**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: Буц Ярослав Ростиславович

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 3:*

«Стрибаючі кульки 2». Вздовж дошки розташовані лунки, і в кожній лунці лежить куля чорного або білого кольору. Одним ходом дозволяється міняти місцями дві будь-яких кулі. Добитися того, щоб спочатку йшли білі кулі, за ними – чорні. Якщо загальне число лунок дорівнює n, то для розв’язку задачі достатньо зробити не більш ніж [n/2] ходів.

Описання структури програми

**procedure** CreatingFieldToPlay – створю поле для

гри і та початкове положення курсору та кульок.

**procedure** keystrokes(key: integer)- оброблює

інформацію з клавіатури.(використовується одна й

та ж назва для процедур такого типу як в меню так

і в грі)

**procedure** game – запускає гру.

**procedure** CheckForOrder перевіряє поле для гри на

виконання завдання.

**procedure** DestroyingObjects – знищує об’єкти, що

використовувались у грі після гри.

**procedure** menucreate – виводить меню з фунціями

запуску гри, запуску допомоги та виходу з меню.

**procedure** help – виводить інформацію про суть гри

а також опис використання клавіш при керуванні.

**procedure** DestroyingMenu - знищує об’єкти, що

використовувались у виводі меню після переходу по

одному з пунктів.

**procedure** DestroyingHelp - знищує об’єкти, що

використовувались у виводі вікна help після

виходу з нього.

Main

Gameinit

Menu

game

menucreate

creatingfieldtoplay

keystrokes

keystrokes

help

DestroyingObjects

destroingmenu

CheckForOrder

destroinghelp

Код програми

**Main**

**Program** main\_dkr;

**uses** menu,gameinit;

**begin**

menucreate;

**end**.

**Unit Menu**

**unit** menu;

**interface**

**procedure** menucreate;

**implementation**

**uses** abcobjects,graphABC,gameinit;

**var**

menuitems1:**array** [1..3] **of** rectangleABC;{Array of punkts of menu}

menuitems2:**array** [1..3] **of** pictureABC;{Array of pictures for punkts om menu}

txt1:**array** [1..12] **of** TextABC;{array for text in help}

mframe:pictureABC;{frame as a cursor in menu}

k:integer; {variable to use not more then 1 help at one time}

**procedure** keystrokes(key: integer); **forward**;

**procedure** DestroyingHelp;{Destroing of objects after using help}

**var** i:byte;

**begin**

**for** i:=1 **to** 12 **do**

txt1[i].Destroy;

**end**;

**procedure** DestroyingMenu; {Destroing of objects after using menu}

**var** i:byte;

**begin**

**for** i:=1 **to** 3 **do**

**begin**

menuitems1[i].Destroy;

menuitems2[i].Destroy;

**end**;

mframe.Destroy;

**end**;

**procedure** help; {creating of help}

**begin**

txt1[1]:=textABC.Create(250,50,25,'Help');

txt1[2]:=textABC.Create(100,100,10,'In this game you need to permute balls with each other');

txt1[3]:=textABC.Create(100,120,10,'to put at first white balls and after - black.');

txt1[4]:=textABC.Create(100,140,10,'You have half of number of balls moves to do that');

txt1[5]:=textABC.Create(250,170,15,'Controls');

txt1[6]:=textABC.Create(100,200,10,'Press "escape" to return to the menu');

txt1[7]:=textABC.Create(100,220,10,'Press "enter" to choose item of the menu');

txt1[8]:=textABC.Create(100,240,10,'Press "space" to choose the ball for permuting');

txt1[9]:=textABC.Create(100,260,10,'Press "left" to move your cursor on 1 ball to left');

txt1[10]:=textABC.Create(100,280,10,'Press "right" to move your cursor on 1 ball to right');

txt1[11]:=textABC.Create(100,300,10,'Press "up" to move your cursor on 1 point of menu to up /1 ball to up');

txt1[12]:=textABC.Create(100,320,10,'Press "down" to move your cursor on 1 point of menu to down /1 ball to down');

**end**;

**procedure** menucreate;{Creating of menu}

**begin**

setwindowsize(600,500);

menuitems1[1]:=rectangleABC.Create(100,50,300,50,clwhite);

menuitems1[1].bordered:=false;

menuitems1[1].text:='Play';

menuitems1[1].FontStyle:=fsBoldItalic;

menuitems2[1]:=pictureABC.Create(410,30,'play.png');

menuitems2[1].Scale(0.7);

menuitems1[2]:=rectangleABC.Create(100,150,300,50,clwhite);

menuitems1[2].bordered:=false;

menuitems1[2].text:='Help';

menuitems1[2].FontStyle:=fsBoldItalic;

menuitems2[2]:=pictureABC.Create(410,130,'help.png');

menuitems2[2].Scale(0.7);

menuitems1[3]:=rectangleABC.Create(100,250,300,50,clwhite);

menuitems1[3].bordered:=false;

menuitems1[3].text:='Exit';

menuitems1[3].FontStyle:=fsBoldItalic;

menuitems2[3]:=pictureABC.Create(420,240,'exit.png');

menuitems2[3].Scale(0.5);

mframe:=pictureABC.Create(85,31,'mframe.png');

mframe.scaleX:=0.9;

mframe.scaley:=0.2;

ToBack(mframe);

k:=0;

onKeyDown:=keystrokes;

**end**;

**procedure** keystrokes(key: integer);{reading keyboard events}

**begin**

**case** (key) **of**

vk\_up:**if** mframe.Top > 31 **then begin**

mframe.MoveOn(0,-100);

**end**;

vk\_down: **if** mframe.top < 185 **then begin**

mframe.MoveOn(0,100);

**end**;

vk\_enter: **begin**

**if** mframe.Intersect(menuitems1[1]) **then begin**

DestroyingMenu;

game;

**end**;

**if** mframe.Intersect(menuitems1[2]) **then begin**

DestroyingMenu;

k:=k+1;

**if** k=1 **then** help;

**end**;

**if** mframe.Intersect(menuitems1[3]) **then** halt;

**end**;

vk\_escape: **begin**

DestroyingHelp;

menucreate;

**end**;

**end**;

**end**;

**begin**

**end**.

**Unit gameinit**

**unit** gameinit;{Unit initiate game start}

**interface**

**procedure** game;

**implementation**

**uses** menu,abcobjects,graphABC;

**const** m=4;n=4;{constants to create array}

**var**

Board: ObjectBoardABC;

hole: CircleABC;

pict: PictureABC;{Picture 'winner' or 'game over'}

ball: PictureABC;

fun: PictureABC; {Picture 'have fun'}

frame: PictureABC; {frame as a cursor in game}

rect: RectangleABC; {Rectangle for number of moves that lefts}

txt1,txt2: TextABC; {2 strings to write about number of moves that lefts}

balls: **array** [1..m,1..n] **of** pictureABC;

holes: **array** [1..m,1..n] **of** CircleABC;

Ar:**array**[1..m,1..n] **of** byte;{array that keep color of every ball where 0-black and 1-white }

x,y,x1,y1,x2,y2:word; Fl,Fl2:boolean;{x,y - current ball at what placed cursor

x1,y1 - 1st picked ball, x2,y2 - 2-nd picked ball; Fl-flag that controls where are

1-st and 2-nd balls, Fl2 - flag that controls is not game task finished yet}

k,q:byte;{k- current number of used moves, q - variable for replacement in array Ar}

**procedure** DestroyingObjects;{Dostroying objects of game after game}

**var** i,j:byte;

**begin**

frame.destroy;

**for** i:=1 **to** m **do**

**for** j:=1 **to** n **do**

**begin**

Balls[i,j].Destroy;

Holes[i,j].Destroy;

**end**;

board.Destroy;

rect.Destroy;

txt1.Destroy;

txt2.Destroy;

fun.Destroy;

**end**;

**procedure** CheckForOrder;{Cheking for is game task finished}

**var** i,j:byte;

**begin**

Fl2:=true;

**for** i:=1 **to** m **do**

**begin**

j:=1;

**if** i>1 **then**

**if** Ar[i,j]>Ar[i-1,n] **then** Fl2:=false;

**for** j:=2 **to** n **do**

**if** Ar[i,j]>Ar[i,j-1] **then** Fl2:=false;

**end**;

**end**;

**procedure** CreatingFieldToPlay;{creating field for game}

**var** i,j:byte;

**begin**

randomize;

fun:=pictureABC.Create(20+n\*100,m\*40,'HaveFun.png');

fun.scaleX:=0.6;

fun.ScaleY:=m\*3/20;

setwindowsize(250+n\*100,m\*100);

Board:= ObjectBoardABC.create(0,0,n,m,100,100,clbrown);

ToBack(Board);

x:=0;y:=0;k:=0;

**for** i:=1 **to** m **do**

**for** j:=1 **to** n **do**

**begin**

hole:= **new** circleABC(50+x,50+y,40,clblack);

hole.Filled:=false;

hole.BorderWidth:=10;

x:=x+100;

**if** j = n **then**

**begin**

x:=0;

y:=y+100;

**end**;

holes[i,j]:=hole;

**end**;

x:=0;y:=0;

**for** i:=1 **to** m **do**

**for** j:=1 **to** n **do**

**begin**

Ar[i,j]:=random(2);

**if** Ar[i,j]=0

**then**

**begin**

ball:=pictureabc.Create(18+x,19+y,'black\_ball.png');

ball.Scale(1.4);

balls[i,j]:=ball;

x:=x+100;

**if** j=n **then**

**begin**

x:=0;

y:=y+100;

**end**;

**end**

**else**

**begin**

ball:=pictureabc.Create(18+x,19+y,'white\_ball.png');

ball.Scale(1.4);

balls[i,j]:=ball;

x:=x+100;

**if** j=n **then**

**begin**

x:=0;

y:=y+100;

**end**;

**end**;

**end**;

frame:=pictureabc.Create(0,0,'ramk216.png');

frame.scale(0.035);

txt1:=TextABC.Create(20+n\*100,0,16,'the remaining number');

txt1.TransparentBackground:=false;

txt1.BackgroundColor:=clred;

txt2:=TextABC.Create(20+n\*100,20,16,'moves:');

txt2.TransparentBackground:=false;

txt2.BackgroundColor:=clred;

rect:=RectangleABC.Create(20+n\*100,50,round(100\*n/4),round(100\*m/4),Clwhite);

rect.number:= round(m\*n/2-k);

rect.color:=clyellow;

**end**;

**procedure** keystrokes(key: integer);{reading keyboard events}

**begin**

**case** (key) **of**

vk\_left: **if** frame.Left>0 **then begin**

frame.MoveOn(-100,0);

x:=x-100;

**end**;

vk\_right: **if** frame.Left<(n-1)\*100 **then begin**

frame.MoveOn(100,0);

x:=x+100;

**end**;

vk\_up:**if** frame.Top > 0 **then begin**

frame.MoveOn(0,-100);

y:=y-100;

**end**;

vk\_down: **if** frame.top < (m-1)\*100 **then begin**

frame.MoveOn(0,100);

y:=y+100;

**end**;

vk\_space: **if** Fl=false

**then begin**

Fl:=True;

x1:=x;

y1:=y;

**end**

**else begin**

Fl:=false;

x2:=x;

y2:=y;

swappositions (balls[round((y1+81)/100),round((x1+82)/100)],

balls[round((y2+81)/100),round((x2+82)/100)]);

ball:=balls[round((y1+81)/100),round((x1+82)/100)];

balls[round((y1+81)/100),round((x1+82)/100)]:=balls[round((y2+81)/100),round((x2+82)/100)];

balls[round((y2+81)/100),round((x2+82)/100)]:=ball;

q:=Ar[round((y1+81)/100),round((x1+82)/100)];

Ar[round((y1+81)/100),round((x1+82)/100)]:=Ar[round((y2+81)/100),round((x2+82)/100)];

Ar[round((y2+81)/100),round((x2+82)/100)]:=q;

k:=k+1;

rect.number:= round(m\*n/2-(k+0.25));

CheckForOrder;

**if** Fl2=true **then begin**

DestroyingObjects;

setwindowsize(400,400);

pict:=pictureabc.Create(0,0,'winner4.png');

sleep(5000);

pict.Destroy;

Menucreate;

**end**;

**if** k>= m\*n/2-0.5 **then begin**

DestroyingObjects;

setwindowsize(400,400);

pict:=pictureabc.Create(0,0,'gameover.png');

pict.Scale(4/3);

sleep(5000);

pict.Destroy;

Menucreate;

**end**;

**end**;

vk\_escape: **begin**

DestroyingObjects;

Menucreate;

**end**;

**end**;

**end**;

**procedure** game;{iniciating game}

**begin**

CreatingFieldToPlay;

x:=18;y:=19;Fl:=false;

onKeyDown:=keystrokes;

**end**;

**begin**

**end**.

Результати роботи







