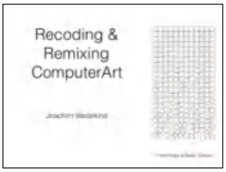
Recoding & Remixing ComputerArt

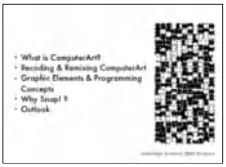

Recoding & Remixing ComputerArt

Joachim Wedekind

Mes dames et messieurs, je suis très heureux de vous présenter aujourd'hui mon petit projet pour la programmation de l'art informatique. Mais je peux vous rassurer: le reste de mon report je vais tenir dans la langue de la conférence …

My first slide is an old slide. This was the last slide of my Teach-Meet-Contribution at the Scratch2015 conference in Amsterdam. There I was so careless to announce what I was <u>planning</u> to do about ComputerArt. Today I would like to tell you what I <u>have done</u> so far.

I have to notice in advance that I am not a computer scientist and I am not an artist. I'm just a retired educational technologist with a growing interest in ComputerArt and its programming. And in the meantime I have learned so much about both thematic areas that I decided to share my experiences.


- What is ComputerArt?
- Recoding & Remixing ComputerArt
- Graphic Elements & Programming Concepts
- Why Snap! ?
- Outlook

In my presentation you will hear about:

What is ComputerArt?
Why Recoding & Remixing ComputerArt
What are the Graphic Elements & Programming Concepts - and a Toolbox

I will finish with some remarks **why I have chosen Snap!** and of course also this presentation will have a last slide **Outlook**, announcing what I am planning from now on …

BTW, nearly all graphics shown in this presentation are recodings of typical works of the pioneers of early ComputerArt and as such a tribute (homage) to those pioneers**.**

What is Computerart?

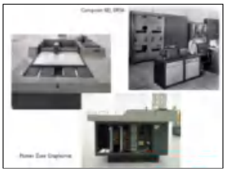
- What is ComputerArt?

Computer art is any art in which computers play a role in production or display of the artwork.

[Wikipedia, 2017]

This is the definition from wikipedia: *Computer art is any art in which computers play a role in production or display of the artwork.*

The definition applies, on the one hand, to the software, that are the programs with which the data to be displayed are generated, on the other hand, the hardware, that is the computers on which the programs are running and the output devices with which the data are displayed as perceptible artifacts.



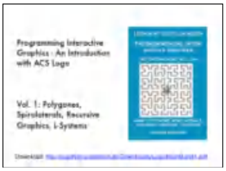
In our context, however, it is almost more important that ComputerArt is also a direction, a genre in art, that lasted from 1965 to about 1980. The software ran on mainframe computers and the results were output on plotters. For example the Zuse Graphomat Z 64 was the plotter of choice for the German computer artists.

After a short flowering period of the said 15 years, ComputerArt disappeared again from the scene. For some years, however, interest has reawakened and its role as precursor and pioneer of modern media art, characterized by multimediality and interactivity, is now recognized.

Why did I start Recoding & Remixing ComputerArt?

At my retirement I decided to pick up two old hobbies again: to do programming and to deal with art.

As an educational technologist I got to know very early the programming language Logo. I did my first programming with it using MIT Logo for the Apple II, BTW an already amazingly powerful version of Logo.


Programming Interactive Graphics : An Introduction with ACS Logo
Vol. 1: Polygones, Spirolaterals, Recursive Graphics, L-Systems
Download:

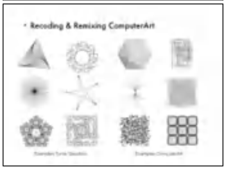
For my Logo-revival I was looking for a suitable actual Logo-version and found it with ACS Logo for the Apple Macintosh. The first result of my efforts was a booklet, Vol. 1 on Programming Interactive Graphics. You can download it from my website, but I have to warn you: it is written in German.

I have stopped Vol. 2 only half-finished, because now the art came into play.

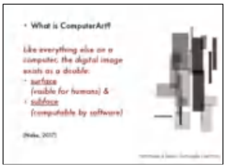
At that time my daughter studied and did her master's thesis at Frieder Nake in Bremen. As she told me, he was one of the pioneers of the so-called ComputerArt (which I had only known casually before) and so I got to know the pictures from this period of art history. These are some examples.

• Recoding & Remixing ComputerArt

I found the similarity of these images to typical pictures made with the turtle graphic of Logo quite amazing. I thought, to reprogram early ComputerArt with Logo, should actually be relatively simple - and so my interest in programming merged with my interest in art.


• Recoding & Remixing ComputerArt

The just mentioned Frieder Nake has characterized this twofold approach in brief and to the point:


• What is ComputerArt?

Like everything else on a computer, the digital image exists as a double:
• surface (visible for humans) &
• subface (computable by software)

(Nake, 2017)

The digital image exists as a double: the surface, which is visible for humans and the subface, which is computable by software.

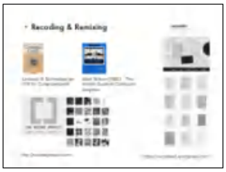
My project to recode and remix such pictures can be considered as a trial to treat the surface and the subface of the digital images equally important.

Of course I am not the first nor the only one to recode and to remix ComputerArt or other art genres. Shortly after the end of the first flowering period of ComputerArt, some books appeared, intended to introduce into the reconstruction of this art direction. For example, Schneeberger used a FORTRAN-library to produce his graphics, Wilson used BASIC, which was available on most microcomputers at that time.

• Recoding & Remixing

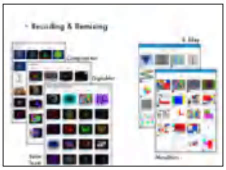
The latest efforts to make the historical works of ComputerArt accessible can be found on the net.

The ReCode Project has set itself the task "to preserve computer art by translating it into a modern programming language“ (which is Processing).


• Recoding & Remixing

The project recodeArt concentrates in a similar way on the work of Reiner Schneeberger. Here too, the implementation is carried out with Processing.

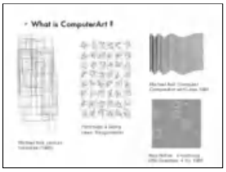
Both projects contribute to the preservation of digital art.



Also in the Explore-section on the Scratch Website you can find various relevant collections, for example studios on Computer Art, on Digital Art or Selim Tezel's Art Projects, as well as collections of projects with Mondrian-like or Riley-like paintings.

What is ComputerArt?

Ok, the definition of ComputerArt has already been introduced. Now its time to name some typical properties.



My procedure was as follows: First I tried to find as many examples as possible. This was not always easy because the documentation of ComputerArt is very sparse and it is seldom object of a museum's collection.

What I did with my still growing collection was as follows:

- The analysis, which patterns and structures are characteristically for the images,
- the compilation of a compendium of the graphic elements used (like points, lines or more complex structures),
- the analysis of whether and which recurrent patterns are recognizable,
- and the analysis of random deviations of forms, number of elements and the arrangement of objects.

By the way, this analytical approach has changed my view of the works of modern art as a whole: in the stylistic directions of concept art, suprematism, and especially geometric abstraction, very often series of recurring patterns and structures are found, tempting to reconstruct them algorithmically.

You have seen some examples of ComputerArt already on the preceding slides. On this slide there are further very typical examples by some of the pioneers: Michael Noll, Georg Nees and Vera Molnar.

Graphic Elements

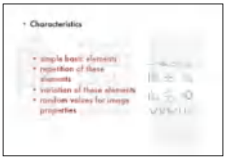
They illustrate almost completely the list of typical graphical elements which can be found in early ComputerArt:

- Graphic Elements
  - black & white
  - lines & polygonal chains
  - line hatching
  - squares, polygons
  - circles, ellipses
  - (sine) curves

- black & white (to plot with colors was awkward to handle at that time)
- the use of lines as well as line hatching
- The use of squares, polygons, circles, and curves

Characteristics

There is also a set of structural characteristics of those works:

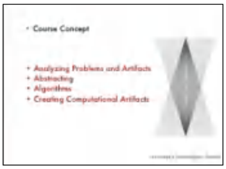


- Characteristics
  - simple basic elements
  - repetition of these elements
  - variation of these elements
  - random values for image properties

- the use of the named simple basic elements
- the repetition of these elements
- the variation of these elements
- and random values for selected image properties

Course Concept

The next step was to develop a syllabus or rather a <u>course</u> based on those graphical elements and structural characteristics and to combine them with Computational Thinking Practices, like



- Course Concept
  - Analyzing Problems and Artifacts
  - Abstracting
  - Algorithms
  - Creating Computational Artifacts

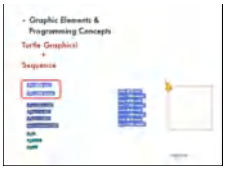
- Analyzing Problems and Artifacts (that is in our case the analysis of examples of early ComputerArt)
- Abstracting (that is what we did with the description of structural characteristics of early ComputerArt)
- Algorithms (which we have to formulate for the recoding of an example)
- Creating Computational Artifacts (that is to translate the algorithms into programs which produce the desired graphics)

I have tried to put all this together and the result is (or will be, to be more precisely) the book „Coded Art". I am a great advocate of visual coding, therefor the language of my choice was Snap! (but probably 95% of my examples can also be realized with Scratch).

- Course Concept

I have to insert at this point that my addressees are - not only but mainly - older people. Not least because of that I have placed great value on introducing all language elements in the case they are required for realizing a concrete project and thus avoiding „learning ahead“. At the end I was astonished myself that this was possible from the early beginning.
To illustrate that, I want to show you the first exemplary steps:

Graphic Elements & Programming Concepts



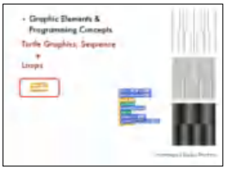
- Graphic Elements & Programming Concepts

Turtle Graphics!

+

Sequence

In all projects graphics will be produced. Of course we need as an indispensable basis a set of turtle graphics commands. Fortunately these are very intuitive with its natural geometry. And of course it is necessary to bring the commands in a meaningful order of steps (sequencing).



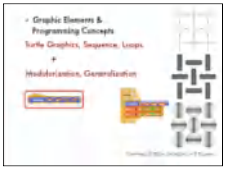
- Graphic Elements & Programming Concepts

Turtle Graphics, Sequence

+

Loops

The first recoding of a work of an early computer artist (in this case Rhythms by Erwin Steller), in addition to the turtle graphics commands needs an iteration (repeating a series of instructions). And that is actually all you need for it.



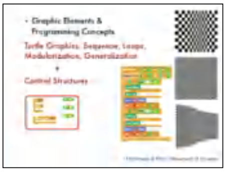
- Graphic Elements & Programming Concepts

Turtle Graphics, Sequence, Loops

+

Modularization, Generalization

Our second example - by Roger Vilder - is made of a lot of rectangles. This calls for simplification. Therefore, variables and procedures are introduced at this point. With this generalization it is easy to generate Vilders variations on 9 squares.



- Graphic Elements & Programming Concepts

Turtle Graphics, Sequence, Loops, Modularization, Generalization

+

Control Structures

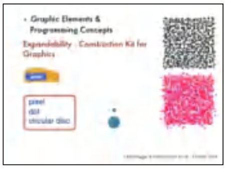
The third example is Bridget Rileys Movement in Squares, showing an optical illusion. To achieve this, in addition to the previous commands now control structures and conditional statements are required. Thus It is possible to define the boundaries at which the rectangles change the increase and decrease of their width.

And that is really everything you need up to this point.

In a similar way further concepts are introduced, like randomness, recursion, lists, event handling, messaging, cloning etc.

And when something is missing, we are able to add it by writing a new procedure.

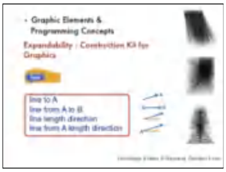
In our case a special graphic library is added. This library includes procedures for



- Graphic Elements & Programming Concepts

Expandability : Construction Kit for Graphics

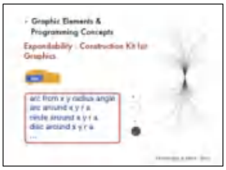
pixel
dot
circular disc



- Graphic Elements & Programming Concepts

Expandability : Construction Kit for Graphics

line to A
line from A to B
line length direction
line from A length direction



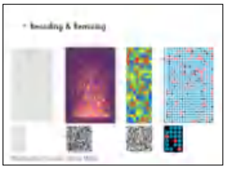
- Graphic Elements & Programming Concepts

Expandability : Construction Kit for Graphics

arc from x y radius angle
arc around x y r a
circle around x y r a
disc around x y r a
...

- different point shapes
- procedures for different lines
- and procedures for different arcs and circles.

Recoding & Remixing



- Recoding & Remixing

Equipped with this toolbox, a wide range of examples of early ComputerArt now can be recoded & remixed. The remixing can affect structures (for example in the case of Nees), colors and random distributions (at Struycken), shapes and colors (at Sýkora) or number of elements and colors (as in the case of Molnar). The range of possibilities is nearly limitless.

Multimedia for All!

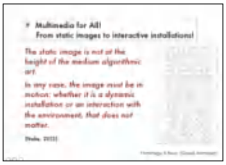
- Multimedia for All!

- Color
- Large Output Devices
- Interactivity
- Sensors & Feedback
- Drawing Robots
- Live Coding

We have seen earlier the characteristics and associated with them the limited possibilities of early ComputerArt, mostly due to technical restrictions of the available hard- and software. The rapid disappearance of ComputerArt certainly can be ascribed also to this.

But these restrictions no longer apply. Today there is color with the highest resolution, interactivity through a variety of input options, advanced controls and much more.

This is why static images for one of the pioneers are no longer at the height of time.


Multimedia for All!
From static images to interactive installations!

The static image is not at the height of the medium algorithmic art.

In any case, the image must be in motion: whether it is a dynamic installation or an interaction with the environment, that does not matter.

(Nake, 2022)

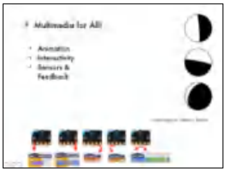
Through interactive elements, through sensors and feedback, the viewers are included in the work of art: they become contributors. And such objects and installations can be realized with the presented means.

Multimedia for All
- Animation

There have been rare animations already in early ComputerArt (as in this example of van Weeghel), but nowadays they are almost indispensable in multimedia installations.

Multimedia for All
- Animation
- Interactivity

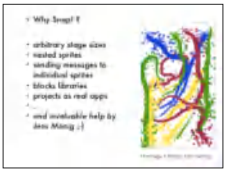
The control by the viewer can be carried out in the simplest case by inputting values within the program. In such a case the viewer must have access to parts of the code in order to change the inputs.


Multimedia for All
- Animation
- Interactivity
- Sensors & Feedback

Far more flexible is the use of microcontrollers, like the BBC MicroBit, Here I have made with it kind of a tiny game console. Using the buttons, motion sensors and accelerometer of the controller board, the various attributes of the animation can be controlled.

My hope is. that I have shown that without greater difficulties interactive installations are possible now for really everybody. In my experience, not only playful applications can be created this way but also serious art objects of high quality, which can even be shown in exhibitions and multimedia festivals.

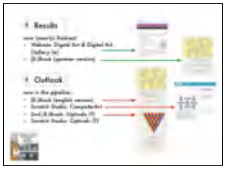
Why Snap! ?


Why Snap! ?
- arbitrary stage sizes
- nested sprites
- sending messages to individual sprites
- blocks libraries
- projects as real apps
- ...
- and invaluable help by Jens Mönig ;-)

As I said, most of the projects can also be realized with Scratch. But some features and attributes of Snap! can make life easier and improve the results. That was important for me because I needed really big pictures for exhibitions and interactive exhibits at festivals. At such events for example standalone executables (built with Snapp!) are extremely helpful.

Last but not least, without the help and the enthusiasm of Jens I probably would not have started my project on recoding & remixing ComputerArt. Special thanks to you, Jens!

Outlook


Results

now (nearly) finished:
- Website: Digital Art & Digital Art Gallery (e)
- (E-)Book (german version)

Outlook

now in the pipeline:
- (E-)Book (english version)
- Scratch Studio: ComputerArt
- 2nd (E-)Book: Optivals (?)
- Scratch Studio: Optivals (?)

I come to my last slide. Referring to my Scratch 2015 contribution, I can show two results: There is now my website Digital Art, including the Digital Art Gallery, which is even in English. The book on Coding Art is nearly finished.

I hope there will be an english version at some time. But I already have a second book (with a similar approach) as a work in progress.

Thank you for your attention!

http://digitalart.joachim-wedekind.de
Twitter: @jowede
E-Mail: j.wedekind@iwm-tuebingen.de

So I'm curious myself what I can report at the next conference ;-)