introduction to media computing
Assignment 01

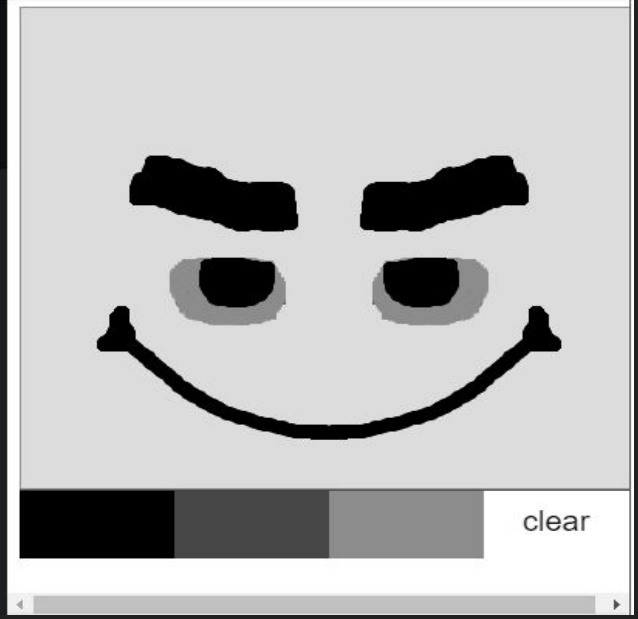
Example skeleton code

JS

```
//  GLOBAL variables
let active_brush = 0; //  Active Brush INDEX
let active_color = 0; //  Active Color Index

let color_0 = 0; //  Brush Color #0, BLACK
let color_1 = 70; //  Brush Color #1, GRAY
let color_2 = 140; //  Brush Color #2, LIGHT
GRAY

let back_color = 220; //  Drawing Area
Background color

function setup() {
  createCanvas(400, 400);
  background(back_color);
  noStroke();
  textSize(20);

  //  Draw the menu bar (we only draw once)
  //  Area #0, a BLACK area
  fill(color_0);
  rect(0, 350, 100, 50);

  //  Area #1, a DARK GRAY area
  fill(color_1);
```

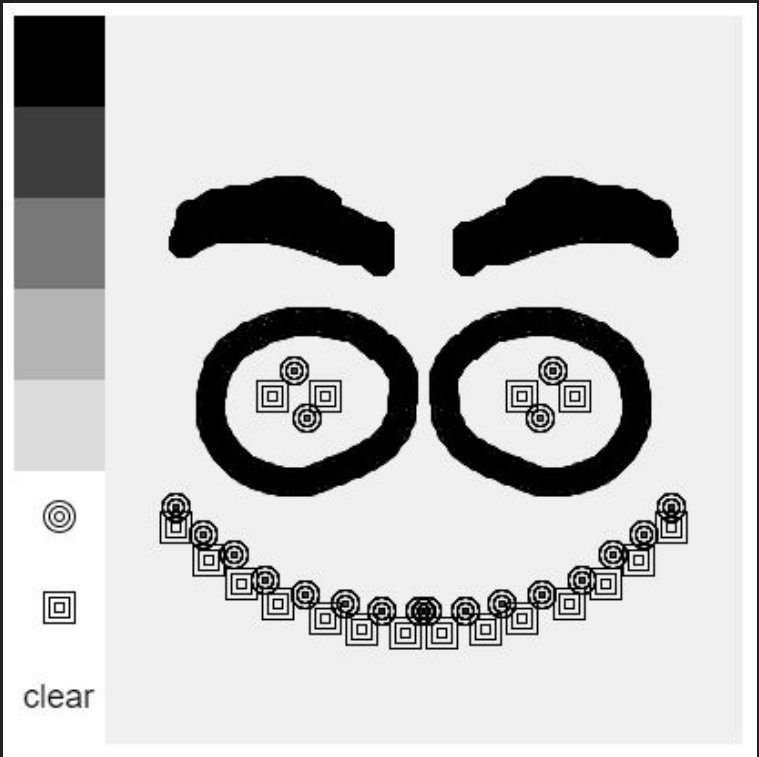
Result

EDIT ON CODEPEN



A simplified skeleton sketch as shown is provided as a basic framework for you to start coding your own nano painting app.

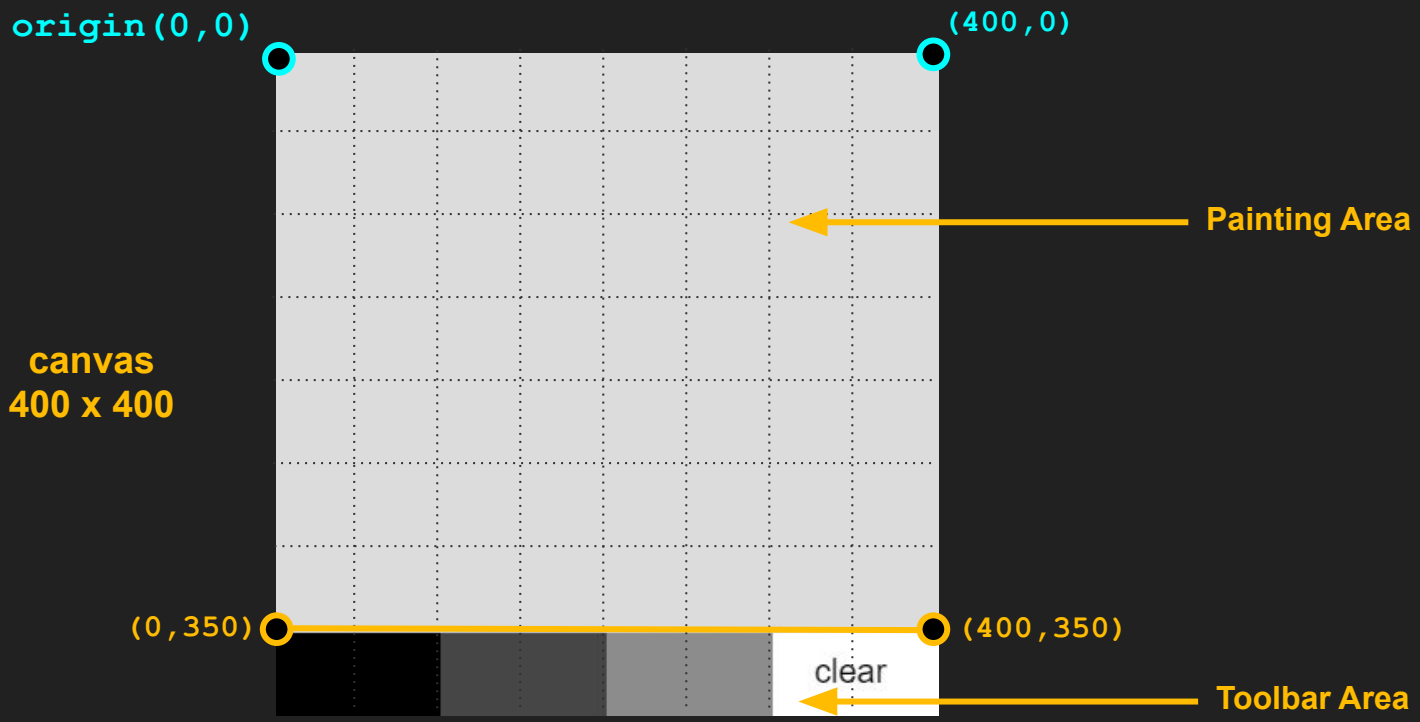
Your Assignment



BASIC REQUIREMENTS

1. Use a 400 x 400 canvas
2. TOOLBAR A vertical toolbar on the LEFT with at least 4 colors and 2 brushes to choose from.
3. BRUSHES must be visualized on the toolbar, and EACH brush uses a `while()` loop to draw (you may design your own brushes).
4. Symmetric painting behaviour similar to the one in the sample app.
5. All code must be well formatted and FULLY documented.

Example skeleton



a1_skeleton.html code structure

Line 6 - 14
Global variables declaration and initialization.

activeColor and activeBrush are variables to store currently active (user chosen) color (and brush).

color0, color1, and color2 allow us to redefine brush colors quickly.

backColor background color

```
let active_brush = 0;
...

function setup() {
  // draw toolbar at the bottom
}

function draw() {
  if (mouseIsPressed) {
      if (mouseY > 350) {
         // toolbar clicks
      else {
         // drawing
      }
  }
}
```

a1_skeleton.html code structure

Line 16 - 42

Draws the user interface ONCE.

```
let active_brush = 0;
...

function setup() {
  // draw toolbar at the bottom
}

function draw() {
  if (mouseIsPressed) {
      if (mouseY > 350) {
         // toolbar clicks
      else {
         // drawing
      }
  }
}
```

a1_skeleton.html code structure

`mouseY > 350`

lines 52 - 63

Classifies where the user is clicking to decide 'which color' the user has chosen, and stores the choice as a number into `activeBrush` and `activeColor`.

lines 64-68

The user is clicking on the 'clear' button, so it overpaints the drawing area using a rectangle using the `backColor`.

```
let active_brush = 0;
...

function setup() {
  // draw toolbar at the bottom
}

function draw() {
  if (mouseIsPressed) {
      if (mouseY > 350) {
          // toolbar clicks
      else {
          // drawing
      }
  }
}
```

a1_skeleton.html code structure

lines 71 - 95

Draws rectangle or ellipse according to the value of `activeBrush` and `activeColor` using if-else blocks

```
let active_brush = 0;
...

function setup() {
  // draw toolbar at the bottom
}

function draw() {
  if (mouseIsPressed) {
      if (mouseY > 350) {
         // toolbar clicks
      else {
         // drawing
      }
  }
}
```

Submission:

Your submission sketch must be saved as a self-contained HTML file (ready to run in a browser). Name your .html file `imc_a1_<SIS_ID>.html`, and then zip it into a ZIP archive.

Name your .zip archive as `imc_a1_<student_no>.zip`

`Example: imc_a1_55123456.zip`

Submit your `.zip` via Canvas on or before Sept. 30 (Mon) 23:59