Gameboard
entireGame: TicTacBoard*[9]
currentBoard: int
boardWinner: int
#currentTurn: int
+GameBoard() :
+displayGameBoard(boardNumbersDisplay: int) : void
+getElementInOneSquare(square: TicTacBoard&, index: int) : char
+setElementInOneSquare(boardIndex: int, squareIndex: int, newElement: char) : void
+getOneTicTacBoard (index: int) : TicTacBoard
+getCurrentBoardNumber () : int
+setCurrentBoard (newBoard: int) : void
+getBoardWinner():int
+setBoardWinner (newWinner : int) : void
+getTurn() : int
+setTurn(newTurn : int) : void
+checkWin() : bool
+changePlayer(newPlayer:int) : void

Display : protected Gameboard
-game:Gameboard
-XMAX:int
-YMAX:int
-fixchax: char[18]
-fixchao: char[18]
-beegx: char[160]
-beego: char[160]
+drawGrid(board:GameBoard*,color:int,grid:int) : void
+drawChips(dest:int*,chip:char,board:Gameboard*) : void
+drawBeegChips(dest:int*,chip:char,board:Gameboard*) : void
+convertToCoordsBeeg(boardNum:int,coord,int*) : void
+startLittle(boardNum:int,coord:int*) : void
+addLittle(coord:int*,place:int) : void
+convertLittle(coord:int*,boardNum:int,place:int) : int

- This was meant to be polymorphism but was not able to be fully implemented for reasons explained in the power point.

TicTacBoard
-avaiableSquares: char[9]
-winner: char
+TicTacBoard():
+getElement(index:int) : char
+setElement(index:int,newElement:char) : void
+displayBoard(currentCursorY:int,currentCursorX:int, displayAll:bool) : void
+checkWin() : bool

Logic
-game:Gameboard
-currentBoardNum:int
+logic() :
+gameOpened(): void
+playGame(game:GameBoard&) : void
+ifBoardAlreadyWon(game : GameBoard&) : void
+menuForPlayingGame(game:GameBoard&) : char
+saveGame(game:GameBoard&) : void
+loadGame(game:GameBoard&) : bool
+menuGameOpened():int