Fish Dot Game Planning:

Jason Alexander and Vlad Satchek

The most basic software component we need is a representation of the pieces of the game. This can be broken down into sub-components which are the hexagonal tiles which contain a certain amount of fish and a list of other tile neighbors and the penguin pieces which rest on certain tiles and move to other tiles. A collection of tiles (with a certain dimension) and player pieces will make up a board. So the board state will be an integral part of our design because it will allow games themselves to have a representation.

The next software component is the “plug-in” protocol for the player, which we will refer to as a software interface for the AI players of Fish. This interface will interact with the representation of the pieces of the game to provide endpoints to the player around the actions needed to complete a Fish game. These actions most broadly construed are initializing a game, completing a turn, and responding to an end game state (counting fish, etc.). This interface would need to have knowledge of the representation of the pieces of the game to allow a player to interact with that representation of a game to carry out the actions above. After this interface is implanted, we should also as a sub-component of that implement at least one “house” player AI to be able to test our system and be of possible use in future tournaments.

We would also have a component of a runner of games or more simply a “referee”. This component would make sure that the AI players are following the AI player interface (not cheating and malfunctioning) as well as be able to move the game along (removing tiles in the beginning, assigning penguin color, running placement rounds, moving to next turn, etc.). This component would have to have knowledge of component for the board representation as well as have knowledge of the interface for the AI player so that it can communicate with the player over that interface and understand the board state to know legal moves.

Outside of a single game of Fish, we would need to develop a component to manage tournaments of Fish. This component would allow players to sign-up for the tournament (which includes specifying a time period for sign-ups), create matchups (which will include AI players that have implemented the player interface) and announcing winners and losers of the tournament. This component should not know about the internal state of individual games, but instead should just know about the winners/losers of players in the tournament and use this information to match players and referees to start a new game. The sign-up and winner/loser information will be enough to complete and tournament and award players.

Finally, the players (or other stakeholders) would want to see how their player is doing. For this there will be a component that allows a player to easily see a game’s current state and the overall state of a tournament. So, this component would need knowledge about a game representation, as well as allow the player to access that representation in the player interface. For tournament visualization, it would need knowledge about wins and players that the tournament runner component would have.