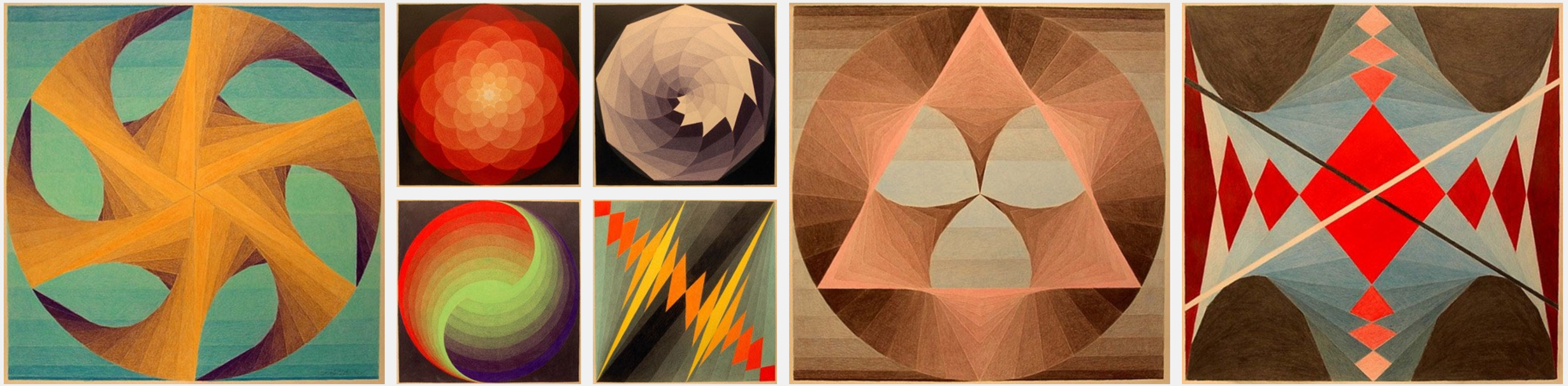


TRANSFORMATIONS, VARIATIONS

Week 7

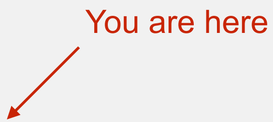
December 2019

Lior Ben-Gai

images from butdoesitfloat

COURSE TOPICS:

- Procedural drawing
- Functions
- Variables
- Conditionals
- loops
- **Transformations**
- Arrays, sound visualization
- Perlin Noise, trigonometry
- Data visualization
- Recursion
- Images, Off-screen buffers
- 3D, Noise, Vectors
- Intro to HTML
- Final Submission (30.01.2020)

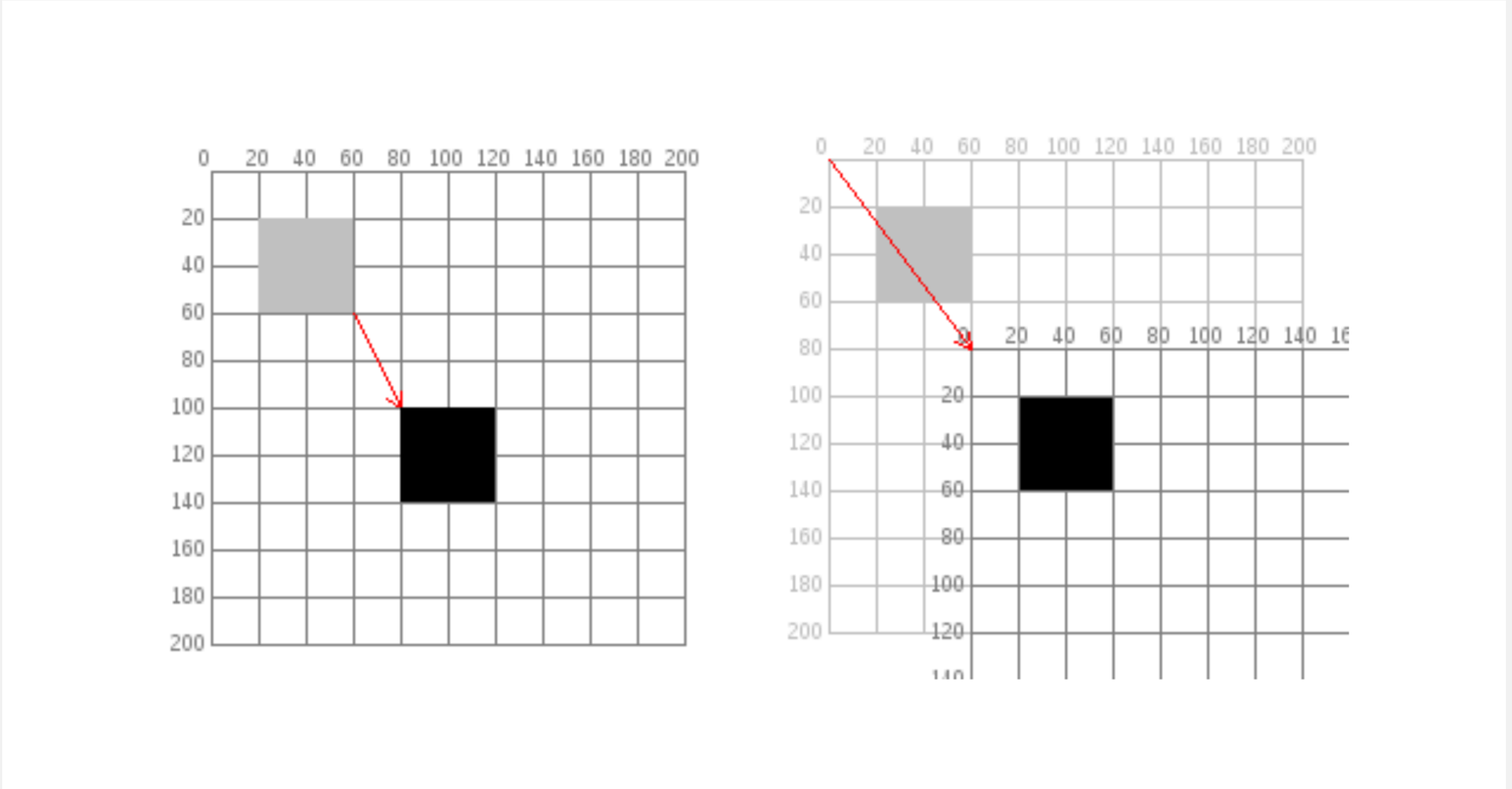


TRANSFORMATIONS

```
translate(x, y);
rotate(angle);
```

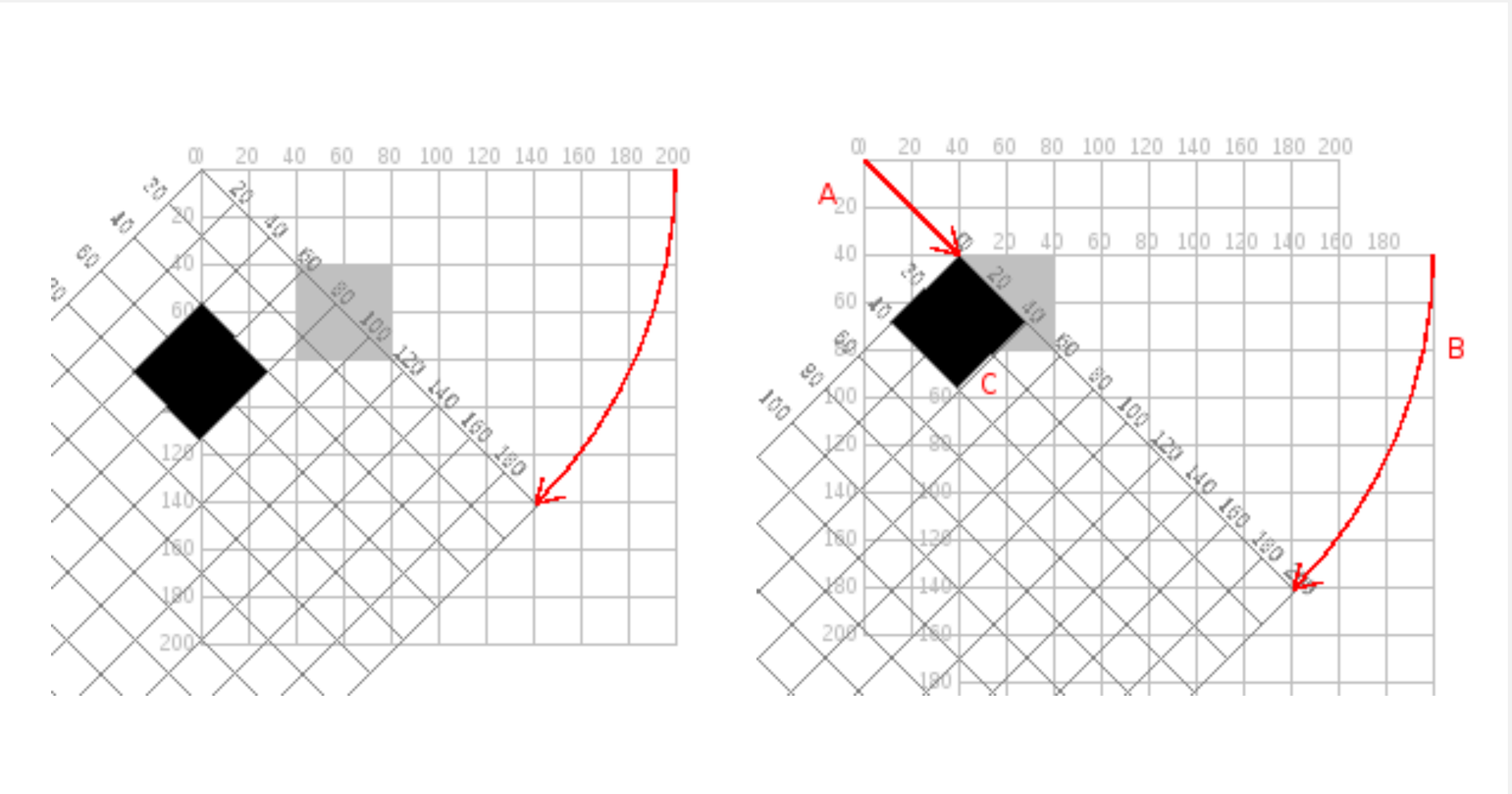
```
push();
pop();
```

TRANSFORMATIONS



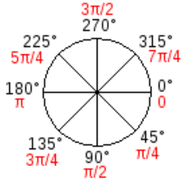

```
translate(x, y);
```

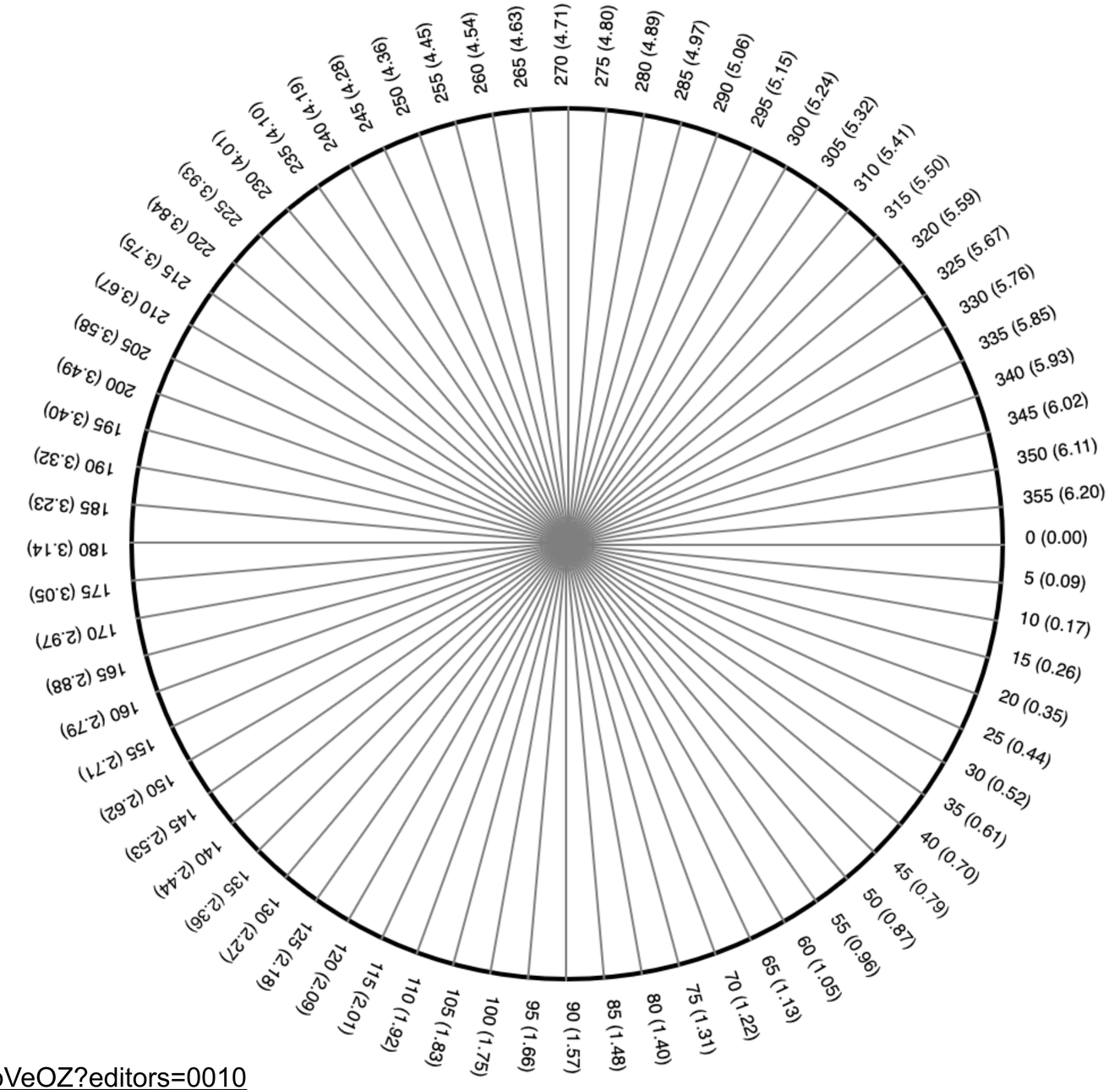
TRANSFORMATIONS




```
rotate(angle);
```

RADIANS





https://codepen.io/soogbet/pen/xxbVeOZ?editors=0010

TRANSFORMATIONS

```
push();
pop();
```

Push >> Place a piece of tracing paper on the sketch

define styles (colors, etc)

perform transformations (move the paper)

draw anything

Pop >> remove the paper (but keeps the stuff you drew of course)

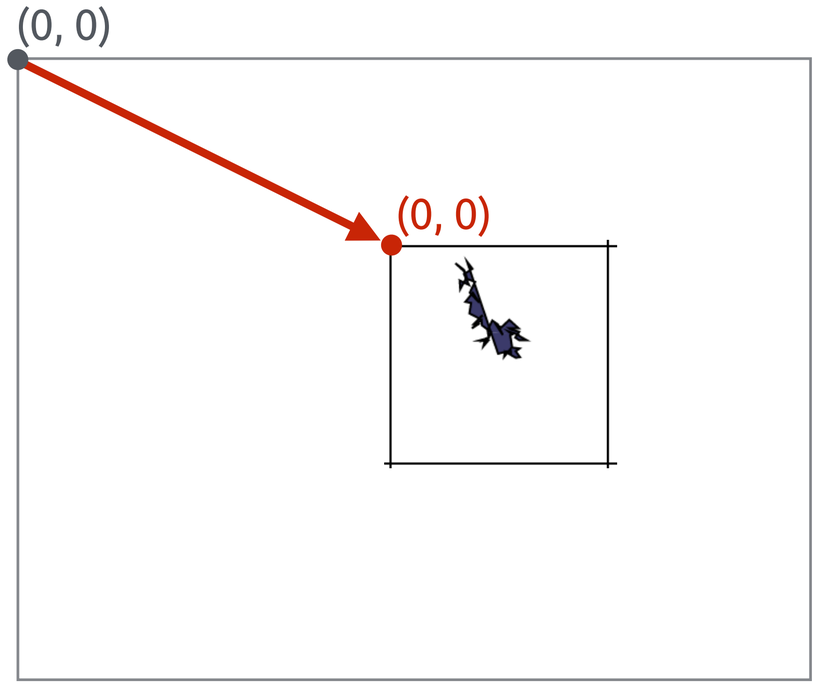
https://p5js.org/reference/#/p5/push

TRANSFORMATIONS

```
push();
    push();
        push();
            push();
                push();
                    push();
                    // SOME CODE HERE
                    pop();
                pop();
            pop();
        pop();
    pop();
pop();
```

push() and pop() commands may be nested

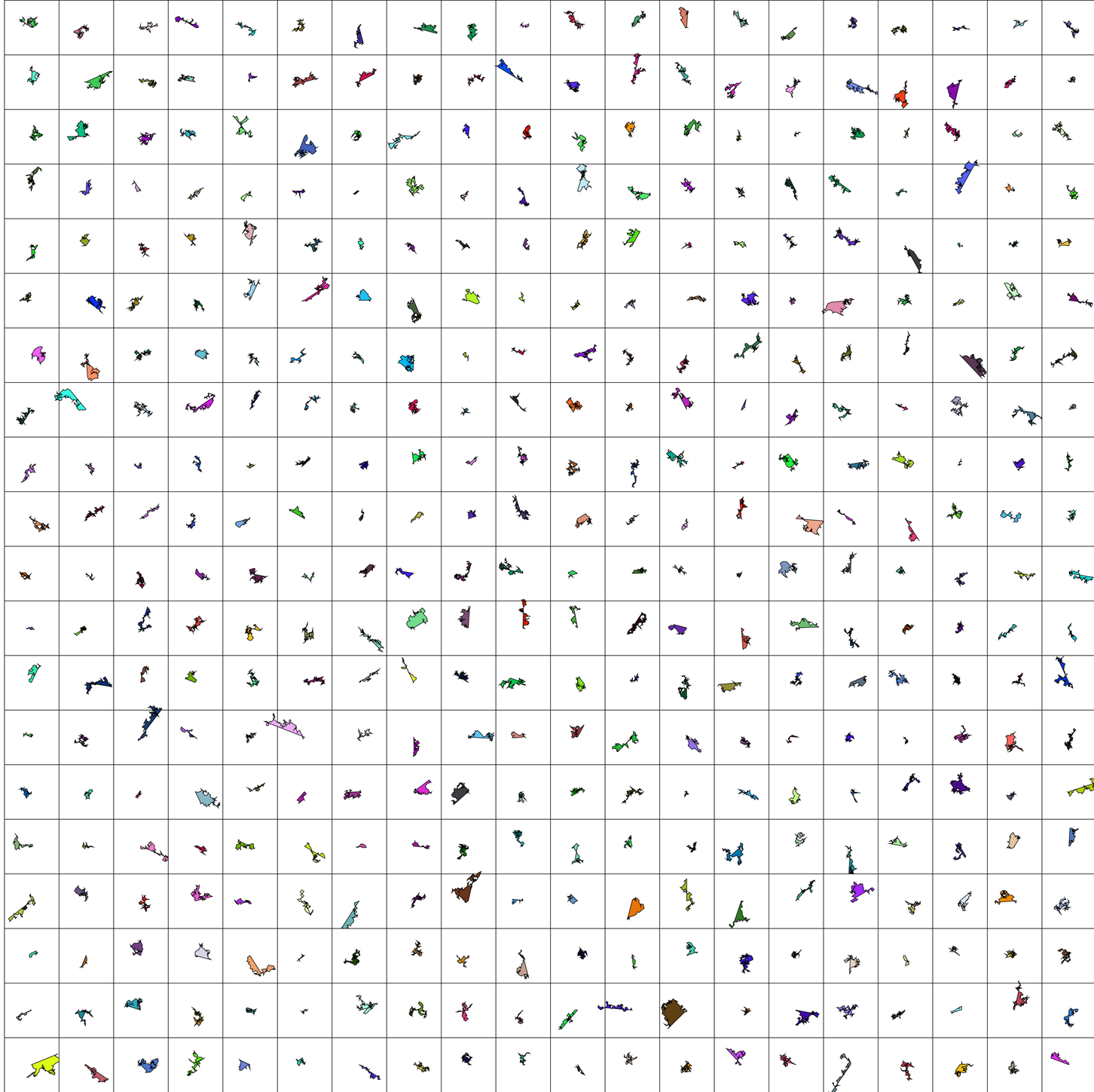
BUT DON’T PUSH IT (;



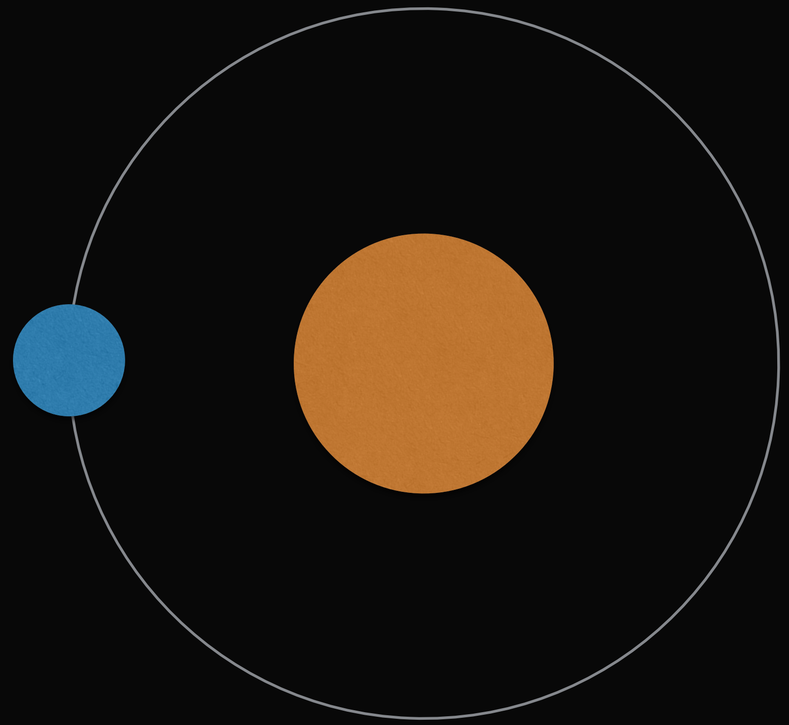
ORIGAMI

Transformations allow functions to draw independently

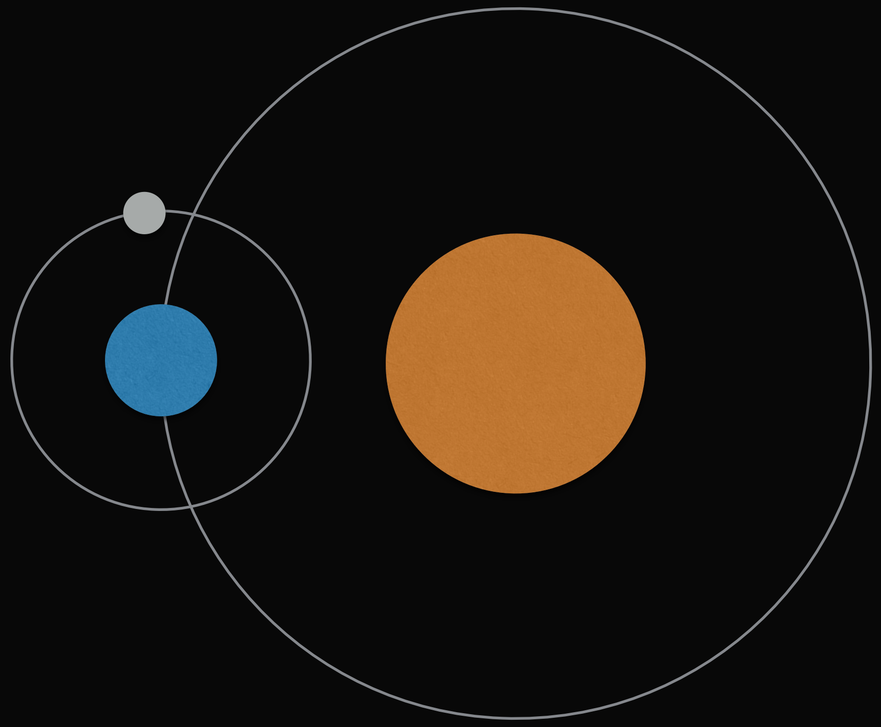
https://codepen.io/soogbet/pen/abbeOeY



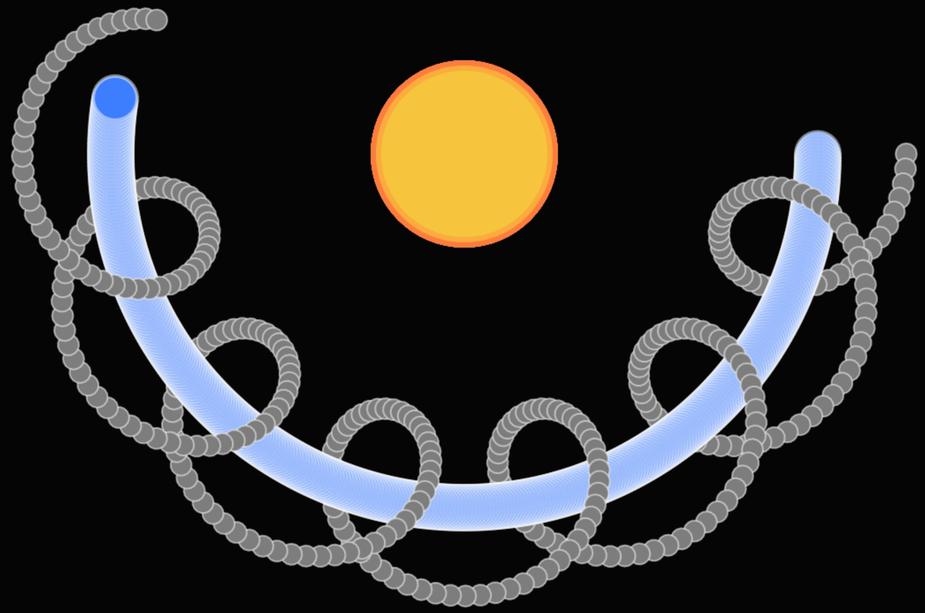
orbit



compounding orbits



compounding orbits



HOMEWORK

Please design and develop a digital Kaleidoscope.

- Emphasis on geometric detail and accuracy

- Go for visual impact (but don’t over-do it!)

- Your Kaleidoscope should constantly evolve and optionally take mouse input

https://en.wikipedia.org/wiki/Kaleidoscope