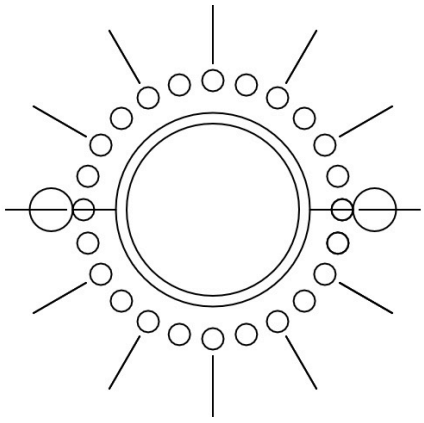
Generative Identity

Submission VI – Documentation
Data Visualisation
Malay Vasa

Ideating

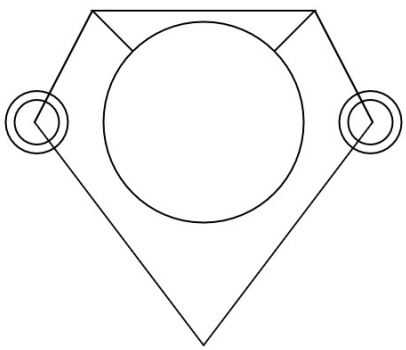
Each and every alphabet required a unique shape to not only be designed, but also coded in javascript. This presented a unique challenge as i could use only basic shapes such as circles, triangles, quadrilaterals etc. But fortunately in p5.js I had the ability to overlap these, and i had the power of loops.

Being unsure whether I would be able to actually code what I drew, I began the process of brainstorming the first few designs in code itself. Here the idea of making something that didn't represent any meaning was handy, I had explored this in my drawing and mark making class.



```
push();

translate(400,400);
stroke(scolor);
noFill();
ellipse(0,0,200);
ellipse(0,0,225);
line(237.5-400,0,287.5
-400,0);
line(512.5-400,0,562.5
-400,0);
circle(212.5-400,0,50);
circle(587.5-400,0,50);

pop();
```

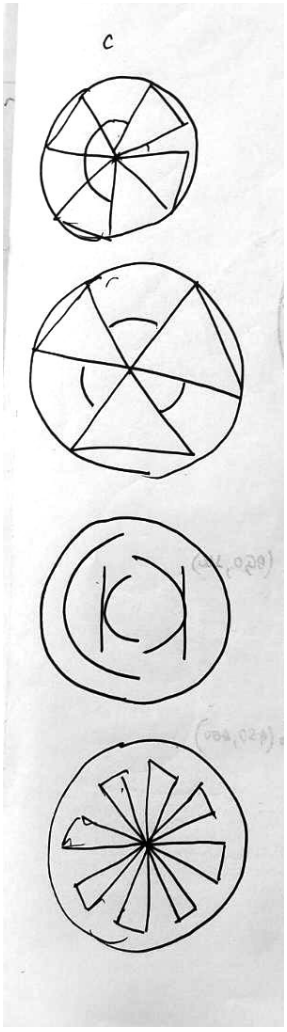
```
var a=0; var b=0;
for(var i = 0; i<26;i++)
{
noFill();
stroke(scolor);

circle(400+150*cos(a),4
00+150*sin(a),25);

line(400+170*cos(b),40
0+170*sin(b),400+240*c
os(b),400+240*sin(b));

a=a+15;
b=b+30;
}
```

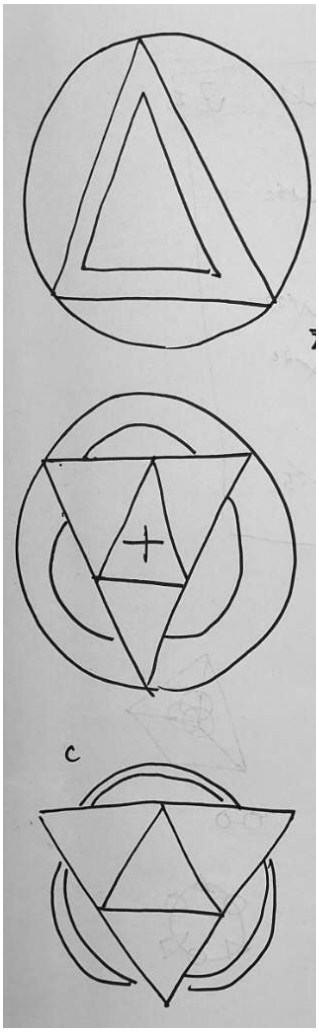


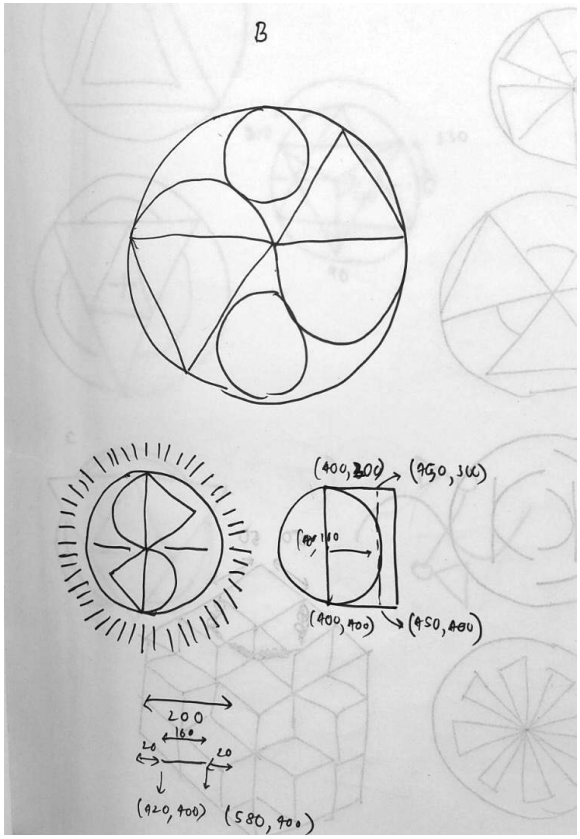
```
 stroke(scolor);
   fill(bcolor);

triangle(400,400,400-1
25,400-125,400+125,40
0-125);

push();
stroke(scolor);
fill(bcolor);

translate(400,400);
ellipse(0,0,225);
```

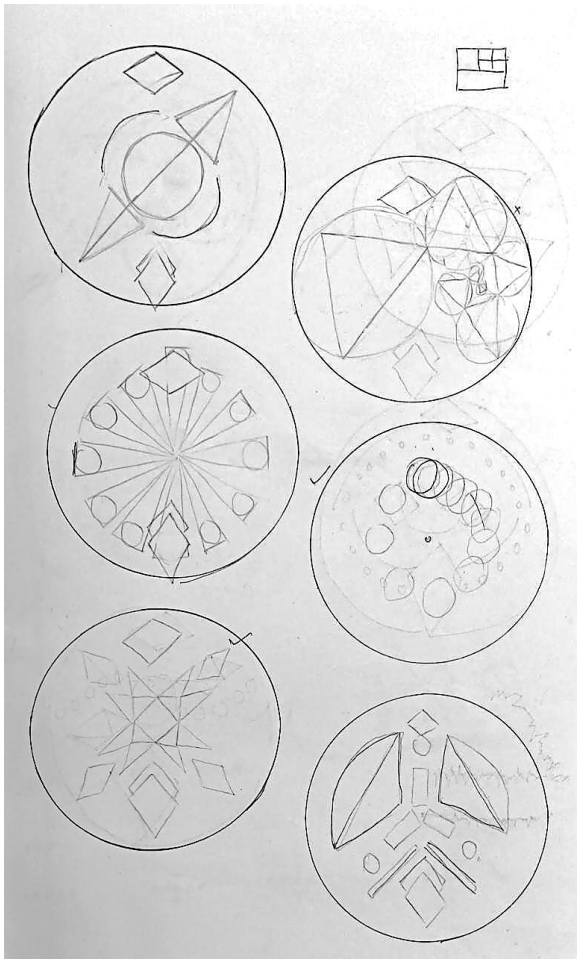
```
circle(212.5-400,0,70);
circle(212.5-400,0,50);
circle(587.5-400,0,70);
circle(587.5-400,0,50);
pop();

line(400-125,400-125,2
10,400);
line(210,400,400,650);
line(400,650,590,400);
line(590,400,400+125,4
00-125);
```

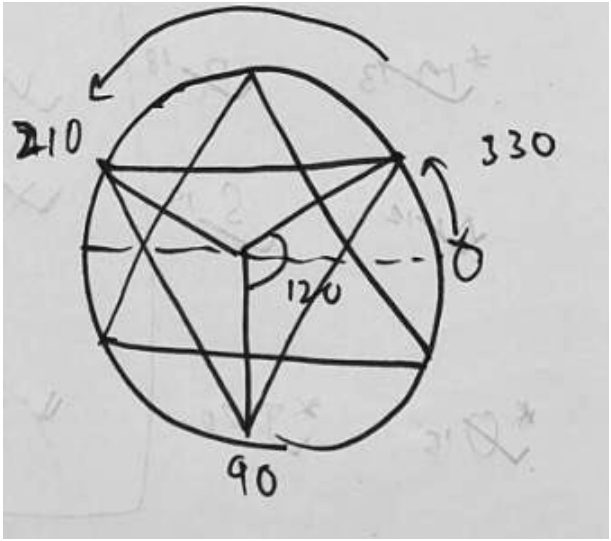
Eventually, i realised that brainstorming like this would take a lot of time and i switched to pen & paper. By this time i had realised what freedom javascript gave me as well.









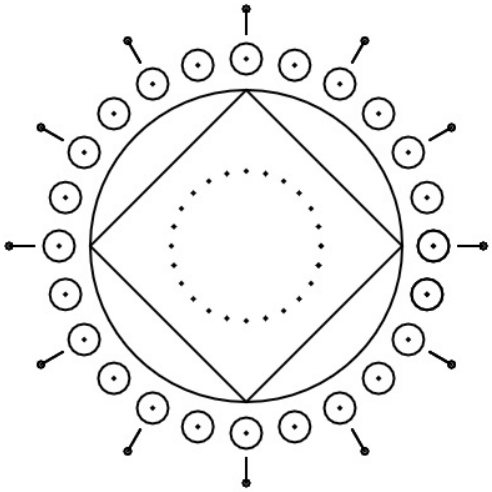


Coding It Out

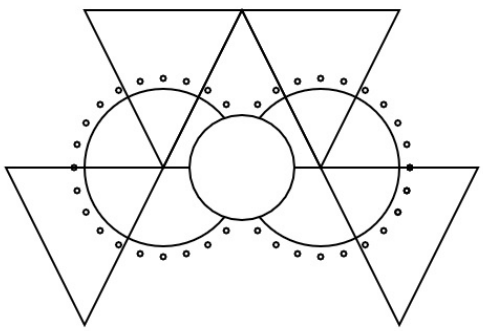
After ideating, i got to actually coding all those ideas. This involved a lot of math. I had to revisit a lot of 10th grade linear equation and circle formulas.

This article by **Programming Design Systems** helped a lot as well : https://programmingdesignsystems.com/shape/custom-shapes/index.html

In the end, the code ended up being quite complex for many intricate designs. But i was satisfied as they all became a part of a minimal theme.

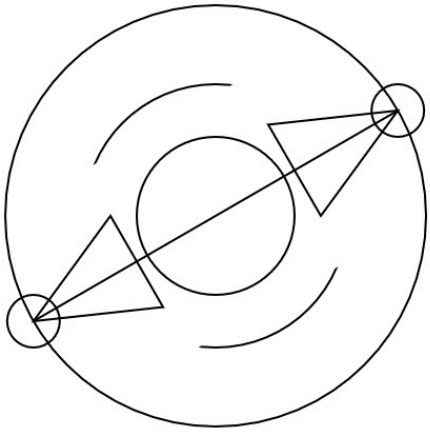


```
var a=0; var b=0;
      for(var i = 0;
i<26;i++){
         noFill();
         stroke(scolor);

circle(400+150*cos(a),4
00+150*sin(a),25);
         fill(bcolor);

circle(400+150*cos(a),4
00+150*sin(a),2);

circle(400+60*cos(a),40
0+60*sin(a),2);

circle(400+190*cos(b),4
00+190*sin(b),5);

line(400+170*cos(b),40
0+170*sin(b),400+190*c
os(b),400+190*sin(b));
         a=a+15;
         b=b+30;
      }
               noFill();

quad(400-125,400,400,
400+125,400+125,400,4
00,400-125);

circle(400,400,250);
```

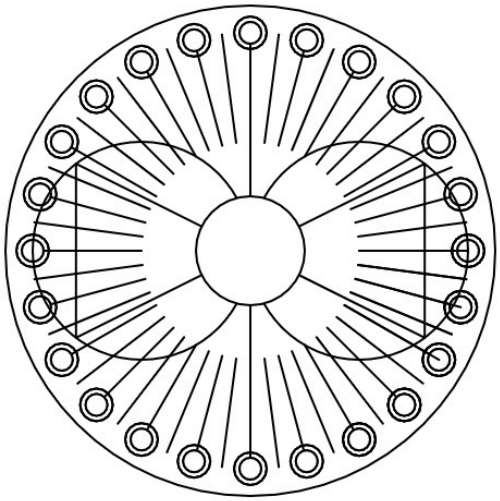


```
 var a=0; var b=0;
      for(var i = 0;
i<=26;i++){
      noFill();
      stroke(scolor);


circle((400-75)+85*cos
(a),(400)+85*sin(a),5);

circle((400+75)+85*cos
(a),(400)+85*sin(a),5);
      a=a+15;
      }

      fill(bcolor);
      stroke(scolor);

circle(400-75,400,150);

circle(400+75,400,150);

      noFill();
      stroke(scolor);
```

```
triangle(400-75,400,40
0+75,400,400,400-150)
;

triangle(400-75,400,40
0,400-150,400-150,40
0-150);

triangle(400+75,400,40
0,400-150,400+150,40
0-150);

triangle(400-75,400,40
0-150-75,400,400-150,
400+150);

triangle(400+75,400,40
0+75+150,400,400+150,
400+150);

      fill(bcolor);
      stroke(scolor);

circle(400,400,100);
```



```
noFill();
      stroke(scolor);

circle(400,400,250);
      fill(bcolor);
      noStroke();
      quad(400 + 200 *
cos(270+30),400+200*s
in(270+30),400 + 200 *
cos(270+60+30),400+2
00*sin(270+60+30),400
+ 200 *
cos(90+30),400+200*si
n(90+30),400 + 200 *
cos(90+60+30),400+20
0*sin(90+60+30));
      stroke(scolor);

circle(400,400,150);
      triangle(400 + 100
*
cos(270+30),400+100*s
in(270+30),400 + 200 *
cos(270+30+30),400+2
00*sin(270+30+30),400
+ 100 *
cos(270+60+30),400+10
0*sin(270+60+30));
      triangle(400 + 100
*
cos(90+30),400+100*si
n(90+30),400 + 200 *
cos(90+30+30),400+20
0*sin(90+30+30),400 +
100 *
cos(90+60+30),400+10
0*sin(90+60+30));
      line(400 + 200 *
cos(270+30+30),400+2
00*sin(270+30+30),400
+ 200 *
cos(90+30+30),400+20
0*sin(90+30+30));
      noFill();
      circle(400 + 200 *
cos(270+30+30),400+2
00*sin(270+30+30),50)
;
      circle(400 + 200 *
cos(90+30+30),400+20
0*sin(90+30+30),50);

circle(400,400,400);
```



```
 var a=0;var b=0;
      for(var i = 0;
i<=52;i++){
          fill(bcolor);

circle(400+200*cos(a),
400+200*sin(a),20);

          noFill();
          stroke(scolor);

circle(400+200*cos(a),
400+200*sin(a),30);

line(400+100*cos(b),40
0+100*sin(b),400+200*
cos(b),400+200*sin(b))
;

          a=a+15;
          b=b+7.5;
      }
       noFill();
      stroke(scolor);

circle(300,400,200);

circle(400,400,450);

circle(500,400,200);
      line(400,400,300
+ 100 *
cos(180+53),400+100*s
in(180+53));
line(400,220,400,400);

line(400,400,400,580);
      line(400,400,500
+ 100 *
cos(53),400+100*sin(5
3));
      line(300 + 100 *
cos(180-53),400+100*s
in(180-53),500 + 100 *
cos(-53),400+100*sin(
-53));
      line(300 + 100 *
cos(180+53),400+100*s
in(180+53),300 + 100 *
cos(180-53),400+100*s
in(180-53));
      line(500 + 100 *
cos(53),400+100*sin(5
3),500 + 100 *
cos(-53),400+100*sin(
-53));

      fill(bcolor);
      stroke(scolor);

circle(400,400,100);
```

The Other Features

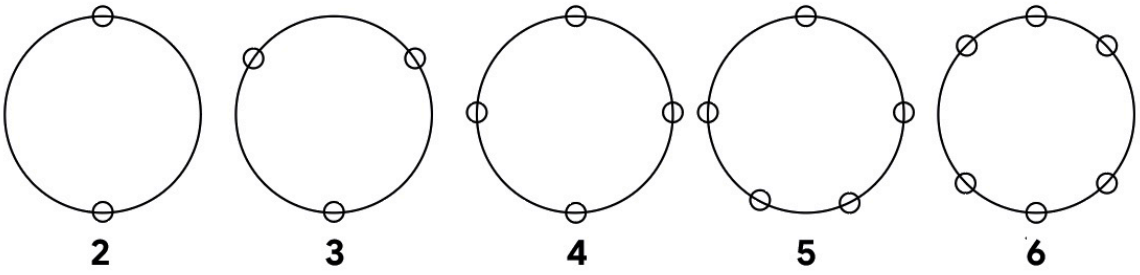
Deciding the other features was a complicated process, as they had to look in place with all the shapes.

The Central Dots

I had decided to involve color as less as i could, this was because i wanted all the generated identities to have a certain sense of similarity when looked over at collectively. This led me to the central dots. They are assigned according to the number of letters in the name.

The Orbiting Circles

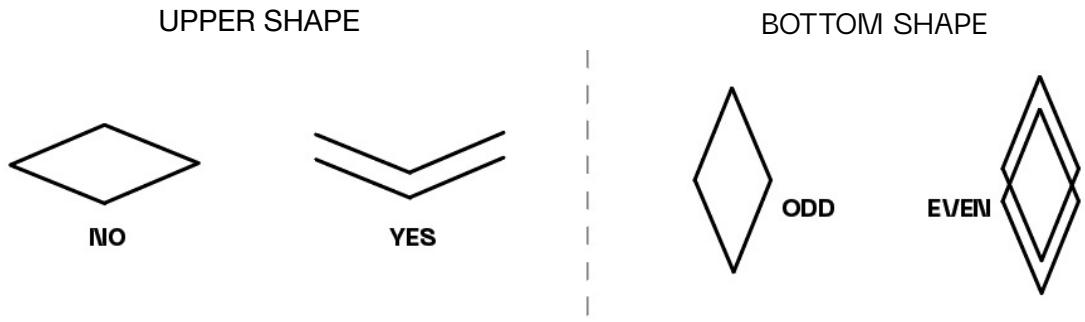
I wanted to experiment with a bit of animation in code as well but at the same time i did not want it to overwhelm the visual either, so the outer orbiting circles are the only part of the visual that are supposed to move. These are assigned according to the number of vowels in the name.




```
else if(countVowel==6)
  {
  stroke(scolor);
  noFill();
  var x = centerX + radius * cos(angle6_1);
          var y = centerY + radius *
sin(angle6_1);
  ellipse(x, y, r, r);
   var x2 = centerX + radius * cos(angle6_2);
          var y2 = centerY + radius *
sin(angle6_2);
 ellipse(x2, y2, r, r);
    var x3 = centerX + radius * cos(angle6_3);
          var y3 = centerY + radius *
sin(angle6_3);
  ellipse(x3, y3, r, r);
    var x4 = centerX + radius * cos(angle6_4);
          var y4 = centerY + radius *
sin(angle6_4);
 ellipse(x4, y4, r, r);
    var x5 = centerX + radius * cos(angle6_5);
          var y5 = centerY + radius *
sin(angle6_5);
  ellipse(x5, y5, r, r);
     var x6 = centerX + radius *
cos(angle6_6);
          var y6 = centerY + radius *
sin(angle6_6);
  ellipse(x6, y6, r, r);

  angle6_1 = angle6_1 + speed*(180/PI);
  angle6_2 = angle6_2 + speed*(180/PI);
  angle6_3 = angle6_3 + speed*(180/PI);
  angle6_4 = angle6_4 + speed*(180/PI);
  angle6_5 = angle6_5 + speed*(180/PI);
     angle6_6 = angle6_6 + speed*(180/PI);
  }
```

Upper & Bottom Shapes

Diamonds were my go-to for these as they complemented all the central shapes very well. The criteria for the upper shape was if the name started and ended with the same letter. For the bottom shape it was whether there were odd/even number of letters in the name.



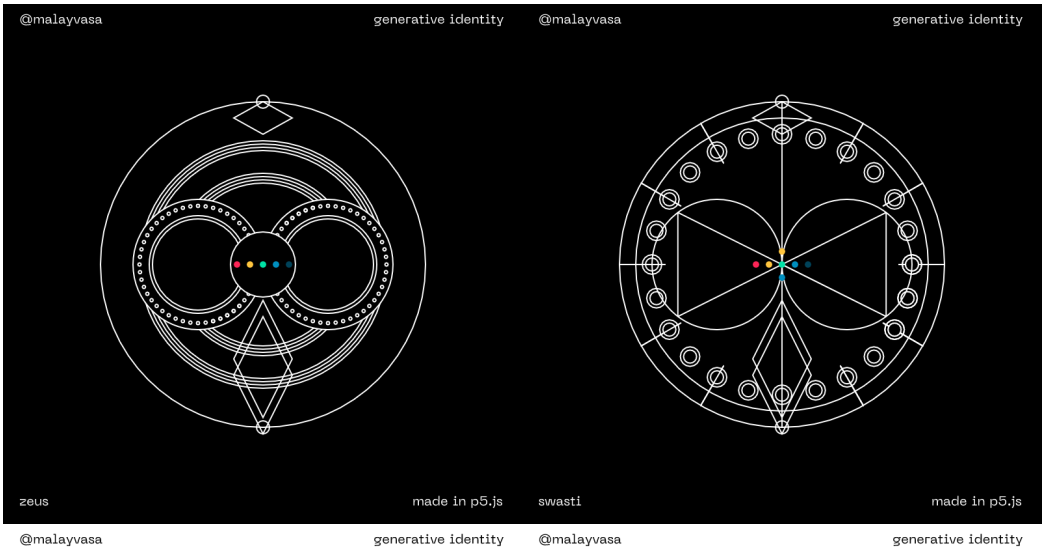
The Color Scheme

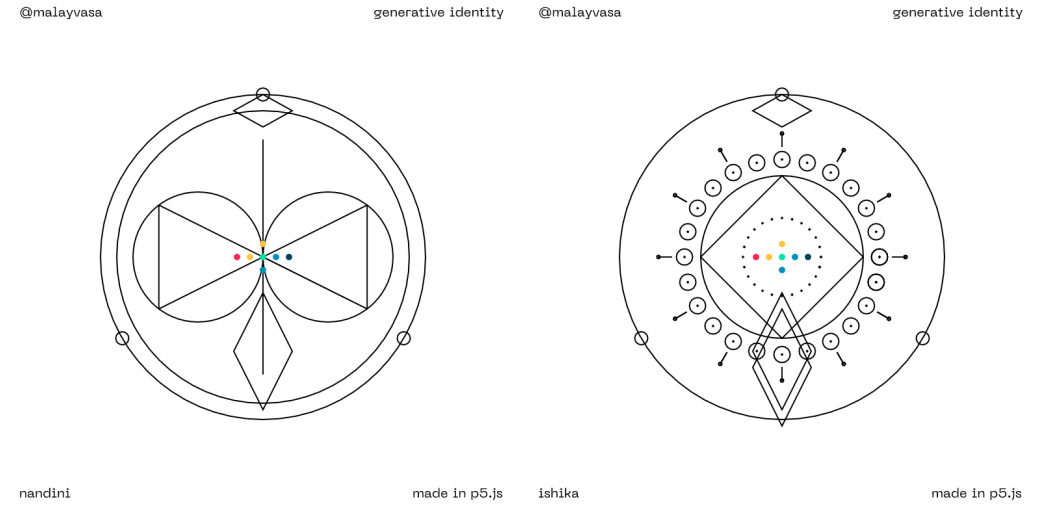
This feature was added at a very late stage, the identities felt as if they were lacking something. These helped to spice things up a little bit. These are chosen by counting the number of letters in the name that occur alphabetically before M (including M) and after M.

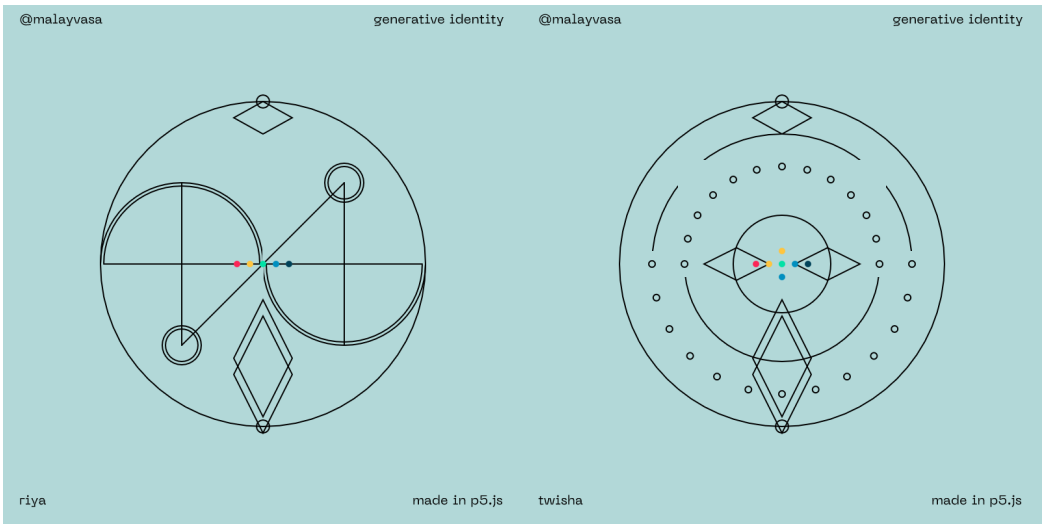

```
for(var j=0; j<name.length ;j++){
      value=name[j].charCodeAt() - 64;

   if(value<=(77-64))
     sCount++;
   else
     mCount++;

   }

   if(sCount>mCount)
  {document.body.style.background = "white";
     background("white");
     scolor = "black";
   bcolor="white";
   document.body.style.color = "black";
   }else if(mCount>sCount){
     background("black");
     scolor = "white";
     bcolor="black";
     document.body.style.background =
"black";
     document.body.style.color = "white";
   }else{
     background("#b2d8d8");
     scolor = "black";
     bcolor = "#b2d8d8";
     document.body.style.background =
"#b2d8d8";
     document.body.style.color = "black";
   }
```

The Final Outcome

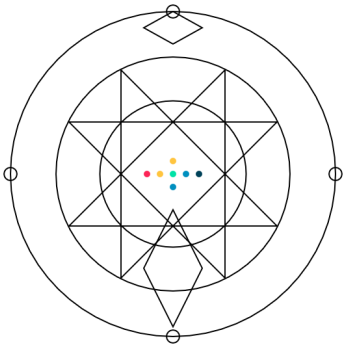
All of these features, over 1100 lines of code and names came together to form the following visuals.





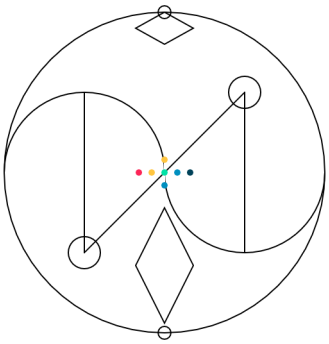


@malayvasa generative identity



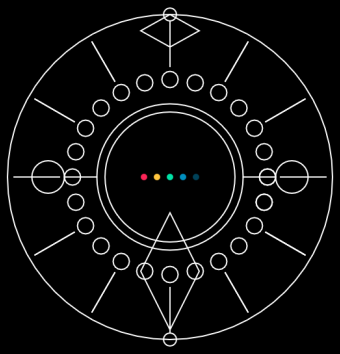
lareina made in p5.js

@malayvasa generative identity



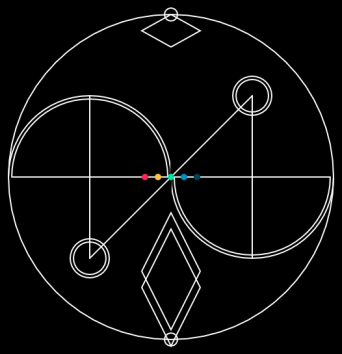
darshan made in p5.js

@malayvasa generative identity



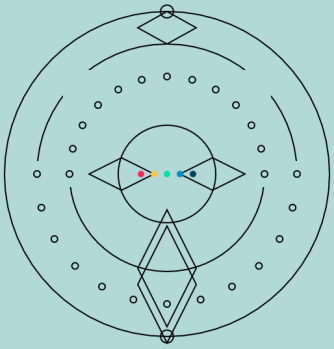
aswin made in p5.js

@malayvasa generative identity



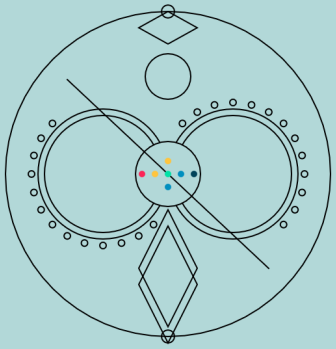
rose made in p5.js

@malayvasa generative identity



tara made in p5.js

@malayvasa generative identity

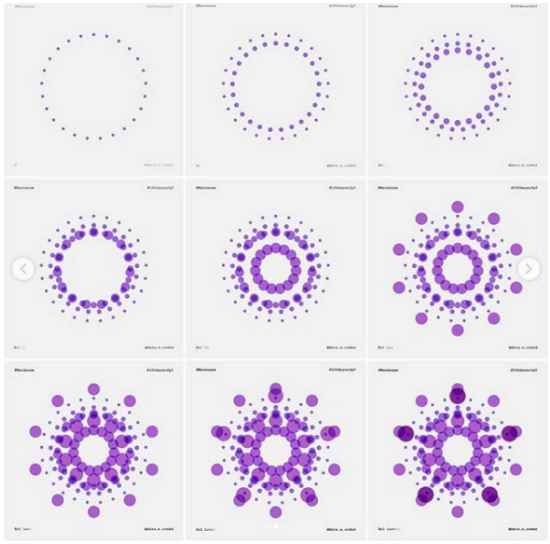


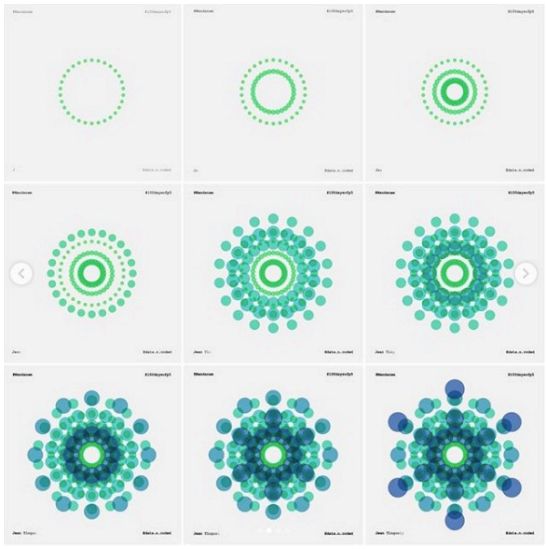
vedant made in p5.js

Bibliography

Inspiration :

1) Rasagy Sharma Principal Designer at Gramener
Data 'n' Coded Project – Mandalanaam
Generating Mandalas from artist's names.





2) Aishwarya Anand Singh – Name Constellation Project
Generating star constellation from names

Links :

Name Constellation inspiration –
https://www.behance.net/gallery/84799849/Generative-Identity-Name-Constellation

Data 'n'Coded inspiration –
https://www.instagram.com/data.n.coded/

javascript tutorials –
https://www.w3schools.com/js/default.asp

p5.js tutorials –
https://p5js.org/reference/

Neue Machina Font –
https://www.behance.net/gallery/78376829/Neue-Machina-Free-Typeface