

PV259

Generative Design Programming

Week 8

Image Processing

MUNI
FI

Megi Kejstová & Marko Řeháček
makej@mail.muni.cz, rehacek@mail.muni.cz

HW presentations

→

Dead presidents

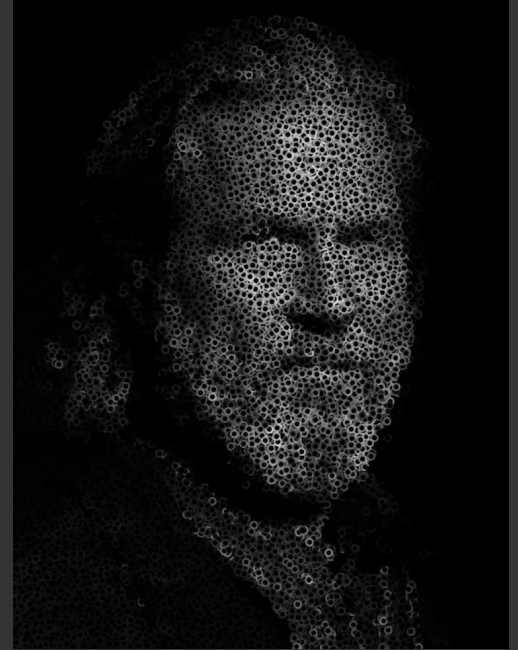
Mike Bronobjerg



→

Processing portraits

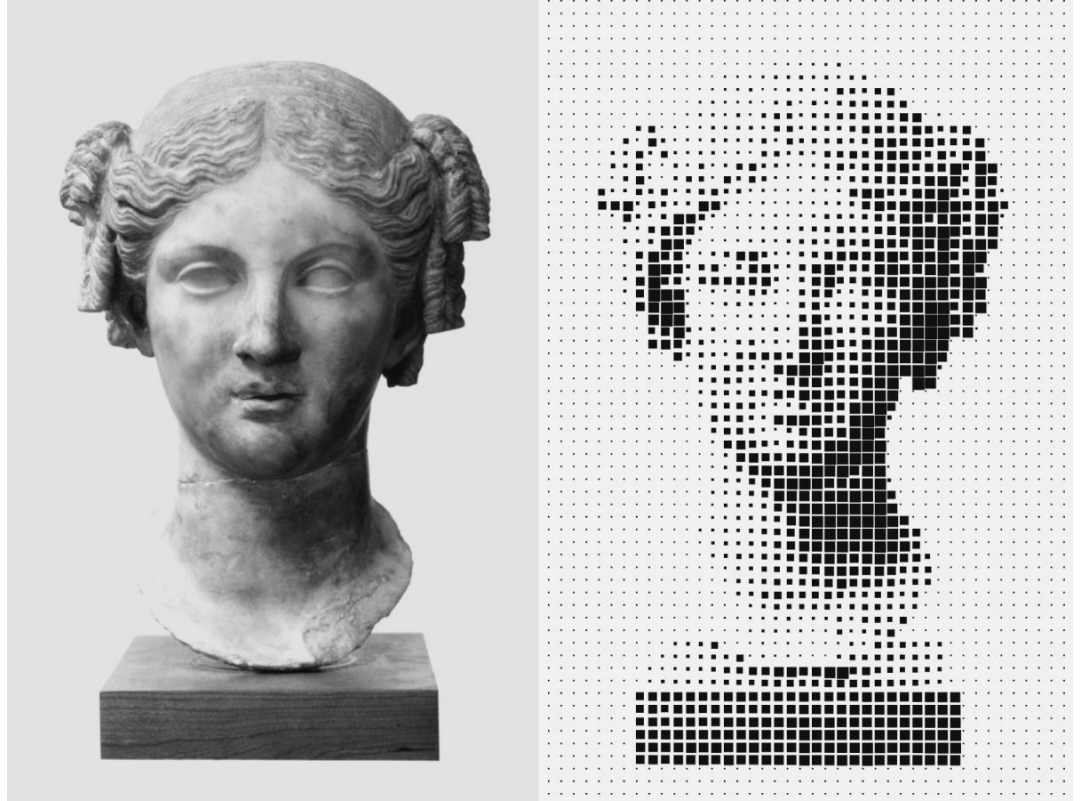
Unknown



→

Bauhaus 101

Tim Roderbröker

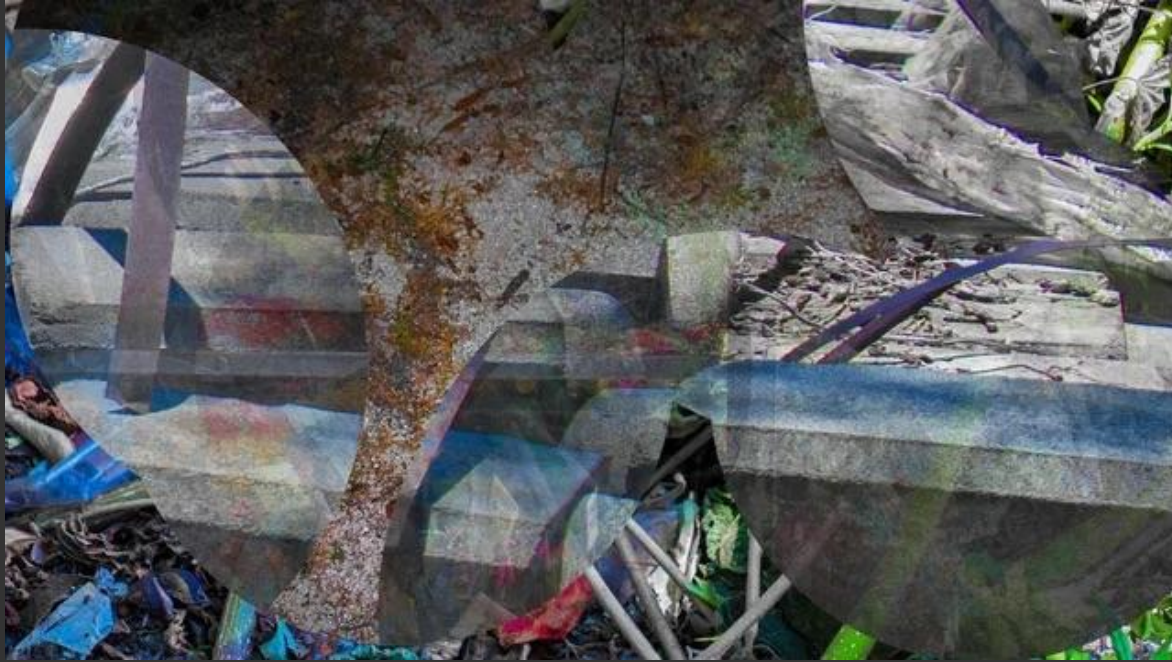


→

No name

Ra'afat





Basics of image processing

Pixel array

List containing all the pixels displayed on the screen

- RGBA (red, blue, green, alpha) for each pixel
 - each pixel described with 4 values in the array
 - range from 0 to 255

```
pixels = [p1_R, p1_G, p1_B, p1_A, p2_R, p2_G, p2_B, p2_A, ... ]
```

Pixel array

List containing all the pixels displayed on the screen

- RGBA (red, blue, green, alpha) for each pixel
 - each pixel described with 4 values in the array
 - range from 0 to 255

```
pixels = [p1_R, p1_G, p1_B, p1_A, p2_R, p2_G, p2_B, p2_A, ... ]
```

```
pixels = [0, 0, 0, 255, 255, 255, 255, 255, ... ]
```

Preparation of pixels

Pixel density - to ensure, that there is only one pixel per pixel

```
pixelDensity(1);
```

Loading pixels - uploading pixels of an image into pixel array

```
img = loadImage('imagePath')
```

```
image(img, x, y);
```

x-coordinate of top-left corner of the image

y-coordinate of top-left corner of the image

It is convenient to load image in the `preload()` function

Navigating through pixel array

before using pixels array

```
img.loadPixels()
```

double **for loop** to go through every column and every row of your screen

```
var index = (x + y * width)*4;
    pointing on first value of the pixel (value of red)
    red → img.pixels[index];
    green → img.pixels[index + 1];
    blue → img.pixels[index + 2];
    alpha → img.pixels[index + 3];
```

index position of red value
in the array

0	4	8	12	16	20	24	28
32	36	40	44	48	52	56	60
64	68	72	76	80	84	88	92
96	...						

If you make any change in the pixels, you need to update image in order to see them

```
img.updatePixels()
```

Useful p5.image functions

```
img.get(x,y)
```

→ returns whole pixel, you can use then red(), green(), blue()

```
img.width and img.height
```

```
img.set(x, y, color);
```

Filters

p5 filter function

changing the appearance of an image by altering the colors of the pixels

`filter`(PARAMETER)

- THRESHOLD
- INVERT
- BLUR

<https://p5js.org/reference/#/p5/filter>

using only `filter`(PARAMETER) will filter the canvas

using `img.filter`(PARAMETER) will change the img and its pixels

Example

->

Custom filters

Filters are a sequence of mathematical operations on pixels, based on either their position or color

Can be used as an image effect (imitation of an oil painting) or when image preprocessing is needed (edge detection, blur)



Example

->

Portrait

Find an image of a portrait/animal/object and remove the background

<https://www.remove.bg>

Create an abstract version of that image
using shapes, flow fields, lines, etc.

Glitch

Create glitch + maybe some interesting input for the glitch?

Like the automata?

Next week - bring headphones, microphone