**Joey Tong**

**Project 05 Milestone 3 (FINAL)**

**World:**

Some scattered islands in the ocean. The idea of the simulation is to allow the player to play as the captain of a ship in first person AS WELL as command a squad of ships in naval maneuvers.

**Steering Behaviors:**

My complex following is programmed in such a way that the ships cut some sharp corners in favor of speed, but in 99.99% of cases, they will not ignore loops, circles, and crossed paths even though they are programmed to cut corners.   
They also apply braking forces while approaching sharp turns, so in 99.99 percent of cases, they will not be pushed completely off course by momentum (unless the area is overcrowded with other ships applying separation forces).

When following the leader, ships will still maintain distance from the player AND other ships, and they follow a point behind the ships center, roughly right behind the ships rudders.

The path setting is programmed such that points cannot be greater than 750m apart (as this messes with some parts of the AI programming). Also, if the start and end points of a route are <750m apart from each other, the ships will automatically loop around the path.

Due to the way I programmed stopping distances, intelligent path finding, and steering, I don’t do the “orthogonal projection” strategy shown in class for path following, but in most cases, my AI still achieves a very similar behavior with lighter calculations.

Both AI, and player ships simulate the physical behaviors of ships in real life. When turning, they list to the opposite side to simulate centripetal forces. The ships also bob up and down and rotate about their z axis to simulate buoyancy, gravity, and waves. Gravity and buoyancy use actual force calculations. Centripetal forces use light calculations that quite accurately simulate the real thing. There are some small issues with the rotation “shaking” that I haven’t been able to solve.

**Entity interactions:**

Ships will avoid each other so that they don’t crash into each other. They will also avoid the player. When the player toggles leader mode, ships will follow the leader.

**Resources:**

None besides c# documentation. Formulae & coding strategies either already known or coded from scratch.

**Unresolved Bugs:**

**\*these are inconsequential to the actual AI but make the “ship” experience slightly diminished. I simply didn’t have time to figure out workarounds for some of these issues on top of the many I had already solved.\***

**AI ships with “pop” into their initial rotation & turn too quickly for a ship** because the way the ship rotation works is it looks in the direction of its net forces. This works fine unless the ship is moving from a complete standstill. I was trying to figure out a workaround for this, but it was more important to get the path following and leader following working first. The general strategy I was pursuing was to correct the normal vector of the direction the ship moves in after the forces are applied. I.E. find the maximum amount the ship can rotate, and then multiply the netforce magnitude by the normal of the \*correct\* direction that’s physically possible for them to move in.  
**“Shaky” rotation during extreme centripetal forces from turning** an issue I haven’t been able to find the source of.

**Unity throws unityExceptions** that don’t seem to affect the game in any way whatsoever that don’t seem to be fixable (I tried removing the part of the code that threw the error and nothing happened), so I didn’t bother resolving them.

**Player Controls:**

**WASD** for movement

**LMB** to place waypoints (in the order that they’re traveled to)

* Waypoints must be within 750 meters of each other (roughly 2.5 ship lengths)
* The beginning and end waypoint of a path must be within 750 meters of each other for the path to loop. Otherwise, ships will stop.

**RMB** to toggle “leader mode” on the player. All ships within an 800 meter radius will follow the player when this is toggled.

**R** to clear all waypoints and create a new path

**CTRL** to toggle debug lines, target lines are white, velocity lines are red

**Space** to switch between fps and aerial cameras

**Asset Resources:**

**IJN Yamato:** <https://www.yobi3d.com/#!/search?q=free%25203d%2520model%2520yamato>

**USS New Jersey:** [**http://pan.baidu.com/s/1o6MBDb0**](http://pan.baidu.com/s/1o6MBDb0)