Getting Started

Game Kit 2D

Game Kit Introduction

This guide will walk you through setting up an empty scene to start creating a new level with the Game Kit. This will walk you through some of the fundamentals used in this Kit to create gameplay.

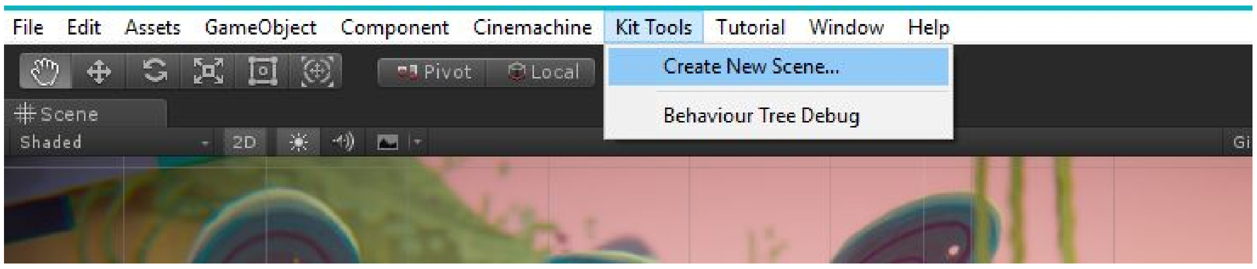
The Kit comes with a premade game where you can find example of every part of the Kit in use if you get stuck for ideas.

Creating a new scene

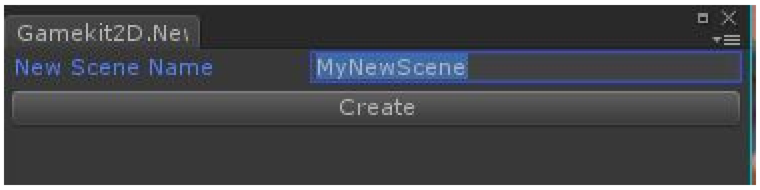
The project contains a utility to automatically create a new default scene. This scene will contain Ellen (our hero), a small platform, Health UI and everything you need to move around and attack.

Let's create a new scene;

- Click on **Kit Tools** from the top menu
- Choose **Create New Scene...**



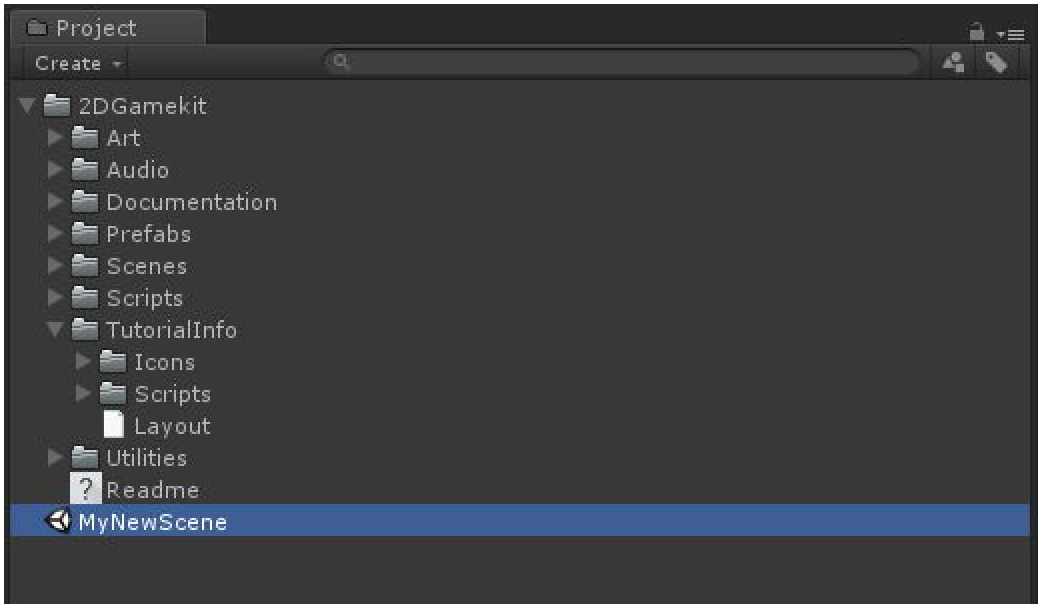
- A new Window will open



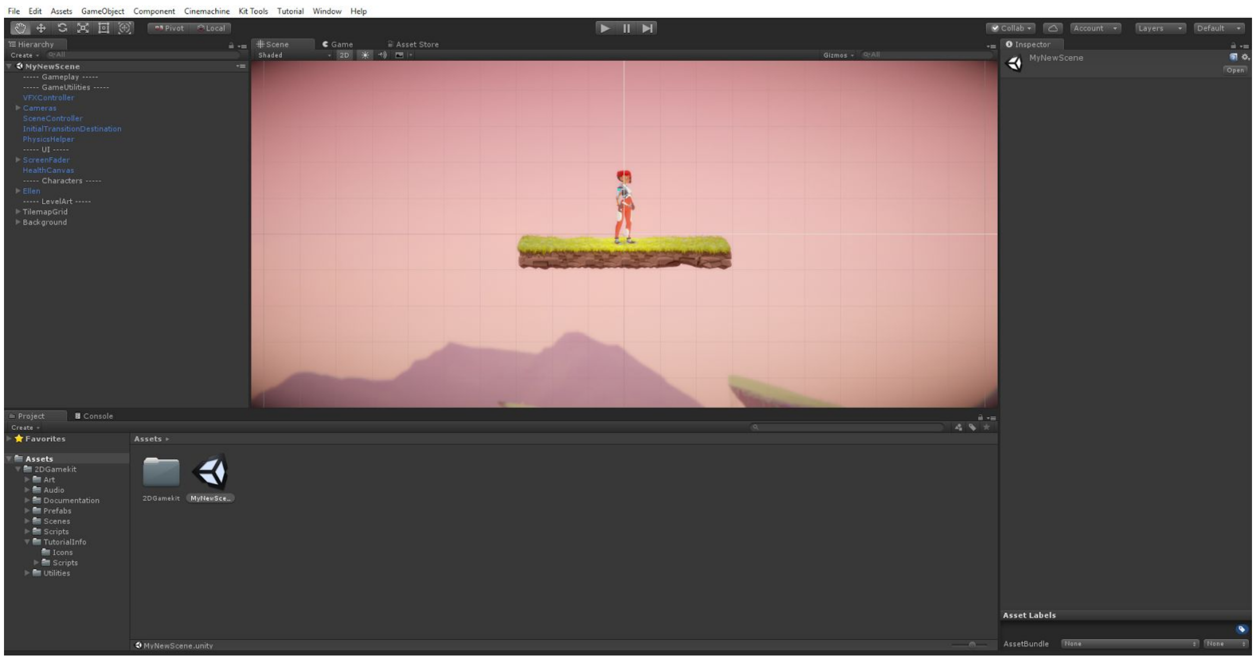
New Scene Name MyNewScene

- In the empty field next to **New Scene Name**, enter the name you would like to call your scene
- Click **Create**

A new scene will be created in the root of your project and the scene will be added to the build settings for you.



The scene we just created will also be opened in the editor so you can start working on it.

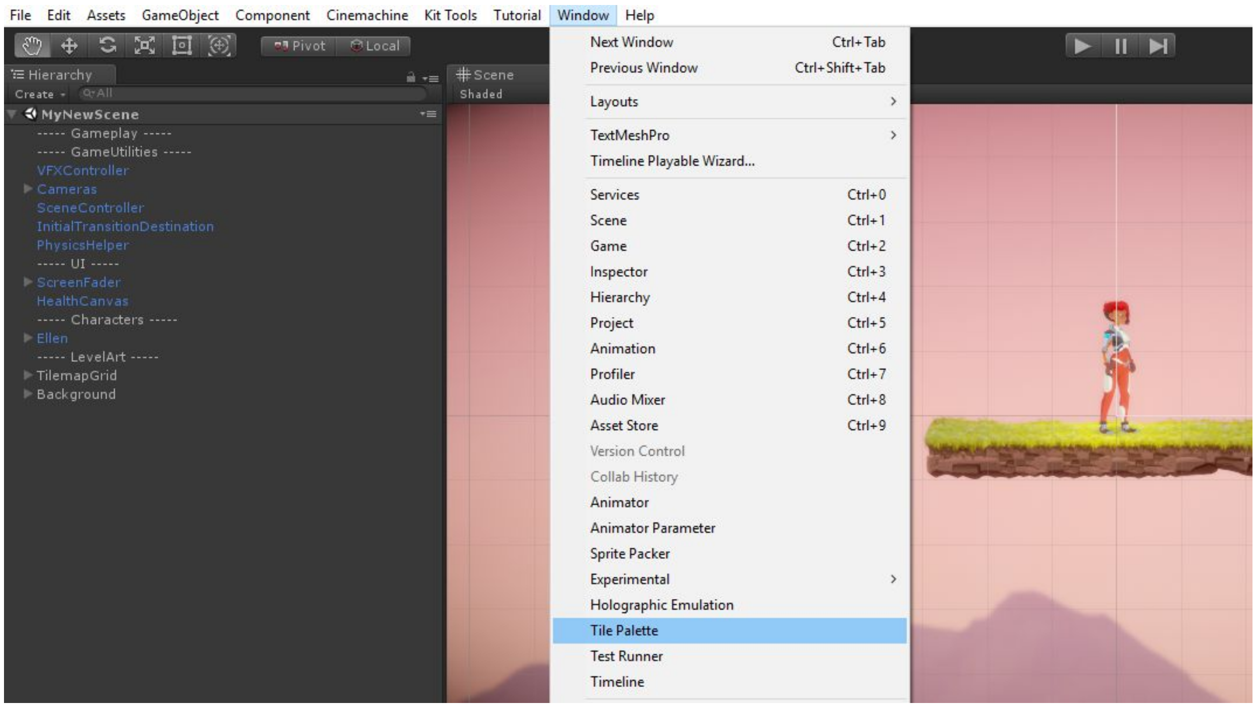


Painting a Level

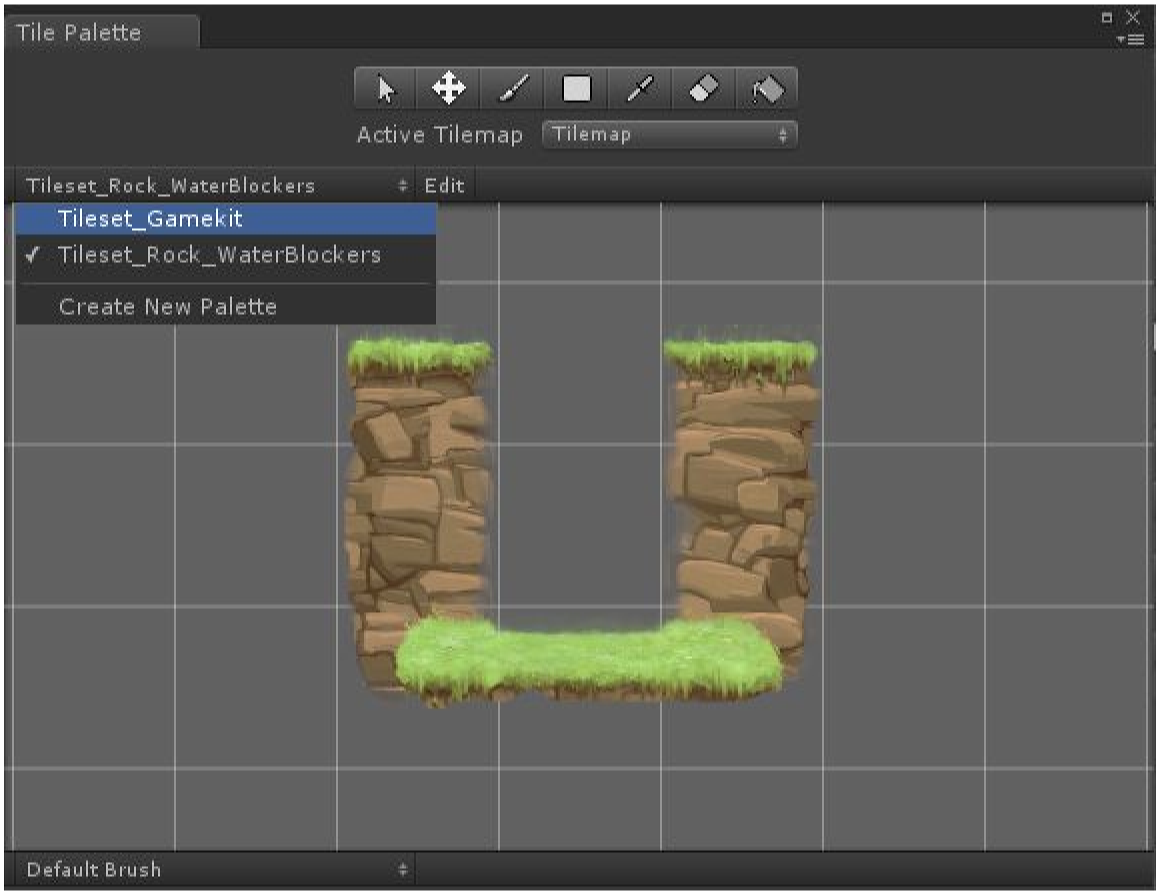
In 2D we can easily paint a level, the Kit uses Unity's ***Tilemap*** feature you can easily draw out a level. The Kit has everything set up for you so you can quickly get up and running.

Let's start designing a level:

- Go to **Window > Tile Palette**



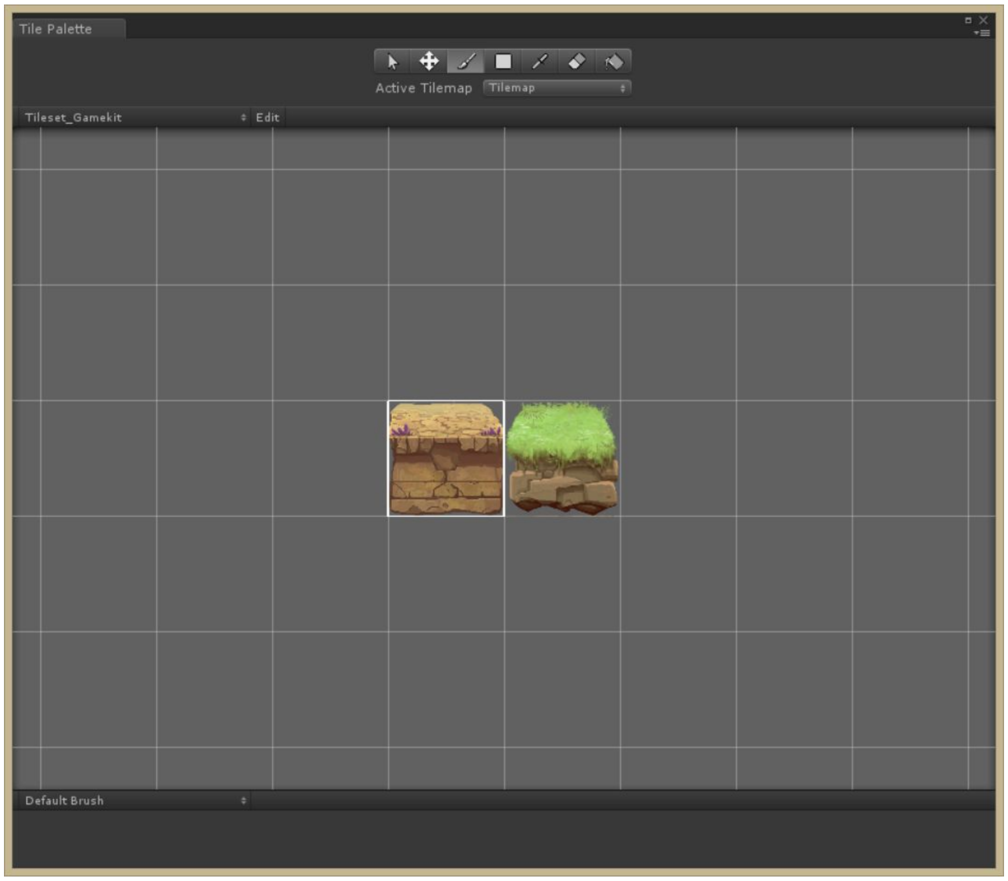
The Tile Palette window will open;



We have created two **Tilesets** that will allow you to draw out levels, these tiles use the **Scriptable Tile** system of the Unity **Tilemap**. They automatically choose the right sprite for a platform based on settings in the **Rule Tile Asset**. We have two styles of sprites; Grassy Rocks and Alien Structure.

We need to make sure the correct set of tiles in **Tile Palette** is selected;

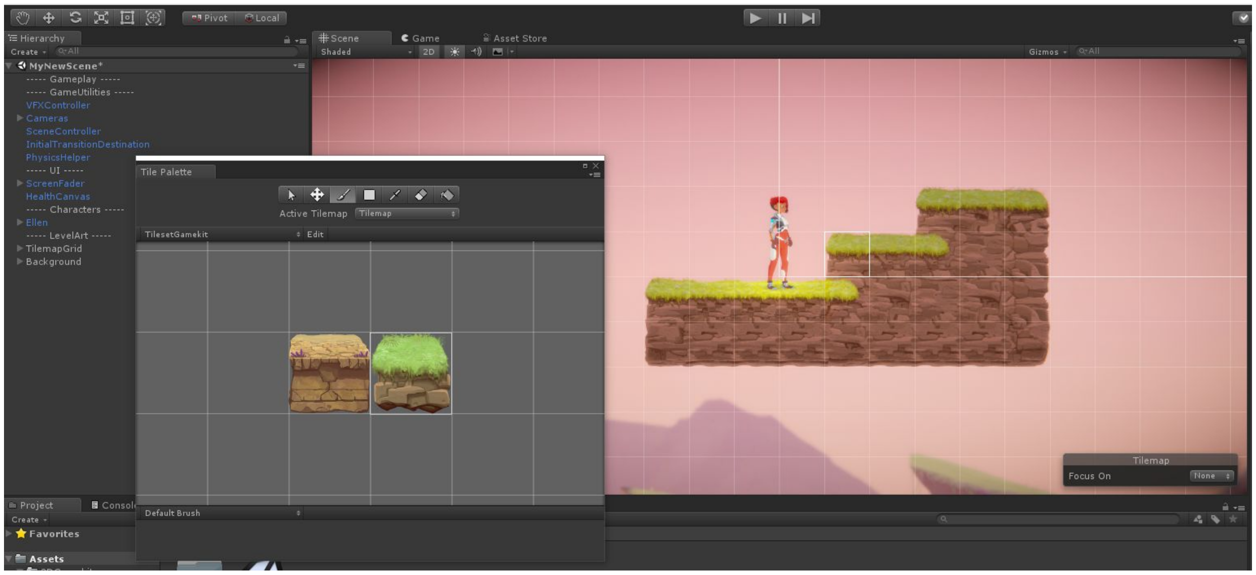
- Click the **Tileset Menu**
- Click on **Tileset_Gamekit**
- Pick the type of tile you like best for your level



Now let's paint some tiles in the scene to create a level;

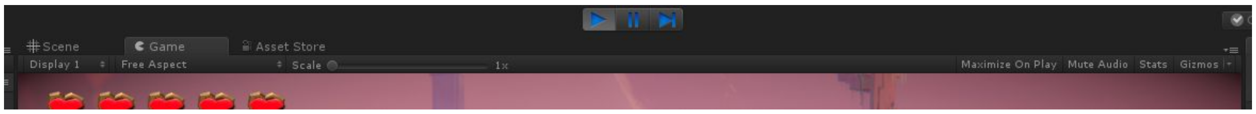
- Select one of the two tiles by clicking on it in the **Tile Palette**
- Navigate to the **Scene View**
- Left click and drag to draw out your level

If you make a mistake you can erase Tiles by holding **Shift** and left click or drag.



Testing your Level

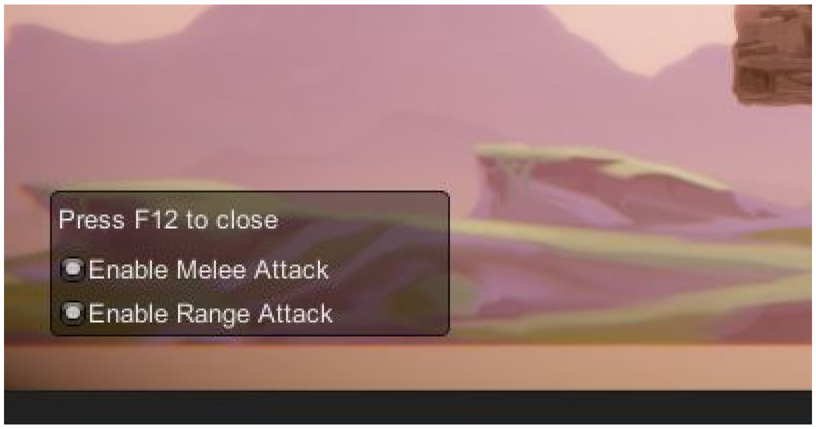
Now that you have created a level, let’s run around in it.



- Press **Play** at the top of the Editor

The controls for Ellen (our player character) are as follows

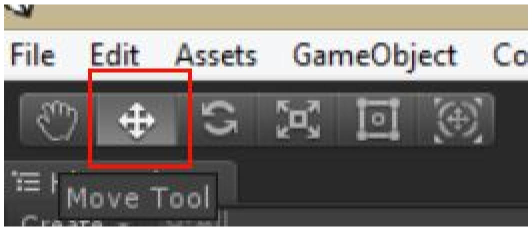
Move	**A, D**
Jump	**Space**
Crouch	**S**
Shoot	**O**
Melee	**K**
Activate	**E**
Pass Through Platform	**S** and **Space**



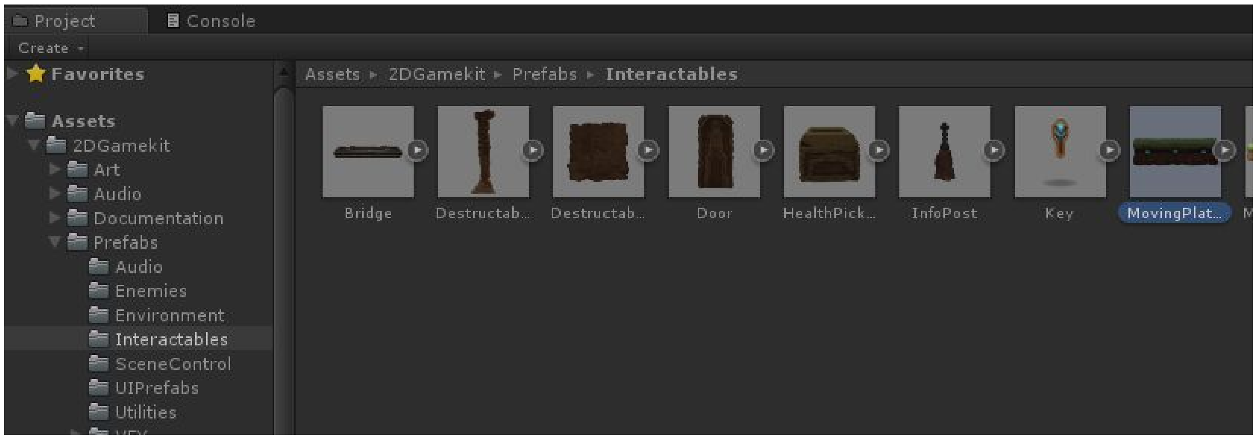
We have included a debug menu, pressing **F12** on your Keyboard will bring up the option to enable or disable each of Ellen's weapons in the **Game View**.

Adding a Moving Platform

Before we begin manipulating **GameObjects** in the **Scene View**, make sure that you are not in paint mode by closing the **Tile Palette** or selecting the **Move Tool** at the top left of the editor or by pressing **W** on your keyboard.



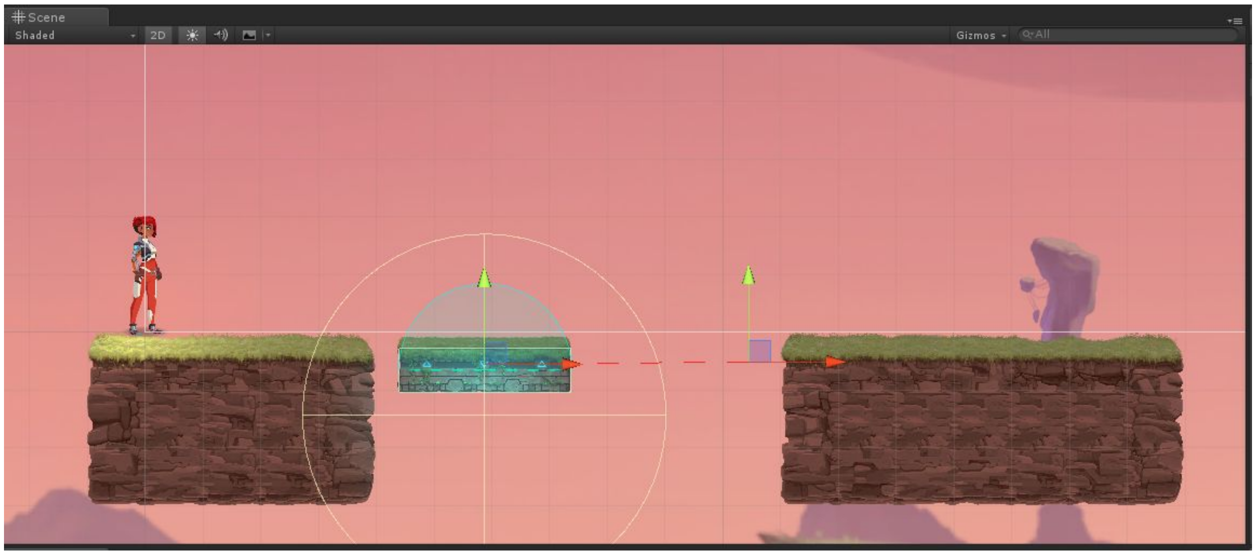
Now let’s add a moving platform to our level;



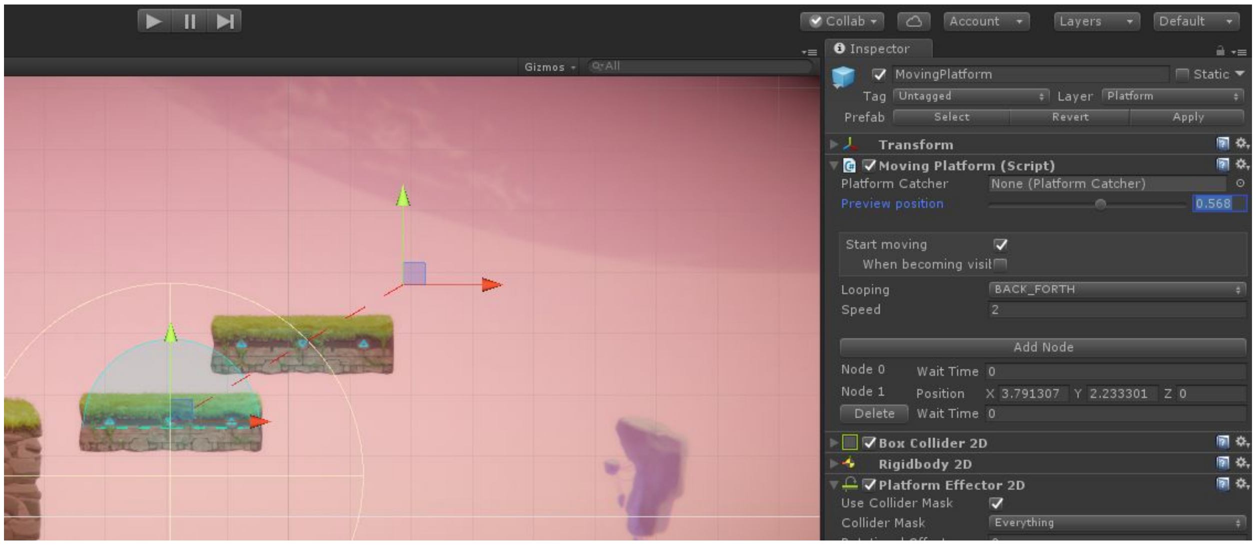
- Navigate to the **Project Window**
- Go to **Prefabs** < **Interactables**
- Right click and drag **MovingPlatform** into the **Scene View**
- Make sure the **MovingPlatform** is selected in the Hierarchy
- Press **W** to use the Move Tool

Using the Move Tool on the Moving Platform, position it to where you would like in your level.

There is a red dotted line and a move tool at the end of it, this indicates the path the platform will take and end up in.



- Use the move tool manipulator at the end of the red dotted line to indicate a path you would like the platform to take



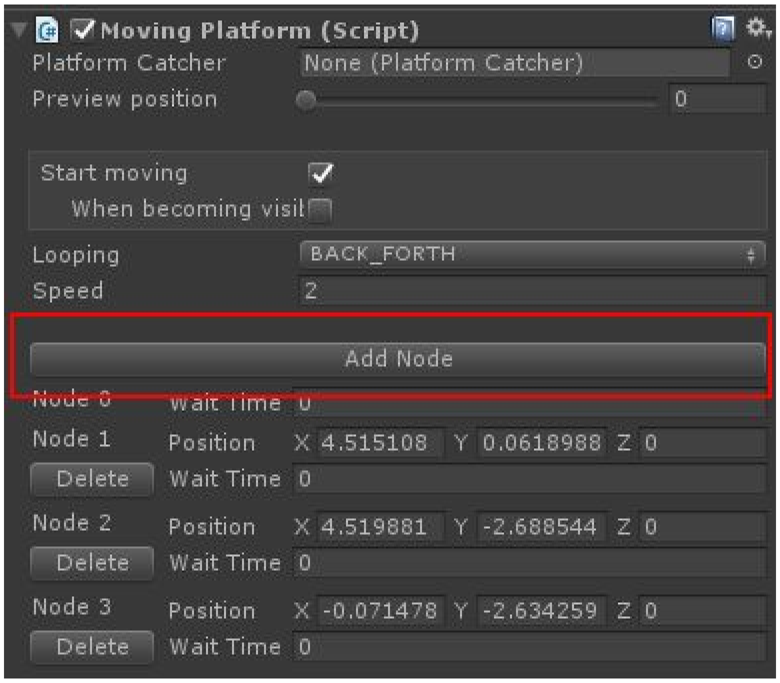
MovingPlatform ✓ Static
Tag Untagged Layer Platform
Prefab Select Revert Apply
Moving Platform (Script)
Platform Catcher None (Platform Catcher)
Preview position 0.568
Start moving ✓
When becoming visil
Looping BACK_FORTH
Speed 2
Add Node
Node 0 Wait Time 0
Node 1 Position X 3.791307 Y 2.233301 Z 0
Delete Wait Time 0
Use Collider Mask ✓
Collider Mask Everything

We can preview the path the Moving Platform will take, with the Moving Platform selected;

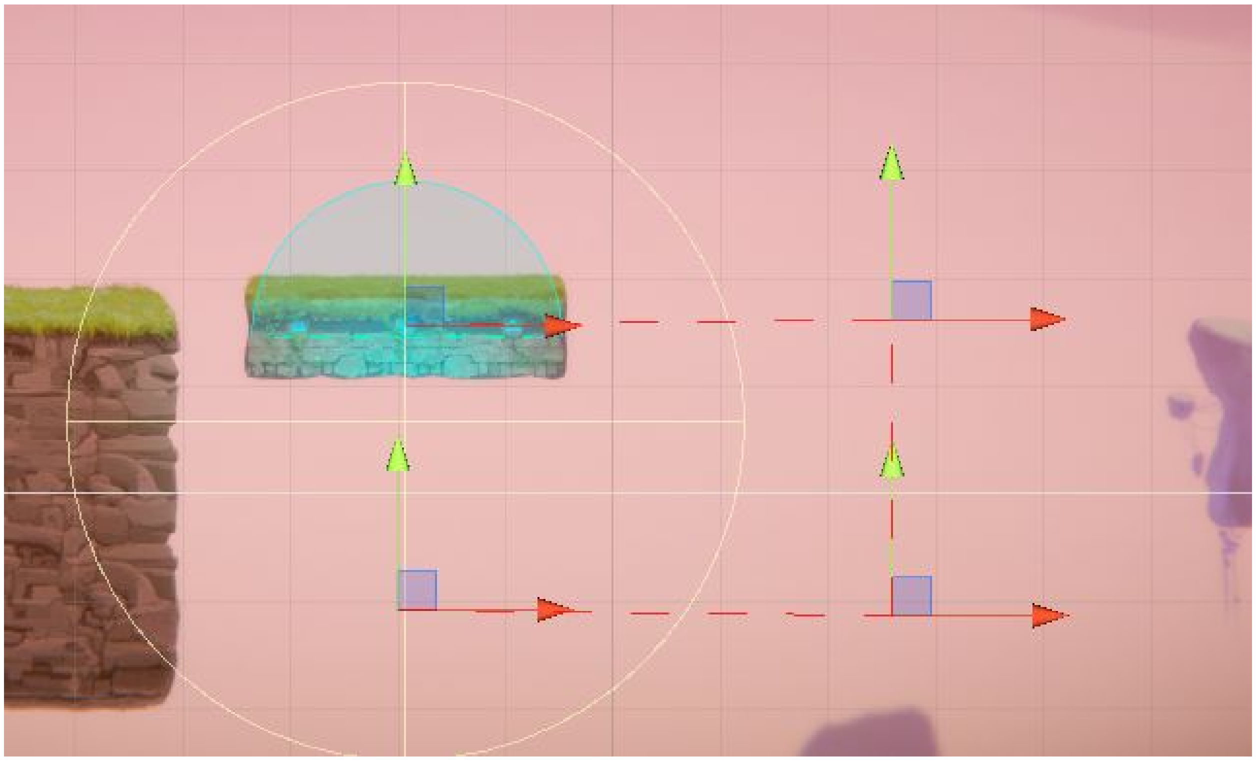
- Navigate to the **Inspector**
- In the **Moving Platform** Component find the **Preview Position** slider
- Click and drag to scrub and see where the Moving Platform will go.

Let's make the path of this platform a little more complex by making it loop in a square, we do this by adding **Nodes.**

Nodes are additional navigation points for the Moving Platform.


Moving Platform (Script)
Platform Catcher None (Platform Catcher)
Preview position 0
Start moving ✓
When becoming visil
Looping BACK_FORTH
Speed 2
Add Node
Node 0 Wait Time 0
Node 1 Position X 4.515108 Y 0.0618988 Z 0
Delete Wait Time 0
Node 2 Position X 4.519881 Y -2.688544 Z 0
Delete Wait Time 0
Node 3 Position X -0.071478 Y -2.634259 Z 0
Delete Wait Time 0

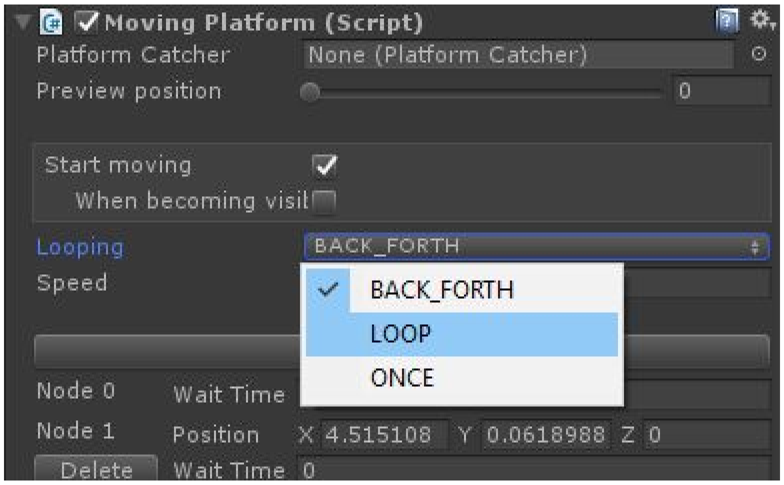
- In the **Inspector** locate the **Moving Platform** Component
- Click **Add Node** twice



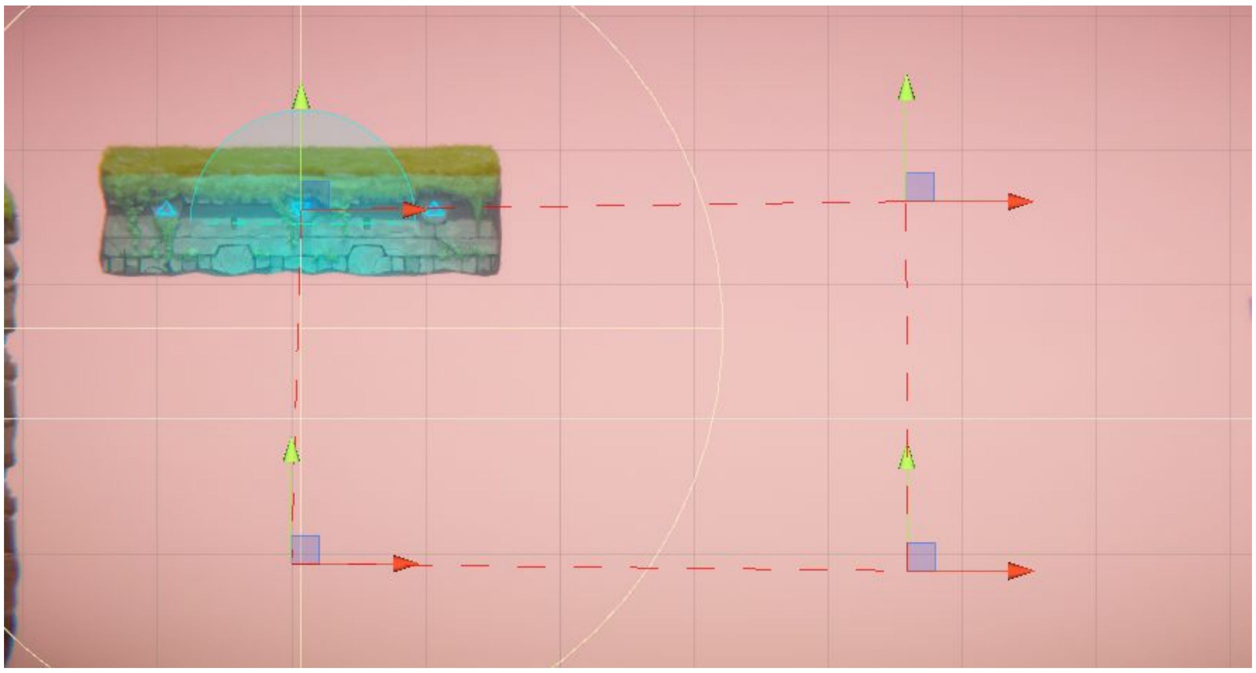
This will add an additional two red dotted lines and gizmos, use the gizmos to position your extra **Nodes** into a square shape.

You’ll notice that there isn’t a path going from the last node back to beginning. If we try this now it will stop at the last point and go backwards on the path and start again.

We can change the settings of the platform to make it form a loop;


Moving Platform (Script)
Platform Catcher None (Platform Catcher)
Preview position 0
Start moving ✓
When becoming visil
Looping BACK_FORTH
Speed
Node 0 Wait Time
Node 1 Position X 4.515108 Y 0.0618988 Z 0
Delete Wait Time 0

- In the **Moving Platform** Component find the **Looping** option
- Click the dropdown that says **BACK_FORTH**
- Select **Loop**



A loop will now be formed in the gizmos and the platform will take this path. Note: The preview slider will not preview a complete loop, press play to see it fully.

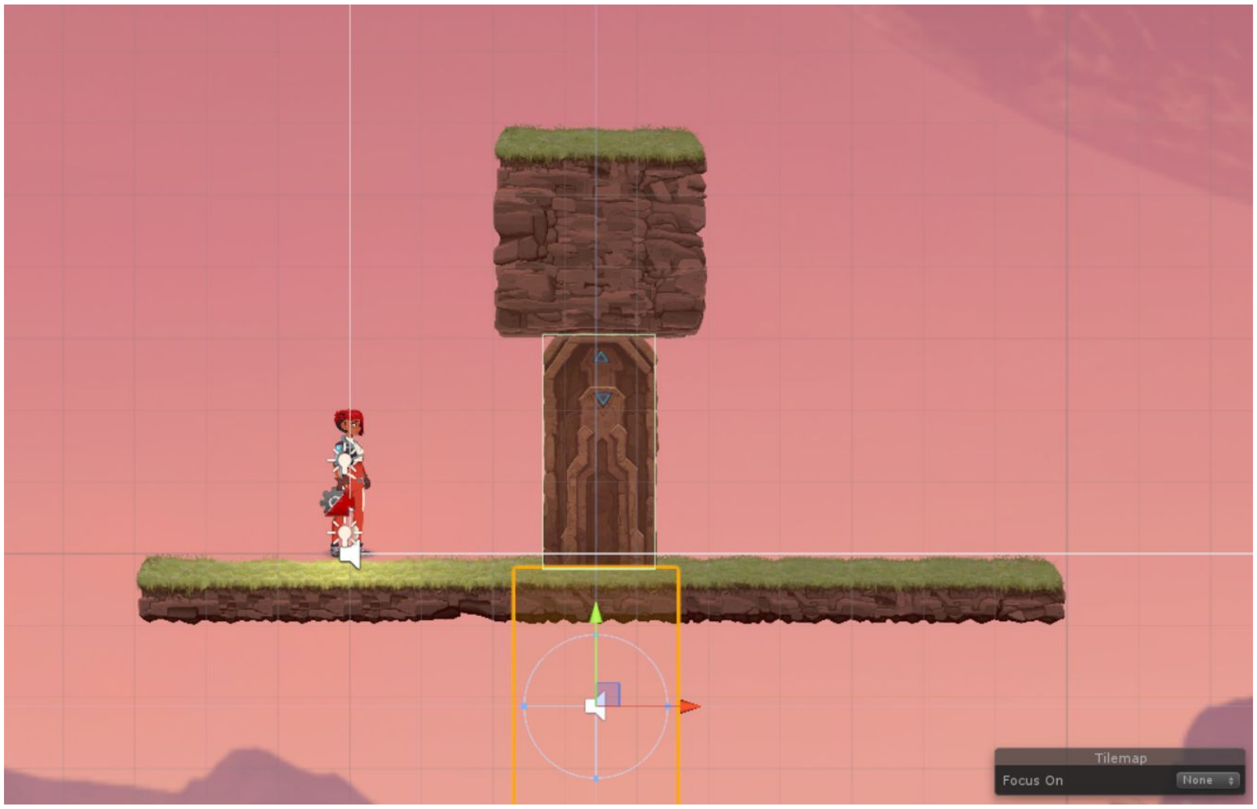
Every Component we have provided in the Kit has many options to customise the way they work. For more information on what every option does in the **Moving Platform** Component, check the Component Documentation and find the Moving Platform section.

Opening a Door with Events

Creating actions in the Kit happens with Events, we'll be using Events to trigger a door to open by stepping on a pressure pad.

Let's start by adding in a Door;

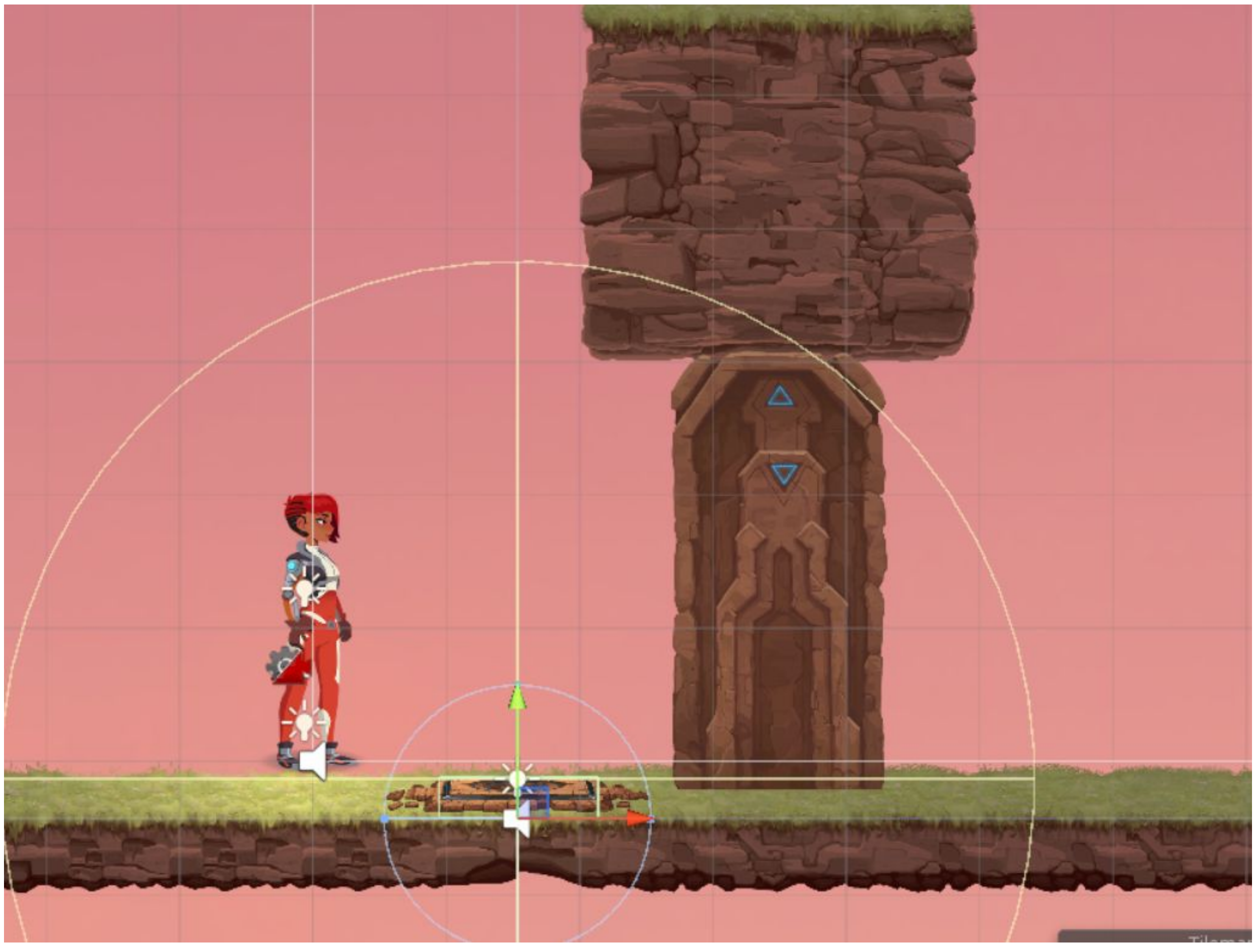
- In the **Project Window** go to **Prefabs** < **Interactables**
- Find the **Door** Prefab and drag it into the **Scene View** and block the players path



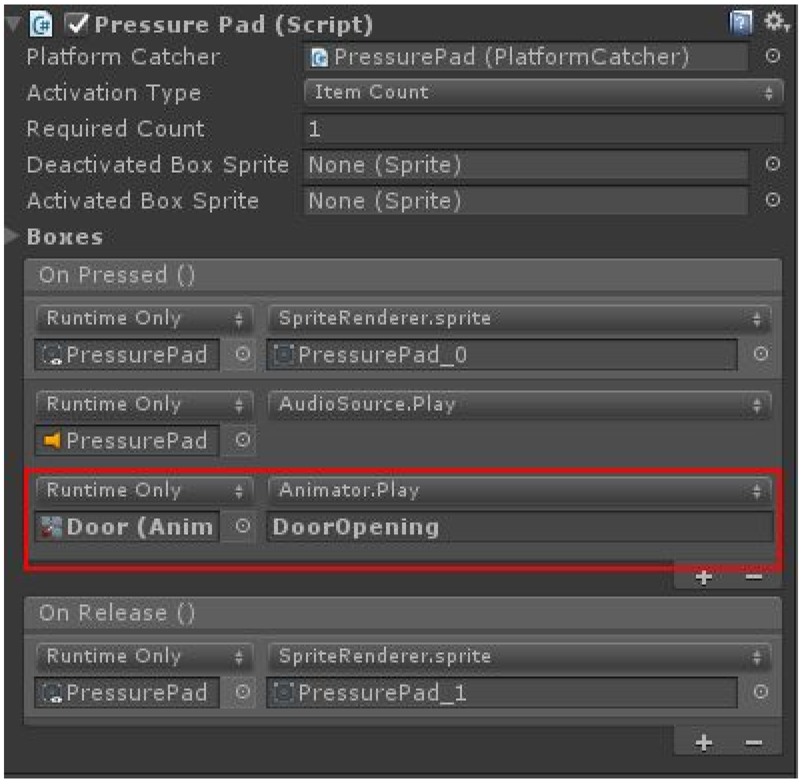
Now let’s add the pressure pad into the scene

- In **Prefabs** < **Interactables** find the **PressurePad** Prefab
- Drag it close to the ground in front of the **Door**

Note: If the player seems to be blocked by the pressure pad instead of stepping on it, lower it slightly. The player can easily get stuck against the collider.



Now let’s connect the **PressurePad** to the **Door**. Some events are already defined, when stepped on, the **PressurePad** will play a sound and light up.



Pressure Pad (Script)

Platform Catcher	PressurePad (PlatformCatcher)
Activation Type	Item Count
Required Count	1
Deactivated Box Sprite	None (Sprite)
Activated Box Sprite	None (Sprite)

Boxes

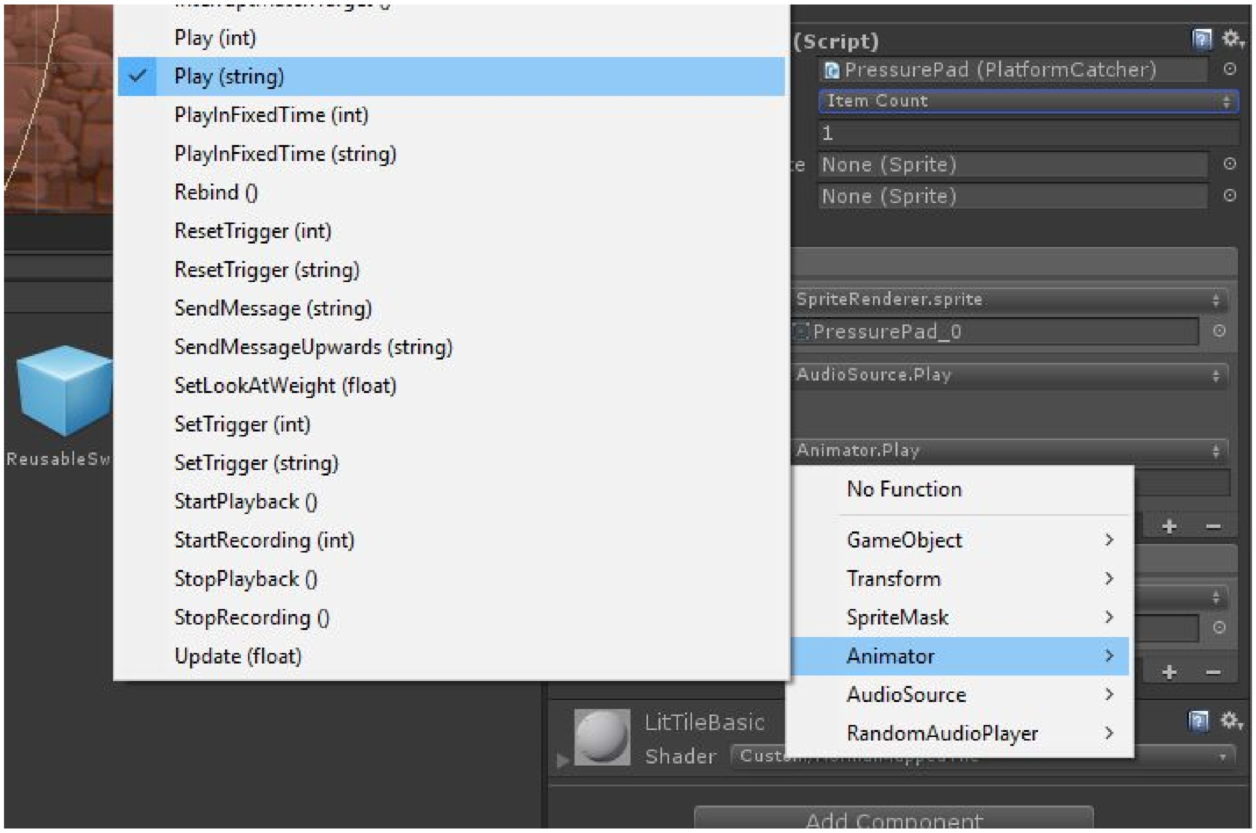
On Pressed ()

Runtime Only	SpriteRenderer.sprite
PressurePad	PressurePad_0
Runtime Only	AudioSource.Play
PressurePad	
Runtime Only	Animator.Play
Door (Anim	DoorOpening

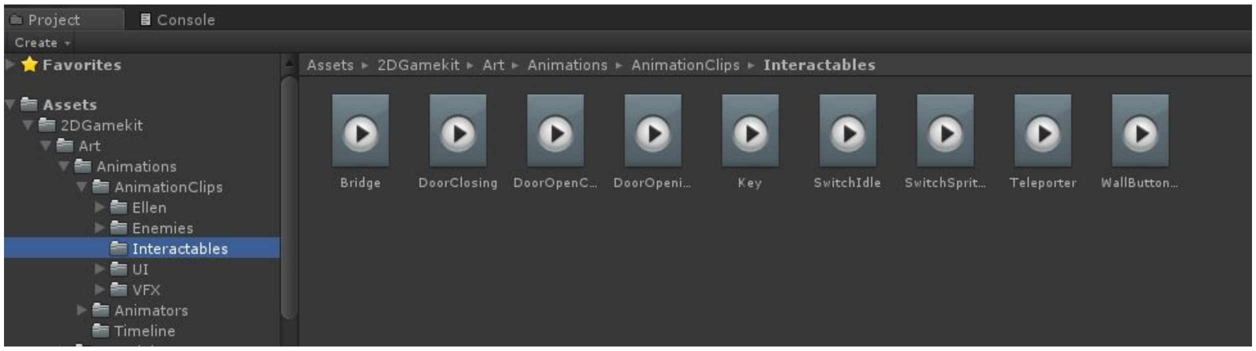
On Release ()

Runtime Only	SpriteRenderer.sprite
PressurePad	PressurePad_1

- In the **Hierarchy** select the **PressurePad**
- In the **Inspector**, find the **PressurePad** component
- In the **On Pressed** list, click the **+** at the bottom right to add a new event
- Drag the **Door** from the **Hierarchy** to the **None(Object)** field in the event



- In the **No Function** dropdown, find **Animator** < **Play(string)**
- In the text box that appeared under the dropdown, enter the text: **DoorOpening**
- Press **Play** and run onto the Pressure Pad using **WASD**, watch the door open.



All the **Animation Clips** are stored in **Art** < **Animations** < **Animation Clips** in the **Project** Window. To play different Animators in an Event, you must match the name (String) of the **Animation Clip** exactly.

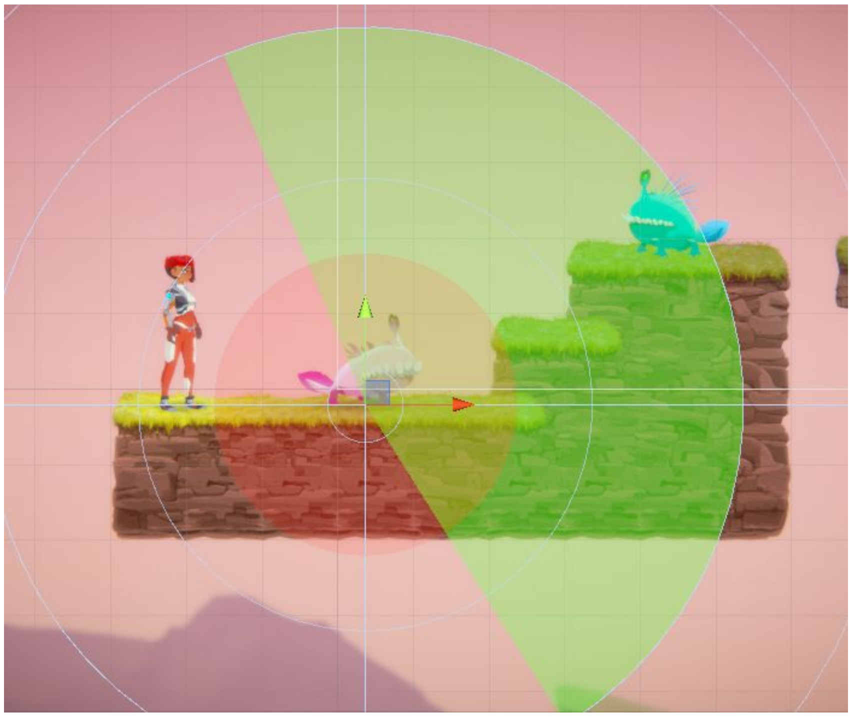
If you wanted to shoot a switch instead of a pressure pad, you can do that by using the same steps but selecting the **ResusableSwitch** in the prefabs folder instead. Give it a try!

Enemies

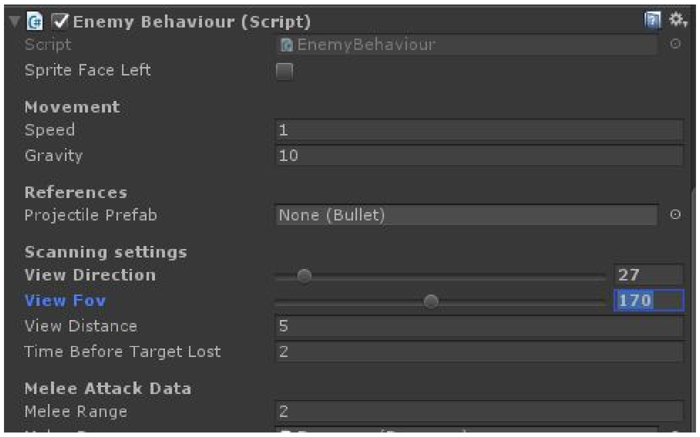
We have two pre-made enemies in the Kit, Chomper and Spitter.
They can be found in **Prefabs** < **Enemies**

They are both controlled by the **Enemy Behaviour** Component in the Inspector. This is where you can adjust their speed, field of view (FOV), health and much more. Each enemy has the same component they are just set up slightly differently.

Try adding a **Chomper** in your scene and adjust his speed and field of view (FOV), start playing around with the settings. Don’t forget you can attack by pressing **O** or **K** you can press **F12** to make sure weapons are enabled.



Here we can see a **Chomper** is selected, his **View FOV** and **View Direction** have been adjusted so he won't spot Ellen until he turns around. For more information on the **Enemy Behaviour** Component check the Component Documentation.

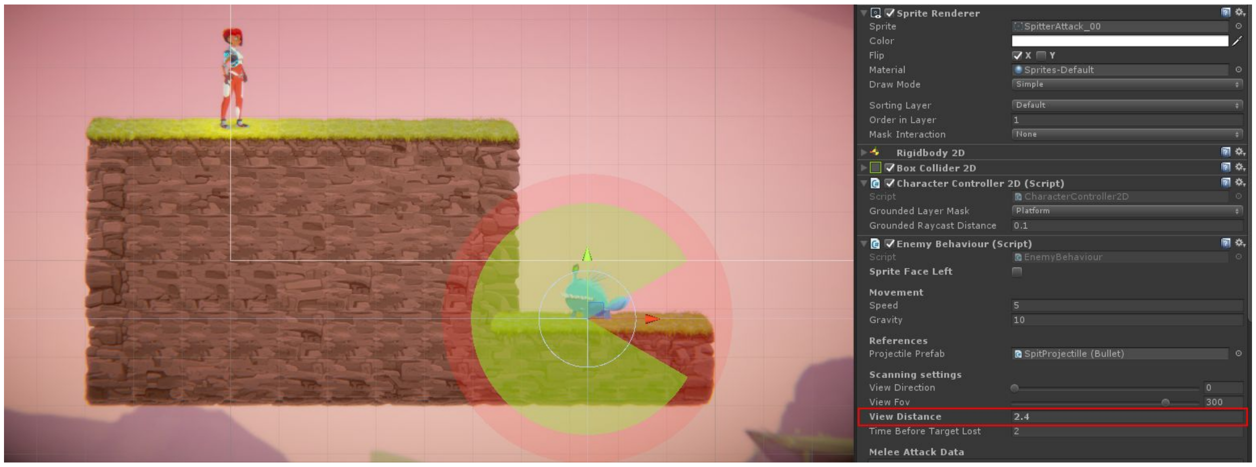


```
Enemy Behaviour (Script)
Script                      EnemyBehaviour
Sprite Face Left

Movement
Speed                       1
Gravity                     10

References
Projectile Prefab           None (Bullet)

Scanning settings
View Direction                                27
View Fov                                      170
View Distance               5
Time Before Target Lost     2

Melee Attack Data
Melee Range                 2
```

Damaging with Objects

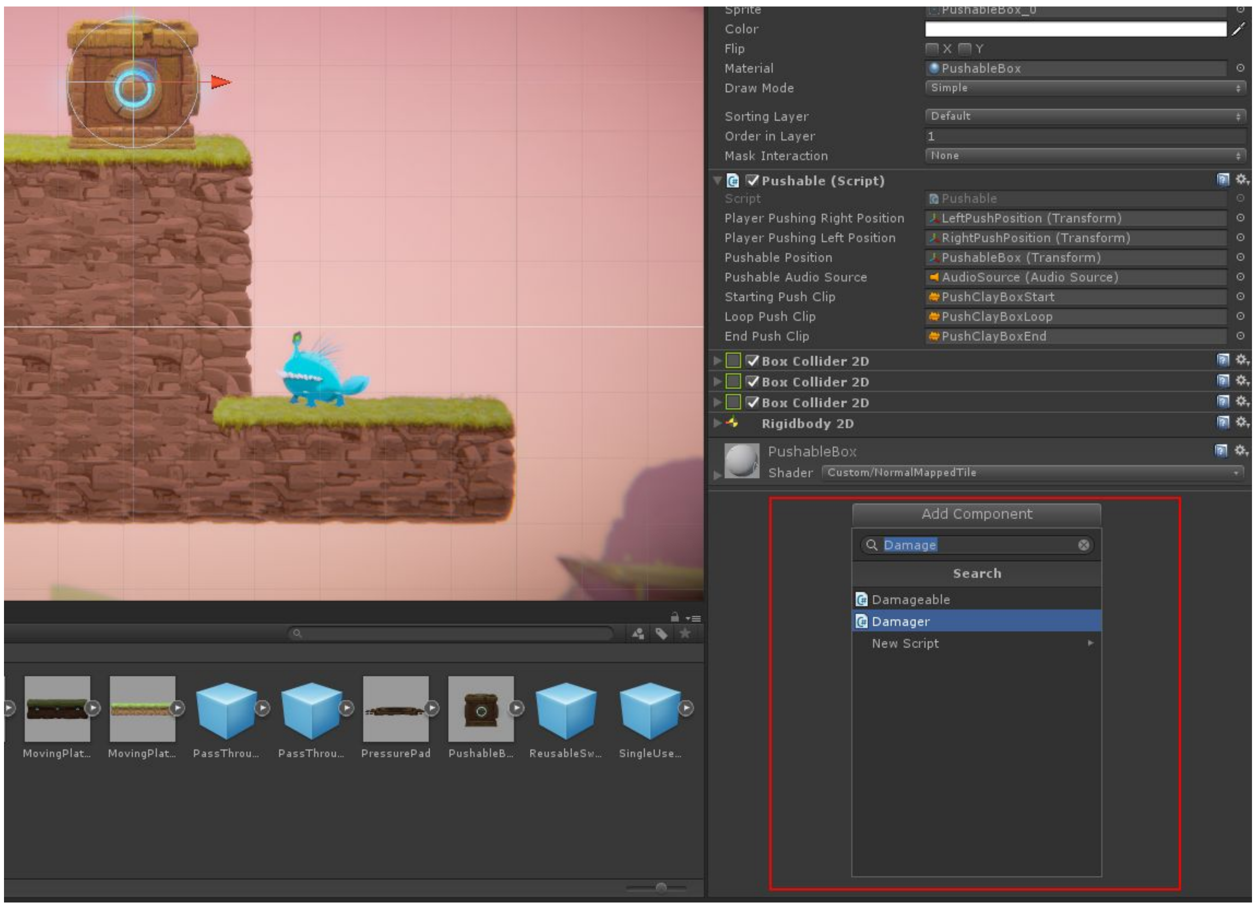
In this section we'll go through dropping a box on a Spitter to kill him (sorry little guy) and thus exploring the damage system.

Start by drawing out a level where Ellen is higher up and there is a drop with a platform for Spitter to be placed.



```
Sprite Renderer
Sprite                      SpitterAttack_00
Color
Flip                        ✓ X  Y
Material                    Sprites-Default
Draw Mode                   Simple

Sorting Layer               Default
Order in Layer              1
Mask Interaction            None
Rigidbody 2D
Box Collider 2D
Character Controller 2D (Script)
Script                      CharacterController2D
Grounded Layer Mask         Platform
Grounded Raycast Distance   0.1
Enemy Behaviour (Script)
Script                      EnemyBehaviour
Sprite Face Left

Movement
Speed                       5
Gravity                     10

References
Projectile Prefab           SpitProjectille (Bullet)

Scanning settings
View Direction              0
View Fov                    300
View Distance               2.4
Time Before Target Lost     2

Melee Attack Data
```

- Go to **Prefabs** < **Enemies** and drag a Spitter into the **Scene View**
- Place him on the lower portion of your level close to the cliff face
- With **Spitter** selected locate the **Enemy Behaviour** Script
- Reduce the **View Distance** number so that **Spitter** does not shoot you immediately so you can test this gameplay (You can also click and drag left/right on the word View Distance to scrub through values.)



Sprite	PushableBox_0
Color	
Flip	X Y
Material	PushableBox
Draw Mode	Simple
Sorting Layer	Default
Order in Layer	1
Mask Interaction	None

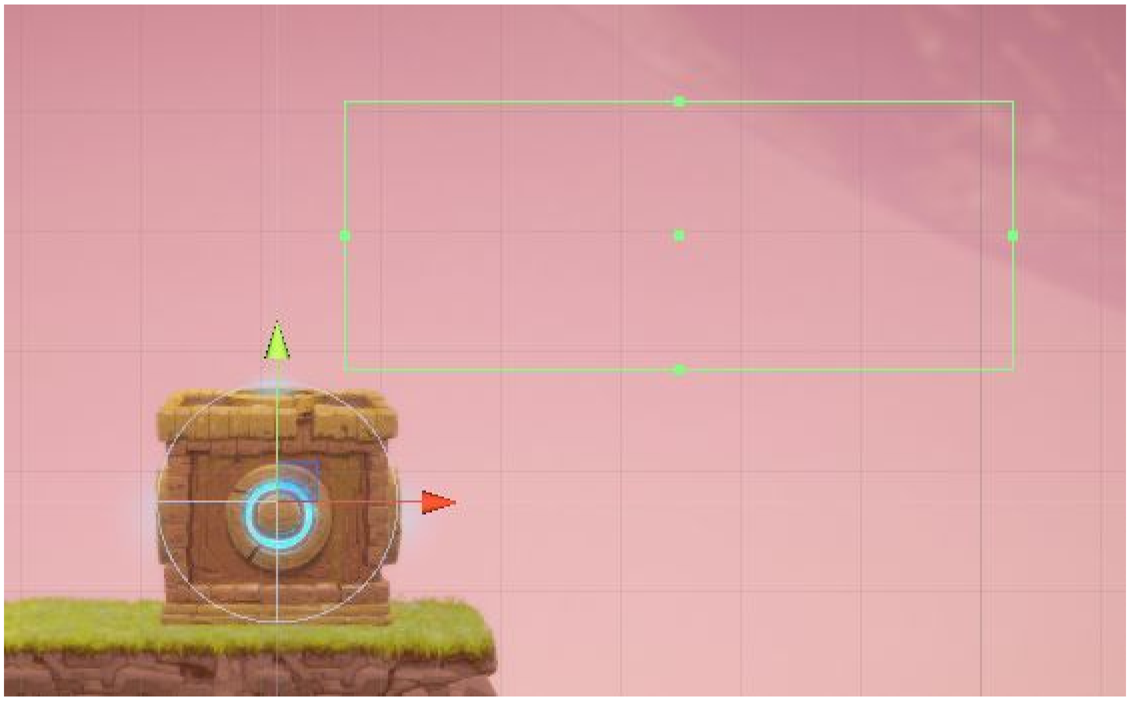
Pushable (Script)

Script	Pushable
Player Pushing Right Position	LeftPushPosition (Transform)
Player Pushing Left Position	RightPushPosition (Transform)
Pushable Position	PushableBox (Transform)
Pushable Audio Source	AudioSource (Audio Source)
Starting Push Clip	PushClayBoxStart
Loop Push Clip	PushClayBoxLoop
End Push Clip	PushClayBoxEnd

Box Collider 2D
Box Collider 2D
Box Collider 2D
Rigidbody 2D
PushableBox
Shader Custom/NormalMappedTile

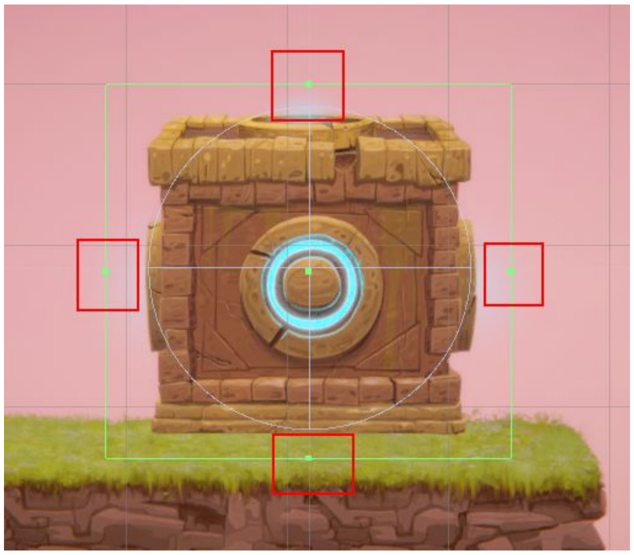
- In the **Project Window** got to **Prefabs** < **Interactables** click and drag the **PushableBox** Prefab into the **Scene View**
- With the **PushableBox** selected, at the bottom of the **Inspector** click **Add Component**
- In the **Search Box** type **Damage**
- Click on **Damager** to add it to the **Pushable Box**

A **Damage**r is a component that will tell any object that has a **Damageable** component on it, like a Spitter or a Chomper, to give it damage. There is more in depth information on this system in the Component Documentation.



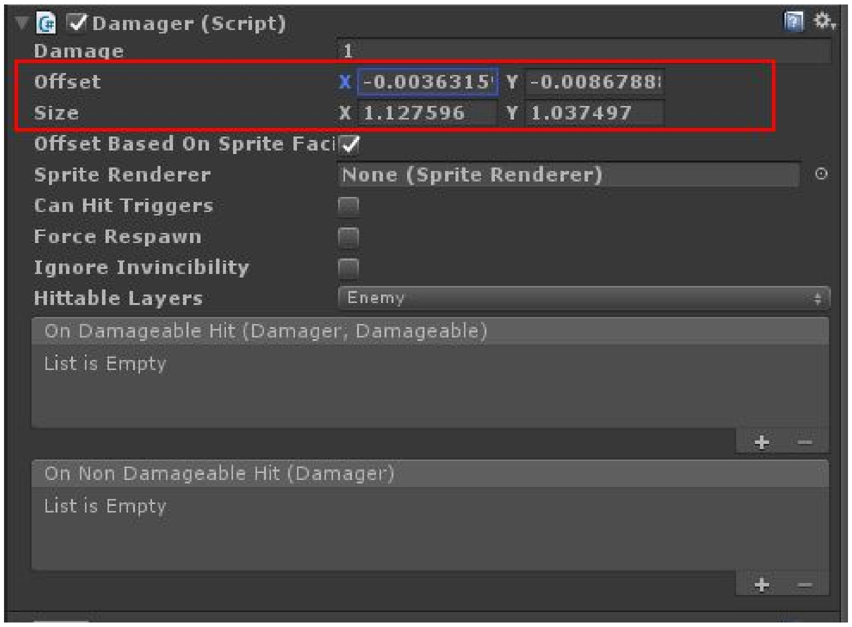
The **Damager** is represented by a green collision box as shown, this is the area which causes damage. Since this is not covering the **PushableBox** right now, when we push the box onto the Spitter, it will not damage him.

Let's move this box so that it's roughly the size and position of the **PushableBox**, this can be achieved in two ways;



- Select and drag the green dots on the edges of the green collision box to be over the **PushableBox**

Or


Damager (Script)

Damage	1
Offset	X -0.0036315 Y -0.0086788
Size	X 1.127596 Y 1.037497
Offset Based On Sprite Faci	✓
Sprite Renderer	None (Sprite Renderer)
Can Hit Triggers	
Force Respawn	
Ignore Invincibility	
Hittable Layers	Enemy

On Damageable Hit (Damager, Damageable)

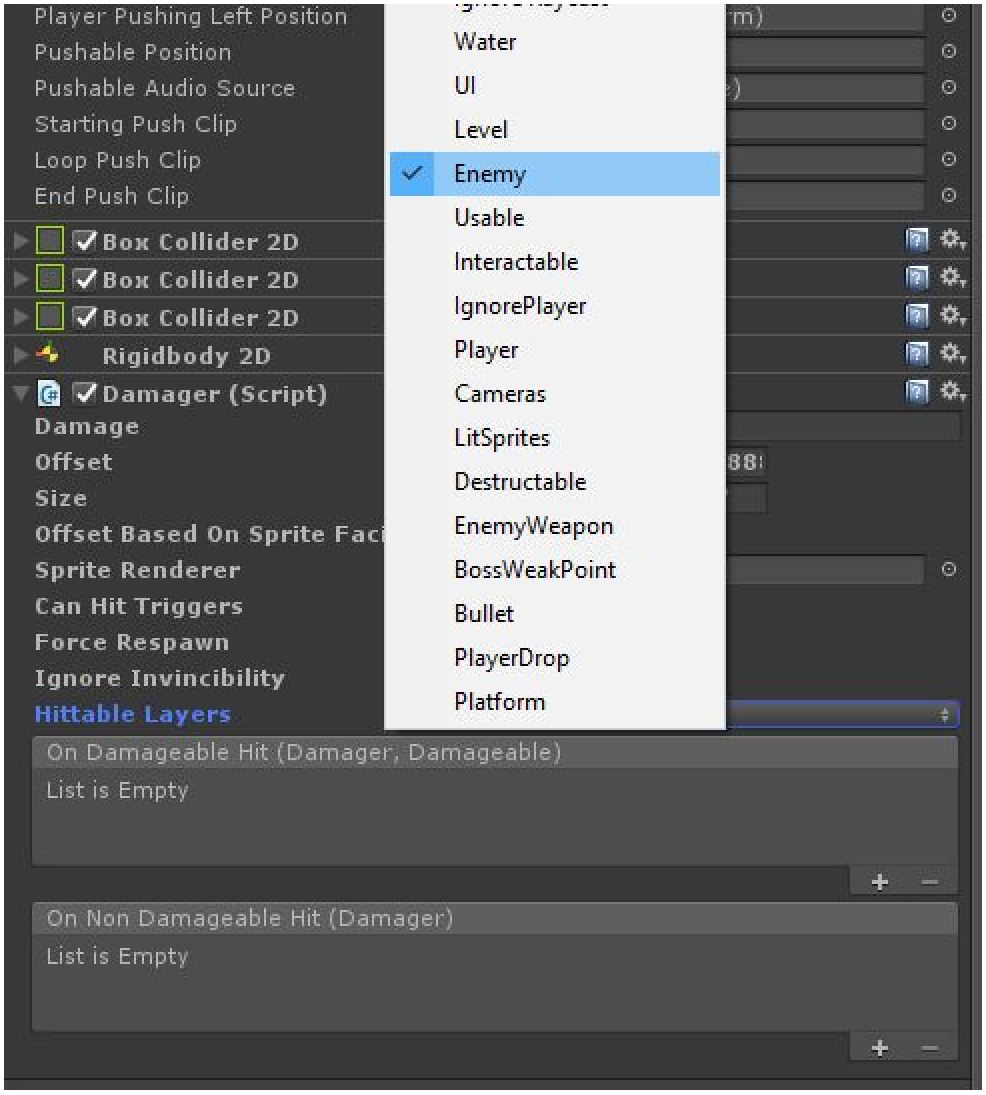
List is Empty

On Non Damageable Hit (Damager)

List is Empty

- In the **Inspector** locate the **Damager Component**
- Adjust the **Offset** and **Size** by left clicking on the words and dragging left and right to scrub through values to position and size the collision box.

Lastly we need to make sure the damage is given to the right objects, we separate objects into ***Layers*** in the Editor so that they can easily be found and seperated.


Player Pushing Left Position
Pushable Position
Pushable Audio Source
Starting Push Clip
Loop Push Clip
End Push Clip
Box Collider 2D
Box Collider 2D
Box Collider 2D
Rigidbody 2D
Damager (Script)
Damage
Offset
Size
Offset Based On Sprite Faci
Sprite Renderer
Can Hit Triggers
Force Respawn
Ignore Invincibility
Hittable Layers
On Damageable Hit (Damager, Damageable)
List is Empty
On Non Damageable Hit (Damager)
List is Empty

- Select the **PushableBox**
- In the **Inspector**, find the **Damager** Component
- On the **HittableLayers** dropdown select the **Enemy** Layer

The **PushableBox** will now cause damage to anything on the **Enemy** Layer, like our Spitter.

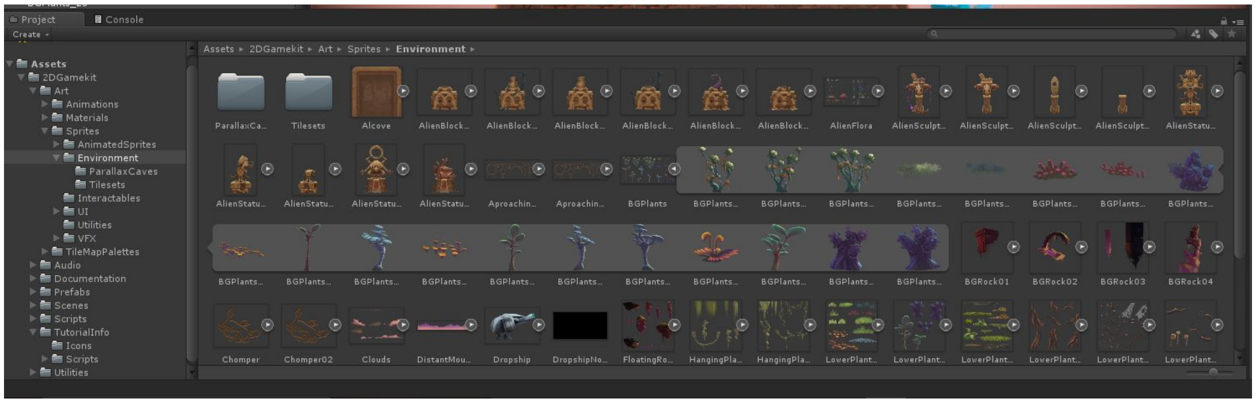
What other objects can you use to cause damage to an enemy, or even Ellen?

Decorating

The Kit also includes decorative sprites, these are all used in the example game, The Explorer that is included in the Kit. Tip: If something catches your eye in the example, you can look up the name and search it in the Project View.

To make your level Instragrm worthy, all the decorations are located in
Art < **Sprites** < **Environment**

A lot are stored in sub folders, so don't forget to expand the small arrows and see what other sprites are in that category.

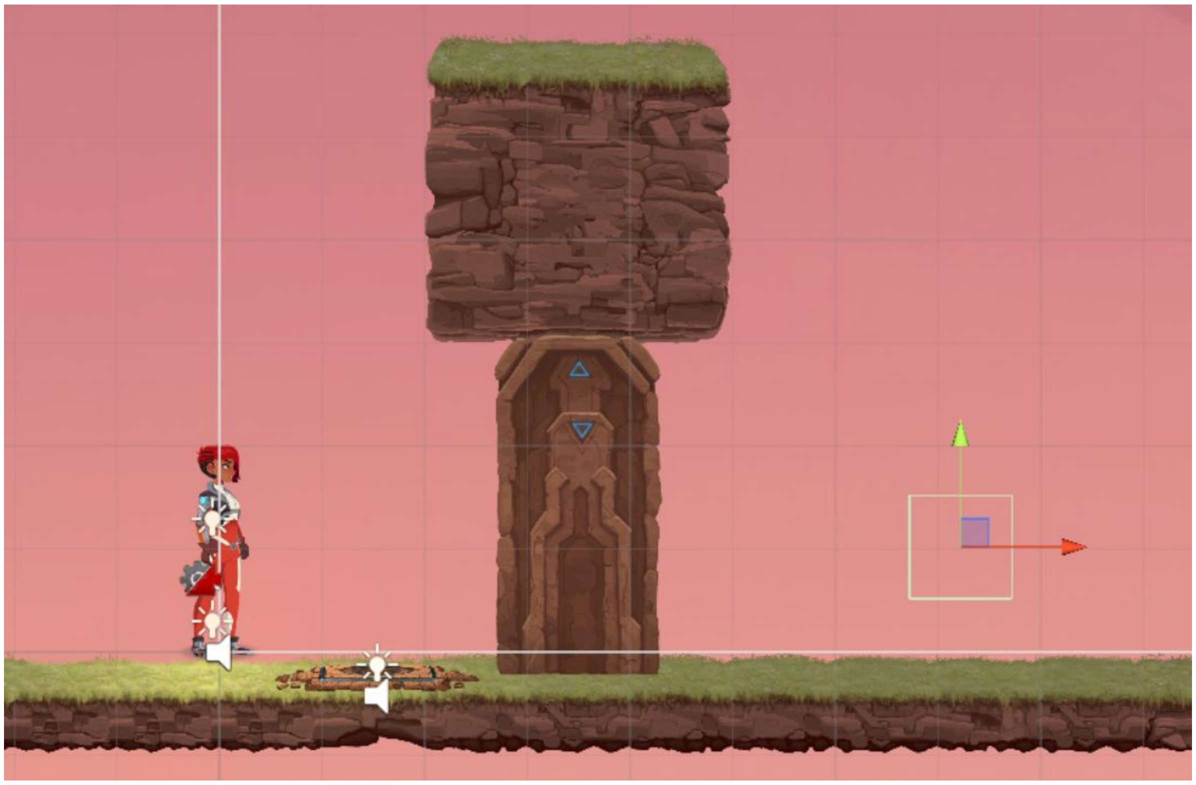


Teleporting the Player

It is possible to teleport a player from one area in a scene to another, or between different levels.

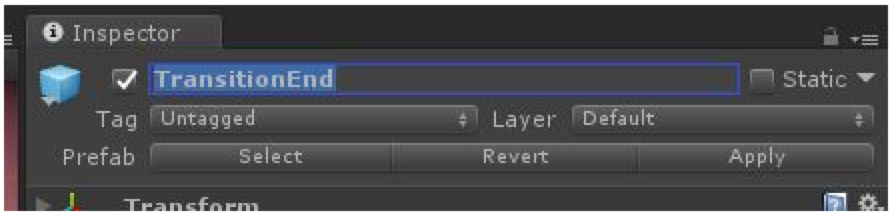
Teleporting within a Scene

To teleport the player within a scene, we will setup a **transition**.



First we need to setup the starting point of the transition:

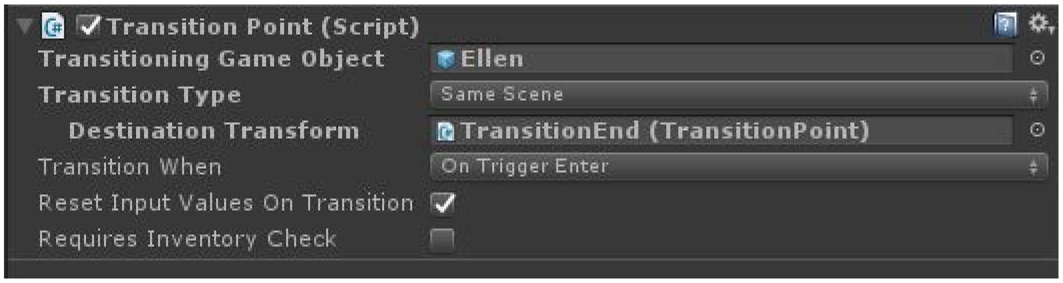
- Go to **Prefabs > SceneControl** in the Project folder
- Find the **TransitionStart** Prefab
- Drag TransitionStart to the **Scene View** (for example behind the door)
- Place it at a position where the player will collide with the Collider (the green box) when walking


Inspector
TransitionEnd
Static
Tag: Untagged
Layer: Default
Prefab: Select Revert Apply
Transform

To set up the destination;

- Drag another **TransitionStart** Prefab from the **SceneControl** folder into the **Scene View**
- In the Inspector, rename this to **TransitionEnd**

Now let's link the two together;


Transition Point (Script)
Transitioning Game Object: Ellen
Transition Type: Same Scene
Destination Transform: TransitionEnd (TransitionPoint)
Transition When: On Trigger Enter
Reset Input Values On Transition: ✓
Requires Inventory Check:

- In the **Hierarchy**, select the **TransitionStart** object
- In the **Inspector** find the **TransitionPoint** Component
- Find the **Ellen** GameObject in the **Hierarchy**
- Drag the **Ellen** GameObject from the Hierarchy into the **Transitioning Game Object** slot of the **Transition Point** Component.
- Set **Transition Type** as **Same Scene**

This has ensured that Ellen will be the only object teleported and that it within the same scene. Now let's set the destination;

- In the Hierarchy, find the **TransitionEnd** you made earlier
- Click and drag **TransitionEnd** into **DestinationTransform** in the **TransitionPoint** Component
- Set **Transition When** to **On Trigger Enter**

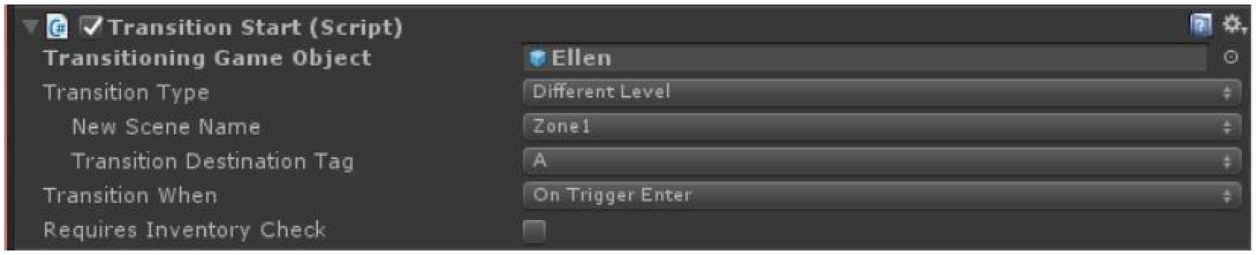
On Trigger Enter means that the Transition only activates when the player enters the Collider and not on a key press (if you want to teleport only when the player press the interact key (e), just set that to **Interact Pressed**)

Teleporting to another Scene

To make the player transition to a new level, the same prefab is used. But instead of setting it to send the player to another point in the same level, we set it to send them to a **TransitionDestination** into another level.

To simplify the setup, let’s make the player transition to the 1st level of the game. In your **TransitionStart** component, change the settings to :

- Set **Transition Type** as Different level
- In **New Scene Name**, choose a scene to send the player. Here, choose Zone 1.
- Set the **Transition Destination** to **A**
  All transition destinations have an associated letter. Selecting **A** will send you to the Destination with the tag **A** in the Zone1 scene.



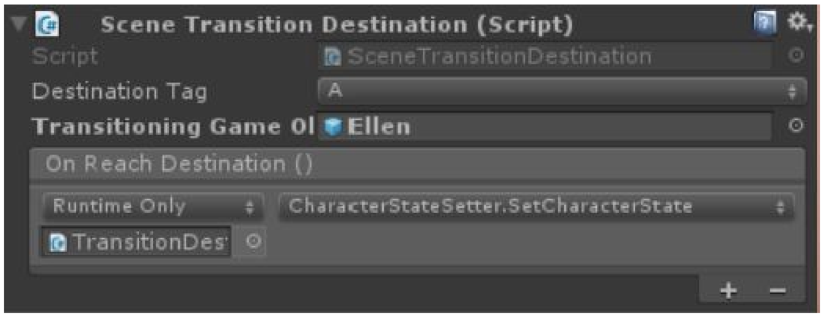
Transition Start (Script)

Transitioning Game Object	Ellen
Transition Type	Different Level
New Scene Name	Zone1
Transition Destination Tag	A
Transition When	On Trigger Enter
Requires Inventory Check	☐

Now that you have set up your basic scene, press Play, and walk to where the transition is and you should travel all the way to the start of Zone1!

Transitioning to your own Scenes

If you want to add transition to another scene you created (let’s say you called it **MyScene2**), just find the prefab **Prefabs** < **SceneControl** < **TransitionDestination** and place it in **MyScene2**, select that scene in the **Transition Destination** drop down on your **TransitionStart.**
Don’t forget to check if it’s added to your build setting (File> Build Settings) otherwise it won’t appear in the drop down!



```
Scene Transition Destination (Script)
Script                      SceneTransitionDestination
Destination Tag             A
Transitioning Game Ol       Ellen
On Reach Destination ()
Runtime Only                CharacterStateSetter.SetCharacterState
TransitionDes
```

Drag Ellen in the **Transitioning Game Object** slot, and to set the same **Destination Tag** on both the **TransitionStart** and the **TransitionDestination**

Having Fun

Most objects within the kit play with the system of events seen in the Pressure Pad setup.

Use the existing scenes (Zone 1 to 5) to see how some other types of objects are set up using events and triggers.

Refer to the Components Documentation of the project, which lists all the components and their parameters in detail.

Happy level designing!