FRUSTUM CULLING PACKAGE DOCUMENTATION

"Thank you for buying this package and trusting Pathiral"

Find more packages from Pathiral here

ABOUT

Disable game objects completely that're out of the camera's view (frustum).

P.S: Unity does *Frustum Culling* by default under *Occlusion Culling* for objects/prefabs but it disables the renderers only, but you'll need a custom solution to completely disable the object. This is where this package comes in. This comes extremely helpful to disable objects that are very costly to have on but you need them enabled when in view. Things like an NPC standing still, audios maybe, small lights, etc..

GETTING STARTED

1. Add a simple cube game object to your scene
2. Move the object so that it's in front of your game camera
3. Add the *FrustumCulling* script to the cube
4. Drag and drop your camera to the **MainCam** property
5. Click on the **Build Objects** button to make the tool build all the game objects. This is all done in editor-time thus improving performance drastically on game start.
6. Play the game
7. Now move your camera away from the object **OR** move the object away from the camera's field of view. In both cases the same thing happens. The cube gets disabled.
8. That's it!

PROPERTIES & METHODS

autoCatchCamera -> (bool) set whether you want the tool to automatically get the active main camera on game start or not. Might increase game start time if too many objects use it.

mainCam -> (Camera) set the main camera of the scene manually

cameraLeftPad -> (float) set the value of the left padding of the camera. Game objects will disable when they exceed this padding.

cameraRightPad -> (float) set the value of the right padding of the camera. Game objects will disable when they exceed this padding.

cameraTopPad -> (float) set the value of the top padding of the camera. Game objects will disable when they exceed this padding.

cameraBottomPad -> (float) set the value of the bottom padding of the camera. Game objects will disable when they exceed this padding.

autoBuildObjects -> (bool) set whether you want the tool to automatically build all the placeholder-gameobjects on script awake or not. Might increase game start time if too many objects use it.

distanceCull -> (bool) set whether you want the distance to be taken into consideration. Distance culling only happens when the object is first outside the view.

Example for **distanceCull**: *if you look at an object to be culled and keep going further and further away it won't disable even when the distance (distanceToCull) is exceeded, <u>until it's outside of the view then it'll disable</u>. Now here's the thing, If you look again with the same distance exceeded, it won't enable unless you get closer and minimize the distance. This makes things look a little better in your games, since objects with exceeded distances will not disable when in player is looking.*

distanceToCull -> (float) set the distance for the distance cull.

prioritizeDistance -> (bool) set whether you want to prioritize distance over frustum view for culling.

Example for **prioritizeDistance**: If prioritizeDistance is set to true and *you look at an object to be culled and go further away, if you exceed the distanceToCull it will disable no matter what. It doesn't matter whether it's first outside the view or not because now the distance is the priority. You might want for some reason to prioritize the distance but it will certainly make your game look weird since things will disable right in front of the player if distance is exceeded while still in view!*

distanceCullOnly -> (bool) if set to true, frustum culling will be disabled and the enabling/disabling of objects will solely depend on distance.

Build Objects Button

Use this button to pre-build all placeholder-objects of your game objects/prefabs in editor-time before game start. Drastically improving the performance. *You should always try to utilize this even in dynamically generated objects by adding the script to your prefabs and caching the renderers in editor-time before starting the game and dynamically instantiating them programmatically.*

BuildObjects() -> (method) Build all the secondary-objects programmatically.

*Feel free to email me at **muradelboudy95@gmail.com** for support and custom stuff.*

Find more packages from Pathiral here