Semi-automatic A.I.-Specified Stereographic Scene Rendering Techniques Across Various Displays

Shunsuke Tanagi[1] and *Michael* Cohen[1*]

[1]Spatial Media Group, University of Aizu; Aizu-Wakamatsu, Fukushima 965-8580; Japan

Abstract. Semi-automatic, A.I.-specified stereographic scene processing is achieved by extending Blender with ChatGPT. We can generate objects and set scenes in Blender semi-automatically by invoking appropriate text prompts. Stereography techniques include anaglyphy, chromastereoscopy, and side-by-side display using binocular viewing. One can create a similar scene by copying and pasting the prompt used. It cannot always work identically, but it behaves roughly the same. First, we established Skybox and added scene Objects (Camera, Cube, etc.) in 3D view, materials, and animation with prompt. Stereographic images are created semi-automatically and compared across some qualities. The goal of this research is to validate those various stereographic techniques as rendered automatically in Blender. Keywords: A.I., binocular display, ChatGPT, compositing, prompt engineering, rendering, stereography, shading, text-to-image generation.

1 Introduction

Stereoscopy is making imagery that appears to have depth when viewed binocularly. The human eyes see different images in the left and right eye, and parallax between the eyes enables perception of images in three dimensions. Stereoscopic techniques include single-image anaglyphy, chromastereoscopy, and side-by-side display. Anaglyphy uses double images in red and cyan to express binocular parallax depth. Chromastereoscopy is based on differences in diffraction blazing-induced dispersion through specialized prismatic lenses symmetrically integrated into eyewear. Side-by-Side juxtaposition employs specialized eyewear to present paired images corresponding to the left and right eyes. Blender is a 3D CAD (computer-aided design) tool equipped with shading, rendering, and compositing functions. Shading is used for material and texture configurations, rendering involves compiling realistic scene imagery, and compositing is used to refine rendered imagery in post-processing. It is important to note that compositing cannot affect the scene or change the rendered image. Blender is free downloadable 3DCG software. It is popular, and there are many of samples on Youtube and websites, and it is easy to use

[*] Corresponding author: xilehence@gmail.com

even for amateurs. In Blender, scenes are created by arranging objects. If this process can be automated with the assistance of ChatGPT, it can become easier to use scenes. In our system, blender interfaces with ChatGPT through OpenAI API. First, the prompt issues instructions to ChatGPT. Second, ChatGPT generates Python code, directly returned to Blender. Finally, this code executes in Blender, which features an interpretive Python interface for scripting. The created scene can be imported into Unity and similar platforms as a file for further use. Producing a stereograph with Blender requires certain settings, but with ChatGPT's automation, many users will be able to create and enjoy stereoscopic images freely. ChatGPT is an AI service provided by OpenAI. There are two models: GPT-3 (free to use) and GPT-4 (subscription-based, with higher performance). These come with an API key enabling integration into applications such as Blender. ChatGPT can be introduced into applications using the API provided by OpenAI. This allows us to automate the behavior of our application. Through the following research questions, we compare the extent to which each stereograph can be automated and the extent to which color can be reflected. By identifying aspects the areas that require human intervention, the stereograph rendering process can be omitted, and manual corrections can be immediately understood.
Q1: Can ChatGPT automatically configure stereo display?
Q2: How accurately is color displayed?
Q3: Is animation supported automatically?
Q4: Regarding interactivity, can the image be panned, tilted, zoomed or rotated?

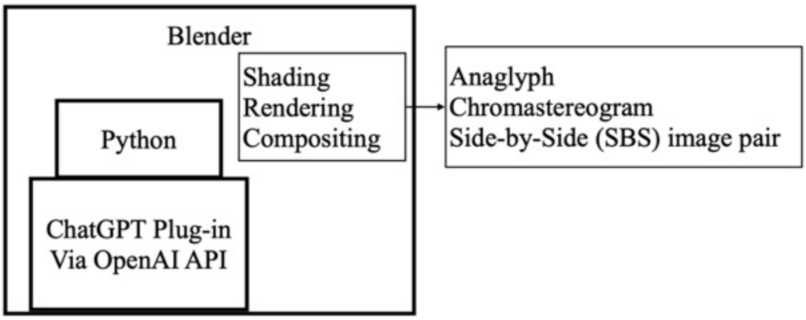

Fig. 1. System Schematic

2 Animation & Render Setting: Shading, Rendering, Compositing

In this section, we outline the process of establishing SkyBox, camera, lighting and scene objects using prompts as a preliminary step. If the prompt threw an error, we must manually manipulate the numbers and object ourself, or copy and past Python code automatically generated by ChatGPT and run it in Blender's Python console. In this research, we used the scene shown in Figure 2. The scene has one cube with Voronoi texture, three white cubes, and skybox background.

2.1 Setting SkyBox, Camera, Object

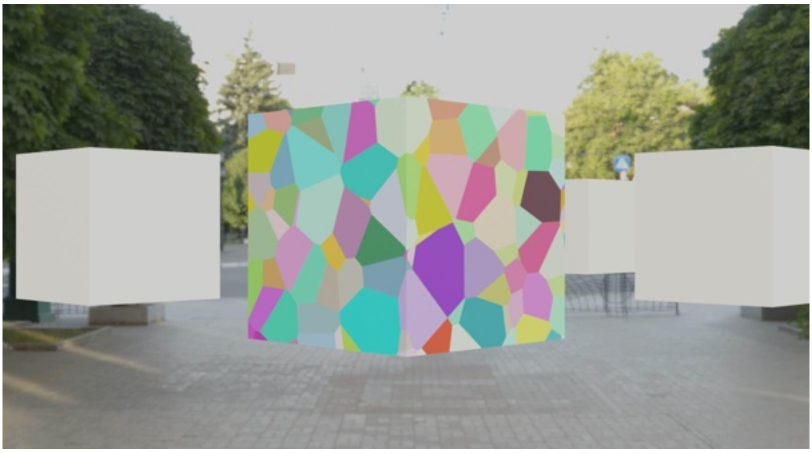

Fig. 2. Scene used in this comparison

2.1.1 Setting Skybox

Via manual (human) intervention, we specified the skybox, the image background (at virtual infinity range): World Properties/Surface/Open → Choose HDRI Image (originally downloaded from Internet) from local file system folders.

2.1.2 Setting Camera

The camera pose was set semi-automatically:
Prompt: `"Camera", Move to (0, 0, 0) & Rotation (0, 0, 0)`
Prompt: `"Camera", Rotation X -> 90, Z -> 90`
Combining the above two prompts into a single command:
Prompt: `"Camera", Move to (0, 0, 0) & Rotation (90, 0, 90)`

2.1.3 Setting Object

1. Add and Setting Object
Prompt Example 1:

```
Create a Cube, Object named "Cube", move to (0, 5, 0)
```

Prompt Example 2:

```
Create 2 Cube and move to (5, 0, 0) and (8, 3, 0)
```

Prompt Example 3:

```
Object named "Cube", rotate 45 z-axis
```

Setting other shaped objects (Cylinder, Sphere, etc...) is possible as well if desired.

2. Add Material to the Object

Prompt Example 1:

```
  Object named “Cube”, Add Material, Base Color(R=1, G=1, B=1,
Alpha=1), Metallic 1, Specular 1, Roughness 0, Use Nodes
```

Prompt Example 2:

```
  Object named “Cube”, Shading, Base Color(Hue=1,
Saturation=0.5, Value=1, Fac=1), Metallic=1, Specular=1,
Roughness=0, Use Nodes
```

If Blender cannot implement addition of Material using a prompt such as Example 1, the user can resort to an input like Example 2, where equal signs are interpolated to parameter settings, potentially improving success rates. If we attach “Use Nodes” in the above prompt and move to Shading window, we can add some nodes using prompts. Alpha, a numerical value raging from 0 to 1, sets transparency. Fac (Factor) controls the amount of influence the node has on the output image. Hue specifies hue rotation of the image. When saturation is 0, hue is suppressed, and the image becomes grayscale. A shift greater than 1.0 increases saturation. Value adjusts the overall brightness of the image. By adding “Use Nodes”, added material will use the same coefficients as the Principled BSDF node in the Shading window.

Prompt Example 1: `Object named “Cube”, Add Voronoi texture node`

Prompt Example 2: `Object named “Cube”, Add Checker texture node`

.

2.2 Programed Animation

This subsection provides two examples of the prompts and what animations that can be implemented.

Prompt:

Example 1:

```
  Set Animation “Cube (Object name in Scene Collection)”, move
(-2, -1, 0) to (7, -1, 0)  & return, 200 frames
```

Example 2:

```
  Set Animation, “Camera”, Rotation 180 on Z-axis, 80 frames
```

Example 3:

```
  Add Animation, “Camera”, First Rotation 180 on Z-axis 160
frames, Next Rotation 90 on X-axis 80 frames
```

Example 1 added animation to an object named “Cube”. The cube moves from (x, y, z)=(-2, -1, 0) to (7, -1, 0) in 200 frames.

Example 2 added animation to a camera object. The camera rotated 180° on z-axis in 80 frames.

Example 3 added animation to a camera object. First, the camera rotated 180° on z-axis in 160 frames. Next, the camera rotated 90° on x-axis in 80 frames. (Total 240 frames animation).

2.3 Render Setting

In this section, we elaborated on setting anaglyphic, chromastereoscopic, and side-by-side scenes. Each subsequent prompt implicitly refers to a directory, setting the folder where the project was archived.

2.3.1 Anaglyphy

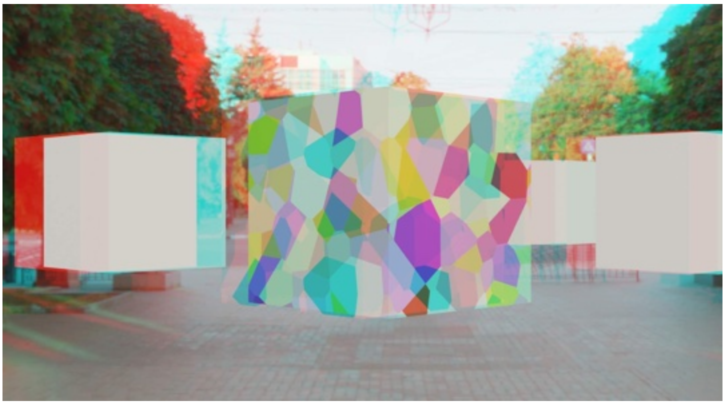

Fig. 3: Anaglyph for anaglyphic viewing

In this section, we describe semi-automatically configuring the render engine required for Anaglyphic and the settings in the output properties. Activate the item Stereoscopy and switch from Multi View to Stereo 3D. Stereo Mode become Anaglyph (Red-Cyan). Then we define the rendering aspect ratio, output file format, and save the destination file.

Prompt:

```
Scene, Render Engine, Eevee & Output properties, Stereoscopy become Active, Multi-View -> Stereo 3D & Format, Resolution X -> 1080 px, Resolution Y -> 1080 px & Output, /Users/personal_folder/, File Format -> PNG & Object named "Camera", Object Data Properties, Type is Perspective
```

Next, we perform the following operations manually. In the output properties, select Stereo 3D for Views format in the Views item. In the items added by this operation, set Stereo Mode to Anaglyph and Anaglyph Type to Red-Cyan.

2.3.2 Chromastereoscopy: asymmetric prismatic grating causing refraction and chromatic dispersion

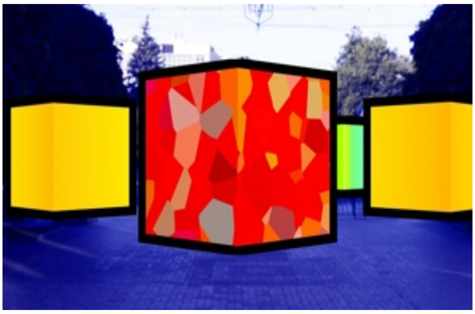

Fig. 4: Chromastereogram for Chromadepth viewing, including shape outline

For chromastereoscopy, we must establish data flow via connecting nodes and setting values of nodes in compositing graph editor window with human intervention. In this setting, Blender compositing graph adjusts rendered scene image. This rendered image includes objects and skybox. To generate chromastereoscopic imagery, output properties setting, stereoscopy is not active, so we use prompt: Scene, Output Properties, Stereoscopy become disabled/off.

1. Setting Render Engine and Output Properties

This section shows how we semi-automatically set the Output properties essential for rendering, specifying the render engine, image aspect ratio, save folder, and file format. This is the setting required to save the rendered image on a computer.

Prompt:

```
    Scene, Render Engine, Eevee & Scene, Output properties,
Format, Resolution X -> 1960px, Resolution Y -> 1080px &
Output, /Users/personal_folder/, File Format -> AVI JPEG
```

2. Setting Compositing Node for Rendering

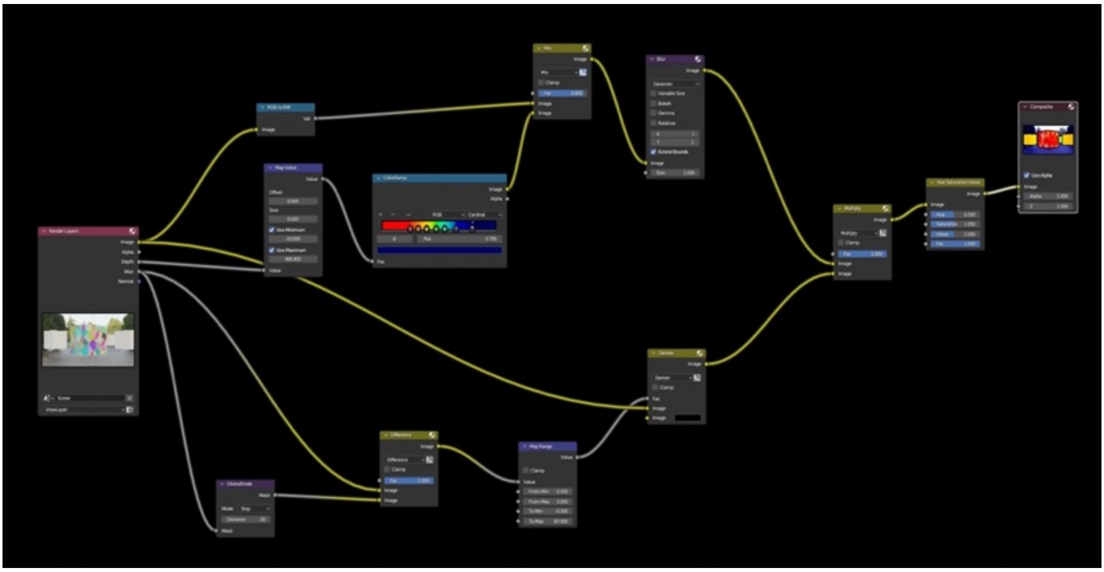

Fig. 5: Chromastereoscopic compositing graph in Blender

By setting the compositing node, we could composite various effects into the rendered image. Figure 6 is a graph showing how colors are applied to the rendering image by the ColorRamp node in compositing node graph. By using compositing function and setting as shown in Figure 5, we could render chromastereoscopy. Figure 4 is a chromastereogram rendered using the settings of Figures 5 and 6.

1) Setting node for Color Gradient

Prompt:

```
1.Compositing, Use Nodes
2.ViewLayerProperties, Passes, "Deliver Z value pass." become
  Active. Compositing, Add Map Value node, ColorRamp node,
  RGB to BW node, Mix node, Blur nodes
```

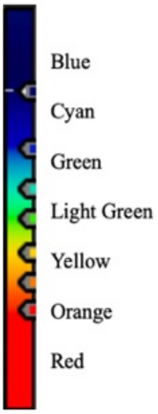

Fig. 4

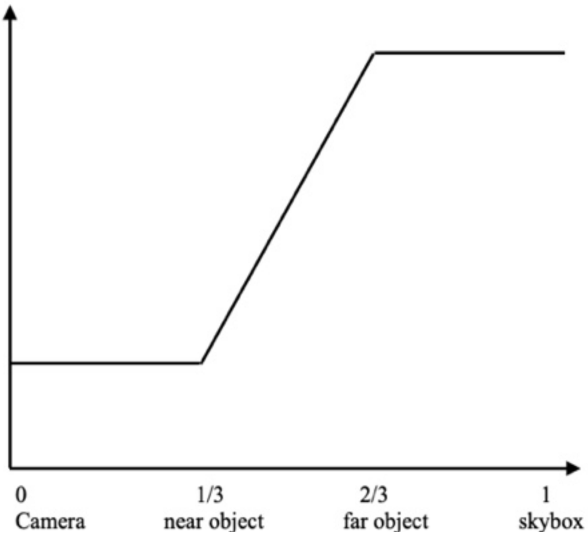

Fig. 6: Color mapping gradient of the ColorRamp node

2) Setting node for Object Outline

Prompt:

```
   In Compositing Window, Add Dilate/Erode node, Map Range node
and  Hue  Saturation  node  &  Add  3  Mix  nodes  that  Mix  ->
Difference, Mix -> Darken and Mix -> Multiply
```

3) Setting node values (human intervention)

The parameters below are the numbers set in the node used the graph shown in Figure 5.

Dilate/Erode node: Distance → -30

Map Range node: From Min → 0.1 & From Max→ 1 & To Min → -0.3 & To Max → 87

Hue Saturation Value node: Hue → 0.5 & Saturation → 1.05 & Value → 1 & Fac (Factor: Controls the amount of influence the node has on the output image) → 1

These values need to be adjusted so that objects in the Scene are properly colored when we create our own Chromadepth image.

2.3.3: Side-by-Side (SBS)

In this section, we describe semi-automatically configuring the render engine required for Side-by-Side and the settings in the output properties, activating Stereoscopy and switching from Multi View to Stereo3D. Stereo Mode become Side-by-Side. Then, we define the rendering aspect ratio, output file format, and save the destination file.

Prompt:

Scene, Render Engine, Eevee & Output properties, Stereoscopy become Active, Multi-View->Stereo 3D & Format, Resolution X -> 1080 px, Resolution Y -> 1080 px & Output, /Users/personal_folder/, File Format -> AVI JPEG & Object named "Camera", Object Data Properties, Type Panoramic

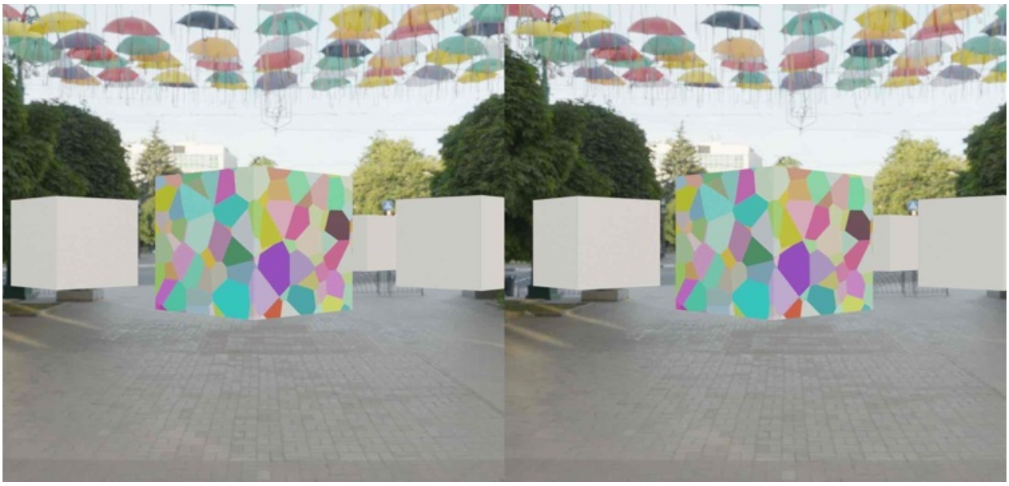

Fig. 7: Side-by-Side for Berezin Screen Scope, Loreo Stereo Viewer, 3D TV, or Google Cardboard. Note binocular parallax.

3 Preparation

This section mainly describes how to setup the Blender ChatGPT plugin. One needs to prepare in advance Blender, the plugin, and a ChatGPT API key. The prompts for the camera rotation animation we used in our experiment are provided.

3.1 Shared Scene

Across Anaglyph, Chromadepth and Side-by-Side, each setup shares the same objects, materials, and lighting. However, Chromadepth cannot use same colors because it forces color to encode depth.

3.2 Animation

This subsection describes the instruction prompts for implementing the animations used to compare each stereograph. Blender animation can include camera motion, object displacement, and scaling. Our animation features 180° camera rotation around z-axis followed by a 90° rotation around x-axis.

Prompt example:

```
  Add Animation, "Camera", First Rotation 180 on Z-axis 160
frames, Next Rotation 90 on X-axis 80 frames
```

3.3 ChatGPT Interface

Blender connects with ChatGPT. By inputting prompts into the Blender ChatGPT interface, ChatGPT generates Python code to be executed in Blender.

4 Comparison

In this section, we examine anaglyphy, chromastereoscopy, and side-by-side display (Berezin ScreenScope, 3D TV (Sharp), Google Cardboard, Loreo stereo viewer) with following four questions. This allowed us to verify what each stereoscopy looked like and whether it was capable of automatic scene generation, and to clarify details.Through this validation, it is easier to determine which type of stereoscopy to use.

1 Anaglyphic (red & cyan)

Q1. Yes. But, if it was not working prompt exactly, we should do operation human intervention.

Q2. Not great. In anaglyphy, red and cyan color filter glasses are used to split the image into the left and right eyes according to the bend of light. Since this image method uses two complementary colors, the image color is compromised.

Q3. Yes

Q4. Yes

2 Chromadepth

Q1. Human tuning gradation & connect nodes.

Q2. Compromised, since color coding overrides natural color.

Q3. Yes

Q4. Yes

3 Side-by-Side (SBS)

3a) Berezin Screen Scope or Loreo stereo viewer

Q1. Yes

Q2. Great
Q3. Yes
Q4. Bad

3b) 3D tv (Sharp)
Q1. Yes
Q2. Great
Q3. Yes
Q4. Bad

3c) Google Cardboard

We can create Google Cardboard scene app for using Unity. We should download Google cardboard-xr-plugin and CameraController.cs for create this scene. We import Scene file from blender, and it was imported into Unity.

Q1. Yes. But, not working prompt exactly, we should do operation human intervention.

It set the aspect ratio to 1:1.

Q2. Great
Q3. Yes
Q4. Yes

Table 1: Summary Comparison

	Anaglyph	**Chroma-stereoscopic**	**Side-by-Side (SBS)**	**3D TV**	**Google Cardboard**
Automatic Configuration	○	△	○	○	○
Color display	△	×	+	+	+
Animation	○	○	○	○	○
Interactivity	○	○	×	×	○

×: bad. △: not good. ○: yes. +: Great.

5 Conclusion

Most parts of each stereoscopic image could be rendered completely automatically on Blender with ChatGPT, but some aspects of rendering require human intervention because error could occur when running prompts. If ChatGPT version is GPT-4 instead of GPT-3.5, it might work more accurately and resolve errors. If prompt appear errors, we need to set scene setting with human intervention or refresh Blender until prompt work exactly. Also, in Chromadepth rendering, we can add nodes to compositing graph with prompts, but we should connect compositing nodes and set Chromadepth colors for ColorRamp nodes with human intervention. In each rendering images comparison, we can implement animation and interactivity for all stereoscopies with prompts. Regarding the two types of stereoscopic images besides the side-by-side method, color displays comply with the respective image techniques, so it is impossible to directly display the colors of textures etc. assigned to objects in the scene. When setting objects using prompts, it was not possible to create objects with complex shapes, such as human figures or animals. Also, in the animation settings, simple physical movements such as movement, rotation, and bouncing were possible, but it was not possible to make complex animation like people walking. By using this research to automatically create scenes and render stereograms, anyone can easily enjoy stereoscopy. In the future, if the version of ChatGPT updates beyond 4.0, it may be possible to send more advanced commands to Blender through prompt.

1. OpenAI API Key. https://platform.openai.com/account/api-keys.
2. Add-on Blender x ChatGPT: https://github.com/gd3kr/BlenderGPT
3. Blender Chroma Depth - Node Setup: https://blenderartists.org/t/chroma-depth-node-setup/375465
4. Erito Kaio, Blender Compositor Outlines: https://www.youtube.com/watch?v=NB1S1zhce4E
5. Takashi Kawai, Hiroshi Morikawa, Keiji Ota, Nobuaki Abe, "Basics of 3D stereoscopic image expression – from basic principles to production technology" 共著=河合 隆史, 盛川 浩志, 太田 啓路, 阿部 信明, 「3D立体映像表現の基礎-基本原理から制作技術まで-」 (2010) ,ISBN 978-4-274-06816-4
6. Blender Academy, Blender Tutorial – How to Use AI to Create 3D Models (ChatGPT and Blender). https://www.youtube.com/watch?v=x60zHw_z4NM
7. Google Cardboard XR Plugin for Unity: https://github.com/googlevr/cardboard-xr-plugin
8. CameraController.cs Sample Code: https://ideone.com/buALlz