

Министерство науки и высшего образования РФ
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по курсу «Программирование»
на тему «Разработка программы сложной структуры методом нисходящего
программирования. Игра “Змейка”»

Выполнил: ст. гр. 25ВВВ4

496.
13.05.26

Кашкин И. А.

Принял:

246

Кандидат т.н. Токарев А. Н.

Пенза 2026

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"



Зав. кафедрой ВТ

М.А. Митрохин

«___» _____ 20__ г

ЗАДАНИЕ

на курсовое проектирование по курсу

Программирование

Студенту Кашкин Илья Андреевич Группа 25 ВВВ4

Тема проекта: *Разработка программы сложной структуры методом нисходящего программирования.*

Исходные данные (технические требования) на проектирование

Игра "Змейка". Требования:

- Управление движением змейки осуществляется с помощью клавиатуры.
- Система набора очков.
- Должен быть предусмотрен рост и уменьшение размеров змейки.
- Изменение цветовой палитры.
- Сохранение результатов предыдущих игр в файл.

Обязательные требования к программе:

1. Многомодульность.
2. Использование сложных типов данных – структур, списков, файлов.
3. Режим работы видеосистемы – текстовый / графический.
4. Устройство ввода информации – клавиатура, мышь.
5. Пользовательский интерфейс должен быть построен на основе меню и панели инструментов.
6. Наличие заставки.
7. Операционная система MS Windows.
8. Язык программирования – Си и Ассемблер.
9. Среда разработки ПО – Microsoft Visual Studio.

1. Расчетная часть

2. Графическая часть

Схема данных, схема ресурсов системы, схема работы системы, иерархическая структура программы, схема взаимодействия программы.

3. Экспериментальная часть

Отладка программы.

Срок выполнения проекта по разделам

В соответствии с графиком.

Дата выдачи задания "18" 02 2026

Дата защиты проекта “_____” _____

Руководитель _____

Задание получил "25" 02 2026 г.

Студент В. Калинин Иван Андреевич

Содержание

Введение.....	5
Постановка задачи	7
Выбор решения.....	9
Описание разработки программы	11
Отладка и тестирование.....	14
Описание программы.....	15
Руководство пользователя	21
Заключение	24
Список используемых источников	25
Приложение В. Результаты работы программы	26
Заставка	26
Главное меню.....	26
Игровое поле.....	26
Файл результата.....	27

Введение

Компьютерные игры являются одним из наглядных примеров программ, в которых одновременно используются алгоритмы обработки данных, работа с пользовательским вводом, вывод информации на экран, хранение результатов и организация кода по модулям. Даже простая игра требует продуманной структуры: необходимо хранить состояние объектов, обрабатывать события клавиатуры, изменять координаты, проверять столкновения и своевременно обновлять изображение.

В данной курсовой работе рассматривается разработка консольного приложения `The_zmeika`, реализующего игру «Змейка». Пользователь управляет змейкой с помощью клавиш-стрелок, собирает еду, увеличивает длину и набирает очки. В программе предусмотрены два режима сложности: классический режим без дополнительных препятствий и режим с ядом, в котором на поле появляются опасные клетки.

Особенностью программы является использование многомодульной структуры на языке C и ASM. Отдельные файлы отвечают за игровой цикл, отрисовку, объекты поля, сохранение результатов, настройки, интерфейс меню и вспомогательные функции. Такой подход соответствует методу нисходящего программирования: сначала определяется общая логика приложения, затем она разбивается на подзадачи, каждая из которых реализуется отдельным модулем.

Приложение работает в консоли Windows и использует функции WinAPI для управления цветом текста, положением курсора и размерами окна. Для ввода применяются функции `_kbhit()` и `_getch()`, что позволяет реагировать на нажатия клавиш без ожидания ввода строки. Часть вычислительной логики реализована с применением вставок ассемблера, что демонстрирует работу с низкоуровневыми командами процессора в рамках программы на C.

Разрабатываемая программа предназначена не только для демонстрации игрового процесса, но и для закрепления практических навыков

проектирования сложной структуры программы, работы с массивами структур, файлами, консольным интерфейсом и обработкой событий пользователя.

Постановка задачи

Необходимо разработать консольное приложение The_zmeika, реализующее игру «Змейка» для операционной системы Windows. Программа должна иметь понятный текстовый интерфейс, главное меню, выбор режима сложности, игровое поле, отображение текущего счёта и длины змейки.

Основная цель игры заключается в управлении змейкой, которая перемещается по прямоугольному полю, собирает еду и увеличивает свою длину. При столкновении со стеной или собственным телом игра должна завершаться. В режиме повышенной сложности на поле должны появляться ядовитые клетки, уменьшающие очки и длину змейки либо завершающие игру при недостаточной длине.

В ходе разработки необходимо определить структуры данных для хранения координат, тела змейки, ядовитых клеток и записей таблицы результатов. Необходимо использовать массив координат для тела змейки, массив ядовитых клеток, глобальные переменные состояния игры и набор функций для обработки каждого логического блока программы.

Программа должна быть разделена на модули. Модуль main.c должен содержать точку входа, инициализацию консоли и главный цикл меню. Модуль interface.c должен отвечать за заставку, главное меню и выбор сложности. Модуль game.c должен реализовывать игровой цикл. Модули graphics.c, gameObjects.c, gameSaving.c, settings.c и helper.c должны отвечать соответственно за отрисовку, генерацию объектов, работу с файлами результатов, сохранение настроек и вспомогательные операции.

Устройствами ввода-вывода являются клавиатура и экран монитора. Пользователь должен управлять змейкой клавишами-стрелками, завершать игру клавишей X, переключать цветовую схему клавишей C и выбирать пункты меню клавишами-стрелками с подтверждением Enter.

После завершения игры программа должна запрашивать имя игрока и сохранять результат в текстовый файл в каталоге РЕЗУЛЬТАТЫ. Таблица

рекордов должна считываться из файлов, сортироваться по количеству очков и показывать лучшие записи пользователю.

В результате выполнения задачи должно быть создано работоспособное консольное приложение с игровым процессом, меню, сохранением результатов, настройками палитры и модульной структурой исходного кода.

Выбор решения

Для реализации выбран язык C, так как он позволяет напрямую работать с массивами, структурами, файлами и функциями операционной системы. В качестве среды разработки используется Microsoft Visual Studio, проект собран как консольное приложение Windows.

Игровое поле имеет размер 40 на 20 клеток. Границы отображаются символом #, голова змейки — символом O, тело — символом o, еда — символом @, яд — символом %. Для повышения наглядности применяются цветовые палитры консоли.

При запуске программы выполняется настройка кодовой страницы 1251, инициализация генератора случайных чисел, получение дескриптора консоли, скрывание курсора, создание каталога РЕЗУЛЬТАТЫ и загрузка сохранённой цветовой схемы. Затем пользователю показывается заставка и главное меню.

Главное меню содержит пункты: «Начать игру», «Рекорды», «Сменить цвет», «Выход». При выборе начала игры открывается меню сложности. В классическом режиме игрок взаимодействует только с едой и границами поля. В режиме с ядом на поле дополнительно появляются опасные клетки, которые живут ограниченное количество игровых тактов.

Координаты объектов хранятся в структуре Point. Состояние змейки описывает структура Snake, содержащая массив координат тела, длину, направление движения и признак жизни. Ядовитые клетки описаны структурой PoisonCell, содержащей позицию, активность и таймер существования. Для таблицы результатов используется структура ScoreRecord.

Обновление игры выполняется в цикле: проверка клавиатуры, отрисовка поля, обработка логики движения, пауза Sleep(). Скорость движения зависит от длины змейки: чем длиннее змейка, тем меньше задержка между кадрами, но минимальная задержка ограничена 40 миллисекундами.

Для работы с консолью используются функции Windows API:

- SetConsoleTextAttribute();

- SetConsoleCursorPosition();
- SetConsoleCursorInfo();
- SetConsoleScreenBufferSize();
- SetConsoleWindowInfo();
- CreateDirectoryA();
- FindFirstFileA();
- FindNextFileA().

Это позволяет управлять цветами, курсором, размером окна и поиском файлов результатов.

Выбранное решение удобно тем, что основные части программы независимы друг от друга. Игровые объекты, отрисовка, сохранение и интерфейс вынесены в отдельные модули, поэтому каждую часть можно разрабатывать и проверять отдельно.

Описание разработки программы

Разработка программы выполнялась поэтапно. Сначала была определена общая структура приложения: главное меню, выбор режима, игровой цикл, сохранение результатов и таблица рекордов. После этого были выделены основные структуры данных и создан общий заголовочный файл `HeaderSnake.h`.

Список файлов:

- `HeaderSnake.h`: общие константы, структуры, глобальные переменные и прототипы функций.
- `main.c`: запуск приложения, настройка консоли и главный цикл меню;
- `interface.c`: заставка, главное меню и выбор сложности;
- `game.c`: подготовка новой игры и основной игровой цикл;
- `graphics.c`: отрисовка поля и игровая логика;
- `gameObjects.c`: создание еды, создание и обновление яда;
- `gameSaving.c`: сохранение результатов и вывод таблицы рекордов;
- `settings.c`: сохранение и загрузка палитры;
- `helper.c`: вспомогательные функции работы с консолью и проверки координат.

Файл `HeaderSnake.h` содержит подключение библиотек, константы размеров поля, максимальной длины змейки, параметров яда, кодов клавиш, цветов консоли, описания структур `Point`, `Snake`, `PoisonCell` и `ScoreRecord`, а также объявления глобальных переменных и функций всех модулей.

`main.c` — содержит точку входа программы. Здесь выполняется настройка консоли, создание каталога результатов, загрузка настроек, установка размеров окна и основной цикл главного меню. В зависимости от выбора пользователя программа запускает игру, показывает рекорды, переключает палитру или завершает работу.

interface.c — отвечает за заставку, главное меню и меню выбора сложности. Все меню построены по одному принципу: выбранный пункт подсвечивается, перемещение выполняется стрелками, подтверждение — клавишей Enter.

game.c — реализует игровой цикл. В начале новой игры задаются начальные координаты змейки, длина, направление, счёт, состояние еды и яда. Затем программа повторяет цикл обработки клавиатуры, отрисовки, логики и задержки до наступления условия завершения.

graphics.c — содержит функции draw() и logic(). Функция draw() отображает игровое поле, границы, тело змейки, еду, яд и строку состояния. Функция logic() рассчитывает новое положение головы, проверяет столкновения, обрабатывает сбор еды и взаимодействие с ядом. В этой функции используются вставки ассемблера для вычисления направления движения, проверки границ, прибавления и вычитания очков.

gameObjects.c — реализует генерацию еды и яда. Функция spawnFood() выбирает случайную свободную клетку, не занятую змейкой или ядом. Функция trySpawnPoison() создаёт ядовитую клетку при наличии свободного слота и ограничивает количество активного яда. Функция updatePoisons() уменьшает таймеры и удаляет истёкшие клетки.

gameSaving.c — отвечает за сохранение результатов и вывод таблицы рекордов. Каждый результат записывается в отдельный текстовый файл с именем вида «попытка <имя> <номер>.txt». При отображении рекордов программа считывает файлы из папки РЕЗУЛЬТАТЫ, формирует массив записей, сортирует его по убыванию очков и выводит первые десять результатов.

settings.c — сохраняет и загружает выбранную цветовую палитру в файле settings.txt. Благодаря этому при следующем запуске программа восстанавливает последнюю выбранную схему оформления.

helper.c — содержит вспомогательные функции setColor(), gotoxy(), hideCursor(), showCursor(), isOnSnake() и isOnPoison(). Они инкапсулируют работу с консолью и проверку занятости клеток игрового поля.

По порядку разработки можно сделать вывод, что использовался нисходящий подход. Сначала была определена общая задача и основные режимы работы, затем каждая крупная функция была разделена на более мелкие функции и вынесена в отдельный модуль.

Отладка и тестирование

Отладка выполнялась в среде Microsoft Visual Studio. Для проверки использовались запуск программы в режиме отладки, пошаговое выполнение, просмотр значений переменных, проверка содержимого массивов тела змейки и ядовитых клеток, а также анализ файлов результатов.

Основное внимание уделялось корректности движения змейки, обработке клавиш, появлению еды и яда, проверке столкновений, сохранению результата и чтению таблицы рекордов. Также проверялась работа цветowych палитр и восстановление выбранной палитры после перезапуска.

Тестирование включало следующие сценарии:

- запуск программы и отображение заставки;
- навигация по главному меню;
- выбор классического режима;
- выбор режима с ядом;
- сбор еды;
- столкновение со стеной;
- столкновение с телом змейки;
- попадание на яд;
- сохранение результата под введённым именем;
- просмотр таблицы рекордов;
- смена цветовой схемы.

В ходе тестирования были выявлены и исправлены ошибки, связанные с появлением объектов поверх тела змейки, исчезновением яда, сохранением файлов, отображением кириллицы в консоли и корректной обработкой специальных клавиш клавиатуры.

Результатом тестирования стало подтверждение работоспособности основных функций программы: игровой цикл выполняется стабильно, меню реагирует на ввод, рекорды сохраняются в файлы, настройки палитры восстанавливаются, а состояние игры корректно завершается при проигрыше.

Описание программы

Приложение The_zmeika.exe является основным исполняемым файлом программы. После запуска пользователь видит заставку, затем попадает в главное меню. Из главного меню можно начать игру, открыть таблицу рекордов, сменить цветовую палитру или выйти из программы. В таблице 1 описаны элементы управления

Таблица 1 – Описание элементов управления

Действие	Описание
Стрелка вверх	Изменить направление движения вверх, если змейка не движется вниз.
Стрелка вниз	Изменить направление движения вниз, если змейка не движется вверх.
Стрелка влево	Изменить направление движения влево, если змейка не движется вправо.
Стрелка вправо	Изменить направление движения вправо, если змейка не движется влево.
Enter	Подтвердить выбранный пункт меню.
X	Выйти из игры или меню.
C	Сменить цветовую схему во время игры.

Схема взаимодействия программы:

- main.c запускает интерфейс меню;
- interface.c возвращает выбор пользователя;
- game.c запускает игровой цикл;
- graphics.c и gameObjects.c обновляют состояние поля;
- gameSaving.c сохраняет результат, settings.c хранит пользовательские настройки.

На рисунке 1 представлена схема взаимодействия программы.

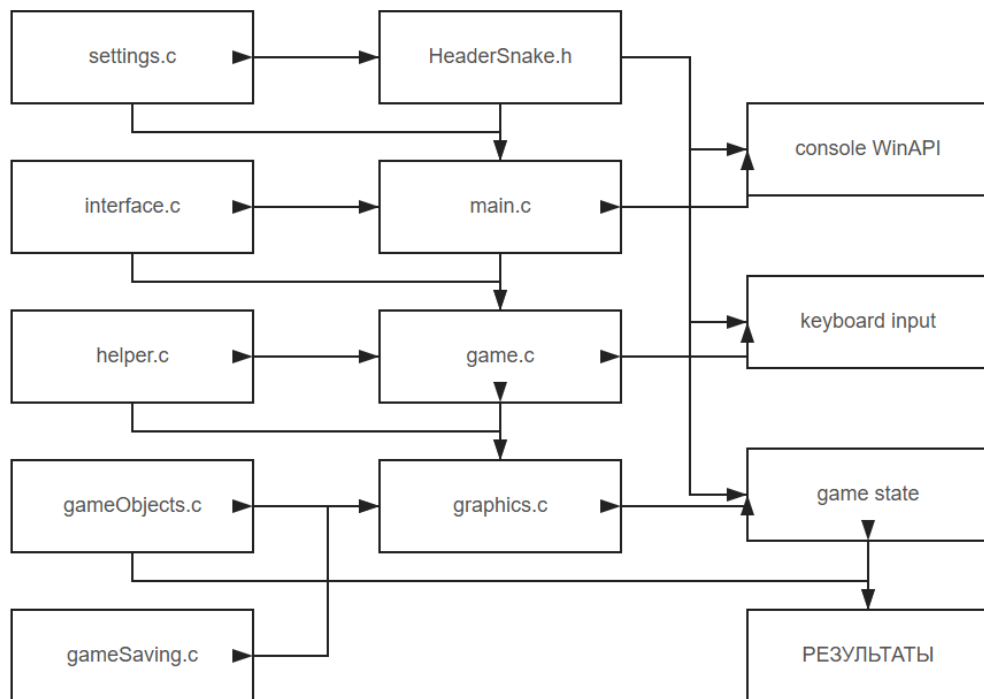


Рисунок 1- Схема взаимодействия программы

На рисунке 2 представлена схема данных.

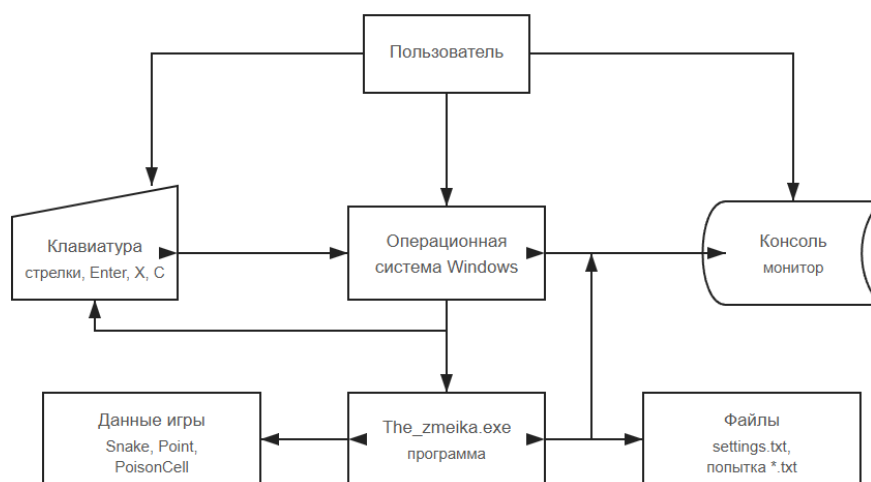


Рисунок 2 - Схема данных

Блок-схема Main представлена на рисунке 3, рисунке 4, рисунке 5.



Рисунок 3 - Функция `main`

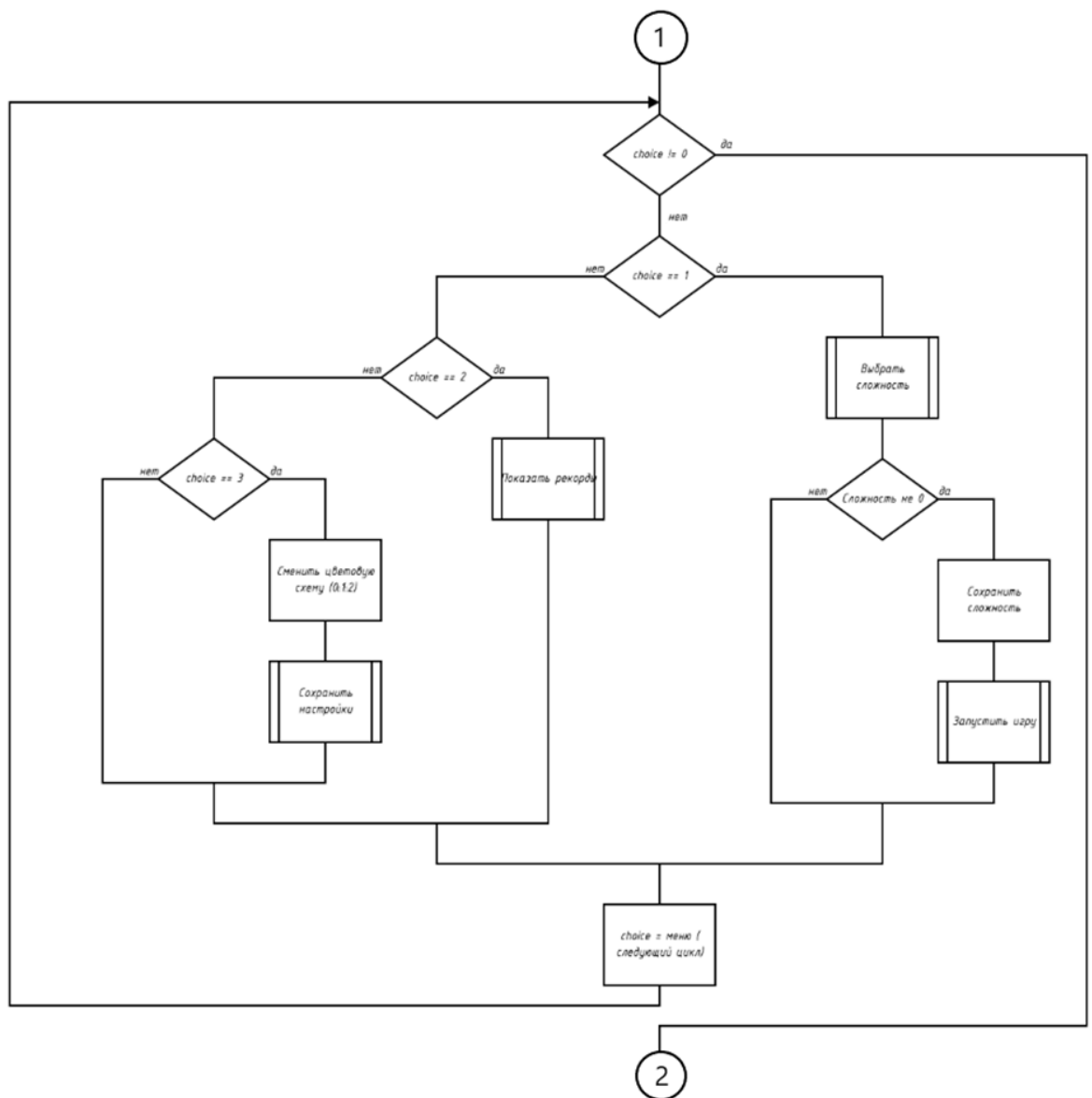


Рисунок 4 - Функция main



Рисунок 5 - Функция main

На рисунке 6 представлена блок-схема ассемблерской вставки из graphics.c.

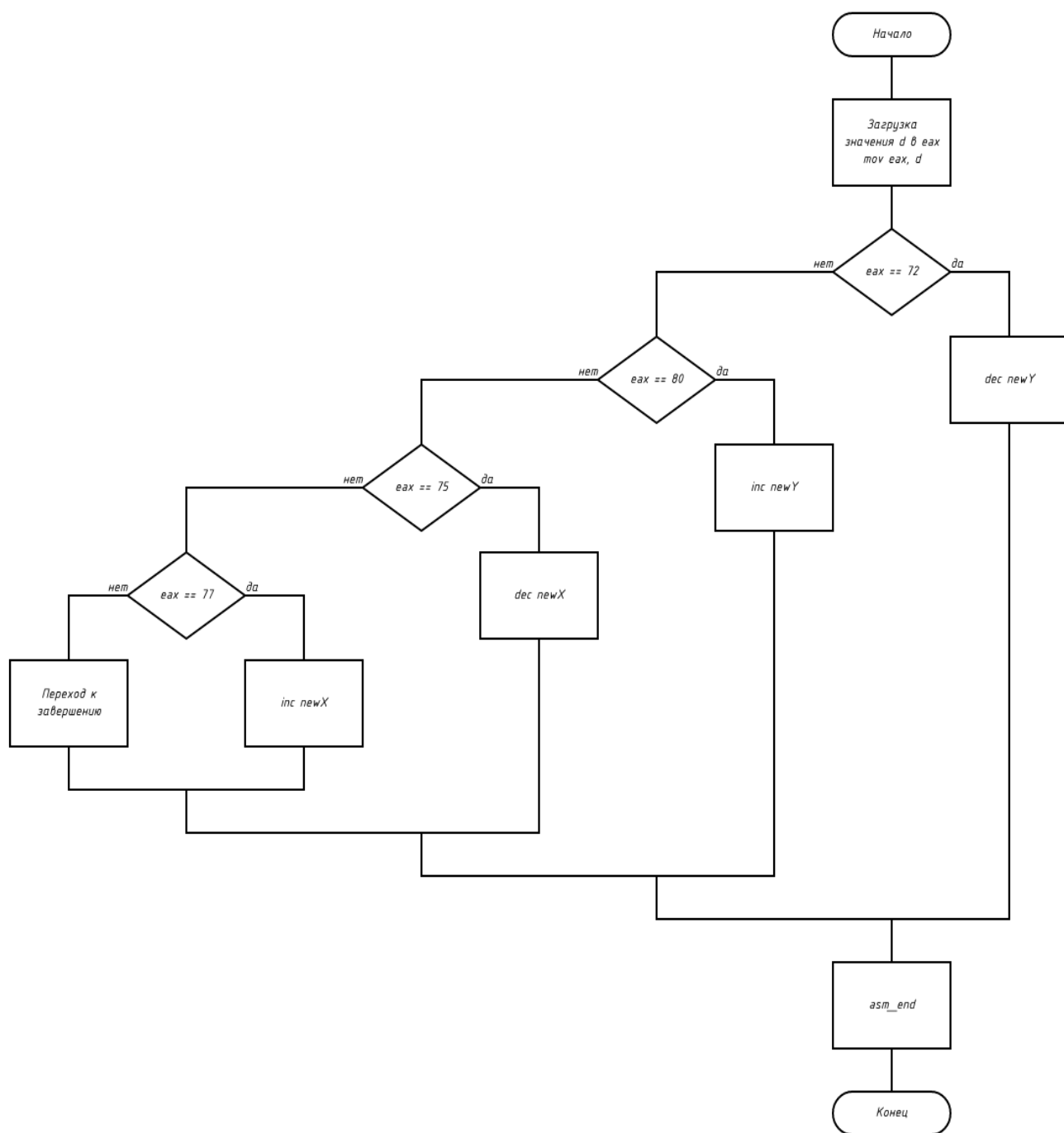


Рисунок 6 - Блок-схема ассемблерской вставки

Руководство пользователя

Программа The_zmeika.exe предназначена для игры в консольную версию «Змейки». Для работы программы необходимо запустить исполняемый файл. После запуска откроется консольное окно с заставкой. Для перехода дальше можно нажать любую клавишу или дождаться окончания заставки.

В главное меню (Рисунок 7) перемещение между пунктами выполняется клавишами-стрелками вверх и вниз. Выбранный пункт выделяется цветом. Для подтверждения выбора необходимо нажать Enter. Для выхода можно выбрать пункт «Выход» или нажать клавишу X.

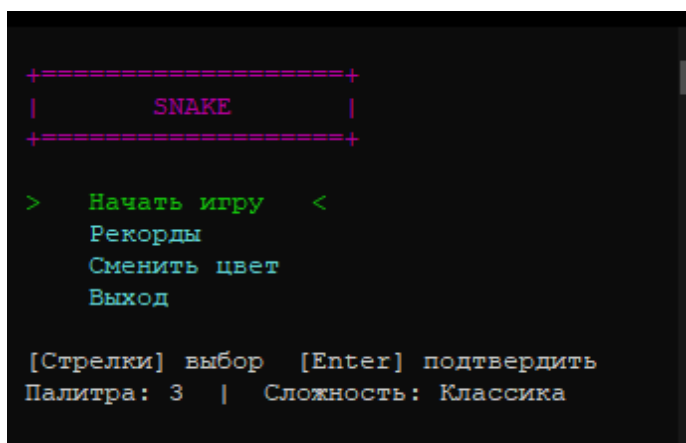


Рисунок 7 - главное меню

Для начала игры необходимо выбрать пункт «Начать игру». После этого откроется меню выбора сложности (Рисунок 8). В режиме «Классика» на поле присутствует только змейка и еда. В режиме «С ядом» дополнительно появляются ядовитые клетки, которые усложняют прохождение.

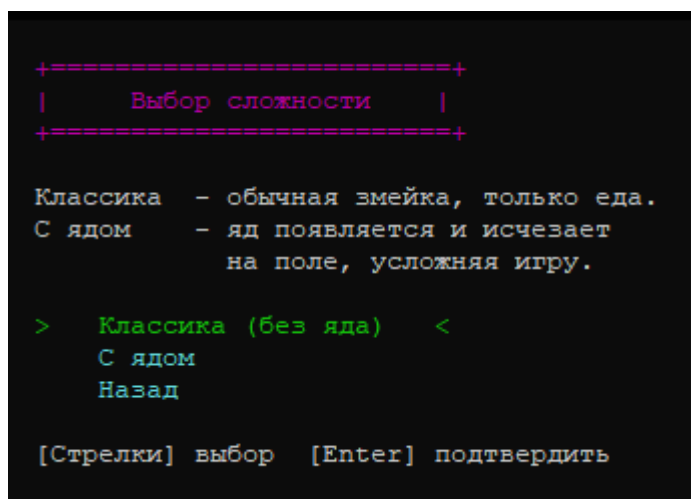


Рисунок 8 - Выбор сложности

Во время игры змейка движется автоматически. Пользователь управляет направлением с помощью клавиш-стрелок (Рисунок 9). Нельзя сразу повернуть в противоположную сторону, так как это привело бы к мгновенному столкновению с телом.

Цель игры — собирать еду, обозначенную символом @. За каждую съеденную еду начисляется 10 очков, а длина змейки увеличивается на 1. После сбора еды новая еда появляется в случайной свободной клетке поля.

В режиме с ядом на поле появляются клетки, обозначенные символом %. Яд существует ограниченное время и перед исчезновением мигает. При попадании на яд у игрока отнимается 5 очков и уменьшается длина змейки. Если длина слишком мала, игра завершается.

Игра завершается при столкновении со стеной, столкновении с собственным телом, опасном попадании на яд или при нажатии клавиши X. После завершения программа показывает счёт, длину змейки и выбранный режим.

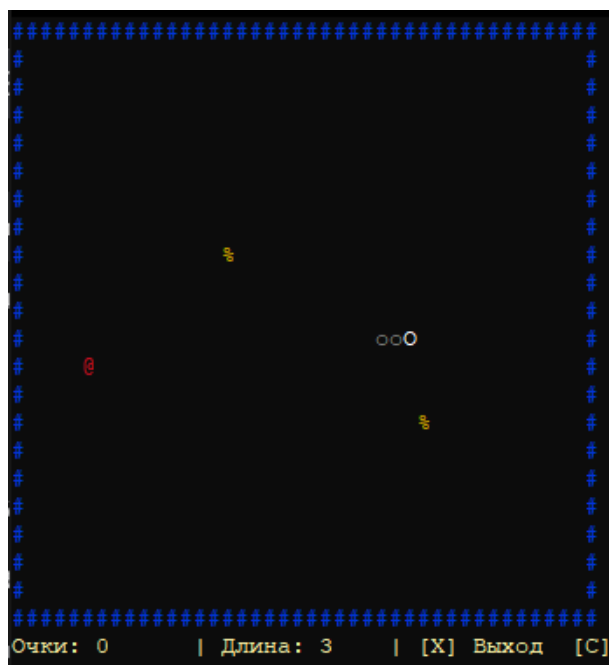


Рисунок 9 - Игровое поле

После окончания игры необходимо ввести имя игрока. Если оставить поле пустым, результат будет сохранён под именем «Аноним». Файл результата создаётся в папке РЕЗУЛЬТАТЫ (Рисунок 10).

Имя	Дата изменения	Тип	Размер
settings.txt	12.05.2026 22:53	Текстовый докум...	1 КБ
попытка 1441 1.txt	12.05.2026 21:22	Текстовый докум...	1 КБ
попытка 2442 1.txt	12.05.2026 21:14	Текстовый докум...	1 КБ

Рисунок 10 - Папка "РЕЗУЛЬТАТЫ"

Для просмотра лучших результатов нужно выбрать в главном меню пункт «Рекорды». Программа покажет таблицу (Рисунок 11) с именем игрока, очками, длиной змейки и режимом игры. Выводятся лучшие записи, отсортированные по количеству очков.

```
===== ТАБЛИЦА РЕКОРДОВ =====
```

№	Имя	Очки	Длина	Режим
1		0	0	С ядом
2	1441	0	3	Классика
3	213	0	3	С ядом
4	2442	0	3	Классика

Рисунок 11 - Таблица рекордов

Для смены оформления можно выбрать пункт «Сменить цвет» в главном меню (Рисунок 12) или нажать С во время игры. Выбранная палитра сохраняется в файле settings.txt и будет восстановлена при следующем запуске программы.

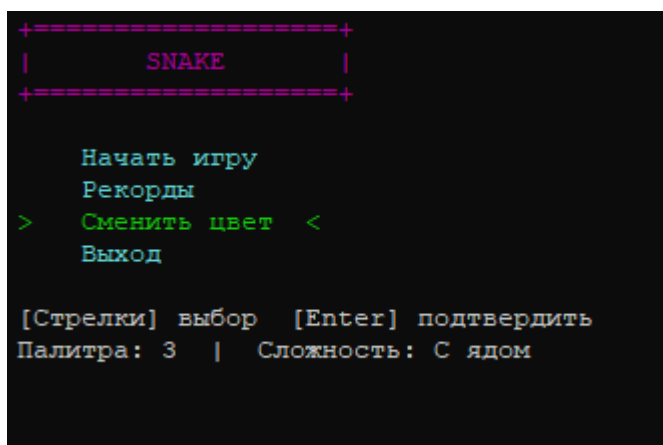


Рисунок 12 - Выбор цветовой палитры

Заключение

В ходе выполнения курсовой работы была разработана консольная игра The_zmeika на языке C и ASM. Программа реализует классический игровой процесс «Змейки», выбор сложности, цветной консольный интерфейс, сохранение результатов, таблицу рекордов и сохранение пользовательских настроек.

При разработке были закреплены навыки проектирования многомодульной программы, работы со структурами, массивами, файлами, функциями Windows API и обработкой клавиатурного ввода. Также были применены вставки ассемблера для отдельных операций игровой логики.

Разделение программы на модули упростило разработку и отладку. Каждый файл отвечает за отдельную часть приложения, что делает исходный код более понятным и удобным для дальнейшего развития.

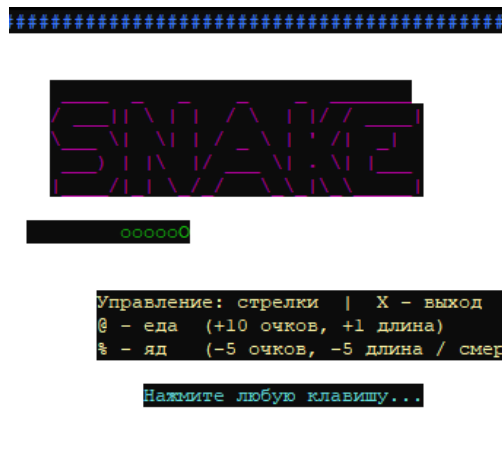
В дальнейшем программу можно улучшить: добавить выбор размера поля, несколько уровней скорости, паузу, сохранение общей таблицы рекордов в одном файле, расширенные препятствия, звуковые эффекты и переносимость на другие операционные системы без зависимости от WinAPI.

Список используемых источников

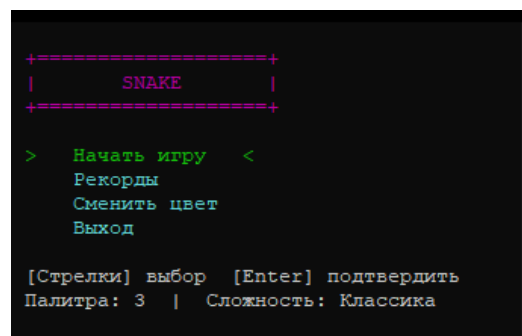
1. Керниган Б., Ритчи Д. Язык программирования С. — 2-е изд. — М. : Вильямс, 2021. — 304 с.
2. Microsoft Learn. Документация по Windows Console API [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/windows/console/> (дата обращения: 01.05.2026).
3. Microsoft Learn. Документация по языку С и компилятору MSVC [Электронный ресурс]. — URL: <https://learn.microsoft.com/ru-ru/cpp/c-language/> (дата обращения: 01.05.2026).
4. Павловская Т. А. С/С++. Программирование на языке высокого уровня. — СПб. : Питер, 2013. — 464 с.
5. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013. — 816 с.
6. ГОСТ Р 2.105-2019. Единая система конструкторской документации. Общие требования к текстовым документам. — М. : Стандартиформ, 2019. — 42 с.
7. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. — М. : Издательство стандартов, 1990. — 26 с.

Приложение В. Результаты работы программы

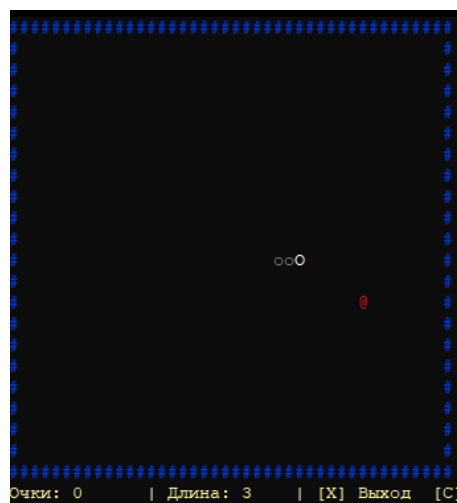
Заставка



Главное меню



Игровое поле



Файл результата

Игрок : Аноним

Попытка : 1

Очки : 40

Длина : 7

Сложность: Классика