In this memo, we provide a description of the software system we plan to build for the Fish game. We plan on including three separate components. The first is the game server, which is responsible for managing a single game. This component will include the model and controller for a given game; it will communicate directly with player AIs, and with the game client and tournament server. Next, the game client is the view layer for the game, and allows players to view how their AI is faring in a given game. It receives messages directly from the game server, telling it what to render. Finally, the tournament server will organize the tournament itself, setting up game servers and communicating with them to determine winners, who will move onto new game servers.

Next, we'll further describe each component in detail.

Game server:

- Accepts client messages, validates them, updates the gamestate accordingly, and sends the clients the new game state.
- Only communicates with the 2-4 clients specified on startup. This will be given to the game server when the tournament process spawns it or can be given locally to play the game locally.
- When the game ends, the server will send the clients the final updated gamestate with the winning player(s).

Game client:

- Accepts gamestate messages from the server showing the current game state. These may be rendered by a gui to show the game to human players in a more appealing form. This is the View in the traditional MVC architecture.
- Can send the server a move or interaction which the server validates and uses to update the gamestate accordingly.

Tournament server:

- Manages tournaments, moving through three successive stages:
    1. Handle player-sign up and data collection (age, entry fee, etc.). Waits for all players to sign-up before starting the tournament.
    2. Starts the tournament, scheduling each game on the game server with the participating players. At the end of each game, the game server should inform the tournament system of the player(s) that weren't disqualified. Continues scheduling games until a winner is decided for the tournament.
    3. Stops the tournament and distributes prize money to the winner.