OpenGL 2

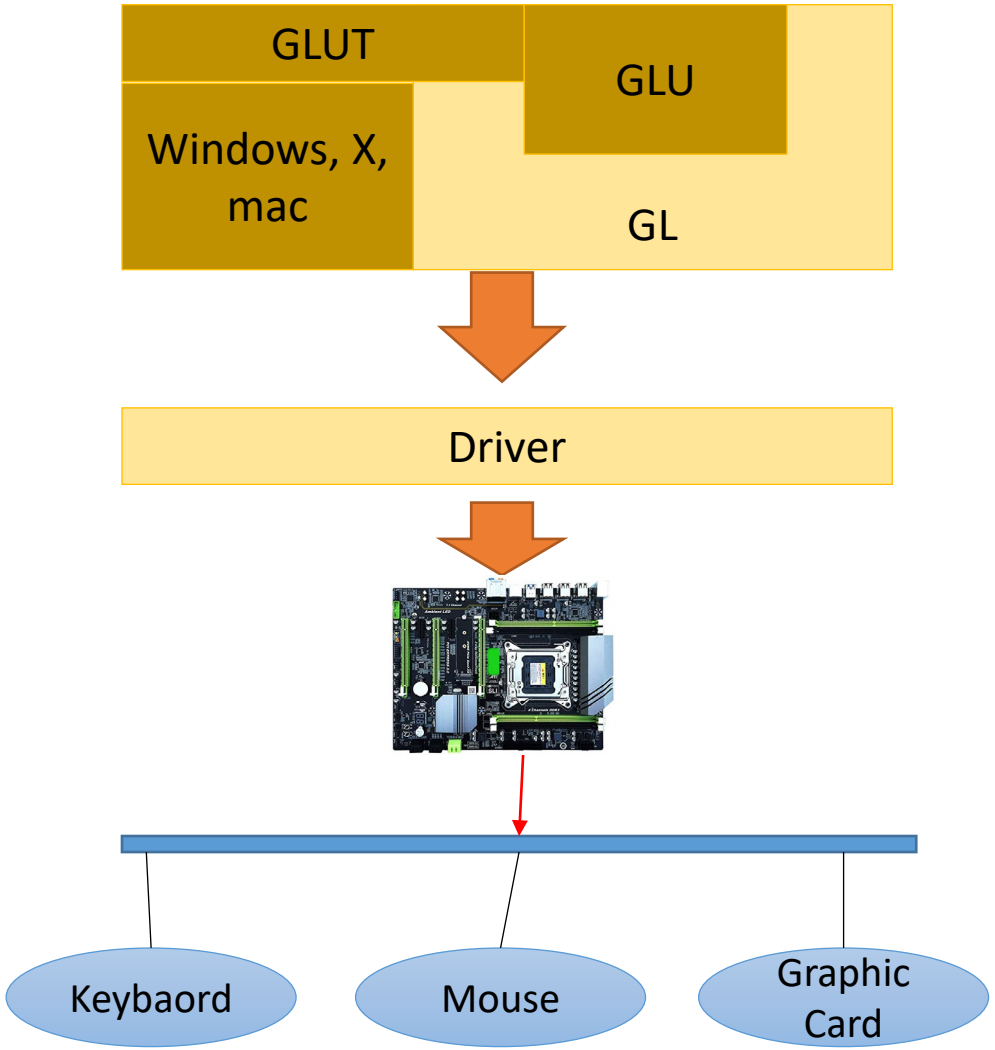
2021년 봄학기 가상현실론

2021/03/08

목차

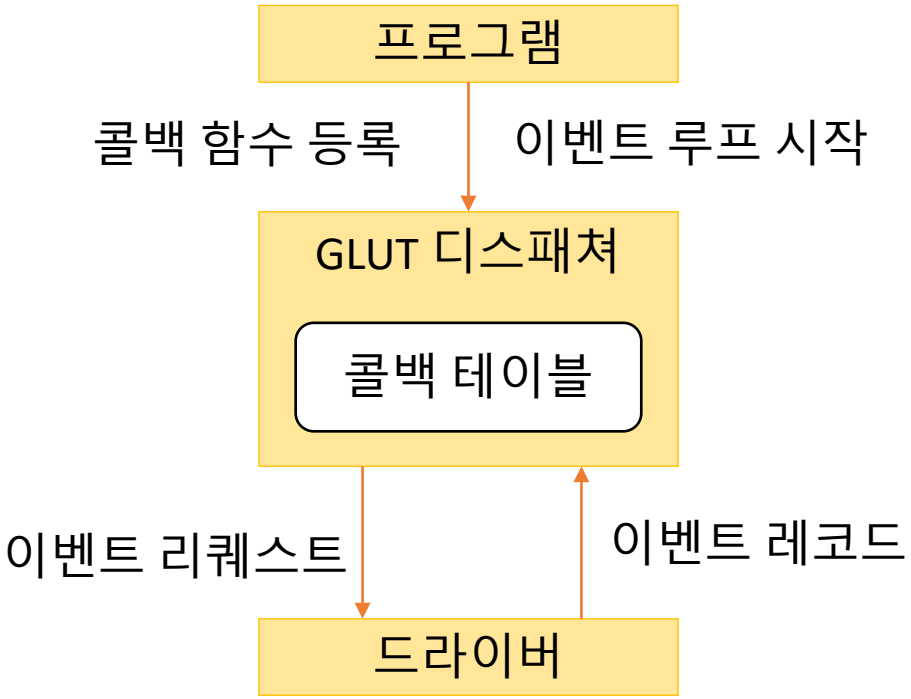
- 복습
- 윈도우와 뷰포트(Viewport)
- 콜백 프로그래밍(Callback programming)

복습



- GLProgram components
  - GL: OpenGL Core Liabrary
    렌더링 기능을 제공하는 함수 라이브러리
  - GLU: OpenGL Utility Library
    약 50개의 함수로 구성되며 사실상 GL 라이브러리의 도우미 역할을 함. 다각형 분할 (Tesselllation), 투상 (Projection), 2차원 곡면 (Quadratic Surface), NURBS 등 고급기능을 제공. GLU 함수는 결국 GL 함수 호출로 변환.
  - GLUT: OpenGL Utility Toolkit
    사용자 입력을 받아들이거나 화면 윈도우를 제어하기 위한 함수로, 윈도우 운영체제가 실행하는 기능들

복습



- 입력 callback과 GLUT
  - Glut는 GL 프로그램과 드라이버 사이에서 인터페이스 역할을 한다. (프로그래머가 직접 핸들링 하지 않도록)

이벤트 타입	콜백함수 (ex)
DISPLAY	display()
RESHAPE	reshape()
KEYBOARD	keyboard()
MOUSE	mouse()
IDLE	idle()

GLUT 프로그램 기본구조

```
void display() {};
void keyboard() {};
void mouse(int button, int state, int x, int y ) {};
void myInit() {};
int main(int argc, char** argv) {
      glutInit(&argc, argv);
      glutInitDisplayMode(GLUT_RGB);
      myInit();
      glutCreateWindow("OpenGL Basic structure");
      glutDisplayFunc(display);
      glutKeyboardFunc(keyboard);
      glutMouseFunc(mouse);
      glutMainLoop();
      return 0;
}
```

디스플레이 콜백함수 정의

키보드 콜백함수 정의

마우스 콜백함수 정의

윈도우 초기화 및 설정

GL 상태 변수 설정

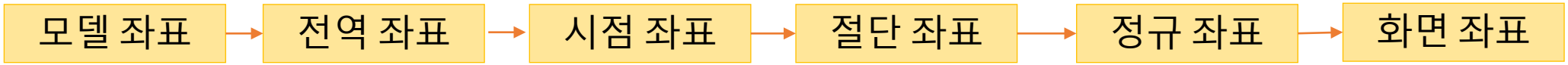
Callback 함수 등록

이벤트 루프 진입

1. Glut 함수를 사용하여 윈도우 관련 상태 변수 값을 설정
2. 배경화면의 색, 광원의 위치 등 지엘의 상태 변수 중 전체 프로그램을 통해 변하지 않는 상태 변수 설정. myInit() 등 함수 사용.
3. 콜백함수 등록
4. 이벤트 루프 진입

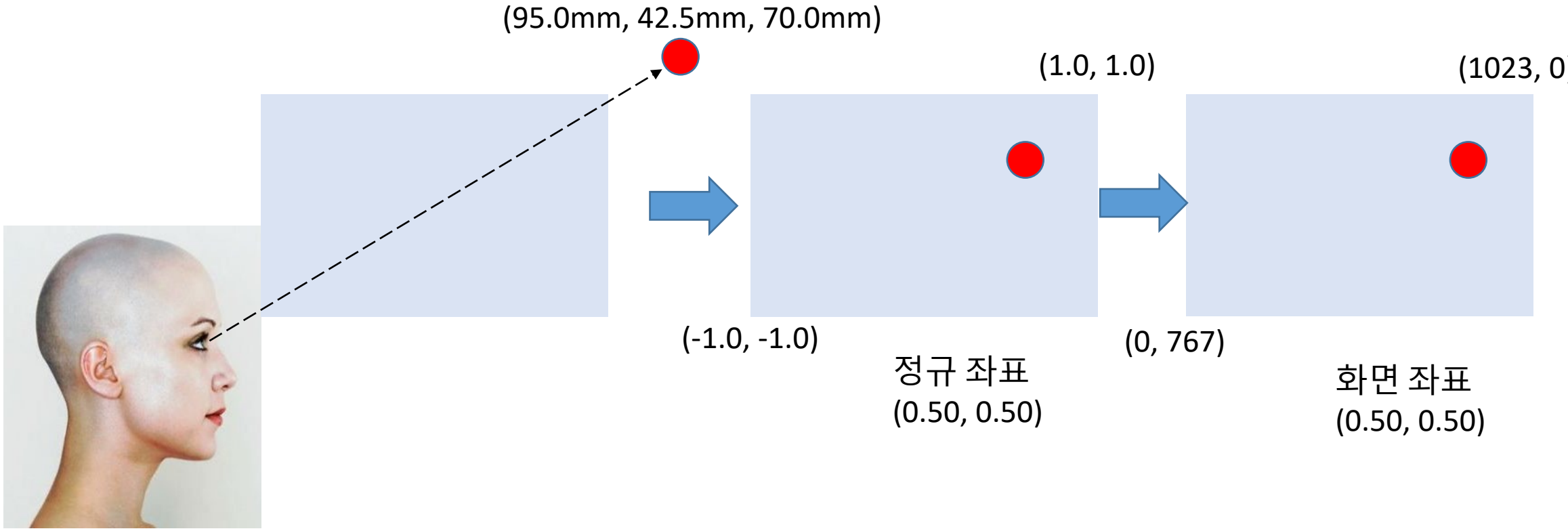
윈도우와 뷰포트(Viewport)

정규좌표와 화면 좌표

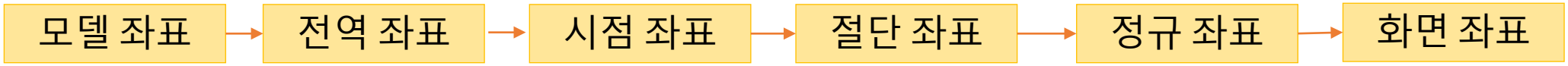


- 모델 좌표 (local or model coordinate)
  - 모델, 물체별로 설정된 좌표계
- 전역 좌표 (global coordinate)
  - 모델을 포괄하는 가상 세계의 전역 좌표계
- 시점 좌표
  - 물체를 바라보는 시점을 기준으로 표현한 것
- 절단 좌표
  - 시점으로부터 보이지 않는 물체를 잘라내기 편하게 설정한 것
- 정규 좌표 (NDC: Normalized Device Coordinate)
  - 1을 기준으로 하는 (normalized된) 2차원 좌표

정규좌표와 화면좌표


(95.0mm, 42.5mm, 70.0mm)
(1.0, 1.0)
(1023, 0)
(-1.0, -1.0)
(0, 767)
정규 좌표
(0.50, 0.50)
화면 좌표
(0.50, 0.50)

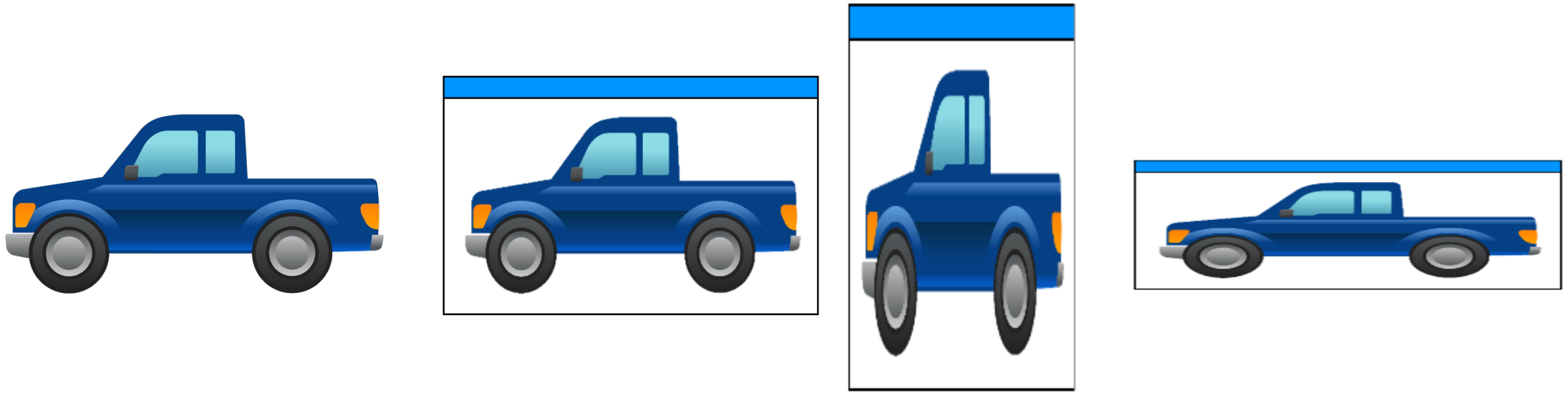
정규좌표와 화면 좌표



- 모델 좌표 (local or model coordinate)
  - 모델, 물체별로 설정된 좌표계
- 전역 좌표 (global coordinate)
  - 모델을 포괄하는 가상 세계의 전역 좌표계

윈도우, 뷰포트, GLUT 모델링

- 뷰 포트 (viewport)
  - 윈도우 내부에 디스플레이를 위해 설정한 창
  - 프로그래머가 별도로 뷰 포트를 설정하지 않으면 묵시적으로 현재 윈도우 전체가 하나의 뷰 포트로 간주. 이 경우 왜곡(distortion)이 일어날 수 있음 (예: 400 x 300 사이즈의 윈도우는 가로가 길게 표현).



예제

```
#include <GL/glut.h>

void display();
void myInit();

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_RGB);
   glutInitWindowSize(300, 300);
   glutInitWindowPosition(0, 0);
   glutCreateWindow("openGL ex2-viewport");
   glutDisplayFunc(display);
   glutKeyboardFunc(keyboard);
   myInit();
   glutMainLoop();
   return 0;
}
```

1) GLUT 윈도우 함수

```
void myInit()
{
   glClearColor(0.0, 0.0, 0.0, 1.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
}

void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glViewport(0, 0, 150, 150);
   glColor3f(1.0, 1.0, 1.0);
   glBegin(GL_POLYGON);
      glVertex3f(-0.5, -0.5, 0.0);
      glVertex3f(0.5, -0.5, 0.0);
      glVertex3f(0.5, 0.5, 0.0);
      glVertex3f(-0.5, 0.5, 0.0);
   glEnd();
   glFlush();
}
```

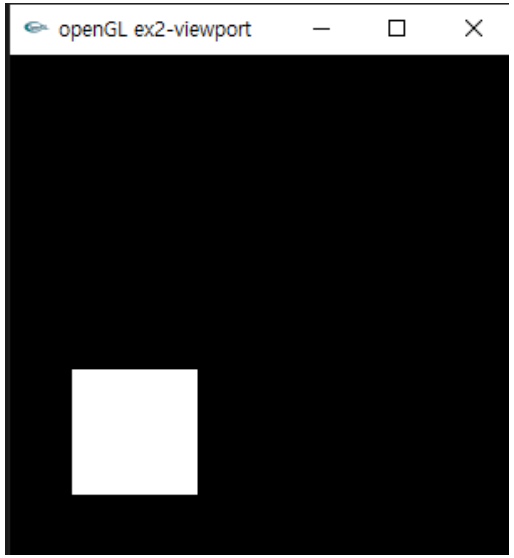
2) GL 상태 변수 설정

3) GL 상태 변수 설정

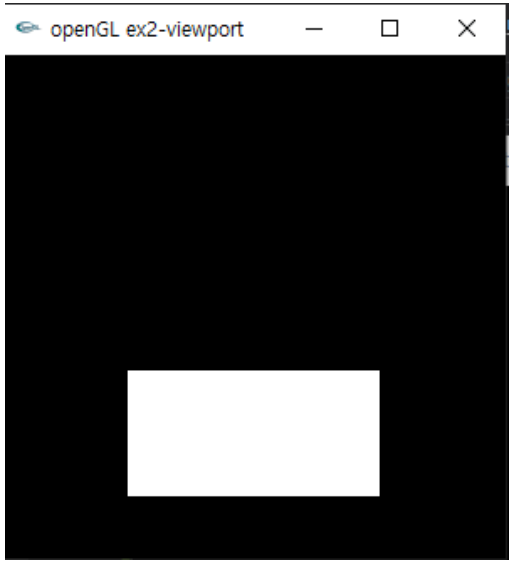
4) 입력 기본요소 정의

예제

- glutInit(): GLUT 라이브러리를 초기화하고 윈도우 운영체제와 연결하는 세션 형성
- glutInitDisplaymode(GLUT_RGB): 윈도우의 기본 컬러모드를 RGB 모드르 설정
- glutWindowSize(300, 300): 윈도우 폭 300, 높이 300
- glClear(GL_COLOR_BUFFER): 컬러버퍼, 즉 프레임 버퍼를 초기화. 초기화에 사용될 색은 초기에 glClearColor()에서 설정한 색 RGB(0, 0, 0) & alpha(0). Alpha가 1.0이면 100% 불투명.



glViewport(0, 0, 150, 150)



glViewport(0, 0, 300, 150)

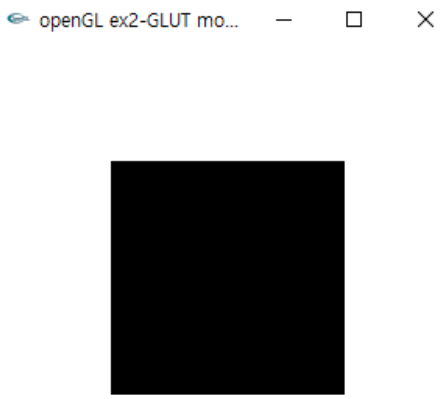
윈도우 크기가 (300, 150)이고 별도로 뷰포트를 설정하지 않은 경우 -> 왜곡

-> 사용자의 윈도우 크기 조절 행위에 동적으로 대응하는 콜백함수 필요

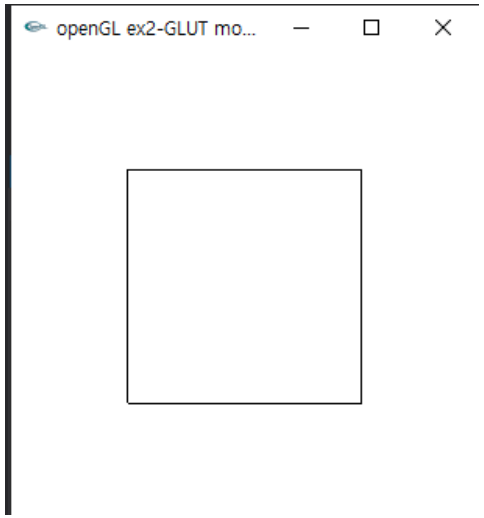
GLUT 모델링

- GLUT은 이미 모델링된 몇 가지 물체를 제공. 정육면체(Cube), 원구(Sphere), 원환체(Torus), 원뿔(Cone), 정사면체(Tetrahedron), 정팔면체(Octahedron), 정12명체(Dodecahedron), 정20면체(Icosahedron), 차 주전자(Teapot) 등.
- 이를 그리려면 display callback function에서 해당 모델에 해당하는 명령을 호출하면 됨.
- 정육면체
  - void glutSolidCube(GLdouble size);
  - void glutWireCube(GLdouble size);
  - glutSolidCube()는 물체 겉면이 칠해진 형태(Solid rendering)로, glutWireCube()는 물체 뼈대만 선으로 표시한 형태(wireframe rendering)으로 그려냄. 여기서 size는 정육면체 한 변의 길이
- 원구
  - void glutSolidSphere(GLdouble radius, GLint slices, GLint stacks);
  - void glutWireSphere(GLdouble radius, GLint slices, GLint stacks);
  - Radius는 원구의 반지름, slices는 경선(longitudinal lines)의 수, stacks는 위선(latitudinal lines)의 수

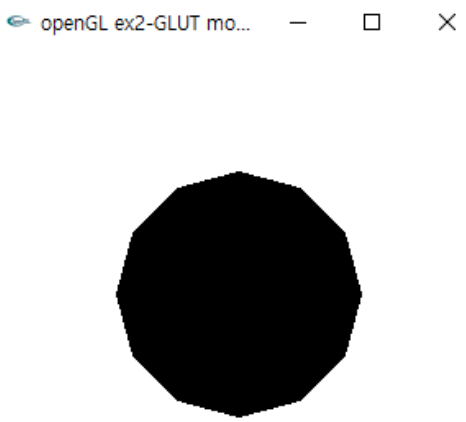
Examples



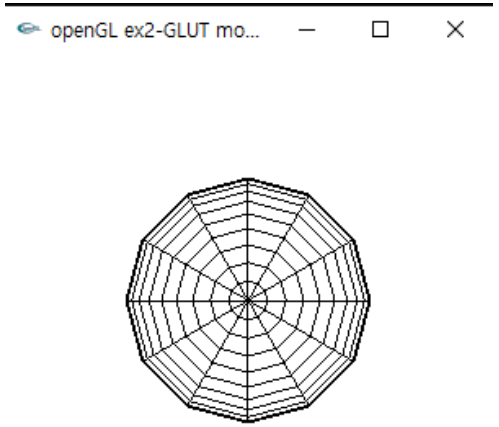
glutSolidCube(1.0);



glutWireCube(1.0);



glutSolidSphere(0.5, 12, 20);

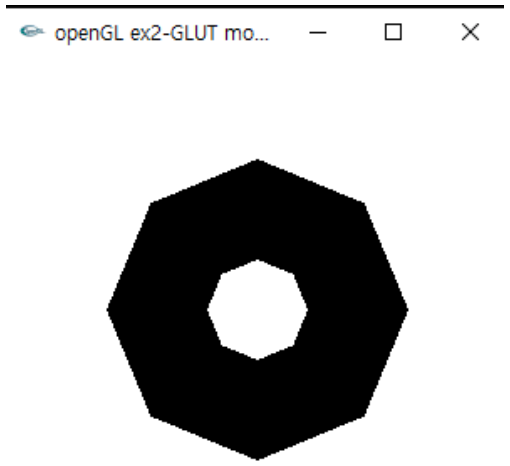


glutWireSphere(0.5, 12, 20);

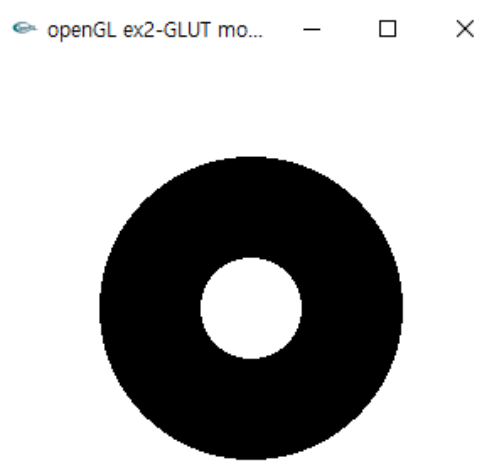
GLUT 모델링

- 원환체
  - void glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nsides, GLint rings);
  - void glutWireTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nsides, GLint rings);
  - innerRadius, outerRadius는 원환체의 안쪽과 바깥쪽 반지름. nsides는 튜브 단면을 몇 개의 선분으로 근사화할 것인지 나타내며 rings는 튜브 윤관을 몇 개의 선분으로 근사화하는지를 나타냄
- 원뿔
  - void glutSolidCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
  - void glutWireCone(GLdouble base, GLdouble height, GLint slices, GLint stacks);
  - Base는 원뿔 밑면의 반지름, 높이는 원뿔의 높이, slices는 z축 둘레를 몇 개의 선분으로 근사화할 것인지, stacks는 z축을 따라서 몇 개의 면으로 근사화할 것인지 나타냄.
- 차주전자
  - void glutSolidTeapot(GLdouble size);
  - void glutWireTeapot(GLdouble size);
  - size는 주전자의 상대적 크기

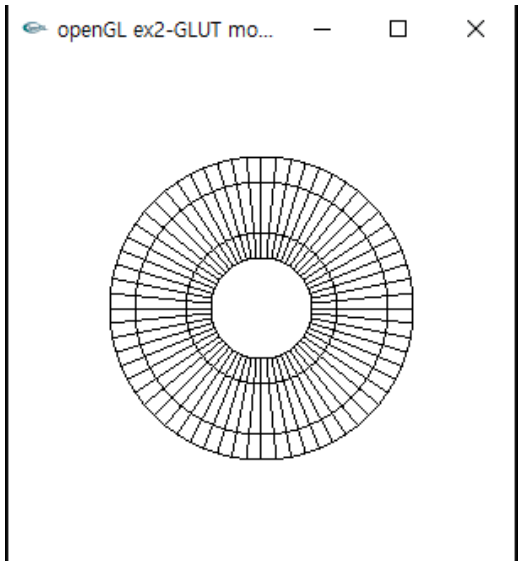
Examples



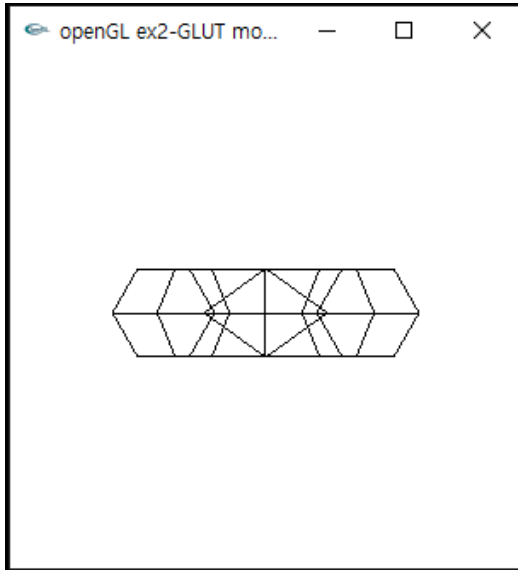
glutSolidTorus(0.2, 0.4, 6, 8);



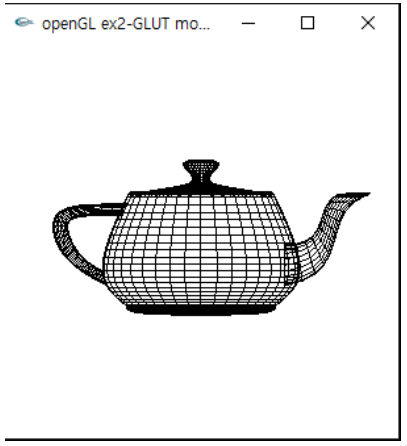
glutSolidTorus(0.2, 0.4, 6, 64);



glutWireTorus(0.2, 0.4, 6, 64);



glutWireTorus(0.2, 0.4, 6, 8);



glutWireTeapot(0.5);

콜백 프로그래밍 (Callback Programming)

리셰이프 콜백 (reshape callback)

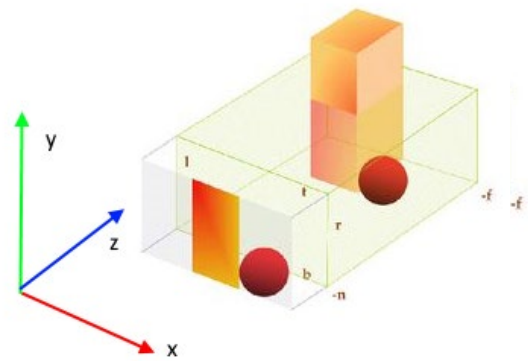
- GLUT는 아래의 경우에 리셰이프 이벤트 (reshape event)가 발생한 것으로 취급한다.
  - 처음 윈도우를 열 때
  - 윈도우 위치를 옮길 때
  - 윈도우 크기를 조절할 때
- 리셰이프 이벤트를 등록하기 위한 콜백함수 프로토타입
  - void glutReshapeFunc(void(*func)(int width, int height));
  - func에 reshape 콜백 함수 이름을 넣으면 됨. 예를 들어 reshape function을 myReshape이라고 한다면 glutReshapeFunc(myReshape). myReshape은 프로토타입에 명시된대로 myReshape(int width, int height)의 형태로 선언되어야 하며 width와 height는 변형된 윈도우의 폭과 높이를 나타낸다.

리셰이프 콜백 (reshape callback)

- gluOrtho()
  - 윈도우 크기 조절에 따른 왜곡을 방지하기 위해 사용할 수 있다.
  - 예)

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
glOrtho(-1.0, 1.0, -1.0, 1.0, -10, 1.0);
```

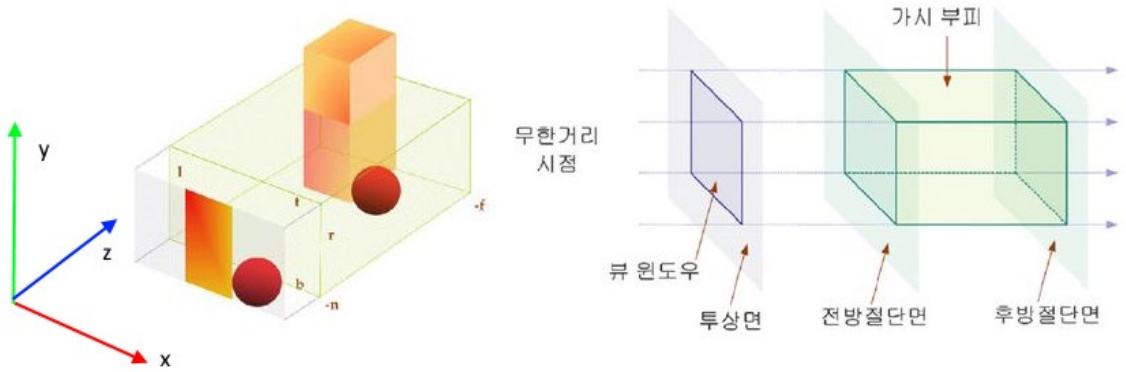
  - glMatrixMode()는 투상행렬(GL_PROJECTION)을 변환대상으로 설정하라는 명령이며, glLoadIdentity()는 투상 행렬에 identity matrix를 로드하라는 명령임.



- 좌측의 그림은 3차원 물체 영상이 관찰자 앞에 놓인 2차원 화면에 맺히는 모습임
- 물체 영상이 투상(Projection)되는 평면이라는 의미에서 해당 면을 투상면(projection plane)이라고 함.

리셰이프 콜백 (reshape callback)

- glOrtho()
  - glOrtho()는 평행 투상의 일종으로 구체적으로는 그림처럼 직육면체 형태의 가시 부피(view volume)을 설정함으로써 정의.
  - 가시부피는 '화면에 보이고자 하는 물체의 범위 ' 를 말함. 따라서 지정된 가시 부피 밖의 물체는 화면에 보이지 않음 (그림의 주황색 큐브의 윗부분은 가시부피 밖에 있기 때문에 보이지 않음).



- 가시부피는 6개의 면으로 정의됨.
- 투상면에 나란하면서 관찰자에 가까운 면을 전방 절단면(front clipping plane, front plane, near plane, hither), 먼 면을 후방 절단면(back clipping plane, back plane, far plane, yon)이라고 함.

리셰이프 콜백 (reshape callback)

- glOrtho()
  - void glOrtho(GLdouble left, GLdouble right, GLdouble, bottom, GLdouble top, GLdbouel near, GLdouble, far);
  - 괄호 안의 파라미터는 순서대로 가시 부피의 좌 우 상 하 전 후 면의 위치를 나타냄. Left는 왼쪽 면의 x 좌표, rigth는 오른쪽 면의 x 좌표, top, botoom은 각각 윗면, 아랫면의 y 좌표를 나타냄. Near와 far는 각각 전방 절단면과 후방 절단면의 z 좌표를 나타냄. OpenGL에서는 전면의 z 값보다 후면의 z 값이 크다 (하지만 표현하는 좌표계는 시스템에 따라서 다르기 때문에 주의해야 함. 예를 들어 일반적으로 z 축은 사용자 쪽에 값이 더 큼)

예제)

```
#include <GL/glut.h>

void display();
void reshape(int width, int height);
void keyboard(unsigned char key, int x, int y);

int main(int argc, char** argv)
{
   glutInit(&argc, argv);
   glutInitDisplayMode(GLUT_RGB);
   glutInitWindowSize(300, 300);
   glutInitWindowPosition(0, 0);
   glutCreateWindow("OpenGL reshape ex");
   glClearColor(1.0, 1.0, 1.0, 1.0);
   glutReshapeFunc(reshape);
   glutDisplayFunc(display);
   glutKeyboardFunc(keyboard);
   glutMainLoop();
   return 0;
}
```

```
void display()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(0.5, 0.5, 0.5);
   glBegin(GL_POLYGON);
      glVertex3f(-0.5, -0.5, 0.0);
      glVertex3f(0.5, -0.5, 0.0);
      glVertex3f(0.5, 0.5, 0.0);
      glVertex3f(-0.5, 0.5, 0.0);
   glEnd();
   glFlush();
}

void reshape(int width, int height)
{
   glViewport(0, 0, width, height);
   GLfloat w_factor = (GLfloat)width / (GLfloat)300;
   GLfloat h_factor = (GLfloat)height / (GLfloat)300;
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glOrtho(-1.0*w_factor, 1.0*w_factor, -1.0*h_factor, 1.0*h_factor, -1.0,
1.0);
}
```

예제)

- Reshape 함수를 보면 윈도우 크기가 바뀔 경우 (혹은 프로그램이 시작할 경우) 우선 뷰 포트를 윈도우 크기에 맞춤
- W_factor와 h_factor는 윈도우 크기 변화에 따라서 뷰포트의 폭과 높이의 증가율을 계산함. 즉, 윈도우 크기가 늘어나면 투상면의 폭이나 높이를 늘리고 반대의 경우 줄임으로써 디스플레이되는 객체의 원래 크기를 유지할 수 있음.

디스플레이 콜백

- void glutDisplayFunc(void(*func)());
- glutDisplayFunc에 의해서 등록된 디스플레이 콜백 함수는 다음 다섯 가지의 경우에 한해 자동으로 호출됨
  - 처음 윈도우를 열 때
  - 윈도우 위치를 옮길 때
  - 윈도우 크기를 조절할 때
  - 앞 윈도우에 가려져 안 보이던 뒤 윈도우가 활성화되어 앞으로 드러날 때
  - glutPostRedisplay() 함수에 의해 이벤트 큐에 flag이 개시될 때
- 실시간 업데이트를 요하는 경우 이벤트를 기다리지 않고 아이들 함수 등으로 주기적으로 업데이트 함.

키보드 콜백

- 키보드 입력에 대해 호출되는 함수가 keyboard callback 함수.
- 키보드 콜백 함수를 등록하기 위한 프로토타입 함수
  - void glutKeyboardFunc(void(*func)(unsigned char key, int x, int y))
  - void glutSpecialFunc(void(*func)(int key, int x, int y));
  - glutKeyboardFunc는 문자 및 숫자 키에 대한 콜백 함수를 등록하기 위한 것. 등록된 콜백 함수는 keyboard(unsigned char key, int x, int y)의 형태로 선언되어야 함.
  - 키보드 이벤트가 발생했을 때 GLUT는 눌러진 키를 key 파라미터를 통해서 콜백 함수에 전달. 또한 키가 눌러진 순간 마우스의 위치는 GLUT의 화면 좌표계로 표시되어있는 파라미터 x, y에 의해 전달.
  - 방향 키, 특수 키에 대한 콜백 함수 등록은 glutSpecialFunc()의 프로토타입을 사용함.

키보트 콜백

- 특수키에 대한 GLUT 정의

함수 키	방향 및 이동 키
#define GLUT_KEY_F1 1	#define GLUT_KEY_LEFT 100
#define GLUT_KEY_F2 2	#define GLUT_KEY_UP 101
#define GLUT_KEY_F3 3	#define GLUT_KEY_RIGHT 102
#define GLUT_KEY_F4 4	#define GLUT_KEY_DOWN 103
#define GLUT_KEY_F5 5	#define GLUT_KEY_PAGE_UP 104
#define GLUT_KEY_F6 6	#define GLUT_KEY_PAGE_DOWN 105
#define GLUT_KEY_F7 7	#define GLUT_KEY_HOME 106
#define GLUT_KEY_F8 8	#define GLUT_KEY_END 107
#define GLUT_KEY_F9 9	#define GLUT_KEY_INSERT 108
#define GLUT_KEY_F10 10	
#define GLUT_KEY_F11 11	
#define GLUT_KEY_F12 12	

예제

```
void keyboard(unsigned char key, int x, int y)
{
   switch (key)
   {
   case 27:
   case 'Q' :
   case 'q' :
      exit(0);
      break;
   }
}
```

마우스 콜백

- 마우스 이벤트는 마우스 버턴을 누를 때 또는 마우스가 움직일 때 발생
- 마우스 콜백 함수를 등록하기 위한 프로토타입 함수
  - void glutMouseFunc(void(*func)(int button, int state, int x, int y));
  - 등록된 마우스 콜백 함수는 void mouse(int button, int state, int x, int y)의 형태로 선언되어야 함.
  - Button 파라미터에는 GLUT_LEFT_BUTTON, GLUT_RIGHT_BUTTON, GLUT_MIDDLE_BUTTON 등 버튼 종류를 뜻하는 상수 전달
  - State 파라미터에는 GLUT_DOWN, GLUT_UP 등 해당 버튼이 눌러진 상태인지, 아닌지를 의미하는 상수 전달
  - x, y 파라미터에는 이벤트 발생시의 커서 위치가 전달

마우스 콜백

- 마우스를 움직일 때 버튼 상태에 따른 별도의 콜백 함수 정의
  - void glutMotionFunc(void(*func)(int x, int y));
  - void glutPassitveMotionFunc(void(*func)(int x, int y);
  - glutMotionFunc()은 버튼을 누른 상태에서 마우스를 움직일 때 호출되는 콜백 함수 등록.
  - glutPassiveMotionFunc()은 아무런 버튼을 누르지 않은 상태에서 마우스를 움직일 때(passive motion) 호출되는 콜백 함수 등록
- 마우스가 윈도우 안으로 들어오거나 밖으로 나가는 이벤트에 대한 콜백 함수
  - void glutEntryFunc(void(*func)(int state));
  - State 파라미터에는 GLUT_ENTERED(윈도우 밖에서 안으로) 또는 GLUT_LEFT(윈도우에서 나감) 상수 할당.