Today

- Core Graphics Drawing
- Working with Images
- Core Animation
- Integrating with Other Apps
Quartz

- 2D drawing engine for iOS
  - works with all Core Animation, OpenGL ES, and UIKit
- layers painted to canvas
  - objects drawn in the order they appear
Context

- CGContextRef: where to draw output
  - UIGraphicsGetCurrentContext() for current window
  - can also draw to PDFs, bitmaps, etc.
- drawRect: UIView method called to draw contents
  - setNeedsDisplay to force redraw
Paths

- **defined by** `CGMutablePathRef` *(created via `CGPathCreateMutable()`)*
- **point**: single point in 2D space
  - `CGPathMoveToPoint`
- **line**: defined by endpoints (one endpoint is current point, so define other)
  - `CGPathAddLineToPoint`
Paths

- cubic Bezier curves: defined by 3 control points and endpoint
  - `CGPathAddCurveToPoint`
- quadratic Bezier curves: defined by 2 control points and endpoint
  - `CGPathAddQuadCurveToPoint`
Paths

- `CGPathCloseSubpath`: close path
- `CGContextAddPath`: add path to context
- `CGContextFillPath`: create shape from path
- `CGContextStrokePath`: create outline from path
Paths

- `CGContextSetLineWidth`: set stroke weight
- `CGContextSetLineCap`: set how lines end
- `CGContextSetLineDash`: draw dotted line
- `CGContextSetStrokeColorWithColor`: set color
  (UIColors have property for CGColor)
Paths

- PathsExample
Shadows

- defined by $x$-offset, $y$-offset, and blur
- `CGContextSetShadow` or `CGContextSetShadowWithColor` to draw shadows
Saving State

- push and pop from stack of states
  - resetting everything is annoying
- CGContextSaveGState: push state onto stack
- CGContextRestoreGState: pop value off stack
Shadows and State

▶ ShadowsExample
Gradients

- `CGGradientRef` can create axial and radial gradients (`CGGradientCreateWithColorComponents`)
  - axial: color varies along line (same color along perpendicular)
  - radial: color varies along concentric circles (same color along given circumference)
- need color space, colors, and locations for each color
Color Spaces

- `CGColorSpaceCreateDeviceRGB()`: RGB (red, green, blue)
- `CGColorSpaceCreateDeviceCMYK()`: CMYK (cyan, magenta, yellow, key)
- `CGColorSpaceCreateDeviceGray()`: grayscale
Gradients

▶ GradientsExample
Custom Views

- **drawRect** is a method of any `UIView`
- **subclass** `UIView`, `UIButton`, etc. to create custom views
Custom Views

- CustomCellsExample
- CustomButtonExample
UIImageView

- image: `UIImage` to be displayed
  - `UIImageView` can load remote URL
  - handles scaling and aspect ratios for you!
UIImagePickerController

- modal view controller allowing for selection of image from library
- `sourceType`: where to get images from
- `presentModalViewController` (just like any other view controller) to display
UIImagePickerControllerDelegate

- make sure to set delegate property!
- `imagePickerController:didFinishPickingMediaWithInfo: user selected image`
  - can get `UIImagePickerControllerEditedImage` or `UIImagePickerControllerOriginalImage` from passed dictionary
- `imagePickerControllerDidCancel: user clicked “Cancel” instead of selecting image`
Image Picker

▶️ ImageViewExample
UIImage ready to use with Core Graphics via CGImage

- `drawInRect:` draw UIImage in given CGRect
- `CGContextDrawImage:` draw CGImage in given CGRect
  - different coordinate system, so image will be upside-down!
Transforms

- `CGContextRotateCTM`: rotate about a point
- `CGContextScaleCTM`: change size
- `CGContextTranslateCTM`: move in a direction
Clipping

- rectangles are boring!
- remember CGMutablePathRef?
- CGContextClip restricts all drawings to last path
UIImage

▶ ImageManipulationExample
Core Animation

- animate 2D layers in 3D space
- implicit: set new properties, get smooth animation
- explicit: full control over timing, etc.
Core Animation

- add QuartzCore.framework to project
- #import "QuartzCore/CAAnimation.h"
Implicit Animation

- static methods to `UITableView` set animation properties
- enclosed in `beginAnimations:context:` and `commitAnimation`
- changed properties will animate automatically
Implicit Animation

- setAnimationTransition: lots of stock animations built in
- setAnimationDuration: time, in seconds, of animation
- setAnimationDelegate, setAnimationDidStopSelector: register callbacks
Implicit Animation

- ImplicitAnimationExample
Layers

- `CALayer` has contents to be animated
  - `UIView` has underlying layer, so we can animate them
  - provide content via `content` property, via delegate, or subclass
Keyframe Animation

- `CAKeyframeAnimation` creates a custom animation
- key path specifies what property will be animated
Animation Paths

- `CGMutablePathRef` defines an animation path
- `path` specifies path layer can be animated along
- `duration` for animation in seconds
Transforms

- CATransform3DMakeRotation: *rotation matrix*
- CATransform3DMakeScale: *scaling matrix*
- CATransform3DMakeTranslation: *translation matrix*
- values gives NSArray of frames
Keyframe Animations

- ExplicitAnimationExample
Opening Other Apps

- just like Android, other apps opened via URLs
  - `[[UIApplication sharedApplication] openURL: url]`
  - where `url` is an NSURL (schemes like http://, tel:, sms:)
Being Opened by Other Apps

- specify your own URLs with `<app name>-Info.plist` (like AndroidManifest.xml)
- `application:handleOpenURL: fired` when app opened from URL
Integrating with Other Apps

▶ IntegrationExample