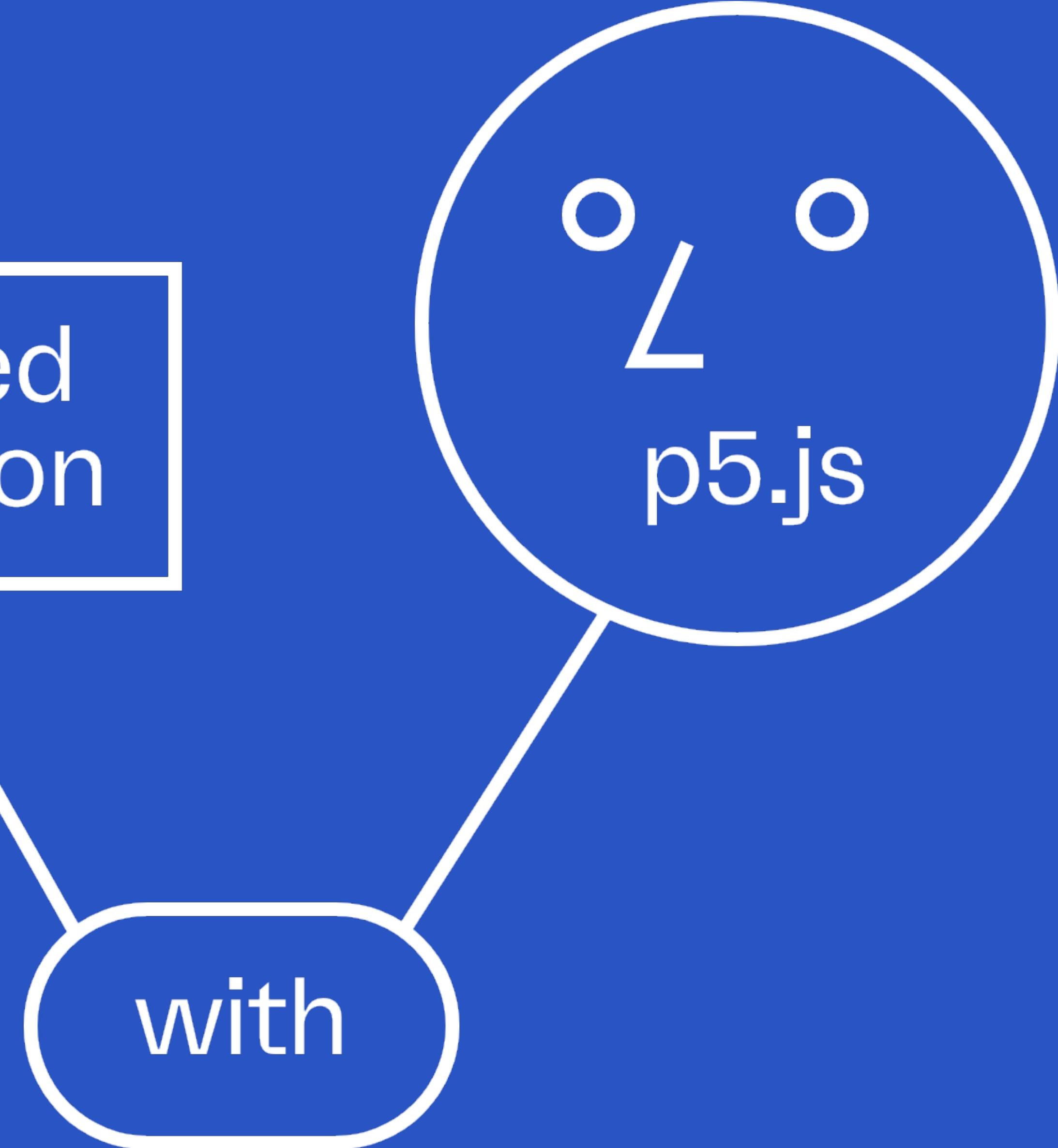


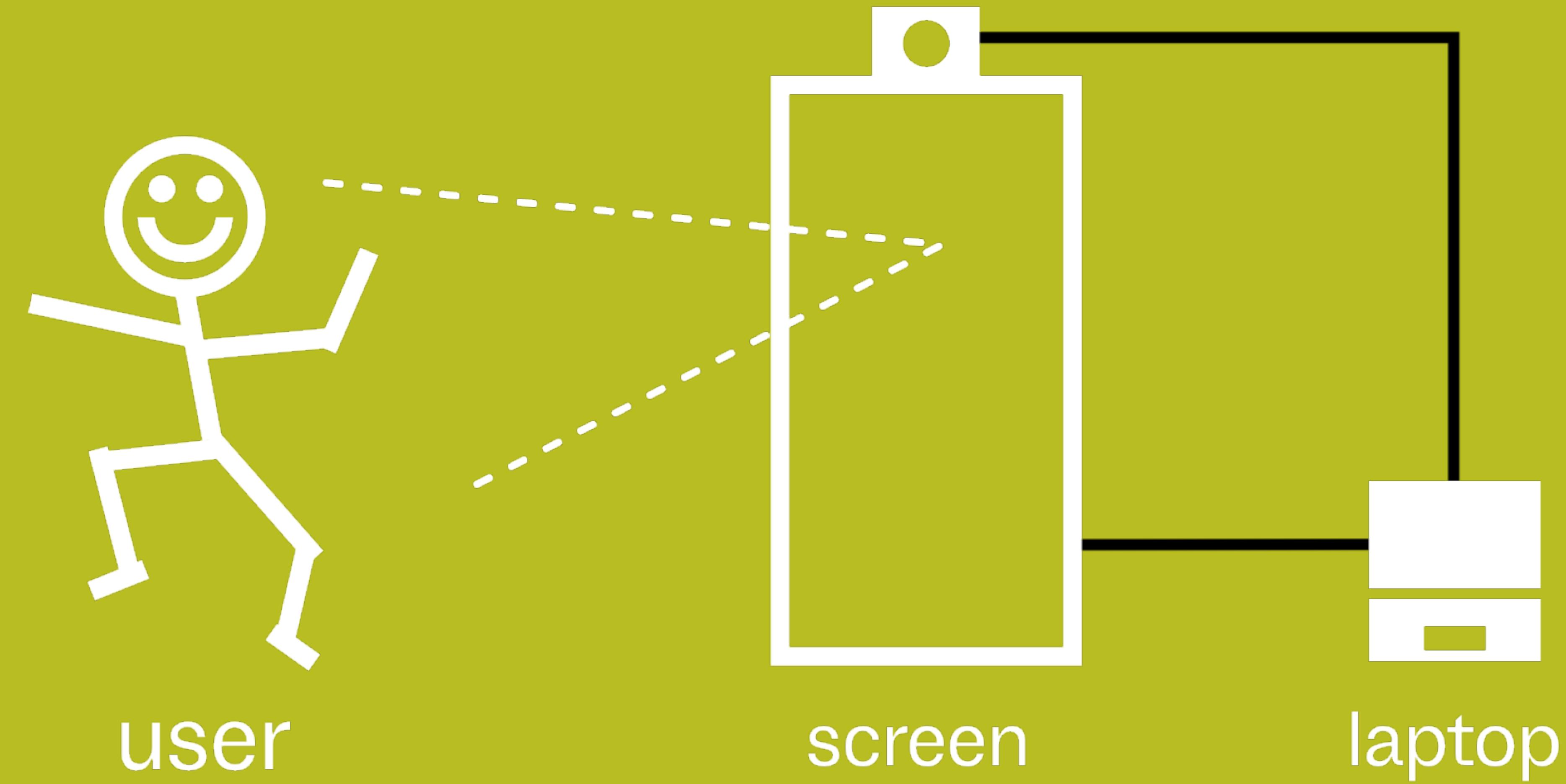
embodied
interaction



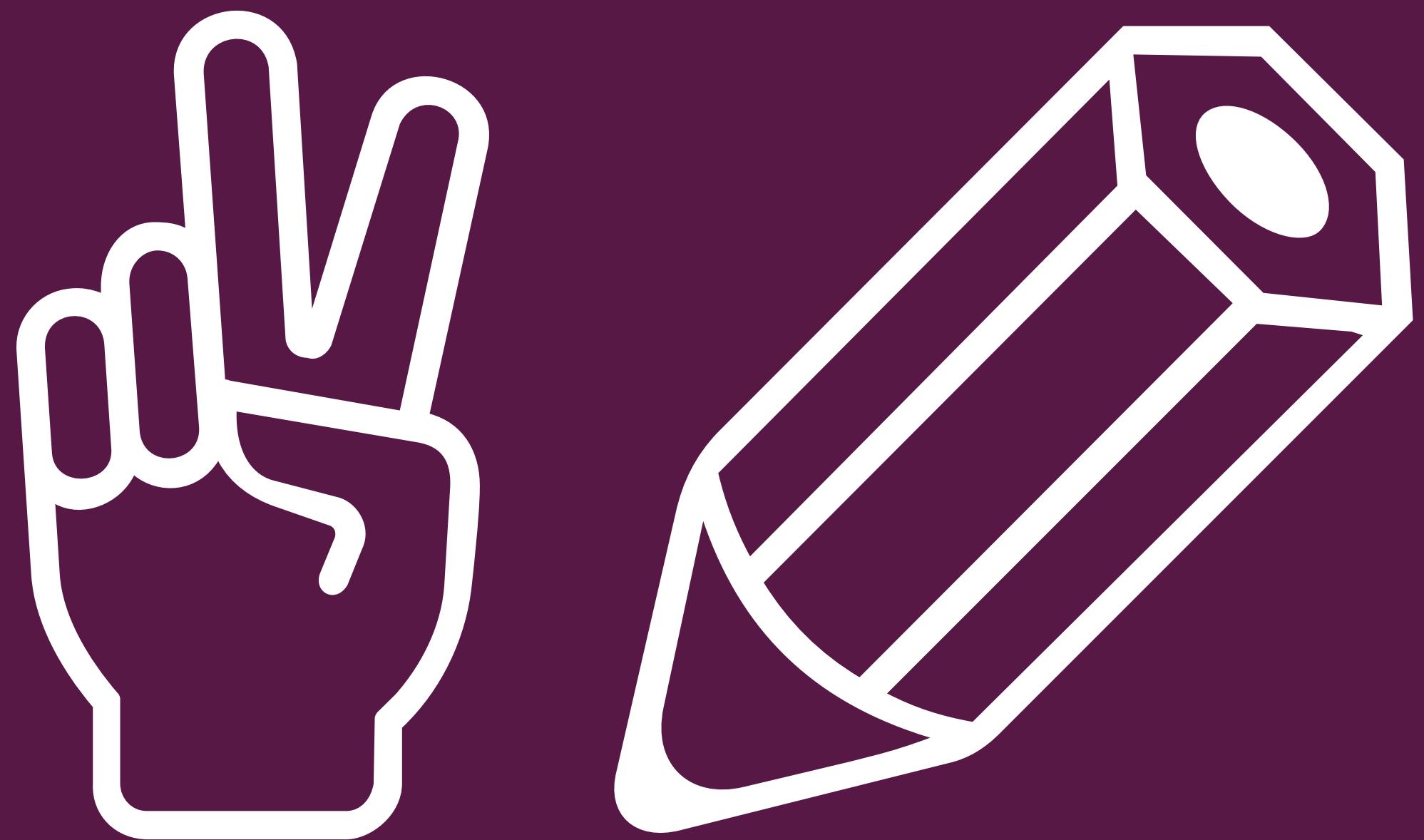
Body-Tracking
mit p5.js und
mediapipe



FHV
10.10.2025



Eure Aufgabe



- Entwickelt eine interaktive Body-Tracking-Installation für ein Museum, oder eine Ausstellung.
- Spielt mit dem Körper als Interface.
- Eignet euch Coding als Werkzeug an.
- 😊 Nutzt das Design-Forum!

Session 1

- 10.10.2025 (online)
 - Kurseinführung, p5.js Basics
 - 📝 Spielen
-

Session 2

- 17.10.-10.11.2025 (on demand)
 - mediapipe Basics
 - 📝 Spielen & Idee entwickeln
-

Workshop

- 14.11. & 15.11.2025
 - 2 Tage Workshop vor Ort
 - 📝 Finalisieren!
-

Feedback

- Ende Nov. / Anfang Dez.
-

Präsentation

→ 23.01.2025

- Hallo :-)
- Was ist Embodied Interaction? (30min)

☕ Kaffepause (10 min)

- Coding I: p5.js Basics (55min)



1 – Hallo

a designer lost
in the world
of code ...



Freiburg,
Hannover, Prag

Design & Code
Wissenschaft
Visual Storytelling

Freelancer, Papa, Lehre

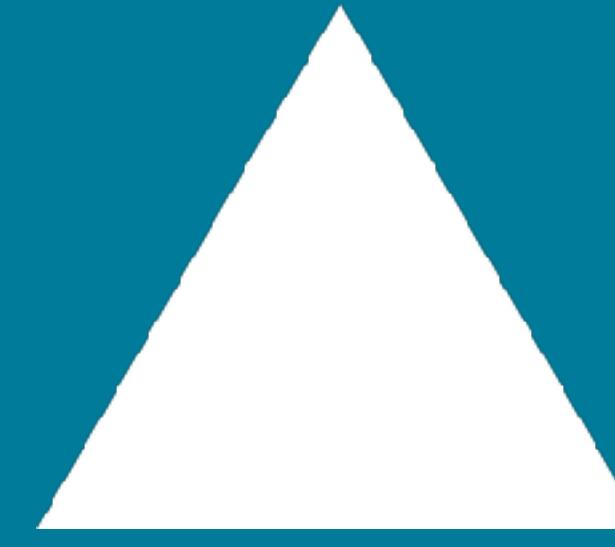




science

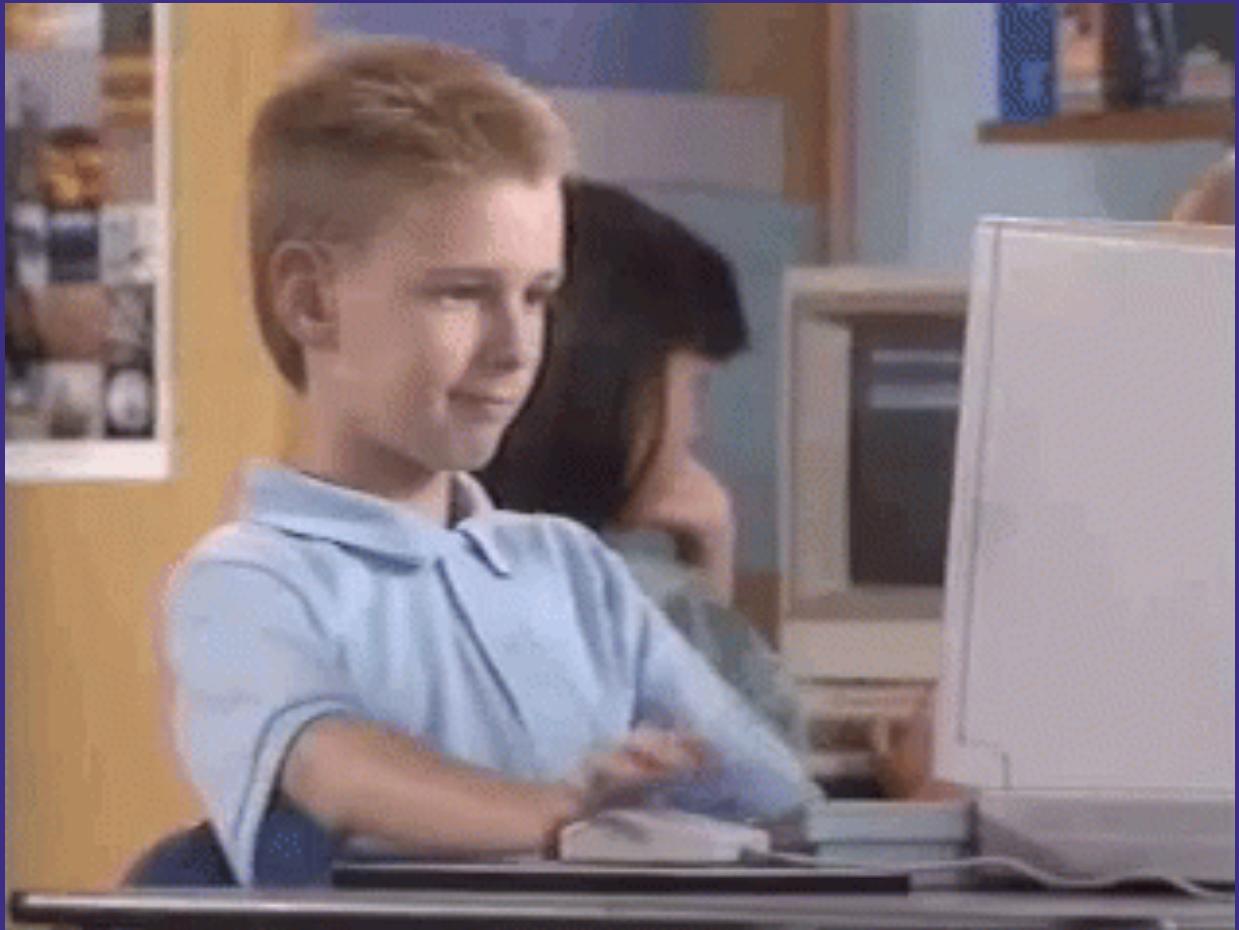
art

tech



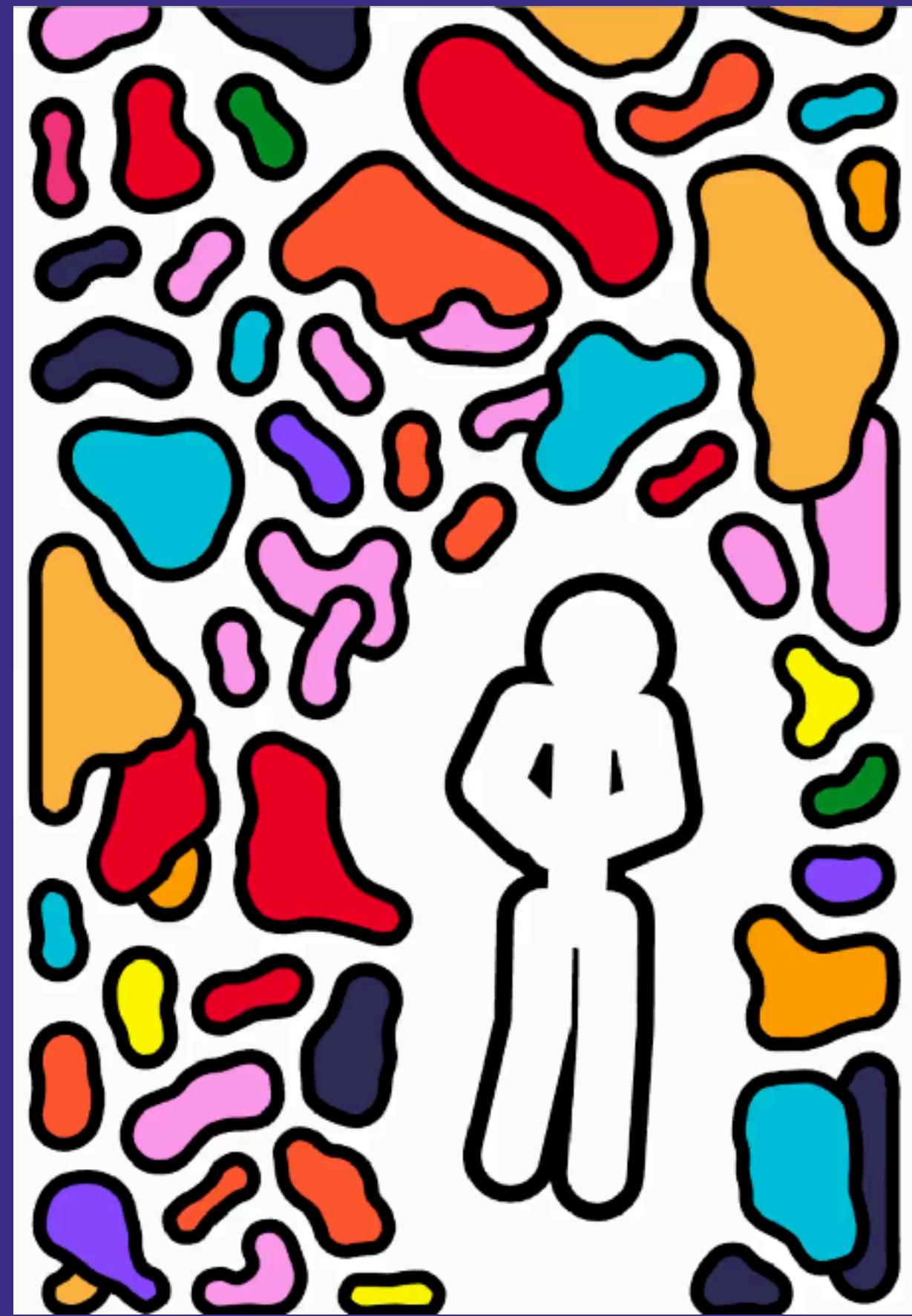
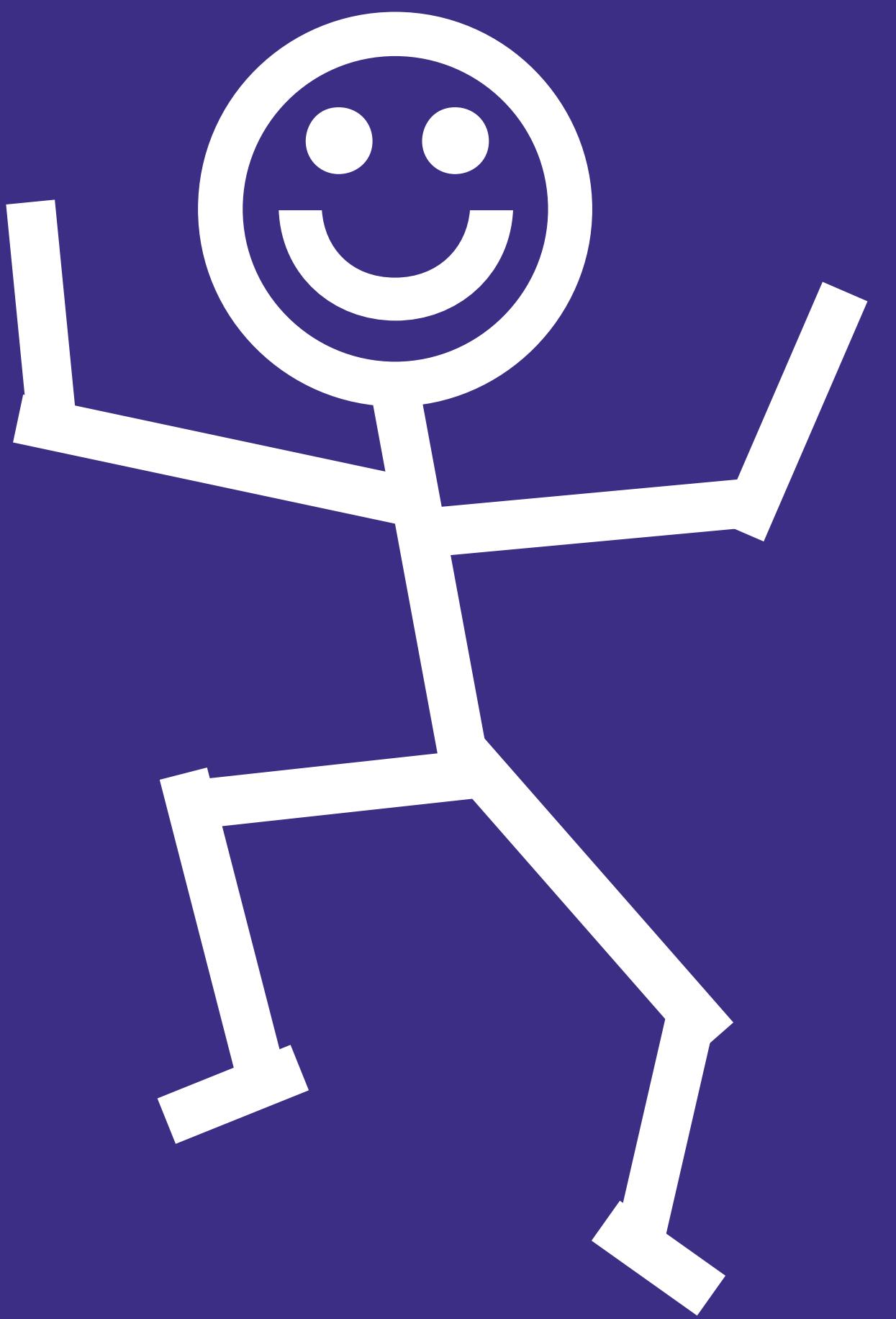
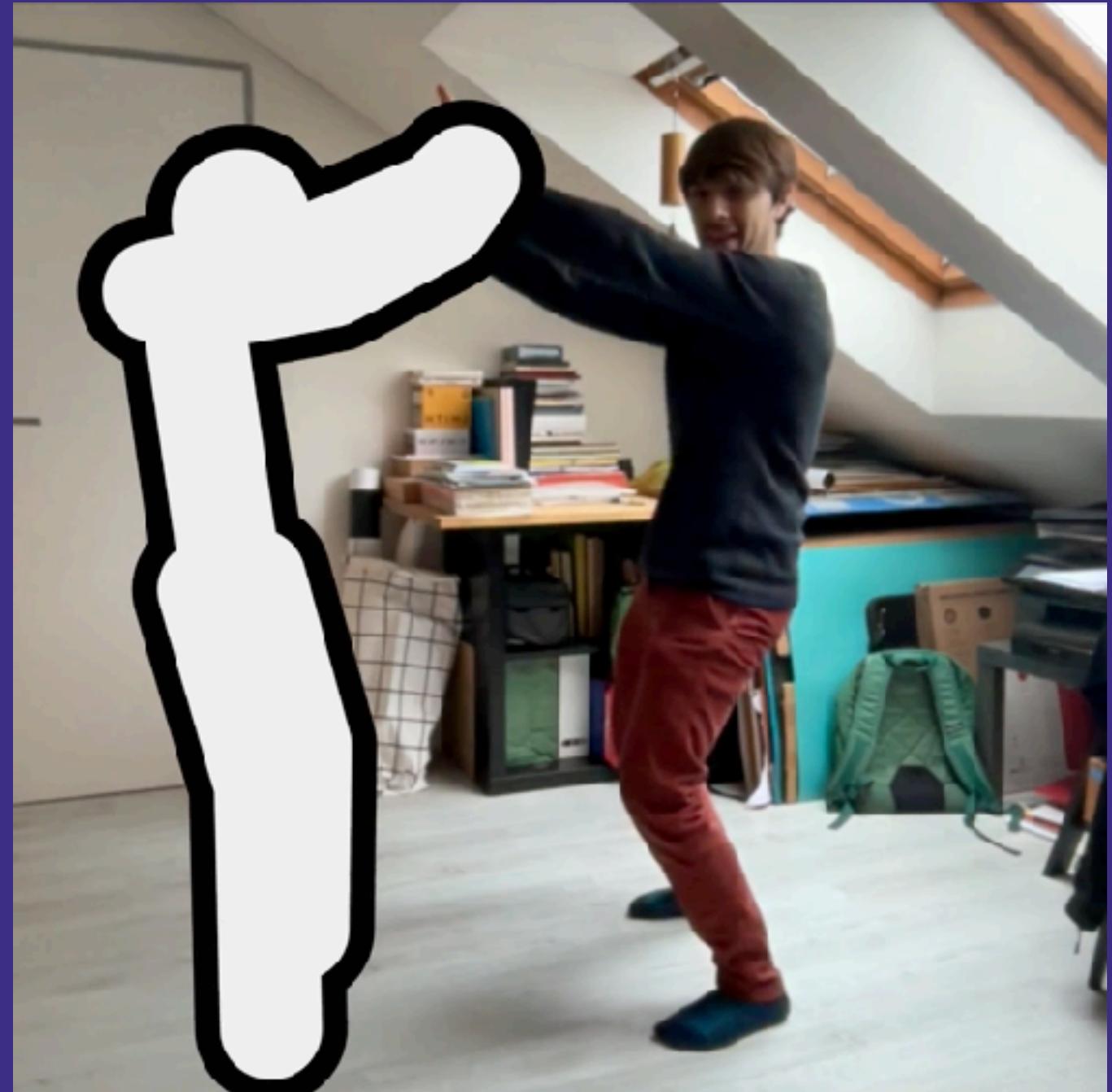
2 – Embodied Interaction

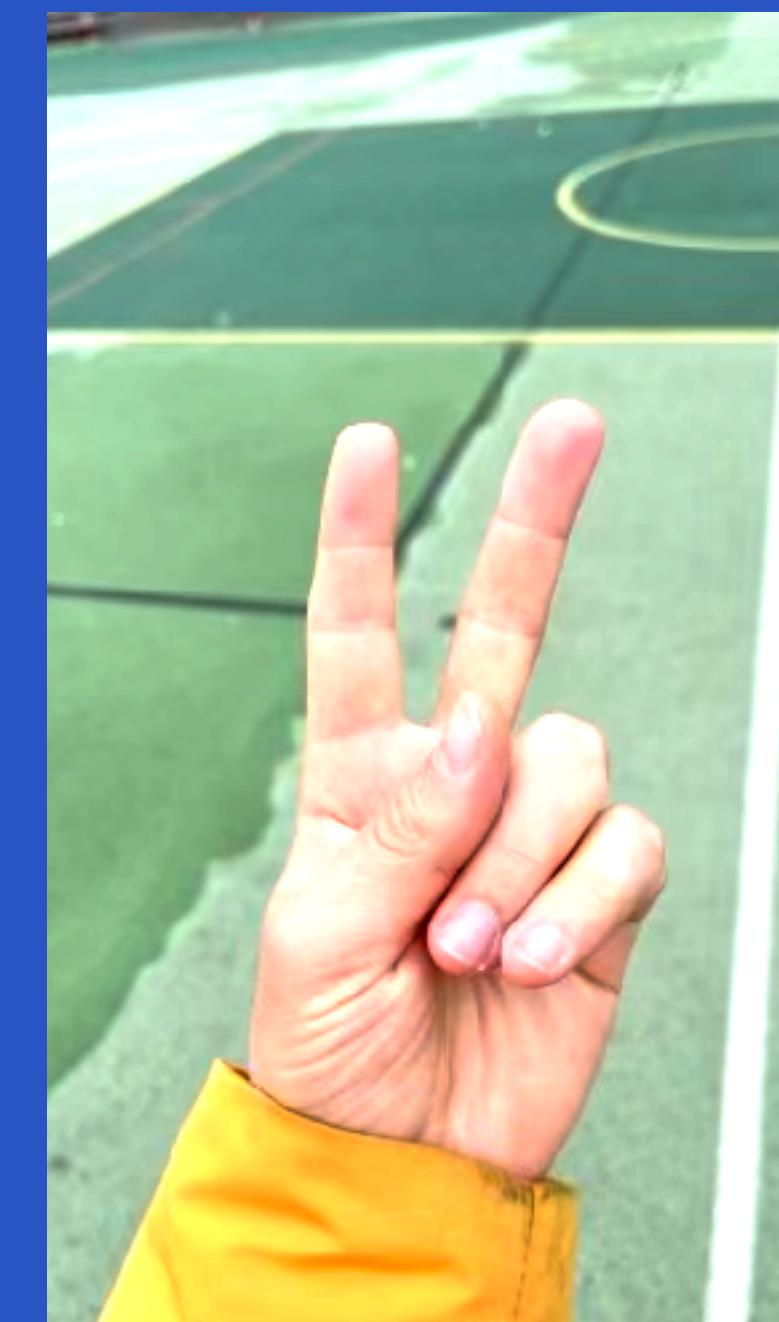
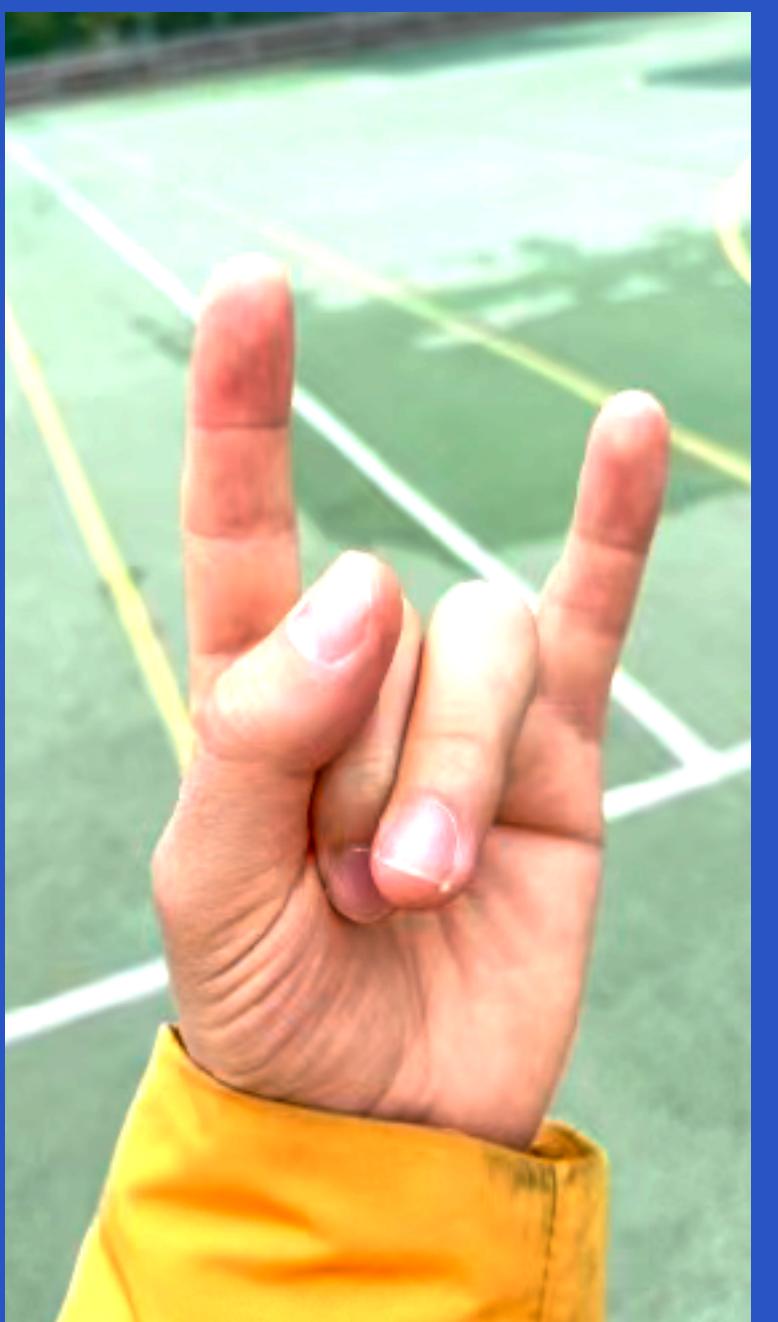
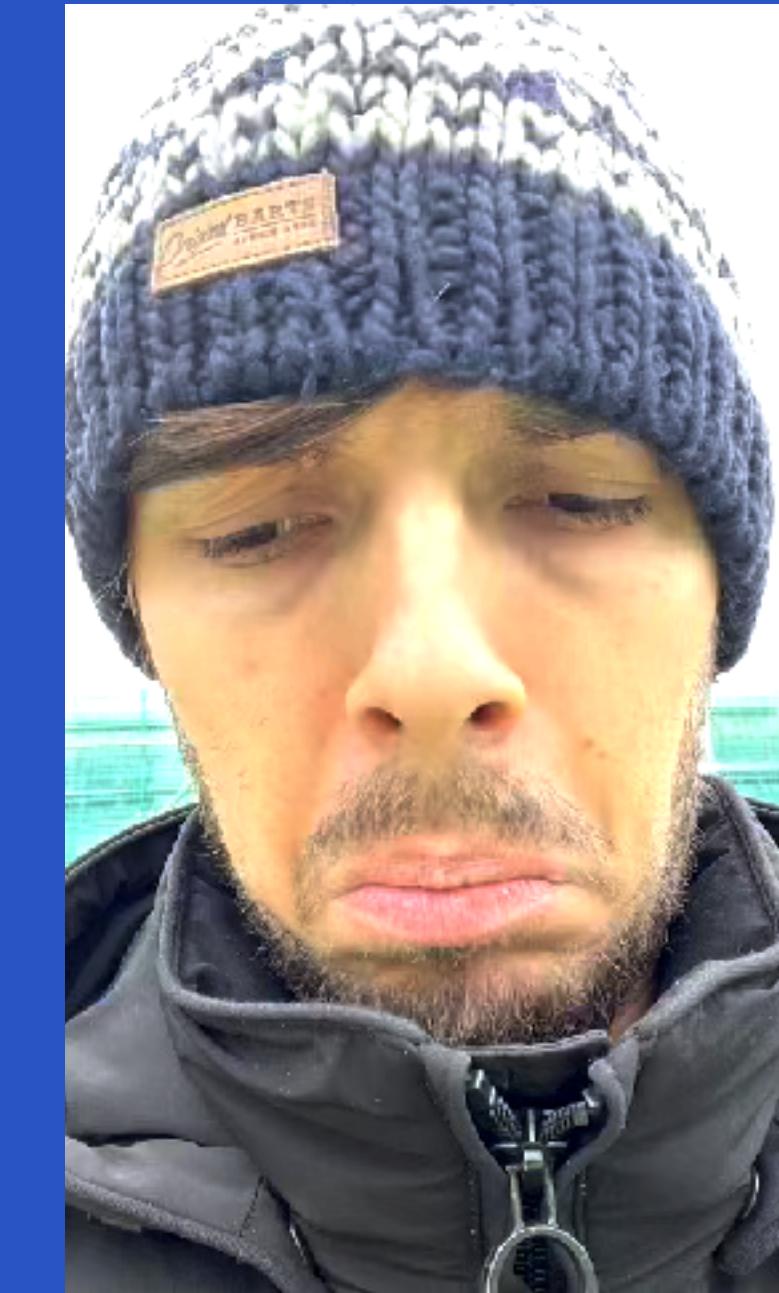
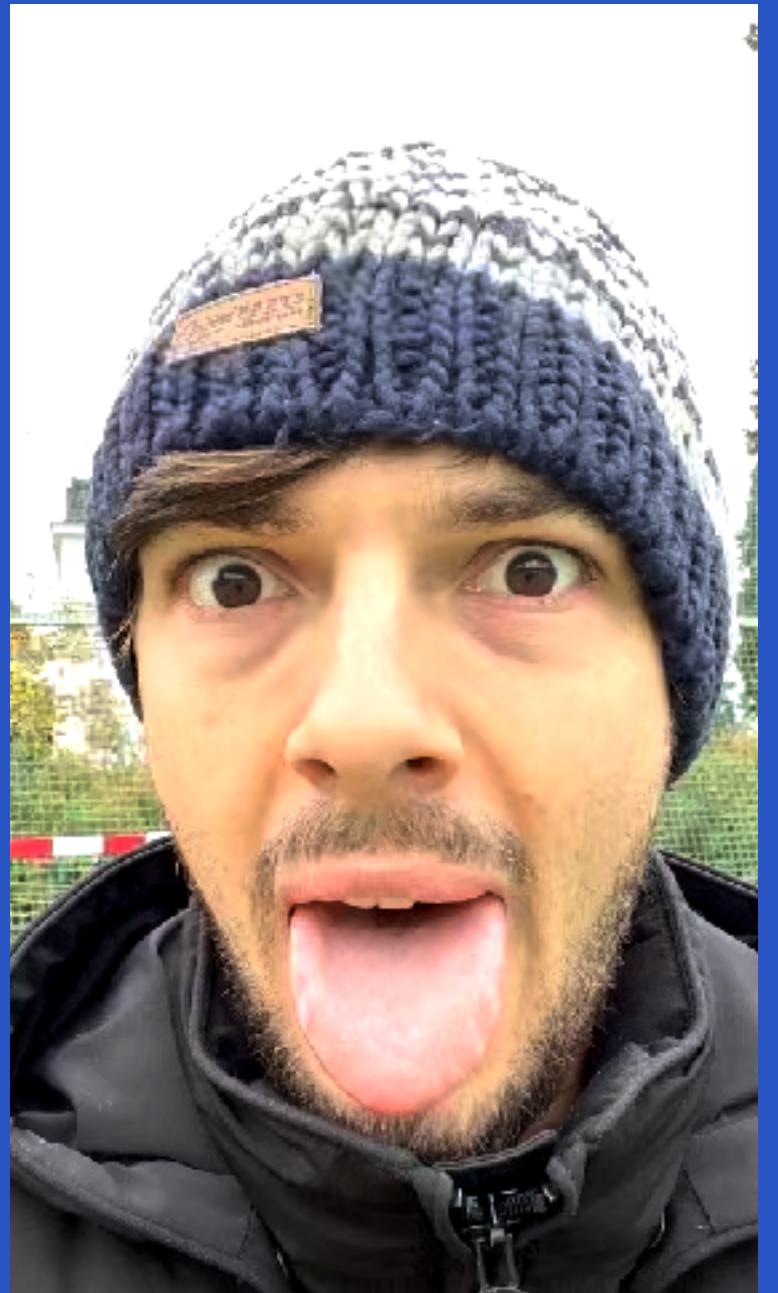
Wie interagieren wir mit Computern?



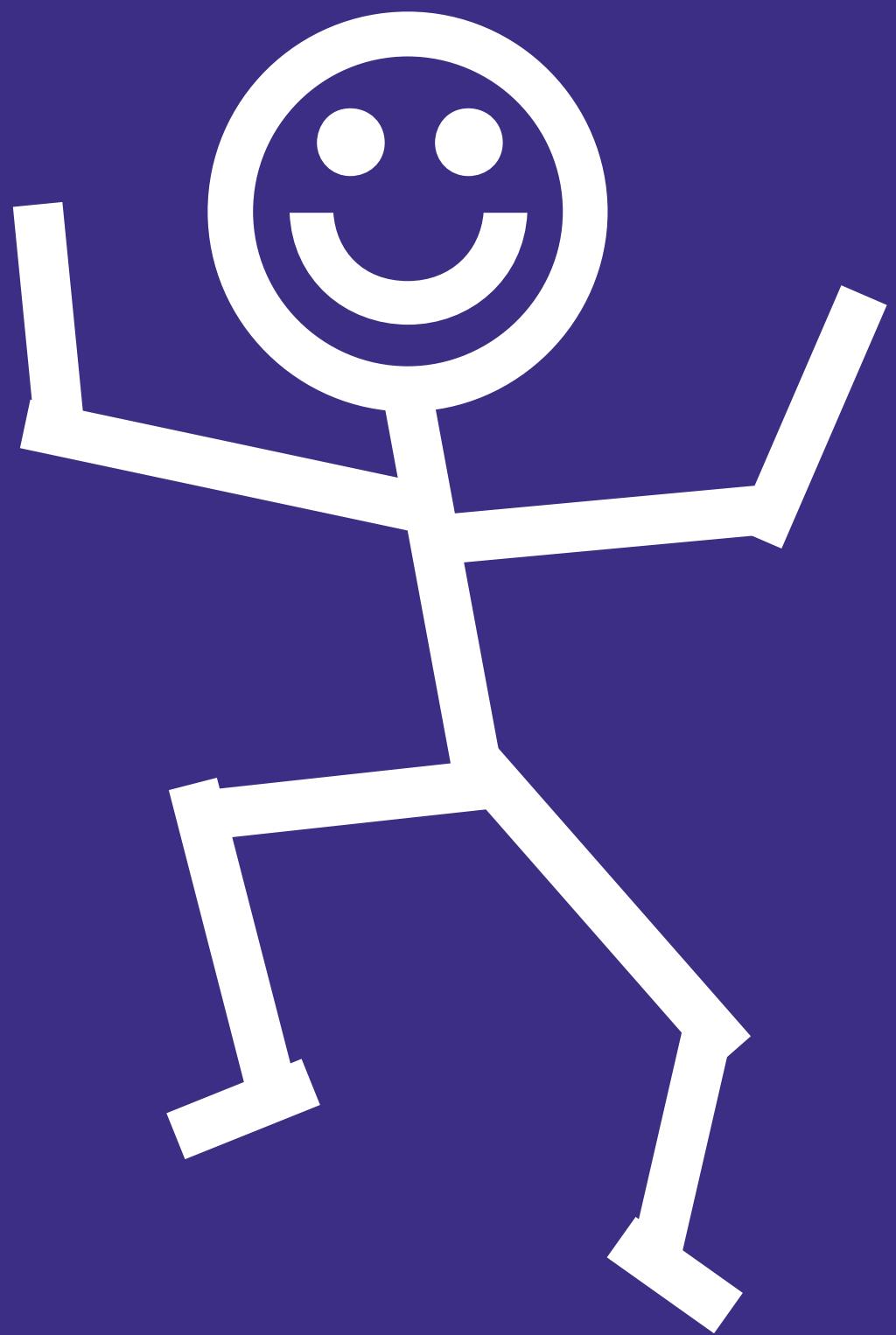
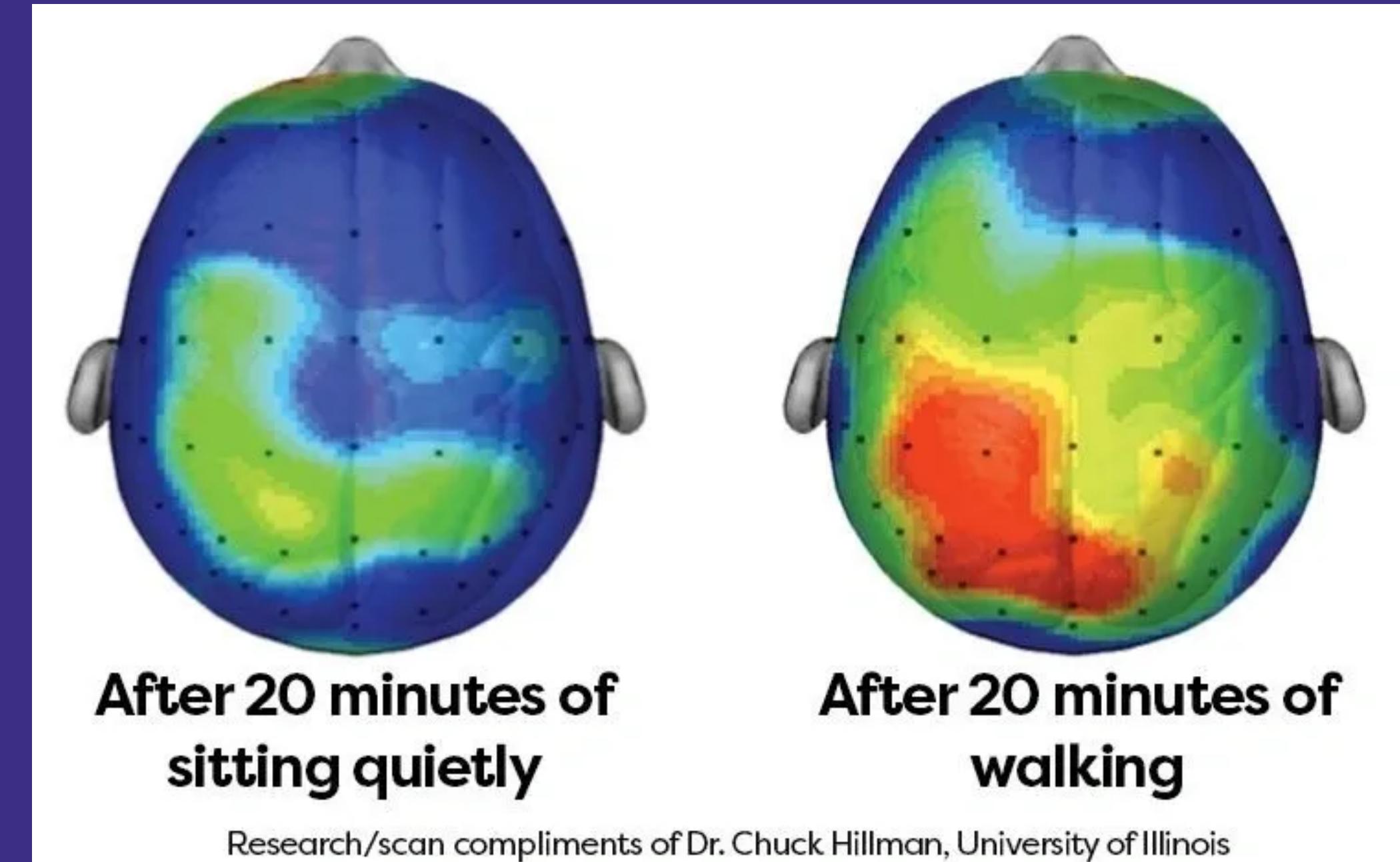


Geht das nicht
anders?





Gehirnaktivität

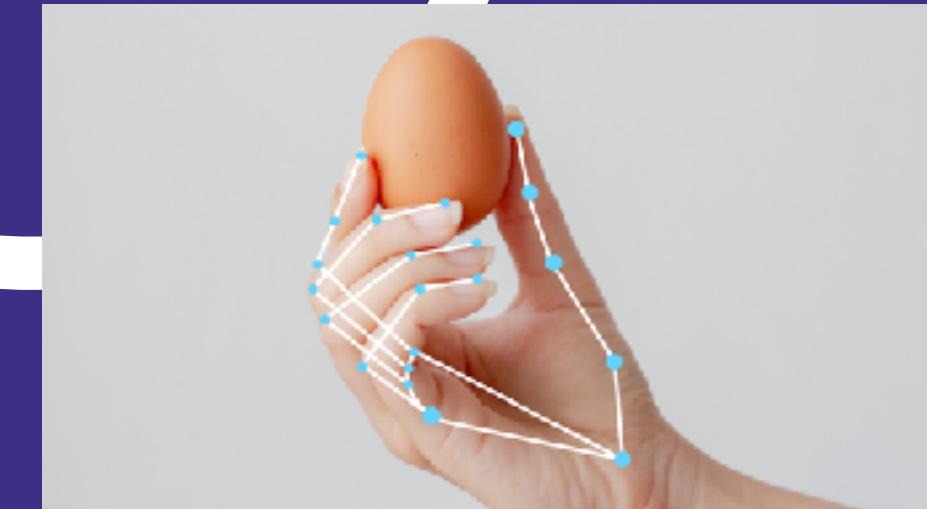
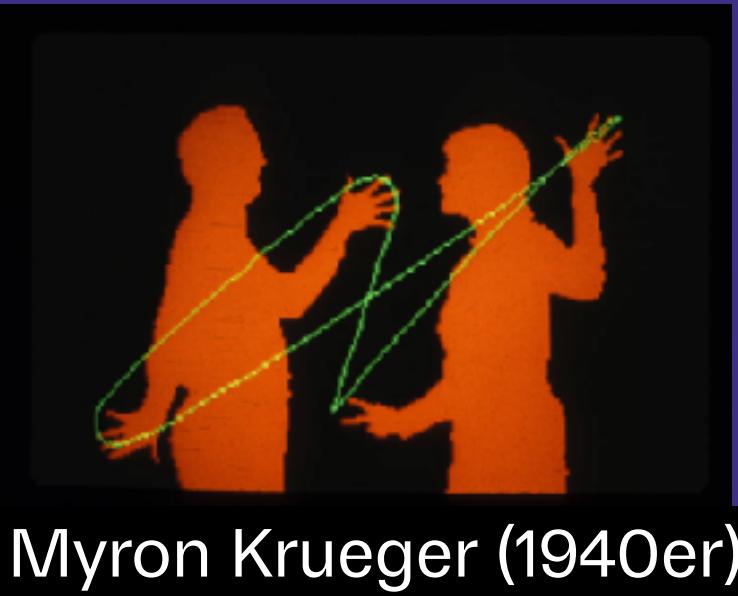


Warum jetzt?



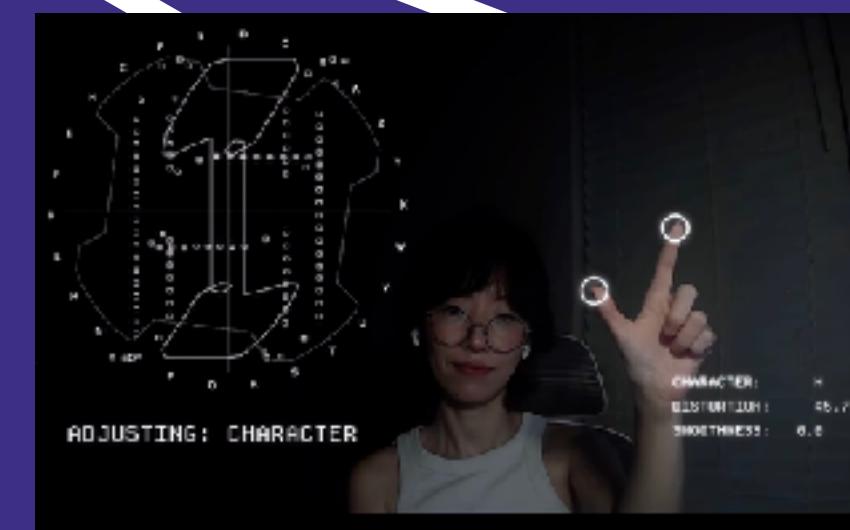
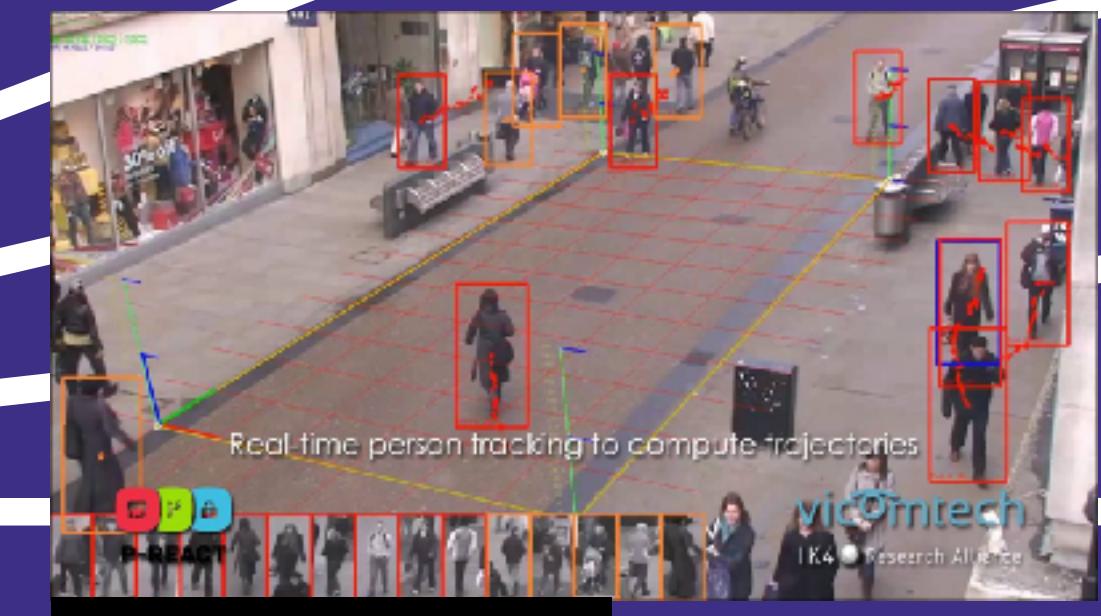
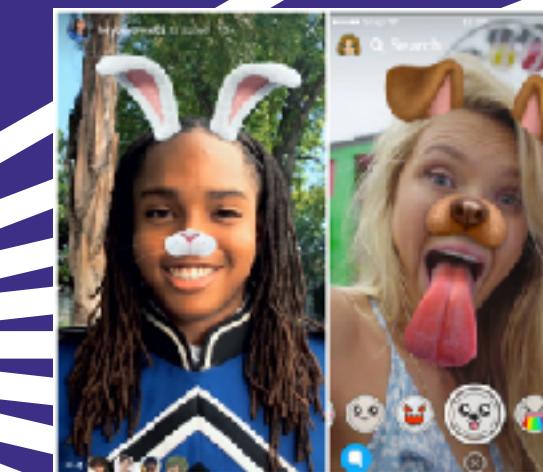
pioneers

tech

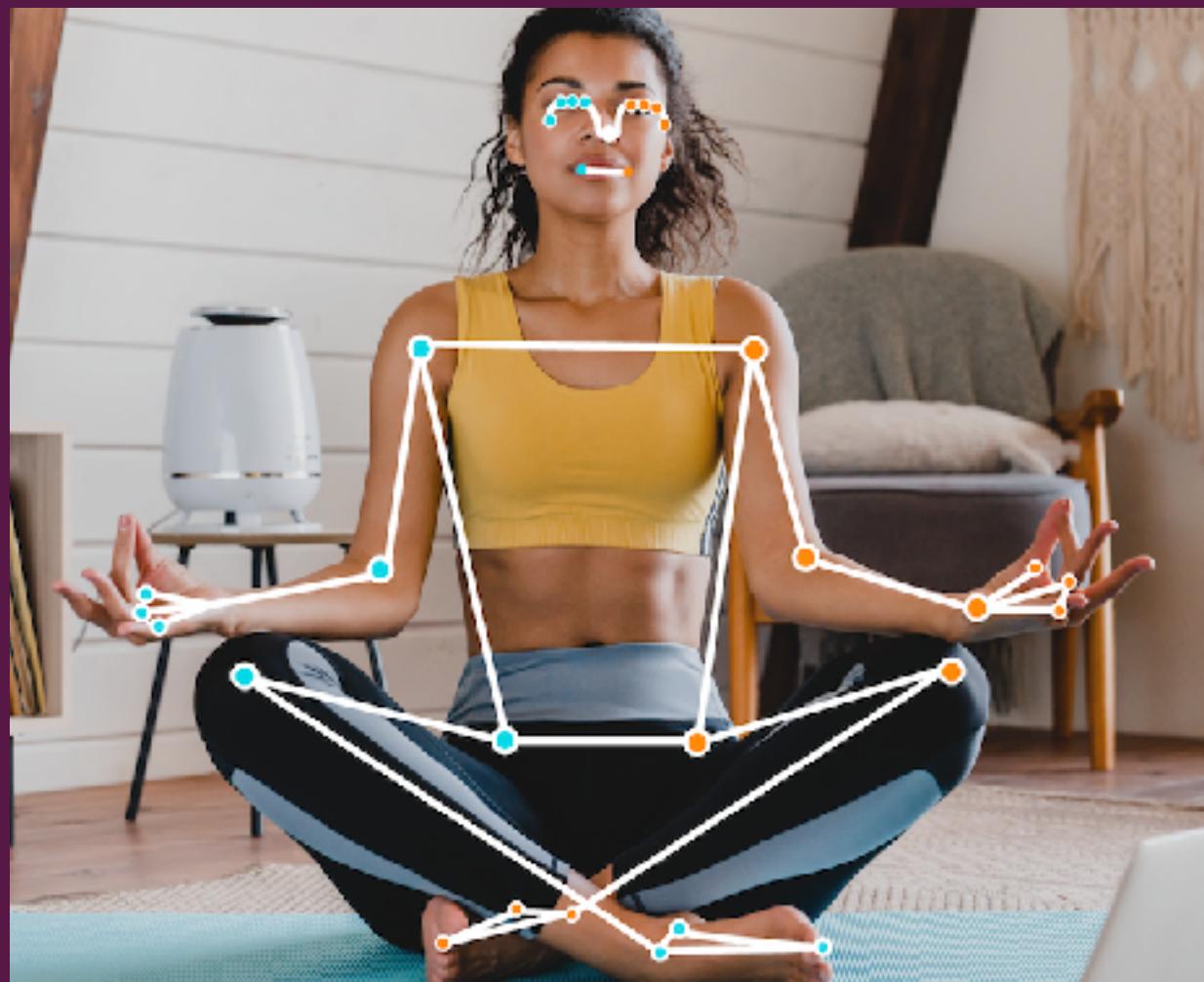


ml5.js (2018)

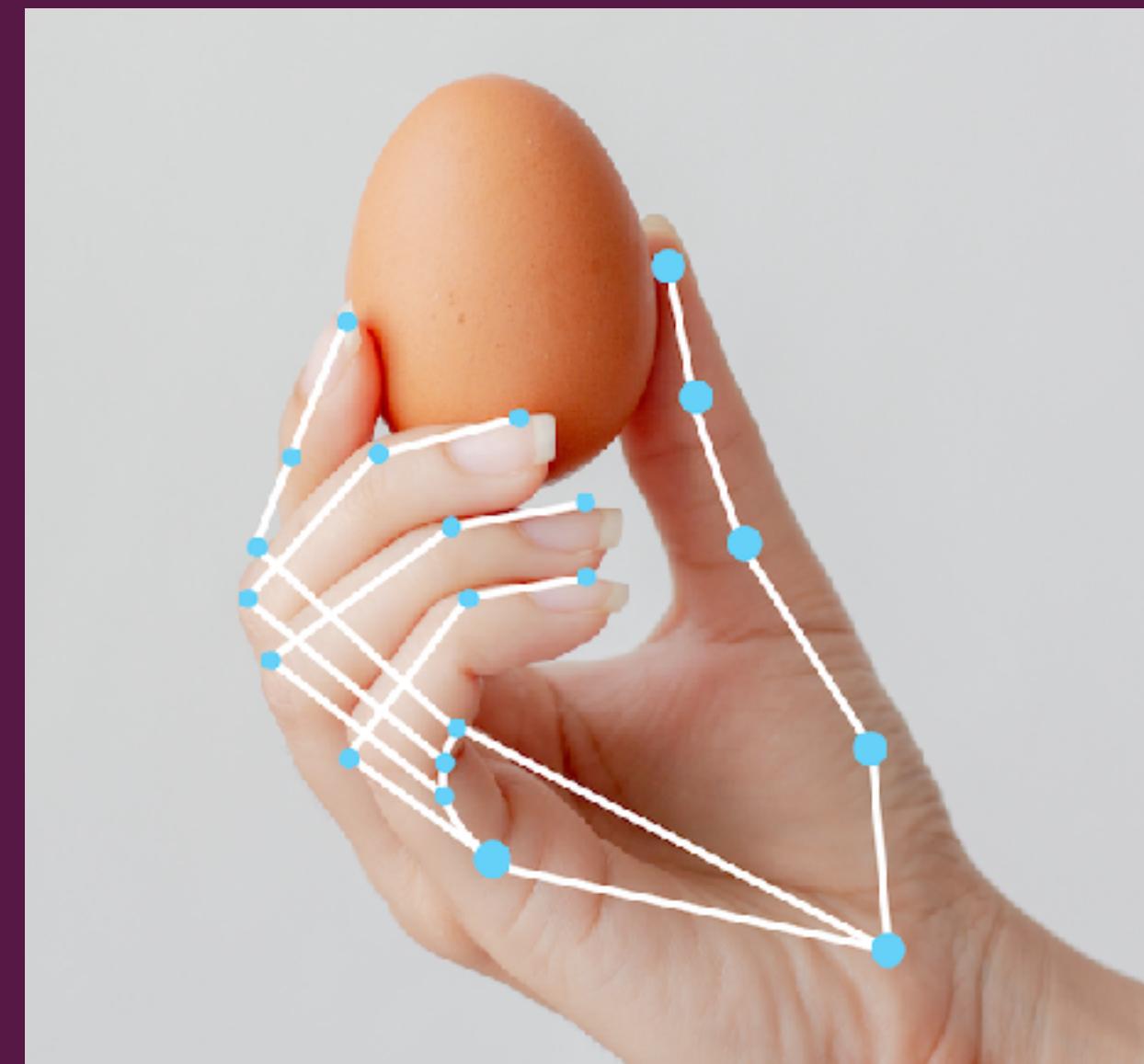
MediaPipe (2022)



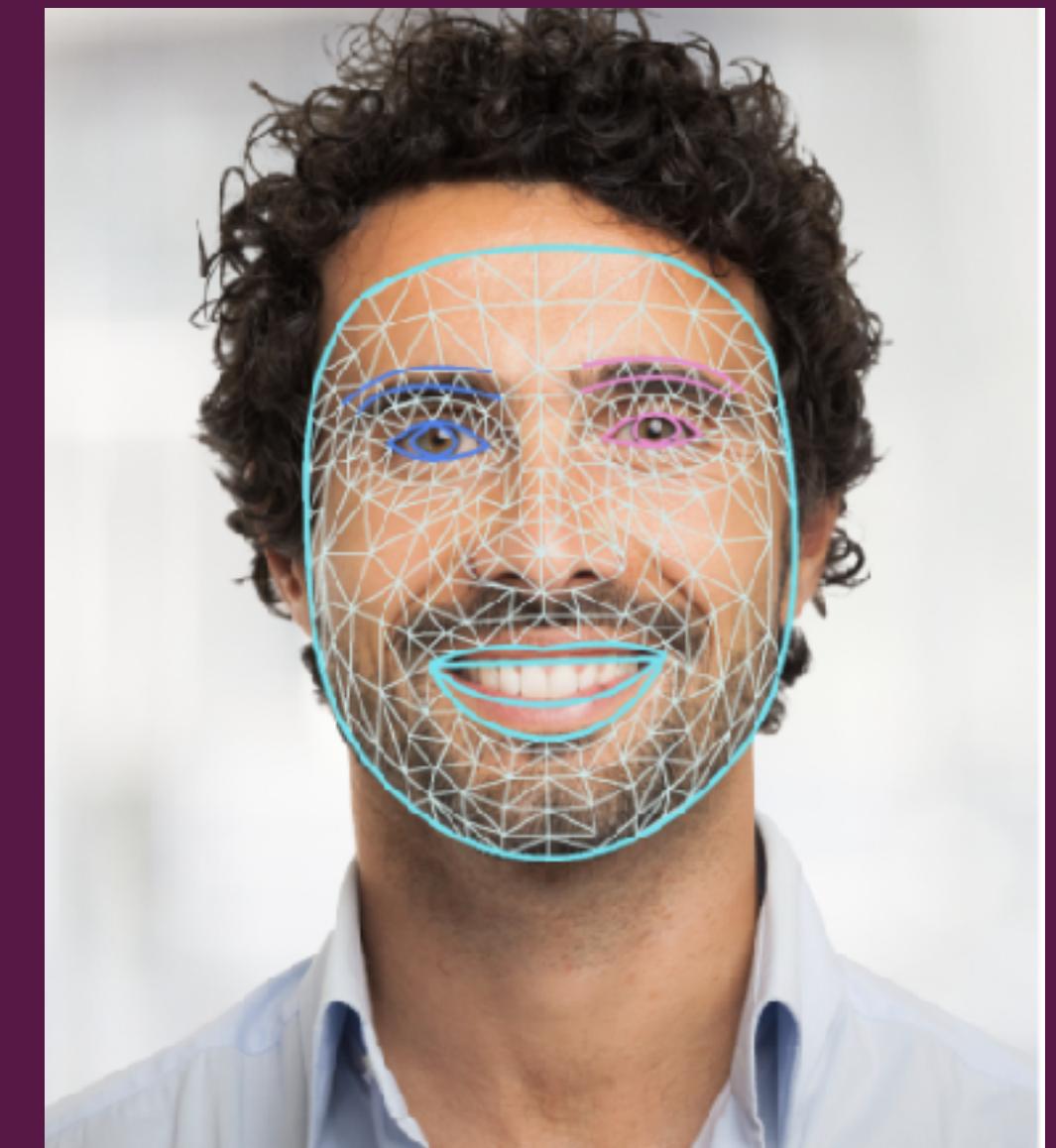
Wie macht man das?



pose



hand(s)

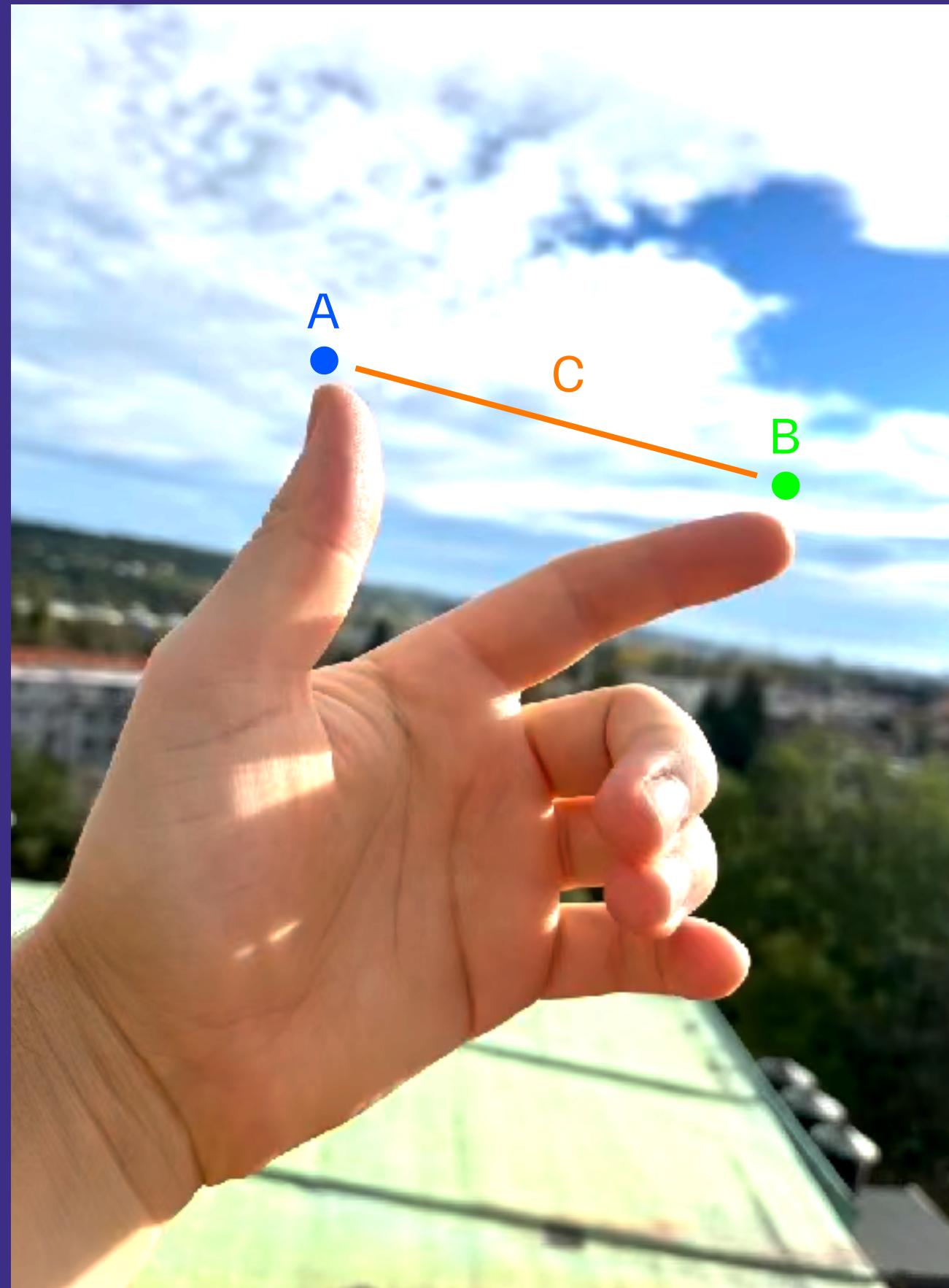


face

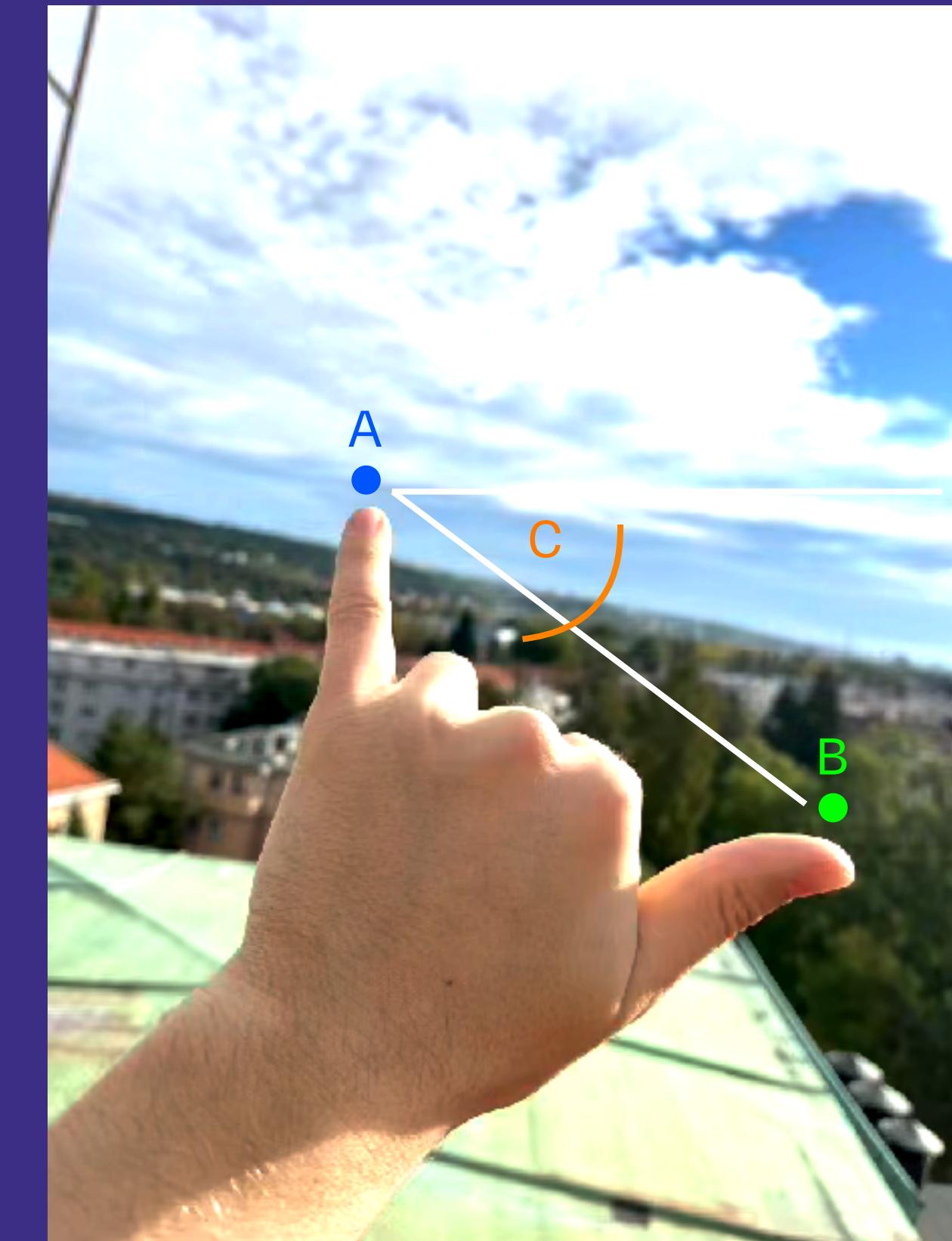
tracking
models

<https://ai.google.dev/edge/mediapipe>
<https://codepen.io/mediapipe>

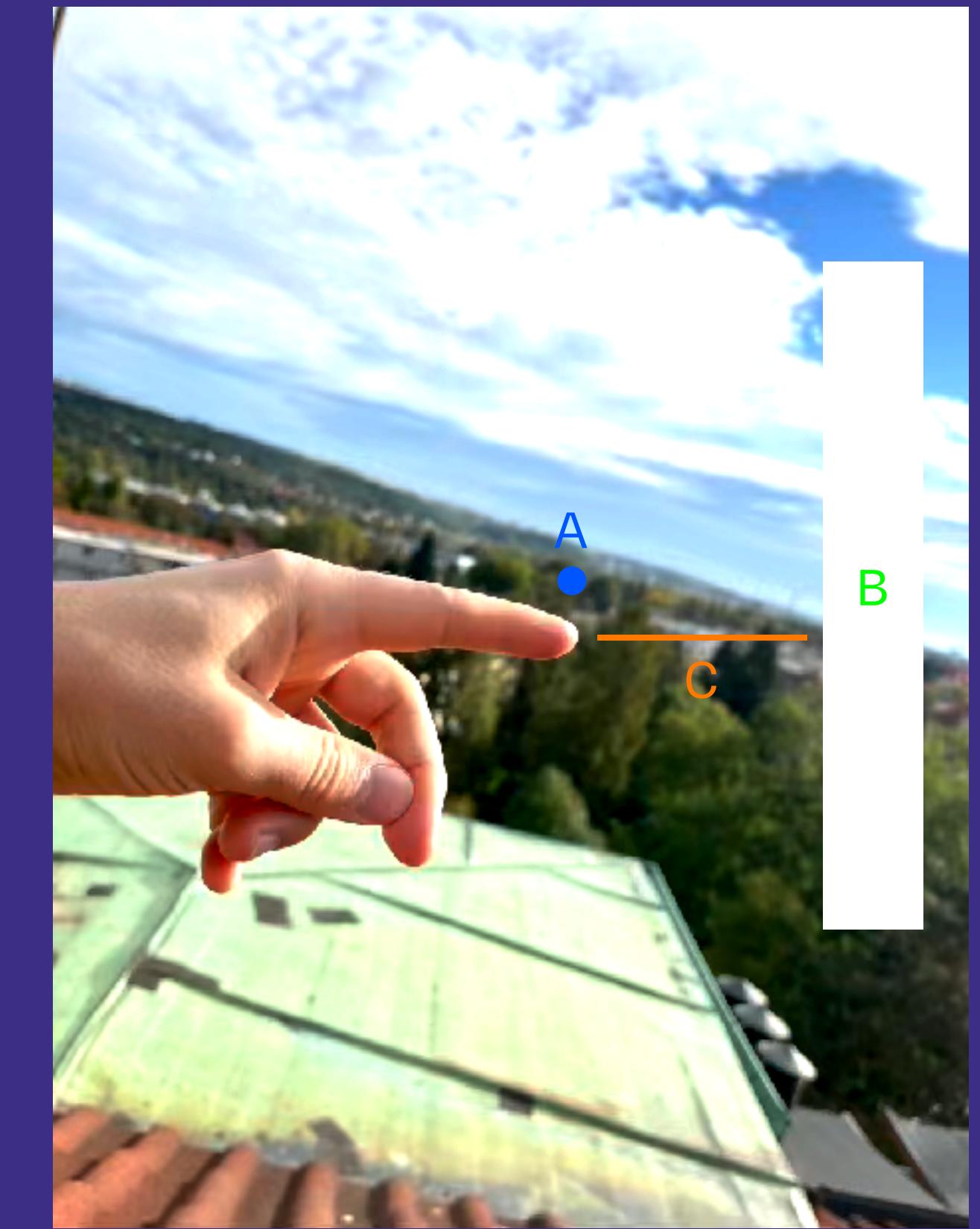
Interfaces neu denken



distance

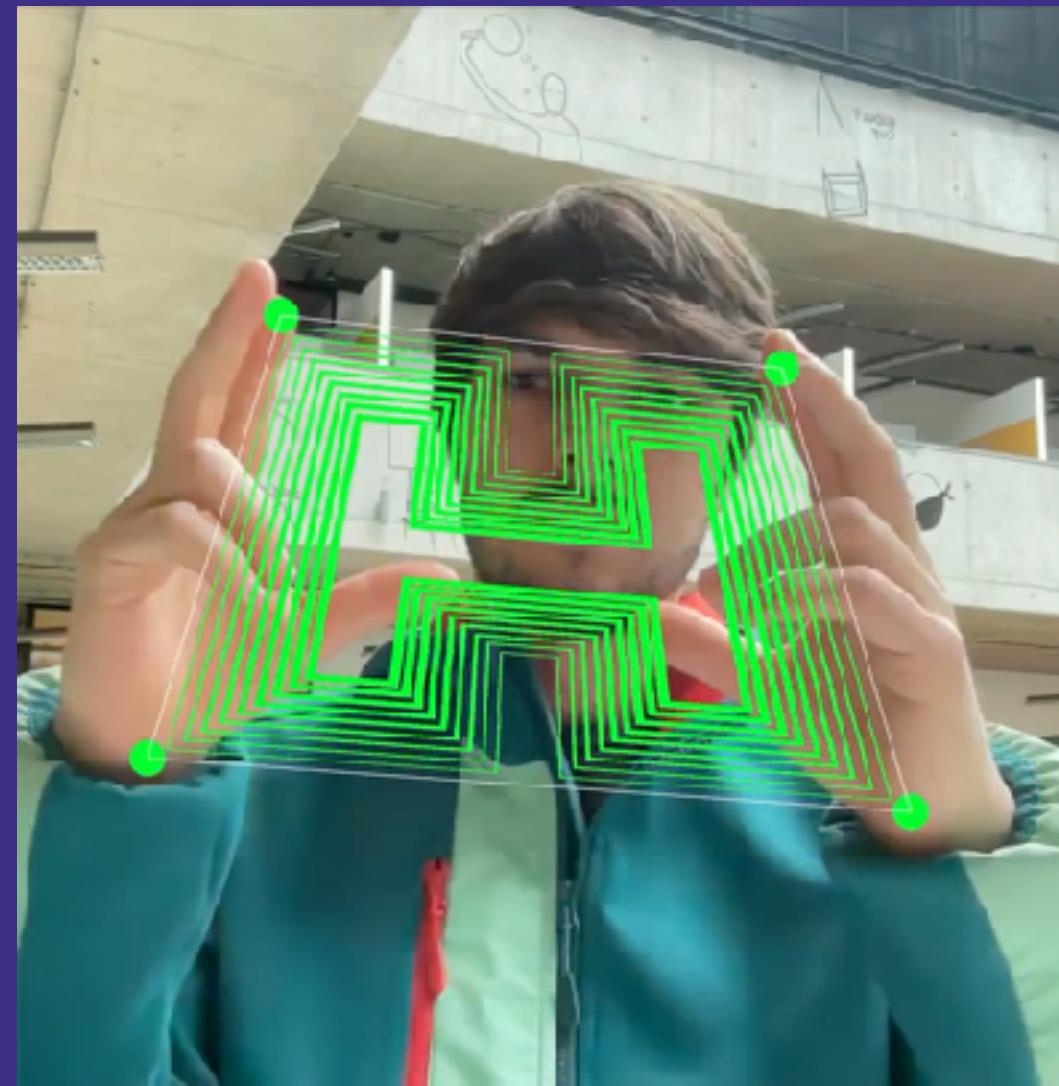


angle



distance to object

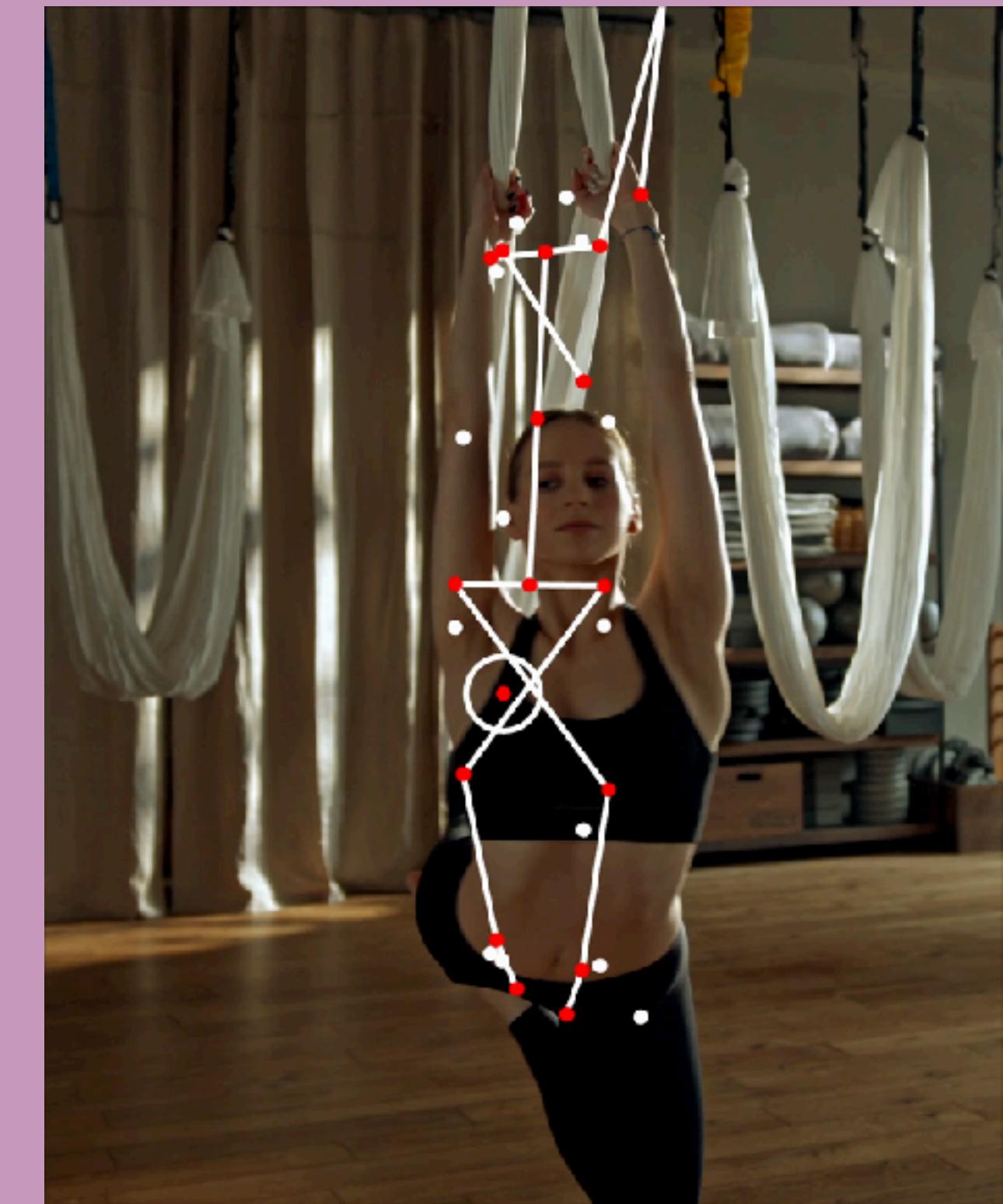
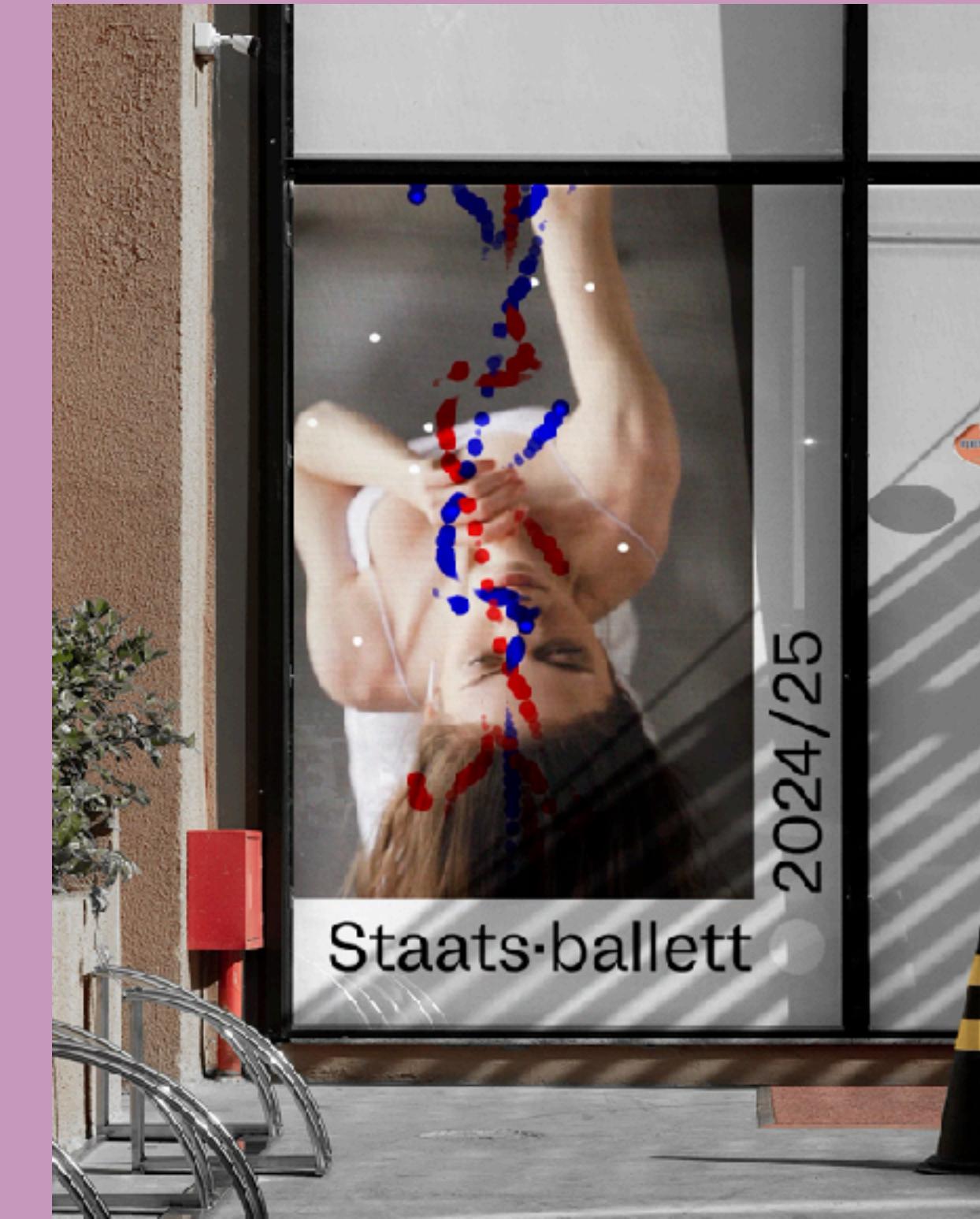
Potential

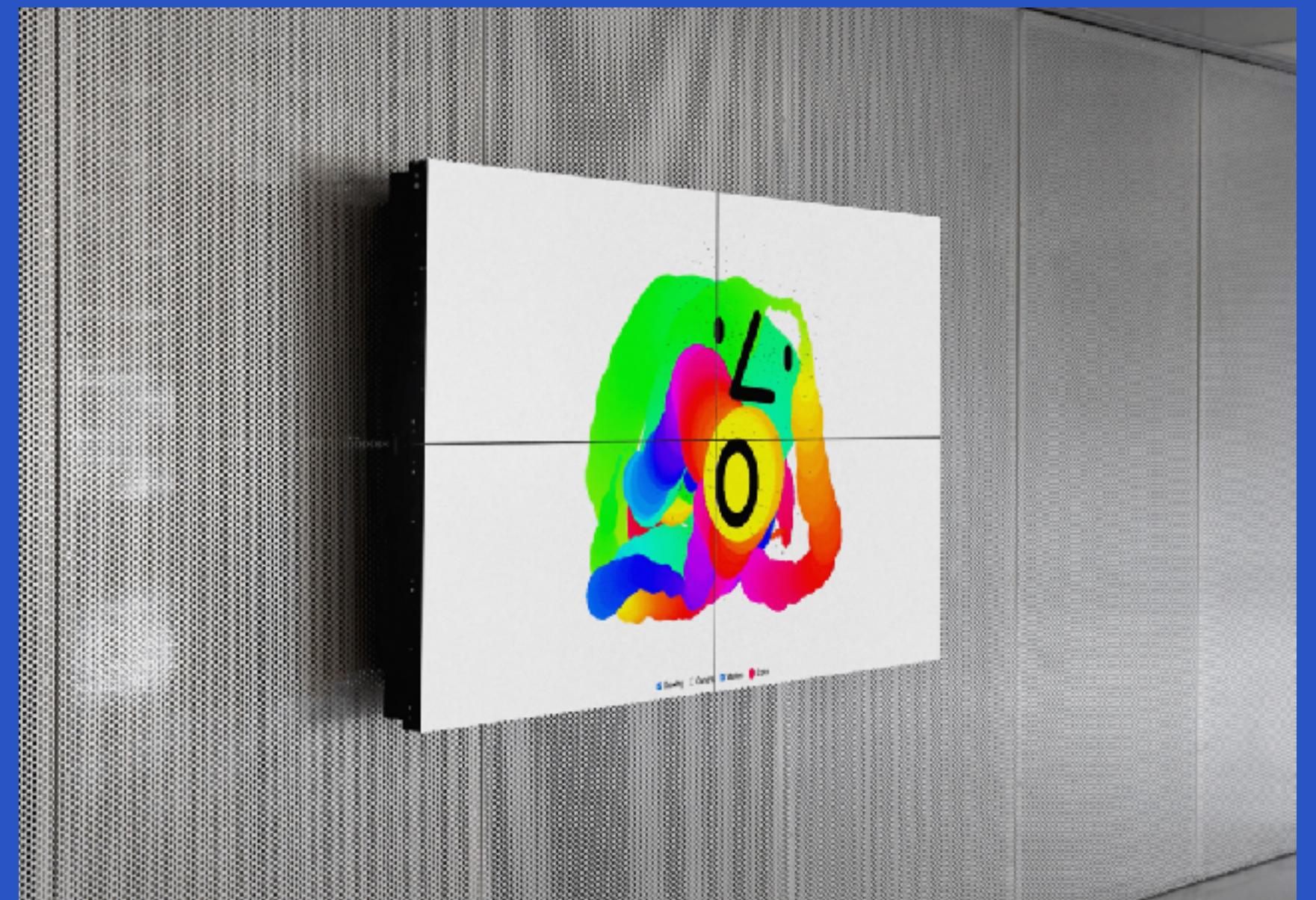


36 Days of Type, 2023



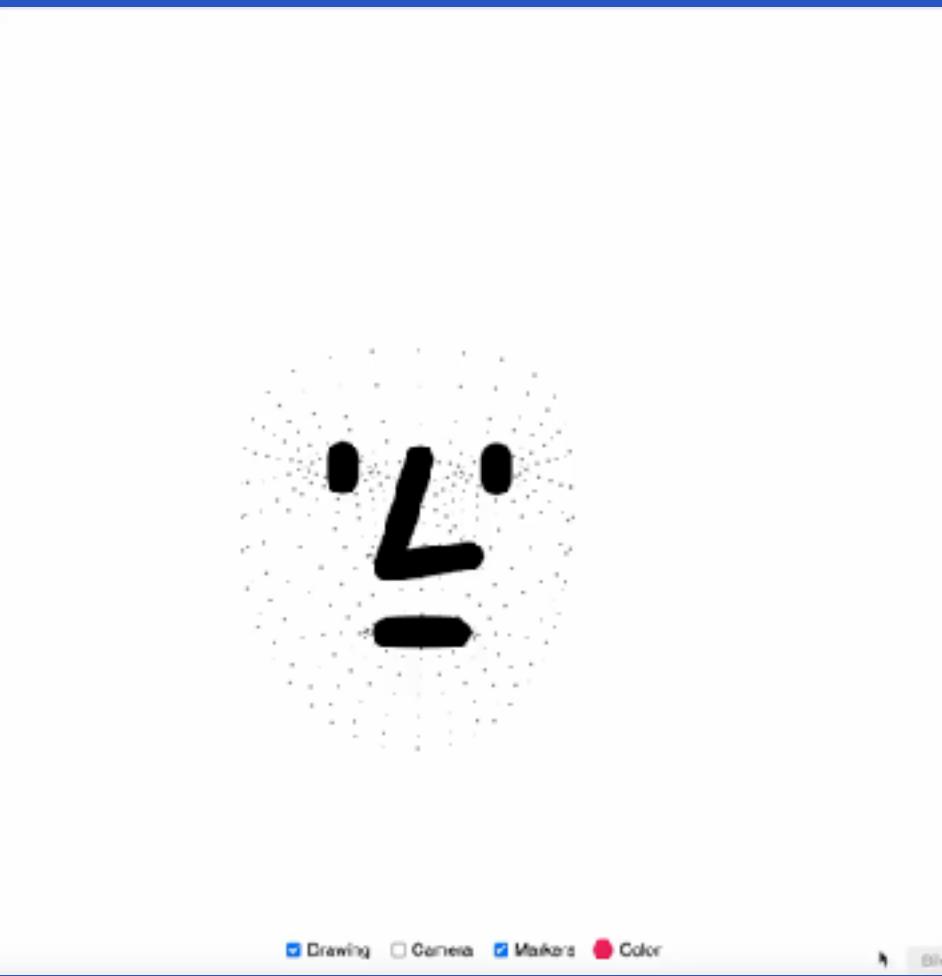
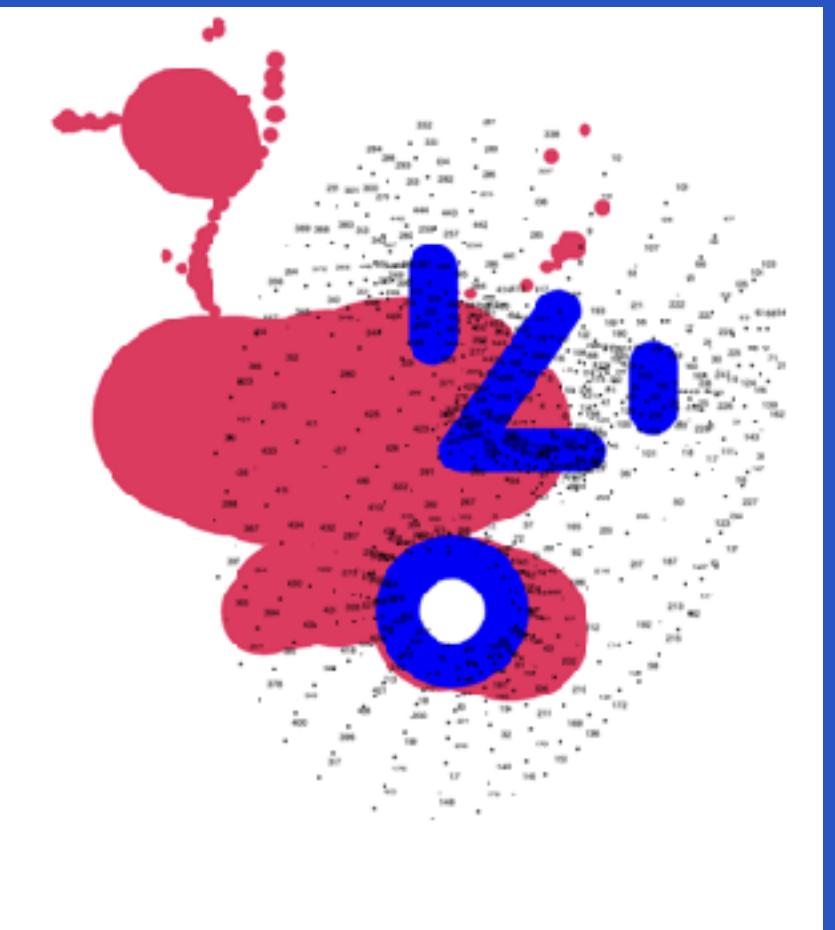
Staats·ballett



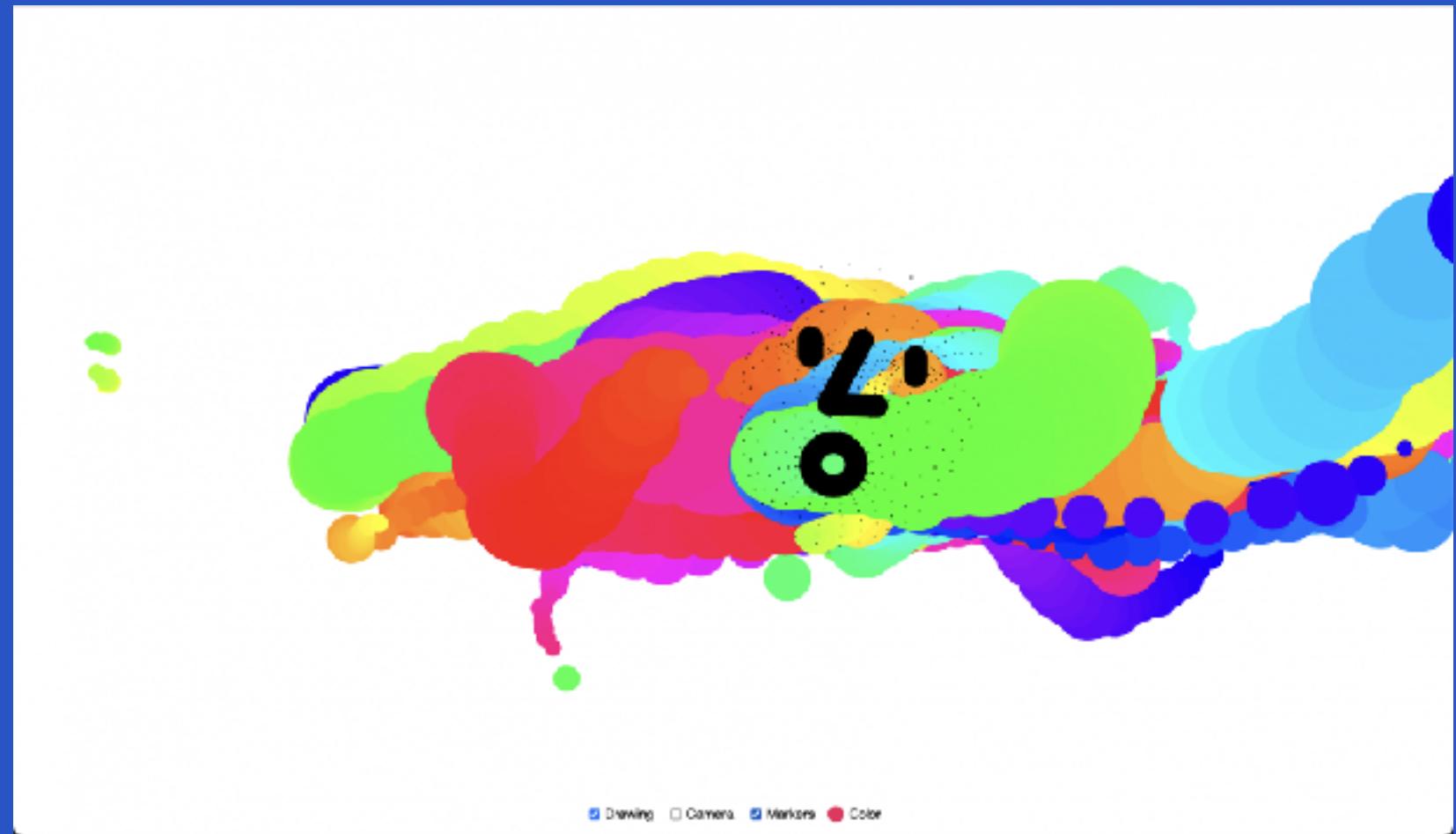
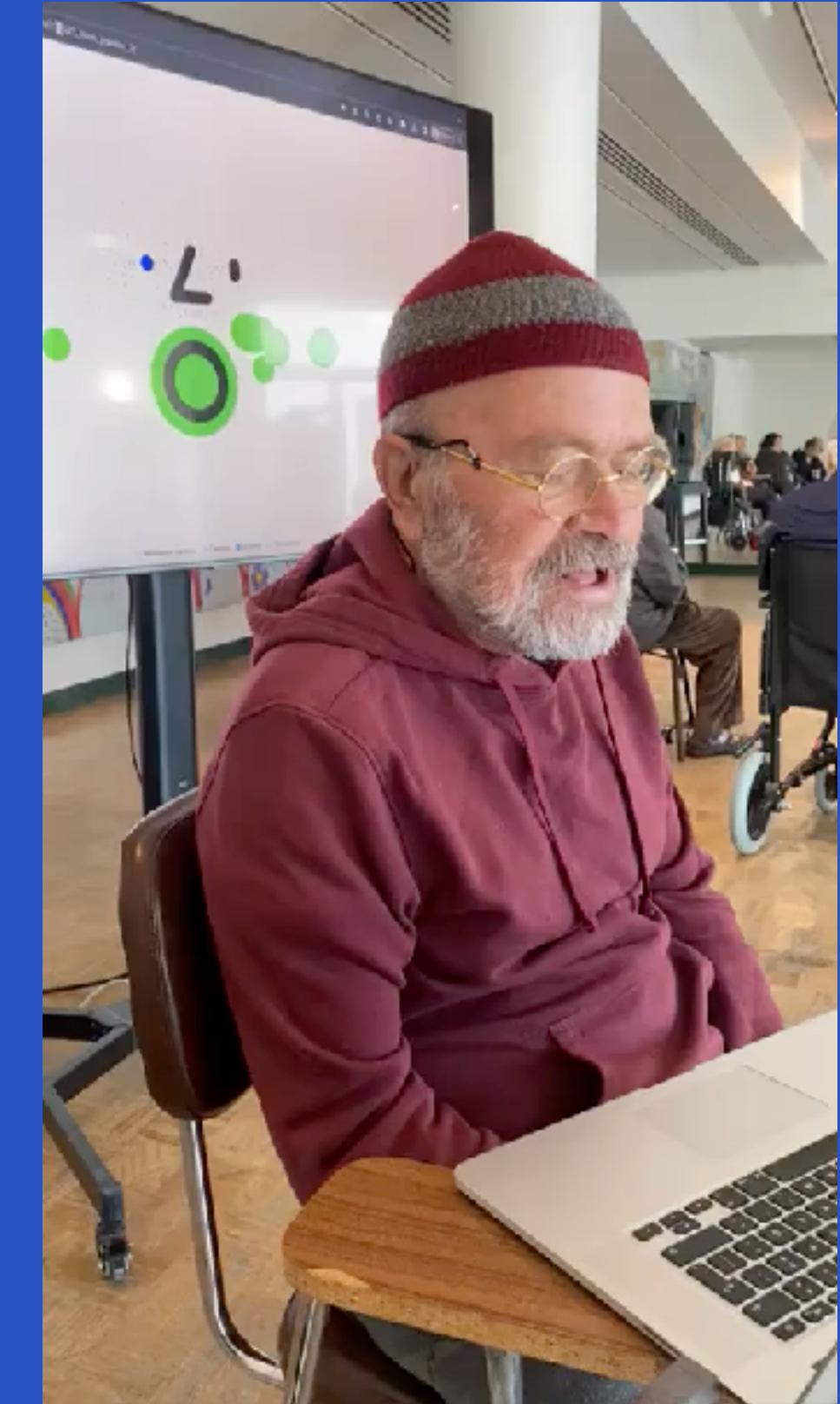


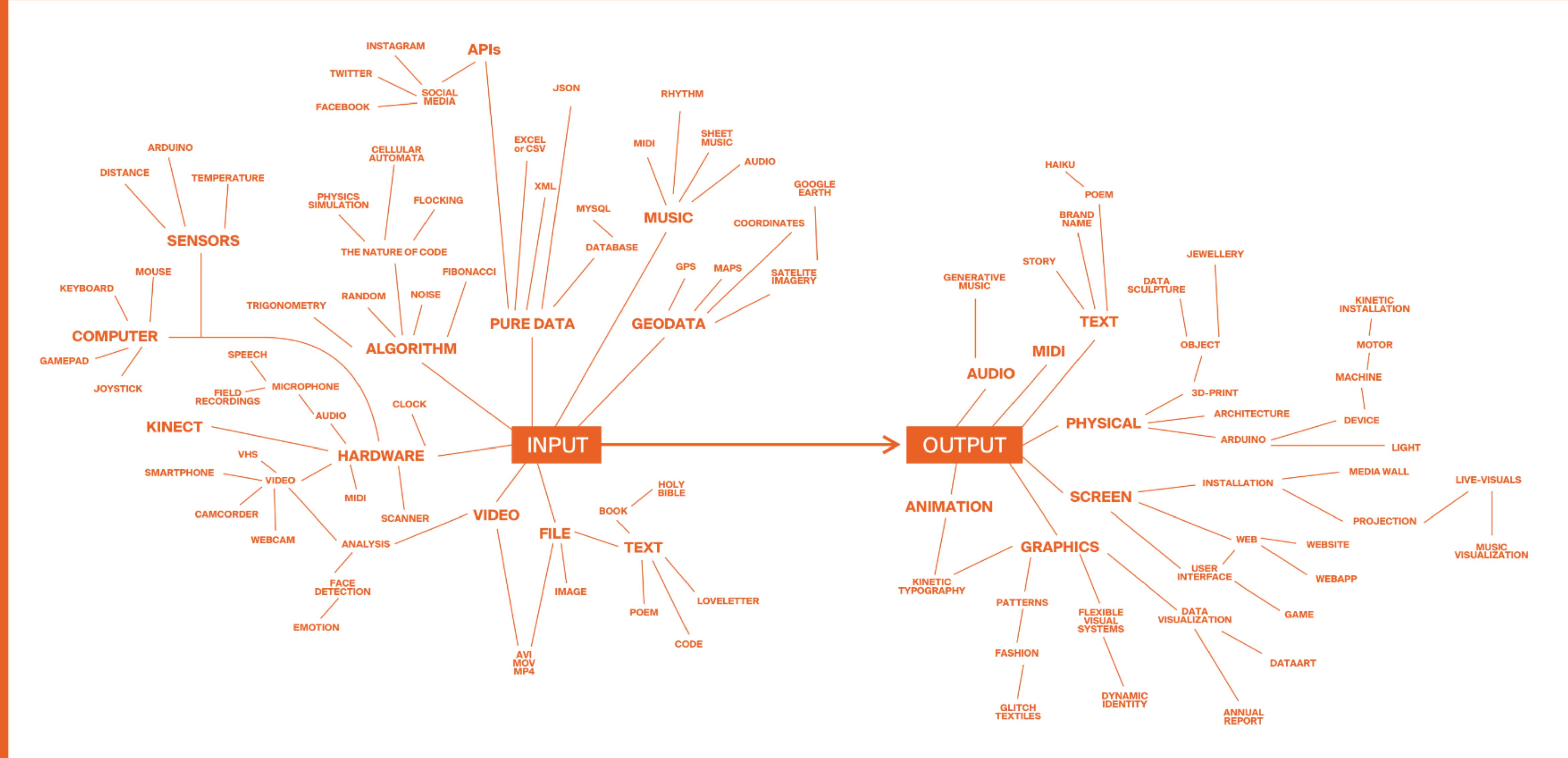
Face Painter

(mit Rudolfo Quintas)



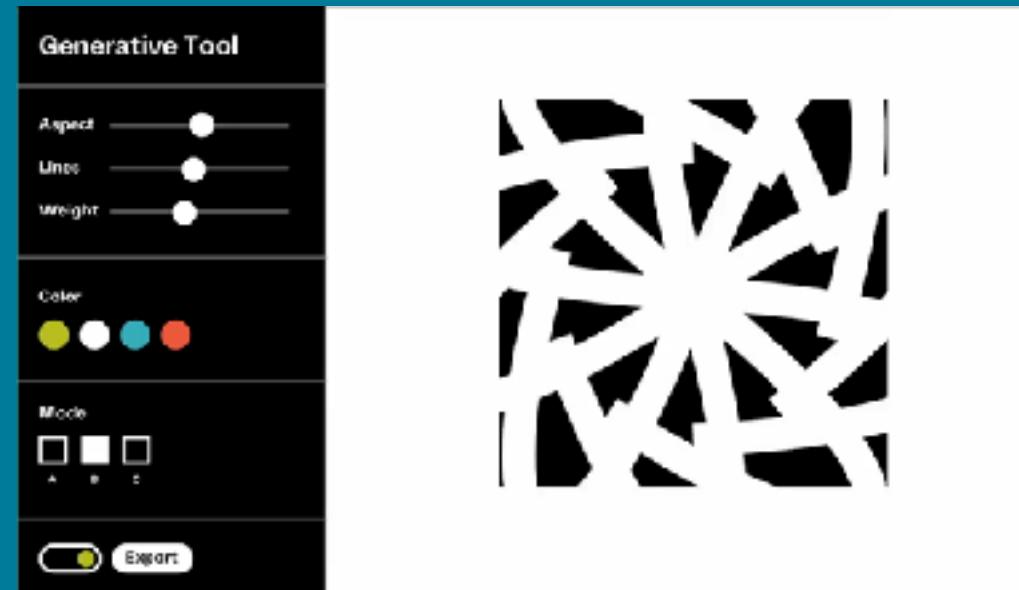
<https://nahuelgerth.de/tools/face-painter/>





→ explore new
frontiers through
creative coding

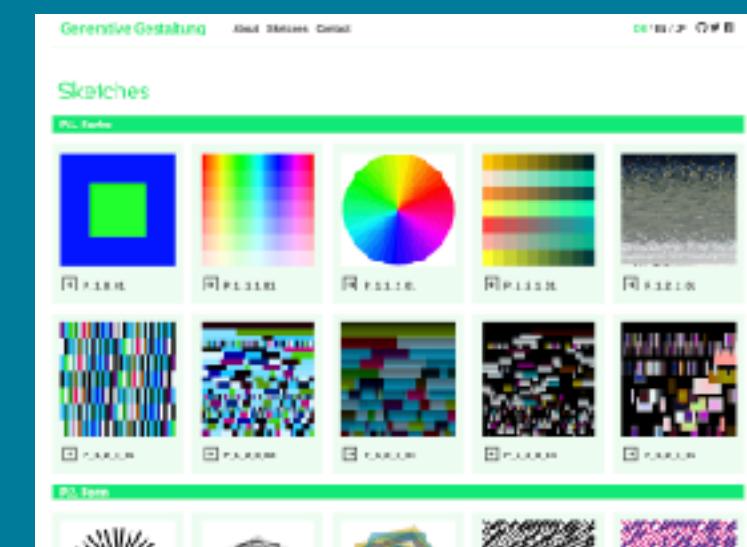
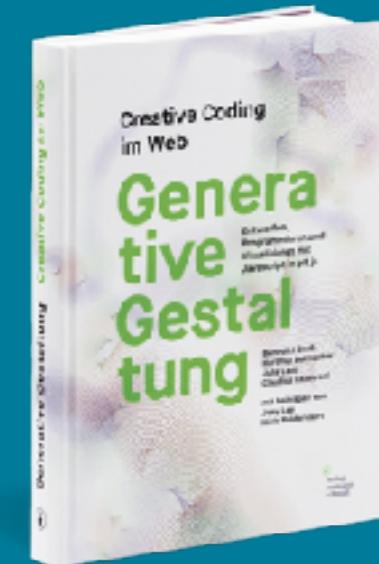
Lernen (p5.js)



AGD Workshop:
Generative Tools für Designer

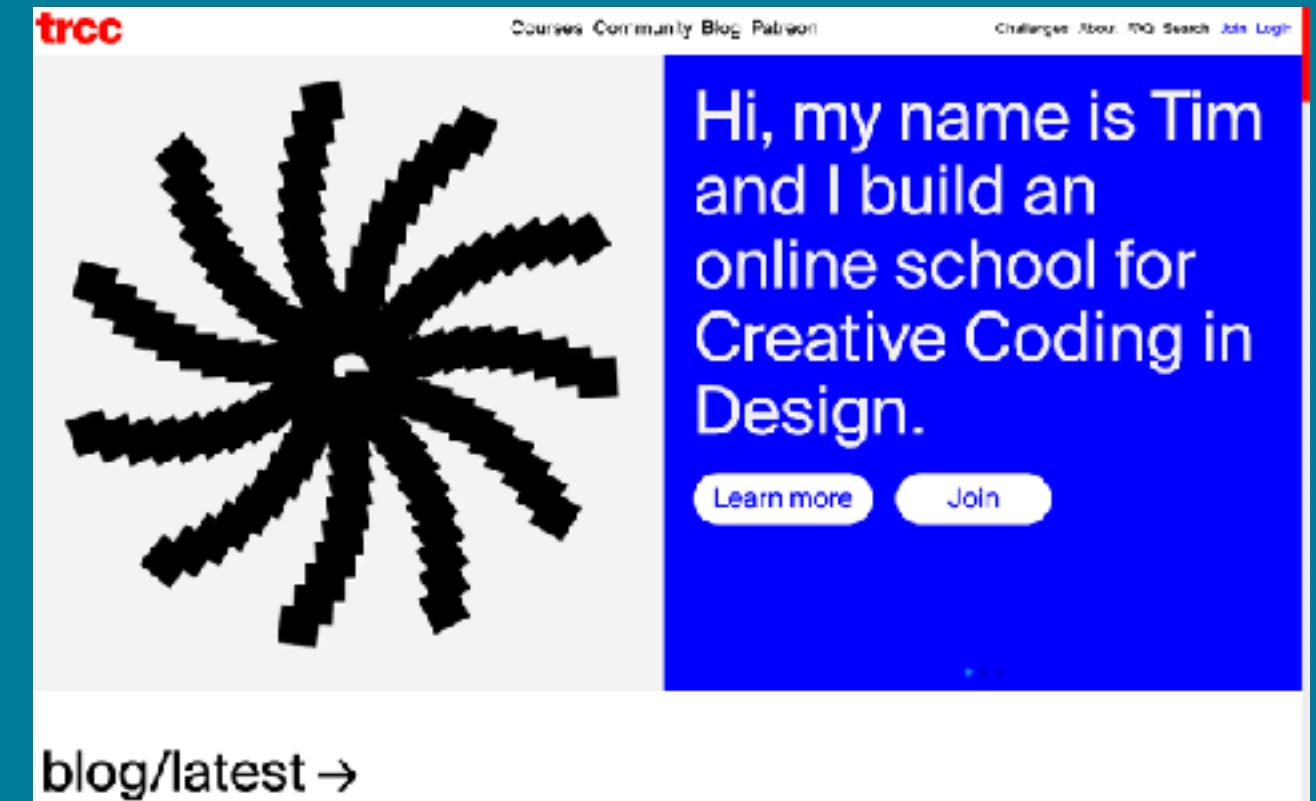
[https://agd.de/veranstaltungen/
einfuehrung-in-creative-coding/](https://agd.de/veranstaltungen/einfuehrung-in-creative-coding/)

Generative
Gestaltung



<http://www.generative-gestaltung.de/2/>

Tim Rodenbröker



blog/latest →

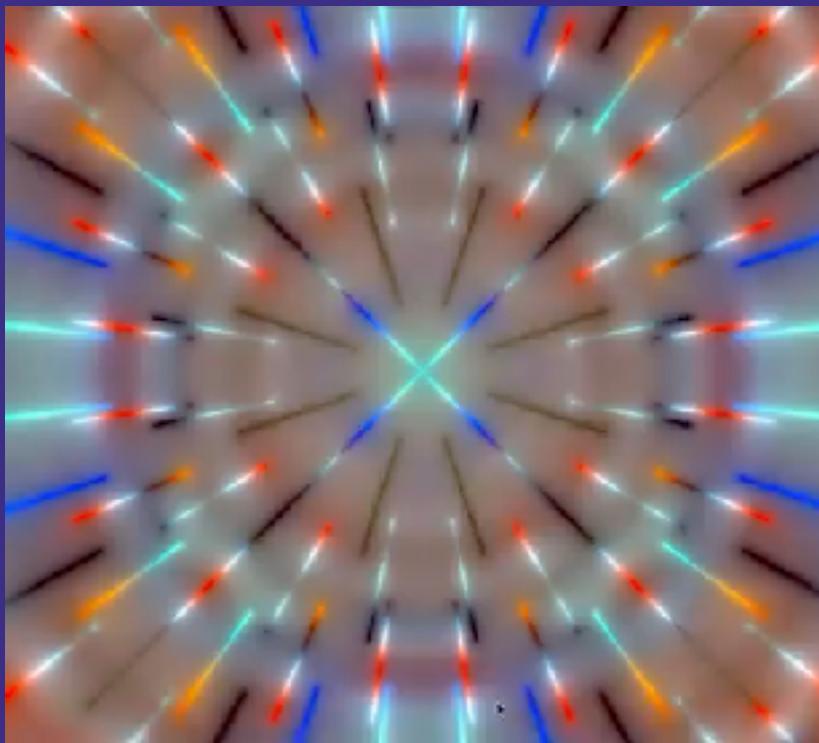
<https://timrodenbroeker.de/>

<https://natureofcode.com/>
<https://thecodingtrain.com/>



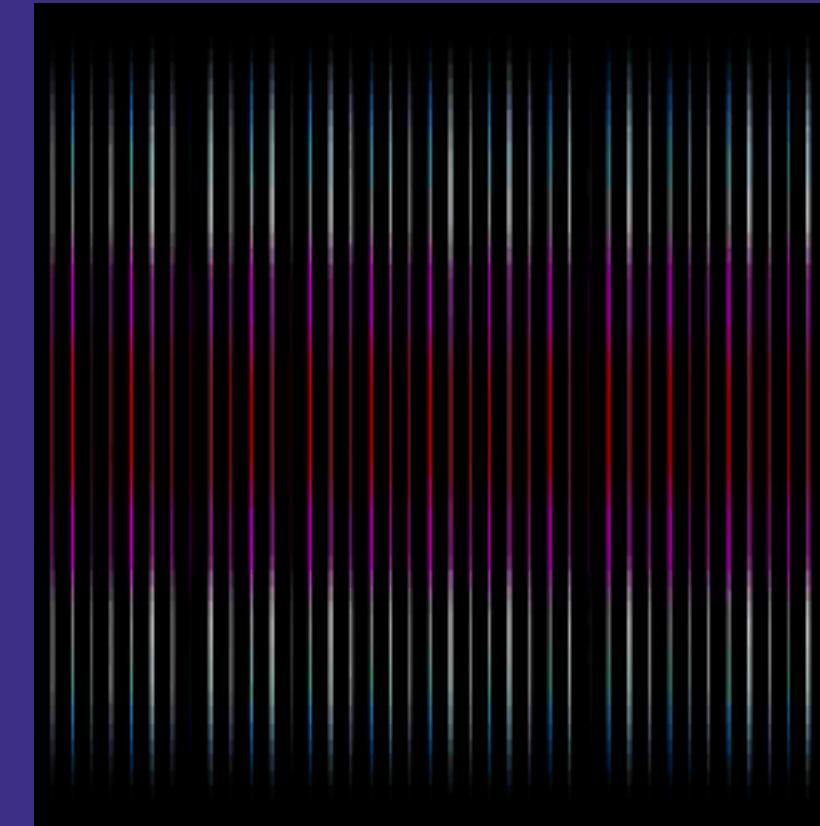
Coding Train &
Nature of Code





Zach Lieberman

instagram.com/zach.lieberman
Tägliche Code-Experimente



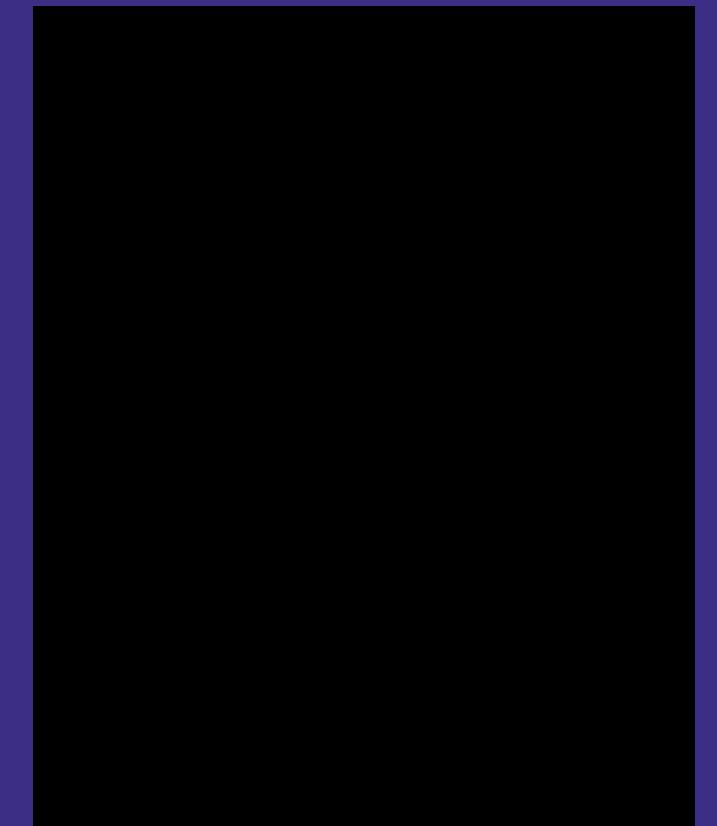
Vera van de Seyp

instagram.com/veravandeseyp
Typografie + Code



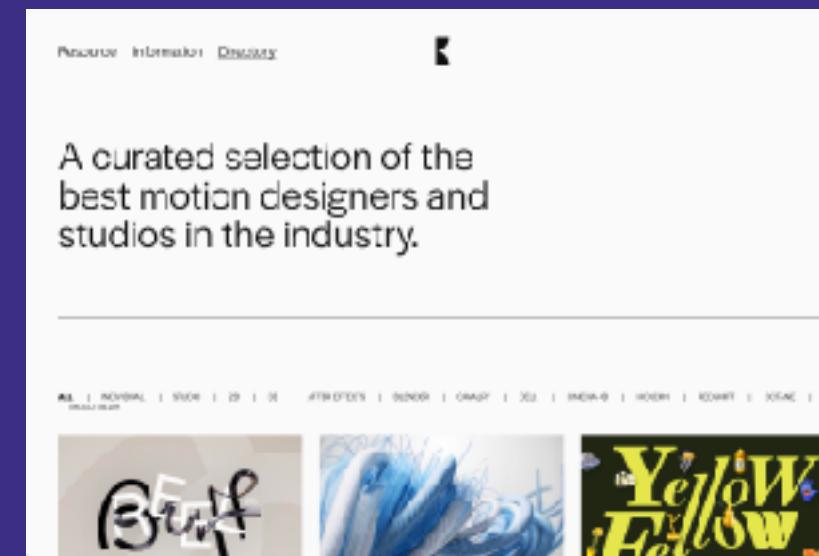
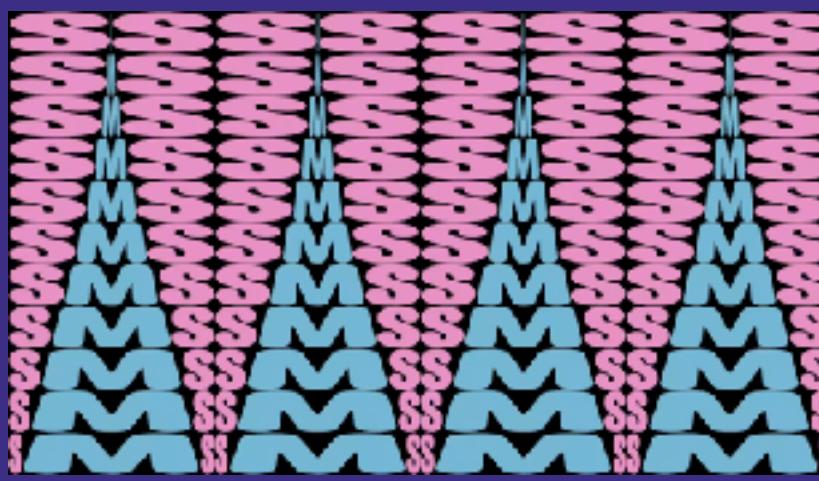
Stig Møller Hansen

instagram.com/stigmollerhansen
Klassisches Grafikdesign + Code



Studio Dumbar

studiodumbar.com
Motion Branding



DIA TV

dia.tv
Motion Branding

Patrick Hübner

patrik-huebner.com
Branding + Code

Brands in motion

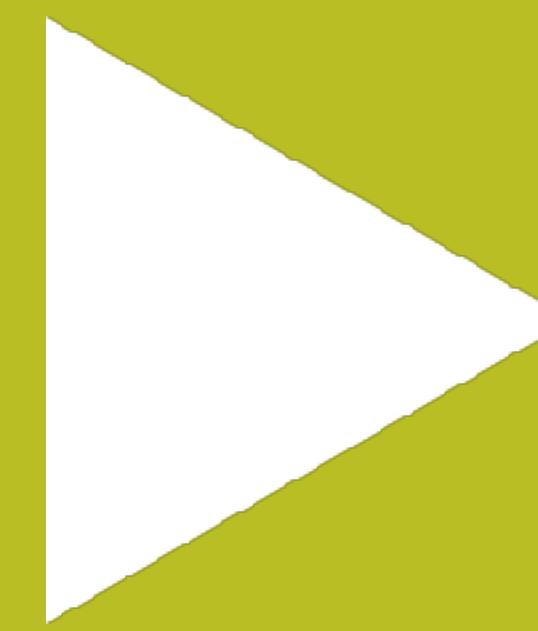
brandsinmotion.xyz
Alles rund um Motion Branding

Lava

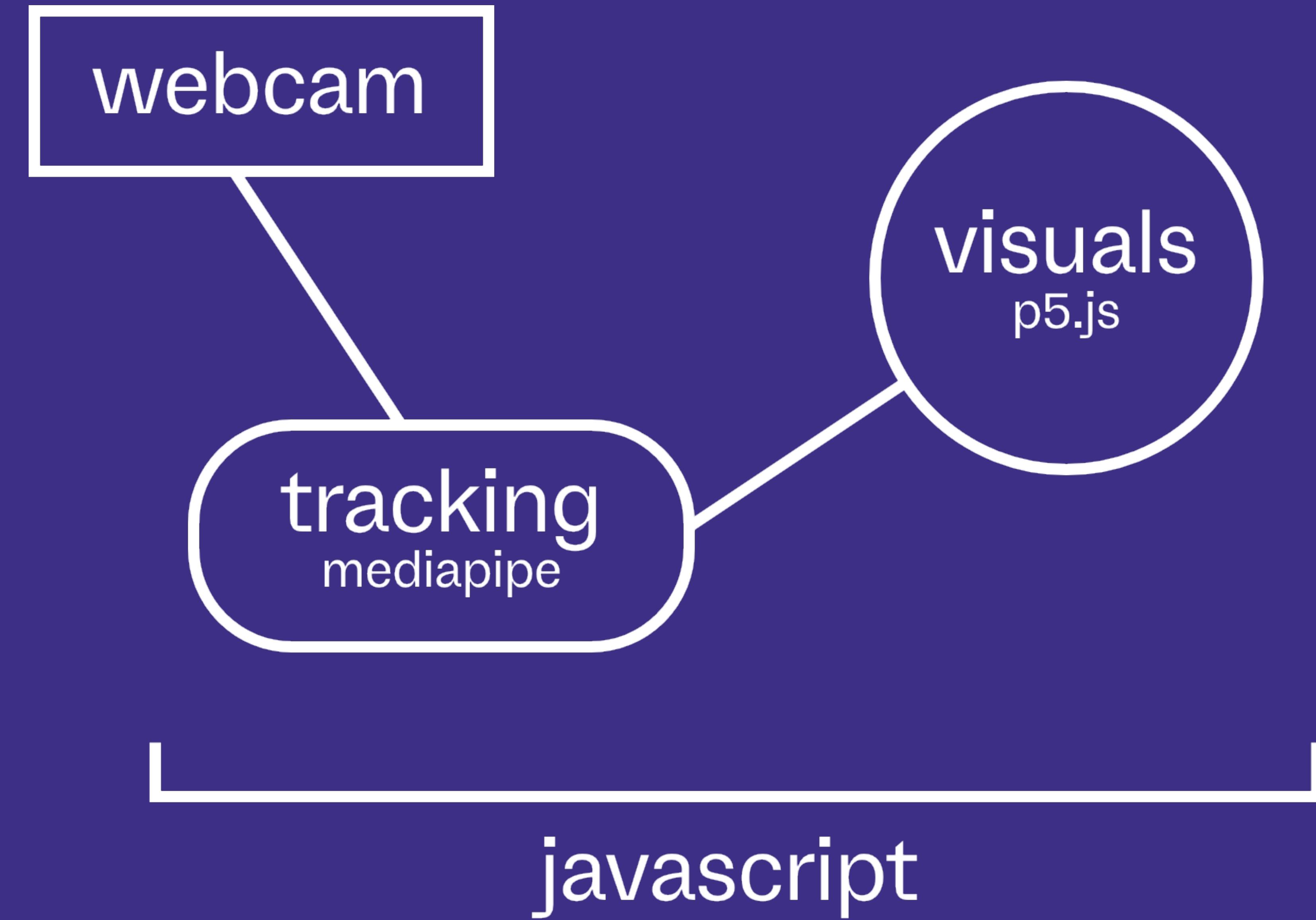
lava.nl
Motion Branding

Kurze Pause

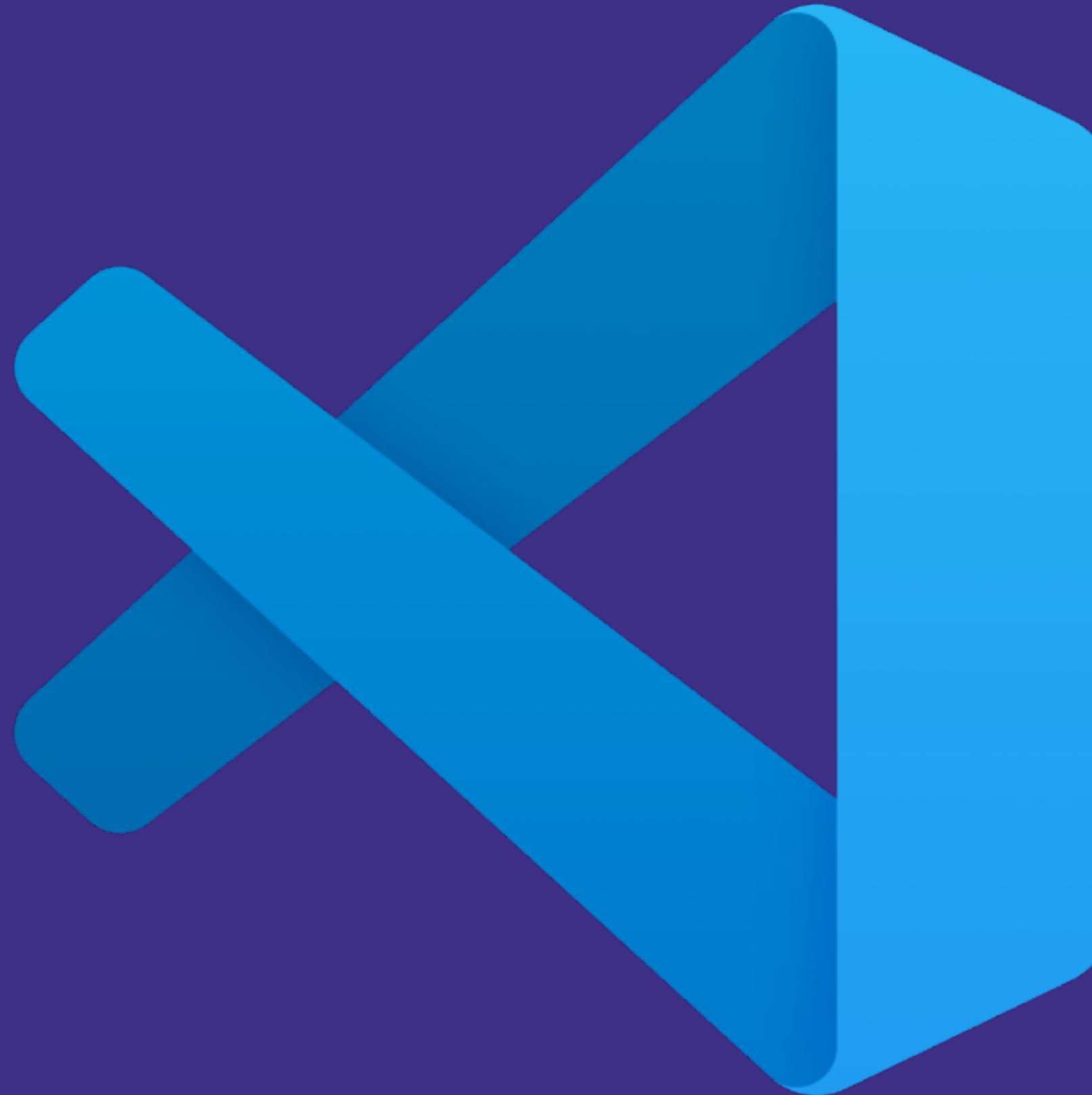




3 - p5.js Basics

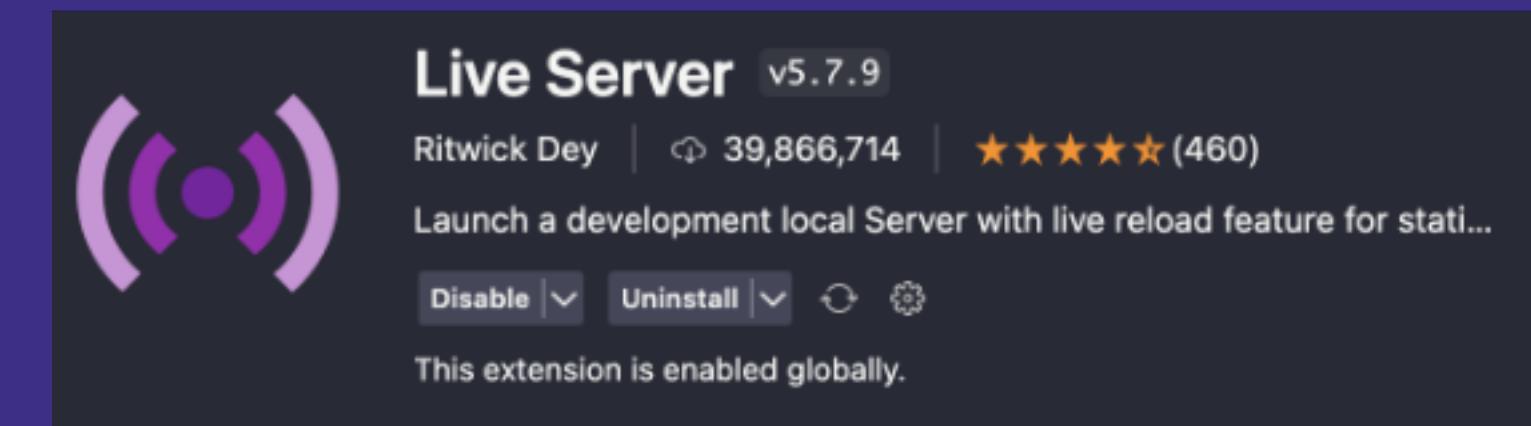


Workspace



- Installiere VS code
- Installiere eine Erweiterung
VS code/view/extensions

Live Server (Ritwick Dey)

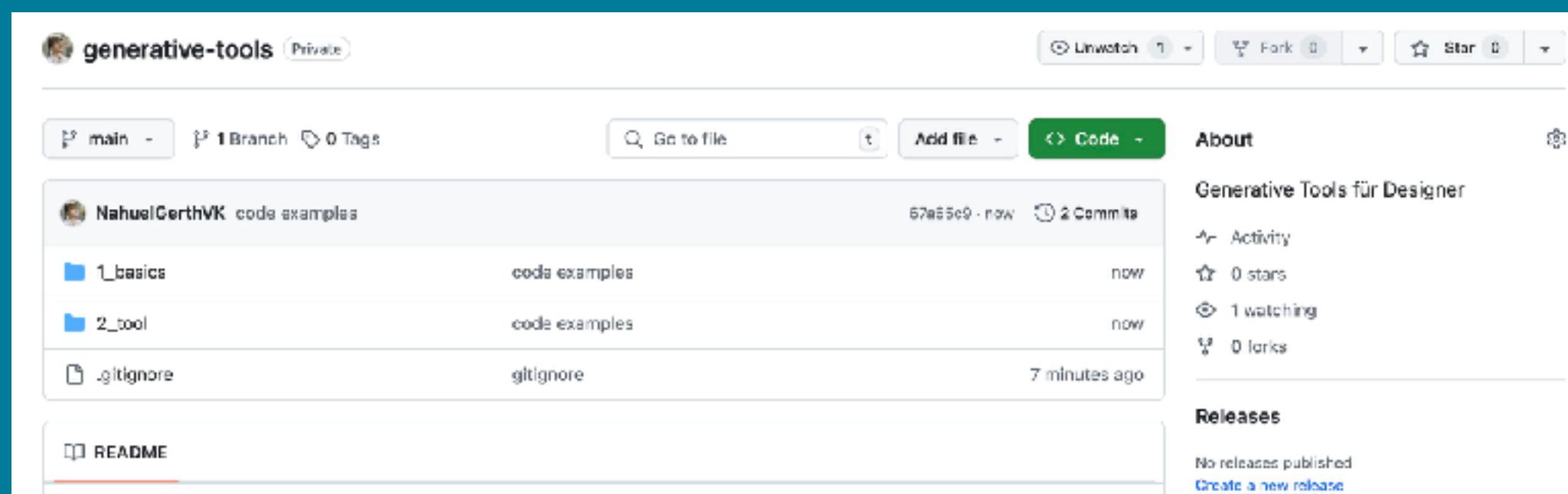


Unser Repository

→ lade unser Kurs-repository herunter (grüner Button → Download)

→ kopiere das erste Beispiel, benenne es um und öffne den gesamten Ordner in VS Code

→ klick auf "Go Live" für Vorschau im Browser



https://github.com/NahuelGerthVK/fhv_embodied_interaction

Dokumentation!

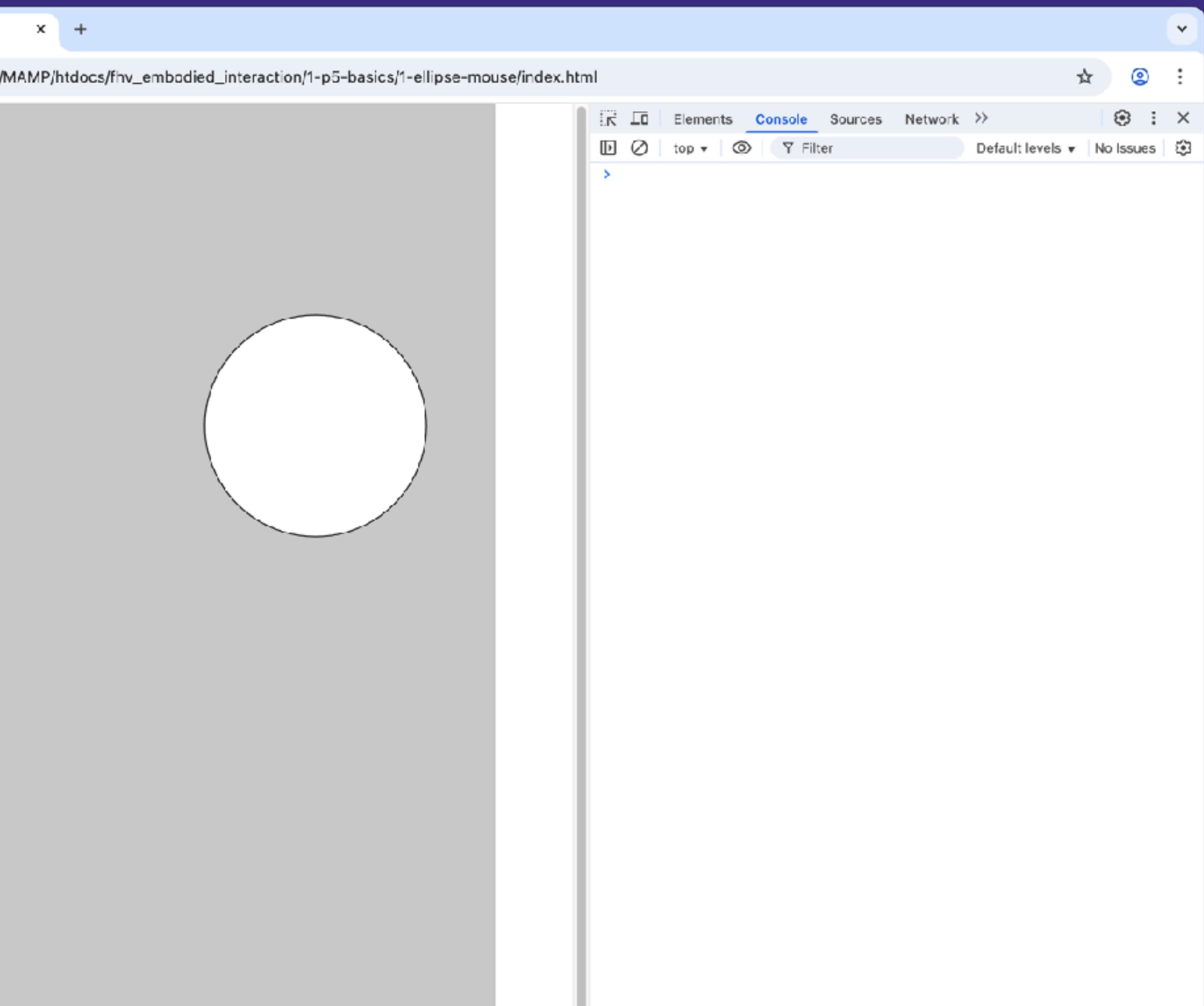
Mac

- Screenshot:
cmd + shift + 4
- Video:
cmd + shift + 5

Windows

- Screenshot:
Windows-Taste + shift + s
- Video:
?

Console

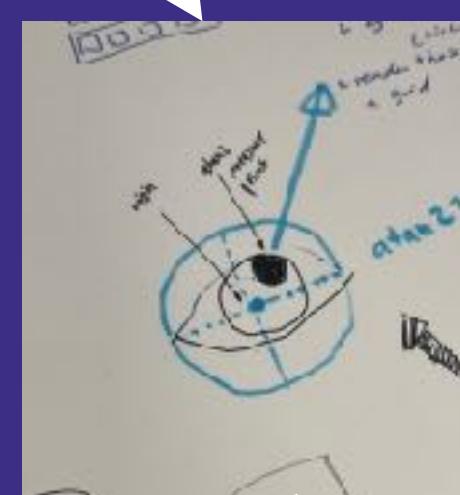


Chrome

- Rechtsklick / Inspect
(oder: option + cmd + I)
- Dann auf den Tab “Console”



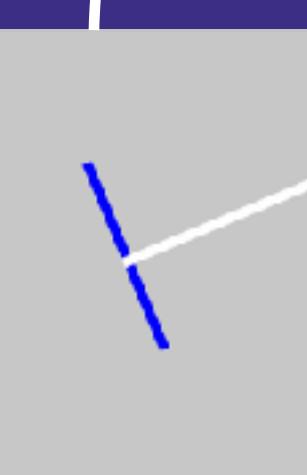
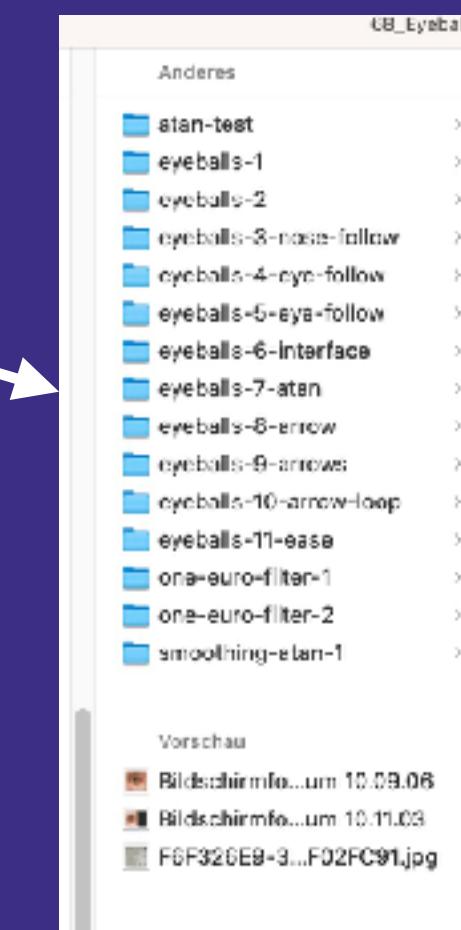
unausgereifte Idee



atan2?



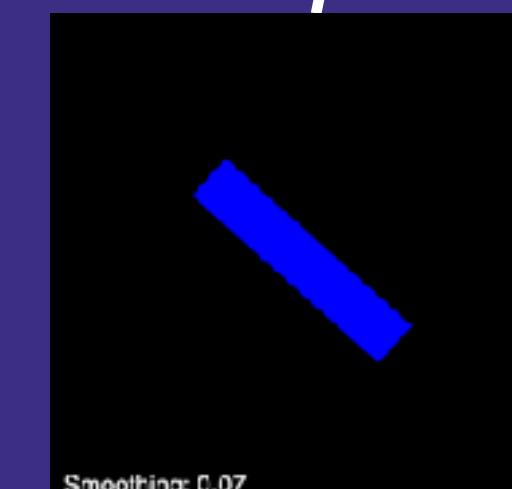
Tracking Punkte



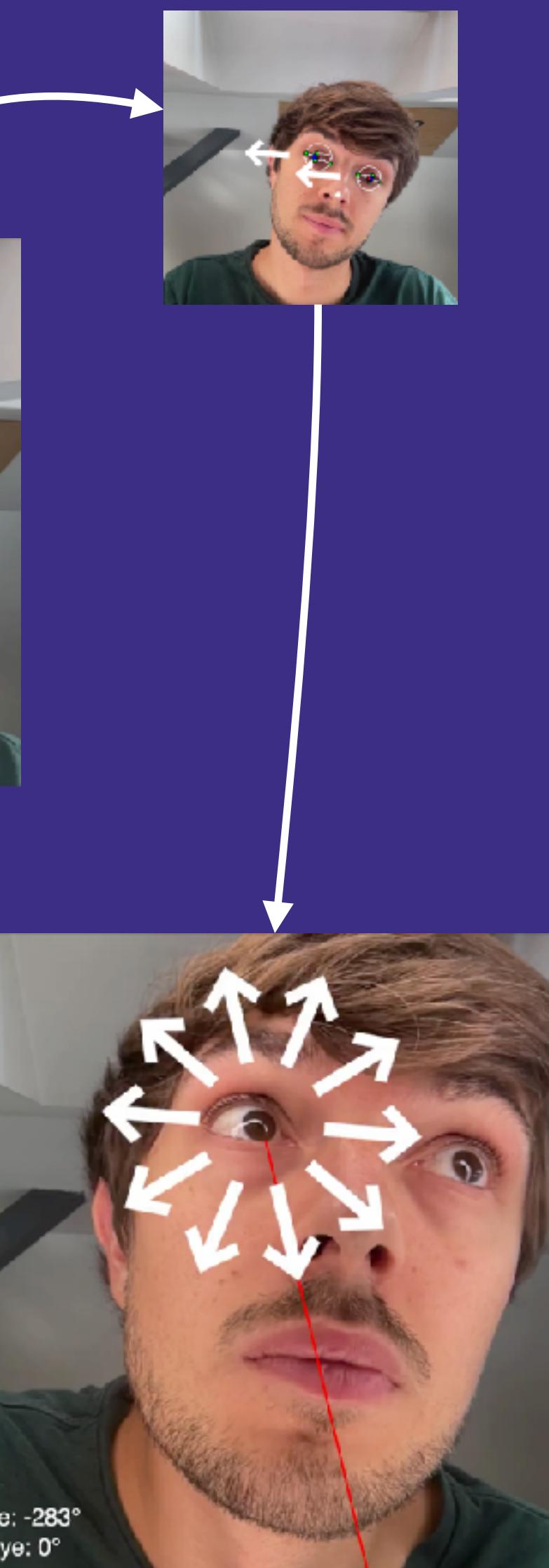
atan2 test



Wackelaugen?



etwas Smoothing!



meeeehr Pfeile!

our working file

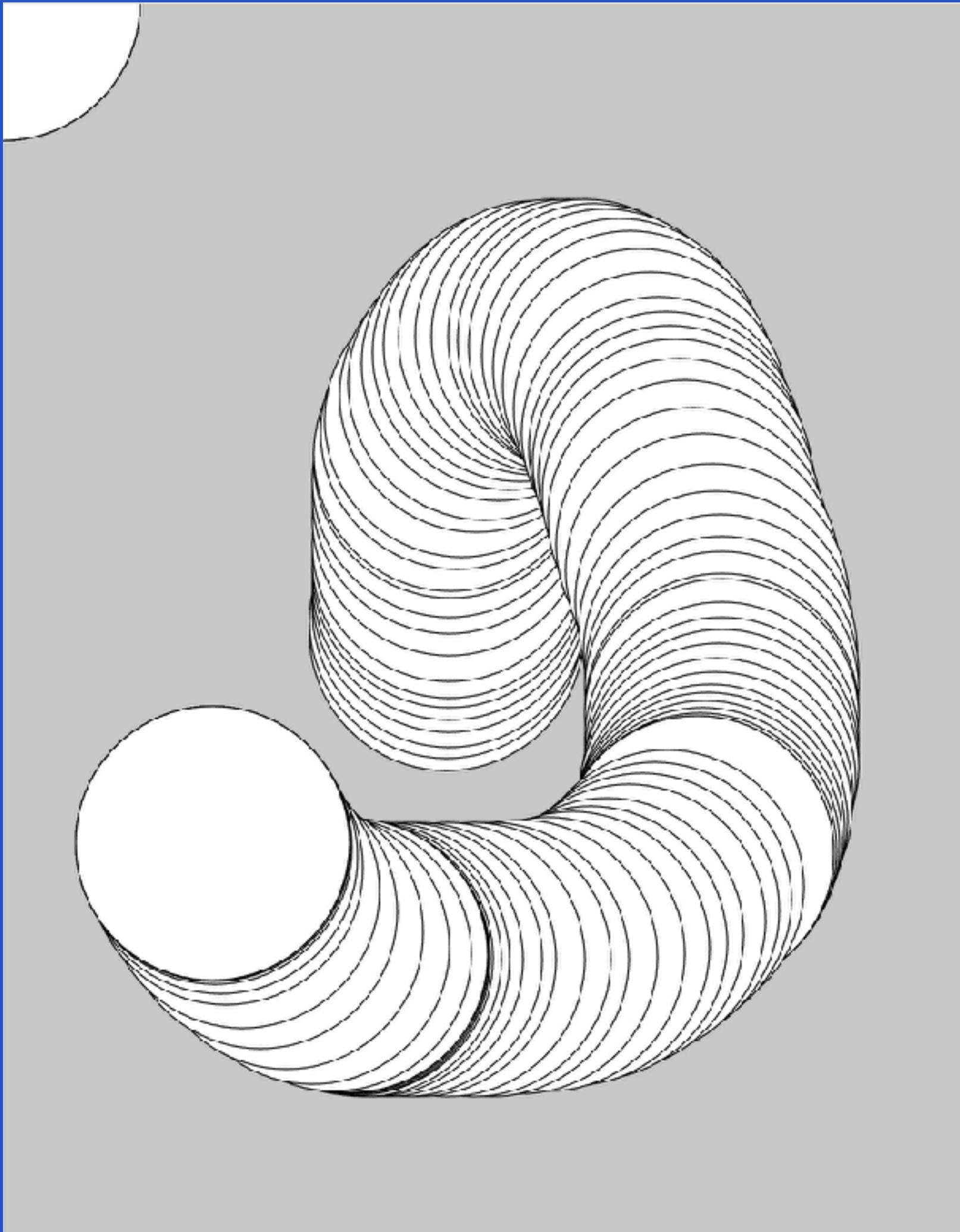
The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar containing files: '231024_1' (libraries, index.html, jsconfig.json, sketch.js, style.css), 'OUTLINE', and 'TIMELINE'. The main area shows the 'sketch.js' file with the following code:

```
sketch.js > setup
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7 }
8
```

A white circle highlights the code area. A callout points from the text 'here goes our code' to this circle.

In the bottom right corner of the editor, there is a circular icon with a play button symbol and the text 'Go Live' next to it. A white circle highlights this icon, and a callout points from the text 'live preview in browser' to it.

1-ellipse-mouse

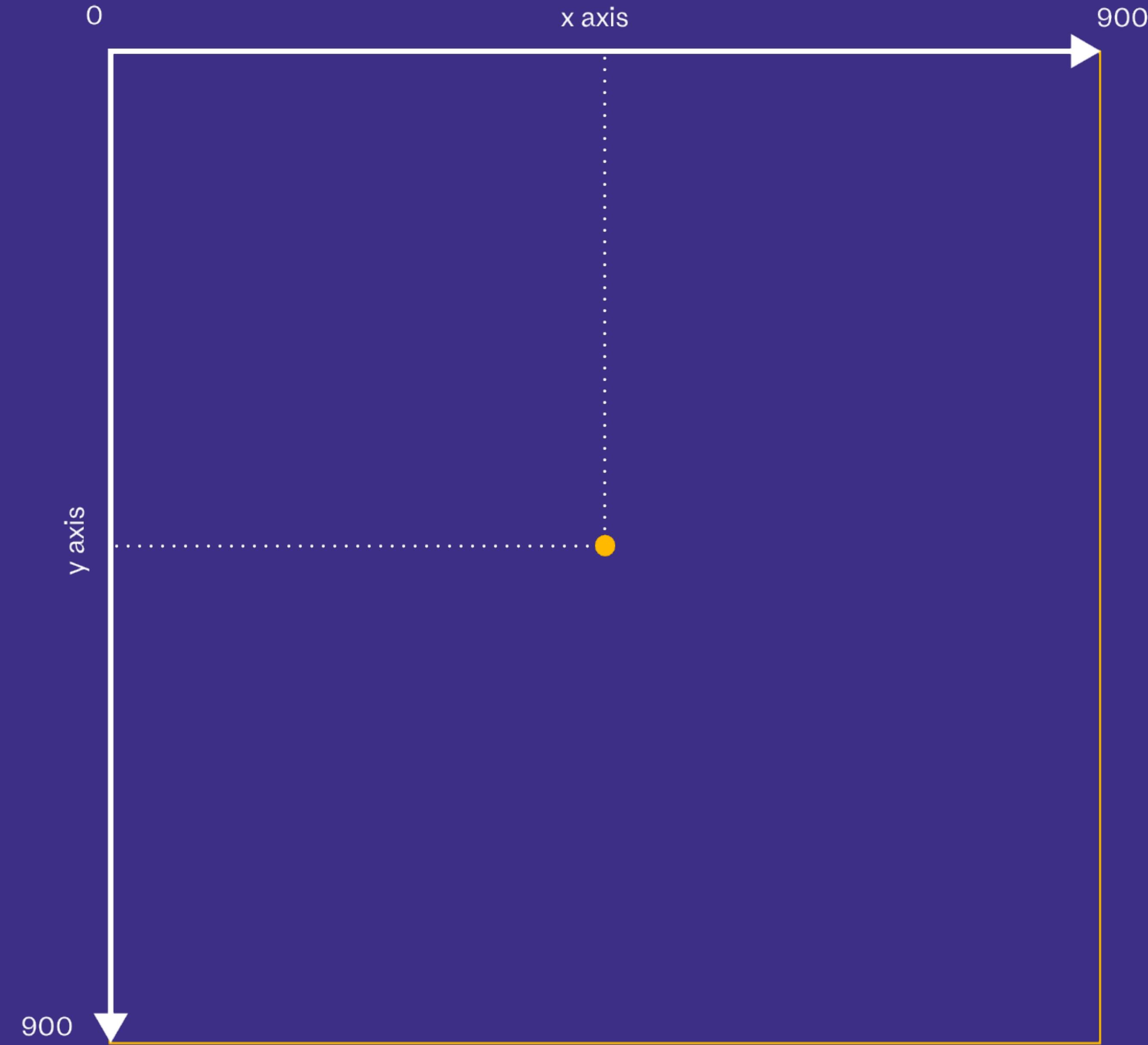


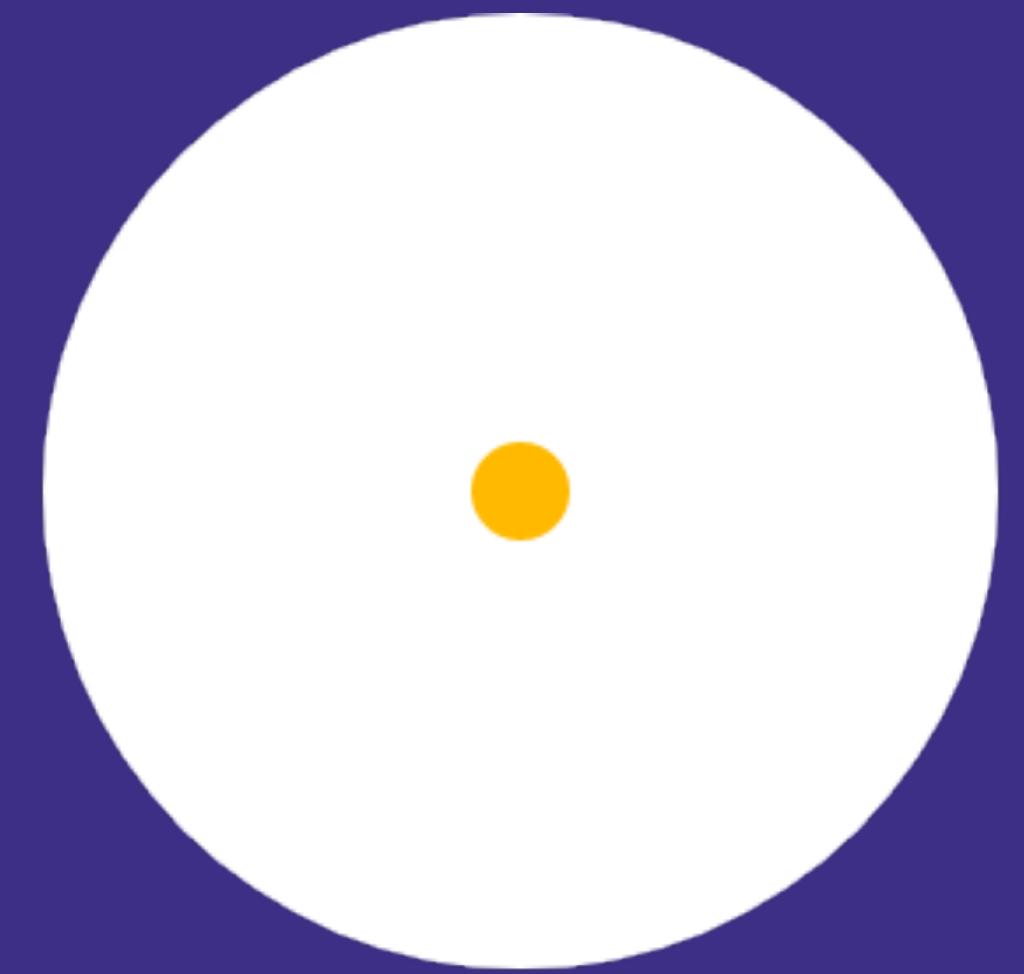
Beispiel 1:

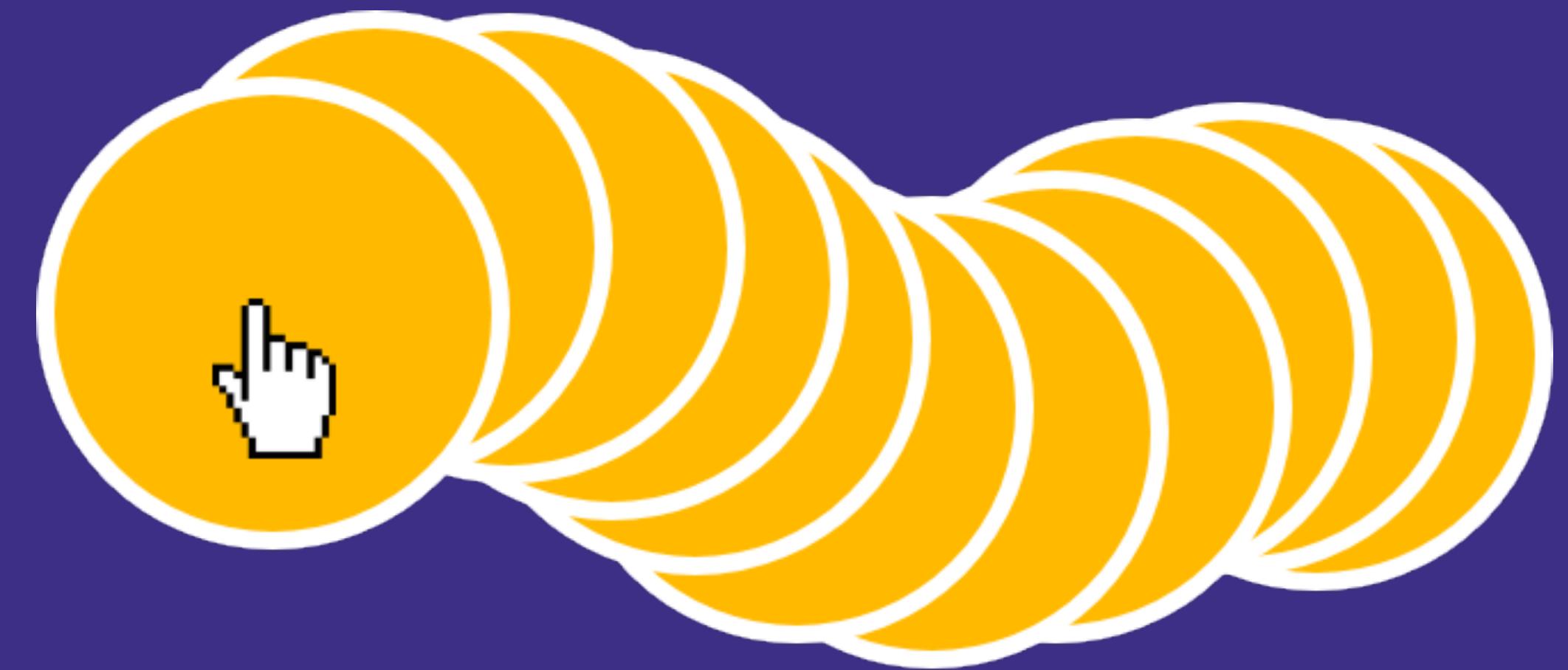
- gesamten Ordner in VS Code öffnen
- klick auf "Go Live" für Vorschau im Browser

☞ coordinate system

point (450, 450);
point (width/2, height/2);



 ellipse function`ellipse(x, y, w, h);`

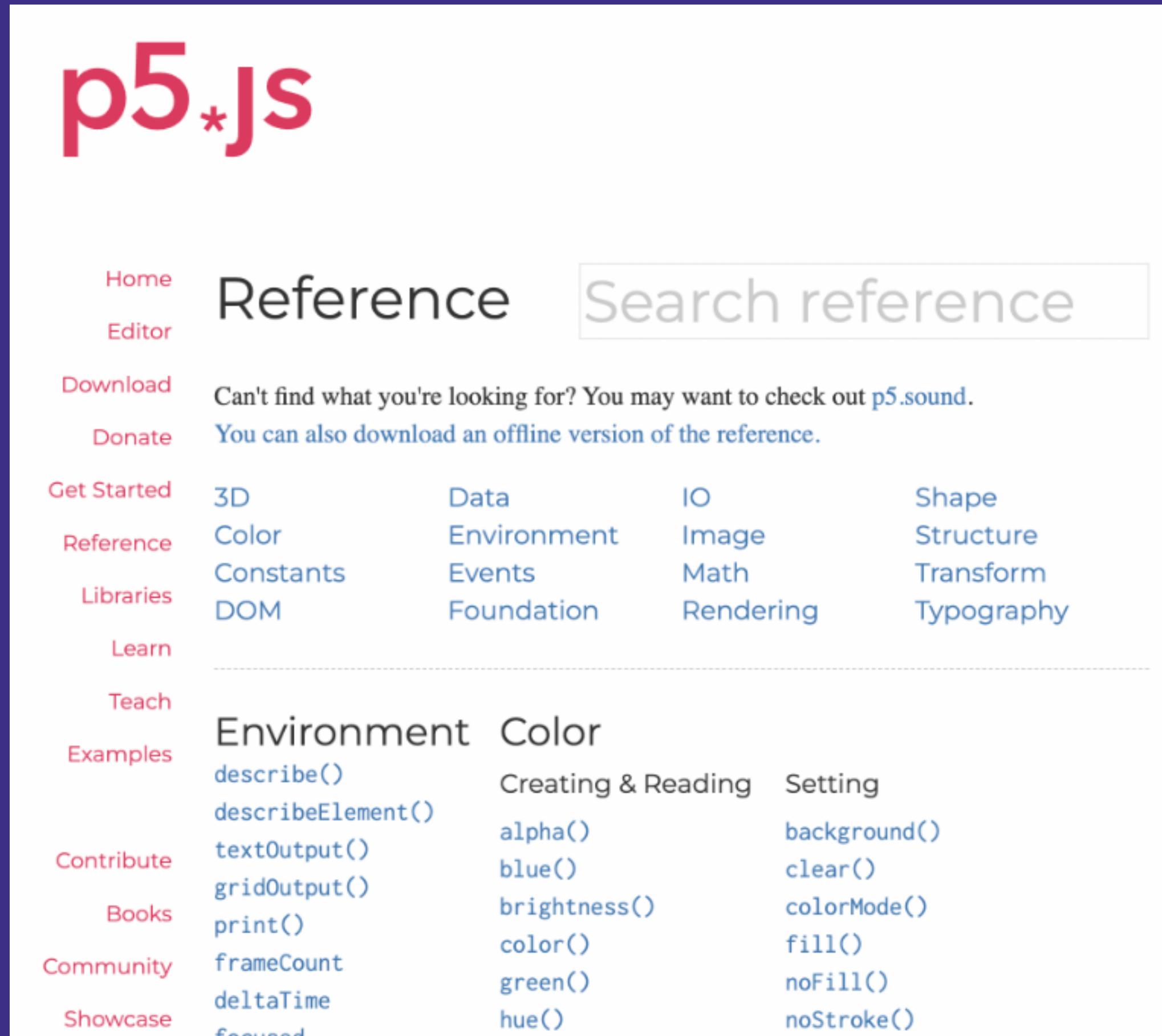
 interactivity

```
// background("black");
ellipse(mouseX, mouseY);
```

 basics

- `setup()`
runs only once
- `draw()`
loops continuously
(60 times per second)
- `createCanvas()`
our window size
(width and height)
- `// comment`
ignored by the program

👉 p5.js reference



The screenshot shows the p5.js Reference website. At the top left is the p5.js logo. On the left side, there's a sidebar with links: Home, Editor, Download, Donate, Get Started, Reference, Libraries, Learn, Teach, Examples, Contribute, Books, Community, and Showcase. The main content area has a "Reference" heading and a search bar. Below the search bar is a section titled "Environment Color" with a grid of functions:

	Environment	Color
describe()	Creating & Reading	Setting
describeElement()	alpha()	background()
textOutput()	blue()	clear()
gridOutput()	brightness()	colorMode()
print()	color()	fill()
frameCount	green()	noFill()
deltaTime	hue()	noStroke()
focused		

☞ order of the code

```
draw blue background;
```

```
    draw a rectangle;
```

```
        take green color;
```

```
            draw a circle;
```

```
                take blue color;
```

```
                    draw a triangle;
```

→ last line gets rendered on top (like the highest Photoshop layer)

→ we draw inside the **draw()** loop

→ always close your brackets!
() [] {}

→ separate functions with semicolons
;

 Aufgabe (20 mins)

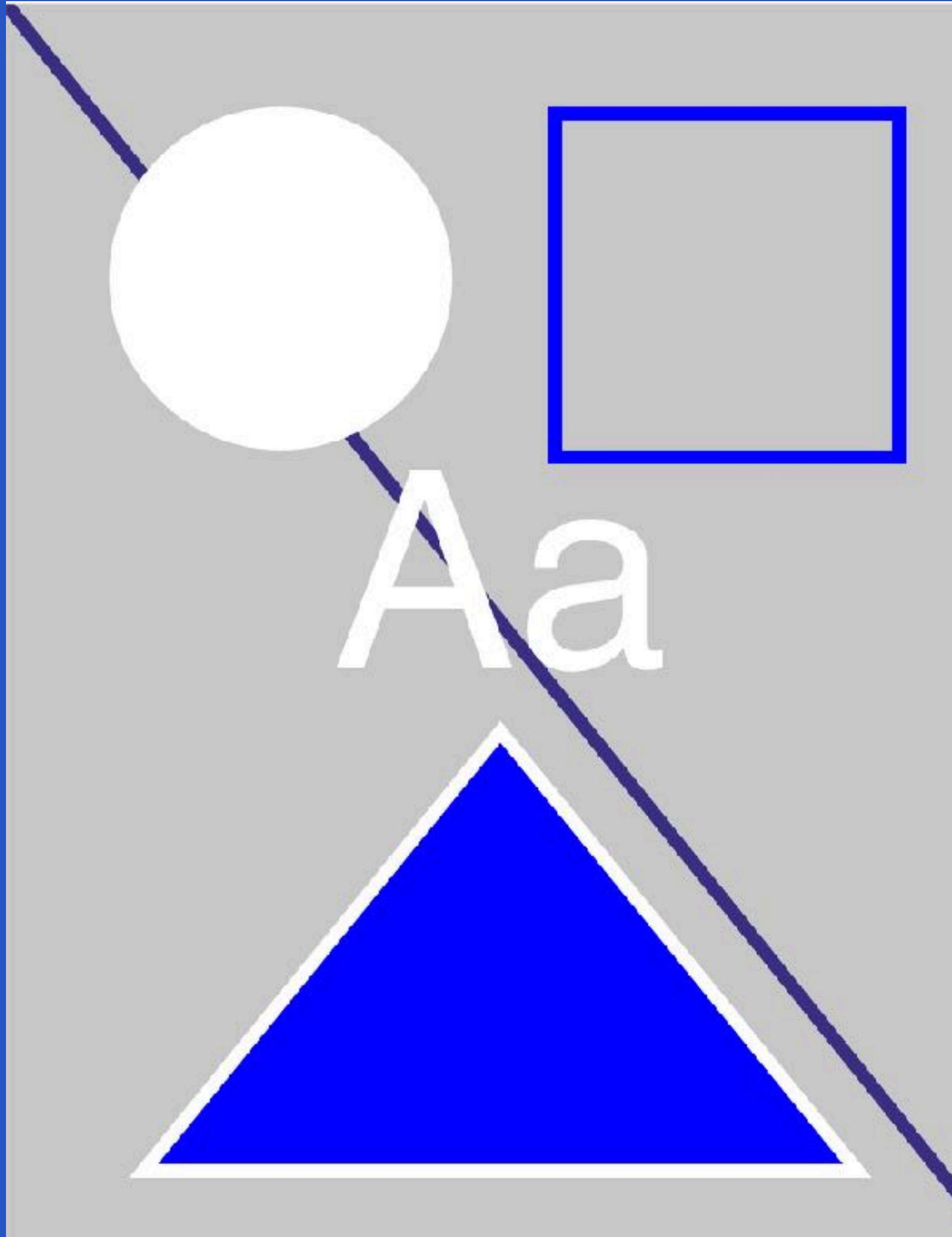
copy example
01_follow_mouse



change color, stroke
and size of ellipse



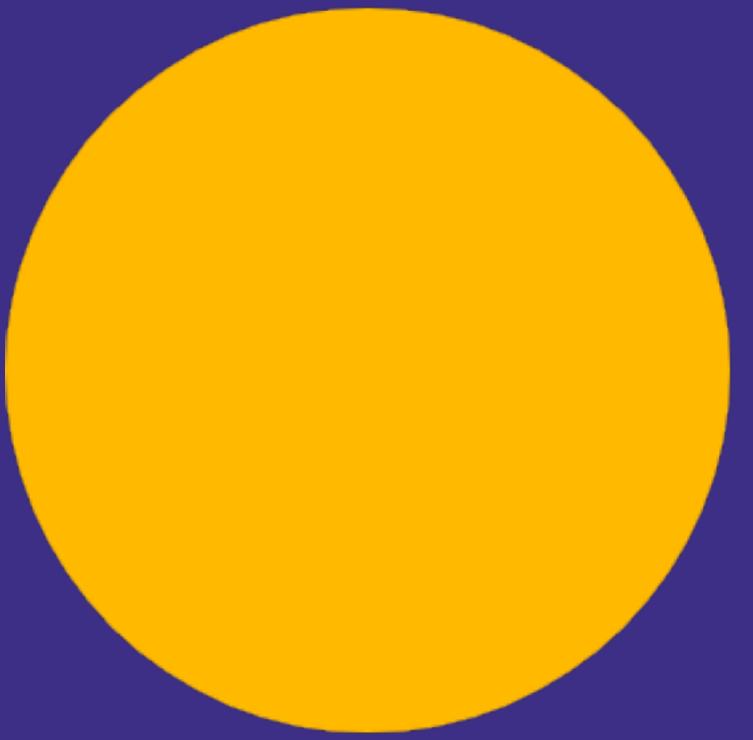
play

 2-shapes-and-stylingTesten

- verschiedene Formen zeichnen
- Formen platzieren und stylen

 styling

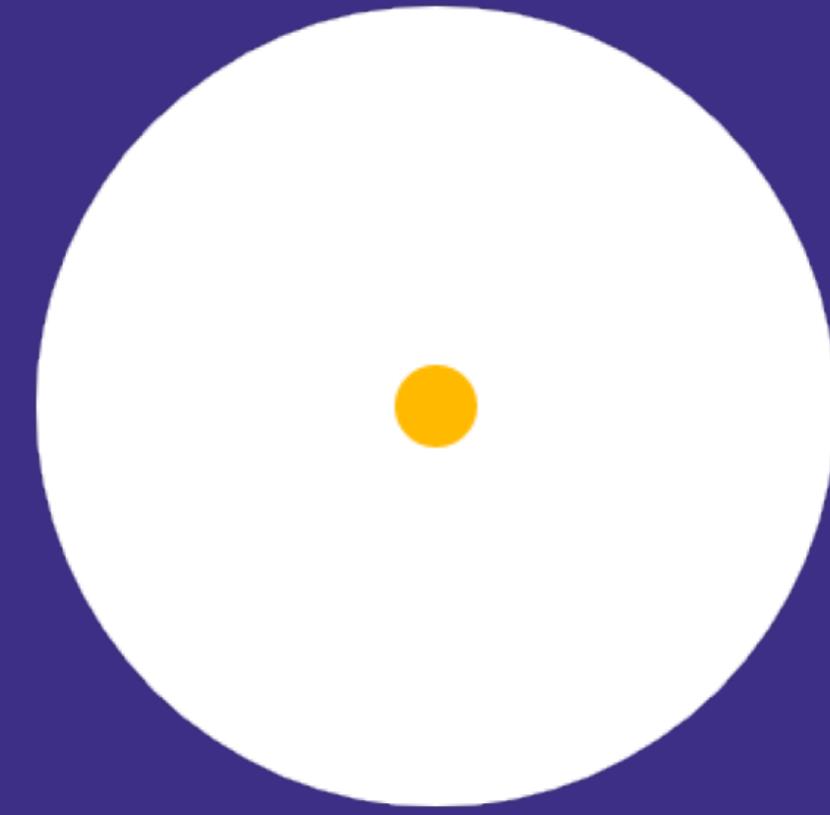
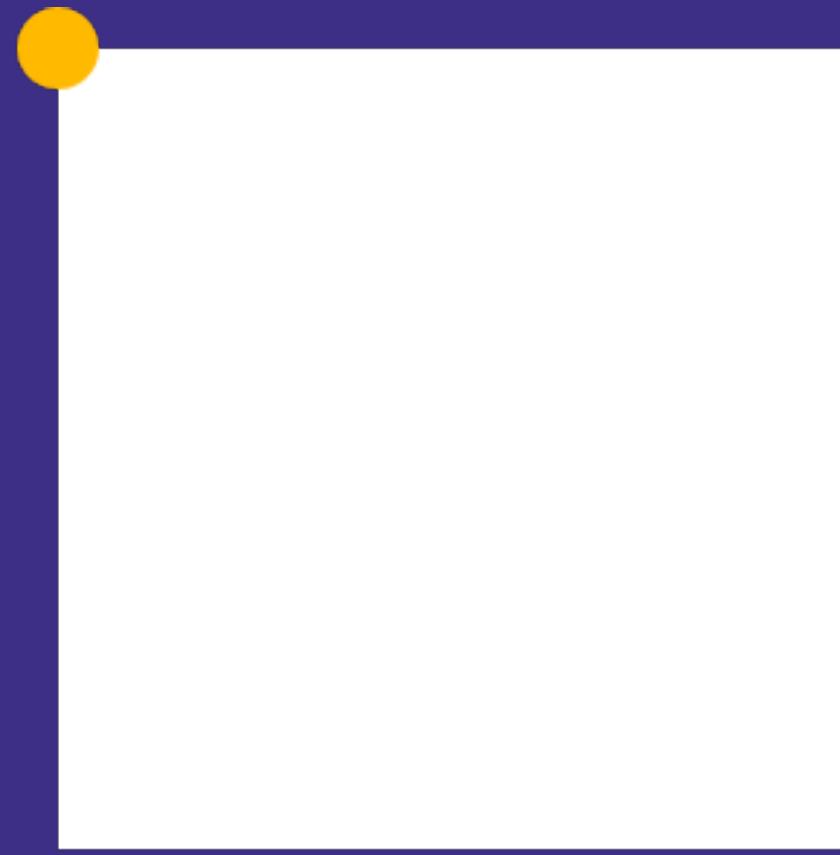
hex color picker
<https://g.co/kgs/9h4Esb>

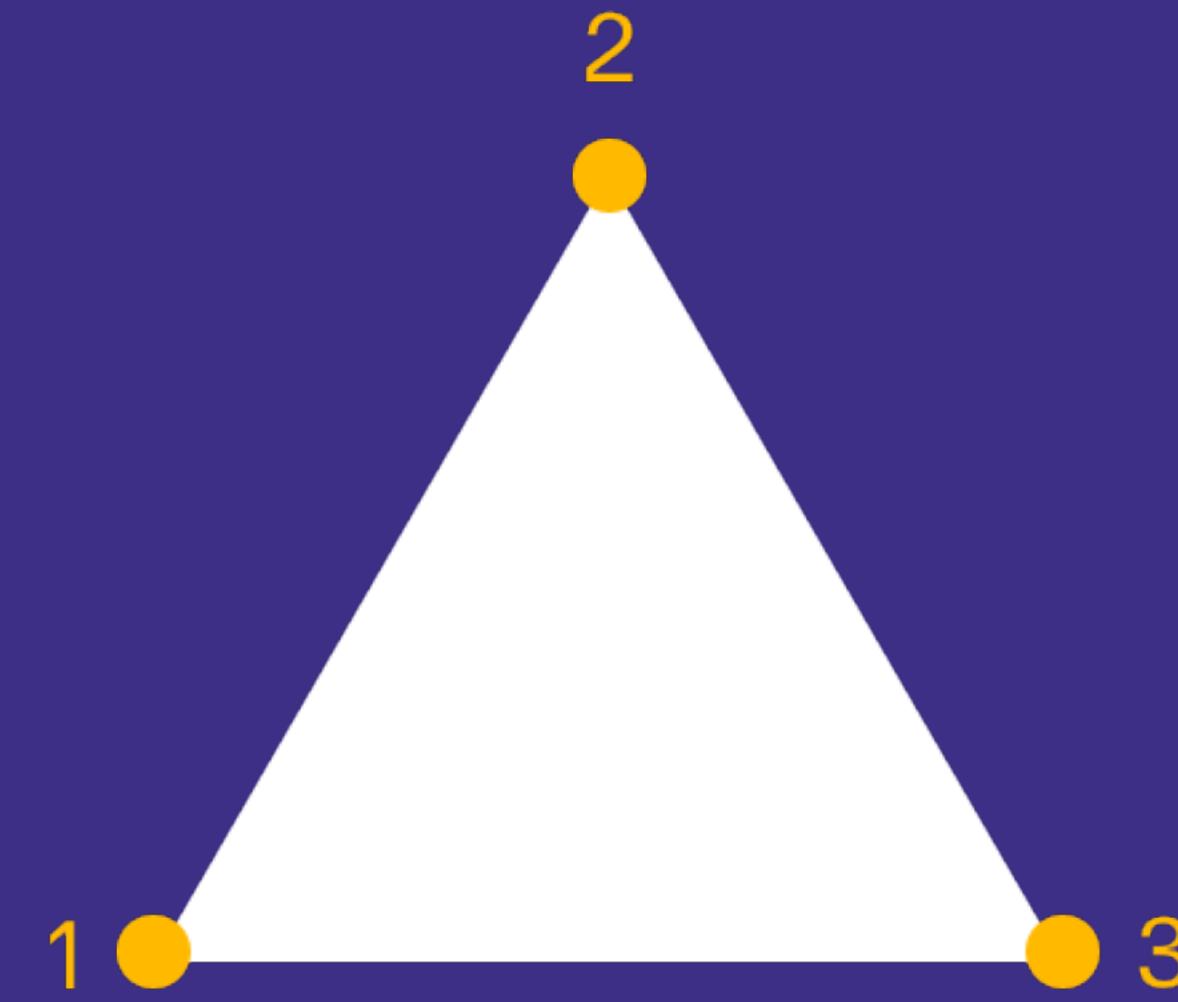


```
fill("#FFBA00");  
noStroke();
```



```
stroke("#FFBA00");  
strokeWeight(20);  
noFill();
```

 geometric shapes`ellipse(x, y, w, h);``rect(x, y, w, h);`

 geometric shapes

`line(x1, y1, x2, y2);`

`triangle(x1, y1 ,x2, y2, x3, y3);`

 text function

Aa

```
text("Aa", x, y);
```



text styling

→ size

```
textSize(20);
```

→ fill

```
noFill();  
fill("#FFBA00");
```

→ alignment (x, y)

```
textAlign(CENTER, CENTER);  
textAlign(LEFT, TOP);  
textAlign(RIGHT, BOTTOM);
```

→ stroke

```
noStroke();  
noFill();  
stroke("#FFBA00");  
strokeWeight(20);
```

👉 custom font

- **save** your font to your sketch **folder**
- create a **variable** to hold the font
- **preload()** load the font from your folder
- **textFont()**
 - set the font before drawing text

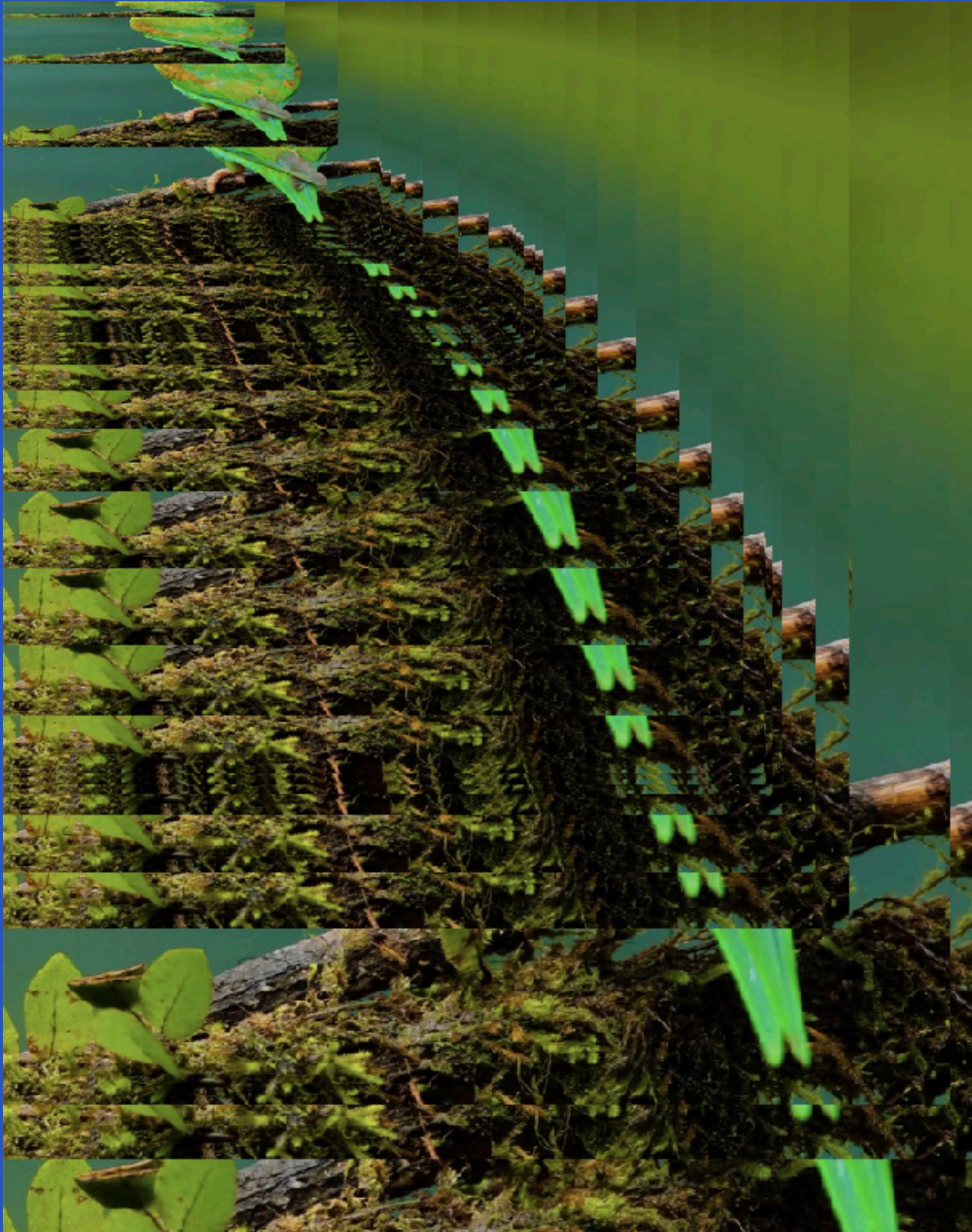
```
let myFont;

function preload(){
  myFont = loadFont("fonts/my-nice-font.ttf");
}

function setup() {
  createCanvas(900, 900);
  textFont(myFont);
}

function draw() {
  background(255);

  text("Neat text in nice font", 100, 0, 900, 900);
}
```

 3-imageTesten

- eigene Bilder platzieren
- transparentes .png einsetzen
- Schnittstelle von Photoshop / Illustrator zu Code

👉 image function



image(myImage, x, y, w, h);

☞ image function

- **save** your image to your sketch **folder**
- create a **variable** to hold the image
- **preload()**
load the image from your folder
- **draw()**
place the image on the canvas

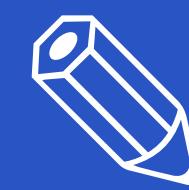
```
let myImage;

function preload(){
  myImage = loadImage("my-nice-image.jpg");
}

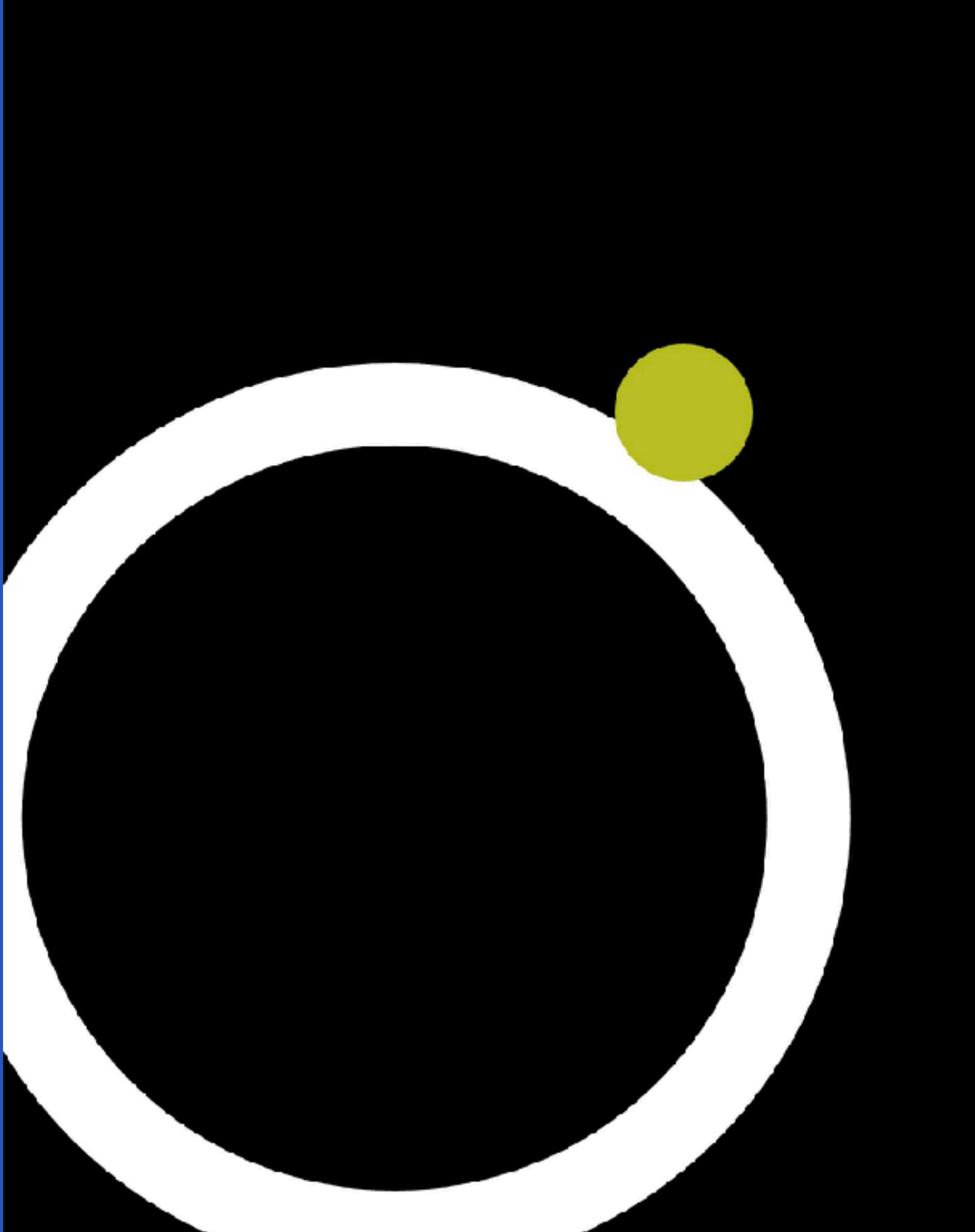
function setup() {
  createCanvas(900, 900);
}

function draw() {
  background(255);

  image(myImage, 0, 0, 900, 900);
}
```



4-mouse-interaction

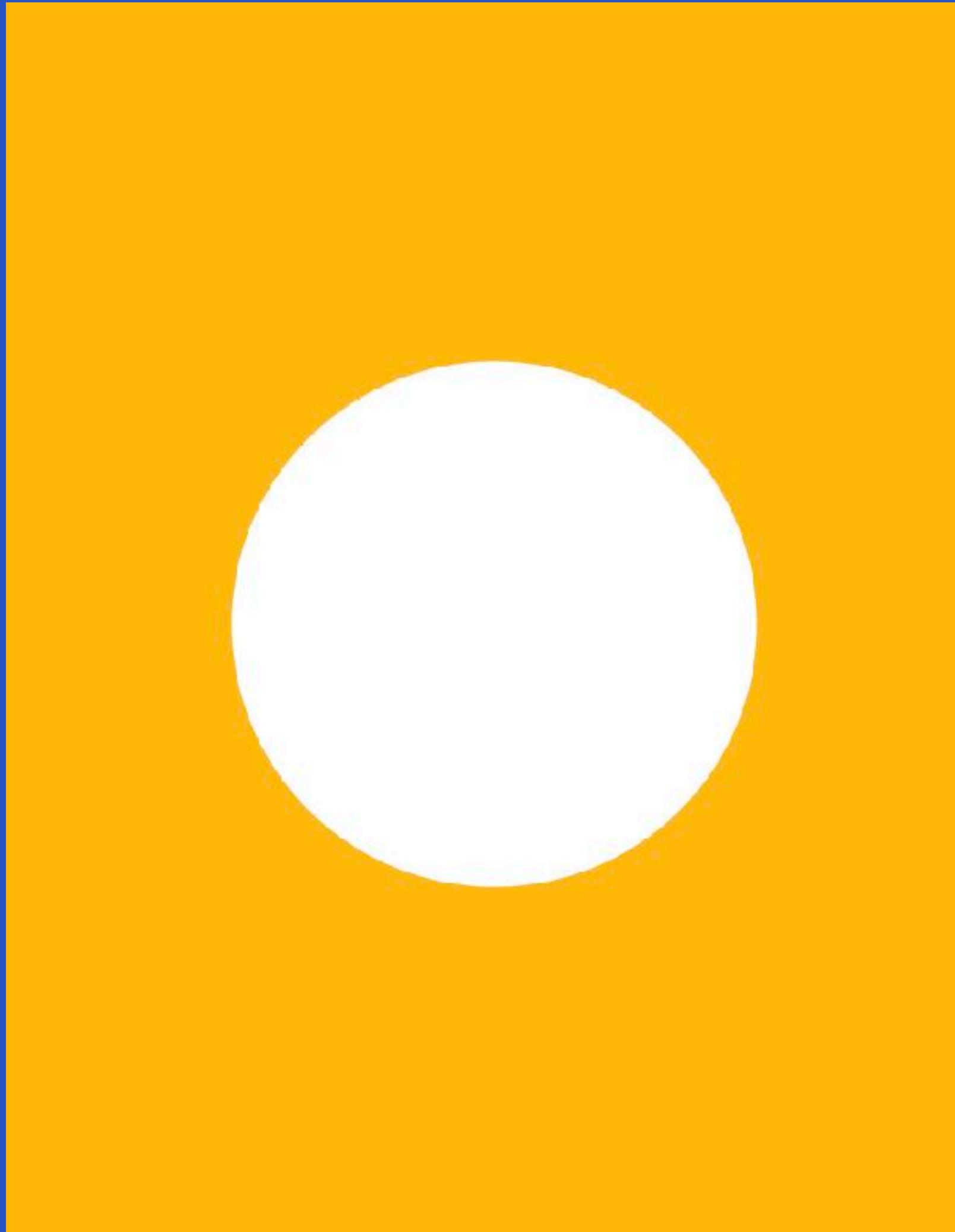


Testen

- was könnte man alles mit der Maus machen?
- im Prinzip können wir jeden Zahlenwert auch über die Maus steuern (Größe, Dicke, Position, etc.)
- wir können die Mauswerte auch anpassen, z.B. halbieren, oder anderes: `mouseX * 0.5`



5-animation



Testen

- Geschwindigkeit ändern
- andere Form animieren
- andere Zahl animieren
- Extreme ändern (min, max)

 map function

```
map(mouseX, 0, width, 0, 100);
```

☞ map function

```
map(mouseX, 0, width, 0, 100);
```

take this



now recalculate the
values between this range

to this range

Thank you :-)



IG @nahuel.gerth
GitHub/nahuelGerthVK

info@nahuelgerth.de
www.nahuelgerth.de