МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Измерительно-вычислительные комплексы»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | |  | | Тестовая документация  на лабораторную работу №10  по дисциплине «Алгоритмы и структуры данных»  Тема «Разработка компьютерной игры Крестики-нолики» | | | | | | |
|  | | |  | |  | | | |
|  | | Исполнитель  студент гр. ИСТбд-22  Камчаров М.С.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |
|  | | 2024 | | | | | | |

**Тестовая документация для игры "Крестики-нолики".**

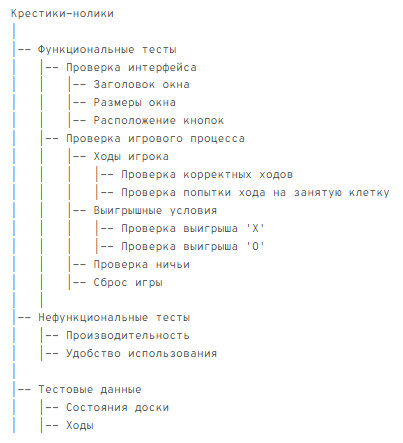
**1. Назначение программы:**

Программа "Крестики-нолики" позволяет играть в классическую игру "Крестики-нолики" против компьютера, который использует алгоритм Minimax для оптимальных ходов.

**2. Описание тестируемых функций:**

* init(self, master): Инициализирует игру, создает игровую доску, устанавливает начальные значения и элементы интерфейса.
* click\_button(self, row, col): Обрабатывает клик игрока по кнопке, обновляет состояние доски и проверяет на победу или ничью.
* bot\_move(self): Выполняет ход бота с использованием алгоритма Minimax.
* minimax(self, board, is\_maximizing\_player): Рекурсивная функция для алгоритма Minimax, которая вычисляет оптимальный ход для текущего игрока.
* check\_win(self, player): Проверяет наличие выигрышной комбинации для заданного игрока.
* check\_draw(self): Проверяет, закончилась ли игра ничьей.
* show\_winner(self, winner): Выводит сообщение о победителе или ничьей.
* reset\_game(self): Сбрасывает игру в исходное состояние.

1. **Mind Map**



1. **Чек-лист**

* Функциональность:
  + Проверка на победу/ничью для всех возможных комбинаций.
  + Корректное отображение выигрыша/ничьей.
  + Правильная работа алгоритма Minimax (оптимальные ходы бота).
  + Правильная работа кнопки "Сброс".
  + Проверка на недопустимые ходы (ход на занятую клетку).
* Интерфейс:
* Кнопки должны реагировать на клики.
* Информационное поле отображает текущего игрока.
* Размер окна подходит для удобного использования.
* Стабильность:
* Отсутствие ошибок при запуске и во время игры.
* Отсутствие зависаний.
* Юзабилити:
* Простая и понятная логика игры.
* Интуитивно понятный интерфейс.

1. **Набор тест-кейсов для игры"Крестики-нолики".**

**1. Функциональность**

**1.1 Победа игрока**

Победа игрока "X" по горизонтали

* Описание: Игрок "X" делает ходы, чтобы получить три "X" в одной строке.
* Шаги:
  1. Игрок "X" ставит "X" в клетку (0,0).
  2. Компьютер ставит "O" в клетку (1,1).
  3. Игрок "X" ставит "X" в клетку (0,1).
  4. Компьютер ставит "O" в клетку (2,0).
  5. Игрок "X" ставит "X" в клетку (0,2).
* Ожидаемый результат: Отображается сообщение "Победил X".

Победа игрока "O" по вертикали

* Описание: Игрок "O" (компьютер) делает ходы, чтобы получить три "O" в одном столбце.
* Шаги:
  1. Игрок "X" ставит "X" в клетку (0,0).
  2. Компьютер ставит "O" в клетку (0,1).
  3. Игрок "X" ставит "X" в клетку (1,0).
  4. Компьютер ставит "O" в клетку (1,1).
  5. Игрок "X" ставит "X" в клетку (2,2).
  6. Компьютер ставит "O" в клетку (2,1).
* Ожидаемый результат: Отображается сообщение "Победил O".

Победа игрока "X" по диагонали

* Описание: Игрок "X" делает ходы, чтобы получить три "X" по диагонали.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (1,1).
3. Игрок "X" ставит "X" в клетку (1,0).
4. Компьютер ставит "O" в клетку (2,2).
5. Игрок "X" ставит "X" в клетку (2,0).

* Ожидаемый результат: Отображается сообщение "Победил X".

**1.2 Победа компьютера**

Победа компьютера "O" по горизонтали

* Описание: Компьютер "O" делает ходы, чтобы получить три "O" в одной строке.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (0,1).
3. Игрок "X" ставит "X" в клетку (1,0).
4. Компьютер ставит "O" в клетку (0,2).
5. Игрок "X" ставит "X" в клетку (2,1).

* Ожидаемый результат: Отображается сообщение "Победил O".

Победа компьютера "X" по вертикали

* Описание: Компьютер "X" (в роли "O") делает ходы, чтобы получить три "O" в одном столбце, когда игрок "X" первым ходом ставит "X".
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (1,0).
3. Игрок "X" ставит "X" в клетку (0,1).
4. Компьютер ставит "O" в клетку (2,0).

* Ожидаемый результат: Отображается сообщение "Победил O".

Победа компьютера "O" по диагонали

* Описание: Компьютер "O" делает ходы, чтобы получить три "O" по диагонали.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (1,1).
3. Игрок "X" ставит "X" в клетку (0,2).
4. Компьютер ставит "O" в клетку (2,0).

* Ожидаемый результат: Отображается сообщение "Победил O".

**1.3 Ничья**

* Описание: Игрок "X" и компьютер "O" делают ходы, но никто не может выстроить выигрышную комбинацию.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (0,1).
3. Игрок "X" ставит "X" в клетку (1,0).
4. Компьютер ставит "O" в клетку (1,1).
5. Игрок "X" ставит "X" в клетку (0,2).
6. Компьютер ставит "O" в клетку (1,2).
7. Игрок "X" ставит "X" в клетку (2,0).
8. Компьютер ставит "O" в клетку (2,1).
9. Игрок "X" ставит "X" в клетку (2,2).

* Ожидаемый результат: Отображается сообщение "Ничья".

**1.4 Недопустимый ход**

Недопустимый ход на занятую клетку

* Описание: Игрок пытается сделать ход на клетку, которая уже занята.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (0,1).
3. Игрок "X" пытается поставить "X" в клетку (0,0).

* Ожидаемый результат: Ход игнорируется, игрок "X" может сделать новый ход на свободную клетку.

**1.5 Кнопка "Сброс"**

* Описание: Нажатие на кнопку "Сброс" во время игры.
* Шаги:

1. Игрок "X" ставит "X" в клетку (0,0).
2. Компьютер ставит "O" в клетку (0,1).
3. Игрок "X" нажимает кнопку "Сброс".

* Ожидаемый результат: Игра сбрасывается в исходное состояние, все клетки пустые, информационное поле отображает "Ход: X".

**2. Интерфейс**

**2.1 Размеры окна**

Проверка размеров окна

* Описание: Проверка размеров окна при запуске программы.
* Шаги: Запустить программу.
* Ожидаемый результат: Окно имеет размеры 400x500 пикселей.

**2.2 Кнопки**

Реакция кнопок на клики

* Описание: Кнопки реагируют на клики, меняя текст и обновляя состояние доски.
* Шаги:

1. Кликнуть по любой свободной клетке на доске.

* Ожидаемый результат: Текст кнопки меняется на "X" (или "O"), состояние доски обновляется.

**2.3 Информационное поле**

Отображение информации о текущем ходе

* Описание: Информационное поле отображает текущего игрока ("Ход: X" или "Ход: O").
* Шаги:

1. Запустить программу.

* Ожидаемый результат: Информационное поле отображает "Ход: X".

1. Сделать ход игроком "X".

* Ожидаемый результат: Информационное поле отображает "Ход: O".

**3. Стабильность**

**3.1 Запуск**

* Описание: Проверка запуска программы без ошибок.
* Шаги: Запустить программу.
* Ожидаемый результат: Программа запускается без ошибок, появляется окно игры.

**3.2 Работа во время игры**

* Описание: Проверка отсутствия ошибок во время игры.
* Шаги: Провести несколько ходов, включая выигрыш, ничью и недопустимые ходы.
* Ожидаемый результат: Программа работает без ошибок, не зависает, не падает, не выдает некорректных сообщений.

**4. Алгоритм Minimax**

**Оптимальные ходы компьютера**

* Описание: Проверка, что компьютер "O" делает оптимальные ходы, исходя из алгоритма Minimax.
* Шаги: Провести серию игр, где игрок "X" делает различные ходы, наблюдая за ходами компьютера "O".
* Ожидаемый результат: Ходы компьютера "O" соответствуют оптимальной стратегии Minimax, т.е. он не делает ходов, которые заведомо ведут к проигрышу.

Примечание: Тестирование алгоритма Minimax является сложной задачей, требующей анализа большого количества сценариев игры и сравнения ходов компьютера с теоретически оптимальными ходами.