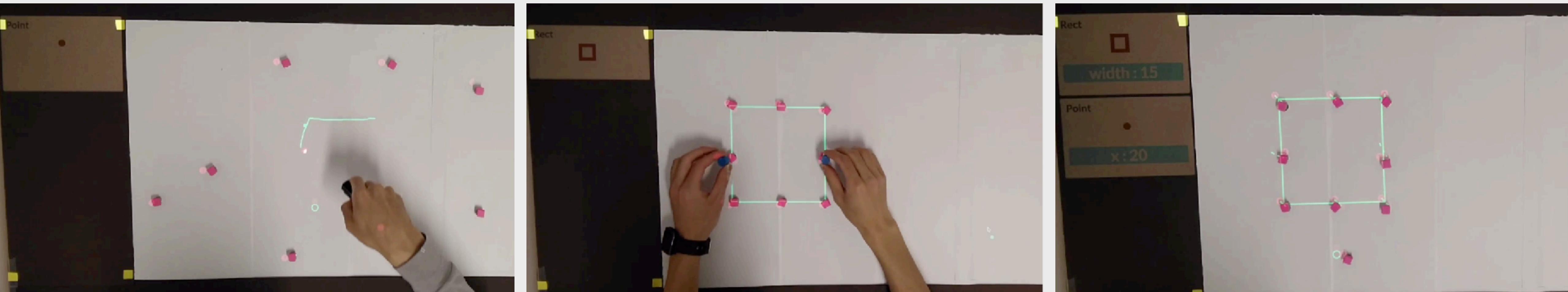
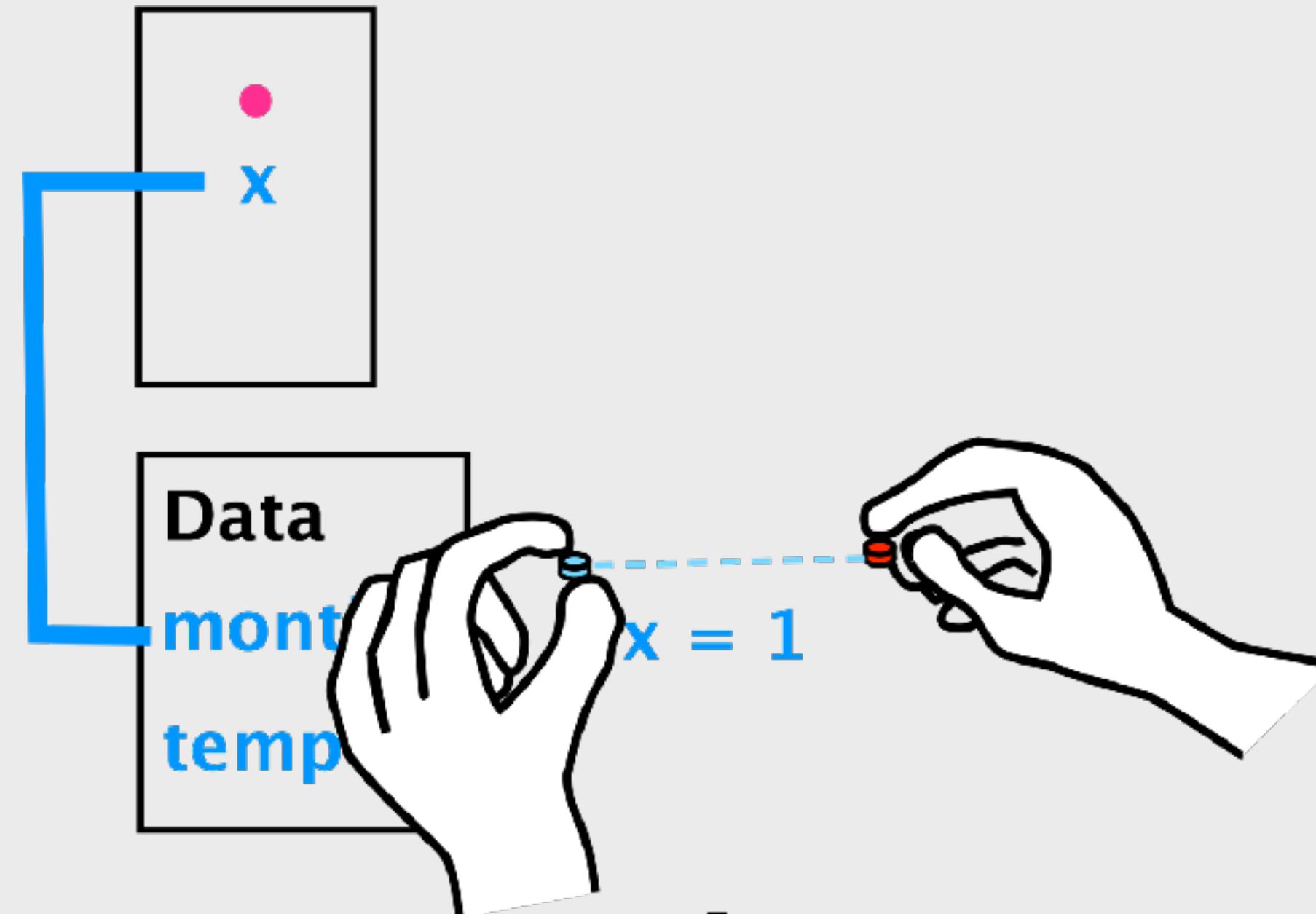


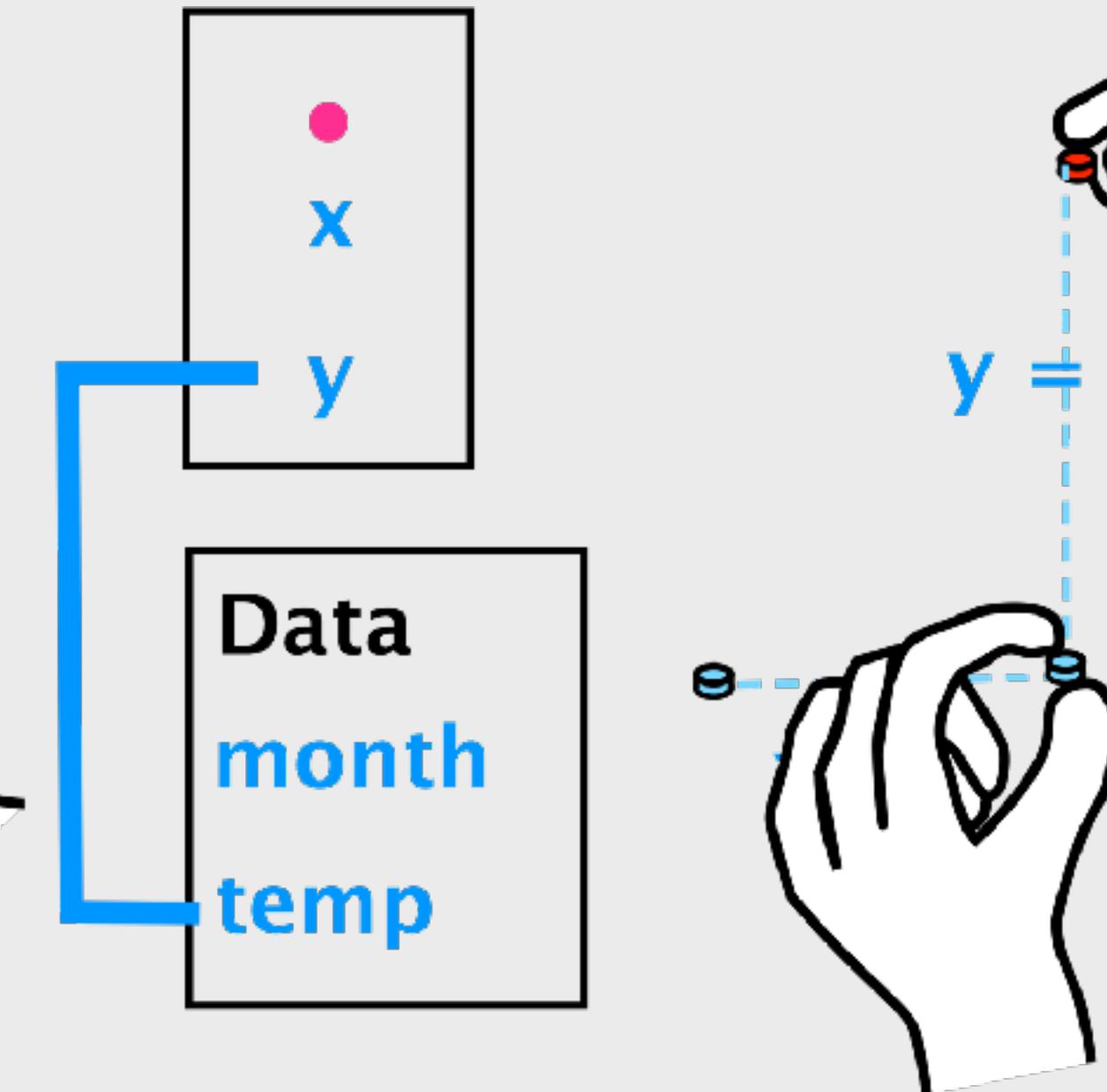
Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation



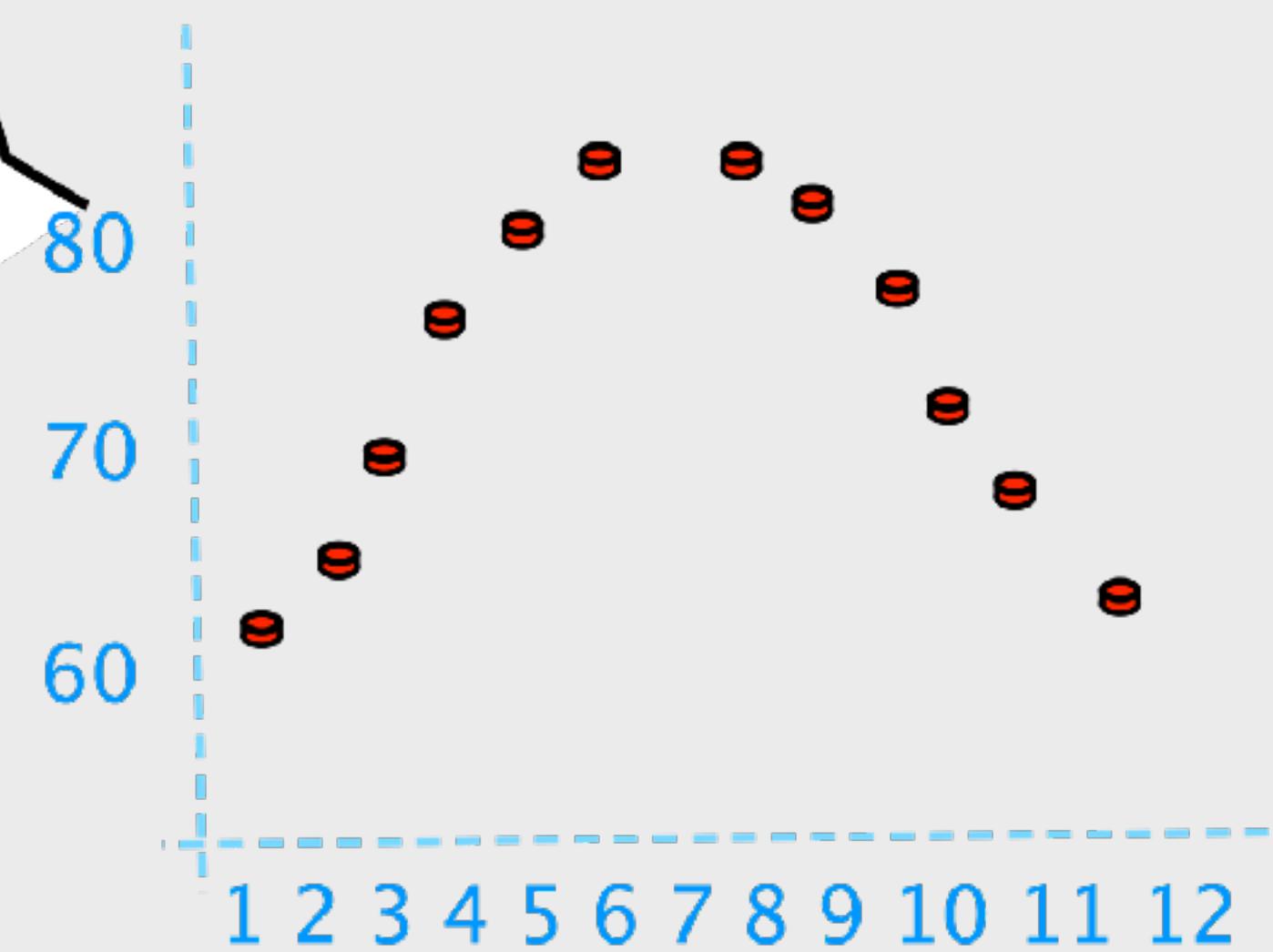
Ryo Suzuki, Jun Kato, Mark Gross, Tom Yeh



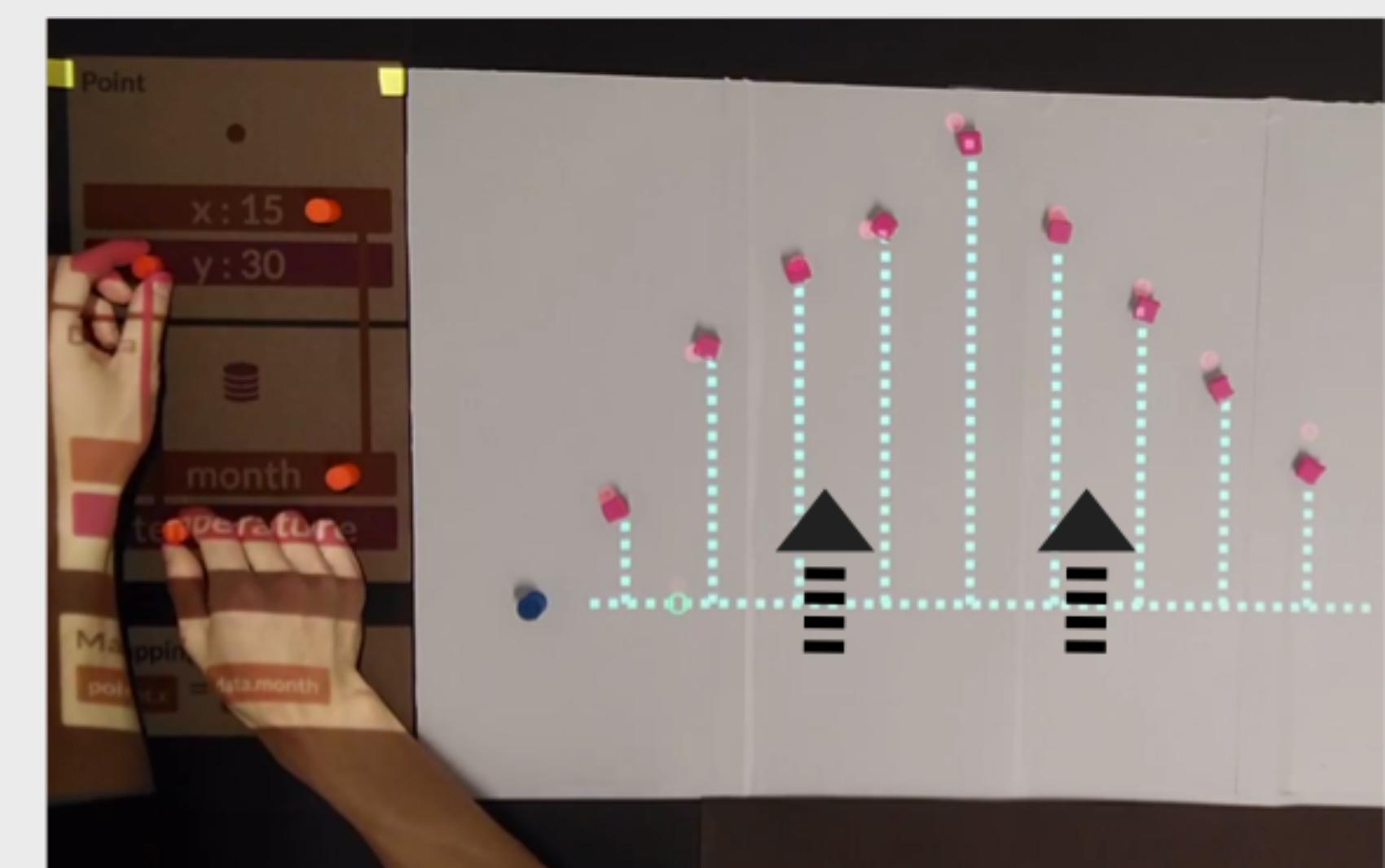
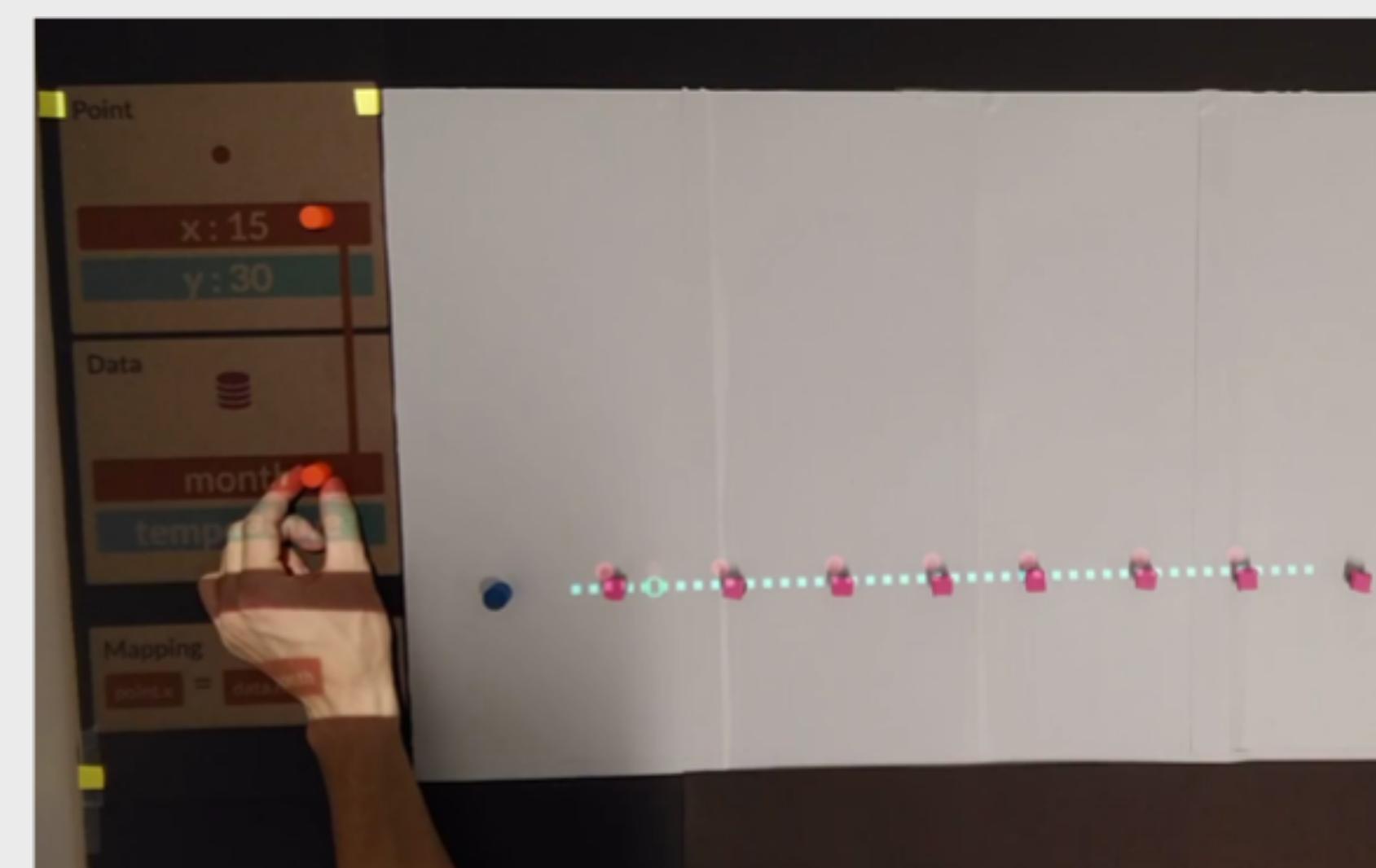
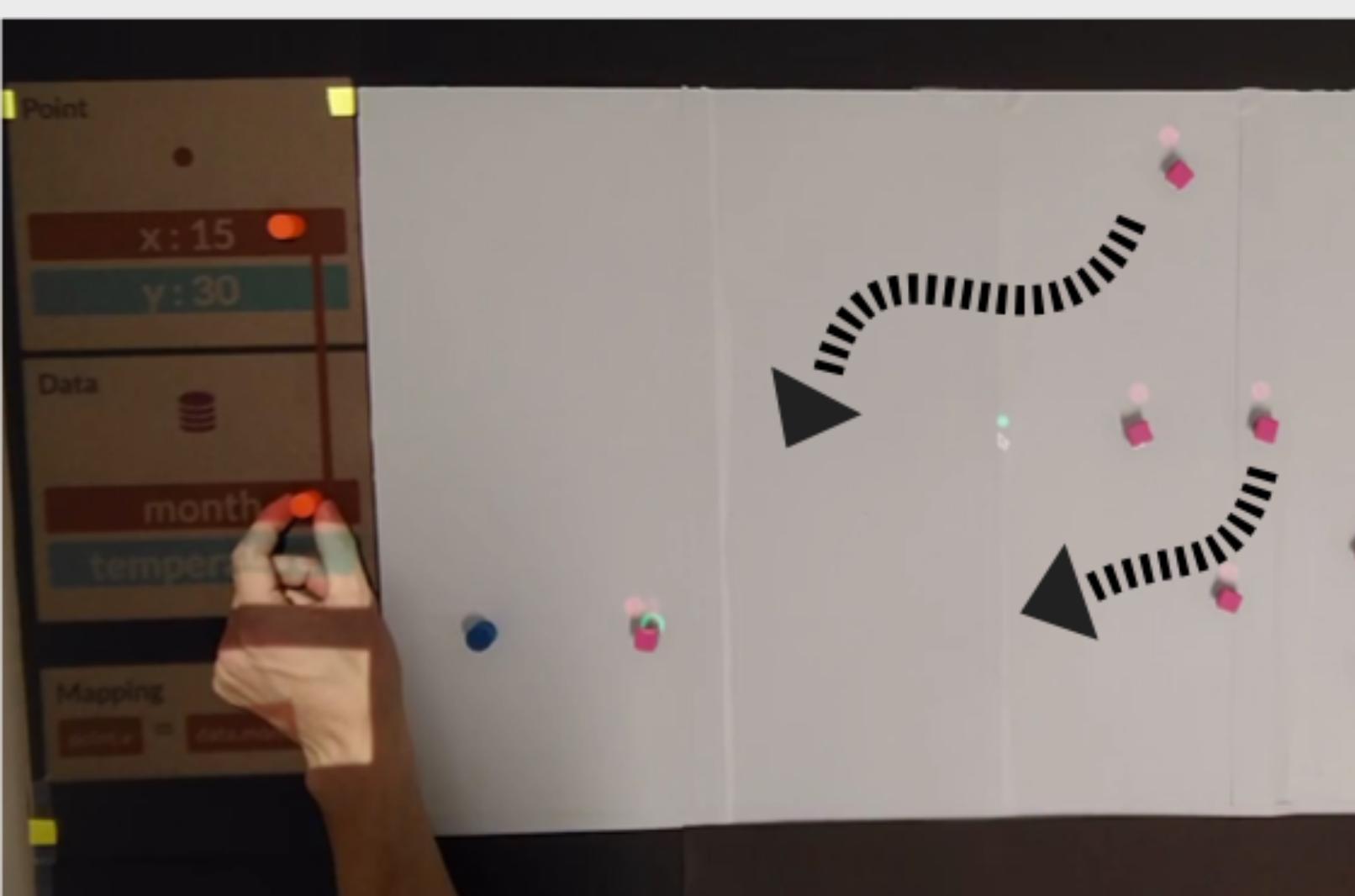
A

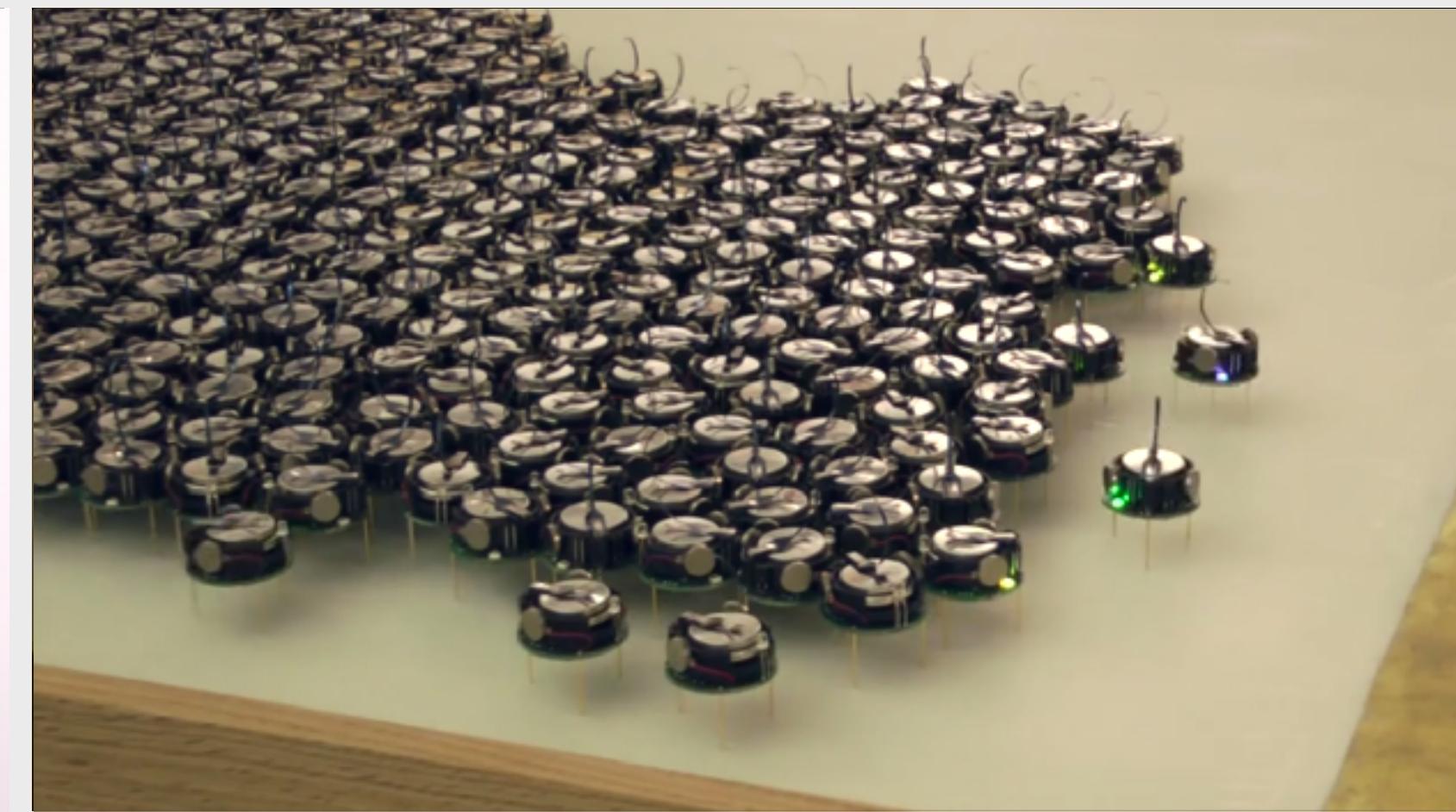


B

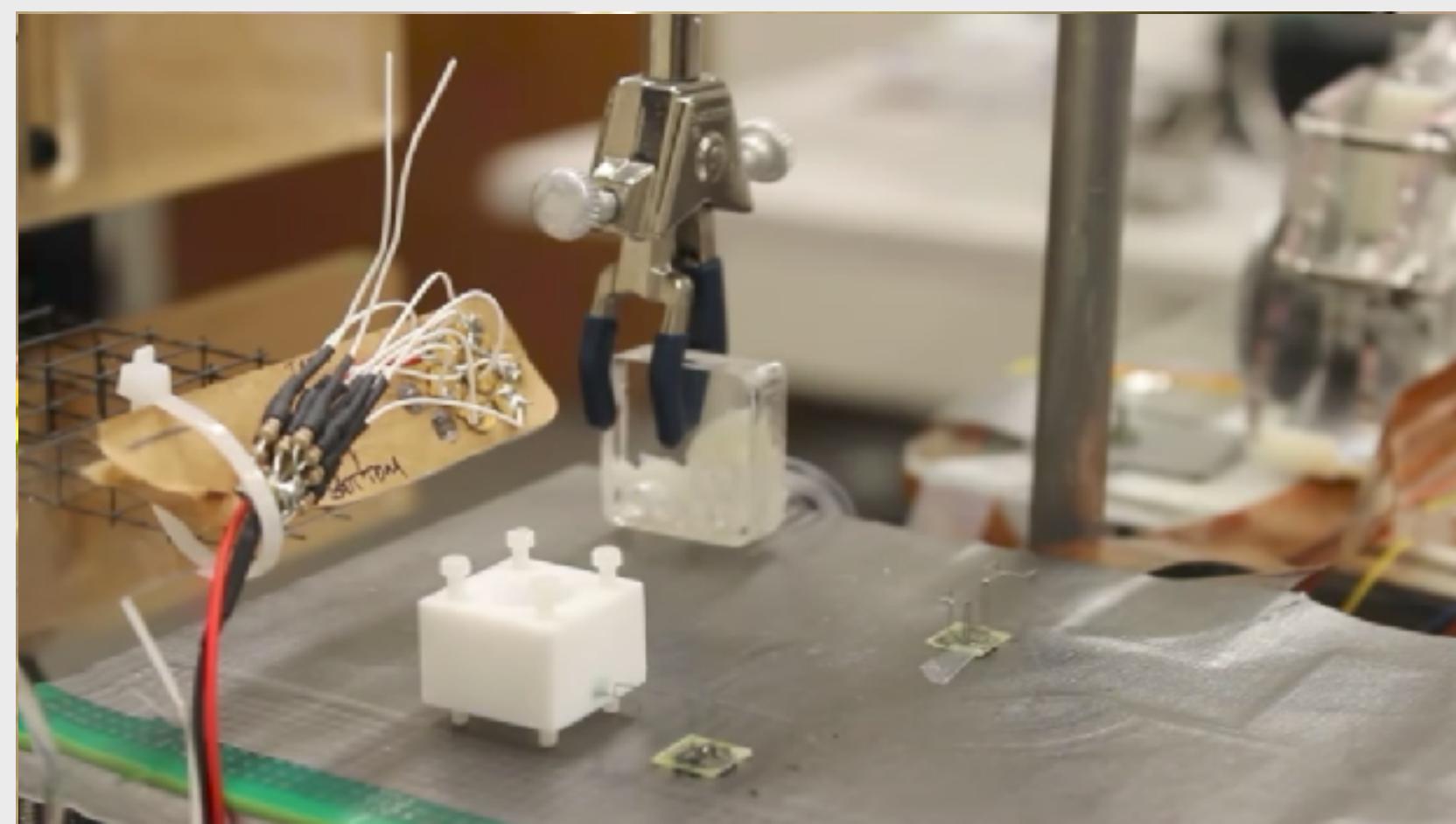
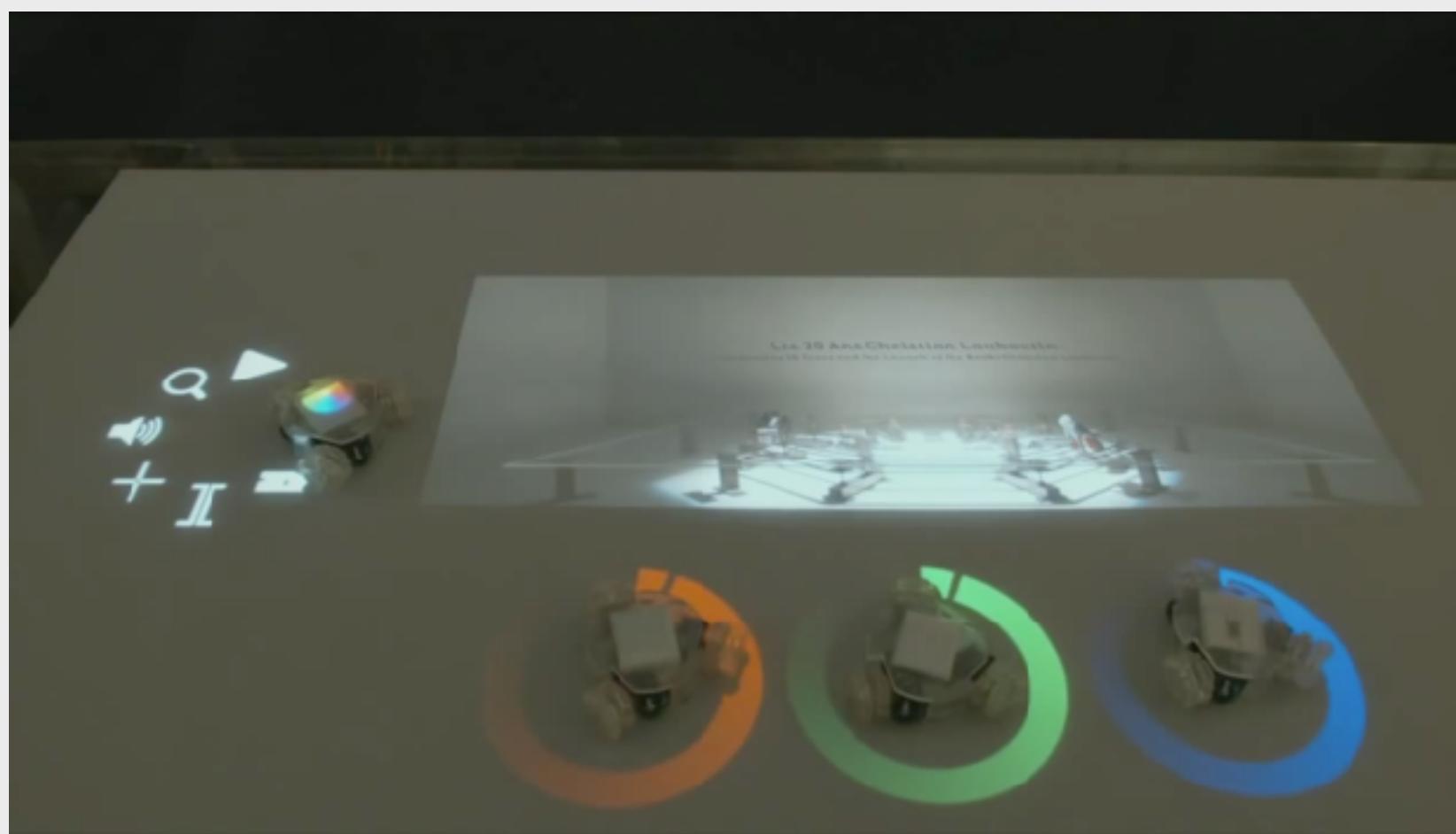


C





Swarm User Interfaces



**Application
Paradigm**

**Programming
Paradigm**

Robot

Swarm Robots

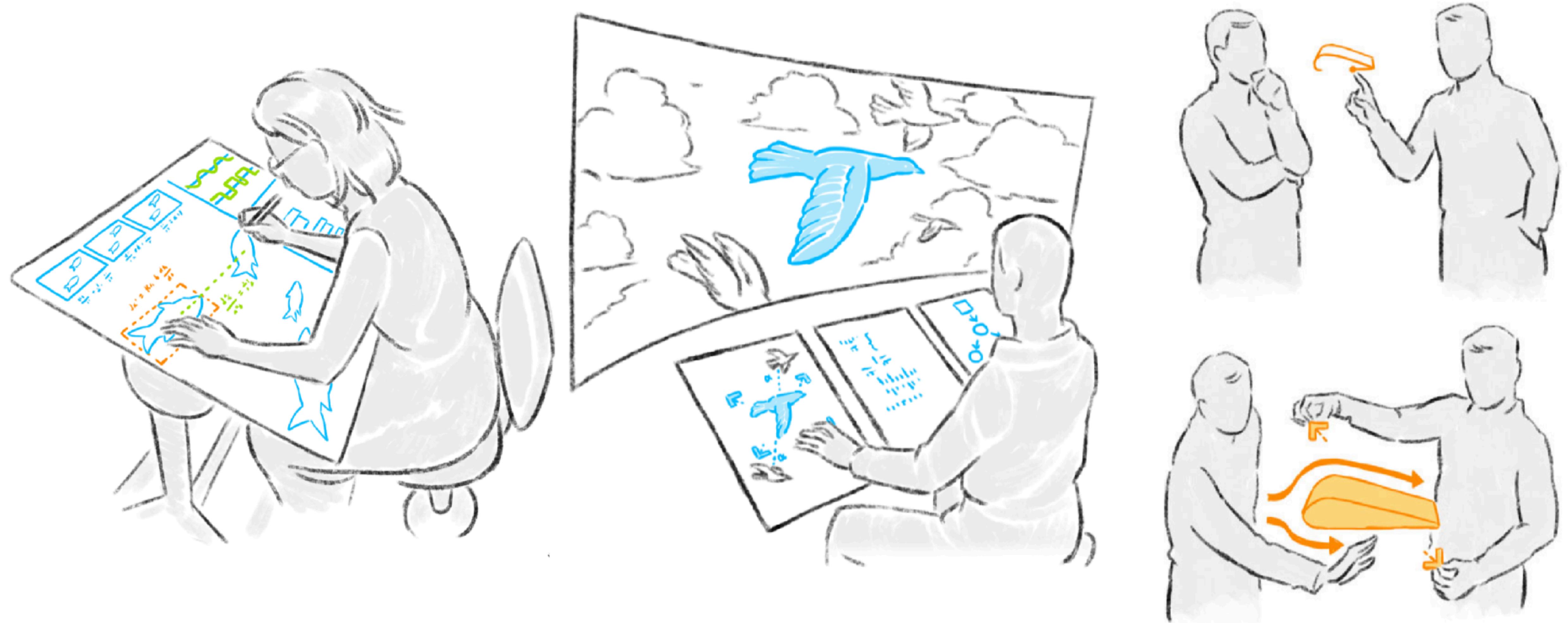
**Robot
Programming**

Dynamic Media

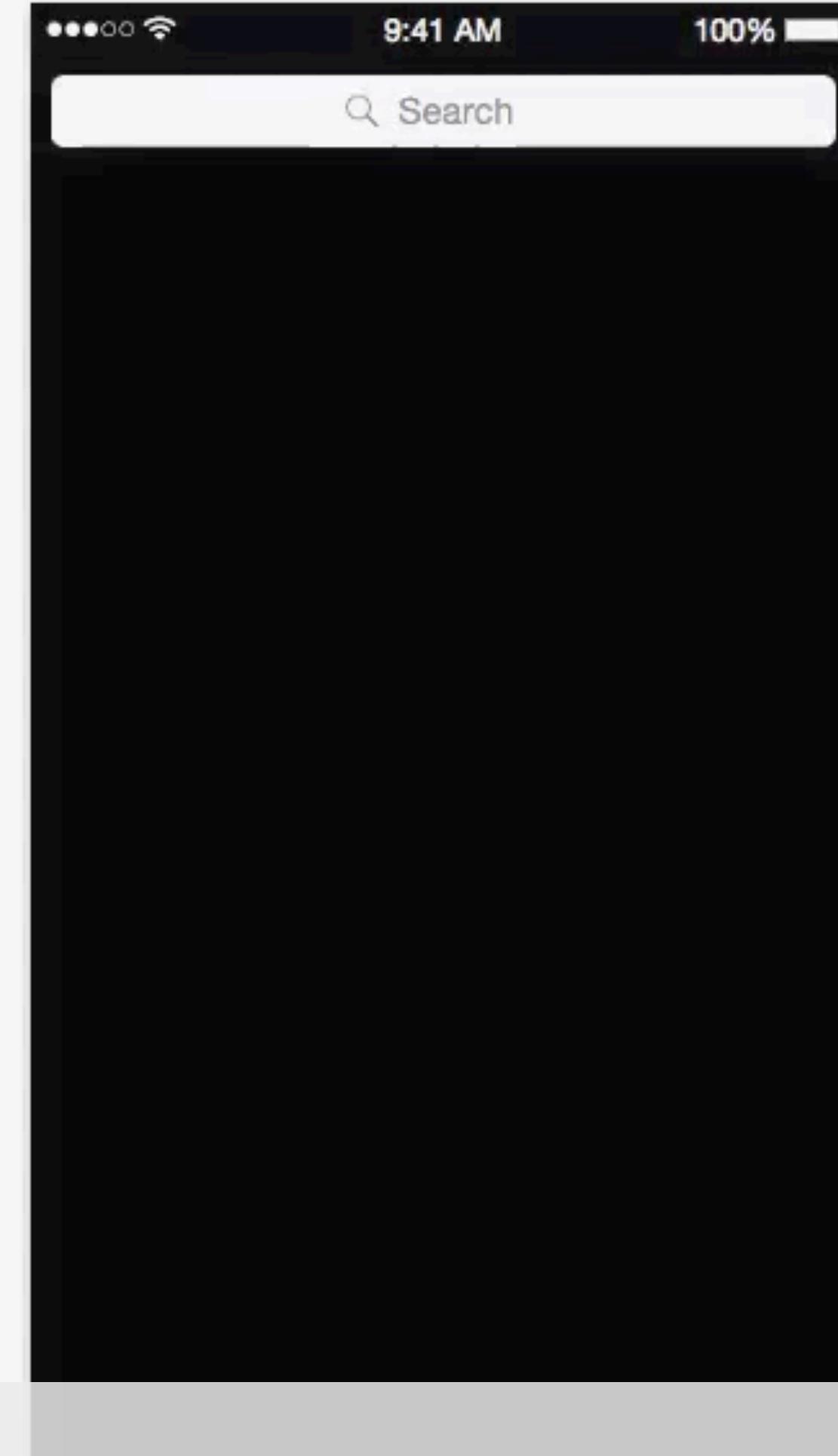
Swarm UI



Programming = Authoring Dynamic Materials



Sketches from “Human Representation of Thoughts” by Bret Victor [UIST 2014 Keynote]



The screenshot shows a mobile application interface. At the top, there is a navigation bar with a search bar containing the text "Search". Below the navigation bar is a large, dark rectangular area. In the center of this area, there is a card with the title "Playlist Name" and a "Songs >" button. A small circular icon with a gear symbol is located in the top right corner of the card.

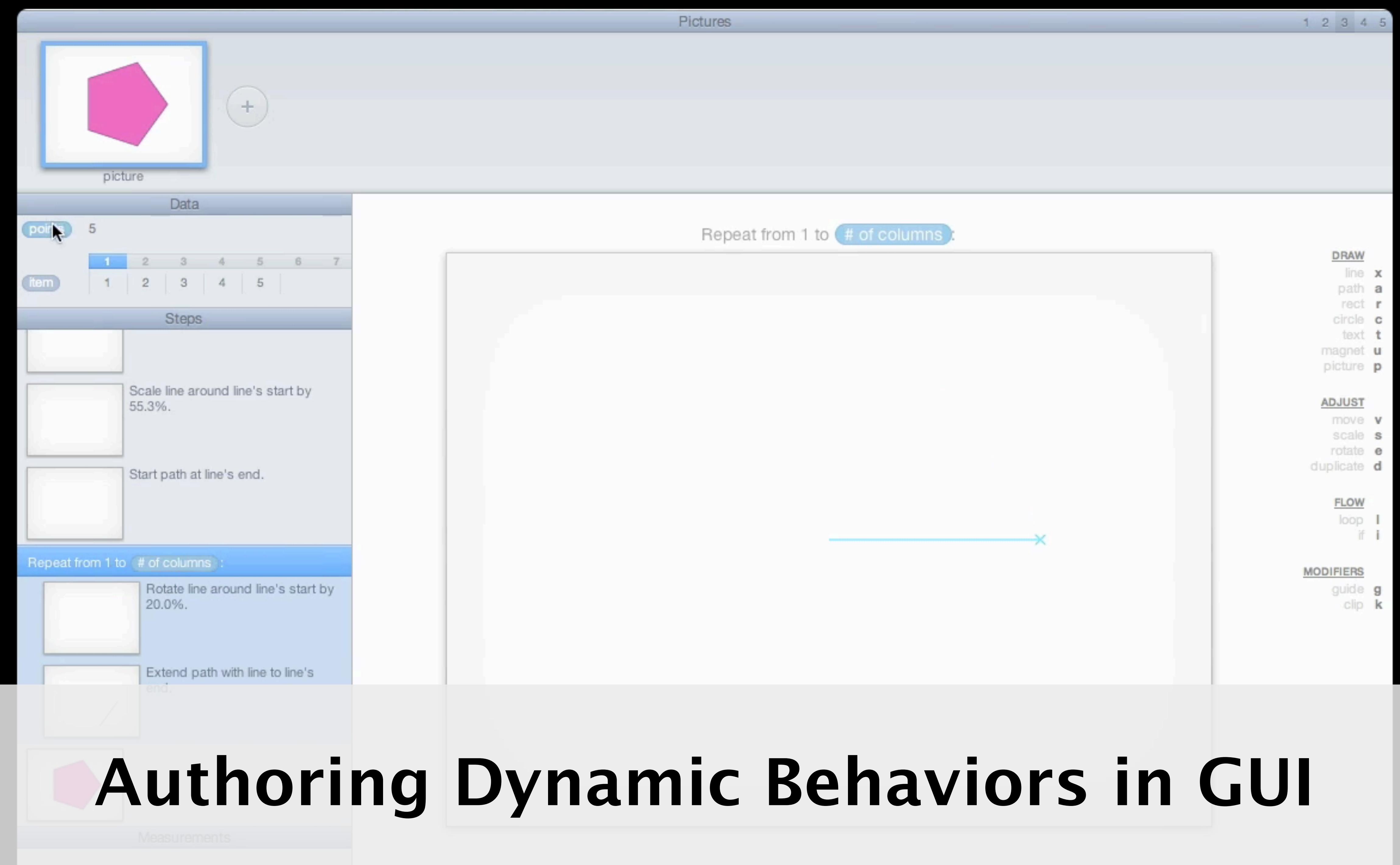
Data Source

JSON <https://api.spotify.com>

items (10) 0/10 < >

- 0
 - collaborative : false
 - external_urls
 - spotify : "http://open.spotify.com/user/fit..."
 - href : "https://api.spotify.com/v1/users/fitness..."
 - id : "OygJXhIK0XktOPBEIrWOAc"
 - images
 - height : 640
 - url : "https://mosaic.scdn.co/640/97..."
 - width : 640
 - 1
 - height : 300
 - url : "https://mosaic.scdn.co/300/97..."
 - width : 300
 - 2
 - height : 60
 - url : "https://mosaic.scdn.co/60/97..."
 - width : 60
 - name : "Running"
 - owner
 - external_urls
 - spotify : "http://open.spotify.com/u..."
 - href : "https://api.spotify.com/v1/users/ft..."
 - id : "fitnessguru.com"
 - type : "user"
 - uri : "spotify:user:fitnessguru.com"
 - public : null
 - snapshot_id : "nVhVsPJZDIHGGi8OuYRrvEFgX..."
 - tracks
 - href : "https://api.spotify.com/v1/users/f..."
 - total : 25
 - type : "playlist"
 - uri : "spotify:user:fitnessguru:playlist:0..."
- 1
 - 2

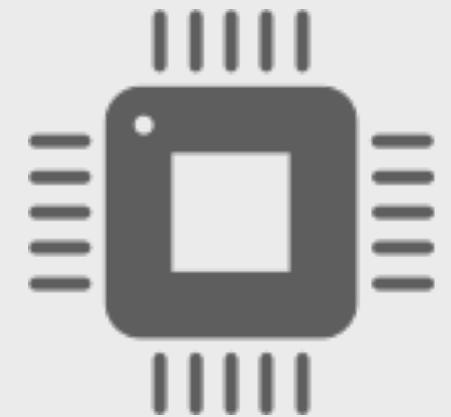
Authoring Dynamic Behaviors in GUI



Robot Programming

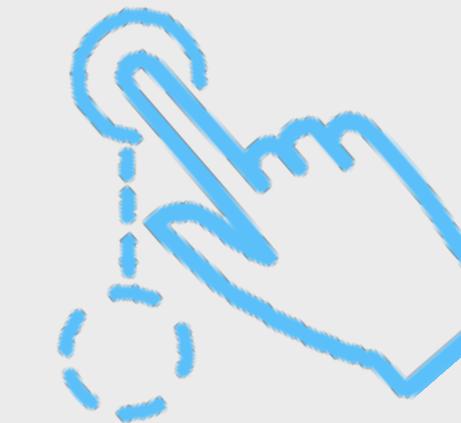


computer screen



low-level controls

Swarm UI Programming

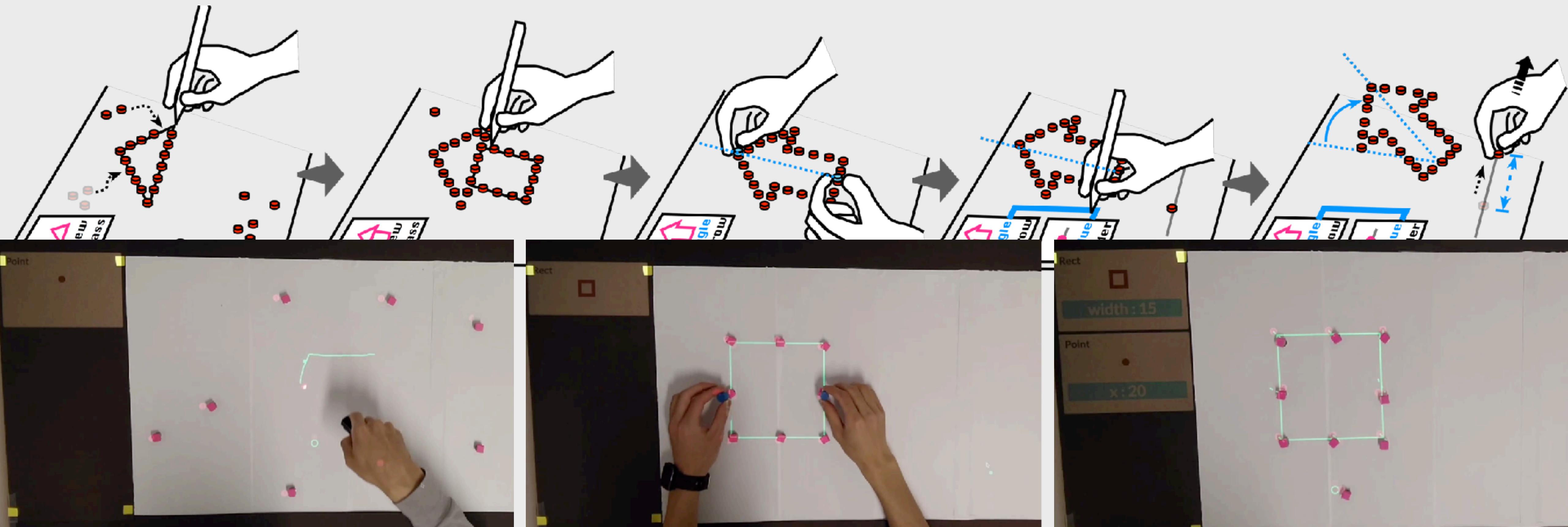


direct physical manipulation



high-level interface design

Swarm UI Programming



1. Summary

2. Related Work

3. Design and Workflow

4. Implementation

5. Limitations and Future Work

Direct Manipulation in Physical Space

Interacting with the **Pre-programmed** Behaviors



DigitalDesk
[Wellner 1993]

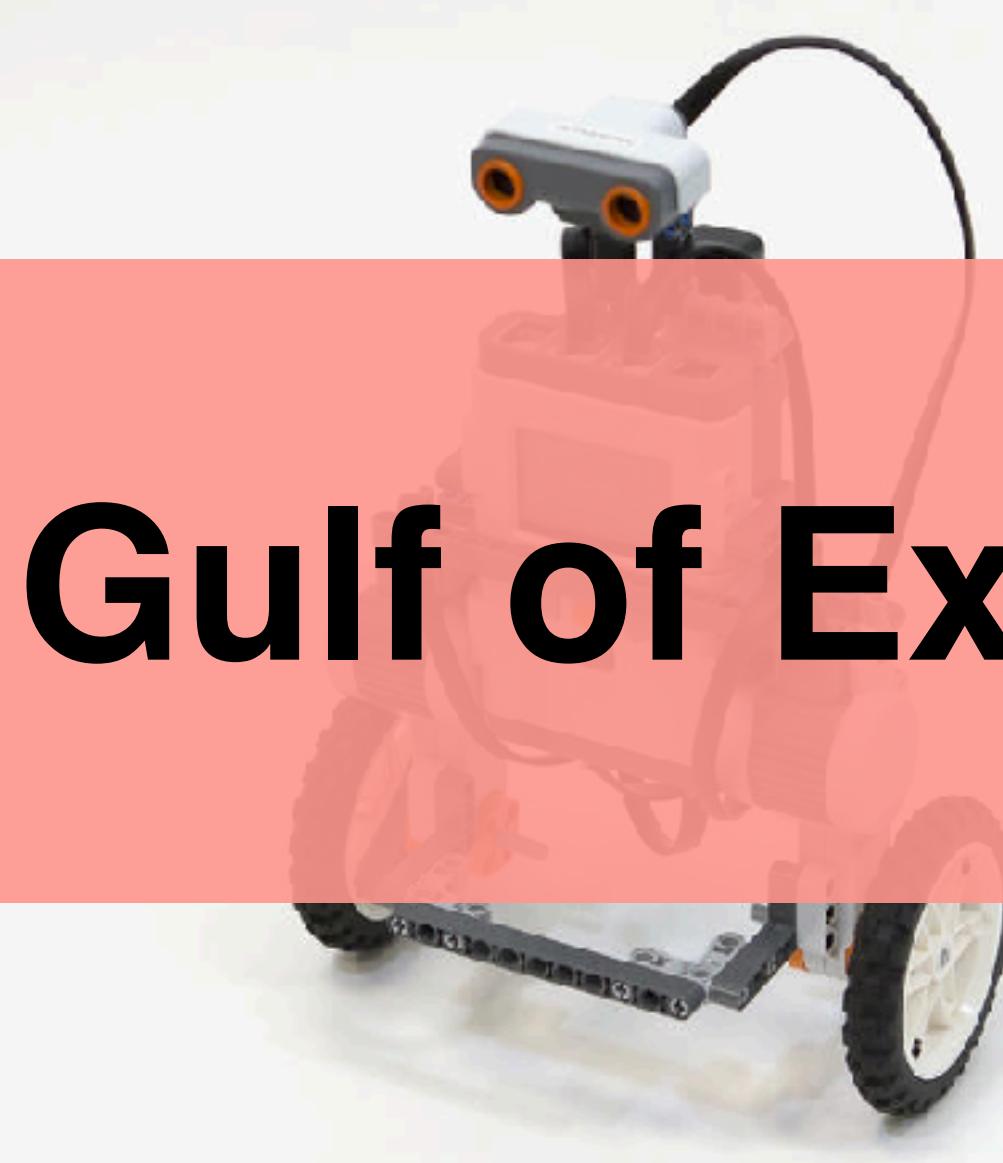


I/O Bulb
[Underkoffler and Ishii 1998]

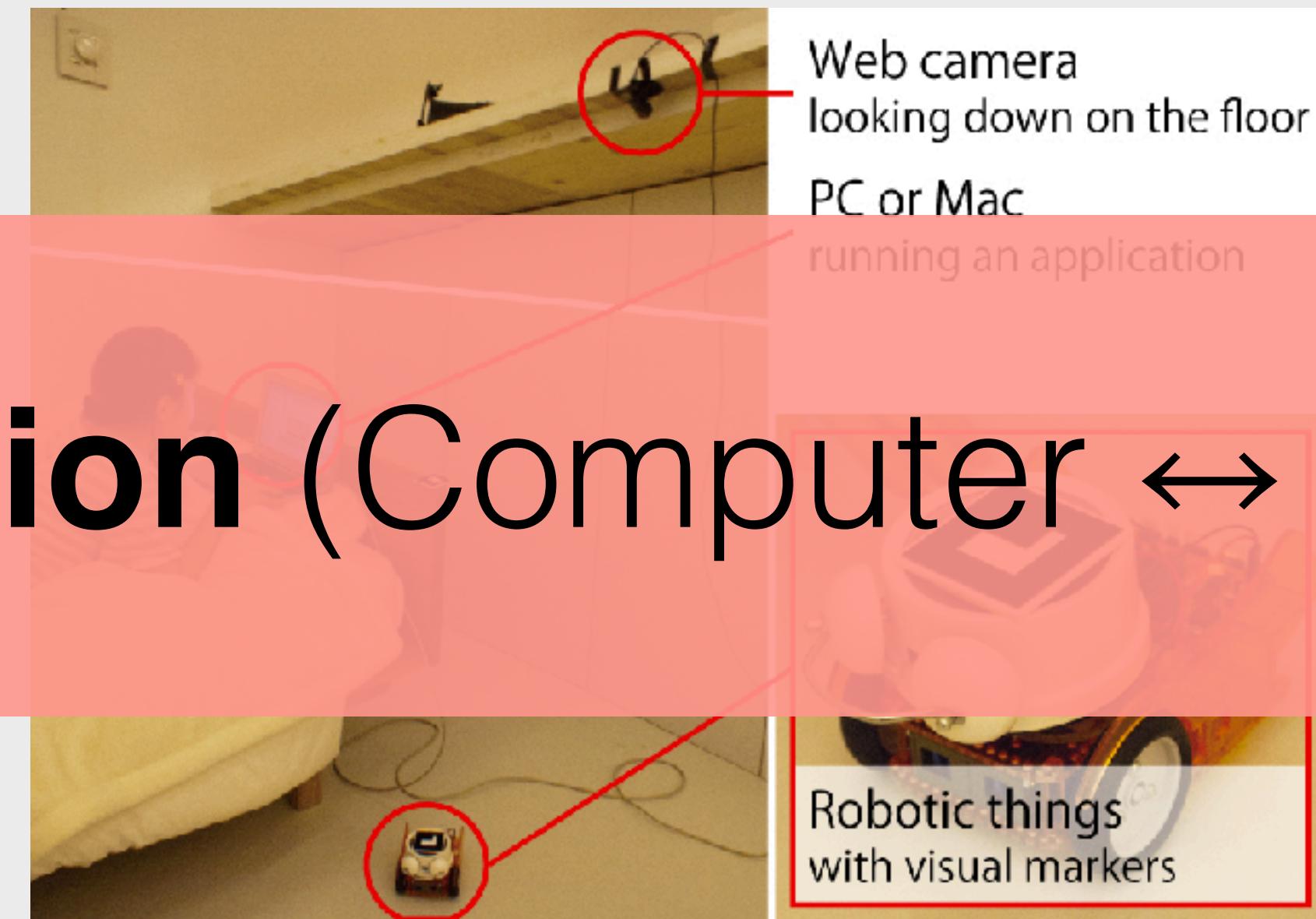


Mockup Builder
[Bruno et al. 2012]

Toolkits for Robot Programming



Gulf of Execution (Computer ↔ Physical World)



LEGO Mindstorms

Phybots
[Kato et al. 2012]

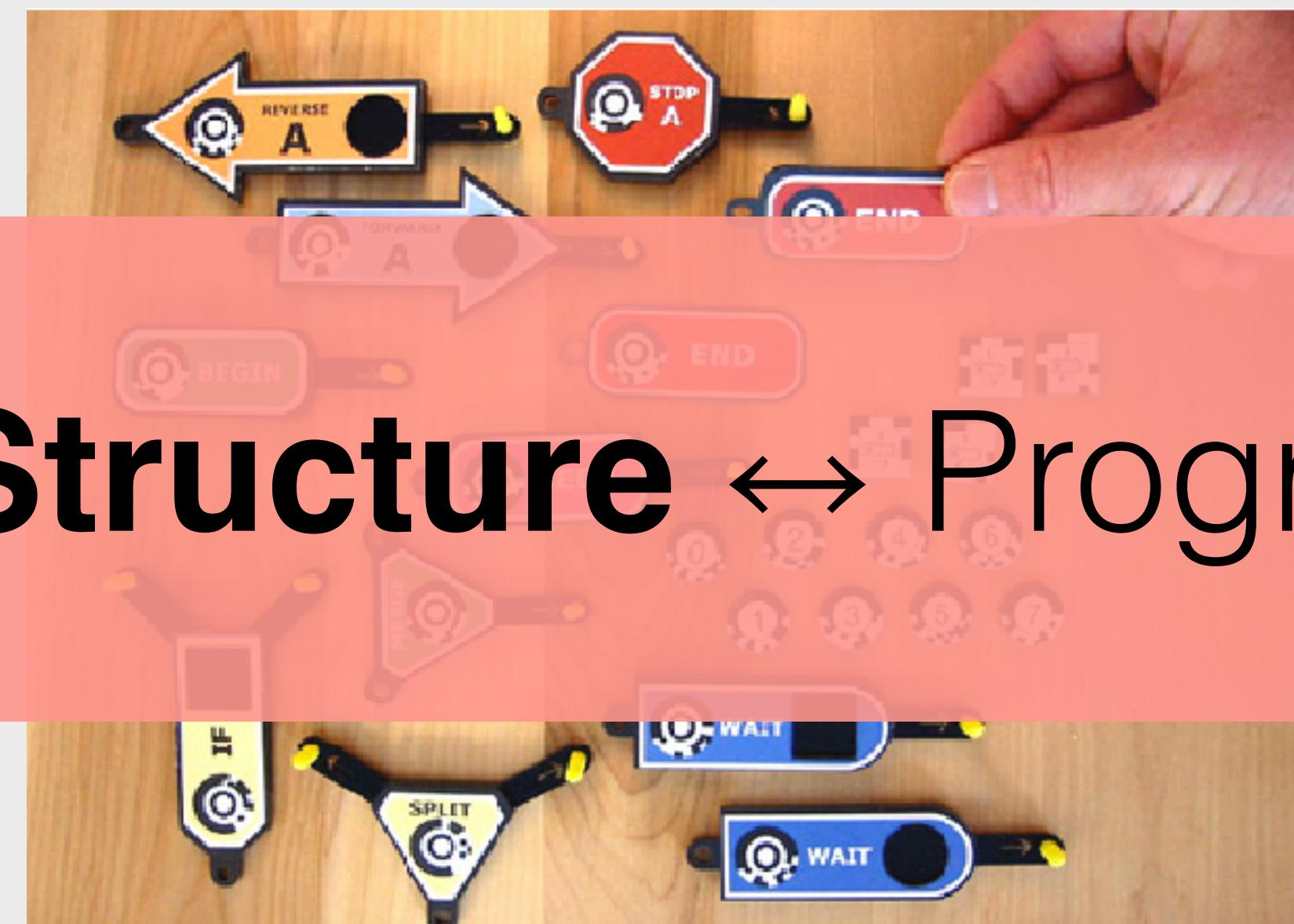
GUI Robots
[Guiness et al. 2017]

Tangible Programming Languages

Program **Structure** \leftrightarrow Program **States**



AlgoBlock
[Suzuki and Kato 1995]



Quetzal
[Horn and Jacob 2007]



Osmo (Strawbies)
[Hu et al. 2015]

Programming by Demonstration



Simple Repetitive Operations for a Single Robot

Topobo
[Raffle et al. 2004]

Robot Programming by Demonstration
[Billard et al. 2007]

1. Summary

2. Related Work

3. Design and Workflow

4. Implementation

5. Limitations and Future Work

Step 1

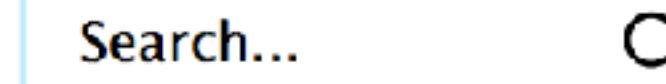
Create Elements

Button

Web UI



Search...



Button

Step 2

Abstract Attributes

Button

background-color

A A

font-size

A A

font-size

HTML DOM

HTML Attributes

Step 3

Specify Behaviors

A A

text.font-size



slider.value



Event Handling

A A

text.font-size

slider.value

Step 4

Propagate Changes

Text

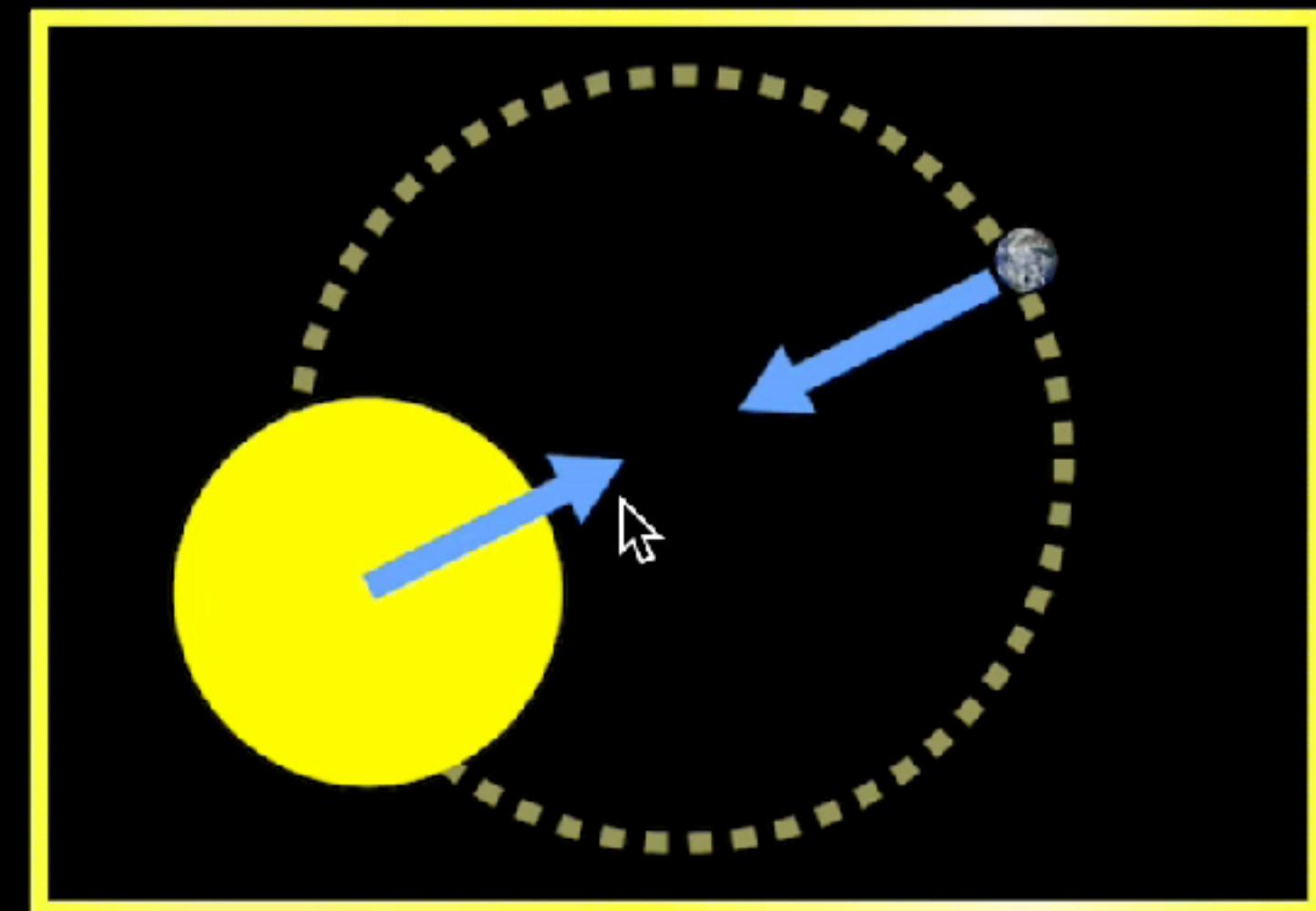


→

Text

Appearance Changes

Gravity And Orbits

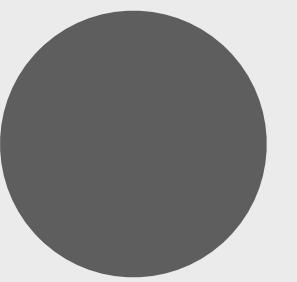


Model

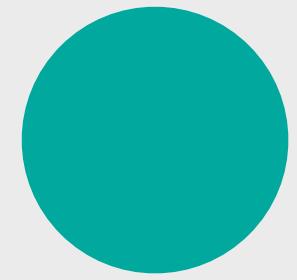


To Scale

```
<circle id="Sun"  
/>
```

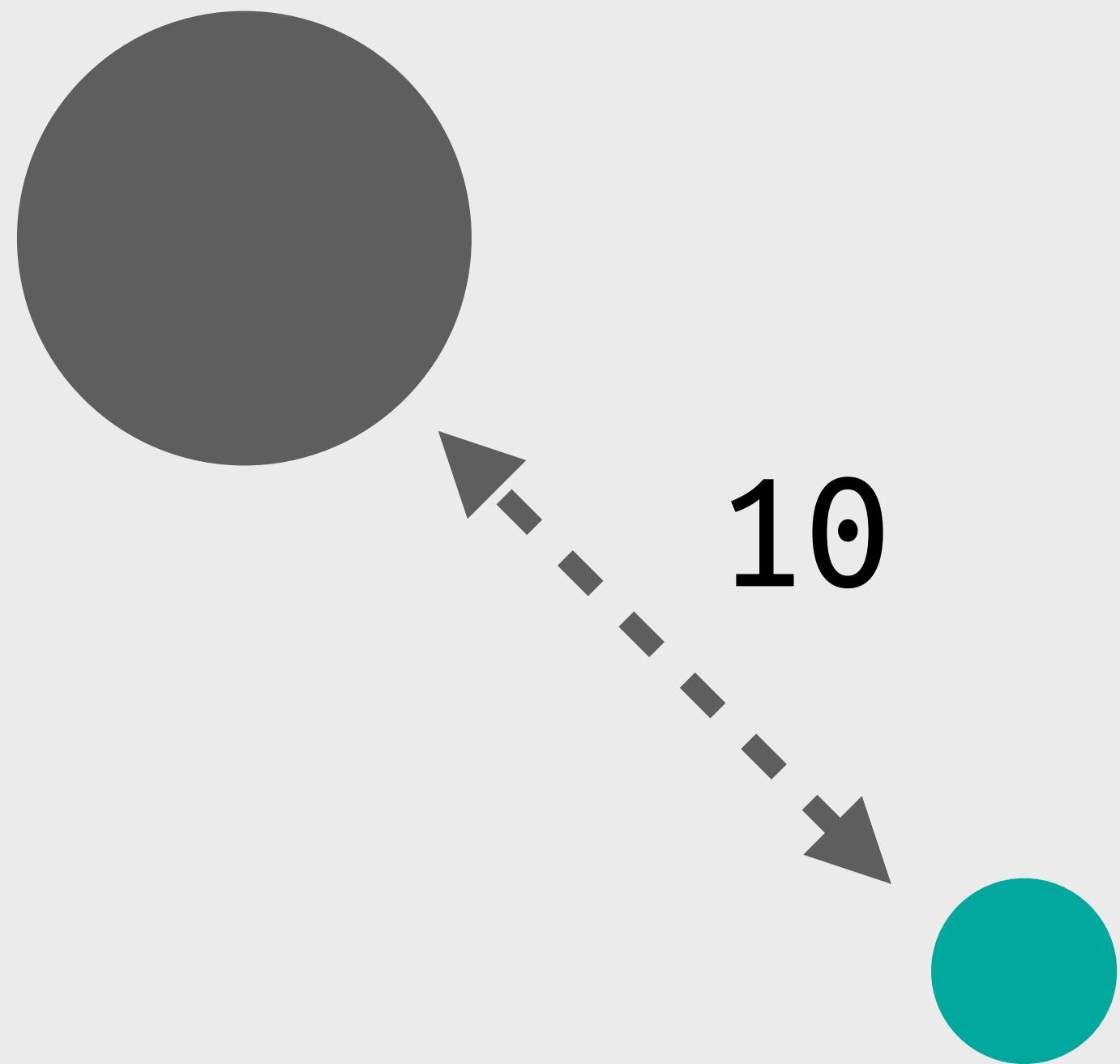


```
<circle id="Earth"  
/>
```



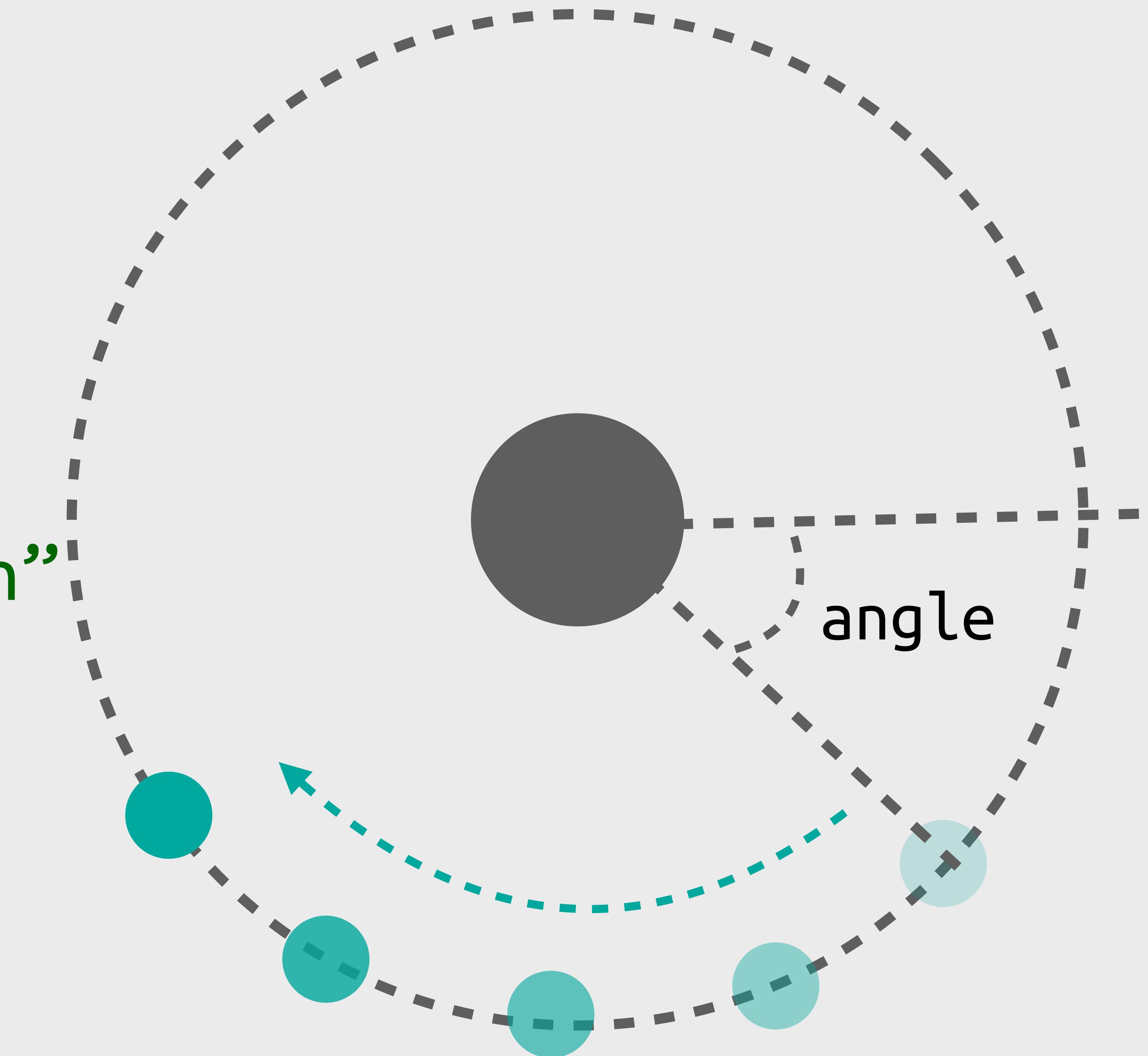
```
<circle id="Sun"  
       radius = 5  
/>
```

```
<circle id="Earth"  
       radius = 1  
       distance = 10  
/>
```



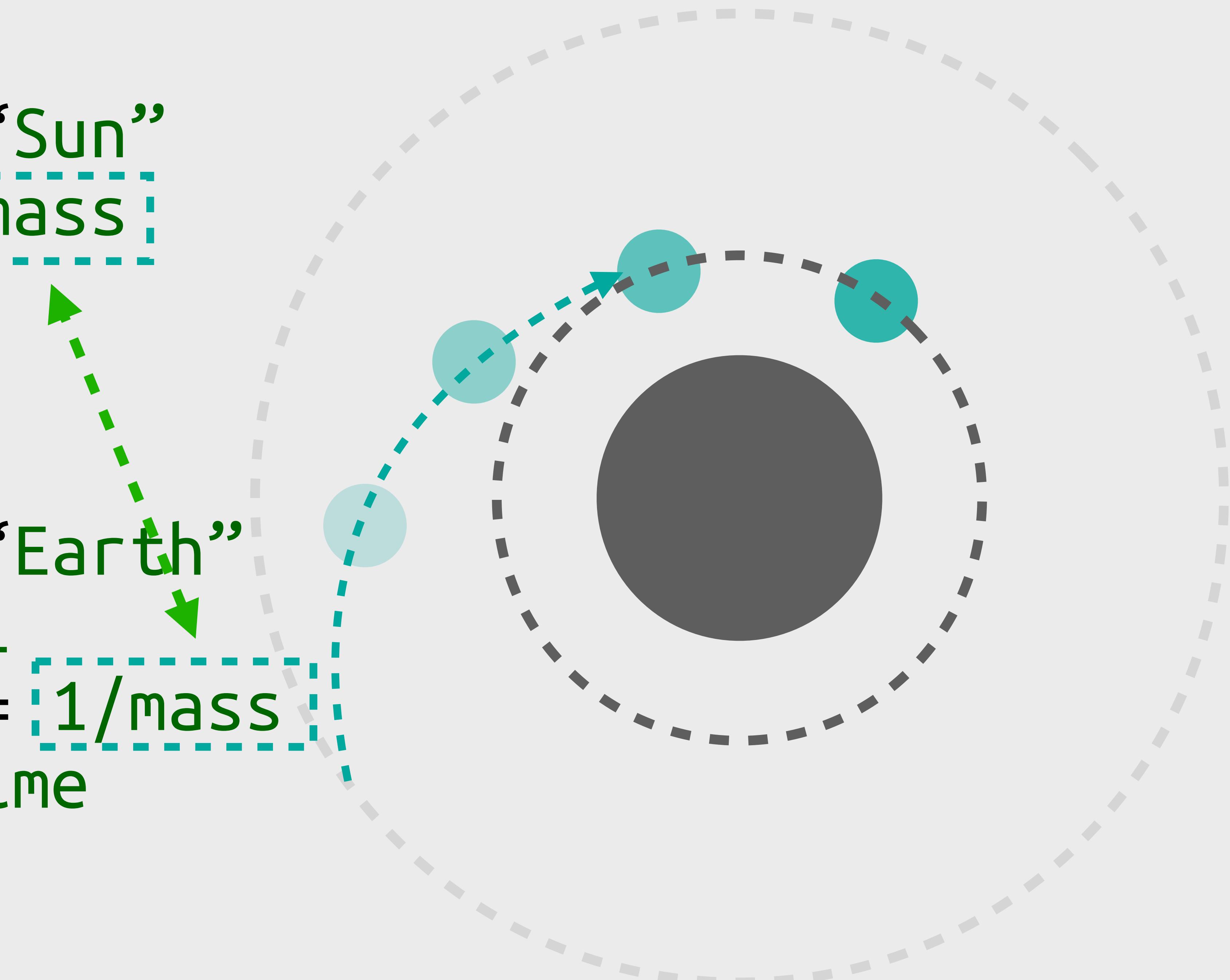
```
<circle id="Sun"  
radius = 5  
/>
```

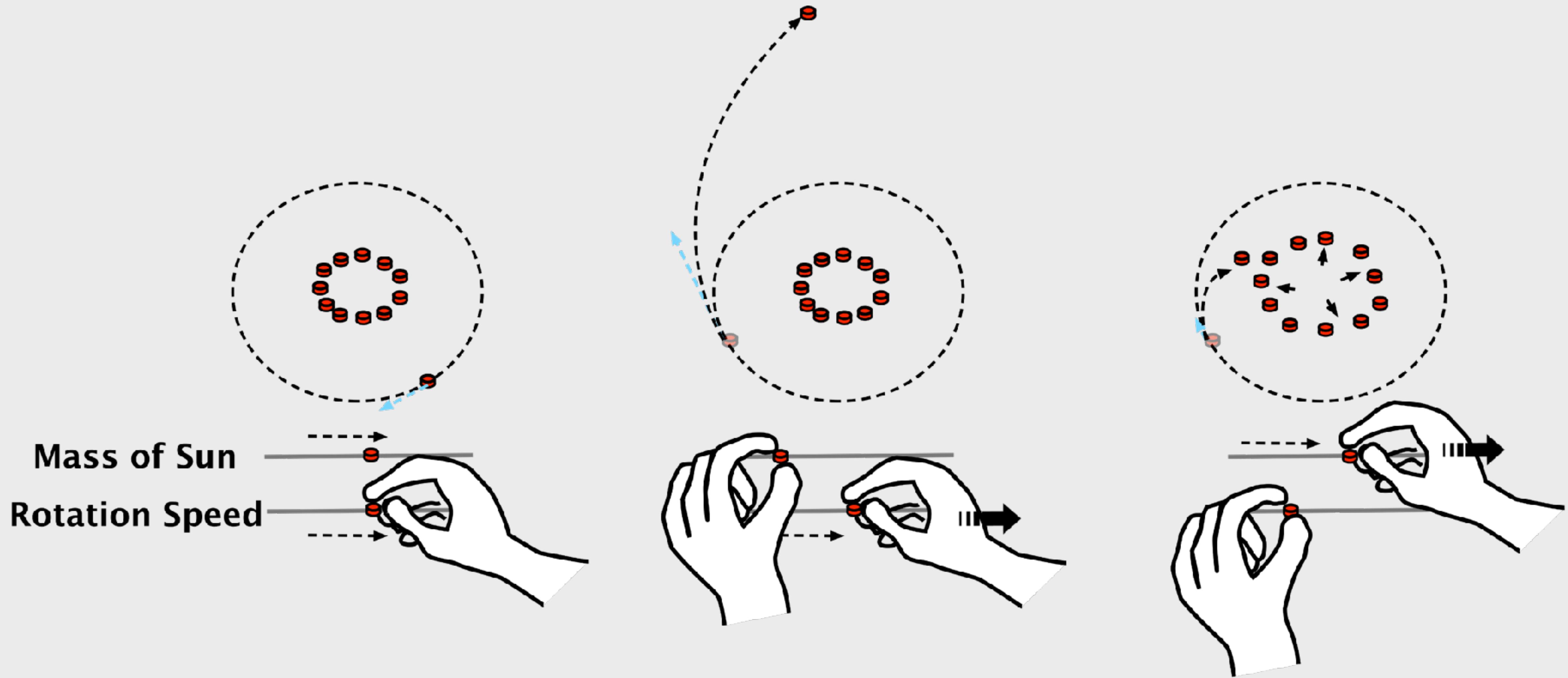
```
<circle id="Earth"  
radius = 1  
distance = 10  
angle = time  
/>
```

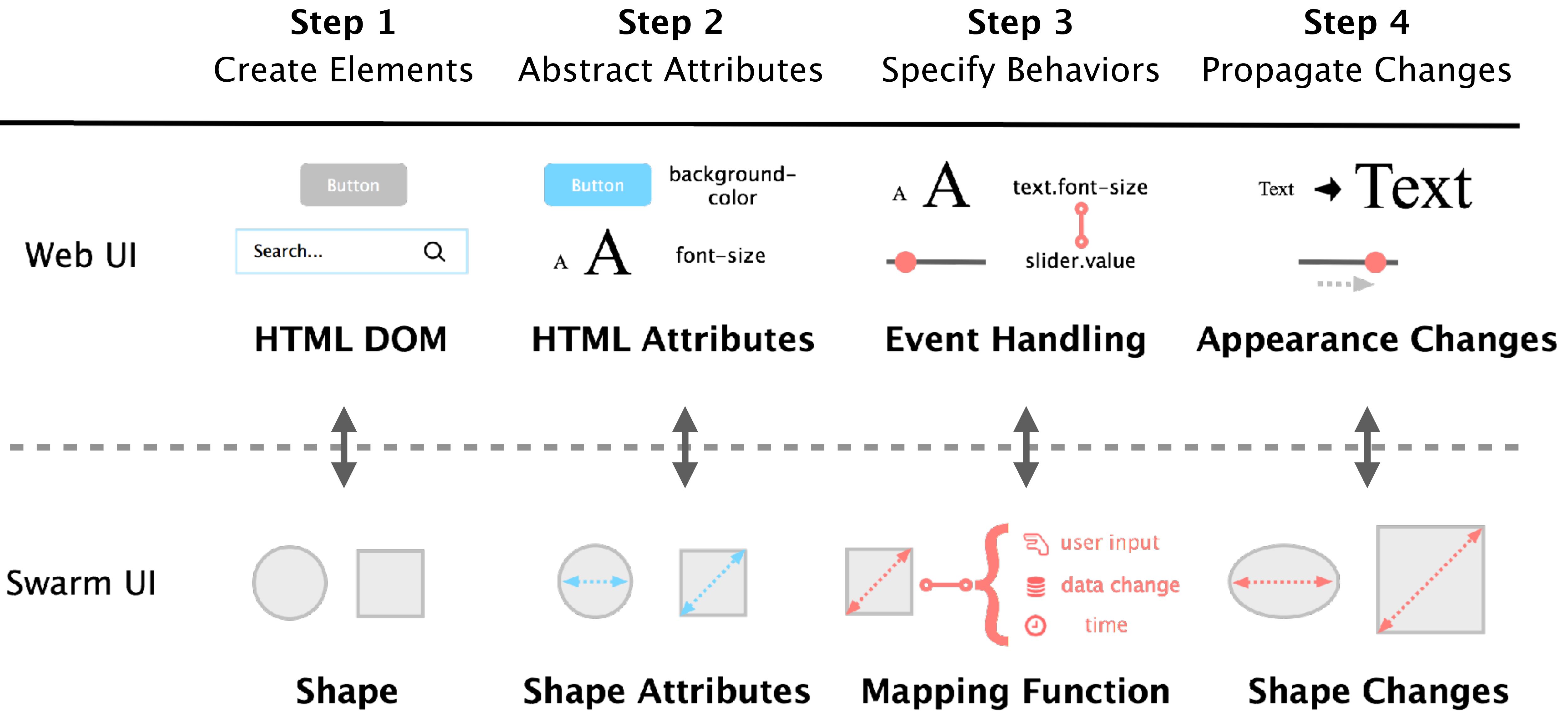


```
<circle id="Sun"  
radius = [mass]  
/>
```

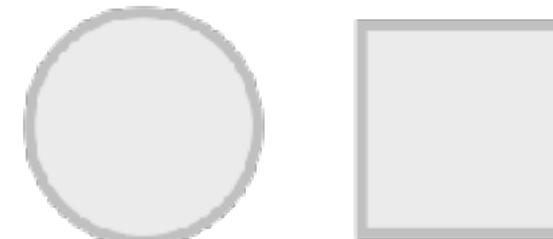
```
<circle id="Earth"  
radius = 1  
distance = [1/mass]  
angle = time  
/>
```







Swarm UI



Step 1

Create Elements

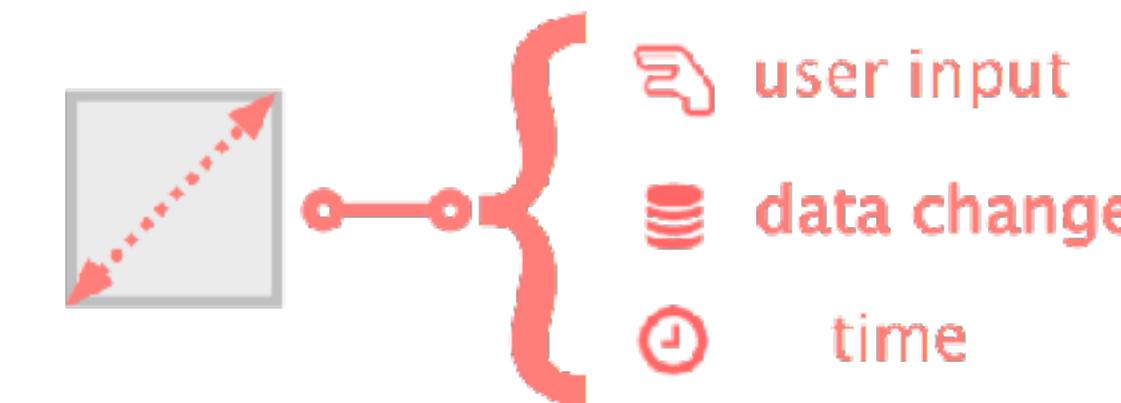
Step 2

Abstract Attributes



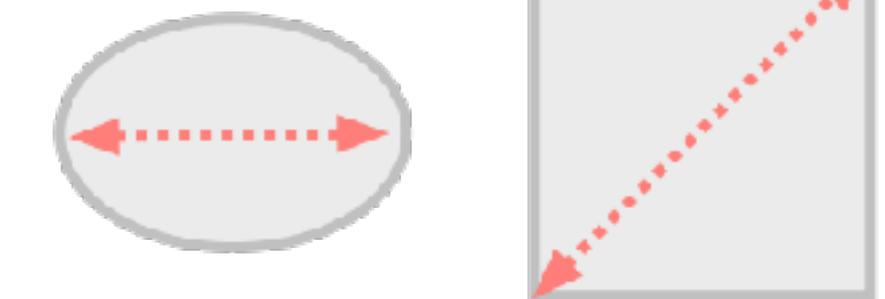
Step 3

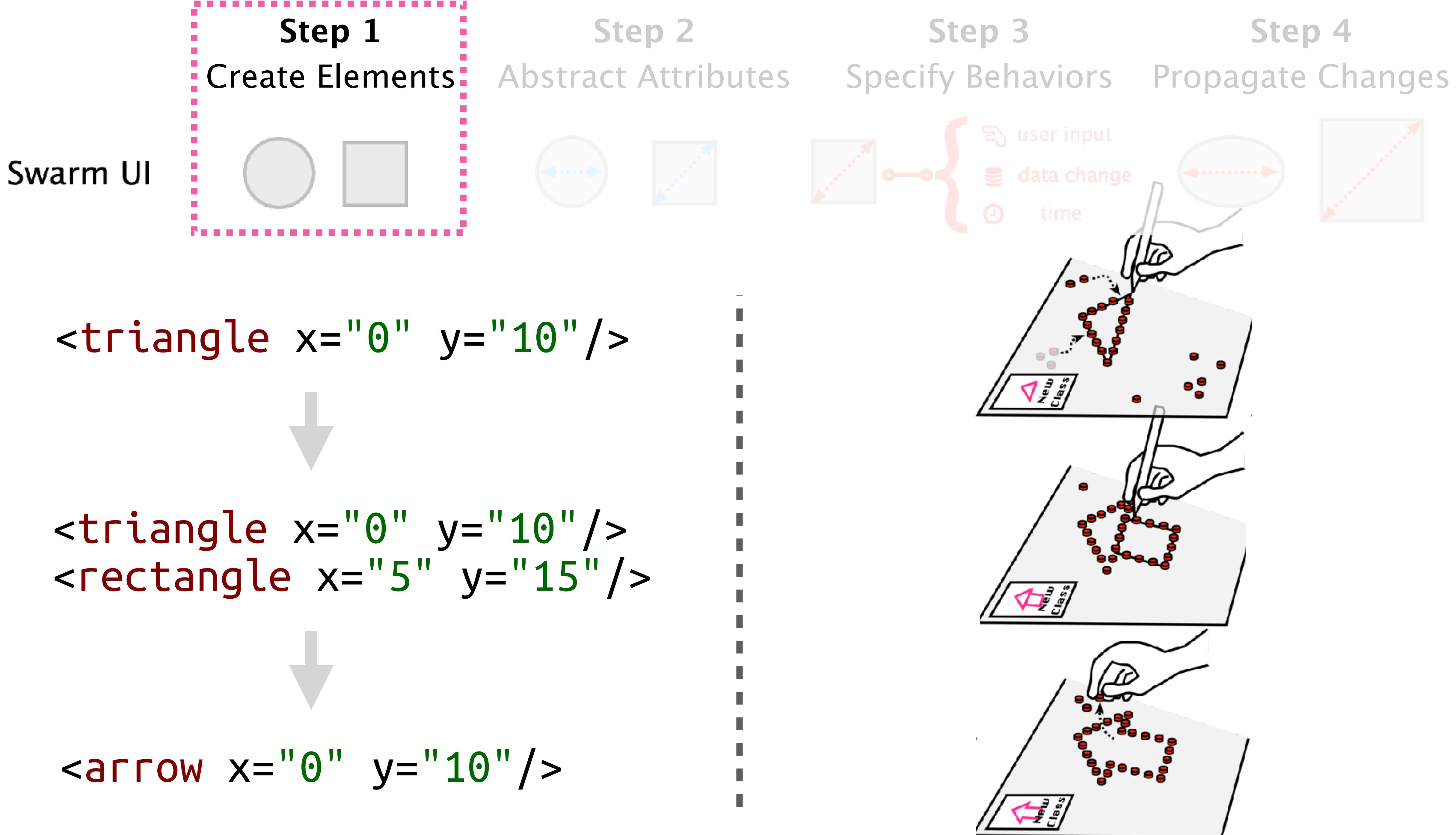
Specify Behaviors



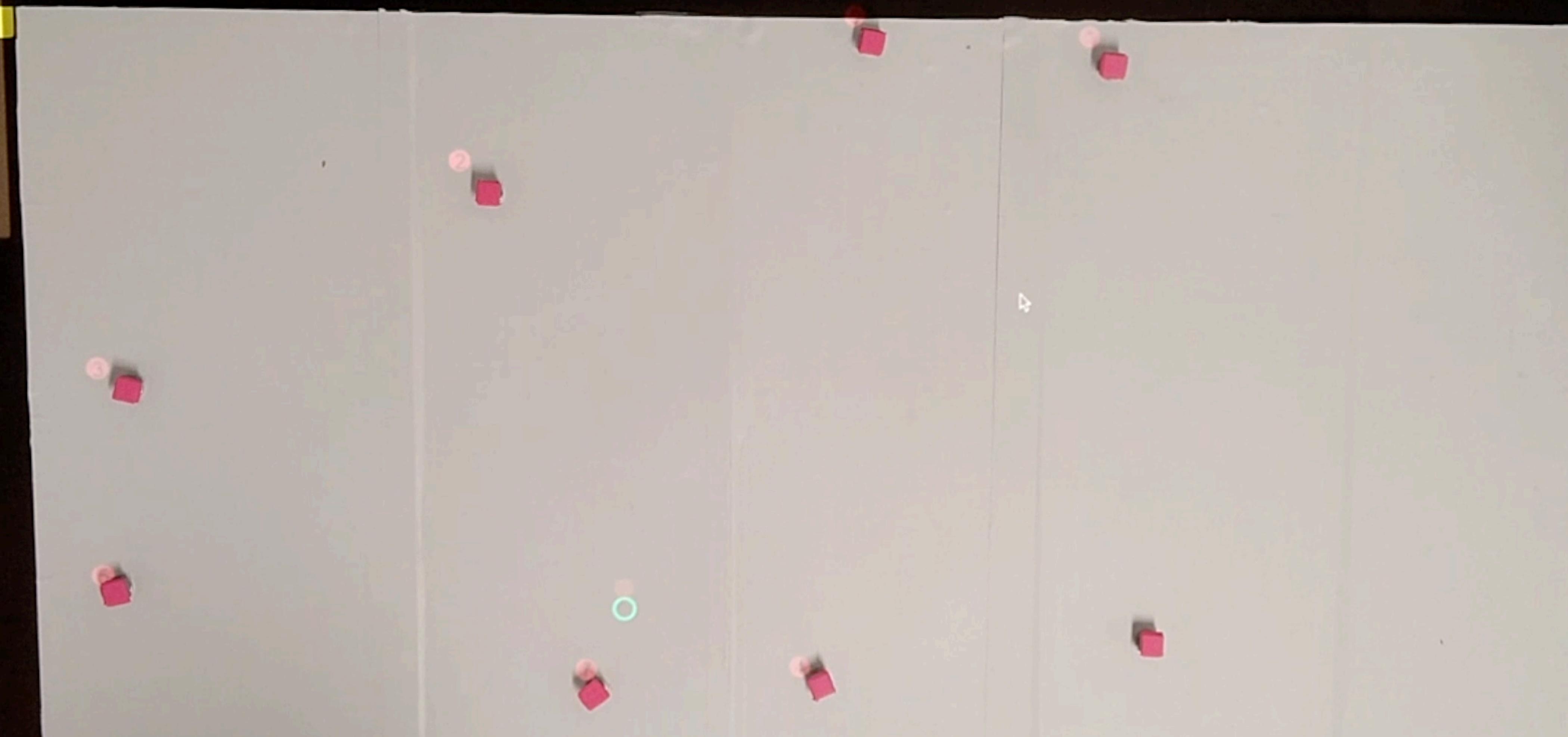
Step 4

Propagate Changes



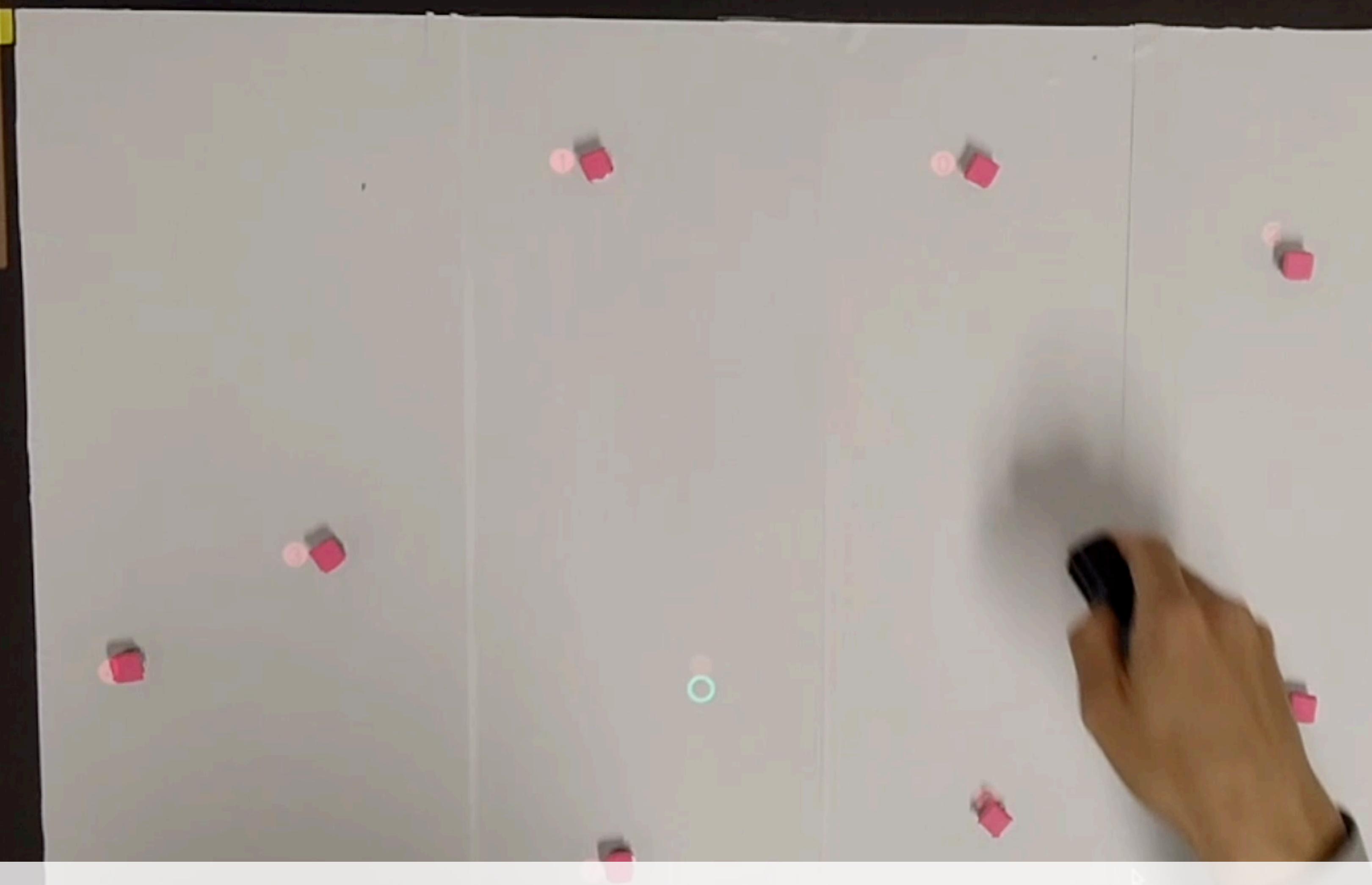


point



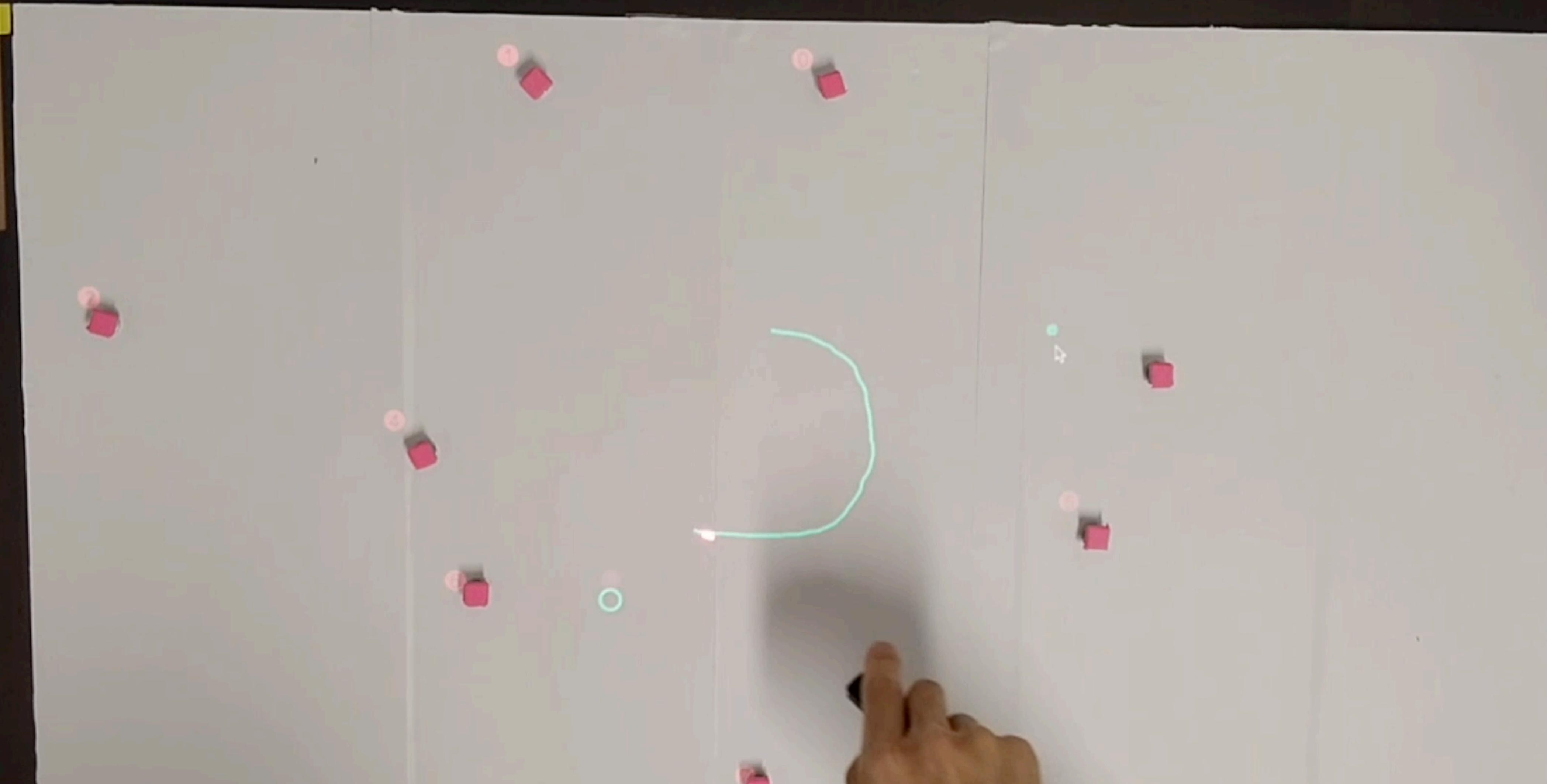
triangle

point



rectangle

point



circle

Step 1 Create Elements Step 2 Abstract Attributes Step 3 Specify Behaviors Step 4 Propagate Changes

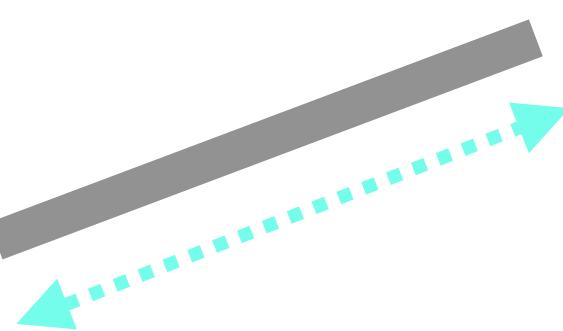
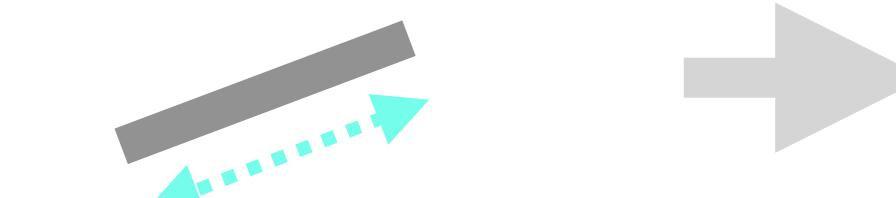
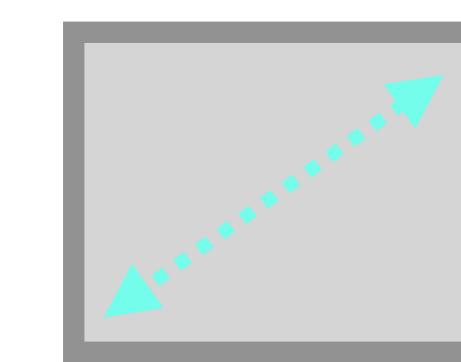
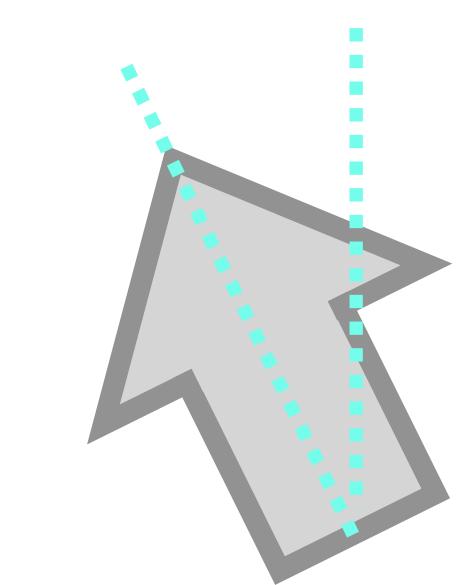
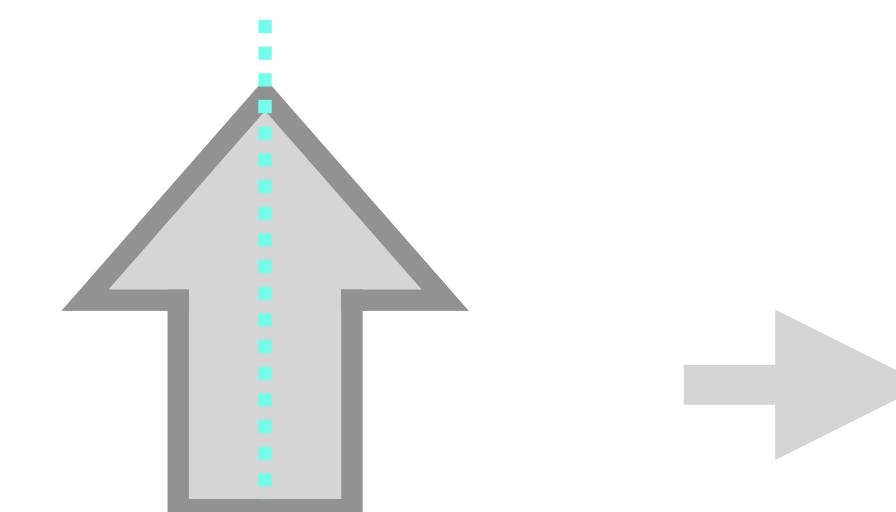
Swarm UI



```
var angle = 30  
<arrow angle={ angle } />
```

```
var scale = 2  
<rectangle scale={ scale } />
```

```
var length = 10  
<line length={ length } />
```



Step 1 Create Elements Step 2 Abstract Attributes Step 3 Specify Behaviors Step 4 Propagate Changes

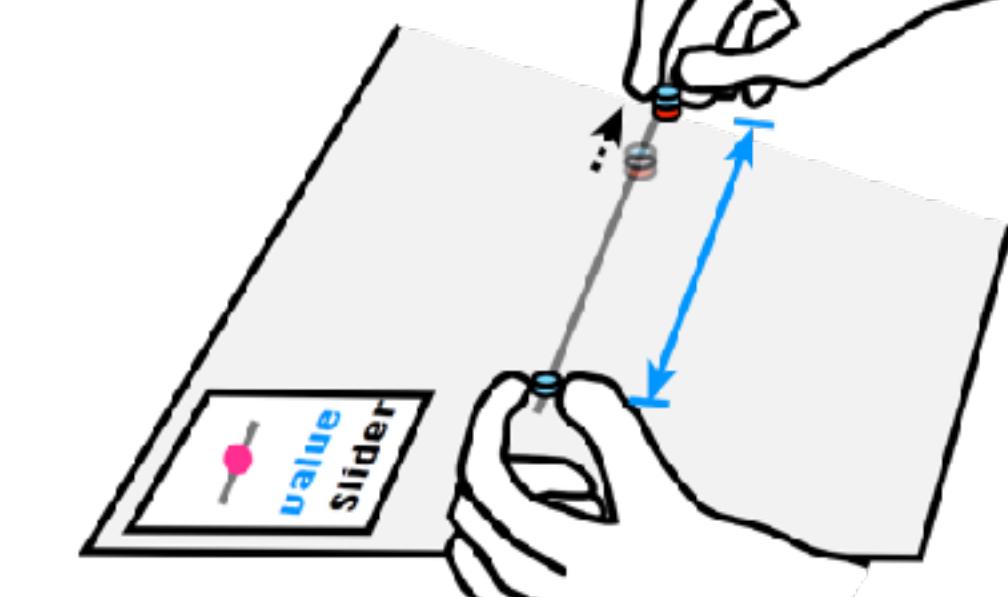
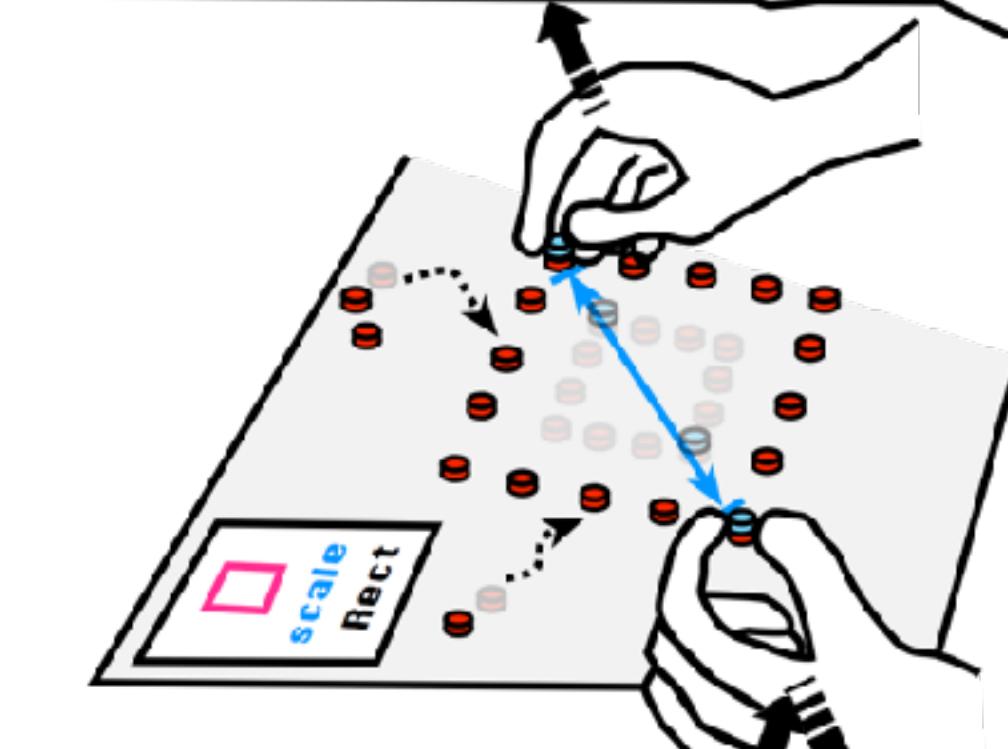
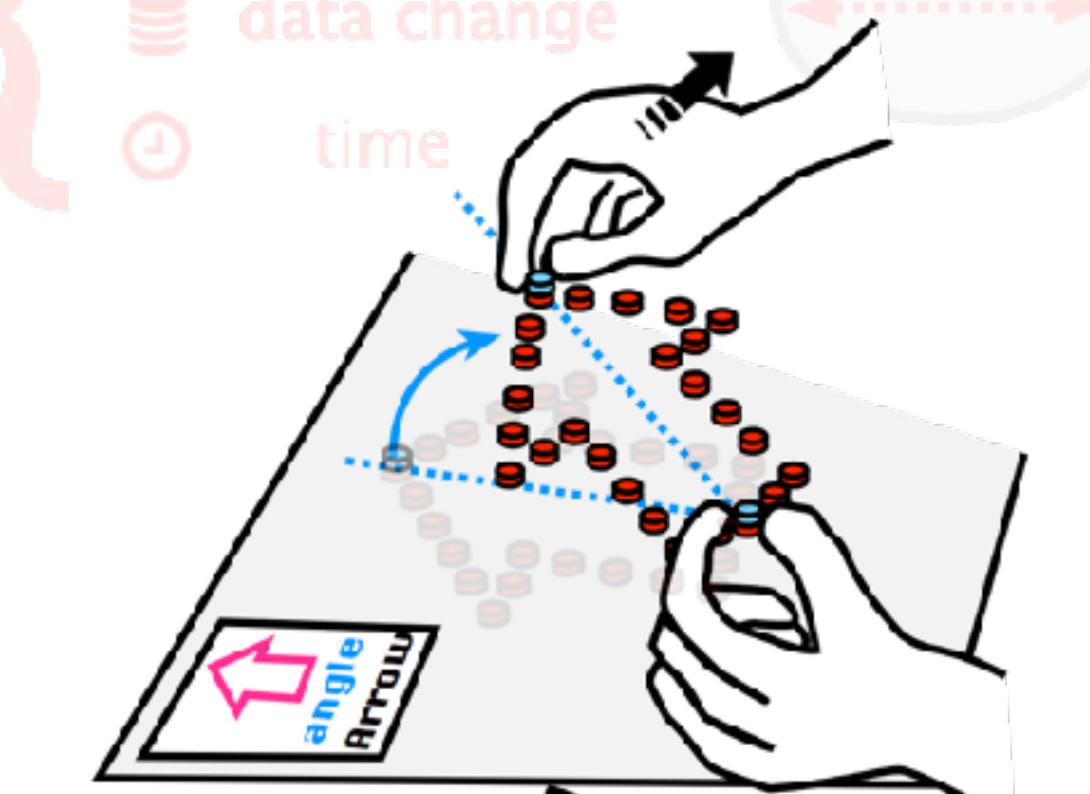
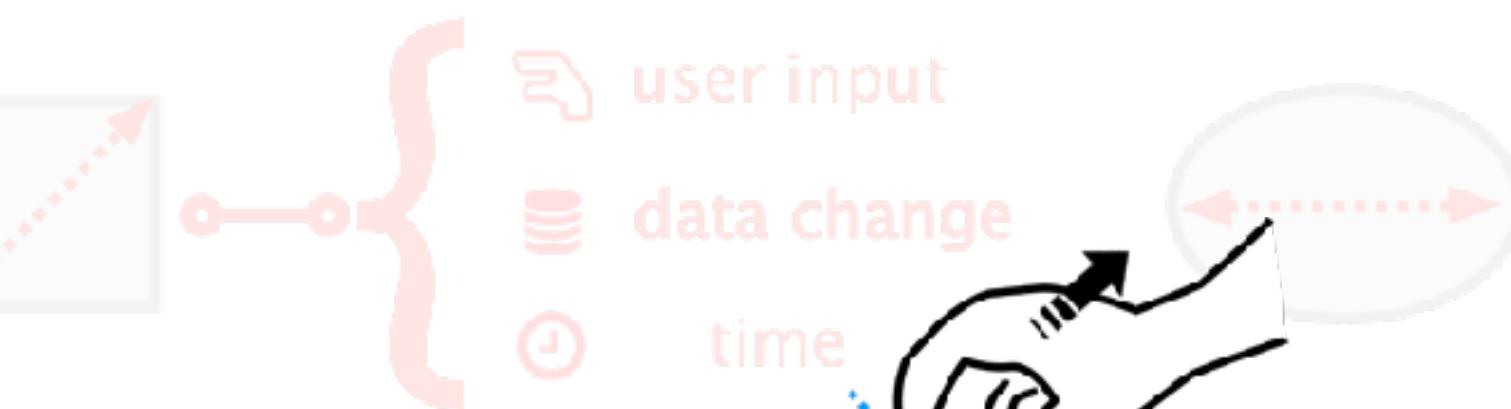
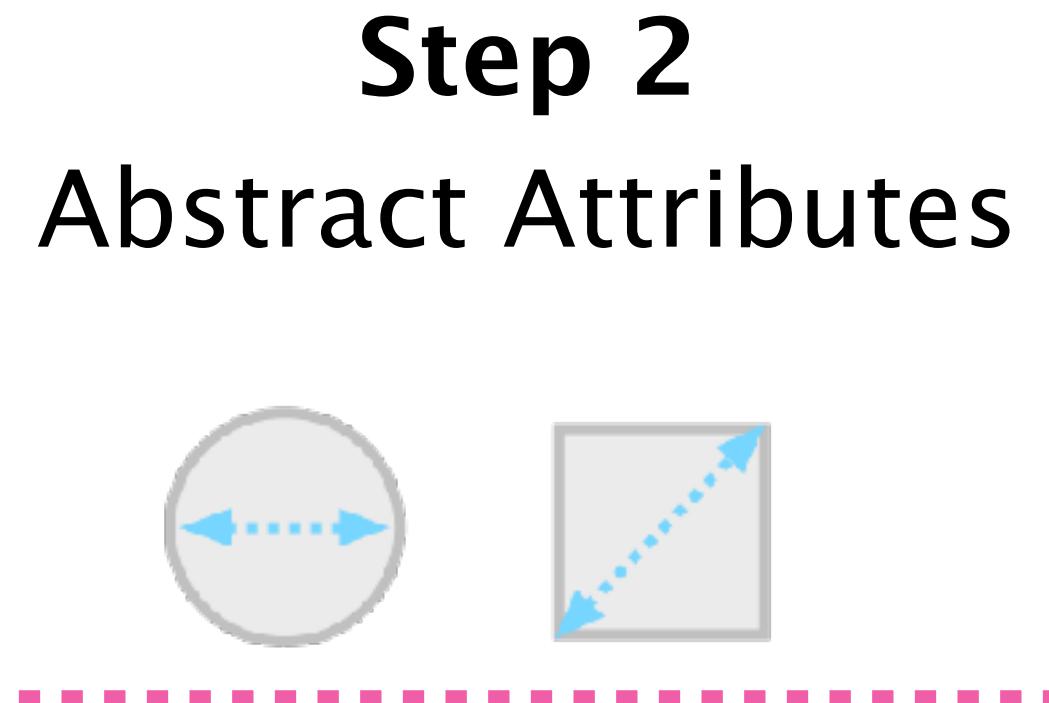
Swarm UI

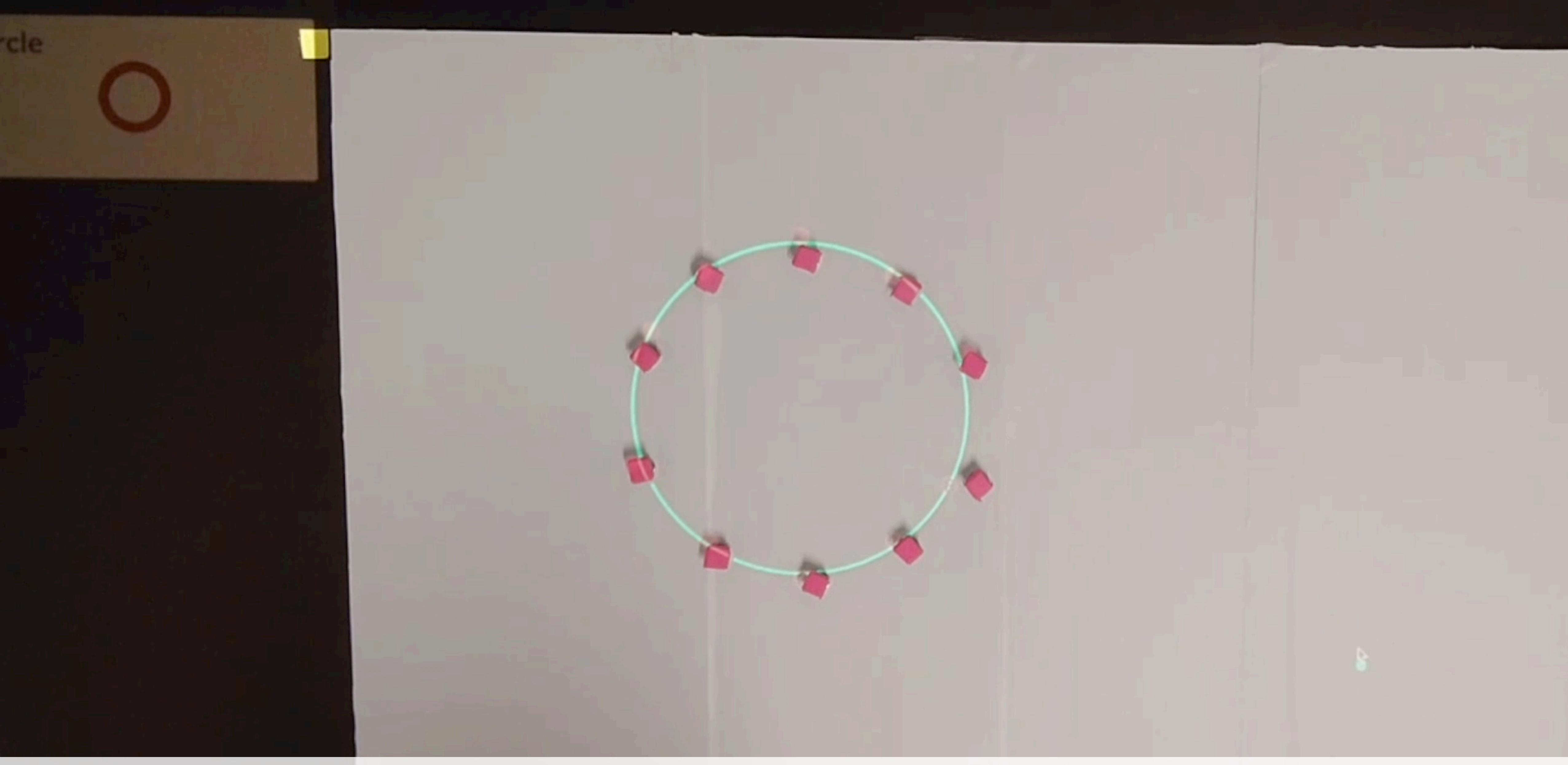


```
var angle = 30  
<arrow angle={ angle } />
```

```
var scale = 2  
<rectangle scale={ scale } />
```

```
var length = 10  
<line length={ length } />
```





circle: diameter

rectangle: width → scale → height

Point



point: $x \rightarrow \text{angle} \rightarrow y$

Step 1

Create Elements

Swarm UI



```
<slider value={ value } />  
<arrow angle={ angle } />
```



```
bind(angle, value / 10)
```

```
<slider value={ value } />  
<arrow angle={ angle } />
```

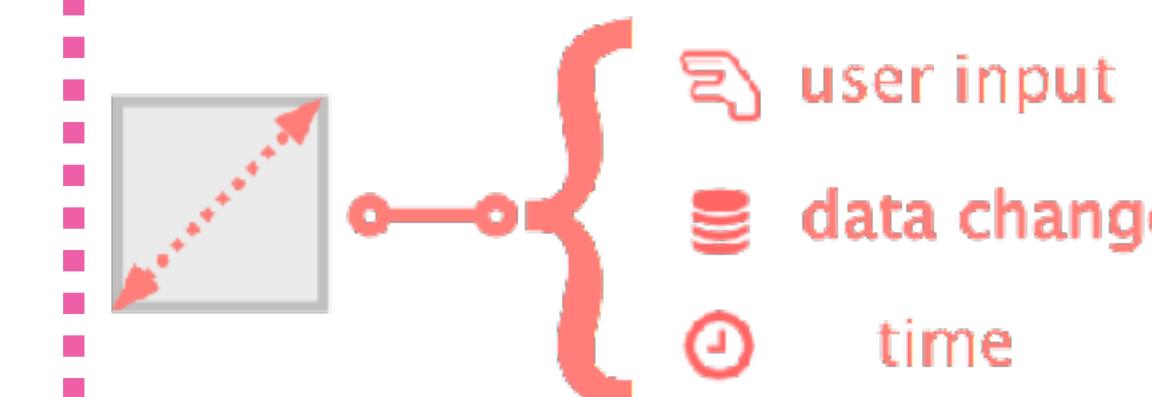
Step 2

Abstract Attributes



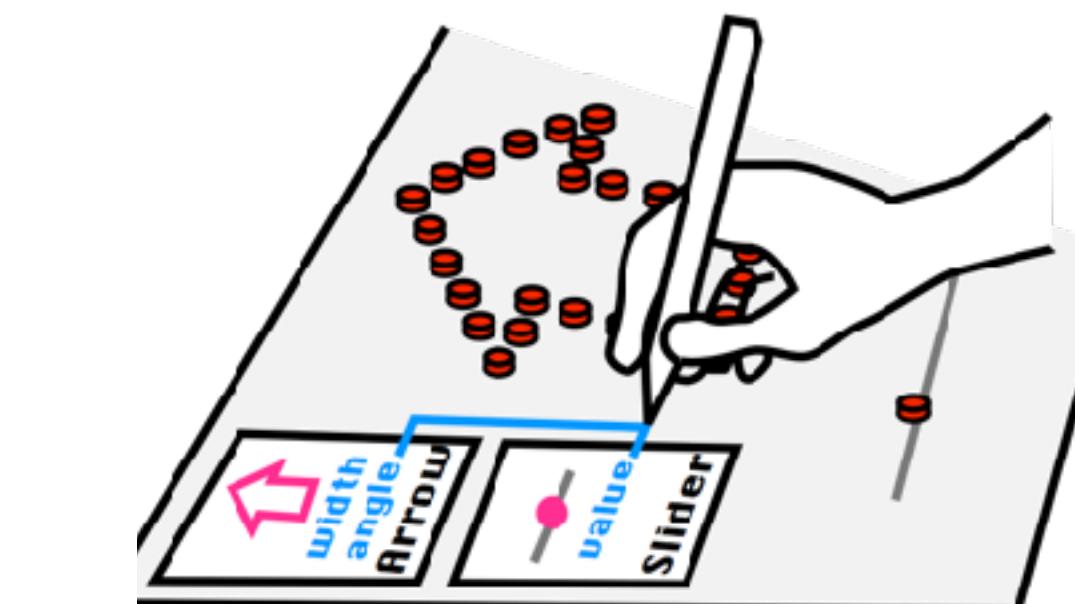
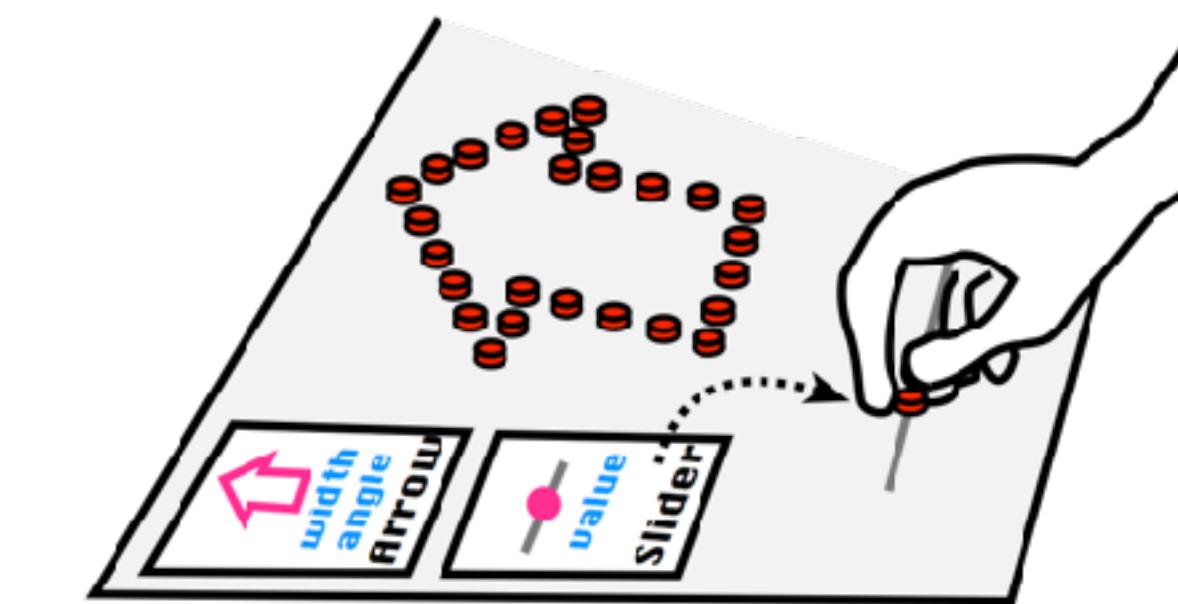
Step 3

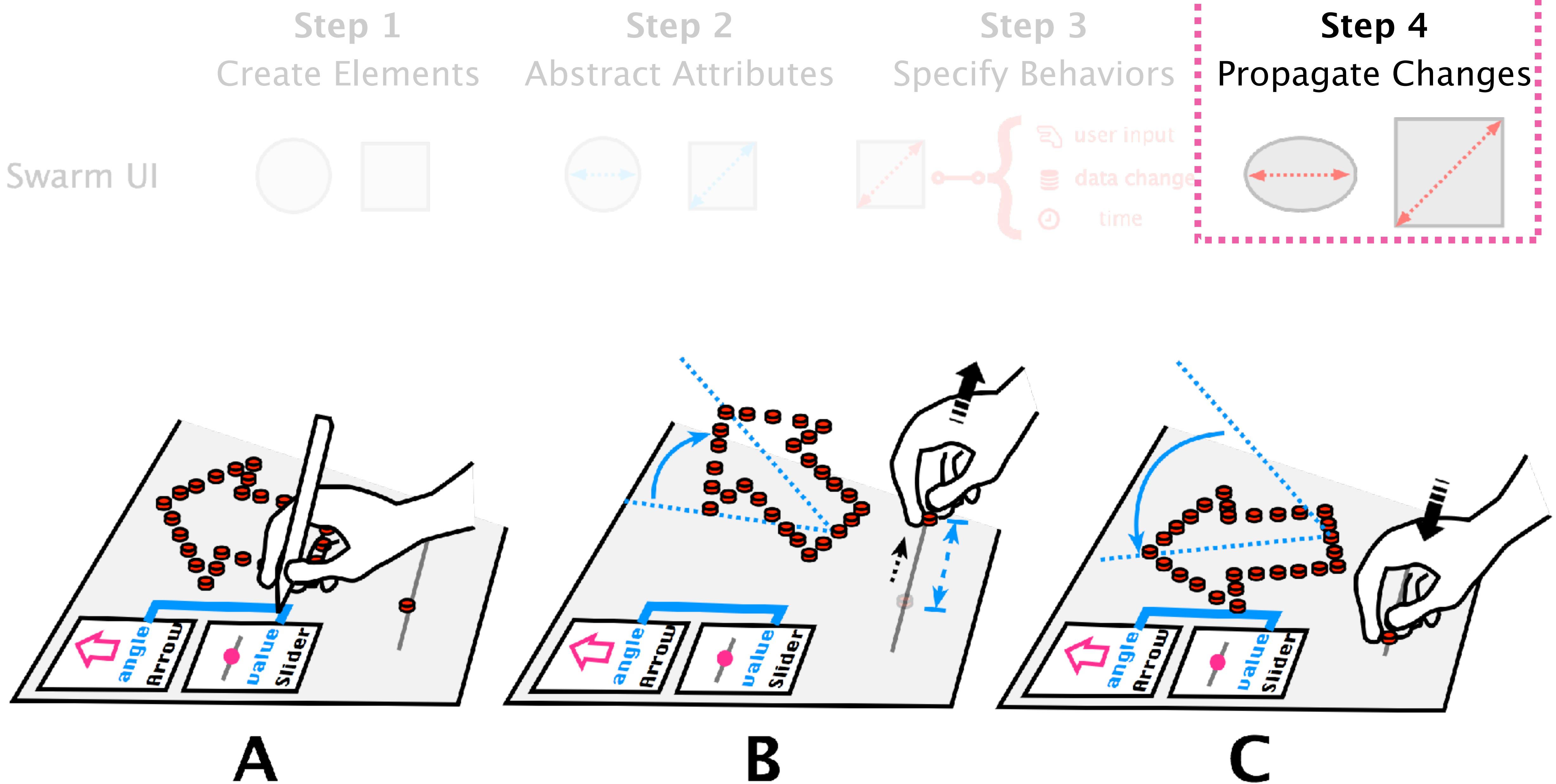
Specify Behaviors



Step 4

Propagate Changes



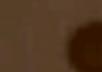


ect

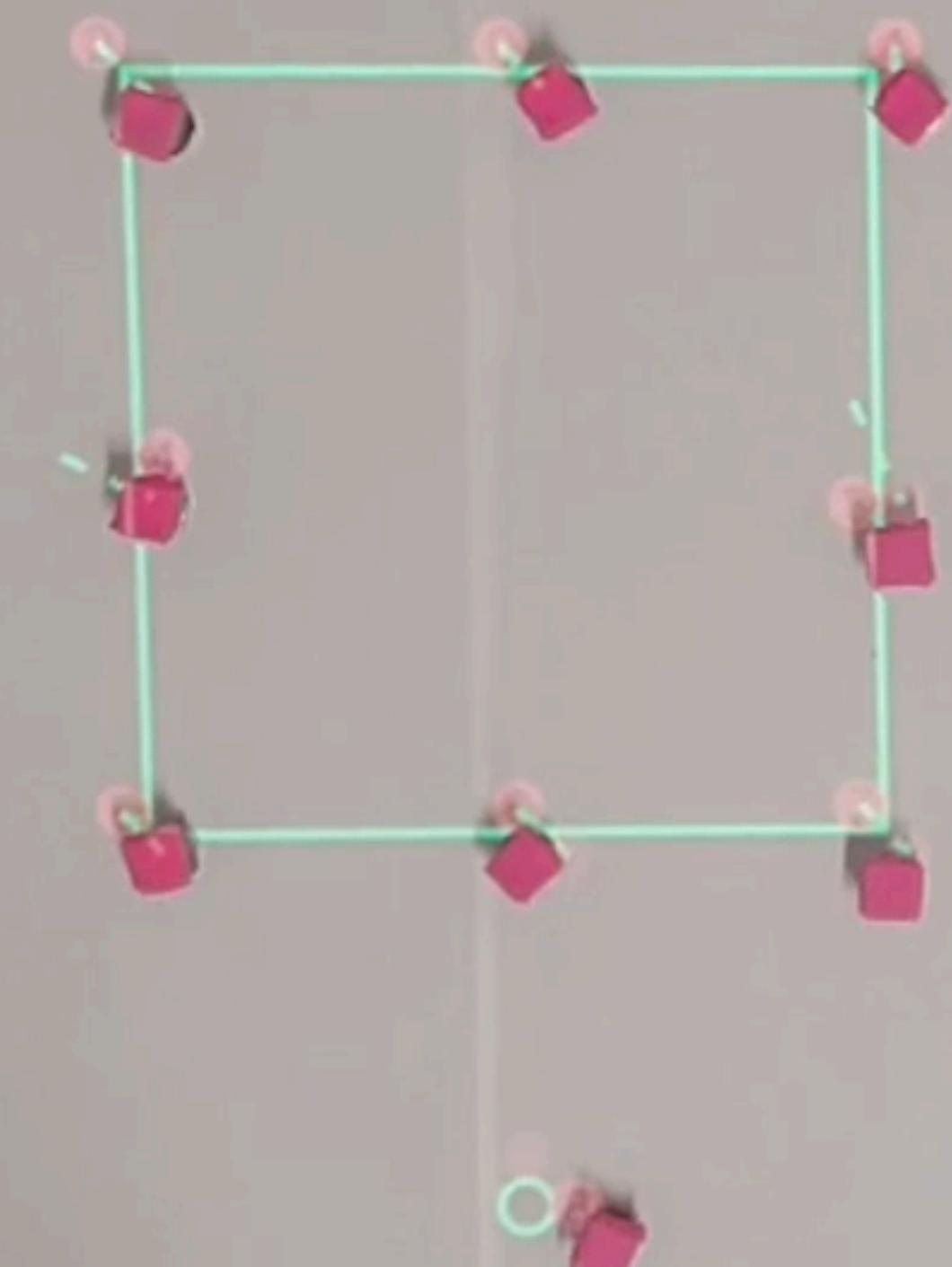


width: 15

Point



x: 20



bind(rectangle.width, point.x - 5)

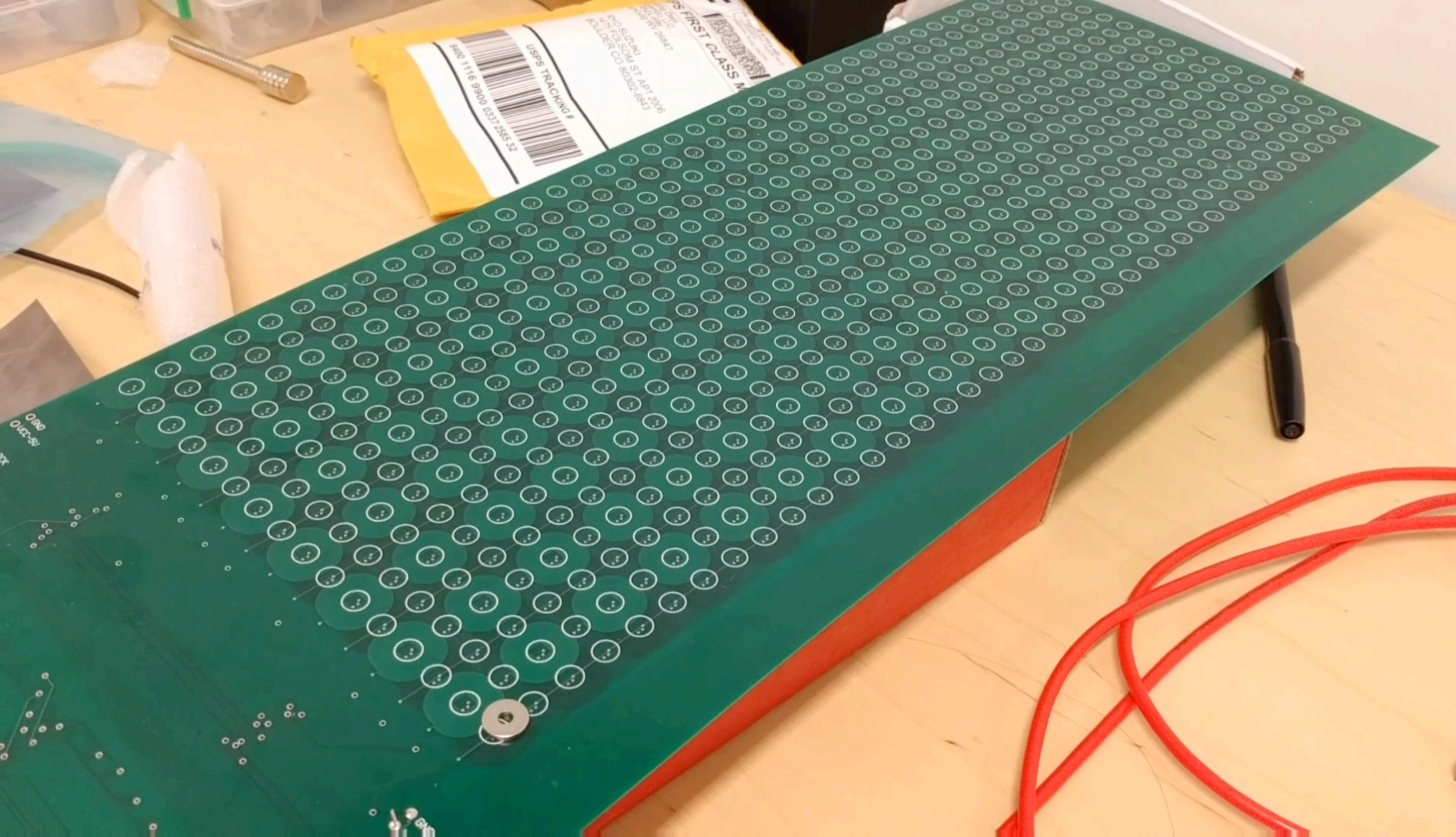
1. Summary

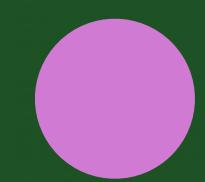
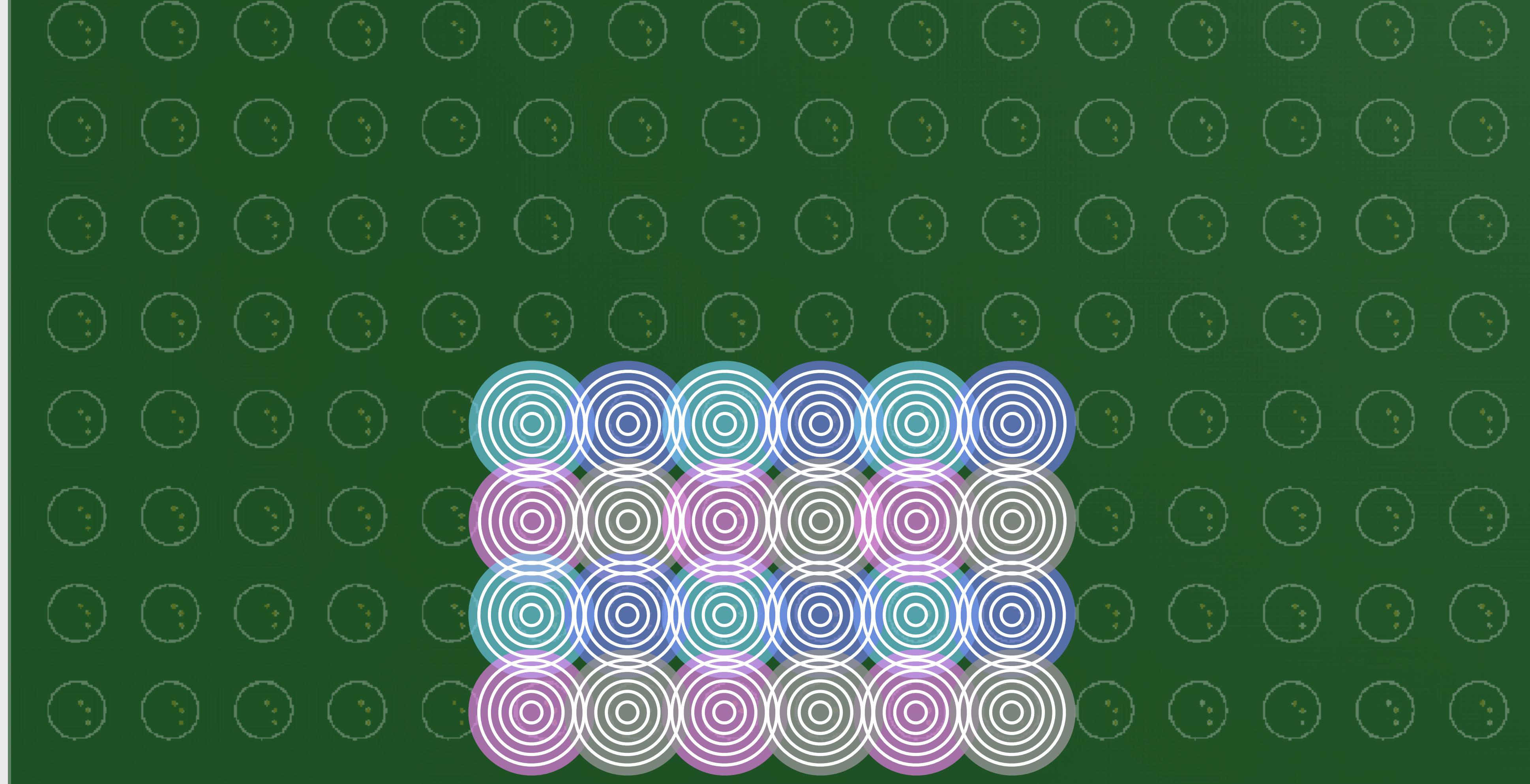
2. Related Work

3. Design and Workflow

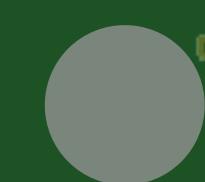
4. Implementation

5. Limitations and Future Work

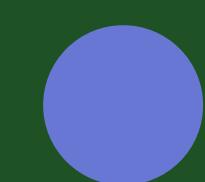




1st Layer



2nd Layer



3rd Layer



4th Layer



GND



VCC-5V



P_CLOCK

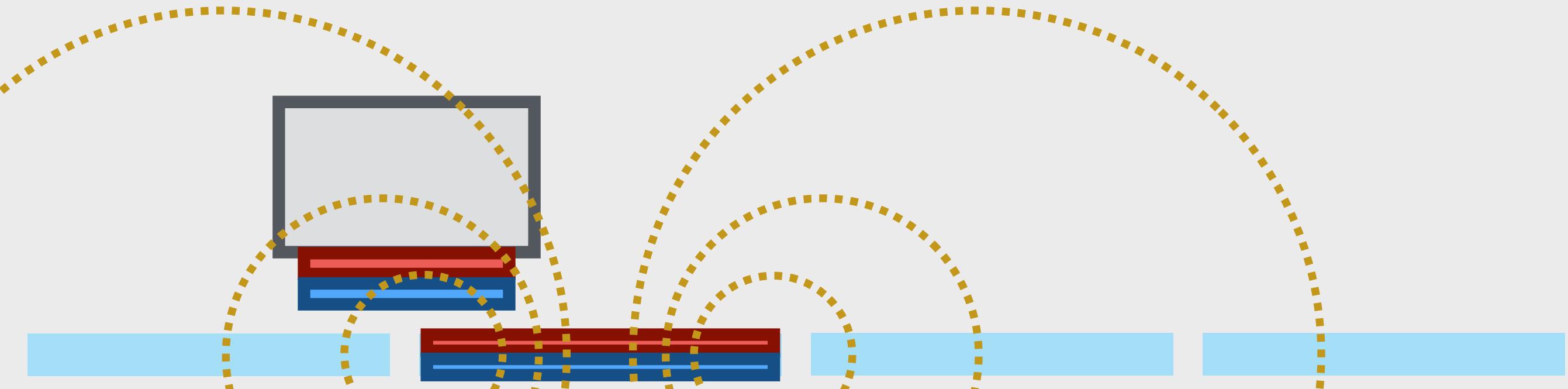


P_LATCH

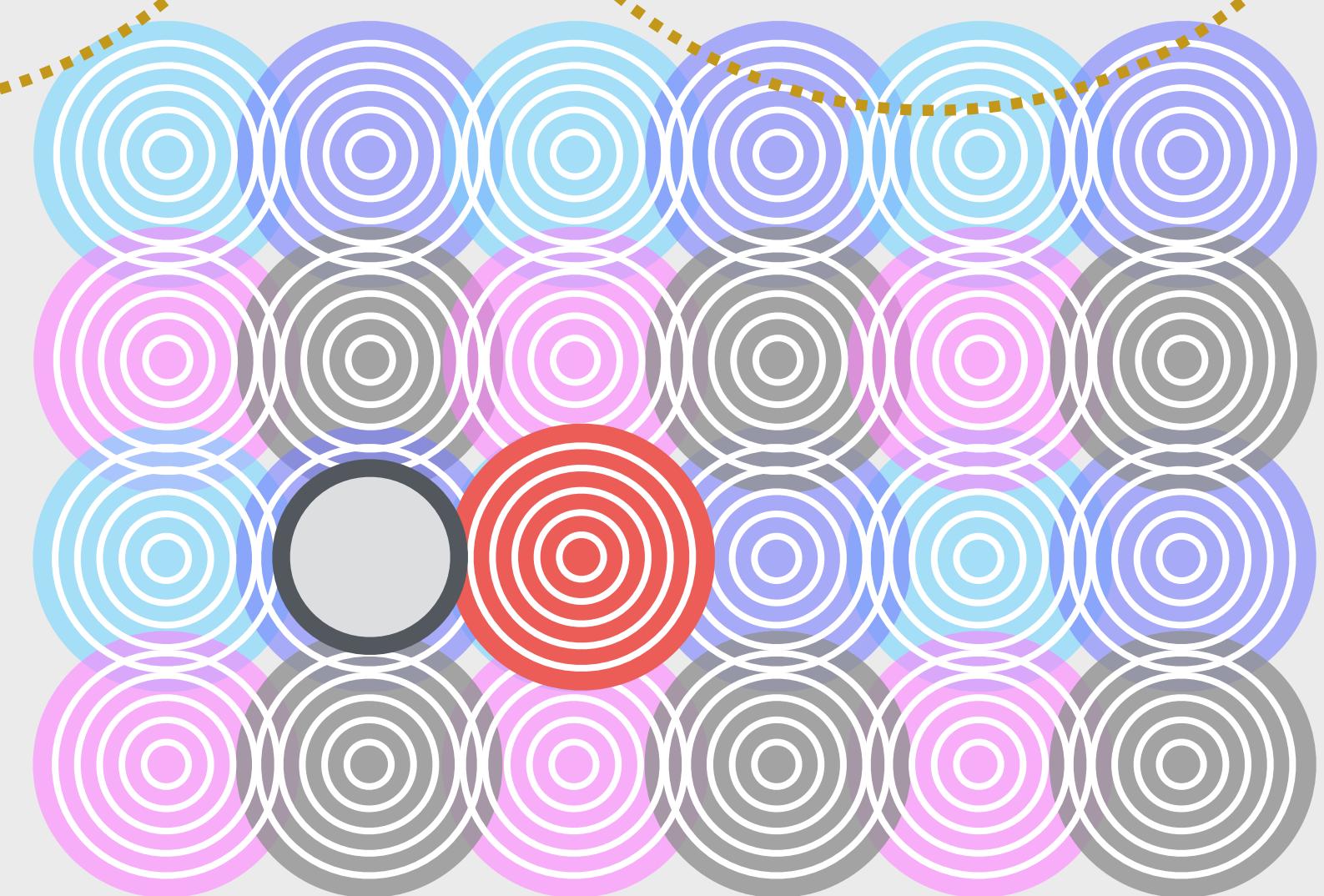


P_DATA_OUT

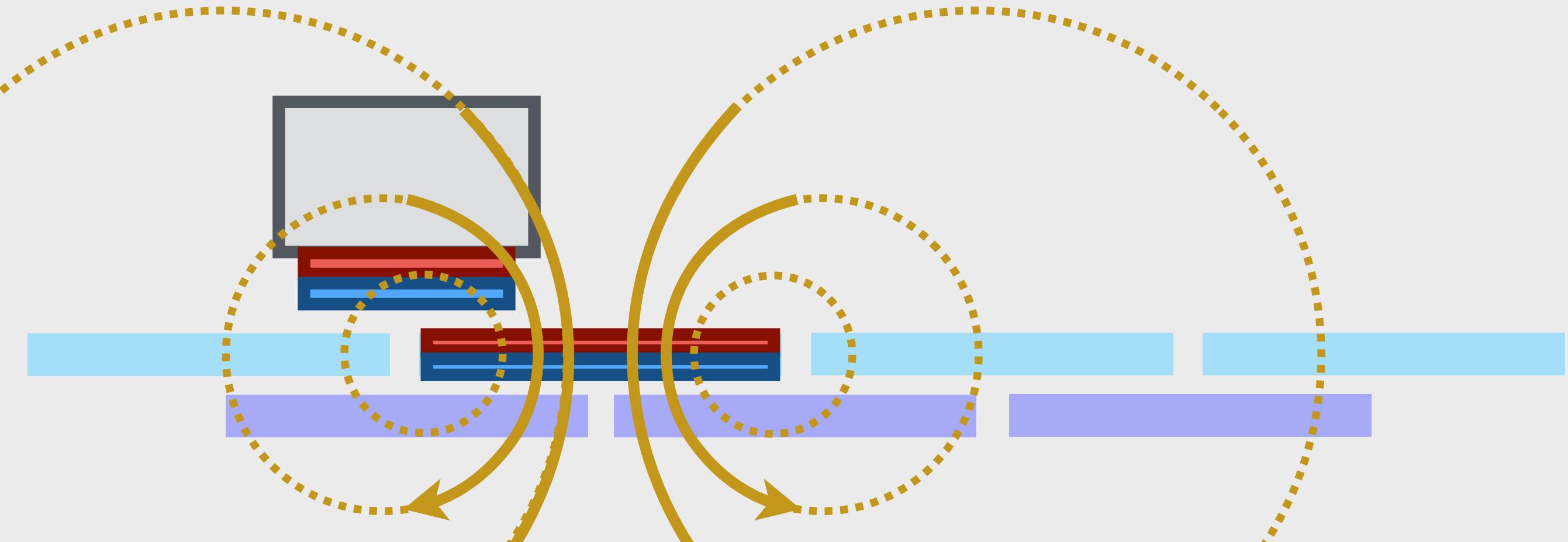
Side View



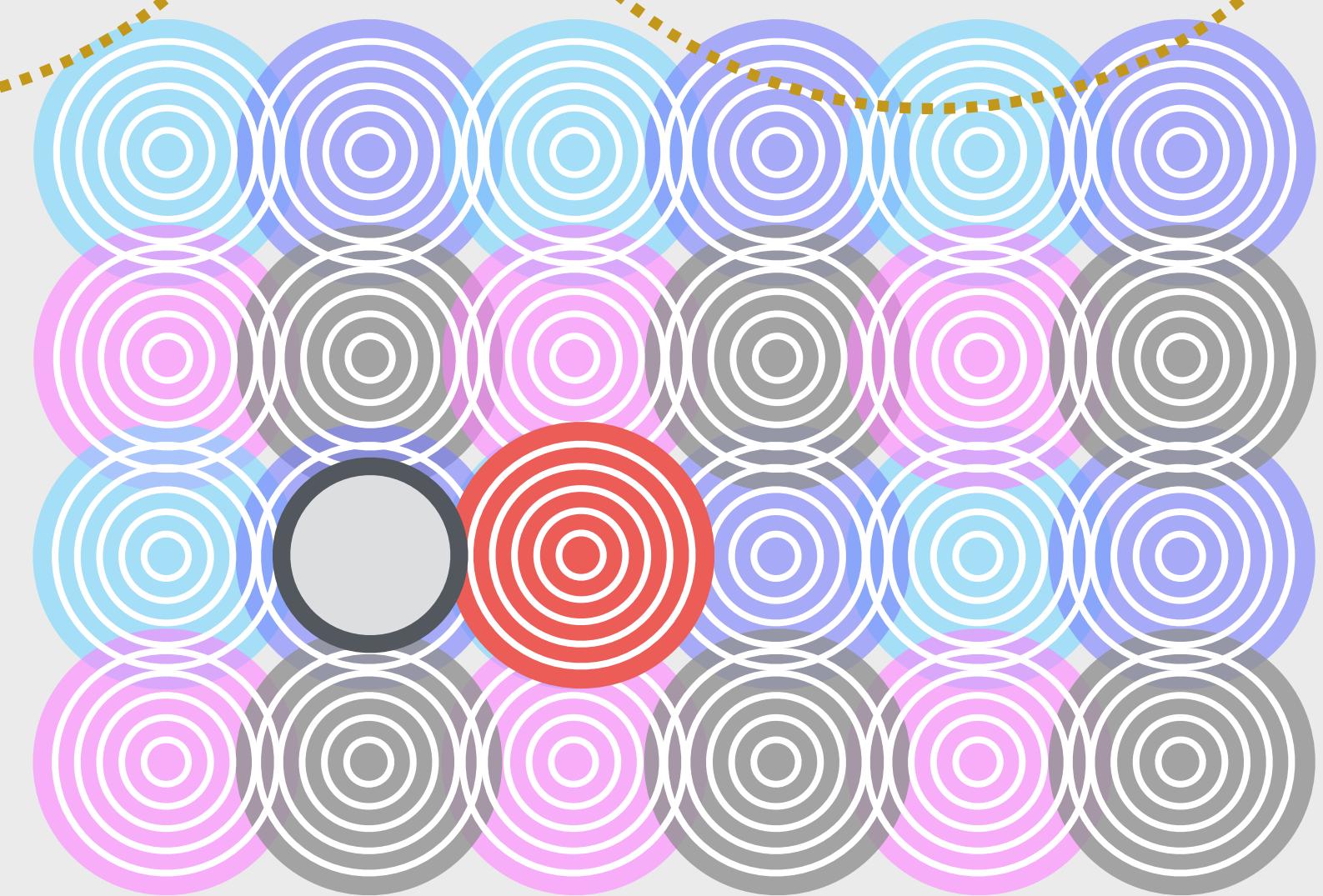
Top View



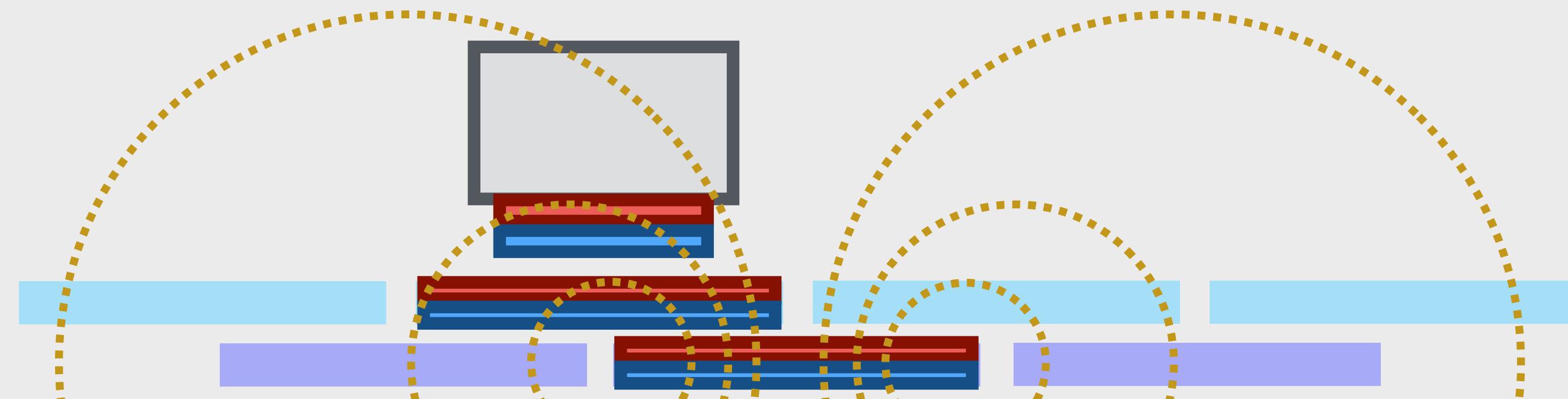
Side View



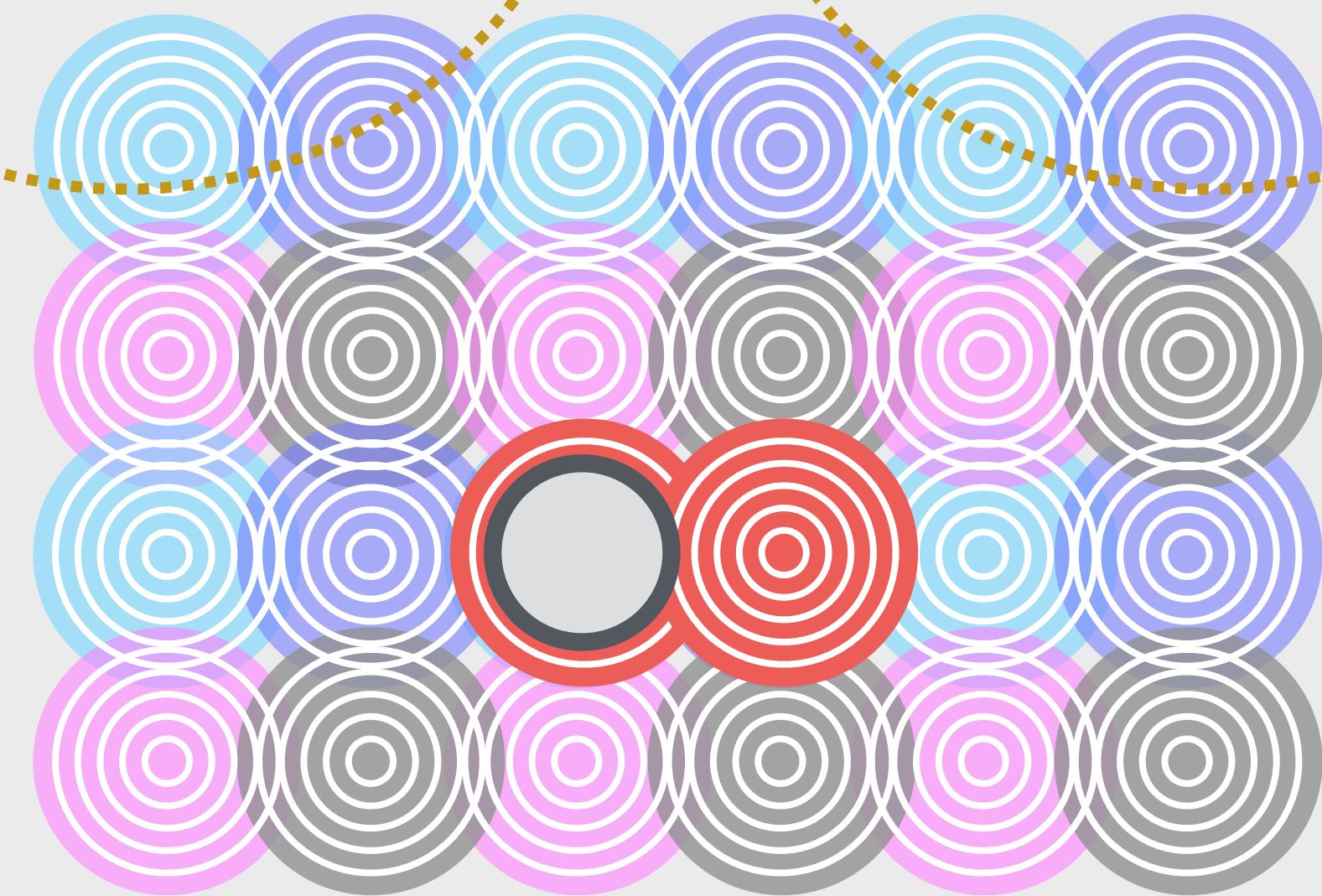
Top View



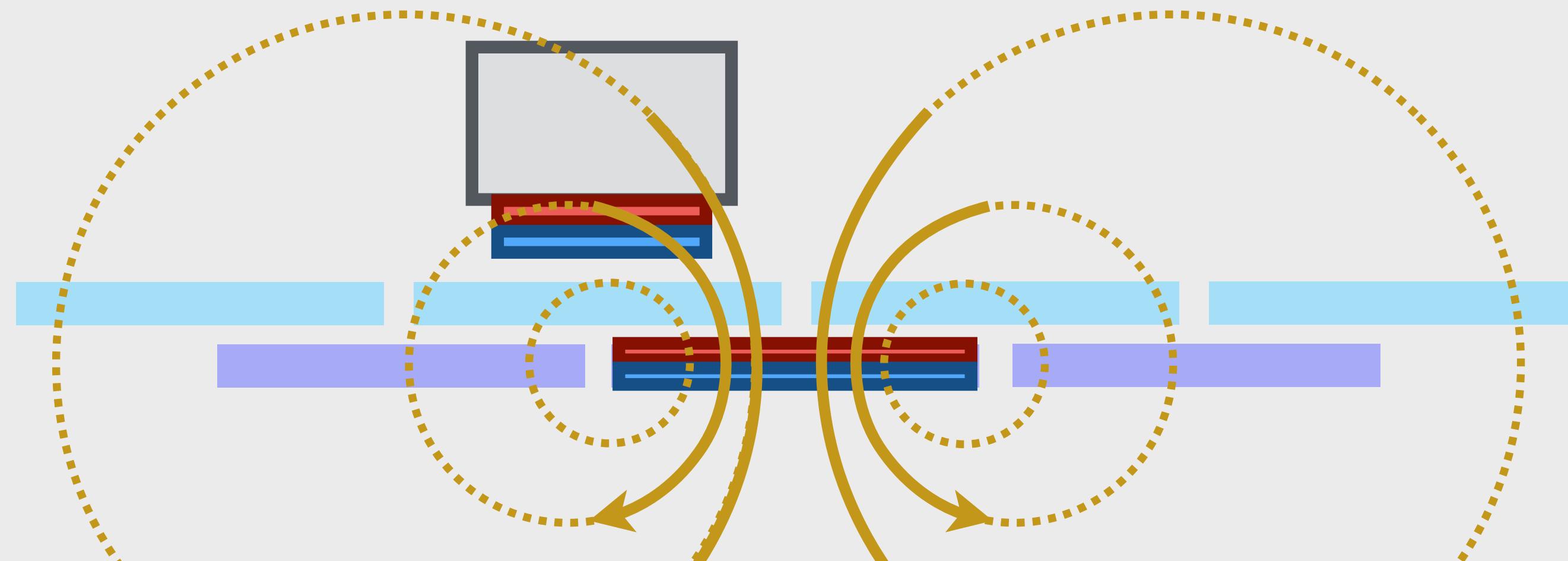
Side View



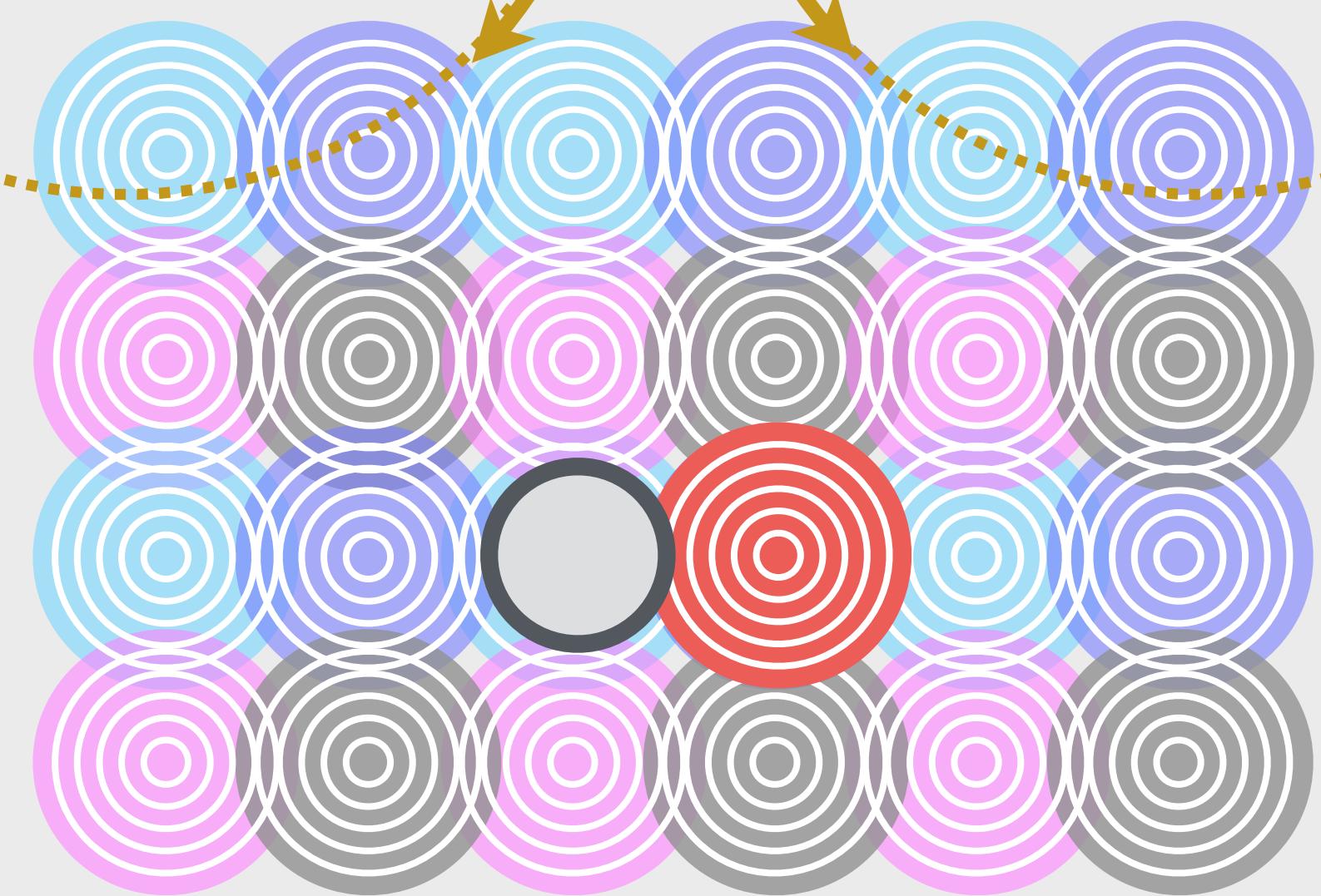
Top View



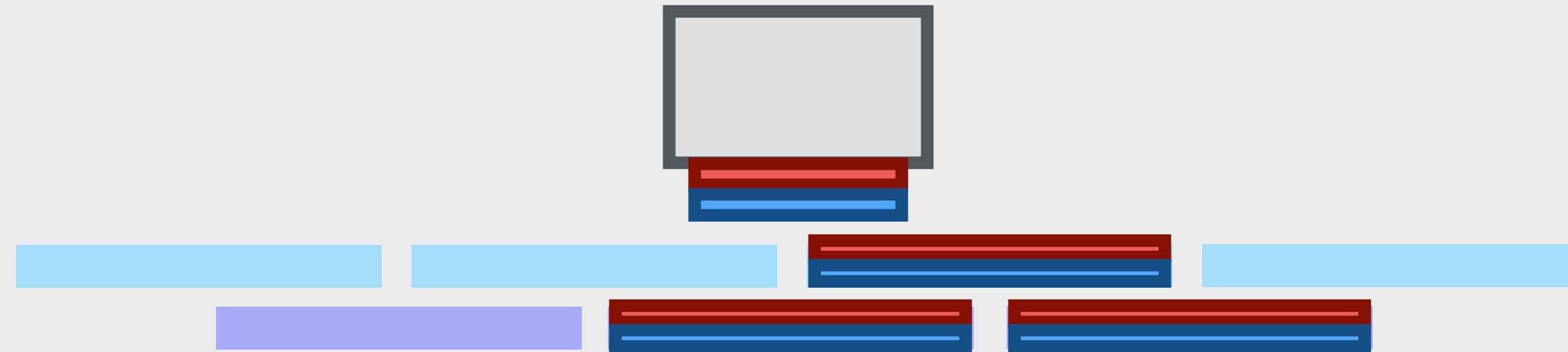
Side View



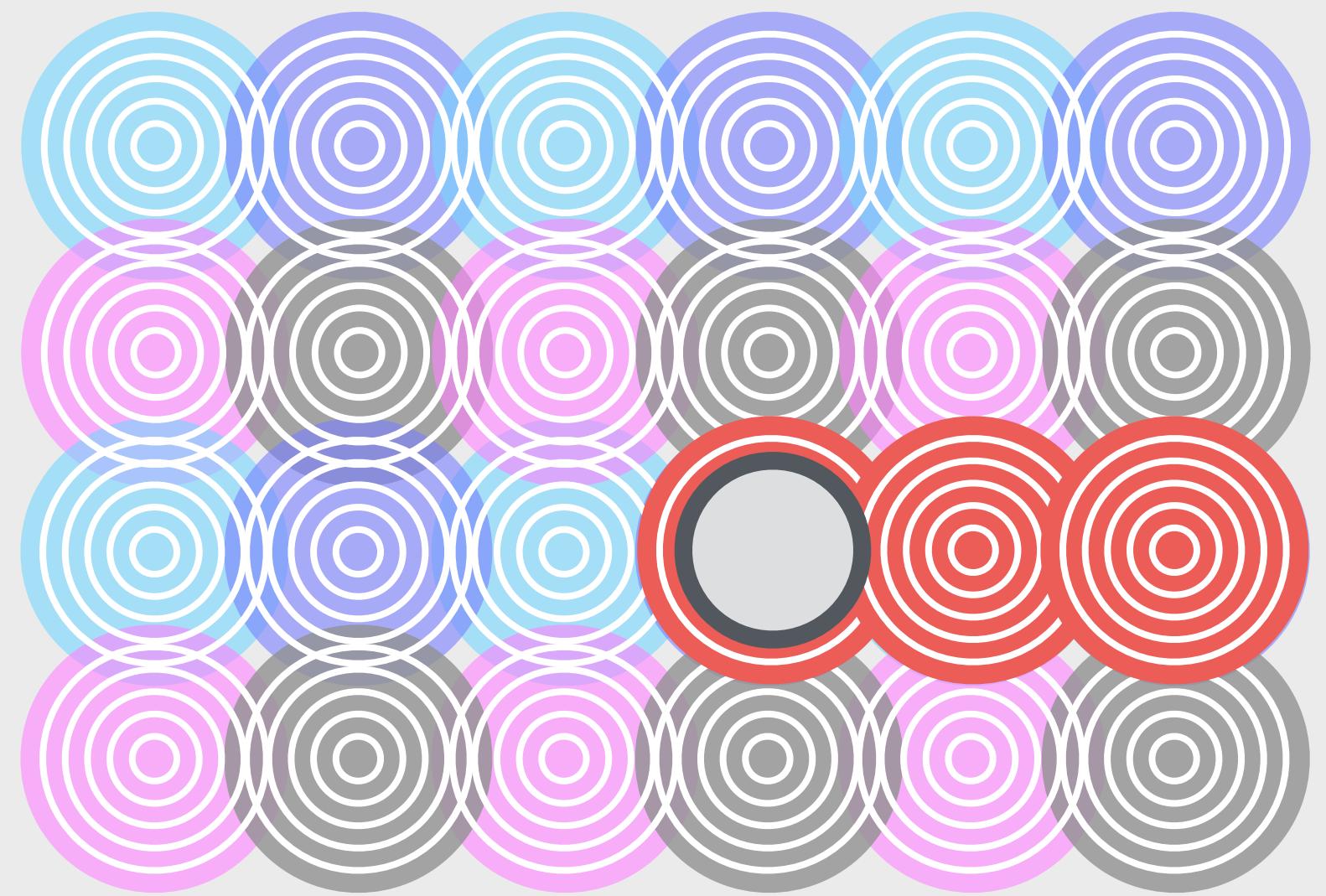
Top View

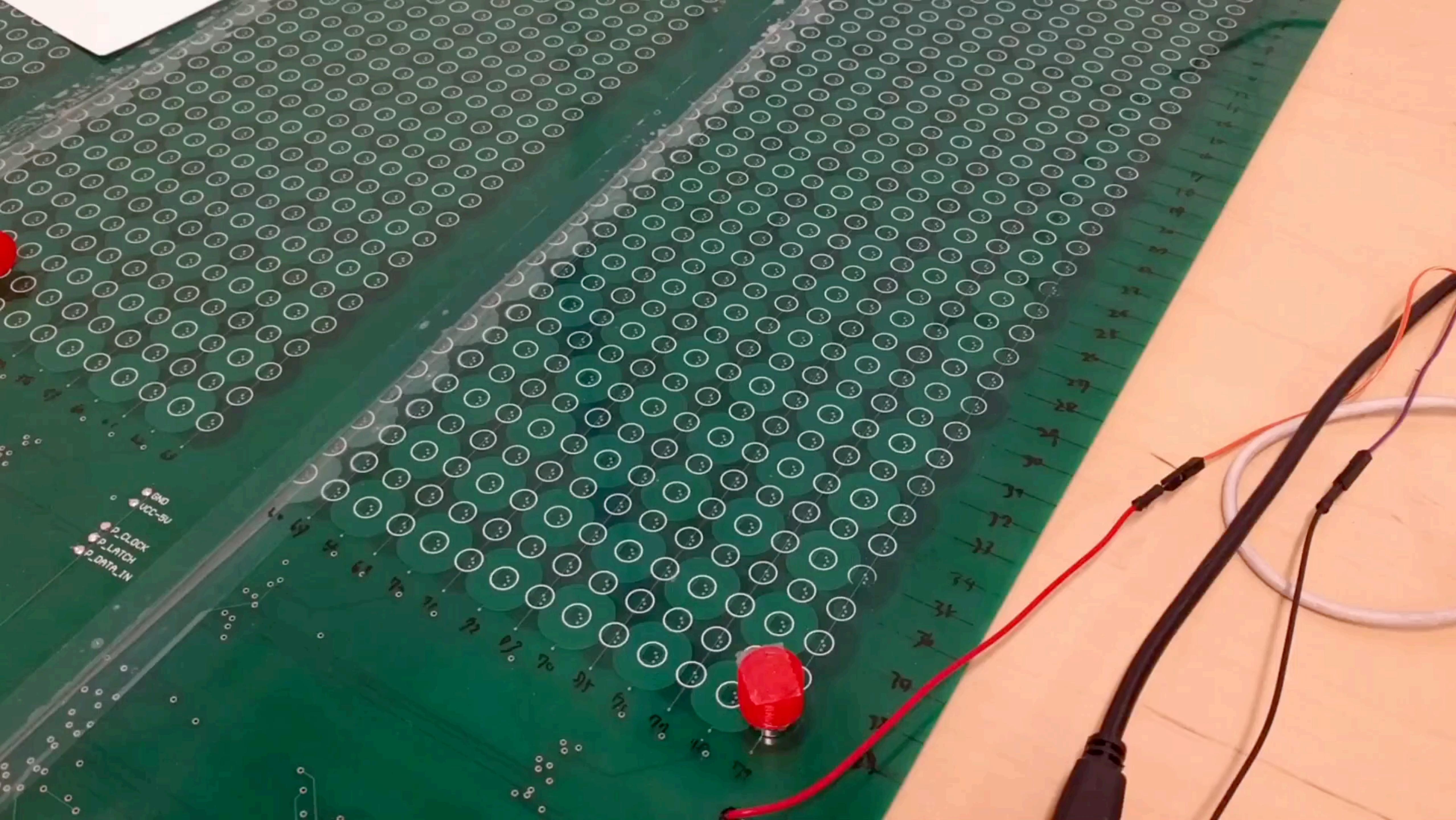


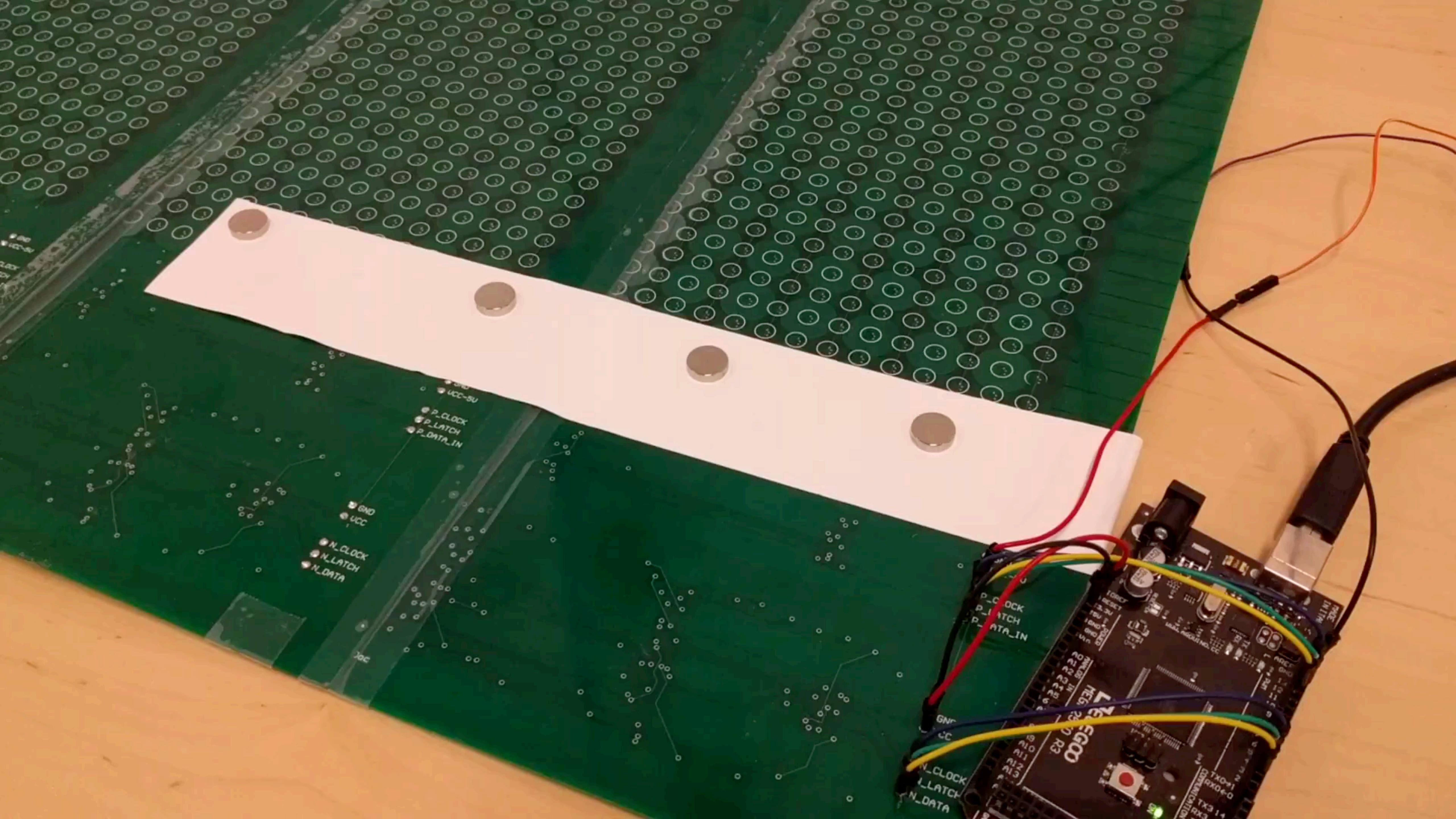
Side View



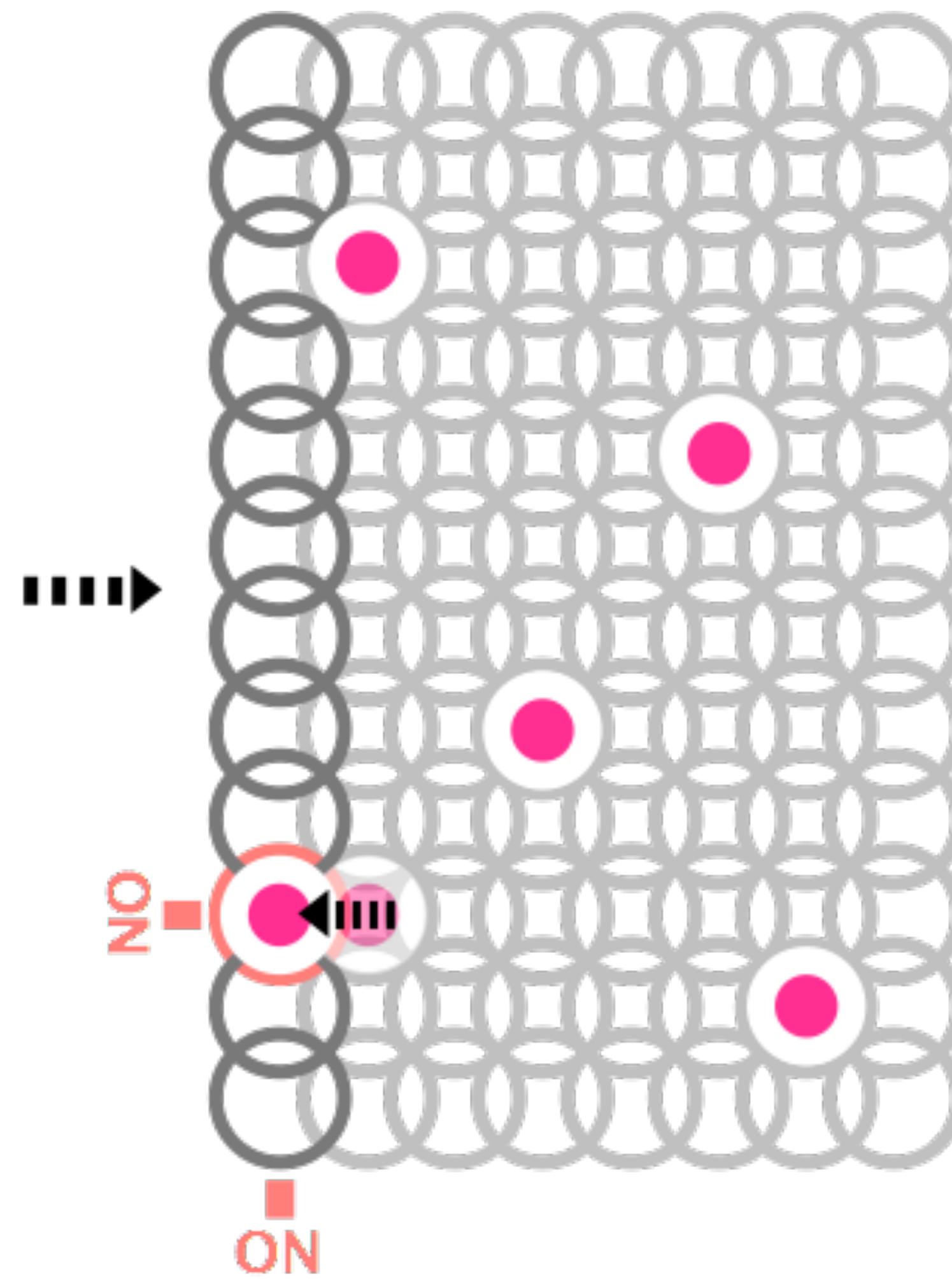
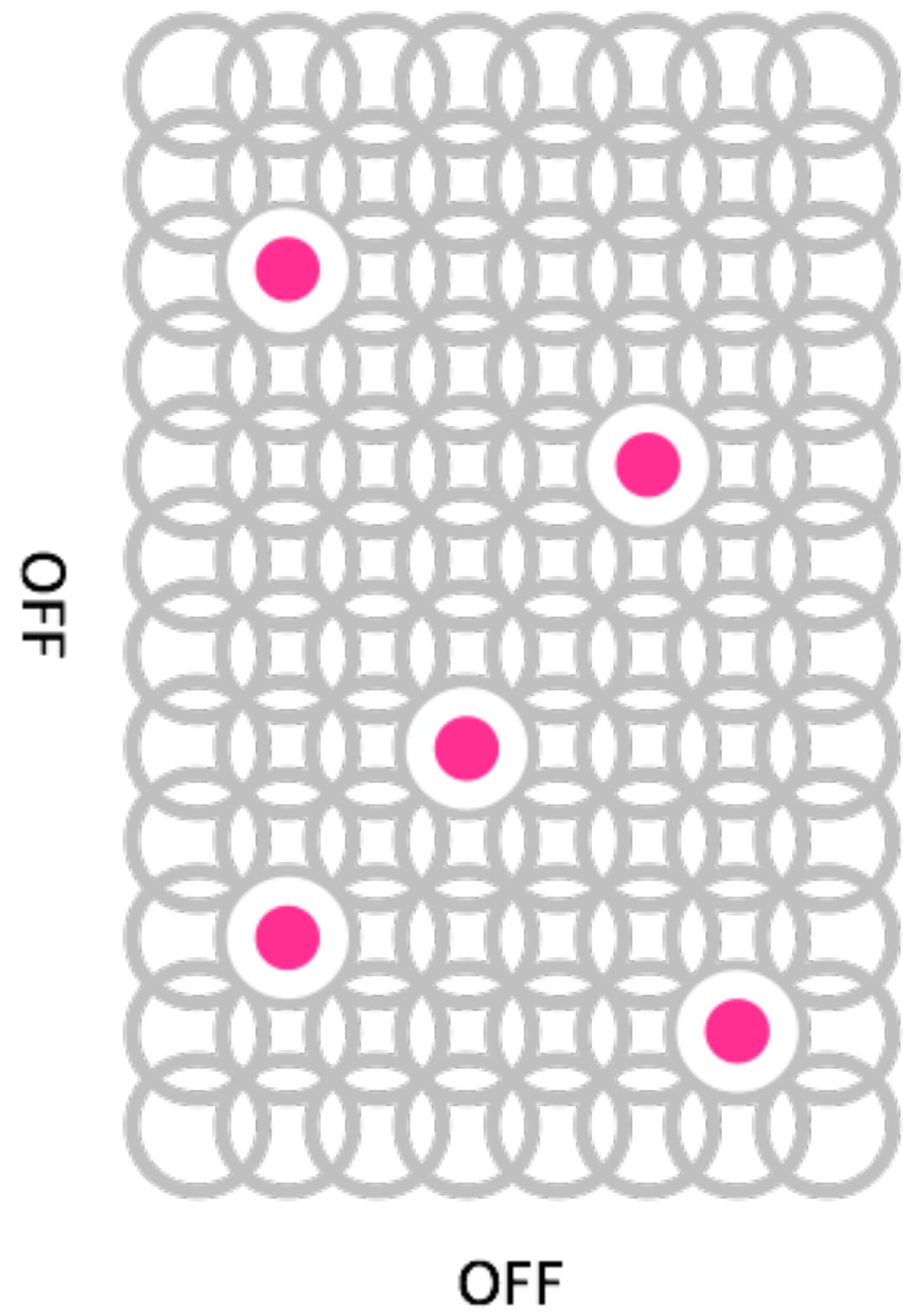
Top View



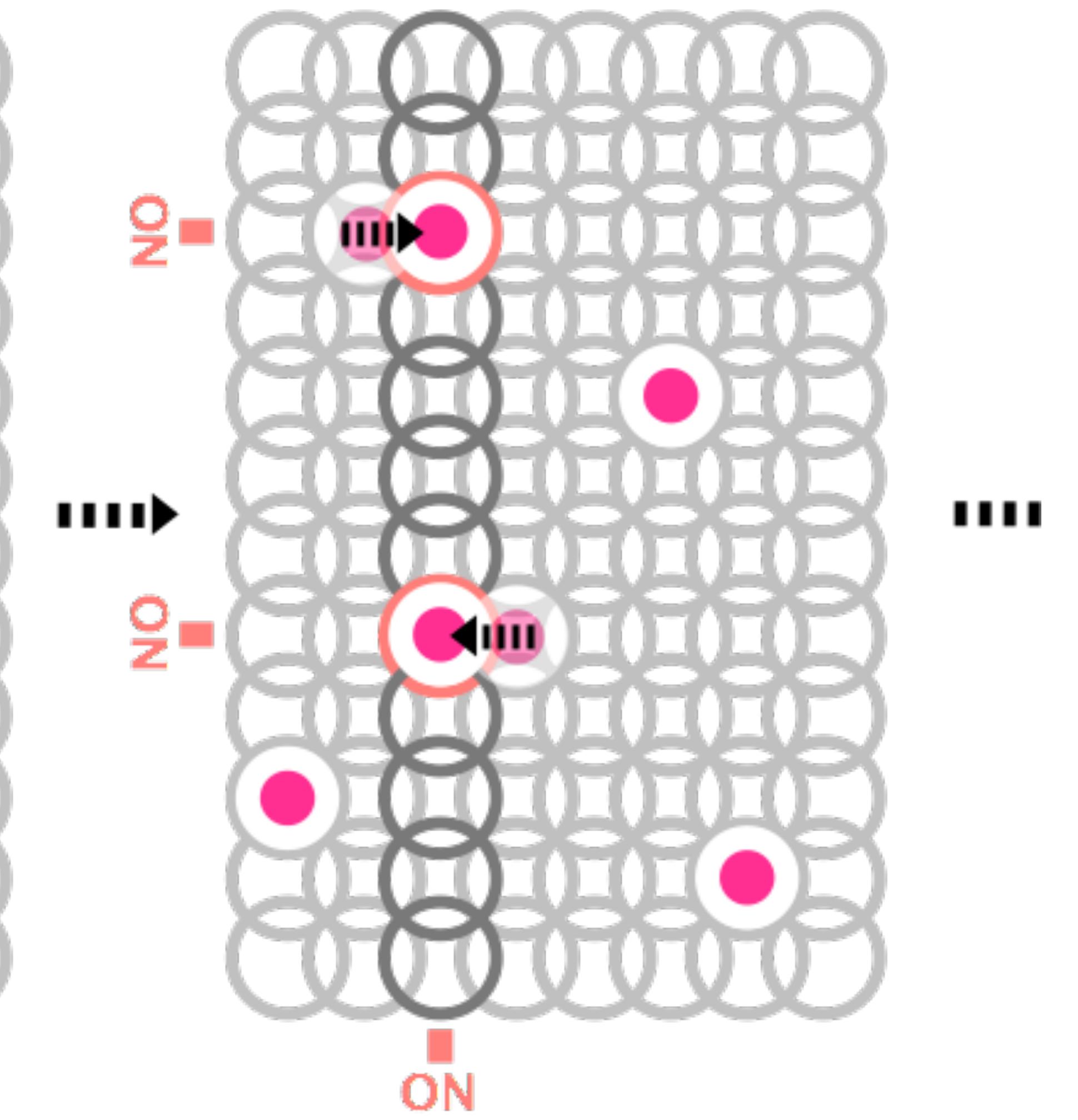


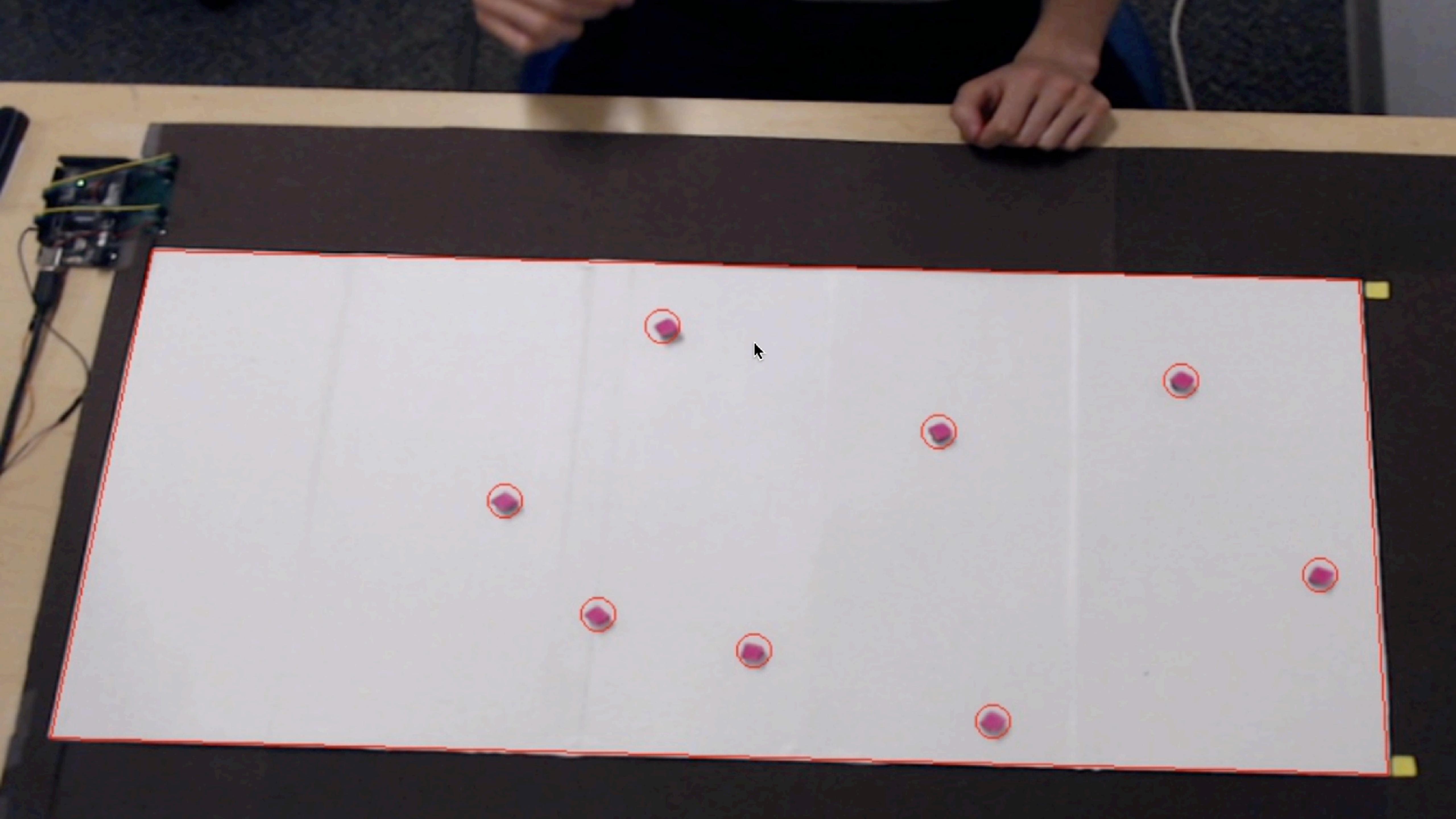


N-ch MOSFETs



P-ch MOSFETs





1. Summary

2. Related Work

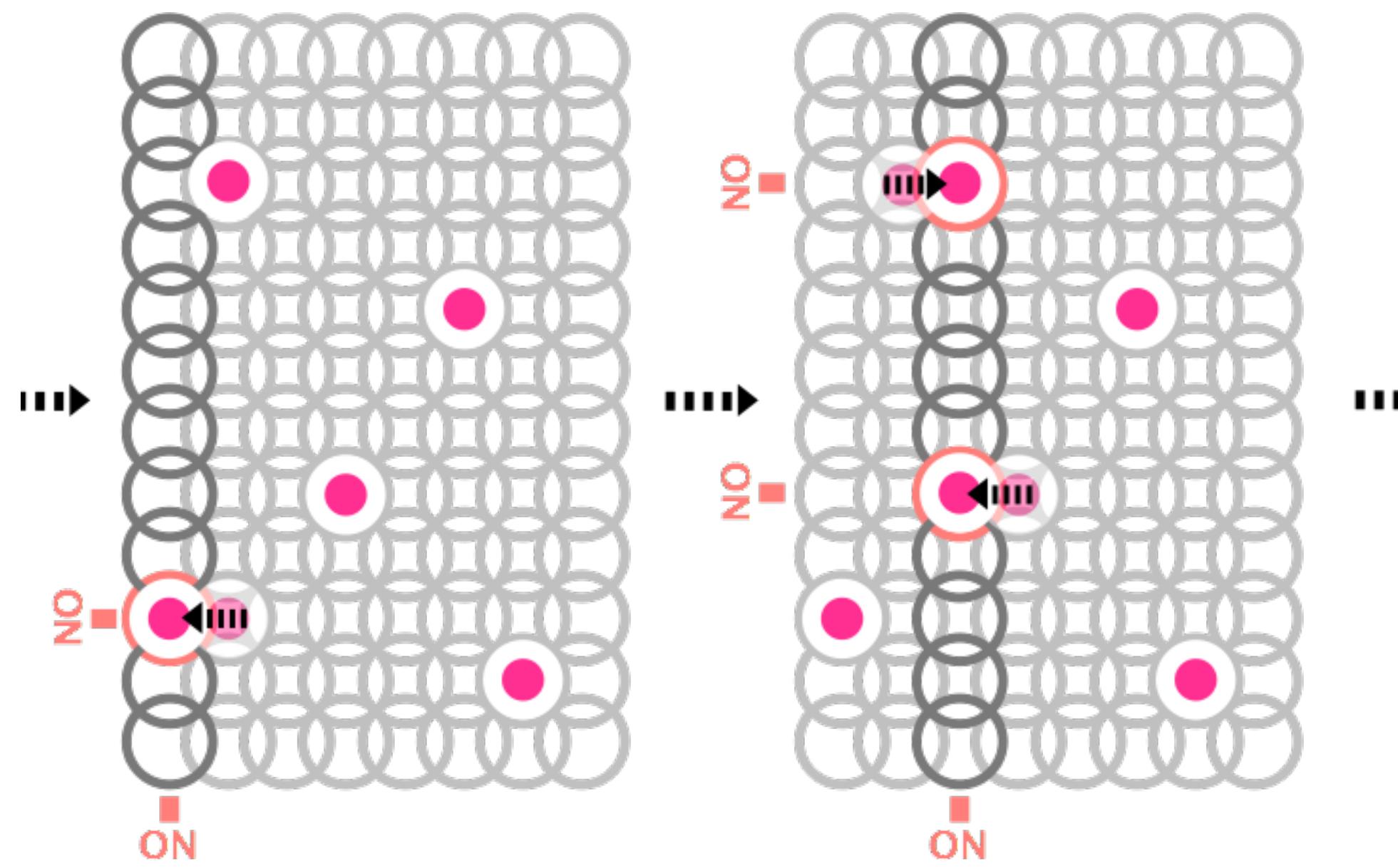
3. Interaction Design

4. Implementation

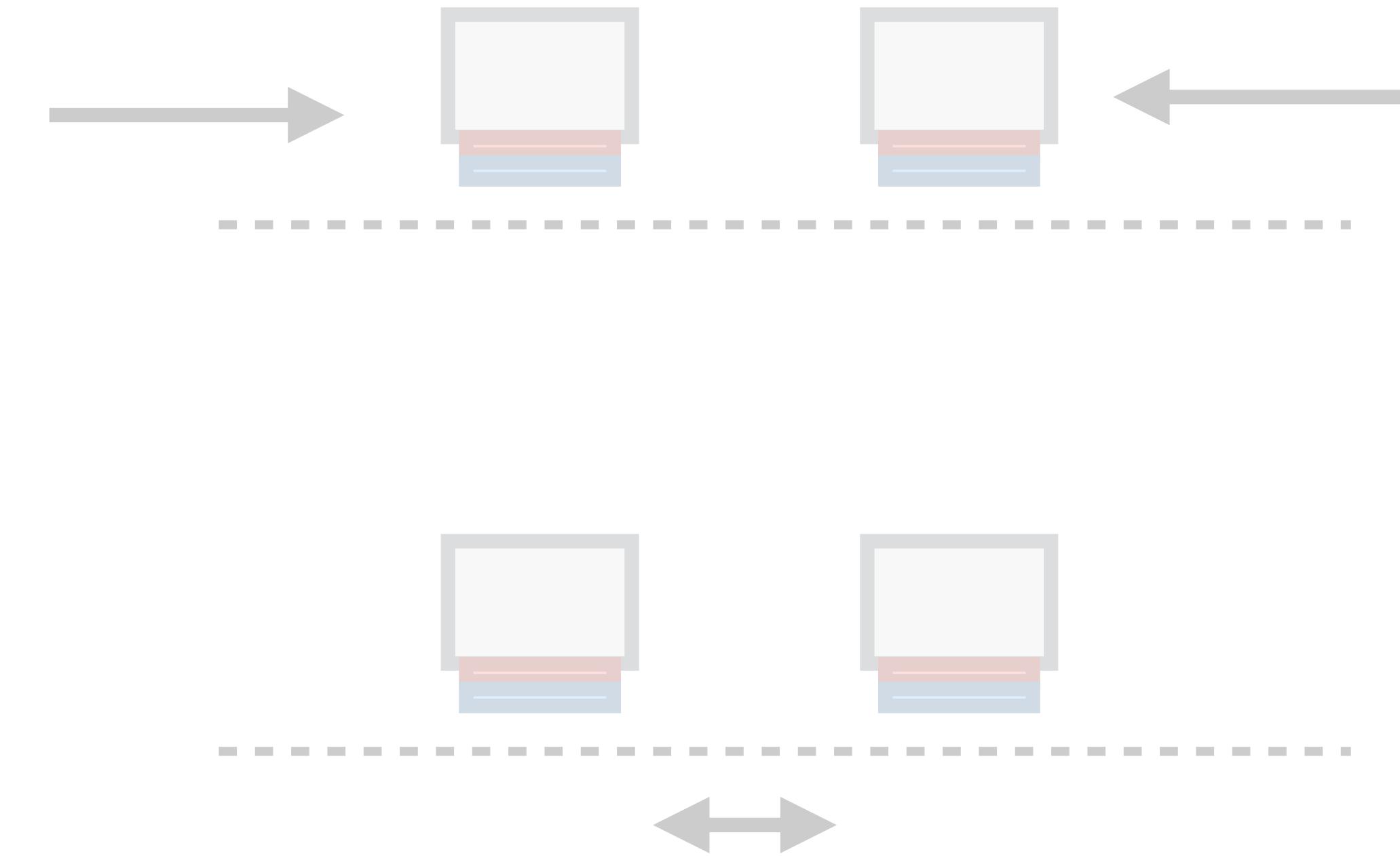
5. Limitations and Future Work

Limitations

Slow refresh rate

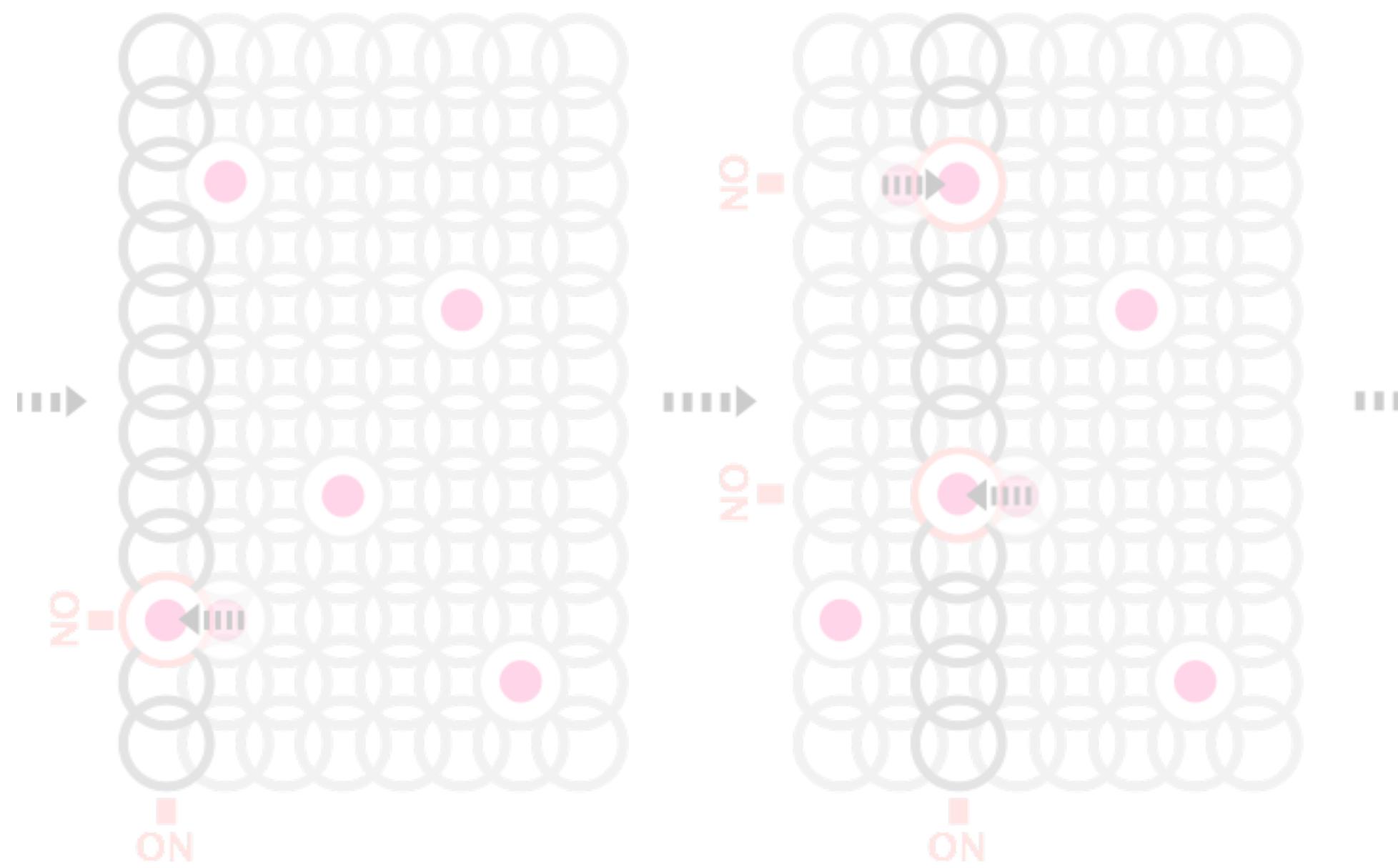


Minimum distance

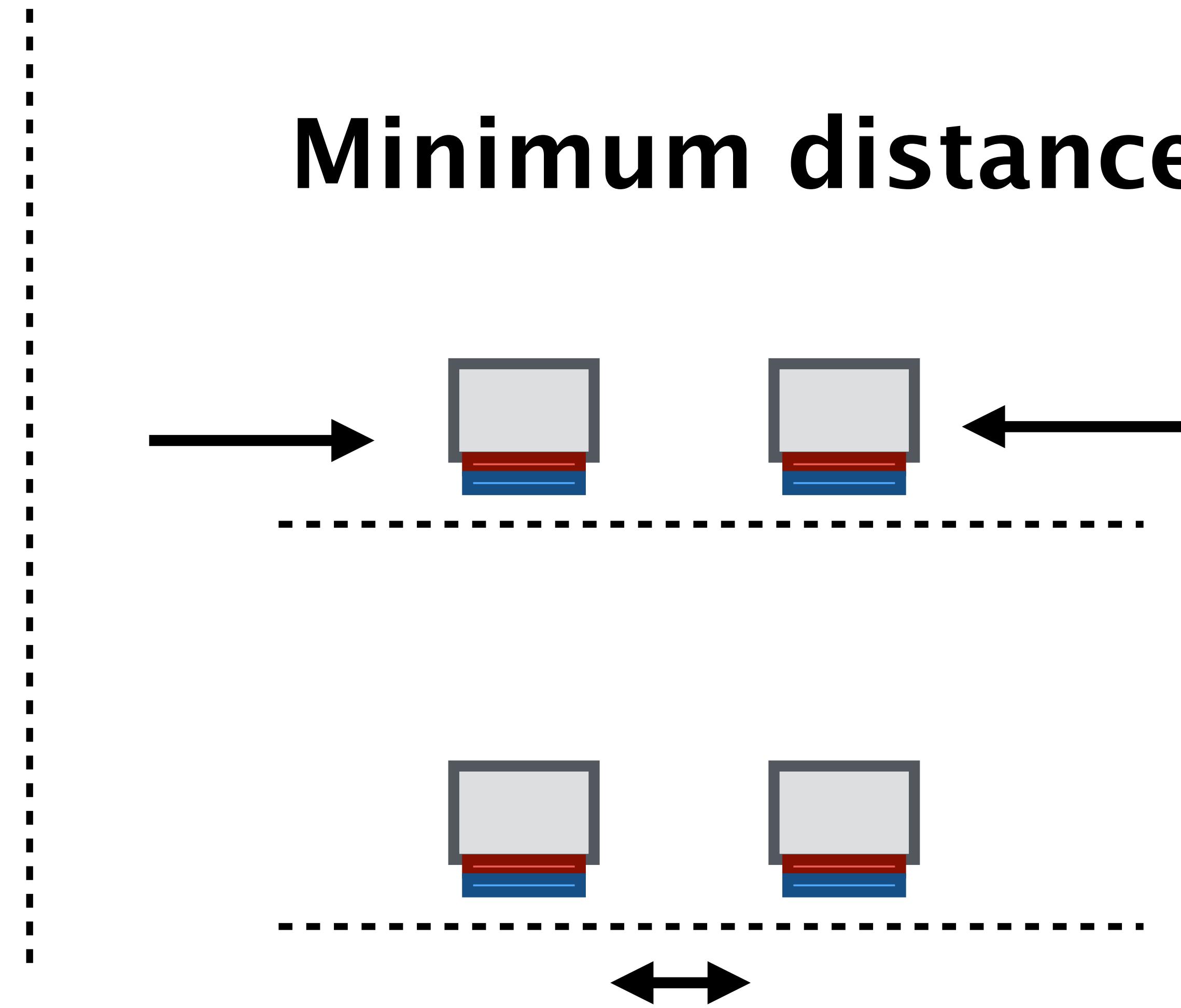


Limitations

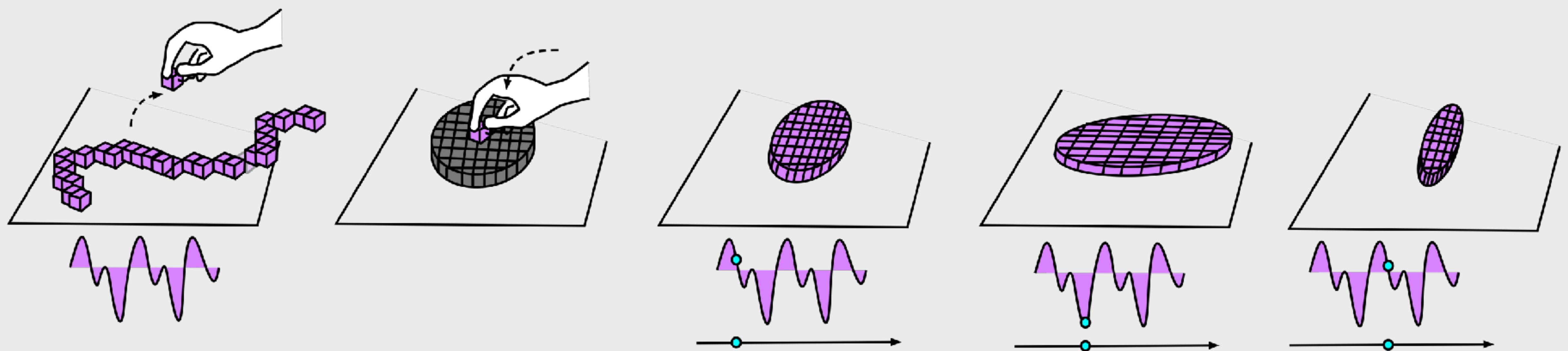
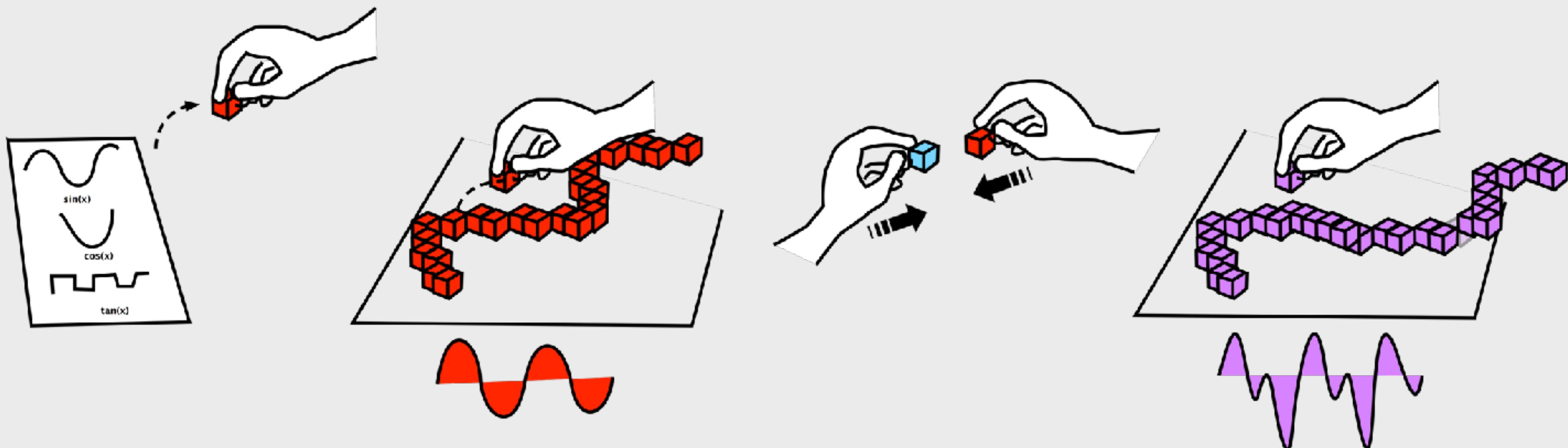
Slow refresh rate



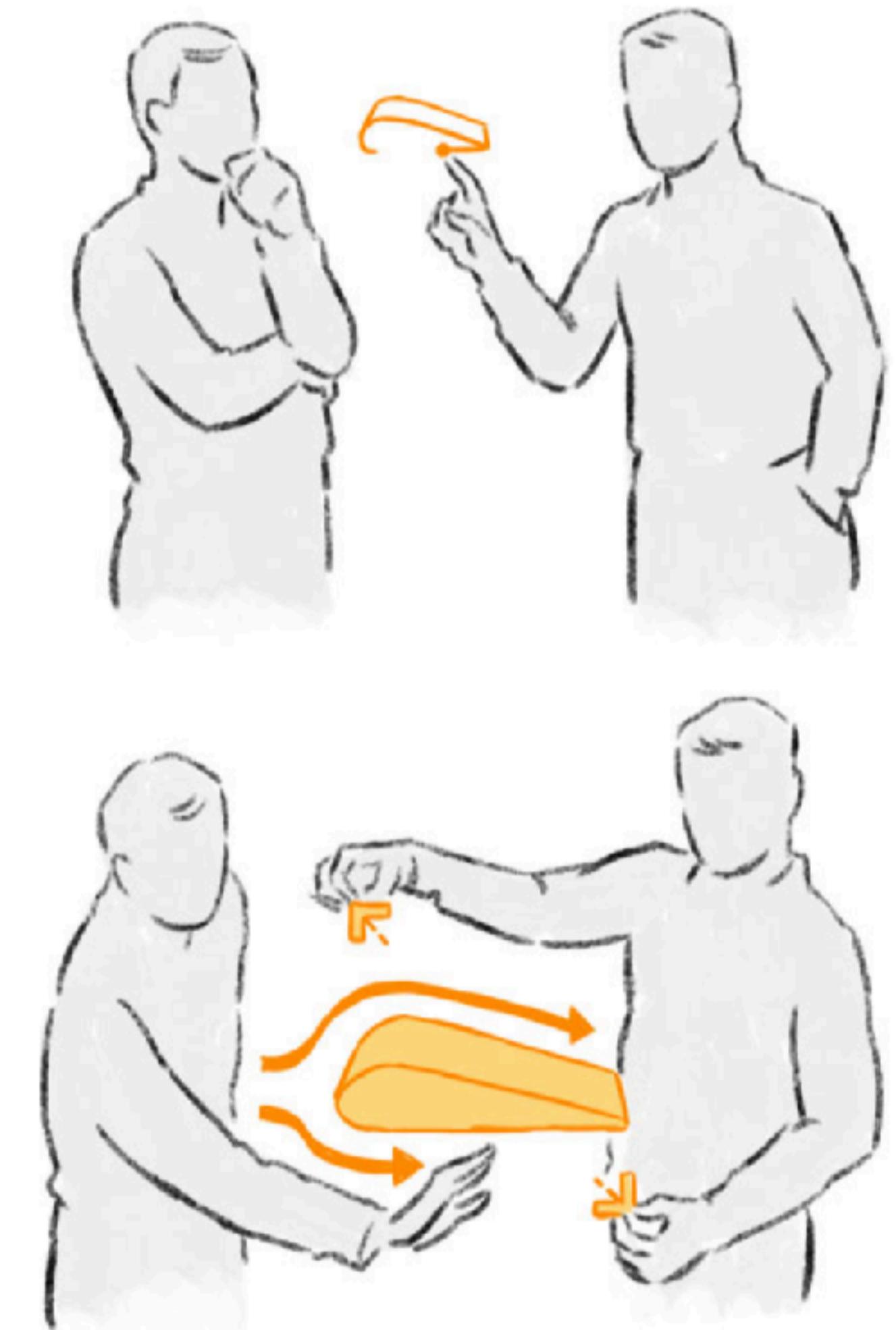
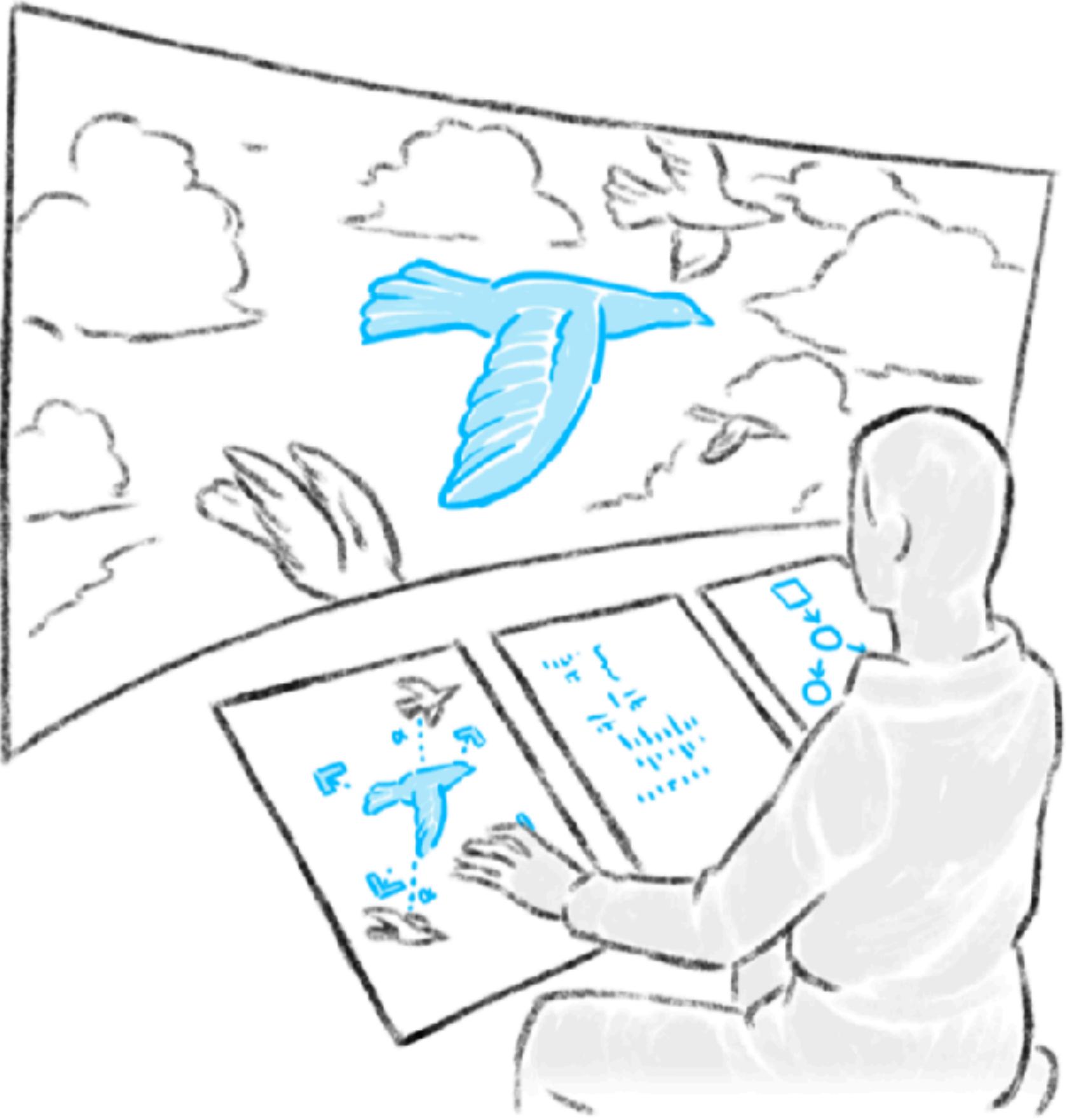
Minimum distance



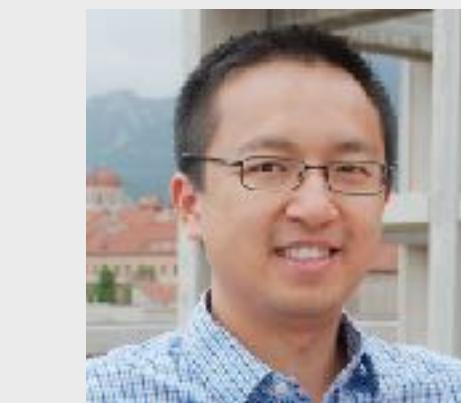
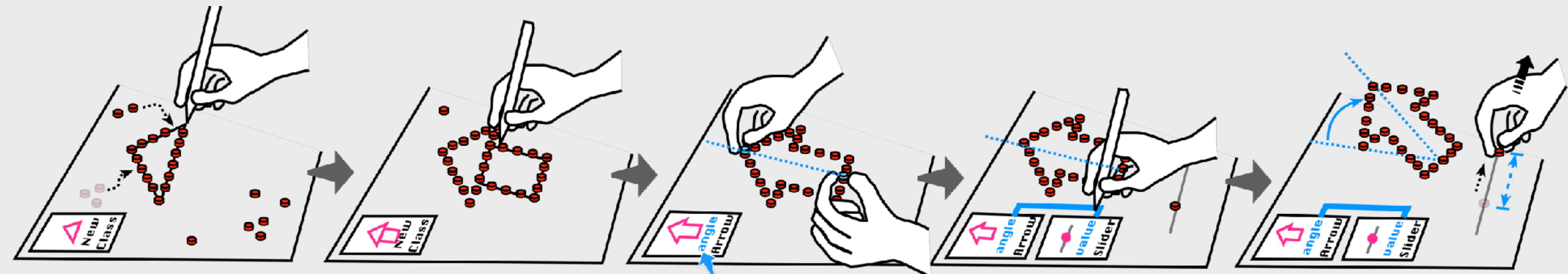
Future Work







Reactile: Programming Swarm User Interfaces through Direct Physical Manipulation



Ryo Suzuki, Jun Kato, Mark Gross, Tom Yeh