

Mesh picking For unity



Content

What is it?	2
How to install:	2
How to use:	2
Technical aspect:	3
License:	4
Bug or problem report:	4
Author:	4

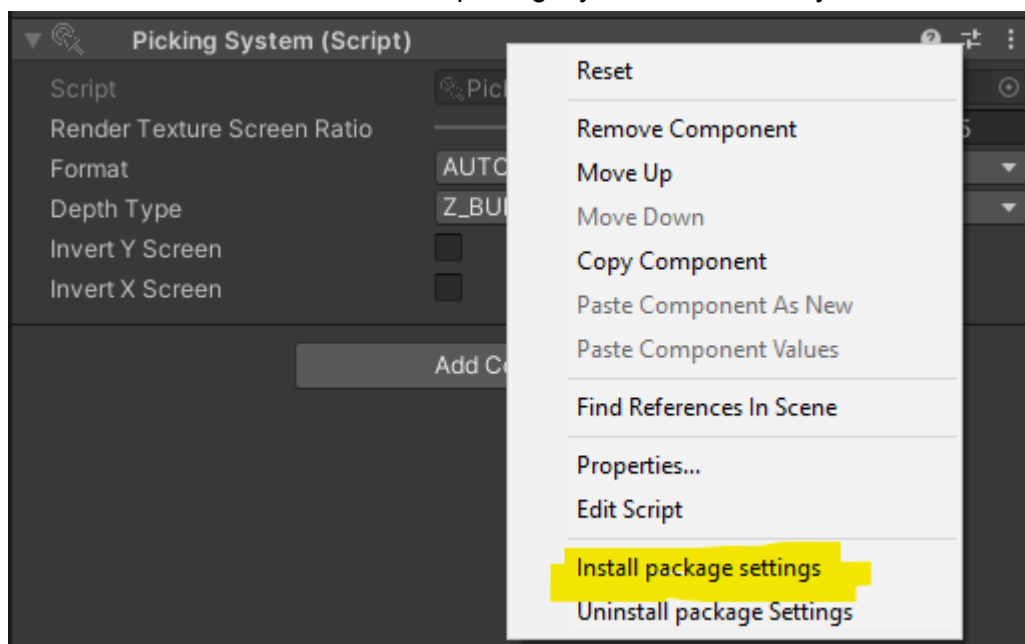
What is it?

Mesh picking is a feature that allows you to get game objects through a render pass. It allows you to have a high level of accuracy without using space partitioning, collision and ray cast.

When you call the Picking function, the mesh visible with the Mesh picking script is drawn in a specific render pass and receives the desired pixel at a position to know the ID of the selected object.

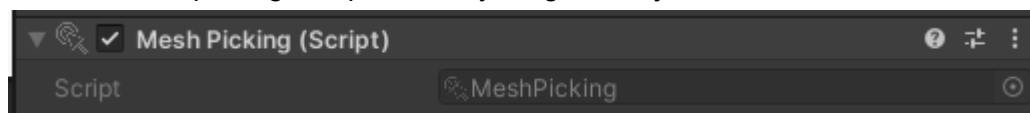
How to install:

Add the PickingSystem component to a gameObject and right-click on it. Then click on Install. The installation will add the picking layer and shader to your default compiled shader.



How to use:

Add the mesh picking component to your gameObject.

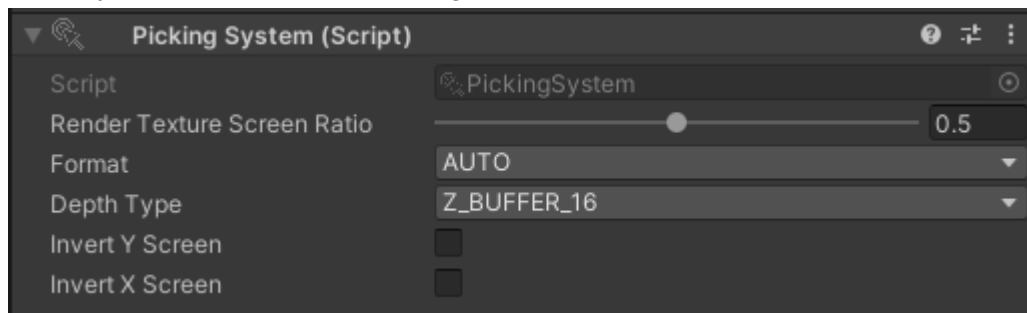


Then call the PickingSystem.Instance.Picking function to process the picking and get the selected gameObject.

```
void Update()
{
    GameObject obj = PickingSystem.Instance.Picking(Input.mousePosition);
    if (obj)
        Debug.Log(obj.name);
}
```

And it's all! Only the gameObject with the mesh picking component can be selected.

If you want to improve the accuracy or optimize the picking process, you can manually add the PickingSystem component to a gameObject. Inside this component, you can define the size of your render buffer according to the screen, its channel, the precision of the Z-buffer...



You have tooltips for all properties to help you.

The only things you need to use is the MeshPicking and PickingSystem scripts. The picking shaders and rendering pipeline are used internally.

Technical aspect:

This tool has been tested and build on an Android and Windows application.

Each time you want to select an element, a new rendering buffer is created and is placed in the temporary buffer of Unity. This allows to simplify the code and to modify the render texture specification without recreating it.

A custom render pipeline is used. If you use a GPU debugger like renderDoc, you can see that this pass is as simple as possible.

69-3495	Camera.CustomRender
88-127	MousePickingRenderPipeline
91	glClear(Color = <0.000000, 0.000000, 0.000000, 0.000000>, Depth = <1.000000>)
92	MousePickingRenderPipeline
94-127	RenderLoop.Draw
106	glDrawElements(36)
112	glDrawElements(24432)
118	glDrawElements(240)
121	glDrawElements(240)
124	glDrawElements(240)
127	glDrawElements(240)
150-151	WaitForRenderJobs

The main camera, materials, and layers of the game objects are modified for the picking pass and reset afterwards.

This feature work with SkinnedMesh so with animation.

All GameObject ID are created and recycled. These ID are encoded in float and send to GPU. When we read pixel value, we transform decode it from float to int.

License:

This tool is based on the unity license and the MIT license. So you are free to modify this tool according to unity rules.

Bug or problem report:

Feel free to report your problem or bug in my mail below. Please be consistent in your description to reproduce the bug in my machine.

Author:

Six Jonathan

Email: Six-Jonathan@orange.fr