Графика OpenGL

Семестр 1

Семинар 8

OpenGL

- открытая графическая библиотека
- спецификация, то есть документ, описывающий набор функций и их точное поведение
- кроссплатформенность (Windows, Unix, Playstation 3, Mac OS)
- создание
  - компьютерных игр,
  - виртуальной релаьности,
  - систем автоматизированного проектирования,
  - визуализации в научных исследованиях
- не путать с DirectX (только Windows)
- SGI, 3Dlabs, Matrox, Evans & Sutherland, ATI, NVIDIA, IBM, Apple, Dell, Hewlett-Packard, Sun Microsystems, id Software

GLUT

- OpenGL Utility Toolkit
- «надстройка» над OpenGL
- полная кроссплатформенность
- библиотека, которая в основном отвечает за системный уровень операций ввода-вывода при работе с операционной системой
- не требует знаний API операционной системы

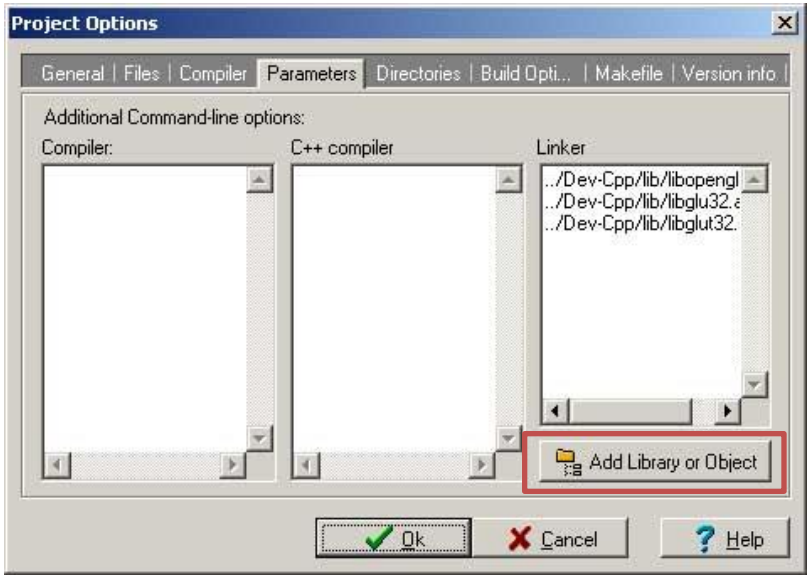
Возможности

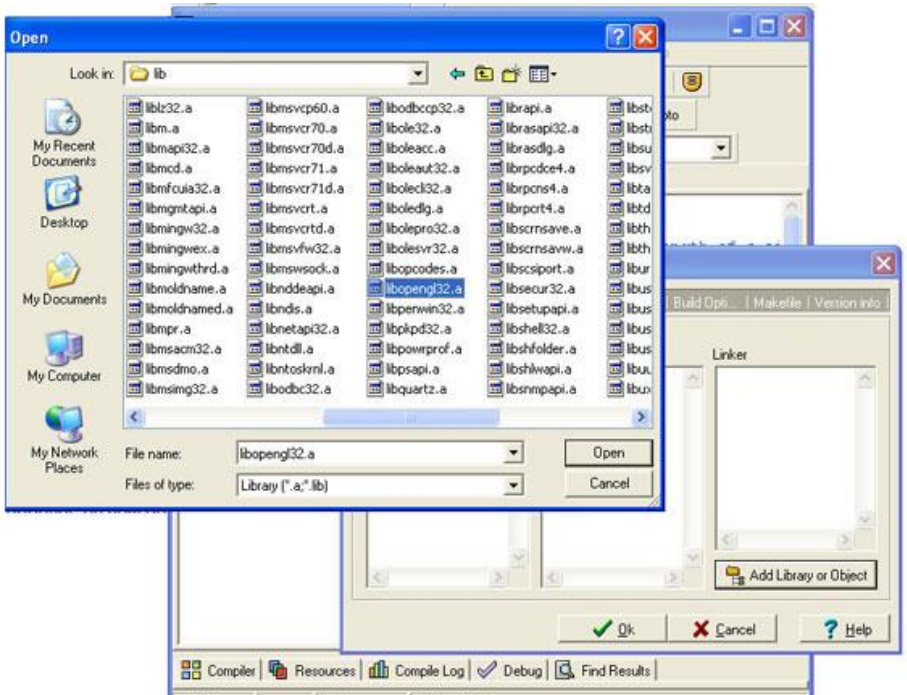
- Геометрические (точки, линии, полигоны) и растровые (массив bitmap и образ image) примитивы.
- Использование B-сплайнов (для рисования кривых по опорным точкам).
- Видовые и модельные преобразования (расположение обьектов в пространстве, вращение, изменение формы, изменение положения камеры).
- Работа с цветом.
- Удаление невидимых линий и поверхностей. Z-буферизация.
- Двойная буферизация. Изображение каждого кадра сначала рисуется во втором(невидимом) буфере, а потом, когда кадр полностью нарисован, весь буфер отображается на экране.
- Наложение текстуры.
- Сглаживание (изменяет интенсивность и цвет пикселей около линии, при этом линия смотрится на экране без всяких зигзагов).
- Освещение (источники света, расположение, интенсивность).
- Атмосферные эффекты (туман, дым).
- Прозрачность объектов.
- Использование списков изображений.

Doom 3

Установка (DevCpp)

1. glut.h → D:\Dev-Cpp\include\GL
2. libglut32.a → D:\Dev-Cpp\lib
3. glut32.dll → C:\Windows\System32
4. Создать пустой проект
5. Добавить библиотеки линкеру в свойствах проекта:
   /lib/libopengl32.a /lib/libglu32.a /lib/libglut32.a





Минимальная программа

main00.c

1. Инициализация GLUT

   void glutInit(int *argcp, char **argv); //обычно из main

2. Установка параметров окна.
3. Создание окна.

   int glutCreateWindow(const char *title);

4. Установка функций, отвечающих за рисование в окне и изменении формы окна.
5. Вход в главный цикл GLUT.

Установка параметров окна

- **void glutInitWindowSize(int width, int height);**

  размер окна
- **void glutInitWindowPosition(int x, int y);**

  положение относительно верхнего левого угла
- **void glutInitDisplayMode(unsigned int mode);**

  режим отображения информации: используемая цветовая модель, количество различных буферов, и т.д.

  void glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);

Установка параметров окна

Константа	Значение
GLUT_RGB	Для отображения графической информации используются 3 компоненты цвета RGB.
GLUT_RGBA	То же что и RGB, но используется также 4 компонента ALPHA (прозрачность).
GLUT_INDEX	Цвет задается не с помощью RGB компонентов, а с помощью палитры. Используется для старых дисплеев, где количество цветов например 256.
GLUT_SINGLE	Вывод в окно осуществляется с использованием 1 буфера. Обычно используется для статического вывода информации.
GLUT_DOUBLE	Вывод в окно осуществляется с использованием 2 буферов. Применяется для анимации, чтобы исключить эффект мерцания.
GLUT_ACCUM	Использовать также буфер накопления (Accumulation Buffer). Этот буфер применяется для создания специальных эффектов, например отражения и тени.
GLUT_ALPHA	Использовать буфер ALPHA. Этот буфер, как уже говорилось используется для задания 4-го компонента цвета - ALPHA. Обычно применяется для таких эффектов как прозрачность объектов и антиалиасинг.
GLUT_DEPTH	Создать буфер глубины. Этот буфер используется для отсечения невидимых линий в 3D пространстве при выводе на плоский экран монитора.
GLUT_STENCIL	Буфер трафарета используется для таких эффектов как вырезание части фигуры, делая этот кусок прозрачным. Например, наложив прямоугольный трафарет на стену дома, вы получите окно, через которое можно увидеть что находится внутри дома.
GLUT_STEREO	Этот флаг используется для создания стереоизображений. Используется редко, так как для просмотра такого изображения нужна специальная аппаратура.

Установка функций рисования и изменения формы окна

- **void glutDisplayFunc(void (*func)(void));**

  Перерисовка окна (первый запуск, разворот)

  void Draw();

  glutDisplayFunc(Draw);

- **void glutReshapeFunc(void (*func)(int width,int height));**

  Изменение размеров окна.

  void Reshape(int w, int h);

  glutReshapeFunc(reshape);

Установка функций рисования и изменения формы окна

```
void reshape(int w, int h)
{
    glViewport(0, 0, w, h); //область вывода изображения
                            //размером со все окно
    glMatrixMode(GL_PROJECTION); //загрузим матрицу проекции
        //отвечает за добавление в нашу сцену перспективного вида
    glLoadIdentity(); //заменим ее единичной (сброс)
    gluOrtho2D(0, w, 0, h); //и установим ортогональную проекцию

    glMatrixMode(GL_MODELVIEW); //загрузим модельно-видовую матрицу
        //"место", где сохранена информация о наших объектах
    glLoadIdentity(); //и заменим ее единичной
}
```

Вход в главный цикл GLUT

void glutMainLoop(void);

Сердце GLUT:

взаимосвязь между операционной системой и теми функциями, которые отвечают за окно, получают информацию от устройств ввода/вывода

Примитивы

glBegin (тип_примитива)…glEnd ()

- **GL_POINTS** — каждая вершина задает точку
- **GL_LINES** — каждая отдельная пара вершин задает линию
- **GL_LINE_STRIP** — каждая пара вершин задает линию (т.е. конец предыдущей линии является началом следующей)
- **GL_LINE_LOOP** — аналогично предыдущему за исключением того, что последняя вершина соединяется с первой и получается замкнутая фигура
- **GL_TRIANGLES** — каждая отдельная тройка вершин задает треугольник
- **GL_TRIANGLE_STRIP** — каждая следующая вершина задает треугольник вместе с двумя предыдущими (получается *лента* из треугольников)
- **GL_TRIANGLE_FAN** — каждый треугольник задается первой вершиной и последующими парами (т.е. треугольники строятся вокруг первой вершины, образуя нечто похожее на диафрагму)
- **GL_QUADS** — каждые четыре вершины образуют четырехугольник
- **GL_QUAD_STRIP** — каждая следующая пара вершин образует четырехугольник вместе с парой предыдущих
- **GL_POLYGON** — задает многоугольник с количеством углов равным количеству заданных вершин

Анимация

main01.c : функция таймера

void glutTimerFunc
(unsigned int millis, void (*func)(int value), int value);

- время таймера
- функция, вызываемая по истечении таймера
- идентификатор таймера

```
void timf(int value) // Timer function
{
	glutPostRedisplay(); // Redraw windows
	glutTimerFunc(40, timf, 0); // Setup next timer
}

glutTimerFunc(40, timf, 0); // Set up timer for 40ms, about 25 fps
```

Мышь

Функционал мыши добавляется аналогично

void glutMouseFunc

(void (*func) (int button, int state, int x, int y));

- функция, вызываемая при клике мышью; должна принимать три параметра:
  - номер кнопки мыши (0 – левая, 1 – средняя, 2 - правая)
  - состояние (0 – нажали, 1 – отпустили)
  - координаты

```
void MouseFunc(int button, int state, int x, int y)
{
    printf(" %d", button);
}

glutMouseFunc(MouseFunc);
```

Кливиатура

Функционал клавиатуры добавляется аналогично

void glutKeyboardFunc

(void (*func)(unsigned char key, int x, int y));

- функция, вызываемая при нажатии клавиши; должна принимать три параметра:
  - код нажатого символа
  - координаты мыши в момент нажатия

```
void KBFunc(unsigned char key, int x, int y)
{
    printf(" %d", key);
}
glutKeyboardFunc (KBFunc);
```

Задачи

0. Нарисовать круг.
1. Залить круг градиентом и заставить вращаться.
2. По клику мышью рисовать на месте клика небольшой синий квадрат.
3. Нарисовать снеговика.
4. Нарисовать график параболы.
5. «Индикатор с планеты Блук». При клике на квадрат он перелетает на произвольный участок экрана и меняет цвет. После десяти кликов меняет цвет на черный и на клики не реагирует.
6. «Два нейтрона». Два шарика одинакового радиуса и разного цвета расположены в разных углах экрана. С клавиатуры задаются четыре числа: скорости и углы вылета шариков. Шарики упруго отскакивают от стен и друг от друга. Рисовать до клика мышью.