

Evan Lanning , Jeremiah Robinson Washington, Nathaniel Carter

## **The Boys Tic Tac Toe**

We want to make a 4x4 board of tic-tac-toe.

Increase board size

As in (3x3) →(4x4)

Added more inputs

As in 9 total digits → 16 total digits

Made win harder

So that 4 in a row = win instead of 3

These are the changes we made to code we found online that is generalized and can not be sourced to one person

Our general functionality of our code is to create a 4x4 board of tic tac toe.

We seek to solve the lack of a 4x4 tic tac toe board in the world

Our objective is to create a functioning 4x4 tic tac toe board

A functioning computer setup

A pseudo code representation

Function print\_board(board):

For each row in the board:

Print each cell in the row separated by " | "

Print a line of "-" with length 16

Function check\_win(board, player):

# Check rows for a win

For each row in the board:

If all cells in the row are equal to the player:

Return True (player wins)

# Check columns for a win

For each column (0 to 3):

If all cells in the column are equal to the player:

Return True (player wins)

# Check diagonals for a win

If the main diagonal (top-left to bottom-right) all equal the player:

Return True (player wins)

If the anti-diagonal (top-right to bottom-left) all equal the player:

Return True (player wins)

Return False (no win)

Function get\_move():

Repeat indefinitely:

Ask the player to enter a move (1-16)

If the input is valid and within range (1-16):

Convert the move to 0-indexed row and column (using divmod)

Return the row and column as a tuple

Else:

Print an error message and ask again

Function play\_tic\_tac\_toe():

Initialize a 4x4 board with empty spaces (" ")

Set current\_player to "X"

For 16 turns (maximum number of moves):

Print the current board

Get the row and column of the player's move using get\_move()

If the chosen spot is empty:

Place the player's mark ("X" or "O") at the selected spot

If the player has won (check\_win returns True):

Print the final board

Print the winning player

End the game

Switch current\_player (from "X" to "O" or vice versa)

Else:

Print a message saying the spot is taken and ask for another move

Print the final board

Print "It's a tie!" (if no one wins after 16 turns)

Main execution:

Call play\_tic\_tac\_toe() to start the game

spot is taken and ask for another move Print the final board Print "It's a tie!" (if no one wins after 16 turns)

Main execution: Call play\_tic\_tac\_toe() to start the game

We discover this algorithm by looking it up and copying and then manipulating it