**Tic Tac Toe Basic**

For a draw, the maximum number of moves is the length of the board squared.

Counter move = 1

Do:

Turns determined by move % 2: NB will need to change if more than 2 players

* If odd, player 1 % = 1
* If even, player 2 % = 0

Player gets instruction to input coordinates or q to give up

Input is validated:

Quits if q is pressed

Otherwise input is validated to ensure the format is correct (int comma int)

If coords are not in valid format, return error message ask player to try again.

If in valid format, validate coords are within range of the array & position is not already occupied

If not in valid range or position is already filled, return error message and ask player to try again

If valid, place X onto board

Move ++

While IsAWin is not true or move > 9 (if the board is filled)

Move = 1

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 0,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 2

2 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player two enters 0,1

Input is in correct format

Coord is valid

Coord placed on board

Move = 3

3 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 0,2

Input is in correct format

Coord is valid

Coord placed on board

Move = 4

4 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player two enters 1,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 5

5 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 1,1

Input is in correct format

Coord is valid

Coord placed on board

Move = 6

6 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player one enters 1,2

Input is in correct format

Coord is valid

Coord placed on board

Move = 7

7 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 2,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 8

8 % 2 = 0 player 2 turn

It is a win, game over

Move = 1

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 0,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 2

2 % 2 = 0 – player 2 turn

It is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player one enters 1,1

Input is in correct format

Coord is valid

Coord placed on board

Move = 3

3 % 2 = 1 – player 1 turn

It is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 0,3

Input is in correct format

Coord is valid

Coord placed on board

Move = 4

4 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player one enters 1,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 5

5 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 1,2

Input is in correct format

Coord is valid

Coord placed on board

Move = 6

6 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player one enters 2,2

Input is in correct format

Coord is valid

Coord placed on board

Move = 7

7 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 2,0

Input is in correct format

Coord is valid

Coord placed on board

Move = 8

8 % 2 = 0 – player 2 turn

Is not a win

Move is not greater than 9

Instruction: “Player 2 enter a coord x,y to place your O or enter ‘q’ to give up

Player one enters 0,1

Input is in correct format

Coord is valid

Coord placed on board

Move = 9

9 % 2 = 1 – player 1 turn

Is not a win

Move is not greater than 9

Instruction: “Player 1 enter a coord x,y to place your X or enter ‘q’ to give up

Player one enters 2,1

Input is in correct format

Coord is valid

Coord placed on board

Move = 10

10 % 2 = 0 – player 2 turn

Is not a win

Move is greater than 9

It is a draw

Game over

**Any size Tic Tac Toe**

***Expected changes***

*Number of consecutive coords needs to be determined. Will start with 3 as per the basic game.*

IsAWin needs to access the player.playerid

* Add to interface IWinningBoard

*Initialise Board*

Write to console “What size would you like the board?”

Read input from console

Send input to Print method

Print method writes board to console

Size = board.length NB determined by players – from 3x3 to 10x10 (defined as const)

Reverse = size – 1 – defined on DiagonalDeterminator

\*\* Replace for loop through rows with foreach loop

*Column determinator logic:*

Count == 0

\*\*For loop through rows

For loop through columns

If board[col][row] equals player.PlayerId

Count++

If count equals 3

Return true

Break

Else

Count = 0

Return false

End loop through columns

End loop through rows

*Row determinator logic:*

Count == 0

\*\*For loop through rows

For loop through columns

If board[row][column] equals player.PlayerId

Count++

If count equals 3

Return true

Break

Else

Count = 0

Return false

End loop through columns

End loop through rows

*Diagonal determinator logic:*

Count == 0

\*\*For loop through rows

For loop through columns

If board[row][row] equals player.PlayerId

Or

If board[row][reverse] equals player.PlayerId

Count++

If count equals 3

Return true

Break

Else

Count = 0

Return false

End loop through columns

End loop through rows