Creating 3D experiences using WebGL & babylon.js

David Rousset
Sr Program Manager
Microsoft Corp
@davrous

Microsoft

DISCLAIMER

The current speaker is **French**

(expect a strange **accent** & different **jokes**)

The current speaker is from **Microsoft**

(he's using a Windows **PC** & browses the web with **IE** !)

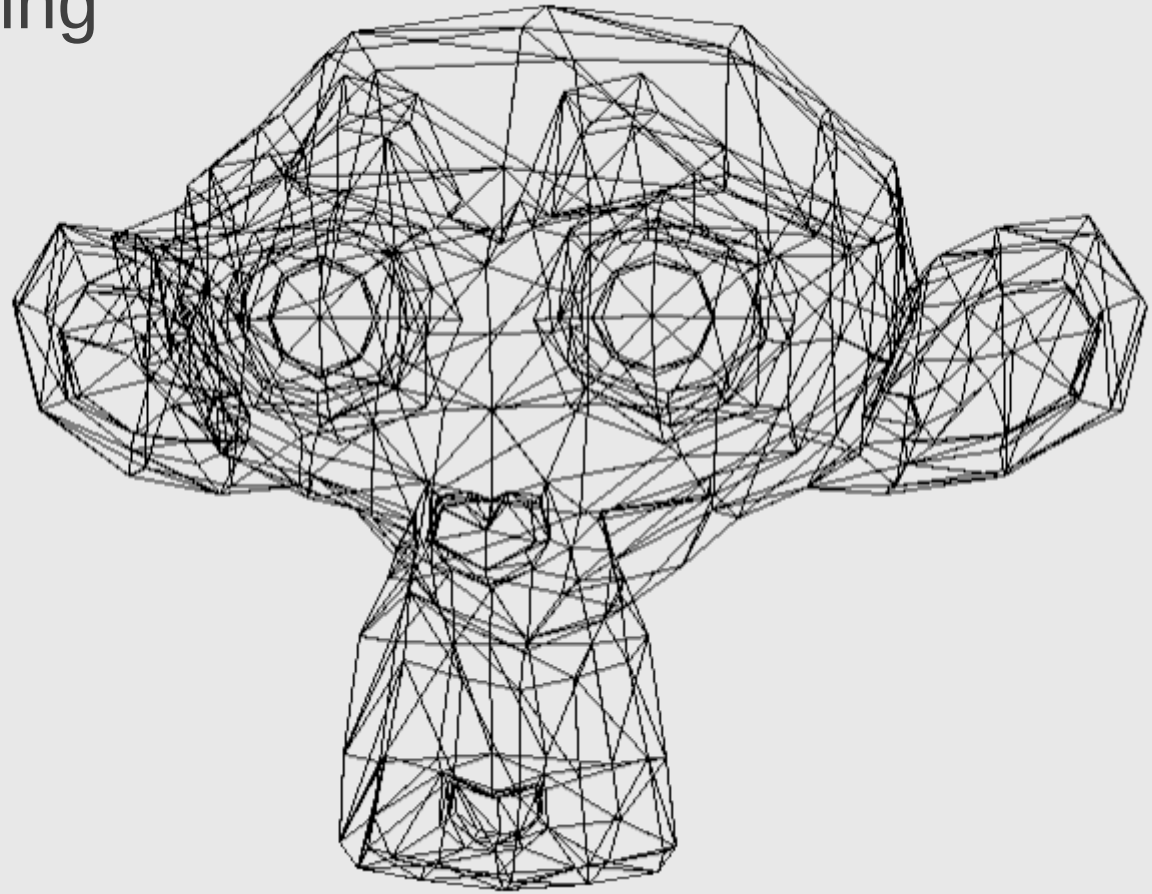
The current speaker is a **developer**

(he's remarkably **bad** in basic obvious **design** principles)

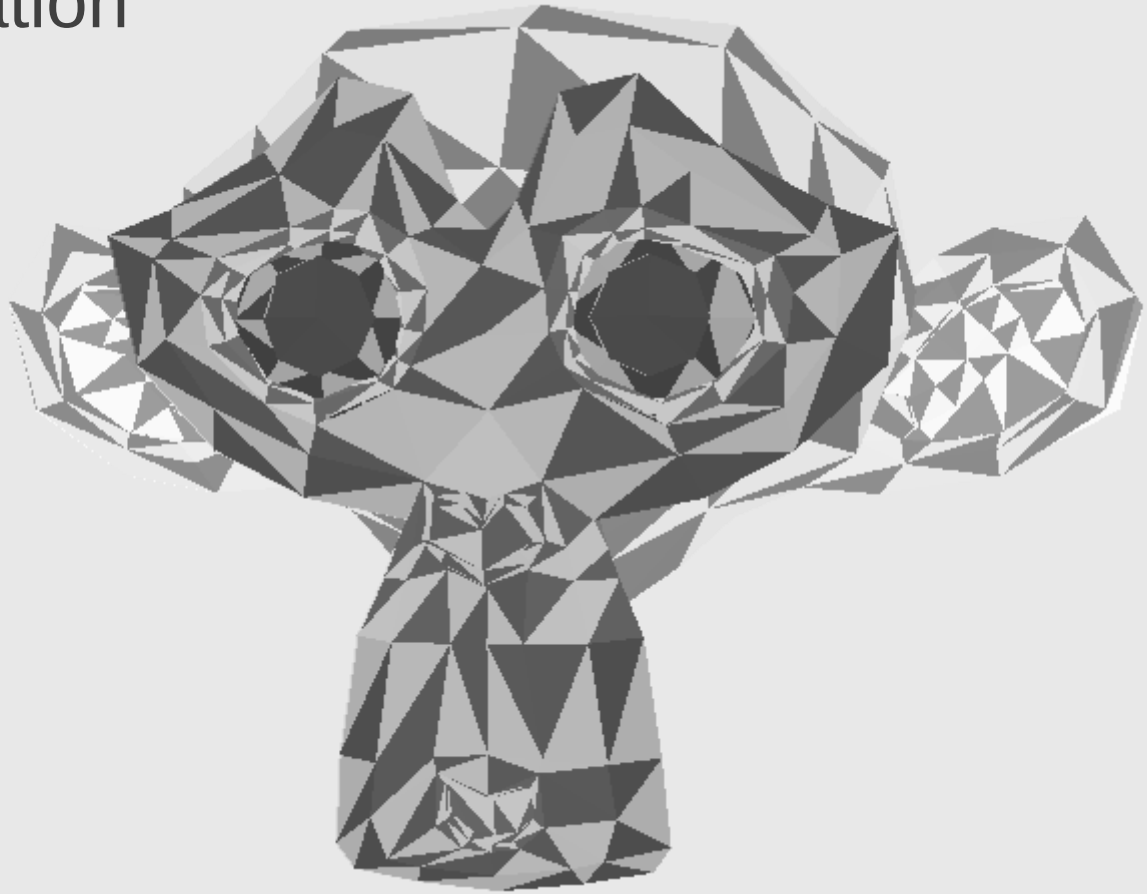
Why building a **WebGL** 3D engine ?

Understanding 3D Basics via a soft engine

Wireframing

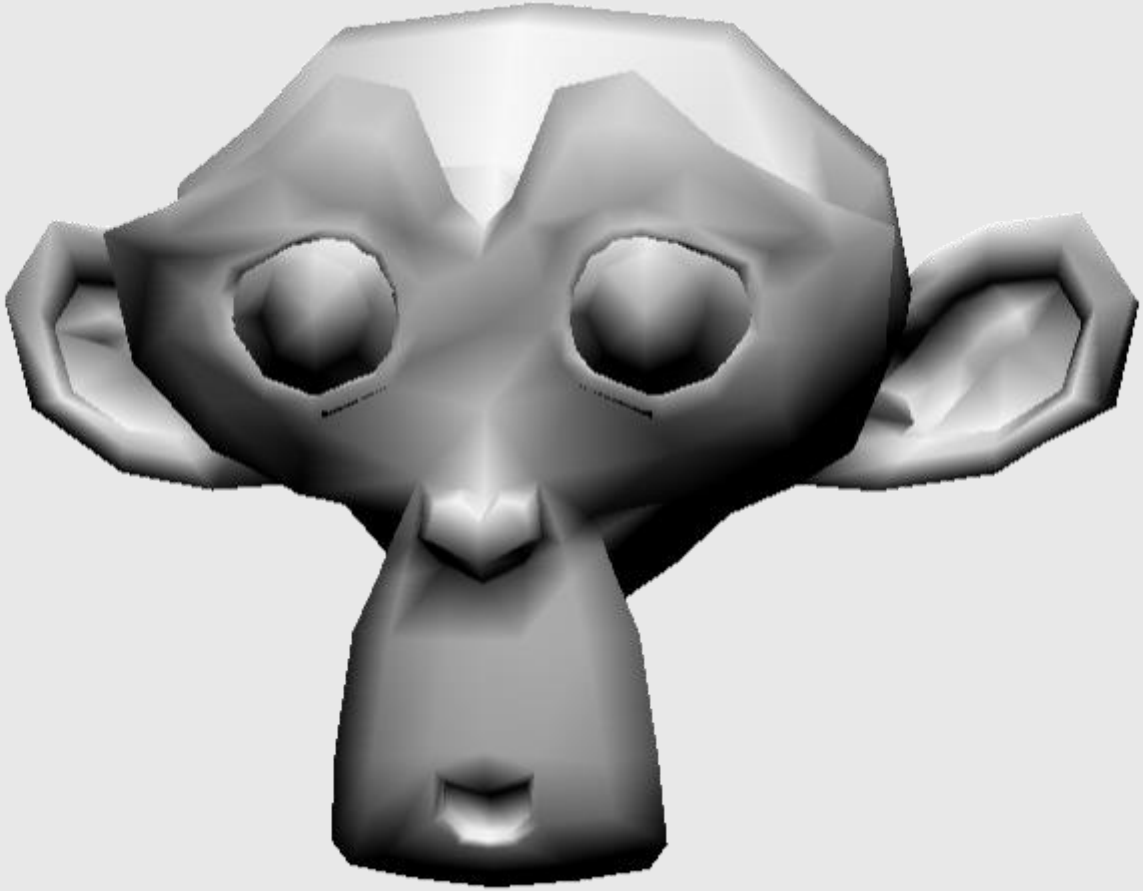


Rasterization



Flat Shading

Gouraud Shading



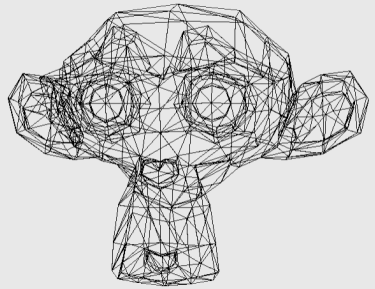
Texture mapping

3D Software Engine

DEMONSTRATION

Some 3D engine vocabulary

- A point in the 3D world = a **vertex**
- Multiple vertex = **vertices**
- **Vector3** (x,y,z) is used for a 3D position or a direction
- Triangle = **face**
- A 3D object = a **mesh**



Using WebGL directly

Requires a **compatible** browser or device

A new **context** for the canvas:

```
canvas.getContext("webgl", { antialias: true}) ||
canvas.getContext("experimental-webgl", { antialias: true});
```

Using WebGL directly

WebGL is a **low** level API

Need to handle **everything** except the *rendering*:

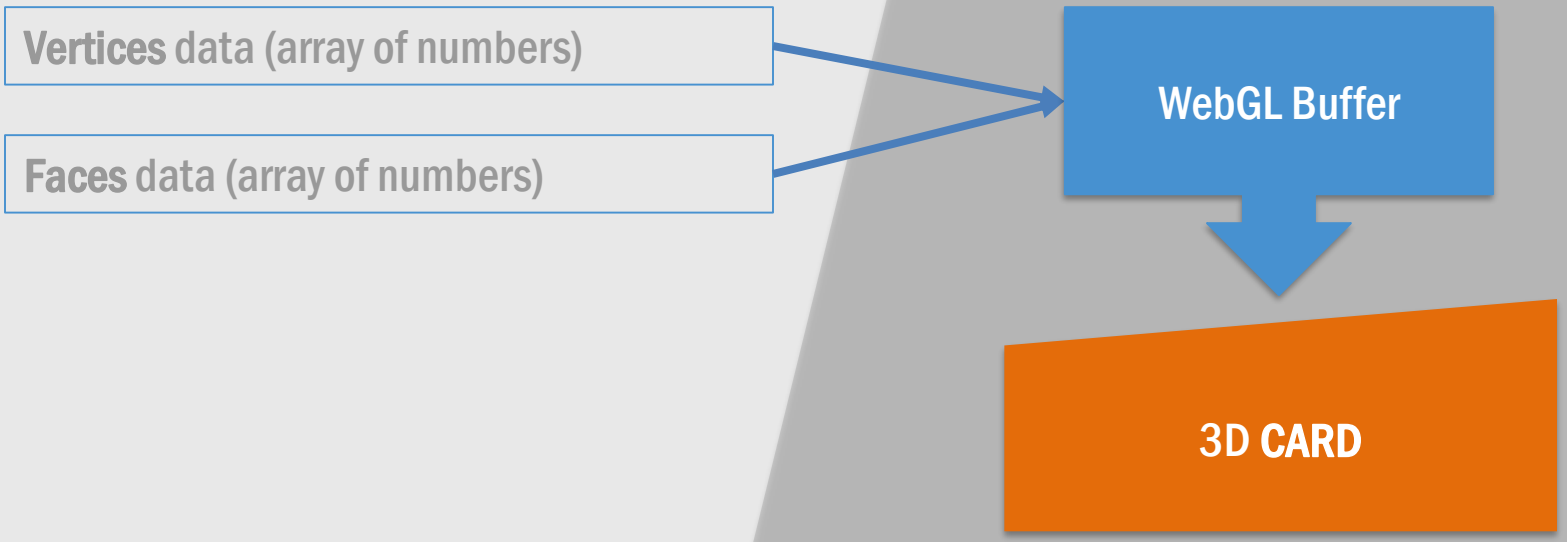
- Shaders code (loading, compilation)
- Geometry creation, topology, transfer
- Shaders variables management
- Texture and resources management
- Render loop

Simplest WEBGL code EVER

DEMONSTRATION

Understanding geometries and shaders

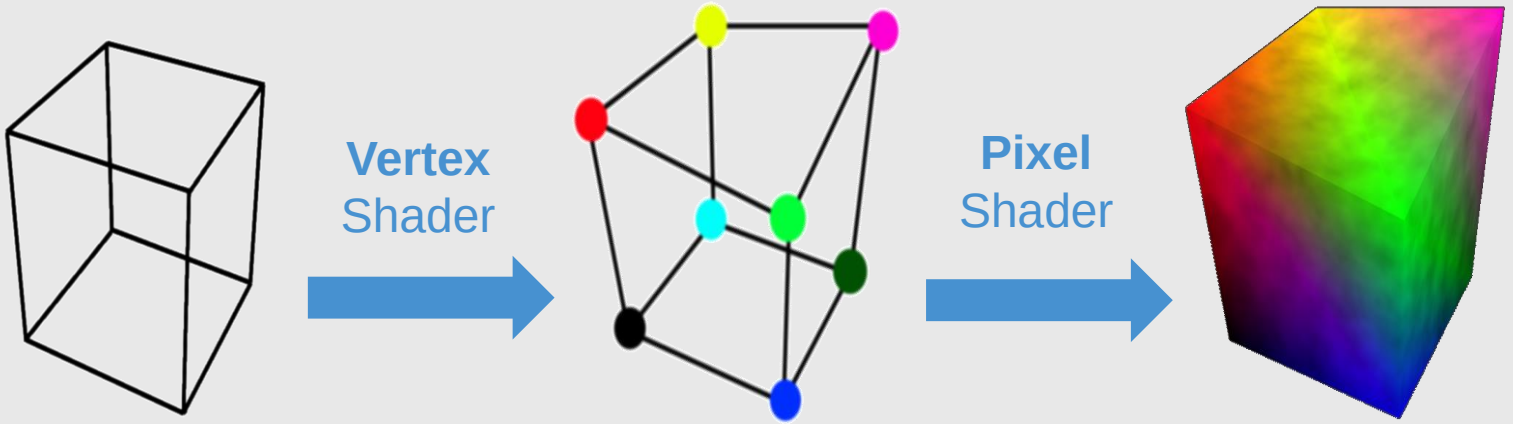
Geometries


Vertices data (array of numbers)
Faces data (array of numbers)
WebGL Buffer
3D CARD
CPU SIDE
GPU SIDE

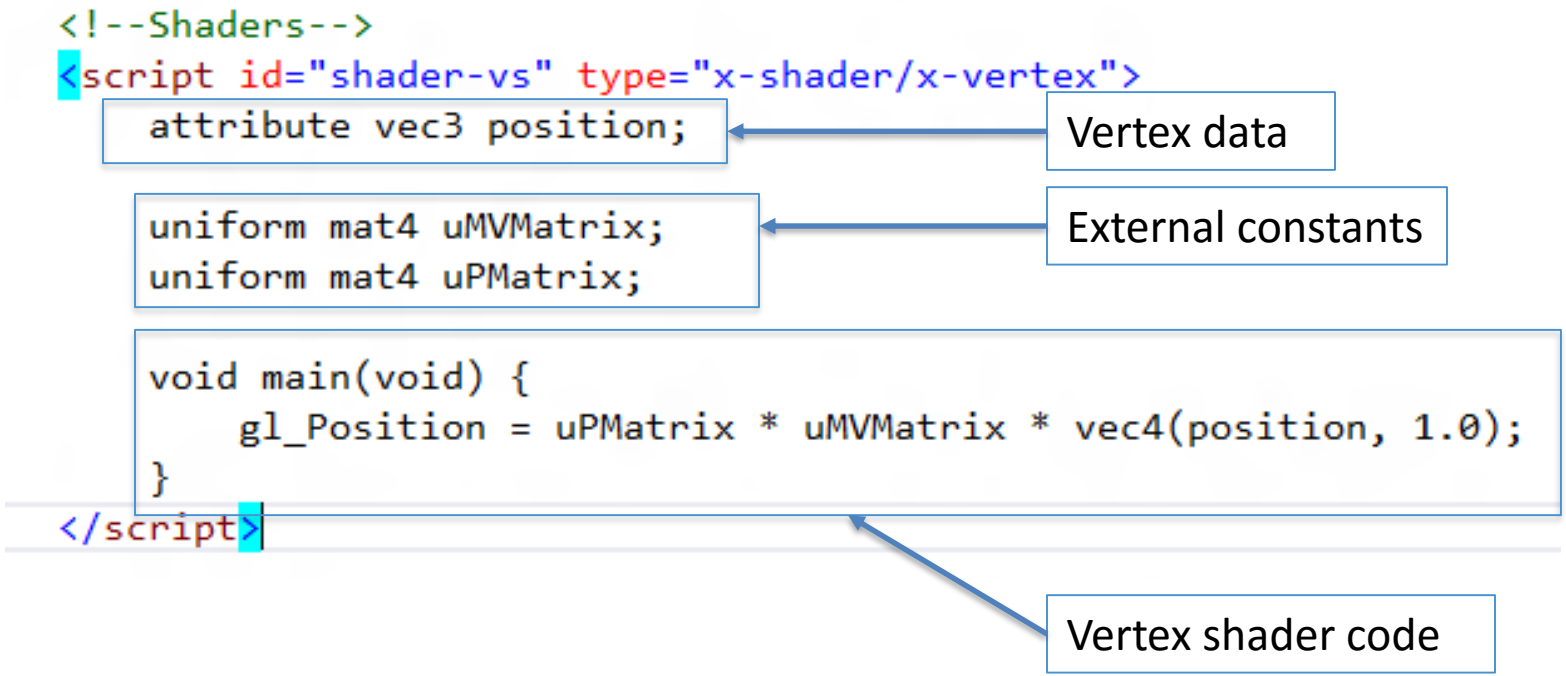
Shaders

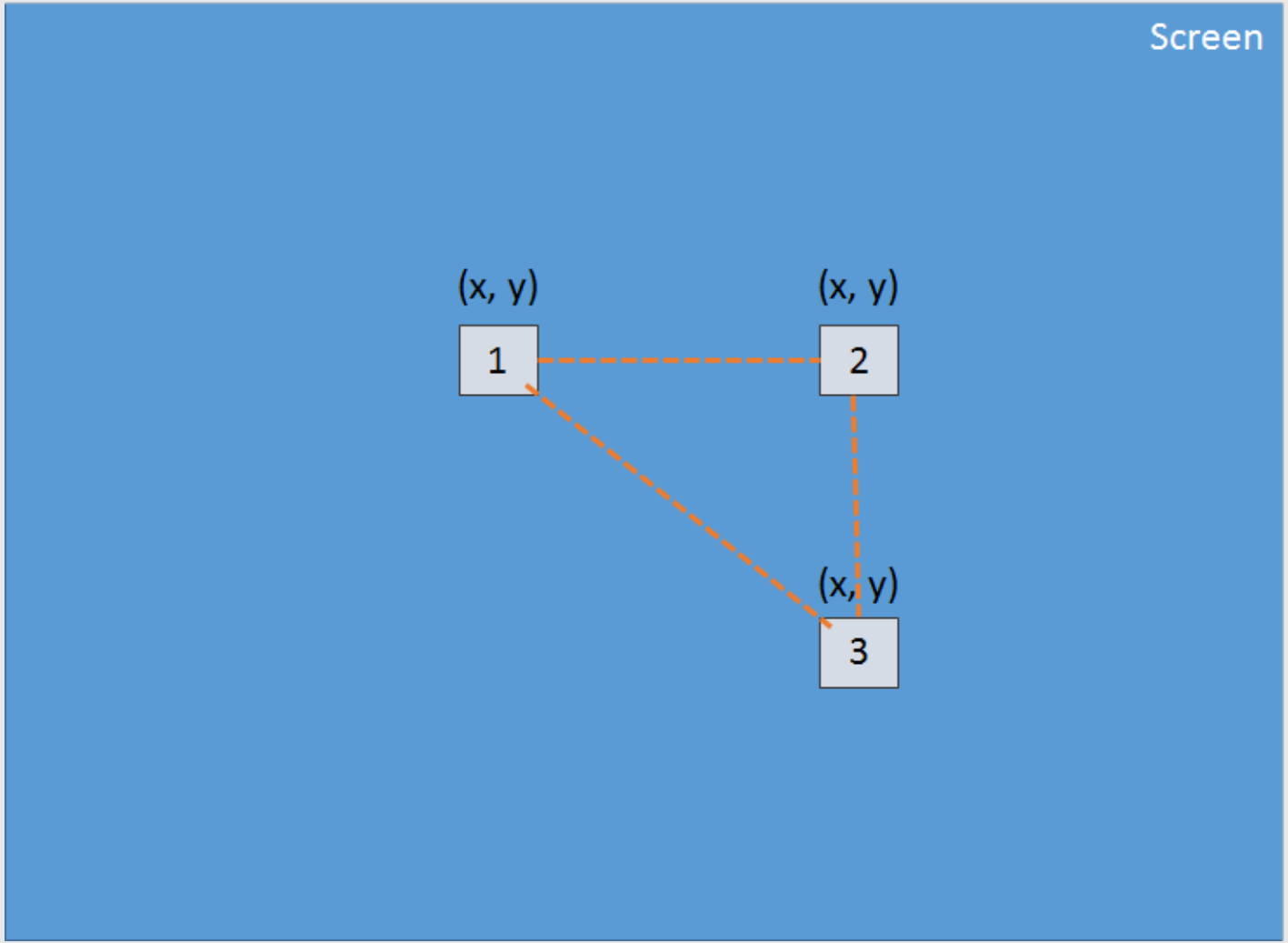
- Shaders are code for the **GPU**
- Language used is **GLSL** (Graphics Library Shader Language)

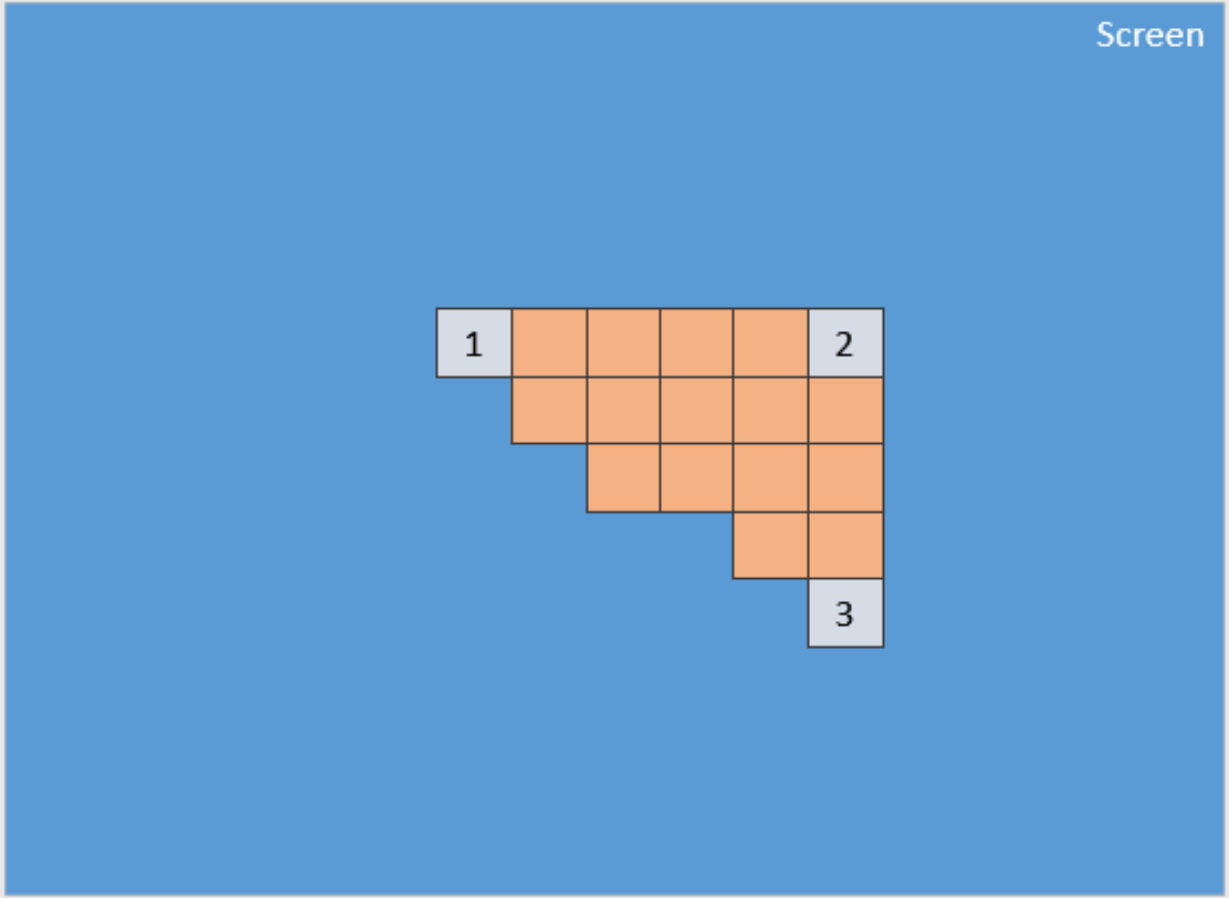
- Vertex shaders are about **transforming** geometry
- Pixel shaders are about computing **pixel color**


Vertex
Shader
Pixel
Shader

Anatomy of a vertex shader




Screen
(x, y)
1
(x, y)
2
(x, y)
3


Screen
1
2
3

Anatomy of a pixel shader

```
<script id="shader-fs" type="x-shader/x-fragment">
    precision mediump float;

    void main(void) {
        gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0);
    }
</script>
```

Pixel shader code

Draw me a **triangle**

DEMONSTRATION

Performance considerations

Performance first

Going under the hood...

STATE CACHING

WebGL is a state machine and changing states is expensive

GARBAGE COLLECTOR

Removing memory pressure to avoid FPS drops due to GC

SMART SHADERS

Compiling cutting edge shaders

Babylon.js ?

WebGL. simple. powerful.

Free & open source project (Apache 2 license):
https://github.com/babylonjs/babylon.js

Written in **TypeScript**

Our philosophy?

- **Simple** to use
- High **performance**
- Run **everywhere**

Advanced features

Blender, 3DS Max & Unity exporters
Design & render +
babylonjs.com/sandbox

Offline support
IndexedDB

Complete **collisions** and **physics** engine

Network optimizations
Incremental loading

Advanced features

Advanced texture support (Bump, **DDS**)

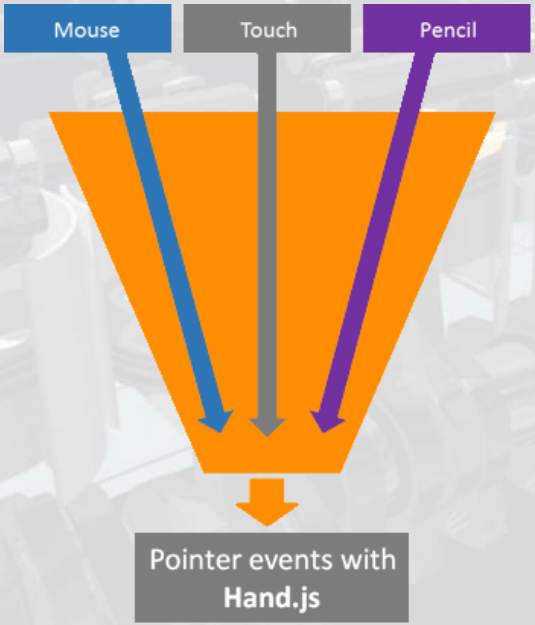
Smart shaders engine and postprocesses

Touch, Gamepad, Oculus & virtual joysticks

Complete **Web Audio** engine

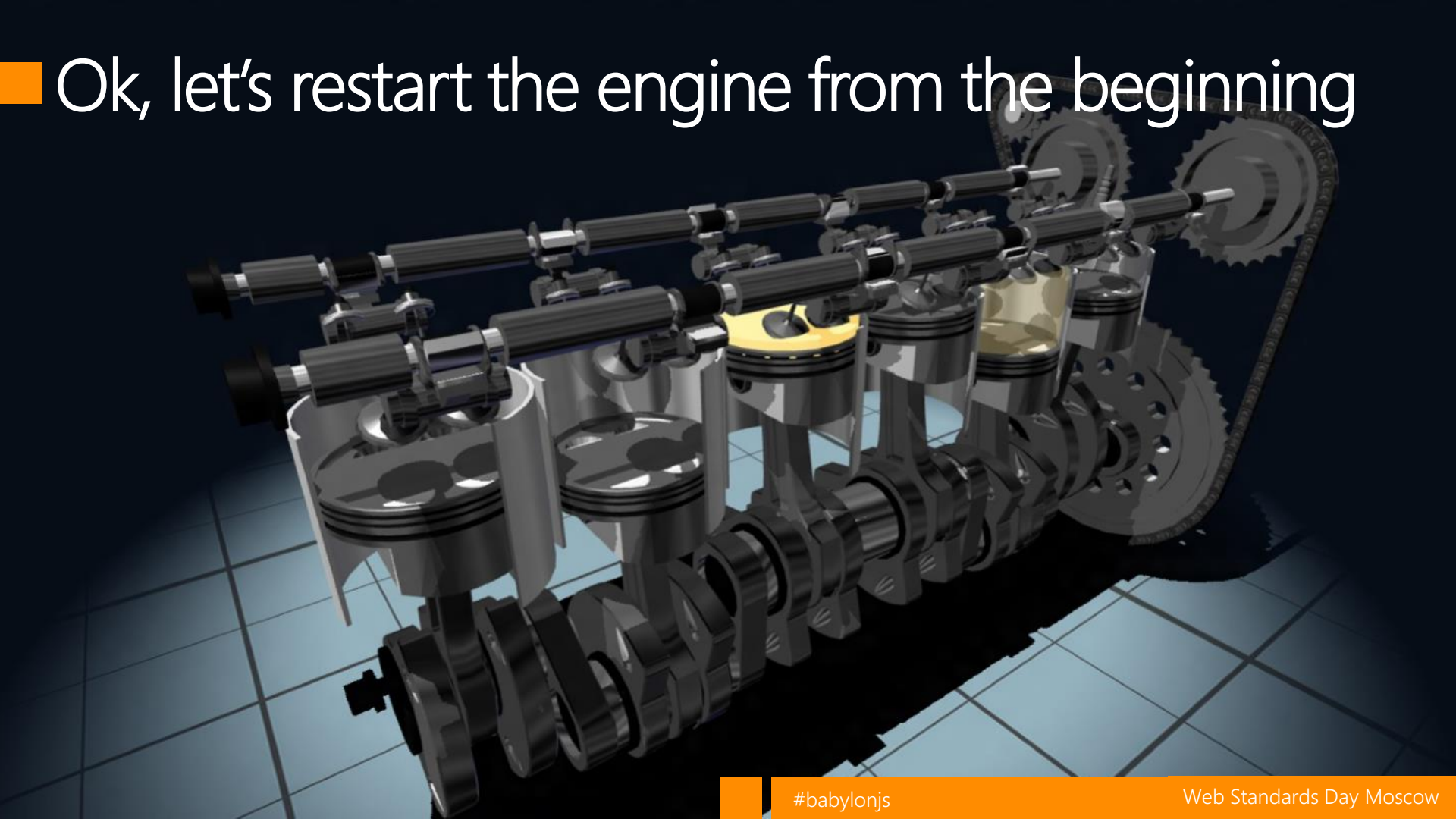
Handling **touch** devices

One event to rule them all!



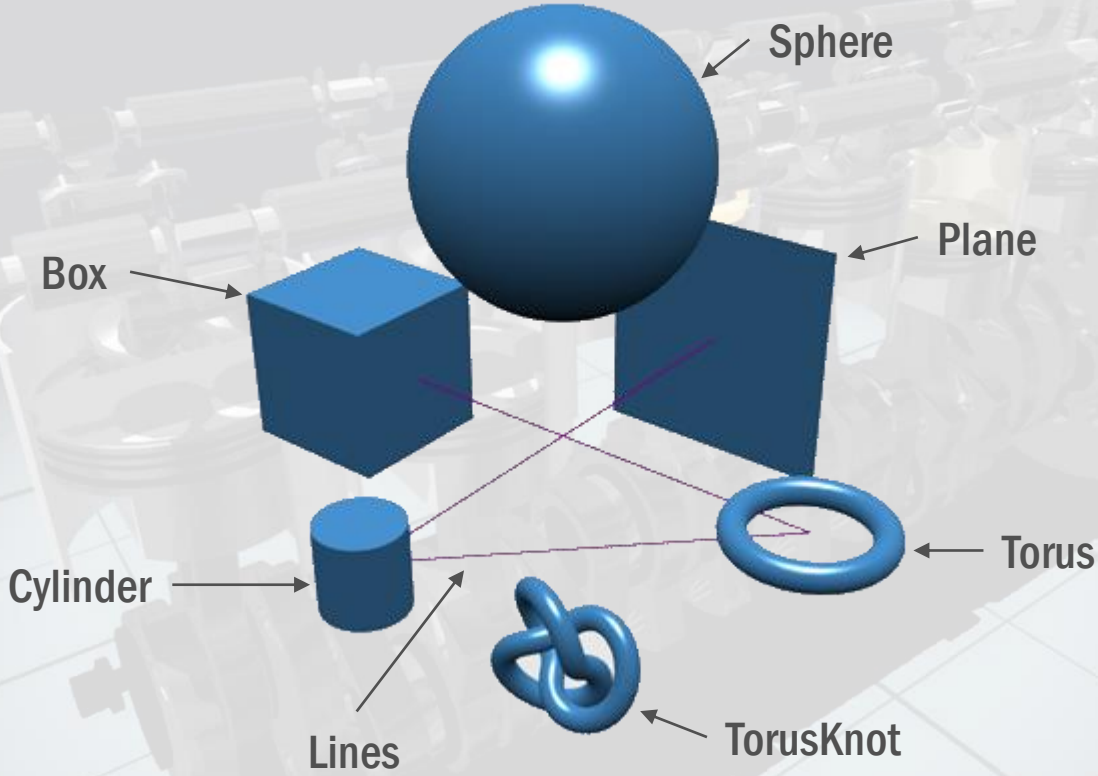

http://race.assassinscreedpirates.com/
Assassin's Creed Pirates De...
babylon JS
DEMONSTRATION

Ok, let's restart the engine from the beginning



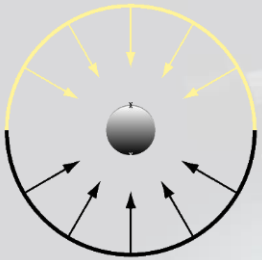
BABYLON.JS **Meshes**

Main primitives

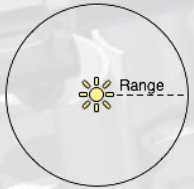


BABYLON.JS **Lights**

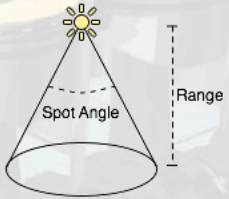
HemisphericLight



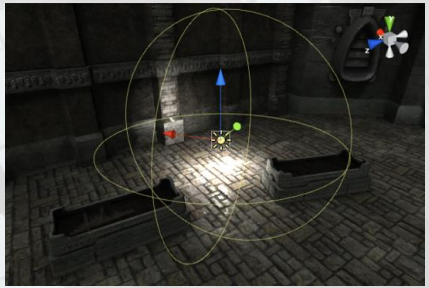
PointLight

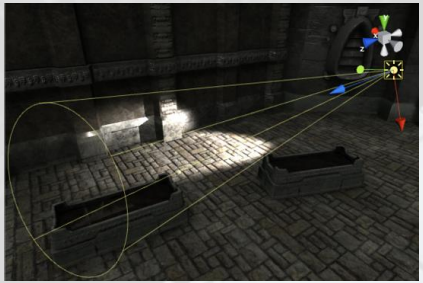


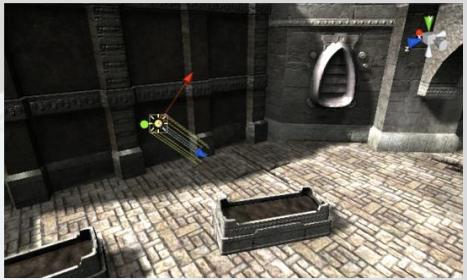
SpotLight



DirectionalLight







Learning Babylon.js using the **playground**

The power of TypeScript!


```
var createScene = function () {
    var CreateBosquet = function (name, x, y, z, scene, shadowGen
        var bosquet = BABYLON.Mesh.CreateBox(name, 2, scene);
        bosquet.position = new BABYLON.Vector3(x, y, z);
        bosquet.material = grassMaterial;

        var bosquetbawl = BABYLON.Mesh.CreateBox(name + "bawl", 1
        bosquetbawl.position = new BABYLON.Vector3(x, y + 1, z);
        bosquetbawl.material = grassMaterial;

        shadowGenerator.getShadowMap().renderList.push(bosquet);
        shadowGenerator.getShadowMap().renderList.push(bosquetbaw
    }

    var CreateTree = function (name, x, y, z, scene, shadowGenera
        var trunk = BABYLON.Mesh.CreateCylinder(name + "trunk", 7
        trunk.position = new BABYLON.Vector3(x, y, z);
        trunk.material = woodMaterial;

        var leafs = BABYLON.Mesh.CreateSphere(name + "leafs", 20,
        leafs.position = new BABYLON.Vector3(x, y + 5.0, z);
        leafs.material = grassMaterial;

        shadowGenerator.getShadowMap().renderList.push(trunk);
```

- Get sample code
- Try and experiment
- Share with friends
- Learn by reading examples

Playing with simple meshes

DEMONSTRATION

PLAYING WITH **INPUT**

Touch

Camera based on **pointer events**

Device Orientation

Camera based on **Device Orientation API**

Virtual Joysticks

Using pointer events, this camera generates **two joysticks** on top of the scene

Anaglyph

Use this camera with **Red/Green** glasses

Oculus

Control camera orientation with **Oculus Rift** device

Gamepad

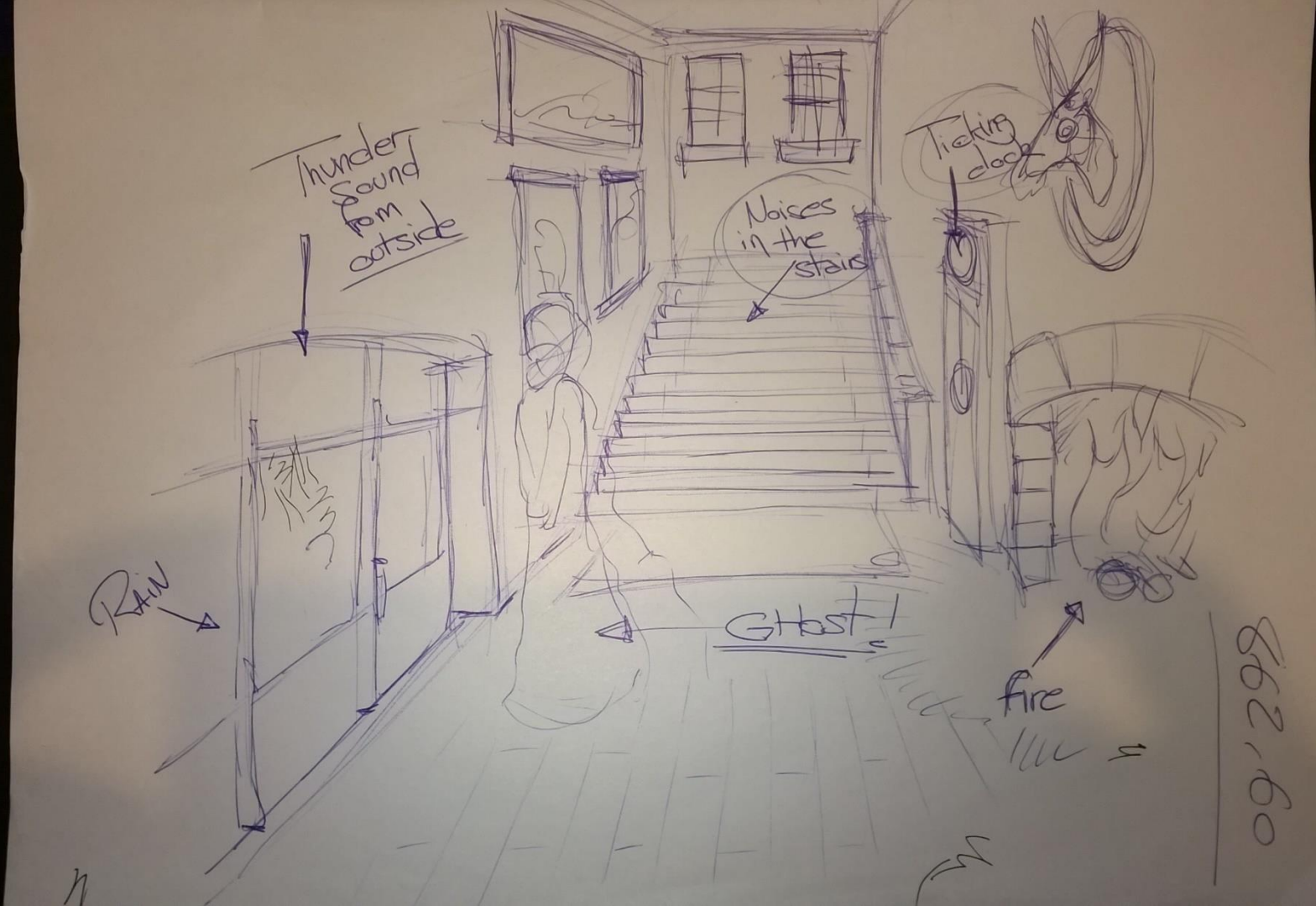
Use your **gamepad** to control your camera

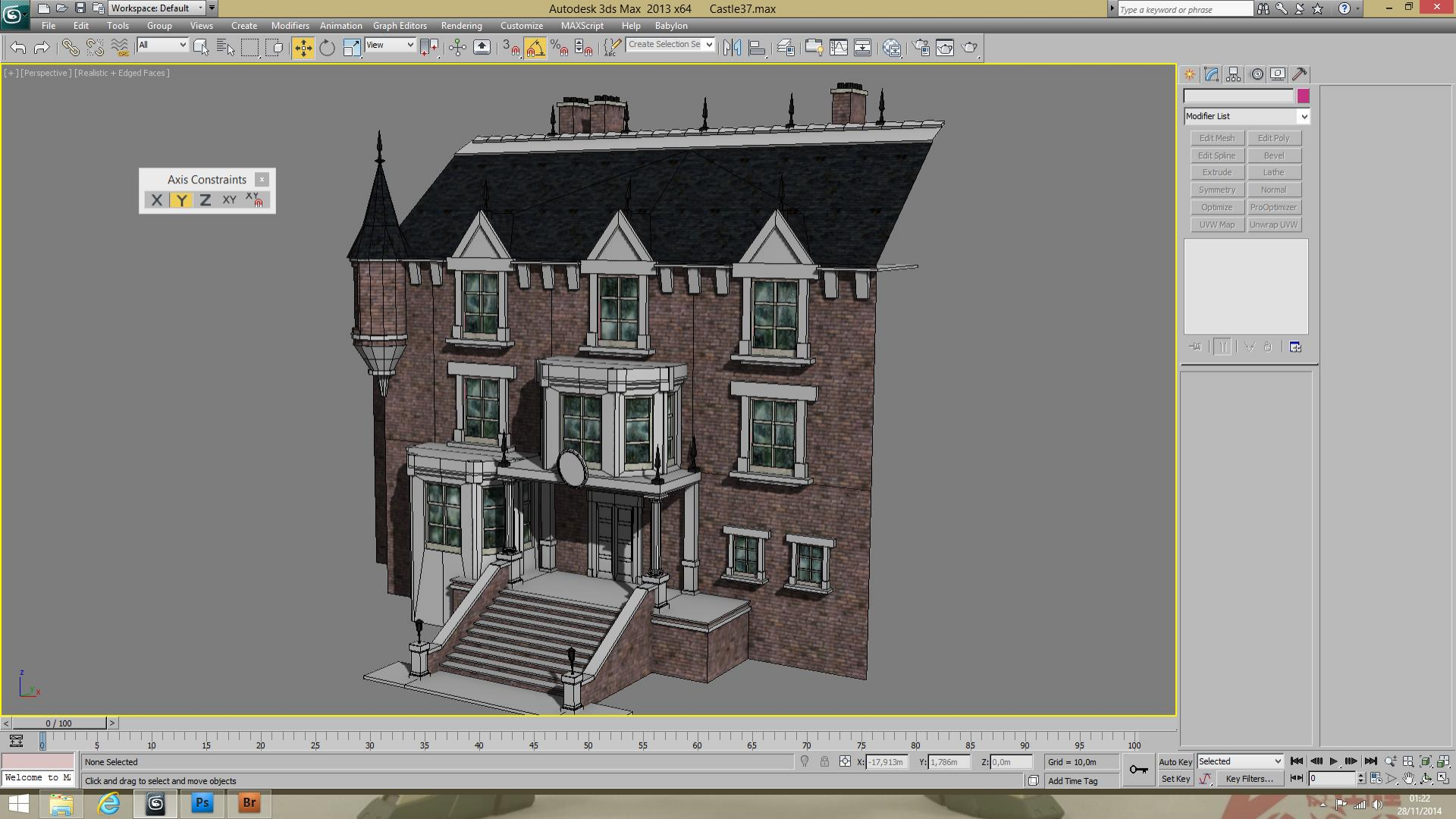
Switching cameras

DEMONSTRATION

Working with 3D artists


Night / outdoor env
Night Sounds
distant Thunder
Owl in the Trees
Frog Sound
Pond
Gravel Sound


Thunder Sound from outside
Noises in the stairs
Ticking clock
Rain
Ghost!
fire


Workspace: Default
Autodesk 3ds Max 2013 x64 Castle37.max
Type a keyword or phrase
File Edit Tools Group Views Create Modifiers Animation Graph Editors Rendering Customize MAXScript Help Babylon
[+] [Perspective] [Realistic + Edged Faces]
Axis Constraints
X Y Z XY XY
Modifier List
Edit Mesh Edit Poly
Edit Spline Bevel
Extrude Lathe
Symmetry Normal
Optimize ProOptimizer
UVW Map Unwrap UVW
0 / 100
None Selected
Click and drag to select and move objects
Welcome to M
Grid = 10,0m
Add Time Tag
Auto Key Selected
Set Key Key Filters...
01:22
28/11/2014

Creation Pipeline

From 3D tooling to WebGL using 0 line of code!

.FBX

.OBJ

.FBX

.OBJ

offline converter

.babylon

Sandbox for babylon JS

Visual Studio

Unity & Clara.io Exporter

DEMONSTRATION

Physics simulation

2 physics engines via a plug-in system

- Based on **Oimo.js** by default & **Canon.js** available
- Absolutely **uncorrelated from the native collision engine**

```
scene.enablePhysics(new BABYLON.Vector3(0, -10, 0), new BABYLON.OimoJSPlugin());
```

Set impostors

- Choose the right **impostor** for your mesh:
  - BABYLON.PhysicsEngine.**Plane**Imposter
  - BABYLON.PhysicsEngine.**Box**Imposter
  - BABYLON.PhysicsEngine.**Sphere**Imposter
  - BABYLON.PhysicsEngine.**Compound**Imposter

```
yourMesh.setPhysicsState(
   BABYLON.PhysicsEngine.BoxImpostor,
      {
            mass: 0,
            friction: 0.5,
            restitution: 0.7
      });
```

- To generate a physic effect on a mesh:
  - Let the **gravity** do its job
  - **Collisions between meshes** with physics enabled
  - Apply an **impulse** on the selected mesh at a given point

```
yourMesh.applyImpulse(direction, point); /* both BABYLON.Vector3 */
```

USING OIMO.JS WITH ESPILIT

DEMONSTRATION

Babylon.js audio engine
Simplicity again as a foundation

Based on **Web Audio**

Supports **ambient**, **omnidirectional** or **directional**

3D sound using **linear attenuation** by default

Managed by **code** or by loading our **.babylon** format

Supported by our **3DS Max exporter** (Blender & Unity to come)

Audio engine

DEMONSTRATION

Debug layer

Tool to help you reviewing performance issues

Draw calls

Time spent per feature

Number of objects

Number of active vertices

Are you GPU / CPU locked?

User marks + F12

Other interesting features

There's a lot more!

LOD

Simd.js support

Web Workers for collisions

PBR rendering pipeline, Reflection Probes and much more!

Useful links

What we're working on in MS Edge: status.modern.ie

- like WebRTC, Subclassing (ES6), Pointer Lock, etc.

Play with Babylon.js demos on www.babylonjs.com

- and try some tutorials via our playground: www.babylonjs.com/playground
- Documentation: http://doc.babylonjs.com
- Forum: http://www.html5gamedevs.com/forum/16-babylonjs/

Contact the MS Edge Developer Relations team on twitter: @msedgedev

And Babylon.js developers: @deltakosh & @davrous

Questions?

@deltakosh
@davrous
@rousseau_michel

www.babylonjs.com
#babylonjs