Unity Games Development Workshop

Starting Development

https://github.com/KyleHammer/ProjectBeachDay

Clone/Download the Project

https://github.com/KyleHammer/ProjectBeachDay

What are we doing today?

Designing level layouts

Balancing the game

Programming

Bug Fixing

And most importantly, learning new skills!

Brief:

A roguelike that is about to be released for early access, however the game is still contains some obvious bugs and does not feel good to play.

It’s up to you to polish up the game and add your own spin to it!

Controls

WASD – Movement
Space – Dash
Left Mouse – Shoot (Very Bugged)

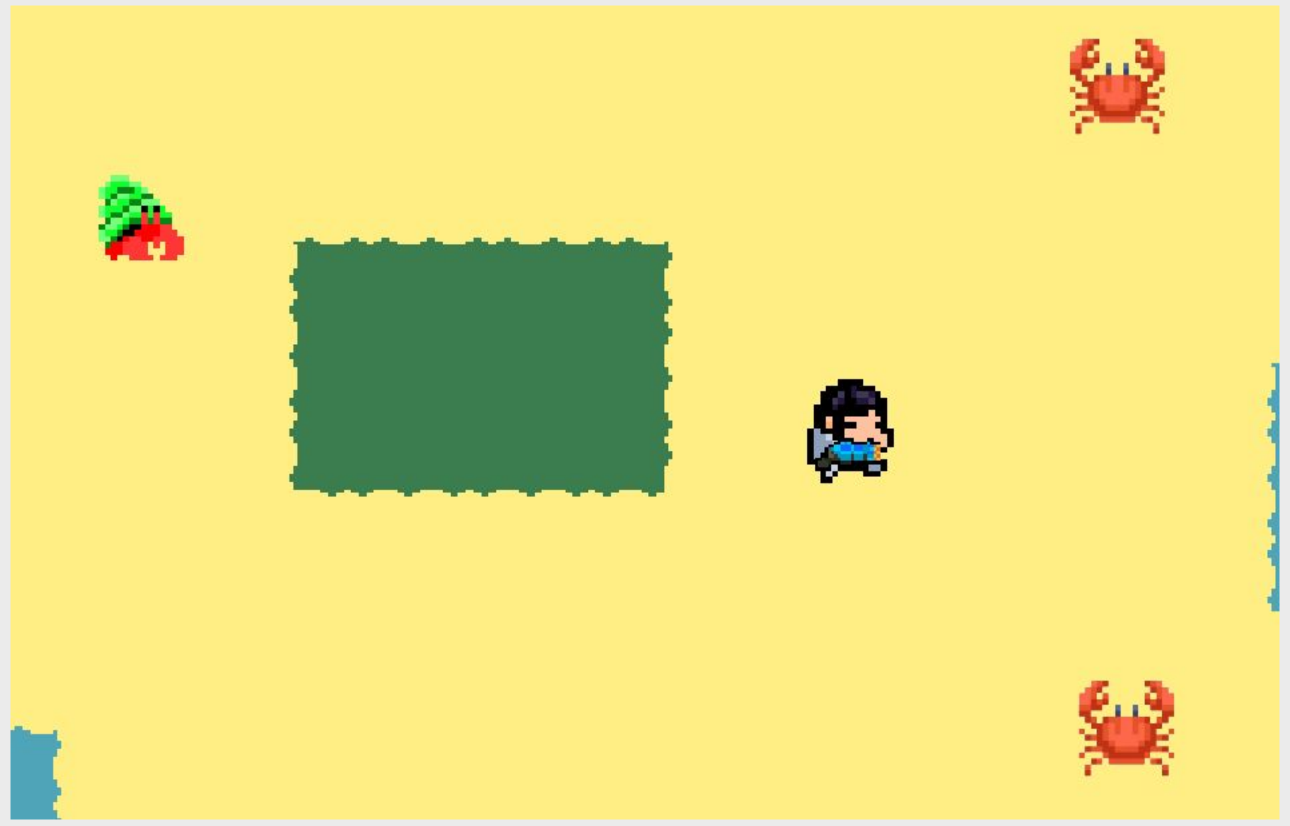
Hold R – Restart
Esc – Quit Game

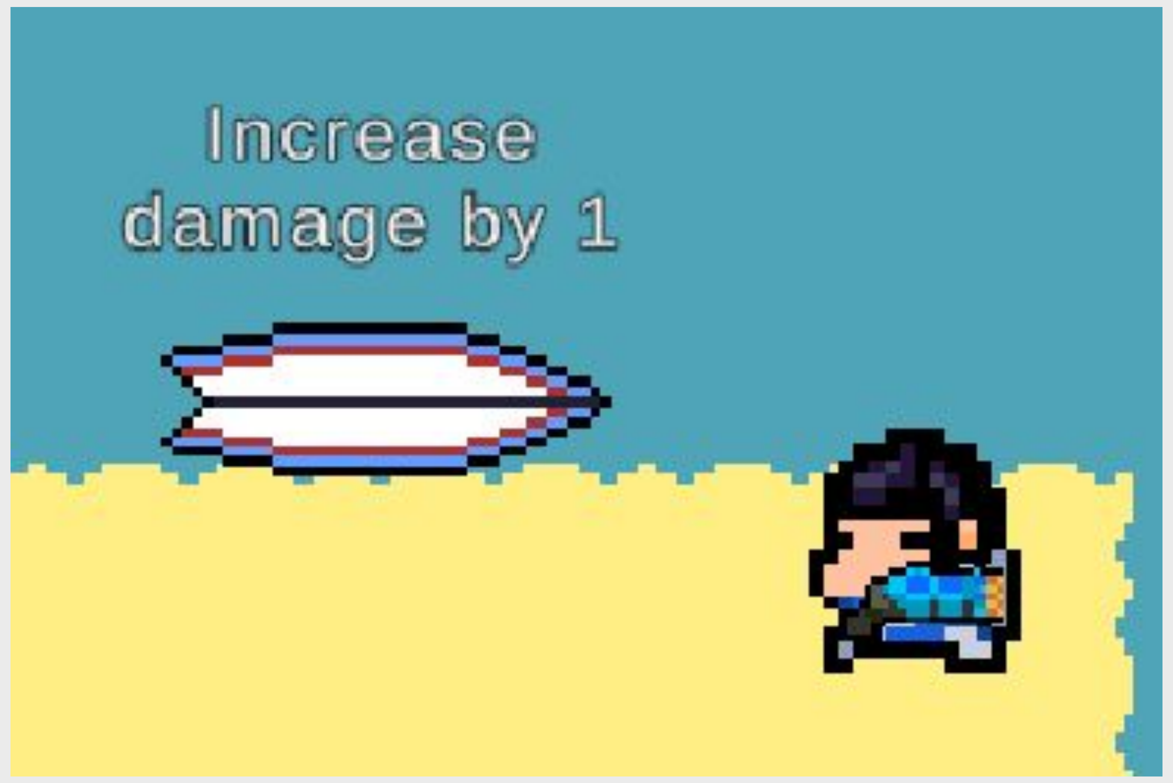
Game Progression

Player clears room of enemies

Player gets stat upgrade

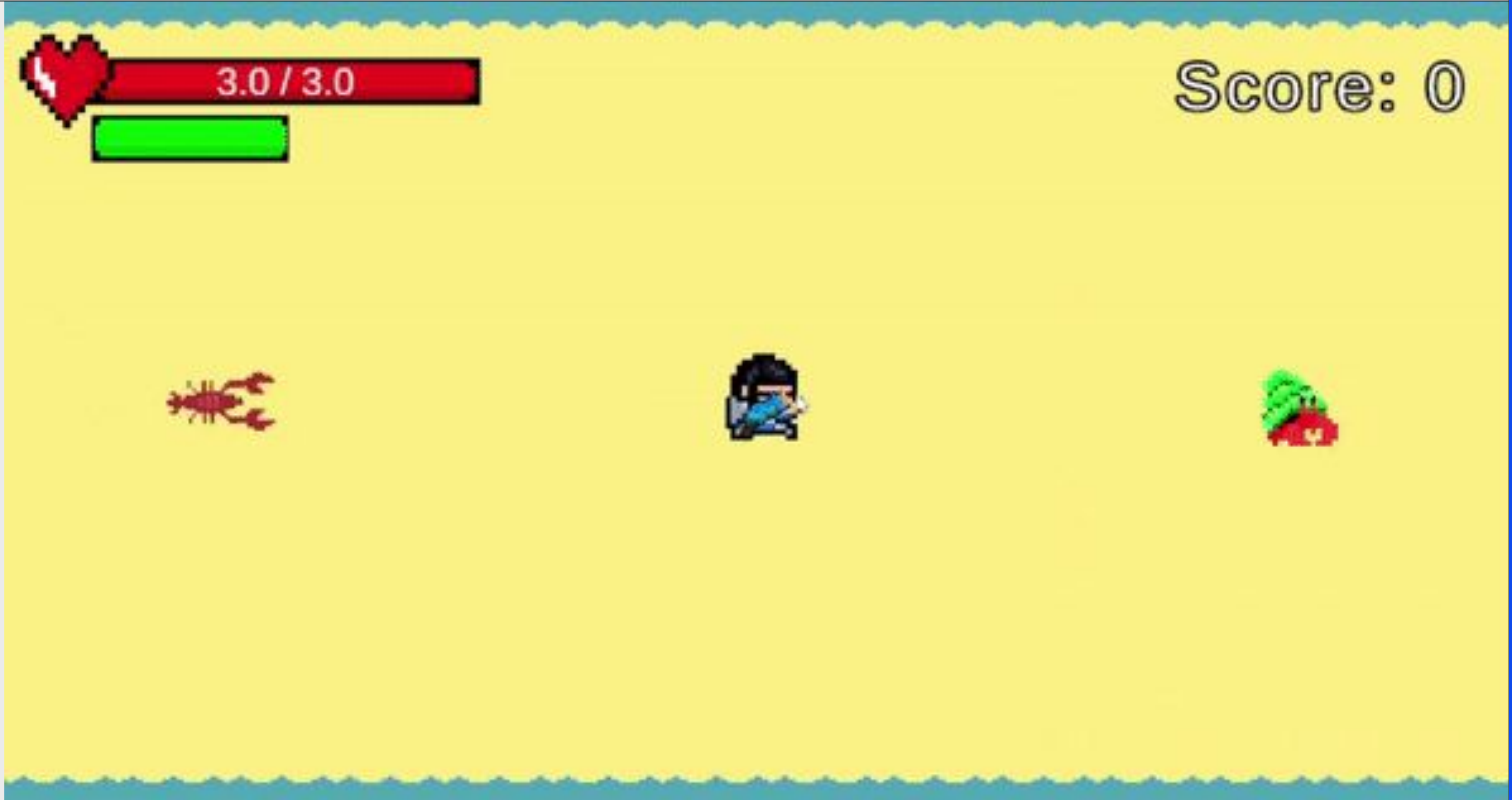
Player selects next room reward





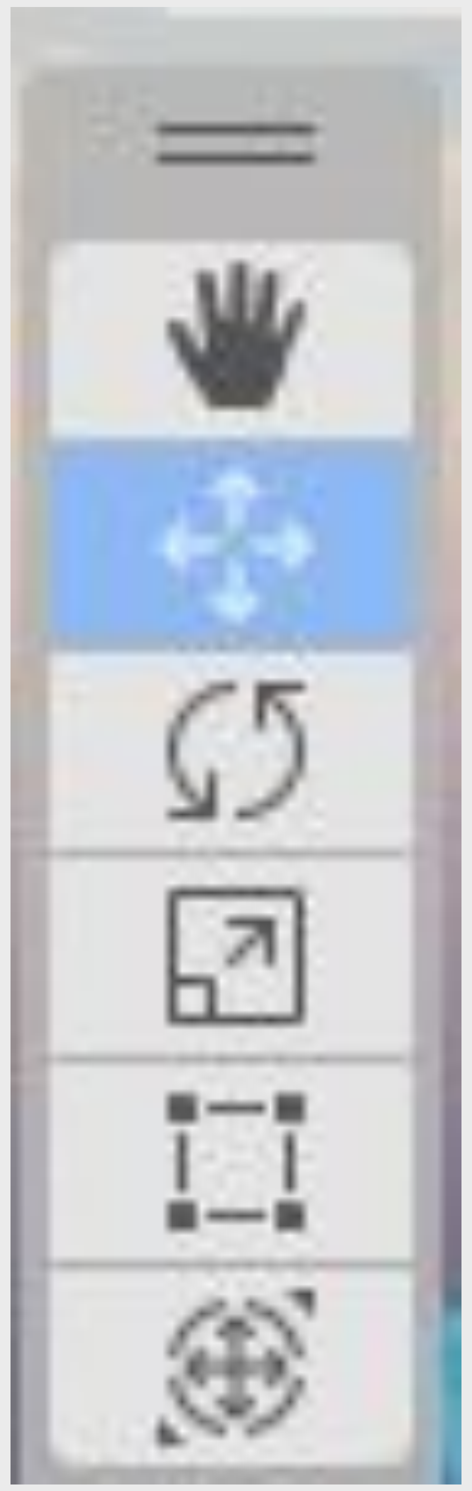
…Or at least that is how it should work

Example End Product



Navigating the Game Engine

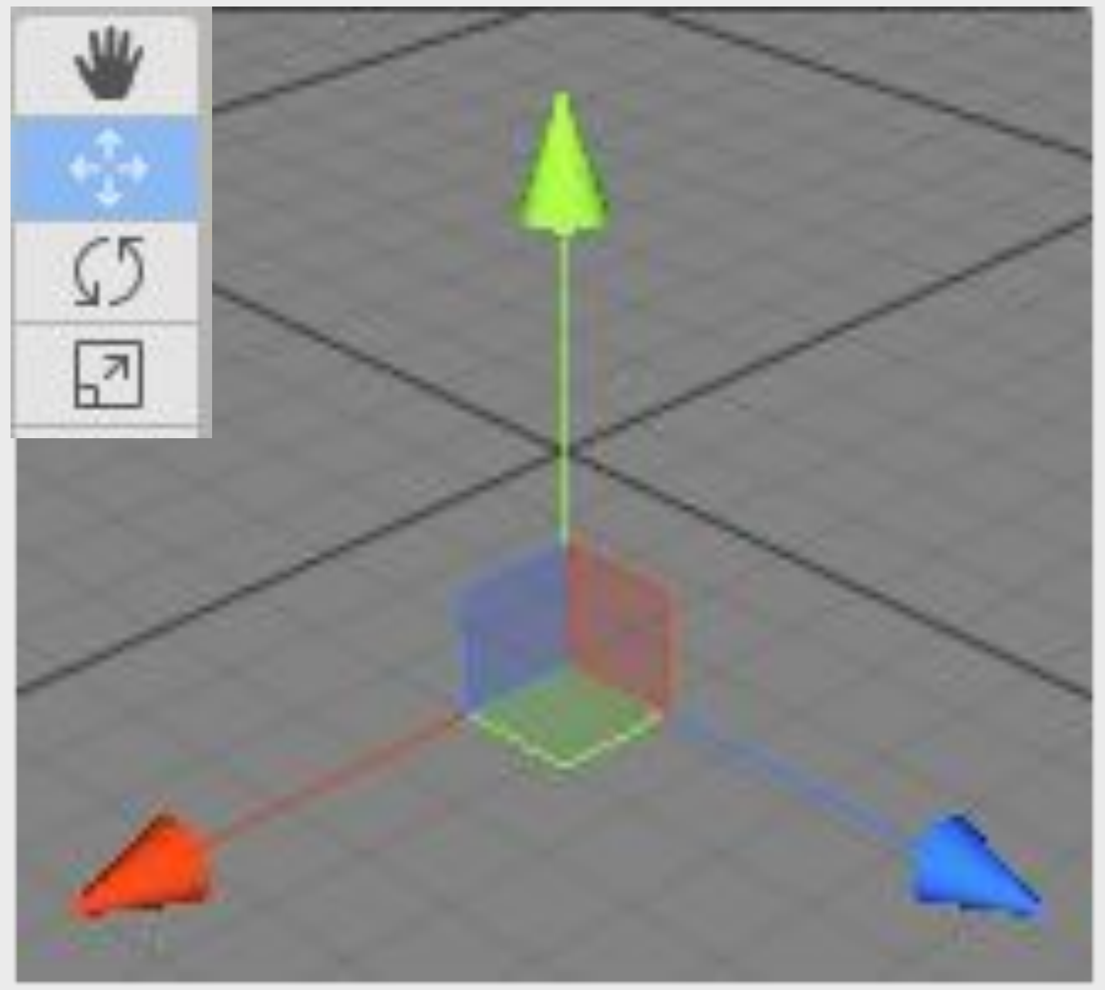
Tool Tip!
Top Left Icons



Q
W
E
R
T
Y

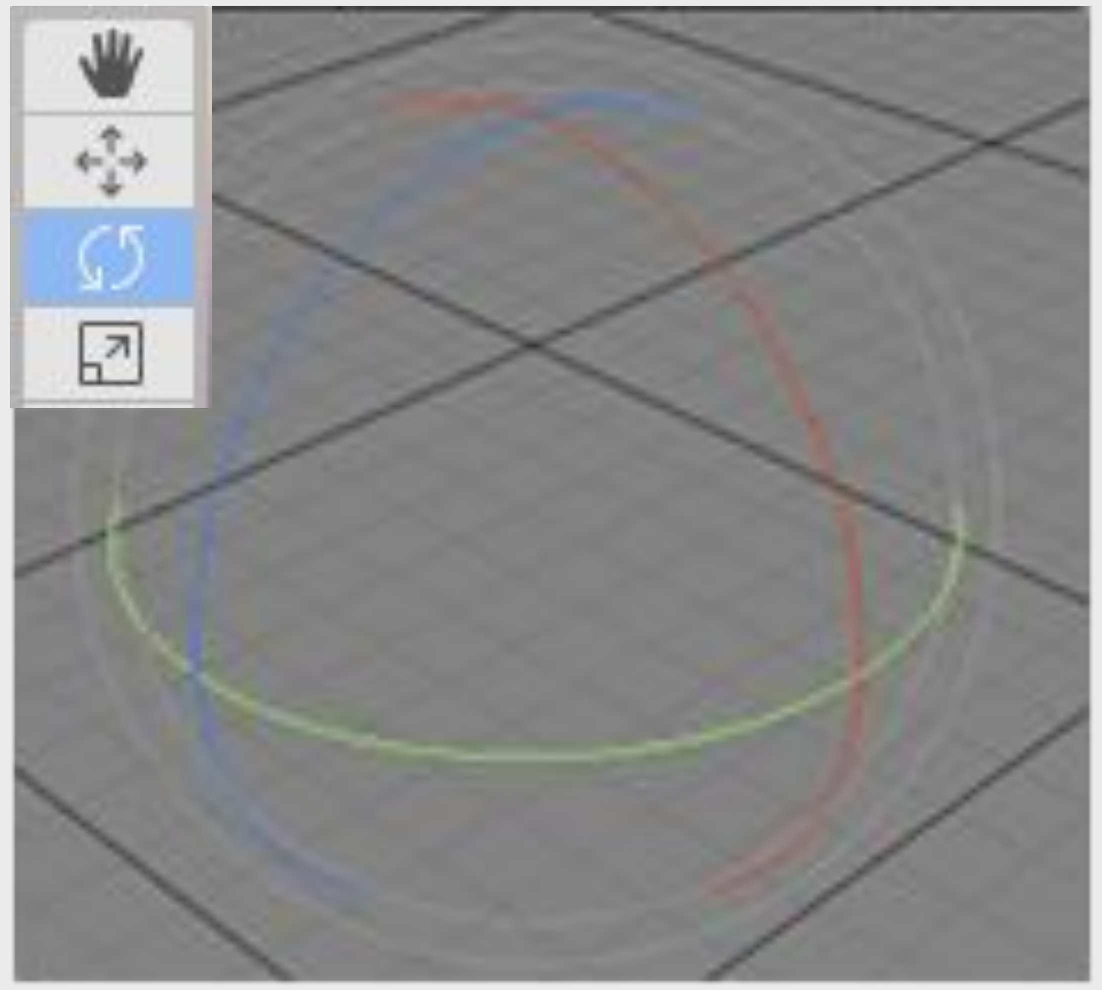
Use these tools to customize your objects in game

Tool Tip!
Icons on the top left



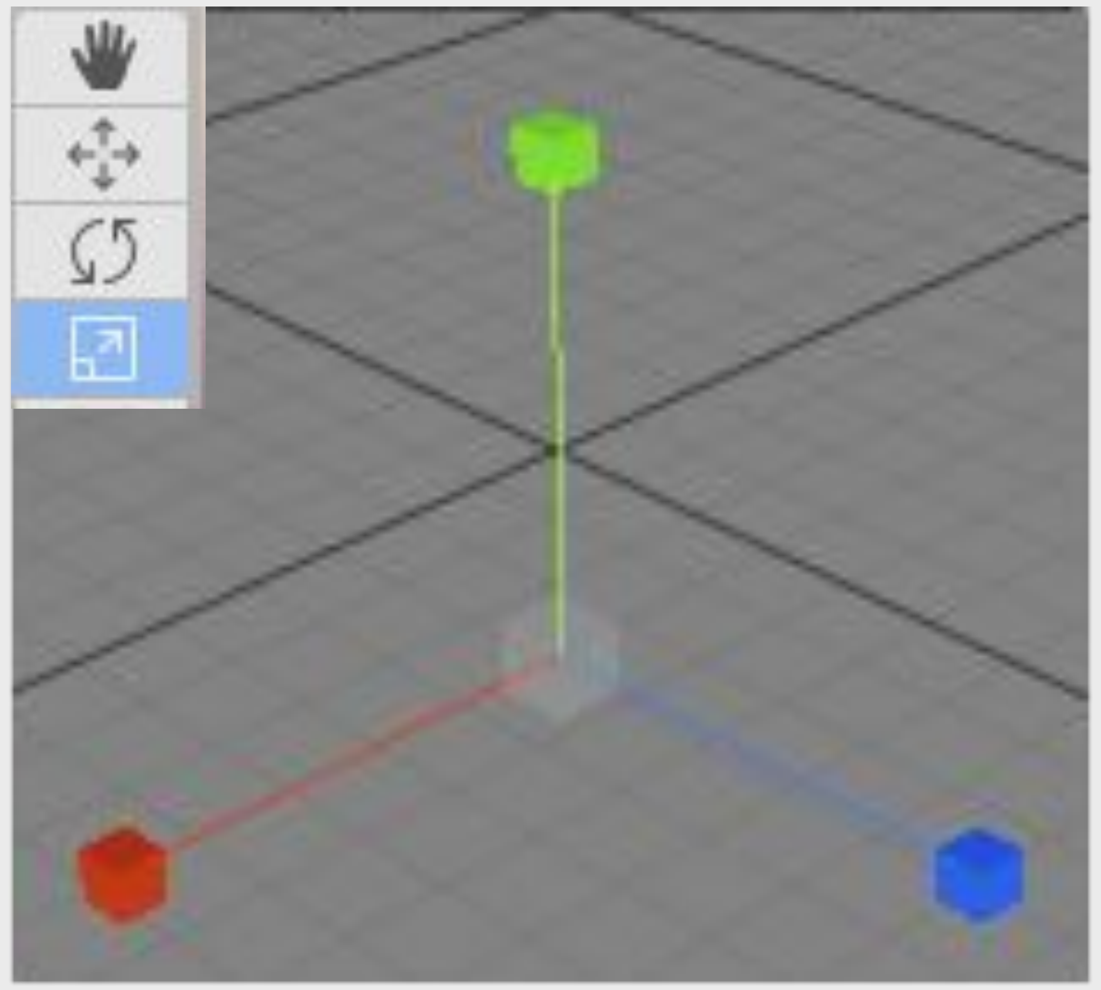
MOVE OBJECT

[w]



ROTATE OBJECT

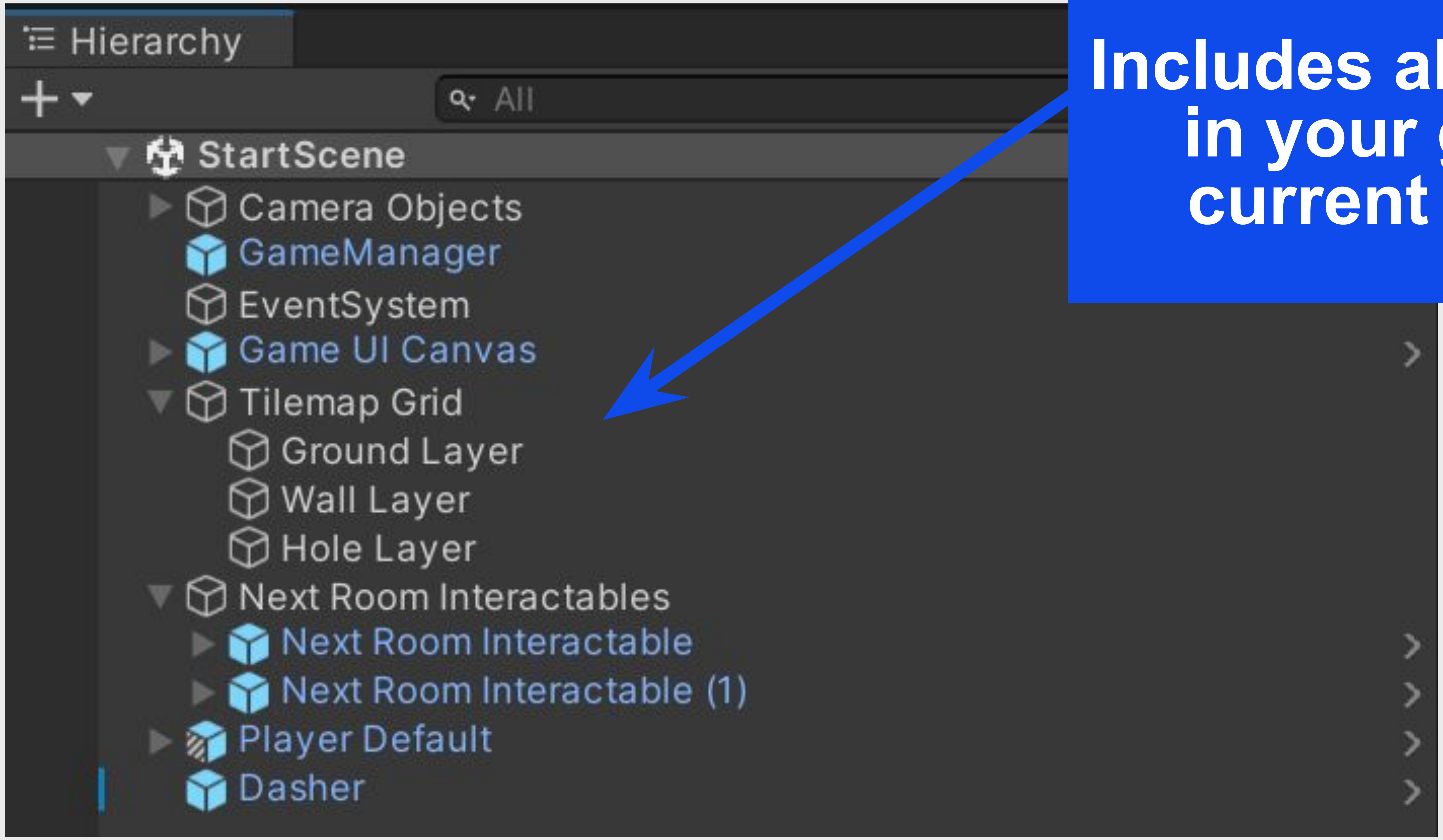
[e]



SCALE OBJECT

[r]

Tool Tip!
Hierarchy


Hierarchy

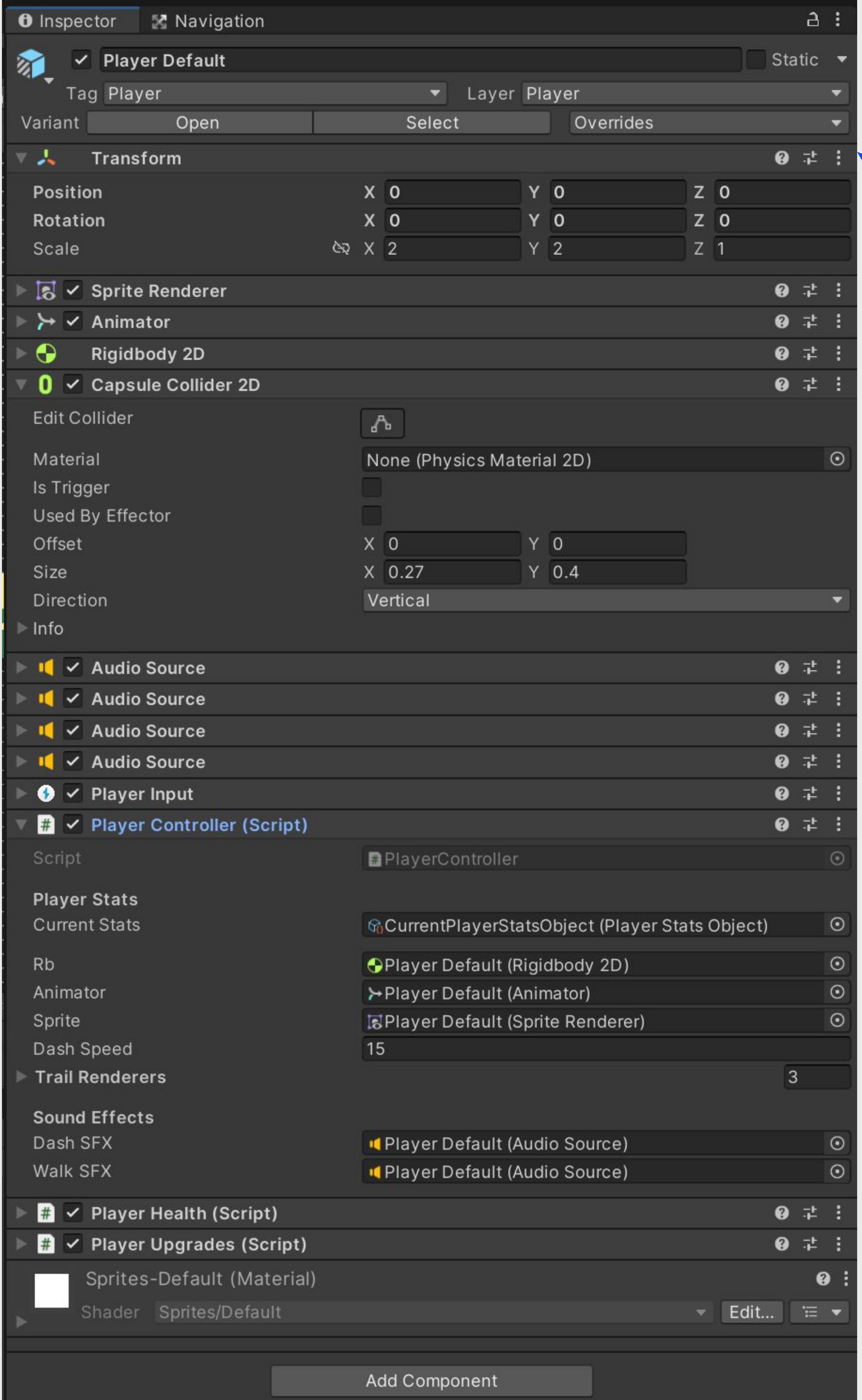
All

StartScene
Camera Objects
GameManager
EventSystem
Game UI Canvas
Tilemap Grid
Ground Layer
Wall Layer
Hole Layer
Next Room Interactables
Next Room Interactable
Next Room Interactable (1)
Player Default
Dasher

Includes all GameObjects in your game (in the current level/scene)

Tool Tip!
Inspector

Includes all things related to the object



Inspector | Navigation

Player Default — Static

Tag: Player | Layer: Player

Variant: Open | Select | Overrides

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	2	2	1

Sprite Renderer

Animator

Rigidbody 2D

Capsule Collider 2D

Edit Collider	
Material	None (Physics Material 2D)
Is Trigger	☐
Used By Effector	☐
Offset	X 0, Y 0
Size	X 0.27, Y 0.4
Direction	Vertical
Info	

Audio Source

Audio Source

Audio Source

Audio Source

Player Input

Player Controller (Script)

Script	PlayerController
Player Stats	
Current Stats	CurrentPlayerStatsObject (Player Stats Object)
Rb	Player Default (Rigidbody 2D)
Animator	Player Default (Animator)
Sprite	Player Default (Sprite Renderer)
Dash Speed	15
Trail Renderers	3
Sound Effects	
Dash SFX	Player Default (Audio Source)
Walk SFX	Player Default (Audio Source)

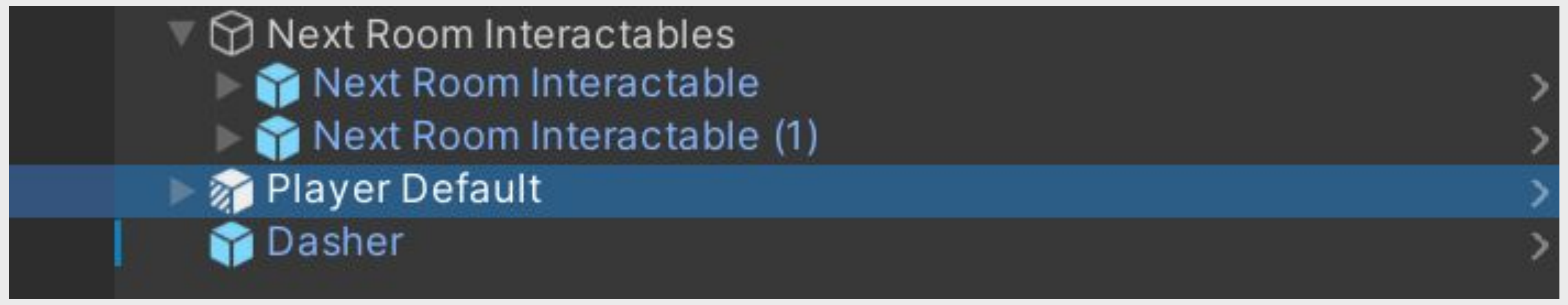
Player Health (Script)

Player Upgrades (Script)

Sprites-Default (Material)

Shader: Sprites/Default | Edit...

Add Component

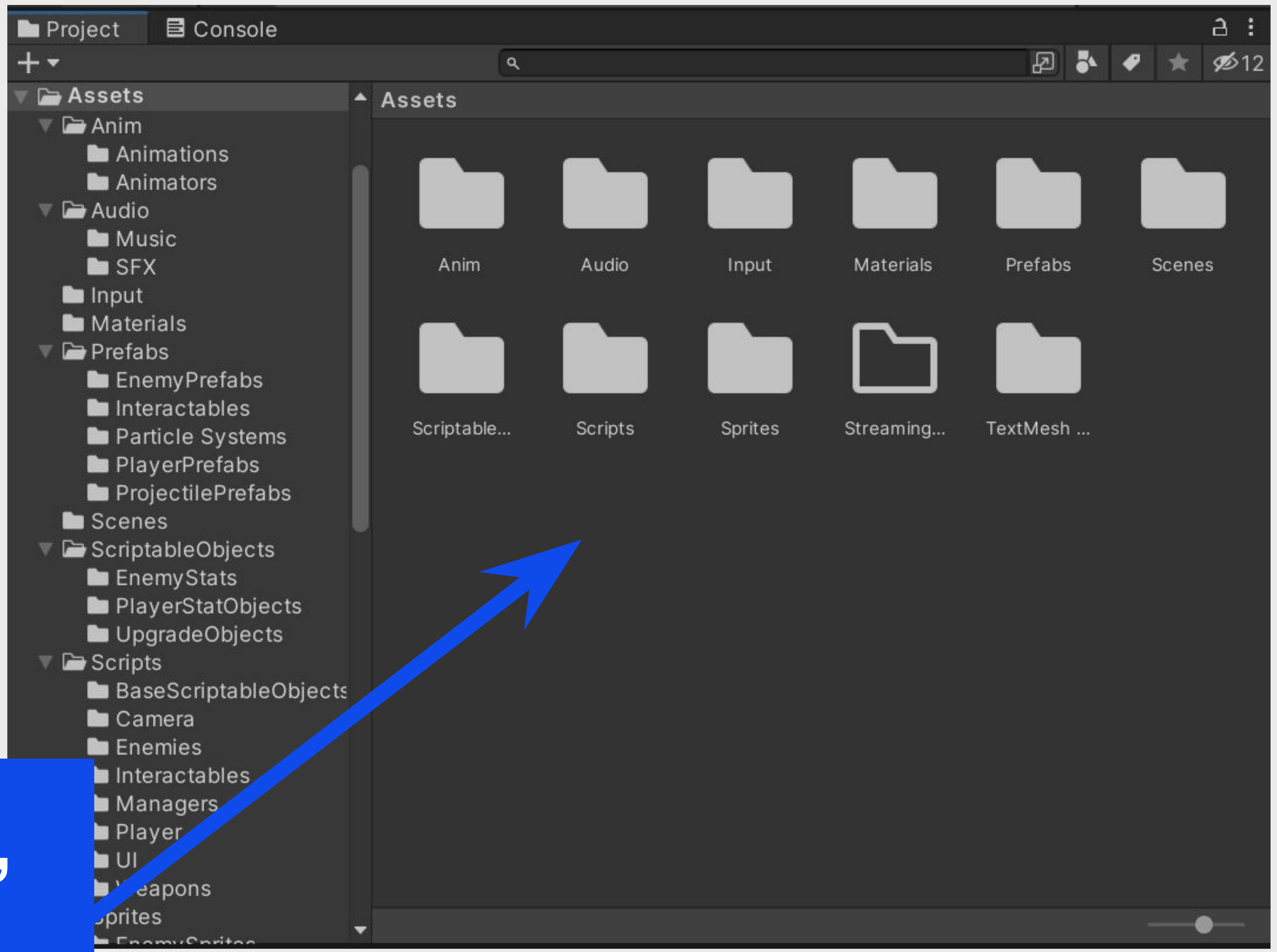


Next Room Interactables
- Next Room Interactable
- Next Room Interactable (1)

Player Default

Dasher

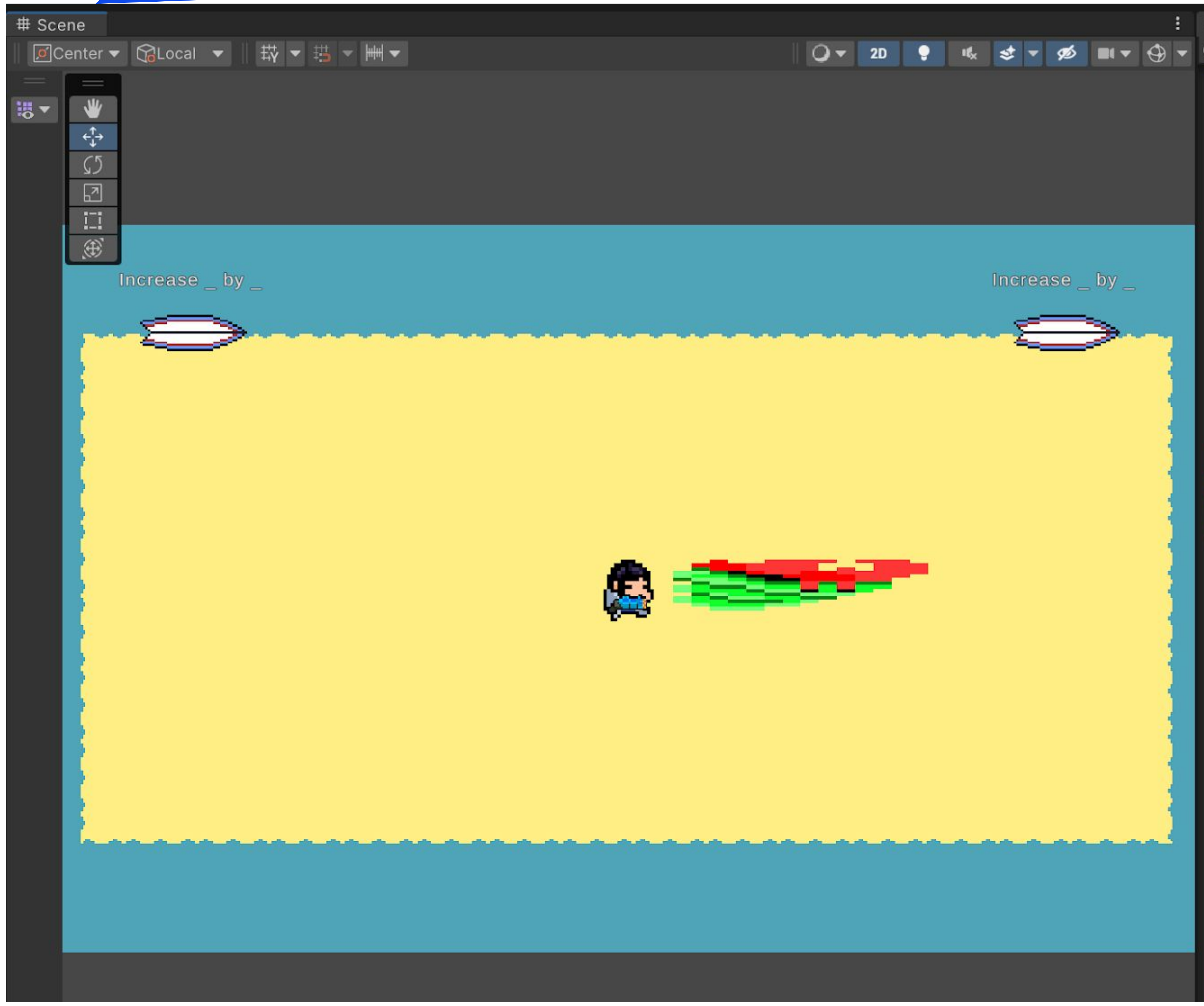
Tool Tip!
Project Window - Folders and Assets

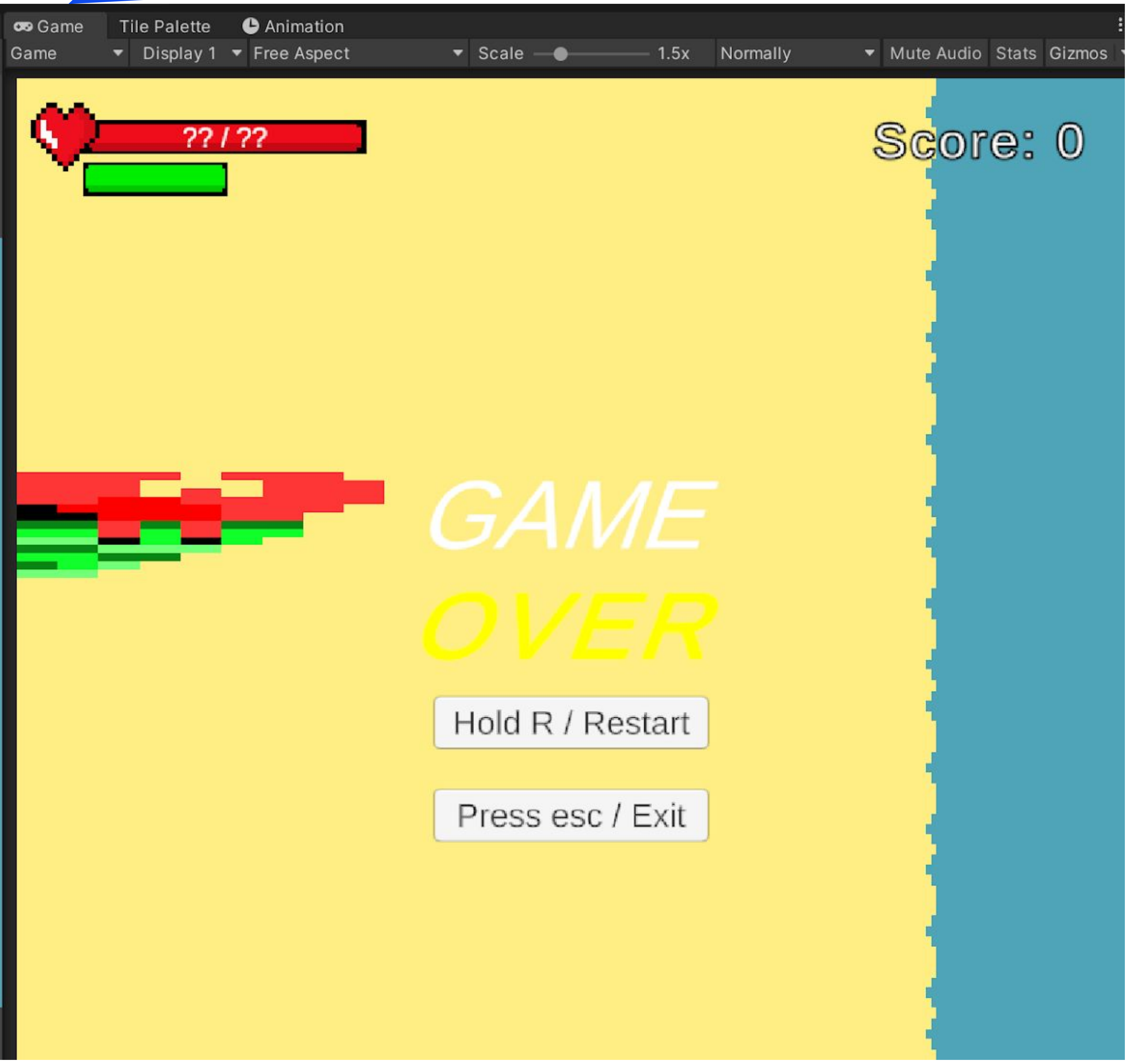


Includes scripts, music, characters and level design objects

Move objects around in the Scene Tab

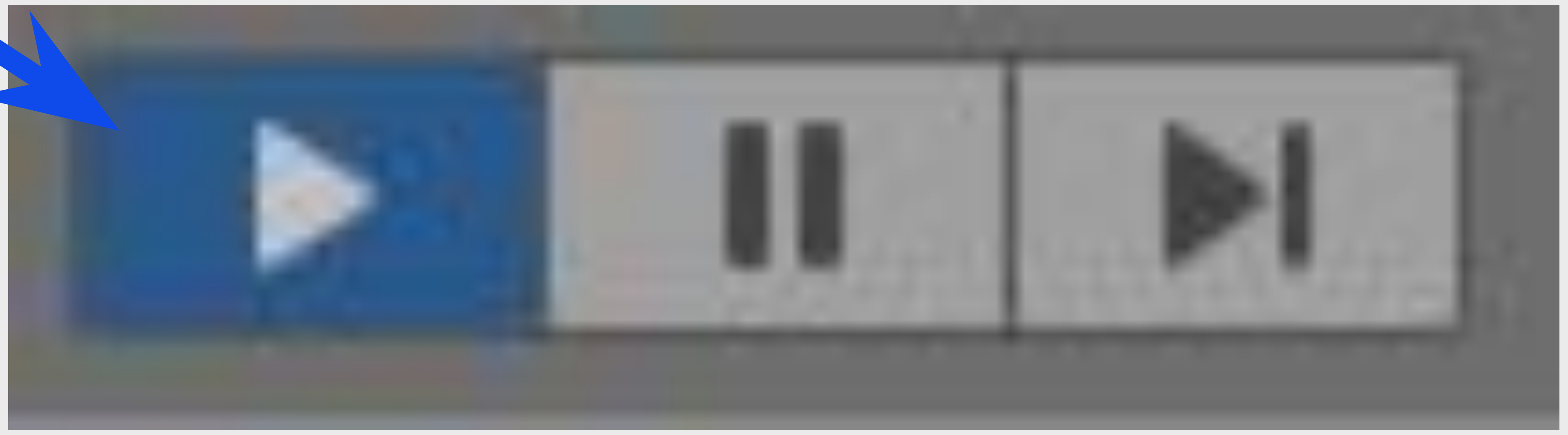
Test your game in the Game Tab





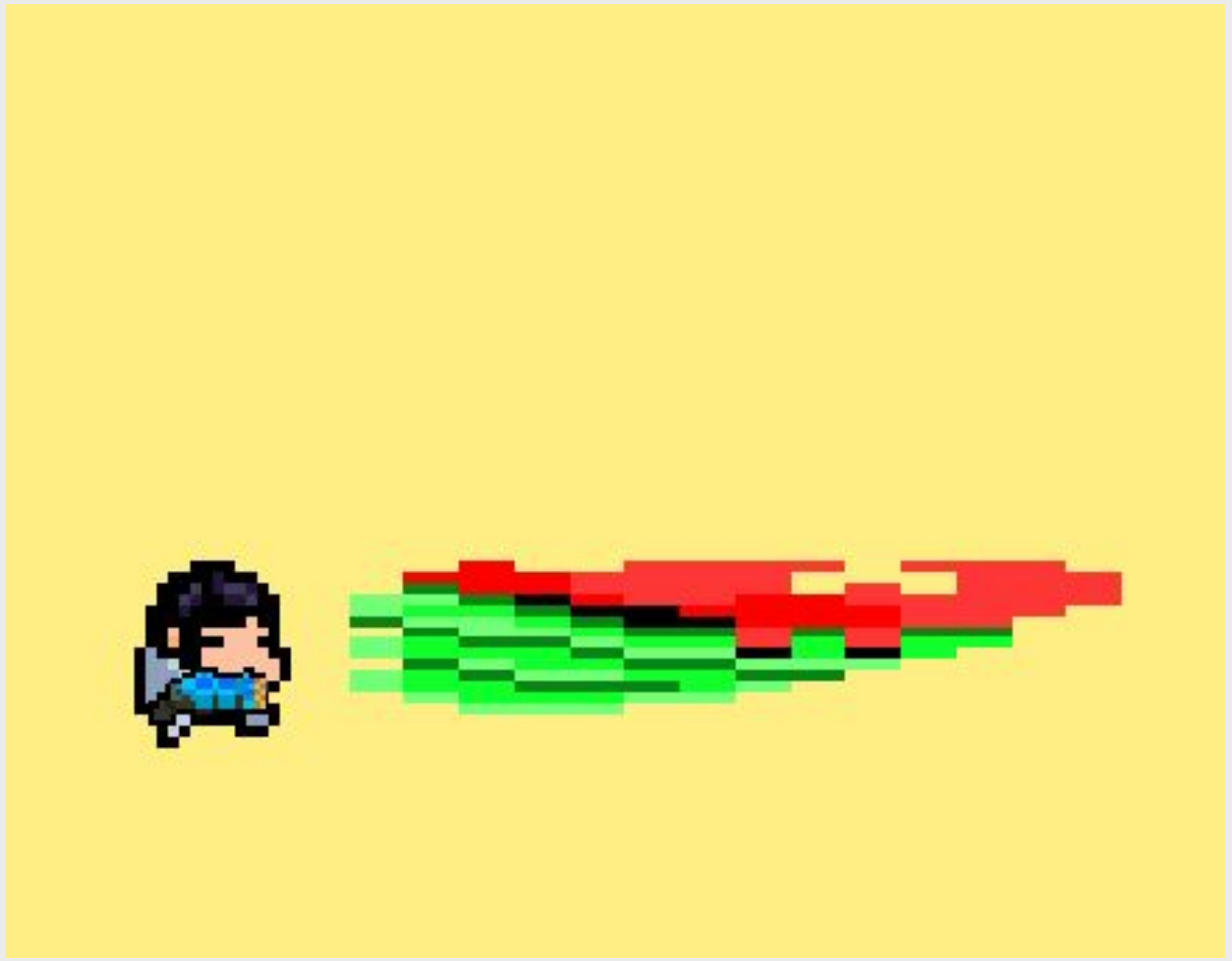
Tool Tip!
Middle Icons

Turn it off before editing!



Click ‘Play’ to test the game

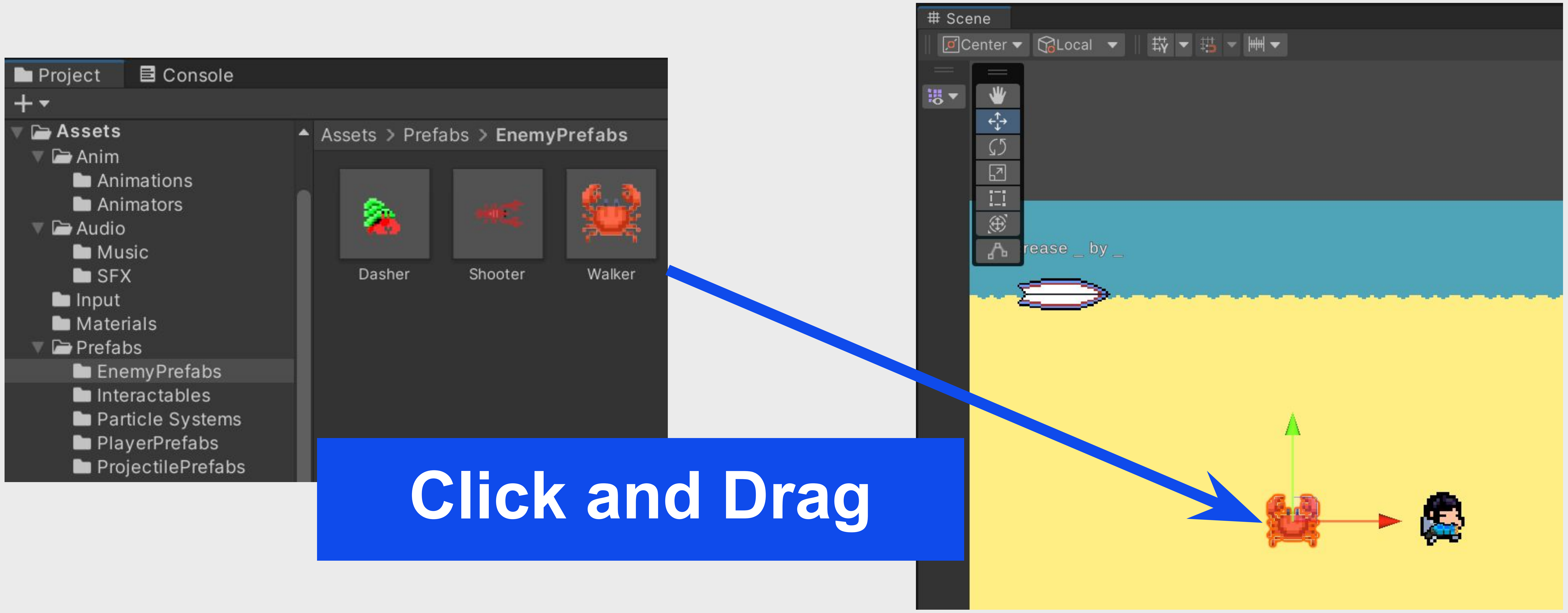
First task! Fixing the Hermit Crab



Step 1. Identify what is wrong with this guy

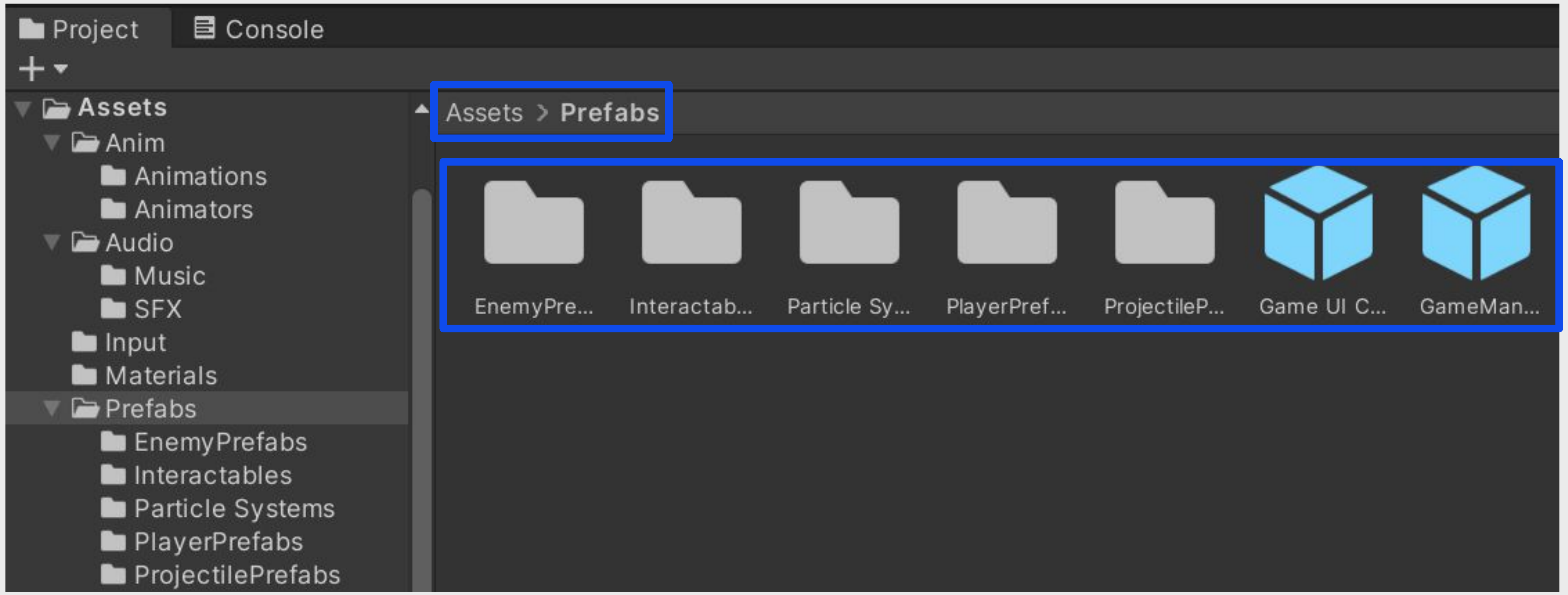
Step 2. Use your knowledge of tools to fix him up

Getting stuck? Ask questions
(to each other and to me)



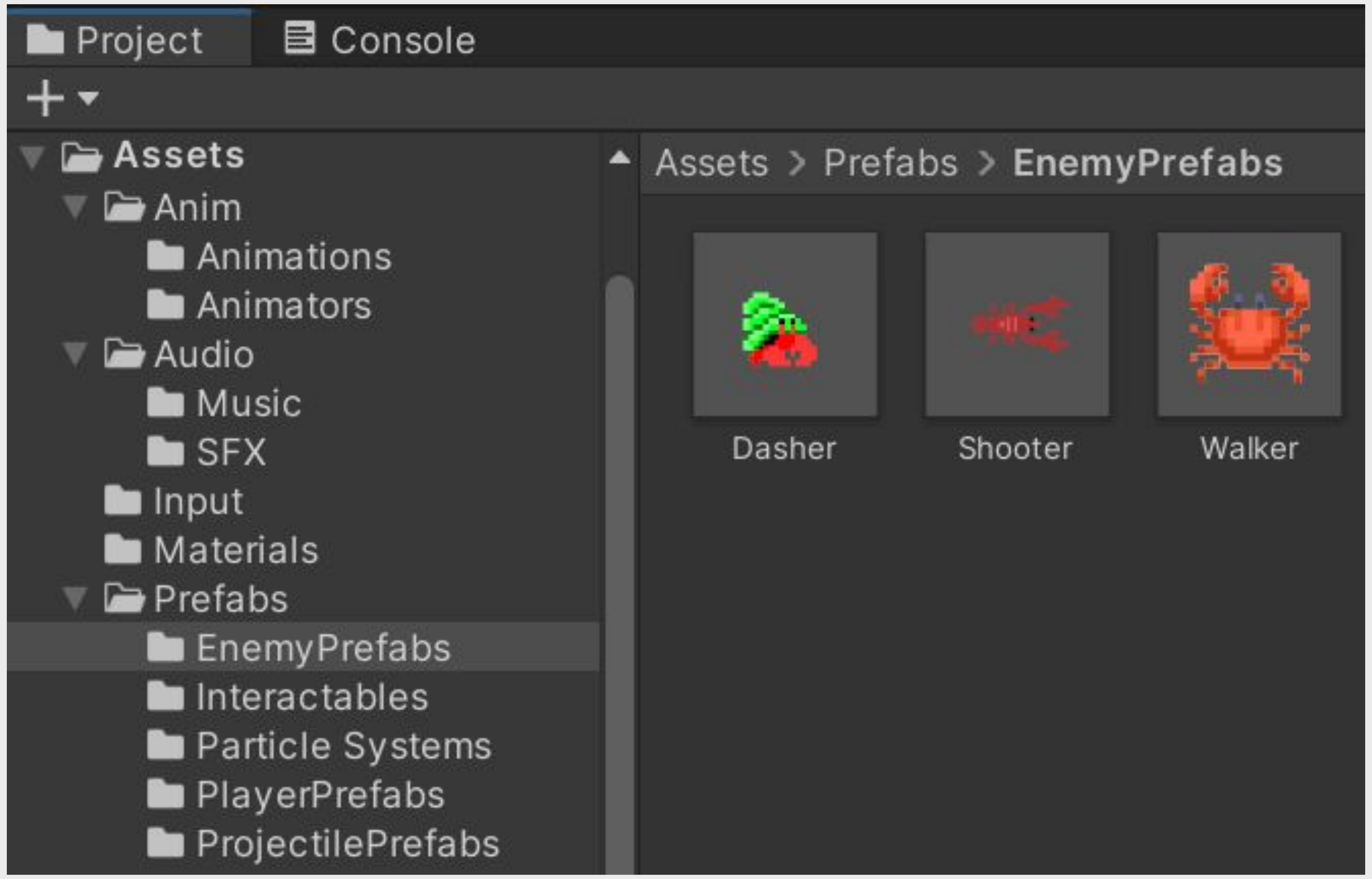
Too easy? Try making a more interesting level by adding more enemies

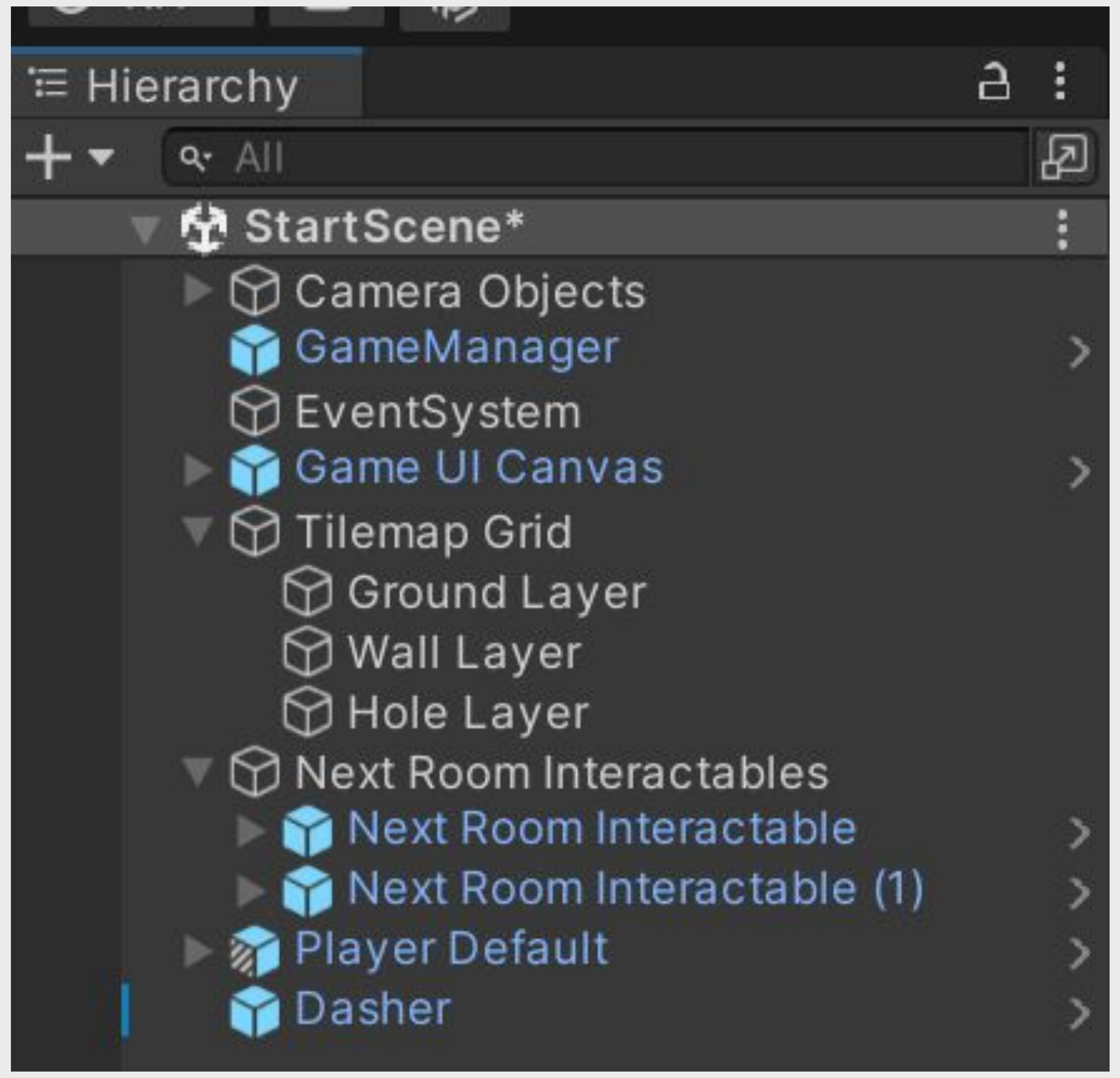
Let’s talk Prefabs



Prefabs

GameObjects that have been saved for reuse (you actually have added some prefabs in your scene already)

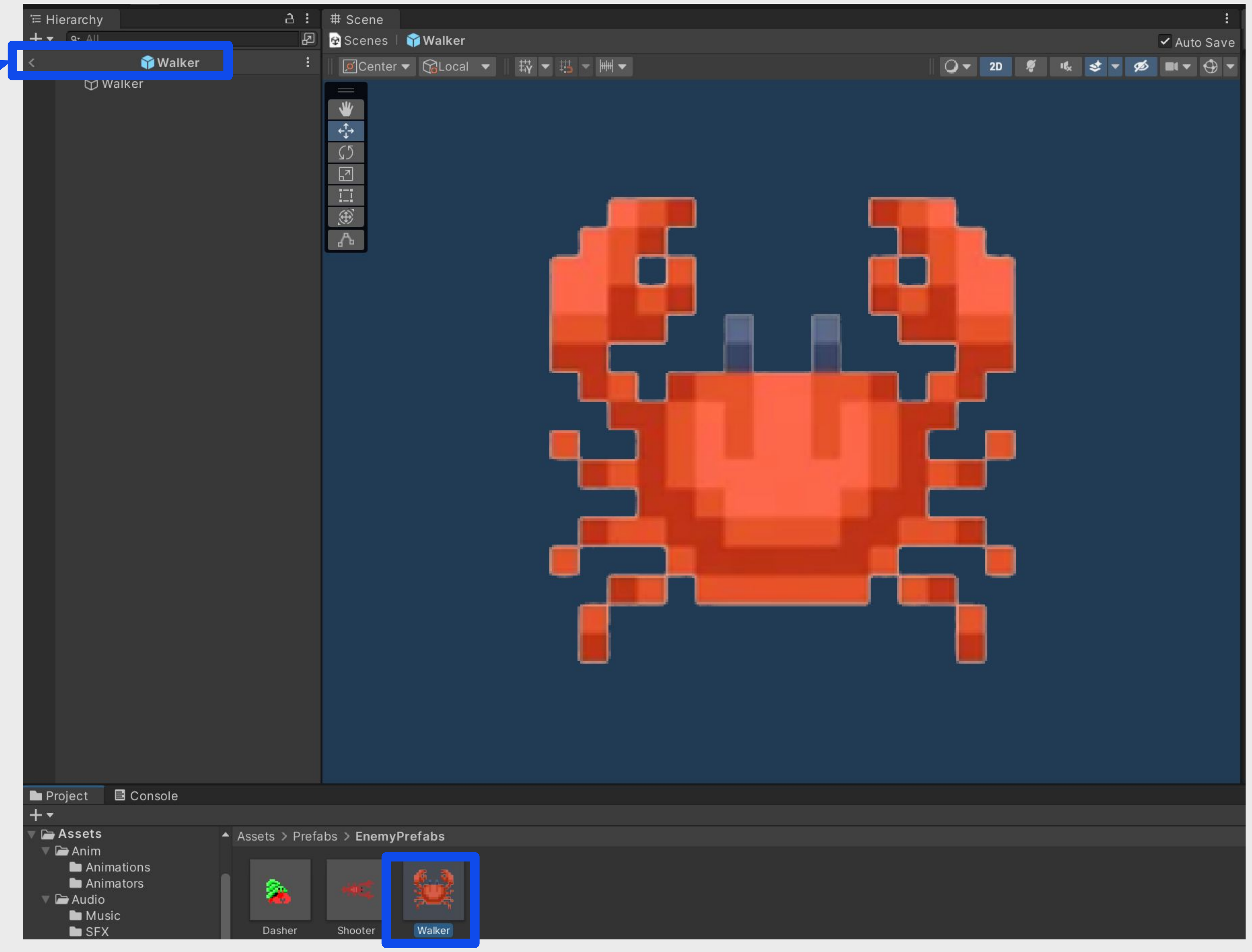




Prefabs have a blue icon and text

Prefabs

Double click a prefab in the project window to open it up in it's own scene

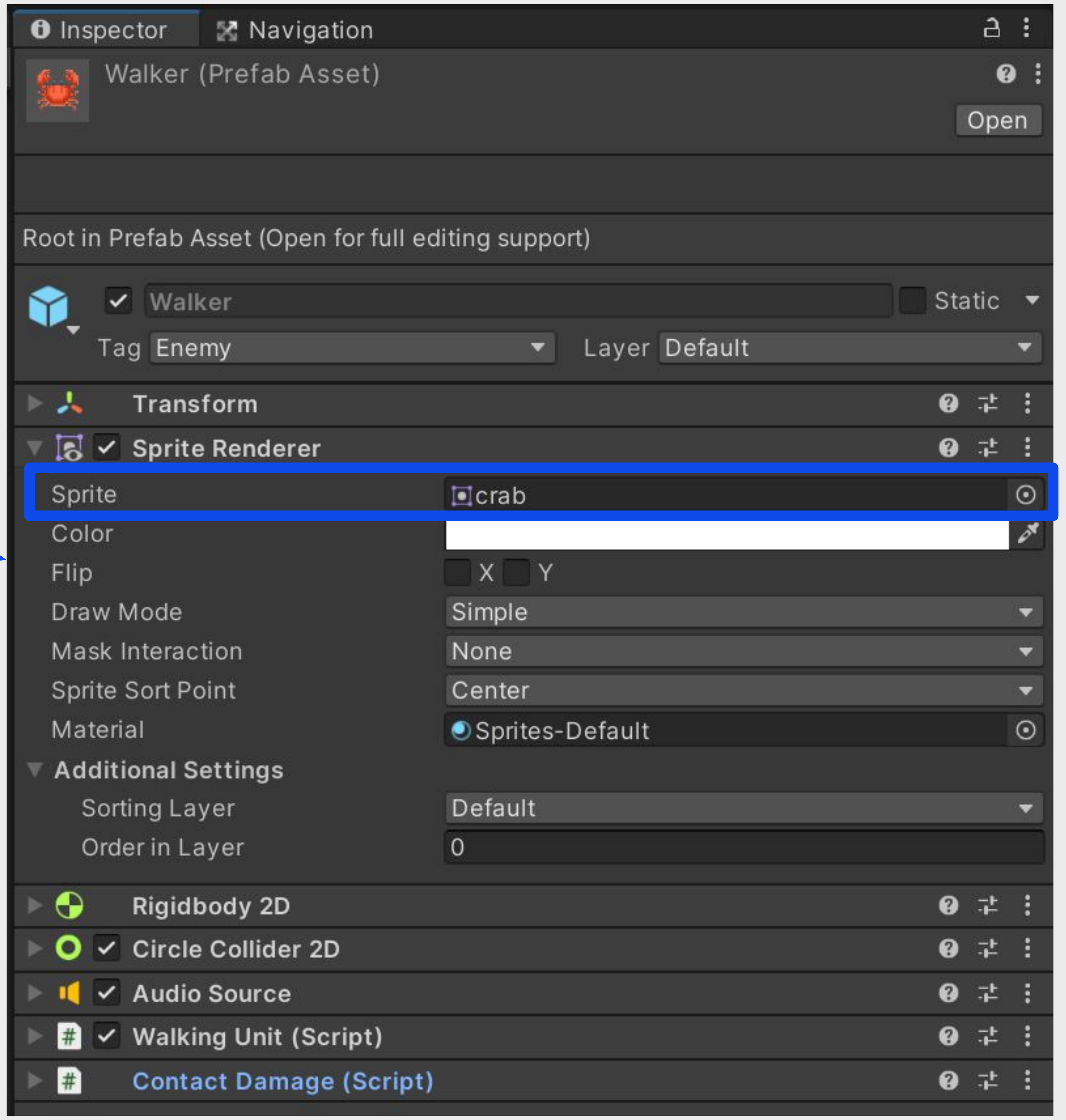


Return back to the level scene

Prefabs

Useful for making changes to all of the same GameObjects in the scene

Project Window



Walker (Prefab Asset)

Root in Prefab Asset (Open for full editing support)

Walker Static
Tag Enemy Layer Default

Transform

Sprite Renderer

Sprite	crab
Color	
Flip	X Y
Draw Mode	Simple
Mask Interaction	None
Sprite Sort Point	Center
Material	Sprites-Default

Additional Settings

Sorting Layer	Default
Order in Layer	0

Rigidbody 2D

Circle Collider 2D

Audio Source

Walking Unit (Script)

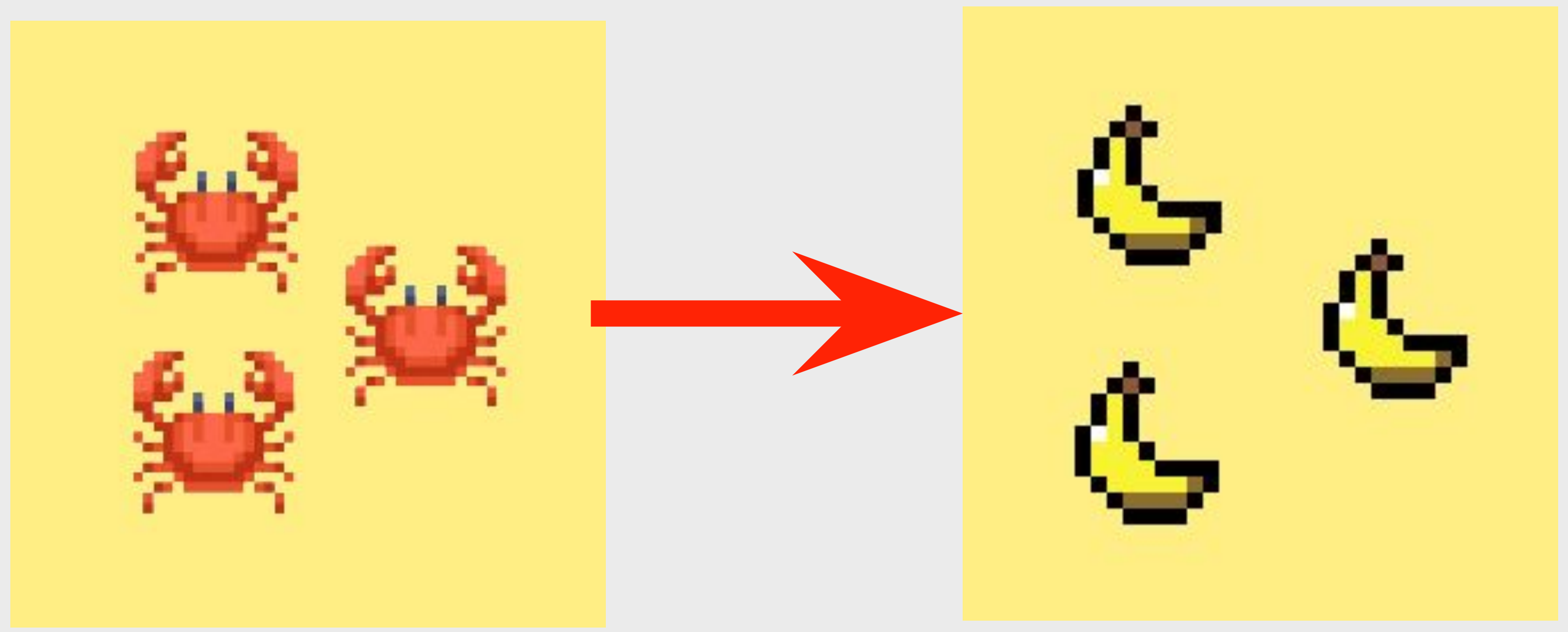
Contact Damage (Script)

E.g. We can change the sprite in the prefab to turn all

Crab Walkers

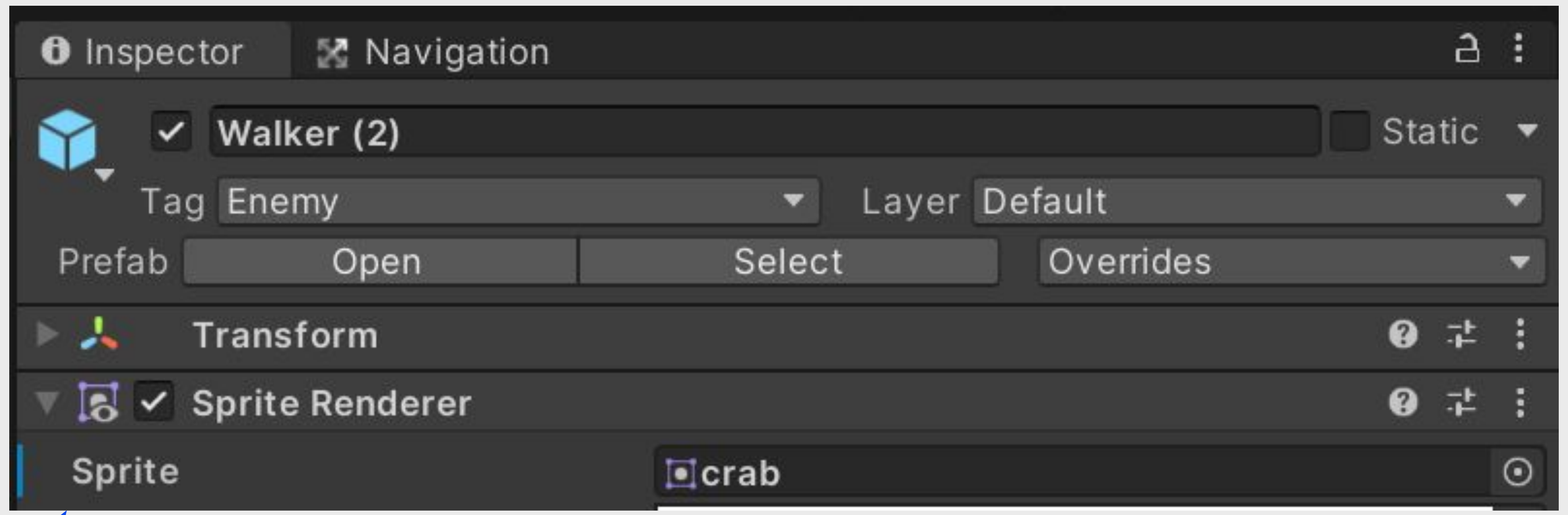
Into

Banana Walkers

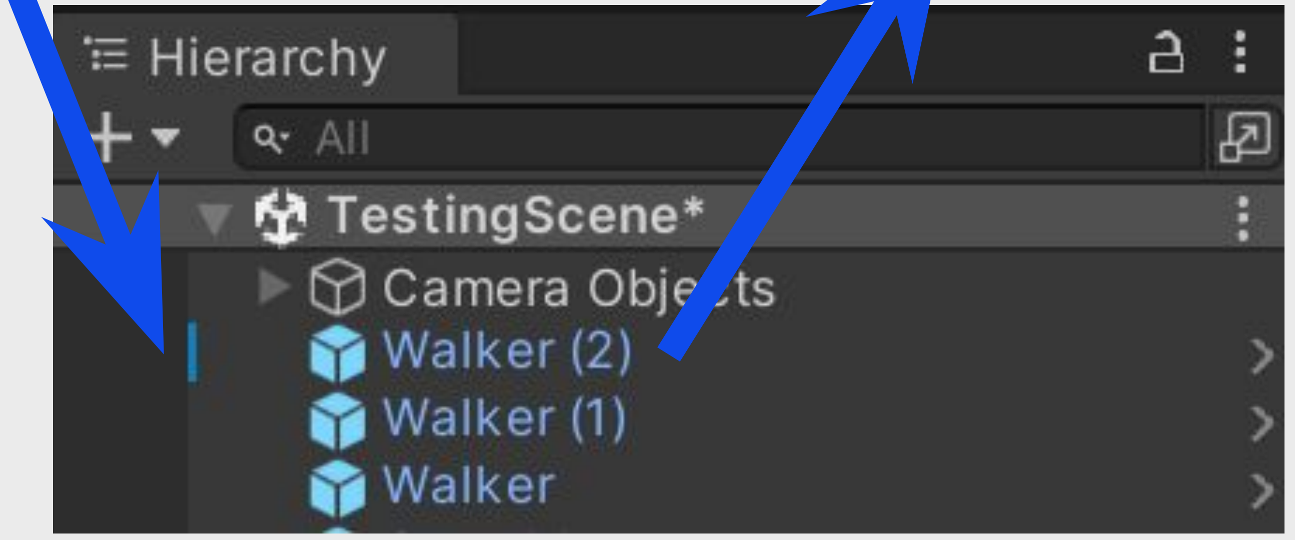


Prefabs

HOWEVER, these changes will be overridden if we have modified the prefab in the scene



Walker (2)
Tag Enemy
Layer Default
Sprite crab

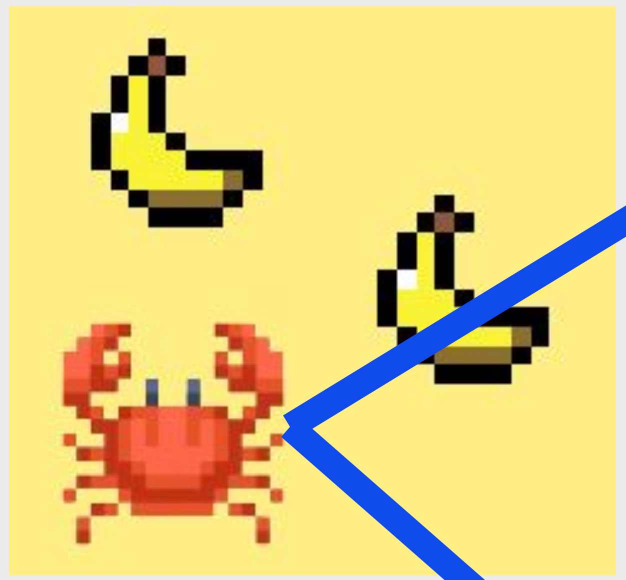


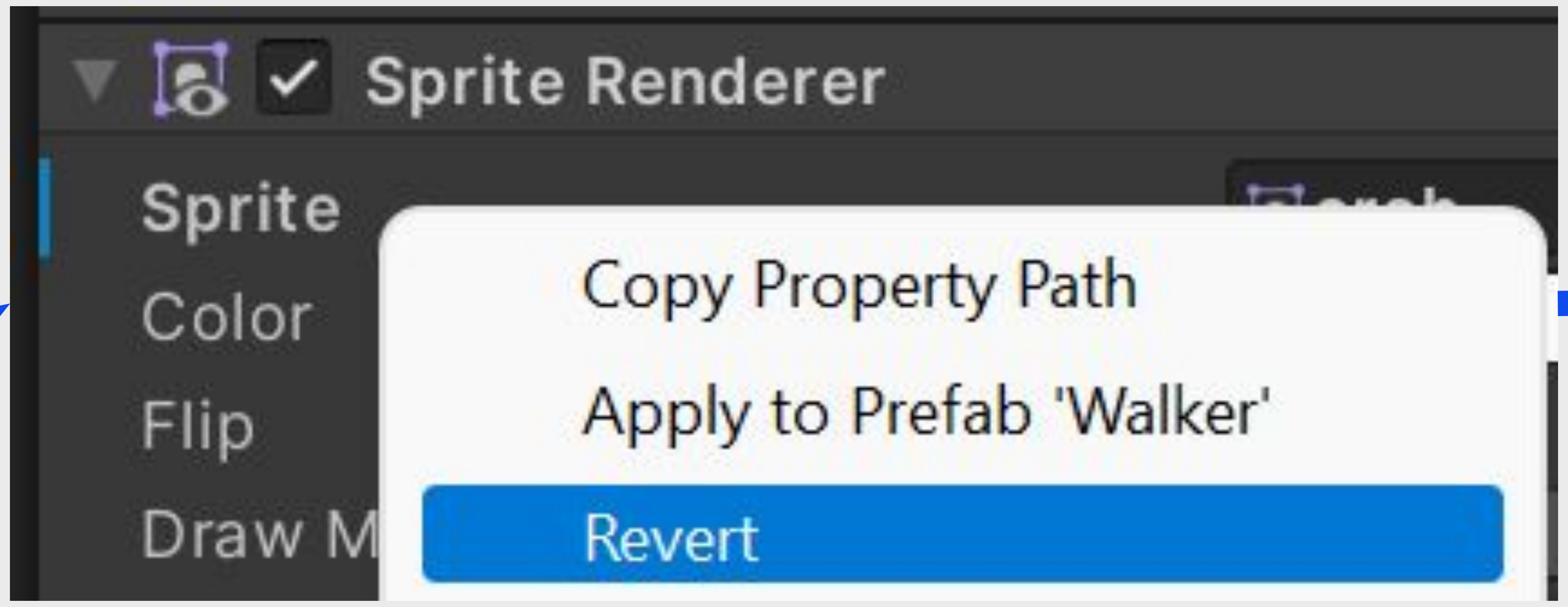
TestingScene*
Camera Objects
Walker (2)
Walker (1)
Walker

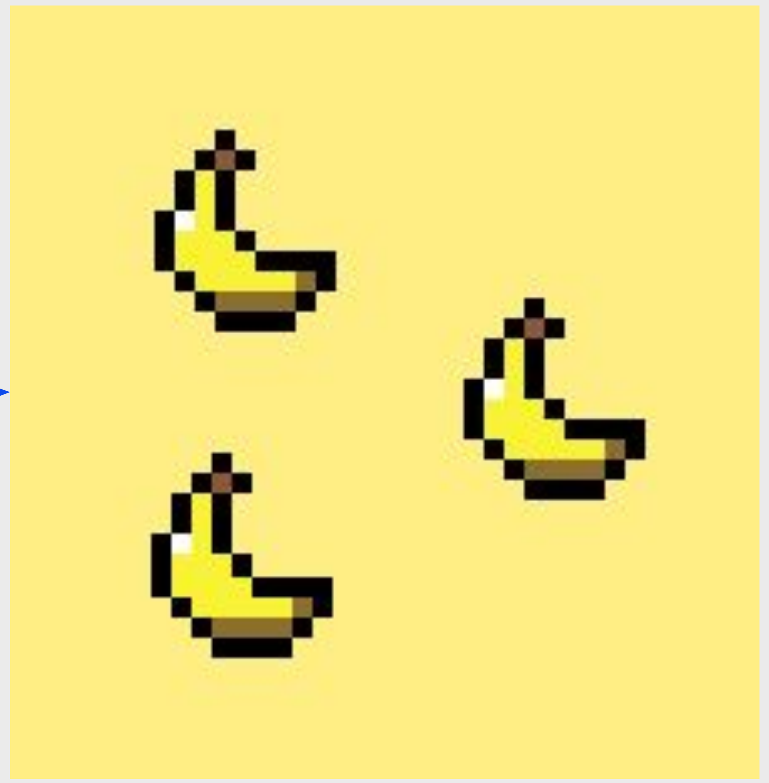
The small blue line shows that the prefab has been changed in the scene. Crab is Walker (2)

Prefabs

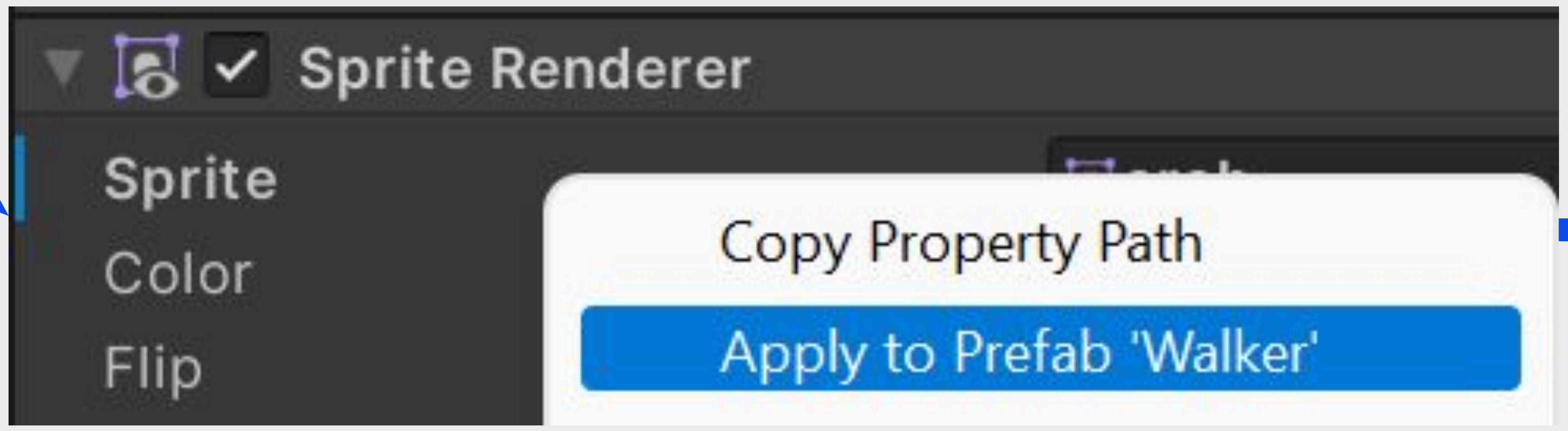
You can revert the prefab attribute back to it's prefab state by right clicking and selecting Revert

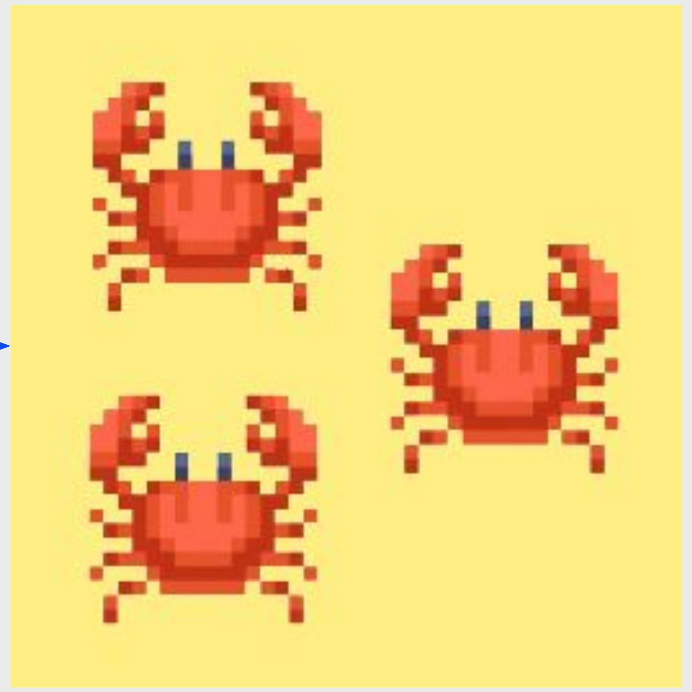






Alternatively, you can also Apply the change to the prefab to affect every other prefab of the same type

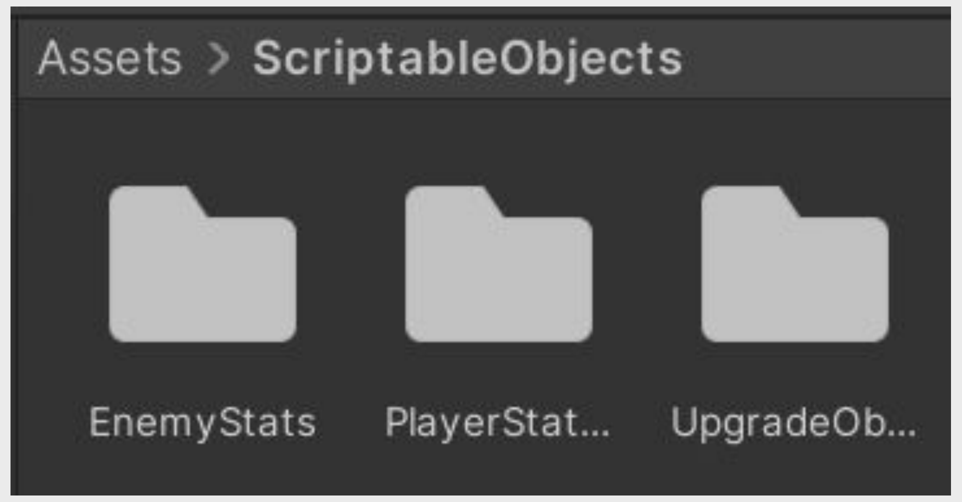


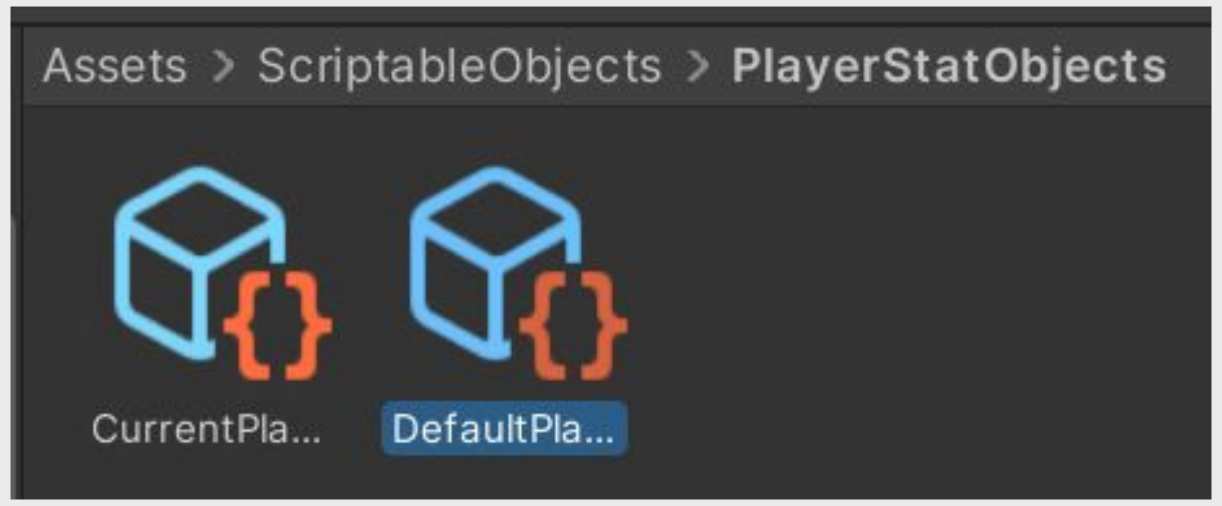


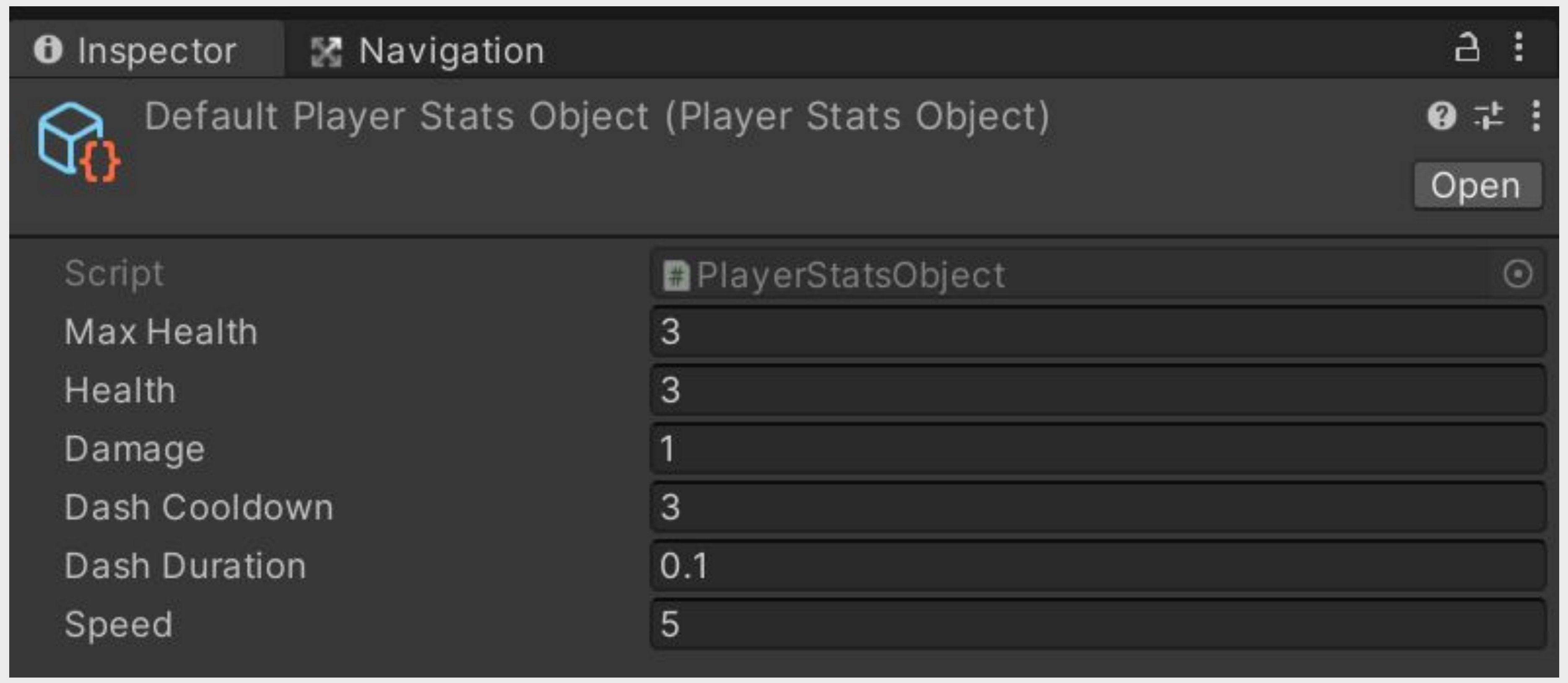
What scenarios (other than enemies) might prefabs be useful?

Scriptable Objects

Simple version of prefabs. They hold data only





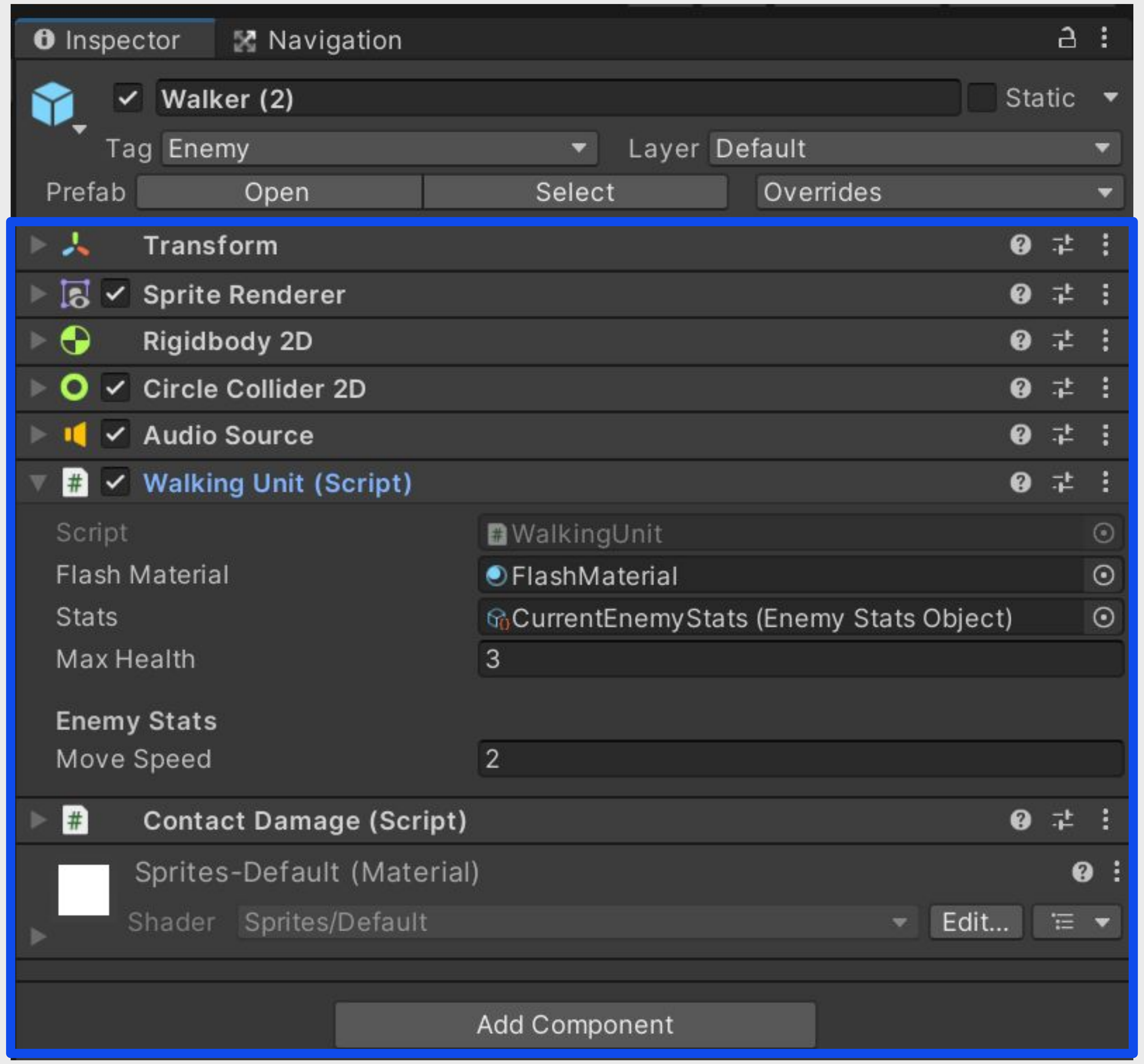


Default Player Stats Object (Player Stats Object)

Script	PlayerStatsObject
Max Health	3
Health	3
Damage	1
Dash Cooldown	3
Dash Duration	0.1
Speed	5

Scriptable objects save even during play mode. Useful for storing changing game data (E.g. player stats upgrading)

Let’s talk Components



Inspector | Navigation

✓ Walker (2) | Static

Tag: Enemy | Layer: Default

Prefab: Open | Select | Overrides

- Transform
- ✓ Sprite Renderer
- Rigidbody 2D
- ✓ Circle Collider 2D
- ✓ Audio Source
- ✓ Walking Unit (Script)

```
Script          WalkingUnit
Flash Material  FlashMaterial
Stats           CurrentEnemyStats (Enemy Stats Object)
Max Health      3

Enemy Stats
Move Speed      2
```

- Contact Damage (Script)

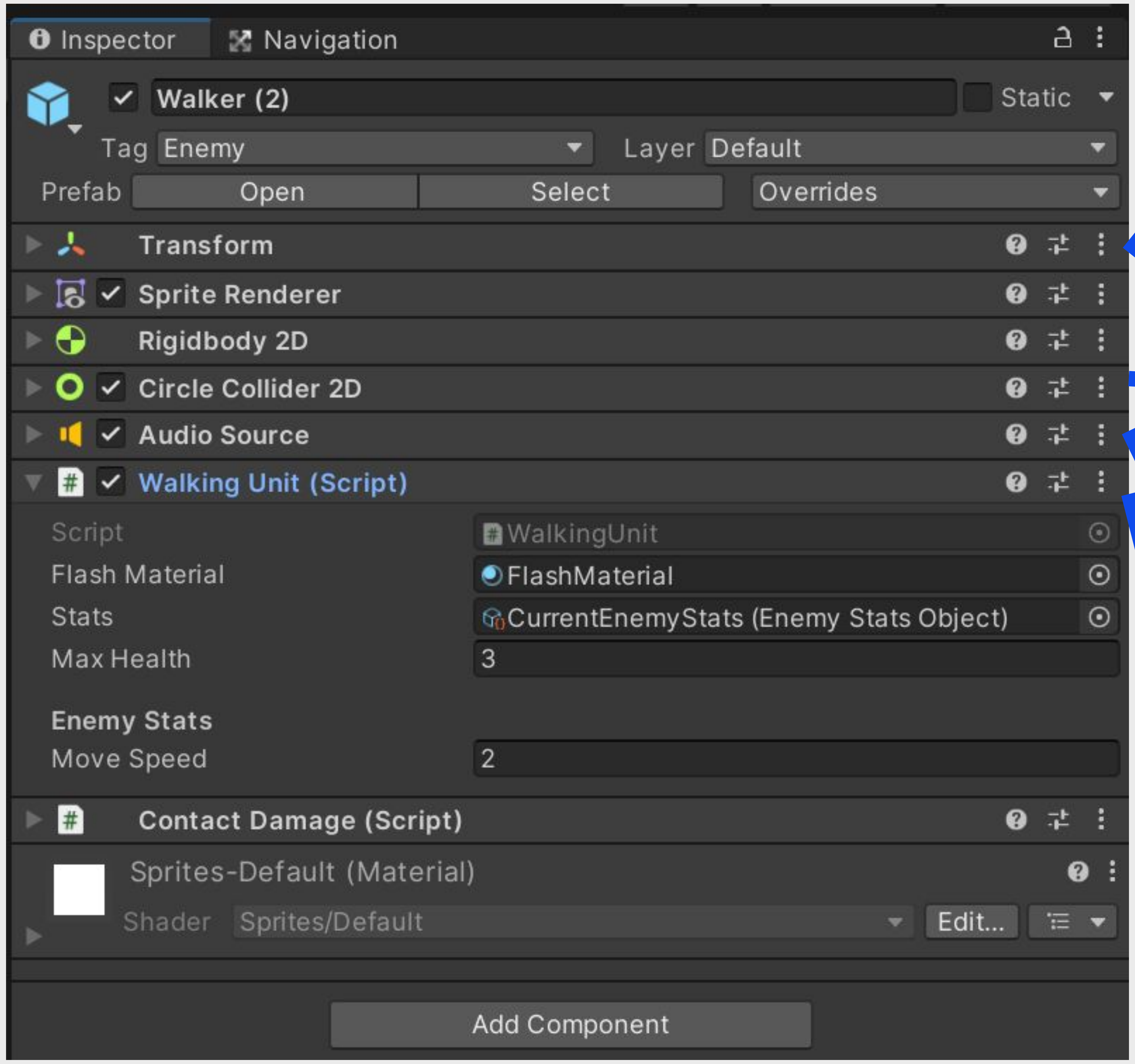
Sprites-Default (Material)

Shader: Sprites/Default | Edit...

Add Component

Components

Attachments to GameObjects. Components are the added properties to GameObjects


Inspector | Navigation

Walker (2) — Static
Tag: Enemy — Layer: Default
Prefab: Open | Select | Overrides

Transform
Sprite Renderer
Rigidbody 2D
Circle Collider 2D
Audio Source
Walking Unit (Script)

Script	WalkingUnit
Flash Material	FlashMaterial
Stats	CurrentEnemyStats (Enemy Stats Object)
Max Health	3

Enemy Stats

Move Speed	2

Contact Damage (Script)
Sprites-Default (Material)
Shader: Sprites/Default — Edit...

Add Component

Transform sets position, rotation and scale

Circle collider gives the enemy a hitbox+hurtbox

AudioSource can be played for a sound effect

WalkingUnit is a custom script with simple AI

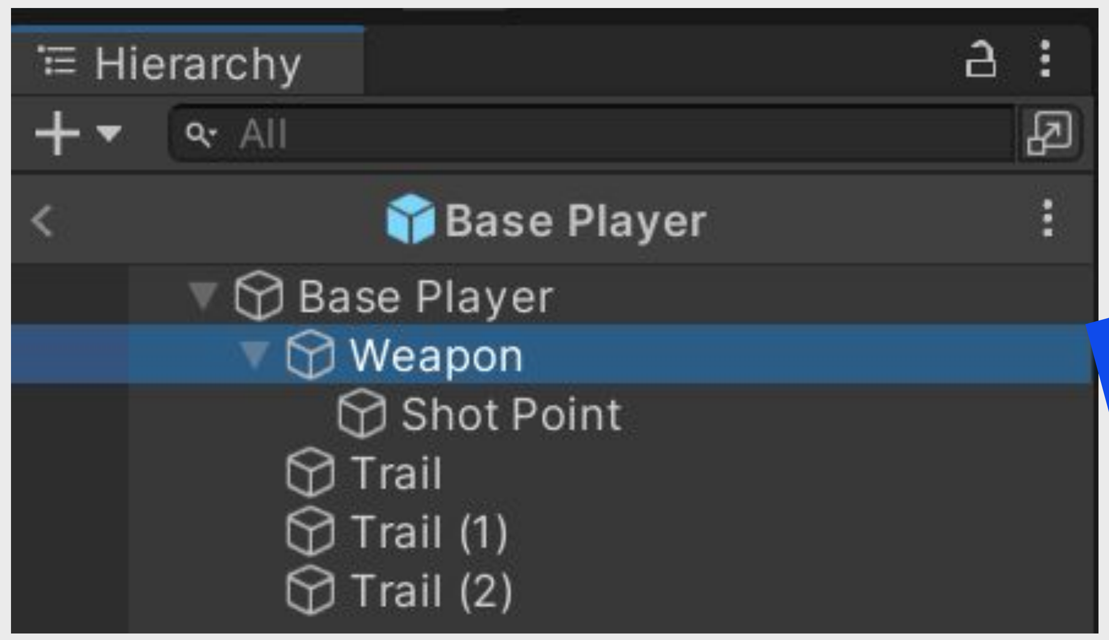
Second task! Modifying stats

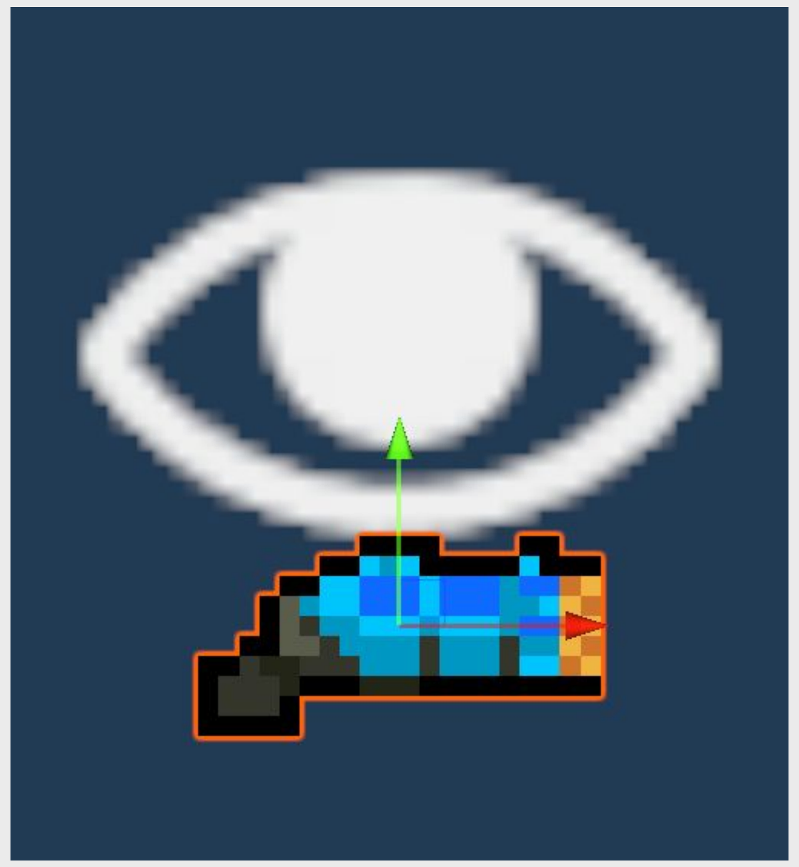
- Modify your enemy's starting stats (e.g., Damage, speed and HP) Saved on the enemy prefabs
- Modify the default player stats Saved in scriptable objects
- How do you modify the stats? Use your knowledge of Prefabs and Components to find out
- Strike a good difficulty balance. Have someone test out the game

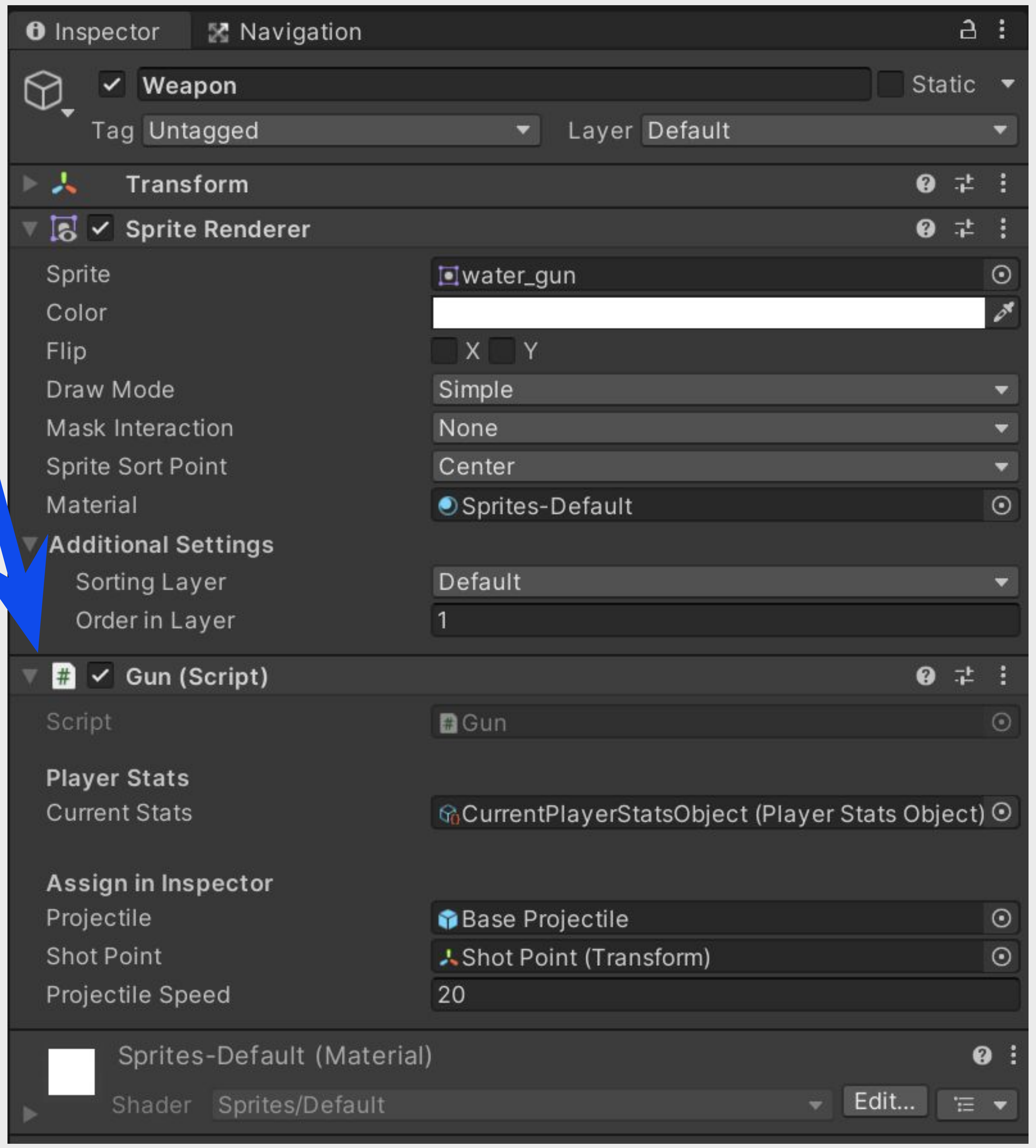
Third Task. Lets finally look at fixing the gun implementation!

Coding

- Start by finding the player prefab
- Locate the “Weapon” on the player character





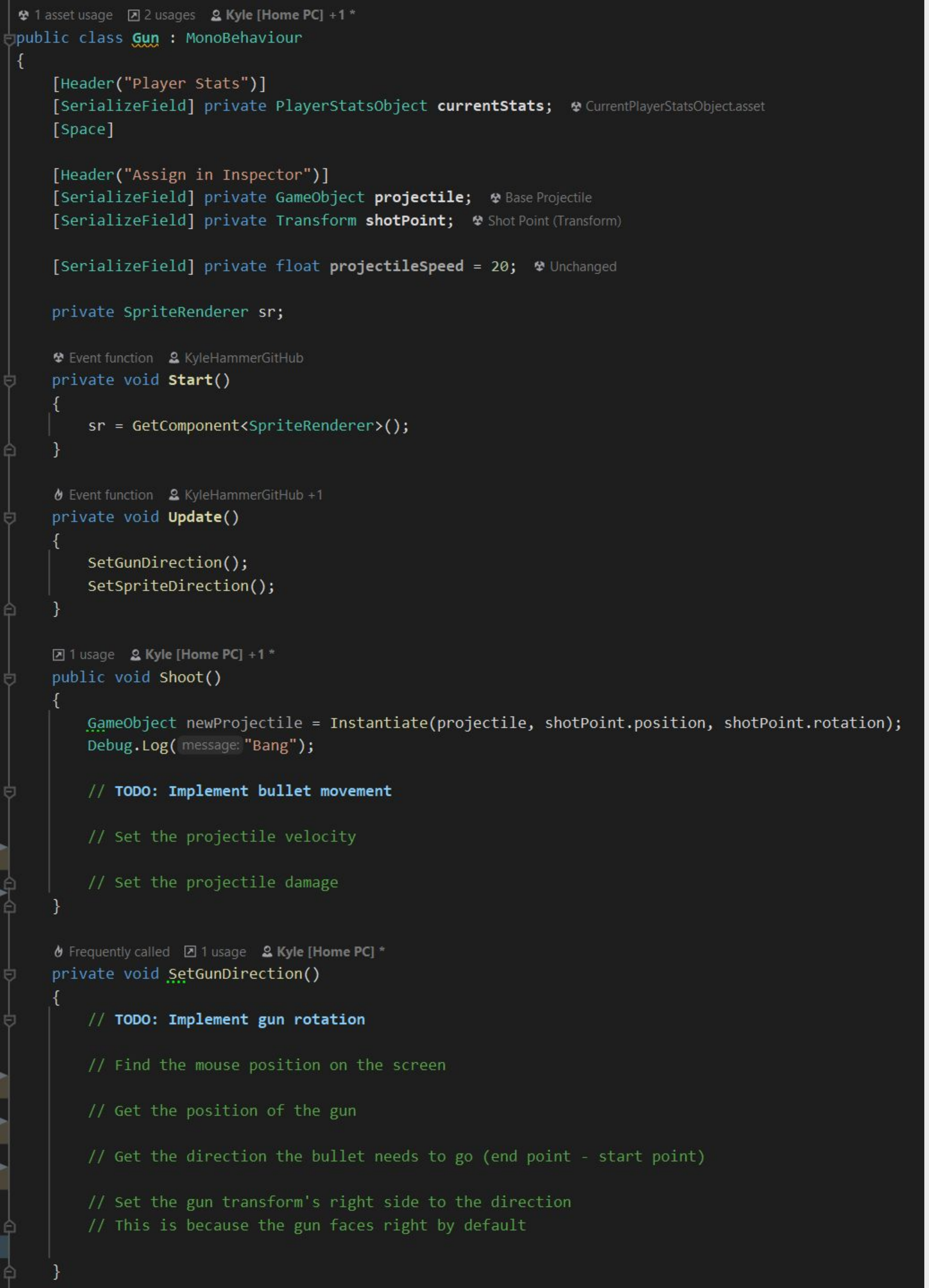


Weapon
Tag Untagged Layer Default
Transform
Sprite Renderer
Sprite water_gun
Color
Flip X Y
Draw Mode Simple
Mask Interaction None
Sprite Sort Point Center
Material Sprites-Default
Additional Settings
Sorting Layer Default
Order in Layer 1
Gun (Script)
Script Gun
Player Stats
Current Stats CurrentPlayerStatsObject (Player Stats Object)
Assign in Inspector
Projectile Base Projectile
Shot Point Shot Point (Transform)
Projectile Speed 20
Sprites-Default (Material)
Shader Sprites/Default
Edit...

Right click Gun (Script) and hit Edit Script

Switch to Visual Studio Code (Should be installed). This process usually takes a bit

Ask for help if needed

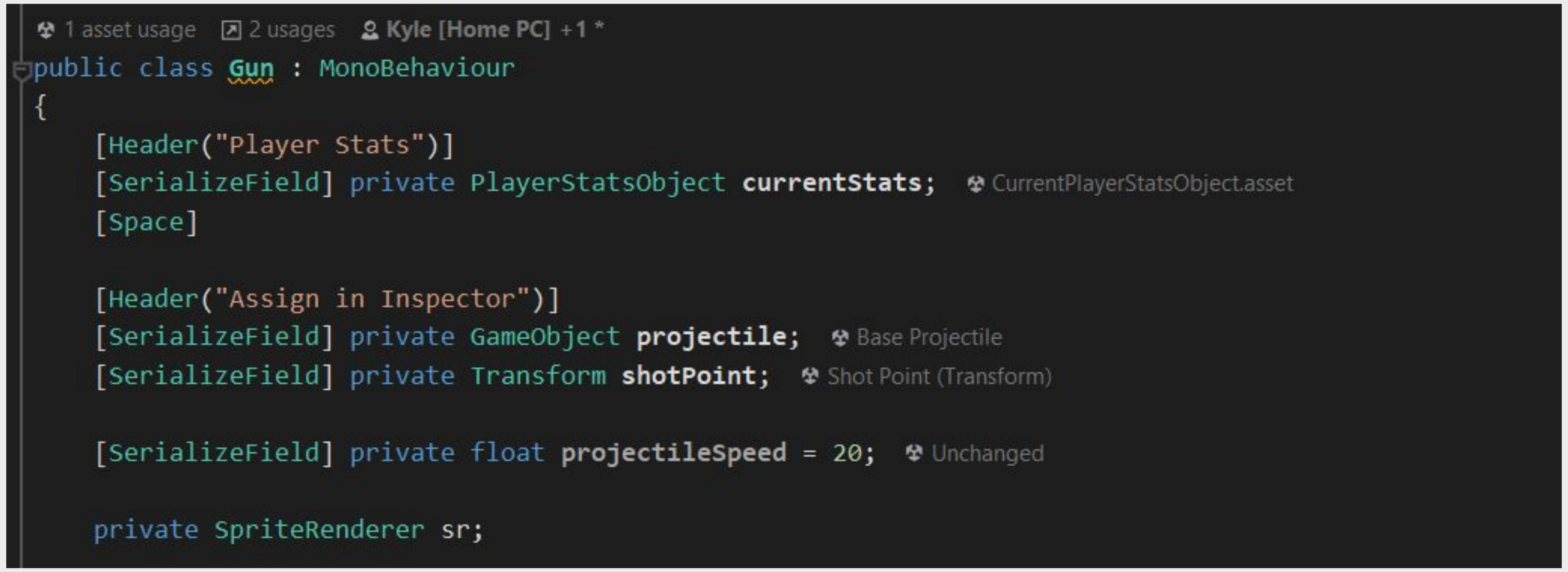

```
1 asset usage   2 usages   Kyle [Home PC] +1 *
public class Gun : MonoBehaviour
{
    [Header("Player Stats")]
    [SerializeField] private PlayerStatsObject currentStats;   CurrentPlayerStatsObject.asset
    [Space]

    [Header("Assign in Inspector")]
    [SerializeField] private GameObject projectile;   Base Projectile
    [SerializeField] private Transform shotPoint;   Shot Point (Transform)

    [SerializeField] private float projectileSpeed = 20;   Unchanged

    private SpriteRenderer sr;

    Event function   KyleHammerGitHub
    private void Start()
    {
        sr = GetComponent<SpriteRenderer>();
    }

    Event function   KyleHammerGitHub +1
    private void Update()
    {
        SetGunDirection();
        SetSpriteDirection();
    }

    1 usage   Kyle [Home PC] +1 *
    public void Shoot()
    {
        GameObject newProjectile = Instantiate(projectile, shotPoint.position, shotPoint.rotation);
        Debug.Log( message: "Bang");

        // TODO: Implement bullet movement

        // Set the projectile velocity

        // Set the projectile damage
    }

    Frequently called   1 usage   Kyle [Home PC] *
    private void SetGunDirection()
    {
        // TODO: Implement gun rotation

        // Find the mouse position on the screen

        // Get the position of the gun

        // Get the direction the bullet needs to go (end point - start point)

        // Set the gun transform's right side to the direction
        // This is because the gun faces right by default

    }
```

Don’t be too overwhelmed, we won’t be going over everything about coding. Just a couple essentials

Variables:

Like a noun. They store a thing…

- projectile is a GameObject that stores our bullet
- currentStats is a PlayerStatsObject that stores the player stats
- projectileSpeed is a float (decimal number) that determines the bullet velocity



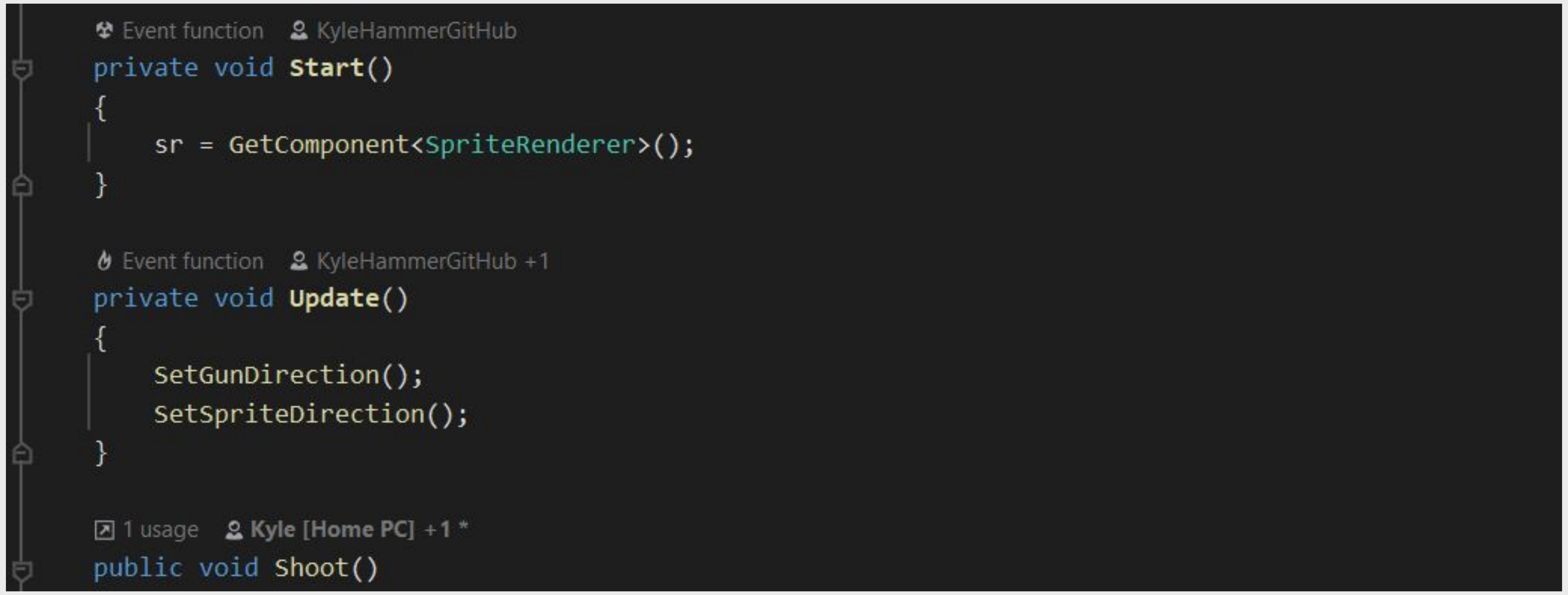
```
1 asset usage   2 usages   Kyle [Home PC] +1 *
public class Gun : MonoBehaviour
{
    [Header("Player Stats")]
    [SerializeField] private PlayerStatsObject currentStats;   CurrentPlayerStatsObject.asset
    [Space]

    [Header("Assign in Inspector")]
    [SerializeField] private GameObject projectile;   Base Projectile
    [SerializeField] private Transform shotPoint;   Shot Point (Transform)

    [SerializeField] private float projectileSpeed = 20;   Unchanged

    private SpriteRenderer sr;
```

It helps to name variables something suitable to their task

Functions:

Like a verb. They are responsible for doing something upon request…

- **Start() is performed on the first frame of the level**
- **Update() is performed every frame**
- **SetGunDirection() is performed every time Update() is run**



```
Event function   KyleHammerGitHub
private void Start()
{
    sr = GetComponent<SpriteRenderer>();
}

Event function   KyleHammerGitHub +1
private void Update()
{
    SetGunDirection();
    SetSpriteDirection();
}

1 usage   Kyle [Home PC] +1 *
public void Shoot()
```

When might we want to use Start() and Update() in a game?

Modify the Shoot() function

```
public void Shoot()
{
        if (!canShoot) return;

        canShoot = false;
        currentCooldown = fireRateCooldown;

        GameObject newProjectile = Instantiate(projectile, shotPoint.position, shotPoint.rotation);

        // TODO: Implement bullet movement

        // Set the projectile velocity
        newProjectile.GetComponent<Rigidbody2D>().velocity = transform.right * projectileSpeed;

        // Set the projectile damage
        newProjectile.GetComponent<Projectile>().SetDamage(currentStats.damage);
}
```

//
Is a comment. It is simply text used to help the programmer understand what the task is. It does not run

Insert the blue lines into your Gun script

Modify SetGunDirection()

```
private void SetGunDirection()
{
        // TODO: Implement gun rotation

        // Find the mouse position on the screen
        Vector2 mousePosition = Camera.main.ScreenToWorldPoint(Mouse.current.position.ReadValue());

        // Get the position of the gun
        Vector2 weaponPosition = transform.position;

        // Get the direction the bullet needs to go (end point - start point)
        Vector2 direction = mousePosition - weaponPosition;

        // Set the gun transform's right side to the direction
        // This is because the gun faces right by default
        transform.right = direction;
}
```

Coding can be difficult, especially with such little time. So don’t feel too bad if you don’t understand what is going on and directly copy instead

Test it out in play mode. Did it work?

Don’t feel discouraged if it doesn’t

Most programmers never get it right on the first go anyway!

Operator Cheat Sheet

```
; // The "Full Stop" of coding, ends every command* *but not comparison
a = b; // Set the value of a to the value of b
b = a; // Set the value of b to the value of a
```

```
If (Criteria is met)
{
        Do this task;
}
```

```
a++; // Make a equal it's value + 1
b--; // Make b equal it's value – 1
a += 10 // Make a equal it's value + 10
a = b + 10; // Make a equal b + 10
```

```
If (a > b) {} // Compare if the value of a is larger than b
If (a == b) {} // Compare if the value of a is the same value as b
If (a >= b) {} // Compare if the value of a larger than or is same value as b
```

```
// Compare if a is the same value as b AND b is larger than c
If (a == b && b > c) {}
```

```
// Compare if a is the same value as b OR b is larger than c
If (a == b || b > c) {} // | is the symbol above the enter key while holding shift
```

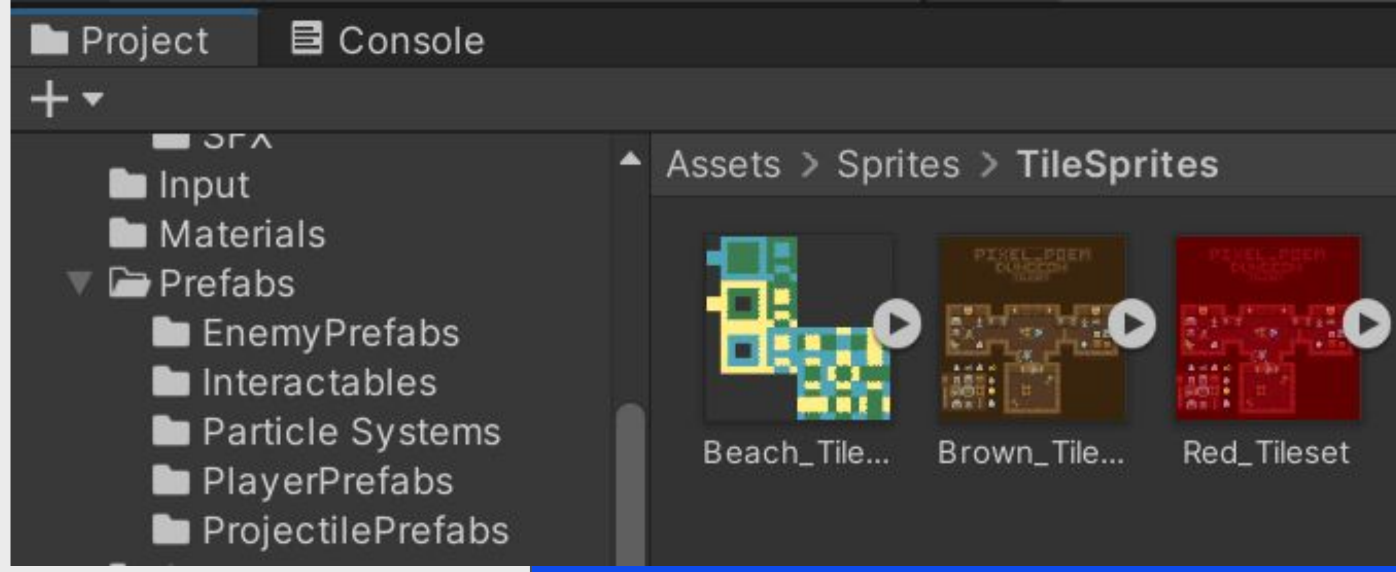
How does && and || differ from each other?

Let's move onto something more fun, level designing!

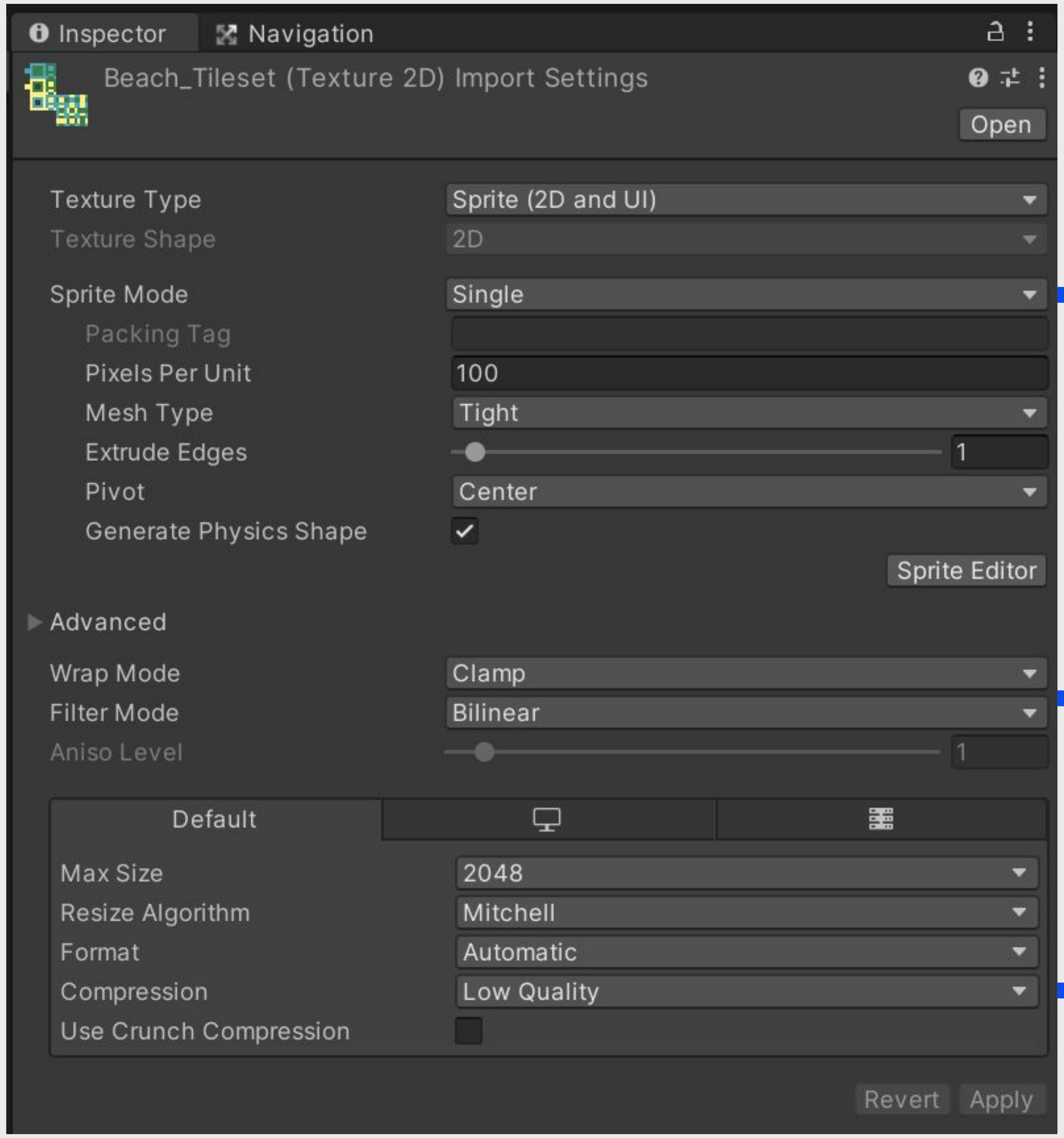
Task Four – Level Designing

We need tiles to paint with, so lets look into splitting a sprite sheet

Importing Textures



Locate the Beach_Tileset in the sprites folder, select it and make the following changes



Inspector | Navigation

Beach_Tileset (Texture 2D) Import Settings

Open

Texture Type	Sprite (2D and UI)
Texture Shape	2D
Sprite Mode	Single
Packing Tag	
Pixels Per Unit	100
Mesh Type	Tight
Extrude Edges	1
Pivot	Center
Generate Physics Shape	✓

Sprite Editor

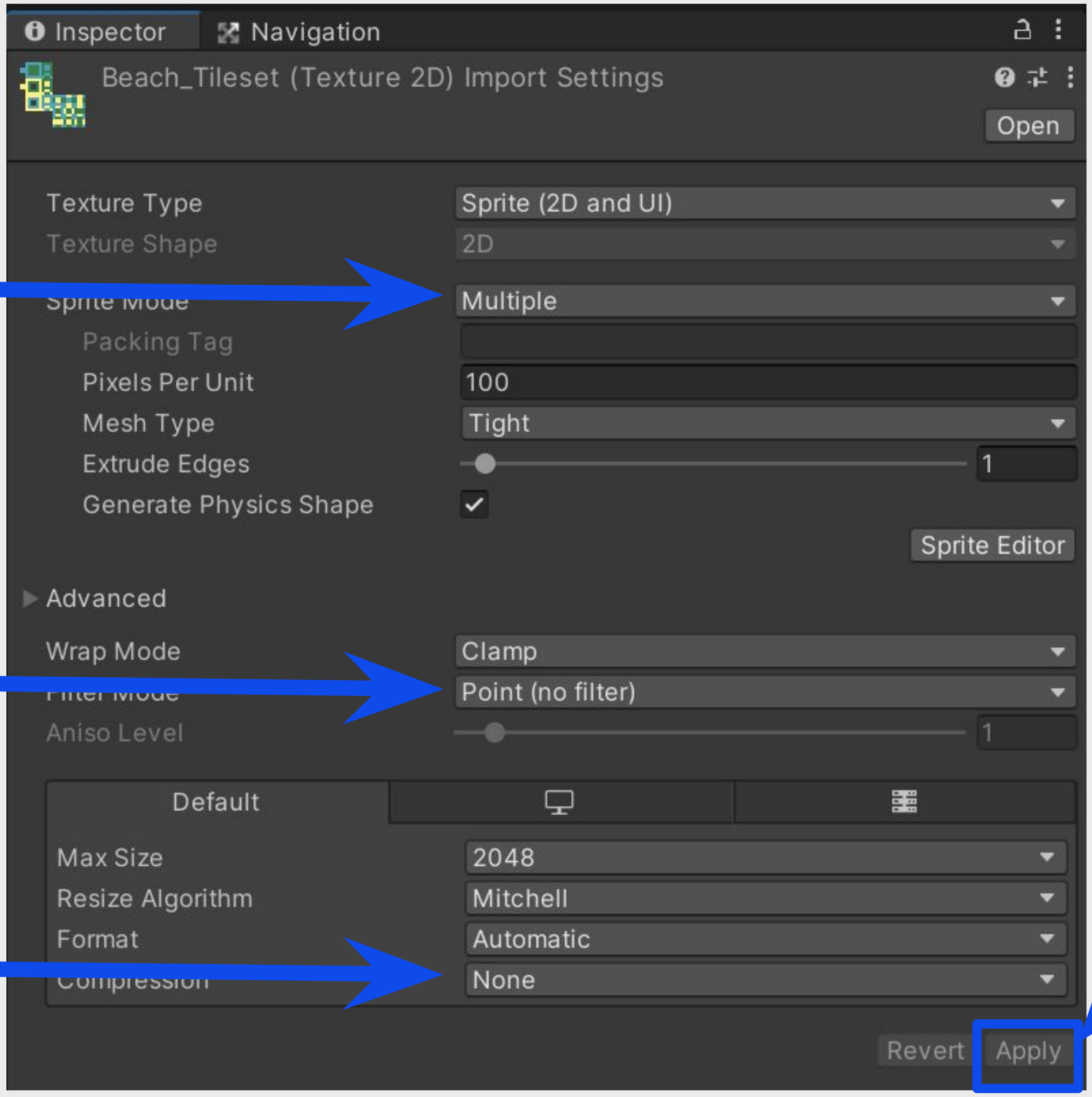
Advanced

Wrap Mode	Clamp
Filter Mode	Bilinear
Aniso Level	1

Default

Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	Low Quality
Use Crunch Compression	

Revert Apply



Inspector | Navigation

Beach_Tileset (Texture 2D) Import Settings

Open

Texture Type	Sprite (2D and UI)
Texture Shape	2D
Sprite Mode	Multiple
Packing Tag	
Pixels Per Unit	100
Mesh Type	Tight
Extrude Edges	1
Generate Physics Shape	✓

Sprite Editor

Advanced

Wrap Mode	Clamp
Filter Mode	Point (no filter)
Aniso Level	1

Default

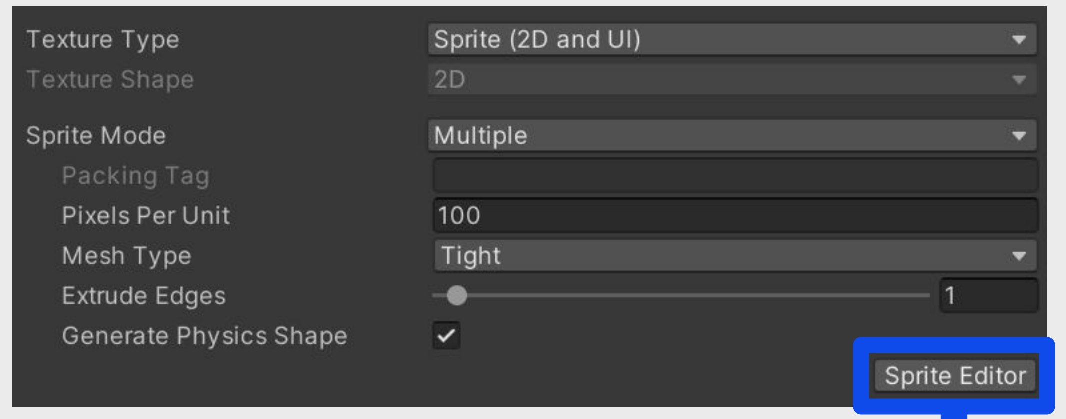
Max Size	2048
Resize Algorithm	Mitchell
Format	Automatic
Compression	None

Revert Apply

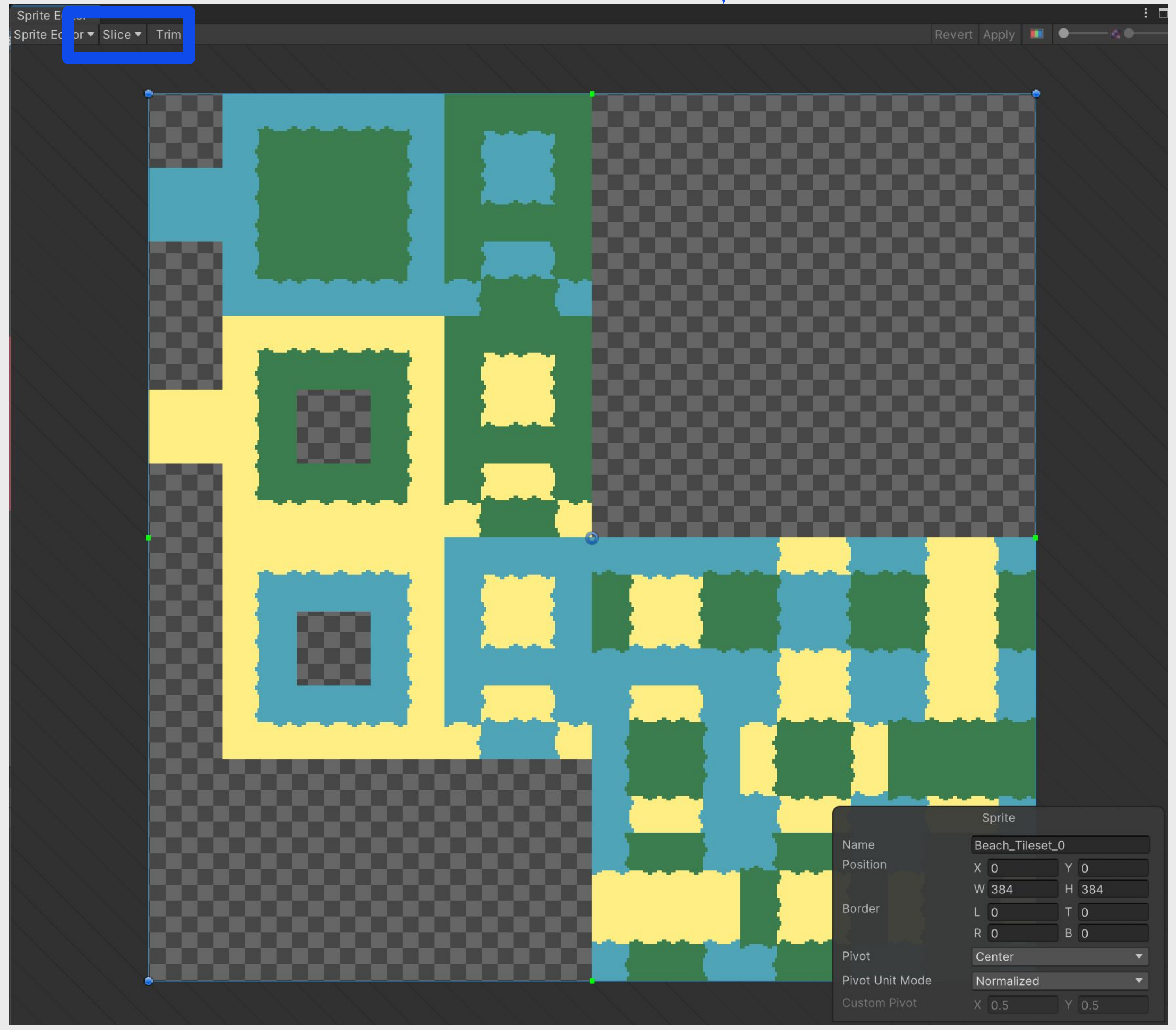
Hit apply when you're done!

Slicing Textures

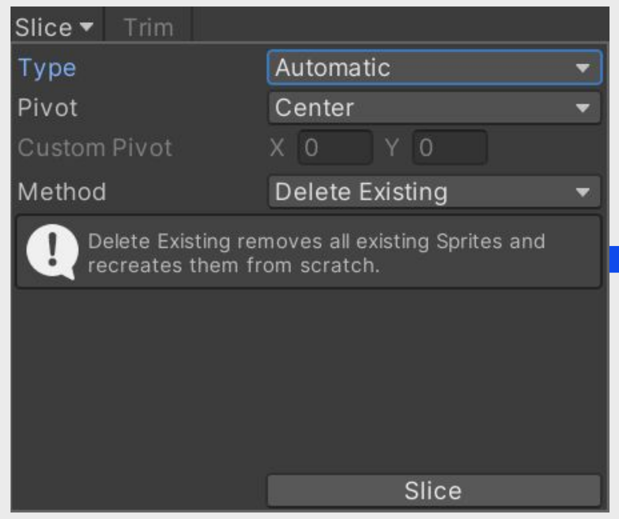
Next select the sprite editor so we can slice up the image



Texture Type: Sprite (2D and UI)
Texture Shape: 2D
Sprite Mode: Multiple
Packing Tag:
Pixels Per Unit: 100
Mesh Type: Tight
Extrude Edges: 1
Generate Physics Shape: ✓

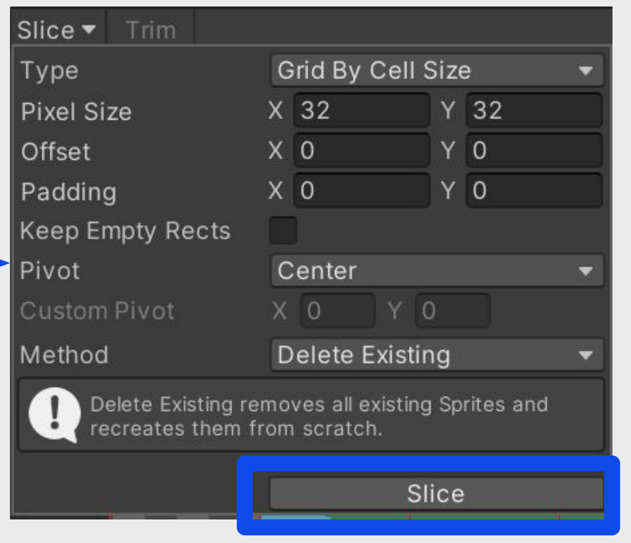


Sprite
Name: Beach_Tileset_0
Position: X 0 Y 0 W 384 H 384
Border: L 0 T 0 R 0 B 0
Pivot: Center
Pivot Unit Mode: Normalized
Custom Pivot: X 0.5 Y 0.5

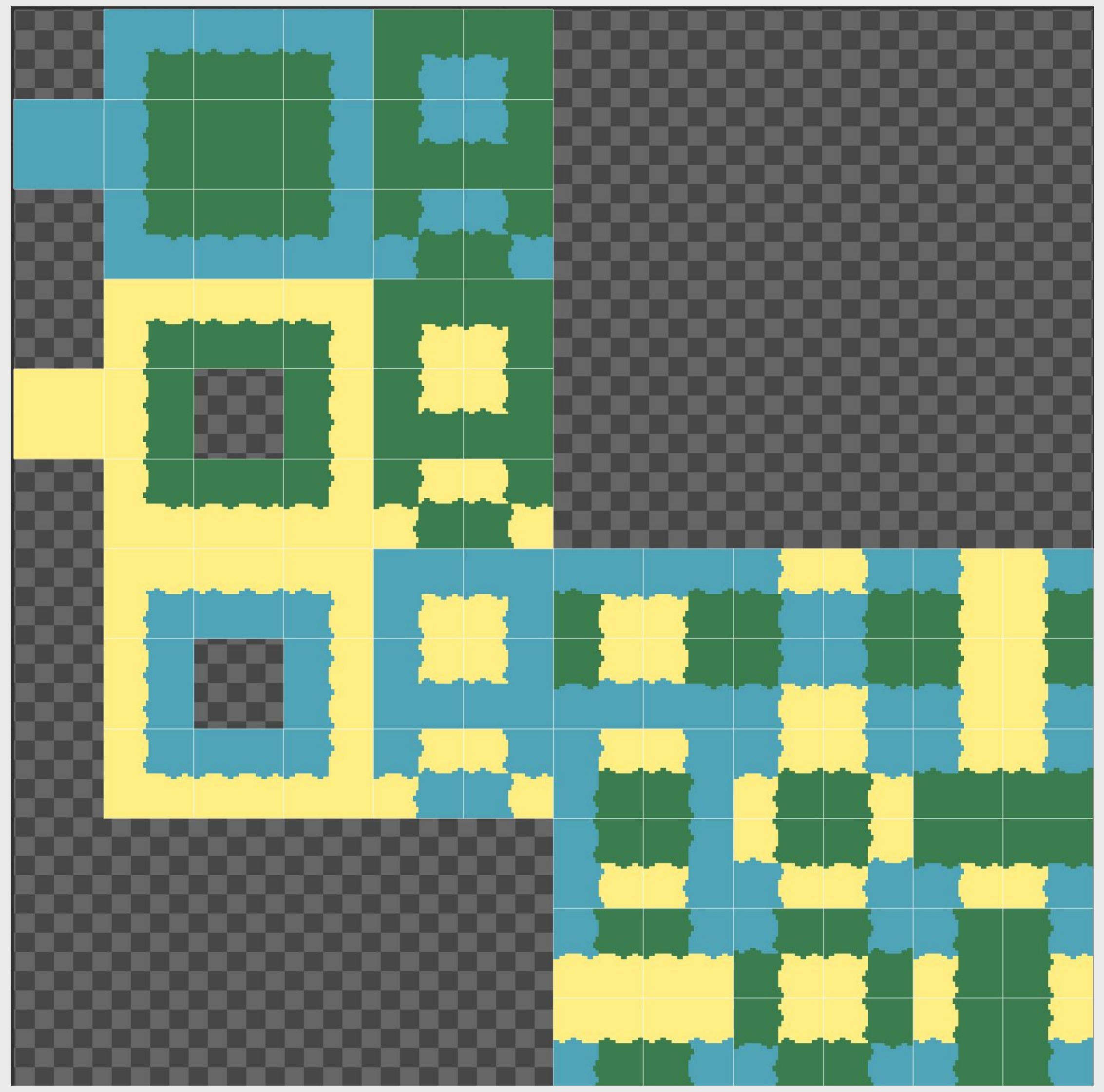


Type: Automatic
Pivot: Center
Custom Pivot: X 0 Y 0
Method: Delete Existing
Delete Existing removes all existing Sprites and recreates them from scratch.

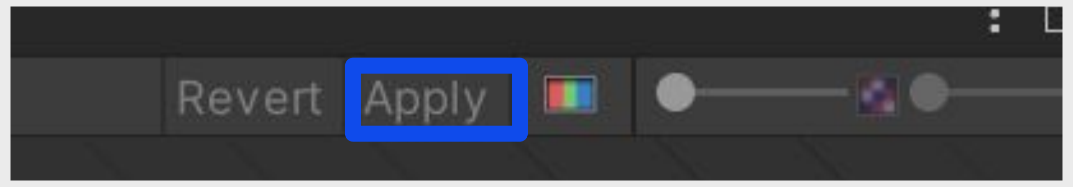
Change to Grid by Cell Size. Each cell/tile is 32px by 32px in this image



Type: Grid By Cell Size
Pixel Size: X 32 Y 32
Offset: X 0 Y 0
Padding: X 0 Y 0
Keep Empty Rects:
Pivot: Center
Custom Pivot: X 0 Y 0
Method: Delete Existing
Delete Existing removes all existing Sprites and recreates them from scratch.



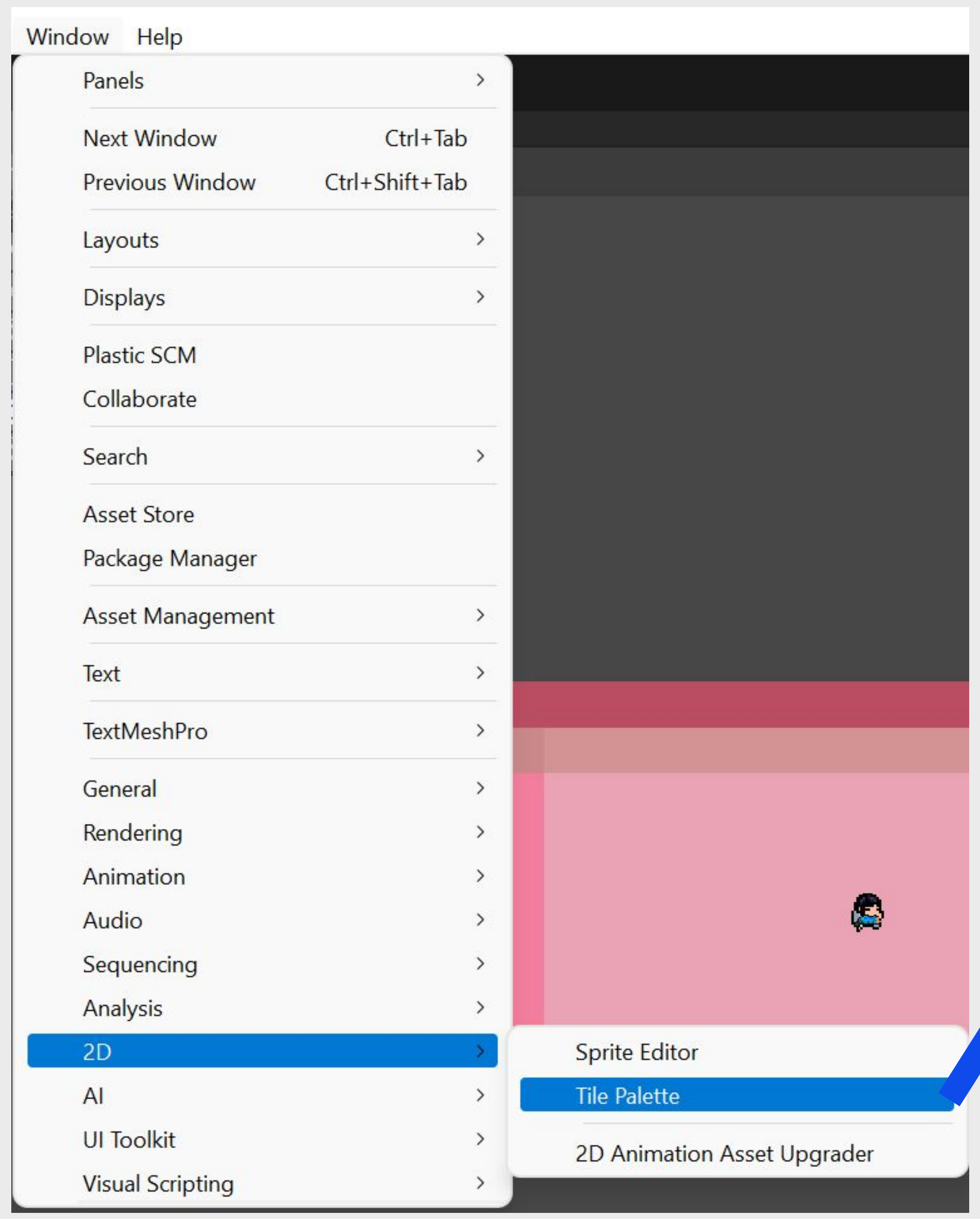
The result

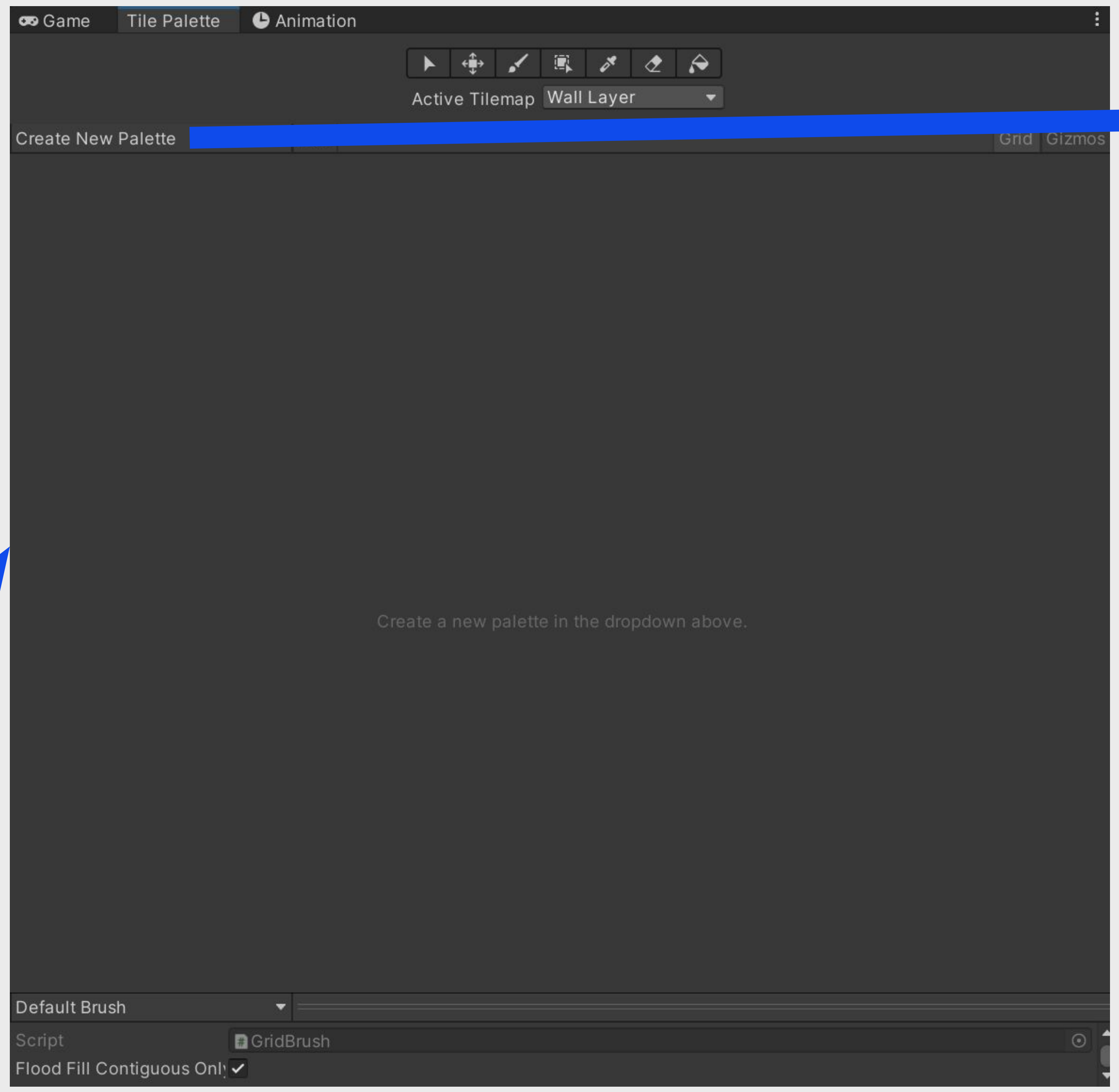


Hit apply when you're done!

Making a Palette

Let turn these textures into a palette we can paint levels with





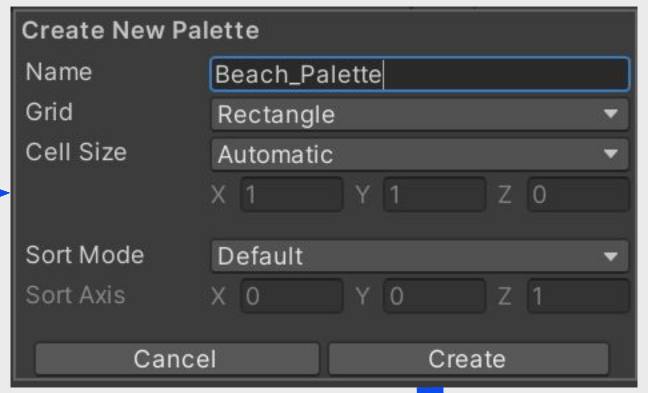
Active Tilemap: Wall Layer

Create a new palette in the dropdown above.

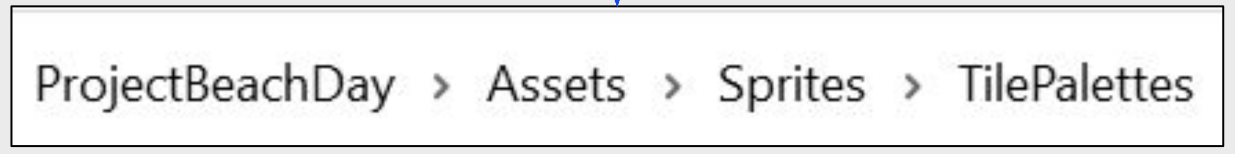
Default Brush

Script: GridBrush

Flood Fill Contiguous Only ✓


Create New Palette

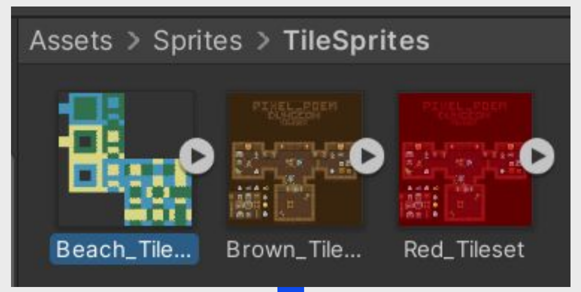
Name	Beach_Palette
Grid	Rectangle
Cell Size	Automatic
	X 1 Y 1 Z 0
Sort Mode	Default
Sort Axis	X 0 Y 0 Z 1


ProjectBeachDay › Assets › Sprites › TilePalettes

Save it in this location

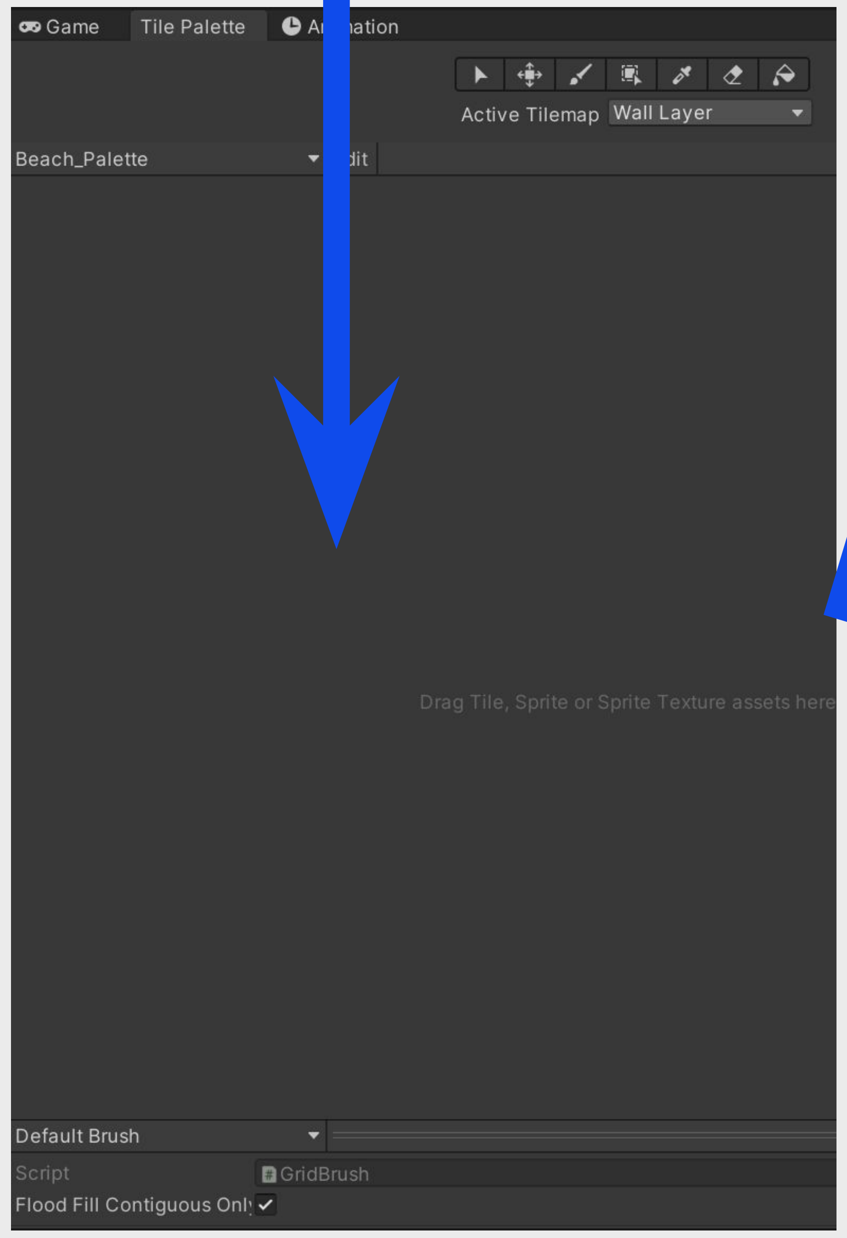
Making a Palette

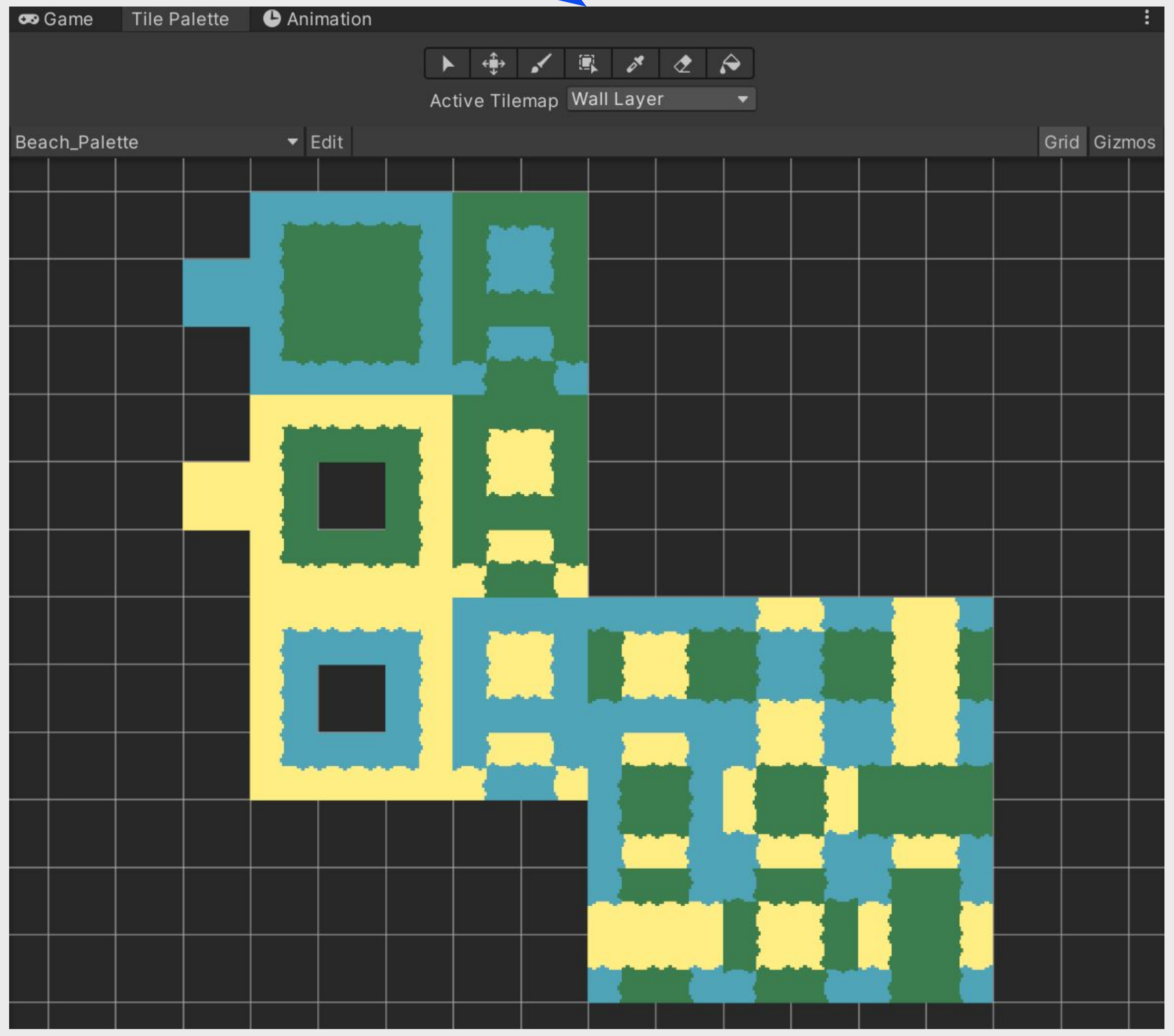
Drag and drop our textures in the project window into the palette window



ProjectBeachDay › Assets › Sprites › TileAssets › Beach_Tiles

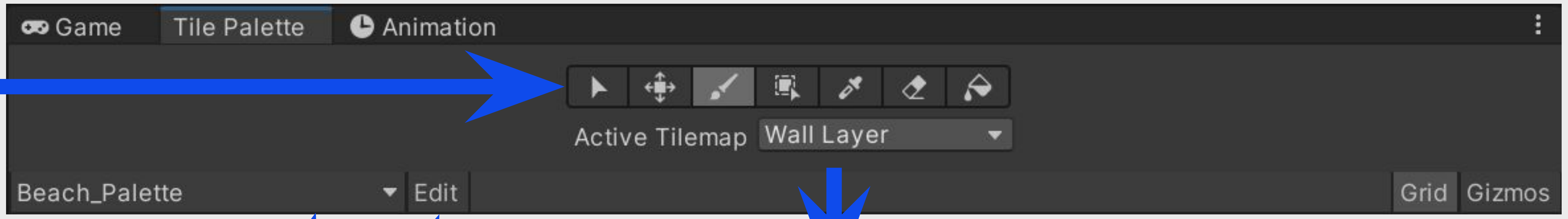
A tile will be generated for each texture. Create a folder called "Beach_Tiles" and save it there





This is what the end result should look like!

Palette painting tips

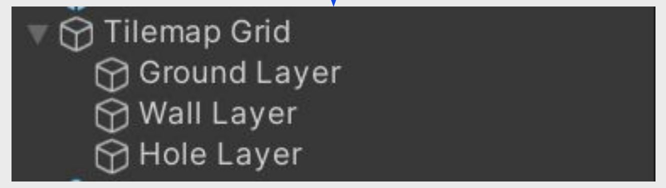


These tools work similarly to other painting programs. Hover over them to see what they do

Colour picker, brush and eraser are probably the most useful

Change the palette your using if you have multiple palettes

Used to "Edit" the tile palette. Useful for making changes to the palette like removing or moving tiles within the palette.



Layers in the hierarchy

Layers!

It's important to make sure you are painting on the correct layer (that way you're not painting walls on the floor layer for example)

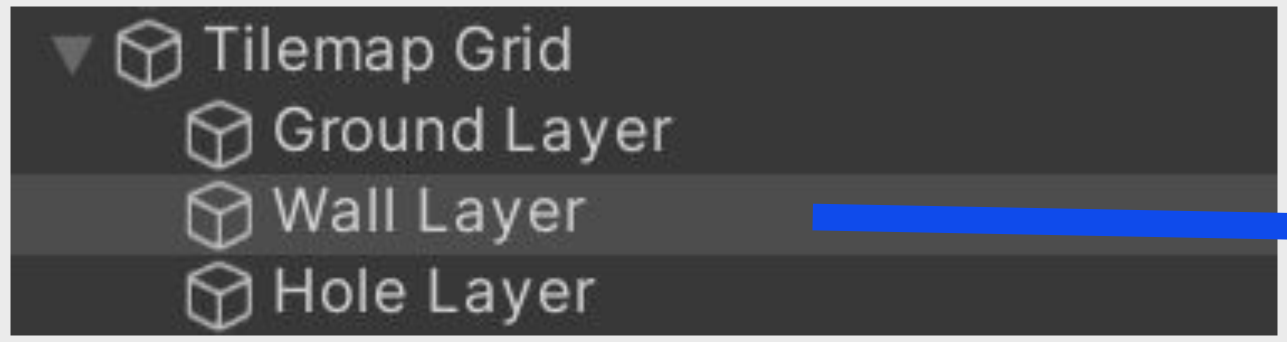
We currently have 3 layers specified

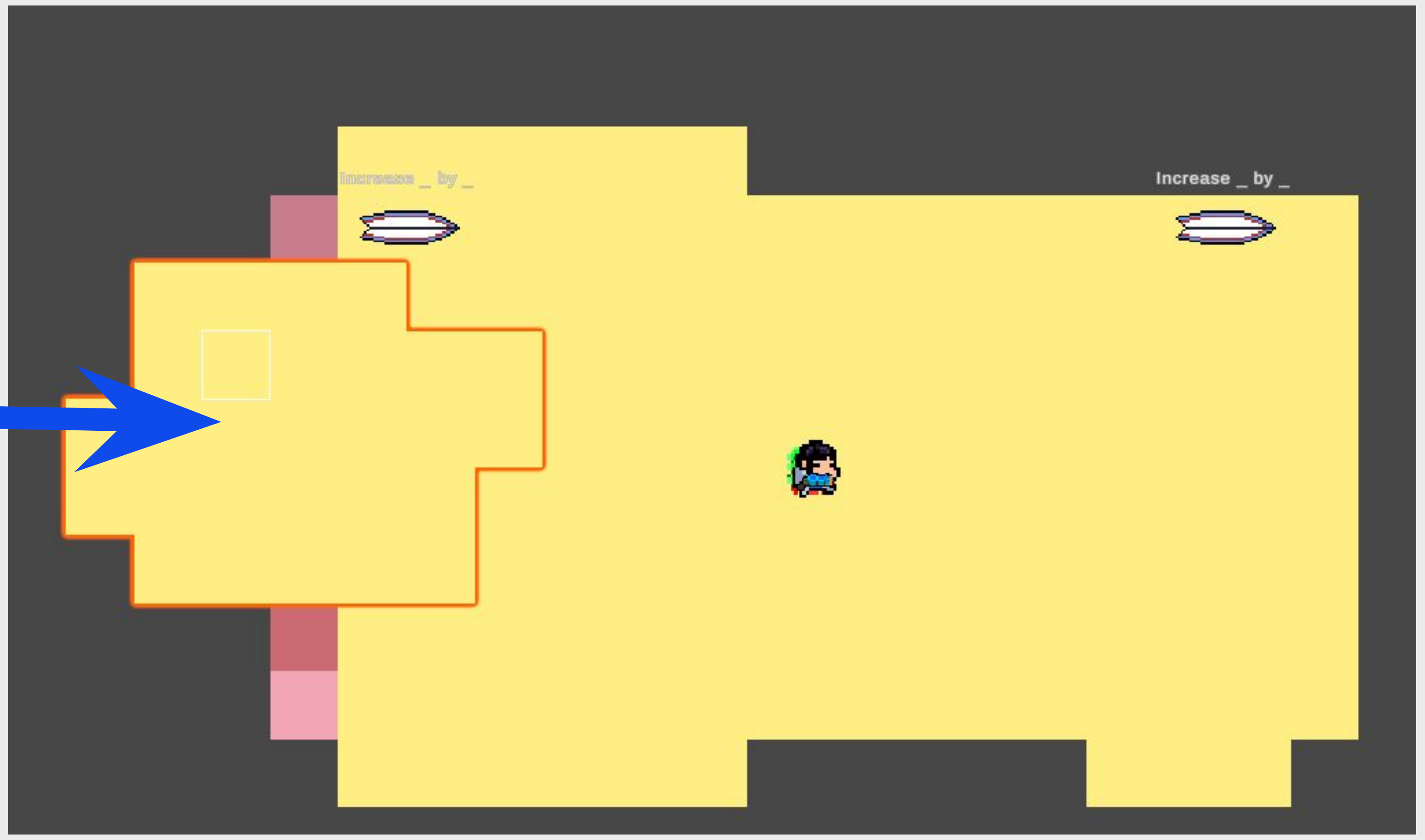
Ground layer: Area where the player and enemies can walk

Wall layer: Area that blocks the player, enemies and projectiles

Hole layer: Area that blocks the player, enemies but not projectiles (think like a hole in the ground)

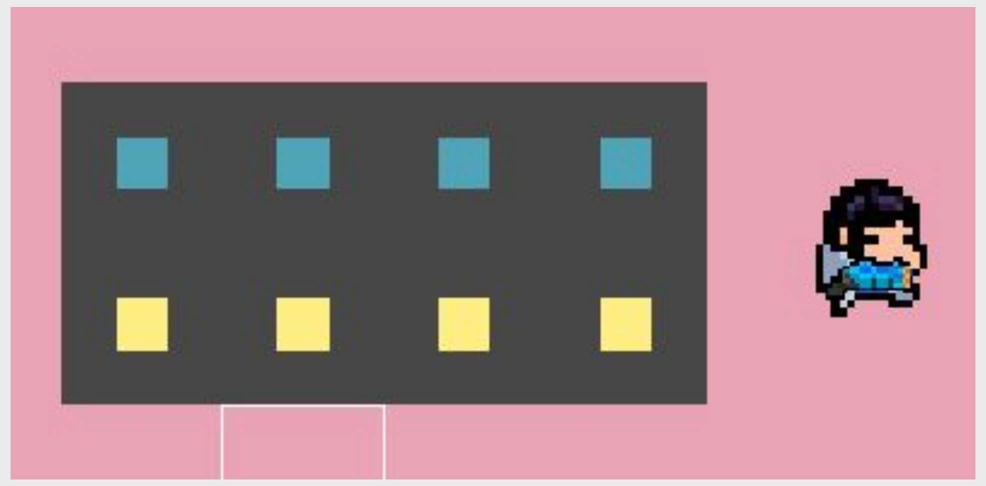
You can select the layer in the hierarchy window to see what tiles are on what layer

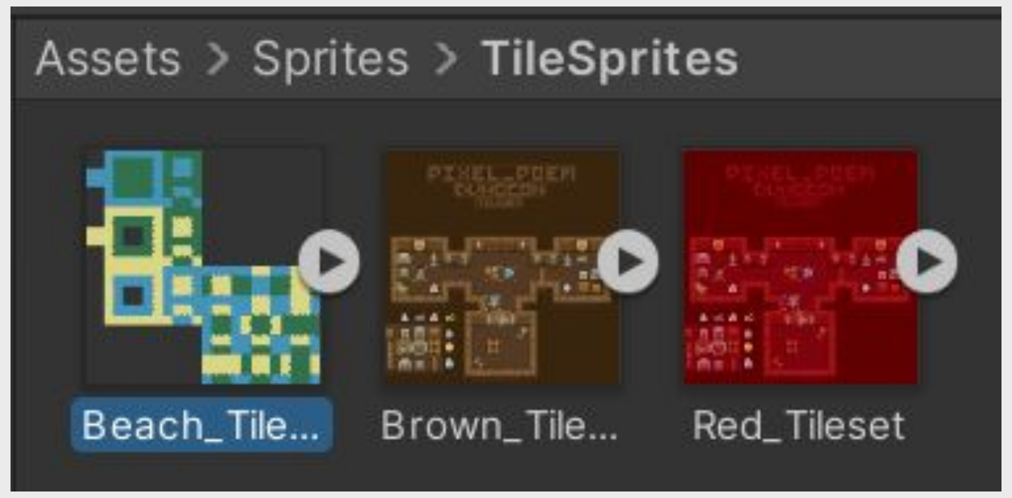




Wrong sized tiles?

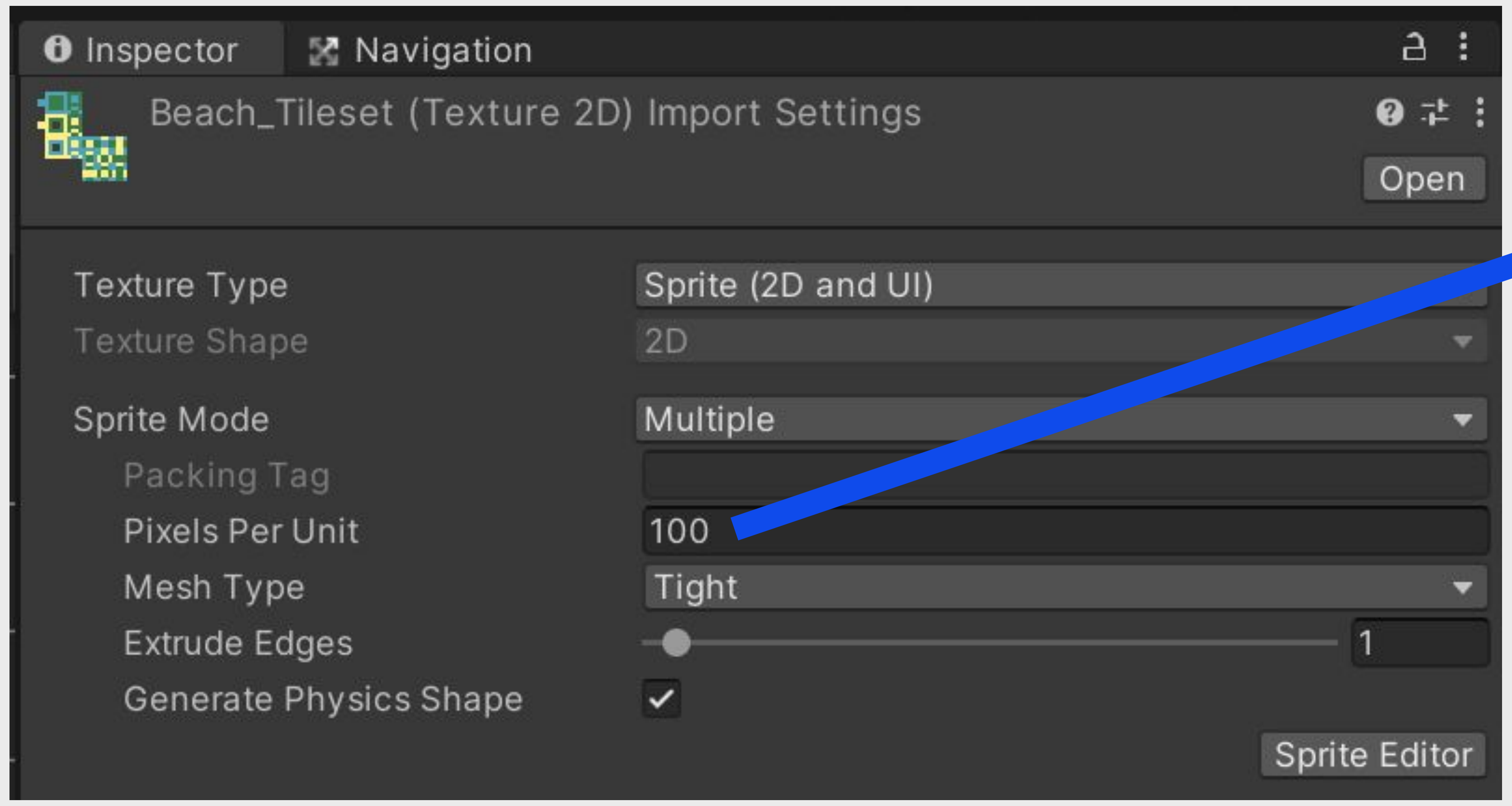
Our tiles too small. Let's fix that





Assets > Sprites > TileSprites

Beach_Tile... Brown_Tile... Red_Tileset



Inspector | Navigation

Beach_Tileset (Texture 2D) Import Settings

Open

```
Texture Type              Sprite (2D and UI)
Texture Shape             2D
Sprite Mode               Multiple
  Packing Tag
  Pixels Per Unit         100
  Mesh Type               Tight
  Extrude Edges           1
  Generate Physics Shape  ✓
```

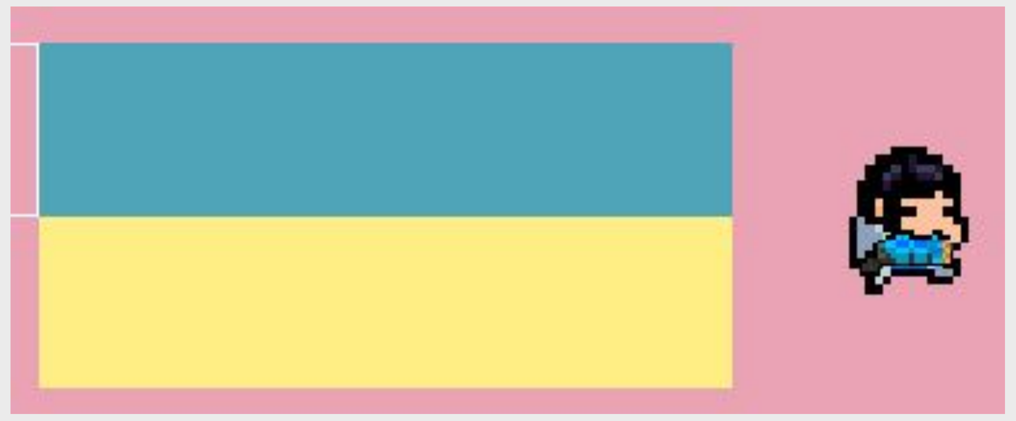
Sprite Editor

Change to 32

Don't forget to hit apply!

Project Window

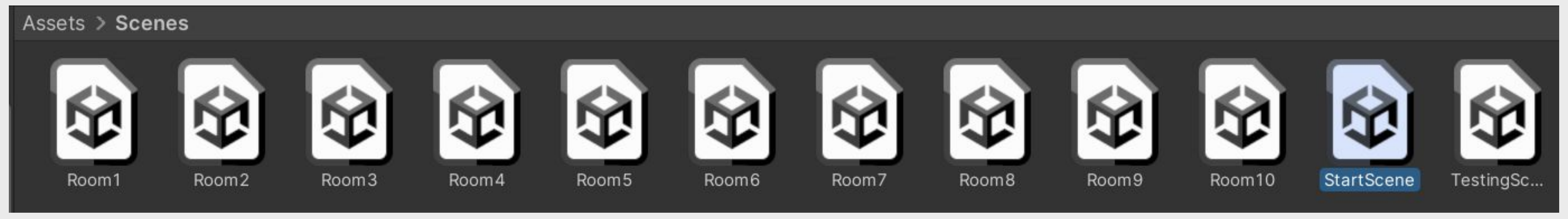
Inspector Window



Full tiles!

Let’s start building some levels now!

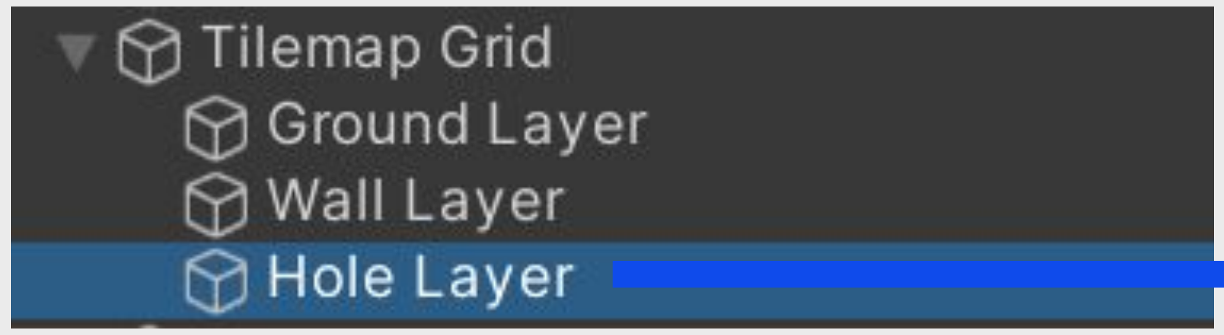
Go to your scenes folder and double click "StartScene"

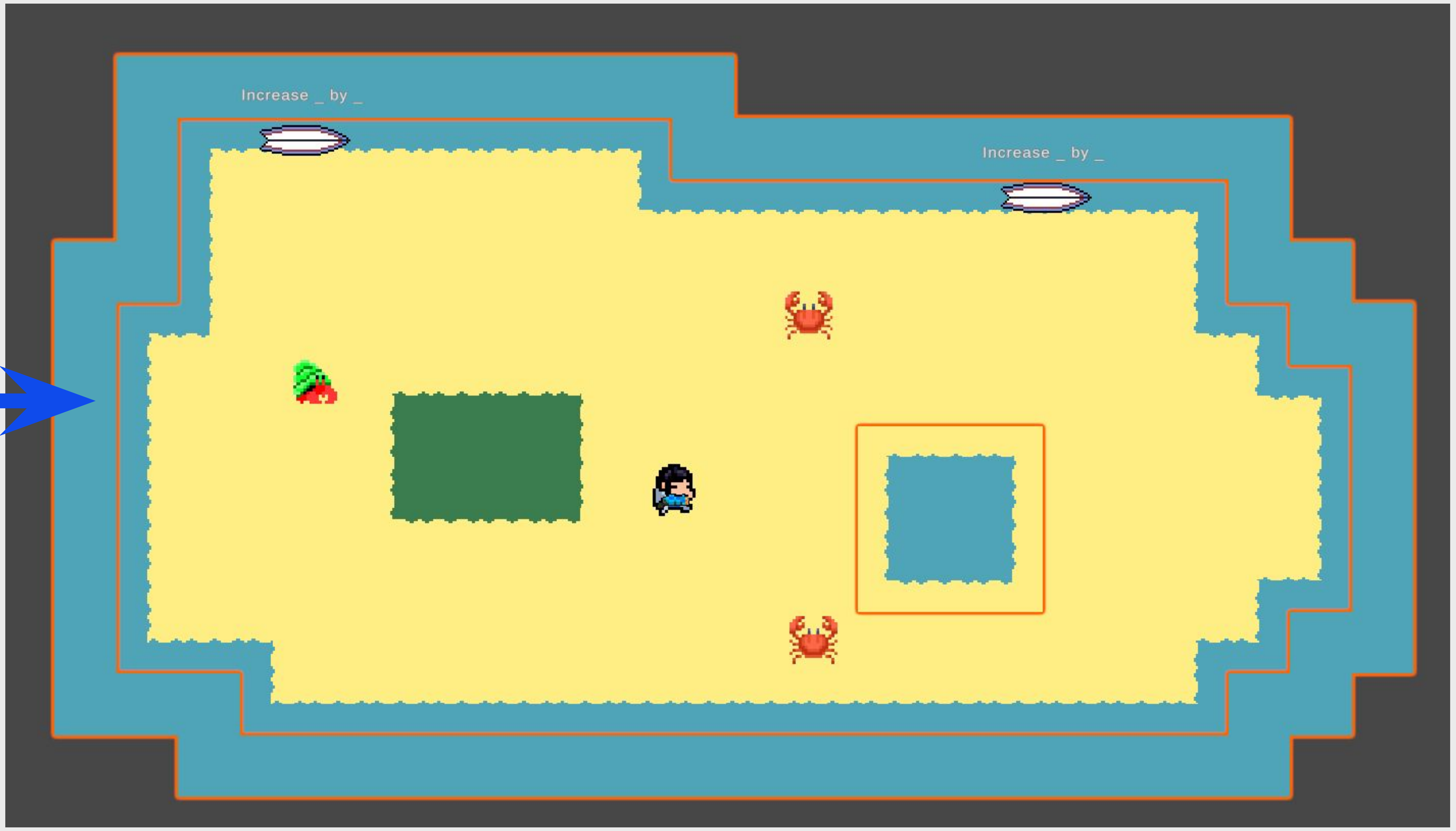


Design the game's first level!

Make it fairly easy for the player, it is the first level after all. Make sure to place walls/holes.

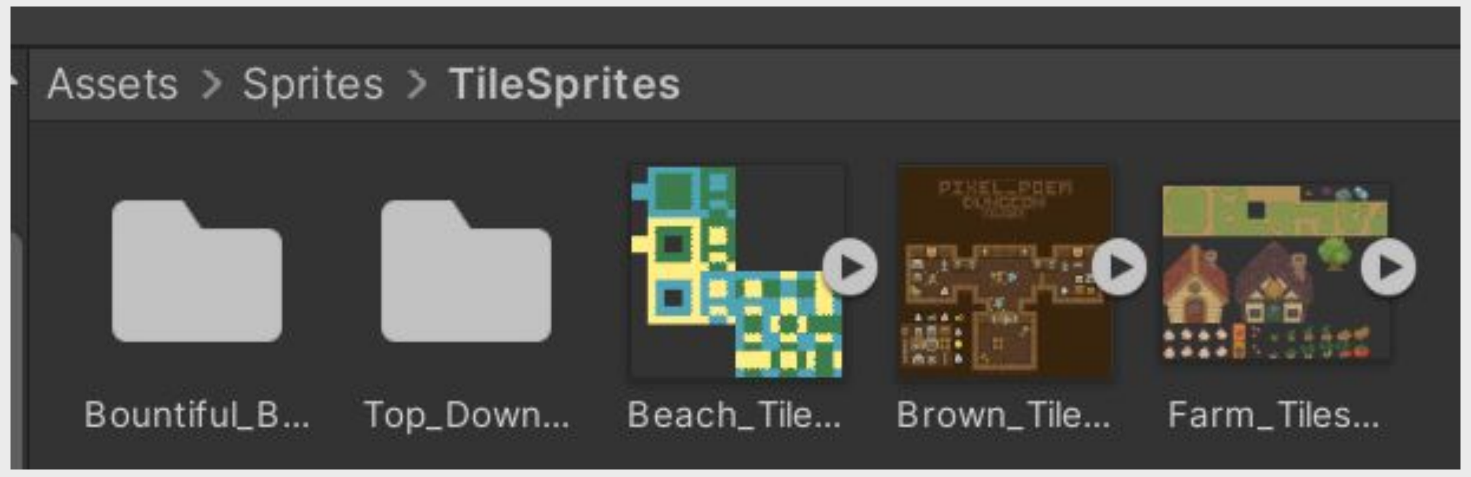
Don't forget to place enemies.



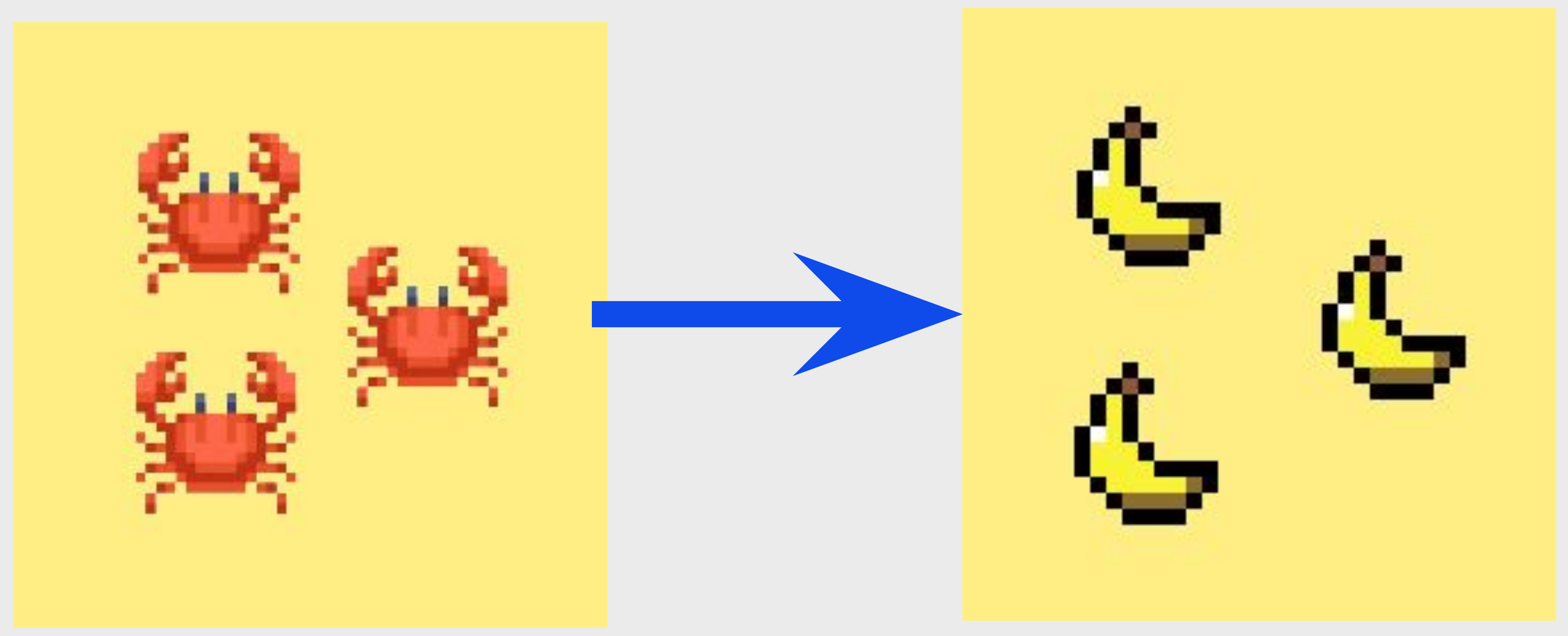


There's more tiles!

Need more tile variety? There's more sprite sheets available in the project

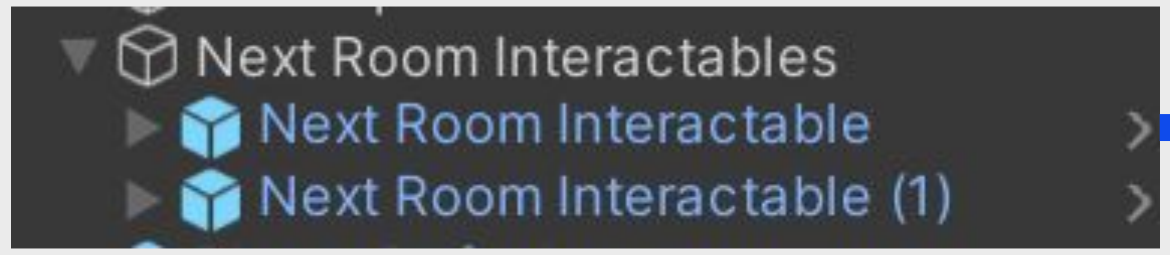


Or download your own sprite sheets online and add them to the project (simply copy the files into the project)!



You can also import new sprites too, if you're looking for more enemy textures

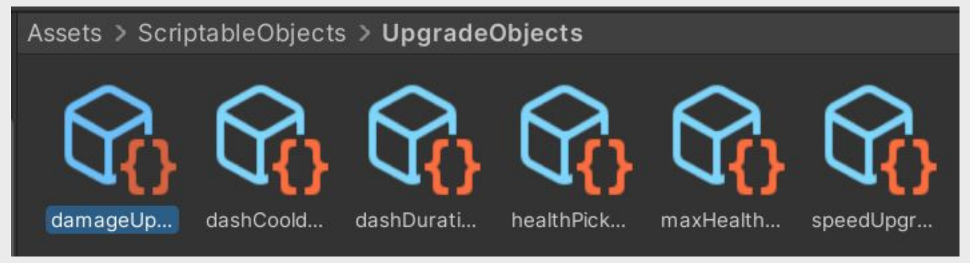
Remember the progression mechanics?


Next Room Interactables
Next Room Interactable
Next Room Interactable (1)

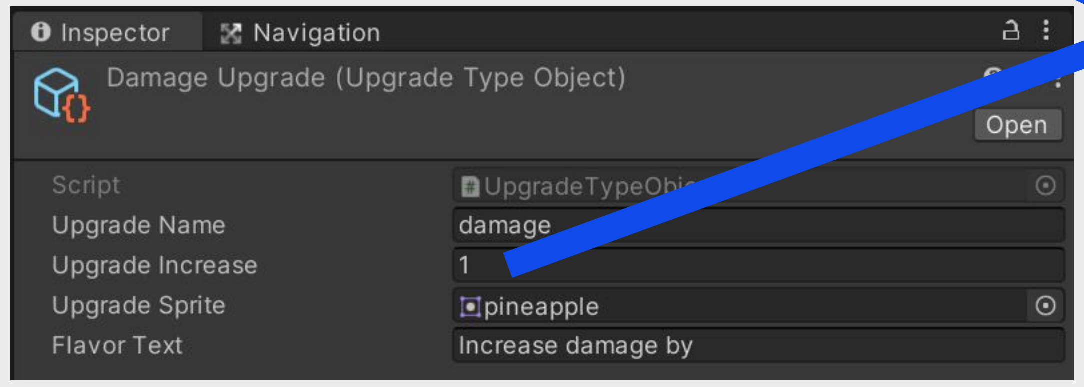
Increase
damage by 1

Each surfboard (Next Room Interactable) gives the player access to new levels. Feel free to move their spawn position or add even more Next Room Interactables if you wish

Room Rewards


Assets > ScriptableObjects > UpgradeObjects

damageUp... dashCoold... dashDurati... healthPick... maxHealth... speedUpgr...


Inspector | Navigation

Damage Upgrade (Upgrade Type Object)

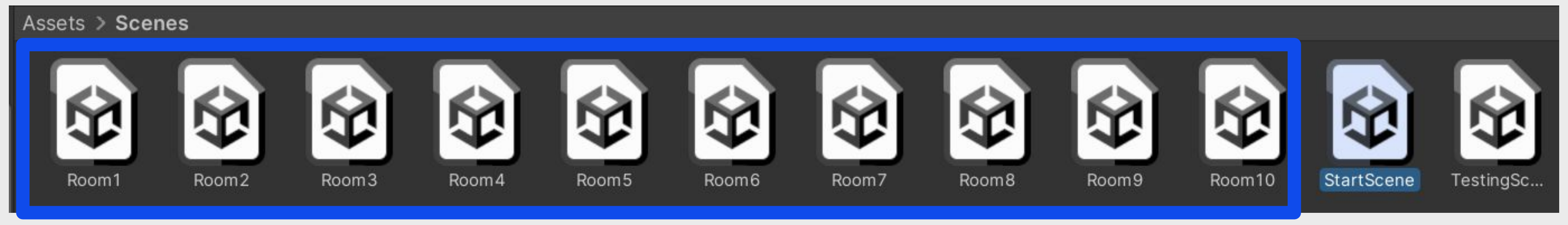
Open

Script	UpgradeTypeObj
Upgrade Name	damage
Upgrade Increase	1
Upgrade Sprite	pineapple
Flavor Text	Increase damage by

Feel free to modify the existing room reward increase

Adding more room rewards is a bit more tricky, so feel free to ask for help

Once you're done, move onto designing more rooms!

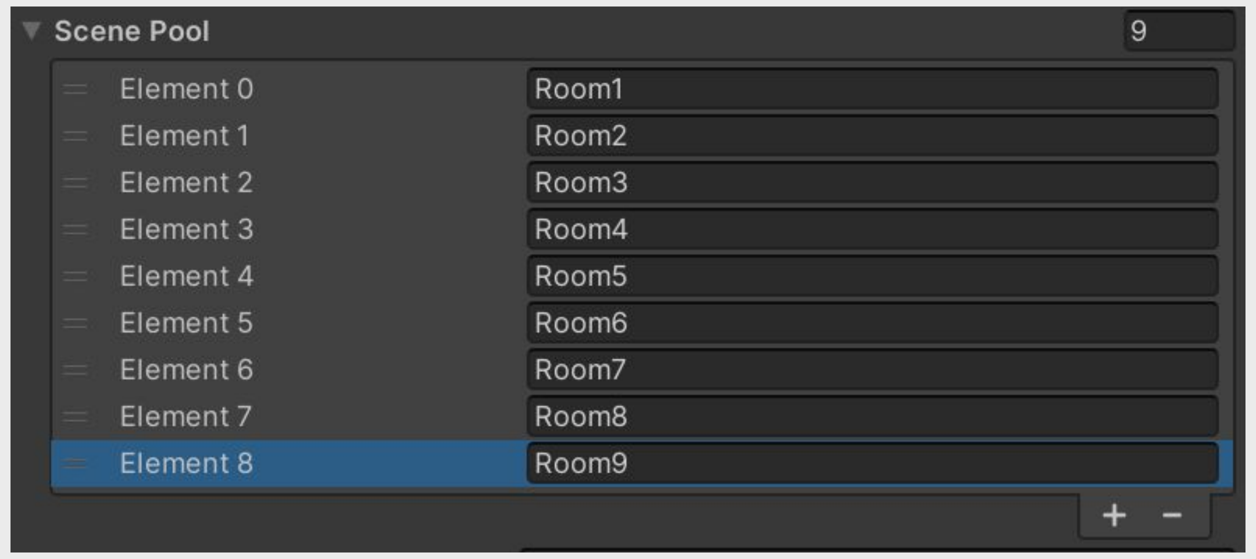

Assets > Scenes

Room1 Room2 Room3 Room4 Room5 Room6 Room7 Room8 Room9 Room10 StartScene TestingSc...

Too many rooms?

Select the room to be deleted and hit the delete key. A confirmation prompt will also appear

In the inspector, select the scene you removed and hit the – (minus) button



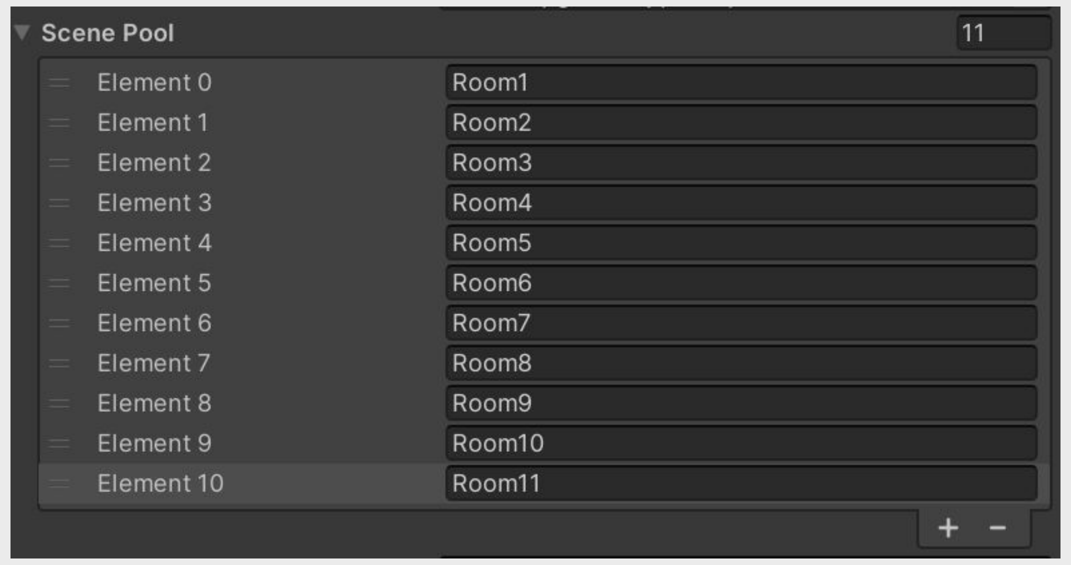
Scene Pool 9

Element 0	Room1
Element 1	Room2
Element 2	Room3
Element 3	Room4
Element 4	Room5
Element 5	Room6
Element 6	Room7
Element 7	Room8
Element 8	Room9

Not enough rooms?

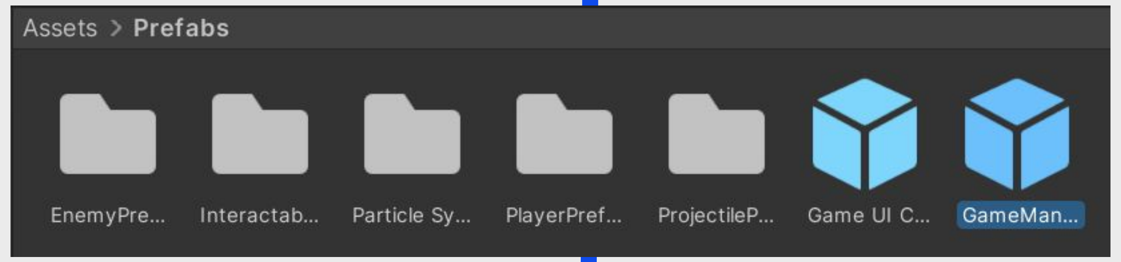
Copy paste a scene and give it an appropriate name. Then modify that scene to your liking

In the inspector, hit the + and type in the name of your newly added scene



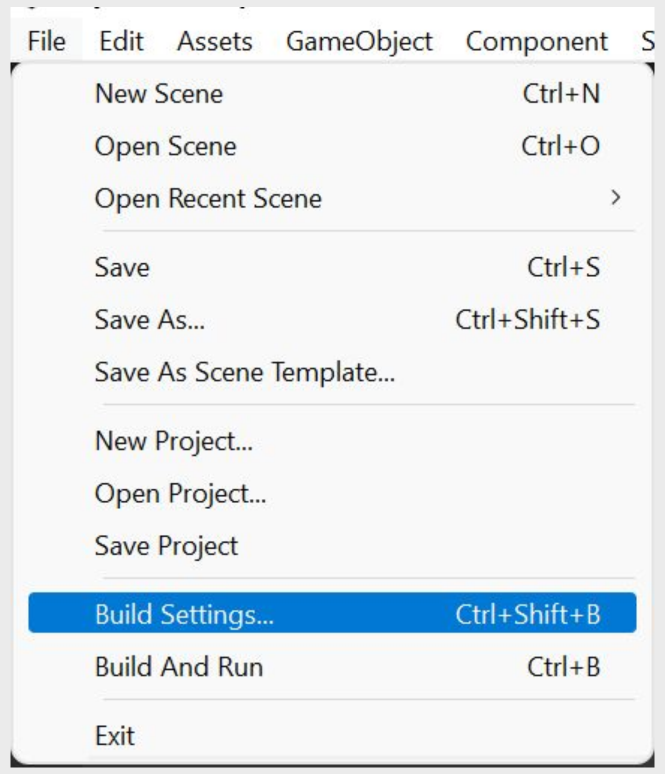
Scene Pool 11

Element 0	Room1
Element 1	Room2
Element 2	Room3
Element 3	Room4
Element 4	Room5
Element 5	Room6
Element 6	Room7
Element 7	Room8
Element 8	Room9
Element 9	Room10
Element 10	Room11

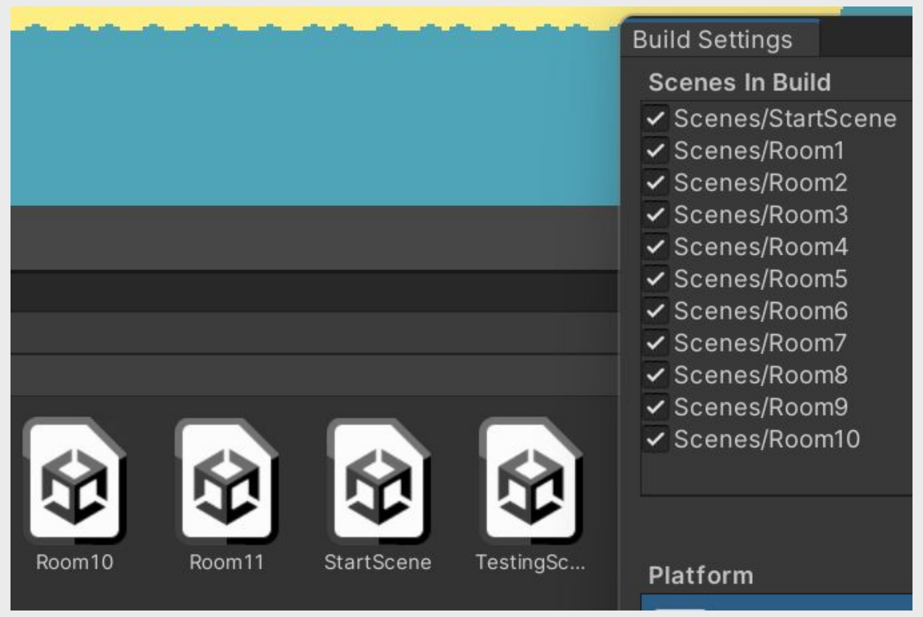


Modify the game manager prefab

After doing the above, go to File > Build Settings



If you want to remove scenes, right click the scene to be removed in the build settings and "Remove Selection"



Build Settings

Scenes In Build

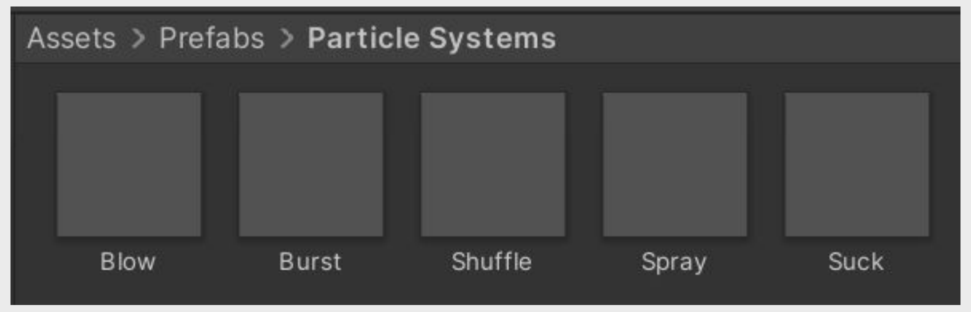
- ✓ Scenes/StartScene
- ✓ Scenes/Room1
- ✓ Scenes/Room2
- ✓ Scenes/Room3
- ✓ Scenes/Room4
- ✓ Scenes/Room5
- ✓ Scenes/Room6
- ✓ Scenes/Room7
- ✓ Scenes/Room8
- ✓ Scenes/Room9
- ✓ Scenes/Room10

If you have new scenes, click and drag them from the project window to the "Scenes In Build"

Challenges and Extras

Particle Systems

Really make your effects shine with particles (Some programming may be involved)

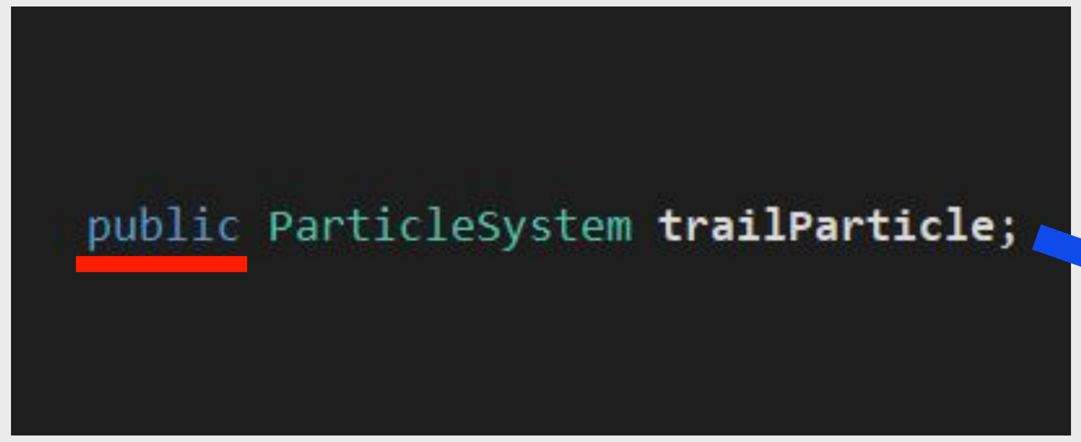

Assets > Prefabs > **Particle Systems**

Shuffle is attached to the GameObject projectile. It can be added to anything that moves

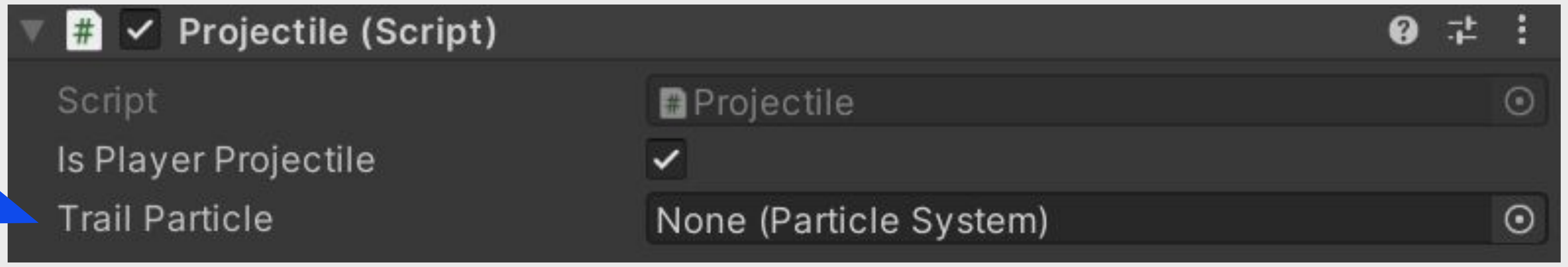
Spray and Suck are constantly running. They can be turned on and off in code

Blow and Burst fire once, then need to be stopped. This can be done in code

In Projectile (Script)



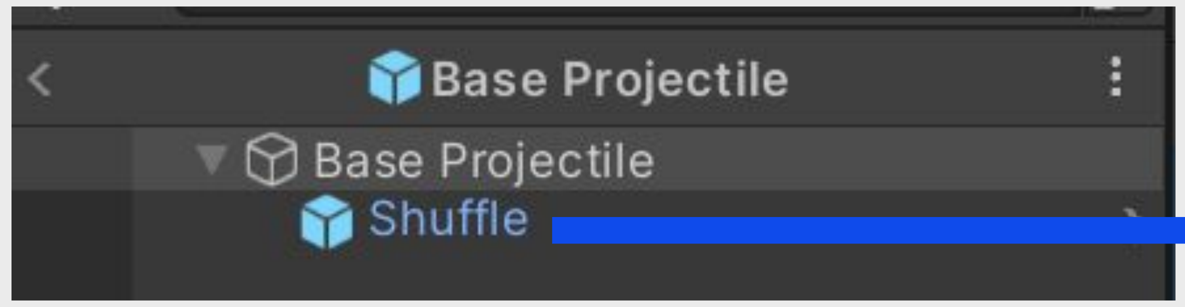
```
public ParticleSystem trailParticle;
```



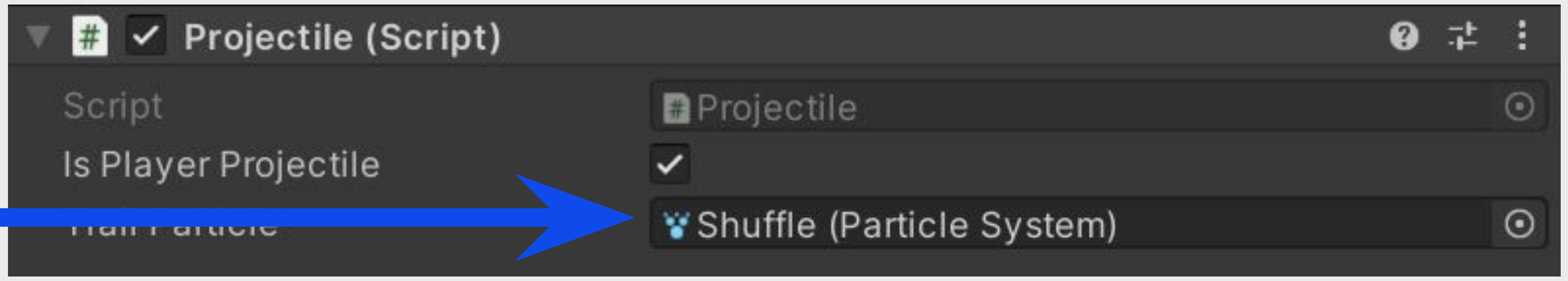
Projectile (Script)

Script	Projectile
Is Player Projectile	✓
Trail Particle	None (Particle System)

Add the effect to the Prefab



Click and drag the effect into the inspector



Projectile (Script)

Script	Projectile
Is Player Projectile	✓
Trail Particle	Shuffle (Particle System)

Find the right place in the Script. You can start and stop the particles like so

particleSystemName.**Play();**

particleSystemName.**Stop();**

There's more to it, so have a look online for more particle system knowledge

https://docs.unity3d.com/ScriptReference/ParticleSystem.html

Get your game tested by someone else.

Did they enjoy it? Was it too difficult/easy?

What else to do?

Here's some extras you can add to your game if you still have time:

Programming

Have a look at existing scripts and see if you can figure out what is happening

Modify existing scripts to change the behaviour of certain things

Design

Add more rooms (scenes)

Modify the player rewards (scriptable object)

Modify the enemy scaling factor (scriptable object)

Mix

Add a start menu screen

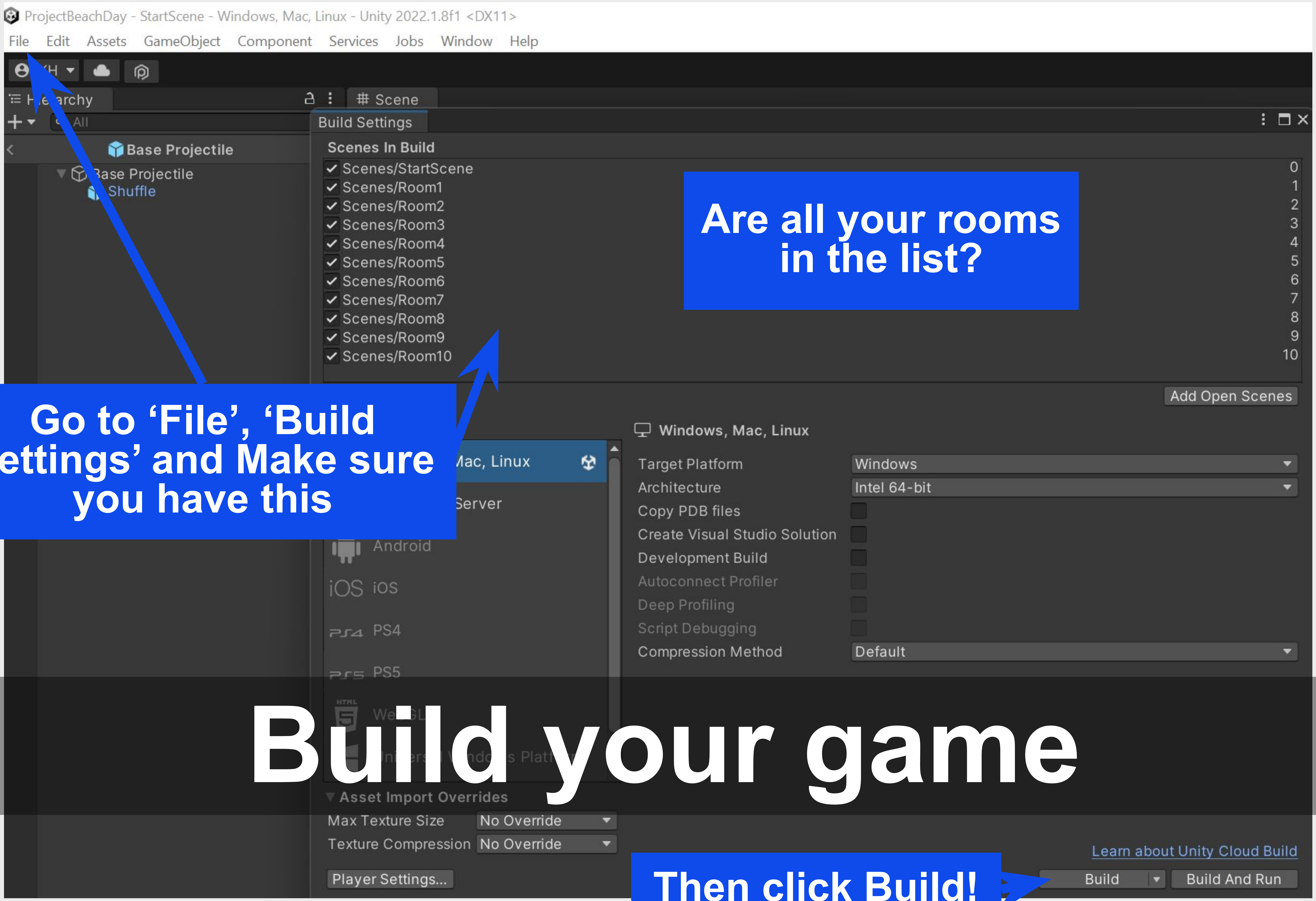
Add in your own sound effects and replace the existing ones

If you’re looking to do this, feel free to use online sources and ask for assistance

Build your game!

We are now going to ‘build’ your game so you can play it without opening Unity.

Build your game



ProjectBeachDay - StartScene - Windows, Mac, Linux - Unity 2022.1.8f1 <DX11>

File Edit Assets GameObject Component Services Jobs Window Help

Build Settings

Scenes In Build

✓ Scenes/StartScene	0
✓ Scenes/Room1	1
✓ Scenes/Room2	2
✓ Scenes/Room3	3
✓ Scenes/Room4	4
✓ Scenes/Room5	5
✓ Scenes/Room6	6
✓ Scenes/Room7	7
✓ Scenes/Room8	8
✓ Scenes/Room9	9
✓ Scenes/Room10	10

Add Open Scenes

Windows, Mac, Linux

Target Platform	Windows
Architecture	Intel 64-bit
Copy PDB files	☐
Create Visual Studio Solution	☐
Development Build	☐
Autoconnect Profiler	☐
Deep Profiling	☐
Script Debugging	☐
Compression Method	Default

Asset Import Overrides

Max Texture Size	No Override
Texture Compression	No Override

Player Settings...

Learn about Unity Cloud Build

Build | Build And Run

Are all your rooms in the list?

Go to 'File', 'Build Settings' and Make sure you have this

Then click Build!

Save your game to a USB or your own computer

Once it is built, you can just double click on it and it will run!