Team 22 Project: Virtual Whiteboard (Version 1)

**Hardware:**

1. LED pen (build on our own) CLIENT END
2. Android phone (for image capturing and some image processing) CLIENT END
3. BeagleBoard-xM (for combining clients’ information) SERVER END\*
4. Projector/screen (display output)

\*There are other boards that can be the server board as well. The only requirements for the board are that it supports USB/Wi-Fi and has video output.

**Software:**

1. Android development (Phones/Clients)
2. Ubuntu/C (BeagleBoard-xM)
3. USB Driver (version 1)
4. Wi-Fi Driver (version 2)

**Protocols:**

1. USB (localized version, version 1)
2. Wi-Fi (advanced version, version 2)
3. VGA/DVI video output (from BeagleBoard to projector)

**Overall Architecture:**

The basic architecture is like server-client architecture which each phone and its LED pen works as a client end and the board is the server end. When the phone launches the program on the phone, it will start generating continuous video information in a fixed rate (50-60Hz). The user can just press the button on the pen to activate the LED and start writing in front of the phone’s camera. Based on the video information captured by the phone, a program should do some image processing to render out the “points list” that is created by the movement of LED. Then feed this list to the server board. The server board should be updating the output at a fixed rate as well (50-60 Hz). Upon receiving any “points list” update from one client, the board should update the output information based on the “points list” and then output it through its video output.

Therefore, we have a shared board. Multiple users can write in the air using LED pen and have their message displayed on the same board.

**Detailed implementation:**

1. LED pen:
   1. The LED pen should be a simple LED circuit which the light is on by pressing the button.
2. Android phone:
   1. The phone should have a class that is capturing image at a fix rate (50-60 Hz) and updating an image buffer. This program should only use fixed size of memory for it only needs that part of memory to store image information.
   2. Another class that takes in the image buffer as input and return “points list” as output. This program should pull the data from image buffer once the buffer is updated (or at a slower rate). Then it should rendering out the trace of LED light and packs them into an output format. (points list)
   3. Another class that send the packed data through USB (version 1) or Wi-Fi (version 2).
3. BeagleBoard:
   1. The board should have a USB or Wi-Fi interface that can accept data from multiple clients and unpack the data back to points list.
   2. The board should have a separate program that takes in all the “points lists” and combine them into one output image.
   3. Output the image through video output at a fixed rate.
4. Projector:
   1. Receive the video output information and output.

**Possible Upgrades (ambitions):**

From Number of Users Stand Point:

1. The version 1 uses USB as connection between client and server, which the board can only support limited number of clients. So we can use Wi-Fi to overcome this problem and our board is able to support more clients. But this virtual board is still limited in a room.
2. If 1 is working, we can establish links between multiple “server ends” to create a “virtual conference room” where multiple virtual boards can be created and shared.

From Efficiency and Effectiveness Stand Point:

1. Integrate RF-ID with LED pens and phones, so that one phone can only track one LED pen and will not get interference from other LED pens.

From User Experience Stand Point:

1. Create a “virtual eraser” that works with LED pens so that users can erase what they wrote without resetting their phone’s point list. May be another device equipped with set of LEDs and accelerometer, which looks like a square. When user wave the 'eraser' it should change accelerometer readings and then trigger LEDS.
2. Make the LED pen be able to shine in different colored LED so that people can write in different colors or users can select the color from the phone.

From Virtual Conference Stand Point:

1. Users should also be able to upload files from their android phones and share across other boards, or display on this board.

**Risks:**

1. The communication between phone and board may fail, which then the server-client communication won’t be able to establish.
2. The lag between when user write to when the strokes are displayed may be too long and make the experience not good.
3. The phone tracker may not be able to pick up LED information correctly.
4. Concurrency problem when combining multiple clients’ information into one output.