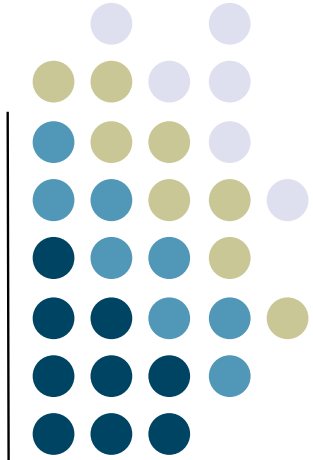
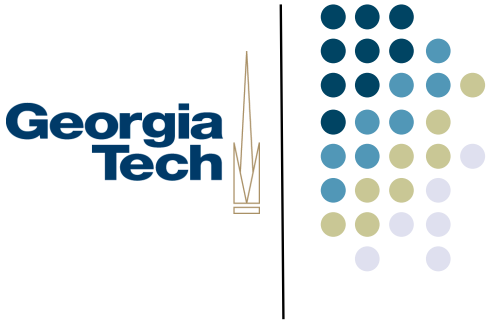
Placelab



Georgia Tech



What's Placelab?

- System for estimating the position of client devices
- *Privacy-preserving:*
  - Clients figure out their location from the infrastructure...
  - ... rather than the infrastructure figuring out the location of clients
- Basic idea: estimate position based on known (or estimated) location of fixed radio beacons
  - WiFi access points: locations change relatively infrequently, often located by wardrivers
  - GSM towers: locations change *very* infrequently
  - We'll be using WiFi
- Java-based implementation (with source) freely available
  - www.placelab.org

Developing Using Placelab (1)

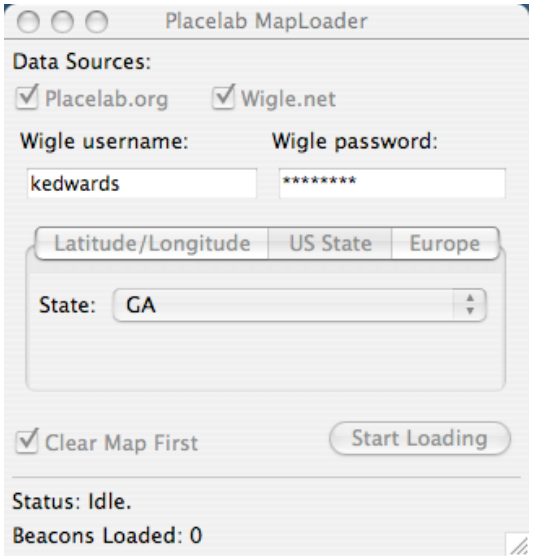
- Download from www.placelab.org
- Verify your installation
  - Open a command line shell
  - CD to the run directory
  - Try running WiFiSpotter and APViewer (APViewer won't actually show your position)
- This will verify that you've downloaded everything correctly
- NOTE that APViewer won't show your location yet...

Developing Using Placelab (2)

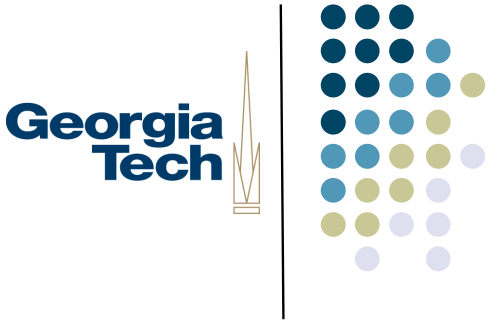
- For Placelab to actually figure out your position, you need to download a snapshot of radio beacon locations
  - Multiple freely-available databases of WiFi access point (AP) locations
    - Smaller database maintained at placelab.org
    - Much larger database at wigle.net (create a free account there to access)
  - Run MapLoaderGUI. This is the tool that will download a map database
  - Click placelab.org and/or wigle.net (if you've created a wigle account)
  - Click on the US State tab and select GA
  - Click Start Loading...
    - Will load AP locations from the databases you've selected
    - If just using the Placelab database, very quick
    - If using wigle.net, may take a long time depending on network speed.
- Only need to do this once

Developing Using Placelab (3)

- Once you've downloaded AP locations you're good to go
- NOTE: Placelab uses a database internally to store AP locations
  - If you have to kill a process using the database, you may leave it in a "locked" state meaning other processes can't use it.
  - If you need to reset things, go to the placelabdata directory
    - Delete hsqlmap.lck to remove the lock
    - You can delete the database files if necessary (but you'll have to reload the map data)
- Now, run APViewer again. It should show your location in latitude/longitude cordinates

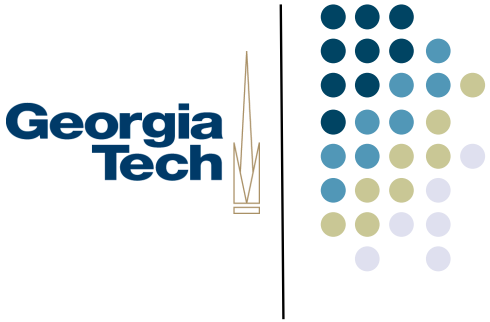


Using Placelab from Jython



- There are two things you must know in order to integrate Placelab into your Jython code
  - How to use the Placelab Java APIs from Jython
  - How to tweak Jython slightly so that it finds various parameters and files that Placelab needs in order to run.
- The following slides show how to do both of these

Using the Placelab APIs from Jython



- Only two classes you really need to know:
  - PlacelabWithProxy -- an "all in one" Java class for starting Placelab and getting estimates from it
  - EstimateListener -- a Java interface that defines the methods that will be invoked when Placelab updates its estimated location
- Both are easy to use from Jython

A Placelab/Jython example

```
import org.placelab.client as client
import org.placelab.client.tracker as tracker

class Listener(tracker.EstimateListener):
    def estimateUpdated(self, tracker, estimate, measurement):
        print "Estimated position is " + str(estimate.getCoord())

if __name__ == "__main__":
    placelab = client.PlacelabWithProxy()
    placelab.addEstimateListener(Listener())
    placelab.pulse()
```

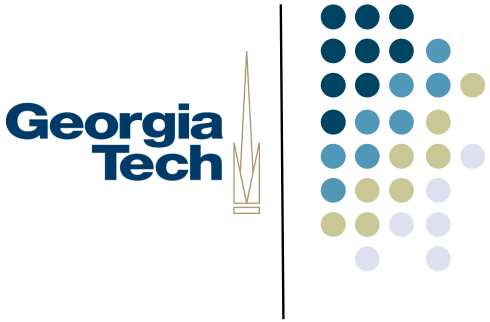
Import necessary packages

Define a new class with a method to be called when the location is updated

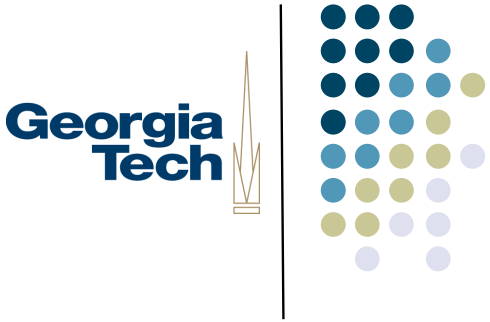
estimateUpdated is called whenever the location changed

Create a new PlacelabWithProxy instance, add the listener, and call pulse() to have it send updates every couple of seconds. That's it!

Configuring Jython to work with Placelab

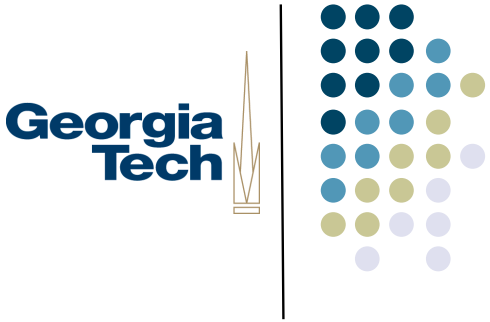


- Setting CLASSPATH
  - In order for Jython to "see" the Placelab code, it must be in your CLASSPATH environment variable
  - Under the Placelab installation directory, in lib:
    - placelab.jar, jdbm-0.12.jar, mysql.jar, hsqldb.jar (provide full paths to these)
- Configure placelab.ini
  - Lives in placelabdata
  - Specifies the path to placelabdata
  - Edit this to provide an *absolute* path to placelabdata
- Extra command line arguments
  - Specify where the *native code* needed for Placelab lives (interacts with WiFi hardware)
  - Specify where placelab.ini lives:
    - jython -Djava.library.path=/path/to/native -Dplacelab.ini=/path/to/placelab.ini yourcode.py



Gotchas

- All of the little scripts and configuration files that Placelab uses use *relative* paths to files (e.g., ../lib/placelab.jar)
- This means they won't work if you run them from any directory other than the Placelab installation directory
- Edit the files to fix this problem--you'll save yourself hassle:
  - placelabdata/placelab.ini
  - run/placelab.env
  - run/* (all of the scripts that run the demos)



Pragmatics

- *You may not be able to get this to work correctly!*
  - Example: some problems with WiFi hardware on certain Dell machines
  - Example: WiFi drivers unsupported by Placelab
  - Example: can't find any known APs around where you debug
- Therefore, I'll be grading for properly *integrating and using* the Placelab code from Jython, not whether the code actually *works* to find the location
- Build yourself a "safety net" in case of bugs/problems
  - Should allow *either* manually entering lat/long coords, or select from a list of known places (CoC, TSRB, elsewhere)
- Don't spend a lot of time fighting Placelab. Get set up to the point you can call it from your code and then move on from there.