GameDev.tv

Unity 2020.1

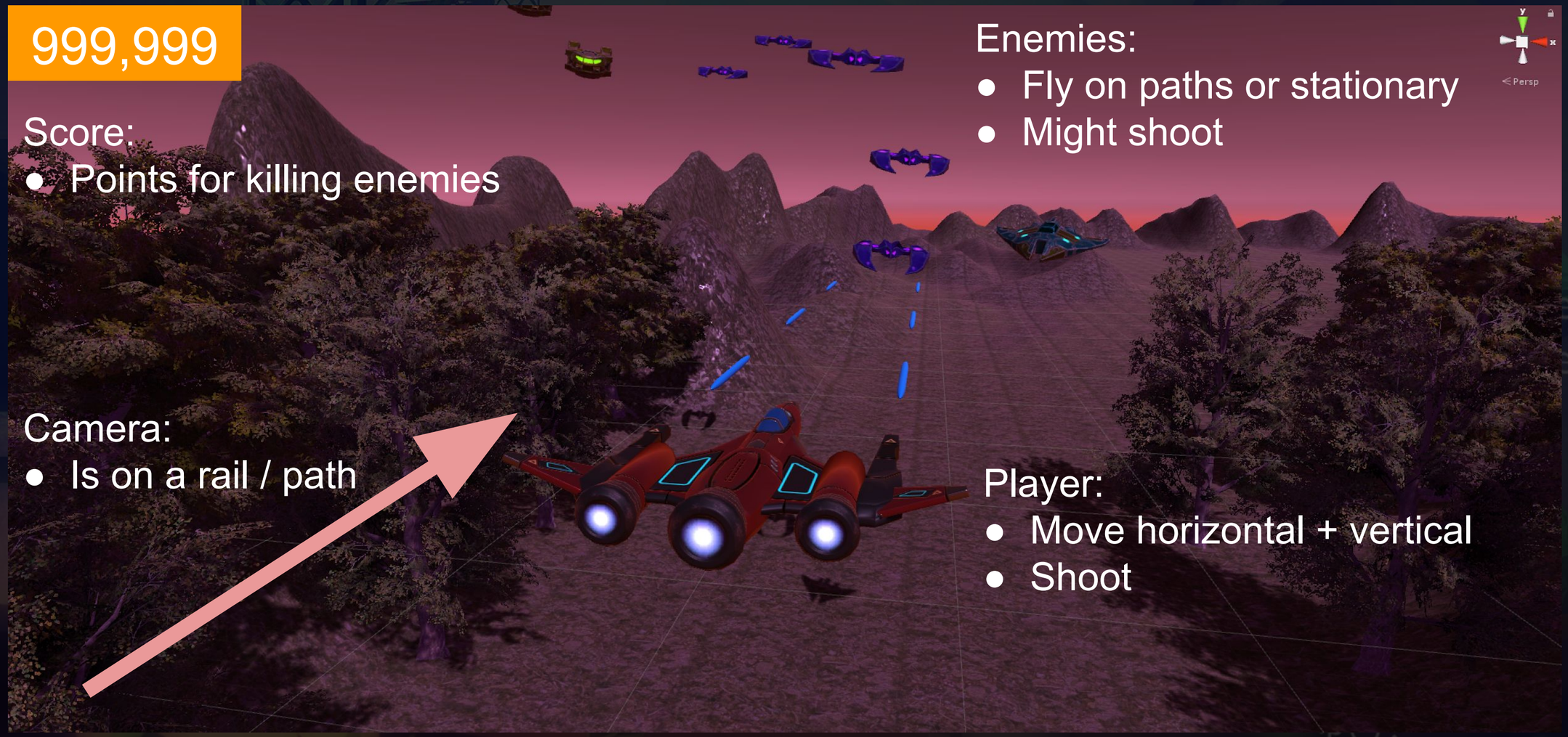
Section Intro - Argon Assault

GameDev.tv

Unity 2020.1

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

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

Features - In Priority

- **Camera Rail:** Path through the level that the camera follows.
- **Player Movement:** Horizontal and vertical movement.
- **Shooting:** Player shoot bullets which do damage to opponent.
- **Health:** Enemies have health that depletes when shot.
- **Enemy Paths:** Enemies should travel on paths and be hand placed by the designer.
- **Score:** Points are given for killing enemies.
- **Game Loop:** Upon dying, player restarts the level.

Other Features You Can Consider

- **Multiple Levels:** Finish a level and load the next.
- **Player Shield:** When damaged, the player's shield depletes.
- **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.
- **Weapon Upgrades:** Flying over a pickup could increase weapon damage.

Your Game Name And 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

How To Add Terrain

Create A Starting World

- Create a 600 x 600 terrain.
- Lower the terrain on y axis, then paint back up to 0 by setting your flatten value to 100.
- Play around with raising and lowering.
- Add some placeholder mountains.

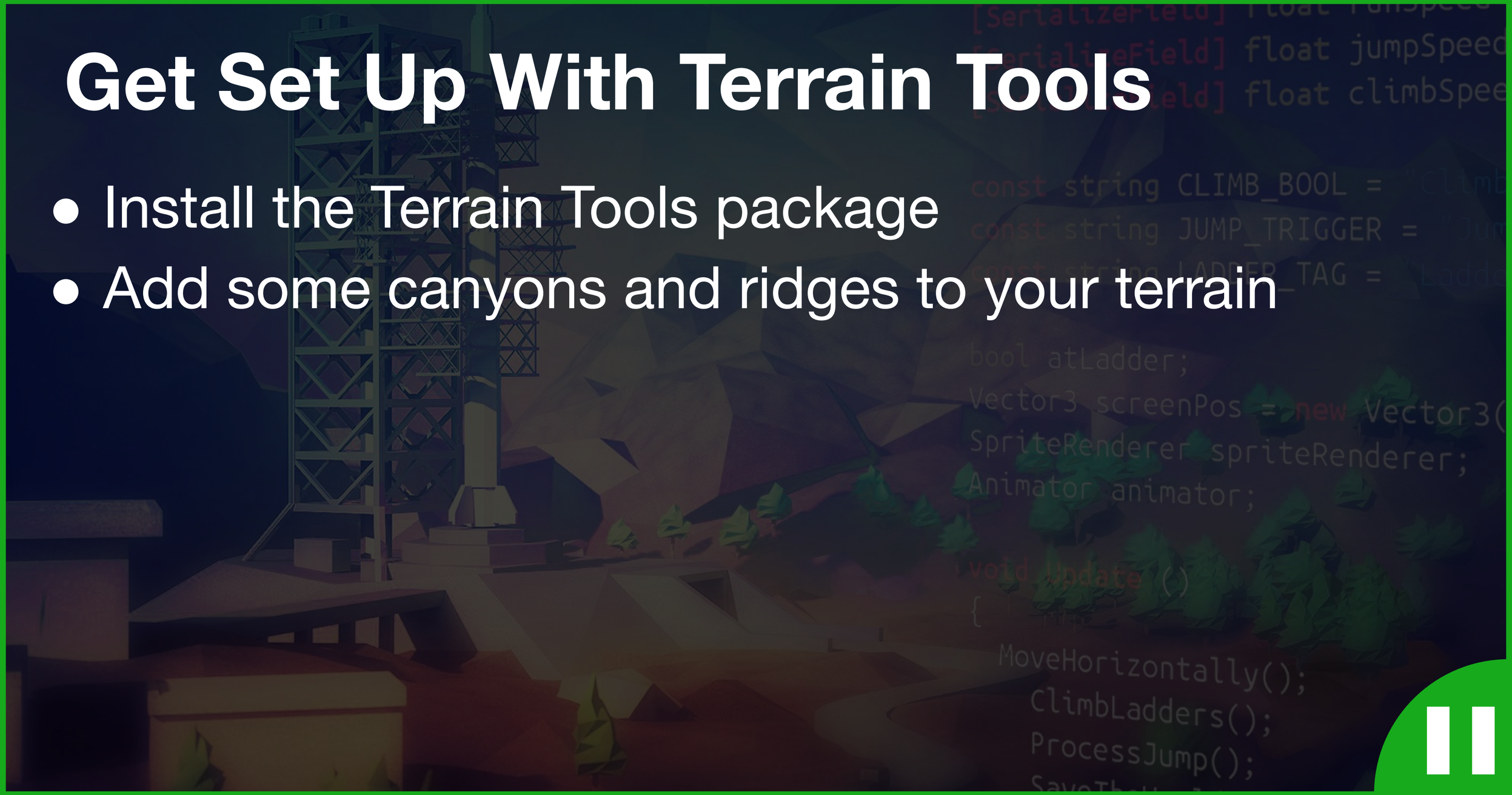
GameDev.tv

Unity 2020.1

Unity Terrain Tools

Get Set Up With Terrain Tools

- Install the Terrain Tools package
- Add some canyons and ridges to your terrain



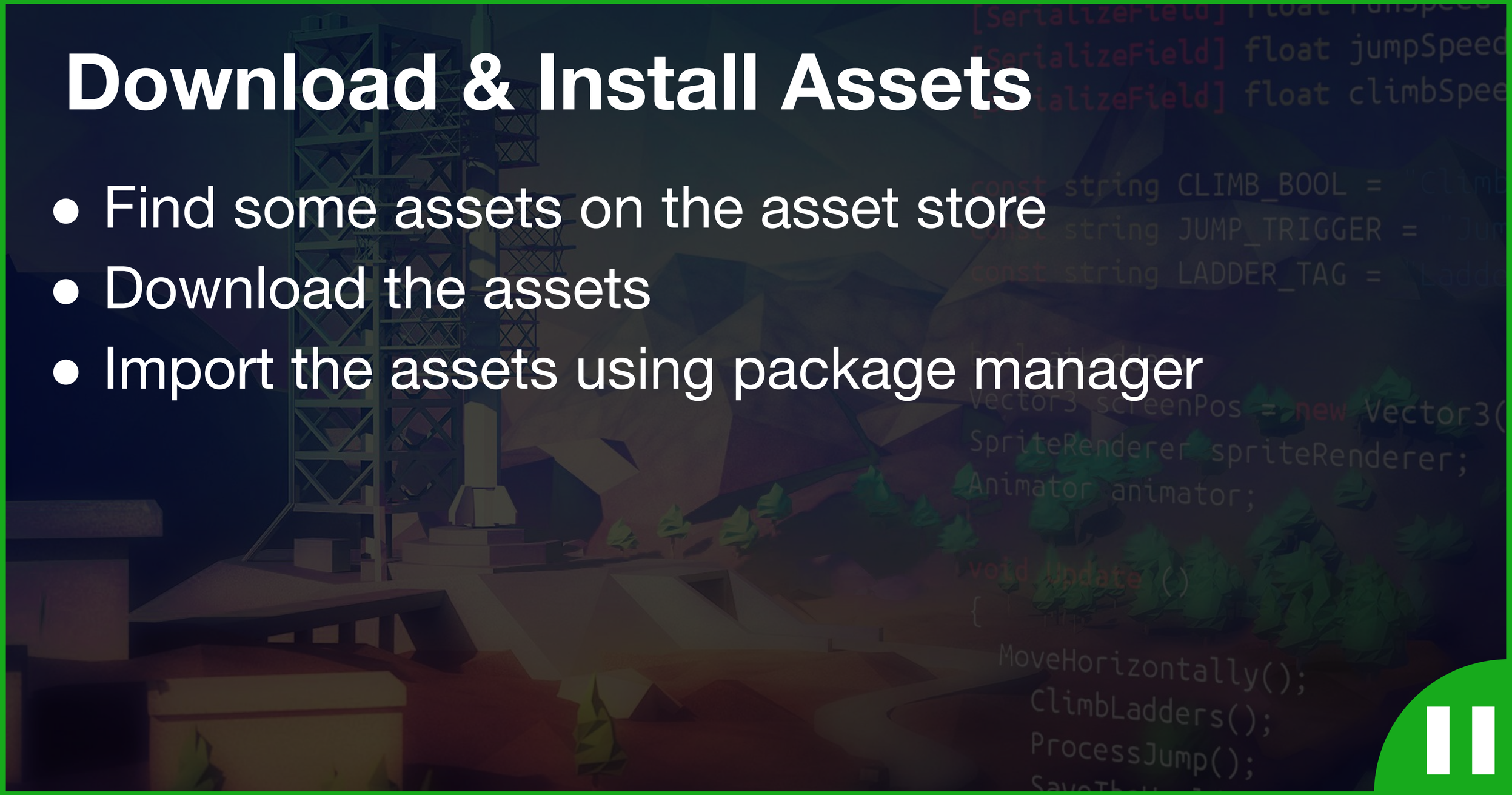
GameDev.tv

Unity 2020.1

How To Use Unity Asset Store

Download & Install Assets

- Find some assets on the asset store
- Download the assets
- Import the assets using package manager

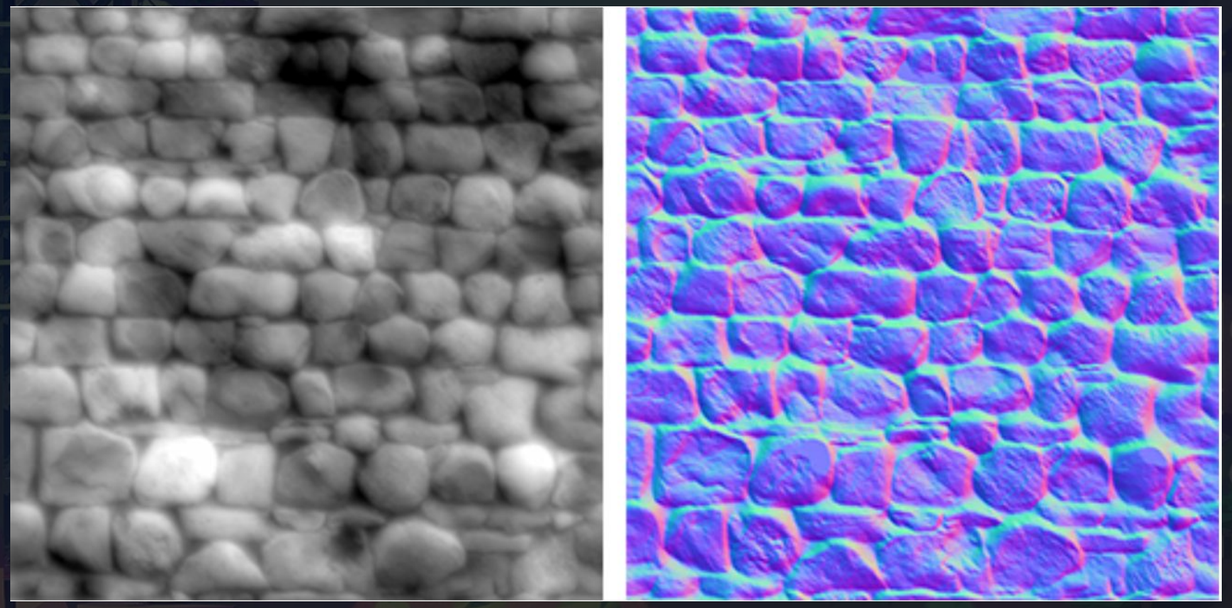


GameDev.tv

Unity 2020.1

Texturing Terrain In Unity

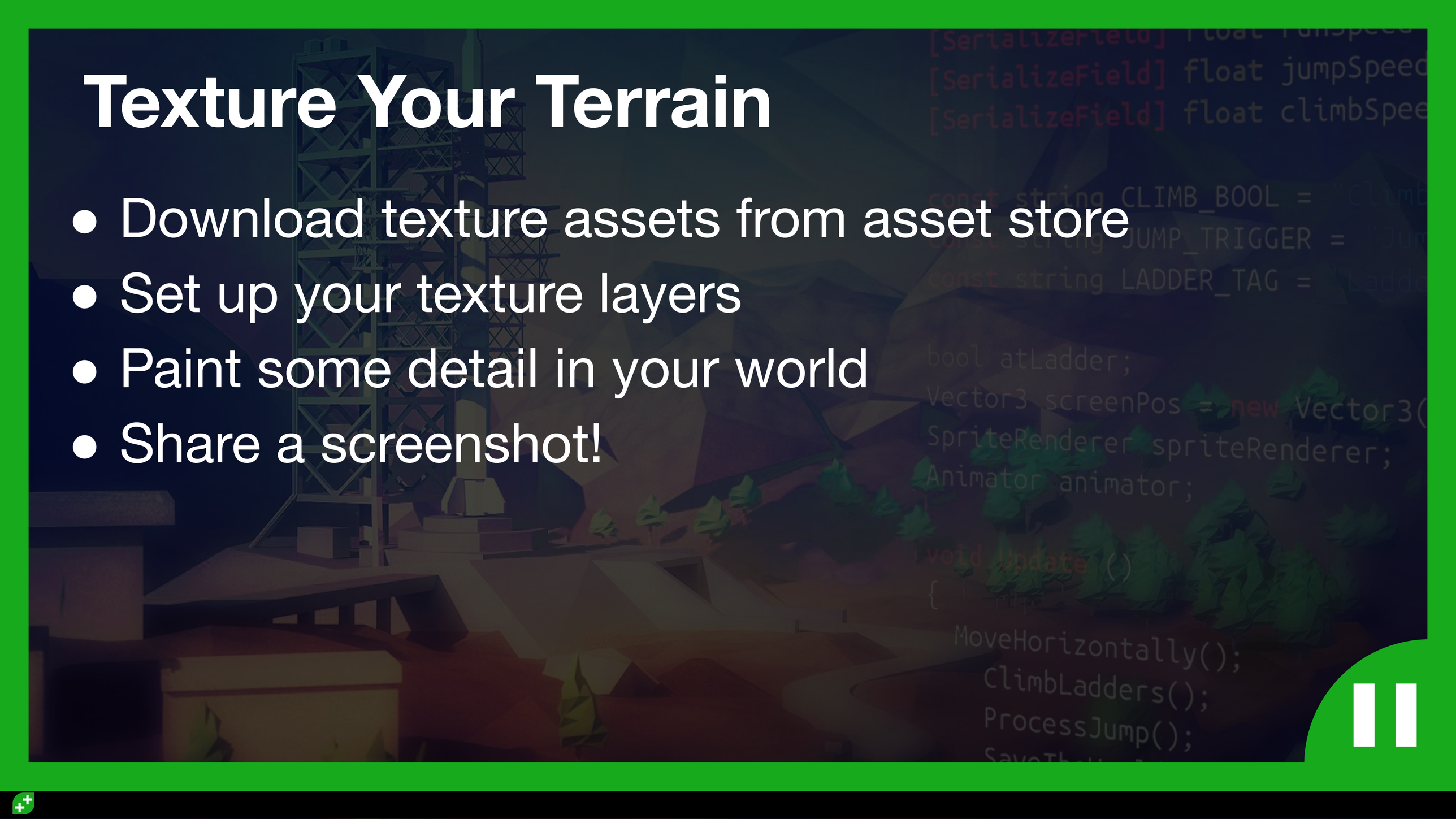
Bump Map, Height Map, Normal Map



- **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 Your Terrain

- Download texture assets from asset store
- Set up your texture layers
- Paint some detail in your world
- Share a screenshot!



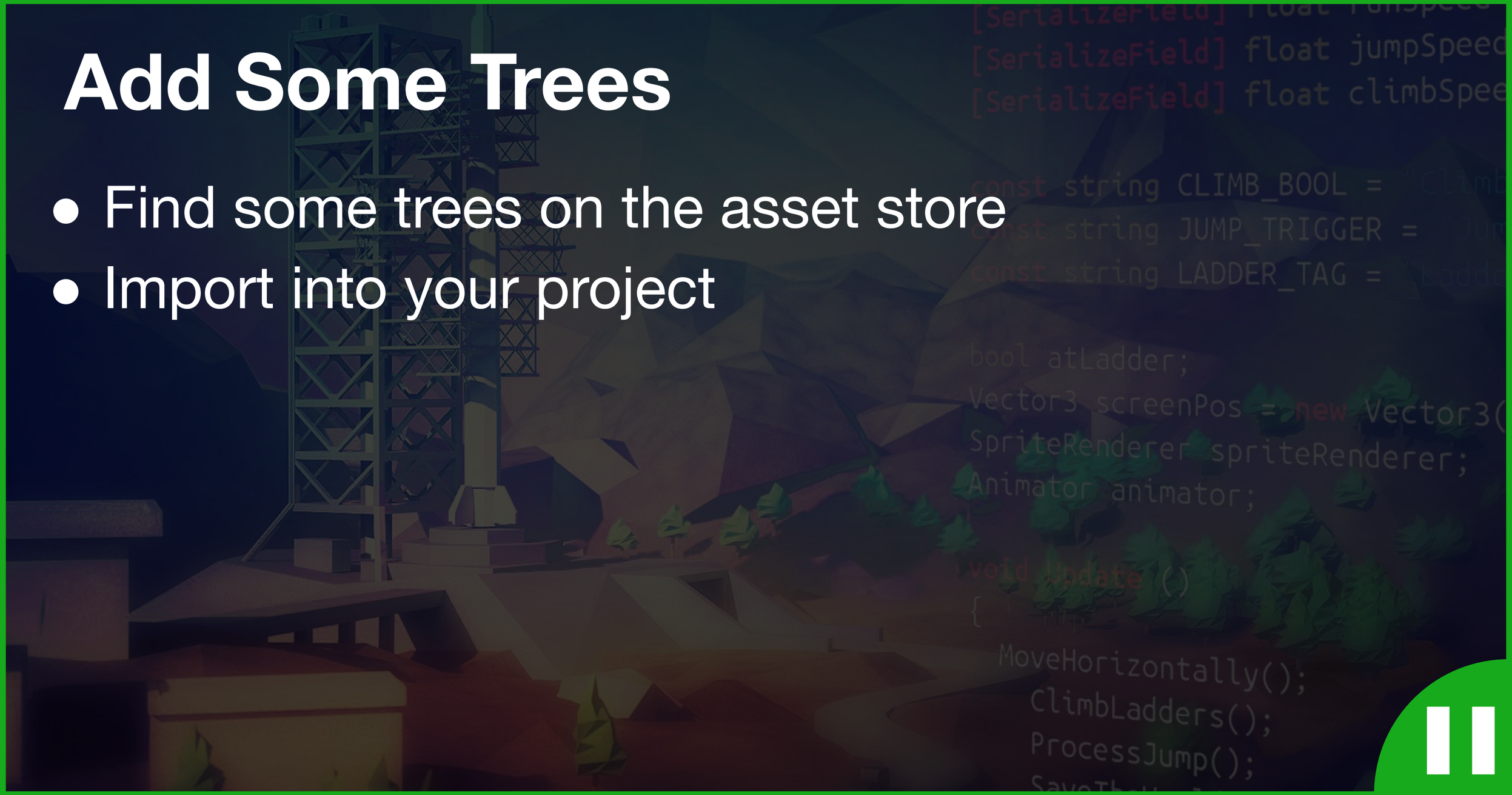
GameDev.tv

Unity 2020.1

Add Trees To Terrain

Add Some Trees

- Find some trees on the asset store
- Import into your project



GameDev.tv

Unity 2020.1

Master Timeline For Player Rail

Create Your First 5 Seconds

- Add Master Timeline
- Animate your player rig
- Adjust your grid colours if need be
- Create 5 seconds of your rail

GameDev.tv

Unity 2020.1

Animate Enemy Using Timeline

Create An Enemy Path

- Add an enemy placeholder (I’m adding a sphere)
- Using Timeline, animate that enemy so that the player narrowly misses it (and could shoot it)

GameDev.tv

Unity 2020.1

Import Player Ship Asset

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.

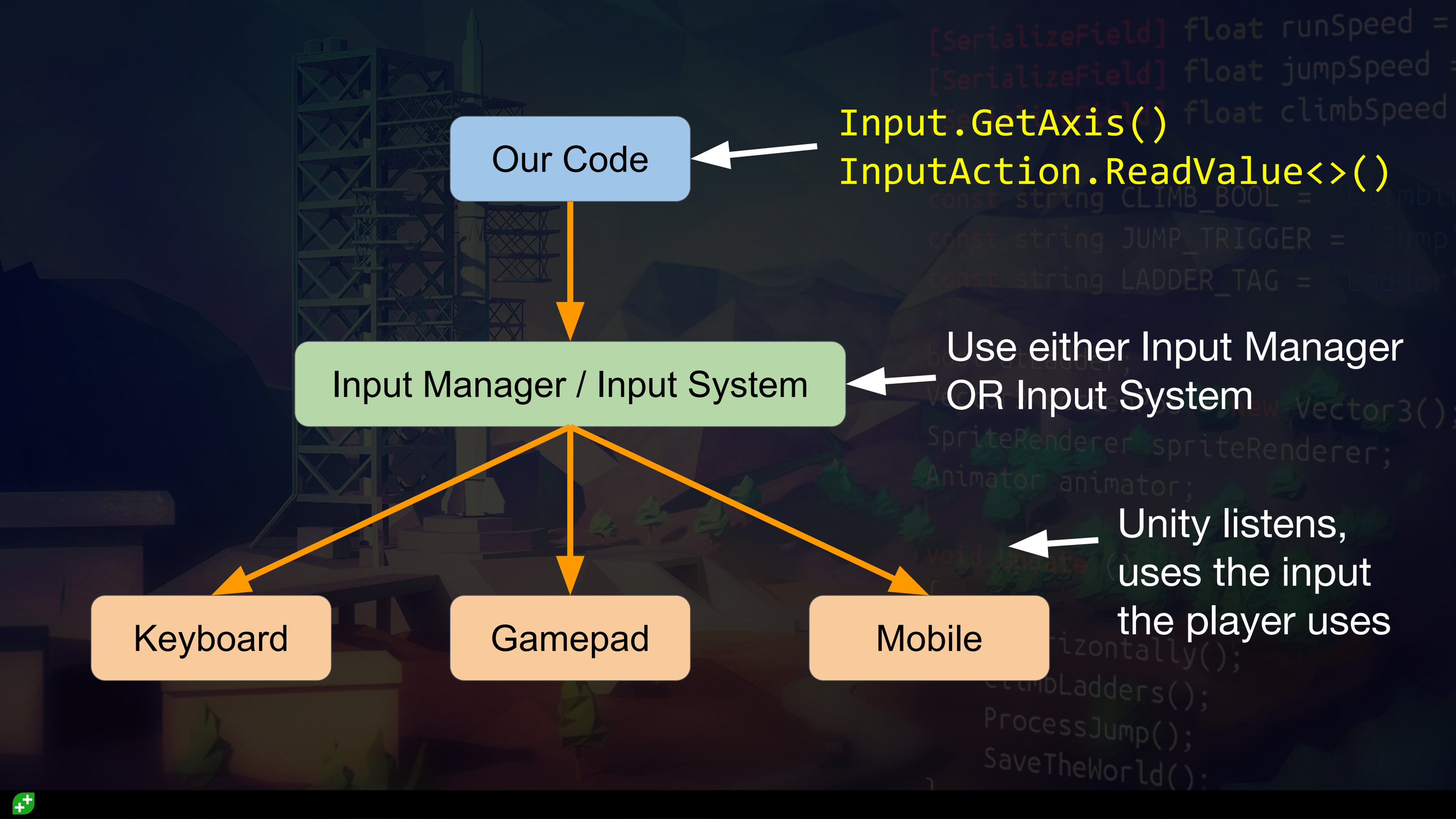
GameDev.tv

Unity 2020.1

Using GetAxis() For Movement

Unity Input System

- I'll be showing you 2 different Input systems that Unity has available
- We only need to use one system for our game
- I'll be using the "old" system throughout this section because the "new" system isn't yet stable
- Our code *should* allow you to easily use the new system as it becomes more stable
- Feel free to use the new system if you prefer


Our Code
Input.GetAxis()
InputAction.ReadValue<>()
Input Manager / Input System
Use either Input Manager
OR Input System
Keyboard
Gamepad
Mobile
Unity listens,
uses the input
the player uses

Add The Same For The Vertical Axis

- Following the same flow, print out to the console when the player is pushing up or down.

GameDev.tv

Unity 2020.1

Unity's New Input System

GameDev.tv

Unity 2020.1

Start Moving Our Player

Move The Player To The Right

- Focus just on X axis for this lecture
- We'll use localPosition when moving player
- We need to add 2 things:
  - Where the player is located right now
    +
  - How far we'd like to move it this frame
- Then we need to say, "move to this new position"

Move The Player To The Right

- Change our new Vector3 so that the ship is moving slowly to the right.
- In other words: Current position + an x offset

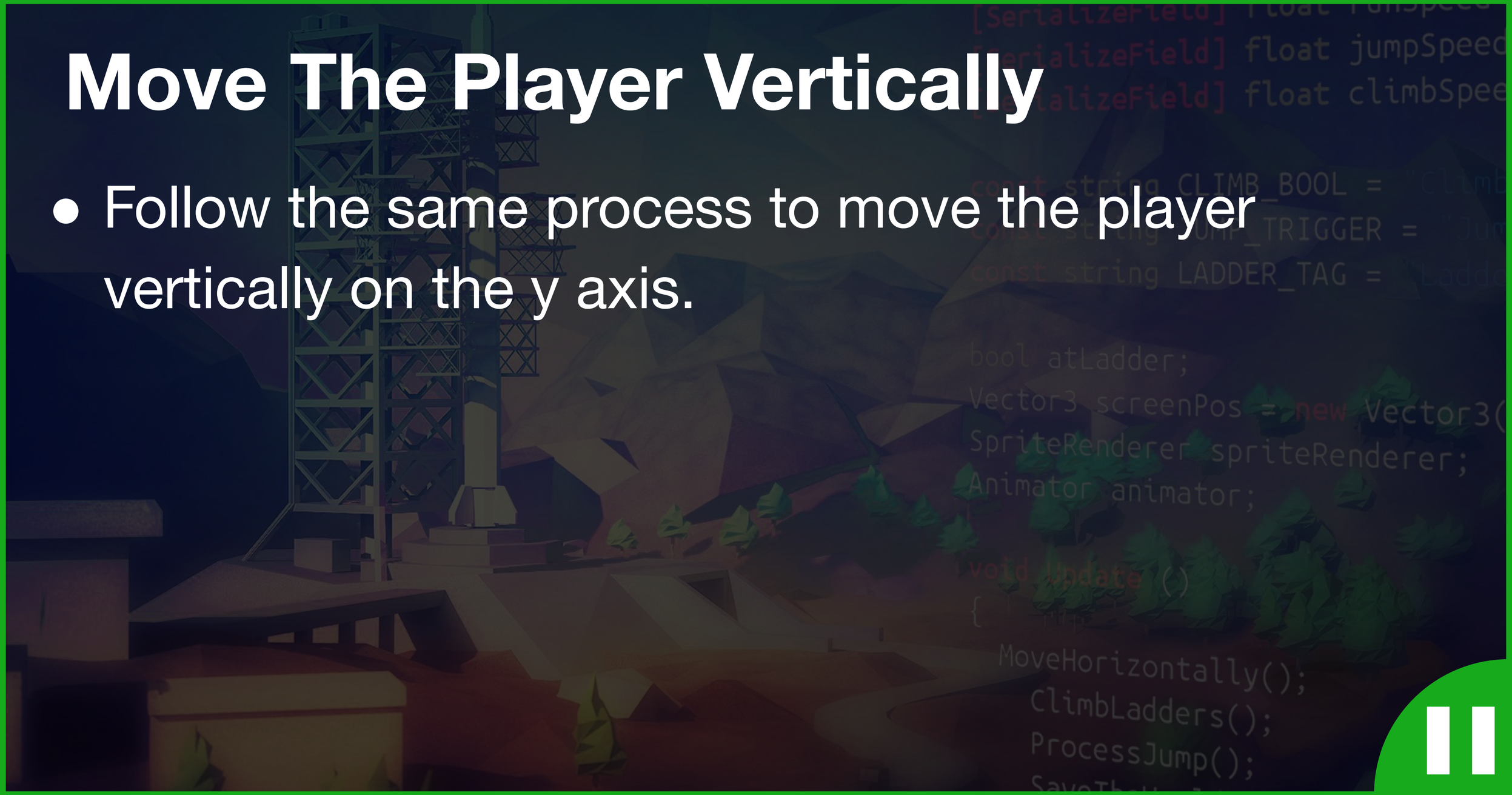
GameDev.tv

Unity 2020.1

Move Player Using Input

Move The Player Vertically

- Follow the same process to move the player vertically on the y axis.



GameDev.tv

Unity 2020.1

Mathf.Clamp() To Constrain Movement

Apply The Same Logic For Vertical

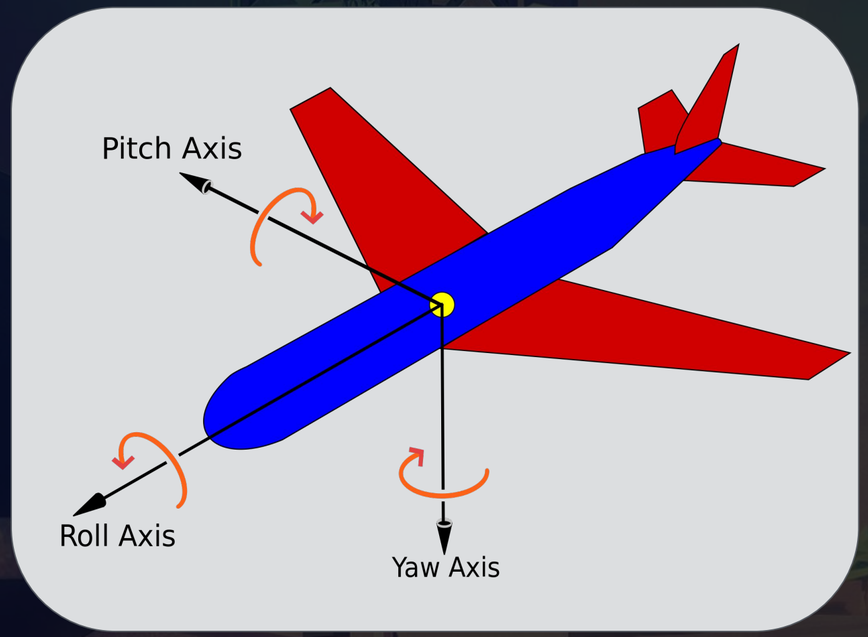
- Clamp our player movement on the Y axis.

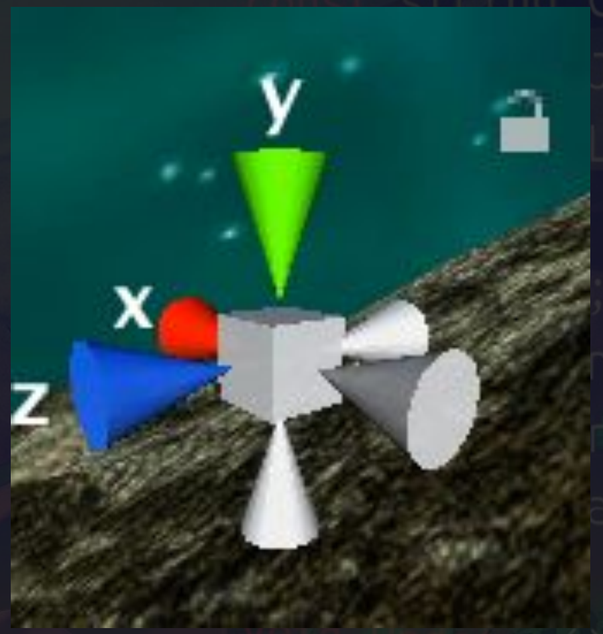
GameDev.tv

Unity 2020.1

How To Set Local Rotation

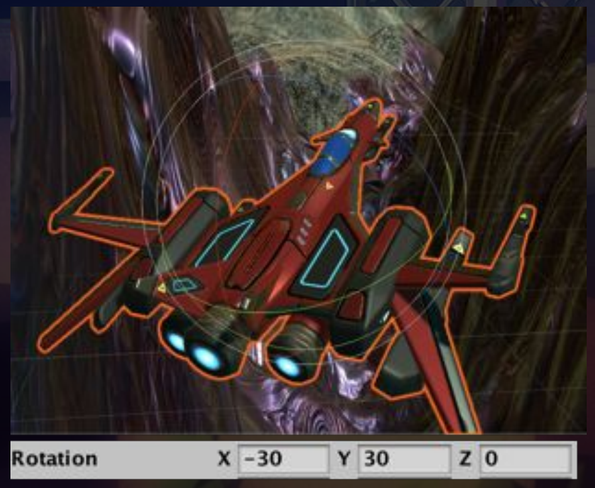
Roll, Pitch and Yaw



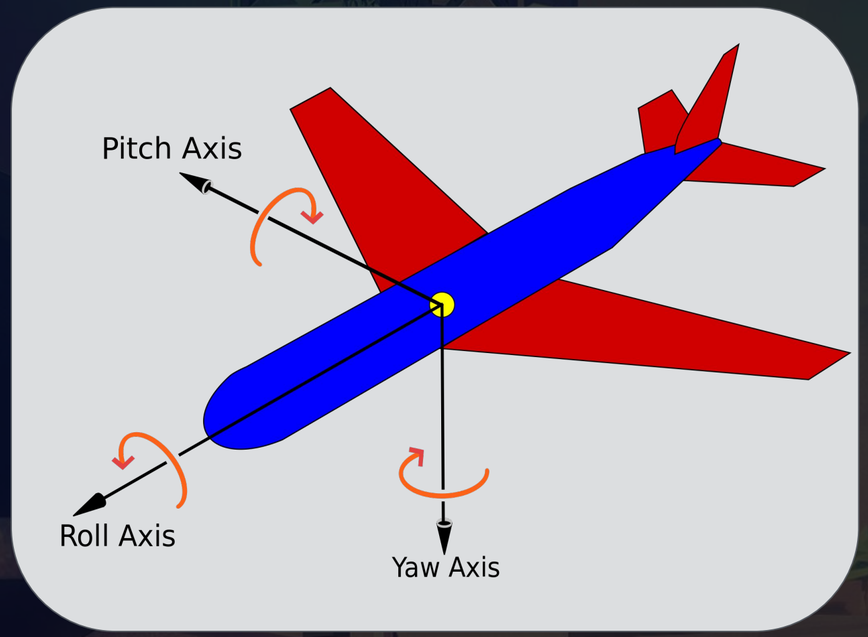


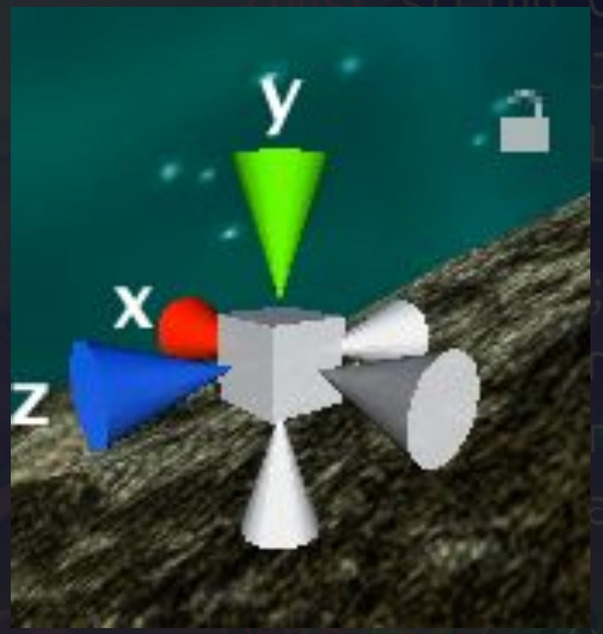
Explore Rotations

- 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);
```

GameDev.tv

Unity 2020.1

Rotate Ship With Position & Throw

Our Rotation Couplings

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

Finish Our Rotation Scheme

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

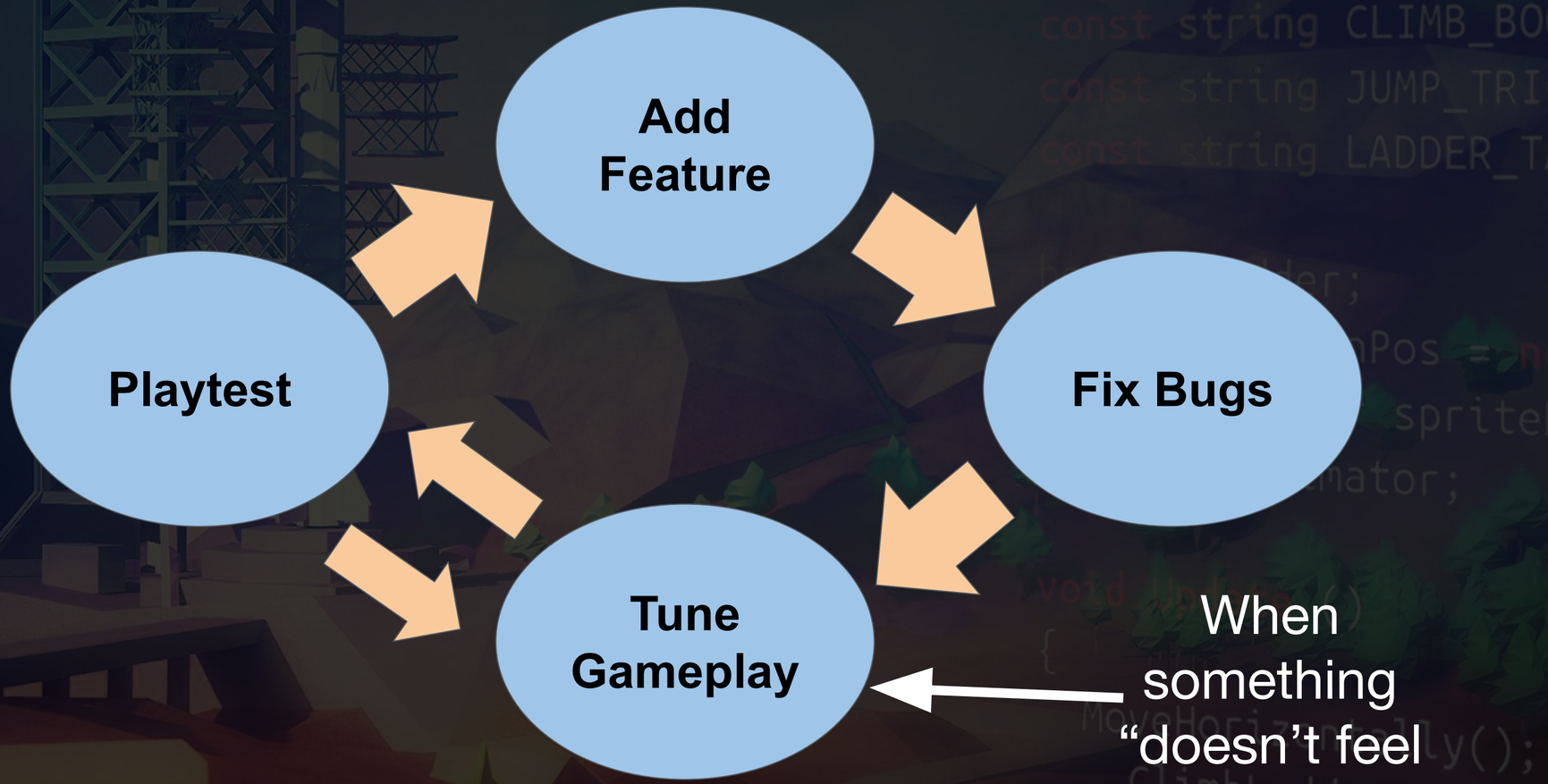
- Complete our rotation scheme as per the table
- Your parameters should feel good.
- Show off your work, maybe a 20s video.

GameDev.tv

Unity 2020.1

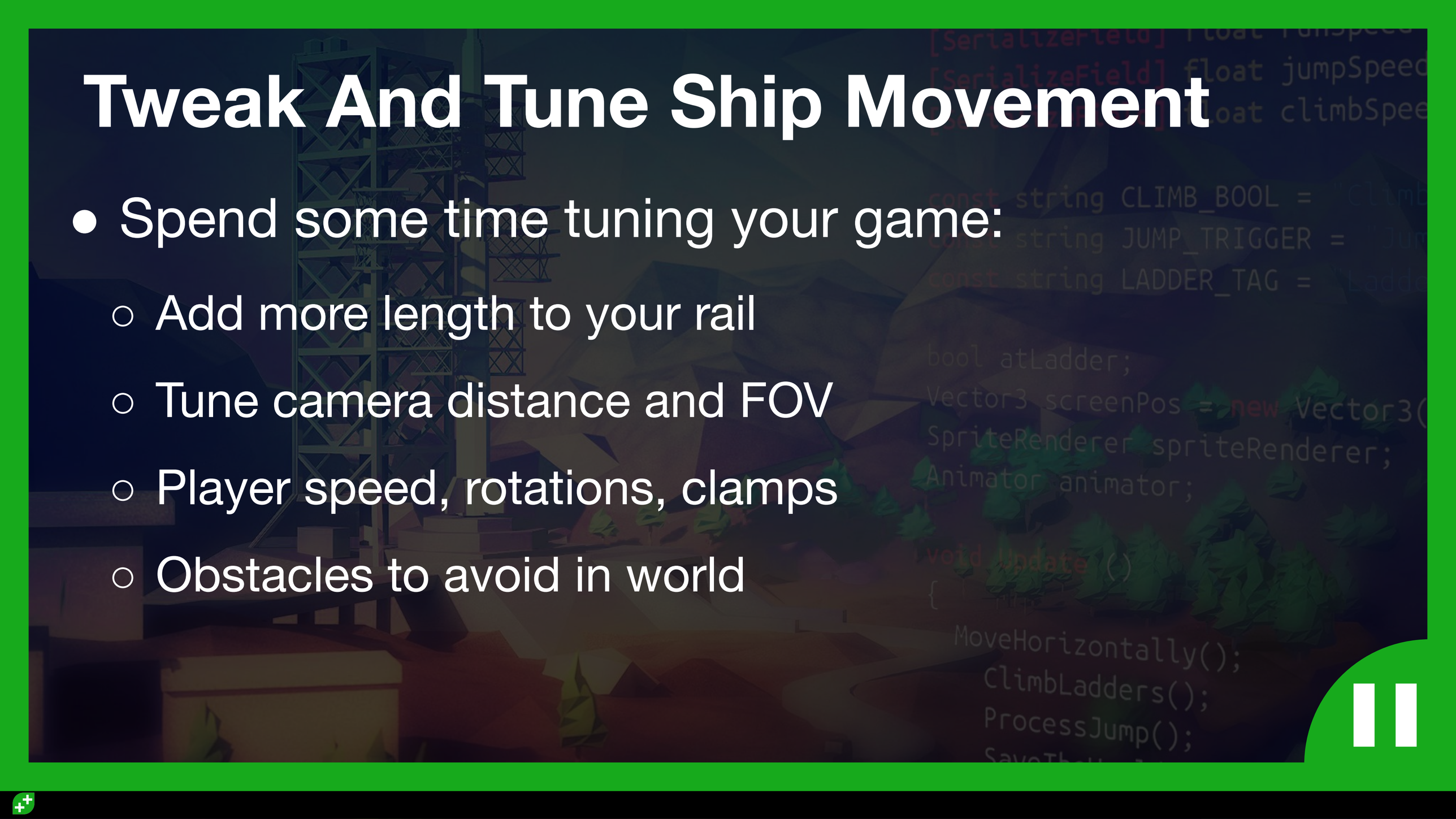
Time To Tune And Tweak

When To Tune Your Gameplay



Tweak And Tune Ship Movement

- Spend some time tuning your game:
  - Add more length to your rail
  - Tune camera distance and FOV
  - Player speed, rotations, clamps
  - Obstacles to avoid in world



GameDev.tv

Unity 2020.1

Particle System Laser Bullets

Particles System Component

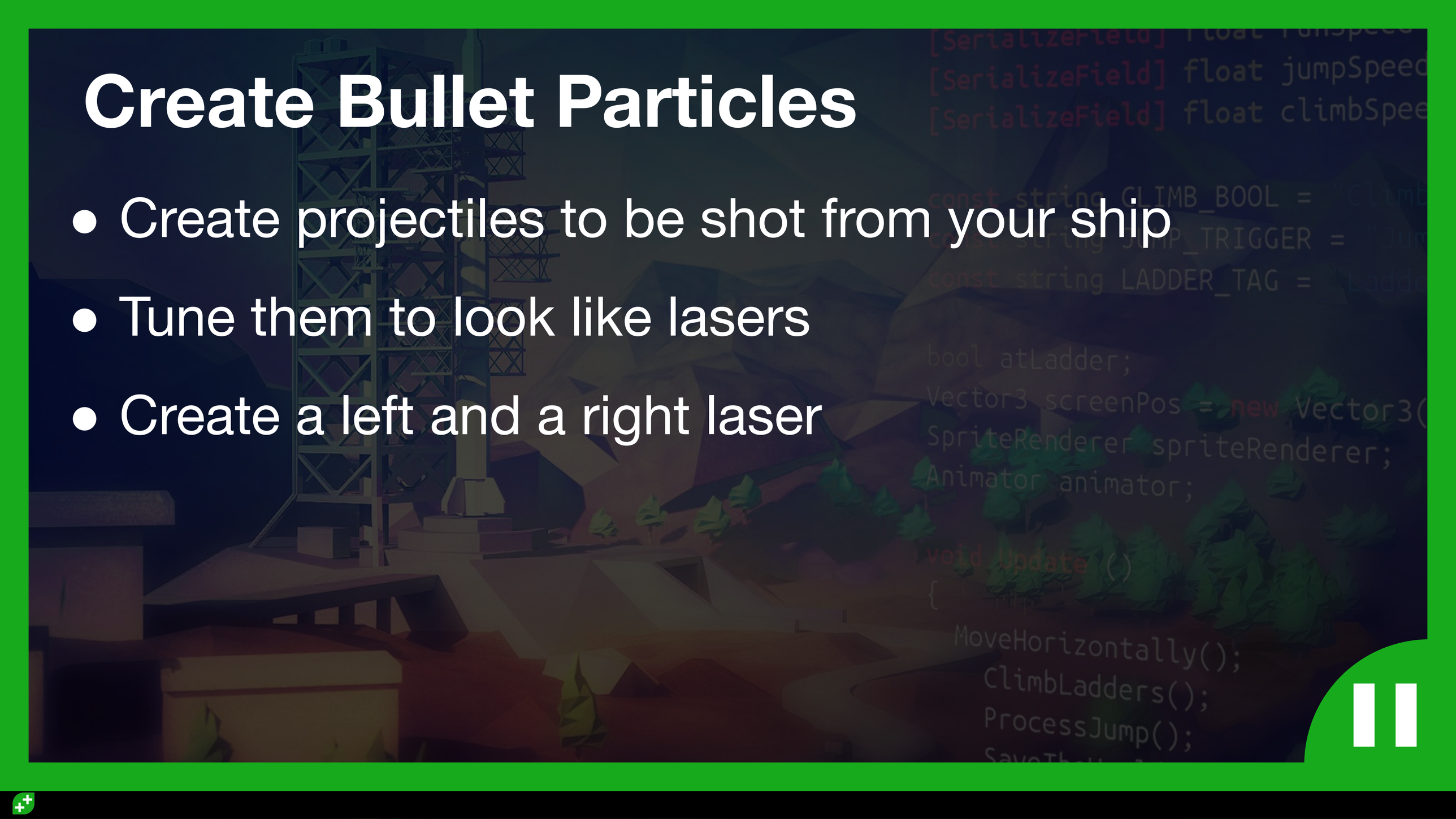
Particle System is a Component added to a Game Object

We use Modules for controlling behaviour

Each particle is not a Game Object

Create Bullet Particles

- Create projectiles to be shot from your ship
- Tune them to look like lasers
- Create a left and a right laser



GameDev.tv

Unity 2020.1

Exploring Nested Prefabs In Unity

Fix Your Nested Prefabbing

- We (me!) have created a bit of a mess
- Fix your nested prefabbing so each object is “owned” by the correct parent
- Set up your player ship so that editing the laser prefab will be updated within the game without issues

GameDev.tv

Unity 2020.1

Set Up Firing Input

Set Up Your Input System

- Using the old or the new input system (whichever is available to you), log to the console when the player pushes the fire button.

New Input System

```
[SerializeField] InputAction fire;

void OnEnable()
{
    fire.Enable
}
void OnDisable()
{
    fire.Disable
}
```

GameDev.tv

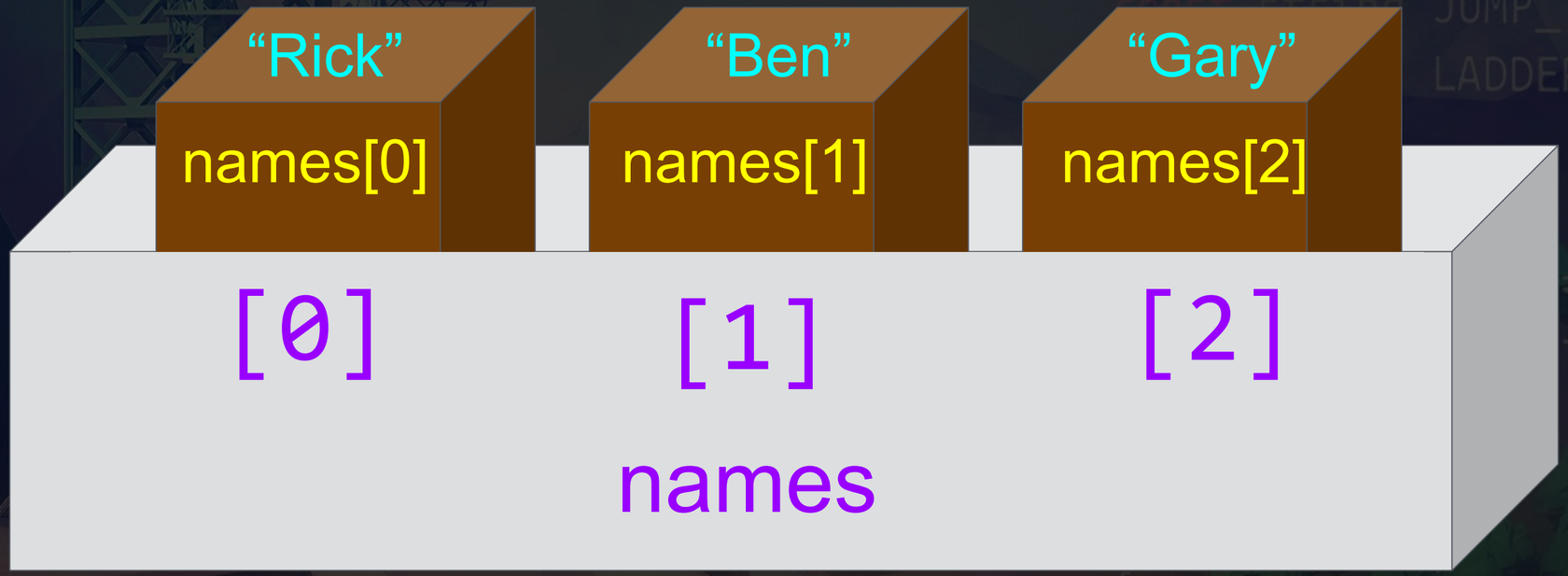
Unity 2020.1

Arrays & Foreach Loops

What Is An Array?

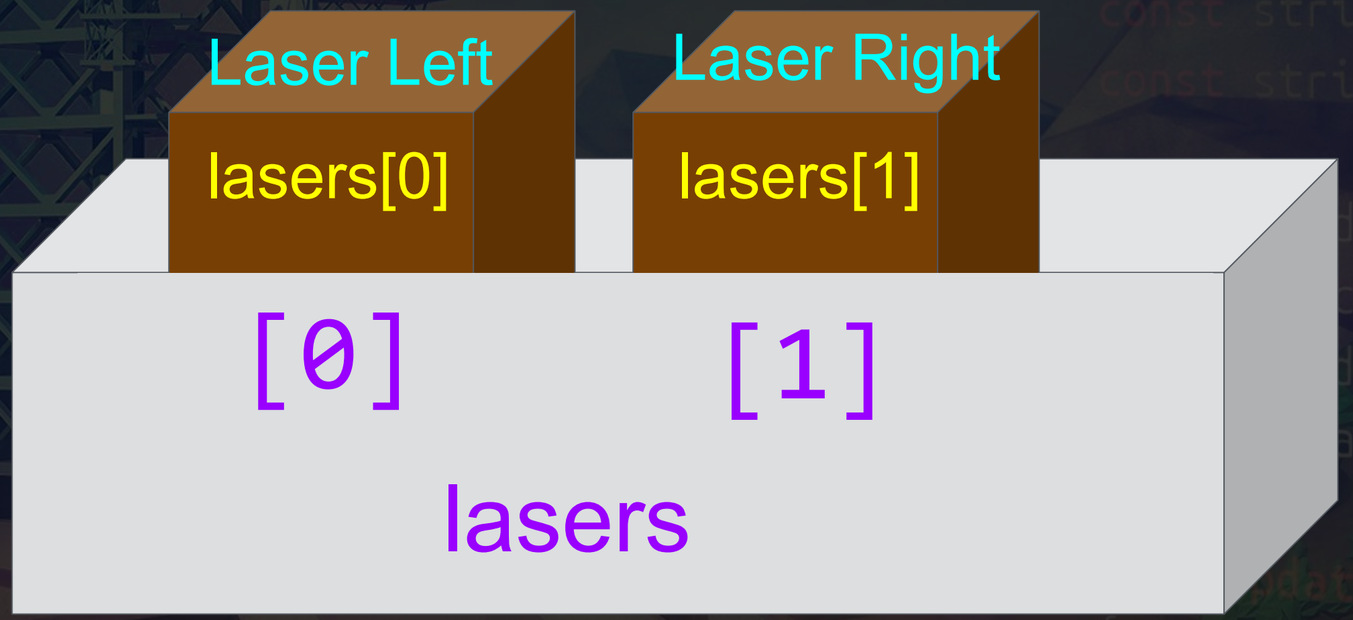
- If variables are like boxes, then arrays are boxes to store other boxes
  - Ie. Array allows us to store multiple objects in one variable
- We can only store the same Type of objects in the array variable

Arrays Are Boxes For Variables




```
string[] names = { "Rick", "Ben", "Gary" };
```

Array For Our Laser GameObjects




```
lasers[0] = Laser Left;
lasers[1] = Laser Right;
```

What Is A foreach Loop?

- Control flow statement for traversing a collection
- It’s a way of saying “do this to everything in our collection”

How to iterate over a collection

```
foreach (ObjectType item in things)
{
    item.DoSomething();
    // etc
}
```

Deactivate The Lasers

- When you stop firing, the lasers should stop
- When you start firing again, the lasers should start
- Yes, we are duplicating some code, we'll improve on that at a later point

GameDev.tv

Unity 2020.1

Deactivating Particle System Emission

Disable Particle System Emission

- Instead of enabling and disabling the entire Game Object, just enable or disable the emission module on the Particle System component

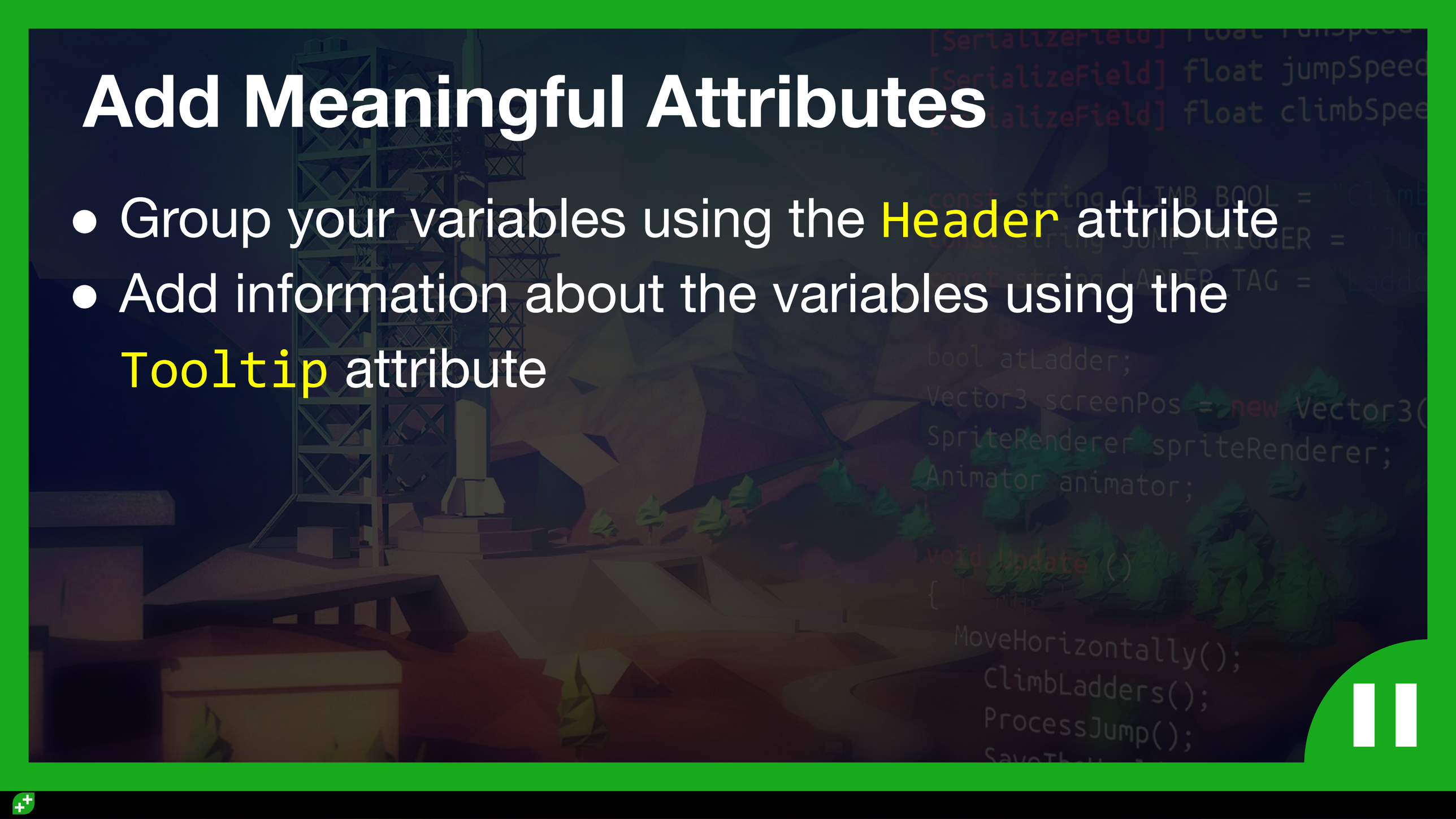
GameDev.tv

Unity 2020.1

Header & Tooltips Attributes

Add Meaningful Attributes

- Group your variables using the `Header` attribute
- Add information about the variables using the `Tooltip` attribute



GameDev.tv

Unity 2020.1

Understanding Collisions & Triggers

Prepare Debug Statements

- Print to the console when the player
  - Collides with something
  - Triggers something
- Bonus points: make things clearer by printing the Game Object names of the 2 things that had the collision or trigger event (eg. Player Ship bumped into Enemy)

	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

GameDev.tv

Unity 2020.1

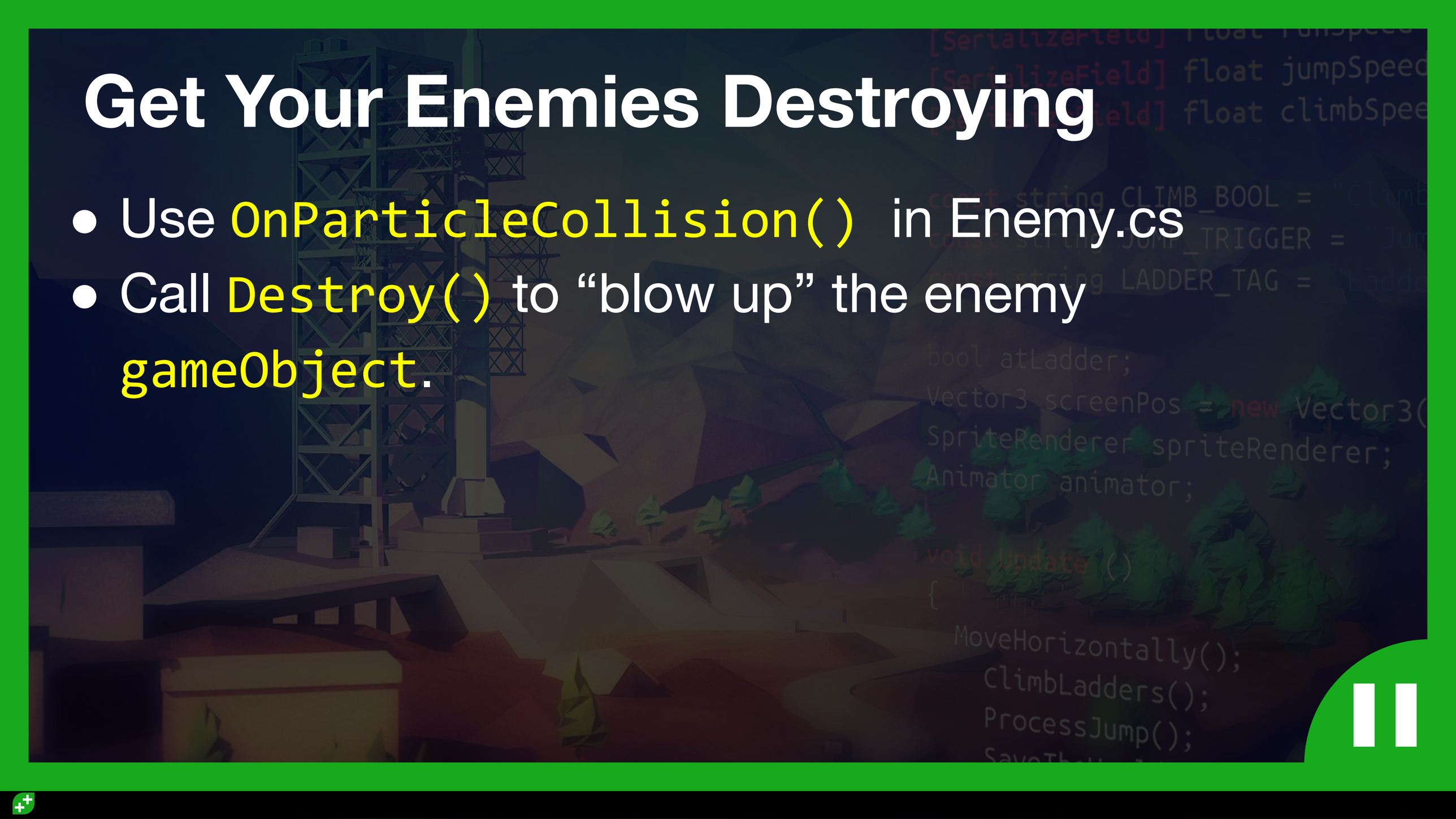
Detecting Particle Collisions

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

- Use `OnParticleCollision()` in Enemy.cs
- Call `Destroy()` to “blow up” the enemy `gameObject`.



GameDev.tv

Unity 2020.1

Reload Scene After Collision

Restarting...

- When the player crashes into something:
  - Disable player controls
  - Wait 1 second
  - Reload the level
- HINT: Looks at Project Boost repo

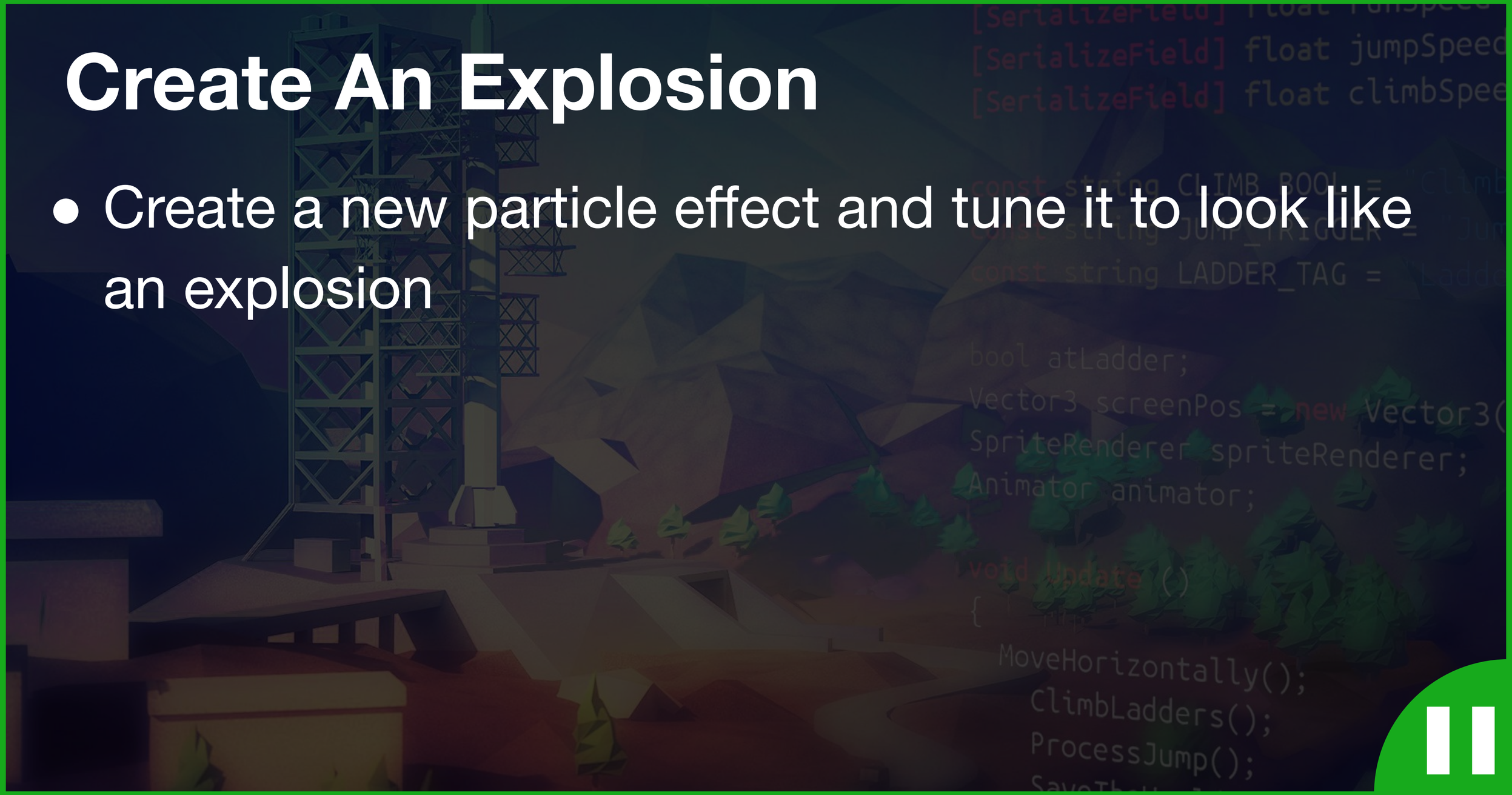
GameDev.tv

Unity 2020.1

Create Explosion Particle Effect

Create An Explosion

- Create a new particle effect and tune it to look like an explosion



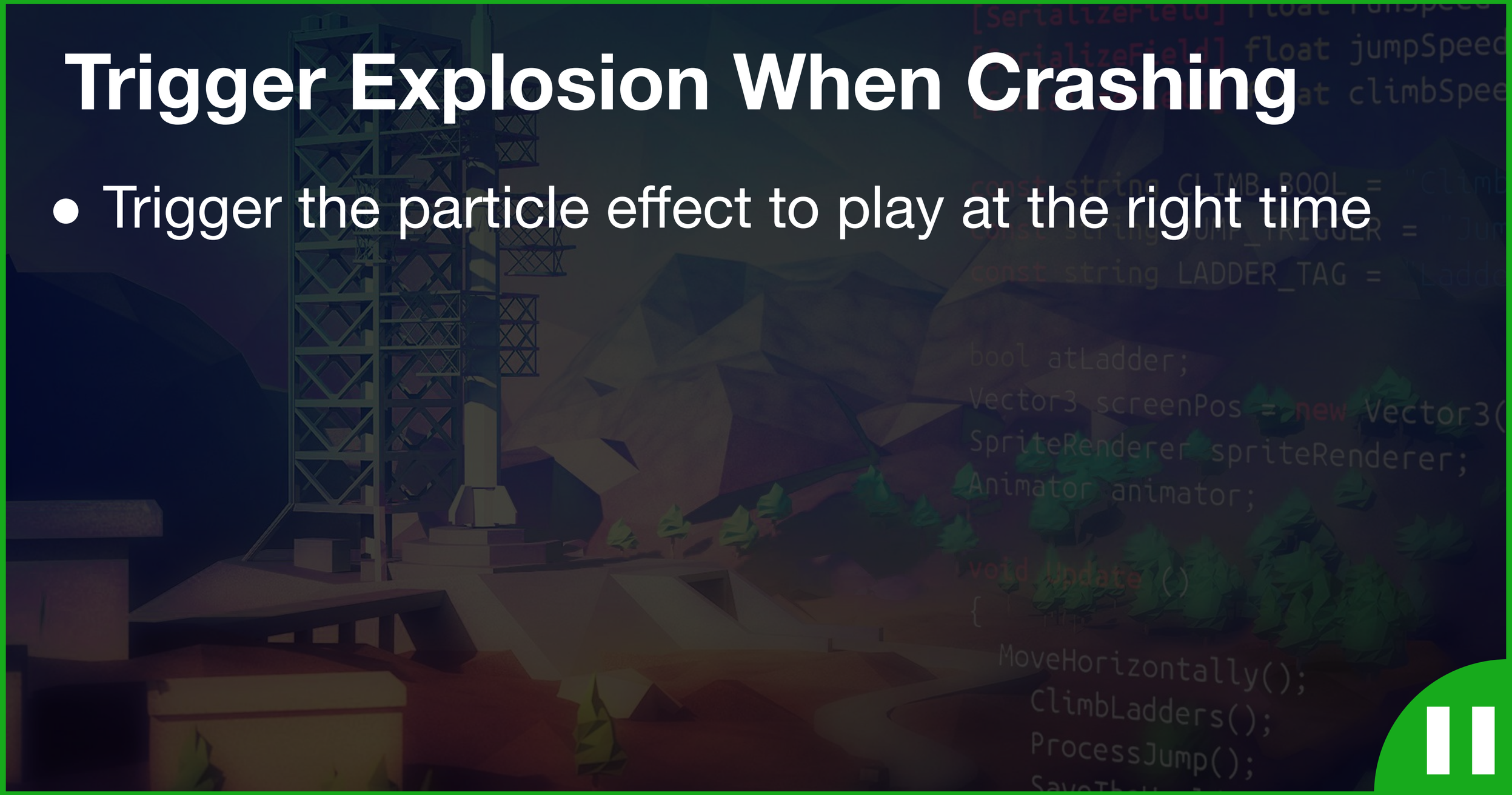
GameDev.tv

Unity 2020.1

Trigger Player Explosion

Trigger Explosion When Crashing

- Trigger the particle effect to play at the right time



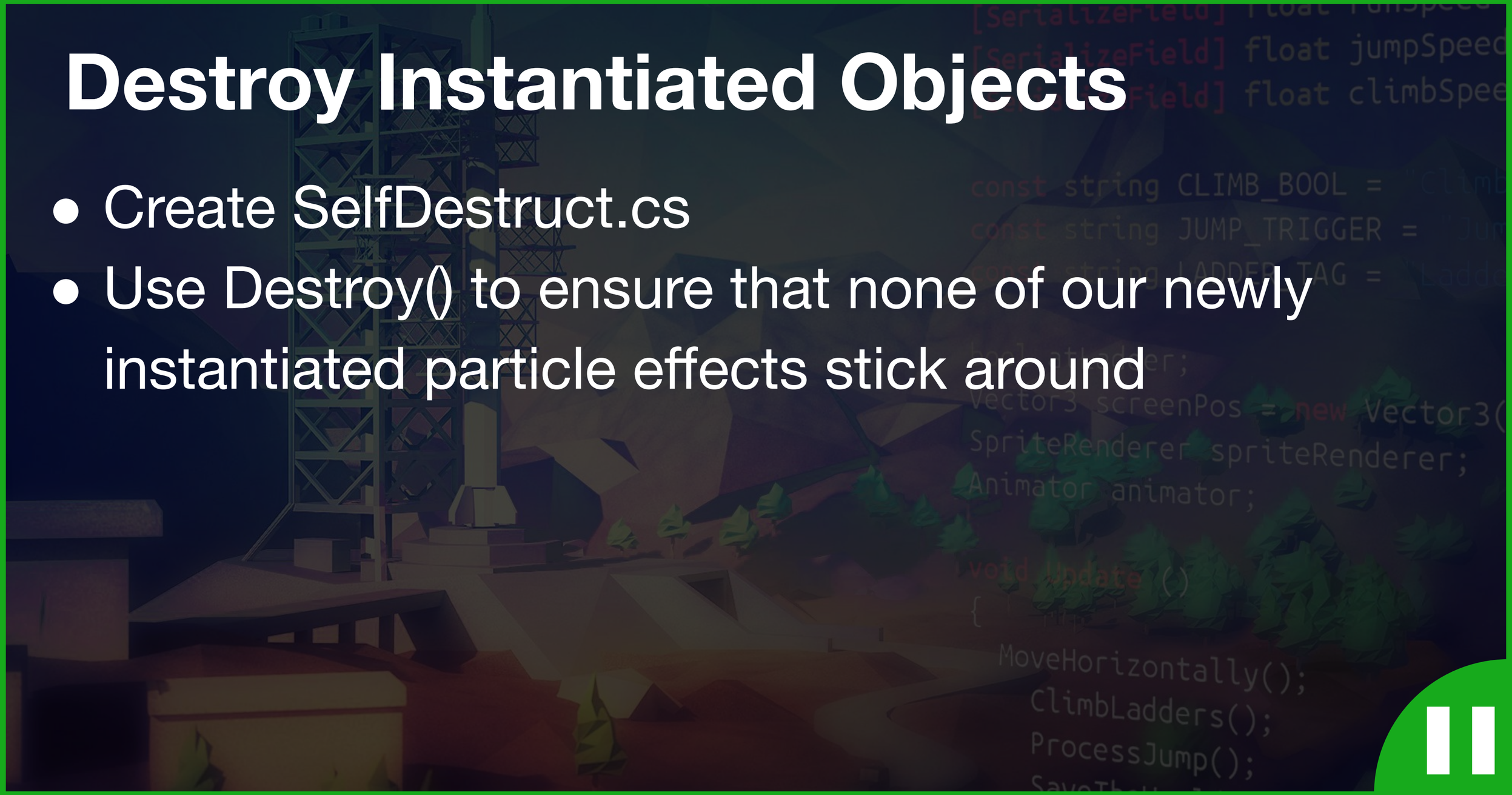
GameDev.tv

Unity 2020.1

Instantiate At Runtime

Destroy Instantiated Objects

- Create SelfDestruct.cs
- Use Destroy() to ensure that none of our newly instantiated particle effects stick around



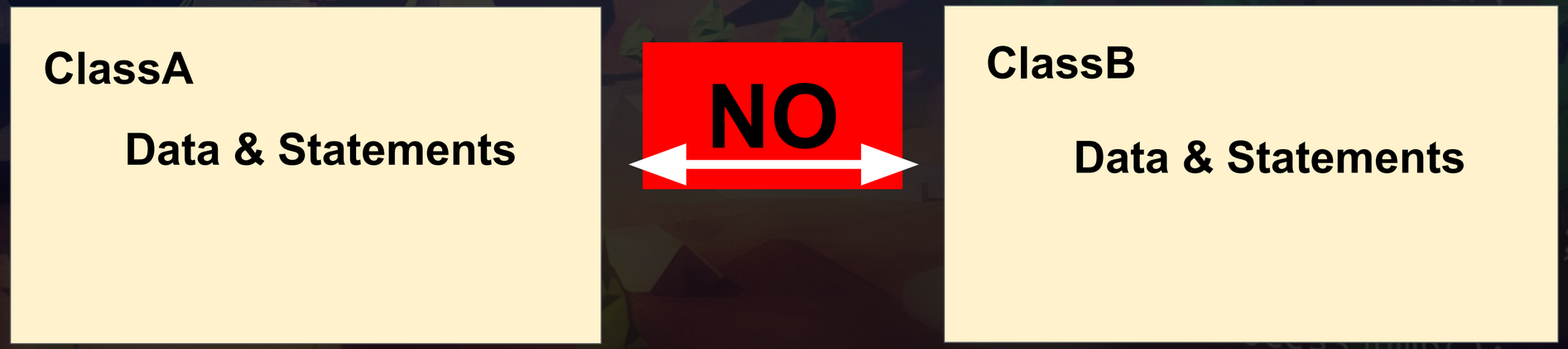
GameDev.tv

Unity 2020.1

Public Methods In Unity C#

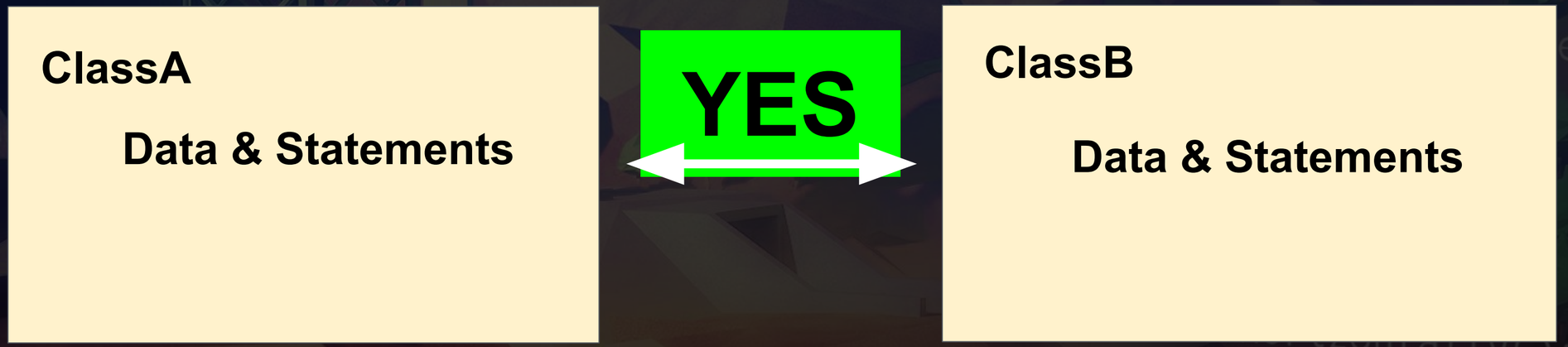
Remember Encapsulating Our Code?

- En-capsule-ating - putting in a capsule.
- Different parts of your code have a “need to know basis” level of access.
  - Ie. Don’t let everything access everything else.

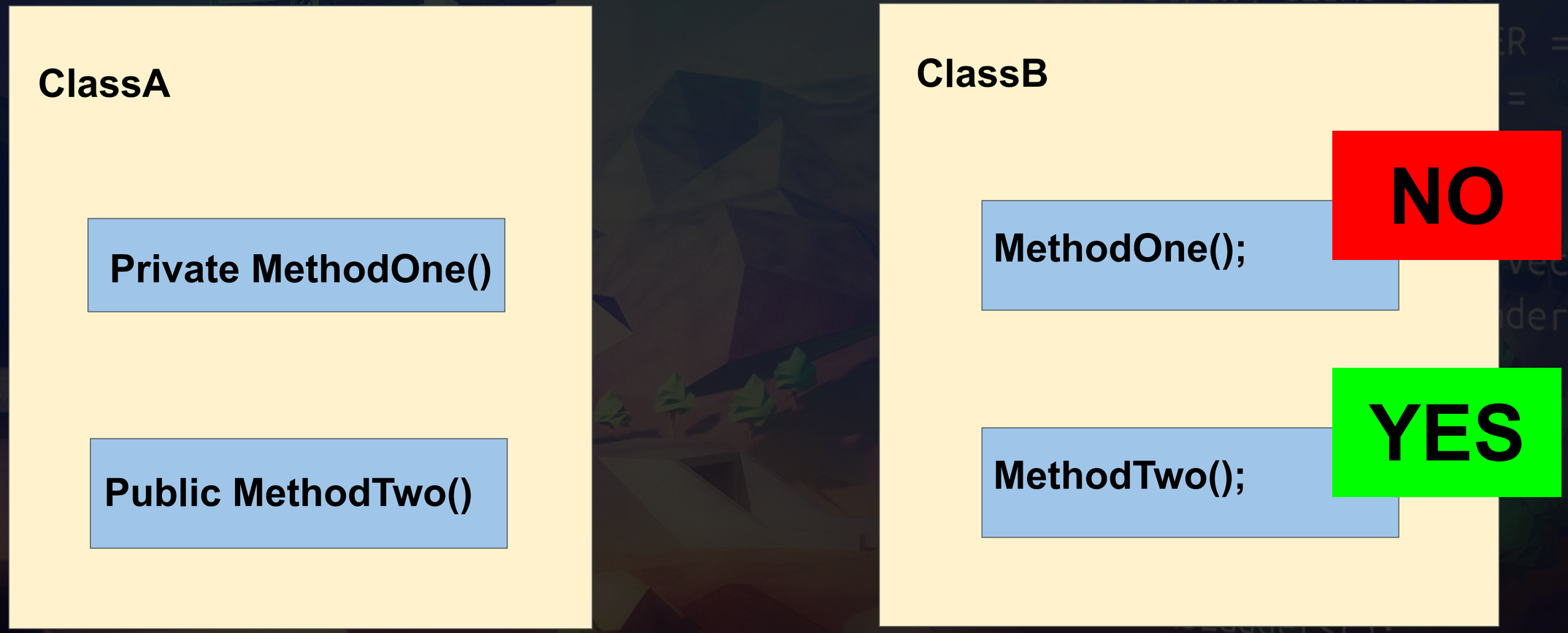


Remember Encapsulating Our Code?

- But now we'd like to let one class (Enemy) influence another class (ScoreBoard).



Public Versus Private Access Modifiers



Public Access Modifier

Access Modifier
Scope of use

Return type
void = return nothing

Method name
WHAT to do

Parameter
() = nothing

Increase Player Score

- Call our public method to increase the score
- Create a tuneable variable for how much to increase score per hit
- Print to the console the score each time it updates

GameDev.tv

Unity 2020.1

Simple User Interface For Score

Add A Simple UI

- Add your text element
- Pick a font
- Format color and size

GameDev.tv

Unity 2020.1

ToString() To Display Score

Show Our Actual Score

- Our text field requires a string to display properly
- `ToString()` is a method which converts other types of data into a String
- Use `ToString()` to show our updated score when shooting enemies

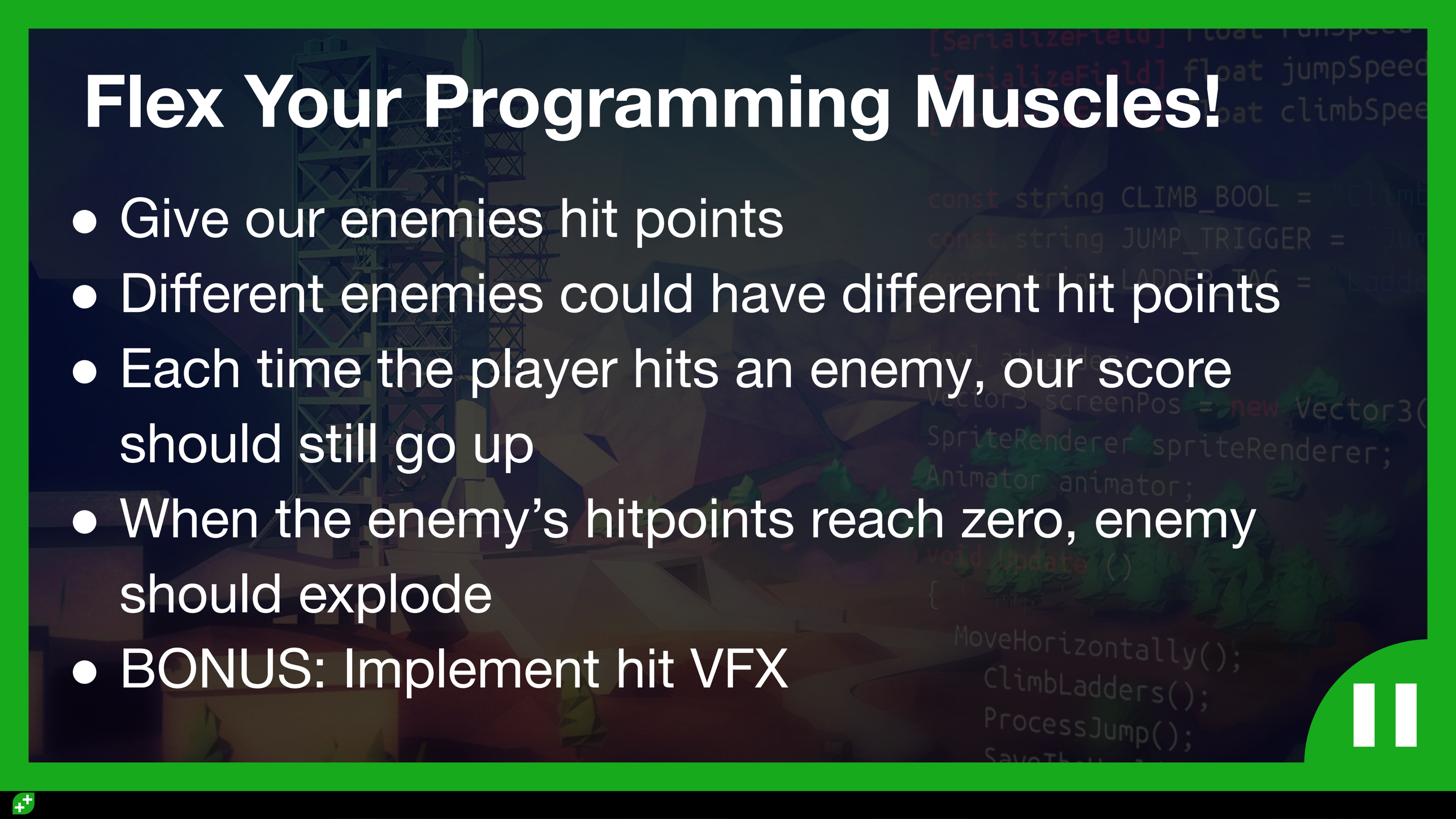
GameDev.tv

Unity 2020.1

Enemy Hit Points

Flex Your Programming Muscles!

- Give our enemies hit points
- Different enemies could have different hit points
- Each time the player hits an enemy, our score should still go up
- When the enemy’s hitpoints reach zero, enemy should explode
- BONUS: Implement hit VFX



GameDev.tv

Unity 2020.1

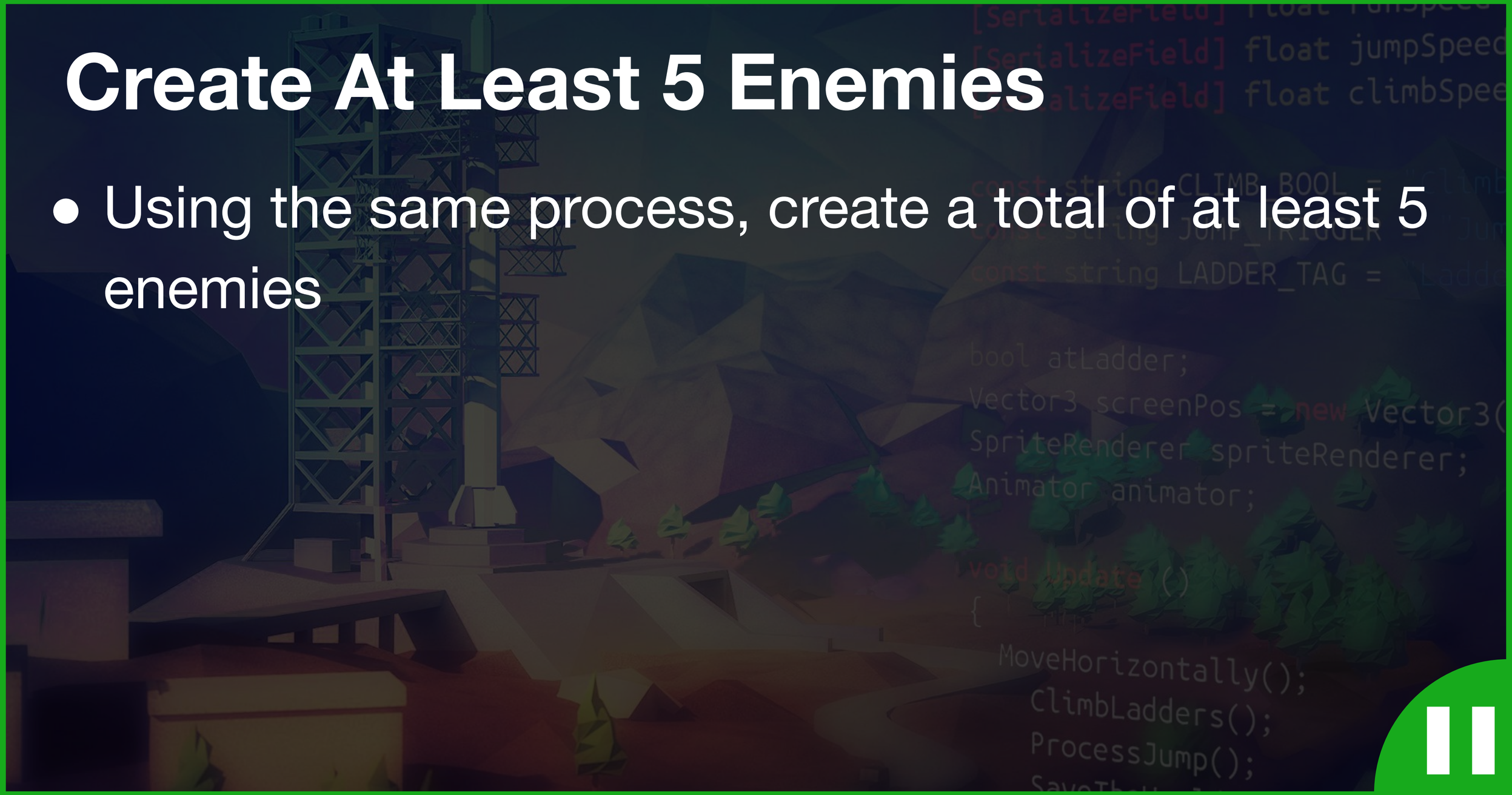
Set Up Enemy Prefabs

Add A RigidBody At Runtime

- Using `AddComponent<>()` add a RigidBody to each enemy game object when the game starts
- Bonus points: Ensure that Use Gravity is disabled

Create At Least 5 Enemies

- Using the same process, create a total of at least 5 enemies



GameDev.tv

Unity 2020.1

Using FindWithTag()

Set Parent Reference

- Create a new tag for our Spawn At Runtime game object
- Set the reference for our parent game object using `FindWithTag()`
- Update other areas of our code accordingly to make this work
- Note: Pay attention to ‘Type’

GameDev.tv

Unity 2020.1

Control Tracks For Enemy Waves

Set Up Enemy Waves

- Create at least 5 waves of enemies
- Add those waves to your timeline
- Tune the timing so that the player can shoot and dodge the enemies

GameDev.tv

Unity 2020.1

Timeline For Dialogue

Add Some Character Dialogue

- Either using my assets, or creating your own, place instructions or information for the player using Timeline.
- Share a screenshot or video of your game with us.

GameDev.tv

Unity 2020.1

Singleton Pattern For Music Player

Find Some Music For Your Game

- Search for royalty free music to add to your game.
- Try to match the player experience (eg. hectic experience = hectic music).

GameDev.tv

Unity 2020.1

Sneaky Explosion SFX

Adding Explosion SFX

- What is the simplest way you can think of to implement an explosion SFX?

GameDev.tv

Unity 2020.1

Skybox & Fog

Grab A Skybox

- Head over to the asset store and grab a skybox asset which matches your game.
- If you don’t want to or can’t get to the asset store, you can just use the procedural skybox.

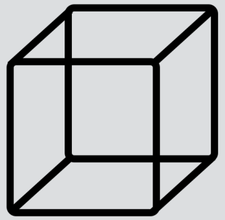
GameDev.tv

Unity 2020.1

Unity Post-Processing

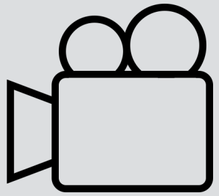
Post Processing

- Add effects to your Camera to change the look of your game



Volume

Where in the world



Layer

Which camera triggers PP

Profile

Specific settings for effects

Experiment With Post Processing

- Add each of the post processing effects to see if it gives your game the feel you want.
- Take a screenshot and share in the community forum.

GameDev.tv

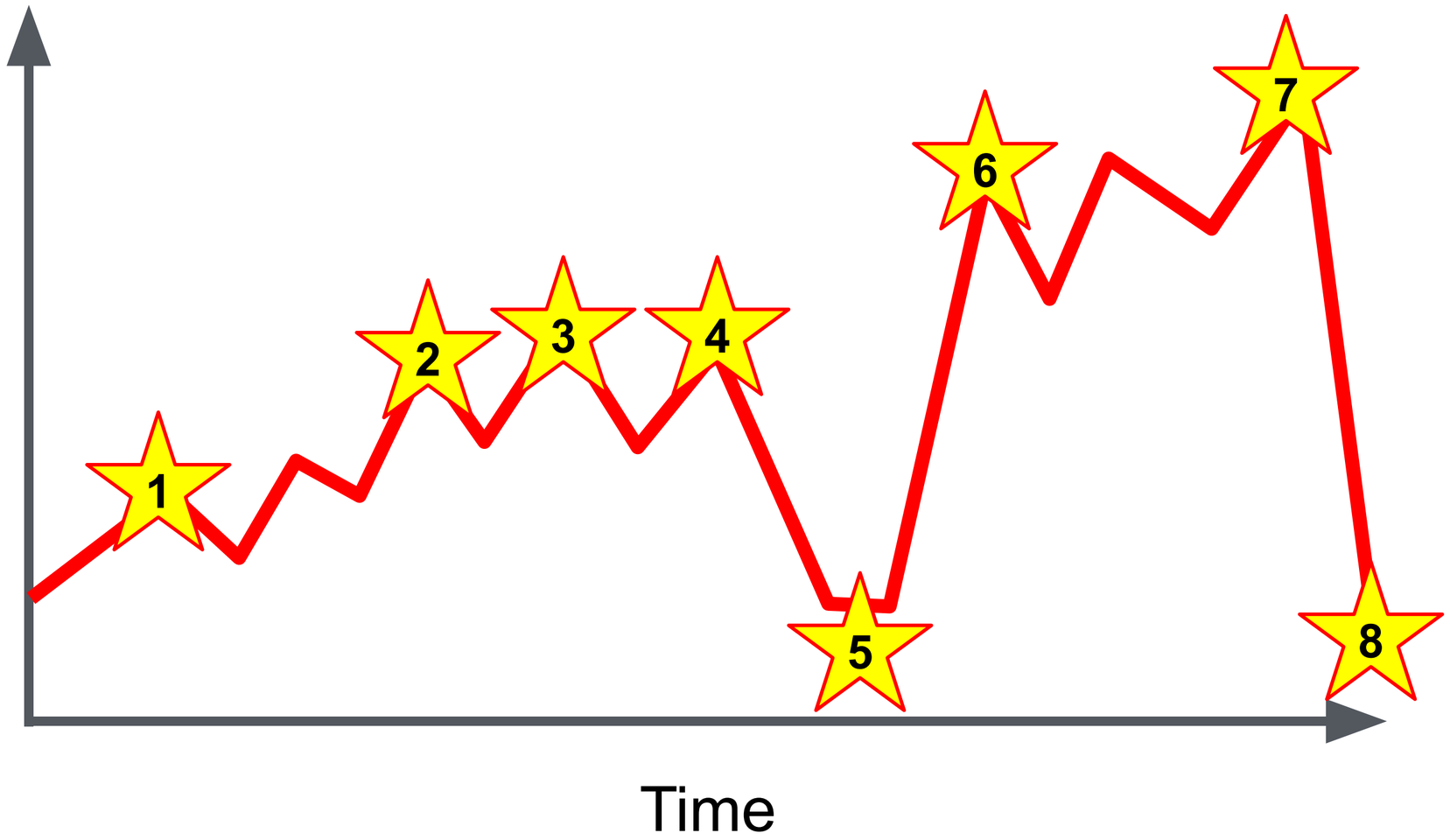
Unity 2020.1

Your 3 Minute Experience

Creating Your 3 Minute Experience

- Your level should be a roller coaster with moments from 1 to 10 in intensity
- Consider your “beat chart”...

Level Design Beat Chart



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.

Creating Your 3 Minute Experience

- Variety is king:
  Each 10 second moment should be unique
- Always be thinking of your player experience
- Give your player choices

Variety

- There are thousands of combinations of variables to create variety:
  - Enemies: appearance, size, speed, hit points, movement patterns, FX, score
  - Player rail: speed of movement, where you fly, rotation, movement patterns
  - Environment: Textures, shapes, trees, obstacles, explosions, colours, lighting, visibility
  - Player: Laser speed and colour, controls, camera

Create Your 3 Minute Experience

- Take the time now to create a three minute experience.
  - Create a player path you're happy with
  - Add enemies and enemy waves
  - Add voice overs / characters / story
  - Tune the game so that its the right difficulty
- Create a build of your game, upload to sharemygame.com and share with your friends and with our community

GameDev.tv

Unity 2020.1

Wrap Up - Argon Assault


[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();
}