Interaction

321190
2007년 봄학기
3/23/2007
박경신

Objectives

- 디스플레이 리스트 (Display Lists)
  - 유지모드 그래픽스 (Retained mode graphics) 방식
- 지적 (Picking)
  - 화면상의 한 객체를 사용자가 식별 할 수 있도록 하는 입력연산

Display Lists

- 그래픽스 효율을 증가시키기 위해서 네트워크 상의 클라이언트와 서버 구조를 가진 디스플레이 프로세서라 불리는 특수 목적 컴퓨터를 구성
- 초창기 그래픽스 시스템의 구조
  - 디지털 아날로그 변환기를 통해 CRT에 연결된 Host에 기반을 두고 있음
- 디스플레이 프로세서 (Display processor) 구조
  - 래스터화(Rasterization)는 디스플레이 프로세서에서 이루어짐.
  - DPU(Display processor unit)는 제한된 명령어 집합 (instruction set)을 가지고 있음. 대부분이 CRT에 기본 요소를 그리기 위한 것.
  - 사용자 프로그램은 호스트 컴퓨터에서 수행되어 컴파일된 명령 리스트가 되어 디스플레이 프로세서로 보내지고, 그 명령어들은 디스플레이 메모리(display memory)에 디스플레이 파일 (display file) 또는 디스플레이 리스트 (display list)로 저장됨.

OpenGL Display List

- OpenGL에서 디스플레이 리스트는 디스플레이할 때 사용하기 위해 저장된 OpenGL 명령어의 집합.
  - 자주 실행될 때 유용
  - 수행 효율의 향상을 위해 설계됨
- 디스플레이 리스트 정의
  - glNewList(list, mode)
    - *list* 는 디스플레이 리스트의 식별자 (ID)
    - *mode* 는 GL_COMPILE (서버에 리스트를 보내지만 내용을 디스플레이 하지 않도록 지시), GL_COMPILE_AND_EXECUTE (즉시 디스플레이)
  - glEndList()
- 디스플레이 리스트 실행
  - glCallList(list)
- 디스플레이 리스트 이름 지정
  - glGenList(range)

OpenGL Display List

```
#define BOX 1
glNewList(BOX, GL_COMPILE);
    glBegin(GL_POLYGON);
        glColor3f(1.0, 0.0, 0.0);
        glVertex2f(-1.0, -1.0);
        glVertex2f(1.0, -1.0);
        glVertex2f(1.0, 1.0);
        glVertex2f(-1.0, 1.0);
    glEnd();
glEndList();


glCallList(BOX);
```

Display List Functions

- 디스플레이 리스트 생성

```
GLuint id;
void init()
{
  id = glGenLists( 1 );

  glNewList( id, GL_COMPILE );

  /* other OpenGL routines */

  glEndList();
}
```

- 생성된 디스플레이 리스트 호출

```
void display()
{
  glCallList( id );
}
```

Display List

- 여러 개의 디스플레이 리스트를 부르는 함수
  - glCallLists(GLsizei num, GLenum type, const GLvoid * lists_ptr)
  - Type은 GL_BYTE, GL_UNSIGNED_BYTE, GL_SHORT, GL_UNSIGNED_SHORT, GL_INT, GL_UNSIGNED_INT, GL_FLOAT, GL_2_BYTES, GL_3_BYTES, GL_4_BYTES
- 디스플레이 리스트 안에서 다른 디스플레이 리스트도 호출 가능
- 디스플레이 리스트 생성은 네스트(nested)하지 못함
  - 예를 들어 glEndList() 전에 glNewList()가 두번째로 불리면 GL_INVALID_OPERATION라는 에러(error) 발생
- glFlush(), glGet*(), glIsEnabled() 처럼 즉시처리를 요하는 함수는 디스플레이 리스트 안에 불릴 수 없음
- 디스플레이 리스트는 한번 생성되면 glDeletelists() 가 불릴 때까지 내용을 바꿀 수 없음.

Display Lists and State

- 디스플레이 리스트 안에 대부분의 OpenGL 함수들이 들어갈 수 있음
- 디스플레이 리스트 안에서 상태(State)를 변화시킬 수 있고, 이런 변경이 디스플레이 리스트가 실행된 후에도 예측하지 못한 효과를 줄 수도 있음.
  - 예를 들어, 빨간색 네모를 그리는 디스플레이 리스트가 실행될 때마다 그림의 색이 적색으로 설정되어 프로그램에서 이후에 정의된 기보 요소들 또한 적색을 가지게 됨.
- 따라서, OpenGL의 행렬과 속성 스택의 사용을 권장함.
  - 디스플레이 리스트안에 시작과 끝에 행렬과 속성 스택 사용

```
glPushAttrib(GL_ALL_ATTRIB_BITS);
glPushMatrix();
…
glPopMatrix();
glPopAttrib();
```

Display Lists and State

- 디스플레이 리스트안에 시작과 끝에 행렬과 속성 스택 사용

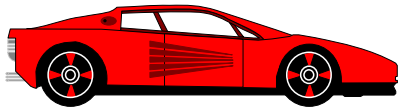
```
#define RED_SQUARE

glNewList(RED_SQUARE, GL_COMPILE);
    glPushAttrib(GL_CURRENT_BIT);
    glColor3f(1, 0, 0);
    glRectf(-1, -1, 1, 1);
    glPopAttrib();
glEndList();
```

Stack: LIFO

Hierarchy and Display Lists

- 자동차 모델 예제
  - 섀시 (chassis) 디스플레이 리스트 생성
  - 바퀴 (wheel) 디스플레이 리스트 생성

```
glNewList(CAR, GL_COMPILE);
   glCallList(CHASSIS);
   glTranslatef(…);
   glCallList(WHEEL);
   glTranslatef(…);
   glCallList(WHEEL);
…
glEndList();
```



Picking

- 지적(Picking)은 화면 상의 한 객체를 사용자가 식별할 수 있도록 하는 입력연산
- OpenGL에서 지적 연산을 구현하기 어려움.
  - 객체 (object) 정의
    - 디스플레이 리스트 사용
    - 기본요소의 그룹을 표시하는 태그 시스템을 생성
  - 객체 지적(picking an object)을 정의
    - 기본요소를 마우스로 클릭하나? 아니면 가까이 가는 것으로 하나?
    - 여러 객체가 중첩(overlap) 됐다면?
  - 구현 이슈

Picking Solutions

- 선택모드 (Selection mode)
  - 클리핑 영역과 시역을 조절하여 작은 클리핑 영역 내의 어떤 기본요소(primitive)가 커서(cursor)부근의 영역으로 렌더링되는지 추적. 적중 리스트(hit list)에 기록되어 후에 사용자 프로그램으로 조사될 수 있음.
- 경계 박스 (Bounding box)
  - 가장 간단한 방법
- 후면 버퍼(Back buffer)
  - color-coded objects 과 glReadPixel() 사용

Rendering Modes

- OpenGL의 glRenderMode(mode)에서 선택된 3가지 모드 중 하나로 렌더링.
  - 렌더모드 (GL_RENDER): 프레임 버퍼에 정상적인 렌더링 (default)
  - 피드백모드 (GL_FEEDBACK): 렌더링된 기본 요소의 리스트를 얻는데 사용
  - 선택모드 (GL_SELECTION): 선택모드로 전환하고 장면을 렌더링하면, 뷰볼륨 안에 있는 각 기본 요소는 적중 레코드 (hit record)를 발생하여 추후에 사용하도록 네임스택(name stack)이라고 하는 버퍼 안에 저장됨.

Selection Mode Functions

- void glSelectBuffer(GLsizei n, GLunint *buffer)
  - 네임스택 지정. 선택된 데이터가 배열 안에 있는지 확인가능
  - 선택 데이터를 넣을 배열 buffer의 크기를 n으로 설정
- void glInitNames()
  - 네임스택을 초기화
- void glPushName(GLuint name)
  - name을 네임스택에 넣는 함수
- void glPopName()
  - 탑(top)에 있는 name을 네임스택으로부터 꺼내는 함수
- void glLoadName(GLuint name)
  - 네임스택의 탑을 name으로 대체하는 함수
- name은 객체를 식별하기 위해 응용 프로그램에서 지정함

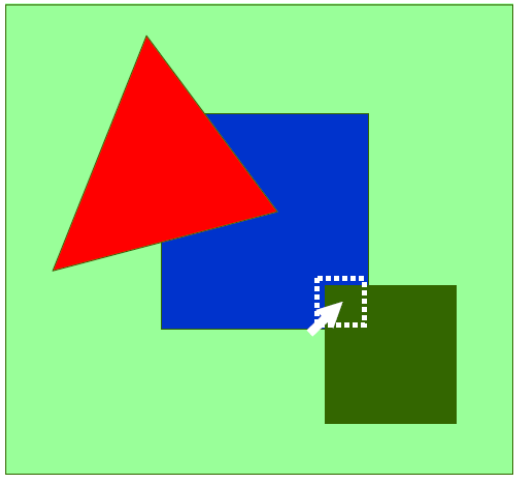
Using Selection Mode

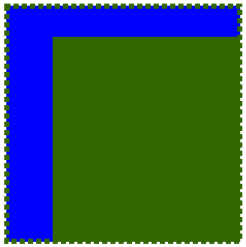
1. 네임스택을 초기화
2. 선택모드로 전환
   - 일반적으로 선택 모드에 들어가기 위해서는 마우스 답신을 사용하고 마우스 답신이 끝나기 전에 선택모드를 떠남.
3. 사용자가 정의한 ID로 장면을 렌더링
4. 다시 정상적인 렌더모드로 전환
   - 이 과정에서 다수의 히트(number of hits)를 넘겨받음.
5. 네임스택 (name buffer)에 히트 레코드(hit records) 조사
   - 히트 레코드는 ID와 깊이 정보를 포함

Selection Mode and Picking

- 사용자가 정의한 커서에 중심을 갖는 조그만 사각형 안에 렌더링 된 모든 객체를 지적한다고 가정. 사각형의 크기가 지적 감도의 척도가 됨. 투영행렬과 지적행렬을 곱하기 위해서 gluPickMatrix를 사용.
  - void gluPickMatrix(GLdouble x, GLdouble y, GLdouble witdth, GLdouble height, GLint viewport[4])

Selection Mode





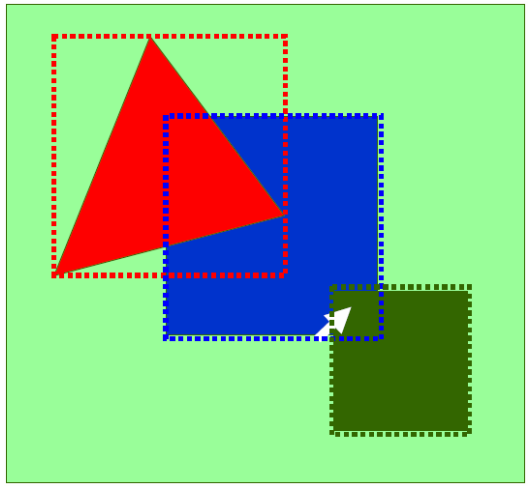
Hit list

Object id = 3
Min, Max depths
Object id = 2
Min, Max depths

Using Bounding Box

- 객체의 범위 (the extent of an object)을 정의.
  - 객체의 범위는 객체를 포함하고 있는 좌표축에 정렬된 (axis-aligned) 가장 작은 사각형
- 세계 좌표계 안에 어떤 사각형이 화면 좌표계의 특정 점과 대응하는지를 결정하는 것은 비교적 쉬운 일.
- 객체들과 경계 사각형들을 관련짓는 간단한 자료구조를 응용 프로그램이 유지한다면 근사한 지적을 구현가능.

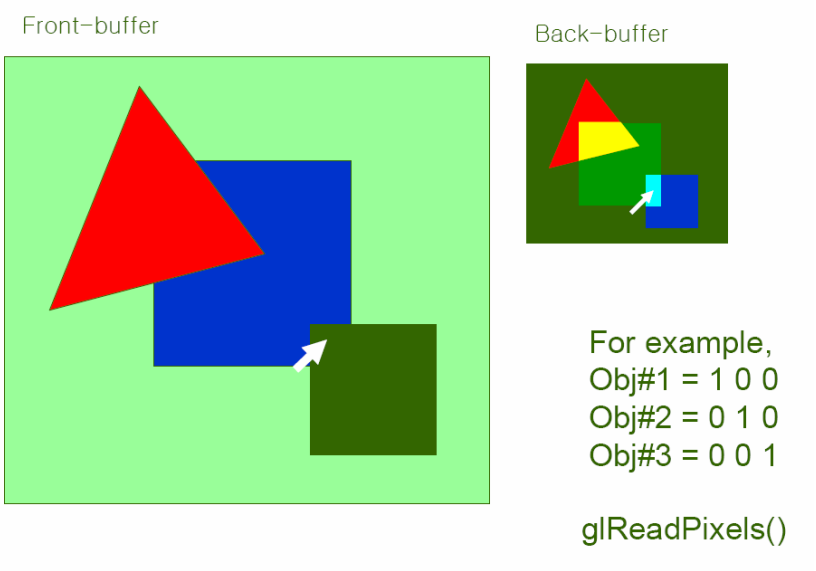
Bounding Box



Using Back buffer

- 객체들(objects)는 각각 다른 색으로 후면 버퍼에 렌더링.
- 응용프로그래머는 새로운 객체가 프로그램으로 정의될 때마다 단순히 색을 변화시켜 객체의 내용을 결정
- 후면버퍼를 사용한 지적 연산 단계:
  1. 객체를 지적 색(pick color)로 후면버퍼에 그림
  2. 마우스 답신함수를 이용하여 마우스의 위치 값(mouse position)을 얻음
  3. glReadPixels() 함수를 이용하여 마우스 위치에 해당하는 프레임 버퍼 안의 위치를 찾고 이 위치의 색을 구함
  4. 읽혀진 색에 해당하는 객체를 색 테이블을 이용하여 찾음

Using Back buffer



References

- http://www.lighthouse3d.com/opengl/picking
  OpenGL Picking Tutorial

Announcement

- 4/2 (월) 밤 10시까지 실습과제-1을 e-Learning 사이버강의실에 제출할 것.