Computational Thinking

Analysis

Canva size: 600x600 bg color is black

variables

- baseSize: base size for the circles
- maxSets: maximum nb of sets to draw
- fillOpacity: Opacity for the circles fill color
- minCircles: minimum nb of circles in each set
- maxCircles: max nb of circles in each set

functions:

```
setup();
draw();
keyPressed();
```

set the angleMode to degrees

Abstraction:

- Create Canva createCanvas(600, 600);
- Disable continous loops noLoop();
- fill background color (0); → is black
- Move the origin of the drawing to the center using translation translate(width/2, height/2) so (0,0)
- Use a loop to create multiple layers of circles Each layer/set will have its unique characteristics

Randomization

generate random colors for filling the circles

randomly determine the nb & size of the circles

To calculate the circles positin use trigonometry

Pattern Recogniti

Usuage of loop to create sets of circles
Randomization of colors for each circle
circle placement is random

multiple circles are created with random properties but they all follow a structured approach for placement & coloring.

algorithmic Design

Variables
Var baseSize;
Var maxSets;
Var fillOpacity;
Var minCircles;
Var maxCircles;

To center the pattern

Translate (width/2 height/2);

JMA® Opening the future

Outer loop for sets of circles

```
for( i= 0 ; i< maxSets; i++)
```

This loop is responsible for creating multiple layers of circle

push( ); inorder to save the current state

rotate(random(360); (Applies a random rotation to the layer)

Color generation RGB

```
var ColorR = random(150, 255);
var ColorG = random(150, 255);
var ColorB = random(150, 255);
```

so the function will pick a random value for each color between 150 & 255.

```
fill (colorR, colorG, colorB);
```

fill opacity

This controls how transparent the color will be
Its a value between 0 & 255

Calculates the distance
radius = baseSize * i * 0.3 from the center

i is starting from 0 to max sets
as i increases the distance from the center increases

eg. when i = 1
radius = baseSize * 1 * 0.3

0.3 : each layers radius will be 30% of the baseSize so the circles can fit of the canvas

nomCircles = int(random(minCircles, maxCircles));

we use the int( ); function so we dont get a decimal nomber.

2nd loop

for(j=0 ; j < nomCircles; j++);

This loop is responsible for drawing multiple circles

inside the loop : a circle will be drawn and it will have ~~two~~ three instructions

angle = random(360); generates a random angle

x = cos(angle); calculates x posit
y = sin(angle); calculates y posit

Both * radius to calculate positi

JMA® Opening the future

80% of base size ← → 120% of base size

circleSize = baseSize * random(0.8, 1.2);

ellipse(x, y, height, width);

ellipse(x, y, circleSize, circleSize);

x, y are calculated from cos & sin

We used circleSize twice because width & height are same for a circle

pop();

function keyPressed() for user interaction

if key == L redraw();