Model

FishModel: The FishModel component will represent the current state of the game. The model manages the data and keeps track of the rules, constraints and logic for the game to keep running. In the model, it should keep track of the board, the list of players who have signed up for the game, whose turn it is right now, the current state of the game (either in the midst of playing or has ended) and if there is a winner.
Player: This component will represent a single player that has signed up for the game. It will keep track of the personal information of the player such as the age which will be used to determine whose turn it is. It will also keep track of the score the player has accumulated throughout the game. Other than that, it will keep track of the color and a list of penguins it is assigned to, used as the player's representation in the game. This component will be linked to the FishModel as a field represented in the form of a list of players.
Penguin: This component is used to represent a single penguin. It will be linked to a Player as a field represented in the form of a list of penguins. It will mainly keep track of its color (represents which player is the owner of this penguin during the visualization of the game) and the tile it is currently on.
Tile: This component is used to represent a single tile on the board. It is mainly used by the penguin to determine its placement on the board. This component will also keep track of the number of fish it has which will become 0 once a penguin moves out of it while also keeping track of a list of possible neighboring tiles a penguin could move to from this tile.

Controller

The FishGameController component is used to respond to the player's input/movements. It will perform the interactions appropriately by modifying the model and view such as, updating the state of the board and penguin when the player moves its penguin or it will skip the player if the model determines that the player has no penguins to move. It will then command the view to redraw to represent the new state of the game. The controller is also the "middle person" between the model and view as it will command the view to make appropriate changes to the visual representation of the game by giving it information from the model. Thus, it is connected to the model and view by having them as fields.

View

Graphics View (Frame): The GraphicsView component will represent the renderings of the data in the model or visual representation of the game. This component provides the visual platform that allows the players to interact where players can input their command such as to move their penguins etc. When a player movement is detected by the controller, the view gets the relevant information from the model through the controller to draw the updated state of the game. This component should not have any way to manipulate the game state or model itself. The graphics view will keep track of the panel which is the component where the visual renderings are actually drawn on.
Panel: The Panel component will be where the game is drawn on. It will draw the essential components like penguin, board, tile, fish, scoreboard etc and will redraw once the game state has changed.

Server

The server is the platform that allows players to connect to the game through their 'AI' players. The game server will allow the players to sign up for their chosen game during the sign-up period and run the tournament once the period is over. It will establish the protocol that needs to be adhered to and check if the clients adhere to those protocols in order to sign up for a game.