iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

iOS: Core Graphics and Core Animation

Tommy MacWilliam

`tmacwilliam@cs.harvard.edu`

April 16, 2012

Today

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

Quartz

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- 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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- 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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

Paths

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

Paths

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

Paths

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- ▶ `PathsExample`

Shadows

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

Saving State

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

- `ShadowsExample`

Gradients

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- ► `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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

- ▶ `GradientsExample`

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

Custom Views

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- `CustomCellsExample`
- `CustomButtonExample`

UIImageView

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

UIImagePickerController

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

**Working with
Images**

Core
Animation

Integrating
with Other
Apps

- 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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

**Working with
Images**

Core
Animation

Integrating
with Other
Apps

- make sure to set `delegate` property!
- `imagePickerController:didFinish PickingMediaWithInfo:` user selected image
  - can get `UIImagePickerControllerEditedImage` or `UIImagePickerControllerOriginalImage` from passed dictionary
- `imagePickerControllerDidCancel:` user clicked "Cancel" instead of selecting image

► `ImageViewExample`

UIImage

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

**Working with
Images**

Core
Animation

Integrating
with Other
Apps

- `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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

Clipping

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

UIImage

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- `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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

**Core
Animation**

Integrating
with Other
Apps

- static methods to `UIView`s 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

▶ `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

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

- ► `CAKeyframeAnimation` creates a custom animation
- ► key path specifies what property will be animated
  - ► animatable properties: `http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/CoreAnimation_guide/Articles/AnimProps.html`

Animation Paths

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

- `CATransform3DMakeRotation`: rotation matrix
- `CATransform3DMakeScale`: scaling matrix
- `CATransform3DMakeTranslation`: translation matrix
- `values` gives `NSArray` of frames

- `ExplicitAnimationExample`

Opening Other Apps

iOS: Core
Graphics and
Core
Animation

Tommy
MacWilliam

Core Graphics
Drawing

Working with
Images

Core
Animation

Integrating
with Other
Apps

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

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

- `IntegrationExample`