

Voxel Engine Concepts

Cubiquity is a *voxel engine*, which means that it represents its objects as samples on a 3D grid. This is conceptually similar to the way a bitmap image represents a photograph as samples on a 2D grid (the pixels). In many ways a voxel can be considered the 3D equivalent of a pixel. The value stored at a voxel may simply be a colour (as in the case in our ColoredCubesVolume) or it may be some more advanced representation.

Rendering such a 3D grid directly is not trivial on modern graphics cards as they are designed for rendering triangles rather than voxels. Therefore the usual approach is to create a triangle mesh which has the same shape as the underlying voxels and then render this instead. This triangle mesh can also be used for other purposes such as collision detection.

Cubiquity largely hides this implementation detail from the user. All you need to do is write the desired values into the voxels, and Cubiquity automatically takes care of keeping the corresponding mesh synchronized with them. These voxel values may be read from disk, be generated procedurally, or created through other approaches. This is largely up to you.