**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Измерительно-вычислительные комплексы»

Курсовая работа

По дисциплине «Алгоритмы и структуры данных»

**Инв. № подл.**

**Подп. и дата**

**Взам. инв. №**

**Инв. № дубл.**

**Подп. и дата**

Тема Компьютерная логическая игра «Скифские шашки»

Пояснительная записка

Р.02069337.21/837-13 ТЗ-01

Листов 7

**Руководитель разработки**:

доцент каф. ИВК, к.т.н., доцент

*Шишкин Вадим Викторинович*

« » 2022 г.

**Исполнитель**:

студент гр. ИСТбд-22

*Молофеев Иван Сергеевич*

« » 2022 г.

**2022**

**Введение**

Наименование игры: Скифские шашки.

Условное наименование: COURSE.

Базовые правила «Скифских шашек» совпадают с правилами русских. Но есть и отличия: доска 9x9, у каждого игрока есть новая шашка – вождь. Она ходит, как обычная шашка, но, если кто-то съедает вождя противника – он автоматически выигрывает, потому что вождь – главная шашка для каждого.

**1. Проектная часть**

**1.1 Постановка задачи на разработку приложения**

Определяется общей постановкой задачи в задании на курсовую работу.

**1.2 Математические методы**

Математические методы не применялись.

**1.3 Алгоритмы**

**1.3.1 Алгоритм проверка хода игрока**

В начале алгоритма надо создать список возможных ходов для игроков, далее проверить ход на возможность, сначала для белых потом для черных. В результате либо делаем ход или откатываем все до попытки этого хода.



Рисунок 1 – Проверка хода игрока

**1.3.2 Алгоритм составление списка ходов**

Данных алгоритмов будет два, бля белых и черных, рассмотрим данных алгоритм только для белых, так как они идентичны. В начале алгоритма следует составить список обязательных ходов, если список будет пустым проверить остальные всевозможные ходы и вернуть список.



Рисунок 2 – Составление списка ходов

**1.3.3 Алгоритм составление наличия обязательных ходов**

Данный алгоритм будет включать в себя два алгоритма, первый будет проверять и создавать список всех обязательных ходов, второй проверять каждую клетку поля. Этот алгоритм составляет список рандомных ходов.

Составление списка обязательных ходов будет заключаться в проверке каждой клетке игрового доски на обязательный ход.

Проверка наличия обязательных ходов будет проверять возможность хода, определять правильность хода и возвращать ход если он обязателен.

Как и в прошлом алгоритме будет рассматривать возможность для белых шашек. Для черных алгоритм будет подобным.



Рисунок 3 – Составление списка обязательных ходов

Рисунок 4 – Проверка наличия обязательных ходов

**1.3.4 Алгоритм проверка наличия всех ходов**

Данный алгоритм будет составлять список всех возможных ходов, опять делаем это для белых, для черных алгоритм идентичный.



Рисунок 5 – Составление всех ходов

**1.4 Тестирование**

Тестирование программы – это этап, на котором проверяется, как ведет себя программа на как можно большем количестве входных наборов данных, в том числе и на заведомо неверных.

Тестирование – это процесс выполнения программы с целью выявления ошибок. Процесс разработки ПО предполагает три стадии тестирования:

* автономное тестирование – это тестирование компонентов ПО;
* комплексное тестирование;
* системное (оценочное) тестирование – тестирование на соответствие основным критериям качества.

Принципы тестирования:

* избегать тестирования программы самим автором;
* предполагаемые результаты должны быть известны до тестирования;
* необходимо изучать результаты каждого теста;
* необходимо проверять действие программы на неверных данных.

Существует два принципиально различных подхода к формированию тестов:

* структурный – известна структура тестируемого ПО, в том числе его алгоритмы. Тесты строят так, чтобы проверить правильность реализации заданной логики в ходе программы (белый ящик);
* функциональный – структура ПО неизвестна. Тесты строят по функциональным спецификациям (черный ящик; подход, управляемый данными).

При проведении тестирования следует помнить, что никакое тестирование не может доказать отсутствие ошибок в ПО. Удачным считают тест, который обнаруживает хотя бы одну ошибку. Вероятность наличия необнаруженных ошибок пропорциональна количеству уже найденных ошибок в программе.

**1.4.1 Методика проведения и результаты тестирования**

При разработке данной программы были допущены следующие синтаксические ошибки:

* неправильное использование операторов присваивания;
* неверное объявление циклов.

При тестировании были выполнены следующие принципы:

* тщательность подбора данных для теста программы, не только для правильных входных данных, но и для неправильных;
* доскональное изучение результатов тестирования.

**1.4.2 Отладка выявленных ошибок, обнаруженных при тестировании**

Во время выполнения отладки, были использованы следующие методы:

* анализ кода без исполнения программы;
* запуск программы из под отладчика.

Все синтаксические ошибки были исправлены при компиляции проекта, учитывая синтаксические особенности среды программирования. Также в программном средстве возникали логические ошибки, которые были исправлены путем пересмотра кода алгоритмов и его последующего исправления.

Отладка ряда ошибок прошла успешно, были внесены изменения в интерфейс программы, а также были предусмотрены некоторые исключительные ситуации.

**2. Источники, использованные при разработке**

1. Python Checkers AI Tutorial Part 2 — Реализация и визуализация (Minimax) // YouTube URL: Python Checkers AI Tutorial Part 2 — Реализация и визуализация (Minimax) – YouTube (дата обращения: 25.12.2022).

2. Algorithms Explained – minimax and alpha-beta pruning // YouTube URL: https://www.youtube.com/watch?v=l-hh51ncgDI (дата обращения: 26.12.2022)

3. Checkers-Python // Medium URL: https://medium.com/analytics-vidhya/checkers-python-eff2786b985b (дата обращения: 22.12.2022)

4. Игра Шашки на Питоне // CyberForum URL: https://www.cyberforum.ru/blogs/914198/blog4540.html (дата обращения: 21.12.2022).

5. Tkinter — создание графического интерфейса в Python // python-scripts URL: https://python-scripts.com/tkinter (дата обращения: 23.12.2022).

6. Введение в Tkinter // Habr URL: https://habr.com/ru/post/133337/ (дата обращения: 23.12.2022).

7. Введение в Python // Habr URL: https://habr.com/ru/post/450474/ (дата обращения: 23.12.2022).

8. Изучаем Декораторы в Python // python-scripts URL: https://python-scripts.com/decorators (дата обращения: 02.01.2023).

9. Функции \_\_str \_\_() и \_\_repr \_\_() в Python // pythonim URL: https://pythonim.ru/string/funktsii-str-repr-python (дата обращения: 03.01.2023).

10. Путь к пониманию декораторов в Python // Habr URL: https://habr.com/ru/company/wunderfund/blog/657355/ (дата обращения: 03.01.2023).

11. Python Tkinter // YouTube URL: https://www.youtube.com/playlist?list=PLQAt0m1f9OHsd6U5okp1XLoYyQR0oBjMM (дата обращения: 04.01.2023).

12. Минимакс с Альфа-Бета-обрезкой в Python // pythobyte URL: https://pythobyte.com/minimax-and-alpha-beta-pruning-in-python-fe960495/ (дата обращения: 02.01.2023).

**3. Приложения**

1)Приложение 1 – «Техническое задание».

2)Приложение 2 – «Руководство программиста».