CODEGUPPY PROJECTS

BOOKLET

Nazia Fakhruddin

Project 1- Holographic Fun

In this fun project we will learn about placement and rotation of the sprite at different positions and angles to create a holographic illusion come alive inside a bottle.

Create a new project using CodeGuppyPlayground.Start by giving the background() color.

```
background(0);
```

Next, create a global variable of **'p'.** Now use this variable to call the sprite of your choice, we will need the same sprite at four different positions. In this case ,it is a 'Santa' at x-axis of 400 and y-axis of 100 , keep the size 0.36.Our sprite appeared at the *top*. Using the show(), select the movement you want the sprite to do.

```
var p;
p=sprite('santa',400,100,0.36) ;
p.show('run');
```

To rotate the sprite use rotation and give it the angle for example 0,90,180 or 270. Observe when you change the angle your sprite should rotate to a new angle.

```
p.rotation=0;
```

In the same way ,let's make the sprite appear at the *right hand side*.First make a new variable **'q'** . Now call the sprite using the variable q at x-axis of 650 and y-axis of 300.

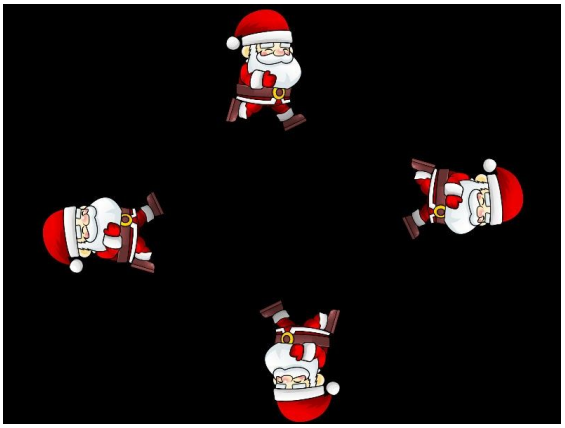
```
q=sprite('santa',650,300,0.36) ;
```

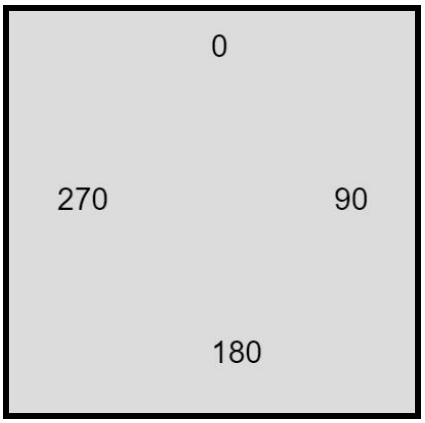
Rotate the sprite to **90 degrees**.

```
q.rotation=90;
```

Challenge :

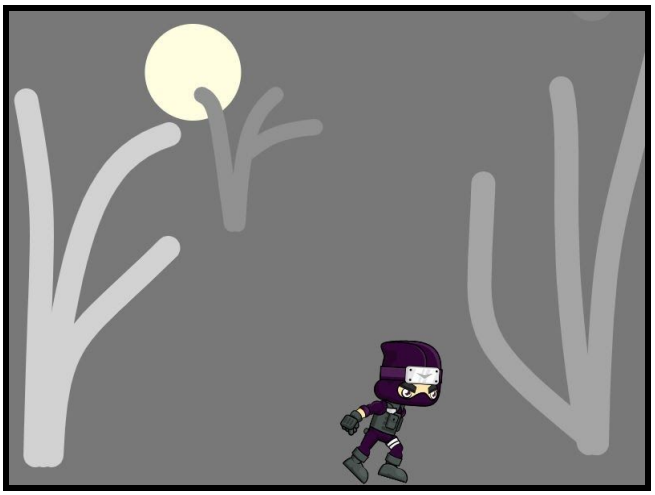
Try , making the sprite appear at the *bottom* and *left* positions.Hint: Make seperate variables .When positioning at bottom the x-axis remains the same as the top sprite . In the same way ,left sprite will have the same y-axis as that of sprite on the right.Rotate the sprites using the diagram below.





Finally , place the water bottle in the middle and look from the side .Congratulations!!! You made your very own Hologram.

Project 2- Interactive Forest with Bezier Curve

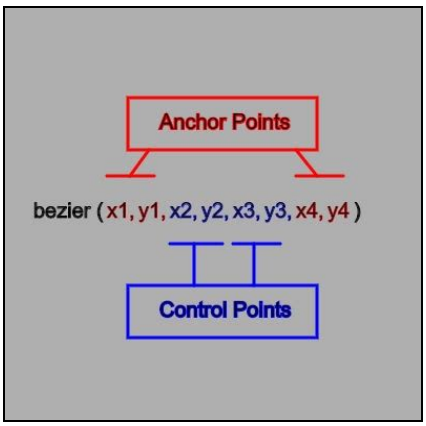


In this project we will be exploring how to use Bezier curve to create a scene and use mouse interaction to make it interactive.

Create a new project using CodeGuppyPlayground.Start by making a function loop() and give background() color.Now use clear() function to clear the canvas repeatedly to prevent smudging.

```
function loop() {
    clear();
    background('white');
}
```

How the bezier curve works?



These curves have ***anchor*** and ***control points.***The bezier function takes four sets of parameters .First and last sets of x-axis and y-axis are ***anchor points***,while the middle two sets are ***control points*** , responsible for the shaping smooth curves .

How to find the coordinates on the canvas?

The trick is to make them appear as text using ***text()*** . The text function takes three parameters for example : ***text(mouseX , x-axis , y-axis) .*** This the case , ***mouseX*** value we want to trace ,while the ***x*** and the ***y axis*** is the position of the text on the canvas. As we move around the mouse these values can be seen changing , identifying the position of the mouse.

```
text(mouseX , 300 , 30);

text(mouseY , 360 , 30);
```

Making a Tree:

Let's start by making the first bezier curve in the function loop(), give it stroke(color) , strokeWeight(thickness) and noFill().You can try making some more branches.

```
stroke( 210 );
strokeWeight( 30 );
noFill();
bezier( 48 , 555 , 81 , 273 , 112 , 186 , 179 , 157 );
bezier( 38 , 555 , 44 , 266 , 57 , 277 , 1 , 115 );
bezier( 58 , 555 , 65 , 400 , 90 , 410 , 178 , 298 );
```

Challenge:

Try , making more trees using the bezier curves at different distances to make a forest.

Making the Trees sway with the Mouse Interaction:

To give swaying functionality to the trees make a local variable of **'m'** and use it to call the map() function . mouseX is use in the map() function and it's original range(which is ***0 to width***) is then remap to a new range (in this case 0-50) .

```
let m = map( mouseX , 0 , width , 0 , 50 );
```

Add the **m** variable in the x-axis of the tips of the branches.This will make the tree branches sway as you move the mouse across.

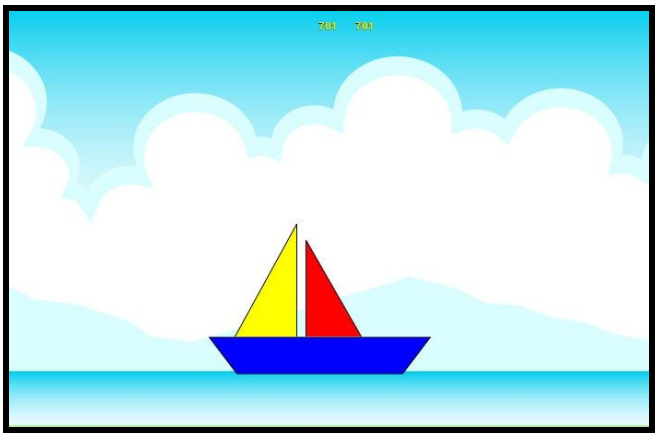
```
bezier( 48 , 555 , 81 , 273 , 112 , 186 , 179+m , 157 );
```

You can also create a moon using circle() function or add a sprite of your choice and give some velocity in x-axis to make it move across .

```
p.velocity.x = 0.9;
```

This will make the scene more realistic.**Happy Coding!!!.**

Project 3 - Sail Boat with Vertex



In this project, you will create a boat by using the vertex function and make the boat move.
Create a new project using CodeGuppyPlayground.Start by making a function loop() and give background() color.Now use clear() function to clear the canvas repeatedly to prevent smudging.

```
function loop() {
    clear();
    background(220);
}
```

How to find the coordinates on the canvas?

The trick is to make them appear as text using ***text()*** . The text function takes three parameters for example : ***text(mouseX , x-axis , y-axis) .*** This the case , ***mouseX*** value we want to trace ,while the ***x*** and the ***y axis*** is the position of the text on the canvas. As we move around the mouse these values can be seen changing,identifying the position of the mouse.

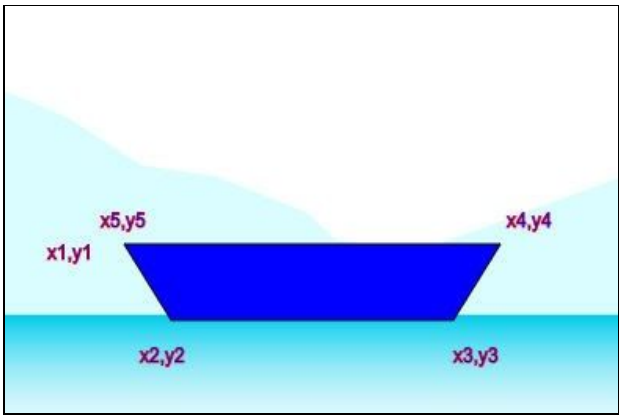
```
text(mouseX , 400 , 30);
text(mouseY , 440 , 30);
```

Next step, select a scene of your choice. In this case we have selected ‘Road’.

```
background('Road');
```

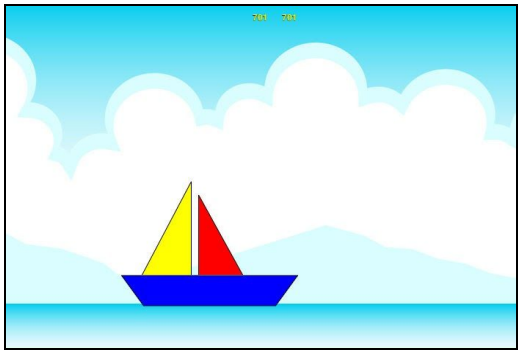
Making the Body and Sails of the Boat:

To make a boat using vertex() function, select the position with the help of the mouse. Then within beginShape() and endShape() , give the **vertex(x-axis , y-axis)** positions to make the body of the boat .




```
beginShape();
  // vertex(x1 , y1)
 vertex(80 , 410);
  // vertex(x2 , y2)
 vertex(110 , 455);
  // vertex(x3 , y3)
  vertex(290 , 455);
   // vertex(x4 , y4)
  vertex(320 , 410);
  // vertex(x5 , y5)
  vertex(80 , 410);
endShape();
```

Challenge:



Try, making the sails of the boat ,use the mouse positions to find the coordinates.

Make the Boat move across:

Final fun step, make global variable **'x'** and initiate it with **0.** Now increment it inside the function loop by **0.5**.

```
x = x+ 0.5;
```

Add the variable x, to the x-axis of each vertex of the body and sails of the boat.

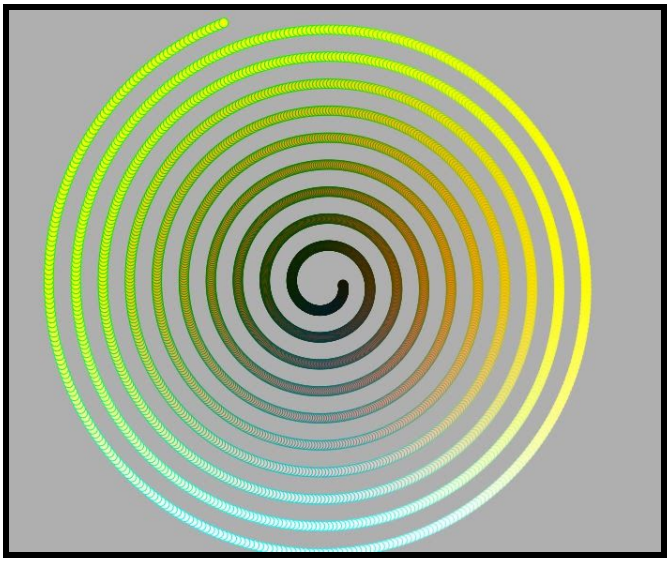
```
vertex(185+x,409);
vertex(185+x,290);
vertex(245+x,409);
```

Yay!! The boat is moving ,but once it reaches the end of the canvas the boat disappear.Oops!!. Let's make a conditional ,that once the boat reaches the maximum width ,make it reappear at the initial position, otherwise keep it moving.

```
if(x > width){
      x = 0;
  }else{
      x = x+ 0.5;
 }
```

Congratulations!!! It works.Enjoy making your own creations with vertex.

Project 4 - Trigonometric Art



In this project , you will learn to use trigonometric functions to make artwork.
Create a new project using CodeGuppyPlayground.Start by making a function loop() and give background() color.

```
function loop(){
        background(175);
}
```

Now , make a circle using ***ellipse(x-axis, y-axis, width, height)*** function in the middle of the canvas.Give some stroke() and fill().You will now see a circle in the middle of a canvas.

```
stroke(0);
fill(255 , 0, 0);
ellipse(width/2 , height/2 ,10 ,10);
```

Let's use the translate() function, which changes the origin of the canvas. Make the new origin ***width/2*** and ***height/2.***

```
translate( width/2 , height/2 );
```

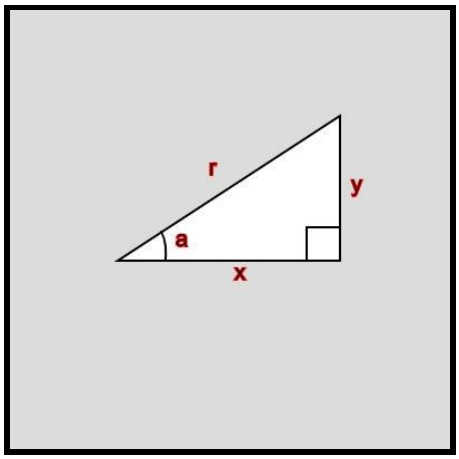
Now , replace the x-axis and y-axis in the ellipse with 0,0 because now the origin of the ellipse is controlled by the translate.

```
ellipse(0 , 0, 10, 10);
```

Place all the above code inside push() and pop() , which is meant to consolidate the code.

```
push();
translate(width/2,height/2);
stroke(0);
fill(255 ,0 ,0);
ellipse(0 , 0, 10, 10);
pop();
```

Using Trigonometry to make art:



Trigonometric Formula:

sin(a) = y / r

cos(a) = x / r

In order to make patterns ,we first need to find the x and y values, using the same trigonometric formula:

x = r * cos(a)
y = r * sin(a)

In this formula , we don't have the values of **'r'** for radius and **'a'** for angle **.** So, let's make a global variable of **'r'** and **'a'** give it a value of 20 and 0 respectively.

```
var r = 20;
var a = 0;
```

And initialize them inside the function loop().

```
r+= 0.08;
a+= 1.5;
```

Make the local variable of x and y .And calculate it using the same formula.

```
var x = r * cos(a);
var y = r * sin(a);
```

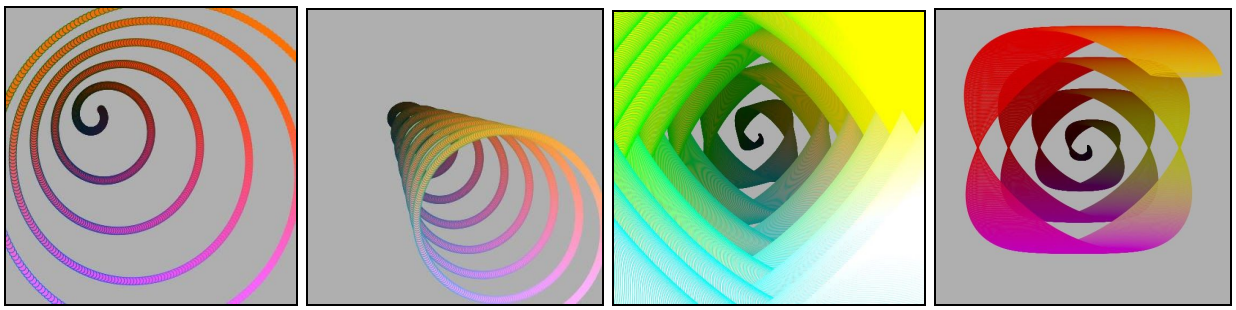
Finally, replace the x-axis and y-axis in ellipse with variables x and y.

```
ellipse( x , y , 10 , 10 );
```

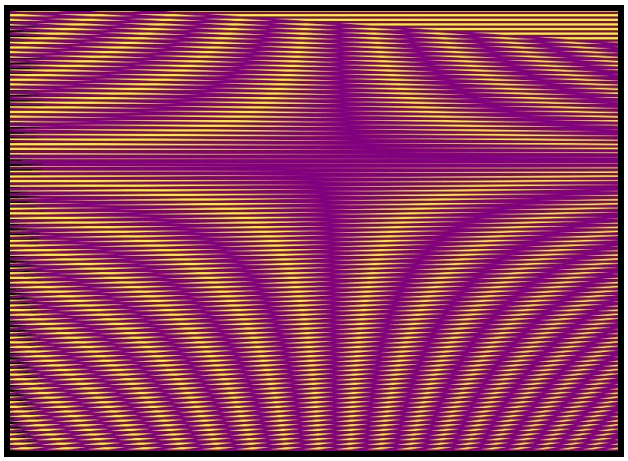
There you go your very own trigonometric artwork.Well Done!!!.

Challenge:

Try, making more patterns. *Hint*: Change the values of r and a. Replace the value of width and height in the ellipse(), with the variables x , y , r and a.You can use the same variables in the stroke() and fill() to create different colors.



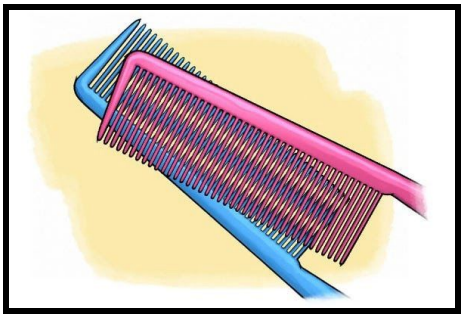
Project 5 - Moiré Pattern



In this project, you will be creating an optical illusion by the name Moiré Pattern.

What is “Moiré Pattern”? ‘ Moiré ’ is a French word meaning a silk fabric that has been subjected to heat and pressure rollers after weaving to give it a rippled appearance. The “Moiré Pattern” is an interference pattern produced by placing the similar templates , slightly offset by spacing or angle . These patterns can also be observed in photography and radiology, but as an artifact.In photography ,the odd stripes and pattern is called Moiré Effect .Some of the cameras now come with anti aliasing filter to remove the Moiré effect.

Experiment:



Let’s do a simple experiment, take two combs and place them over each other,and then move slowly

over each other.You observe the pattern appearing as move.

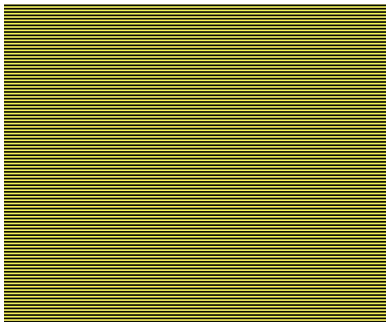
Let's try creating some optical illusion of Moiré Pattern ourselves.Create a new project using CodeGuppyPlayground.Start by making a function loop() and give background() color.

```
function loop(){
        background(0);
}
```

Inside function loop(), make a ***line(x1, y1 ,x2 , y2)*** in the center of the canvas.Now give the stroke() color and strokeWeight() that is the thickness of the stroke.

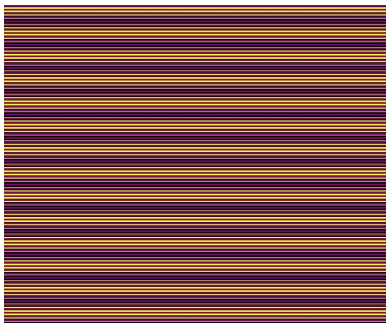
```
line( 0 , 200 , 0 ,width)
```

Now, make the **'for loop'** which will make multiple lines .Inside it initiate *variable 'i'* with zero and repeat 1000 times with a spacing of 6 in between .Put the ***line(x1, y1 ,x2 , y2)*** function with it properties inside it and replace the second and fourth parameters with the variable 'i' in the line() . This will appear on the canvas as multiple lines.



```
for (var i = 0; i < 1000; i += 6) {
      stroke('yellow');
      strokeWeight(3);
      line( 0 , i , width , i );
  }
```

Challenge:



Make another *for loop* , just like the previous one, with a spacing of 7 .(*Hint:* j+=7).Put the line() function in again .Replace the second and fourth parameter like before.

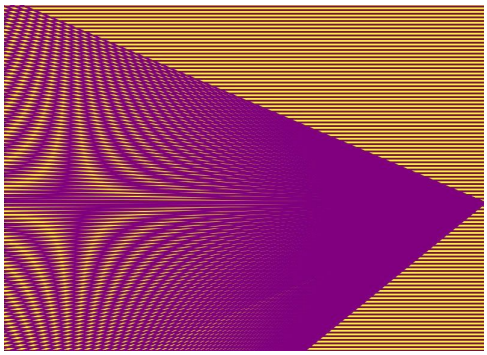
Make it dynamic:

This is not much fun to see, let's make a new global variable 'y' and initiate it with 0.

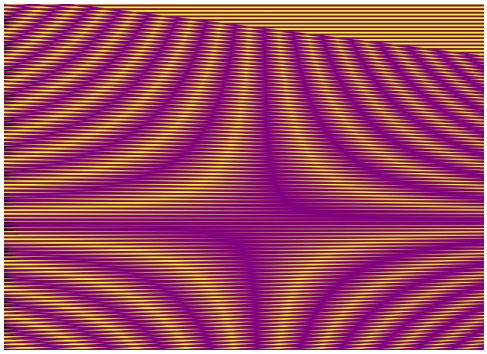
```
let y = 0;
```

In the function loop() , increment the variable 'y' by 1. Now use this variable 'y' to move the lines up and down by putting in the variable y in the fourth parameter of the second set of lines.It will make them move from one side .

```
line( 0 , j , width , y );
y = y + 1;
```



Finally, you can see it moving from up to down, making amazing patterns ,but it is not covering the whole canvas it seems to be tapered and in the corner.



To make it fill the canvas ,change the third parameter of the second set of lines to 3000 .

Now, to make it repeatedly changing , we need some help from conditionals . So, we will set the condition that if it becomes greater than the height of the canvas ,return to zero(or start) , else continue moving.

```
    if (y > height) {
      y = 0;
  } else {
      y = y + 1;
  }
```

Congratulations!! you made your very own mysterious Moire Pattern. Happy Coding!!

About the Author

Nazia has been working as a Computer Science Masterclass speaker in the Royal Institution of Britain for the past 2 years and as a CodeClub volunteer for the past 6 years. And has done her bachelors in Medicine and Surgery and is a self taught programmer. She is the author of the book “Superfun P5.js Projects ” and has written several multidisciplinary computer programming tutorials in the digital magazine Medium and her content is also available on Byte Size Coding website(https://bytesizecoding.com/) and YouTube channel .