**January 2020 Workshop · XR User Interfaces**

**GOALS**

* **Understanding the XR medium**
* **Planning and prototyping for XR**
* **Understanding techniques for UX/UI development**
* **Owning an overview of tools**
* **Setting up a sandbox in Unity / Web VR**

**Introduction** 1h

* Presentation
* Fast story of XR
* Definitions of XR
* Applications of XR
* 5G: What’s expected

**Computer Graphics Basic Concepts**

* Euclidean Space
* Vertex, edges, polygons
* Tessellations
* Modeling
* Shaders

**Computer Vision**

* Computer Vision
* AI Image Detection / Odometry
* Deep learning

**Projections**

* 2D
* Stereoscopic
* Stereographic
* Equirectangular
* Cube mapping

**Peripherals** **2020**

* HMD
* Cameras
* Controllers
* Hand Tracking

**VR Environments Features**

* Plausibility
* The tech constraints in 2019
* Resolution
* Cadence
* Camera position
* Camera movements 3DOF /6 DOF
* Self Awareness

**User interface in XR**

* Diegetic vs. non diegetic
* Elements position regarding the camera
* Elements composition
* Interactive elements
* Feedback and States
* Reticle feedback
* Objects feedback
* Operations
* Typography
* Interactables
* Grids, Safety
* Buttons
* Shadows
* Tooltips
* Keyboards

**Techniques of interaction**

* Degrees of interaction
* Raycasting
* Gaze
* Controllers
* Trackers
* Handtracking

**VR Multimedia Creation**

* Photo
* Video
* Motion graphics
* 3D Modeling

**Engines**

* Unreal
* Unity
* A-Frame Javascript
* Frame AR
* AR.js
* React 360
* AR Studio
* ARKit
* ARCore
* Three.js
* More AR

**Deployment Platforms**

* Web XR
* Unity

**Workshop**

**Practice proposal - WebVR**

With this workshop **we will learn how to make a toolset for VR, how to integrate it and how to deploy it.** The idea is to have a playful environment to test and develop further ideas.

It’s super easy to add and remove elements into the playground. We just need to know very basic HTML selectors.

* Let’s have our **WebVR** set up ready!
  + **Get your text editor running**
    - [**https://code.visualstudio.com/**](https://code.visualstudio.com/)
    - [**https://www.sublimetext.com/3**](https://www.sublimetext.com/3)
    - [**https://atom.io/**](https://atom.io/)
  + **Boilerplate in HTML**
    - <https://aframe.io/docs/1.0.0/introduction/>
  + **Download this repository**:
    - <https://github.com/screeeen/vr-sandbox>
  + **Inspector**
    - **Ctrl+Alt+I**
  + **Explanation for an interactive cursor**
  + **Animations**
  + **Import Images**
  + **Import Models**
  + **Making an Interactive Card**
  + **The A-frame Registry** [**https://aframe.io/aframe-registry/**](https://aframe.io/aframe-registry/)
  + **If you want to test this on a headset Log in on your github (or Sign up on a fresh one!):**
    - <https://github.com/>
  + **How to upload your code to the github repository**
    - At the github repositories page click **New**
    - Type the project name at the Repository name
    - Create your project
      * Some index.html ( follow the boilerplate!)
      * <https://glitch.com/edit/#!/aframe?path=index.html:1:0>
      * Copy and paste in the terminal the indications of your repository
      * **Add a remote:** git remote add origin
      * **Push the code:** git push -u origin master
      * You can check your remote: git remote -v
      * Secuence to push code (write in the terminal):
        + git add .
        + git commit -m “name of the commit”
        + git push origin master
      * Every time the code is pushed, if you have a github page, the page will be updated
  + **How to deploy your website on github.io**
    - On your repository page, go to settings
    - Scroll down to **Github Pages**
    - Select ‘master branch’ on Source dropdown
    - That’s it! Check the URL given at the Github pages section
    - **Your site will be published at <URL>**

**WEB AR with a-frame AR**

* + **Boilerplate in HTML**
    - <https://aframe.io/blog/arjs/>
  + **Download the Hiro Mark**
    - [**https://jeromeetienne.github.io/AR.js/data/images/HIRO.jpg**](https://jeromeetienne.github.io/AR.js/data/images/HIRO.jpg)

**VR with Unity**

* Let’s have our **Unity VR** set up ready!
* https://www.dropbox.com/sh/7xjbtc8ir50j55n/AADSLJ5fXCMa-1OTEShd7adsa?dl=0
* Installing **Unity**
  + What is Unity: <https://en.wikipedia.org/wiki/Unity_(game_engine)>
  + **Latest version to install**. Just download and install: <https://store.unity.com/download?ref=personal>
* **IMPORTANT: Navigating!!!** [**https://docs.unity3d.com/Manual/SceneViewNavigation.html**](https://docs.unity3d.com/Manual/SceneViewNavigation.html)
* Download the **google VR Camera**!
  + Google VR SDK plugin for Unity <https://developers.google.com/vr/develop/unity/download>
  + control/command + Shift B triggers **build settings**
  + **On build settings** switch platform to **android**
  + Click **switch platform**
  + Then, once is switched, click on **player settings**
  + **XR settings**
  + Enable Virtual Reality
  + Select virtual reality SDK -> cardboard
* **Set up our Scene:**
  + ~~Assets>Import>Custom Package: open the downloaded .unitypackage camera~~
  + ~~Import all the components from Google VR Package~~
  + **~~Set Up the Camera~~**~~:~~
    - **~~Add to the scene~~** ~~the following prefabs to the Hierarchy:~~
    - **~~GvrControllerMain~~**
    - **~~GvrEditorEmulator~~**
  + **~~Add this script~~** ~~to the main camera~~
    - ~~Add~~ **~~GvRPointerPhysics~~** ~~Script~~
  + ~~And Add this prefab as a child of the camera:~~
    - **~~GvrReticlePointer~~**
  + Add the skybox
    - Create it in the materials->skybox
    - Swap it in Window>Lighting menu
  + Add some props, why not?!
    - <https://www.kenney.nl/assets>
    - <https://www.turbosquid.com/>
  + Be creative! Enjoy!

**WEB VR with React360**

* Let’s have our **WebVR** set up ready!
  + **Using the CLI (Command line interface) / Terminal**
    - Windows: <https://www.makeuseof.com/tag/a-beginners-guide-to-the-windows-command-line/>
    - Mac / Linux: <https://blog.teamtreehouse.com/introduction-to-the-mac-os-x-command-line>
  + **Installing node.js**
    - What is it? <https://en.wikipedia.org/wiki/Node.js>
    - Checking if we have a version installed: node -v
    - <https://nodejs.org/en/>
  + **Installing npm**
    - What is it? <https://docs.npmjs.com/about-npm/>
    - Checking if we have a version of  npm installed: npm -v
    - <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>
  + **Installing React 360**
    - What is it?
    - Checking if we have a version installed: react-360 -v
    - <https://facebook.github.io/react-360/docs/setup.html>
  + **Three.js**
    - What is it? <https://en.wikipedia.org/wiki/Three.js>
    - <https://threejs.org/>
    - <https://threejs.org/editor/>

**Useful Pano Components**

<https://github.com/hggeorgiev/CentroUI>

<https://github.com/SpectivOfficial/live-tour-lab>

**Image Conversions and editings**

<https://graphicdesign.stackexchange.com/questions/84774/how-to-convert-a-normal-photo-into-an-equirectangular-image>

<https://jaxry.github.io/panorama-to-cubemap/>

<https://www.jamesfmackenzie.com/2016/10/18/convert-equirectangular-projection-to-cube-faces>

<https://www.360toolkit.co/preview-panorama-viewer.html>

<http://paulbourke.net/miscellaneous/sphere2persp/>

<https://medium.com/@k_ymok/simple-equirectangular-architectural-rendering-using-blender-58b006834339>

<https://github.com/tmarrinan/cube2equirect>

<https://stackoverflow.com/questions/43413445/how-to-add-a-2d-texture-to-an-equirectangular-image>

Photoshop Brushes

<https://www.brusheezy.com>

<https://blog.hubspot.com/marketing/photoshop-brushes>

ForFree

<http://photopin.com/free-photos/equirectangular>

Attic

<https://medium.com/airbnb-engineering/prototyping-with-react-vr-4d5ab91b6f5a>

<https://en.wikipedia.org/wiki/Category:Vector_arcade_games>

<http://linoit.com>

Web VR

<https://medium.com/@joshmarinacci/running-webvr-on-the-oculus-go-60c3b0a2c3c7>

AR

<https://github.com/jeromeetienne/AR.js>

<https://github.com/nicolocarpignoli/GeoAR.js>

A-frame:

<https://github.com/aframevr/aframe/tree/master/examples>

GLTF models

<https://www.khronos.org/gltf/>

Fbx → gltf converter

<https://blackthread.io/gltf-converter/>

Oculus development

<https://developer.oculus.com/>