//Qianzi Xu(Rachel) Assignment 3

void setup() {

size(displayWidth, displayHeight);

background(0);

noStroke();

translate(width/2, height/2);

int points = 20;//the number of vertices in the original star

int numOfStars = 40;//draw the star 40 times

float radius = displayHeight \* 0.5;//outer radius

float inner = 0.25 \* radius;//inner radius

float radStep = radius/numOfStars;

float inStep = inner/numOfStars;

//each time drawing the star, decrease the radius by

//radStep (outer radius) or inStep (inner radius)

float alphaStep = 255/numOfStars;

float rotateStep = 80/numOfStars;

for (int i = 0; i < numOfStars; i++) {

stroke(#CE1581, 200-alphaStep\*i);

strokeWeight(2-i/20);

//changing of color as i increases

if (i < 10) {

fill(#76B9FF, 150-alphaStep\*i\*2);

}

else if (i < 20) {

fill(#F7D93E, 180-alphaStep\*i\*2);

}

else {

fill(255, 100-alphaStep\*i\*2);

}

pushMatrix();

rotate(rotateStep\*i\*PI/180);

drawStar(0, 0, radius-radStep\*i, inner-inStep\*i, points);

popMatrix();

}

}

void drawStar(float cx, float cy, float radius, float inner, int points) {

float theta = 0, delta = 2\*PI/points;

float x1 = cx + radius\*cos(theta);

float y1 = cy + radius\*sin(theta);

float x2, y2;

beginShape();

vertex(x1, y1);

//if i is even, draw the inner points

//if i is odd, draw the outer points

for (int i = 0; i < points; i++) {

theta += delta;

if (i % 2 == 0) {

x2 = cx + inner\*cos(theta);

y2 = cy + inner\*sin(theta);

}

else {

x2 = cx + radius\*cos(theta);

y2 = cy + radius\*sin(theta);

}

vertex(x2, y2);

}

endShape(CLOSE);

}