

HW# 2 Tic-Tac-Toe Pseudocode Plan

public class TicTacToe

- Store board in 3x3 2D Array

private int[][] positions = new int[3][3];

- Enums for X, O, & EMPTY

private enum elements {X, O, Empty}

Main -

- Set gameboard / positions to Empty

↑ gameBoard_Data

- Scanner input "Play 2 player or 1 Player?"

if 1 Player : "Go 1st or 2nd?"

2 Player Pro.v.s. Pro -

Scanner Input - Get int from 1-9

↳ player 1 Position

while gameBoard_Data.contains(*)

Print "Position Taken"

p1 Pos. = Scan.nextInt()

Place Piece (p1 = X, gameBoard_Data, player 1 Pos.)

Scanner Input

P2 = O

Place Piece

checkwinner() < placePiece()

gameBoard[][] = elements.X print()

need to make it of type elements

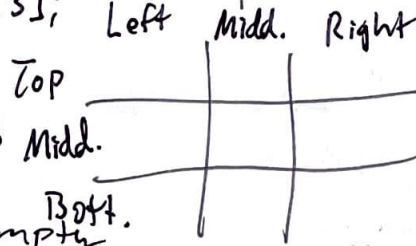
[]
[]
[]

→ place Enums in each array slot

- Initialize all to Empty

- Place X or O in there

- 8 diff. ways to win



ex. Top+Left → X

Place Piece (gameBoard, user)

if user = X

O = User

else if user = O

switch

case: #'s 1-9

- place in array the Enum

checkwinner()

store all combos of how to win, & cross check that with the state of the Board

ArrayList<List> Win

type List of combos

TopRow = 1, 2, 3

Cross 1 = 1, 5, 9 etc.

Win.add(TopRow)

for (List a : Win)

if (gameBoard.containsAll(a))

switch for enum

Print "Congrats!" Player

- 2D Array with 2 different Enums in it

Win: 3 enums in a row in specific order in a 2D array

• 2D Array [3][3]

• Enum Elements

• Constructor → Empty

• 2 User Play

• User decides go 1st or 2nd vs. CPU

~ Print gameBoard

var = gameBoard_viz

Make Base images
of all winning combinations
& compare to GameBoard

[X]		
[X]
[X]

Base

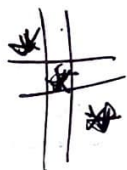
[0][0]
[1][1]
[2][2]

[X]		[O]
[O]	X	
[X]

GameBoard

Loop int through [][]
if [][] equals Base[j][i]
→ winner

x-coordinate // y-coordinate
(0,0) (0,1) (0,2)
(1,0) (1,1) (1,2)
(2,0) (2,1) (2,2)



3x3 Array
of chars
representing
ways to win

winning ways

~~2D~~ Cross 1
2D Array
with ~~X~~ on
(0,0)
(1,1)
(2,2)

If gameBoard count = 0

for int a; a < 3; a++
for int b < 3; b++

if gameBoard[a][b] == cross1[a][b]

count++

Loop

If count = 3

↳ "Winner"

Computer vs. Computer

Need the computer to try to
win

- Find Paths ~~###~~
open
and take them

Random Seed

Decides
List of Paths [2 diff.
options]

- Then CPU takes
one.

checks the Path
if Empty enum.

- If taken by other
CPU, random #
to generate new
path / plan.

What happens if they
need to change plans though?
↳ Need a checker - for
the if the plan
is still good

if [][] != EMPTY
if [][] != EMPTY
if [][] != EMPTY
Then choose
this plan