Connect 4

1) MVC - How you delineated your objects

Model: The model is composed of composed of the essential variables for running the game. It contains the constant variables for the row and column sizes. It also keeps track of the current player to make a move. It has a variable to determine whether the game is over. Also the array that contains the current state of the connect 4 board.

Controller: The controller is what governs the game mechanics. Therefore, it is mostly comprised of functions, rather than variables. The handle click function is a function that runs off an event listener. Any time a player clicks on an element with an event handler, the function will detect whether the move is a valid move. Then it will check if the game is over, else it will send the valid move to the view to update for the User side GUI.

View: This part is self-explanatory, many of the functions are there to replace the empty tile with a premade picture of a red/yellow tile depending on the parameter it is given. It is also in charge of the message the game sends back to the user. This includes, win, draw, loss, how they won, and who is turn it is.

2) Objects - Javascript/PHP - Serialization

Passing info with Javascript Objects to PHP Objects with Cookies

Cookies – Cookies were used to store the user’s last previous username login attempt. Beyond this, we also tried to utilize cookies and session variables to communicate the between the AI and the JS event handling. How it was supposed to work was that after an event was handled, the php script would be ran and would set the session variable as the current number of moves, then the user would click a button signaling a post for the AI to send its move. However, this part of the project was incomplete.

3) Reading/Writing Files/Local Storage

JSON – Javascript

For connect 4, we did not utilize JSON in our current revision. It was simple enough to store, username, password, wins, losses inside a database already and there really was not a need. Perhaps games can be stored via a JSON object, which kept track of the names of the two players, the results, and the moves made.

4) Databases SQL - Identify Entities, Xref, and Enum Tables

Entities – We had a database that stored user entities which contained the username, password, and email.

Xref and enum tables were not utilized.

5) Form Validation - Reqular Expressions

Our form validation and regex usage came in the form of verifying whether or not the user sign up variables were valid in accordance to the username characters and password length security.

6) User-Admin-Login

Our admin section had the ability to edit user data.

The user section just had the ability to play games.

7) Cookies - Sessions - Securing Pages

Cookies – Cookies were used to store the user’s last previous username login attempt. Beyond this, we also tried to utilize cookies and session variables to communicate the between the AI and the JS event handling. How it was supposed to work was that after an event was handled, the php script would be ran and would set the session variable as the current number of moves, then the user would click a button signaling a post for the AI to send its move. However, this part of the project was incomplete.