UI Advanced Mask

UI Advanced Mask is a plugin highly focused on UI masking. It's designed to work with unity native UI system, but also able to work in 3d game world.

Unity natively includes a UI mask feature, but it's not perfect. It's using cutoff and can't have smooth edge or half transparent. Therefore i develop this ortho-projector-like UI mask plugin with ui pointer raycasting, so that we can do beautiful masking on 2D/3D UI.

All position,scale,rotation,size is based on RectTransform even if it's not in a canvas.

UIAdvancedMask.cs
Just like the built in UI mask
Texture texture;
texture for mask

float alpha = 1;
alpha multiply

bool updateEveryFrame;
update mask material value in everyframe. you can set it off when the RectTransform of the mask is static. You can also call function updateMaskMaterials() manually.

bool UseRedAsAlpha;
In case your texture is no in alpha channel

bool flippedAlpha;
flipped the source value for alpha

bool forceClampX;
clamp the texture horizontally in the shader way

bool forceClampY;
clamp the texture vertically in the shader way

bool maskOnRaycast;
if enabled, pointer raycasting will pass through area that alpha is smaller than 0.5

void updateMaskMaterials()

after changing the Transform of the mask, you should update its value to its mask material with this function. Or you can enable updateEveryFrame and let it do it every frame

RemoteUIAdvancedMask.cs
Texture texture;
texture for mask

float alpha = 1;
In case your texture is no in alpha channel

public bool updateEveryFrame;
update mask material value and influenced objects in everyframe. you can set it off when the RectTransform of the mask and target objects list is static. You can also call function updateMaskedObjects() and updateMaskMaterials() manually.

bool UseRedAsAlpha;
In case your texture is no in alpha channel

bool flippedAlpha;
flipped the source value for alpha

bool forceClampX;
clamp the texture horizontally in the shader way

bool forceClampY;
clamp the texture vertically in the shader way

RectTransform[] targets;
Because it's remote, you have to add your targets.

public bool addToChildren;
Make the mask effect also works on targets' children

RemoteUIAdvancedMaskCanvasRaycast.cs
public RemoteUIAdvancedMask target;
use target RemoteUIAdvancedMask to check pointer raycast.

MaskTextureRenderer.cs
This is for you to render textures to a RenderTexture so that you can do (so call) multi-mask. Of course, you can do the same result with camera and mesh renderers. But it requires layering, which may make the project more complex to manage.

RenderTexture renderTarget;
link the rendertexture for rendering

bool renderEveryFrame = true;
automatically call render() every frame

Shader renderShader
Please link Unlit/Transparent here. All rendering will use this shader

bool debugDisplay
display final result in edit view

public void render()
render all child GameObjects with MaskTextureNode into the target RenderTexture

MaskTextureNode.cs
You need to add this component to the child GameObjects of a MaskTextureRenderer. The size is based on RectTransform even if it's not in a canvas. You can use texture or sprite on it but only one will work for each node.

Texture texture
the texture source for rendering

Sprite sprite
the sprite source for rendering

Tips:

- RemoteUIadvancedMask is heavier than UIAdvancedMask on performace.
- For Canvas Raycast feature, try to only use texture with read/write enabled. If not, my script will use a tricky way to get color at the pointer position, which is much slower.
- You cannot use a mask inside a mask
- Every boolean option of the mask component is actually representing a step in the shader. Please concern this fact when you are preparing textures. For example, set texture as clamp instead of using force clamp in the mask.
- You can write your shader based on mine. My ui mask script will try to set the same parameters into your material.

Future
There is something i probably could do but i am not sure if people really need. Please leave a comment in asset store if you want one of these

- During the development, the first goal is to make it easy to use. I only sent little time on optimzation, and i probably can do more.
- Some functions for programmers to play with, eq: getting alpha value of a point.
- when you unparent an ui object from a masked object, the masked material still stay in that ui object. I can make automatically release it.
- transition map on mask, so that you can do creative transition on the mask instead of just fading