МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информационные технологии и общенаучные дисциплины»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**на лабораторную работу**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Разработка компьютерной игры «крестики-нолики»**

Р.02069337. №23/36-9 ТЗ-00

Листов 19

|  |
| --- |
|  |
| Руководитель разработки: |
| кандидат технических наук, доцент |
| Шишкин Вадим Викторинович |
| «\_\_\_» \_\_\_\_\_\_\_\_ 202 г. |
|  |
| Исполнитель: |
| студент гр. АИСТбд-21 |
| Пруцков Ярослав Александрович |
| «\_\_\_» \_\_\_\_\_\_\_\_ 202 г. |

2024

**Содержание**

|  |  |
| --- | --- |
| Аннотация................................................................................................................. | 3 |
| Техническое задание................................................................................................ | 4 |
| Пояснительная записка............................................................................................ | 9 |
| Руководство программиста...................................................................................... | 13 |
| Текст программы...................................................................................................... | 17 |
|  |  |
|  |  |

**Аннотация**

В этом лабораторном отчете подробно описана реализация игры «Крестики-нолики» на языке Python с искусственным интеллектом, использующим Tkinter для графического пользовательского интерфейса и алгоритм Minimax с альфа-бета-отсечением для принятия решений ИИ. В отчете рассматривается реализация основной игровой логики, стратегии искусственного интеллекта и взаимодействия с пользователем.

**Текст программы**

|  |
| --- |
| import tkinter as tk  import tkinter.messagebox as messagebox  from tkinter import simpledialog  def check\_win(player):  for i in range(3):  if all(buttons[i \* 3 + j]["text"] == player for j in range(3)):  return True  if all(buttons[i + j \* 3]["text"] == player for j in range(3)):  return True  if all(buttons[i]["text"] == player for i in [0, 4, 8]):  return True  if all(buttons[i]["text"] == player for i in [2, 4, 6]):  return True  return False  def check\_draw():  return all(button["text"] != "" for button in buttons)  def button\_click(row, col):  global current\_player  index = row \* 3 + col  if buttons[index]["text"] == "":  buttons[index]["text"] = current\_player  if check\_win(current\_player):  messagebox.showinfo("Победа!", f"Игрок {current\_player} победил!")  disable\_buttons()  game\_over()  elif check\_draw():  messagebox.showinfo("Ничья!", "Ничья!")  disable\_buttons()  game\_over()  else:  current\_player = "O" if current\_player == "X" else "X"  label.config(text=f"Ход игрока: {current\_player}")  if current\_player == bot\_player:  bot\_move()  current\_player = player\_choice  label.config(text=f"Ход игрока: {current\_player}")  def disable\_buttons():  for button in buttons:  button.config(state=tk.DISABLED)  def bot\_move():  best\_score = -float('inf')  best\_move = None  for i in range(9):  if buttons[i]["text"] == "":  buttons[i]["text"] = bot\_player  score = alphabeta(buttons, 0, -float('inf'), float('inf'), False)  buttons[i]["text"] = ""  if score > best\_score:  best\_score = score  best\_move = i  buttons[best\_move]["text"] = bot\_player  if check\_win(bot\_player):  messagebox.showinfo("Победа!", f"Бот ({bot\_player}) победил!")  disable\_buttons()  game\_over()  elif check\_draw():  messagebox.showinfo("Ничья!", "Ничья!")  disable\_buttons()  game\_over()  def alphabeta(board, depth, alpha, beta, maximizing\_player):  if check\_win(bot\_player):  return 10  if check\_win(player\_choice):  return -10  if check\_draw():  return 0  if maximizing\_player:  max\_eval = -float('inf')  for i in range(9):  if board[i]["text"] == "":  board[i]["text"] = bot\_player  eval = alphabeta(board, depth + 1, alpha, beta, False)  board[i]["text"] = ""  max\_eval = max(max\_eval, eval)  alpha = max(alpha, eval)  if alpha >= beta:  return beta  return max\_eval  else:  min\_eval = float('inf')  for i in range(9):  if board[i]["text"] == "":  board[i]["text"] = "X" if bot\_player == "O" else "O"  eval = alphabeta(board, depth + 1, alpha, beta, True)  board[i]["text"] = ""  min\_eval = min(min\_eval, eval)  beta = min(beta, eval)  if alpha >= beta:  return alpha  return min\_eval  def game\_over():  result = messagebox.askyesno("Игра окончена", "Хотите сыграть ещё раз?")  if result:  get\_player\_choice()  else:  root.destroy()  def start\_new\_game():  global current\_player, bot\_player, buttons  for button in buttons:  button['text'] = ""  button.config(state=tk.NORMAL)  bot\_player = "O" if player\_choice == "X" else "X"  if player\_choice == "O":  current\_player = bot\_player  bot\_move()  current\_player = player\_choice  label.config(text=f"Ход игрока: O")  else:  current\_player = "X"  label.config(text=f"Ход игрока: X")  def get\_player\_choice():  global player\_choice, current\_player  root.withdraw()  player\_choice = simpledialog.askstring("Выбор игрока", "Выберите, чем хотите играть (X или O):", initialvalue="X").upper()  if player\_choice not in ("X", "O"):  messagebox.showerror("Ошибка", "Неверный ввод. Пожалуйста, введите X или O.")  get\_player\_choice()  root.deiconify()  start\_new\_game()  def center\_window(window):  window.update\_idletasks()  width = window.winfo\_width()  height = window.winfo\_height()  screen\_width = window.winfo\_screenwidth()  screen\_height = window.winfo\_screenheight()  x = (screen\_width - width) // 2  y = (screen\_height - height) // 2  window.geometry(f"{width}x{height}+{x}+{y}")  root = tk.Tk()  root.title("Крестики-нолики")  buttons = []  bot\_player = "O"  label = tk.Label(root, text=f"Ход игрока: ", font=("Helvetica", 16))  label.grid(row=0, column=0, columnspan=3, pady=10)  for i in range(3):  for j in range(3):  button = tk.Button(root, text="", width=10, height=5, font=("Helvetica", 32), command=lambda row=i, col=j: button\_click(row, col))  button.grid(row=i + 1, column=j, padx=5, pady=5)  buttons.append(button)  get\_player\_choice()  center\_window(root)  root.mainloop() |