

# How to export your p5.js as a video

## How to Video Capture your p5.js Sketch

If you're creating a piece of time-based work in p5.js you may want to capture the canvas as a video. For example, if you have a generative design that evolves over time, or an animation sequence. In this tutorial you will learn how to:

1. Download and include a custom library to your p5.js sketch
2. Program that library to access your camera as a capture card.

### Custom Library

The first step we need to take is to download the custom library for this process. This library can be found at the following Moodle [link](#). Following that link will automatically download a **src.zip** folder.

make sure you sign into Moodle to access the files.

Once this folder is downloaded you can extract it to either your desktop or downloads folder (anywhere you can easily find it). We are going to upload the files from this folder to our sketch now.

make sure you turn off "auto-refresh" if you have it enabled as this could cause the video capture to crash as you're programming

### Your Sketch

1. There is a **red arrow ">"** button beneath the **play** and **stop** buttons. Click this to expand the menu which shows all your sketch files. You will need to create a new folder called **src** and upload all the separate files from the **src.zip** you downloaded at the beginning.

If you do not create a folder called **src** and upload the files into it then you may encounter errors with the next code block. You will see that my files are referened to the path **"./src/"**

2. Click on the "index.html" file in the side bar. We know need to add the following lines of code in the head of our html file. You should see the default p5.js scripts. Feel free to add these underneath.

```
<script src = "./src/CCapture.js"> </script>
<script src = "./src/webm-writer-0.2.0.js"></script>
<script src = "./src/download.js"></script>
```

If you don't add these scripts to the head of the html file you will not be able to use the functions. If you encounter errors that describe undefined functions with regard to the capture code then check here first.

3. Return to your "sketch.js" file where we can now add the following lines of code to the top of sketch as a global variable. This will create a new CCapture object that we can store in the variable "capture". We also set a captureLength variable. I set it to 60, which will capture 1 second of material at 60 frames per second.

```
let capture = new CCapture({
  frameRate: 60,
  format: "webm",
});

let captureLength = 60;
```

7. Go to your draw function and put this code at the beginning to start the capturing

```
if (frameCount == 1) {
  capturer.start();
}
```

This line of code is telling the sketch to start recording once the frameCount == 1. We do this so that the program can run the first frame at setup before beginning to capture. If you wanted to delay the recording by a certain amount of frames you can change the boolean check.

8. The last block of code in your draw function should be this. When your program reaches this point it is going to check if the frameCount is still below the captureLength, and if that is true, it will keep recording. The minute the frameCount is over the captureLength the code block is going to evaluate with the else clause which will end the capture and save the output.

```
if (frameCount < captureLength) {
  capturer.capture(canvas);
} else if (frameCount === captureLength) {
```

```
capturer.save();
capturer.stop();
}
```

## The Draw Function in Full

```
function draw() {
  //start capturing
  if (frameCount == 1) {
    capturer.start();
  }

  //start coding

  background(220);

  //stop capturing
  if (frameCount < captureLength) {
    capturer.capture(canvas);
  } else if (frameCount === captureLength) {
    capturer.save();
    capturer.stop();
  }
}
```

This is a link to a finished version if you're having problems with your own.

[\*\*Video Exporter Template\*\*](#)

---

Revision #5

Created 27 October 2022 14:05:45 by Michael-Jon Mizra

Updated 14 April 2023 13:00:29 by Michael-Jon Mizra