Read Me

This is Space Graphics Toolkit’s **Basic Pack**, which contains a wide range of components for making any space game. The most important feature of this package is the **Floating Origin System**.

What is a Floating Origin System?

Normal Unity scenes store positions using three floats (Vector3), where a position of 0,0,0 is called the origin of the scene. This system works great for most games that take place in a small area, but space games often need to take place over thousands or millions of kilometers, which the Vector3 alone is simply not capable of doing consistently.

The issue is that float values lose decimal precision the higher they get (or lower for negative), which means if your spaceship is 1 million meters/units from the center of the scene, the accuracy of the camera and position values will be very poor, giving you graphical issues, physics issues, and so on. So you want to keep your position values as close to the origin (0,0,0) as possible, so float accuracy remains high.

One solution is to keep your camera fixed at 0,0,0, and instead move all the objects in your scene by the distance the camera would have normally moved. This keeps all rendered objects close to the origin, but for fast paced games means you need to move every object around the camera constantly, which can be slow, especially when the physics system is involved.

A better solution that SGT uses is instead of keeping the camera at 0,0,0 all the time, it snaps it back to 0,0,0 when it moves too far away. This approach gives good performance because objects only need to have their positions shifted occasionally, it also maintains good rendering precision because values are always near the origin, and it also means most camera systems will still work, because you can still manipulate the camera position normally.

Another issue with this idea is that it still doesn’t solve float accuracy issues for other objects in the scene. Even if the camera is kept near the origin at all times, distant planets and stars will still be placed millions of kilometers away, which means their positions will have degraded accuracy.

To fix this, SGT implements the **SgtFloatingPoint** component, which stores the position of an object using a very high precision alternative to Vector3 values. These high precision values are so accurate that positions can be stored for any point inside the observable universe (46.6 billion lightyear radius). Of course Unity still renders everything using Vector3 values, but these are calculated every time the camera snaps back to the origin, so there is no loss of data as you move through the scene.