

Layer Selection Tool

User Manual

Thanks for downloading our asset, we want to provide the best service for you as possible.

If you need help with this product or wish to provide suggestions, please send us email at:

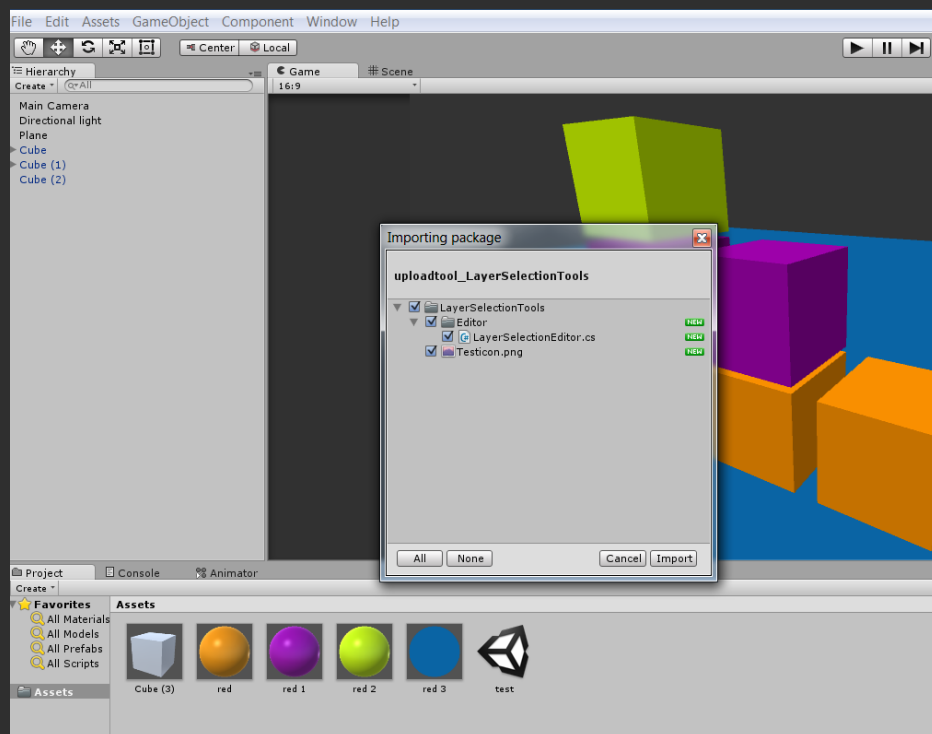
gbFactoryStudio@gmail.com

What I can find here?

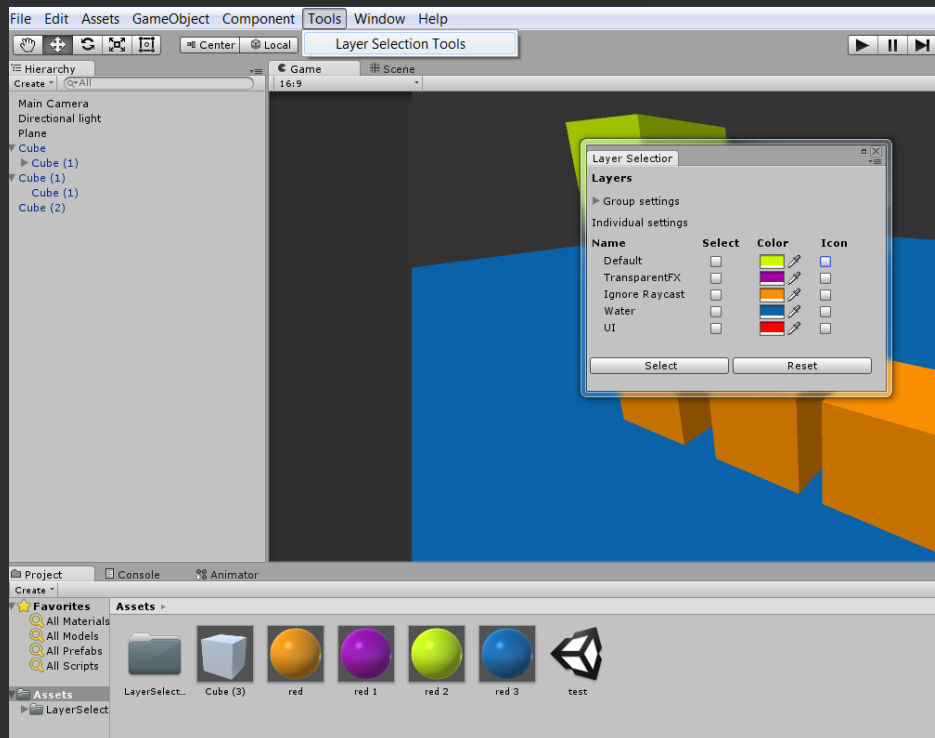
Layer Selection Tool allows you easily select GameObjects in the scene based on selected layers.

How to start?

Just import the package, and you are ready to go.

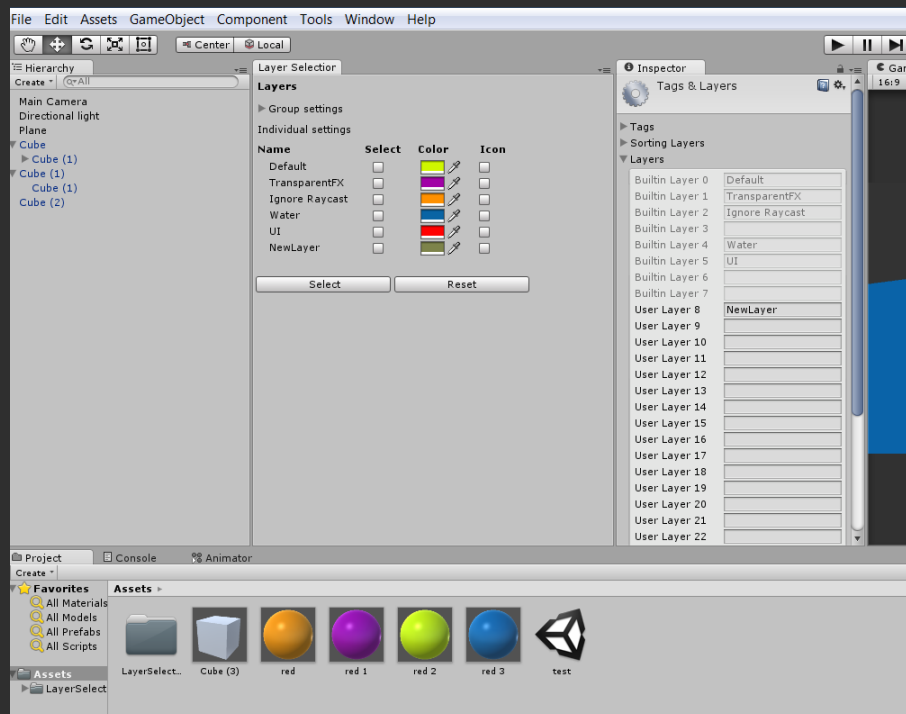


In the top bar, you will have new item Tools, there you can find **Layer Selection Tools** item.

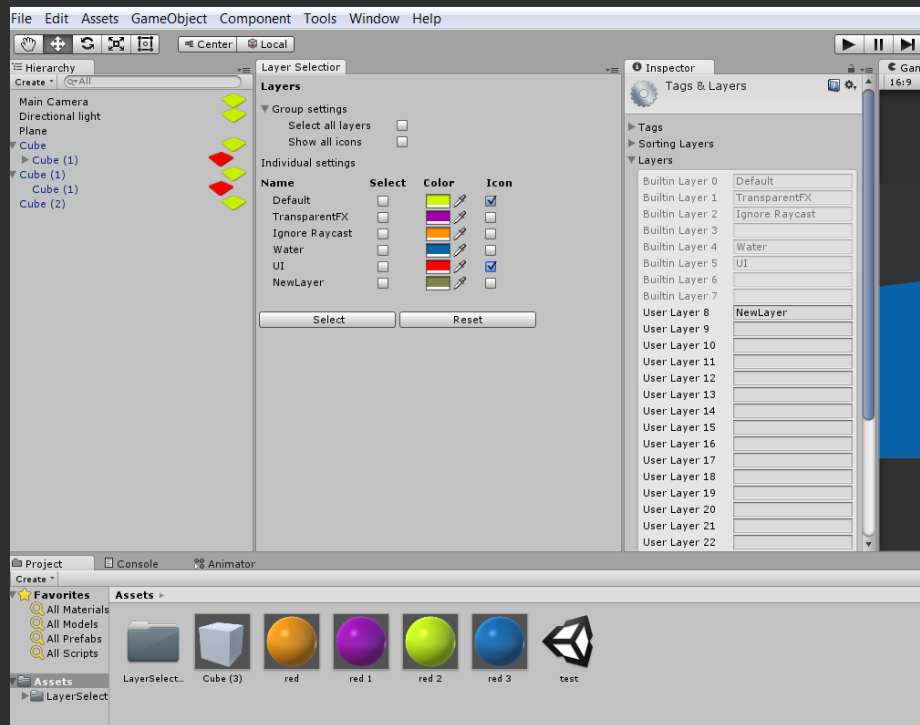


Description:

Tool adapts to layers created in Unity editor.



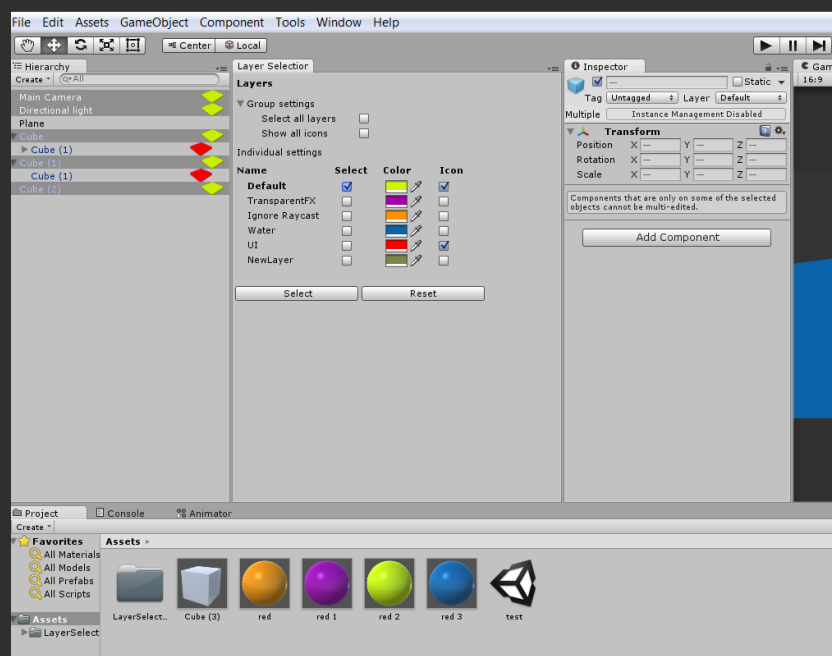
Tool adds icon next to GameObject in hierarchy window, to easily detect and recognize layers. You can also disable the icons if you don't like them.



From the editor window you can pick what layers you want to select – you can pick one, two or all.

There are setting to manipulate whole group or you can pick individually.

Then you click Select button. And magic happens :)



With button Reset, you reset your selection in editor window.

If you don't want to open another editor window, you can just manipulate the selection by clicking the icon next to GameObject in hierarchy view.

In Editor window you can adjust the color of the icon.

If you want to replace the image of the icon, just replace the picture in LayerSelectionTool folder.

Additional info:

Layer Selection Tool will select all objects on active scene, but will skip the inactive.

to change that you need to modify the code of LayerSelectionToolEditor. You need to replace 222 line in code in function Select(). In the script there are possible ways to replace the code, with explanations.

```
//Other possible ways of searching available gameObjects.  
//- need to select first in hierarchy, then it's selecting from those  
// var objs = Selection.GetFiltered(typeof(GameObject), SelectionMode.Deep);  
  
// - will not select inactive objects  
//Object.FindObjectsOfType(typeof(GameObject)))  
  
// - looks also in asset folder  
//Resources.FindObjectsOfTypeAll(typeof(GameObject)))  
  
foreach (GameObject obj in GameObject.FindObjectsOfType(typeof(GameObject)))  
{  
    if (obj.transform.parent == null)  
    {  
        Traverse(obj, ref allObjectsInScene);  
    }  
}
```

If you like our asset, please share in the Asset Store any positive feedback