


```
val shader = RuntimeShader("...shader code ...")
val brush = ShaderBrush(shader)

Sketch(
    onDraw = { time →
        // Get dimensions from DrawScope.size
        shader.setFloatUniform(
            "iResolution",
            size.width, size.height
        )

        // From Sketch!
        shader.setFloatUniform("iTime", time)

        drawRect(brush)
    }
)
```



[The following text is a dense, handwritten manuscript, likely a letter or a page from a book. It is written in a cursive script and is mostly illegible due to the quality of the scan. The text appears to be a continuous paragraph, possibly discussing a topic related to the page number 10. The handwriting is very close together, and the ink is dark, making it difficult to discern individual words. The overall appearance is that of a historical document or a personal letter.]






```
val shader = RuntimeShader("...shader code ...")
val brush = ShaderBrush(shader)

Sketch(
    onDraw = { time →
        // Get dimensions from DrawScope.size
        shader.setFloatUniform(
            "iResolution",
            size.width, size.height
        )

        // From Sketch!
        shader.setFloatUniform("iTime", time)

        drawRect(brush)
    }
)
```



Shaders in 5 Steps 🤯

1. thebookofshaders.com !
2. use shadertoy.com & shaders.skia.org as a playground
3. GLSL - convert to AGSL; SKSL - use as is
4. plug into **RuntimeShader**
5. use size for **iResolution**, Sketch, or any animating time value for **iTime**