HackMe 2.0 - XperienZ

HKME-35 – 360º Live Virtual Tour with Integrated VR Experience

Version 1.0

Published on 01/13/2017

Document Revision History

Date	Version	Description	Author
01/12/2017	1.0	First Draft	Saurabh Nutan Sohoni

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to demonstrate capabilities of a web application in creating Virtual Reality (VR) experience for a user taking virtual tour of a scene at home, with live capture and streaming of scenic elements. With addition of media and navigation APIs to HTML5 specification, web applications are becoming more and more powerful and now can simulate a VR experience across platforms. These applications will be written in JavaScript and need only a web browser to run the simulation.

1.2 Scope

This demo is created by developing a web application for simulating VR experience using following APIs/libraries:

a. Live video capture and streaming using **WebRTC**
b. Dynamic video streaming mapping from 2D to stereoscopic environment using **Three.js**
c. Head movement tracking using HTML5 **deviceOrientation** API

1.3 Hardware requirements

Very few hardware components are required for this simulation. It includes:

a. A mobile device, smartphone or a tablet with working rear camera module, to be mounted on a stand, for capturing live scenic elements of a location.
b. A mobile device, smartphone of the screen size greater than 5 inch, for viewing simulated experience inside a VR headset.
c. A VR headset, even as crude as a Google Cardboard is sufficient.

1.4 Software requirements

Since this demo is completely created in JavaScript, only a web browser is required. Some veterinary browsers do not support advanced HTML5 standards and so to be very sure, at least following versions of browsers are required:

a. Web Browser: Google Chrome (version >=7.0), Firefox (version >= 6.0)
b. Mobile Browser: Chrome for android (version >= 3.0), Firefox (version >= 6.0), Safari Mobile (version >= 4.2)

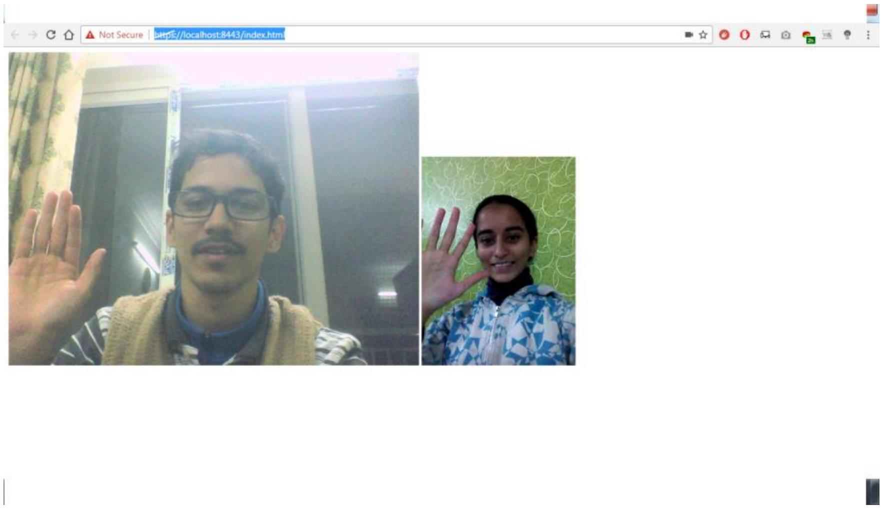
2. HEADSTART ON APIs AND LIBRARIES USED

Short side demonstrations given below help to provide basic understanding of major APIs used in the development of virtual tour application.

2.1 Video chat application using WebRTC

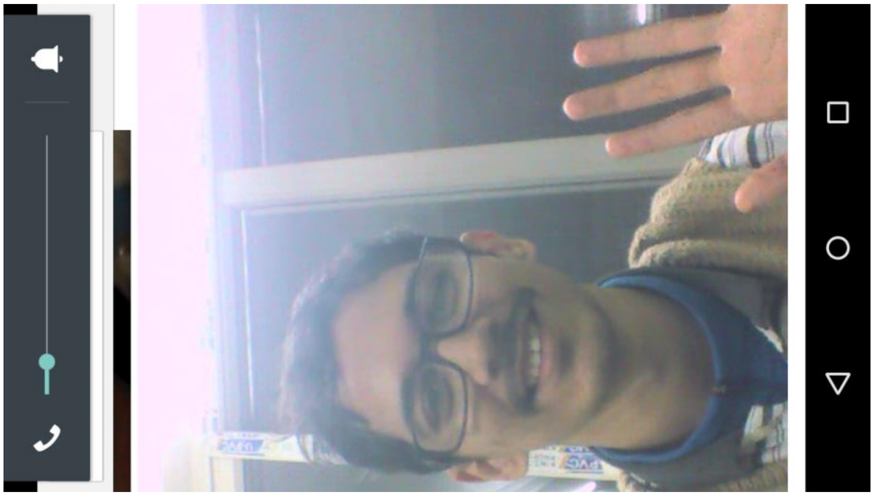
A sample video chat application was created using WebRTC, as a beginner's step to explore video streaming and data transfer capabilities of WebRTC. Below are few snapshots captured during video chat:

a. Snap taken on desktop



Video on the left is user's own stream while the one on the right is remote user's stream

b. Snap taken on android phone:



2.2 2D to 3D video render using Three.js

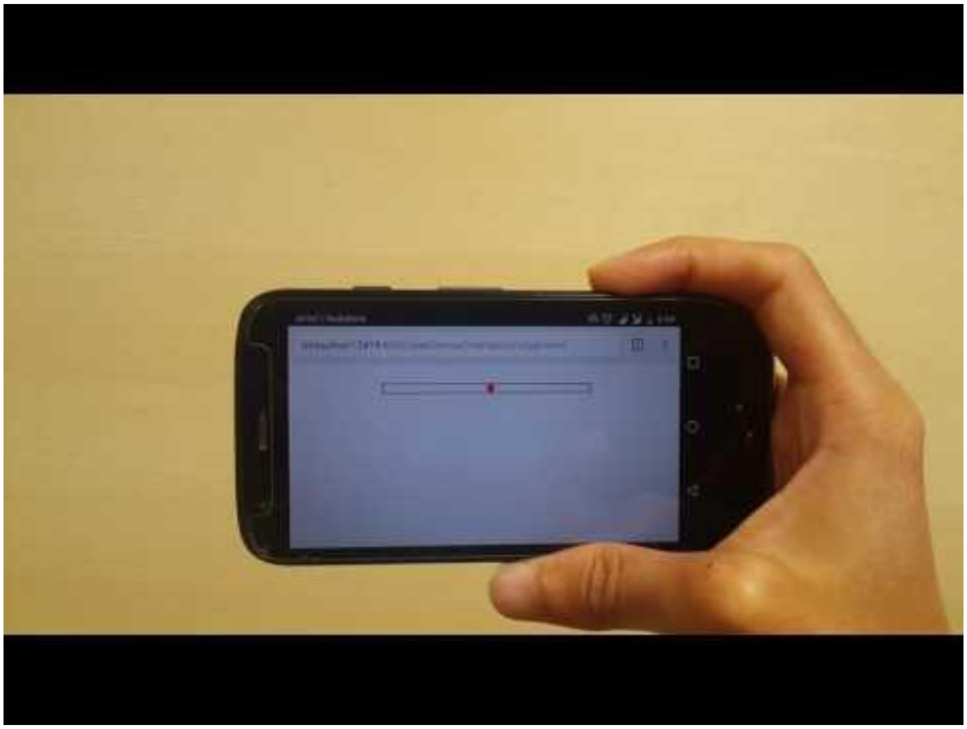
Recent advancements in technology has made it easy to record a 360 degree video of a scene using 360 degree or stereoscopic cameras. This has certainly replaced three or four camera assembly used traditionally for video recording. That said, such devices are not readily available in the market. In the virtual tour application, a video is recorded using a 2D video recording camera but the stream is converted to stereoscopic view for VR experience.

Three.js comes in very handy for this task. A beginner can start with conversion of local video before handling live stream as demonstrated in the video attached below:

2.3 Movement tracking using HTML5 deviceOrientation API

As mentioned in section above, scenic elements of a location would be captured using 2D rear camera unit available in a mobile device. This will not result in 360 degree view of the location. To simulate a 360 degree experience, the head movements of a user can be tracked and the recording device can be spanned in vertical and horizontal axes. This will record different planes in the scene and the end user will be tricked into a 360 degree view.

Head movement tracking in horizontal axis was achieved by using alpha angle values returned by deviceOrientation API. This can be mapped to one dimensional view as demonstrated in the video attached below:



3. LET'S GET A VIRTUAL TOUR

Having talked about different modules in the section above, let's integrate them into one web application. This application will act as a proof of virtual tour concept talked earlier. To run the application, execute following steps:

a. Navigate to code base root folder. Execute command: npm install
b. To start https local server, execute command: node server.js
c. Server will start on localhost on port no: 8443. Open browser and type this URL to open the application:
https://localhost:8443/index.html

Demonstration of the concept can be seen in the video attached below:

4. FUTURE SCOPE

Virtual tour concept can be implemented in true sense after adding a servo motor and a microcontroller. When a camera is mounted on this motor with the help of a pivot, this motor will act to provide a directional sense. Movement of the motor, controlled by head movements of the remote user, will rotate the pivot and camera in horizontal and vertical axes and simulate a 360 degree experience. This gesture based motion will be controlled by the microcontroller unit to which the motor is attached.

Once done, the whole system will act like an ecosystem, every module working in unison and sharing live events across each other.