Министерство образования и науки Российской Федерации

Агентство по образованию

Тихоокеанский государственный экономический университет

Экономический институт

**«Кратчайший путь передвижения короля по шахматному полю»**

Выполнил: Прудникова Л.И.

Владивосток 2009

Введение

Условие решаемой задачи дословно по заданию звучит следующим образом: «найти кратчайший путь передвижения короля по заданному клеточному полю, соединяющих два заданных поля доски»

Целью представленной работы является разработка приложения “Поиск кратчайшего пути”, которое создает шахматную доску, находит кратчайший путь передвижения короля и отображает его.

Перед началом вычисления пользователь должен указывать в программе следующую информацию:

- размерность поля

- установить слона на начальную позицию и указать конечную (при помощи мыши)

После этого программа должна показать кратчайший путь (пути) движения короля, выделяя его другим цветом.

Неформальная постановка задачи

Задачу ставит заказчик, а принимает разработчик. Между разработчиком и заказчиком должен состояться диалог, цель которого выяснить, одинаково ли они понимают задачу. Диалог может длиться день, месяц… В моей ситуации требуется написать программу, чтобы она могла найти кратчайший путь передвижения короля по заданному клеточному полю, соединяющих два заданных поля доски. Причём начальная и целевая клетки указываются при помощи мыши в запущенной программе.

Формальная постановка задачи

Разработка или поиск алгоритма решения задачи

Проект программы:

Задаём размер поля n.

Проверим, чтобы король и его местоположение должны находиться на полях одного цвета.

Образуем матрицу для расчёта пути размерности n+1.

С помощью полученной матрицы во избежание выхода за шахматное поле заполним окаймляющие элементы значениями false.

Выберем все возможные клетки, когда король из начального местоположения и точки назначения, используя ранее полученную матрицу.

Спецификация функций программы

В появившемся при вызове программы окне вводим размерность поля.

а) если мы вводим размерность поля меньше или больше указанного диапазона, то выводится сообщение

В главном окне отмечаем расположение короля и цели. Отметив на поле короля, мы делаем проверку, чтобы пользователь не мог отметить клетку-цель на поле, не совпадающее цвету, поля короля, а также на поле где расположен сам король.

Тем самым указываются все случаи, которые могут возникнуть при решении задачи на компьютере. Каждый такой случай оформляется как некоторая функция (в смысли достижения цели)

Руководство пользователя

При разработке приложения применялся принятый в среде Delphi объектно-ориентированный подход реализации интерфейса. При реализации алгоритмов обработки данных использовался структурный подход при проектировании к написании программ приложения.

В появившемся при вызове программы окне вводим размерность поля.

В главном окне отмечаем расположение слона и цели. Отметив на поле короля, мы делаем проверку, чтобы пользователь не мог отметить клетку-цель на поле, не совпадающее цвету, поля короля, а также на поле где расположен сам короля.

Если мы вводим размерность поля меньше или больше указанного диапазона, то выводится сообщение

Проектирование программы

Все предшествующие этапы необходимы для сбора информации, используемой для проектирования программы. Проект программы это не сама программа, а её эскиз, в котором чётко зафиксированы все функции задачи, как некоторые её подзадачи.

Нахождение кратчайшего пути передвижения короля от начальной позиции до целевой клетки

Размерность поля (n), начальное месторасположение короля, целевая клетка

Кратчайший путь (пути)

Программирование

procedure TForm1.Button1Click(Sender: TObject);

var code: integer; // Сюда функция val запишет ошибку, в случае ее возникновения

begin

val (edit2.text, razmerY, code); // Получаем размер поля

val (edit1.text, razmerX, code); // из текстовых полей

// В случае возникновения введенного числа возможному размеру поля надо выдать ошибку и завершить выполнение процедуры

if ((razmerX<4) or (razmerX>25)) then begin application.MessageBox('Неправильная циферка!', 'Шахматы', MB\_APPLMODAL); exit; end;

Form2.execute(razmerX, razmerY); // Передаем данные на Form 2

form2.ShowModal; // Показываем Form 2

end;

procedure TForm1.Edit1Change(Sender: TObject);

begin

edit2.text:=edit1.text; // Поле - квадрат

end;

procedure TForm1.FormPaint(Sender: TObject);

begin

if unit2.tf=true then self.Close; // Если пользователь нажимает Выход на Form 2, нужно завершить работу проги

end;

end.

procedure Execute(x, y: integer);

function max(x, y:integer):integer;

procedure procClick(sender: tobject);

procedure dda\_line(x1, y1, x2, y2:integer);

end;

var

Form2: TForm2;

img: array[1..20, 1..20] of timage;

etap: integer;

korolX, korolY: integer;

nadoX, nadoY:integer;

razmerX, razmerY:integer;

tf: boolean;

implementation

{$R \*.dfm}

procedure TForm2.dda\_line(x1,y1, x2, y2: integer);

// Процедура получает координаты конца и начала и «рисует» линию между ними по алгоритму ДДА

var

i, L, xstart, ystart, xend, yend: integer;

dx, dy: real;

x, y: array [0..1000] of real;

begin

xstart:=x1;

ystart:=y1;

xend:=x2;

yend:=y2;

L:=max(abs(xend-xstart), abs(yend-ystart));

dx:=(x2-x1)/L;

dy:=(y2-y1)/L;

i:=0;

x[i]:=x1;

y[i]:=y1;

inc(i);

while(i<L) do begin

x[i]:=x[i-1]+dx;

y[i]:=y[i-1]+dy;

inc(i);

end;

x[i]:=x2;

y[i]:=y2;

i:=1;

// В клетки, где проходит линия, загрузим голубую картинку

while (i<L) do begin

img[round(x[i]), round(y[i])].Picture.LoadFromFile('put.bmp');

inc(i); end;

// В мемо запишем текущую информацию для пользователя

memo1.Clear;

memo1.Lines.Add('Вам требуется' + inttostr(L) + ' ходов:))');

//

// Изменим внешний вид и функцию кнопки под мемо

bitbtn1.Kind:=bkRetry;

bitbtn1.Caption:=' Еще раз!';

bitbtn1.tag:=2;

end;

procedure TForm2.procClick(sender: tobject);

// Процедура постановки короля на поле и указания клетки - цели

var x, y:integer;

begin

//

// Получаем координаты по tag

x:=(sender as timage).Tag div 100;

y:=(sender as timage).tag mod 100;

// Если это постановка короля, загрузить «короля» в выбранную клетку

if etap=postanovka then begin

if ((x +y) mod 2)=0 then (sender as timage).Picture.LoadFromFile('krch.bmp')

else

(sender as timage).Picture.LoadFromFile('krbl.bmp');

etap:=selectPlace; // Теперь переходим к указанию места назначения

korolX:=x;

korolY:=y;

memo1.Clear;

memo1.Lines.add('Укажите целевую клетку');

end;

// Мы выбрали клетку - цель

if etap=selectPlace then begin

// Если аккурат в этой клетке стоит король, выходим из процедуры, пользователь должен выбрать другое место

if ((korolX=x) and (korolY=y)) then exit;

// Загрузим в клетку маркер

(sender as timage).Picture.LoadFromFile('zel.bmp');

nadoX:=x;

nadoY:=y;

// Изменим значение переменой etap, чтобы больше ничего нельзя было делать: ни расставлять короля, ни выбирать клетку

inc(etap);

// И прочертим линию от короля до клетки – назначения – это и будет кратчайший путь

dda\_line(korolX, korolY, nadoX, nadoY);

end;

//else

end;

procedure TForm2.Execute(x, y: integer);

// Процедура вызывается, когда от пользователя получена информация о размерах X и Y поля

var wid, i, j: integer;

begin

razmerX:=x;

razmerY:=y;

etap:=postanovka; // Сейчас пользователь будет ставить короля, развернем окно на весь экран

self.WindowState:=wsMaximized;

//

// Высчитываем размер клетки исходя из размеров окна и количества клеток

wid:=self.Width div (max(x, y)+2);

for i:=1 to x do

for j:=1 to y do begin

img[i,j]:=timage.create(self); // Выделяем память

img[i,j].Parent:=self; // Указываем родителя, чтобы клетка была именно на Form 2, а не где - то

img[i,j].Top:=j\*wid;

img[i,j].Stretch:=true; // Чтоб весь рисунок помещался

img[i,j].Left:=i\*wid;

// В зависимости от координат поля загружаем то черный, то белый фоны для клеток

if ((i+j) mod 2)=0 then

img[i,j].picture.loadfromfile('ch.bmp') //:=inttostr(x+y);

else

img[i,j].picture.loadfromfile('bl.bmp');//:=inttostr(x+y);

img[i,j].Width:=wid;

img[i,j].Height:=wid;

img[i,j].Tag:=i\*100+j;

// Устанавливаем обработчик нажатия на клетку

img[i,j].onclick:=procClick;

end;

// Для пущей верности

self.WindowState:=wsMaximized;

// Снабдим мемо текущей информацией

memo1.Clear;

memo1.Lines.Add('Укажите начальное место расположения короля');

memo1.Left:=wid\*(x+2);

memo1.Font.Name:='Times New Roman';

memo1.Font.Size:=25;

memo1.Width:=round(form2.width /2.5);

memo1.Height:=self.Height div 2;

// Придадим кнопке под мемо соответствующий вид

bitbtn1.Left:=memo1.left;

bitbtn1.top:=memo1.Top+memo1.Height;

bitbtn1.Width:=memo1.Width-25;

bitbtn1.Tag:=1;

bitbtn1.Kind:=bkCancel;

bitbtn1.Font:=memo1.font;

bitbtn1.height:=memo1.font.size\*2;

bitbtn1.Caption:='Выход';

end;

procedure TForm2.Button1Click(Sender: TObject);

begin

// //

end;

procedure TForm2.FormCreate(Sender: TObject);

begin

self.WindowState:=wsMaximized;

unit2.tf:=false;

end;

procedure TForm2.BitBtn1Click(Sender: TObject);

var i, j:integer;

begin

// Если tag=1, то выходим из программы

if (sender as tbitbtn).tag=1 then begin self.close; tf:=true; end;

// Иначе вновь загружаем Form 1 для указания размеров

for i:=1 to razmerX do

for j:=1 to razmerY do

img[i,j].Free;

self.WindowState:=wsNormal;

self.Visible:=false;

end;

end.

Тестовый пример программы

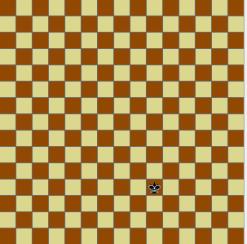


Рис.1. Рабочее окно приложения – ставим короля

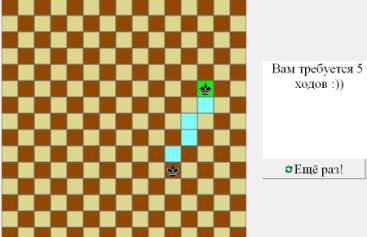


Рис.2. Рабочее окно приложения – нахождение кратчайшего пути шахматного короля

Заключение

Результатом работы над курсовой работой создано приложение среде Delphi, которое находит в нем кратчайший путь и визуализирует его на форме приложения. Приложение является полупрофессиональным. Выполненные многочисленные тестовые примеры позволяют утверждать, что надежность программного обеспечения проекта довольно высока.

Список литературы

1. Иванов Б.Н. Дискретная математика. Алгоритмы и программы: Учеб. Пособие. – Владивосток: Изд - во ДВГТ, 2000. – 288с.
2. Молчанова Л.А., Прудникова Л.И. Delphi в примерах и задачах: Учеб. пособие. Владивосток: Изд - во ТГЭУ, 2006. – 92с.