A computer network diagram with arrows pointing to the earth

Description automatically generated with medium confidenceA cartoon of a person sitting on a chair

Description automatically generated**Cairo University**

**Faculty of Computers and Artificial**

**Intelligence**

**Object-Oriented-Programming**

**Sent to: Dr. Mohamed El-Ramly**

**CS213**

* **Assignment:** Assignment 2.
* **Task:** Task 2,3,4,5.
* **Section:** S 23.
* **Project Name:** Report of our Game Center.
* **Name, IDs and E-Mails:**

|  |  |  |
| --- | --- | --- |
| Name | ID | E-Mail |
| Aly El- Deen Yasser Ali | 20231109 | ali.el.badry.747@gmail.com |
| Fatema El-Zhraa Ahmed Mohamed | 20230280 | fatmaelfeky922@gmail.com |
| Alaa Tarek Mohammed Salah El-Deen | 20230064 | alaatarek2090@gmail.com |

|  |  |
| --- | --- |
| Name | What does he/she do? |
| Aly El-Deen Yasser Ali | **Game 3, 6, 7 and UI** |
| Fatema El-Zhraa Ahmed Mohamed | **Game 2, 5, 8 and UI** |
| Alaa Tarek Mohammed Salah El-Deen | **Game 1, 4, Ui and Report** |

* **Description for classes created for games:**

 **MisereTicTacToc**  
A modified Tic-Tac-Toe game where players aim to avoid winning. It handles board updates, display, and checks for win, draw, or game-over conditions.

 **MisereTicTacTocPlayer**  
Represents a human player who provides move coordinates. It manages player input using their symbol and name.

 **MisereTicTacTocRandomPlayer**  
A player that makes random moves by selecting empty positions on the board, adding unpredictability to the game.

 **MisereTicTacTacAIPlayer**  
Implements an AI player that uses a recursive algorithm to find the best move, ensuring optimal play to avoid winning.

 **TicTacTocSize4Board**  
A 4x4 version of Tic-Tac-Toe. It manages board updates, display, and game outcomes like wins, draws, or game-over states.

 **TicTacTocSize4Player**  
Represents a human player for the 4x4 game, handling input and validating moves during gameplay.

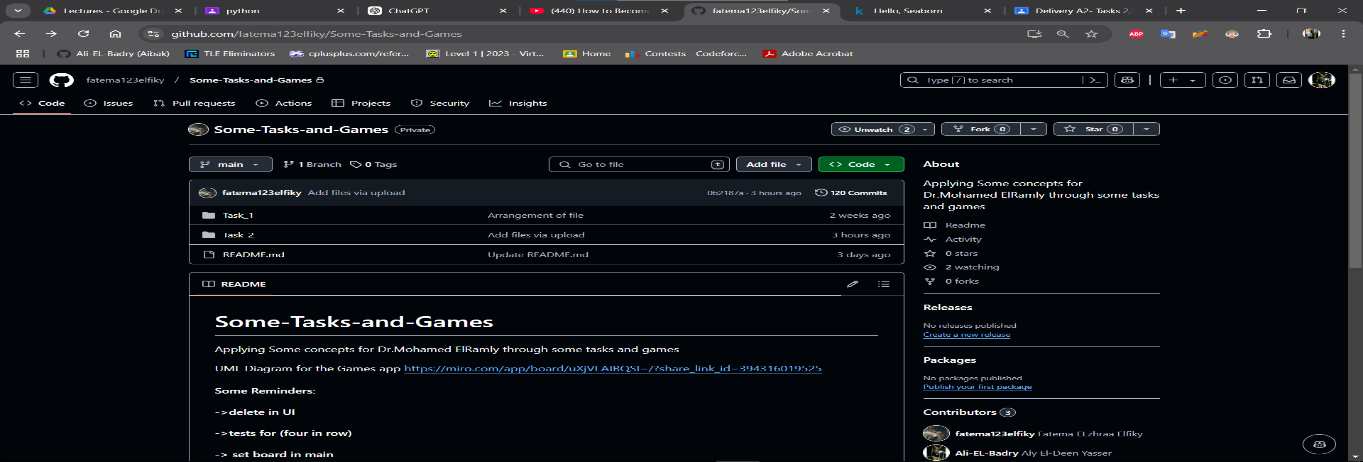
 **TicTacTocSize4RandomPlayer**  
A random player for the 4x4 board, automatically choosing moves to provide simple, unpredictable play.

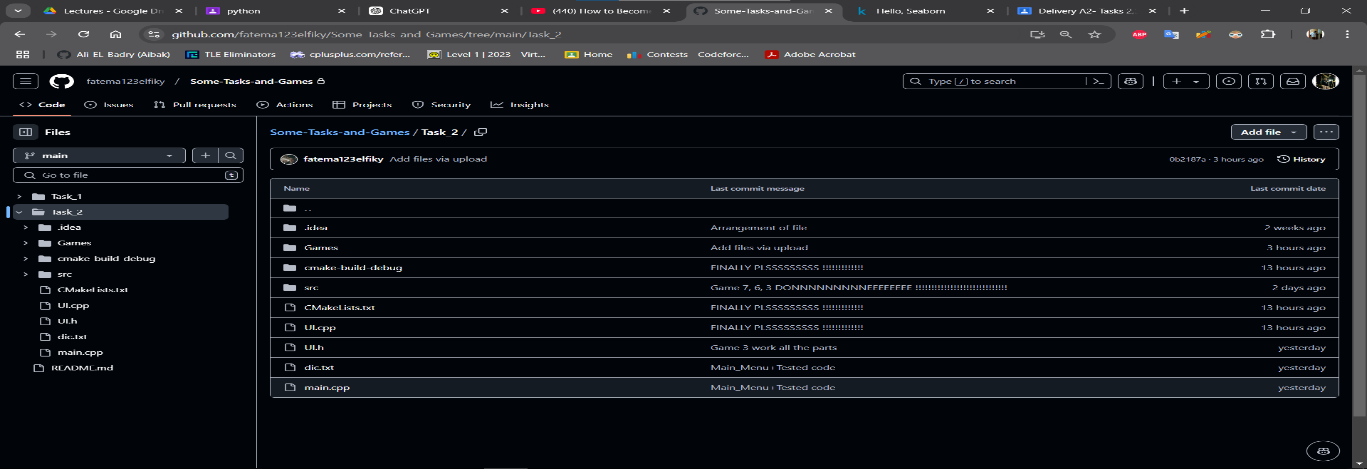
 **TicTacToc\_Size5**  
Implements a 5x5 Tic-Tac-Toe game with additional logic for tracking and checking winning patterns using a map structure.

 **TicTacTocPlayer**  
Represents a human player for the 5x5 game, capturing and validating player moves.

 **TicTacTocRandomPlayer**  
A random player for the 5x5 game that automatically selects moves, adding unpredictability to gameplay.

* **UML Design Digram :**
* **Github Repository:**

****

****

* **AI Bounce Explaination:**

**In Game 6 : The AI player uses the Minimax algorithm to determine the best move in a Misère Tic-Tac-Toe game. The code comprises two main functions:**

1. **calculateMinMax  
   This recursive function evaluates the board state to predict possible outcomes:**
   * **If the game is won (is\_win()), the function returns 1 or -1 based on whether the current player (Maximizing) wins or loses.**
   * **If the game is a draw (is\_draw()), it returns 0.**
   * **For each empty position, the function simulates a move, switches turns to the opponent, and recursively evaluates the next board states.  
     It minimizes the opponent's score while maximizing the AI's score to ensure optimal play.**
2. **getBestMove  
   This function iterates through all available moves to find the optimal one:**
   * **For each empty position, the function simulates a move and uses calculateMinMax to evaluate its outcome.**
   * **The move that yields the highest value (best outcome) is selected.**

**By analyzing all possible future moves and counter-moves, the AI ensures it always plays optimally, making it challenging for an opponent to win.**

* **Aly El-Deen Yasser Ali’s Report:**

My Review of Fatema’s Code:

It is a perfect code that handles many test cases, it is also good as a user interface, but the code has some issues as some of the variable names are inconspicuous and need to be clearer as it is hard to know what is the target from that variable, also, many comments consist of code parts that is commented and not used again.

My Review of Alaa’s Code:

It is a good code that is easy to understand and easy to discover any part of it but there are some issues as it has the same problem as Fatema that many comments consist of code parts that are commented on and not used again, also in the part that takes the name of the user if you entered your name separated with a space it will cause problem and only one part of name will be taken to to the name of player not all the parts.