1번.

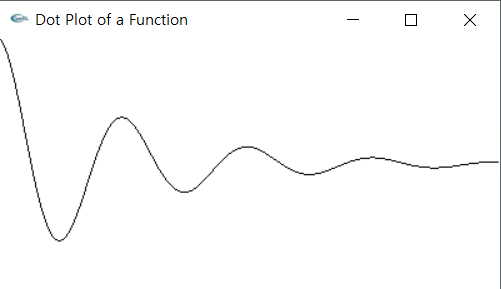
For문을 사용하여 x가 0부터 4까지 0.005씩 증가하여 f(x)가 결정되며 이를 그래프로 나타냈다.

for (float x = 0; x <= 4.0; x = x + 0.005) {

double func = exp(-x) \* cos(2 \* PI \* x);

glVertex2f(x, func);

}

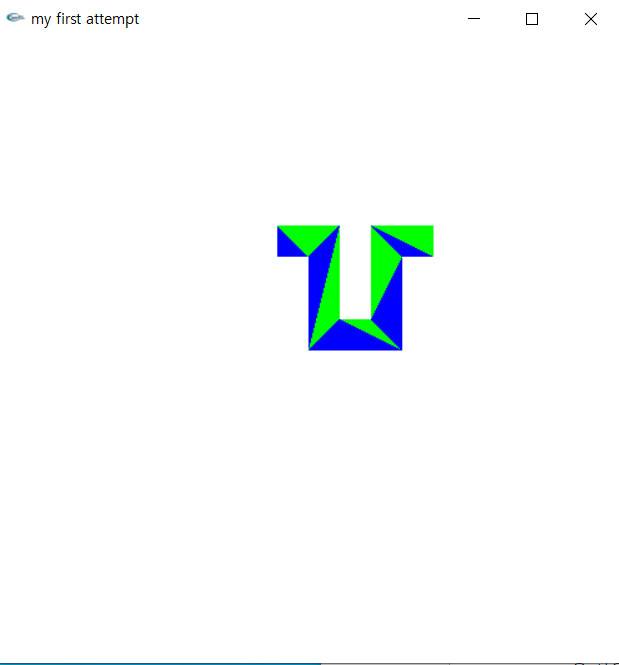


2번.

총 12개의 vertex로 구성하였으며 각 삼각형의 색을 vertex별로 지정하였고 glShadeModel(GL\_FLAT)을 사용하여 마지막 정점의 색을 해당 삼각형의 색으로 지정하였다.

모든 triangle의 orietation을 같게 만들면 아래 그림과 같은 형태가 나오지 않는다.

만약 삼각형 왼쪽이 1 위쪽이 2 오른쪽이 3으로 가정하고 시계방향으로 orietation을 정해 다음 4번째 vertex의 위치를 고려할때 2-3을 잇는 직선 오른쪽에는 4번 vertex가 존재할 수 없으므로 왼쪽에 위치하게 된다. 아래의 도형은 2-3을 잇는 직선 오른쪽에 4번 vertex가 위치하므로 불가능하다.



3번.

void drdisc(float r, float x, float y) {

glBegin(GL\_TRIANGLE\_FAN);

int N = 1000;

float a; float b;

for (float i = 0; i <= 2 \* PI; i += PI / N) {

a = x + r \* cos(i);

b = y + r \* sin(i);

glVertex2f(a, b);

}

glEnd();

}

void ellipse(float r, float x, float y) {

glBegin(GL\_TRIANGLE\_FAN);

int N = 1000;

float a; float b;

glColor3f(0, 0, 0);

for (float i = 0; i <= 2 \* PI; i += PI / N) {

a = pow(sqrt(r)+1,2) \* cos(i);

b = r \* sin(i);

glVertex2f(x + a, y + b);

}

glEnd();

}

Void main() {

~~~~

glColor3f(1, 0, 0);

drdisc(20, 25, 75);

glColor3f(1, 1, 1);

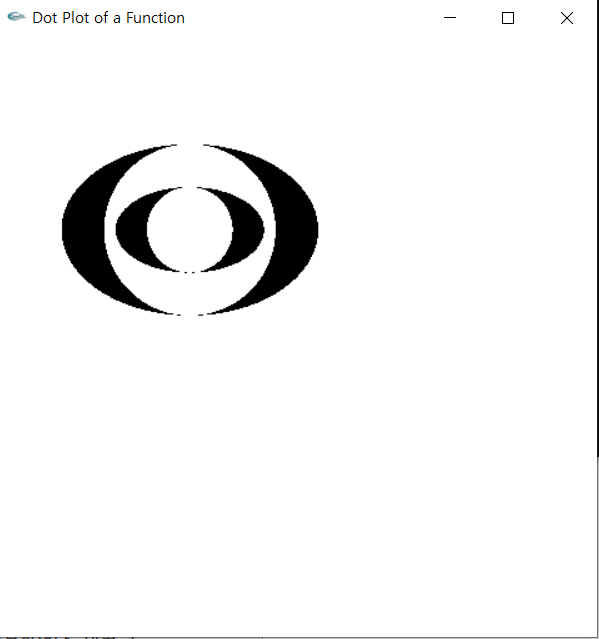
drdisc(10, 25, 75);

~~~~

}

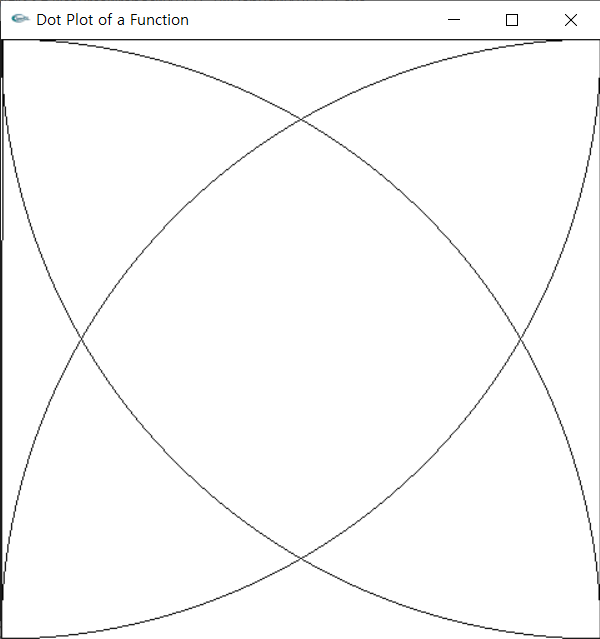
drdisc에서 반지름, x좌표, y좌표를 매개변수로 전달받아 x를 a로, y를 b로 치환하여 각 i가 변화함에따라 glBegin(GL\_TRIANGLE\_STRIP)을 이용, 원을 그려내고, 2번째 호출에선 반지름이 10인 원을 그려 빨간원 안에 하얀색 원이 있는 그림을 그려낸다. Ellipse에선 타원을 그리는데 x축을 장축으로 두기위해 반지름의 제곱근에 1을 더하고 이를 제곱하여 b보다 긴 장축으로 설정하여 위치를 ( (x+a), (x+b) )로 하여 그려낸다. 이 역시 x를 a로 y를 b로 치환하여 계산한다.

타원을 검은색으로 찍어내고, 흰색으로 원을 찍어내는걸 2번 반복하여 총 3번 덮어씌워 아래와 같은 모양을 만들어 낼 수 있다. 가시공간은 (-20, 120, -20, 120)으로 설정하였고 윈도우크기도 같은 비율로 480x480으로 설정하였다.



4번

(a) 좌표 0,0을 기준으로 반지름이 1인 원을 90도씩만 그리고 좌표를 옮겨가며 시작 각도를 조정하여 다음과 같은 그림을 그려낸다. 원의 방정식을 이용하여 치환을 이용, GL\_LINE\_STRIP으로 그려낸다.

.

(b)

2중 포문을 이용하여 glViewport(i\*120, j\*120, 120, 120)으로 설정하여 i값과 j값이 변함에따라 viewport의 위치가 달라져 tilling 구현이 가능하다. Window의 크기를 패턴의 타일이 4x4이고 하나의 크기가 120이므로 480x480으로 설정하였다.

