

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

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе
по курсу «Программирование»
на тему «Игра “Крестики-нолики”»

Выполнил:

студент группы 21ВВ1

Федулов Е.Д.

Приняли:

Голотенков Н.О.

Токарев А.Н.

Пенза, 2022

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 Кудрявцев Е.Д. Группа 21БВТ

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

Исходные данные (технические требования) на проектирование

Примечание: игра "Крестики-нолики".
Функционал: сохранение результатов игры в файл, вывод результатов на экран 15 игр на экран.

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.

Объем работы по курсу

1. Расчетная часть

Разработка программы.

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 “ 1 ” Март 2022

Дата защиты проекта “ 22 ” 06. 12

Руководитель Таланчук К.О.

Задание получил “ 1 ” Март 2022г.

Студент Григорьев Е.В.

Содержание

Введение	3
1. Постановка задачи.....	4
2. Выбор решения.....	5
2.1 Определение необходимых модулей программы	5
2.2 Определение структуры файла базы данных.....	7
3. Описание разработки программы.....	8
4. Отладка и тестирование.....	10
5. Описание программы	11
6. Руководство пользователя	12
Заключение	15
Список используемых источников.....	16
Приложение А:	17
Листинги программы	17
Приложение А.1: Заголовочные файлы.....	17
header.h	17
Приложение А.2: Файлы .cpp.....	17
menu.cpp	17
chistka.cpp.....	18
exit.cpp.....	19
hod.cpp.....	19
kursa4.cpp.....	20
lastgames.cpp.....	20
na4alo.cpp.....	21
proverka.cpp	23
visual.cpp.....	24

Введение

Игры – неперенная составляющая любой человеческой культуры. Слово «игра» мы прилагаем почти ко всякому детскому занятию. Посредством игры наиболее интенсивно развивается целостное восприятие окружающего мира, наглядно-образное мышление, творческое воображение. Поэтому кроме программ, которые облегчают людям работу, программисты придумали увлекательное развлечение - компьютерные игры. Сначала в эти игры играли только они сами, но очень скоро их эстафету приняли дети и взрослые по всему миру. Сейчас уже трудно найти человека, имеющего компьютер и ни разу не игравшего в компьютерные игры.

Для создания игры “Крестики-нолики” в рамках данной курсовой работы используется язык Си и ассемблера.

Язык ассемблера – язык низкого уровня, команды которого полностью или частично соответствуют машинным кодам. Язык позволяет написать максимально эффективный и максимально компактный код.

Язык Си – компилируемый статически типизированный язык программирования общего назначения. Язык позволяет создавать сложные структуры для удобного хранения и предоставления информации, при этом позволяет работать с памятью компьютера, что открывает возможности для реализации разных методов работы с элементами калькулятора.

В процессе выполнения курсовой работы будут получены навыки разработки программы сложной структуры методом нисходящего программирования, а также улучшены навыки работы со сложными типами данных, таких как массивы, структуры, списки и файлы.

В качестве результата выполнения технического задания к курсовой работе, будет получена полноценная игра.

1. Постановка задачи

Разработать программу: Игра “Крестики-нолики”. В программе должен быть удобный интерфейс, управление, а также она должна передавать смысл настоящей игры.

Программа должна предоставлять пользователю управлять ей с помощью клавиатуры или мышки.

В программе должны использоваться сложные типы данных, такие как массивы, структуры и файлы. Это необходимо для структурирования игры, для обработки данных в коде программы.

Устройствами ввода информации являются клавиатура и мышь. Навигация по пунктам меню и данные происходит посредством клавиатуры.

Программа должна иметь стартовую заставку, отображающую информацию о программе и разработчике.

Программа должна быть разработана для работы в операционной системе Microsoft Windows.

Средой разработки является Microsoft Visual Studio 2010, языками программирования Си и Ассемблер.

2. Выбор решения

2.1 Определение необходимых модулей программы

Разработанная программа состоит из следующих модулей:

1. chistka.cpp
2. exit.cpp
3. hod.cpp
4. kursa4.cpp
5. lastgames.cpp
6. menu.cpp
7. na4alo.cpp
8. proverka.cpp
9. visual.cpp

Основным модулем приложения является файл `kursa4.cpp`, в котором вызывается функция заставки и главное меню игры, при котором программа переходит в рабочее меню в модуле.

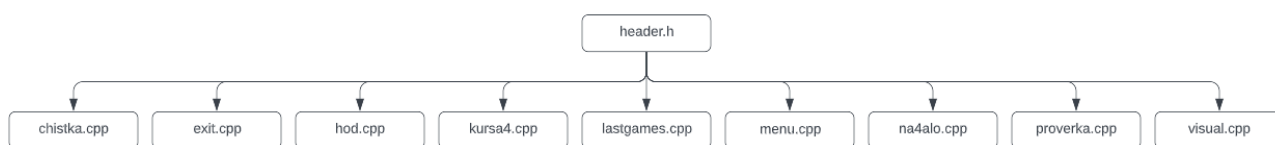


Рисунок 2.1.1 - Схема взаимодействия системы

Управление программой осуществляется при помощи клавиатуры. Программа позволяет начать игру, посредством написания “1” в консоли, посмотреть результаты последних 15 игр, посредством написания “3” в консоли, и выйти из нее, посредством написания “6”. После перехода к игре, появляется поле и программа ожидает написания номера ячеек, путем ввода двухзначного числа состоящего из цифр 0,1,2 (вида 00, 01, 22 и т.д), таким образом происходит игра. Меню поделено на три пункта, после выбора нужного пункта вызывается игра, результаты последних 15 игра или программа закрывается.

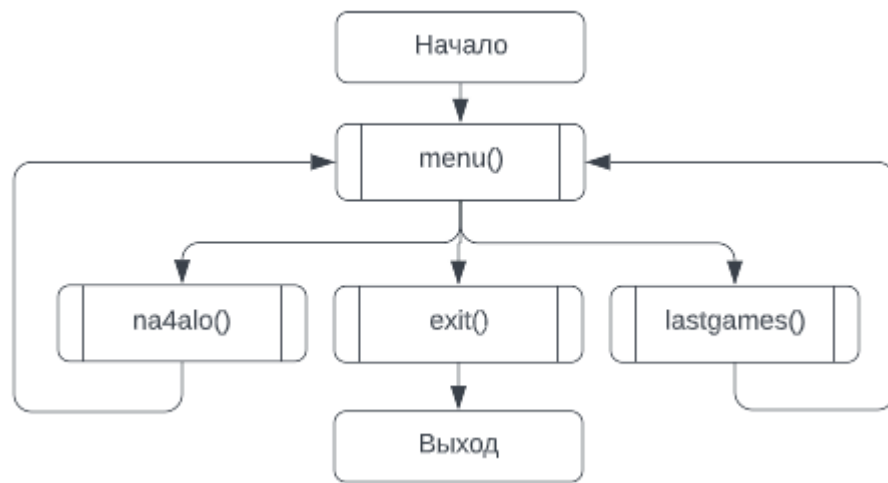


Рисунок 2.1.2 - Схема ресурсов системы

2.2 Определение структуры файла базы данных

Структур в данной программе не было использовано, так как для игры важна скорость и простота, а структуры сильно усложнили бы программу.

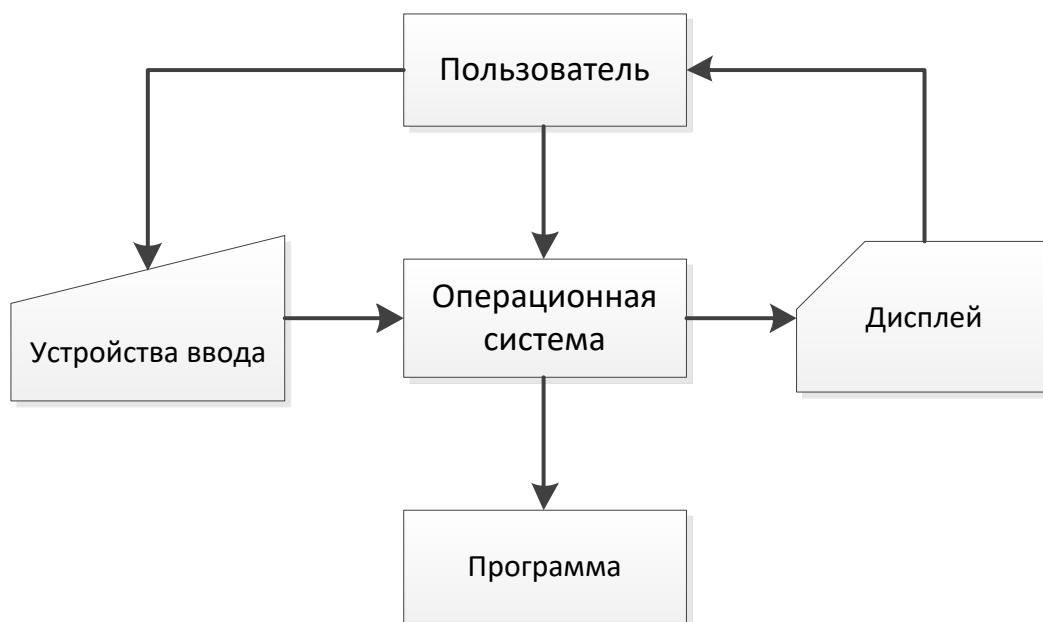


Рисунок 2.2.1 - Схема данных

3. Описание разработки программы

Для написания данной программы использовался язык программирования Си. Язык Си является одним из наиболее популярных языков программирования. Язык интересен тем, что его легко выучить из-за небольшого числа ключевых слов, а код, написанный на нем, легко читается.

Структура языка Си позволяет наилучшим образом использовать возможности современных ЭВМ. На языке Си программы обычно отличаются компактностью и быстротой исполнения. Программа, написанная на Си для одной вычислительной системы, может быть перенесена с небольшими изменениями (или вообще без них) на другую.

В качестве среды программирования был выбран программный продукт Microsoft Visual Studio 2010, так как это одна из самых удобных версий.

Информация о разработчике программы предоставляется на заставке, которая появляется во время запуска программы, после чего ожидается дальнейшее действие от игроков.

Интерфейс информационной системы выполнен в виде текста в консоли, выбор пунктов которого осуществим при помощи ввода определенной цифры в консоль .

В меню:

- Пункт первый - позволяет перейти к началу игры.
- Пункт второй - позволяет посмотреть результаты последних 15 игр.
- Пункт третий - позволяет выйти из программы.

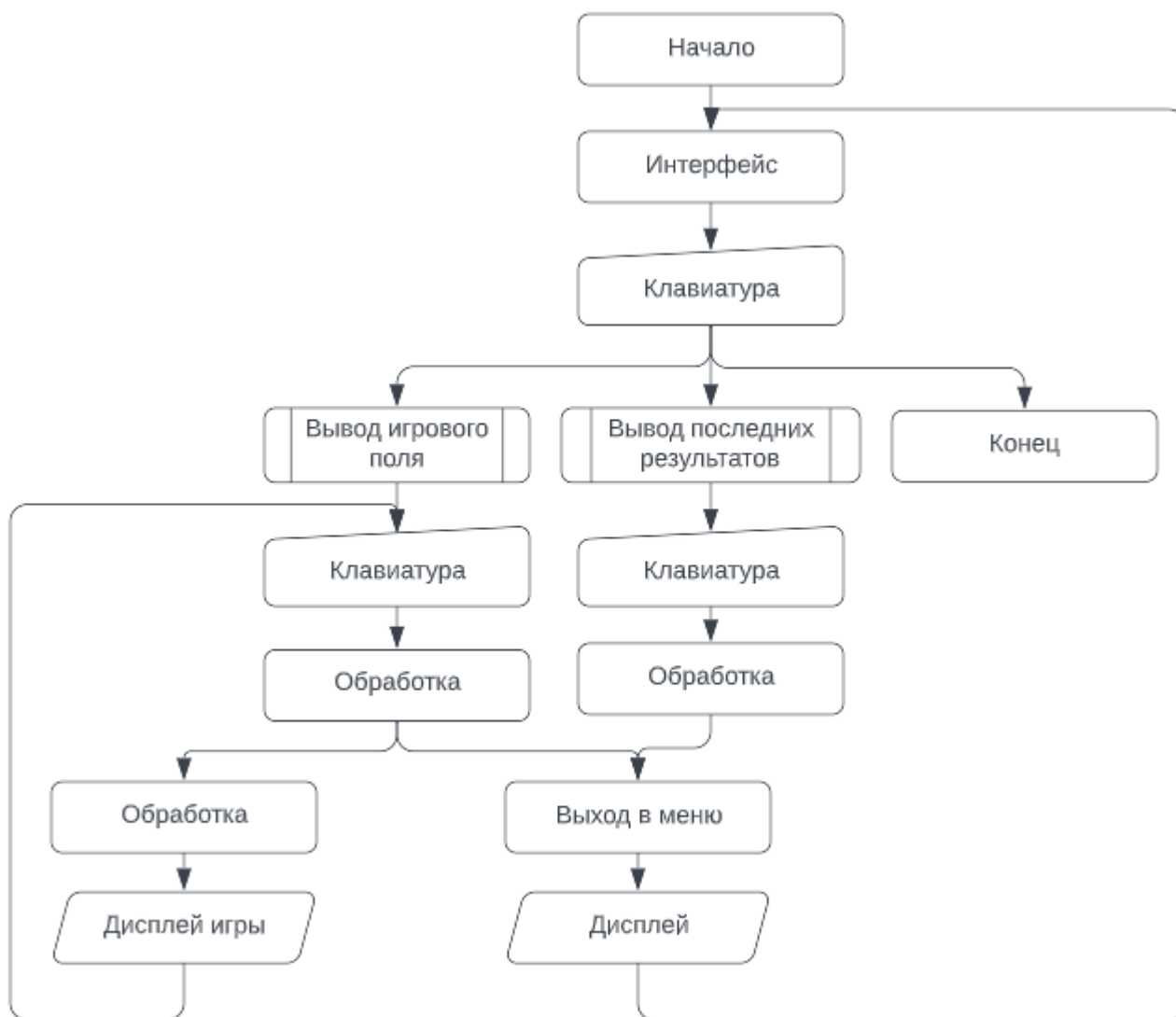


Рисунок 3.1 - Схема взаимодействия пользователя с программой

4. Отладка и тестирование

В качестве среды разработки была выбрана программа Microsoft VisualStudio 2010, которая содержит в себе все необходимые средства для разработки и отладки модулей и программ. Для отладки программы использовались различные инструменты: точка останова, пошаговое выполнение кода программы, анализ содержимого глобальных и локальных переменных, результаты работы некоторых функций.

Тестирование проводилось в рабочем порядке, в процессе разработки и после завершения написания программы. В процессе тестирования было выявлено и исправлено множество ошибок, связанных с созданием многомодульности и правильным занесением данных о ходе игрока.

5. Описание программы

После запуска программы на экран выводится заставка и ожидается написание цифры 1, 3 или 6 для того, чтобы вызвать игру, результаты последних 15 игр и выйти из программы, соответственно. После ввода определенной цифры происходит обработка соответствующего пункта меню.

Заставка реализована в файле `kursa4.cpp`. Структура заставки выглядит следующим образом:

1. «Играть»
2. «Результаты последних 15 игр»
3. «Выход»

Командное окно и окно заставки прописаны в файле `kursa4.cpp`. Игра находится в файлах `chistka.cpp`, `hod.cpp`, `na4alo.cpp`, `proverka.cpp`, `visual.cpp`. `na4alo.cpp` – это файл, в котором вызывается обработка и проверка внесенных данных, а так же вывод результата игры в консоль и отдельный файл; `chistka.cpp` – это файл, в котором проводится очистка поля после игры; `hod.cpp` – это файл, в котором обрабатывается ход игрока; `proverka.cpp` – это файл, в котором проходит обработка результатов на победу после каждого хода; `visual.cpp` – это файл, в котором находятся визуальные составляющие хода игры.

Иерархическая структура программы представлена.

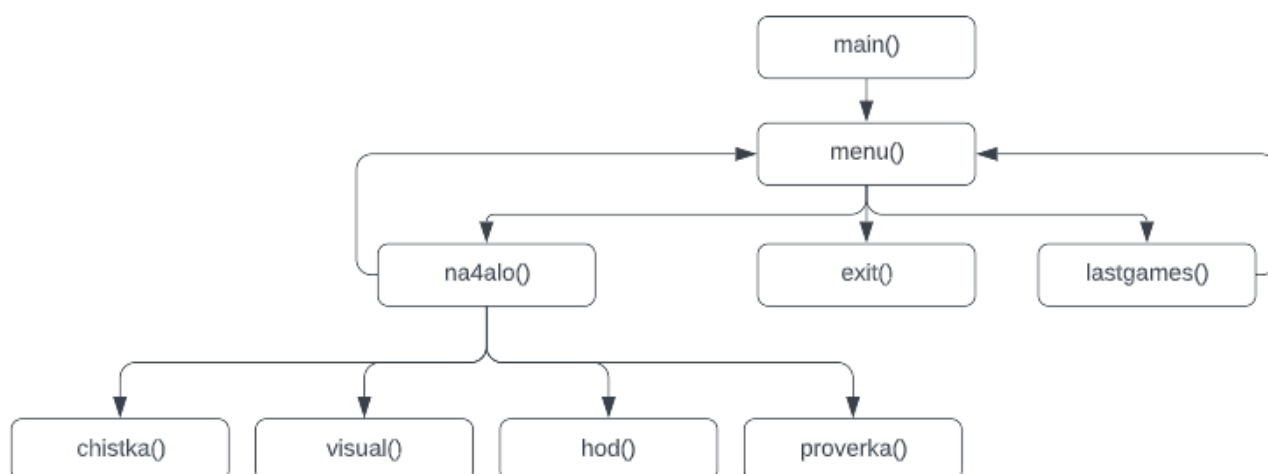


Рисунок 5.1 - Иерархическая структура программы

6. Руководство пользователя

Программа представляет собой игру “Крестики-нолики”. Программа имеет простой и понятный интерфейс.

После запуска приложения на экран выводится заставка-приветствие вместе с основным меню.

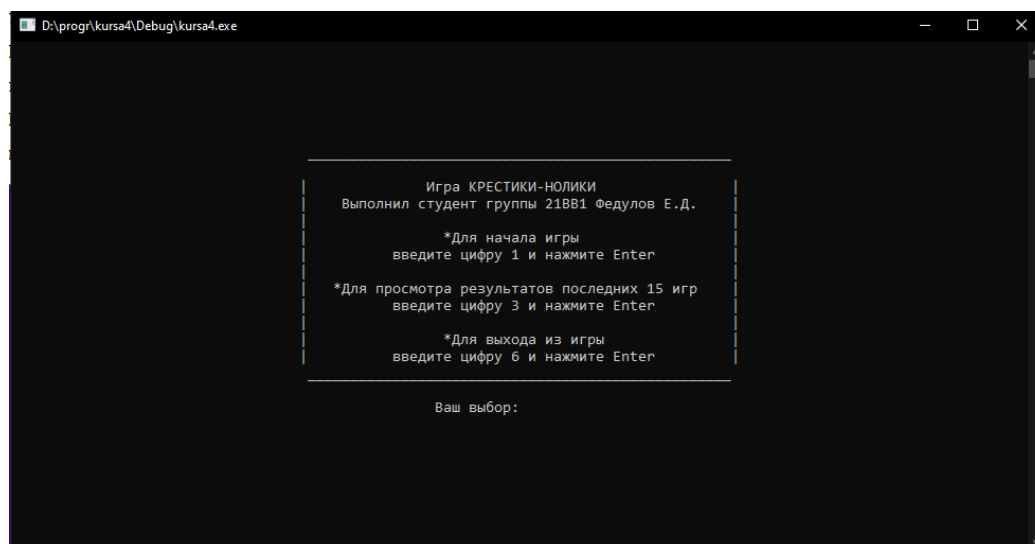


Рисунок 6.1 - Заставка-приветствие и меню программы

После ввода “1”, программа переходит к началу игры.

Если написать “3”, то программа выведет для пользователя результаты последних 15 игр с автообновлением.

При написании “6”, программа завершится.

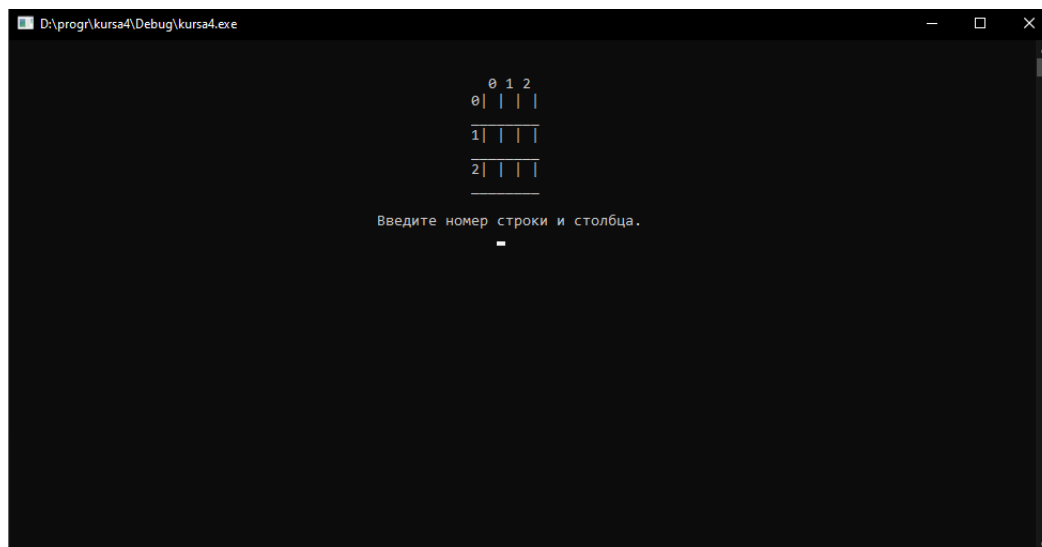


Рисунок 6.2 – Игровое поле

Вначале игры ожидается ввод пользователем номера ячейки, в формате двухзначного числа с цифрами от 0 до 2. После этого происходит постановка нолика/крестика на поле. В конце игры предоставляется возможность выхода в меню посредством написания “1” в консоли.

На рисунках 6.3, 6.4, 6.5 представлены итоги игры.



Рисунок 6.3 – Игрок за нолик выиграл



Рисунок 6.4 – Игрок за крестик выиграл



Рисунок 6.5 – Ничья

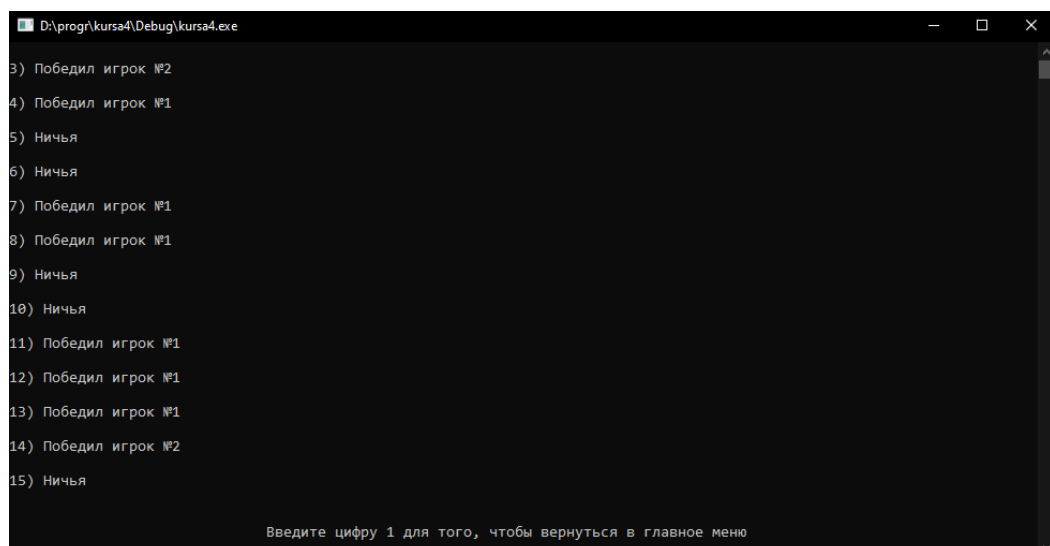


Рисунок 6.6 – Просмотр последних 15 игр

15 последних игр выводятся из отдельного файла, в котором хранятся данные. Данные выводятся с порядковым номером и в точности совпадают с прошедшими играми. Благодаря сохранению в файл, данные об играх не стираются и будут доступны снова при открытии программы, либо же открытии txt файла. Из данного окна меню можно выйти, написав “1” в консоль.

Заключение

При выполнении данной курсовой работы были получены базовые навыки программирования на языках Си и Ассемблер, усовершенствованы навыки разработки многомодульных программ. Изучены основные возможности среды программирования Microsoft Visual Studio 2010, получены навыки отладки и тестирования программ. Были рассмотрены некоторые функции из стандартной библиотеки языка Си. Были изучены базовые алгоритмы обработки данных.

В результате выполнения курсовой работы была разработана программа игра “Крестики-нолики”. Не требует большого количества системных ресурсов и обеспечивает быструю обработку данных, имеет удобный и понятный интерфейс.

В дальнейшем можно реализовать полноценный графический интерфейс.

Список используемых источников

1. Б. Страуструп – «Язык программирования С++ специальное издание»
2. О.Калашников – «Ассемблер – это просто»
3. Б.В. Керниган, Д.М. Ричи. – «Язык С.»
4. М. Уэйт, С. Прата, Д. Мартин -«Язык Си руководство для начинающих.»
5. Р.Стивенс - «Алгоритмы. Теория и практическое применение»

Приложение А

Листинги программы

Приложение А.1: Заголовочные файлы

header.h

```
#ifndef header_h
#define header_h

extern int a[3][3];

void exit();
void lastgames();
void chistka();
void visual();
void hod(int n);
int proverka();
void na4alo();
void menu();

#endif
```

Приложение А.2: Файлы .срр

menu.cpp[illegible]

printf("\n");		Выполнил студент группы 21BB1 Федулов Е.Д.
printf("\n");		
printf("\n");		*Для начала игры \n");
printf("\n");		введите цифру 1 и нажмите Enter \n");
printf("\n");		\n");
printf("\n");		*Для просмотра результатов последних 15 игр
printf("\n");		
printf("\n");		введите цифру 3 и нажмите Enter \n");
printf("\n");		\n");
printf("\n");		*Для выхода из игры \n");
printf("\n");		введите цифру 6 и нажмите Enter \n");
printf("\n");		
<hr/>		
printf("Ваш выбор: ");		\n\n");
scanf("%d", &cifra);		
if (cifra == 1){		
na4alo();		
}		
if (cifra == 3){		
lastgames();		
}		
if (cifra == 6){		
exit();		
}		
}		

chistka.cpp

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void chistka()
{
    int i,j;

    // Проверка по строкам
    for(i=0; i<3; i++)
    {

```

```
// Проверка по столбцам
    for (j=0;j<3;j++)
    {
        a[i][j]=0;
    }
}
}
```

exit.cpp

```
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void exit()
{
    exit(0); // Выход из программы
}
```

hod.cpp

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void hod (int n)
{
    int hi;
    int sto;
    int str;
    int ver=0;

    while (ver==0)
    {
        printf ("\n                ");
        Введите номер строки и столбца.

        scanf("                %d", &hi);

        // Определяем номер столбца - это последняя цифра в ходе
        sto=hi%10;
```

```

        // Определяем номер строки хода - это первая цифра
        str=(hi-sto)/10;

        // Проверка корректности хода, клетка должна быть пуста и номер
        // должен быть один из номеров таблицы 3 на 3
        if (a[str][sto]==0 && (hi==00 || hi==01 || hi==02 || hi==10 || hi==11 ||
        hi==12 || hi==20 || hi==21 || hi==22 )) (ver=1);
    }

    // Если ход первого игрока, то 1
    if ( n==1) {a[str][sto]=1;}
    // Если ход второго игрока, то 2
    if ( n==2) {a[str][sto]=2;}
}

```

kursa4.cpp

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"
int a[3][3] = {(0,0,0),(0,0,0),(0,0,0)};
void main()
{
    menu();
}

```

lastgames.cpp

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void lastgames()
{
    int cifra2;
    system("cls");
    int k=0;
    char string[25];
    FILE *stroka = fopen("D:/progr/kursa4/kursa4/lastgames.txt","r");
    do{
        k=k+1;
    } while ( fgets(string, 25, stroka) != NULL);
}

```

```

fclose(stroka);
FILE *input = fopen("D:/progr/kursa4/kursa4/lastgames.txt", "r");
if(k>15)
{
    for(int i = 1; i<k; i++)
    {
        if((k-i)>15)
        {
            fgets(string, 25, input);
        }
    }
    for(int i = 1; i<16; i++)
    {
        fgets(string, 25, input);
        printf("%d", i);
        puts(string);
    }
}
else
{
    for(int i = 1; i<k; i++)
    {
        fgets(string, 25, input);
        printf("%d", i);
        puts(string);
    }
}
fclose(input);
printf("\n
главное меню\n
Введите цифру 1 для того, чтобы вернуться в
");
scanf("%d", &cifra2);
if (cifra2 == 1)
{
    menu();
}
getch();
}

```

na4alo.cpp

```

#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void na4alo()

```

```

{
    int cifra2;
    int s4et=0;
    system("cls");
    int igra;
    chistka();
    visual();
    igra=0;

    while (igra==0)
    {
        // Задаем ход первого игрока
        hod(1);
        // Отображаем поле
        visual();
        // Проверка победы
        igra=proverka();
        // Ход второго, в случае невыполнения условий выше
        s4et=s4et+1;
        _asm{
            mov eax, s4et
            cmp eax, 9
            je t
            jmp final
t:
            mov ebx, igra
            cmp ebx, 0
            je t2
            jmp final
t2:
            mov igra, 3
final:
        }
        if (igra==0)
        {
            // Задаем ход второго игрока
            hod(2);
            // Отображаем поле
            visual();
            // Проверка победы
            igra=proverka();
            s4et=s4et+1;
        }
    }

    // Отображение результатов игры с записью в файл
    FILE *output = fopen("D:/progr/kursa4/kursa4/lastgames.txt", "a");

```

```

if (igra==1)
{
    printf ("Победил игрок №1\n");
    fprintf (output,"Победил игрок №1\n");
}
if (igra==2)
{
    printf ("Победил игрок №2\n");
    fprintf (output,"Победил игрок №2\n");
}
if (igra==3)
{
    printf ("Ничья\n");
    fprintf (output,"Ничья\n");
}
fclose(output);
printf("\nВведите цифру 1 для того, чтобы вернуться в
главное меню\n");
scanf("%d", &cifra2);
if (cifra2 == 1)
{
    menu();
}
getch();
}

```

proverka.cpp

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

int proverka()
{
    int i;
    int res;
    res=0;
    // Проверяем победу первого игрока
    // Строки
    for (i=0; i<3; i++)
    {
        // Проверка все ли в строке нолики
    }
}

```

```

        if (a[i][0]==1 && a[i][1]==1 && a[i][2]==1 ) {res=1;}
    }

    // Столбцы
    for (i=0; i<3; i++)
    {
        // Проверка все ли в столбце нолики
        if (a[0][i]==1 && a[1][i]==1 && a[2][i]==1 ) {res=1;}
    }

    // Диагонали
    if (a[0][0]==1 && a[1][1]==1 && a[2][2]==1 ) {res=1;}
    if (a[0][2]==1 && a[1][1]==1 && a[2][0]==1 ) {res=1;}

    // Проверяем победу второго игрока
    // Строки
    for (i=0; i<3; i++)
    {
        // Проверка все ли в строке крестики
        if (a[i][0]==2 && a[i][1]==2 && a[i][2]==2 ) {res=2;}
    }

    // Столбцы
    for (i=0; i<3; i++)
    {
        // Проверка все ли в столбце крестики
        if (a[0][i]==2 && a[1][i]==2 && a[2][i]==2 ) {res=2;}
    }

    // Диагонали
    if (a[0][0]==2 && a[1][1]==2 && a[2][2]==2 ) {res=2;}
    if (a[0][2]==2 && a[1][1]==2 && a[2][0]==2 ) {res=2;}
    return res;
}

```

visual.cpp

```

#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <conio.h>
#include <locale.h>
#include <cstdlib>
#include "header.h"

void visual()
{
    int i,j;
    printf ("\n\n

```

```

0 1 2\n");

```

```

// Строки
for(i=0; i<3; i++)
{
    printf ("                %d", i);
    // Столбцы
    for (j=0;j<3;j++)
    {
        if (a[i][j]==0){printf("| ");};
        if (a[i][j]==1){printf("|O");};
        if (a[i][j]==2){printf("|X");};
    }

    // Переход на след. строку

    printf ("\\n");
    printf ("                \\n");
}
}

```