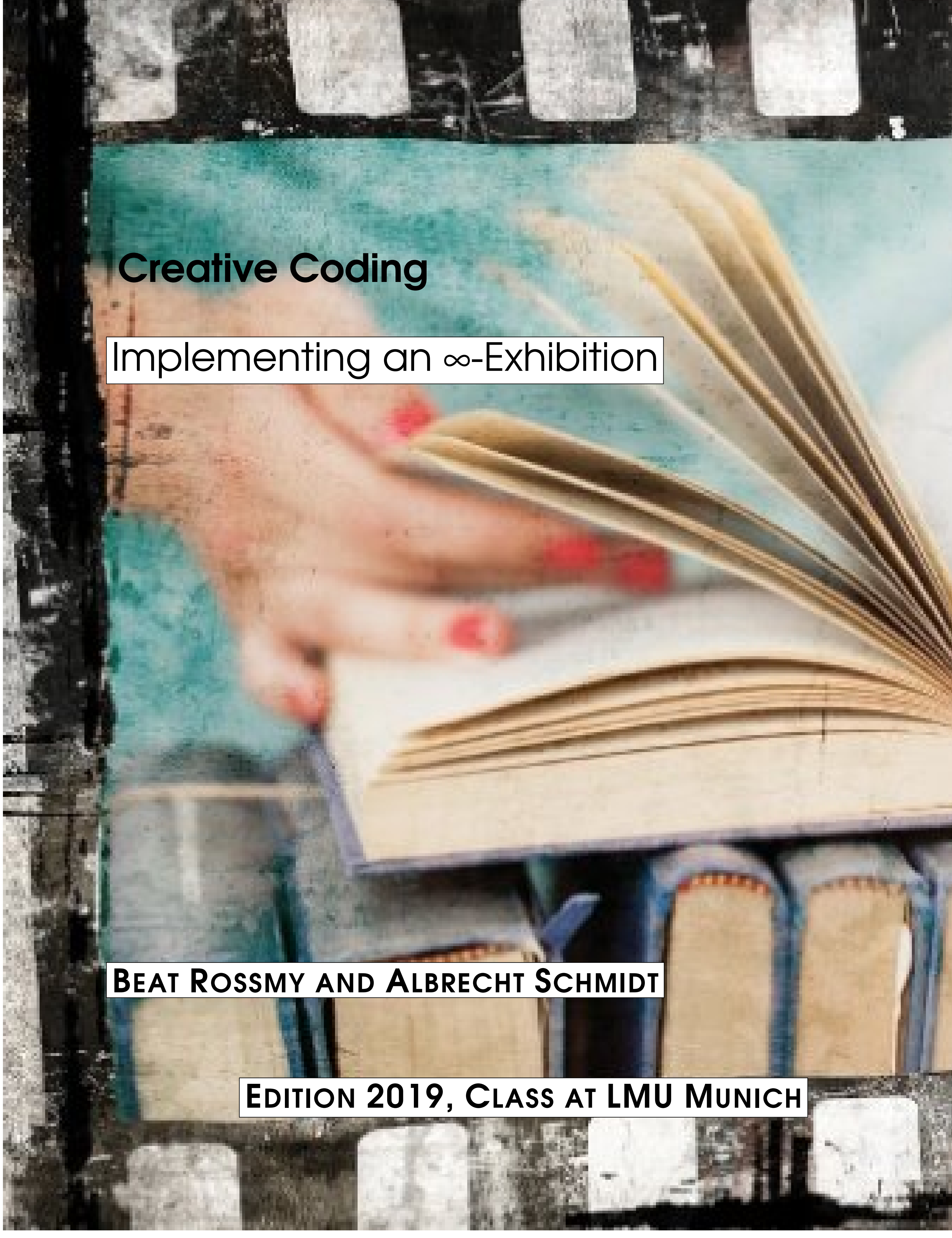

Creative Coding

Implementing an ∞-Exhibition

Beat Rossmy and Albrecht Schmidt

Edition 2019, Class at LMU Munich

Creative Coding

Implementing an ∞-Exhibition

BEAT ROSSMY AND ALBRECHT SCHMIDT

EDITION 2019

THIS BOOK IS WORK IN PROGRESS AND BASED ON A CLASS AT LMU MUNICH

Table of Contents

1. Programming and Creative Coding

1.1 Concept of this Book

The concept of the book is to get people excited for programming artistic pieces. In this book we teach basic concepts that make up building blocks for interactive and potentially artistic programs. The focus is to help the reader to create creative software and to learn how to program such software. The artistic part is up to the reader, but we hope to give good starting points, where readers can bring in their creativity.

In parts we suggest critical reflection on creative coding and programming art pieces. In the lecture at LMU this is done as discussions, that are not captured in the book. We recommend readers to show their ideas and programs to other and discuss and defend it. This aids reflection and learning, similar to what we do in the course.

Besides the programming concepts and the implementation techniques, we discuss different topics, including:

- How aesthetics can be programmed?
- Creativity vs. determined by algorithms?
- Creative information visualization vs. art?
- Evolutionary systems and creativity?
- ...

1.2 The Goal: an ∞-Exhibition

In the course we create different interactive software programs. With the selection of the topics we try to cover many different programming concepts and implementation techniques.

All the programs are designed to create changing new outputs, either after some time or based on user interaction. All programs together can then be set-up as an exhibition that is ever changing, which we call an ∞-Exhibition. If the screens with the software are set-up in a circle, where screes face outwards, a visitor to exhibition can walk around an will always see new exhibits.

1.3 Target Group and Expected Skills

The course and book is designed for readers studying at the intersection of art, media, and technology. We expect that readers have an interest in interactive art and that they want to improve their abilities to create interactive installations.

We expect that readers have basic programming skills in processing. If readers have no programming skills at all, we suggest to do an introductory tutorial (e.g. `https://processing.org/tutorials/`). For readers with programming experience in another languages, we expect that they can follow the course and read up on processing details as they go along.

2. Aesthetics and Information

2.1 Programming a Random Mondrian

Implement a program that creates a drawing that is inspired by Mondrian.

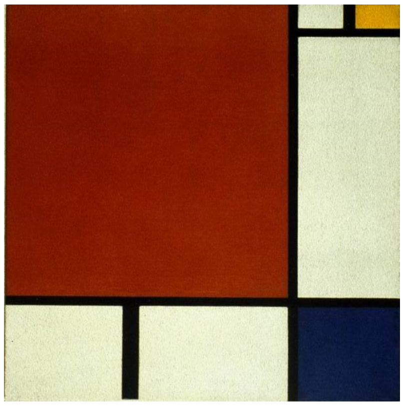

Figure 2.1: Example of a Mondrian Painting.

The program should create a new picture every 120 seconds or when the user presses a button on the keyboard.

Useful Commands

```
void keyPressed();
random();
randomSeed();
second();
minute();
hour();
```

The following listing is a starting point for your project.

```
size(400,400); // draving canvas, coordinate systems
background(255); // background is white

fill(255, 0, 0); // color is red
rect(50, 100, 250, 100);

fill(0, 255, 0); // color is green
rect(250, 250, 100, 100);

fill(0, 0, 255); // color is blue
rect(50, 320, 150, 10);

fill(255, 255, 0); // color is yellow
rect(50, 340, 150, 6);

fill(255, 255, 255); // color is white
rect(50, 360, 150, 4);

fill(0, 0, 0); // color is black
rect(50, 380, 150, 3);
```

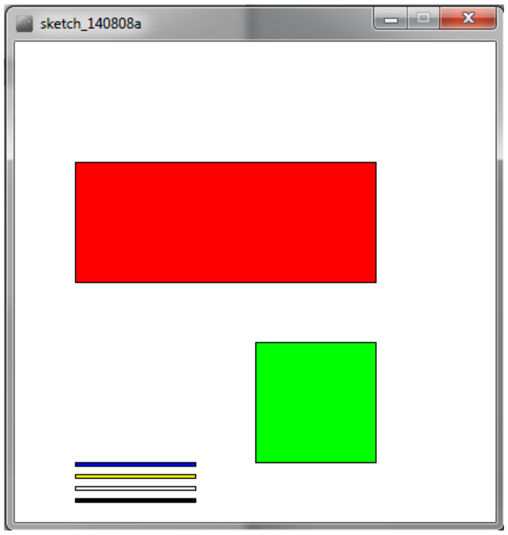

Figure 2.2: Resulting Image from the example code.

See `https://processing.org/reference/random_.html` for examples how to use the random function.

2.2 Random Creation and Aesthetics

Experiment with the parameters in the random creation. Explore the impact of parameters such as the size, dimension, line width, colors, and number of element on the aesthetics of the resulting picture.

Why are some pictures are aesthetically more pleasing than others? How can you change your algorithm to always create aesthetic drawings, while still having a great variety?

2.3 Encoding Information in Art

People have discussed the idea of informative art, see [Fer07; HS03; RSH00]. Can this utilitarian approach to creating visual displays based on know artworks be considered art?

Create a display that shows the time. Take a well know artwork as inspiration and encode the time in your visualization (a 15 minutes resolution is sufficient). To a casual observer it should not be apparent that the visualization shows the time.

What are the difficulties in designing such a visual display?

The following listing shows a basic clock. You can use this as a starting point for your own project.

```
// Variabeln definieren - ausserhalb von setup() und draw()
// da sie in beiden Funktionen verwendet werden
int stunde, minute, sekunde;

void setup() {
// Fenstergroesse festlegen
   size(260, 100);

   // Variabeln mit 0 initialisieren
   stunde = 0;
   minute = 0;
   sekunde = 0;

   // die Funktion draw() wird 2x pro Sekunde ausgefuehrt
   frameRate(2);
}

void draw() {
   // loesche den Hintergrund: setze ihn schwarz20background(0);
   // setze Zeichenfarbe fuer Text und Recheck: weiss
   fill(255, 255, 255);

   // speichere in den Variablen die aktuelle Zeit
   stunde = hour();
   minute = minute();
   sekunde = second();

   // Kontrollausgabe der Zeit in der Console - kann spaeter
       geloeschtwerden
   // in processing kann man mit "+" einen String aus Buchstaben und
       Zahlenbauen
   println(stunde + ":" + minute + ":" + sekunde);

   // zeige die Zeit im Fenster an
   textSize(32);
   text(stunde + ":" + minute + ":" + sekunde, 10, 30);

   // zeichne einen Sekundenbalken
   // Sekunde hat einen wert von 0 bis 59
   // Sekunde multipliziert mit 4 hat einen Wert zwischen 0 und 236
   rect(10, 40, sekunde*4, 20);
}
```

Ther is another example of a analog clock that may be helpful: `https://processing.org/examples/clock.html`

2.4 Tasks - 1st lecture

In the 1st session we try to mimic paintings of the famous dutch painter Piet Mondrian. He is famous for his abstract paintings that are reduced to a minimal palette of colors and geometric structures. His systematic approach invites to be recreated with algorithms.

Preparation

If Processing isn't already installed on your computer, go to `https://processing.org/download/` download the program and install it on your machine. Open some example sketches and familiarize yourself with the Processing IDE. If you feel comfortable, proceed with the following task.

Breakout Session

During the following task we try to approach the generation of random Mondrianesque images in a systematic way. Step by step we go from the analysis of real and fake Mondrians, over theoretical concepts for the description of Mondrianesque structures, to working implementations of the algorithm.

- Search on the internet for pictures of the painter Mondrian (see Figure 2.3). What are the connecting commonalities? Can you spot artworks not painted by Mondrian? Try to figure out the "recipe" that makes a distinguishable Mondrian.
- Let us discuss your approach to generate Mondrianesque images with the help of an algorithm. Try to sketch such structures and formalize them first.
- Try to implement one of the approaches we discussed in the plenum. Does it lead to the expected result? Does it work as simply as expected or have there been unexpected complications?
- If your algorithm is working, try to extend it to add your own style to the generated "Mondrians". Tweak parameters, change the color palette or even introduce more randomness.

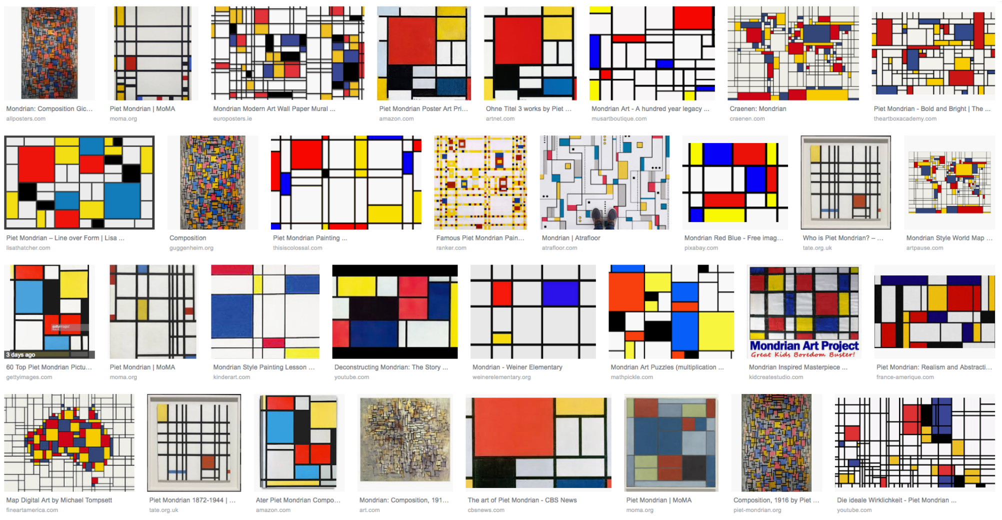

Figure 2.3: Artworks painted by Mondrian and Mondrian-inspired artworks.

Discussion

We just created algorithms that mimic the style of a famous artist. Let us discuss your impressions and thoughts along the way. As a starting point think about the following questions:

- Can you call the product of these algorithms "art"?
- Can art in general be scripted? (Pollock, Warhol, ...)
- Does the used technology influence such a differentiation? (Style transfer performed with neural networks vs classic "conditional" algorithms)

Homework

1. **Reading:** Read the following papers [Fer07; HS03; RSH00] and write a short (300-500 words) essay on Informative Art. Your essay should contain an explanation what it is, why people would want to do it, pros and cons.
2. **Implementation:** Create a display that shows the time. Take Mondrian algorithm and encode the time in your visualization (a 15 minutes resolution is sufficient).
3. Collect feedback on your implementation. Record 3 videos from people seeing your implementation (15 seconds each).

Bibliographie

[Fer07] Alois Ferscha. “Informative art display metaphors”. In: *International Conference on Universal Access in Human-Computer Interaction*. Springer. 2007, pages 82–92 (cited on pages 10, 12).

[HS03] Lars Erik Holmquist and Tobias Skog. “Informative art: information visualization in everyday environments”. In: *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. ACM. 2003, pages 229–235 (cited on pages 10, 12).

[RSH00] Johan Redström, Tobias Skog, and Lars Hallnäs. “Informative art: using amplified artworks as information displays”. In: *Proceedings of DARE 2000 on Designing augmented reality environments*. ACM. 2000, pages 103–114 (cited on pages 10, 12).