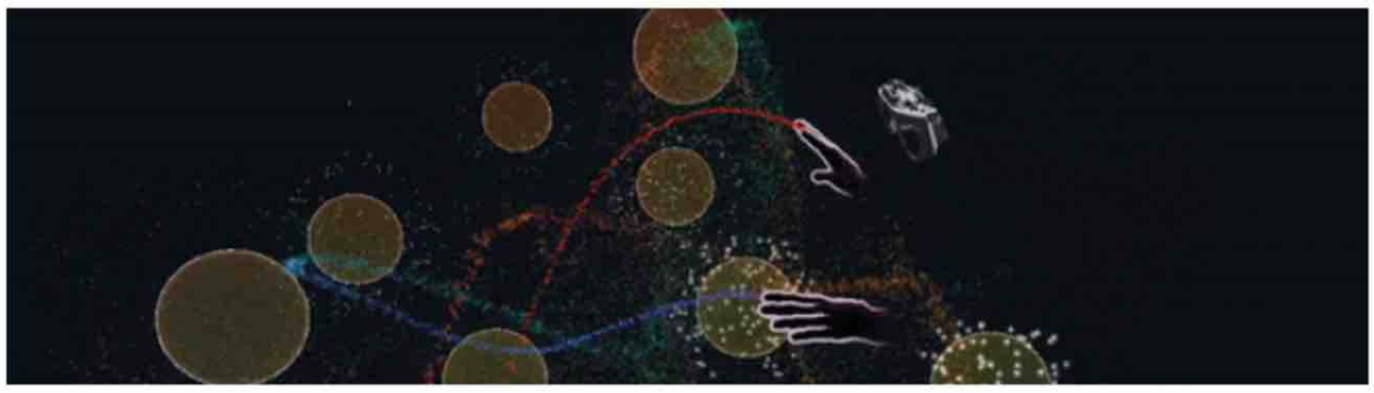
Home

Sam Bilbow edited this page 2 hours ago · 7 revisions

polaris~



An Audiovisual Augmented Reality Experience Built on Open-Source Hardware and Software

Platform Windows Environment Unity/Pd Publication NIME Guide Wiki Project Blog Discord XRt Space

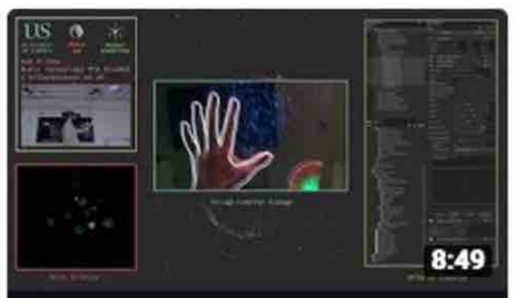
If an AR system can be thought of as one that combines real and virtual processes, is interactive in real-time, and is registered in three dimensions; why do we witness the majority of AR applications utilising primarily visual displays of information? I propose a practice-led compositional approach for developing multisensory AR experiences', arguing that, as an medium that combines real and virtual multisensory processes, it must explored with a multisensory approach.

This project uses the open-source Project North Star HMD from Leap Motion alongside bone-conduction headphones to deliver a spatialised audio-visual experience via Unity called polaris~. This repository started off as a fork of the Software Companion for Project North Star, hence the other repository contributors and long list of commits. However, the experience itself including all audio-visual / artistic / musical content was added afterwards.

To get started with `polaris~`, follow the guide shown on the sidebar of this page. First you may want to familiarise yourself with the components used in the experience and the features they provide.

Presentation

32 views • 6 months ago

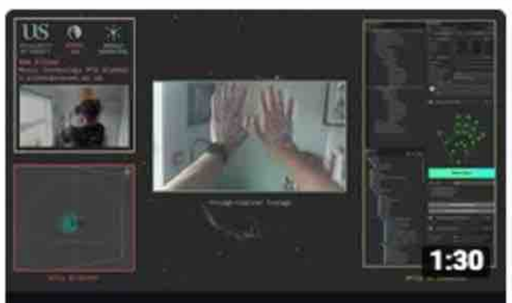


Demonstration

18 views • 1 year ago

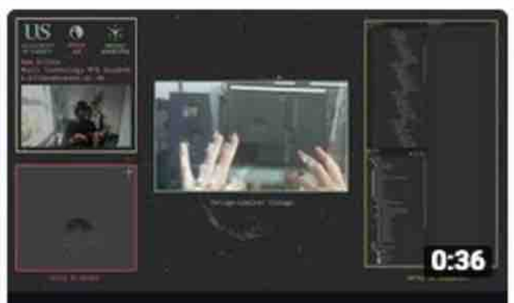
Prototype

205 views • 1 year ago



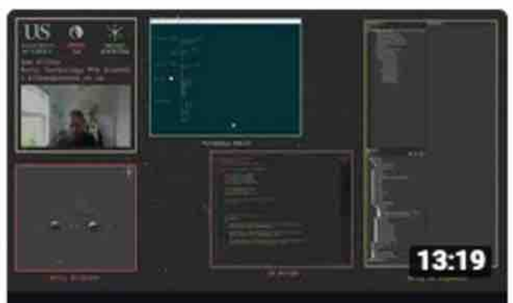
Palm Synth

80 views • 1 year ago



Finger Synth

43 views • 1 year ago



LibPd Explainer

184 views • 1 year ago

Citation

Bilbow, S. (2022). Evaluating polaris~ - An Audiovisual Augmented Reality Experience Built on Open-Source Hardware and Software. NIME 2022. https://doi.org/10.21428/92fbeb44.8abb9ce6

or with BibTeX

Clone this wiki locally

https://github.com/sambilbow/po

Components

Sam Bilbow edited this page 2 hours ago · 1 revision

- Software Companion for the Project North Star open-source AR headset that allows developing Unity scenes with MRTK/Leap Motion assets.
- LibPdIntegration: a wrapper for libpd that allows for the implementation of Pure Data patches into Unity
- Automatonism: a library of Pure Data Vanilla patches that emulate modules of a synthesizer.
- A set of example scripts and scenes that use the above components to demonstrate possible interactions between head/hand tracking and patch parameters in Pd, with the chief aim of creating a set of expressive multisensory AR instruments / experiences.

Features

Hardware

- Six degrees-of-freedom (3D position / orientation) head tracking via Intel T261
- 90 fps, 170° hand tracking via Ultraleap
- Single piece optical combiner allowing for up to 110° horizontal FoV
- 2x 120Hz displays per-eye for a total resolution of 2880x1600
- 2x 3-metre cables (1x miniDP, 1x USB-A 3.1)
- Spatial audio AR (the ability to hear localised sound whilst being able to hear your real audio environment) via Unity3D and Aftershokz Aeropex bone conduction headphones.

Engine

- The ability to create 3D scenes that contain 'GameObjects' that in turn can have visual attributes such as 3D meshes, material colours, and textural properties; physical attributes such as edges, position, mass, velocity and real-time parameterisation via C# scripting.
- Thanks to the Software Companion, the headset is created as a GameObject with real-time position / orientation.
- Thanks to LeapMotion, hands (all the way down to individual finger joints) are created as GameObjects with real-time position / orientation relative to the headset.

Audio

- LibPdIntegration uses **native Unity3D audio spatialisation**. This is great because it means that a GameObject can output the signal of a Pd patch whilst moving, rotating and scaling. The effect of these can perceived in real-time because the AudioListener is anchored to the real-time headset position. This, for example, means that the volume of a Pd patch whose signal is being transmitted from a GameObject located in space is automatically scaled dependent on its distance to the participants head (quieter as it gets further away, louder as it is brought closer).
- LibPdIntegration can **'instance' Pd patches**, meaning it can use one patch on multiple GameObjects, but maintain processes like randomness within them as they are technically different 'instances' or versions of the patch.
- Pure Data allows **extended audio techniques** through an extensive library of algorithmic 'objects' that can create and manipulate audio signals.
- LibPdIntegration allows **real-time parameter control** in Unity of any object in a Pd patch via "receive" objects and a specific C# method.
- The combination of "Play Mode" toggling in Unity, and the quick visual patching style of Pure Data means that audio-visual interactions can be **prototyped very rapidly**

Software Requirements

Sam Bilbow edited this page 2 hours ago · 4 revisions

Compatibility

- Tested with Windows 10, 11
- Tested with PureData 0.51-0

Installation

- Download the repository as a .zip or via `git clone https://github.com/sambilbow/polaris` in your terminal emulator

Unity Editor

- 2020.3.18f1 available @ Unity3D Archive

Ultraleap SDK

- Gemini 5 or later available @ LeapMotion

Intel Realsense

`Note:` this device is now end-of-life, running the project with other 6DoF sensors will require Software Companion compatibility with them first.

- Intel® RealSense™ SDK 2.0 available @ GitHub Release

PureData

- Pure Data 0.51-0 (tested) available @ PureData
- Pure Data Latest available @ PureData

Clone this wiki locally

`https://github.com/sambilbow/po`

Hardware Requirements

Sam Bilbow edited this page 2 hours ago · 5 revisions

PC Specifications

- Minimum system requirements for the North Star headset are found on the Project Northstar Documentation Website. I used a custom PC with an i9 CPU, AMD RX590 GPU, running Windows 11.
- A DisplayPort connection that can output 4k@60Hz for the headset
- A USB (A) 3.2 Gen 2 connection (10Gb/s) for the headset
- The headphones used require a stable Bluetooth audio connection.

Visual

This project guide assumes you have a calibrated Project North Star headset. If you don't, check out the below link to get started, and make use of their Discord server!

- Project North Star Headset (used Combine Reality Deck X)
  - Ultraleap Stereo IR 170 info
  - Intel Realsense T261 info
    - `note:` this device is now end-of-life, Xvisio SeerSense XR50 is now recommended, ask on Project North Star's Discord Server for more details.

Audio

- Bone Conduction Headphones (used Aftershokz Aeropex, now Shokz Openrun)

Clone this wiki locally

`https://github.com/sambilbow/po`

Mappings

Sam Bilbow edited this page 1 hour ago · 3 revisions

Unity Scene

1. Add the Unity project in `polaris/unity/` to Unity Hub
2. Open `polaris/unity/EskySettings.json`, to include your headset offsets and update your `"displayWindowSettings"` to suit your monitor + headset display layout.
3. Open the Unity project.
4. Open the polaris~ scene: `polaris/unity/Assets/Scenes/Experiences/polaris_study.unity`

Hand Menu

polaris~ features a hand menu that shows to the right of the the left palm when it is facing the participant. The button toggles streams of particles from the users hands.

Particles

The particle streams make use of the PureData patch located at `polaris/unity/Assets/StreamingAssets/PdAssets/polaris_study_hands/main.pd`. Feel free to open that up and check out the mappings.

Orbs

The orbs make use of the PureData patch located at `polaris/unity/Assets/StreamingAssets/PdAssets/polaris_study_sphere/main.pd`. Feel free to open that up and check out the mappings. ParticleForcefields are active on the floating Orb GameObjects, which means that the hand particles flow in interesting ways between the orbs.

More detailed information about musical parameter mappings can be found in the publication and blog for this project. Additionally, the NIME presentation and experience demonstration provides audio-visual aid in understanding the parameter mappings.

Publication NIME

Project Blog

Presentation

32 views • 6 months ago

Demonstration

18 views • 1 year ago

Clone this wiki locally

`https://github.com/sambilbow/po`