

Flutter *SHADERS* Beyond the Gimmick

Hello there 🙌



My name is
Renan Araujo
@reNotANumber

I live in **Porto**,
OSS Contributor for **Blue Fire** & OSS Engineer at **Very Good Ventures**,
You can see my socials and stuff at **renan.gg**

renan.gg

SHADERS

renan.gg

Fragment *SHADERS*

renan.gg

Custom Fragment *SHADERS*

1. What is a Fragment Shader
2. Interpolation
3. Textures
4. Sampling
5. Go seek help



renan.gg



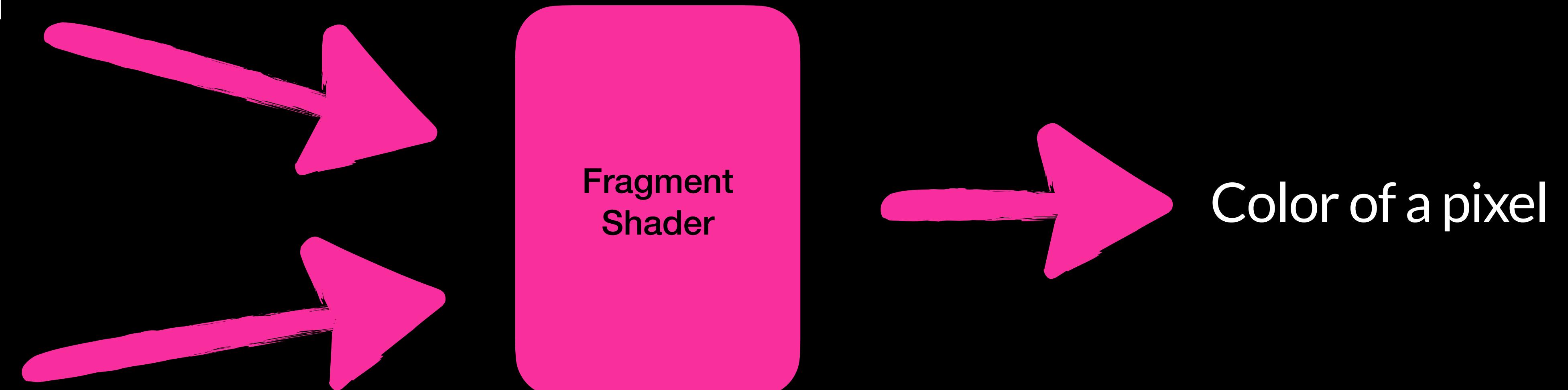
What is a Fragment Shader

What color and opacity will a pixel have?

First Fact: Answers a question

User-provided
input

Framework
provided
input



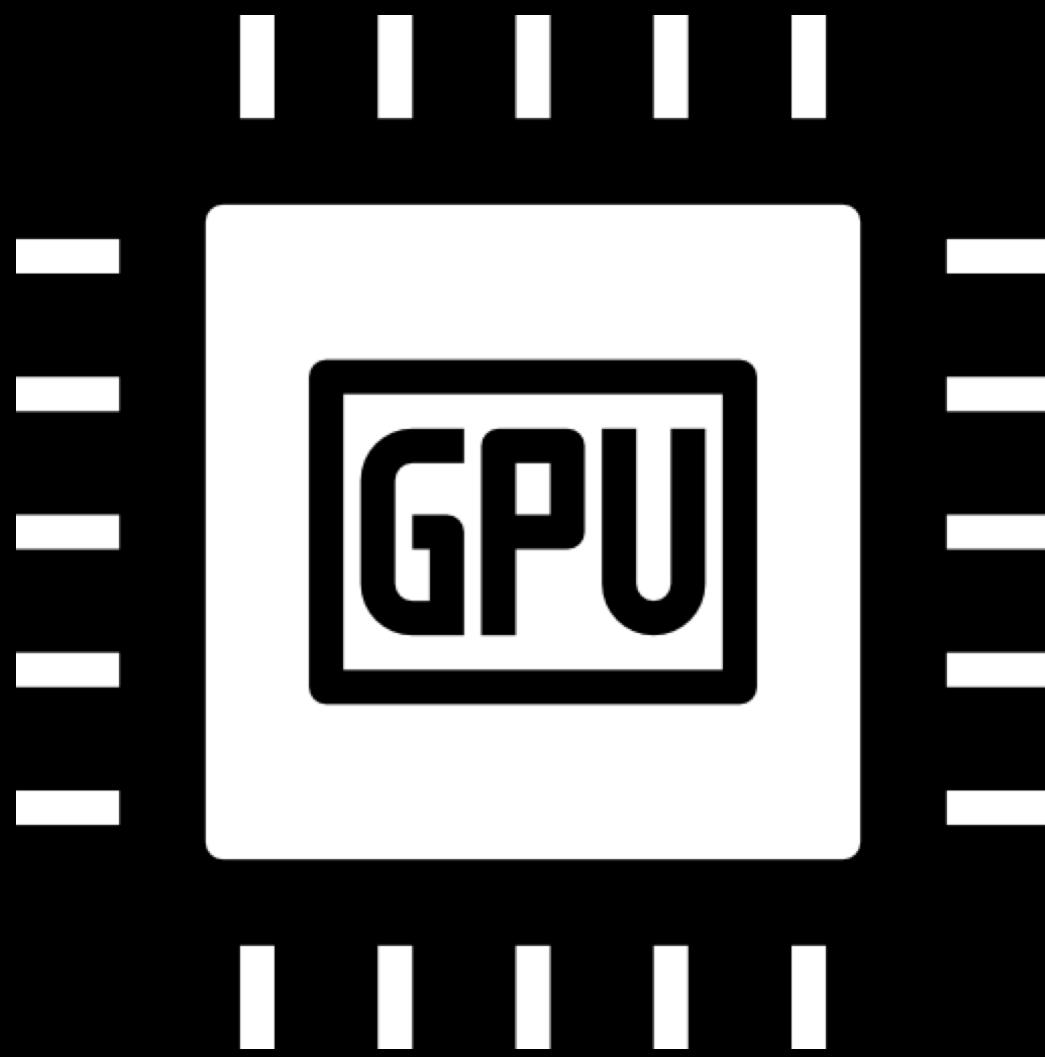
Second Fact: It works like a function

User- provided input

- Numeric inputs:
 - Sizes, coordinates, vertices, opacity, colors, transform matrices, math params...
- Texture inputs:
 - Literally some bitmaps

Framework provided input

- The Pixel coordinate



Third Fact: It runs on the GPU

```
// Specify the version of GLSL to use.  
#version 460 core  
  
// Specify the name of the fragment shader.  
precision mediump float;  
  
// Include the Flutter-provided code.  
#include <flutter/runtime_effect.glsl>  
  
// User provided inputs (uniforms)  
uniform vec2 uSize;  
uniform sampler2D tTexture;  
  
// Declare the output variable for the fragment shader.  
out vec4 fragColor;  
  
void main() {  
    fragColor = vec4(1.0, 0.0, 0.0, 1.0);  
}
```

- This is not dart: GLSL
- Mind the data types

How to add **that** to a Flutter application

```
flutter:  
  assets:  
    - assets/  
    - fonts/  
  uses-material-design: true  
  shaders:  
    - |shaders/my_shader.glsL
```

1. Add the **GLSL** file path to the pubspec

How to add **that** to a Flutter application

```
void main() {  
  
    final program = FragmentProgram.fromAsset('shaders/my_shader.glsl');  
  
    runApp(const MyApp());
```

2. Load the shader

How to add **that** to a Flutter application

```
final shader = program.fragmentShader();  
  
shader  
    ..setFloat(0, size.width)  
    ..setFloat(1, size.height);  
  
canvas.drawRect(  
    Offset.zero & size,  
    Paint()..shader = shader,  
);
```

3. Use it within a Canvas-based API

Custom
Painter

Flame Engine

Authored
Render Objects

Custom
Painter

Flame Engine

Authored
Render Objects

Using flutter_shaders can make things easier

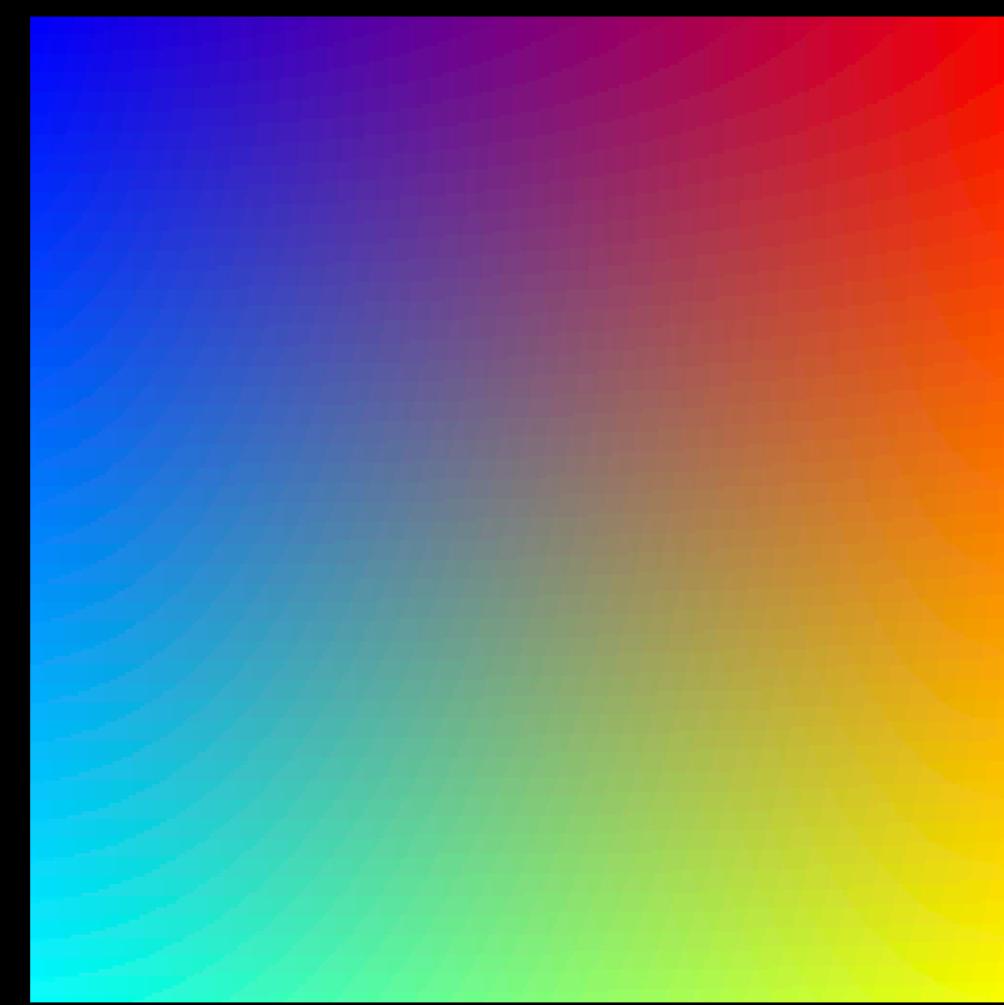
```
return ShaderBuilder(  
    assetKey: 'shaders/my_shader.glsl',  
    (context, shader, child) {  
        return CustomPaint(  
            painter: SomePainter(shader),  
        ); // CustomPaint  
    },  
); // ShaderBuilder
```

```
    shader.setFloatUniforms((s) {  
        s  
            ..setSize(size)  
            ..setFloat(0.5)  
            ..setColor(const Color(0xFF000000));  
    });
```

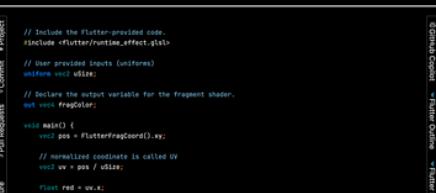

EXAMPLE 1

Our first shader

[« Go back](#)

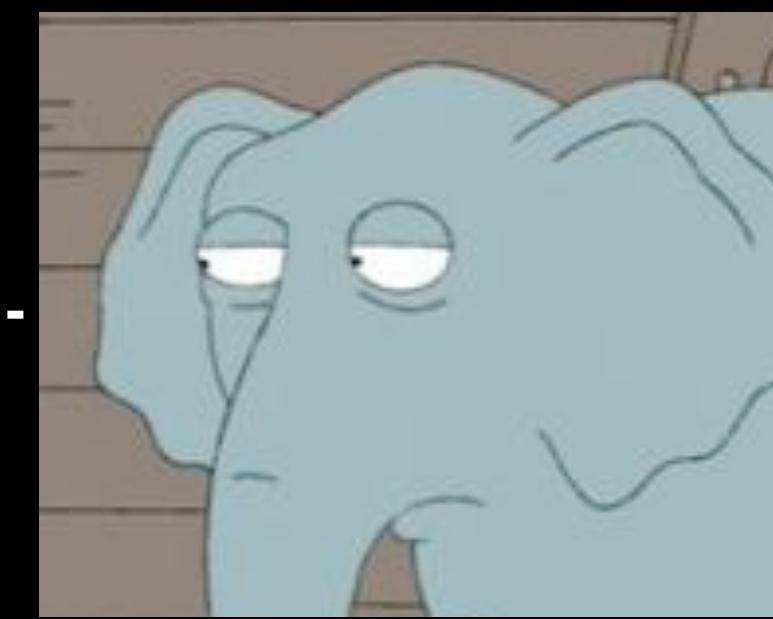


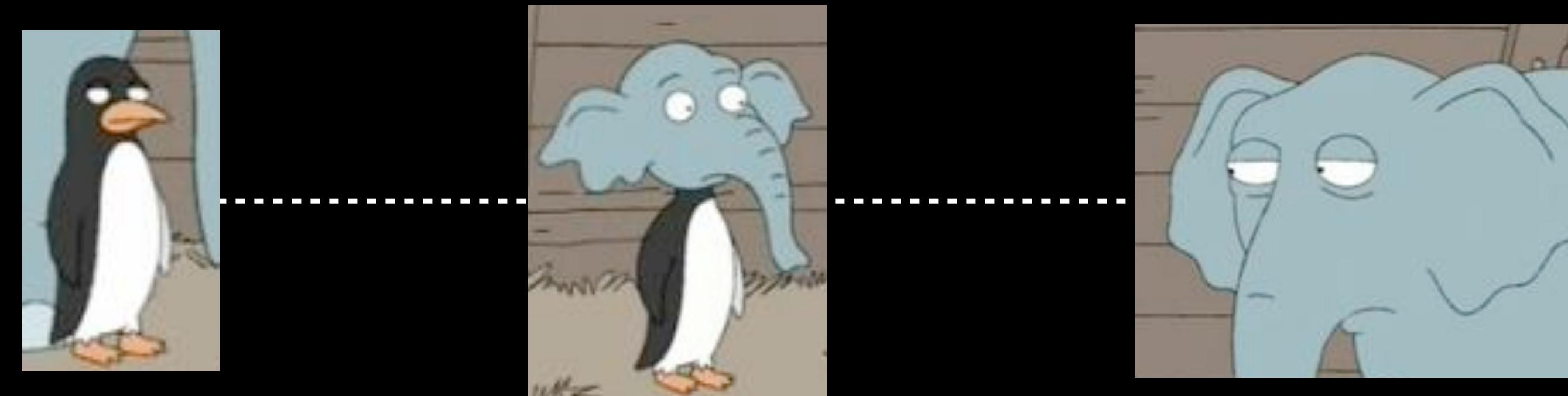
```
7  
8 // Include the Flutter-provided code.  
9 #include <flutter/runtime_effect.glsL>  
10  
11 // User provided inputs (uniforms)  
12 uniform vec2 uSize;  
13  
14 // Declare the output variable for the fragment shader.  
15 out vec4 fragColor;  
16  
17 void main() {  
18     vec2 pos = FlutterFragCoord().xy;  
19  
20     // normalized coordinate is called UV  
21     vec2 uv = pos / uSize;  
22  
23     float red = uv.x;  
24     float green = uv.y;  
25     float blue = 1.0 - red;  
26  
27     fragColor = vec4(red, green, blue, 1.0);  
28 }  
29
```



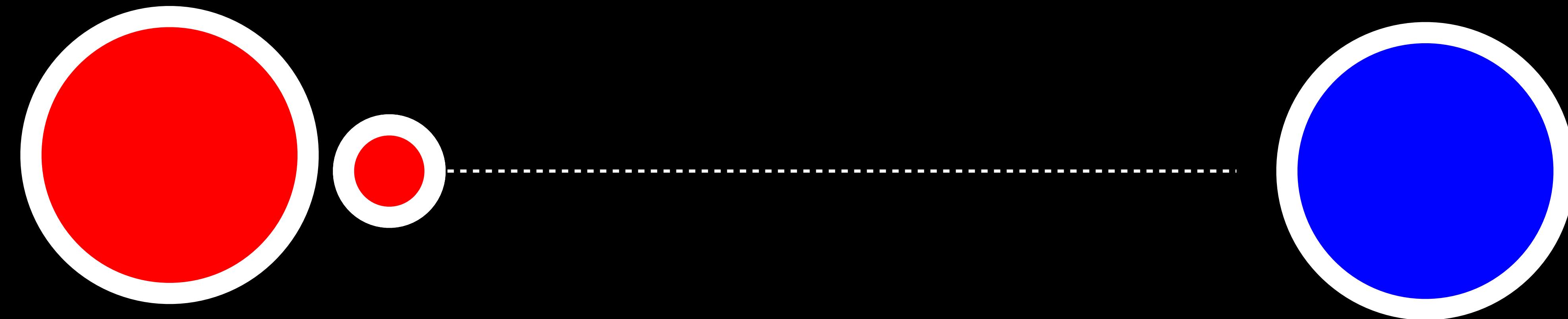
Summary

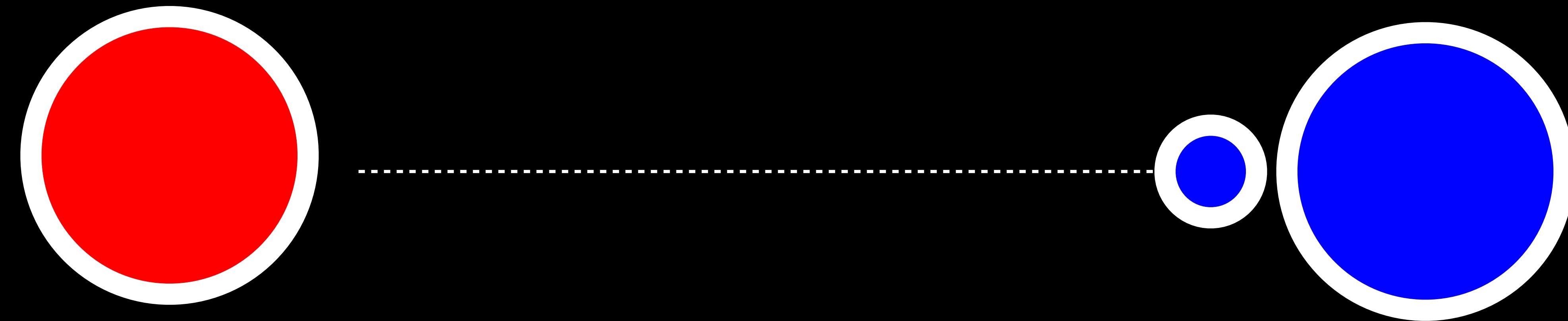
- ✓ Shaders are a straightforward way to create custom visual patterns
- ✓ It runs where graphical operations should run
- ✓ Numeric-based: the sky is the limit
- ✓ Can be combined with several math techniques to create visual effects.





Interpolation





renan.gg

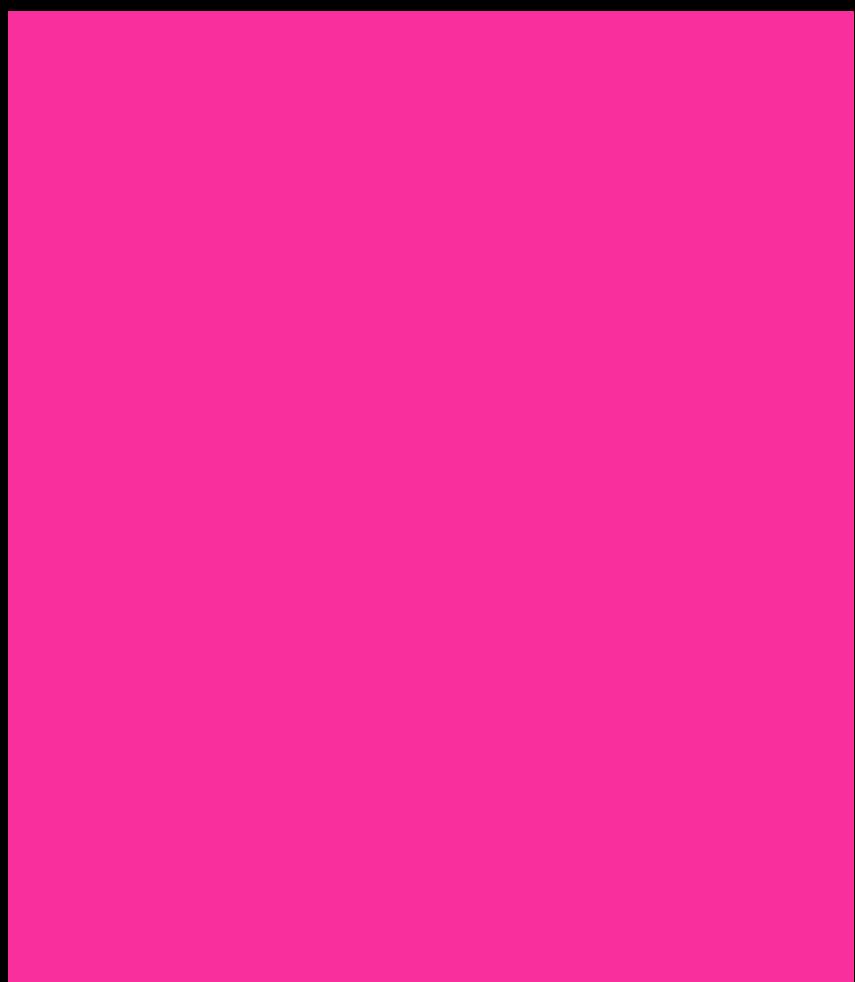


Interpolant: 0.0 to 1.0

```
double width = 100;  
double height = 120;  
  
...  
  
return AnimatedContainer(  
  duration: const Duration(milliseconds: 500),  
  width: width,  
  height: height,  
  color: Colors.renanPink,  
);
```



```
double width = 100;  
double height = 120;  
  
...  
  
return AnimatedContainer(  
    duration: const Duration(milliseconds: 500),  
    width: width,  
    height: height,  
    color: Colors.renanPink,  
);  
  
...  
  
setState(() {
```

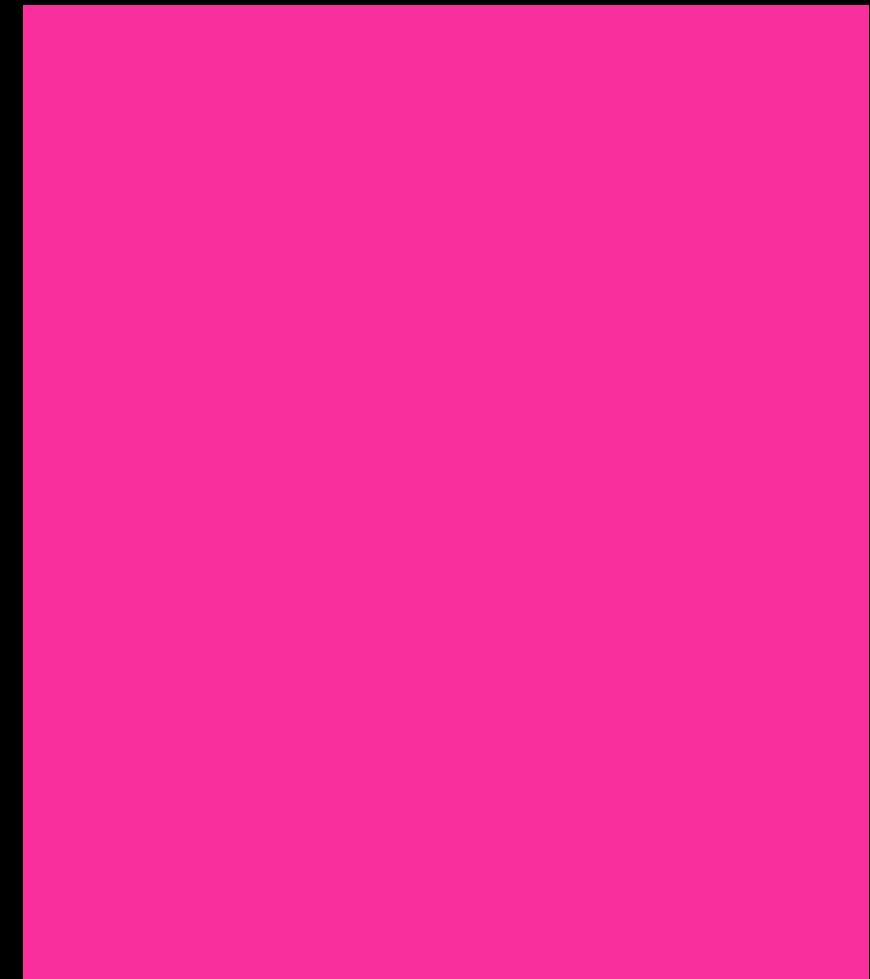


• • •

```
return AnimatedContainer(  
  duration: const Duration(milliseconds: 500),  
  width: width,  
  height: height,  
  color: Colors.renanPink,  
) ;
```

• • •

```
setState(() {  
  width = 200;  
  height = 200;  
});
```

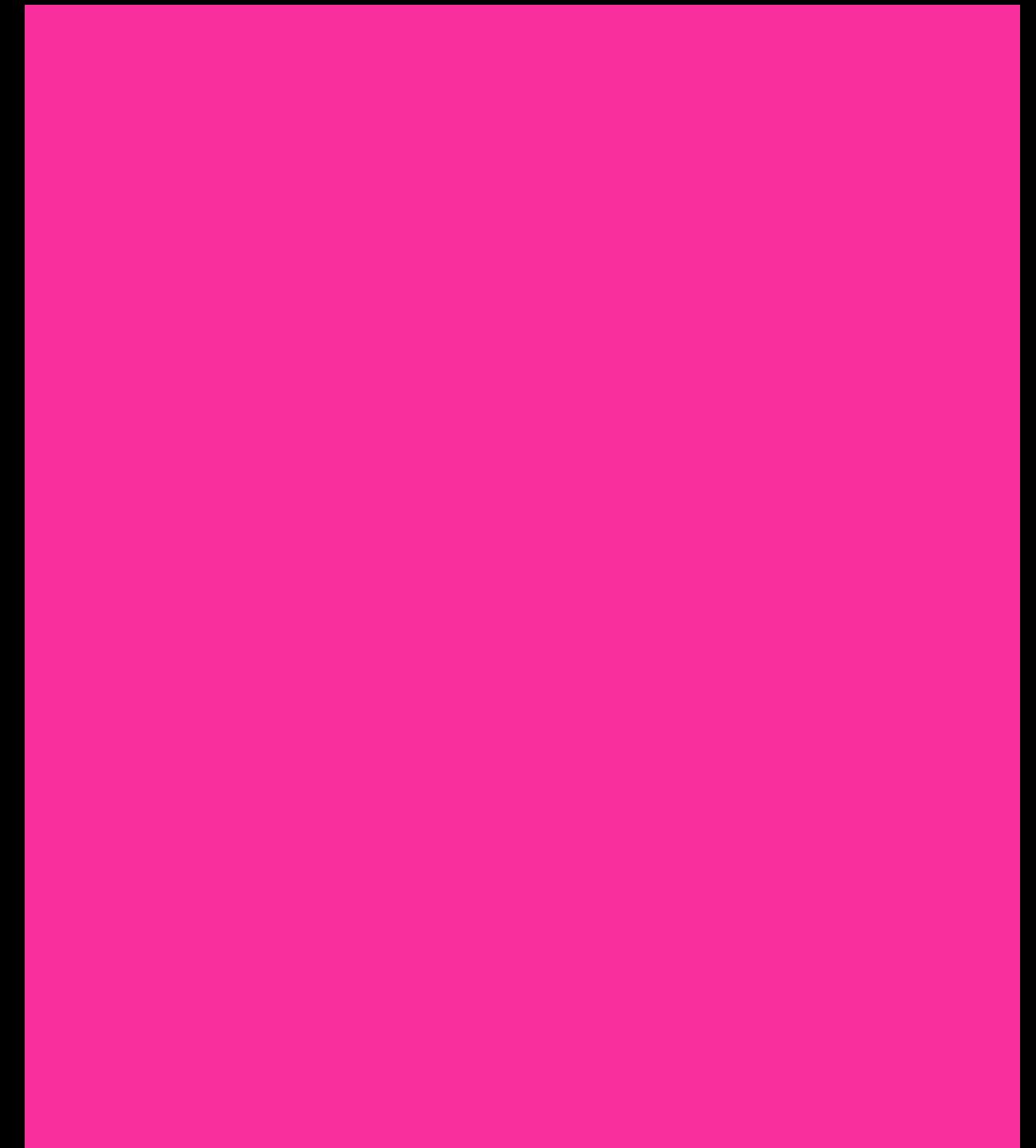


• • •

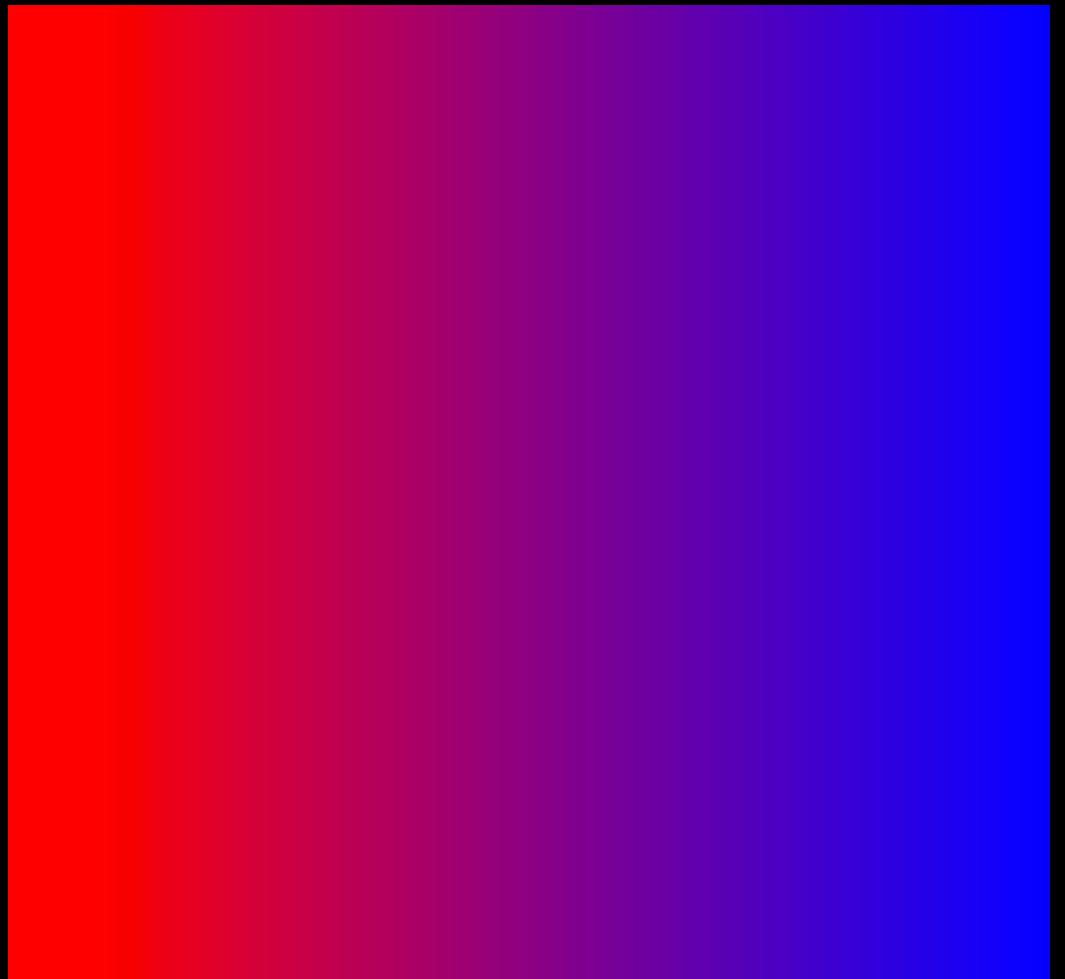
```
return AnimatedContainer(  
  duration: const Duration(milliseconds: 500),  
  width: width,  
  height: height,  
  color: Colors.renanPink,  
) ;
```

• • •

```
setState(() {  
  width = 200;  
  height = 200;  
});
```

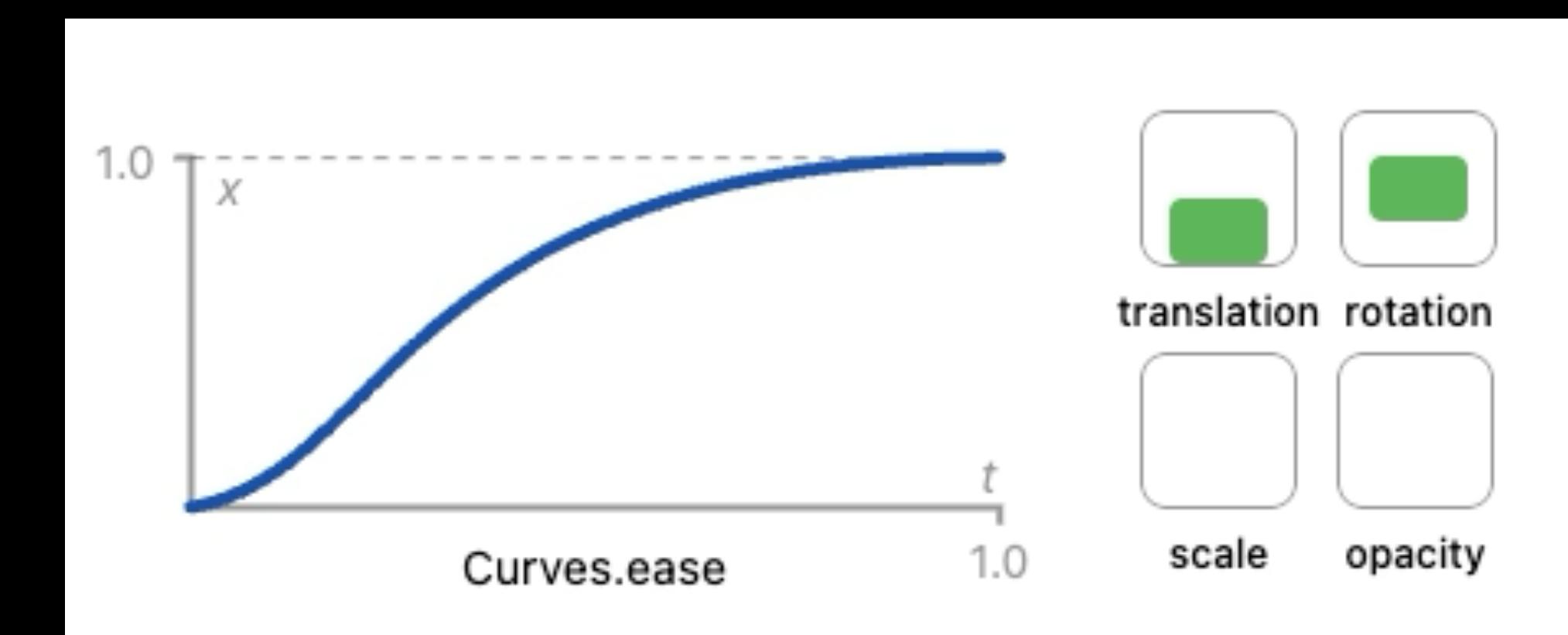
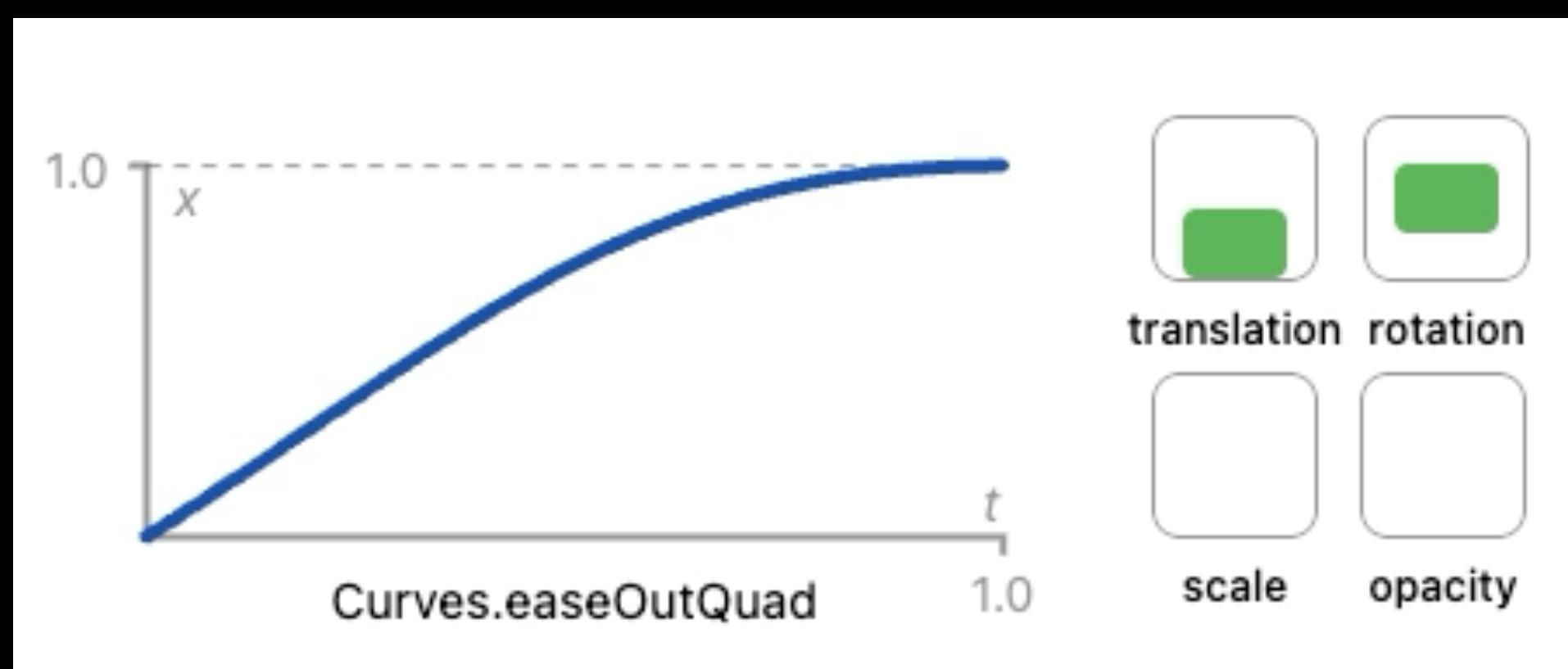
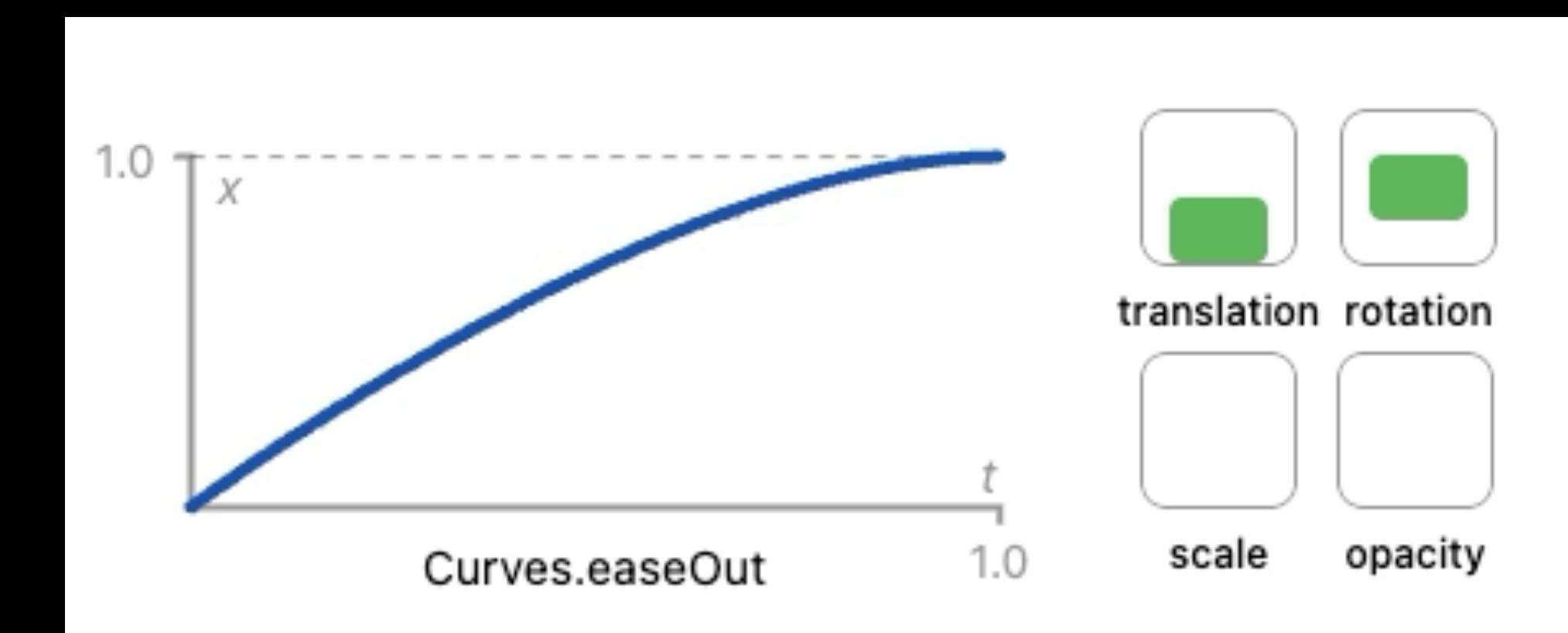
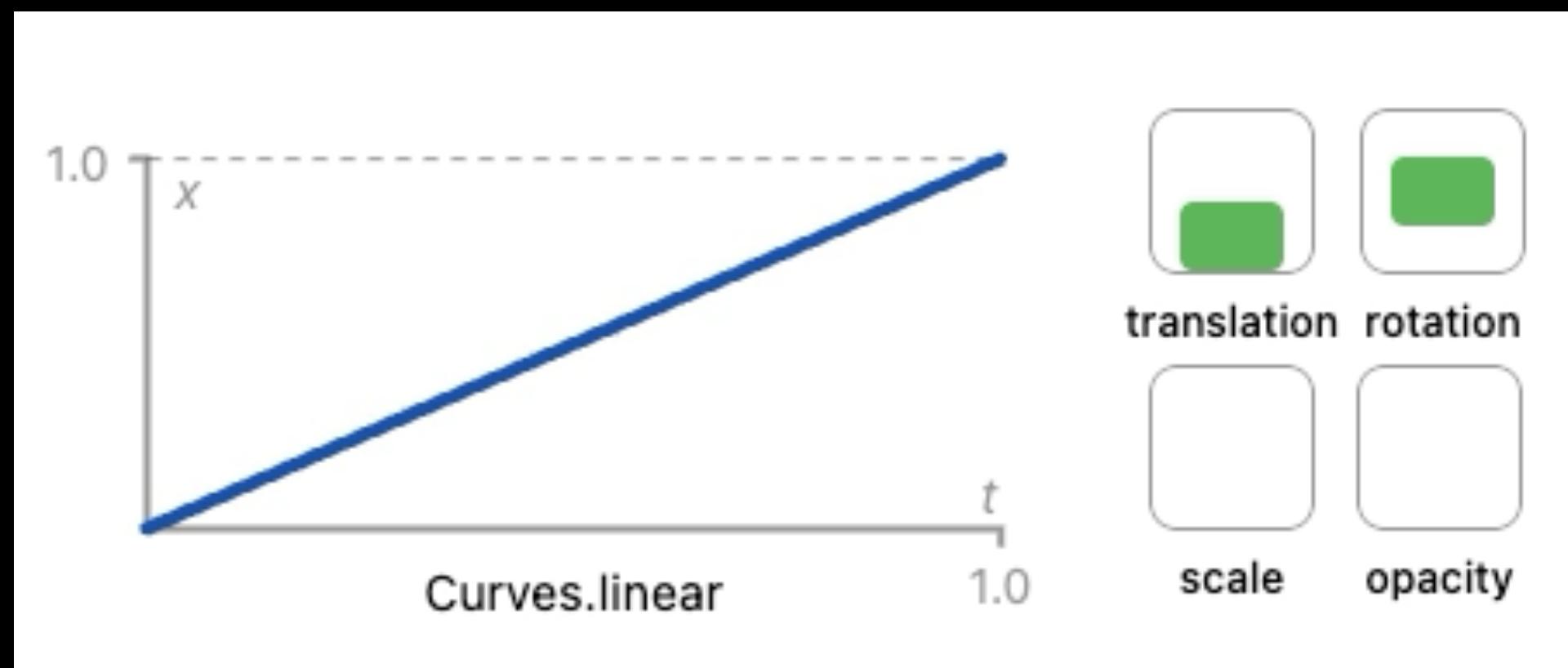


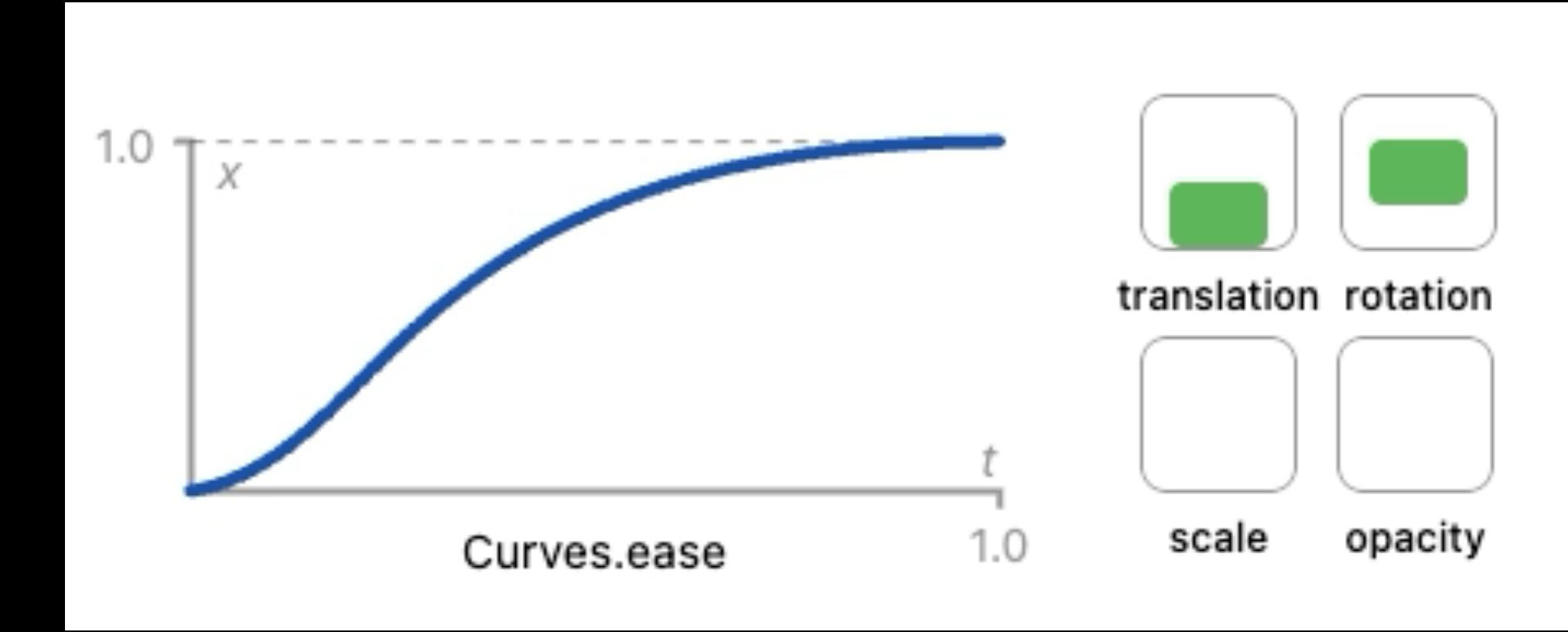
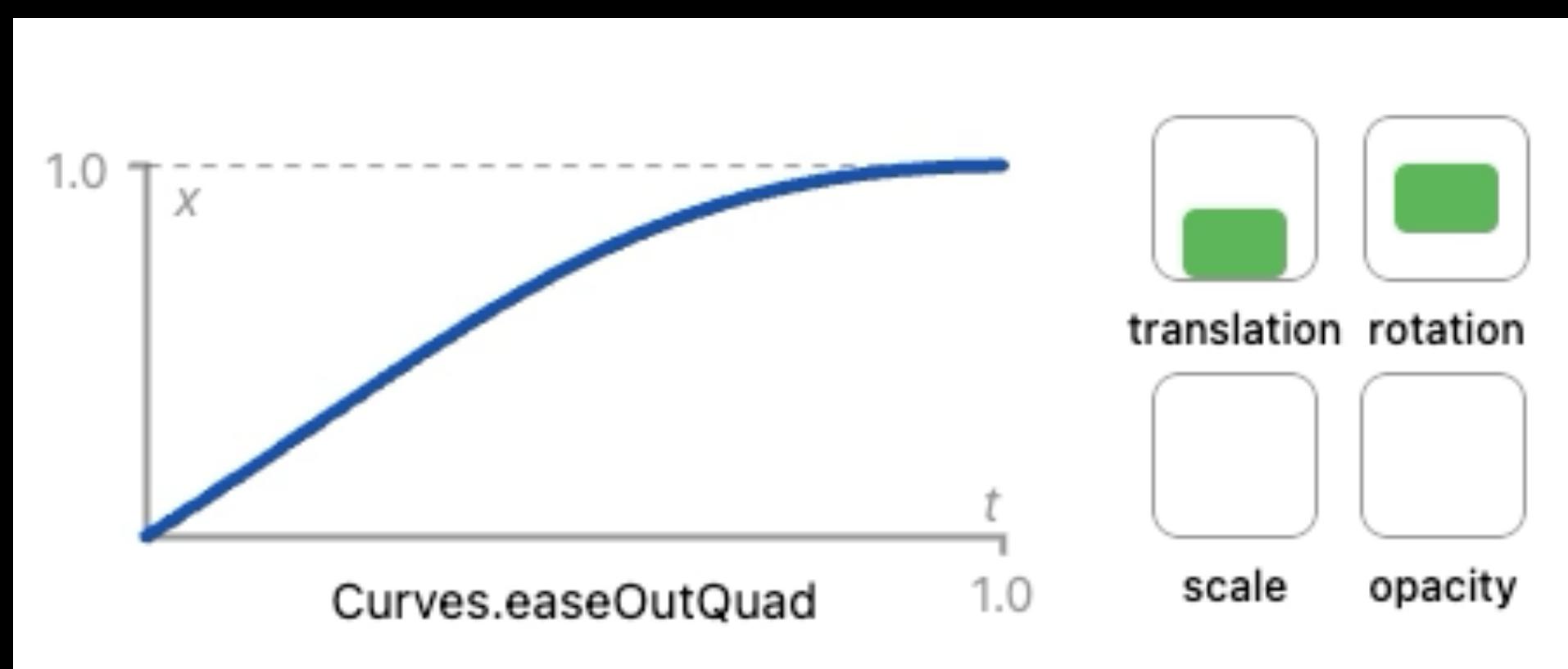
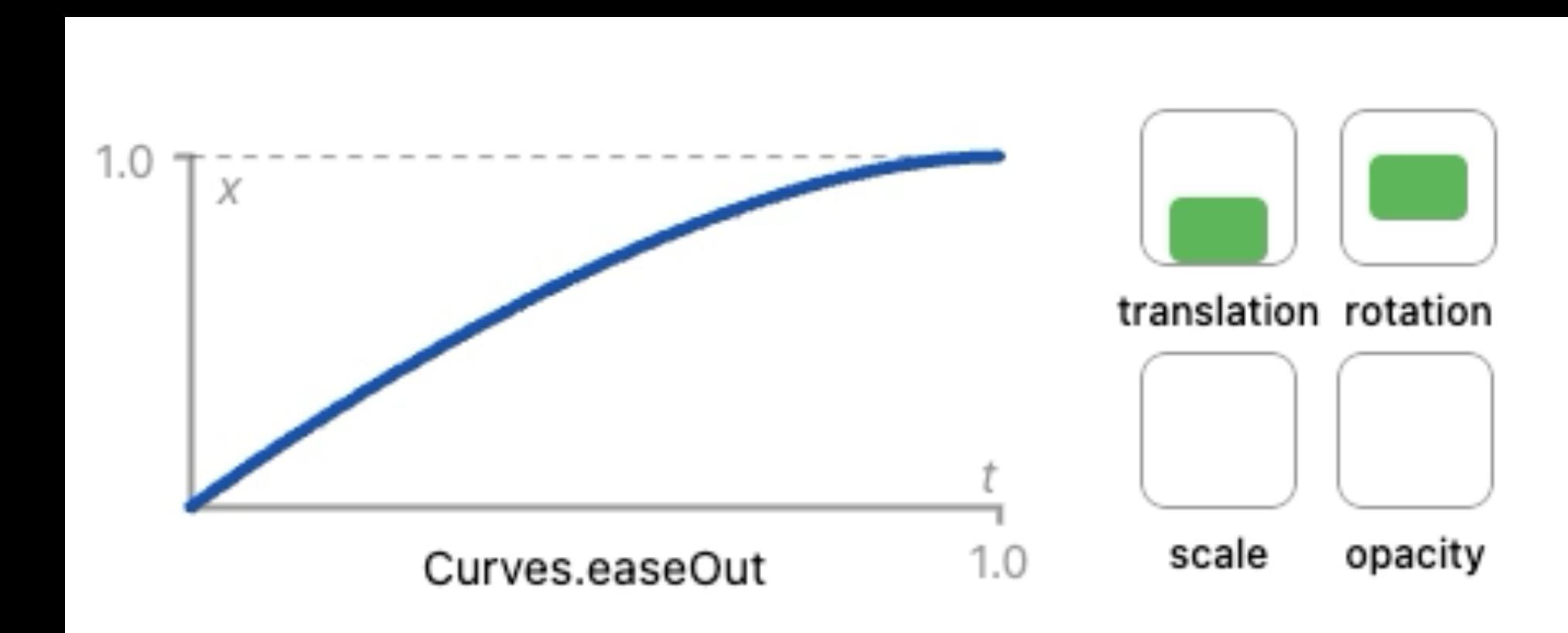
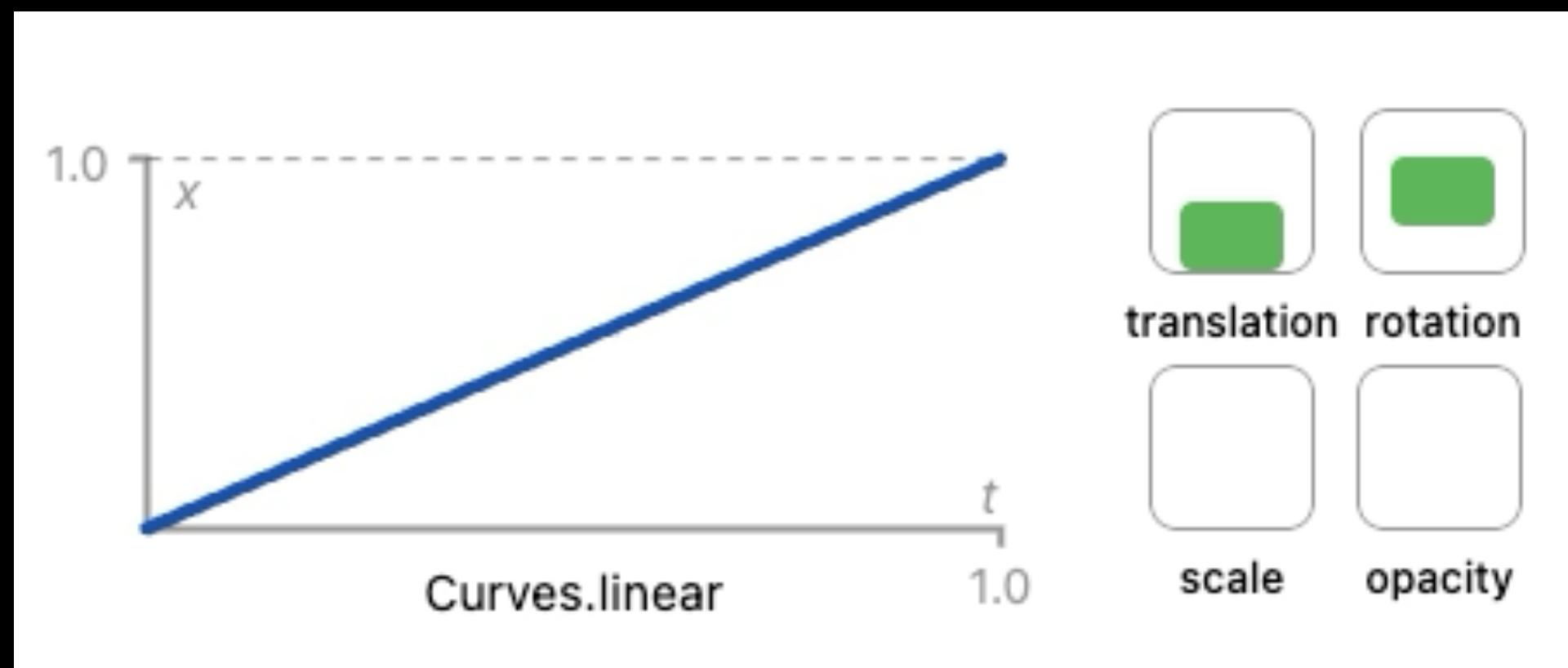
```
return Container(  
    width: width,  
    height: height,  
    decoration: const BoxDecoration(  
        gradient: LinearGradient(  
            colors: [  
                Color(0xFFFF0000),  
                Color(0xFF0000FF),  
            ],  
        ),  
    ),  
);
```



```
return AnimatedContainer(  
    duration: const Duration(milliseconds: 500),  
    width: width,  
    height: height,  
    color: Colors.renanPink,  
);
```

```
return AnimatedContainer(  
    duration: const Duration(milliseconds: 500),  
    width: width,  
    height: height,  
    color: Colors.renanPink,  
    curve: Curves.easeInOut,  
);
```







renan.gg

EXAMPLE 2

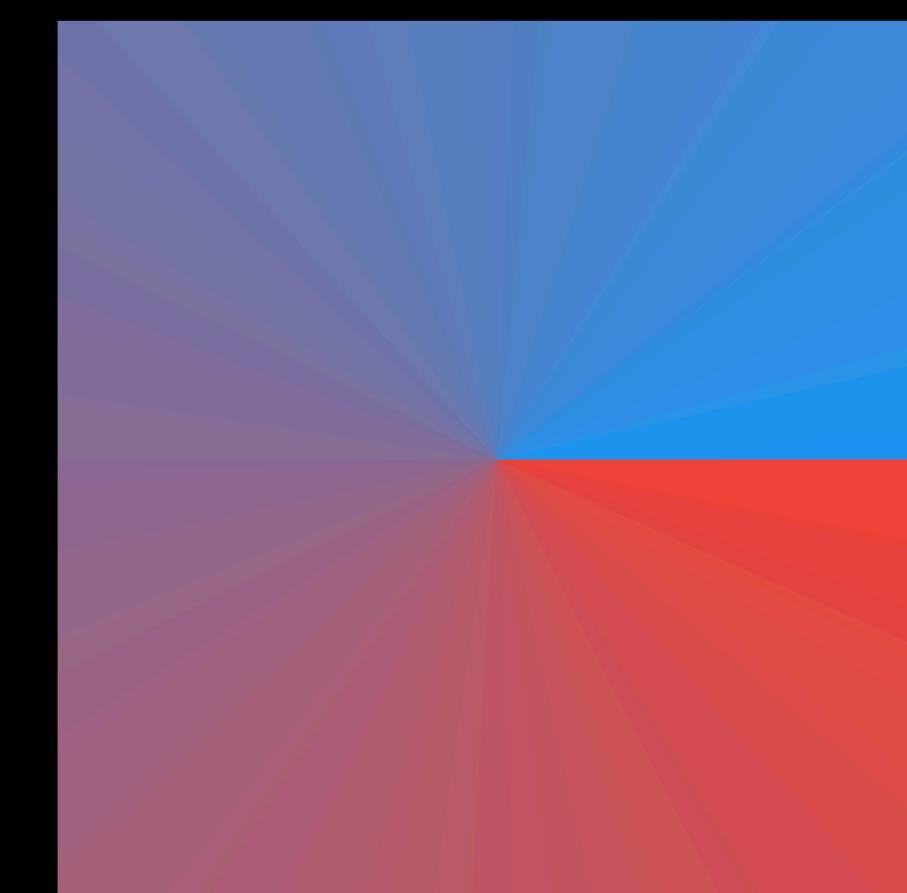
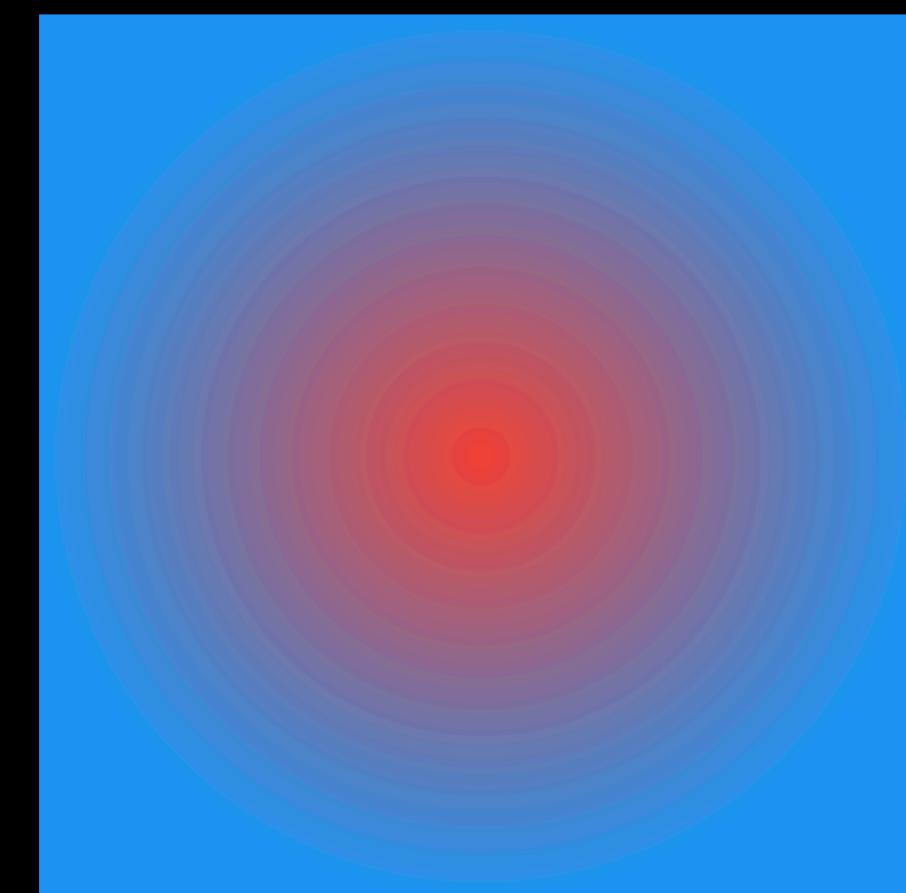
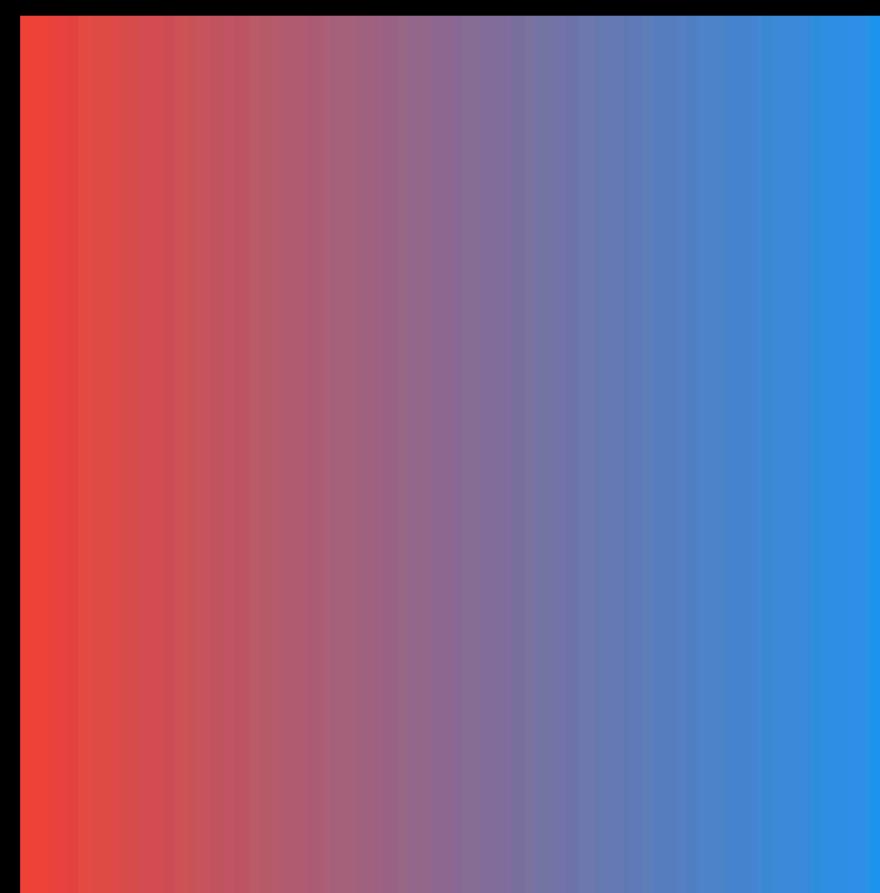
Easing Gradients

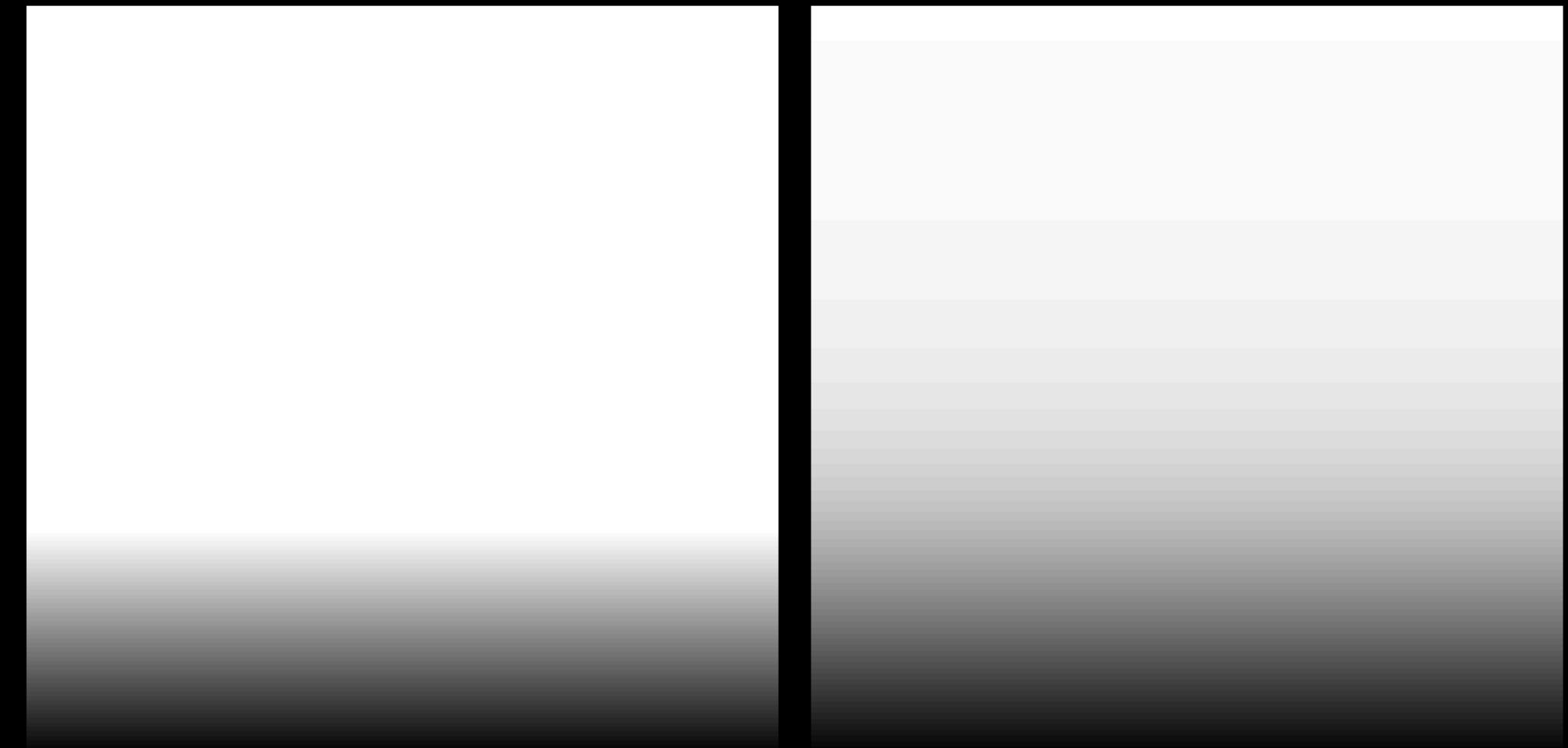
We can interpolate

- ✓ Color
- ✓ Coordinates
- ✓ Size
- ✓ Transform matrices
- ✓ Forms and shape
- ✓ Speed
- ✓ Other interpolations

I really love gradients

I really love gradients





renan.gg

```
gradient: LinearGradient(  
    colors: const [  
        Colors.white,  
        Colors.black,  
    ],  
    stops: stops,  
    begin: Alignment.topCenter,  
    end: Alignment.bottomCenter,  
) ,
```

```
return CustomPaint(  
    foregroundPainter: EasingGradientPainter(  
        shader: shader,  
        color1: Colors.white,  
        color2: Colors.black,  
        stop1: stops[0],  
        stop2: stops[1],  
        easing: easing,  
    ),  
    child: SizedBox.fromSize(  
        size: const Size.square(370),  
    ),  
);
```

```
    shader.setFloatUniforms(setter) {  
        setter  
            ..setSize(size)  
            ..setColor(color1)  
            ..setFloat(stop1)  
            ..setColor(color2)  
            ..setFloat(stop2)  
            ..setFloat(easing.a)  
            ..setFloat(easing.b)  
            ..setFloat(easing.c)  
            ..setFloat(easing.d);  
    });
```

```
void fragment(vec2 uv, vec2 pos, inout vec4 color) {
    float interpolant = uv.y;
    if(interpolant < stop1) {
        color = color1;
        return;
    }
    if(interpolant > stop2) {
        color = color2;
        return;
    }
    interpolant = (interpolant - stop1) / (stop2 - stop1);

    interpolant = cubicInterpolation(interpolant, cubic.x, cubic.y, cubic.z, cubic.w);
    color = mix(color1, color2, interpolant);
}
```

What if we need a new type of gradient?

What if we need a new type of gradient?

EXAMPLE 3

Mesh Gradient

The screenshot shows the Stripe homepage with a vibrant orange-to-yellow gradient background. At the top, the Stripe logo is followed by navigation links: Products, Solutions, Developers, Resources, Pricing, Contact sales, and Sign in. The main headline reads "Payments infrastructure for the internet". Below it, a sub-headline states: "Millions of businesses of all sizes—from startups to large enterprises—use Stripe's software and APIs to accept payments, send payouts, and manage their businesses online." Two calls to action are present: "Start now" and "Contact sales". To the right, there are two overlapping interface snippets: a payment form for "Increment Magazine" showing an Apple Pay button and a reporting dashboard for "ROCKET RIDES" displaying financial metrics like Gross volume, Balance, and Payouts, along with line graphs for Reports summary.

stripe Products Solutions Developers Resources Pricing

Contact sales Sign in

Payments infrastructure for the internet

Millions of businesses of all sizes—from startups to large enterprises—use Stripe's software and APIs to accept payments, send payouts, and manage their businesses online.

Start now Contact sales

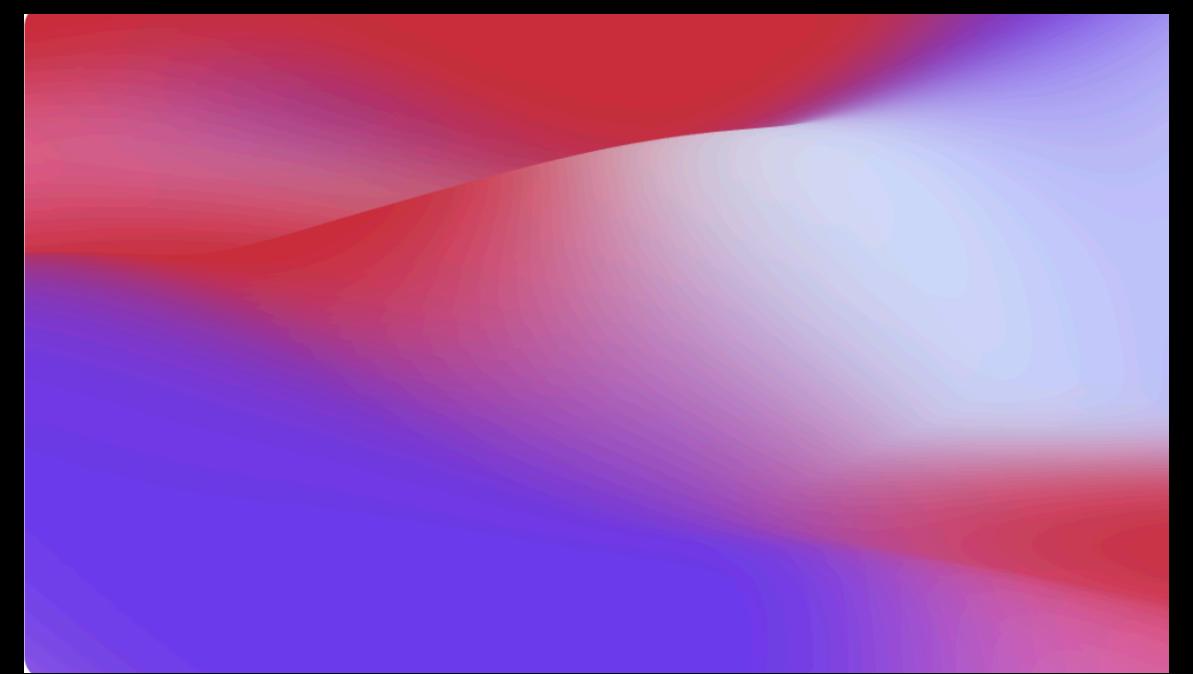
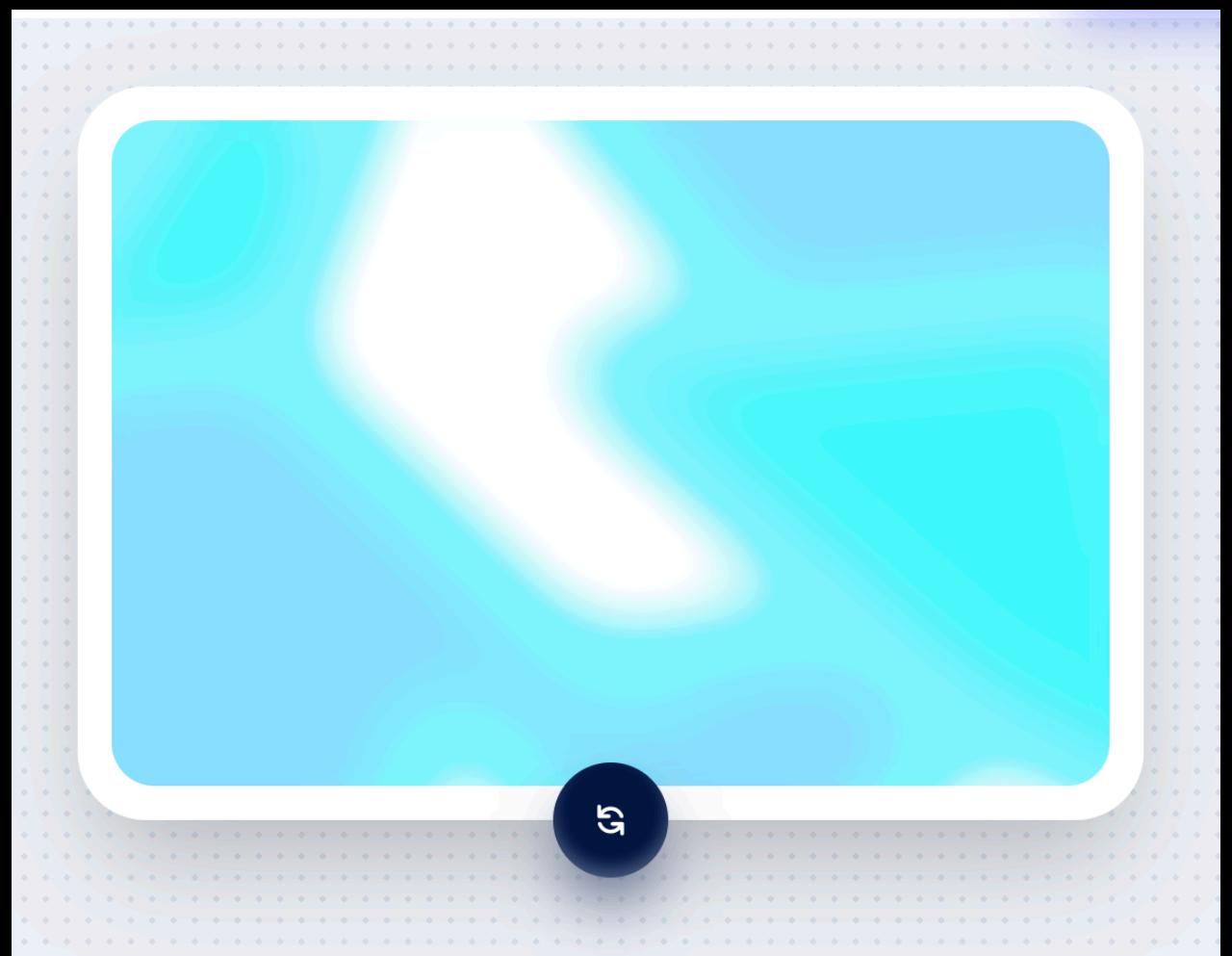
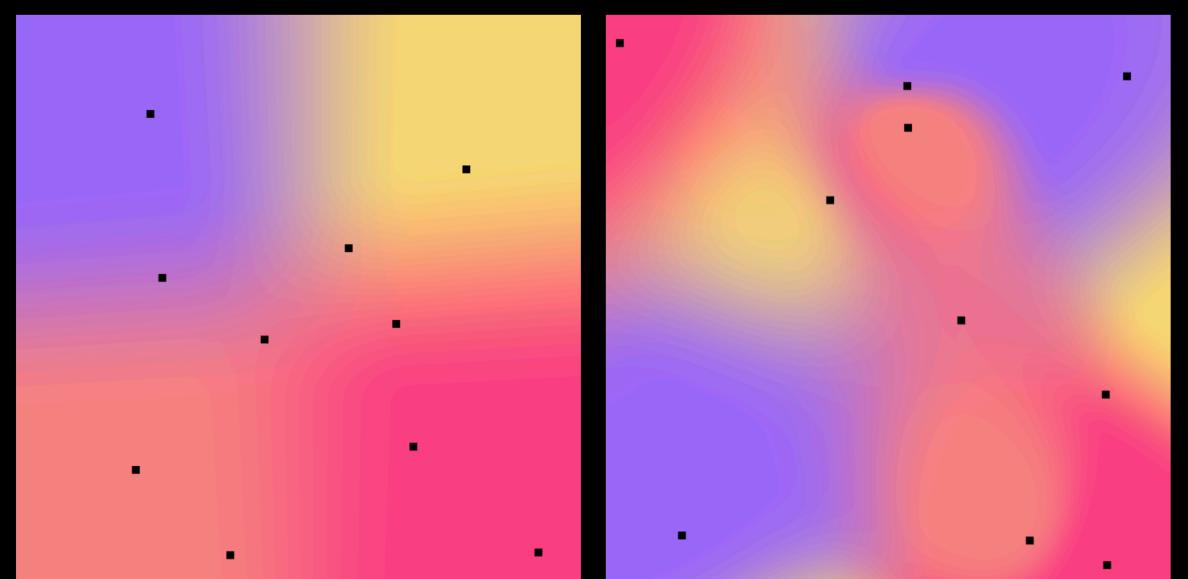
Booking.com statista SHARENOW joyn

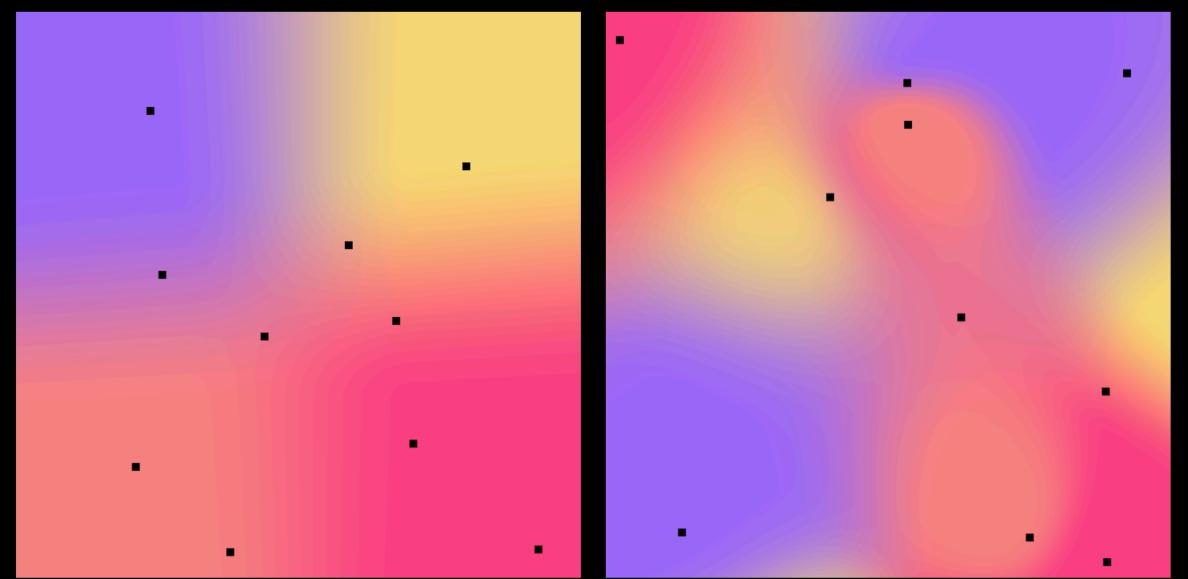
ASOS MARKETPLACE salesforce axel springer shopify

Unified platform

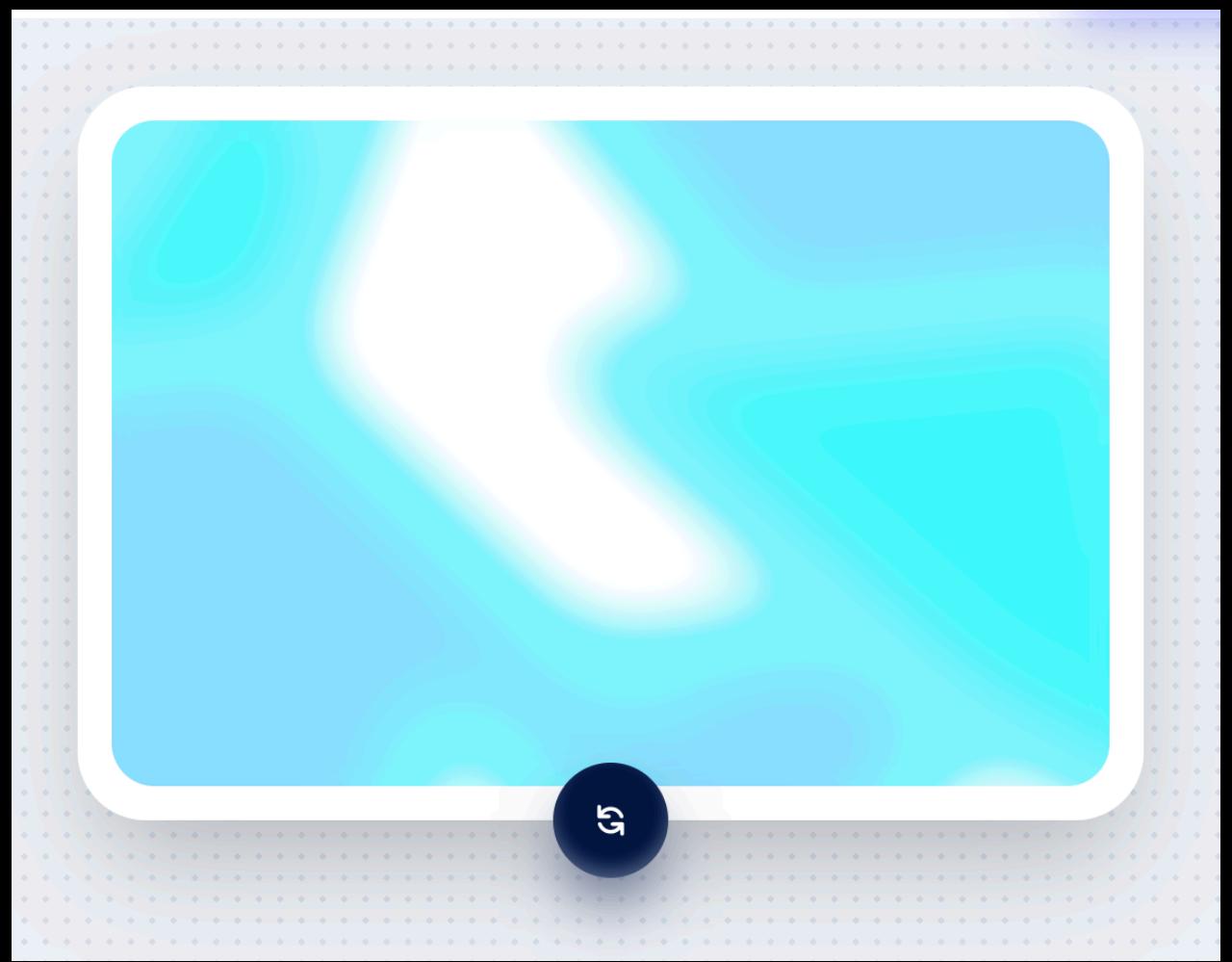
A fully integrated suite of

renan.gg

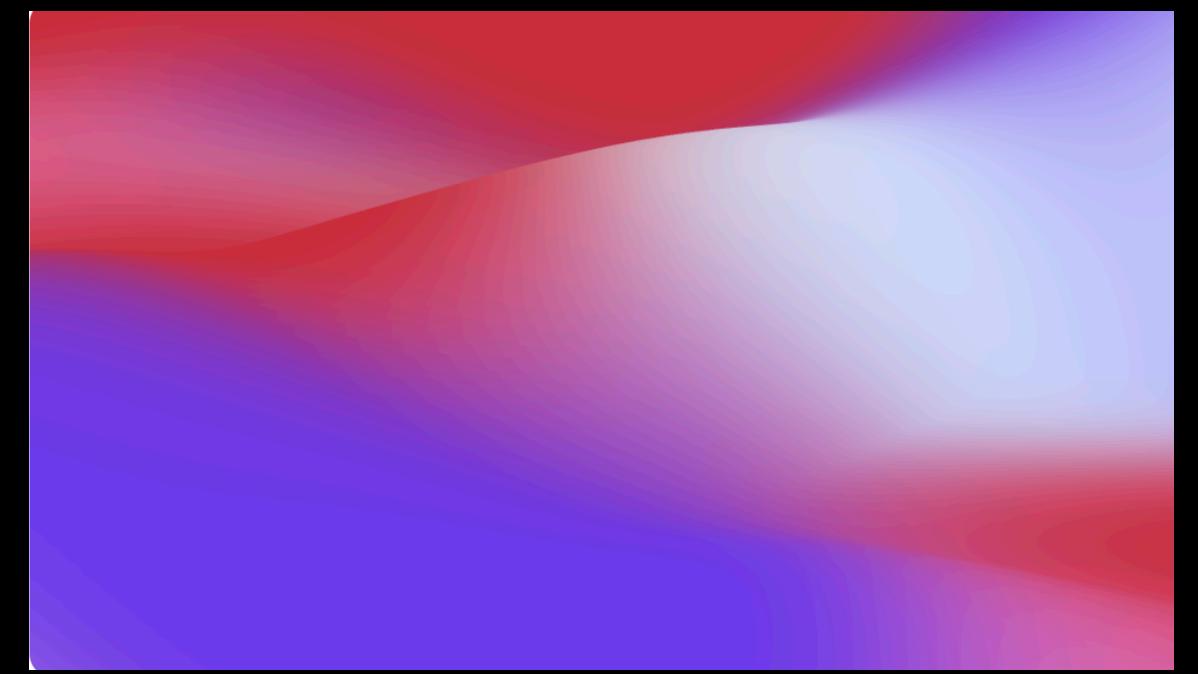




Tool 1

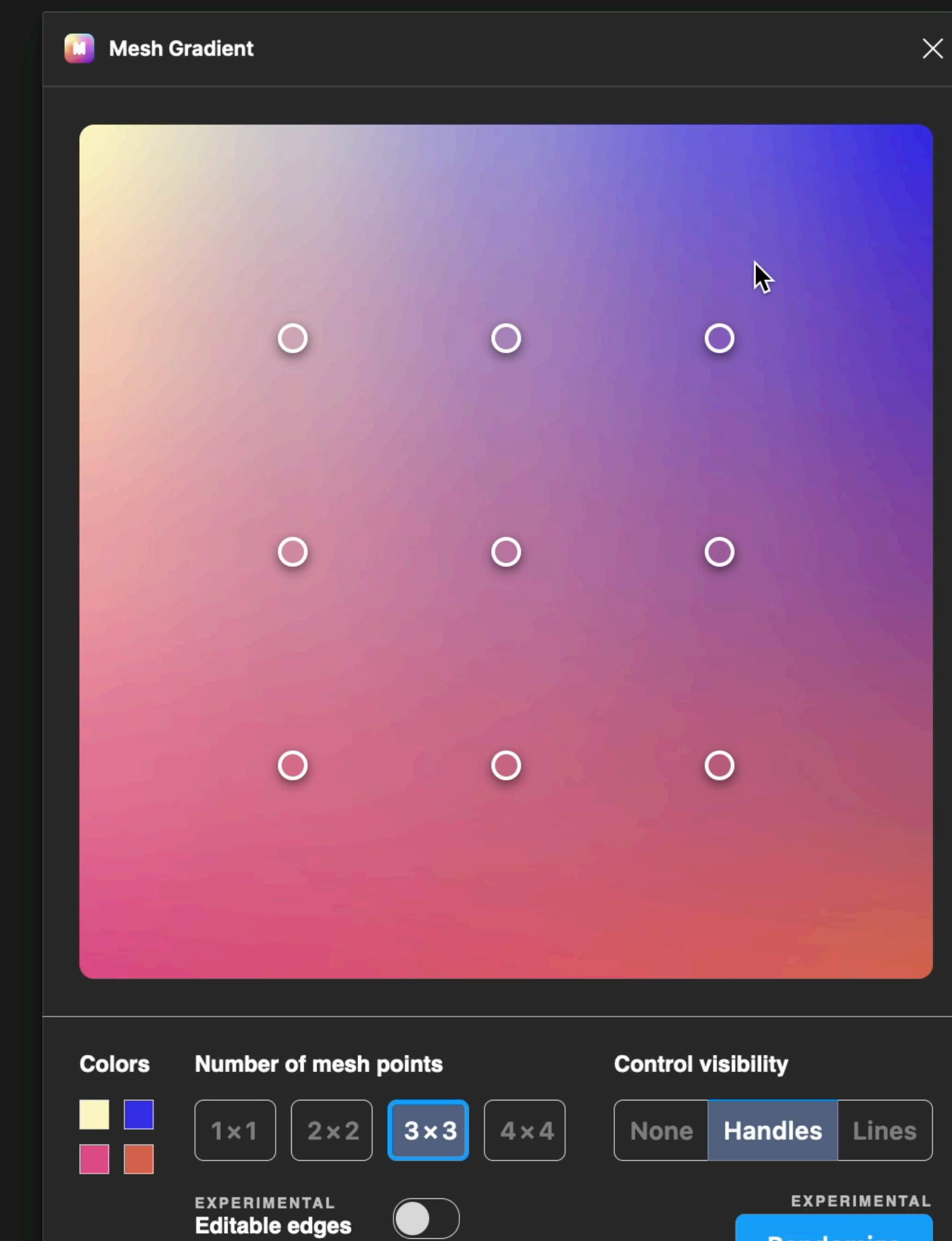


Tool 2

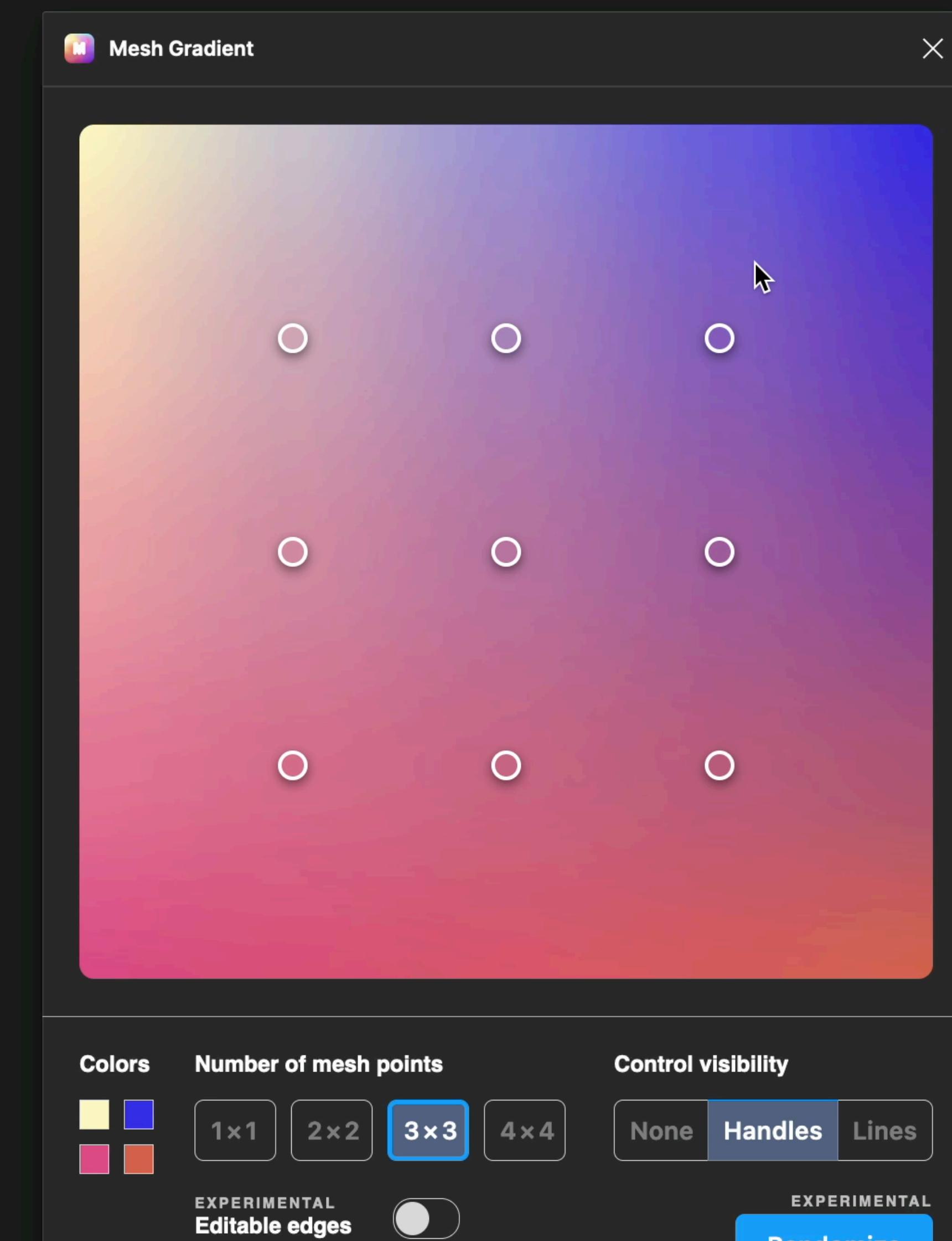


Tool 3

The Figma plugin



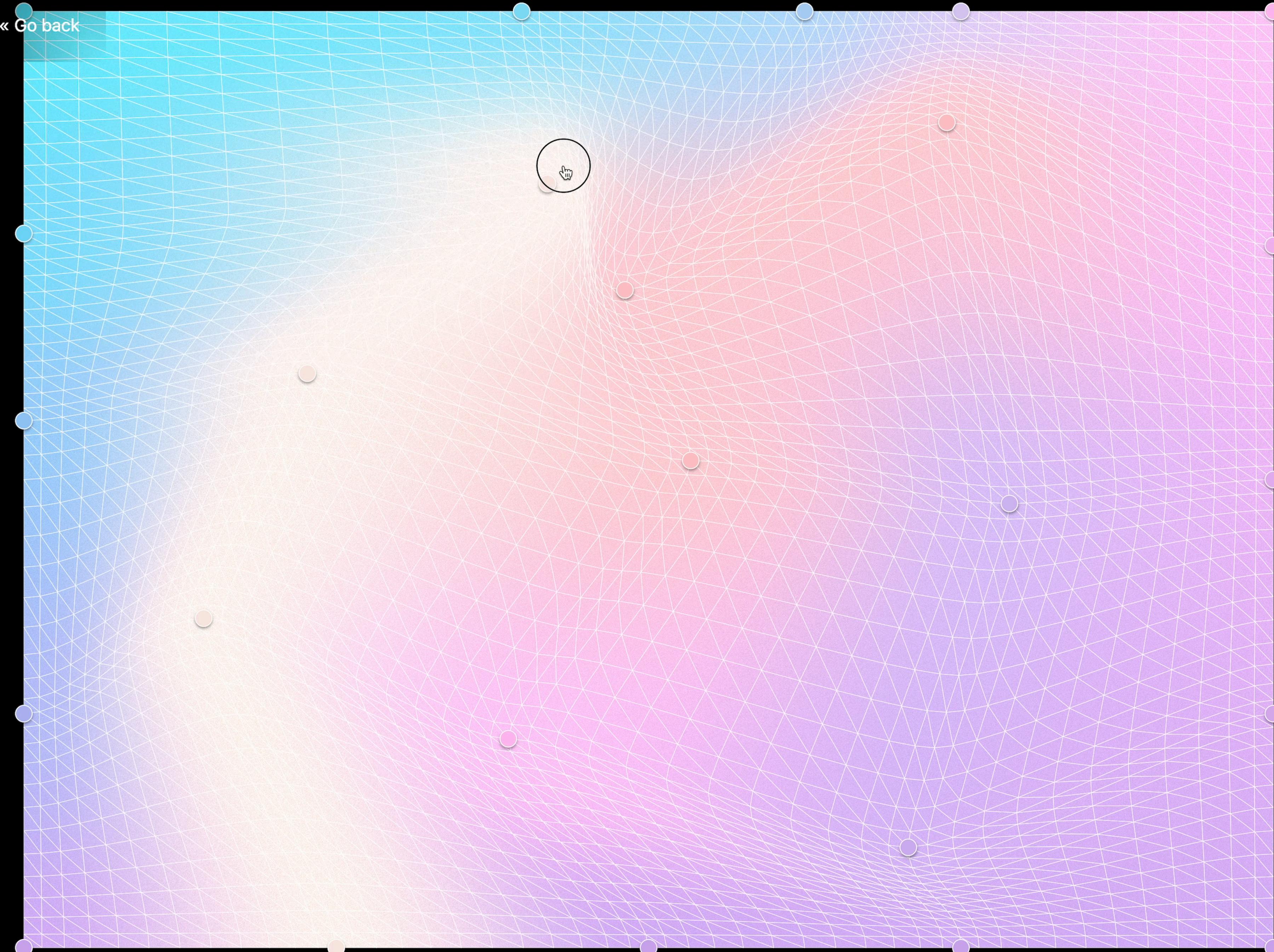
The Figma plugin



Requirements

- ✓ Vertices can be moved around
- ✓ Each Vertices can carry its own color
- ✓ Let's start with a big mesh: 5x5 vertices
- ✓ Animatable and portable

« Go back



Tess:

Easing X:

Easing Y:



Show mesh

Show color pickers

Aspect Ratio: 1.33

16:9

4:3

3:4

9:16

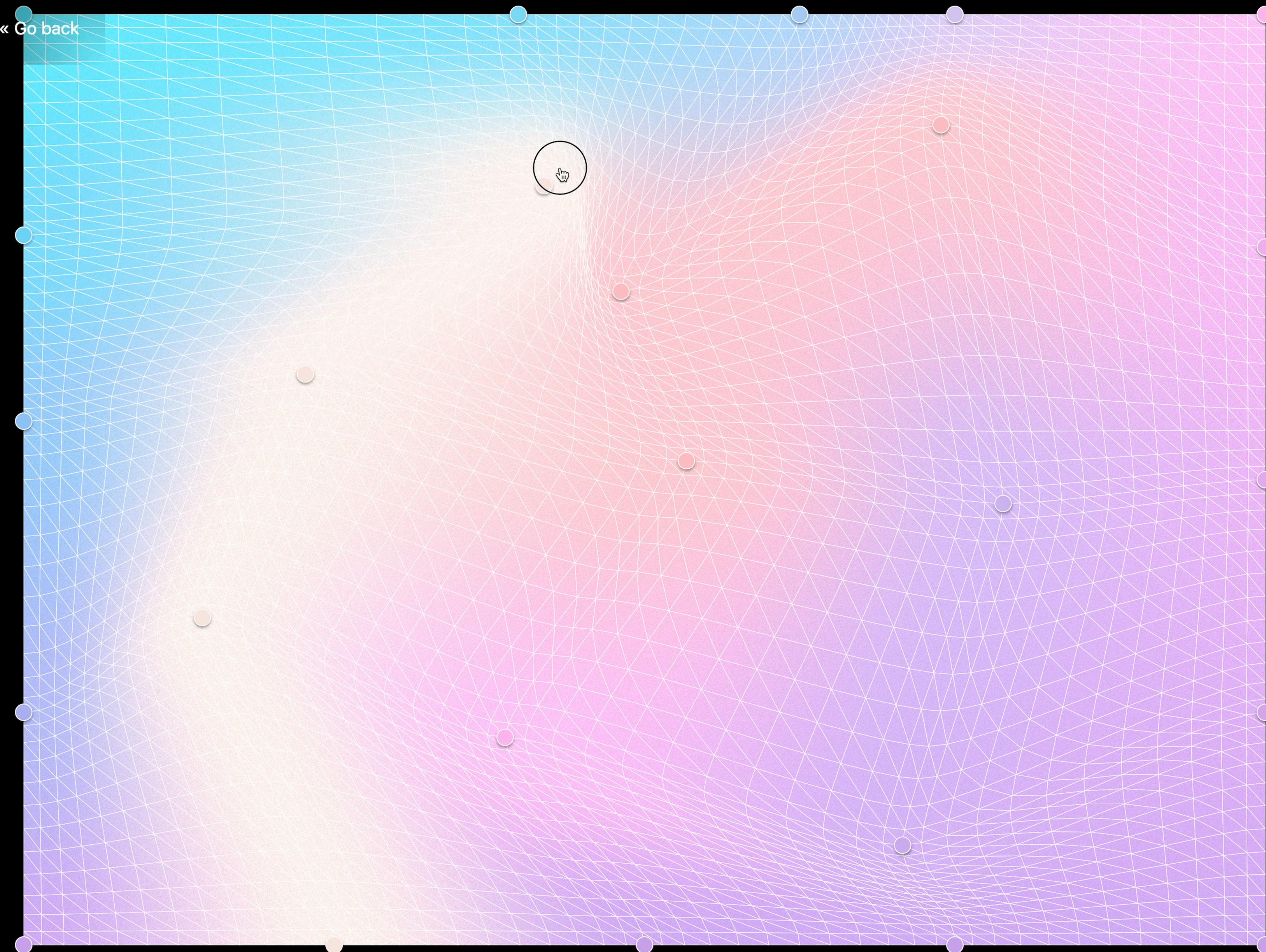
Deform

Reset

```
789ce365b43bfdf52f031030d9cbff7c  
0362b08208bb62c63c36bb6b53f6da  
d90472b3db7df39d663723f83987bd  
ad6c90eda76d325c6035f7cf3de3b6c  
bf069b13b7adb9bc79e63e129bbafb9  
32bcf69eeef56dab3f52c12b0130eb7b  
5577d3943d0ee5893aabdd9d6f42f  
6a6a625f6295ddf3944ffff7a7a5f004  
4f0fcfbfffa8e0fff3bde7183b8ec202  
e0788600371ff073dfe0592fb7f72f17b  
10e6b1f3397316840165695835
```

set hash

« Go back



Tess:

Easing X:

Easing Y:



Show mesh

Show color pickers

Aspect Ratio: 1.33

16:9

4:3

3:4

9:16

Deform

Reset

```
789ce365b43bfdf52f031030d9cbff7c  
0362b08208bb62c63c36bb6b53f6da  
d90472b3db7df39d663723f83987bd  
ad6c90eda76d325c6035f7cf3de3b6c  
bf069b13b7adb9bc79e63e129bbafbd  
32bcf69eef56dab3f52c12b0130eb7b  
5577d3943d0ee5893aabddb9d6f42f  
6a6a625f6295ddf3944ffff7a7a5f004  
4f0fcfbfffa8e0fff3bde7183b8ec202  
e0788600371ff073dfe0592fb7f72f17b  
10e6b1f3397316840165695835
```

set hash

Summary

- ✓ We learned what are interpolations and what shaders can do in that field
- ✓ We learned a use case that is only doable via CFS
- ✓ We learned that we can go from the subtle to the freacking crazy
- ✓ We learned that we can combine CFS with other Flutter APIs to create amazing effects



renan.gg

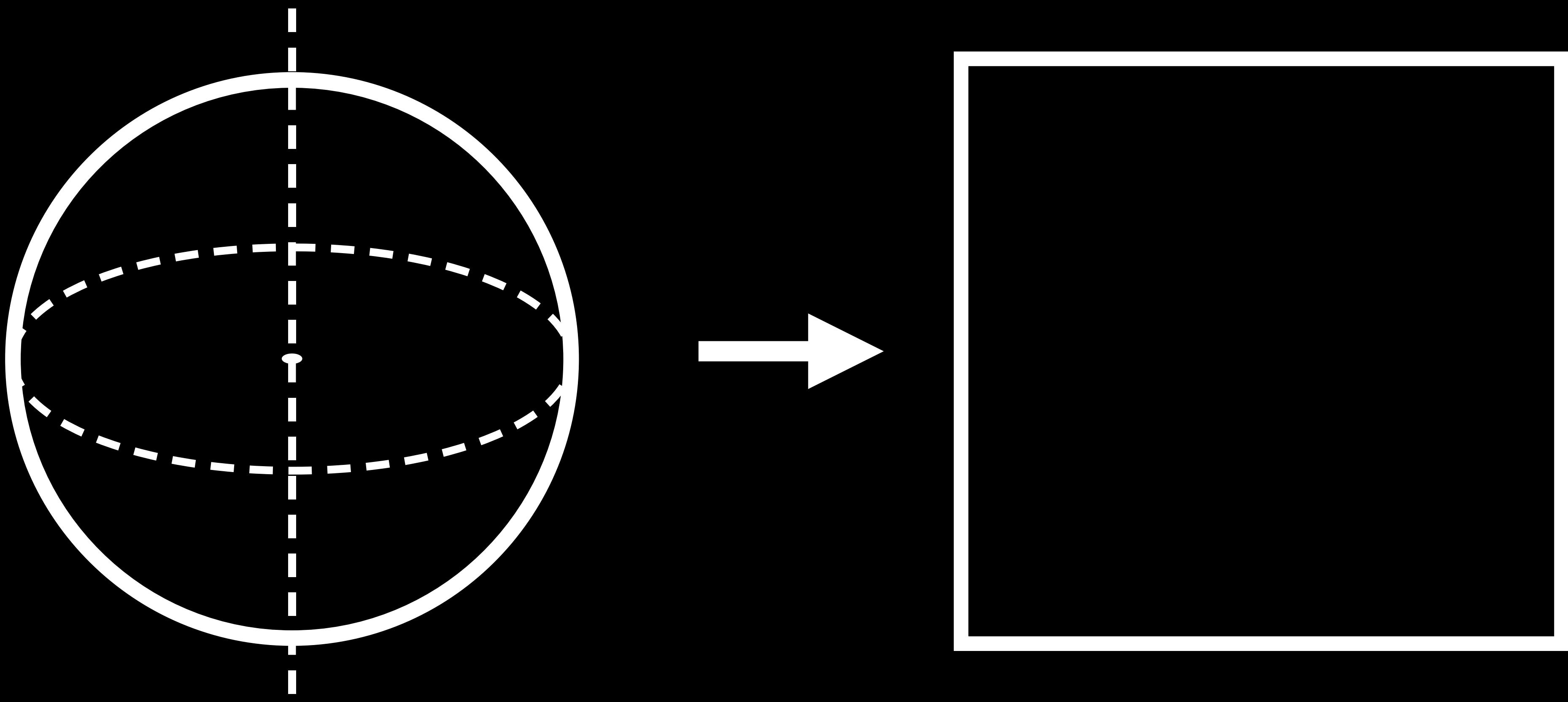


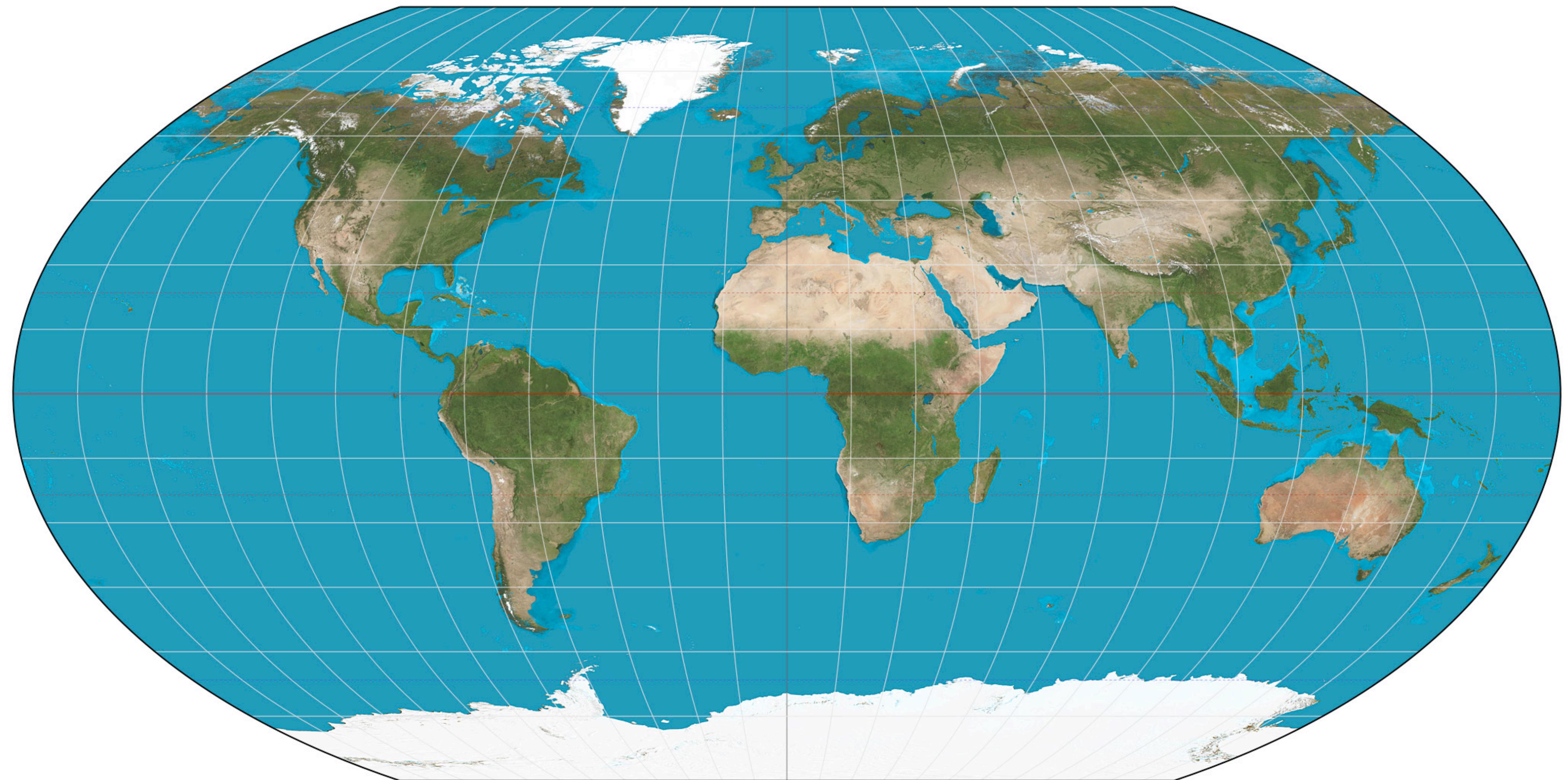
Textures

```
shader.setFloatUniforms((value) {  
    value.setFloat(myFloatValue);  
});
```

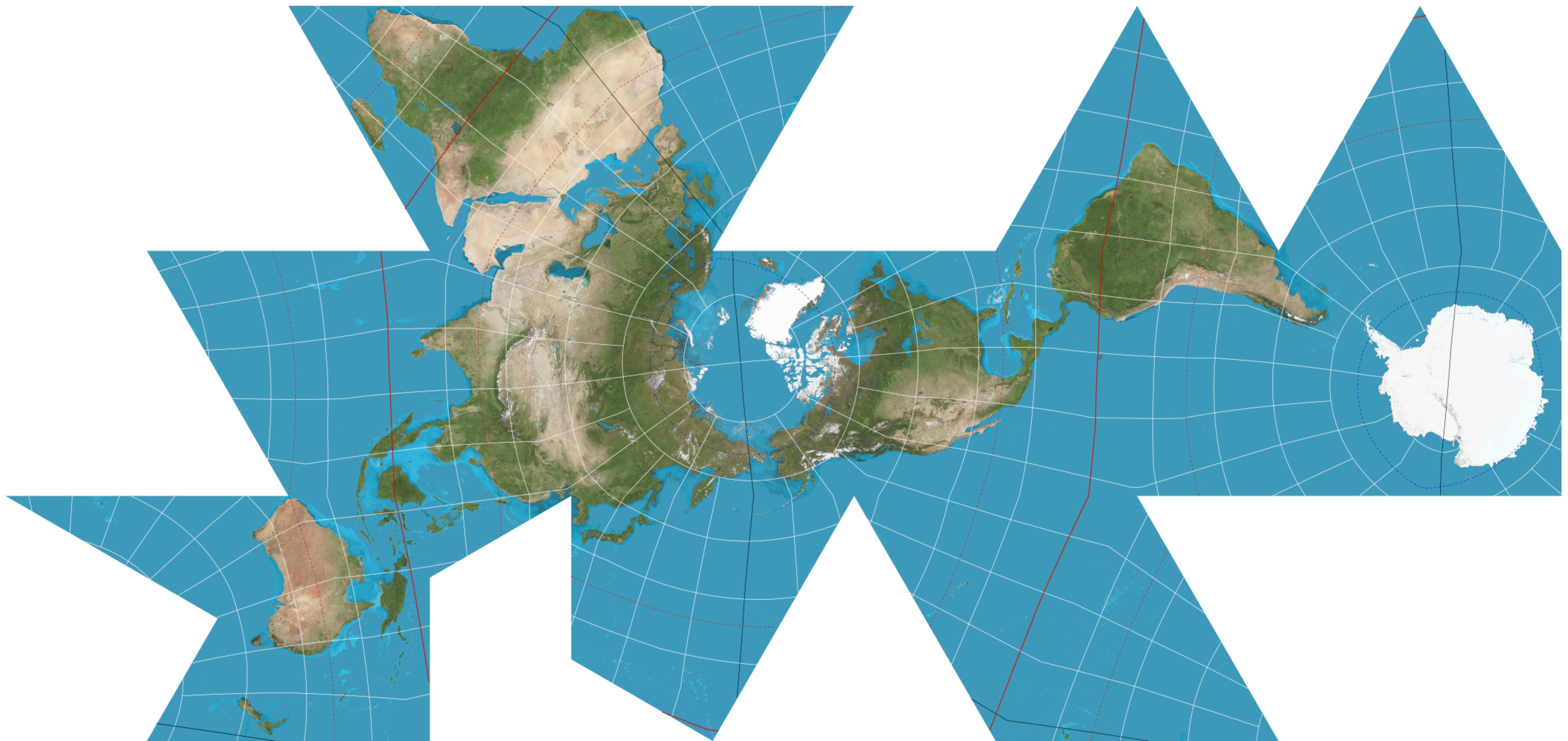
```
shader.setImageSampler(0, image); // pass a dart:ui Image to the shader
```

```
uniform sampler2D uTexture;
```

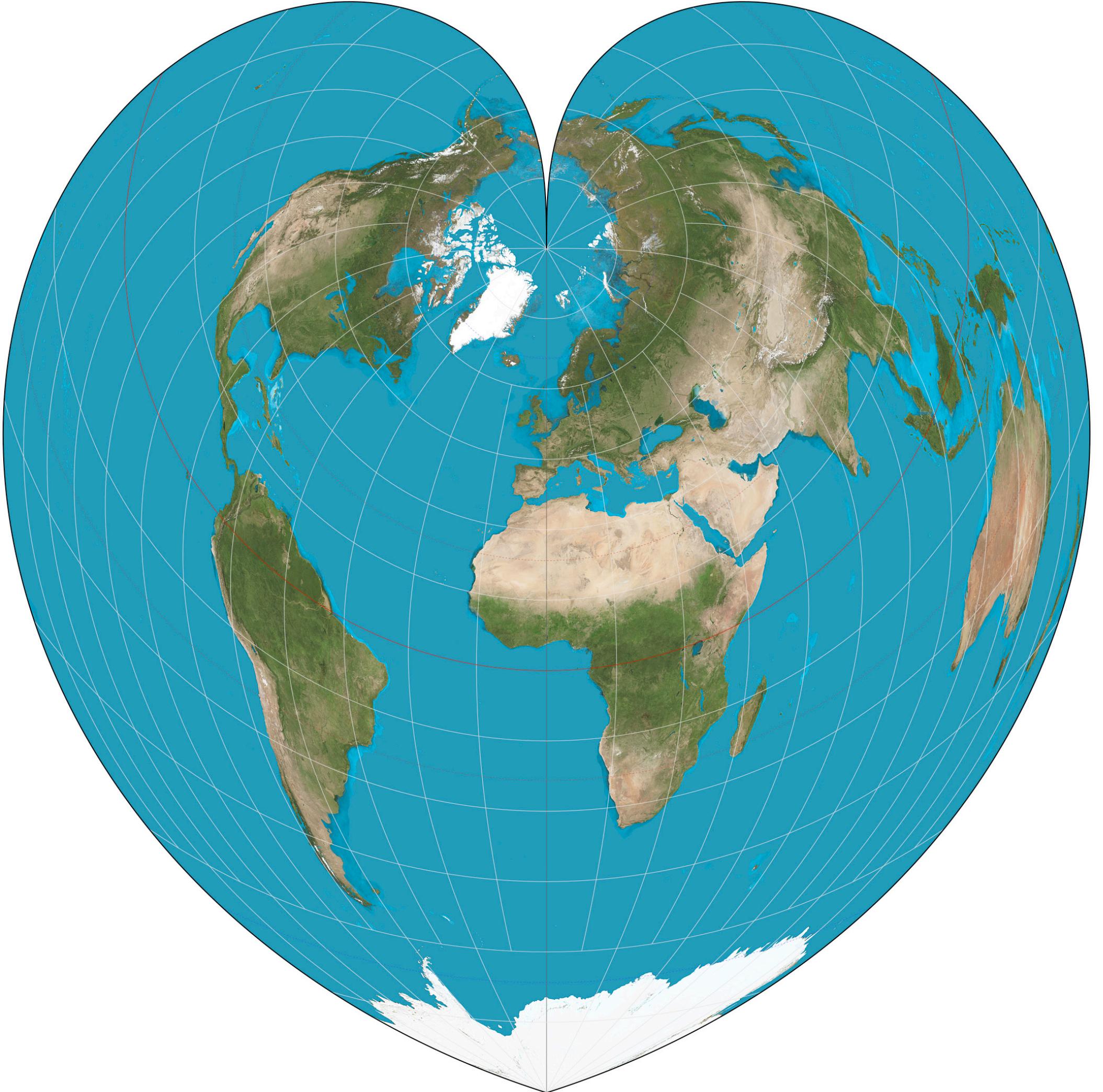




Wagner VI

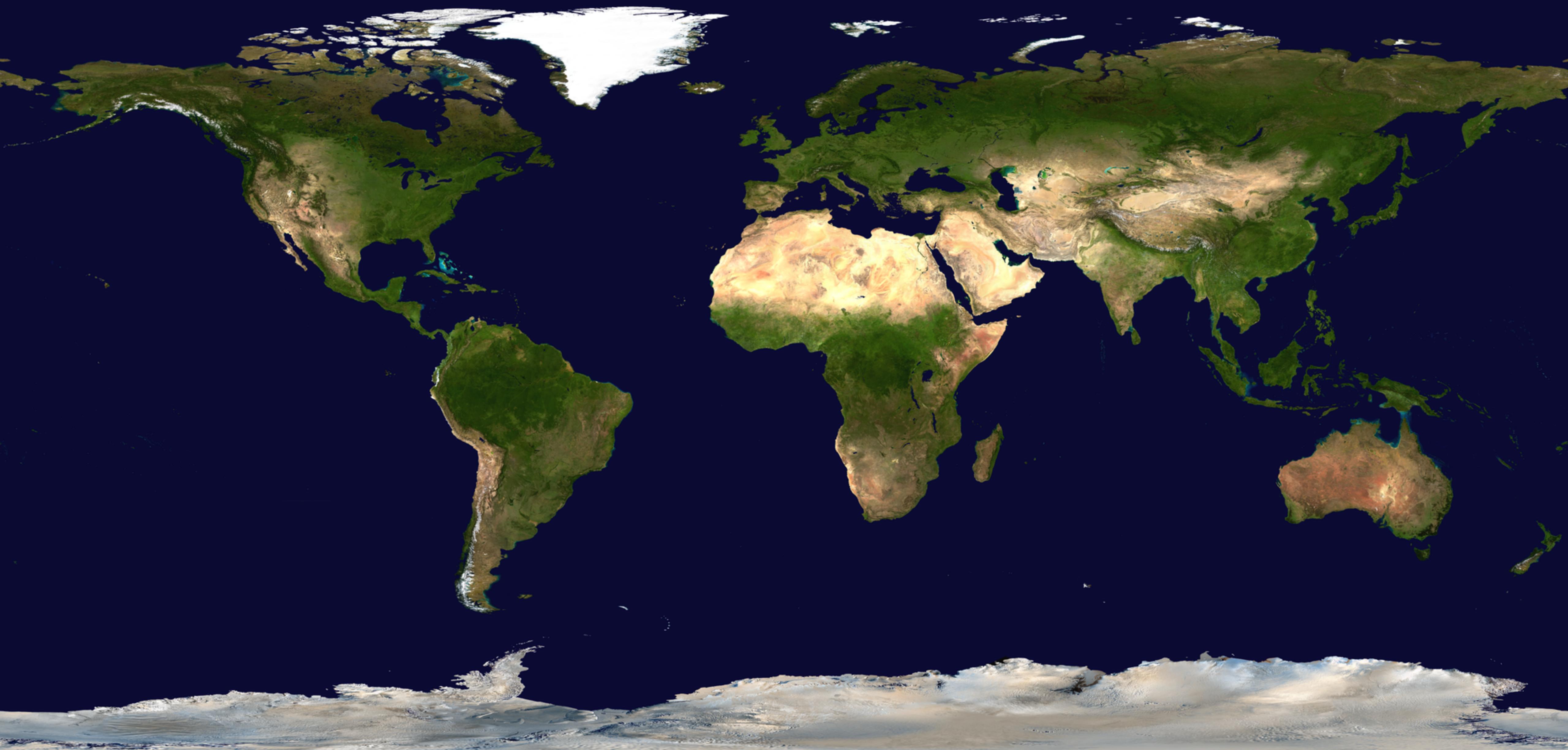


Dymaxion map

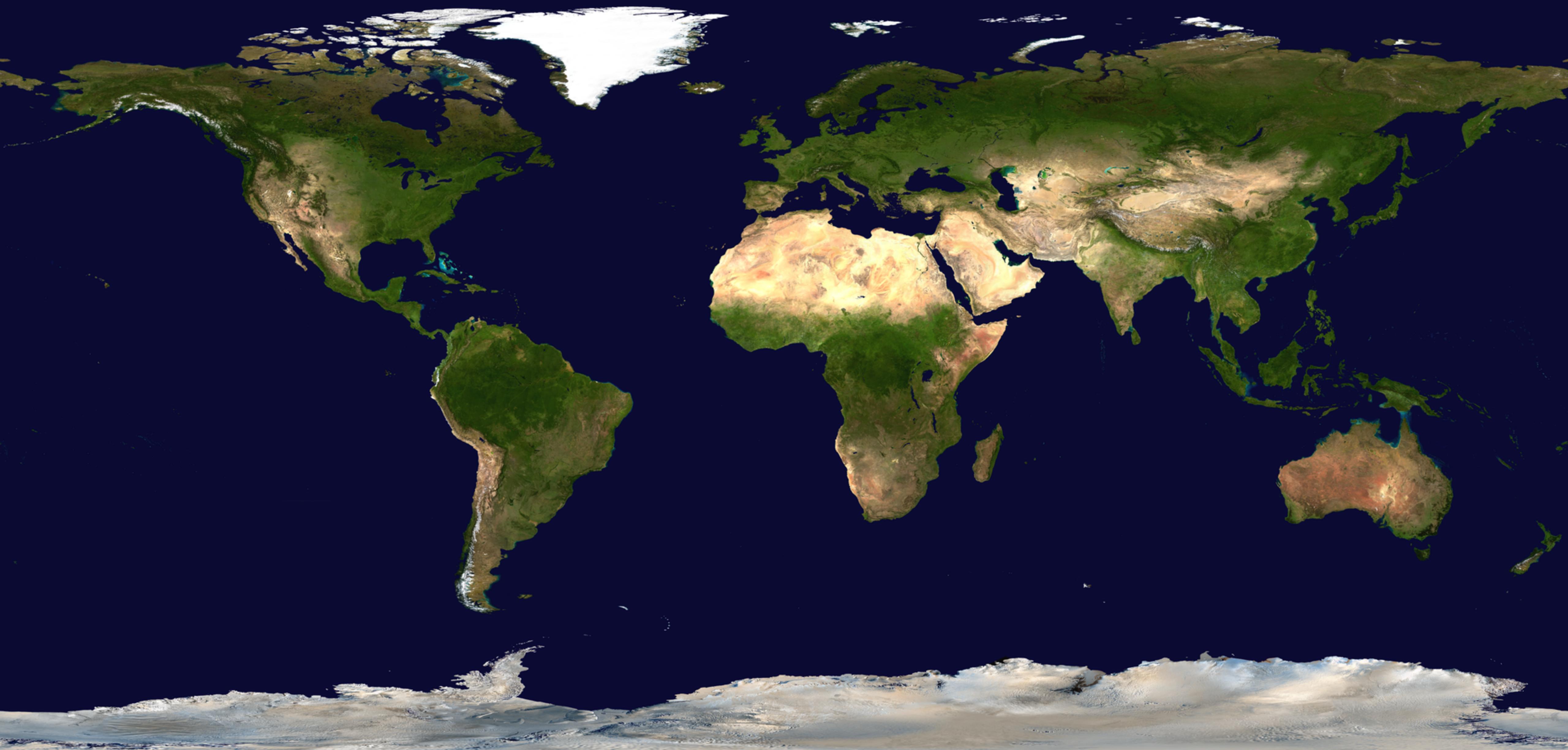


Werner

renan.gg



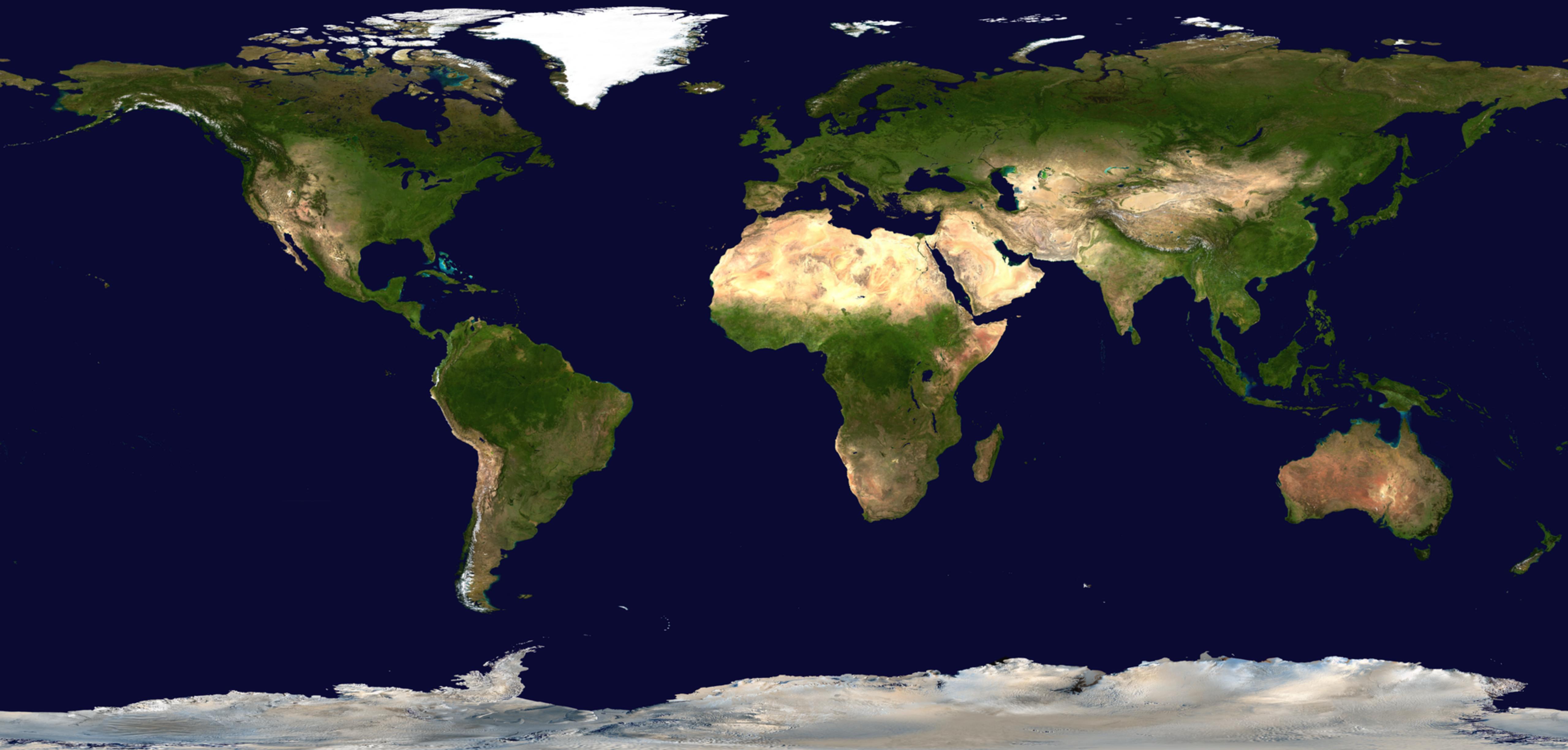
Thanks NASA for this pic



Thanks NASA for this pic



Thanks NASA for this pic



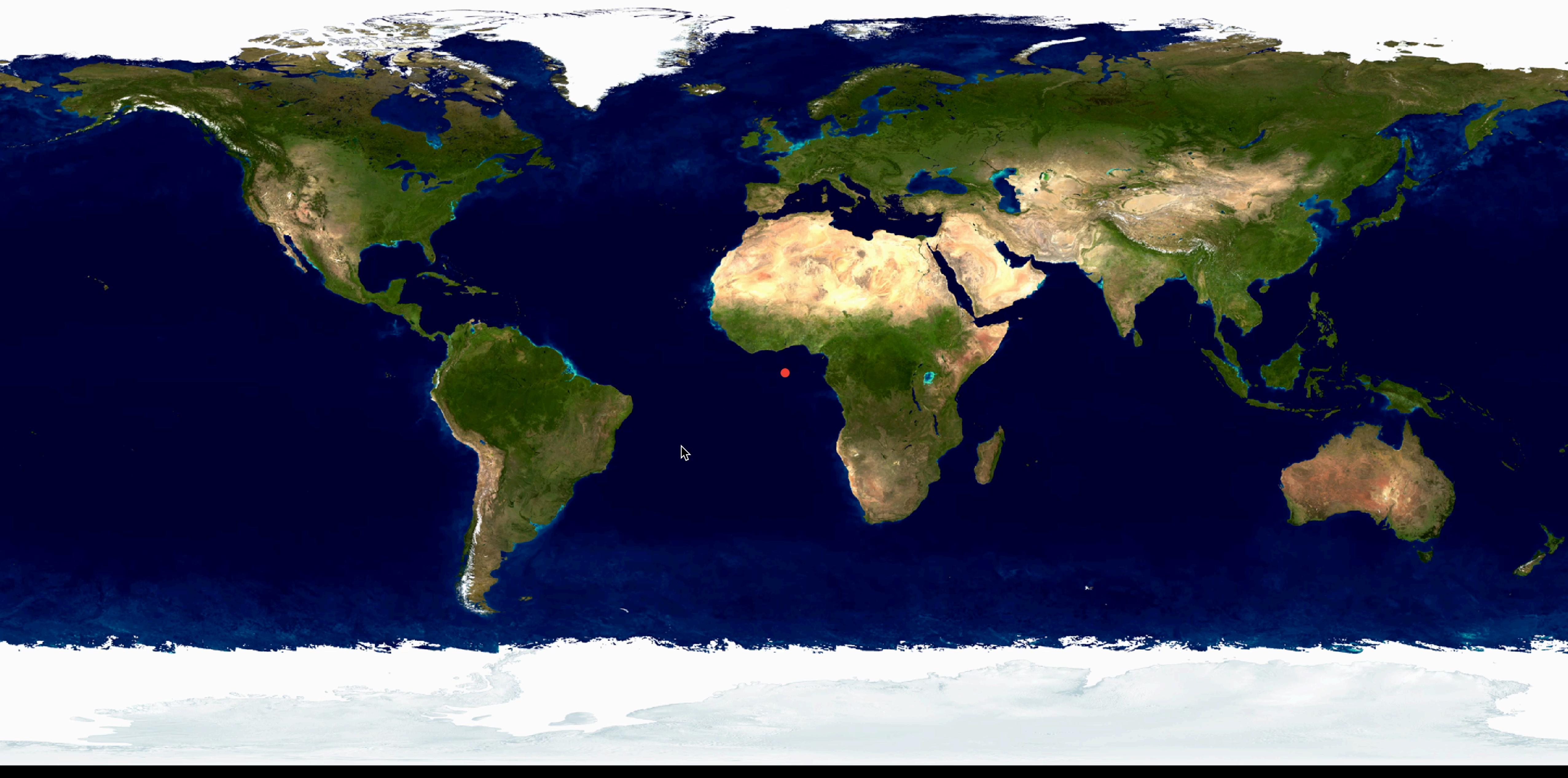
Thanks NASA for this pic

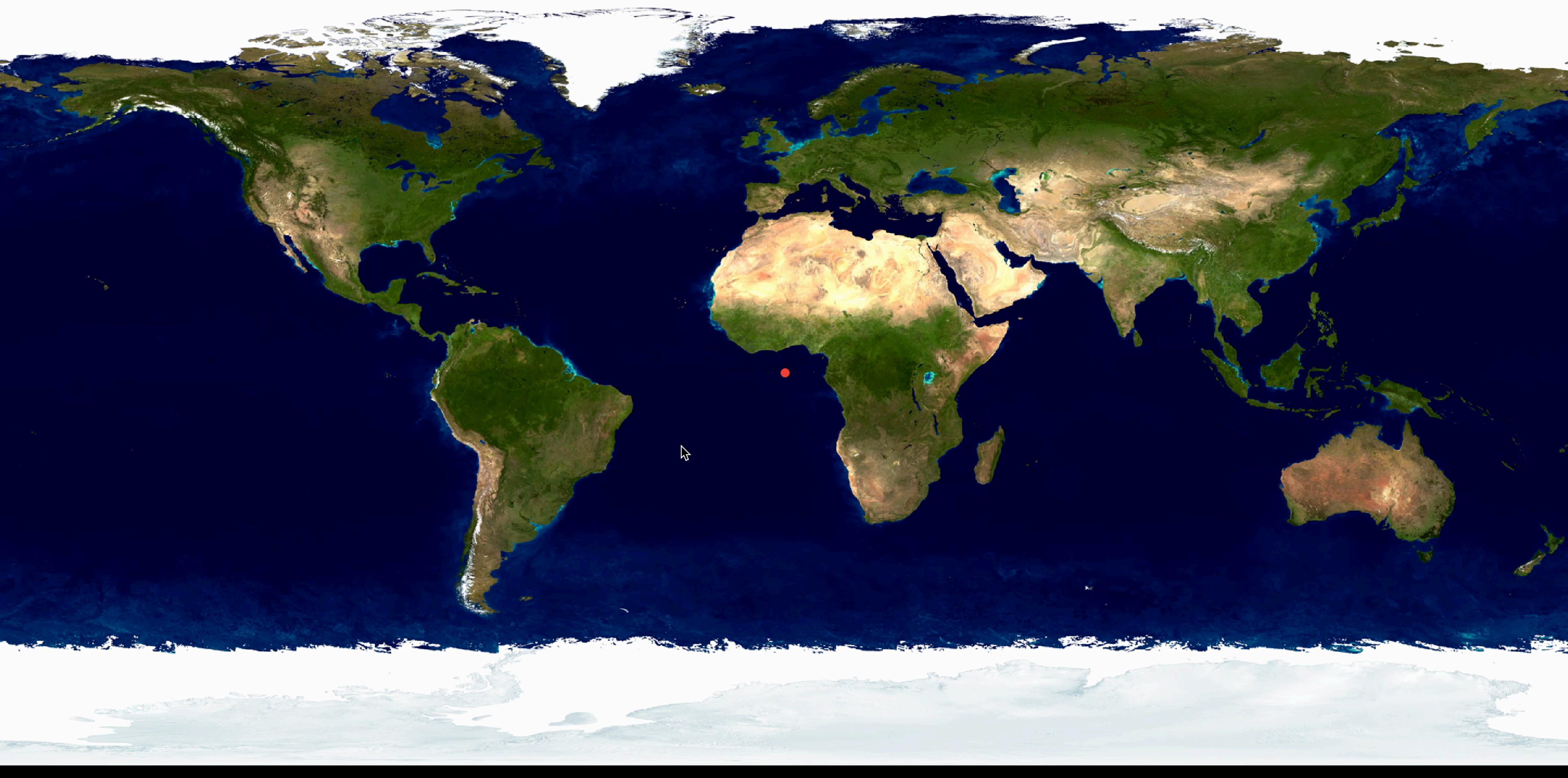


Thanks NASA for this pic

EXAMPLE 4

The earth is not flat







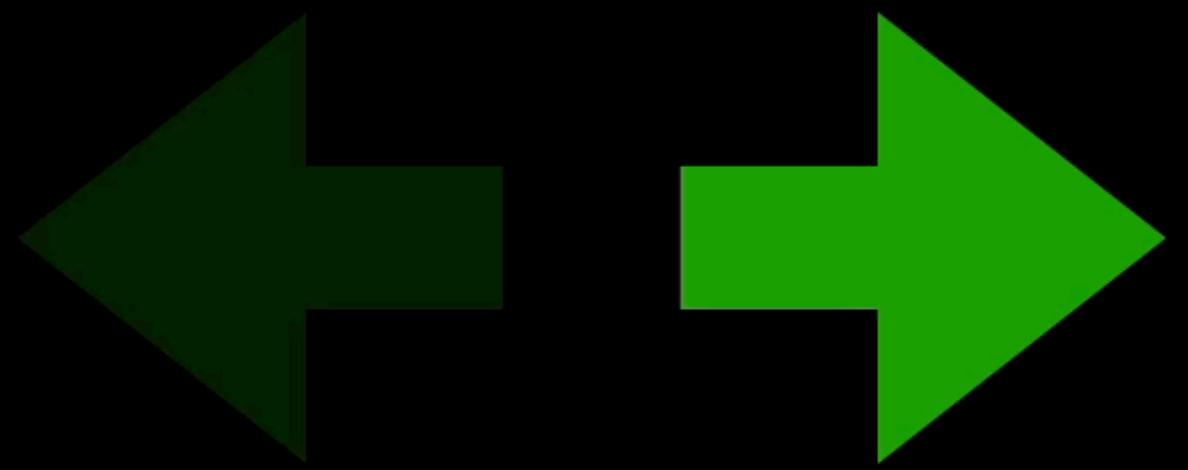
renan.gg



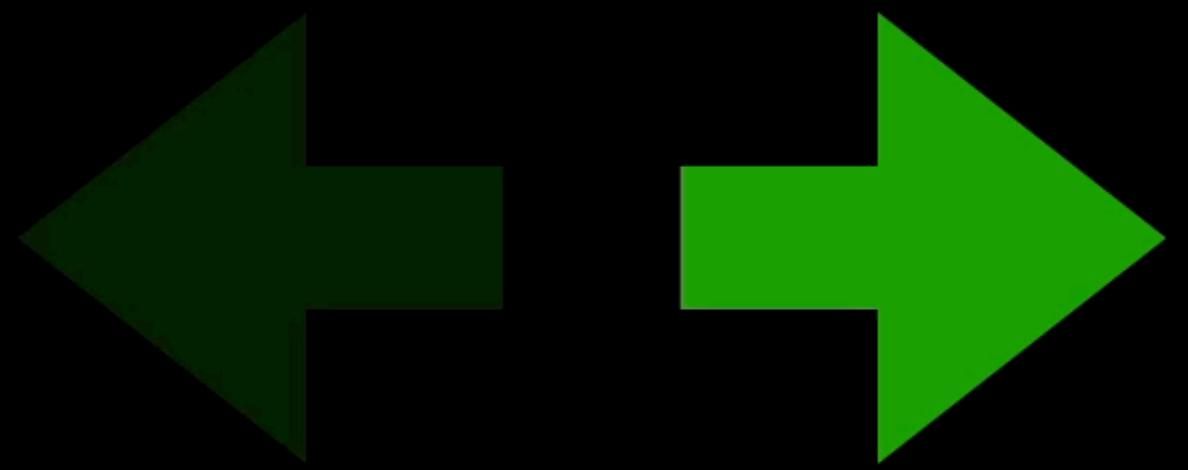
UI Sampling

Using an entire widget tree as texture

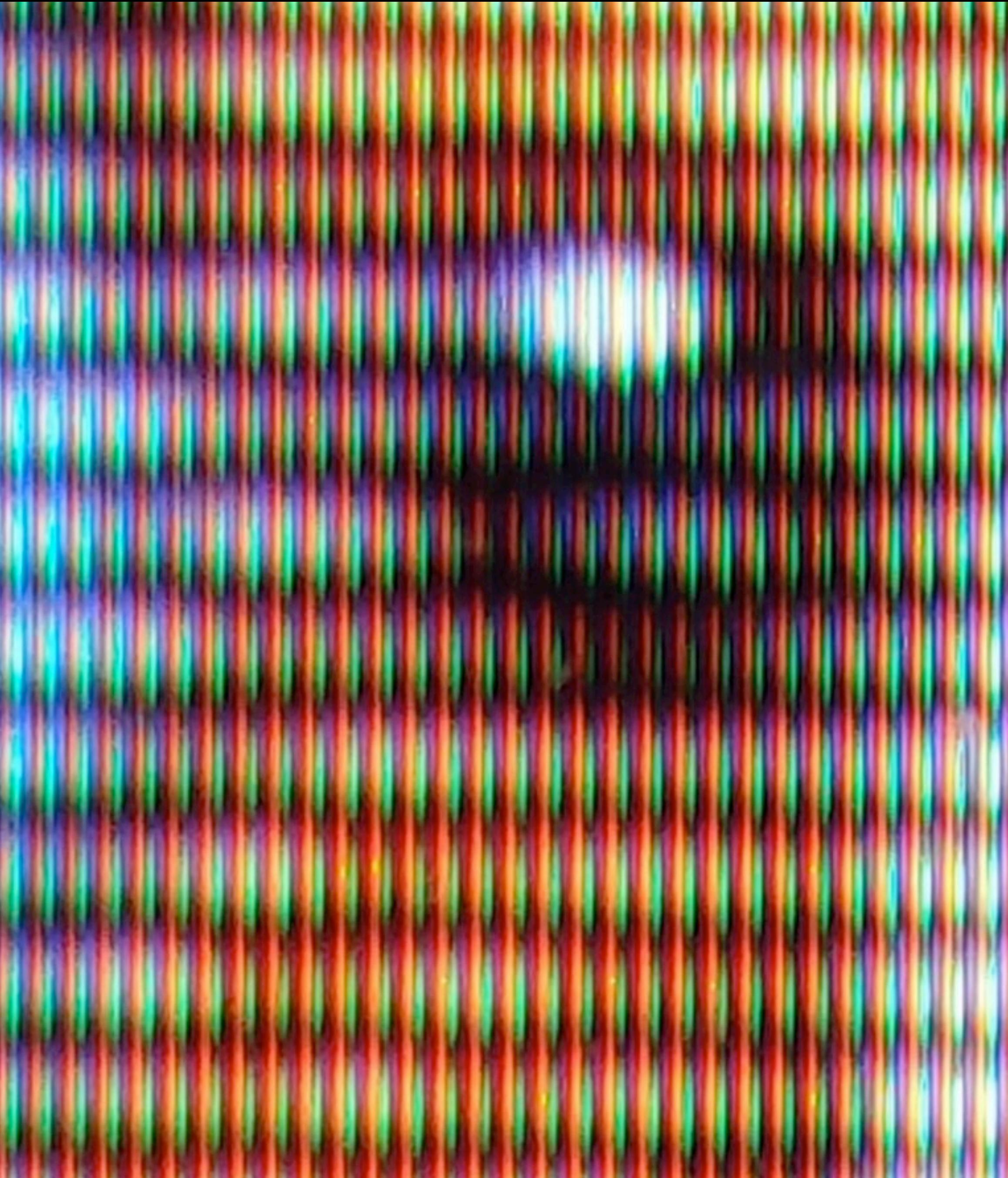
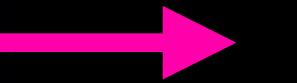
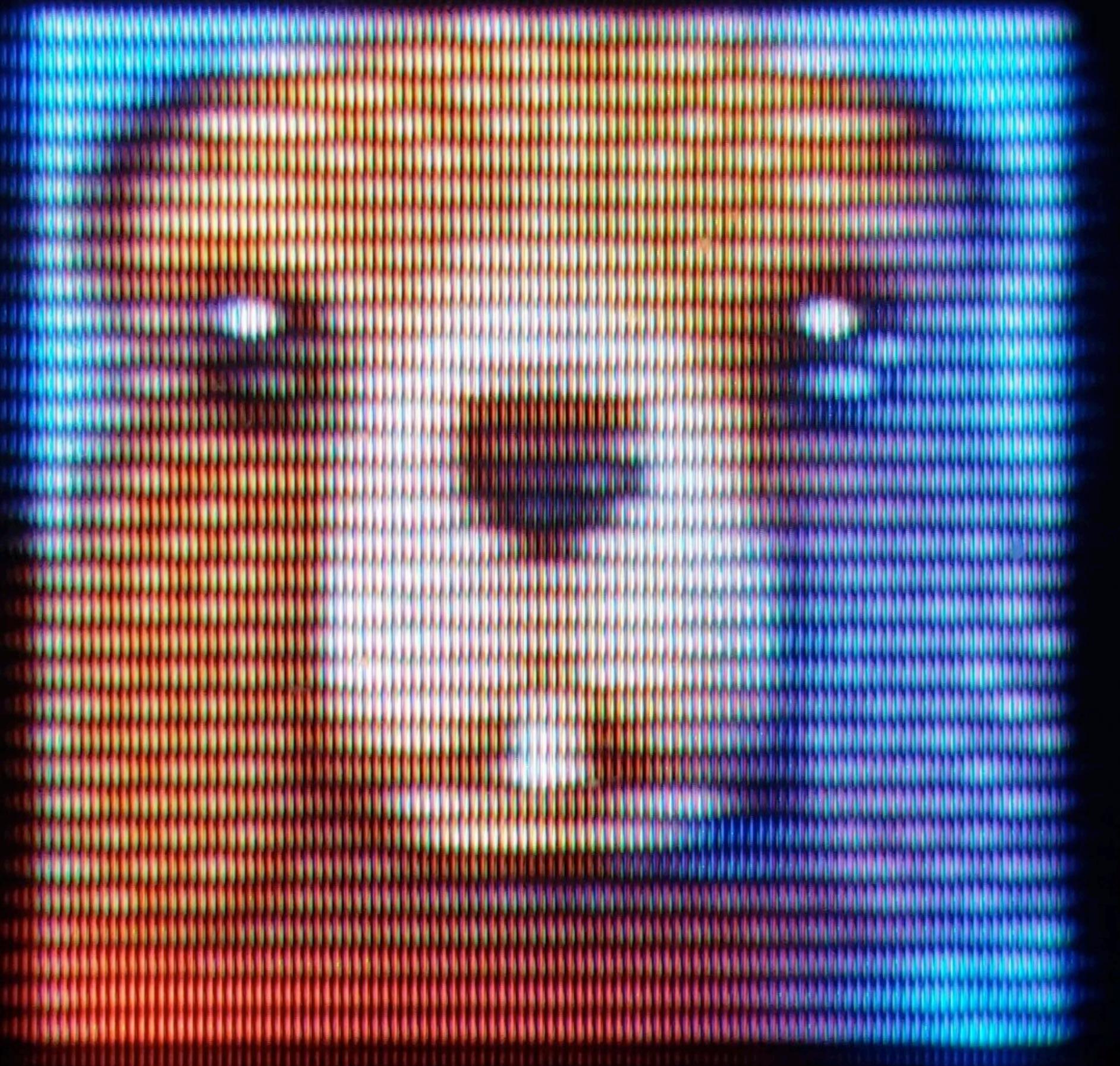
```
return AnimatedSampler(  
    child: const SomeWidget(),  
    (image, size, canvas) {  
        shader  
            ..setFloatUniforms((value) {  
                value.setSize(size);  
            })  
            ..setImageSampler(0, image);  
  
        canvas.drawRect(  
            Offset.zero & size,  
            Paint()..shader = shader,  
        );  
    },  
);
```



renan.gg



renan.gg



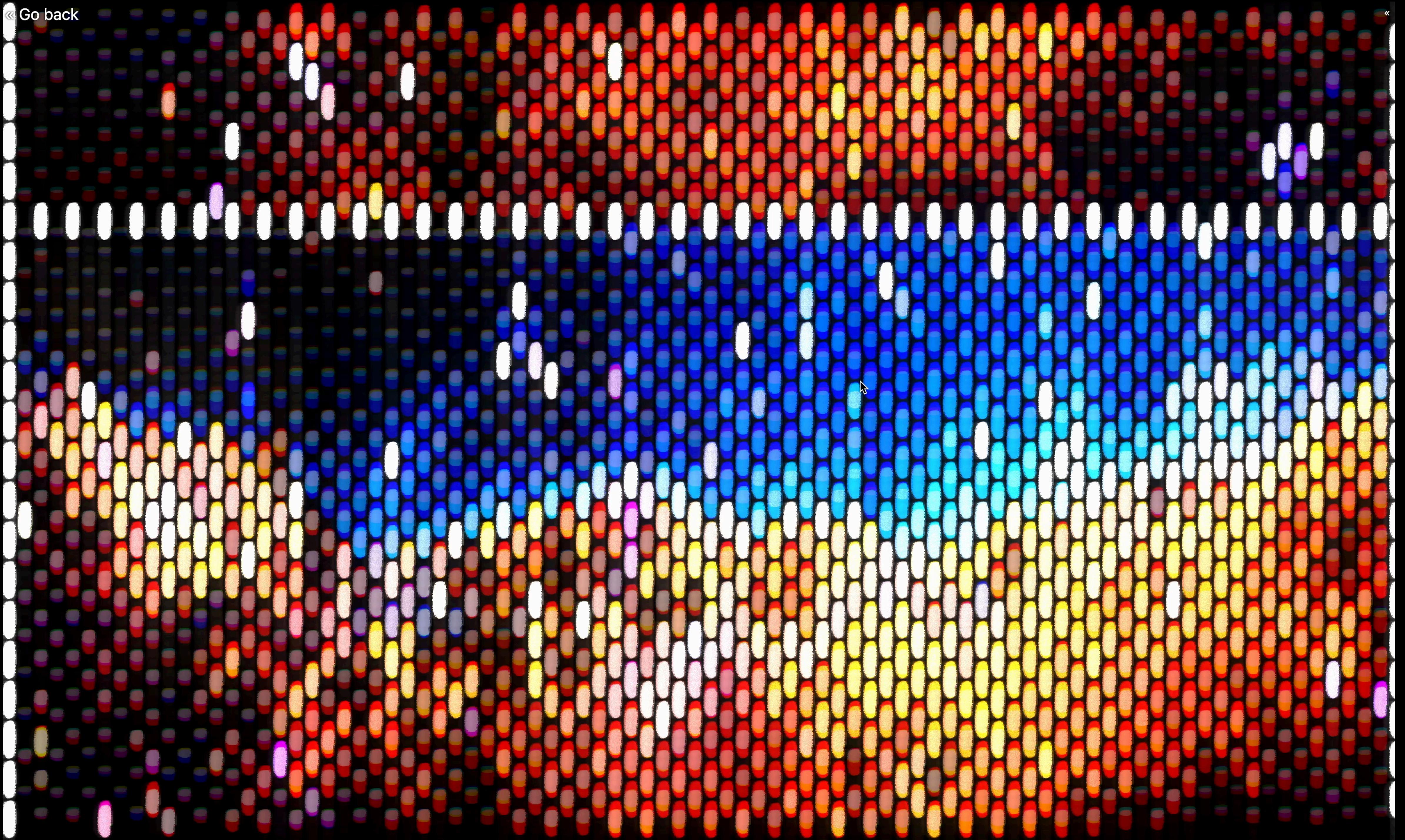
Actual close up photographs of a CRT display

by [@MrmoTarius](#) on Twitter

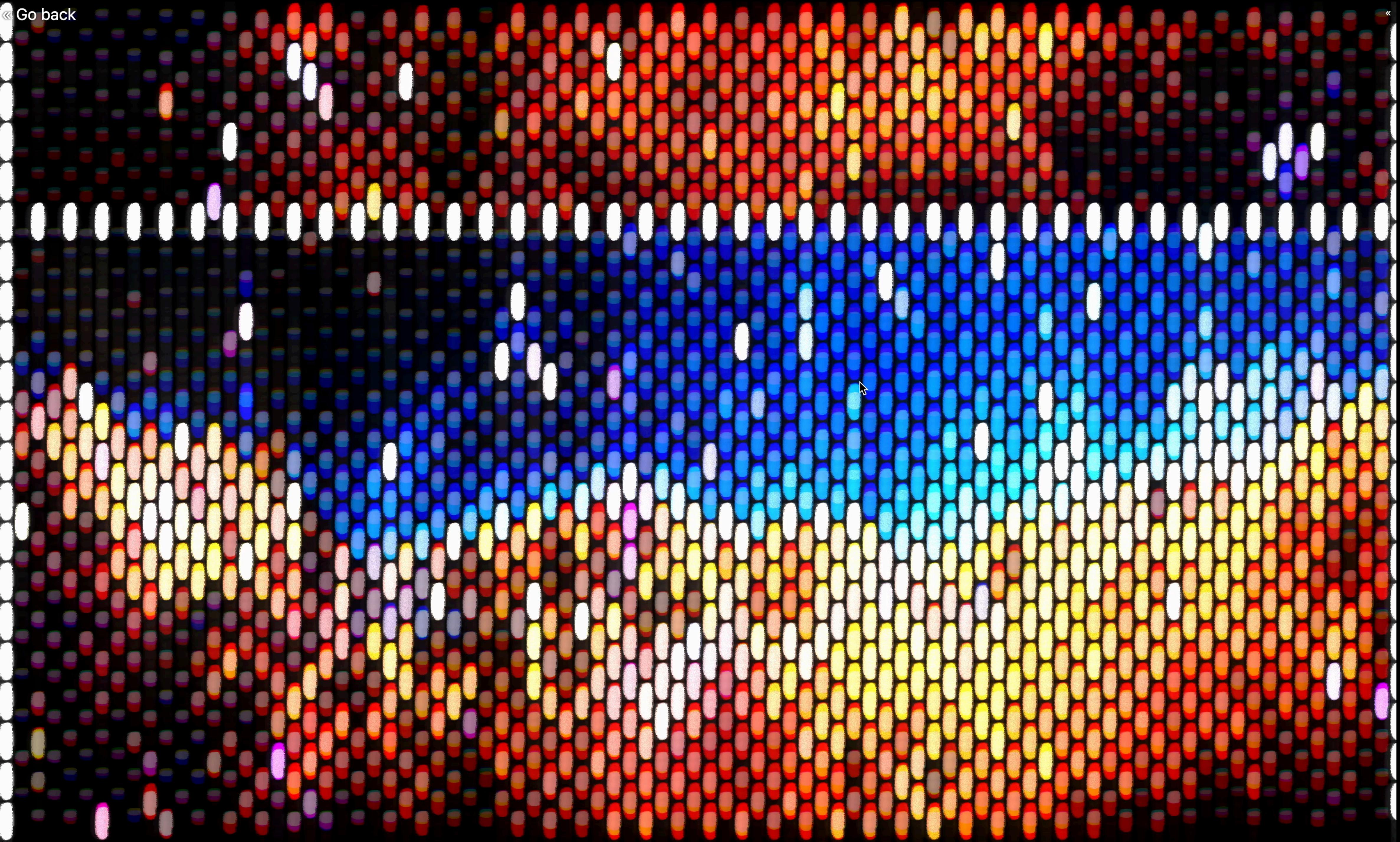
EXAMPLE 5

Nostalgia engineering

[Go back](#)



[Go back](#)





Lightrunners game on the Between invertase and VGV booth

renan.gg

Summary

- ✓ We learned that we can pass images to the shaders, and render them the way we want
- ✓ We learned that we can use the graphical result of entire widget trees as a texture uniform
- ✓ We learned that we can bring retro aesthetics to apps



renan.gg

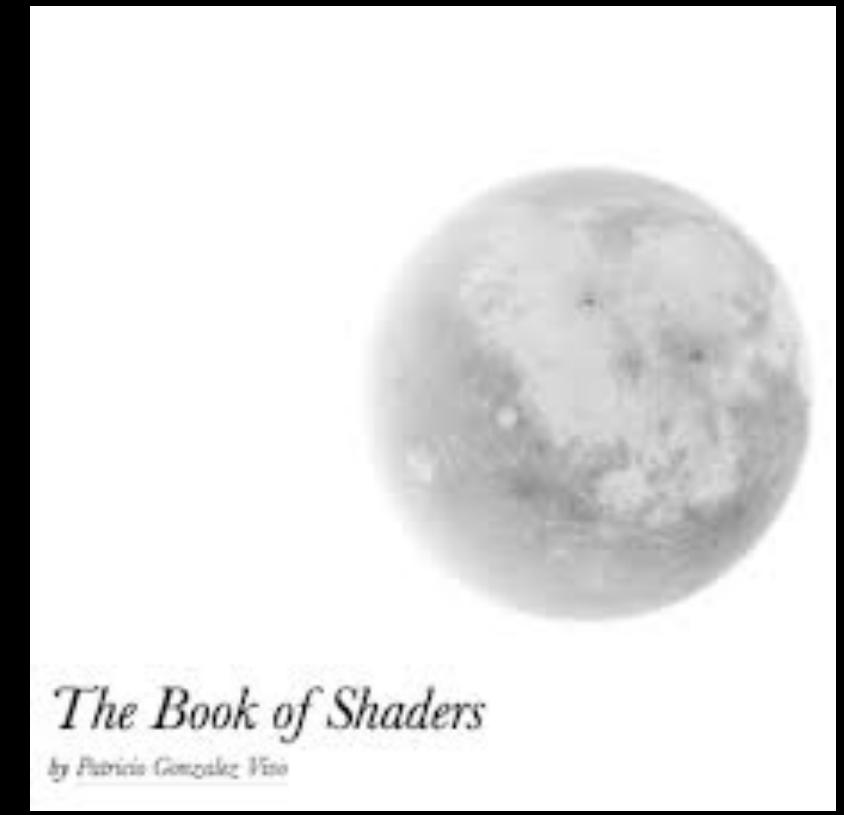


Go seek help

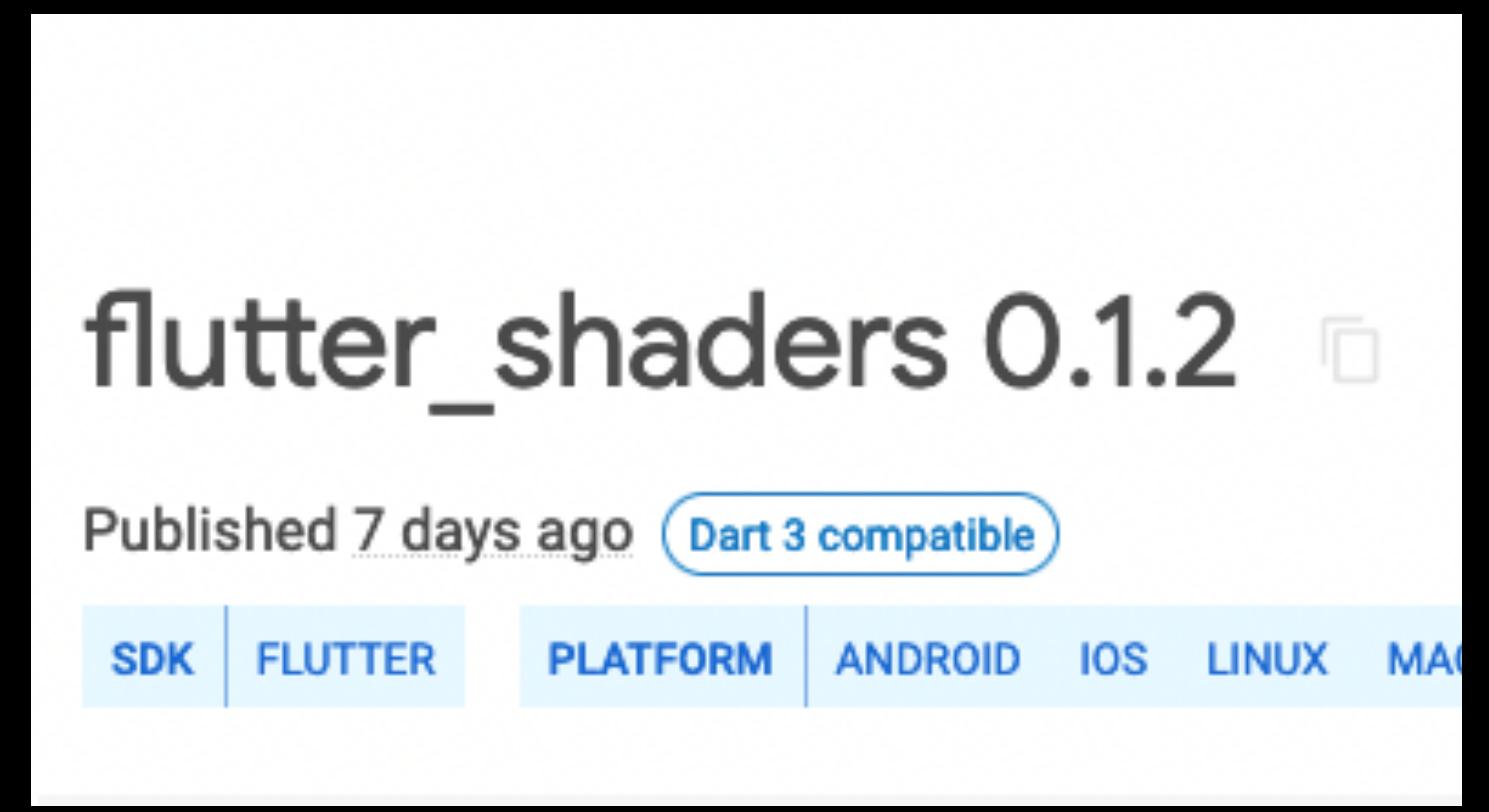
Further information



Building custom fragment shaders
“Flutter” on Youtube



The Book of shaders
thebookofshaders.com



flutter_shaders by Jonah Williams
Package on pub.dev

Every example will be open sourced
as well as this slide deck
as well as examples I didn't have time to finish talk

renan.gg