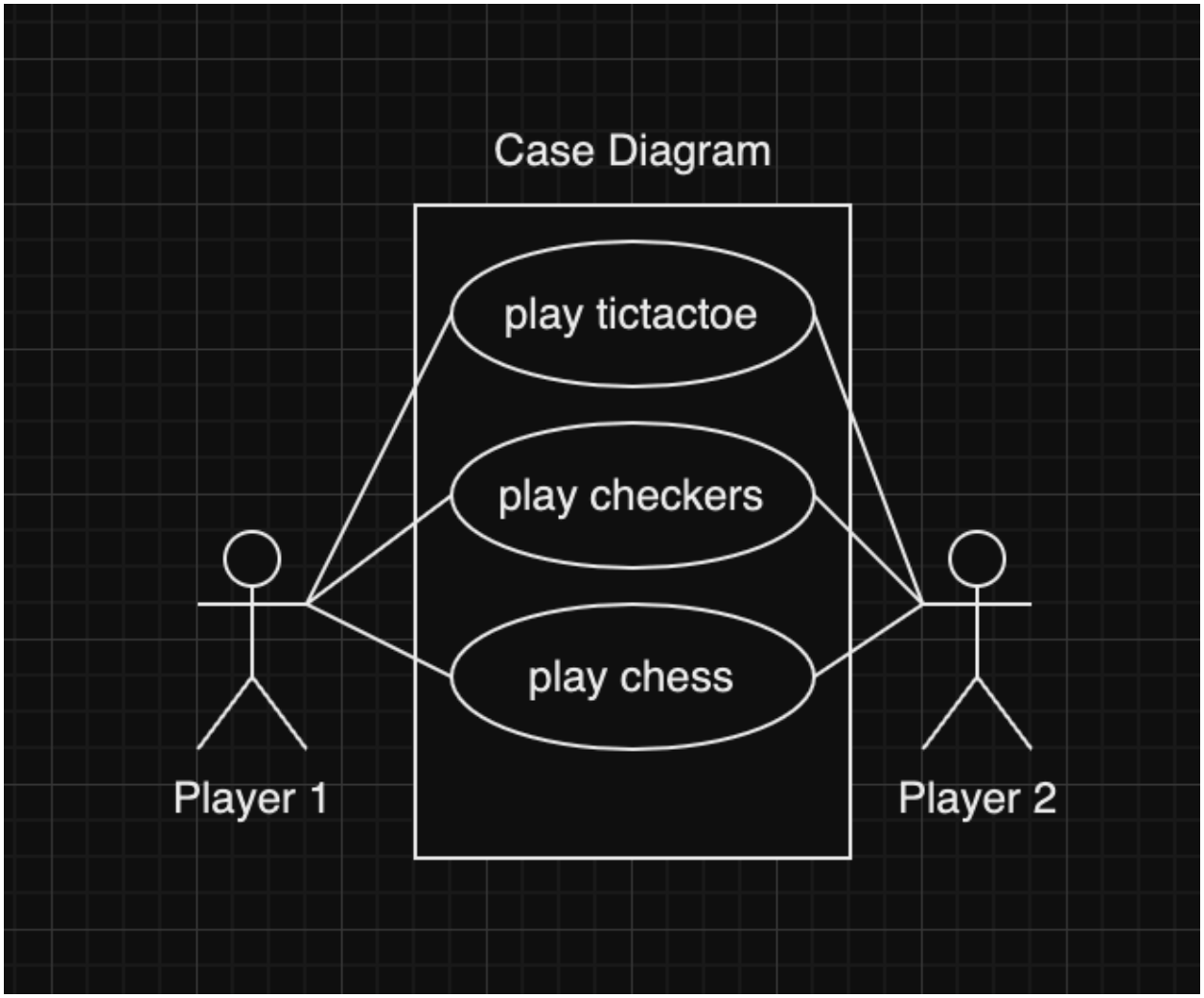
CEN4010 Group 3
Iteration 1



SCENARIO 1

Use Case: Play TicTacToe
Actors: User (player1, player2)
Goals: Play game of tic tac toe
Steps:

(1) Players play TicTacToe

Actor/User's Action	System's Response
	1. System shows 3 Brain games for the user to choose
2. User Selects TicTacToe	3. System shows a 3x3 grid starting with Player 1's turn
4. Player 1 clicks unselected grid	5. System marks the grid with an "X"
6. Player 2 clicks unselected grid	7. System marks the grip with an "O"
8. Steps 4 and 6 are repeated until either of the users win or the game ends in a draw	9.System shows which player won or if it ended in a draw and gives the option to reset the grid for next game or go to main menu

SCENARIO 2

Use Case: Play checkers
Actors: User (player1, player2)
Goals: Play game of Checkers
Steps:

Actor/User's Actions	System's Response
	1. System shows 3 Brain games for the user to choose
2. User selects checkers	3. System shows an 8x8 grid with 12 red and 12 black pieces alternating on each side
4. Player 1 selects a red piece	5. System shows the available tiles for that red piece to legally move
7. Player 1 selects the grid where they want to move it to	8. System moves the selected piece to the new grid
9. Player 2 selects a black piece	10. System shows the available moves for that that black piece
11. Player 2 select the grid where they want to move the piece to	12. System moves the selected piece to the new grid
13. Steps 4 to 12 are repeated until one of the players wins	14. System shows which player won and give the option to reset the game or go to main menu

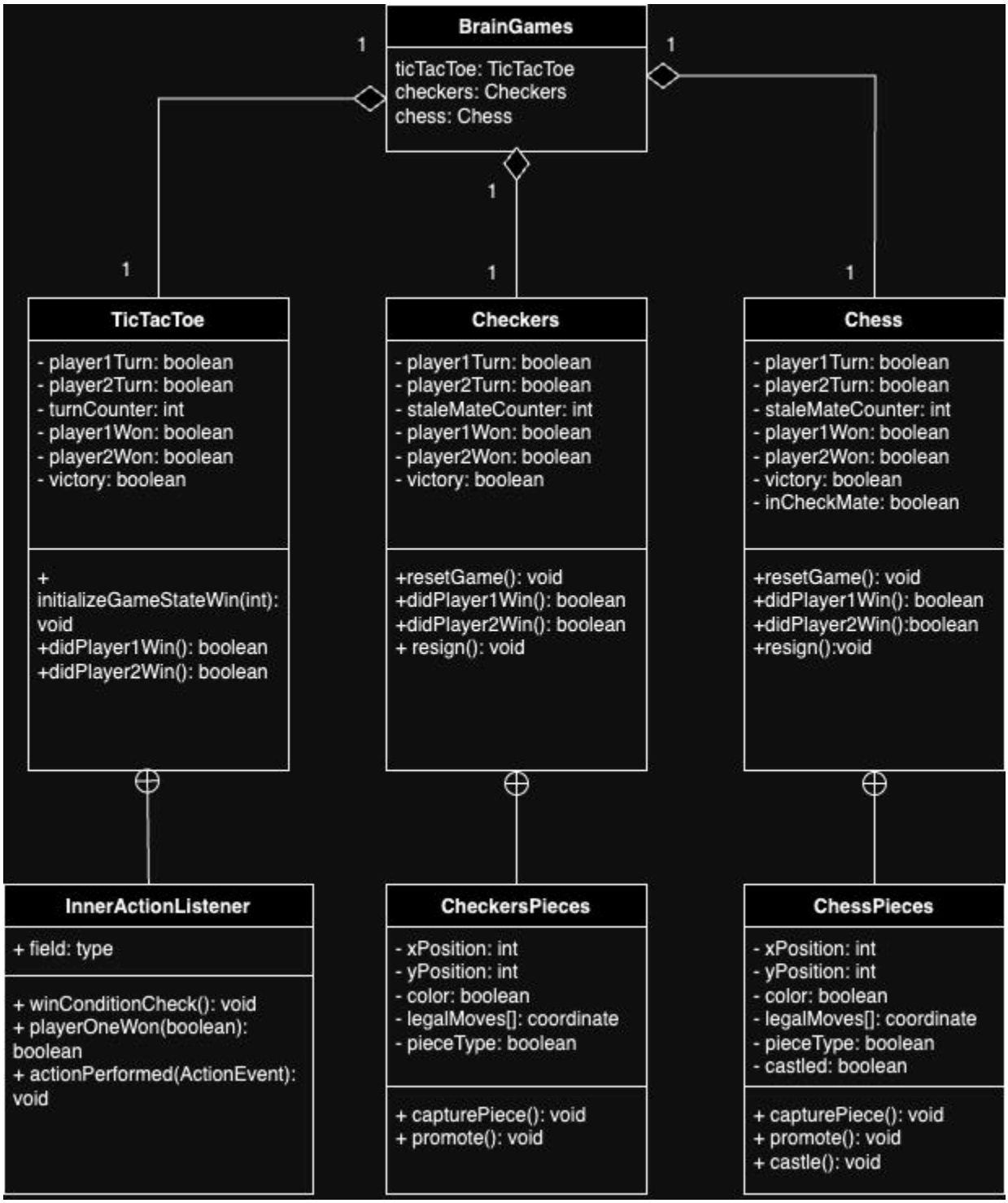
SCENARIO 3

(3) Players play Chess

Use Case: Play chess
Actors: User (player1, player2)
Goals: Play game of chess
Steps:

Actor/User's Actions	System's Response
	1. System shows 3 Brain games for the user to choose
2. User chooses Chess	3. System shows an 8x8 grid with all white and black chess pieces on opposite sides of the board
4. Player 1 selects the white piece they want to move	5. System highlights the piece and shows the available spaces on the grid it can move to
6. Player 1 selects the grid they want to move their white piece to	7. System moves the piece
8. Player 2 selects the black piece they want to move	9. System highlights the selected piece and shows the available spaces on the grid it can move to
10. Player 2 selects the grid they want to move their black piece to	11. System moves the piece
12. Steps 4 to 12 are repeated until one of the players wins or it ends in a draw	13. System shows the winner or draw condition and gives the option to reset the game or go to main menu

UML DIAGRAM


BrainGames
ticTacToe: TicTacToe
checkers: Checkers
chess: Chess
1
1
1
1
1
1
TicTacToe
- player1Turn: boolean
- player2Turn: boolean
- turnCounter: int
- player1Won: boolean
- player2Won: boolean
- victory: boolean
+ initializeGameStateWin(int): void
+didPlayer1Win(): boolean
+didPlayer2Win(): boolean
Checkers
- player1Turn: boolean
- player2Turn: boolean
- staleMateCounter: int
- player1Won: boolean
- player2Won: boolean
- victory: boolean
+resetGame(): void
+didPlayer1Win(): boolean
+didPlayer2Win(): boolean
+ resign(): void
Chess
- player1Turn: boolean
- player2Turn: boolean
- staleMateCounter: int
- player1Won: boolean
- player2Won: boolean
- victory: boolean
- inCheckMate: boolean
+resetGame(): void
+didPlayer1Win(): boolean
+didPlayer2Win():boolean
+resign():void
InnerActionListener
+ field: type
+ winConditionCheck(): void
+ playerOneWon(boolean): boolean
+ actionPerformed(ActionEvent): void
CheckersPieces
- xPosition: int
- yPosition: int
- color: boolean
- legalMoves[]: coordinate
- pieceType: boolean
+ capturePiece(): void
+ promote(): void
ChessPieces
- xPosition: int
- yPosition: int
- color: boolean
- legalMoves[]: coordinate
- pieceType: boolean
- castled: boolean
+ capturePiece(): void
+ promote(): void
+ castle(): void