**Міністерство освіти та науки України  
Національний технічний університет України  
«Київський політехнічний інститут»  
Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих  
комп’ютерних систем**

**Розрахунково-графічна робота**

**з дисципліни «Програмування»**

Виконав: Шевель Олександр Сергійович

Студент групи КВ-42

Перевірив(ла)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Постановка задачі

Створити ігрову програму мовою Паскаль.

Розробка і реалізація ігрових програм має вестися з врахуванням графічних та звукових можливостей, що надаються конкретним комп’ютером.

Програма мусить коректно розв’язувати поставлену задачу. Логічно відокремлені частини алгоритма реалізувати за допомогою окремих процедур або функцій.

Також потрібно передбачити та забезпечити виконання всіх можливих розгалужень алгоритма, тобто програма повинна коректно реагувати на будь-які можливі ситуації (наприклад, виникнення помилкових ситуацій, перевірка файлів на порожність, правильність введених з клавіатури значень і т.д.). Передбачити взаємодію з користувачем (наприклад, можливість виводу правил гри, допомоги), таймер, лічильник числа ходів.

*Завдання варіанту 24:*

«Задача про голандський прапор». Вздовж дошки лежать кулі трьох кольорів – червоні, сині, білі. Одним ходом дозволяється міняти місцями дві будь-яких кулі. Добитися того, щоб всі червоні кулі йшли першими, посередині – білі, останні – сині. Якщо загальна кількість лунок дорівнює n, то для розв’язання задачі достатньо зробити n-1 кроків.

Описання структури програми

**procedure**DrawField(Size : word)– процедура, яка використовуючи клас ObjectBoardABC, який є нащадком класу ObjectABC, створює поле для гри. Крім того ця процедура виставляє початкові настройки вікна.

**procedure**CreateMenu– створює усі об’єкти, що потрібні для меню – написи, кнопки, малюнки, використовуючи клас ObjectABC та його нащадків. Зчитуючи правила гри з файла виводить їх на екран.

**procedure**Menu– власне меню, яке зчитуючи координати місця, де була натиснута ліва кнопка миші за допомогою методу класу ObjectABC– PtInside перевіряє, на який об’єкт натиснув гравець. В залежності від цього виконуються відповідні дії.

**procedure**MouseDown(x, y, mb: integer) – процедура-обробник події, яка реагує на натиснення кнопки миші.

**procedure** Check; - процедура якак перевіряє поле на виграш після кожного кроку, та виводить повідомлення про виграш.

**procedure** DrawBounce; -процедура що заповнює поле кульками.

**procedure** KeyDown(Key: integer); - процедура що зчитує натиснення клавіш під час гри, та віконує відповідні дії.

**procedure** Select\_Object; - процедура яка створює рамки навколо вибраних елементів, що допомогає покращує взаємодію з гравцем.

**Схематичний алгоритм використання процедур, їх взаємодії:**

**Блок меню**

**Блок гри**

Блок меню

Блок

**CreateMenu**

**DrawField**

**MouseDown**

**DrawBounce**

**Menu**

**Select\_Object**

**KeyDown**

**Check**

Код програми

**Program** RGR;

**uses** GraphABC, ABCObjects;

**type**

Coord = **record**

x, y: byte;

**end**;

**var**

RedB,WhiteB,BlueB :PictureABC;

ChangeSizeM, ChangeSizeP :PictureABC;

Background :PictureABC;

SizeT,Finish,Title,T1,RulesT,Rules,Start :TextABC;

Point,ChoosePoint :SquareABC;

Field :ObjectBoardABC;

y1,x1,xMouse,yMouse:integer;

Size,px,py,b: word;

FlChoose: boolean;{false = fail, true = victory}

sound := **new** system.Media.SoundPlayer;

Arr:**array**[1..9,1..9]**of** byte;

**procedure** DrawField(Size:word); **forward**;

**procedure** Select\_Object; **forward**;

**procedure** DrawBounce; **forward**;

//MENU PROCEDURE

**procedure** MouseDown(x, y, mb: integer);

**begin**

**if** mb = 1 **then**

**begin**

xMouse := x;

yMouse := y;

**end**;

**end**;

**Procedure** DrawMenu;

**var**

s: string;

f:text;

ch:char;

**begin**

ClearWindow;

Size:=480;

SetWindowSize(800,500);

Background:=**new** PictureABC(0,0,'Menu.jpg');

Background.Width:=800;

Background.Height:=500;

Title:=**new** TextABC(160,50,30,'Голанский флаг',clWhite);

Title.FontSize:=50;

Title.FontStyle:=fsitalic;

SetFontName('Georgia');

T1 := **new** TextABC(250, 350, 20, 'Изменить размер поля', clWhite);

ChangeSizeM:= **new** PictureABC(250, 395, 'minus.png');

str(Size **div** 80,s);

SizeT := **new** TextABC(380, 400, 15, s + 'x' + s, clWhite);

ChangeSizeP:= **new** PictureABC(500, 395, 'plus.png');

Start:=**new** TextABC(340,190,40,'Start',clBlack);

Rules:=**new** TextABC(365,280,20,'Rules',clBlack);

Finish:=**new** TextABC(650,420,23,'Выход',clWhite);

assign(f,'rules.txt');

reset(f);

s := '';

**while not** eof(f) **do**

**begin**

read(f,ch);

s := s + ch;

**while not** eoln(f) **do**

**begin**

read(f,ch);

s := s + ch;

**end**;

**end**;

SetFontName('Georgia');

RulesT := **new** TextABC(120,25,15,s, clBlack);

RulesT.Visible := false;

RulesT.FontName := 'Georgia';

**end**;

**procedure** Menu;

**var**

s: string;

FRules:boolean;

**begin**

FRules:=false;

DrawMenu;

**while** true **do**

**begin**

xMouse := -1;

**while** xMouse = -1 **do**

OnMouseDown := MouseDown;

**if** ChangeSizeM.PtInside(xMouse, yMouse) **then**

**if** Size > 240 **then**

**begin**

Size := Size - 240;

str(Size **div** 80, s);

SizeT.Text := s + 'x' + s;

**end**;

**if** ChangeSizeP.PtInside(xMouse, yMouse) **then**

**if** Size <720 **then**

**begin**

Size := Size + 240;

str(Size **div** 80, s);

SizeT.Text := s + 'x' + s;

**end**;

**if** Rules.PtInside(xMouse, yMouse) **then**

**begin**

**if** FRules=false **then**

**begin**

Rules.Text:='Back to menu';

Rules.MoveOn(-50,0);

**end**

**else**

**begin**

Rules.Text:='Rules ';

Rules.MoveOn(50,0);

**end**;

ChangeSizeM.Visible:=**not** ChangeSizeM.Visible;

ChangeSizeP.Visible:=**not** ChangeSizeP.Visible;

Start.Visible:=**not** Start.Visible;

T1.Visible:=**not** T1.Visible;

Finish.Visible:=**not** Finish.Visible;

SizeT.Visible:=**not** SizeT.Visible;

Title.Visible:=**not** Title.Visible;

RulesT.Visible:=**not** RulesT.Visible;

FRules:=**not** FRules;

**end**;

**if** Start.PtInside(xMouse, yMouse) **then**

**begin**

ClearWindow;

DrawField(Size);

DrawBounce;

Select\_Object;

**end**;

**if** Finish.PtInside(xMouse, yMouse) **then**

halt;

**end**;

**end**;

//GAME PROCEDURE

**procedure** Check;

**var** Sq:SquareABC; CheckWin:boolean; i,j:byte;

**begin**

CheckWin:=true;

**for** i:=1 **to** Size **div** 80 **do**

**for** j:=1 **to** Size **div** 80 **do**

**begin**

**if** i<=size **div** 80/3 **then**

**if** Arr[i,j]<>2 **then begin** CheckWin:=false; **break**; **end**;

**if** ((i>size **div** 80/3) **and** (i<=2\*(size **div** 80/3))) **then**

**if** Arr[i,j]<>1 **then begin** CheckWin:=false; **break**; **end**;

**if** ((i>2\*(size **div** 80/3)) **and** (i<=size **div** 80)) **then**

**if** Arr[i,j]<>0 **then begin** CheckWin:=false; **break**; **end**;

**end**;

**if** CheckWin=true **then**

**begin**

Sound.SoundLocation:='victory.wav';

Sound.Play;

Sq:=**new** SquareABC(0,0,Size);

Sq.Filled:=false;

Sq.BorderColor:=clLightGreen;

Sq.BorderWidth:=5;

Sq.FontStyle:=fsBoldItalic;

Sq.FontColor:=clGreen;

Sq.Text:='You WIN!!!';

**end**;

**end**;

**procedure** DrawField;

**begin**

ClearWindow;

SetWindowSize(Size, Size);

CenterWindow;

SetWindowIsFixedSize(true);

SetWindowCaption('РГР, вариант 24. Шевель О.С, "Голандский флаг"');

Field := **new** ObjectBoardABC(0, 0, Size **div** 80, Size **div** 80, 80, 80, clwhite);

Field.BorderWidth := 1;

Background:= **new** PictureABC(0,0,'Background.png');

Background.Width:=Size;

Background.Height:=Size;

BlueB:=**new** PictureABC(0,0,'BlueB.png');

RedB:=**new** PictureABC(0,0,'RedB.png');

WhiteB:=**new** PictureABC(0,0,'WhiteB.png');

**end**;

**procedure** DrawBounce;

**var** CellNum,kB,kR,kW,i,j,r:byte;

**begin**

CellNum:=sqr(Size **div** 80);

kB:=0;

kW:=0;

kR:=0;

i:=1;

j:=1;

**While** j\*i<=CellNum **do**

**begin**

r:=random(3);

**case** r **of**

0:**if** kB<CellNum/3 **then**

**begin**

Field.Items[j,i]:=blueb.Clone0;

inc(kB);

Arr[i,j]:=0;

**if** ((j=Size **div** 80) **and** (i<>sqrt(CellNum))) **then begin** inc(i); j:=1; **end**

**else** inc(j);

**end**;

1:**if** kW<CellNum/3 **then**

**begin**

Field.Items[j,i]:=Whiteb.clone0;

inc(kW);

Arr[i,j]:=1;

**if** ((j=Size **div** 80) **and** (i<>sqrt(CellNum))) **then begin** inc(i); j:=1; **end**

**else** inc(j);

**end**;

2:**if** kR<CellNum/3 **then**

**begin**

Field.Items[j,i]:=Redb.clone0;

inc(kR);

Arr[i,j]:=2;

**if** ((j=Size **div** 80) **and** (i<>sqrt(CellNum))) **then begin** inc(i); j:=1; **end**

**else** inc(j);

**end**;

**end**;

**end**;

WhiteB.Destroy;

BlueB.Destroy;

RedB.Destroy;

**end**;

**procedure** KeyDown(Key: integer);

**begin**

**case** Key **of**

VK\_Left: **if** px>80 **then begin** Point.MoveOn(-80,0); px:=px-80; **end**;

VK\_Right: **if** px<Size **then begin** Point.MoveOn(80,0); px:=px+80; **end**;

VK\_Up: **if** py>80 **then begin** Point.MoveOn(0,-80); py:=py-80; **end**;

VK\_Down: **if** py<Size **then begin** Point.MoveOn(0,80); py:=py+80; **end**;

VK\_Space: **begin**

**if** FlChoose=true **then**

**begin**

b:=Arr[y1,x1];

Arr[y1,x1]:=Arr[py **div** 80,px **div** 80];

Arr[py **div** 80,px **div** 80]:=b;

Field.SwapObjects(x1,y1,px **div** 80,py **div** 80);

sound.SoundLocation := 'click.wav';

sound.Play;

Check;

ChoosePoint.Visible:=false;

FLChoose:=false;

**end**

**else**

**begin**

x1:=px **div** 80;

y1:=py **div** 80;

ChoosePoint.Visible:=true;

ChoosePoint.MoveTo(px-80,py-80);

FlChoose:=true;

**end**;

**end**;

VK\_Back: halt;

**end**;

**end**;

**procedure** Select\_Object;

**begin**

FlChoose:=false;

Point:=**new** SquareABC(0,0,81);

Point.Filled:=false;

Point.BorderWidth:=5;

Point.Color:=clDarkGray;

ChoosePoint:=**new** SquareABC(0,0,81);

ChoosePoint.Filled:=false;

ChoosePoint.BorderWidth:=5;

ChoosePoint.Visible:=false;

ChoosePoint.BorderColor:=clLightGreen;

px:=80; py:=80;

OnKeyDown := KeyDown;

**end**;

**begin**

Menu;

**end**.

Результати роботи

 

 