PascalABC.NET – это современная реализация языка Паскаль, которая сочетает в себе простоту и мощные возможности. Язык отличается строгой типизацией, что позволяет учащимся лучше понимать структуру данных и логику программирования. Ясная синтаксическая структура делает его доступным для начинающих, а также способствует формированию правильных привычек в написании кода

PascalABC.NET для обучения программированию в школах выбирают благодаря легкости освоения для начинающих, богатой стандартной библиотеке, поддержке графики и работы с файлами, возможности создания консольных и оконных приложений.

В данной статье предлагается несколько учебных проектов для реализации на уроках информатики и внеурочных занятиях.

Игра «Три в ряд»

В данной игре реализованы процедуры, множественные циклы, циклы с постусловием, а также модуль для работы с графикой GraphABC.

**uses** GraphABC**;**

**type**

Coordinat **=** **record**

x**,** y **:** integer**;**

**end;**

**var**

bg **:** Picture**;**

PicArr**:array[**1**..**100**,**1**..**100**]** **of** Picture**;**

map **:** **array[**1**..**100**,**1**..**100**]** **of** integer**;**

col **:** List**<**Coordinat**>** **:=** new List**<**Coordinat**>();**

zn **:** integer**;**

q**:** Coordinat**;**

xp**,**yp **:** integer**;**

i**,**j **:** integer**;**

W**,**H**,** size **:** integer**;**

bEnd**,** bClick **:** boolean**;**

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

**begin**

// MoveTo(x,y);

xp **:=** x**;**

yp **:=** y**;**

**if** map**[**yp **div** size**,** xp **div** size**]** **<>** 0 **then**

bClick **:=** true**;**

**end;**

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

**begin**

TextOut**(**0**,**0**,**'x = '**+** IntToStr**(**x **div** size**)+** ' y = '**+** IntToStr**(** y **div** size**));**

**end;**

//---------------------------------------------------//

// Вывод рисунков

//---------------------------------------------------//

**procedure** ShowArr**;**

**var**

i**,**j **:** integer**;**

**begin**

**for** i**:=** 1 **to** **(**W **div** size**)-**2 **do**

**for** j**:=** 1 **to** **(**H **div** size**)-**2 **do** **begin**

PicArr**[**i**,**j**].**Load**(**map**[**j**,**i**]+**'.png'**);**

PicArr**[**i**,**j**].**Draw**(** i**\***size **,** j**\***size **);**

**end;**

**end;**

//---------------------------------------------------//

// обход по кругу

//---------------------------------------------------//

**procedure** FindElem **(** y**,** x **:** integer**;** n **:** integer **);**

**begin**

map**[**y**,**x**]** **:=** 0**;**

**if** **(**y**-**1 **>** 0**)** **then**

**if** **(**n **=** map**[**y**-**1**,**x**])** **then** **begin** map**[**y**-**1**,**x**]:=**0**;** q**.**x **:=** x**;** q**.**y **:=** y**-**1**;** col**.**Add**(**q**);** **end;**

**if** **(**y**+**1 **<** H **div** size **)** **then**

**if** **(**n **=** map**[**y**+**1**,**x**])** **then** **begin** map**[**y**+**1**,**x**]:=**0**;** q**.**x **:=** x**;** q**.**y **:=** y**+**1**;** col**.**Add**(**q**);** **end;**

**if** **(**x**-**1 **>** 0**)** **then**

**if** **(**n **=** map**[**y**,**x**-**1**])** **then** **begin** map**[**y**,**x**-**1**]:=**0**;** q**.**x **:=** x**-**1**;** q**.**y **:=** y**;** col**.**Add**(**q**);** **end;**

**if** **(**x**+**1 **<** W **div** size**)** **then**

**if** **(**n **=** map**[**y**,**x**+**1**])** **then** **begin** map**[**y**,**x**+**1**]:=**0**;** q**.**x **:=** x**+**1**;** q**.**y **:=** y**;** col**.**Add**(**q**);** **end;**

**end;**

**function** FindElemCount **(** y**,** x **:** integer**;** n **:** integer **)** **:** integer**;**

**var**

c **:** integer**;**

**begin**

c **:=** 1**;**

**if** **(**y**-**1 **>** 0**)** **then**

**if** **(**n **=** map**[**y**-**1**,**x**])** **then** inc**(**c**);**

**if** **(**y**+**1 **<** H **div** size**)** **then**

**if** **(**n **=** map**[**y**+**1**,**x**])** **then** inc**(**c**);**

**if** **(**x**-**1 **>** 0**)** **then**

**if** **(**n **=** map**[**y**,**x**-**1**])** **then** inc**(**c**);**

**if** **(**x**+**1 **<** W **div** size**)** **then**

**if** **(**n **=** map**[**y**,**x**+**1**])** **then** inc**(**c**);**

Result **:=** c**;**

**end;**

**procedure** ArrRewCol**(**ic **:** integer**);**

**var**

st**,** st2 **:** integer**;**

b **:** boolean**;**

**begin**

b **:=** true**;**

**for** **var** k**:=**13 **downto** 1 **do** **begin**

//

**for** **var** i**:=**13 **downto** 1 **do**

**if** map**[**i**,**ic**]** **=** 0 **then** **begin** st **:=** i**;** break**;** **end;**

//

**for** **var** j**:=**st **downto** 1 **do**

**if** map**[**j**,**ic**]** **<>** 0 **then** **begin** st2 **:=** j**;** b **:=** true**;** break**;** **end** **else** b **:=** false**;**

//

**if** **(**b**)** **then**

**for** **var** i**:=** st2 **to** st**-**1 **do**

**begin** map**[**i**+**1**,**ic**]:=** map**[**i**,**ic**];** map**[**i**,**ic**]:=**0**;** **end;**

**end;** // for k

**end;**

//---------------------------------------------------//

// //!!!!!!!!!!!!!

// смещение в право

//---------------------------------------------------//

**procedure** ArrRewRow**(**ic **:** integer**);**

**var**

st**,** st2 **:** integer**;**

b **:** boolean**;**

**begin**

b **:=** true**;**

**for** **var** k**:=** 18 **downto** 1 **do** **begin**

//

**for** **var** i**:=**18 **downto** 1 **do**

**if** map**[**ic**,**i**]** **=** 0 **then** **begin** st **:=** i**;** break**;** **end;**

//

**for** **var** j**:=**st **downto** 1 **do**

**if** map**[**ic**,**j**]** **<>** 0 **then** **begin** st2 **:=** j**;** b **:=** true**;** break**;** **end** **else** b **:=** false**;** // false

**if** **(**b**)** **then**

**for** **var** i**:=** st2 **to** st**-**1 **do**

**for** **var** jc**:=** 1 **to** 18 **do**

**begin**

map**[**jc**,**i**+**1**]:=** map**[**jc**,**i**];**

map**[**jc**,**i**]:=**0**;**

**end;**

**end;** // for k

**end;**

**procedure** ArrRewALL**;**

**begin**

**for** **var** i**:=**1 **to** 18 **do**

ArrRewCol**(**i**);**

**end;**

**begin**

W **:=** Window**.**Width**;**

H **:=** Window**.**Height**;**

size **:=** 32**;**

bEnd **:=** false**;**

bClick **:=** false**;**

Randomize**;**

**for** i**:=** 1 **to** **(**W **div** size**)-**2 **do**

**for** j**:=** 1 **to** **(**H **div** size**)-**2 **do** **begin**

map**[**j**,**i**]** **:=** Random**(**1**,**2**);**

PicArr**[**i**,**j**]:=** Picture**.**Create**(**map**[**j**,**i**]+**'.png'**);**

// PicArr[i,j].Draw( j\*size , i\*size);

**end;**

bg **:=** Picture**.**Create**(**'bg.jpg'**);**

bg**.**Draw**(**0**,**0**);**

ShowArr**;**

**repeat**

**if** **(**bClick**)** **then** **begin**

TextOut**(**0**,**10**,**inttostr**(**xp **div** size**)+**' '**+**inttostr**(**yp **div** size**));**

zn **:=** map**[**yp **div** size**,** xp **div** size**];**

q**.**x **:=** xp **div** size**;**

q**.**y **:=** yp **div** size**;**

col**.**Add**(**q**);**

**if** **(** FindElemCount**(**col**[**0**].**y**,** col**[**0**].**x**,** zn **)** **>** 1**)** **then**

**begin**

**repeat**

FindElem**(**col**[**0**].**y**,** col**[**0**].**x**,** zn**);**

col**.**RemoveAt**(**0**);**

**until(** col**.**Count **=** 0**);**

ArrRewALL**;**

ArrRewRow**(**13**);**

bg**.**Draw**(**0**,**0**);**

ShowArr**;**

**end** **else** col**.**RemoveAt**(**0**);**

bClick **:=** false**;**

**end;** // if bClick

OnMouseDown **:=** MouseDown**;**

OnMouseMove **:=** MouseMove

**until** **(**bEnd**);**

**end.**

На рисунках 1 и 2 представлен результат работы программы. Нужно нажимать на елочные шары, расположенные в ряд по цвету, постепенно открывая картинку.

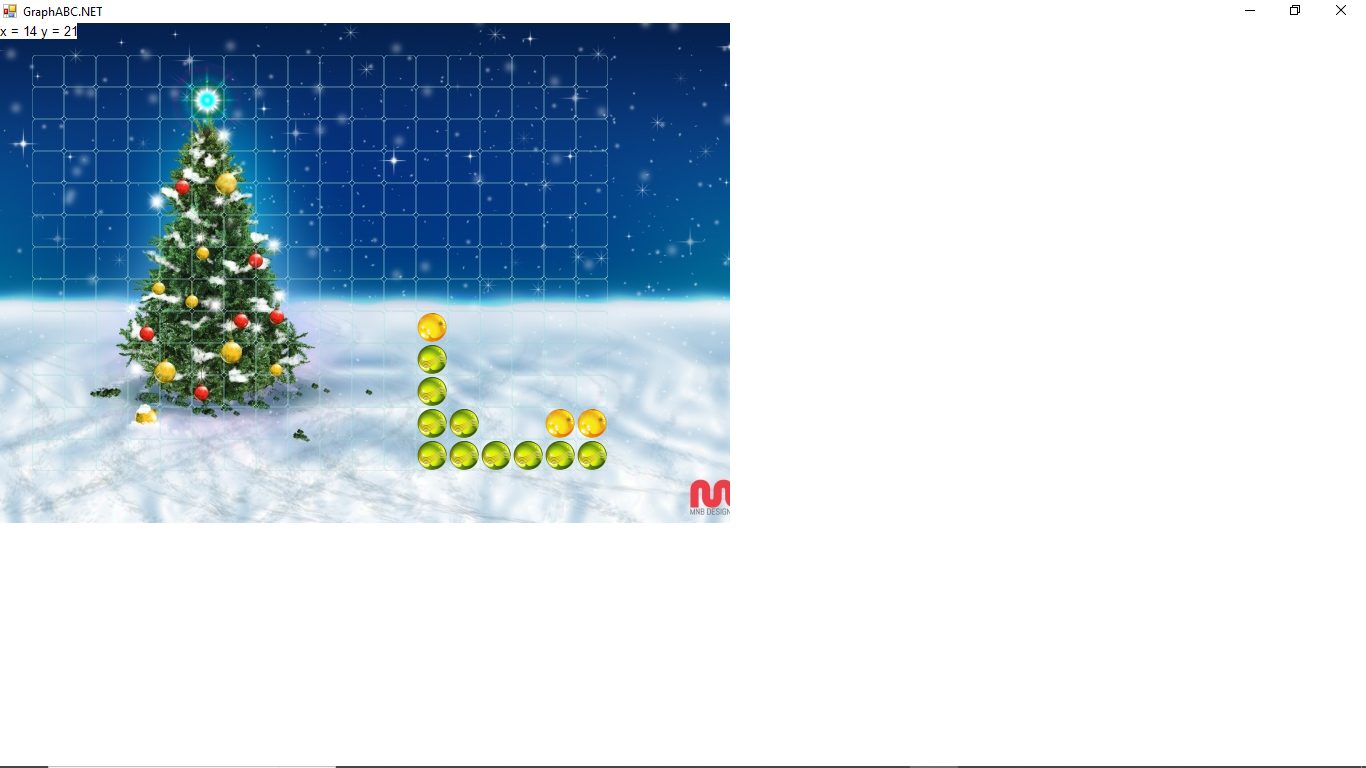
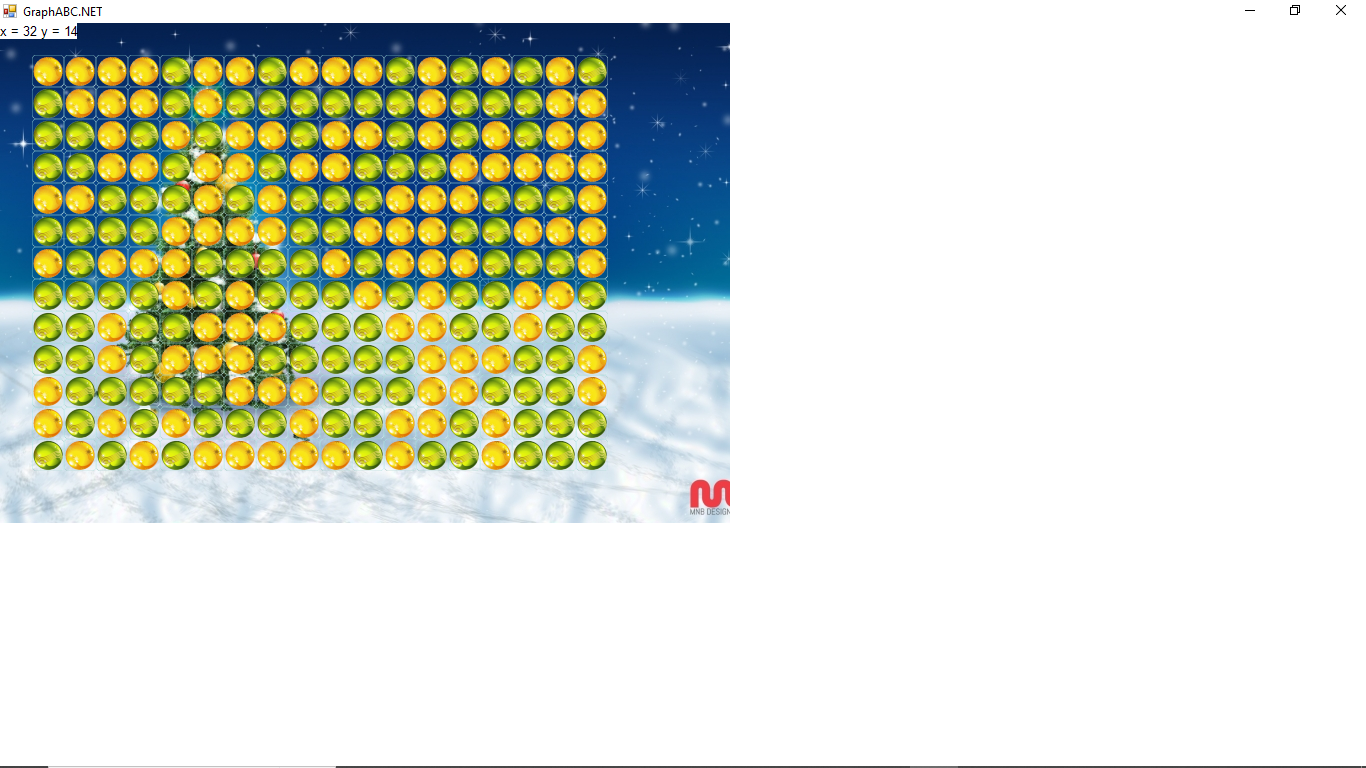


Рисунок 1 – Начало игры Рисунок 2 – В процессе игры

Игра «Одень снеговика»

В данном проекте использован модуль ABCObjects, который реализует векторные графические объекты с возможностью масштабирования, наложения друг на друга, создания составных графических объектов и многократного их вложения друг в друга. Каждый векторный графический объект корректно себя перерисовывает при перемещении, изменении размеров и частичном перекрытии другими объектами.

Модуль ABCObjects предназначен для раннего обучения основам объектно-ориентированного программирования, а также для реализации графических и анимационных проектов средней сложности. Он реализован на основе модуля GraphABC.

Также в данном проекте применены INI-файлы – это текстовые файлы, в которых хранятся пары «ключ – значение», сгруппированные в разделы. Их используют для сохранения пользовательских настроек. В языке Pascal для работы с INI-файлами используют модуль IniFiles и класс TINIFile.

**uses**

GraphABC**,** ABCObjects**,** Inifile **;**

// Описание комплект. для снеговика

**type**

TSnowMan **=** **record**

Pic**:** PictureABC**;** // рисунок части тила/одежды

Index **:** integer**;** // номер загружаемого рисунка

**end;**

// Описание снега

**type**

TSnow **=** **record**

X**,** Y**:** integer**;** // координаты старта

dX**,** dY**:** integer**;** // смещение снежинки

Pic**:** PictureABC**;** // рисунок

**end;**

**const**

CountSnow **=** 100**;** // количество снежинок

**var**

fEND**:** boolean**;** // проверка на завершение игрового цикла

W**,** H **:** integer**;** // область отрисовки

Snow **:** **array** **of** TSnow**;** // массив снежинок

bgr**:** PictureABC**;** // загрузка объектов сцены

SnowMan **:** PictureABC**;** // загрузка снеговика

SMDetail**:** **array** **of** TSnowMan**;** // массив для хранения активных вещей

Fileini **:** TIniFile**;** // для загрузки INI-файла

// массив и имена Button

Button**:** **array** **of** PictureABC**;**

NameButton **:** **array** **of** **string** **:=**

**(**'BGolova'**,**'BGlaza'**,**'BNos'**,**'BRot'**,**'BSharf'**,**'BRuki'**,**'BMetla'**,**'BTelo'**,**'BPodarok'**);**

//------------------------------------------------------//

//-------- Чтение данных из файла ini --------------//

**procedure** LoadPictureOfIni**(** int **:** integer**;** str **:** **string** **);**

**var**

px**,**py **:** integer**;**

sx**,**sy **:** real**;**

**begin**

**var** path **:=** 'data/SMan/'**;**

**var** count **:=** Fileini**.**ReadInteger**(**'List'**,** str**,** **-**1**);**

**if** **(**SMDetail**[**int**].**Index **<** count **)** **then**

SMDetail**[**int**].**Index **:=** SMDetail**[**int**].**Index **+** 1

**else** SMDetail**[**int**].**Index **:=** 1**;**

px **:=** Fileini**.**ReadInteger**(**str**+**IntToStr**(**SMDetail**[**int**].**Index**),**'PosX'**,-**1**);**

py **:=** Fileini**.**ReadInteger**(**str**+**IntToStr**(**SMDetail**[**int**].**Index**),**'PosY'**,-**1**);**

sx **:=** StrToFloat**(**Fileini**.**ReadString **(**str**+**IntToStr**(**SMDetail**[**int**].**Index**),**'ScalX'**,**'-1'**));**

sy **:=** StrToFloat**(**Fileini**.**ReadString**(**str**+**IntToStr**(**SMDetail**[**int**].**Index**),**'ScalY'**,**'-1'**));**

SMDetail**[**int**].**Pic**.**MoveTo**(** px **,**py **);**

SMDetail**[**int**].**Pic**.**ScaleX **:=** sx**;**

SMDetail**[**int**].**Pic**.**ScaleY **:=** sy**;**

SMDetail**[**int**].**Pic**.**ChangePicture**(**path**+**str**+**'/'**+**str**+**SMDetail**[**int**].**Index**+**'.png'**);**

**end;**

//------------------------------------------------------//

//--- Обработчик нажатия по кнопке мыши ---------------//

// В данном случае идет обработка по нажатию

// на кнопки "одежда снеговика"

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

**begin**

**var** ob1 **:=** ObjectUnderPoint**(**x**,** y**);**

**case** ob1**.**Number **of**

1**:** LoadPictureOfIni**(** 0**,** 'Golova' **);**

2**:** LoadPictureOfIni**(** 1**,** 'Glaza' **);**

3**:** LoadPictureOfIni**(** 2**,** 'Nos' **);**

4**:** LoadPictureOfIni**(** 3**,** 'Rot' **);**

5**:** LoadPictureOfIni**(** 4**,** 'Sharf' **);**

6**:** LoadPictureOfIni**(** 5**,** 'Ruki' **);**

7**:** LoadPictureOfIni**(** 6**,** 'Metla' **);**

//8:

//9:

**end;** // case

**end;**

//------------------------------------------------------//

//-- Процедура проверяющая выход снега за придела экрана

//-- и устанавливающая в начальные координаты

//------------------------------------------------------//

**procedure** UpdataSnow**;**

**var**

i**:** integer**;**

**begin**

**for** i **:=** 0 **to** CountSnow**-**1 **do** **begin**

Snow**[** i **].**pic**.**Moveto**(**Snow**[** i **].**x**,** Snow**[** i **].**y**);**

Snow**[** i **].**y **:=** Snow**[** i **].**y **+** Snow**[** i **].**dy**;**

Snow**[** i **].**x **:=** Snow**[** i **].**x **+** Snow**[** i**]** **.**dx**;**

**if** **(**Snow**[** i **].**y **>** H**)or(**Snow**[** i **].**x **>** W**)** **then** **begin**

Snow**[** i **].**y **:=** **-**10**;**

Snow**[** i **].**x **:=** Random**(**3**\***W**)-**W**;**

**end;**

**end;**

**end;**

//------------------------------------------------------//

//------ Загрузка сцены -----------------------------//

//------------------------------------------------------//

**procedure** LoadScene**(** n **:** integer **);**

**begin**

**case** **(** n **)** **of**

0**:**

**begin**

bgr **:=** PictureABC**.**Create**(**0**,** 0**,** 'data/Background/bgr\_1.png'**);**

bgr **:=** PictureABC**.**Create**(**0**,** 125**,** 'data/Background/bgr\_2.png'**);**

**end;**

1**:**

**begin**

bgr **:=** PictureABC**.**Create**(**0**,** 0**,** 'data/Background/bgr\_3.png'**);**

bgr**.**ScaleX **:=** 1.14**;**

**end;**

**end;**

**end;**

//------------------------------------------------------//

//----------- Загрузка кнопок ---------------------//

**procedure** LoadButton**;**

**var**

x**,** y **:** integer**;**

**begin**

y **:=** 10**;** x **:=** 690**;** // начальные координаты первой кнопки

Button **:=** new PictureABC**[**9**];**

**for** **var** i **:=** 0 **to** 8 **do** **begin**

Button**[**i**]** **:=** PictureABC**.**Create**(**x**,** y**,** 'data/Button/'**+**NameButton**[**i**]+**'.png'**);**

Button**[**i**].**ScaleX **:=** 0.6**;** Button**[**i**].**ScaleY **:=** 0.6**;**

Button**[**i**].**Number **:=** i**+**1**;** Button**[**i**].**TextVisible **:=** false**;**

**if** **(** y **+** 120 **<** 500 **)** **then** y **:=** y **+** 120 **else**

**begin** y **:=** 10**;** x **:=** x **-** 120**;** **end;**

**end;**

**end;**

//------------------------------------------------------//

//----------- Загрузка одежды снеговика -----------//

**procedure** LoadSnowmanDetail**;**

**begin**

SMDetail **:=** new TSnowMan**[**9**];**

SMDetail**[**0**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**1**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**3**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**2**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**5**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**4**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**6**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**7**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

SMDetail**[**8**].**Pic **:=** PictureABC**.**Create**(**100**,** 100**,** 'data/null.png'**);**SMDetail**[**0**].**Index **:=** 0**;**

**end;**

//------------------------------------------------------//

//---------- Главная часть программы ------------------//

//------------------------------------------------------//

**begin**

Window**.**Width **:=** 800**;**

Window**.**Height **:=** 600**;**

Window**.**IsFixedSize **:=** true**;** // запрет на изменения размера окна

W **:=** 600**;** // ограничение для снега

H **:=** 400**;** // ограничение для снега

Randomize**;**

// загружается случайный фон локации

**var** x **:=** Random**(**2**);**

LoadScene**(**x**);**

// Загружаем падающий снег

Snow **:=** new TSnow**[**CountSnow**];**

**for** **var** i **:=** 0 **to** CountSnow**-**1 **do** **begin**

Snow**[** i **].**X **:=** Random**(**3**\***W**)-**W**;**

Snow**[** i **].**Y **:=** Random**(**H**);**

Snow**[** i **].**Pic **:=** PictureABC**.**Create**(**Snow**[** i **].**x**,**Snow**[** i **].**y**,**'data/Background/snow1.png'**);**

Snow**[** i **].**dY **:=** Random**(**3**)+**1**;**

Snow**[** i **].**dX **:=** Random**(**3**)** **-** 1**;**

**end;**

// Загрузка и установка снеговика

SnowMan **:=** PictureABC**.**Create**(**150**,** 160**,** 'data/snowman2.png'**);**

// Загрузка INI-файла

Fileini **:=** TIniFile**.**Create**(**'data/res.ini'**);**

// Первоначальная загрузка одежды снеговика

LoadSnowmanDetail**;**

// Отображение кнопок

LoadButton**;**

// Game Loop

**repeat**

UpdataSnow**;**

// Таймер последней отрисовки

// время для отрисовки одного кадра

// Window.Caption := IntToStr( MillisecondsDelta);

// Обработчики прерываний

OnMouseDown **:=** MyMouseDown**;**

**until** **(**fEND**);** // end gameloop

**end.**

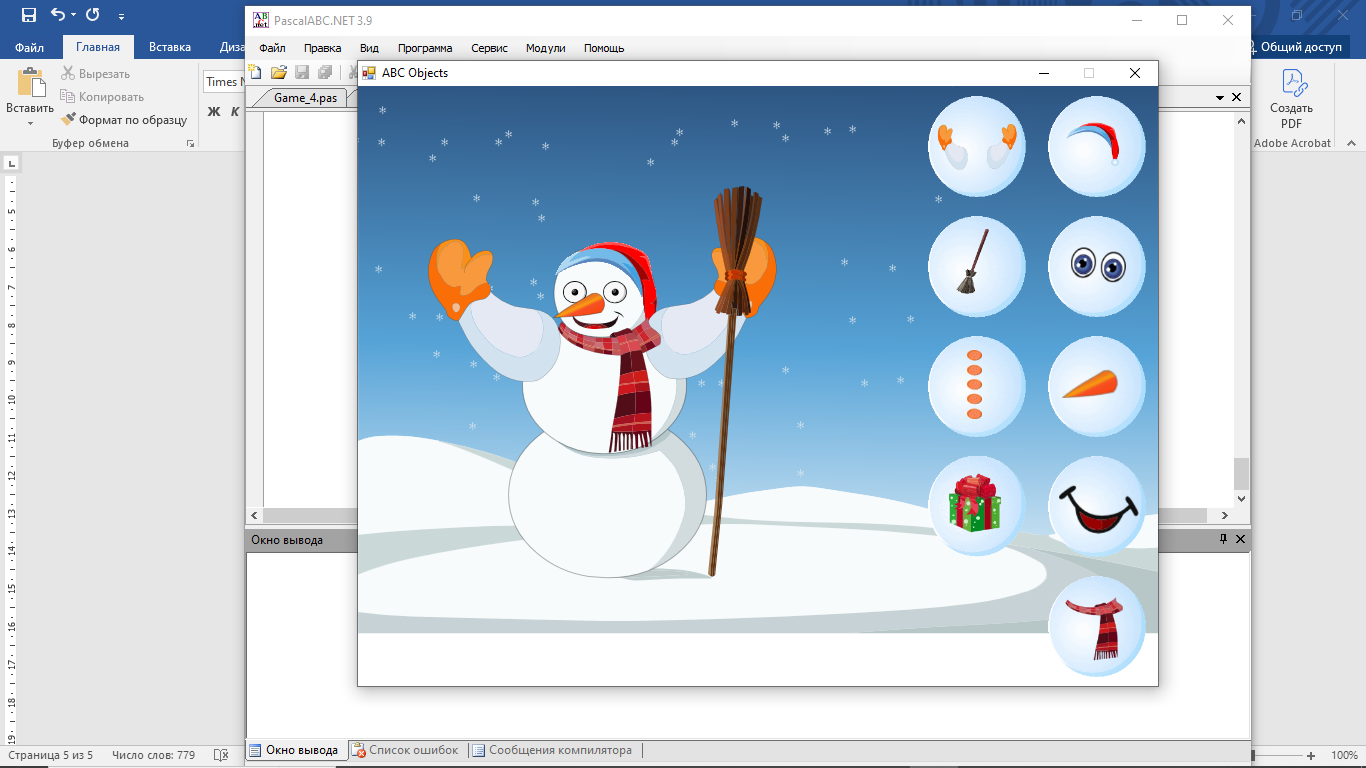


Рисунок 3 – Результат работы программы