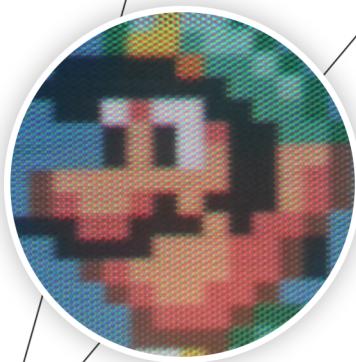


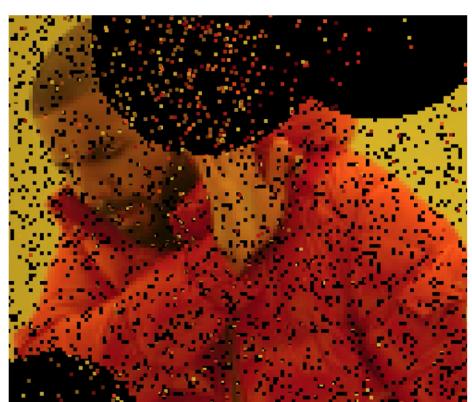
Creative Image Transition

Creative Coding Assignment 01



Inspiration

I was inspired by the low pixel portrait from the second assignment, which reminded me of CRT TVs with lower quality.



Artistic, Creative Related Decisions

As display technology is advanced nowadays, it is always uncomfortable and unpleasant to watch some relatively low-resolution videos.

Thus, my artistic and creative decision for this image transition assignment is to change the original picture into a low-quality picture like it's on the CRT monitor, and then the picture is broken pixels by pixels, and finally, another high-resolution picture is displayed.

Implementation

In order to achieve the effect mentioned on the first page, the screen transition is divided into 4 parts.

The first is to set up for the coming parts. Then, the second is to use particle effects to create the effect of breaking the screen. The third is to use the scattered pixel particles to recompose the second picture. The last is to make the recomposed picture transition to a clear version.

01 Setup

Since the next two parts of the image transition effects involve particle movement, a particle object class is created to store their properties, and then an ArrayList is also created to control particle movement.

In order to achieve the low pixel effect, a global variable called particleSize is created, which is used to set the density of each pixel.

In order to load images and use them for sampling, two PImage variables - "original" and "endImage" are created. Then, a for loop is wrote to scan the "original" and "endImage" according to the value of "particleSize", and stores the value of each scanned pixel in the particle object using a constructor.

Dist(pixelX, pixelY, mouseX, mouseY)



02 Image Breaking

In the second part, In order to make the effect of this image transition more interesting. Two modes are provided.

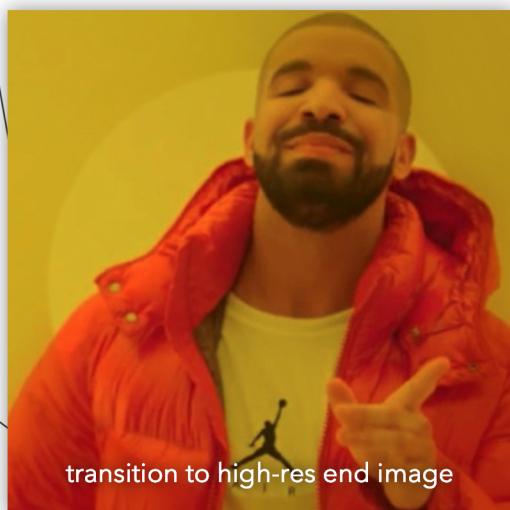
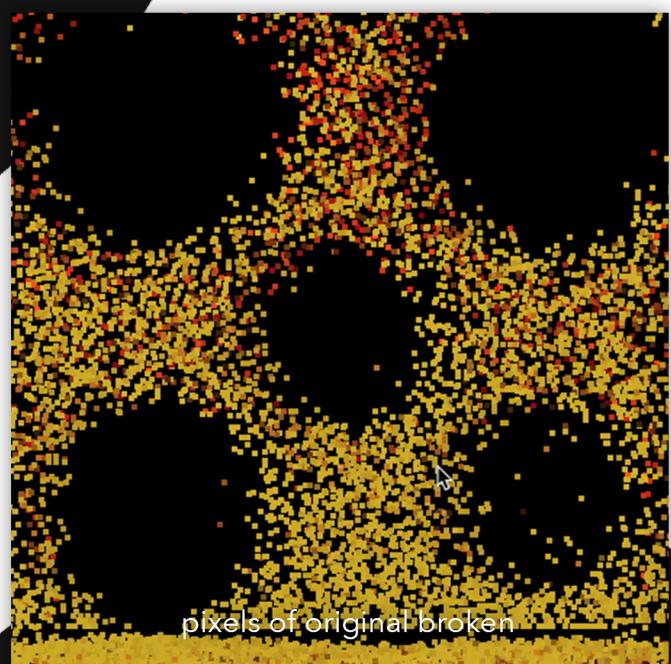
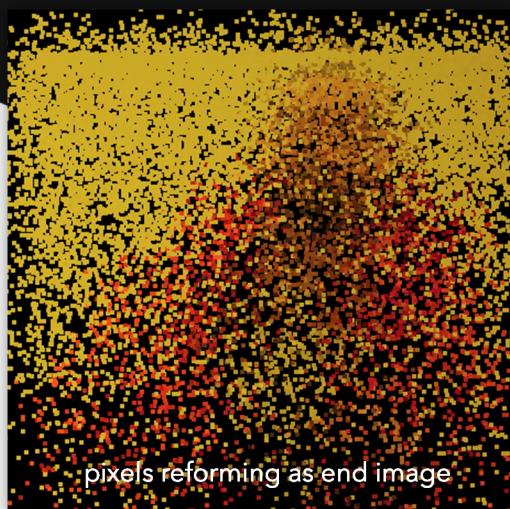
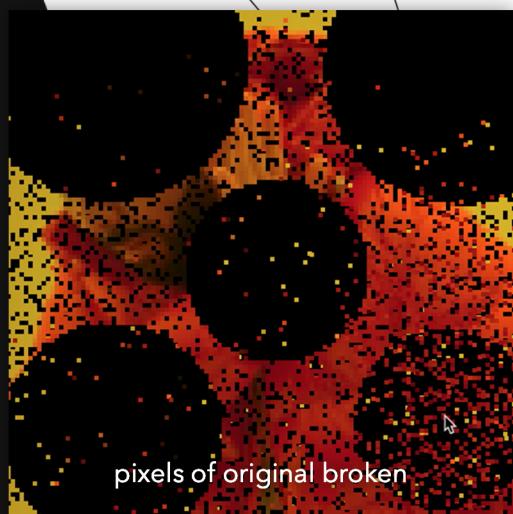
By modifying the global variable "mouseInteract", user can decide whether to randomly select the broken position or click it manually. A for loop and a boolean trigger are wrote to detect the distance between the pixel, the broken position and the timing of the trigger. If trigger is equal to true, the circular position within the distance will be broken and the scattered pixels will leave the screen.

03 Image Reform

In the third part, the pixel particles that were scattered and dropped outside the screen are returning to the canvas to form another image.

In order to achieve this effect, a function which uses to disrupt the original ArrayList and rearrange them randomly is written. Each particle object has x, y, endX, and endY fields which are the current and end position. In the setting part, the particle will be assigned the x, y position of the original position, and then use the function to disrupt pixel particles, and then assign new positions to particles.

In this way, the particles can be broken down and returned to a new position to form a second image.



04 Transition to End Image

In the final part, since the reformed image in the previous part is also in low-pixel, a transition animation is needed to transition it from low-pixel to high-pixel. In order to achieve this effect, a PGraphic "imageMask" is created. A for loop is used to generate multiple circles in the middle of the "imageMask". The size of the circles will be enlarged by the offset variable over time, and finally disappear out off the screen, forming a transition effect like ripples.

