

● 程式碼截圖：

```
float x,y,r = 0;
float xoff,yoff,zoff = 0;
float noiseMax = 0;
float spin = 0;
float k,s = 0;
import processing.sound.*;
SoundFile song;

void setup(){
    song = new SoundFile(this, "music.mp3");
    song.loop();
    colorMode(HSB,360,100,100);
    size(600,600);
}

void draw(){
    background(0);
    stroke(s,20,20);
    fill(s,50,50);
    strokeWeight(5);
    translate(width/2,height/2);
    if (mousePressed &&(mouseButton) == LEFT){
        scale(map(mouseX,0,width/2,0.5,2));
    }
    for (float n = 0.5; n<5 ; n = 1.8*(n+0.5)-1){
        beginShape();
        noiseMax = map(mouseX, 0, width/2, 0 ,25);
        for(float a = 0; a<TWO_PI; a+=0.01){
            float xoff = map(cos(a + spin), -1, 1, 0, noiseMax);
            float yoff = map(sin(a + spin), -1, 1, 0, noiseMax);
            r = map(noise(xoff, yoff, zoff), 0, 1, 50*n, 100*n);
            x = r*cos(a);
            y = r*sin(a);
            vertex(x,y);
        }
        if (n==0.5){
            line(0, 0, x, y);
        }
        if (n>0.5){
            noFill();
        }
    }
}
```

```

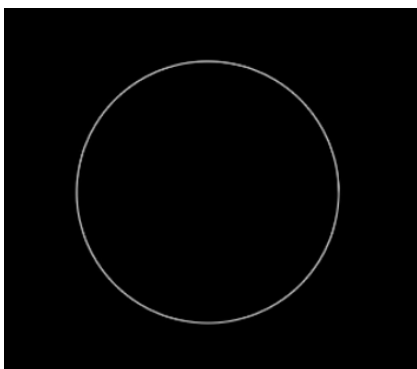
endShape(CLOSE);
if (k == 300){
    s += 40;
    if (s == 360){
        s = 0;
    }
    k = 0;
}
stroke(s,30*1.5*n,30*1.5*n);
zoff += 0.05;
if (mousePressed &&(mouseButton) == RIGHT){
    spin += 0.01;
}
else{
    spin += 0.005;
}
k++;
}
}

```

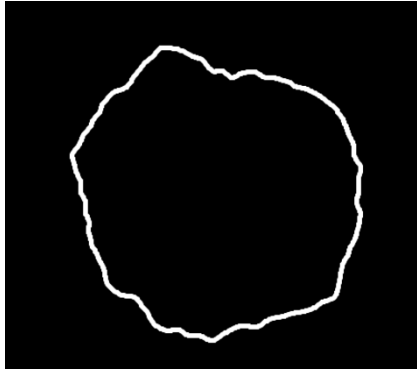
- 「原先」動畫運行之流程與截圖：

1. 將原點 (0, 0) 移至畫面中點，並利用 for 迴圈、

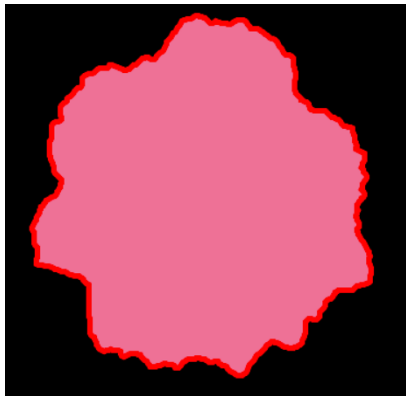
beginShape()、endShape()、vertex 在畫面中央畫一個圓



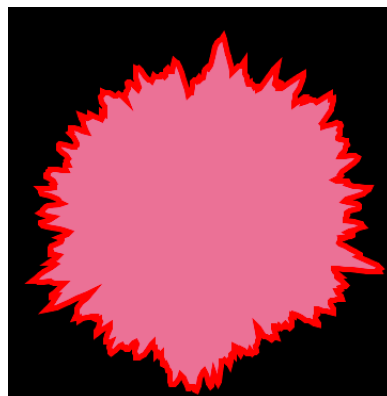
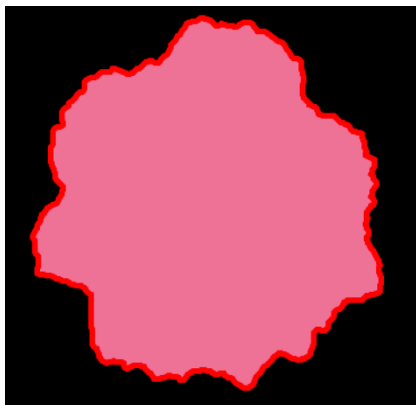
2. 用 noise 函數值調整其半徑，並用 map 拉大 noise 函數的隨機值範圍分佈，並讓圓上每一點到原點的距離不斷地變動，然後用 strokeWeight 調整線的粗度



3. 用 fill、stroke、line 將圖形填滿著色

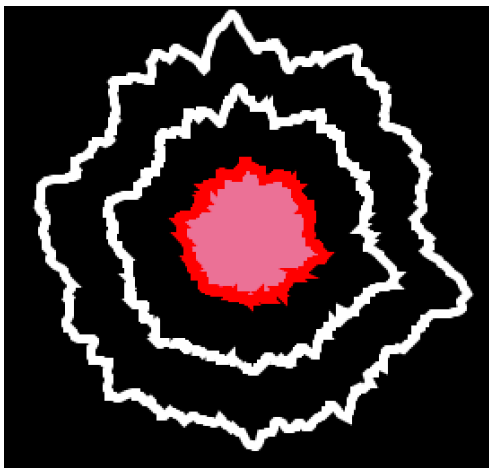


4. 以變數 noiseMax 決定圖形變動劇烈程度，並用滑鼠往左往右來調整其劇烈程度，再設 spin 變數，讓圖形有旋轉的感覺。

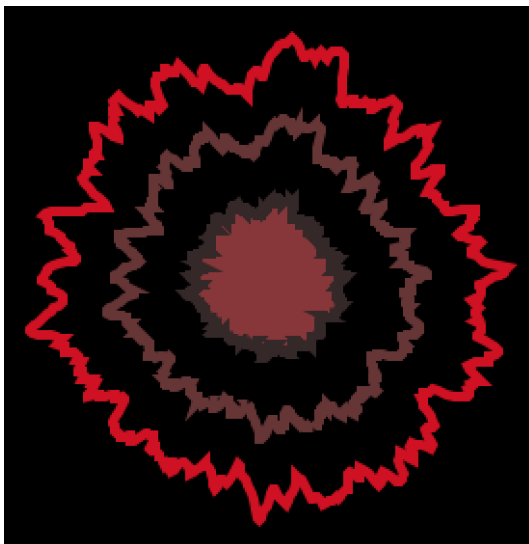


- 請說明你做了哪些「修改」，並附上截圖：

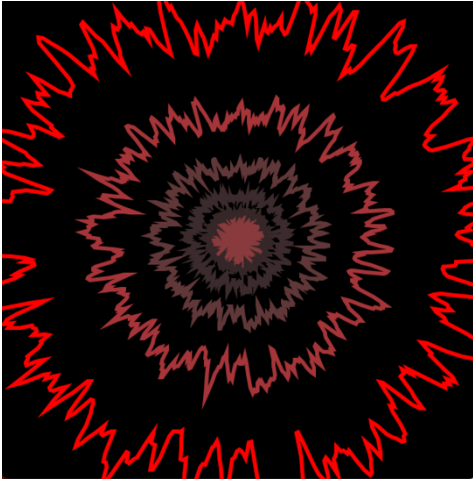
1. 運用 sound 函式庫播放音樂(須先安裝 sound 函式庫)，用 loop 迴圈使音樂循環播放。
2. 新增 for 迴圈，讓其在外圍多增加幾個變動的圓，並以 if 判別第一個圓為實心的，剩下外面的圓為空心的。



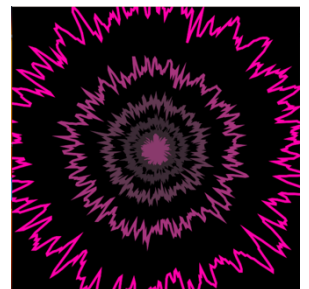
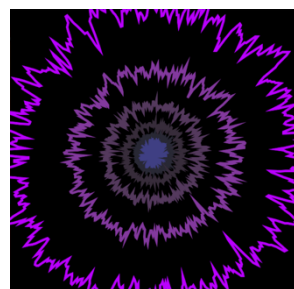
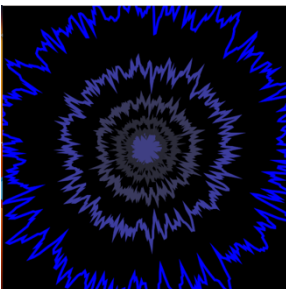
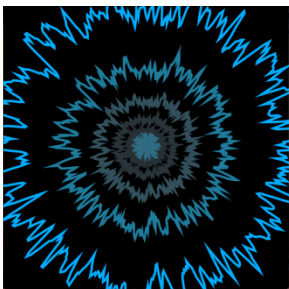
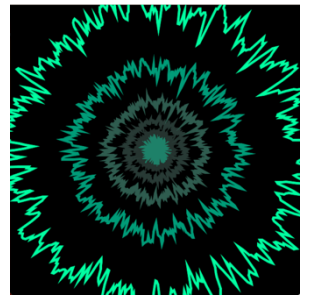
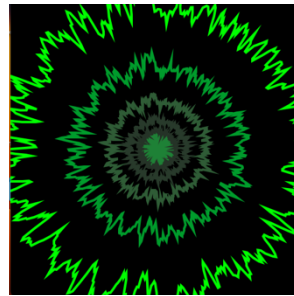
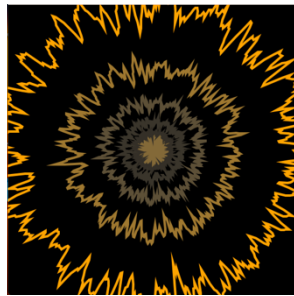
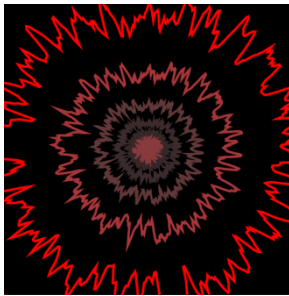
3. 以變數調整各層的颜色變化



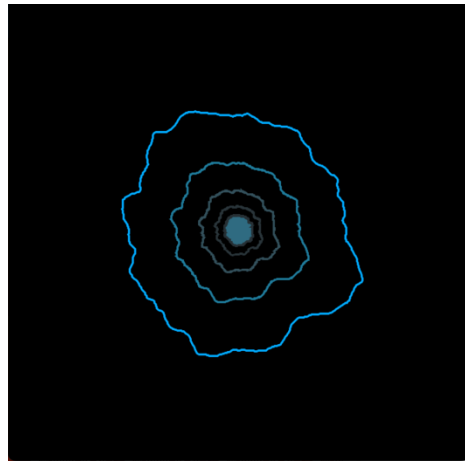
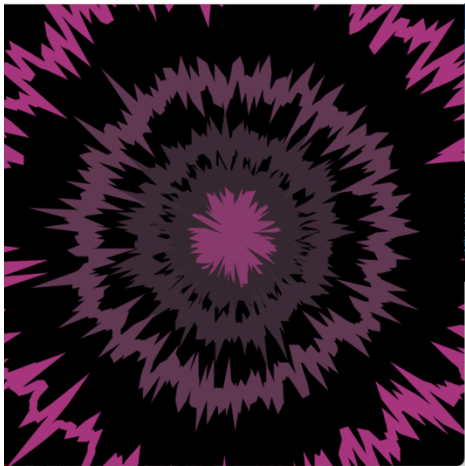
4. 調整變數 n 的變量，讓每個圓的間距不一



5. 以 `colorMode` 更改顏色的範圍值，接著以 `if` 函式判斷其顏色變化的時機，隨著時間，顏色依序變化



6. 按左鍵並往右拉，圖形會變大且線條變化劇烈，按左鍵往左拉，
圖形會縮小且線條變化趨緩



7. 按右鍵，可以讓圖形的轉動速度變快



個人心得：期中過後，老師又教了 3D 效果與 noise 函數，我本來以為 processing 只能做 2D 的設計，沒想到還能做立體的視覺效果，創作「星球運轉動畫」那幾堂課讓我印象深刻，星球運轉的向量、速度等都要考慮進去，雖然這堂課內容有一點困難，可是卻讓我收穫滿滿，做出成品後，也相當自豪。

接著便是 noise 噪波函數的單元了，noise 噪波函數標榜用最少的程式碼，做出最自然的視覺效果，它的原理不難理解，不過如何運用倒是我覺得困難的一個點，感覺沒有那麼的直觀，需要思考一下。經過幾堂 noise 為主題的課後，我驚訝於 noise 函數所做出的設計，尤其是「從飛機上鳥瞰地形」那個動畫，noise 函數與 3D 效果結合在了一起，更讓我驚喜，而且程式碼並不會很多，畢竟簡潔、美麗、適用性廣一直都是 Noise 函數的特色。

經過一個學期後，我對 processing 有一定的了解，同時也發現 processing 的視覺化的設計，似乎沒有太大的局限，雖然製作過程偶爾會碰壁，抑或是腦中想像的圖像走向跟實際程式跑出來的動畫不一樣，不過在嘗試與思考的過程中，漸漸找出了答案，望著自己完成的動畫作品，覺得很成就感。

關於課程的小建議，希望可以放多一點時間在 noise 函數上，我覺得它應用起來還蠻有趣的。