Making a Terrestrial Planet

This tutorial will teach you how to make a basic terrestrial planet (like Earth).

Step 1 - Make a Sphere

You can use a basic Unity sphere for this. For example, right click in the Hierarchy window, and select **3D Object → Sphere**, and your scene will have a new Sphere GameObject.

If you want an even rounder sphere, I recommend you use the **Geosphere40** mesh that comes with SGT. Keep in mind the SGT sphere has a radius of 1.0 unit, whereas the default Unity one has a radius of 0.5, so your collider settings may need updating.

Step 2 - Add an Atmosphere

Right click your Sphere GameObject in the Hierarchy window, and select **Space Graphics Toolkit → Atmosphere**, your sphere will now have the Atmosphere GameObject added as a child, with the SgtAtmosphere and SgtSharedMaterial components.

Step 3 - Set up the Atmosphere

At the bottom of the SgtAtmosphere component in the Inspector window, you will see 3 buttons. If you click these, then your atmosphere will automatically be set up using basic values.

The first button adds the **SgtAtmosphereDepthTex** component, which will automatically generate the **Inner Depth Tex** and **Outer Depth Tex** settings for your SgtAtmosphere.

The second button sets the **Outer Mesh** to use the **Geosphere40** mesh, as well as calculate the **Outer Mesh Radius**.

The third button adds your basic sphere from step 1 into the **SgtSharedMaterial.Renderers** list, and calculates the **Inner Mesh Radius field**. This renderers list lets the SgtAtmosphere know which renderers you want the atmosphere material to be applied to.

Step 4 - Done!

You now have a basic planet with a fully volumetric atmosphere, so your camera can seamlessly travel from space to the surface.

I recommend you now experiment with all the atmosphere settings to understand what they alter. If you're unsure what a particular setting does, then hover your mouse over the field to see a tooltip, or read the associated documentation file.

NOTE: If you're using the floating origin system then you also need to add the **SgtFloatingPoint** and **SgtFloatingObject** components.

Optional Step - Adding Lighting

To add lighting to your planet, first enable the **Lit** setting, and then click the **Add LightingTex** button.

Next, you need to let the atmosphere know which lights you want to use. To do this, add the **SgtLight** component to your main scene light (By default this will be the **Directional Light** GameObject).

NOTE: Adding lighting means the nearest shadow will also be received by the planet.

Optional Step - Casting Shadows

If you want this planet to cast shadows using the SGT shadow system (e.g. so it can cast on rings), then add the **SgtShadowSphere** component alongside the SgtAtmosphere component.

To make the shadows look correct the **Radius Min/Max** values should be similar to the **SgtAtmosphere** component's **Inner Mesh Radius** setting.

Optional Step - Adding Clouds

If you want your planet to have a basic cloud layer above it, then right click your Sphere GameObject in the Hierarchy window, and select **Space Graphics Toolkit → Cloudsphere**, your sphere will now have the Cloudsphere GameObject added as a child, with the SgtCloudsphere component attached.

You then have to specify the **MainTex** setting, which should be a cubemap of your cloud surface texture. You can generate cube maps from normal cylindrical (equirectangular) textures from the texture input settings in the inspector.

Like with the planet, you can enable the **Lit** setting if you want lighting, and adjust the settings to your liking.

Optional Step - Adding Aurora

If you want your planet to have aurora above and below it, then right click your Sphere GameObject in the Hierarchy window, and select **Space Graphics Toolkit → Aurora**, your sphere will now have the Aurora GameObject added as a child, with the SgtAuroracomponent attached.

You can then click the **Add MainTex** button, which will add the **SgtAuroraMainTex** component, which generates the **Main Tex** setting for you.

Keep in mind aurora mesh generation is quite complex, and has many settings. I recommend you experiment with each to see what they do until you get results you like.