C# API

All Space Graphics Toolkit features can be accessed from C#. Most features have the same names as seen in the inspector. For example:

```
var myBoxStarfield = new GameObject("My Box Starfield").AddComponent<SgtBoxStarfield>();
```

or

```
var myBoxStarfield = SgtBoxStarfield.CreateBoxStarfield();

myBoxStarfield.MainTex = myMainTex;

myBoxStarfield.UpdateMaterial();

myBoxStarfield.Seed      = 1234;
myBoxStarfield.StarCount = 5000;

myBoxStarfield.UpdateMeshesAndModels();
```

One major difference between editing in the inspector and editing in C# is that from C# you must sometimes call special methods to force the component to Update. For example, modifying the BoxStarfield's MainTex field requires you to call UpdateMaterial, and modifying star distribution settings like Seed and StarCount require you to call UpdateMeshesAndModels.

The reason why these calls aren't automatically done each time a value is modified is because if you modify multiple variables that require the same update call, you would waste CPU time updating multiple times when it only needs to get called once.

The reason why these calls aren't automatically delay called (e.g. in Update) is because that adds a lot of complexity to certain components, and makes easy for hard to find bugs to creep in.

The easiest way to see what method call is required to update a component is by looking at the inspector code for the setting. For example:

```
DrawDefault("MainTex", ref updateMaterial);
```

This is the line of code that draws the MainTex field for the SgtBoxStarfield component (or more accurately the SgtQuads base class) which as you can see is associated with the updateMaterial method. You can see the inspector code for each component at the top its C# file inside the UNITY_EDITOR block.