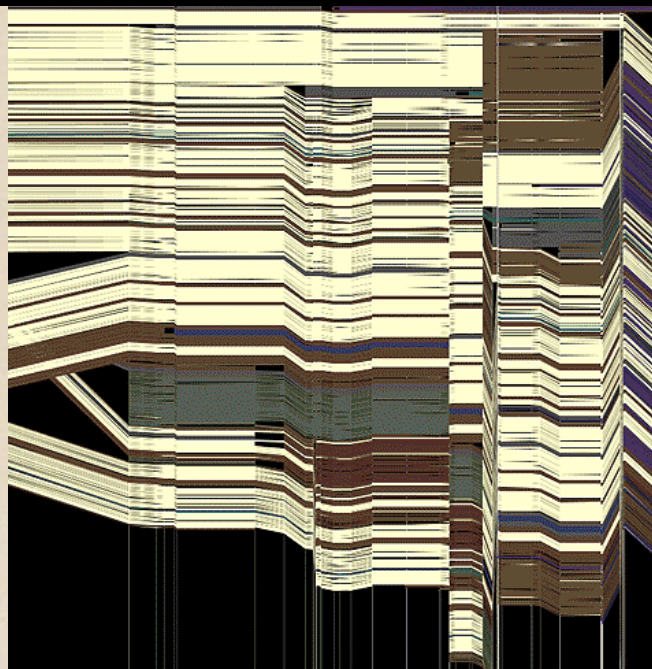
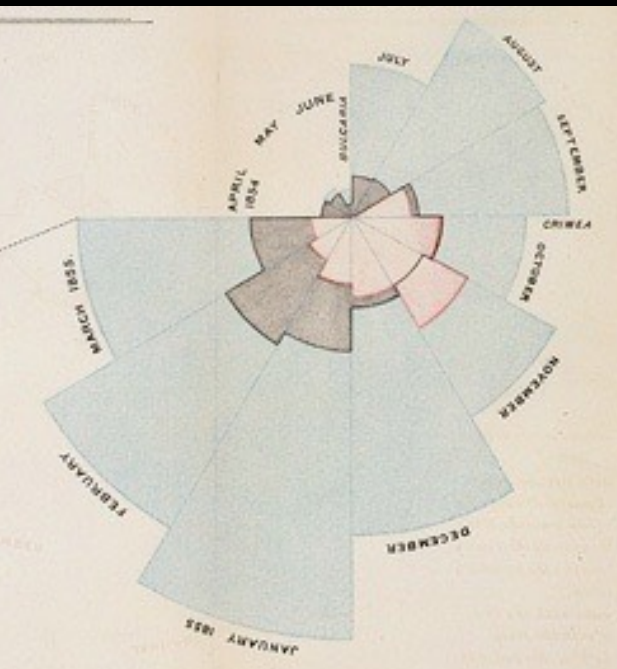


# CSE 512 - Data Visualization

# Animation



Jeffrey Heer University of Washington

# Why Use Motion?

Visual variable to encode data

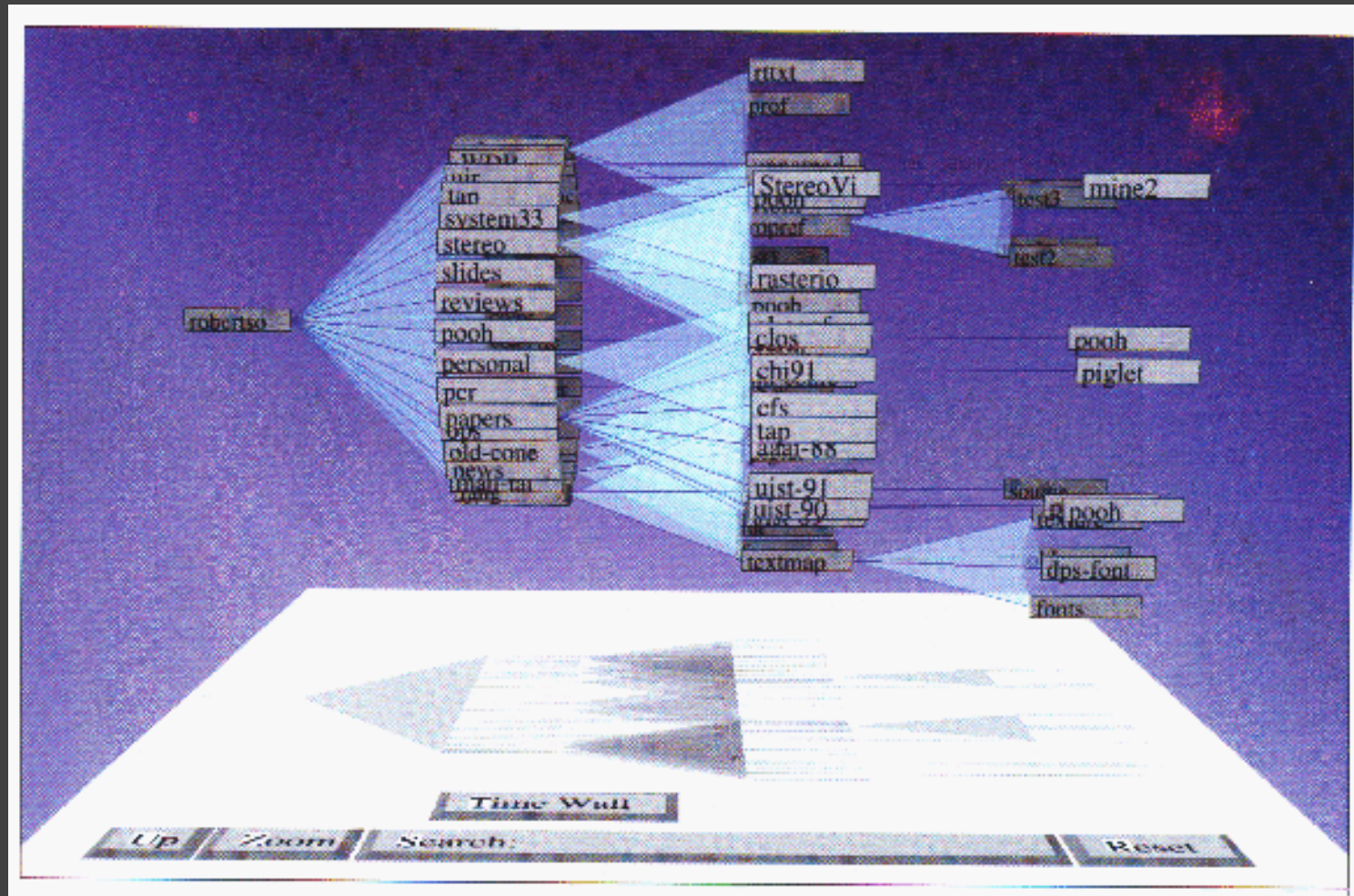
Direct attention

Understand system dynamics

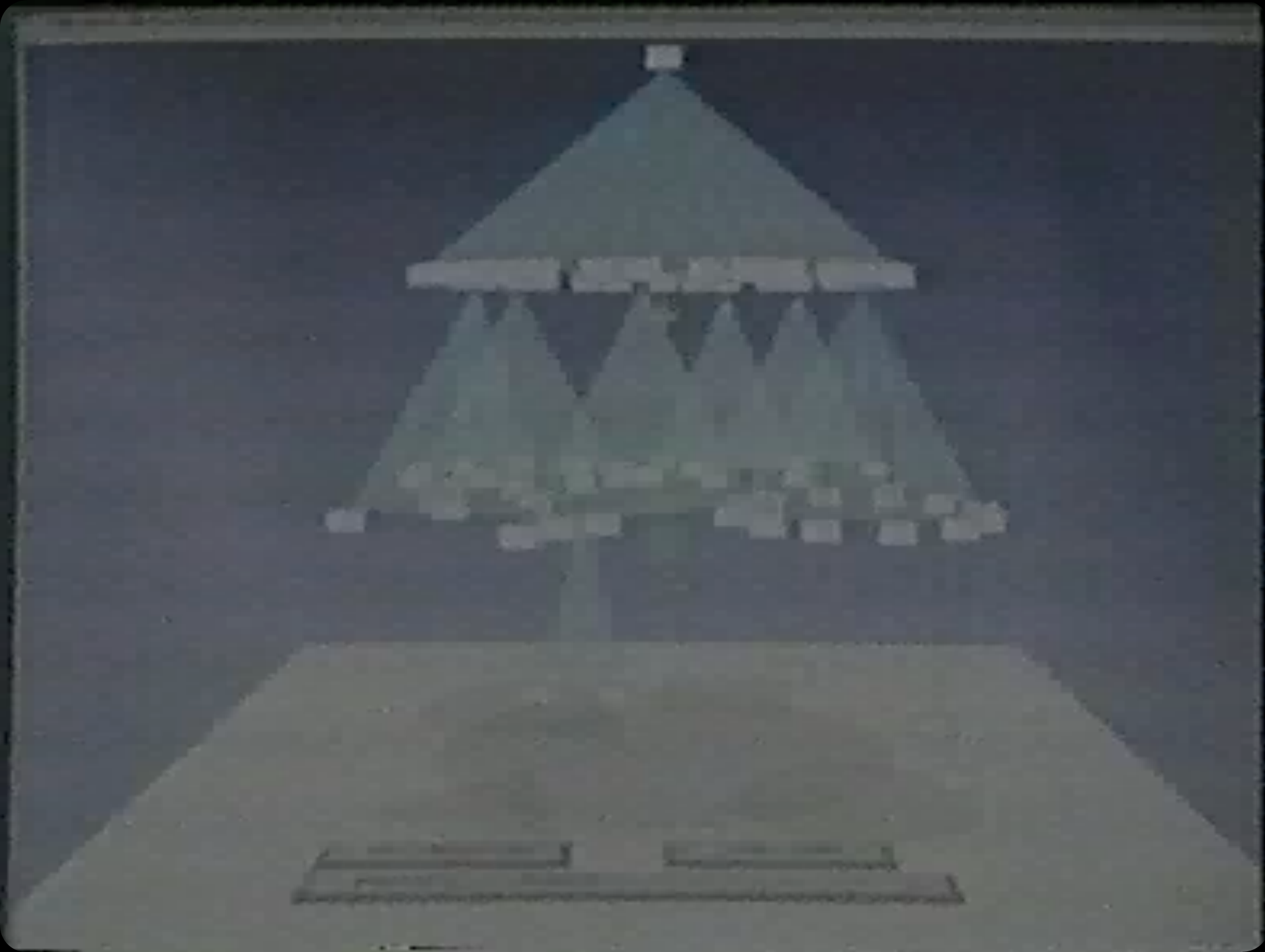
Understand state transition

Increase engagement

# Cone Trees [Robertson 91]

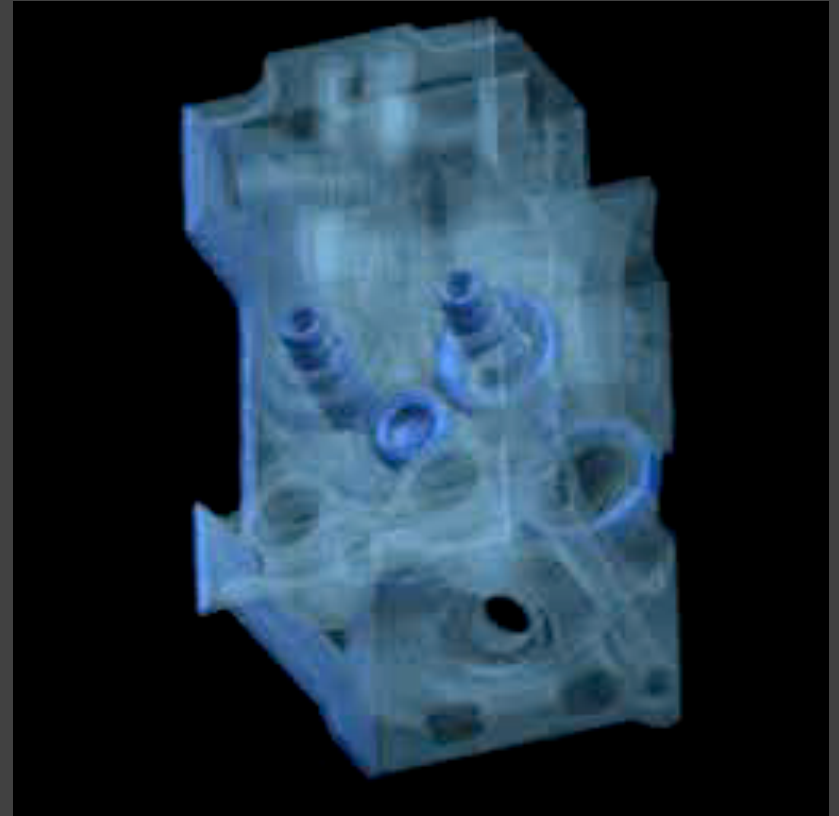


[Video](#)



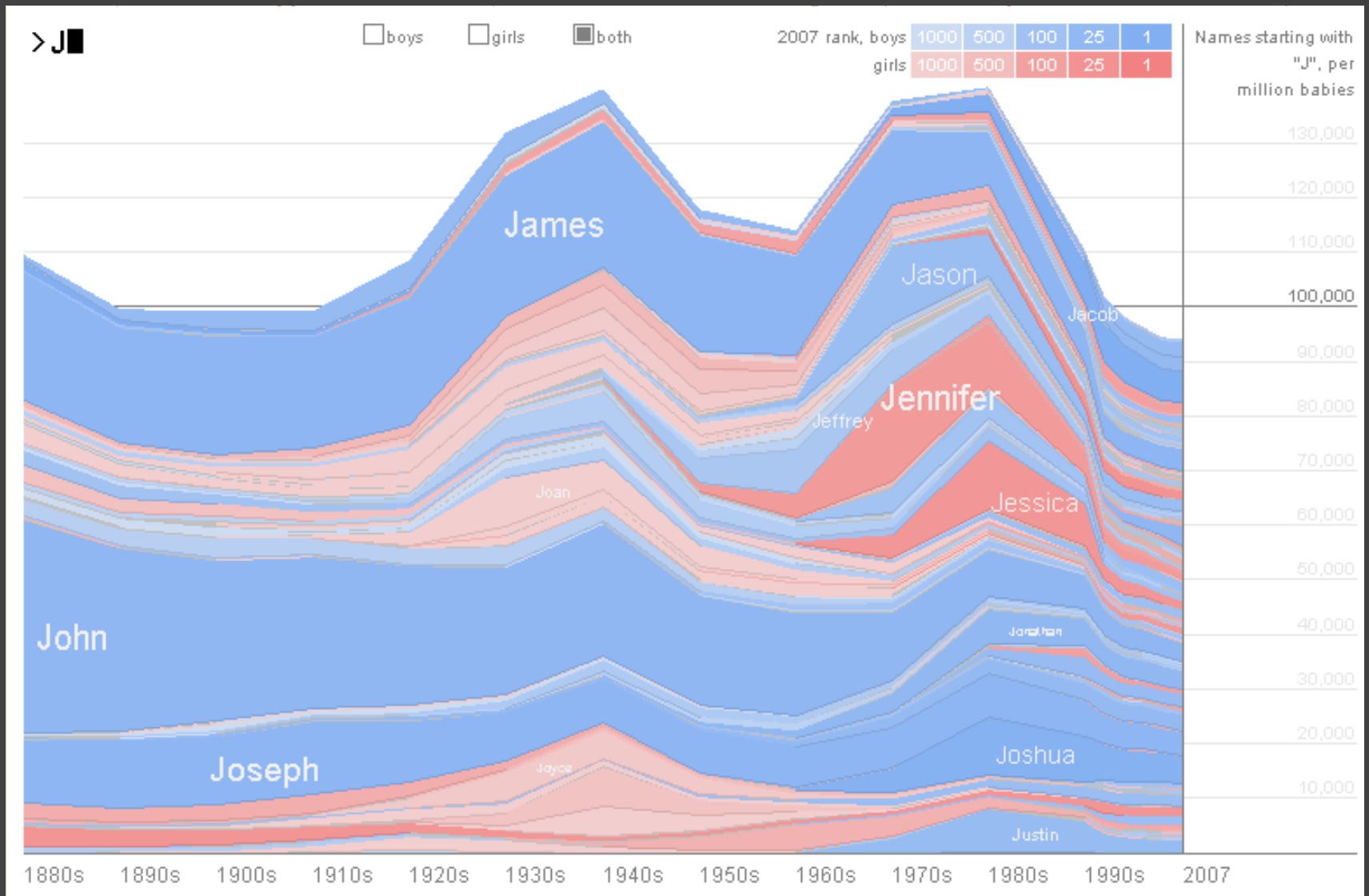


# Volume Rendering [Lacroute 95]



Video

# NameVoyager [Wattenberg 06]



# Topics

Motion perception

Animated transitions in visualizations

Implementing animations

# Motion Perception

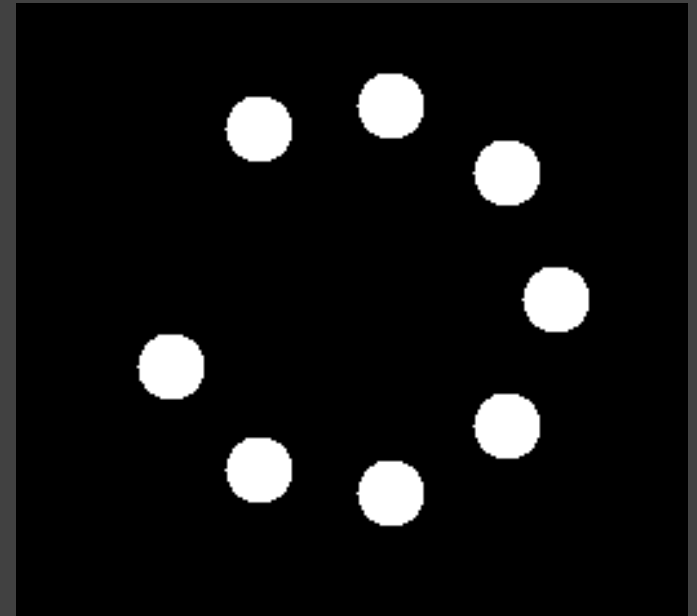


# Perceiving Animation

Under what conditions does a sequence of static images give rise to motion perception?

Motion is perceived at about  
~10 frames/sec (100 ms).

But this does not have to be  
*smooth* motion! We can tell  
that frames are discrete yet  
still perceive movement.



# Motion as Visual Cue

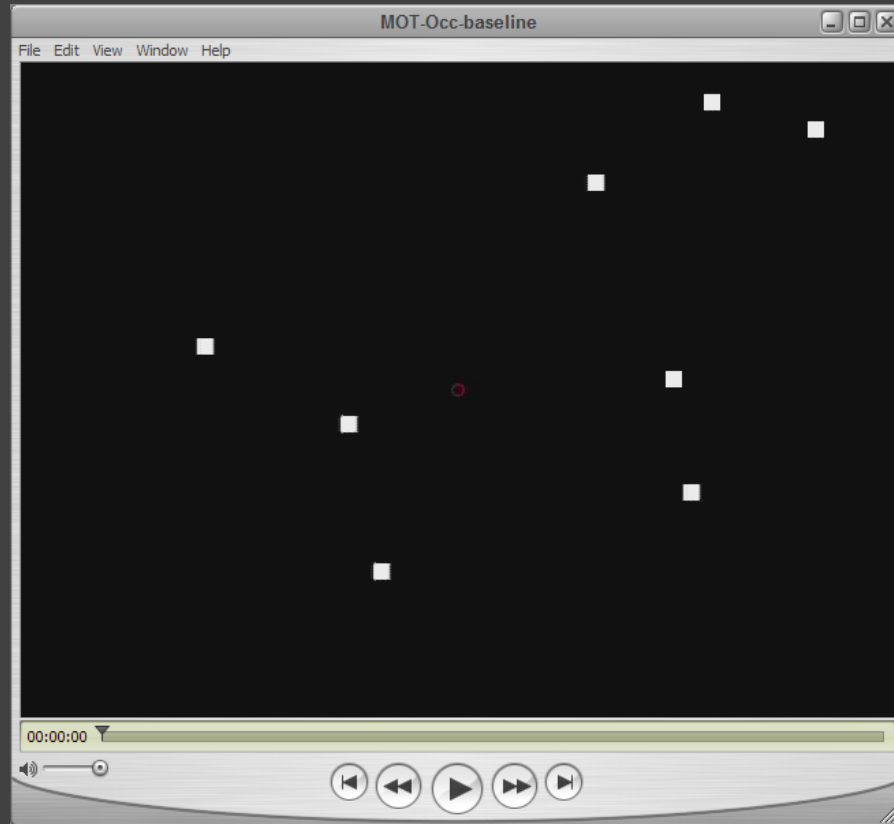
Pre-attentive, stronger than color, shape, ...

More sensitive to motion at periphery

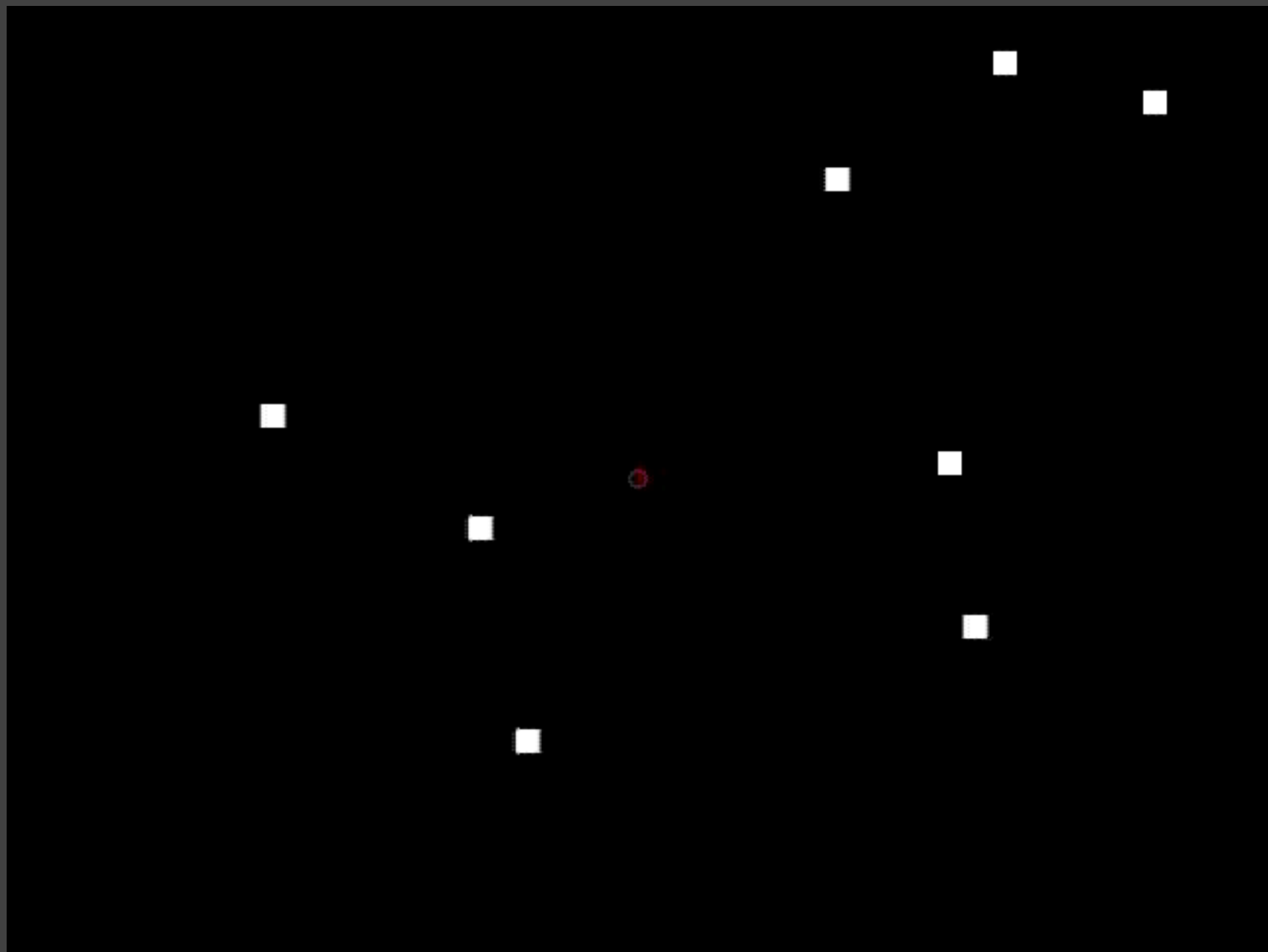
Similar motions perceived as a group

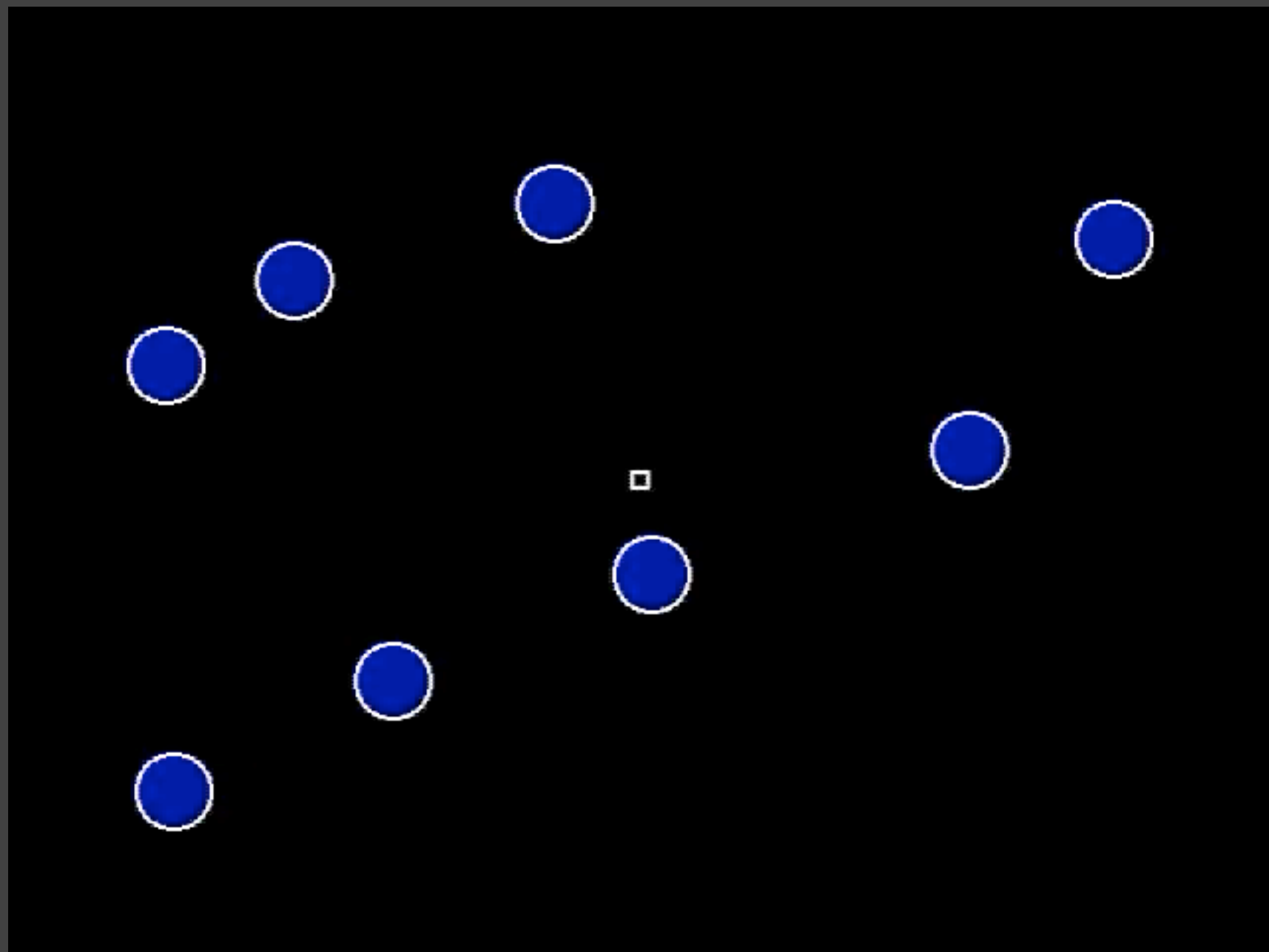
Motion parallax provide 3D cue (like stereopsis)

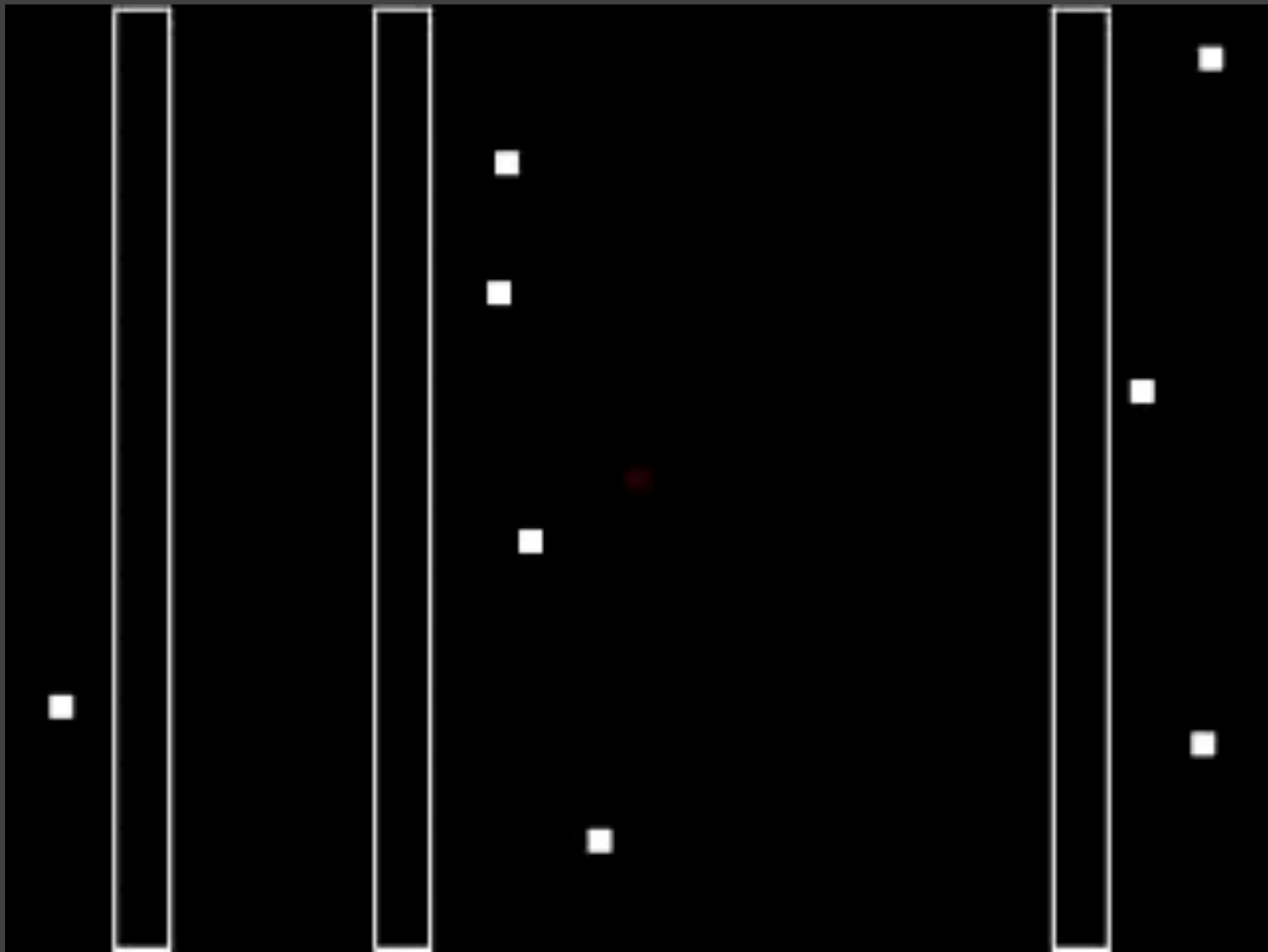
# Tracking Multiple Targets



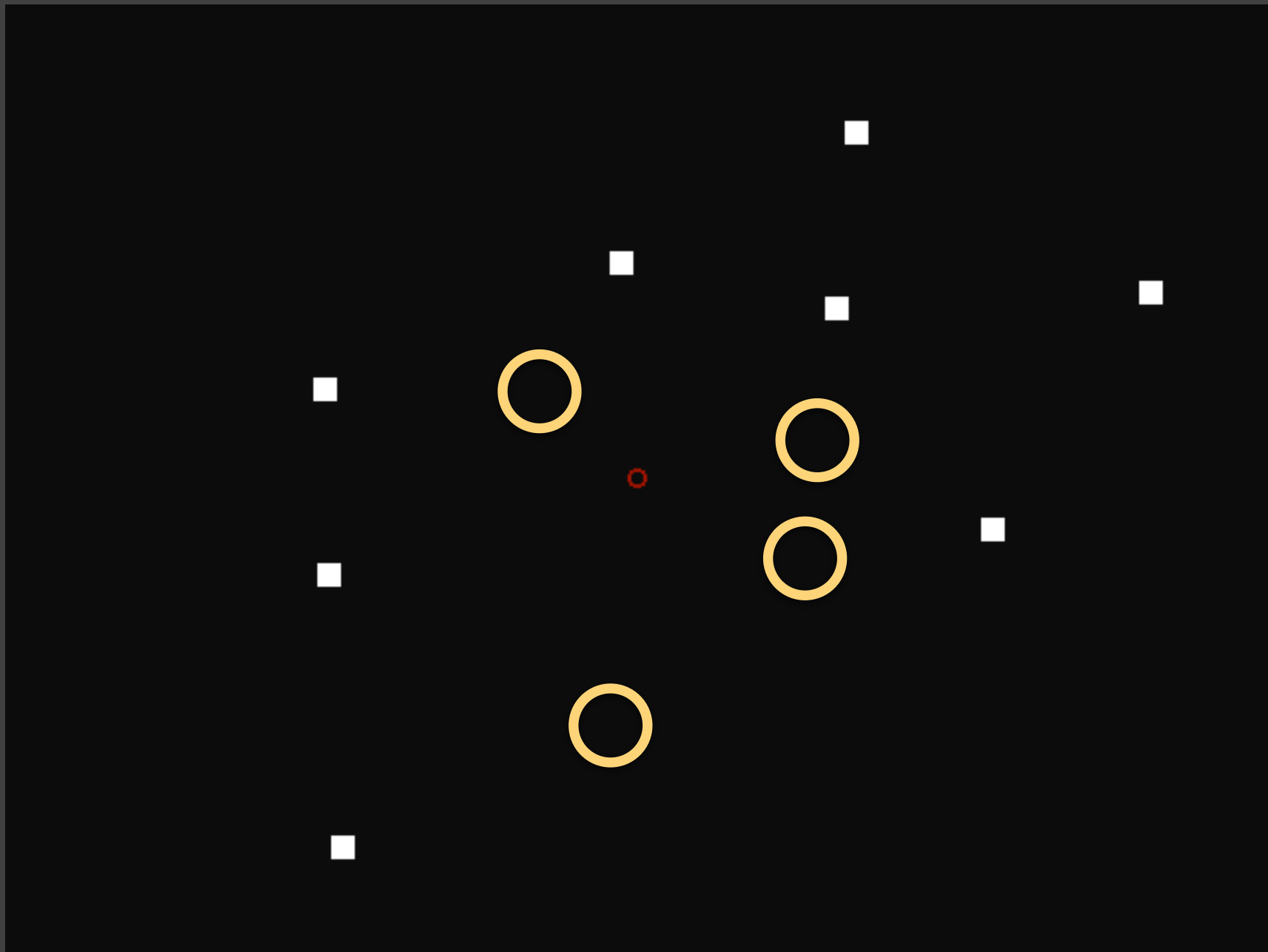
How many dots can we simultaneously track?

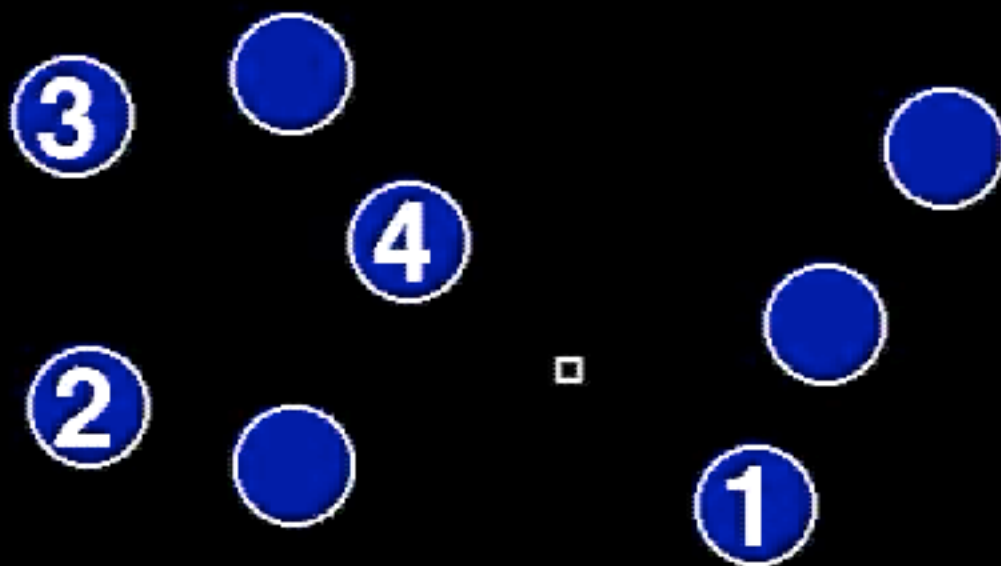




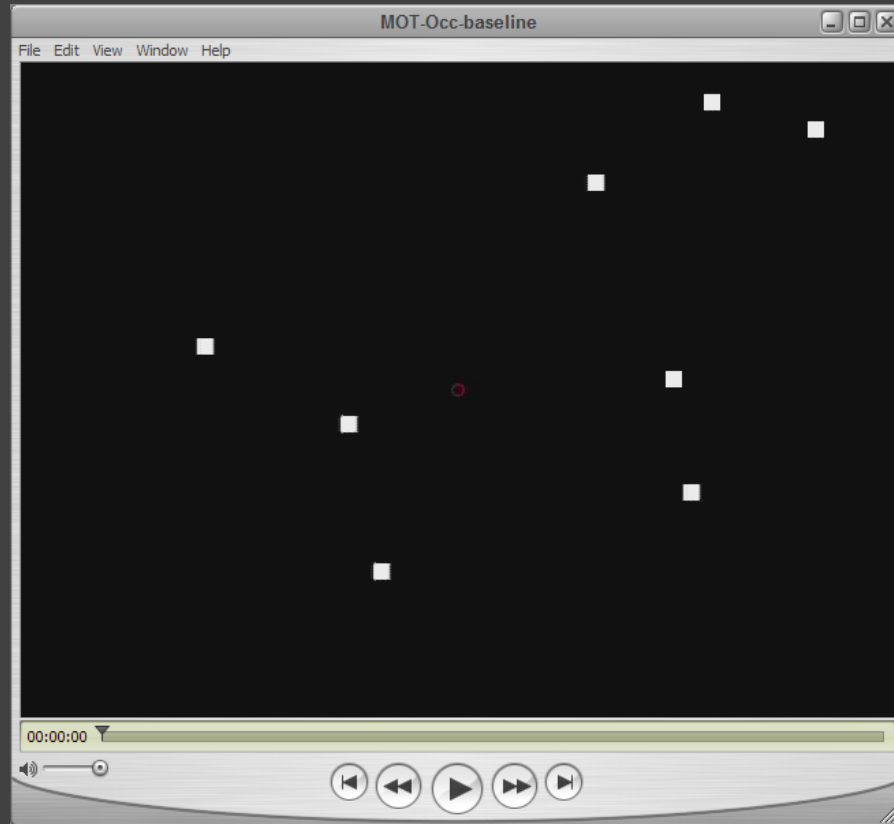








# Tracking Multiple Targets



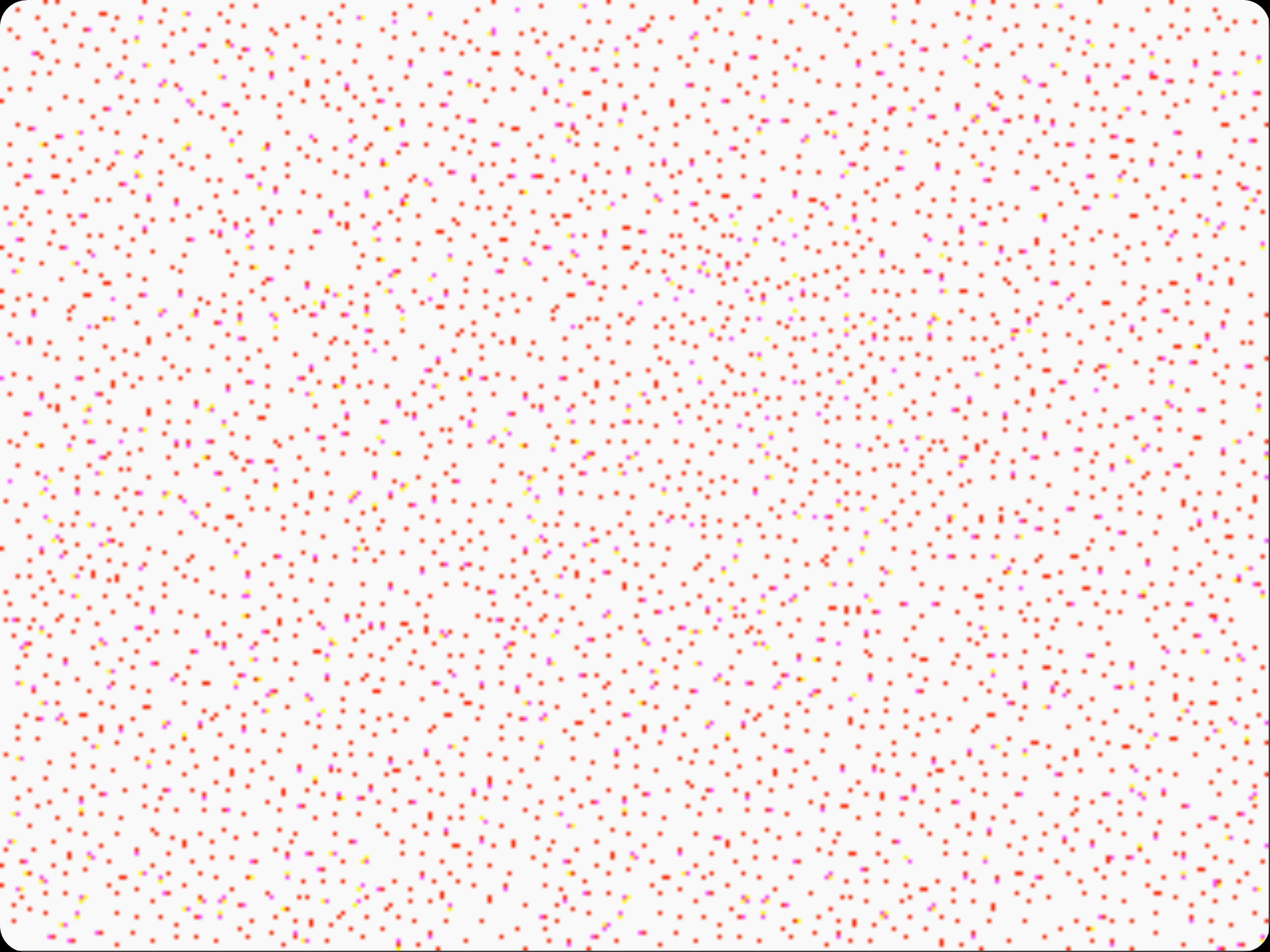
How many dots can we simultaneously track?

~4-6. Difficulty increases sig. at 6. [Yantis 92, Pylyshn 88, Cavanagh 05]

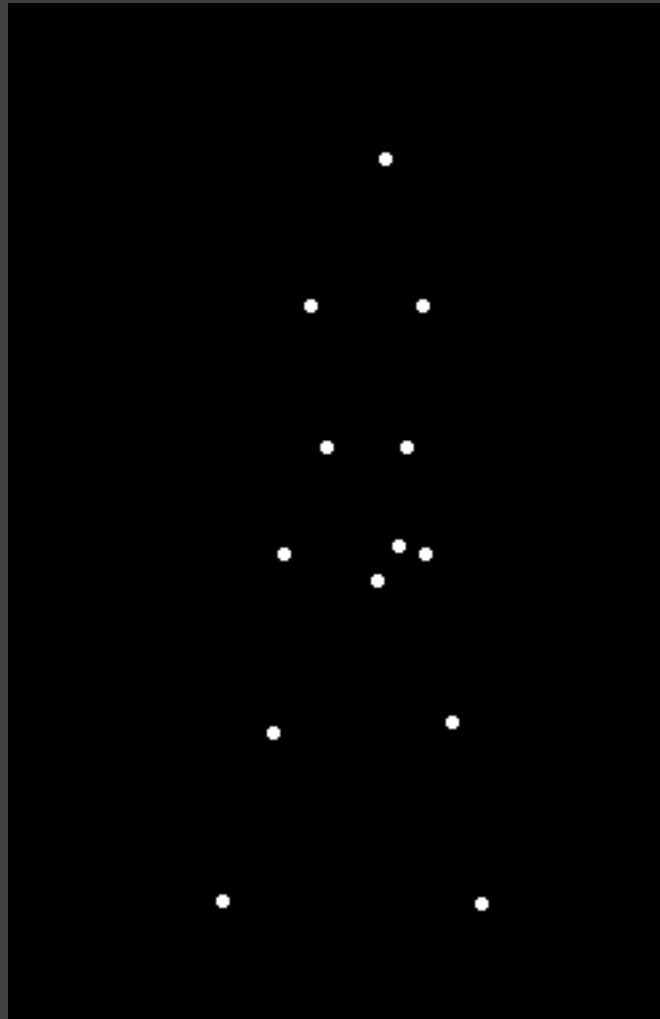
# Grouped Dots Count as 1 Object



Dots moving together are grouped



# Grouping of Biological Motion



[Johansson 73]

[http://www.lifesci.sussex.ac.uk/home/George\\_Mather/Motion/WALK.MOV](http://www.lifesci.sussex.ac.uk/home/George_Mather/Motion/WALK.MOV)



# Motions Show Transitions

See change from one state to next



start

# Motions Show Transitions

See change from one state to next



end

# Motions Show Transitions

See change from one state to next



start

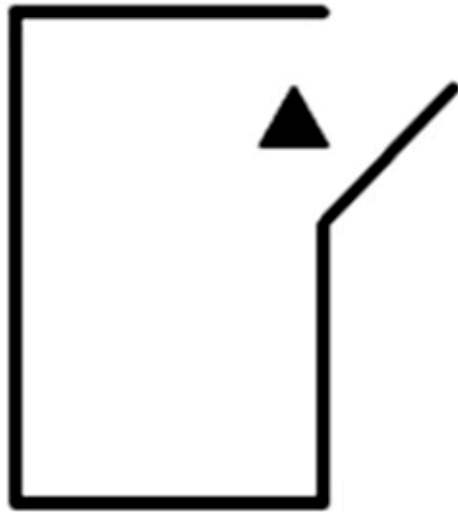
end

**Shows transition better, but**

Still may be too fast, or too slow

Too many objects may move at once

# Constructing Narratives [Heider 44]



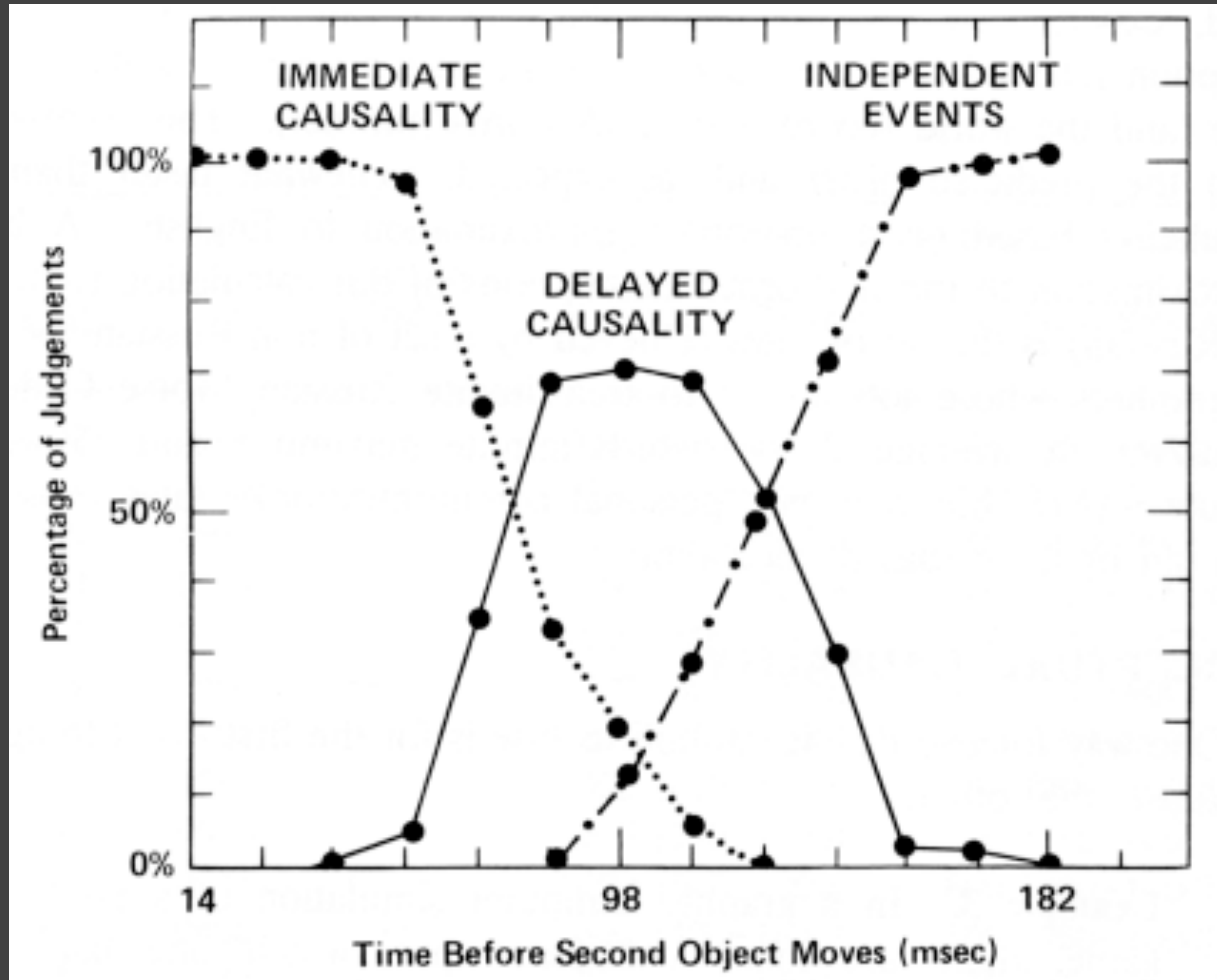
# Attribution of Causality [Michotte 46]

**Michotte demonstration 1.** What do you see? Most observers report that "the **red ball** hit the **blue ball**." The **blue ball** moved "**because** the **red ball** hit it." Thus, the **red ball** is perceived to "**cause**" the **blue ball** to move, even though the balls are nothing more than color disks on your screen that move according to a programme.



<http://cogweb.ucla.edu/Discourse/Narrative/michotte-demo.swf>

# Attribution of Causality [Michotte 46]



[Reprint from Ware 04]



# Animation

## Helps?

## Hurts?

*Attention*

direct attention

distraction

*Constancy*

change tracking

false relations

*Causality*

cause and effect

false agency

*Engagement*

increase interest

"chart junk"

*Calibration*

too slow: boring

too fast: errors



# Problems with Animation [Tversky]

Difficult to estimate paths and trajectories

Motion is fleeting and transient

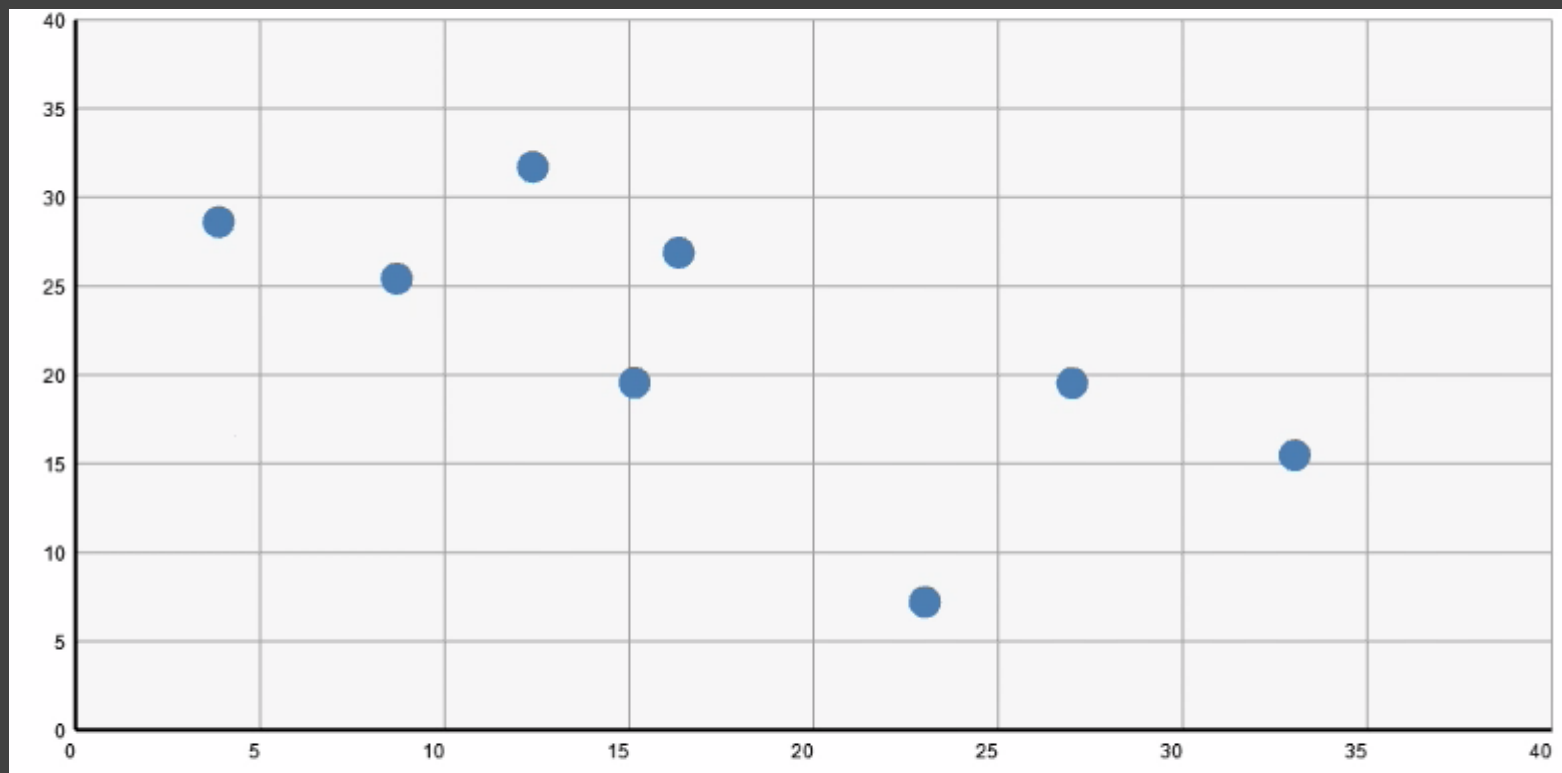
Cannot simultaneously attend to multiple motions

Parse motion into events, actions and behaviors

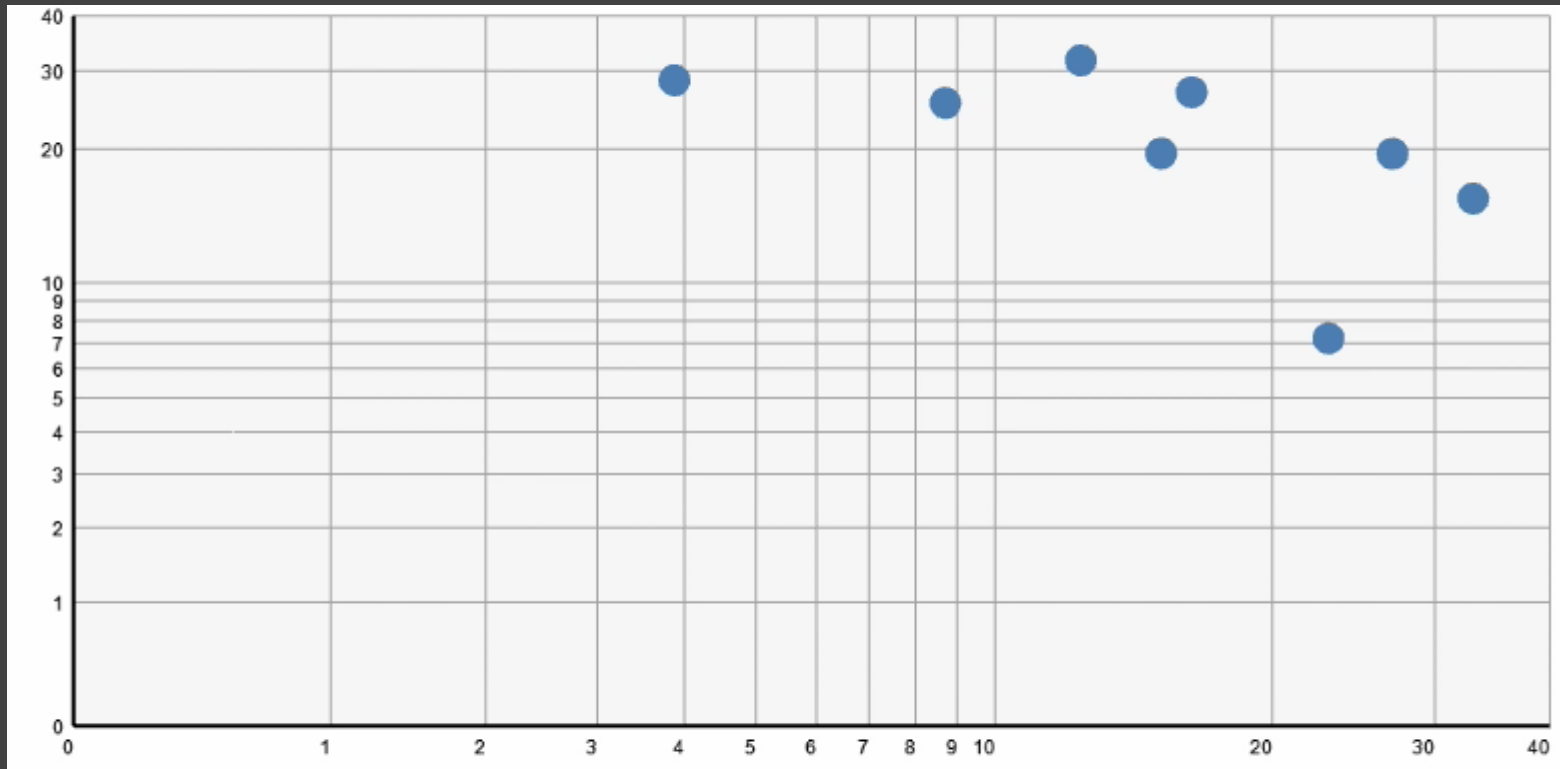
Misunderstanding and wrongly inferring causality

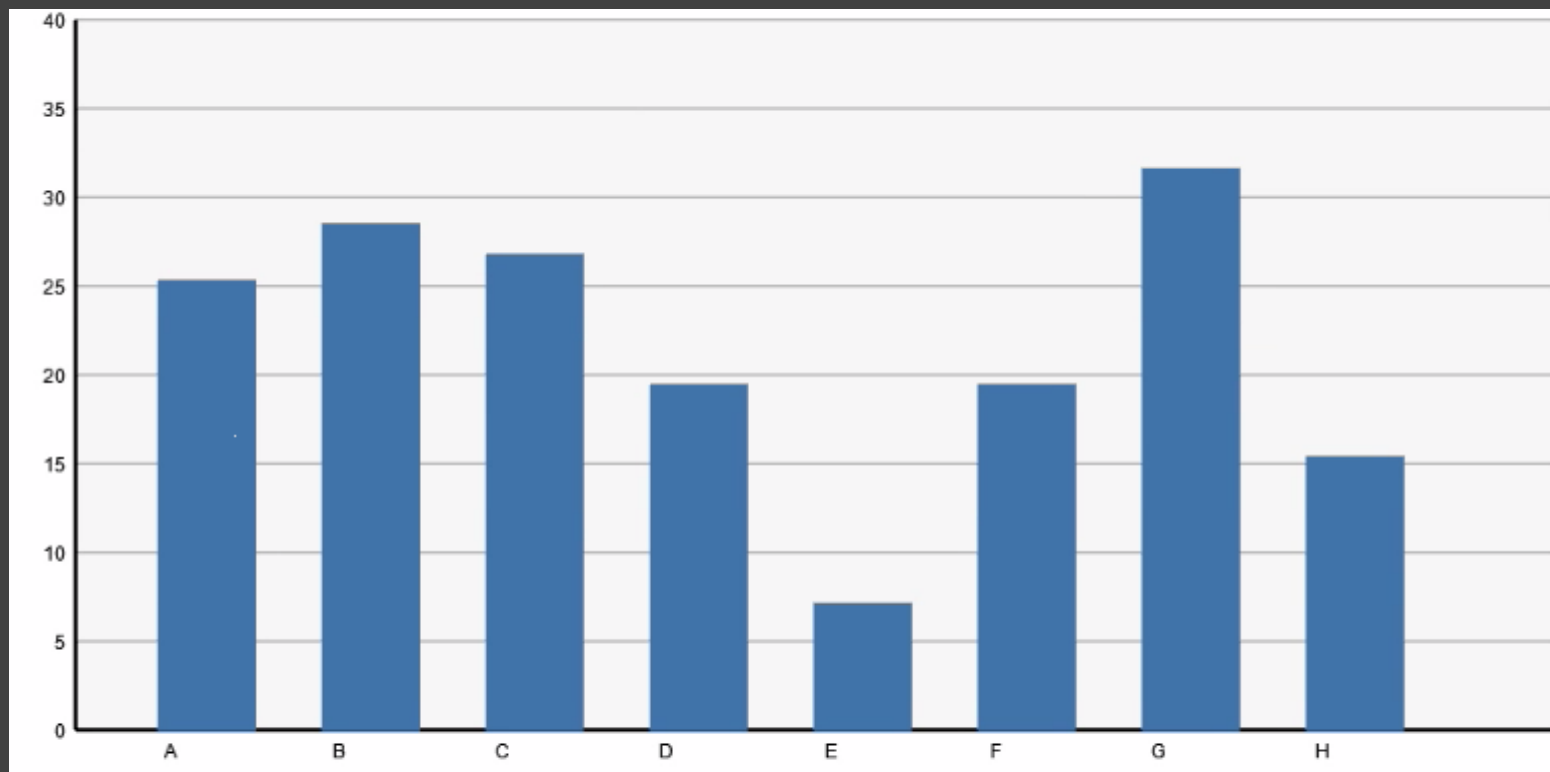
Anthropomorphizing physical motion may cause confusion or lead to incorrect conclusions

# Animated Transitions in Statistical Graphics



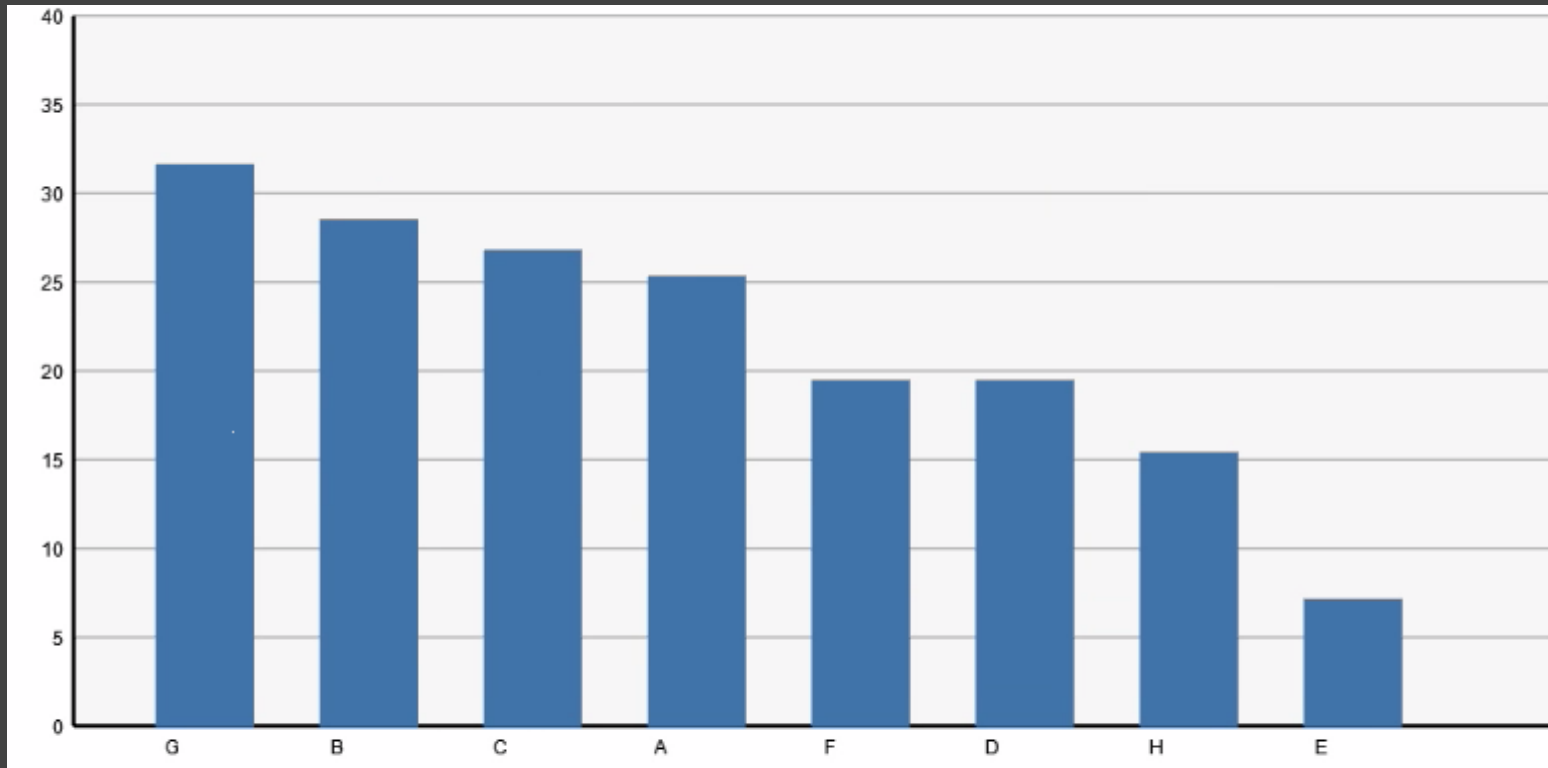
# Log Transform

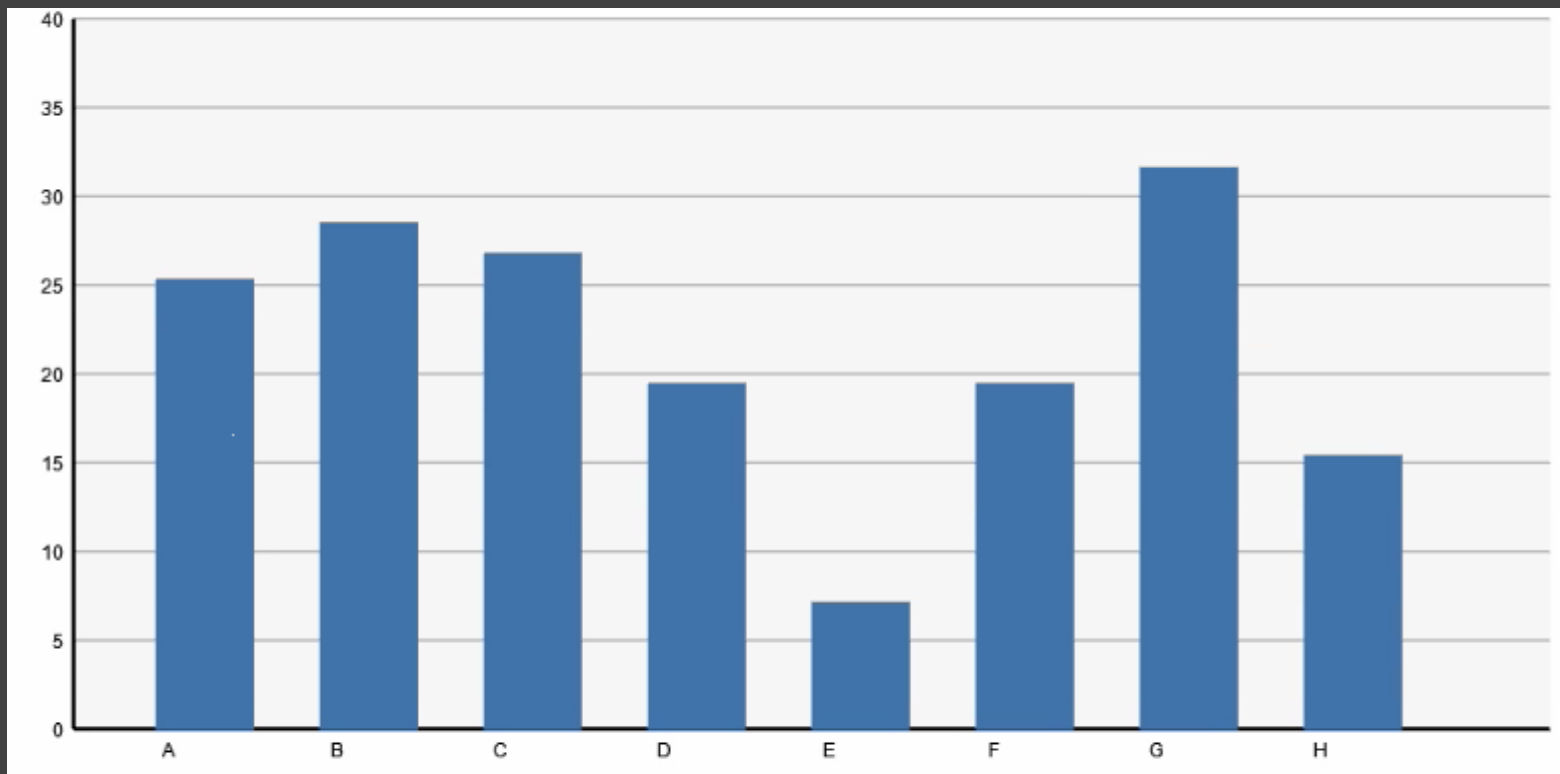




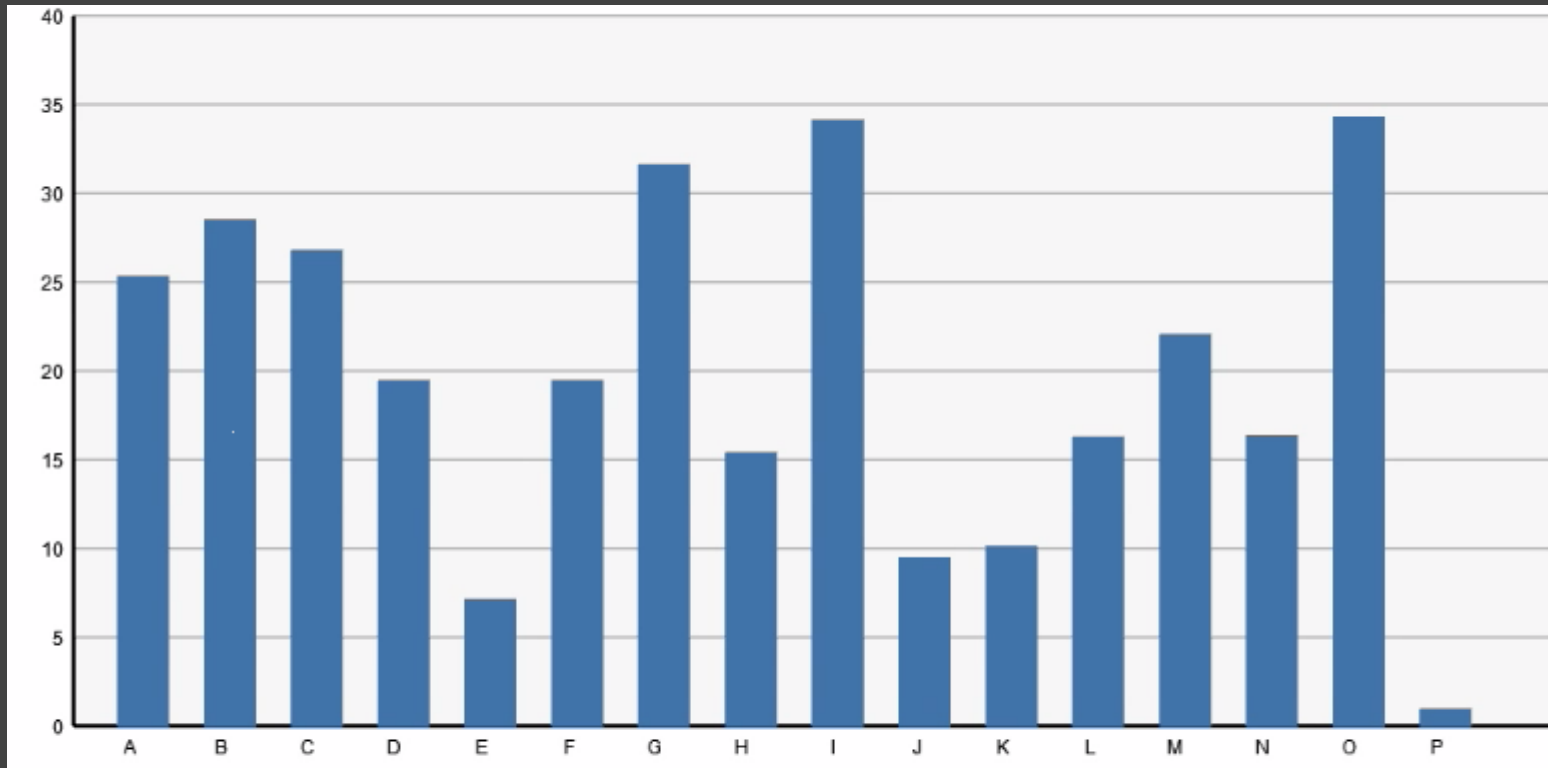


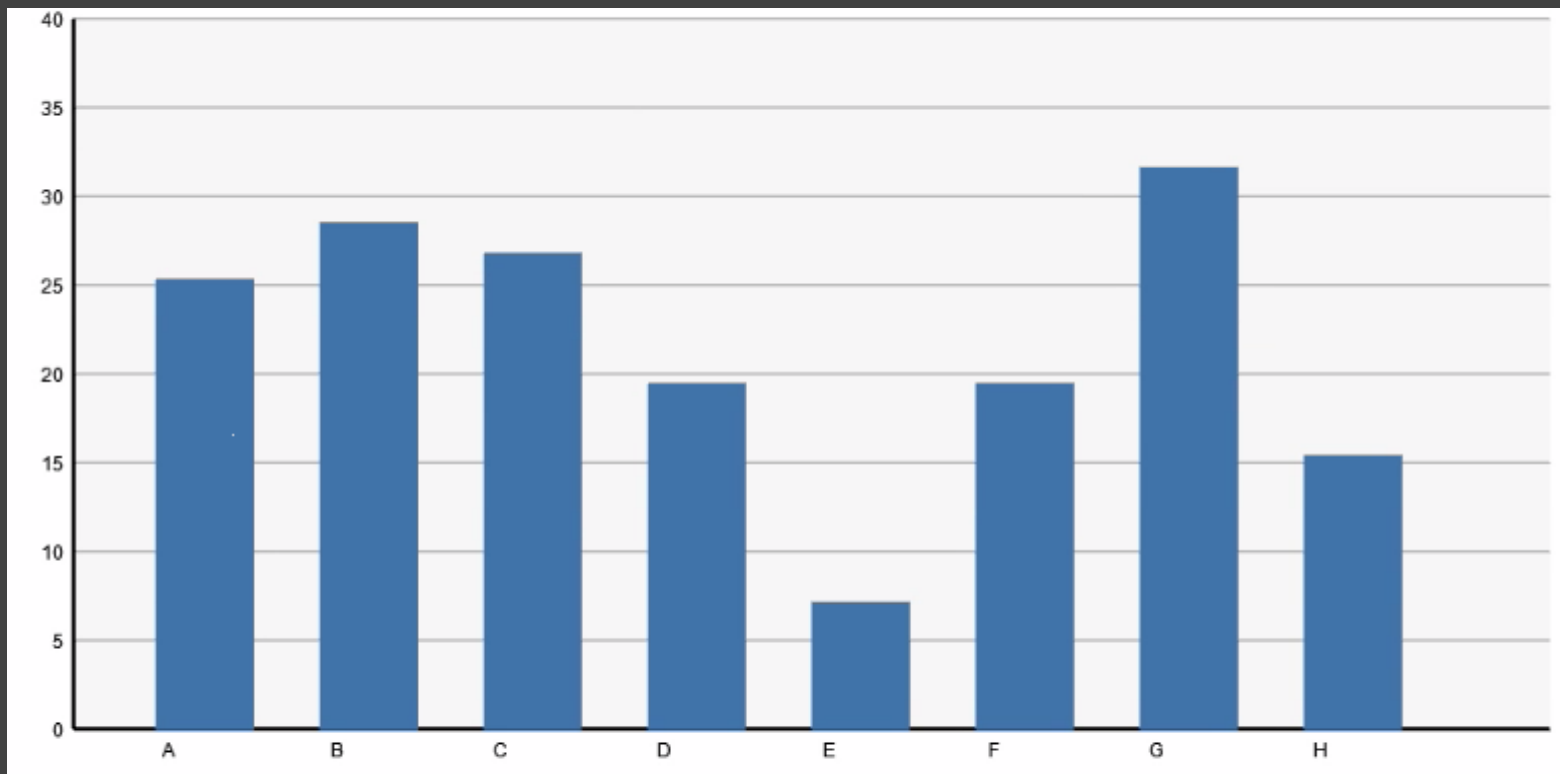
# Sorting

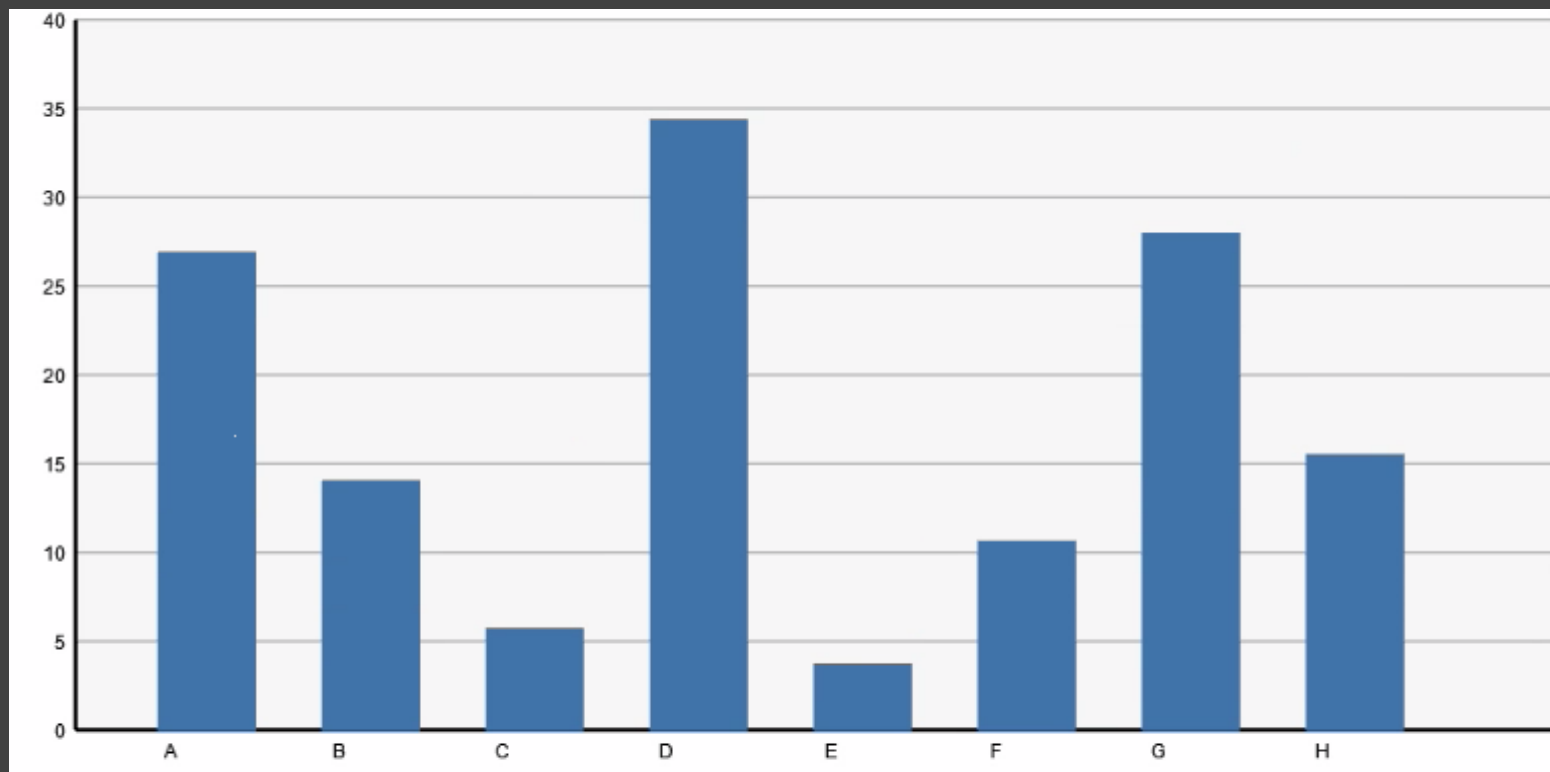


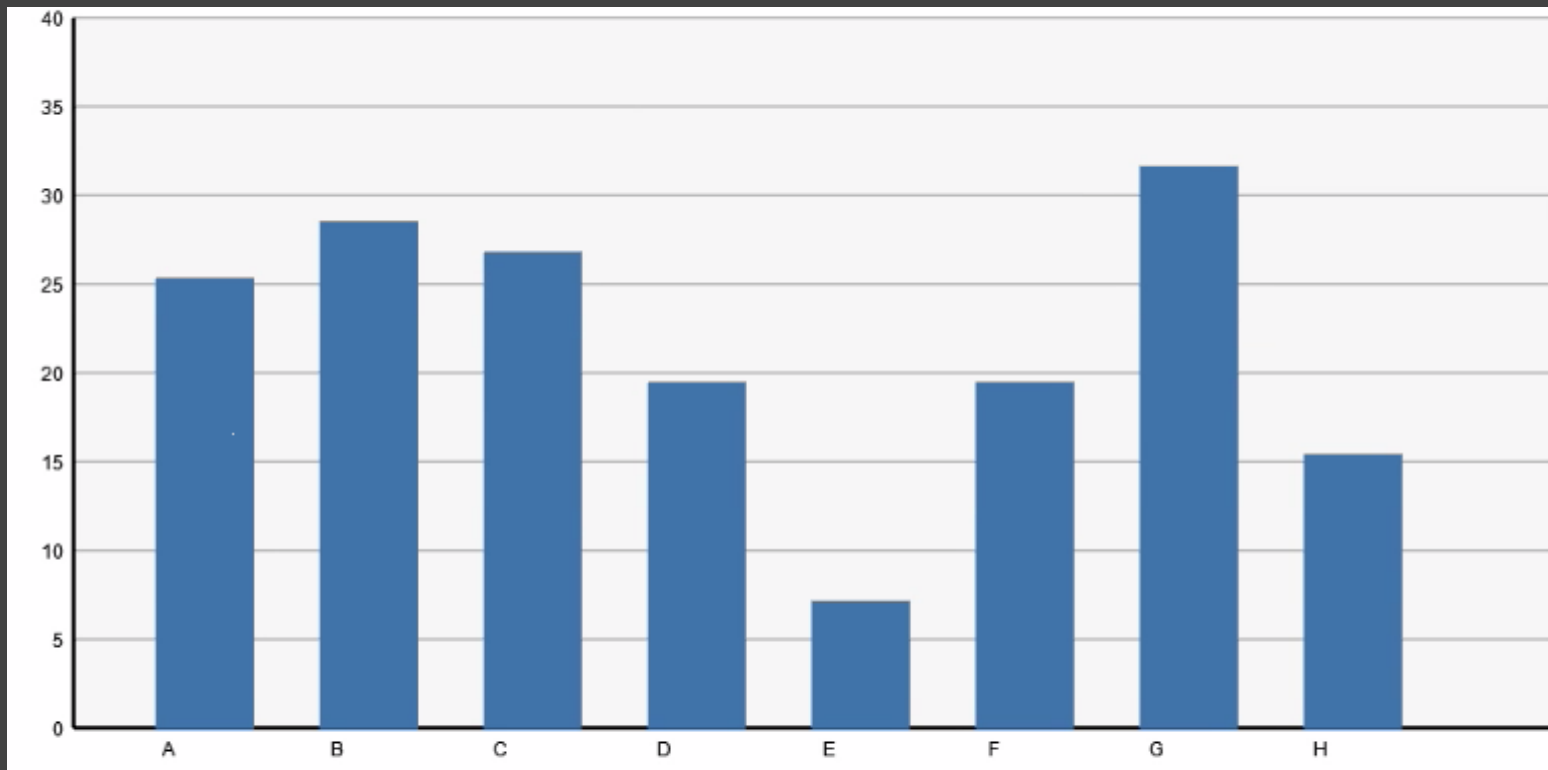


# Filtering



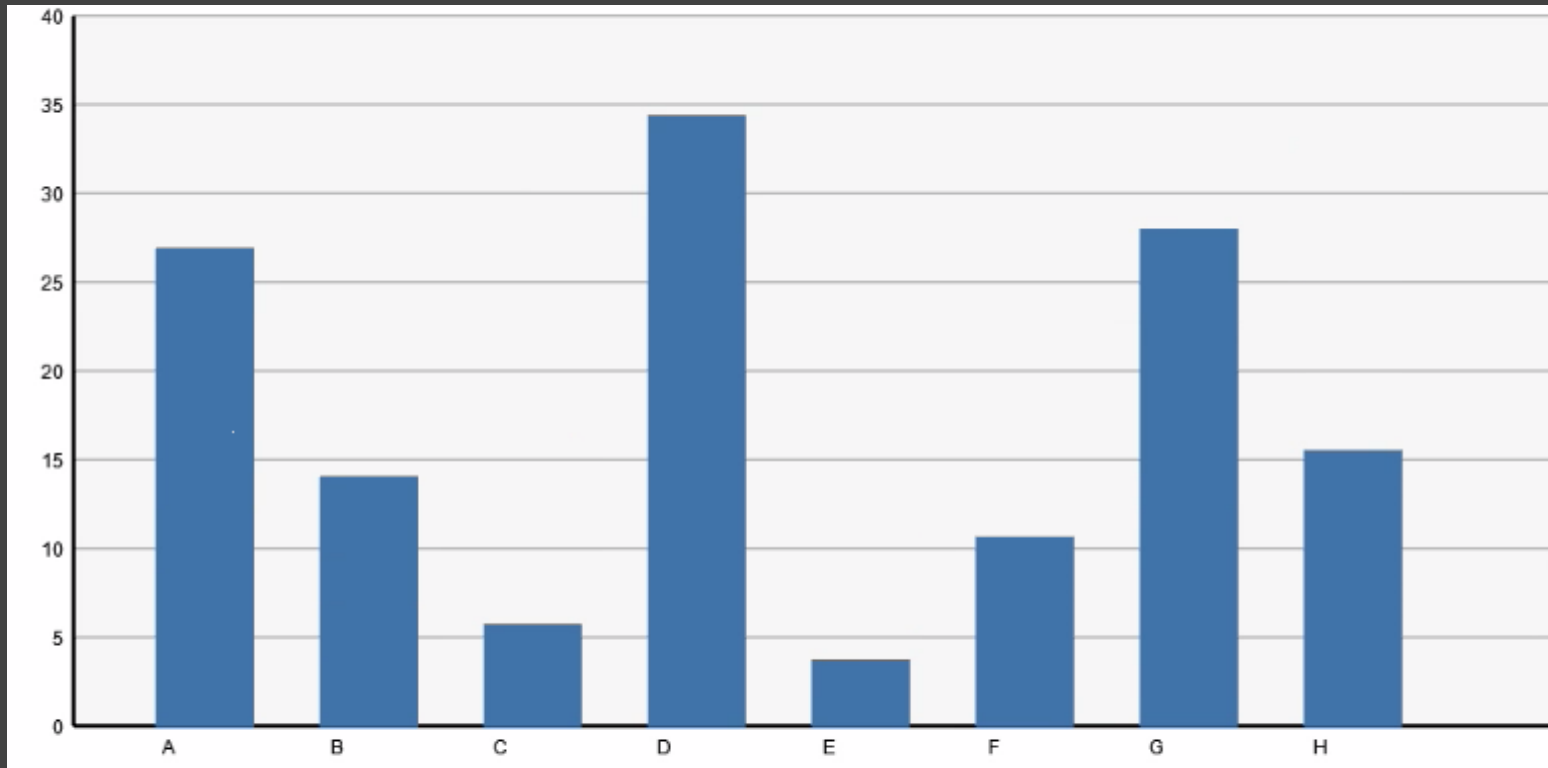




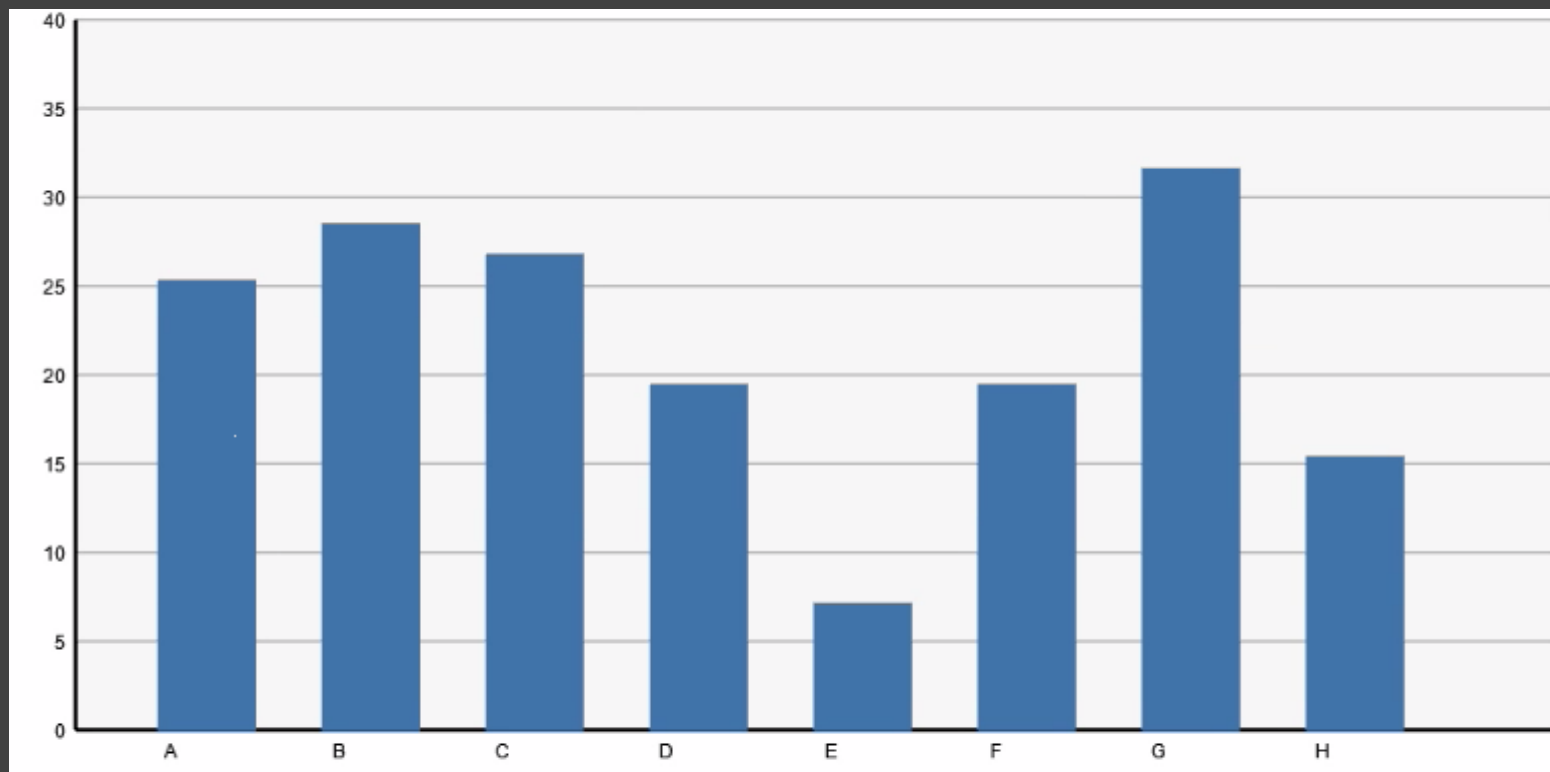


Month 1

# Timestep

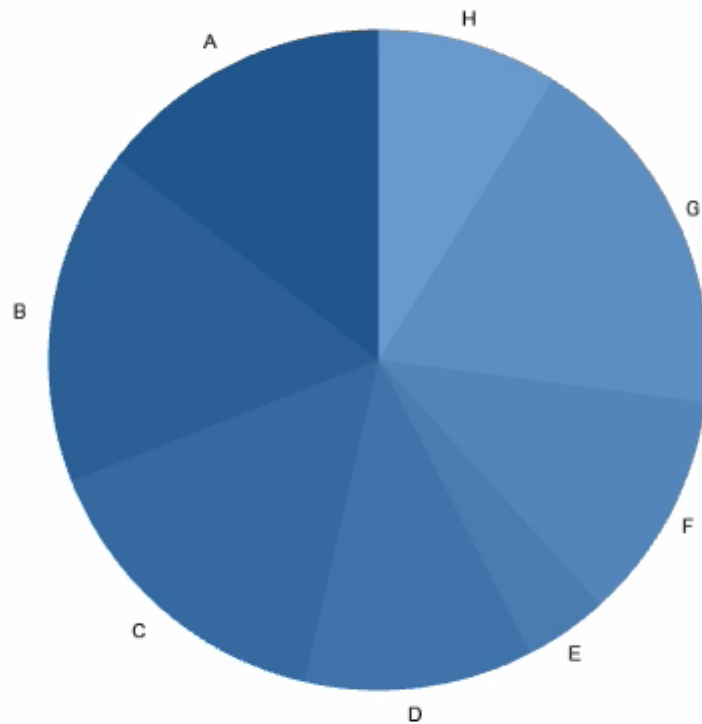


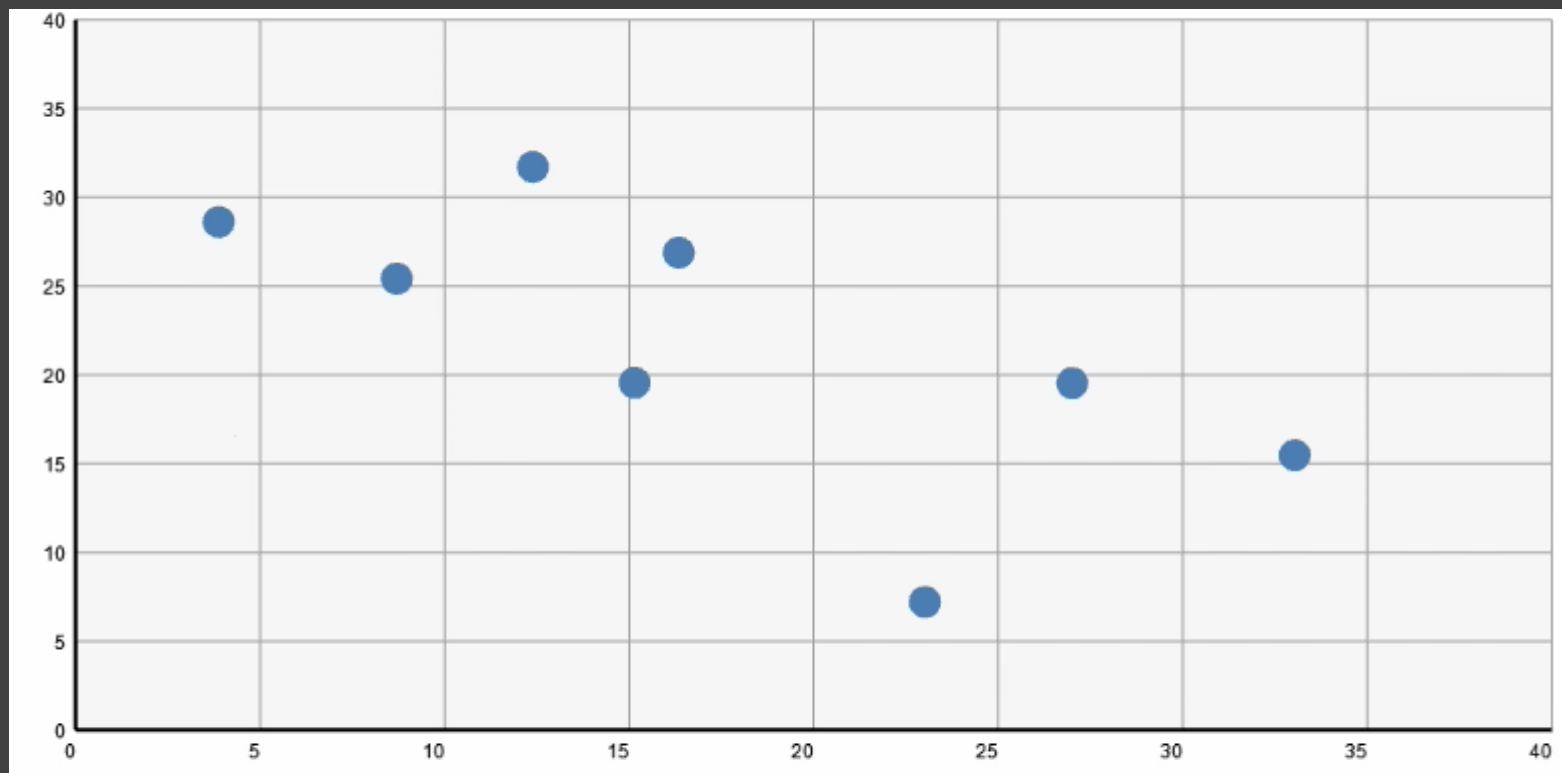
Month 2



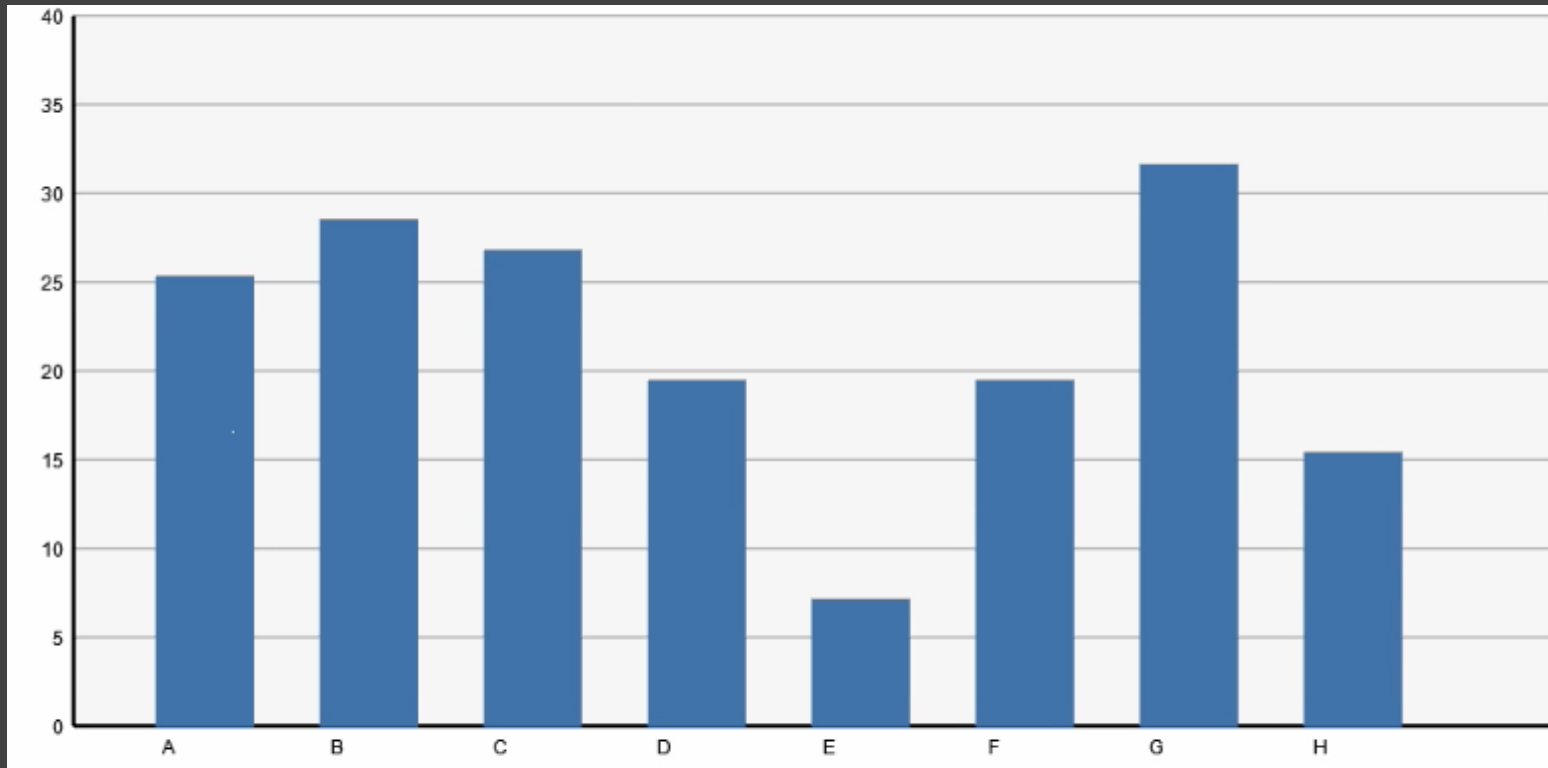


# Change Encodings

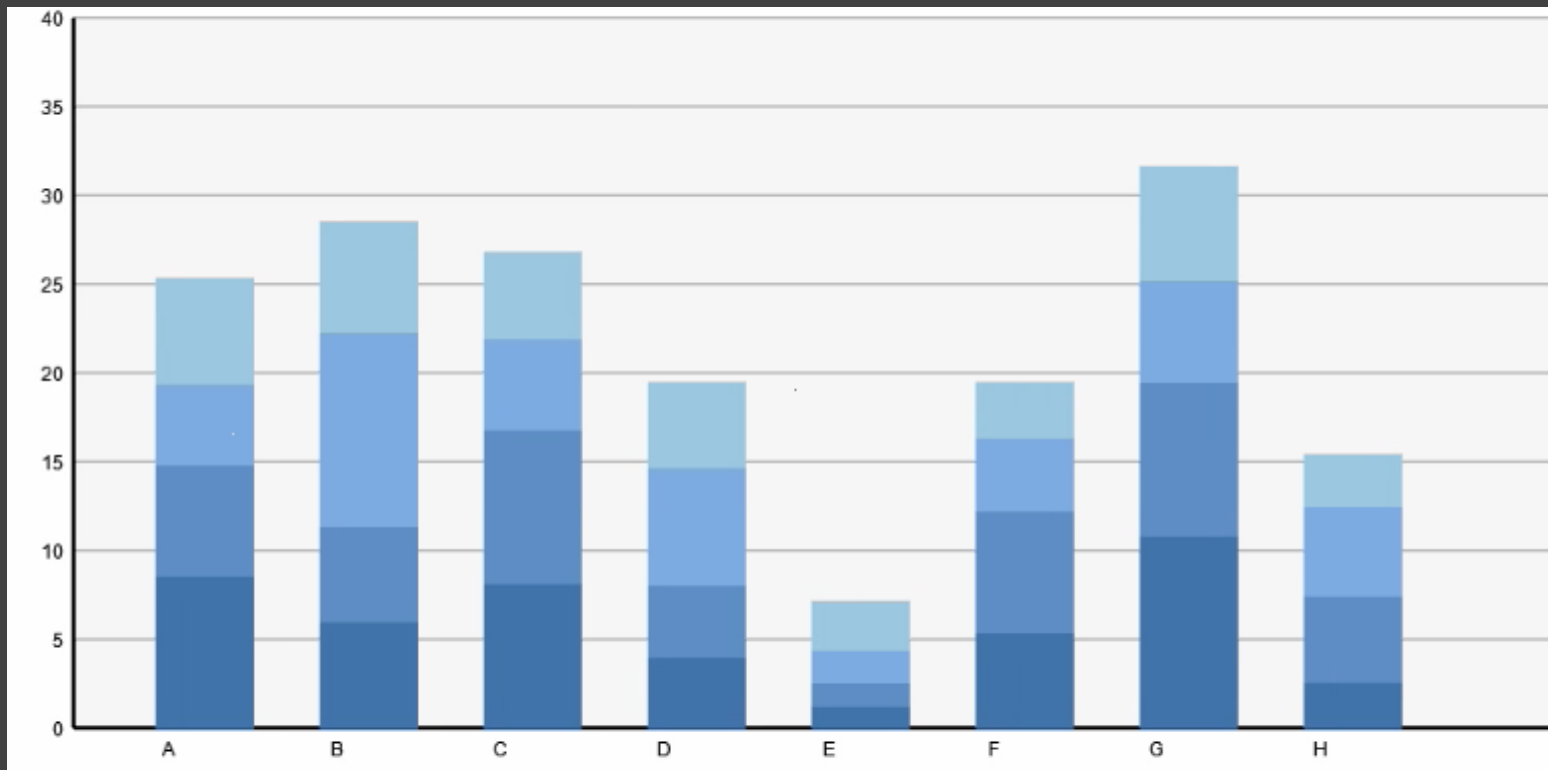




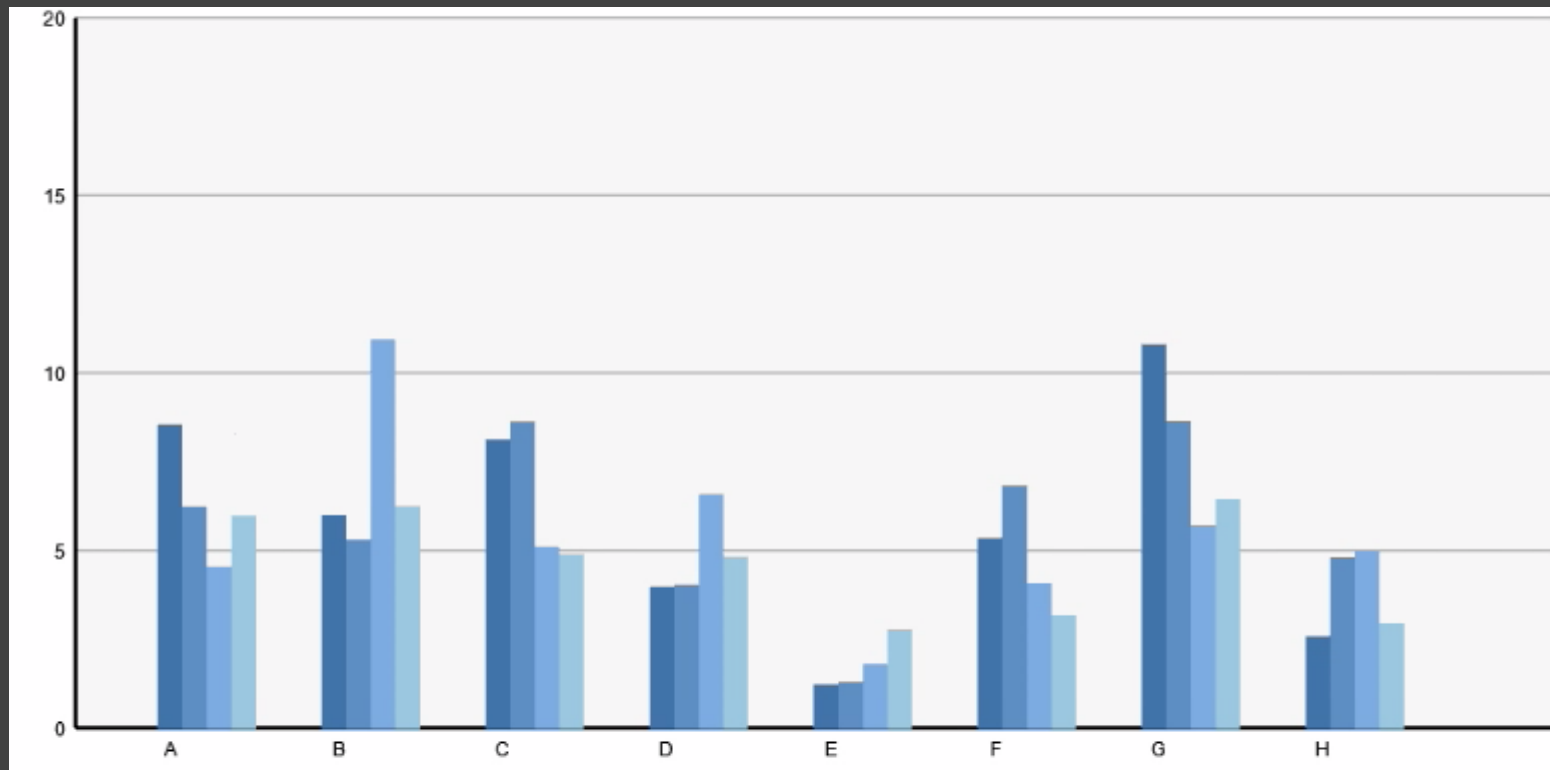
# Change Data Dimensions



# Change Data Dimensions



# Change Encodings + Axis Scales

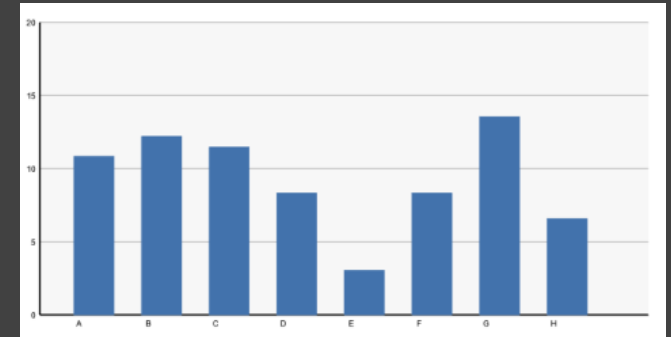


# Data Graphics & Transitions

Category	Sales	Profit
A	11	7
B	13	10
C	12	6
D	8	5
E	3	1



Visual Encoding



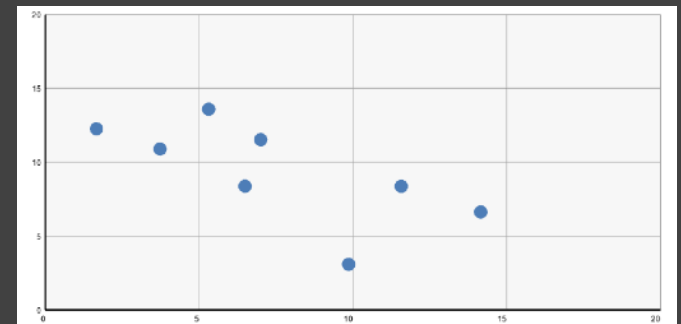
Change selected data dimensions or encodings



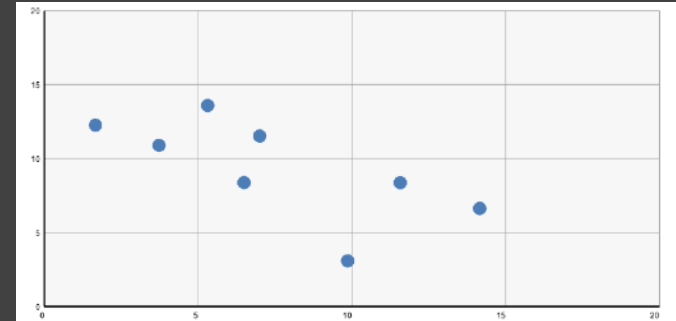
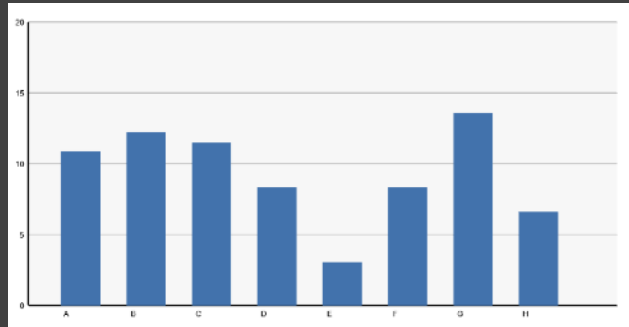
Category	Sales	Profit
A	11	7
B	13	10
C	12	6
D	8	5
E	3	1



Animation to communicate changes?



# Transitions between Data Graphics



During analysis and presentation it is common to transition between *related* data graphics.

**Can animation help?**

**How does this impact perception?**

# Principles for Animation

## **Congruence**

*Expressiveness?*

The structure and content of the external representation should correspond to the desired structure and content of the internal representation.

## **Apprehension**

*Effectiveness?*

The structure and content of the external representation should be readily and accurately perceived and comprehended.

[from Tversky 02]



# Principles for Animation [Heer]

## Congruence

- Maintain valid data graphics during transitions ←
- Use consistent syntactic/semantic mappings
- Respect semantic correspondence
- Avoid ambiguity

## Apprehension

- Group similar transitions
- Minimize occlusion
- Maximize predictability
- Use simple transitions
- Use staging for complex transitions
- Make transitions as long as needed, but no longer

# Principles for Animation [Heer]

## Congruence

Maintain valid data graphics during transitions

Use consistent syntactic/semantic mappings

Respect semantic correspondence

Avoid ambiguity



## Apprehension

Group similar transitions

Minimize occlusion

Maximize predictability

Use simple transitions

Use staging for complex transitions

Make transitions as long as needed, but no longer

Visual marks should  
always represent the  
same data tuple.

# Principles for Animation [Heer]

## Congruence

Maintain valid data graphics during transitions

Use consistent syntactic/semantic mappings

Respect semantic correspondence

Avoid ambiguity

## Apprehension

Group similar transitions


Minimize occlusion

Maximize predictability

Use simple transitions

Use staging for complex transitions

Make transitions as long as needed, but no longer



Different operators  
should have distinct  
animations.

# Principles for Animation [Heer]

## Congruence

Maintain valid data graphics during transitions

Use consistent syntactic/semantic mappings

Respect semantic correspondence

Avoid ambiguity

## Apprehension

Group similar transitions


Minimize occlusion

Maximize predictability

Use simple transitions

Use staging for complex transitions

Make transitions as long as needed, but no longer



Objects are harder to track when occluded.

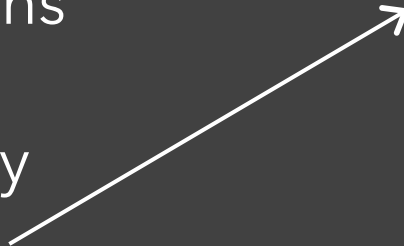
# Principles for Animation [Heer]

## Congruence

- Maintain valid data graphics during transitions
- Use consistent syntactic/semantic mappings
- Respect semantic correspondence
- Avoid ambiguity

## Apprehension

- Group similar transitions
- Minimize occlusion
- Maximize predictability
- Use simple transitions
- Use staging for complex transitions
- Make transitions as long as needed, but no longer



Keep animation as simple as possible. If complicated, break into simple stages.

# **Animated Transitions in Statistical Data Graphics**

**Jeffrey Heer  
George G. Robertson**

Microsoft  
**Research**

# Study Conclusions

**Appropriate animation improves graphical perception**

**Simple transitions beat *"do one thing at a time"***

**Simple staging was preferred and showed benefits**  
*but timing important and in need of study*

**Axis re-scaling hampers perception**

Avoid if possible (use common scale)

Maintain landmarks better (delay fade out of lines)

**Subjects preferred animated transitions**

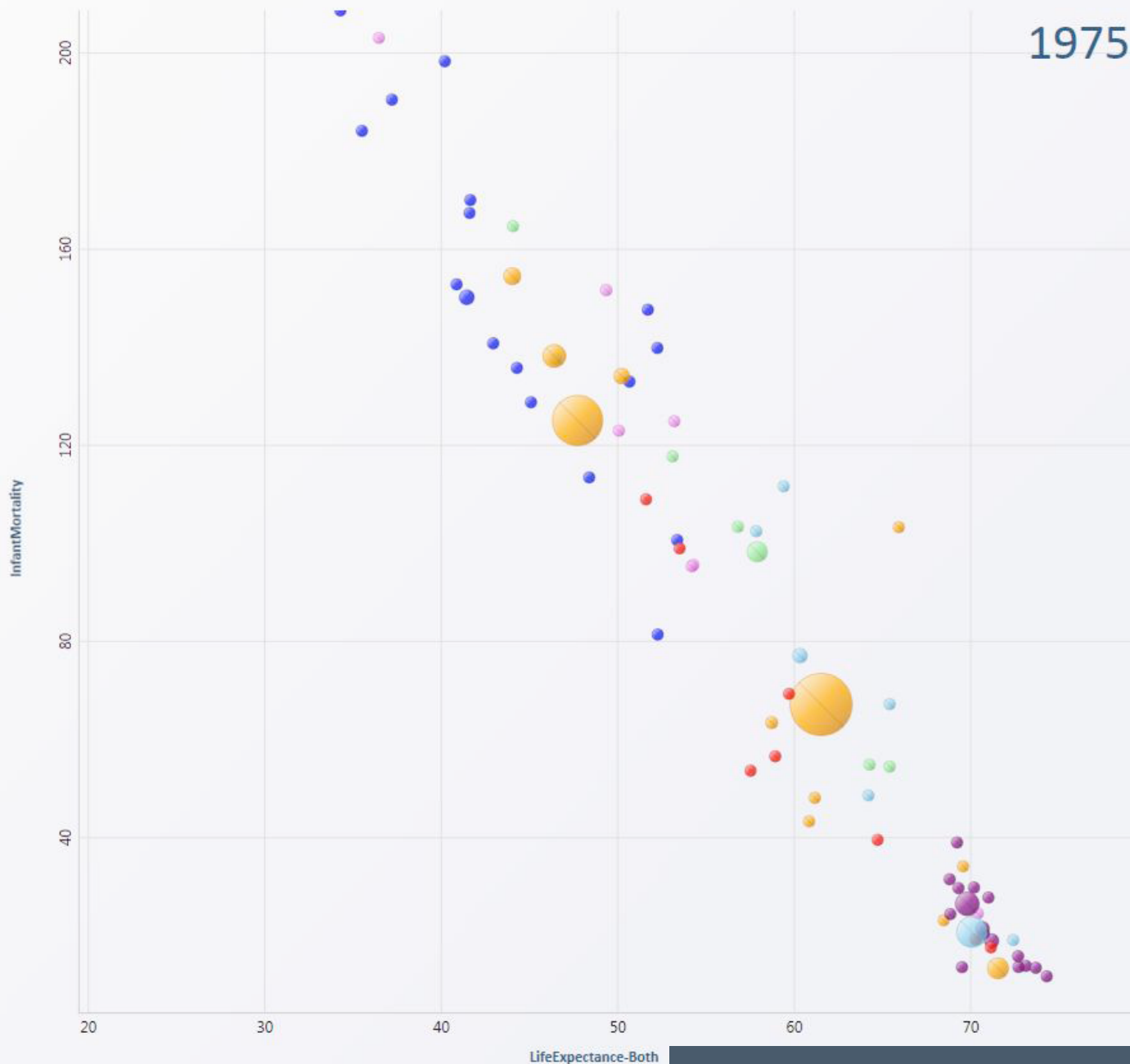
# Animation in Trend Visualization

Heer & Robertson study found that animated transitions are better than static transitions for estimating changing values.

How does animation fare vs. static time-series depictions (as opposed to static transitions)?

Experiments by Robertson et al, InfoVis 2008  
*(10 Year Test-of-Time Award at InfoVis 2018!)*





#### Color Legend (continent)

- Africa
- Asia
- Europe
- Middle East
- North America
- Oceania
- South America

#### Task

Select two countries with decreasing InfantMortality, but little change in life expectancy.

Ctrl-Click on a country (in chart) to set an answer.

Answers set: 0/2

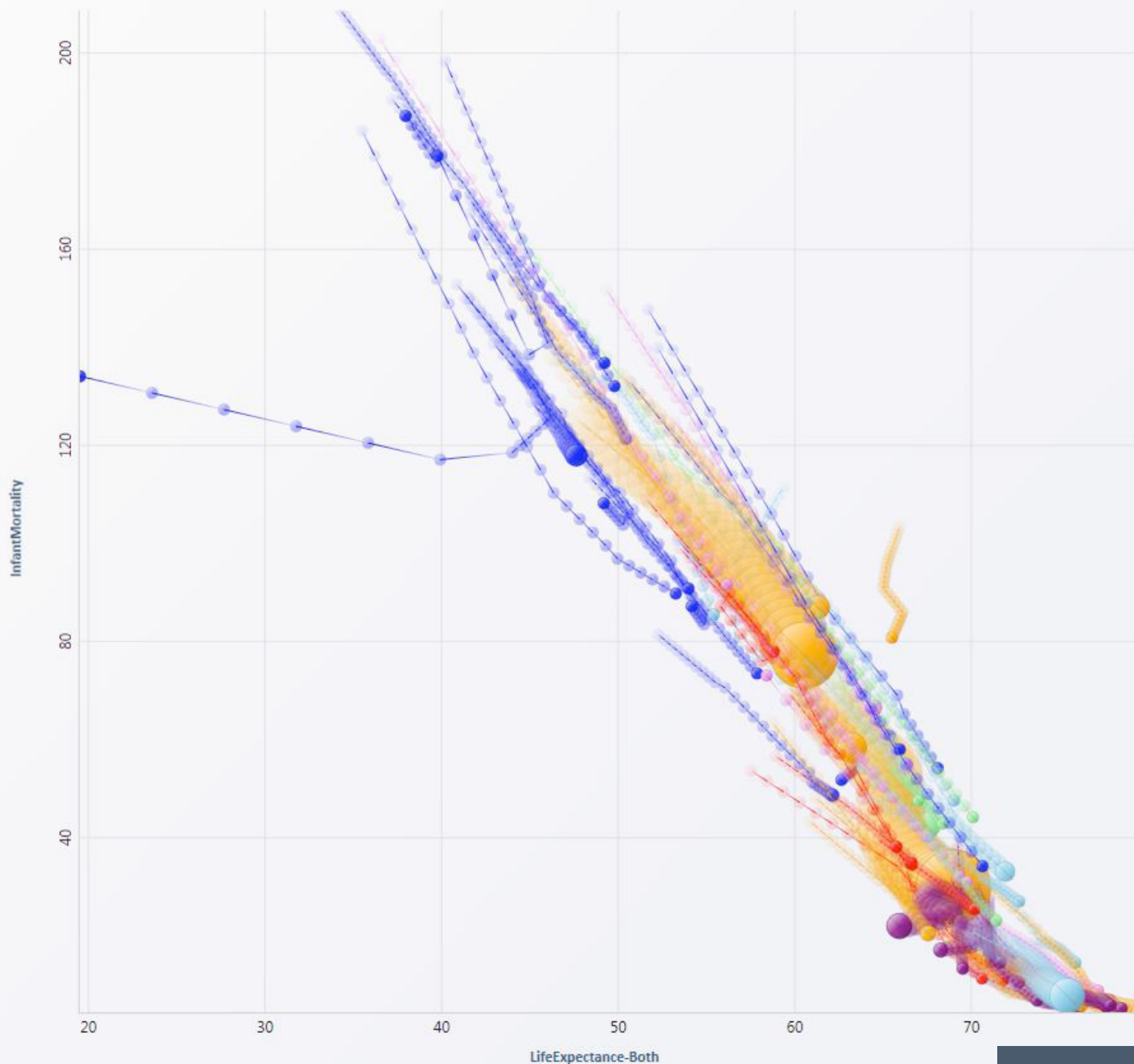
#### Next

Click on "Next" when finished (or "Give Up" if you cannot find all the answers)

Give Up

Next

Animated Scatterplot [Robertson 08]



#### Color Legend (continent)

- Africa
- Asia
- Europe
- Middle East
- North America
- Oceania
- South America

#### Task

Select two countries whose InfantMortality dropped first, then increased later.

Ctrl-Click on a country (in chart) to set an answer.

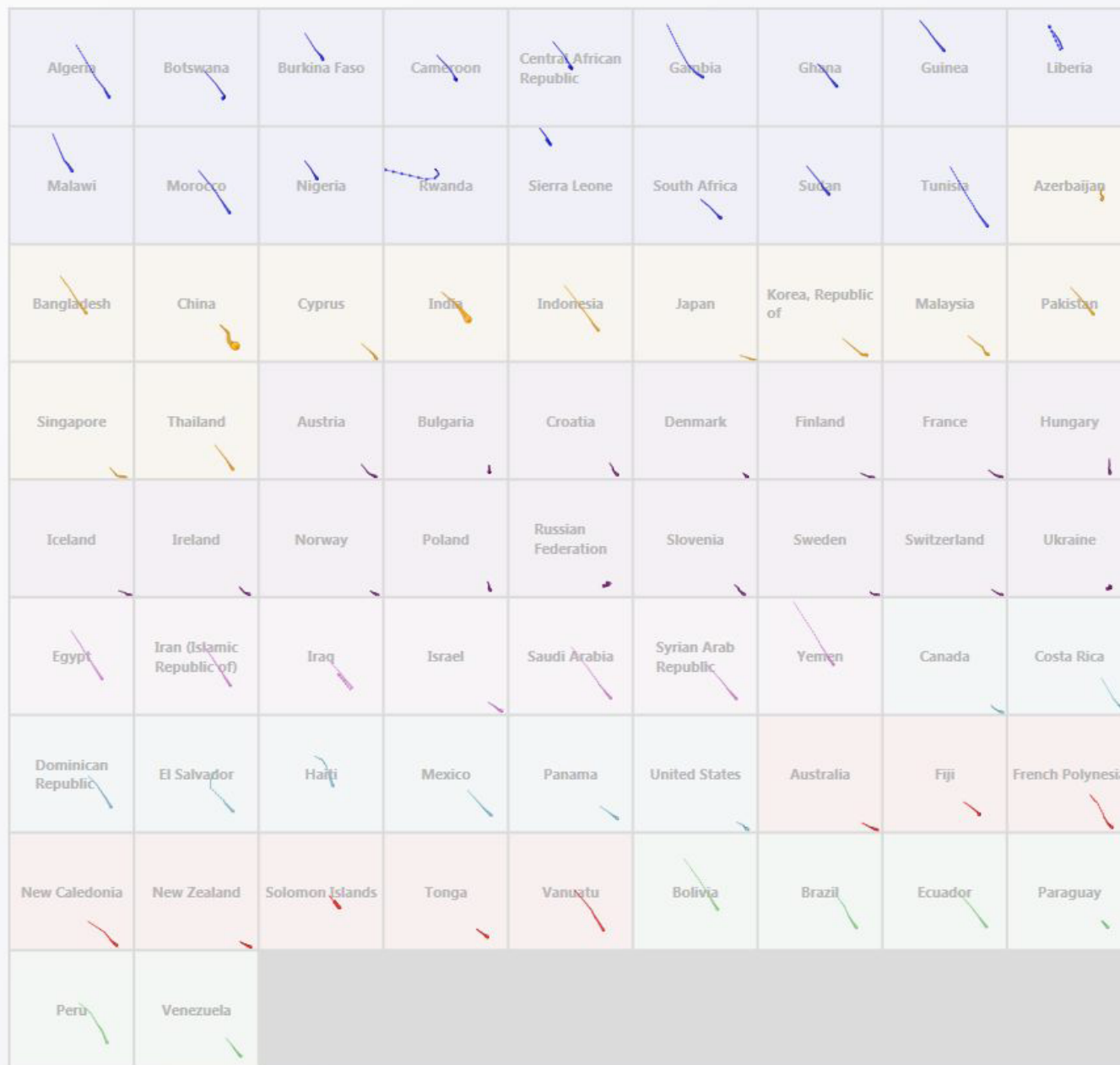
Answers set: 0/2

#### Next

Click on "Next" when finished (or "Give Up" if you cannot find all the answers)

Give Up

Next



## Color Legend (continent)

- Africa
- Asia
- Europe
- Middle East
- North America
- Oceania
- South America

## Task

Select two countries whose InfantMortality dropped first, then increased later.

Ctrl-Click on a country (in chart) to set an answer.

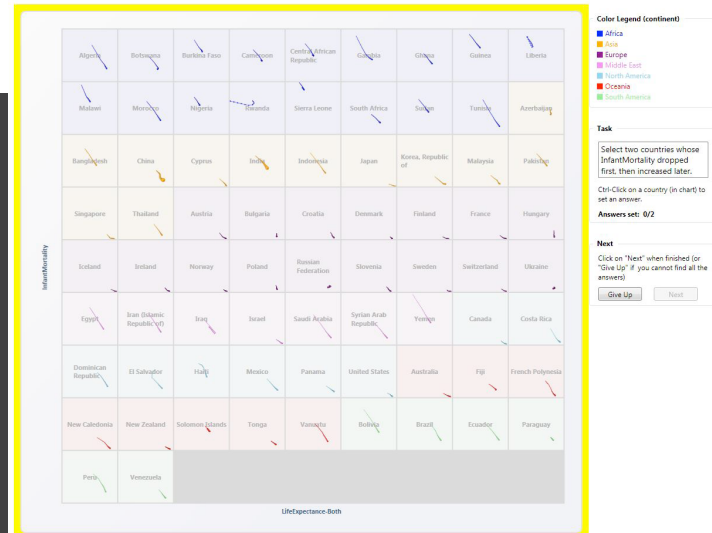
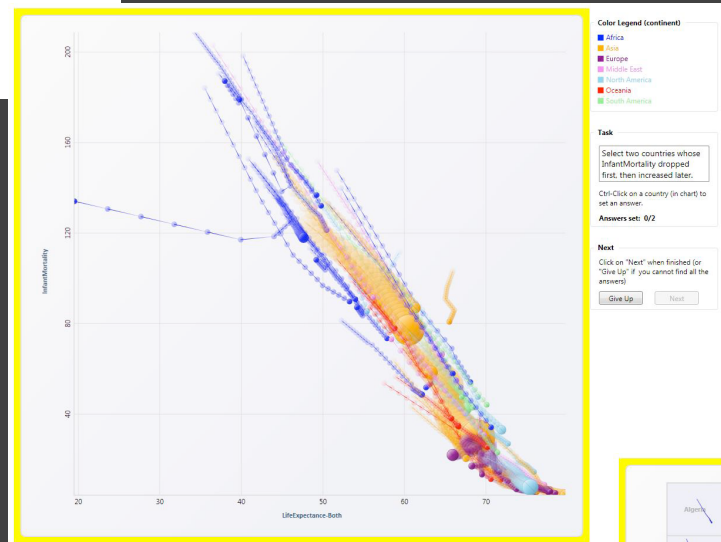
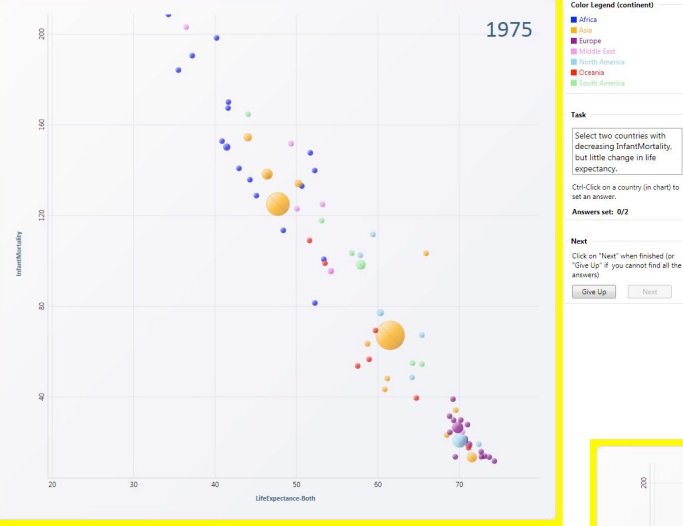
Answers set: 0/2

## Next

Click on "Next" when finished (or "Give Up" if you cannot find all the answers)

Give Up

Next



Which to prefer for analysis?  
For presentation?

# Study: Analysis & Presentation

Subjects asked comprehension questions.  
Presentation condition included narration.

Multiples 10% *more accurate* than animation

*Presentation: Anim. 60% faster* than multiples

*Analysis: Animation 82% slower* than multiples

User preferences favor animation (even though  
less accurate and slower for analysis!)

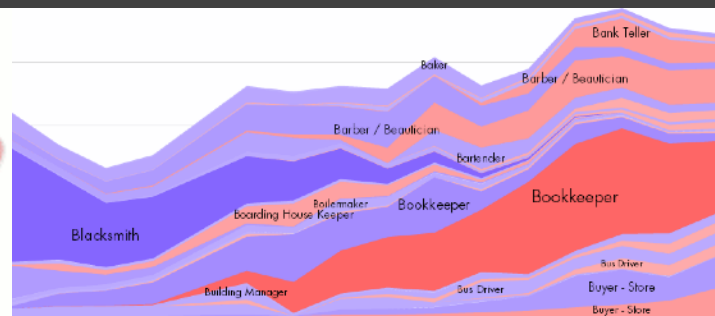
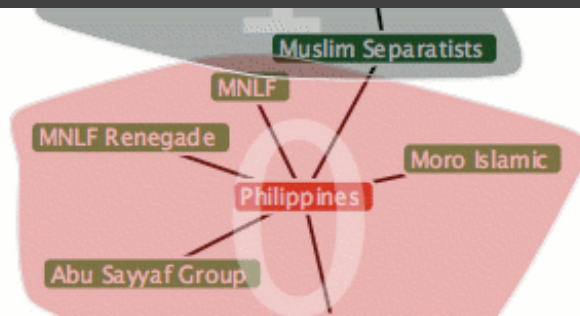
# Administrivia

# A3: Interactive Prototype

Create an interactive visualization. Choose a driving question for a dataset and develop an appropriate visualization + interaction techniques, then deploy your visualization on the web.

Due by *11:59pm* on **Monday, May 8.**

Work in project teams of 3-4 people.



# Implementing Animation



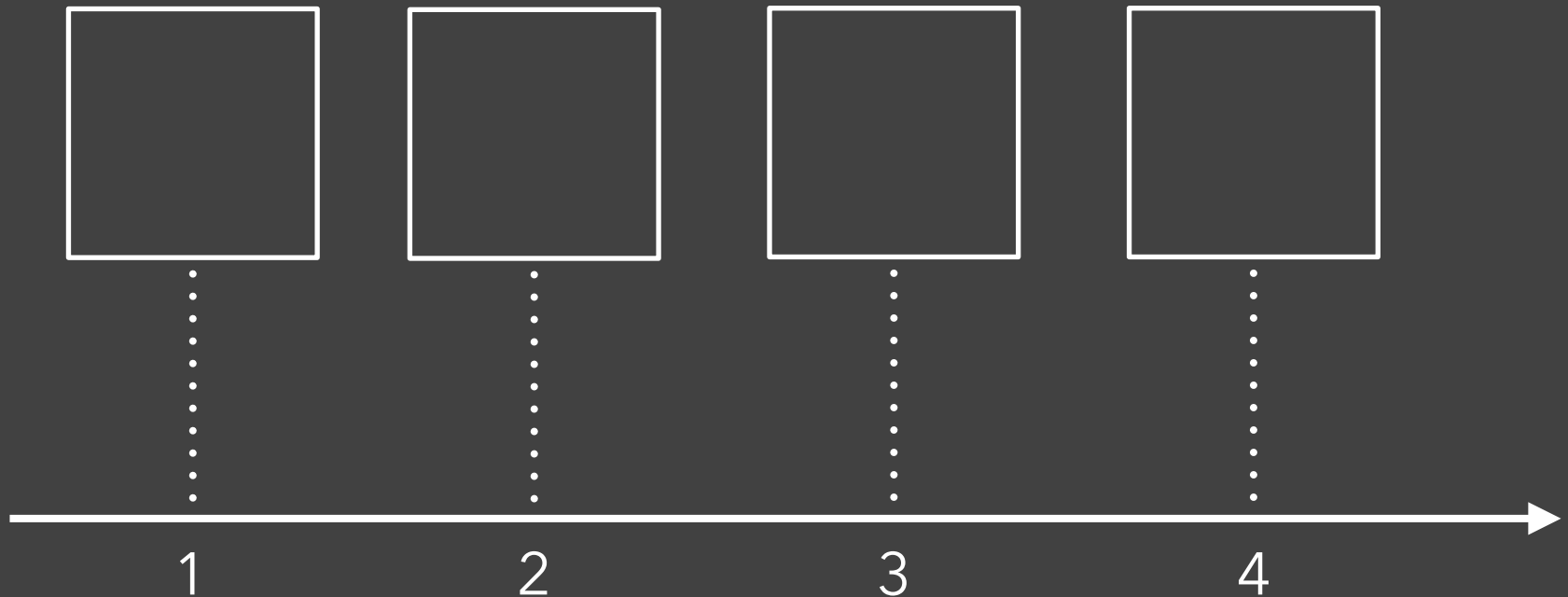
# Animation Approaches

## Frame-Based Animation

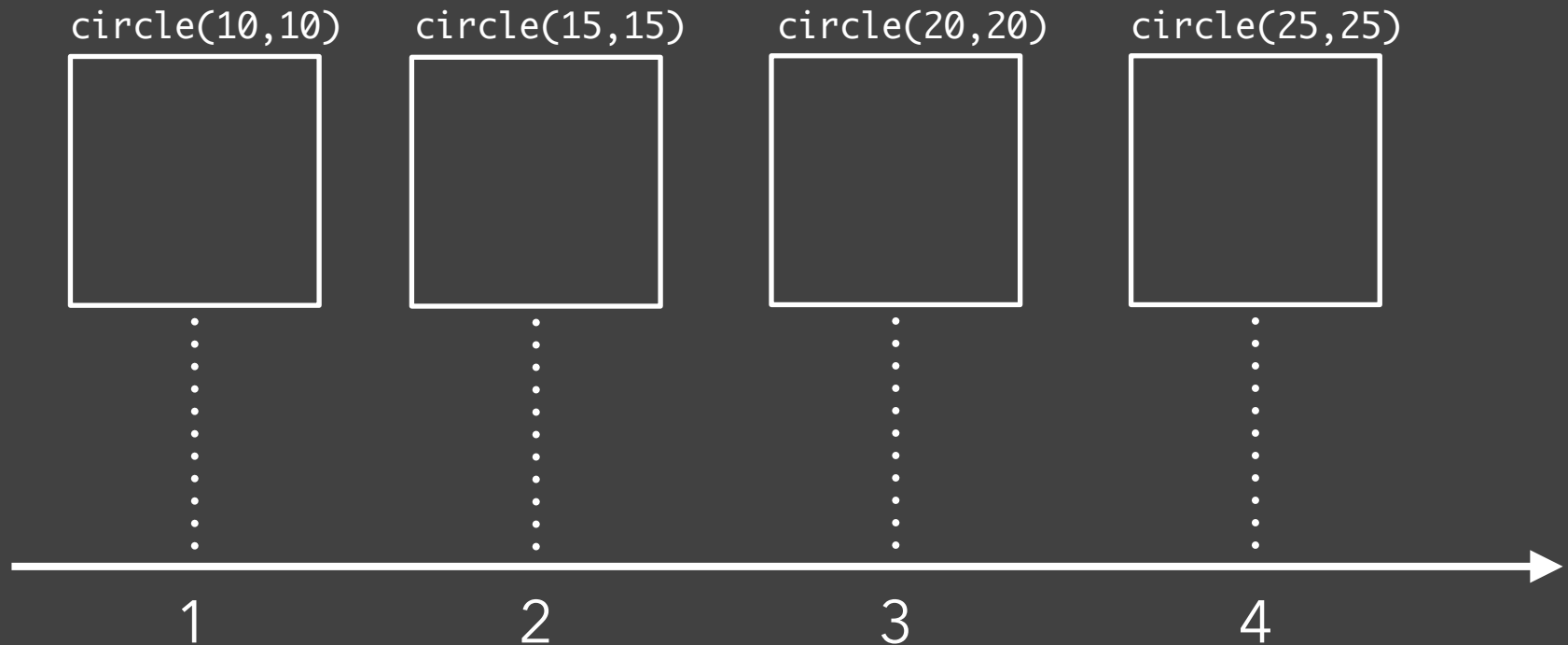
Redraw scene at regular interval (e.g., 16ms)

Developer defines the redraw function

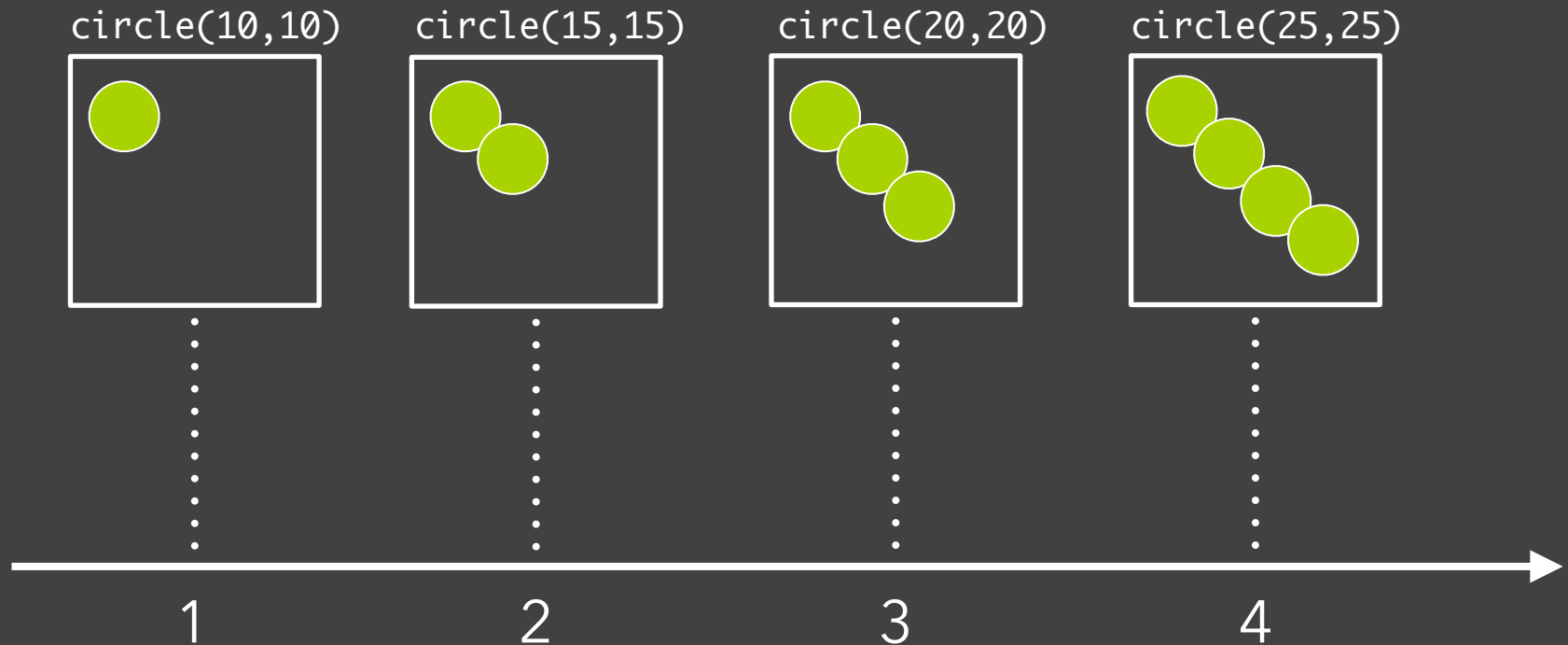
# Frame-Based Animation



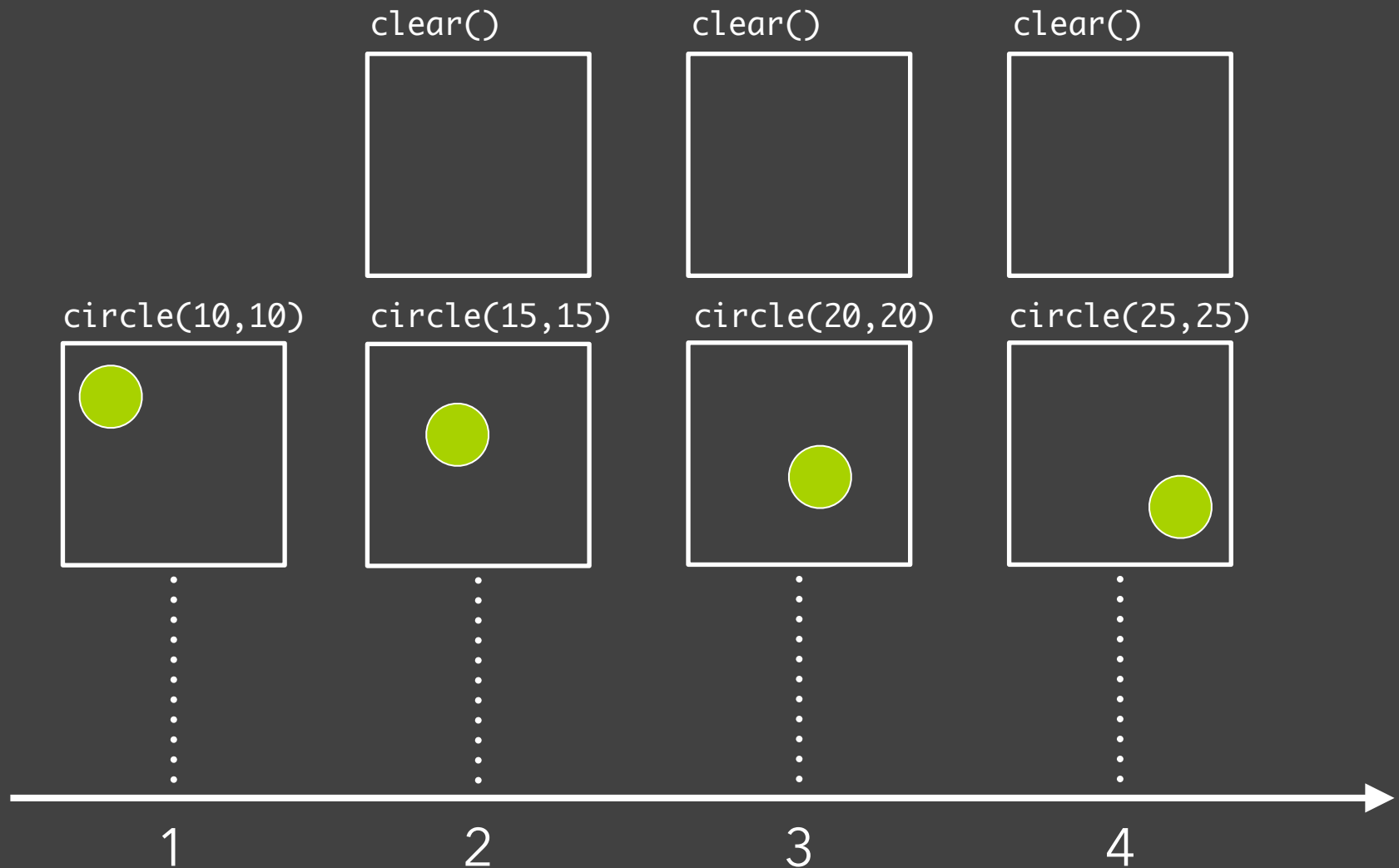
# Frame-Based Animation



# Frame-Based Animation



# Frame-Based Animation



# Animation Approaches

## Frame-Based Animation

Redraw scene at regular interval (e.g., 16ms)

Developer defines the redraw function

# Animation Approaches

## Frame-Based Animation

Redraw scene at regular interval (e.g., 16ms)

Developer defines the redraw function

## Transition-Based Animation (Hudson & Stasko '93)

Specify property value, duration & easing

Also called **tweening** (for “in-betweens”)

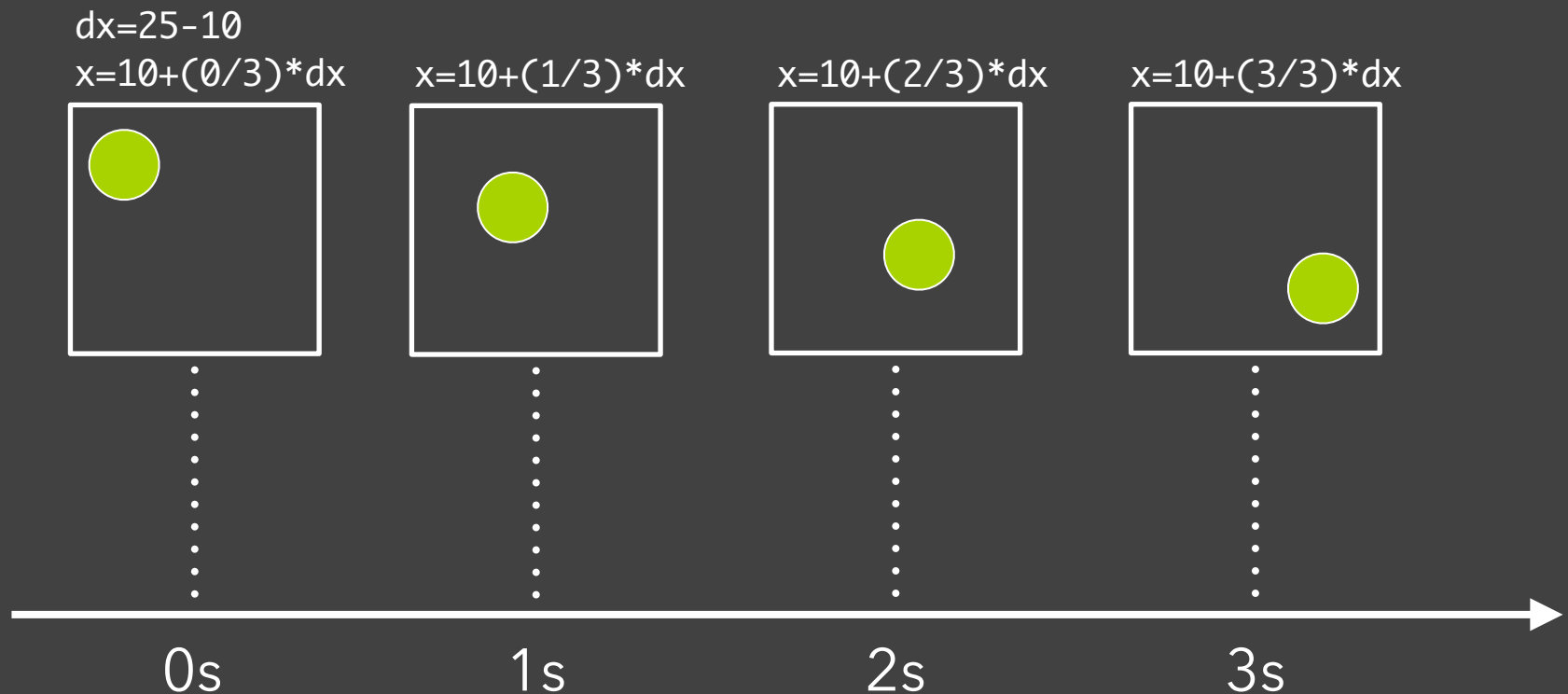
Typically computed via **interpolation**

$\text{step}(\text{fraction}) \{ x_{\text{now}} = x_{\text{start}} + \text{fraction} * (x_{\text{end}} - x_{\text{start}}); \}$

Timing & redraw managed by UI toolkit

# Transition-Based Animation

**from:** (10,10) **to:** (25,25) **duration:** 3sec

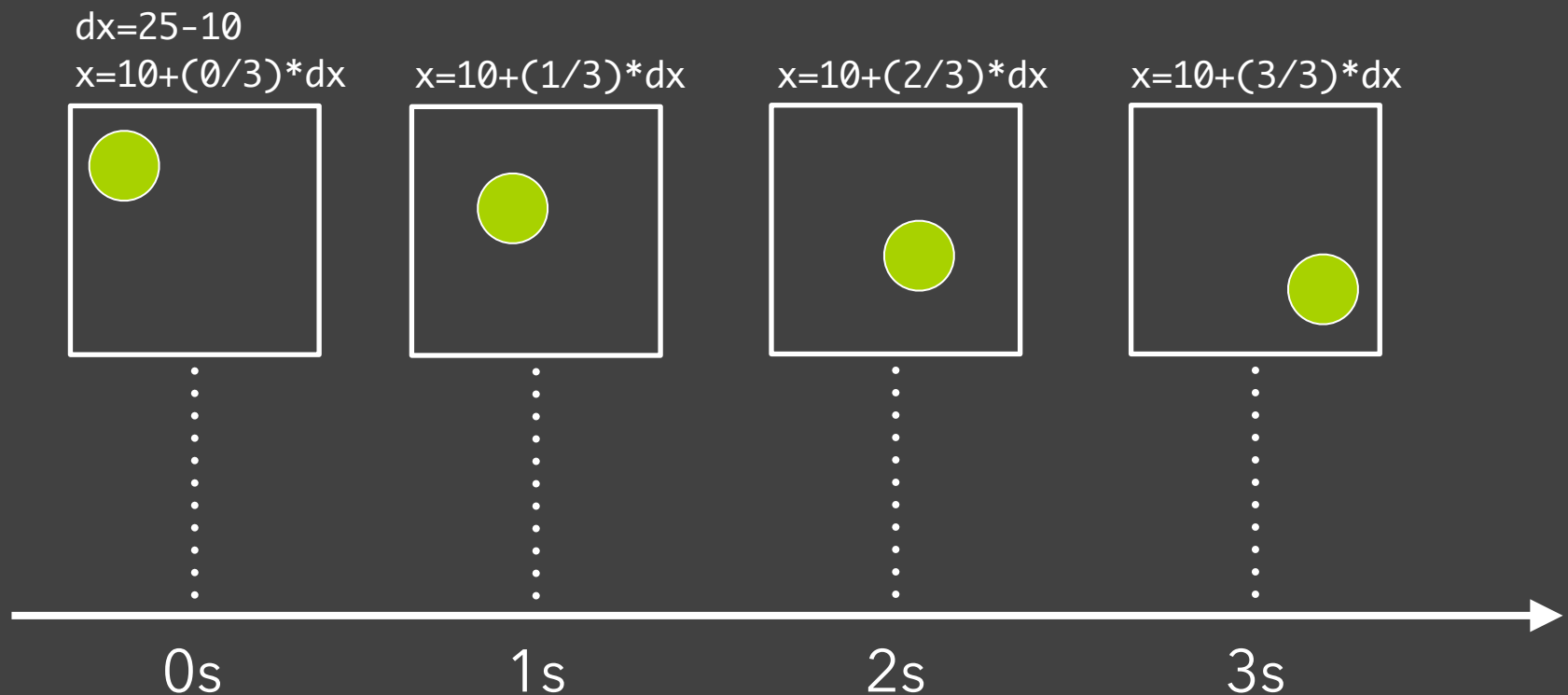




# Transition-Based Animation

**from:** (10,10) **to:** (25,25) **duration:** 3sec

*Toolkit handles frame-by-frame updates!*



# D3 Transitions

Any d3 *selection* can be used to drive animation.

# D3 Transitions

Any d3 ***selection*** can be used to drive animation.

// Select SVG rectangles and bind them to data values.

```
var bars = svg.selectAll("rect.bars").data(values);
```

# D3 Transitions

Any d3 ***selection*** can be used to drive animation.

```
// Select SVG rectangles and bind them to data values.
```

```
var bars = svg.selectAll("rect.bars").data(values);
```

```
// Static transition: update position and color of bars.
```

```
bars
```

```
  .attr("x", d => xScale(d.foo))
```

```
  .attr("y", d => yScale(d.bar))
```

```
  .style("fill", d => colorScale(d.baz));
```

# D3 Transitions

Any d3 ***selection*** can be used to drive animation.

```
// Select SVG rectangles and bind them to data values.
```

```
var bars = svg.selectAll("rect.bars").data(values);
```

```
// Animated transition: interpolate to target values using default timing
```

```
bars.transition()
```

```
  .attr("x", d => xScale(d.foo))
```

```
  .attr("y", d => yScale(d.bar))
```

```
  .style("fill", d => colorScale(d.baz));
```

# D3 Transitions

Any d3 ***selection*** can be used to drive animation.

```
// Select SVG rectangles and bind them to data values.
```

```
var bars = svg.selectAll("rect.bars").data(values);
```

```
// Animated transition: interpolate to target values using default timing
```

```
bars.transition()
```

```
  .attr("x", d => xScale(d.foo))
```

```
  .attr("y", d => yScale(d.bar))
```

```
  .style("fill", d => colorScale(d.baz));
```

```
// Animation is implicitly queued to run!
```

# D3 Transitions, Continued

```
bars.transition()  
  .duration(500)           // animation duration in milliseconds  
  .delay(0)                // onset delay in milliseconds  
  .ease(d3.easeBounce)    // set easing (or "pacing") style  
  .attr("x", (d) => xScale(d.foo))  
  ...
```

# D3 Transitions, Continued

```
bars.transition()  
  .duration(500)           // animation duration in milliseconds  
  .delay(0)                // onset delay in milliseconds  
  .ease(d3.easeBounce)    // set easing (or "pacing") style  
  .attr("x", (d) => xScale(d.foo))  
  ...
```

```
bars.exit().transition() // animate elements leaving the display  
  .style("opacity", 0)   // fade out to fully transparent  
  .remove();             // remove from DOM upon completion
```



# Easing (or “Pacing”) Functions

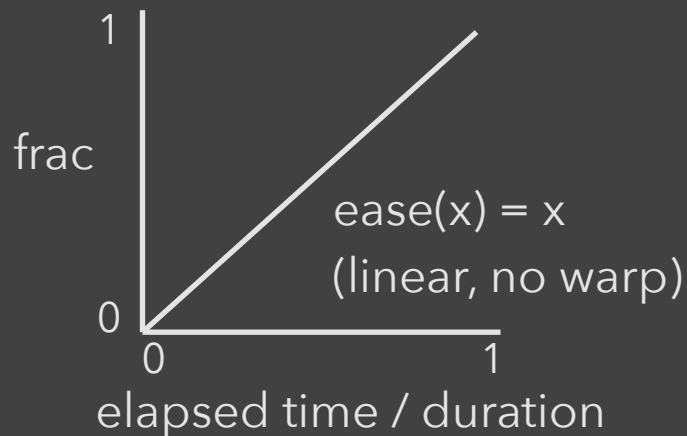
**Goals:** stylize animation, improve perception.

Basic idea is to warp time: as *duration* goes from start (0%) to end (100%), dynamically adjust the *interpolation fraction* using an **easing function**.

# Easing (or "Pacing") Functions

**Goals:** stylize animation, improve perception.

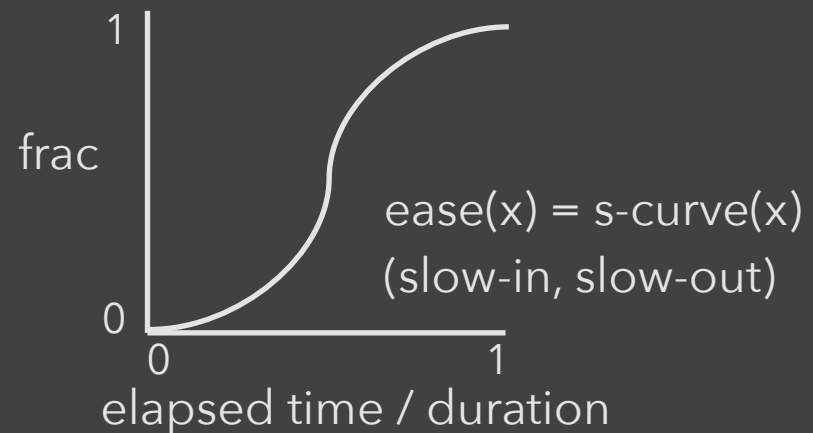
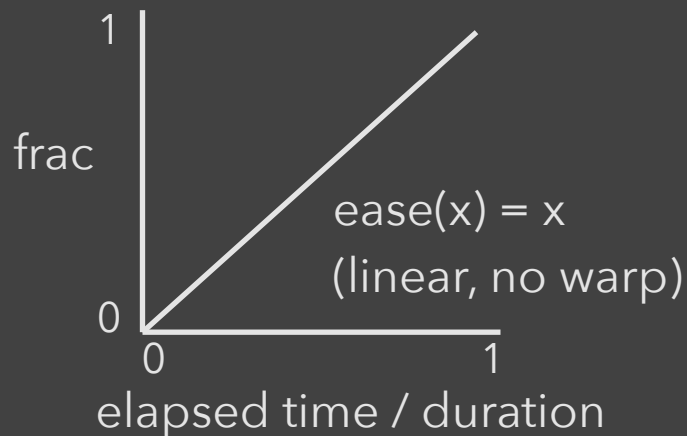
Basic idea is to warp time: as *duration* goes from start (0%) to end (100%), dynamically adjust the *interpolation fraction* using an **easing function**.



# Easing (or "Pacing") Functions

**Goals:** stylize animation, improve perception.

Basic idea is to warp time: as *duration* goes from start (0%) to end (100%), dynamically adjust the *interpolation fraction* using an **easing function**.



easeInSine



easeOutSine



easeInOutSine



easeInQuad



easeOutQuad



easeInOutQuad



easeInCubic



easeOutCubic



easeInOutCubic



easeInQuart



easeOutQuart



easeInOutQuart



easeInQuint



easeOutQuint



easeInOutQuint



easeInExpo



easeOutExpo



easeInOutExpo



easeInCirc



easeOutCirc



easeInOutCirc



easeInBack



easeOutBack



easeInOutBack



easeInElastic



easeOutElastic



easeInOutElastic



easeInBounce



easeOutBounce



easeInOutBounce



<http://easings.net/>

# CSS Transitions

## Extends CSS with Animated Transitions

```
a {  
  color: black;  
  transition: color 1s ease-in-out;  
}
```

```
a:hover {  
  color: red;  
}
```

# CSS Transitions

## Extends CSS with Animated Transitions

```
a {  
  color: black;  
  transition: color 1s ease-in-out;  
}
```

**Property**      **Duration**      **Easing**

```
a:hover {  
  color: red;  
}
```

# CSS Transitions

## Extends CSS with Animated Transitions

```
a {  
  color: black;  
  transition: color 1s ease-in-out;  
}
```

**Duration** ↓

↑ **Property**      ↑ **Easing**

```
a:hover {  
  color: red;  
}
```

← **Animate color transition upon mouse in / out.**

# Summary

**Animation is a salient visual phenomenon**

Attention, object constancy, causality, timing

Design with care: congruence & apprehension

For transitions, animation has demonstrated benefits, but **consider task and timing**