



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
САМАРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Институт автоматизации и информационных технологий
Кафедра «Информатика и Вычислительная техника»

ОТЧЕТ
о выполнении лабораторной работы №2

по дисциплине Компьютерные средства искусственного интеллекта

на тему Крестики-нолики

Преподаватель

А.А. Тюгашев

(должность)

(подпись)

(дата)

(инициалы, фамилия)

К.В. Портнов

(должность)

(подпись)

(дата)

(инициалы, фамилия)

Студенты

4-ИАИТ-119

Е.А. Щаев

(группа)

(подпись)

(дата)

(инициалы, фамилия)

Самара 2024 г.

Цель работы. Разработать консольное приложение: бота для игры в Крестики-нолики

Программный код на языке Python:

```
import math

# Вывод поля
def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)

# Проверка на победу
def check_winner(board, player):
    for row in board:
        if all(cell == player for cell in row):
            return True
    for col in range(3):
        if all(board[row][col] == player for row in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player
for i in range(3)):
        return True
    return False

# Проверка на ничью
def is_draw(board):
    return all(cell != " " for row in board for cell in row)

# Алгоритм минимакс
def minimax(board, is_maximizing):
    if check_winner(board, "O"):
        return 1
    if check_winner(board, "X"):
        return -1
    if is_draw(board):
        return 0

    if is_maximizing:
        best_score = -math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "O"
                    score = minimax(board, False)
                    board[i][j] = " "
                    best_score = max(score, best_score)
            return best_score
    else:
        best_score = math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "X"
                    score = minimax(board, True)
                    board[i][j] = " "
                    best_score = min(score, best_score)
        return best_score
```

```

# Лучший ход
def best_move(board):
    best_score = -math.inf
    move = None
    for i in range(3):
        for j in range(3):
            if board[i][j] == " ":
                board[i][j] = "O"
                score = minimax(board, False)
                board[i][j] = " "
                if score > best_score:
                    best_score = score
                    move = (i, j)
    return move

# Запуск игры
def tic_tac_toe(step):
    board = [[" " for i in range(3)] for i in range(3)]
    print("Игрок играет за 'X', а бот за 'O'.")

    # Первый ход компьютера
    if not step:
        print("Бот ходит первым")
        board[1][1] = "O"
    else:
        print("Ваш первый ход")

    print_board(board)

    while True:
        # Ход игрока
        try:
            move = int(input("Введите номер клетки (1-9): ")) - 1
            row, col = divmod(move, 3)
            if board[row][col] != " ":
                print("Эта клетка уже занята, попробуйте другую.")
                continue
            board[row][col] = "X"
        except (ValueError, IndexError):
            print("Неверный ввод. Введите число от 1 до 9.")
            continue

        # Проверка победителя или ничьей после хода игрока
        print_board(board)
        if check_winner(board, "X"):
            print("Поздравляем! Вы победили!")
            break
        elif is_draw(board):
            print("Ничья!")
            break

        # Ход бота
        print("Ход бота:")
        bot_move = best_move(board)
        if bot_move:
            board[bot_move[0]][bot_move[1]] = "O"
            print_board(board)

        # Проверка победителя или ничьей после хода бота
        if check_winner(board, "O"):
            print("Бот победил!")
            break
        elif is_draw(board):
            print("Ничья!")
            break

```

Пример работы программы:

Рисунок 1 – Начало игры

Рисунок 2 – Ничья

```
yes
Игрок играет за 'X', а бот за 'O'.
Бот ходит первым
  |  |
-----
  | O |
-----
  |  |
-----
Введите номер клетки (1-9):
```

Рисунок 3 – Первый ход бота

```
Ход бота:
X | O |
-----
O | O | O
-----
X | X |
-----
Бот победил!
Продолжаем? yes/no
```

Рисунок 4 – Победа бота

Вывод: Разработали приложение и бота для игры в Крестики-нолики.