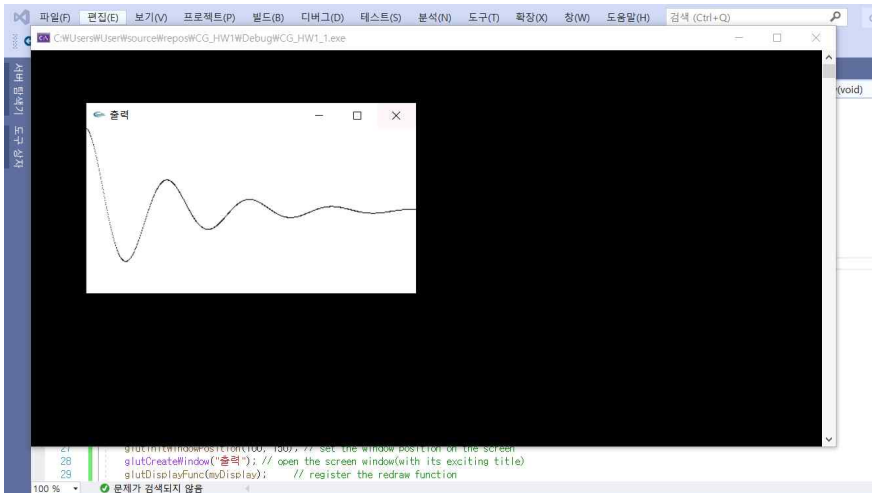


컴퓨터 그래픽스 HW1

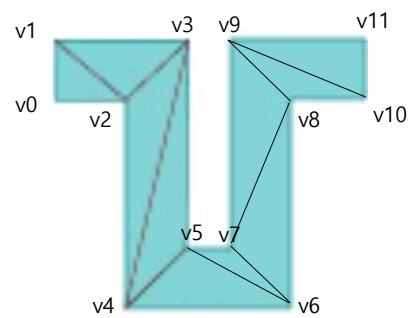
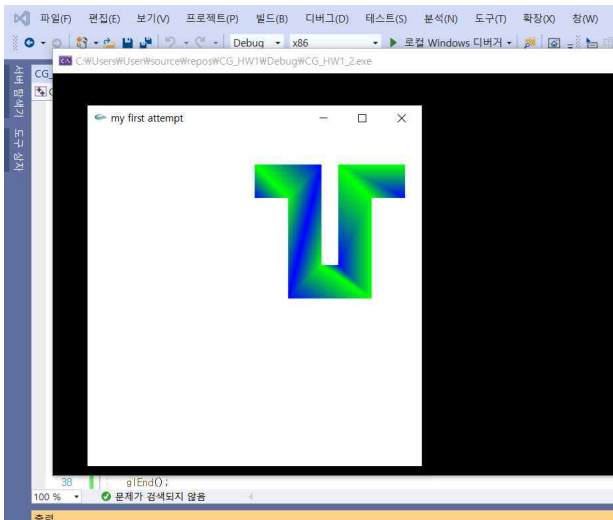
201501489 최영진

1. 실행화면



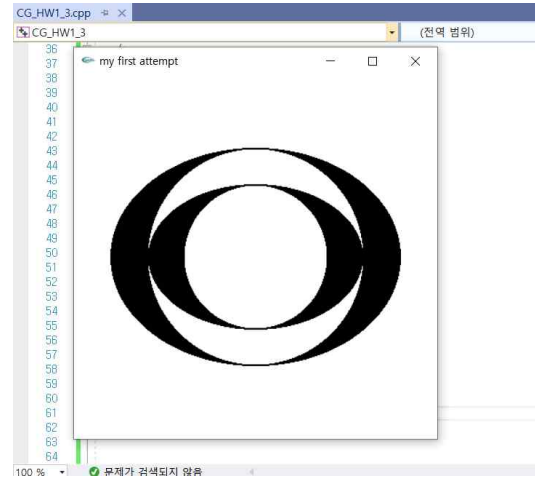
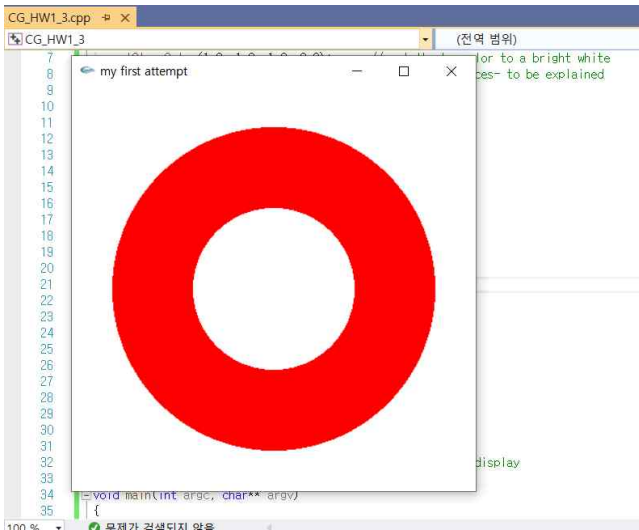
반복문 : GL_POINTS를 이용해야 했으므로 x의 테이타타입은 double로 설정하고 for문을 돌며 2개의 인자를 갖는 vertex(x,f(x))를 생성하도록 만들었다. 반복문은 x 값이 0에서 4까지 0.005씩 증가하는 동안 glVertex2d(x, exp(-x) * cos(2 * M_PI * x)) 를 그린다.

2. 실행 화면



토의 : 총 12개의 vertex이용, 시계방향으로 orientation을 동일하게 맞추었다. (v0, v1, v2), (v2,v1,v3), (v2,v3,v4), (v4,v3,v5), (v4,v5,v6), (v6,v5,v7), (v6,v7,v8), (v8,v7,v9),(v8,v9,v10), (v10, v9, v11)



3. 실행화면



토의 : drdisc함수는 반지름 r을 갖고, 중심이 (a,b)인 원 위의 점 좌표가 $(a+r*\cos\theta, b+r*\sin\theta)$ 인 것을 이용하여 그렸다. 샘플링 할 개수를 임의로 5000이라 두고 for문을 돌며 점을 찍어내도록 했다. 물론 가장 첫 번째 점은 중심점으로 잡도록 for문 이전에 첫 번째 vertex로 지정해두었다.

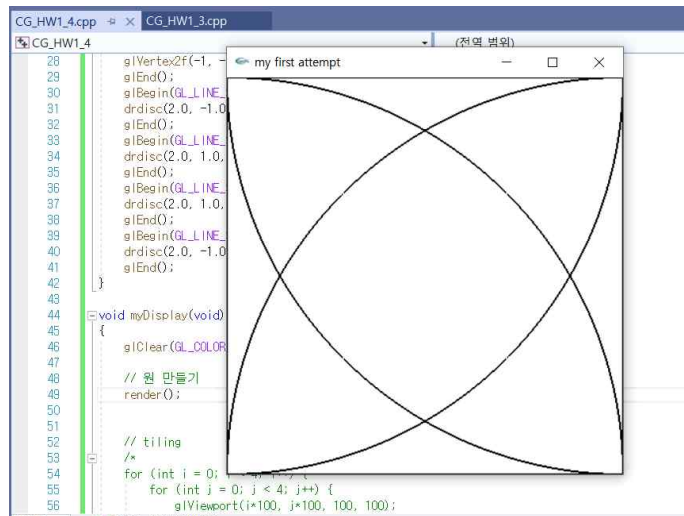
오른쪽 타원과 겹친 원의 경우 타원을 만드는 함수 drdisc_2를 만들어 똑같이 타원 위의 점 공식을 이용하여 $(c+a*\cos\theta, d+b*\sin\theta)$ 의 점을 for문을 돌며 만들어내게 했다. 먼저 가장 큰 검은 타원을 그리고 그 다음 내부에 큰 흰 원을 그렸다. 이어서 큰 흰 원 내부에 작은 검은 타원을 그린 후 마지막으로 그 내부에 작은 흰 원을 그리도록 했다. 그림은 가장 처음 그림 타원 위에 총 3번을 겹쳐 그렸다.

윈도우 크기는 정사각형 모양이 되도록 가로, 세로 모두 400픽셀로 맞추었다. 가시공간은 원들의 중심으로 잡아둔 (25, 75)가 정중앙에 오고 주어진 원들이 비율에 맞게 배치되도록 설정했다. `gluOrtho2D(0.0, 50.0, 50.0, 100.0)` ←빨간 도넛 기준 가장 바깥쪽 원의 좌표에서 5픽셀씩 여유가 있도록 지정해두었다.

	<pre>glColor3f(1.0, 0.0, 0.0); drdisc(20.0, 25.0, 75.0); // 빨간색 큰 원 먼저 glColor3f(1.0, 1.0, 1.0); // white drdisc(10.0, 25.0, 75.0); // 흰색 작은 원 나중에</pre>	
---	--	---

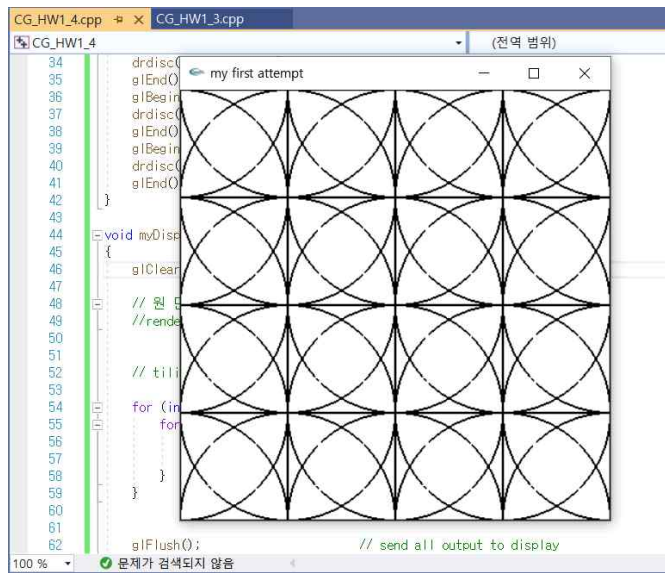
4.

(a) 실행 화면



토의: 가시공간을 x축 -1에서 1까지 두고 y축도 -1에서 1까지 둔 후 가시 공간의 각 꼭짓점을 중심으로 하는 반지름 2인 원을 4개 그렸다. 이때 가시공간이 제한되어있기에 각 원의 일부분(사분원)만이 4개 보이게 된다. 가시공간을 둘러싸는 선도 보이게 하기 위해 선의 두께는 2픽셀로 지정했다. 해당 식은 밑의 (b)에서도 사용하므로 render라는 함수로 만들다.

(b) 실행 화면



토의 : 위의 (a)에서 만든 render함수를 viewport에 집어넣으면 된다. (a)의 그림이 가로 4번, 세로 4번으로 총 16번 들어가게 되는 것이므로 for문을 두 개 열어 내부에서 viewport를 지정해서 render를 그려주면 된다. 내가 지정한 윈도우의 크기는 가로, 세로 400픽셀이므로 각 viewport의 크기는 가로, 세로 100픽셀이다. → glViewport(?, ?, 100, 100)

이제 for문을 따라 100픽셀씩 이동하도록 코드를 채워 넣으면 glViewport(i*100, j*100, 100, 100)과 같은 식으로 그림을 그릴 수 있다.