

Stage 1–Basic Rendering: We will start simple: all we need for the first milestone is to *render* the UI, with no functionality just yet. We will need to render the page that hackers will sign their players up on and will render the Fish game board in a new window if a “Play Fish” button on that same sign up page is clicked. Rendering obviously requires the presentation component, but we will also need the data component in order to store information about the game board we display (how many hexagons, etc). The logic component will be utilized in order to move from displaying the sign up page to Fish game board.

Stage 2–Get Fish Game Working for 4 players (max): When a “Play Fish” button on the sign up page is clicked, a window will be displayed which takes in the number of human players (at most 4) and game board dimensions. An avatar is randomly assigned for players 1...N. It is up to the humans to note which player they are. Then, the game board is rendered and the game will be played manually by the 4 players. Rendering will require the presentation component; we will also need the data component in order to store information about who the players are, the game board we display, and the game state. The logic component will be utilized in order to move from displaying the sign up page to Fish game board and coordinate the gameplay, along with determining winners, losers, or a standstill.

Stage 3–Get Fish Game Working with self-built AI players: Create 4 AI players with different strategies to Fish, and then get them to play the game instead of humans. We need to find a way to hook these players up and integrate these players’ software into the system, getting the AI software to interact with our general Fish game software, which will be handled by the logic component. The logic component will be utilized in order to move from displaying the sign up page to Fish game board and coordinate the gameplay, along with determining winners, losers, or a standstill. Rendering obviously requires the presentation component. We will also need the data component in order to store AI players’ software data, the game board, and the game state.

Stage 4–Get Fish Game Working with Hackers’ AI players (max 4): We will now need to give functionality to the signup system/termination system. The data layer will need to store information about the hackers and the AI players/their software. The logic layer will need to help hook up this software up to the data layer so that the game can be played. We also need a mechanism to determine whether the AI software is malfunctioning and put a malfunctioning AI’s hacker on our blacklist (stored in data layer). We can also demo what happens when a terminated player tries to sign up. Rendering requires the presentation component.

Stage 5–Finishing Touches, Integrate Payment Subsystem: We need to integrate into our server a good payment subsystem that allows the hackers to pay the entry fee and receive their prize money (requires the logic and data components). We also will add any finishing touches, especially making the presentation component more aesthetically pleasing, and will address any outstanding bugs that have not already been fixed. Rendering requires the presentation component. After this stage, we will go live! (and then enter into an iterative development cycle based on inevitable customer complaints)