Completed Goals:

- Update UI
- Understand Logic versus UI of tank and bullets for UML and Implementation

Meeting 5: 3Hrs: both team members were present. We met with Karen and went over things we needed to accomplish for the first milestone. Afterwards we worked on using an observable and observer pattern. We updated our poller to be sure that it only polls and does not need a grid.

Meeting 6: 5Hrs: both team members were present. We implemented a portrait screen orientation to resolve crashing when it would try to rotate. Cleaned up the image of our game a bit by changing the size of the buttons. Implemented a shake listener for the shake to fire. We completed the constraint checking. We made a UIUpdate class as well as a GridUpdater. This helped to make sure the UIUpdate was no longer a god class. It allows for the GridUpdater to solely update the grid while the UIUpdate solely updates the UI.

We also redid our UML diagram. This now looks much better with cleaner and smaller paths due to no more god classes, loose coupling and high cohesion.

We implemented the observer and observable patterns after we tried to get the Event Bus to work but could not. We spoke to Karen and she stated that observer and observable were okay to use instead.

Meeting 7: 4 Hrs: both team members were present. We worked to implement a bullet observer class to be sure that our tank can only have two bullets on the screen at one time. We implemented enums in our code to be clearer when doing the constraints and the integers for moving. We added an opening screen to start the game. We implemented JUnit testing for move, turn and fire. We then went through each of our classes and added javadoc comments for each necessary method as well as the overall class. We left out the javadocs for the getter and setter methods due to their low complexity. We updated the UML to be sure it followed our code.

Upcoming Goals:

- *Fix wait for move – ask Karen about this*
- End game handling
- Add Rejoin
- Center game/make pretty