



Pixel Platformer - Responsive 2D Platformer Toolkit V1.2

User Guide

1) Introduction

The Pixel Platformer asset seeks to help in the creation of any platformer game offering most of the common & advanced features you will need while also focusing on making your platformer game "Feel" Good using different advanced techniques learnt from other successful games such as Super Meat Boy, Celeste, TowerFall, etc. Besides offering full tilemap support (unity 2017 or higher) and a professional pixel-art tileset and character allowing the game designer to create carefully hand-designed levels in a matter of minutes.

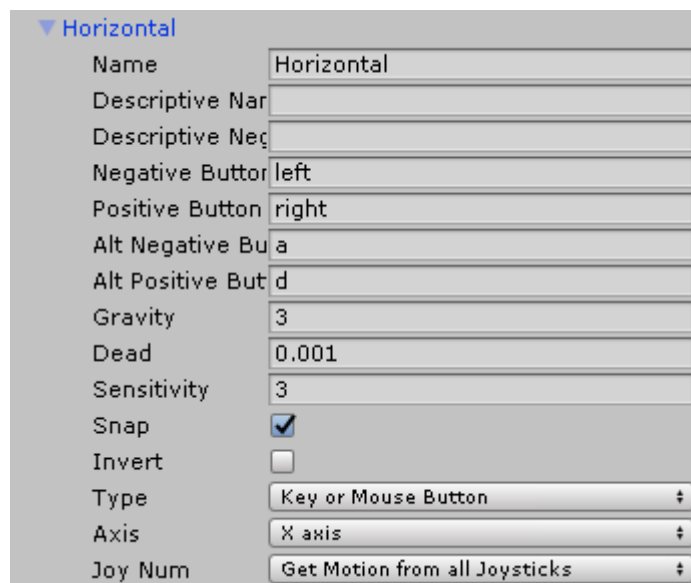
2) Setup Guide

-Create a new unity project and import Pixel Platformer - Responsive 2D Platformer Toolkit Package into the project.

-Create four layers in your project named Solid, OneWay, Bumper and Ladder. These will be used to mark the level tiles and prefabs that the player or actors can: be blocked by (solid), pass-through (OneWay), the bumpers which make the moving platforms/blocks change direction and the ladders. The creation of these layers is essential for the sample projects to work.



-Go to the Unity Input Manager (Edit/Project Settings/Input) and setup/modify 6 Axes: Horizontal (can use the default one), Vertical (can use the default one), Jump, Dash, Attack and Shoot. These are necessary for the Player component to work correctly. Mine are setup to work with keyboard only so if you use controller you'll need a different setup on the axes.



▼ Vertical

Name	Vertical
Descriptive Name	
Descriptive Negative	
Negative Button	down
Positive Button	up
Alt Negative Button	s
Alt Positive Button	w
Gravity	3
Dead	0.001
Sensitivity	3
Snap	<input checked="" type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

▼ Jump

Name	Jump
Descriptive Name	
Descriptive Negative	
Negative Button	
Positive Button	space
Alt Negative Button	c
Alt Positive Button	
Gravity	1000
Dead	0.001
Sensitivity	1000
Snap	<input type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

▼ Dash

Name	Dash
Descriptive Name	
Descriptive Negative	
Negative Button	
Positive Button	x
Alt Negative Button	
Alt Positive Button	
Gravity	1000
Dead	0.001
Sensitivity	1000
Snap	<input type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

▼ Attack

Name	Attack
Descriptive Name	
Descriptive Negative	
Negative Button	
Positive Button	z
Alt Negative Button	
Alt Positive Button	
Gravity	1000
Dead	0.001
Sensitivity	1000
Snap	<input type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

▼ Shoot

Name	Shoot
Descriptive Name	
Descriptive Negative	
Negative Button	
Positive Button	v
Alt Negative Button	
Alt Positive Button	
Gravity	1000
Dead	0.001
Sensitivity	1000
Snap	<input type="checkbox"/>
Invert	<input type="checkbox"/>
Type	Key or Mouse Button
Axis	X axis
Joy Num	Get Motion from all Joysticks

For more info on how the Input Manager works or how to set these to be used with controllers instead of keyboard visit the [Conventional Game Input Unity Manual](#)

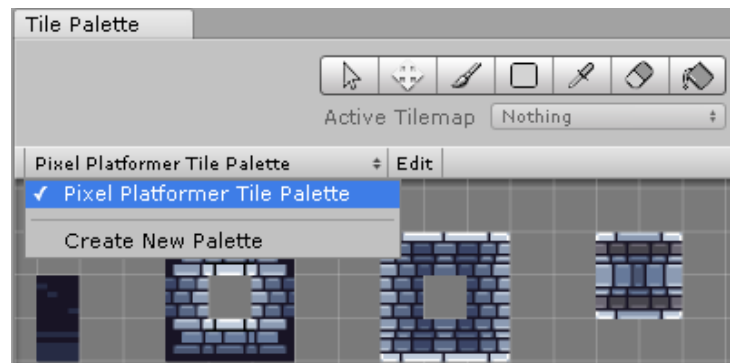
3) Create a level using TileMap

- Create a new Scene and set the camera's size to the size you want, remember the size of the camera times 2 is the amount of vertical pixel that will fit into the screen (72 or 108 are the recommended sizes with this tileset).

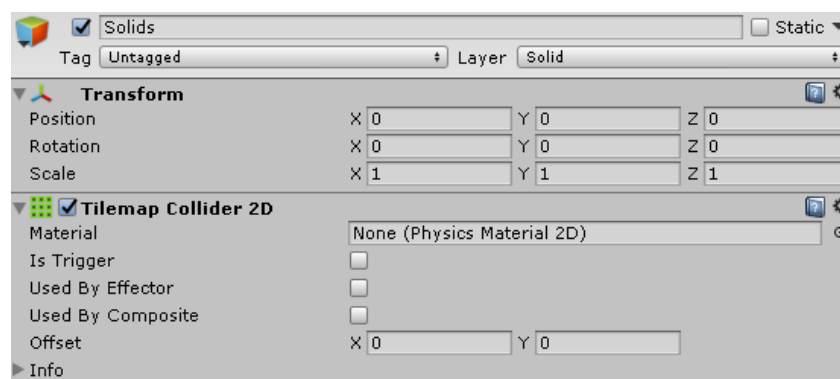
- Before going further it is recommended that you follow along all unity's Live Session tutorials about TileMaps, Knowing how to use unity's Tilemap and all its features is a requirement to work with the asset pack:

<https://unity3d.com/learn/tutorials/topics/2d-game-creation/intro-2d-world-building-w-tilemap>

-Open the Tile Palette Window (Menu Item Window > Tile Palette), inside the tile palette window you should have a palette called "Pixel Platformer Tile Palette" pick it, this is the palette that contains the tileset tiles.

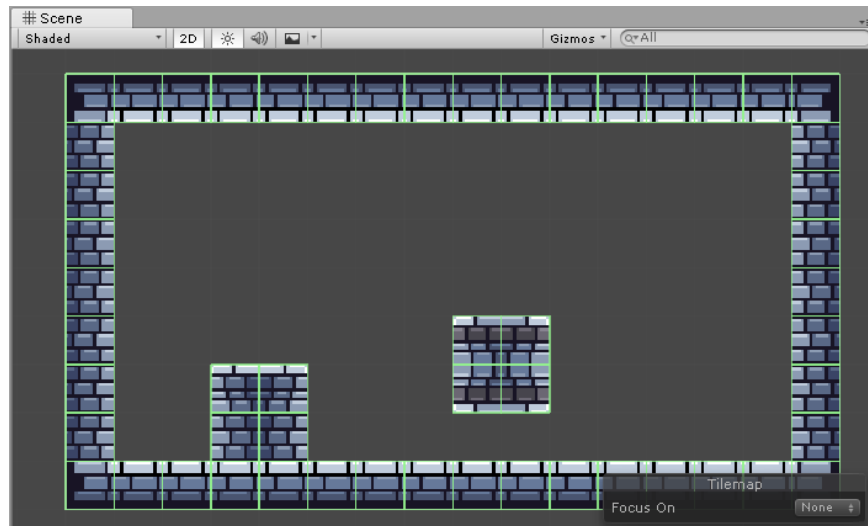


-Create a new Tilemap (Menu Item GameObject > 2D Object > Tilemap), Set Tilemap Grid cell size to 16,16,0 (X ,Y, Z) and name the first tilemap "Solids", add a tilemap collider 2d component to it and set it's layer to the Solid Layer, This is the tilemap where you will place all solid non-moving objects that the actors will collide with such as walls and floor tiles (do not place anything besides solids non-pass-through and non-moving you want the player to collider with in this layer).

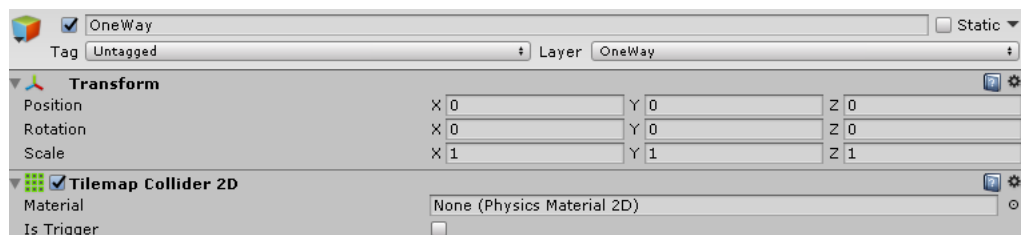
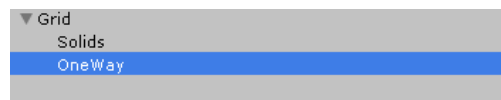


-Start adding all solid tiles to the level using the tile palette and make sure the Active Tilemap in it is the Solids Tilemap you just created.

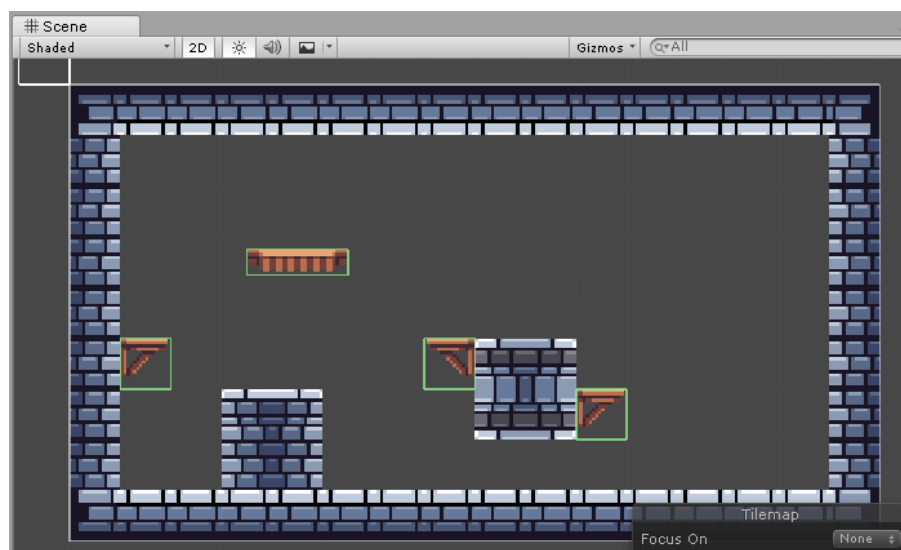




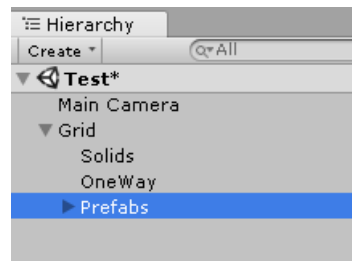
-Create a new Tilemap and name it "OneWay", add a tilemap collider 2d component to it and set it's layer to the OneWay layer you created earlier, this is the tilemap where you will place all static pass-through/one-way objects such as static pass-through platforms.



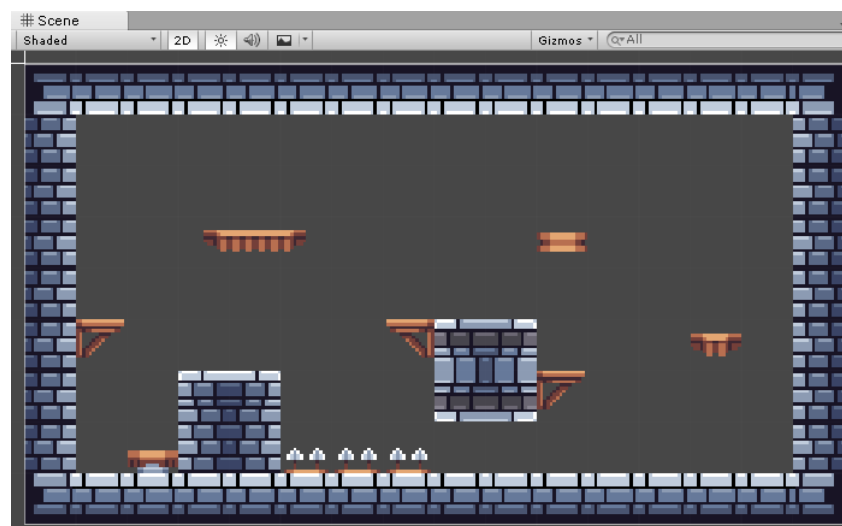
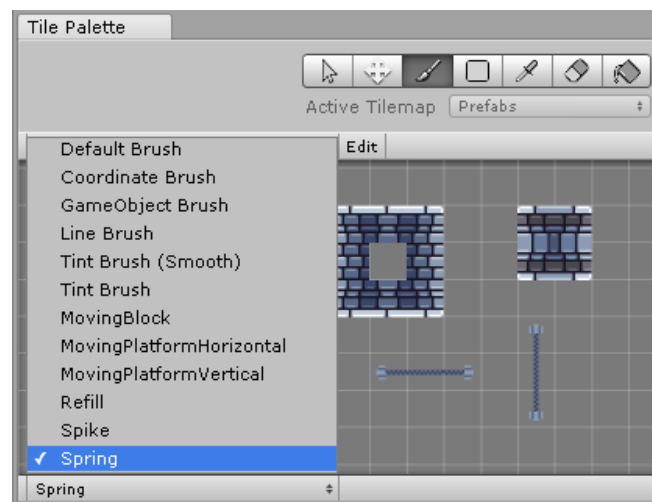
-Add all pass-through platform tiles you'd like to add to the level using the tile palette but first make sure the active Tilemap in the palette is the OneWay Tilemap you just created.



-Create a new Tilemap and name it "Prefabs", don't add any type of collider or layer to it (leave it on the default or whichever layer you want that is not any of the 3 you created before). This is the Tilemap where you will place all the prefab tiles which include (moving platforms/blocks, springs, spikes and the dash refill).

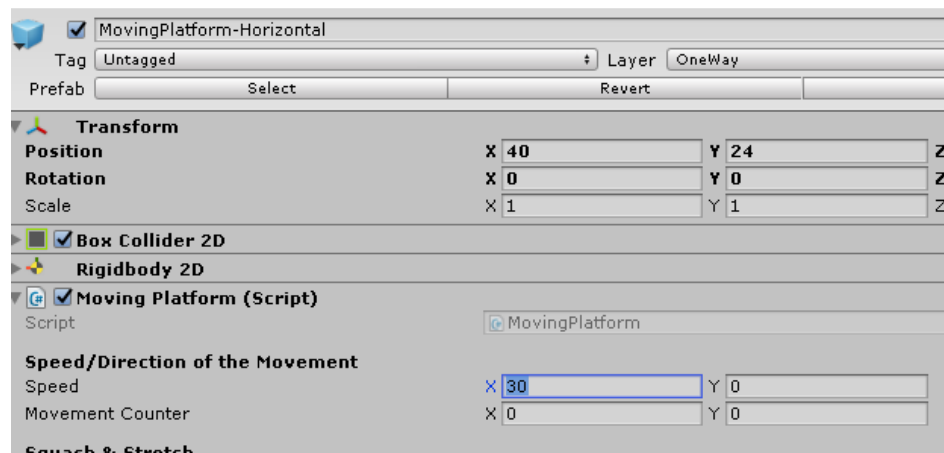


-To start adding the prefab tiles set the activate Tilemap on the tile palette to the "prefabs" tilemap you just created, and change the default brush to the prefab brush of the prefab you'd like to add to the scene there are: moving platforms vertical & Horizontal, moving blocks, springs, spikes and the dash refill.

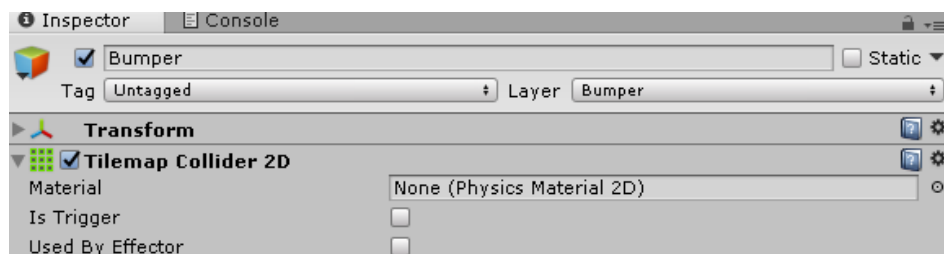
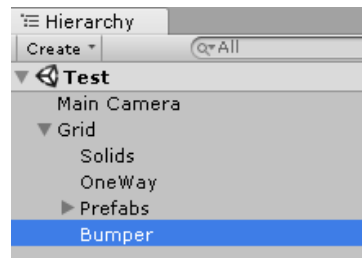


-You can hit the play button now but you'll see that the moving platforms, if you placed any on the scene, move infinitely and don't turn around. That is because we still need to add bumpers to the scene which are objects that make moving platforms or blocks turn around when they collide with them.

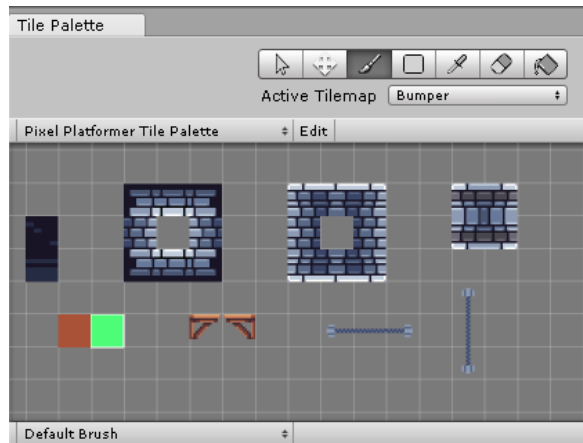
-You can change the direction or speed of any moving block or platform by looking for it on the Childs of the "prefabs" tilemap you created and changing the Speed values (x for horizontal y for vertical).



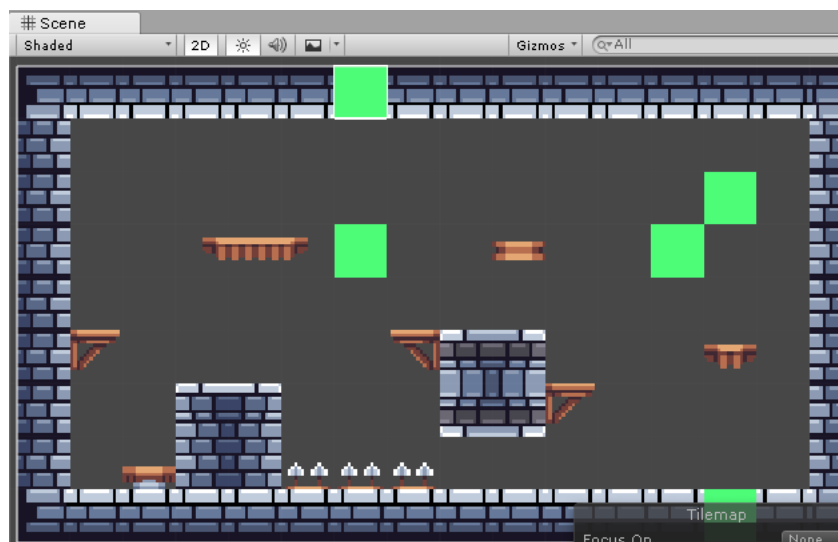
-Create a new Tilemap and name it "Bumpers", add a tilemap collider 2d component to it and set its layer to the Bumper layer you created earlier, this is the tilemap where you will place all the bumpers to make moving platforms and blocks turn around.



-Don't forget to swap the brush back to the default one and change the tile palette's active tilemap to the bumper tilemap.



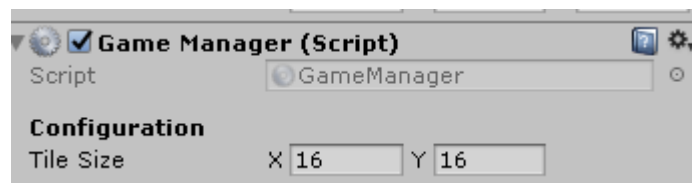
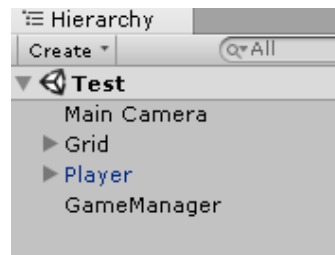
Pick any tile you'd like to use for the bumper, use the green square so it is very noticeable where bumpers are when editing the level, and place bumpers accordingly so when the moving platforms or blocks collide with the bumper they change direction (turn around). Also remember to set the bumper tilemap order in layer to a lower value so the bumpers are hidden behind the rest of the level after you've finished editing your level.



-Ladders: Just like with previous steps create a new tilemap for ladders set it's layer to the ladder layer you created before and before drawing ladders tile into the scene make sure that the active tilemap on the tile palette is the ladders tilemap you just created and start adding ladder tiles to the scene (present in the V1.2 of the Tilemap).

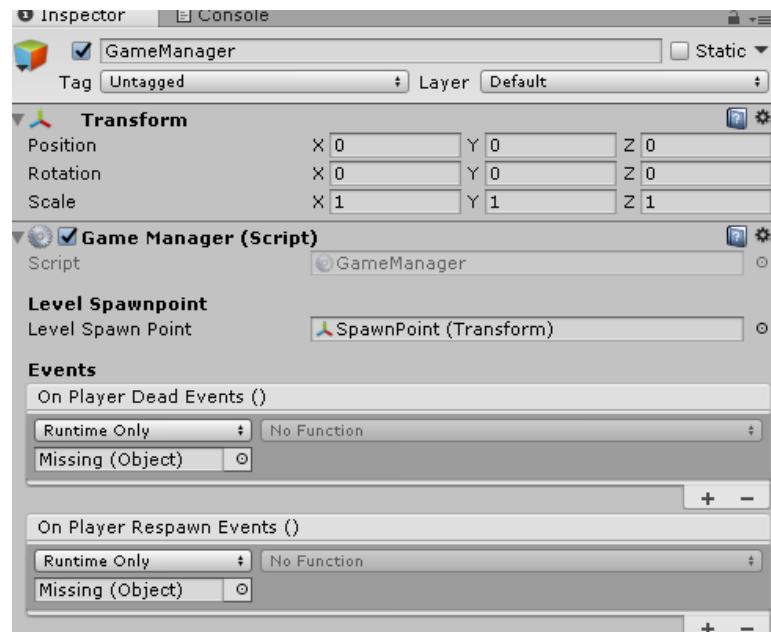
-Drag the GameManager and the Player prefab from MooseStache/Pixel Platformer/Prefabs folder into your scene. For now the only thing you need to change on the game manager is the Tile Size field on the Game Manager script attached to it (the tiles that come with the engine are 16 pixels Wide and 16

pixels Width) do not change any other of the values that these components come populated with (once you get a hold of how the asset works you can go your merry way altering any and every thing).

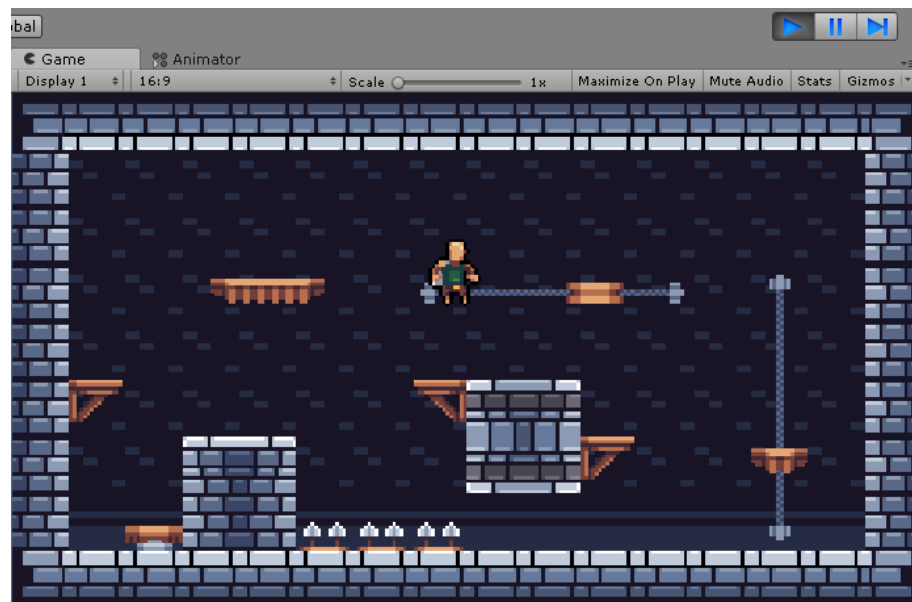


-Create an empty GameObject, call it "Spawnpoint" and move both it(spawn point) and the player to where you'd like the player to start the level and respawn after dying, make sure the X and Y position of these objects after placing them is an integer since movement is pixel perfect (1 pixel = 1 Unity Working Unit because of the pixels per unit on the sprites is set to 1 in this asset) and placing them in float values could cause problems with the movement, assign the spawn point to the Level Spawnpoint field in the GameManager.





-You can play the game perfectly now but you might want to add background and decorative items to the scene. These should be placed on a different tilemap with no colliders and with a lower order in layer value than -1 (player is -1) and it is recommended to use 2 different Tilemaps one for the background and one for decorative items like the rails for the moving platforms, with the background tilemap having a lower value than the decorative tilemap.



Congratulations!!

If you've got this far you should have a fully working level with respawn on death, and all features fully working. If you need more info on how everything works please refer to the scripts as they are very well commented and explained or check out the example scene inside the Moose Stache/Pixel Platformer/Scenes folder.

Thank you for buying **Pixel Platformer - Responsive 2D Platformer Toolkit** if you wish us to add any feature, have any suggestion or found a bug don't doubt letting us know and we'll get on it ASAP! And If you think the asset is good help us spread the word about it and together we can make it greater!

Also please rate the asset on the asset store and a review if you have the time, this helps us greatly and we appreciate it from the bottom of our hearts.

Contact:

Forum: <https://forum.unity.com/threads/released-pixel-platformer-engine-responsive-2d-platformer-toolkit.533964/>

Email: moose_stache@hotmail.com

Pixel Platformer Engine Changelog:

Features **V1.0:**

- **Pixel perfect collisions & Movement support.**
- Basic player movement (**idle, run, jump, wall jump & slide**).
- Advanced player movement (**variable jump, dash, fast-fall**).
- Responsive Controls (**jump buffer, jump grace time, wall sticking, Squash & Stretch**)
- One-Way platforms.
- Moving One-Way platforms & Blocks (horizontal & vertical).
- Crush entities with moving blocks.
- Springs & Spikes.
- Falling platforms.
- Level Entry & Spawn Point.
- Pickups Example: Dash Refill/Refresh.
- Screenshake.
- Basic particle examples (dust effects).
- Player death with room fade-in/out.
- Fully working **professional pixel art tileset & character** with animations.
- Example Scenes created with Unity's built in **TileMap** & 2D Extras.

Features **V1.1:**

- **Modular Health System**
- Melee Attack (**hitboxes**)
- Ranged Bow Attack (**projectiles**)
- Basic Enemy with AI (worm)
- **Sprite Flashing**
- 2 new Shaders (Sprite flashing & Sprite Outlines)
- Traditional UI with Hearts/lives health bar
- Falling Platforms
- **Slopes**

Features **V1.2:**

- **New Full Tilemap** with a lot of new tiles to play with and extended background
- **Several bug fixes**
- Extended the Collision Detection to support Collision Points and Collisions At Place (actual position + extra added position).
- **Drop from one-way platforms** (Press down + jump button to drop down from a one-way platform)
- **Crouching** / Crouch Slide (Customizable friction while in the sliding state)
- **Ladders** (Up and Down)
- **Ledge Grabbing**
- **Ledge Climb/Mount** (Climb a ledge after grab)
- Objects Push (**Push Blocks**)

- **Zoner AI** (Crawl up/down/left/right on walls)
- **Checkpoints**
- Pickup Example (**health potion**/heart)
- Improved and Updated Documentation
- **Smooth Camera Follow** example

As you might see by the changelog **V1.2** was a really **big update** and we hope you enjoy it, if you have any suggestion or want us to add anything on **V1.3** don't doubt contacting us.

All the best, **Moose Stache** Team.