

2/11/2025

UNITYCODER.COM

POINT CLOUD VIEWER & TOOLS

v3.0

Point Cloud Viewer & Tools

Table of Contents

ONLINE DOCUMENTATION.....	2
Online documentation with most recent updates: https://github.com/unitycoder/UnityPointCloudViewer/wiki	2
HISTORY	2
USAGE INSTRUCTIONS.....	2
Builtin RenderPipeline	2
Setup for URP	2
Setup for HDRP	2
Point cloud To Binary Converter	3
Point Cloud source file.....	3
Input file format	3
Import settings	4
Point cloud to Mesh assets Converter	5
Point Cloud source file.....	5
Import settings	5
Advanced Settings	6
Mesh Output Settings.....	6
Using the Old Unity Webplayer *This is deprecated.....	6
TUTORIALS.....	7
USING BREKEL PRO POINT CLOUD DATA.....	9
POINT SIZE FOR MESH IN DX11 MODE.....	10
CUSTOM BINARY FORMAT	10
OLD REFERENCES / SHOWCASE	10
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	15

Point Cloud Viewer Tools

ONLINE DOCUMENTATION

Online documentation with most recent updates:

<https://github.com/unitycoder/UnityPointCloudViewer/wiki>

HISTORY

See "changes.txt" file from Documentation folder

USAGE INSTRUCTIONS

Builtin RenderPipeline

This package is uploaded for BIRP, so it works as is.

Setup for URP

- Online guide: [https://github.com/unitycoder/UnityPointCloudViewer/wiki/SRP-\(URP-&-HDRP\)#setup-for-urp](https://github.com/unitycoder/UnityPointCloudViewer/wiki/SRP-(URP-&-HDRP)#setup-for-urp)
 - Install URP
 - Enable URP in Project Settings/Graphics/, assign "URP Asset" to SRP pipeline setting field *Or you can use your own URP rendering asset too..
 - You only need to do next steps, IF you want to use VR or you have WorldSpaceCanvas in the scene (or other rendering issue with URP)
 - Open V3 sample scene: Assets\PointCloudTools\Demos\PointCloudViewer\Scenes\PointCloud-TilesViewerV3.unity
 - From inspector Enable SRP: [x] Use URP Custom Pass
 - Select **PointCloudTools/SRP/URP Asset_Renderer** (or assign to your own urp rendering asset)
 - Click Add Renderer Feature
 - Select PointCloudCustomPassURP from the dropdown
 - Press Play!
- *tested on 2021.3.19f1 (regular URP) & 6000.0.32f1 (uses RenderGraph)

Setup for HDRP

- Install HDRP
- Enable HDRP in Project Settings/Graphics/, assign "HDRRenderPipelineAsset" to SRP pipeline setting field
- Open V3 sample scene: Assets\PointCloudTools\Demos\PointCloudViewer\Scenes\PointCloud-TilesViewerV3.unity

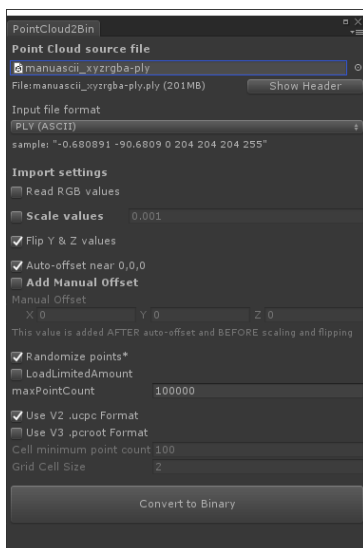
- From inspector Enable SRP: [x] Use HDRP Custom Pass
- Select MainCamera
- Add "CustomPassVolume" component
- In custom volume, Press + and add "PointCloudCustomPassHDRPV3" (this is for V3 viewer)
- Press + to Add Viewers and assign "@TilesViewer V3" to that field (you can assign multiple viewers here)
- Press Play!

Point cloud To Binary Converter

This tool converts single point cloud file into custom binary format file. (much faster to load in viewer).
Usually good place to save your output files is inside StreamingAssets/ folder (so they get included with the build)

Online guide: <https://github.com/unitycoder/UnityPointCloudViewer/wiki/PointCloud2Binary>

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



Point Cloud source file

- Point Cloud input file that you want to convert into binary point cloud
- Filename & size is displayed under this field
- New: Click "Show Header" button to print few lines from the file (this is useful for checking what data it contains, and what scale the point data looks like)

Input file format

- You have to manually select correct import format

Import settings

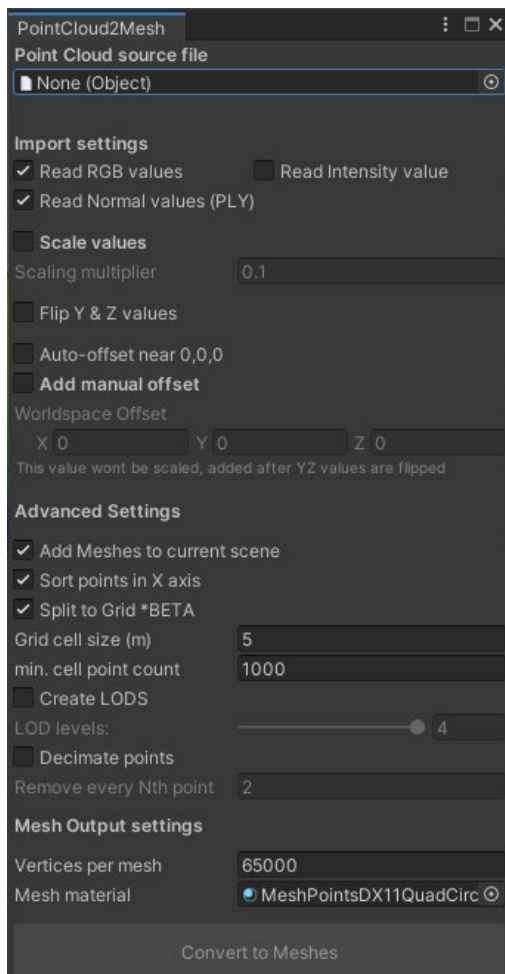
- Read RGB values : Reads color data (if included in the file and if supported in selected import format)
 - Note: V2 and V3 formats require RGB data (otherwise cloud is black or might not load)
- Read INT/REF : Only visible for .PTS. Allows reading intensity or reflectivity value. This can be useful with gradient shader to map that intensity into color range.
 - Additional 0-255 toggle appears, if enabled, intensity range is expected to be in 0-255 range. Otherwise -2048 to 2047 is expected.
- Scale values : multiplies XYZ values with given value (used for scaling points into meters)
- Flip Y & Z values : In Unity Y is UP, so need to swap the values
- Auto-offset near 0,0,0 : If the first row of your data is for example "10.1192 5.2643 0.5481", then we'll use that as an offset value, first point location becomes: $X-10.1192=0$, $Y-5.2643=0$, $Z-0.5481=0$ and later points will have this same offset subtracted from XYZ data, so the cloud will be nearer to 0,0,0 position in scene
- Add Manual Offset : You can also add manual offset, that value is subtracted from XYZ data
- Randomize points : Randomizes point order, to use dynamic resolution (this is always enabled for V3, optional for V2)
- Load Limited Amount : Reads only this many points. Good for testing your import settings and scaling, before loading large cloud
- Use V2 .ucpc format : Saves into V2 format, it loads faster than V1 .bin
- Use V3 .pcroot format : Saves into V3 format, this is special tiled+lod format for large clouds
- Cell minimum point count : Skip tiles that has less than this amount of points, good for cleaning up stray points, improves performance (V3 only)
- Grid Cell Size : V3 Format splits cloud into tiles of this size. If you have large outdoor scan, this might need to go up to 30 (meters), for smaller areas few meters can be good value. If you set this value too small for large cloud, you get too many tiles. (V3 only)
- Convert to Binary button : Opens output dialog (to specify filename and path)

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.

Online guide: <https://github.com/unitycoder/UnityPointCloudViewer/wiki/PointCloud2Mesh>

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



Point Cloud source file

- Point Cloud input file that you want to convert into binary point cloud
- Filename & size is displayed under this field

Import settings

- Read RGB values : Reads color values (usually expected to be in this order X Y Z R G B)
- Read intensity value : Reads intensity value (usually expected to be in this order X Y Z I)
- Read normal values (PLY) : Reads vertex normals and assigns them to output mesh, only supported on PLY format currently
- Scale values : multiplies XYZ values with given value (for scaling between millimeters / meters etc)
- Flip Y & Z values : In Unity Y is UP

- Auto-offset near 0,0,0 : If the first row of your data is for example "10.1192 5.2643 0.5481", then we'll use that as an offset value, first point location becomes: $X-10.1192=0$, $Y-5.2643=0$, $Z-0.5481=0$ and later points will have this same offset subtracted from XYZ data, so the cloud will be nearer to 0,0,0 position in scene
- Add Manual Offset : You can also add manual offset, that value is subtracted from XYZ data

Advanced Settings

- Add Meshes to current scene : Disable this if you convert heavy clouds (because unity can hang with large amount of meshes), so meshes are saved as assets only (not added to scene automatically)
- Sort points in X axis : Sorts point by X axis, so that mesh pieces get culled from view better *But not recommended yet, because in some cases the points lose lots of precision when they are sorted(?)
- Split to Grid: Splits meshes into grid cells, so that they get culled better (try with lods enabled too)
- Create LODS : Experimental features, creates LOD meshes with less points (but lods are not grouped yet, so it doesn't decrease draw calls.. just less vertices on screen)
- Decimate points : Removes every Nth point *You should rather use real point cloud filtering tools to create evenly distributed points (like the filters in Catia)

Mesh Output Settings

- Vertices per mesh : 65000 is the maximum vertices on a single mesh object
- Mesh material : For meshes, you should use the non-dx11 materials, or you can test any other materials and shaders also. Shader needs to support vertex colors, if you want to see point cloud in colors.

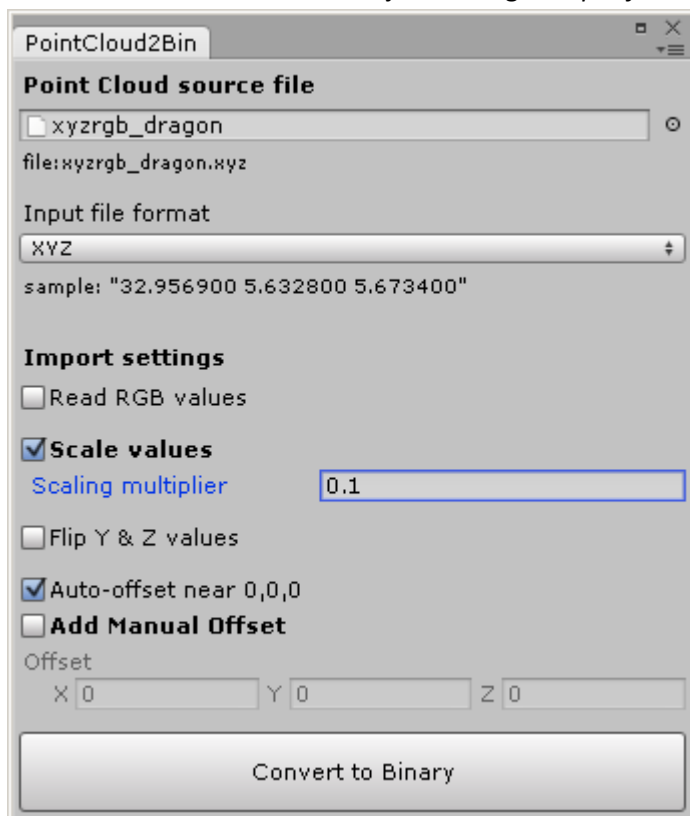
Using the Old Unity Webplayer *This is deprecated

- 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>

TUTORIALS

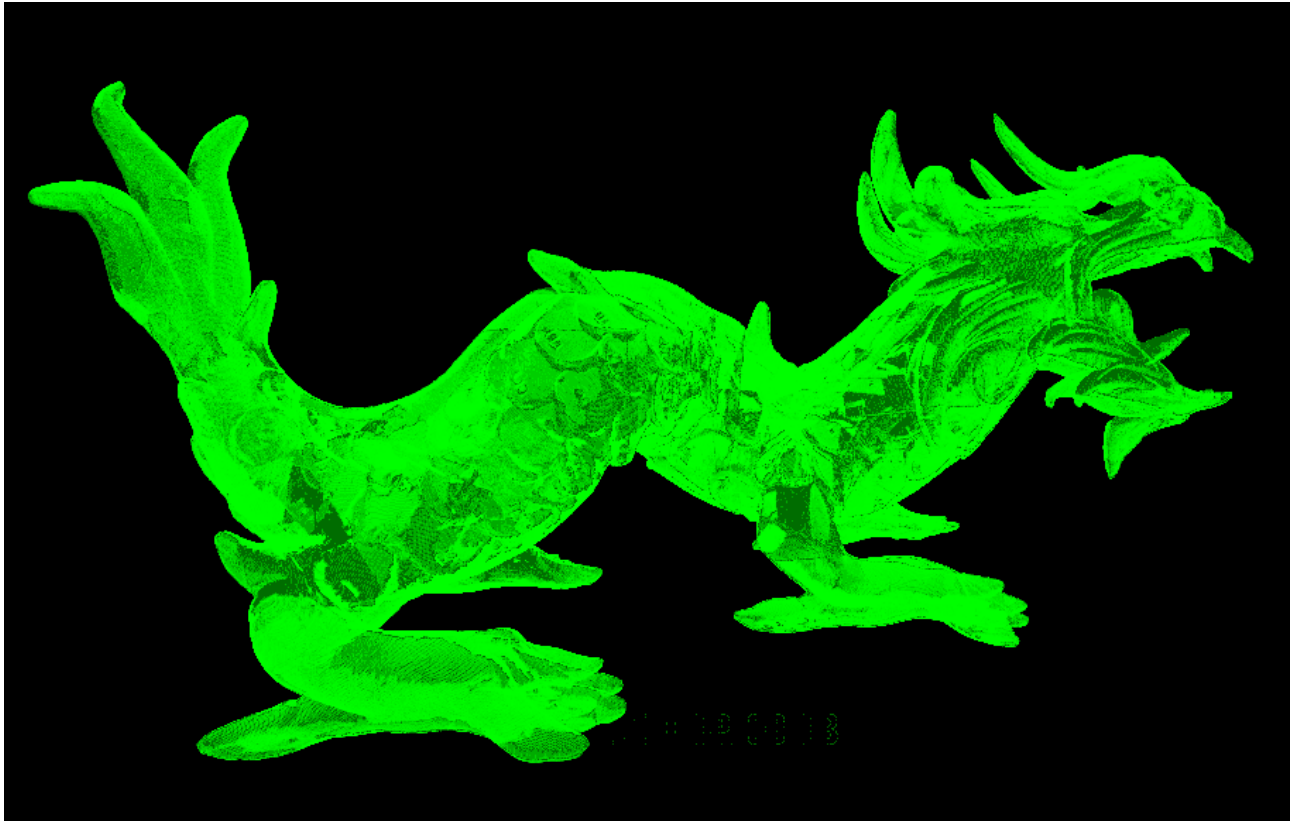
Stanford Dragon to XYZ

- 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

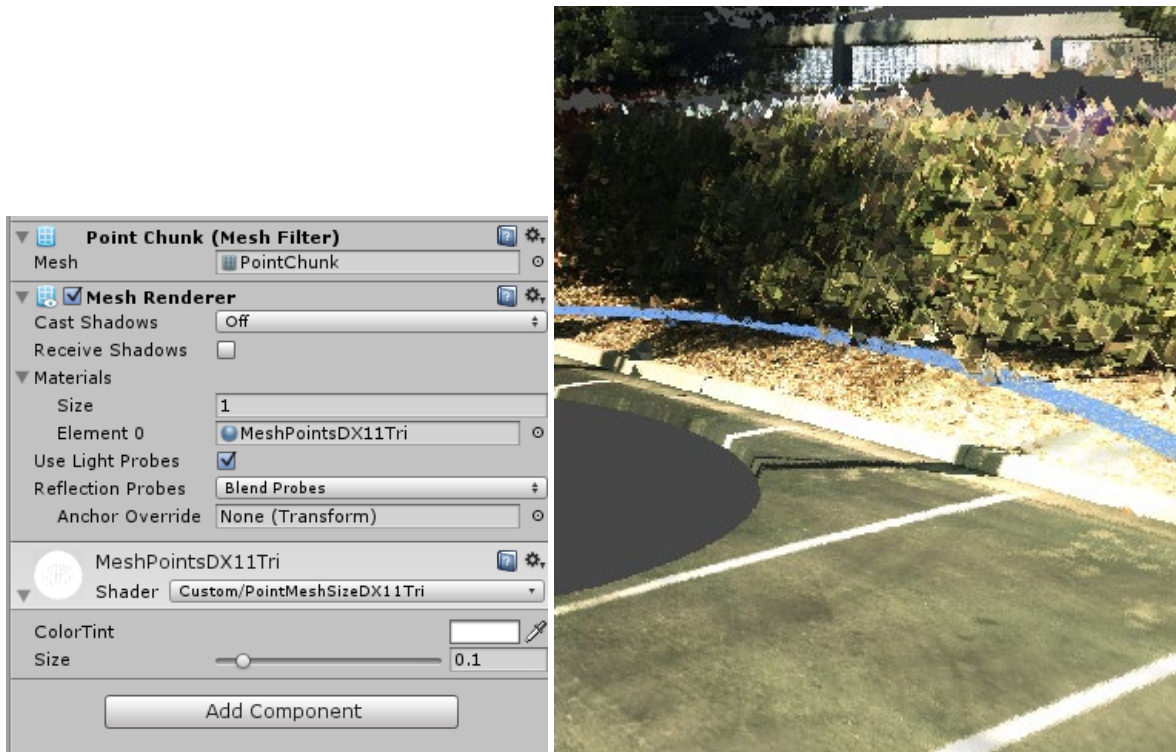
- (Single binary file/frame conversion is not currently supported, will be added later)

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)
- [x] 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

Use Materials with name *DX11*Mesh* (they support geometry billboards for point size in Meshes with DX11)



CUSTOM BINARY FORMAT

View binary format here (v1, v2, v3 file format structures)

<https://github.com/unitycoder/UnityPointCloudViewer/wiki/Binary-File-Format-Structure>

OLD REFERENCES / SHOWCASE

Point Cloud Tools were used:

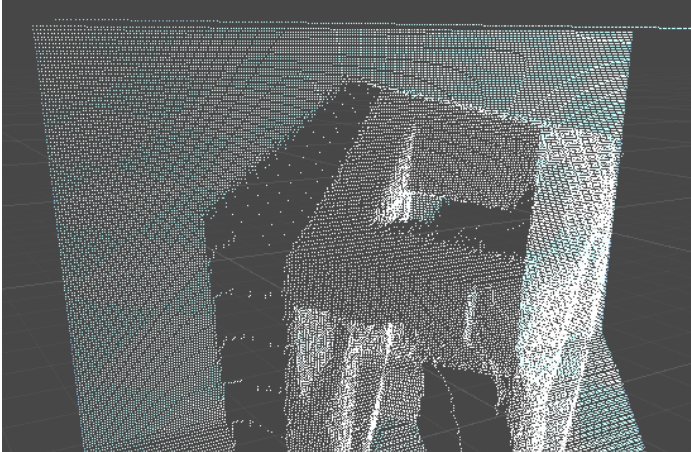
- <https://www.youtube.com/watch?v=AjuEogkFBGo>
- <http://datumexplorer.universallassemblyunit.com/>

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)
 - Have tested up to ~152 million points (with ~2fps in editor, gtx970m laptop)
- Does this work on mobile devices?
 - No (well, the dx11 version doesn't)
- Does this work on Apple Mac's?
 - There is experimental Apple Metal shader material included
 - 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)
- Can I adjust point size for DX11 viewer?
 - ~~Not for DX11 (point size seems to work only with non-dx11 mode, with unity meshes)~~
 - Yes, there are billboard shaders for point meshes now, supports DX11 mode
 - 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

As a point cloud viewer customer, you get 50% discount on this LAS/LAZ to GLTF conversion tool:

<https://las2gltf.kelobyte.fi/>

With this tool, you can convert point cloud files into GLB files and view them inside Unity as meshes, using those GLTF importers for Unity! (no need special point cloud viewers or materials).

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

- <https://www.danielgm.net/cc/>
 - Convert files
- <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

Github is used for latest bug reports, requests and documentation:

<https://github.com/unitycoder/UnityPointCloudViewer>

For private issues:

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

<https://discussions.unity.com/u/mgear/summary>

Linkedin: <https://www.linkedin.com/in/unitycoder/> (feel free to add contact!)

GLFH!