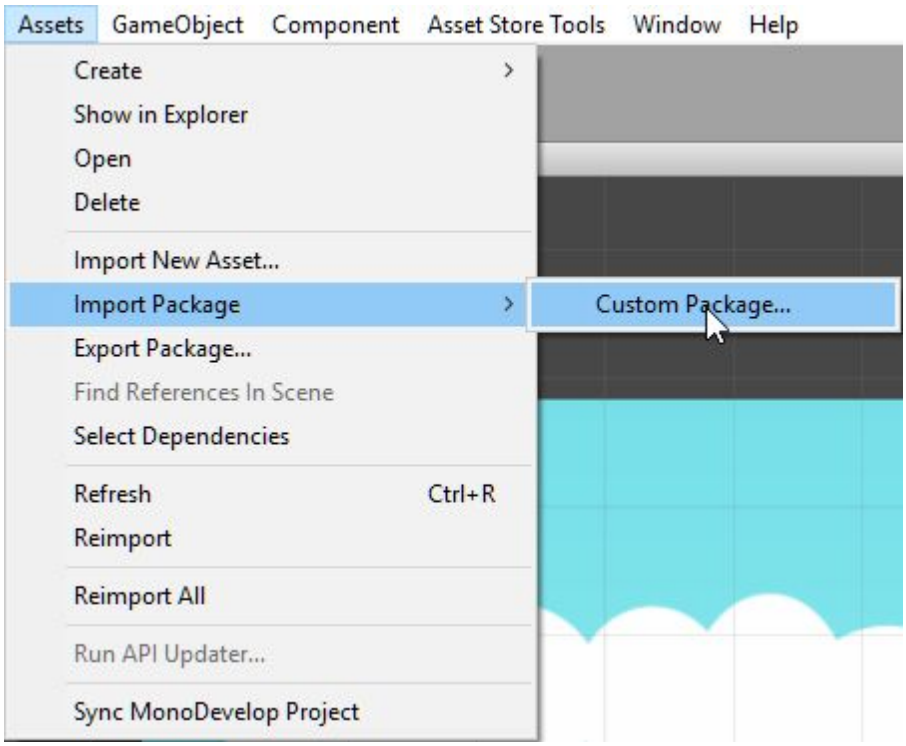
Pixul Physics 2D - Quick Setup

Version 1.0

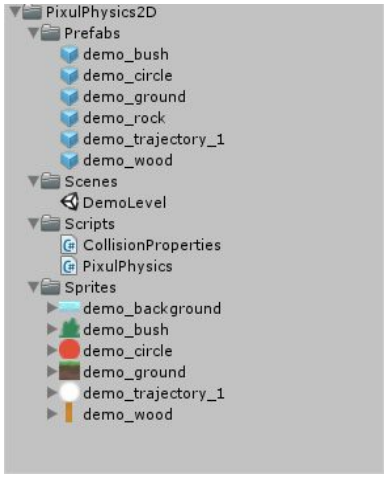
This document contains step-by-step instructions on how to quickly get Pixul Physics 2D running in your own 2D Unity project using our default settings. We recommend you check out the demo scene included in the package.

Step 1: Importing.

Import the PP2D package into your project by going to Assets > Import Package > Custom Package and browsing to your PP2D download location.

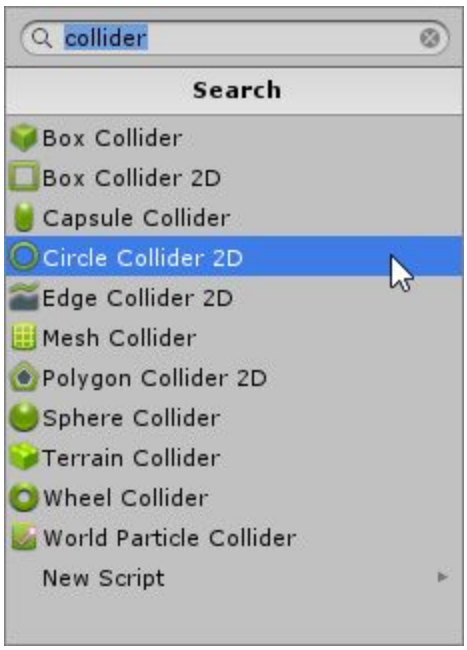


Expand the PP2D folder in your project's hierarchy view to see everything that comes with the package.



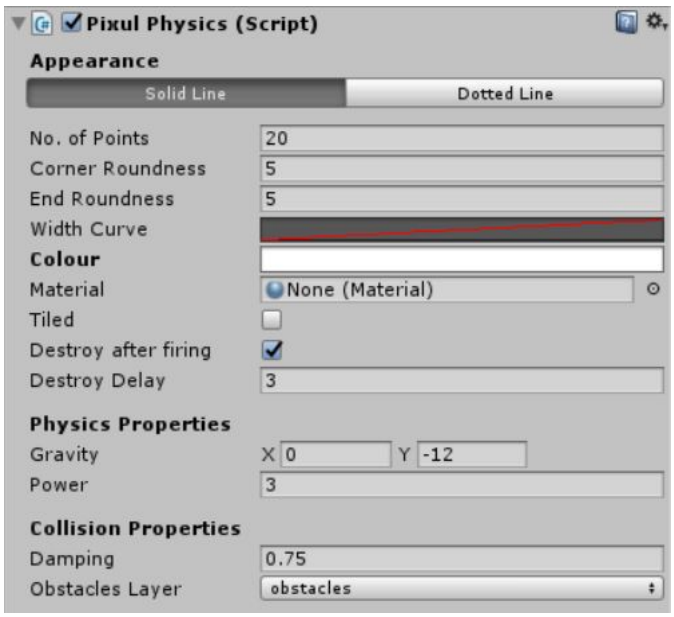
Step 2: Colliders.

Next, we'll set up your throwable object. The object that you want to throw requires a collider 2D. To add this, select your object and click Add Component in the Inspector. Search for "collider" and select the option you need. In the case of the demo we used "Circle Collider 2D".



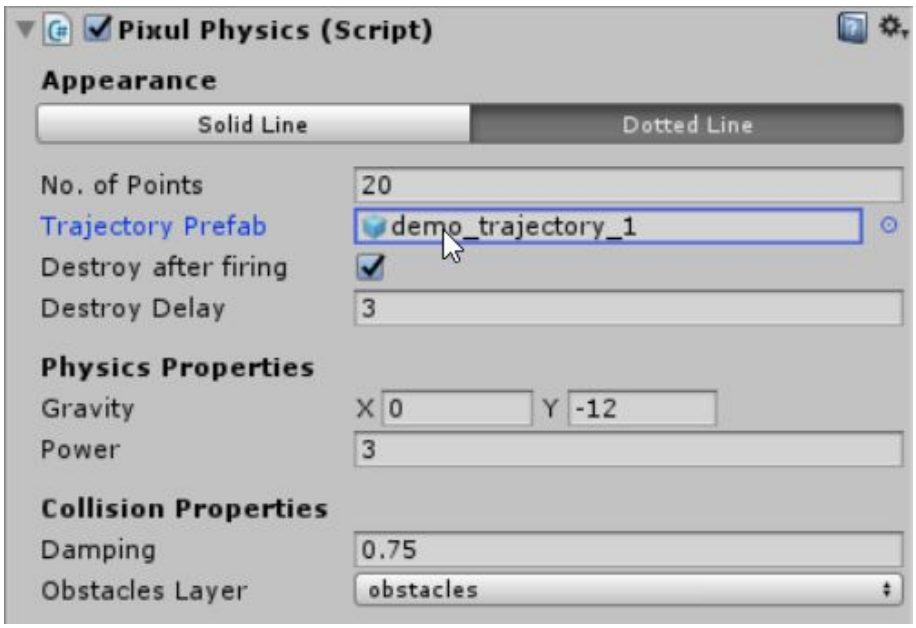
Step 3: Applying PixulPhysics2D.

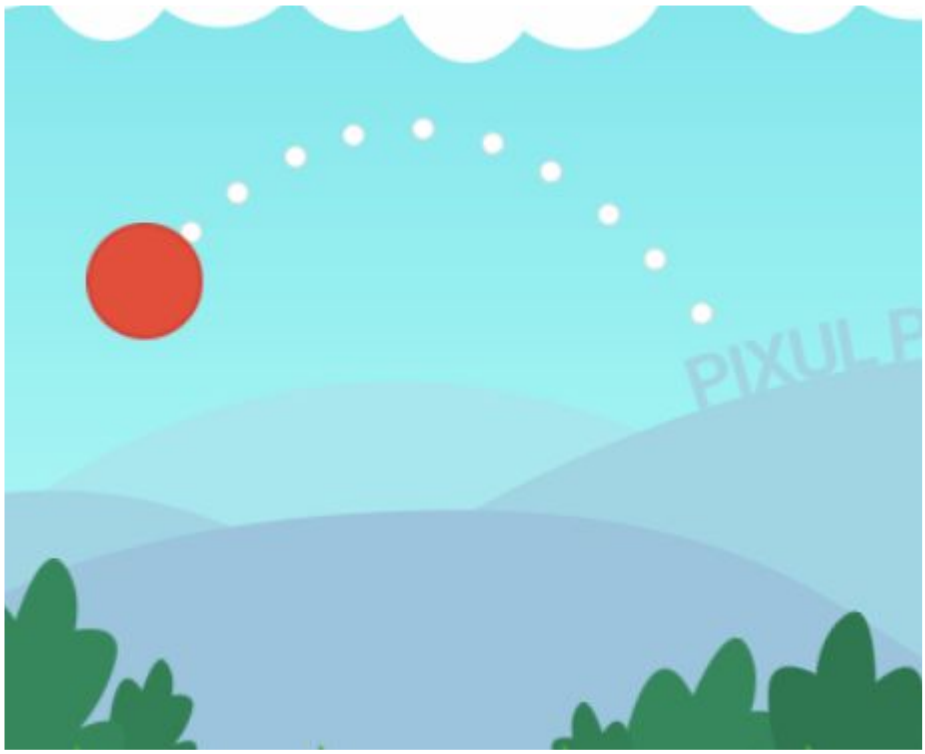
Still with your throwable object selected, drag the “PixulPhysics” script onto the inspector. This is located in the “PixulPhysics2D/Scripts” folder. This will add the physics functionality to your object.



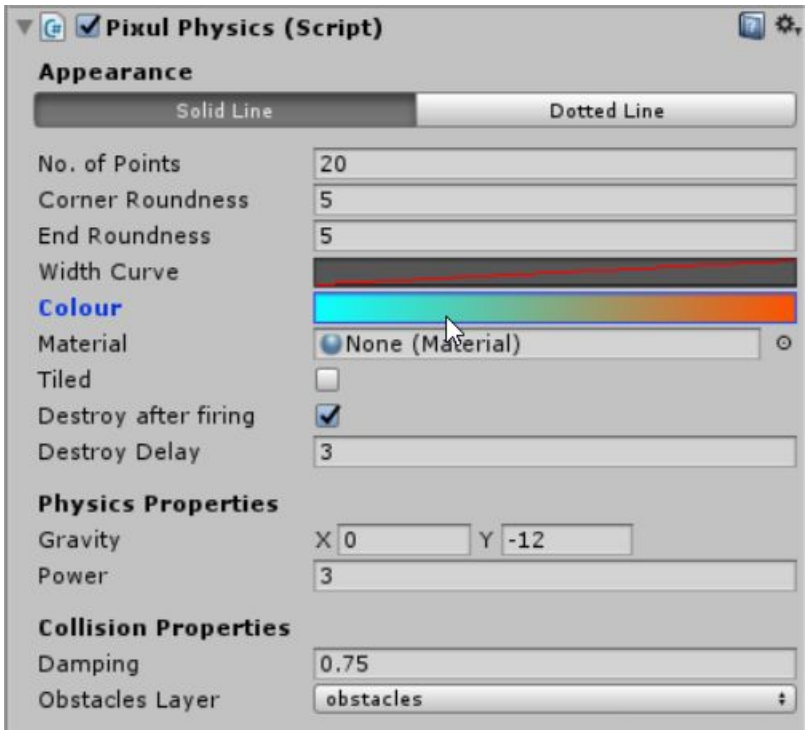
Step 4: Trajectory.

The PixulPhysics script requires some settings before it will work correctly. First, our trajectory settings. Let’s add a prefab to use for our dotted line. Drag and drop “demo_trajectory_1” (located in PixulPhysics/Prefabs) into the “Trajectory Point Prefab” slot in the script. The “No. of Points” setting changes the number of points displayed in your trajectory. Run your project and click and drag on your throwable object to preview the trajectory.



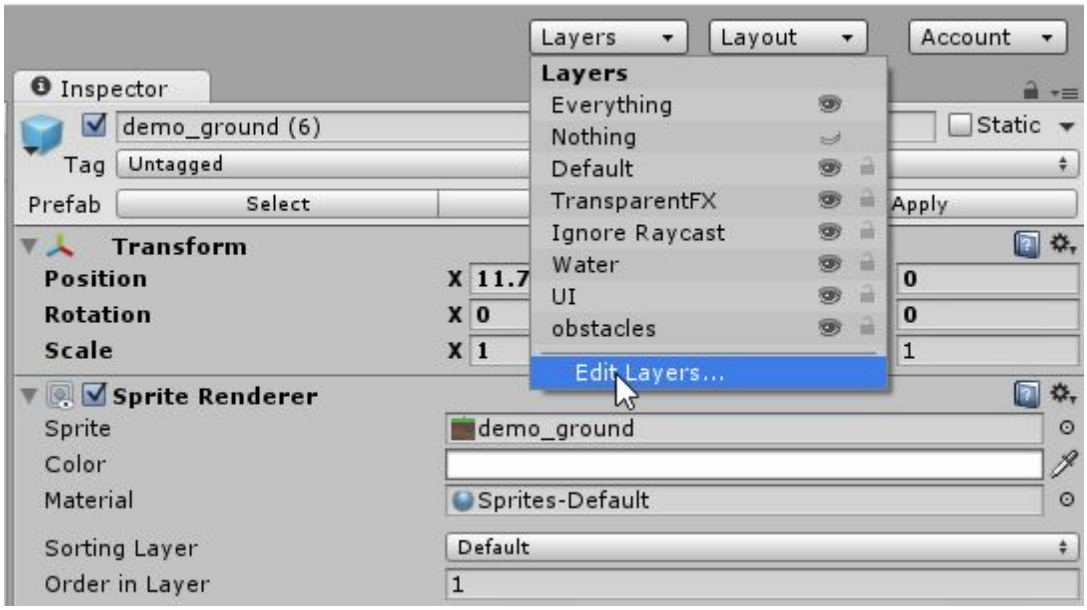


If you would like a solid line simply choose the “Solid Line” option in the PixulPhysics script and set the desired options such as colour and width. (**Note:** Leaving the material to ‘none’ will use the default sprite material).

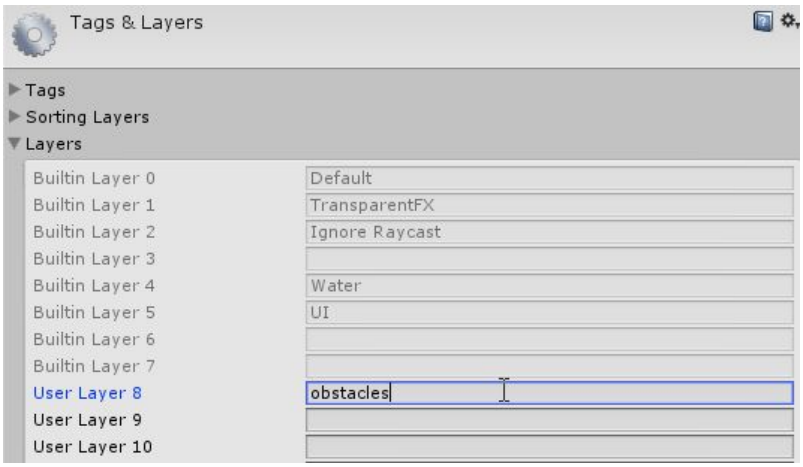


Step 5: Obstacles.

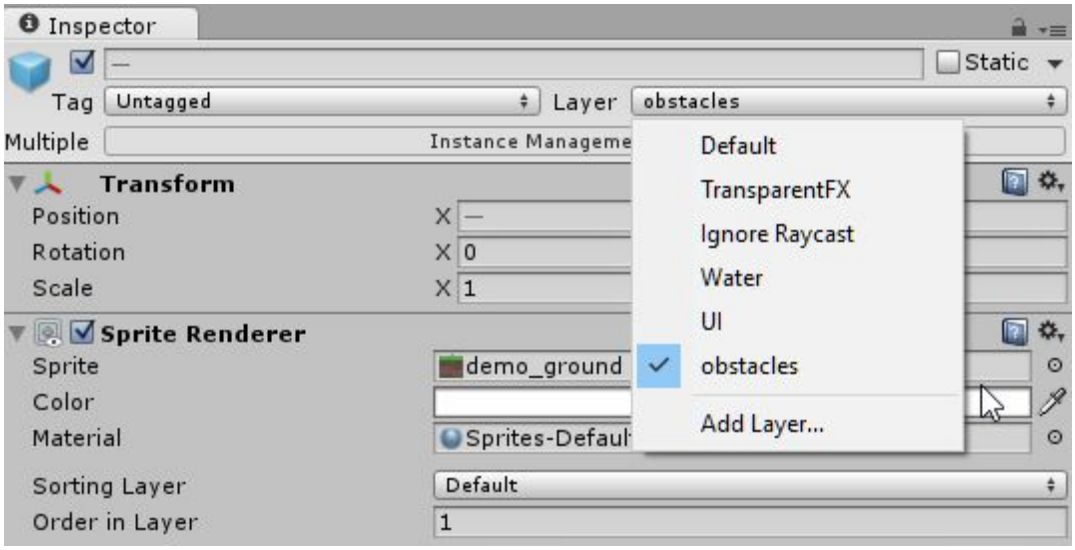
Now let's set up our obstacles layer. This is anything you want your throwable object to bounce off. In our demo, we have the demo_ground objects assigned to the "obstacles" layer. To make a new layer go to Layers > Edit Layers in the top right of your Unity window. Alternatively, you can go to Edit > Project Settings > Tags & Layers, and expand Layers.



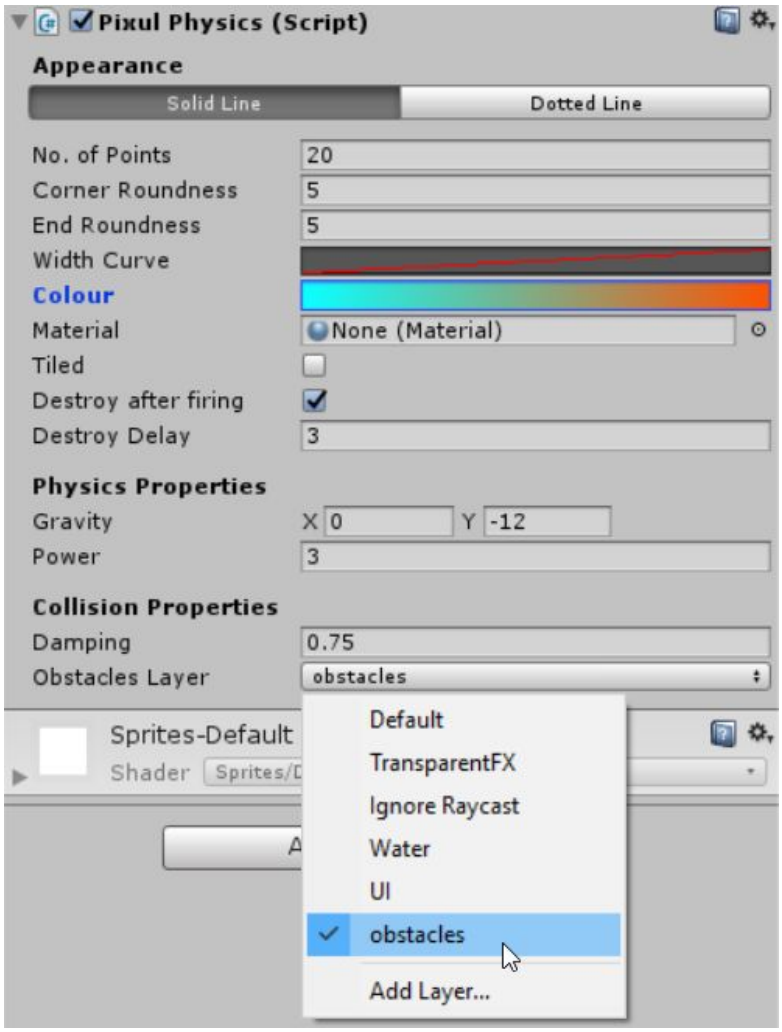
In a User layer slot, name your layer something sensible like "obstacles".



Select all of the objects you wish to have in this layer and click the Layer dropdown at the top of the inspector. Select your obstacles layer.



Select your throwable object and set the “Obstacles Layer Mask” at the bottom of the PixulPhysics script to be your obstacles layer.



Your throwable object and real-time trajectory will now interact with these objects. Yay!

You're done! That's it for quickly getting PixulPhysics2D running in your 2D Unity project using our default settings.

Thanks for reading.

The Pixul Team.

Need support? Want to give feedback or request features? Please don't hesitate to email us at contact@pixul.io and we'll get back to you as soon as possible.

Want to tweak PixulPhysics2D's settings? Want to dive into the code? Please check out the reference document included in the package for a detailed breakdown for all your tweaking needs.