ЗАДАНИЕ

На учебную практику по технологии разработки программного обеспечения

Вам предлагается за время учебной практики ознакомиться со следующими разделами технологии разработки программного обеспечения:

1.Стадия технического задания

1.1 Постановка задачи. Сбор исходной информации. Определение реквизитов задачи

1.2 Определение формы входных и выходных документов. Определение структуры входных и выходных данных

1.3 Классификация объектов по признакам

1.4 Определение требований к программе. Определение стадий и этапов разработки программ и документации

1.5 Разработка тестов для задачи

1.6 Оформление документа «Техническое задание»

2. Стадия технического проекта

2.1 Уточнение структуры входных и выходных данных

2.2 Определение алгоритма решения задач

2.3 Определения формы представления входных и выходных данных. Определение структуры файлов базы данных

2.4 Построение диаграмм комплекса с использованием UML

2.5 Декомпозиция задач. Разработка структуры программы

2.6 Разработка тестов для модулей

2.7 Разработка пояснительной записки

3. Стадия рабочего проекта

3.1 Программирование головной программы

3.2 Программирование модулей

3.3 Комплексное тестирование

3.4 Корректировка программ

3.5 Разработка документов рабочего проекта

3.6 Системное тестирование

Составить:

-документ «Рабочий проект»

-программу «Игра Арканоид (простыестрелялки)». Модуль работы с созданием графических изображений. Модуль работы с движением графических изображений. Модуль работы со справочной системой.

Программа уже готова на АВС паскале

usesABCObjects,Events,GraphABC,Timers,Utils;

var

kLeftKey,kRightKey: boolean;

kSpaceKey: integer;

Player: RectangleABC;

t: integer;

EndOfGame: boolean;

Enemies: TextABC;

StaticObjectsCount: integer;

NewGame: TextABC;

Wins,Falls: integer;

WinsABC,FallsABC: TextABC;

r1: RectangleABC;

type

KeysType=(kLeft,kRight);

Pulya=class(CircleABC)

constructor Create(x,y: integer);

begin

inherited Create(x,y,5,clRed);

dx:=0; dy:=-5;

end;

procedure Move;

var j: integer;

begin

inherited Move;

if Top<0 then

Visible:=False;

for j:=StaticObjectsCount+1 to ObjectsCount do

if (Objects[j]<>Self) and Intersect(Objects[j]) then

begin

Objects[j].Visible:=False;

Visible:=False;

end;

end;

end;

Enemy=class(RectangleABC)

constructor Create(x,y,w: integer);

begin

inherited Create(x,y,w,20,clRandom);

if Random(2)=0 then

dx:=5

else dx:=-5;

dy:=0;

end;

procedure Move;

begin

if Random(2)<>0 then Exit;

if Random(10)=0 then dy:=5;

if (Left<0) or (Left+Width>WindowWidth) or (Random(30)=0) then

dx:=-dx;

inherited Move;

ifdy<>0 then dy:=0;

if Top>WindowHeight-50 then

EndOfGame:=True;

end;

end;

functionNumberOfEnemies: integer;

var i: integer;

begin

Result:=0;

for i:=1 to ObjectsCount do

if Objects[i] is Enemy then

Inc(Result);

end;

procedureCreateObjects;

var

i: integer;

r1: RectangleABC;

begin

Player:=RectangleABC.Create(280,WindowHeight-30,100,20,clTeal);

for i:=1 to 100 do

begin

r1:=Enemy.Create(Random(WindowWidth-50),40+Random(10),50);

r1.TextVisible:=True;

r1.Number:=i;

end;

end;

procedureDestroyObjects;

var i: integer;

begin

for i:=ObjectsCount downto StaticObjectsCount+1 do

Objects[i].Destroy;

end;

procedureMoveObjects;

var i: integer;

begin

for i:=StaticObjectsCount+2 to ObjectsCount do

Objects[i].Move;

end;

procedureDestroyKilledObjects;

var i: integer;

begin

for i:=ObjectsCount downto StaticObjectsCount+2 do

if not Objects[i].Visible then

Objects[i].Destroy;

end;

procedureKeyDown(Key: integer);

begin

case Key of

vk\_Left: kLeftKey:=True;

vk\_Right: kRightKey:=True;

vk\_Space: if kSpaceKey=2 then kSpaceKey:=1;

end;

end;

procedureKeyUp(Key: integer);

begin

case Key of

vk\_Left: kLeftKey:=False;

vk\_Right: kRightKey:=False;

vk\_Space: kSpaceKey:=2;

end;

end;

procedure Timer1;

var

i,j,n: integer;

p: Pulya;

begin

ifkLeftKey and (Player.Left>0) then

Player.MoveOn(-10,0);

ifkRightKey and (Player.Left+Player.Width<WindowWidth) then

Player.MoveOn(10,0);

ifkSpaceKey=1 then

begin

p:=Pulya.Create(Player.Left+Player.Width div 2,Player.Top-10);

kSpaceKey:=0;

end;

MoveObjects;

DestroyKilledObjects;

RedrawObjects;

n:=NumberOfEnemies;

Enemies.Text:='Врагов: '+IntToStr(n);

if n=0 then

EndOfGame:=True;

ifEndOfGame then

begin

StopTimer(t);

if n>0 then

begin

Inc(Falls);

FallsABC.Text:='Поражений: '+IntToStr(Falls);

RedrawObjects;

ShowMessage('Выпроиграли!');

DestroyObjects;

Enemies.Text:='Врагов: '+IntToStr(NumberOfEnemies);

RedrawObjects;

end

else

begin

Inc(Wins);

WinsABC.Text:='Побед: '+IntToStr(Wins);

RedrawObjects;

ShowMessage('Вывыиграли!');

DestroyObjects;

Enemies.Text:='Врагов: '+IntToStr(NumberOfEnemies);

RedrawObjects;

end;

end;

end;

procedureMouseUp(x,y,mb: integer);

begin

ifNewGame.PTInside(x,y) then

StartTimer(t);

end;

procedureKeyPress(Key: char);

begin

if ((UpCase(Key)='G') or (UpCase(Key)='П')) and EndOfGame then

begin

EndOfGame:=False;

StartTimer(t);

CreateObjects;

kSpaceKey:=2;

kLeftKey:=False;

kRightKey:=False;

end;

end;

begin

LockDrawingObjects;

EndOfGame:=True;

r1:=RectangleABC.Create(0,0,WindowWidth,38,clWhite);

NewGame:=TextABC.Create(WindowWidth-180,5,14,clBlack,'G - Новаяигра');

Enemies:=TextABC.Create(10,5,14,clRed,'Врагов: '+IntToStr(NumberOfEnemies));

WinsABC:=TextABC.Create(150,5,14,clRed,'Побед: '+IntToStr(Wins));

FallsABC:=TextABC.Create(280,5,14,clRed,'Поражений: '+IntToStr(Falls));

StaticObjectsCount:=ObjectsCount;

Enemies.Text:='Врагов: '+IntToStr(NumberOfEnemies);

OnKeyDown:=KeyDown;

OnKeyPress:=KeyPress;

OnKeyUp:=KeyUp;

OnMouseUp:=MouseUp;

t:=CreateTimer(10,Timer1);

StopTimer(t);

RedrawObjects;

end.

1.Стадия технического задания

1.1 Постановка задачи. Сбор исходной информации. Определение реквизитов задачи

1.2 Определение формы входных и выходных документов. Определение структуры входных и выходных данных

1.3 Классификация объектов по признакам

1.4 Определение требований к программе. Определение стадий и этапов разработки программ и документации

1.5 Разработка тестов для задачи

1.6 Оформление документа «Техническое задание»

2. Стадия технического проекта

2.1 Уточнение структуры входных и выходных данных

2.2 Определение алгоритма решения задач

2.3 Определения формы представления входных и выходных данных. Определение структуры файлов базы данных

2.4 Построение диаграмм комплекса с использованием UML

2.5 Декомпозиция задач. Разработка структуры программы

2.6 Разработка тестов для модулей

2.7 Разработка пояснительной записки

3. Стадия рабочего проекта

3.1 Программирование головной программы

3.2 Программирование модулей

3.3 Комплексное тестирование

3.4 Корректировка программ

3.5 Разработка документов рабочего проекта

3.6 Системное тестирование

20-25 страниц Times New Roman 14 пт, межстрочный интервал 1.

Срок 28 ноября утром.