Link to Version of this Manual that is Constantly Being Updated

https://docs.google.com/document/d/1UCxu07cAwVSxPS3i7ouwJIBSrV2yh9v0bNlYWYeJJcs/edit

---

How to Use
Step 1: Drag or Add one the scripts that starts with “concaveOutline” or “convexOutline” onto a game object with a sprite renderer that is concave or convex respectively
Step 2: IF you want to change the settings from the inspector then drag the correct inspector script onto the gameobject as well (v2 script uses v2 inspector script, etc) [NOTE: only the v2 and v4 script version can be used with the inspector scripts]

There are 2 Groups of Outlines for different type of Sprites

CONCAVE (for concave sprites)	**CONVEX** (for convex sprites)
-----OPTIMIZATION VARS-----	
Update Sprite Every Frame	
-----DEBUGGING VARS-----	
Show Outline Game Objects in Hierarchy	
-----SPRITE OVERLAY VARS-----	
Active	
Order in Layer	
Color	
-----CLIPPING MASK VARS----- (allows outlines to work well with semi transparent sprites)	
Clip Center	
Alpha Cut Off (only relevant if “Clip Center” == True)	
Custom Range (only relevant if “Clip Center” == True)	
Front Layer (only relevant if “Custom Range” == True)	

Back Layer (only relevant if “Custom Range” == True)	
-----OUTLINE VARS-----	
Active	
Color	
Order in Layer	
Size (world space)	Size (Scale Value)
Outline Scale X With Sprite	
Outline Scale Y With Sprite	
Force Retain Proportions With Children	
-----PUSH VARS-----	
N/A	Push Type is Regular or Custom
N/A	Count of Objects Making Outline
N/A	Start Angle (for type regular)
N/A	Push Type Radial or Square (for type regular)
N/A	Std Size (for type custom)

In each group there are 4 types of scripts

1. Fastest (No Children or Parent Allowed) (Not Inspector Friendly)
2. Fast (No Children or Parent Allowed) (Inspector Friendly)
3. Slow (Children and Parents Allowed) (Not Inspector Friendly)
4. Slowest (Children and Parents Allowed) (Inspector Friendly)

NOTE:

- For the Inspector Friendly Types (2 and 4) you will need to add the regular script and then the appropriate inspector script
- Your Can Only set “Force Retain Proportions With Children” if you have children

- For the outline to work properly with an animation, you must set the "Animator" "Update Mode" to "Animate Physics
- Duplication of the Object in (Edit -or- Play) Mode will create the object but with all the DEFAULT outline settings
- NOTE: this Object "Executes In Edit Mode" For the Sole purpose of choosing your settings
    - HOWEVER... when you are actually using the asset you MUST change the variables from code
    - REASON... this is because this object is [NOT SERIALIZABLE]... so unity will clone it and use its copy in game mode
    - REASON ITS NOT ADDED... it's a lot more trouble than it's worth and can cause significant problems if not used properly
- Since I am using the sprite to create an outline... if the sprite SOURCE is semi transparent then the outline and the overlay will also be semi transparent

FUSE -vs- OVERLAP

This Refers to object A and object B that have their own respective outlines

- FUSE
    - by DEFAULT outline A will "Interact" with outline B so it seems like object A and object B are merging into 1 sprite
- OVERLAP
    - IF obj A and obj B have outlines with Masks that affect different layers (and of course the outlines are being clipped within their own perspective range) outline A will overlap outline B [If (outline A orderInLayer) > (outline B orderInLayer)]

Family Relationships

- [IF] someone wants to be our parent -AND- they are not already our child -> (Let Them be your parent)
- [ELSE] (don't let them because you are their parent)
- Children must not have an inspector helper script... or they will not follow their parent properly
- FOR NOW only outlines that are of the same version number should be parented (they CAN have different group types [convex, concave])
    - EX: Parent = convexV1 & Child = concaveV1 -> WORKS
    - EX: Parent = convexV1 & Child = convexV2 -> DOESN'T WORK
- When you change the size of the parent you change the size of the children outline in proportion that the size the children had with the parent before the size change
- Sometimes children's outlines might get to small and become 0, then throwing off the proportion they maintained with their parent, so you can choose to limit how large or small you can make the parent outline based on how small the children outlines should get
- Currently we only pass (1) Sprite Overlay, AND, (2) Basic Outline Data (3) Clipping Mask [on/off] to our children
- Use code snippet below in variable set area to pass that variable value to your children
    - Keep in Mind the Concave and Convex Outline Types don't share all the variables
    - I Considered most of the settings they don't share sprite specific
    - Remember to replace concaveOutline and convexOutline for the proper class name of the version you are using

Code Snippet

```
for (int i = 0; i < children.Count; i++)
{
    if (children[i] != null)
    {
        if (children[i].GetComponent<concaveOutline>() != null)
            children[i].GetComponent<concaveOutline>().Active_SO = active_SO;
        if (children[i].GetComponent<convexOutline>() != null)
            children[i].GetComponent<convexOutline>().Active_SO = active_SO;
    }
    else
    {
        children.RemoveAt(i);
        i--;
    }
}
```