HITMAN: ANDROID

- Pete Vilter <vilterp@uchicago.edu>
- Kevin Zhang <kevin@kevinzhang.org>
Team

Kevin
- client-server development
- real-time architectures
- Python/Django, JS/Node, Java

Pete
- also client-server development
- Java experience, some Android experience
Problem / Opportunity

- Assassins is a fun & popular real-world RPG
- Versions of it as a mobile app have been done before
  - Gameplay mechanics based on GPS & camera (photo of target) used for kill
- Potential for more sensors to be used and in conjunction with each other
- Assassins as a mobile app allows for a fusion of real-world + computer game play mechanics
Idea: Innovative gameplay mechanics using multiple sensors in tandem

- **hints** based on velocity, location data, and reverse geocoding API
  - e.g. "target just walked past <some building>"

- **picture-taking** mechanic
  - for each target $X$, players who are not that target's assassin are notified when $X$ is nearby, and encouraged to take pictures of people around them
  - These pictures are sent to the assassin. They may or may not be of the target — the detective work is up to the assassin!

- **kill** mechanic
  - when in proximity, assassin challenges target to "fight to the death"
  - some kind of minigame, possibly based on low-latency device-to-device communications
Background

• Kaos (previous class project)
  ○ 'ping' mechanic that allows assassins to obtain their target's compass direction
  ○ novel picture-taking mechanism for 'kill' mechanic with 3rd-party verification

• assassins.appspot.com (vaporware)
  ○ each assassin has a 'deadly range' based on GPS location
  ○ in-game currency system

• "Making friends by killing them: using location-based urban gaming to expand personal networks", Coe and Chen, 2010
Made Possible By...

- **Android thin client**
  - background Service sends location data to server periodically
  - BroadcastReceiver gets push notifications from server using Google Cloud Messaging API

- **Server**
  - Real-time, high availability, high-concurrency, exa-scale, non-blocking async I/O architecture in the cloud
  - Actually, it's just a desktop
  - Django + PostGIS backend
  - Keeps track of game state
  - Analyze and aggregate sensor data so that relevant notifications/events/warnings can be sent to targets and assassins

- **Sensor accuracy, connection bandwidth/latency/reliability are not critical to the app as we make use of aggregation; a constant stream of data is not required**