

Mixed Reality for Javascript Devs

Ron Dagdag

SpaceE

@rondagdag



Award Categories

AI, Windows Development,
Internet of Things, Mixed Reality

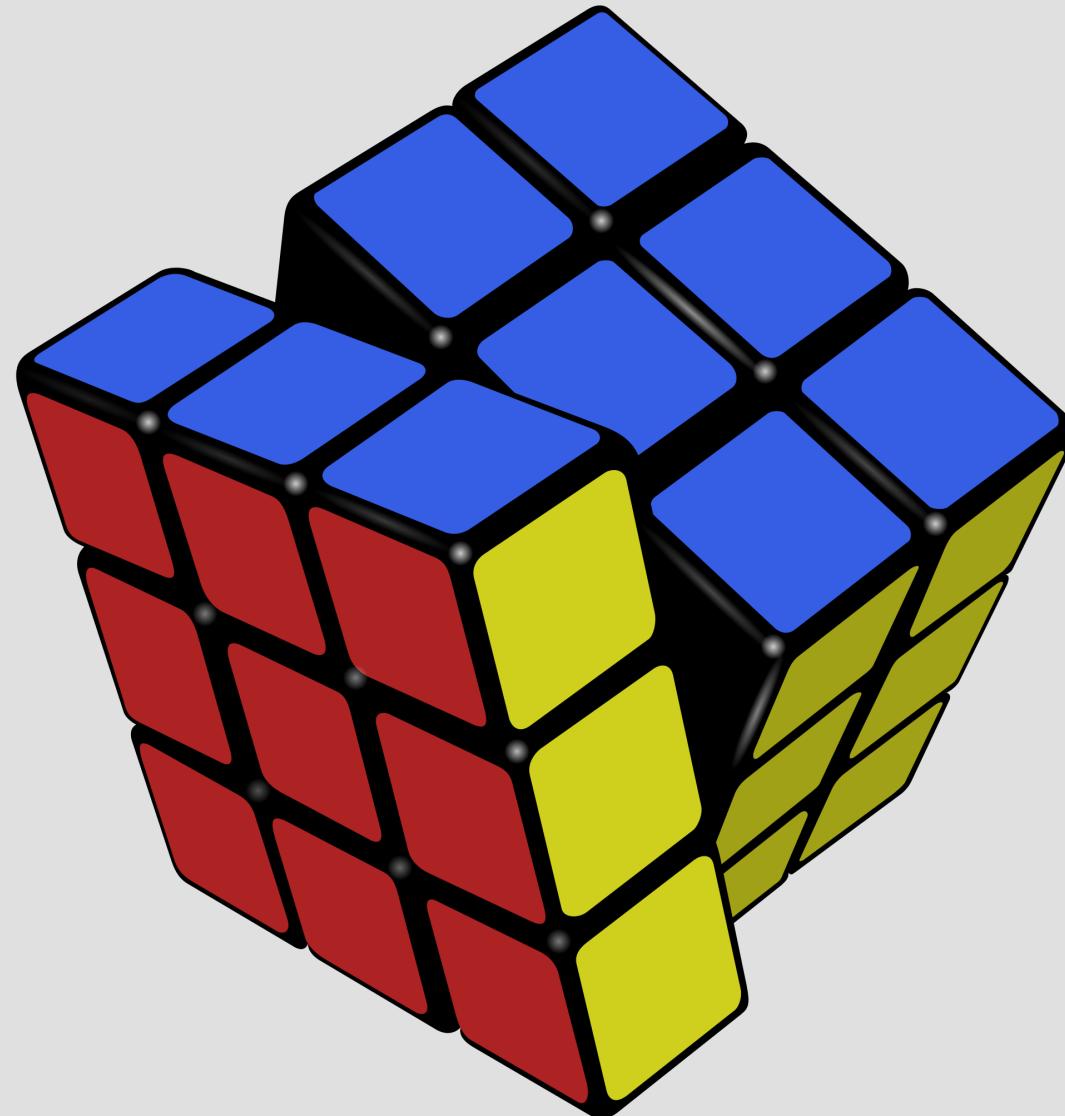
First year awarded:
2017

Number of MVP Awards:
6



Mixed Reality Devs
Virtual Reality
Augmented Reality
Javascript Devs

It was
originally
called a
'Magic Cube'

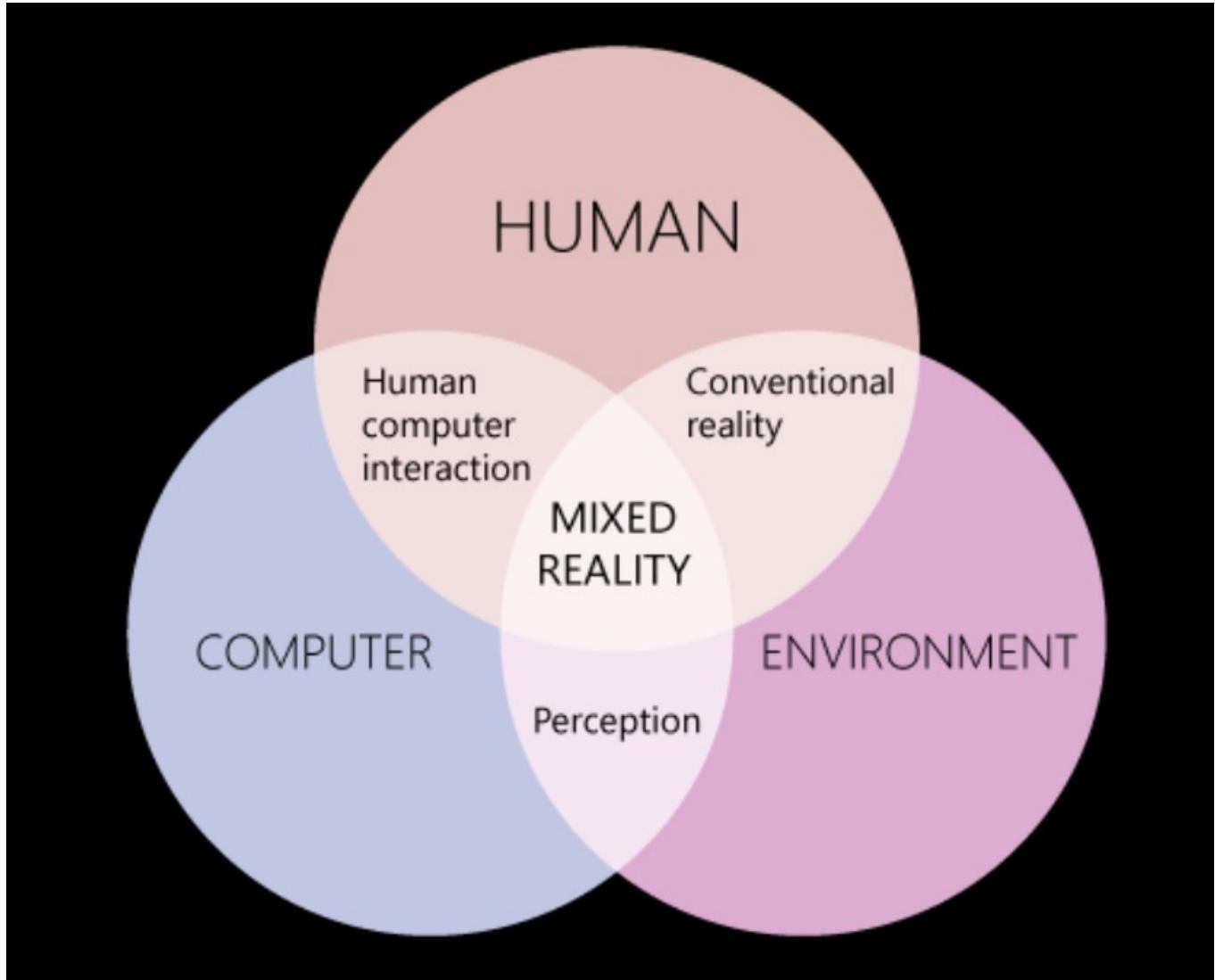


Agenda

- What is Mixed Reality?
- What is WebXR?
- Types of Virtual Reality Experiences
- Types of Augmented Reality Experiences
- What are open-source JS libraries available?

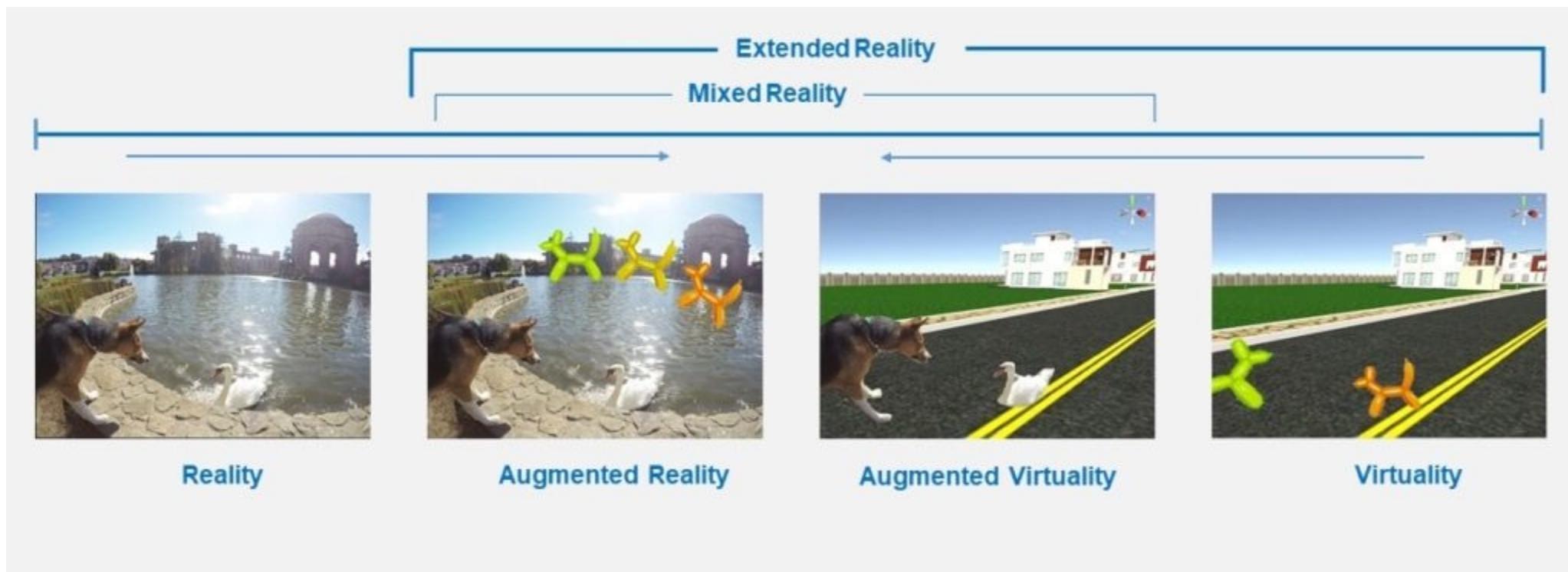
Mixed Reality

- blend physical and digital worlds
- unlocking natural and intuitive 3D
 - Human
 - Computer
 - environmental interaction



Mixed Reality

- advancements in computer vision, graphical processing, display technologies, input systems, and cloud computing
 - *Environmental understanding*: spatial mapping and anchors.
 - *Human understanding*: hand-tracking, eye-tracking, and speech input.
 - *Spatial sound*.
 - *Locations and positioning* in both physical and virtual spaces.
 - Collaboration on *3D assets* in mixed reality spaces.



<https://xr4all.eu/xr/>

Types of Virtual Reality experiences

Non-Immersive VR

- see the real world and virtual objects at the same time
- Video games

Semi-Immersive VR

- see the virtual world, some visual connection to the real world
- 360 Virtual tour

Fully-Immersive VR

- completely immerses the user in a virtual world
- blocks out the real world

Social VR (Collaborative)

- multiple users to interact with each other in a virtual environment



Types of Augmented Reality experiences

Marker-based AR

- uses a marker, (QR code or image) to trigger display of virtual content

Markerless AR

- uses CV to track the real-world environment and display virtual content without marker

Location-based AR

- uses GPS or CV technologies to display virtual content specific to a location

Projection-based AR

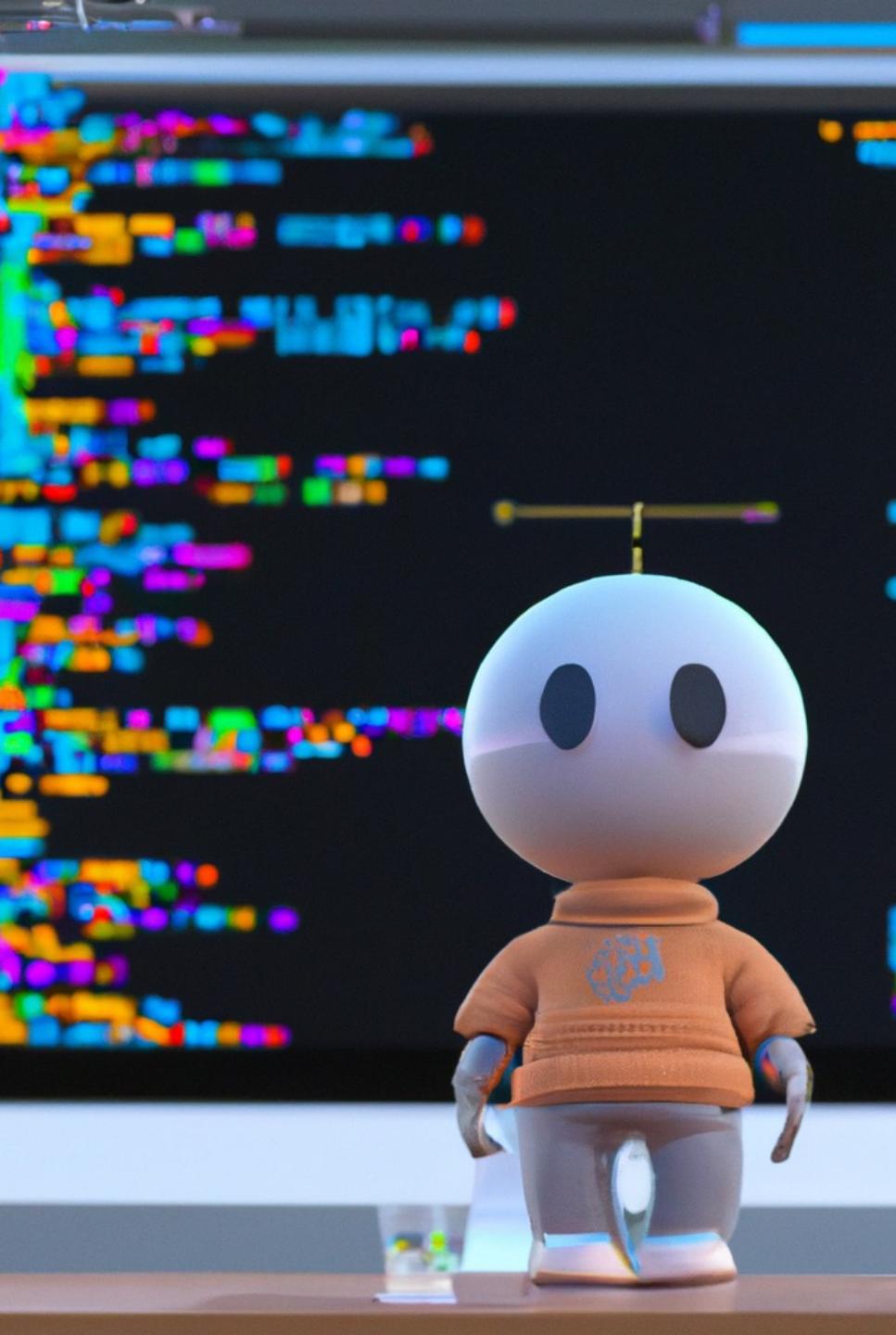
- uses projector to display virtual content onto a real-world surface

Superimposition-based AR

- replace or overlay virtual objects in real world,
- provide real-time information



There are 43
quintillion
possible
combinations



What is WebXR?

- unified API for VR and AR devices
- VR headsets, AR glasses, and smartphones w/ browsers
- web platform and ecosystem
- <https://immersiveweb.dev/>
- iOS support - still experimental

Advantage of WebXR

Open Platform

- Permissionless publishing

Sharable

- Send a link

Accessible

- Low barrier of entry

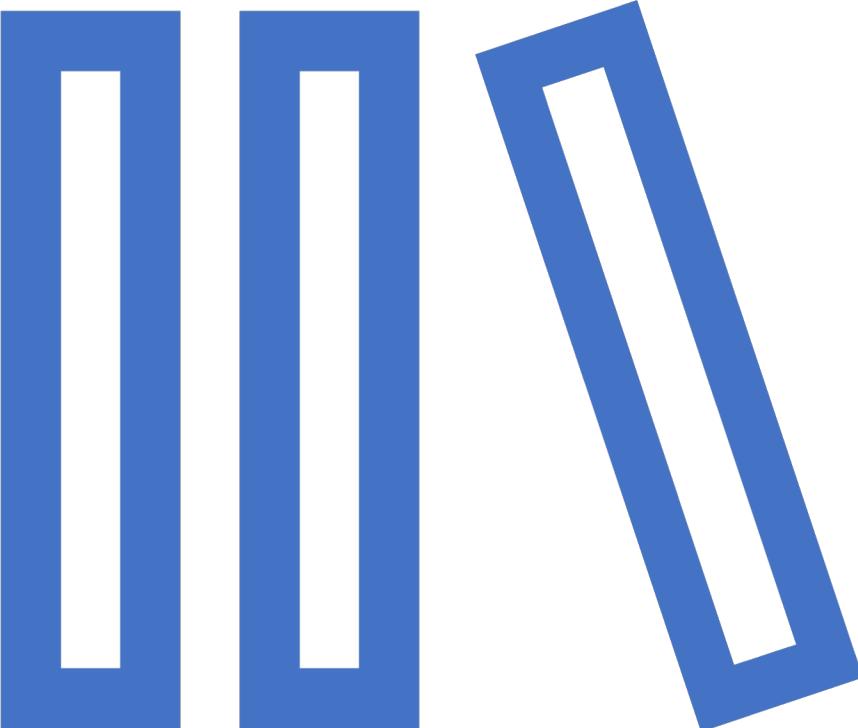
Cross Platform

- “Just works”

The world record is...

Yusheng Du from China in 2018 who scored an incredible 3.475 seconds





Javascript Libraries Available

- ModelViewer
- A-Frame
- MindAR / Pictarize
- BabylonJS
- Three.JS
- MediaPipe
- ReactXR
- Wonderland Engine
- PlayCanvas
- Snap Lens Studio

ModelViewer (modelviewer.dev)



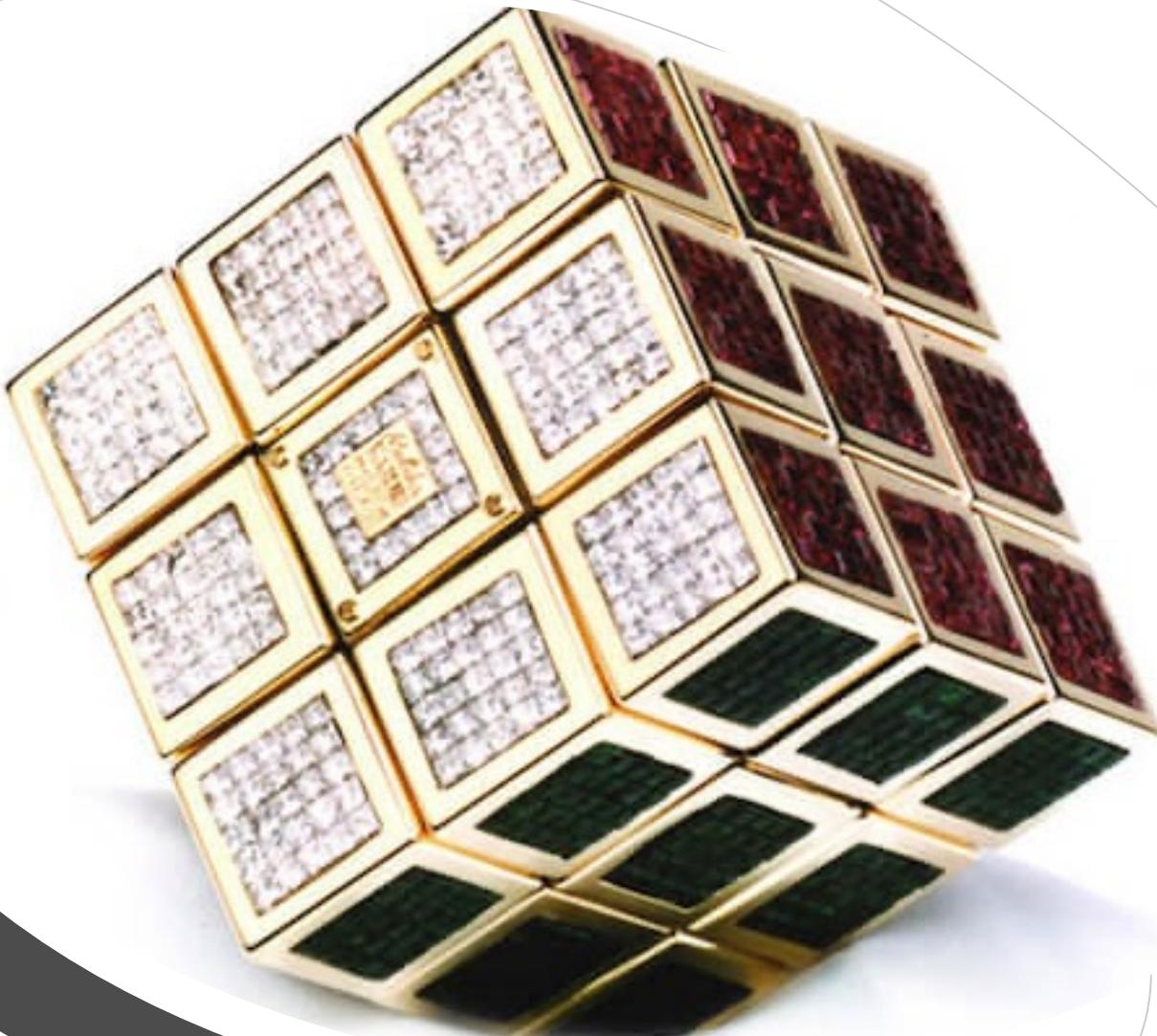
- a custom HTML element
- displaying 3D models and viewing in AR
- <https://modelviewer.dev/editor/>

```
<script type="module" src="https://unpkg.com/@google/model-viewer/dist/model-viewer.js"></script>
<script nomodule src="https://unpkg.com/@google/model-viewer/dist/model-viewer-legacy.js"></script>
```

<!--Use it like any other HTML element-- >

```
<model-viewer src="examples/assets/Astronaut.glb" ar alt="A 3D model of an astronaut" auto-rotate  
camera-controls background-color="#455A64"></model-viewer>
```





@rondagdag

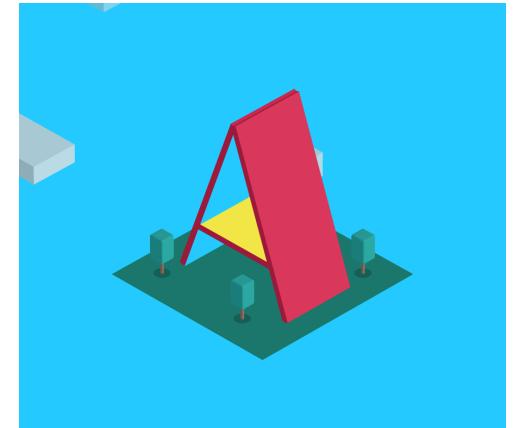
The Masterpiece Cube costs \$2.5 million

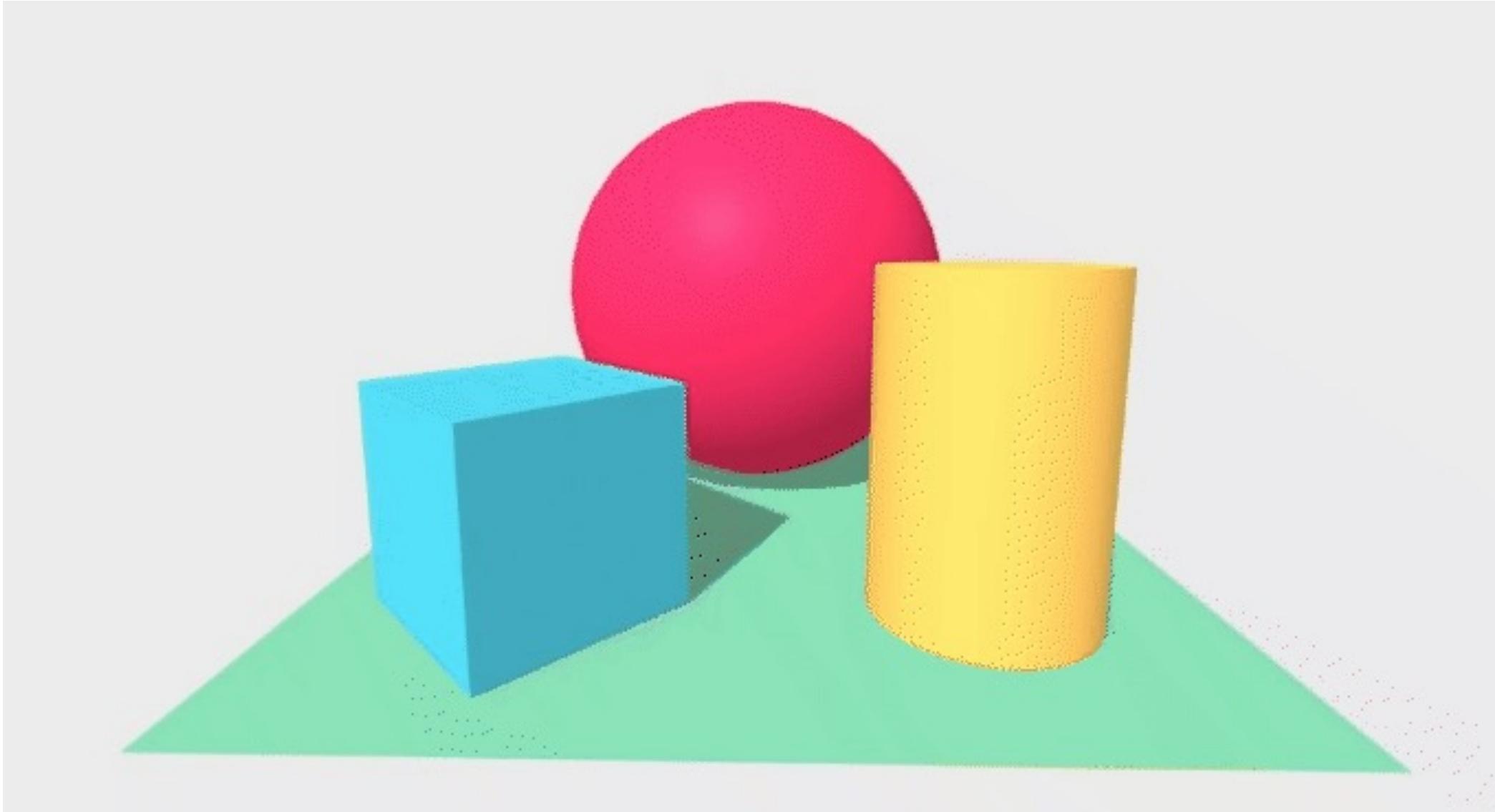
created in 1995 by Diamond Cutters International to commemorate the 15th anniversary

A-Frame (aframe.io)

- web framework for building 3D/AR/VR experiences using a combination of HTML and Javascript.
- <https://aframe.io/examples/showcase/helloworld/>

```
<html>
  <head>
    <script src="https://aframe.io/releases/1.2.0/aframe.min.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box position="-1 0.5 -3" rotation="0 45 0" color="#4CC3D9"></a-box>
      <a-sphere position="0 1.25 -5" radius="1.25" color="#EF2D5E"></a-sphere>
      <a-cylinder position="1 0.75 -3" radius="0.5" height="1.5" color="#FFC65D"></a-cylinder>
      <a-plane position="0 0 -4" rotation="-90 0 0" width="4" height="4" color="#7BC8A4"></a-plane>
      <a-sky color="#ECECEC"></a-sky>
    </a-scene>
  </body>
</html>
```





<ctrl> + <alt> + i on any **A-Frame** scene to open up the Inspector

Entity Component System

Entity

- general-purpose object -> positioned and transformed in a scene.

Component

- behavior or functionality that can be attached to an Entity.
Reusable Modules.

System

- global scope, management, and services for classes of components.
- systems handle the logic, components act as data containers

Box = Position + Geometry + Material

Light Bulb = Position + Light + Geometry + Material + Shadow

VR Controller = Position + Rotation + Input + Model + Grab + Gestures

Ball = Position + Velocity + Physics + Geometry + Material

Player = Position + Camera + Input + Avatar + Identity

Someone solved a Rubik's Cube while skydiving!

- Dan Knight jumped out of a plane and managed to solve the puzzle in the 30 seconds it took him to fall to the ground



<https://www.youtube.com/watch?v=dtRsKWAECb8>

@rondagdag

MindAR

- web augmented reality library.
- supports Image Tracking and Face Tracking.
- <https://hiukim.github.io/mind-ar-js-doc/>



<https://hiukim.github.io/mind-ar-js-doc/tools/compile>

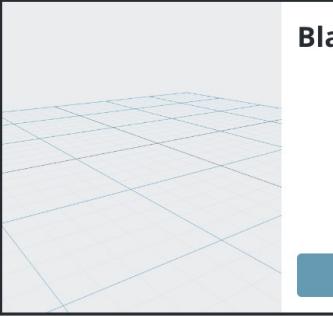
```
<html>
  <head>
    <script src="https://aframe.io/releases/1.3.0/aframe.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/mind-ar@1.2.0/dist/mindar-image-aframe.prod.js"></script>
  </head>
  <body>
    <a-scene mindar-image="imageTargetSrc: ./rubiks.mind;" color-space="sRGB" renderer="colorManagement: true,
physicallyCorrectLights" vr-mode-ui="enabled: false" device-orientation-permission-ui="enabled: false">
      <a-assets>
        <a-asset-item id="avatarModel" src="../rubiks_cube.glb"></a-asset-item>
      </a-assets>
      <a-camera position="0 0 0" look-controls="enabled: false"></a-camera>
      <a-entity mindar-image-target="targetIndex: 0">
        <a-gltf-model rotation="0 0 0 " position="0 0 0.1" scale="5 5 5" src="#avatarModel" animation="property: position;
to: 0 0.1 0.1; dur: 1000; easing: easeInOutQuad; loop: true; dir: alternate">
      </a-entity>
    </a-scene>
  </body>
</html>
```

Pictarize Studio

web based drag-n-drop editor, building and publishing interactive web AR apps with 3D models, videos, audios and texts!

Projects Create Project...

Create a blank project or start from a template X



Blank Project

Create



Slideshow

- multiple videos
- carousel effect (script)

Clone



Flash Cards

- multiple targets
- background music
- audio on click (script)

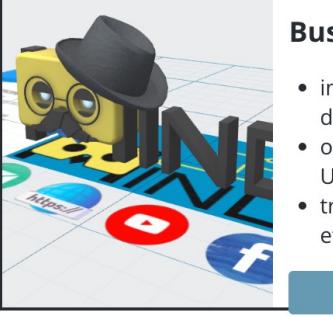
Clone



Performance

- real persons
- green screen background removal

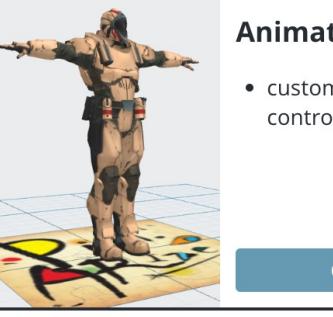
Clone



Business Card

- information display
- open external URLs
- transition effects (script)

Clone



Animated Models

- custom animations control (script)

Clone

<https://pictarize.com/>

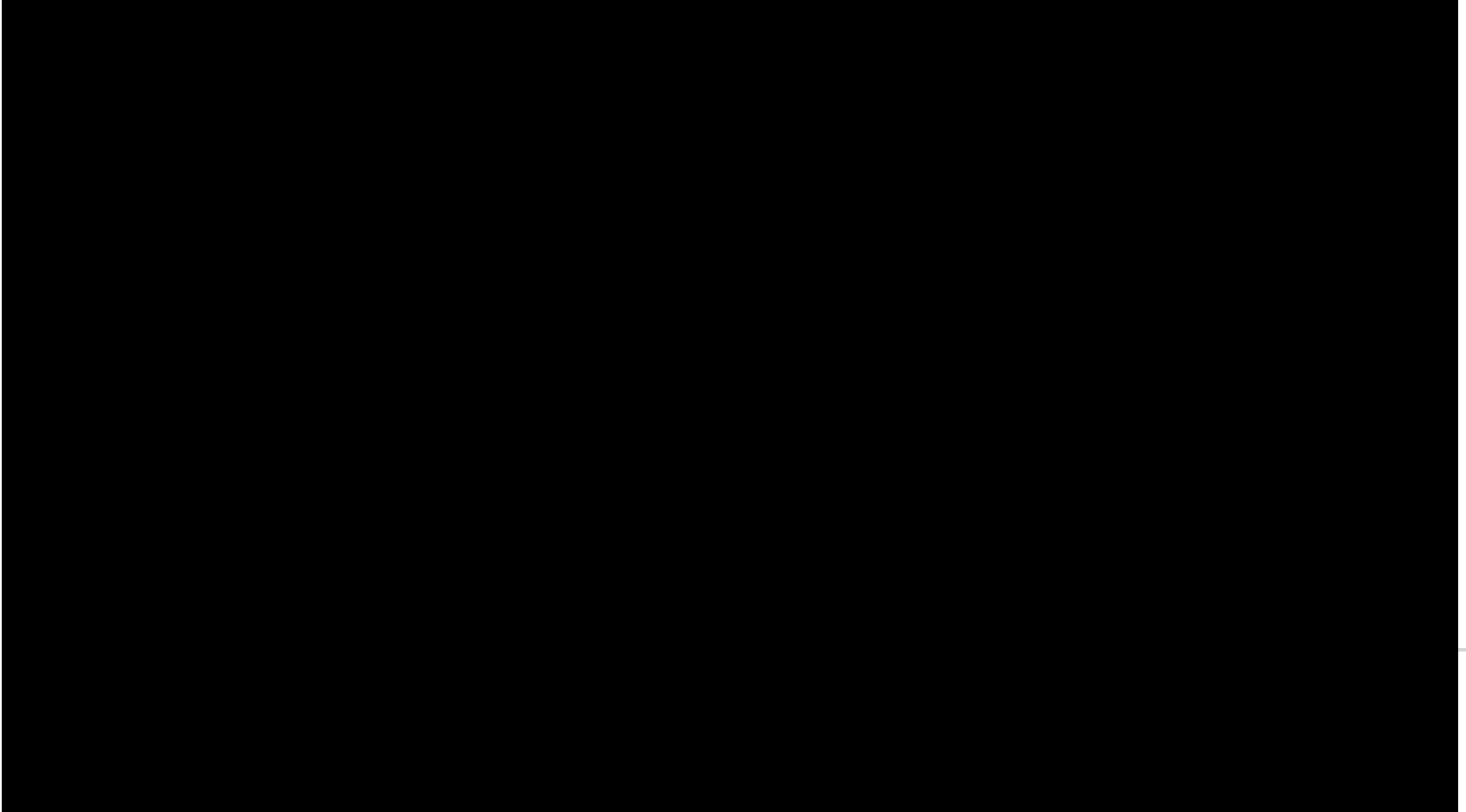
@rondagdag

Pictarize Studio

<https://pictarize.com/>



@rondagdag



Largest Rubik's Cube

<https://youtu.be/SkwIRTX2ecA?t=35>

BabylonJS

- real-time 3D game engine built using TypeScript
- full WebXR support out of the box, including gaze and teleportation support, AR experimental features
- open-source library to create 3D experiences, animations, and games in the browser
- fast, efficient, and flexible



BabylonJS

- tools for creating interactive 3D scenes
- create 3D models, manipulate camera movements, lighting/shadows/animations.
- physics simulation, collision detection
- Supported Devices
 - Android Phones
 - Hololens 2
 - Quest

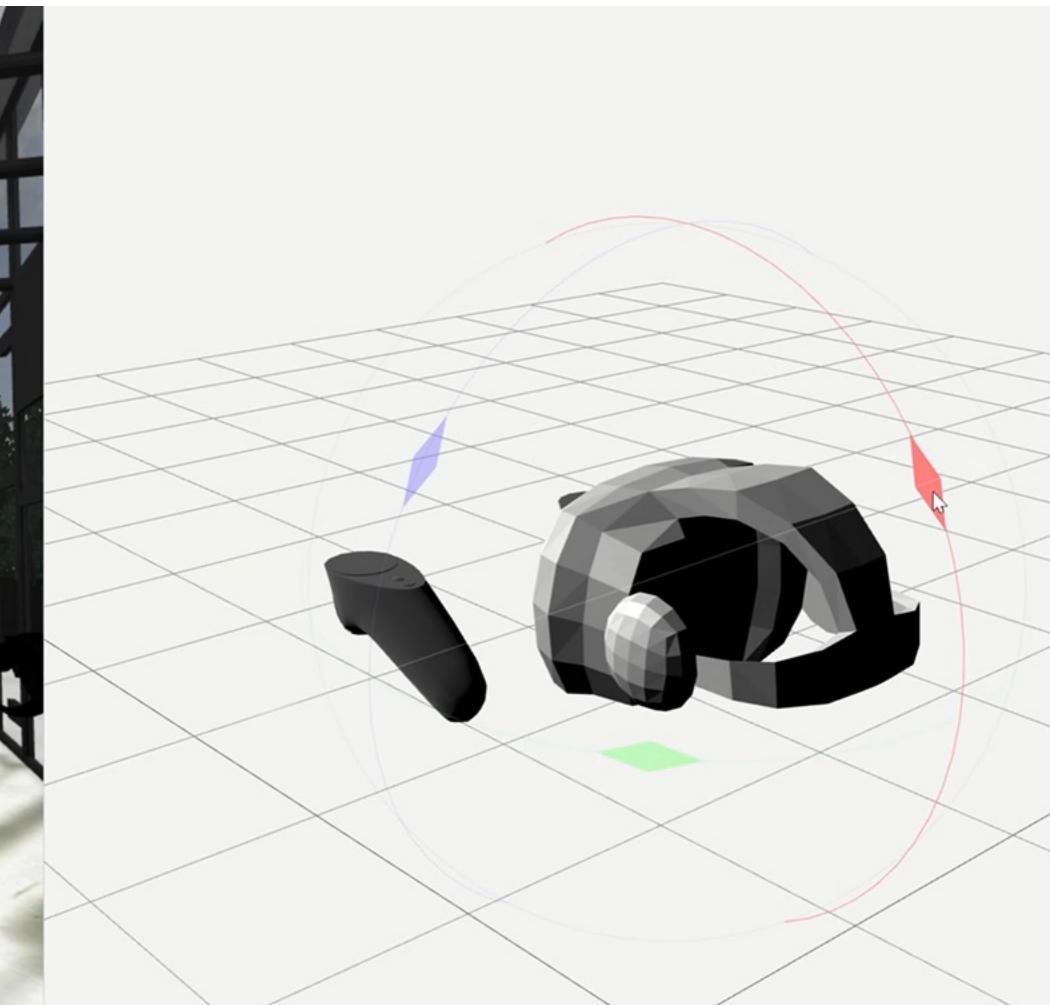
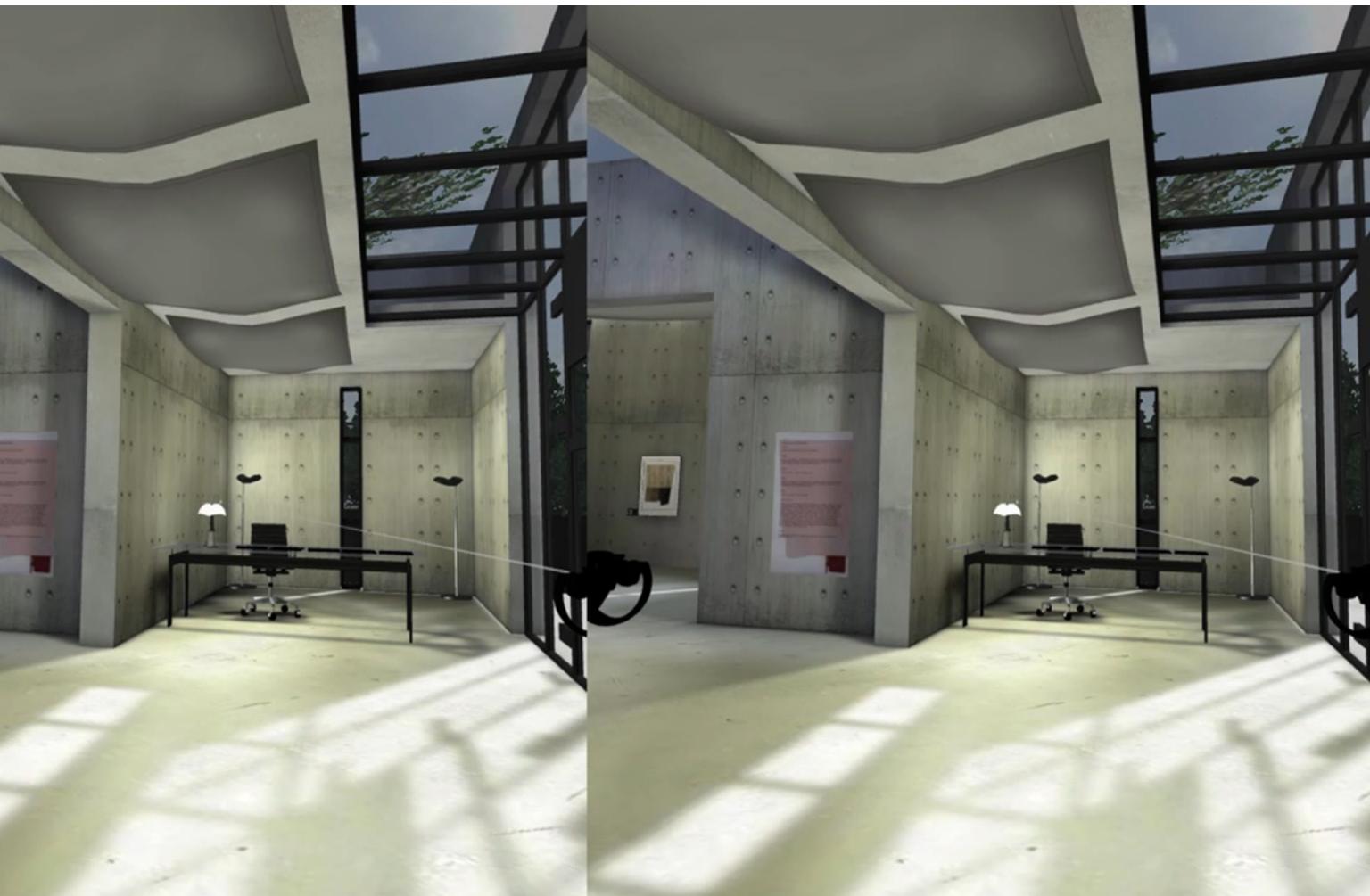


```
var createScene = function () {
    // Playground needs to return at least an empty scene and default camera
    var scene = new BABYLON.Scene(engine);
    var camera = new BABYLON.FreeCamera("camera1", new BABYLON.Vector3(0, 5, -10), scene);

    // Async call
    BABYLON.SceneLoader.Append("https://www.babylonjs.com/Scenes/Espilit/",
        "Espilit.babylon", scene, async function () {
            var xr = await scene.createDefaultXRExperienceAsync(
                {floorMeshes: [scene.getMeshByName("Sols")]}));
        });

    return scene;
};
```

<https://playground.babylonjs.com/#JA1ND3#164>

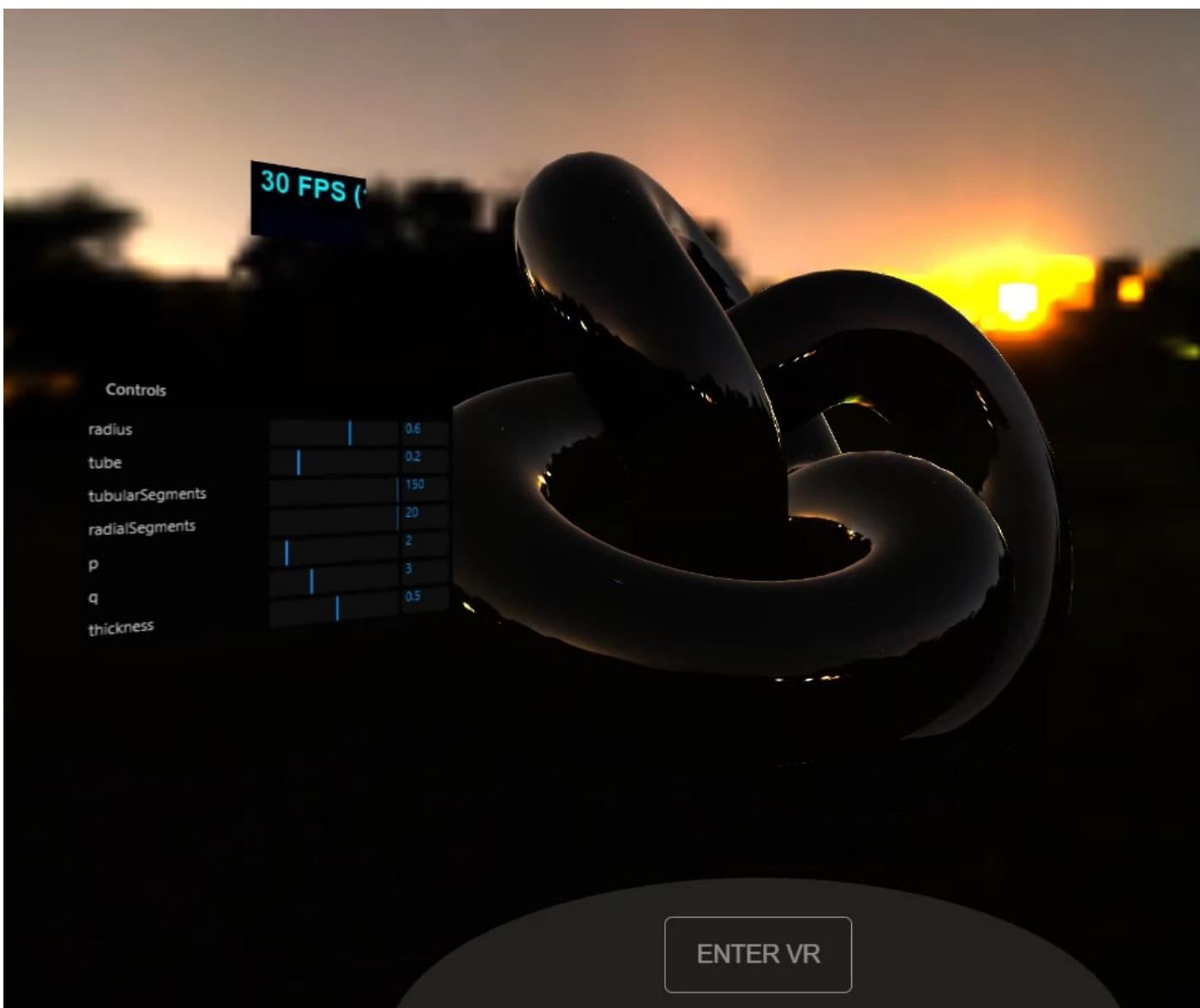


@rondagdag

three.js

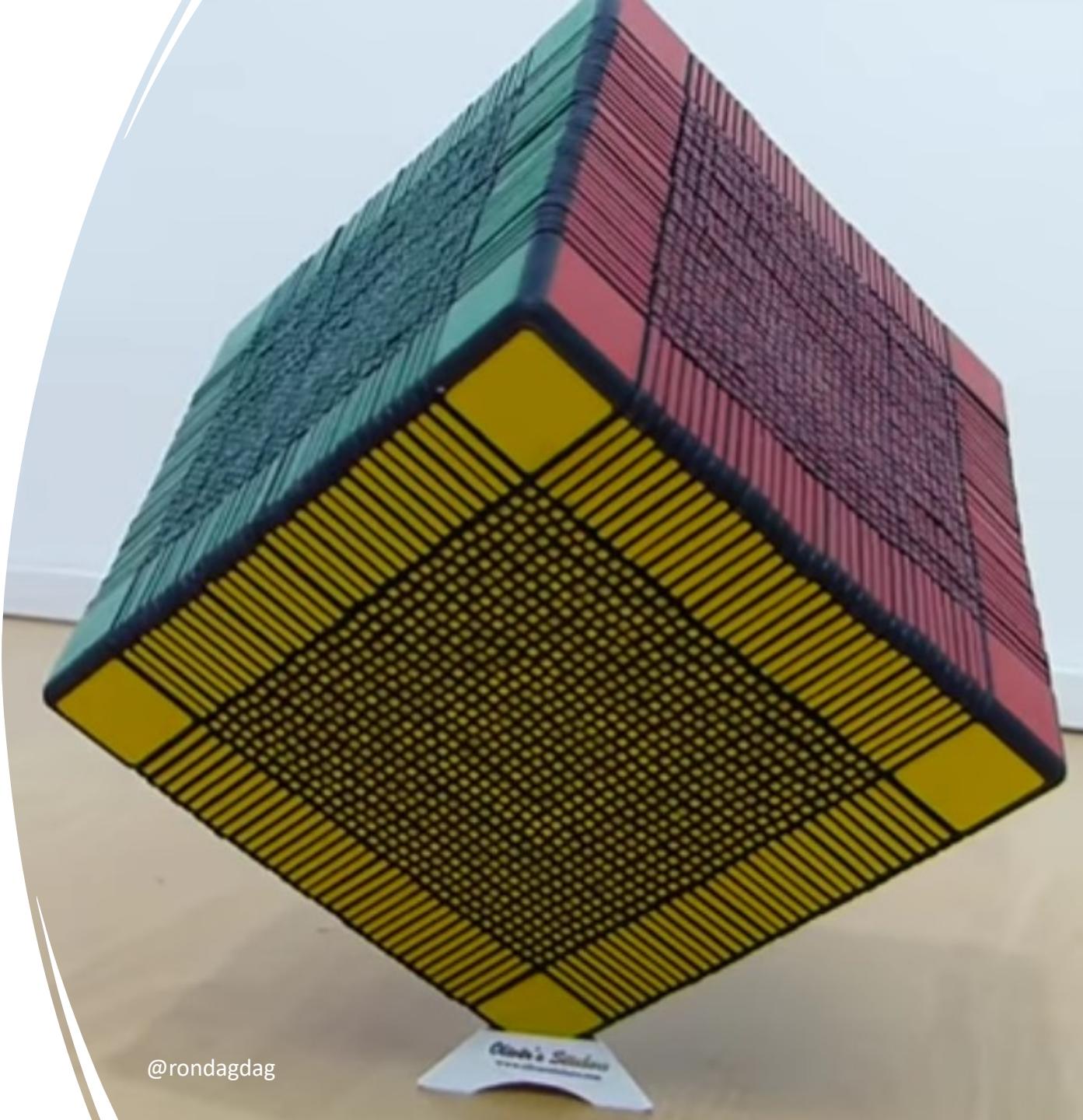
- Open source, cross-browser JavaScript library
- Large community, good docs, and many examples.
- create and display animated 3D computer graphics in a web browser
- <https://threejs.org/examples/?q=webxr>





https://raw.githubusercontent.com/mrdoob/three.js/master/examples/webxr_vr_sandbox.html

WORLD RECORD $33 \times 33 \times 33$ RUBIK's CUBE

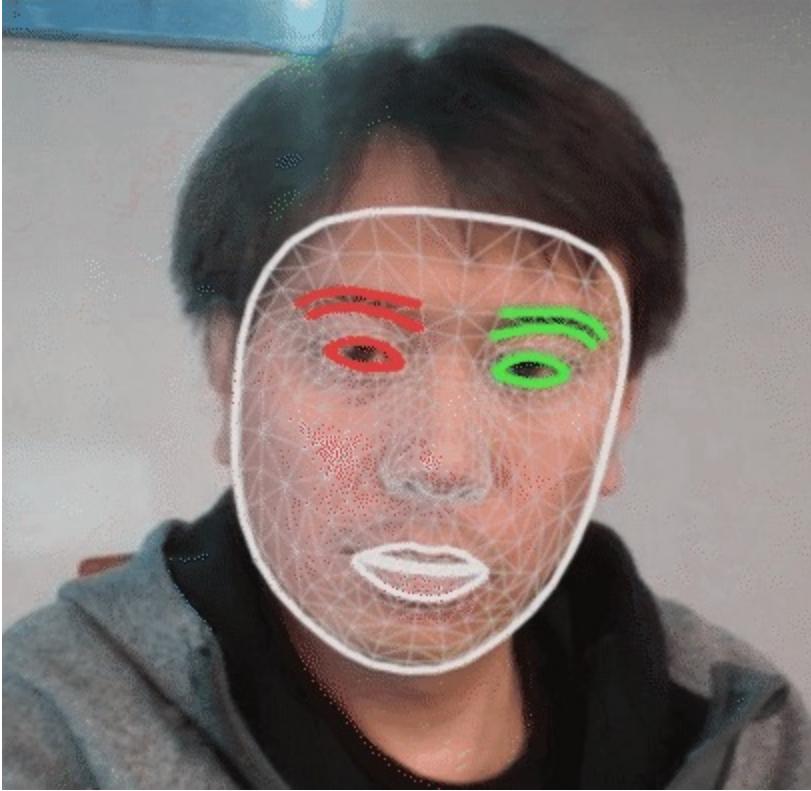


@rondagdag

Mediapipe

open-source framework, cross-platform, multi-device apps - computer vision and media processing

| Solution | NPM Package | Example |
|--|---|---|
| <u>Face Mesh</u> | <u>@mediapipe/face_mesh</u> | <u>mediapipe.dev/demo/face_mesh</u> |
| <u>Face Detection</u> | <u>@mediapipe/face_detection</u> | <u>mediapipe.dev/demo/face_detection</u> |
| <u>Hands</u> | <u>@mediapipe/hands</u> | <u>mediapipe.dev/demo/hands</u> |
| <u>Holistic</u> | <u>@mediapipe/holistic</u> | <u>mediapipe.dev/demo/holistic</u> |
| <u>Objectron</u> | <u>@mediapipe/objectron</u> | <u>mediapipe.dev/demo/objectron</u> |
| <u>Pose</u> | <u>@mediapipe/pose</u> | <u>mediapipe.dev/demo/pose</u> |
| <u>Selfie Segmentation</u> | <u>@mediapipe/selfie_segmentation</u> | <u>mediapipe.dev/demo/selfie_segmentation</u> |



Snap Lens Studio



- scripting engine for creating rich interactive experiences
- respond to
 - touch input
 - play animation and audio
 - modify Scene Objects.
- Helper Script - Behavior Script, Tween Manager, World Object Controller
- API Reference - <https://docs.snap.com/api/home#lens-studio>

Snap Lens Studio

```
// -----JS CODE-----  
  
//@input SceneObject targetObj;  
//@input string tweenName;  
  
var event = script.createEvent("MouthOpenedEvent");  
event.faceIndex = 0;  
event.bind(function (eventData) {  
    global.TweenManager.startTween(script.targetObj, script.tweenName);  
    print("Mouth was opened on face 0");  
});
```



Robots can
solve it in
under one
second.

https://youtu.be/cS2g_C6M7Bs?t=123

Summary

What is Mixed Reality?

- Blend physical and digital world

What is WebXR?

- Mixed Reality via web browsers

Types of Virtual Reality Experiences

- Non-immersive, Semi-immersive, Fully-immersive, Social

Types of Augmented Reality Experiences

- Marker-based AR, Markerless AR, Location-based AR, Projection-based AR, Superimposition-based AR

What are open-source JS libraries available?

- ModelViewer, A-Frame, MindAR / Pictarize, BabylonJS, Three.JS

About Me

Ron Dagdag



<https://linktr.ee/rondagdag>

