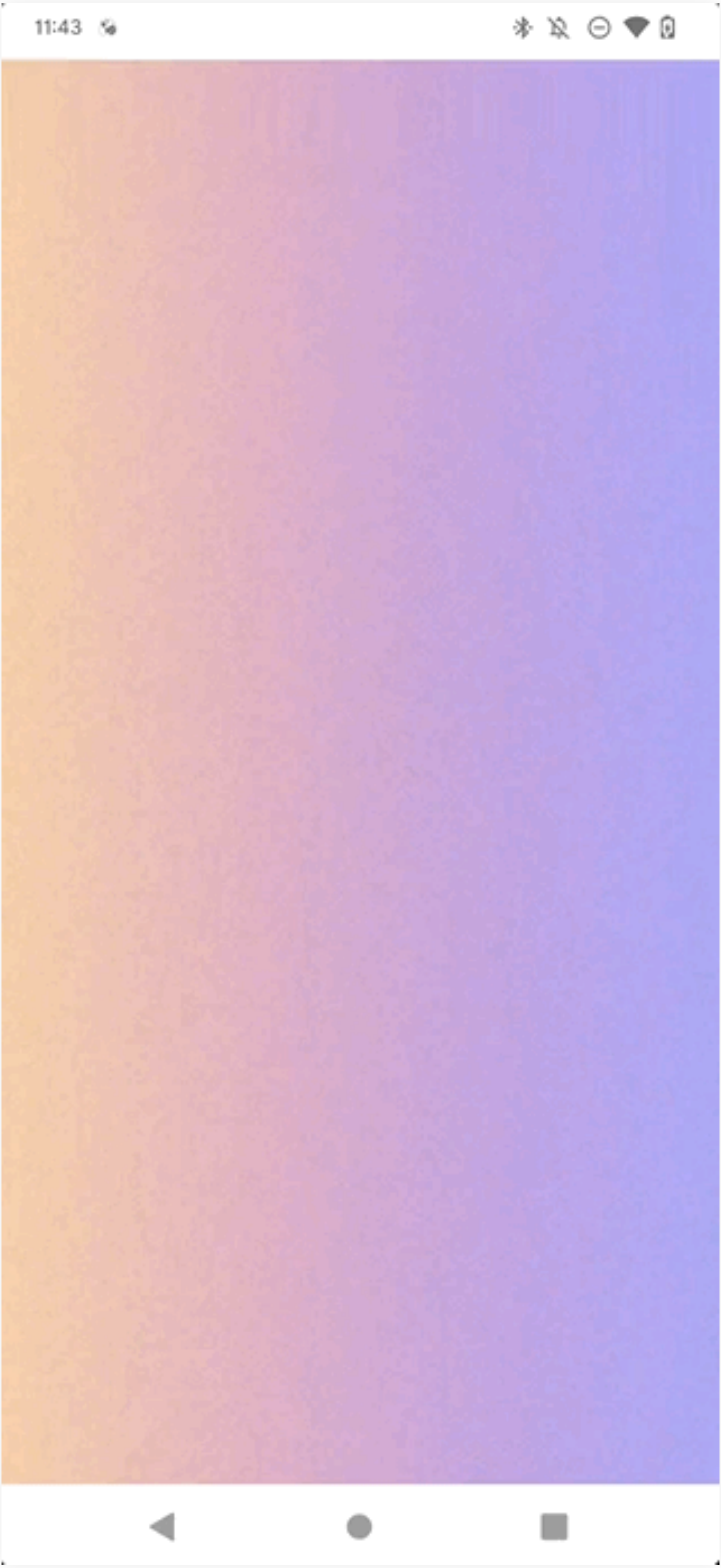
```
// AGSL ⬇️

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);
}
```


11:43

[redacted]

11:43

11:43

// AGSL ⬇️

```
uniform float2 iResolution; // 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)
```

```
vec3 col = 0.8
```

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

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

```
//  Output to screen
```

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

}

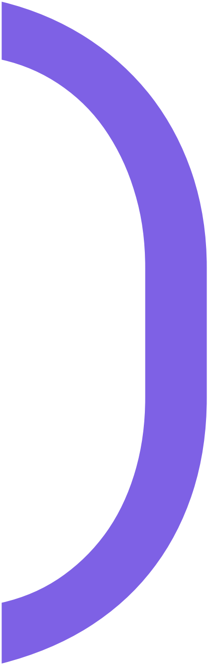
```
val shader = RuntimeShader(“...shader code ...”)
val brush = ShaderBrush(shader)
```

```
onDraw = { time →
```

Sketch

```
// Get dimensions from DrawScope.size
```

```
shader.setFloatUniform(
```



```
"iResolution",
```

// From Sketch!

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

```
size.width, size.height
```

}

drawRect(brush)