INFORMATION VISUALIZATION

Alvitta Ottley

Washington University in St. Louis
Recap...
LAST CLASS

Centered around visualization design:
1. What’s a good design?
2. What’s a bad design?
3. How do you improve a bad design?
From the reading: "Our study showed that if memorability is important, visual imagery can help to fix a chart in a viewer’s memory. However, we do not advocate this strategy as a general principle, because the use of strong images in charts is contentious – for example, images convey messages that may intentionally or unintentionally bias the viewer’s interpretation of the data."

Does this mean that a visual of this type is only necessary as a secondary reference rather than a primary reference to reveal data? For instance, when you're pointing out an anomaly, but don't want to imply what the anomaly means?
Today…

Processing
IN-CLASS EXERCISE: DRAW A BUTTON

- Draw a button (size and location unspecified)
- Choose a color for your button
- Choose some text to be placed on your button
- Clicking the button should change the color of your button. A subsequent click should revert the color to the previous one.
- Clicking outside of the button should do nothing.
- Creativity encouraged!
```cpp
int bgColor = 255;
color textColor = color(0, 102, 153);
int initialX = 30;
int initialY = 20;
int rectWidth = 55;
int rectHeight = 55;

void setup(){
  size(300, 300);
}

//loop
void draw() {
  fill(bgColor);
  rect(initialX, initialY, rectWidth, rectHeight, 7);
  fill(textColor);
  textAlign(CENTER, CENTER);
  text("click me", initialX + rectWidth/2, initialY + rectHeight/2);
}

void mouseClicked() {
  if((mouseX >= initialX && mouseX <= initialX + rectWidth) &&
     (mouseY >= initialY && mouseY <= initialY + rectHeight)){
    if (bgColor == 0) {
      bgColor = 255;
    } else {
      bgColor = 0;
    }
  }
}
```
IN-CLASS EXERCISE: BOUNCING BALL

• Draw a ball (size and initial location unspecified)
• Choose a color for your ball
• Your ball should appear to be bouncing (directionality and speed unspecified)
• Your ball should not leave the canvas while bouncing
• Creativity encouraged!
JOHN’S BOUNCING BALL SOLUTION

```cpp
float initialX = 60.0;
float initialY = 80.0;
float cWidth = 55.0;
float cHeight = 55.0;
color bgColor = color(0, 102, 153);
float vectorX, vectorY, x, y = 0.0;
int speed = 3;
int canvasWidth = 600;
int canvasHeight = 600;

void setup(){
  size(600, 600);
  x = initialX;
  y = initialY;
  ellipseMode(CORNER);
  ellipse(x, y, cWidth, cHeight);
}

//loop
void draw() {
  background(bgColor);
  x += (vectorX * speed);
  y += (vectorY * speed);
  if(x + cWidth > canvasWidth || x < 0)
    vectorX *= -1;
  if(y + cHeight > canvasHeight || y < 0)
    vectorY *= -1;
  ellipse(x, y, cWidth, cHeight);
}

void mouseClicked() {
  vectorX = random(10.0);
  vectorY = random(10.0);
}
```
NEXT TIME...

Data Types &
2-Dimensional Data Representations