T09 - Hub Dawgs

Sam Bonafe, Cameron Connell, Tri Nguyen, and Evan Williams

- “Chess holds its master in its own bonds, shackling the mind and brain so that the inner freedom of the very strongest must suffer.” – Albert Einstein

The Game: Extinction Chess

- There is no checkmate or check
- Instead, a player wins when their opponent does not have at least one piece of every type
- All other standard rules apply

Progress:

- Finalized functionality in server-class skeletons:
  - MovePiece can validate all possible moves
  - Board detects win conditions
- Finished Game and Player classes & finalized User class functionality
- Requests to server to move piece
- Significant improvements in test-coverage
  - Added test cases for java classes
  - Added test cases for react classes

Process & Decisions:

Process:

- Started the sprint with most of the general functionality implemented in back-end classes
- Finalized functionality in edge classes & made adjustments to accommodate special cases
- Spent most of the Sprint connecting the front and back ends

Decisions:

- Used MongoDB instead of black-bottle
- Work on requests and connecting front to back before implementing more specific functionality
- Try to use snackbar alerts for notifications/errors

Design Patterns:

There was not much time or need to implement many design patterns but we did use:

- Template for the Pieces
  - Allowed for piece-specific move pattern implementation
- High Cohesion
  - Worsened coupling issues, but classes are well-defined and execute singular, aggregate tasks (making a move)

If we had the time to implement them, these are some of the patterns that may have been useful:

- Factory for the Pieces
  - Reduce complexity in game-board initialization
- Builder for the Board
  - Board class would not have to initialize itself
- Decorator (or something similar) for special moves or pawn promotion
  - Reduce MovePiece class complexity

Ceremonies:

- Schedule:
  - 1 Sprint during P3
- Things that went well:
  - Active communication
  - Teamwork to problem-solve product code and environment issues
  - Work was thoroughly reviewed
- Things that we need to improve:
  - Similar things from previous sprints (distributing work, less wait-time on PRs)
  - GitHub etiquette and writing clean code
- One thing we would change for the future:
  - Clearly defining expectations from team-members

Project Demonstration: