

12/5/2015

UNITYCODER.COM

POINT CLOUD VIEWER & TOOLS

v1.7

Point Cloud Viewer & Tools

Table of Contents

INTRODUCTION	2
FEATURES / TOOLS	2
REQUIREMENTS.....	2
HISTORY	3
FILES IN THE PACKAGE.....	3
USAGE INSTRUCTIONS.....	4
Point cloud To Binary Converter	4
Point cloud to Mesh assets Converter	5
Using the Webplayer Viewer	6
TUTORIAL.....	7
USING BREKEL PRO POINT CLOUD DATA.....	9
POINT SIZE FOR MESH IN DX11 MODE.....	10
CUSTOM BINARY FORMAT	10
REFERENCES / SHOWCASE	11
EXTRAS FOLDER	11
FAQ	12
KNOWN ISSUES.....	13
EXTERNAL TOOLS.....	13
SAMPLE POINT CLOUD DATA(S) FROM WEB.....	13
SUPPORTED FILE FORMAT/ HEADER EXAMPLES	14
SUPPORT & FEEDBACK	16

Point Cloud Viewer Tools

INTRODUCTION

Basic Point Cloud Viewer & Tools for Unity3D (DX11) and also Point Cloud to Unity mesh assets (non-DX11).

FEATURES / TOOLS

- Up to ~40 million points can be displayed with single color shader (* could depend on your pc hardware also) or around 20 million points with RGB colors
- Point cloud to binary file converter (editor plugin)
 - Saves point cloud data as binary file (which is much faster to read than parsing text files)
 - Import formats: XYZ, XYZRGB, CGO, ASC, CATIA ASC, PLY (ASCII), Brekel Binary, .LAS, .PTS
 - Option to scale point locations to Unity meters (usually they are in mm)
 - Option to Flip Y<>Z values (in Unity Y is up)
 - Option to AutoOffset near world 0,0,0 point
 - Option to add custom offset
- Custom shaders included
 - Single Color, RGB colors, Color+Alpha
- For Non-DX11: Point cloud to mesh assets converter
 - Options: Vertices per mesh, material
 - Cannot display as many points as DX11, but still useful.
 - Import vertex normals (PLY format only)
- Experimental Brekel Binary & Animated Binary support

REQUIREMENTS

- Unity 5.x (old version works with 4.x)
- DX11 support on your GPU (PC)
More info here: <http://docs.unity3d.com/Documentation/Manual/DirectX11.html>
- Point cloud to mesh assets converter works without DX11
 - (but those meshes DO NOT seem to work on all mobile devices?)

HISTORY

See “changes.txt” file from Documentation folder

FILES IN THE PACKAGE

Files inside the “**PointCloudTools/**” folder (v1.6)

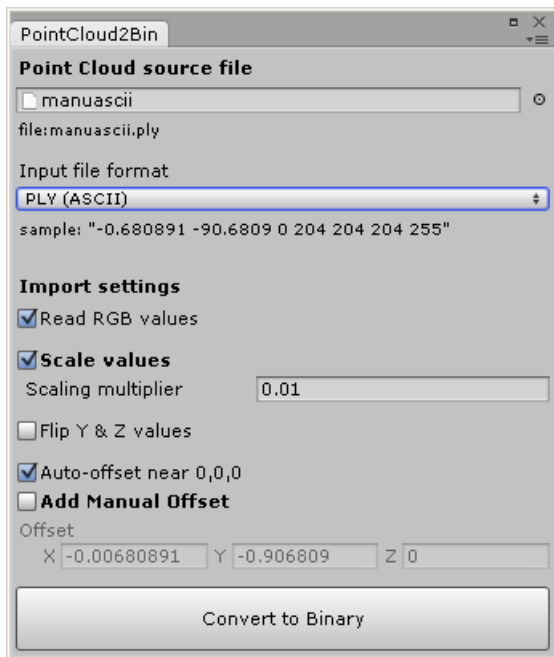
FOLDER / FILE	INFO
Common / Scripts / EscToQuitViewer	Simple script to quit application on ESC
Common / Scripts / FramesPerSecond	UnifyWiki FPS Counter
Common / Scripts / SimpleCameraFly	Basic flying camera for viewer scene
Common / Scripts / SimpleSmoothMouseLook	Smooth mouselook (from unity forums)
Documentation / changes.txt	Small notes about what has been changed
Documentation / PointCloudViewerTools##.pdf	(this file)
Editor / PointCloud2BinaryConverter.cs	Editor plugin: point cloud to binary converter
Editor / PointCloud2MeshConverter.cs	Editor plugin: point cloud to mesh converter
Editor/*	Other small extra tools
Examples / Mobile / *	Mobile test, point meshes and touch controls
Examples / PushVertices / *	Example with vertex push around given point
Materials / Shaders / PointCloudColorsDX11	Shader for RGB point clouds (dx11)
Materials / Shaders / PointCloudColorsMesh	Vertex color shader (non-dx11)
Materials / Shaders / PointCloudColorsMeshAlpha	Shader with Queue = Transparent
Materials / Shaders / PointCloudMeshColorsSize	Shader with point size property (non-dx11)
Materials / Shaders / PointCloudColorsNormalsMesh	Shader with vertex normals (non-dx11)
Materials / Shaders / PointCloudMeshSingleColor	Single color shader (non-dx11)
Materials / Shaders / PointCloudMeshSingleColorSize	Single color shader with point size (non-dx11)
Materials / Shaders / PointCloudSingleColorDX11	Single color shader for point clouds (dx11)
Materials / Shaders / vertexColorsDisplacePseudoLight	(non-dx11) Point brightness & displacement based on distance to given position
Materials / PointCloudColorsDX11	Material for RGB point clouds (dx11)
Materials / PointCloudColorsMesh	Vertex color material (non-dx11)
Materials / PointCloudColorsMeshPointSize	Material with point size property (non-dx11!)
Materials / PointCloudColorsNormalsMesh	Material with vertex normals (non-dx11)
Materials / PointCloudMeshSingleColor	Single color material (non-dx11)
Materials / PointCloudSingleColorDX11	Single color material for point clouds (dx11)
PointCloudBinaryViewerDX11/Scenes/BinaryViewerDX11	Binary viewer demo scene
PointCloudBinaryViewerDX11/WebplayerViewerDX11	Binary viewer demo scene (for webplayer)
PointCloudBinaryViewerDX11/Scripts/BinaryViewerDX11	Binary reader & viewer
PointCloudBinaryViewerDX11/WebplayerViewerDX11	Binary reader & viewer (webplayer)
Resources / sample.bin	Sample point cloud binary data
Standard Assets (Mobile) / *	Unity mobile standard assets
Extras/ *	Miscellaneous extras

USAGE INSTRUCTIONS

Point cloud To Binary Converter

This tool converts single point cloud file into custom binary data file. (much faster to read with viewer).

Start this tool from: **Window / Convert point cloud to binary**

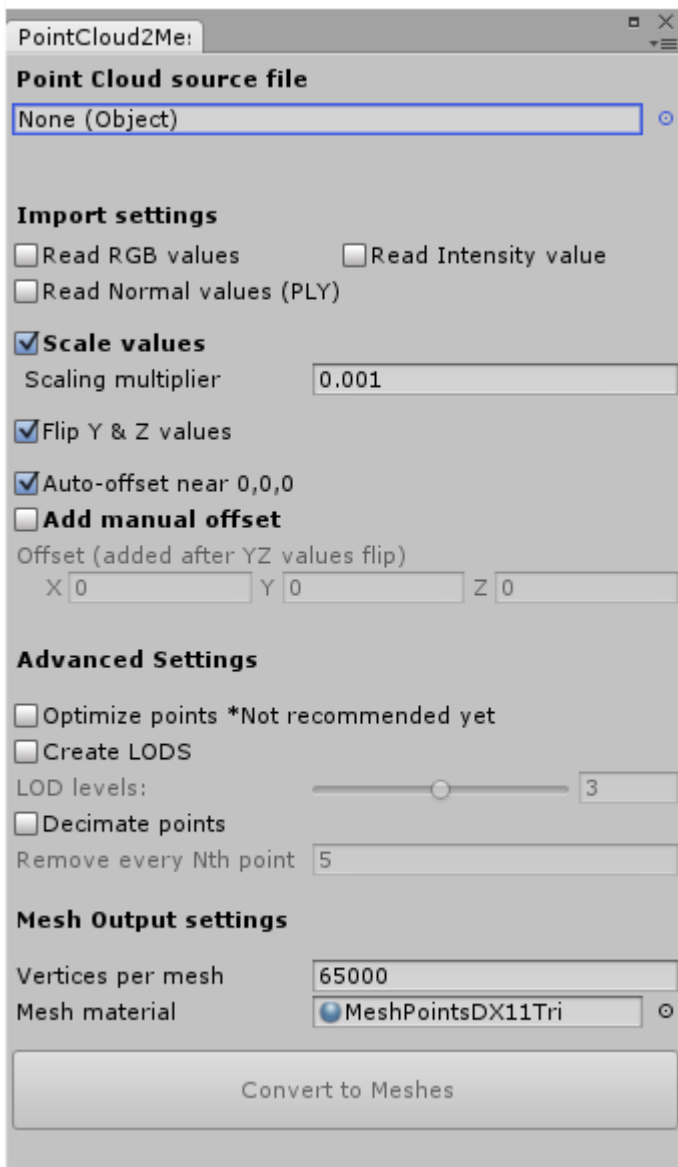


- Point Cloud source file:
 - Input file to read (object field)
 - Filename is displayed under this field
- Input file format
 - You have to manually select correct format (actual file extension doesn't matter)
 - More detailed point cloud data samples at the end of this document
- Import settings
 - [x] Read RGB values : yes or no
 - [x] Scale values : multiplies XYZ values with given value
 - Example: your data is in millimeters, set Scaling multiplier to 0.001
 - [x] Flip Y & Z values
 - In Unity Y is UP
 - [x] Auto-offset near 0,0,0
 - If the first row of your data is "10.1192 5.2643 0.5481", then we'll use that as an offset, first point location becomes: X-10.1192, Y-5.2643, Z-0.5481 = 0,0,0 and later points will have this same offset subtracted from XYZ data.
 - [x] Add Manual Offset
 - You can also add manual offset, that value is subtracted from XYZ data.
- Convert to Binary
 - Starts the conversion progress
 - Output file dialog will be displayed (this is the actual binary file to save into)

Point cloud to Mesh assets Converter

This tool converts single point cloud file into multiple unity mesh assets (saved into output folder) and also added into current scene.

Start this tool from: **Window / Convert point cloud to unity meshes**



- Point Cloud source file:
 - Input file to read (object field)
 - Filename is displayed under this field
- Import settings
 - [x] Read RGB values : yes or no
 - [x] Read Normal values : yes or no (*Only for PLY (ascii))
 - [x] Scale values : multiplies XYZ values with given value
 - Example: your data is in millimeters, set Scaling multiplier to 0.001
 - [x] Flip Y & Z values
 - In Unity Y is UP

- [x] Auto-offset near 0,0,0
 - If the first row of your data is "10.1192 5.2643 0.5481", then we'll use that as an offset, first point location becomes: $X-10.1192$, $Y-5.2643$, $Z-0.5481 = 0,0,0$ and later points will have this same offset subtracted from XYZ data
- [x] Add Manual Offset
 - You can also add manual offset, that value is subtracted from XYZ data
- Advanced Settings
 - [] Optimize Points *Does not work properly yet (loses precision on sort?)
 - [] Create LODS : Creates lower resolution LOD meshes ([read more](#) about lods)
 - [] Decimate points : Removes every #nth point
- Mesh output settings
 - Vertices per mesh (max 65000), point cloud is split into multiple meshes, each containing this amount of vertices (*Warning: low values could results millions of files*)
 - Mesh material : Assign meshes with this material (Should use vertex material shaders)
- Convert to Meshes
 - Starts the conversion progress
 - Output file dialog will be displayed (this is the actual output folder where mesh assets will be saved into)

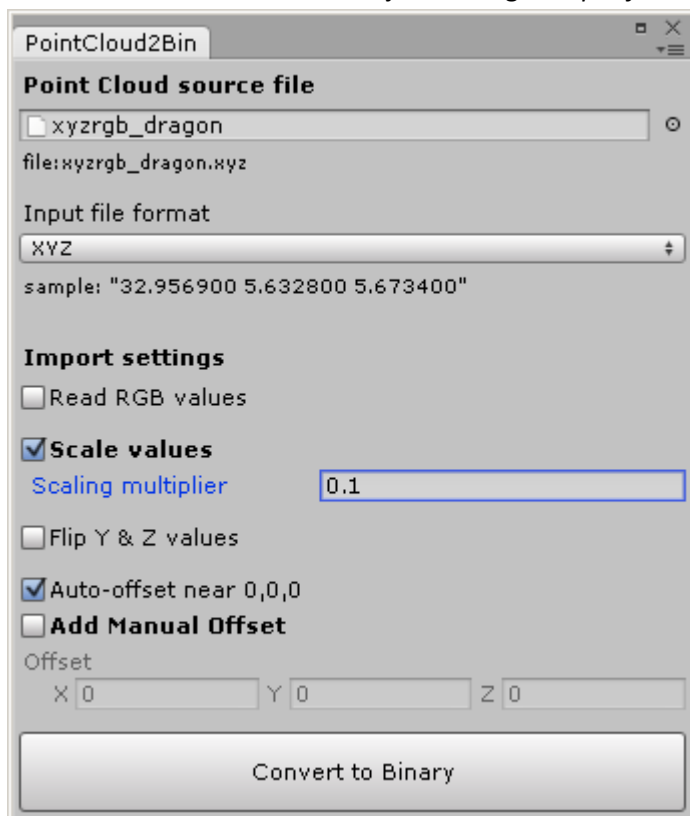
Using the Webplayer Viewer

- Open "WebplayerViewerDX11" scene
- Select "@WebplayerViewer" gameobject from hierarchy
- Set your URL (address from where the binary data will be loaded)
- Example url is included (but may be removed later, if it brings too much traffic..)
<http://unitycoder.com/upload/demos/PointCloudViewerDX11/web1/sample.bin>
- Note! You may need to have the "crossdomain.xml" on your server :
<http://docs.unity3d.com/Manual/SecuritySandbox.html>

TUTORIAL

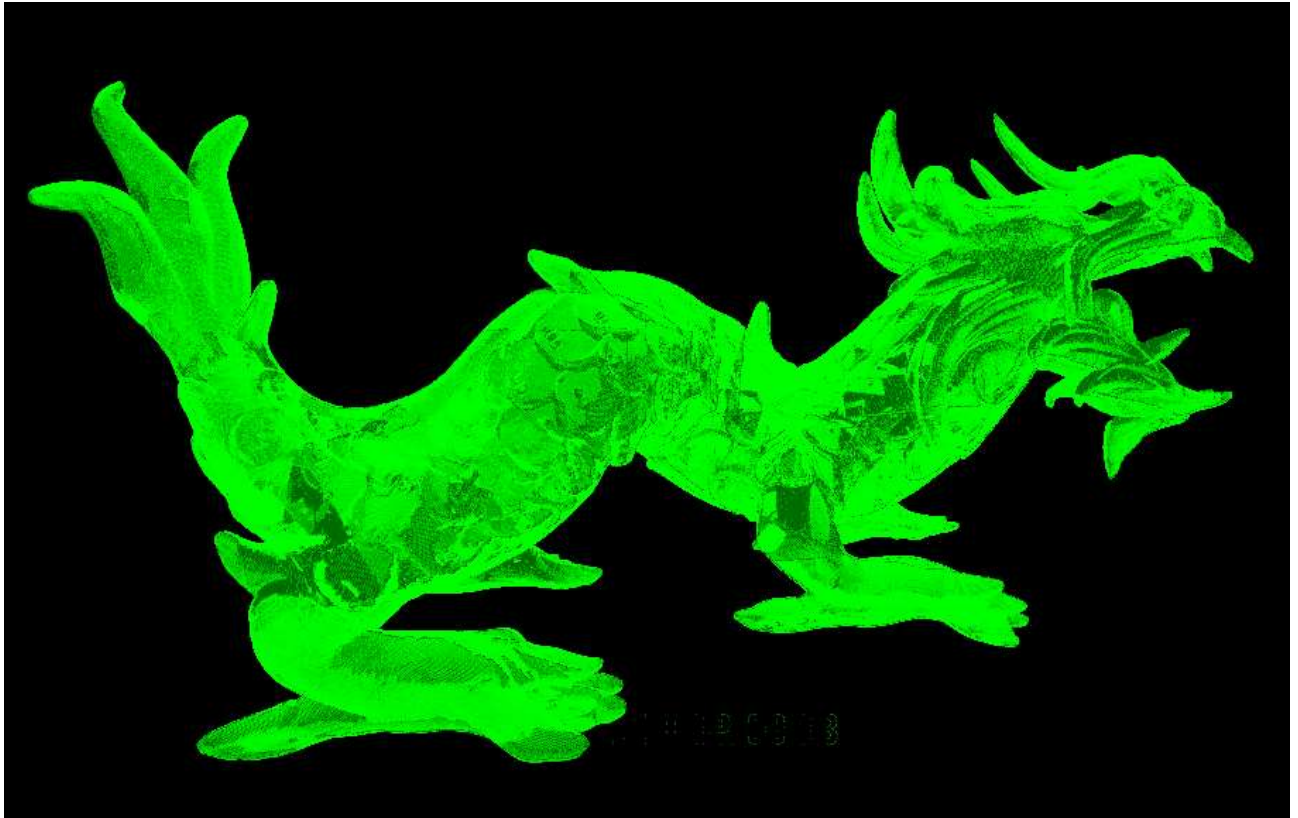
Stanford Dragon to XYZ

- Download http://graphics.stanford.edu/data/3Dscanrep/xyzrgb/xyzrgb_dragon.ply.gz
- (Unfortunately it doesn't include RGB data even though it says so..?)
- Unzip it
- Open the file "xyzrgb_dragon.ply" in MeshLab <http://meshlab.sourceforge.net/>
- File / Export Mesh as...
- Select format "XYZ point cloud (with or without normals) (*.xyz)"
- Choose saving options: [] Normal (disable normals, they don't work so well yet)
- Press ok, file is now saved
- Now put the "xyzrgb_dragon.xyz" into your unity PointCloudTools/SampleData/ folder
- Then start: Window / PointCloudTools / Convert Point Cloud to Binary
- Then select file as source and adjust settings: **Input format selection is no longer used in v1.6+*



- Click Convert to Binary
- Set output folder & filename
- Now open BinaryViewerDX11 scene from:
- PointCloudTools / PointCloudBinaryViewerDX11 / Scenes / BinaryViewerDX11.scene
- Select "@BinaryViewer" object from hierarchy window
- Set base folder name, for example: "PointCloudTools/Output/"
- Set filename, for example: "xyzrgb.bin"
- Set material, for example: "PointCloudSingleColorDX11" (single color shader)
- Press Play (run)

- You should see this:



Building EXE from this tutorial

- Place the point cloud binary file inside "StreamingAssets/" folder, then it gets included to the build (remember to adjust "@BinaryViewer" path also)
- Or if you use other folder, you can copy it manually to the build data folder: "Data/YourFolder/"-folder

USING BREKEL PRO POINT CLOUD DATA

Unity Point Cloud Viewer v1.4 has experimental support added for viewing Brekel Pro PointCloud data.

More info about Brekel Pro PointCloud v2": <http://brekel.com/brekel-pro-pointcloud-v2/>

*Currently supported version is v2.08 to v2.??

EXPORTING DATA FROM BREKEL SOFTWARE

- (more guide to be added, but it's just a matter of saving/exporting the data into unitycoder point cloud binary format *.bin)

VIEWING BREKEL DATA IN UNITY

- Copy file inside Unity Assets/ folder (you can use any subfolders also)
- Open "BinaryViewerDX11.scene"
- Select " @BinaryViewer" from hierarchy
- Set BaseFolder and Filename fields to point into your file
- Note. Filename needs ".bin" extension
- ☐ Is animated, Enable this ONLY if the data contains animated frames and also adjust the playback delay time
- Click "Play" to test it

CONVERT BREKEL BINARY DATA INTO MESH ASSETS

- (Single binary file/frame conversion is not currently supported, will be added later)

CONVERT ANIMATED BREKEL DATA INTO MESH ASSETS

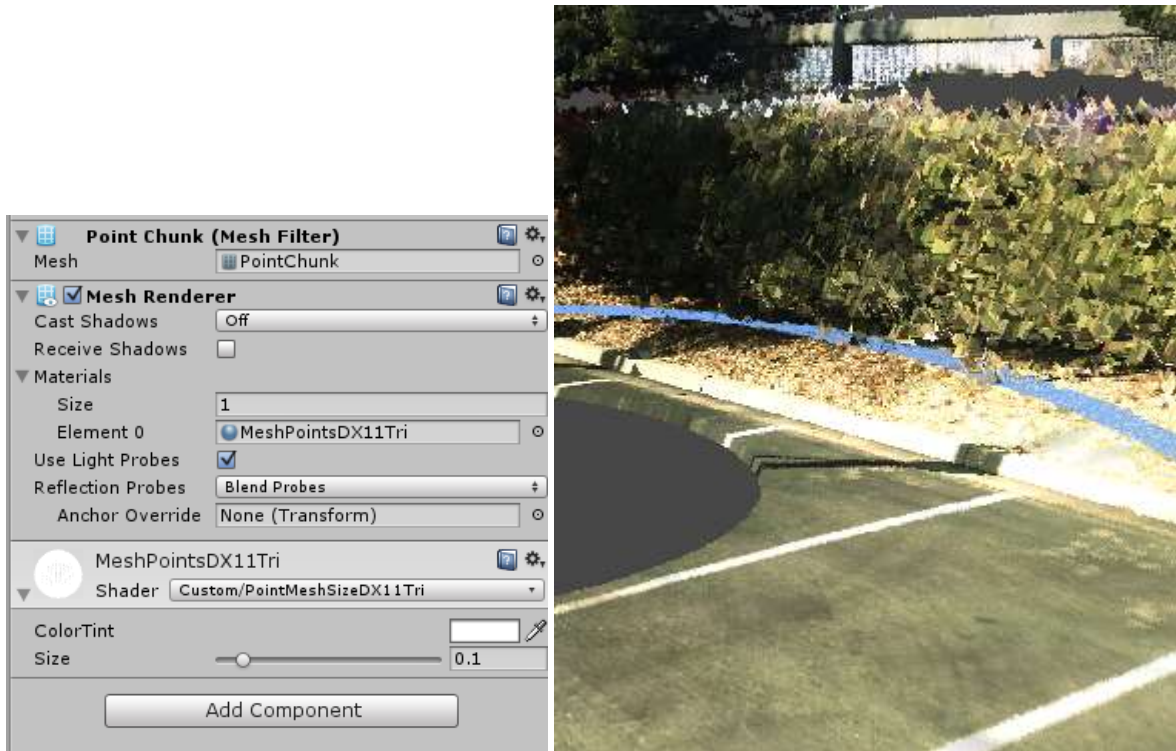
- Window/PointCloudTools/Convert Point Cloud to Unity Meshes
- Assign your binary file into "Point Cloud source file"- field
- Input file format: BIN (Brekel Animated Binary)
- ☒ Read RGB values *Should be usually enabled
- ☐ Scale values *No need to scale, unless you require to
- ☐ Flip Y & Z values *Not needed
- ☐ Auto offset near 0,0,0 *Enable if required
- ☐ Add manual offset * Enable if required
- Vertices per mesh: 65000
- Mesh material: PointCloudColorsDX11 *Use with RGB color clouds
- Click Convert to Meshes button
- Save dialog (its good idea to create new empty directory for the mesh assets)
- *Meshes are also added to the current open scene

POINT SIZE FOR MESH IN DX11 MODE

DX11 mode doesn't support shader point size, current work-around is to use these shaders:

<https://gist.github.com/unitycoder/dd2761962789f3473c06>

- Save those 2 shaders to your project (for example PointCloudTools/Materials/Shaders/ folder as "MeshPointSizeDX11Quad.shader" and "MeshPointSizeDX11Tri.shader"
- Then create new material for them and use that material on your meshes



CUSTOM BINARY FORMAT

Point Cloud Tools saves data to custom binary format.

Saving happens around line 356 inside "PointCloud2BinaryConverter .cs" and file structure is:

binaryVersion (byte)

- 0 = original, default format (not used)
- 1 = default binary data (currently used)
- 2 = custom animated Brekel binary frames data *rest of the binary format is different for this data

numberOfPoints (Int32)

readRGB (bool)

// then rest of the data is XYZ coordinates and RGB data (if any)

x (float), y (float), z (float), r (float), g (float), b (float)

x (float), y (float), z (float), r (float), g (float), b (float).....

REFERENCES / SHOWCASE

Point Cloud Tools were used:

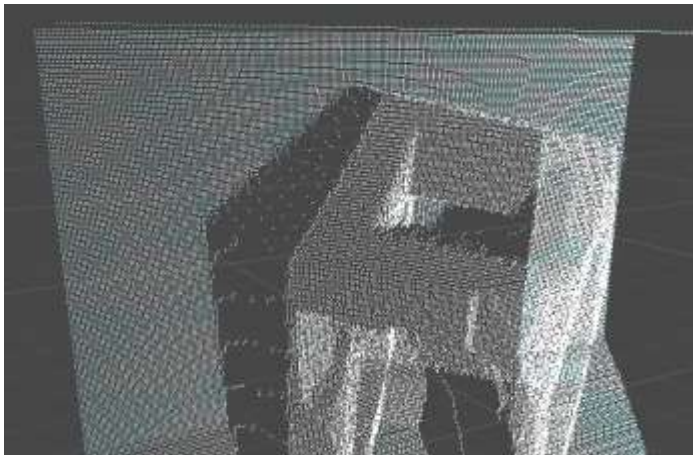
<http://datumexplorer.universalassemblyunit.com/>

EXTRAS FOLDER

Some misc extras/tests will be added in this folder. Currently it has:

VideoExtrude/

- Custom shader to extrude vertices based on depth map
- You can assign 2 videos to the "VideoDepthExtrude"-material, first is used for getting color info and 2nd video (depth map / grayscale) is used for extruding
- You can probably get color + depth map video recorded with Kinect



- Video Depth Conversion tools : <http://research.microsoft.com/en-us/downloads/29d28301-1079-4435-9810-74709376bce1/>

FAQ

- It doesn't work / Screen is Black / Cannot see points
 - Make sure your Unity is in DX11 mode (titlebar should have text <DX11>)
 - More info here: <http://docs.unity3d.com/Documentation/Manual/DirectX11.html>
 - Try using different material(shader), if you try to use "PointCloudColorsDX11" without the model having RGB data, it will show black..
 - Try adjusting camera background color, maybe the points are black
- It doesn't work when build exe (dx11 viewer)
 - See the error message on screen, it says that cannot find the point cloud file. So copy it into that folder
 - Or, add the point cloud files into StreamingAssets/ folder, they are included to the build
- How many points it can display? (dx11)
 - Up to ~40 million (single color)
 - Around ~20 million (using colors)
- Does this work on mobile devices?
 - No (well, the dx11 version doesn't)
- Does this work on Apple Mac's?
 - No DX11 mode there, so it won't.
 - Unity mesh assets should work (non-dx11)
- Does this work in web player? (with pc & DX11 hardware)
 - Not, because "using System.IO;" is used for reading the file
 - You would need to load the point data some other way
- Can I show multiple point clouds?
 - With DX11 only 1 point cloud at a time, so you need to combine points to a single cloud
 - Using unity mesh assets, you can display as many clouds as your computer can handle..(but it won't be much more than 10-15m points total, each mesh object is 65k points)
- Does it work with Unity 4.x?
 - Yes, but not updating the old version anymore.. Only 5.x version is updated.
- Does it work with Unity Indie (free version of Unity) ?
 - Yes
- Can I adjust point size for DX11 viewer?
 - Not for DX11 (point size seems to work only with non-dx11 mode, with unity meshes)
 - V1.7 has link for shader that supports MeshPoint sizes in DX11 mode also (billboarded vertex quads or tri's)

KNOWN ISSUES

Notification message when converting point cloud to mesh assets:

“Shader wants normals, but the mesh cloud001 doesn't have them
UnityEditor.AssetDatabase.CreateAsset(Object, String)”

EXTERNAL TOOLS

Useful tools for point cloud related tasks (file format conversions etc)

- <http://meshlab.sourceforge.net/>
 - Convert PLY binary to PLY Ascii, and between other formats
- <http://www.faro.com/faro-3d-app-center/stand-alone-apps/scene-lt>
 - Viewer, Converter etc.

SAMPLE POINT CLOUD DATA(S) FROM WEB

- <http://graphics.stanford.edu/data/3Dscanrep/>
 - Note. Convert PLY Binary into PLY Ascii with MeshLab
- <http://www.libe57.org/data.html>
 - Can use Scene LT for converting
- <http://kos.informatik.uni-osnabrueck.de/3Dscans/>
- <http://www.liblas.org/samples/>
- <https://www.ucl.ac.uk/3dim/pointcloud>

SUPPORTED FILE FORMAT/ HEADER EXAMPLES

XYZ

```
32.95690000 5.63280000 5.67000000
32.95690000 5.63280000 5.67000000
32.95690000 5.63280000 5.66990000
```

XYZRGB

```
32.95690000 5.63280000 5.67000000 220 220 220
32.95690000 5.63280000 5.67000000 220 220 220
32.95690000 5.63280000 5.66990000 220 220 220
```

CGO

```
2295027
683,099976 880,200012 5544,700195
704,700012 879,400024 5538,500000
730,099976 877,699951 5530,799805
```

ASC

```
-1192.9 2643.6 5481.2
-1188.3 2627.2 5502.4
-1184.4 2618.8 5527.2
```

CATIA ASC

```
!
! ASCII generated by Digitized Shape Editor/CATIA
!
! Without Scans
! Point Format          = 'X %f Y %f Z %f'
!
! Total number of points = 353655
!
X 31022.199219 Y -3314.100098 Z 6152.500000
X 31024.000000 Y -3316.099854 Z 6187.100098
X 31023.199219 Y -3316.400146 Z 6162.399902
```

PLY (ASCII) **v1.3 also normals can be read from PLY (ascii)*

```
Ply
format ascii 1.0
comment VCGLIB generated
element vertex 2155617
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
property uchar alpha
element face 4305818
property list uchar int vertex_indices
end_header
-0.680891 -90.6809 0 204 204 204 255
-0.680891 -89.3191 0 204 204 204 255
-0.815892 -90.8159 0 204 204 204 255
```

LAS

(binary format #2)

http://www.asprs.org/a/society/committees/standards/LAS_1_4_r13.pdf

PTS

155201

0.53060000 -2.99310000 1.58400000 1617 246 244 213

0.51360000 -2.97220000 1.57180000 1729 254 245 219

0.49800000 -2.95860000 1.56340000 1473 235 236 205

SUPPORT & FEEDBACK

Send comments / feedback & request to my blog:

<http://unitycoder.com/blog/2014/03/19/asset-store-point-cloud-viewer-unity/>

**Unity asset store forum link will be added there later*

Email: support@unitycoder.com

- Add product name: "PointCloudViewerDX11" to the subject
- Include screenshot(s) if something strange is happening (helps solving the problem)
- Having problem with your own point cloud data?
Include sample how the data is formatted or what format it is,
example data row:
234234.1239843 8473.12323 899834.2138585 128 128 128

Or, Using Unity3D forums private message: *but email is preferred

<http://forum.unity3d.com/members/mgear.22727/#info>

Or, You can also post comments/questions here:

<http://unitycoder.com/blog/2012/09/01/xyz-point-cloud-data-viewer-dx11/>

Linkedin: <http://fi.linkedin.com/pub/mika-makkonen/68/992/22> (could be interested on .SG/.MY/.UK - Unity jobs)

** If you need some custom solutions, ask quote for a paid work, but if it's some common feature request, post to forum as "new feature request/wish"**