Core Graphics

Justin Miller • @incanus77

Introduction

- I work at MapBox
- Primarily iOS apps & frameworks
- Cocoa programming for eight years

What Is Core Graphics?

- It’s a framework on Mac & iOS
- Low-level, C-based API
- Primarily concerned with 2D rendering

Main Components

- CGContext & CGBitmapContext
- CGColor
- CGImage
- CGPath
- CGGradient
- CGPDF*

Practical Uses

- Create & modify graphics on-the-fly
- Draw shadows & subtle details
- Use & reuse graphical assets efficiently

Contexts

- Works like a scratch pad
- Drawing destination (offscreen)
- Normal contexts
  - Drawing vectors, etc.
- Bitmap contexts
  - Drawing & altering existing images

Managing Contexts

- Implicit: UIView's drawRect: or CALayer's drawInContext:
- Explicit: CGBitmapContextCreate

UIKit Conveniences

- Managing bitmap contexts
  - UIGraphicsGetCurrentContext
- Getting image contents
  - UIGraphicsGetImageFromCurrentImageContext
- Threading concerns
  - Only use these on main thread!

Basic Drawing

- Paths
- Bezier curves
- Stroke & fill
- Images & patterns
- Shadows

Demo

Basic Drawing #1-3

Advanced Drawing

- Line styles (e.g., dash), caps, join
- Blend modes
- Masks
- Transparency layers
- Transformations
- Text

Demo

Advanced Drawing #1

Graphical Tiling

- Why?
  - Maps: 100,000+ pixels per view edge
  - Large documents like PDFs

Tiling Images

- Two approaches
  - Cutting an image on-the-fly
  - Generating pre-made tiles
- CGContextDrawImage

Tiling PDFs

- PDFs is a vector format
- Documents can be highly zoomed
- Documents can be large & detailed
  - Only want to draw limited bounds
- CGContextDrawPDFPage

Demo

Advanced Drawing #2

Wrap Up

- Demo project will be on GitHub
  - http://github.com/incanus
- Reach me at @incanus77
- Thanks!