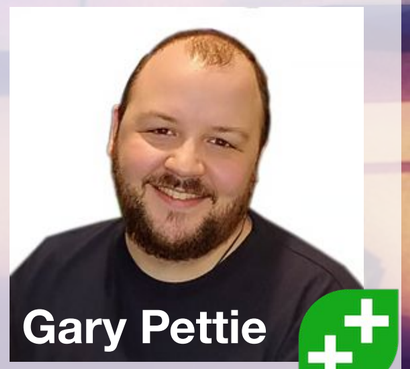
GameDev.tv

Unity 2020.1

Section Intro - Realm Rush



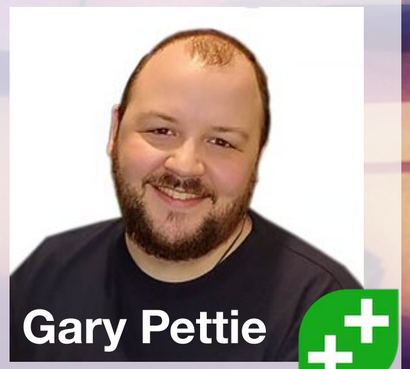
Gary Pettie


[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

Introduction - Realm Rush



Gary Pettie

Realm Rush

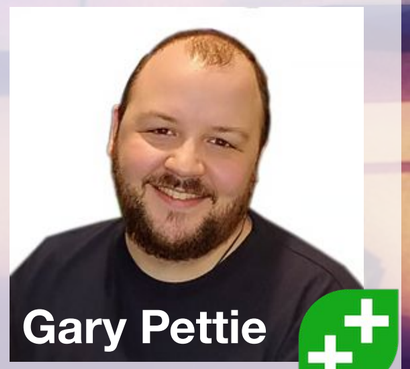
A 3D isometric tower defense style game that showcases object instantiation and management, as well as AI pathfinding, namely Breadth First Search (BFS).

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

GameDev.tv

Unity 2020.1

Realm Rush Game Design



Gary Pettie

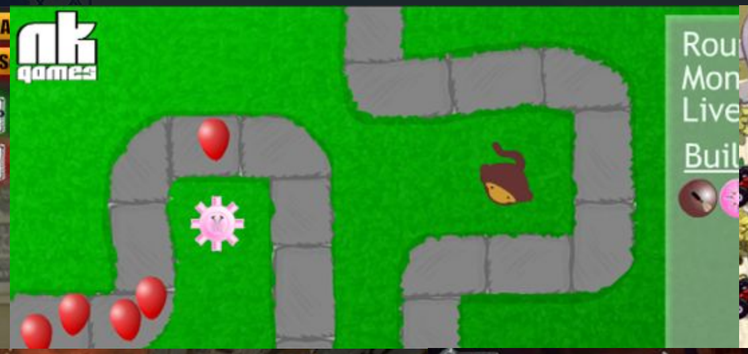
Research

- What is a “Tower Defense” game?

- Place defensive structures
- Defend an objective
- Waves of enemies

Research

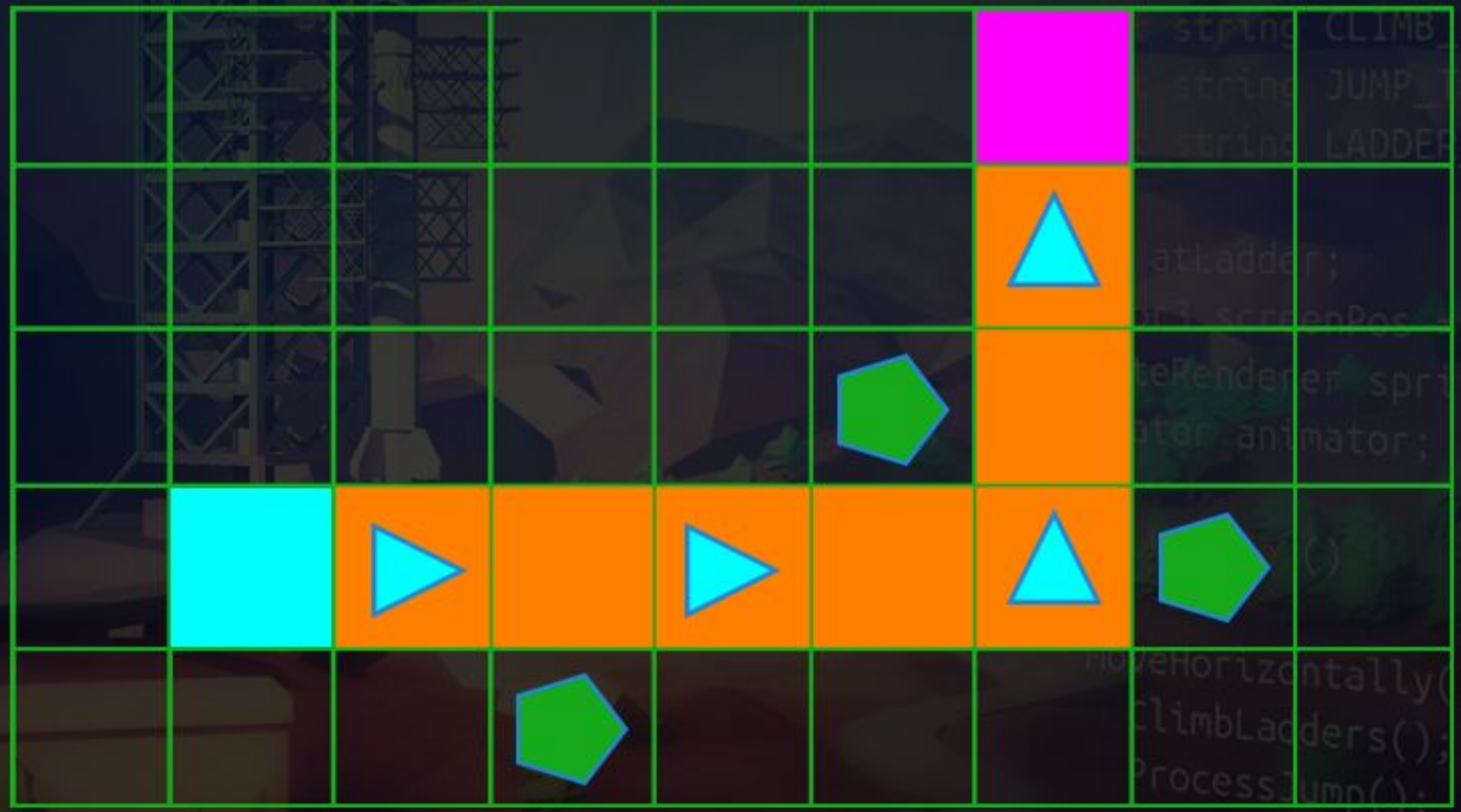

Score: 6
12
20
16
Normal


nk games


175
Menu


Spring Trap
36
3/10
79215

Requirements



Realm Rush Game Design

Player Experience:

Tactical / Strategic

Core Mechanic:

Using limited resources, strategically place towers to stop enemies from reaching their goal

Core game loop:

Survive against waves of enemies for as long as possible!

Level reloads when the player runs out of gold

Game Theme

Story:

Defend against an invading kingdom who have come to pillage your castle!

Art:

Medieval Fantasy

Voxel Art

Tech Setup

- Platform: PC / Mac / Linux
- Aspect Ratio: 1920x1080 16:9
- Input: Mouse

Game Screen

Stretch Goals

Enemy Pathfinding

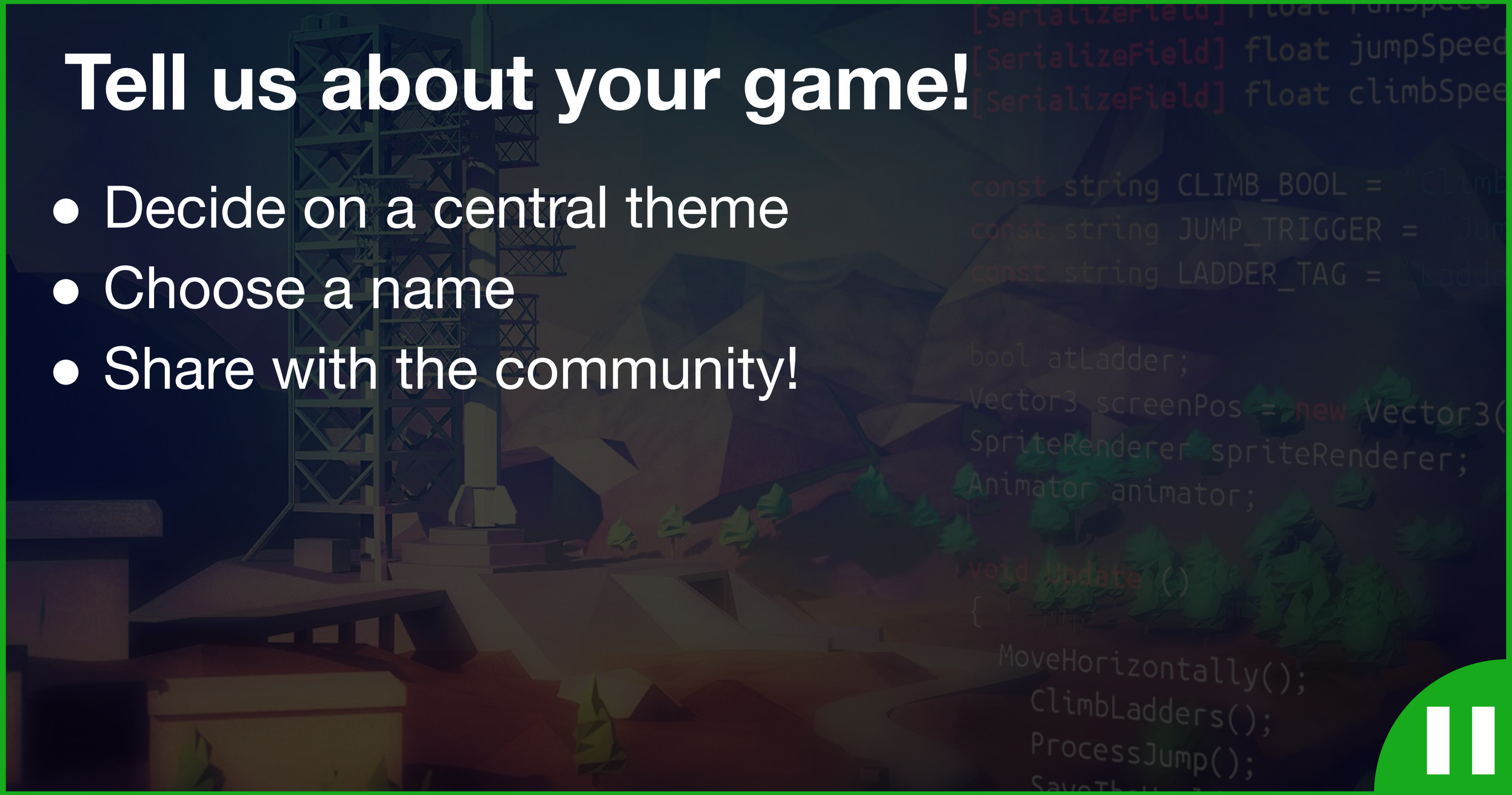
Enemies will dynamically change their path

Build Timer

Towers will take n-seconds to build

Tell us about your game!

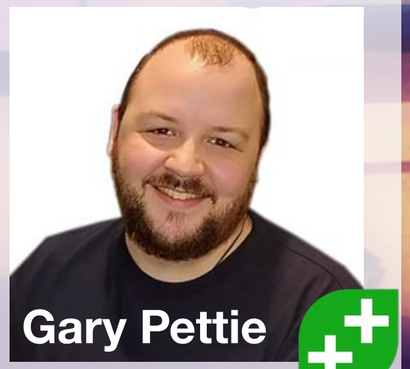
- Decide on a central theme
- Choose a name
- Share with the community!



GameDev.tv

Unity 2020.1

Grid Snapping

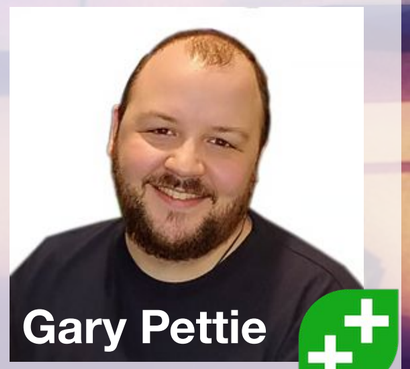


Gary Pettie

GameDev.tv

Unity 2020.1

Text Labels



Gary Pettie

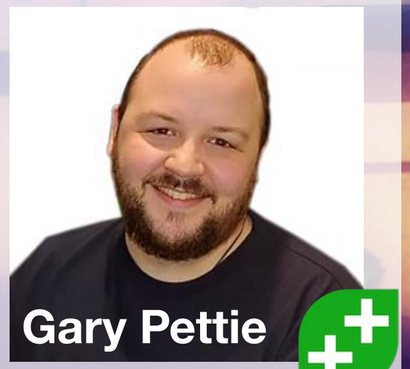
Setup Your Coordinate Labels

- Import the TextMesh Pro Asset
- Align it to the top of your tile
- -99, -99 should fit on the top
- Create a prefab of your tile

GameDev.tv

Unity 2020.1

Coordinate System

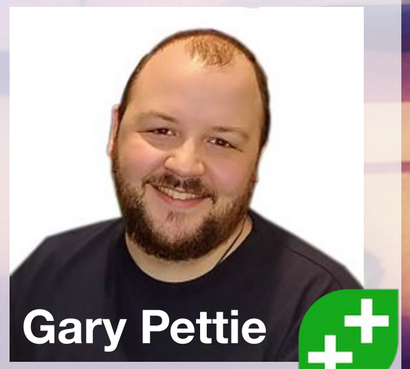


Gary Pettie

GameDev.tv

Unity 2020.1

Lists



Gary Pettie

```
[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();
}
```

Lists vs Arrays in C#

	Array	List
Collection Size	Fixed	Variable
Performance	Fast	Varies
Ease of Use	Easy	Easy
Flexibility	Low	High

“If in doubt, whip a list out”

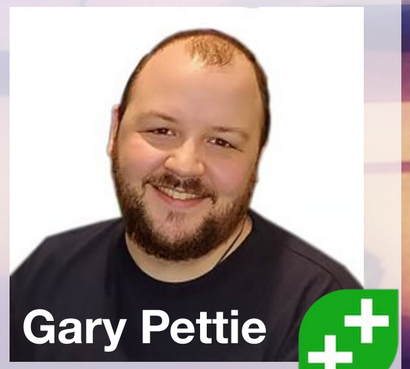
Get Your Enemy Patrolling

- Get your enemy moving along your path!

GameDev.tv

Unity 2020.1

Introducing Coroutines



Gary Pettie

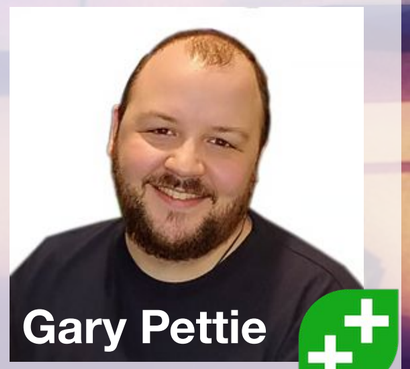
Get Your Enemy Patrolling

- Get your enemy moving along your path!

GameDev.tv

Unity 2020.1

Importing Assets

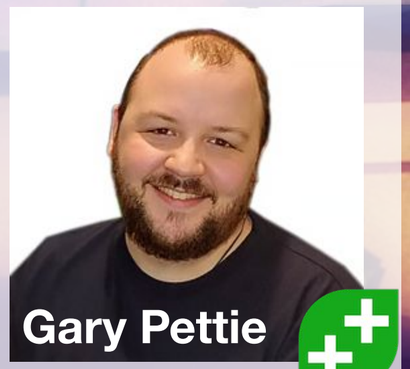


Gary Pettie

GameDev.tv

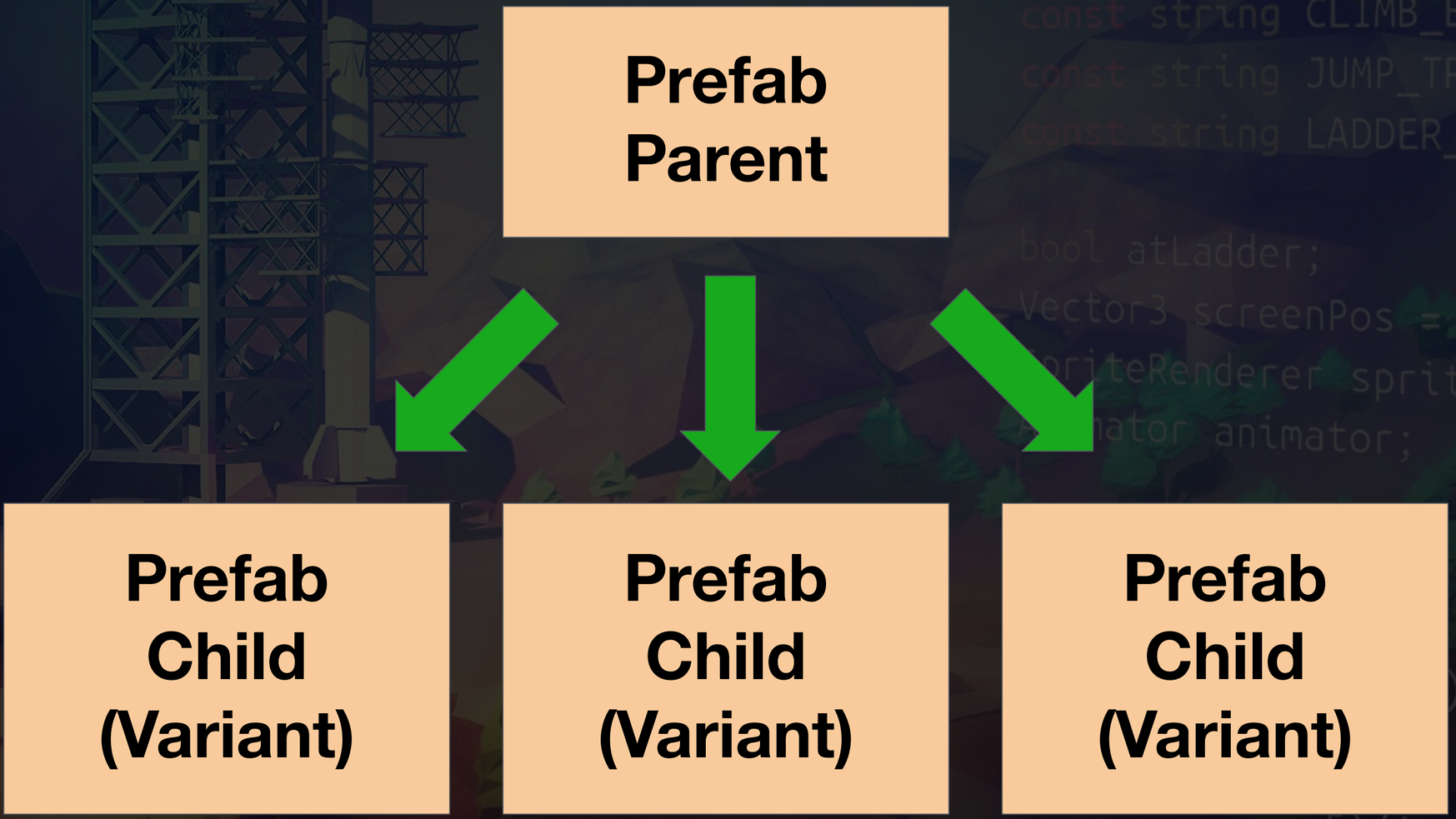
Unity 2020.1

Prefab Variants



Gary Pettie

Prefab Hierarchy

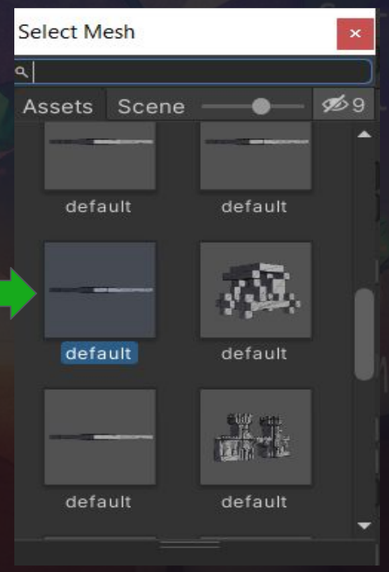


Create a prefab variant

- Create a prefab variant of our Tile Prefab
- Assign a new mesh to make it a corner piece
- Name the prefab variant Tile_RoadCorner

Hint:

The corner piece should be this mesh



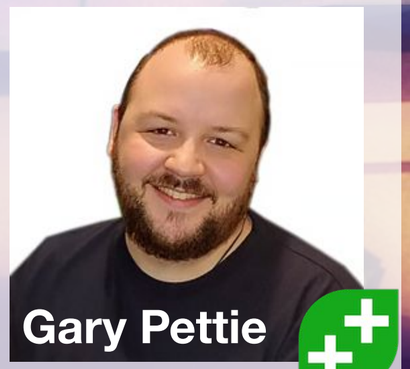
Prefab variant of a prefab variant

- Create 3 new prefab variants from our “Tile_RoadCorner” prefab variant
- Rotate the meshes by 90, 180, and 270
- Name the prefab variants:
  - “Tile_RoadCorner90”
  - “Tile_RoadCorner180”
  - “Tile_RoadCorner270”

GameDev.tv

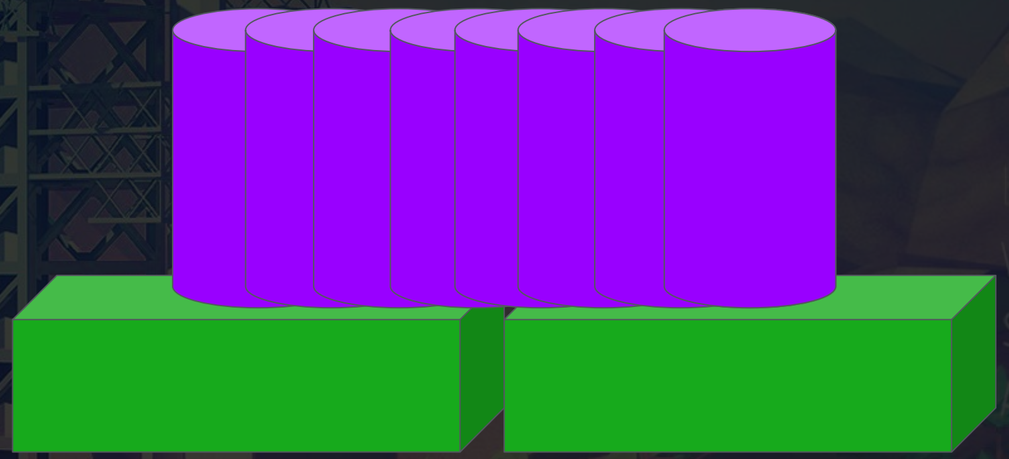
Unity 2020.1

Smooth Enemy Movement



Gary Pettie

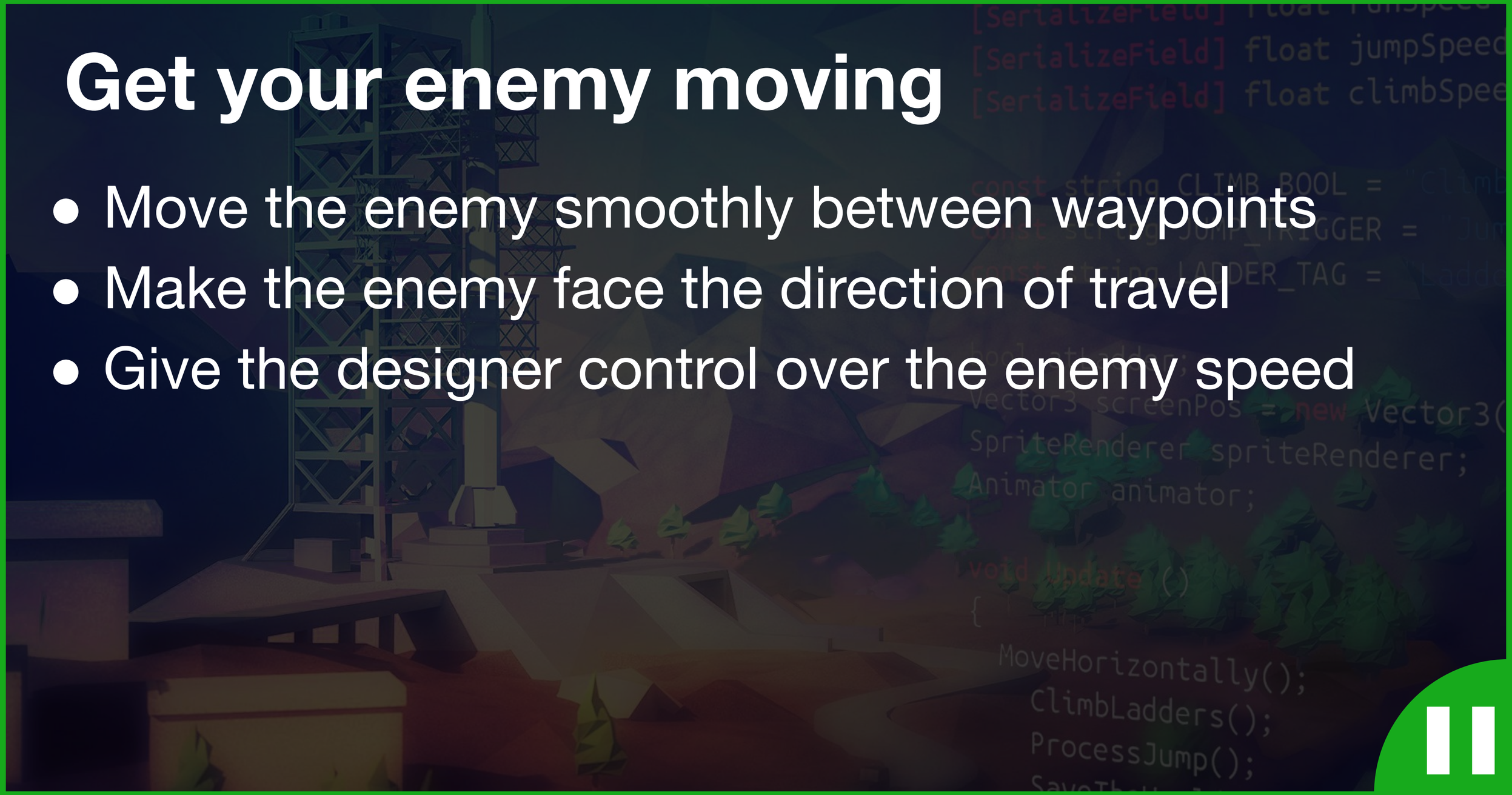
LERP (Linear Interpolation)



```
Vector3.LERP(startPosition, endPosition, travelPercent)
```

Get your enemy moving

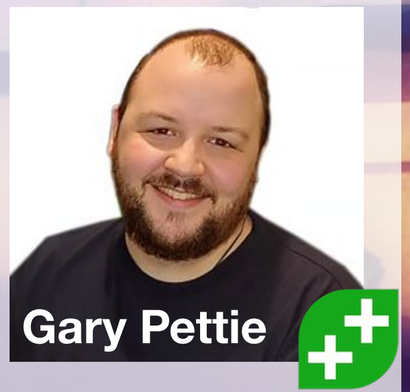
- Move the enemy smoothly between waypoints
- Make the enemy face the direction of travel
- Give the designer control over the enemy speed



GameDev.tv

Unity 2020.1

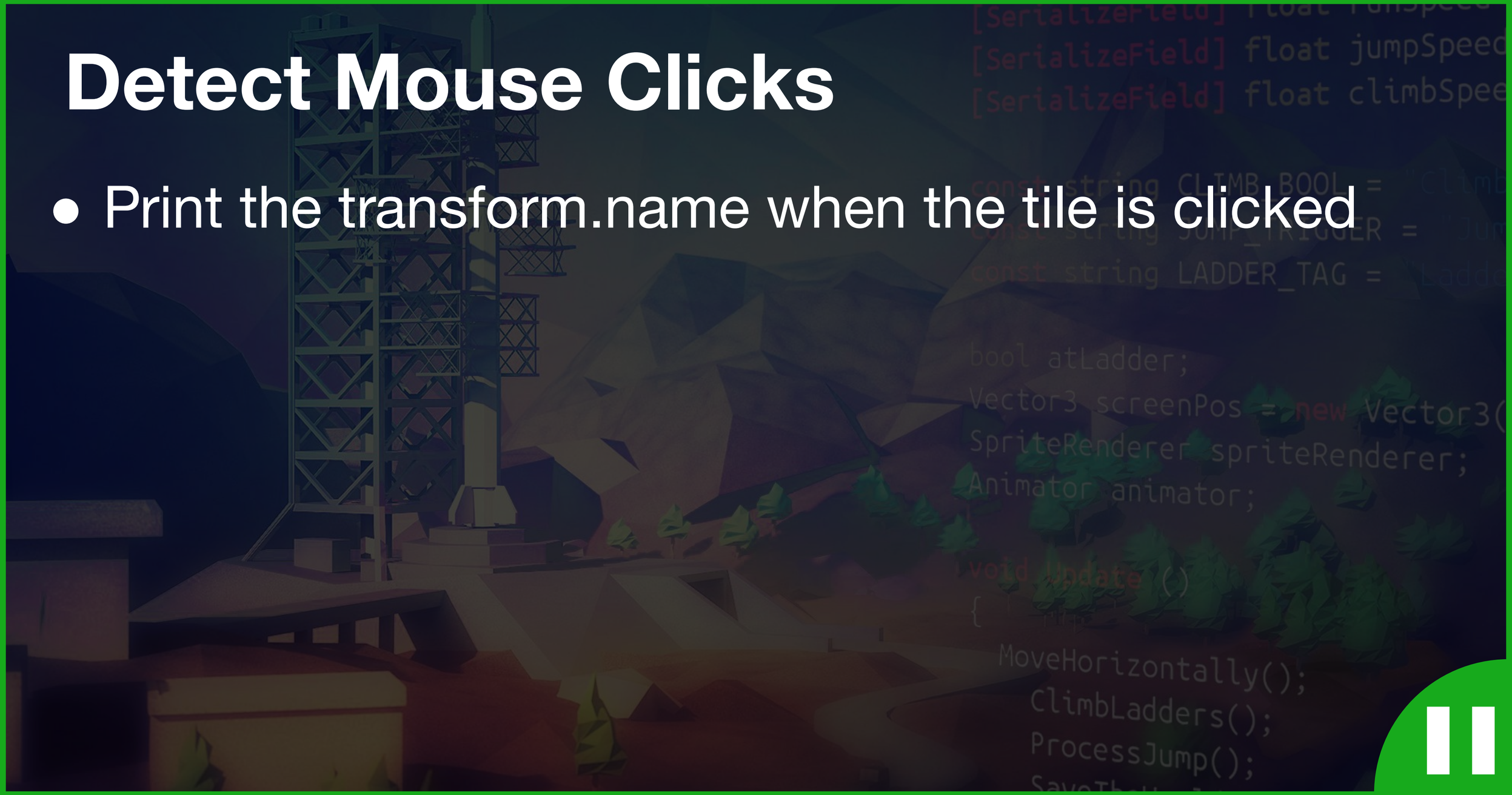
Detect Mouse Input



Gary Pettie

Detect Mouse Clicks

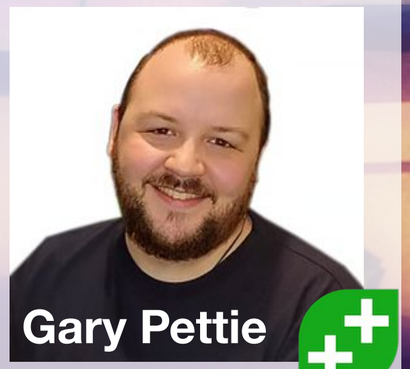
- Print the transform.name when the tile is clicked



GameDev.tv

Unity 2020.1

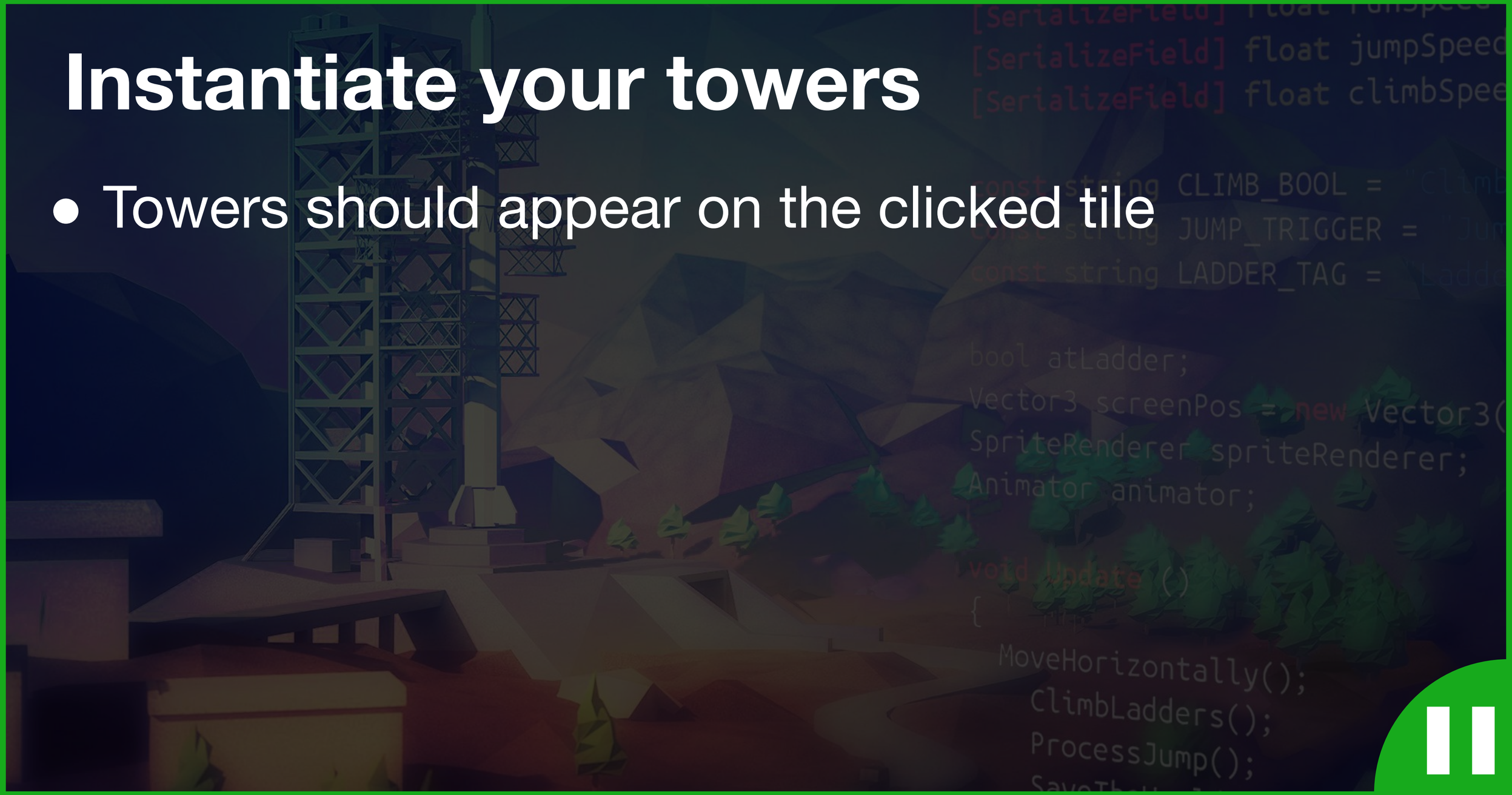
Targeting Enemies



Gary Pettie

Instantiate your towers

- Towers should appear on the clicked tile

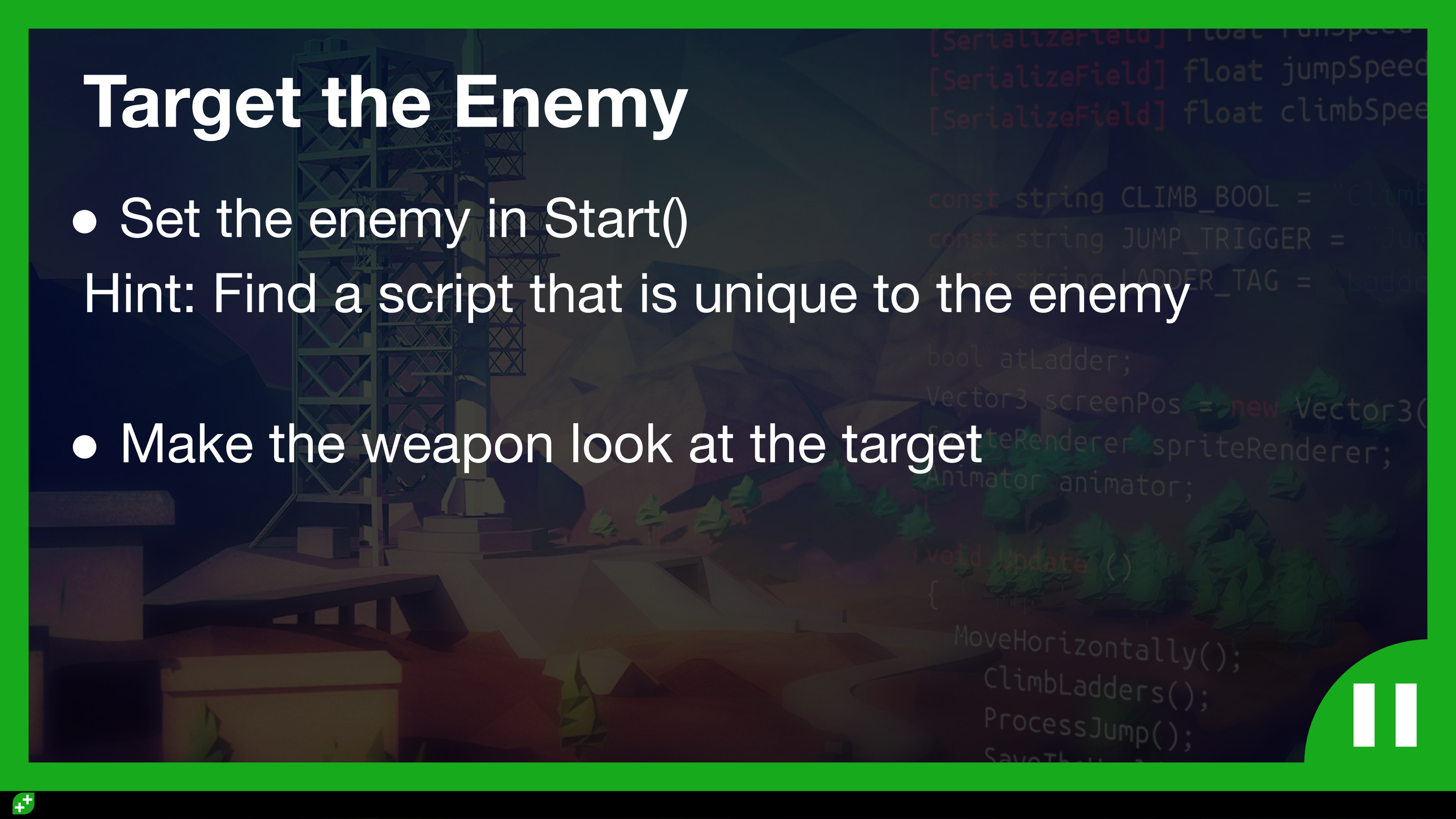


Target the Enemy

- Set the enemy in Start()

Hint: Find a script that is unique to the enemy

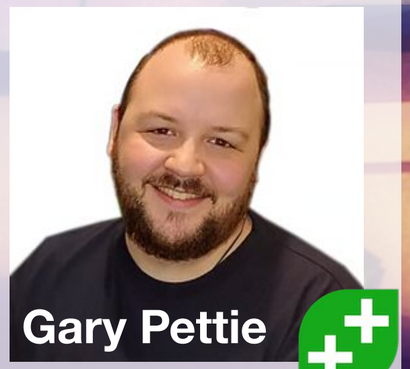
- Make the weapon look at the target



GameDev.tv

Unity 2020.1

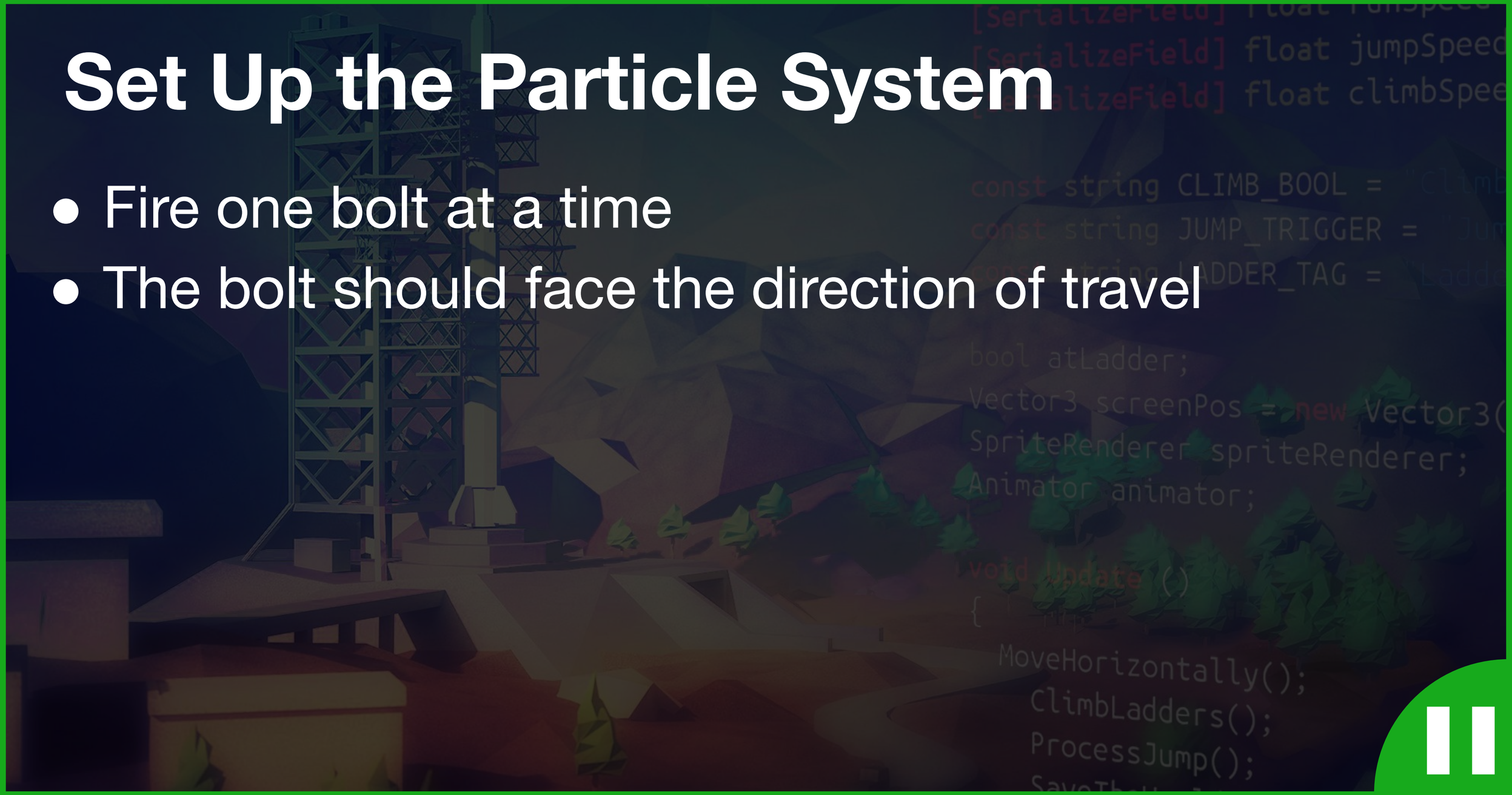
Damaging Enemies



Gary Pettie

Set Up the Particle System

- Fire one bolt at a time
- The bolt should face the direction of travel



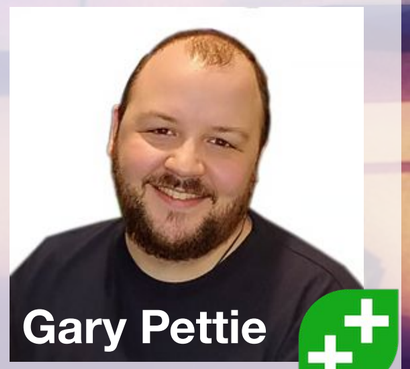
Register Enemy Damage

- Create the EnemyHealth script using your knowledge from Argon Assault
- Set currentHitPoints to maxHitPoints in Start()
- Register Particle Collisions
- Reduce currentHitPoints when a collision occurs

GameDev.tv

Unity 2020.1

Debugging Tools



Gary Pettie

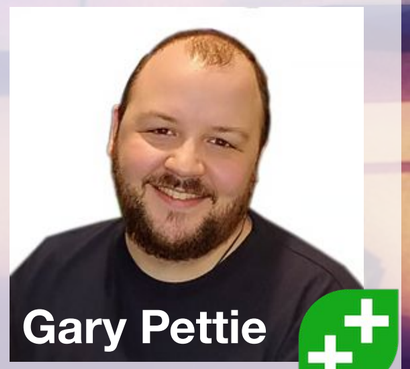
Change the Label Color

- Change the color of the label based on the value of IsPlaceable in the Waypoint script

GameDev.tv

Unity 2020.1

Find the Path

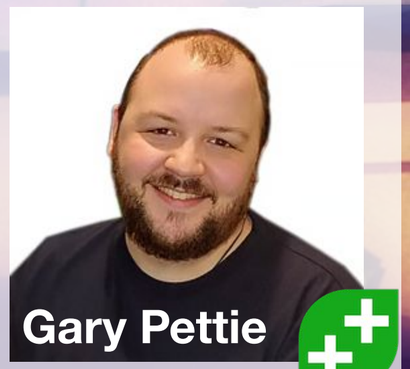


Gary Pettie

GameDev.tv

Unity 2020.1

Instantiating Enemies

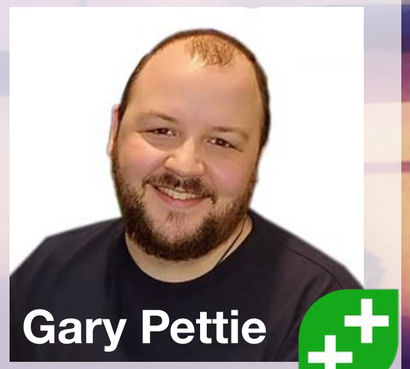


Gary Pettie

GameDev.tv

Unity 2020.1

Object Pools

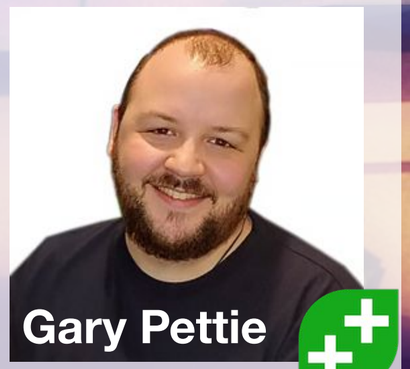


Gary Pettie

GameDev.tv

Unity 2020.1

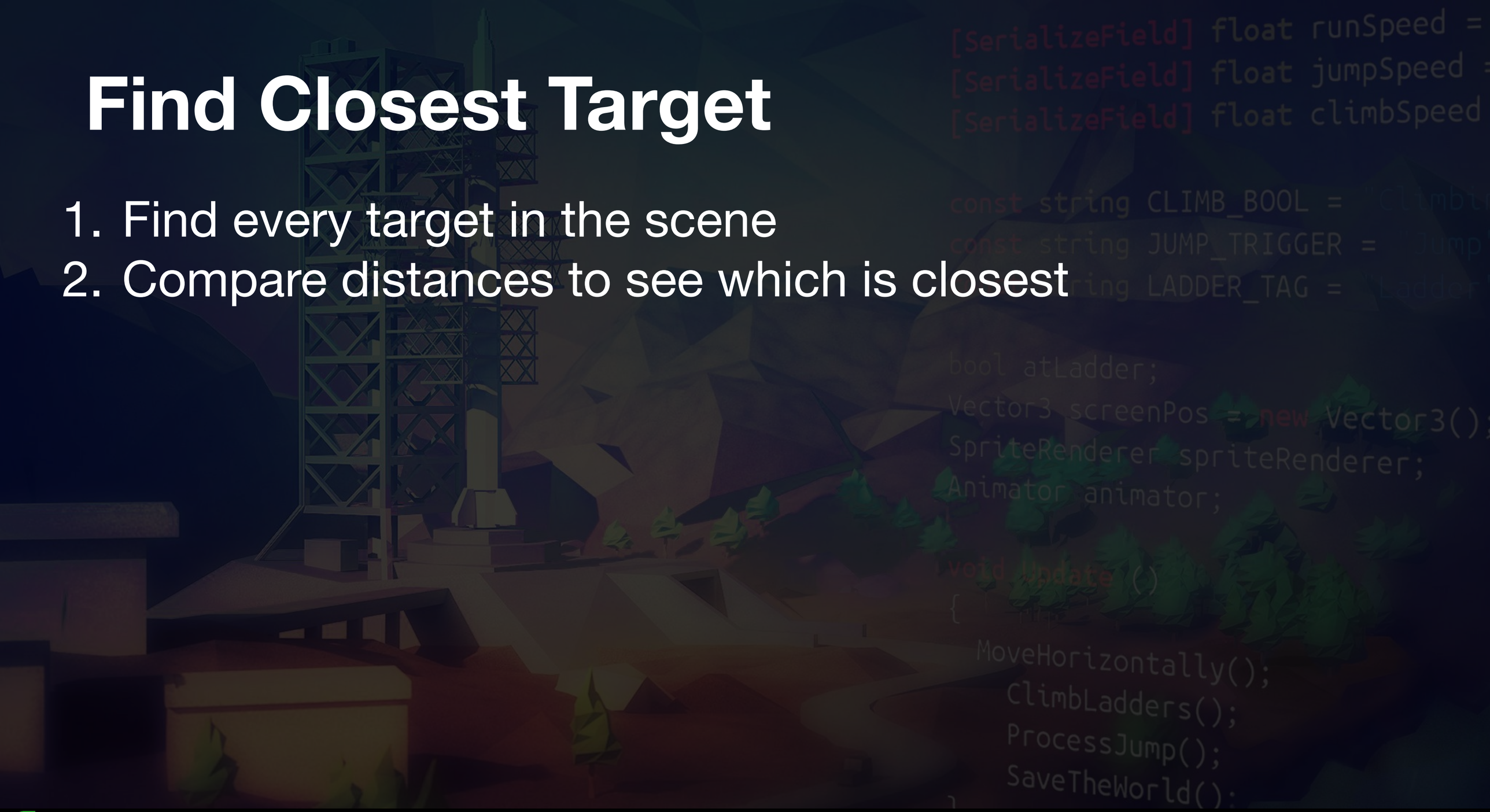
Target Closest Enemy



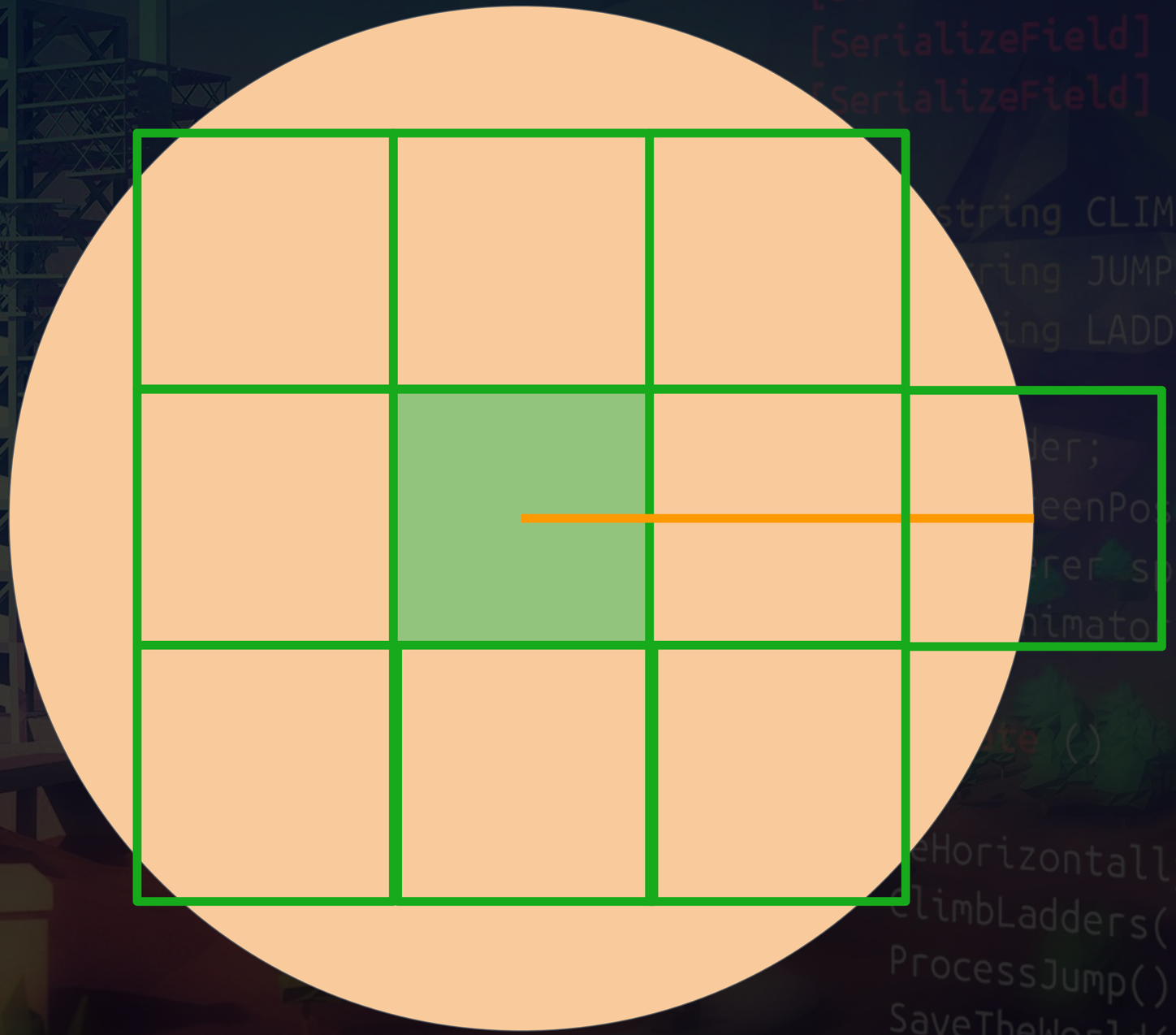
Gary Pettie

Find Closest Target

1. Find every target in the scene
2. Compare distances to see which is closest



Range



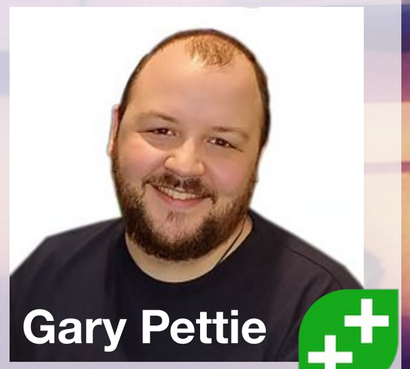
Control the Particle System

- Write the Attack() method
  - Control the active state of the emissionModule with the isActive parameter
- Call Attack() from AimWeapon()
  - Attack(true) when target is in range
  - Attack(false) when the target is out of range

GameDev.tv

Unity 2020.1

Damaging Enemies



Gary Pettie

Create Infinite Enemies!

- Instantiate an enemy every 1 second using a coroutine

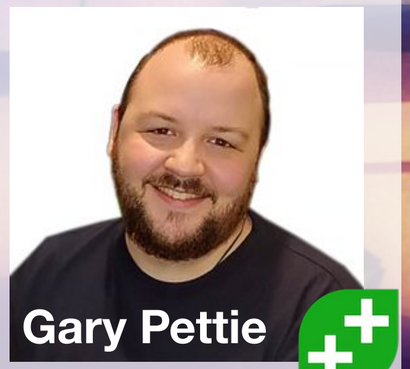
Hint:

Try using a ‘while’ loop

GameDev.tv

Unity 2020.1

Object Pools



Gary Pettie

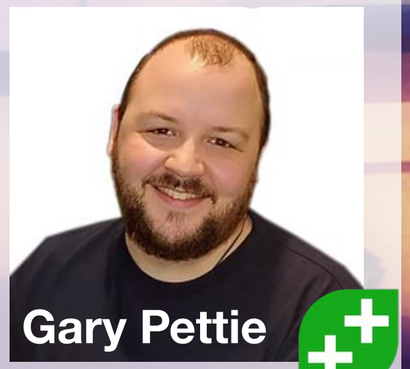
Enable Objects in the Pool

- Loop through each element in our ‘pool’ array
- Check if the object is Active In the Hierarchy
- If it is not:
  - Set it to be Active
  - return from the method early

GameDev.tv

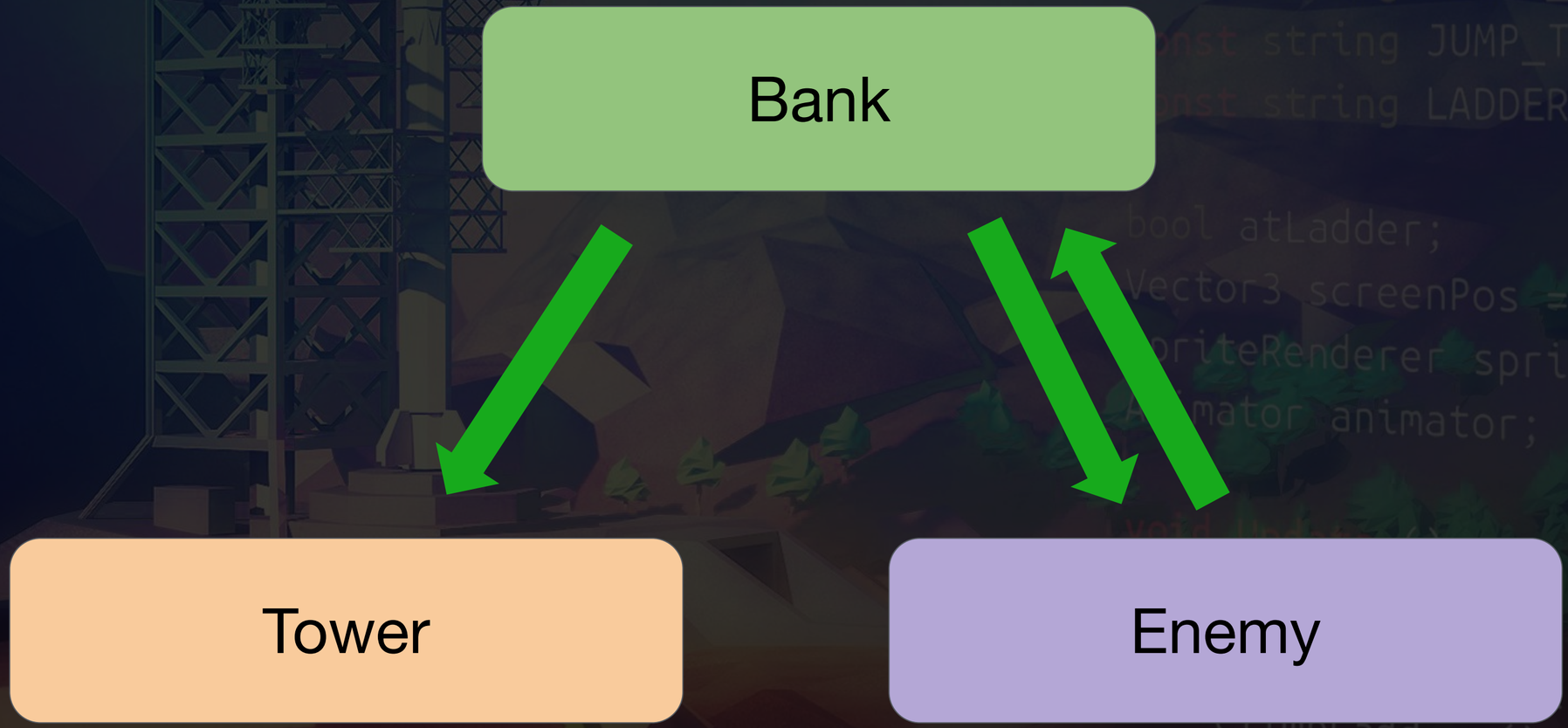
Unity 2020.1

Currency System (Part 1)



Gary Pettie

Bank of the Realm



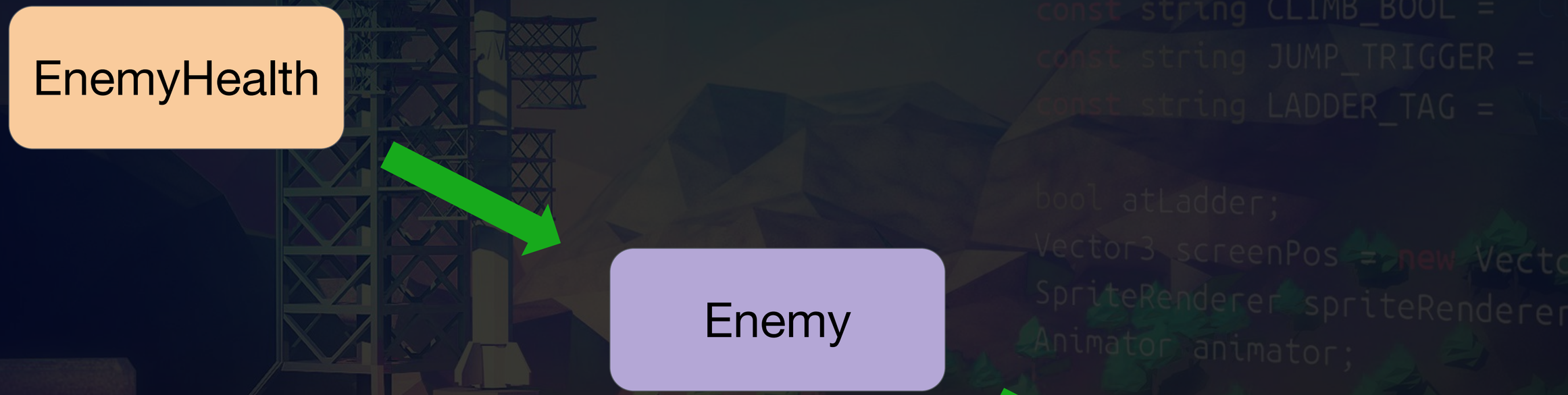
Withdrawals

- Write the Withdraw() method

Hint:

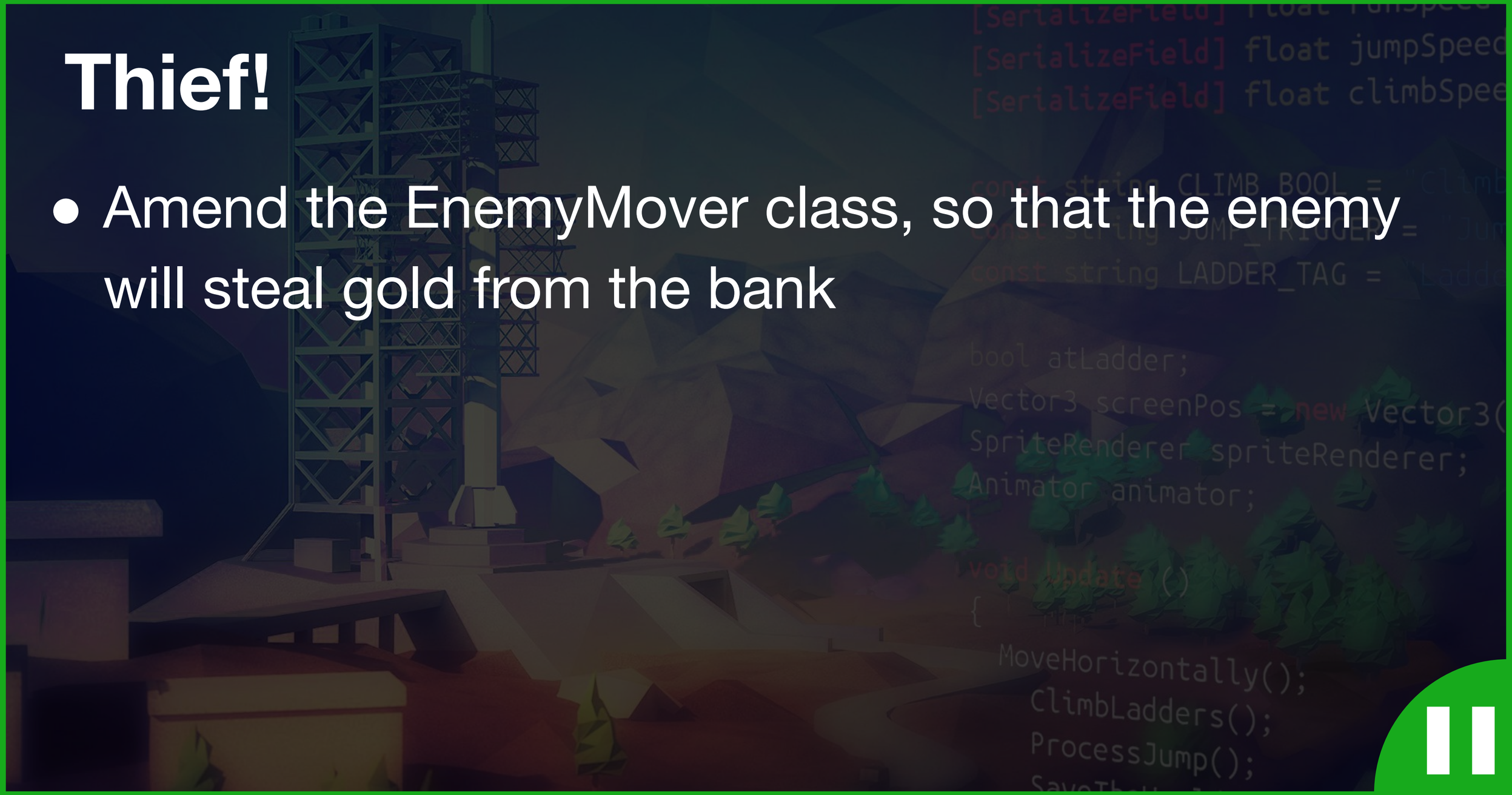
Use the Deposit(int amount) method as a template

Pass it on!



Thief!

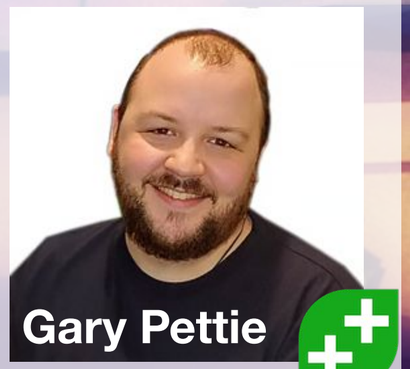
- Amend the EnemyMover class, so that the enemy will steal gold from the bank



GameDev.tv

Unity 2020.1

Currency System (Part 2)

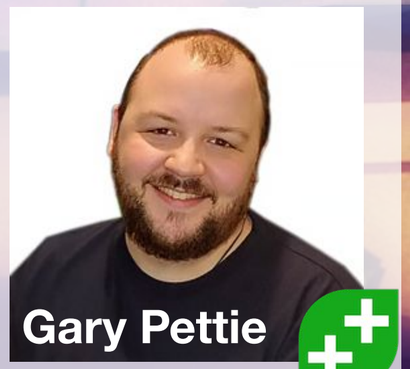


Gary Pettie

GameDev.tv

Unity 2020.1

UI Text

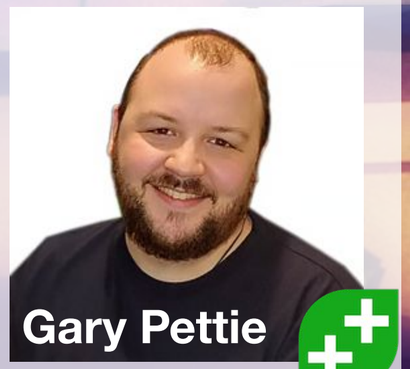


Gary Pettie

GameDev.tv

Unity 2020.1

Increasing Difficulty

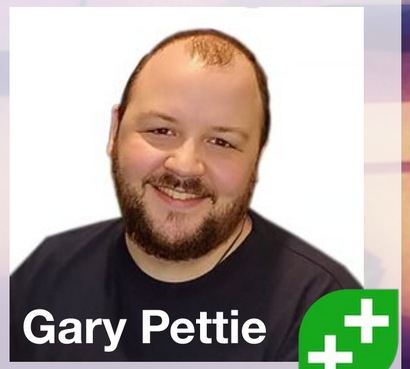


Gary Pettie

GameDev.tv

Unity 2020.1

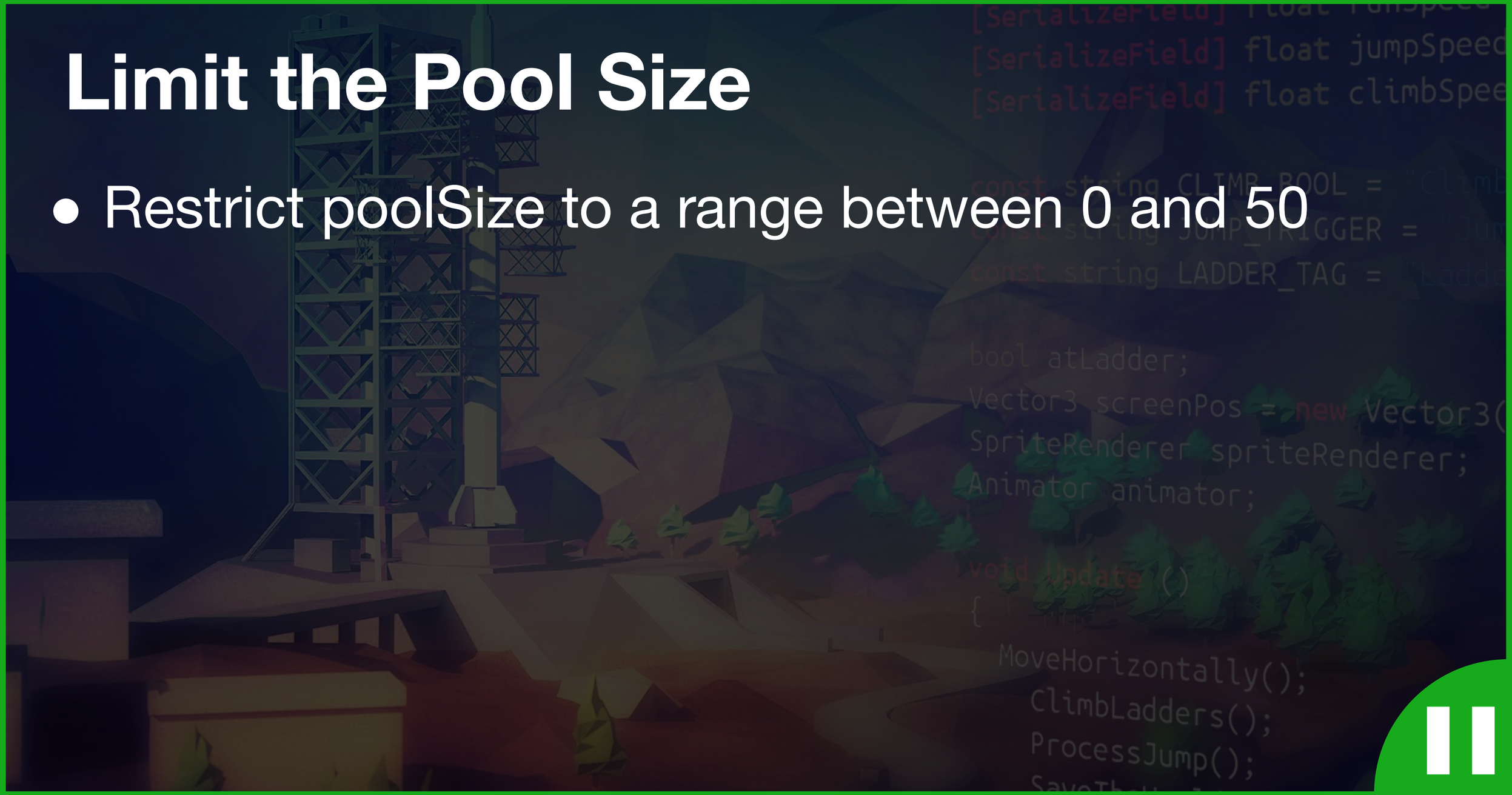
Refactoring



Gary Pettie

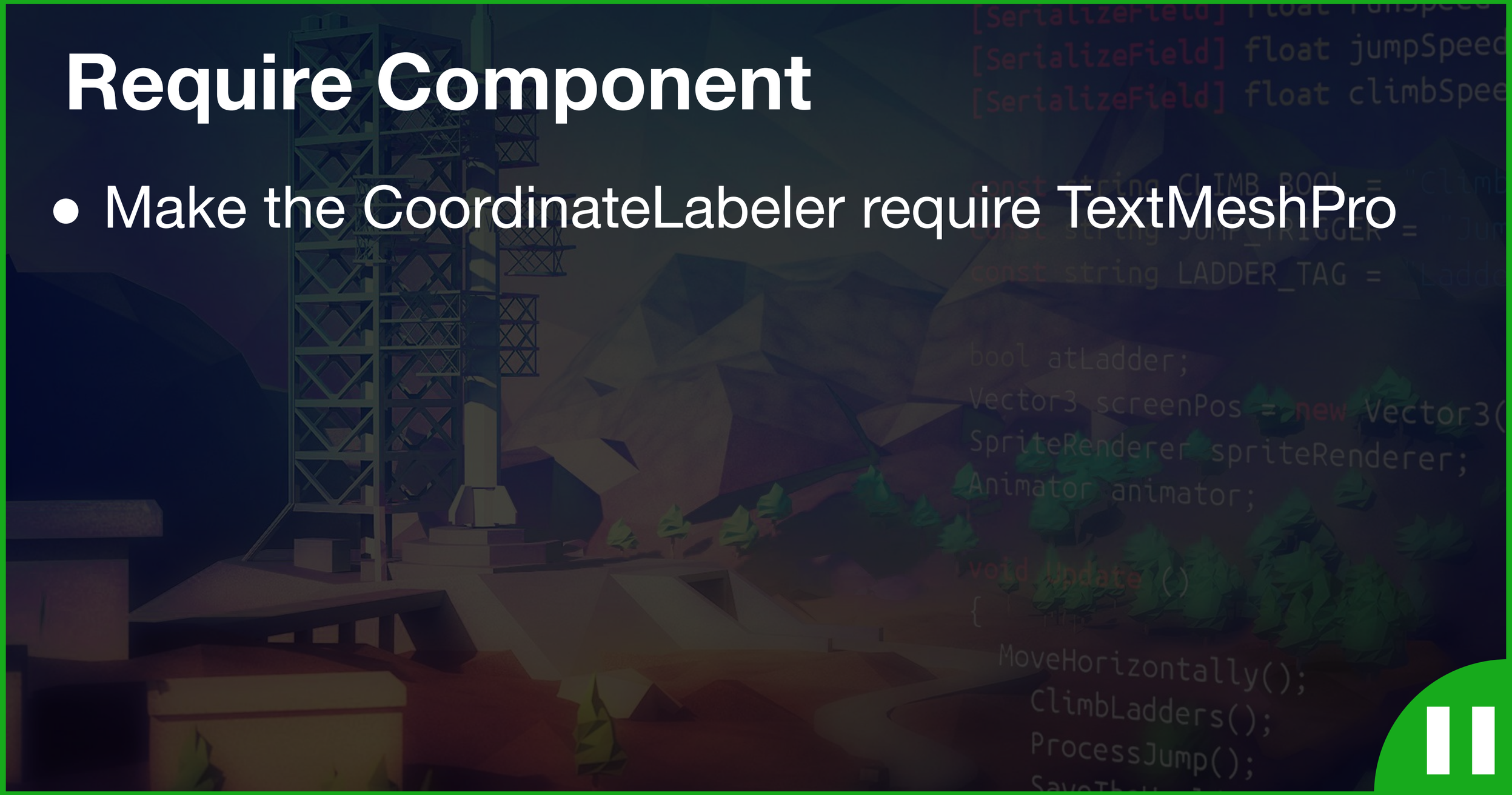
Limit the Pool Size

- Restrict poolSize to a range between 0 and 50



Require Component

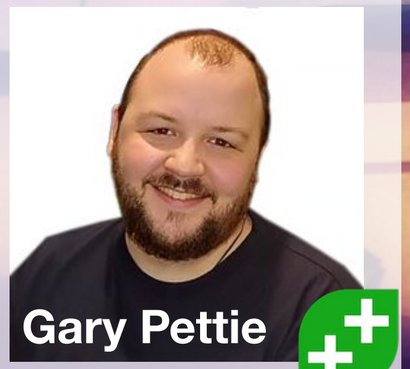
- Make the CoordinateLabeler require TextMeshPro



GameDev.tv

Unity 2020.1

Playtest and Balance



Gary Pettie

Solo Challenge

- Playtest your game
  - Create an interesting level
  - Use prefab variants to add more tiles
  - Add trees to block tower damage
- Balance everything!
  - Tower range, shooting speed
  - Enemy speed, hitpoints, damage ramp
  - Change the size of the object pool

WARNING!

- We’re at the mid-point for this section
- In the next half, we will be changing our design (specifically enemy pathfinding)
- Spend as long as you like on this current prototype But know that we will be changing some of the core elements in upcoming lectures.

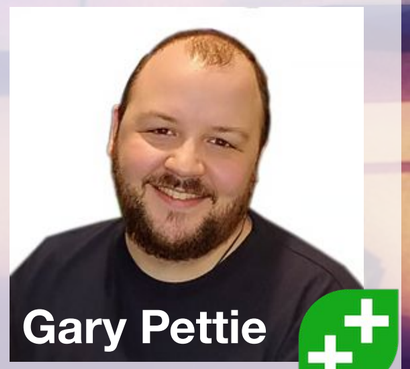
Share and Playtest

- Build your game
- Share your game with the community
  - Ask them to playest and give feedback
  - Give feedback on a game from a fellow student

GameDev.tv

Unity 2020.1

Review and Reflect

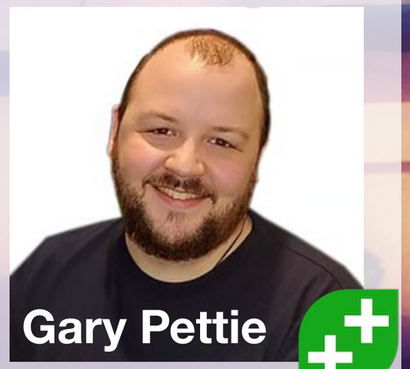


Gary Pettie

GameDev.tv

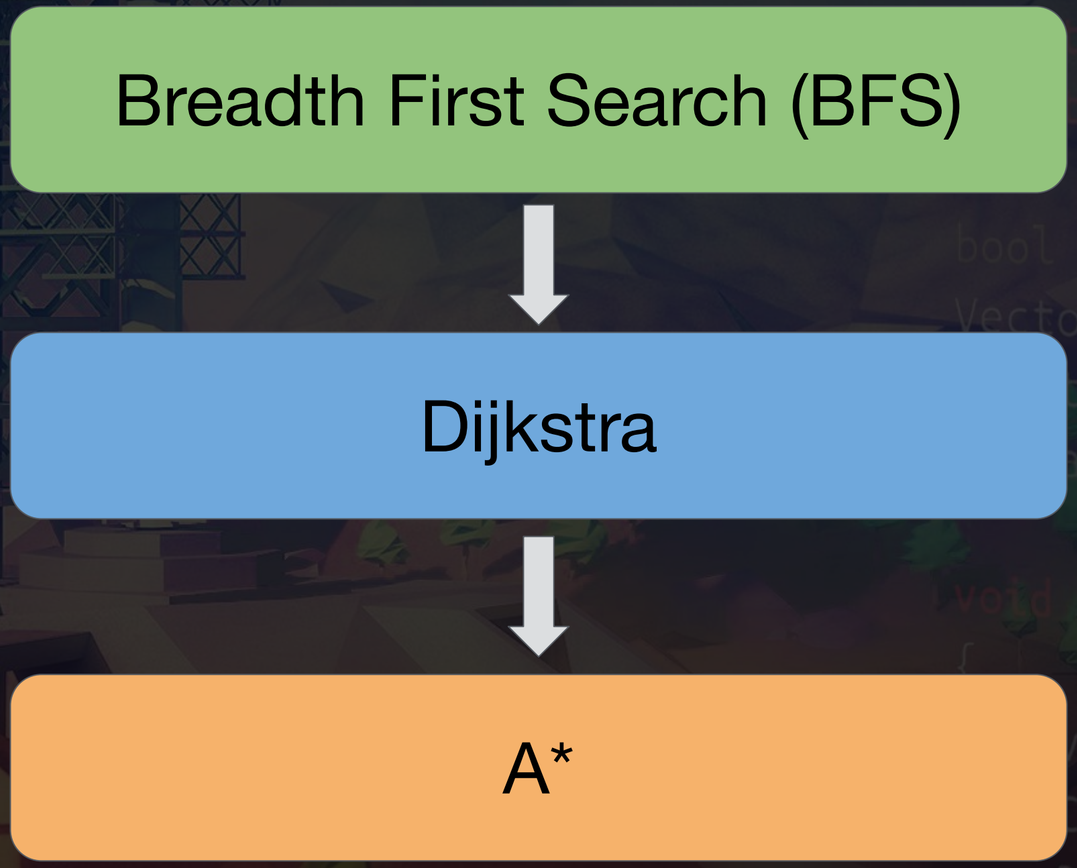
Unity 2020.1

Pathfinding Decisions



Gary Pettie

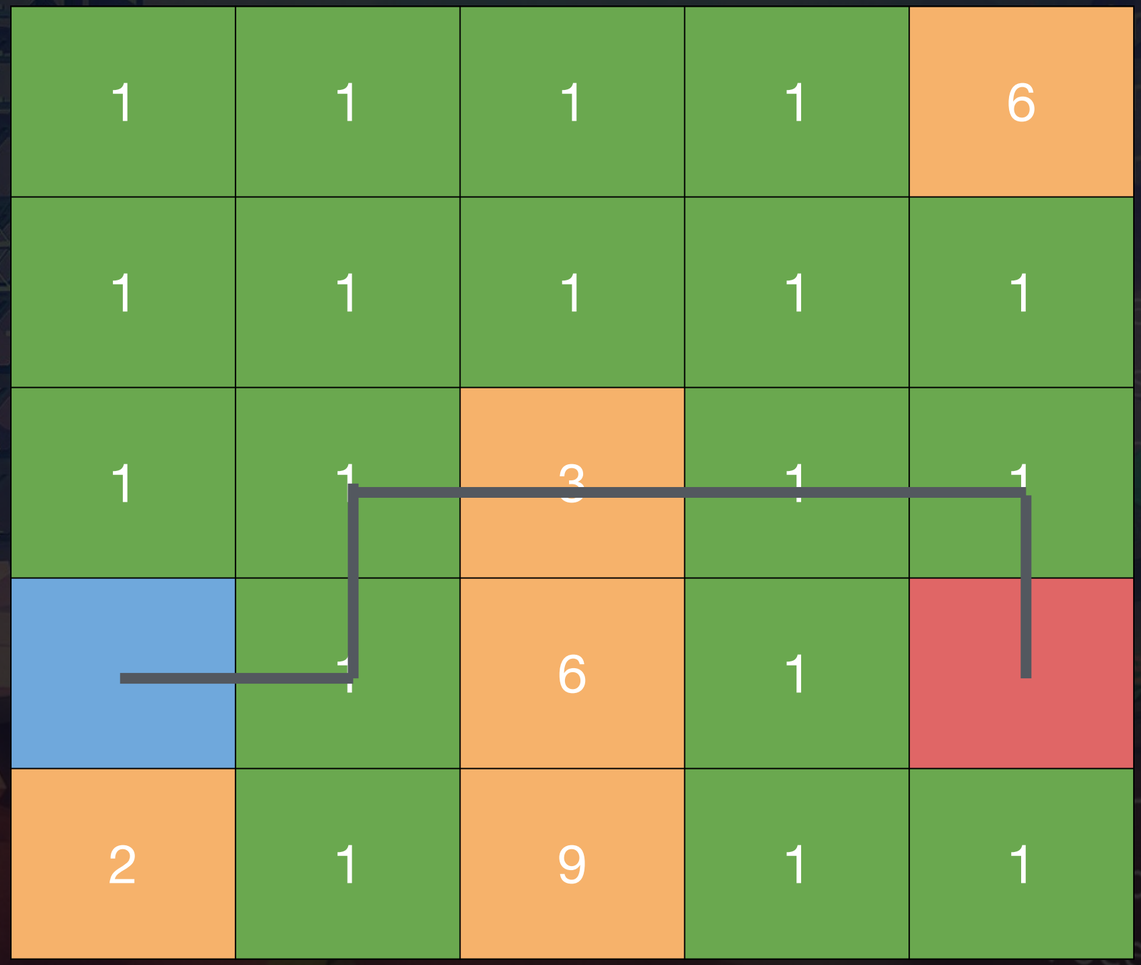
Which Algorithm is Best?



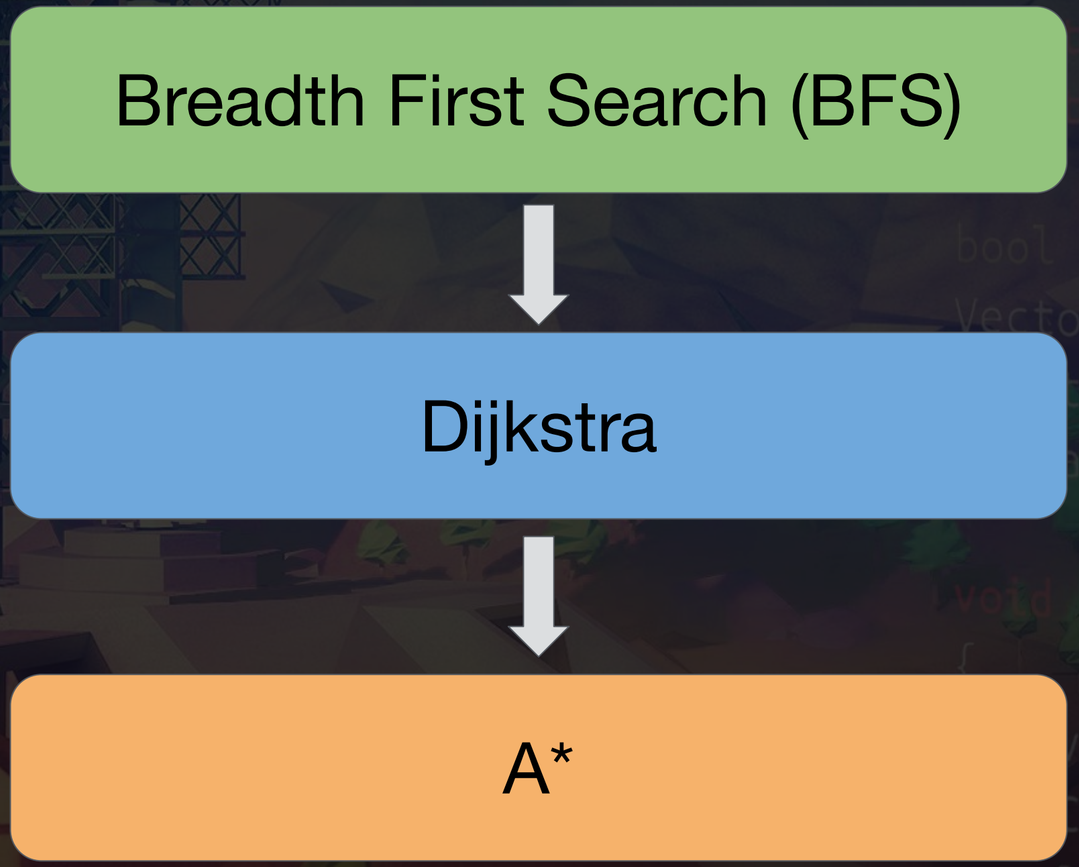
Search Algorithms Overview

Algorithm	Always Shortest Path	Movement Costs	Multiple Start or End Points	Speed
Breadth First Search	✅	❌	✅	Medium
Dijkstra	✅	✅	✅	Slow
A*	❓	✅	❌	Fast

Search Algorithms Overview



Which is “Right” for us?



Search Algorithm Speed Comparison

Approximate worst-case number of calculations to find the shortest path...

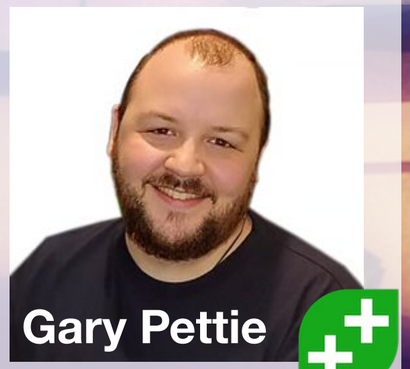
Algorithm	Speed	10x10 Grid	20x20 Grid	100 x 100 Grid
BFS	V + E	280	1160	29800 “Medium”
Dijkstra	E + V log V	380	1801	59800 “Slow”
A*	Varies	180	760	19800 “Fast”

**A star has the benefit that we can chose the trade-off between speed and accuracy to suit our game.*

GameDev.tv

Unity 2020.1

Breadth First Search

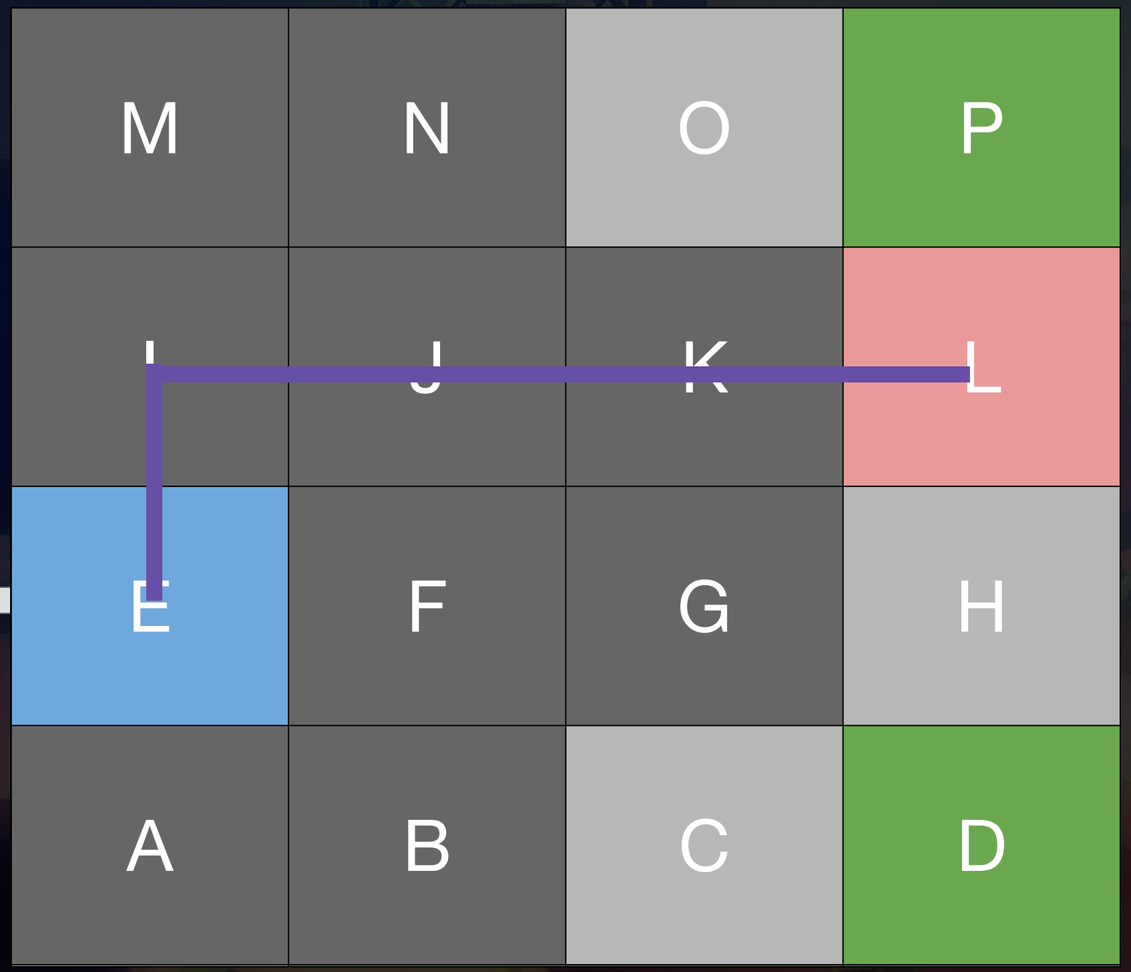


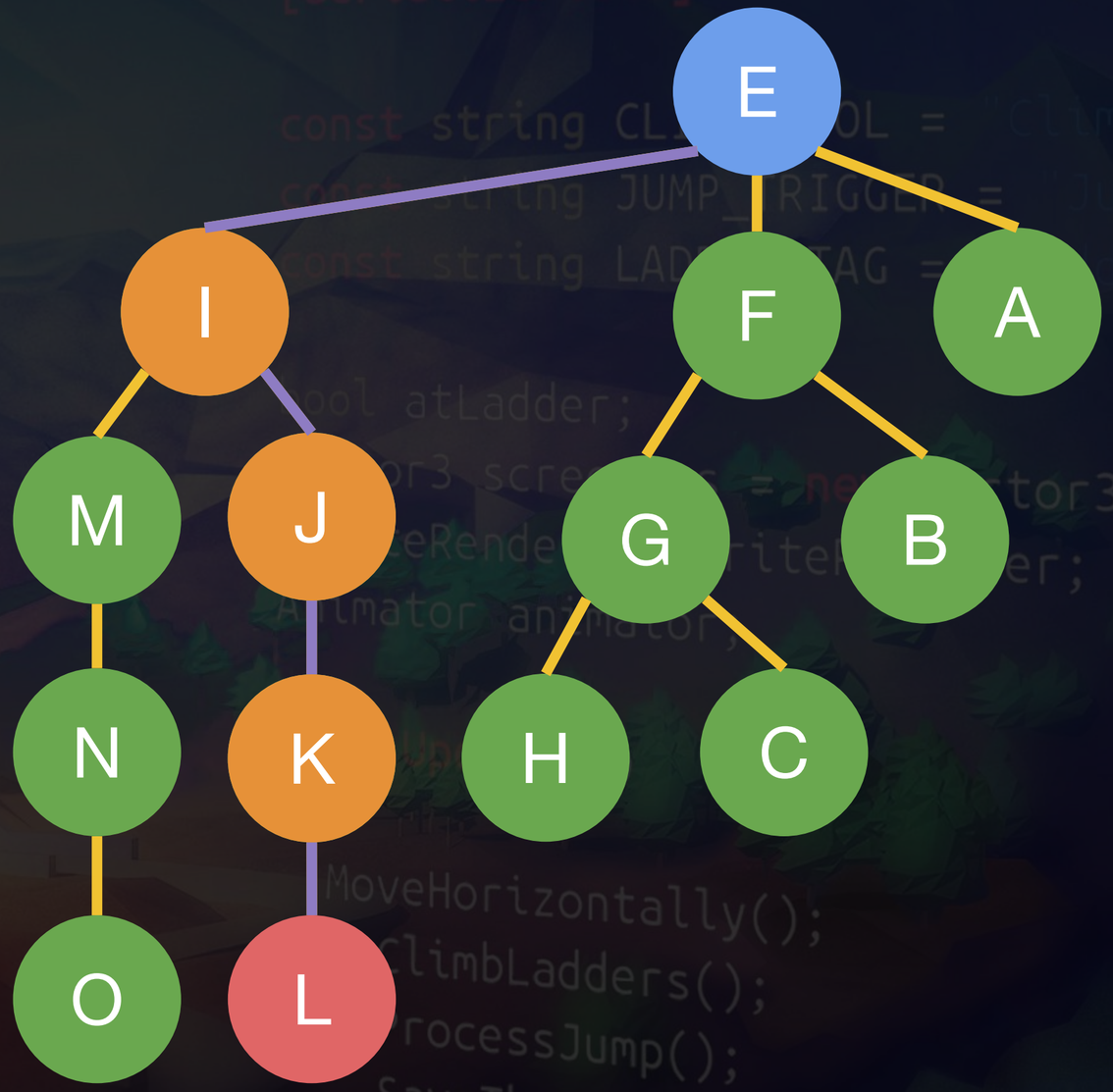
Gary Pettie

Nodes and Edges

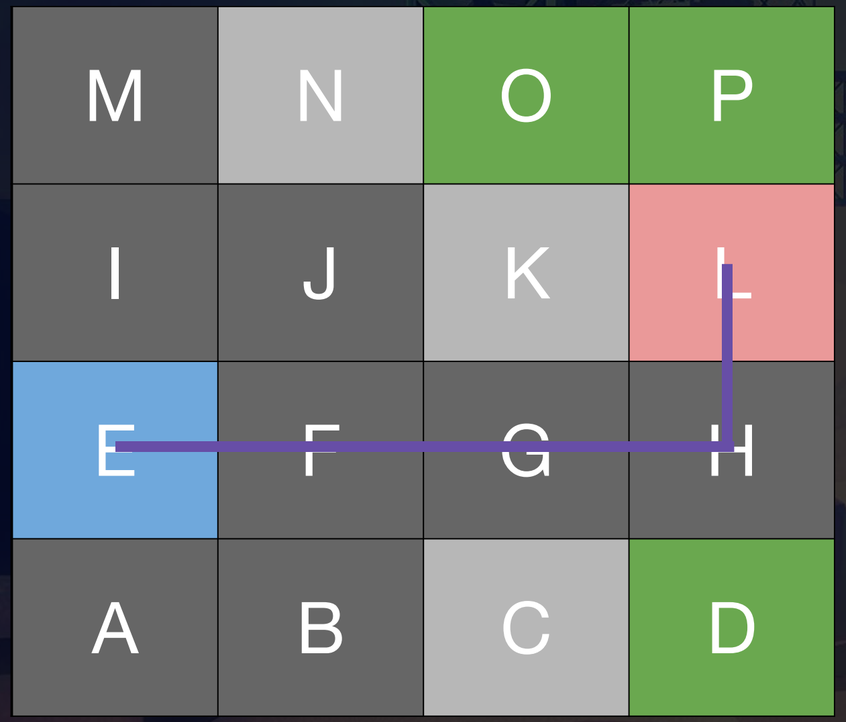
M	N	O	P
I	J	K	L
E	F	G	H
A	B	C	D

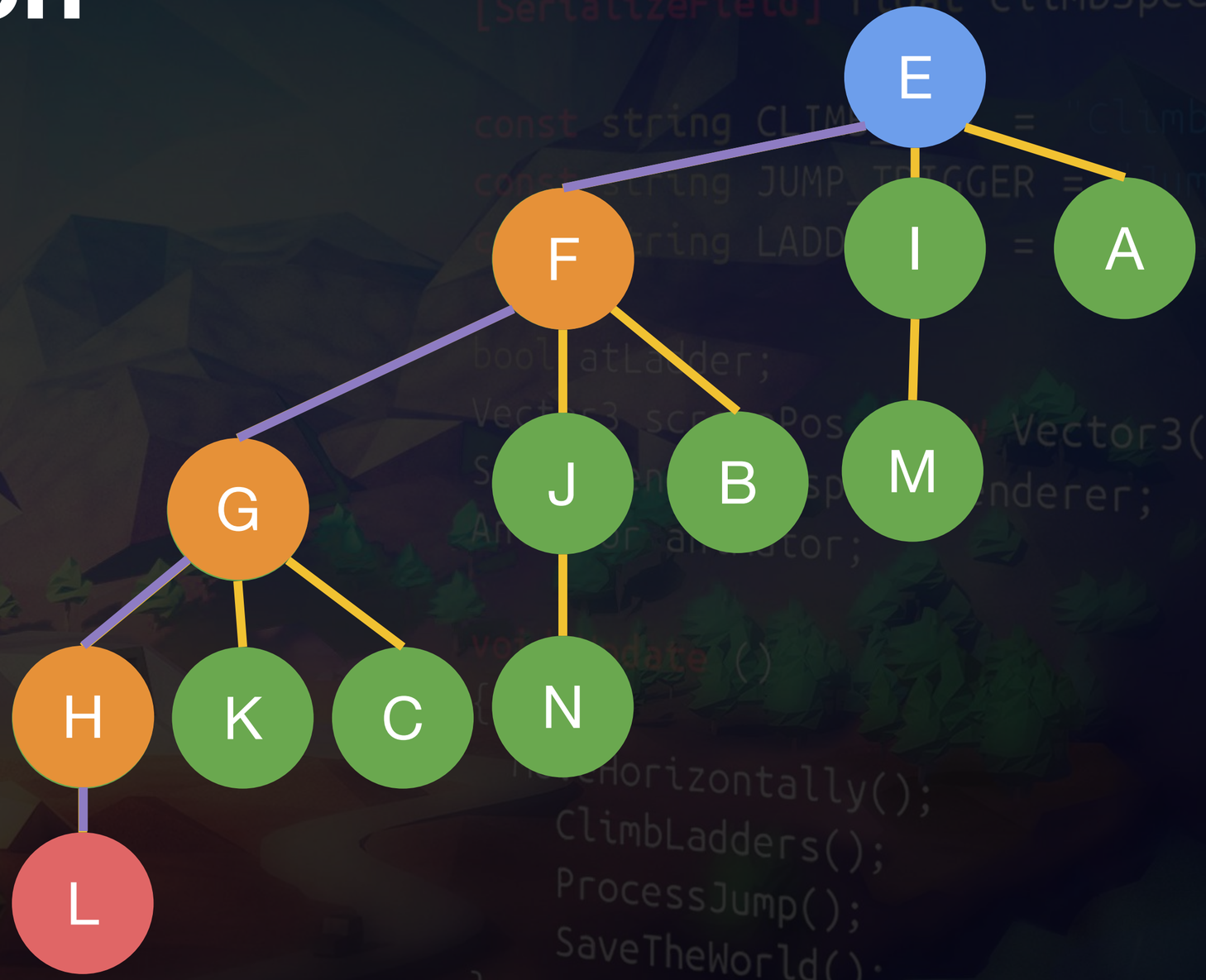
Navigating a Graph





Search Direction





Breadth First Search

1. Choose your search direction
2. Add the current node to the tree
3. Add the neighbours of the current node to the tree
4. Move to the next node in the tree
5. If you’re not at the goal, go to step 3
6. Record your journey back up the tree
7. Reverse the list to get the path in the correct order

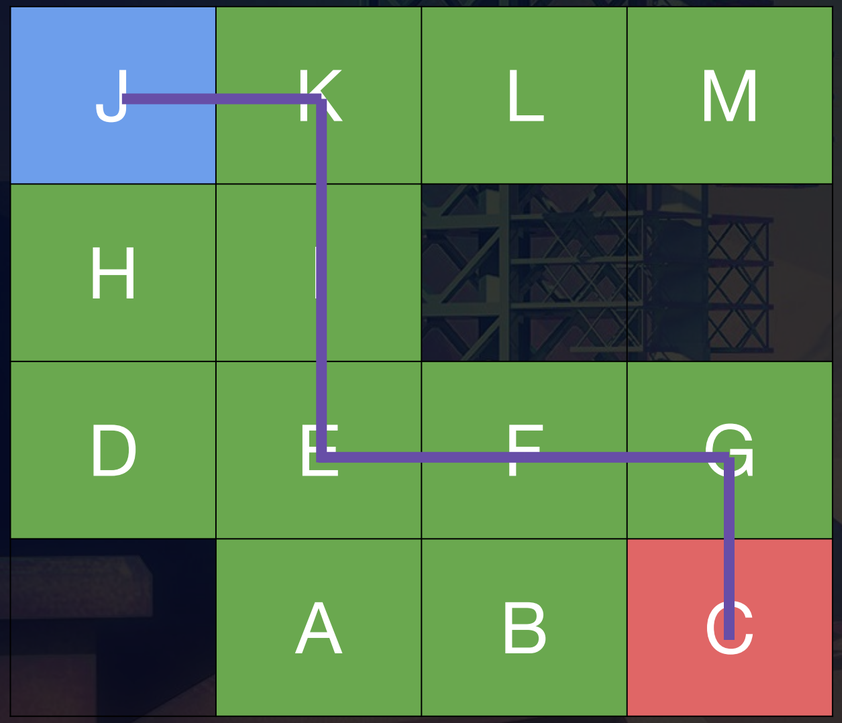
Find the Shortest Path

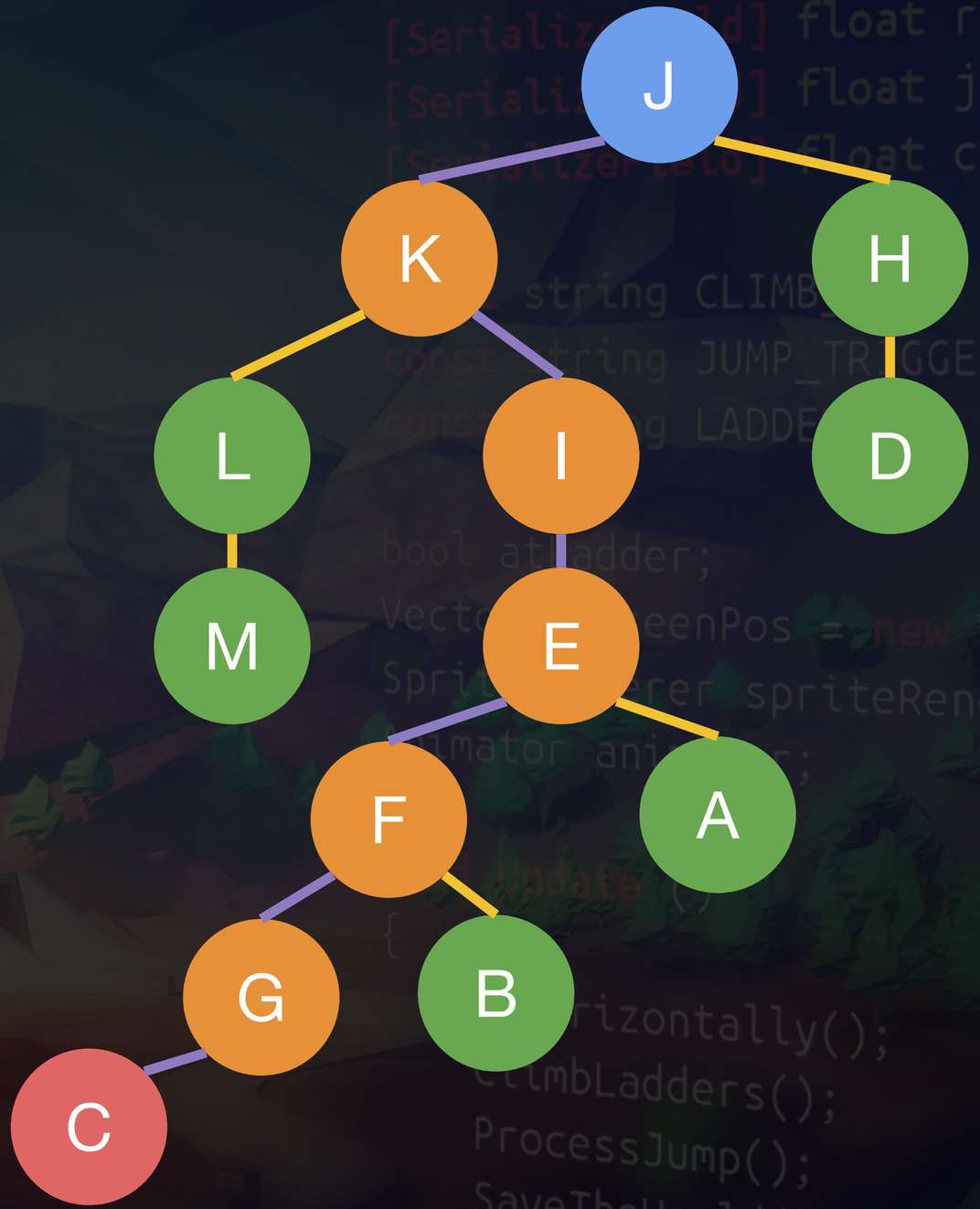
- Try different direction priorities

1. 
2. 

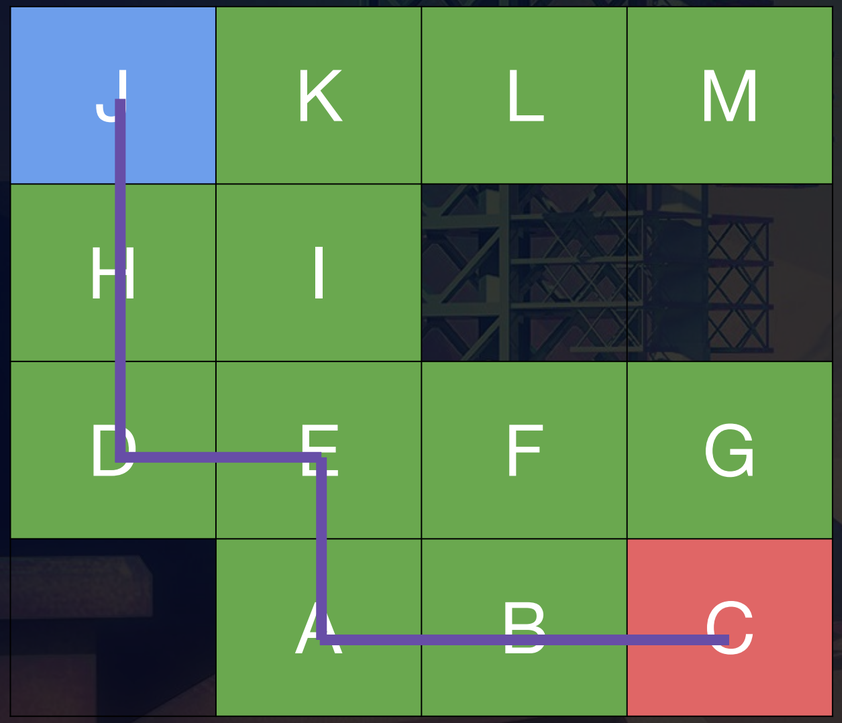
J	K	L	M
H	I		
D	E	F	G
	A	B	C

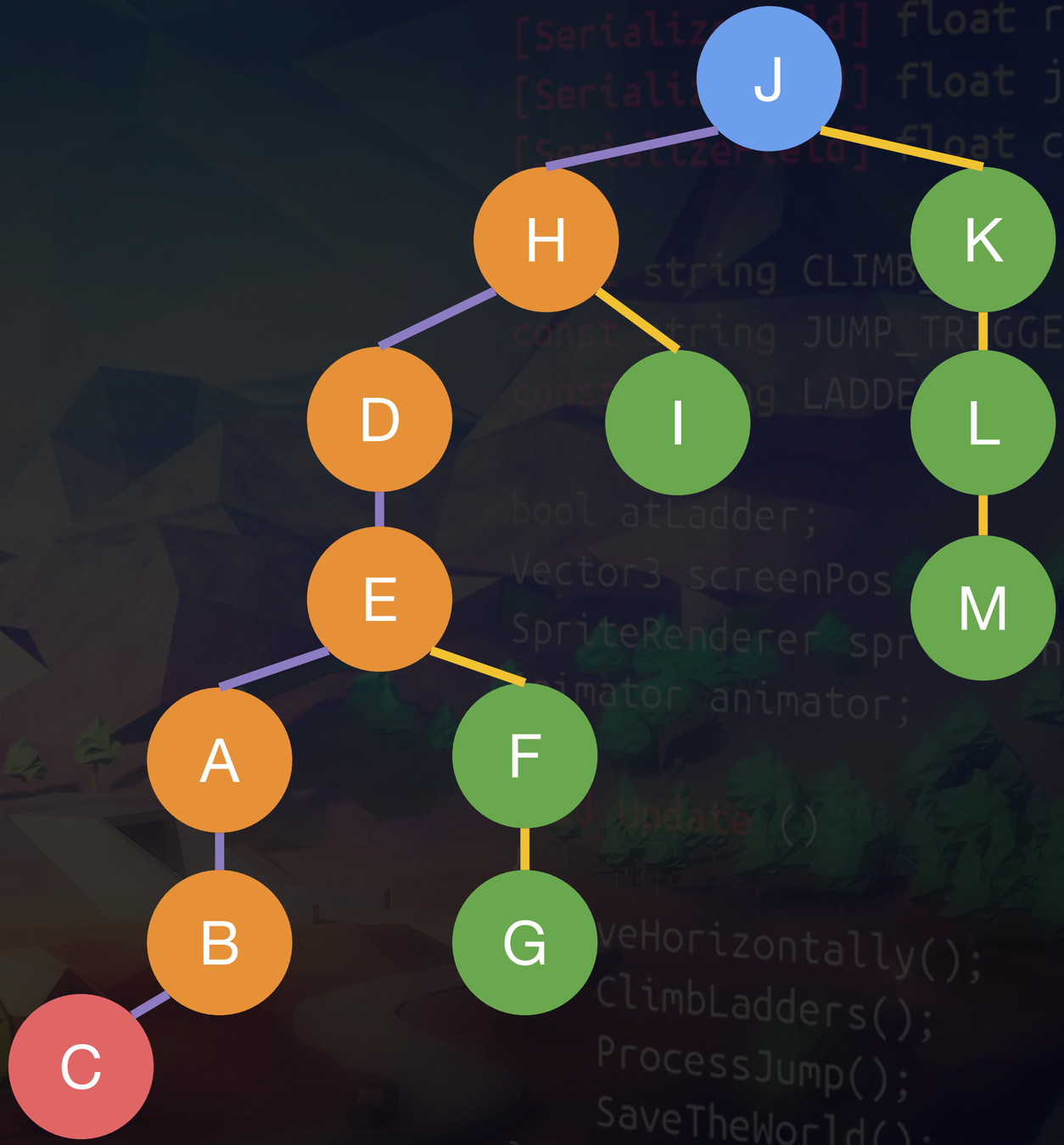
Option 1





Option 2

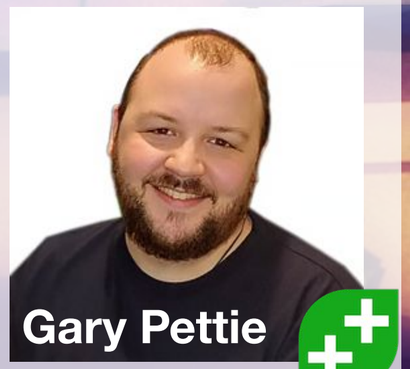




GameDev.tv

Unity 2020.1

Pure C# Classes

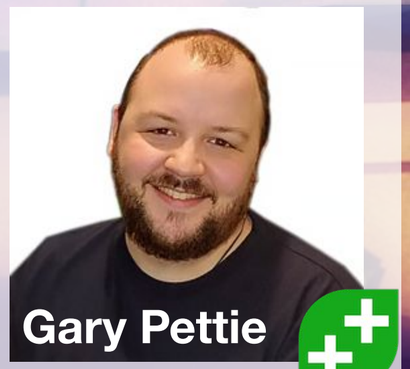


Gary Pettie

GameDev.tv

Unity 2020.1

Dictionaries



Gary Pettie

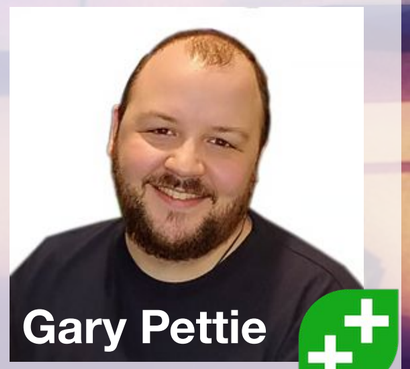
What are dictionaries?

- Stores a key-value pair
- **Keys** link to **values**
- The **keys** must be unique, and are usually very simple
- The **values** can be more complex types
- **Values** can be null
- **Keys** cannot be null
- The lookup is very fast from key to value
- The lookup is much slower from value to key.

GameDev.tv

Unity 2020.1

More Debugging Tools

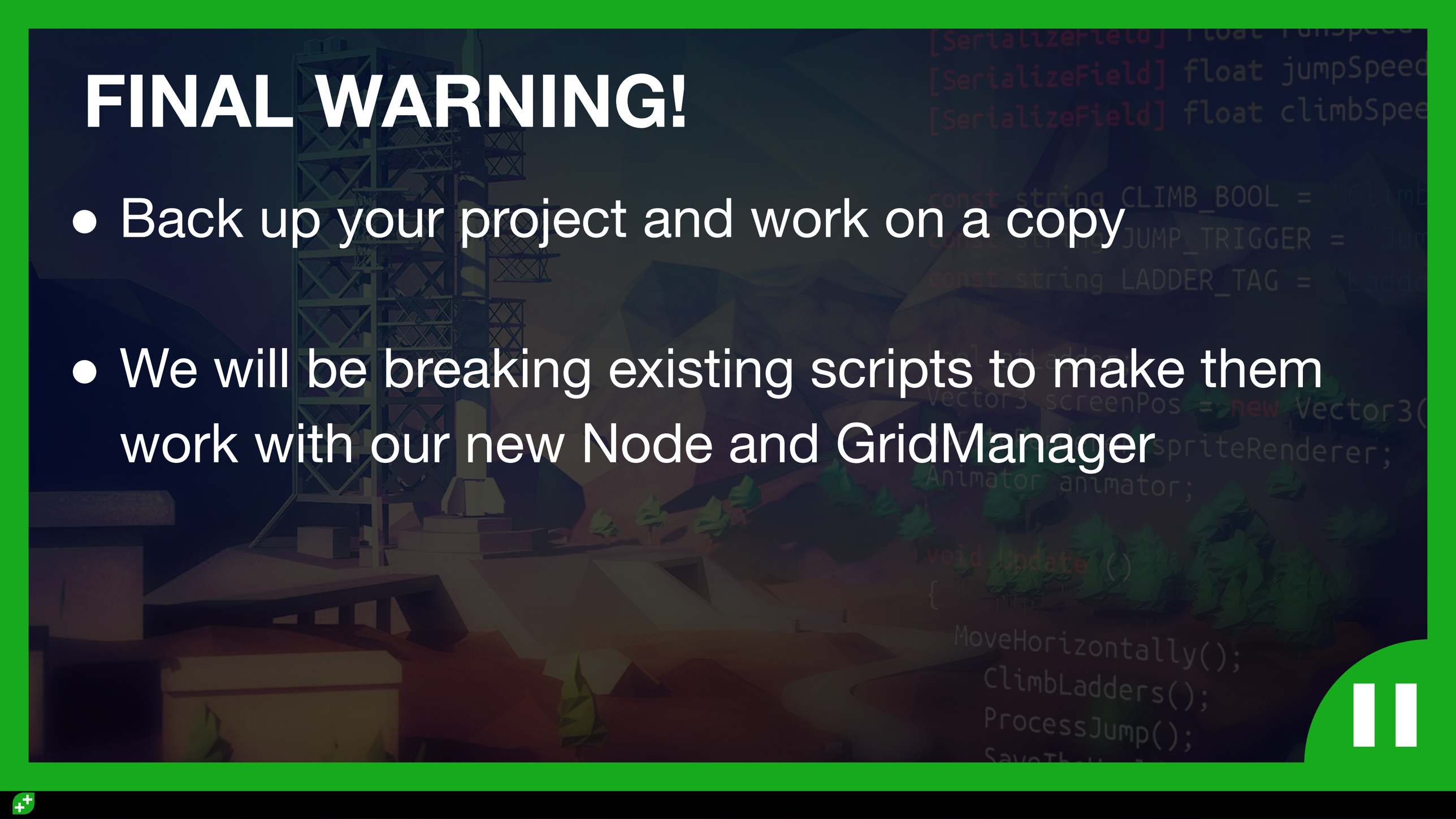


Gary Pettie

FINAL WARNING!

- Back up your project and work on a copy

- We will be breaking existing scripts to make them work with our new Node and GridManager



Finish SetLabelColor()

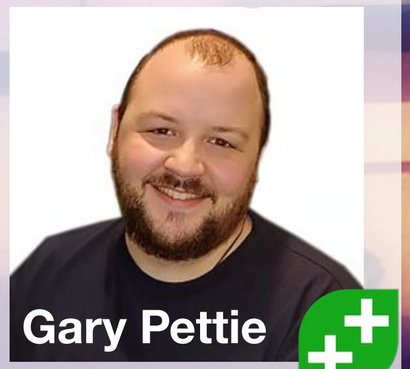
- We have three colors were yet to use:
  - pathColor
  - exploredColor
  - defaultColor

- Think about the best order for these to be checked

GameDev.tv

Unity 2020.1

Exploring Neighbors



Gary Pettie

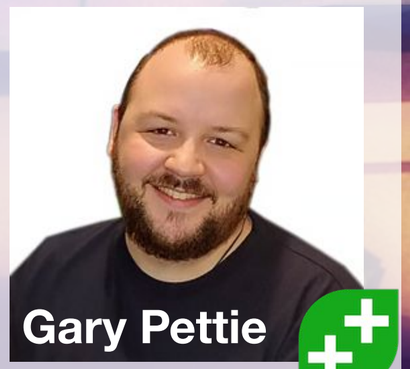
Explore Neighbors

- Create an empty list called “neighbors”
- Loop through all for directions
- Calculate the coordinates of the node in that direction from our currentSearchNode
- Check if the neighbor’s coordinates exist in the grid
- If it does exist in the grid, add it to our neighbors list

GameDev.tv

Unity 2020.1

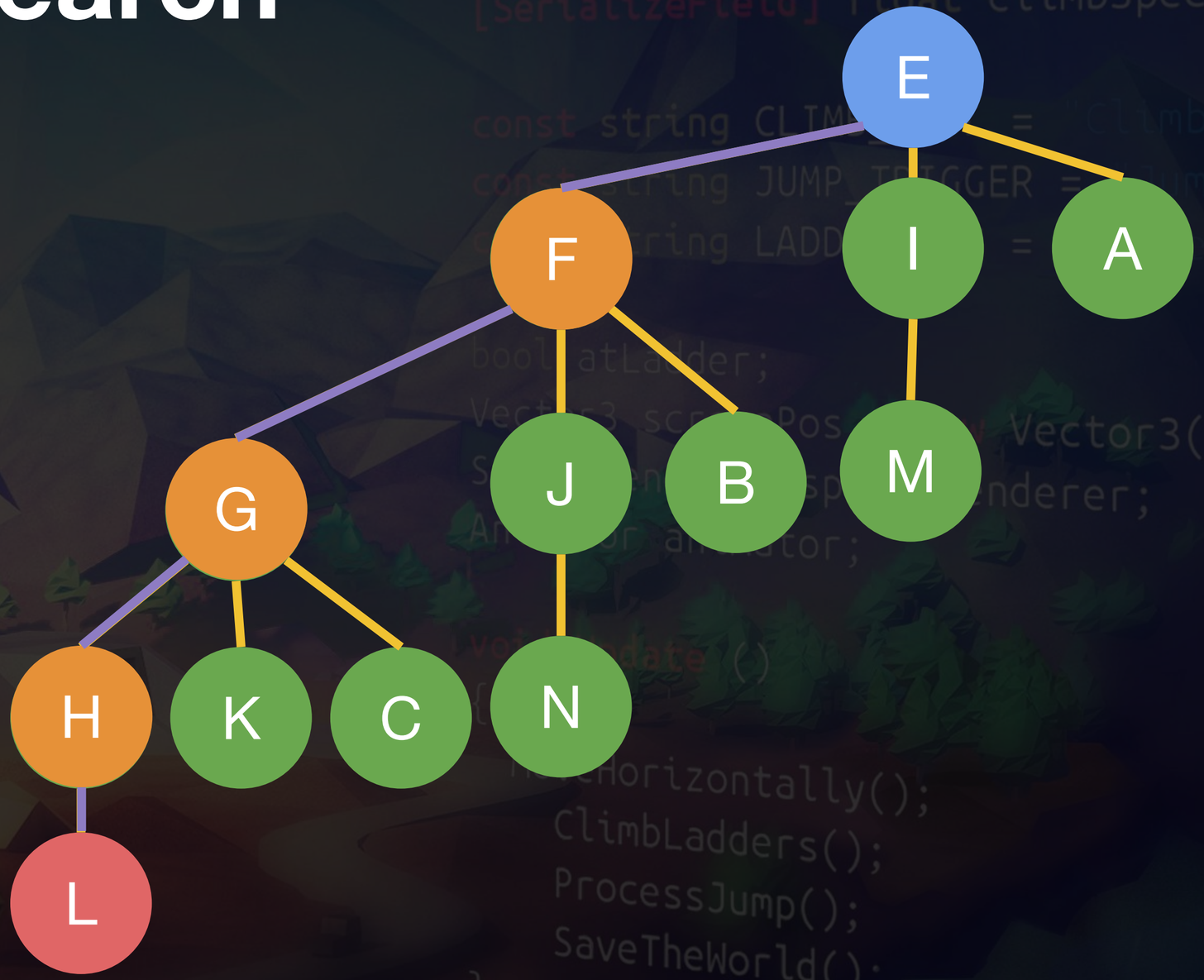
Exploring the World



Gary Pettie

Breadth First Search

M	N	O	P
I	J	K	L
E	F	G	H
A	B	C	D



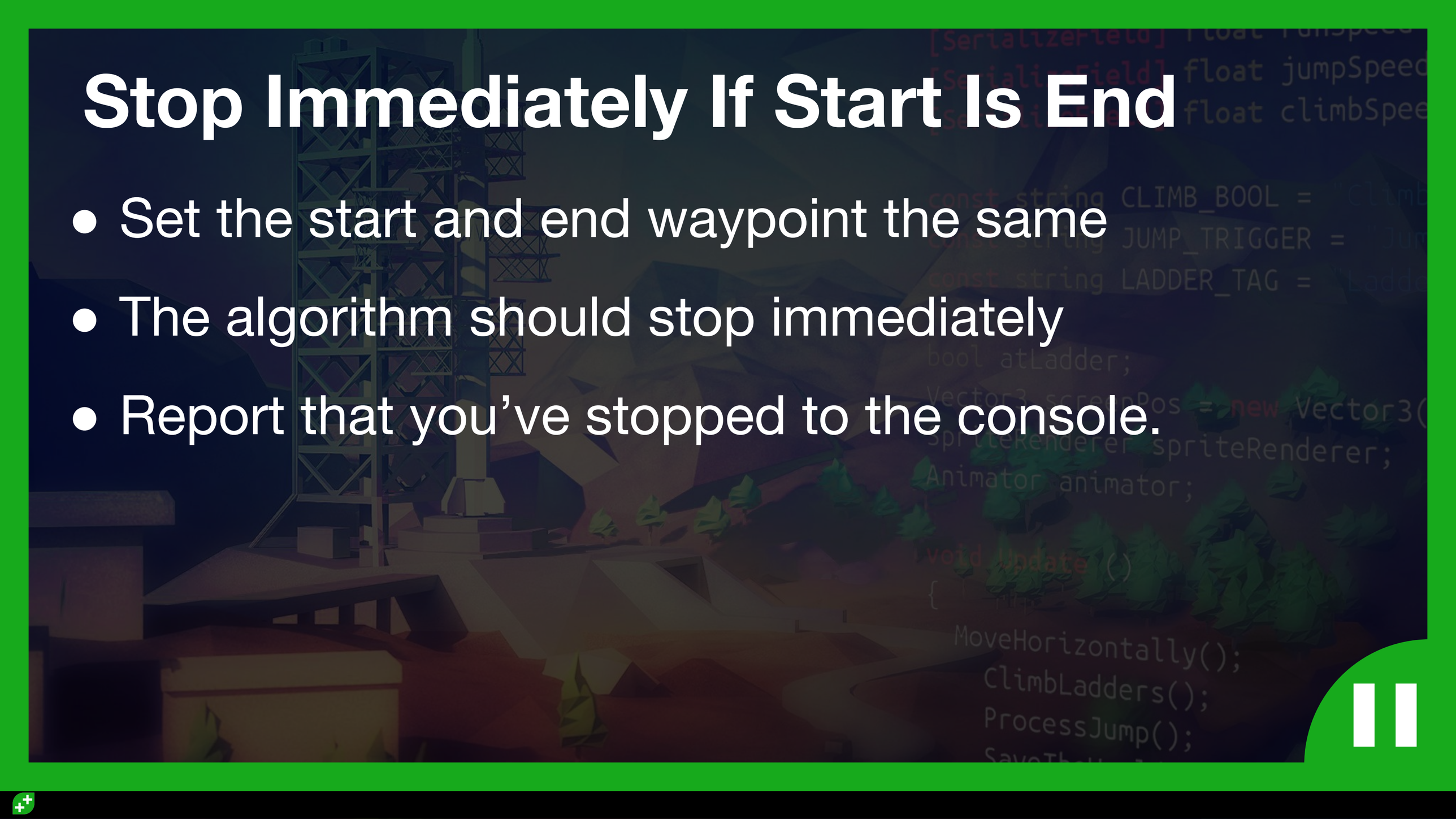
Queues

- A special kind of list
- Enforces a FIFO order

- `Queue<Node> queue`
- `queue.Enqueue()` adds to end of queue
- `queue.Dequeue()` returns the front of the queue
- `queue.Peek()` look but don’t touch

Stop Immediately If Start Is End

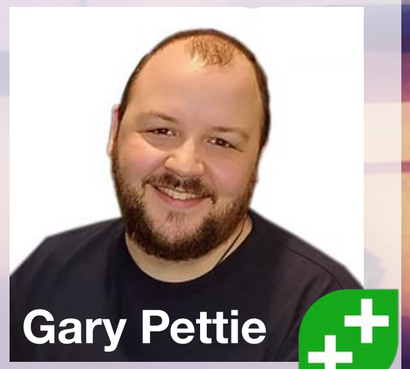
- Set the start and end waypoint the same
- The algorithm should stop immediately
- Report that you've stopped to the console.



GameDev.tv

Unity 2020.1

Finding the Path (again!)

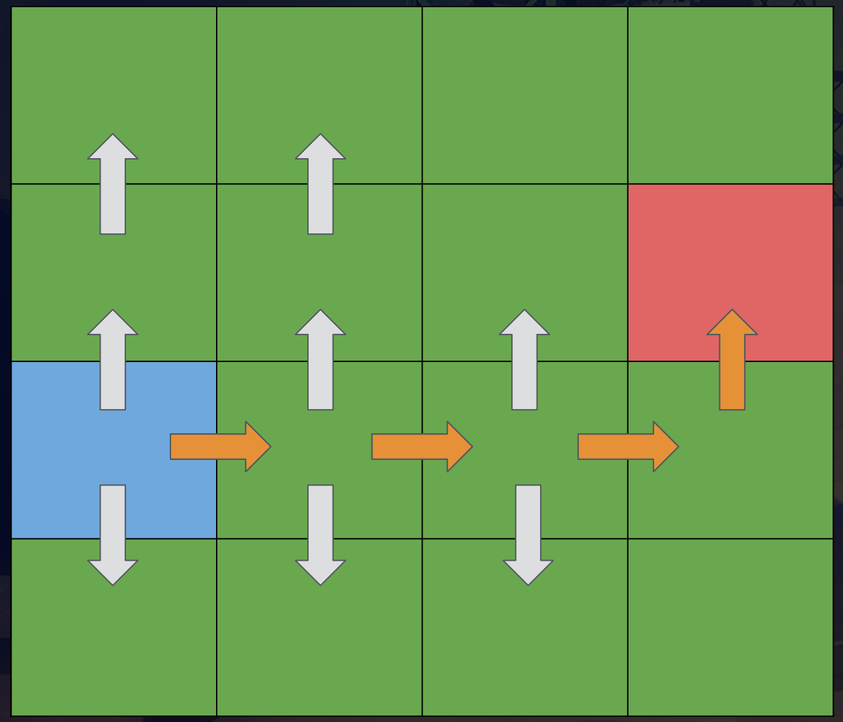


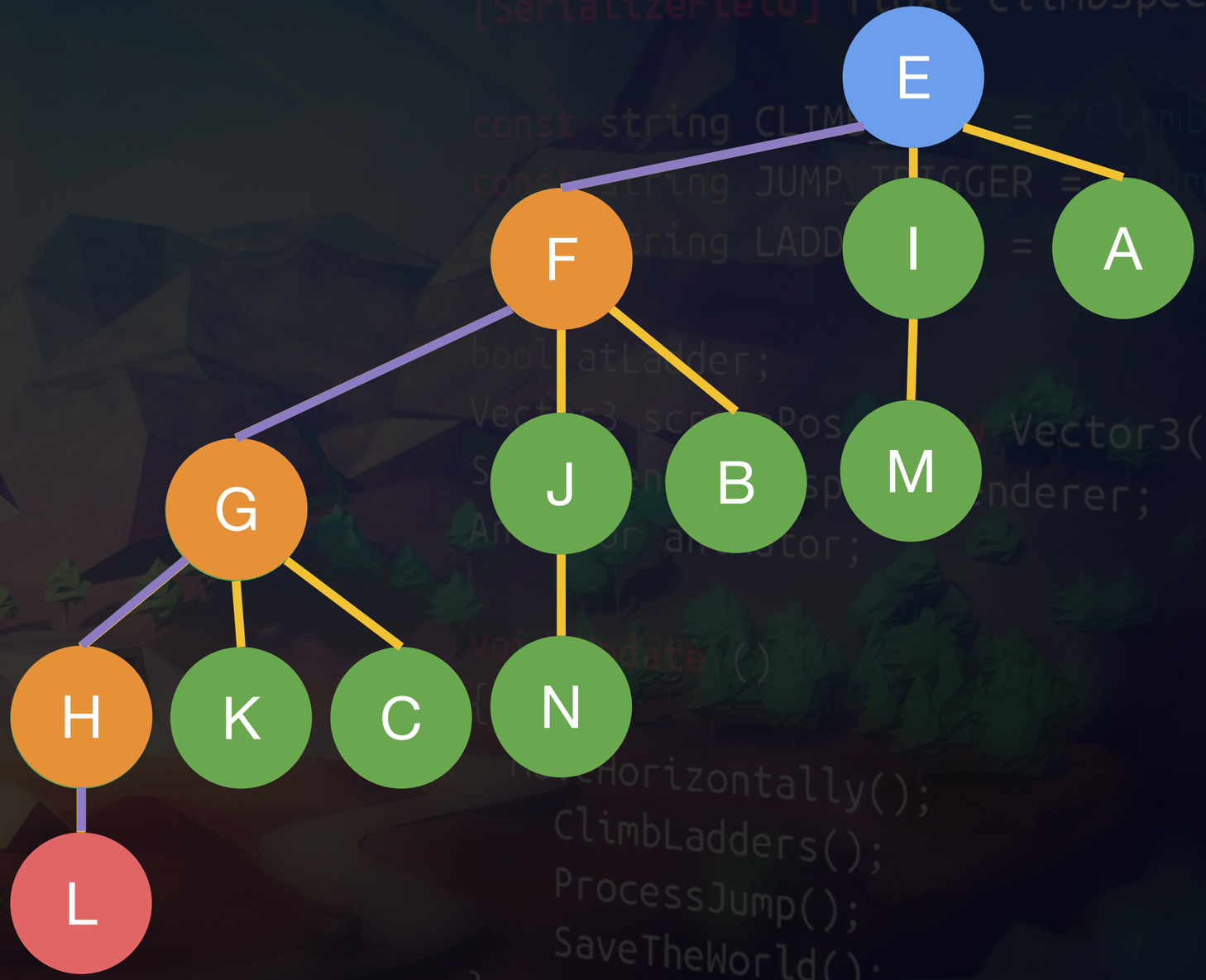
Gary Pettie

Crawling the Map

M	N	O	P
I	J	K	L
E	F	G	H
A	B	C	D

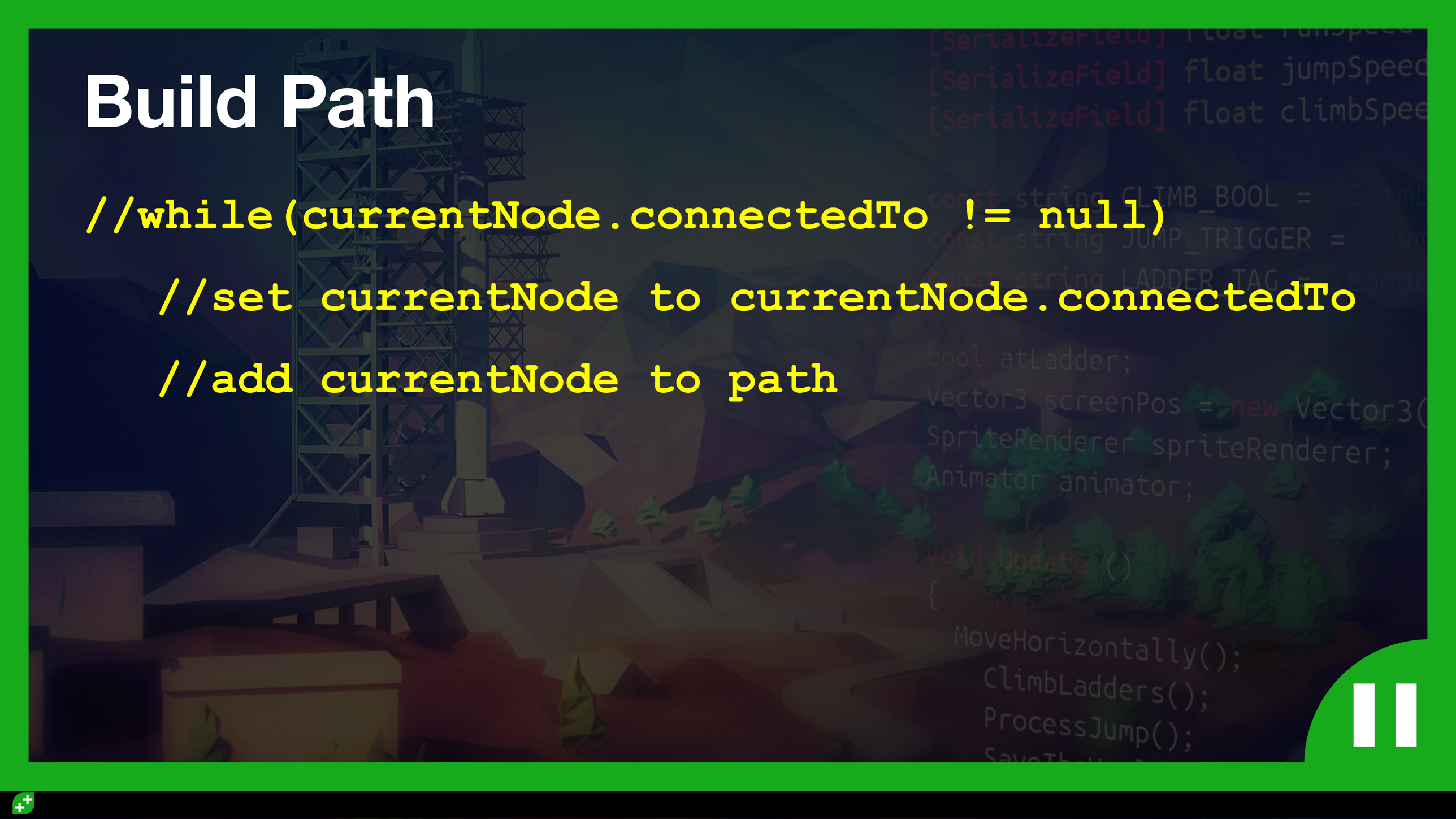
Flow Field





Build Path

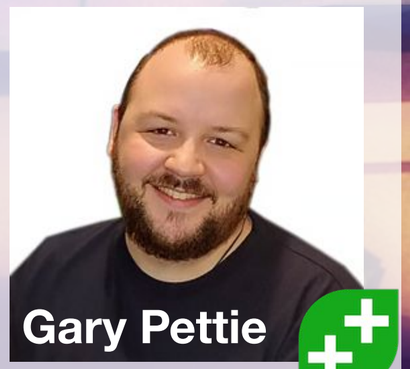
```
//while(currentNode.connectedTo != null)
    //set currentNode to currentNode.connectedTo
    //add currentNode to path
```



GameDev.tv

Unity 2020.1

Blocking Nodes



Gary Pettie

GetPositionFromCoordinates()

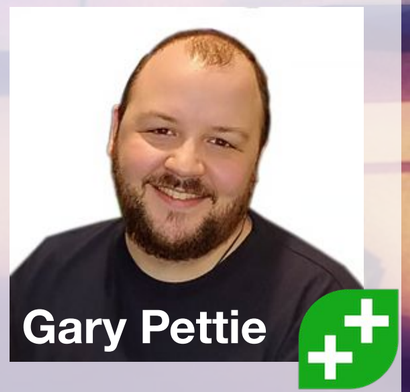
Hint:

- It has a similar structure to GetCoordinatesFromPosition()
- You'll need to do some algebra to rearrange things in terms of the position instead of the coordinates

GameDev.tv

Unity 2020.1

Valid Path

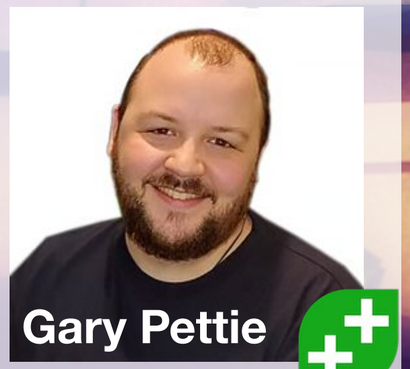


Gary Pettie

GameDev.tv

Unity 2020.1

Script Execution Order



Gary Pettie

Challenge

- Set the endPosition to be the position of the next node in the path

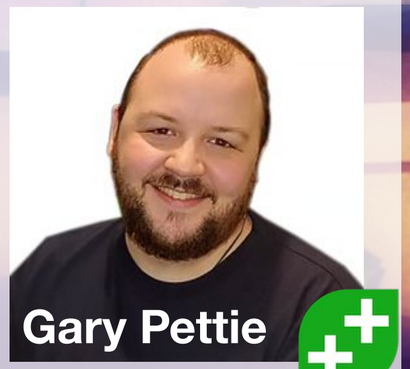
Hint:

- We used similar code in ReturnToStart()

GameDev.tv

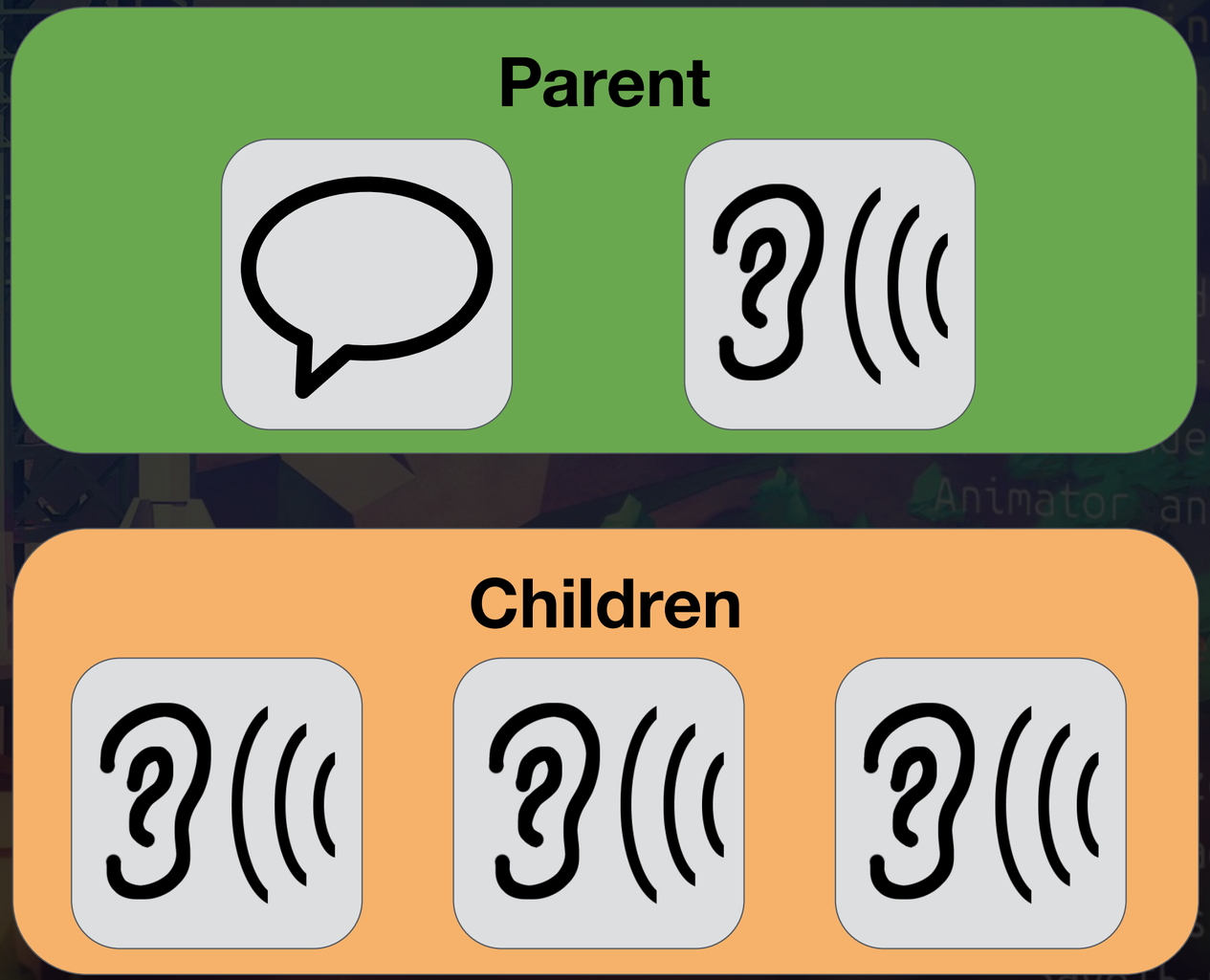
Unity 2020.1

Broadcast Messages



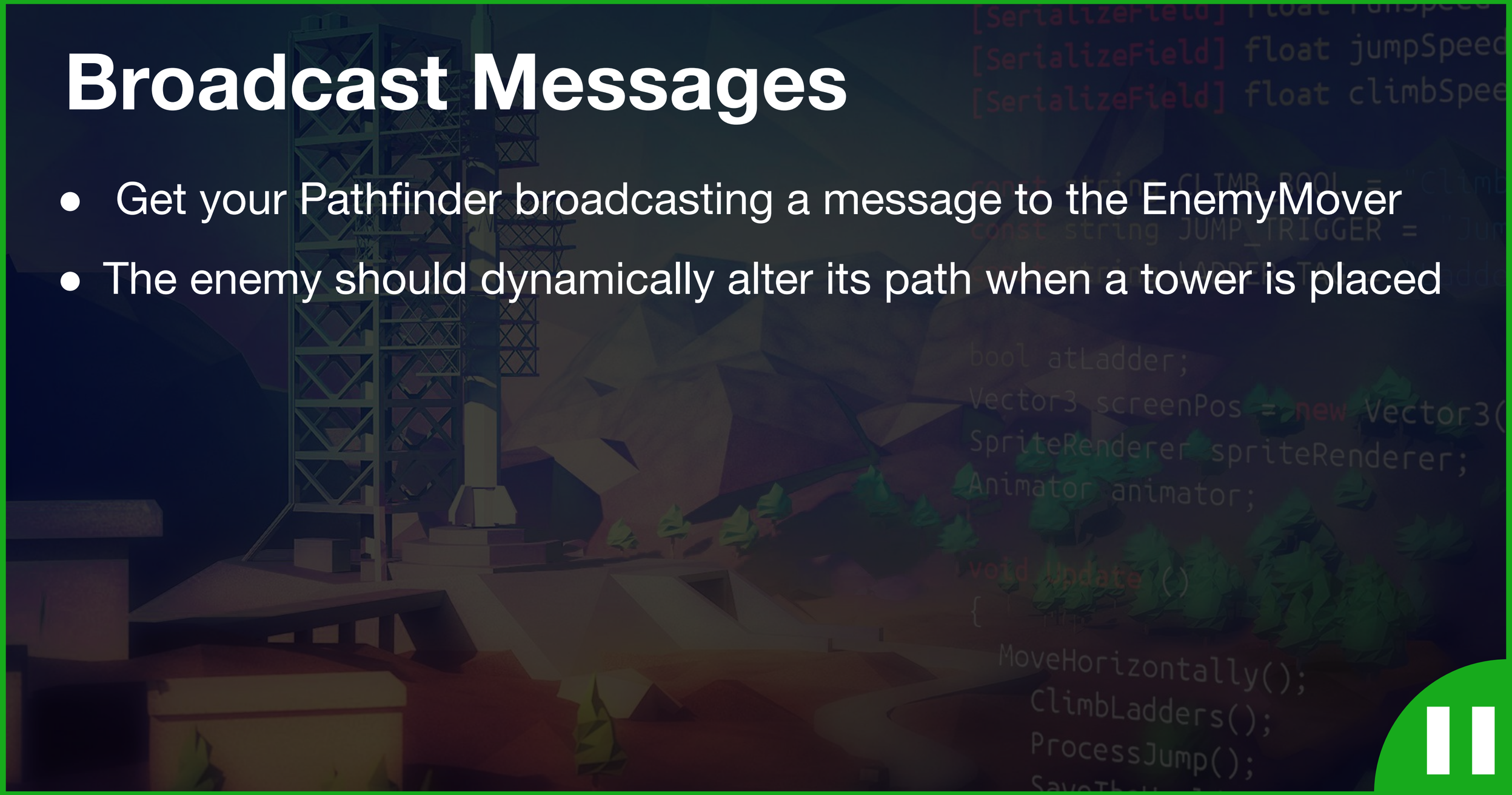
Gary Pettie

BroadcastMessage(“Dinner”)



Broadcast Messages

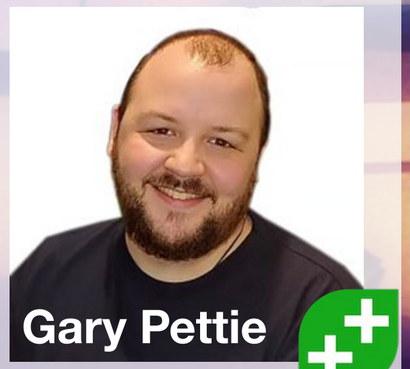
- Get your Pathfinder broadcasting a message to the EnemyMover
- The enemy should dynamically alter its path when a tower is placed



GameDev.tv

Unity 2020.1

Overloading Methods



Gary Pettie

Overloaded Methods

- A type of polymorphism
- ‘poly’ = many
- ‘morphism’ = shapes
- Methods can have the same name if they have different signatures

e.g.

```
Instantiate(object)
Instantiate(object, position, rotation)
etc.
```

Overload GetNewPath()

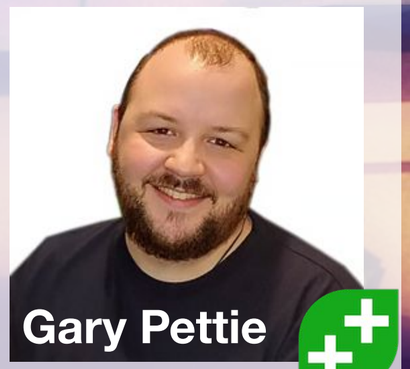
- Write a new version of GetNewPath() that takes the current coordinates as an arguement

- GetNewPath() should pass startCoordinates to BreadthFirstSearch()
- The overloaded method should pass the provided coordinates

GameDev.tv

Unity 2020.1

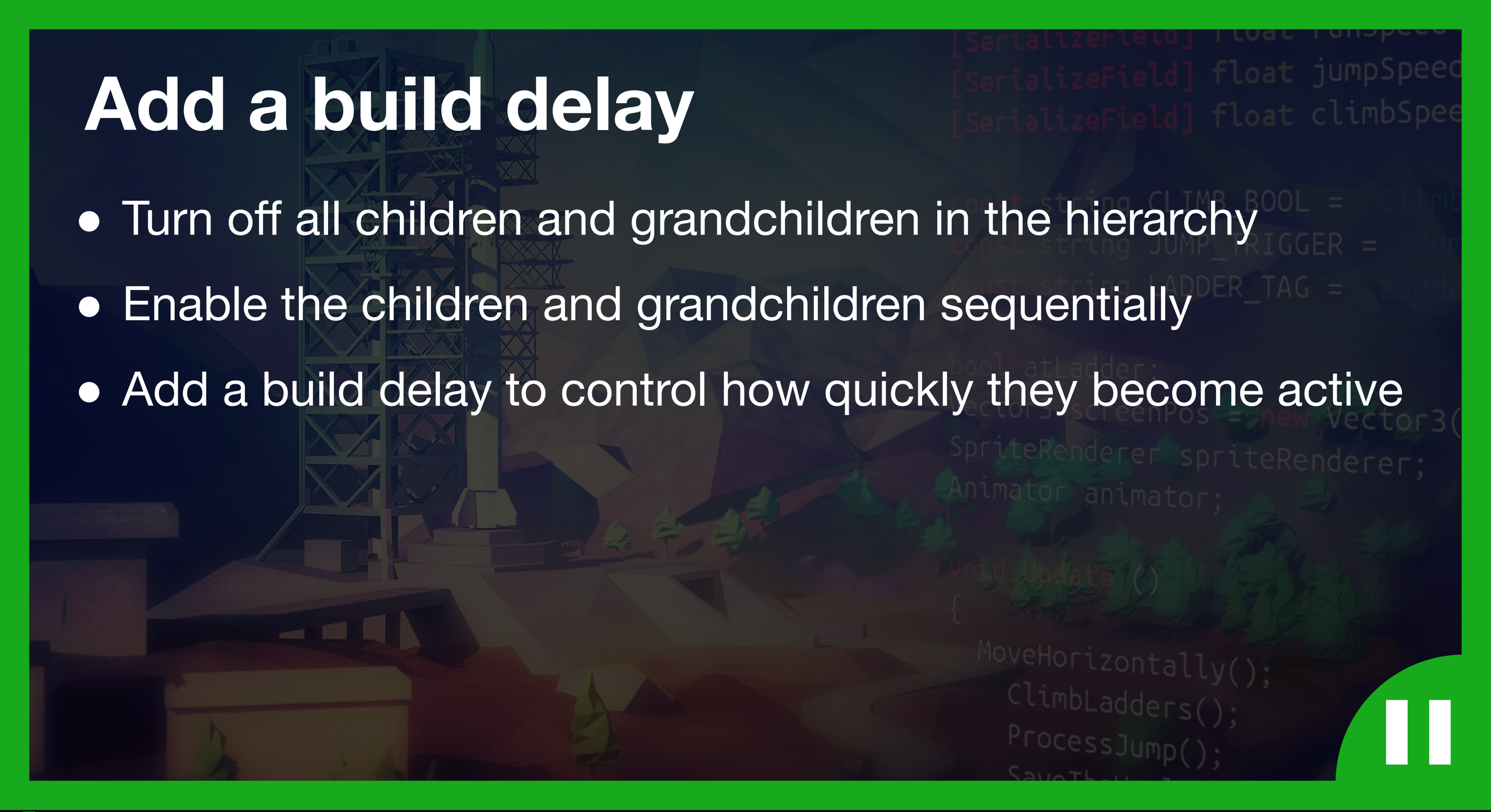
Build Timer



Gary Pettie

Add a build delay

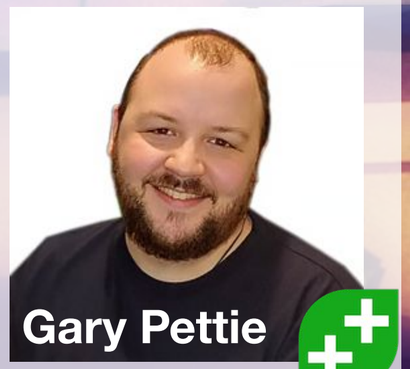
- Turn off all children and grandchildren in the hierarchy
- Enable the children and grandchildren sequentially
- Add a build delay to control how quickly they become active



GameDev.tv

Unity 2020.1

Ambience



Gary Pettie

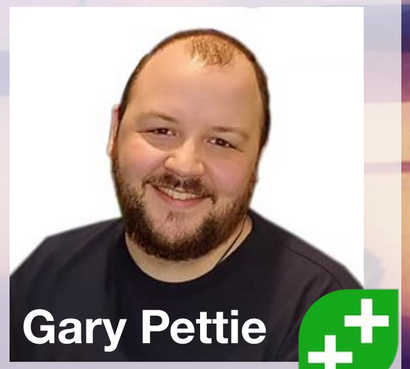
Challenge

- Add some additional lights to your game
- Add a few small particle effects to add interest to your level

GameDev.tv

Unity 2020.1

Post Processing

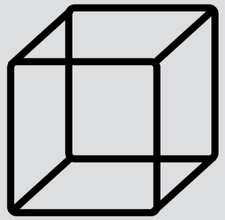


Gary Pettie

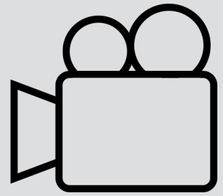
Post Processing

- In camera effects used to change the look of your game

Profile



Volume



Layer

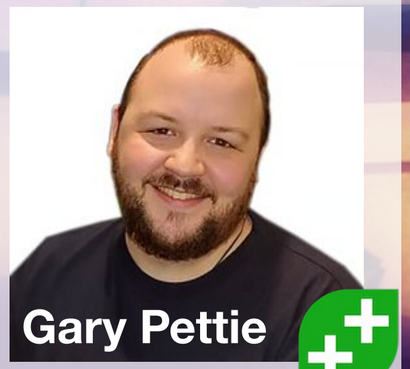
Post Processing Stack

- Play around with all of the different effects
- Make your game look as good as possible
- Share screenshot with the community!

GameDev.tv

Unity 2020.1

Wrap Up - Realm Rush



Gary Pettie


[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;
MoveHorizontally();
ClimbLadders();
ProcessJump();
SaveTheWorld();
}

OLD SLIDES

GameDev.tv

Unity 2020.1

Limitations Of Unity Pathfinding

Convince Yourself Of Limit

- You should feel clear why we’re starting from scratch.
- Feel free to discuss in forums, or on our Discord chat server.

GameDev.tv

Unity 2020.1

Types Of Pathfinding

GameDev.tv

Unity 2020.1

The Path Ahead

What to expect in the next 2 lectures

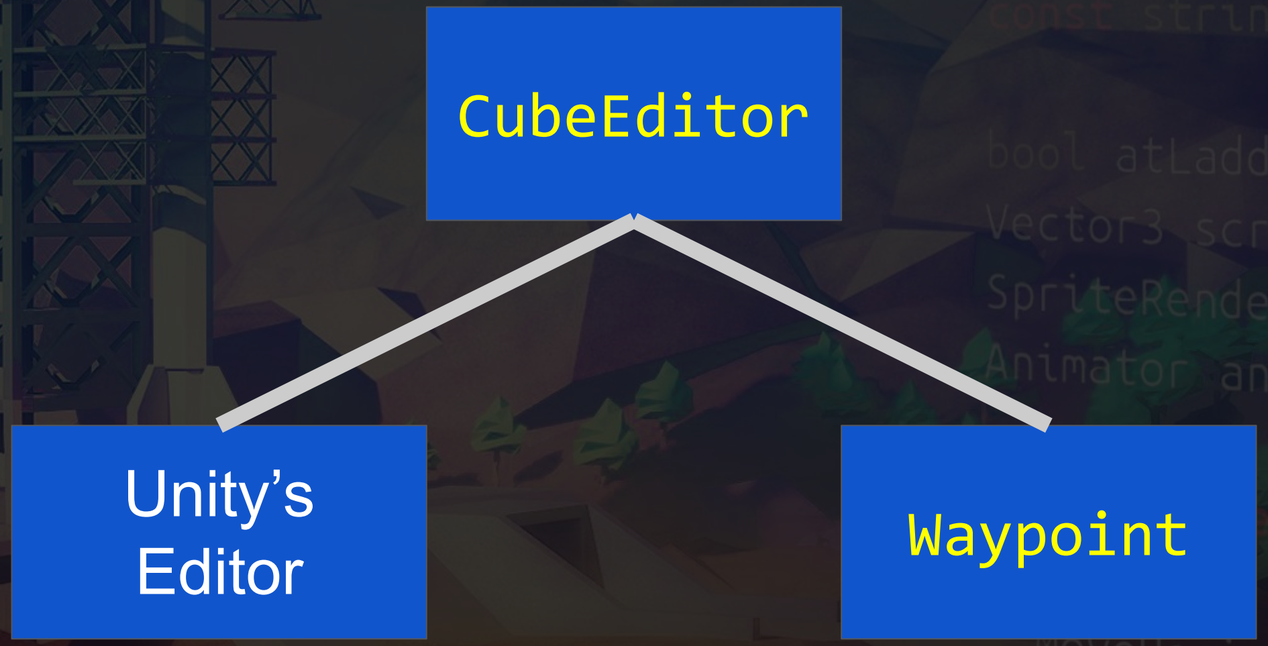
- Lots of refactoring
- Thinking about instance variables vs constants
- Try and zoom out and see the purpose
- Engage on the community.

GameDev.tv

Unity 2020.1

Instance Variables And Constants

Dependencies



Re-wire Snapping & Labelling

- Snapping to grid in editor works
- The cube labelling works
- The game plays even with all Cube Editor scripts disabled.

How we're organising our data

- The `gridSize` should be in just one place
- Instance (member) variables aren't this one place
- Revisiting `const` from Terminal Hacker
- Being careful about what's dependant on what.

GameDev.tv

Unity 2020.1

The Dictionary Data Structure

What are dictionaries?

- Think of them like the index of a book
- **Keys** (words) link to **values** (pages)
- The **keys** must be unique, and are usually simple
- The **values** can be more complex types
- The lookup is very fast from key to value
- The lookup is much slower from value to key.

Use dictionaries when...

- You have simple values that have meaning in their value, such as the position of a waypoint
- This meaningful value must be unique, for example one waypoint per grid position
- You want to associate this unique value with some other value, such as the block at that position.

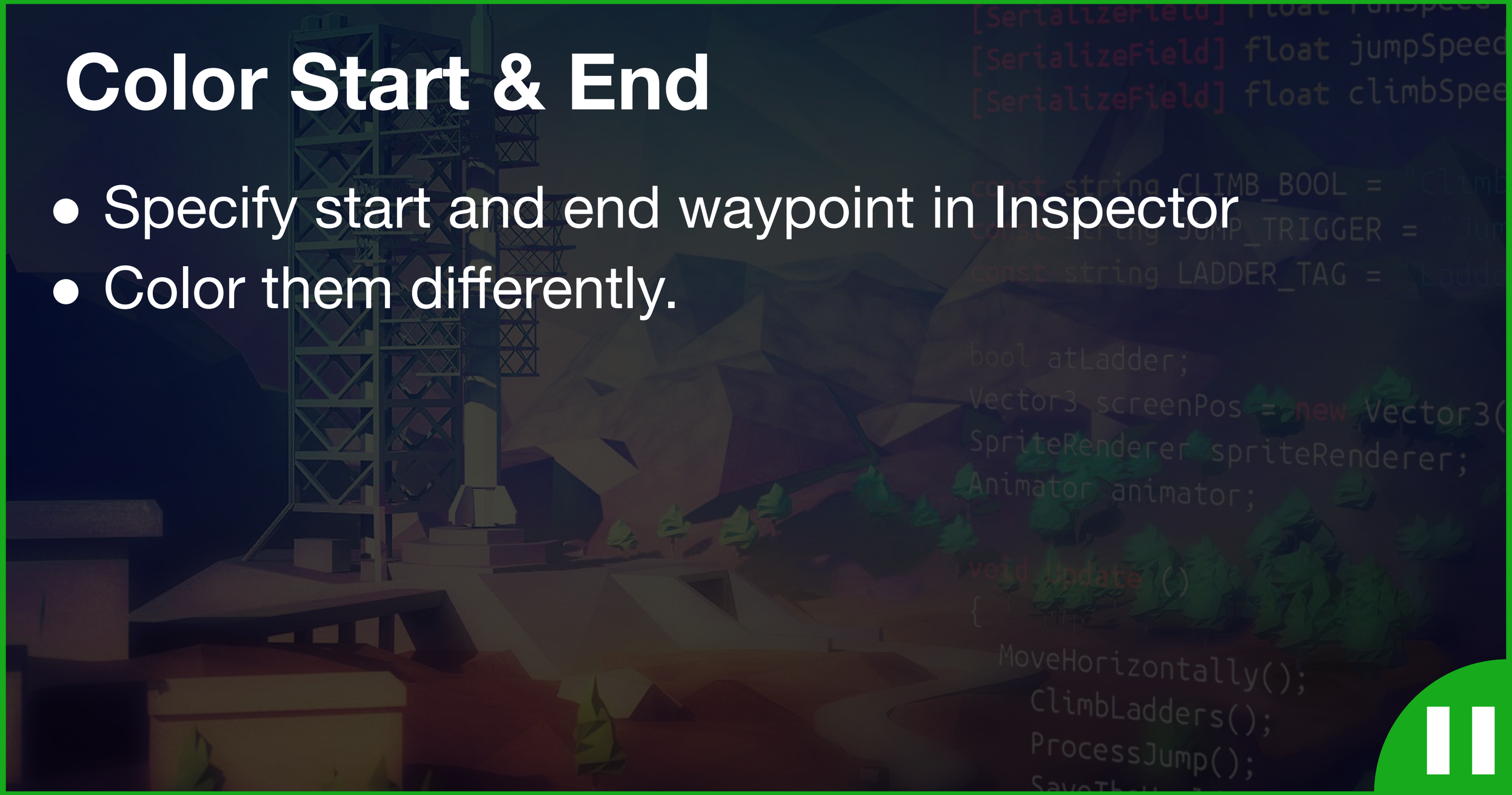
GameDev.tv

Unity 2020.1

Finding Game Objects By Name

Color Start & End

- Specify start and end waypoint in Inspector
- Color them differently.



Print exploration log

- Console should read “Exploring X,Y” for each neighbour
- Don’t worry if there’s a waypoint there or not
- Explored coordinates should change if you change the start block.

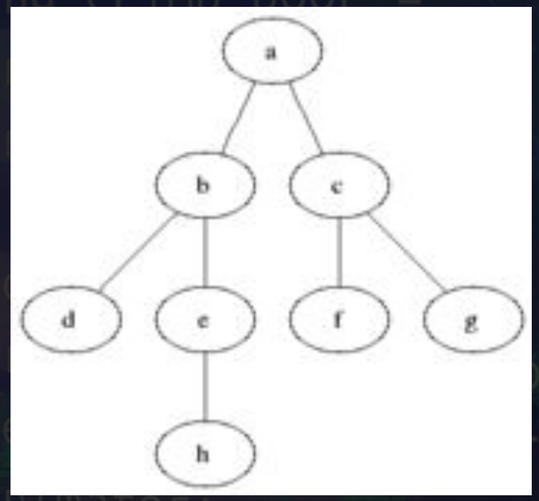
GameDev.tv

Unity 2020.1

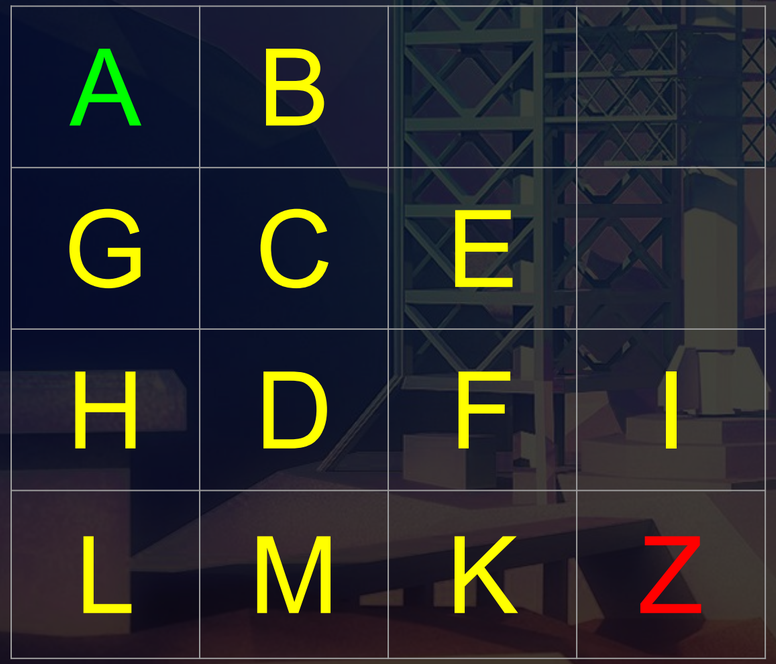
Breadth First Search Algorithm

Breadth First Search Pseudocode

```
// en-queue the start waypoint
// while the queue has items
    // if we find the goal stop
    // de-queue frontier waypoint
    // for each direction from frontier
        // queue new unexplored waypoint
    // mark frontier as explored
```



What Order Would This Map Search?



- What was the queue order?
- What is the shortest path?
- Assume directions are up, right, down, left in that order
- Assume starts at A, stops at Z
- Share your answers and reasons with the community.

GameDev.tv

Unity 2020.1

Introducing C# Queues

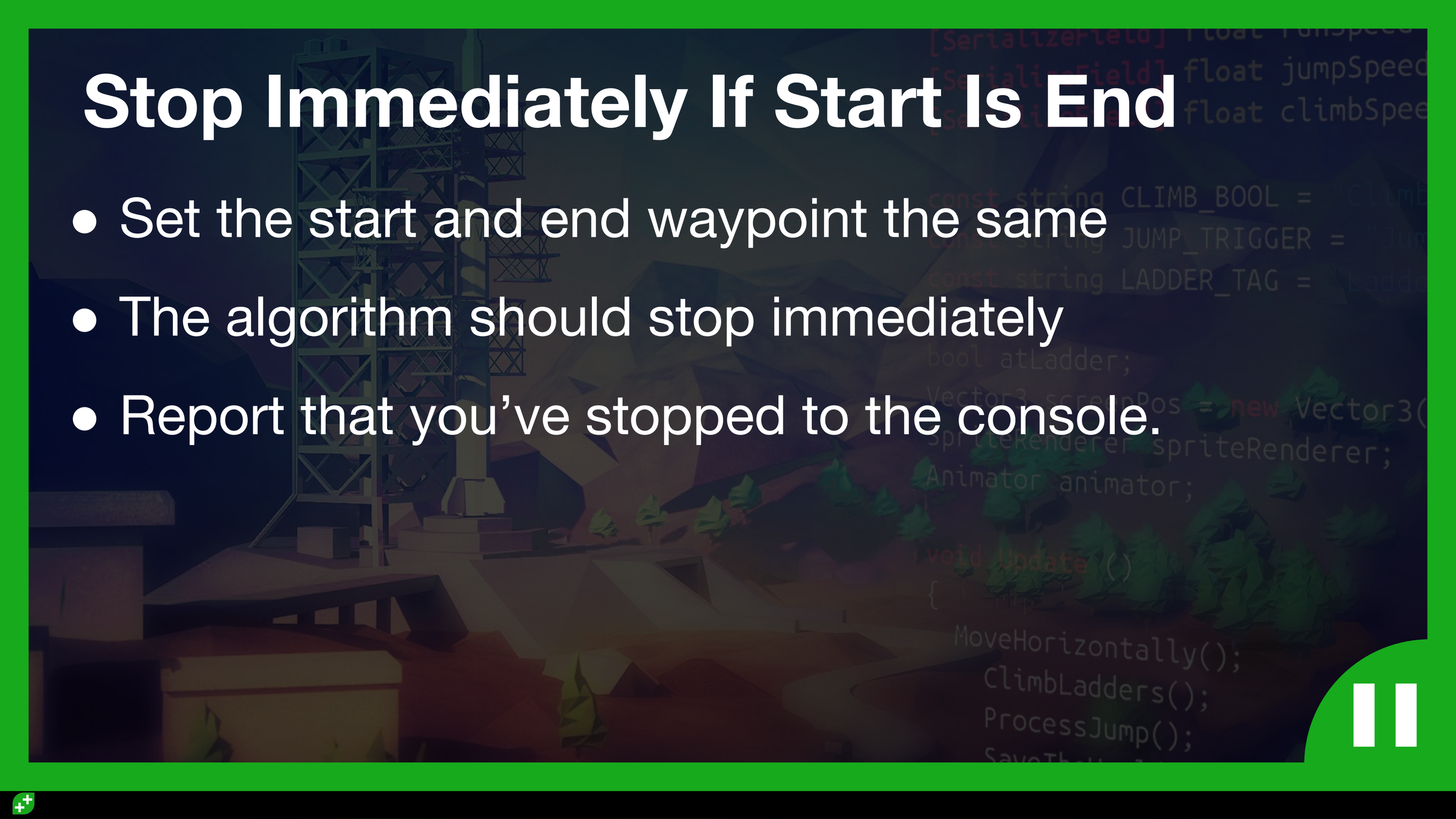
FIFO vs LIFO

- FIFO = First In, First Out (e.g. dinner queue)
- LIFO = Last In, First Out (e.g. stack of plates)

- Queues in C# are FIFO
- `Queue<Waypoint> queue`
- `queue.Enqueue()` adds to end of queue
- `queue.Dequeue()` returns the front of the queue

Stop Immediately If Start Is End

- Set the start and end waypoint the same
- The algorithm should stop immediately
- Report that you've stopped to the console.



GameDev.tv

Unity 2020.1

Running Manual Tests

A Note About `public`

- If a class just stores data (e.g. the `Waypoint`) then I find it ok to make state (e.g. `isExplored`) public

- This saves writing trivial methods like `SetExplored()` and `GetExplored()`

Verify The Pathfinding Operation

- Verify the operation of our pathfinder for yourself.

GameDev.tv

Unity 2020.1

A Breadcrumb Trail

When it’s OK to use public

- When you would simple write `SetX()` and `GetX()` anyway, you may as well just make x public!

Optional Challenge

- Make the Waypoint update its own color.

GameDev.tv

Unity 2020.1

Reversing A List

How To Calculate Path

1. Start at the end waypoint
2. Add it to a list
3. Add all intermediate waypoints in a loop
4. Add the start waypoint
5. Reverse the list.

Finish the Create Path Method

- The list in the inspector should be correct
- Use `path.Reverse()` to reverse order
- Try it on a few different maps
- Celebrate pathfinding!

GameDev.tv

Unity 2020.1

Adding Production Assets

Add Assets To Your Scene

- Update your path blocks.
- Add the friendly and enemy structures.
- Replace the enemy.
- Don’t do the tower just yet.

GameDev.tv

Unity 2020.1

Tower LookAt Enemy

Tower LookAt Enemy

- Write the method for the tower to look at the enemy.
- who.LookAt(what)

GameDev.tv

Unity 2020.1

MagicaVoxel Bonus Content

MagicaVoxel Community Course

1. Course content for our community.
2. Outside of this course because some people aren’t interested in asset creation.
3. We received a very positive response on our Facebook post.

GameDev.tv

Unity 2020.1

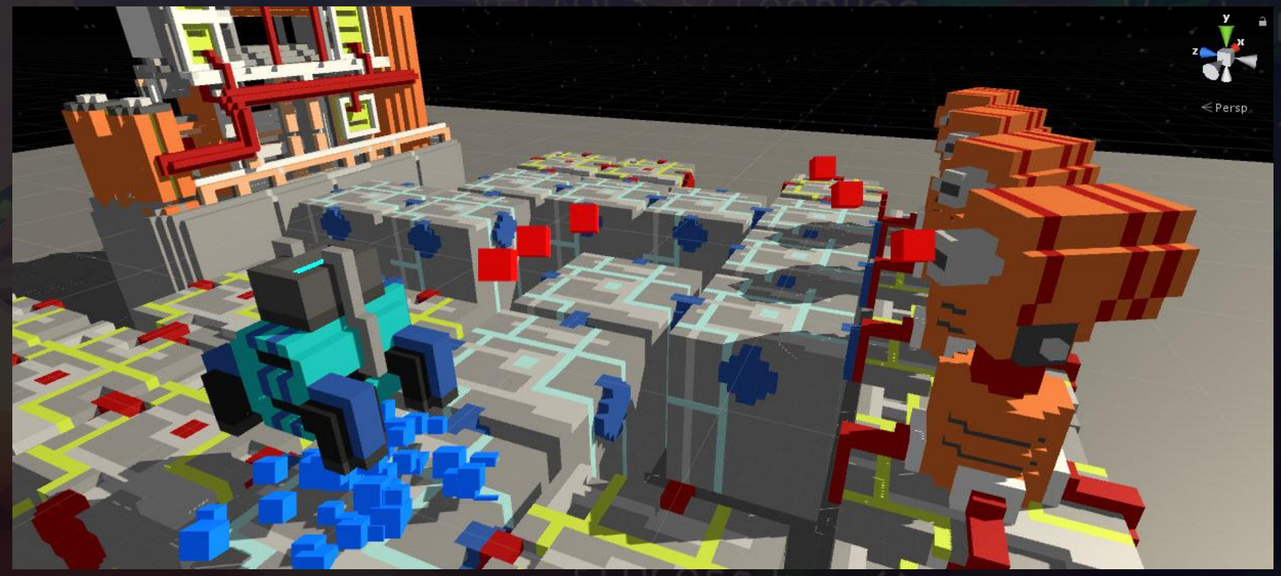
Another Solo Challenge

Main Feature Buckets

1. Shoot & damage enemies
2. Place towers
3. Spawn enemies
4. Damage base

Considerations For Shooting

1. Create particle system for bullets
2. Collision on particles and on enemies
3. Start and stop shooting based upon enemy distance
4. Do damage to enemy
5. Visualise damage



Solo Challenge

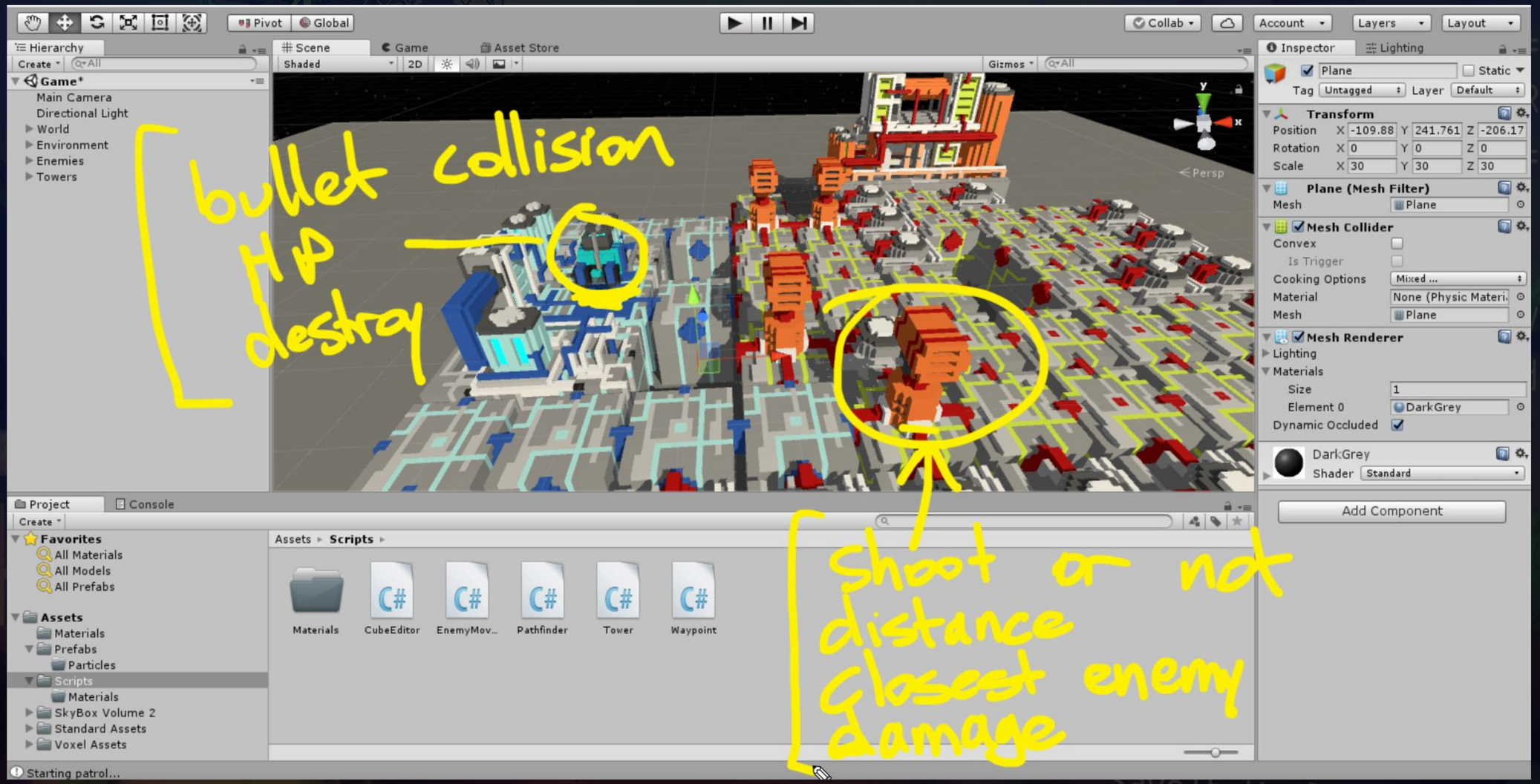
- Using Argon Assault as reference, implement:
  - Towers shoot projectiles
  - Projectiles collide with enemy
  - Damage effect is triggered
  - Enemies lose hit points
  - Enemies die

GameDev.tv

Unity 2020.1

Make Tower Shoot

Content Of Scripts



GameDev.tv

Unity 2020.1

Enemy HitPoints

Kill Enemy

- If the enemy’s hitpoints are less than 1, destroy the enemy.

GameDev.tv

Unity 2020.1

Check For Distance

Shoot If Enemy Is Close

- Follow our logic and shoot if enemies are less than X distance away.

```
Vector3.Distance(enemy, tower);
```

Logic For Shooting

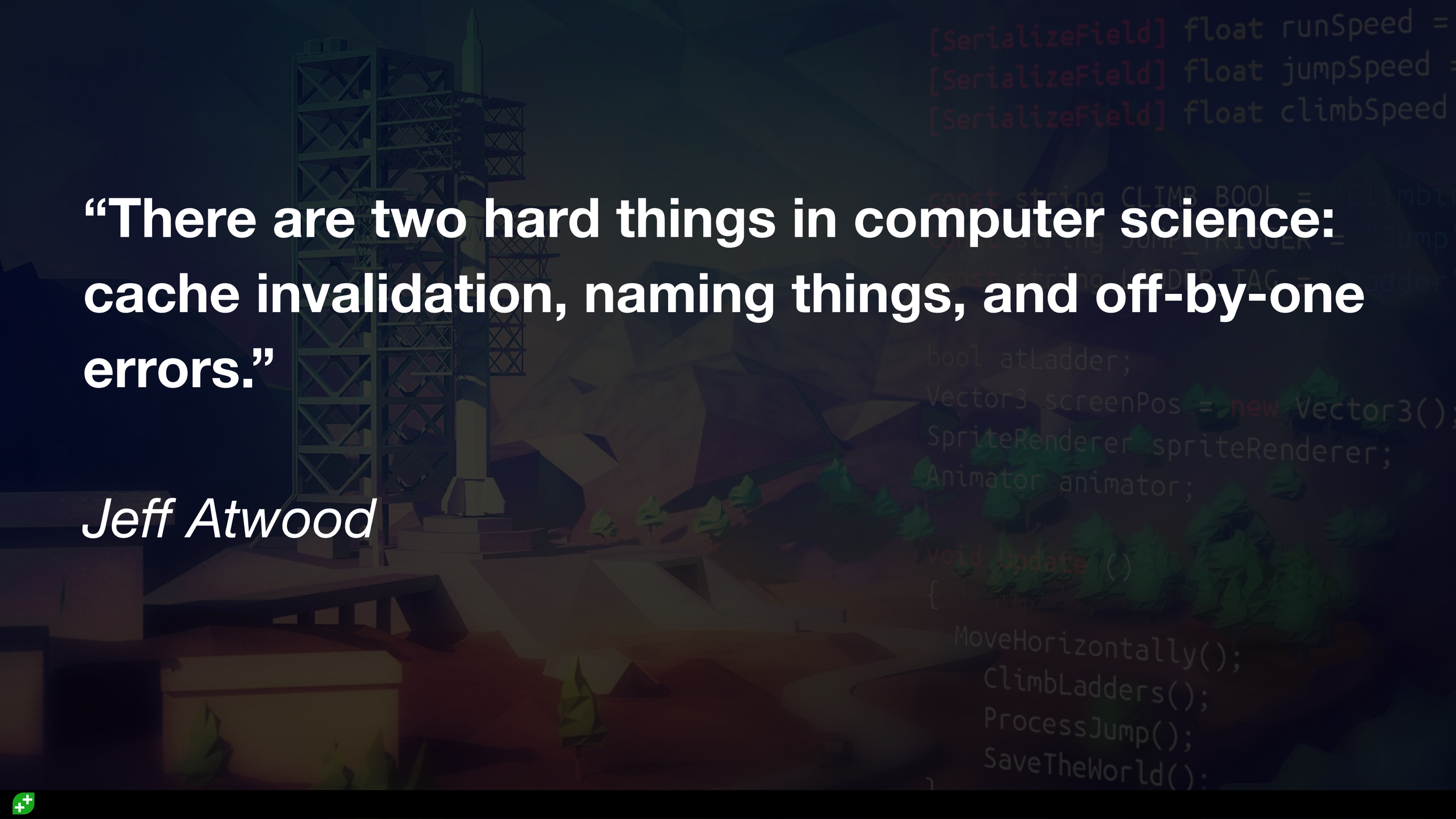
1. Define attack range
2. Check that an enemy exists
3. Check for distance from tower to enemy
4. If distance is less than attack range, fire at enemy
5. Towers shoot bullets by turning on bullet particle system

```
Vector3.Distance(enemy, tower);
```

GameDev.tv

Unity 2020.1

Subtleties Of Spawning



“There are two hard things in computer science: cache invalidation, naming things, and off-by-one errors.”

Jeff Atwood

Store The Path

- If `GetPath()` is called a 2nd time, you should return the path you've already calculated, and not attempt to calculate it again.
- If this is done right, the console errors will go away.

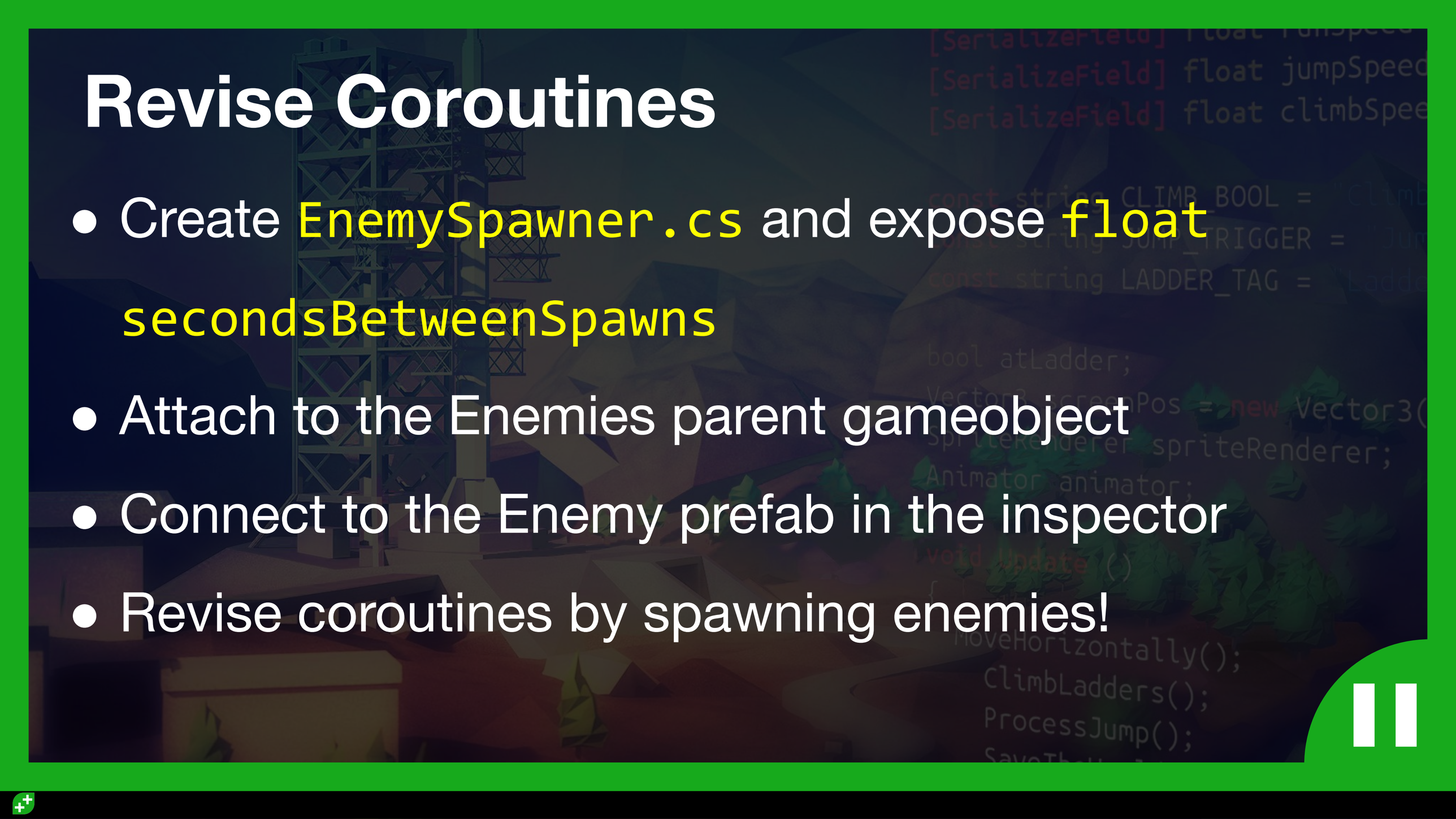
GameDev.tv

Unity 2020.1

Revise Coroutines & Much More

Revise Coroutines

- Create `EnemySpawner.cs` and expose `float secondsBetweenSpawns`
- Attach to the Enemies parent gameobject
- Connect to the Enemy prefab in the inspector
- Revise coroutines by spawning enemies!



GameDev.tv

Unity 2020.1

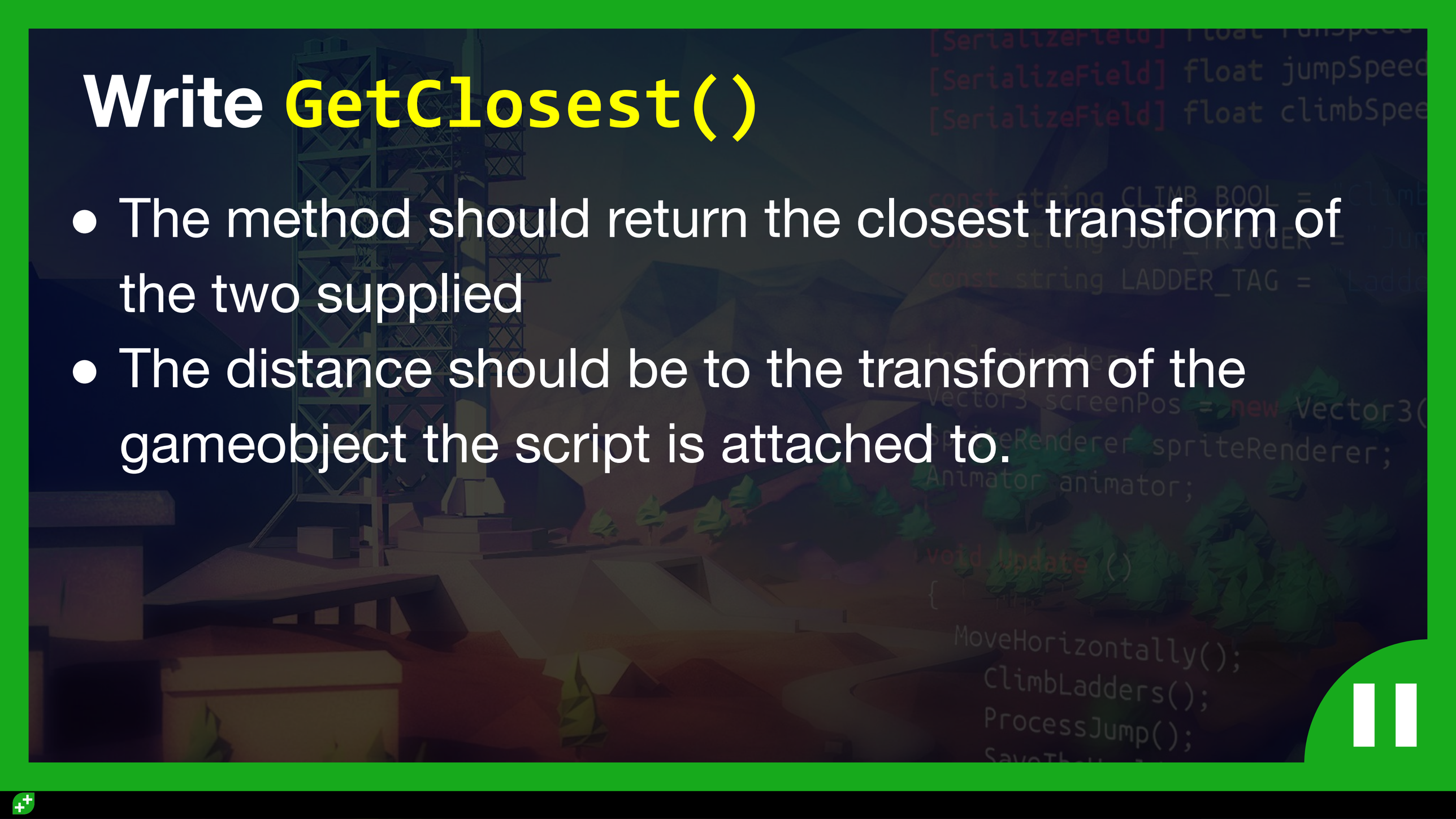
Target Closest Enemy

A recipe for superlatives

```
// Get the collection of things
// Assume the first is the “winner”
// For each item in collection
    // Update winner
// Return the winner
```

Write `GetClosest()`

- The method should return the closest transform of the two supplied
- The distance should be to the transform of the gameobject the script is attached to.



GameDev.tv

Unity 2020.1

Conditional Instantiation

Pattern For Filtering Mouse Clicks

```
if (Input.GetMouseButtonDown(0)) // left click
{
    if (isPlaceable)
    {
        print(gameObject.name + " tower placement");
    }
    else
    {
        print("Can't place here");
    }
}
```

Only Log If Placeable

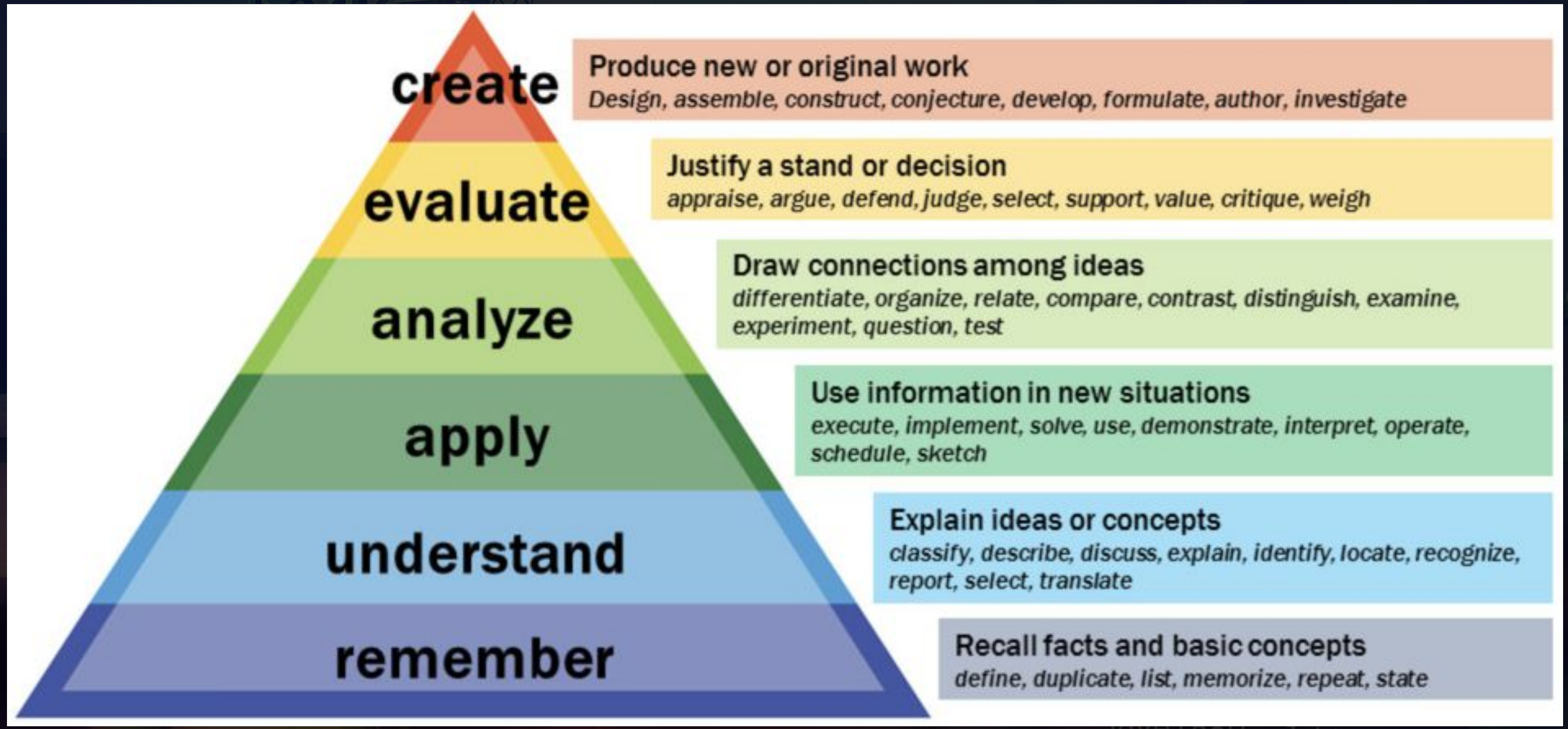
- Simply also “filter” for the block being placeable.

GameDev.tv

Unity 2020.1

Bloom's Taxonomy

Introducing Bloom's Taxonomy



An Integration Challenge

- Expose `towerPrefab` in Waypoint
- Towers should spawn if `isPlaceable`
- Prevent towers placing on towers
- Check the game plays properly.

Hint 0: Always connect prefabs to prefabs, not prefabs to instances.

GameDev.tv

Unity 2020.1

Particles & Algorithm Improvements

Particle Collision Module Settings

Min Kill Speed	Particles travelling below this speed after a collision will be removed from the system.
Max Kill Speed	Particles travelling above this speed after a collision will be removed from the system.

Play New Particle On Death

- When the player dies the death particle plays.

Hint 0: Expose in similar way to hit particle.
Hint 1: Instantiate the particle and keep a reference.
Hint 2: Play the particle after creating it.

GameDev.tv

Unity 2020.1

Circular Or Ring Buffers

www.how

Another mnemonic for question asking...

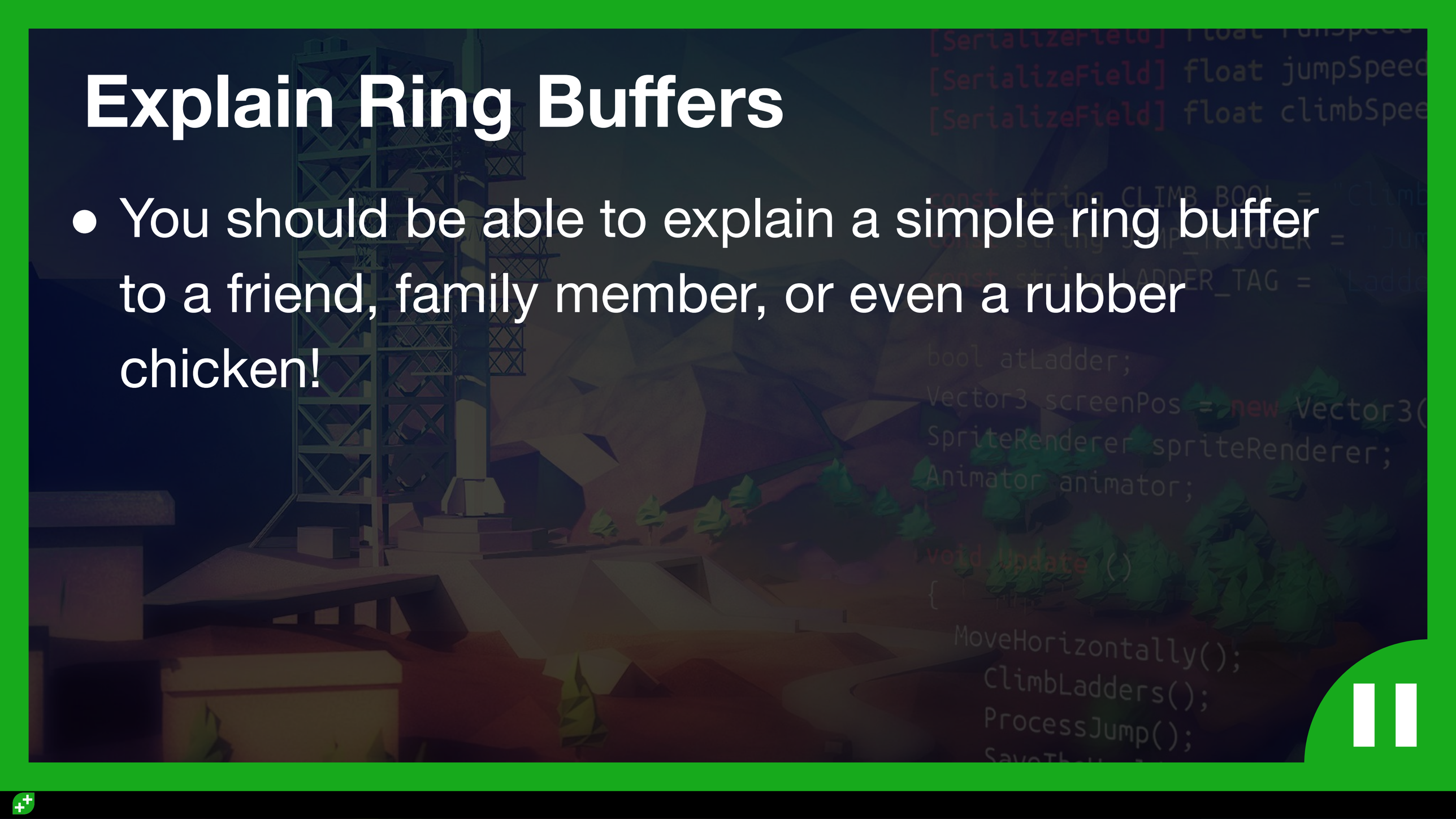
What? Are we talking about?
Why? Do we care?
Who? Already knows?
How? Do I do it?

Why Ring Buffers Rock

- Instantiating & destroying can be slow operation
- Destroying leaves a “hole” in memory
- Over time memory becomes fragmented
- Fragmented memory performs badly
- Basically it’s recycling vs disposable
- You’ll need to build your objects to last.

Explain Ring Buffers

- You should be able to explain a simple ring buffer to a friend, family member, or even a rubber chicken!



GameDev.tv

Unity 2020.1

Implementing A Ring Buffer

To-do List

- Create a Tower Factory class
- Move the tower prefab, and max towers there
- Note we'll tidy-instantiated objects later
- Wire-up our tower factory, at least in outline.

Detect Tower Limit

- Console should log when you try and place too many towers, and not instantiate a new tower.

GameDev.tv

Unity 2020.1

Revising C# Queues

A queue as the basis of a ring buffer

- Add to the top of the “stack of plates”
- When full, take one off the bottom
- Move this to the top

We’ll also need to set the waypoint bases as “placeable”, and tell the moved tower what its new base is.

Create a queue of towers

- Create an empty queue of towers for our use
- Add towers to the queue when you instantiate them
- Prove to yourself the queue is working
- Bonus: de-queue towers when you move them.

GameDev.tv

Unity 2020.1

How To Destroy Particles

Notes on destroying particles

- Typical case, a character dying
- You want to destroy character immediately
- This deletes references to the particle VFX
- … and the whole script that’s running
- Use Unity’s `Destroy(object, delay);`

Keep the hierarchy tidy

- Death effects destroy themselves
- Death effects are organized in hierarchy
- Towers are organised in hierarchy
- Enemies are organised in hierarchy.

GameDev.tv

Unity 2020.1

Complete Game Loop

When Enemy Reaches Goal

- Destroy enemy
- Play enemy death particles
- Destroy enemy death particles
- BONUS: Create different goal explosion

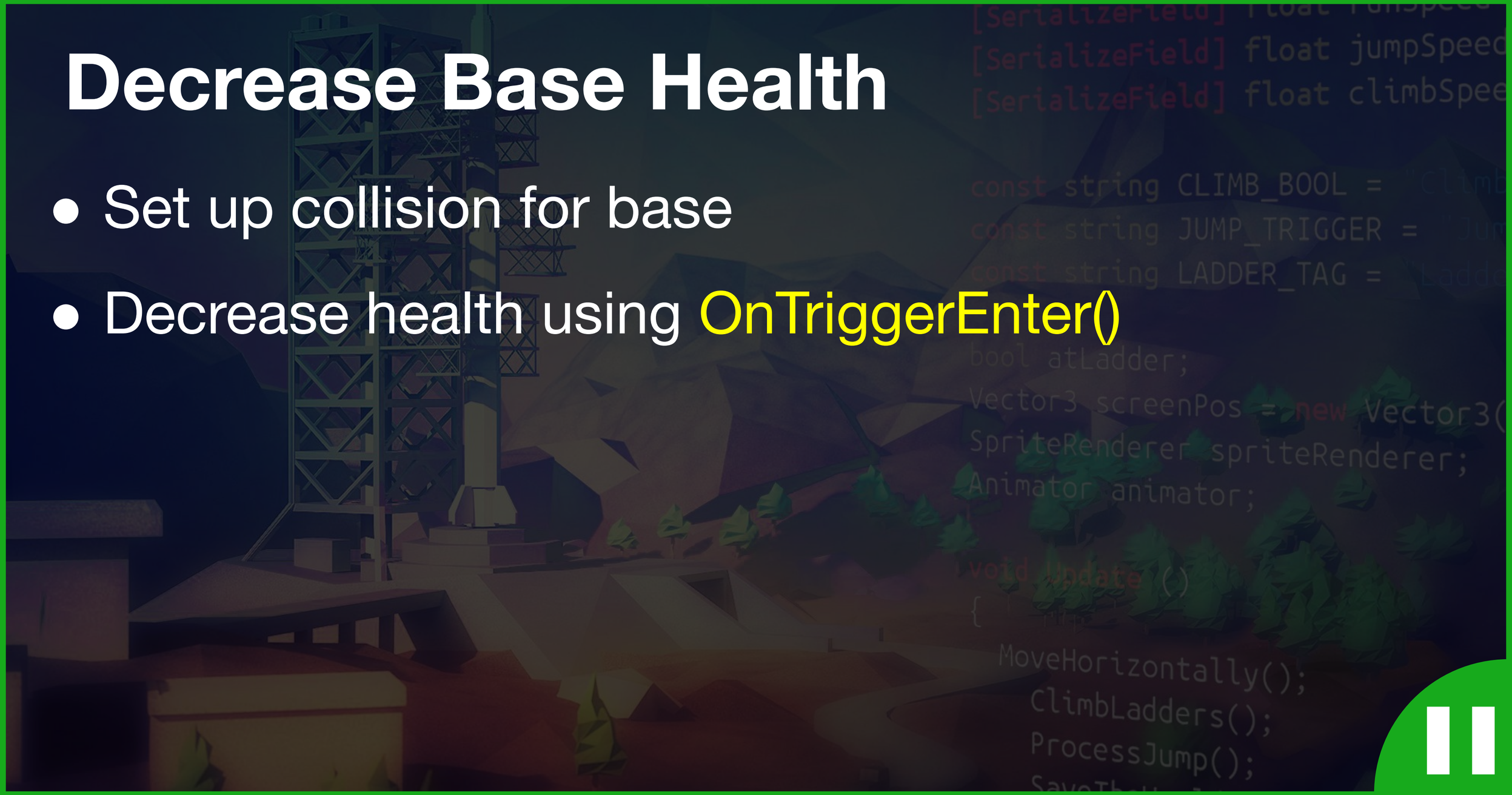
GameDev.tv

Unity 2020.1

Health For Base

Decrease Base Health

- Set up collision for base
- Decrease health using OnTriggerEnter()



GameDev.tv

Unity 2020.1

Display Health & Score

Add Score Text

- Add text UI element
- Add code to keep track of each time an enemy is spawned
- Update the text field each time an enemy is spawned

GameDev.tv

Unity 2020.1

Unity Post Processing Stack

Post Processing Stack

- Download / import Post Processing Stack
- Add PostProcessingBehavior.cs to camera
- Create profile asset
- Assign profile asset to camera
- Modify

Add Post Processing Stack

- Set up your Post Processing Stack
- Make your game look different to the default look.
- Please share a screenshot!

GameDev.tv

Unity 2020.1

Hook Up SFX

SFX Requirements For Your Game

1. What is your game's theme / experience / tone
2. What are all the moments that need SFX
3. Where can you source / create assets

Let's go through that exercise...

Implement SFX For...

- Enemy being spawned
- Enemy reaching player's base
- Enemy taking damage
- BONUS: Enemy being killed
  (note: solution to bonus will be in following video)

GameDev.tv

Unity 2020.1

PlayClipAtPoint() For SFX

Solve The Riddle

- Figure out why our death SFX is quiet compared to other SFX.
- Complete the SFX implementation so that the death SFX is same volume as other SFX.
- HINT 1: Use Debug.Break() to compare enemy death SFX to enemy hit SFX.
- HINT 2: Where in our scene are 2D sounds heard?

GameDev.tv

Unity 2020.1

Tune Your Game Moment

Creating Your Game Moment

- Aim to create an interesting 2 minute moment.
- Find the fun in your game moment by tuning what you have rather than looking to add more features.
- Intention -> Execution.

Design Levers Available

- Enemy HPs
- Enemy speed
- Enemy spawn rate
- Tower shot distance
- Tower rate of fire
- Number of towers
- Size of level
- Path through level
- Player health
- Lighting / visibility
- Camera position
- Camera angle
- Size of enemies
- Size of towers
- Art assets
- Particle assets

Tune And Share

- Tune your game to find your moment.
- Share your game with others and see what they like and want more of.
- Give people a very specific challenge (eg. last 2 mintues, or beat XYZ score).
- Share a video of your final product with our community.

GameDev.tv

Unity 2020.1

Section 5 Wrap Up

END OF SLIDES

GameDev.tv

Unity 2020.1

Assertions For Error Checking

What are assertions?

An assertion in Unity is a condition check that can log an error to the console, continuing execution.

This can be left out in the final game builds, but included in development builds.

Why use assertions

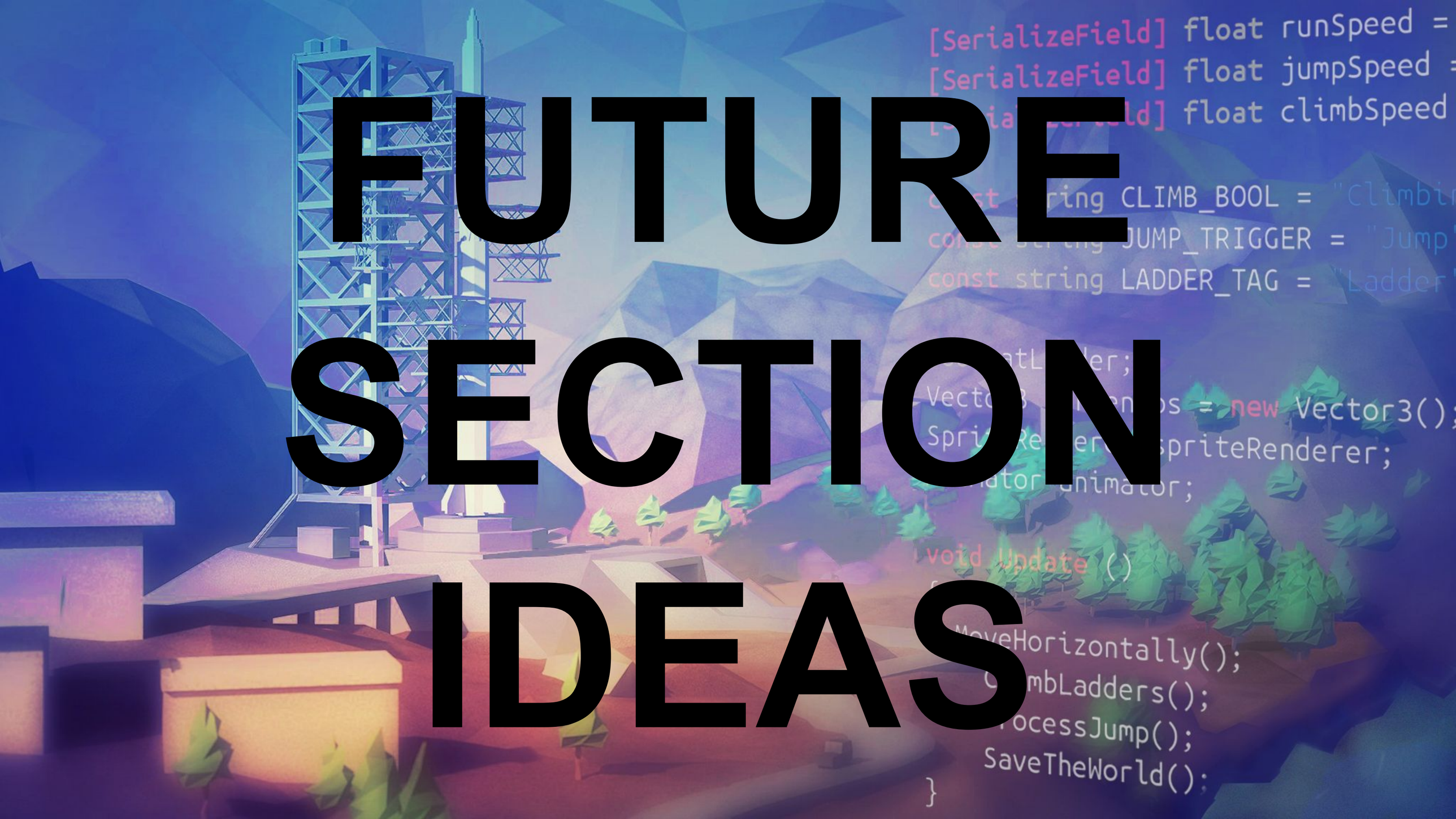
Assertions can be handy to check that your game is setup properly.

In our case we will use them to easily identify overlapping blocks, which could confuse the pathfinder.

Write your first assert statement

- Overlapping blocks should create a console error
- This should happen when you first play the game
- You should easily be able to identify the block

Why not share a screenshot to celebrate your first assert, and let us know your thoughts?


FUTURE
SECTION
IDEAS

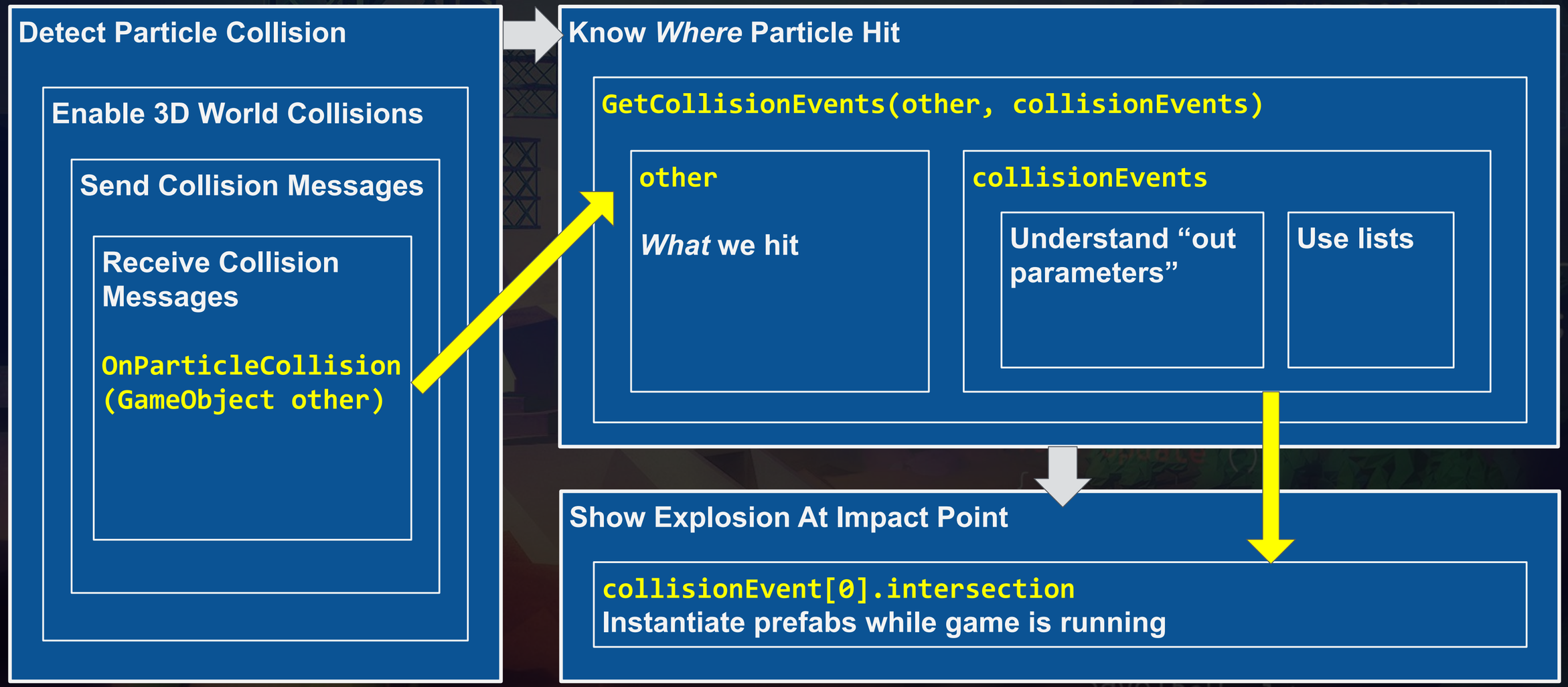
How To Encapsulate Well

- Objects work on their own member variables
- Ideally no public member variables
- A few public functions to communicate
- A sprinkling of private “helper” functions
- This minimises the number of reasons to change
- In games keep your objects as “real” as poss.

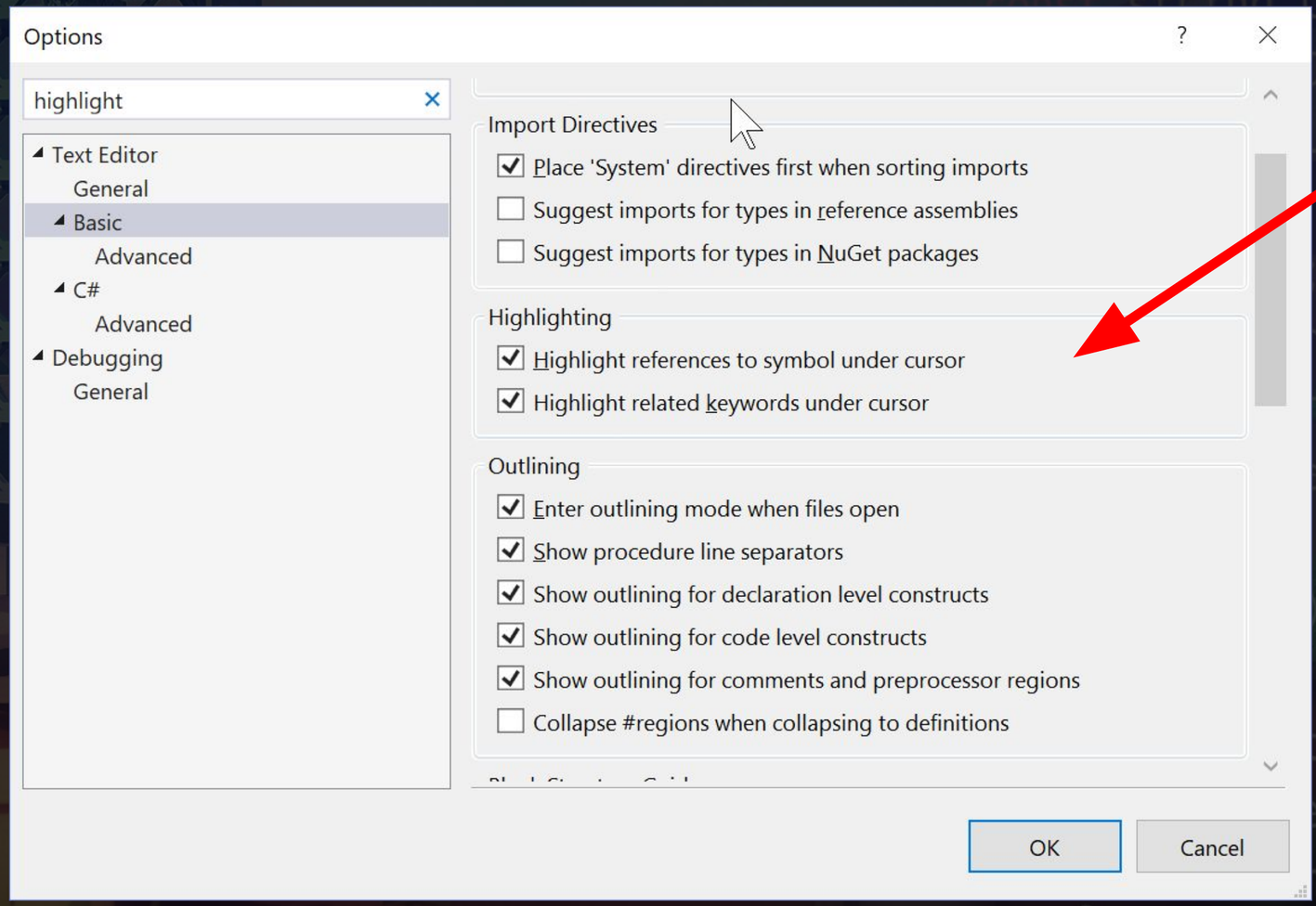
“Code can be copied once, but that when the same code is used three times, it should be extracted into a new procedure”

https://en.wikipedia.org/wiki/Rule_of_three_(computer_programming)

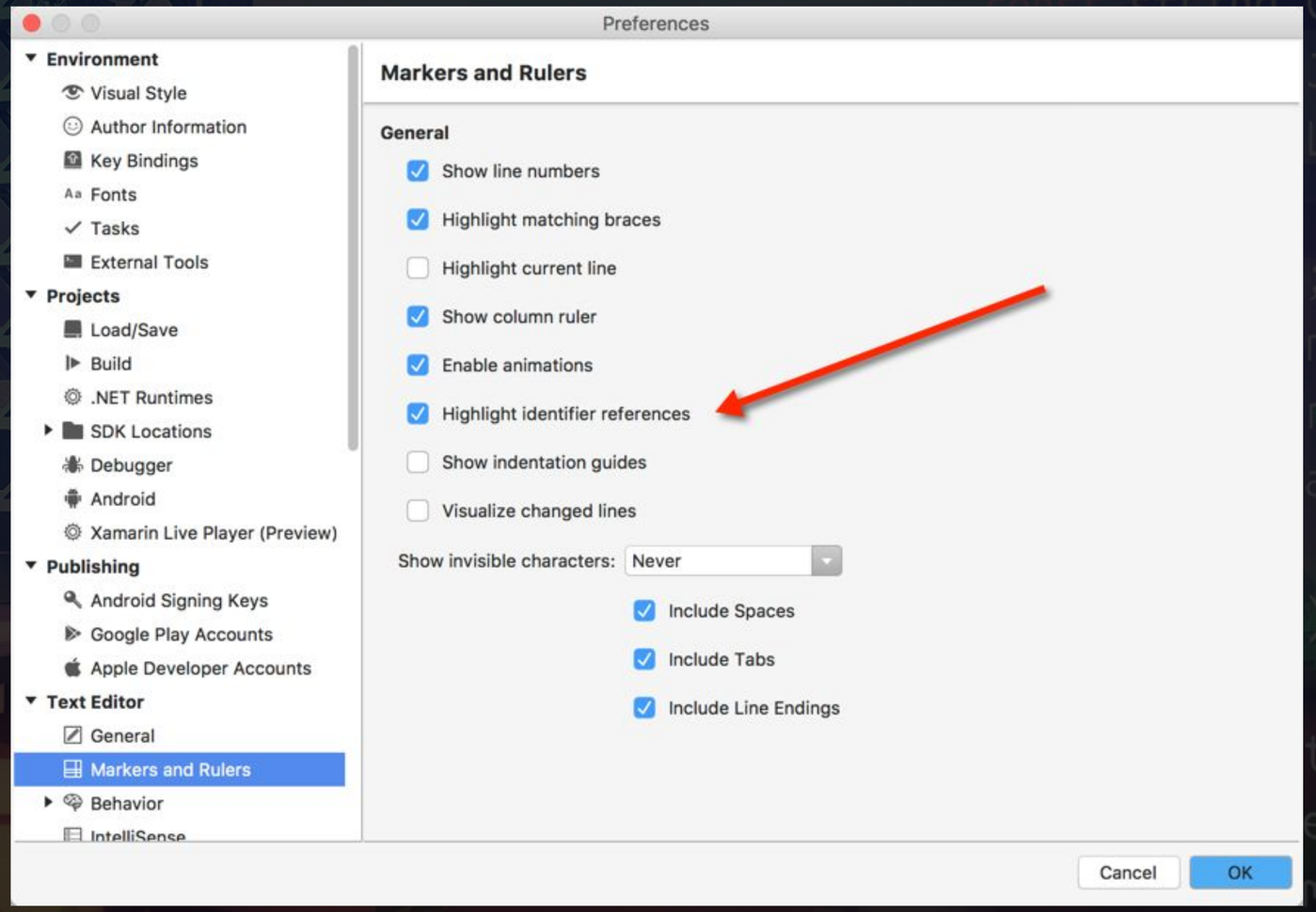
Make Particles Explode On Impact



Reference Highlighting - PC



Reference Highlighting - Mac



The % Operator For Remainder

buildIndex

buildIndex	/ 7	remainder
0	0 + 0/7	0
1	0 + 1/7	1
2	0 + 2/7	2
3	0 + 3/7	3
4	0 + 4/7	4
5	0 + 5/7	5
6	0 + 6/7	6
7	1 + 0/7	0
8	1 + 1/7	1
9	1 + 2/7	2