**BY: RIYA GUNDA 11B & PRATHIK GUNTURU 11A**

**IP PROJECT**

**CONNECT 4**

**ABSTRACT:**

Connect 4 is a two-player game where each of the player aims to get four pieces of their own colour lined up in a row horizontally, vertically or on an angle (cross ways). The first player to get four of their own pieces in a row wins the round.

The players’ turns will be alternating starting with player 1. He / She would place their piece in any column of only the first row by entering a column number into pop-up box on the top-left corner. Player 2 would then place his/ her piece on top of player 1's piece or in the first row of any other column. The pieces get stacked in the game, so unless there is another piece under your piece, it would not be able to go higher than the first row.

Using the turtle library on python, we imported a source code and edited it to our preference and requirements. After every edit made, the code was run to see if it was still functioning and performed the tasks we desired.

The game consists of 2 pages:

* Home page
* Game page

The project consists of a home page and game page with the following features:

* A home page with instructions for the game and a play button leading to the game page.
* Box that allows you to insert column value of where you want to place your piece on the game page.
* Quit window to leave the game; enter quit into the box on the top-left corner
* A side bar displaying the winners color and player number on the game page

**SYSTEM REQUIREMENTS:**

Hardware requirements:

1. Computer
2. X86 64-bit intel architecture
3. AM4
4. 64-bit AMD architecture
5. 4GB RAM
6. 5GB storage

Software requirements:

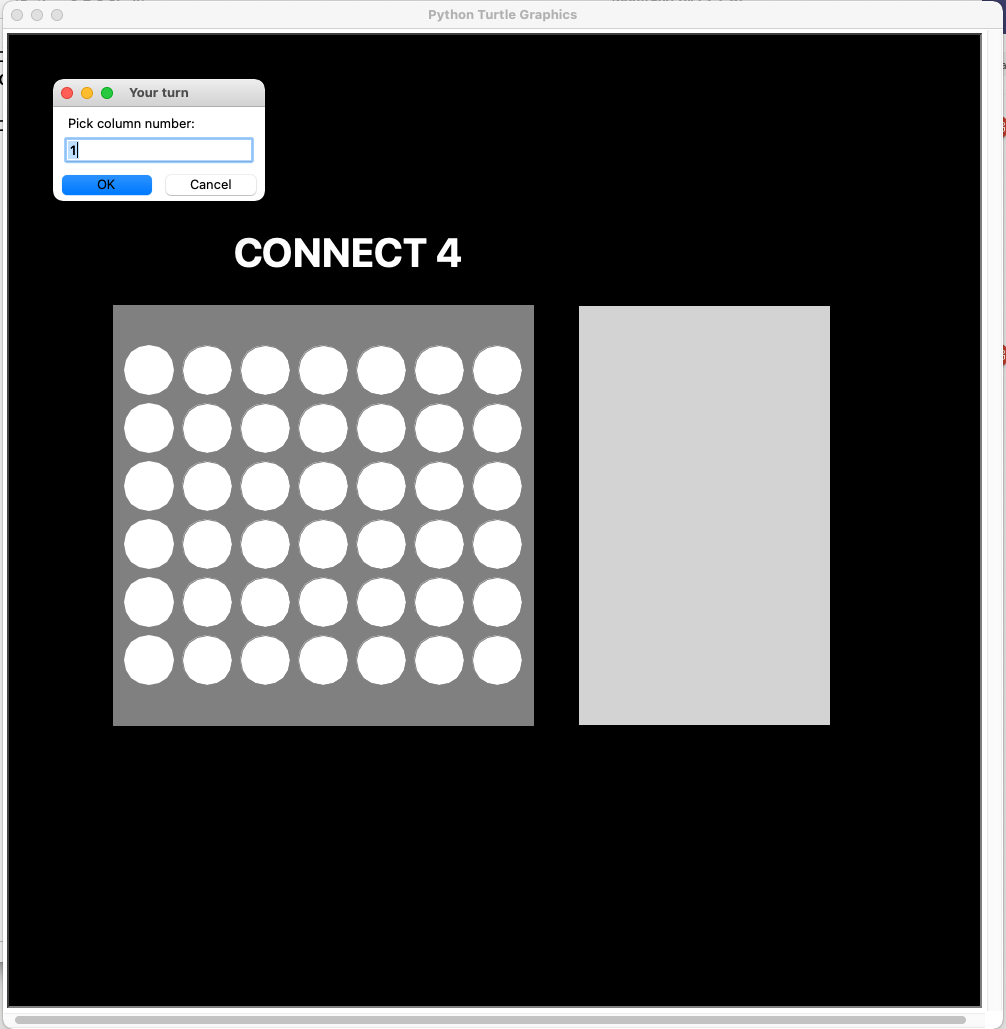
1. Python
2. Turtle library

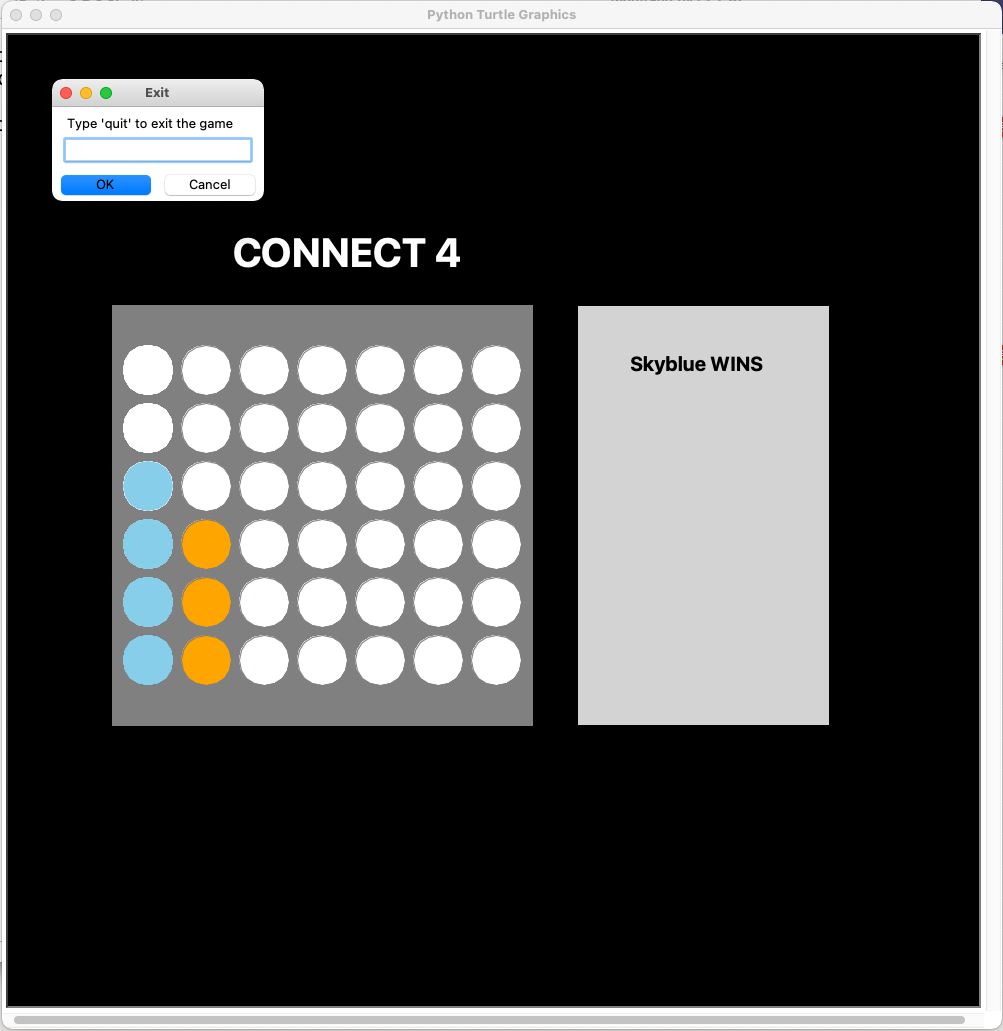
**CONNECT 4 CODE:**

import turtle  
     
def new\_wind():  
    turtle.clearscreen()  
    Pen=turtle.Turtle()  
    Pen.hideturtle()  
    Pen.speed(500)  
    window = turtle.Screen()  
    window.screensize(1000,400)  
    window.bgcolor("black")  
    Pen.speed(0)  
    Pen.\_tracer(8,25)  
    turtle.setup(1000, 1000)  
    #WindColor=turtle.textinput("Background Colour","Please input colour name: ")  
    #window.bgcolor(WindColour)  
    #Player1color=turtle.textinput("Player 1 colour","Please enter player 1 colour: ")  
    #Player2color=turtle.textinput("Player 2 colour","Please enter player 2 colour: ")  
    #PlayingWindcolor=turtle.textinput("Playing window colour(not to be same as background color)","Please enter desired color: ")  
  
  
    connect4=[[0 for c in range(7)] for x in range(6)]  
  
    def draw\_grid(grid):  
        Pen.setheading(0)  
        Pen.goto(-350, 130)  
        for rower in range(0, 6):  
            for col in range(0, 7):  
                if grid[rower][col] == 0:  
                    Pen.fillcolor("white")  
                elif grid[rower][col] == 2:  
                    Pen.fillcolor("skyblue")  
                elif grid[rower][col] == 1:  
                    Pen.fillcolor("orange")  
  
                Pen.begin\_fill()  
                Pen.circle(25)  
                Pen.end\_fill()  
  
                Pen.penup()  
                Pen.forward(58)  
                Pen.pendown()  
            Pen.setheading(270)  
            Pen.penup()  
            Pen.forward(58)  
            Pen.setheading(180)  
            Pen.forward(58 \* 7)  
            Pen.setheading(0)  
            Pen.getscreen().update()  
  
  
    def draw\_board():  
        Pen.up()  
        Pen.setheading(0)  
        Pen.goto(-386, -200)  
        Pen.begin\_fill()  
        for b in range(4):  
            Pen.color("grey")  
            Pen.pendown()  
            Pen.forward(420)  
            Pen.left(90)  
        Pen.up()  
        Pen.end\_fill()  
  
  
    def draw\_game\_panel():  
        Pen.up()  
        Pen.setheading(0)  
        Pen.goto(80, 219)  
        Pen.begin\_fill()  
        for rectangle in range(2):  
            Pen.color("lightgrey")  
            Pen.down()  
            Pen.forward(250)  
            Pen.right(90)  
            Pen.forward(418)  
            Pen.right(90)  
        Pen.end\_fill()  
        Pen.up()  
        Pen.color("white")  
        Pen.goto(-150, 250)  
        Pen.write("CONNECT 4", True, align="center", font=("Broadway", 40, "bold"))  
  
  
    def check\_if\_winner(grid, color):  
        # Vertical row checking  
        for r in range(6):  
            for c in range(4):  
                if grid[r][c] == color and grid[r][c+1] == color and grid[r][c+2] == color and grid[r][c+3] == color:  
                    return color  
        # Horizontal row checking  
        for x in range(3):  
            for y in range(7):  
                if grid[x][y] == color and grid[x+1][y] == color and grid[x+2][y] == color and grid[x+3][y] == color:  
                    return color  
        # Diagonal checking  
        for i in range(3):  
            for z in range(4):  
                if grid[i][z] == color and grid[i+1][z+1] == color and grid[i+2][z+2] == color and grid[i+3][z+3] == color:  
                    return color  
        # Diagonal checking  
        for d in range(5, 2, -1):  
            for c in range(4):  
                if grid[d][c] == color and grid[d-1][c+1] == color and grid[d-2][c+2] == color and grid[d-3][c+3] == color:  
                    return color  
  
        return 0  
  
  
    def display\_winner(winners):  
        if winners == 2:  
            Pen.penup()  
            Pen.color("Black")  
            Pen.goto(200, 150)  
            Pen.write("Skyblue WINS", True, align="center", font=("Broadway", 20, "bold"))  
            Pen.getscreen().update()  
            return True  
        elif winners == 1:  
            Pen.penup()  
            Pen.color("Black")  
            Pen.goto(200, 150)  
            Pen.write("Orange WINS", True, align="center", font=("Broadway", 20, "bold"))  
            Pen.getscreen().update()  
            return True  
  
  
    def loop():  
        # Game loop, play up to 42 turns  
        for player\_turn in range(1, 43):  
            column\_string = window.numinput("Your turn", "Pick column number:", 1, minval=1, maxval=7)  
            column = int(column\_string)  
            column\_minus = column - 1  
            while connect4[0][column\_minus] != 0:  
                # This column is already full, pick another one  
                column\_string = window.numinput("Your turn", "Pick other column number row is full:", 1, minval=1, maxval=7)  
                column = int(column\_string)  
                column\_minus = column - 1  
  
            # Make the chips stack up one another  
            row = 5  
            while connect4[row][column\_minus] != 0:  
                row = row - 1  
            # Find out the colour of the current player (1 or 2)  
            playerColor = int((player\_turn % 2) + 1)  
            # Place the token on the grid  
            connect4[row][column\_minus] = playerColor  
            # Draw the grid  
            winner = check\_if\_winner(connect4, playerColor)  
            draw\_grid(connect4)  
            if display\_winner(winner):  
                user\_input = window.textinput("Exit", "Type 'quit' to exit the game")  
                while True:  
                    if user\_input == 'quit':  
                        print("Game has been exited!")  
                        exit()  
                    else:  
                        user\_input = window.textinput("Exit", "Type 'quit' to exit the game")  
                break  
            draw\_grid(connect4)  
  
  
    draw\_game\_panel()  
    draw\_board()  
    draw\_grid(connect4)  
    loop()  
  
  
  
def check\_button(x, y):  
    if -300 < x < 300 and 0 < y < 200:  
        new\_wind()  
  
screen = turtle.Screen()  
screen.setup(1100,900)  
screen.bgcolor("Orange")  
turtle.hideturtle()  
turtle.speed(2000)  
turtle.penup()  
turtle.goto(0,200)  
turtle.write("Connect 4",move=True, align="center", font=("Broadway", 150, "bold"))  
turtle.penup()  
turtle.goto(-300,50)  
turtle.pendown()  
turtle.begin\_fill()  
turtle.fillcolor("Black")  
turtle.fd(600)  
turtle.left(90)  
turtle.fd(150)  
turtle.left(90)  
turtle.fd(600)  
turtle.left(90)  
turtle.fd(150)  
turtle.left(90)  
turtle.end\_fill()  
[turtle.ht](http://turtle.ht/)()  
turtle.penup()  
turtle.goto (5, 55)  
turtle.color("Skyblue")  
turtle.write("PLAY", move=False, align="center", font=("Broadway", 120, "bold"))  
turtle.penup()  
turtle.goto(-10,-300)  
turtle.color("Black")  
turtle.write("""    1)   Players will choose to be either player 1 (Blue piece) or player 2 (Orange piece).  
  
            2)  The players will be alternating turns starting with player 1.  
  
            3)  Player 1 will start the game by placing his/her piece in any column of the first row.  
                Player 2 will then place his/her piece in the first row of any other column or on top of player 1’s piece.  
  
            4)  The pieces get stacked in the game, so to place your piece on a higher row, there must be a piece under yours.  
  
            5)  The player that connects 4 of their own pieces horizontally, vertically or on an angle (cross) will win the game.""",move=False, align="center",font=("Kristen ITC",20))  
  
turtle.onscreenclick(check\_button)  
  
turtle.mainloop()  
  
  
turtle.mainloop()

**SCREEN DESIGN:**

****





The game still has improvements that can be made. We plan to include the following features in the future:

* Add code to keep track of the players score
* Allow players to pick their own colors and names
* Include a menu page
* Adding a feature to allow players to place their pieces by clicking on the grid rather than entering a column number.
* Adding buttons (quit buttons, instruction button, home button, etc.)

Source code 🡪 https://github.com/ILLUSIONack/Connect-4-Turtle/blob/master/Connect4.py