Lecture 2

CS2049: Intermediate iPhone Development

Instructor: Daniel Hauagge
Today’s Lecture

• Frameworks
  • AVFoundation
  • CIDetector
• Layers and Views
• Language features:
  • Lazy initialization
  • try!
Today’s App
Video Capture

with AVFoundation
# iPhone Camera

<table>
<thead>
<tr>
<th>Front Camera</th>
<th>Rear Camera</th>
</tr>
</thead>
<tbody>
<tr>
<td>5MP</td>
<td>12MP</td>
</tr>
<tr>
<td>HD video</td>
<td>4K video</td>
</tr>
<tr>
<td>Burst Mode</td>
<td>Burst Mode</td>
</tr>
<tr>
<td>Retina Flash</td>
<td>True Tone Flash</td>
</tr>
<tr>
<td></td>
<td>Slow motion: 120 to 240 FPS</td>
</tr>
<tr>
<td></td>
<td>Video stabilization (optical and digital)</td>
</tr>
<tr>
<td></td>
<td>Time lapse video</td>
</tr>
<tr>
<td></td>
<td>Panoramas</td>
</tr>
</tbody>
</table>
Cool Apps
## Levels of

<table>
<thead>
<tr>
<th>UImagePickerController</th>
<th>AVFoundation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy of use</td>
<td>Greater control</td>
</tr>
<tr>
<td>Only one class to deal with</td>
<td>Breaks down pipeline into input, output, session management</td>
</tr>
<tr>
<td>Provides UI for capture</td>
<td></td>
</tr>
</tbody>
</table>
AVFoundation

• Still and video capture
• Video preview using CALayer
• Audio level metering
• Access to raw data
• Video stabilization
• HDR Video
AVFoundation

• Makes sense when:
  • You need fine grained control
  • Independent points of interest for focus or exposure
  • Access to video frame data (with accurate timing data)
  • Per frame metadata (e.g., exposure)
  • Configurable frame-rate, output format, resolution
Capture Basics

Hardware

Inputs
- Camera
- Microphone

Output
- Compressed Video to File
- Still Frame
- Audio Data
- Video Data
Capture Basics

Hardware    Inputs    Output    Display

AVCaptureDevice → AVCaptureDevice Input → AVCaptureSession → AVCaptureVideo PreviewLayer
Open this webpage to see instructors code

http://10.148.6.140:8000
Capture Basics

Hardware

AVCaptureDevice

Inputs

AVCaptureDevice Input

AVCaptureVideo DataOutput

AVCaptureSession

Output

sampleBufferDelegate (our code)

Display

CALayer

AVCaptureVideo PreviewLayer
# Layers and Views

<table>
<thead>
<tr>
<th>UIView</th>
<th>CALayer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can receive touch events</td>
<td>Represent position, shape and anchor point</td>
</tr>
<tr>
<td>Are always backed by a Layer on iOS</td>
<td>Do not receive touch events</td>
</tr>
<tr>
<td></td>
<td>Light-weight</td>
</tr>
</tbody>
</table>
|                               | No implicit animations                  | Implicit animations
Layers and Views

UIKit views

Core Animation layers
CIDetector

• Faces
  • Eyes
  • Smile detection
  • Face angle
• Rectangles
• QR Codes
• Text
try!

var x = try! someFunc() ≈

do {
    var x = someFunc()
    assert(false)
} catch {

• In case you are sure an exception won't be thrown

• Compiler wraps code in runtime assertion

• Runtime error (you app dies) if an actual exception is thrown
try?

```swift
var x = try? someFunc()
```

≈

```swift
var x: Int?
do {
    x = someFunc()
} catch {
    x = nil
}
```

- An exception can be thrown but you can continue by setting the variable to nil
- No runtime error
guard

var y : Int?
guard var x = y else {
    return
}

≈

var y : Int?
if y == nil {
    return
}
x = y!

• An exception can be thrown but you can continue by setting the variable to nil

• No runtime error
guard