Announcement

- Final Project Descriptions are due on Tuesday by 11:59 PM
- Final Project Presentations are on Wednesday
- Lab 4 is due on Wednesday July 17th

Today’s Topics

- Final Projects
- Mapkit
- Core Location
Project Descriptions

- Team Member’s names
  - “Groups” of 1 – 2 students
- 1 Paragraph description of the project
- Project Name
- Due by Tuesday July 10th at 11:59 PM

Final Project Point Distribution and Due Dates

- The final project accounts for 30% of your final grade
  - Final project score is out of 100 points
- Final Project Group Description – 5 points
  - Due on Tuesday July 9th by 11:59 PM
- Project Proposal Presentation – 15 points
  - Submit as PPT, Keynote, or Google slides
  - Due on Wednesday July 10th by 5:30 PM
- Project Update Presentation – 10 points
  - Provide a brief description of what has been accomplished in the email
  - Submit code to demonstrate the accomplishments
  - Due on Wednesday July 24th by 5:30 PM
- Final Project Code - 70 points
  - Due on Wednesday July 31st by 5:30 PM

Submit all portions of the final project to cse438ta@gmail.com

Late submissions will result in a 0 for that portion of the final project
Final Project Proposal Presentations

• Each group will have up to 10 minutes to present their app

• If someone from your group is not here to present your project you will receive a 0 that portion of the project

Requirements for Project Proposal Presentation

• Motivate the need for the app

• Explain the pieces involved in creating the app

• Show a distribution of work for each team member

• Provide a detailed timeline with tasks and deliverable dates
  – What will be accomplished by the update presentation?
    • Will you demo a skeleton GUI by the update meeting?
    – What will you show at the final presentation?

• Include a wireframe for your entire application
  – Each “screen” of your app should be included
  – Consider using Keynote, PowerPoint, fluidui.com or ninjamock.com
Final Project
Student Ideas?

Apps from Previous Semesters
Matt Lanter’s App

- WashU Maps
  - Integrate with information about dining hours and menus
  - Adding campus transportation and metro bus routes to map along with schedules
  - Add department locations to map (e.g. it will show you which building they are in)

Matt Lanter’s Apps

- WebSTAC
  - Adding other WebSTAC functionality (grades, gpa, course listings, registration, registration worksheet, etc.)
  - Add Telesis functionality
  - Add support for adding campus card points (either using function of WebSTAC or https://acadinfo.wustl.edu/eTransact/)
  - Add other campus life information, such as dining locations, hours, menus, important phone number
Meetups

- Created by Jake LaMountain and James Farner
- Helps you keep track of your friends’ schedules and locations
- Use GPS coordinates and a map of WashU

I am Here

- Created by HT Kwon and Andrew Shaw
- Game where you “tag” your friends
- Uses coreLocation and mapSDK
**eFlick**

- Created by Justin McClain and Simon Tam
- Game with a purpose
- Help tag events happening at WashU
- Uses core animation and gestures

**iFitness Manager**

- Developed by Eric Peters
- Continued on as an independent study
- Made it to the App Store
  - Paid and Free Versions available
iFitness Manager

iDrink

- Developed by Josh Mason and Julie Betlach

- Elevator Pitch:
  - Do you have various alcohols and other ingredients and need to make a drink, but don’t know what to mix together?
Gomoku and Sudoku

Amber Alert
Games with a Purpose

http://en.wikipedia.org/wiki/Human-based_computation_game

Guest Presenters

- **Team from Olin Business School**
  - Sandy Boillot
    - sboillot@wustl.edu
  - Jill Tomich
  - Bart Day

- **Business Management App for small businesses**
MapKit

What is MapKit?

- API to display Maps
- Classes to translate between CLLocation and human-readable addresses
- Support for “annotations” (pins on a map)
- Reverse Geocoding
MKMapView

- Handles display of map
- “Map” & “Satellite” types
- Panning and Zooming
- Annotations
- Display User Location

Properties in MKMapView

```swift
var region: MKCoordinateRegion
var centerCoordinate: CLLocationCoordinate2D
var userLocation: MKUserLocation
var annotations: [MKAnnotation]
var delegate: MKMapViewDelegate?
MKMapType mapType
```
MKMapViewDelegate

- Callback methods about loading state:

  func mapViewWillStartLoadingMap(_ mapView: MKMapView)
  func mapViewDidFinishLoadingMap(_ mapView: MKMapView)
  func mapViewDidFailLoadingMap(_ mapView: MKMapView, withError error: Error)

- Callback methods about region changes:
  func mapView(_ mapView: MKMapView, regionWillChangeAnimated animated: Bool)
  func mapView(_ mapView: MKMapView, regionDidChangeAnimated: animated Bool)

- Callback methods to customize and interact with annotations

  func mapView(MKMapView, viewFor: MKAnnotation)
  func mapView(MKMapView, didAdd: [MKAnnotationView])
  func mapView(MKMapView, annotationView: MKAnnotationView, calloutAccessoryControlTapped: UIControl)
MKAnnotation

• A protocol - not a class

• Add to a MapView to plot pins
  var coordinate: CLLocationCoordinate2D

  var title: String?
  var subtitle: String?

MKPlacemark

• Conforms to MKAnnotation protocol
• Convenience for holding human-readable addresses alongside Coordinate

init(coordinate: CLLocationCoordinate2D, addressDictionary: [String : Any]?)

• Easy to convert between AddressBook addresses and location:
  – thoroughfare, subThoroughfare, locality, subLocality,
    administrativeArea, subAdministrativeArea, postalCode, country,
    countryCode
MKUserLocation

- Special case of an MKAnnotation
- Represents device’s location only
  - You do not create instances of this class directly
  - Retrieve an existing MKUserLocation object from userLocation property of map

```swift
var location: CLLocation?
var isUpdating: Bool
var title: String?
var subtitle: String?
```
Core Location

- What is it?

- Core Location

  Activate service

  Location ring
Core Location

[Image of a smartphone with location icons]
Core Location

• Location Technologies

Bootstrap
Crosscheck
Complement
Core Location Framework

• The core classes and protocols
  • Classes
    – CLLocation
      • Represents a point and vector in the real world
    – CLLocationManager
      • Allows you to get a CLLocation
  • Protocol
    – CLLocationManagerDelegate
Core Location Framework

- `CLLocationManagerDelegate` protocol

- Several useful optional methods
  
  ```swift
  func locationManager(_ manager: CLLocationManager, didUpdateLocations: [CLLocation])
  func locationManager(_ manager: CLLocationManager, didFailWithError: Error?)
  ```

- Called asynchronously on main thread
- Issues movement-based updates

Getting a Location

- Starting the location service
  
  ```swift
  let locationManager = CLLocationManager()
  locationManager.delegate = self
  locationManager.requestWhenInUseAuthorization()
  locationManager.startUpdatingLocation()
  ```
Getting user location

• iOS 8 introduced additional requirements to obtain your location
  – Call the requestWhenInUseAuthorization method
  – Add an entry to your plist file to request location
    • NSLocationWhenInUseUsageDescription

Getting a Location – Using Event Data

```swift
func locationManager(manager: CLLocationManager, didUpdateLocations locations: [CLLocation]) {

    let aLocation = locations[0]
    let howRecent = aLocation.timestamp.timeIntervalSinceNow

    if (howRecent < -10) { return }

    if (aLocation.horizontalAccuracy > 100) { return }

    double lat = aLocation.coordinate.latitude
    double long = aLocation.coordinate.longitude

}
```
Desired Accuracy

- Choosing an appropriate accuracy level
  locationManager.desiredAccuracy = kCLLocationAccuracyBest

- Choose an appropriate accuracy level
  - Higher accuracy impacts power consumption
  - Lower accuracy is “good enough” in most cases

- Can change accuracy setting later if needed

- Actual accuracy reported in CLLocation object

Distance Filter

- Choosing an appropriate update threshold

- New events delivered when threshold exceeded
  locationManager.distanceFilter = 3000
### Stopping the Service

```swift
locationManager.stopUpdatingLocation()
```

- Restart the service later as needed

- Also able to pause service and run in background
  - `var pausesLocationUpdatesAutomatically: Bool`
  - `var allowsBackgroundLocationUpdates: Bool`

### Responding to Errors

- User may deny use of the location service

- Results in a `kCLErrorDenied` error

- Protects user privacy

- Occurs on a per-application basis
Responding to Errors

- Location may be unavailable
- Results in a kCLErrorLocationUnknown error
- Likely just temporary
- Scan continues in background

Demo
GPS Data
Demo
Geocoding

https://github.com/ooper-shlab/GeocoderDemo-Swift