Accessible art creation tools, a generative arts application.

R. Clair, MEng, MSc, PhD Student.
Université François Rabelais Tours, Laboratoire d'informatique, Tours, France.
www.li.univ-tours.com
e-mail: romain.clair@univ-tours.fr

Dr N. Monmarché, PhD.
Université François Rabelais Tours, Laboratoire d'informatique, Tours, France.

Pr M. Slimane, PhD.
Université François Rabelais Tours, Laboratoire d'informatique, Tours, France.

Premise

Art is an important activity for human beings and the apparition of computers revolutionized our practice. Some tried to use them to generate art, providing a lot of methods to produce graphic art or music automatically. We are working on accessible art creation programs and we propose to use these methods as interactive powerful tools to assist people who cannot practice regular art creation in a new form of art creation experience.

Designing accessible software means to follow conception guidelines in three main areas to ensure that most people will be able to :

- Get information from the program, by designing accessible interfaces, usually adopting multi-modality.
- Control the program. We provide some different way of commanding the application and make sure that they can be substituted by adapted devices.
- Understand the program. The program concepts should adapt to the user, or at least it will be simple and limit the required specific knowledge.

We give detail about two of our projects setting up these ideas. The first one is a virtual music instrument. It offers the opportunity to play music in real time using only

a keyboard or a mouse. It integrate an automatic accompaniment engine playing variations on the user's themes.

The second one is an accessible drawing workshop. It constituted a simple tool to create pictures. It provide the regular drawing tools like pencils and brushes but also more advanced ones based on generative arts.

1 The project context

When taking an interest in art, we are forced to admit that it is a very large, and not so well defined domain. In our research, and in this paper as well, we are interested only in graphic art and music. This two particular fields of art creation have bounded close links with computers as soon as they appeared.

Automatic or semi-automatic art creation, generative arts in a way can be a main interest for computer researchers because it involves most of the existing metaheuristics. Studying the methods used to generate arts, we have encounter Markov chains [1], generative grammars [2], artificial neural networks [3], constraint programming [4], genetic algorithms [5], artificial ant colonies algorithms [6] *etc.* It is a very special kind of problem on which this generic methods can be applied. Thus it constitute an amazing and excellent way of testing and sometime improving this techinques.

Our research team works on using computers to help disabled people. Obviously we are working on improving the basics of their everyday life: for using the web [7], helping therapists [8], or for video games [9]. But the idea of accessible arts seems also important for us. Thus we worked on the idea of accessible art creation tools, using generative art as an art creation assistance for disabled people.

After the definition of the notion of accessibility and its impact on the design of programs, we'll describe two of our projects associating generative arts and accessibility: a virtual music instrument and a drawing workshop.

2 General consideration on accessibility

2.1 Definition

Accessibility covers a lot of different notions. To define it, we can refer to its application on the world wide web. The World Wide Web Consortium (W3C) with the Web Accessibility Initiative (WAI) provides advices and guidelines to make the web more accessible. They define accessibility like this:

“Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging.”[10]

This definition can easily be extended to every computer software. Thus we have to deal, since the beginning of application design, with three main problems:

- Perception. We should by-pass sensory disabilities and make sure our interfaces provides information for each and every user.

- Comprehension. Cognitive deficiencies as well as cultural or knowledge differences must not be an obstacle to understand our softwares.
- Control. Mainly linked with motor disabilities, we should ensure that everyone can control our applications.

Obviously, dealing with handicap is one of the main issue designing accessible programs, but it can also involve cultural habits and economical problems. It follow the ideal and may be utopian goal of universal access.

2.2 Interfaces

The application interface, the more often a Graphical User Interface (GUI) is the medium used to communicate information to the user. We have all get used to the standard application window, with its “file – edition ...” scrolling menu. This kind of interface is not accessible to everyone and specially not for blind or visually impaired.

Multi-modality is way to avoid this problem. The idea is simple: if everyone cannot access to the information on the firstly planned way, alternative ways should be proposed in addition. For example if using a classical GUI, a vocal or a Braille display should be also provided, or a text-only, that can more easily be magnified. Indeed, accessibility can rely on external tools, provided by the operating system for example. Most of the present operating systems offers a zoom to enlarge a part of the screen, or a speech synthesis to pronounce the textual elements. But the program conception must assure that this external tool can be efficient. For example, the graphical structure of information, like columns, giving a meaning to the position of elements can disable a speech synthesizer which reads the screen line to line.

2.3 Controllers

In the other direction, controllers are the way an application can get information from the user. There is a wide range of devices for this purpose, from the most standard ones, like keyboards and mice, to the most specific, like motion detectors, breath captors, joysticks *etc.*

In this domain, the mouse – keyboard couple is the most commonly used. Sometimes, they set up two different ways of controlling the application, but they are also often combined to form a two handed complex controller. Of course it offers powerful command abilities, but it is reserved for able people. In the area of accessibility, it cannot be a satisfying solution.

As for interfaces, controllers should also be multiple and distinct. Multi-modality applies for input devices too. There should always be two or more different way of making each of a program expected action.

Assistive technologies are various here, adapted devices exists for a lot of disabilities. Most of them are substitute for keyboards and mice, but they do not produce exactly the same behaviour and offer the same opportunities. This differences have to be taken into account when designing accessible controllers. For example, using a virtual keyboard cannot be as fast as a trained user on a regular one.

2.4 Concepts

The third and may be the most difficult goal is to ensure that what we propose can be understood by everyone. It covers the technical aspects, the way the programmer have encoded the software. As it seems obvious for anyone that this technical

considerations should not appear to the user, we just are forced to admit that it is usually not the case, and that most of the programs required few knowledge in using computers. It is particularly the case with computer graphics or music. Many specific notions associated with the way the computer deals with images or music have to be understood by a user willing to use a software. With computer graphics, the comprehension of image resolutions, compression algorithms, *etc.* are required, and for examples in music: sample rates for digitalized music or the midi standard.

The specificities associated with the activity proposed by the program should also be taken into account. For example, do I have to know the music theory to play music with a computer program ? We suggest that being accessible also means that the required specific knowledge should be at least limited.

All this constraints can be beneficial during the conception of the program, but it should constitute a limitation to the capacities offered. The more a software can do, the more complex it is. But this consideration is not a fate. An ideal accessible software is capable to adapt to the user needs, which means not only that it will not overwhelmed a user facing difficulties. It also means that it should get richer when its user learn to use it. The assistance provided is important, but may be not in every case.

3 The virtual music instrument

3.1 Goals

This first project focus on music. The founding idea is that computers can be a way to access a form of musical expression for those who cannot use traditional instrument, for example. It leads us to the concept of a virtual music instrument.

The targeted audience tends of course to be as vast as possible. But to be more specific, in the field of handicap, we aimed more at motor and cognitive disabilities than at sensory ones. The reason is simple, blind and visually impaired are able to practice music, and there are some famous examples. On the contrary, music for deaf people needs to be very specific, based on the few frequencies they are able to perceive. We also want to provide a creation tool for the people who do not play music because they do not know music.

Therefore, we developed a computer program for playing music in real-time and of course, it should be as accessible as possible. It obeys the following rules:

- Integrating several controllers
- A maximum of build in music knowledge
- No visible technical information
- As easy to use as possible
- Assisting the use with an automatic accompaniment

3.2 Accessibility considerations

As it is a musical tool, we limit the visual information to the minimum. In facts, there was no need for displaying anything, so the software interface is only a blank window

where the mouse can move. This limitation to produce only sound and no graphics is also for testing purpose, because we do not want to have visual “interferences“ to the feeling users can have about the produced music.

There are, for now, two possible controller for this instrument. The keyboard is the first one. It imitate the behaviour of a classic piano keyboard. Pressing a key triggers a sound which stops when the key is released. Each note appears more than once on the keyboard, the idea is that we do not offers to many possible notes, but each note can be played using different keys. This way, we offer the opportunity to play major triads using adjacent keys. From left to right we mount over the scale and on top of each of the bottom note, there is its third, followed by its fifth. Of course, the user is not supposed to know these harmonic notions. He or she will just feel that going to a key to the upper one will produce a pleasant result, actually a rather consonant one.

The second is based on the mouse. Of course, it can be substituted by an adapted device, so the mouse controller is only base on the mouse movement. The buttons are only used to toggle on or off the mouse controller. The functioning is based on the distance crossed by the pointer since the last note played. When the pointer moves away from its last origin, nothing is done, but when this distance reduces, a note is played and the origin is reset to the current position. The pitch of the note played is proportional to the percentage of the global window crossed; the higher the percentage, the deepest the note. It must be a new and original approach of producing music, at least we hope so.

To limit the music knowledge required for using the software, we have adopted two measures. Firstly, we have limited the available notes to a two octave major scale. The underlying idea is to ensure a more pleasant results by limiting the dissonances. Thus, the instrument has a 15 note range. Secondly, we chose to mask the notions of tempo. The ideal case is to detect automatically the tempo in what the user plays. The detection is not yet implemented, the program works at a fixed tempo.

3.3 Automatic accompaniment

To assist the user in his or her musical experience, the virtual music instrument has a built in automatic accompaniment. It starts when the user plays, and the result depends on what he or she plays.

It is based on an interactive artificial ant colony algorithm. Additional bass notes will be played according to what the user plays, as soon as he or she begins. We ensure that this accompaniment is consistent with the user's playing by using his or her played notes pitches and rhythms as the basis for our generator. The engine is only able to produced variations of the user's playing by combining the already played notes with the already played rhythmic patterns. The behaviour of the engine when let on his own his to create and repeat a cycle of notes. But the user influence it when he or she plays and the automatic accompaniment tends to follow the user.

For more detail about the functioning of the algorithm, see [11]

3.4 First results

We have started a test phase where we have presented the program to its presumed audience. We have met therapists, musician, children, able and disabled people and we have collected their returns.

Some found an interest in using this software. The obvious first reason for it is fun. As it is an uncommon and surprising tool, people and specially children find it entertaining. From the therapist's point of view, it seems that the program can be used for relaxation. All the sighted people feels like the program should also provide a visual feedback, specially when using the mouse.

Of course, the program do not pleased everybody and the main reason seems to be that when explaining what the software is supposed to do, people preconceived an idea of what they will get and it can be quite far from what the program really propose.

4 The accessible drawing workshop

4.1 Goals

The second project carrying out these ideas of accessible art creation tools is graphic oriented. It consists in an accessible drawing workshop, with the usual tools like pencils and brushes but also integrating most advanced tools based on generative methods.

Once again, we wanted this program to be accessible to the largest possible audience. Of course, it is not planed to compete with existing art creation tools. This tools are much more mature but above all, they are not designed in the same purpose. These are tools for artists and in fact for artists able to learn the use of complex computer programs. See [12] for more detail about existing art creation tools.

We aimed more at motor and cognitive handicaps here, for similar reasons. Our workshop should also be easily used even by young children. Thus it was designed following this guidelines:

- Integrating several controllers
- No complex settings
- No visible technical information
- As easy and pleasant to use as possible
- Assisting the user with semi-autonomous generative tools

4.2 Interface

Obviously, a drawing workshop offers a Graphical User Interface (GUI). According to the previous rules, we have chosen not to match with the standard of GUI design. The program do not integrate the usual scrolling menu bar. It is commanded by graphic bubble shaped buttons. They are grouped in a tool bar bringing together the drawing tools, a circular flower shaped group with the general purpose command buttons and a rosette for colour choices. We wanted the GUI to be adapted for each user so each user has his own profile where he can choose the position of each buttons group. Figure 1 shows an example of the position of the GUI elements.

Figure 39: Drawing workshop GUI

There is another pop-up menu which appears near the actual position of the mouse pointer when it is not moving. Indeed, some people can get easily tired of crossing long distance with a mouse pointer so we wanted to always offer a close access to their usual tools.

As everyone have already guessed, the program can be controller with a mouse. We have taken a particular attention to this controller to limit tiredness. With the pop-up menu of course, but the maintaining of a pressed mouse button has been reduced to the minimum and every tool works with a toggle. The mouse button activate the pencil which draws until the button is pressed again.

Keyboard can also be used with a single key short-cut to select each tool and the opportunity to draw using the keyboard arrows.

4.3 Advanced tools

In addition to the standard tools which for now are a pen, a line, we added more advanced ones. The first included tool is also based on artificial ant colonies.

Pressing the button when one of the ant tool is selected will drop ants on the picture. They will move on it laying down a colour behind them. Their movement imitate the collective behaviour of real ants. In fact, each ant is looking for a specific colour. It will explore the picture randomly but tends to be attracted to the area containing its seeked colour. The Figure 2 is a picture produced using this tool with one single click on the centre of the image.

Figure 40: Example of ants drawings

We planed to have more advanced tools, based on fractals, L-systems or genetic algorithms for example, but as the project is still under development, they are not yet implemented. For the same simple reason, we do not have now enough evaluation to produce real results, but the very first test are encouraging.

5 Conclusion

Generative arts can also be used to help disabled people access a form of art creation. We have designed accessible art creation tools, adapting automatic art generation tools as an assistance for people which encounter difficulties in accessing regular art creation. Following accessibility guidelines and making this automatic art generators interact with the user, we have conceived two programs: A virtual music instrument to play real-time music accompanied automatically by the software and a accessible drawing workshop with which one can create digital pictures using art generating tools.

As an objective evaluation of an art production is inappropriate, we have proceeded to a test phase where we have gathered the comments of real users while they experience the software. The first results of the subjective evaluation of one of this tools seems to be promising.

Bibliography

1: Iannis Xenakis, Formalized Music, University Press, New York, USA, 1971.
2: Jon McCormack, Grammar Based Music Composition. In Complex Systems 96, ISO Press, 1996.
3: Michael C. Mozer, Neural network music composition by prediction, Connection Science, 1994.

4: Martin Henz, Stefan Lauer and Detlev Zimmerman, COMPOzE - Intention-based Music Composition through Constraint Programming, 1996
5: John Al Biles, GenJam: A genetic algorithm for generating jazz solos, 1994
6: Nicolas Monmarché, Isabelle Mahnich and Mohamed Slimane, Artificial Art made by Artificial Ants. In The Art of Artificial Evolution, Springer, 2007.
7: Sonia Colas, Nicolas Monmarché and Mohamed Slimane, Reaching Web accessibility compliance with to an automatic correction tool, Special issue of the AMSE journal, 2008.
8: A. Puret et al., A robot-toy to incite the communication from a child with autism, Special issue of the AMSE journal, 2008.
9: Alexis Sepchat, Romain Clair, Nicolas Monmarché & Mohamed Slimane, Using ants' task division for better game engines, 2008
10: Education and Outreach Working Group (EOWG), Web Accessibility Initiative (WAI), 2005, http://www.w3.org/WAI/
11: Romain Clair, Nicolas Monmarché and Mohamed Slimane, An interactive ant algorithm for real-time music accompaniment. In Evolution artificielle, 9th international conference, Strasbourg, october 2009.
12: Romain Clair, Nicolas Monmarché and Mohamed Slimane, Art, an accessibility issue. In Accessible Design in a Digital World (ADDW), York, 2008.