# Mobile Application Programming OpenGL ES Introduction

# OpenGL ES



- C-Based Performance-Oriented Graphics Library
  - Wrapper libraries provided for Swift, C#, etc.
- Produces 2D images from 2D or 3D geometric data
- Mobile version of OpenGL
  - Includes nearly all OpenGL functionality
  - Removes seldom-used or legacy features
  - Used by non-mobile platforms also (eg. Playstation 4)









#### Hardware Acceleration Apple A7



2 Processors

Image Source: Chipworks





PowerVR G6430 (4x)

PowerVR SGX543 (3x)

Image Source: Chipworks

# Comparison of Apple GPUs

Mobile SoC GPU Comparison								
	PowerVR SGX 543	PowerVR SGX 543MP2	PowerVR SGX 543MP3	PowerVR SGX 543MP4	PowerVR SGX 554	PowerVR SGX 554MP2	PowerVR SGX 554MP4	PowerVR G6430
Used In	-	iPad 2/iPhone 4S	iPhone 5	iPad 3	-	-	iPad 4	iPhone 5s
SIMD Name	USSE2	USSE2	USSE2	USSE2	USSE2	USSE2	USSE2	USC
# of SIMDs	4	8	12	16	8	16	32	4
MADs per SIMD	4	4	4	4	4	4	4	32
Total MADs	16	32	48	64	32	64	128	128
GFLOPS @ 300MHz	9.6 GFLOPS	19.2 GFLOPS	28.8 GFLOPS	38.4 GFLOPS	19.2 GFLOPS	38.4 GFLOPS	76.8 GFLOPS	76.8 GFLOPS

www.anantech.com/show/7335/the-iphone-5s-review/7

# **OpenGL** Environment

#### UIWindow

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

Window →GLKView →GLES20 →Shaders



Data read from Scene and OBJ files





Data read from Scene and OBJ files



#### Barycentric Coordinates



## Primitives



## Vertex Shader



- Takes in vertex data and modifies it before using it to draw primitives (particularly triangles)
- Each vertex is processed separately from others (in parallel)
- Allows you to position and orient objects in the scene
  - E.g. Load a monster mesh in spread-eagle stance, move him in the world, and position his arms, legs, feet, so he is running

## Vertex Shader

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



Data read from Scene and OBJ files



## Rasterization



## Rasterization



# Fragment Shader



- Run once for each fragment (or pixel) that a triangle covers in the output pixel buffer to decide its color
- Each fragment is processed separately from others (again, in parallel)
- Allows you to color pixels to achieve effects (shading)
  - E.g. Draw your running monster, but cover him in green skin. Also make it look like the sun is shining on him

# Fragment Shader





# OpenGL Debugging



# OpenGL Debugging



# A Little Math - Vectors

#### Vectors

- Addition & Subtraction
- Scalar Multiplication
- Dot & Cross Product
- Magnitude & Normalization



# A Little Math - Matrices

- Matrices
  - Concatenation
  - Vector Multiplication
- Orthographic Transform
- Perspective Transform
- Viewport Transform

