

15-104 Introduction to Computing for Creative Practice

Fall 2020

05 Interaction

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

1



dist

- Function that calculates and returns the distance between two points.
(can also do three dimensions later)
- `dist(x1, y1, x2, y2)`
Parameters:
 - `x1` Number: x-coordinate of the first point
 - `y1` Number: y-coordinate of the first point
 - `x2` Number: x-coordinate of the second point
 - `y2` Number: y-coordinate of the second point
- Example:
`dist(0, 0, 30, 40)` returns 50

2

Example

```

var x = 150;
var y = 100;
var d = 25;
var g = 0;

function setup() {
  createCanvas(300, 200);
}

function draw() {
  print(g.toString());
  background(g);
  ellipse(x, y, d, d);
}

function mousePressed() {
  if (dist(mouseX, mouseY,
          x, y) < d/2) {
    g += 32;
  }
}

```

What does this program do?

3

What is the difference?

```


var x = 150;
var y = 100;
var d = 25;
var g = 0;

function setup() {
  createCanvas(300, 200);
}

function draw() {
  if (mouseIsPressed) {
    if (dist(mouseX,
            mouseY, x, y) < d/2) {
      g += 32;
    }
  }
  print(g.toString());
  background(g);
  ellipse(x, y, d, d);
}

```


4



mouseIsPressed vs. mousePressed()

- `mouseIsPressed` is a Boolean value
- `mousePressed()` is a function
- `mouseIsPressed` is true as long as the mouse is pressed down
- `mousePressed()` is called at the moment when the mouse is pressed down
- Can you now explain the behavior of both of these examples?
- NOTE: Open the console while you run the program to see the value of `g` as the program runs.

5



Data Types

- *Number* in Javascript is an approximation of a real number in math. Real numbers include things like pi, 7.0, 11/7, and 1.1. We say approximation because numbers in Javascript have finite precision: about 12 digits. However, integers (up to about 16 decimal digits) are represented exactly.
 - Literal numbers are written as decimal integers (1776), with decimal points (98.6), or scientific notation (1.23e-10).
- *Boolean* in Javascript is `true` or `false`. Boolean values result from comparison operators (e.g. `4 < 5` has the Boolean value of `true`) or logical combinations of Boolean values (e.g. `true || false` has the Boolean value of `true`).
 - Operators `&&`, `||`, `==` and `===` return Boolean values.
 - `===` compares for “true” equality: same value, same type
 - `==` compares for loosely equal values, but `false == 0`, and `“10” == 10`
 - In an if statement with a condition that is a Number, any not-zero number is interpreted to mean true.

6

Detecting mouse buttons

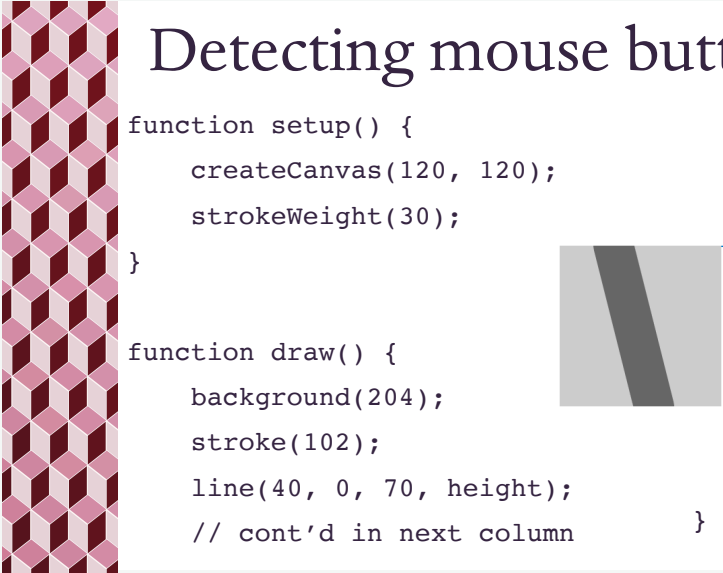
```

function setup() {
  createCanvas(120, 120);
  strokeWeight(30);
}

function draw() {
  background(204);
  stroke(102);
  line(40, 0, 70, height);
  // cont'd in next column
}

if (mouseIsPressed) {
  if (mouseButton == LEFT)
  {
    stroke(255);
  } else {
    stroke(0);
  }
  line(0, 70, width, 50);
}

```



7

Detecting Mouse Buttons

- `mouseButton` can have three values:
LEFT
CENTER
RIGHT
- Set based on whatever button was pressed most recently.
- Different browsers may track `mouseButton` differently.
- On Macs, right click is usually done by holding down the control key while you click.

8

What does this program do?

```

var x;
var offset = 10;

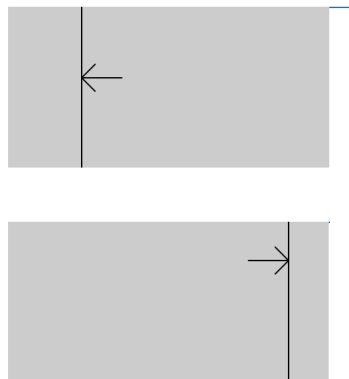
function setup() {
  createCanvas(240, 120);
  x = width / 2;
}

function draw() {
  background(204);
  if (mouseX > x) {
    x += 0.5;
    offset = -10;
  }
  // cont'd in next column
  if (mouseX < x) {
    x -= 0.5;
    offset = 10;
  }
  line(x, 0, x, height);
  line(mouseX, mouseY,
        mouseX + offset,
        mouseY - 10);
  line(mouseX, mouseY,
        mouseX + offset,
        mouseY + 10);
  line(mouseX, mouseY,
        mouseX + offset * 3,
        mouseY);
}

```


9

Chase The Arrow!



- How is `offset` used?
- What commands draw the arrow?
- What happens when:
 - `mouseX < x`?
 - `mouseX > x`?
 - `mouseX == x`?

10




Another way to draw the arrows

```

var size = 10;
Then in the draw function:
line(x, 0, x, height); // vertical line at x
if (offset < 0) {      // arrow to the left
  line(mouseX, mouseY, mouseX - size, mouseY - size);
  line(mouseX, mouseY, mouseX - size, mouseY + size);
  line(mouseX, mouseY, mouseX - size * 3, mouseY);
} else { // arrow to the right
  line(mouseX, mouseY, mouseX + size, mouseY - size);
  line(mouseX, mouseY, mouseX + size, mouseY + size);
  line(mouseX, mouseY, mouseX + size * 3, mouseY);
}

```

11



Color Flip (Boolean flag/toggle)

```

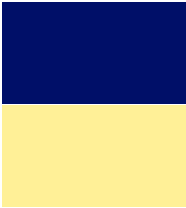
var on = false;
var r = 255;
var g = 240;
var b = 151;

function setup() {
  createCanvas(300, 200);
}


function draw() {
  if (on == true) {
    background(r, g, b );
  } else {
    background(255-r, 255-g, 255-b);
  }
}

function mousePressed() {
  if (on == true) {
    on = false;
  } else {
    on = true;
  }
}

```



12




Boolean shortcuts

```
function draw() {
  if (on == true) {
    background(r, g, b );
  } else {
    background(255-r, 255-g, 255-b);
  }
}
```

```
function draw() {
  if (on) {
    background(r, g, b );
  } else {
    background(255-r, 255-g, 255-b);
  }
}
```

13



Boolean shortcuts

```
function draw() {
  if (on == false) {
    background(255-r, 255-g, 255-b);
  } else {
    background(r, g, b);
  }
}
```

```
function draw() {
  if (!on) {
    background(255-r, 255-g, 255-b);
  } else {
    background(r, g, b);
  }
}
```

14

Boolean shortcuts

```
function mousePressed() {
  if (on) {
    on = false;
  } else {
    on = true;
  }
}
```

```
function mousePressed() {
  on = !on;
}
```

```
function mousePressed() {
  if (!on) {
    on = true;
  } else {
    on = false;
  }
}
```

15

Try this:

- `pmouseX` and `pmouseY` store the position of the mouse from the previous frame.
- Write a p5.js program that allows you to “write” on the canvas by drawing a line from the previous mouse position to the current one, over and over as the user moves the mouse along the canvas.
- Update the program so that when the mouse is clicked, the canvas is erased and the user can draw on a fresh canvas.

16