ALPHA MASK: UI, SPRITES, QUADS

by Tag of Joy

This document describes usage guidelines for the **Alpha Mask** developed by **Tag of Joy**. The plugin should be purchased on the Unity Asset Store (http://u3d.as/bzJ). Sharing or distribution is not permitted.

CONTACT INFORMATION

You can contact the developers of the Alpha Mask by e-mail: info@tagofjoy.lt

HOW TO USE

- Create an empty Game Object as a sibling of the elements that need to be masked (the masked elements can have a deeper level of hierarchy – they will still be masked). Rename the new object, so that it's clearly identified (e. g. "Mask");

- Attach the Alpha Mask component to the new Game Object and assign a texture that you want to use as the mask to its material slot.

- Click "Apply Mask to Siblings in Hierarchy", which will detect all siblings (including their children) and apply the mask to all them (if they use the Default Sprite or Unlit/Transparent shader).

- The mask can be moved, scaled and rotated freely in the Editor, but it can only be rotated over a chosen axis (depending on what mapping axis is selected).

INSPECTOR FIELDS

- **Masking Enabled** – does the mask need to have the effect on the siblings? Can be enabled/disabled in the Editor, as well as in run-time.
- **Mask Mapping World Axis** – defines, over which axis the mask should be applied. This is usually the axis, which corresponds with the camera direction.
- **Invert Axis** – in case you need to map the mask over an inverted axis.
- **Clamp Alpha Horizontally** – if the texture isn't clamped by Unity (in import settings), then you can choose to clamp it horizontally only (it will be repeated vertically, unless chosen otherwise).
- **Clamp Alpha Vertically** – if the texture isn't clamped by Unity (in import settings), then you can choose to clamp it vertically only (it will be repeated horizontally, unless chosen otherwise).
- **Clamping Border** – if one of the two above settings are enabled, you can use this variable to tweak the “edge” of clamping. Depending on the alpha texture size and its usage, you might run into texture clamping issues. In that case, try increasing (or lowering) the Clamping Border value.
- **Use Mask Alpha Channel (not RGB)** – the mask uses the texture RGB channels by default. Toggle "Use Mask A Channel (not RGB)" to use the Alpha channel of the texture instead.
- **Display Mask** – toggle this setting to enable or disable the visibility of the mask (only the visibility, not its effect). This setting is only available in the Editor and when not running the game.

MATERIALS

The Alpha Mask component automatically creates the needed masked materials for the masked elements, when you click “Apply Mask to Siblings in Hierarchy”. However, if you want to have your own materials, you can create a material manually, assign one of the masked shaders to it, and assign the material to all masked elements (for a single Mask hierarchy, one masked material is usually enough).

This is especially important when using prefabs (see section “Using the Alpha Mask with prefabs”).

USING THE ALPHA MASK WITH PREFABS

To simplify the usage of the Alpha Mask, materials and necessary components are created automatically. However, when the masked objects and/or the Mask itself is used in prefabs, some of the components break. So, there are a few additional steps necessary to make it work in prefabs. The steps depend on whether you only need the masked elements to be in a prefab or if you need the mask to be in the prefab, too – if the mask doesn't need to be in a prefab, you can skip steps 4, 5, 6:

1. Create the Mask and the Sprites/UI elements/Quads that you will mask (as described in the “How to Use” section, but do not click “Apply Mask to Siblings in Hierarchy” just yet).
2. Manually create a material (within the assets) and assign either the "Alpha Masked/Sprites Alpha Masked – World Coords" (if you’re masking Sprites or UI elements) or the “Alpha Masked/Unlit Alpha Masked – World Coords” (if you’re masking Unlit/Transparent objects) shader to it.

3. Drag this material into the material slot of all the Sprites/UI elements/Quads that you've created in step 1 (the ones that you want to mask).
4. Create another material and assign the "Unlit/Transparent" shader to it. Also assign the Mask texture to this material.
5. Drag this material into the material slot of the Mask that you've created (on the Mesh Renderer of the Mask object).
6. On the Mesh Filter of the Mask object choose "Quad" (the standard Unity mesh object) instead of the already assigned mesh.
7. Now click “Apply Mask to Siblings in Hierarchy” on the object with the Alpha Mask. You should now see the Sprites/UI elements/Quads being masked as they should be.
8. If everything worked so far, create a prefab from the hierarchy. You can now instantiate the prefab, and it should work as intended.

If the prefab hierarchy contains the masked elements, as well as the Mask itself, and you are instantiating multiple instances of that prefab, there is one additional step that has to be performed: after instantiating an isntance of a prefab, you have to call **DuplicateMaskedMaterials()** for its Mask.

CHANGING THE MASK TEXTURE IN RUN-TIME

In the Mask script (the one that is attached to the Mask) there is a method called **ChangeMaskTexture(Texture texture)**. This method can be used to easily change the Mask texture in run-time.