Cairo University
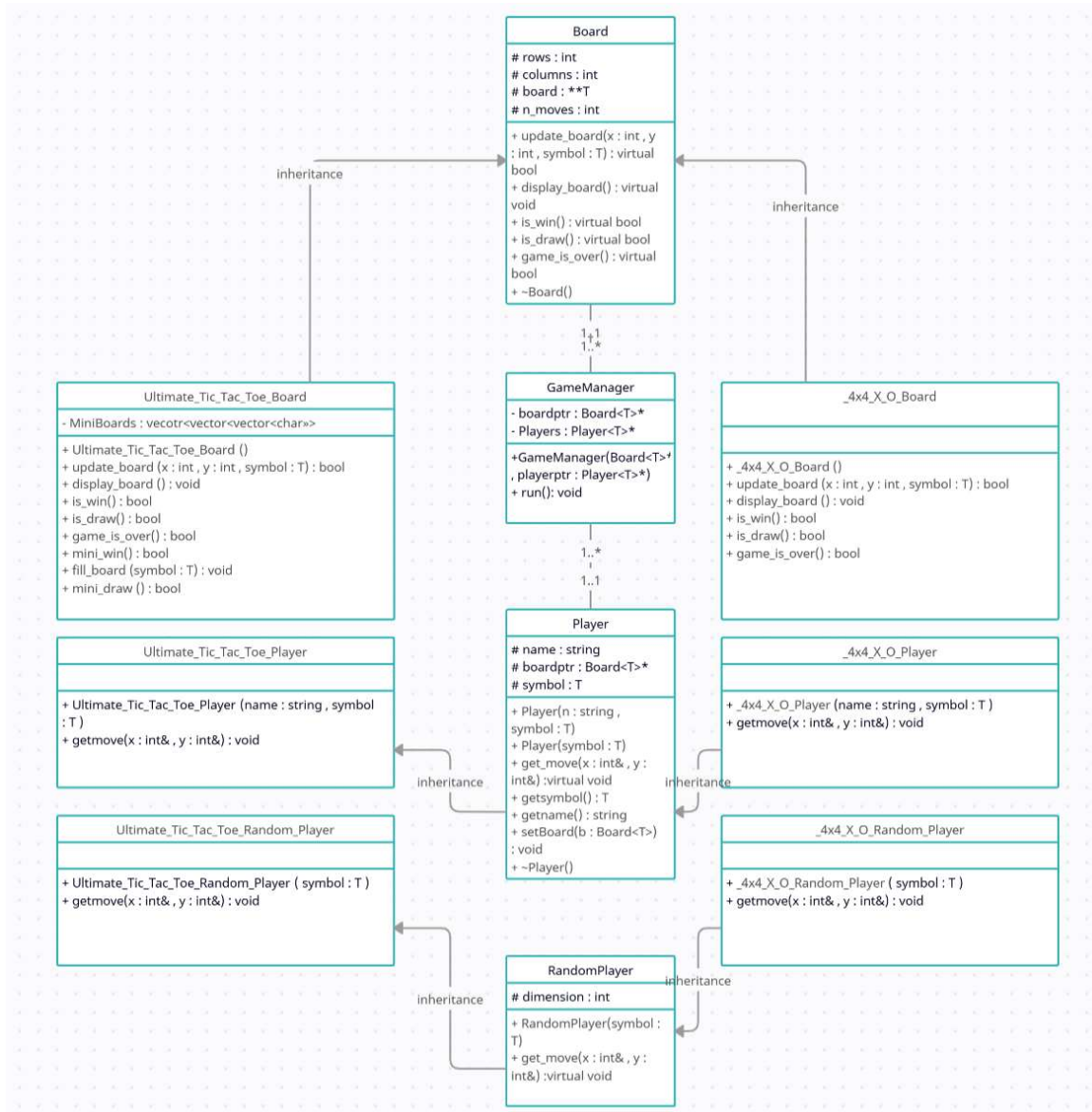# Faculty of Computers and Artificial Intelligence
CS-213
second year
Assignment 2 – Task2,3
Section: S27
Subjected To: Dr Mohamed EL Ramly

Youssef Hossam Ibrahim	20230490	yh55694@gmail.com
Mario Dawod Botros	20230304	mariodawod12@gmail.com
Fatma Hashim Mohamed	20230285	fatma.hashim20233@gmail.com

UML design for the relation of the classes:

Board

rows : int
columns : int
board : **T
n_moves : int

+ update_board(x : int , y : int , symbol : T) : virtual bool
+ display_board() : virtual void
+ is_win() : virtual bool
+ is_draw() : virtual bool
+ game_is_over() : virtual bool
+ ~Board()

inheritance

inheritance

1+1
1..*

Ultimate_Tic_Tac_Toe_Board

- MiniBoards : vecotr<vector<vector<char»>

+ Ultimate_Tic_Tac_Toe_Board ()
+ update_board (x : int , y : int , symbol : T) : bool
+ display_board () : void
+ is_win() : bool
+ is_draw() : bool
+ game_is_over() : bool
+ mini_win() : bool
+ fill_board (symbol : T) : void
+ mini_draw () : bool

GameManager

- boardptr : Board<T>*
- Players : Player<T>*

+GameManager(Board<T>* , playerptr : Player<T>*)
+ run(): void

1..*
1..1

_4x4_X_O_Board

+ _4x4_X_O_Board ()
+ update_board (x : int , y : int , symbol : T) : bool
+ display_board () : void
+ is_win() : bool
+ is_draw() : bool
+ game_is_over() : bool

Ultimate_Tic_Tac_Toe_Player

+ Ultimate_Tic_Tac_Toe_Player (name : string , symbol : T)
+ getmove(x : int& , y : int&) : void

Player

name : string
boardptr : Board<T>*
symbol : T

+ Player(n : string , symbol : T)
+ Player(symbol : T)
+ get_move(x : int& , y : int&) :virtual void
+ getsymbol() : T
+ getname() : string
+ setBoard(b : Board<T>) : void
+ ~Player()

inheritance

inheritance

_4x4_X_O_Player

+ _4x4_X_O_Player (name : string , symbol : T)
+ getmove(x : int& , y : int&) : void

Ultimate_Tic_Tac_Toe_Random_Player

+ Ultimate_Tic_Tac_Toe_Random_Player (symbol : T)
+ getmove(x : int& , y : int&) : void

_4x4_X_O_Random_Player

+ _4x4_X_O_Random_Player (symbol : T)
+ getmove(x : int& , y : int&) : void

RandomPlayer

dimension : int

+ RandomPlayer(symbol : T)
+ get_move(x : int& , y : int&) :virtual void

inheritance

inheritance

Work break-down table:

The whole team collaborate in each function to create the vole machine perfectly but to be more specified:

Mario Dawod Botros	Youssef Hossam Ibrahim	Fatma Hashim Mohamed
Did these 3 games 1. Four-in-a-row 2. Numerical Tic-Tac-Toe 3. Ultimate Tic Tac Toe	Did these 3 games 1. 5 x 5 Tic Tac Toe 2. Misere Tic Tac Toe 3. 4 x 4 Tic-Tac-Toe	Did these 2 games 1. Pyramic Tic-Tac-Toe 2. Word Tic-tac-toe

Report of the quality of the code of other members:

Mario Dawod Botros	Youssef Hossam Ibrahim	Fatma Hashim Mohamed

1-Pyramic tic-tac-toe:
  Works well but didn't use inheritance for the game manager.
2-5X50 tic-tic-tac-toe:
  The code is mostly fine and the game works well without any issues to address.
3-WordBoard tic-tac-toe:
  The game works fine but changed Game Manger implementation.
	1-four in row game:
	  Very clean code with using all the instructions of the assignment
	2-pyramic tic-tac-toe:
	  Works well but leaks to the instructions of using gameboard class
		1-5X50 tic-tac-toe:
		  Good handling to is_win function
		2-numerical tic-tac-toe:
		  Innovative and accurate rule implementatio

GitHub Work:

https://github.com/mario985/BoardGame