manual randomness → **coded system generation + data recording**

How can randomness be structured through system design, and what new visual outcomes emerge when generative rules replace intuitive choice?

Auto-refresh Random shapes generator-5 (different color)) by jinonw000 p5.js 1.11.5

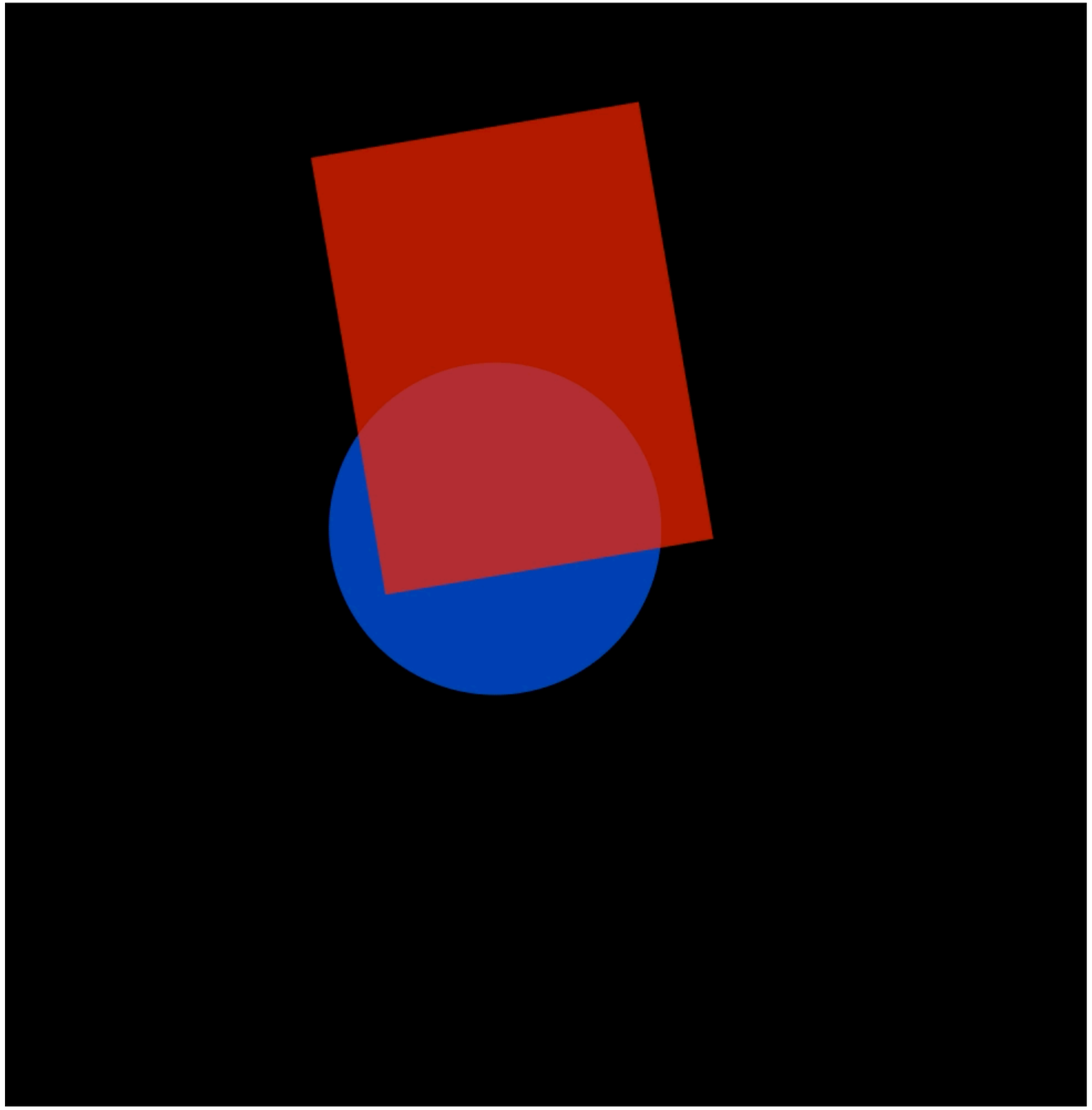
sketch.js Saved: about 10 hours ago

```
let shapes = [];
let colors = [];

function setup() {
  createCanvas(500, 500);
  background(0);
  frameRate(1);
  noStroke();

  // Define your 8 colors
  colors = [
    color(255, 39, 0, 180),
    color(0, 234, 255, 180),
    color(255, 0, 198, 180),
    color(0, 228, 43, 180),
    color(255, 138, 0, 180),
    color(142, 0, 177, 150),
    color(0, 96, 255, 180),
    color(255, 229, 7, 180)
```

Console Clear

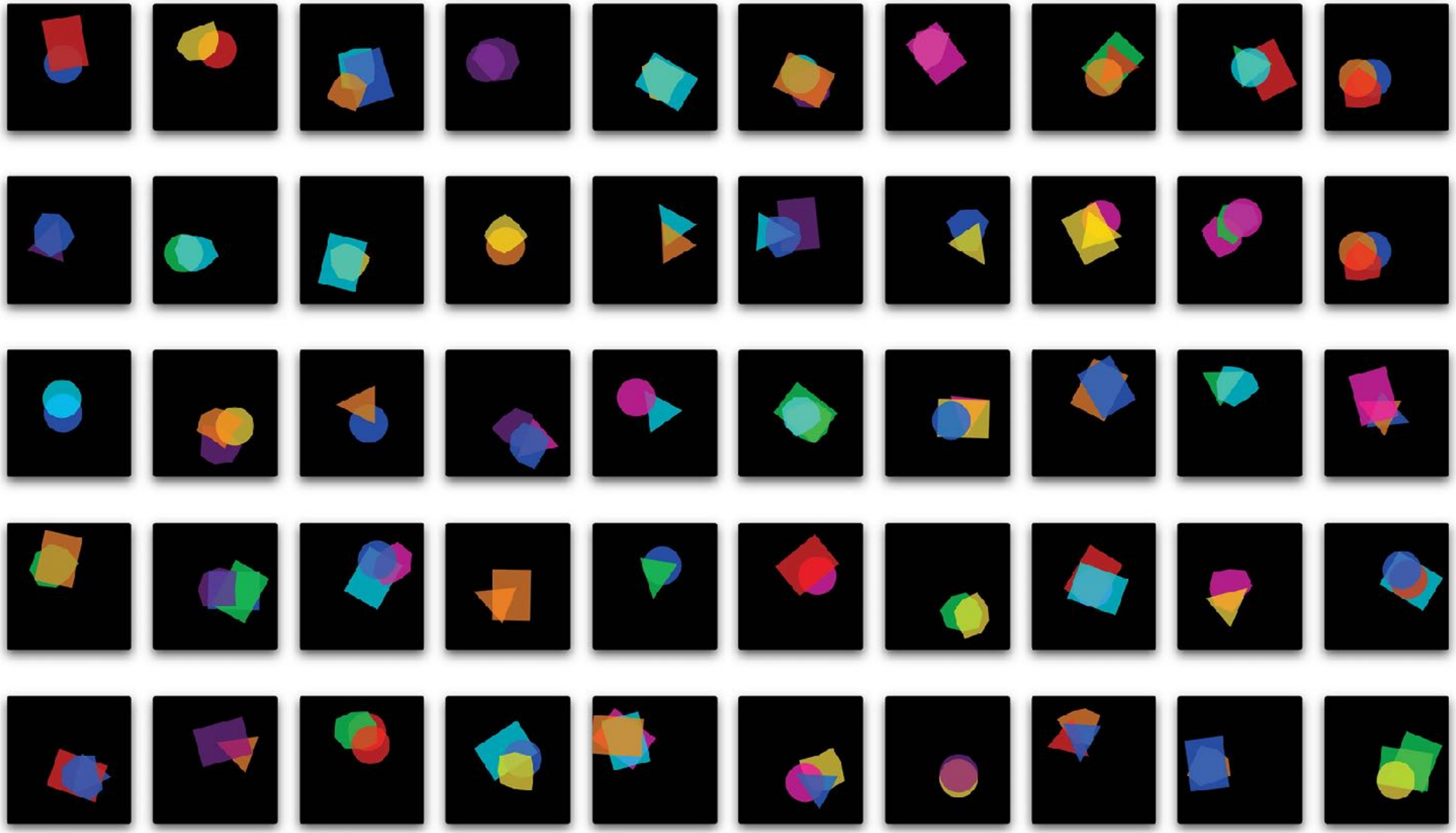


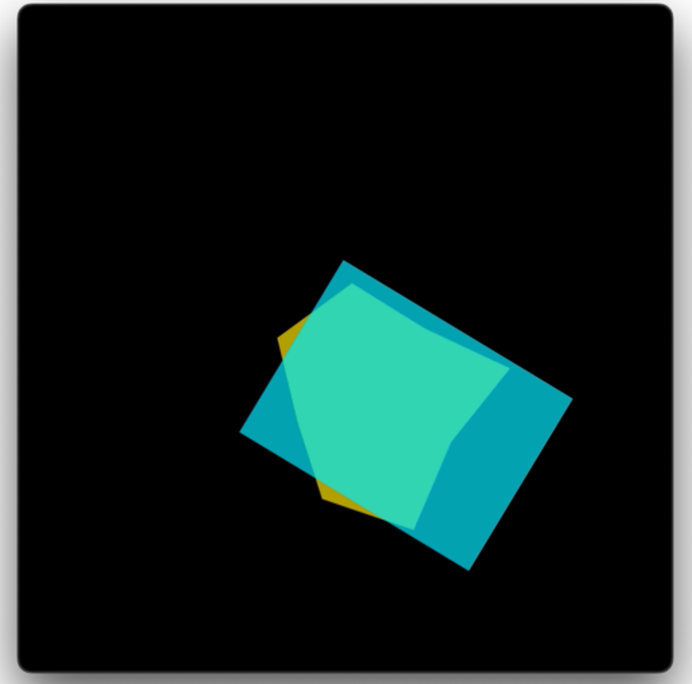
Colors used

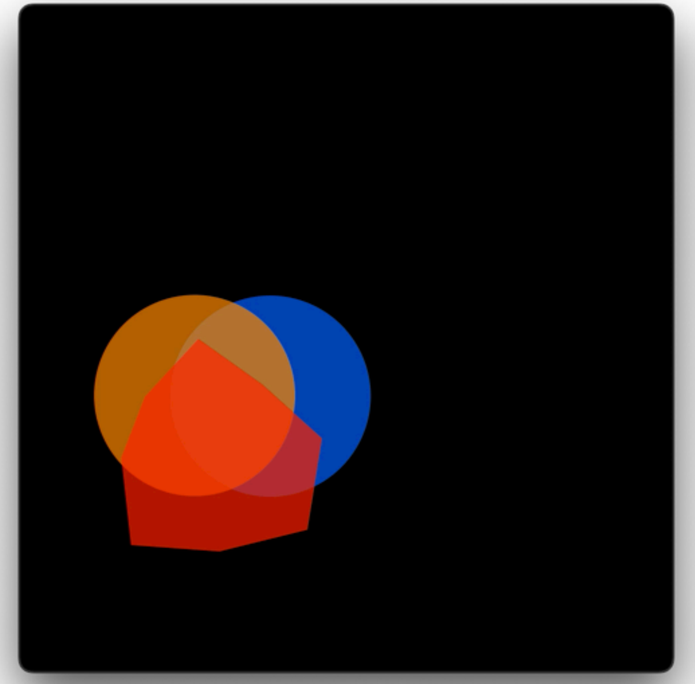
```
color(255, 99, 71, 180),
color(135, 206, 250, 180),
color(255, 182, 193, 180),
color(144, 238, 144, 180),
color(255, 165, 0, 180),
color(221, 160, 221, 180),
color(173, 216, 230, 180),
color(255, 215, 0, 180)
```

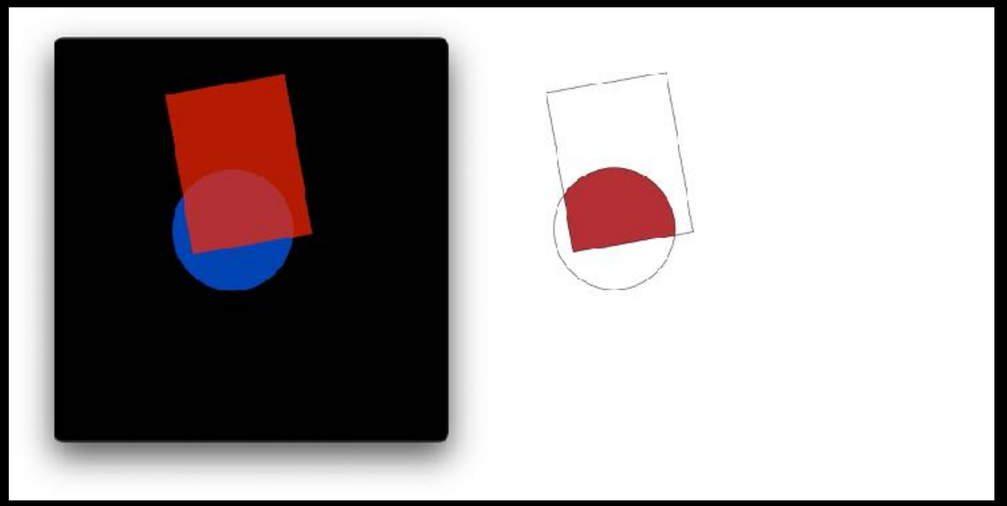
The second part of iterations was developed using p5js, which generated random combinations of shapes based on a defined set of rules. This system enabled the creation of 50 new compositions beyond my intuitive control. This method revealed emerging patterns of color interaction that were less predictable, yet more complex than those generated through manual composition.

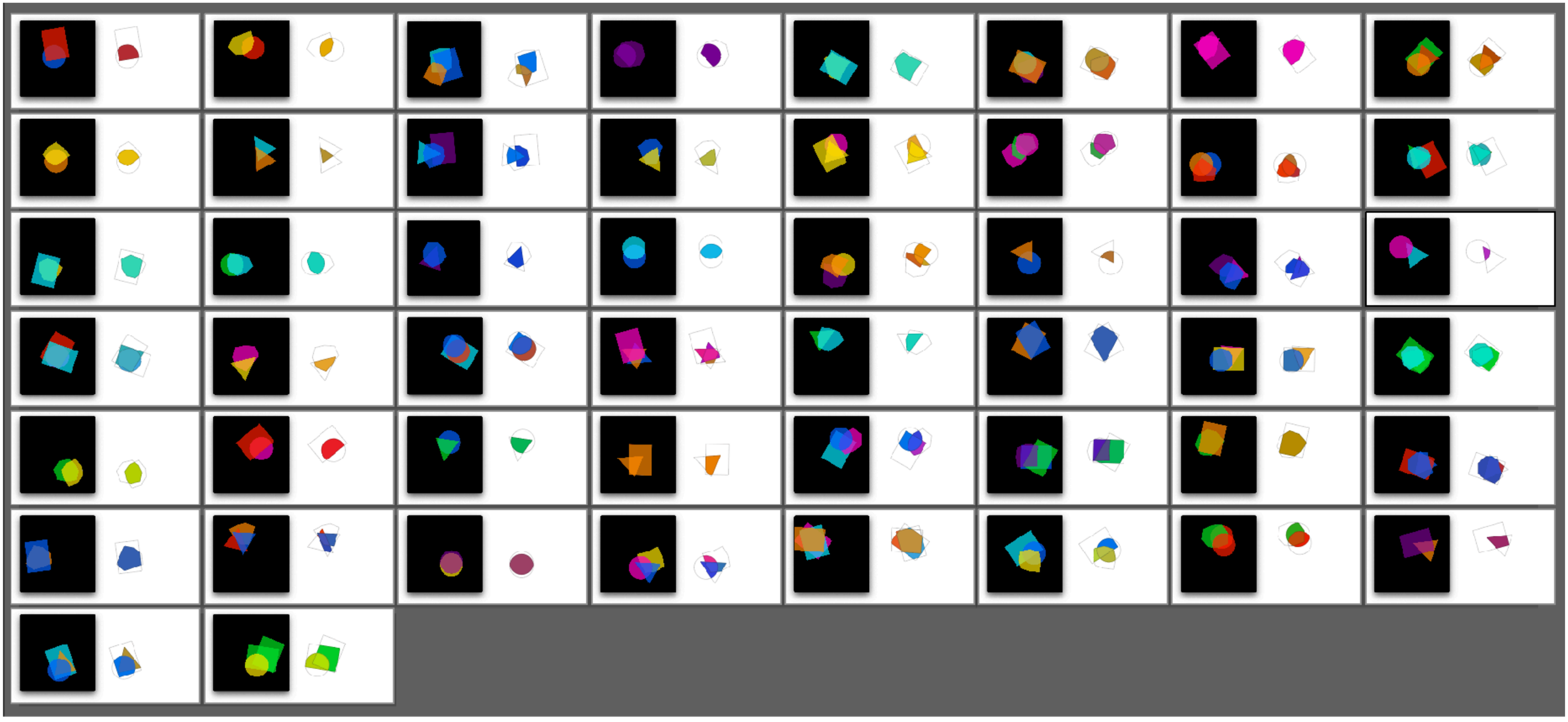
1. Randomly select a fixed number of shape fragments from a predefined pool.
2. Assign each shape a random position within the composition frame.
3. Randomly layer the selected shapes to allow overlaps.
4. Retain each shape’s original color and transparency.
5. Allow new colors to emerge visually through overlapping











New set of Colors

#b43035
#e8ab02
#0275e9
#b49334
#b48345
#b47533
#770093
#33d6b5
#ce601e
#bb923d
#ea00b5
#b34b06
#e97704
#b48f06
#09d8bd
#34adb4
#37a918
#b47533
#e93e0d
#eb3700
#b33035
#b42e96
#c52ea0
#32a246
#e8a32d

#0474f1
#b49334
#eac003
#33d6b5
#03d4bd
#1643d3
#04b9e9
#ea9100
#f0900a
#cca123
#ce601e
#b47533
#9f02a4
#1643d3
#2d43e3
#3543de
#b32fc1
#03d4bd
#02e3bf
#00d027
#e8a32d
#4672c1
#3374b4
#3561b4
#e8209c

#c81c8d
#0275e9
#345bc5
#b44d35
#e8a32d
#34adb4
#44acc1
#35a5dd
#b4d10d
#e91c25
#00b654
#ea7d00
#3452ed
#531cb2
#544b8a
#00b654
#b48f06
#344bb3
#b33035
#9d2767
#34a91b
#c54d07
#e92400
#0275e9
#b3d338

#b6c448
#b4b63a
#e86029
#c39341
#35a5dd
#e82e8d
#3543de
#4451dc
#3374b4
#9e4269
#eb3700
#4655b5
#3561b4
#b49334
#326dc3
#0275e9
#00d027
#b4df12
#b4d10d
#f9d212
#e9d005
#b4b63a
#1643d3

Initial 8 colors

- Systems and constraints can lead to more unexpected outcomes than freeform decisions.
- See how computational tools can expand creative outcomes.