Class Responsibilities

Game

- Dictates the general data flow of the program.
- Takes in user's selection from the **Menu, Mode, GameHistory, and GameBoard** classes.
- It has states based on the user's selection.
- It contains all saved games, holding them in a data structure, such as an ArrayList.
- Will dictate what game will load based on state.

Menu

- It contains label greetings and buttons: Start Game, Load Game, Exit.

Mode

- It contains label greetings and buttons: Easy, Hard, Exit.

GameHistory

- It contains all saved games in ArrayList.
- Is responsible for displaying, and loading the game based on user requests.

GameBoard

- The transaction-handling agent of the **Game.**
- Handles any and all user requests, such as storage, reset, and difficulties.
- Takes in user's input from the **ActionEvent** to generate the valid moves for AI via the **AI** and **GameLogic** classes.
- It is responsible for displaying the saved games.

GameLogic

- It is responsible for generating the board array.
- Handles any and all functionalities, such as isWinner, isPositionAvailable, reset, etc.

AI

- It is responsible for computing the moves based on user requests such as random moves (Easy mode), and optimal moves (Hard mode).

Class Relationships

- **Game** dictates the general data flow of the program based on the user's selection.

- The **menu** is dependent on **GameHistory**.
- The **menu** is dependent on **GameBoard**.
- **GameHistory** is dependent on **GameBoard.**
- The **GameBoard** is dependent on **AI**.
- The **GameBoard** is dependent on **GameLogic**.

Example of Class Relationships: Starting a new game

1. The player starts the game.
2. The Game asks the Mode class to display the level of difficulties.
3. The player picks easy or hard and the Game saves this info to a “hardMode” variable and calls the functionalities based on the mode.
4. Game asks the GameBoard class to display the board for the user to play.
5. If the game is over, the player can reset the game by clicking the “Reset” button.

Example of Class Relationships: Saving a new game

1. While playing the game, the player can save a current game for later by clicking the “Save” button.
2. GameBoard saves all the steps as a string and adds it to an ArrayList.
3. Game navigates the players to the main menu.

Example of Class Relationships: Loading a game

1. Player load the game.
2. Game asks the GameHistory class to display all the saved games.
3. The player picks the specific game and Game class receives that game.
4. Game asks the GameBoard class to load that specific game.