Sci Fi Kit Volume 1 - Bohn Studios

This kit contains atlased assets that are designed to snap to scale of 1.0 , 0.5, and 0.25

We use atlases to make levels load fast and have minimal impact on the cpu.

For questions and support regarding this package, please submit inquiries at
https://www.bohnstudios.com/contact/

What is a Texture Atlas?
Other than being the cornerstone of responsible 3d optimization, texture atlases are best described as single texture that contain many textures for many 3d models in a scene/game/visualization/etc. The alternative to texture atlasing is to use many different textures/materials for seperate objects; one of the most common ways to get poor frame-rates and cpu bottlenecks in your project. While it’s not necessary to understand in depth, it is helpful to know that using texture atlases have direct impact on CPU performance, and even your memory (RAM) if you’re able to get creative with your usage of atlases and “trim sheets”. One cool trick is to use a color sheet, along with your most important textures. Combined with using decals (also atlased), it is possible to create large areas using just a handful of texture atlases.

Trim Sheets: The Super Atlas
One of the downsides to using atlases for some textures is the fact that with most UV mapping functionality, you are not able to repeat, say a 512x512 chunk, out of a 2048 atlas. We can enjoy the best of both worlds by using a nice little compromise called a trim sheets...and they are quite beautiful. Simply take advantage of horizontal repeating by making your atlas contain imagery that reaches from the left side all the way to the right side of your 3d texture. A common example would be to use this for a fence, wall, or road, that is commonly repeated across an environment.

This has double advantage in that you can use this concept along with your actual mesh creation/modeling to potentially save hundred of thousands of vertices and triangles that have to be rendered. *(insert image of a fence with 10 separate parts next to the same fence with 1 part. Compare the vertices and triangle count; show open world example where the fence is repeated 1,000 times and compare the vertices count)*

Atlases and Dynamic Objects

In the Unity3d Game Engine, dynamic objects that share a material (ideally an atlas), are automatically made to run faster through dynamic batching. To take this a step farther, you can incorporate “pooling” for your most commonly used moving objects such as projectiles, health packs, ammo, loot crates, etc. Basically anything that repeats itself constantly throughout your game. The next step further would be to use GPU instancing on top of this. GPU instancing is most effective if your have hundreds or even thousands of the exact same object. Think of an outer space scene with an asteroid belt with thousands of small rocks. If you only used dynamic batching for this, you would run into a bottleneck pretty quickly. So if you have thousands of repeating objects, GPU instancing might be the best way to accomplish your goals with low overhead. Just remember that the device you are targeting must have dx11 or higher (or equivalent).