

Creative Coding

How to load external assets?

COD 207 - Week 09 Class →

Table of Contents

- 1. Creative Coding
- 2. Table of Contents
- 3. Loading External Media
- 4. preload() function
- 5. Practice in class!
- 6. Image Processing
- 7. Result
- 8. BREAK
- 9. Generative Collage
- 10. Case Studies on openProcessing
- 11. Some other case studies
- 12. Assignments

Loading External Media

 Images → jpg, gif, png

 Audio → mp3, ogg, wav

 Video → mp4, webp, ogv

preload() function

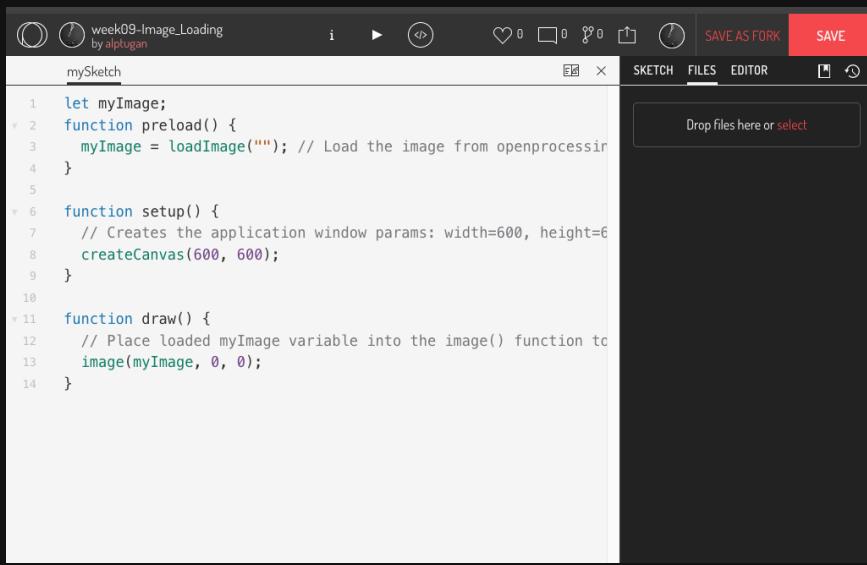
Images can be loaded and displayed to the screen at their actual size or any other size.

```
1 let myImage;
2 function preload() {
3     myImage = loadImage("image_name.jpg"); // Load the image from openprocessing into myImage variable
4 }
5
6 function setup() {
7     // Creates the application window params: width=600, height=600
8     createCanvas(600, 600);
9 }
10
11 function draw() {
12     // Place loaded myImage variable into the image() function to display it on the canvas
13     image(myImage, 0, 0);
14 }
```

Practice in class!

Before uploading your image, check its resolution and filesize. Large files are loaded slowly. Resize your image in photoeditor app of your choice before uploading. To upload your files;

1. Goto openprocessing
2. Open settings and goto "FILES" tab
3. Drag & Drop your image.
4. Example σ

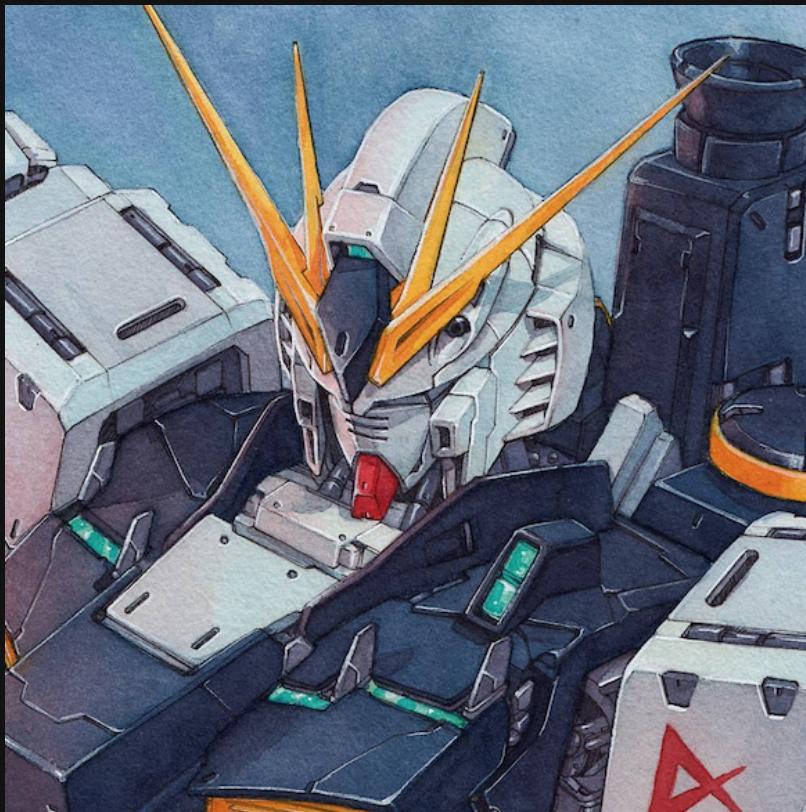


The screenshot shows the OpenProcessing sketch editor interface. The title bar reads "week09-Image_Loading by alptugan". The menu bar includes "SKETCH", "FILES", "EDITOR", and "SAVE" (which is highlighted in red). A sidebar on the right says "Drop files here or select". The main code area contains the following pseudocode:

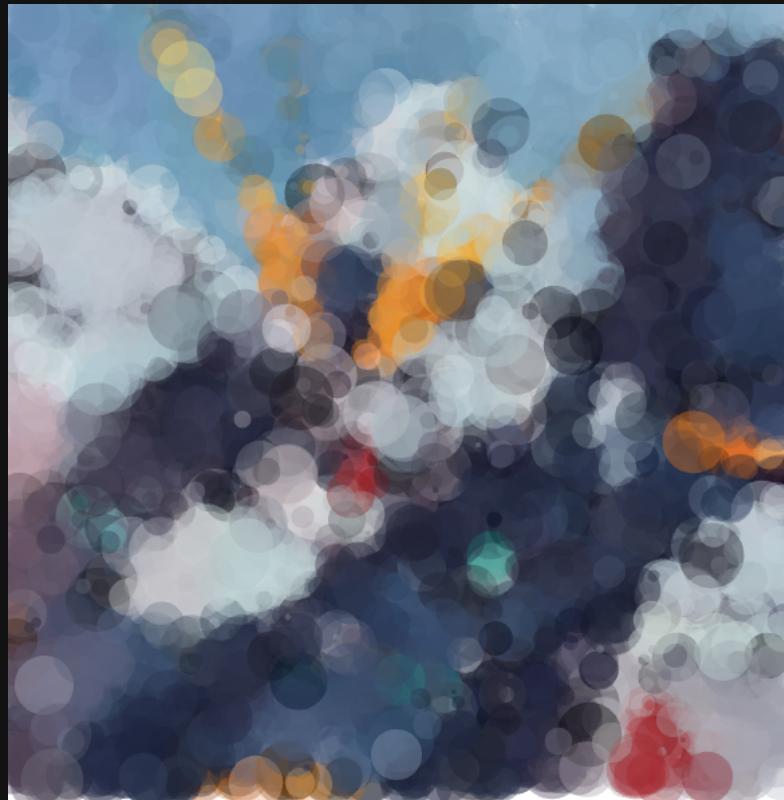
```
mySketch
1 let myImage;
2 function preload() {
3   myImage = loadImage("");
4 }
5
6 function setup() {
7   // Creates the application window params: width=600, height=600
8   createCanvas(600, 600);
9 }
10
11 function draw() {
12   // Place loaded myImage variable into the image() function to
13   image(myImage, 0, 0);
14 }
```

Image Processing

```
1 let myImage;  
2  
3 function preload() {  
4     myImage = loadImage("gundam.png"); // Load the image  
5 }  
6  
7 function setup() {  
8     // Creates the application window params: width=600  
9     createCanvas(600, 600);  
10 }  
11  
12 function draw() {  
13     // Place loaded myImage variable into the image() f  
14     //image(myImage, 0, 0, width, height);  
15     noStroke();  
16     let x = floor(random(0, myImage.width));  
17     let y = floor(random(0, myImage.height));  
18     let pix = myImage.get(x, y); // returns the pixel c  
19  
20     fill(red(pix), green(pix), blue(pix), 100); // set  
21     circle(x, y, random(3, 50));  
22 }
```



Result





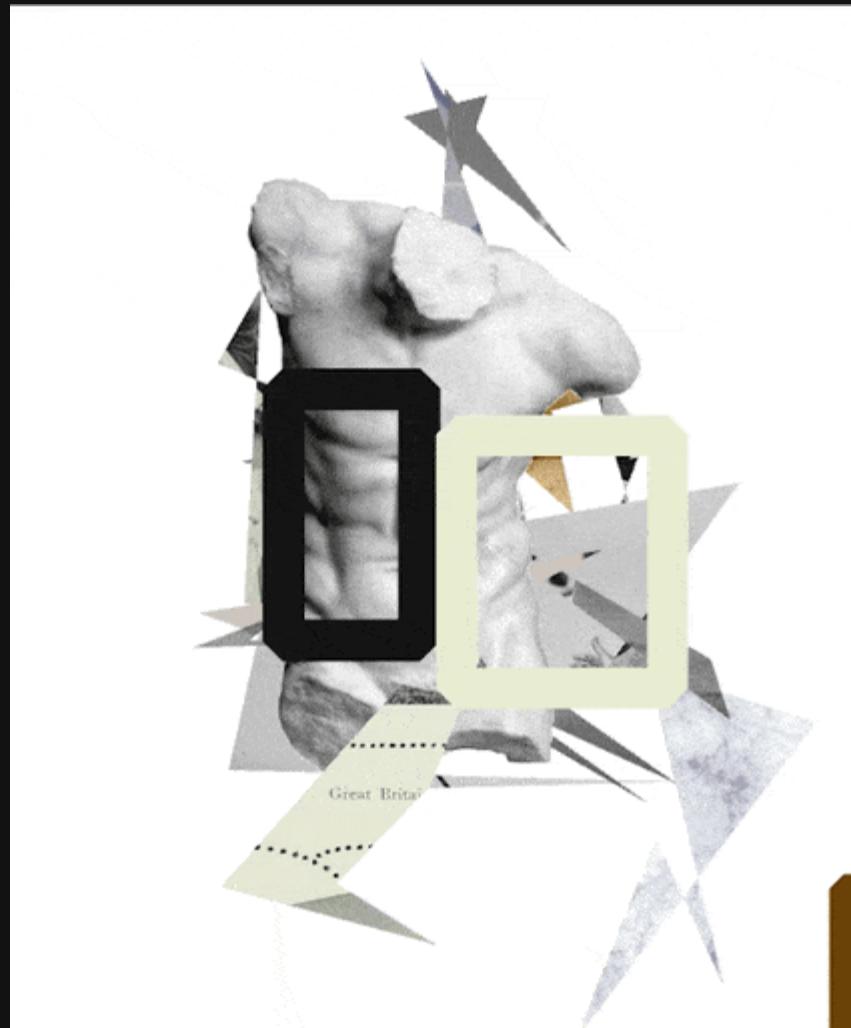
BREAK

10 mins.

Generative Collage

1. Choose a concept → "analog devices, black & white, space..."
2. Collect at least 10 images
3. Resize and crop them according to your canvas size.
4. Save all of the images as transparent png files.
5. Upload to the openprocessing
6. Use randomness (A weighted decision-approach controls the flow of elements.)

image source:  Patrick Hüebner



Case Studies on openProcessing

1. Load multiple images using Arrays
2. Display all of the images in random positions
3. Rotate, set opacity, scale up and down images algorithmically
4. Mirro Images, play with blendModes
5. Some trigonometry

Some other case studies

Following examples are developed using p5JS editor.

[Load Image from URL \(grabs ai generated images from a website\)](#)

[Pixelate the image](#)

[Mosaic Like Image](#)

[Diagonally mirrored image](#)

[Image Processing 1](#)

[Image Processing 2](#)

[Use Image as Texture](#)

Assignments

1. Collect at least 20 different images.
2. Crop and resize them in Photoshop. Save as transparent PNG.
3. Create a generative collage in photoshop (Goto Pinterest for inspiration).
4. Implement your design in p5JS and randomize the design whenever the user pressed r key on keyboard. 3 0 PTS
5. Submit 5 different variations of your sketch as jpg. Do not submit the code. 2 0 PTS
6. ! Submit the openprocessing link. 5 PTS
7. ! Submit the sketch source code as zip file as well. 5 PTS