Team: PokerAI

Product Name: PocketStrategist

Date: 07/24/2023

Release Summary

Key user stories and acceptance criteria:

Sprint 2, User Story 1: As a poker player, I want to know which action I should take with the hand provided

- Acceptance Criteria:
  - Develop a clear, yet primitive set of rules regarding which action to take in a particular circumstance(without optimization or benchmarking) .
  - Has frontend integration with the backend(by means of Axios) been implemented and tested?
  - Study tool in the frontend must be sufficiently complete and display the results of backend responses
  - Create two dropdown boxes to input the starting hands from the user given a range of {A, K, J, …, 2}when the box is clicked, and when the card is selected, it changes the color to identify if data is correctly input.

Sprint 3, User story 2: As a poker player, I want to learn the optimal action based on an imaginary situation with a poker engine and environment.

- Acceptance Criteria:
  - Benchmark optimality of recommended actions for three positions relative to a functional poker engine and based on these tests rigorously optimize rules for accuracy, over 90%.(N.B. While the verbiage of these two user stories is similar, they are differentiated by their respective acceptance criteria - namely, that the Sprint 2, User Story 1 requires primitive rules and optimal actions, while the acceptance criteria for Sprint 3, User Story 2 requires rigorous testing with at least 90% accuracy on all three positions).

Sprint3, User Story 1: As a poker player, especially as a beginner, I want to have a user-friendly and aesthetically pleasing UI that helps me understand more clearly and intuitively on an optimal action and keeps me motivated to learn.

- Acceptance criteria:
  - Decrease the possibility that the user mistakenly inputs their information by improving UI
    - When the two same number cards are selected, it disables the “suited” button.
    - When multiple “position” buttons are selected, discard the old one and update only the most recent-selected button.
    - Until all the conditions to be able to go to the result page are satisfied, it disables the “VIEW PREFLOP” button.
  - Display an intuitive result page with simple result page and face expression icons depending on the optimal result.
    - When the optimal action is RAISE, then it displays the simple text message and icon which is associated with the accurate mood
    - When the optimal action is FOLD, then it displays the the simple text message and icon which is associated with the accurate mood
  - Introducing a consistent color theme throughout the whole pages that is associated with poker and casino themes.

Known Bugs and Problems:

- A few hands do not output the optimal action that they should in comparison to GTOWizard’s recommended action.
- Potential unforeseen cases on the backend side that could broadcast incorrect server responses to the frontend.
- Despite the presence of “explanation” return values through the backend code, this is currently not implemented in the frontend.
- After different number of the starring cards are selected, and the same number cards are reselected, it disables the “suited” button, however, the check mark remains

- Once one card is selected but not the other one, then it clicks “suited” button, then the other card is clicked, then clicks the “VIEW PREFLOP” button, it returns 400 error message.
- Dropdown box works fine on the web, but on IOS it goes upwards instead of downwards for unknown reasons.

Product Backlog:

As regards to a product backlog, these would be the high priority user stories for a potential follow-on project.

- As a player, I want the Poker AI Engine to make accurate decisions based on pot odds logic, so that I can play against a challenging and intelligent opponent.
- As a player, I want to see a chart of possible hands to play at each turn, along with their respective probability values, so I can make strategic decisions based on the statistics provided.
- As a player, I would like the program to replace the existing logic to instead use a Counterfactual Regret Minimization (CFR) based algorithm, to better improve the decision making capabilities of the program.

Bug Fixes

- Backend side would need to further identify cases of bugs that would lead to cases of failure on both frontend and backend sides.
- Backend side needs to check for bugs in the hand-strength calculation to ensure the right action is taken for certain corner cases.
- With more time, further corner cases and failure modes in both the frontend and backend need to be identified and addressed - the latter entails an exhaustive battery of unit tests exercising corner cases in the hand strength calculator and optimal hand recommender, while the former requires extensive user testing and feedback.