

// AGSL 

```
uniform float2 iResolution; // Viewport resolution (px)
```

```
uniform float iTime; // Shader playback time (s)
```

```
vec4 main(in float2 fragCoord) {
```

```
    // Normalized pixel coordinates (from 0 to 1)
```

```
    vec2 uv = fragCoord/iResolution.xy;
```

```
    // Time varying pixel color
```

```
    vec3 col = 0.8
```

```
    + 0.2*cos(iTime*2.0+uv.xxx*2.0+vec3(1,2,4));
```

```
    // Output to screen
```

```
    return vec4(col,1.0);
```

```
}
```



[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 or a series of lines of writing.]





///

AGSL



uniform float2 in Resolution; // Viewport resolution(px)

```
vec4 main(in float2 fragCoord) {
```

uniform float iTIME; // Shader playback time (s)

`vec2 uv = fragCoord / iResolution.xy;`

// Normalized pixel coordinates (from 0 to 1)

vec5col = 0.8



+0.2*cos(iTine*2.0+uv.xxx*2.0+vec3(1,2,4));

///Output to screen

return vec4(col, 1.0);



val shader = RuntimeShader('...shader.cde...')

```
val brush = ShaderBrush(shader)
```

onDraw \equiv { time \rightarrow

Sketch((

//Get dimension from DrawScope.size

shader.setUniform(





"iRResolutiveion",



```
shader.setFloatUniform("itime", time)
```

size.width, size.height



drawRect(ctx)



