15 Functions

Circle Hit Testing

Chapter 9, Examples 9-1, 9-2, 9-8

Chapter 5, Example 5-14
## Hit Testing (Bounds Detection)

<table>
<thead>
<tr>
<th>Area</th>
<th>Side of canvas</th>
<th>rectangle</th>
<th>circle</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lecture 08</td>
<td>Lecture 09</td>
<td>Lecture 15</td>
</tr>
</tbody>
</table>
(Review) conditional_dot2 (if else)

... 

// the conditional statement
if (mouseX > width / 2) {
  fill("#FF0000"); // red
  ellipse(0.75 * width, 50, 30, 30);
} else {
  fill("#0000FF"); // blue
  ellipse(0.25 * width, 50, 30, 30);
}

... 

https://editor.p5js.org/cs105/sketches/nuHTR21Ci
(Review) rect-hittest

...  

// the hit test
if (mouseX >= x && mouseX <= x + w &&
    mouseY >= y && mouseY <= y + h) {
    fill("#ff0000"); // red
} else {
    fill("#FFFFFF"); // white
}

...
circleHitTest
Circle Hit Testing (Bounds Detection)

- Rectangle hit test won’t work for a circle because...

Rectangle Hit Test: [https://editor.p5js.org/sanghosuh/sketches/uKhBBQvv](https://editor.p5js.org/sanghosuh/sketches/uKhBBQvv)
Circle Hit Testing (Bounds Detection)

- Ideally...

Rectangle Hit Test: [https://editor.p5js.org/sanghosuh/sketches/qfaYUvTX](https://editor.p5js.org/sanghosuh/sketches/qfaYUvTX)
Circle Hit Testing (Bounds Detection)

Hit

Miss
Circle Hit Testing - Intuition

- On what basis can we say that the green point is inside the circle and the red point is not?

We can tell by looking at it, but computers cannot see.
Circle Hit Testing - Intuition

- On what basis can we say that the green point is inside the circle and the red point is not?

What if we cannot see the border?
Circle Hit Testing - Intuition

- On what basis can we say that the green point is inside the circle and the red point is not?

... but know the radius of the circle? (like computers)
Circle Hit Testing - Intuition

- On what basis can we say that the green point is inside the circle and the red point is not?

if distance from origin to the point is less than radius ...
Circle Hit Testing - Intuition

- On what basis can we say that the green point is inside the circle and the red point is not?
- If **distance** from mouse to circle center is less than *radius*, then circle is “hit” (e.g. rollover, clicked on, etc.)
- *radius* is the distance between circle centre to circle edge
Circle Hit Testing - Intuition

Hit

Miss
Distance Between Two Points

strokeWeight(10);
point(77, 81); // P
point(13, 45); // Q

// calculate distance between point P and Q
let d =

how to do this?
Pythagorean Theorem

\[ c^2 = a^2 + b^2 \]

\[ c = \sqrt{a^2 + b^2} \]
myDist

function to calculate distance between two points

use built-in function sqrt() for square root

// returns the distance between point P at (pX, pY)
// and point Q at (qX, qY)
function myDist(pX, pY, qX, qY) {
  let a = pY - qY; // y difference
  let b = pX - qX; // x difference
  let c = sqrt((a * a) + (b * b));
  return c;
}

Starter: https://editor.p5js.org/sanghosuh.sketches/8gbwZfsg
https://editor.p5js.org/sanghosuh.sketches/hJSqiQQA
circleHitTest

// returns true if pX, pY is inside a circle
// that is centred at cX, cY with radius r
function circleHitTest(pX, pY, cX, cY, r) {
    let d = myDist(pX, pY, cX, cY);
    if (d <= r) {
        return true;
    } else {
        return false;
    }
}
Processing has a *built-in* function for distance

- returns a Number

\[ \text{dist}(x_1, y_1, x_2, y_2) \]

Use this **built-in dist function** to calculate the distance between two points in your labs and assignments!
Mouse Speed

- now that we can use `dist()`, we can also use it to calculate `speed`
Speed

- If a person runs 100 m in 20 seconds, the speed is 5 m per second
  \((= \frac{100 \text{ m}}{20 \text{ sec}})\)

- If a car runs 120 km in 2 hour, the speed is 60 km per hour
  \((= \frac{120 \text{ km}}{2 \text{ hour}})\)

\[
\text{speed} = \frac{\text{distance}}{\text{time}}
\]
Mouse Speed - Example

draw() {
    frame 1
}

? pixels
(24, 56)

... ? pixels
(68, 56)

frame 2

speed = ? pixels
1 frame
Mouse Speed

- The distance between the current mouse position and the previous mouse position represents the speed of the mouse ("pixels per frame")
  - `pmouseX` and `pmouseY` are built-in global variables for the previous mouse position

```javascript
let speed = dist(mouseX, mouseY, pmouseX, pmouseY);
```
Mouse Speed - Example

draw() {
    (24, 56)
}

GLOBAL VARIABLE
pmouseX;
pmouseY;

frame 1
Mouse Speed - Example

```cpp
draw() {
  (24, 56)
}
```

frame 1

GLOBAL VARIABLE
pmouseX = 24
pmouseY = 56
Mouse Speed - Example

```
draw() {
    (24, 56)
}
```

```
draw() {
    ? pixels
    (68, 56)
}
```

GLOBAL VARIABLE
```
pmouseX = 24
pmouseY = 56
```
Mouse Speed - Example

draw() {
}

frame 1

let speed = dist(mouseX, mouseY, pmouseX, pmouseY);

draw() {

? pixels

(68, 56)

}

frame 2

GLOBAL VARIABLE

pmouseX = 24
pmouseY = 56
Mouse Speed - Example

GLOBAL VARIABLE
pmouseX = 24
pmouseY = 56

let speed = dist(68, 56, 24, 56);
Mouse Speed - Example

frame 1

draw() {
  (24, 56)
}

frame 2

draw() {
  ? pixels
  (68, 56)
}

let speed = 44;

GLOBAL VARIABLE
pmouseX = 24
pmouseY = 56

CS 105 - Functions 29
Mouse Speed - Example

```javascript
let speed = dist(68, 86, 68, 56);
```
Mouse Speed - Example

```javascript
let speed = 30;
```

GLOBAL VARIABLE

```javascript
pmouseX = 68
pmouseY = 56
```
Mouse Speed

- since we calculate it each time draw is called, the speed is in “pixels per frame”
- since frameRate is 60 frames per second, multiply by 60 to get “pixels per second”

\[
\text{speed} = \frac{3 \text{ pixels}}{1 \text{ frame}} \times \frac{60 \text{ frames}}{1 \text{ sec}} = 180 \text{ pixels per sec}
\]
let maxSpeed = 0;

function draw() {
    background(220); // grey

    let speed = dist(mouseX, mouseY, pmouseX, pmouseY);

    // update the max speed so far
    if (speed > maxSpeed) {
        maxSpeed = speed;
    }

    // display the max speed and current speed
    fill(0); // black
    text(round(maxSpeed), width / 2, height / 2 - 40);
    text(round(speed), width / 2, height / 2 + 40);
}

https://editor.p5js.org/cs105/sketches/dmaaz93v7
let speed = dist(mouseX, mouseY, pmouseX, pmouseY);

// draw an ellipse with size representing speed
fill(255, 128); // semi transparent white
ellipse(mouseX, mouseY, speed, speed);

// draw line with hue based on speed
colorMode(HSB);
let hue = map(speed, 0, 30, 200, 0);
strokeWeight(10);
stroke(hue, 100, 100);
line(pmouseX, pmouseY, mouseX, mouseY);

https://editor.p5js.org/cs105/sketches/LxaXWkTfa