Agenda

*Date: 05/04/13*

*Location: Calit 2, Second Floor*

*Actual Time: 12:30pm - 2pm*

*Planned Length: 1 hour 30 min*

***Please add and comment on anything and everything in the agenda!***

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member(s) | Topic | Time | Goals |
| ~~Sorin~~ | ~~Announcements~~ | ~~5 min~~ | ~~All team members are updated on new developments~~ |
| ~~Dorothy and Shannon~~ | ~~Subteam Updates~~ | ~~10 min~~ | ~~The software team is updated on subteam progress, issues, and next steps.~~ |
| ~~Backend~~ | ~~Software Architecture Discussion~~ | ~~20 min~~ | ~~The suggested OOP structure is reviewed. The OOP structure contains places for all features/user stories~~ |
| Backend | Algorithm Clarification(algorithms doc on github) | 25 min | Discuss and clarify strabismus, astigmatism, and cataracts feature extraction strategies. |
| Backend | Quarter Planning(Quarter plan doc on github) | 10 min | Plan the next steps for the quarter; assign weekly deliverables |
| Everyone | Other topics | ? | Could include change of strategy (if any) because we don’t have internet; feedback we want customer input on; strategies to get a less accurate but workable deliverable out. |
| Sorin | Recap and Next steps | 5 min. | Meeting progress/ goal accomplishment is recapped. Next steps are outlined so that every team member goes into next week knowing what they are expected to contribute |

**Time total:** 5 + 10 + 20 + 25 + 10 + ? + 5 + 15 (chat) = **1 hour 30 min**

# Summary

# Notes

## Front End

Framework - bootstrap or django?

## Backend

Pair programming, schedule?

# TODO

## Backend

* Go over testing log to fix errors - Kevin and Shannon
* get software architecture up and working - JT and Shannon
* developer’s guide/wiki - Paul
* UML diagram - Kevin/Sorin/?
* fix nostril detection - Kevin
* refactor JT’s code - JT
* draft pupil detection - David and

Secondary Eye Detection Resource Pseudo:

max\_dist = 0  
maxL = 20  
maxR = 0  
  
lc =0  
rc =0  
  
maxLP =(0,0)  
maxRP =(0,0)  
  
for point in cornerMem:  
 center = int(point[0]), int(point[1])  
  
 x = point[0]  
 y = point[1]  
  
  
 if ( x<colorImage.width/5 or x>((colorImage.width/4)\*3) ) and (y>40 and y<70):  
 #cv.Circle(image,(x,y),2,cv.RGB(155, 0, 25))  
  
 if maxL > x:  
 maxL = x  
 maxLP = center  
  
  
 if maxR < x:  
 maxR = x  
 maxRP = center  
  
 dist = maxR-maxL  
  
 if max\_dist<dist:  
 max\_dist = maxR-maxL  
 lc = maxLP  
 rc = maxRP  
  
  
  
  
  
 cv.Circle(colorImage, (center), 1, (200,100,255)) #for every corner  
  
cv.Circle(colorImage,maxLP,3,cv.RGB(0, 255, 0)) # for left eye corner  
cv.Circle(colorImage,maxRP,3,cv.RGB(0,255,0)) # for right eye corner