Getting Started with Virtual Reality

Benjamin Knorlein
Center for Computation and Visualization, Brown University

DSCoV: Data Science, Computing, and Visualization workshops
Thursday, June 6, 2019

Overview

- Introduction to VR
- VR Devices
- Applications
  - Paraview
  - A-Frame
  - Unity 3D
- Demos

The Ultimate Display

The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal.

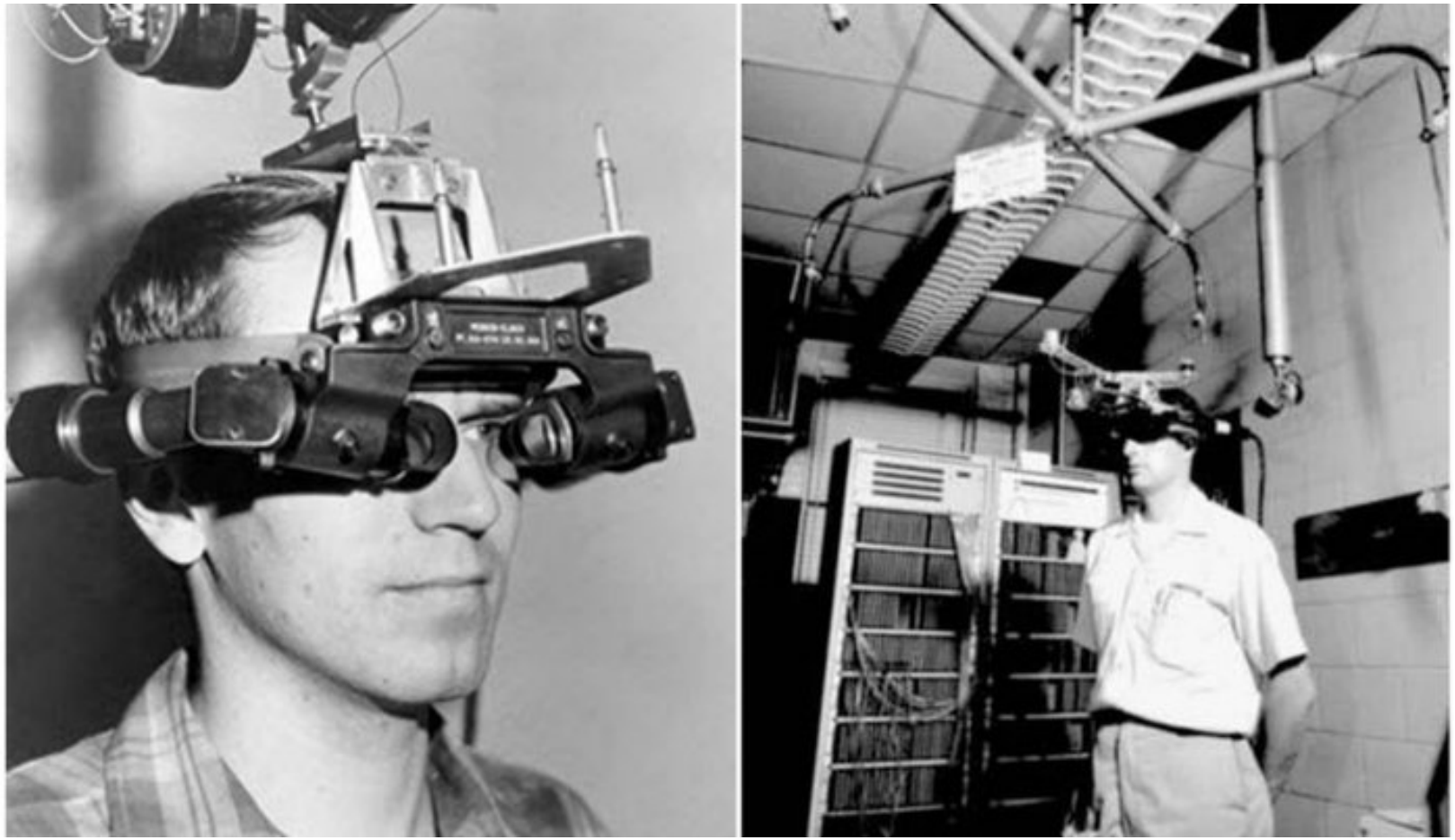
With appropriate programming such a display could literally be the Wonderland into which Alice walked.

Ivan Sutherland, 1965

The Ultimate Display – the Holodeck

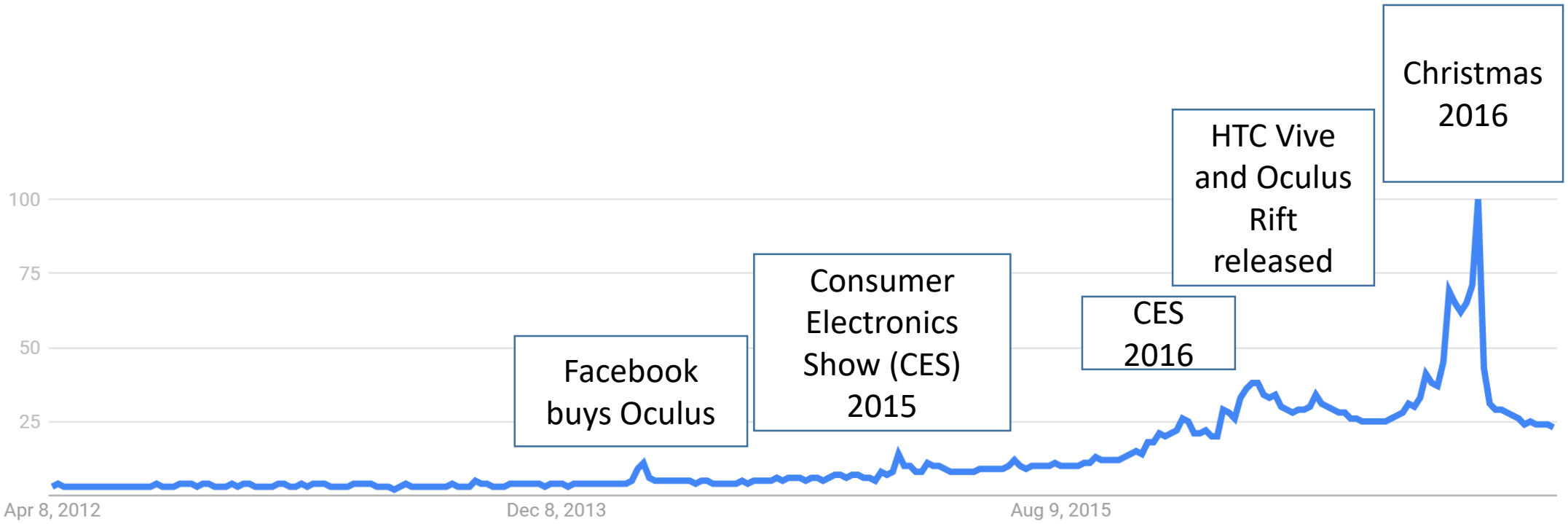
First appearance in "*Star Trek: The Animated Series - The Practical Joker*“ (Recreation room), 1974

The Ultimate Display



The sword of Damocles, *Ivan Sutherland and Bob Sproull*, 1968

Google Trend



Google Trends – “Virtual Reality”

Virtual Reality

An artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment;

Merriam-Webster

Virtual Reality

An artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment;

Merriam-Webster

Sensors

Computer

Display

Mobile phones and 3DOF devices

Cheap and light Displays

Built-in sensors : Compass, Accelerometer, Gyroscope

Advantage: Many users

Limitation: Mostly sensing (tracking) of rotation only

Head Mounted Displays (6DOF)

< 1000USD

Tracking of position and rotation

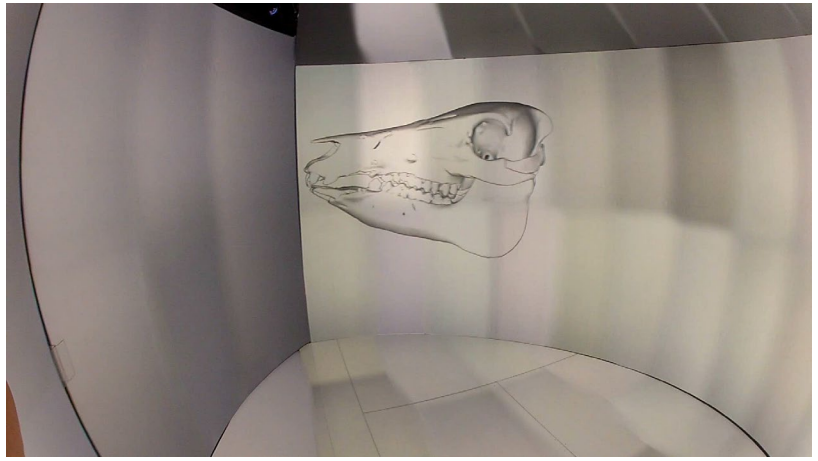
Controllers for interaction

Most devices require a PC

CAVE – Cave Automatic Virtual Environment

Carolina Cruz-Neira, Daniel J. Sandin and Thomas A. DeFanti. "Surround-Screen Projection-based Virtual Reality: The Design and Implementation of the CAVE", *SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*,

YURT – YURT Ultimate Reality Theater

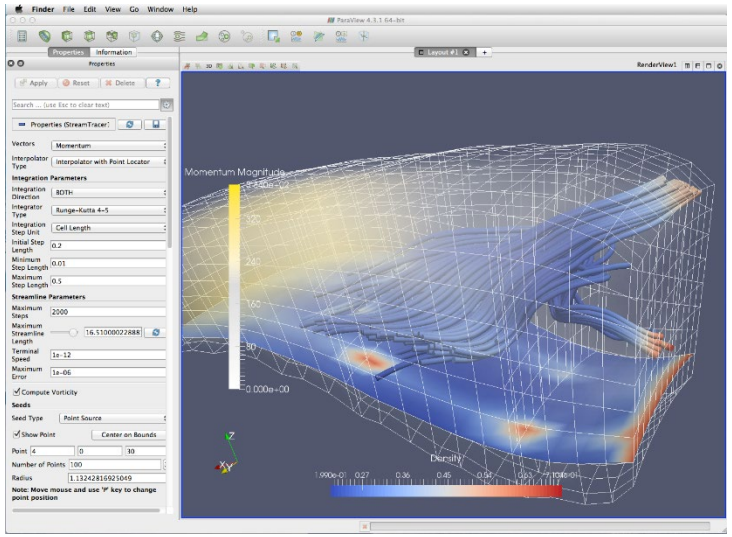




Sign up for a tour at : https://ccv.brown.edu/services/infrastructure/yurt/

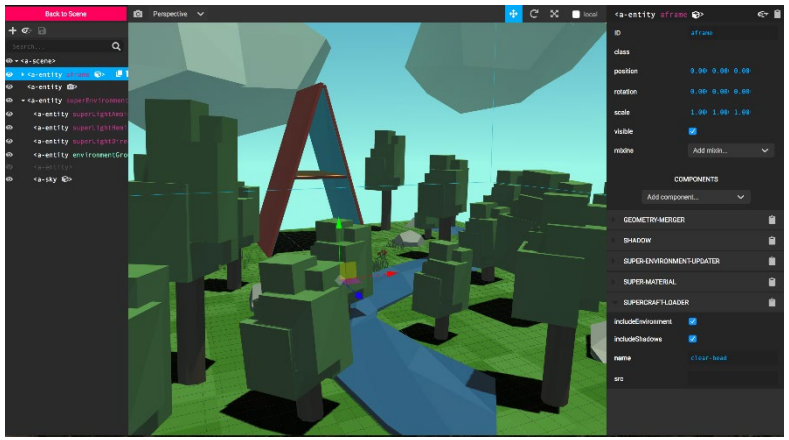
20 nodes
69 projectors
69 graphic cards
138 HDMI cables
143 million pixel


NSF #0923393 Development of a Next-Generation Interactive Virtual-Reality Display Environment for Science
David Laidlaw, Andries van Dam, George Karniadakis, Jan Hesthaven

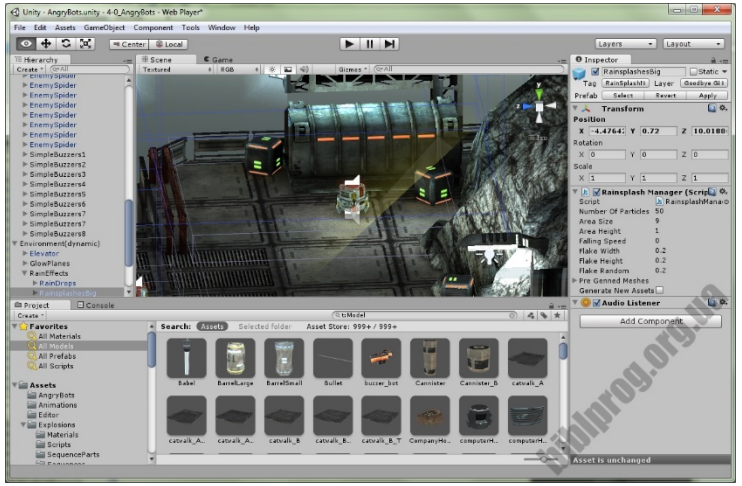
Applications



Paraview
Scientific data visualization



A-Frame
VR web framework



Unity 3D
Game development platform

Focus : Head Mounted Display with 6DOF

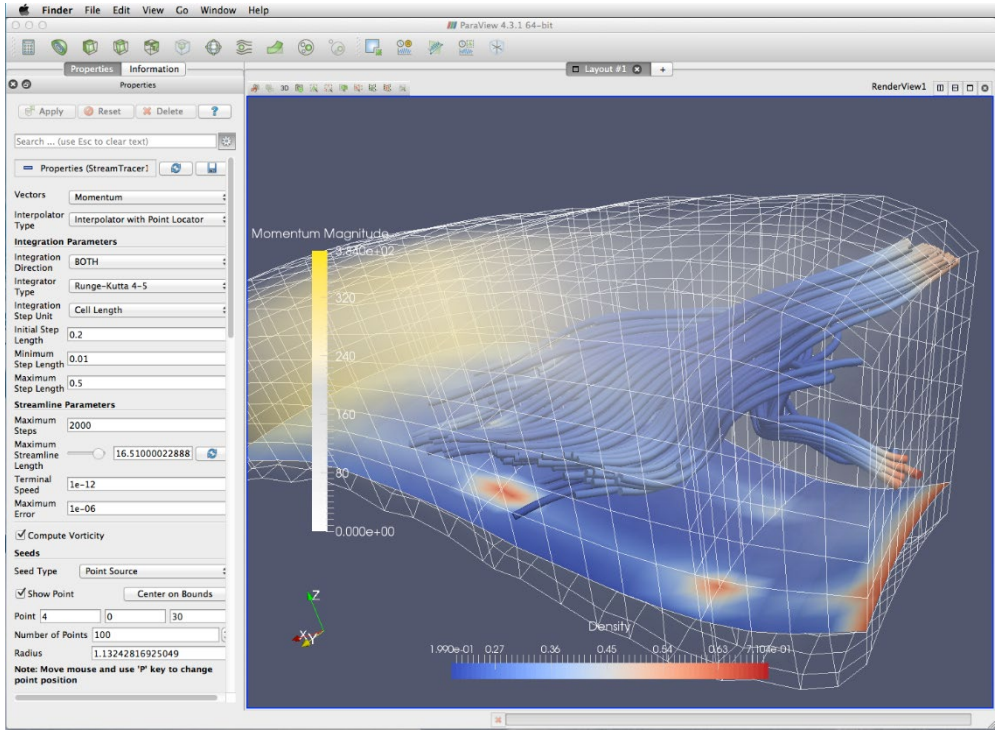
Device	Oculus Rift	HTC Vive	Windows Mixed Reality	Oculus Quest
Image				
Tracking	External (cameras)	External (laser)	Internal (cameras)	Internal (cameras)
PC required	yes	Yes	Yes	No
SDK	Oculus SDK SteamVR	SteamVR	Windows MR SDK SteamVR	Oculus SDK

https://store.steampowered.com/app/250820/SteamVR/
https://store.steampowered.com/app/719950/Windows_Mixed_Reality_for_SteamVR/

Paraview - Kitware, Inc.

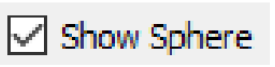
- Open-source, multi-platform visualization application
- Uses VTK
- Many datatypes and features
  - Structured, unstructed, polygonal, images …
  - Vector fields (glyphs), Contour, isosurfaces
  - Filters, e.g. Clipping

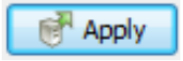
  And many more
- Plugins, e.g. VR



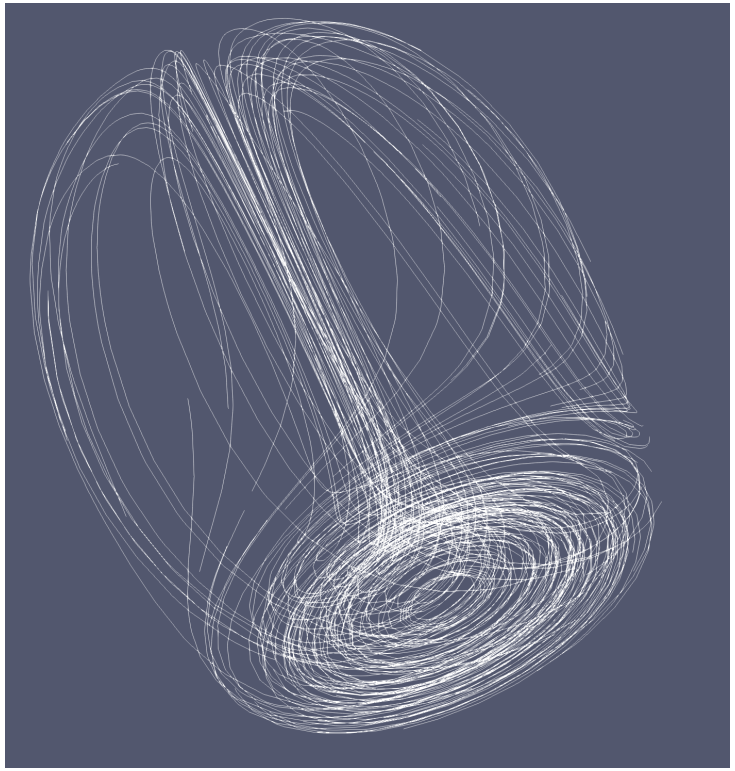
Vector Streamlines Example (Paraview Tutorial)

1. Open the file disk_out_ref.ex2, load all variables, 

2. Add the stream tracer filter  to disk_out_ref.ex2.
3. Change the Seed Type parameter in the properties panel to Point Source.
4. Uncheck the Show Sphere checkbox 

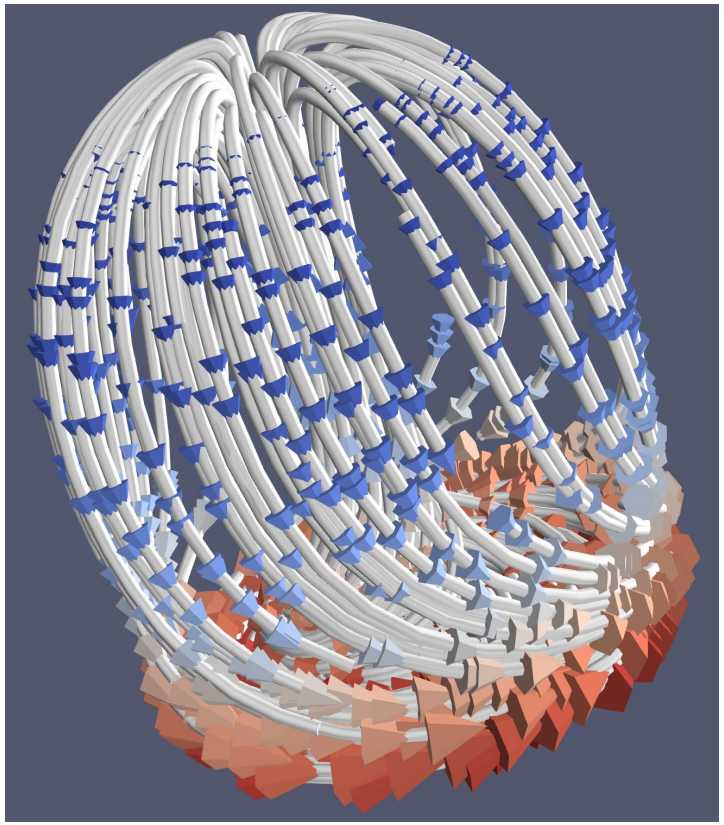
in the properties panel under the seed type.
5. Click the 

button to accept these parameters.



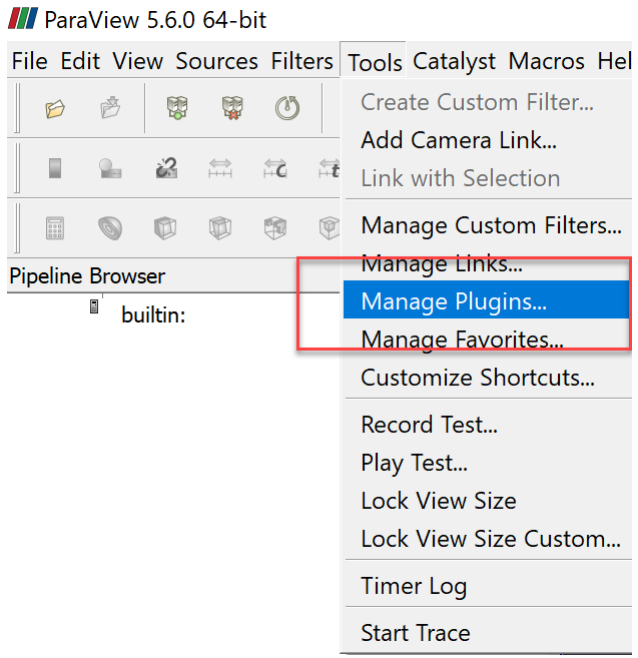
Vector Streamlines Example (Paraview Tutorial)

1. Use the quick launch (ctrl+space Win/Linux, alt+space Mac) to add the Tube filter to the streamlines.
2. Hit the Apply button.
3. Select StreamTracer1 in the pipeline browser.
4. Add the glyph filter to StreamTracer1.
5. In the properties panel, change the Glyph Type option to Cone.
6. In the properties panel, change the Vectors option to V.
7. Scrolling down a bit, change Scale Mode to vector.
8. Click the reset button next to Scale Factor.
9. Hit the Apply button.
10. Color the glyphs with the Temp variable.

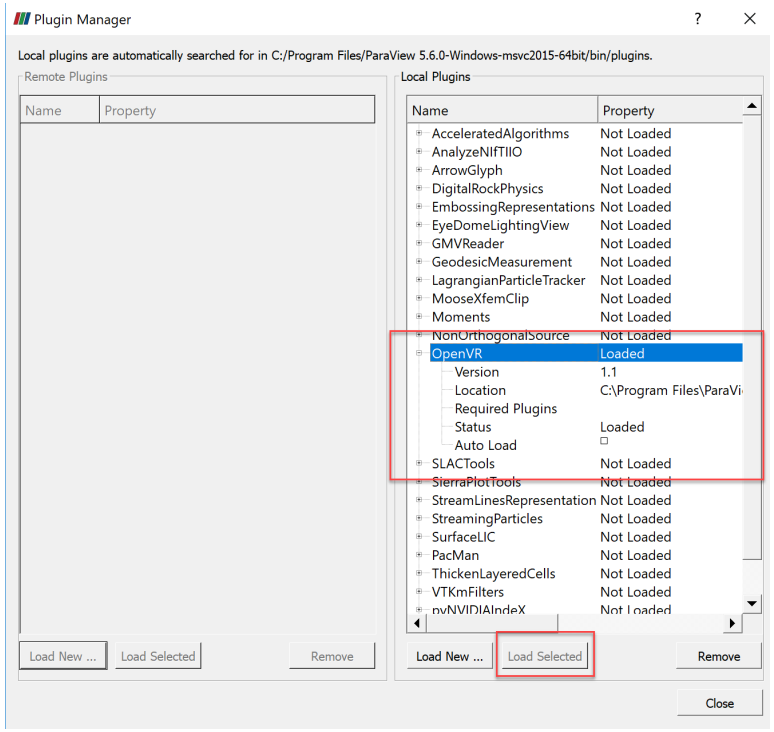


VR Plugin

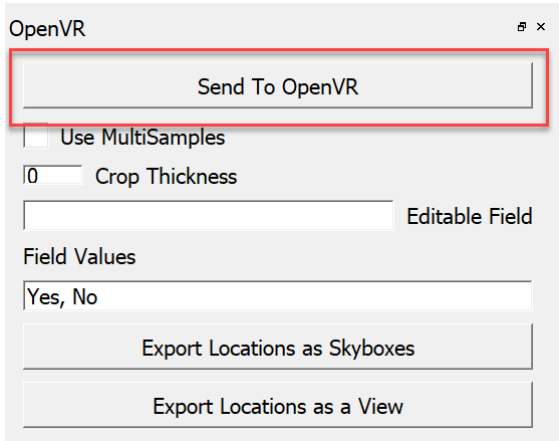
1. Go to Tools->Manage Plugins



2. Select OpenVR plugin and load selected

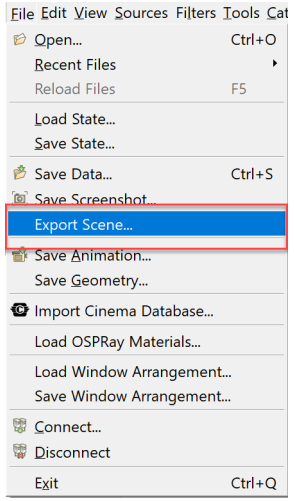


3. Send to VR

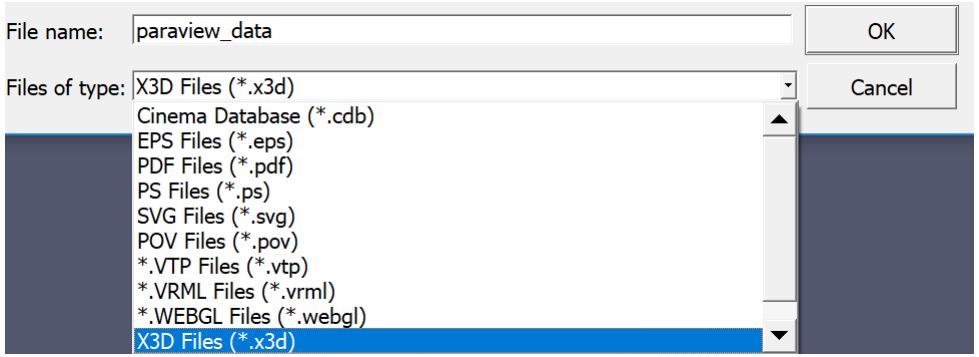


Export for A-Frame and Unity 3D

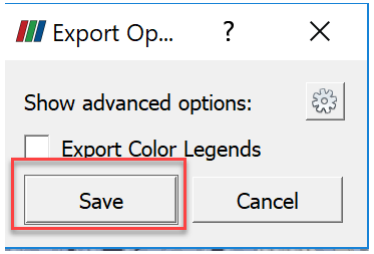
1. Go to File->Export Scene



2. Save as x3d file



3. Click Save

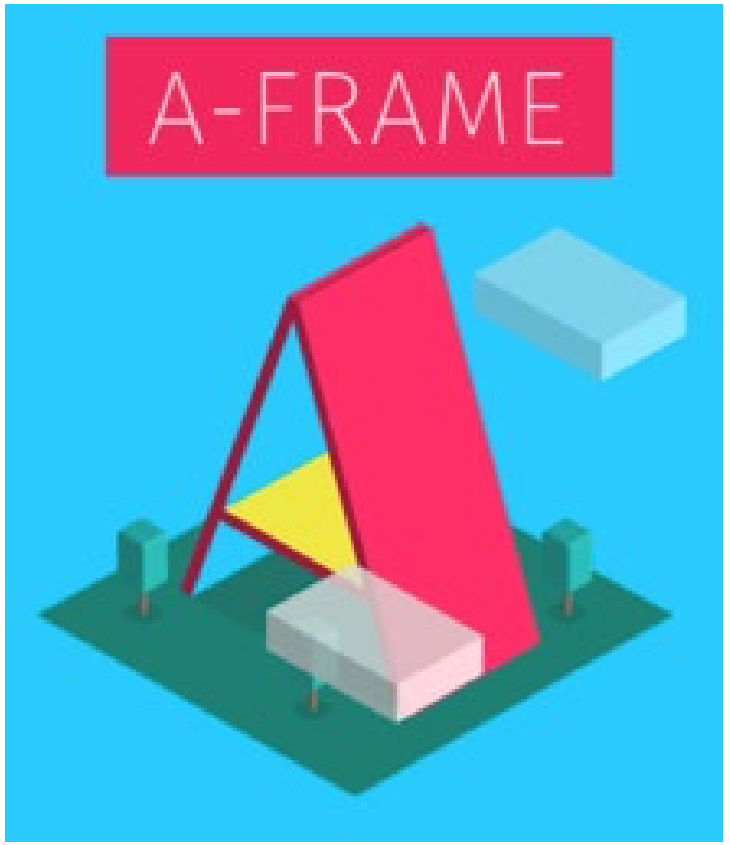


More Information

- Tutorial : https://www.paraview.org/Wiki/The_ParaView_Tutorial
- Forum : https://discourse.paraview.org/
- Youtube: https://www.youtube.com/results?search_query=paraview
- Book : https://www.paraview.org/paraview-guide/

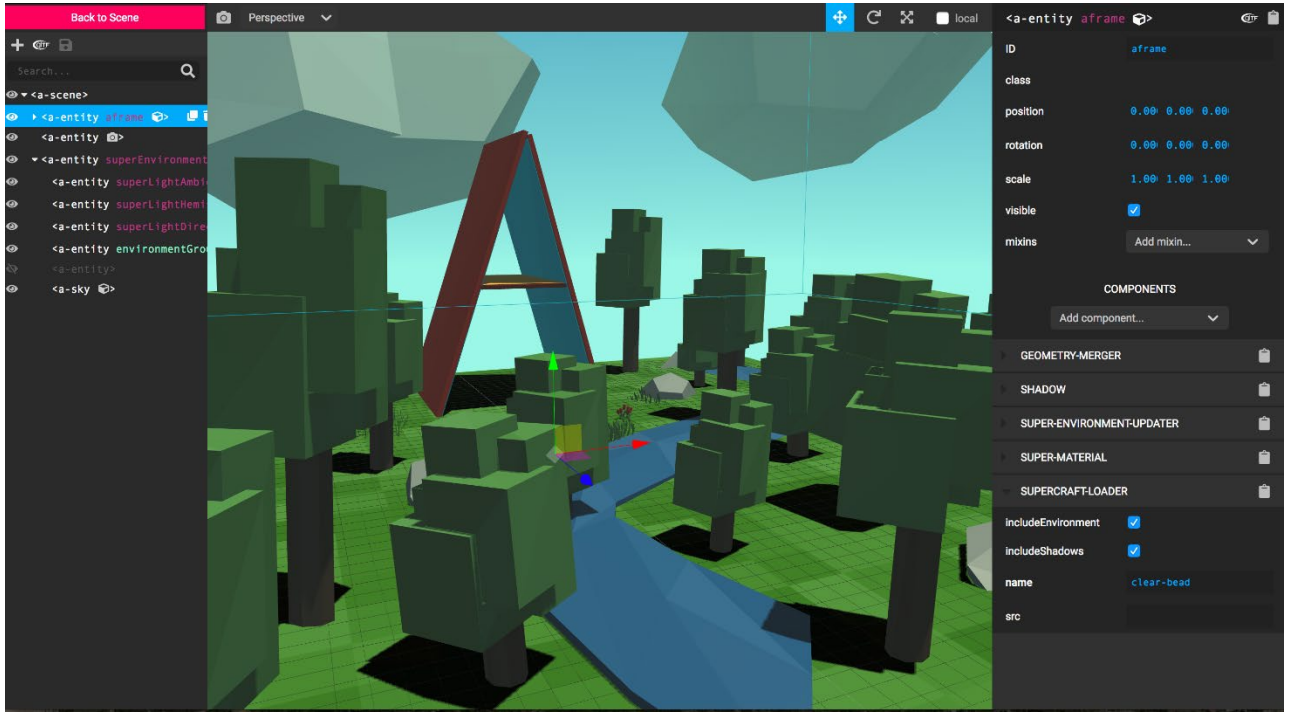
A-Frame

- web framework for building virtual reality (VR) experiences
- provides a declarative, extensible, and composable structure to three.js
- supports most VR headsets such as Vive, Rift, Windows Mixed Reality, Daydream, GearVR, Cardboard, Oculus Go, and can even be used for augmented reality.
- Only supported in some browsers, e.g. Firefox https://support.mozilla.org/en-US/kb/view-virtual-reality-firefox-webvr



Features

- VR Made Simple
- Declarative HTML
- Entity-Component Architecture
- Cross-Platform VR
- Performance
- Visual Inspector
- Components
- Proven and Scalable



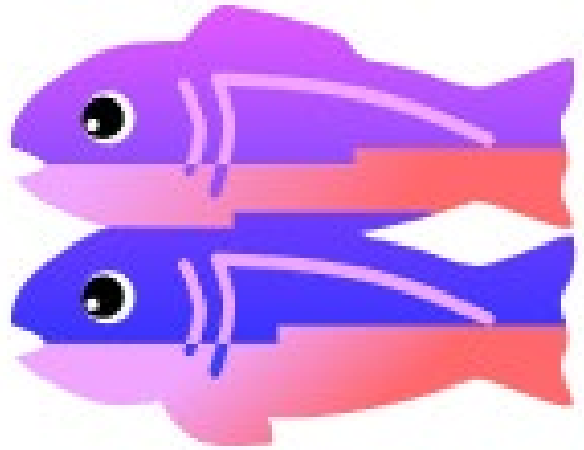
Entity Component System

- Entity :
  - Container, base of all objects in the scene
- Components :
  - Reusable modules, providing appearance, functionality, etc.
- System :
  - Optional: global scope, management, and services for classes of components

```
<a-entity geometry="primitive: sphere; radius: 1.5"
          light="type: point; color: white; intensity: 2"
          material="color: white; shader: flat; src: glow.jpg"
          position="0 0 -5"></a-entity>
```

Glitch

- https://glitch.com
- Live online editor
- Remix other projects
- Instant hosting and automated deployment



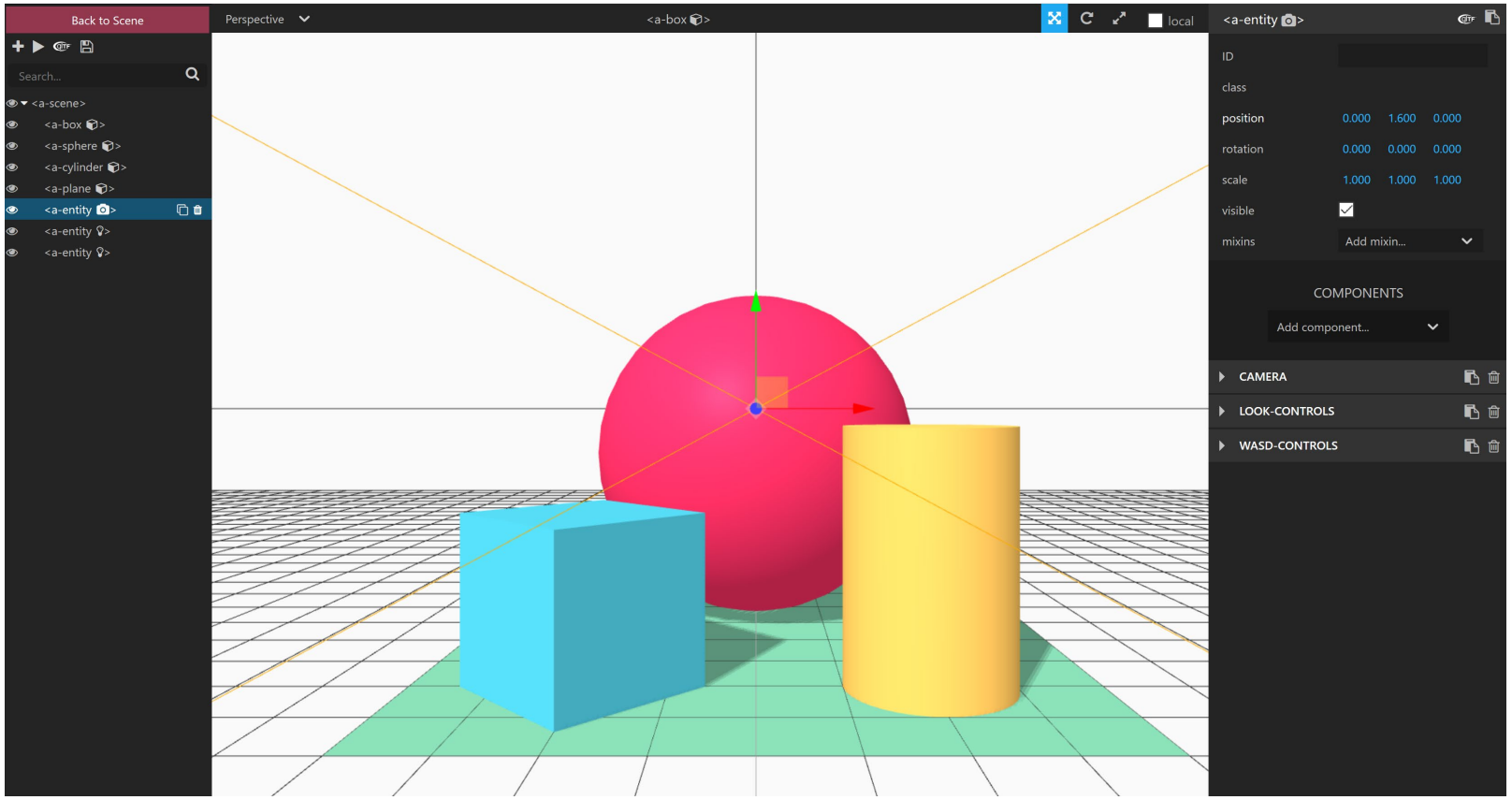
Basic A-frame

- https://glitch.com/~aframe
- Click view source and remix to edit

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hello, WebVR! • A-Frame</title>
    <meta name="description" content="Hello, WebVR! • A-Frame">
    <script src="https://aframe.io/releases/0.9.2/aframe.min.js"></script>
  </head>
  <body>
    <a-scene background="color: #FAFAFA">
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9" shadow></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E" shadow></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D" shadow></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4" shadow></a-plane>
    </a-scene>
  </body>
</html>
```

Visual Inspector

- Ctrl + Alt + I



Adding VR

- Simply click the VR icon and the scene will be displayed in VR

- Controls are a little bit more difficult but many components exist, e.g.
  - https://unpkg.com/super-hands@2.1.0/

```
<script src="//cdn.rawgit.com/donmccurdy/aframe-extras/v3.12.4/dist/aframe-extras.min.js"></script>
<script src="https://unpkg.com/super-hands@2.1.0/dist/super-hands.min.js"></script>
```

Adding Controls with Superhands

- Hands and Controllers are a new entity

```
<a-entity sphere-collider="objects: a-box" super-hands hand-controls="right"></a-entity>
```

- Control type is defined by a component, e.g.

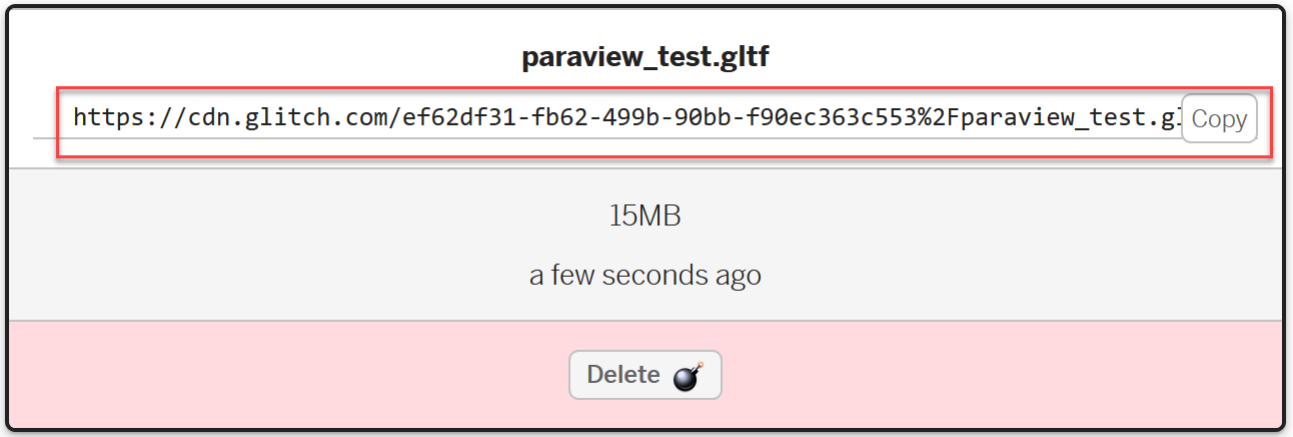
  tracked-controls <!-- Hook into the Gamepad API for pose. -->
  vive-controls <!-- Vive button mappings. -->
  oculus-touch-controls <!-- Oculus button mappings. -->
  hand-controls <!-- Appearance (model), gestures, and events. -->
  laser-controls <!-- Laser to interact with menus and UI. -->
  sphere-collider <!-- Listen when hand is in contact with an object. -->

- Add interaction to object

```
<a-box hoverable grabbable stretchable draggable dropppable position="0 0 -1" rotation="0 45 0" color="#4CC3D9" shadow></a-box>
```

Uploading our Data

- Go to assets
- Drag and Drop the file (paraview_test.gltf)
- Click on the asset
- Note down the url



Adding our Data to the Scene

- Custom models can be added as assets

```
<a-assets>
   <a-asset-item id="paraview_test" src="https://cdn.glitch.com/04c585d0-a754-45fd-8bc8-9caf3361dab1%2Fparaview_test.gltf"></a-asset-item>
</a-assets>
```

- and added to the scene as entity

```
<a-entity gltf-model="#paraview_test"></a-entity>
```

- with position and rotation

```
<a-entity gltf-model="#paraview_test" rotation="-90 0 0" scale="0.1 0.1 0.1" position="0 1.2 -0.5"></a-entity>
```

Adding Interaction and Animation

- Interaction

```
<a-entity sphere-collider="objects: .collider;" super-hands hand-controls="right"></a-entity>
<a-entity class="collider" hoverable grabbable stretchable draggable dropppable
          gltf-model="#paraview_test" rotation="-90 0 0" scale="0.1 0.1 0.1w" position="0 1.2 -0.5"></a-entity>
```

- Animation

```
animation="property: rotation; to: -90 360 0; loop: true; dur: 10000; easing: linear"
```

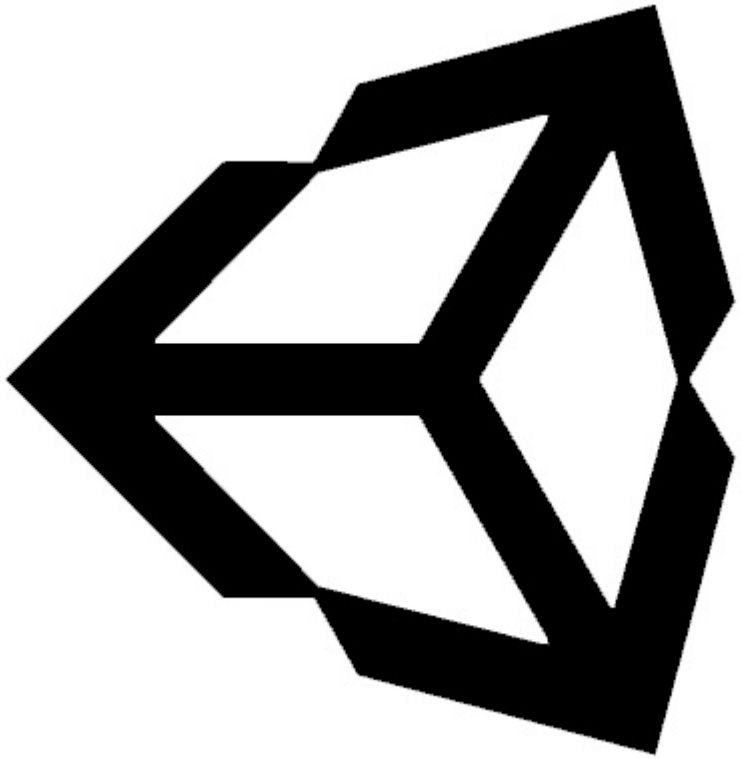
https://glitch.com/edit/#!/dscov-paraview-file

More Information

- Documentation : https://aframe.io/docs/0.9.0/introduction/
- Slack : https://aframe.io/slack-invite/
- Community : https://stackoverflow.com/questions/tagged/aframe
- Youtube: https://www.youtube.com/results?search_query=%22a-frame%22+vr
- NPM : https://www.npmjs.com/search?q=aframe-component

Unity 3D

- https://unity.com/ (use 2018.4.1f1)
- cross-platform game engine (2D + 3D)
- Unity editor + C# Scripts (https://visualstudio.microsoft.com/downloads/)
- more than 25 different platforms, including mobile, desktop, consoles, and virtual reality
- half of new mobile games and 60 percent of augmented reality and virtual reality content
- not only used for games, e.g. Unity Machine Learning Agents



Core Elements of Unity

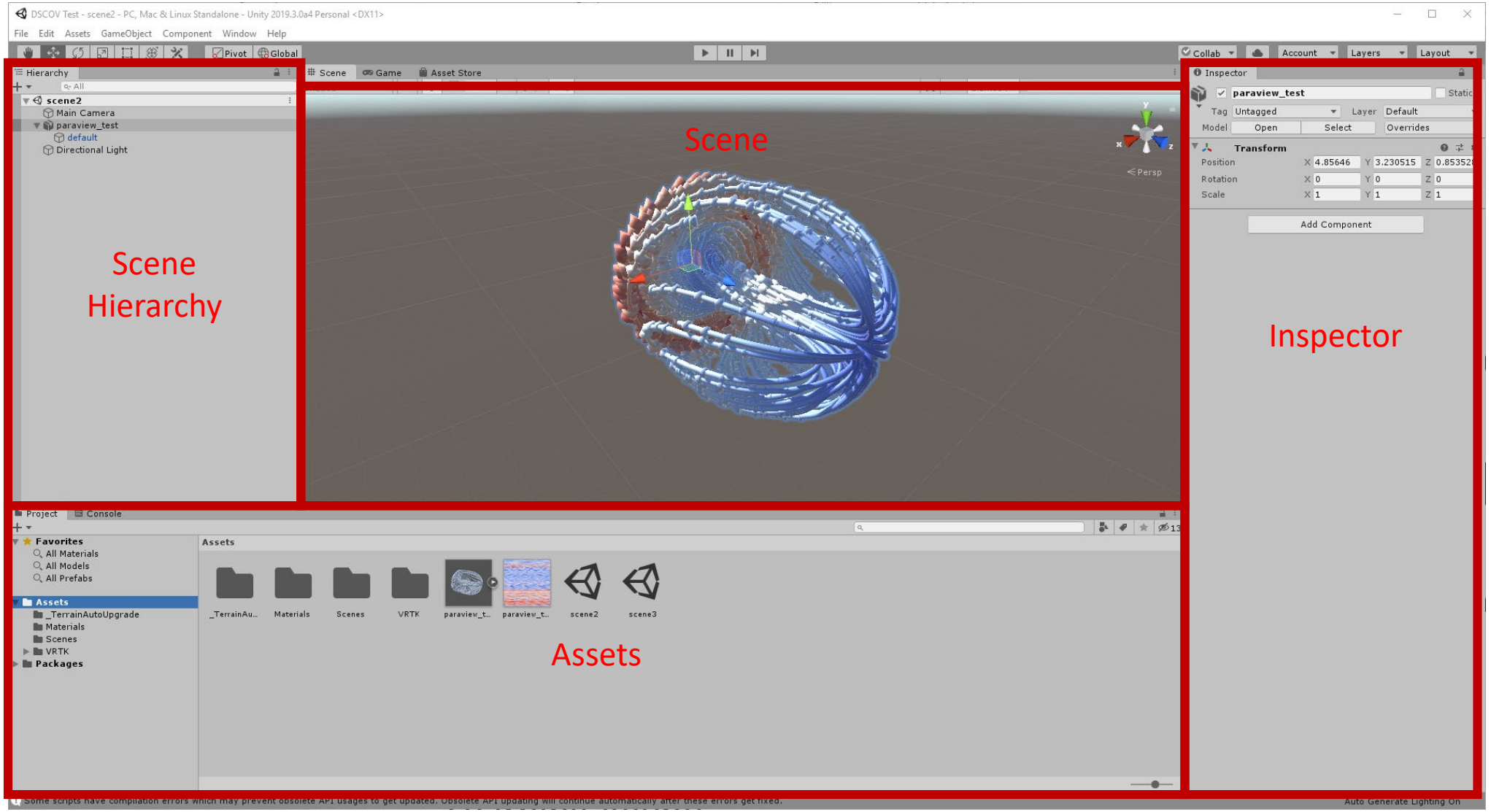
- GameObject
  - fundamental objects in Unity that represent characters, props and scenery
  - Acts as a container (similar to an Entity in ECS)

- Component
  - Adds functionality to the GameObjects
  - E.g. Transform (Position and Rotation)
  - Can be scripted using C#

- Asset
  - Any media that can be used
  - Can be downloaded from the Asset store

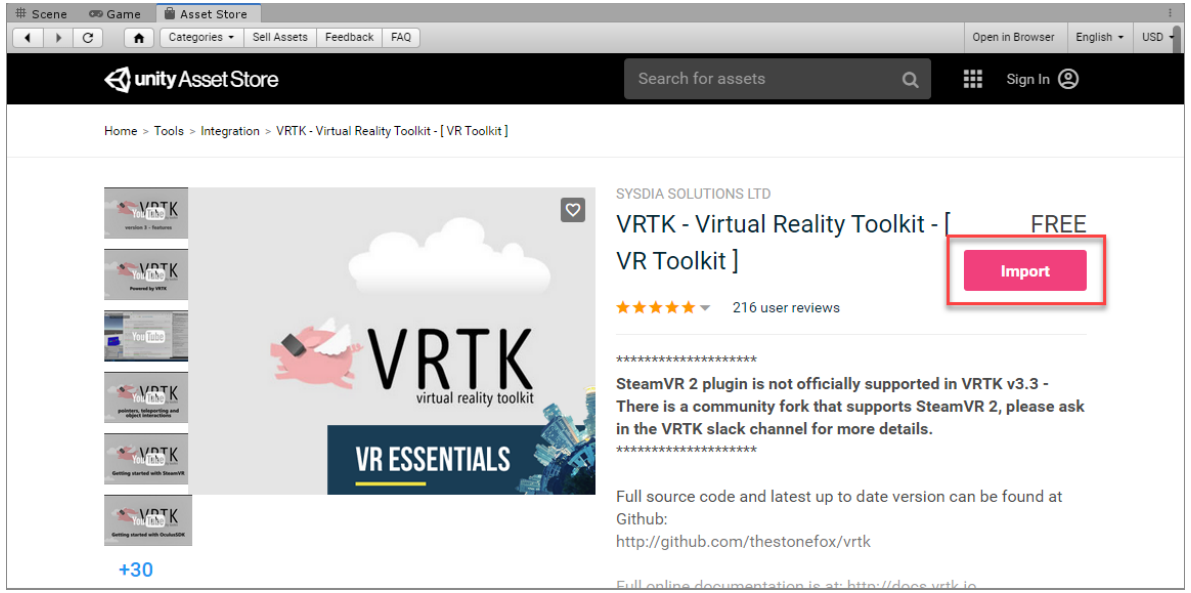
- Prefab
  - GameObject complete with all its components, property values, and child GameObjects as a reusable Asset
    .

User Interface



VRTK – Virtual Reality Toolkit

- https://vrtoolkit.readme.io/
- Prefabs and scripts for VR
- Can be installed from the Asset Store



Note: VRTK currently has a bug with the newest Version of Unity 3D. To fix exchange

renderer.material.SetTexture("_MainTex", new Texture());

to

renderer.material.SetTexture("_MainTex", Texture2D.whiteTexture);

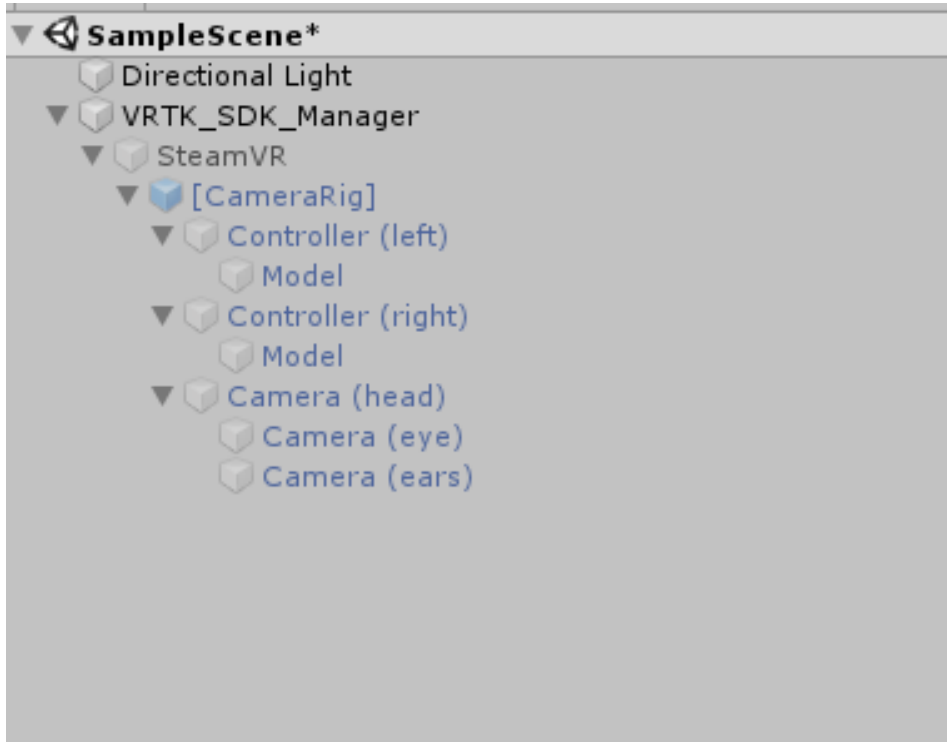
At line 146 of $PROJECTFOLDER\Assets\VRTK\Scripts\Interactions\Highlighters\VRTK_MaterialColorSwapHighlighter.cs

SteamVR Plugin

- VRTK requires version 1.2.3 https://github.com/ValveSoftware/steamvr_unity_plugin/releases/download/1.2.3/SteamVR.Plugin.unitypackage
- Drag and Drop to Assets
- Add OpenVR package
  - Go to *Window -> Package Manager* and enable the OpenVR (desktop) package

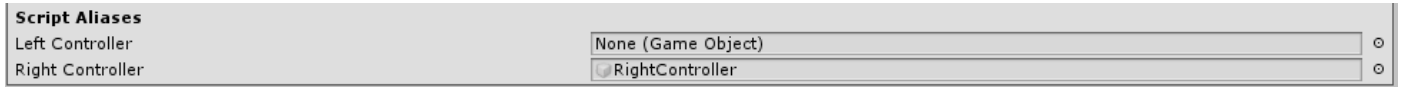
Setting up VRTK and SteamVR

1. Create an empty GameObject and call it VRTK_SDK_Manager
2. Add the VRTK_SDKManager script it (Add Component)
3. Add another GameObject to the VRTK_SDK_Manager and call it SteamVR
4. Add VRTK_SDKSetup script to it.
5. Drag and Drop the SteamVR Prefab Camera Rig to it
6. On the SteamVR set the SDK Selection to the respective VR SDK.
7. Click “Populate now”
8. On the VRTK_SDK_Manager click “Auto Populate”
9. Delete the Main Camera GameObject
10. Click the play button to test



Adding a Controller

1. Create an empty GameObject and call it RightController
2. Drag and Drop it to the right controller of the VRTK_SDK_Manager



3. Add the VRTK_InteractTouch script to the RightController
4. Add the VRTK_InteractGrab script to the RightController
5. Create a cube : Gameobject->3D Object -> Cube
6. Set position to (0 1 0);
7. Add the VRTK_InteractableObject script to the Cube
8. Set is Grabbable to true
9. Click the play button to test

Moving around

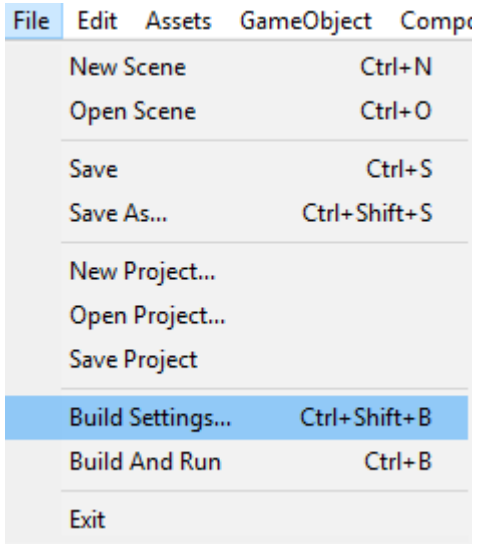
1. Add Ground plane : GameObjects -> 3D Objects -> Plane
2. Add VRTK_Pointer script to RightController
3. Add VRTK_Straight_Pointer_Renderer script to RightController
4. Set the Pointer Renderer of the VRTK_Pointer to the renderer component
5. Add VRTK_Basic_Teleport script
6. Click the play button to test

Add our Data

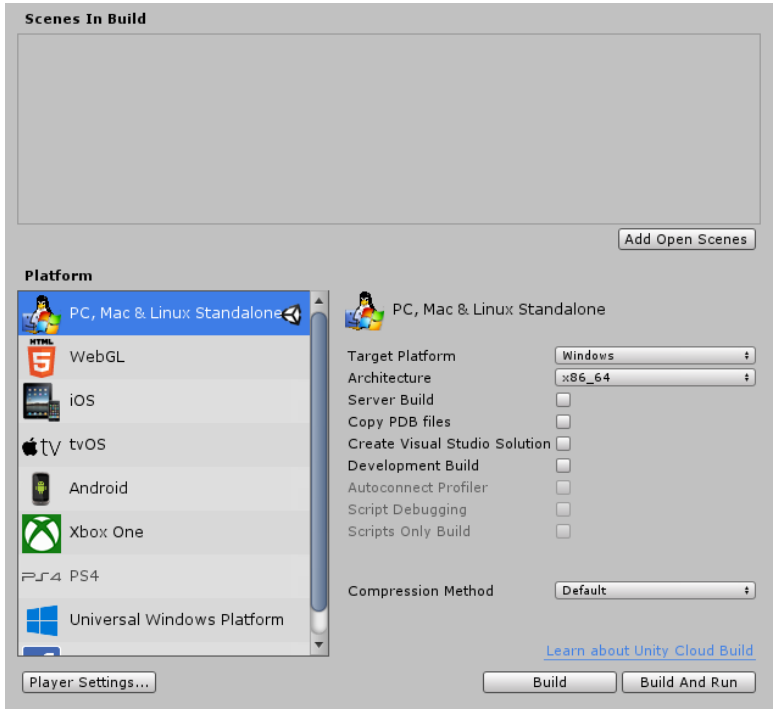
- Drag and Drop paraview_test.obj and paraview_test_tex.png to the asset area
- Drag the object to the scene
- Adjust scale, position and orientation
- Drag the png-file on the object in the scene
- Add VRTK_InteractableObject script to the object and set is Grabbable
- Add a collider, e.g. sphere collider and adjust the center and radius
- Click the play button to test

Creating an Executable

1. Click File->Build Settings



2. Click Build



More Information

- User Manual : https://docs.unity3d.com/Manual/index.html
- Tutorials : https://unity3d.com/learn/beginner-tutorials
- Community : https://unity3d.com/community
- Youtube: https://www.youtube.com/results?search_query=unity+
- VRTK : https://vrtoolkit.readme.io/

Thanks for your attention

Contact us at support@ccv.brown.edu