7/22/2014

UNITYCODER.COM

POINT CLOUD VIEWER & TOOLS

Unity Point Cloud Viewer & Tools

Point Cloud Viewer & Tools

Table of Contents

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)
  - 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)

REQUIREMENTS

- Unity 4.x (indie or pro)
- DX11 requires Windows Vista or later and at least a DX10-level GPU (preferably DX11-level)
  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?)

INSTALLATION

- Import package and you are done..

FILES IN THE PACKAGE

Files inside the “**PointCloudTools/**” folder:

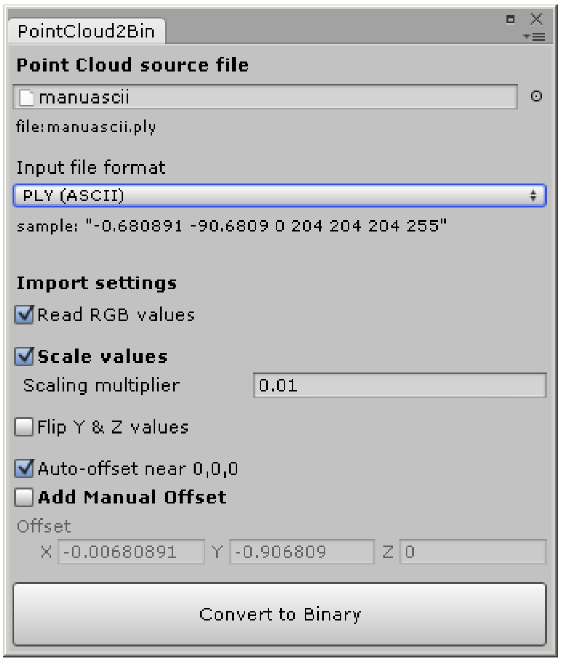
FOLDER / FILE	**INFO**
Common / Scripts / EscToQuitViewer	Simple script to quit application on ESC
Common / Scripts / SimpleCameraFly	Basic flying camera for viewer scene
Common / Scripts / SimpleSmoothMouseLook	Smooth mouselook (from unity forums)
Documentation / changes.txt	Small notes about what has been changed
Documentation / PointCloudViewerTools.pdf	(this file)
Editor / PointCloud2BinaryConverter	Editor plugin: point cloud to binary converter
Editor / PointCloud2MeshConverter	Editor plugin: point cloud to mesh converter
Materials / Shaders / PointCloudColorsDX11	Shader for RGB point clouds (dx11)
Materials / Shaders / PointCloudColorsMesh	Vertex color shader (non-dx11)
Materials / Shaders / PointCloudMeshColorsPointSize	Shader with point size property (non-dx11)
Materials / Shaders / PointCloudColorsNormalsMesh	Shader with vertex normals (non-dx11)
Materials / Shaders / PointCloudSingleColorDX11	Single color shader for point clouds (dx11)
Materials / Shaders / PointCloudMeshSingleColorMesh	Single color shader (non-dx11)
Materials / Shaders /	(shaders folder)
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)
Output /	(empty folder, can be used as output)
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

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

manuascii

file:manuascii.ply

Input file format

PLY (ASCII)

sample: "-0.680891 -90.6809 0 204 204 204 255"

Import settings

☑ Read RGB values

☑ **Scale values**

Scaling multiplier 0.01

☐ Flip Y & Z values

☑ Auto-offset near 0,0,0

☐ **Add Manual Offset**

Offset

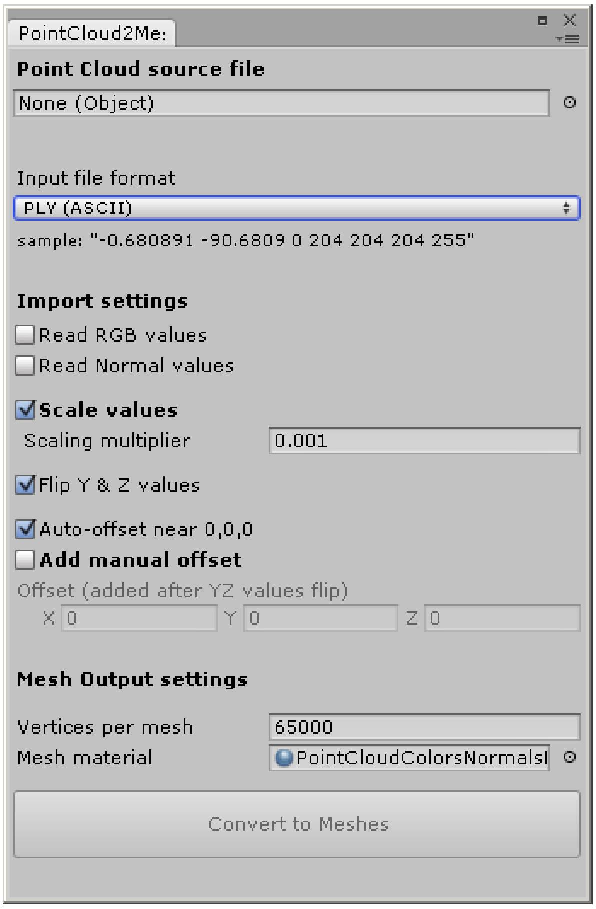
X -0.00680891 Y -0.906809 Z 0

- 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**



PointCloud2Me:

Point Cloud source file

None (Object)

Input file format

PLY (ASCII)

sample: "-0.680891 -90.6809 0 204 204 204 255"

Import settings

☐ Read RGB values

☐ Read Normal values

☑ **Scale values**

Scaling multiplier 0.001

☑ Flip Y & Z values

☑ Auto-offset near 0,0,0

☐ **Add manual offset**

Offset (added after YZ values flip)

X 0 Y 0 Z 0

Mesh Output settings

Vertices per mesh 65000

Mesh material PointCloudColorsNormalsI

- 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] 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
- 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)

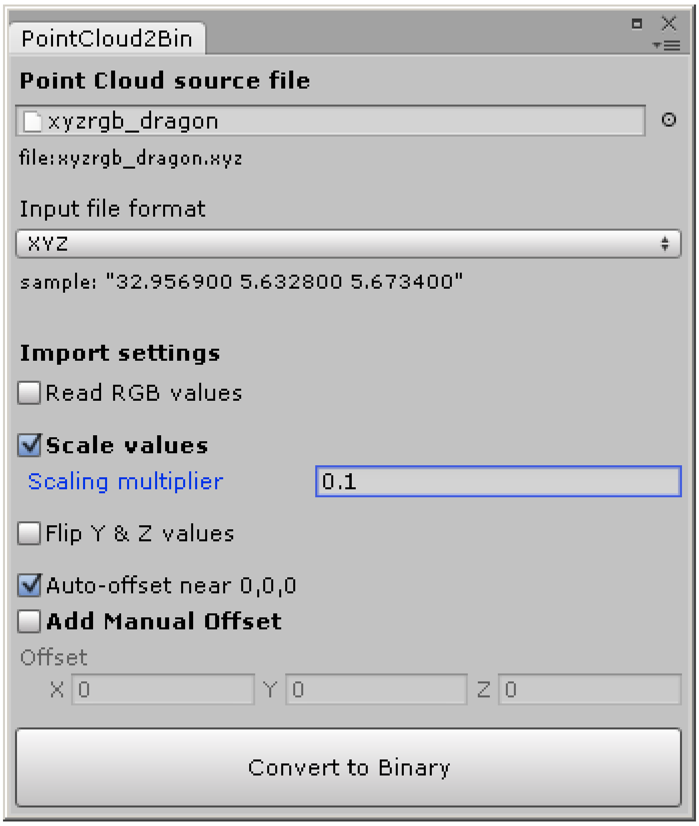
Using the Webplayer Viewer

- 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 this, reading normals is not yet supported)
- 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:



Point Cloud source file

xyzrgb_dragon

file:xyzrgb_dragon.xyz

Input file format

XYZ

sample: "32.956900 5.632800 5.673400"

Import settings

☐ Read RGB values

☑ **Scale values**

Scaling multiplier 0.1

☐ Flip Y & Z values

☑ Auto-offset near 0,0,0

☐ **Add Manual Offset**

Offset

X 0 Y 0 Z 0

- 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

- If your point cloud binary file is NOT inside "Resources/" folder, it won't be included in the build (remember to adjust "@BinaryViewer" path also)
- Or you can copy it manually to the build "Data/PointCloudTools/Output/" folder

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)
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)…..*

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

- 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

- 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 3.x?
    - No

- 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)

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

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
```