

```

class Player:
    def __init__(self, symbol):
        self.symbol = symbol

    def getMove(self):
        return input(f"Player {self.symbol}, choose a column from 0 to 6!: ")

class Board:
    def __init__(self):
        self.rows = 6
        self.columns = 7
        self.grid = []
        for _ in range(self.rows):
            row = []
            for _ in range(self.columns):
                row.append("0")
            self.grid.append(row)

    def display(self): #create game board
        for row in self.grid:
            print(" ".join(row))
        print("-" * (2 * self.columns - 1))
        print("0 1 2 3 4 5 6\n")

    def discDrop(self, column, symbol):
        available_row = -1
        for i in range(self.rows):
            if self.grid[i][column] == "0":
                available_row = i # keep updating to latest row
        if available_row != -1:
            self.grid[available_row][column] = symbol
            return True
        else:
            print("Column full!")
            return False

    def full(self): # define a tie game
        for row in self.grid:
            for cell in row:
                if cell == "0":
                    return False
        return True

    def winner(self, symbol): #define a winner
        # Horizontal win
        for row in self.grid:
            for col in range(self.columns - 3):
                if row[col] == symbol and row[col + 1] == symbol and row[col + 2]
                    return True

```

```

# Vertical win
for col in range(self.columns):
    for row in range(self.rows - 3):
        if (self.grid[row][col] == symbol and
            self.grid[row + 1][col] == symbol and
            self.grid[row + 2][col] == symbol and
            self.grid[row + 3][col] == symbol):
            return True

# Diagonal win (from bottom left to top right)
for row in range(self.rows - 3):
    for col in range(self.columns - 3):
        if (self.grid[row][col] == symbol and
            self.grid[row + 1][col + 1] == symbol and
            self.grid[row + 2][col + 2] == symbol and
            self.grid[row + 3][col + 3] == symbol):
            return True

# Diagonal (from top left to bottom right)
for row in range(3, self.rows):
    for col in range(self.columns - 3):
        if (self.grid[row][col] == symbol and
            self.grid[row - 1][col + 1] == symbol and
            self.grid[row - 2][col + 2] == symbol and
            self.grid[row - 3][col + 3] == symbol):
            return True
return False

class Connect4Game: # put all pieces of the game together
    def __init__(self):
        self.players = [Player("X"), Player("Y")]
        self.board = Board()
        self.current_player_index = 0
    def switch_player(self):
        self.current_player_index = 1 - self.current_player_index
    def play(self):
        print("Lets play Connect 4!\n")
        self.board.display()
        while True:
            player = self.players[self.current_player_index]
            try:
                move = int(player.getMove())
            except ValueError:
                print("Enter a valid number from 0 tp 6!")
                continue
            if not self.board.discDrop(move, player.symbol):
                continue
            self.board.display()
            if self.board.winner(player.symbol):

```

```
        print(f"{player.symbol} wins!")
        break
    if self.board.full():
        print("TIE!")
        break
    self.switch_player()

# Run game
if __name__ == "__main__":
    game = Connect4Game()
    game.play()
```