**Введение**

Данная пояснительная записка описывает разработку приложения “Крестики-нолики с ИИ”, реализующего классическую игру “Крестики-нолики” с использованием алгоритма Minimax для искусственного интеллекта. Приложение разработано на языке программирования Python с использованием библиотеки Tkinter для создания графического интерфейса. Цель работы – демонстрация реализации алгоритма Minimax с альфа-бета отсечением для создания игры с интеллектуальным оппонентом.

**1. Проектная часть**

**1.1. Постановка задачи**

Задача заключается в разработке приложения “Крестики-нолики с ИИ” в соответствии с требованиями технического задания.

**1.2. Математическая модель**

В основе алгоритма ИИ лежит алгоритм Minimax с альфа-бета отсечением. Minimax – это рекурсивный алгоритм поиска в дереве игры, позволяющий оценить все возможные варианты развития игры и выбрать ход, максимизирующий выигрыш (или минимизирующий проигрыш) для ИИ. Альфа-бета отсечение используется для оптимизации процесса поиска, отсекая ветви дерева игры, которые заведомо не приведут к лучшему результату. Функция alphabeta в представленном коде реализует данный алгоритм.

**1.3. Архитектура и алгоритмы**

1.3.1 Архитектура

Приложение состоит из трех основных модулей:

Модуль графического интерфейса (GUI): Создан с помощью Tkinter, отвечает за визуальное представление игрового поля и обработку пользовательского ввода. Функция start\_new\_game инициализирует GUI.

Модуль игровой логики: Реализует правила игры “Крестики-нолики”, включая проверку условий победы, поражения и ничьей. Функция check\_win определяет победителя, а check\_draw проверяет не закончилась ли игра ничьей.

Модуль ИИ: Включает в себя реализацию алгоритма Minimax с альфа-бета отсечением. Функция bot\_move выполняет ход ИИ.

1.3.2 Алгоритм Minimax с альфа-бета отсечением

Алгоритм Minimax с альфа-бета отсечением рекурсивно оценивает все возможные ходы, начиная с текущего состояния игры. Для каждого хода вычисляется оценка позиции с помощью функции alphabeta. Алгоритм выбирает ход с наилучшей оценкой для максимизирующего игрока (ИИ). Альфа-бета отсечение позволяет сократить количество оцениваемых позиций, повышая эффективность алгоритма.

**1.4. Тестирование**

1.4.1 Описание интеллектуальной карты приложения

Тестирования были проведены: победа ИИ, ничья, обработка некорректного ввода.

1.4.2 Цель тестирования

Проверка корректности работы всех модулей приложения, включая алгоритм Minimax, и отсутствие критических ошибок.

1.4.3 Методика тестирования

Использовалось ручное тестирование различных сценариев игры, включая победы ИИ и ничью.

1.4.4 Проведенные тесты

Были проведены полноценные игры с сильной позицией игрока (проверка, что бот не может проиграть в такой ситуации), а также с сильной позицией бота (в этом случае бот выигрывал)

1.4.5 Выводы

Тестирование показало корректную работу приложения и реализованного алгоритма Minimax.

**2. Источники**

1. Шишкин В.В., Афонин Д.С. Разработка логических компьютерных игр с графическим интерфейсом в среде Питон. - УлГТУ, 2023
2. Воронина В.В., Шишкин В.В. Компьютерная графика.-УлГТУ, 2023