



Babylon.js Update

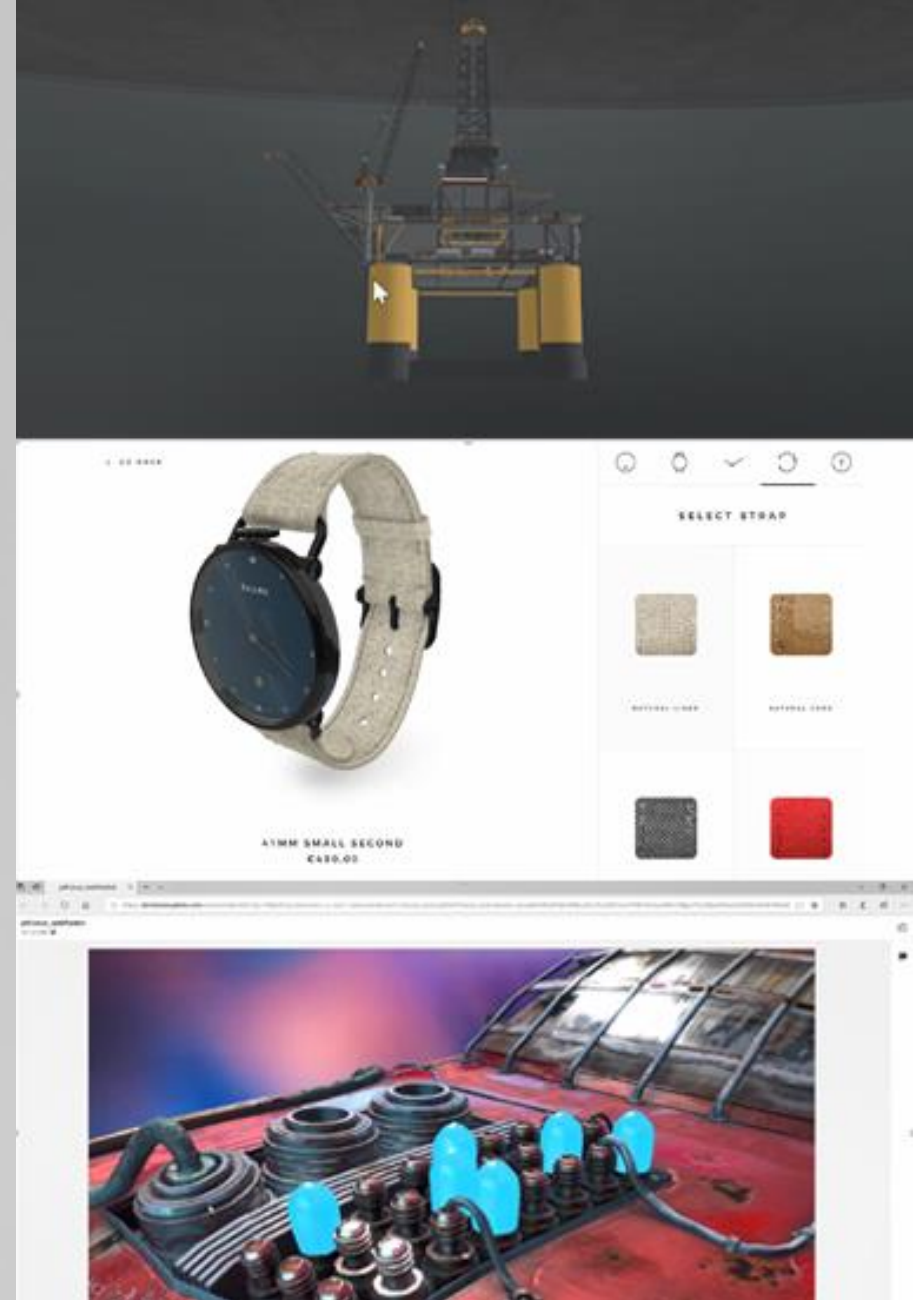
WebGL BOF at SIGGRAPH 2019

Gary Hsu
@bghgary

Babylon.js at a Glance

- 3D engine built on top of **WebGL** (1 and 2)
- **Open** source (Apache 2 license)
- **Community** driven
- Written in **TypeScript**
- Used by a lot of web apps
 - Office, SharePoint Spaces, OneDrive, Bing, Visio
 - Adobe, Samsung, Toyota, Sony Electronics, US Army, Ubisoft

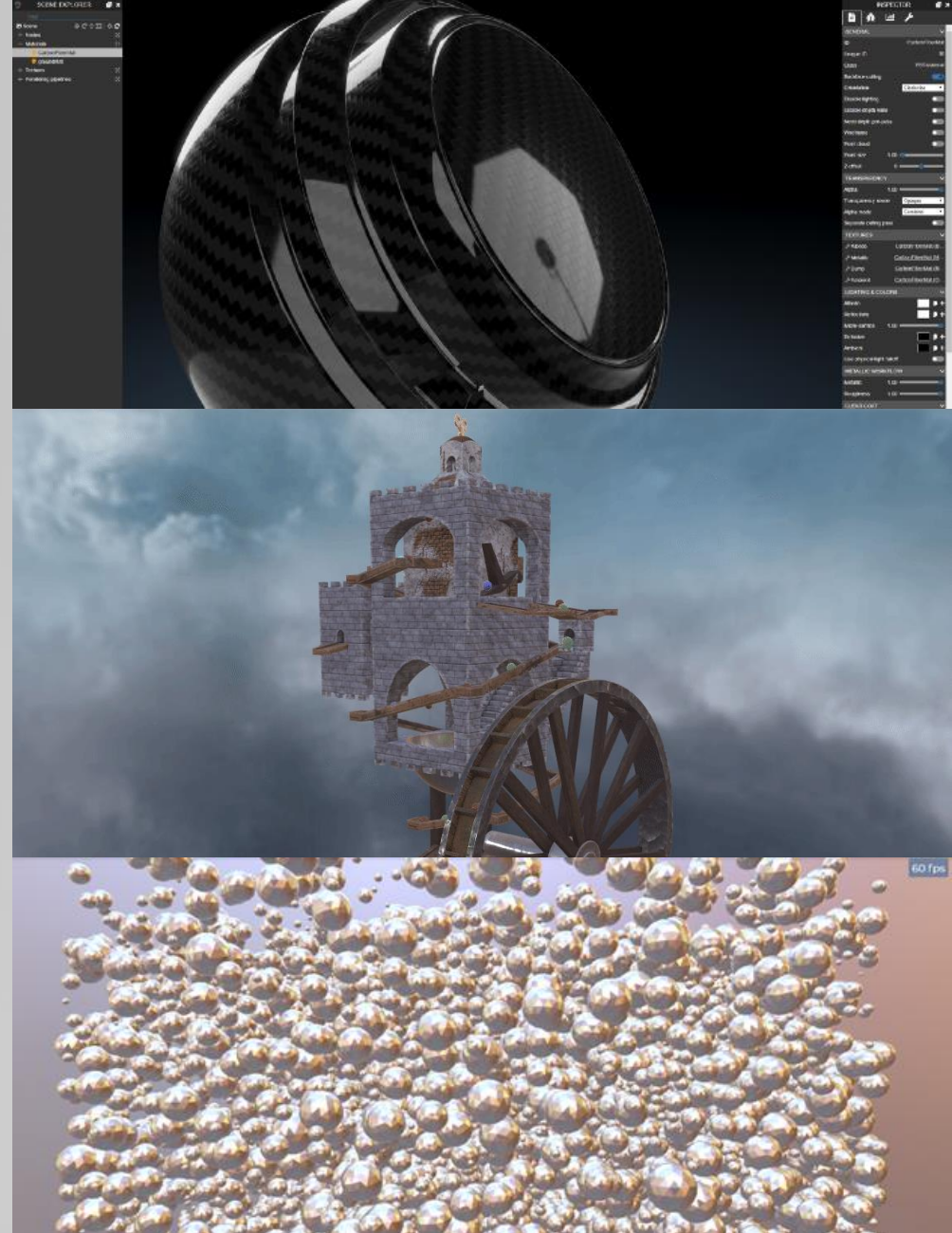
<https://www.babylonjs.com>



4.0 Released!

- New Inspector
- PBR Next
- ESNext Modules
- AmmoJS
- Optimizations

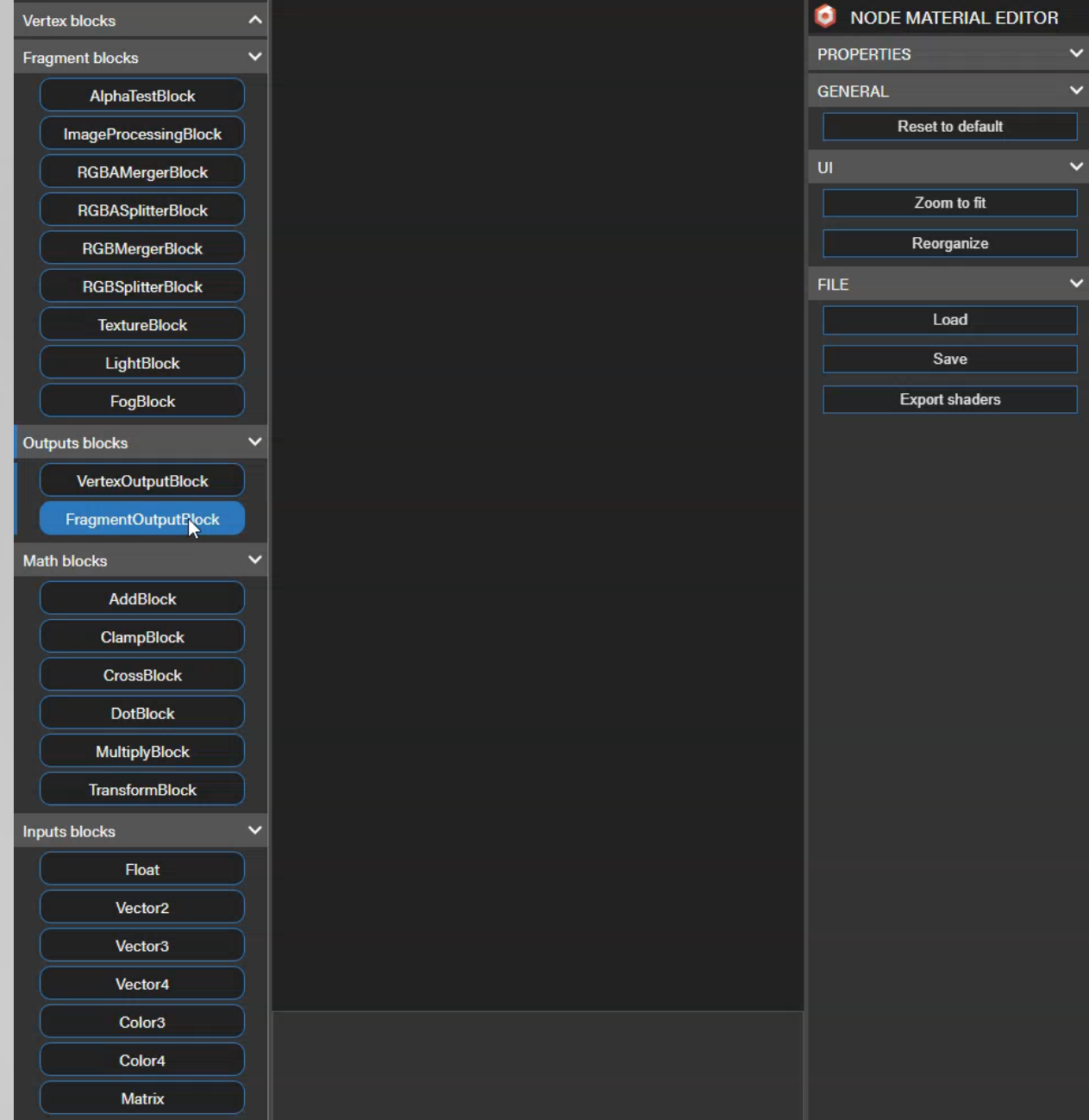
<https://doc.babylonjs.com/whats-new>



Coming Soon

- Node Material
- 2D Controls
- Navigation Mesh with Crowd Agents

<https://bit.ly/2YrKS4L>



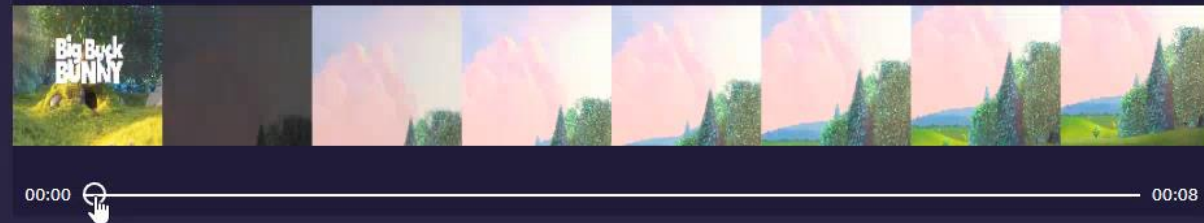
Coming Soon

- Node Material
- 2D Controls
- Navigation Mesh with Crowd Agents

<https://doc.babylonjs.com/features/timeline>

This demo represents a timeline rendered through WebGL to speed up thumbnails management.

Using the power of the GPU can dramatically improve the user experience whilst scrolling accross thousands of pictures.

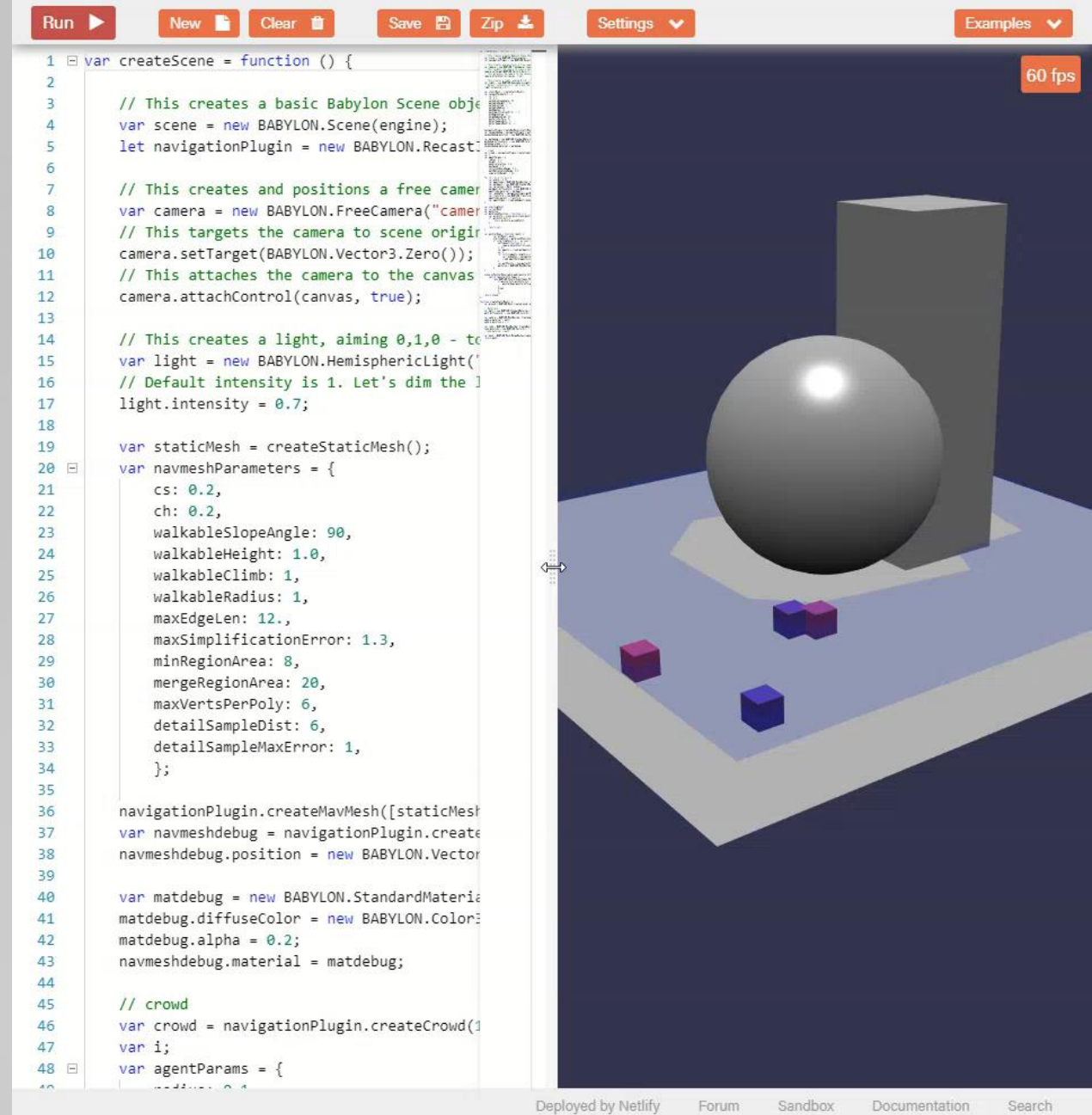


Coming Soon

- Node Material
- 2D Controls
- Navigation Mesh with Crowd Agents

<https://doc.babylonjs.com/extensions/navigation-mesh>

<https://doc.babylonjs.com/extensions/crowd>

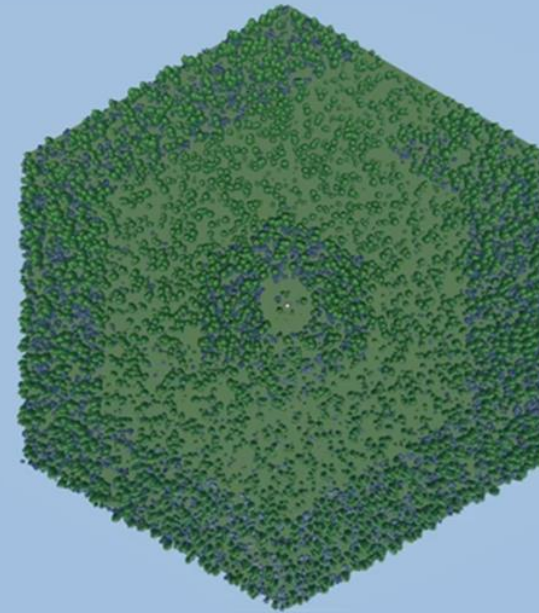


Web Standards

- WebGPU
- WebXR
- glTF with KTX2 + Basis Universal

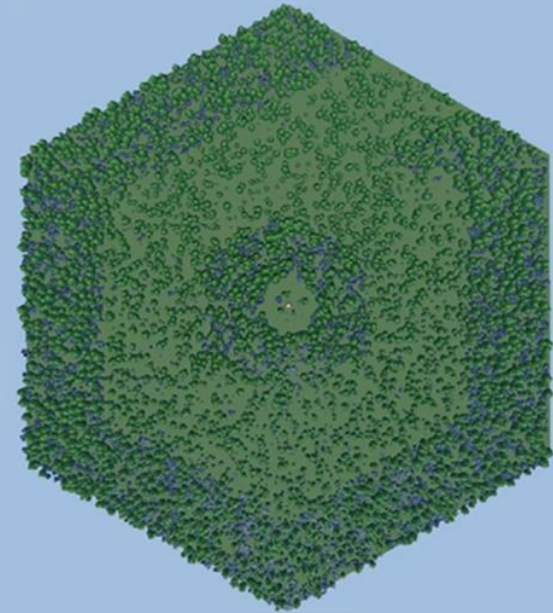
<https://doc.babylonjs.com/extensions/webgpu>

WebGL



12 fps - 82 CPU ms

WebGPU

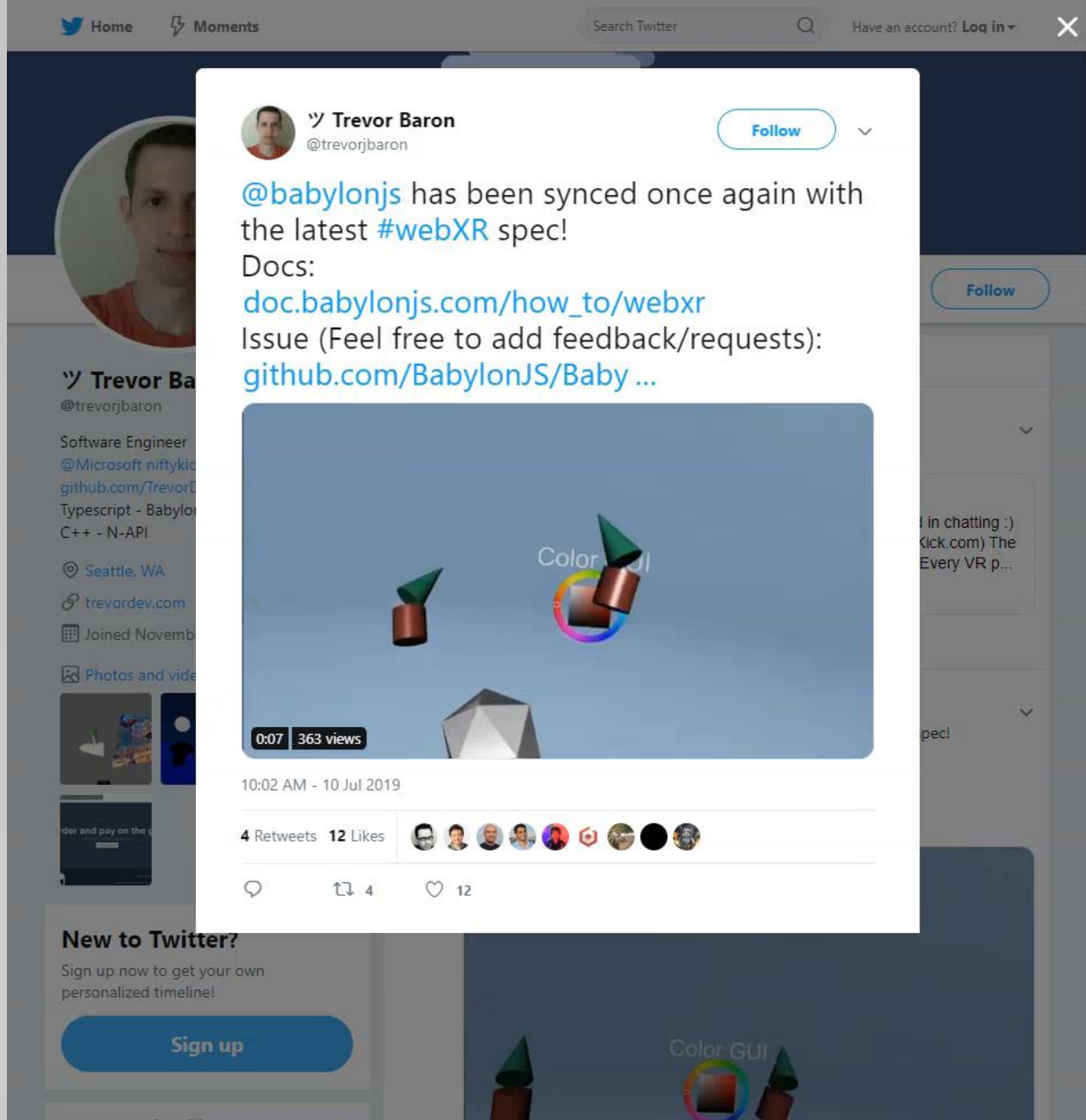


33 fps - 0.07 CPU ms

Web Standards

- WebGPU
- WebXR
- glTF with KTX2 + Basis Universal

https://doc.babylonjs.com/how_to/webxr



Web Standards

- WebGPU
- WebXR
- glTF with KTX2 + Basis Universal

<https://bit.ly/30U2sff>

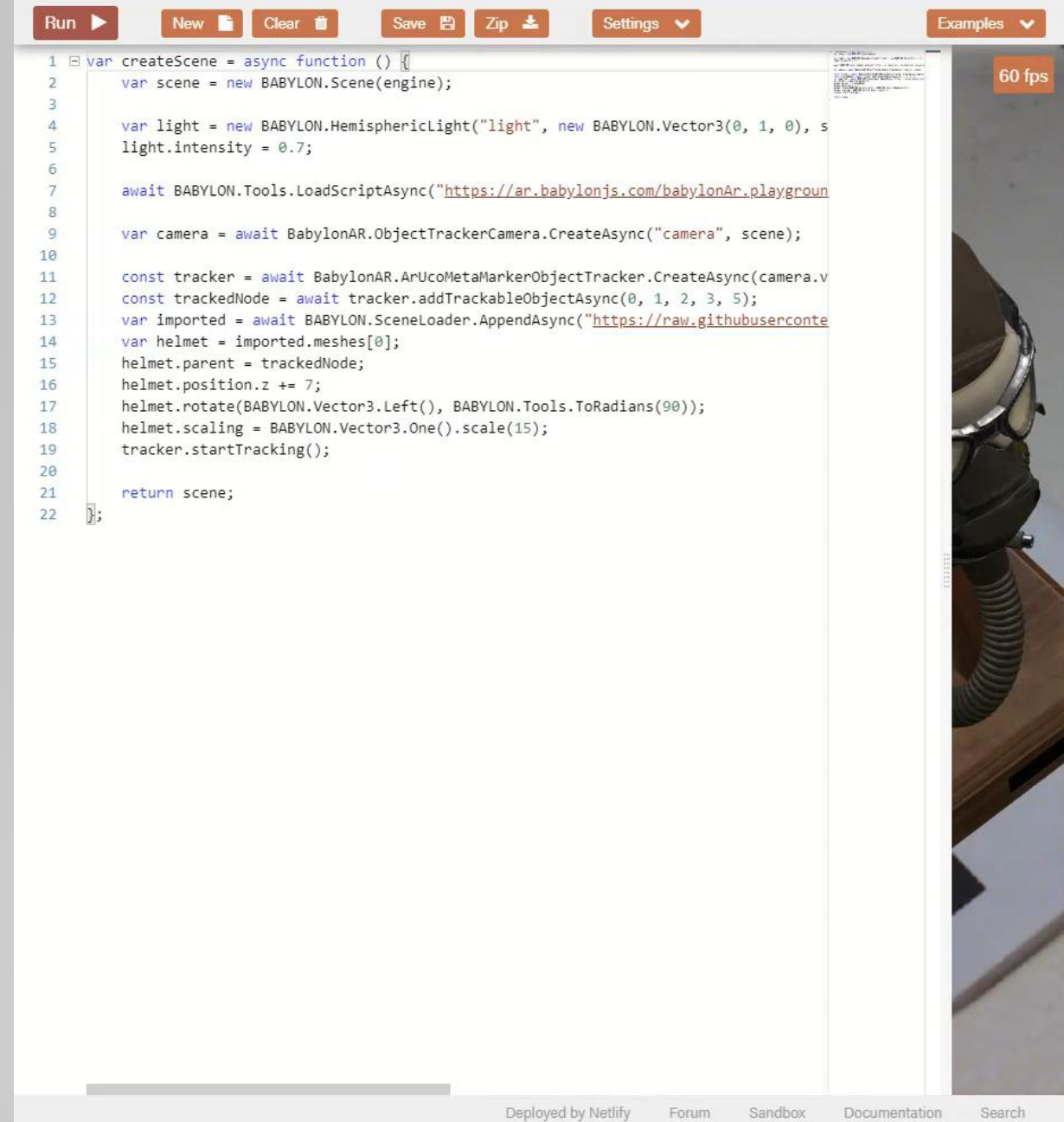


Experiments

- Babylon AR
- Babylon Native

<https://github.com/BabylonJS/BabylonAR>

<https://www.babylonjs-playground.com/#TU0ZCR>

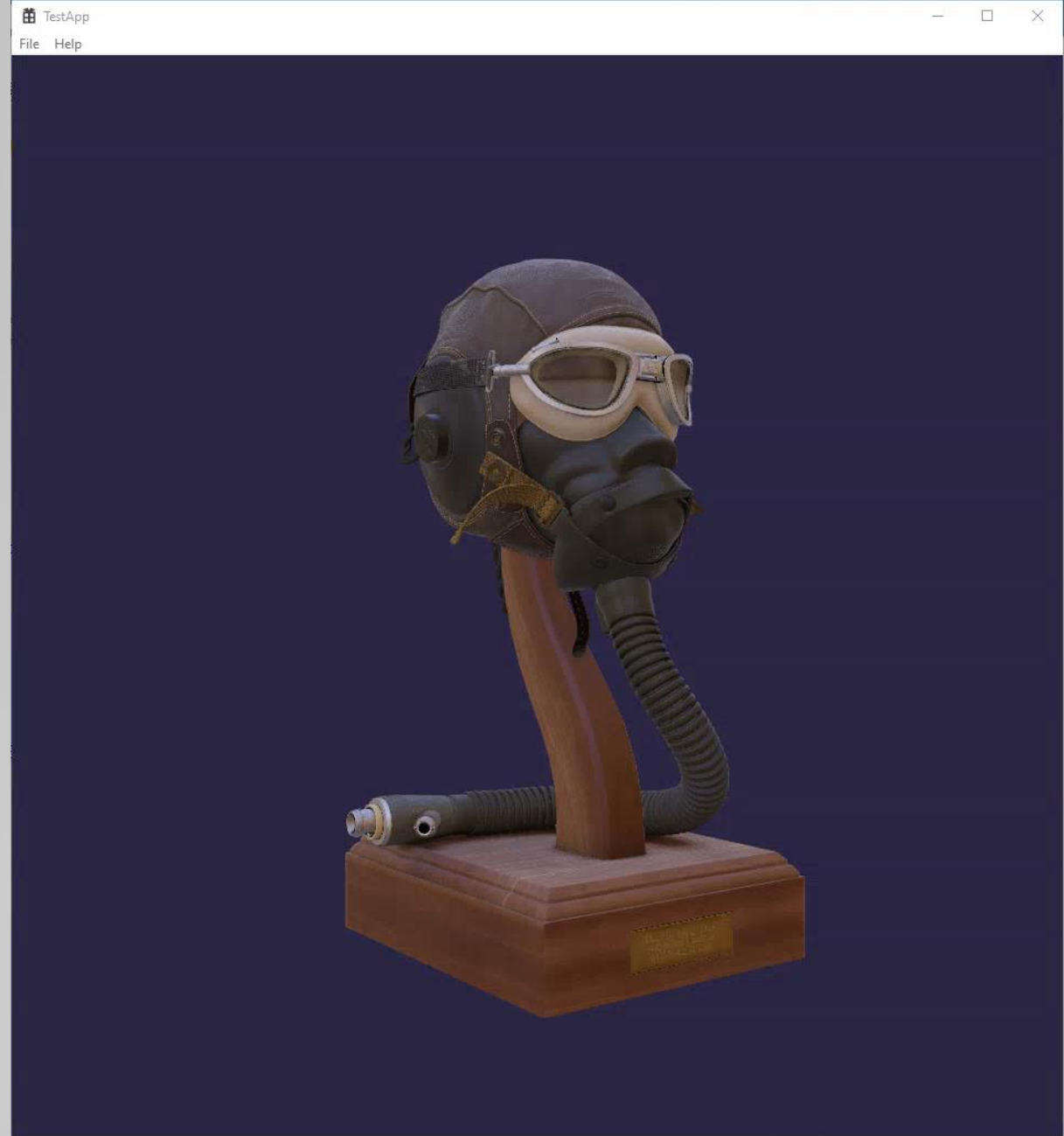


```
1 var createScene = async function () {
2     var scene = new BABYLON.Scene(engine);
3
4     var light = new BABYLON.HemisphericLight("light", new BABYLON.Vector3(0, 1, 0), scene);
5     light.intensity = 0.7;
6
7     await BABYLON.Tools.LoadScriptAsync("https://ar.babylonjs.com/babylonAr.playground");
8
9     var camera = await BabylonAR.ObjectTrackerCamera.CreateAsync("camera", scene);
10
11     const tracker = await BabylonAR.ArUcoMetaMarkerObjectTracker.CreateAsync(camera, scene);
12     const trackedNode = await tracker.addTrackableObjectAsync(0, 1, 2, 3, 5);
13     var imported = await BABYLON.SceneLoader.AppendAsync("https://raw.githubusercontent.com/BabylonJS/BabylonAR/master/Assets/Models/helmet.obj");
14     var helmet = imported.meshes[0];
15     helmet.parent = trackedNode;
16     helmet.position.z += 7;
17     helmet.rotate(BABYLON.Vector3.Left(), BABYLON.Tools.ToRadians(90));
18     helmet.scaling = BABYLON.Vector3.One().scale(15);
19     tracker.startTracking();
20
21     return scene;
22 };
```

Experiments

- Babylon AR
- **Babylon Native**

<https://github.com/BabylonJS/BabylonNative>





Node Material Demo