

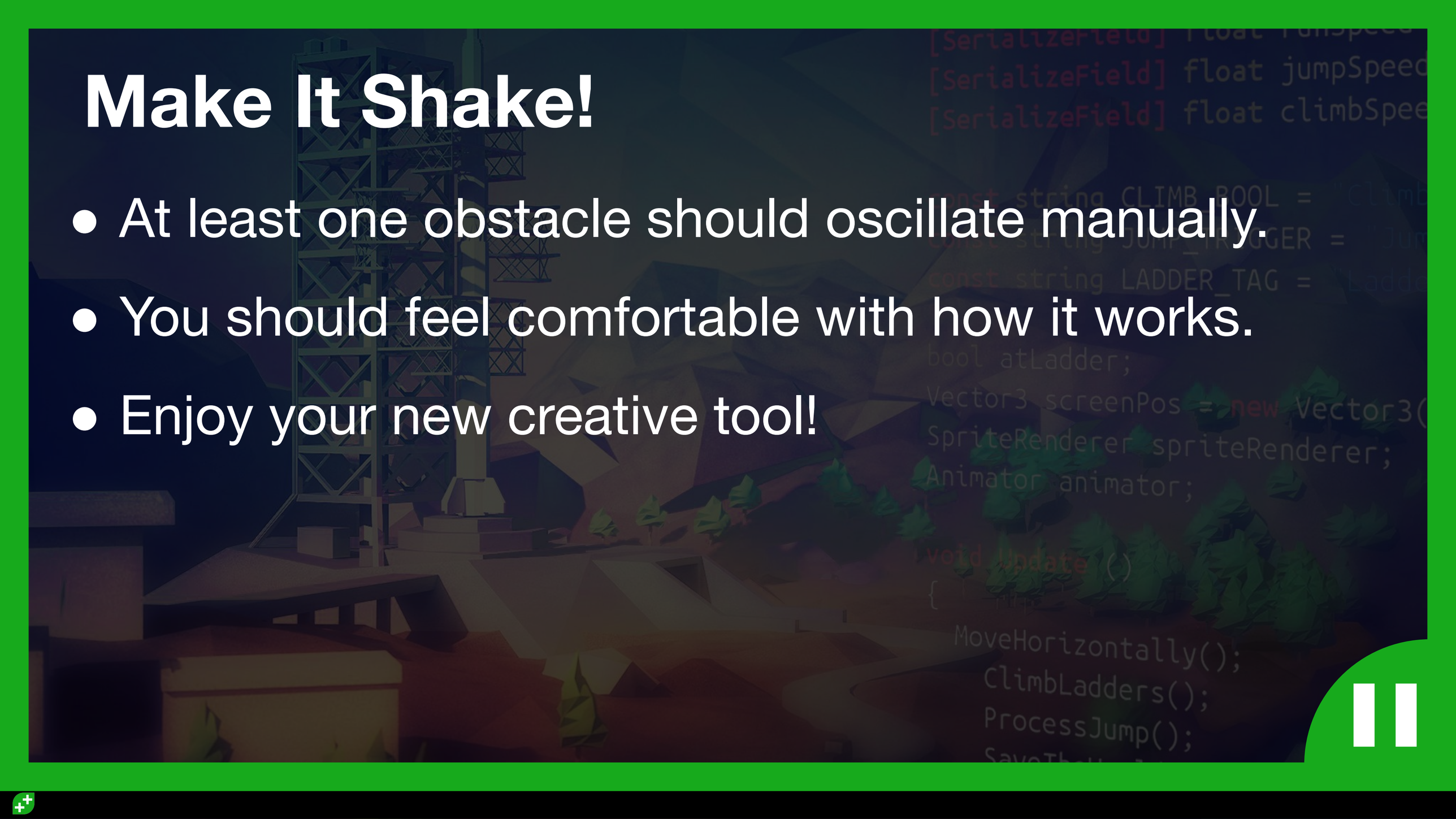
Moving Platform Pattern

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!



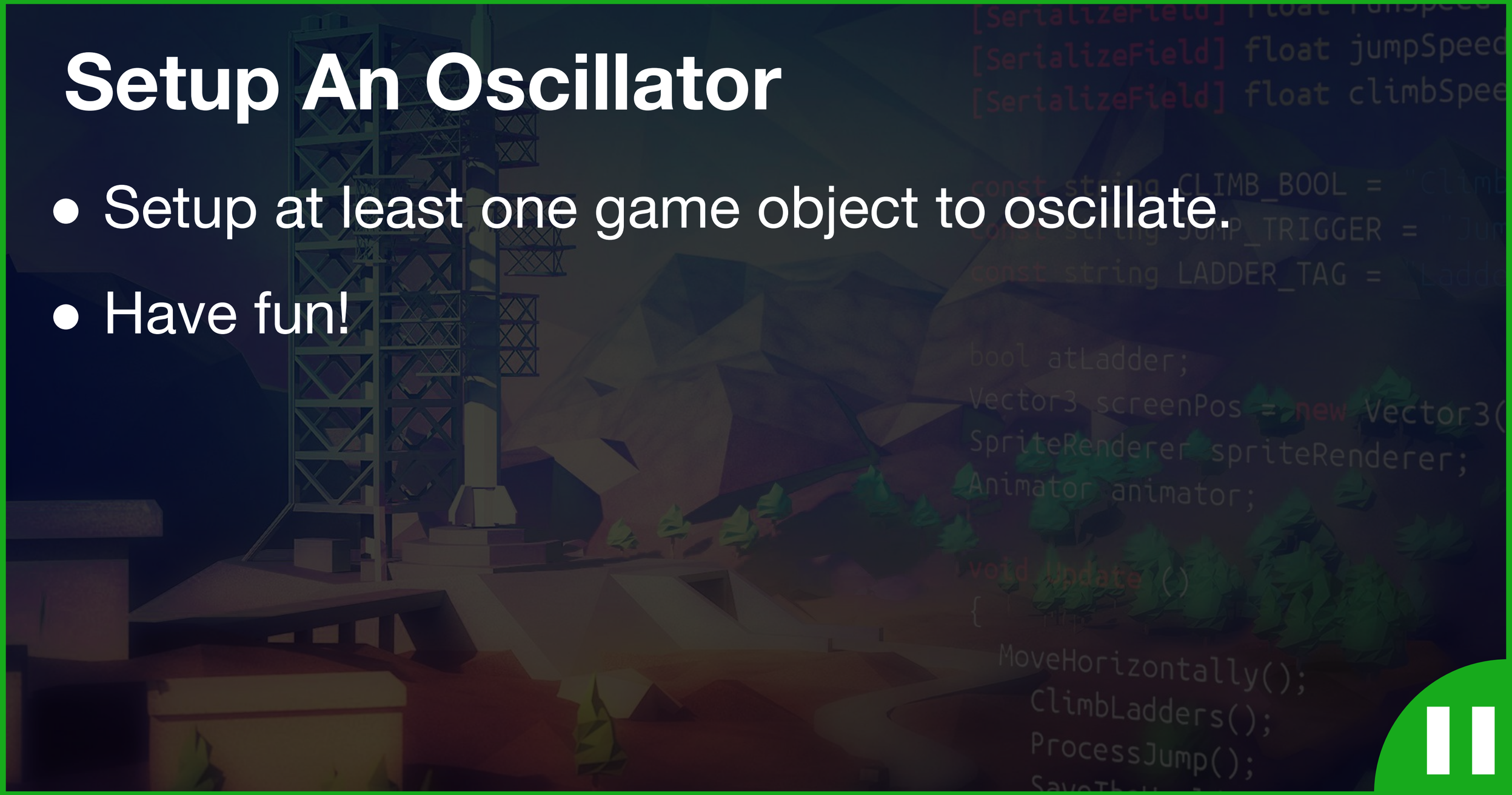
Unity 2017.1.1

Mathf.Sin() For Movement Cycles


amplitude (m)
period (s)

Setup An Oscillator

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



Unity 2017.1.1

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; }
```

Unity 2017.1.1

Organise Your Assets

Organising Your Assets

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

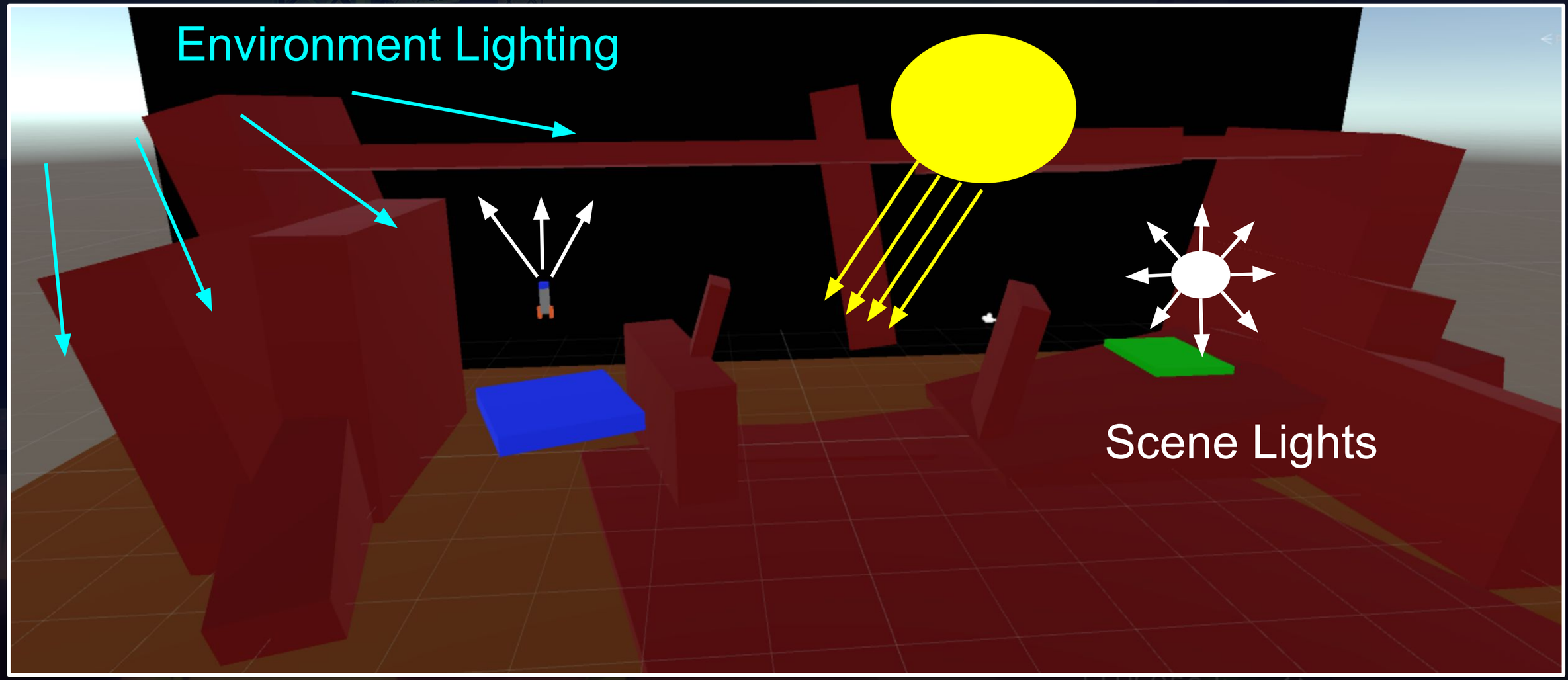
Organise Your Assets

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

Unity 2017.1.1

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.

Unity 2017.1.1

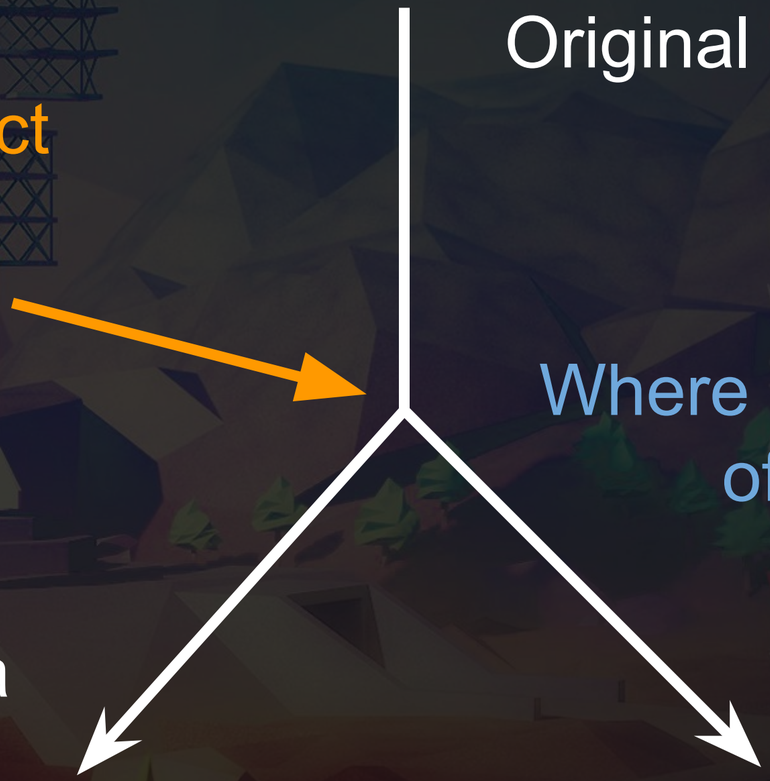
Nested Prefab Joy

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



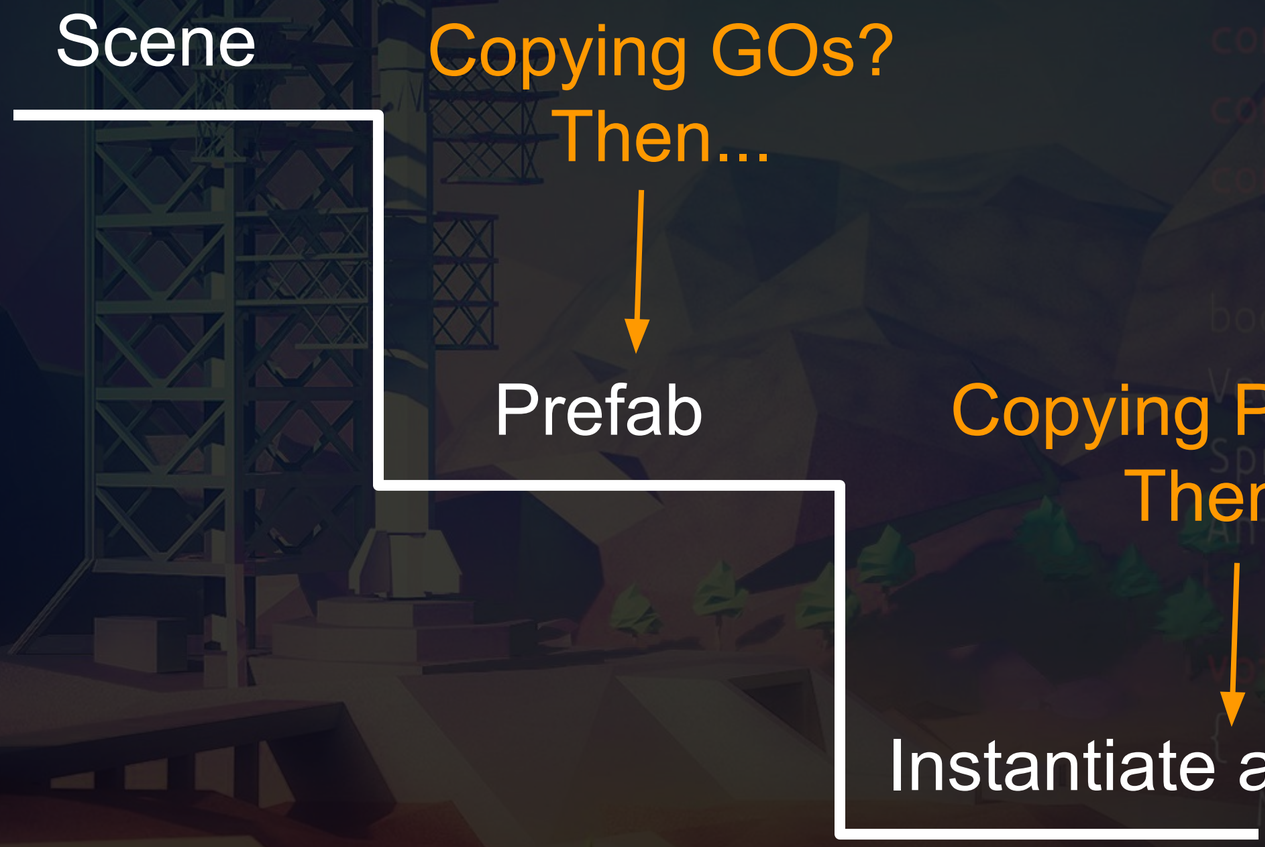
Original Particle Effect

Where is the “Source of Truth”?

Particle Effect Data on Rocket

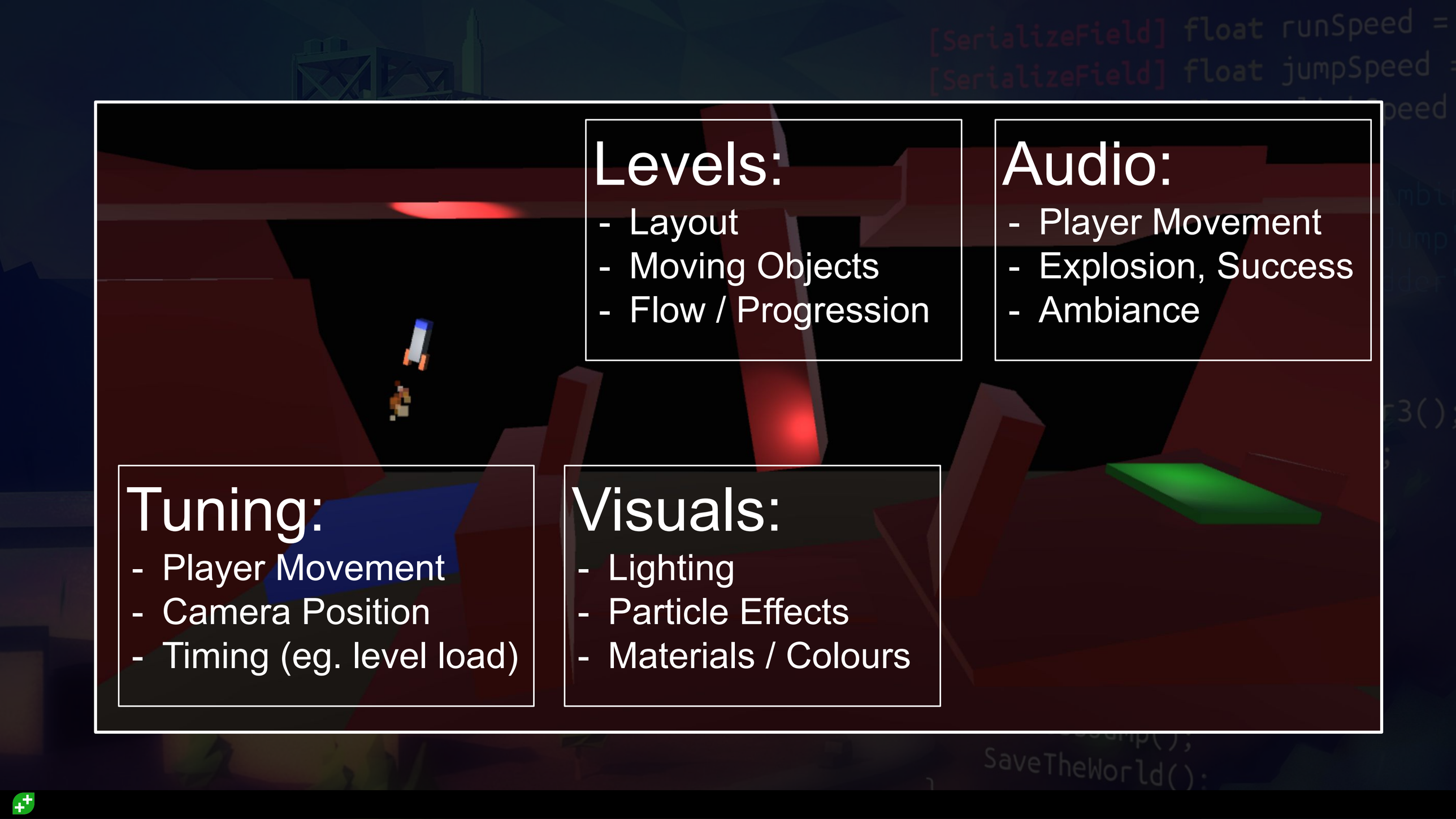
Particle Effect Data on Prefab

Where Is The Data?



Unity 2017.1.1

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

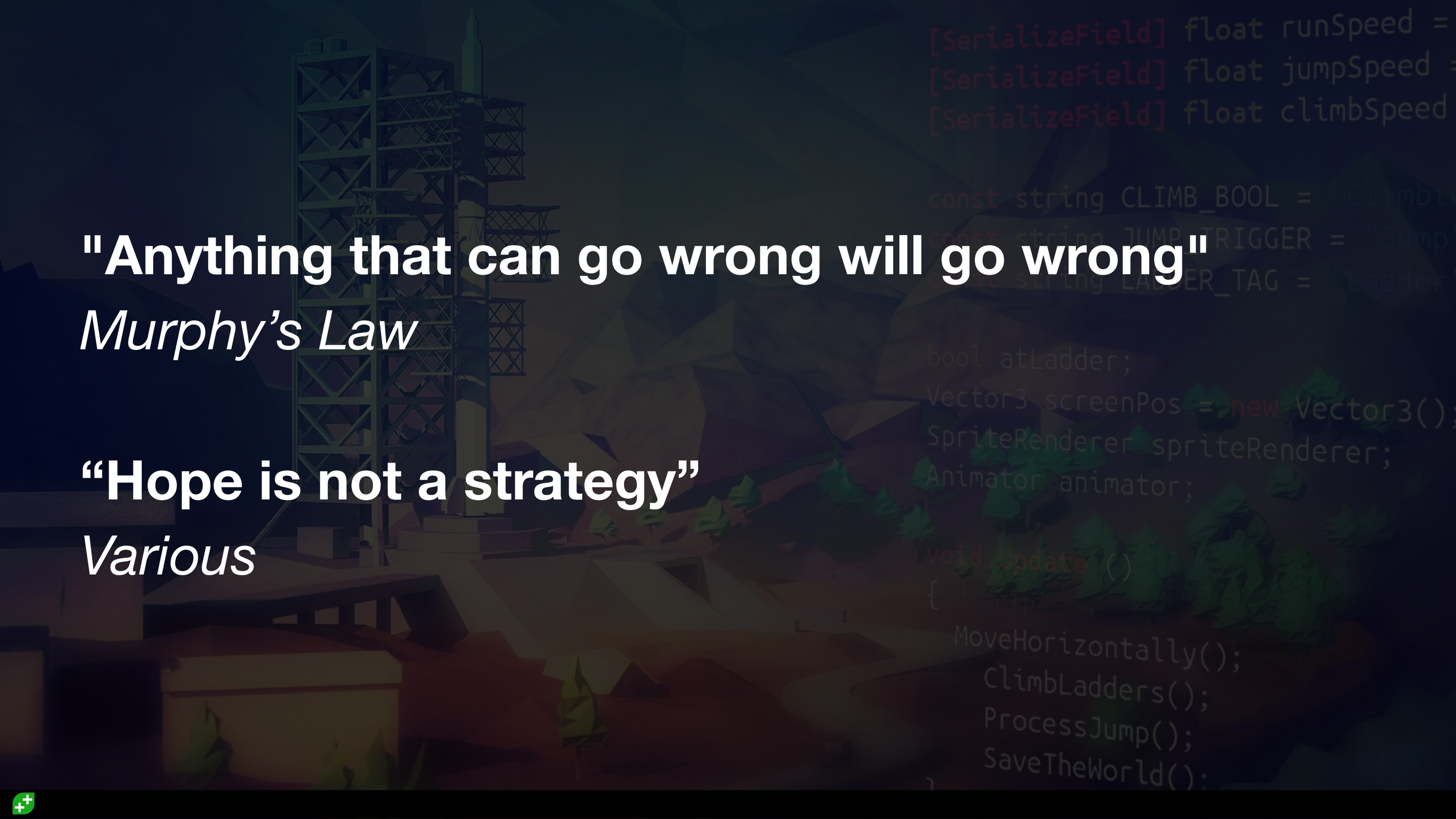
- We need to keep our players engaged for as long as we can
- Two options with our current game:
  1. Randomised levels of similar challenge level
  2. Sequential levels of increasing challenge level

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.

Unity 2017.1.1

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.

Unity 2017.1.1

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?

Unity 2017.1.1

Looping Through Levels

```
[SerializeField] float runSpeed =
[SerializeField] float jumpSpeed =
[SerializeField] float climbSpeed

const string CLIMB_BOOL =
const string JUMP_TRIGGER =
const string LADDER_TAG =

bool atLadder;
Vector3 screenPos = new Vector3();
SpriteRenderer spriteRenderer;
Animator animator;

void Update ()
{
    MoveHorizontally();
    ClimbLadders();
    ProcessJump();
    SaveTheWorld();
}
```

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

Unity 2017.1.1

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.

Unity 2017.1.1

Spit & Polish

```
[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();
}
```

In This Video...

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

Unity 2017.1.1

Section 3 Wrap-Up

```
[SerializeField] float runSpeed =
[SerializeField] float jumpSpeed =
[SerializeField] float climbSpeed

const string CLIMB_BOOL =
const string JUMP_TRIGGER =
const string LADDER_TAG =

bool atLadder;
Vector3 screenPos = new Vector3();
SpriteRenderer spriteRenderer;
Animator animator;

void Update ()
{
    MoveHorizontally();
    ClimbLadders();
    ProcessJump();
    SaveTheWorld();
}
```

```
[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();
}
```