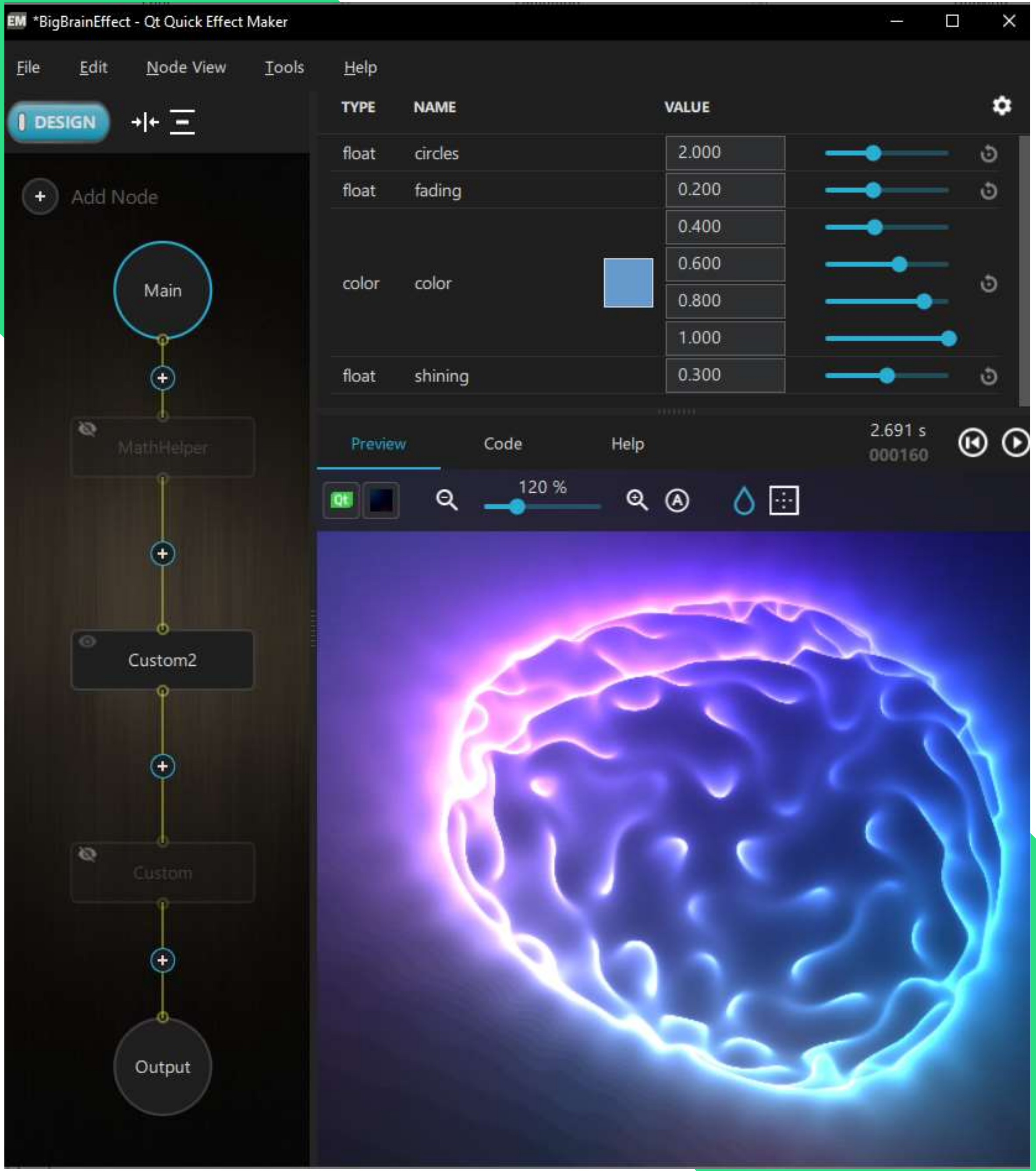
Qt Quick Effect Maker

The Basics and Advanced Usage

Qt Contributors Summit 2023

30.11.2023 *Kaj.Gronholm@qt.io*

Qt Development



TYPE	NAME	VALUE
float	circles	2.000
float	fading	0.200
color	color	0.400
		0.600
		0.800
		1.000
float	shining	0.300

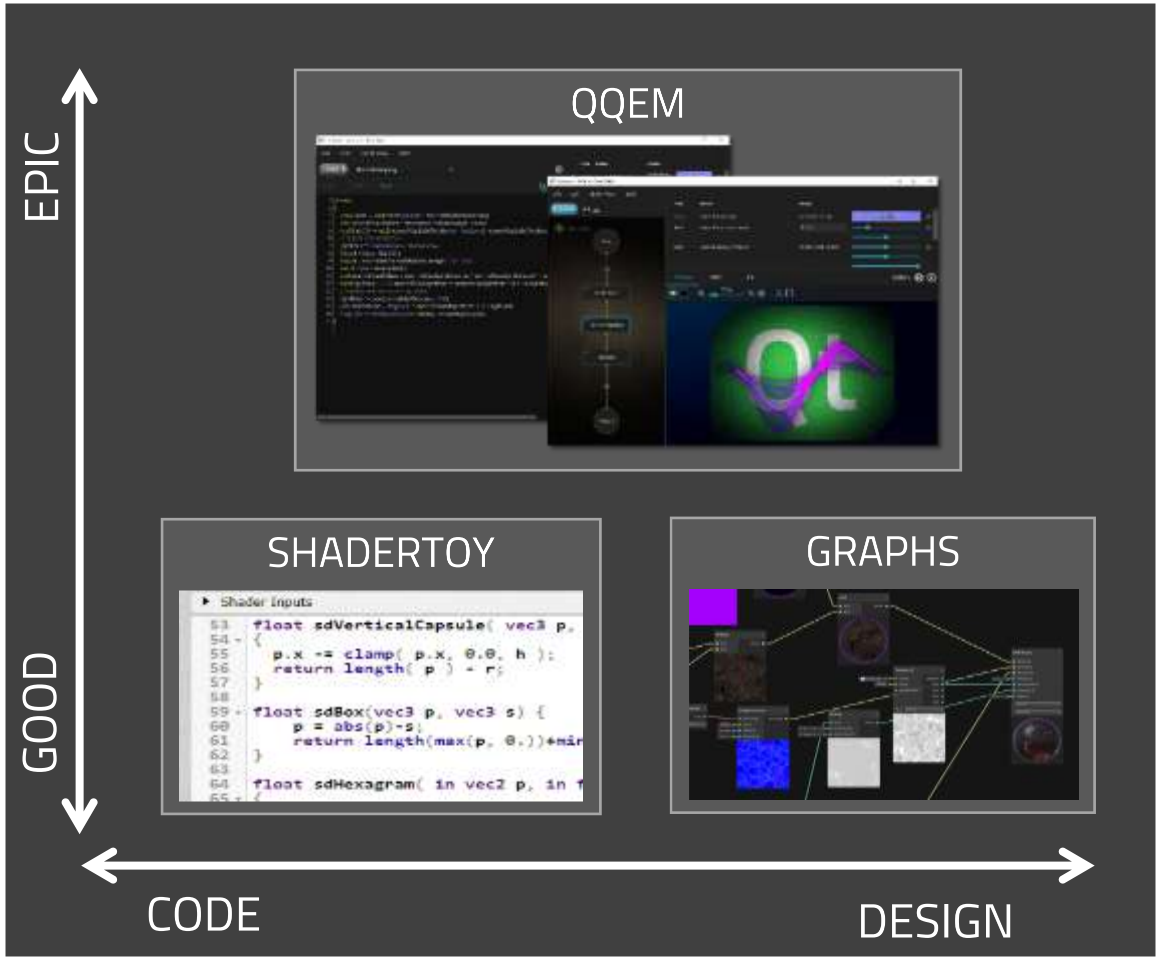
Presentation Agenda

1. Introduction into QQEM.
2. Going through the QQEM UI parts and the basic usage.
3. Porting an effect from Shadertoy into your Qt Quick application using QQEM.
4. Some public examples where QQEM has already been used.

What is Qt Quick Effect Maker?

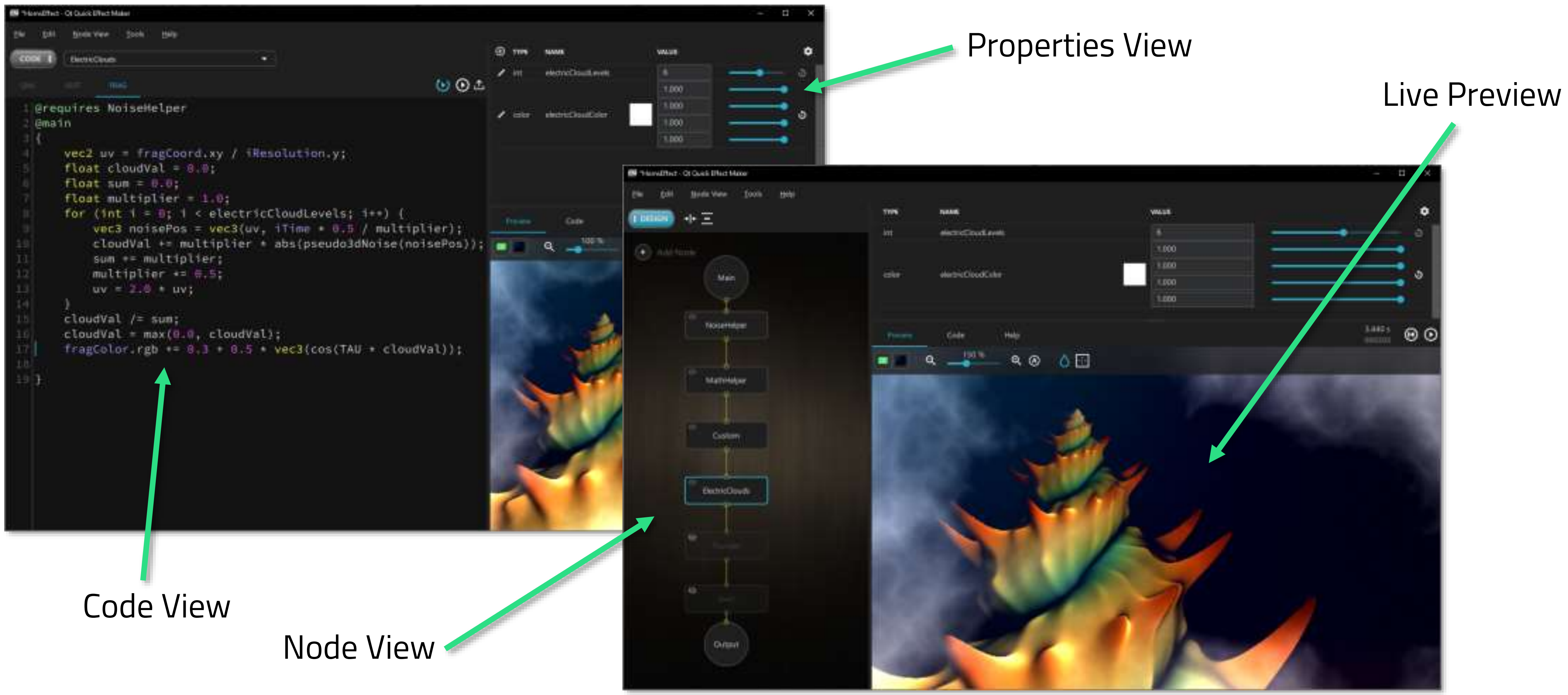
- Tool for creating **custom shader effects**.
- Specifically made with and for **Qt Quick**.
- Together with **MultiEffect**, replacing usage of Qt Graphical Effects in Qt 6.
- Contains **30+ effect nodes** which can be used as-is or customized freely.
- **Hybrid Editor**, suitable for developers and designers *)

*) Full power available only if willing to do some GLSL code editing 😉



```
float sdVerticalCapsule( vec3 p,
{
  p.x -= clamp( p.x, 0.0, h );
  return length( p ) - r;
}

float sdBox(vec3 p, vec3 s) {
    p = abs(p)-s;
    return length(max(p, 0.))+min
}

float sdHexagram( in vec2 p, in f
{
```

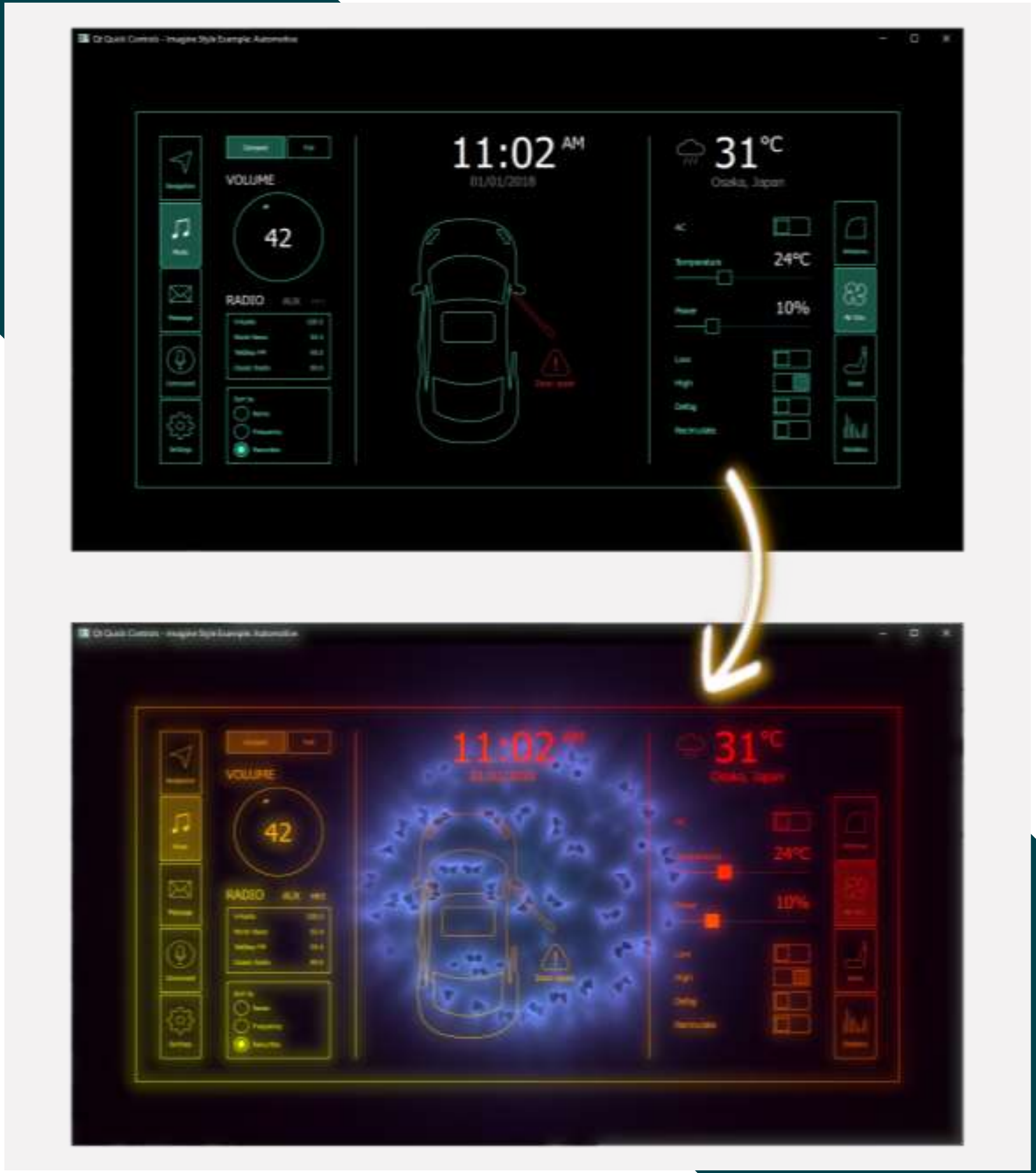
QQEM UI and Basic Usage

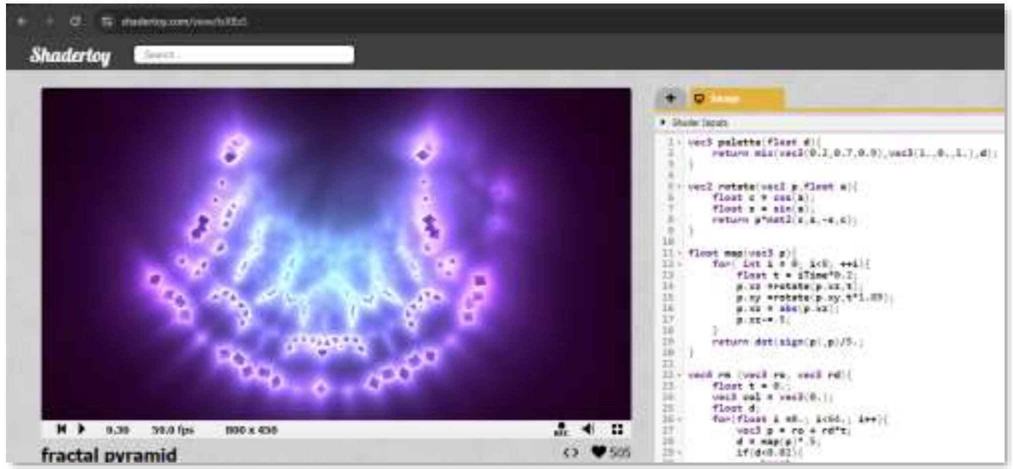


```
@requires NoiseHelper
@main
{
    vec2 uv = fragCoord.xy / iResolution.y;
    float cloudVal = 0.0;
    float sum = 0.0;
    float multiplier = 1.0;
    for (int i = 0; i < electricCloudLevels; i++) {
        vec3 noisePos = vec3(uv, iTime * 0.5 / multiplier);
        cloudVal += multiplier * abs(pseudo3dNoise(noisePos));
        sum += multiplier;
        multiplier *= 0.5;
        uv = 2.0 * uv;
    }
    cloudVal /= sum;
    cloudVal = max(0.0, cloudVal);
    fragColor.rgb += 0.3 + 0.5 * vec3(cos(TAU * cloudVal));

}
```

Example: Adding Bling Into Qt Quick Application

- Qt contains "automotive" example which demonstrates theming capabilities of Qt Quick Controls.
- Let's make it more interesting by adding few effects using QQEM.
- Porting effects from Shadertoy is straightforward. But remember the license!
  - https://www.shadertoy.com/view/tsXBzS

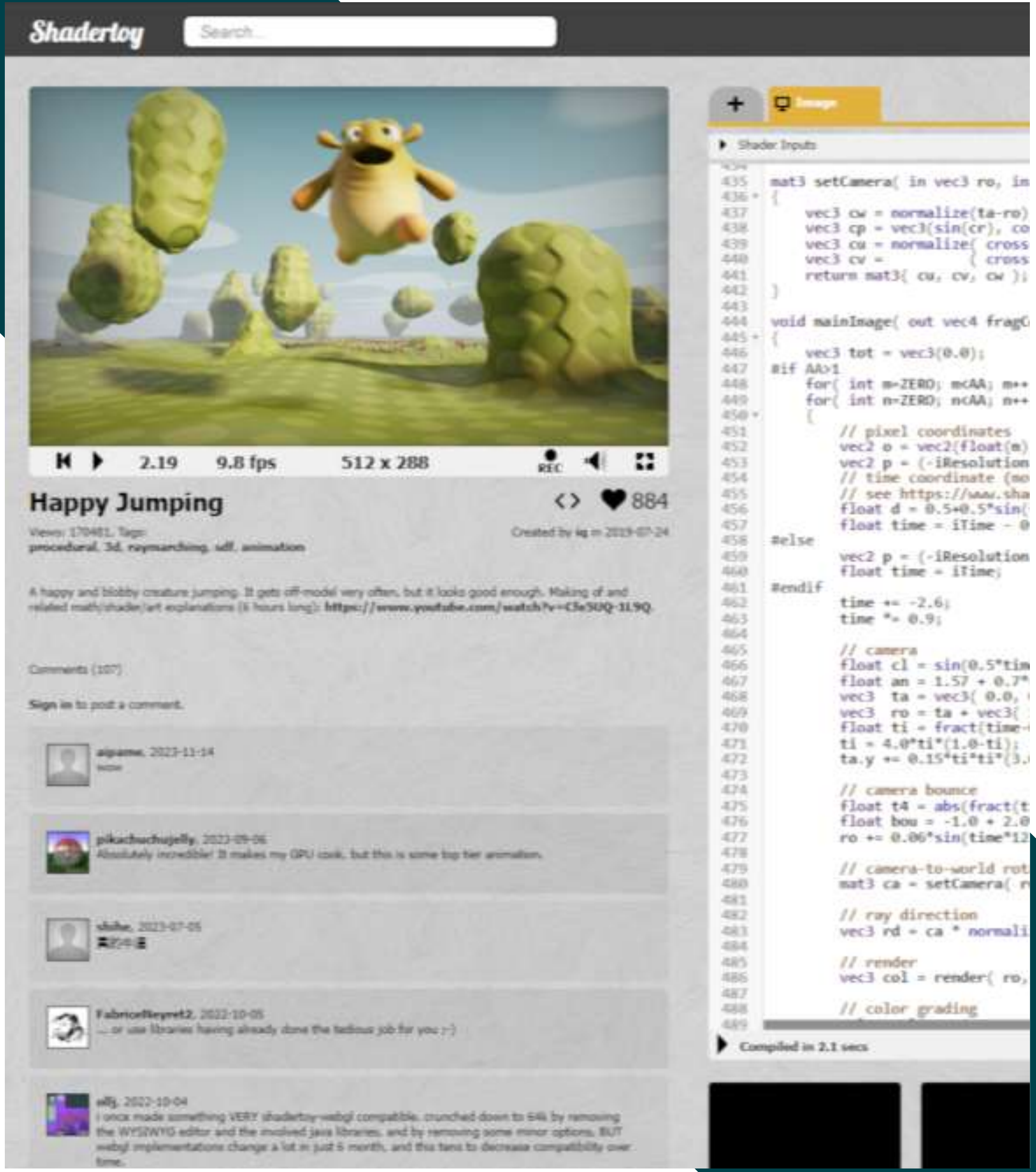




```
vec3 palette(float d){
    return mix(vec3(0.2,0.7,0.9),vec3(1.,0.,1.),d);
}

vec2 rotate(vec2 p,float a){
    float c = cos(a);
    float s = sin(a);
    return p*mat2(c,s,-s,c);
}

float map(vec3 p){
    for( int i = 0; i<8; ++i){
        float t = iTime*0.2;
        p.xz =rotate(p.xz,t);
        p.xy =rotate(p.xy,t*1.89);
        p.xz = abs(p.xz);
        p.xz-=.5;
    }
    return dot(sign(p),p)/5.;
}

vec4 rm (vec3 ro, vec3 rd){
    float t = 0.;
    vec3 col = vec3(0.);
    float d;
    for(float i =0.; i<64.; i++){
        vec3 p = ro + rd*t;
        d = map(p)*.5;
        if(d<0.02){
```

Shadertoy Performance Considerations

- Shadertoy contains plenty of awesome shaders to learn from. But as the name says, it is kinda "toy".
- Shadertoy supports only fragment shaders. With QQEM, you should utilize also vertex shaders and precalculate uniform values in QML/C++ side.
- Shadertoy support only built-in textures. QQEM supports custom textures and using them can notably improve the performance.
- Use QQEM properties for easier live editing and API for the effects.
- QQEM doesn't support all Shadertoy features (multipass, audio input, cubemaps).



Happy Jumping

Views: 170481, Tags: procedural, 3d, raymarching, sdf, animation

Created by iq in 2019-07-24

A happy and blobby creature jumping. It gets off-model very often, but it looks good enough. Making of and related math/shader/art explanations (6 hours long): https://www.youtube.com/watch?v=Cfe5UQ-1L9Q

Comments (107)

Sign in to post a comment.

aipame, 2023-11-14
wow

pikachuchujelly, 2023-09-06
Absolutely incredible! It makes my GPU cook, but this is some top tier animation.

shihe, 2023-07-05
[illegible]

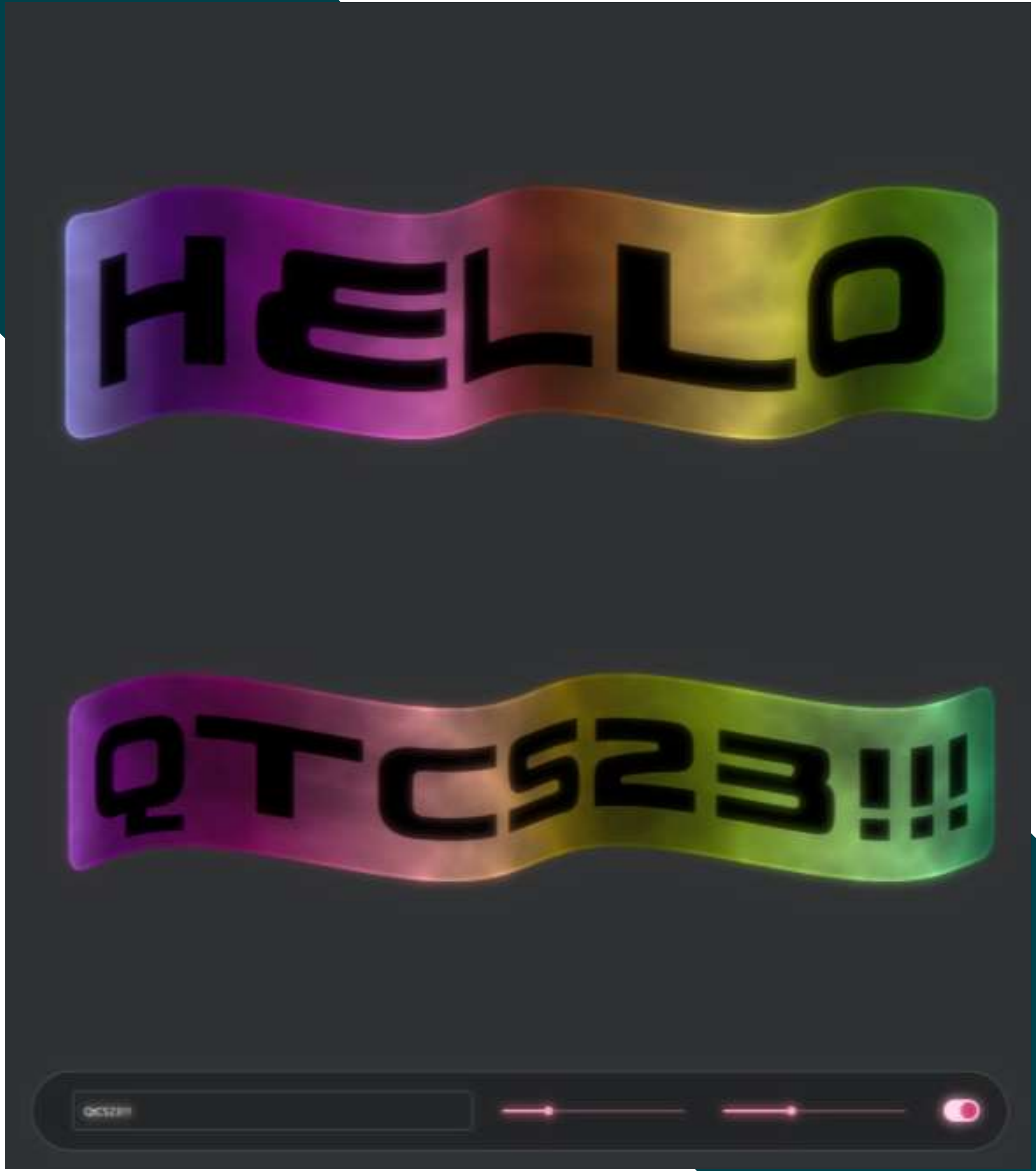
FabriceNeyret2, 2022-10-05
... or use libraries having already done the tedious job for you ;-)

elfj, 2022-10-04
i once made something VERY shadertoy-webgl compatible, crunched down to 64k by removing the WYSIWYG editor and the involved java libraries, and by removing some minor options. BUT webgl implementations change a lot in just 6 month, and this tens to decrease compatibility over time.

```
mat3 setCamera( in vec3 ro, in
{
    vec3 cw = normalize(ta-ro)
    vec3 cp = vec3(sin(cr), co
    vec3 cu = normalize( cross
    vec3 cv =          ( cross
    return mat3( cu, cv, cw );
}

void mainImage( out vec4 fragC
{
    vec3 tot = vec3(0.0);
#if AA>1
    for( int m=ZERO; m<AA; m++
    for( int n=ZERO; n<AA; n++
    {
        // pixel coordinates
        vec2 o = vec2(float(m)
        vec2 p = (-iResolution
        // time coordinate (mo
        // see https://www.sha
        float d = 0.5+0.5*sin(
        float time = iTime - 0
#else
        vec2 p = (-iResolution
        float time = iTime;
#endif
        time += -2.6;
        time *= 0.9;

        // camera
        float cl = sin(0.5*tim
        float an = 1.57 + 0.7*
        vec3  ta = vec3( 0.0, 
        vec3  ro = ta + vec3( 
        float ti = fract(time-
        ti = 4.0*ti*(1.0-ti);
        ta.y += 0.15*ti*ti*(3.

        // camera bounce
        float t4 = abs(fract(t
        float bou = -1.0 + 2.0
        ro += 0.06*sin(time*12

        // camera-to-world rot
        mat3 ca = setCamera( r

        // ray direction
        vec3 rd = ca * normali

        // render
        vec3 col = render( ro,

        // color grading
```

Compiled in 2.1 secs

Usage Examples



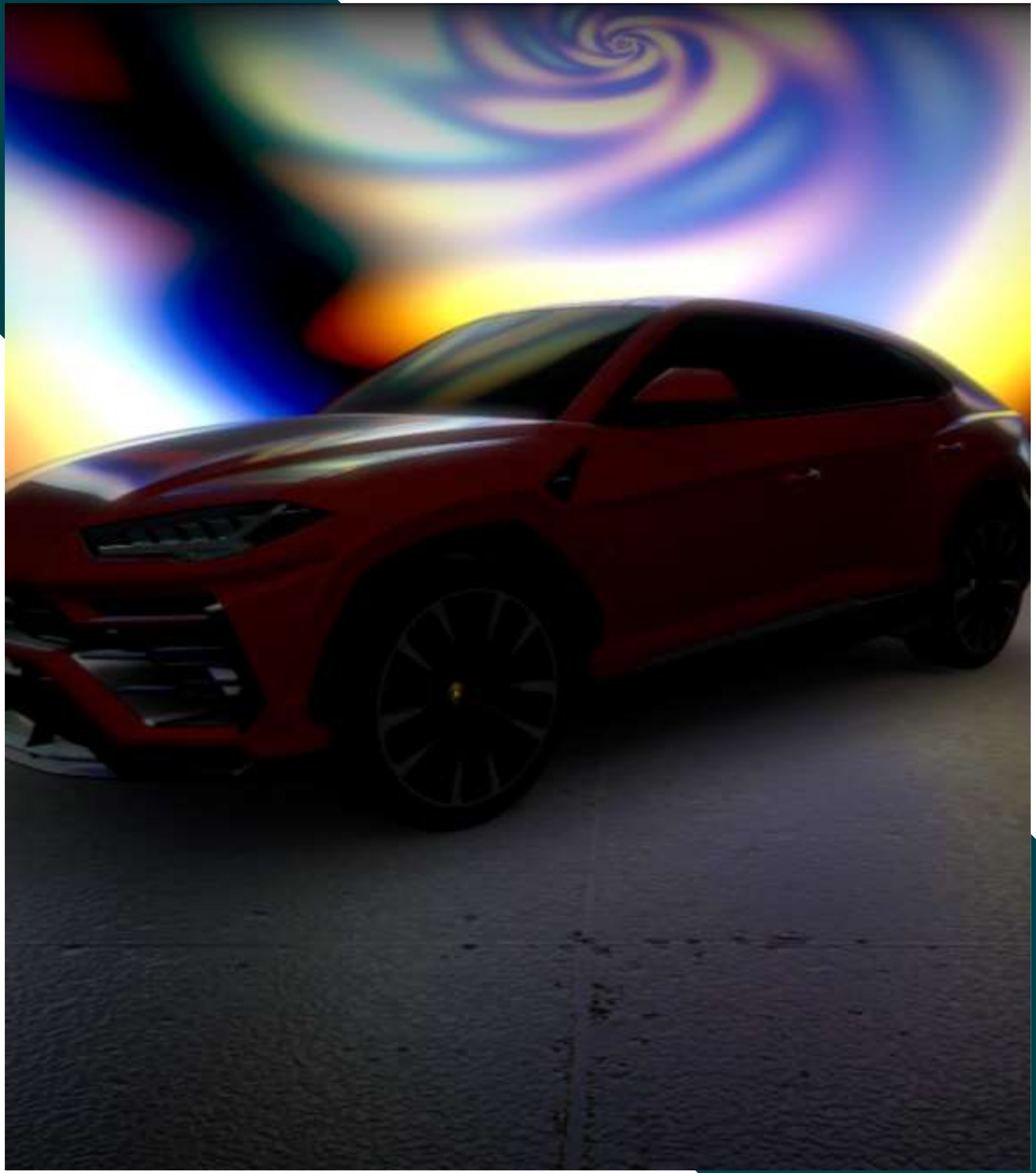
Case: Wiggly

- Qt had old example called “Wiggly” using widgets: QLineEdit, QVBoxLayout, QFontMetrics etc.
- In big examples rework it was removed, so we re-implement something similar using modern Qt Quick technologies.
- Documentation of the example: https://doc.qt.io/qt-6/qtquickeffectmaker-wiggly-example.html

Waveshaper - Exploratio...
Flashbacks
PYLOT
2:11
-3:16
SCANDR

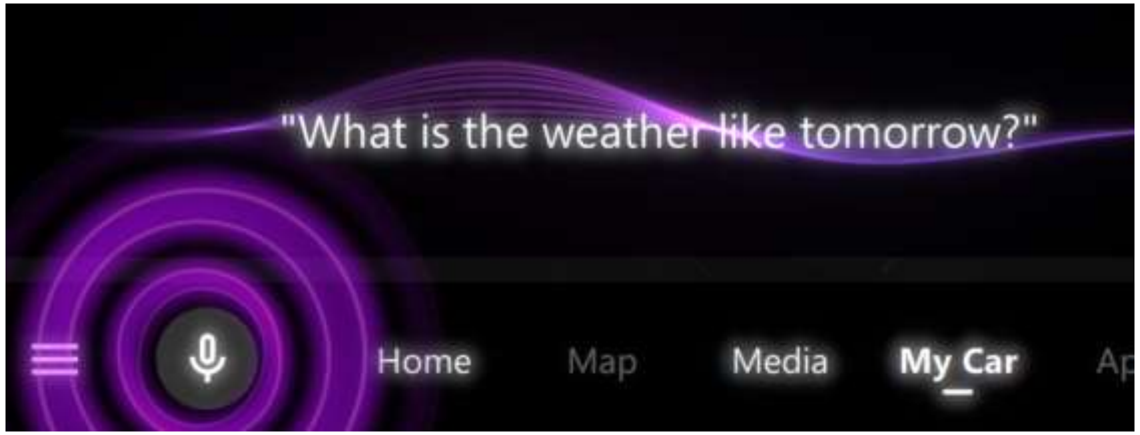
Case: Smart Watch Demo

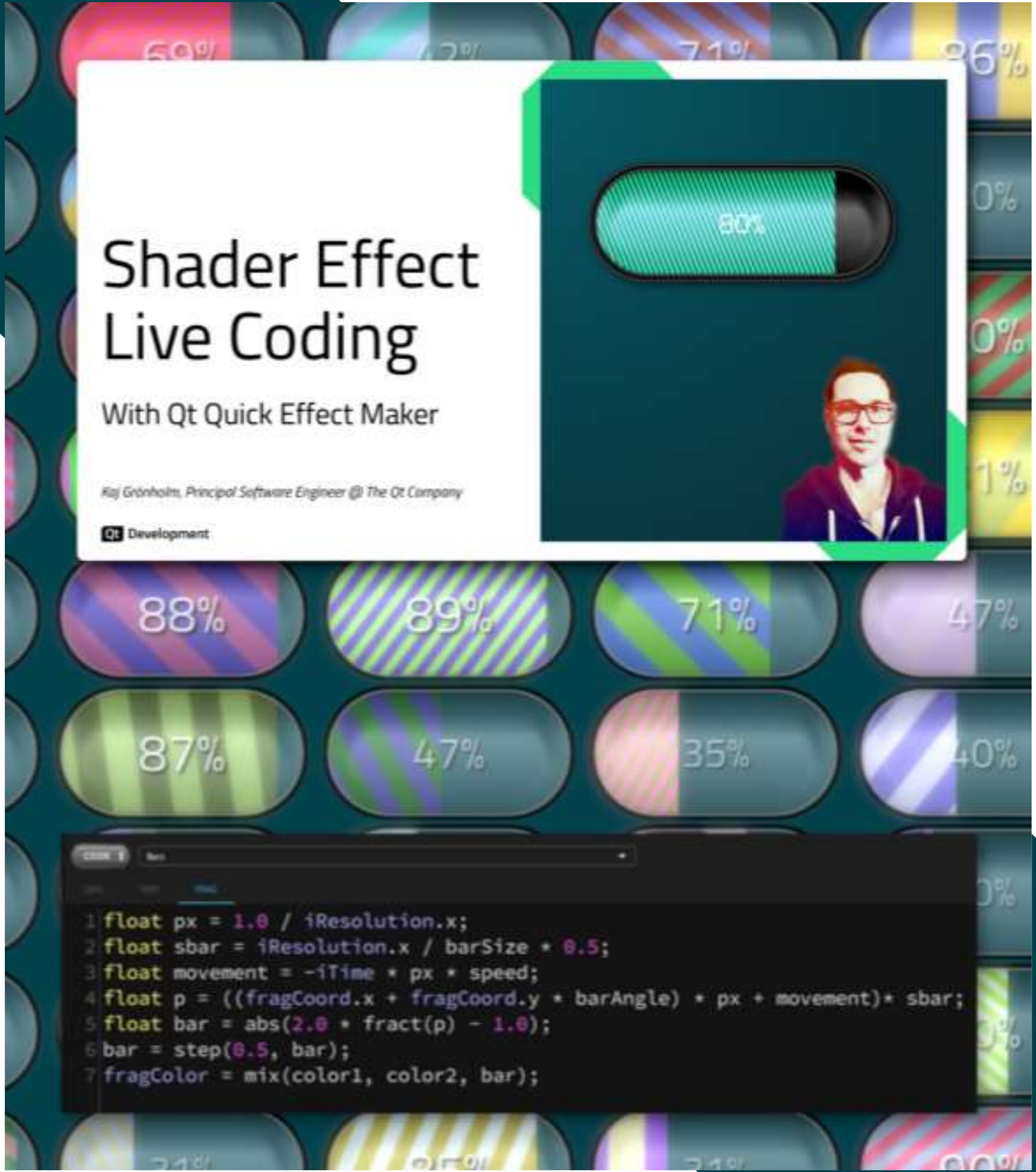
- This demo was used as a showcase of the Qt Quick Effect Maker in Qt 6.5 release.
- Demonstrates embedding Qt Quick UI (with effects) into Qt Quick 3D scene.
- Some of the effects ended up as nodes into QQEM for everyone to use: Bend, ColorLUT, Sunburst, Swirl, Clouds, SeaReflection.
- Easter egg added as a Qt Hackathon project.



Case: Car Configurator & Outrun Demos

- In the Qt Car Configurator demo, background effects which get reflected into 3D scene are implemented as QQEM effects.
- Qt automotive Outrun IVI Demo uses effects created with QQEM in few places, like the virtual assistant effects seen below.





Shader Effect
Live Coding

With Qt Quick Effect Maker

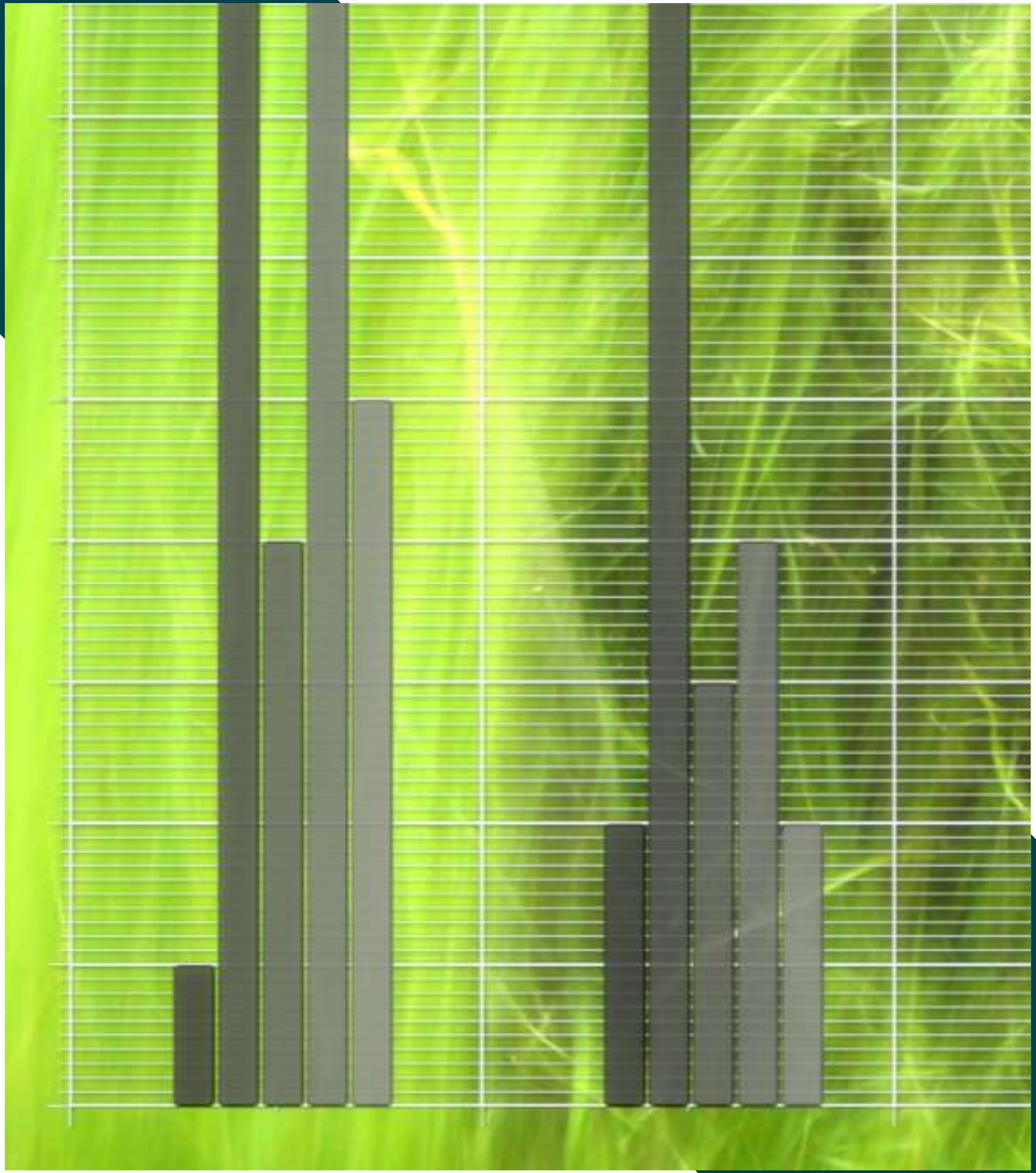
Kaj Grönholm, Principal Software Engineer @ The Qt Company

Qt Development

```
float px = 1.0 / iResolution.x;
float sbar = iResolution.x / barSize * 0.5;
float movement = -iTime * px * speed;
float p = ((fragCoord.x + fragCoord.y * barAngle) * px + movement)* sbar;
float bar = abs(2.0 * fract(p) - 1.0);
bar = step(0.5, bar);
fragColor = mix(color1, color2, bar);
```

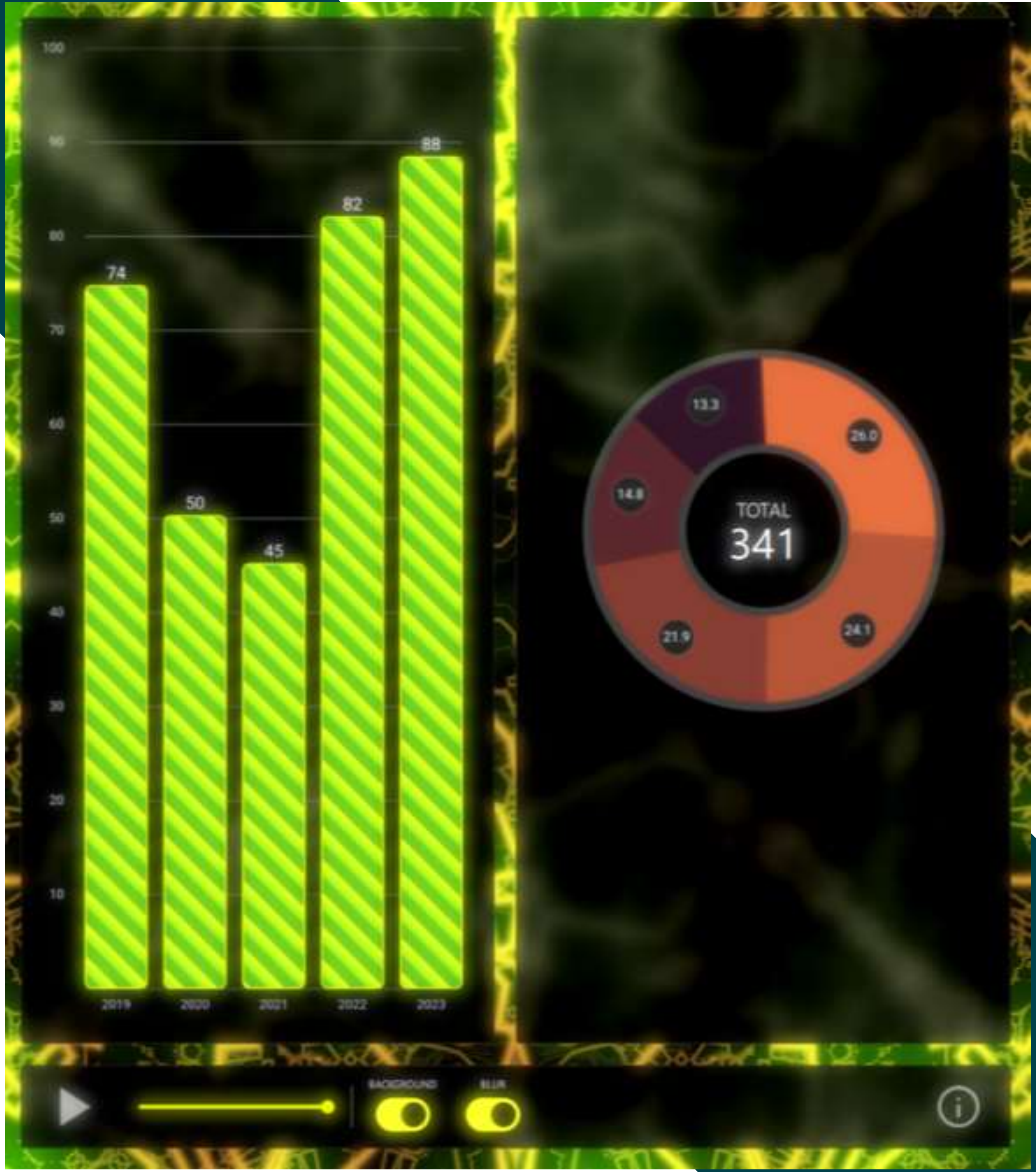
Case: Bars Effect

- Live shader coding using QQEM
  https://youtu.be/gKnb5LNOsMo
- Bars effect with customizable bars colors, width, angle, antialiasing, animation speed etc. Smooth rendering and high performance.
- I have seen similar bars implemented with more Designer-oriented tools, with less customization and performance.
- Can be used with different Qt Quick items, here in example with Quick Controls Progressbar and Slider components.



Case: QtGraphs Axis Grid

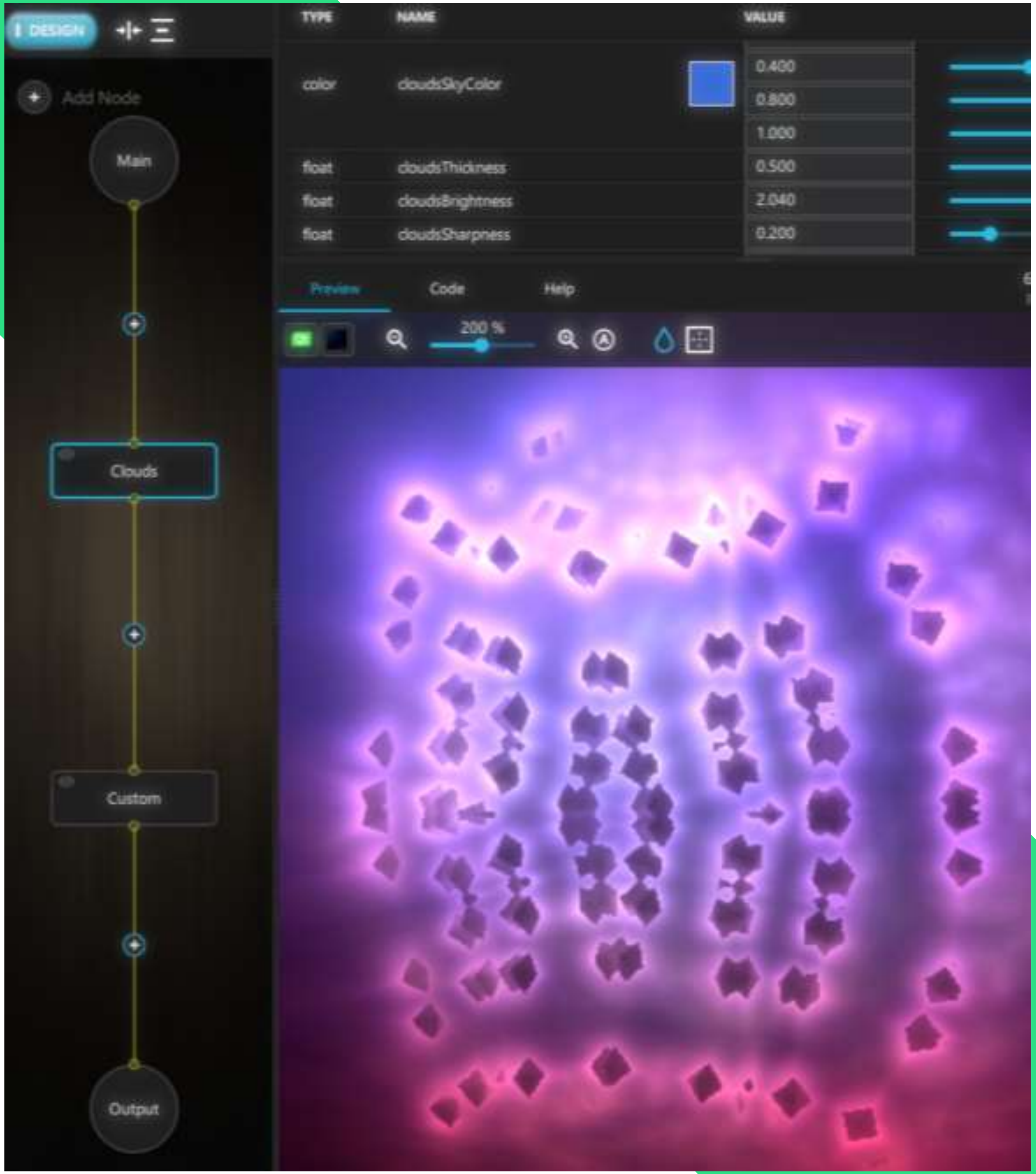
- For the new work-in-progress Qt Graphs 2D, we implemented axis grid and tickers with QQEM.
- Pros of this approach:
  - Development was fast and API was easy to build and prototype.
  - Performance remains stable even with large number of lines, no increase in vertex count.
  - Freely adjustable antialiasing amount, which is nice especially with thin lines.
  - This antialiasing also makes it possible to implement drop shadow without a costly blur effect as seen in the screenshot.



Case: Fun Graph

- Shadertoy effect ported with QQEM was used in “FunGraph” prototype.
- Main target of the demo was to show how custom animated graphs could be done.
- Panels background blur and mask use Qt Quick MultiEffect.
- Using QNanoPainter with Qt RHI backend (OpenGL, Vulkan, Metal, Direct3D).

Thanks!



TYPE	NAME	VALUE
color	cloudsSkyColor	0.400
		0.800
		1.000
float	cloudsThickness	0.500
float	cloudsBrightness	2.040
float	cloudsSharpness	0.200

Contact: *Kaj.Gronholm@qt.io*