

Advanced Core Graphics on the GPU

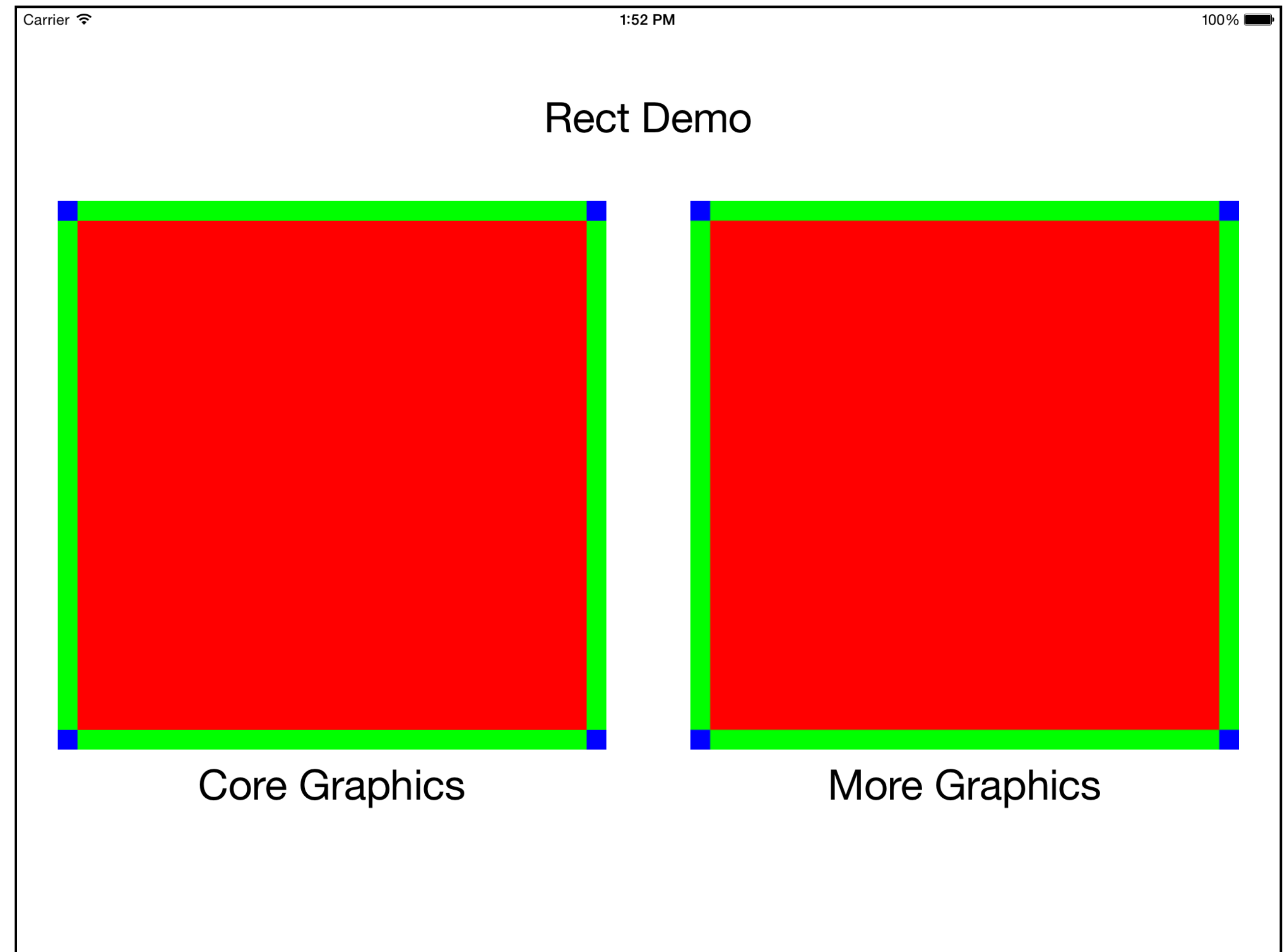
Ricardo Rendon Cepeda
@rendoncepeda



Content & Deliverables

Today's keywords... and demos...

- Core Graphics
- OpenGL ES
- Rasterization (CPU vs GPU)
- Shaders
- Primitives
- Case studies and war stories



Graphics Technologies in iOS [paraphrased]

*“**Core Graphics** is the native drawing engine for iOS apps... Although not as fast as OpenGL ES rendering, this framework is well suited for rendering custom 2D shapes dynamically.”*

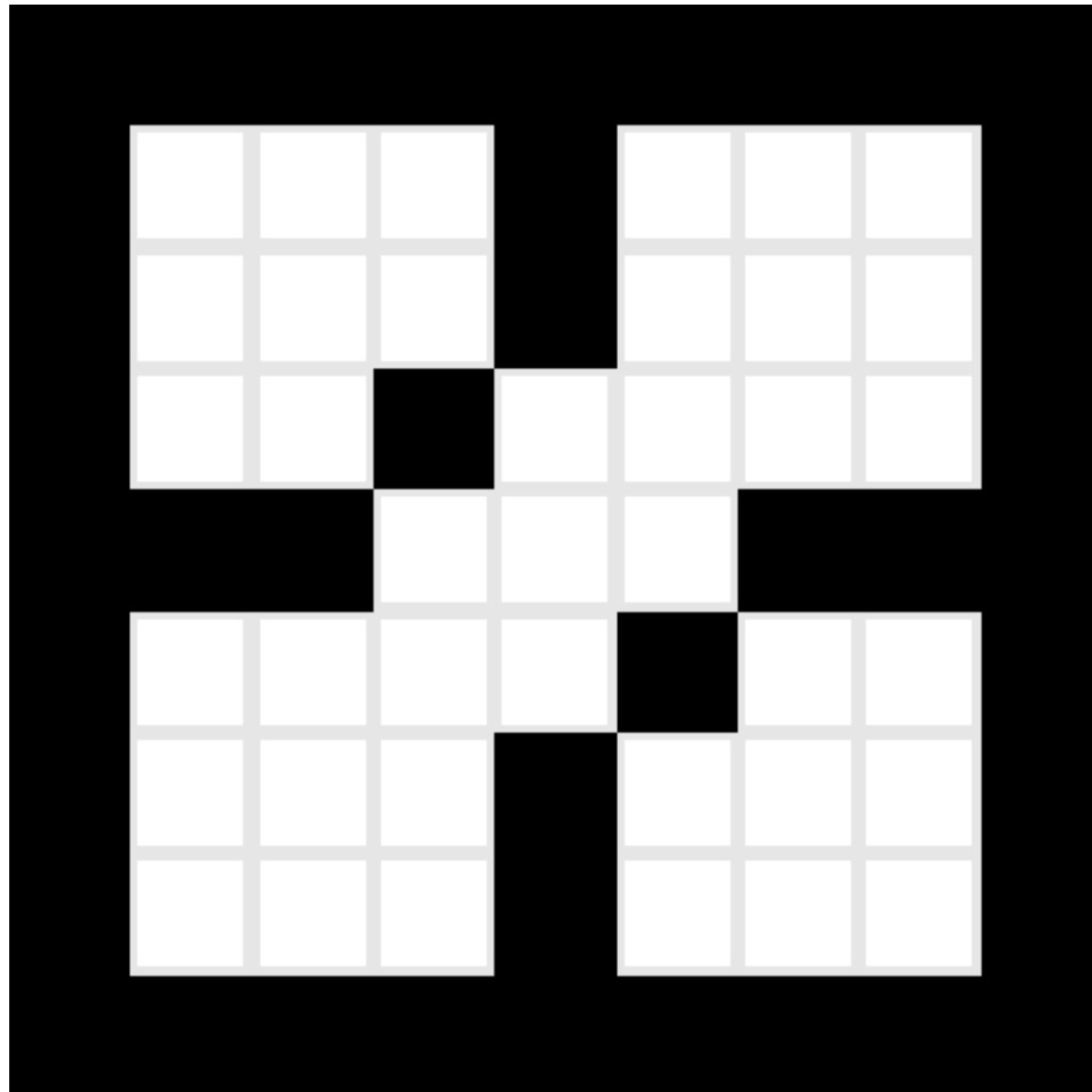
*“**OpenGL ES** handles advanced 2D and 3D rendering using hardware-accelerated interfaces... This framework gives you full control over the rendering process.”*

iOS Technology Overview: Media Layer

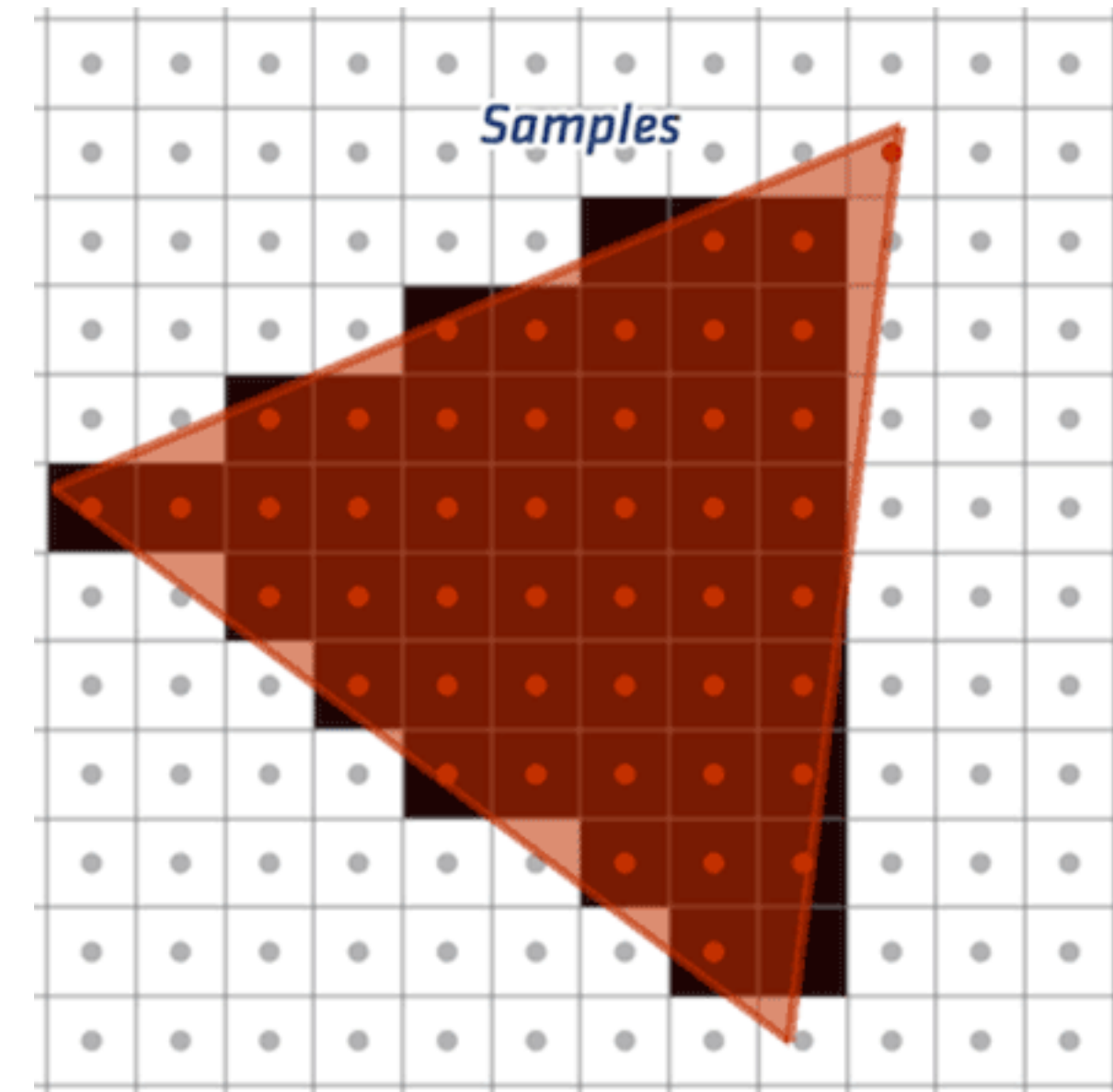
<https://developer.apple.com/library/ios/documentation/miscellaneous/conceptual/iphoneostechoverview/MediaLayer/MediaLayer.html>

Rasterization

Vectors (Shapes) -> Bitmap (Pixels)



CPU

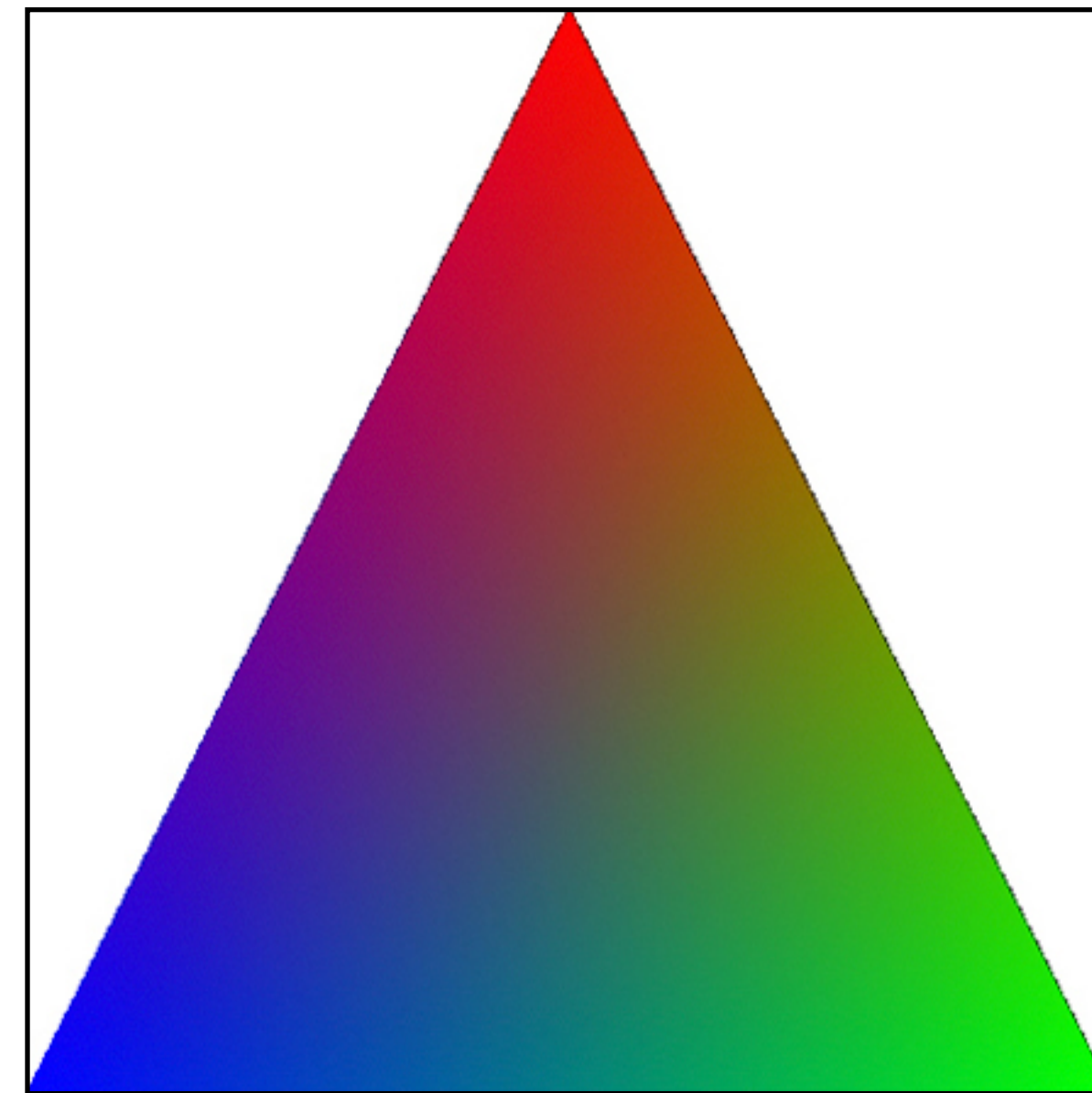


GPU

Shaders

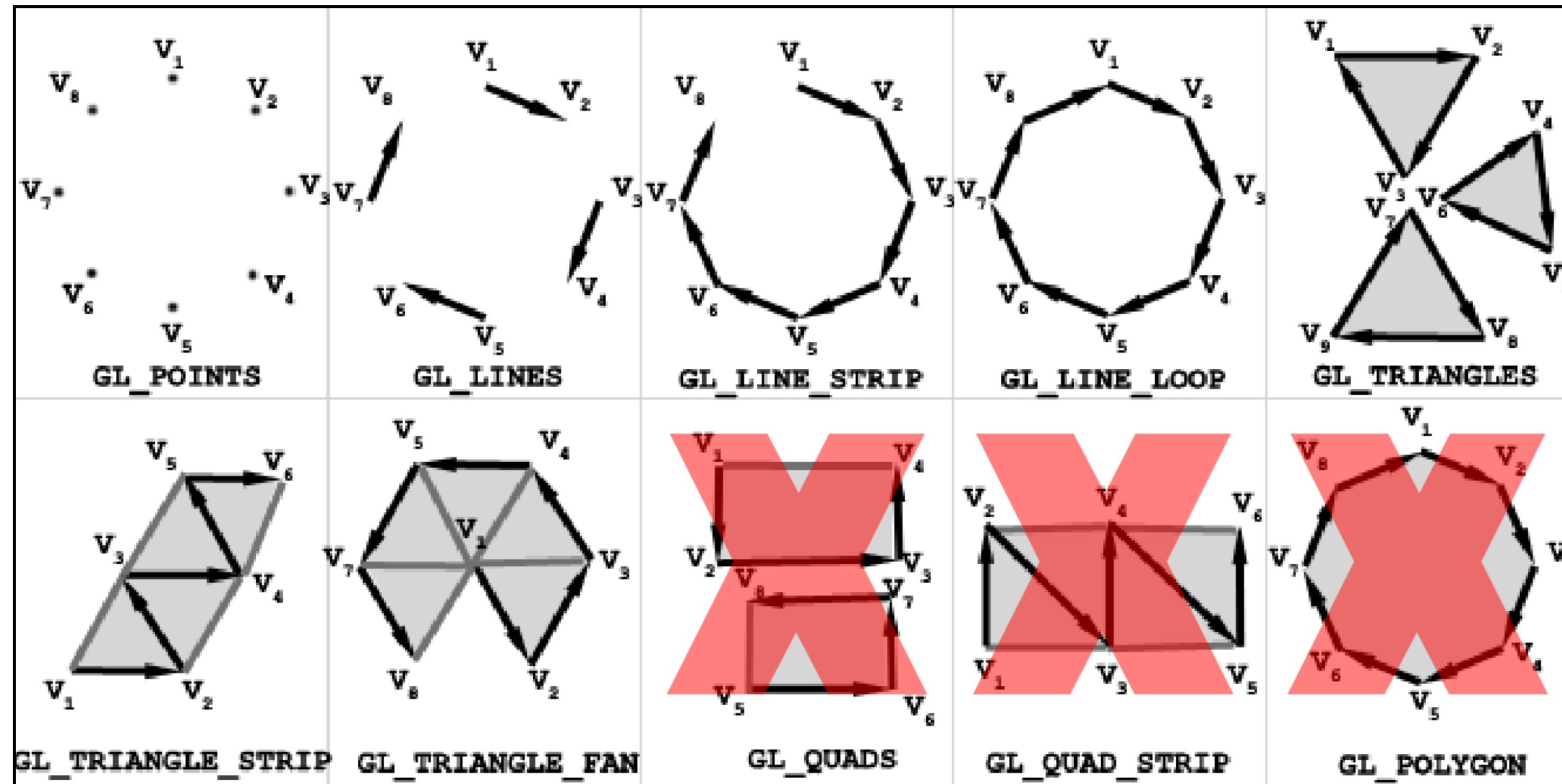
GPU mini-programs for shading

- Programmable pipeline
- Open**GL** Shading Language (*GLSL*)
- **Vertex** shader: *gl_Position*
- **Fragment** shader: *gl_FragColor*



Primitives

Vertex stream interpretations

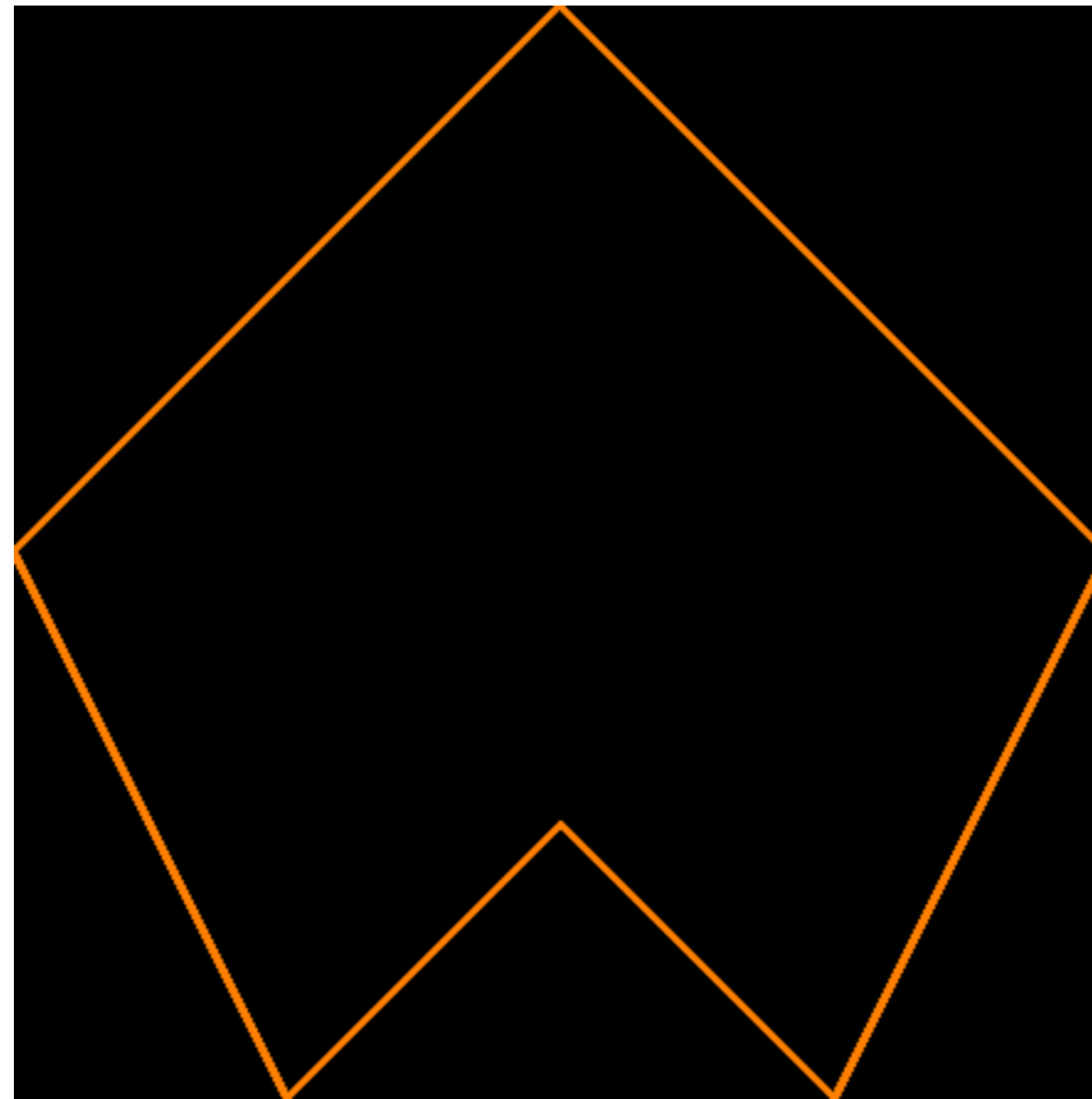


- Demo Time -

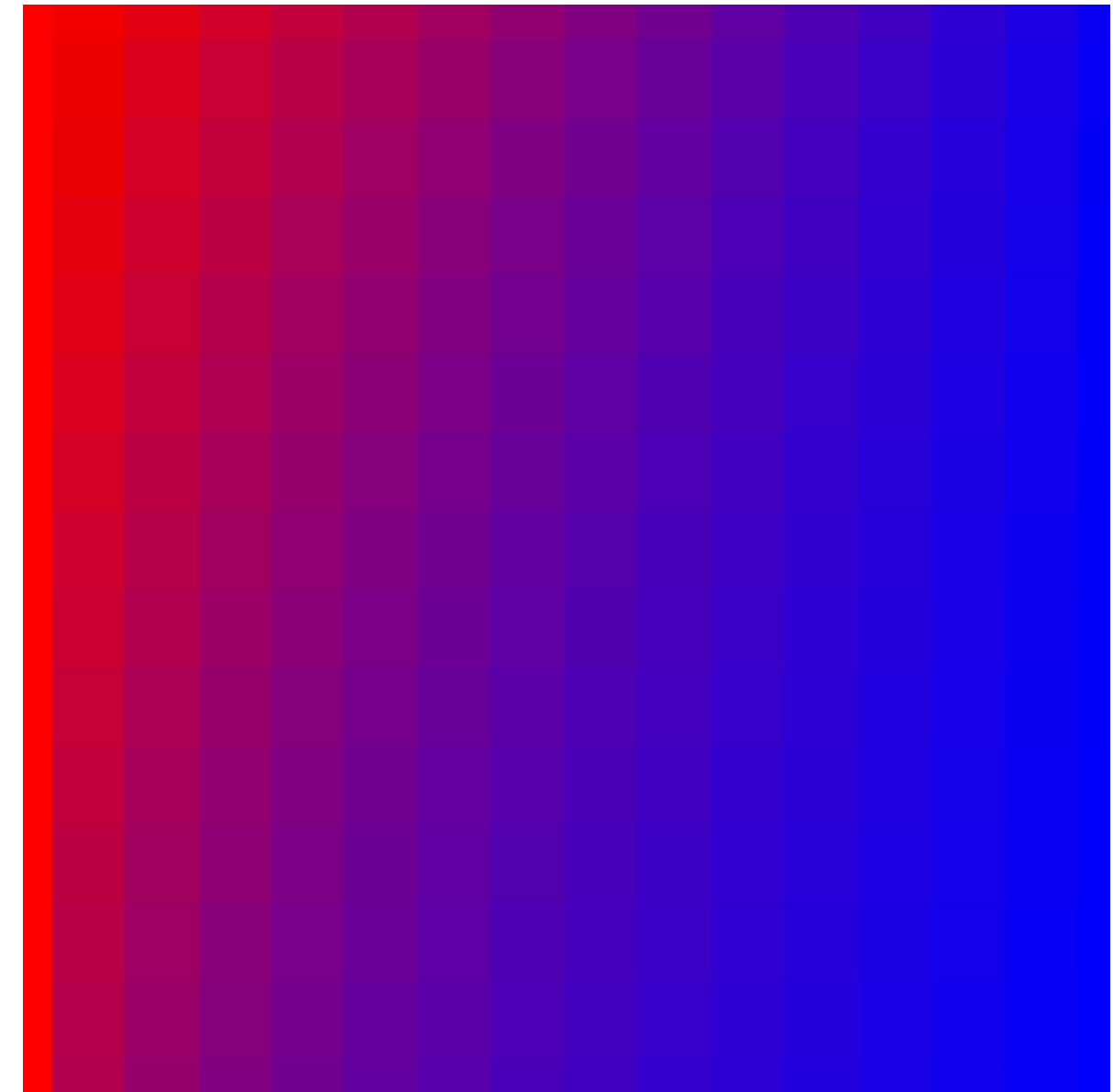
To summarize...



Bad.



Interesting...



Good!

Resources

Links, links, links

- Session Resources: github.com/ricardo-rendoncepeda/MoreGraphicsLibrary
- Drawing Bézier Curves: ciechanowski.me/blog/2014/02/18/drawing-bezier-curves
- Ray Wenderlich Tutorials: raywenderlich.com/tutorials
- Idean Work: idean.com/work



Thank You