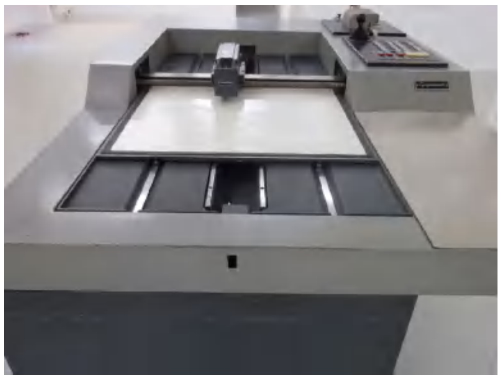
Snap!shot 2020 12th December 2020

How much CS does an artist need?

Programming Art with Snap!

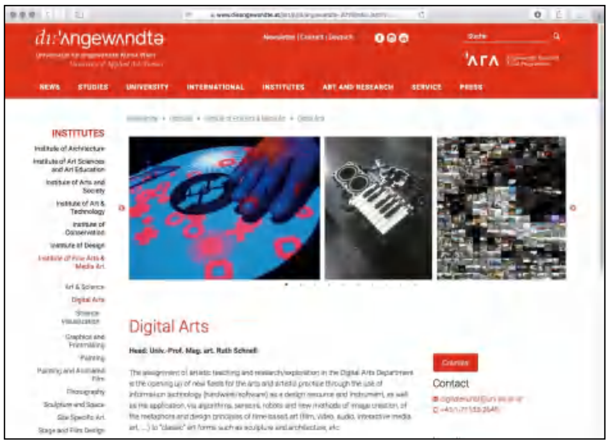
presented by Joachim Wedekind

Almost exactly fifty-five years ago, there was the first exhibition of Computer Art. The artists (who, by the way, did not understand or describe themselves as artists!) were mathematicians (like Frieder Nake), engineers (like Georg Nees), physicists (like Michael Noll) or computer scientists (like Kenneth Knowlton). This is not surprising, because the computers of the time had to be programmed with FORTRAN, with ALGOL or even in machine language in order to output the images on the first plotters. The range of skills required was therefore very technical in the early days.



What about today, when algorithmic art or generative art are the successors of computer art and can manifest themselves in multimedia, in animated and interactive installations (like in the example shown here: Flow, interactive exhibit by Karl Sims, 2018)?

There are now separate degree programmes for media art / digital media / generative art. These training courses for media artists include a high proportion of basic IT knowledge and technical introductions to the corresponding tools.



di:'Angewandte

NEWS STUDIES UNIVERSITY INTERNATIONAL INSTITUTES ART AND RESEARCH SERVICE PRESS

INSTITUTES

Institute of Architecture

Institute of Art Sciences and Art Education

Institute of Arts and Society

Institute of Art & Technology

Institute of Conservation

Institute of Design

Institute of Fine Arts & Media Art

Art & Science

Digital Arts

Science Visualization

Graphics and Printmaking

Painting

Painting and Animated Film

Photography

Sculpture and Space

Site Specific Art

Stage and Film Design

Digital Arts

Head: Univ.-Prof. Mag. art. Ruth Schnell

The assignment of artistic teaching and research/exploration in the Digital Arts Department is the opening up of new fields for the arts and artistic practice through the use of information technology (hardware/software) as a design resource and instrument, as well as the application, via algorithms, sensors, robots and new methods of image creation, of the metaphors and design principles of time-based art (film, video, audio, interactive media art, ...) to "classic" art forms such as sculpture and architecture, etc.

Contact

As an example, a quote from the University of Applied Arts in Vienna, the Digital Art programme: *It is about opening up new fields for art and artistic practice through the use of information technology (hardware / software) ... as well as the application, via algorithms, sensors, robots and new methods of image creation*.

Today there are even special programming environments specifically aimed at artists and designers: Processing (which is the absolute market leader), Touchdesigner, NodeBox, Pure Data, to name just the best known.

Processing

TOUCHDESIGNER BY DERIVATIVE

pd~

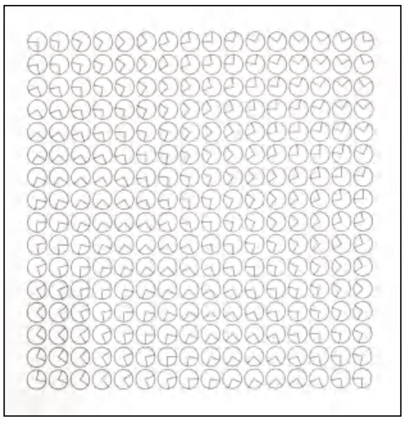
So if you come from the artistic side (like me), you will face considerable barriers to entry. The tools mentioned are quite complex (which is of course also due to their great power) and have a steep learning curve.

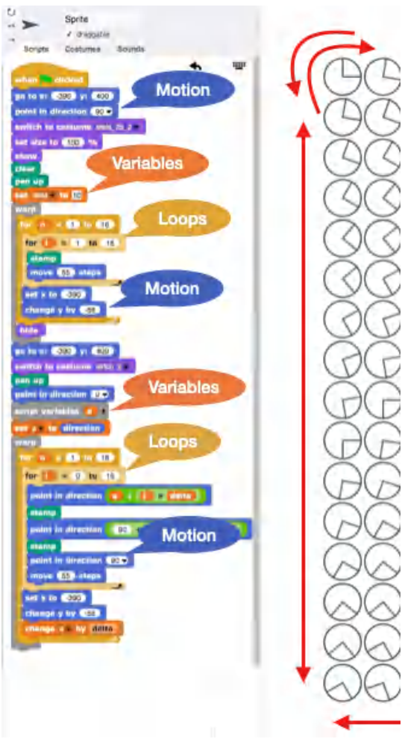
An interim conclusion could therefore be: It takes a good knowledge of computer science and technology to create digital, generative art.

Fortunately, I didn't let myself be deterred from getting into computer art. And luckily with Snap! I have found a great tool that is not only suitable for beginners but also for ambitious projects.

For anyone who, like me, started with the recoding & remixing of early computer art, very basic concepts are enough at first - like repetition, variables or dealing with controlled chance.

Here is such an example. Horst Bartnig (o.T., 1985) is a German representative of Concrete Art. He worked intensively on relationships between simple geometric shapes. His work with the computer allowed him the experimental implementation of the rules he defined. This example (from 1985) combines circles with two lines each. The result almost resembles a wall with world clocks.





The recoding of this example is actually very simple: Beside commands to move the turtle (MOTION), the concepts of LOOPS and VARIABLES are sufficient here.

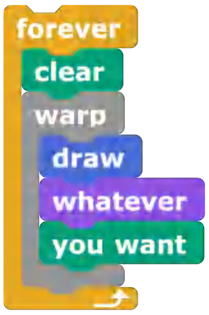
It made sense for me to turn this into an interactive installation. The viewers should be able to influence the behaviour of the system themselves.

Let's take first the possibility of controlling the speed of the rotation of the pointers, preferably separately for the two pointers:

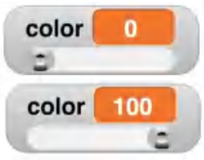
The controlling should be done via the horizontal and vertical movements of the mouse.

The result should then look like an endless rotation. Down left the speed is zero. Upwards it increases for the first pointer, to the right it increases for the second pointer.

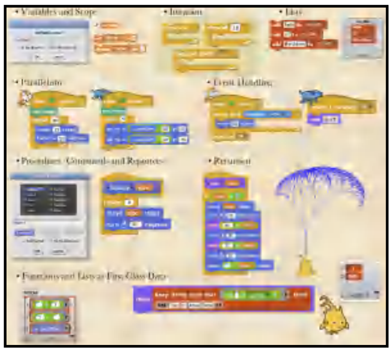
The traditional animation method, that is the stop motion technique, is unfortunately not suitable for this. With the necessary sensor queries, the program would be bloated and the process would slow down considerably.



animation



interaction

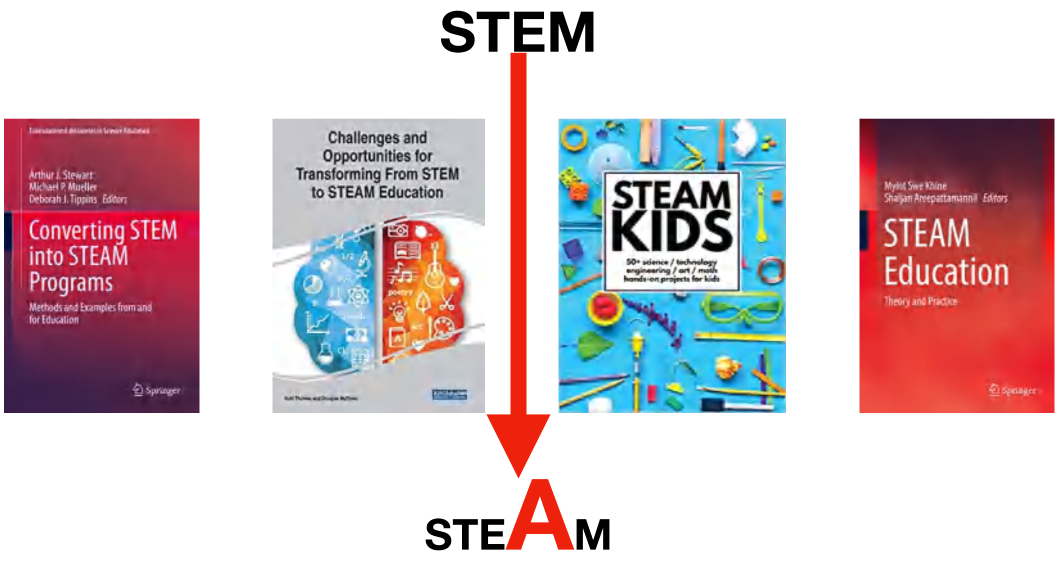


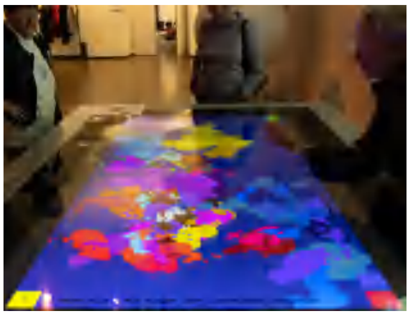
Luckily, I had learned something about the *Big Ideas in CS and Programming* in the *Beauty and Joy of Computing* course. From this I learned that it would be better to work with objects in this case. In Snap! I have sprites and clones available for this.

Without going into detail, it should only be mentioned here, that the pointers are cloned and then given specific properties. But they can also react to global changes.

So I have to admit: As an artist, I need to know quite a lot about programming concepts. If only because if I want to know whether and how I can achieve something. Then I have to know the possibilities and limits of my tools! Otherwise I easily overestimate or underestimate what is actually feasible.

So I should of course be happy that there are initiatives to add an A to the STEM approach in school.


STEM
Converting STEM into STEAM Programs
Challenges and Opportunities for Transforming From STEM to STEAM Education
STEAM KIDS
STEAM Education
STEAM


Painting with the Fingers - Exhibit with an interactive table, 28th November 2019, IWM, Tübingen

Behind this is the demand to integrate art into the teaching of science, technology, engineering and math. In principle, this is a good approach. However, I get sometimes annoyed when I all too often find simple random graphics with lines, dots, squares or other geometric objects as concrete examples and then people already start talking about computer art.

For me, this requires an intensive examination of models and examples from art history. So the A in STEAM needs careful design. Not an alternative, but an important addition, in my opinion, is therefor the integration of the “big ideas“ of Computer Science and Programming into the subject of art.

This can lead to exciting projects with surprising results that demonstrate the power of these Big Ideas in meaningful applications. At the same time, thus can be shown that the acquisition of solid computer science skills is indeed a worthwhile undertaking for artists.