GameDev.tv

Unity 2017.3

Welcome To Section 5

Realm Rush

This is a 3D isometric tech demo designed to teach the fundamentals of AI pathfinding, namely Breadth First Search (BFS).

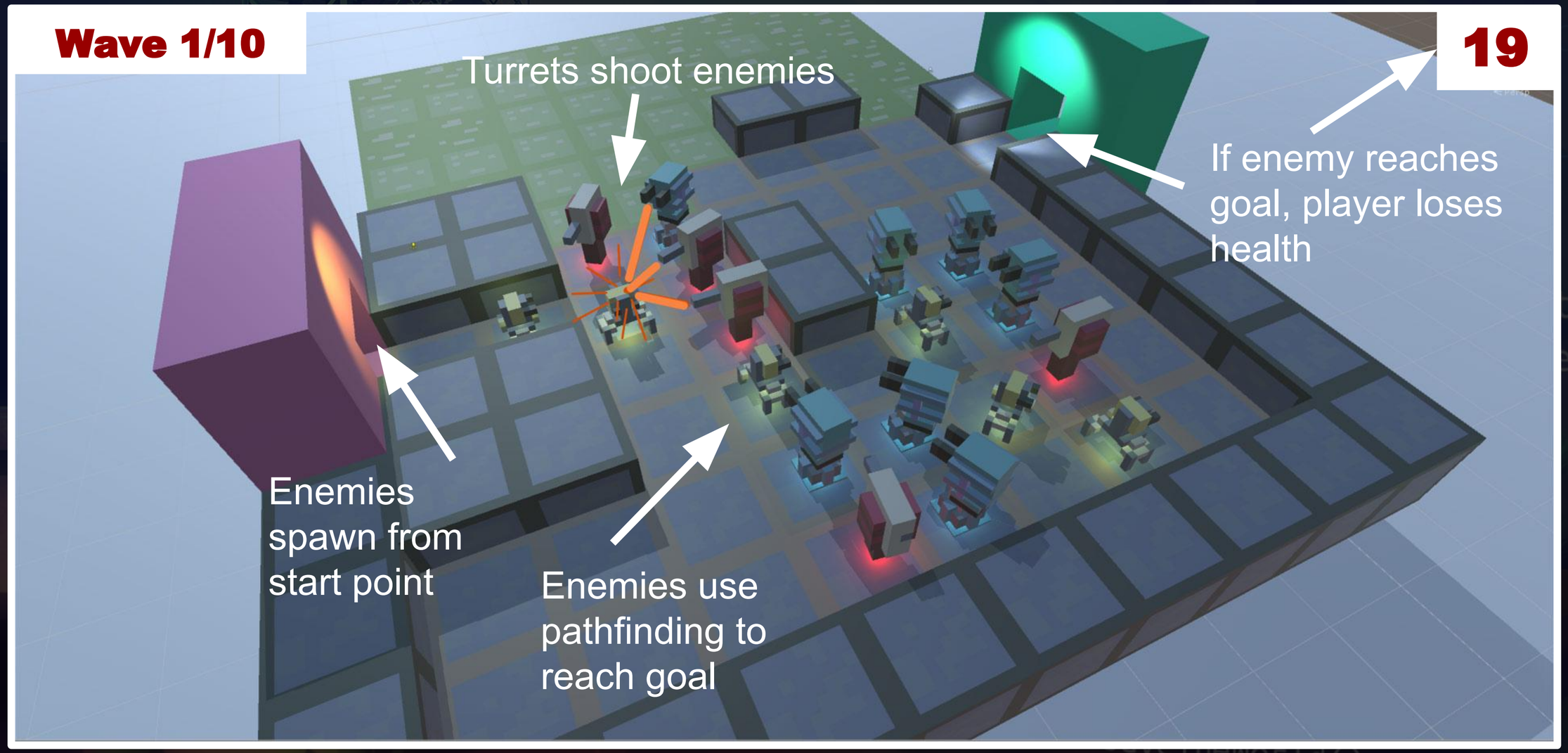
This is the fifth section of the Complete Unity Developer 2.0 course.

GameDev.tv

Unity 2017.3

Realm Rush Game Design

Core Gameplay Overview



Realm Rush Game Design

Player Experience:

Strategic Planning

Core Mechanic:

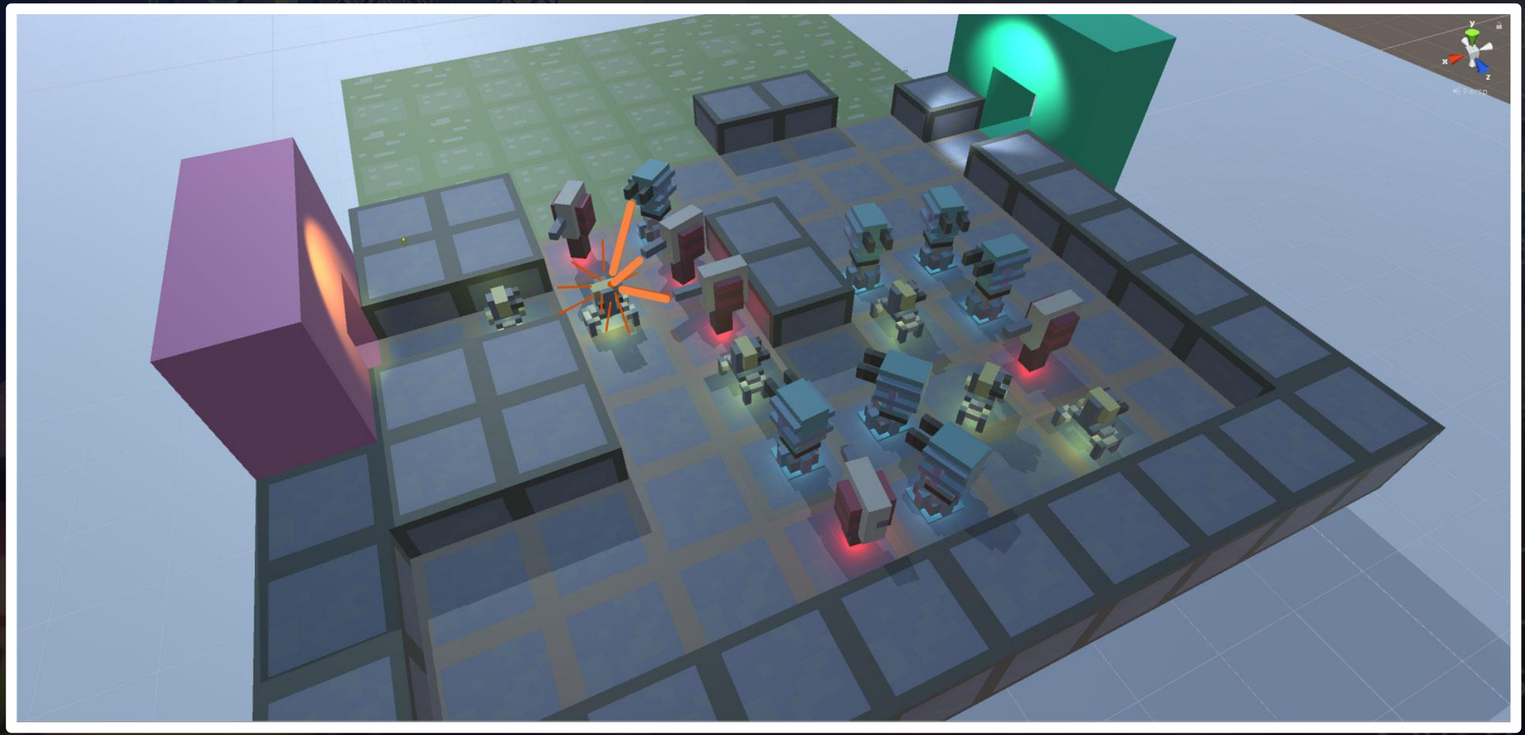
Place towers which shoot enemies.

Core game loop:

Place towers to kill enemies in order to stop enemies reaching player’s base and causing damage.

Game Theme

- Sci Fi robots enemies and robotic towers doing robotic things in robotic ways.

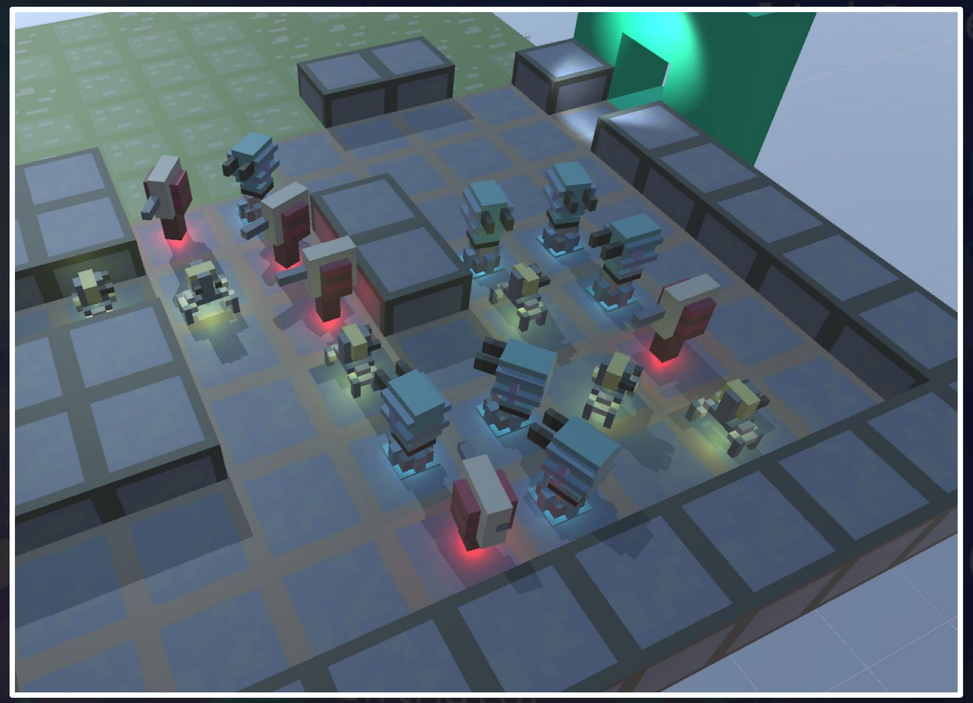


Tech Setup

- Created using 3D assets.
- 1920 x 1080 resolution.
- Variation in maps to take advantage of pathfinding.

MVP Features

- **Enemy pathing:** Enemies spawn and follow best path to reach player base.
- **Place towers:** Player can place towers that shoot at enemies within range.
- **Enemy health:** Multiple hits to be killed.



Wish List:

- **Tower scarcity:** Maximum number of towers can be built.
- **Player Health:** When enemy reaches goal, player loses some health.

An Ongoing Challenge

- As you work through the pathfinding coding, think about how you can turn this functionality into a new type of gameplay.

GameDev.tv

Unity 2017.3

Limitations Of Unity Pathfinding

Our Pathfinding vs Unity's

	Unity's NavMesh System	Our BFS System
Underlying algorithm	A* (assumed)	Breadth First Search
Movement type	Continuous analog mesh	Discrete digital grid
Runtime recalculation	With limitations	Up to us, very flexible
Pathfinding speed	Very fast	Slow, but can improve as needed
Navigation space(s)	Two: visual, and navigation	Unified
Tweakable parameters	Dozens	3 (start, end, speed)
What you'll learn	How to use Unity's system	Fundamentals of AI pathfinding - Powerful new C# features - How to take control of Unity

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 2017.3

Z-Fighting And Quads

Some notes on cubes...

- A cube is made of 6 quads (12 triangles)
- If you want a label exactly on a surface the label quad will “fight” for the position with the face
- Solution is to compose the cube of quads
- Bonus is that you can color each side :-)

Setup & Prefab Your Cube

- The cube should consist of 6 quads
- It should be visible from all sides
- Center of top face should be at origin
- Bonus: try different colours on each face
- Share you cool new cube with us.

GameDev.tv

Unity 2017.3

[ExecuteInEditMode] Attribute

Edit Mode Scripts

- Run in the Unity editor, when the game is NOT playing
- Very useful for creating world-building tools
- `Update()` gets moved whenever you manipulate the scene.

Get Z Snapping Working

- Dragging should snap to x and z now
- Bonus 0: Add a `gridSize` parameter
- Bonus 1: Add a `Range[]` attribute to grid size…

`[Range(1f, 20f)]` for example restricts the exposed field to the specified range and puts a slider in the inspector.

GameDev.tv

Unity 2017.3

Using Text Mesh For Labels

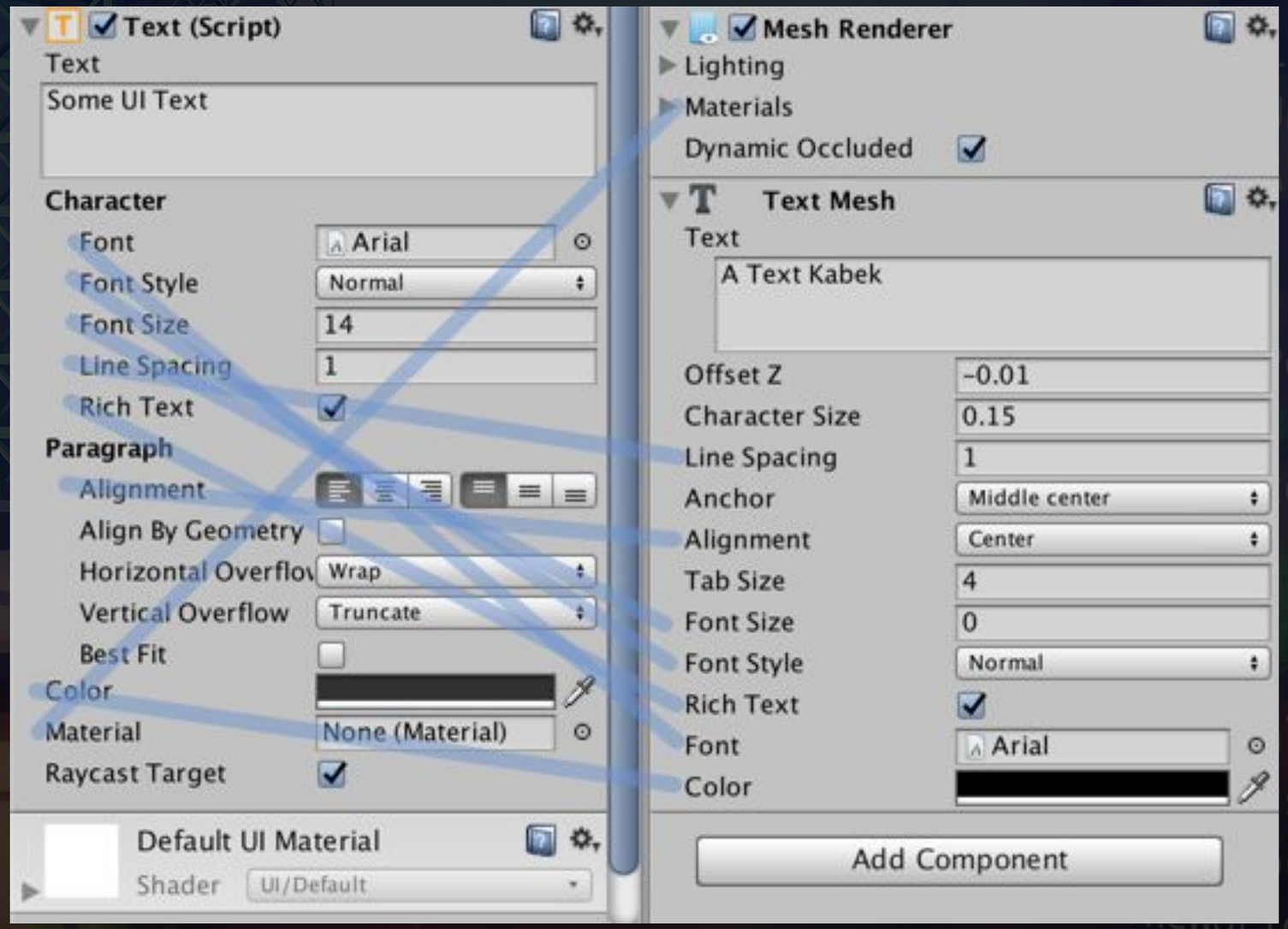
Setup Your Text

- 99, 99 should fit on the top
- Choose any font you like
- The font should be sharp
- What did you tweak?
- Why did it help?

Text Mesh vs UI Text

	Text Mesh	UI Text
Typical use	Text labels	User Interface!
Simplicity	Simple	Complex
Interactable?	No	Yes
World / Screen in 3D	World	Choice

Text Mesh vs UI Text In Unity



GameDev.tv

Unity 2017.3

C# Lists For Custom Path

Lists vs Arrays in C#

	Array	List
Collection Size	Fixed*	Variable
Performance	Fast	Varies
Ease of use	Easy	Easy
Flexibility	Low	High
How to use	`Block[]`	`List<Block>`

"If in doubt, whip a list out"

Print Out All The Waypoints

- The console should print the block names
- They should be in the order specified

Note how fast this happens, use Pause and Step.

GameDev.tv

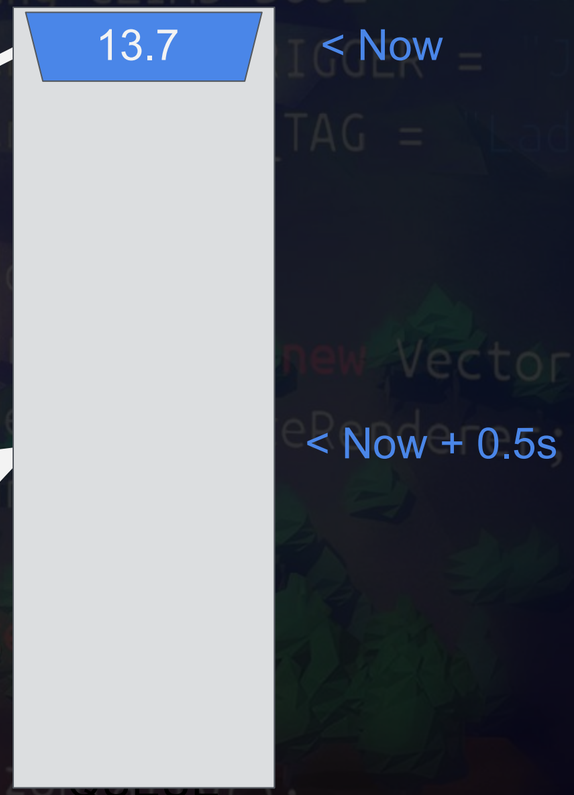
Unity 2017.3

Introducing Coroutines

Code To Wait At Each Waypoint

```
void Start()
{
   StartCoroutine(FollowPath());
}


IEnumerator FollowPath()
{
    foreach(Waypoint waypoint in path)
    {
        transform.position = waypoint.transform.position;
        yield return new WaitForSeconds(dwellTime);
    }
}
```



Get Your Enemy Patrolling

- Get the log printing the following…
- “Starting patrol…”
- “Visiting block: 1,0” etc
- “Ending patrol”

GameDev.tv

Unity 2017.3

Types Of Pathfinding

Search Algorithms Overview

Search Algorithm	Always Shortest Path?	Movement Cost Can Vary	Multiple End Points	Speed On 100x100 Grid	Notes
Breadth First Search (BFS)	✔	✘	✔	Medium	Simple, versatile, fast enough most of the time.
Dijkstra's algorithm	✔	✔	✔	Slow	Movement cost allows for swamps, roads etc.
A Star	❓ **Choice**	✔	✘	Fast	Complex. Can trade speed against accuracy.

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.*

Share why A* isn't always the best

- You've shared a forum post, Tweet or blog on why A* isn't necessarily the first choice for pathfinding
- Keep it really simple, just one thing that struck you
- Enjoy the process of learning by teaching!

GameDev.tv

Unity 2017.3

The Path Ahead

What to expect in the next 2 lectures

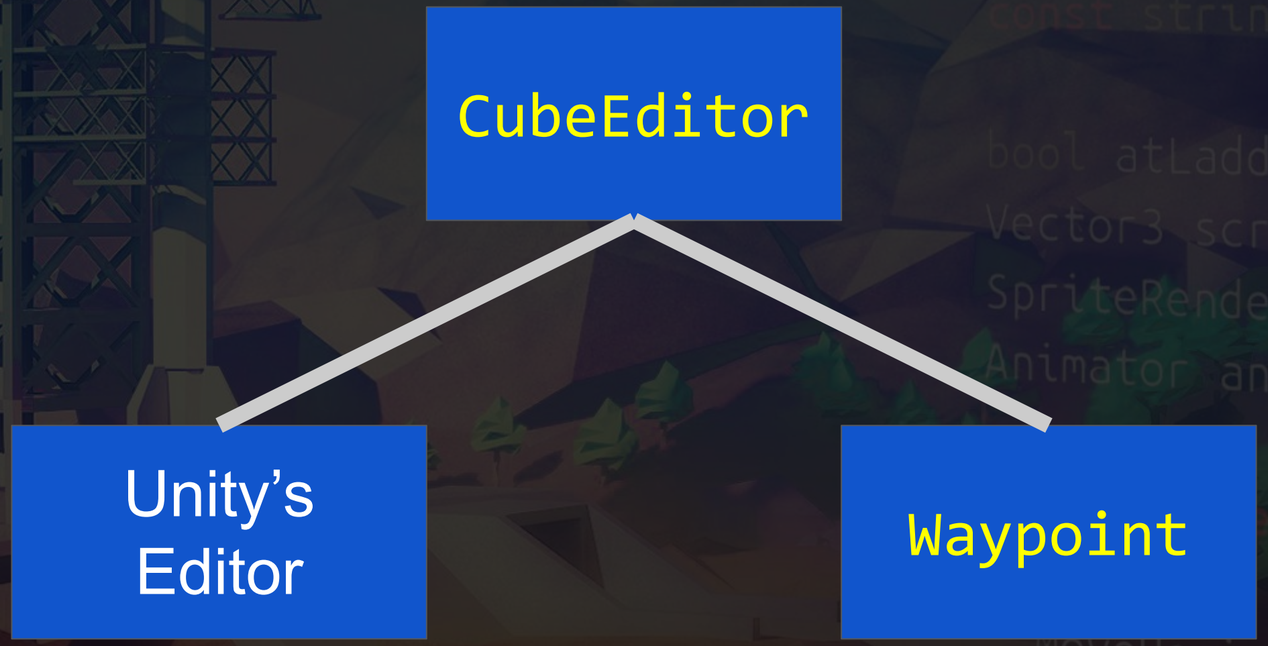
- Lots of refactoring
- Thinking about instance variables vs constants
- Try and zoom out and see the purpose
- Engage on the community.

GameDev.tv

Unity 2017.3

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 2017.3

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 2017.3

Finding Game Objects By Name

Color Start & End

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

GameDev.tv

Unity 2017.3

Explore Thy Neighbour

```
[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();
}
```

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.

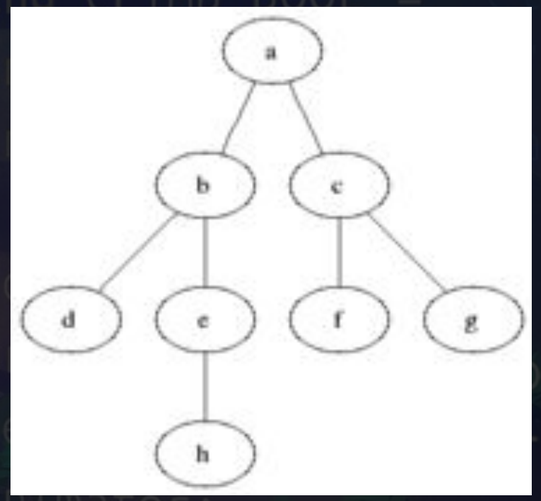
GameDev.tv

Unity 2017.3

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?

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

- 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 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

Tower LookAt Enemy

Tower LookAt Enemy

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

GameDev.tv

Unity 2017.3

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 2017.3

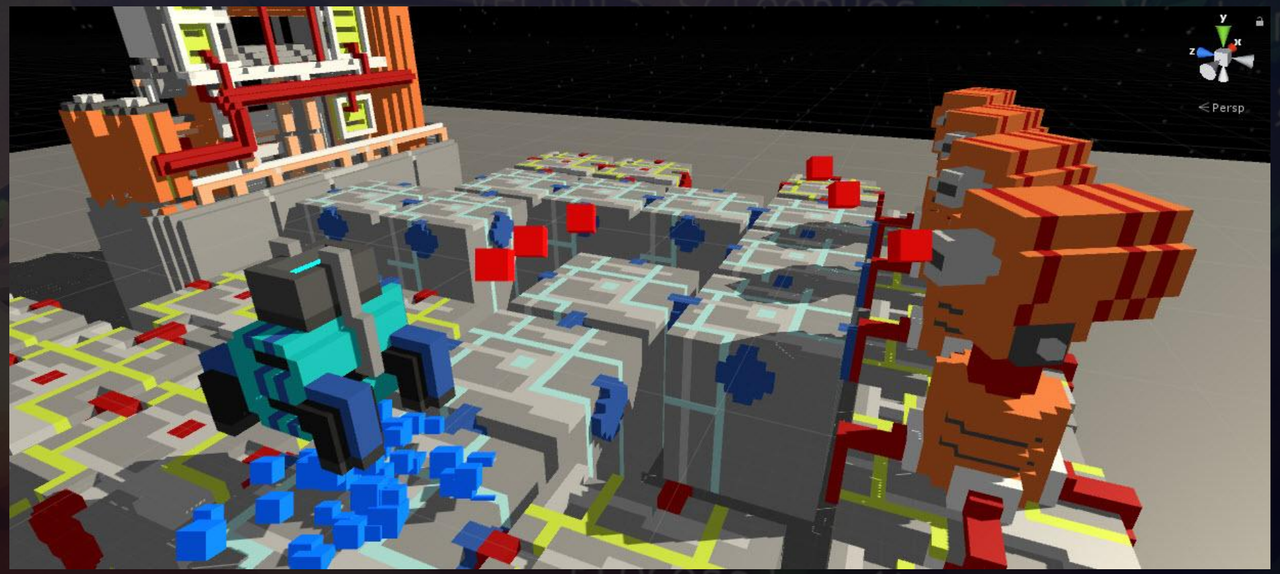
Another Solo Challenge

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

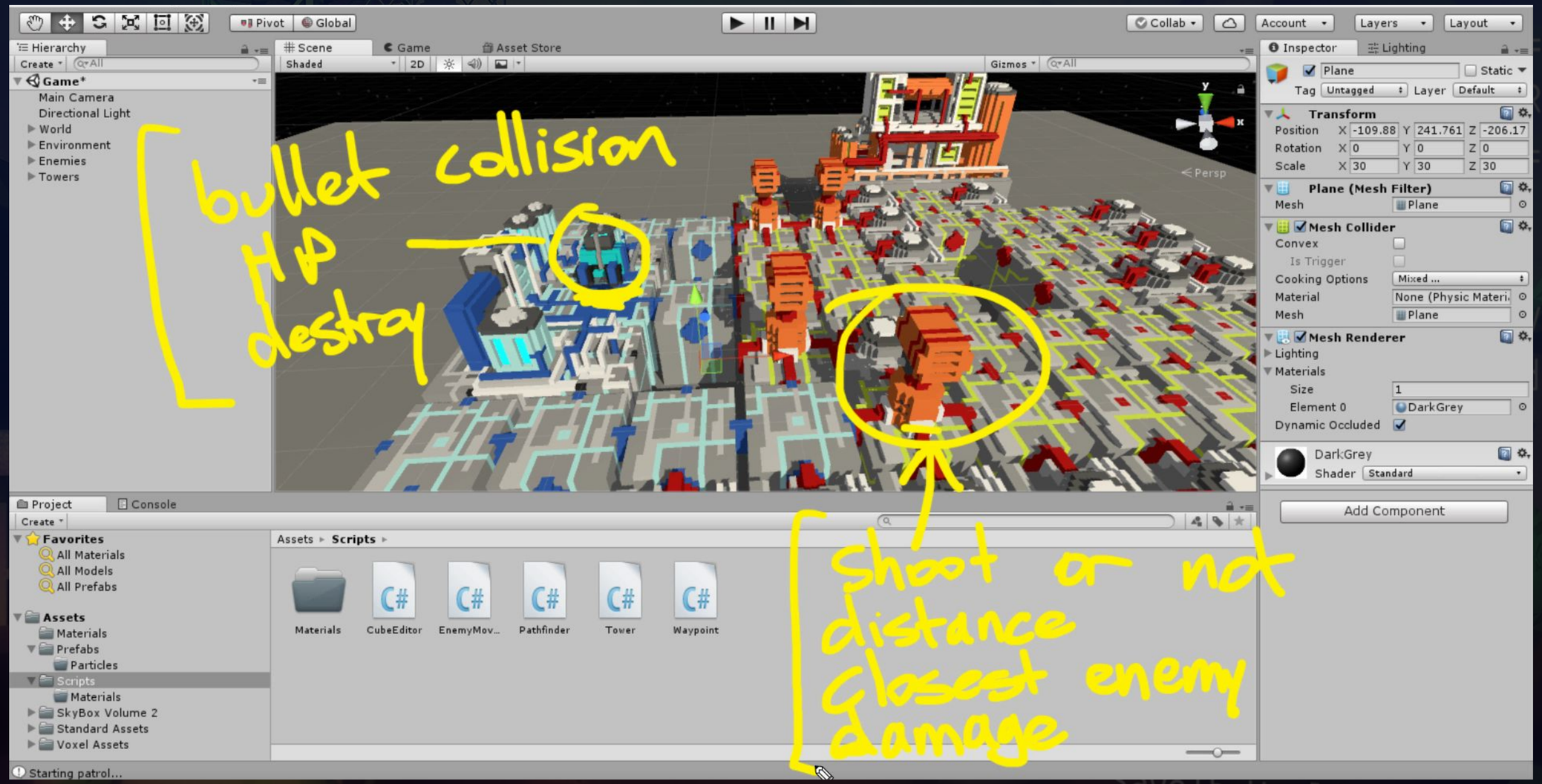
- Using Argon Assault as reference, implement:
  - Towers shoot projectiles
  - Projectiles collide with enemy
  - Damage effect is triggered
  - Enemies lose hit points
  - Enemies die

GameDev.tv

Unity 2017.3

Make Tower Shoot

Content Of Scripts



GameDev.tv

Unity 2017.3

Enemy HitPoints

```
[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();
}
```

Kill Enemy

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

GameDev.tv

Unity 2017.3

Check For Distance

```
[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();
}
```

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

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 2017.3

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 2017.3

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 2017.3

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 2017.3

Detecting Mouse In 3D

OnMouseOver()

- Called every frame when mouse “over” collider
- Handy general purpose as it works in 2D too
- Is simpler than “raycasting”.

Report Mouse Over Blocks

- Lookup OnMouseOver() in Unity's docs
- See if you can log which block it's over.

GameDev.tv

Unity 2017.3

Conditional Instantiation

```
[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();
}
```

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

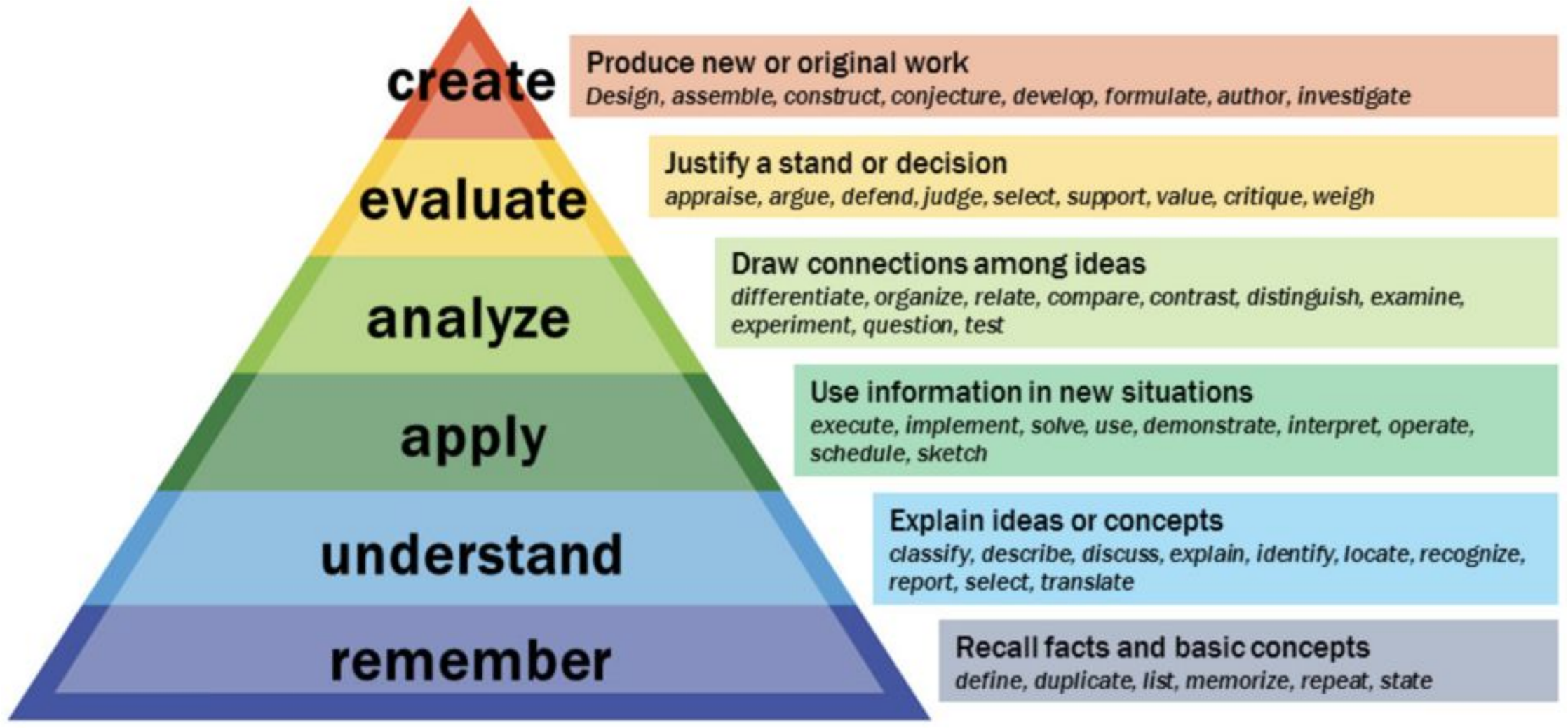
- Simply also “filter” for the block being placeable.

GameDev.tv

Unity 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

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 2017.3

Health For Base

Decrease Base Health

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

GameDev.tv

Unity 2017.3

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 2017.3

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 2017.3

Hook Up SFX

```
[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();
}
```

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 2017.3

PlayClipAtPoint() For SFX

```
[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();
}
```

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 2017.3

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 2017.3

Section 5 Wrap Up

[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 2017.3

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.

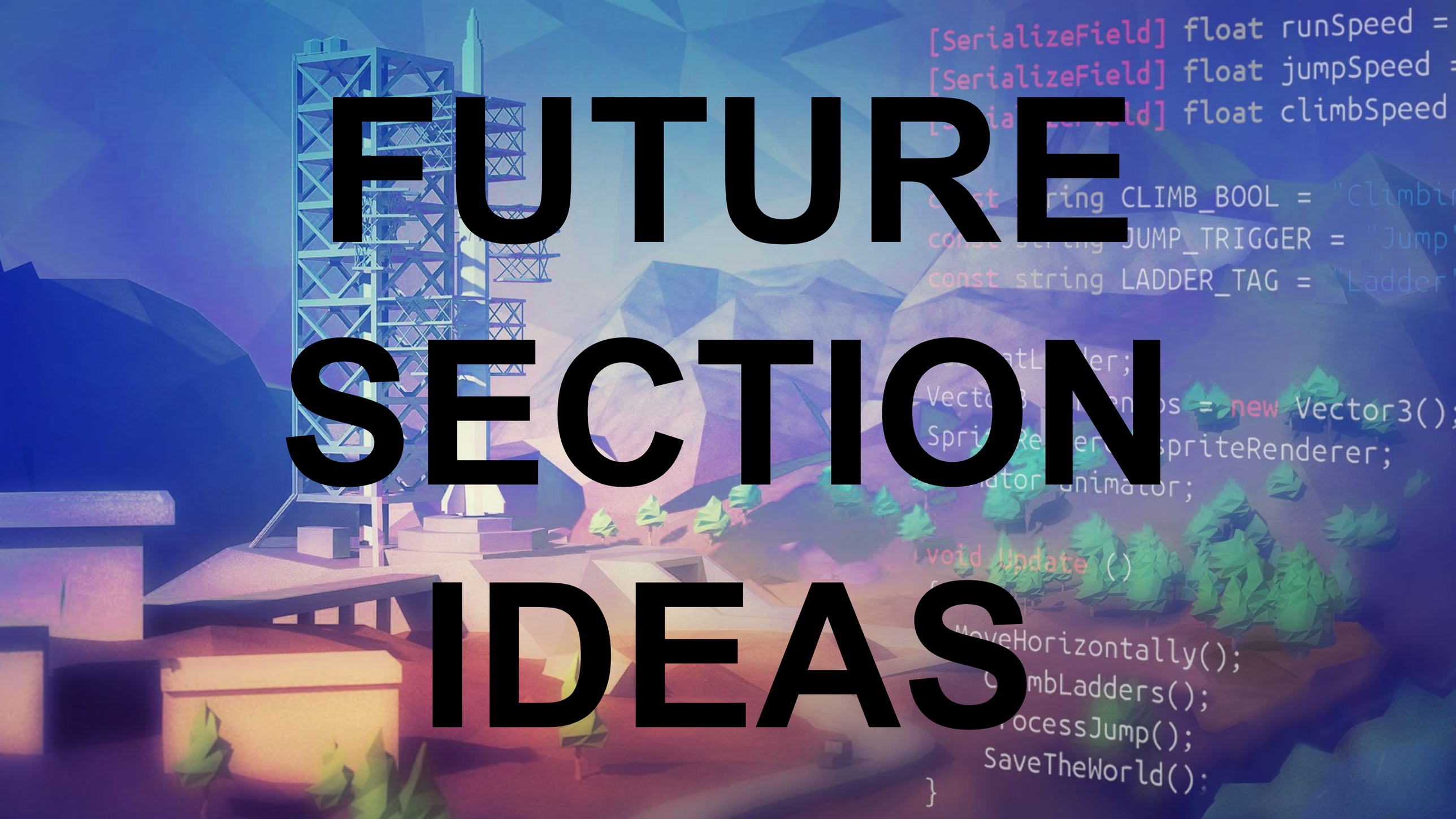
In our case we will use them to easily identify overlapping blocks, which could confuse the pathfinder.

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

Detect Particle Collision

- **Enable 3D World Collisions**
  - **Send Collision Messages**
    - **Receive Collision Messages**

      `OnParticleCollision (GameObject other)`

Know *Where* Particle Hit

`GetCollisionEvents(other, collisionEvents)`

- `other`

  What we hit

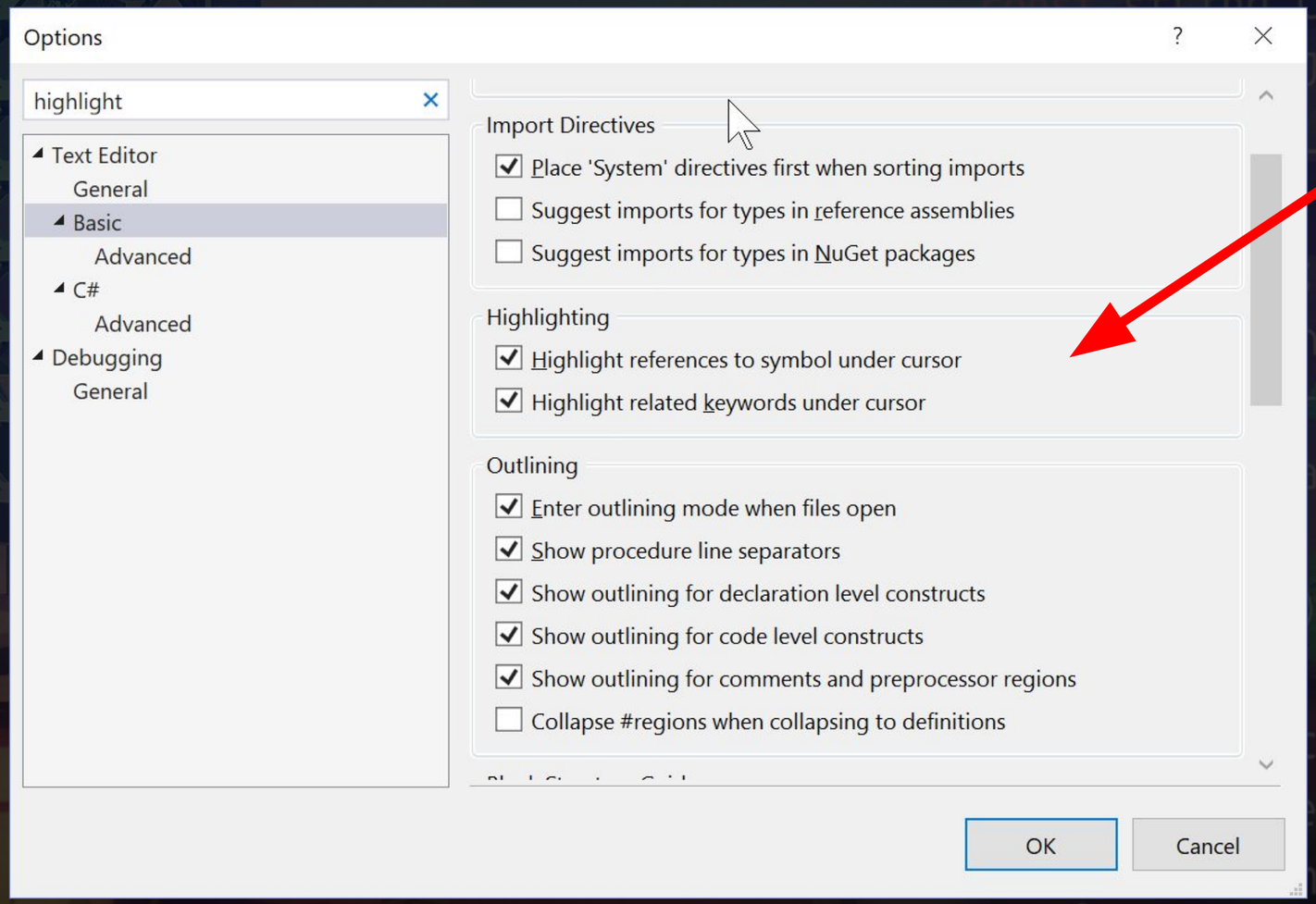
- `collisionEvents`
  - Understand "out parameters"
  - Use lists

Show Explosion At Impact Point

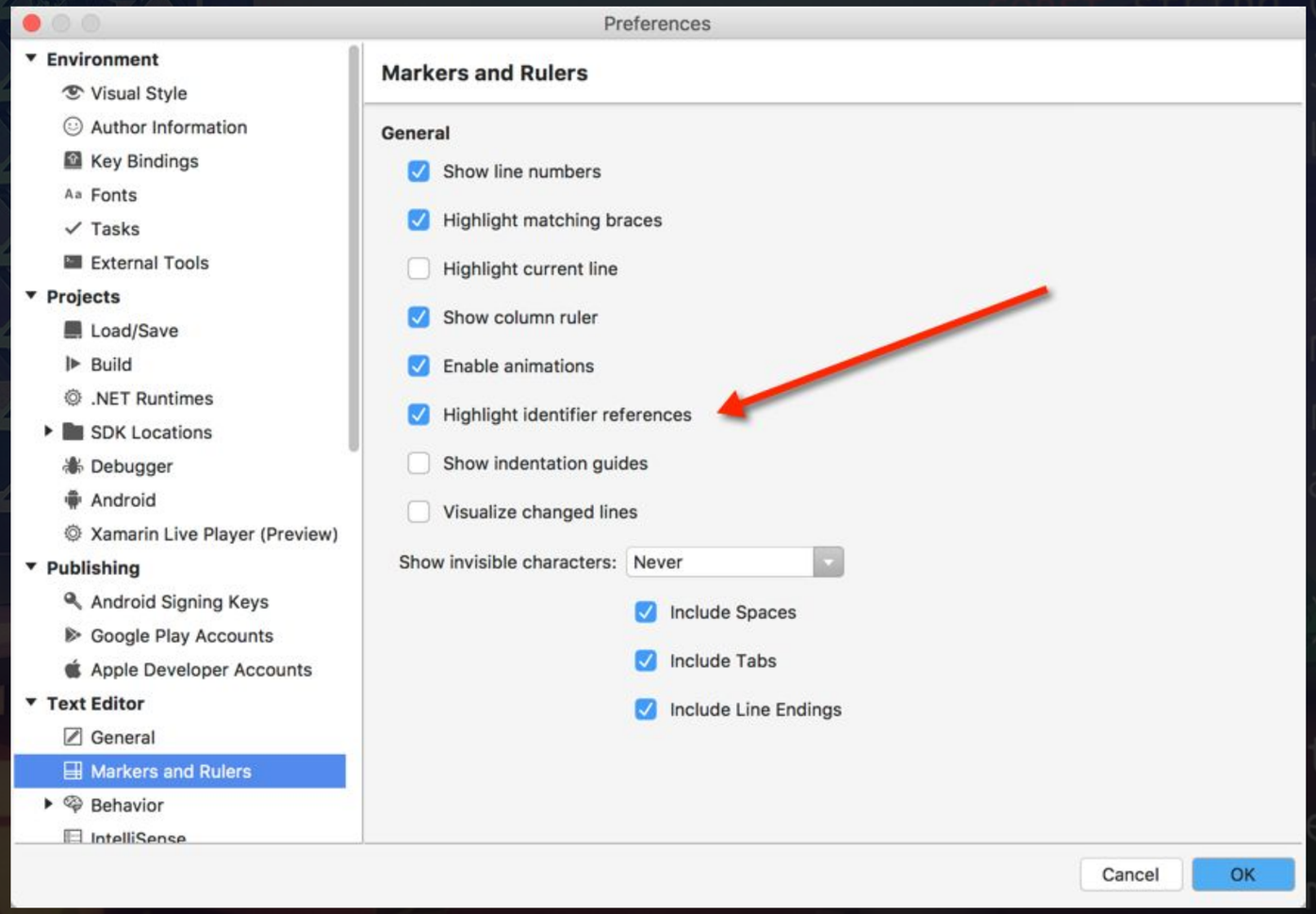
`collisionEvent[0].intersection`

Instantiate prefabs while game is running

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