

# turbulenz

WebGL Camp Europe · David Galeano · July 2012

# David Galeano

*Co-founder of Turbulenz Limited in 2009*

*Technical Director of EA Tech Graphics*

*Team leader of RenderWare Graphics*

*Game developer at Dinamic Multimedia*







# BETWEEN US WE'VE MADE A TON OF GAMES

(and RenderWare, an industry leading game engine)



Turbulenz is a vertical solution for  
**developing** and **publishing** high  
quality **games** on the **web**.



# Three game demos

3D



FPS Tech Demo

3D



Space Ark

2D



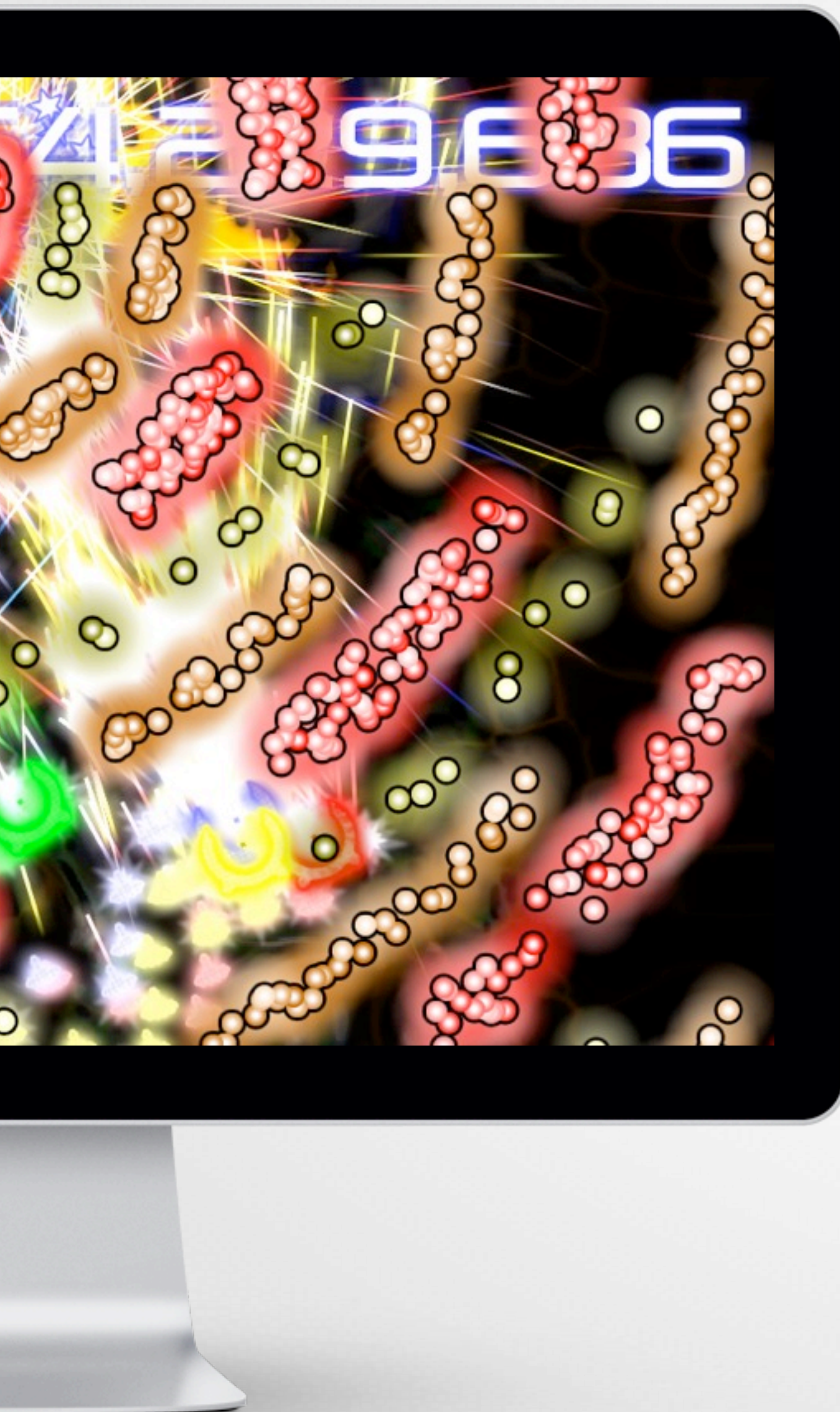
Score Rush



# Demo: Score Rush







# Score Rush

- 8000 bullets on INSANE mode, hundreds of enemies
  - ✓ Using sprite sheets / texture atlas for batching
- Massive textures, some 4096x4096
  - ✓ DXT compression critical for laptops with lower VRAM
  - ✓ Reduced dimensions when quality not critical
- Fullscreen post FXs, GPU limited
  - ✓ Minimize samples per pixel in fragment shaders



# Demo: Space Ark







# Space Ark

- **Massive GC pauses of ~500ms**
  - ✓ Reduced total number of objects from 4 million to 250k
  - ✓ Using Typed Arrays
  - ✓ Saved several hundred MB and GC pauses are now <50ms
- **~100MB of textures resident**
  - ✓ Using compressed DXT textures for fast loading and rendering performance
- **Unused assets not always garbage collected between levels**
  - ✓ Explicit destruction of WebGL objects when no longer needed
- **Some meshes have >100k vertices**
  - ✓ Splitting in 64K chunks



# FPS Tech Demo







# FPS Tech Demo

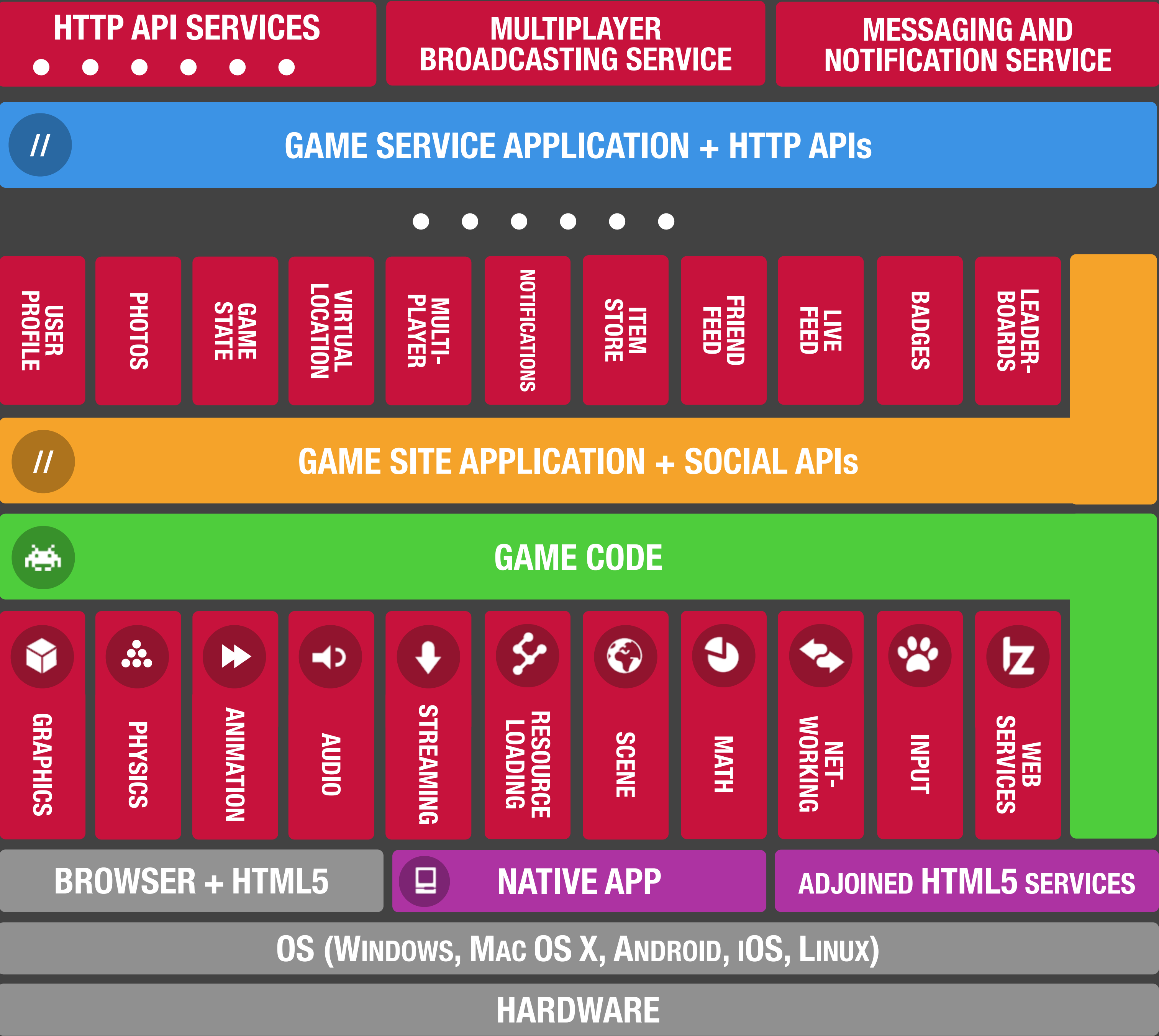
- **1694 assets to load**
  - ✓ Compression: 7-Zip better than gzip
  - ✓ Caching: assets with unique names to be cached for 10 years
  - ✓ Archives: group textures on tar files
- **353 lights, 346 particle systems**
  - ✓ Culling: portals, bounding box trees
- **55 shading techniques, 451 materials**
  - ✓ Flexible rendering system: CgFX files converted to JSON + GLSL
  - ✓ Minimize state changes: sort by technique and material
- **Multiple file formats (.md5 .prog .map .cm .ase)**
  - ✓ Unified JSON based format



ENGINE AND SITE  
ARCHITECTURE

INTERNET  
GENERATION  
GAME  
ENGINE

CLASSIC  
GAME  
ENGINE

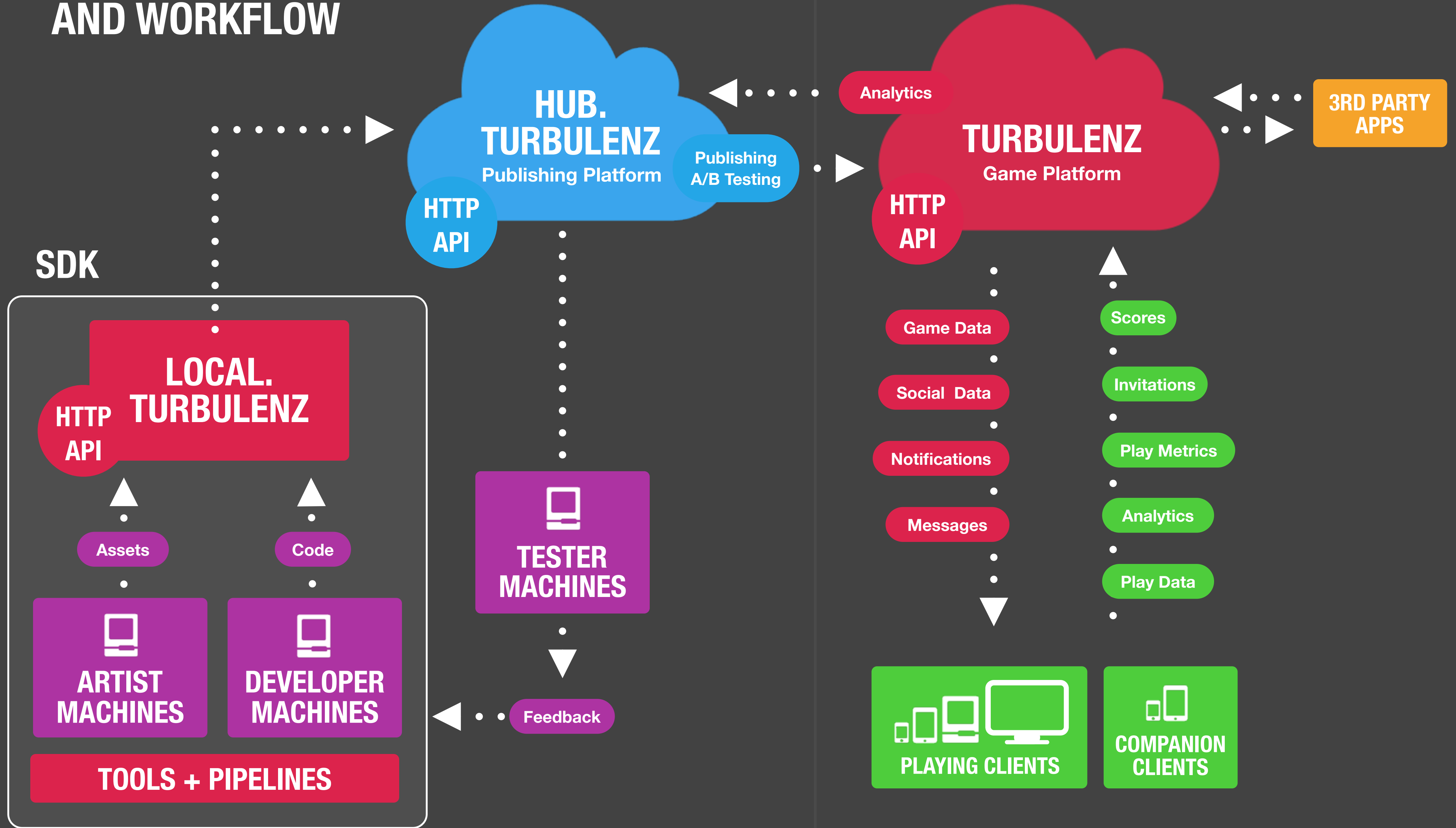


JAVASCRIPT

C++



# PLATFORM OVERVIEW AND WORKFLOW





Game site (beta live now)

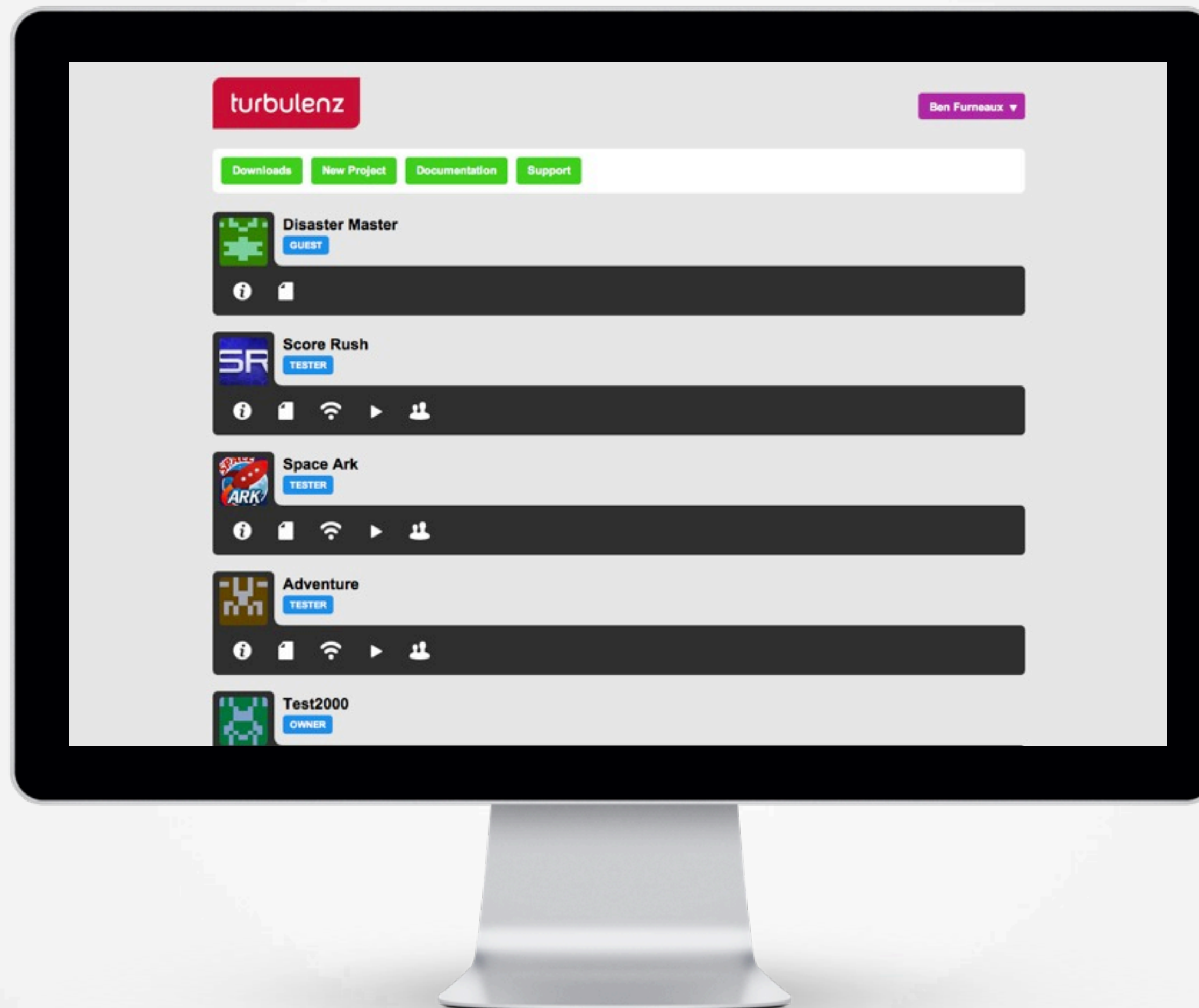
# turbulenz.com





Developer site and SDK download

# hub.turbulenz.com





# Online SDK documentation

# docs.turbulenz.com

## 3.3.5. Loading

We next expand the sample above to demonstrate building a CgFX shader and loading it for use at runtime.

1. Run the following command from the environment prompt in your project directory to build a CgFX shader into a JSON file:

```
*INSTALLDIR*/tools/bin/win32/cgfx2json.exe -i *INSTALLDIR*/assets/shaders/generic3D.cgfx -o generic3D.cgfx.json
```

2. Add the following code just after the floor is created:

```
var shader = null;
var technique = null;

// Load Shader
var requestHandler = RequestHandler.create({});
requestHandler.request({
  src: 'generic3D.cgfx.json',
  onload: function (shaderJSON)
  {
    var shaderParameters = JSON.parse(shaderJSON);
    shader = gd.createShader(shaderParameters);
    technique = shader.getTechnique('vertexColor3D');
  }
});

// Technique Parameters
var techniqueParameters = gd.createTechniqueParameters({
  worldViewProjection: md.m44BuildIdentity()
});

// Create a vertex buffer for a cube
var vertLBF = [ -20, -20, 20, 1, 0, 0, 1 ];
var vertRTF = [ -20, 20, 20, 0, 1, 0, 1 ];
var vertRTB = [ -20, 20, -20, 0, 1, 0, 1 ];
var vertLBF = [ -20, -20, 20, 1, 0, 0, 1 ];
var vertLBB = [ -20, -20, -20, 0, 0, 1, 1 ];
var vertLTB = [ -20, 20, -20, 1, 1, 0, 1 ];
var vertRTB = [ -20, 20, -20, 1, 0, 0, 1 ];
var vertRBB = [ 20, -20, -20, 0, 1, 0, 1 ];
var vertRBF = [ 20, -20, 20, 1, 1, 0, 1 ];
var vertData = [].concat(
  vertLTF, vertLBF, vertRTF, vertRTB, vertLBF, vertRBF, // front
  vertRTF, vertRBF, vertRTB, vertRTB, vertRBF, vertRBB, // right
  vertLTF, vertLBB, vertLTF, vertLTF, vertLBB, vertLBF, // left
  vertRTB, vertRBB, vertLTB, vertLTB, vertRBB, vertLBB, // back
  vertLTB, vertLTF, vertRTB, vertRTB, vertLTF, vertRTF, // top
  vertLBF, vertLBB, vertRBF, vertRBF, vertLBB, vertRBB // bottom
);
var numVerts = vertData.length;

var vertexBuffer = gd.createVertexBuffer({
  numVertices: numVerts
```



# Thanks for listening!

## Questions?

Game site (beta live now)

**turbulenz.com**

Developer site and SDK download

**hub.turbulenz.com**

Online SDK documentation

**docs.turbulenz.com**

David Galeano

**davidgaleano@turbulenz.com**