



# *A virtual tour to* Augmented and Virtual Reality

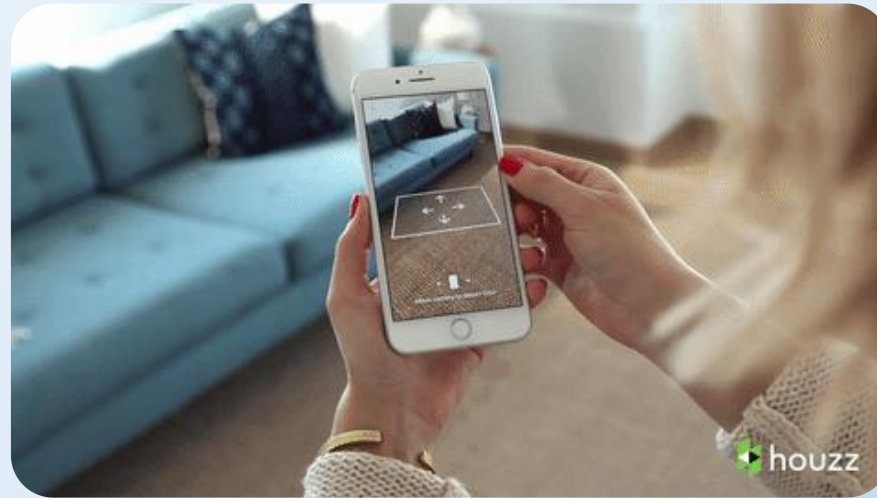
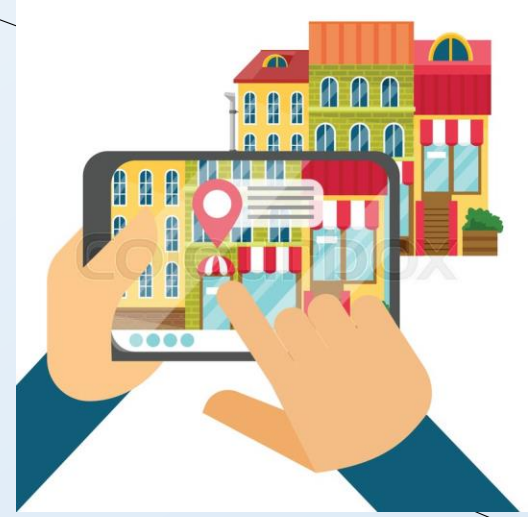
Dr. Sarwan Singh  
NIELIT Chandigarh





# Agenda

- Tools - three.js, 3D Models,
- Visual Inspector and Dev Tools
- Virtual Environment





# References

## Websites :

- [developers.google.com/ar](https://developers.google.com/ar/),
- [dev.to/arunkumarvallal](https://dev.to/arunkumarvallal), [mobidev.biz](https://mobidev.biz),  
[gerardfriel.com/ar/the-history-of-ar](https://gerardfriel.com/ar/the-history-of-ar)
- Harvard Business Review – “*Managers-Guide-to-AR*”
- “Virtual Reality/Augmented Reality White Paper”  
CAICT, Huawei Technologies Co.

## Books

- “Theory and applications of marker-based augmented reality” – Sanni Siltanen
- “Computer graphics” - Hearn and Baker





# Three.js

- Three.js is a cross-browser JavaScript library and Application Programming Interface (API) used to create and display animated 3D computer graphics in a web browser.
- Three.js uses WebGL.
- The source code is hosted in a repository on GitHub.

**WebGL** (Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins



# History – three.js

- Three.js was first released by Ricardo Cabello to GitHub in April 2010.
- The origins of the library can be traced back to his involvement with the demoscene in the early 2000s.
- Javascript based WebGL was main contributing factor for publishing lot of content based on three.js on github
- Three.js runs in all browsers supported by WebGL 1.0.
- Three.js is made available under the MIT license





- Effects: Anaglyph, cross-eyed and parallax barrier.
- Scenes: add and remove objects at run-time; fog
- Cameras: perspective and orthographic; controllers: trackball, FPS, path and more
- Animation: armatures, forward kinematics, inverse kinematics, morph and keyframe
- Lights: ambient, direction, point and spot lights; shadows: cast and receive
- Materials: Lambert, Phong, smooth shading, textures and more
- Shaders: access to full OpenGL Shading Language (GLSL) capabilities: lens flare, depth pass and extensive post-processing library
- Objects: meshes, particles, sprites, lines, ribbons, bones and more - all with Level of detail



# Feature of three.js

- Geometry: plane, cube, sphere, torus, 3D text and more; modifiers: lathe, extrude and tube
- Data loaders: binary, image, JSON and scene
- Utilities: full set of time and 3D math functions including frustum, matrix, quaternion, UVs and more
- Export and import: utilities to create Three.js-compatible JSON files from within: Blender, openCTM, FBX, Max, and OBJ
- Support: API documentation is under construction, public forum and wiki in full operation
- Examples: Over 150 files of coding examples plus fonts, models, textures, sounds and other support files
- Debugging: Stats.js, WebGL Inspector,[10] Three.js Inspector
- Virtual reality: accessing WebVR



# WebVR

- WebVR is a JavaScript API for creating immersive 3D, virtual reality experiences in your browser. Or simply put, allows VR in the browser over the Web.
- A-Frame uses the WebVR API to gain access to VR headset sensor data (position, orientation) to transform the camera and to render content directly to VR headsets. Note that WebVR, which provides data, should not be confused nor conflated with WebGL, which provides graphics and rendering.





# A-Frame Supports

- VR on desktop with a headset
- VR on mobile with a headset
- VR on standalone headset
- Flat on desktop (i.e., mouse and keyboard)
- Flat mobile (i.e., magic window)
- AR headsets (e.g., Magic Leap, HoloLens)
- mobile (i.e., magic window, ARKit, ARCore)

- A-Frame supports VR for any browser that implements the [WebVR specification](#), and flat 3D for most browsers.
- Firefox, chrome, oculus, Microsoft edge, Samsung internet

## VR Headset support

- HTC Vive
- Oculus Rift
- Oculus Quest
- Oculus Go
- Google Daydream
- Samsung GearVR
- Vive Focus ....



# Building UIs

- <https://aframe-building-ui.glitch.me/>