Surfing with Narcissus: Updating the Technology used in Thea Musgrave's work for Solo Clarinet

Luan Luiz Gonçalves
ALICE - Arts Lab in Interfaces, Computers, and Else
UFSJ - Federal University of São João del-Rei - Brazil
human@flossuniverse.org

Rafael Alves Soares de Andrade
ALICE - Arts Lab in Interfaces, Computers, and Else
UFSJ - Federal University of São João del-Rei - Brazil
rafael.asa@outlook.com

Iura de Rezende Ferreira Sobrinho
Music Department
UFSJ - Federal University of São João del-Rei - Brazil
iuraderezende@gmail.com

Flávio Luiz Schiavoni
ALICE - Arts Lab in Interfaces, Computers, and Else
UFSJ - Federal University of São João del-Rei - Brazil
fls@ufsj.edu.br

ABSTRACT

Narcissus is a work created in 1987 by Thea Musgrave for clarinet (or flute) and delayed audio that uses a delay box. In order to recreate the piece using modern technology, our research team developed a web application to emulate the functions of the original equipment in Narcissus. The present paper is our report about the experience of recreating Musgrave's work using web audio and web music APIs.

1. INTRODUCTION

The piece Narcissus, created by the Scottish composer Thea Musgrave, is an important work in the repertoire for clarinet with live electronics. This piece, presented in Section 2, was composed for clarinet in B-flat (or flute) and a delay box. Several technical details alongside the musical notation determine the parameters of the electronic effects and the interaction between the musician and his/her delayed sound, as shown in Figure 1.

The delay box originally chosen by Musgrave in her composition was the system VESTA KOZO DIG-411, which has been discontinued from production. Therefore, in order to perform this music today, one needs to emulate the original gadget using updated technology. The attempt to emulate the original electronic hardware of this work of art using updated resources led our research team to develop the project presented in this paper. Since the project itself deals with art music supported by technology, a team of researchers from the fields of Music and Technology – formed by members of the Music Department and the Computer Science Department at the University – was put together in a multidisciplinary effort to guarantee the integrity of the composer's vision and facilitate its performance in the 21^{st} century [5].

The first step taken to recreate this piece of music was the emulation of the original hardware functions, in which a DAW was applied, as presented in Section 3.

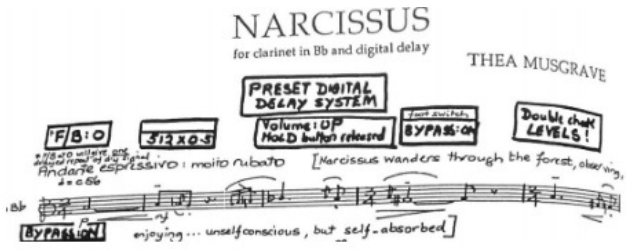

Figure 1: The original score mixing musical information and technical details.

Once the full performance was heard and understood, the research team developed a web-based system, described in Section 4. Finally, Section 5 brings some discussion about this work, raises some issues and presents lessons learned in the project implementation and Section 6 brings some final considerations.

2. NARCISSUS

Scottish-American composer Thea Musgrave, b.1928, wrote the work Narcissus (1987) for flute or clarinet in Bb and Digital Delay System as a commission for the National Endowment of Arts and for individuals Wendy Rolfe, Harvey Sollberger, Patricia Spencer, and Robert Willoughby [8]. Flutist Wendy Rolfe premiered the piece using the Digital Delay System called Vesta Kozo Dig 411, which has since been discontinued from production.

The artistic representation of the Greek mythical character Narcissus is recurrent throughout history. Musgrave's

Web Audio Conference WAC-2021, July 5–7, 2021, Barcelona, Spain.

Narcissus is a piece of music seemingly inspired by Romantic aesthetics, as it uses compositional devices with strong allusions to the Romantic musical period. The music is shaped around the story of Narcissus, the synopsis of which is presented fully in the score, as done by many composers of program music[1] in the nineteenth century. Musgrave writes:

> Narcissus wanders through the forest, observing, enjoying... unselfconscious but self-absorbed. He sees a pool of water and then as he approaches notices his reflection in the water. He is intrigued and then jumps back in fright. Once more he approaches... It is still there. Narcissus steps away from the pool to consider this phenomenon. Several times he approaches, the figure is always there watching him. In the shimmering sunlight Narcissus seems to see this glorious and attractive being moving in the rippling water. He is dazzled and slowly holds out his arms. To his amazement the figure responds. In awe and wonder Narcissus approaches closer and closer. With a sudden change of mood Narcissus dances happily and playfully... the figure echoing him. But then Narcissus begins to question anxiously the lack of any independent response... is he being mocked? He gets more and more agitated and finally in a fury he rushes headlong into the water to grapple with the figure. The waves surge up and Narcissus is drowned. There is a distant shimmering vision of Narcissus and his reflection. Then in the setting sun the vision disappears, the forest is empty and the pool lies undisturbed. [8, n.p.]

Musgrave chose to represent the character Narcissus through the use of a recurrent musical theme, reminiscent of the concept of the Wagnerian Leitmotiv[2], which represents or is associated with the main character in the story. The character Narcissus experiences a variety of emotional states in the piece, which are reflected in the transformation of the Narcissus Leitmotiv, as well as of the electronic effects. These particular treatments of musical material and the use of a program as structural element for musical composition are long understood as hallmarks of Romantic Music and exist in different musical genres of the period as well.

Musgrave's work reaches its dramatic peak when Narcissus "in a fury (...) rushes headlong into the water to grapple with the figure" and drowns. This moment of fury is represented musically through a rupture with tonality. The work comes to an end in the final coda, in which the Narcissus theme reappears along with other motifs of the piece, transformed by pitch variations in the electronic effects and by a change into a much slower tempo.

3. DIVING INTO NARCISSUS' POOL OF WATER

[1]Program music is defined as an instrumental music that tells a story or follows a narrative or other sequence of events, often spelled out in an accompanying text called a program. [4, A15].

[2]Music analysts since Richard Wagner's time called certain of his compositional techniques Leitmotiv (German, leading motive), which are musical themes each associated with a particular character, thing, event or emotion. [4, p.689].

To recreate Narcissus, we started by reconstructing the necessary electronic effects to play it. The original score reveals several details to set up the Kozo Delay box but, since the original equipment is not available to our team, it was necessary a closer look at the piece to understand how to set the piece up. Our work started with the reading of the paper "Analysis and Reconstruction of Interactive Electroacoustic Works for Obsolete Technology: Thea Musgrave's Narcissus" [12].

Since our development team was spread in several cities and only one clarinetist was part of the team, we decided that a recorded version of the piece could be very useful to play with and test all the FX parameters.

In order to test the information and presets, the piece was recreated inside a DAW (Digital Audio Workstation) utilizing the dry recorded clarinet audio to map, experiment and automate the presets parameters. Narcissus was reconstructed inside Reaper 6 with a simple delay plugin containing mainly mix, delay time and feedback parameters (cf. Fig. 2). Those parameters were automated to match the sequence of presets of the original hardware and from there, to experiment with the settings in order to emulate possible undesired artifacts caused by preset automation and live time delay tempo change, being that delay time altered according to player's tempo with the use of a bpm detector.

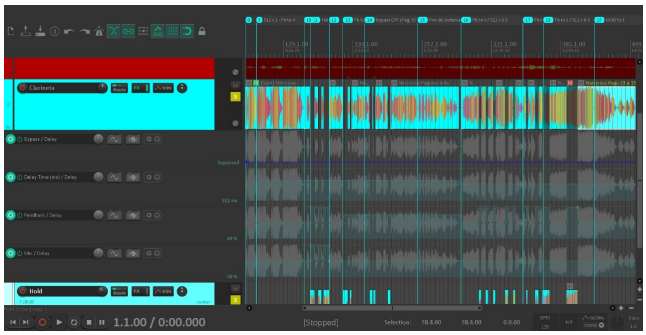

Figure 2: Reaper screenshot: map, experiment and automate the presets parameters.

The first issue encountered in the DAW's recreated version of Narcissus was that the delay signal would not match the clarinet's dry signal. That occurred due to the fact that the performer had recorded the clarinet audio without the perception of the delay tempo, so when the delay was later added it mismatched the dry clarinet's timing in some very rhythmically rich sections, decharacterizing the piece. Also, in order to emulate the behaviour of a software that could adjust the delay time according to the performer's dynamics as well as it's bpm, automation tracks were created to control the delay tempo variation. But those variations resulted in a undesired pitch shifting effect.

All in all, our first experimentation with Musgrave's "Narcissus" helped us understand that the reconstruction of the piece could need some effort to develop a software to recreate the original piece.

4. SURFING IN A POOL OF WATER, A WEB-BASED NARCISSUS

Table 1: Presets to play the piece

Preset	Delay Time	Feedback	Modulation
1	256ms	0%	0
2	256ms	25%	0
3	512ms	50%	0
4	512ms	75%	0
5	256ms	50%	0
6	256ms	75%	1
7	256ms	75%	2
8	1024ms	75%	2
9	1024ms	75%	1
10	1024ms	75%	0

Once we understood the piece setup, our next step was to develop a software to recreate the original piece. Several possibilities were considered and our intention was to develop an open source software to allow the access to other players interested in playing this piece. For this reason, we started thinking about a web based version of the piece.

Firstly, we developed an application based on Web Audio to play with. To help developers to understand the piece's parameters and to test it without playing an instrument, two input was defined: a microphone input and an audio file input. The output was also developed to grant communication to our team, with possibilities to listen the output or to record it on an audio file. Figure 3 presents the input, output and AudioNode – generic interface for representing an audio processing module – of our system, as well as the interactions between the end-user and the web-based Narcissus.

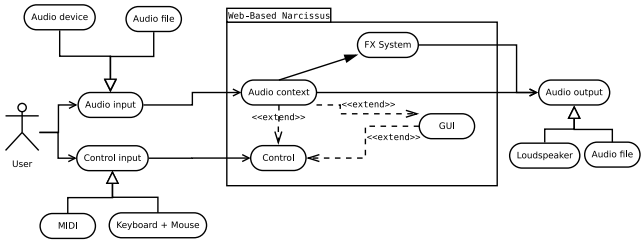

Figure 3: Use case diagram: the inputs and outputs of our system.

The end-user can choose the source for the audio input (Audio device / Microphone or Audio file). It is also possible to control the application with a MIDI controller (Control) or the input devices Mouse and Keyboard (GUI - " Graphical User Interface"). The audio output can be the loudspeaker or audio file.

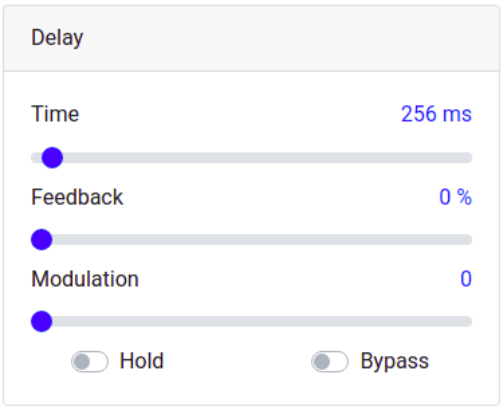

Figure 4: A GUI to define Delay parameters.

Our FX System was a delay system based on the Web Audio API. The schematic presented in Figure 5 shows the routing of all AudioNode objects inside the audio context.

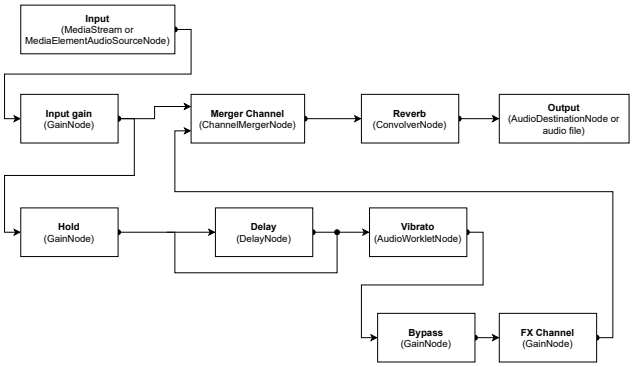

Figure 5: AudioContext: routing AudioNode objects.

Most part of the Audio Nodes were developed using native nodes from the Web Audio API. However, this API does not offer a Vibrato implementation, so we had to add our own solution. The implementation of vibrato consists of modulating the frequency of the original audio to a certain depth and time period. The implementation follows the idea of audio pitch shift, using a sine table to define when the audio will be played faster or slower, modulating the frequency. In the case of playing the audio faster, it was necessary to interpolate values between the original audio to obtain sufficient sampling.

Originally, the piece depends on some foot switch: a hold button, to keep the delay "infinite", a bypass button, to eliminate the FX in the system, and a volume to define the output gain of the FX system. In our implementation, the hold system would define the value 0 (zero) to *Hold* (GainNode) (Fig.5) and the bypass and FX volume would define the value of *Bypass* (GainNode), disconnect the Bypass from Merger Channel and connect the Input Gain on the same channel of the Merger. The delay feedback is a attribute of Delay class (GainNode) and would define the amount of repetitions.

A GUI based on Bootstrap was created to define the parameters of the delay, as presented in Fig.4. This framework was chosen to offer a simple and friendly GUI. To play the

Table 2: Presets to play the piece

Function	Shortcut	MIDI Control
bypass	b	6
hold	h	7
next preset	→	9
previous preset	←	10
volume	-	12

piece, a set of 10 presets was prepared with parameters predefined in Tab.1 as Narcissus default presets. The user can change the preset values and also remove and add new presets.

To help the musician to control the piece beyond the mouse, we defined a set of shortcut keys to select the presets and also control the record, bypass, hold and output volume. We also define a set of MIDI messages to allow the control using a MIDI controller, like a pedal board. The MIDI input was based on Web MIDI API. Tab.2 presents the controls defined to the application.

5. DISCUSSION AND LESSONS LEARNED

The API compatibility in different browsers and operating systems was one of the first issues to be tackled during the development of our work. The Web Midi API is still facing some difficulties to run on Chromium browser in Linux when installed using the snap packet manager. An application installed by snap runs in a "sandbox" and cannot access the local configurations necessary to run Web Midi API. We solved the issue running a version of the browser installed directly from the source code.

Another issue related to several differences between the same web API in different browsers was about the latency of the system. Certainly, our intention was to implement a real time system with short latency. In order to do so, a latency test was applied (as depicted in Fig. 6) but unfortunately it is not compatible with every Internet browser.

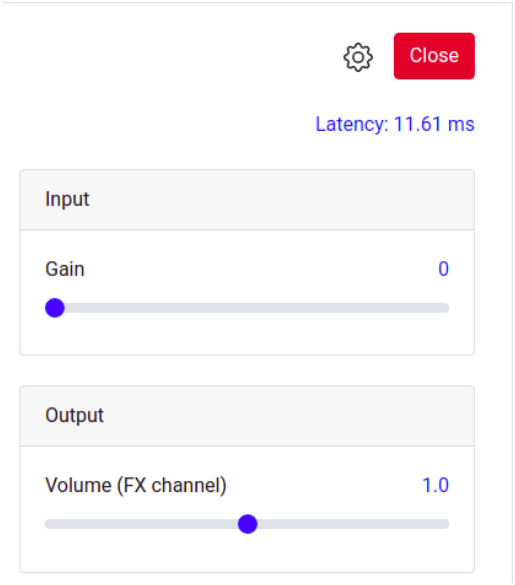

Figure 6: The latency measurement of the system.

The third problem related to the development was about the performance of the piece itself. Since our team had only one clarinetist, our developers were not able to test the software themselves. The possibility to play it by an audio file and to record the output was the only possible solution found to help.

In order to process audio using JavaScript, the team used the AudioWorkletNode interface to replace the ScriptProcessorNode (deprecated feature). These interfaces allow for custom audio processing, generation and analyzing. A new issue arose for implementing vibrato using the deprecated ScriptProcessorNode, which presented an unsatisfactory result due to the presence of glitches – probably because of a data race problem during the reading and writing of the buffer featured in this interface. To improve the audio latency we are replacing the ScriptProcessorNode interface by AudioworkletNode. AudioWorklet – interface of the Web Audio API thats allow the audio data requests to be handled in a background thread using the WebWorker technology to implement threading. The AudioWorkletNode is the main audio processing thread and can communicate with other threads. Custom DSP can be written in JavaScript or WebAssembly, and communicate with the main thread [9, 10].

An aesthetics issue emerged when we started trying our application. Since the musician was at home, using headphones and without real loudspeakers, the final sound was to dry and a little bit artificial. To improve this experience, a reverb fx was added on both audio channels to avoid the original dry sound. A convolution reverb was implemented using ConvolverNode and audio samples (Impulse Responses[3]). These samples describe how the room responds to an impulse (short clap) and the user can choose between 38 room options to play.

6. FINAL CONSIDERATIONS

This paper presented our web journey to recreate Narcissus, a piece from the 1980's created by Thea Musgrave, using contemporary technology. In this paper we presented part of our system, a prototype that can evolve and became a nice platform for musicians and composers. However, the developed system is not the main outcome of our project.

The possibility to recreate this piece is, surely, a chance to put other eyes in the conservation of mixed music repertoire. We are not sure if our system can long too many years but we believe that it is important to update the technology and that this recreation can help other musicians to access this repertoire, the source code of our application and all the context around this work.

We started to develop this piece in a multidisciplinary effort between researchers from the Music Department and the Computer Science Department in the Federal University of São João del-Rei, Minas Gerais, Brazil. Musicians that know how to read a score and play an instrument and computer scientist that knows how to program a computer started to create this piece. Probably, a perfect team to work together and no challenges to overcome.

What would be easy in any normal situation became a strange situation in the beginning of this project when the COVID-19 pandemic guided us to a social isolation in our homes. How does one exchange information and create a

[3]https://www.voxengo.com/impulses/

specific musical application when programmers that do not play instruments and musicians that do not know how to program are isolated in their homes without the chance to sit in chairs, face to face?

Since the computer science researchers had experience in web art and Web Audio development[4], probably a web application to recreate Narcissus would be our normal first option. However, in 2020, a web application was the best possibility, maybe the only one, to create the piece remotely and distributed.

We must continue to investigate technologies, frameworks and design standards to improve. The Web Audio Conference[5] is our main source for updating our knowledge about the state of the art for the development of web audio applications. There are many works that support us to implementing a better Web-based Narcissus [3, 6, 7, 2, 1].

In a near Future we intend to create other versions of the System, to make it more available to anyone that intend to play it using a computer instead of the original delay box. To recreate a piece like Narcissus is an amazing opportunity to create a partnership between programmers and classical musicians to work together.

This journey was not normal and will be remembered due to the difficulties faced in our project. If one intend to check our code, contribute with it or just take a look on it, check out our repository https://github.com/Alice-ArtsLab/narcissus.

7. ACKNOWLEDGMENTS

The authors would like to thank all the members of our Research Lab, ALICE, who made this research possible. We also would like to thanks the colleagues of CEGeME and professor Maurício Alves Loureiro, that first played this piece. We would also like to thank the funding agencies CNPq (151975/2019-1), UFSJ and FAPEMIG (APQ-02148-18).

8. REFERENCES

[1] L. Arandas, J. Gomes, and R. Penha. Towards large-scale artistic practice with web technologies. In A. Xambó, S. R. Martín, and G. Roma, editors, *Proceedings of the International Web Audio Conference*, WAC '19, pages 87–91, Trondheim, Norway, December 2019. NTNU.

[2] F. Bernardo, C. Kiefer, and T. Magnusson. An audioworklet-based signal engine for a live coding language ecosystem. In A. Xambó, S. R. Martín, and G. Roma, editors, *Proceedings of the International Web Audio Conference*, WAC '19, pages 77–82, Trondheim, Norway, December 2019. NTNU.

[3] M. Buffa, J. Lebrun, J. Kleimola, O. Larkin, G. Pellerin, and S. Letz. Wap: Ideas for a web audio plug-in standard. In J. Monschke, C. Guttandin, N. Schnell, T. Jenkinson, and J. Schaedler, editors, *Proceedings of the International Web Audio Conference*, WAC '18, Berlin, Germany, September 2018. TU Berlin.

[4] J. P. Burkholder, D. J. Grout, and C. V. Palisca. *A History of Western Music: Tenth International Student Edition*. WW Norton & Company, 2019.

[4]More about the Arts Lab ALICE can be found in [11].

[5]https://webaudioconf.com/

[5] I. de Rezende, R. A. S. de Andrade, and F. L. Schiavoni. Lições aprendidas com a peça narcissus de thea musgrave. In *Anais do XXX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1–11, Manaus, AM, Brasil, dec 2020. ANPPOM.

[6] J. Kleimola and O. Campbell. Native web audio api plugins. In J. Monschke, C. Guttandin, N. Schnell, T. Jenkinson, and J. Schaedler, editors, *Proceedings of the International Web Audio Conference*, WAC '18, Berlin, Germany, September 2018. TU Berlin.

[7] B. Matuszewski. Soundworks - a framework for networked music systems on the web - state of affairs and new developments. In A. Xambó, S. R. Martín, and G. Roma, editors, *Proceedings of the International Web Audio Conference*, WAC '19, pages 65–70, Trondheim, Norway, December 2019. NTNU.

[8] T. Musgrave. *Narcissus: for solo flute with digital delay*. Novello, 1988.

[9] C. Pendharkar, P. Bäck, and L. Wyse. Adventures in scheduling, buffers and parameters: Porting a dynamic audio engine to web audio. In S. Goldszmidt, N. Schnell, V. Saiz, and B. Matuszewski, editors, *Proceedings of the International Web Audio Conference*, WAC '15, Paris, France, January 2015. IRCAM.

[10] C. Roberts. Metaprogramming strategies for audioworklets. In J. Monschke, C. Guttandin, N. Schnell, T. Jenkinson, and J. Schaedler, editors, *Proceedings of the International Web Audio Conference*, WAC '18, Berlin, Germany, September 2018. TU Berlin.

[11] F. Schiavoni, A. Gomes, A. de Paulo, C. de Souza, F. Carvalho, F. Resende, G. Rocha, I. S. Junior, I. Andrade, J. Araújo, L. Santos, L. Gonçalves, M. C. Canito, O. da Silva, R. Costa, T. Morandi, and V. S. Junior. Alice: Arts lab in interfaces, computers, and everything else - research report (2019). In *Anais do XVII Simpósio Brasileiro de Computação Musical*, pages 157–164, Porto Alegre, RS, Brasil, 2019. SBC.

[12] D. Wetzel. Analysis and reconstruction of interactive electroacoustic works for obsolete technology: Thea musgrave's narcissus. In *ICMC*, 2004.