



SINGLE-SCRIPT PARALLAX DOCUMENTATION

This documentation is current as of October 9th, 2018.

For a video walkthrough of setting this up, [click here](#).

Have questions? You can reach us at contact@dossamer.io.

SETUP

Make sure your camera is orthographic, and that its near-clip distance is a value like negative 1000. This is so that you can position parallax elements ahead of the camera in addition to just behind it.

Once your camera is set up, drag a few background sprites into your scene. To make each sprite parallax-enabled, add to each the component **ParallaxBehavior**.

Give the sprites varying depths by changing their positions on the z-axis. Moving them closer to the horizon point (represented by a yellow rectangle in the scene view) will make them move more slowly relative to the camera.

Once you've positioned the sprites, you're done! You can fiddle with the horizon point depth if you want, or override the camera used to perform the parallax calculations, but you can also just press play, and everything should work.

A nifty advantage of this system is that it lets you set up parallax **layers**. Make an **Empty**, add the **ParallaxBehavior** component, and give it some children. (Like a particle system, a 3D model, or something animated.) All of the children of that **Empty** will move together as a single parallax layer. This means you can build complex scrolling layers out of various types of distinct components.

A not-so-nifty disadvantage of this system is that because it relies on manipulating object transforms directly to pull off the parallax effect, you can only render one camera at a time. (So no split-screen.) Parallax elements can't have two positions at once, so they can't be rendered correctly by two separate cameras.

That's about it! Happy parallaxing.