This Week's Progress

We met with Garrett Genereux, climbing coordinator, who manages the Crags Climbing Center at UW and proposed our project idea and needs. Garrett was very open and agreed to give us off-hours access to the climbing center in order to conduct our research and project. We also proposed having an open "CV/AR climbing night" towards the end of the quarter where people could come in and try out our game/project. This will allow us to have uninterrupted development time when the gym is closed to gather images, videos, and test our program.

We reserved a projector from the University of Washington Student Technology Fund (STF) to borrow for the duration of the project. We also acquired a Logitech webcam to use as our image/video source for the project. The camera records 640x480 resolution, which if it turns out to be insufficient quality, we have a Mac laptop which has a built in webcam recording at 1280x720 resolution. We will experiment with both and see which will produce high quality results / data for us to use.

Outline / Milestones

We will start by building an image processing API to support our setup and provide the foundations for developing games.

Rock Wall Image Processing API

- Obtain image samples of walls (simple camera photographs: maybe some with webcams / laptop cameras)
- Compute location of handles accurately and reliably. Ideas:
  - Sum of gradient magnitudes
  - Distribution of gradient magnitudes
  - Grey (wall) subtraction
  - Open CV Object Detection
- Detect Human (Histogram of Gradient human detector)
- Pose Detection
  - Many possible avenues, but some implementations use neural network pose estimation and HoG descriptors regressing to some constraints
  - Example Code
- Display images on the rock wall with little to no distortion
  - Compute incline of wall and transform image accordingly
  - Project a dot-grid on the wall and compute the relative depth (warping) via dot pixel size, apply inverse to image being projected. Or use a depth camera.
- Higher level APIs like: getHandlePositions(), didHumanTouchHandle(), didTouchXY(), trackHandleProgress()