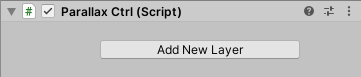
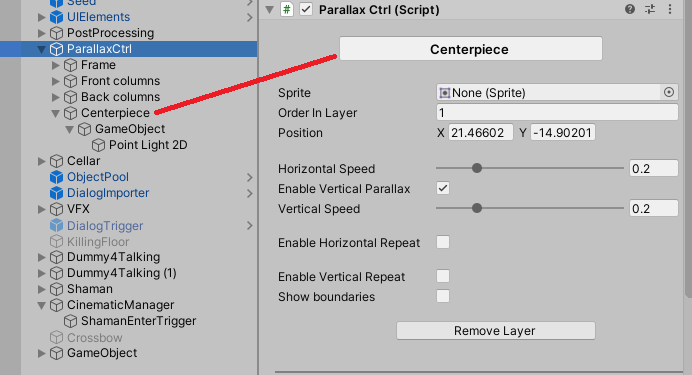
Parallax Controller (Parallax Ctrl)

*Parallax Controller* is the old script. The new one is called *Parallax Ctrl*. A custom editor was made for it. While empty, it has only one button to add a new layer:



Once the button is clicked, a new game object is created as a child of ParallaxCtrl. By default it has a sprite renderer, but you could also leave it empty and just use it as a container for other game objects as shown in the following picture:

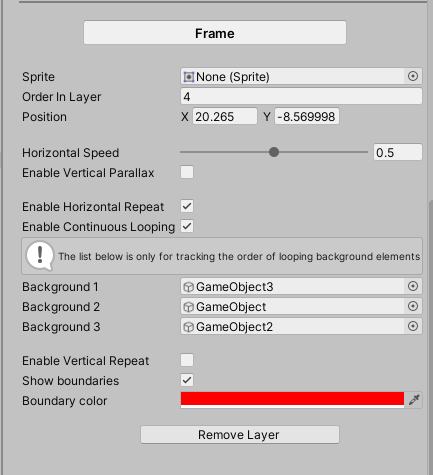


Do not add/edit/delete children of ParallaxCtrl directly! The custom editor should be used for that. You can, however, manually add children to each layer (in the above picture, Centerpiece’s children *GameObject* and *PointLight2D* have been added manually).

At the top, you can edit the layer’s name. Optionally, you can choose a sprite and the order in layer. The position has to be changed in the editor as well, instead of using the transform options as this position will override the layer’s transform.

There are options for the horizontal and vertical speed of the layer (vertical speed can be disabled if it is not necessary).

If HorizontalRepeat is enabled, new options will appear:



HorizontalRepeat means that the layer will reappear on the other side of the screen once it reaches a certain boundary. Instead of repeating the same image, ContinuousLooping can be enabled which requires three images that will be in a loop.

The same goes for VerticalRepeat, although VerticalLooping is not implemented yet, only repeating the same image.

For debugging and testing purposes, ShowBoundaries can be enabled to see where is the line that an image has to cross in order to appear on the other side of the screen.

The layers will follow the camera at the selected H and V speeds.

TODO & improvement ideas:

* Add vertical continuous looping
* Layers that do not repeat, will keep moving even when they are far away from the player (and thus not visible to the player). This could be optimized bearing in mind that the player can always decide to go back requiring to show those layers properly again (hence, destroying them is not an option).
* Testing can mess up the layers’ positions currently. The layers follow the camera but if the player is placed elsewhere in the level, the camera will not follow while in editor mode but only when we play the game. Since the camera makes a big leap in that case, the layers get messed up. If in the editor both the player and the camera are moved to a new position, the layers will work correctly.