Създаване на OpenGL прозорец с GLUT

Последната версия на библиотеката GLUT се намира на сайта www.opengl.org. След като я свалите ще трябва да я инсталирате към системата. Библиотеката се състои от 3 файла – glut32.dll, glut32.lib и glut32.h, като всеки от тях трябва да бъде разположен на правилното място:

- glut32.dll – поставете в папката Windows;
- glut32.lib – съответно при останалите *.lib файлове в папката LIB на вашия компилатор, а най-вероятно това е Microsoft Visual C++. Ето директорията: *C:/ Microsoft Visual Studio/ VC98/ LIB*, където C:/ е името на съответното дисково устройство;
- glut32.h – съответно при останалите хедърните файлове в папката Include, но най-добре ще е да го сложите в подпапката GL. Директорията *C:/ Microsoft Visual Studio/ VC98/ Include/ GL*.

Много е важно при построяване на вашия проект да изберете Win32 Console Application, а не Win32 Application. След като сте готови с проекта трябва да свържете библиотеките: Project/Settings/Link/ в реда Object/Library Modules пред kernel32.lib трябва да добавите файловете opengl32.lib и glut32.lib.

Скелет на OpenGL програмите:

1. Първо трябва да включите хедърните файлове;
2. След това идва ред на функцията main( ), която приема целочисления аргумент argc и масива от низови указатели *argv [ ];
3. Инициализираме библиотеката GLUT с функцията : glutInit (int *argc, char *argv [ ]);
4. След това трябва да пределим режима на рендериране с функцията glutInitDisplayMode, която приема няколко аргумента, разделени със символ “I”. Това са:

- GLUT_SINGLE или GLUT_DOUBLE за единичен или двоен буфер. Винаги използвайте двоен буфер, освен ако няма да рендерирате някоя 2D фигура. При GLUT_DOUBLE само единият буфер е видим, а в другия се рендерира графиката и след това двата буфера се разменят.;

- GLUT_DEPTH, GLUT_STENCIL и/или GLUT_ACCUM : Първата константа определя наличието на z-буфер. Ако правите триизмерна графика, неговото включване е задължително.;

- GLUT_RGBA или GLUT_INDEX определят начина на изграждане на цветовете в програмата. При GLUT_INDEX цвета се определя от т.нар. цветова палитра, която може да съдържа 16 или 256 цвята и на всяко число от 0 до 15 или от 0 до 255 съответства точно определен цвят, но този стандарт вече е остарял и дори възпрепятства използването на други ефекти, като динамична светлина, например. Използвайте винаги GLUT_RGBA, при който цвета се сформира от количествата на RED, GREEN, BLUE цветовете и ALPHA канала (полупрозрачност).;

5. Сега трябва да определим големината на прозореца с функцията: glutInitWindowSize (int width, int height);
6. Създаваме прозореца с функцията: glutCreateWindow (char*string), която приема избраното от нас име на прозореца;

7. Извикваме функцията glutDisplayFunc (void (*func)(void)), която извиква на свой ред нашата рендерираща функция, всеки път когато прозорецът трябва да бъде прерисуван. Това може да се случи, когато прозорецът се извиква за първи път, когато е повреден или ако се извика функцията glutPostRedisplay( ).;
8. Накрая извикваме функцията glutMainLoop ( ), коят стартира цикъла на съобщенията.

Нашата main функция е вече готова.

Следва нашата рендерираща функция:

void Render ( )

Първо трябва да изчистим буфера от предишното изображение. Това става с функцията glClear ( ). Тя може да приема няколко от следните аргументи, разделени с "I":

- GL_COLOR_BUFFER_BIT – изчиства цветовия буфер;
- GL_DEPTH_BUFFER_BIT – изчиства z-буфера;
- GL_ACCUM_BUFFER_BIT – изчиства Accumulation буфера;
- GL_STENCIL_BUFFER_BIT - изчиства Stencil буфера.

Друга изключително полезна функция е glClearColor( ), с която можете да запълните цветовия буфер с определен цвят или по-просто казано да определите цвета на фона. Нейният прототип е :
void glClearColor(GLclampf red, GLclampf green, GLclampf blue, GLclampf alpha) – тя приема RGBA цвета на фона.

Ако ползвате двоен буфер, задължително трябва да извикате функцията glutSwapBuffer( ), която разменя двата буфера. В случай, че предпочетете единичен буфер, трябва да извикате функцията glFlush ( ).

Функцията glutPostRedisplay ( ) съобщава на GLUT, че прозореца трябва да бъде прерисуван. Това води до извикване на glutDisplayFunc (), която пък извиква рендериращата ни функция. Този цикъл е задължителен, т.к. в противен случай програмата ще покаже само първият кадър от 3D графиката и ще спре.