

# OpenGL 기초

2021-2 컴퓨터 그래픽스

# 실습 1

- 필요한 라이브러리들을 설치한다.
  - freeGLUT, GLEW, GLM 라이브러리를 설치한다.
- 화면에 800 x 600 크기의 윈도우를 (0, 0) 위치에 띄운다.
  - 초기 배경색은 흰색
- 윈도우를 띄우고 배경색을 키보드 입력에 따라 다양하게 적용 해 보기
  - 키보드 입력 값:
    - R: 빨간색
    - G: 초록색
    - B: 파란색
    - A: 랜덤색
    - W: 백색
    - K: 검정색
    - T: 타이머를 설정하여 특정 시간마다 랜덤색으로 계속 바뀌게 한다.
    - S: 타이머 종료
    - Q: 프로그램 종료

# 실습 1

```
#include <iostream>
#include <gl/glew.h>
#include <gl/freeglut.h>
#include <gl/freeglut_ext.h>
GLvoid drawScene ( GLvoid );
GLvoid Reshape (int w, int h);
GLvoid Keyboard ( unsigned char key, int x, int y);
```

```
void main ( int argc, char** argv )
{
```

```
    //--- 윈도우 생성하기
```

```
    glutInit ( &argc, argv );
    glutInitDisplayMode ( GLUT_DOUBLE | GLUT_RGBA );
    glutInitWindowPosition ( 0, 0 );
    glutInitWindowSize ( 800, 600 );
    glutCreateWindow ( "Example1" );
```

```
    //--- GLEW 초기화하기
```

```
    glewExperimental = GL_TRUE;
    if (glewInit() != GLEW_OK)
    {
        std::cerr << "Unable to initialize GLEW" << std::endl;
        exit(EXIT_FAILURE);
    }
    else
        std::cout << "GLEW Initialized\n";
```

```
    glutDisplayFunc ( drawScene );
    glutReshapeFunc ( Reshape );
    glutKeyboardFunc ( Keyboard );
    glutMainLoop ();
}
```

```
// 필요한 헤더파일 include
```

```
//--- 윈도우 출력하고 콜백함수 설정
```

```
// glut 초기화
// 디스플레이 모드 설정
// 윈도우의 위치 지정
// 윈도우의 크기 지정
// 윈도우 생성 (윈도우 이름)
```

```
// glew 초기화
```

```
// 출력 콜백함수의 지정
// 다시 그리기 콜백함수 지정
// 키보드 입력 콜백함수 지정
// 이벤트 처리 시작
```

# 실습 1

```
GLvoid drawScene ( )  
{  
    //--- 변경된 배경색 설정  
    glClearColor ( ...);  
    glClear ( GL_COLOR_BUFFER_BIT );  
    glutSwapBuffers ( );  
}
```

//--- 콜백 함수: 그리기 콜백 함수

// 바탕색을 변경  
// 설정된 색으로 전체를 칠하기  
// 화면에 출력하기

```
GLvoid Reshape ( int w, int h )  
{  
    glViewport ( 0, 0, w, h );  
}
```

//--- 콜백 함수: 다시 그리기 콜백 함수

```
GLvoid Keyboard ( unsigned char key, int x, int y )  
{  
    switch (key) {  
        case 'R':    ...; break;  
        case 'G':    ...; break;  
        case 'B':    ...; break;  
        ...  
    }  
    glutPostRedisplay ();  
}
```

//--- 배경색을 빨강색으로 설정  
//--- 배경색을 초록색으로 설정  
//--- 배경색을 파랑색으로 설정

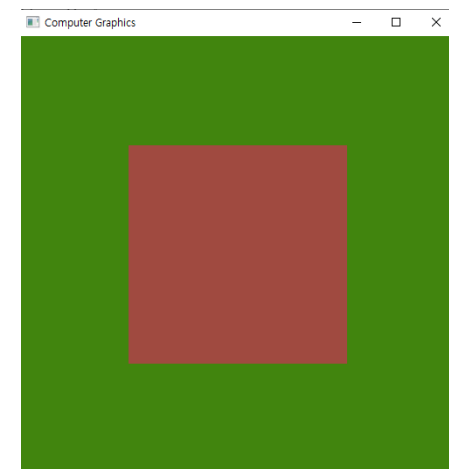
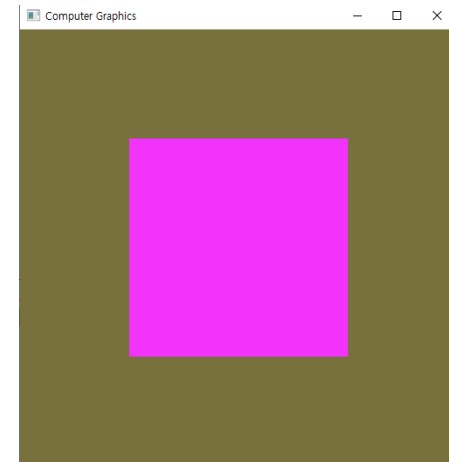
//--- 배경색이 바뀔때마다 출력 콜백함수를 호출하여 화면을 refresh 한다

## 실습 2

- 윈도우를 띄우고 마우스 명령을 실행 해 본다.
  - 화면 중앙에 사각형을 그린다.
  - 마우스를 클릭하여 색상을 바꾼다.
    - 사각형 내부 클릭: 사각형 색상을 랜덤하게 바꾸기
    - 사각형 외부 클릭: 배경색을 랜덤하게 바꾸기
- 사각형 그리기 함수
  - `void glRectf (GLfloat x1, GLfloat y1, GLfloat x2, GLfloat y2);`
    - (x1, y1): 좌측 하단 좌표값
    - (x2, y2): 우측 상단 좌표값
- 사각형 색상 바꾸기
  - `void glColor3f (GLfloat r, GLfloat g, GLfloat b);`
    - 현재 색상 설정하기
    - (r, g, b): red, green, blue 색상, 0.0 ~ 1.0 사이의 값으로 `glRectf` 를 호출하기 직전 설정한다.

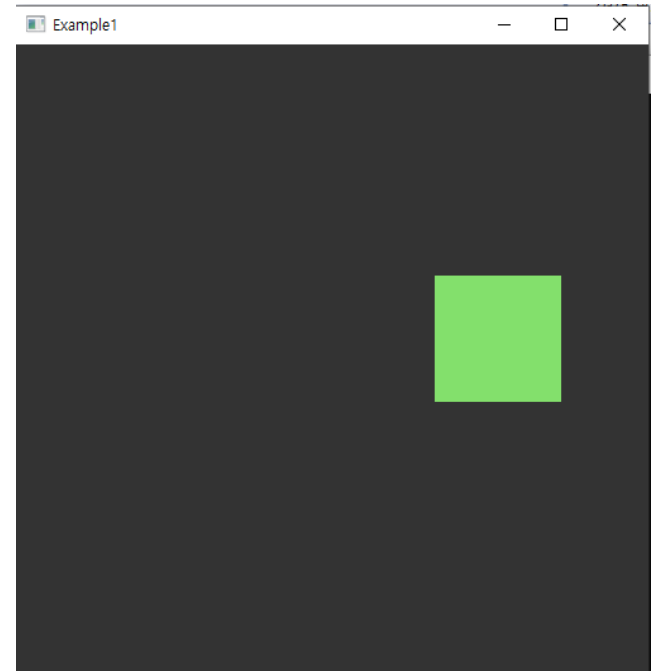
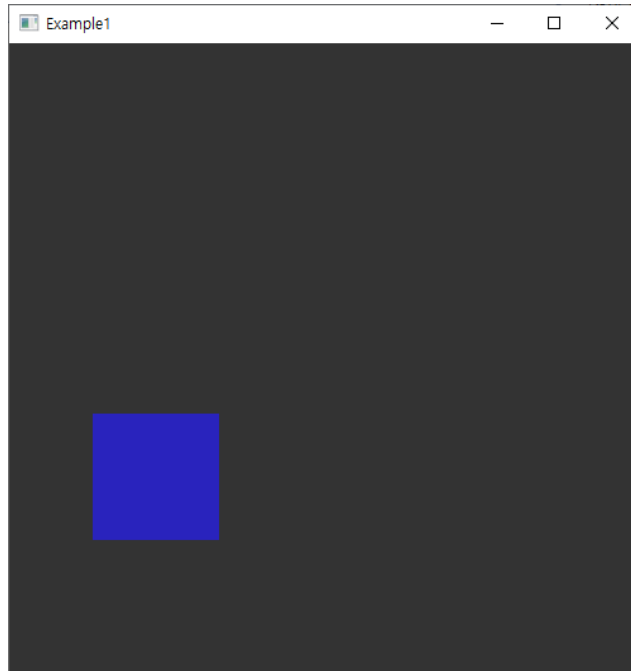
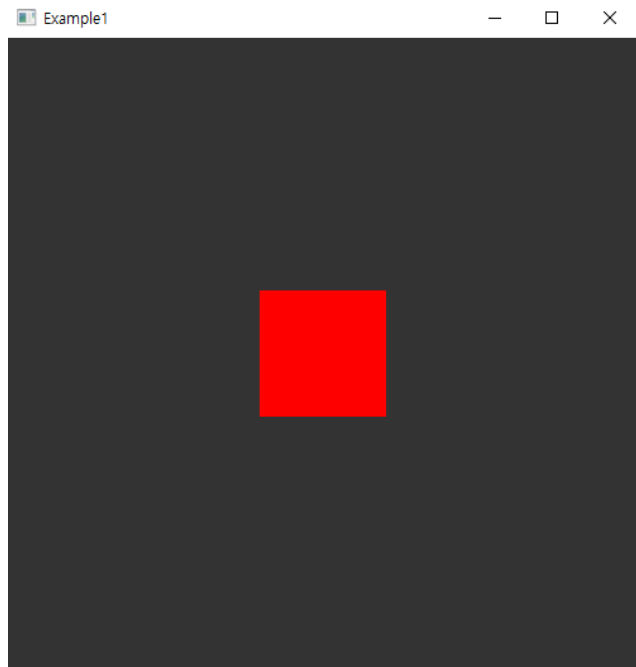
**\*\* Modern OpenGL에서는 사용할 수 없는 deprecated 함수**

**\*\* 셰이더를 사용하게 되면 이 함수는 사용할 수 없음**



## 실습 3

- 실습 2번 문제 업그레이드 하기
  - 실습 2번에서 사각형을 그린 후,
  - 마우스 버튼을 클릭한 채로 드래그 하면
    - 사각형의 위치가 이동된다.
    - 마우스를 놓으면 더 이상 사각형이 이동하지 않는다.
    - 마우스를 클릭하면 사각형의 색이 바뀐다.



## 실습 4

- 실습 3번 문제 업그레이드 하기
  - 배경색은 짙은 회색으로 정하고, 사각형은 랜덤한 색으로 그린다.
  - 마우스를 클릭하는 곳이 중심이 되어 사각형을 그린다.
  - 최대 5개의 사각형을 그린다.
- 키보드 입력:
  - a: 사각형들은 각각 대각선으로 이동하고 벽에 닿으면 튕겨 다른 방향으로 이동한다.
  - c: 애니메이션이 진행 중에 사각형의 크기가 변경된다.
  - s: 애니메이션이 멈춘다.
  - m: 원래 그린 위치로 사각형들이 이동한다.
  - r: 사각형들이 사라지고 리셋한 후 마우스 입력을 받을 수 있다.
  - q: 프로그램을 종료한다.

