

Mobile Application Programming

OpenGL ES Shaders

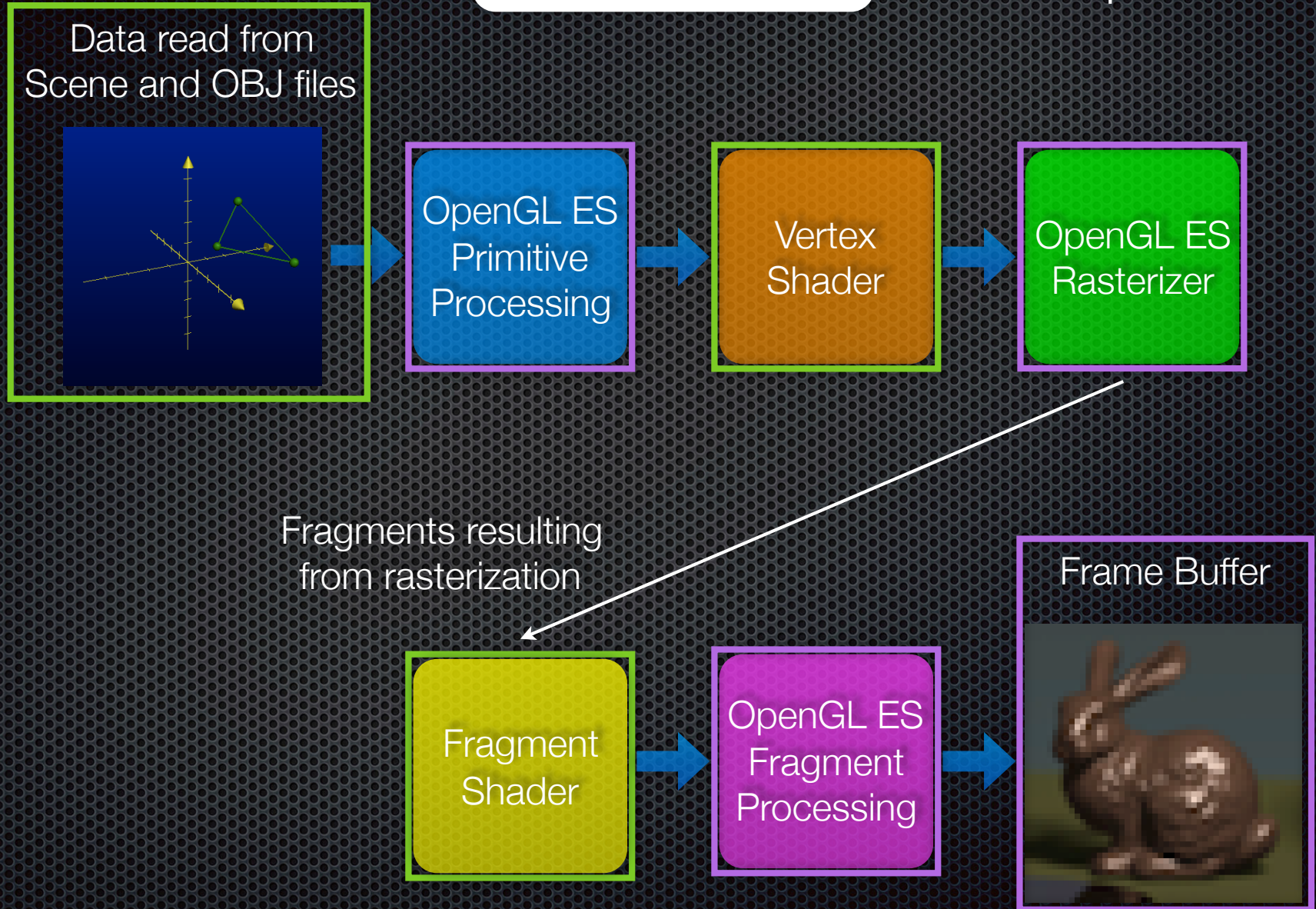
OpenGL Environment

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

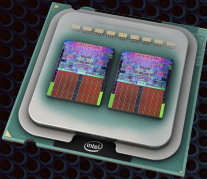
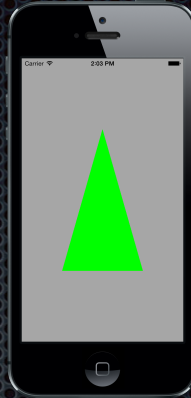




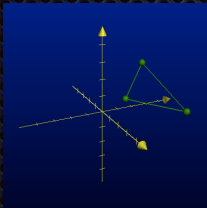
— Your Code
— OpenGL Library



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)
```

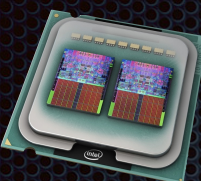
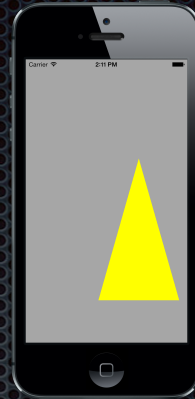


```
attribute vec2 position;
void main()
{
    gl_Position = vec4(position.x, position.y, 0.0, 1.0);
}
```

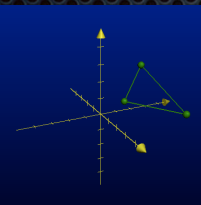


```
void main()
{
    gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
}
```

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)
```

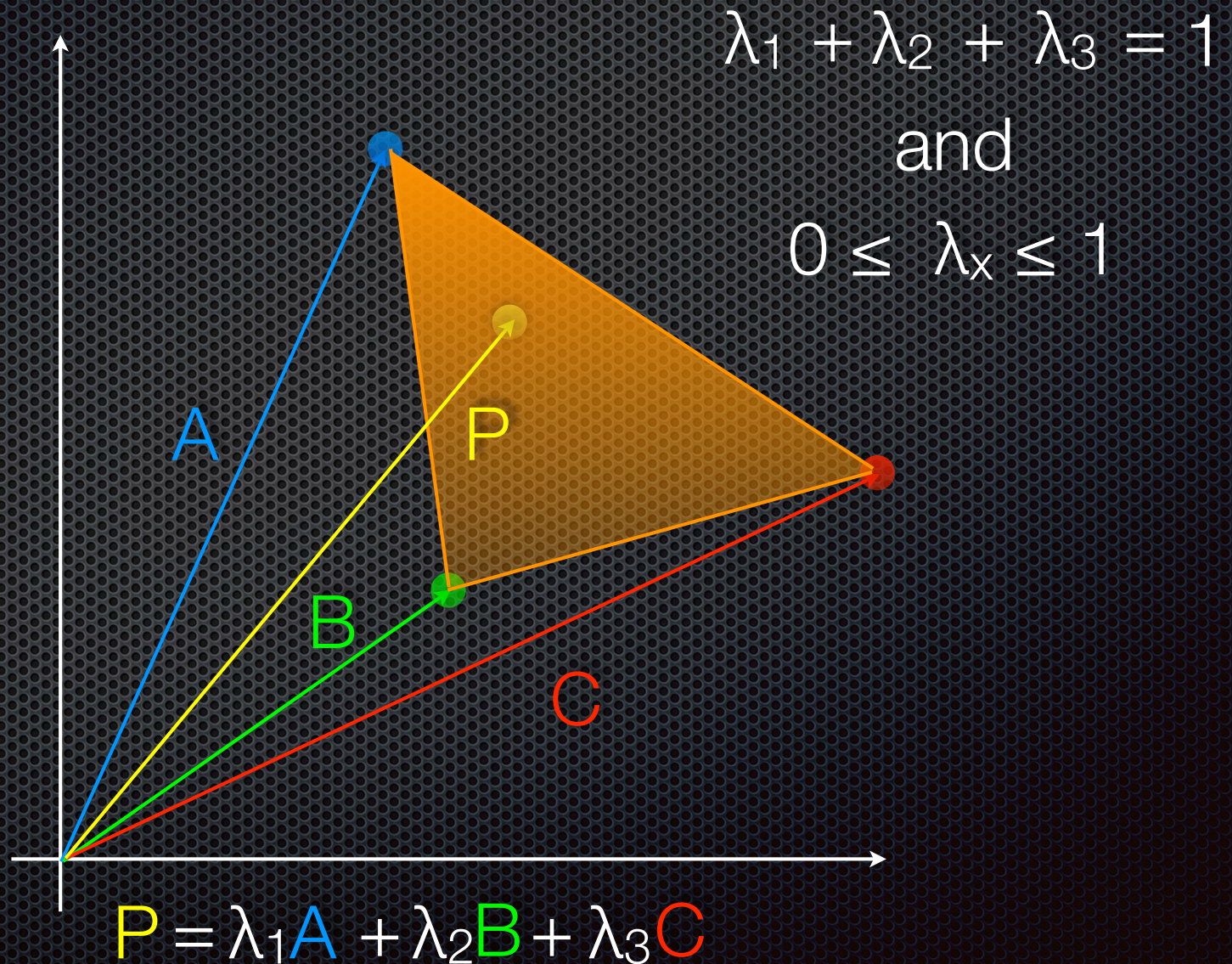


```
attribute vec2 position;
uniform vec2 translate;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
}
```

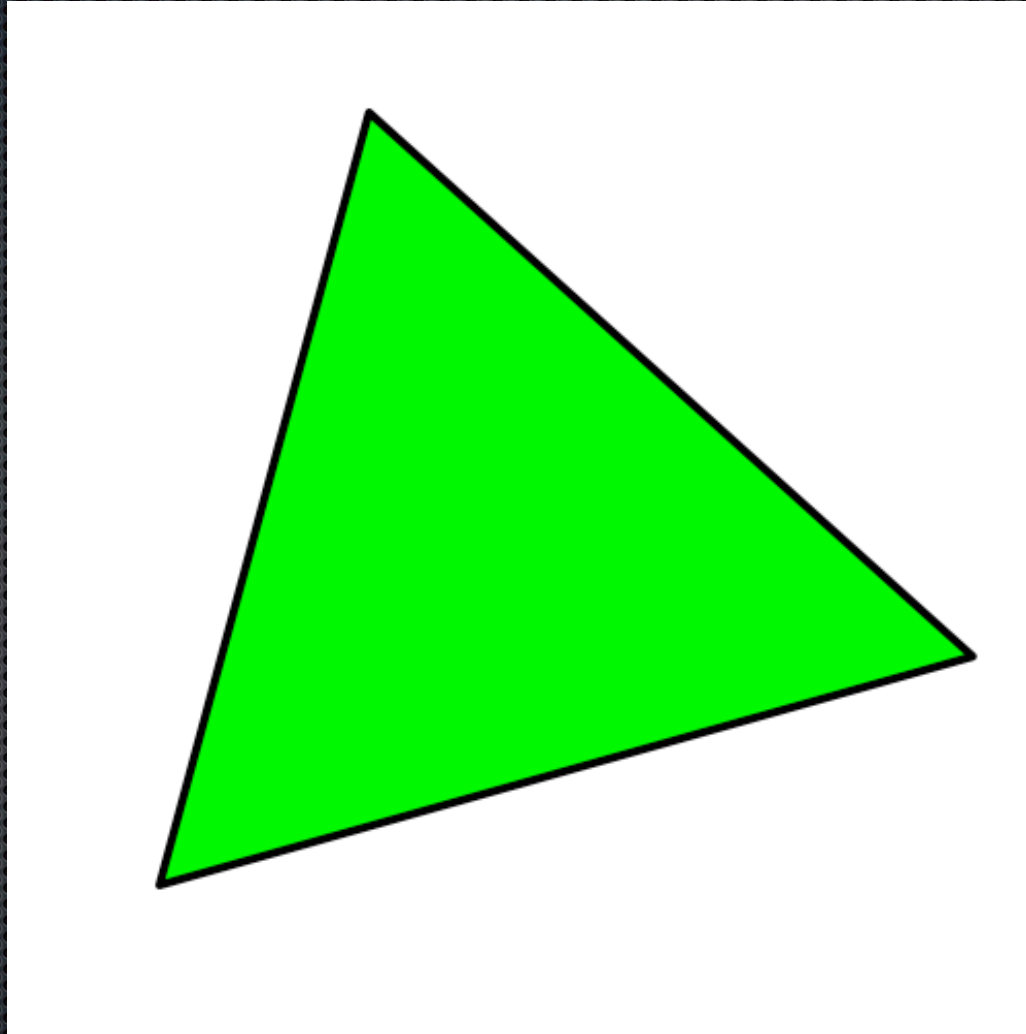


```
uniform highp vec4 color;
void main()
{
    gl_FragColor = color;
}
```

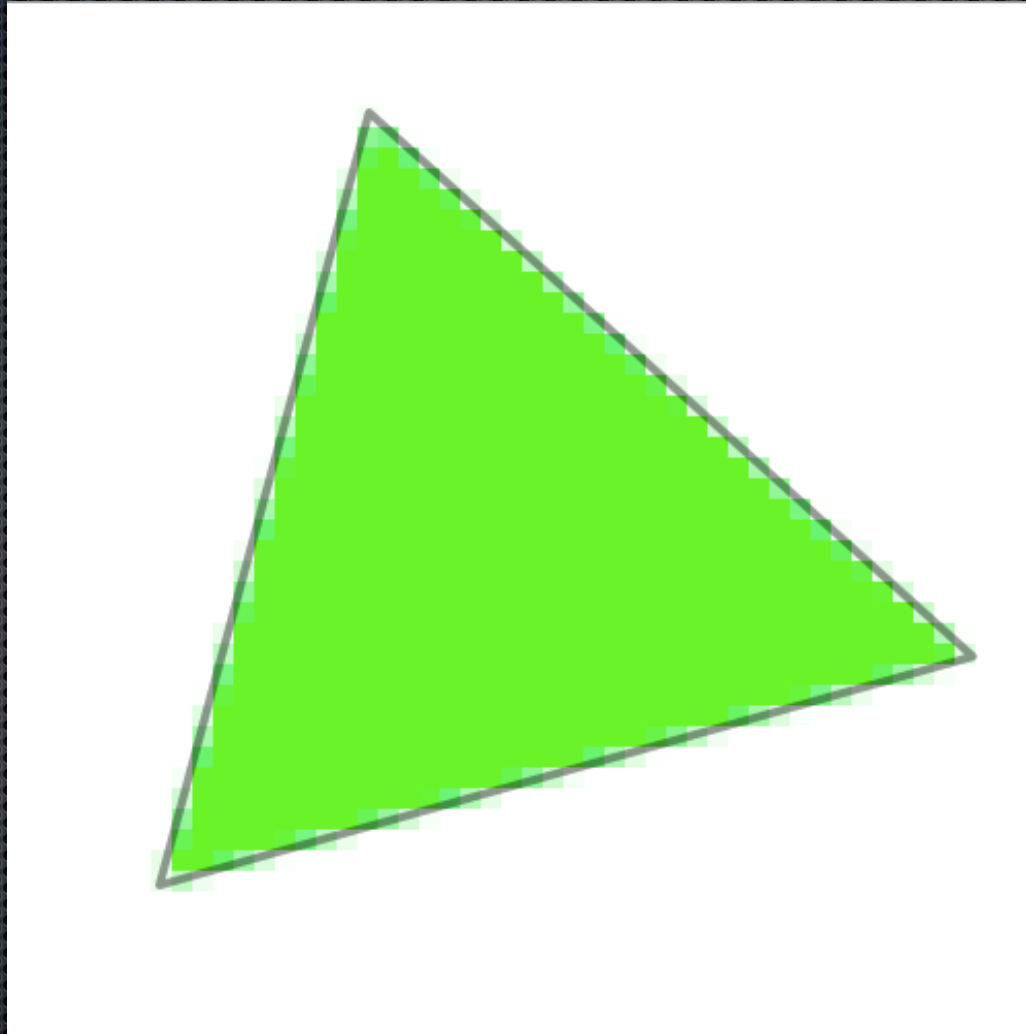
Barycentric Coordinates



Rasterization



Rasterization



Barycentric Coordinates

```
2308, -0.058077, 0.541154, 0.516923, -0.063462, 0.541923, 0.521538, -0.068846, 0.542692, 0.526
5000, 0.545000, 0.540000, -0.090385, 0.545769, 0.544615, -0.095769, 0.546538, 0.549231, -0.101
8846, 0.563077, -0.117308, 0.549615, 0.567692, -0.122692, 0.550385, 0.572308, -0.128077, 0.551
5154, -0.144231, 0.553462, 0.590769, -0.149615, 0.554231, 0.595385, -0.155000, 0.555000, 0.600
1154, 0.557308, 0.613846, -0.176538, 0.558077, 0.618462, -0.181923, 0.558846, 0.623077, -0.187
1154, 0.636923, -0.203462, 0.561923, 0.641538, -0.208846, 0.562692, 0.646154, -0.214231, 0.563
0000, -0.230385, 0.565769, 0.664615, -0.235769, 0.566538, 0.669231, -0.241154, 0.567308, 0.673
7308, 0.569615, 0.687692, -0.262692, 0.570385, 0.692308, -0.268077, 0.571154, 0.696923, -0.273
8462, 0.710769, -0.289615, 0.574231, 0.715385, -0.295000, 0.575000, 0.720000, -0.300385, 0.575
8846, -0.316538, 0.578077, 0.738462, -0.321923, 0.578846, 0.743077, -0.327308, 0.579615, 0.747
8462, 0.581923, 0.761538, -0.348846, 0.582692, 0.766154, -0.354231, 0.583462, 0.770769, -0.359
5769, 0.784615, -0.375769, 0.586538, 0.789231, -0.381154, 0.587308, 0.793846, -0.386538, 0.588
7692, -0.402692, 0.590385, 0.812308, -0.408077, 0.591154, 0.816923, -0.413462, 0.591923, 0.821
9615, 0.594231, 0.835385, -0.435000, 0.595000, 0.840000, -0.440385, 0.595769, 0.844615, -0.445
8077, 0.858461, -0.461923, 0.598846, 0.863077, -0.467308, 0.599615, 0.867692, -0.472692, 0.600
1538, -0.488846, 0.602692, 0.886154, -0.494231, 0.603462, 0.890769, -0.499615, 0.604231, 0.895
5769, 0.606538, 0.909231, -0.521154, 0.607308, 0.913846, -0.526538, 0.608077, 0.918462, -0.531
0385, 0.932308, -0.548077, 0.611154, 0.936923, -0.553462, 0.611923, 0.941538, -0.558846, 0.612
5385, -0.575000, 0.615000, 0.960000, -0.580385, 0.615769, 0.964615, -0.585769, 0.616538, 0.969
1923, 0.618846, 0.983077, -0.607308, 0.619615, 0.987692, -0.612692, 0.620385, 0.992308, -0.618
2692, 1.006154, -0.634231, 0.623462, 1.010769, -0.639615, 0.624231, 1.015385, -0.645000, 0.625
9231, -0.661154, 0.627308, 1.033846, -0.666538, 0.628077, 1.038462, -0.671923, 0.628846, 1.043
8077, 0.631154, 1.056923, -0.693462, 0.631923, 1.061538, -0.698846, 0.632692, 1.066154, -0.704
5000, 1.080000, -0.720385, 0.635769, 1.084615, -0.725769, 0.636538, 1.089231, -0.731154, 0.637
8077, -0.747308, 0.639615, 1.107692, -0.752692, 0.640385, 1.112308, -0.758077, 0.641154, 1.116
4231, 0.643462, 1.130769, -0.779615, 0.644231, 1.135385, -0.785000, 0.645000, 1.140000, -0.790
7308, 1.153846, -0.806538, 0.648077, 1.158462, -0.811923, 0.648846, 1.163077, -0.817308, 0.649
5923, -0.833462, 0.651923, 1.181538, -0.838846, 0.652692, 1.186154, -0.844231, 0.653462, 1.190
0385, 0.655769, 1.204615, -0.865769, 0.656538, 1.209231, -0.871154, 0.657308, 1.213846, -0.876
9615, 1.227692, -0.892692, 0.660385, 1.232308, -0.898077, 0.661154, 1.236923, -0.903462, 0.661
9769, -0.919615, 0.664231, 1.255385, -0.925000, 0.665000, 1.260000, -0.930385, 0.665769, 1.264
538, 0.668077, 1.278461, -0.951923, 0.668846, 1.283077, -0.957308, 0.669615, 1.287692, -0.962
1923, 1.301538, -0.978846, 0.672692, 1.306154, -0.984231, 0.673462, 1.310769, -0.989615, 0.674
4615, -1.005769, 0.676538, 1.329231, -1.011154, 0.677308, 1.333846, -1.016539, 0.678077, 1.338
2692, 0.680385, 1.352308, -1.038077, 0.681154, 1.356923, -1.043462, 0.681923, 1.361538, -1.048
```

Barycentric Coordinates

```
2308, -0.058077, 0.541154, 0.516923, -0.063462, 0.541923, 0.521538, -0.068846, 0.542692, 0.526
5000, 0.545000, 0.540000, -0.090385, 0.545769, 0.544615, -0.095769, 0.546538, 0.549231, -0.101
8846, 0.563077, -0.117308, 0.549615, 0.567692, -0.122692, 0.550385, 0.572308, -0.128077, 0.551
5154, -0.144231, 0.553462, 0.590769, -0.149615, 0.554231, 0.595385, -0.155000, 0.555000, 0.600
1154, 0.557308, 0.613846, -0.176538, 0.558077, 0.618462, -0.181923, 0.558846, 0.623077, -0.187
1154, 0.636923, -0.203462, 0.561923, 0.641538, -0.208846, 0.562692, 0.646154, -0.214231, 0.563
0000, -0.230385, 0.565769, 0.664615, -0.235769, 0.566538, 0.669231, -0.241154, 0.567308, 0.673
7308, 0.569615, 0.687692, -0.262692, 0.570385, 0.692308, -0.268077, 0.571154, 0.696923, -0.273
8462, 0.710769, -0.289615, 0.574231, 0.715385, -0.295000, 0.575000, 0.720000, -0.300385, 0.575
8846, -0.316538, 0.578077, 0.738462, -0.321923, 0.578846, 0.743077, -0.327308, 0.579615, 0.747
8462, 0.581923, 0.761538, -0.348846, 0.582692, 0.766154, -0.354231, 0.583462, 0.770769, -0.359
5769, 0.784615, -0.375769, 0.586538, 0.789231, -0.381154, 0.587308, 0.793846, -0.386538, 0.588
7692, -0.402692, 0.590385, 0.812308, -0.408077, 0.591154, 0.816923, -0.413462, 0.591923, 0.821
9615, 0.594231, 0.835385, -0.435000, 0.595000, 0.840000, -0.440385, 0.595769, 0.844615, -0.445
8077, 0.858461, -0.461923, 0.598846, 0.863077, -0.467308, 0.599615, 0.867692, -0.472692, 0.600
1538, -0.488846, 0.602692, 0.886154, -0.494231, 0.603462, 0.890769, -0.499615, 0.604231, 0.895
5769, 0.606538, 0.909231, -0.521154, 0.607308, 0.913846, -0.526538, 0.608077, 0.918462, -0.531
0385, 0.932308, -0.548077, 0.611154, 0.936923, -0.553462, 0.611923, 0.941538, -0.558846, 0.612
5385, -0.575000, 0.615000, 0.960000, -0.580385, 0.615769, 0.964615, -0.585769, 0.616538, 0.969
1923, 0.618846, 0.983077, -0.607308, 0.619615, 0.987692, -0.612692, 0.620385, 0.992308, -0.618
2692, 1.006154, -0.634231, 0.623462, 1.010769, -0.639615, 0.624231, 1.015385, -0.645000, 0.625
9231, -0.661154, 0.627308, 1.033846, -0.666538, 0.628077, 1.038462, -0.671923, 0.628846, 1.043
8077, 0.631154, 1.056923, -0.693462, 0.631923, 1.061538, -0.698846, 0.632692, 1.066154, -0.704
5000, 1.080000, -0.720385, 0.635769, 1.084615, -0.725769, 0.636538, 1.089231, -0.731154, 0.637
8077, -0.747308, 0.639615, 1.107692, -0.752692, 0.640385, 1.112308, -0.758077, 0.641154, 1.116
4231, 0.643462, 1.130769, -0.779615, 0.644231, 1.135385, -0.785000, 0.645000, 1.140000, -0.790
7308, 1.153846, -0.806538, 0.648077, 1.158462, -0.811923, 0.648846, 1.163077, -0.817308, 0.649
5923, -0.833462, 0.651923, 1.181538, -0.838846, 0.652692, 1.186154, -0.844231, 0.653462, 1.190
0385, 0.655769, 1.204615, -0.865769, 0.656538, 1.209231, -0.871154, 0.657308, 1.213846, -0.876
9615, 1.227692, -0.892692, 0.660385, 1.232308, -0.898077, 0.661154, 1.236923, -0.903462, 0.661
9769, -0.919615, 0.664231, 1.255385, -0.925000, 0.665000, 1.260000, -0.930385, 0.665769, 1.264
538, 0.668077, 1.278461, -0.951923, 0.668846, 1.283077, -0.957308, 0.669615, 1.287692, -0.962
1923, 1.301538, -0.978846, 0.672692, 1.306154, -0.984231, 0.673462, 1.310769, -0.989615, 0.674
4615, -1.005769, 0.676538, 1.329231, -1.011154, 0.677308, 1.333846, -1.016539, 0.678077, 1.338
2692, 0.680385, 1.352308, -1.038077, 0.681154, 1.356923, -1.043462, 0.681923, 1.361538, -1.048
```

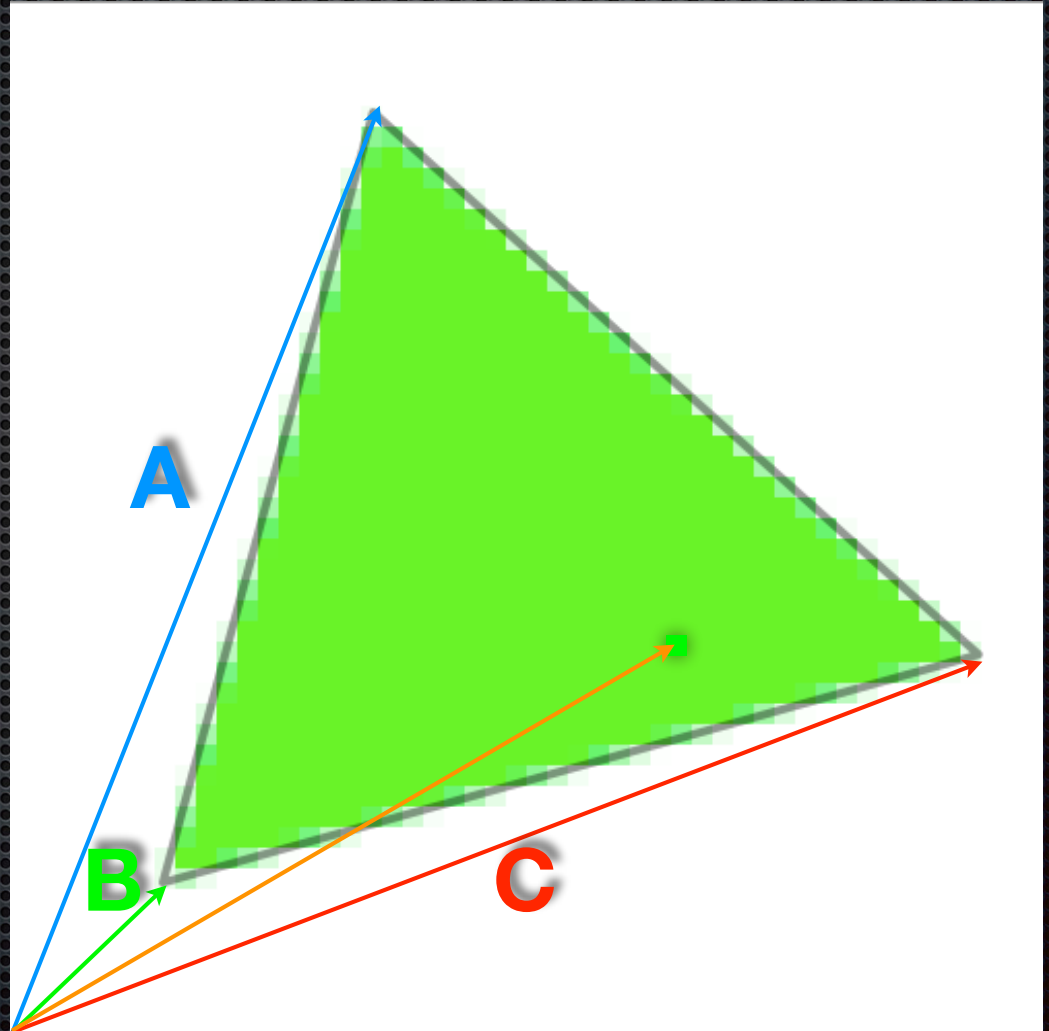
What else can we do with this data?

Barycentric Coordinates

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

and

$$0 \leq \lambda_x \leq 1$$



Barycentric Coordinates

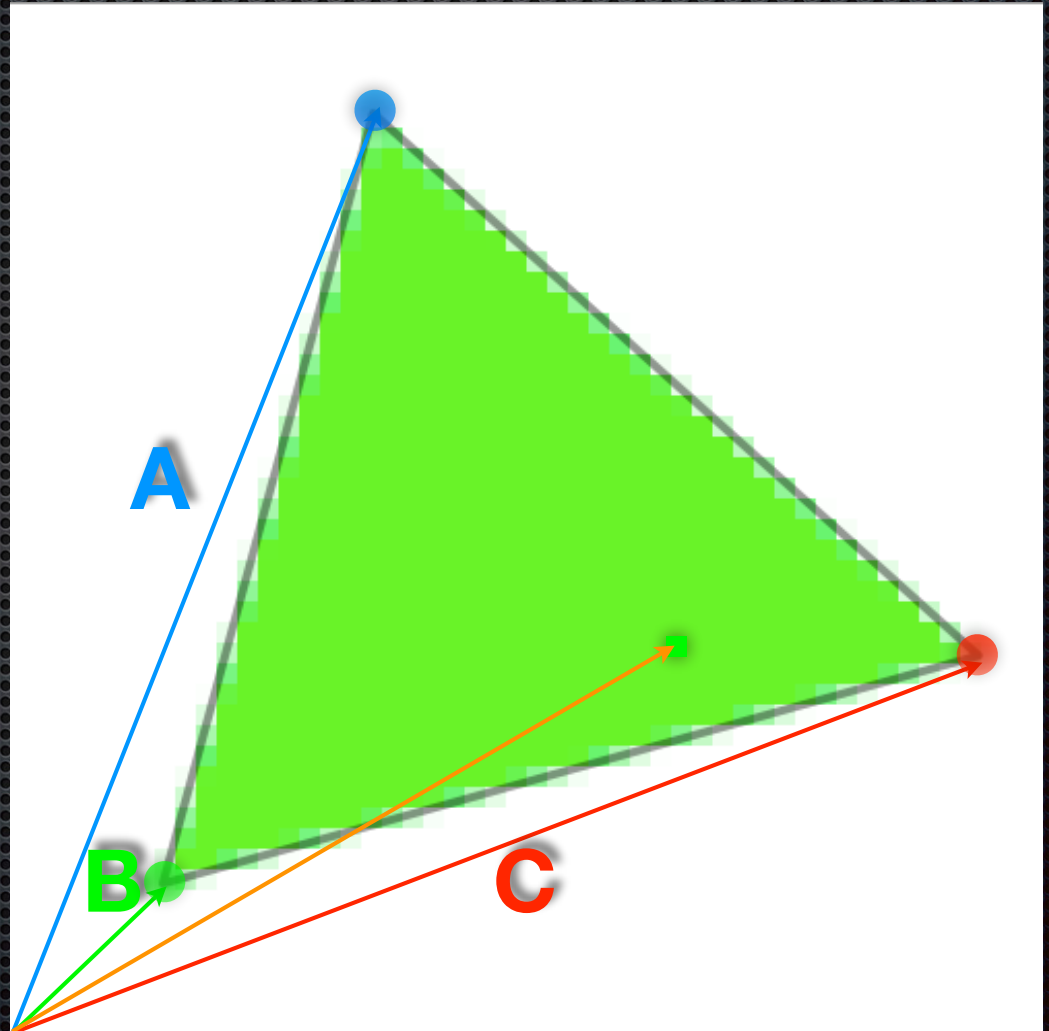
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

and

$$0 \leq \lambda_x \leq 1$$



Associate a color
with each point



Barycentric Coordinates

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

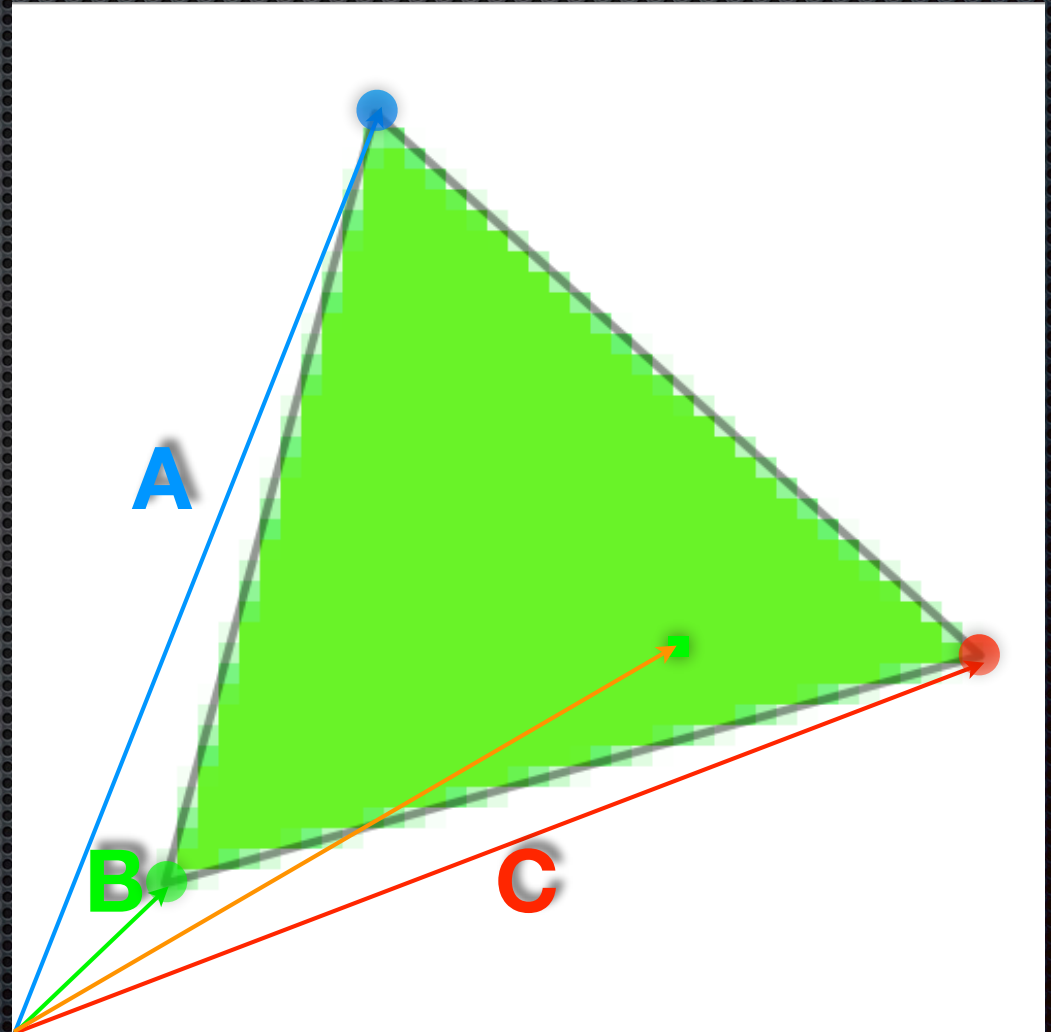
and

$$0 \leq \lambda_x \leq 1$$

● $0 \leq \lambda_1 \leq 1$

● $0 \leq \lambda_2 \leq 1$

● $0 \leq \lambda_3 \leq 1$



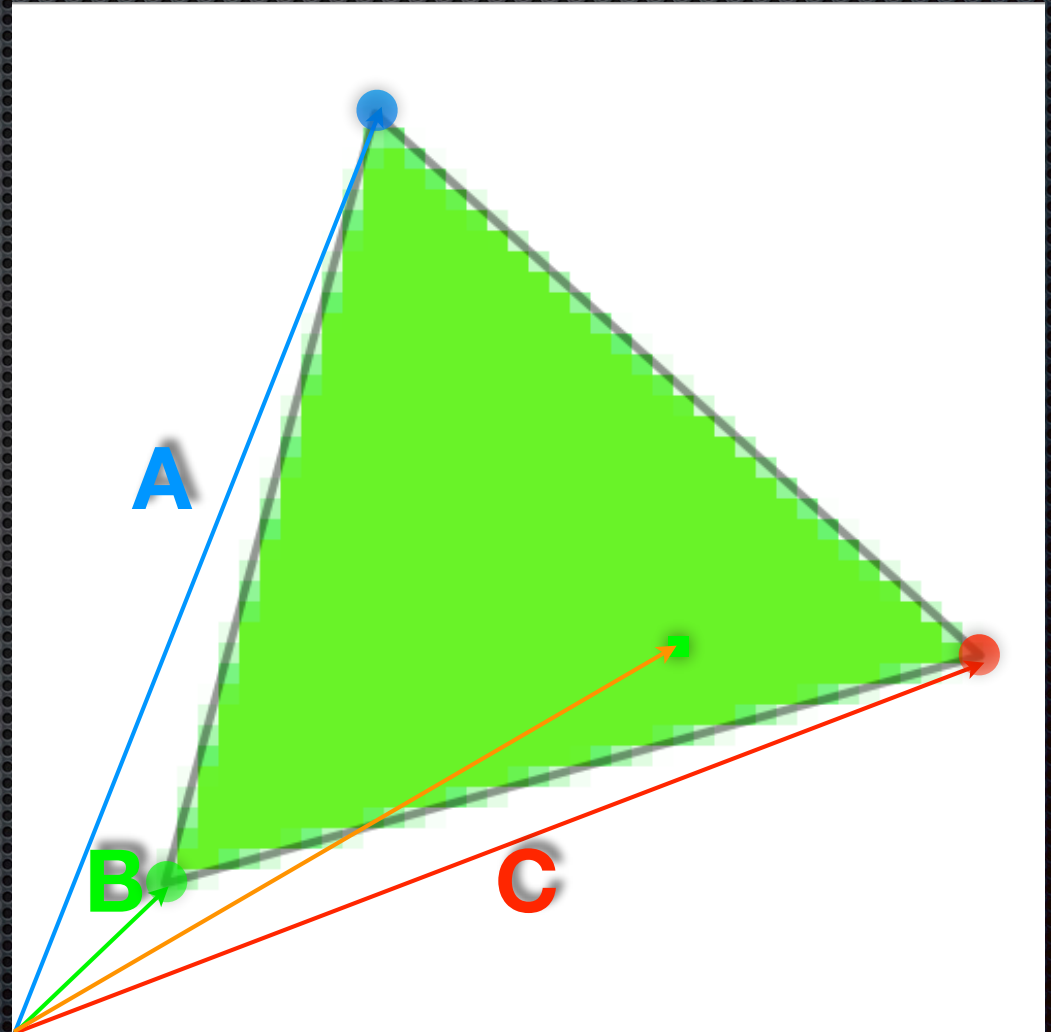
Barycentric Coordinates

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

and

$$0 \leq \lambda_x \leq 1$$

- $\lambda_1 = \text{Weight 1}$
- $\lambda_2 = \text{Weight 2}$
- $\lambda_3 = \text{Weight 3}$

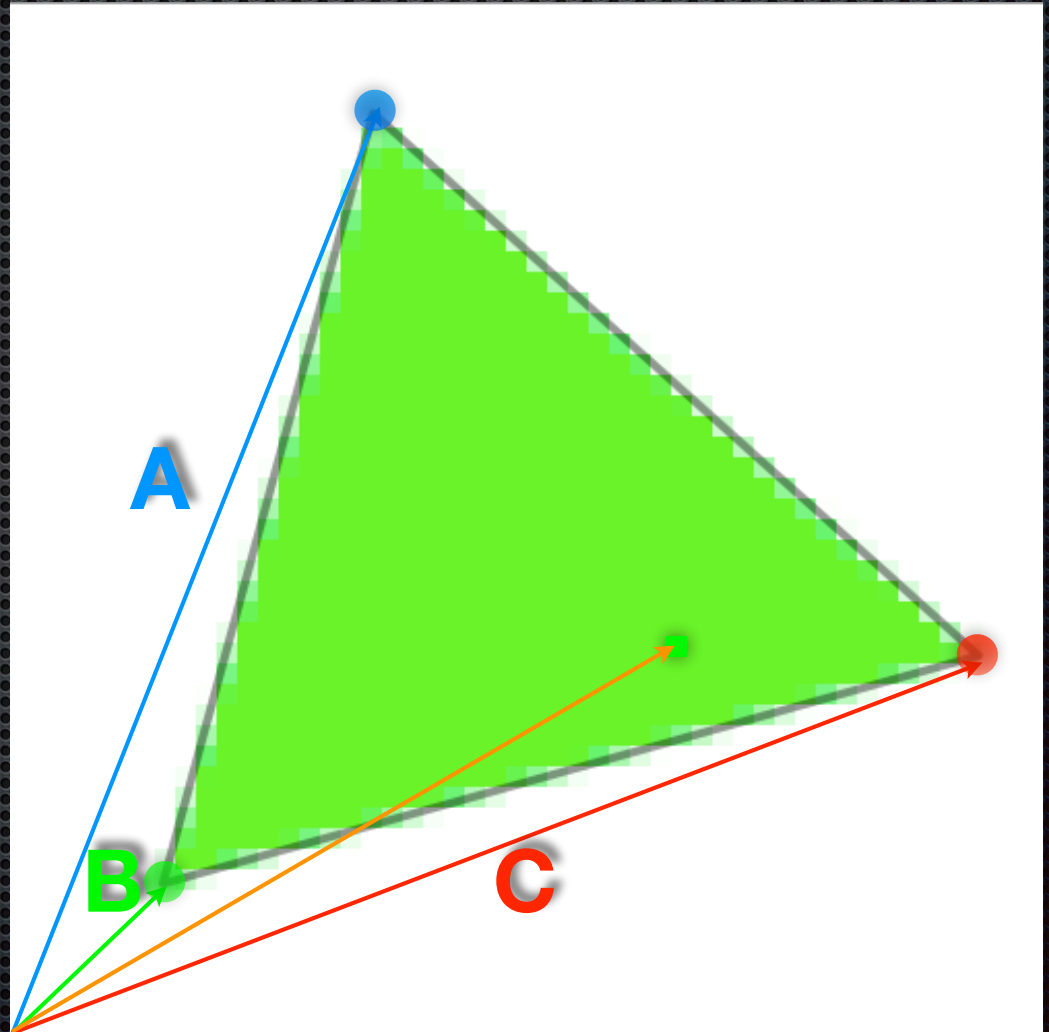
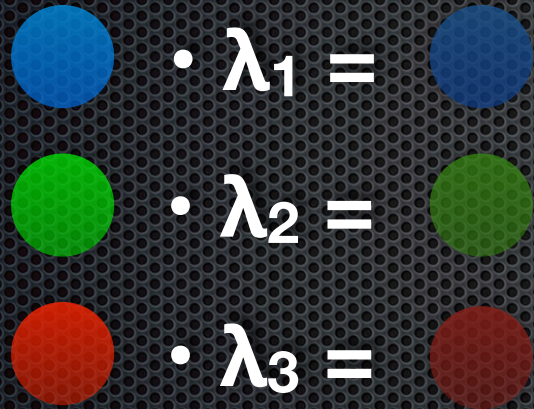


Barycentric Coordinates

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

and

$$0 \leq \lambda_x \leq 1$$



Barycentric Coordinates

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

and

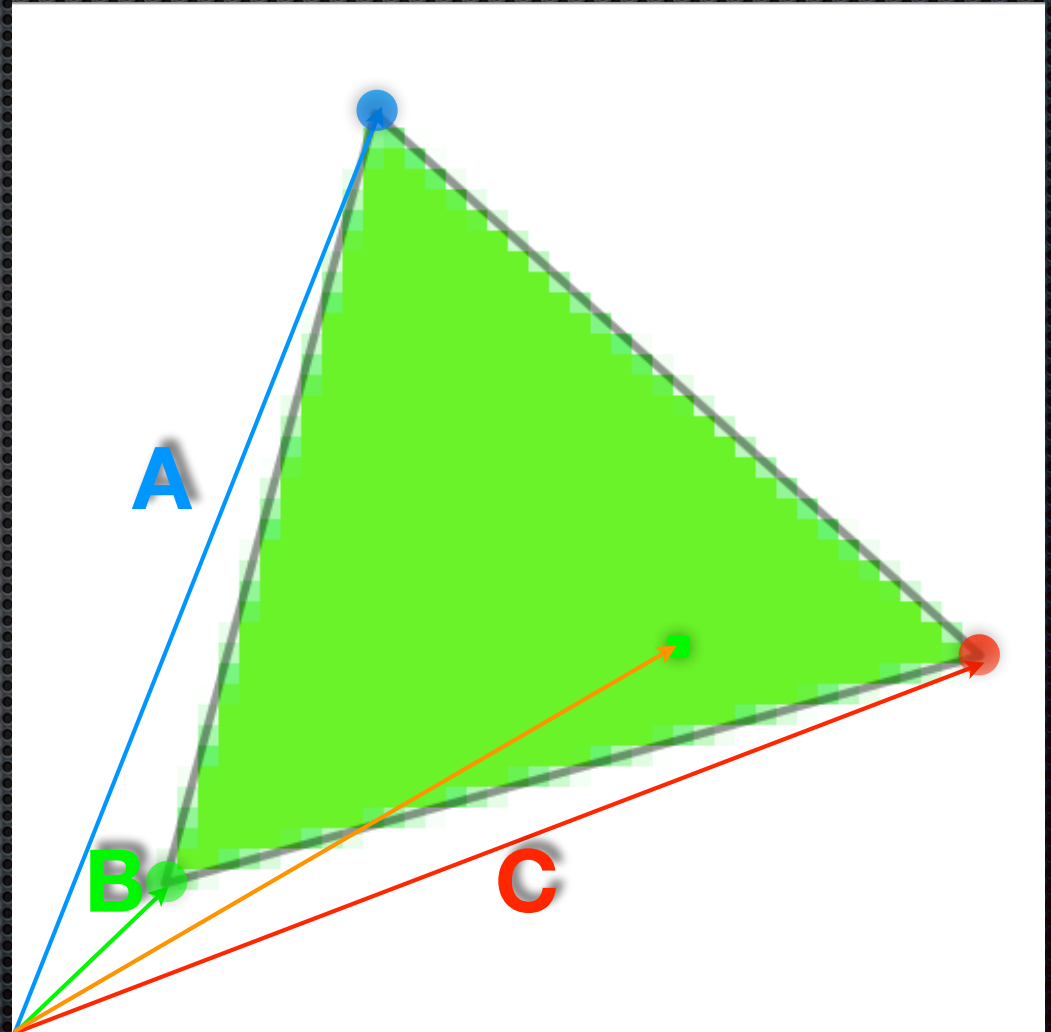
$$0 \leq \lambda_x \leq 1$$

• $\lambda_1 +$

• $\lambda_2 +$

• $\lambda_3 +$

=



Barycentric Coordinates

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

and

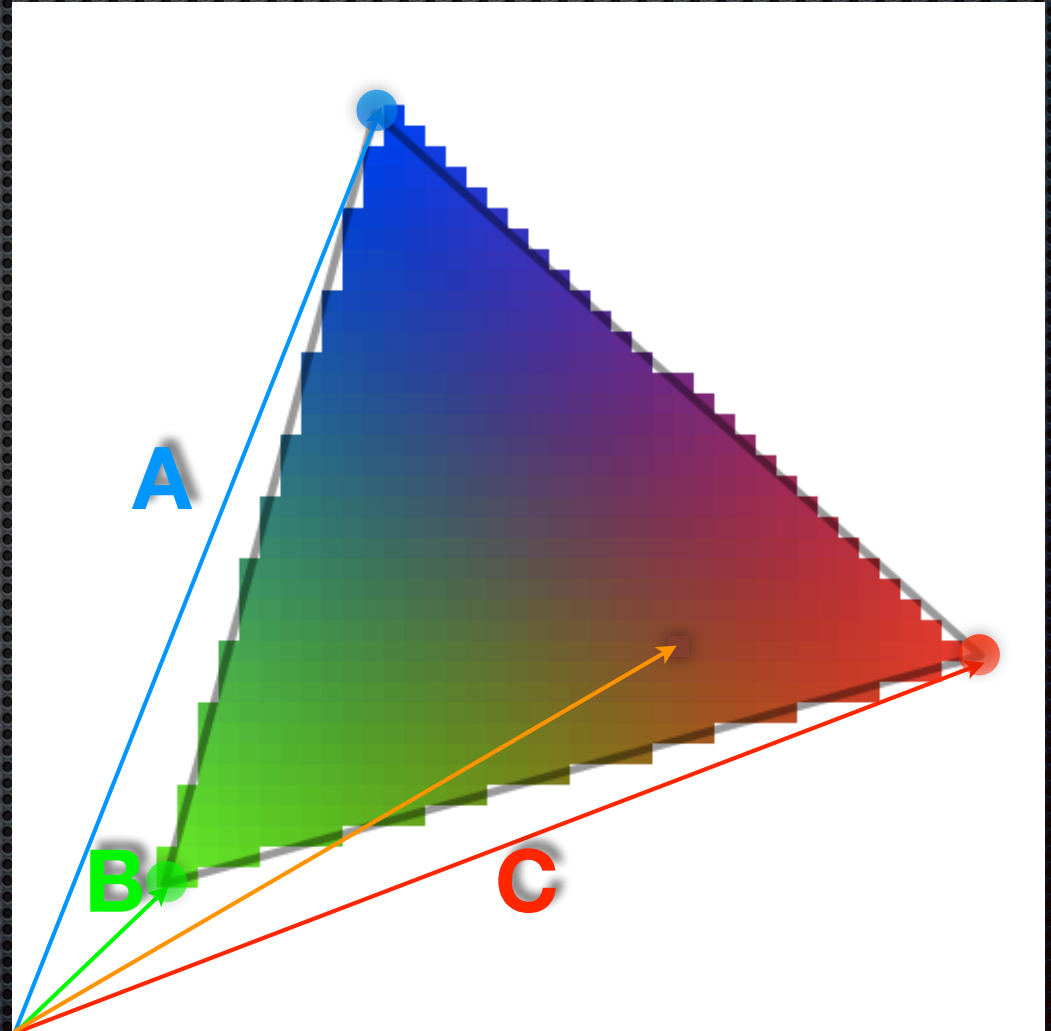
$$0 \leq \lambda_x \leq 1$$

• $\lambda_1 +$

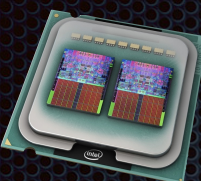
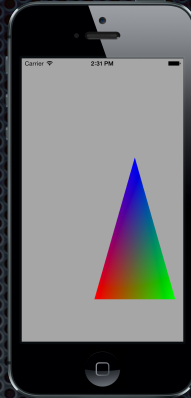
• $\lambda_2 +$

• $\lambda_3 +$

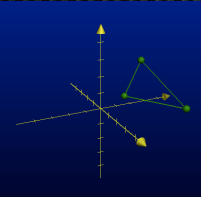
=



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
```

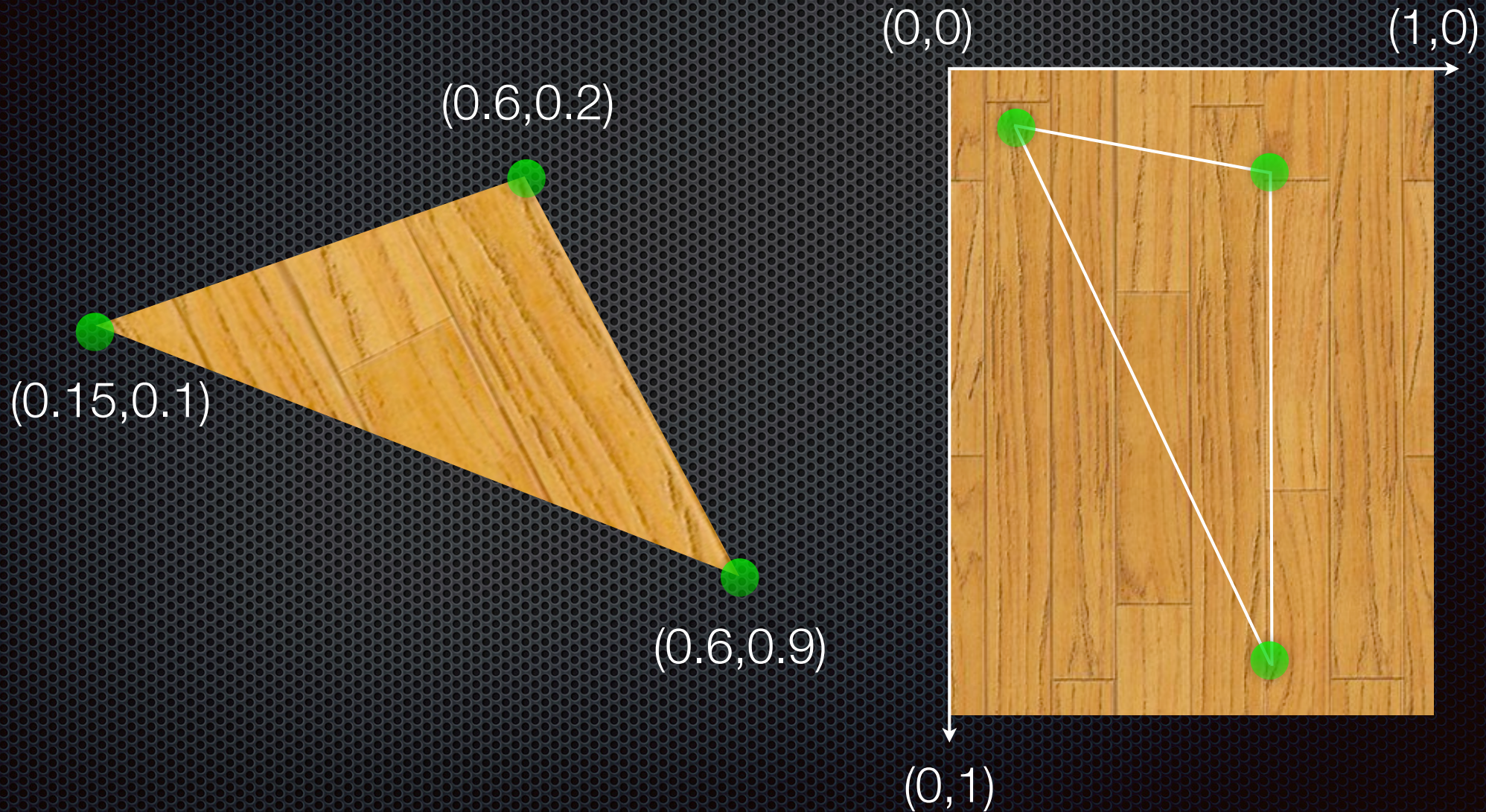


```
attribute vec2 position;
attribute vec4 color;
uniform vec2 translate;
varying vec4 colorInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    colorInterpolated = color; // color data interpolated by rasterizer
}
```

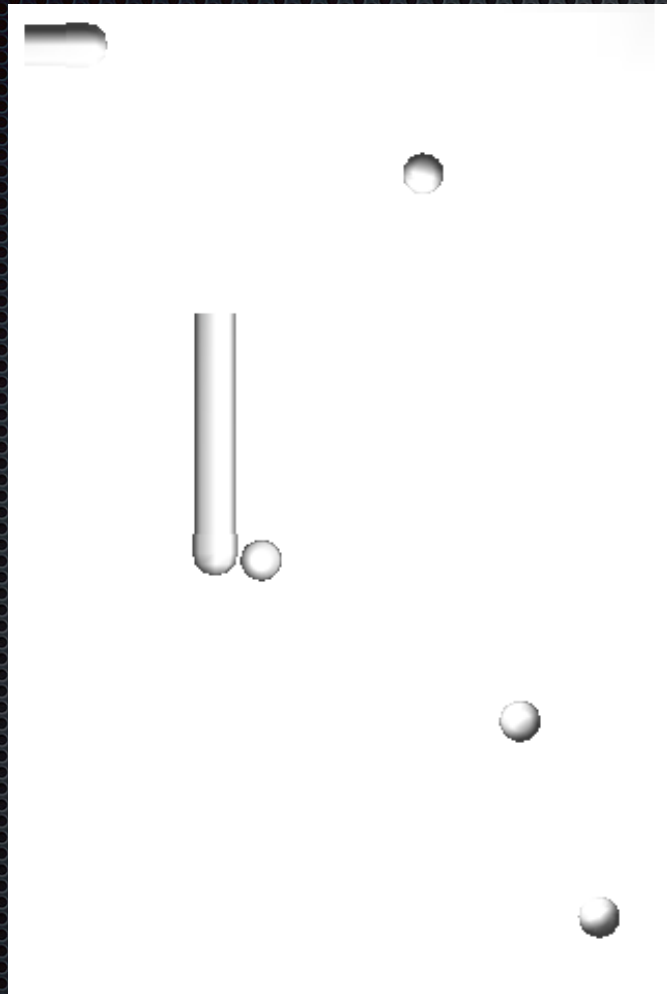


```
varying highp vec4 colorInterpolated;
void main()
{
    gl_FragColor = colorInterpolated;
}
```

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)
```

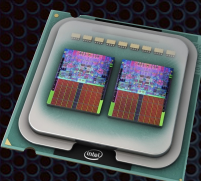
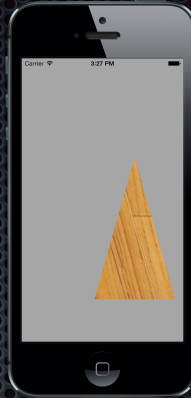
MUCH easier than it used to be!

Swapping Textures

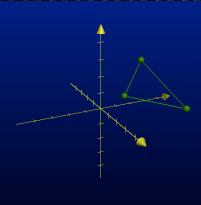
- ✦ Many textures can be loaded at one time
- ✦ Switch among loaded textures by calling `glBindTexture`
- ✦ Textures can be unloaded if there is insufficient memory
- ✦ Calling `glBindTexture` reloads the data automatically if it is not video memory resident



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
}
```

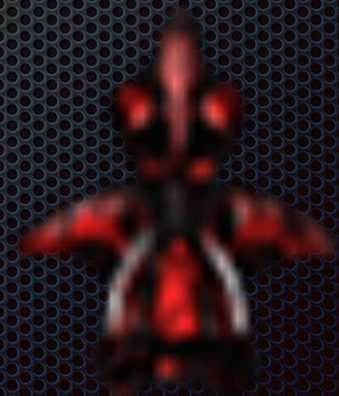


```
varying highp vec4 varyingTextureCoordinate;
uniform sampler2D textureUnit;
void main()
{ gl_FragColor = texture2D(textureUnit, varyingTextureCoordinate); }
```

Alpha Blending

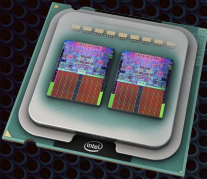
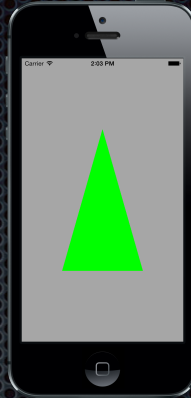
Problem: My sprites are all squares!

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

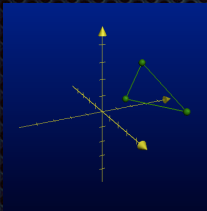


Shader Program Comparison

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)
```

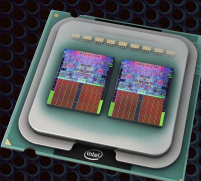
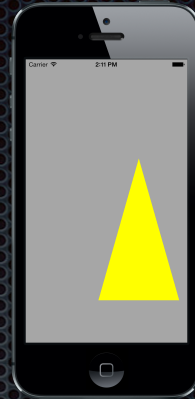


```
attribute vec2 position;
void main()
{
    gl_Position = vec4(position.x, position.y, 0.0, 1.0);
}
```

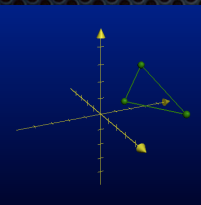


```
void main()
{
    gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
}
```

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)
```

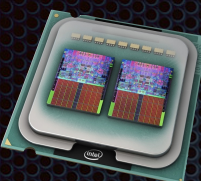
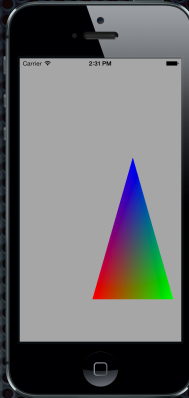


```
attribute vec2 position;
uniform vec2 translate;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
}
```

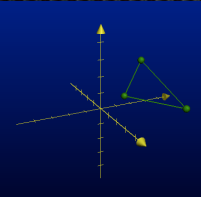


```
uniform highp vec4 color;
void main()
{
    gl_FragColor = color;
}
```

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
```

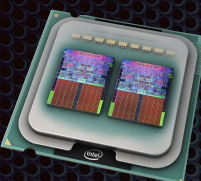
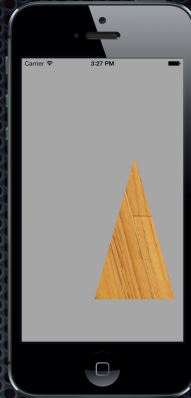


```
attribute vec2 position;
attribute vec4 color;
uniform vec2 translate;
varying vec4 colorInterpolated;
void main()
{
    gl_Position = vec4(
        position.x + translate.x, position.y + translate.y, 0.0, 1.0);
    colorInterpolated = color; // color data interpolated by rasterizer
}
```

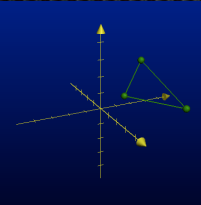


```
varying highp vec4 colorInterpolated;
void main()
{
    gl_FragColor = colorInterpolated;
}
```

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); }
```