CSci 130 – Fall 2023 - Web Programming

Project – Checkers game (Group project: Maximum 2 students)

The project is worth 15% of the CSci 130 course (total 100 points).

Learning outcomes

Students will work independently and combine the different elements that were seen during the class and the labs. Students will implement a website with both client side and server side, using a relational database management system. Students will use HTML with forms, CSS, JavaScript, and PHP to implement a dynamic website.

Submission

Submission will be done on Canvas by submitting a zip file containing all the files (HTML files, images, CSS files, JavaScript files, PHP files). The project should work from any computer, i.e., no absolute paths in the folder.

Description of the game (from Wikipedia)

Checkers is played by 2 opponents on opposite sides of the game board. One player has dark pieces (usually black); the other has light pieces (usually white or red). Black moves first, then players alternate turns. A player cannot move the opponent's pieces. A move consists of moving a piece diagonally to an adjacent unoccupied square. If the adjacent square contains an opponent's piece, and the square immediately beyond it is vacant, the piece may be captured (and removed from the game) by jumping over it.

Only the dark squares of the checkerboard are used. A piece can only move diagonally into an unoccupied square. When capturing an opponent's piece is possible, capturing is mandatory in most official rules. If the player does not capture, the other player can remove the opponent's piece as a penalty (or muffin), and where there are two or more such positions the player forfeits pieces that cannot be moved (although some rule variations make capturing optional). In almost all variants, the player without pieces remaining, or who cannot move due to being blocked, loses the game.

The game is played with 2 players on **the same screen**. We will consider that the player who is logged into the game will be the first player. The second player will be played by another person on the same machine, same screen.

At the end of the game, we will only record the information from Player 1. At the end of each game, the score, the duration of the game, and the number of turns will be saved in the RDBMS on the server side, so they can be displayed in the leaderboard page. The server side will only be used to save the results of each game, keep information about Player 1.

Main Functionalities and requirements

Documents

- A readme file explaining how to install and use the webpages.
- A file giving details of who has implemented what, what work was undertaken weekly.

Main Pages

- Main page with a menu (index.html)
- Help page: to explain how to play the game (help.html)
- Contact page: short description of the authors of the game (the members of the project) (contact.html)
- Login age: To log into the system + Sign up page: To create a new account.
- The main game (game.html)
- Leaderboard page: To display the best players, to sort players by ascending/descending order of won games, time played, number of games played. To display all the games played by the player who is logged into the system. (leaderboard.html)

Main functionalities

- To have some well-organized HTML5 code with roles to the key elements (e.g., navigation)
- To use appropriate CSS3 functionalities (external CSS file) for the presentation of the web pages using a Fresno State layout, using the appropriate Fresno State logo and images to decorate the site. Use black & white pawns, with a board in wood, like the real game.
- To use Object Oriented Programming (OOP) - classes.
- To display the board on the screen in an HTML table or in a JavaScript Canvas
- To have a button to start the game, to restart the game.
- Options
  - To play on a 8x8 or 10x10 board
  - To change the color/texture of the board
  - To change the color of the pieces for player 1, and player 2.
- Function to place the different pieces on the board (and manage associated events)
- Functions to determine who is currently winning, who has won the game
- Single player or Two player modes
  - Single player: create a second player that plays to maximize the number of taken pawns.
- Hints:
  - Display where you can place a pawn once selected
- Display:
  - The time since the beginning of the game
  - The number of pieces left for each player.
- Server side:
  - A PHP script to create the database automatically and populate it with data
  - Create appropriate Tables in the RDBMS to manage games and users with PHP scripts
  - Return the information to be displayed on the client side.