МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РФ

ФГБОУ ВО «Кубанский государственный   
аграрный университет имени И.Т. Трубилина»

Кафедра системного анализа и обработки информации

**Курсовая работа**

по дисциплине: Программирование

на тему «Разработка игровое приложение пазлы»

выполнил студент группы: БИ1602

Горишний Егор

Допущен к защите

Руководитель проекта Крамаренко Татьяна Анатольевна

Нормоконтролер Крамаренко Т.А.

(подпись, расшифровка подписи)

Защищен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Члены комиссии\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата, расшифровка подписи)

Краснодар - 2017

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РФ

ФГБОУ ВО «Кубанский государственный   
аграрный университет имени И.Т. Трубилина»

Кафедра системного анализа и обработки информации

**УТВЕРЖДАЮ:**

Зав. кафедрой Барановская Т.П.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ЗАДАНИЕ**

**на курсовую работу**

Студенту: БИ1602 группы 1 курса

Факультета прикладной информатики а Специальности: 38.03.05 – Бизнес-информатика

Горишний Егор ы

(ФИО)

Тема проекта: «Разработка игрового приложение пазлы»

Содержание задания: Программная реализация игры пазлы

м

Объем работы:

а) пояснительная записка к работе\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_листа формата А4

б) графическая часть\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_лист формата А4

Рекомендуемая литература: Мурлин А.Г., Ефанова Н.В., Иванова Е.А., Гончар О.М. – Программирование (С\С++) – Краснодар, 2012. – 117 с.

Срок выполнения проекта: с “ ” по “ ” 2017 г.

Срок защиты: “ ” 2017 г.

Дата выдачи задания: “ ” 2017 г.

Дата сдачи проекта на кафедру: “ ” 2017 г.

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_Крамаренко Татьяна Анатольевна, к.п.н..

(подпись, Ф.И.О., звание, степень)

Задание принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись, дата)

Краснодар – 2017

**РЕФЕРАТ**

35 стр., 16 рис.

ИГРА, ПАЗЛЫ, ГРАФИКА, ГРАФИКА, ПЯТНАШКИ, C++, Visual Studio, С++ Builder

Целью работы является разработка игрового приложения «Пазлы» в среде Visual Studio.

Объект исследования – создание игры пазлы с выбором любого изображения формата на основе классической игры пятнашки

Предмет исследования – объектно-ориентированные средства языков программирования. Разработанная программа позволяет играть в пазлы с любым изображением.

Также в программе присутствуют параметры режимы игры и возможность выбрать изображение в формате bmp. Программа работает в оконном режиме.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc484703952)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc484703953)

[1.1 Инструменты разработки 6](#_Toc484703954)

[1.2 Принцип работы генетических алгоритмов 6](#_Toc484703955)

[1.3 Подключение и работа с библиотекой SFML 9](#_Toc484703956)

[2 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ 12](#_Toc484703957)

[2.1 Структура и описание классов. 12](#_Toc484703958)

[2.2 Примеры и описание функций классов 17](#_Toc484703959)

[2.3 Принцип работы алгоритма искусственного интеллекта 18](#_Toc484703960)

[3 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 21](#_Toc484703961)

[ЗАКЛЮЧЕНИЕ 24](#_Toc484703962)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc484703963)

[ПРИЛОЖЕНИЕ 26](#_Toc484703964)

# **ВВЕДЕНИЕ**

В течение десятилетий пазлы использовались только для географических карт. Затем их стали использовать и для изучения других предметов, в частности сохранились пазлы из хронологической таблицы с датами правления английских королей и королев. Чего только в дальнейшем не разделяли на кусочки — портреты, изображения исторических сражений и даже библейские события.

Но XIX век внес в судьбу складных картинок свои коррективы: уже к его середине благодаря широкому распространению в Америке штамповки по картону пазлы значительно подешевели и стали доступны каждому. Это привело к широкой популярности пазлов во всем мире. Затем вместо черно-белых появились цветные картинки.

Однако пик популярности пазлов наступил в ХХ веке. Пазлы с художественными репродукциями стали изготавливать в качестве развлечения для взрослых. Игра стала популярным салонным развлечением в Европе.

Целью данной работы является разработка игрового приложения пазлы на основе игры пятнашки в среде разработке С++ Builder

Задачи данного проекта:

* Написать логику игры;
* Подобрать изображения;
* Создать удобный интерфейс;
* Сделать удобное меню;
* Реализовать систему оповещений;

1. **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Игровое приложение будет разработано на объектно-ориентированном языке программирования С++ в среде разработки C++ Builder. Для изображений будет использоваться точечный формат bmp.

* 1. **Инструменты разработки**

С++ компилируемый строго типизированный язык программирования общего назначения. Поддерживает разные парадигмы программирования: процедурную, обобщённую, функциональную; наибольшее внимание уделено поддержке объектно-ориентированного программирования.

Для работы используется среда разработки - C++ Builder - это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию.

Для создания пользовательского интерфейса используется. Встроенный дизайнер в C++ Builder.

* 1. **Принцип игры**

Игра в 15 или пятнашки, популярная головоломка, придуманная в 1878 году Ноем Чепмэном. Представляет собой набор одинаковых квадратных костяшек с нанесёнными числами, заключённых в квадратную коробку. Длина стороны коробки в четыре раза больше длины стороны костяшек для набора из 15 элементов, соответственно в коробке остаётся незаполненным одно квадратное поле. Цель игры - перемещая костяшки по коробке, добиться упорядочивания их по номерам, желательно сделав как можно меньше перемещений. Пример игры представлен на рисунке 1.



Рисунок – 1 Пример игры пятнашки

* 1. **Реализация пазла с изображением**

Для реализации пазла с изображением на основе пятнашек, мы будем загружать выбранное пользователем изображение в точечном формате windows bmp. После этого размер изображения будет поделен на равные 16 частей одна из которых будет пустой. Для каждого фрагмента будет представлен свой порядковый номер. Когда все фрагменты будут собраны в единую картинку, доесть номера клеток будут стоять по порядку, игра будет закончена. Передвижение клеток осуществляется по нажатии одной из них. Если с рядом с нажатой клетка есть пустая, то передвигаем её на место пустой клетки.

1. **ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ**
   1. **Описание разработки программы.**

Для того чтобы загрузить изображение мы создадим нужный компанент и вызовем у него функцию открытия окна выбора файла fn = OpenDialog1->FileName, имя выбранного файла сохраняем в переменную. После чего мы загрузим в выбранное изображения в переменную Graphics::TBitmap \*pic. Посоле чего размер окна изменится под размер изображения и посчитается размеры для клеток. Для управления фрагментами картинками создан массив с элементами присвоенными по своему порядку. После чего вызываем функцию перемешивания Mixer, которая передвигает элементы в матрице относительно пустого фрагмента под номером 16. При нажатии мышкой на экран считывается нажатие и подсчитываются координаты нажатой клетки, если клетка стоит рядом с пустой то мы её двигаем на место пустой клетки функцией Move, которая принимает координаты нажатой клетки. После каждого нажатия вызывается функция Finish проверяющая закончилась ли игра, она возвращает true если все элементы стоят по порядку и вызывает окно уведомления предлагающие пользователю начать новую игру или закрыть программу. Для отрисовки игры реализована функция ShowPole, которая вызывается автоматически при обновление окна.

* 1. **Примеры и описание основных функций**

**Описание** **функции** **Mixer:**

//---------------------------------------------------------------------------

//функция для перемешивания клеток

void TForm1::Mixer()

{

int x1, y1;//координаты пустой клетки

int x2, y2;//коориднаты этой клетки нужо переместить

int d;//направление перемещения относительно пустой клетки

Randomize();//оптимизируем функция генерации случайных чисел

x1 = 3, y1 = 3;//ставим положение координат

for (int i = 0; i < 150; i++)//передвигаем случайно клетки 150 раз

{

do

{

x2 = x1;//присваиваем координат перестановки

y2 = y1;//x и y

d = RandomRange(1,5);//задаем случайное направление

switch(d)// в зависимости от напрвления

{

//увеличиваем или уменьшаем нужные координаты

case 1: x2--; break;

case 2: x2++; break;

case 3: y2--; break;

case 4: y2++; break;

}

} while ((x2 < 0) || (x2 >= W) || (y2 < 0 || (y2 >= H)));//делаем это пока мы попадаем в клетки

//здесь мы определяем фишку которую

//надо переместить в пустую клетку

pole[y1][x1] = pole[y2][x2];

pole[y2][x2] = 16;

//присваиваем координаты

x1 = x2;

y1 = y2;

}

//запоминаем координаты пустой клетки

ex = x1;

ey = y1;

}

//---------------------------------------------------------------------------

**Описания функции Move:**

//---------------------------------------------------------------------------

//функция Move передвигает клетку, координаты которой в неё послали

void TForm1::Move(int cx, int cy)

{

if ((abs(cx - ex) == 1) && (cy - ey == 0) ||

(abs(cy - ey) == 1) && (cx - ex == 0))//если клека рядом с пустой клеткой

{

pole[ey][ex] = pole[cy][cx];//перемещаем клетки

pole[cy][cx] = 16;//ставим пустую

ex = cx;//запоминаем координаты пустой

ey = cy;//по x и y

ShowPole();//вызываем функцию отрисовки поля

if (Finish())//если игра закончена

{

GameOver = true;//стави true

ShowPole();//показываем поле

//показываем окно оповещения

int r = MessageDlg("Цель достигнута! Еще раз?", mtInformation,

TMsgDlgButtons() << mbYes << mbNo, 0);

if (r = mrNo)//если нажали no

Form1->Close();//закываем программу

else//иначе если нажали yes

{

NewGame();//запускаем новую игру

ShowPole();//рисуем поле

}

}

}

}

//---------------------------------------------------------------------------

**Описания функции ShowPole:**

void TForm1::ShowPole()

{

TRect src, dst;//фрагмент карттинки и область её отображения,

int sx, sy;//координаты для определения угла клетки

for (int i = 0; i < H; i++)

{ //проходимся по матрице через 2 цикла

for (int j = 0; j < W; j++)

{

sx = ((pole[i][j]-1) % W) \* wc;//преобразуем фрагмент картинки

sy = ((pole[i][j]-1) / H) \* hc;//в координаты левого верхнего угла

src = Bounds(sx, sy, wc, hc);//по координатам угла создаем фрагмент

dst = Bounds(j\*wc, i\*hc, wc, hc);//и область отображения

if ((pole[i][j]) != 16 || GameOver)//если клетка не пустая или игра окончена

Canvas->CopyRect(dst, pic->Canvas, src);//рисуем клетка с картинкой

else//иначе если клетка пустая

{

Canvas->Brush->Style = bsSolid;//стави стиль кисти

Canvas->Brush->Color = clBtnFace;//стави цвет кисти

Canvas->Rectangle(dst);//рисуем пустую клетку

}

}

}

if (N4->Checked)//если включени параметр номера

{

Canvas->Brush->Style = bsClear;//ставим кисти прозрачность

Canvas->Font->Size = 20;//задаем размер шрифта

Canvas->Font->Color = clWhite;//задаем цвет шрифта

for (int i = 0; i < H; i++)//проходмся по клеткам

for (int j = 0; j < W; j++)//через два цикла

Canvas->TextOutA(wc\*j, hc\*i, IntToStr(pole[i][j]));//и рисуемм цфиру в зависимости от номера клетки

}

}

//---------------------------------------------------------------------------

1. **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

При запуске программы появляется окно показанное на рисунке 2.

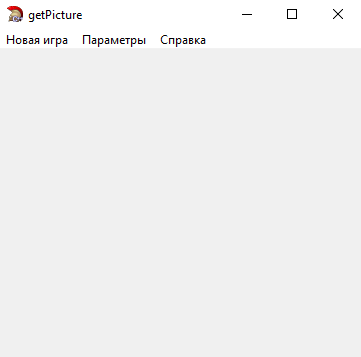


Рисунок – 2 Запуск программы

В пункте меню есть три кнопки, при нажатии на кнопку новый игры открывается окна с выборам файла. После выбора нужного изображение, оно загружается, разбивается на фрагменты и случайно растравливаются, пример можно увидеть на рисунке 3.

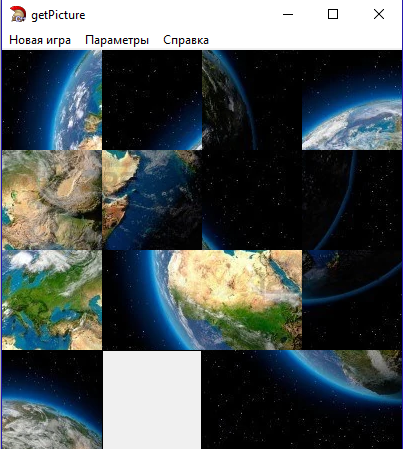


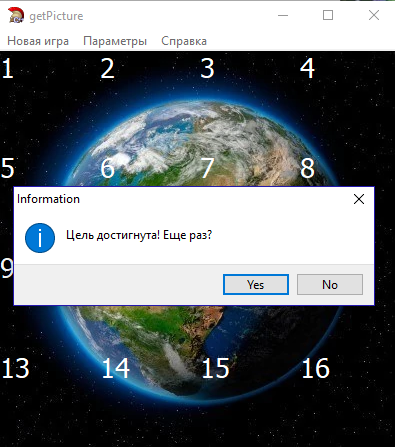
Рисунок – 3 Начало игры

Для перемещения фигурки нужно нажать на фрагмент рядом с пустой клеткой. Если во вкладки параметры включить параметр номера, то фрагментах появятся их порядковые нмоера как на рисунке 4.



Рисунок – 4 Нумерация фрагментов

После расстановки фрагментов в правильном порядке появится окно с оповещением победы, и предложением сыграть еще раз, пример на рисунке 5.



Рисонок – 5 Оповещение о победе

# **ЗАКЛЮЧЕНИЕ**

В заключении можно сказать, что работа была выполнена в полном объёме и были решены все поставленные задачи.

В данной курсовой работе можно выделить такие достоинства как:

* Приятный пользовательский интерфейс;
* Грамотная реализация функция;
* Возможность выбрать любое изображение;
* Поддержка любого разрешения;

Таким образом были выполнены следующие выше поставленные цели:

* Разработано игровое приложение пазлы;
* Улучшены навыки программирования;
* Освоена среда разработки С++ builder;
* Описана логика игры;

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Айвор Хортон. Visual C++ 2010. Полный курс. : Пер. с англ. – М. : Издательский дом «Вильямс», 2012. – 624 с.
2. Литвиненко Н.А. технология программирования на С++. Win32 API-приложения. – СПб. : БХВ-Петербург, 2013. – 288 ил.
3. Сидорина Т.Л. Самоучитель Microsoft Visual Studio C++ и MFC. – СПб.: БХВ-Петербург, 2013. – 848 ил.
4. Глушаков С.В. Программирование на С++ /, Т.В. Дуравкина. – изд. 2-е, доп. и переработ. – М. : АСТ, 2015. – 685 с.
5. Шилдт Г. C++ базовый курс. : Пер. с англ. – М. : Издательский дом «Вильямс», 2012. – 624 с.
6. Павловская Т. А. C/C++. Структурное программирование : Практикум / Т. А. Павловская, Ю.А.Щупак. – СПб. : Питер, 2011. – 240 с.
7. Подбельский В.В. Язык С++: Учебное пособие. – 5-е изд. -М. : Финансы и статистика, 2015. -560 с.

# **ПРИЛОЖЕНИЕ**

**Листинг кода**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit1.h"

#include "math.hpp"

#pragma package(smart\_init)

#pragma resource "\*.dfm"

//---------------------------------------------------------------------------

#define W 4

#define H 4

//---------------------------------------------------------------------------

TForm1 \*Form1;

byte wc, hc;

byte pole[H][W];

int ex, ey;

bool GameOver;

AnsiString fn;

TSearchRec SearchRec;

Graphics::TBitmap \*pic;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

pic = new Graphics::TBitmap();

}

//---------------------------------------------------------------------------

void TForm1::NewGame()

{

if (OpenDialog1->Execute())

{

fn = OpenDialog1->FileName;

}

try

{

pic->LoadFromFile(fn);

}

catch (EFOpenError &e)

{

MessageDlg("Ошибка при загрузке файла",

mtWarning, TMsgDlgButtons()<<mbOK<<mbHelp,0);

return;

}

wc = pic->Width / W;

hc = pic->Height / H;

ClientWidth = wc \* W;

ClientHeight = hc \* H;

int k = 1;

for (int i = 0; i < H; i++)

for (int j = 0; j < W; j++)

pole[i][j] = k++;

GameOver = false;

Mixer();

ShowPole();

}

//---------------------------------------------------------------------------

void TForm1::Mixer()

{

int x1, y1;

int x2, y2;

int d;

Randomize();

x1 = 3, y1 = 3;

for (int i = 0; i < 150; i++)

{

do

{

x2 = x1;

y2 = y1;

d = RandomRange(1,5);

switch(d)

{

case 1: x2--; break;

case 2: x2++; break;

case 3: y2--; break;

case 4: y2++; break;

}

} while ((x2 < 0) || (x2 >= W) || (y2 < 0 || (y2 >= H)));

pole[y1][x1] = pole[y2][x2];

pole[y2][x2] = 16;

x1 = x2;

y1= y2;

}

ex = x1;

ey = y1;

}

//---------------------------------------------------------------------------

void TForm1::ShowPole()

{

TRect src, dst;

int sx, sy;

for (int i = 0; i < H; i++)

{

for (int j = 0; j < W; j++)

{

sx = ((pole[i][j]-1) % W) \* wc;

sy = ((pole[i][j]-1) / H) \* hc;

src = Bounds(sx, sy, wc, hc);

dst = Bounds(j\*wc, i\*hc, wc, hc);

if ((pole[i][j]) != 16 || GameOver)

Canvas->CopyRect(dst, pic->Canvas, src);

else

{

Canvas->Brush->Style = bsSolid;

Canvas->Brush->Color = clBtnFace;

Canvas->Rectangle(dst);

}

}

}

if (N4->Checked)

{

Canvas->Brush->Style = bsClear;

Canvas->Font->Size = 20;

Canvas->Font->Color = clWhite;

for (int i = 0; i < H; i++)

for (int j = 0; j < W; j++)

Canvas->TextOutA(wc\*j, hc\*i, IntToStr(pole[i][j]));

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormMouseDown(TObject \*Sender, TMouseButton Button, TShiftState Shift,

int X, int Y)

{

int cx, cy;

cx = X / wc;

cy = Y / hc;

Move(cx, cy);

}

//---------------------------------------------------------------------------

void TForm1::Move(int cx, int cy)

{

if ((abs(cx - ex) == 1) && (cy - ey == 0) ||

(abs(cy - ey) == 1) && (cx - ex == 0))

{

pole[ey][ex] = pole[cy][cx];

pole[cy][cx] = 16;

ex = cx;

ey = cy;

ShowPole();

if (Finish())

{

GameOver = true;

ShowPole();

int r = MessageDlg("Цель достигнута! Еще раз?", mtInformation,

TMsgDlgButtons() << mbYes << mbNo, 0);

if (r = mrNo)

Form1->Close();

else

{

NewGame();

ShowPole();

}

}

}

}

//---------------------------------------------------------------------------

bool TForm1::Finish()

{

bool result;

int row, col;

int k = 1;

result = true;

for (row = 0; row < H; row++)

{

for (col = 0; col < W; col++)

{

if (pole[row][col] == k)

k++;

else

{

result = false;

break;

}

}

if(!result)

break;

}

return result;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::FormPaint(TObject \*Sender)

{

ShowPole();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N1Click(TObject \*Sender)

{

NewGame();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N3Click(TObject \*Sender)

{

ShowMessage("Автор: Егор Горишний БИ1602");

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::N4Click(TObject \*Sender)

{

N4->Checked = !N4->Checked;

ShowPole();

}

//---------------------------------------------------------------------------