



# Creative Coding

Image Processing

COD 207 - Week 07 Class →



# Table of Contents

- 1. Creative Coding
- 2. Table of Contents
- 3. Algorithmic Art: Image Processing
- 4. Example: Basics
- 5. Example: Paint with custom brush
- 6. BREAK
- 7. Add Interactivity
- 8. Built-in Functions: Keyboard
- 9. Built-in Functions: Mouse
- 10. GUI via DOM Manupulation
- 11. Exemplified Case Study
- 12. Case Studies for Inspiration
- 13. Download The CheatSheet
- 14. Assignments

# Algorithmic Art: Image Processing

link to ALAP Database



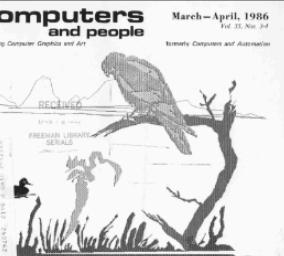
Kamran Moojedi

Stephen Hawking portrait



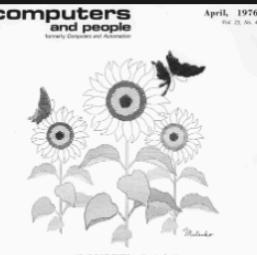
Kamran Moojedi

Pollock



Meelan Leong

Parrot With Duck in Landscape



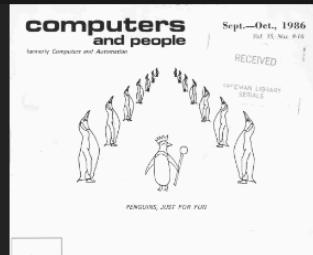
Mutsuko Sasaki

Hello Sunflowers



Virginia Hines

Leaves & Birds & Sun Design



Diana Wong

Penguins, Just For Fun

# Example: Basics

Load the image, replace the pixel data with shapes



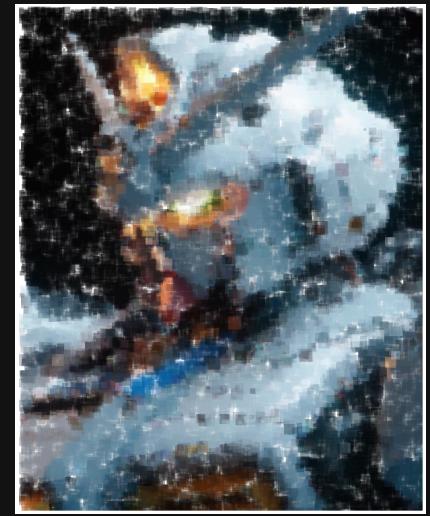
1. Load image



2. Swap pixels with  
circle()



3. Increase the circle()  
using random(min, max)



4. Swap pixels with rect()  
using random(min, max)

link to source code

# Example: Paint with custom brush

Load the image, replace the pixel data with another image.



Load image

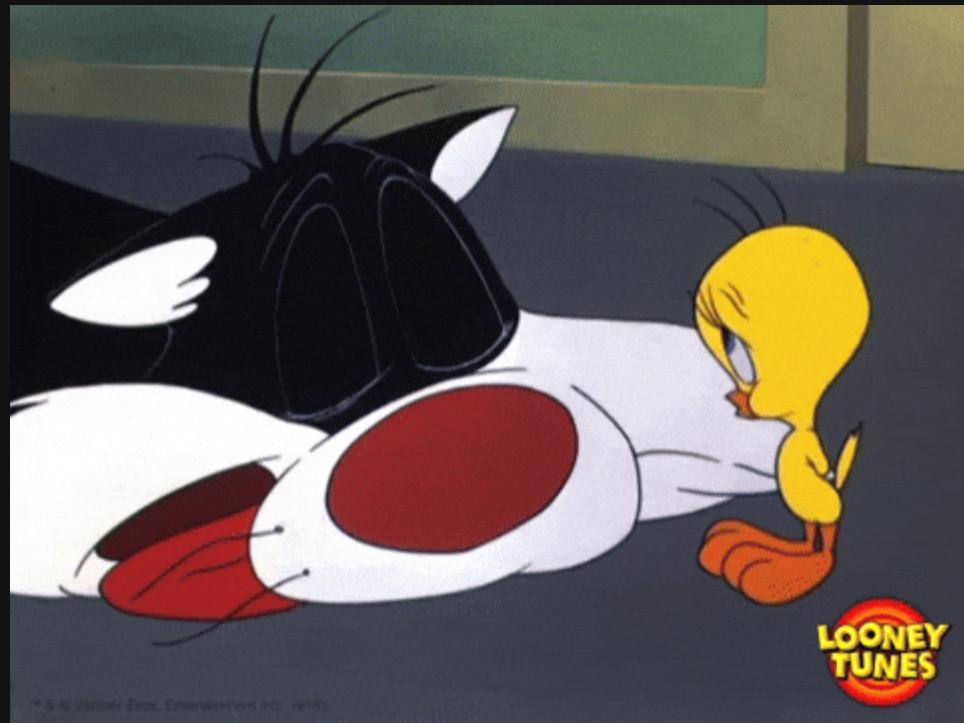


Create a brush in Photoshop in pure white color



Replace pixels with the brush image. Use `tint()` to set color

link to source code 



# BREAK

10 mins.

# Add Interactivity

# Built-in Functions: Keyboard

```
1  function setup() {  
2      // Creates the application window params: width=600, height=600  
3      createCanvas(600, 600);  
4  
5  }  
6  
7  function draw() {  
8      // Set the background color of the window params: Red: 100, Green: 20, Blue: 20  
9      // R,G,B values must be between 0 - 255  
10     background(100,20,20);  
11  }  
12  
13 // key is built-in variable. You don't need to declare on top of your code.  
14 function keyPressed() {  
15     if(key == 'r') {  
16         // set the background to a random color  
17     }  
18 }
```

# Built-in Functions: Mouse

```
function setup() {  
  // Creates the application window params: width=600, height=600  
  createCanvas(600, 600);  
  ^  
}  
  
function draw() {  
  // Set the background color of the window params: Red: 100, Green: 20, Blue: 20  
  // R,G,B values must be between 0 - 255  
  background(100,20,20);  
  circle(width/2,height/2,300);  
}  
  
// this function will be triggered whenever the user clicks on mouse or touchpad  
function mousePressed() {  
  // Increase the size of the circle whenever the user clicks on mouse or touchpad  
}
```

## DOM

**changed()**  
Calls a function when the element changes.

**createA()**  
Creates an element that links to another web page.

**createAudio()**  
Creates a hidden element for simple audio playback.

**createButton()**  
Creates a element.

**createCapture()**  
Creates a element that "captures" the audio/video stream from the webcam and microphone.

**createElement()**  
Creates a checkbox element.

**createColorPicker()**  
Creates a color picker element.

**createDiv()**  
Creates a element.

**createElement()**  
Creates a new p5.Element object.

**createFileInput()**  
Creates an element of type 'file'.

**createImg()**  
Creates an element that can appear outside of the canvas.

**createInput()**  
Creates a text element.

**createP()**  
Creates a element.

**createRadio()**  
Creates a radio button element.

**createSelect()**  
Creates a dropdown menu element.

**createSlider()**  
Creates a slider element.

**createSpan()**  
Creates a element.

**createVideo()**  
Creates a element for simple audio/video playback.

**input()**  
Calls a function when the element receives input.

**removeElements()**  
Removes all elements created by p5.js, including any event handlers.

**select()**  
Searches the page for the first element that matches the given CSS selector string.

**selectAll()**  
Searches the page for all elements that matches the given CSS selector string.

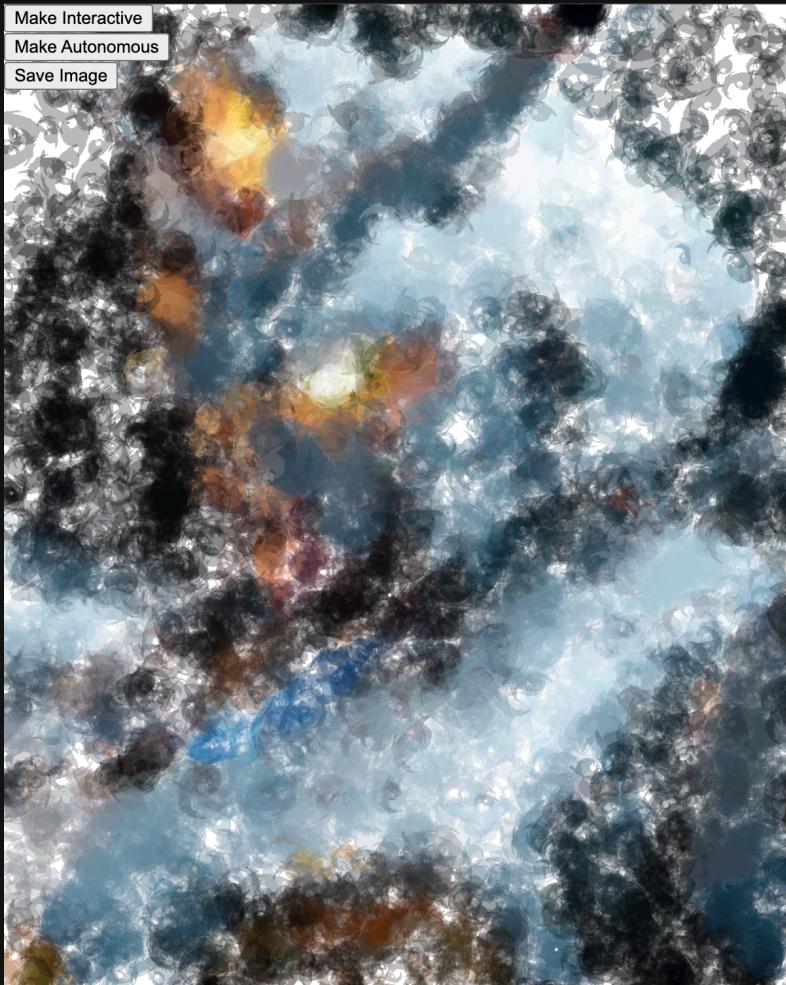
# GUI via DOM Manipulation

You can create HTML elements to make your application interactive. Play around with other elements to create and change your application parameters such as slider, toggle, div elements, etc...

🔗 [createButton\(\) reference](#)

🔗 [Available DOM Element in p5Js](#)

Make Interactive  
Make Autonomous  
Save Image



# Exemplified Case Study

In this version you can see html `<button>` elements created and utilized using p5JS to interactively switch between autonomous and interactive modes.

🔗 [Source Code](#)

# Case Studies for Inspiration

Following examples are developed using p5JS editor.

[Pixelate the image](#)

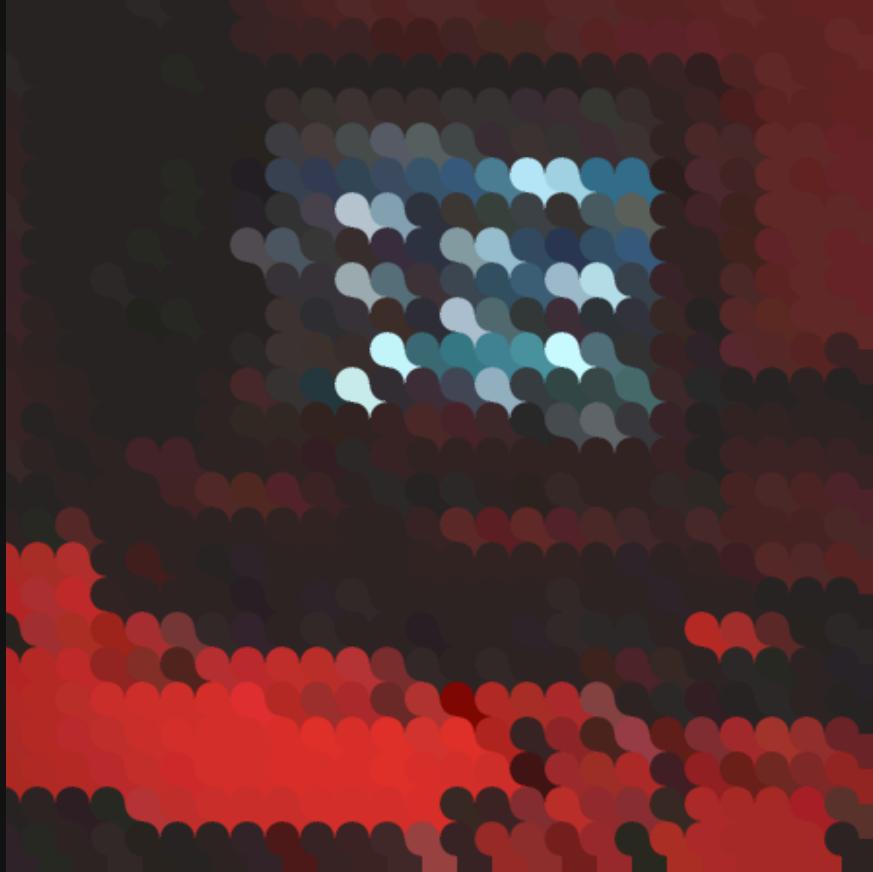
[Mosaic Like Image](#)

[Diagonally mirrored image](#)

[Image Processing 1](#)

[Image Processing 2](#)

[Use Image as Texture](#)



# Download The CheatSheet

Cheatsheets / Learn p5.js



## Interaction

### mouseX and mouseY

The `mouseX` and `mouseY` variables always store the current x and y coordinates of the mouse relative to the origin of the canvas. So if the mouse was currently at the x position of 150 pixels and the y position of 200 pixels, the value of the `mouseX` variable would be 150 and the value of the `mouseY` variable would be 200.

```
function draw() {  
    // The ellipse's x and y positions  
    // follow the mouse  
    ellipse(mouseX, mouseY, 100, 100);  
}
```

### The mouseIsPressed Variable

`mouseIsPressed` is a built-in boolean variable that is `true` when the mouse button is pressed, and `false` when it is not pressed. The `mouseIsPressed` variable is commonly used in `if` statements to perform actions based on whether the mouse button has been pressed or not.

```
// Draws ellipse if mouse is pressed  
if (mouseIsPressed) {  
    ellipse(200, 200, 180, 180);  
}
```

Download The CheatSheet

# Assignments

1. Choose a portrait image of someone (E.g super hero, philosopher, artist/actrist, any real or imagery individual)
2. Review the code templates in the presentation as a source of inspiration.
3. Create an image processing work.
4. Implement your design in p5JS and modify different parameters whenever the user keyboard, mouse, or Html GUI elements. PTS
5. Submit 3 different variations of your sketch as jpg. Do not submit 3 different code. PTS
6. Submit the openprocessing link. PTS
7. Submit the sketch source code as zip file as well. PTS

⚠ Warning

DO NOT JUST CHANGE THE IMAGE IN ONE OF THE CODE TEMPLATES ABOVE AND SUBMIT. YOU ARE EXPECTED TO MODIFY THE CODE.