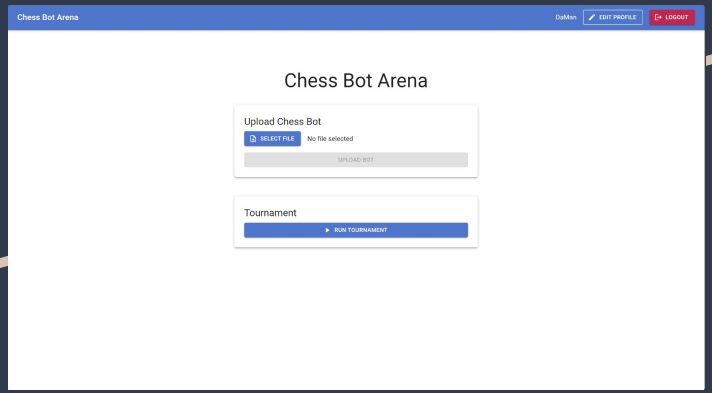
Online Chess Engine Comparator

Orion Gregory, Sebastian Shirk, Tejas Bhadoria

Sprint 2

GitHub:
https://github.com/OrionGregory/ChessEngineComparator

Project Overview



- Create a web app where users can upload chess bots and test them against other user uploaded chess bots in a tournament style.

- Provide log outputs of each tournament and general statistics comparing bot win rates.

- ^^^ That pretty much summed up the whole project according to what she wants. Not sure what else to say.

Feedback from Sponsor

- Need to focus on letting bots play against each other, tournament is main priority over playing individual games
- OAuth is a significant priority
- Allowing users to upload their own bots is another significant priority
- Do not need to focus on visualization of chess game whatsoever
- Do not need to focus on Human vs Chess games

Sprint Goals

- Setup OAuth for user login through google.
- Setup a cloud-hosted Postgresql database.
- Allow users to upload and manage multiple bots.
- Implement tournament functionality.
- Generate and display tournament logs.

Sprint Backlog

ID	Story	Estimated Hours	Priority (1-5)	(Expected) Sprint When Finished	Finished
1	Locally ran chess bot	2	1	1	
2	Create Chessboard on index view	2	1	1	
3	Create upload bot functionality	5	1	1	
4	Create Navbar	1	2	1	
5	Delete User uploaded bots	4	1	1	
6	Move History for chess games	5	1	2	
7	Migrate to Postgressql backend	8	1	2	
8	Bot vs Bot implementation	8	1	2	
9	Chess Bot Difficulty Selector	7	3	2	
10	User Login Functionality	6	1	2	
11	Admin User Management (CRUD)	5	1	2	
12	User Bot Repository	12	2	2	
13	User edit, and delete profile	3	4	2	
14	Leaderboard for bots	4	5	3	
15	Real Time chess visualizations	6	2	3	
16	Tournament Mode	6	5	3	
17	Move analysis	15	3	3	
18	Download other users public bots	15	3	3	

Sprint 1 ^

Cyan - Finished
Yellow - In Progress
Blue - To Be Done

ID	Story	Estimated Hours	Priority(1-5)	(Expected) Sprint When Finished	Finished
1	Locally ran chess bot	2	1	1	
2	Create Chessboard on index view	2	1	1	
3	Create upload bot functionality	5	1	1	
4	Create Navbar	1	2	1	
5	Delete User uploaded bots	4	1	1	
1	OAuth implementation	4	2	2	
5	Valid FEN implentation	1	1	2	
6	Move History for chess games	5	1	2	
7	Migrate to Postgressql backend	8	1	2	
16	Tournament.py working with generic bots	2	1	2	
8	Bot vs Bot implementation	8	1	2	
10	User Login Functionality	6	1	2	
11	Admin User Management (CRUD)	5	1	2	
12	User Bot Repository	12	2	2	
12	Log Output for each match	1	1	2	
13	User edit, and delete profile	3	4	2	
9	Docker	10	1	2	
14	Leaderboard for bots	4	5	3	
15	Real Time chess visualizations	6	2	3	
16	Users and Tournament Logs relationship	4	3	3	
17	Move analysis	15	3	3	
18	Download other users public bots	15	3	3	

Sprint 2 ^

Sprint Reflection

What went well:

- OAuth implementation
- Cloud Postgres database creation
- Upload multiple bot functionality
- Tournament Functionality

What didn’t go well:

- Docker implementation
- Leaderboard implementation
- Restructuring our code

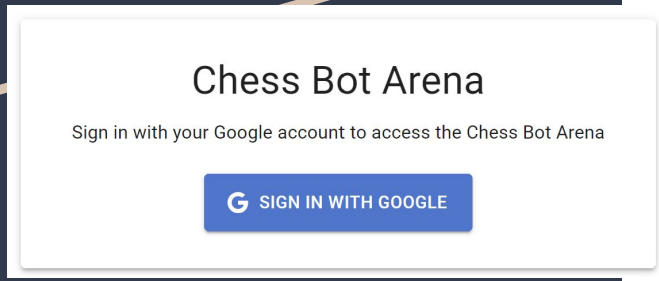
What could be improved:

- Github branch management
- Researching helpful python packages and libraries

Challenges:

- Docker
- Chess matches take a while, especially in large tournaments (5 bots can take 10 minutes)

Sprint Reflection



Orion Gregory:

- Wrote more chess bots/updated chess bots
- Created Tournament logic for bots
- Created Tournament Log system
- Updated Github Pages
- Nav Bar functionality
- Edit User functionality

Sebastian Shirk:

- Local Postgres db/user functionality for testing
- Documentation and ReadMes
- Attempt at Docker (WIP)

Tejas Bhadoria:

- OAuth implementation
- Transfer local Postgres db to cloud
- Server Hosting
- Im probably missing a lot (its late, forgive me)

Demo Time