



R E P O R T

실습과제 03

내용

과제 명세

실습 3.1: 컴퓨터에 GLUT 설치하기.

실습 3.4: 자신의 이름 그리기.

실습 3.5: Sierpinski Gasket

고찰 및 느낀점

과제 명세

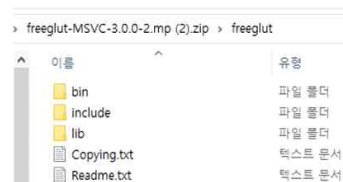
실습과제 03

- 강의자료의 다음 실습을 구현하시오.
 - 실습 3.1: 컴퓨터에 GLUT 설치하기.
 - 업로드 된 동영상 참조
 - 실습 3.4: 자신의 이름 그리기.
 - 실습 3.5: Sierpinski Gasket

4

실습 : GLUT 설치

- GLUT는 직접 설치해야 함
 - Visual Studio를 설치하면 GL, GLU는 자동으로 설치되어 있음.
 - OpenGL 홈페이지(<http://www.opengl.org/>)부터 탐색해 볼 것
 - 탐색하다가 **freeglut 3.0.0 MSVC Package** 를 다운로드
 - 이제 설치하면 됨. (How???)
 - 동영상 확인



- GLUT Source File도 있음
 - 화려한(?) Windows Api 기술이 들어 있음
 - 인터넷에서 찾아볼 것. **glut-3.7.6-src.zip**
 - 여러 가지 test / sample / demo 프로그램들 테스트
 - Sample 폴더의 파일들을 컴파일해 볼 것
- C:\Program Files (x86)\Windows Kits\10\Include\10.0.17763.0\um\Wgl
- C:\Program Files (x86)\Windows Kits\10\Lib\10.0.17763.0\um\Wx86
- C:\Windows\SysWOW64

실습3.4 : 자신의 이름을 그리는 함수

- 이름을 도형으로 그려본 후, 각 정점의 좌표를 계산
- 계산된 좌표를 이용해 자신의 이름을 출력하는 함수를 구현한다. 이때 다각형(GL_POLYGON)으로 그리는 경우 조심해야 할 부분이 있다. 다각형의 형태가 볼록(convex)해야 할 것이다. 오목(concave)한 부분이 있는 다각형은 왜 원하는 대로 출력되지 않는지 생각해 보라. 실행 결과의 예는 다음과 같다.



73

실습3.5 : 재미있는 그림을 그리는 프로그램

- Sierpinski Gasket
 - ① 삼각형 내부에서 무작위로 한 개의 시작점을 잡는다.
 - ② 무작위로 세 개의 정점 중 한 개를 선택한다.
 - ③ 시작점과 무작위로 선택된 정점의 중간점을 잡는다.
 - ④ 그 위치에 작은 원 같은 표지를 함으로써 새로운 점을 표시.
 - ⑤ 이 새로운 점을 시작점으로 대치
 - ⑥ 단계 2로 돌아간다. (이 과정을 주어진 횟수만큼 반복)



74

실습 3.1: 컴퓨터에 GLUT 설치하기.

- 설치과정

1. <https://www.transmissionzero.co.uk/files/software/development/GLUT/freeglut-MSVC.zip> 다음 링크에서 freeglut.3.0.0 MSVC Package를 다운받는다.

freeglut 3.0.0 MSVC Package

This package contains 32 and 64 bit Windows DLLs, import libraries, and header file allowing freeglut applications to be written using Microsoft's Visual C++. The package was built from source code using Visual Studio 2013. The DLL is binary compatible with both my MinGW DLL, and the GLUT for Win32 DLL provided by Nate Robins. The DLL has been tested on Windows 98 SE, Windows ME, Windows 2000, Windows XP, Windows Vista, Windows 7 (64 bit), and Windows 8 (64 bit).

[Download freeglut 3.0.0 for MSVC](#) (with PGP signature and PGP key)

2. <http://user.xmission.com/~nate/glut/glut-3.7.6-src.zip> 다음 링크에서 glu 3.7.6 src를 다운받는다.



The OpenGL Utility Toolkit (GLUT), originally written by [Mark Kilgard](#), ported to Win32 (Windows 95,98,Me,NT,2000,XP) by Nate Robins. For more information check out the [README-win32.txt](#) provided in the distribution for installation instructions, and Win32 specific information..

The latest version of the library is 3.7.6 (Nov 8, 2001)!

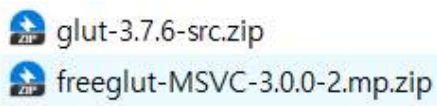
[glut-3.7.6-bin.zip \(117 KB\)](#)

GLUT for Win32 dll, lib and header file (everything you need to get started programming with GLUT).

[glut-3.7.6-src.zip \(4.76 MB\)](#)

GLUT source code distribution (including a whole slew of great example programs + data).

3. 다음 freeglut 파일의 압축을 해제하고 아래 과정을 진행한다.



- 4-1. [freeglut-MSVC-3.0.0-2.mp](#)[freeglut.lib](#)[freeglut.lib](#)파일을 아래 주소로

복사해준다.

C:\Program Files(x86)\Windows Kits\10\Lib\10.0.19041.0\um\x86

freeglut-MSVC-3.0.0-2.mp\freeglut\include\WGL

이름	수정한 날짜	유형	크기
freeglut.h	2015-03-14 오후 3:34	C/C++ Header	1KB
freeglut_ext.h	2015-03-14 오후 3:34	C/C++ Header	11KB
freeglut_std.h	2015-07-22 오전 8:45	C/C++ Header	27KB
glut.h	2015-03-14 오후 3:34	C/C++ Header	1KB

C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\Wgl

이름	수정한 날짜	유형	크기
freeglut.h	2015-03-14 오후 3:34	C/C++ Header	1KB
freeglut_ext.h	2015-03-14 오후 3:34	C/C++ Header	11KB
freeglut_std.h	2015-07-22 오전 8:45	C/C++ Header	27KB
GL.h	2020-06-23 오후 4:35	C/C++ Header	68KB
GLU.h	2020-06-23 오후 4:35	C/C++ Header	19KB
glut.h	2015-03-14 오후 3:34	C/C++ Header	1KB

4-2. freeglut-MSVC-3.0.0-2.mp\freeglut\include\WGL 디렉터리 내의 파일을 아래 주소로 복사해준다.

C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\Wgl

freeglut-MSVC-3.0.0-2.mp\freeglut\include\WGL

이름	수정한 날짜	유형	크기
freeglut.h	2015-03-14 오후 3:34	C/C++ Header	1KB
freeglut_ext.h	2015-03-14 오후 3:34	C/C++ Header	11KB
freeglut_std.h	2015-07-22 오전 8:45	C/C++ Header	27KB
glut.h	2015-03-14 오후 3:34	C/C++ Header	1KB

C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\Wgl

이름	수정한 날짜	유형	크기
freeglut.h	2015-03-14 오후 3:34	C/C++ Header	1KB
freeglut_ext.h	2015-03-14 오후 3:34	C/C++ Header	11KB
freeglut_std.h	2015-07-22 오전 8:45	C/C++ Header	27KB
GL.h	2020-06-23 오후 4:35	C/C++ Header	68KB
GLU.h	2020-06-23 오후 4:35	C/C++ Header	19KB
glut.h	2015-03-14 오후 3:34	C/C++ Header	1KB

4-3. freeglut-MSVC-3.0.0-2.mp\freeglut\bin\freeglut.dll 파일을 아래 경로로 복사해준다.

C:\Windows\SysWOW64

freeglut-MSVC-3.0.0-2.mp\freeglut\bin\freeglut.dll

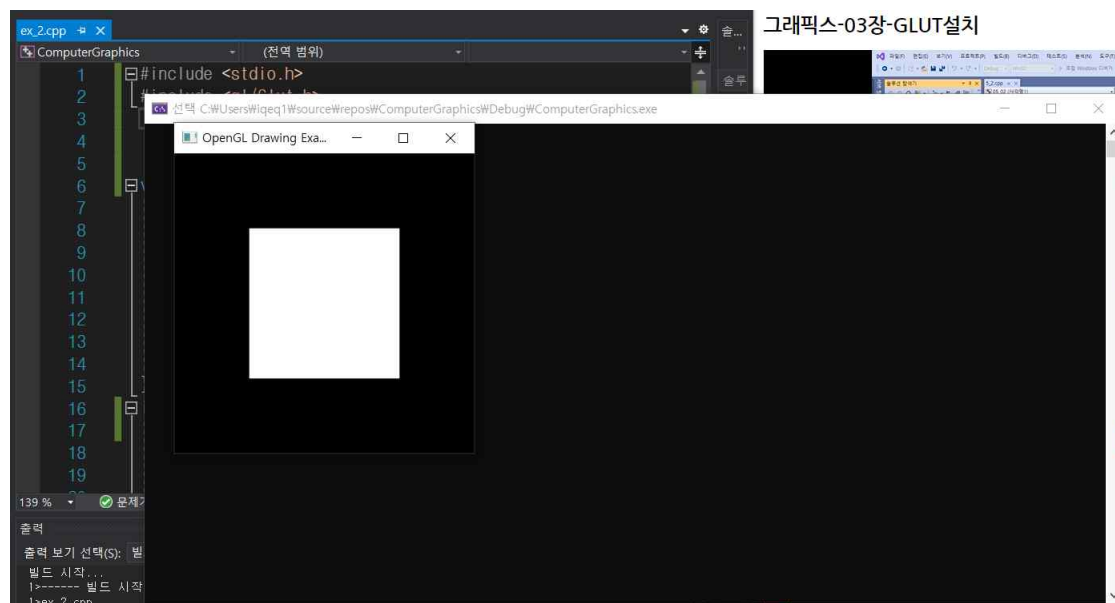
#freelut-MSVC-3.0.0-2.mp#freelutWbin			
bin 검색			
이름	수정한 날짜	유형	크기
x64	2015-07-22 오전 8:43	파일 폴더	
freelut.dll	2015-03-15 오후 2:28	응용 프로그램 확장	194KB

C:\Windows\SysWOW64

C:\Windows\SysWOW64	SysWOW64 검색
files	

5. glut-3.7.6-src의 여러 가지 test / sample / demo 프로그램을 테스트하고 sample 폴더의 예제를 컴파일해보기

대신해서 "OpenGL로 배우는 3차원 컴퓨터 그래픽스"의 프로그램을 실행화면을 보이겠습니다.



위와 같이 잘 실행됨을 확인할 수 있습니다.

실습 3.4: 자신의 이름 그리기

- 코드설명

헤더파일
<pre>#include <stdio.h> #include <math.h> #include <gl/GLut.h> // 원을 그리기 위해 math.h, glut를 사용하기 위해 gl/GLut.h를 추가했다.</pre>
void MyDisplay() 메서드
<pre>glClear(GL_COLOR_BUFFER_BIT); // GL_COLOR_BUFFER_BIT 색상버퍼를 지우는 버퍼 // DEPTH, STENCIL, ACCUM 등 깊이, 스텐실, 누적을 나타내는 버퍼도 존재 ----- glBegin(GL_POLYGON); // -> 확으로 쓰이는 부분을 나타내기 위해 사용 glBegin(GL_LINE_LOOP); // -> for문으로 'o'을 나타내기 위해 사용 /* glBegin() : GL_POINTS 독립적인 점 GL_LINE_STRIP 연결된 선분 GL_LINE_LOOP 시작점과 끝점을 이은 선분 GL_LINES 두개의 정점들을 이은 선분 GL_TRIANGLES 세개씩 연결한 삼각형 GL_TRIANGLE_STRIP 연결된 삼각형 GL_TRIANGLE_FAN 중심을 공유하는 삼각형 GL_QUADS 정점 4개씩을 연결하여 사각형을 그린다. GL_QUAD_STRIP 연결된 사각형 GL_POLYGON 연결된 볼록 다각형 */ ----- void glBegin(GLenum mode); // -> 정점의 시작을 의미 /* 이 부분에 glVertex3f(); 메서드를 넣는다. glVertex3f(X, Y, Z); // -> X, Y, Z 좌표를 표현하기 위해 사용 */ void glEnd(void); // -> 정점의 끝을 의미 void glflush(void); // -> 정점이 그려진 곳의 버퍼를 비우는 용도</pre>
main문
<pre>int main(int argc, char** argv) { glutInit(&argc, argv); glutCreateWindow("OpenGL Drawing Example"); glutDisplayFunc(MyDisplay); glutMainLoop(); return 0; }</pre>

```

/*
main의 인수
: int argc, char** argv
glutInit( )메서드의 인수를 전달하기 위해 사용
argc == arguments count : main 함수에 전달된 인자의 개수를 의미
argv == arguments vector : arguments vector로서 가변적인 개수의 문자열을 의미

glut를 초기화하기 위한 메서드
: glutInit(&argc, argv); // -> void glutInit(int *argcp, char **argv);

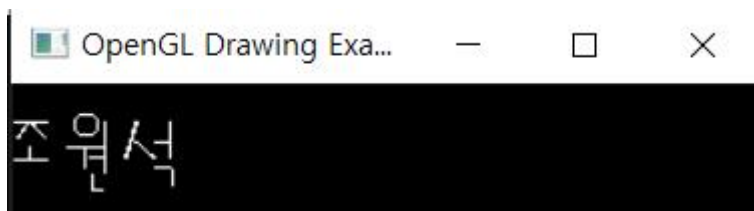
glut의 제목을 띄우기 위한 메서드
: glutCreateWindow("OpenGL Drawing Example");

glut를 도스창에 띄우기 위한 메서드.
사전에 정의한 MyDisplay()메서드를 인수로 갖는다.
: glutDisplayFunc(MyDisplay);

glut를 도스창에 끊임없이 띄우기 위한 메서드.
이를 하지 않을 시 OpenGL Drawing Example를 띄운 윈도우가 바로 꺼진다.
: glutMainLoop();
*/

```

- 실행결과



실습 3.5: Sierpinski Gasket

- 코드설명

전처리

```
#define GLUT_DISABLE_ATEXIT_HACK
// glut을 이용한 OpenGL 코드를 컴파일할 때 링커 에러가 방지하는 용도

#include <windows.h>
#include <GL/glut.h>
#include <iostream>
#include <random>

// GLint 타입을 사용하기 위해 windows.h, glut를 사용하기 위해 GL/glut.h를 난수를 사용하
// 기 위해 iostream과 random 헤더를 추가했다.
```

void MyDisplay() 메서드

[변수 선언부]

```
// GLint
/* 보통 int의 크기는 cpu 비트에 의존하지만 GLint는 OpenGL 타입으로 항상 32bit로 고정
할당되어있다. */
glColor3f(0.0, 0.0, 0.0);           // 색을 나타내기 위한 각각 R, G, B 순서
    GLint vx[] = { 100, 0, 200 };
    GLint vy[] = { 0, 200, 200 };
    // -> (100, 0), (0, 200), (200, 200)로 뿔어나간다.

    GLint px = 100;
    GLint py = 100;
    // -> ① 삼각형 내부에서 무작위로 초기 값 px와 py를 잡는다.
    std::random_device rd;           //시드값을 얻기 위한 random_device 생성.
    std::mt19937 gen(rd());          //random_device를 통해 난수 생성 엔진을 초기화한다.
    // 0 부터 2 까지 균등하게 나타나는 난수열을 생성하기 위해 균등 분포 정의.
    std::uniform_int_distribution<int> dis(0, 2);
```

[glux 관련 부분]

```
glClear(GL_COLOR_BUFFER_BIT);
// GL_COLOR_BUFFER_BIT           색상버퍼를 지우는 버퍼
// DEPTH, STENCIL, ACCUM 등 깊이, 스텐실, 누적을 나타내는 버퍼도 존재
glColor3f(0.0, 0.0, 0.0);           // 색을 나타내기 위한 각각 R, G, B 순서
// glBegin( ) : 정점의 시작을 의미, glEnd( ) : 정점의 끝을 의미
glBegin(GL_POINTS);                 // GL_POINTS 독립적인 점
void glEnd(void);                   // -> 정점의 끝을 의미
void glflush(void);                 // -> 정점이 그려진 곳의 버퍼를 비우는 용도
```

```

for (GLint i = 0; i < 10000; i++) {
    //-> ② 무작위로 세 개의 정점 중 한 개를 선택한다.
    int whichV = dis(gen) % 3;

    //-> ③ 시작점과 무작위로 선택된 정점의 중간점을 잡는다.
    int newPx = (px + vx[whichV]) / 2;
    int newPy = (py + vy[whichV]) / 2;

    //-> ④ 그 위치에 작은 원 같은 표지를 함으로써 새로운 점을 표시
    glVertex2i(newPx, newPy);

    //-> ⑤ 무작위로 세 개의 정점 중 한 개를 선택한다.
    px = newPx;
    py = newPy;
    //-> ⑥ 단계 2로 돌아간다. (이 과정을 주어진 횟수만큼 반복)
}

```

void Init() 메서드

```

glClearColor(1.0, 1.0, 1.0, 1.0);           // 배경색을 나타내기 위함. 흰색

// left, right, top, bottom으로 클리핑 평면의 좌표를 나타낸다.
gluOrtho2D(0, 250, 250, 0);

```

main문

```

glutInit(&argc, argv);
glutInitWindowPosition(100, 100);
glutInitWindowSize(500, 500);
윈도우의 위치, 크기, 값전달 등의 초기 설정을 해주고

glutCreateWindow("Sierpinski Gasket");
으로 윈도우의 이름을 정하고

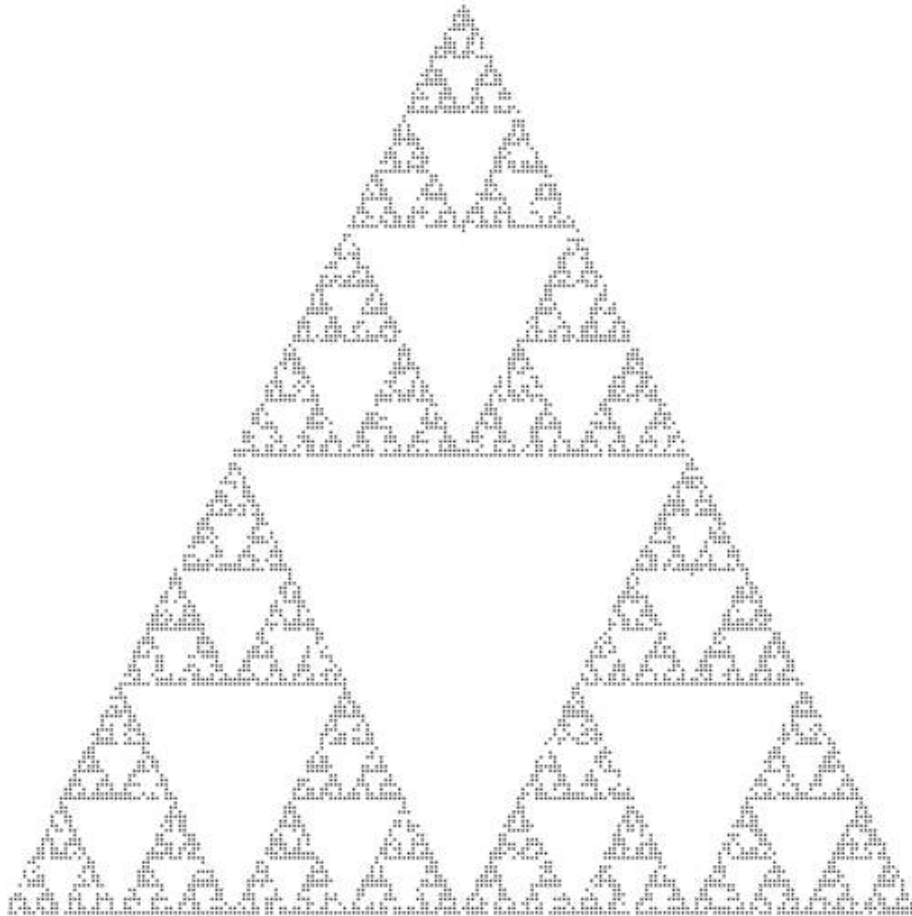
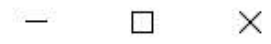
glutDisplayFunc(MyDisplay);와 init();
메서드를 실행한 뒤

glutMainLoop();
루프를 돌려준다.

```

- 실행결과

 Sierpinski Gasket



고찰 및 느낀점

이번 과제에서는 OpenGL을 학습하기 위해 glut.src와 freeglut를 설치 및 세팅하고 glut를 활용해 자신의 이름을 작성하고 Sierpinski Gasket을 그리는 실습을 해보았습니다. 먼저 OpenGL 초기 세팅을 진행하면서 컴파일러의 링킹과 라이브러리를 이식하는 과정을 손으로 직접 해보았었는데, 저수준의 세팅을 진행하는 등의 깊은 학습에 도움이 된 것 같아서 좋았습니다.

다음으로 이름을 작성하는 실습을 했을 때, 설명을 듣기 전에 집에서 혼자 했던 터라 큰 생각 없이 획은 polygon을 활용해 얇은 직사각형으로 만들고 'o'은 LineLoop와 삼각함수를 활용해서 만들어서 정말 반복 작업이 심하다고 느꼈었습니다. 하지만 수업에서 삼각형만을 이용해 이름을 쓸 수 있다는 것을 듣고 단순 반복 작업을 위한 것만은 아니었다는 새로운 관점을 알아갈 수 있었습니다.

마지막으로 Sierpinski Gasket을 그렸습니다. 난수를 발생시키기 위해 c에서의 rand가 아닌 c++의 random 헤더의 random_device class 등을 사용하고 OpenGL에서 Sierpinski Gasket을 그리는 과정을 하나하나 살펴보았습니다. 주어진 조건을 확인해나가며 결과보다 과정에 더 집중할 수 있는 시간을 가질 수 있었던 것 같습니다. 감사합니다.