Media: Video and Sound

p5.dom and p5.sound libraries

Capturing Video

Computer Vision

Playing Movies and Sounds

Chapter 13, Examples 13-1, 13-2, 13-3, 13-4
Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - some examples of computer vision
Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - some examples of computer vision
What is a library?
What is a library?

- A collection of pre-written code (e.g., functions, variables) that offers a particular functionality
What is a library?

p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else! p5.js is free and open-source because we believe software, and the tools to learn it, should be accessible to everyone.

Using the metaphor of a sketch, p5.js has a full set of drawing functionality. However, you’re not limited to your drawing canvas. You can think of your whole browser page as your sketch, including HTML5 objects for text, input, video, webcam, and sound.

https://p5js.org/
p5.js supports a set of drawing functionality

https://p5js.org/reference/
Why use libraries?

- To avoid rewriting the same code from scratch
  - same reason as writing functions
Why use libraries?

- To avoid rewriting the same code from scratch
  - same reason as writing functions
- To extend or add to the p5.js core functionality

Libraries

Using a library

A p5.js library can be any JavaScript code that extends or adds to the p5.js core functionality. There are two categories of libraries. Core libraries (p5.sound) are part of the p5.js distribution, while contributed libraries are developed, owned, and maintained by members of the p5.js community.

To include a library in your sketch, link it into your HTML file, after you have linked in p5.js. An example HTML file might look like this:
Contributed p5 Libraries

- [https://p5js.org/libraries/](https://p5js.org/libraries/)

### Core Libraries
- **p5.sound**
  p5.sound extends p5 with Web Audio functionality including audio input, playback, analysis and synthesis. Created by: Jason Sigal

- **p5.accessibility**
  p5.accessibility makes the p5 canvas more accessible to people who are blind and visually impaired.

### Community Libraries
- **p5.asciitext**
  p5.asciitext is a simple and easy to use image-to-ASCII art converter for p5.js. Created by: Pawel Janicki

- **p5.ble**
  A Javascript library that enables communication between BLE devices and p5 sketches. Created by: Yining Shi, Jingwen Zhu, Tom Igoe

- **p5.collide2D**
  p5.collide2D provides tools for calculating collision detection for 2D geometry with p5.js. Created by: Ben Moren

- **p5.dimensions**
  p5.dimensions extends p5.js' vector functions to work in any number of dimensions. Created by: Smilebags, Max Segal

- **p5.bots**
  With p5.bots you can interact with your Arduino (or other microprocessor) from within the browser. Use sensor data to drive a sketch; use a sketch to drive LEDs, motors, and more! Created by: Sarah Groff-Palermo

- **p5.cmykjs**
  CMYK ColorSpace Created by: JT Nimoy

- **p5.createloop**
  Create animation loops with noise and GIF exports in one line of code. Created by: Peter Hayman

- **p5.geolocation**
  p5.geolocation provides techniques for acquiring, watching, calculating, and geofencing user locations for p5.js. Created by: Ben Moren

- **p5.EasyCam**
  Simple 3D camera control with inertial pan, zoom, and rotate. Major contributions by Thomas Diewald. Created by: jWilliam Dunn

- **p5.experience**
  Extensive library for p5.js that adds additional event-listening functionality for creating canvas based web applications. Created by: Felix Meichelböck

- **grafica.js**
  grafica.js lets you add simple to highly configurable 2D plots to your p5.js sketches. Created by: Javier Graciá Carpio
How to use libraries? (e.g., DOM library)

- libraries must be added ("imported") into your project

p5 web editor already includes “p5.dom” and “p5.sound” libraries, but you need to add them yourself in the Processing IDE.
How to use libraries? (e.g., DOM library)

- make sure to add a link to it

```html
<html>
<head>
  <script language="javascript" type="text/javascript" src="libraries/p5.min.js"></script>
  <script language="javascript" type="text/javascript" src="libraries/p5.dom.min.js"></script>
</head>
<body>
  <script src="sketch.js"></script>
</body>
```
How to use libraries? (e.g., DOM library)

- Every library has a reference explaining what it does & how to use it
- It offers HTML element generation functionality

https://p5js.org/reference/
Your Project is a Webpage

• Your p5.js script runs *inside* an HTML webpage
- HTML code brings together your script, p5.js, and p5.js libraries

```html
<html>
<head>
  <script language="javascript" type="text/javascript" src="libraries/p5.min.js"></script>
  <script language="javascript" type="text/javascript" src="libraries/p5.dom.min.js"></script>
</head>
<body>
  <script src="sketch.js"></script>
</body>
```
Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - some examples of computer vision
Today...

- We will see...
  - how to use libraries in p5.js
  - **how to use video & sound**
  - some examples of computer vision
Using p5.dom Library to Capture Live Video

0) Make sure the p5.dom library is in your project

1) Declare a global variable for a video Element object
   ```javascript
   let camera;
   ```

2) Create the video Element object in setup()
   ```javascript
   camera = createCapture(VIDEO);
   camera.size(width, height);
   ```

3) Use the video Element object like a p5 Image
   - Draw the current video frame on the canvas:
     ```javascript
     image(camera, 0, 0);
     ```
   - Get the colour at a pixel location:
     ```javascript
     camera.get(x, y);
     ```
   - Process the whole pixel array:
     ```javascript
     camera.pixels[i];
     ```
let camera;

void setup() {
    createCanvas(320, 240);
    camera = createCapture(VIDEO);
    camera.size(width, height);
    camera.hide();
}

function draw() {
    image(camera, 0, 0);
}
video-painterly

using .get(x, y) with video frame

```javascript
for (let i = 0; i < 100; i++) {
    let x = random(width);
    let y = random(height);
    stroke(camera.get(x, y));
    let s = random(1, 5);
    strokeWeight(s);
    point(x, y);
}
```

https://editor.p5js.org/sanghosuh/sketches/FZW9WdKdY

gets the colour of video pixel
// loop through all grid positions
for (let x = maxSize / 2; x < width; x += maxSize) {
    for (let y = maxSize / 2; y < height; y += maxSize) {

        // get the colour at the corresponding pixel
        let vx = floor(map(x, 0, width - 1, 0, camera.width - 1));
        let vy = floor(map(y, 0, height - 1, 0, camera.height - 1));
        let pixelColour = camera.get(vx, vy);

        // get brightness and convert it to a size
        let b = brightness(pixelColour);
        let s = map(b, 0, 100, 0, maxSize * 1);

        fill(255);
        ellipse(x, y, s, s);
    }
}
```javascript
image(camera, 0, 0);
loadPixels();
for (let x = 0; x < width; x++) {
  for (let y = 0; y < height; y++) {
    // index into pixels array
    let i = (x + y * width) * 4;
    // extract red, green, blue, alpha
    let r = pixels[i];
    let g = pixels[i + 1];
    let b = pixels[i + 2];
    let a = pixels[i + 3];
    // process pixel here ...
  }
}
```

https://editor.p5js.org/sanghosuh/sketches/BdS_uw6kr
Using p5.dom Library to Play Video

0) Make sure the “p5.dom” library is in your project

1) Add a video file to your sketch’s data folder.

2) Declare a global variable for a video Element object
   let movie;

3) Create the video Element object in preload()
   movie = createVideo("data/flyboard.mp4");

4) Start playback in setup()
   movie.play();

5) Use the video Element object like a p5 Image
   image(movie, 0, 0);

...
let movie;

function preload() {
    movie = createVideo("data/flyboard.mp4");
}

function setup() {
    createCanvas(640, 360);
    movie.size(width, height);
    movie.hide();
    movie.play();
}

function draw() {
    image(movie, 0, 0);
}

https://editor.p5js.org/cs105/sketches/7BZWvfccs
(detect if movie is done playing)

// check if video is over
if (movie.time() >= movie.duration()) {
  ...
}

returns the time of the current movie frame in seconds
returns the length of the whole movie in seconds

https://www.w3schools.com/tags/ref_av_dom.asp
if (movie.duration() > 0) {
    let t = map(mouseX, 0, width, 0, movie.duration());
    movie.time(t);
}

https://editor.p5js.org/cs105/sketches/ZbuoreaRW
Using the p5.sound Library to Play Sounds

0) Add a sound file to your sketch’s data folder.

1) Declare a global variable for the `SoundFile` object
   
   ```javascript
   let honk;
   ```

2) Load the sound file into the variable in `preload()`
   
   ```javascript
   honk = loadSound("data/honk.wav");
   ```

3) Play sound when you want
   
   ```javascript
   honk.play();
   ```

Starter: [https://editor.p5js.org/cs105/sketches/NyM_n_1QF](https://editor.p5js.org/cs105/sketches/NyM_n_1QF)
let honk;
let horn;

function preload() {
    // load sound files from data directory
    honk = loadSound("data/honk.wav");
    horn = loadSound("data/horn.wav");
}

function mousePressed() {
    if (mouseX < 50) {
        honk.play();
    } else {
        horn.play();
    }
}
Today...

- We will see...
  - how to use libraries in p5.js
  - **how to use video & sound**
  - some examples of computer vision
Today...

- We will see...
  - how to use libraries in p5.js
  - how to use video & sound
  - some examples of computer vision
Computer Vision

- "Computer vision" refers to a broad class of algorithms that allow computers to make intelligent assertions about digital images and video (Levin, 2006)

- Computers that can "see"
- Using the camera as a "sensor"
VIDEOPLACE Mini-documentary (1988)
- https://youtu.be/dmmxVA5xhuo?t=133
A Simple Computer Vision Algorithm

for each video frame:
  identify a "special" pixel
  based on some criteria
  use that pixel's location
  to control the computer

This is just a special kind of array operation, ...
very similar to finding the largest element in an array.
Array Operation: Find Largest Element Value

- Search the array to find the largest element value
  - Start by guessing that the largest value is the first element
  - check all other elements one-by-one to see if any are larger
  - if larger value is found, it becomes new largest value found so far

```javascript
// find the largest element value
// (assuming arr has at least length 1)
let largest = arr[0];
for (let i = 1; i < arr.length; i++) {
  if (arr[i] > largest) {
    largest = arr[i];
  }
}
print("largest:", largest);
```
let brightest = 0;
let brightestLoc = [0, 0];

for (let x = 0; x < width; x++) {
    for (let y = 0; y < height; y++) {

        let i = (x + y * width) * 4;
        let r = pixels[i];
        let g = pixels[i + 1];
        let b = pixels[i + 2];
        let bright = brightness(color(r, g, b));

        if (bright > brightest) {
            brightest = bright;
            brightestLoc = [x, y];
        }
    }
}
let best = 255 * 255 * 255;
let bestLoc = [0, 0];

for (let x = 0; x < width; x++) {
    for (let y = 0; y < height; y++) {
        let i = (x + y * width) * 4;
        let r = pixels[i];
        let g = pixels[i + 1];
        let b = pixels[i + 2];

        // t is the target colour to find best match
        let d = dist(red(t), green(t), blue(t), r, g, b);

        if (d < best) {
            best = d;
            bestLoc = [x, y];
        }
    }
}

https://editor.p5js.org/sanghosuh/sketches/mK3oxQuyB
let target;
let sensitivity = 30;

for (let x = 0; x < width; x++) {
    for (let y = 0; y < height; y++) {

        // t is the target colour to find best match
        let d = dist(red(t), green(t), blue(t), r, g, b);

        if (d < sensitivity) {
            pixels[i] = 255;
            pixels[i + 1] = 0;
            pixels[i + 2] = 0;
        }
    }
}
https://evaluate.uwaterloo.ca/