

Создание простой игры в среде C++

ПЛАН:

- 1. Цели и задачи проекта**
- 2. Руководство пользователю**
- 3. Руководство программисту**
- 4. Листинг кода программы**

1. Цели и задачи проекта

Цель самостоятельной работы: создание программы для игры «Собери картинку».

Приложение должно обеспечивать следующую функциональность: выбор игроком любой картинке в формате .bmp, выбор количества плиток в одном ряду, перемешивание плиток, после окончания игры выдача отчета о количестве ходов, сделанных игроком.

В приложении будем использовать:

1) Главную форму - на неё поместим меню программы, выбранное игроком фото и игровое поле.

2) Меню программы будет состоять из следующих разделов:

- новая картинка;
- перемешать;
- настройка.

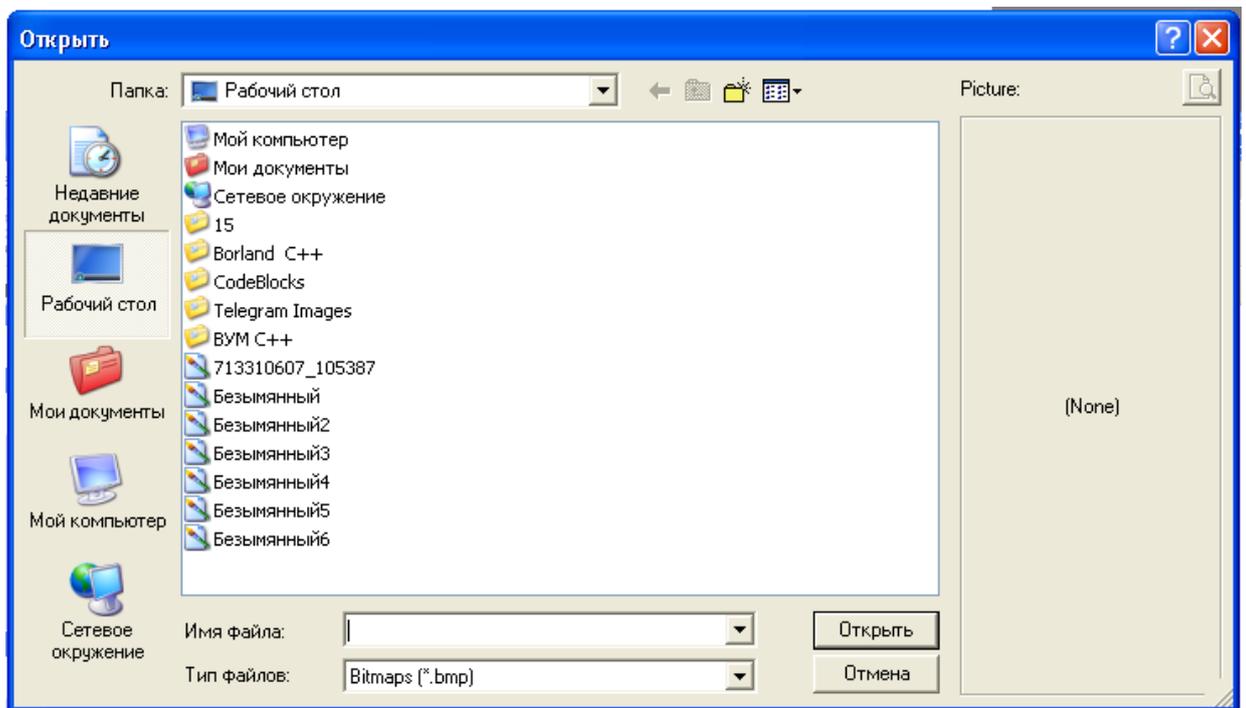
3) Диалоговое окно для выбора картинке.

4) Окно, выводящее результат игры.

2. Руководство пользователю

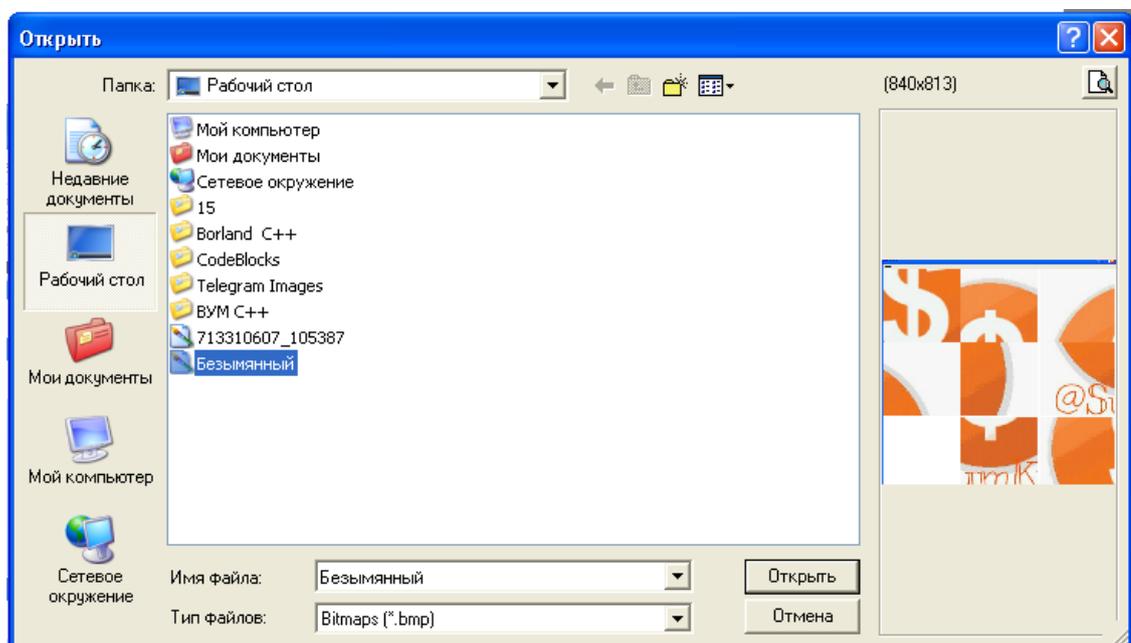
Запускаем программу.

Открывается диалоговое окно для выбора картинке:



В открывшемся окне программы можно увидеть:

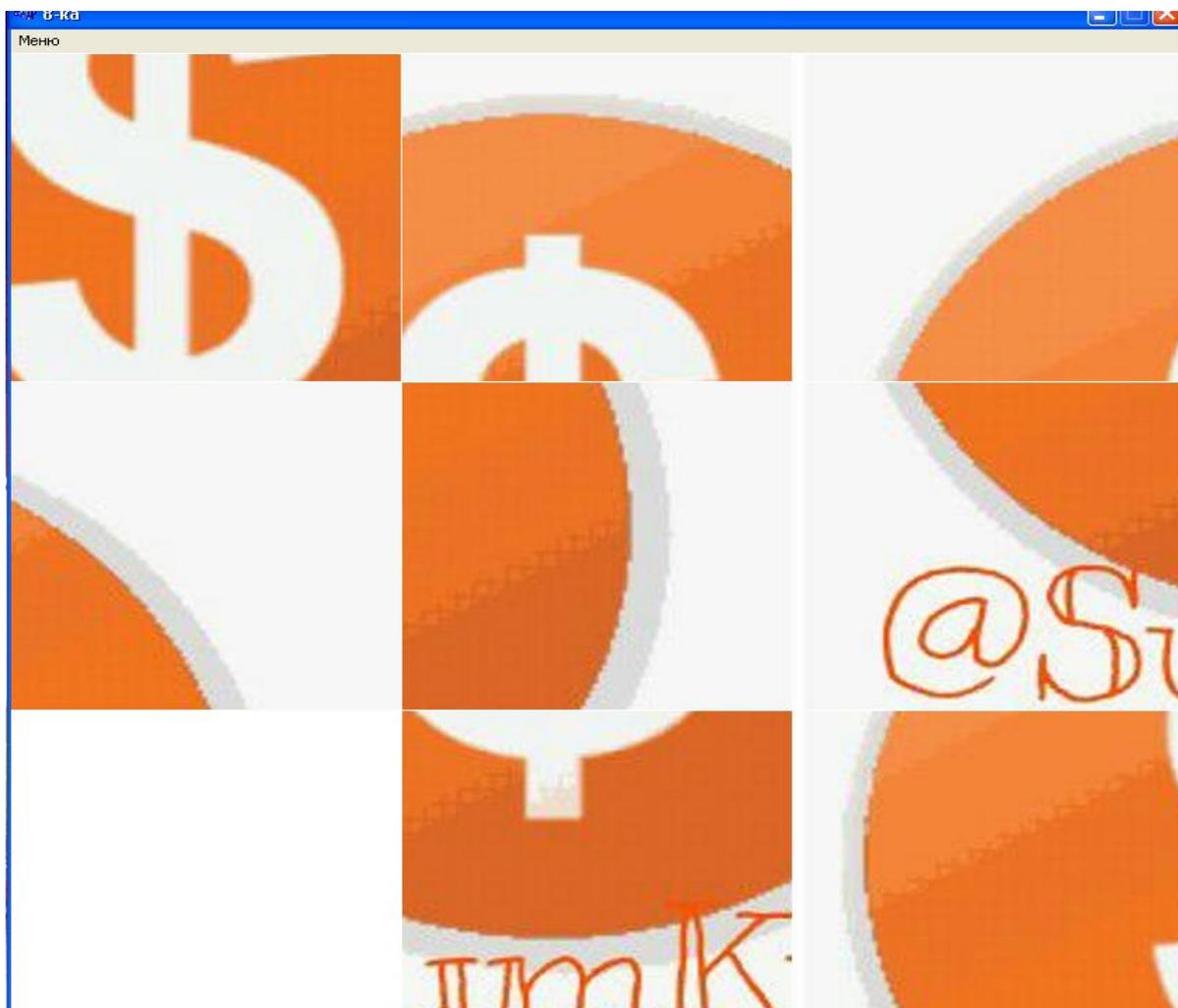
1. Поле выбора картинки
2. Поле имени файла
3. Поле типа файлов (.bmp)
4. Кнопка «Открыть»
5. Кнопка «Отмена»



После выбора картинки необходимо нажать на кнопку «Открыть».

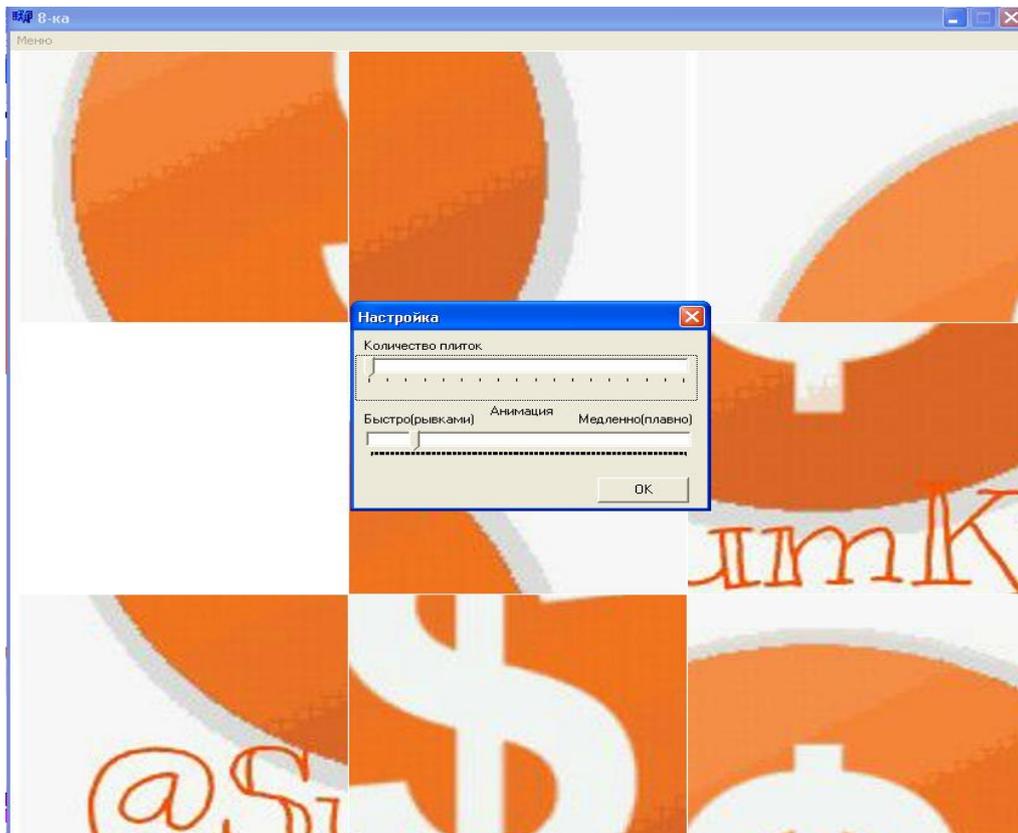
При нажатии кнопки «Открыть» откроется главная форма программы:

Здесь можно увидеть выбранную картинку и пункт меню.



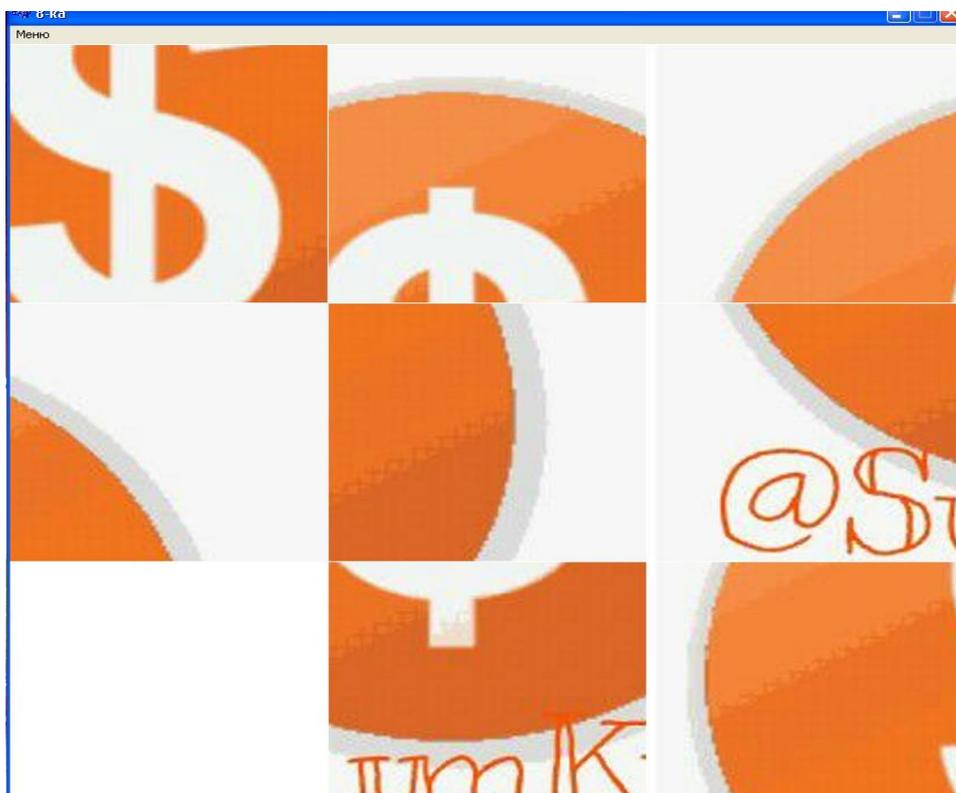
Раздел меню «Новая картинка» применяется при выборе другой картинки.

Нажимаем на раздел меню «Настройка». Откроется новое окно:



В открывшемся окне можно поменять количество плиток картинки и изменить уровень анимации. После окончания настройки программы нажимаем кнопку «ОК».

Далее выбираем раздел меню «Перемешать». Картинка перемешается в произвольной форме:



Нажимая и двигая нужные плитки соберем картинку:



В конце игры выходит окно с результатом игры и количеством сделанных ходов. При нажатии на кнопку «Да» программа перемешивает картинку заново, и игра начнется сначала.

3. Руководство программисту

Программа разработана на языке C++. Среда разработки C++ Builder.

В процессе создания программы также использованы следующие объекты:

TMainMenu;

TTimer;

TOpenPictureDialog;

TButton.

4. Листинг кода программы

```
#include <vcl.h>
#pragma hdrstop
#include<string>
#include<math.h>
#include <sstream>
#include<cstdlib>
#include<ctime>
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
using namespace std;
struct picture{
    int inf;
    Graphics::TBitmap*puzle;//=new Graphics::TBitmap
};
int h0,h1,x0,y0,n,time_h,time_N,sdvig1,k1,k2,l1,l2,kolich_hodov;
picture**a;
string filename;
bool timer1;
Graphics::TBitmap*pic=new Graphics::TBitmap;
Graphics::TBitmap*bitmap1=new Graphics::TBitmap;
//-----

template <typename T>
string toStr(T a1){
    std::ostringstream ost;
    ost << a1;
    std::string str = ost.str();
    return str;
}

TRect r(int x,int y,int h1,int h2){
    return Rect(x,y,x+h1,y+h2);
}

int random(int min,int max){
    return (rand()%(max-min+1)+min);
}

void peremesh(){
    kolich_hodov=0;
    int i1,j1,i2,j2;
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(a[i][j].inf==0){
                i2=i;
                j2=j;
            }
        }
    }
}
```



```

pic->Canvas->FillRect(r(0,0,h0*n,h1*n));
//pic->Canvas->Brush->Color=RGB(0,0,0);
//pic->Canvas->Font->Color=RGB(100,100,255);
pic->Canvas->Font->Size=h0/2;
for(int i=0;i<n;i++){
    for(int j=0;j<n;j++){
        if(a[i][j].inf!=0){
            if(timer1){

if((i>=k1&& i<=k2||i>=k2&& i<=k1)&&((j>=l1&& j<=l2)||j>=l2&& j<=l1)){
                double dx=0,dy=0;
                if(l2!=l1){
                    dx=time_h*h0/time_N;
                    if(l2>l1){
                        dx=-dx;
                    }
                }
                if(k2!=k1){
                    dy=time_h*h1/time_N;
                    if(k2>k1){
                        dy=-dy;
                    }
                }
                pic->Canvas->Brush->Color=RGB(255*time_h/time_N,0,0);
                pic->Canvas->Font->Color=RGB(100,100,255);

                pic->Canvas->Draw(x0+dx+j*h0,y0+dy+i*h1,a[i][j].puzle);
                //pic->Canvas->FillRect(r(x0+dx+j*h0,y0+dy+i*h0,h0-1,h0-1));
                //pic->Canvas-
                >TextOutA(x0+dx+j*h0+h0*(1.0/5),y0+dy+i*h0+h0*(1.0/5),toStr(a[i][j].inf).c_str());
            }
            else{
                pic->Canvas->Brush->Color=RGB(0,0,0);
                pic->Canvas->Font->Color=RGB(100,100,255);

                pic->Canvas->Draw(x0+j*h0,y0+i*h1,a[i][j].puzle);
                //pic->Canvas->FillRect(r(x0+j*h0,y0+i*h0,h0-1,h0-1));
                //pic->Canvas-
                >TextOutA(x0+j*h0+h0*(1.0/5),y0+i*h0+h0*(1.0/5),toStr(a[i][j].inf).c_str());
            }
        }
        else{
            pic->Canvas->Brush->Color=RGB(0,0,0);
            pic->Canvas->Font->Color=RGB(100,100,255);

            pic->Canvas->Draw(x0+j*h0,y0+i*h1,a[i][j].puzle);
            //pic->Canvas->FillRect(r(x0+j*h0,y0+i*h0,h0-1,h0-1));
            //pic->Canvas-
            >TextOutA(x0+j*h0+h0*(1.0/5),y0+i*h0+h0*(1.0/5),toStr(a[i][j].inf).c_str());
        }
    }
}
}
}

```

```

    Form1->Canvas->Draw(0,0,pic);
}
void iniz(){
    //a=NULL;
    if(filename==""){
        Form1->OpenPictureDialog1->Execute();
    }
    else{
        bitmap1->LoadFromFile(filename.c_str());
        h1=bitmap1->Height/n;
        h0=bitmap1->Width/n;
        a=new picture*[n];
        for(int i=0;i<n;i++){
            a[i]=new picture[n];
            for(int j=0;j<n;j++){
                a[i][j].puzle=new Graphics::TBitmap;
                a[i][j].puzle->Width=h0;
                a[i][j].puzle->Height=h1;
                //a[i][j].puzle->Canvas->Draw(-j*h0,-i*h0,bitmap1);
                a[i][j].puzle->Canvas->CopyRect(Rect(0,0,h0-1,h1-1),bitmap1-
>Canvas,Rect(j*h0,i*h1,(j+1)*h0,(i+1)*h1));
                a[i][j].inf=n*i+j+1;
                if(a[i][j].inf>n*n-1){
                    a[i][j].inf=0;
                }
            }
        }
        pic->Width=bitmap1->Width;
        pic->Height=bitmap1->Height;
        //peremesh();
        Form1->ClientWidth=bitmap1->Width;
        Form1->ClientHeight=bitmap1->Height;
        paint();
    }
}
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    filename="";
    time_N=30;
    timer1=false;
    srand(time(NULL));
    n=3;
    x0=0;
    y0=0;
    h0=50;
    iniz();
    Form1->ClientWidth=bitmap1->Width;
    Form1->ClientHeight=bitmap1->Height;
    string str=ToStr(n*n-1)+"-ka";
    Form1->Caption=str.c_str();
}

```

```

//-----
bool proverka(){
    for(int i=0;i<n;i++){
        for(int j=0;j<n;j++){
            if(a[i][j].inf!=n*i+j+1&& a[i][j].inf!=0){
                return false;
            }
        }
    }
    return true;
}
void __fastcall TForm1::FormPaint(TObject *Sender)
{
    if(filename!=""){
        paint();
    }
}
//-----
void __fastcall TForm1::FormMouseDown(TObject *Sender, TMouseButton Button,
    TShiftState Shift, int X, int Y)
{
    if(timer1==false){
        int j1=(X-x0)/h0,i1=(Y-y0)/h1,i2=-1,j2=-1;
        if(i1>=0&&i1<n&&j1>=0&&j1<n){
            for(int i=0;i<n;i++){
                if(a[i][j1].inf==0){
                    i2=i;
                    j2=j1;
                    break;
                }
            }
            if(a[i1][i].inf==0){
                i2=i1;
                j2=i;
                break;
            }
        }
        k1=i1;
        k2=i2;
        l1=j1;
        l2=j2;
        //string str=toStr(i1)+","+toStr(j1)+","+toStr(i2)+","+toStr(j2);
        //Form1->Caption=str.c_str();
        bool tim=false;
        if(i1==i2){
            if(j1>j2){
                for(int i=j2;i<j1;i++){
                    a[i1][i]=a[i1][i+1];
                }
                a[i1][j1].inf=0;
                tim=true;
            }
            else{
                for(int i=j2;i>j1;i--){

```



```

    }
    if(time_h<0){
        timer1=false;
        paint();
    }
}
//-----

void __fastcall TForm1::OpenPictureDialog1CanClose(TObject *Sender,
    bool &CanClose)
{
    filename=OpenPictureDialog1->FileName.c_str();
    iniz();
}
//-----

void __fastcall TForm1::N6Click(TObject *Sender)
{
    OpenPictureDialog1->Execute();
}
//-----

void __fastcall TForm1::N7Click(TObject *Sender)
{
    Form3->ShowModal();
}
//-----

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    int n1=n;
    time_N=Form3->TrackBar2->Position;
    n=Form3->TrackBar1->Position;
    Form1->ClientWidth=bitmap1->Width;
    Form1->ClientHeight=bitmap1->Height;
    string str=toStr(n*n-1)+"-ка";
    Form1->Caption=str.c_str();
    if(n1!=n){
        iniz();
    }
}
//-----

```