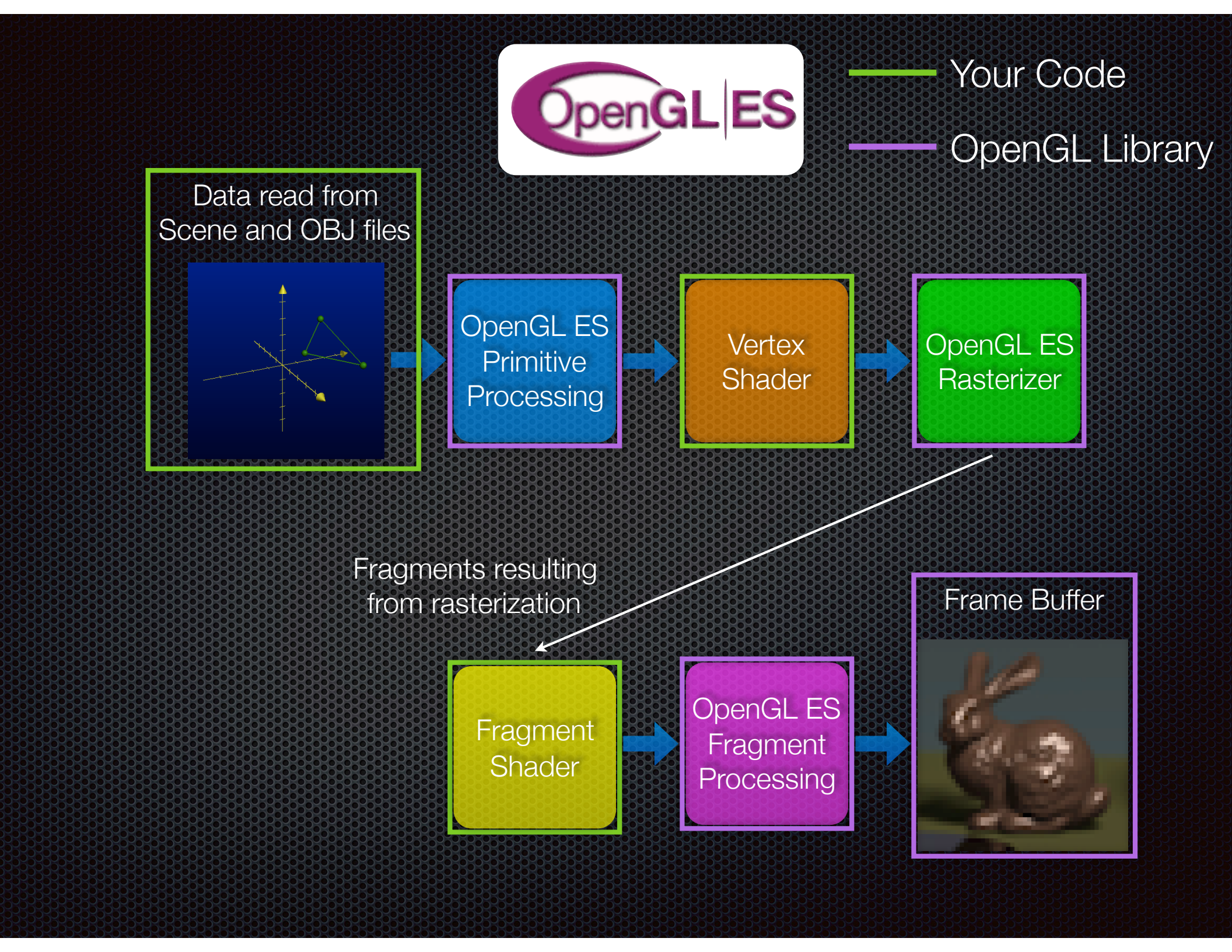
Mobile Application Programming

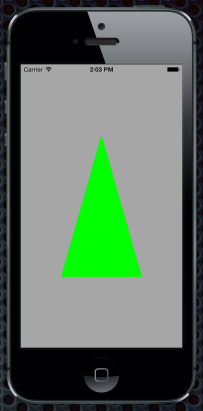
OpenGL ES Shaders

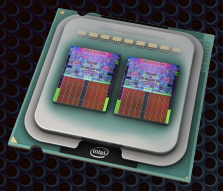
OpenGL Environment

- UIWindow
  - Root VC - GLKViewController
    - GLKView
      - Vertex Shader
      - Fragment Shader
      - Program
        - Uniform Variables
        - Attribute Arrays

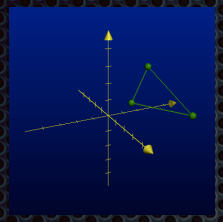

OpenGL ES
Your Code
OpenGL Library
Data read from Scene and OBJ files
OpenGL ES Primitive Processing
Vertex Shader
OpenGL ES Rasterizer
Fragments resulting from rasterization
Fragment Shader
OpenGL ES Fragment Processing
Frame Buffer

Pass-Through Program





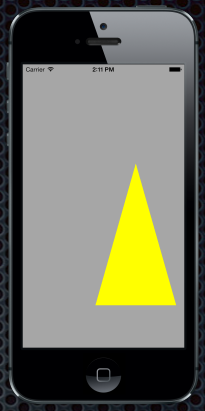
```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glDrawArrays(GLenum(GL_TRIANGLES), 0, 3)
```

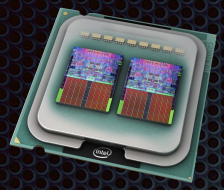


```
attribute vec2 position;
void main()
{
    gl_Position = vec4(position.x, position.y, 0.0, 1.0);
}
```

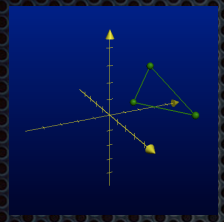
```
void main()
{
    gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
}
```

Uniform Program





```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glUniform4f(glGetUniformLocation(_program, "color"), 1.0, 1.0, 0.0, 1.0)
glDrawArrays(GL_TRIANGLES, 0, 3)
```



```
attribute vec2 position;
uniform vec2 translate;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
}
```

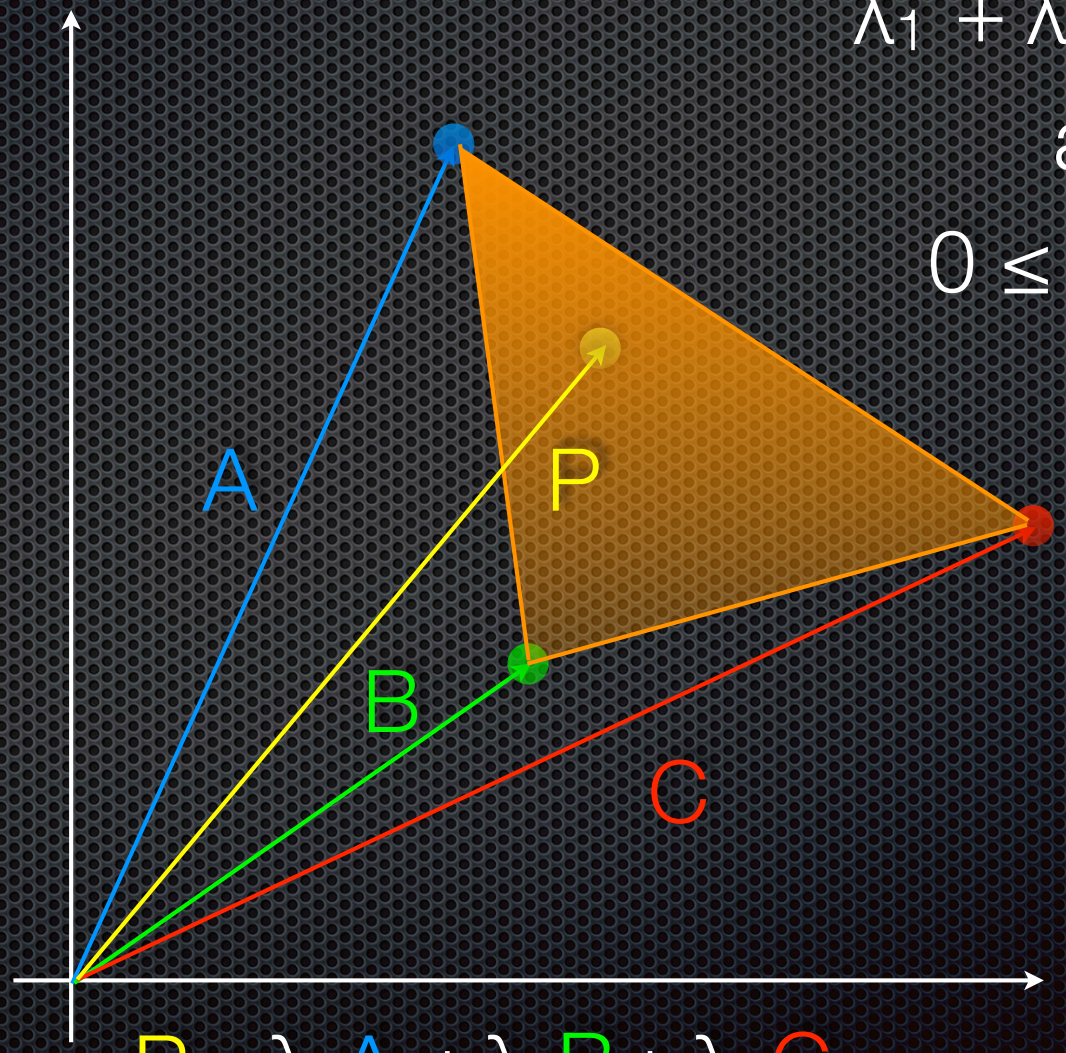
```
uniform highp vec4 color;
void main()
{
    gl_FragColor = color;
}
```

Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

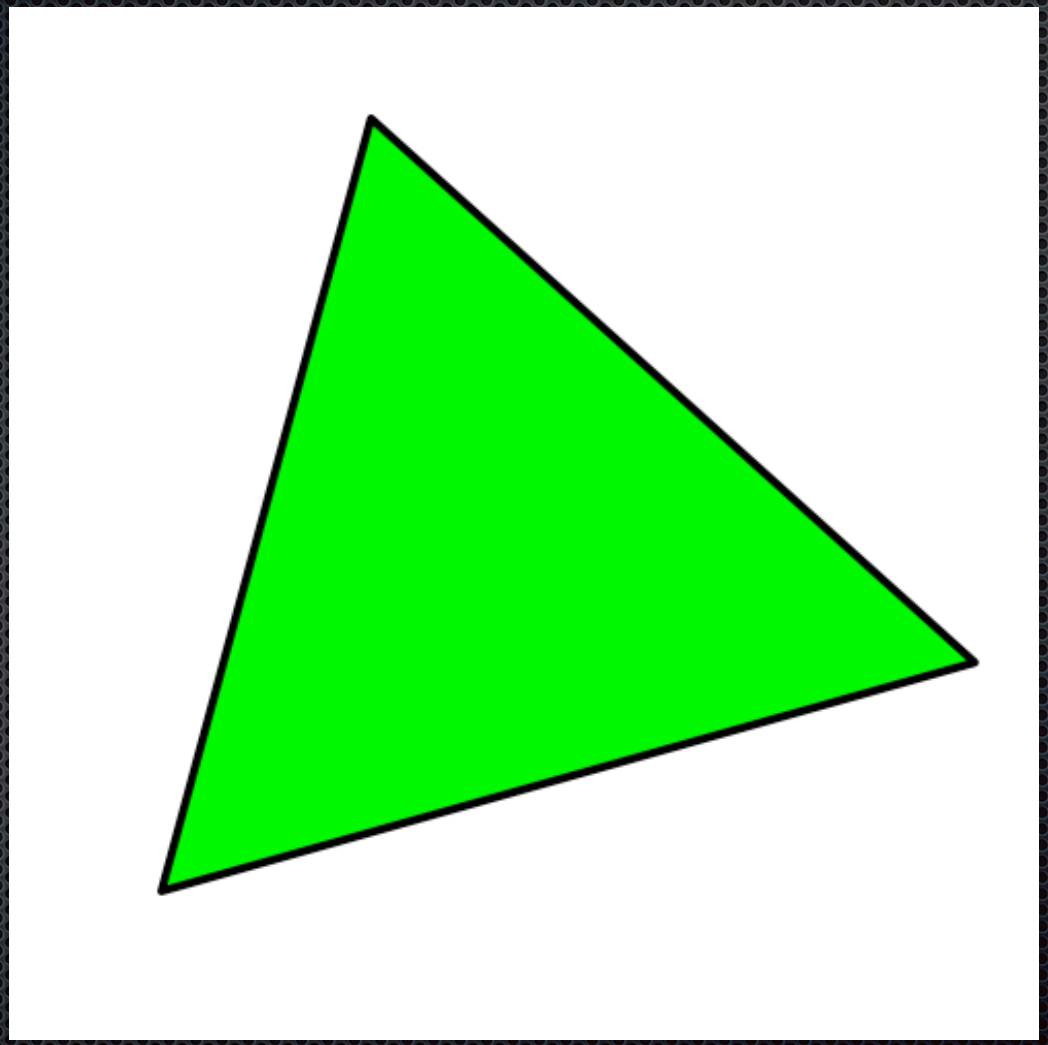
and

$0 \leq \lambda_x \leq 1$

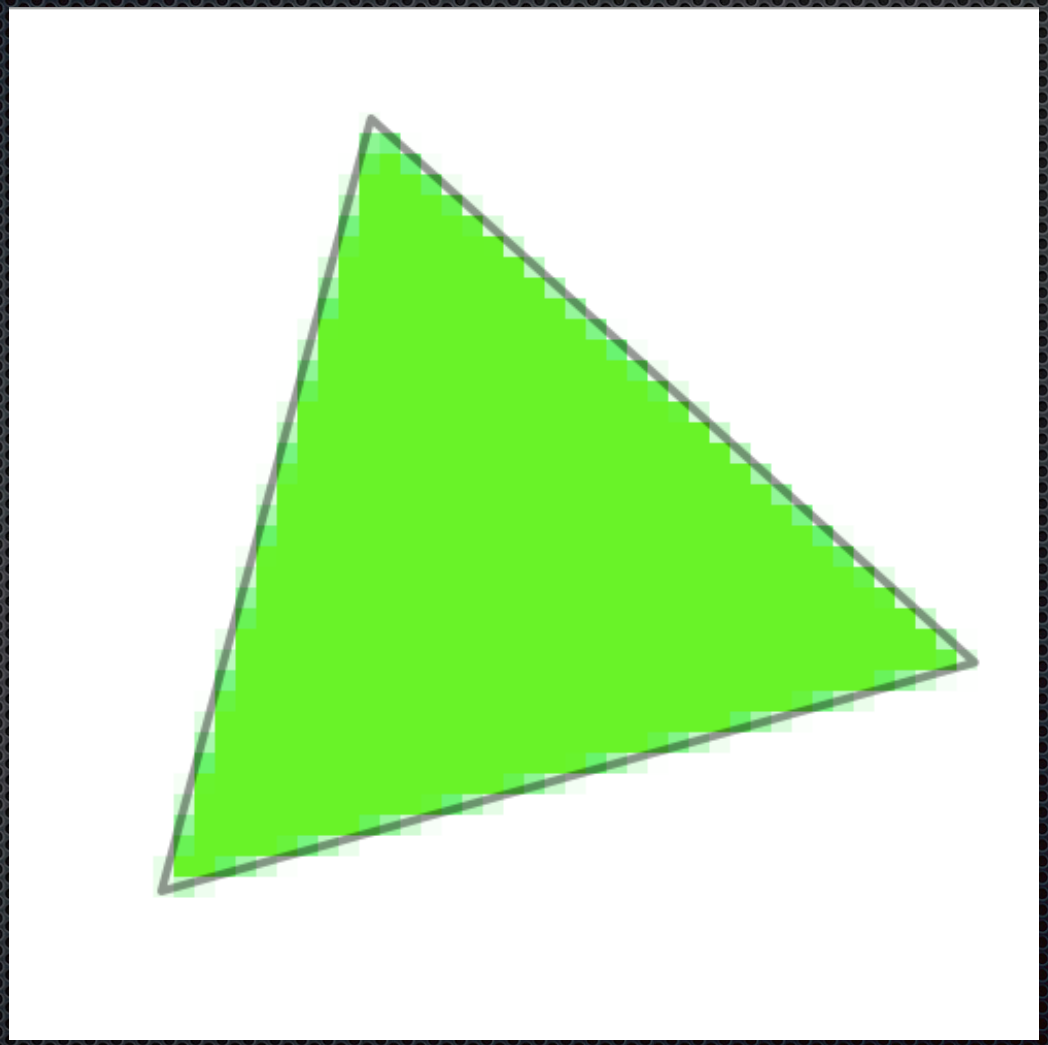


$P = \lambda_1 A + \lambda_2 B + \lambda_3 C$

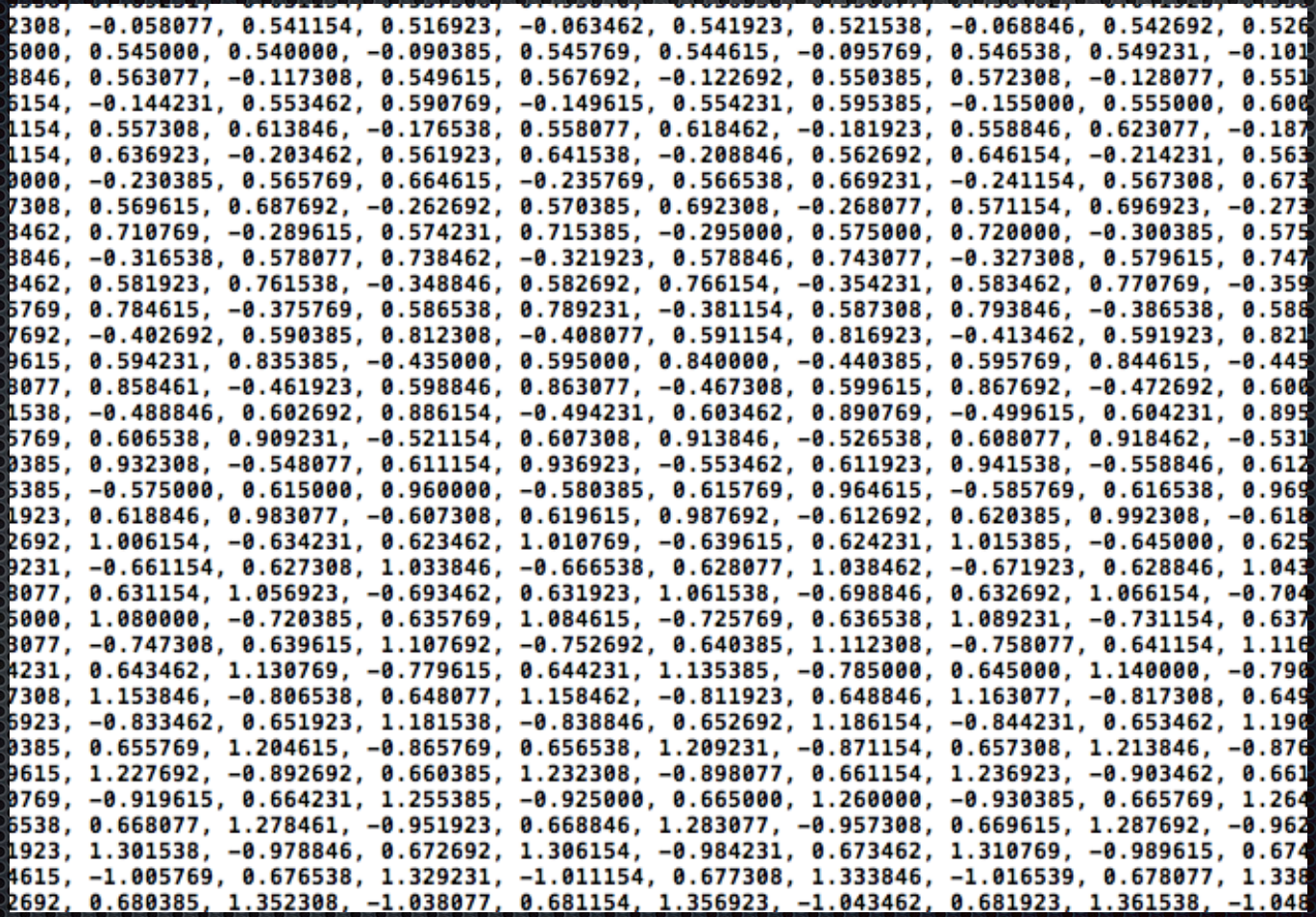
Rasterization



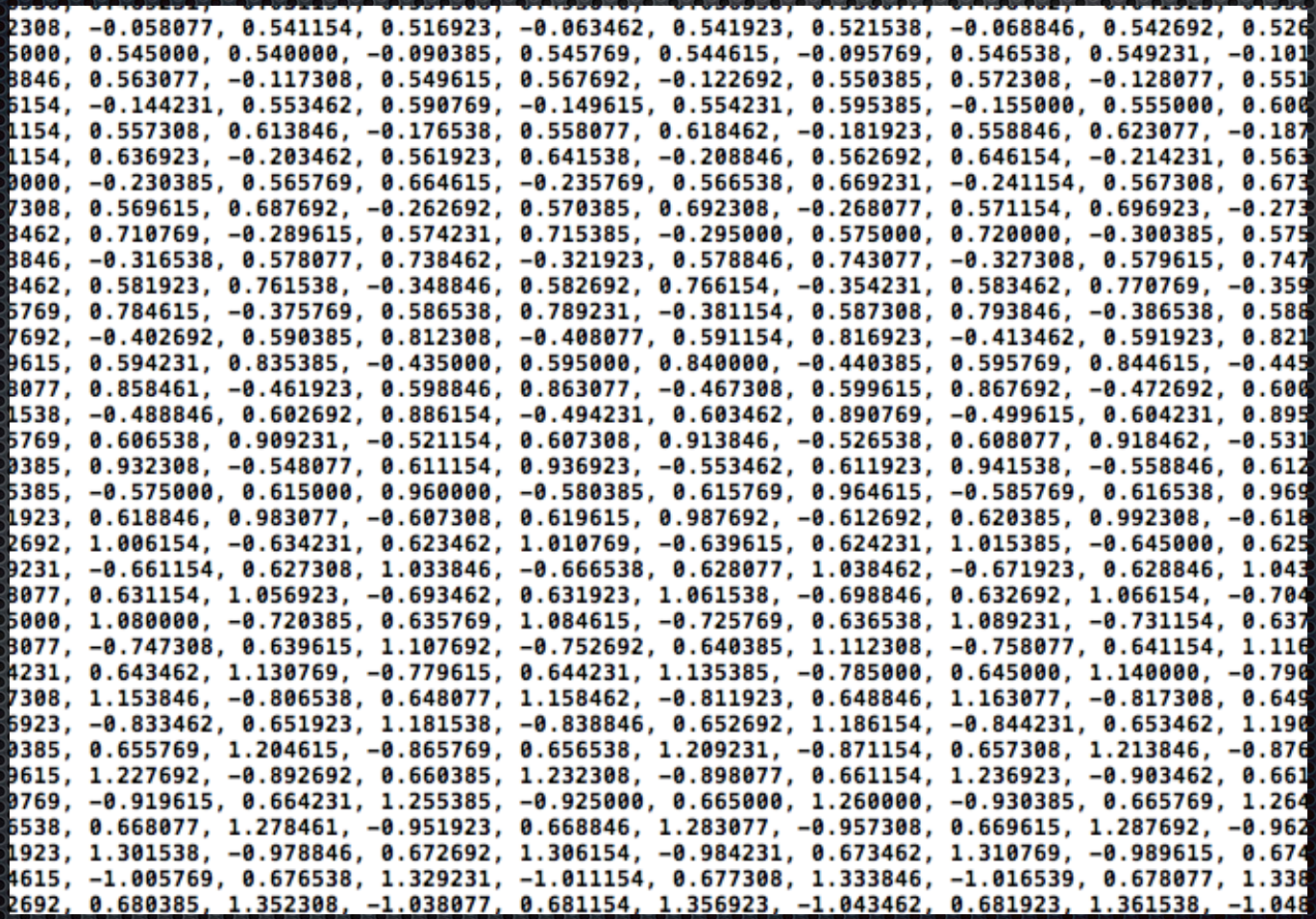
Rasterization



Barycentric Coordinates



Barycentric Coordinates



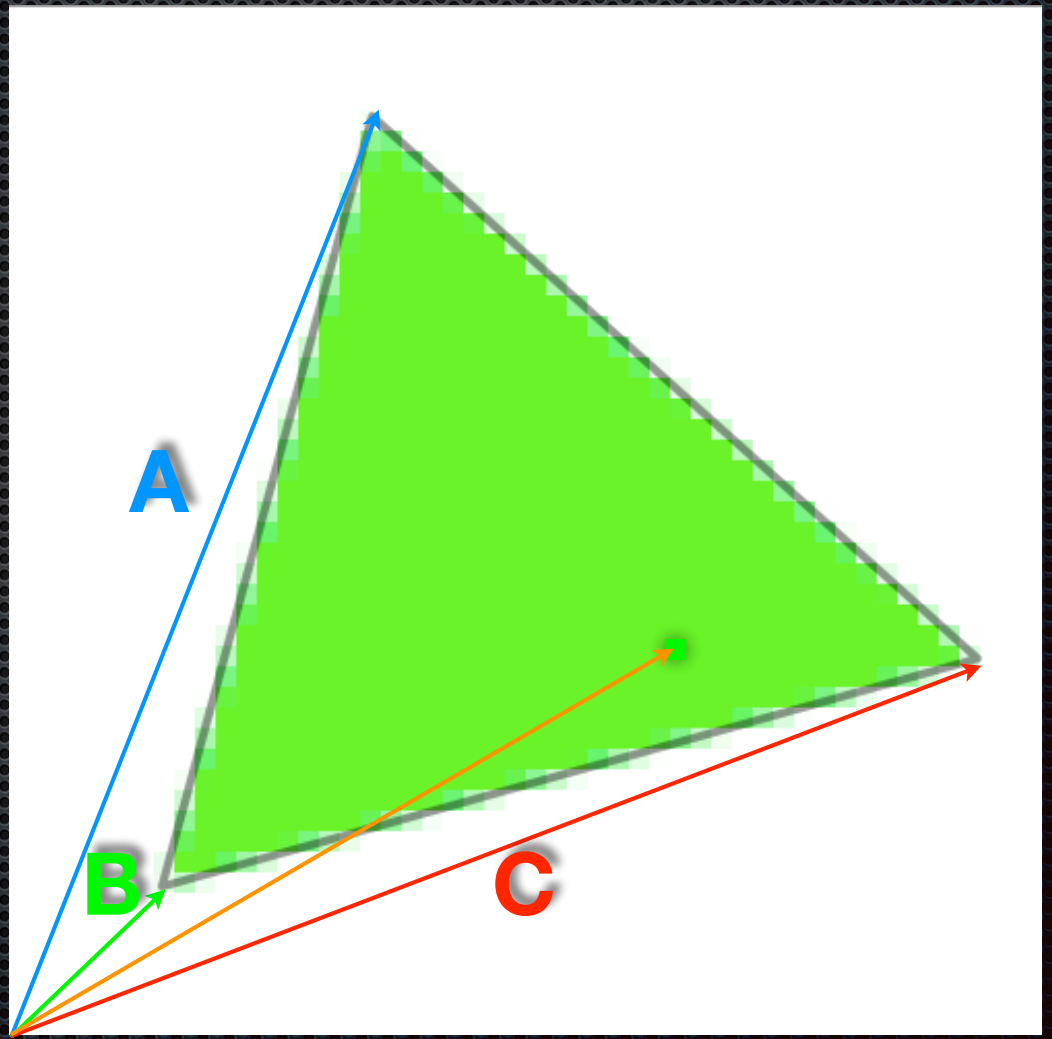
What else can we do with this data?

Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

$0 \leq \lambda_x \leq 1$



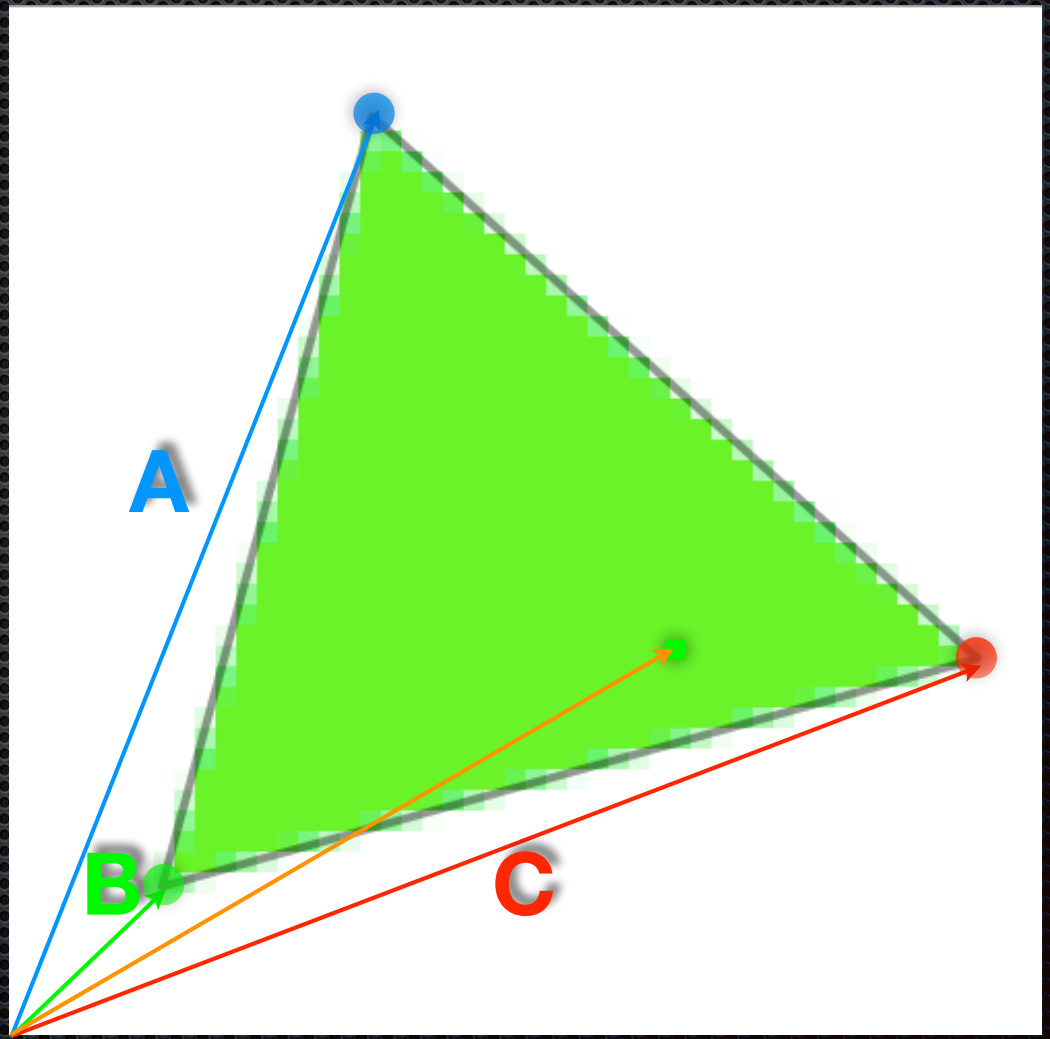
Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

$0 \leq \lambda_x \leq 1$

Associate a color with each point



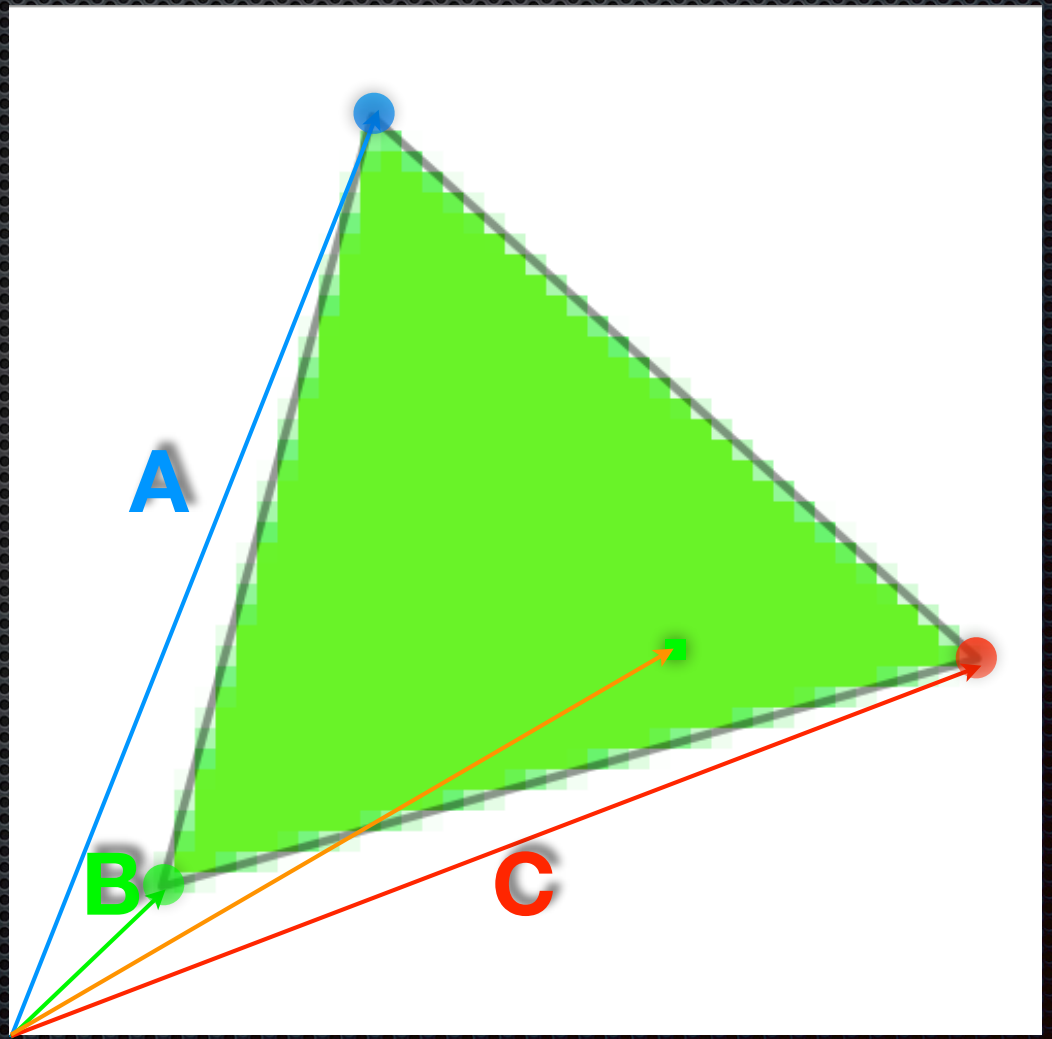
Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

$0 \leq \lambda_x \leq 1$

- $0 \leq \lambda_1 \leq 1$
- $0 \leq \lambda_2 \leq 1$
- $0 \leq \lambda_3 \leq 1$



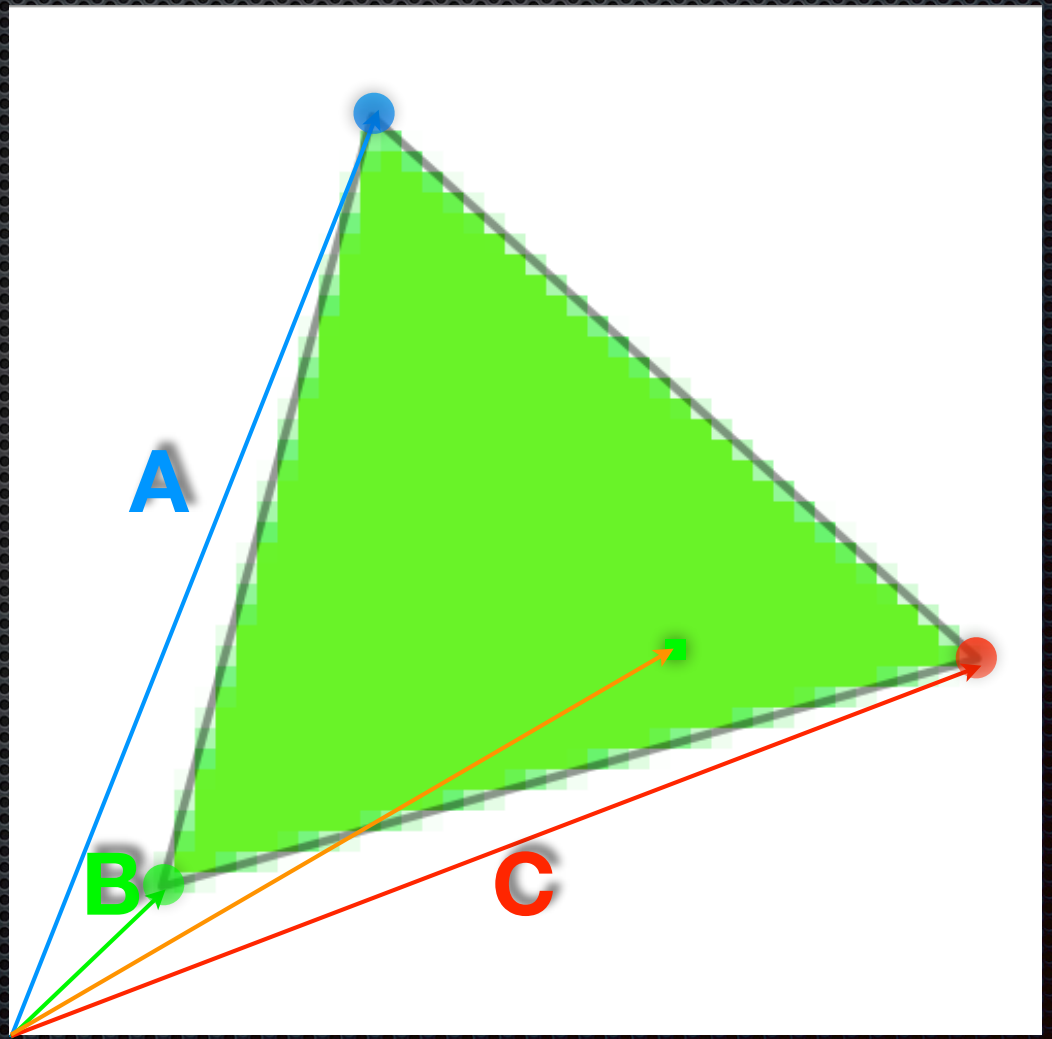
Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

$0 \leq \lambda_x \leq 1$

- λ_1 = Weight 1
- λ_2 = Weight 2
- λ_3 = Weight 3



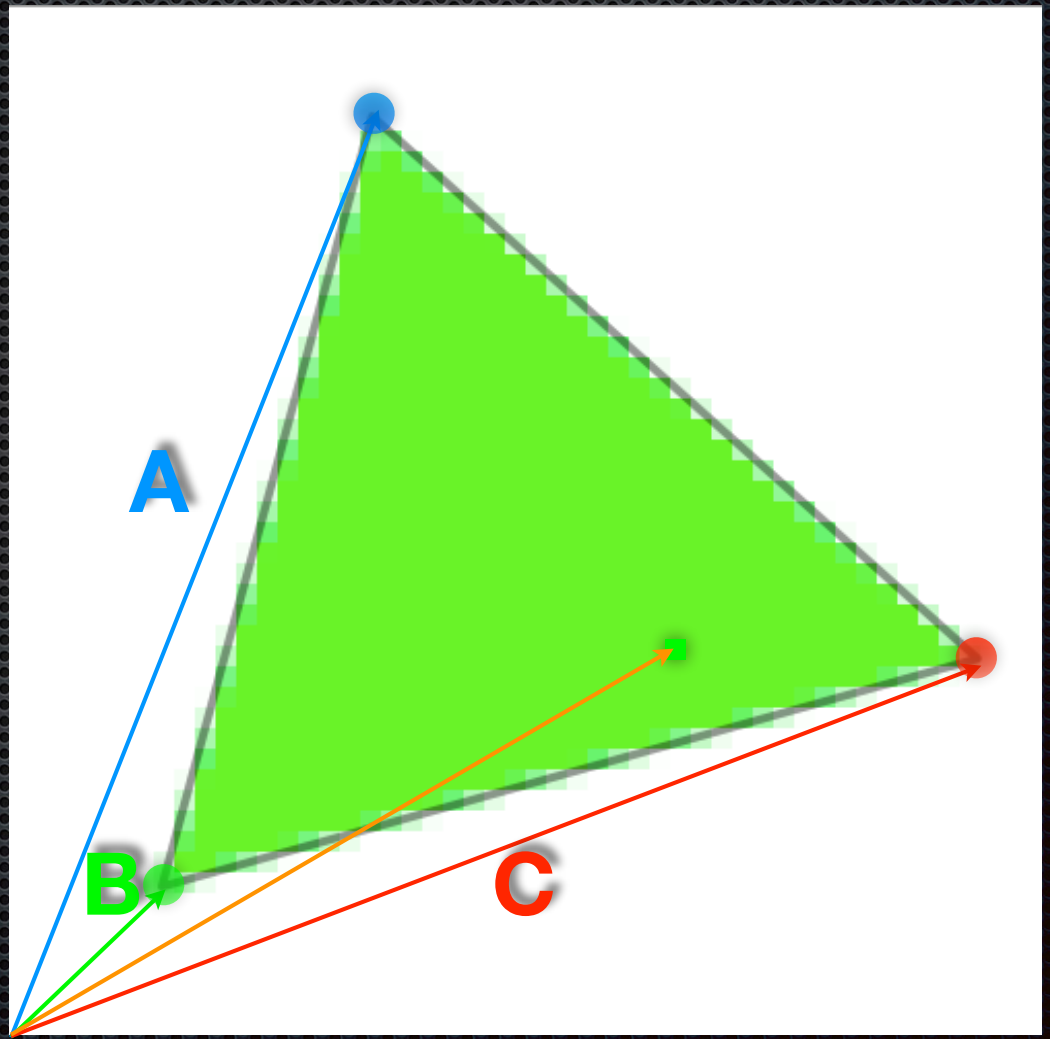
Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

$0 \leq \lambda_x \leq 1$

- λ_1 =
- λ_2 =
- λ_3 =



Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

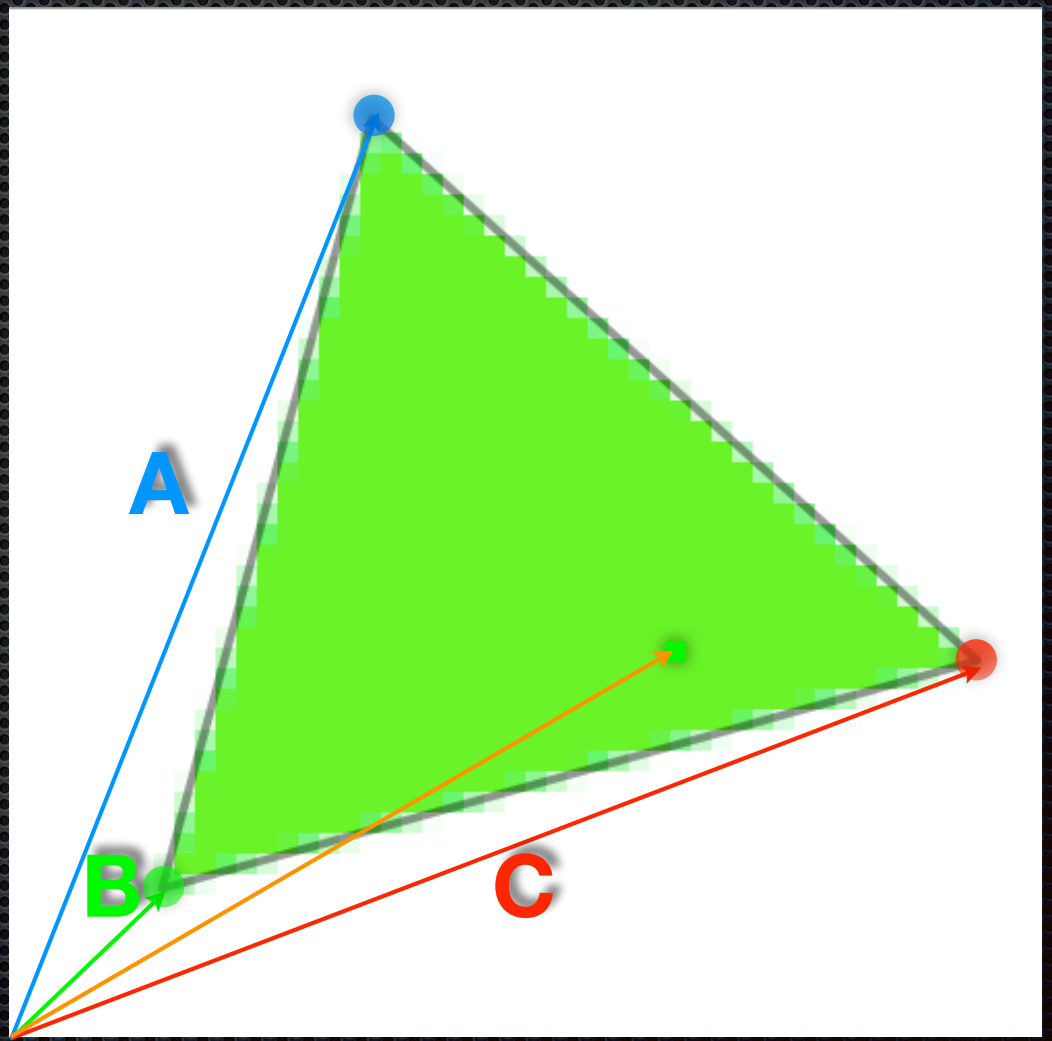
$0 \leq \lambda_x \leq 1$

$\cdot \lambda_1 +$

$\cdot \lambda_2 +$

$\cdot \lambda_3 +$

$=$



Barycentric Coordinates

$\lambda_1 + \lambda_2 + \lambda_3 = 1$

and

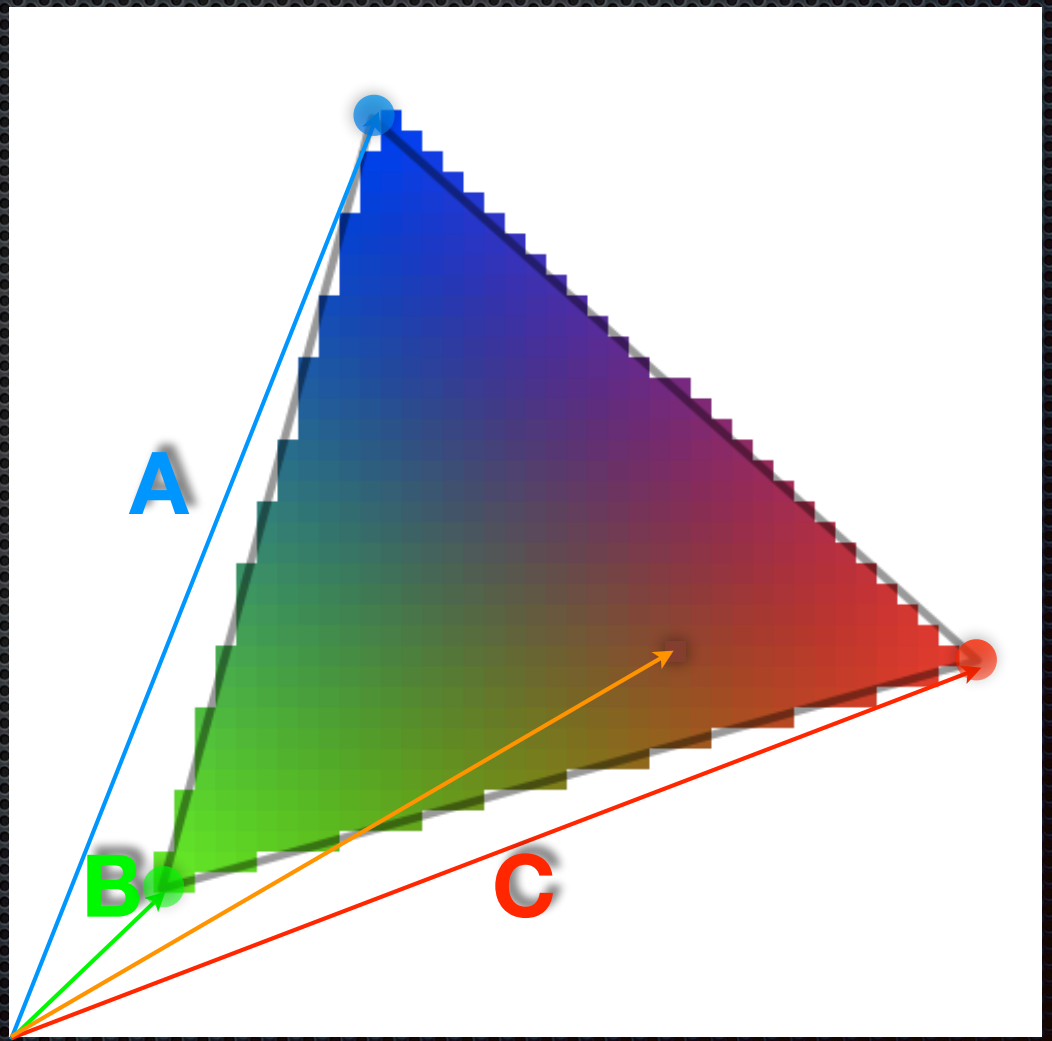
$0 \leq \lambda_x \leq 1$

$\cdot \lambda_1 +$

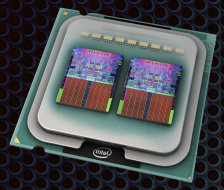
$\cdot \lambda_2 +$

$\cdot \lambda_3 +$

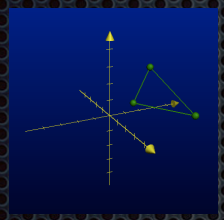
$=$



Interpolation Program



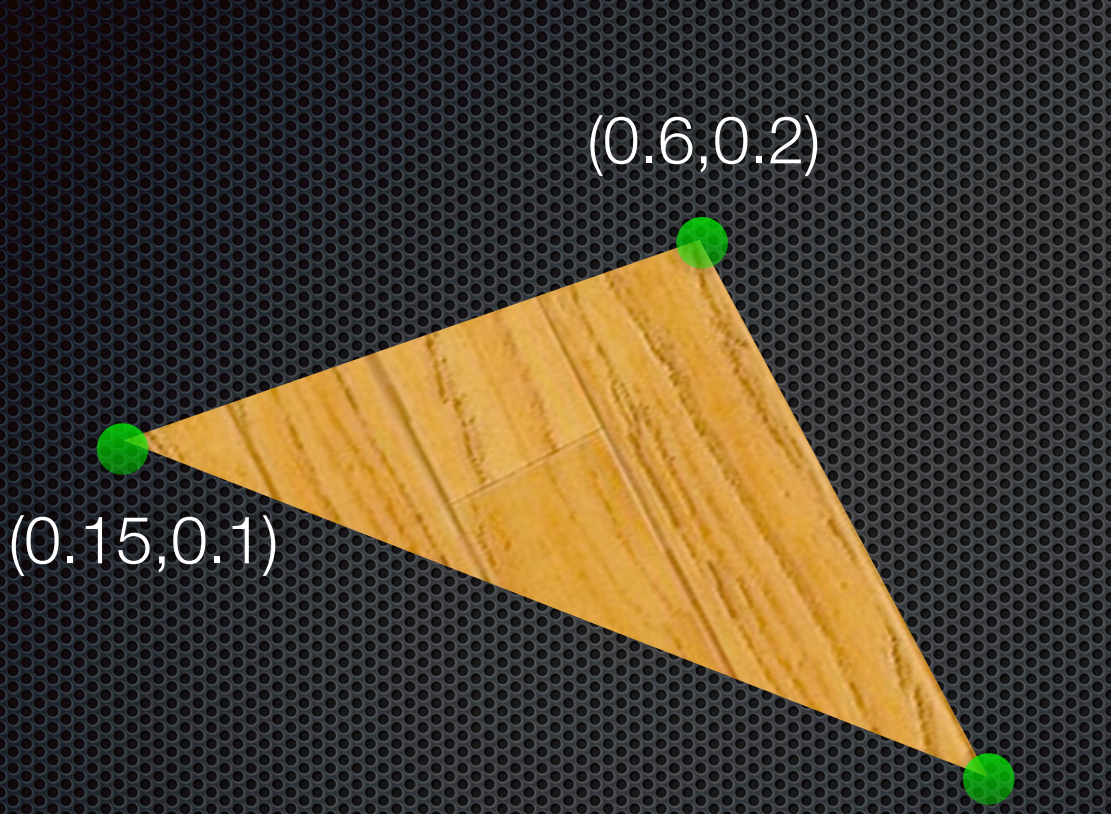
```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glVertexAttribPointer(11, 4, GLenum(GL_FLOAT), GLboolean(FALSE), 0, colors)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glDrawArrays(GLenum(GL_TRIANGLES), 0, 3) // colors array is 12 floats 0.0 -> 1.0
```

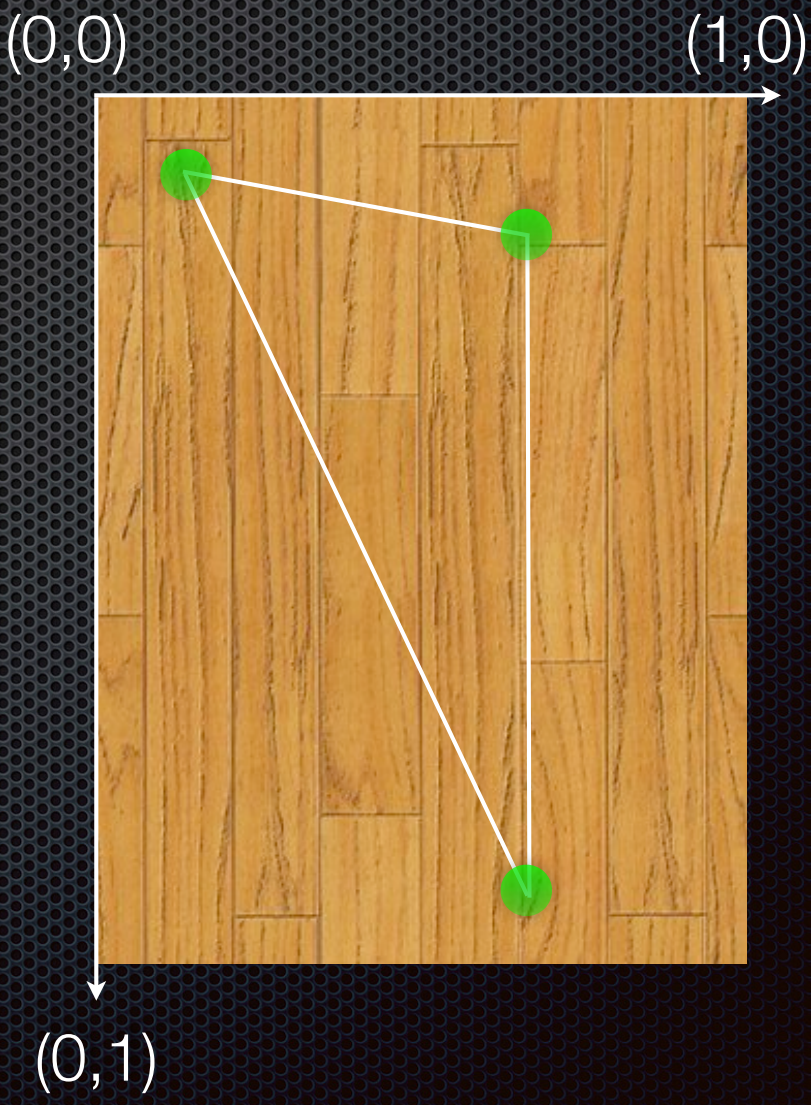


```
attribute vec2 position;
attribute vec4 color;
uniform vec2 translate;
varying vec4 colorInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    colorInterpolated = color; // color data interpolated by rasterizer
}
```

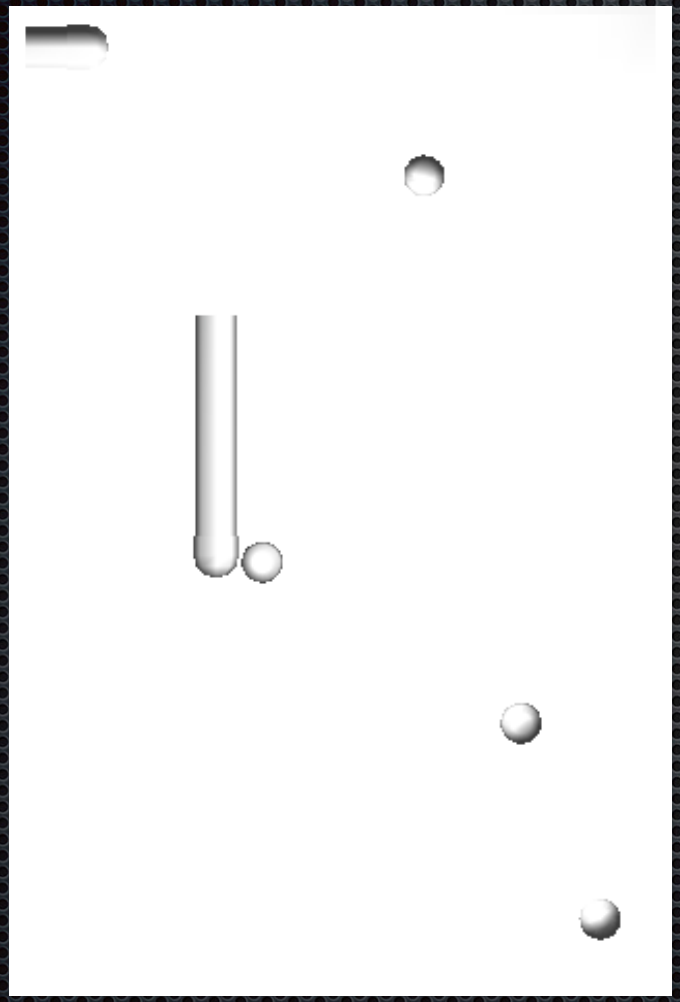
```
varying highp vec4 colorInterpolated;
void main()
{
    gl_FragColor = colorInterpolated;
}
```

Texture Coordinates





Texture Mapping



Setting Up Texturing

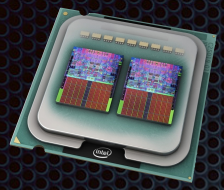
```
// Setup texturing
var woodTextureInfo: GLKTextureInfo = GLKTextureLoader.textureWithCGImage(
    UIImage(named: "texture")?.CGImage, options: nil, error: nil)
// ...
glBindTexture(GLenum(GL_TEXTURE_2D), woodTextureInfo.name)
```

MUCH easier than it used to be!

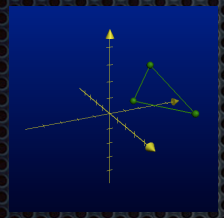
Swapping Textures

- Many textures can be loaded at one time
- Switch among loaded textures by calling glBindTexture
- Textures can be unloaded if there is insufficient memory
- Calling glBindTexture reloads the data automatically if it is not video memory resident

Texturing Program

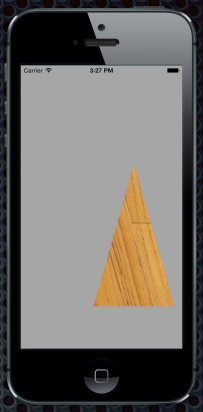


```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glVertexAttribPointer(11, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, texCoords)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glBindTexture(GLenum(GL_TEXTURE_2D), _woodTextureInfo.name)
glDrawArrays(GL_TRIANGLES, 0, 3) // texCoords array has 6 floats 0.0 -> 1.0
```



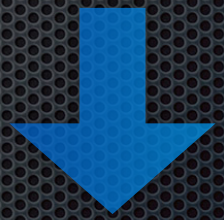
```
attribute vec2 position;
attribute vec2 textureCoordinate;
uniform vec2 translate;
varying vec2 textureCoordinateInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    textureCoordinateInterpolated = textureCoordinate; // interpolated
}
```

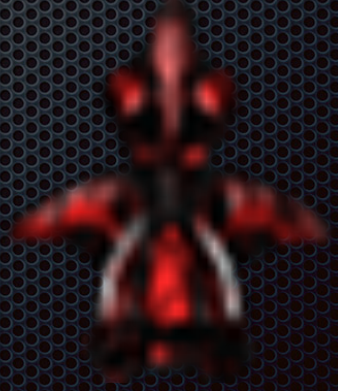
```
varying highp vec4 varyingTextureCoordinate;
uniform sampler2D textureUnit;
void main()
{ gl_FragColor = texture2D(textureUnit, varyingTextureCoordinate); }
```



Alpha Blending

Problem: My sprites are all squares!

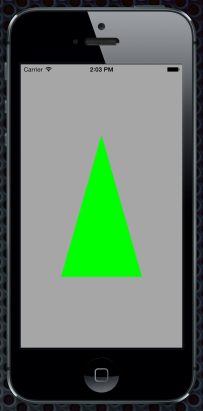


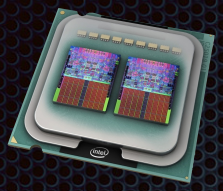


```
glEnable(GL_BLEND)
glBlendFunc(GL_SRC_ALPHA,
  GL_ONE_MINUS_SRC_ALPHA)
```

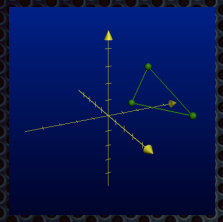
Shader Program Comparison

Pass-Through Program





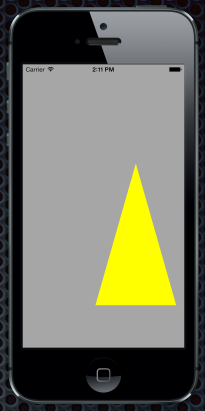
```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glDrawArrays(GLenum(GL_TRIANGLES), 0, 3)
```

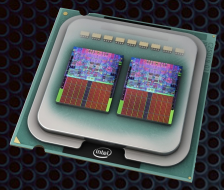


```
attribute vec2 position;
void main()
{
    gl_Position = vec4(position.x, position.y, 0.0, 1.0);
}
```

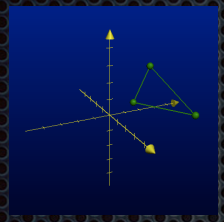
```
void main()
{
    gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
}
```

Uniform Program





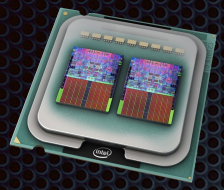
```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glUniform4f(glGetUniformLocation(_program, "color"), 1.0, 1.0, 0.0, 1.0)
glDrawArrays(GL_TRIANGLES, 0, 3)
```



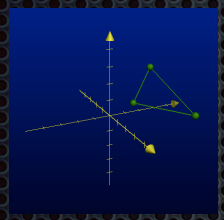
```
attribute vec2 position;
uniform vec2 translate;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
}
```

```
uniform highp vec4 color;
void main()
{
    gl_FragColor = color;
}
```

Interpolation Program



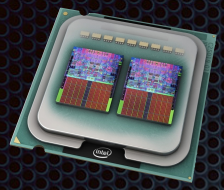
```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glVertexAttribPointer(11, 4, GLenum(GL_FLOAT), GLboolean(FALSE), 0, colors)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glDrawArrays(GLenum(GL_TRIANGLES), 0, 3) // colors array is 12 floats 0.0 -> 1.0
```



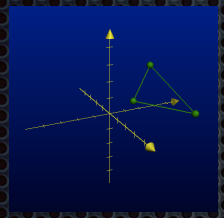
```
attribute vec2 position;
attribute vec4 color;
uniform vec2 translate;
varying vec4 colorInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    colorInterpolated = color; // color data interpolated by rasterizer
}
```

```
varying highp vec4 colorInterpolated;
void main()
{
    gl_FragColor = colorInterpolated;
}
```

Texturing Program



```
glUseProgram(_program) // positions array has 6 floats between -1.0 -> 1.0
glVertexAttribPointer(10, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, positions)
glVertexAttribPointer(11, 2, GLenum(GL_FLOAT), GLboolean(FALSE), 0, texCoords)
glUniform2f(glGetUniformLocation(_program, "translate"), 0.4, -0.2)
glBindTexture(GLenum(GL_TEXTURE_2D), _woodTextureInfo.name)
glDrawArrays(GL_TRIANGLES, 0, 3) // texCoords array has 6 floats 0.0 -> 1.0
```



```
attribute vec2 position;
attribute vec2 textureCoordinate;
uniform vec2 translate;
varying vec2 textureCoordinateInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    textureCoordinateInterpolated = textureCoordinate; // interpolated
}
```

```
varying highp vec4 varyingTextureCoordinate;
uniform sampler2D textureUnit;
void main()
{ gl_FragColor = texture2D(textureUnit, varyingTextureCoordinate); }
```