Interaction

Animation
Events
Built-in Variables
Function Definition

Chapter 5, Examples 5-1, 5-2, 5-3, 5-4, 5.5, 5-6.
function setup() {
  createCanvas(100, 100);
  background(200); // gray
  strokeWeight(10);
  stroke(0, 255, 0); // green
  line(99, 0, 0, 99);
  stroke(200, 0, 200); // magenta
}
What Are Interactive Media?

- Dynamic: Animation
- Interactive: Responsive
What Are Interactive Media?

- Dynamic: Animation
- Interactive: Responsive
The Animation Process From 1938 (0:00 - 2:32)
https://www.youtube.com/watch?v=M2ORkIrHUbg
How do we animate?

- To animate, we need to ...
How do we animate?

- To animate, we need to ...
  - draw multiple figures
  - flip through the pages
How do we animate in Processing?

```java
function setup() {
    createCanvas(400, 400);
}

function draw() {
}
```
How do we animate *in Processing*?

runs 1st, runs **once**

runs 2nd, runs **continuously**

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

function draw() {
}
```
Frames Per Second (FPS) or FrameRate

- **Analogy**: how quickly we flip the pages
- Frame refers to an image
- The `draw()` runs 60 times per second (by default)

Credit: [https://aframe.com/blog/2013/07/a-beginners-guide-to-frame-rates/](https://aframe.com/blog/2013/07/a-beginners-guide-to-frame-rates/)
Frame rates for animation, 60fps, 30fps, and 15fps side by side. (0:14 - 0:50)
https://www.youtube.com/watch?v=npMreLeVD6o
Frames Per Second (FPS) or Frame Rate

- Used in ...
  - movies
  - animations
  - videos (slo-mo?)
  - computer graphics

Credit: https://www.filmindependent.org/blog/hacking-film-24-frames-per-second/
Recap

- showing **multiple frames (images) per second** creates perception of motion
- `draw()` allows us to do this by **drawing multiple frames per second**

```
function draw() {
}
```
draw

function setup() { }

function draw() { }
mouseX, mouseY

values change depending on the location of mouse

https://editor.p5js.org/sanghosuh/sketches/QjUjKTf4Q
function setup() { }
function draw() { }

Built-in variables:
mouseX, mouseY

Try background in draw() or setup()

https://editor.p5js.org/sanghosuh/sketches/ZigRBODra
https://editor.p5js.org/cs105/sketches/7wukIY_bg
(demos: errors)
calling functions “outside” a function block
wrong function definitions
Trace of setup and draw

- see: “05 Interaction (trace).pdf”
print() to trace program flow
(using frameCount, frameRate too)

function setup() {
  createCanvas(200, 200);
  frameRate(1); // set to 1 frame per second
  print("setup");
}

function draw() {
  background(230); // almost white
  fill(255, 0, 0); // red
  ellipse(mouseX, mouseY, 30, 30);
  print("draw frame " + frameCount);
}
What Are Interactive Media?

- Dynamic: Animation
- Interactive: Responsive
What Are Interactive Media?

- Dynamic: Animation
- Interactive: Responsive
What Are Interactive Media?

- Dynamic: Animation
- Interactive: Responsive
Event

CREDIT: http://www.opusa.org/
Event

1. **[Real-life]** An observable occurrence, phenomenon, or an extraordinary occurrence.

2. **[Computer]** A message to notify an application that something happened.

- Examples:
  - Keyboard Events (key press, key release)
  - Mouse Events (button press, button release, moved)
Setup and Draw are Events

- The setup event happens when the program is first run
  - `setup()` is a built-in function that represents this event
    (Processing calls `setup()` when the setup event occurs)
- The draw event happens 60 times per second (by default)
  - `draw()` is a built-in function that represents this event
    (Processing calls `draw()` when the draw event occurs)
- We “customize” `setup()` and `draw()` functions to do something when these events occur
mousePressed Event

- A mousePressed event happens when a mouse button is pressed (or a touch pad tapped)
  - `mousePressed()` is a built-in function that represents this event
  - Processing calls `mousePressed()` when the event occurs
- We define a `mousePressed()` function to “customize” it to do something when this event occurs
- (there’s also a mouseReleased event and corresponding function)

```javascript
function mousePressed() {
  ...
}
```
keyPressed Event

▪ A keyPressed event happens when a key is pressed
  - keyPressed() is a built-in function that represents this event
  - Processing calls keyPressed() when the event occurs
▪ We define a keyPressed() function to “customize” it to do something when this event occurs
▪ (there’s also a keyReleased() event and corresponding function)

function keyPressed() {
  ...
}

paint

keyPressed()
mousePressed()
mouseReleased()

https://editor.p5js.org/cs105/sketches/hOVVeyZ0h
Trace Through: setup() and draw()

// red dot

function setup() {
  createCanvas(200, 200);
  noStroke();
  fill(255, 0, 0); // red
}

function draw() {
  background(240); // almost white
  ellipse(mouseX, mouseY, 30, 30);
}
Trace Through: setup() and draw()

1 function setup() {
2   createCanvas(200, 200);
3   noStroke();
4   fill(255, 0, 0);  // red
5 }

6 function draw() {
7   background(240);  // almost white
8   ellipse(mouseX, mouseY, 30, 30);
9 }

wait for 16ms
function setup() {
  createCanvas(200, 200);
  noStroke();
  fill(255, 0, 0); // red
}

function draw() {
  background(240); // almost white
  ellipse(mouseX, mouseY, 30, 30);
}
Variables

- A symbolic name used to reference an “unknown” value. The value may change, but the symbolic name doesn’t.
- They are useful when creating interactive media.

- Kinds of Variables
  - built-in variables (e.g. mouseX, width)
  - constants (CENTER, BEVEL)
  - user defined variables (next class!)

[https://p5js.org/reference/](https://p5js.org/reference/)
Useful Processing Built-in Variables

mouseX, mouseY
frameCount

width, height

pmouseX, pmouseY
mouseIsPressed
mouseButton

keyIsPressed
key, keycode
.
.

https://p5js.org/reference/
function draw() {
  background(255);
  line(width / 2, height / 2,
       mouseX, mouseY);
}

function draw() {
  background(255);
  line(mouseX - 25, mouseY - 25,
       mouseX + 25, mouseY + 25);
}

https://editor.p5js.org/sanghosuh/sketches/c2zmBZS3a
function draw() {
    background(255);
    line(width / 2, height / 2,
        mouseX, mouseY);
}

function draw() {
    background(255);
    line(mouseX - 25, mouseY - 25,
        mouseX + 25, mouseY + 25);
}

https://editor.p5js.org/sanghosuh/sketches/XHeh9oZe4
mouseX

mouseY
mouseX - 25  mouseX  mouseX + 25
mouseY - 25
mouseY
mouseY + 25
using built-in variables in expressions

Exercises:

Start from hi drawing:
https://editor.p5js.org/cs105/sketches/uYIgbdkpS

1. Make “hi” follow the mouse
2. Make “hi” follow the mouse (with the mouse cursor always at the top point of the “h”)
(30, 20)
Using (Calling) Functions

- We *call* functions all the time in JavaScript ...

```javascript
createCanvas(100, 100);
```

https://p5js.org/reference/
Using (Calling) Functions

- Note that they may have **no**...
Using (Calling) Functions

- Note that they may have different...

```java
createCanvas(100, 100);
ellipse(75, 50, 23, 52);
```

# of parameters
Why Use Functions?

- To avoid tedious repetition (for convenience)
- For example...

```javascript
point(0, 0);
point(0.1, 0.1);
point(0.2, 0.2);
.
.
point(50, 50);
```

```javascript
line(0, 0, 50, 50);
```
Why Use Functions?

- To avoid tedious repetition (for convenience)
- For example...

```plaintext
line(0, 0, 5, 0);
line(5, 0, 5, 5);
line(5, 5, 0, 0);
triangle(0, 0, 5, 0, 5, 5);
```
**Why Use Functions?**

- To avoid tedious repetition (*for convenience*)
- For example...

```plaintext
line(0, 0, 50, 0);
line(50, 0, 50, 50);
line(50, 50, 0, 50);
line(0, 50, 0, 0);
rect(0, 0, 50, 50);
rect(0, 0, 50, 50);
?  
```

![Diagram of a rectangle made with lines and a rectangle using a rectangle function]
What Are Functions?

- Code that is “packaged” so it can be conveniently run “by name”
- Often performs some computation and returns a value (but not always)
Function Declaration

tell JavaScript “declare a function”

function name

function parameters

function setup() {

}
Function Declaration

tell JavaScript
“declare a function”

function name

function parameters

function draw() {

}

code block
What are the values of `mouseX` and `mouseY`?

```javascript
function setup() {
  createCanvas(100, 100);
}
```
Which image is drawn by this code after 10 frames?

```javascript
function setup() {
  background(200);
  strokeWeight(10);
  stroke(0, 255, 0); // green
}

function draw() {
  line(99, 0, 0, 99);
  stroke(200, 0, 200); // purple
}
```
Which image is drawn by this code … one second after the mouse button was pressed?

```javascript
function setup() {
  createCanvas(100, 100);
  background(255);
}

function draw() {
  rect(25, 25, 50, 50);
}

function mousePressed() {
  fill(0); // black
  ellipse(50, 50, 25, 25);
}
```