3D Cell Visualization Utilizing Unity Game Engine

Dr. Joiner[1], Lark Bancairen[2],Cymantha Blackmon[3], Michael Casarona[4], Mary Kate Morales[5], and Annaliese Watson[6]

NJCSTM, Kean University, 1075 Morris Ave, Union, NJ 07083[1,2,3,4,5,6]

ABSTRACT

- We are developing a tool for the visualization of data from a confocal microscope, converting raw images into a 3D visualization.
- We have tested two different networking interfaces within Unity, Forge Remastered and Vivox.
- We have begun the process of building a network aware prototype visualization tool.

INTRODUCTION

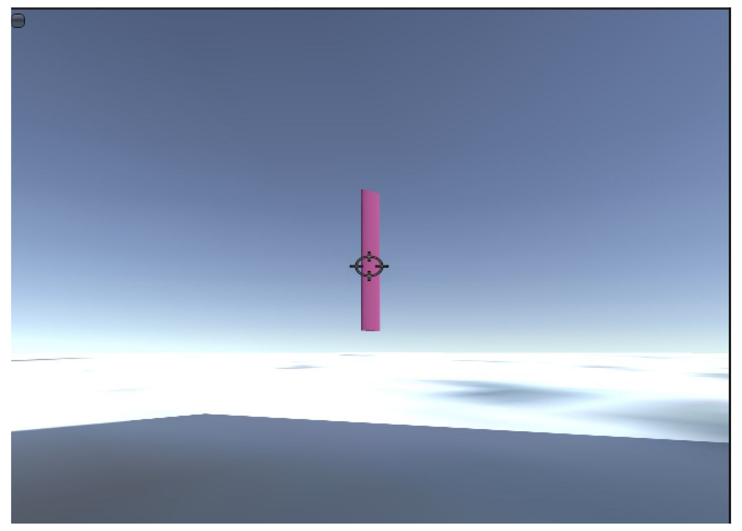
- Microscope raw data in form of layered images with pixels in the x, y direction at various z values.
- Our focus has been on design elements for two issues
  - How will the user interface be different in a VR headset compared to a PC (selection, resize, moving, contrast adjustment, and rotation of data)?
  - How can we add network capability to allow for remote viewing/collaboration? (text/audio chat, shared movement/control)
- Native networking model in Unity has changed multiple times over the years,and third-party networking methods have become more popular in an attempt to keep code consistency as Unity evolves.

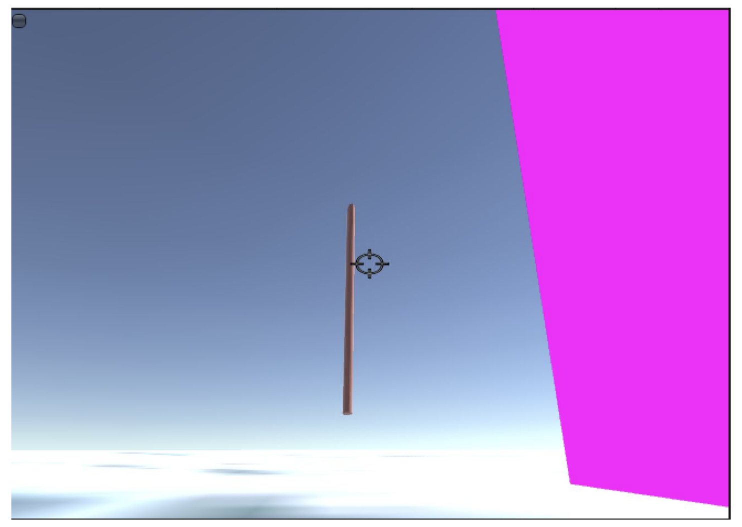
MATERIALS and METHODS

- Software Used:
  - Unity Game Engine, Forge Remastered Networking System, Vivox
- Forge Remastered:
  - Provides for shared network variables that can be synchronized across a server and multiple clients
  - RPC calls allow clients to affect change on server, that is then seen by other clients
  - Item selection and movement can be shared across network connected instances of the tool.
- Vivox:
  - Uses a commercial server (free for our use case) for audio and chat that easily incorporates into Unity

RESULTS

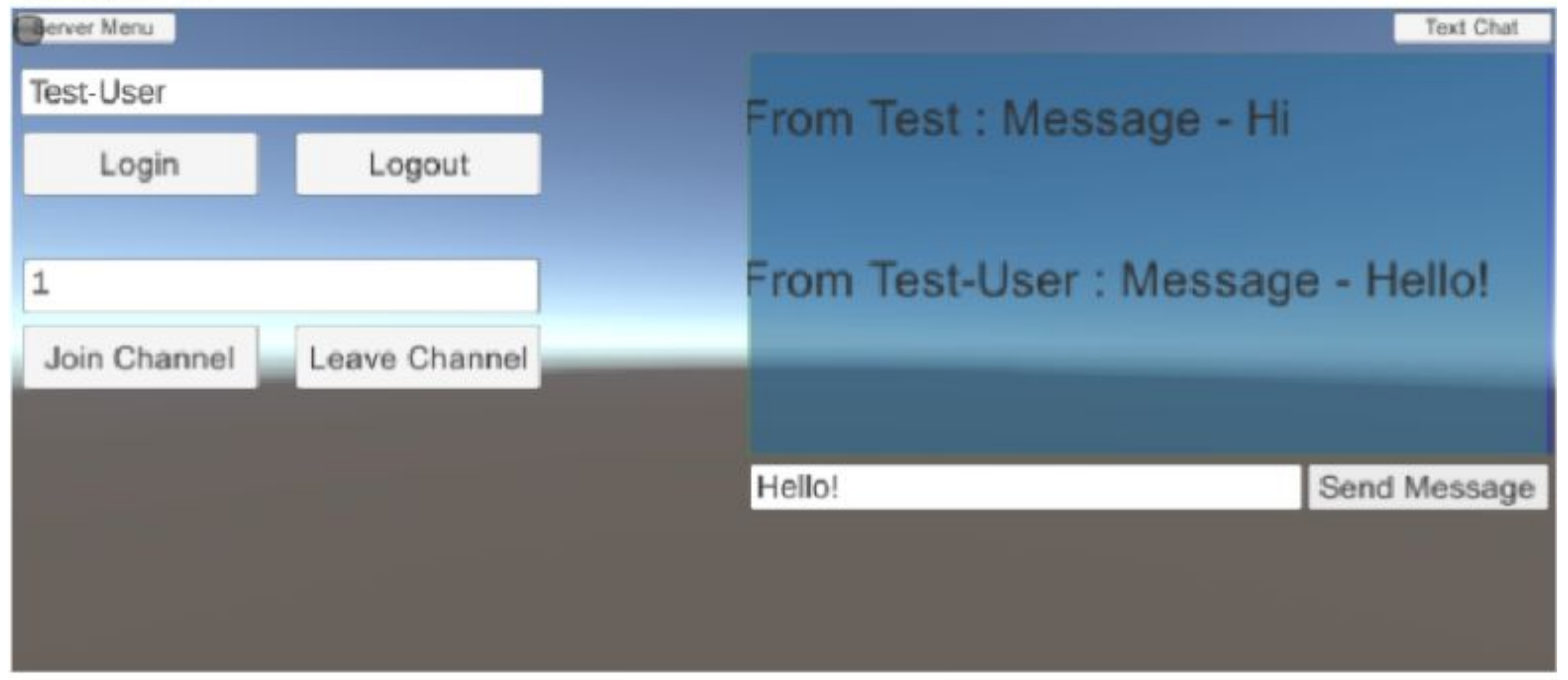
Hovering vs. Selecting using target point objects in 3D



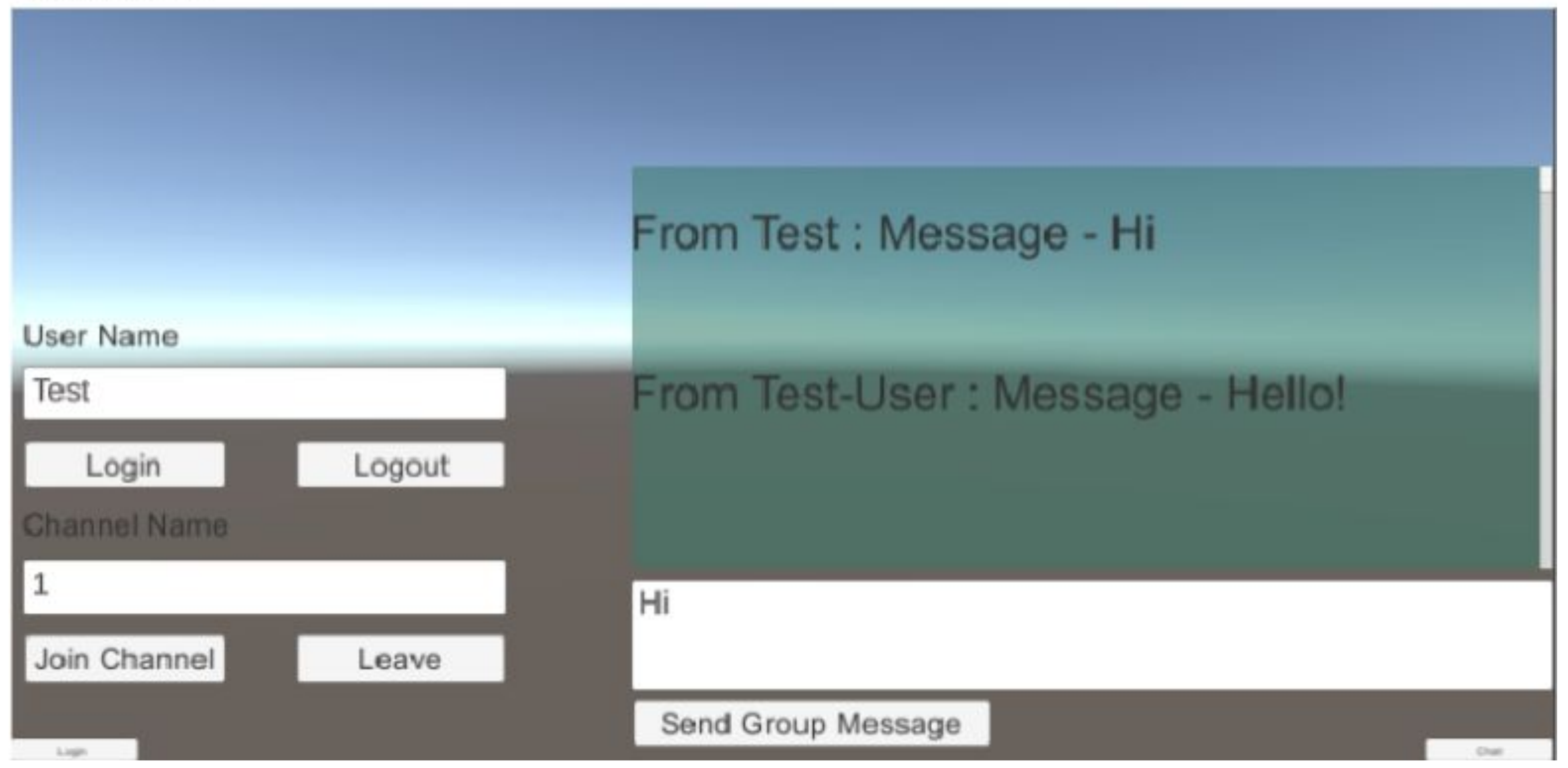


Communication through chat multi user

Machine 1:



Machine 2:



DISCUSSION and CONCLUSION

- Network testing
  - Forge Remastered and Vivox offer different aspects of the network model (control versus voice/chat) and as such can be integrated into the same application. Both were found to have good documentation, have a simple implementation, and work stably in our tests.
- UI control
  - Prototypes of a user interface on client machines for 2.5D selection by click and 3D selection by gaze, movement of an object was completed separately, but still needs to be integrated into a single code base, tested on VR hardware.
- Next stages of prototype should incorporate both networked UI control and communication in a single app, and begin testing with visualized data.

References

Farris, B. [Brent]. (2017, March 22). *Unity Forge Networking Jumpstart 03 - First Example Moving Cube* [Video]. YouTube. https://www.youtube.com/watch?v=feLUmzaogCk

Farris, B. [Brent]. (2017, March 22). *Unity Forge Networking Jumpstart 04 - Adding to Network Blueprint* [Video]. YouTube. https://www.youtube.com/watch?v=G7sq3MoOJuU

Farris, B. [Brent]. (2017, March 22). *Unity Forge Networking Jumpstart 05 - Remote Procedure Call* [Video]. YouTube. https://www.youtube.com/watch?v=Sb8Ex0QTfNI

Farris, B. [Brent]. (2017, March 22).*Unity Forge Networking Jumpstart 06 - Buffered Remote Procedure Call* [Video]. YouTube. https://www.youtube.com/watch?v=Ey3f-nxQpvg

Farris, B. [Brent]. (2017, March 22).*Unity Forge Networking Jumpstart 07 - Remote Procedure Call Arguments* [Video]. YouTube. https://www.youtube.com/watch?v=8O7fU-GGN6Y

Farris, B. [Brent]. (2017, Aug 13). *How to Make Online Chat in Unity for Multiplayer Games* [Video]. YouTube. https://www.youtube.com/watch?v=JGgnPRa2Ets

Murphy, J. [Full Stack Indie]. (2020, December 17). *VivoxVoiceChat #5 - Send Event/Hidden Messages [C# Unity3D].* YouTube. https://www.youtube.com/watch?v=JBFFuuMBvPU

Acknowledgments

Thank you to Dr. Joiner for the continuous support and feedback regarding the unknown aspects of the network process.

Thank you to Dr. Niepelko for the contribution to our project with the data from the confocal microscope that will be used for visualization.

Shader Development & User Interface Design for 3D Scientific Visualization

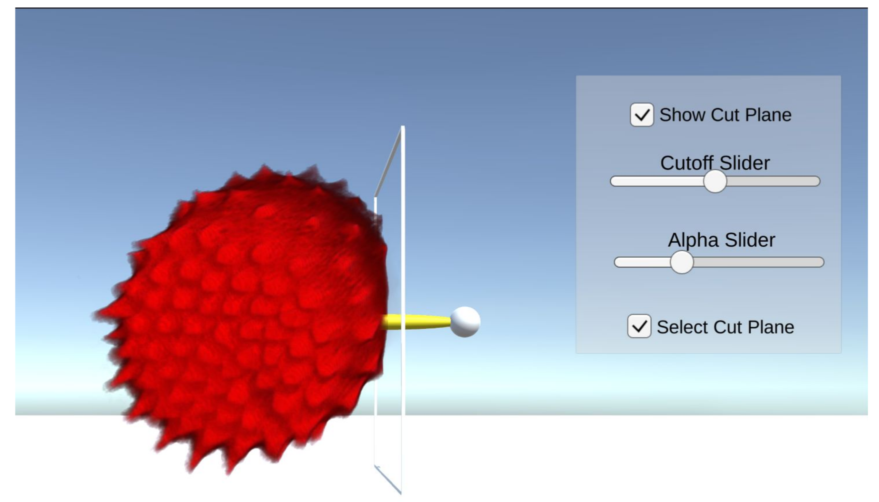
Dr. Joiner[1], Cymantha Blackmon[2], and Annaliese Watson[3]

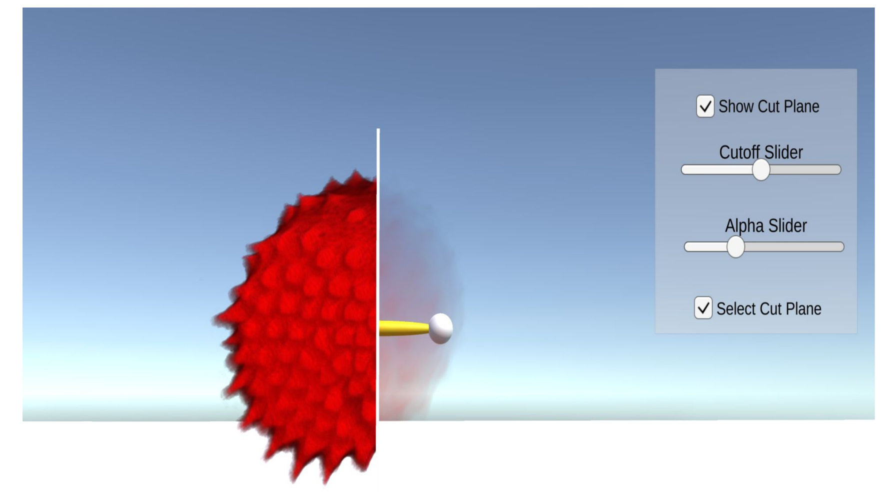
School of Integrated Science and Technology, Kean University, 1075 Morris Ave, Union, NJ 07083[1,2,3]

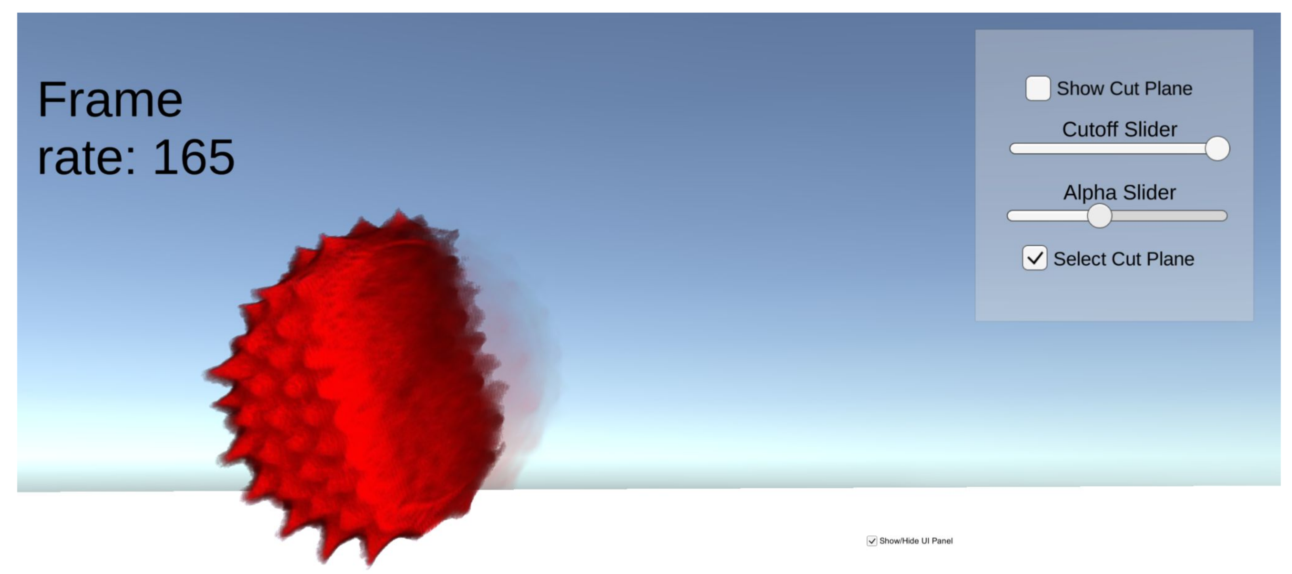
INTRODUCTION

This research focuses on a user interface for viewing 3D data collection of hyper stack images, including pollen grains and fruit fly stem cells, taken with a confocal microscope. The visualization platform is designed to render hyper stack images as a volume by displaying the images with transparency in a Unity scene. The user interface (UI) and visualization are optimized for manipulation of a cut plane adjustable in position and rotation that can show the various layers of the data while maintaining its integrity, movement (slide/rotate/drag), transparent viewing, and other color adjustments. The shaders used are modified to add transparency on one side of the cut plane, defined in terms of a normal vector and base point. Use cases for the visualization tool included validation of object detection routines applied to the automated counting of cells in data, and the creation of a database of 3D pollen grain images.

RESULTS







CONCLUSIONS

The introduction and inclusion of calculated logic in the shader allows the computation of the cut plane while checking for the image's edges. The adjustment allowed for a frame rate reduction of 3% on a laptop with a NVIDIA GeForce 3080 graphics card. Similar results have been seen on lower end laptop hardware.

Future experimentation would include increasing the usability of the user interface. We want to continue testing performance of this shader on VR systems and mobile devices.

MATERIALS and METHODS

The tool was designed using Unity Game Engine, with vertex fragment shaders customized using the CG shader language,

The cut plane was defined in terms of a base position and a normal vector, and the transparent side of the plane was defined such that any pixel relative to the base and normal would satisfy

$$(\vec{p} - \vec{b}) \cdot \vec{n} > 0$$

p: a vector position of a pixel in world space
b: a vector base of the cut plane relative to the object
n: the normal to the cut plane

```
v2f vert (appdata v)
{
    v2f o;
    o.vertex = UnityObjectToClipPos(v.vertex);
    o.worldPos = mul(unity_ObjectToWorld, v.vertex);
    o.uv = TRANSFORM_TEX(v.uv, _MainTex);
    o.normal = normalize(mul(float4(v.normal, 0.0),
                   unity_WorldToObject).xyz);
    return o;
}

fixed4 frag (v2f i) : SV_Target
{
    half camera_normal = abs(dot(normalize
   (_WorldSpaceCameraPos.xyz - i.worldPos.xyz), i.normal));
    if (camera_normal < _NormalCutoff)
        return fixed4(0, 0, 0, 0);
    fixed4 col = tex2D(_MainTex, i.uv);
    half value_squared = dot(col.rgb, col.rgb);
    float dotprod = dot((_Base - i.worldPos), (_Normal));
    half modifier = step(_Cutoff*_Cutoff*0.03,
        value_squared);
    col.a *= ((1-_CutPlaneAlpha) * step(0.0,
        dotprod)+_CutPlaneAlpha)* _Alpha * col;
    col *= modifier;
    return col;
}
```

References

We thank the Center for Biological Imaging at Kean University for assisting with image acquisition and the Niepielko Lab for their helpful comments and fruitful discussions. Research reported in this poster was supported by the Eunice Kennedy Shriver National Institute of Child Health & Human Development of the National Institutes of Health under award No. R15HD10296

Acknowledgments

Thank you to Dr. Joiner for the continuous support and feedback regarding the unknown aspects of the current and future development processes.

Shader Development and User Interface Design for 3D Scientific Visualization in VR Headset

Annaliese Watson and David Joiner

School of Integrated Science and Technology, Kean University, 1075 Morris Ave, Union, NJ 07083

ABSTRACT

The introduction of a VR interface for visualizing confocal microscopy 3D images facilitates interactive data exploration with enhanced transparency and manipulation features. Validated on Oculus VR headsets, this interface promises to improve 3D structured data analysis.

INTRODUCTION

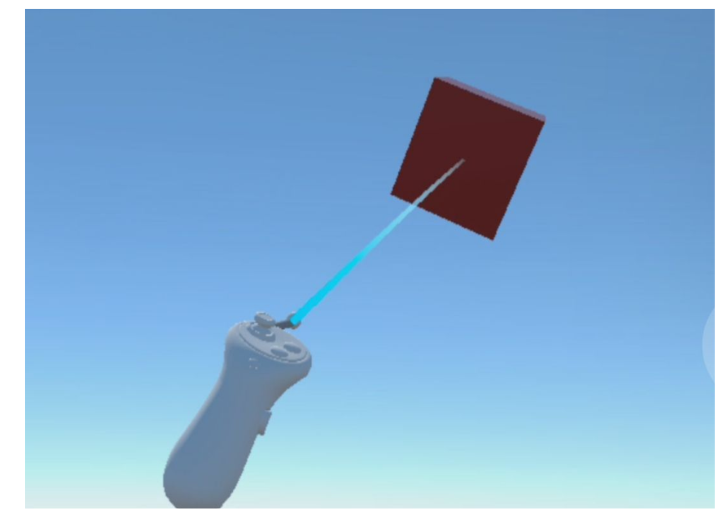
This research introduces an innovative user interface (UI) specifically tailored for visualizing three-dimensional (3D) hyperstack images of pollen grains and fruit fly stem cell specimens, acquired through confocal microscopy. Leveraging virtual reality (VR) technology, the UI provides volumetric renditions of hyperstack images with enhanced transparency and manipulation across x, y, and z cut planes, maintaining data integrity within the VR environment. An interactive menu allows further customization via sliders, checkboxes, and buttons, enhancing the user experience. We are using this tool to validate object detection algorithms for object counting, as well as to develop a 3D pollen grain image database. The interface's efficacy is confirmed through testing on Oculus Quest 2, 3, and Pro headsets.

MATERIALS and METHODS

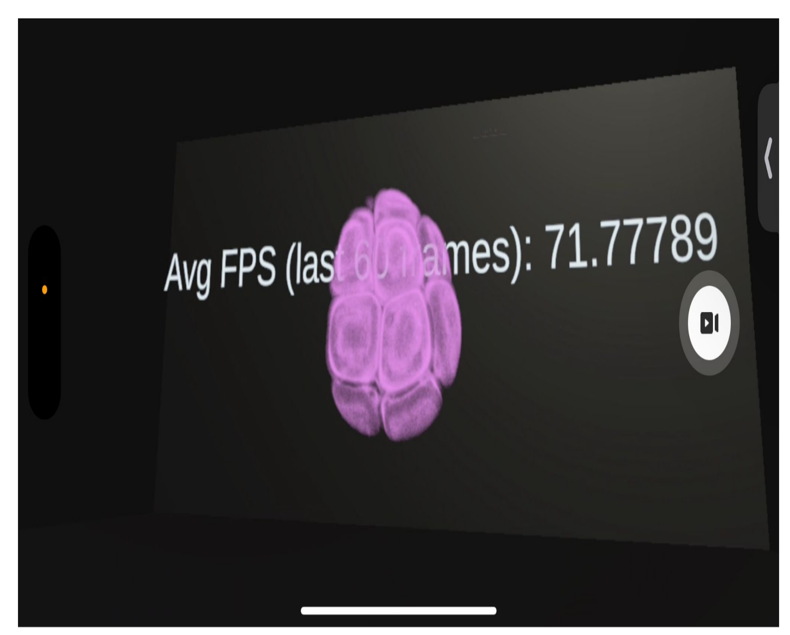
The Unity-built user interface utilized customized shaders and precisely defined clip plane parameters to ensure functionality. Integration with Oculus Quest 2, 3, & Pro headsets was achieved through the XR interaction toolkit, enabling connection between the Unity game scene and the Oculus headset's 3D world space. The interface operates optimally at 50-72 frames per second, featuring shader optimization and controller-based clip plane manipulation in VR.

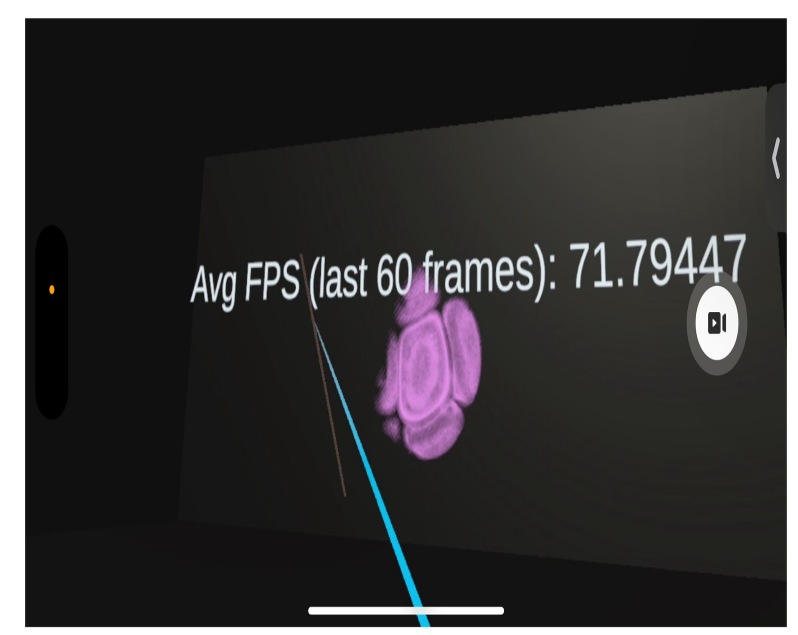
RESULTS

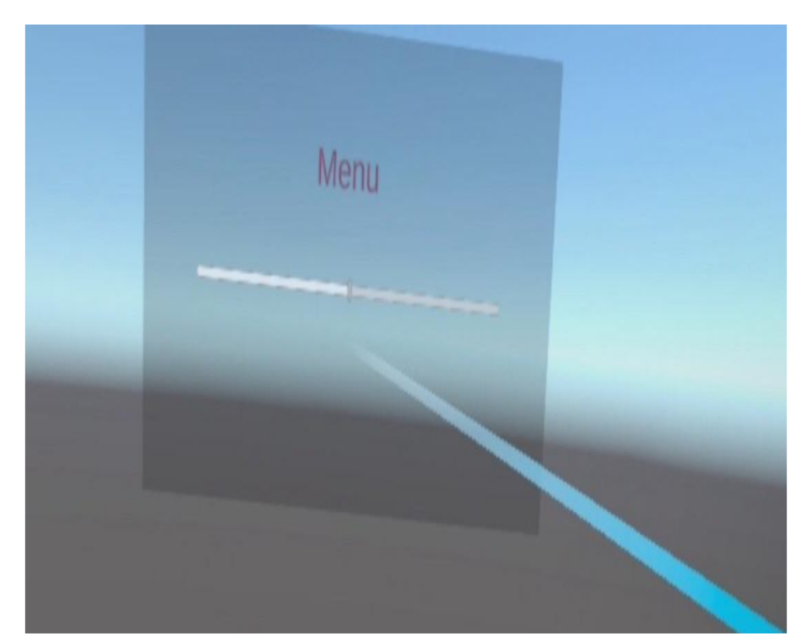
A clipping plane object was implemented using the grabbable component within XR Interaction Tooklit, allowing for easier placement of the clipping plane within VR.



Images were loaded into the VR headset by converting each to a RGBA sprite, placed in the scene spaced in the y direction. Background colors were removed and displayed with transparency, allowing the creation of a volumetric effect simply by image placement. Frames per second were calculated and included in the scene for measurement of performance efficiency. Optimized shaders facilitated clip plane selection for interactive cellular examination, enabling the rendering of specific parts as transparent. Additionally, a Menu Bar, accessible via controller buttons, featured sliders to customize future viewing experiences.







DISCUSSION and CONCLUSION

Continued development of visualization tools in VR requires a rethinking of User Interface, to allow for an understanding of how spatial computing will impact science visualization. Its 3D view user interface (UI), designed for visualizing hyper stack images obtained from confocal microscopy, enables transparent volumetric renditions and manipulation via controllers and an interactive menu. Leveraging Unity's built-in XR Interactable toolkit and compiled on Oculus hardware, it has been successfully tested with Oculus Quest 2, 3, and Pro headsets.

The interface can be further enhanced by integrating exclusively with the Meta world and incorporating features such as directly embedding the menu into the VR environment, as well as adding options for color enhancement and shape-based viewing.

Future work includes development of additional shader based image enhancement and image contrast tools within VR.

References

We thank Kean University, The Center for Aerobiological Research, and Kean's Center for Biological Imaging for providing pollen slides and imaging facilities. Portions of this work were supported by NSF IUSE-HSI 2247157 and the primordial germ cell project was supported by NIH R15HD102960.

Valem Tutorials. (2022, September 4). *How to make a VR game in Unity 2022 - part 7 - user interface*. YouTube. https://www.youtube.com/watch?v=yhB921bDLYA&list=PLpEoiloH-4eP-OKItF8XNJ8y8e1asOJud&index=8

Acknowledgments

Thank you to Dr. Joiner for the continuous support and feedback regarding the unknown aspects of the current and future process.

Thank you to Dr. Niepelko for the contribution to our project with the data from the confocal microscope that will be used for visualization.