Web Game Project Proposal | CMPT 470 | Group 1

The game:

A turn-based online tactical game where two teams control a large number of units that face off against each other. These units can move a limited number of cells or attack a distance away each turn. Each unit can only perform one action per turn and any player on that team can assign that action that turn. Turns have a set timer and if units have not moved then they lose their turn. One team wins when they destroy the other team. The goal is to be efficient and coordinated with your team. The initial depiction of the game will be on the scale of chess, a small variety of game pieces with varied rules about them, played on a limited scale board.

The tech:

- Front end: Angular.js
- Backend: Node.js
- Database: Express.js and PostgreSQL

We will develop an MVC pattern within Node.js. The models will consist game logic, handled by Node.js and stored via the PostgreSQL database. Angular.js will handle the display of the game information via views and send back the interactions from the users, handled by the controllers. We will also be implementing a live chat system within the game allowing for team and global chat.

Goals:

The primary goal of this project is to create an entertaining game that demonstrates a wide variety of web design tricks and techniques we have learned and demonstrates the MVC architectural pattern. A secondary goal is the development of in-demand web development skills that are marketable to employers.

Challenges:

- How do we update the game board and chat live for all users?
    - Socket.IO seems promising place to start, but more research will need to go into this.
- How about the classic race condition scenario of server arbitrating between two users moving the same piece at the same time?
    - For the scale of this project, this won't be much of an issue.
- How do we handle the data between clients, as each team member will need to be updated for every move made per turn?
    - This may be an unscaleable design decision and limit the size of the game and number of players at some point.

Stretch:

- Deeper gameplay with more features and actions.
- Progression between games.
- Integration of Google Maps API for real location for the game board.