Meeting todo

Project: Crazyflie application to allow for better control and visibility using the Crazyflie as well as increased functionality over existing applications.

The base of our project will involve controlling the Crazyflie micro-drone with an Android device and having the Android device also be able to view the video transmitted from a camera onboard the Crazyflie micro-drone. The specifics are as follows:

- An Android app will be built to control the Crazyflie and view the video.
- The Crazyflie will mimic the movements of the Android phone. The standard orientation of the phone will be horizontal. Tilting the phone in any of the 4 standard directions (forward/back and left/right) will cause the Crazyflie to mimic these actions (it will move forward/back and left/right).
- The video will be coming from a 1 gram PAL camera which is onboard the Crazyflie.
- The Android app built will display this video.

There will be room for many additional features if the above is completed well before the deadline. These additional features include:

- Augmenting the app to have a Heads Up Display. This display could overlay the video feed. It could have real time information like battery level, speed, etc.
- Overlaying a feed from OpenCV to allow for object tracking in real time in the app on the phone
- Adding additional functionality to the app, for example a “hover control” button which when pressed keeps the altitude of the Crazyflie constant
- Transmitting voice commands that the Crazyflie responds to

To ensure equal work done by all parties, the components and their main supporters are listed below. It is assumed that the project will not be “compartmentalized” such that each member is working only in their area, however each “main supporter” will become the “group expert” of the area as the semester progresses. There are basically three areas of our project: how the user interacts with the phone and the format of the information available to them from the phone, getting sensor data from the phone and pushing it to the Crazyflie, having the Crazyflie utilize that data.

These three areas and their main supporters are listed below.

- The Android app GUI, HUD, UX, video display ==> Matt
- The Android control functionality and primary Android sensor data support==> Manomit
- The interaction between the app and the Crazyflie and primary Crazyflie sensor data and utilization support ==> Bob
We also acknowledge that previous work has been done in this area and below we detail the applicable previous work and the differences we present:

- Crazyflie Android App ([https://github.com/bitcraze/crazyflie-android-client](https://github.com/bitcraze/crazyflie-android-client)). This controls the Crazyflie using the gyroscope and the accelerometer on the phone. This app is still difficult to use and does not allow for precise control of the craft. Our application will be built from the group up using little of their code to create an app that is better at controlling the craft than the existing app.

- Crazyflie Realtime HUD ([https://www.youtube.com/watch?v=AWSUMGJKt0U](https://www.youtube.com/watch?v=AWSUMGJKt0U)). This uses C++ and ROS to put a HUD on a video feed from the Crazyflie. Ours will not use C++ or ROS and will stream the raw video in the Android app.