**What went well during this Sprint?**

**We got the emulator working and were able to focus on working on the main pages in the actual app.**   
**Another thing that went well this week was that we were able to establish an information stream between the client and the server, and then able to broadcast that information to all players in the game**  
**We got the windows machines working so now we can test/debug our code without having to launch the emulator, which makes it much quicker as well as taking up a lot less battery on our computers.**

**We got a functioning screen changing button as well as a working text box, that can be used through the UI.**

**We also were able to implement the key arrow feature that points in the direction of the player that they are attempting to hunt currently, this being an integral feature of the game and one of the more complicated and server communication heavy aspect of the project.**

**What problems were encountered?**  
**We encountered a couple of problems while working during this sprint as to be expected. One of the problems we found when was implementing the NextPage method we had done independently in a non OppSwap program and we ended up taking a lot of time to fully transfer the code properly and each individually download the MVVM toolkit that was required for the functionality of the button.**

**We encountered a problem with Compass Implementation near the end of the sprint. This was an issue dealing with incompatibilities between the compasses of different devices. But our mentor told us that we knew that there was a solution to this problem (I.e. we know that mobile devices have a compass, that they even use for their own compass app), we could initially develop a solution with a method that returns “simulated” compass headings, so that we could finish other parts of the project, without being delayed by the fact that we don’t yet fully understand how to collect the compass data from a mobile device.**

**We also encountered issues with uploading our application to our mobile devices. When trying to push new versions of the code to an iPhone, we realized that there wa an error with the provisioning profile (the documents that tell an iPhone that the app being installed is not a virus). Again, with similar feedback, we already knew that there was a solution, that could be easily found after the launch of the MVP. So, we decided to just continue working on the application with the mobile emulators we had on our computers.**

**On problem that we encountered with server-side communication was that there was an error in the timing of the initial connection request. Sometimes, we would send a request to connect, and without waiting for our ClientID, we sent another message without this identification, which cause the server to crash**

**Were these problems solved? If so, how, if not, why?**

**We were able to fix the NextPage method by fixing an error that we had overlooked in the Appshell.xaml.cs document where we had forgotten to code a route to both the page and the view model that we were attempting to change the page to.**

**We did not solve the Compass Implementation problem because we were experiencing common and persistent issues with it because our laptops could not use the compass that we were using. Also, this was not adding value to our MVP because we can use test data to simulate the heading that we would receive.**

**We did not solve the issues with uploading our applications to mobile devices because we were experiencing issues near the end of the sprint, so we determined (after receiving advice from our mentor) that the uploading to mobile devices is not crucial to our MVP.**

**What are the most helpful changes you can make to improve your effectiveness as a Team in the next Sprint?**

**The most helpful change that I think we can is creating two separate views, one for the UI testing and one for the server testing. Then merge those together in main. This would save a lot of time and hassle with issues that arise when the server is not running, so the code would crash. That would be the thing Sravan implements when he is scrum master. I think also reprioritizing our tasks is important and implementing that in our sprint backlog by having a hierarchy system in our project board.**

**Another change that we could make is to use more fake data when testing to prevent loss of time because of blockages between multiple developers. We should also begin to use the funretrospectives website that we were introduced to by the Project Mangement Presentation to track issues we are experiencing.**