Problem Statement:

- The client wants a program for a classroom setting in order to engage students in the class' content in a manner that differs or may be more fun than traditional lecturing.

Requirements Documentation:

- Why is there a problem?
    - The client believes the students aren't learning well enough via currently available teaching methods.
- Why is a software system required to solve the problem?
    - 1) If the client were to use a physical Othello board, they would have difficulty communicating the current game state to a full classroom.
    - 2) The client might not possess a physical Othello board or they may not want to store one in their office just for the one class period they'll use it in a semester. It is likely that they will have a computer already with them for their slideshows, notes, and other school materials, so a computerized solution would make use of tools that they already possess.
    - 3) If the client were to try to implement this educational othello game without a computerized solution, they'd have to come up with a method of assigning questions to playable tiles which would be difficult. A computerized solution would easily allow for questions to be assigned to a tile upon a game's start.
    - 4) Should the user want to create new question sets, they'd have to physically have them stored either on paper or notecards. Using a computerized solution, the

client could efficiently create and store repositories of problem sets without having to store and carry questions on notecards or paper.

- What is needed to solve the problem?
    - A computer to run the program.
    - Some sort of user interface to present it to the players.
    - List of 60 questions. Each question is assigned to each not-yet filled in tile.

Specifications Documentation:

- The program must contain the proper ruleset for the game Othello.
    - A “board” consisting of an 8x8 grid of spaces, separated by black lines.
    - The middle 2x2 grid is filled in with alternating black and white tiles.
        - White fills the top left and bottom right tiles.
        - Black fills the top right and bottom left tiles.
    - The player controlling the black tiles will play first.
    - Each counter placed must be placed adjacent to a counter of the opposite color.
    - Upon placing a counter, any opposing pieces laying between the newly placed counter and a counter of the same color will be flipped to the color of the newly placed counter.
        - If a black counter is placed adjacent to a white counter (which has a black counter on the other side of it), then that white counter will be flipped to become a black counter.
    - The game is over once no player has a legal move they can make.

        - The player that has the most counter placed on the board is deemed the winner.
        - A legal move is one where the piece is placed adjacent to a counter of the opposing color.
    - If a player has no legal move, their turn is skipped and the opposing player continues.
        - A player is only skipped if they do not have a legal move, the game ONLY ends once neither player can make a legal move OR the board is filled up (resulting in no legal moves).
- UI Interaction
    - Upon clicking a tile, it would bring up a new form (message box) that contains the question for that tile.
        - The message box will contain the question along with two buttons that read “Correct” and “Incorrect” accordingly
    - Main Menu
        - Start game
            - Asks which question set you’d like to use from your repository list
            - Once selected and accepted, the desired questions will then be assigned to the tiles and the game will begin
        - Question Sets
            - Brings up a menu detailing various options to manipulate sets of questions.

    - Create new question set
    - Delete question set
- Quit game