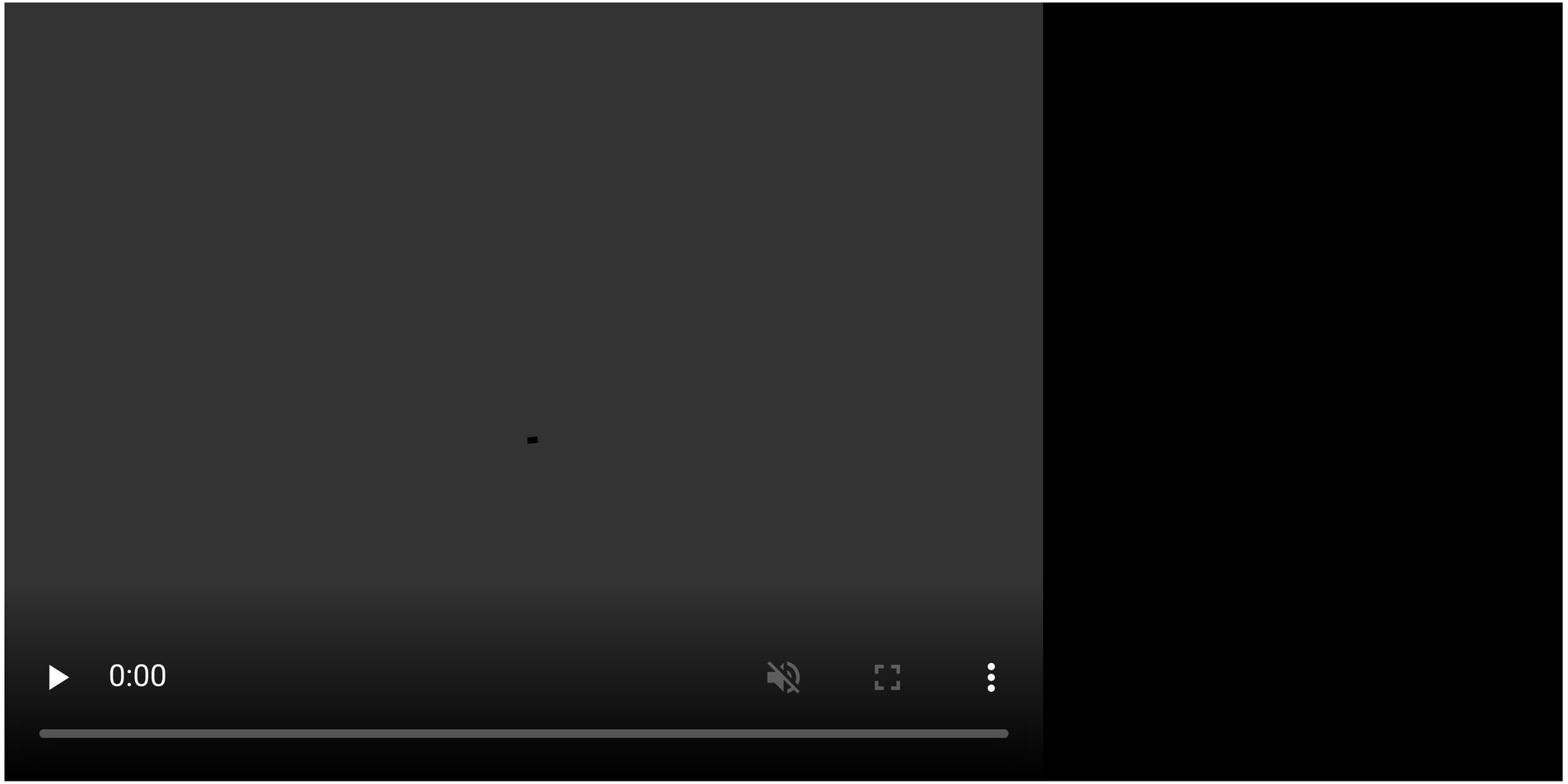
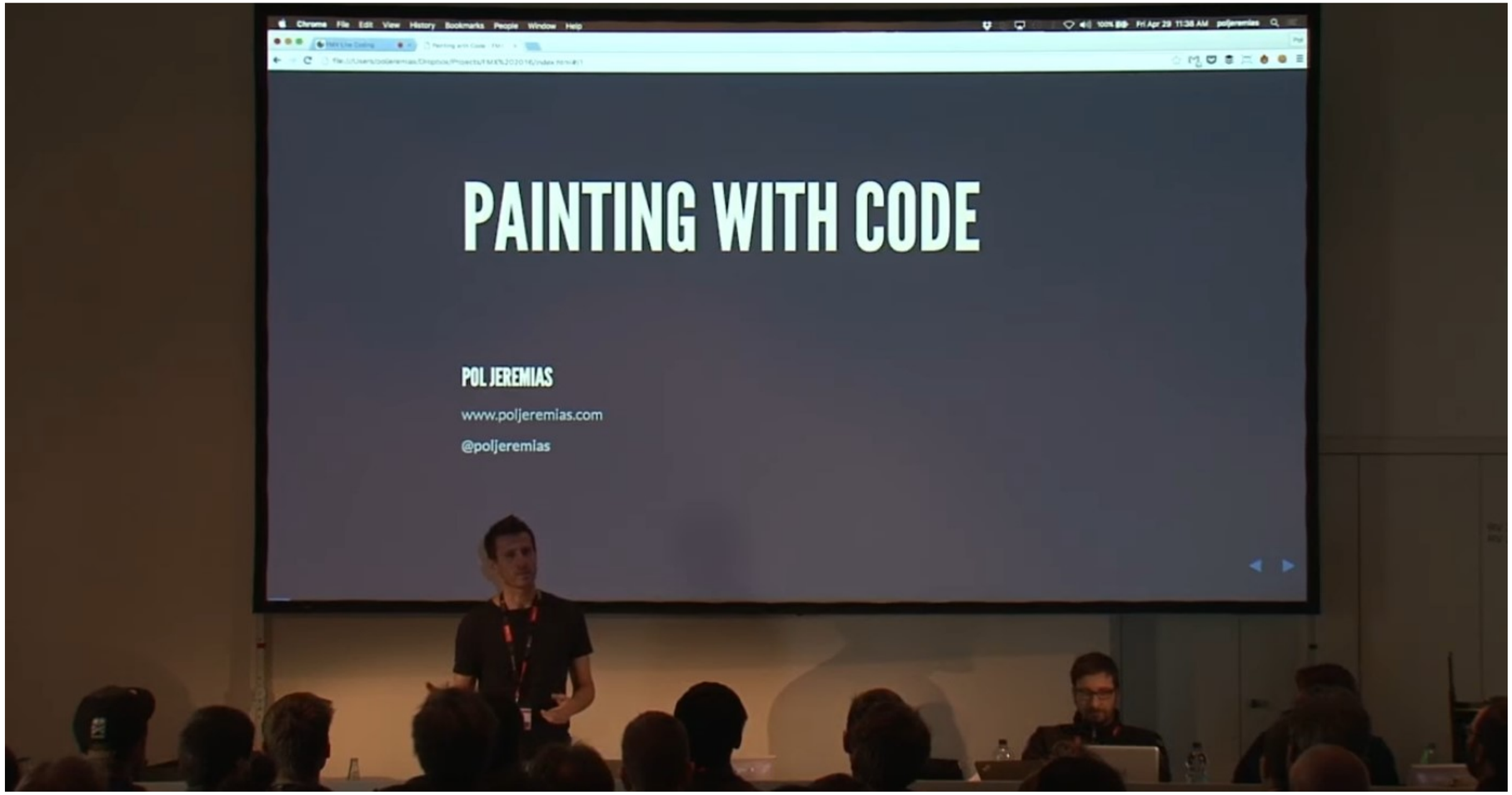
Draw Geometry with Code

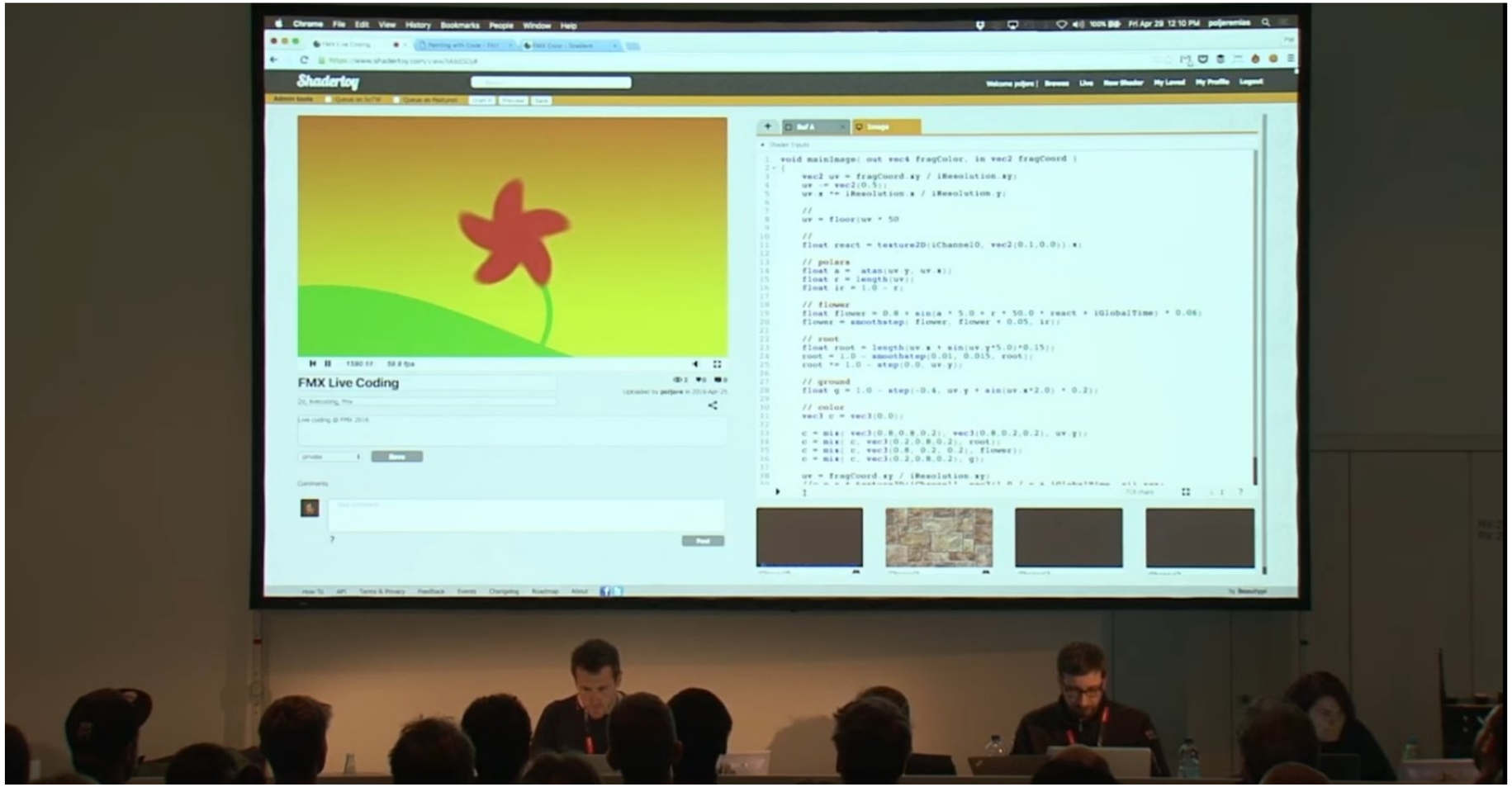


Background



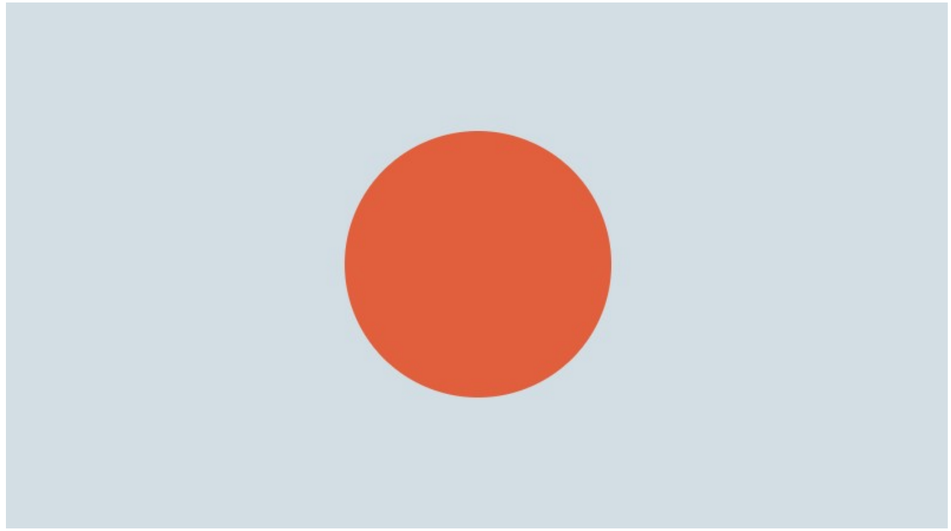
FMX Presentation

Background



FMX Presentation

A Circle with Code



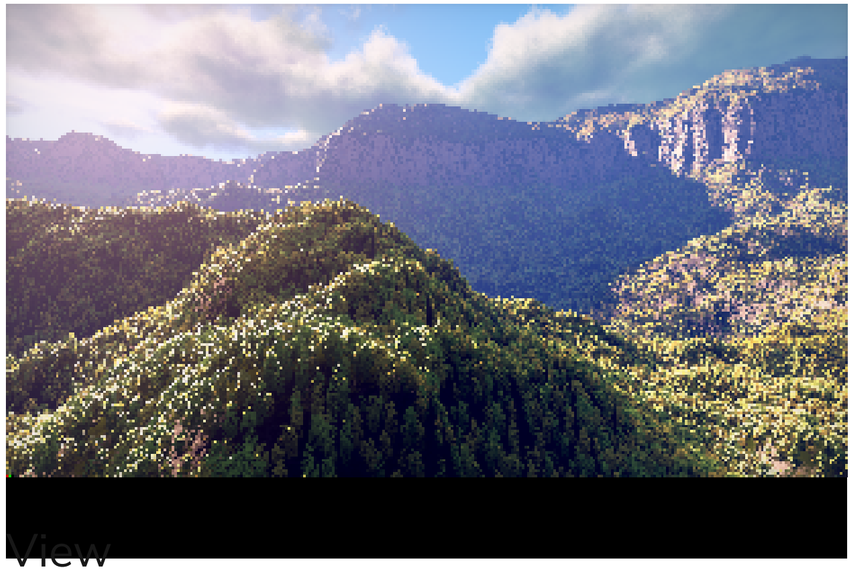
Live Example

```
vec3 rgb(float r, float g, float b) {
    return vec3(r / 255.0, g / 255.0, b / 255.0);
}

vec4 circle(vec2 uv, vec2 pos, float rad, vec3 color) {
    float d = length(pos - uv) - rad;
    float t = clamp(d, 0.0, 1.0);
    return vec4(color, 1.0 - t);
}

void mainImage( out vec4 fragColor, in vec2 fragCoord ) {

    vec2 uv = fragCoord.xy;
    vec2 center = iResolution.xy * 0.5;
    float radius = 0.25 * iResolution.y;

    // Background layer
    vec4 layer1 = vec4(rgb(210.0, 222.0, 228.0), 1.0);

    // Circle
    vec3 red = rgb(225.0, 95.0, 60.0);
    vec4 layer2 = circle(uv, center, radius, red);

    // Blend the two
    fragColor = mix(layer1, layer2, layer2.a);

}
```

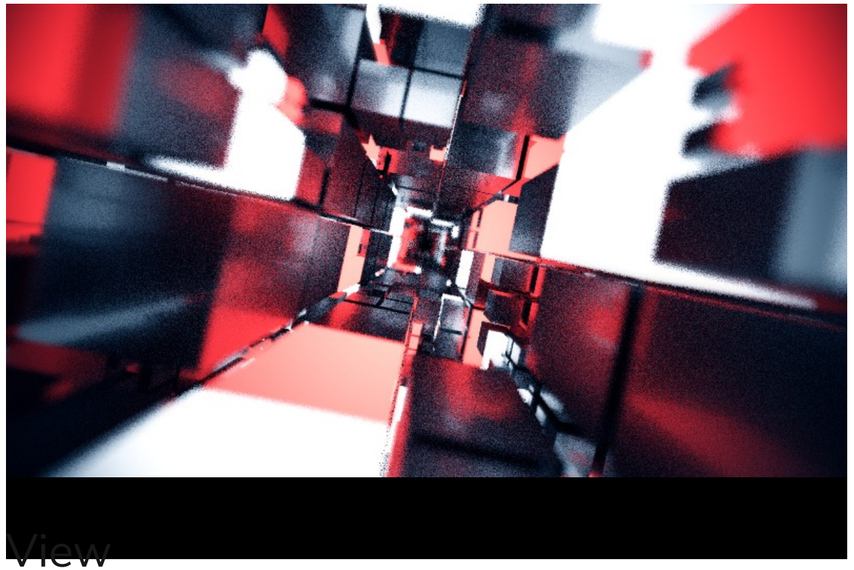
More Examples

View

View



View



View

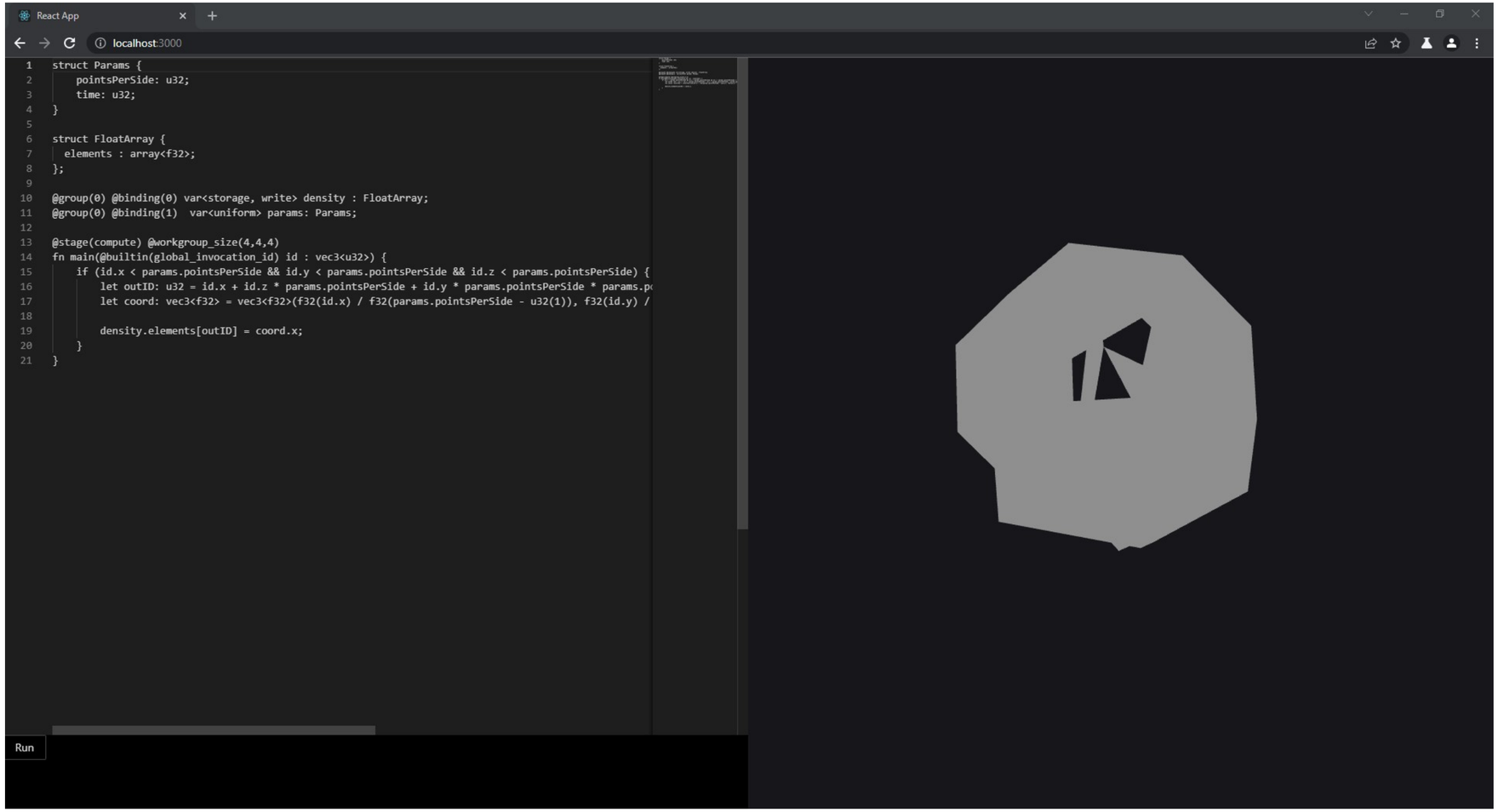
Marching Cubes on GPU

Challange

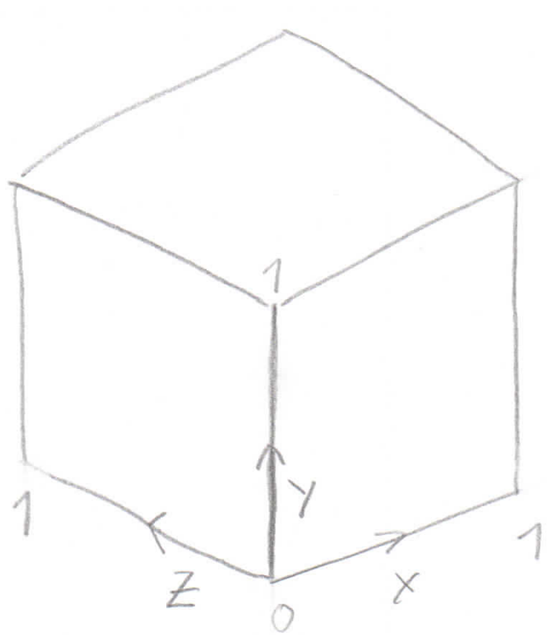
- Generating a connected mesh with threads that cant communicate with each other

Attempt in browser using WebGPU

WebGPU

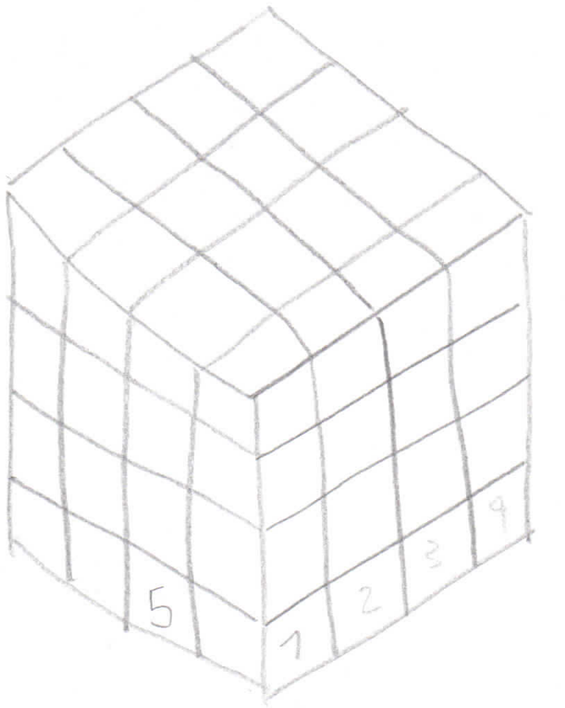


Shader describes the volume of a given space



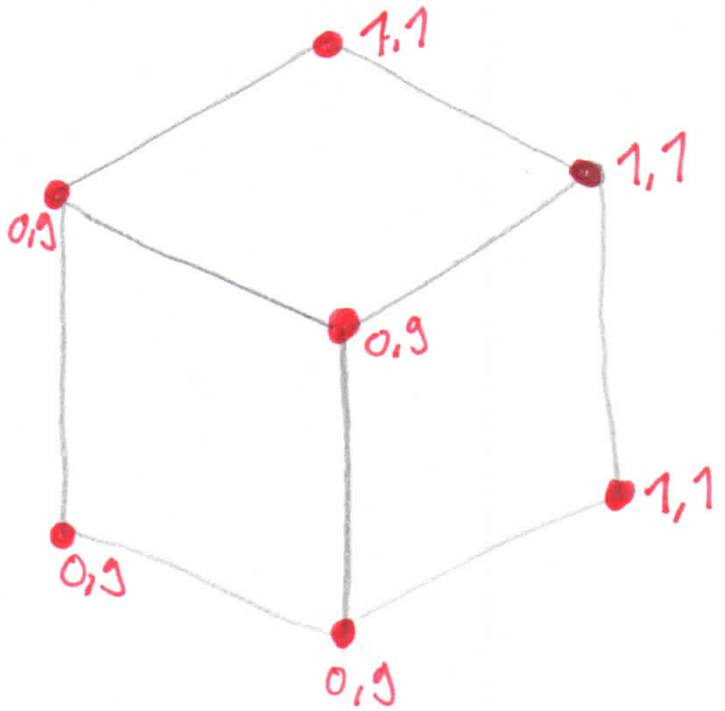

```
float calculateDensity(float x, float y, float z, float time) {
    float3 center = float3(0.5, 0.5, 0.5);
    float radius = 0.25;

    float3 radialVector = center - float3(x,y,z);
    float distance = length(radialVector);
    float density = -distance + 1.1;
    return density;
}
```

Split the space in subsections



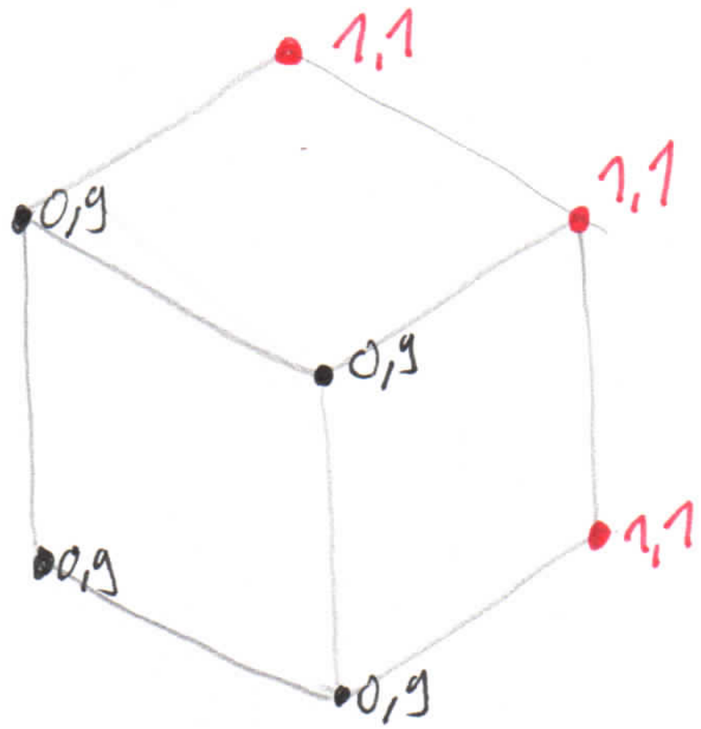
Each cube will be calculated by a thread on the gpu in parallel

For each cube (subsection)



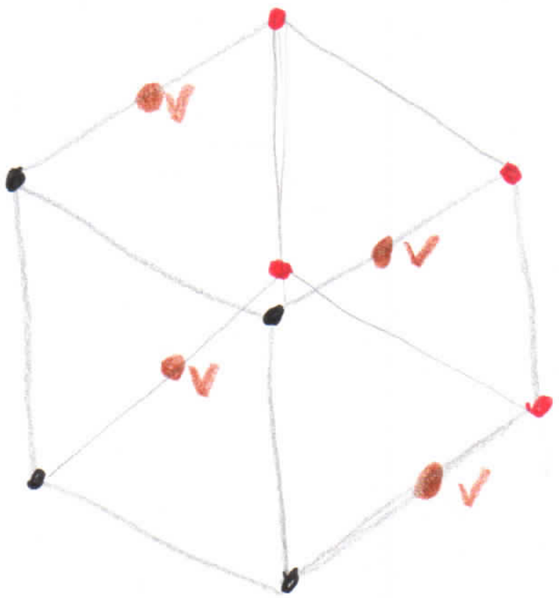
calculate the density of its corners

For each cube (subsection)



if density is higher than 1 the point is inside the mesh

For each cube (subsection)

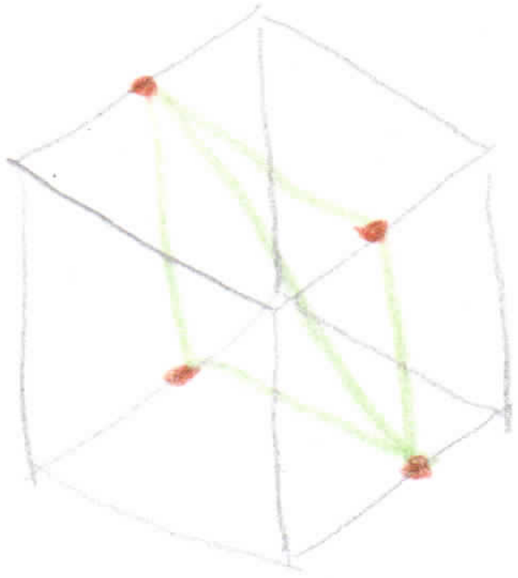


Generate a vert on edges that transition between inside and outside mesh.

-> Give each vert a unique Id for later duplicate removal.

```
struct Vert {
    float3 pos;
    int index;
};
```

For each cube (subsection)

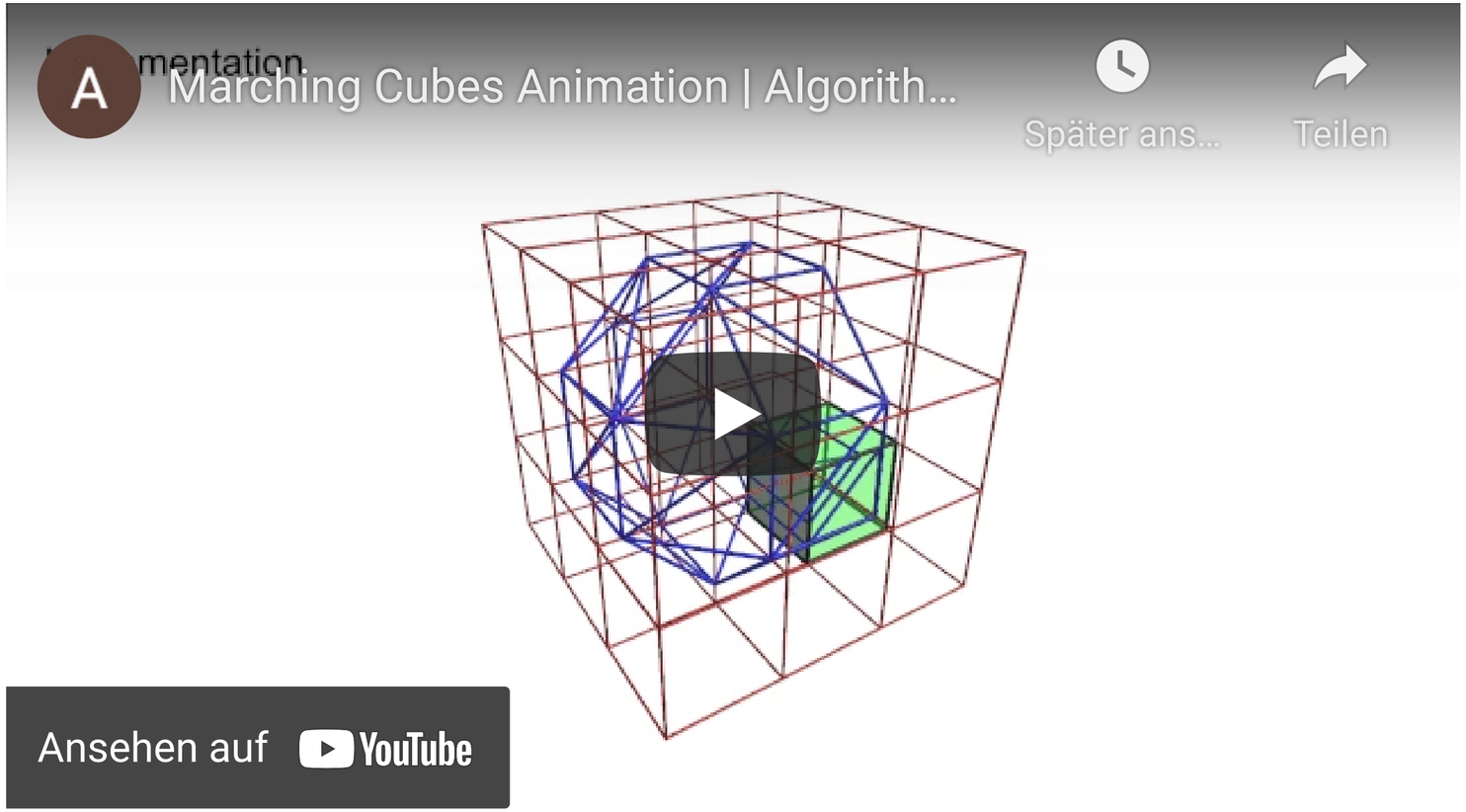


Generate faces with the triangulation lookup table

```
struct Vert {
    float3 pos;
    int index;
};

struct CubeData {
    int infos;
    Vert verts[12];
    int3 faces[5];
};
```

On CPU remove duplicate verts

Marching Cubes



igl ue4