

# Shaders in SwiftUI

First steps and impressions

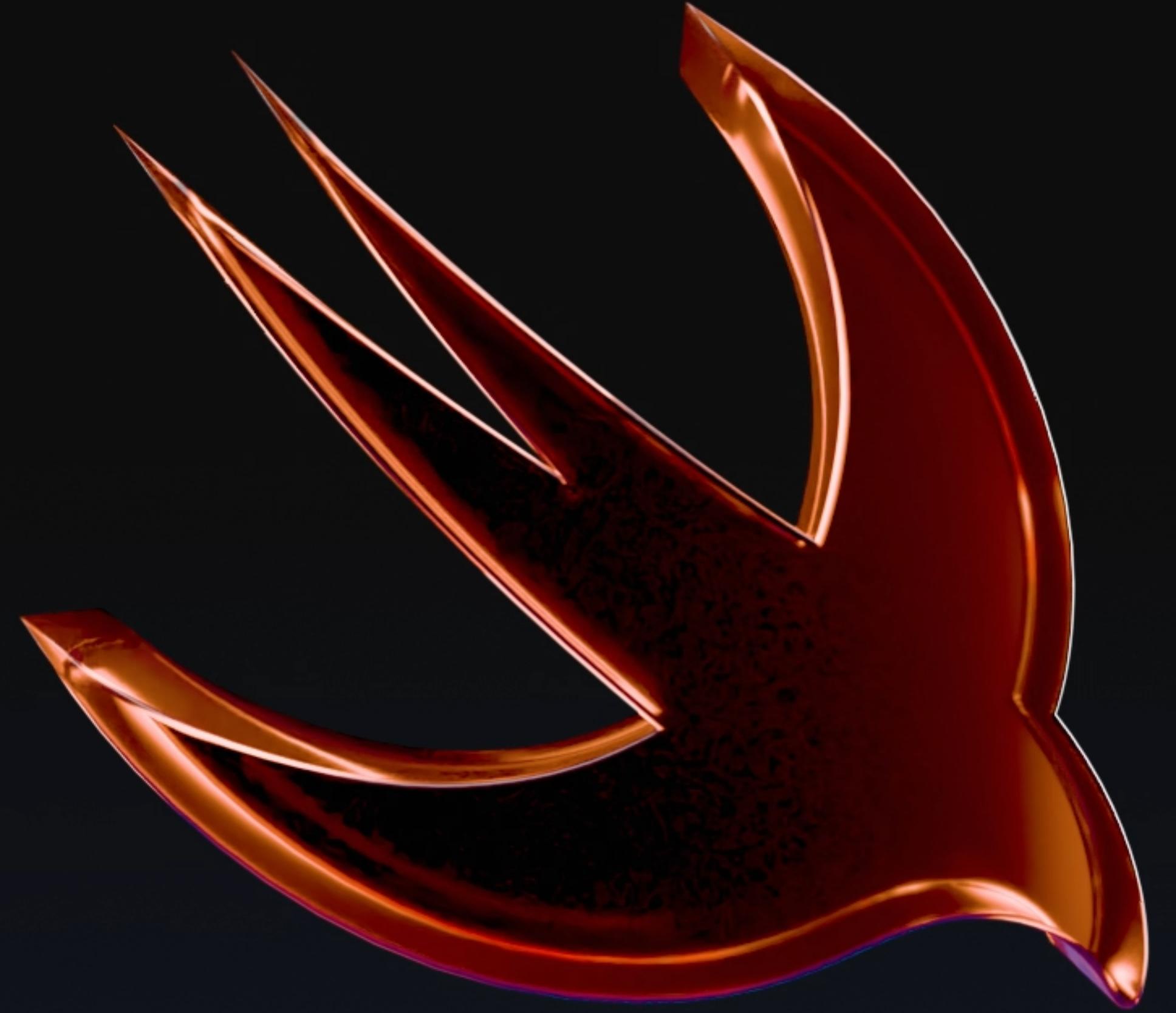
What are Shaders?

Shaders in SwiftUI + Demo

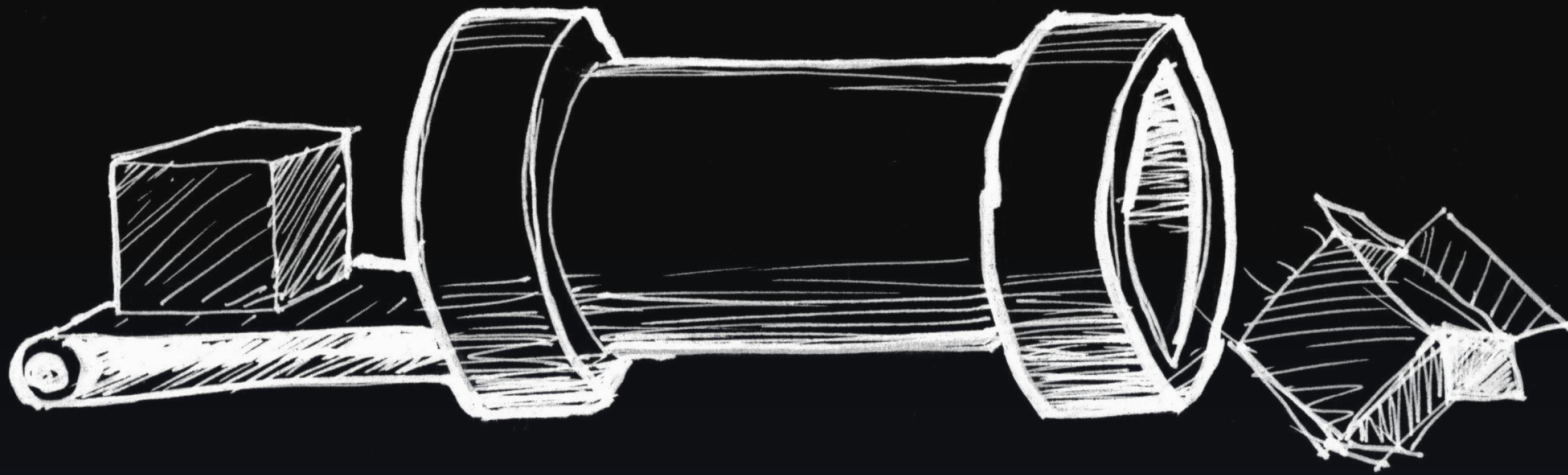
Creating a Basic Shader

Design Considerations

Resources



CPU



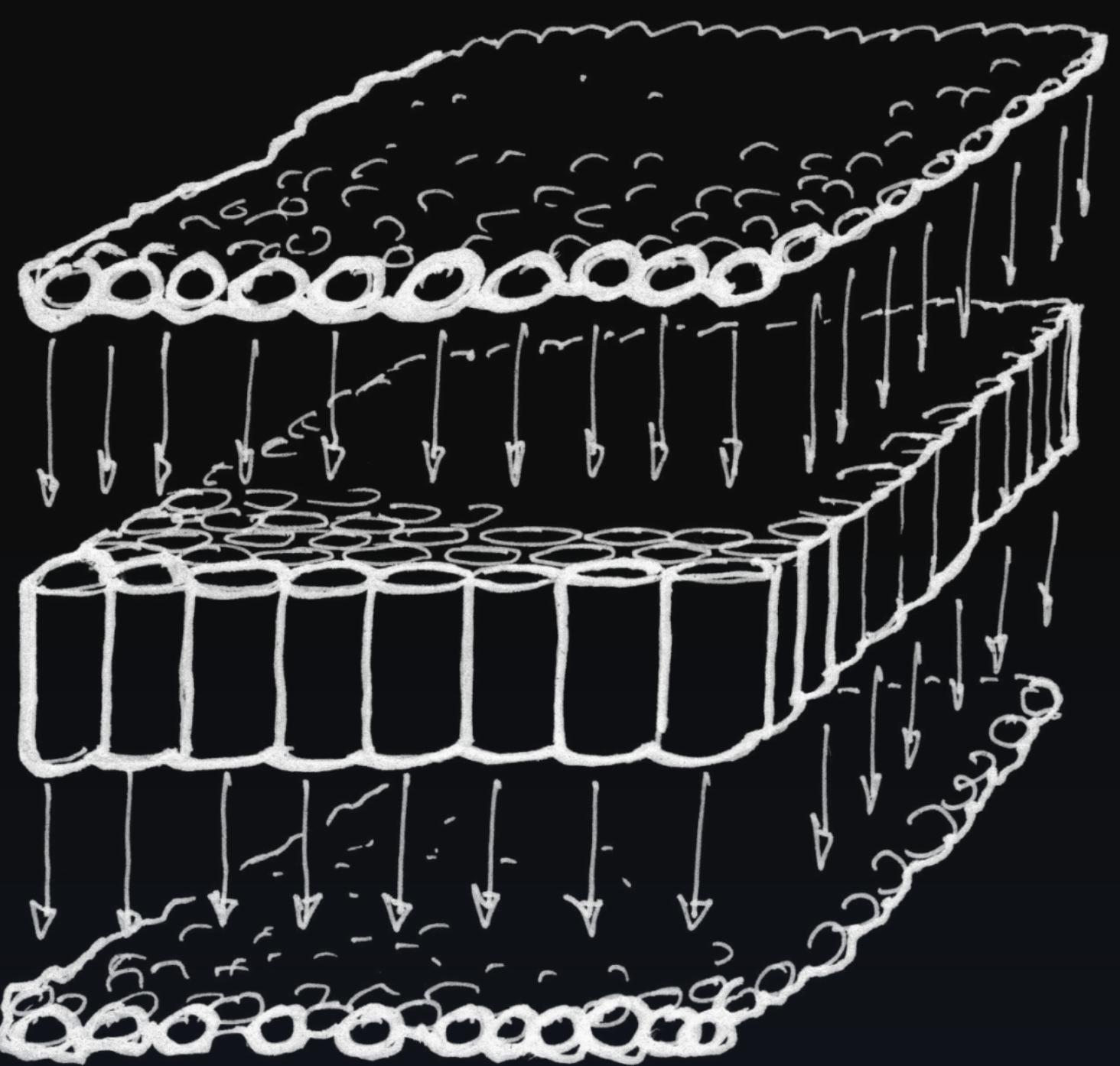
From *The Book of Shaders* by Patricio Gonzalez Vivo & Jen Lowe

CPU



From *The Book of Shaders* by Patricio Gonzalez Vivo & Jen Lowe

# GPU



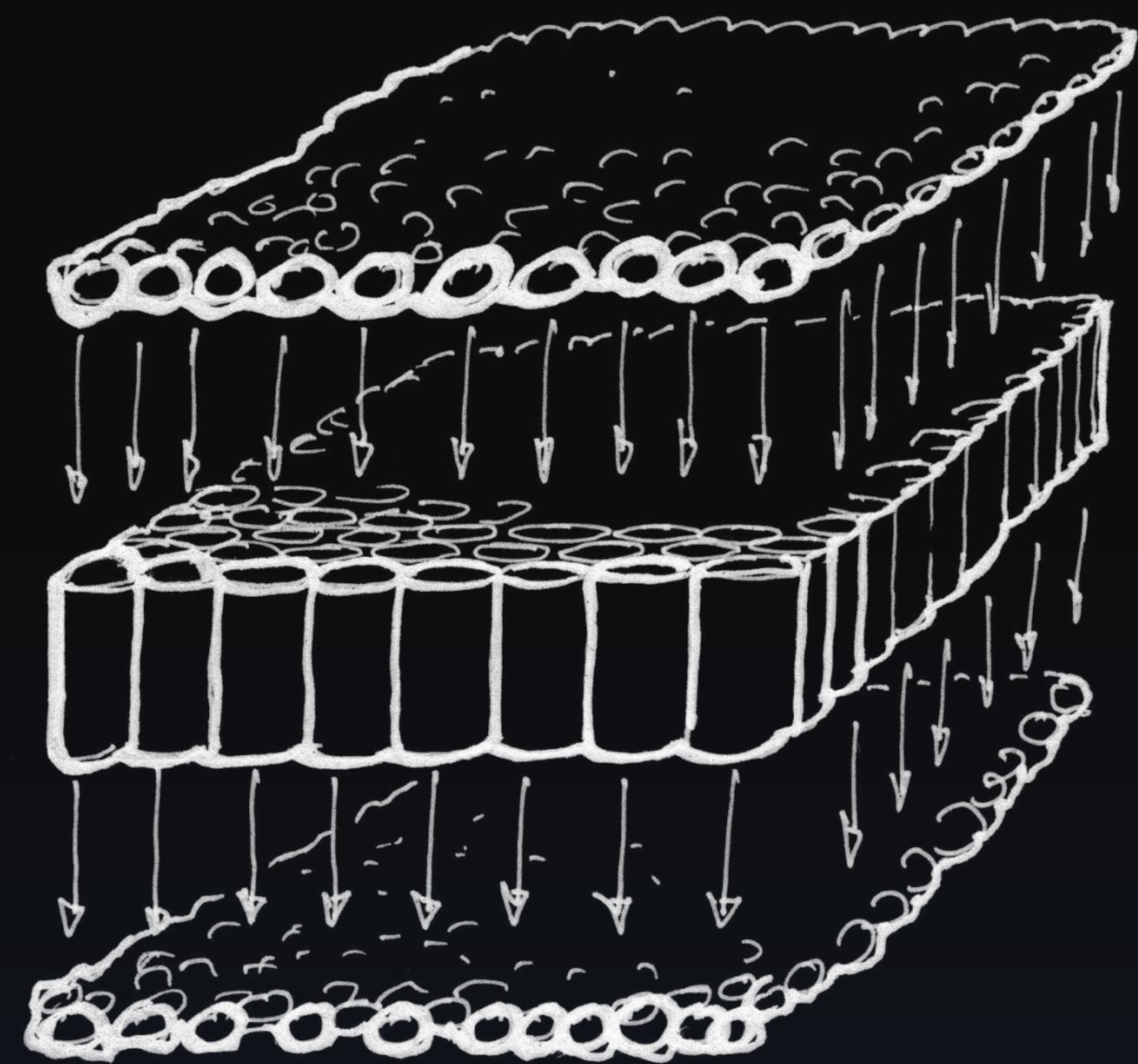
From *The Book of Shaders* by Patricio Gonzalez Vivo & Jen Lowe

# GPU

Parallel computing is ideal for handling pixels

Special math functions are hardware-accelerated

**Every thread runs independently**  
↳ blind and memoryless



From *The Book of Shaders* by Patricio Gonzalez Vivo & Jen Lowe

# Shading Languages

GLSL      OpenGL Shading Language

HLSL      High-Level Shading Language

OSL      Open Shading Language

...

MSL      Metal Shading Language

# Using Shaders in SwiftUI

`colorEffect(_:isEnabled:)`

`distortionEffect(_:maxSampleOffset:isEnabled:)`

`layerEffect(_:maxSampleOffset:isEnabled:)`

`visualEffect(_:)`

## colorEffect(\_:isEnabled:)

Returns a new view that applies shader to self as a **filter on the colour of each pixel**.

```
1 MyAwesomeView()  
2 .colorEffect(ShaderLibrary.myColorEffect())
```

## colorEffect(\_:isEnabled:)

```
1  [[ stitchable ]] half4 myColorEffect(float2 position, half4
2    color, args...) {
3
4    half4 newColor = color;
5
6    // Transform the color of each pixel
7
8    return newColor;
9 }
```

## distortionEffect(\_ :maxSampleOffset:isEnabled:)

Returns a new view that applies shader to self as a **geometric distortion effect on the location of each pixel.**

```
1 MyAwesomeView()  
2     .distortionEffect(ShaderLibrary.myDistortionEffect(),  
maxSampleOffset: .zero)
```

## distortionEffect(\_:maxSampleOffset:isEnabled:)

```
1  [[ stitchable ]] float2 myDistortionEffect(float2 position,  
args...) {  
2      float2 newPos = position;  
3  
4      // Transform the position of each pixel  
5  
6      return newPos;  
7 }
```

## layerEffect(\_:maxSampleOffset:isEnabled:)

Returns a new view that applies shader to self as a **filter on the raster layer created from self**.

```
1 MyAwesomeView()  
2     .layerEffect(ShaderLibrary.myLayerEffect(),  
maxSampleOffset: .zero)
```

## layerEffect(\_:maxSampleOffset:isEnabled:)

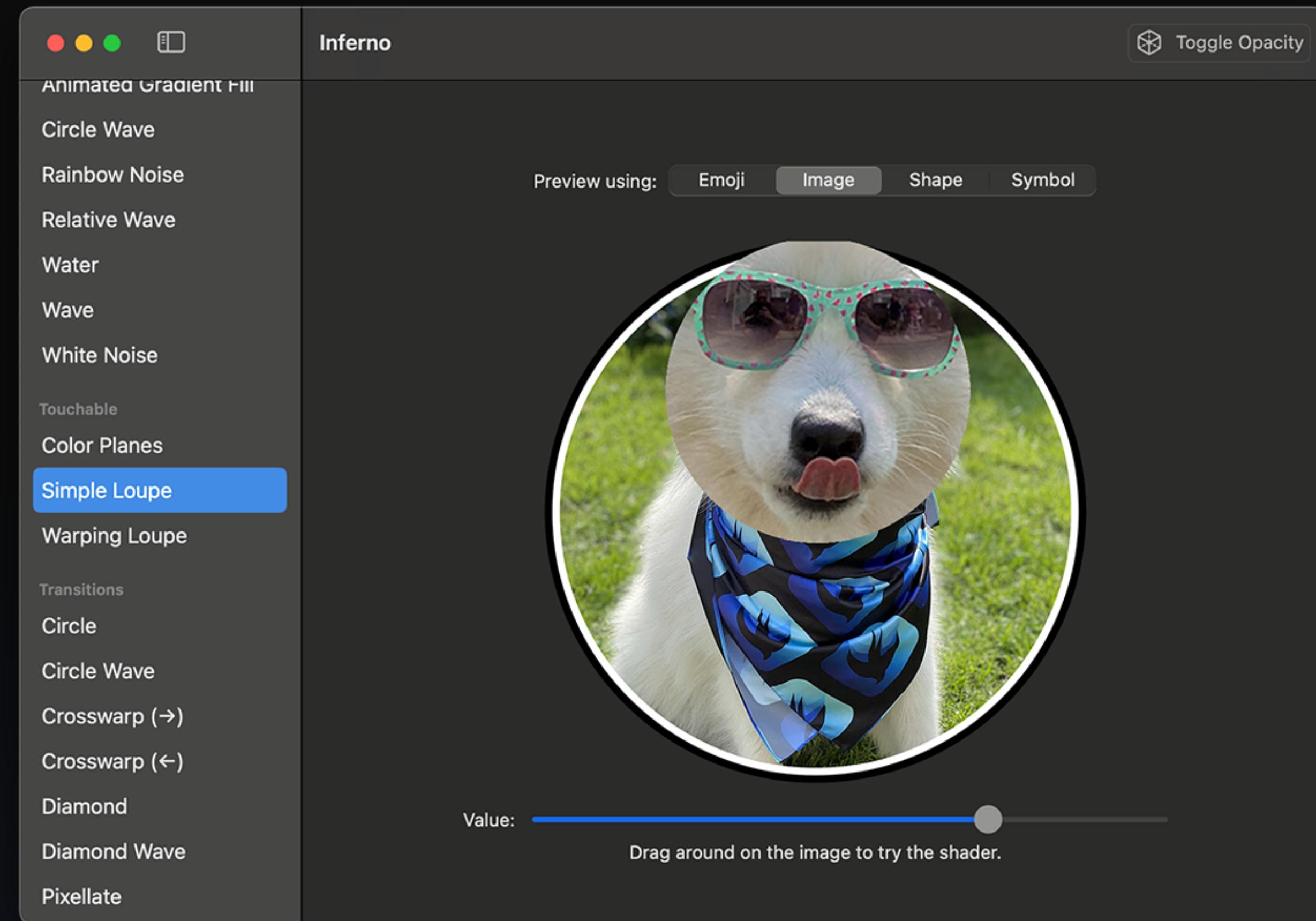
```
1 #include <SwiftUI/SwiftUI_Metal.h>
2
3 [[ stitchable ]] half4 myLayerEffect(float2 position,
SwiftUI::Layer layer, args...) {
4     // Sample color of layer at arbitrary position
5     half4 newColor = layer.sample(position);
6
7     // Transform the color of each pixel
8
9     return newColor;
10 }
```

## visualEffect(\_:)

Applies effects to this view, while providing **access to layout information through a geometry proxy**.

```
1 MyAwesomeView()  
2     .visualEffect { content, proxy in  
3         content.colorEffect(ShaderLibrary.myColorEffect(  
4             .float2(proxy.size)  
5         )  
6     }
```

# Inferno Sample App

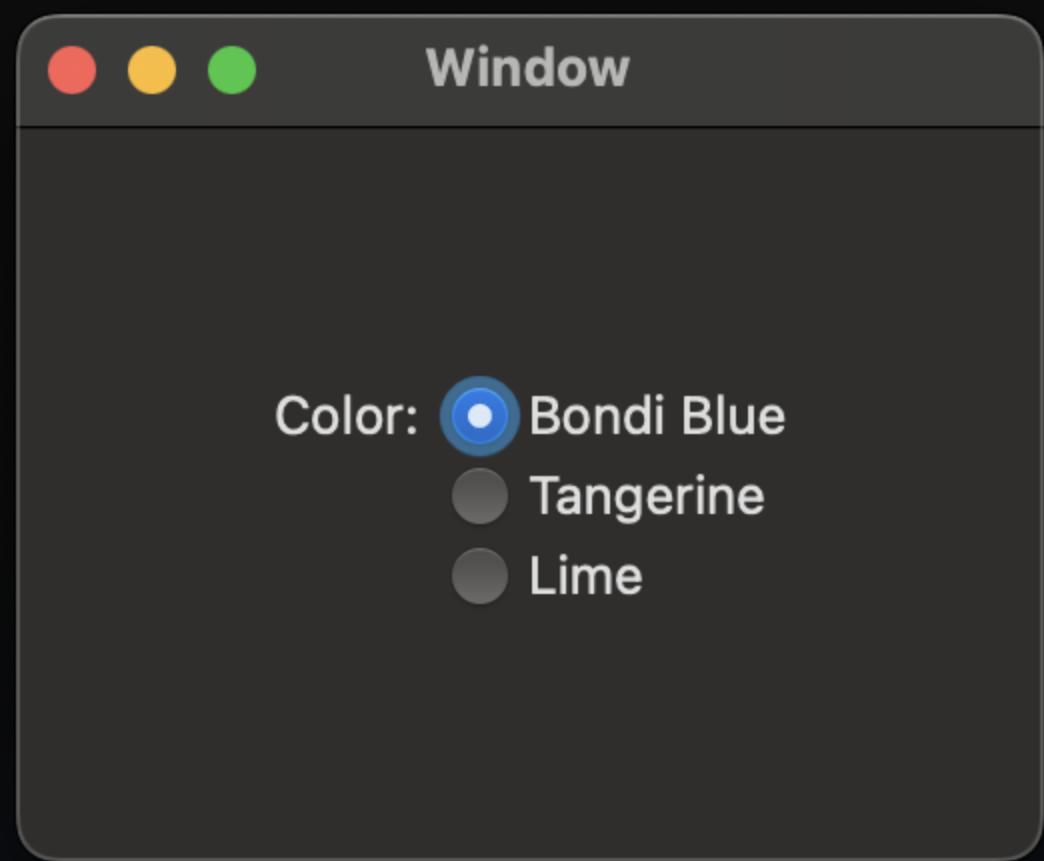


# Caveats

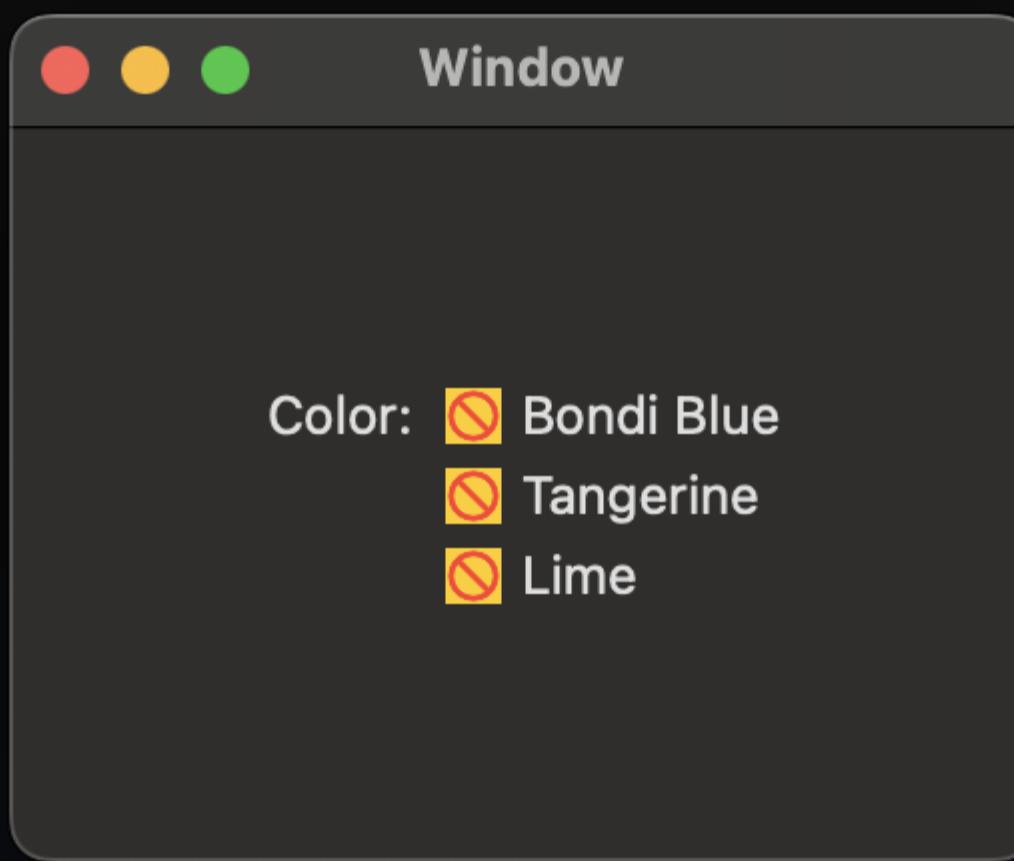
## Important

Views backed by AppKit or UIKit views may not render into the filtered layer. Instead, they log a warning and display a placeholder image to highlight the error.

# Caveats



AppKit-Backed Controls

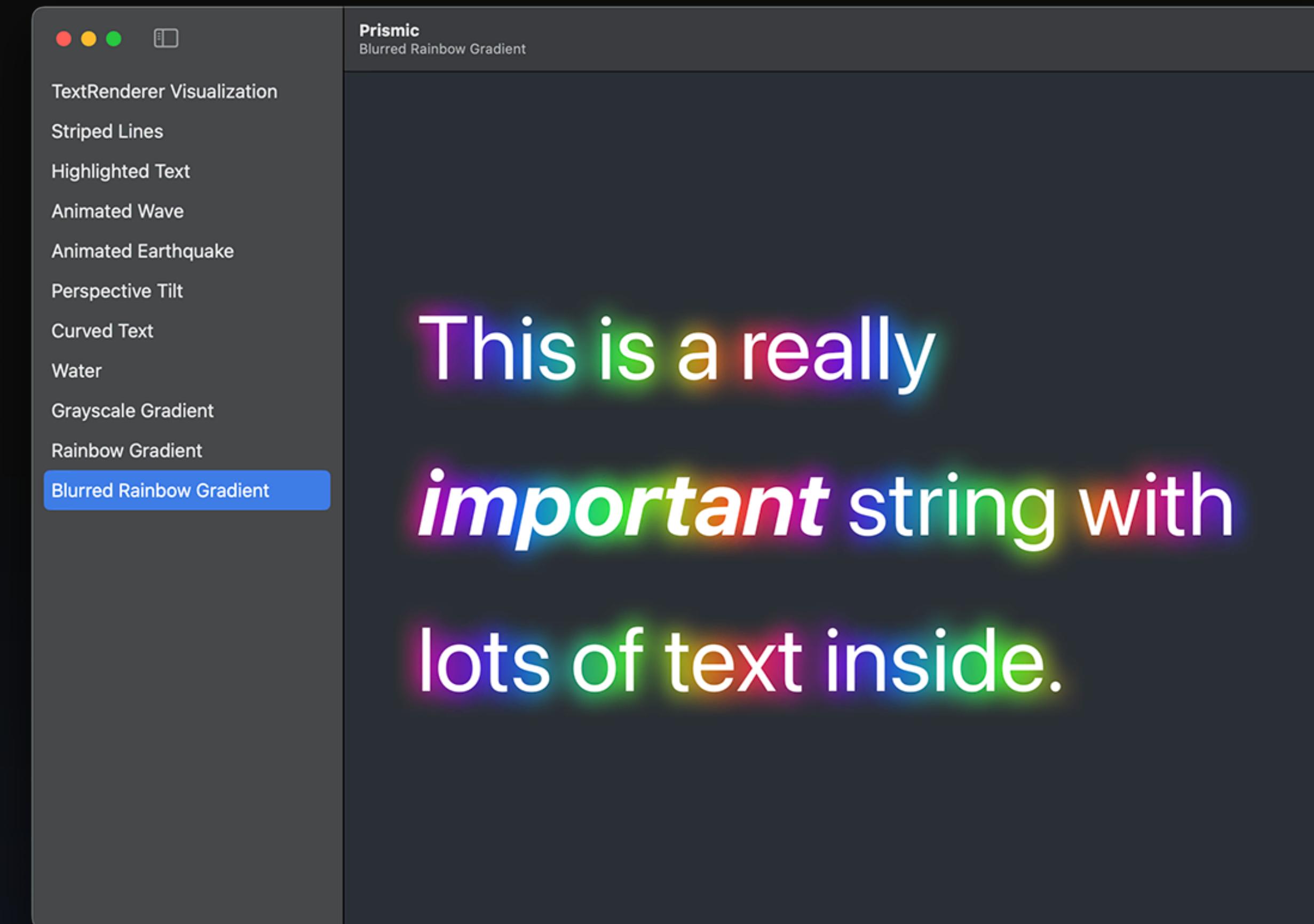


AppKit-Backed Controls  
with shader present

# TextRenderer

Renders text with visual effects  
(including shaders)

Can be applied to specific  
portions of the text



Prismic sample app by Paul Hudson

# Design Considerations

Use shaders purposefully

For decorative effects (in the background, for view transitions)

To emphasise user actions

To draw attention to a specific element

Keep content accessible

Avoid moving/distorting interactive elements

Maintain sufficient contrast to foreground elements

Respect user's accessibility settings (e.g., Reduce Transparency, Reduce Motion)

# Resources

The Book of Shaders  
Patricio Gonzalez Vivo & Jen Lowe

[thebookofshaders.com](http://thebookofshaders.com)

Inferno sample app  
Paul Hudson

[github.com/twostraws/Inferno](https://github.com/twostraws/Inferno)

Shadertoy

[shadertoy.com](https://shadertoy.com)

Thanks!