Demo video: https://youtu.be/epZgnGzbEiI (on a MacBook Pro which had integrated GPU)

Drawing:

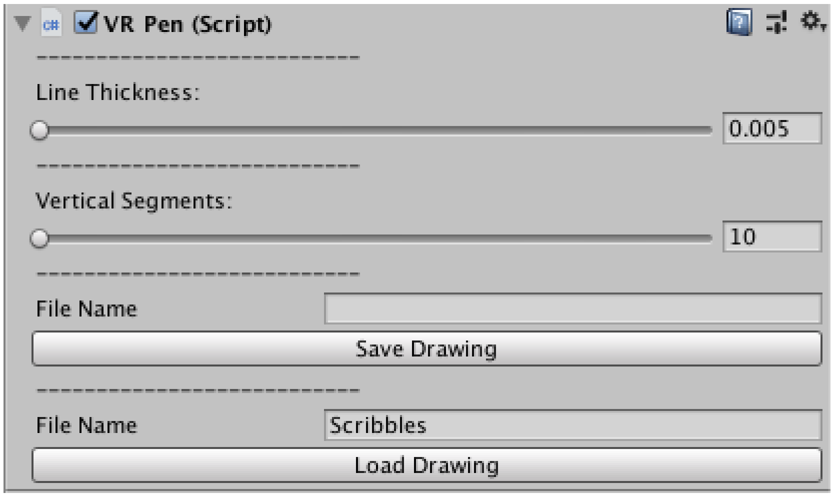
Press play in the Unity editor (keyboard shortcut: Ctrl + P).
(As of now, pre-recorded motion is being used to draw the lines. VR support will be added soon.)

Deleting all the lines:

Press ‘C’ in the keyboard to delete all the lines in the scene.

Adjusting line thickness, saving drawings in the scene, Loading previously saved drawings:

Click on the *VRPen* gameobject in the hierarchy to access these functionalities. They were implemented using Editor scripting.



Line Thickness: 0.005

Vertical Segments: 10

File Name

File Name Scribbles

For future optimization:

- Implement dynamic arrays to avoid calls to Array.resize() every frame when a line is drawn.
- Remove duplicate code for upper and lower hemispheres. Instead use a single hemisphere and transform/index it appropriately, as per the need.

Further improvements:

- More vertices can be added at sharp ends when extreme change of direction is detected (for example, 45 degree turn or above), so that the line doesn’t get squished at these bends.

Scripts:

VRPen.cs, VRPenEditor.cs
LineMesh.cs
Repository.cs
Player.cs

Note:
Hemisphere construction and vertex indexing has been implemented based on the code from http://wiki.unity3d.com/index.php/ProceduralPrimitives by Bérenger.

The project was built using Unity 2018.1.2f1.