Sprite Billboard Flares

Light sources in Unity are only visible when they shine on objects. This means it can be hard to see where they are when you're in space. To make it clear where your lights are, you can use flares to give them a nice looking visual.

SGT comes with several components and materials that allow you to make impressive flared using billboards. A billboard is a flat mesh (e.g. quad or sprite) that rotates toward the camera.

Step 1 - Add a Billboard

To add a billboard to your scene, drag and drop a sprite from your **Project** tab into the **Hierarchy** tab.

For example, the **"Space Graphics Toolkit\Basic Toolkit\Textures\Baked Flare A"** sprite can be used.

Your scene should now contain a new GameObject with the **SpriteRenderer** component.

To turn this sprite into a billboard, simply add the **SgtBillboard** component. I also recommend you enable the **Avoid Clipping** setting.

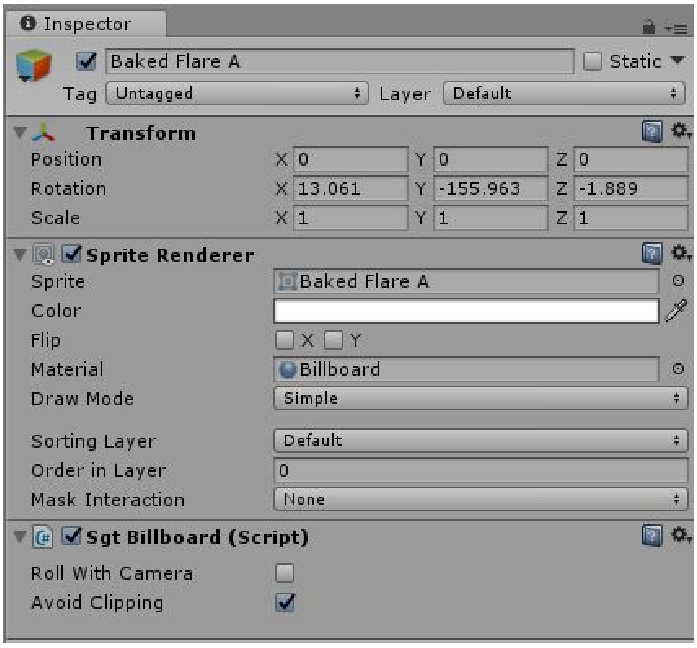
You can now play your scene and your billboard should always face the camera.

NOTE: Because the SgtBillboard component takes over the GameObject's rotation, you shouldn't have any components on this GameObject that rely on the rotation (e.g. directional light).

Step 2 - Make it glow

By default, sprites in Unity use the default sprite shader, which uses alpha blending.

To make them glow, you need to use a shader that uses additive blending. Simply change the **Sprite Renderer** component's **Material** setting to the "**Billboard**" material that comes with SGT.



Inspector

☑ Baked Flare A □ Static

Tag: Untagged Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	13.061	-155.963	-1.889
Scale	1	1	1

☑ **Sprite Renderer**

Sprite	Baked Flare A
Color	
Flip	□ X □ Y
Material	Billboard
Draw Mode	Simple
Sorting Layer	Default
Order in Layer	0
Mask Interaction	None

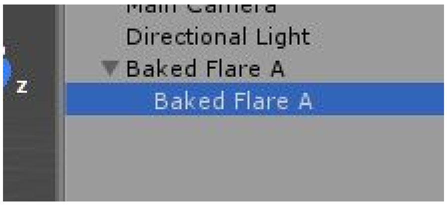
☑ **Sgt Billboard (Script)**

Roll With Camera	□
Avoid Clipping	☑

Step 3 - Make it glow even more

The billboard you just created is a good start, but to make your flare look more impressive you should add a second layer.

To do this, follow step 1 again, but this time drag and drop the sprite onto the sprite GameObject you created earlier. This will add a new sprite as a child of your existing sprite.



Similar to step 2, you need to change the material on your new flare. But this time, you should choose the “**Billboard Overlay**” material. This material will force your second flare layer to render on top of everything.

NOTE: Your parent GameObject already has the **SgtBillboard** component that controls the rotation, so your second layer doesn’t need it.

Step 4 - Make it glow less

You may notice the second billboard layer you added looks great from some angles, but it shines through all foreground objects (e.g. planets), which makes it look bad.

To fix this, begin by changing the **Layer** of all your billboard GameObjects to “**Ignore Raycast”**.

Next, right click your **Hierarchy** window and select **Space Graphics Toolkit** → **Depth Camera**, your scene should now have a **Depth Camera** GameObject.

Next, add the **SgtDepthScale** component to your second billboard layer. This component will use the Depth Camera you added earlier to calculate how much it should shrink the current GameObject based on the optical thickness (occlusion) between the camera, and the light flare.

If you play your game, your first light flare should always be visible and look the same, and your second layer should now grow and shrink based on what’s in front.

If not, try adjusting the **SgtDepthCamera** component’s **Size** setting. This value is in world space and should correspond to your scene scale.

You can also examine the “**Sprite Billboard Flare**” demo scenes to see how they’re setup.