Team Number: Team 2
Team Members: Srihari Meyoor, Jamie King, Jacob Kice, Joseph Hotze, Gunther Luechtefeld

Project Title: Power Chess

Synopsis:

A single screen two player chess game with interactive powerups including improved colorful and vibrant visuals, standard piece logic, and balanced powers for increased fun.

Architecture Description:

Our project is a reimagined version of chess that joins the traditional mechanics of the game with a modern twist of interactive powerups. The goal is to create a multiplayer chess game that both users can play on a single screen with keyboard/mouse inputs.

Piece Class:

The piece class identifies each type of piece on the game board, with each piece containing details for how it can move, what its color is and where it is on the board. A piece's movement abilities can be modified based on power ups. The number of moves a piece has made is also tracked. When a piece is being moved, its eligible move locations are calculated based on its defined set of movement abilities as well as its position on the board and the location of other pieces on the board. It maintains its position information internally and obtains the location of other pieces from the board class. Each type of piece has its own class, which inherits from the base Piece class but is coded with its own movement rules.

Player Class:

The player class distinguishes the two sets of pieces on the board so that two users can play against each other. Pieces are identified as either 'black' or 'white', which affects their interactions with other pieces. Players maintain an array of all of their available pieces on the board. The player also possesses an indicator of whether or not it is their move, and if they are in check. The array of pieces is used in determining check and checkmate status.

Grid Square Class:

The grid square class represents a single tile on the board, storing its position or grid spot such as A5 or F7, the occupying piece and any available powerup. It handles the rendering state (selected, empty etc.) and is used in interaction logic for selecting or moving a piece. Each grid square update, piece or powerup, is based on player actions or random timers and is validated by the backend.

Board Class:

The board class holds the current state of the entire chessboard and serves as the central actor that connects game logic, rendering, and powerups. It contains an 8x8 grid where each space is represented by a Grid Square, possibly containing a chess piece and/or active powerup. It tracks the current state of the board from piece locations to completed moves.

When a player interacts with a piece, the board identifies the piece, obtains and displays its possible moves, and enacts the specific action that the player demands. These actions can be unselecting the piece, thus allowing for the selection of another piece, or selecting a legal move. It then updates the current state of the board checking with the Flask backend for all possible legal moves and actions rendering the necessary elements on the front-end.

Interface Class:

The interface handles displaying the current board elements, what the currently selected piece is for the current player, the possible moves for that piece, and the power up elements on the board. The interface processes user input for each player and makes subsequent calls to the game backend functions to indicate a movement and obtain the updated board information. User input lets the user select a piece, unselect the selected piece, or select a destination for the selected piece. When a piece is selected, its possible moves will be obtained and displayed for the user.