

Лабораторная работа №1 «Основы OpenGL»

Библиотека OpenGL — это стандартная библиотека для всех операционных систем. Для работы с этой библиотекой необходимо список `uses` дополнить модулем **OpenGL**. В раздел `private` описания класса формы необходимо добавить строку

```
hrc:HGLRC; // ссылка на контекст воспроизведения
```

Графическая система OpenGL, как и любое другое приложение Windows (хоть и размещенное в DLL), также нуждается в ссылке на устройство, на которое будет осуществляться вывод. Это специальная ссылка на контекст воспроизведения – величина типа HGLRC (Handle OpenGL Rendering Context, ссылка на контекст воспроизведения OpenGL).

Ссылка на контекст устройства – это величина типа HDC. Для ее получения можно вызвать функцию **GetDC**, аргументом которой является ссылка на нужное окно. Ссылке на контекст устройства соответствует свойство `canvas` Handle формы Delphi. Даже если мы рисуем только на поле формы, мы имеем дело с различными устройствами нам неизвестно, какова графическая плата компьютера и каковы характеристики текущей установки настроек экрана. Например, имея в своем распоряжении более 16 миллионов цветов, приложение не заботится об отображении этой богатой палитры на экране, располагающем всего 256 цветами. Такие вопросы приложение перекладывает на плечи операционной системы, решающей их посредством использования драйверов устройств. Для того чтобы воспользоваться функциями воспроизведения Windows, приложению необходимо только указать ссылку на контекст устройства, содержащий средства и характеристики устройства вывода.

Для корректной работы программы необходимо описать пользовательскую процедуру, определяющую формат пикселя. Это необходимо, поскольку даже на одной операционной системе при разном разрешении экрана пиксель будет выглядеть по-разному.

В разделе **implementation** формы опишите следующую процедуру:

```
procedure SetDCPixelFormat (hdc: HDC);  
var pfd: TPixelFormatDescriptor;  
    nPixelFormat: Integer;  
begin  
    FillChar (pfd, SizeOf (pfd), 0);  
    pfd.dwFlags:=PFD_SUPPORT_OPENGL PFD_DRAW_TO_WINDOW or PFD_DOUBLEBUFFER;  
    nPixelFormat := ChoosePixelFormat (hdc, @pfd);  
    SetPixelFormat (hdc, nPixelFormat, @pfd);  
end;
```

Сервер OpenGL, прежде чем приступить к работе, также должен определиться, на каком оборудовании ему придется работать. Прежде чем получить контекст воспроизведения, сервер OpenGL должен получить детальные характеристики используемого оборудования. Эти характеристики хранятся в специальной структуре, тип которой – **TPixelFormatDescriptor** (описание формата пикселя). Формат пикселя определяет конфигурацию буфера цвета и вспомогательных буферов. Смысл структуры **PixelFormatDescriptor** – детальное описание графической системы, на которой происходит работа.

Полям структуры присваиваются желаемые значения, затем вызовом функции **choosePixelFormat** осуществляется запрос системе, поддерживается ли на данном рабочем месте выбранный формат пикселя, и, наконец, вызовом функции **SetPixelFormat** устанавливается формат пикселя в контексте устройства. Функция **choosePixelFormat** возвращает индекс формата пикселя, который нужен в качестве аргумента функции **SetPixelFormat**.

Заполнив поля структуры **TPixelFormatDescriptor**, мы определяемся со своими пожеланиями к графической системе, на которой будет происходить работа приложения, OpenGL подбирает наиболее подходящий к нашим пожеланиям формат и устанавливает уже его в качестве формата пикселя для последующей работы. Наши пожелания корректируются сервером OpenGL применительно к реальным характеристикам системы.

В данном примере установлены следующие пожелания к формату пикселя: поддержка OpenGL, осуществления вывода в окно, и поддержка режима двойной буферизации.

Минимальная программа OpenGL должна содержать 4 обработчика событий: **onCreate**, **onDestroy**, **onPaint**, **onResize**.

Обработчик **onCreate** срабатывает при создании формы. Его код следующий:

```
SetDCPixelFormat(Canvas.Handle); //задаем формат пиксела  
hrc:=wglCreateContext(Canvas.Handle); // создаем контекст воспроизведения
```

Во второй строке обработчика **onCreate** задается величина типа **HGLRC**, т.е. создается контекст воспроизведения. Аргументом функции **wglCreateContext** является ссылка на контекст устройства, на который будет осуществляться вывод. Сейчас устройством вывода служит окно формы. Для получения этого контекста OpenGL необходима величина типа **HDC**. **Canvas.Handle** и есть ссылка на контекст устройства, связанная с окном формы.

Обработчик **onDestroy** срабатывает при закрытии приложения. Работа данного обработчика состоит в освобождении контекста воспроизведения.

```
wglDeleteContext (hrc);
```

Обработчик **onPaint** в простейшем случае выглядит следующим образом:

```
wglMakeCurrent (Canvas.Handle, hrc); // установить контекст  
glViewport (0, 0, ClientWidth, ClientHeight); // область вывода  
glClearColor (0.5, 0.5, 0.75, 1.0); // цвет фона  
glClear (GL_COLOR_BUFFER_BIT); //очистка буфера кадра  
SwapBuffers(Canvas.Handle) // вывод содержимого буфера на экран
```

Обработчик **onResize** в простейшем случае совпадает с обработчиком **onPaint**. Данная программа выводит окно с заполненным цветом фоном.