

15-104 Introduction to Computing for Creative Practice

Fall 2020

07 More Interaction

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

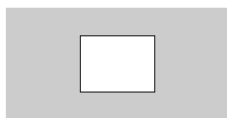
1



Mouse Inside?

```
var x = 80;
var y = 30;
var w = 80;
var h = 60;


function setup() {
  createCanvas(240, 120);
}
```



```
function draw() {
  background(204);
  if ((mouseX > x) &&
      (mouseX < x+w) &&
      (mouseY > y) &&
      (mouseY < y+h)) {
    fill(0);
  } else {
    fill(255);
  }
  rect(x, y, w, h);
}
```

What would happen
if you used || instead
of &&?

2



keyIsPressed and key


```
function setup() {
  createCanvas(240, 120);
}
function draw() {
  background(204);
  line(20, 20, 220, 100);
  if (keyIsPressed) {
    line(220, 20, 20, 100);
  }
}
```

If a key is pressed, display the second line.

```
function setup() {
  createCanvas(120, 120);
  textSize(64);
  textAlign(CENTER);
}
function draw() {
  background(255);
  text(key, 60, 80);
}
```

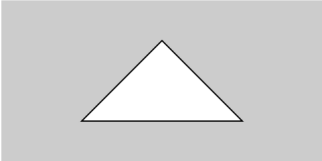
Display the key that is pressed as a string centered on the canvas at (60,80) in 64pt font size.

3



Shape Selector

```
function setup() {
  createCanvas(240, 120);
}
function draw() {
  background(204);
  if (keyIsPressed) {
    if (key == "e") {
      ellipse(120, 60, 120, 60);
    } else if (key == "c") {
      circle(120, 60, 80);
    } else if (key == "r") {
      rect(60, 30, 120, 60);
    } else if (key == "s") {
      square(80, 20, 80);
    } else if (key == "t") {
      triangle(60, 90,
        180, 90, 120, 30);
    }
  }
}
```



4



Question

- How do you update the previous program so it displays the ellipse for any e?

```
if (key == "e" || key == "E") {  
  ellipse(120, 60, 120, 60);  
}
```

5



What does this program do?

```
var count = 0;                                function mousePressed() {  
function setup() {                             count += 1;  
  createCanvas(240, 120);                     }  
  textSize(64);  
  textAlign(CENTER);  
}  
function draw() {  
  background(255);  
  text(count.toString(),  
        60, 80);  
}
```

6

Drag and Drop

```

var x = 200;
var y = 200;
var diam = 5;
var dragging = false;
function setup() {
  createCanvas(400, 400);
  fill(0);
}
function draw() {
  background(255);
  if (dragging) {
    x = mouseX;
    y = mouseY;
  }
  ellipse(x, y, diam, diam);
}

```

```

function mousePressed() {
  if (dist(x, y, mouseX, mouseY)
      < diameter/2) {
    dragging = true;
  }
}

```

```

function mouseReleased() {
  dragging = false;
}

```

What is the `mousePressed` function testing?
How is the Boolean variable `dragging` used?

7

Predict Output

```

function setup() {
  createCanvas(240, 120);
  strokeWeight(12);
}
function draw() {
  background(204);
  stroke(102);
  var gx = _____;
  line(gx, 0, gx, height); // Gray line
  stroke(0);
  var bx = _____;
  line(bx, 0, bx, height); // Black line
}

```

Case 1:

```

gx = mouseX - 60;
bx = mouseX + 60;

```

Case 2:

```

gx = constrain(mouseX, 0, 240);
bx = gx / 2 + 60;

```



8

map function

```
map(value, start1, stop1, start2, stop2, [withinBounds])
```

Parameters:

value	Number: the incoming value to be converted
start1	Number: lower bound of the value's current range
stop1	Number: upper bound of the value's current range
start2	Number: lower bound of the value's target range
stop2	Number: upper bound of the value's target range
withinBounds	Boolean: constrain the value to the newly mapped range (optional)

Purpose: Re-maps a number from one range to another.

Examples: `map(25, 0, 100, 0, 600)` returns 150
`map(125, 0, 100, 0, 600, true)` returns 600

9

map function

```
map(25, 0, 100, 0, 600) returns 150
```

0	25	100
----- ----- ----- -----		
0	150	600

```
map(125, 0, 100, 0, 600, true) returns 600
```

0	100	125
----- ----- ----- -----		
0	600	

10

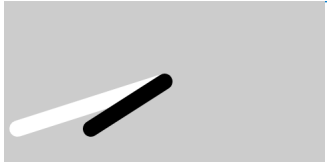
Map Example

```

function setup() {
  createCanvas(240, 120);
  strokeWeight(12);
}

function draw() {
  background(204);
  stroke(255);
  line(120, 60, mouseX, mouseY); // White line
  stroke(0);
  var mx = map(mouseX, 0, width, 60, 180);
  line(120, 60, mx, mouseY); // Black line
}

```



11

Easing

```

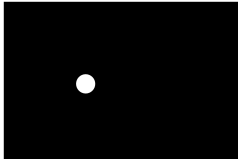
var x = 300;
var y = 300;
var dx = 0;
var dy = 0;

function setup() {
  createCanvas(600, 400);
}

function draw() {
  background(0);
  dx = mouseX - x;
  dy = mouseY - y;
  x = x + 0.1*dx;
  y = y + 0.1*dy;
  ellipse(x, y, 50, 50);
}

```

Why does the ellipse slowly follow the mouse, but catches up when the mouse stops?



12



Try this:

- Modify the “easing” program below so that the speed of following the mouse is controlled by a variable.
 - Try values of 0.2 (faster) and 0.01 (slower).
- Add another ellipse that also follows the mouse, but which has a different rate of convergence.