Unity 2017.2

Welcome To Section 4

Argon Assault

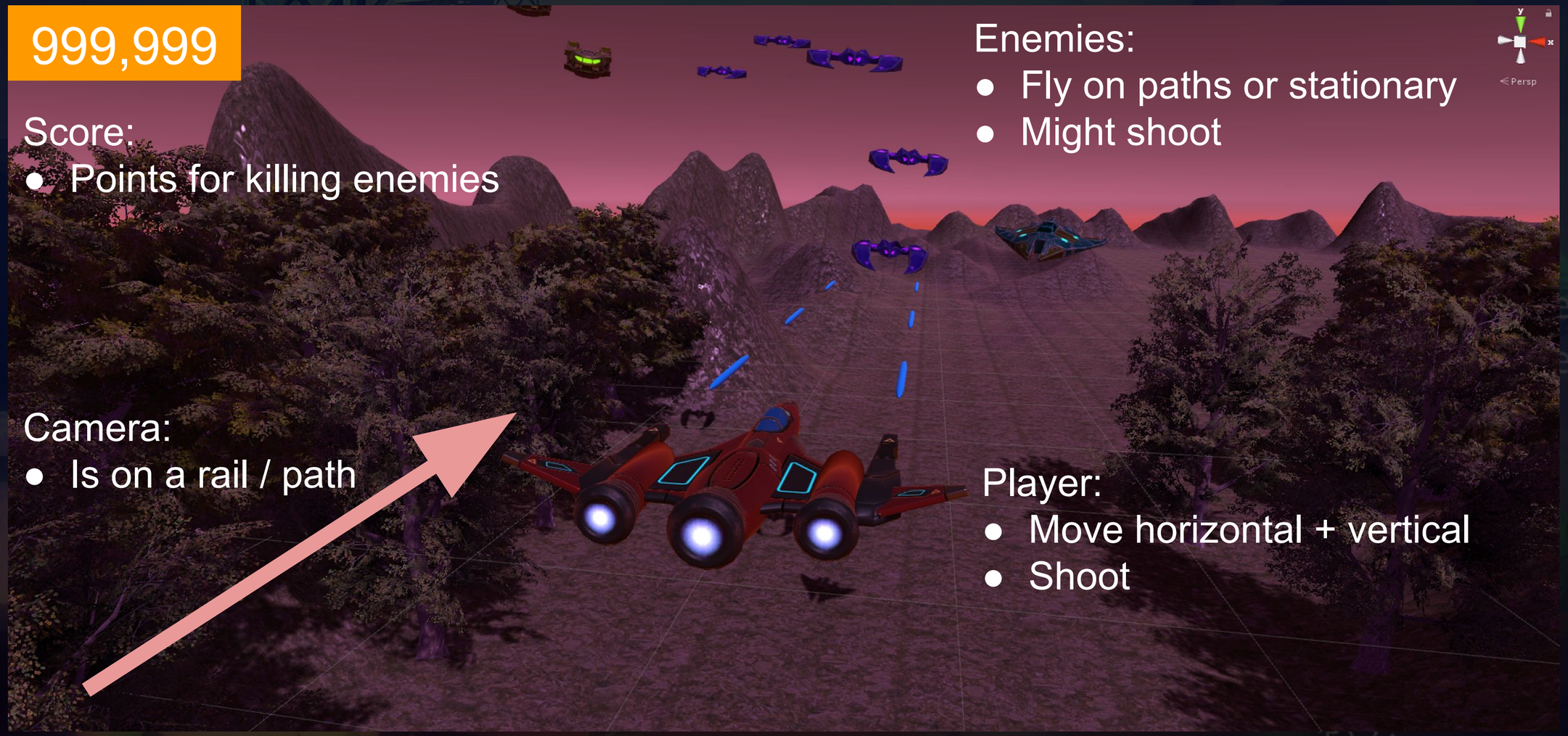
This is a 3D rail shooter inspired by such classics as Star Fox 64 and Panzer Dragoon.

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

Unity 2017.2

Argon Assault Game Design

Core Gameplay Overview



Argon Assault Game Design

Player Experience:

Chaos.

Core Mechanic:

Dodge and shoot.

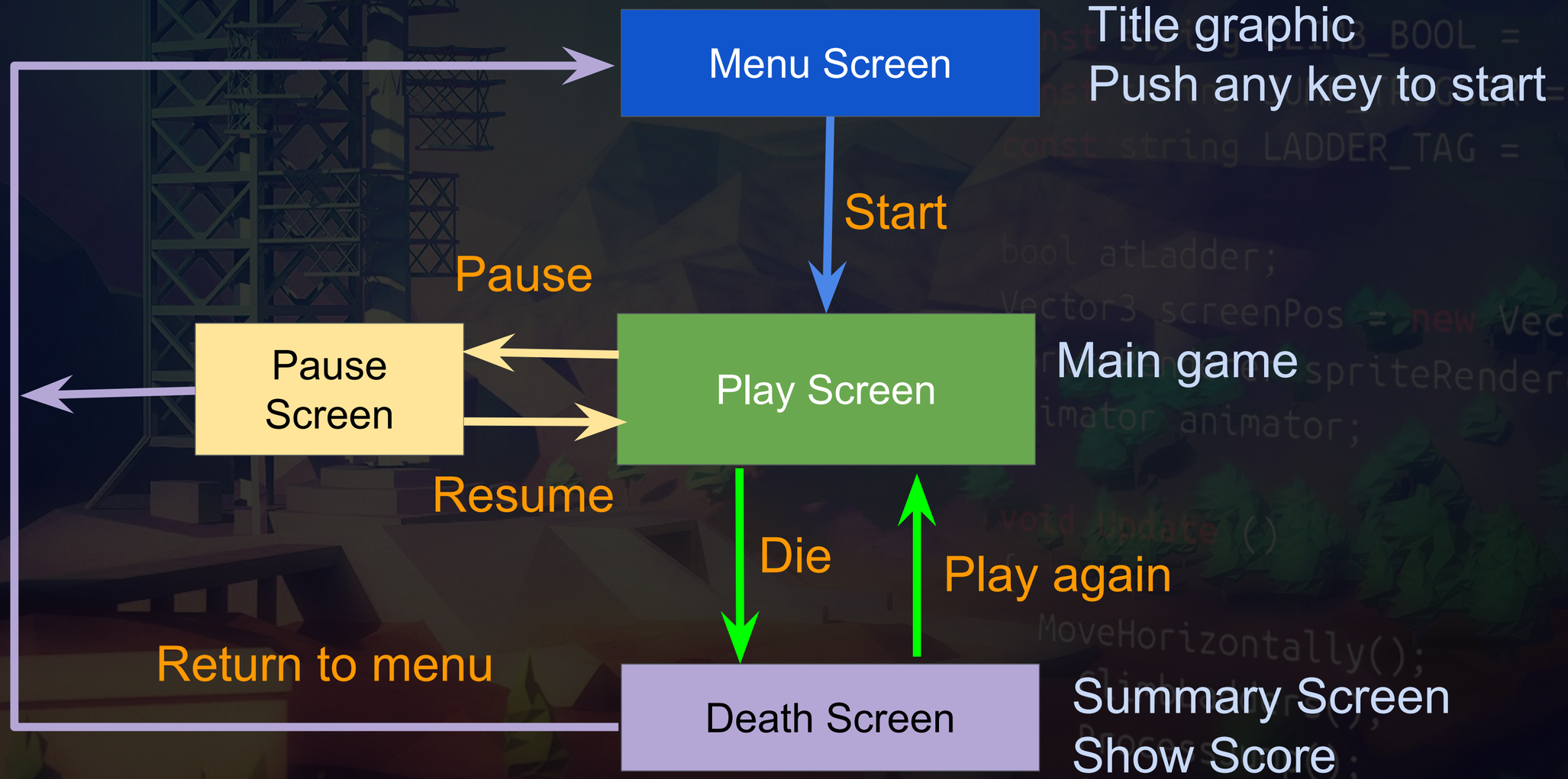
Core game loop:

Get as far as possible without dying in order to get the highest score possible. Start from beginning when die.

Game Theme

- Triggerou must save your planet, Argon, from destruction by an invading force.

Game Flow And Screens



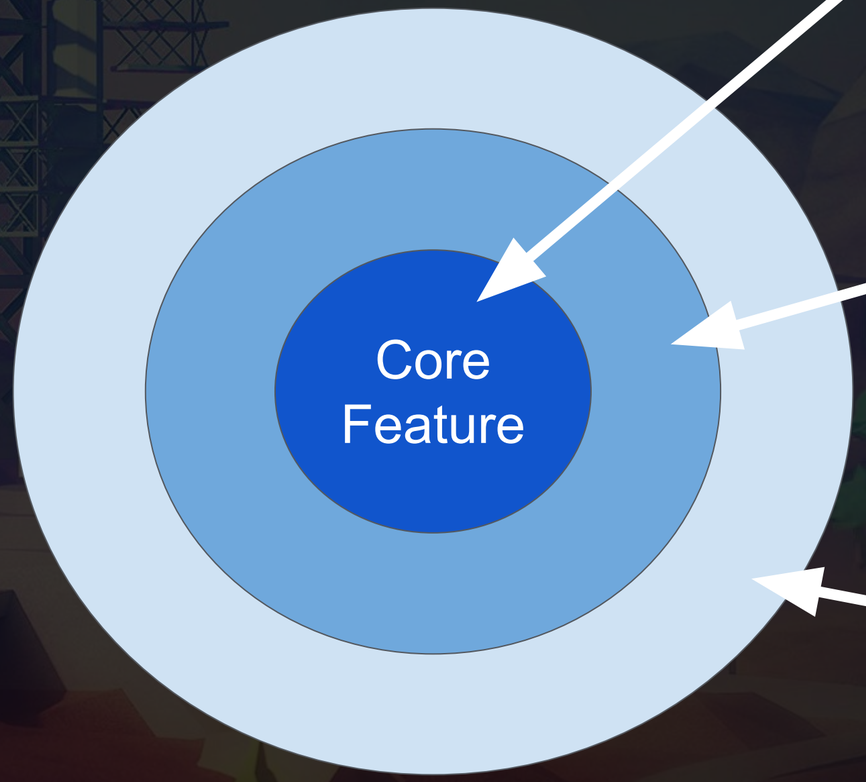
Features And Requirements

- **Player Movement:** Horizontal and vertical movement.
- **Shooting:** Player and enemies can shoot bullets which do damage to opponent. Speed, rate of fire and damage should be tunable.
- **Enemy Paths:** Enemies should travel on paths. Type of enemies, number of enemies and rate of spawn need to be tunable. Difficulty scaling and randomisation of paths is needed.
- **Score:** Points are given for killing enemies.
- **Camera Rail:** Path through the level that the camera follows.
- **Menu system:** Start menu, pause screen, death screen.

Stretch Goals - Damage and Shields

- **Player Shield:** When damaged, the player's shield depletes. Can be increased by picking up a shield pickup.
- **Pickups:** Flying over a shield pickup will increase the player's shields.
- **Momentary invulnerability:** After taking damage, the player cannot take damage again for X time.

Onion Design For Argon Assault



1. Fly on rails
2. Shoot enemies + score
3. Shield + pickups

Tech Specs

- 1920x1080 16:9 aspect ratio.
- Build to PC, Mac, Web.

Triggerour Game Name And Theme?

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

Unity 2017.2

Update Unity & Create Project

Update And Create

- Update Unity.
- Create new 3D project.
- Set up your source control repo.

Unity 2017.2

Create Terrain

Rules Of Prototyping

- Focus on simulating “**actual gameplay**”.
- Be fast & loose. Cheat. Cut corners.
- Don’t be precious.
- Focus on the most important features first.
- Remember your player experience.
- Quickly remove risk.

Our Early Development Priorities

- Establish a simple world
- Solve the risk of camera movement
- Get the player moving and avoiding
- Get the player shooting and hitting

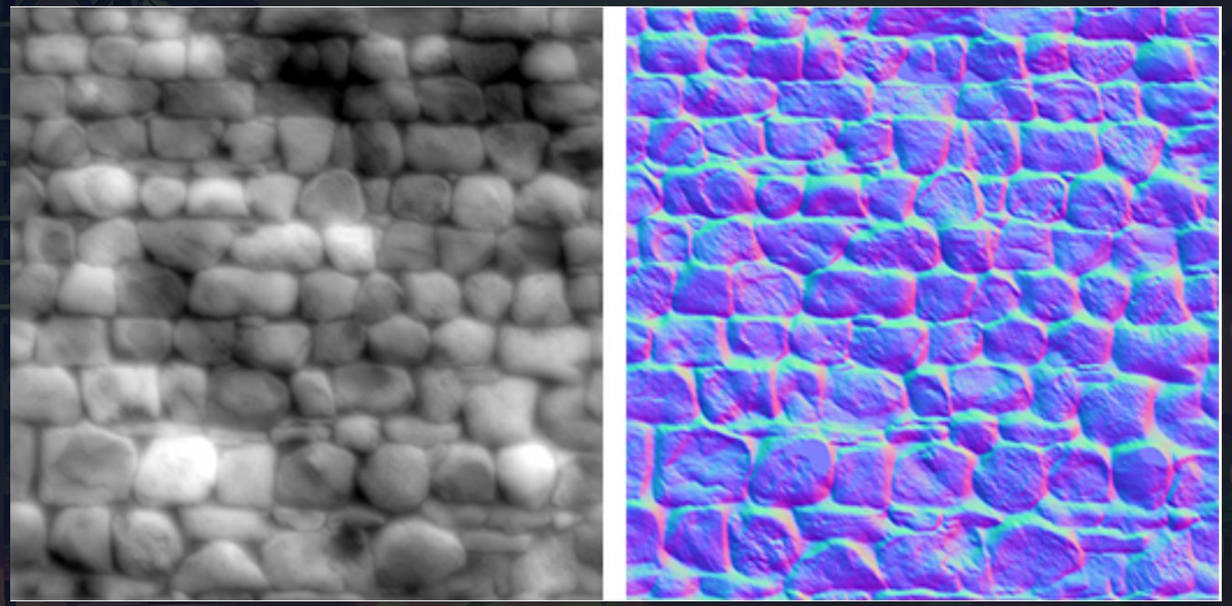
Create A Starting World

- Create a 1000x1000 terrain.
- Lower the terrain on y axis, then paint back up to 0.
- Play around with raising and lowering.
- Add some placeholder mountains.

Unity 2017.2

Texture The Terrain

Bump Map, Height Map, Normal Map



- **Bump map**: texture with information on how light catches a shape.
- **Height map**: uses white-to-black scale to show height.
- Normal map: uses RGB values to indicate x, y, z facing direction.

Texture Triggerour Terrain

- Download textures.
- Set up terrain textures.
- Paint some detail in your world.
- Share a screenshot!

Unity 2017.2

Add A Skybox

Add A Skybox

- Download a skybox asset.
- Set up a skybox material.
- Apply the skybox to your scene.

Unity 2017.2

Add Player Ship

```
[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();
}
```

Modular Asset Pack - Ebal Studios

Import / Create Player Ship

- Import our package or use your own assets.
- Ensure pivot point is central and object is prefabbed.

Unity 2017.2

Setup A Splash Screen

Set The Scene

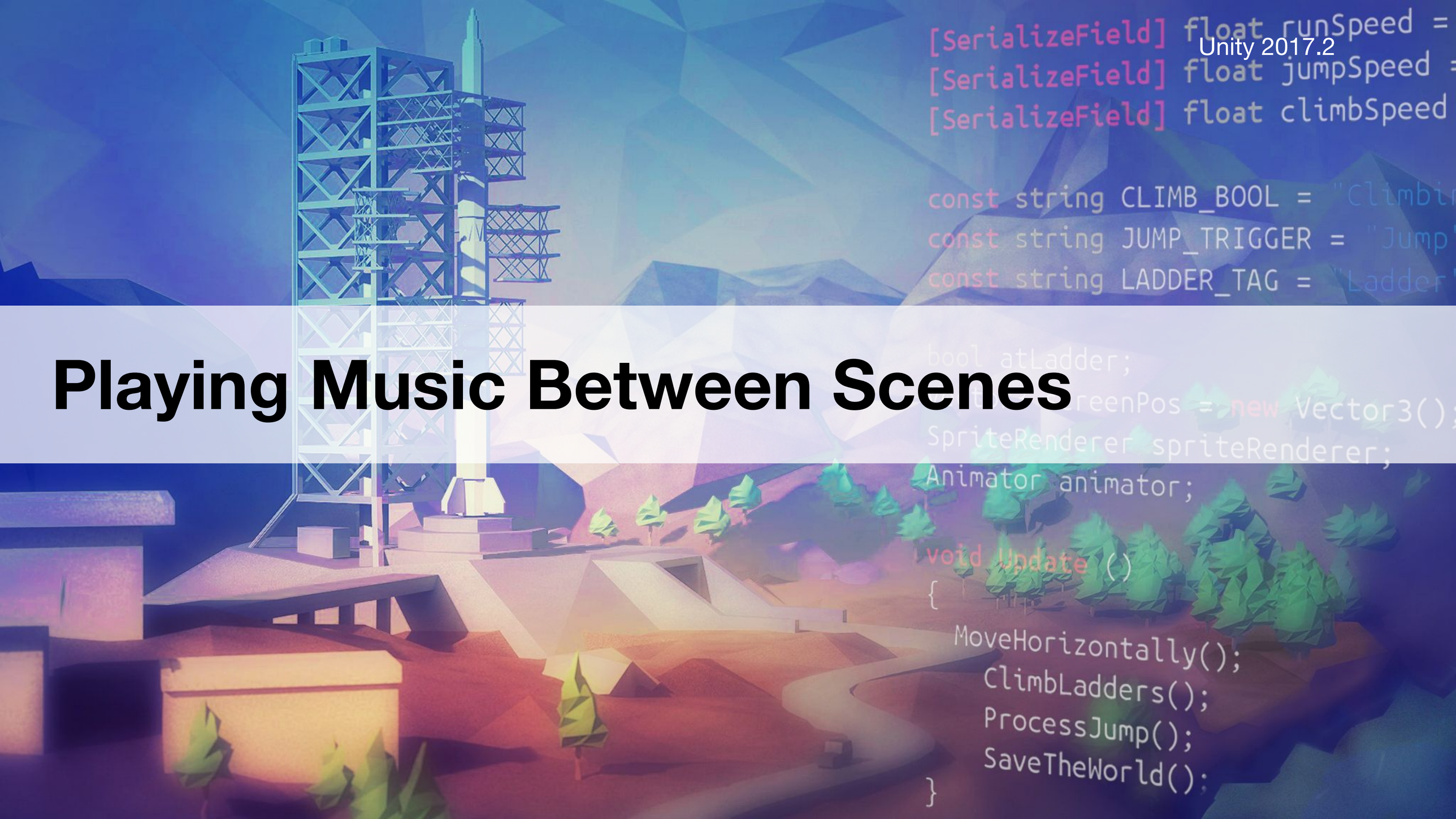
- Find, create, beg, steal* or borrow some music.
- Get something you feel good about.

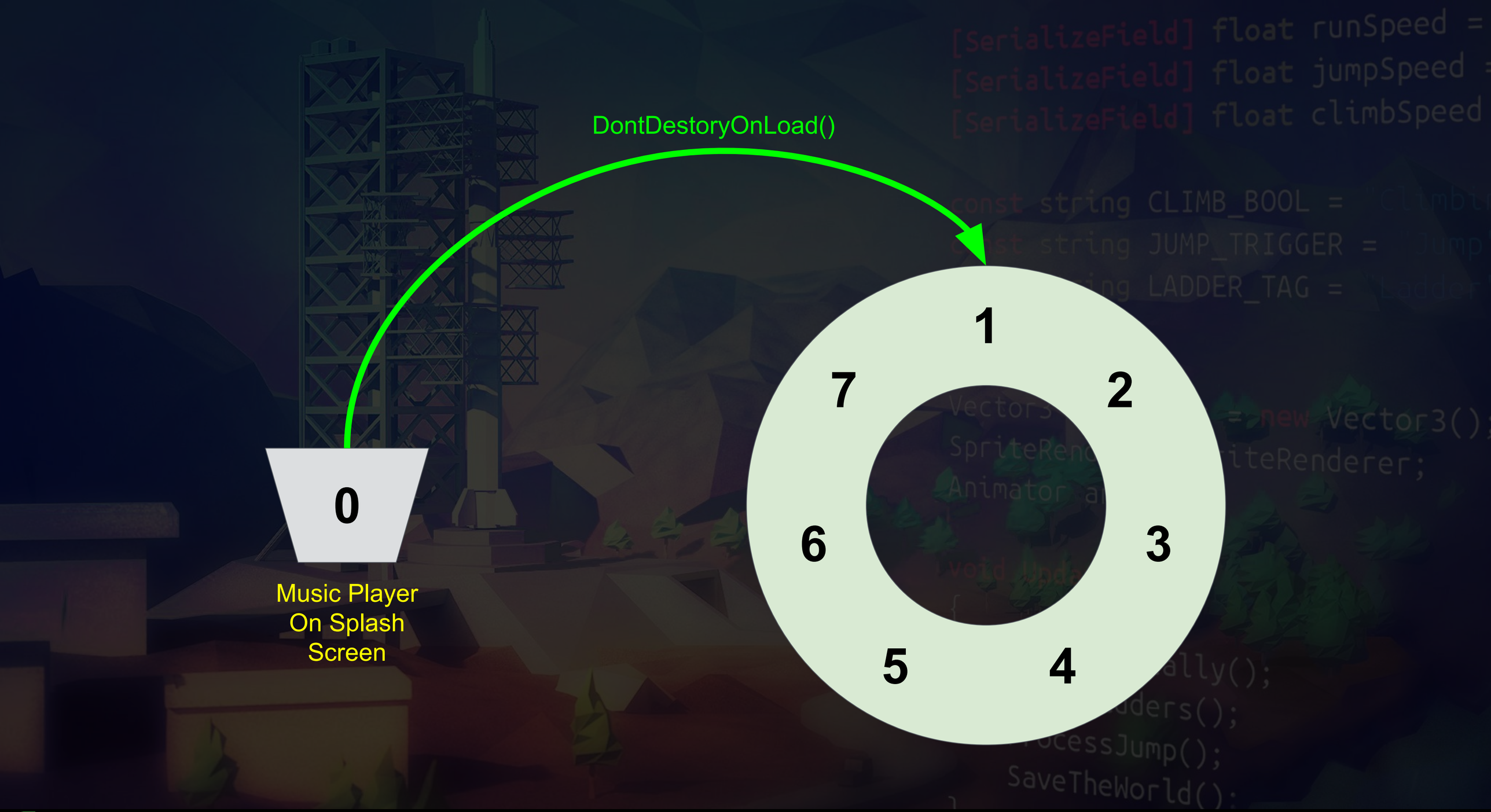
** Please do NOT actually steal it.*

Bonus: Make level load after a short delay.

Unity 2017.2

Playing Music Between Scenes




DontDestoryOnLoad()
0
Music Player
On Splash
Screen
1
2
3
4
5
6
7

The Music Plays Continuously

- The music should continue as the scene loads.

Hint 0: This is just one line of code.

Hint 1: The clue is in the resources.

Unity 2017.2

Unity’s Waypoint Utility Scripts

My Attitude To Asset Packs

- Don’t worry about understanding everything
- Do you understand everything about Unity?
- Do the best with what you have
- Use other people's work where you can
- Have faith you’ll dig-into the details when needed
- AND continue build your fundamental knowledge
- Pick your battles when it comes to learning.

Share Triggerour Initial Fly-through

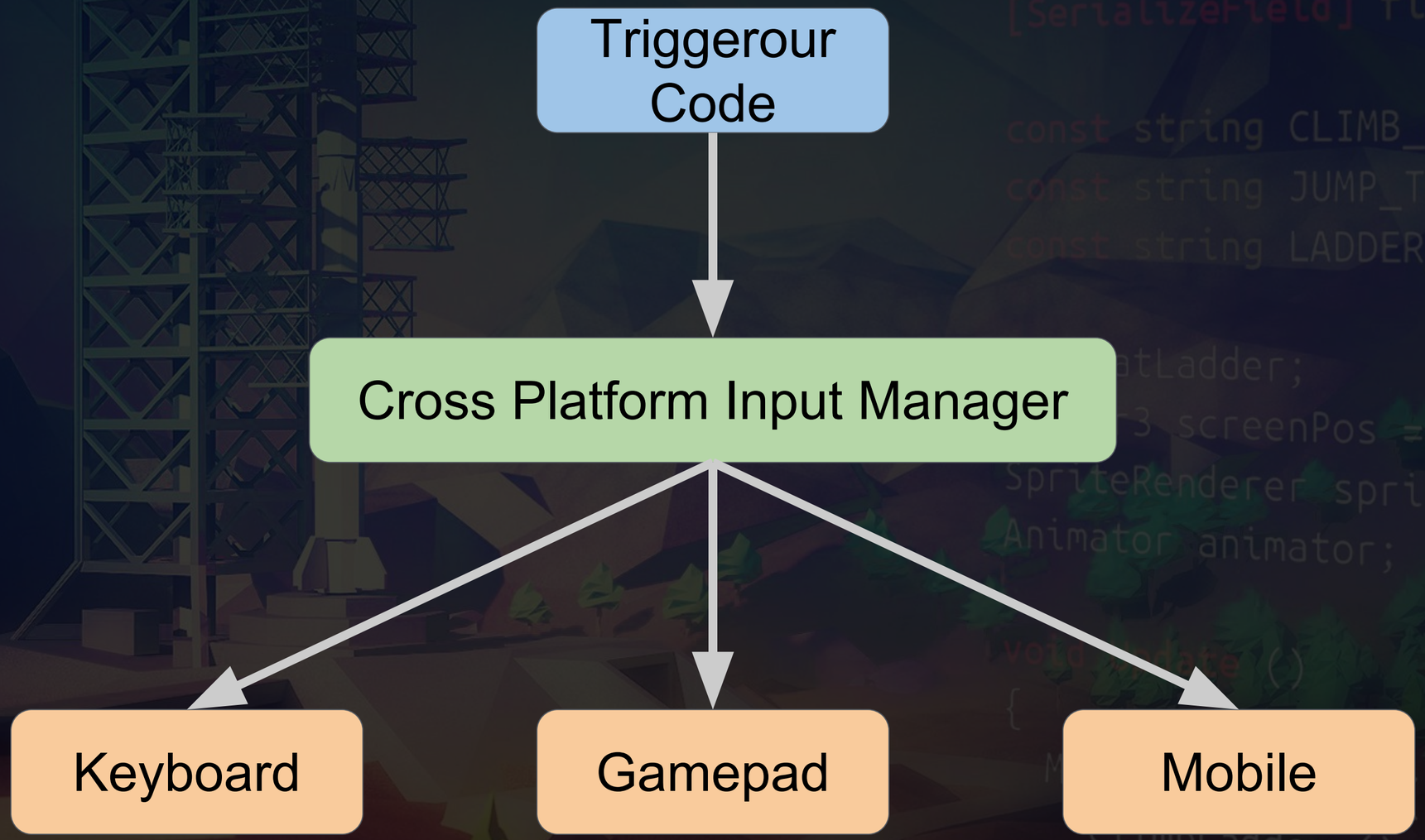
- Don’t spend too long for now
- Terrain and path may change later
- Excuse the sloppiness of the tool
- The camera should fairly smoothly fly loops
- Share a 20s video clip including splash & music.

Unity 2017.2

Instructor Hangout 4.1

Unity 2017.2

Using Cross Platform Input


Triggerour Code
Cross Platform Input Manager
Keyboard
Gamepad
Mobile

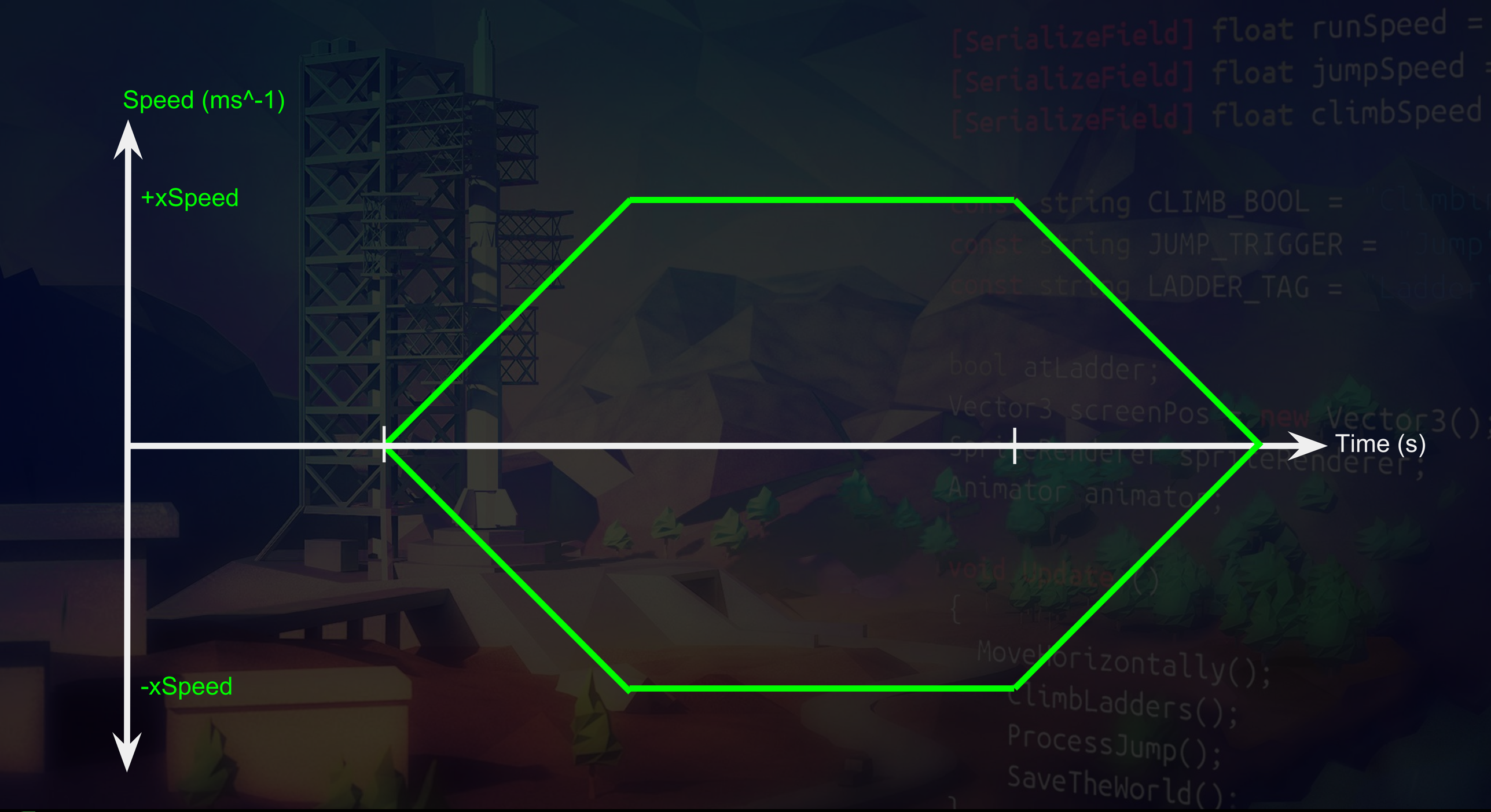
Explore Cross Platform Input

- At least keyboard or gamepad should “work”.
- Explore the values on the console.
- Ideally you have a gamepad plugged-in.

Unity 2017.2

Input Sensitivity & Gravity

Throw (unitless)
+1
-1
sensitivity
gravity
Time (s)
Button pressed / released


Speed (ms^-1)
+xSpeed
-xSpeed
Time (s)

Log `xOffset` Value

- Calculate the required `xOffset` for this frame
- This value should be framerate independent
- Log this value to the console
- Check against the FPS counter that it’s “rightish”.

Unity 2017.2

Mathf.Clamp() To Restrict Movement

Speed (ms^-1)
+xSpeed
-xSpeed
Time (s)

Position (m)
+xRange
-xRange
Time (s)

Using `Mathf.Clamp()`

- Restricts the output to a range of inputs.
- Easier and clearer than using `if` statements.
- Simple to use, for example…

```
pos = Mathf.Clamp(rawPos, -10f, 10f);
```

Triggerour Ship Stays On Screen

- Triggerour ship’s movement is constrained to the screen.
- It feels natural at either edge.
- Ideally use just one number, +/- from centre.

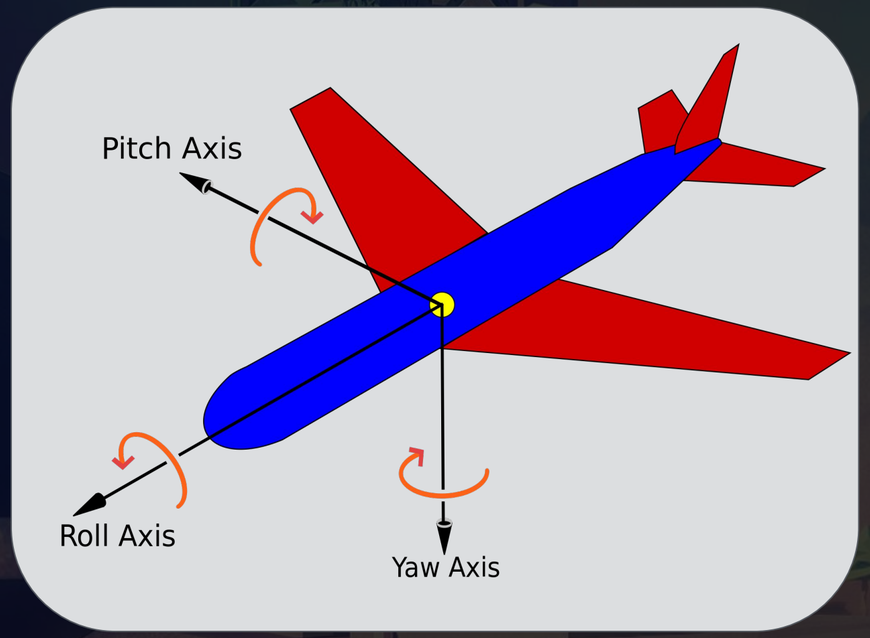
Unity 2017.2

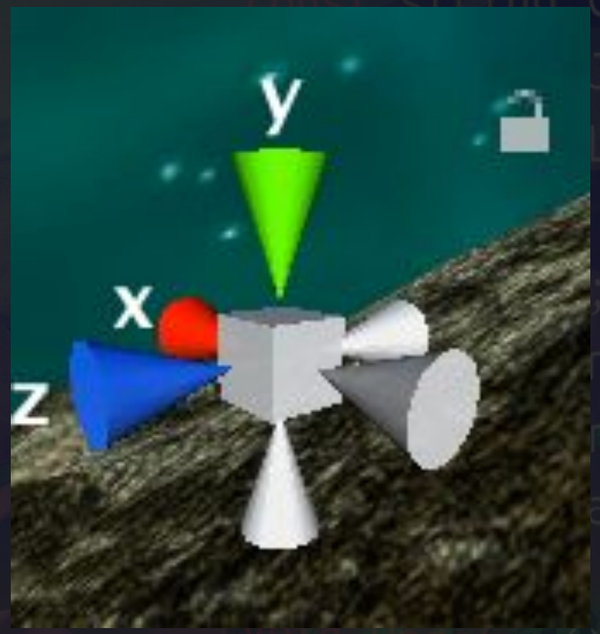
Understanding Roll, Pitch & Triggeraw

Clamp Vertical Movement

- The ship should move vertically.
- The movement should feel natural.
- The range should be clamped.
- This time you can have a separate yMin and yMax if you prefer that your ship starts off-centre.

Roll, Pitch and Triggeraw





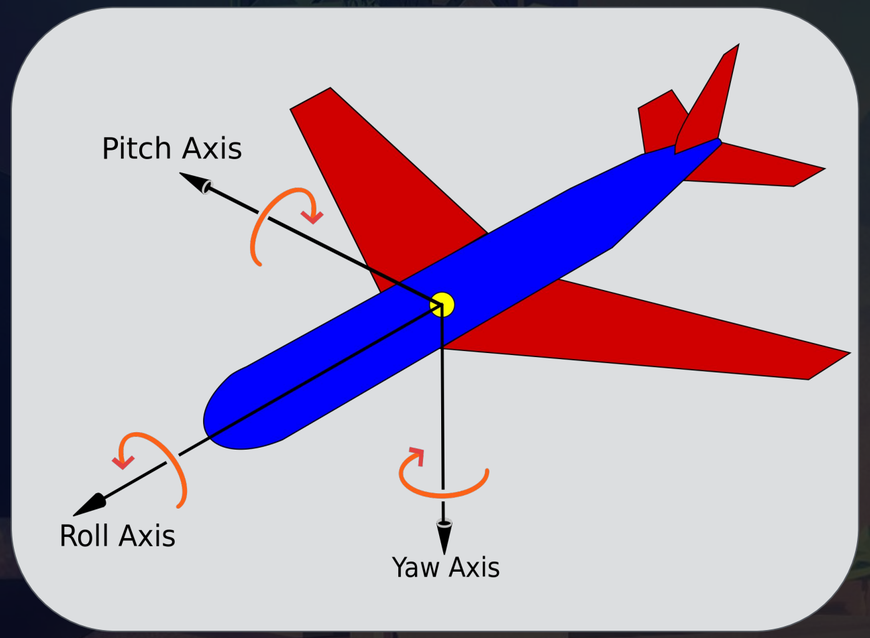
Unity 2017.2

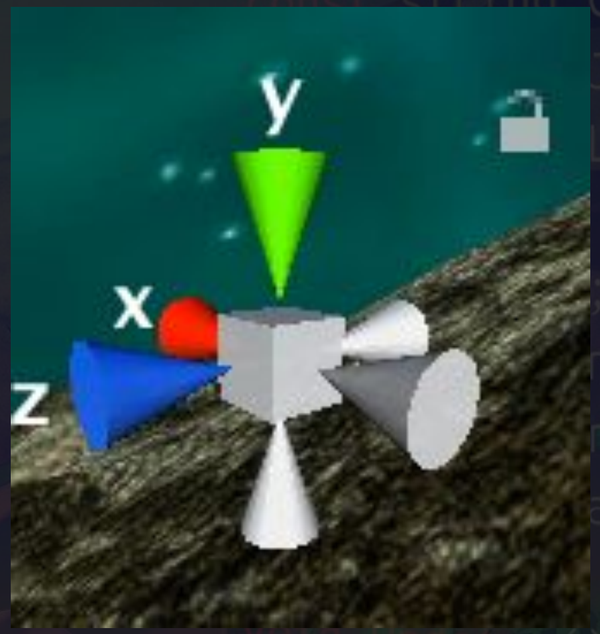
How To Set Local Rotation

Explore Rotations

- Have the editor in Local mode (x key).
- Use only the rotation tool in Scene view.
- Get to the following configuration…

Setting Local Rotation






```
transform.localRotation = Quaternion.Euler(  x  ,  y  ,  z  );
transform.localRotation = Quaternion.Euler(pitch, yaw, roll);
```

Unity 2017.2

Rotate Ship With Position & Throw

Our Rotation Couplings

	Position On Screen	Control Throw
Pitch	Coupled	Coupled
Triggeraw	Coupled	-
Roll	-	Coupled

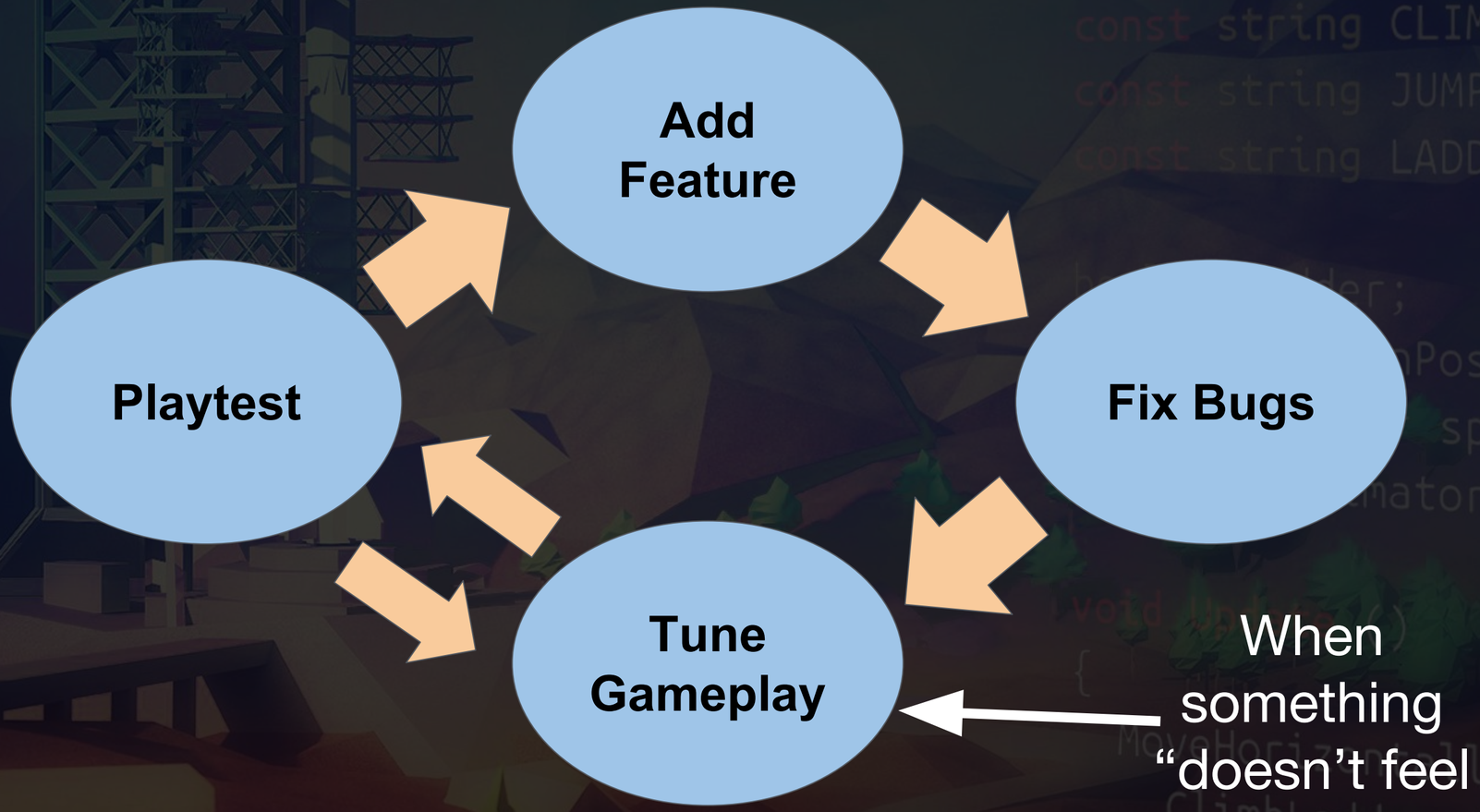
Finish Our Rotation Scheme

- Refer to the table on the previous slide.
- All of our intended rotations should work.
- Triggerour parameters should feel good.
- Show off your work, maybe a 20s video.

Unity 2017.2

Time To Tune And Tweak

When To Tune Trigger Our Gameplay



When something “doesn’t feel right”

Tweak And Tune Ship Movement

- Spend some time tuning your game:
  - Speed of movement on rail (approximate).
  - Camera distance and FOV.
  - Player speed, rotations, clamps.
  - Obstacles to avoid.
  - Obstacles to “shoot”.

Unity 2017.2

Create Bullet Particles

Create Bullet Particles

- Create projectiles to be shot from your ship.
- Tune them to look like lasers.
- Prefab the particle system and create 2 instances to act as dual weapons.

Unity 2017.2

Explosion Particles & SFX

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

Create Explosion Effect

- Use sphere shape.
- Effect lasts about 1 second.
- Make the effect multi-coloured.
- Don’t use standard particle material.

Unity 2017.2

Dodge And Shoot

```
[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();
}
```

Our Core Game

- Dodge
  - Terrain
  - Obstacles
  - Enemies
- Shoot
  - Enemies

Dodge And Shoot

- Create 2 or 3 obstacles to avoid.
- Add a 2 or 3 enemies to shoot.
- Share a screenshot!

Unity 2017.2

Instructor Hangout 4.2

Unity 2017.2

Project Tidying Checklist

When tidying / revisiting a project...

- ❑ Ensure work is committed before “improving”
- ❑ Assets organised clearly and consistently
- ❑ Correct objects “prefabbed” and linked?
- ❑ Scan read code. Check for obvious issues.

Get Triggerour Coding Hat Back On

- Check-items on the previous slide are complete.
- Triggerou feel comfortable with where your project.
- Triggerou are crystal clear on the next step.

Share what your next step would be with us.

Unity 2017.2

Triggers & Collisions In Unity

	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

Which Cell Of Our Table Are We In?

- Which cell are we in for enemy interaction?
- What do we need to change to get messages?
- The message should print to the console.

Unity 2017.2

SendMessage() Between Components

Thinking Of Creating A Manager?

Instead of *SomeManager*, consider one of these more meaningful words...

Coordinator, Builder, Writer, Reader, Handler, Container, Protocol, Target, Converter, Controller, View, Factory, Entity, Bucket.

https://stackoverflow.com/questions/1866794/naming-classes-how-to-avoid-calling-everything-a-whatevermanager

Messaging Other Components

`SendMessage("Method Name")`

Calls methods called `MethodName()` on all other scripts on the same game object.

Freeze Controls On Impact

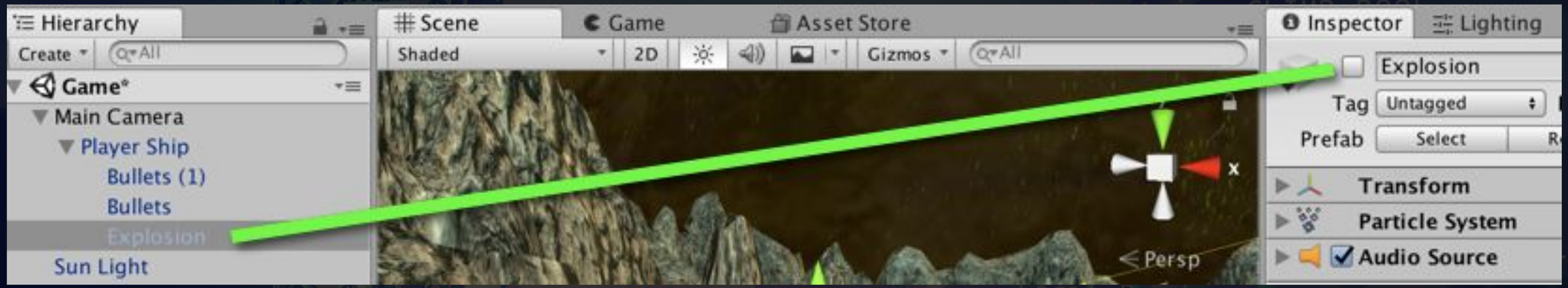
- When the player hits anything, the controls should freeze.
- This should happen using a message sent from `CollisionHandler` to `PlayerController`.

Enjoy your first application of the actor model!

Unity 2017.2

Enable A Gameobject From C#

Enable A Game Object During Play



1. Ensure particles & audio “Play On Awake”.
2. Have a reference to the `gameObject`, e.g. `deathFX`
3. Use `deathFX.SetActive(true);`

Finish The Death Sequence

- If the player hits anything the sequence starts.
- Play an explosion (existing prefab is fine).
- Load the same scene after a configurable delay.

Unity 2017.2

Singleton Pattern Without Statics

Self Destructing Music Players

Tune Your Sound Volumes

- An easy challenge for a break.
- Your music and SFX volumes are balanced.

Unity 2017.2

Detecting Particle Collisions

To Use Particles As Bullets...

1. Ensure particle collision & message sending is on.
2. Add a **non-trigger** collider to the target.
3. Use `OnParticleCollision()` on the target.

Get Your Enemies Destroying

- When the particles from your gun hit an enemy, the enemy `gameObject` should be destroyed.

Unity 2017.2

Making Scripts Add Components

Adding A Component From Script

```
gameObject.AddComponent<TypeToAdd>();
```

For example...

```
Collider someColliderVariableName =
gameObject.AddComponent<SphereCollider>();
```

// now you can set properties via the variable

Enemies Add Box Colliders

- When the game starts any game object with an Enemy.cs script should add a box collider to itself.
- This box collider should have trigger un-checked.

Hint 0: Make sure there is no box collider before entering play mode, and pause to inspect.

Hint 1: use the . to find `isTrigger` property.

Unity 2017.2

Instantiating At Runtime

Instantiating At Runtime

```
[SerializeField] GameObject prefabToUse;

Instantiate(
    prefabToUse,
    worldSpacePosition,
    Quaternion.identity
);
```

Share A Short Clip

- You should have shared a short video clip with us
- This is an exercise in getting players excited.

Unity 2017.2

Add Simple Score UI

Simple Scoreboard Recipe

- Add from GameObject > UI > Text
- Set anchor points, and position and scale
- Choose font and font size / colour
- Write a `ScoreBoard.cs` script
- `FindObjectOfType<ScoreBoard>();`
- Call `ScoreHit()` or similar from dying enemy.

Customise Your Font

- You should have chosen a font, positioned and scaled it to your taste.
- Why not share your best screenshot with the community now?

Unity 2017.2

A ScoreBoard Class

What Is A Class Anyway?

- A **template** for creating instances
- It’s the script equivalent of a prefab
- In Unity Classes are typically components
- Instances can have a different data (like prefabs)
- Classes usually have one or more functions.

Help Each Other Out

- Pop on the Q&A, forum or Discord.

Unity 2017.2

Instructor Hangout 4.3

Unity 2017.2

Introducing Encapsulation

Simple Scoreboard Recipe Revisited

- Add from GameObject > UI > Text
- Set anchor points, and position and scale
- Choose font and font size / colour
- Write a `ScoreBoard.cs` script
- `FindObjectOfType<ScoreBoard>();`
- Call `ScoreHit()` or similar from dying enemy.

Get The Score Updating

- The score should update when an enemy is destroyed.
- Bonus: create some other score-increasing event.

Unity 2017.2

Back-pedalling With Version Control

Ways Of Back-pedaling

- **Stash**: Quickly undo a single stage of changes*
- **Branch & Reset:** Undo a chain of changes*
- **Reset**: Not recommended as destructive.

** Neither will delete files, you'll have to "Remove" manually.*

Restore Your Project

- Restore your project to its state before this lecture started.

Unity 2017.2

Iterating With foreach

```
[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();
}
```

How to iterate over a collection

```
foreach (TypeOfThing thing in things)
{
    thing.DoSomething();
    DoSomethingWith(thing);
    // etc
}
```

Write `DeactiveateGuns()`

- When you stop firing the guns should stop
- When you start firing again they should start
- A new method is fine, duplicating code once is OK.

Unity 2017.2

Enemy Health System

Set Up Enemy Hit Points

- Enemies have hit points.
- Time takes to kill enemy seems about right.

Unity 2017.2

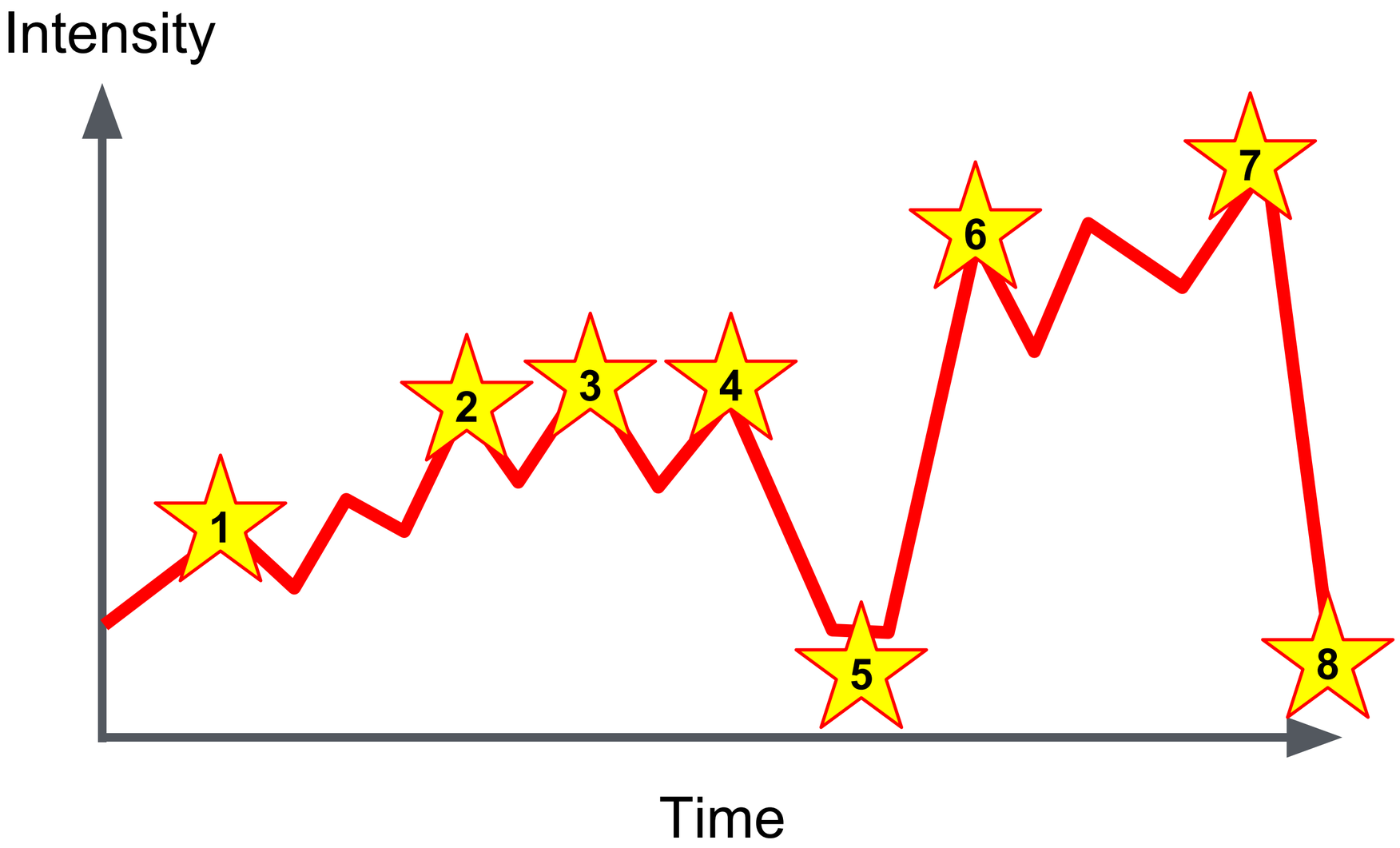
Level Design Beat Chart

Where Are We At?

- Core gameplay (dodge and shoot) - check!
- Reason to play (score) - check!
- Up next:
  - Final level and path.
  - Moving enemies.
  - TIMELINE, baby!

Your Level Beat Chart

- Your level should be a roller coaster.
- Moments from 1 to 10 in intensity.
- Progression (difficulty, enemy type).
- No 10 sec moments should be the same.
- Link to your story and player experience.



1. We’re under Attack!
2. Incoming, sector 5.
3. They’re hitting our shield generators!
4. Watch out near the mines!
5. Phew, I think that’s the worst of them.
6. Help! Help! They’re everywhere!
7. Holy mother of mercy, what is that!
8. You did it.

Create Your Level Beat Chart

- Think of the story you are telling and Player Experience.
- Create your beat chart.
- Share an image with us.

Unity 2017.2

Level Design Iteration

Level Design - What To Look For

- Layout to support your beats.
- Visual variation.
- Interesting dodging.
- Line of sight for spawning.
- Entry points for enemies.
- Room for enemy behaviour.
- Multiple paths.

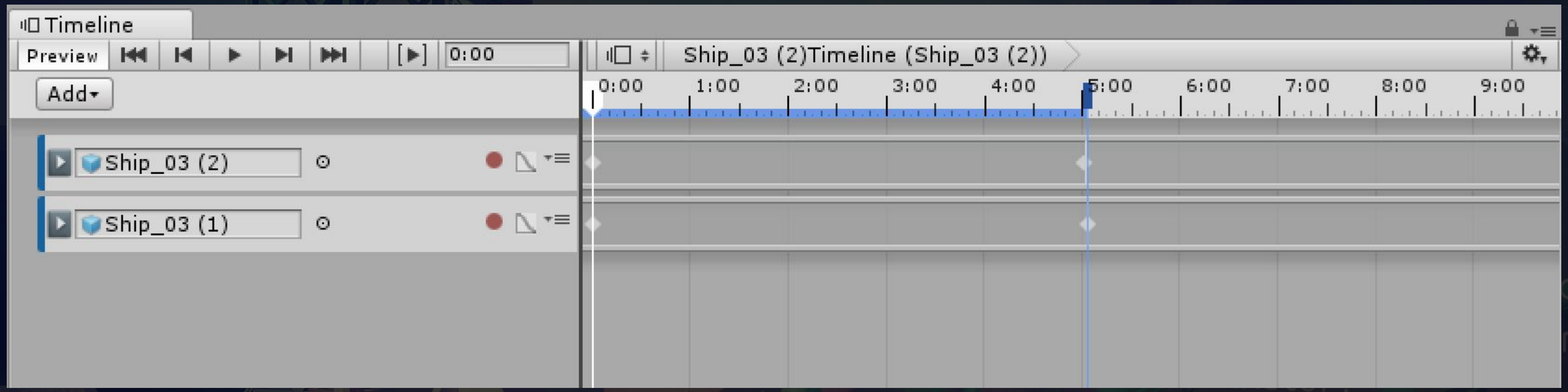
Iterate Your Terrain If Needed

- Create an interesting level that can support your beat chart.
- Think about line of sight.
- Think about enemy entry points.
- Do some practice fly-throughs.
- Prototype path but don’t make your final path yet - we’ll be using Timeline for that.

Unity 2017.2

Introduction To Unity Timeline

What Is Timeline?



- A tool to create cut-scenes, cinematics and gameplay sequences by visually arranging tracks and clips linked to Game Objects in your scene.

Use Timeline To Animate An Enemy

- Add a Playable Director to an enemy.
- Animate that enemy using timeline.
- Shoot and destroy the enemy!

Unity 2017.2

Unity Timeline For Player Path

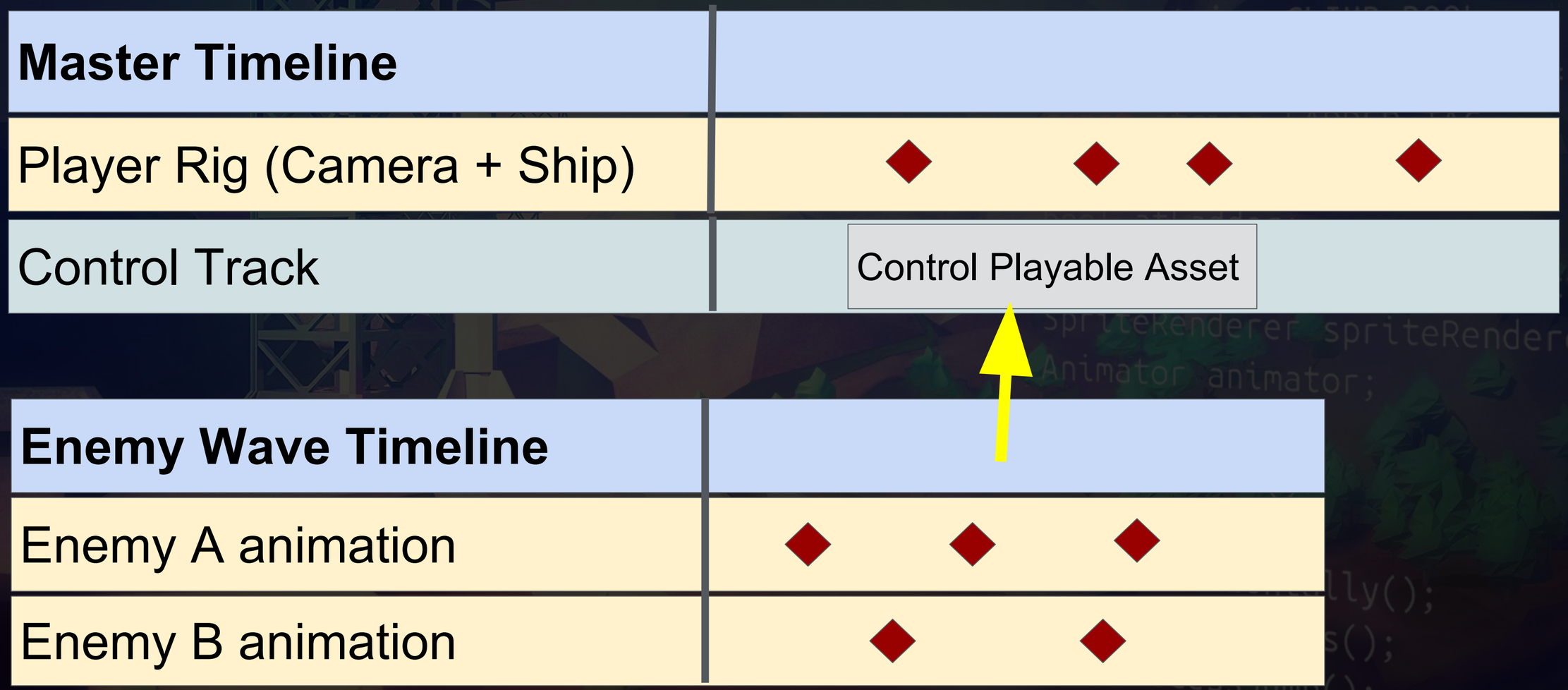
Use Timeline For Player Path

- Create Master Timeline.
- Add Player Rig to animation track.
- Create 1 minute of flight path.

Unity 2017.2

Unity Timeline For Enemy Waves

Nested Timelines On Control Tracks



Create Enemy Waves

- Create Timelines for multiple waves of enemies.
- Animate your enemies to your heart’s content.
- Re-use at least one wave.

Unity 2017.2

Adding Flavour With Timeline

Bit Of Flavour With Timeline

- Add, remove, animate, switch, transition:
  - Pretty much any Game Object
  - SFX, voice over, music
  - Cameras, lights
  - Particle effects
  - User interface elements
  - Character animations
  - World animations
  - And so on...

Add Some Flavour

- Create some additional story or cinematic or gameplay element with Timeline.
- Let us know what you come up with!

Unity 2017.2

Stop Particle Emission In C#

Stop The Particle Emission

- Use `.enabled = false` to stop emission

Hint 0: Use an "intermediate variable"

Hint 1: Start with…

```
var emissionModule =
gun.GetComponent<ParticleSystem>().emission
```

Unity 2017.2

Visual 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();
}
```

Add Your Own Visual Polish

- Consider:
  - Scene lighting
  - Jet particles
  - Fog
  - Textures
  - Overall atmosphere

Unity 2017.2

Section 4 Wrap-Up

Section 4 Wrap-Up

- Terrain, Timeline, Level Design, User Interface
- Classes, encapsulation, instantiating at runtime, singleton pattern
- Version control and good game dev practices

[SerializeField] float runSpeed =
[SerializeField] float jumpSpeed =
[SerializeField] float climbSpeed
const string CLIMB_BOOL =
JUMP_TRIGGER =
LADDER_TAG =
bool atLadder;
Vector3 screenPos = new Vector3();
SpriteRenderer spriteRenderer;
MoveHorizontally();
ClimbLadders();
ProcessJump();
SaveTheWorld();
END OF SLIDES

GameDev.tv

Unity 2017.2

Slide With Logo