Республиканская научно – практическая конференция школьников

«Первые шаги в науку»

Направление: Информатика и информационные технологии

Название работы: Программируем на Delphi 6

Автор работы: Зергетаев Эрдни Басангович

Место выполнения работы: п. Цаган Аман Юстинского района

Муниципальное бюджетное образовательное учреждение

«Цаганаманская гимназия»

Руководитель:

Улюмджиева Н.Б., учитель информатики и математики

2013 г.

Оглавление

Введение

Разработка собственных приложений в наше время вызывает интерес у многих людей, чьи умения в сфере программирования позволяют осуществить поставленные задачи.

Браузер – самая распространенная программа, имеющаяся на каждом персональном компьютере или на мобильном устройстве. Браузер – это окно в интернет, благодаря которому мы можем пользоваться всеми возможностями современных информационных технологий. Поэтому целью моей работы является разработка и создания браузера в среде программирования Delphi 6, а также написание игрового приложения, главная задача которого заставить пользователя отвлечься от повседневных проблем, развлечься, что, собственно является задачей любых игр.

Ставились следующие задачи:

- ✓ Изучить литературу по вопросу создания программ в среде программирования Delphi 6;
- ✓ Освоение создания более сложных программ в данной среде программирования;
- ✓ Создать браузер и игру «Кликомания».

Среда программирования Delphi 6 была выбрана неслучайно. Delphi 6 – это язык программирования, написанный с удобным графическим интерфей1сом для более удобного создания программ.

Как любая подобная система, Delphi предназначена для разработки программ и имеет две характерные особенности: создаваемые ею программы могут работать не только под управлением Windows, а сама она относится к классу инструментальных средств ускоренной разработки программ. Это ускорение достигается за счет следующих свойств Delphi: визуального конструирования форм и широкого использования библиотеки визуальных компонентов.

Визуальное конструирование форм избавляет программиста от многих аспектов разработки интерфейса программы, так как Delphi автоматически готовит необходимые программные заготовки и соответствующий файл ресурсов.

При создании программ используется специальное окно, которое называется окном формы, как прототип будущего окна программы, оно наполняется компонентами, реализующими нужные интерфейсные свойства (разного рода списки, кнопки, полосы прокрутки и т. п.). Компоненты находятся в библиотеке визуальных компонентов. Она предоставляет программисту огромное разнообразие программных заготовок, которые немедленно или после несложной настройки готовы к работе в рамках программы.

Использование компонентов не только во много раз уменьшает сроки разработки программ, но и существенно снижает вероятность случайных программных ошибок.

В Delphi можно составлять проекты для задач практически любого типа:
это и расчетные задачи, и задачи работы с файлами, и обработка баз данных, и другие. Наиболее эффектными и интересными являются задачи моделирования различных логических игр. Они привлекают внимание пользователей и формируют устойчивый интерес к изучению языков программирования.

1. Создание браузера в среде программирования Delphi 6

1.1.Компоненты Delphi 6, используемые в программе

Компонент TWebBrowser

Во многих современных программах необходимо работать с данными в формате HTML. В качестве средства для просмотра таких данных в Delphi применяется компонент TWebBrowser, который использует элемент управления ActiveX WebBrowser, входящий в состав Microsoft Internet Explorer.

Установим на форму компонент "WebBrowser" (он находится на закладке "Internet" или "ActiveX"), появится белый квадрат с именем WebBrowser1. (рис.2)

Компонент CoolBar

Он позволяет строить перестраиваемые панели, состоящие из полос (bands). В полосы могут включаться инструментальные панели ToolBar и любые другие оконные компоненты: окна редактирования, панели и т.п. Каждый из этих компонентов автоматически снабжается средствами перемещения его пользователем в пределах окна CoolBar. В полосы могут вставляться и не оконные компоненты, например, метки. Но они не будут перемещаемыми.

Установим на форму "CoolBar", который находится на закладке "Win32" палитры компонентов. Теперь выделим WebBrowser1 и перейдем в ObjectInspector. Щелкнем по свойству "Align" и в выпадающем списке выберем "alClient". "WebBrowser" должен растянуться на все свободное место формы.

Компонент ToolBar

Компонент Delphi ToolBar представляет собой инструментальную панель, для быстрого доступа к часто используемым функциям нашей программы, позволяет управлять компоновкой быстрых кнопок и компонентов.

Компонент ComboBox

Компонент Delphi ComboBox представляет собой комбинацию списка строк ListBox со строкой ввода Edit. При этом "список строк" компонента Delphi ComboBox вначале скрыт, и раскрывается при щелчке мышкой по треугольнику раскрытия, который находится справа в строке ввода. Таким образом, с помощью Delphi ComboBox место на Форме экономится для расмещения других элементов интерфейса программы. А при необходимости раскрытие списка строк можно вообще запретить.

Теперь установим на CoolBar1 панель "ToolBar" из закладки "Win32" и "ComboBox" из закладки "Standart" палитры компонентов. Все это помещаем именно внутрь CoolBar1. После этого нужно выделить CoolBar1 и перейти в ObjectInspector. Здесь изменяем строку "AutoSize" на "true" (по умолчанию она "false").

Выделяем ComboBox1 (выпадающий список) и переходим в ObjectInspector. Здесь выделяем закладку "Events" и производим двойной щелчок по строке "OnKeyDown". Двойной щелчок нужно производить в правой половине строки. Delphi создаст процедуру. Она будет вызываться каждый раз, когда пользователь будет вводить какую-нибудь букву в "ComboBox". В процедуре пишем:

```
procedure TForm1.ComboBox1KeyDown(Sender: TObject;
 var Key: Word; Shift: TShiftState);
begin
 if Key = VK_RETURN then
 WebBrowser1.Navigate(ComboBox1.Text);
end;
```

1.2. Улучшаем свой браузер

Выделяем ToolBar1 и снова переходим в ObjectInspector. Здесь нужно изменить свойства "AutoSize", "ShowCaption" и "Flat" на "true" (все они по умолчанию равны false). Теперь щелкаем правой кнопкой по ToolBar1 и из появившейся меню выбираем пункт "New Button". На ToolBar1 должна появиться новый объект с именем "ToolButton1". Выделим его и в ObjectInspector поменяем свойство Caption на "Открыть". Создаем еще насколько кнопок с заголовками: Назад, Вперед, Стоять, Обновить и Печать.

Установим еще на форму "OpenDialog" из закладки "Dialogs" палитры компонентов.

Теперь дважды щелкаем по кнопке "Открыть", и Delphi автоматически создаст процедуру, которая будет вызываться при нажатии этой кнопки. В этой процедуре нужно написать следующее:

```
procedure TForm1.ToolButton1Click(Sender: TObject);
begin
 if OpenDialog1.Execute then
 begin
 WebBrowser1.Navigate(OpenDialog1.FileName);
 ComboBox1.Text := OpenDialog1.FileName;
 end;
end;
```

Теперь мы можем запустить программу и открыть с помощью этой кнопки любой файл на диске. Заставим работать остальные кнопки. Дважды щелкаем по кнопке "Назад".

```
procedure TForm1.ToolButton2Click(Sender: TObject);
begin
```

```
 WebBrowser1.GoBack;
end;
```

Повторяем те же операции для кнопки "Вперед", чтобы создать процедуру.

```
procedure TForm1.ToolButton3Click(Sender: TObject);
begin
 WebBrowser1.GoForward;
end;
```

для кнопки «Стоять»

```
procedure TForm1.ToolButton4Click(Sender: TObject);
begin
 WebBrowser1.Stop;
end;
```

Для кнопки "Обновить":

```
procedure TForm1.ToolButton5Click(Sender: TObject);
begin
 WebBrowser1.Refresh;
end;
```

для кнопки "Печать":

```
procedure TForm1.ToolButton6Click(Sender: TObject);
var
 PostData, Headers: OLEvariant;
begin
 WebBrowser1.ExecWB(OLECMDID_PRINT, OLECMDEXECOPT_DODEFAULT, PostData,
 Headers);
end;
```

Установим на форму StatusBar из закладки "Win32" и изменим у него свойство "SimplePanel" в true (по умолчанию false). Теперь выделим WebBrowser1 и щелкнем по закладке "Events" в ObjecInspector-е. Дважды щелкаем по строке "OnStatusTextChange" и пишем в созданной процедуре следующее:

```
procedure TForm1.WebBrowser1StatusTextChange(Sender: TObject;
 const Text: WideString);
begin
 StatusBar1.SimpleText := Text;
end;
```

Здесь мы присваиваем переменную "Text" (в ней хранится текст подсказки) в StatusBar1. Теперь ты сможешь видеть подсказки в строке состояния.

Добавим ещё индикатор загрузки. Для этого поместим на форму ProgressBar из закладки "Win32". Изменим у него свойство "Align" на "alBottom", чтобы он находился вдоль нижней границы формы. Снова выделим WebBrowser1 и щелкнем по закладке "Events" в ObjecInspector-е. Дважды щелкнем по строке "OnProgressChange" и напишем в созданной процедуре:

```
procedure TForm1.WebBrowser1ProgressChange(Sender: TObject; Progress,
 ProgressMax: Integer);
begin
 ProgressBar1.Max := ProgressMax;
 ProgressBar1.Position := Progress;
end;
```

Здесь мы созданному ProgressBar1 (индикатор загрузки) присваиваем максимальное значение (ProgressMax) и текущее значение (Progress).

Теперь надо украсить наши кнопки. Для этого помещаем на форму ImageList и производим по нему двойной щелчок . Перед нами откроется окно. Сюда нужно добавить картинки размером 16x16. Для этого нажмем кнопку "Add", и откроется стандартное окно открытия файла. Найдем картинку и нажмем "Открыть". Повторим эту процедуру 6 раз (6 картинок для 6-и кнопок).

Теперь выделим ToolBar1 и в ObjectInspector-е изменим свойство Images на "ImageList1". На твоих кнопках должны появиться картинки. (рис.3)

1.3. Принципы работы с браузером

Разработанный браузер использует ядро Internet Explorer, который установлен на компьютере. Также добавлена возможность просматривать страницы и изображения, сохранённые на компьютере. Кроме выше описанного, браузер может обновить страницу, остановить обновление, вывести на печать текущую страницу или изображение, содержит кнопки назад или вперёд. URL-адрес вводится нажатием клавиши Enter.

2. Создание игры «Кликомания» в среде программирования Delphi 6

Создадим игру, суть которой за отведенное время успеть найти все фигуры одинакового цвета, я назвал ее «Кликомания».

Открываем Delphi и создаем новый проект. Добавляем на форму компонент DrawGrid с закладки Additional. Изменяем его свойства следующим образом (рис.4.):

1. Свойство ColCount - сделайте равным 20
2. Свойство RowCount - тоже 20
3. Свойство DefaultColWidth = 20
4. Свойство DefaultRowHeight = 20
5. FixedCols = 0
6. FixedRows = 0
7. DefaultDrawing = False

Затем присваиваем каждой ячейке определенный цвет. Всего цветов у нас будет 3 (красный, желтый и фиолетовый).

Далее, нам понадобиться создать двухмерный массив, в ячейках которого будут храниться цвета соответствующих квадратов. т.к. размер игровой области у нас 20 на 20, соответственно и массив у нас будет такой же размерности.

Нам нужно создать процедуру, которая будет при запуске игры случайным образом назначать квадратам соответствующие цвета.

Следующим шагом даем возможность игроку начинать новую игру тогда когда он этого захочет, для этого помещаем на форму компонент Button (кнопка) с закладки Standard и и меняем у него свойство Caption на "Новая Игра". Создаем обработчик событий OnClick на кнопке, кликнув на ней два раза.

Конечно же, при запуске, также необходимо начинать новую игру, поэтому создаем обработчик событий onCreate на форме.

Теперь игровое поле заполнено квадратами разного цвета, осталось только научиться удалять квадраты одинакового цвета.

Последним шагом добавляем таймер, который будет вести обратный отчет, если игрок не успел найти все квадраты за указанное время то Game Over. (рис. 5.)

Заключение

В результате выполнения работы мной были разработаны программа для просмотра веб - страниц и игровая программа под названием “Кликомания”. Было проведено исследование компонентов программной среды Borland Delphi 6.0, которые использовались при создании программ, использовалось множество процедур и функций.

Созданная игровая программа может быть использована для обучения детей различать цвета, фигуры, для отработки навыка работы с мышкой. Приложение можно усовершенствовать, сделать, так что бы при удалении блока одинакового цвета, квадраты которые находились на верху падали вниз, тем самым заполняли пустое пространство.

Браузер работает, минус в том, что новые страницы открываются не в нем, а в Internet Explorer. Есть над, чем еще поработать.

Программы малотребовательны к системным ресурсам компьютера.

В результате учета всех сделанных выше замечаний возможно улучшение созданных программ.

Список литературы

1. Михаил Фленов «Библия Delphi» - М.:БХВ-Петербург, 2004г. 880с.
2. Михаил Голованов, Евгений Веселов «Создание компонентов в среде Delphi», - М.:БХВ-Петербург, 2004г., 545с.
3. Информатика в школе: Приложение к журналу «Информатика и образование». № 1-2007
4. http://articles.org.ru/cn/?c=1

Приложения

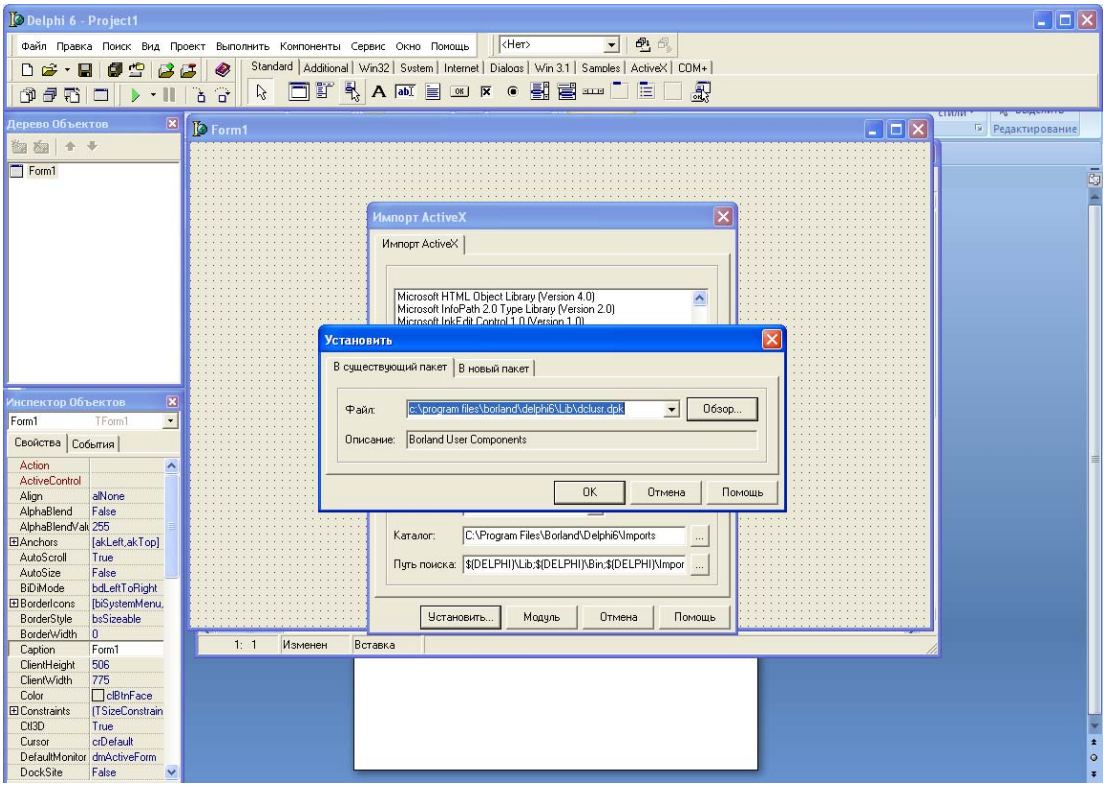

Рис.1. Установка TWebBrowser

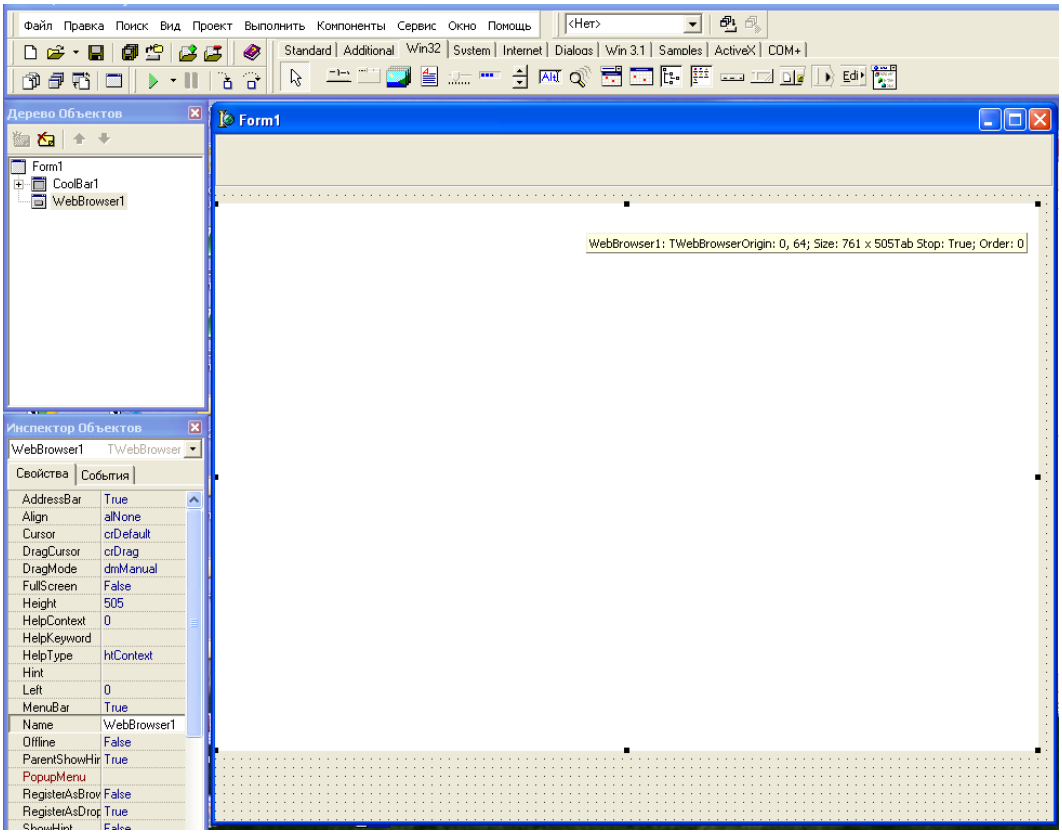

Рис. 2. Установка на форму компонента "WebBrowser"

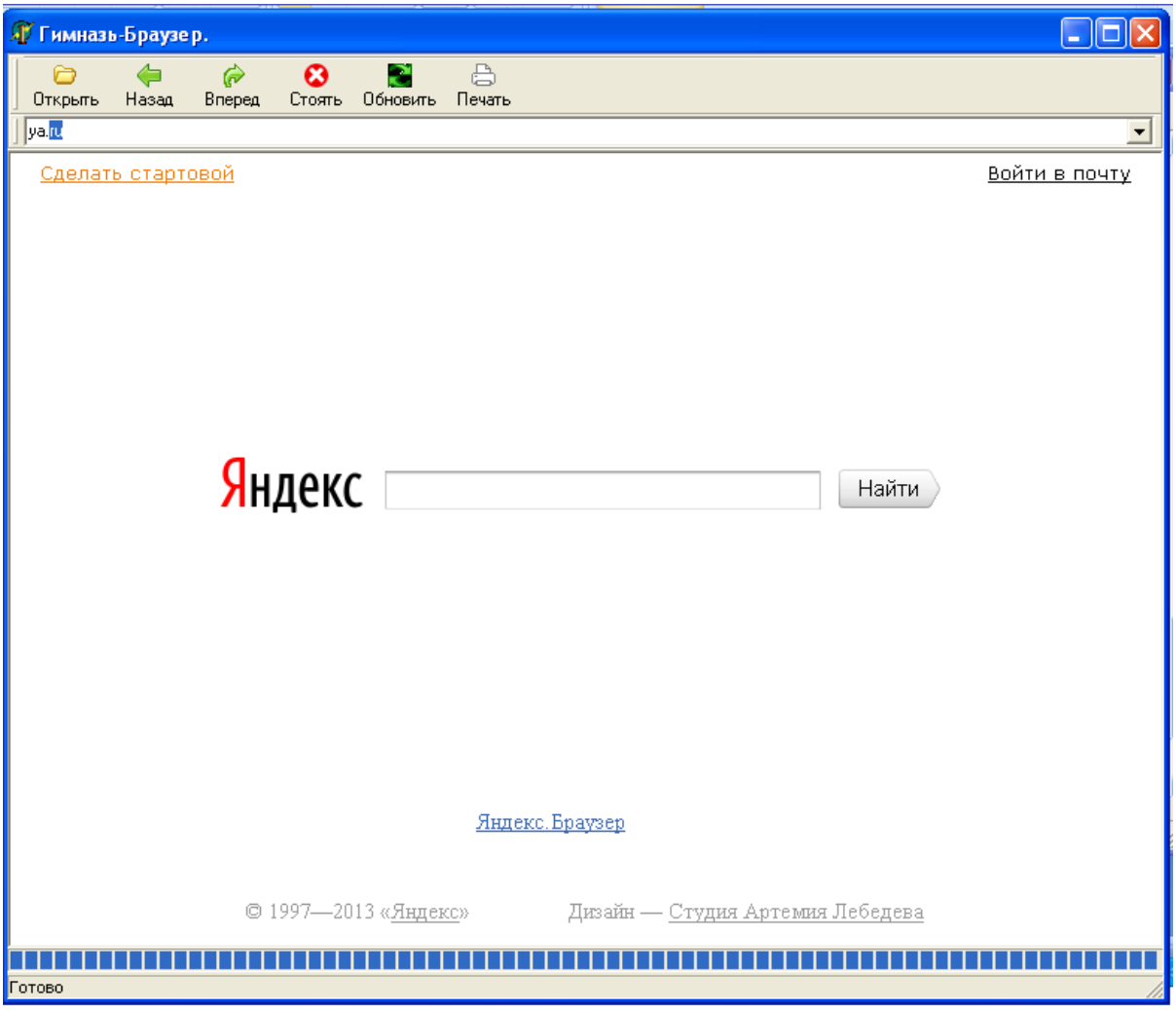

Рис.3. Созданный браузер

Листинг программы «Браузер»

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ToolWin, OleCtrls, SHDocVw_TLB, ImgList;
type
  TForm1 = class(TForm)
    WebBrowser1: TWebBrowser;
    CoolBar1: TCoolBar;
    ToolBar1: TToolBar;
    ComboBox1: TComboBox;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    OpenDialog1: TOpenDialog;
    StatusBar1: TStatusBar;
    ProgressBar1: TProgressBar;
    ImageList1: TImageList;
    procedure ComboBox1KeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure ToolButton1Click(Sender: TObject);
    procedure ToolButton2Click(Sender: TObject);
    procedure ToolButton3Click(Sender: TObject);
    procedure ToolButton4Click(Sender: TObject);
    procedure ToolButton5Click(Sender: TObject);
    procedure ToolButton6Click(Sender: TObject);
    procedure WebBrowser1StatusTextChange(Sender: TObject;
      const Text: WideString);
    procedure WebBrowser1ProgressChange(Sender: TObject; Progress,
      ProgressMax: Integer);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.ComboBox1KeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if Key = VK_RETURN then
    WebBrowser1.Navigate(ComboBox1.Text);
end;
procedure TForm1.ToolButton1Click(Sender: TObject);
begin
  if OpenDialog1.Execute then begin
    WebBrowser1.Navigate(OpenDialog1.FileName);
    ComboBox1.Text:=OpenDialog1.FileName;
  end;
end;
procedure TForm1.ToolButton2Click(Sender: TObject);
begin
  WebBrowser1.GoBack;
end;
procedure TForm1.ToolButton3Click(Sender: TObject);
begin
  WebBrowser1.GoForward;
end;
procedure TForm1.ToolButton4Click(Sender: TObject);
begin
  WebBrowser1.Stop;
end;
```

```
procedure TForm1.ToolButton5Click(Sender: TObject);
begin
  WebBrowser1.Refresh;
end;
procedure TForm1.ToolButton6Click(Sender: TObject);
var
  PostData, Headers : OLEvariant;
begin
  WebBrowser1.ExecWB(OLECMDID_PRINT, OLECMDEXECOPT_DODEFAULT, PostData,
Headers);
end;
procedure TForm1.WebBrowser1StatusTextChange(Sender: TObject;
  const Text: WideString);
begin
  StatusBar1.SimpleText:=Text;
end;
procedure TForm1.WebBrowser1ProgressChange(Sender: TObject; Progress,
  ProgressMax: Integer);
begin
  ProgressBar1.Max := ProgressMax;
  ProgressBar1.Position := Progress;
end;
end.
```

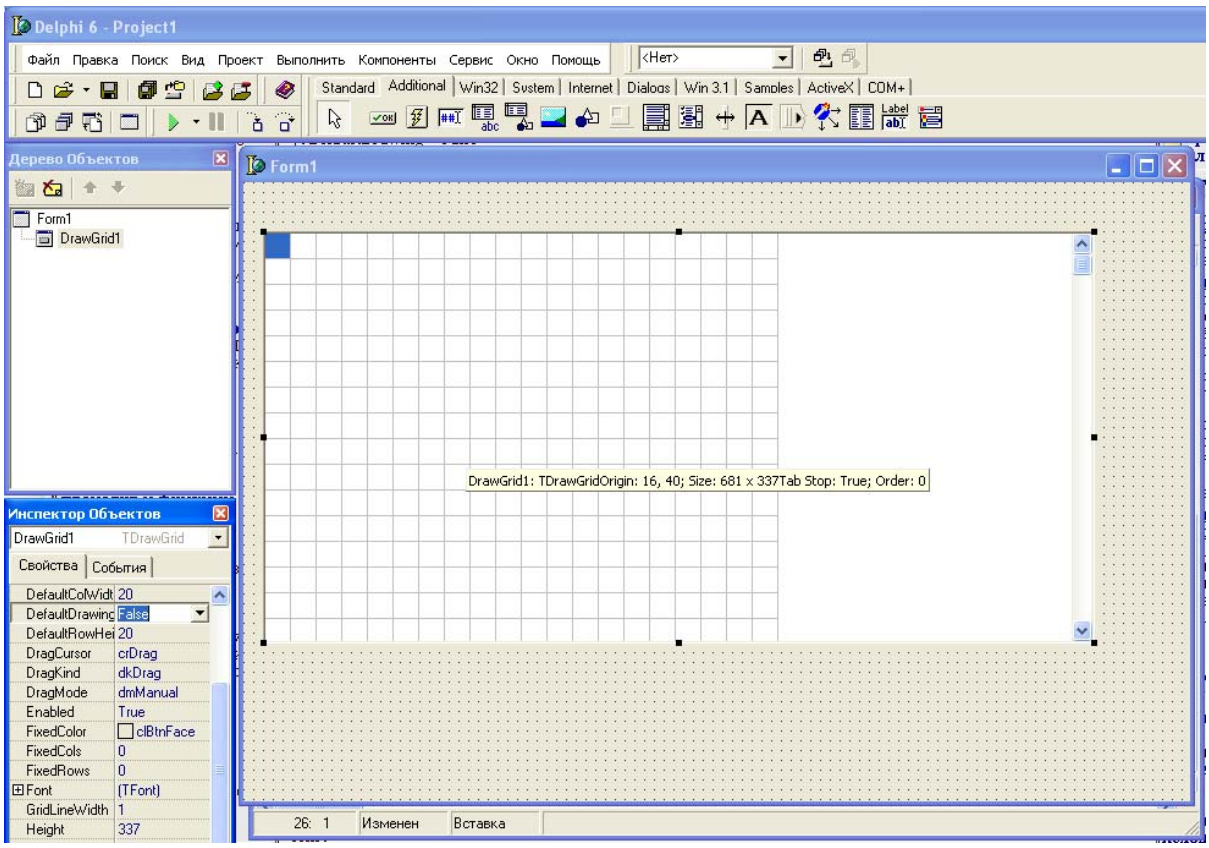

Рис. 4. компонент DrawGrid

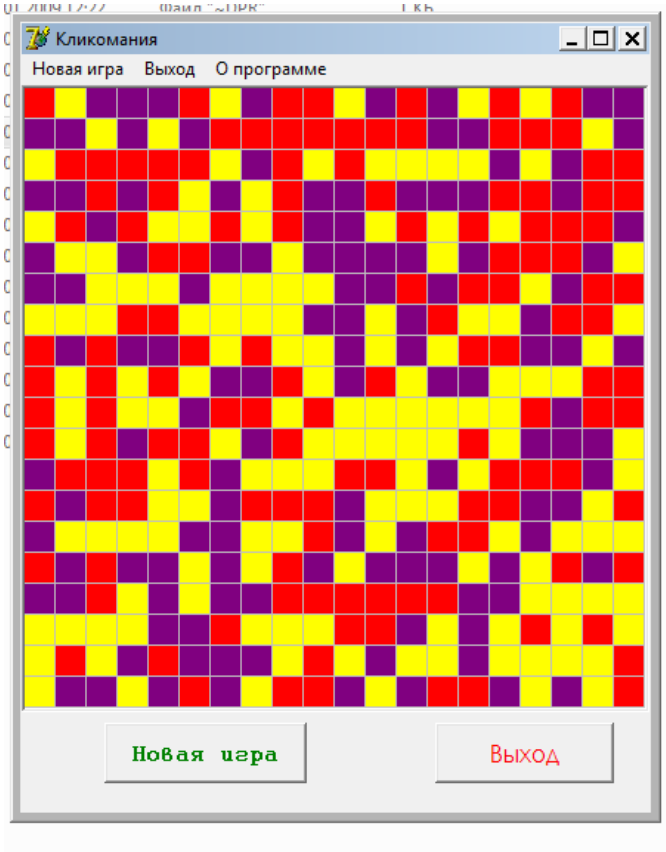

Рис.5. игра «Кликомания»

Листинг программы «Кликомания»

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, StdCtrls, Buttons, jpeg, ExtCtrls, Menus;
type
  TForm1 = class(TForm)
    DrawGrid1: TDrawGrid;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    procedure FormCreate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure DrawGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;
      Rect: TRect; State: TGridDrawState);
    procedure DrawGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
      var CanSelect: Boolean);
    procedure BitBtn2Click(Sender: TObject);
    procedure N1Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure N3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure AssignBrickColors;
    procedure RedrawCells;
    procedure RemoveBricks (cl : TColor; ACol, ARow : integer);
    function IsBrickAlone(ACol, ARow : integer) : Boolean;
  end;
```

```
  const
  MAX_COLORS = 3;
const
  PossibleColors : array [0..MAX_COLORS-1] of TColor = (clYellow, clPurple, clRed);
  const
  MAX_COLUMNS = 20;
  MAX_ROWS = 20;
  const
  WALL_COLOR : TColor = clWhite;
var
  Form1: TForm1;
   ColorOfBrick : array [0..MAX_COLUMNs, 0..MAX_ROWS] of TColor;
implementation
uses Unit2;
{$R *.dfm}
{ TForm1 }
procedure TForm1.AssignBrickColors;
var
  i, j : integer;
begin
 for i := 0 to Form1.DrawGrid1.ColCount-1 do
    for j := 0 to Form1.DrawGrid1.RowCount-1 do
      ColorOfBrick[i][j] := PossibleColors[Random(MAX_COLORS)]
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
 Randomize;
 Bitbtn1.Click ;
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
 AssignBrickColors;
  RedrawCells;
end;
```

```
procedure TForm1.DrawGrid1DrawCell(Sender: TObject; ACol, ARow: Integer;
  Rect: TRect; State: TGridDrawState);
begin
DrawGrid1.Canvas.Brush.Color := ColorOfBrick[ACol][ARow];
DrawGrid1.Canvas.FillRect(Rect)
end;
procedure TForm1.RedrawCells;
var
  i, j : integer;
begin
  for i := 0 to Form1.DrawGrid1.ColCount-1 do
    for j := 0 to Form1.DrawGrid1.RowCount-1 do
      Form1.DrawGrid1DrawCell(Form1, i, j, Form1.DrawGrid1.CellRect(i,j), [])
end;
procedure TForm1.RemoveBricks(cl: TColor; ACol, ARow: integer);
begin
  if ColorOfBrick[ACol, ARow] <> cl then
    exit;
ColorOfBrick[ACol, ARow] := WALL_COLOR;
  if ACol > 0 then
    RemoveBricks(cl, ACol-1, ARow);
  if ACol < Form1.DrawGrid1.ColCount-1 then
    RemoveBricks(cl, ACol+1, ARow);
  if ARow > 0 then
    RemoveBricks(cl, ACol, ARow-1);
  if ARow < Form1.DrawGrid1.RowCount-1 then
    RemoveBricks(cl, ACol, ARow+1);
end;
function TForm1.IsBrickAlone(ACol, ARow: integer): Boolean;
begin
  Result := True;
  if ColorOfBrick[ACol, ARow] = WALL_COLOR then
    exit;
```

```
  if ACol > 0 then
    if ColorOfBrick[ACol-1, ARow] = ColorOfBrick[ACol, ARow] then
      Result := False;
  if ACol < Form1.DrawGrid1.ColCount-1 then
    if ColorOfBrick[ACol+1, ARow] = ColorOfBrick[ACol, ARow] then
      Result := False;
  if ARow > 0 then
    if ColorOfBrick[ACol, ARow-1] = ColorOfBrick[ACol, ARow] then
      Result := False;
  if ARow < Form1.DrawGrid1.RowCount-1 then
    if ColorOfBrick[ACol, ARow+1] = ColorOfBrick[ACol, ARow] then
      Result := False
end;
procedure TForm1.DrawGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
begin
  if not IsBrickAlone(ACol, ARow) then
  begin
    RemoveBricks(ColorOfBrick[ACol, ARow], ACol, ARow);
    RedrawCells
  end
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
 close;
end;
procedure TForm1.N1Click(Sender: TObject);
begin
  AssignBrickColors;
  RedrawCells;
end;
procedure TForm1.N2Click(Sender: TObject);
begin
  close;
```

```
end;
procedure TForm1.N3Click(Sender: TObject);
begin
  Form2.Show;
end;
end.
```