МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

Высшего профессионального образования

**«Вятский государственный университет»**

**(ФГБОУ ВПО «ВятГУ»)**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Создание собственного игрового приложения с применением примитивной компьютерной графики на базе Lazarus

Отчет по лабораторным работам №9, 10 дисциплины

«Программирование»

Выполнил студент группы ИВТб-11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Седов М.Д./

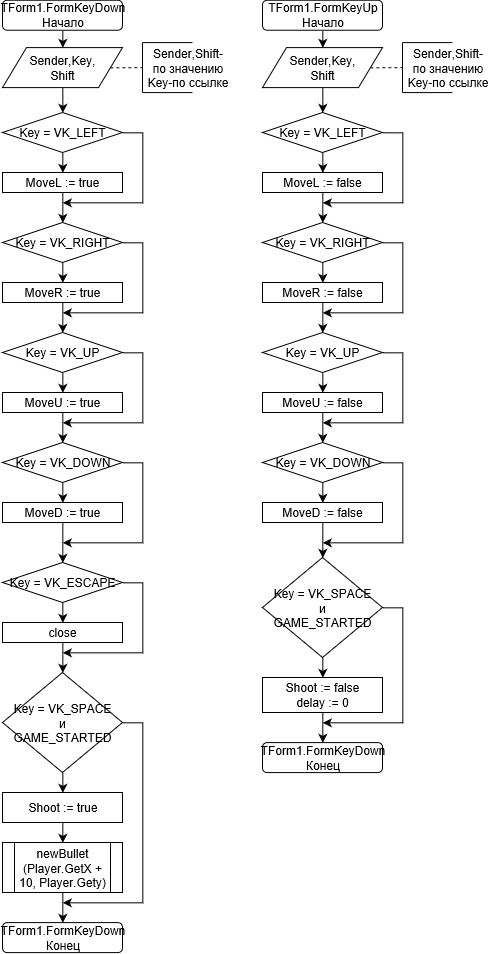
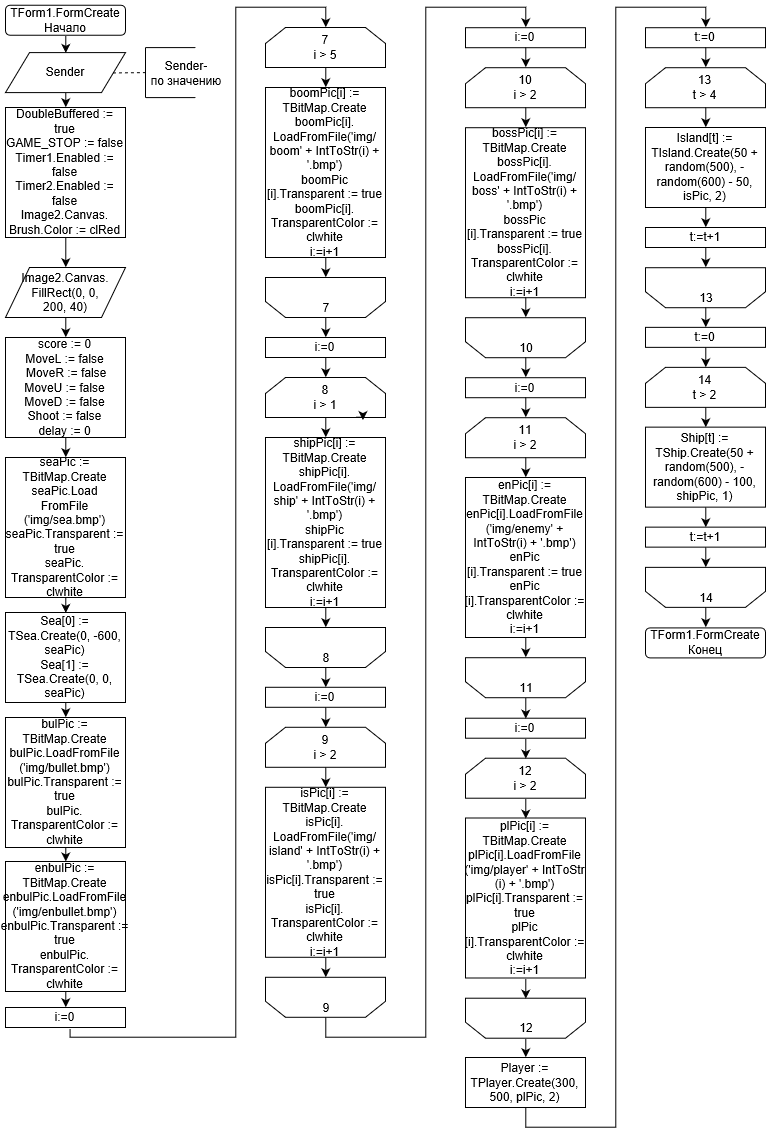
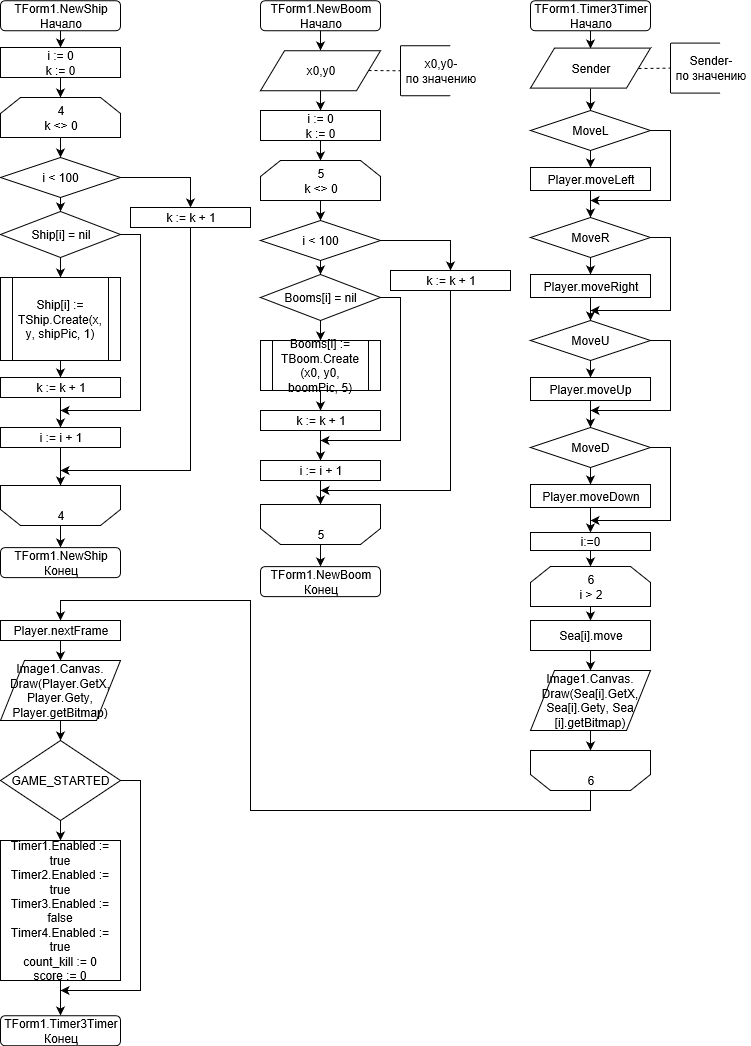
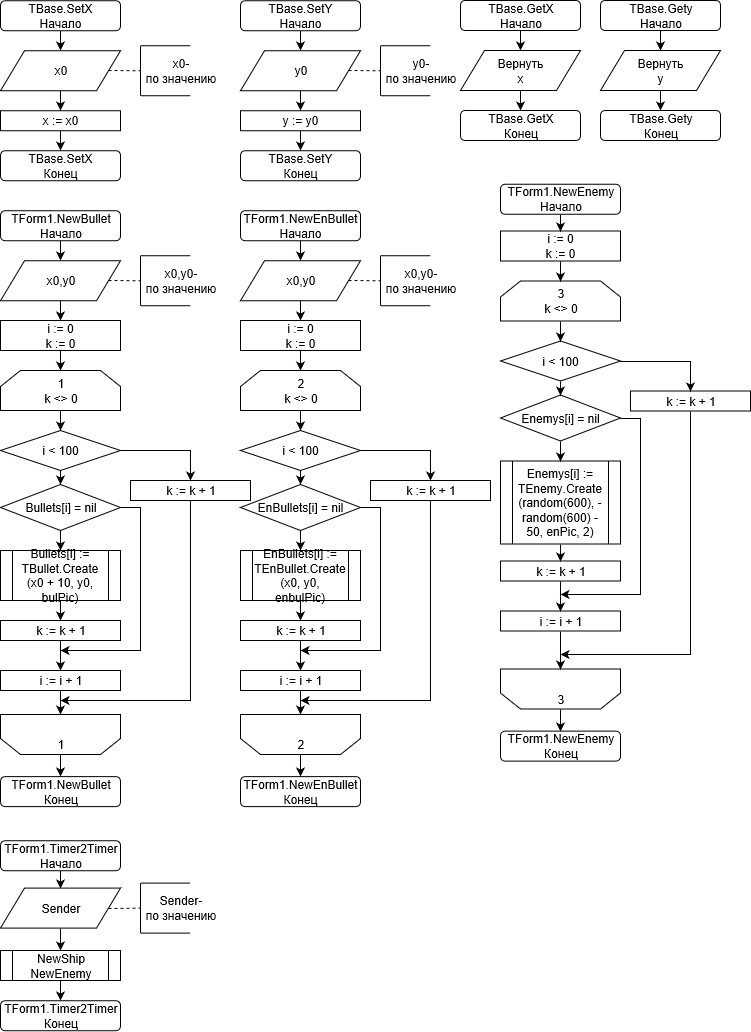
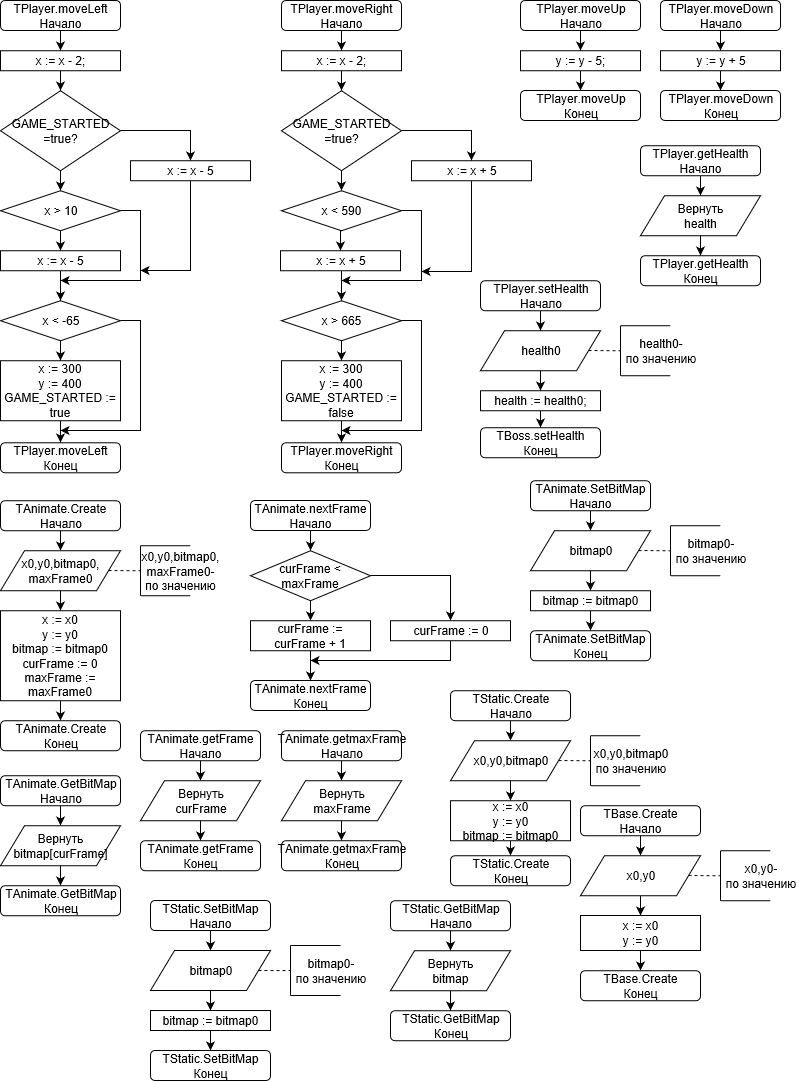
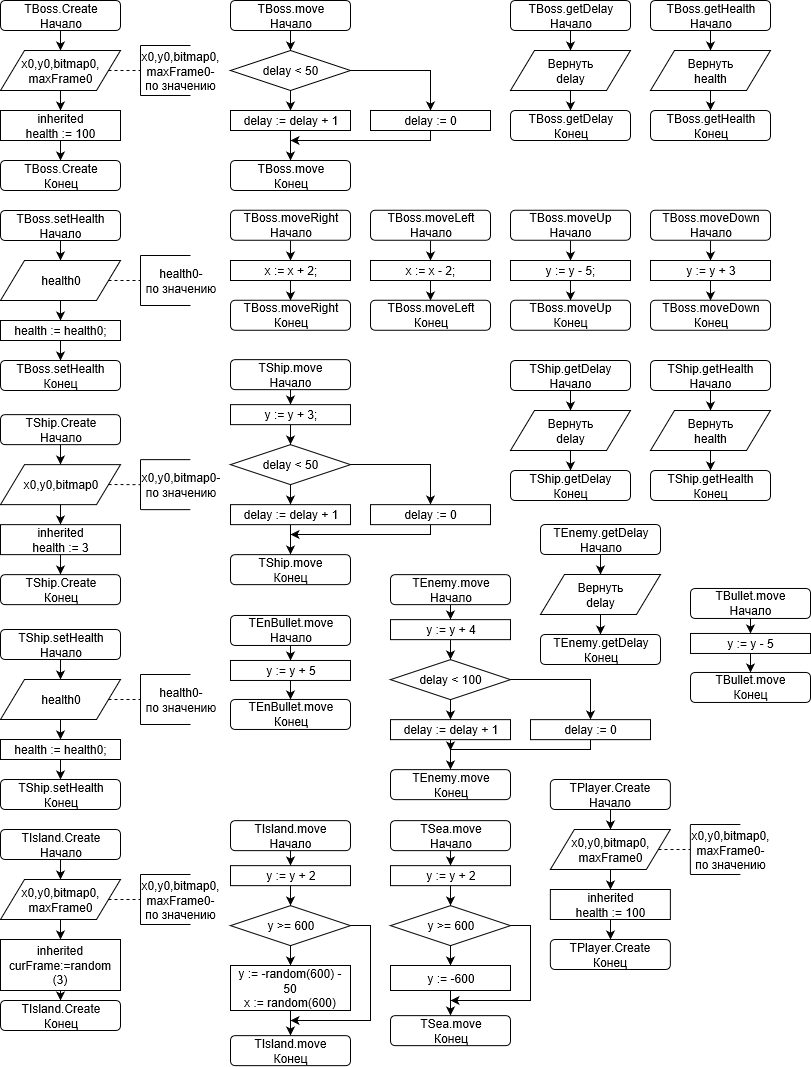
Проверил преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Чистяков Г.А./

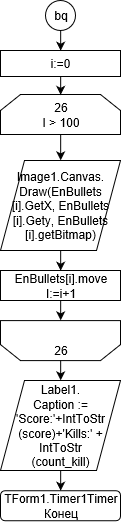
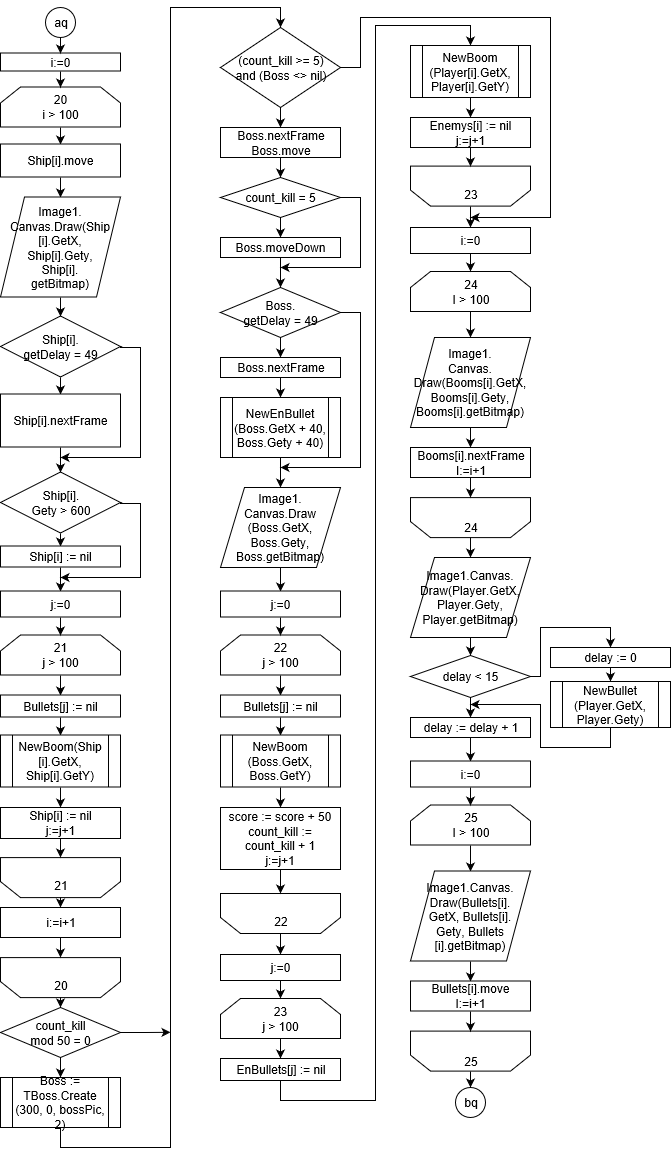
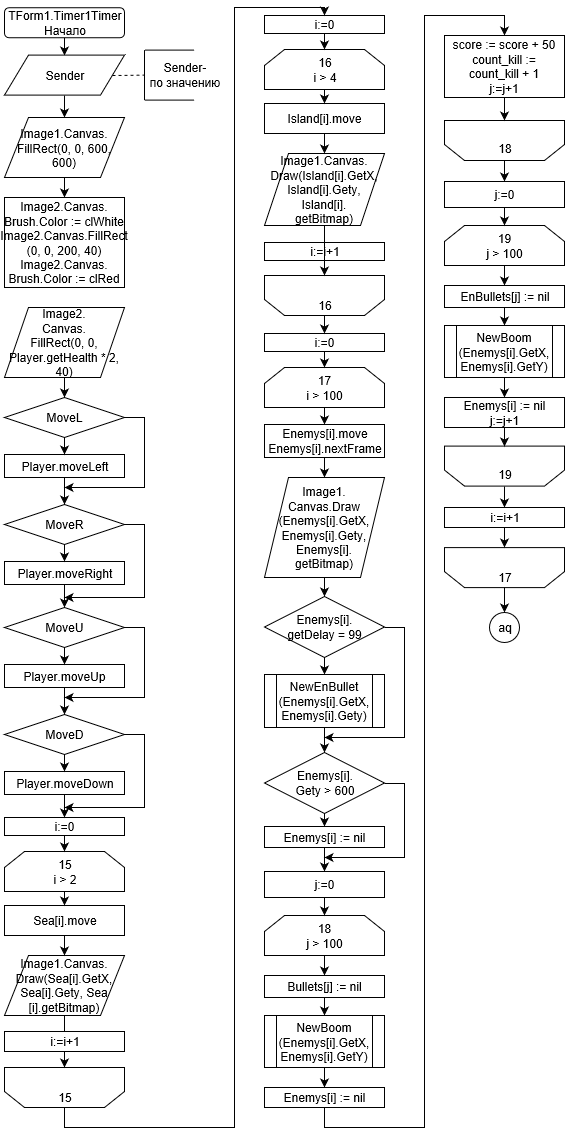
Киров 2018

**Цель работы:** Ознакомление со средой визуального программирования Lazarus и разработка в ней игрового приложения с применением примитивной компьютерной графики, получить навыки работы с базами данных

**Задание:**

1. **Создать собственное игровое приложение с применением примитивной компьютерной графики на базе Lazarus.**
2. **Реализовать рейтинг игроков: список пользователей и набранные очки по окончанию игры**

**Схема алгоритмов:**

****

**Исходный текст программы:**

**unit objs;**

**{$mode objfpc}{$H+}**

**interface**

**uses**

**Classes, SysUtils, Graphics;**

**type**

**{ TBase }**

**TBase = class**

**private**

**x : integer;**

**y : integer;**

**public**

**constructor Create(x0, y0 : integer);**

**procedure SetX(x0 : integer);**

**procedure SetY(y0 : integer);**

**function GetX : integer;**

**function Gety : integer;**

**end;**

**{ TStatic }**

**TStatic = class(TBase)**

**private**

**bitmap : TBitMap;**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : TBitMap);**

**procedure SetBitMap(bitmap0 : TBitMap);**

**function GetBitMap : TBitMap;**

**end;**

**{ TAnimate }**

**TAnimate = class(TBase)**

**private**

**bitmap : array [0..100] of TBitMap;**

**curFrame : integer;**

**maxFrame : integer;**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : array of TBitMap; maxFrame0 : integer);**

**procedure nextFrame;**

**procedure SetBitMap(bitmap0 : array of TBitMap);**

**function GetBitMap : TBitMap;**

**function getFrame : integer;**

**function getmaxFrame : integer;**

**end;**

**{ TPlayer }**

**TPlayer = class(TAnimate)**

**private**

**health : integer;**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : array of TBitMap; maxFrame0 : integer);**

**procedure moveLeft;**

**procedure moveRight;**

**procedure moveUp;**

**procedure moveDown;**

**function getHealth : integer;**

**procedure setHealth(health0 : integer);**

**end;**

**{ TSea }**

**TSea = class(TStatic)**

**public**

**procedure move;**

**end;**

**{ TBullet }**

**TBullet = class(TStatic)**

**public**

**procedure move;**

**end;**

**{ TIsland }**

**TIsland = class(TAnimate)**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : array of TBitMap; maxFrame0 : integer);**

**procedure move;**

**end;**

**{ TShip }**

**TShip = class(TAnimate)**

**private**

**delay : integer;**

**health : integer;**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : array of TBitMap; maxFrame0 : integer);**

**procedure move;**

**function getDelay : integer;**

**procedure setHealth(health0 : integer);**

**function getHealth : integer;**

**end;**

**{ TEnemy }**

**TEnemy = class(TAnimate)**

**private**

**delay : integer;**

**public**

**procedure move;**

**function getDelay : integer;**

**end;**

**{ TEnBullet }**

**TEnBullet = class(TStatic)**

**public**

**procedure move;**

**end;**

**TBoom = class(TAnimate)**

**end;**

**{ TBoss }**

**TBoss = class(TAnimate)**

**private**

**delay : integer;**

**health : integer;**

**public**

**constructor Create(x0, y0 : integer; bitmap0 : array of TBitMap; maxFrame0 : integer);**

**procedure move;**

**function getDelay : integer;**

**function getHealth : integer;**

**procedure setHealth(health0 : integer);**

**procedure moveRight;**

**procedure moveLeft;**

**procedure moveUp;**

**procedure moveDown;**

**end;**

**var GAME\_STARTED : boolean = false;**

**implementation**

**{ TBoss }**

**constructor TBoss.Create(x0, y0: integer; bitmap0: array of TBitMap;**

**maxFrame0: integer);**

**begin**

**inherited;**

**health := 100;**

**end;**

**procedure TBoss.move;**

**begin**

**if (delay < 50) then**

**delay := delay + 1**

**else**

**delay := 0;**

**end;**

**function TBoss.getDelay: integer;**

**begin**

**result := delay;**

**end;**

**function TBoss.getHealth: integer;**

**begin**

**result := health;**

**end;**

**procedure TBoss.setHealth(health0: integer);**

**begin**

**health := health0;**

**end;**

**procedure TBoss.moveRight;**

**begin**

**x := x + 2;**

**end;**

**procedure TBoss.moveLeft;**

**begin**

**x := x - 2;**

**end;**

**procedure TBoss.moveUp;**

**begin**

**y := y - 5;**

**end;**

**procedure TBoss.moveDown;**

**begin**

**y := y + 3;**

**end;**

**{ TShip }**

**constructor TShip.Create(x0, y0: integer; bitmap0: array of TBitMap;**

**maxFrame0: integer);**

**begin**

**inherited;**

**health := 3;**

**end;**

**procedure TShip.move;**

**begin**

**y := y + 3;**

**if (delay < 50) then**

**delay := delay + 1**

**else**

**delay := 0;**

**end;**

**function TShip.getDelay: integer;**

**begin**

**result := delay;**

**end;**

**procedure TShip.setHealth(health0: integer);**

**begin**

**health := health0;**

**end;**

**function TShip.getHealth: integer;**

**begin**

**result := health;**

**end;**

**{ TEnBullet }**

**procedure TEnBullet.move;**

**begin**

**y := y + 5;**

**end;**

**{ TEnemy }**

**procedure TEnemy.move;**

**begin**

**y := y + 4;**

**if (delay < 100) then**

**delay := delay + 1**

**else**

**delay := 0;**

**end;**

**function TEnemy.getDelay: integer;**

**begin**

**result := delay;**

**end;**

**{ TBullet }**

**procedure TBullet.move;**

**begin**

**y := y - 5;**

**end;**

**{ TIsland }**

**constructor TIsland.Create(x0, y0: integer; bitmap0: array of TBitMap;**

**maxFrame0: integer);**

**begin**

**inherited;**

**curFrame := random(3);**

**end;**

**procedure TIsland.move;**

**begin**

**y := y + 2;**

**if (y >= 600) then**

**begin**

**randomize;**

**y := -random(600) - 50;**

**x := random(600);**

**end;**

**end;**

**{ TSea }**

**procedure TSea.move;**

**begin**

**y := y + 2;**

**if (y >= 600) then y := -600;**

**end;**

**{ TPlayer }**

**constructor TPlayer.Create(x0, y0: integer; bitmap0: array of TBitMap;**

**maxFrame0: integer);**

**begin**

**inherited;**

**health := 100;**

**end;**

**procedure TPlayer.moveLeft;**

**begin**

**if (GAME\_STARTED) then begin**

**if (x > 10) then x := x - 5**

**end else x := x - 5;**

**if (x < -65) then**

**begin**

**x := 300;**

**y := 400;**

**GAME\_STARTED := true;**

**end;**

**end;**

**procedure TPlayer.moveRight;**

**begin**

**if (GAME\_STARTED) then begin**

**if (x < 590) then x := x + 5**

**end else x := x + 5;**

**if (x > 665) then**

**begin**

**GAME\_STARTED := false;**

**x := 300;**

**y := 400;**

**end;**

**end;**

**procedure TPlayer.moveUp;**

**begin**

**if (y > 100) then y := y - 5;**

**end;**

**procedure TPlayer.moveDown;**

**begin**

**if (y < 530) then y := y + 5;**

**end;**

**function TPlayer.getHealth: integer;**

**begin**

**result := health;**

**end;**

**procedure TPlayer.setHealth(health0: integer);**

**begin**

**health := health0;**

**end;**

**{ TAnimate }**

**constructor TAnimate.Create(x0, y0: integer; bitmap0: array of TBitMap;**

**maxFrame0: integer);**

**begin**

**x := x0;**

**y := y0;**

**bitmap := bitmap0;**

**curFrame := 0;**

**maxFrame := maxFrame0;**

**end;**

**procedure TAnimate.nextFrame;**

**begin**

**if (curFrame < maxFrame) then**

**curFrame := curFrame + 1**

**else**

**curFrame := 0;**

**end;**

**procedure TAnimate.SetBitMap(bitmap0: array of TBitMap);**

**begin**

**bitmap := bitmap0;**

**end;**

**function TAnimate.GetBitMap: TBitMap;**

**begin**

**result := bitmap[curFrame];**

**end;**

**function TAnimate.getFrame: integer;**

**begin**

**result := curFrame;**

**end;**

**function TAnimate.getmaxFrame: integer;**

**begin**

**result := maxFrame;**

**end;**

**{ TStatic }**

**constructor TStatic.Create(x0, y0: integer; bitmap0: TBitMap);**

**begin**

**x := x0;**

**y := y0;**

**bitmap := bitmap0;**

**end;**

**procedure TStatic.SetBitMap(bitmap0: TBitMap);**

**begin**

**bitmap := bitmap0;**

**end;**

**function TStatic.GetBitMap: TBitMap;**

**begin**

**result := bitmap;**

**end;**

**{ TBase }**

**constructor TBase.Create(x0, y0: integer);**

**begin**

**x := x0;**

**y := y0;**

**end;**

**procedure TBase.SetX(x0: integer);**

**begin**

**x := x0;**

**end;**

**procedure TBase.SetY(y0: integer);**

**begin**

**y := y0;**

**end;**

**function TBase.GetX: integer;**

**begin**

**result := x;**

**end;**

**function TBase.Gety: integer;**

**begin**

**result := y;**

**end;**

**end.**

**unit main;**

**{$mode objfpc}{$H+}**

**interface**

**uses**

**Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, LCLType,**

**ExtCtrls, StdCtrls, objs;**

**type**

**{ TForm1 }**

**TForm1 = class(TForm)**

**Image1: TImage;**

**Image2: TImage;**

**Label1: TLabel;**

**Timer1: TTimer;**

**Timer2: TTimer;**

**Timer3: TTimer;**

**Timer4: TTimer;**

**procedure FormCreate(Sender: TObject);**

**procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);**

**procedure FormKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);**

**procedure Timer1Timer(Sender: TObject);**

**procedure NewBullet(x0, y0 : integer);**

**procedure NewEnBullet(x0, y0 : integer);**

**procedure NewEnemy;**

**procedure NewShip;**

**procedure NewBoom(x0, y0 : integer);**

**procedure Timer2Timer(Sender: TObject);**

**procedure Timer3Timer(Sender: TObject);**

**private**

**public**

**end;**

**var**

**Form1: TForm1;**

**//Графика всякая**

**seaPic : TBitMap;**

**isPic : array [0..2] of TBitMap;**

**plPic : array [0..2] of TBitMap;**

**shipPic : array [0..1] of TBitMap;**

**bulPic : TBitMap;**

**enbulPic : TBitMap;**

**bossPic : array [0..2] of TBitMap;**

**enPic : array [0..2] of TBitMap;**

**boomPic : array [0..5] of TBitMap;**

**//Сами объекты**

**Player : TPlayer;**

**Sea : array [0..1] of TSea;**

**Island : array [0..4] of TIsland;**

**Bullets : array [0..100] of TBullet;**

**EnBullets : array [0..100] of TEnBullet;**

**Enemys : array [0..100] of TEnemy;**

**Booms : array [0..100] of TBoom;**

**Ship : array [0..100] of TShip;**

**Boss : TBoss;**

**MoveL, MoveR, MoveU, MoveD : boolean;**

**Shoot : boolean;**

**GAME\_STOP : boolean;**

**delay : integer;**

**score : integer;**

**count\_kill : integer;**

**implementation**

**{$R \*.lfm}**

**{ TForm1 }**

**procedure TForm1.FormCreate(Sender: TObject);**

**var i, t : integer;**

**begin**

**DoubleBuffered := true;**

**GAME\_STOP := false;**

**Timer1.Enabled := false;**

**Timer2.Enabled := false;**

**Image2.Canvas.Brush.Color := clRed;**

**Image2.Canvas.FillRect(0, 0, 200, 40);**

**score := 0;**

**MoveL := false;**

**MoveR := false;**

**MoveU := false;**

**MoveD := false;**

**Shoot := false;**

**delay := 0;**

**//sea**

**seaPic := TBitMap.Create;**

**seaPic.LoadFromFile('img/sea.bmp');**

**seaPic.Transparent := true;**

**seaPic.TransparentColor := clwhite;**

**Sea[0] := TSea.Create(0, -600, seaPic);**

**Sea[1] := TSea.Create(0, 0, seaPic);**

**//bullet**

**bulPic := TBitMap.Create;**

**bulPic.LoadFromFile('img/bullet.bmp');**

**bulPic.Transparent := true;**

**bulPic.TransparentColor := clwhite;**

**//Враг стреляет**

**enbulPic := TBitMap.Create;**

**enbulPic.LoadFromFile('img/enbullet.bmp');**

**enbulPic.Transparent := true;**

**enbulPic.TransparentColor := clwhite;**

**//Взрыыыыыыыв**

**for i := 0 to 5 do**

**begin**

**boomPic[i] := TBitMap.Create;**

**boomPic[i].LoadFromFile('img/boom' + IntToStr(i) + '.bmp');**

**boomPic[i].Transparent := true;**

**boomPic[i].TransparentColor := clwhite;**

**end;**

**//Вражеские кораблики**

**for i := 0 to 1 do**

**begin**

**shipPic[i] := TBitMap.Create;**

**shipPic[i].LoadFromFile('img/ship' + IntToStr(i) + '.bmp');**

**shipPic[i].Transparent := true;**

**shipPic[i].TransparentColor := clwhite;**

**end;**

**//islands**

**for i := 0 to 2 do**

**begin**

**isPic[i] := TBitMap.Create;**

**isPic[i].LoadFromFile('img/island' + IntToStr(i) + '.bmp');**

**isPic[i].Transparent := true;**

**isPic[i].TransparentColor := clwhite;**

**end;**

**//Боссик**

**for i := 0 to 2 do**

**begin**

**bossPic[i] := TBitMap.Create;**

**bossPic[i].LoadFromFile('img/boss' + IntToStr(i) + '.bmp');**

**bossPic[i].Transparent := true;**

**bossPic[i].TransparentColor := clwhite;**

**end;**

**//вражина**

**for i := 0 to 2 do**

**begin**

**enPic[i] := TBitMap.Create;**

**enPic[i].LoadFromFile('img/enemy' + IntToStr(i) + '.bmp');**

**enPic[i].Transparent := true;**

**enPic[i].TransparentColor := clwhite;**

**end;**

**//player**

**for i := 0 to 2 do**

**begin**

**plPic[i] := TBitMap.Create;**

**plPic[i].LoadFromFile('img/player' + IntToStr(i) + '.bmp');**

**plPic[i].Transparent := true;**

**plPic[i].TransparentColor := clwhite;**

**end;**

**Player := TPlayer.Create(300, 500, plPic, 2);**

**for t := 0 to 4 do Island[t] := TIsland.Create(50 + random(500), -random(600) - 50, isPic, 2);**

**for t := 0 to 2 do Ship[t] := TShip.Create(50 + random(500), -random(600) - 100, shipPic, 1);**

**end;**

**procedure TForm1.FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState**

**);**

**begin**

**if (Key = VK\_LEFT) then MoveL := true;**

**if (Key = VK\_RIGHT) then MoveR := true;**

**if (Key = VK\_UP) then MoveU := true;**

**if (Key = VK\_DOWN) then MoveD := true;**

**if (Key = VK\_ESCAPE) then close;**

**if (Key = VK\_SPACE) and (GAME\_STARTED) then begin Shoot := true; newBullet(Player.GetX + 10, Player.Gety); end;**

**end;**

**procedure TForm1.FormKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);**

**begin**

**if (Key = VK\_LEFT) then MoveL := false;**

**if (Key = VK\_RIGHT) then MoveR := false;**

**if (Key = VK\_UP) then MoveU := false;**

**if (Key = VK\_DOWN) then MoveD := false;**

**if (Key = VK\_SPACE) and (GAME\_STARTED) then begin Shoot := false; delay := 0; end;**

**end;**

**procedure TForm1.Timer1Timer(Sender: TObject);**

**var i, j, t : integer;**

**begin**

**Image1.Canvas.FillRect(0, 0, 600, 600);**

**Image2.Canvas.Brush.Color := clWhite;**

**Image2.Canvas.FillRect(0, 0, 200, 40);**

**Image2.Canvas.Brush.Color := clRed;**

**Image2.Canvas.FillRect(0, 0, Player.getHealth \* 2, 40);**

**if (MoveL) then Player.moveLeft;**

**if (MoveR) then Player.moveRight;**

**if (MoveU) then Player.moveUp;**

**if (MoveD) then Player.moveDown;**

**//Вывод моря**

**for i := 0 to 1 do**

**begin**

**Sea[i].move;**

**Image1.Canvas.Draw(Sea[i].GetX, Sea[i].Gety, Sea[i].getBitmap);**

**end;**

**//Вывод островов**

**for i := 0 to 4 do**

**begin**

**Island[i].move;**

**Image1.Canvas.Draw(Island[i].GetX, Island[i].Gety, Island[i].getBitmap);**

**end;**

**//Вывод врагов**

**for i := 0 to 100 do**

**if (Enemys[i] <> nil) then**

**begin**

**Enemys[i].move;**

**Enemys[i].nextFrame;**

**Image1.Canvas.Draw(Enemys[i].GetX, Enemys[i].Gety, Enemys[i].getBitmap);**

**if (Enemys[i].getDelay = 99) then**

**NewEnBullet(Enemys[i].GetX, Enemys[i].Gety);**

**if (Enemys[i].Gety > 600) then Enemys[i] := nil;**

**for j := 0 to 100 do**

**if (Bullets[j] <> nil) and (Enemys[i] <> nil) then**

**if (Bullets[j].getX + 32 > Enemys[i].getX) and**

**(Bullets[j].getX < Enemys[i].getX + 32) and**

**(Bullets[j].getY < Enemys[i].getY + 32) then**

**begin**

**Bullets[j] := nil;**

**NewBoom(Enemys[i].GetX, Enemys[i].GetY);**

**Enemys[i] := nil;**

**score := score + 50;**

**count\_kill := count\_kill + 1;**

**end;**

**for j := 0 to 100 do**

**if (EnBullets[j] <> nil) then**

**if (EnBullets[j].getX + 32 > Player.getX) and**

**(EnBullets[j].getX < Player.getX + 32) and**

**(EnBullets[j].getY > Player.getY) then**

**begin**

**EnBullets[j] := nil;**

**NewBoom(Player.GetX + 10, Player.GetY);**

**if (Player.getHealth > 0) then begin**

**Player.setHealth(Player.getHealth - 5)**

**end**

**else**

**begin**

**for t := 0 to 10 do NewBoom(50 + random(500), 50 + random(500));**

**GAME\_STARTED := false;**

**end;**

**end;**

**end;**

**//Вывод корабликов**

**for i := 0 to 100 do**

**if (Ship[i] <> nil) then**

**begin**

**Ship[i].move;**

**Image1.Canvas.Draw(Ship[i].GetX, Ship[i].Gety, Ship[i].getBitmap);**

**if (Ship[i].getDelay = 49) then Ship[i].nextFrame;**

**if (Ship[i].Gety > 600) then Ship[i] := nil;**

**for j := 0 to 100 do**

**if (Bullets[j] <> nil) and (Ship[i] <> nil) then**

**if (Bullets[j].getX + 32 > Ship[i].getX) and**

**(Bullets[j].getX < Ship[i].getX + 32) and**

**(Bullets[j].getY < Ship[i].getY + 100) then**

**if (Ship[i].getHealth > 0) then**

**begin**

**Ship[i].setHealth(Ship[i].getHealth - 1);**

**Bullets[j] := nil;**

**NewBoom(Ship[i].GetX, Ship[i].GetY + 100);**

**end**

**else**

**begin**

**Ship[i] := nil;**

**score := score + 100;**

**count\_kill := count\_kill + 1;**

**end;**

**end;**

**if (count\_kill mod 50 = 0) then Boss := TBoss.Create(300, 0, bossPic, 2);**

**//Вывод Босса**

**if (count\_kill >= 5) and (Boss <> nil) then begin**

**Boss.nextFrame;**

**Boss.move;**

**if (count\_kill = 5) then Boss.moveDown;**

**if (Boss.Gety < -10) then Boss.moveDown;**

**if (Boss.Gety > 50) then Boss.moveUp;**

**if (Boss.getDelay = 49) then**

**begin**

**Boss.nextFrame;**

**NewEnBullet(Boss.GetX + 40, Boss.Gety + 40);**

**end;**

**if (Boss.GetX > Player.GetX) then Boss.moveLeft else Boss.moveRight;**

**Image1.Canvas.Draw(Boss.GetX, Boss.Gety, Boss.getBitmap);**

**for j := 0 to 100 do**

**if (Bullets[j] <> nil) then**

**if (Bullets[j].getX + 80 > Boss.getX) and**

**(Bullets[j].getX < Boss.getX + 80) and**

**(Bullets[j].getY < Boss.getY + 80) then**

**if (Boss.getHealth > 0) then**

**begin**

**Boss.setHealth(Boss.getHealth - 5);**

**Bullets[j] := nil;**

**NewBoom(Boss.GetX + 10, Boss.GetY + 80);**

**end**

**else**

**begin**

**Boss := nil;**

**score := score + 500;**

**count\_kill := count\_kill + 1;**

**end;**

**for j := 0 to 100 do**

**if (EnBullets[j] <> nil) then**

**if (EnBullets[j].getX + 32 > Player.getX) and**

**(EnBullets[j].getX < Player.getX + 32) and**

**(EnBullets[j].getY > Player.getY) then**

**begin**

**EnBullets[j] := nil;**

**NewBoom(Player.GetX + 10, Player.GetY);**

**if (Player.getHealth > 0) then begin**

**Player.setHealth(Player.getHealth - 15)**

**end**

**else**

**begin**

**for t := 0 to 10 do NewBoom(50 + random(500), 50 + random(500));**

**GAME\_STARTED := false;**

**end;**

**end;**

**end;**

**//Бум**

**for i := 0 to 100 do**

**if (Booms[i] <> nil) then**

**if (Booms[i].getFrame < Booms[i].getmaxFrame) then**

**begin**

**Image1.Canvas.Draw(Booms[i].GetX, Booms[i].Gety, Booms[i].getBitmap);**

**Booms[i].nextFrame;**

**end**

**else**

**Booms[i] := nil;**

**//Player.nextFrame;**

**Image1.Canvas.Draw(Player.GetX, Player.Gety, Player.getBitmap);**

**if (Shoot) then**

**if (delay < 15) then**

**delay := delay + 1**

**else**

**begin**

**delay := 0;**

**NewBullet(Player.GetX, Player.Gety);**

**end;**

**//Вывод пуль**

**for i := 0 to 100 do**

**if (Bullets[i] <> nil) then**

**begin**

**Bullets[i].move;**

**Image1.Canvas.Draw(Bullets[i].GetX, Bullets[i].Gety, Bullets[i].getBitmap);**

**if (Bullets[i].Gety < -32) then Bullets[i] := nil;**

**end;**

**//Двигаем вражескую пулю**

**for i := 0 to 100 do**

**if (EnBullets[i] <> nil) then**

**if (EnBullets[i].Gety < 600) then**

**begin**

**EnBullets[i].move;**

**Image1.Canvas.Draw(EnBullets[i].GetX, EnBullets[i].Gety, EnBullets[i].getBitmap);**

**end**

**else EnBullets[i] := nil;**

**//Player.nextFrame;**

**//Image1.Canvas.Draw(Player.GetX, Player.Gety, Player.getBitmap);**

**Label1.Caption := 'Score: ' + IntToStr(score) + ' Kills: ' + IntToStr(count\_kill);**

**if (not GAME\_STARTED) then**

**begin**

**Timer1.Enabled := false;**

**Timer2.Enabled := false;**

**Timer4.Enabled := false;**

**Timer3.Enabled := true;**

**for i := 0 to 100 do**

**begin**

**Bullets[i] := nil;**

**EnBullets[i] := nil;**

**Enemys[i] := nil;**

**Booms[i] := nil;**

**Ship[i] := nil;**

**end;**

**score := 0;**

**count\_kill := 0;**

**Label1.Caption := 'Score: ' + IntToStr(score) + ' Kills: ' + IntToStr(count\_kill);**

**Player.setHealth(100);**

**end;**

**end;**

**//Создание пули**

**procedure TForm1.NewBullet(x0, y0: integer);**

**var i, k : integer;**

**begin**

**i := 0;**

**k := 0;**

**while (k = 0) do**

**if (i < 100) then**

**begin**

**if (Bullets[i] = nil) then**

**begin**

**Bullets[i] := TBullet.Create(x0 + 10, y0, bulPic);**

**k := k + 1;**

**end;**

**i := i + 1;**

**end**

**else**

**k := k + 1;**

**end;**

**//Вражеские пули**

**procedure TForm1.NewEnBullet(x0, y0: integer);**

**var i, k : integer;**

**begin**

**i := 0;**

**k := 0;**

**while (k = 0) do**

**if (i < 100) then**

**begin**

**if (EnBullets[i] = nil) then**

**begin**

**EnBullets[i] := TEnBullet.Create(x0, y0, enbulPic);**

**k := k + 1;**

**end;**

**i := i + 1;**

**end**

**else**

**k := k + 1;**

**end;**

**//Создание врагов**

**procedure TForm1.NewEnemy;**

**var i, k : integer;**

**begin**

**i := 0;**

**k := 0;**

**while (k = 0) do**

**if (i < 100) then**

**begin**

**if (Enemys[i] = nil) then**

**begin**

**Enemys[i] := TEnemy.Create(random(600), -random(600) - 50, enPic, 2);**

**k := k + 1;**

**end;**

**i := i + 1;**

**end**

**else**

**k := k + 1;**

**end;**

**//Создание кораблей**

**procedure TForm1.NewShip;**

**var i, k, j : integer;**

**x, y : integer;**

**begin**

**i := 0;**

**k := 0;**

**while (k = 0) do**

**if (i < 100) then**

**begin**

**if (Ship[i] = nil) then**

**begin**

**x := random(600);**

**y := -random(600) - 50;**

**for j := 0 to 100 do**

**if (Ship[j] <> nil) then**

**if (x > Ship[j].GetX) and (x < Ship[j].GetX + 40) then exit**

**else Ship[i] := TShip.Create(x, y, shipPic, 1);**

**k := k + 1;**

**end;**

**i := i + 1;**

**end**

**else**

**k := k + 1;**

**end;**

**//Взрываемся**

**procedure TForm1.NewBoom(x0, y0: integer);**

**var i, k : integer;**

**begin**

**i := 0;**

**k := 0;**

**while (k = 0) do**

**if (i < 100) then**

**begin**

**if (Booms[i] = nil) then**

**begin**

**Booms[i] := TBoom.Create(x0, y0, boomPic, 5);**

**k := k + 1;**

**end;**

**i := i + 1;**

**end**

**else**

**k := k + 1;**

**end;**

**//Таймер для появления врагов**

**procedure TForm1.Timer2Timer(Sender: TObject);**

**begin**

**NewShip;**

**NewEnemy;**

**end;**

**//Таймер для простого полёта перед началом**

**procedure TForm1.Timer3Timer(Sender: TObject);**

**var i : integer;**

**begin**

**if (MoveL) then Player.moveLeft;**

**if (MoveR) then Player.moveRight;**

**if (MoveU) then Player.moveUp;**

**if (MoveD) then Player.moveDown;**

**//Вывод моря**

**for i := 0 to 1 do**

**begin**

**Sea[i].move;**

**Image1.Canvas.Draw(Sea[i].GetX, Sea[i].Gety, Sea[i].getBitmap);**

**end;**

**Player.nextFrame;**

**Image1.Canvas.Draw(Player.GetX, Player.Gety, Player.getBitmap);**

**if (GAME\_STARTED) then**

**begin**

**Timer1.Enabled := true;**

**Timer2.Enabled := true;**

**Timer4.Enabled := true;**

**Timer3.Enabled := false;**

**count\_kill := 0;**

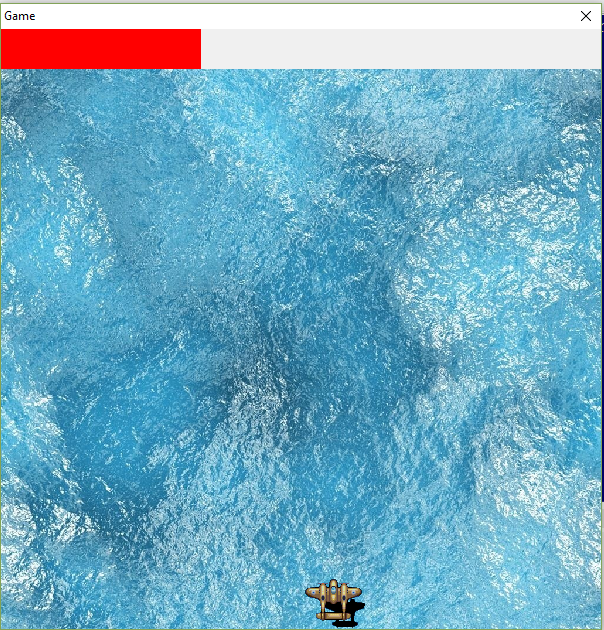
**score := 0;**

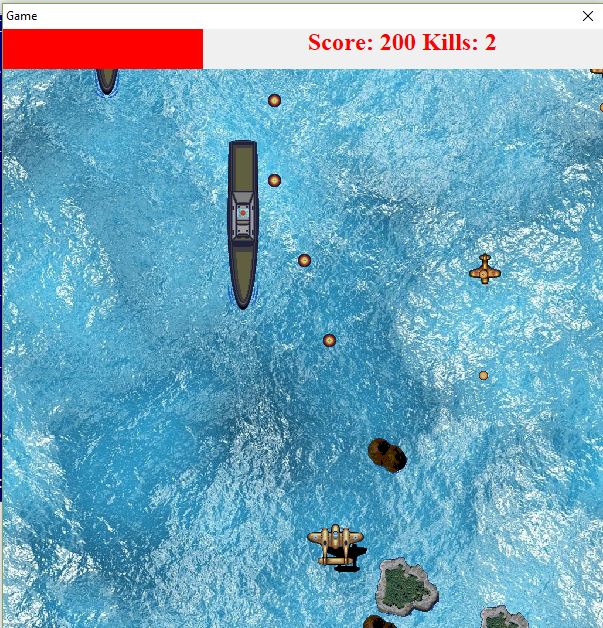
**end;**

**end;**

**end.**

**Скриншоты выполнения программы**

****

****

**Вывод:**

В данной лабораторной работе было разработано игровое приложение. Целью данной игры является нанести максимальный урон сопернику, сохранив при этом свою жизнь. В роли главного игрока выступает боевой самолет, движением которого возможно управлять через стрелки. Движение самолета ограничено окном игры (600х600 пикселов). Также мы можем стрелять пулями, зажимая клавишу пробел.

В игре реализовано живое меню. При запуске мы видим наш самолет, летящий над морем. При движении самолета влево до упора начинается игра.

В роли врагов выступают военные самолеты меньшего размера, а также военные корабли. Раз в определенный период появляется «босс» - вражеский самолет, убить который сложнее, чем других врагов. «Босс» и все вражеские самолеты имеют возможность стрелять пулями. Движения босса повторяют траекторию движения нашего самолета.

При попадании пуль в наш самолет или при столкновении вражеских самолетов с нашим счетчик жизни в верхнем левом углу уменьшается.

В игре реализована статистика – мы имеем возможность увидеть количество убитых нами врагов, а также количество полученных за это очков. За убитого «босса» мы получаем 500 очков, за вражеский корабль 100 очков, за вражеский самолет 50 очков. Скорость движения у объектов игры также разная.

После того, как счетчик жизни иссякнет, игра заканчивается, статистика игры обнуляется, вновь открывается живое меню.