Praktikum Entwicklung von Mediensystemen mit iOS

Wintersemester 2013/2014

Christian Weiß, Dr. Alexander De Luca
Today

- Assignment 2
- More iOS:
  - Text and touch input
  - Accelerometer
  - Animations and drawing
- Assignment 3
- Outlook
Text Input

- UITextField
- Requires UITextFieldDelegate
- Process text input:

```c
// dismiss keyboard
-(BOOL)textFieldShouldReturn:(UITextField *)textField {
    // calls textFieldShouldEnd where you can check
    // for invalid input
    textField resignFirstResponder;
    return YES;
}

// get text input
-(void)textFieldDidEndEditing:(UITextField *)textField {
    NSString* textInput = textField.text;
}
```

- Use UITextView for multiple lines of text
Touch Input

• Use gesture recognizers
  • Init in View Controller or add in Storyboard
  • Create IBAction:
    
    - (IBAction)swipeRecognized:(id)sender {
      // handle gesture
    }

• Use touches methods, e.g.:

    -(void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {
      UITouch *touch = [touches anyObject];
      CGPoint p = [touch locationInView:self.view];
      // Use p.x and p.y
    }
Accelerometer

- g-force values for x, y, z (1g = normal acceleration caused by gravity)
- Access accelerometer by singleton object (requires Delegate)
  
  ```
  [[UIAccelerometer sharedAccelerometer] setDelegate:self];
  ```
Accelerometer

• Get sensor data via Delegate method:
  
  ```
  (void)accelerometer:(UIAccelerometer *)accelerometer didAccelerate:
  (UIAcceleration *)acceleration {
    NSLog(@"x acceleration is %d", acceleration.x);
  }
  ```

• Detect device orientation: Low-pass filter removes instant motion.

• Detect instant motion (e.g. shaking): High-pass filter removes gravity component.
Location

• CLLocationManager

• Configuration (requires Delegate):

```swift
#import <CoreLocation/CoreLocation.h>

CLLocationManager *locationManager = [[CLLocationManager alloc] init];
[locationManager setDesiredAccuracy:kCLLocationAccuracyBest];
[locationManager setDelegate:self];
[locationManager startUpdatingLocation];
```

• Get location data via Delegate method:

```swift
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations
  // Use locations to get longitude and latitude
}

- (void)locationManager:(CLLocationManager *)manager didExitRegion:(CLRegion *)region
- (void)locationManager:(CLLocationManager *)manager didEnterRegion:(CLRegion *)region
```
Animations - Example

Sliding Sam
Animations

• Views can fly around, rotate, fade in/out and much more.
• Animations can make your app appear much more exciting.
• The following properties of the UIView class are animatable:
  • @property frame
  • @property bounds
  • @property center
  • @property transform
  • @property alpha
  • @property backgroundColor
  • @property contentStretch
Example
Fade In / Out

• Change alpha from 0 (transparent) to 1 (opaque) in 3 seconds:

```swift
imageView.alpha = 0.0;
[UIView animateWithDuration:3.0 animations:^{
    imageView.alpha = 1.0;
}];
```

• This **Block** syntax makes your code easier to read. You don’t have to memorize it - code completion is your friend :-)
Rotate

• Rotate by 90° in 3 seconds:

```swift
 UIView.animateWithDuration:3.0 animations:^{
    imageView.transform = CGAffineTransformMakeRotation(M_PI_2);
}];
```
Scale

- Scale from 10% to 100% in 3 seconds:

```objective-c
imageView.transform = CGAffineTransformMakeScale(0.1, 0.1);
[UIView animateWithDuration:3.0 animations:^{
    imageView.transform = CGAffineTransformMakeScale(1, 1);
}];
```
Move

- Move from origin to center:

```swift
imageView.center = CGPointMake(0, 0);

[UIView animateWithDuration:3.0 animations:^{
    imageView.center = imageView.superview.center;
}];
```
Animation Options

- Multiple animations at once are possible

- Options examples:
  - `UIViewAnimationOptionCurveEaseInOut`: start slowly, accelerate, stop slowly
  - `UIViewAnimationOptionTransitionFlipFromLeft`: flip around vertical axis

- Completion examples:
  - Start another animation
  - Play sound
Drawing

• Instead of using PNGs, you can draw custom shapes with CoreGraphics (a.k.a. CG or Quartz 2D).

• Example with drawRect:

```c
@interface CustomShape : UIView

- (void)drawRect:(CGRect)rect {
    CGContextRef context = UIGraphicsGetCurrentContext();
    CGContextSetFillColorWithColor(context, [UIColor redColor].CGColor);
    CGContextFillEllipseInRect(context, rect);
}
```
Drawing

• Core Graphics also supports shadows, gradients, layers etc. (documentation)

• Image filters à la Instagram can be done with Core Image (documentation)

• 3D drawing can be done with OpenGL (documentation)
Assignment 3

• Do 2 out of 4 programming tasks (choose whichever interests you most)

• Due next Wednesday (6.11. 12:00), upload to Uniworx

• For the project phase, form teams of four
Outlook

• 30.10. (today): Assignment 3 is the last one
• 6.11. (next week): Start of team projects (Brainstorming)
• 13.11.: Present your app idea

• Questions?
• Problems with certificate request, provisioning profiles?