Prototype 4:

I am trying to successfully make a user interface for my chess game. I have 2 main choices. I could abandon my initial ideas and make a desktop app or I could try to make use of my existing vue.js components.

I am going to look into the idea of using VUE js. This creates an issue as VUE js is a single page where as flask is more compatible for multiple pages. My solution is to make the website entirely with vue.js with a single webpage with many components. I will then use flask to create a WebSocket server.

To do this is am going to restart with a new folder.

Update. The date is the 7th of February 2023. I am aiming to have finished these items over half term:

* Design part 2
* Prototype 3
* All testing (maybe more)

Currently I have absolutely finished my chess engine. It is fully working and I intend to discuss the extensive algorithm and its use of data structures, OOP and databases in design part 2.

With regards to prototype 3 I have decided to tackle it with a different strategy. I am going to use my VUE js code. The idea is to have a self-contained VUE app with a config page which passes the data (difficulty etc.) to the game play page with the Vue router. This replaces the original idea to send this data to the server to be added to a session before redirecting to the game play page.

I will store the legal moves for the stating position in the JavaScript directly. Whenever a move is made, the legal moves will be checked. Once a valid move is made, a single request will be sent to the server. The server will respond with the following data. This may require multiple requests or a different format etc.

Assuming the game is not over after the users move and that it isn’t over after the computer’s move:

* The computer’s move and the legal moves that follow.

If over after the user’s move then this must be identified.

If over after the computer’s move then this must be identified.

Due to my use of web sockets, multiple requests are possible and this may be best considering the data that must be conveyed.

I currently have the ­­­­­­implemented a chess board in the VUE app. I wish to create an object withing the JavaScript to represent this chess board.

I will create a single object which will be passes to the data method of the Vue component. I will then make methods that interact with both the object and the competent within the Vue component.