

Using Views for Data Retrieval

The C# Client library allows a simpler interaction model for applications that just want to read spatial data from Voxel Farm servers. This is the case of a client-side report for instance, where multiple datasets can be requested and inspected by a client application, and then used to compute additional information.

Like it was the case with rendering a view, it is still necessary to set up the `VoxelFarmView` object with a server, a project, and the desired view configuration.

The `VoxelFarmView` instance provides callbacks that fire whenever information is received from the server. A client application that is interested in only receiving the data, can often do with just setting up some of these callbacks.

The data callback functions are:

<code>OnViewStarted()</code>	Notifies that the view has started. This happens after the client establishes a successful connection to the server.
<code>OnReceiveVoxelData(VoxelData data)</code>	Notifies voxel data for a cell has arrived
<code>OnReceiveMeshData(MeshData data)</code>	Notifies mesh data for a cell has arrived
<code>OnReceiveTextureData(TextureData data)</code>	Notifies texture data for a cell has arrived
<code>OnReceivePointData(PointData data)</code>	Notifies point data for a cell has arrived
<code>OnCellComplete(ulong cell, VoxelFarmCellData data)</code>	Notifies all expected components for a cell have arrived

VoxelData

This object contains voxel data for a cell. It has the following properties:

<code>ulong cellId</code>	Unsigned 64bit identifier for the cell.
<code>byte layer</code>	Identifies the metadata component that originated this data in the view metadata

float[] values	Contains the requested voxel attribute values. For each attribute that was requested in the metadata, a section of VoxelFarmConstant.BLOCK_DIMENSION^3 floats appear in this array. For instance, the values for third attribute that was requested would appear at index (2 * BLOCK_DIMENSION * BLOCK_DIMENSION * BLOCK_DIMENSION)
----------------	---

MeshData

This object contains voxel data for a cell. It has the following properties:

ulong cellId	Unsigned 64bit identifier for the cell.
byte layer	Identifies the metadata component that originated this data in the view metadata
byte submesh	Identifies the section of the cell that is associated to the mesh. A single cell may contain up to 7 different meshes. The first mesh is the contents of the cell, the other six (one for each side of the cell) is a seam mesh that allows the cell mesh to connect to lower LOD neighboring cells.
float[] vertices	Contains the coordinates for the vertices. These coordinates are local to the cell origin, where (0,0,0) corresponds to the origin of the cell and (1,1,1) to the farthest corner from the origin.
float[] normals	Contains the surface normal vectors for each face-vertex combination
float[] colors	Contains an RGB color triplet for each vertex
uint[] faces	Contains triangle definitions for the mesh, each entry is a reference to a vertex in the vertices array.

TextureData

This object contains texture data for a cell. It has the following properties:

ulong cellId	Unsigned 64bit identifier for the cell.
byte layer	Identifies the metadata component that originated this data in the view metadata
byte textureType	Identifies the texture type. The following types are currently supported: 0 – Diffuse texture 1 – Normal map texture

byte textureFormat	Defines the texture format: 0 - DXT1 1 - DXT5 2 - JPG 3 - PNG 4 - RGBA
ushort textureSize	Contains the dimensions in pixels of the texture. Textures for cells are square.
byte[] textureData	Contains an RGBA values (8bits per channel) for the texture

PointData

This object contains point cloud data for a cell. It has the following properties:

ulong cellId	Unsigned 64bit identifier for the cell.
byte layer	Identifies the metadata component that originated this data in the view metadata
float[] vertices	Contains the coordinates for the points. These coordinates are local to the cell origin, where (0,0,0) corresponds to the origin of the cell and (1,1,1) to the farthest corner from the origin.
float[] colors	Contains an RGB color triplet for each point

VoxelFarmCellData

This object contains all resources associated with a cell. Since a view may require multiple components to load for a single cell, this data structure will contain data results for each component in the view metadata. Individual component results can be found in the “layers” property of the object:

List<VoxelFarmCellDataLayer> Layers	Contains a list of “VoxelFarmCellDataLayer” objects, one object per component in the view metadata. Depending on the component type, the object uses a more specialized class that derives from “VoxelFarmCellDataLayer”. These are: 1. VoxelFarmCellDataLayerMesh – Contains mesh resources 2. VoxelFarmCellDataLayerOrtho – Contains texture resources 3. VoxelFarmCellDataLayerPoints – Contains point cloud resources 4. VoxelFarmCellDataLayerVoxels – Contains voxel resources
-------------------------------------	---

Revision #1

Created 17 March 2025 20:29:10 by admin

Updated 17 March 2025 20:30:54 by admin