Reminder: Final Project

● 1-slide from group next Monday (2/6):
  ● 2/40 of final project grade
● Slide should contain 3 bullets
  1. **Problem you intend to work on**
     ● Solve WPI/societal problem (e.g. walking safe at night)
     ● Use at least location, 1 sensor or camera
     ● If games, must gamify solution to real world problem
  
  2. **Why this problem is important**
     ● E.g. 37% of WPI students feel unsafe walking home
  
  3. **Summary of envisioned mobile app (?) solution**
     1. E.g. Mobile app automatically texts users friends when they get home at night

● Can bounce ideas of me (email, or in person)
● Can change idea any time
Location-Aware Apps from Past Offerings
Location-Aware Ideas from Previous Offerings

- **Ground rules:**
  - Apps must use mobile, location or sensors
  - Try to solve problems of benefit to WPI community

- More than half of apps used location.

- **Give me some space:** Bianchi, Chow, Martinez ’16
  - Find available study spaces on campus during exam week
  - Set up geoFences at study locations, count users in/out
Location-Aware Ideas from Previous Offerings

- **HomeSafe**: Nickerson, Feeley, Faust ’16
  - Safety app
  - Automatically sends message to users’ subscribers when they get home safely
Location-Aware Computing

- **Definition:** Location-aware applications generate outputs/behaviors that depend on a user’s location

- **Examples:**
  - Map of user’s “current location”
  - Print to “closest” printer
  - Apps that find user’s friends “closeby”
  - Reviews of “closeby” restaurants

- Above apps require first determining user’s location
Determining User Location on Smartphones
Location Tracking on Smartphones

- **Outdoors:** Uses GPS (More accurate)
- **Indoors:** WiFi or cell tower signals (Location fingerprinting, less accurate)
Global Positioning System (GPS)

- 27 satellites orbiting earth
- **20,000 km above earth** (Medium earth orbit)
- 6 orbital planes with 4 satellites each
- 4 satellites visible from any spot on earth
- Location of any location on earth specified as <longitude,latitude>
- E.g. Worcester MA has **Latitude**: 42.2625, **Longitude**: -71.8027778
GPS User Segment

- **Triangulation:** GPS receiver calculates user’s position by comparing roundtrip delay of signals to multiple satellites at known positions.

- **Accuracy within 5 - 10 meters (16-32 feet)**

http://adamswalk.com/gpx-2/
Determining User Location

- GPS reasonably accurate but
  - Requires line-of-sight between satellite and car receiver
  - Only works OUTDOORS (signals don’t penetrate buildings)
  - Lag/delay in acquiring satellites (~270 msec) or re-acquiring if lost
  - Drains battery power

- **Alternative**: Use Wi-Fi location sensing indoors
WiFi Location Fingerprinting

- **Key insight:** At each (X,Y) location, WiFi APs observed + their signal strengths, is unique

- **WiFi Location fingerprinting:** Infer device’s location based on combination of Wi-Fi access points seen + Signal Strengths
Location Estimation using Wi-Fi Fingerprinting

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>SIGNAL STRENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>AP1  AP2  AP3  AP4</td>
</tr>
<tr>
<td>80</td>
<td>145  32  28  12  8</td>
</tr>
<tr>
<td>40</td>
<td>145  36  20  10  6</td>
</tr>
<tr>
<td>220</td>
<td>355 - 25  36  44</td>
</tr>
<tr>
<td>260</td>
<td>355  4  21  39  42</td>
</tr>
<tr>
<td>350</td>
<td>210  16 -  28  36</td>
</tr>
<tr>
<td>380</td>
<td>145  22  12 -  44</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Google builds and stores this database (APs + Signal Strength) at each X,Y location.

- Inference Algorithms
  - Min. Threshold
  - Euclidean Dist.
  - Joint Probability
  - Bayesian Filters
How to Build table of APs observed at (X,Y) Locations?

- Devices (e.g. smartphone) with GPS and WiFi turned on simultaneously build table
- Send to third party repositories (e.g. Wigle.net) or Google
- Also called war driving
- Can record cell tower signal strength instead of AP

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>AP1</th>
<th>AP2</th>
<th>AP3</th>
<th>AP4</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>145</td>
<td>32</td>
<td>28</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>40</td>
<td>145</td>
<td>36</td>
<td>20</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>220</td>
<td>355</td>
<td>-</td>
<td>25</td>
<td>36</td>
<td>44</td>
</tr>
<tr>
<td>260</td>
<td>355</td>
<td>4</td>
<td>21</td>
<td>39</td>
<td>42</td>
</tr>
</tbody>
</table>

Google gathers Location, AP seen Data if you consent

GPS satellites
Location consent
Allow Google's location service to collect anonymous location data. Some data may be stored on your device. Collection may occur even when no apps are running.

Disagree Agree
Location Sensing in Android Apps
Google Location APIs

- Location API is now part of Google Play Services (newer!)
- Older Android framework location APIs (`android.location`)
  - Used by most books, online sources. We will use that

- `LocationManager`:
  - Android module receives location updates from GPS, WiFi, etc
  - App registers/requests location updates from LocationManager
Requesting Location Updates

Your app

LocationManager

requestLocationUpdates(LocationListener)

onStatusChanged

onProviderEnabled

onProviderDisabled

Create listener for Location info

Callback methods called by Location manager (e.g. when location changes)

Type of location Provider (e.g. cell tower and Wi-Fi based)

Listener that receives callbacks

// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager) this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network location provider.
        makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {} 

    public void onProviderEnabled(String provider) {} 

    public void onProviderDisabled(String provider) {} 
};

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, locationListener);
Requesting User Permissions
https://developer.android.com/guide/topics/location estratégias.html

- Need smartphone owner’s permission to use their GPS

```xml
<manifest ...
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    ...
    <!-- Needed only if your app targets Android 5.0 (API level 21) or higher. -->
    <uses-feature android:name="android.hardware.location.gps" />
    ...
</manifest>
```

- ACCESS_FINE_LOCATION: GPS
- ACCESS_COARSE_LOCATION: WiFi or cell towers
GettingCached Copy of Location (Fast)

https://developer.android.com/guide/topics/location стратегий.html

- Getting current location may take a while
- Can choose to use location cached (possibly stale) from Location Manager

```java
String locationProvider = LocationManager.NETWORK_PROVIDER;
// Or use LocationManager.GPS_PROVIDER

Location lastKnownLocation = locationManager.getLastKnownLocation(locationProvider);
```
Stopping Listening for Location Updates


- Location updates consume battery power
- Stop listening for location updates whenever you no longer need

```java
// Remove the listener you previously added
locationManager.removeUpdates(locationListener);
```
Distance Travelled Updates using Services
Example from Head First Android
Example: Odometer (Distance Travelled) updates as a Services
(Ref: Head First Android pg 541)

- **Services**: long running background processes, no UI

- May want background service (a module in our app) to continuously retrieve location updates from LocationManager, forward our Activity updates

- Ref: Head First Android pg 541
  - Example of using a Service
  - Nice Example app using Odometer Service
  - Tracks distance travelled
  - Gets, displays distance travelled every 10 secs
Example: Odometer (Distance Travelled) updates as a Services
(Ref: Head First Android pg 541)

- Example odometer app that tracks distance travelled
- `getMiles()`, displays distance travelled every 10 seconds

Study this example!!!
Location Representation
Semantic Location

- GPS represents location as \(<\text{longitude,latitude}>\)
- **Semantic location** is better for reasoning about locations
- E.g. Street address (140 Park Avenue, Worcester, MA) or (building, floor, room)
- Android supports:
  - **Geocoding**: Convert addresses into longitude/latitude coordinates
  - **Reverse geocoding**: convert longitude/latitude coordinates into human readable address

**Android Geocoding API**: access to **geocoding** and **reverse geocoding** services using HTTP requests
Google Places API Overview

- Access **high-quality photos** of a place
- Users can also add place information to the database
  - E.g. business owners can add their business as a place in Places database
  - Other apps can then retrieve info after moderation

- **On-device caching:** Can cache places data locally on device to avoid roundtrip delays on future requests
Google Places

- **Place**: physical space that has a name (e.g. local businesses, points of interest, geographic locations)
  - E.g Logan airport, place type is **airport**

- **API**: Provides Contextual information about places near device.
  - E.g: name of place, address, geographical location, place ID, phone number, place type, website URL, etc.

- Compliments geographic-based services offered by Android location services
### Sample Place Types

<table>
<thead>
<tr>
<th>Accounting</th>
<th>Airport</th>
<th>Amusement Park</th>
<th>Aquarium</th>
<th>Art Gallery</th>
<th>ATM</th>
<th>Bakery</th>
<th>Bank</th>
<th>Bar</th>
<th>Beauty Salon</th>
<th>Bicycle Store</th>
<th>Book Store</th>
<th>Bowling Alley</th>
<th>Bus Station</th>
<th>Cafe</th>
<th>Campground</th>
<th>Car Dealer</th>
<th>Car Rental</th>
<th>Car Repair</th>
<th>Car Wash</th>
</tr>
</thead>
<tbody>
<tr>
<td>hospital</td>
<td>insurance agency</td>
<td>jewelry store</td>
<td>laundry</td>
<td>lawyer</td>
<td>library</td>
<td>liquor store</td>
<td>local government office</td>
<td>locksmith</td>
<td>lodging</td>
<td>meal delivery</td>
<td>meal takeaway</td>
<td>movie rental</td>
<td>movie theater</td>
<td>moving company</td>
<td>museum</td>
<td>night club</td>
<td>painter</td>
<td>park</td>
<td></td>
</tr>
<tr>
<td>city hall</td>
<td>clothing store</td>
<td>convenience store</td>
<td>courthouse</td>
<td>dentist</td>
<td>department store</td>
<td>doctor</td>
<td>electrician</td>
<td>electronics store</td>
<td>embassy</td>
<td>establishment (deprecated)</td>
<td>finance (deprecated)</td>
<td>fire station</td>
<td>florist</td>
<td>food (deprecated)</td>
<td>funeral home</td>
<td>furniture store</td>
<td>gas station</td>
<td>general_contractor (deprecated)</td>
<td>grocery_or_supermarket</td>
</tr>
<tr>
<td>physiotherapist</td>
<td>place_of_worship (deprecated)</td>
<td>plumber</td>
<td>police</td>
<td>post office</td>
<td>real_estate_agency</td>
<td>restaurant</td>
<td>roofing_contractor</td>
<td>rv_park</td>
<td>school</td>
<td>shoe store</td>
<td>shopping_mall</td>
<td>spa</td>
<td>stadium</td>
<td>storage</td>
<td>store</td>
<td>subway_station</td>
<td>synagogue</td>
<td>taxi_stand</td>
<td>train_station</td>
</tr>
</tbody>
</table>
Google Places API Overview

- **Use Place picker UI**: allows users to select place from “possible place” on a map.

- **Get current place**: place where device is last known to be located.
  - Returns list of likely places + likelihood device is in that place.
Google Places API Overview

- **Autocomplete**: queries the location database as users type, suggests nearby places matching letters typed in
Learning Google Places API

- Official Google Places website is “decent”, up to date:
  - https://developers.google.com/places/

- Two great references:
  a) Getting started with Google Places API
     https://developers.google.com/places/android-api/start
  b) Tutorial by Paul Trebilcox-Ruiz may be more readable:
     http://code.tutsplus.com/articles/google-play-services-using-the-places-api--cms-23715
Other Useful Google Maps/Location APIs
Other Maps/Useful Location APIs

- **Maps Directions API**: calculates directions between locations (walking, driving) as well as public transport directions
- **Distance Matrix API**: Calculate travel time and distance for multiple destinations
- **Elevation API**: Query locations on earth for elevation information, calculate elevation changes along routes
Other Useful Maps/Location APIs

- **Roads API:**
  - Snaps set of GPS coordinates to road user was likely travelling on (best fit)
  - Returns posted speed limits for any road segment (premium plan)

- **Time Zone API:** request time zone for location on earth
GeoFencing
https://developer.android.com/training/location/geofencing.html

- **Geofence**: Sends alerts when user is within a certain radius to a location of interest

- Can be configured to send:
  - **ENTER** event when user enters circle
  - **EXIT** event when user exits circle

- Can also specify a duration or **Dwell** user must be in circle before triggering event
GeoFencing
https://developer.android.com/training/location/geofencing.html

- Great reference:
  - How to work with GeoFences on Android by Tin Megali
    https://code.tutsplus.com/tutorials/how-to-work-with-geofences-on-android--cms-26639
Some Interesting Location-Aware Apps
MileIQ

- **The Problem:** Mileage tracking is useful but a burden.
  - IRS deductions on taxes
  - Some companies reimburse employees for mileage,
- Passively, automatically tracks business mileage, IRS compliant
- Swipe right after drive to indicate it was a business trip
- Project idea? Implement some of this functionality

- How Android modules? For what?
- What stats to decide if this is tackling important problem?
**Trigger**

- Use geofences, NFC, bluetooth, WiFi connections, etc to set auto-behaviors
  - Battery low -> turn off bluetooth + auto sync
  - Silence phone every morning when you get to work
  - Turn off mobile data when you connect to your home WiFi
  - Silence phone and set alarm once I get into bed
  - Use geofence for automatic foursquare checkin
  - Launch maps when you connect to your car’s bluetooth network

- Project idea? Implement subset of these features

- What triggers would be useful for a WPI student?
References

- John Corpuz, 10 Best Location Aware Apps
- Liane Cassavoy, 21 Awesome GPS and Location-Aware Apps for Android,
- Head First Android
- Android Nerd Ranch, 2nd edition
- Busy Coder’s guide to Android version 6.3
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014