

Основы программирования

ФИСТ 1 курс

Власенко

Олег

Федосович

Лекция 11.

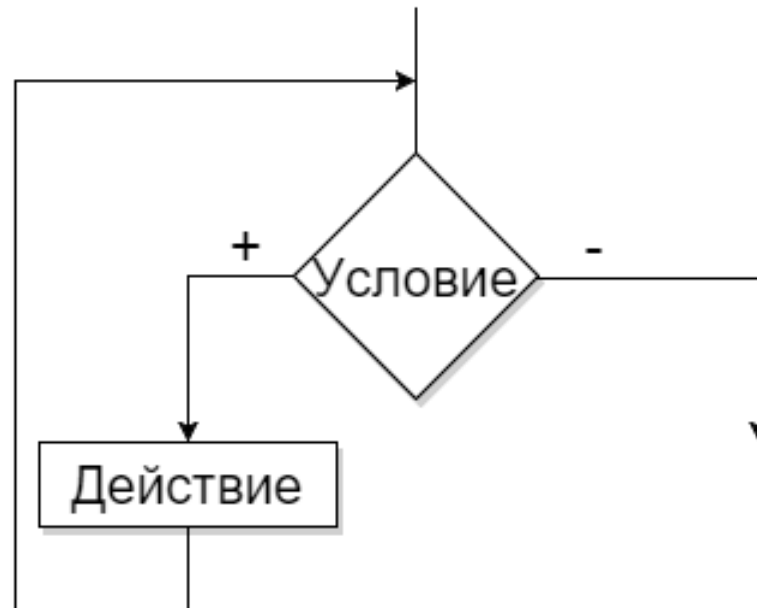
Цикл FOR

Загрузка числовых данных из текстового файла

Игра на основе 2D массива - Sokoban.

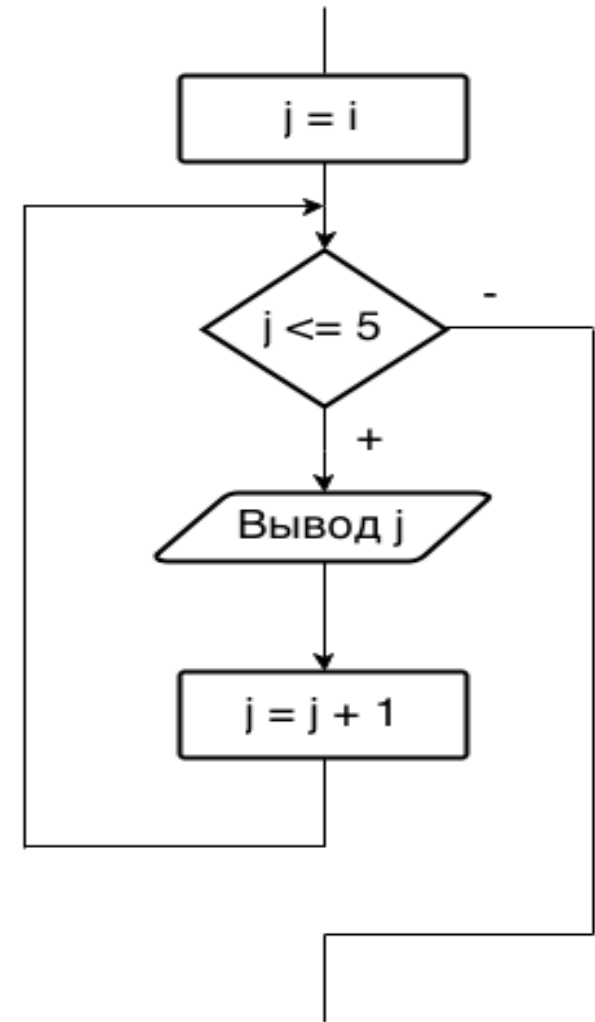
Цикл с предусловием while

```
while (Условие) {  
    Действие;  
}
```



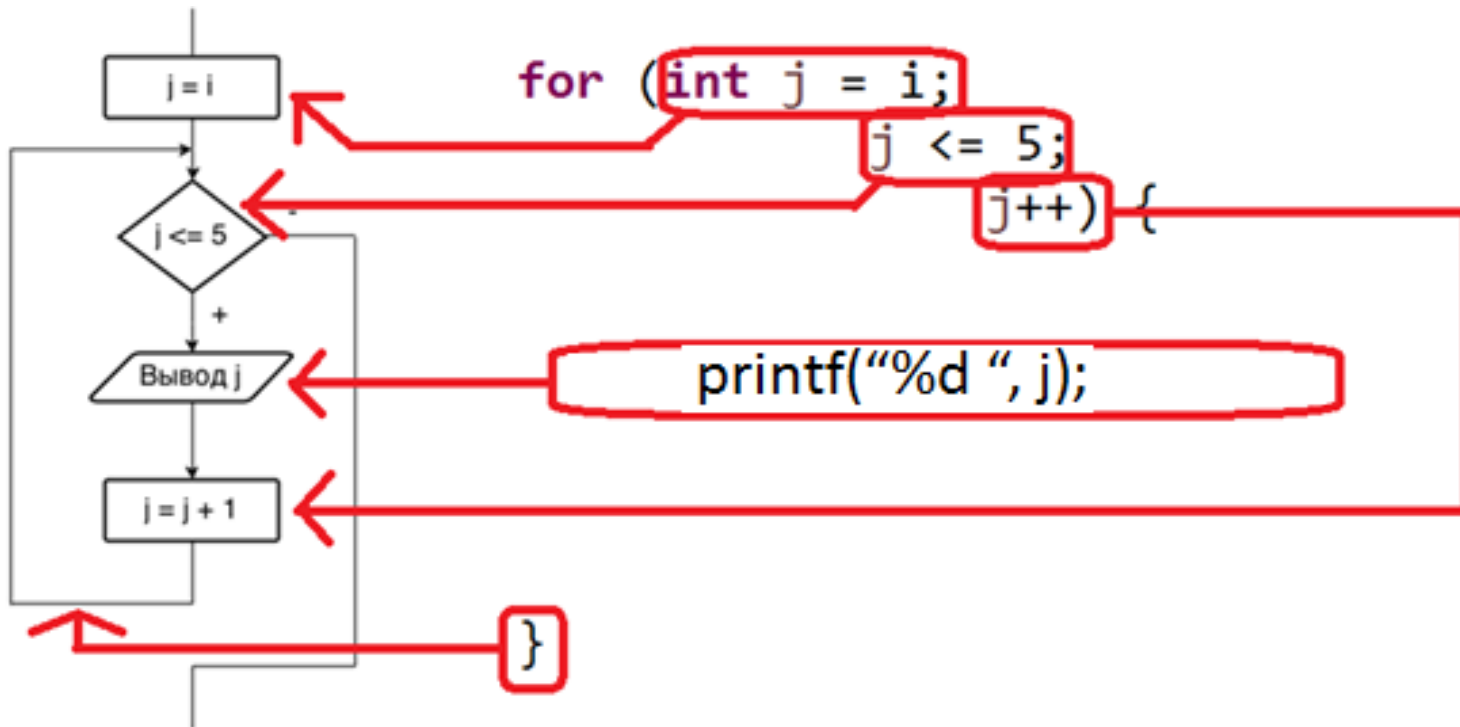
Цикл for

```
int j = i; // инициализация счетчика цикла
while (j <= 5) { // условие продолжения цикла
    printf("%d ", j);
    j++; // изменение счетчика цикла
}
```



Цикл for (2)

```
for (int j = i; j <= 5; j++) {  
    printf("%d ", j);  
}
```



Цикл for – рисуем блок-схему!

```
f = 1;  
for (i = 1; i <= n; i++) {  
    f = f * i;  
}
```

Цикл for – рисуем блок-схему! (2)

```
f = 1;  
for (i = 1; i <= n; i++) {  
    f = f * i;  
}
```

```
f = 1;  
i = 1;  
while (i <= n) {  
    f = f * i;  
    i++;  
}
```

файл

<https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB>

Файл ([англ.](#) *file*) — именованная область данных на [носителе информации](#).

Текстовый файл

Текстовый файл содержит последовательность символов (в основном печатных знаков, принадлежащих тому или иному набору символов). Эти символы обычно сгруппированы в строки (англ. *lines, rows*). В современных системах строки разделяются разделителями строк

https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB

Работа с файлом – общий алгоритм

- 1) Открыть файл
- 2) Работать с файлом
- 3) Заккрыть файл

Задача 1 – прочитать из файла 2 целых числа, подсчитать их сумму, вывести в другой файл

// Чтение из входного файла

// Обработка

// Запись в выходной файл

Входной файл:

3 12

Выходной файл:

15

Задача 1 – прочитать из файла 2 целых числа,
подсчитать их сумму, вывести в другой файл

// Чтение из входного файла

```
FILE *fin;
```

```
int a, b, s;
```

```
fin = fopen("c:\\Temp\\Files\\in1.txt", "rt");
```

```
if (fin == NULL) {
```

```
    printf("File in1.txt is not found");
```

```
    return;
```

```
}
```

```
fscanf(fin, "%d%d", &a, &b);
```

```
fclose(fin);
```

Задача 1 (2)

// Обработка

s = a + b;

Задача 1 (3)

// Запись в выходной файл

```
FILE *fout;
```

```
fout = fopen("c:\\Temp\\Files\\out1.txt", "wt");
```

```
if (fout == NULL) {
```

```
    printf("File out1.txt cannot be created");
```

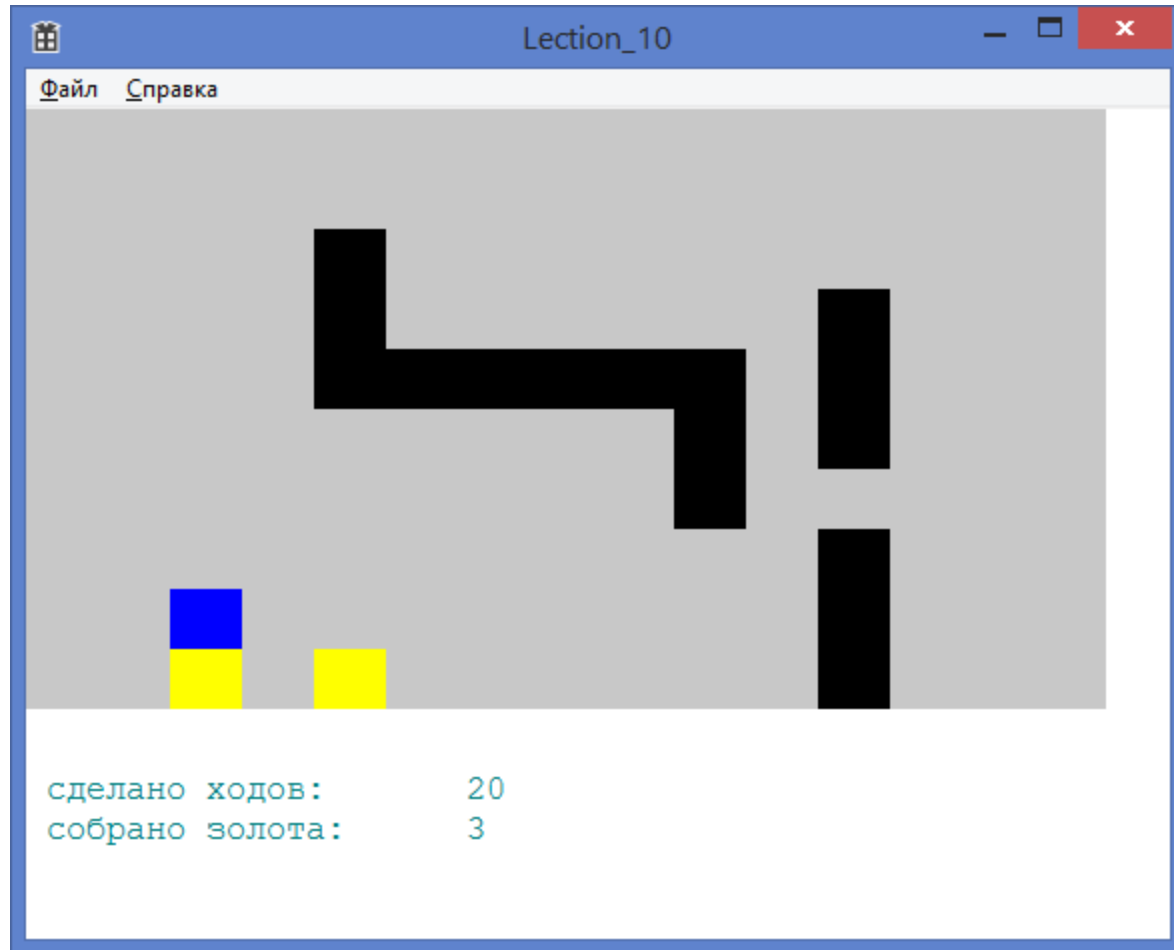
```
    return;
```

```
}
```

```
fprintf(fout, "s = %d", s);
```

```
fclose(fout);
```


Будем делать Sokoban – на основе предыдущей игры



Кодируем состояние игры в 2D массиве

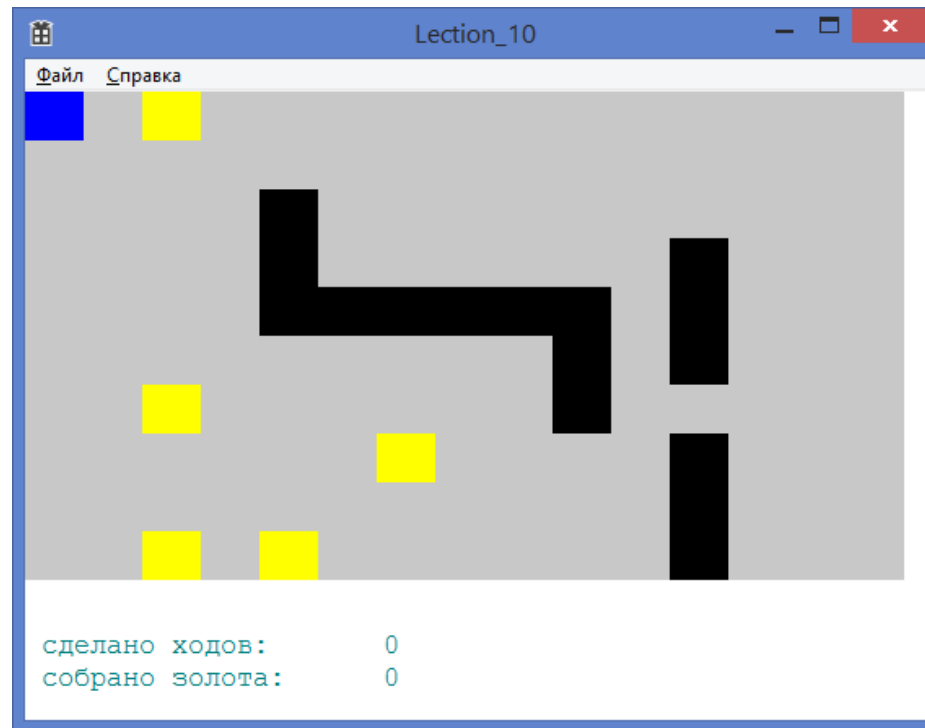
```
#define N 10
```

```
#define M 15
```

```
int a[N][M] = {  
  { 3, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },  
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },  
  { 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 },  
  { 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0 },  
  { 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 2, 0, 0, 0 },  
  
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 2, 0, 0, 0 },  
  { 0, 0, 1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0 },  
  { 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0 },  
  { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0 },  
  { 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0 }  
};
```

```
int steps = 0;
```

```
int gold = 0;
```



Коды ячеек

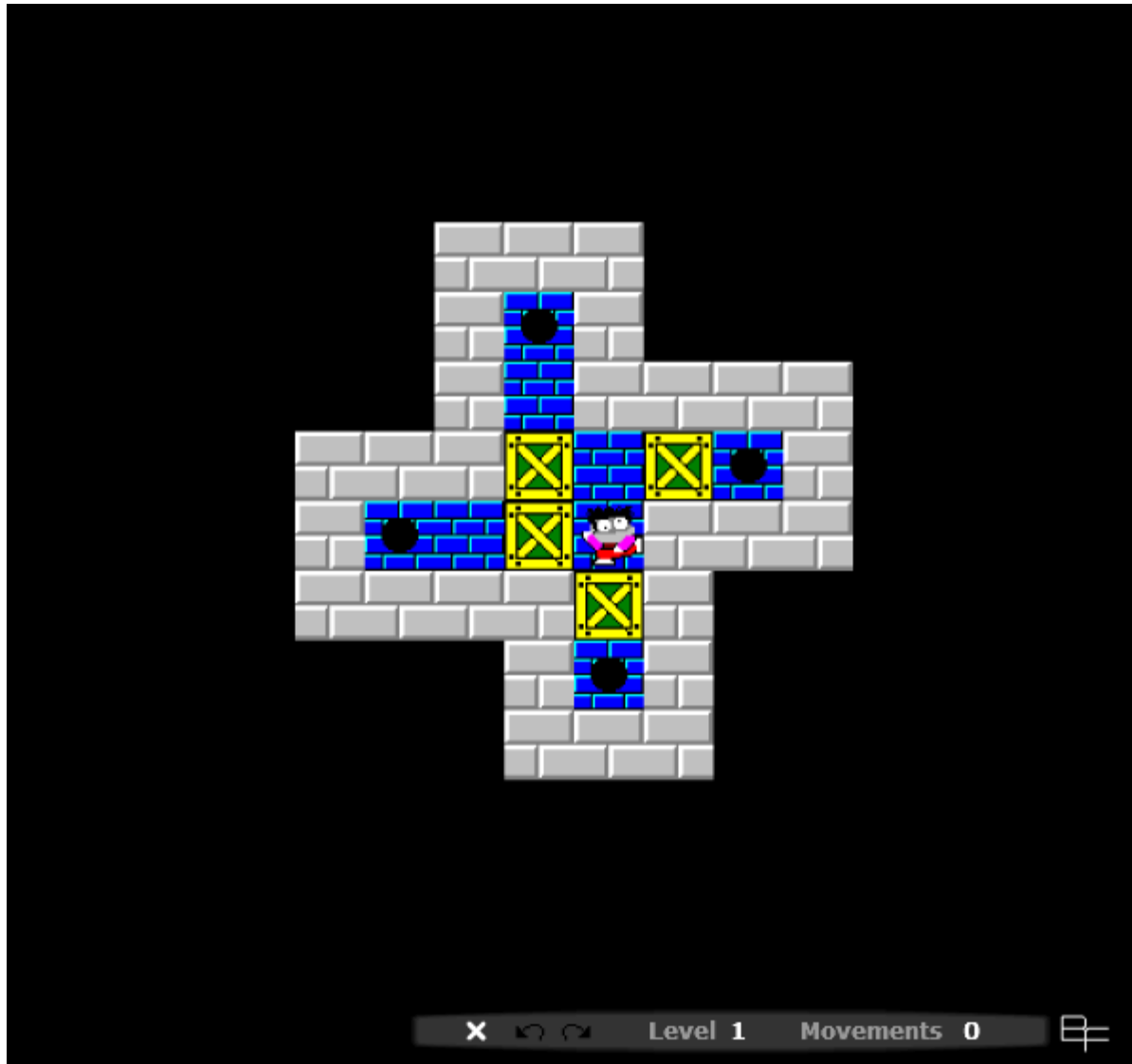
// 0 - свободно

// 1 - золото

// 2 - стена

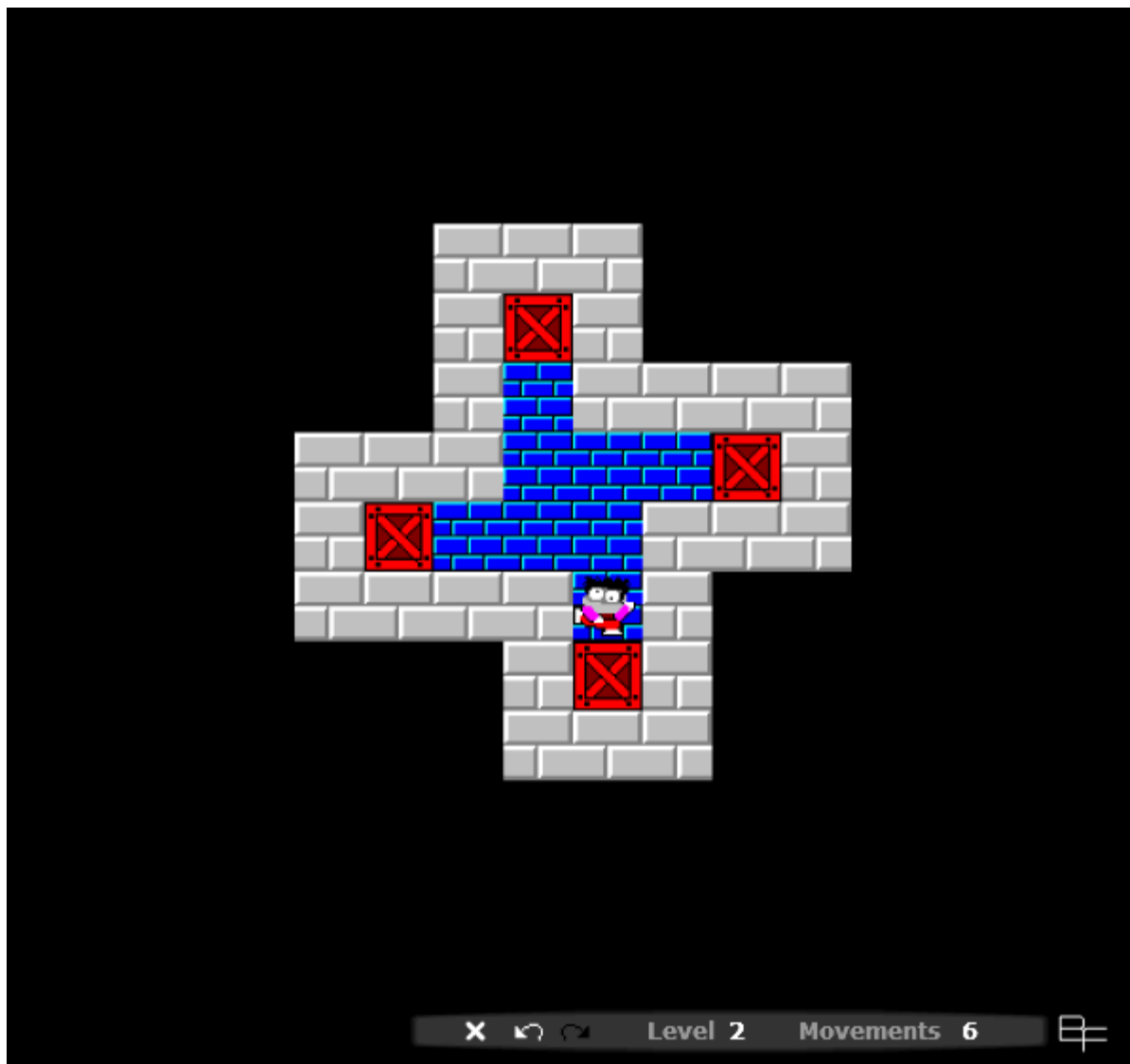
// 3 - игрок

Sokoban



http://www.flashplayer.ru/play_4959.php

Sokoban – нужно растолкать ящики по местам



http://www.flashplayer.ru/play_4959.php

Как будем хранить игровое поле и состояние?

```
#define N 12
```

```
#define M 15
```

```
int field[N][M];
```

```
// 0 – не игровое поле
```

```
// 1 – стена
```

```
// 2 - игровое поле
```

```
// 3 – поле для ящика
```

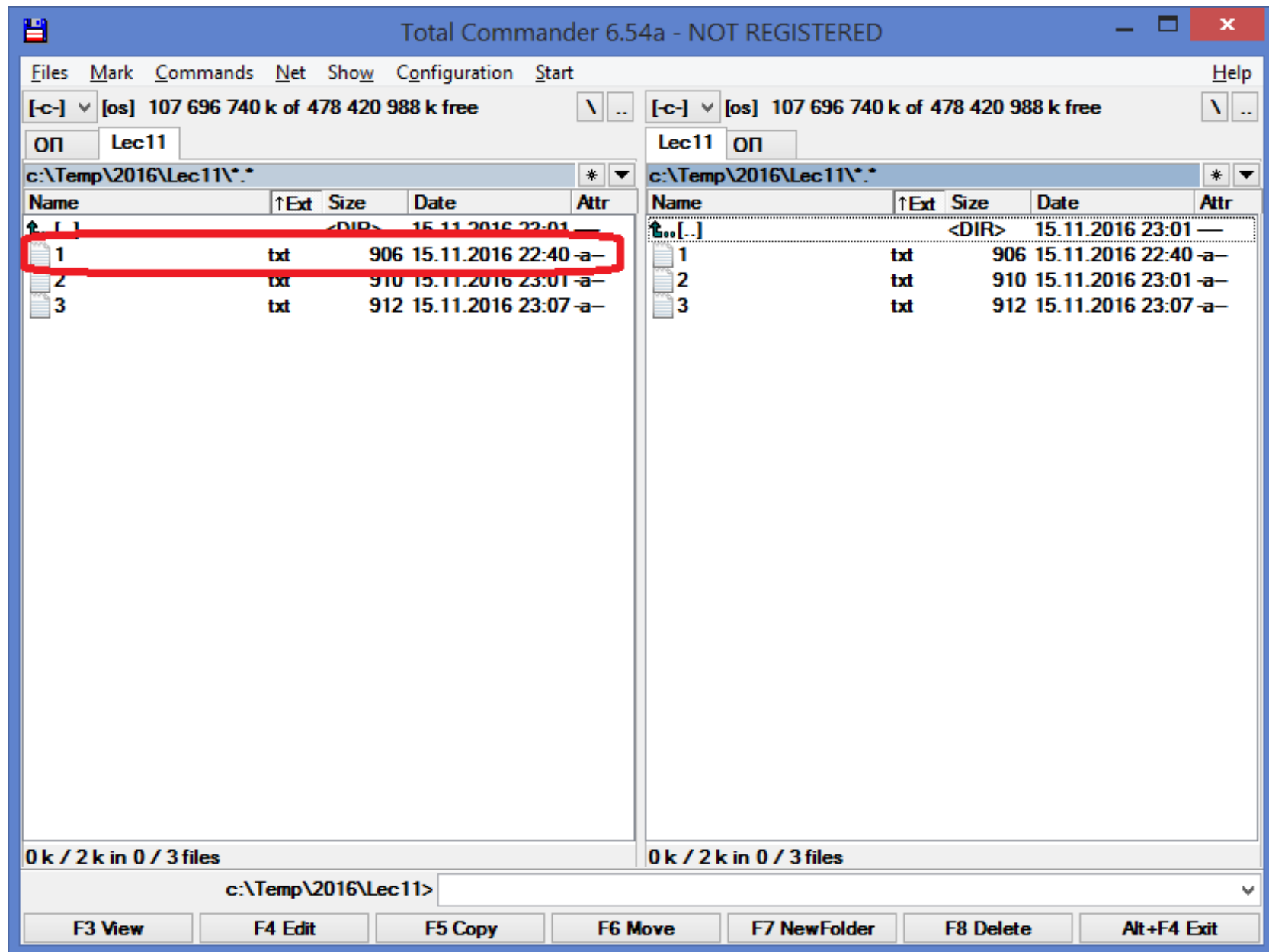
```
int boxes[N][M];
```

```
// 0 – на поле ничего нет
```

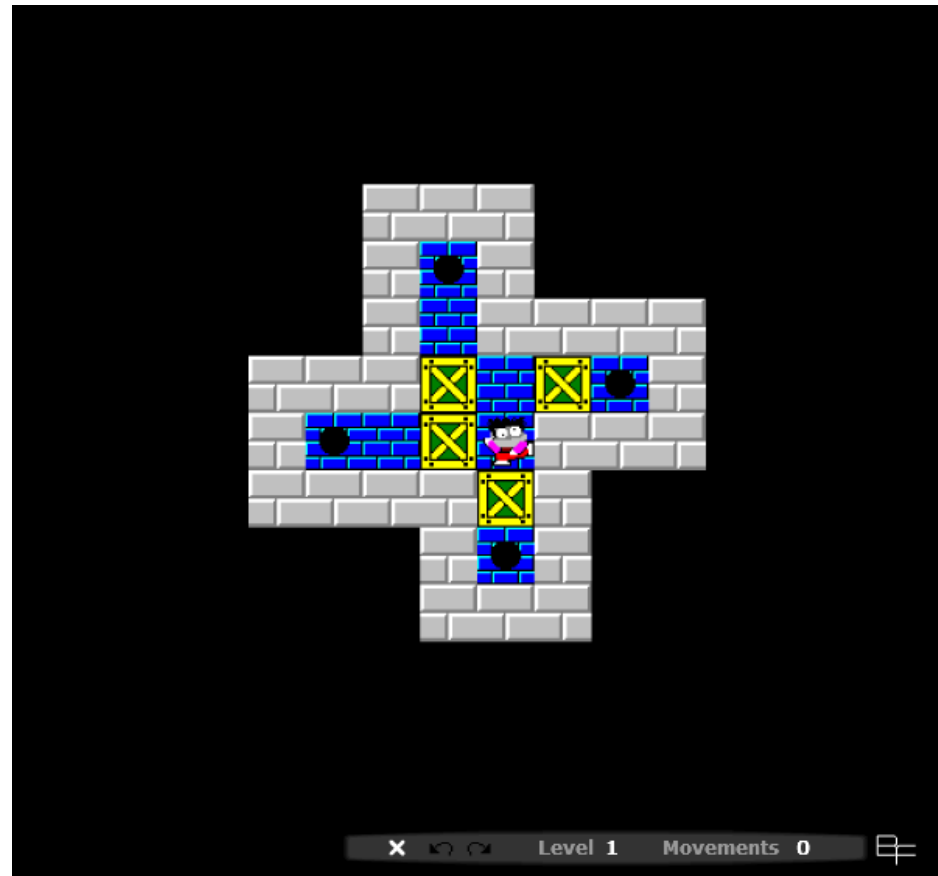
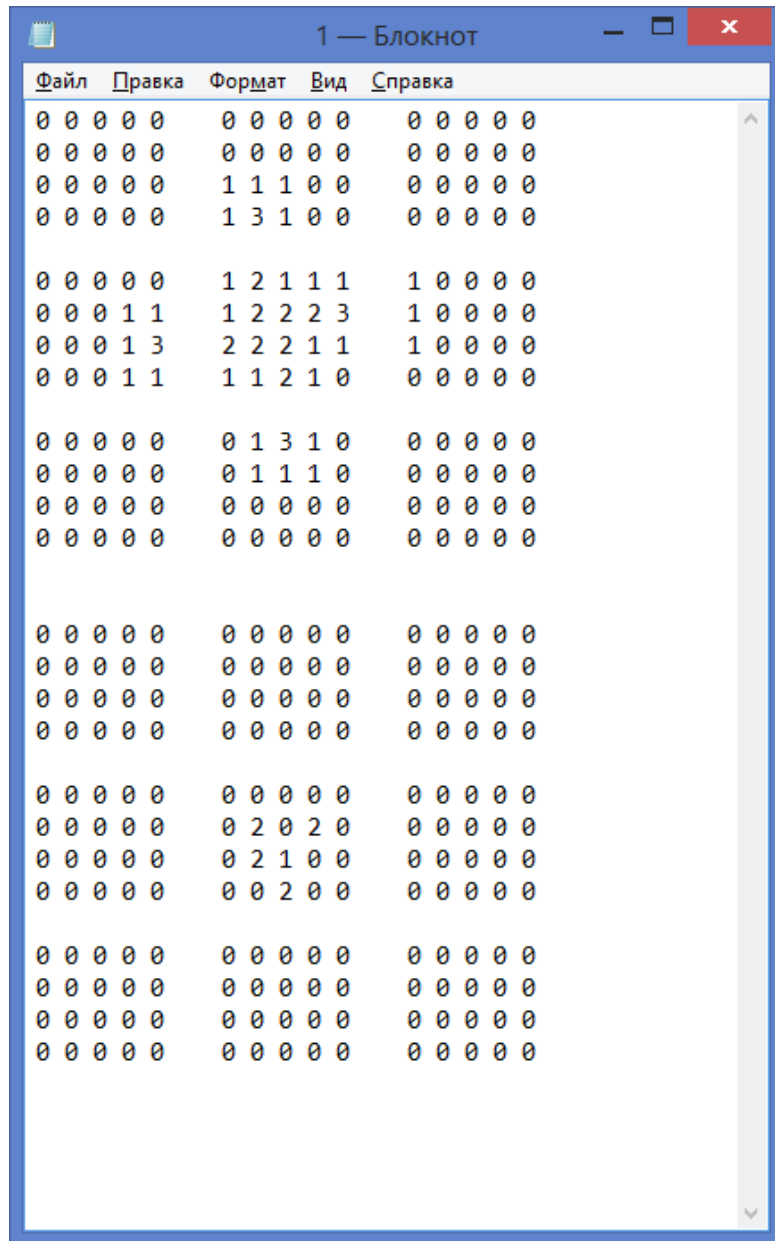
```
// 1 – грузчик
```

```
// 2 - ящик
```

Файл с игровым полем



Содержимое файла 1.txt



Загрузка игрового поля из файла - main

```
void main()
{

    if (loadLevel("c:\\Temp\\2016\\Lec11\\1.txt")) {
        printField();
    }

    {
        int x;
        scanf("%d", &x);
    }

}
```

Загрузка игрового поля из файла – loadLevel (1)

```
int loadLevel(char filename[]) {
```

```
    FILE *fin = fopen(filename, "rt");
```

```
    int i, j;
```

```
    if (fin == NULL) {
```

```
        printf("File %s is not opened", filename);
```

```
        return 0;
```

```
    }
```

Загрузка игрового поля из файла – loadLevel (2)

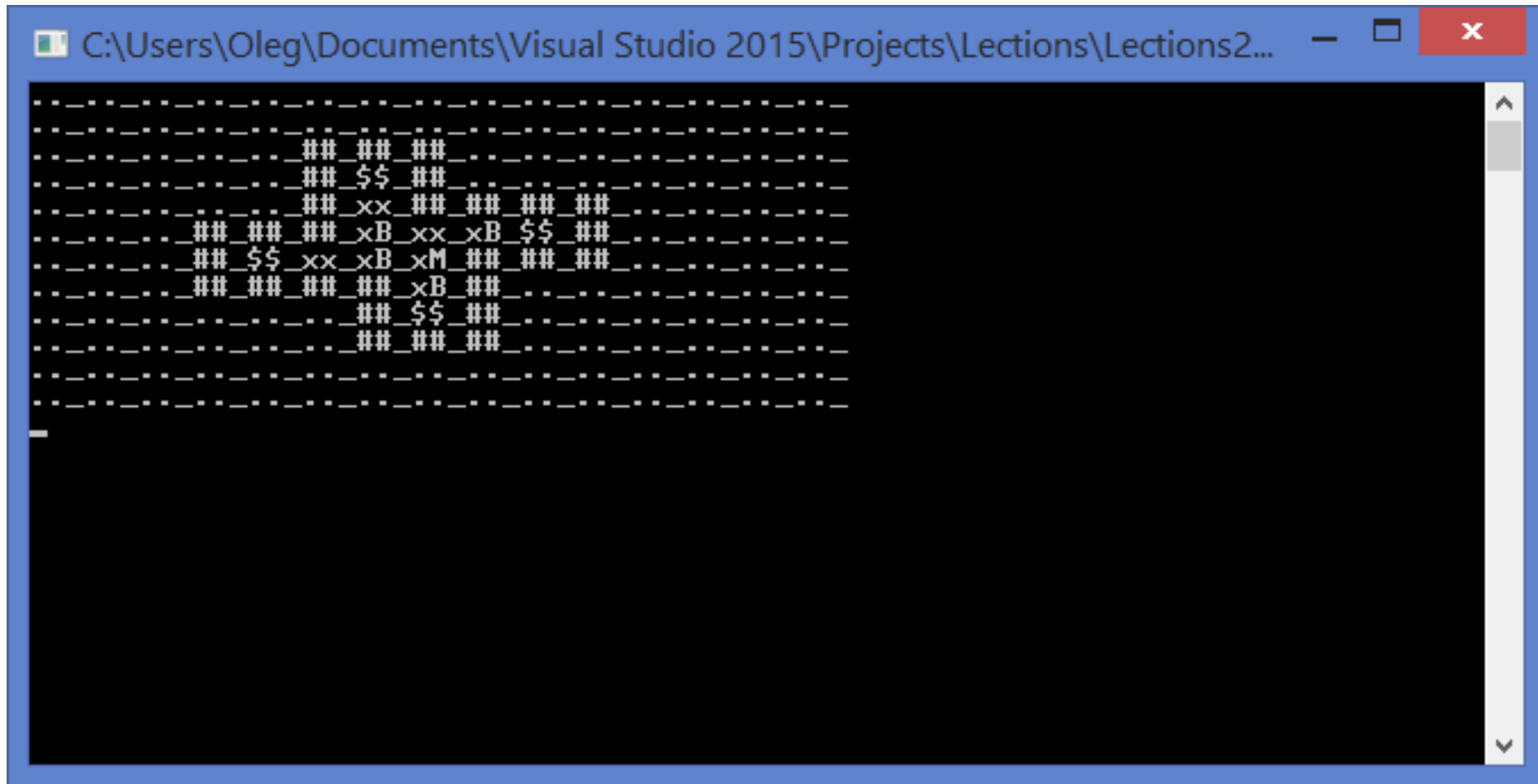
```
for (i = 0; i < N; i++) {  
    for (j = 0; j < M; j++) {  
        fscanf(fin, "%d", &field[i][j]);  
    }  
}
```

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < M; j++) {  
        fscanf(fin, "%d", &boxes[i][j]);  
    }  
}  
fclose(fin);  
return 1;  
}
```

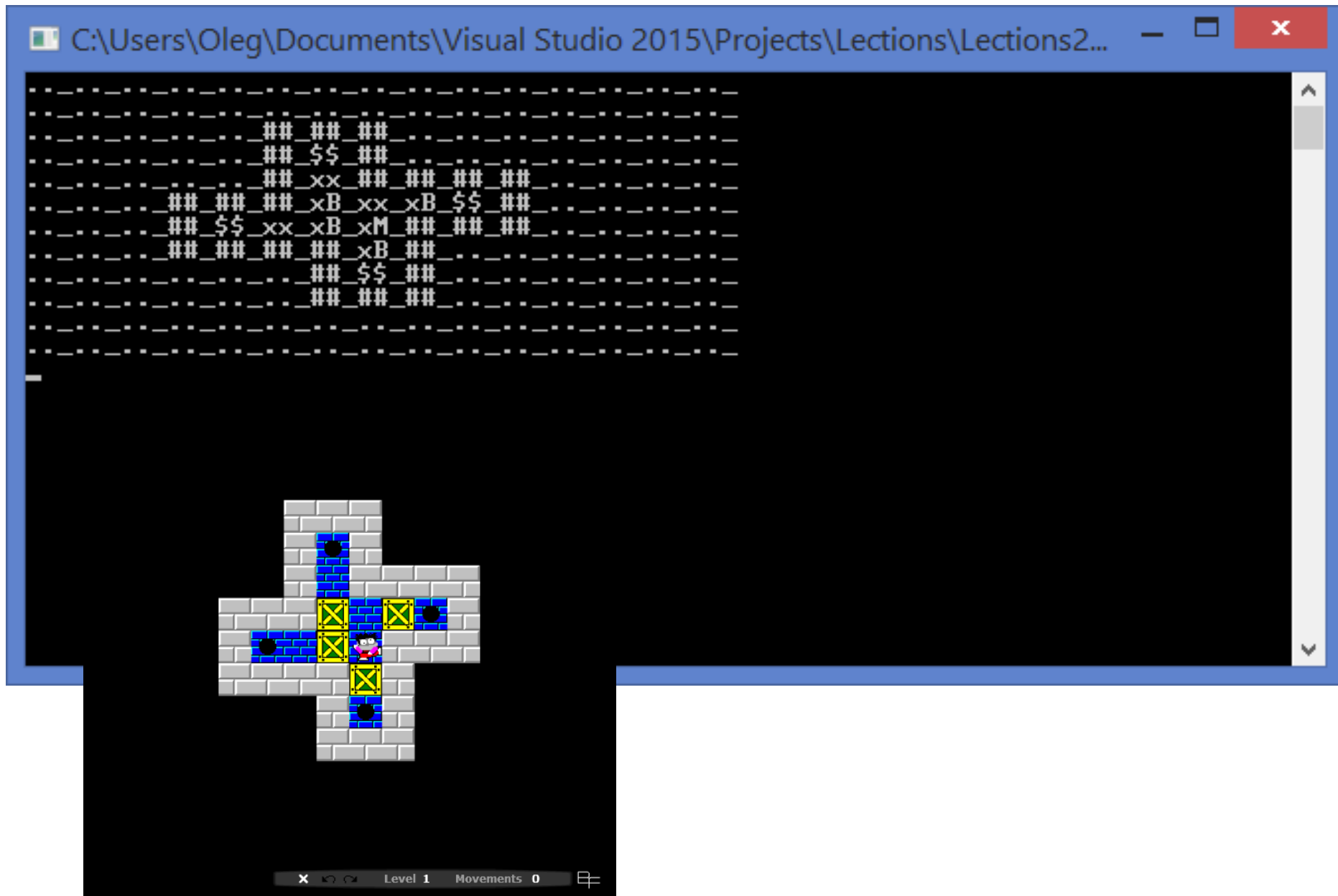

Печать игрового поля

```
void printField() {  
    int i, j;  
    for (i = 0; i < N; i++) {  
        for (j = 0; j < M; j++) {  
  
            char symbol[4][3][3] = {  
                { "..", ".1", ".2" },  
                { "##", "#1", "#2" },  
                { "xx", "xM", "xB" },  
                { "$$", "$M", "$B" }  
            };  
  
            printf("%s_", symbol [field[i][j]] [boxes[i][j]] );  
        }  
        printf("\n");  
    }  
}
```

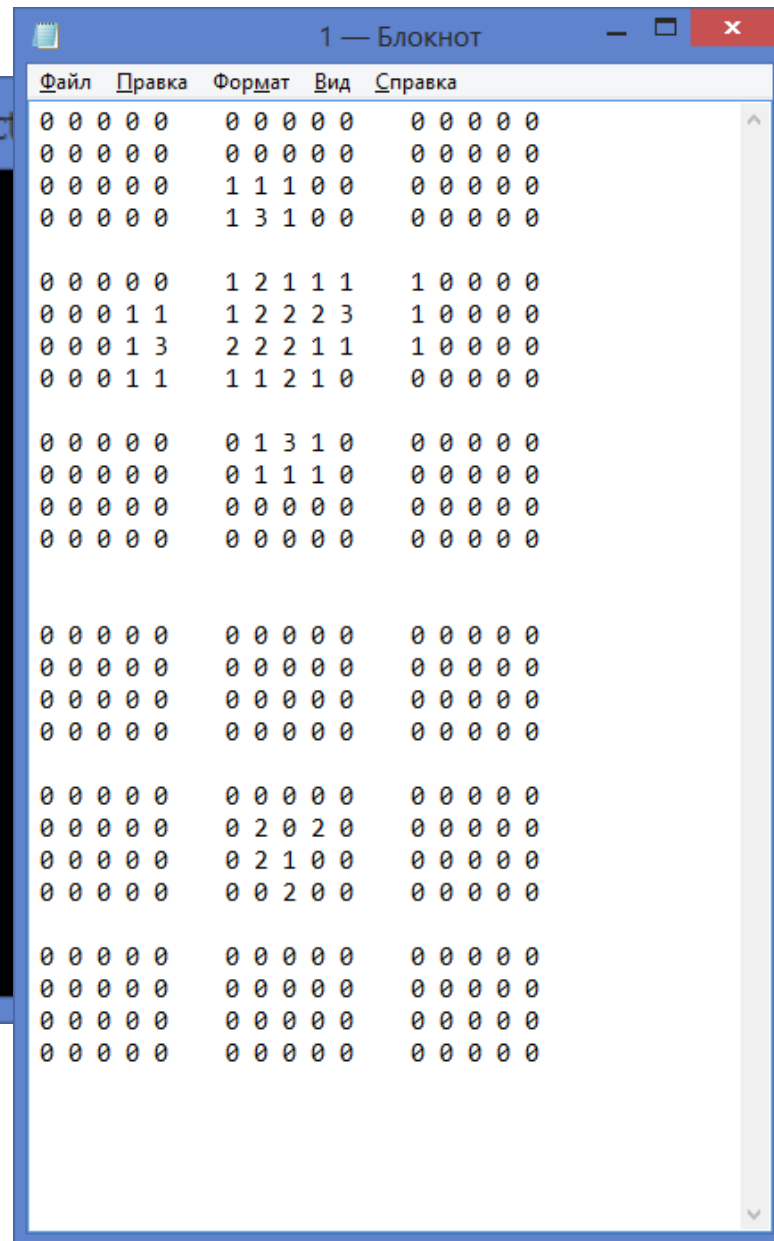
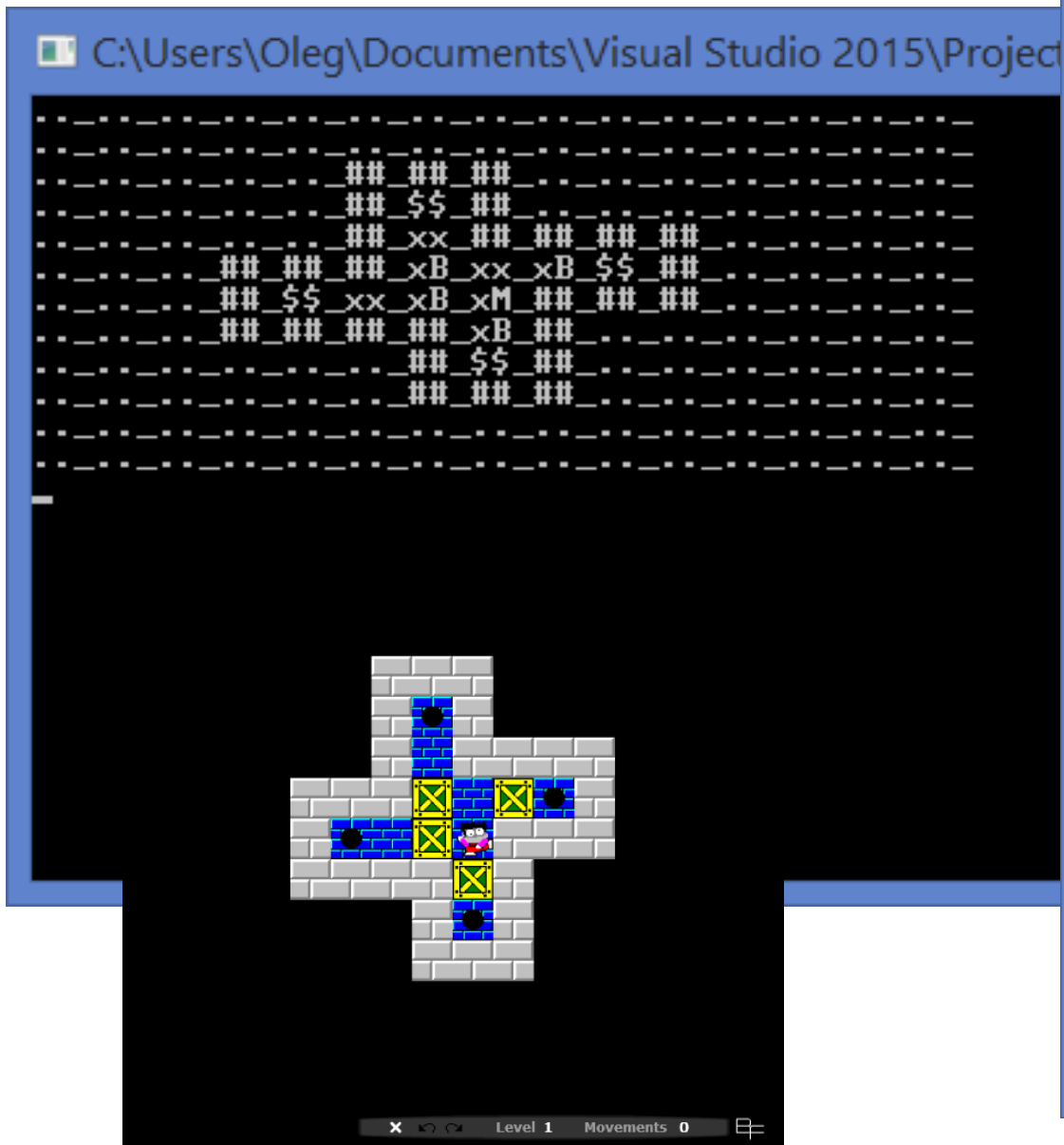
Печать игрового поля (результат работы)



Печать игрового поля (результат работы)



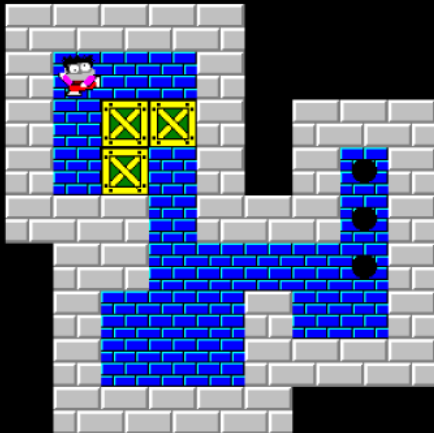
Печать игрового поля (результат работы)



Печать игрового поля 2 (результат работы)

C:\Users\Oleg\Documents\Visual Studio 2015\Projects\

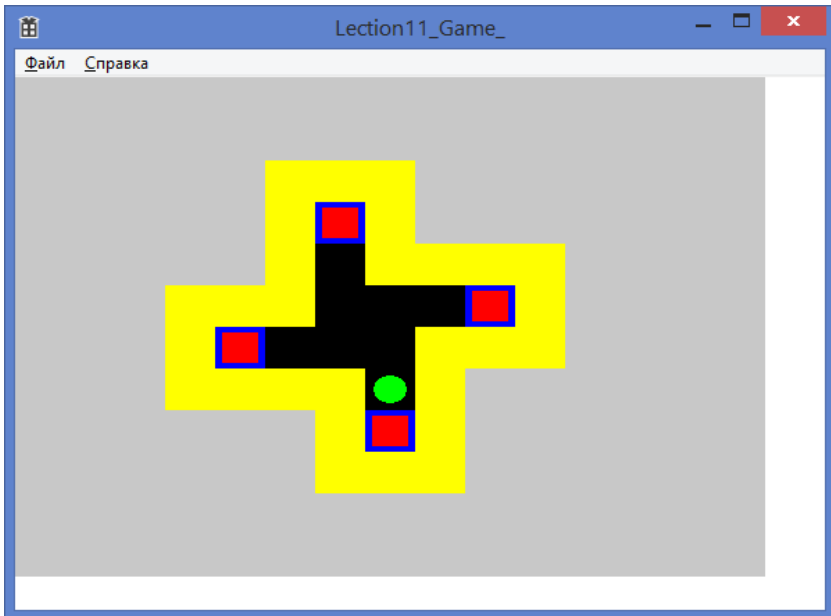
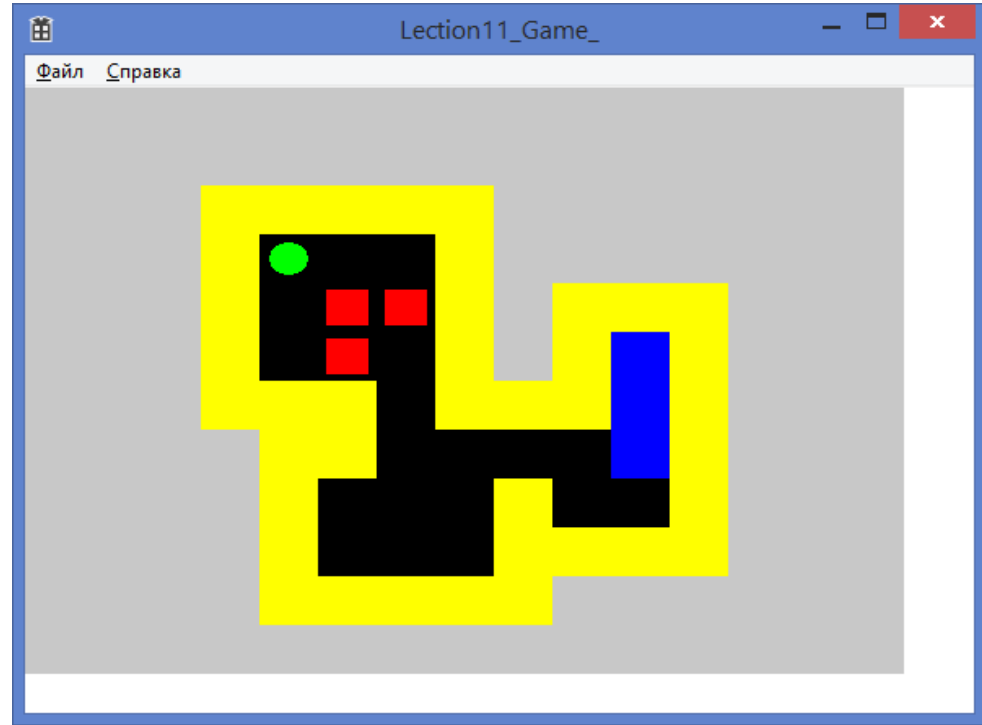
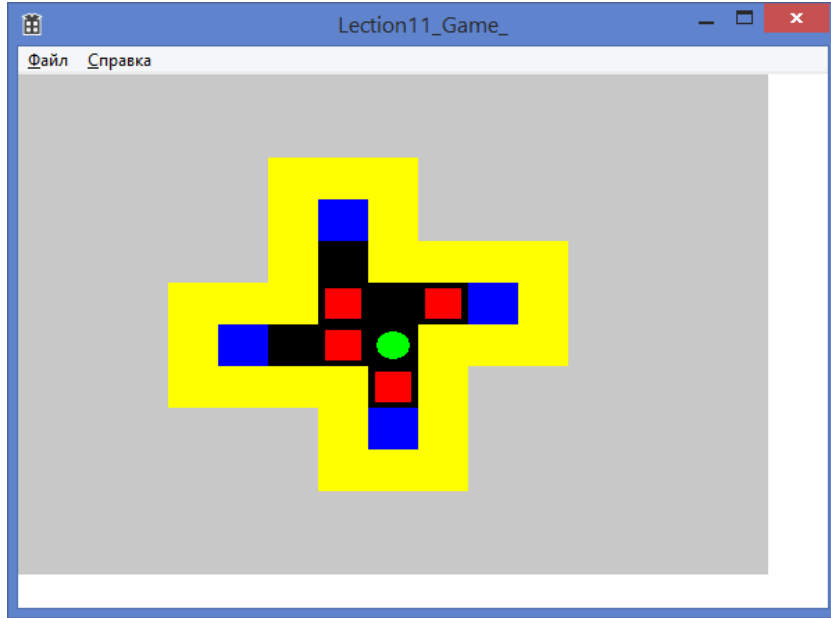
```
.....  
## ## ## ## ##  
## xM xx xx ##  
## xx xB xB ## .. ## ## ##  
## xx xB xx ## .. ## $$ ##  
## ## ## xx ## ## ## $$ ##  
## ## xx xx xx xx $$ ##  
## xx xx xx ## xx xx ##  
## xx xx xx ## ## ## ##  
## ## ## ## ## ..  
.....
```



Level 2 Movements 0

```
2 — Блокнот  
Файл  Правка  Формат  Вид  Справка  
  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 1 1    1 1 1 0 0    0 0 0 0 0  
0 0 0 1 2    2 2 1 0 0    0 0 0 0 0  
  
0 0 0 1 2    2 2 1 0 1    1 1 0 0 0  
0 0 0 1 2    2 2 1 0 1    3 1 0 0 0  
0 0 0 1 1    1 2 1 1 1    3 1 0 0 0  
0 0 0 0 1    1 2 2 2 2    3 1 0 0 0  
  
0 0 0 0 1    2 2 2 1 2    2 1 0 0 0  
0 0 0 0 1    2 2 2 1 1    1 1 0 0 0  
0 0 0 0 1    1 1 1 1 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 1    0 0 0 0 0    0 0 0 0 0  
  
0 0 0 0 0    2 2 0 0 0    0 0 0 0 0  
0 0 0 0 0    2 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0  
0 0 0 0 0    0 0 0 0 0    0 0 0 0 0
```

Собственно игра Sokoban



Состояние игры

```
#define N 12
```

```
#define M 15
```

```
int field[N][M];
```

```
int boxes[N][M];
```

```
#define LEVELS 3
```

```
int level = 0;
```

```
char levelFilename[LEVELS+1][80] = {  
    "c:\\Temp\\2016\\Lec11\\1.txt",  
    "c:\\Temp\\2016\\Lec11\\2.txt",  
    "c:\\Temp\\2016\\Lec11\\3.txt",  
    ""  
};
```

Загрузка состояния игры

```
int loadLevel(char filename[]) {  
    ...  
}  
  
...  
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)  
{  
    ...  
  
    loadLevel(levelFilename[level]);  
    ShowWindow(hWnd, nCmdShow);  
    UpdateWindow(hWnd);  
    return TRUE;  
}
```


Отрисовка поля игры

```
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam,
LPARAM lParam)
{
    switch (message)
    {
        ...
case WM_PAINT:
        {
            PAINTSTRUCT ps;
            HDC hdc = BeginPaint(hWnd, &ps);
            // TODO: Добавьте сюда любой код прорисовки, использующий HDC...

            DrawGame(hdc);

            EndPaint(hWnd, &ps);
        }
        break;
```

Управление грузчиком

case WM_KEYDOWN:

switch (wParam)

{

case **VK_DOWN:**

moveDown();

InvalidateRect(hWnd, NULL, TRUE);

break;

case **VK_LEFT:**

moveToLeft();

InvalidateRect(hWnd, NULL, TRUE);

break;

case **VK_UP:**

moveUp();

InvalidateRect(hWnd, NULL, TRUE);

break;

case **VK_RIGHT:**

moveToRight();

InvalidateRect(hWnd, NULL, TRUE);

break;

Переход на следующий уровень игры

```
case VK_RETURN:
    if (isFinish()) {
        level++;
        if (level >= LEVELS) {
            PostQuitMessage(0);
        }
    }

    if (!loadLevel(levelFilename[level])) {
        PostQuitMessage(0);
    }

    InvalidateRect(hWnd, NULL, TRUE);
break;

} // case WM_KEYDOWN:
break;
```

Проверка завершения выполнения миссии

```
int isFinish() {  
    int i, j;  
    for (i = 0; i < N; i++) {  
        for (j = 0; j < M; j++) {  
            if (boxes[i][j] == 2 // ящик  
                &&  
                field[i][j] != 3 // не на своем месте  
            ) {  
                return 0;  
            }  
        }  
    }  
    return 1;  
}
```

Отрисовка игры (1)

```
int sizeX = 36;
```

```
int sizeY = 30;
```

```
void DrawGame(HDC hdc) {
```

```
    HBRUSH hBrush[4] = {
```

```
        CreateSolidBrush(RGB(200, 200, 200)),
```

```
        CreateSolidBrush(RGB(255, 255, 0)),
```

```
        CreateSolidBrush(RGB(0, 0, 0)),
```

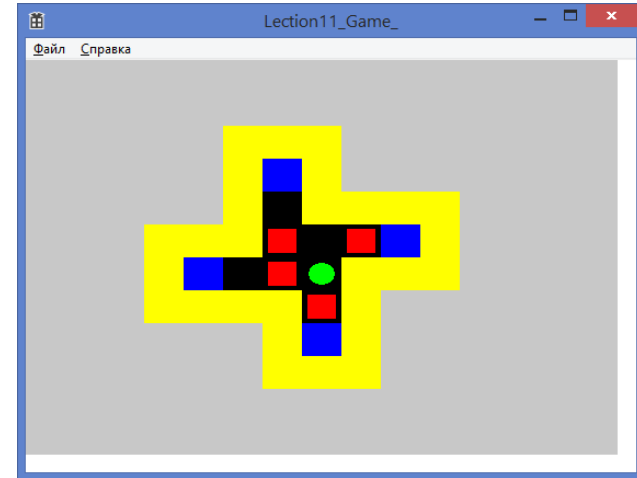
```
        CreateSolidBrush(RGB(0, 0, 255))
```

```
    };
```

```
    HBRUSH hBrushMen = CreateSolidBrush(RGB(0, 255, 0));
```

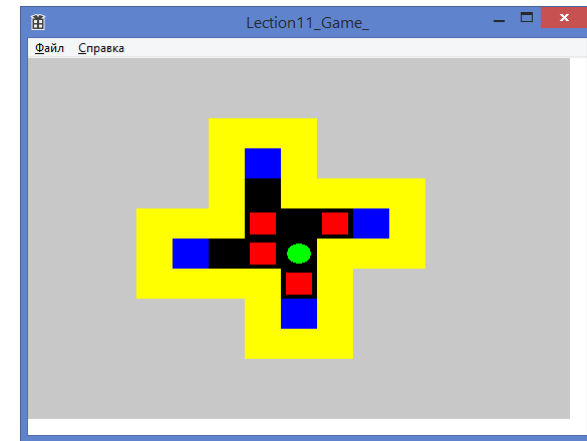
```
    HBRUSH hBrushBox = CreateSolidBrush(RGB(255, 0, 0));
```

```
    int i, j;
```



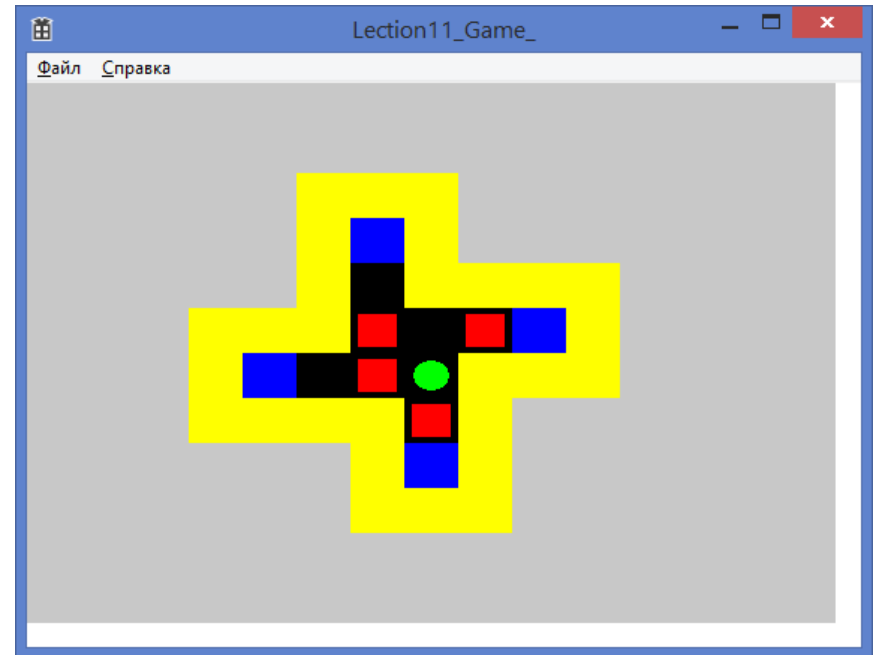
Отрисовка игры (2)

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < M; j++) {  
        RECT rect = {  
            j * sizeX,      i * sizeY,  
            (j + 1) * sizeX, (i + 1) * sizeY  
        };  
  
        FillRect(hdc, &rect, hBrush[field[i][j]]);  
  
        RECT rect2 = {  
            rect.left + int(sizeX * 0.15),  
            rect.top + int(sizeY * 0.15),  
            rect.right - int(sizeX * 0.15),  
            rect.bottom - int(sizeY * 0.15)  
        };  
    }  
}
```



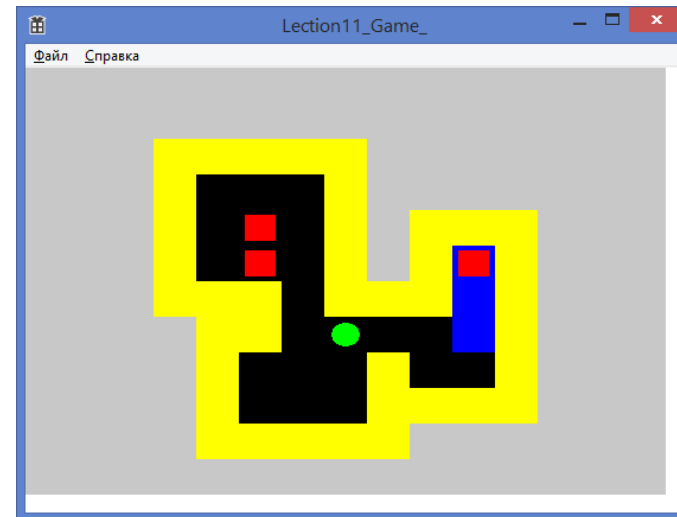
Отрисовка игры (3)

```
if (boxes[i][j] == 1) {  
    SelectObject(hdc, hBrushMen);  
    Ellipse(hdc, rect2.left, rect2.top,  
            rect2.right, rect2.bottom);  
}  
if (boxes[i][j] == 2) {  
    FillRect(hdc, &rect2, hBrushBox);  
}  
} // for (j = 0; j < M; j++) {  
} // for (i = 0; i < N; i++) {  
  
for (i = 0; i < 4; i++) {  
    DeleteObject(hBrush[i]);  
}  
DeleteObject(hBrushMen);  
DeleteObject(hBrushBox);  
}
```



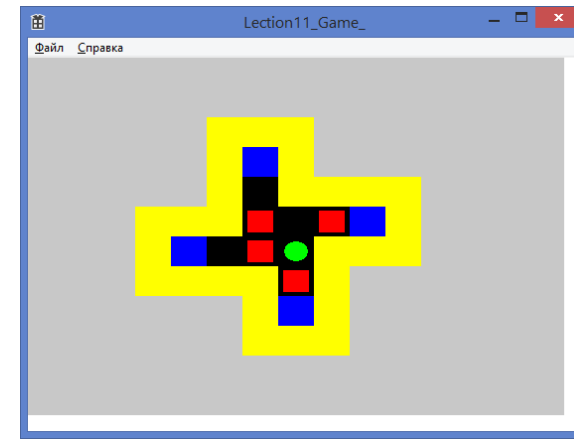
Перемещение грузчика – влево (1)

```
void moveToLeft() {  
    int i, j;  
    for (i = 0; i < N; i++) {  
        for (j = 0; j < M; j++) {  
            if (boxes[i][j] == 1) { // [i][j] = человек!  
                if (boxes[i][j - 1] == 0 // [i][j-1] = пусто!  
                    &&  
                    field[i][j - 1] >= 2 // [i][j-1] = игровые поля  
                ) {  
                    boxes[i][j - 1] = 1;  
                    boxes[i][j] = 0;  
                }  
            }  
        }  
    }  
}
```



Перемещение грузчика – влево (2)

```
else if (boxes[i][j - 1] == 2 // [i][j-1] = ящик!  
        &&  
        boxes[i][j - 2] == 0 // [i][j-2] = пусто!  
        &&  
        field[i][j - 2] >= 2 //[i][j-2]=игровые поля  
    ) {  
        boxes[i][j - 2] = 2;  
        boxes[i][j - 1] = 1;  
        boxes[i][j] = 0;  
    }  
    return; // человек единственный!  
} // [i][j] = человек!  
  
}  
  
}
```



Домашнее задание

*** Sokoban сделать из заготовок или самостоятельно

Источники информации

- msdn