

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Adventures in Quantumland

Permalink

<https://escholarship.org/uc/item/93c1t8vx>

Author

Hamido, Omar Costa

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Adventures in Quantumland

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Integrated Composition, Improvisation, and Technology

by

Omar Costa Hamido

Dissertation Committee:
Professor Michael Dessen, Chair
Professor Christopher Dobrian
Professor Sandy Irani
Professor Kojiro Umezaki
Dr. Christine Corbett Moran

2021

DEDICATION

To my parents

Gracinda and Nijamodine

and my brothers

Amir and Abel

for staying entangled

even when I am so far away

TABLE OF CONTENTS

TABLE OF CONTENTS	III
LIST OF FIGURES	V
ACKNOWLEDGEMENTS	VII
VITA	IX
ABSTRACT OF THE DISSERTATION	X
INTRODUCTION	1
CHAPTER 1: LOOKING-GLASS HOUSE	4
CONCEPTUAL FRAMEWORK.....	4
<i>The Creative Artist</i>	4
<i>The Creative Process</i>	5
HISTORICAL BACKGROUND.....	6
<i>On Computer Music</i>	6
<i>On Computer History</i>	7
<i>On Quantum Computing</i>	9
PREVIOUS WORK ON QM/QC AND THE ARTS.....	10
<i>On QC/QM and Art</i>	10
<i>On QM and Music</i>	12
<i>Learning/Teaching QM and Music</i>	13
<i>Composition and performance with QM</i>	14
<i>A case for a richer palette of sounds</i>	15
<i>On QC and Music</i>	16
MY OWN WORK.....	17
<i>On Music and Painting</i>	17
<i>Initial connections to QC/QM</i>	18
<i>The Human and the Machine</i>	19
<i>Moving forward</i>	21
CHAPTER 2: QAC	22
COMPUTER MUSIC TOOLS AND QUANTUM COMPUTING TOOLS.....	22
PREVIOUS ATTEMPTS FOR AN INTEGRATION.....	24
NEW QAC TOOLS.....	29
OCH.QC-CIRC-2Q.....	31
OCH.MICROQISKIT.....	33
PERFORMANCE OF OCH.MICROQISKIT.....	35
OCH.QISJOB.....	38
PERFORMANCE OF OCH.QISJOB.....	41
IMPLICATIONS OF QAC.....	45
CHAPTER 3: QAD AND MORE	47

REVISITING THE INPUTS AND OUTPUTS OF QC	47
ADVANTAGES OF QC-ENABLED ART	53
<i>Computational power and complexity</i>	53
<i>Thinking differently</i>	55
ENCODING IMAGES AND SOUNDS IN QC.....	56
QAD	58
<i>qc.VFX</i>	60
<i>QuantumSynth reforged</i>	62
A QC-ENABLED AVANT-GARDE.....	64
CHAPTER 4: THE GEDANKEN ROOM	67
THE GEDANKEN ROOM.....	67
4 DISKLAVIER PRELUDES.....	69
<i>Disklavier Prelude #1</i>	70
<i>Disklavier Prelude #2</i>	73
<i>Disklavier Prelude #3</i>	75
<i>Disklavier Prelude #4</i>	81
TELEMATIC ENTANGLEMENTS.....	83
SECOND CORNERSTONE	91
DECOHERENCE	96
INTERLUDES AND THE LARGER NARRATIVE	100
IN RETROSPECT	103
CONCLUSIONS	106
BIBLIOGRAPHY	110
APPENDIX A: QC.VFX ENVIRONMENT	120
APPENDIX B: QC.VFX'S QUANTUMBLUR TOX.....	121
APPENDIX C: STORYBOARD FOR <i>THE GEDANKEN ROOM</i>.....	122
APPENDIX D: SLIDES FOR <i>TELEMATIC ENTANGLEMENTS</i>	123
APPENDIX E: SCORE FOR <i>SECOND CORNERSTONE</i>.....	125
APPENDIX F: POSTER FOR <i>THE GEDANKEN ROOM</i>.....	129

LIST OF FIGURES

Figure 1 - Architecture of Quantum Synth. From (Hamido, Cirillo, and Giusto 2020)	25
Figure 2 - Quantum Synth’s demonstrations for Bell state circuit. From (Hamido, Cirillo, and Giusto 2020)	26
Figure 3 - second GUI for the Quantum Synth presented at (“Festival Della Tecnologia - Musica Con Un Computer Quantistico” 2019)	27
Figure 4 - The interactive quantum vocal system architecture. Reproduced by permission from Eduardo Reck Miranda (2020b, fig. 15)	28
Figure 5 - Earlier generation of fully integrated device	30
Figure 6 - example device chain using och.qc-circ-2q together with a drum rack, och.Param2Param, and och.probGate	32
Figure 7 - och.microqiskit object example patch.....	34
Figure 8 - Comparing MicroQiskit, och.microqiskit, and Qiskit simulating 2,000 shots	36
Figure 9 - Comparing MicroQiskit, och.microqiskit, and Qiskit simulating 20,000 shots	37
Figure 10 - Comparing MicroQiskit, och.microqiskit, and Qiskit simulating 1,000,000 shots	37
Figure 11 - och.qisjob object example patch.....	40
Figure 12 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 2,000 shots	41
Figure 13 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 20,000 shots	42
Figure 14 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 1,000,000 shots.....	42
Figure 15 - All execution times for all the tools and architectures being compared.....	43
Figure 16 - Computer Music Tools and Quantum Computing Tools, and with some specific tools being compared, in context	45
Figure 17 - a quantum circuit a, b, and c	48
Figure 18 - Bloch spheres representing quantum circuit a	49
Figure 19 - och.qsphere representing quantum circuit b	50
Figure 20 - histogram retrieved from running quantum circuit c	51
Figure 21 - histogram from running quantum circuit c on ibmq_athens real QPU device	52
Figure 22 - comparison of QSS with Monte Carlo. Reproduced by permission from Eric R. Johnston (2019, fig. 11.1)	59
Figure 23 - framework for the implementation of qc.VFX.....	61
Figure 24 - frames of the interactive interpolation of QuantumBlur in qc.VFX.....	62
Figure 25 - QuantumSynth v3 saw circuit using statevector backend with 2 and 6 qubits	63
Figure 26 - QuantumSynth v3 saw circuit using simulator backend with 4 and 4096 shots	64
Figure 27 - Still frame of <i>Disklavier Prelude #1</i> . From (Hamido 2021b)	70
Figure 28 - initial notes and harmony in <i>Disklavier Prelude #1</i>	71
Figure 29 - Super spread self standing dominant chord in <i>Disklavier Prelude #1</i>	71
Figure 30 - Still frame of <i>Disklavier Prelude #2</i> . From (Hamido 2021b)	73
Figure 31 - the harmonic progression in <i>Disklavier Prelude #2</i>	74
Figure 32 - excerpt from Scriabin’s "Waltz in A-flat major Op. 38" from 1904	74
Figure 33 - Still frame of <i>Disklavier Prelude #3</i> . From (Hamido 2021b)	75
Figure 34 - Still frame of <i>Disklavier Prelude #3</i> . From (Hamido 2021b)	76

Figure 35 - One automation line triggers och.qc-superposition to control transposition, the other automation line triggers another och.qc-superposition to control a stream of notes on Jazz me	78
Figure 36 - Still frame of Disklavier Prelude #3. From (Hamido 2021b)	80
Figure 37 - Still frame of Disklavier Prelude #4. From (Hamido 2021b)	81
Figure 38 - Still frame of Telematic Entanglements. From (Hamido 2021b)	83
Figure 39 - Still frame of Telematic Entanglements. From (Hamido 2021b)	84
Figure 40 - och.scorestream uses MIDI clips to trigger the change of slides that are sent to performers connected with any browser enabled device	86
Figure 41 - och.qc-circ uses MIDI clips to generate quantum circuits whose result triggers one of the MIDI clips in the track on the left. In turn these MIDI clips trigger a change in och.scorestream	87
Figure 42 - Still frame of Telematic Entanglements. From (Hamido 2021b)	90
Figure 43 - Still frame of Second Cornerstone. From (Hamido 2021b)	91
Figure 44 - Still frame of Second Cornerstone. From (Hamido 2021b)	93
Figure 45 - Reverb effect in Second Cornerstone	94
Figure 46 - och.qc-superposition controlling FX in Second Cornerstone	95
Figure 47 - Still frame of <i>Decoherence</i> . From (Hamido 2021b)	96
Figure 48 - QuantumSynth v3 (left) a simulated square circuit and (right) interface options	98
Figure 49 - Still frame of <i>Decoherence</i> . From (Hamido 2021b)	99
Figure 50 - Still frame of <i>Interlude #2</i> . From (Hamido 2021b)	100

ACKNOWLEDGEMENTS

I need to express my deep appreciation for all the attention and guidance I received from my advisor, Professor Michael Dessen. He has been both a mentor, a source of inspiration, and a friend, who helped me greatly in articulating my work. I also need to acknowledge the most exciting committee I could ever wish for: Professor Chris Dobrian, Professor Sandy Irani, Professor Kojiro Umezaki, and Dr. Christine Corbett Moran. I'm grateful for their attention and feedback, and for really engaging with my creative work.

In addition, I would like to acknowledge the other professors in my program that also accompanied me closely, Nicole Mitchell, Lukas Ligeti, Mari Kimura and her very welcoming family, as well as my colleagues that I had the pleasure to work with. I'm thankful for all the support I received while at UCI, particularly Dr. Stephen Tucker, Peter Chang, and Dean Gillian Hayes for always being available. All of them, together with the many interesting individuals I was able to connect with at UCI, as well as the support of close friends like Pedro Matias, Yubo Dong, Norbert Tsi, Andrea Sloan Pink, among others, made me feel that I was in the right place.

Throughout the chapters of this document, I mention the work of several other artists and researchers, some of which I've had the pleasure to directly interact with, or even collaborate with. Permission has been granted to use figure 4 and figure 22. Thank you to Eduardo Reck Miranda and Eric Johnston for allowing me to use their figures, and for their words of encouragement.

I acknowledge the use of IBM Quantum services for this work, as well as the access to advanced services provided by the IBM Quantum Researchers Program. The views expressed are those of the author, and do not reflect the official policy or position of IBM or the IBM Quantum team. Special thanks to James Wootton and Sebastian Hassinger for pointing me in the right direction.

Being a part of the IBM Qiskit community, and the various different Quantum Computing hackathons, were all incredibly enriching experiences that not only greatly helped in my learning process of QC but also allowed me to make new good friends and collaborators along the way. I would like to thank my friends and hackathon partners, some of whom I went on co-authoring papers with, among others: Giovanni Amedeo Cirillo, Scott Oshiro, Jielun (Chris) Chen, and Jack Woehr who I am in debt to for painstakingly reading my code with me and providing insights.

I would also like to thank the xMPL Board and staff members for granting me the 12 days to record my film-recital. And most importantly, I want to thank, again, the stellar and very diverse cast of performers that gave body to this creative work: Will Vinson, Tina Raymond, Niloufar Shiri, Waeli Wang, and JNTHN STEIN. The production of this film-recital was also a

tour de force thanks to the core production team collaborators, Grant Speich and G. Blake Harrison-Lane, to whom I am very grateful for their dedication and work.

I consider myself very fortunate to be able to pursue this research project which would never be possible without the support received from UCI, IBM, Fulbright, and the 2019-21 support from the Portuguese Foundation for Science and Technology (FCT).

VITA

Omar Costa Hamido

Omar is a performer, composer, and technologist, working primarily in multimedia and improvisation. His current research is on Quantum Computing and Music Composition, telematics, and multimedia. He is passionate about emerging technology, cinema, teaching, and performing new works. In recent years, his work won several relevant awards: Fulbright Exchange Visitor, Beall Center for Art+Technology – 1st Prize Student Artwork, IBM Qiskit Camp Europe – Community Choice Award.

- 2011 B.A. in Music – Jazz – Saxophone, Universidade de Évora, Portugal
- 2013 M.A. in Music Theory and Composition,
Escola Superior de Música, Artes e Espectáculo (ESMAE-IPP), Portugal
- 2016-17 research & development and music technology,
Portuguese Music Research and Information Center/ Miso Music, Portugal
- 2019-21 Teaching Assistant/Associate, University of California, Irvine
- 2021 Ph.D. in Integrated Composition, Improvisation, and Technology,
University of California, Irvine
- 2021 Research Assistant in Quantum Computing and Music,
University of Plymouth, U.K.

(SELECTED) PUBLICATIONS

Hamido, Omar Costa. 2021. *The Gedanken Room*. Music.
<https://www.imdb.com/title/tt14443510/>.

Hamido, Omar Costa, Giovanni Amedeo Cirillo, and Edoardo Giusto. 2020. "Quantum Synth: A Quantum-Computing-Based Synthesizer." In *Proceedings of the 15th International Conference on Audio Mostly*, 265–68. AM '20. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3411109.3411135>.

Oshiro, Scott, and Omar Costa Hamido. 2020. "A Quantum-Classical Network for Beat-Making Performance." *Journal of Network Music and Arts* 2 (1).
<https://commons.library.stonybrook.edu/jonma/vol2/iss1/4>.

ABSTRACT OF THE DISSERTATION

Adventures in Quantumland

by

Omar Costa Hamido

Doctor of Philosophy in Integrated Composition, Improvisation, and Technology

University of California, Irvine, 2021

Professor Michael Dessen, Chair

An increasingly diverse community of researchers worldwide is currently tackling the challenges of understanding and finding use cases for the new computational paradigm of Quantum Computing (QC). In this dissertation, the author explores the implications of QC for new modes of artistic creation, with a focus on music composition and performance. Chapter 1, "Looking-Glass House," provides a conceptual framework for the dissertation, reflecting on the role of the creative artist with respect to the dialogue between science and art. The author offers a brief overview of modern computer music tools as well as a survey of recent work in the field, relating music and visual arts to Quantum Mechanics and QC. Chapters 2 and 3 focus, respectively, on what the author terms Quantum Computing Aided Composition (QAC), and Quantum Computing Aided Design (QAD), including a discussion of what QC can offer to the artist-researcher. This dissertation also introduces *The QAC toolkit* software package that allows musicians and artists to build, simulate, and run quantum circuits using the Max visual programming language. Chapter 4 analyzes the techniques and aesthetic choices, inspired by and making use of QC, present in *The Gedanken Room* film-recital. Through research-based writing, original software, and the production of a film-

recital, *Adventures in Quantumland* engages both with the artistic and QC community, exploring avenues for the development of new QC-based tools informed by realtime creative practices.

Introduction

“Let's pretend the glass has got all soft like gauze, so that we can get through. Why, it's turning into a sort of mist now, I declare! It'll be easy enough to get through” She was up on the chimney-piece while she said this, though she hardly knew how she had got there. And certainly the glass was beginning to melt away, just like a bright silvery mist. (Carroll 1872, 10)

Sometimes, it is hard to explain how I got here. A fire starting overnight, a relationship evaporating during the day, a virus getting out of control... and, pursuing a PhD in Music and Quantum Computing? Perhaps they were all inevitable. Sooner or later, a fire will start, a relationship will end, a virus will spread, and someone will write on music and quantum computing. Admitting this inevitability should be no reason to despair, however. For even in a deterministic Newtonian world, humans, as open systems, can still be free as they develop the sense of self (Ismael 2016).

The current document gives an account of my research which represents more than a technical challenge: it emerges from my inner necessity. It is an attempt to make sense of music, art, emerging technology and ideas that have kept me up at night and daydreaming. These ideas are greatly influenced by my previous research work on the relationship between music and painting, and the need to make sense of the deeper essence of artistic means of expression and creative processes, while recognizing what it has in common with Nature itself. This work is also greatly informed by my creative practice as a performer-composer-technologist where I strive to create new music that is truly contemporary - as in music that is both a reflection of its own time, and something that could not be done in any other moment in history.

Quantum Computing (QC) is an emerging technology that has seen significant advancements in recent years, and has a promising roadmap for the near future (Wehden, Faro, and Gambetta

2021). Both my conceptual and practical work has led me to QC. As I said elsewhere for an interview “I came to quantum computing a little bit by accident - but it was also inevitable at the same time.” (Hamido 2021a). It was inevitable because it is part of a trajectory that is both personal and professional, of expanding and deepening interdisciplinary interests that bring together Art, Science, Technology, Philosophy, in search of the nature of reality.

I decided to focus my current research on the integration of QC with my creative compositional and improvisational practice, asking the following question, and sub-questions:

- What are the implications of newly emerging paradigms in Quantum Computing for new modes of artistic creation?
 - How can Quantum Computing potentiate new and innovative kinds of compositional and improvisational structures and processes?
 - How can realtime creative practices inform the development of new Quantum Computing based tools?

These research questions are contextualized and addressed from different angles, in 4 main chapters. In chapter 1 “[Looking-Glass House](#),” I reflect on the role of the creative artist, and introduce the field of Quantum Computing and how it is paving its way to the art world. This chapter encompasses a review of relevant literature and artistic work, including my own previous work.

In chapter 2 “[QAC](#),” I present the tools that I developed to pursue what I called Quantum Computing Aided Composition, or QAC for short. I also evaluate some new affordances that these tools might provide to musicians and how it relates to the broader landscape of artistic and QC programming environments.

In chapter 3 “[QAD and more](#),” I start by reviewing the basics of working directly with QC and very briefly mention some methods for encoding images and sounds in QC. The chapter continues exploring a couple methods for designing images and sounds with QC, that give body to a Quantum

Computing Aided Design (QAD). Some special attention is also given to discuss the advantages of a QC-enabled art, and what a QC-enabled Avant Garde means.

In the last main chapter, chapter 4 "[The Gedanken Room](#)," I analyze the creative work that accompanies this document, and include descriptions of the tools that I developed and how they are being used. I also give an account of the different collaborations with performers and creative team, as well as the creative process behind each piece and the entire recital narrative.

In the "[Conclusions](#)," I will try to elaborate a more systematic synthesis of the answers to the research questions that are addressed throughout the work. I review the role of the tools and creative work, as well as the advantages of working with QC and possible future work directions.

Returning to my initial metaphor, even though we might live in a classical, mind-independent world, I hope my observations (and measurements) here will be well received by the very diverse community that I'm engaging with. Certainly, from my perspective, the glass is melting away and, on this chimney-piece that I am standing now, I am able to see these new and exciting interdisciplinary connections taking place.

Chapter 1: Looking-Glass House

This is a unique moment of transition in the history of computing. In this chapter, I am both looking at a recent past and possible new futures, inquiring into the role of the creative artist and the relationship with his or her tools, in the context of the changes that Quantum Computing (QC) presents. This chapter includes 4 main sections that provide a series of different contexts for this work. The first section, "[Conceptual framework](#)," addresses "[The Creative Artist](#)" and "[The Creative Process](#)"; the following section, "[Historical background](#)," includes subsections "[On Computer Music](#)," "[On Computer History](#)," and "[On Quantum Computing](#)"; the third section, "[Previous work on QM/QC and the Arts](#)," explores work in the field with subsections "[On QC/QM and Art](#)," "[On QM and Music](#)," "[Learning/Teaching QM and Music](#)," "[Composition and performance with QM](#)," "[A case for a richer palette of sounds](#)," and "[On QC and Music](#)"; finally, the context provided by my own previous work, as well as my own artistic values and goals, is addressed on the last section, "[My own work](#)," that includes "[On Music and Painting](#)," "[Initial connections to QC/QM](#)," "[The Human and the Machine](#)," and "[Moving forward](#)."

Conceptual framework

The Creative Artist

Musicians and artists, as creative practitioners, will use whatever means they have available to create their artistic work. Given a stronger emphasis on a certain aspect - visual, aural, sculptural, etc. - each practitioner could be identified more as a painter, a musician, a sculptor, and so on. And even though different fields of science and artistic disciplines might once have been regarded as equivalent ways to understand the universe, as with the *quadrivium* in the Middle Age ("Quadrivium"

2020), the modern world is mostly characterized by an hyperspecialization (Malone, Laubacher, and Johns 2011).

I am not to argue one in detriment of the other, for I believe specialization has brought to light testaments to the capacity of human expression that would be deemed impossible, while at the same time I recognize the beneficial insight raised by those who were able to understand and relate seemingly isolated events and phenomena, helping others make sense of the encompassing universe.

I will thus argue that the true creative artist has in fact a double role: to understand the universe, and to acquire the skills to express that understanding. This definition might as well fit the description of the astronomer or the physicist. But, being the purpose of my attention devoted here to musicians and artists, I will further define this model by focusing on a differentiation aspect in their approach: the creative process.

composers are more than artisans or entertainers. Like scientists they try to understand and explain the universe and our place in it (Tipei 1989)

The Creative Process

Experimentation, Composition, and Improvisation, are three defining creative approaches that musicians undertake to pursue their goals. These can mean different things for different artists. For example, Composition may include notes on a paper, lines of code on a machine, instructions shared verbally, or even applying physical constraints to the performers. In effect, sometimes Improvisation could be regarded as meta-composition, or even meta-experimentation (Dean 2003, 84). It is also interesting to trace such creative approaches in the visual arts, for example in the way the painter Wassily Kandinsky organized his own artistic work: Impressions, Improvisations, Compositions (Kandinsky 1982, 218).

The tools used to carry the creative process will play a decisive role. The affordances of these tools define the possible outputs, but can also play an *input* role feeding back into the creative process. An acoustic instrument will offer the affordances that its resonant body and mechanical parts will allow. And a *bodiless* instrument like a computer will offer different kinds of affordances. The latter is a special case that interests me particularly, as I am proposing new work in the field of computer music.

Historical background

On Computer Music

Making music with a digital computer was something that only happened in 1950, with CSIRAC, the first digital computer in Australia (“Computer Music” 2020), roughly 10 years after the first Turing-complete electromechanical computer, the Z3 (“Computer” 2020). However, an important mark for the evolution of the field of musical artistic creation was when Bell Labs researcher Max Mathews created the MUSIC programming language, which first played a 17 second composition on a IBM 704 machine in 1957.

And if the computer, in these early stages of computer music, was mostly regarded as a machine to be commanded, to execute orders, in the late 1970s to mid 1980s computers started sharing the stage with the performer-composers. A prime example is the KIM-1 (“KIM-1” 2020) which was used in early networked music performance by the League of Automatic Music Composers, used for an interactive pitch track-enabled system by David Behrman, and even took a role of a musical partner itself in the case of George Lewis' *Voyager* (Dean 2011, 70; Lewis 2000). This was possible, one should note, mostly also thanks to the smaller form factor, more computing power, and easier accessibility of these new machines. These advances in realtime music making were out of reach with state of art computing in the 1950s, as Max Matthews notes:

The IBM 704 and its siblings were strictly studio machines – they were far too slow to synthesize music in real-time. (“Max Mathews” 2019)

For the most part, up until the 1990s, computer music dealing with signal processing seemed to only be developed by those who had access to an electronic music studio lab, not to mention the know-how to operate them.¹ This was then circumscribed to Universities or other research centers who would both have the capacity to get the expensive equipment, and would allow time for the creative practitioners to explore it. Electronic musical instruments were being developed and commercialized all the way through the 1970s and 1980s (“Electronic Musical Instrument” 2020), but the possibility for creative artists to directly access and control the underlying Digital Signal Processing (DSP) was still very limited.

A most notable tool in this regard was the introduction of the MSP component² to the Max visual programming environment (“Max (Software)” 2020). Originally derived from program routines written for a custom build signal processing workstation at IRCAM (the ISPW), MSP is a set of 200+ objects that work as operators or small functions which one can use to build his or her own custom digital circuit, or fully fledged digital signal processing program (Zicarelli 1998; “MSP: Introduction” n.d.). Together with the increasingly accessible, general purpose, personal computers, this toolkit, and other equivalent toolkits that came out in the meantime and around the same time, marked a new era for the creative artistic practice that is still in place today.

On Computer History

As the field of computer science has evolved, so have the affordances of these machines and the applications explored. One such example that has found impressive consequences in today's

¹ For a more complete chronology of computer music events see Appendix in (Dean 2011).

² Which stands for "Max Signal Processing", or Miller Smith Puckette, its original inventor.

world is Machine Learning (ML). With its first incubation period also in the 1950s,³ ML has only recently started to find more far reaching toolkits for musicians and artists after Rebecca Fiebrink's *Wekinator* in 2009, and Google's *Magenta* project in 2016 (“Wekinator | Software for Real-Time, Interactive Machine Learning” n.d.; “Welcome to Magenta!” n.d.). The former is a Java-based standalone application that allows one to quickly build and train different types of ML models. The latter is a research project which is continually putting out very relevant research on ML and the Arts, but most notably they offer a Javascript-based toolkit, called *Magenta Studio*, that, while being easily integrated with the Digital Audio Workstation (DAW) Ableton Live via the Max for Live environment, provides several machine learning algorithms for music generation.

It is worth noting that these last two projects mentioned have their source code publicly available. For even if one is not equipped with the systems that hold the computing power to fully take advantage of each application, at least this will allow the exploration of the underlying processes and conceivably adapt or even extend them (*Magenta/Magenta-Studio* [2018] 2020; Fiebrink [2015] 2020).

Quite possibly the next big breakthrough in the world of computation now is the advent of fully programmable quantum computers. A quantum computer is a machine that performs computation by harnessing Quantum Mechanics (QM) phenomena, like superposition and entanglement (“Quantum Computing” 2020). Quantum Mechanics itself has been a driving force of inspiration for many musicians and artists (Turner n.d.). And very important research centers worldwide, such as CERN, have sometimes promoted a direct dialogue between Art and Science via artistic residencies, as a way to tackle “new approaches to research, experimentation and artistic production”, and help understand these new scientific grounds themselves (“Arts at CERN” n.d.). Perhaps such endeavors signify a sort of a new, post-modern, version of a quadrivium-like paradigm,

³ followed by another in the 1980s and another in the 2000s (“Timeline of Machine Learning” 2020).

or at the very least reflect concerns such as Kandinsky's, when comparing the different means of artistic expression:

Deeply concentric, each art is separated from the other, but on the other hand, they are combined in their innermost tendencies. (Kandinsky 1946, 36)

On Quantum Computing

In turn, Quantum Computing (QC), as an emerging technology, has recently been the object of an increased interest due to a massive new availability of quantum processing units (QPU) (“List of Quantum Processors” 2019). The idea of a quantum computer was first proposed by Benioff in 1980, and by Feynmann in 1982, who argued that quantum mechanical phenomena could only be effectively simulated by using quantum mechanical phenomena (Akama 2015, chap. 1). David Deutsch (1985; 1989) proposed the model for a quantum computer in 1985, and the quantum gate in 1989. And finally, Bernstein and Vazirani (1993) proposed a universal quantum Turing machine.

Computationally speaking, quantum computers obey the Church-Turing thesis, meaning that while a quantum computer should be able to solve any computational problem that a classical computer can solve, the reverse should also be true (“Quantum Computing” 2020). Their main difference then, as in the advantage that QC offers over classical, lies in the fact that quantum computers, due to their different mechanics, will be able to solve certain types of problems much faster than their classical counterparts.

One of the earlier QPUs is the 2-qubit quantum computer developed by Gershenfeld and Chuang at MIT (1997; “Quantum Computing” 2000). The currently available devices range from 1 to several dozens of qubits each, being most of the publicly available equipped with 5 qubits, and they are machines that occupy a large space in a temperature-controlled room. Relatively speaking QC is in its infancy and reveals similarities with the early classical computers: they both are very large machines, and very limited (the Z3 that I mentioned earlier had only 22 bits!). Making QPUs available

in the cloud was first explored at University of Bristol (“Quantum in the Cloud” 2013), and then later made massively available by IBM in 2016 (Gil 2016), and other tech companies that followed the same path, like Rigetti in 2017 (Simonite 2017), and Amazon in 2020 (“Quantum Computing Is Now Available on AWS through Amazon Braket” 2020).

This is a very important distinction of several of the currently available QPUs: the fact that they are publicly available via the cloud. This has been crucial to create a community around QC (Gambetta and Chow 2016), and it is new fertile ground for the creative artist as well. Among the very relevant avenues, where this community building has been taking place, are the hackathons. Several of them also openly invite people with diverse professional backgrounds, other than physics and computer science. Examples of these hackathons include the “Quantum Futures Hackathon” at CERN, “QuID” at ETH, “Qiskit Camp” by IBM, “qhack” by Xanadu, and “iQuHACK” by MIT (“Quantum Futures Hackathon” n.d.; “Hackathon | Quantum Information for Developers” n.d.; IBM Corp. n.d.; “QHack” n.d.; “IQuHACK 2021” n.d.).

Games are actually a very popular focus of recent quantum hackathons projects. One can see in (Huang [2019] 2021) how the number of quantum games has been growing exponentially. This has led to papers such as the ones on procedural map generation by James Wootton (2020a; 2020b). And it has also even led to new University courses being taught, like the one on Quantum Games at Aalto University (Kultima n.d.).

Previous work on QM/QC and the Arts

On QC/QM and Art

In the Art world, QC has been in use for some time. It is important to note, however, that initially it has been mostly marking its presence only metaphorically, often with representations of

the concepts introduced by QM. Lynden Stone in her paper (Stone 2013) calls for a reinterpretation of reality informed by quantum phenomena, and expressed in visual Art, summoning Niels Bohr, Heisenberg, David Deutsch, and Karen Barad as the first prominent figures in science and philosophy to call for this necessary revision. She navigates through the artworks of several artists.

Jonathon Keats' *Quantum Entanglements* (2011) installation work uses "a nonlinear crystal beam splitter [using] sunlight to spray entangled pairs of photons onto couples." Julian Voss-Andreae's *Quantum Man* (2007) and *Night Path* (2009) explore unmeasured superposition and visualize Feynman's "path integral" calculations, respectively. Antony Gormley's *Quantum Cloud* (1999) sculpture series encompass the concept of becoming real when observed. And Daniel Crooks' experimental video *Static No. 12* (2010), according to Stone's perspective, embodies perfect metaphors for quantum superposition, even though it was not part of the original artist's articulated intention.

Similar to Voss-Andreae, physicist turned artist, Libby Heaney, has drawn on metaphors with QM to explain her work (Heaney 2019). But, unlike the previous artists, her recent work now uses QC to generate some of her digital renderings (Heaney 2020). Such is also the approach taken by Alain Lioret, French digital artist and technologist, who has generated digital sculptures using Sympy, QuTip, and PyCTQW with Blender. He mistakenly repeated that "a quantum computer does not exist" to explain the need to use QC simulation to achieve these results. But he also presented the very interesting idea of using qubits to navigate digital artworks and perform operations at the level of "the pixels in an image, the frames of an animation, the vertices of a 3D object, the words of a text or the notes of a musical score" (Lioret 2016). He admits being less interested in "observing the exact scientific formulations" but, rather, in using quantum algorithms to achieve "new experiences" (Lioret 2014).

And while some designers like Russell Huffman are starting to create reproducible paper prints using QC (Huffman 2021), others, like media artist and composer JoAnn Kuchera-Morin, are designing audio visual immersive installations for a 3-story high metal sphere multimedia instrument (Kuchera-Morin et al. 2017). Sometimes it is difficult to understand the artworks and how they are created, but it is also interesting to note that, at the same time, several of these artists also advocate for using their artistic creations as a means to understand QC and QM.

On QM and Music

In the Music world, too, there have been several different works and explorations that deal with QC and QM. Some of these works fall into what I would like to call strictly conceptual music. I am using this term to refer to ideas for musical works that exist only as concepts and are never meant to be realized or heard. This is the case in the paper that received a good deal of attention by the Mathematics Professor Putz and Physicist Svozil. They propose a new colorful notation to represent quantum tones, a collection of notes in a quantum musical state that, hypothetically, when played should render different music in the ears of the different listener-observers. There is no mention to how any of this could be put into practice, the authors themselves claim to be pursuing a “strictly non-artistic, non-aesthetic approach” with no concern for making it “useful or necessary for aesthetics, or for musical expression” (Putz and Svozil 2017). Elsewhere it has been suggested that “an ordinary computer and a set of headphones” might work well to simulate this effect, but such approach has not yet been formalized (“First Quantum Music Composition Unveiled” 2015).

Still in the conceptual realm, Italian philosopher of science Maria Luisa Dalla Chiara, together with Giuntini and Negri, wrote about a “quantum-like holistic semantics for musical scores” where they use QM to conceptualize the musical score. But perhaps even more intriguing is Dennis Gabor's 1947 paper titled “Acoustical Quanta And The Theory Of Hearing.” In it, Gabor describes a

mathematical representation of sound alternative to the ones provided by sound as a signal $s(t)$ - i.e. a function of time -, and the timeless Fourier description $s(f)$ - a function of frequency. He uses both t and f in a new mathematical formulation that includes the uncertainty relation between both. His concept of a “quantum of sound” is then put to the test with some (psychoacoustic) experiments that confirm thresholds of time discrimination and frequency discrimination (Gabor 1947).

On the topic of sound, more precisely, there have been recent practical experiments that directly use QM. Chris Holloway is using Rydberg atoms to record and playback radio signals. He is able to record in multitrack simply by using different atomic species and using different laser frequencies, and has plans to record an album at the lab (Holloway et al. 2019). And a very recent publication is observing quantum phenomena in phononic crystals that may be used for “exploring emerging macroscale systems to be used in applications such as on-chip logic devices for sound manipulation, acoustic signal processing, and sound energy harvesting” (X. Jiang et al. 2020).

Learning/Teaching QM and Music

This articulation between Music and QM has been explored numerous times also to teach QM through Music. James Gimzewski’s Ars Electronica workshop is an example of that (Gimzewski 2020). The Institute for Quantum Computing, at the University of Waterloo, and the Kitchener-Waterloo Symphony have organized a narrated concert that explores the history of music and quantum science over the past century (Outwater 2012). Some of these juxtapositions incidentally work well to explain one and another aspect, but they didn’t originate together.

Similar to what Lynden Stone reported about Daniel Crooks’ work, witnessing music that is inspiring to think about the concepts of QM or QC, but whose original intention was not to relate to it, Miguel Zenón’s *Identities are Changeable* project had the same effect on me. After the quartet

concert version at the Hot Club Lisbon in 2015, I had a very brief opportunity to talk to Zenón and realized he had never thought of such connections, though this was still very inspiring to me:

Miguel Zenón's music is like a spinning electron! Spinning in multiple directions and at the same time. In quantum mechanics, measurement breaks the quantum effect of superposition of the subatomic particle (the multiplicity), in Zenón's music the superposition of measures gives us the idea of his music's multiplicity. Sometimes it's inside a certain measure, other times it's inside another measure, and some other times it's in both at the same time. One could say that it's only a matter of measurement. (Hamido 2015)

On the other hand, there have been a couple music compositions that only relate to QM in the title. Such is the case of Norbert Stein's "Quantum Mechanics" track on his *Silent Sitting Bulls* album (Stein et al. 2010). And it is also the case for Nick Herbert's "Quantum Number - Bell's Theorem Blues" composition that mysteriously appears at the end of his book on Quantum Theory, as the second appendix, with no other explanation in the text (Herbert 1985). Several decades after the publication of his book, Herbert admits in a blog entry that it was written as a joke (Herbert 2014).

Composition and performance with QM

Composer Kim Helweg articulates the different possible ways to draw inspiration from QM to compose music in three approaches. First by using "Quantum Narratives" as "the theoretical texts about quantum mechanics as inspiration for titles, lyrics or program notes." Then, "Sonifications and Frequencies" by performing "simple musical translations of quantum measurements." And lastly, "Other possibilities," that refers to the use of probabilities derived from quantum measurements (Helweg 2018). Helweg was the composer of *Super Position* for 2 pianos (2017), written for the LP Duo.

In turn, LP Duo wrote about this new hybrid instrument they performed that included a digital synthesizer with analogue controls: 90 micro-controllers and magnetic sensors to convert "movements of keyboards into MIDI commands received by electronic instruments or computers"

which hold “a bank of quantum sounds” (Lončar and Pavlović 2018). A revision of the articulation between the composer, performers, and engineers, as well as the concert and context where it was presented, can be found in Medić and Janković-Beguš paper, where they state:

Helweg's composition does not introduce a completely new musical language in comparison with his earlier pieces, [...] what is truly new and original in Helweg's piece, aside from the use of the hybrid piano prototype, is the penetration of music into the world of experimental physics, both as a new source of inspiration for composers, and as an outcome of the physicists' desire to “musicalize” i.e. make audible this distant subatomic world. (Medić and Janković-Beguš 2018)

A case for a richer palette of sounds

These last few papers emerged in a special issue of *Journal Musicology*, which was part of one of the most concerted collective efforts on music and QM/QC, the project *Quantum Music*. This project, co-funded by the Creative Europe Programme of the European Union, produced concerts, workshops, papers, collaborations, and other philosophical discussions, like Chiara Marletto's, where she questions if this marriage of science and art brought upon by the *Quantum Music* project can “permit the achievement of a richer set of aesthetic outputs,” and bring us closer to objective beauty (Marletto 2018).

The project is now on its second edition with an open call where they distribute the software that was used by Helweg and LP Duo, the *Quantum Oscillator*. This is a Max for Live device based on the work of Novković, Peljević, and Malinović on synthesis with the Bose-Einstein Condensate. In their paper, they observe that:

The movement of atoms and the speed of their oscillations at such extremely low temperatures led to the generation of extremely “slow” waves, which are in the frequency range [...] of human hearing. This surprising situation gave us the right to conduct direct translations [...] without the need for additional scaling. (Novković, Peljević, and Malinović 2018)

This list of frequencies derived from the BEC experiment might be precisely what was in use in the “bank of quantum sounds,” mentioned above in Lončar and Pavlović’s “Hybrid Duo” paper. That, and some sounds generated with the Max for Live device, which uses the also mentioned “mathematical formulas that model the energy-vibration condition of the condensate” (Novković, Peljević, and Malinović 2018).

Having had the opportunity to open the device myself, I can see that it is not using QC simulation but only a handful of functions to generate a series of frequencies. It was built by a programmer who is most likely not native to realtime computer music and the Max environment, for they are creating and destroying dsp objects by scripting in javascript. Because javascript runs on the lower priority thread in Max, this may be, indeed, the underlying reason why the user needs to “wait for few seconds [*sic*] before sound is generated”, as warned by the authors, on the release webpage (“Open Call - Beyond Quantum Music” 2020). This calls for new tools, better integrated, and more performant, as I will try to address in the next chapter.

On QC and Music

Also on the topic of sonification, but now more closely related to QC, composer and sound artist Spencer Topel was selected for the year-long artist-in-residence program at the Yale Quantum Institute, where he recorded and sonified data coming from superconducting quantum devices. He also performed live with it, together with Yale Physics graduate students Serniak and Burkhart (“YQI Artist: Spencer Topel” 2019). And more on the topic of music notation, composer Adam Bell, found in QC inspiration to achieve what he has been looking for, to solidify his musical language: “ways in which music can be modelled and formed by adopting a clinical, emotionally removed and scientific approach to the creation of art objects which explore mathematics and science, capitalism and the market, the visual arts, politics and Utopia” (Bell 2019).

A first concrete example of using QC in music notation might be the one found in Hendrik Weimer's quantenblog. Here he is using his C library for quantum computing simulation, libquantum, and provides audio recordings, midi scores, and a short description of his compositional/programming process (Weimer 2010).⁴ However, more thorough examples and explanations of using QC for generating musical notes and scores can be found in the work of James Weaver and Christine Corbett Moran, who have both, each on their own, used small musical compositional challenges to teach QC (Weaver 2018; Moran 2019, chap. 8).

More recently, composers Alexis Kirke and Eduardo Reck Miranda, at the University of Plymouth, have also been composing music that uses QC. In Kirke's *Superposition* (2016), QC is used to generate an accompaniment to a mezzo-soprano, and in Miranda's *Zeno* (2019) to generate an accompaniment to a bass clarinet (Kirke 2018; Miranda 2020a). This newer generation of music and composers using QC is marked by an increasing interest in using QC as a Computer Assisted Composition device. In effect, during the next several years we might witness different Quantum Computing Assisted Composition schools emerging, the same way it happened for its classical counterpart.⁵

My own work

On Music and Painting

My own work sits in this continuum. I've arrived at the need to study QM and QC as a natural development of the conceptual framework and compositional ideas that I was exploring during my previous research on Music and Painting and compositional processes. In the search for a true

⁴ It is an interesting coincidence to note that Weimer published this shortly after releasing his paper about a universal quantum simulator using Rydberg atoms (Weimer et al. 2010).

⁵ For a short review of different Computer Music approaches between 1956 and 1986 see (Ames 1987).

correspondence between Music and Painting,⁶ I decided to deconstruct the material means of artistic expression of both Music and Painting in, metaphorically speaking, subatomic particles. I was left with dual components of what makes an image, and a sound, that, taken individually, aren't sufficient to define an image or a sound, in its entirety.

From here, I proposed a compositional model I called the *Abstract Rhythm Model*, which consisted of abstract symbols indicating temporal relationships visually, and allowed me to write what I then called a "universal score" for a new given piece. In turn, this piece can be interpreted both musically and pictorially. Such was the case with my piece *qup* (2013) for drums and saxophone with resonator tube, or for a painter.

The conceptual framework, that I graphically represented as " $< | >$ ", included this pre-existence stage, or the realm of ideas, but also an existence stage, or human realm, and a reduced, or post-existence stage, or machine realm. The second stage encompasses all the possible and coexisting media, as well as the human interactions and free correspondence interpretations of it. And the last stage, encompasses all the correspondences that are imposed mechanically, via an algorithm, and beyond any human intervention (Hamido 2013).

Initial connections to QC/QM

A year later, Google released the Quantum Playground and I became aware of both QC and the serendipitous resemblance of Dirac notation (Culp and Wroblewski 2014). For a short while, I've tried to explore this and QCL, the first quantum computing programming language (Ömer 2014). But I failed at my attempts to integrate it with my electro acoustic music, and to understand it. I was

⁶ "True correspondence" here is taken as true to the nature of the artistic means of expression, and not dependent on the subjective interpretation of it. This in the light of the conceptual framework and theorization given in (Hamido 2013).

aware, however, that it was going to become something very important in the future. My own music and approach to composition had already started seeing this change, influenced by this new paradigm.

My *qup* (2013) piece relies on the performers to make decisions on which notes to play and how to signal the metric modulations to be able to advance through the score together. And one of my more recent pieces, *The Space Between the Strings* (2019), also relies on the performers to make collective instantaneous decisions to be able to articulate the unpredictable sound variations of each instrument (due to the piano preparations, or the chaotic electronics) and convey the composed sense of direction and intensity with precise timings (Hamido 2020a).

Dale E. Parson's idea of a "musical system as a quantum system" resonates with my own work, with its approach to composition and improvisation, and its interdisciplinary connections. In his paper he defends a "framework of a composition as superposition of states [*sic*] that cascade to music via interaction with performers" which resembles my *Abstract Rhythm Model* and the idea behind the universal score that I mentioned before.

A related idea, though aimed at Machine learning and not QC, is Miller Puckette's reflection that "[...] it's impossible to capture the gestures that the performer for whatever reason decided not to make. In other words, the gesture exists inside a space of possibilities but we can only observe the gesture itself, not the space it lives in" (Puckette 2017).

The Human and the Machine

The human performer is very important in my music. And the reason for wanting to use the computer in no way resonates with the idea of wanting my music to be devoid of feeling. On the contrary, the human-machine interaction serves the purpose to simultaneously stimulate the

performer both cognitively/intellectually and emotionally. If anything, again and again, in Kandinsky's words I do find resonance.⁷ Here, in his "Thesis-Antithesis-Synthesis" text from 1935:

Just as all the recipes in the world can never succeed in creating a work of art by themselves, so too they can never supplant the kind of feeling that is vitally necessary in order to "understand art." One's head is not a bad device. But an "unfeeling" head is worse than a "headless" feeling. At least in art. (Kandinsky 1982, 773)

This perspective is also reflected in my practice as an improviser. Oftentimes I've come across a free improvisation scenario where my first concern was to be able to inhabit a shared musical environment, and to be able to jump between atmospheres instantaneously and together. I've argued that being able to listen, understand, and reply using the same musical grammar is more effective (and intellectually challenging) than just playing random contrasting material.⁸

Integrating QC into my creative process is my next attempt to explore these ideas from a different perspective; which might include, indeed, exploring the third stage of my original conceptual framework "< | >". Given my creative practice with emphasis on realtime electronics, I am personally interested in first exploring ways to think about composition with quantum circuits. And this not only being limited to generating musical notation material, but also controlling FX parameters, and even building new sounds for virtual instruments.

⁷ What's more? Even though my idea of "true creative artist," presented at the start of this chapter, does not appear literally in his treatise, it could very well be present there, somewhere. He does, however, use the expression "true work of art" in passing, when commenting his own statement of what should be the goal of the artist: "The artist should have a message to convey; mere mastery of form should not be his goal, but rather the adaptation of form to inner contentment" (Kandinsky 1946, 93).

⁸ Perhaps this has been so at the cost of not being able to find myself more often in such improvisatory scenarios, for others would prefer to just have everyone playing their own thing. And maybe emotionally that's the difference between being alone together or being together alone.

Moving forward

In some way, moving forward will be similar to retracing the steps of composers like those mentioned in Ames (1987), figuring out how to use QC to assist music composition and creative practice in general. There is still no MUSIC, MSP, Magenta, or wekinator-like toolkit for current QPUs, and contributing to such a development is a key goal in my current work. However, I don't foresee such a new toolkit replacing the tools that musicians currently use, the same way electronic music instruments didn't replace acoustic instruments, MSP didn't replace electronic music instruments, and Magenta didn't replace MSP. But I do foresee these tools that musicians and artists currently use being expanded with new QC-enabled tools. What these will look like, is what I will be discussing in the following chapters.

My current method is a method of exploration, experimentation, skills and toolkit development, making research tools that are also composing and performing tools. In the next chapter, I will present what those tools are, and what they look like, in the context of Quantum Computing Aided Composition (QAC).

Chapter 2: QAC

As presented in the previous chapter, the creative artists who intend to use the new emerging technology of quantum computers will have to find or create new tools for integrating QC in their creative practice. In this chapter, I will focus on the computer music practice, specifically, and its distinct artmaking characteristics and requirements for integration with QC. In order to do that I will first map out the different available tools in "[Computer Music tools and Quantum Computing tools](#)," and consider the "[Previous attempts for an integration](#)." Then, I present the tools that I developed and the reasoning behind them in "[New QAC tools](#)," and a reflection on its first generation with "[och.qc-circ-2q](#)." The 4 sections that follow will address the current implementations that are included in the final version of the developed software toolkit,⁹ including performance testing - "[och.microqiskit](#)," "[Performance of och.microqiskit](#)," "[och.qisjob](#)," and "[Performance of och.qisjob](#)." To finish this chapter, there is a short section, "[Implications of QAC](#)," where I review the importance and implications of this toolkit, the collaboration it might enable, and what might be next in line for this.

Computer Music tools and Quantum Computing tools

Computer Assisted Composition¹⁰ (CAC) history is as long as the history of computers and music, and it can include many different approaches to use the computer for sound and music production, and be articulated into different stages (Dean 2011). Though they have evolved

⁹ *The QAC toolkit*, available at quantumland.art.

¹⁰ Also mentioned in some references as "Computer Aided Composition". From now on, noted as CAC.

significantly, over decades of continuous development, current modern tools still have some limitations for the realtime control of compositional processes. Depending on the complexity of the compositional process, the musical output may not arrive on time, and renders some score instructions impractical (Bouche et al. 2017). Here there might be an avenue where Quantum Computing’s potential increase in processing speed can be an important contribution to the Computer Music field.

Within the current context, and to address my quest of integrating QC with Music Composition, I surveyed the modern Computer Music tools and Quantum Computing tools. Computer Music tools include environments such as score engravers like Musescore, Digital Audio Workstations (DAW) like Ableton Live, and programming environments like Max/MSP.¹¹ And though I offer these three categories here as different examples, the distinction between them may not always be very clear. Quantum Computing tools, on the other hand, include the different programming frameworks (most of them based in the Python programming language) like Qiskit, Cirq and pyQuil,¹² some web apps that allow the design of circuits online like Strangeworks, IBM Quantum Experience, and QPS,¹³ and, because of the emphasis in Python, Quantum Computing scientists often use Jupyter notebooks to write and execute their code.¹⁴

This second group of tools, based on a more traditional computer programming paradigm, could still be regarded as inheriting the “punch card” computation paradigm insofar as they resemble a similar multi-step process of writing the code, submitting the code to be executed, and waiting for the results to come back. On the other hand, within the Computer Music tools group, we often see the

¹¹ See, respectively (“MuseScore” n.d.; *Ableton Live* 2021; *Max* (version 8.1.10) 2021)

¹² See, respectively (Jay Gambetta et al. 2021; Gidney et al. [2017] 2021; Karalekas et al. [2017] 2021)

¹³ See, respectively (“Strangeworks | Quantum Computing Ecosystem” 2021; “IBM Quantum Experience” n.d.; “Quantum Programming Studio” 2018)

¹⁴ For a more complete list of current Quantum Computing tools see (Vogt-Lee [2018] 2021; *Open-Source Quantum Software Projects* [2018] 2021).

predominance of a realtime computation paradigm, where the program is being changed as it is being executed.

As a performer-composer-technologist, my creative practice is strongly informed by this realtime computation paradigm. As a consequence of that, and of my recurring interest in creating new tools and integrating emerging technologies, my toolkit must include the visual programming languages that are focused on enabling this approach for creative artists. For this reason, I end up using both Max and TouchDesigner (“Derivative” n.d.) environments.¹⁵

Previous attempts for an integration

Unsurprisingly, the first few attempts to integrate QC with Music Composition have been strongly modeled around the paradigms for, and mostly within the environments of, the QC tools. The immediately apparent reasons for this are the fact that these tools are very new, and the fact that most of the people who have attempted to engage in such endeavors are coming from a Computer Science background, and so would prefer to use the tools they already use for their non-artistic practice. This is not very inviting for creative artists, who don’t have the same background, as they have to overcome considerable language barriers here. I can relate to this given my experience 8 years ago, when I first started exploring quantum computing with the then very obscure programming language QCL (Ömer 2014), with limited knowledge about traditional programming languages, and even more limited knowledge about quantum computing itself.

A second generation of tools for this integration started to emerge when some programmers decided to either develop software architectures including both the computer music tools and the QC tools (combined as a software stack), or when they tried to recreate new environments altogether.

¹⁵ Other programming environments worth considering for this type of work may include Pure Data (Puckette n.d.), SuperCollider (“SuperCollider” n.d.), and OpenFrameworks (“OpenFrameworks” n.d.).

Examples of this include my own work on the first generation of the Quantum Synthesizer (Hamido, Cirillo, and Giusto 2020), a project that was created in 48 hours at the Qiskit Camp Europe hackathon, in September 2019 (IBM Corp. 2019a). In it, I devised this architecture that uses Max as the frontend where the user can select all the parameters that are sent via OSC to a Python environment (on the same machine or another machine in the local area network). The Python environment, running Qiskit, can then create the circuits and either run them on a local simulator (with or without noise model), or send them to a real machine online, then wait for the results, and send them back to Max via OSC (see figure 1).

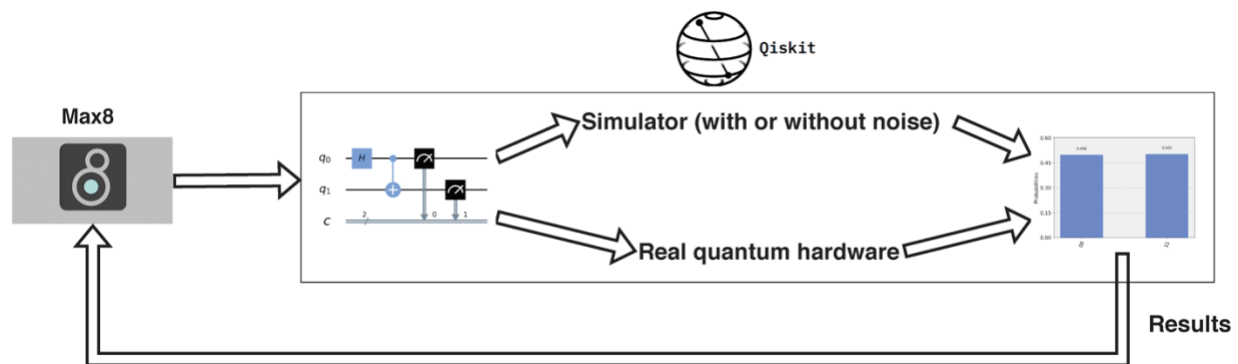


Figure 1 - Architecture of Quantum Synth. From (Hamido, Cirillo, and Giusto 2020)

This is a multi step and multi software process, which can work, but requires the user to be able to set up their Python environment, launch all the applications in the correct order, and ensure that they are talking to each other (see figure 2). I have experimented with more friendly user interfaces (see figure 3), but it never removes the fact that the user needs to account for setting up the backend.

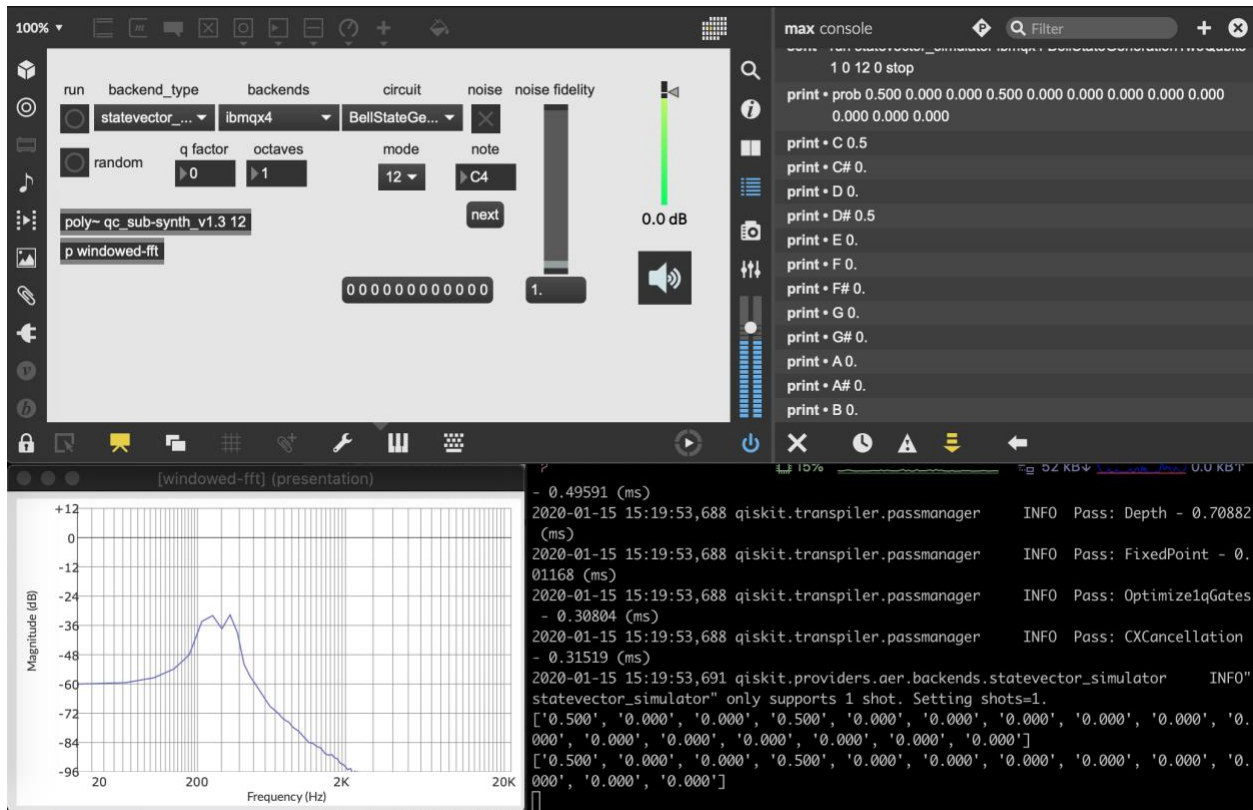


Figure 2 - Quantum Synth's demonstrations for Bell state circuit. From (Hamido, Cirillo, and Giusto 2020)



Figure 3 - second GUI for the Quantum Synth presented at (“Festival Della Tecnologia - Musica Con Un Computer Quantistico” 2019)

This multi-software architecture is also present in the recent work by Eduardo Reck Miranda (see figure 4). In it, there is also an interface for musicians which is independent from, but connected to, the QC process. However, Miranda’s approach is more based on the Quantum Computing tools, relying heavily on Python and Jupyter notebooks. The musical output is managed via Csound scripts that are launched from the Python program as well (“Csound Community” n.d.; Miranda 2020b, 17). Though this approach allows the development of these new tools to be more close to the Quantum Computing framework, guaranteeing that it is up to date with the referenced QC framework, the

learning process for musicians, and the serious constraints imposed for realtime use, may be a considerable barrier for musicians.

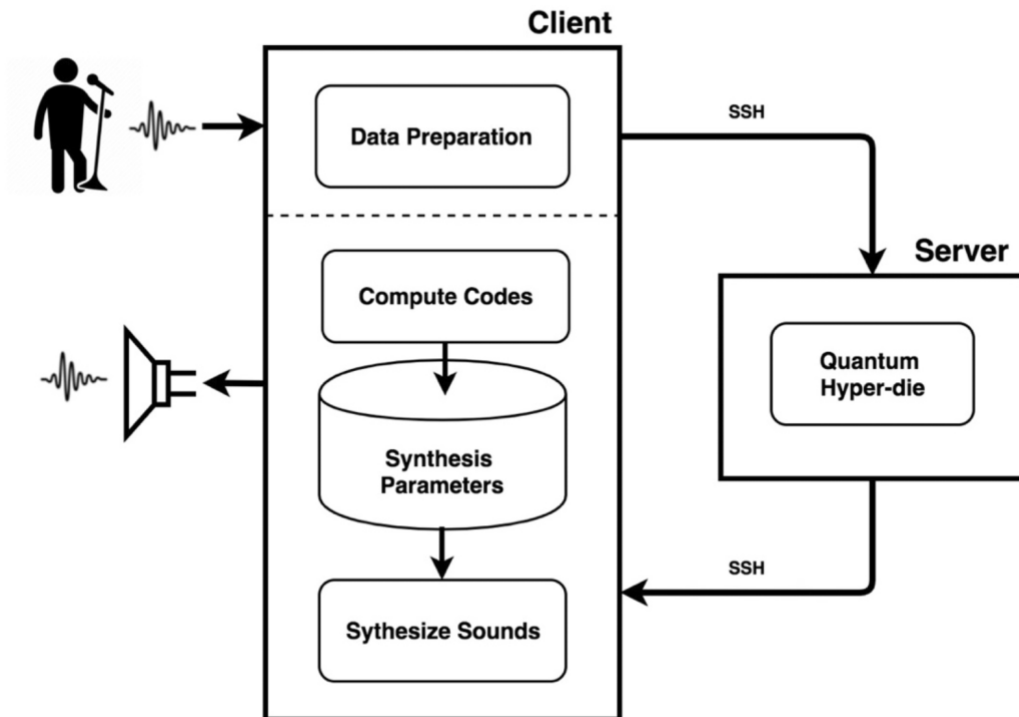


Figure 4 - The interactive quantum vocal system architecture. Reproduced by permission from Eduardo Reck Miranda (2020b, fig. 15)

Another type of approach has been to create entirely new CAC applications that use QC. Such is the approach taken by James Weaver in his web application “Quantum Music Composer”. In it, Weaver created his own interface that allows the user to generate 3rd species counterpoint melodies, based on melody and harmony matrices (Weaver [2018] 2018). The generated output is Lilypond code that can be rendered into a readable musical score using the Lilypond music score engraver (Daniels et al. 2021). This architecture is still complex, requiring the musician to know how to properly deploy and configure this app, and its use is limited to what the software currently allows the user to do. The fact that the software is released as open source opens the possibility for it to be

changed, to accommodate new compositional tasks. But the amount of non-music and non-QC required skills to be able to change it make it somewhat less practical as a tool for new compositional work.

New QAC tools

Moving forward, the next generation of tools for QAC to arrive must be more tightly integrated with the Computer Music tools, and the music composition workflow. In effect, there are still 2 approaches to take. One is to make use, and extend, more developed music libraries native in Python, such as music21 (Cuthbert and Ariza 2010), Abjad (Bača et al. [2013] 2021), or SCAMP (Evanstein 2021).¹⁶ And the other is to create new tools that are native in current Computer Music environments, abstracting away any of the required underlying “software infrastructure” setup, to communicate with real quantum processors in the cloud. If the former might require a considerable effort, creating good documentation to make it accessible for musicians, the latter will take advantage of the language of these environments that is already familiar to musicians. In the current work I focused on the latter.

In my research and creative practice, I use the Max visual programming environment extensively, so this was the platform I chose for developing these new QAC tools. This choice was based on the fact that Max is not only widely used as a Computer Music programming language in itself, but it is also tightly integrated with the Ableton Live DAW, which many other non-programmer electronic musicians use. Thus, I am making this choice in a similar spirit as to what the Magenta team at Google did when choosing to build Max for Live devices, to reach a wider audience.

¹⁶ For more context about these tools see, for example, Tymoczko’s review of music21 (2013), and Evanstein’s comparison between algorithmic composition tools in (2020).

My first attempts at creating these native tools in Max relied on incomplete (now archived) javascript based frameworks (Rubio et al. [2017] 2019). Thanks to the new integration of Node.js into Max in late 2018 (Grosse 2019), and the also important integration of Max in the Ableton Live DAW, via MaxforLive since 2012 (“Max for Live | Ableton” n.d.), I was able to not only natively integrate this javascript based framework into Max, but also create an easy to use, plugin-like, drag-and-drop device with it, in early 2019 (see figure 5). In part due to the limitation of the underlying framework, this device is limited to using a reduced number of qubits, and only in simulation mode.

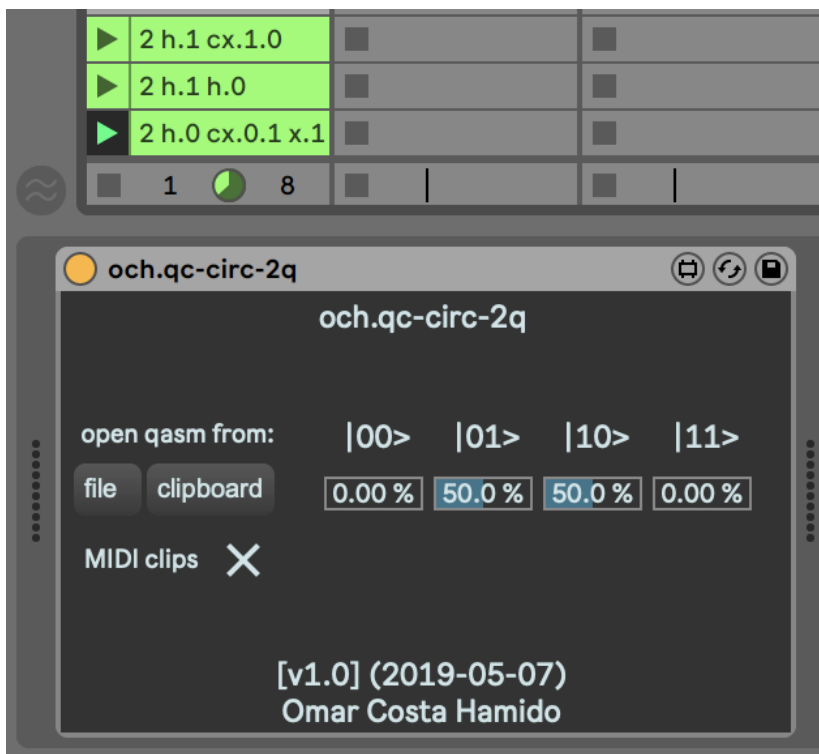


Figure 5 - Earlier generation of fully integrated device

och.qc-circ-2q

The **och.qc-circ-2q**¹⁷ device was hard coded to work with only 2 qubits, and the way to change its state was by loading a qasm file,¹⁸ retrieving qasm code from the clipboard, or reinterpreting the name of the playing midi clip, as a new minified circuit notation that I invented for this research. The underlying mechanism calculates the statevector¹⁹ for the given circuit and translates that into the predicted probabilities for retrieving each possible computational-basis state that the 2 qubits can offer. From here, the user needs to use other devices to map these results into actually changing musical parameters. An example of this can be seen in figure 6, where I used multiple **och.Param2Param** devices to map the percentage of each of the states to **och.probGate** devices - which act as probability gates that allow the MIDI note to pass (and trigger a sample) according to the set probability.

I should also note that the direct translation of the statevector into these *clean* probabilities is not a very helpful way to think and learn about quantum computing, since the output of the real machines will only approximate these values and not consistently give these even probabilities. In terms of QC, the statevector of a qubit is something that we can only read in simulation. As a probabilistic means of computation, QC's probabilities are revealed when the results of the multiple circuit executions are aggregated. These are also affected by noise on the real machine, which sometimes is modeled to be used in high performance simulators. As explored on the Quantum Synth

¹⁷ The new Max devices and objects will be highlighted in bold for ease of identification while reading. They all start with "och.x" because the short name of the toolkit package is och, and default Max package structures prepend this short name to each external object and abstractions. Some object mappings will be available too: e.g. *och.microqiskit* can also be initialized simply by writing *microqiskit*.

¹⁸ Qasm stands for "quantum assembly language". For more information see (Cross et al. 2017).

¹⁹ "Statevector" (one word) is in effect a "state vector" (two words), but the former formulation has been largely adopted for use in the QC world and literature.

paper, between simulators and real machines, there are 4 different outputs we can use in QC: statevector, simulation, simulation with noise model, real machine execution.



Figure 6 - example device chain using och.qc-circ-2q together with a drum rack, och.Param2Param, and och.probGate

The creation of this device was a very important milestone in this research project. However, the strong limitations of the underlying library, for which the possibility to connect to real quantum processors online was a feature that never became possible to fully implement, not to mention the fact that the project was officially archived at some point, ultimately mandated that I had to abandon this approach. Other important factors in this decision were the increasing library dependencies that made the compilation of this device up to 10 minutes long at some point. For simple circuits, with a reduced number of qubits, it could calculate the statevector for each new circuit fairly quickly - as in usable in a realtime scenario - but the same was not always true for higher qubit count circuits.

It was only after this that I became more active with the IBM Qiskit community online, and started participating in hackathons (in person, before the pandemic), including the one where I lead a team, of mostly Italian engineers, creating the aforementioned Quantum Synth (IBM Corp. 2019a). By late October of 2019 I became a Qiskit Advocate (Qiskit n.d.), in early December of 2019 I gave a presentation on my research to other advocates and IBM researchers, and in mid December 2019 I was invited as a mentor for the Qiskit Camp Africa, in South Africa (IBM Corp. 2019b).

och.microqiskit

Inspired by the work of Christine Corbett Moran (Moran 2016a; 2016b) and James Wootton (J. Wootton 2019), I realized that one of the best ways to learn about quantum computing is by actually building your own quantum computing simulator. Thanks to the earlier involvement with the Qiskit community I was able to work with James Wootton on creating a new version of the MicroQiskit library in C++ (Hamido and Wootton 2021). This was pursued mostly between December 2019 and December 2020, at the same time that I was learning some more of C++, and it is now part of the official Qiskit community repository.

All this work paid out when I was able to create my own native Max object, written in C++, to simulate quantum circuits. In effect, this was the reason for starting the work on MicroQiskit C++. For the reference, MicroQiskit is a simplified version of Qiskit, that only runs a simple simulator locally, does not connect to real devices in the cloud, and does not include any other features like noise characterization. The **och.microqiskit** object is just that. And it is a tool that has been in the making since November 2019, and whose development also informed my work with MicroQiskit.

Figure 7 shows an example patch using **och.microqiskit**. The user only needs to create the object instance, just like any other object in Max, and interact with it by sending several different messages. These messages need to contain certain keywords that are recognized by the object, like “QuantumCircuit” and “Simulator” to create a new quantum circuit or simulator, respectively. The format of these messages are in such a way that it still partly resembles the code other programmers would use when working with regular Qiskit in Python. The advantage is, of course, that these messages can also be generated programmatically using the tools that the Max environment provides.

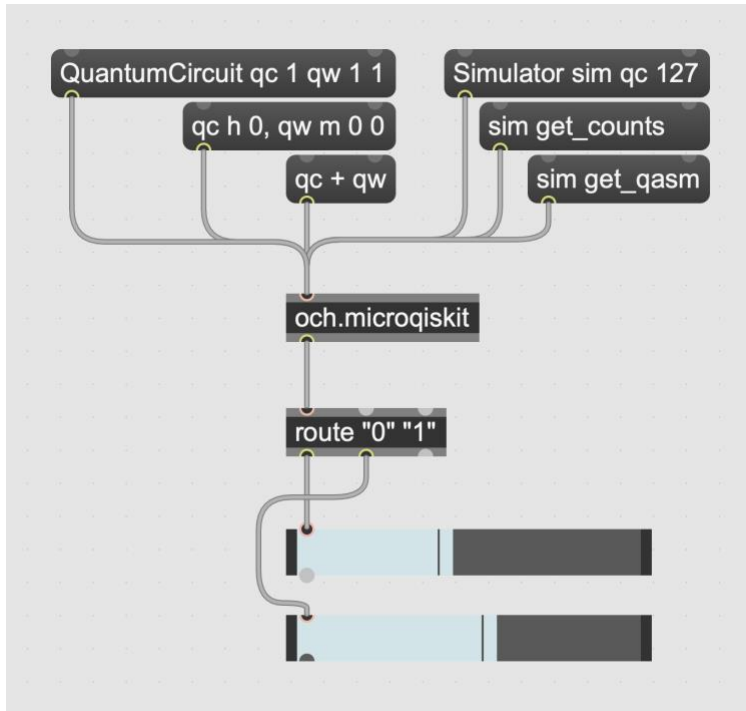


Figure 7 - och.microqiskit object example patch

The structure of these messages is also flexible in some ways that allow both different programming styles and some efficiency with less typing. For example, the user can create circuits by giving them any name, which will later be recognized by the **och.microqiskit** object to refer to the same quantum circuit memory. When creating quantum circuits the user might opt to define only the number of qubits, or set both the number of qubits and classical bits. Multiple circuits, with different configurations, can be created in one go, as shown in figure 7, and they can later be added together, making the first circuit contain the gates of both circuits being added. The user can add gates to a circuit by sending a message with the circuit name, followed by the gate name, and the qubits it is to be applied to with some additional parameters (like rx rotation value).²⁰

²⁰ The list of available gates include: x, rx, h, cx, ox, ch, oh, ccx, oox, ocx, cch, ooh, och, crx, m, rz, ry, z, y. “x” is also known as the NOT gate, “cx” the controlled-NOT gate, “h” is the hadamard gate, and “m” is the measure gate. A complete list of quantum gates and good quick sources for understanding them include (Bernhardt 2019, chap. 7 Quantum Gates and Circuits; Asfaw et al. 2020b; Koch, Wessing, and Alsing 2019)

Once the quantum circuit is defined, we need to pass it to a simulator, giving it a name and a number of shots, and using the previously determined quantum circuit name. This new simulator memory contains the circuit with the configuration it had when it was passed to the simulator, unless the *sim_update* attribute is set to ON, which will automatically update the simulator (a special feature of this object). Once the simulator is defined, we can retrieve different results like the aggregated counts using *get_counts*, the statevector using *get_statevector*, or each individual shot result using *get_memory*.²¹

An important part of the development of this object was the implementation of error checking. While an earlier version of this object would, under certain circumstances, allow the user to construct some impossible circuit configurations, which led to unexpected results, error checking prevents this and helps the user better learn how to use this tool. For example, if the user tries to call a circuit with a name that has not been previously set on this object instance session, it will reply with an error message on the console. The object will also post error messages if, for example, the user tries to add a gate to a circuit on a qubit that is outside of range, tries to run a simulation without measurement gates, or tries to add measurement gates on circuits without classical bits to store them.

Performance of och.microqiskit

With this object running natively in Max, the artist-researcher can integrate it in any other patches with any other tools available in this growing programming environment. It will be interesting to see what and how other users will use this tool to advance the world of CAC and bring

²¹ For a complete list of methods for the och.microqiskit object, please refer to the help patch and reference page included in the distributed Max package.

to life what I called, by extension, QAC, as in Quantum Computing Aided Composition. This is the reason why I decided to release this tool as part of a Max package.²²

Given this object running natively in Max, and with all the concerns that went into making it work seamlessly, I decided to put to the test exactly how well it is running compared to the similar mainstream QC tools. I devised a simple experiment using a very small circuit with only 1 qubit and 1 classical bit, and containing 1 hadamard gate followed by a measurement gate. This setup was duplicated and executed several times, in simulation, using the **och.microqiskit** object in Max, the MicroQsikit in Python terminal, and the full Qiskit also in Python terminal. I ran this experiment first for 2,000 shots (figure 8), then 20,000 shots (figure 9), and finally 1,000,000 shots (figure 10).

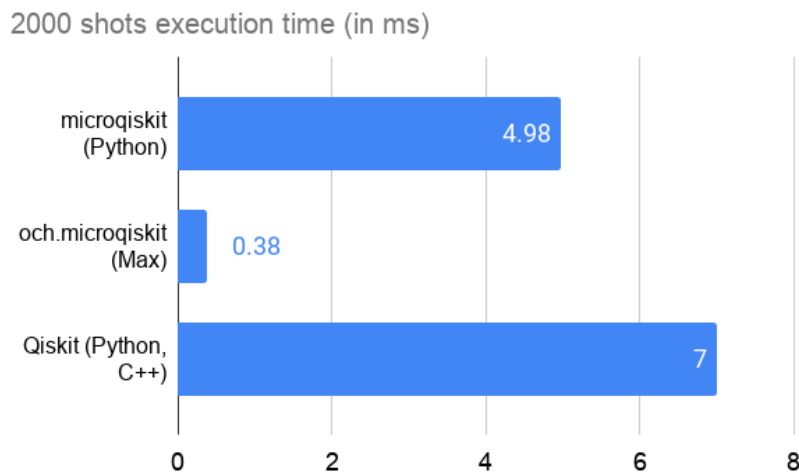


Figure 8 - Comparing MicroQsikit, och.microqiskit, and Qiskit simulating 2,000 shots

²² The latest version of *The QAC toolkit* Max package is available at quantumland.art.

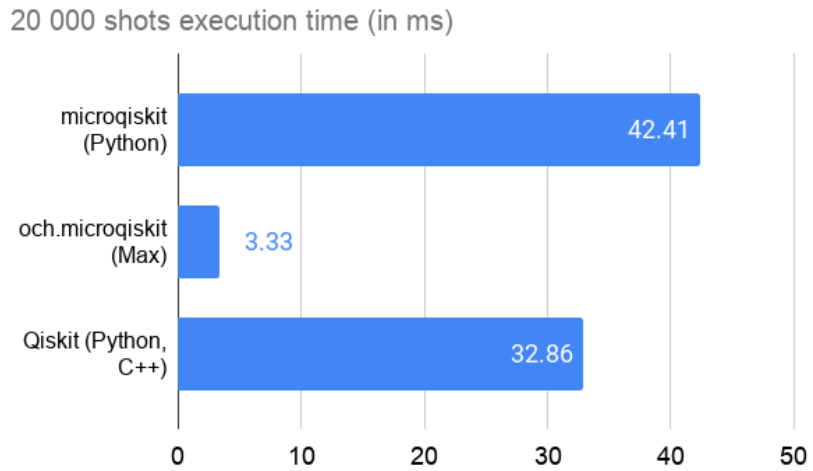


Figure 9 - Comparing MicroQiskit, och.microqiskit, and Qiskit simulating 20,000 shots

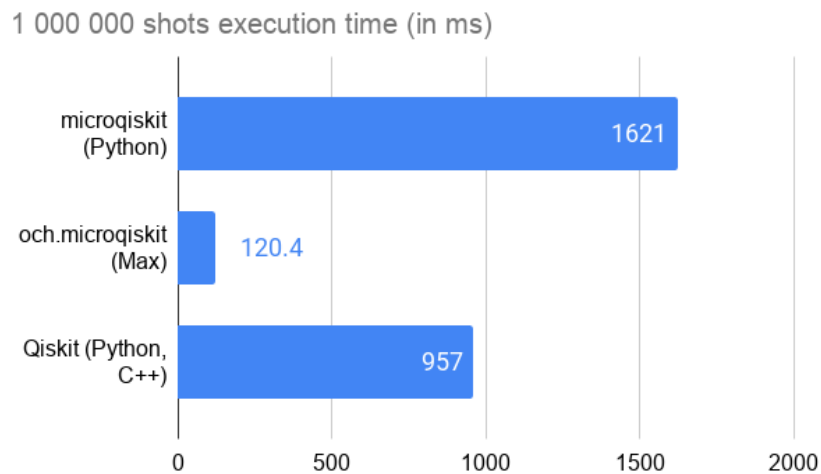


Figure 10 - Comparing MicroQiskit, och.microqiskit, and Qiskit simulating 1,000,000 shots

The results were very surprising: **och.microqiskit** was consistently 10 times faster than MicroQiskit. This fact can be understood when we take into account that C++ is much more efficient than Python.²³ For 2,000 shots, MicroQiskit is around 40% faster than Qiskit. This is known among

²³ Evidence of this has been explored in the literature such as (Prechelt 2000; Fourment and Gillings 2008; Jain et al. 2020).

the Qiskit community that, for very simple circuits MicroQiskit can be faster than full Qiskit, since it has much less dependencies to load. However, when we get to 20,000 shots, Qiskit is already faster than MicroQiskit. The reason for this is also because Qiskit uses high performance simulation components (some written in C++ even). During the last experiment, with 1,000,000 shots, which is the maximum number of shots that Qiskit allows for a single experiment execution, Qiskit is about 70% faster than MicroQiskit, and **och.microqiskit** is still 10 times faster than MicroQiskit and 8 times faster than Qiskit.

Because all of these experiments are done while simulating quantum computation with classical computers, none of them can represent an advantage in processing speed when compared to running other classical processes in classical machines.²⁴ However, the results obtained here in terms of optimization and simulation processing speed are relevant enough to be highlighted. For the computer scientist, this gain in speed might not be relevant, but for the artist-researcher, using realtime electronics, it will make a very noticeable difference, potentially opening up new doors for their realtime creative practice.

The next major breakthrough that I needed in my research was the ability to easily connect to the real quantum computing machines in the cloud, which will eventually allow one to exploit quantum advantage at some point. As discussed earlier, my goal was to remove the need for the user to manually setup a Python environment and establish connections between that environment and the Max environment.

och.qisjob

However much I would have liked to not use Python at all, the amount of work required to recreate all the library components that allow the remote connection was just out of reach for me,

²⁴ For more about the advantages of QC for the creative practices, please see "[Advantages of QC-enabled Art](#)" in the next chapter.

and even today no one else is interested in doing that. I've explored very old Max objects that should be able to interface with Python directly, but most of them would either only work with older versions of Python, which are not supported by the modern QC frameworks, or they would still need the user to have a Python installation on their machine. I have also started exploring the possibility of setting a web server with a Python installation running the QC framework that is able to relay the commands to the real machines, a process similar to one I explored in my QC-enabled interactive story in 2019 (Huang, Hamido, and Campo [2019] 2020). But the concerns about making it available to multiple users at the same time, and managing potential sensitive login credentials, made me put a hold on this avenue.

I realized that I only really needed a simple interface to submit jobs. A job, in quantum computing lingo, is the set of instructions to run on the quantum computer, which include the quantum circuit definition and the number of shots, or number of times to execute the circuit. The solution came, once again, through the interaction with the Qiskit Community. In particular, my good friend and hackathon partner Jack Woehr had a project that he had started also in early 2019 that was just that: a simpler command line interface for submitting jobs to quantum computing hardware (Woehr and Hamido 2021).

During late 2020 and early 2021, I worked both contributing to Jack Woehr's project repository and integrating it into the Max environment. Finally, also after some additional study of C++ threading (Williams 2019), I was able to make it work into the new **och.qisjob** object. In effect, this is a special object that works in tandem with a platform specific executable dependency that is included in the Max package. In the end, the user doesn't have to worry about setting up Python environments, and never has to leave the Max environment. Because this object is only focused on managing jobs, the user can still use the very performant **och.microqiskit** to generate the quantum circuits (see figure 11).

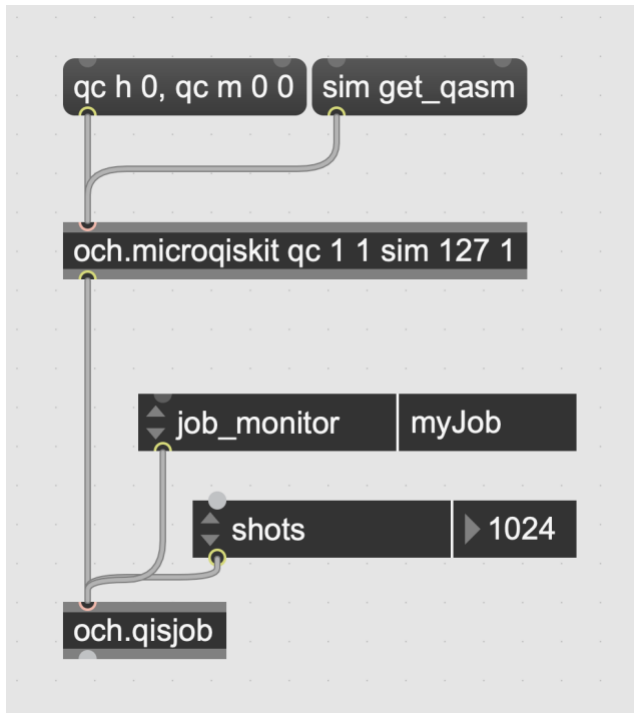


Figure 11 - och.qisjob object example patch

Since this object is expecting openQasm code, we can request **och.microqiskit** to export its contents as qasm code using the *get_qasm* method.²⁵ The **och.qisjob** object has an attribute to set the number of shots for the job, an important parameter to control, given that backend providers often impose some limitations here. It also has an attribute to set a job monitor with a given name, which will keep track of the job and keep the user updated about its place in the queue for execution. There are additional parameters that the user can set, such as which backend to use, or whether to use an online, or local, high-performance, Qiskit-based simulator instead.

²⁵ The och.microqiskit object includes methods to retrieve the circuit definition in Qasm, using *get_qasm*, and Qiskit Python code, using *get_qiskit*. Both of these methods offer the possibility to explore and reuse the current circuits in other platforms and can be easily copied from a popup text box when the method is called in a message that includes the keyword *textbox*.

Performance of och.qisjob

Similar to what I've done for **och.microqiskit**, I decided to test how well this new object runs compared to the previous strategies for doing this type of work - connecting a Computer Music tool to the real QPUs. However, because the submitted jobs on the real machines will always end up at different places in the queue, there isn't a straightforward way to test the time it takes to complete the job. For this reason, I decided to test the efficiency of this object running the local high-performance simulation. In the case of **och.qisjob**, I only need to pass some additional arguments to enable the use of a local *qasm_simulator* that is provided by the included compiled package dependency. And for the alternative strategy, I set up a patch that sends a message, via OSC, to a Python environment that runs the job on the local Qiskit simulator, and sends the results back to Max via OSC. This, in effect, is the strategy used in the mentioned previous works. I ran this experiment first for 2,000 shots (figure 12), then 20,000 shots (figure 13), and finally 1,000,000 shots (figure 14).

2000 shots execution time (in ms)

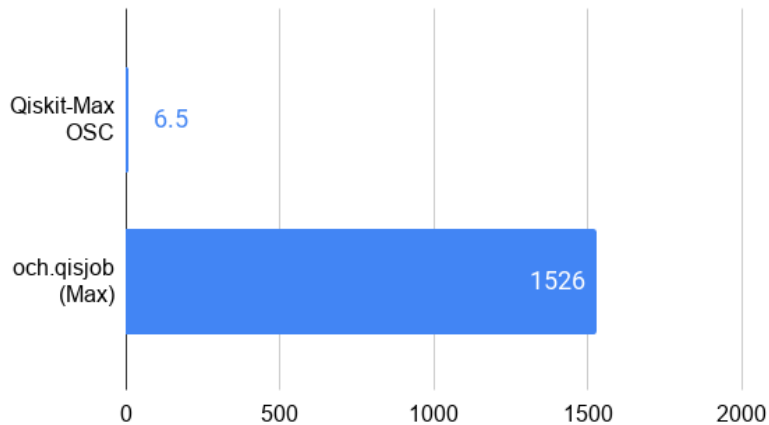


Figure 12 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 2,000 shots

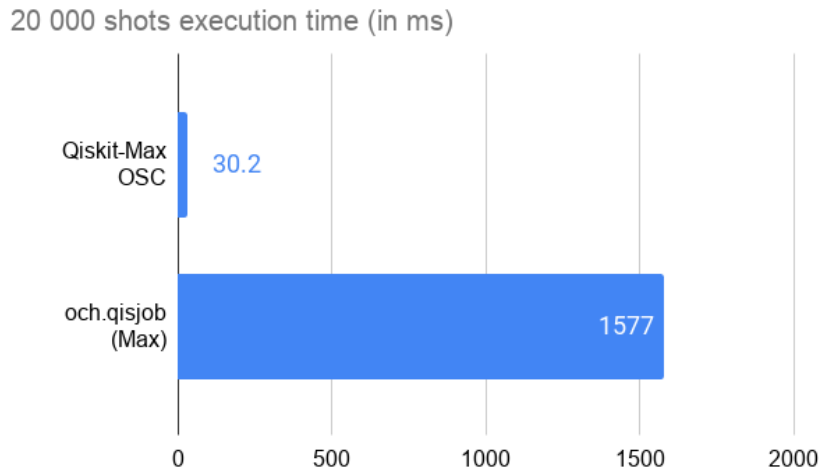


Figure 13 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 20,000 shots

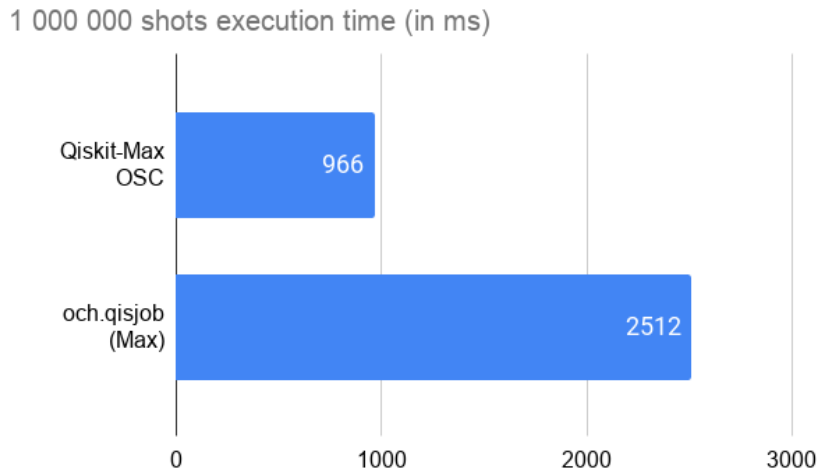


Figure 14 - Comparing Qiskit-Max OSC architecture and och.qisjob simulating 1,000,000 shots

The results here were also very surprising. The **och.qisjob** object was consistently worse than the previous strategies. But as we can see from the graphs, the difference between them tends to get smaller as the job complexity increases. Because both of these are using Qiskit under the hood, the reason for the increased execution time in **och.qisjob** is probably happening somewhere else.

And upon close inspection of these results we can also see that the average absolute execution time added between each experiment is similar for both strategies. After intensive debug sessions, I was able to pinpoint the major bottleneck being the loading process of the compiled Python code that happens every time a job is sent to **och.qisjob**. This is something that did not happen in the Qiskit-Max OSC architecture, because the Python environment needs to be already running in order to receive the OSC message.

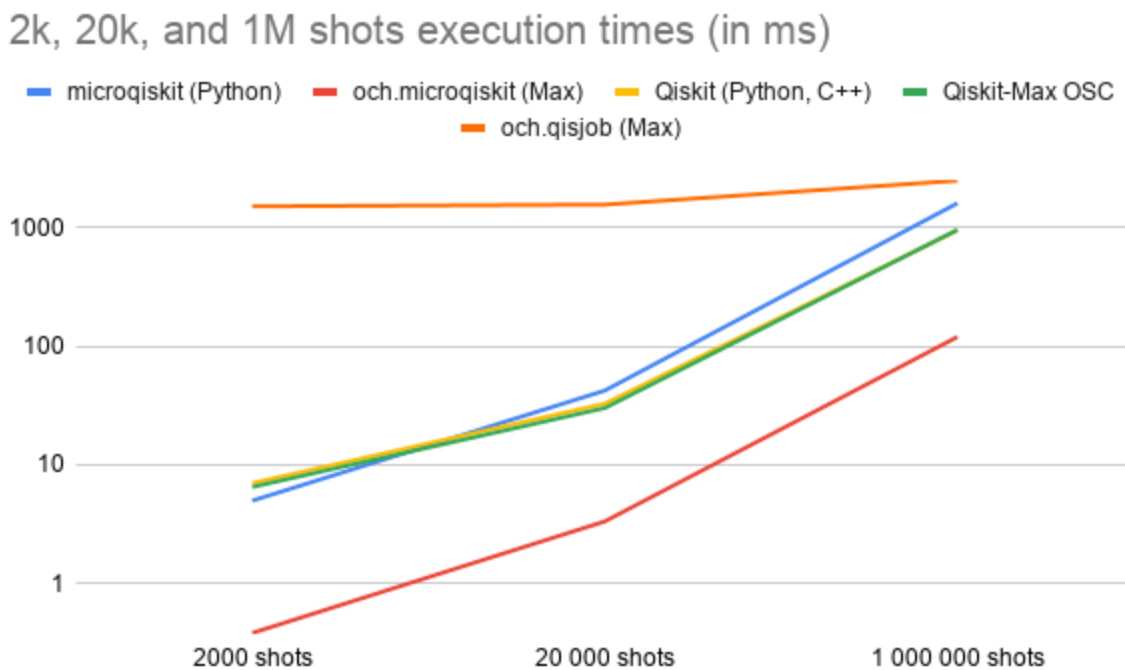


Figure 15 - All execution times for all the tools and architectures being compared

When combining the results of all the different measurements taken (see figure 15), it becomes apparent that the Qiskit-Max OSC architecture behaves very similarly to when I ran the experiment directly in the Qiskit Python environment earlier. It also becomes clear that **och.microqiskit** runs faster than anything else, and **och.qisjob** runs slower than anything else. Further work will be pursued in the future to try to reduce this execution time of **och.qisjob**.

However worse this **och.qisjob** execution time may be, when we look at it in the context of a real use scenario, of sending jobs to be executed on quantum hardware, a waiting time of a couple seconds to several minutes will be unavoidable, since the job will always have to sit in a queue before it is executed.²⁶

It is important to note that the **och.qisjob** delivers on what it was intended to become: an easy to use tool well integrated in a computer music programming environment. Thinking again about the two main groups of tools mentioned earlier in this chapter, we can now place all these different tools and strategies for connecting Computer Music with Quantum Computing in a graphic (see figure 16). If **och.microqiskit** and **och.qisjob** can live inside the “Computer Music tools” circle, the same way that Qiskit and MicroQiskit inhabit the “Quantum Computing tools” circle, the architectures and strategies being used in previous work, by myself, Miranda, and Weaver, always include setting up and dealing with something extra that is not directly related to either of these groups. The **och.qisjob** object seems to be unique in offering this direct connection without any extra steps or tools.

²⁶ There might be providers that can offer to schedule dedicated access time to a machine, but for the majority of users their jobs will have to sit in some sort of a queue for a shared machine. One can anticipate another major revolution when a “personal quantum computer” will become widely available.

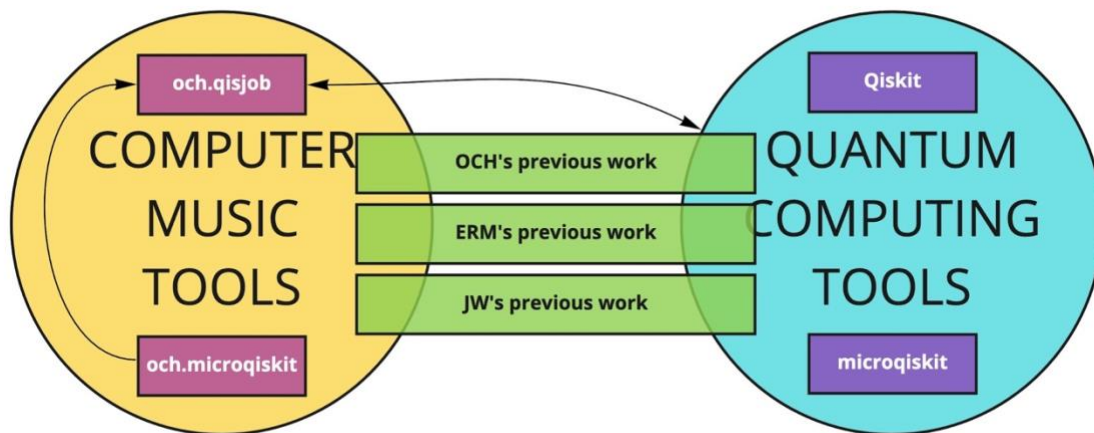


Figure 16 - Computer Music Tools and Quantum Computing Tools, and with some specific tools being compared, in context

Implications of QAC

The importance of doing this work is, first and foremost, to allow me, as an artist-researcher, to more deeply explore the integration of the emerging quantum computing technology with my compositional and creative practice. Equally important is the possibility to more easily create Max for Live devices using the objects and patches from this toolkit. The implications and potential of doing this work are varied.

On the one hand, making the toolkit available to other users and researchers will allow for other use strategies to be explored, and possibly bring new updates to the toolkit itself. On the other hand, sharing the Max for Live devices made with these tools will also allow non-programmers to use, and be early adopters of, this emerging technology. In my practice as a performer-composer-technologist, I've also found enriching experiences developing devices for non-programmers, for example in the DJing community. And that has led to recurring collaborations for building new devices, such as the one I have with Yeuda Ben-Atar (Hamido 2020b).

In the more traditional CAC world, I hope that my toolkit will bring interesting and inspiring ideas to its users, like other CAC Max packages. The Bach project (Agostini and Ghisi n.d.) is a very interesting CAC package, which may now be used in articulation with mine. The possible integration of QC tools in other CAC environments such as Open Music (Bresson et al. [2018] 2021), opus modus (“Opusmodus” n.d.), or the new PWGL (“PWGL” n.d.), will be interesting challenges, but are outside of the scope of my current line of research.

The next direction that my research in QAC is taking is expanding the current set of tools to be able to control quantum hardware at a lower level than the quantum circuit. In particular, I want to focus on exploring pulse control, using Open Pulse (Alexander et al. 2020). This Open Pulse module was created to allow researchers to characterize the noise in quantum computers by allowing them to literally design their own gates at the pulse level. For me, as an artist-researcher, this possibility opens the door to create new gates with new noise profiles. In turn, these can have great influence on creative applications of quantum computing, such as the Quantum Synthesizer, that take advantage of this noise as a way to give character to the artistic work.

In the next chapter, I will review what artists can get from QC and refer to some strategies for representing both images and sound in QC, while presenting what I am also coining as QAD - Quantum Computing Aided Design.

Chapter 3: QAD and more

In this chapter I will present examples of Quantum Computing Aided Design (QAD) and explore the meaning and possibilities of a QC-enabled Art. I will start by reviewing QC basic concepts and its different visual representations in [“Revisiting the inputs and outputs of QC.”](#) Then, I will address speed and complexity in QC, as well as the motivation to think differently, in [“Advantages of QC-enabled Art.”](#) I articulate the difference in digital representation of images and sounds, and present a very brief overview of image and sound encoding approaches in QC in [“Encoding images and sounds in QC.”](#) I will then follow up by presenting recent practical implementations of QC for generation and manipulation of images and sounds using QC, including my qc.VFX project and the new *QuantumSynth* instrument, in [“QAD.”](#) Finally, I will close this chapter reflecting on some forward-thinking explorations in this field, their meaning, and their place in [“A QC-enabled Avant-Garde.”](#)

Revisiting the inputs and outputs of QC

In order to understand how QC can change the way we interact with different artistic means of expression, it is important to understand the basics about what information we can send and retrieve from QC. As we’ve seen in the last chapter, circuit design plays an important role in the way we interact with QC. They consist of a series of logic operations to apply to qubits, to execute our calculations. For this section, we will refer to these 3 similar versions of a circuit. Figure 17 includes a 2 qubit circuit, on version *a* with only a hadamard gate applied to the first qubit, version *b* adds a cx gate between the first and second qubits, and version *c* includes 2 measurement gates that measure the state of each qubit.

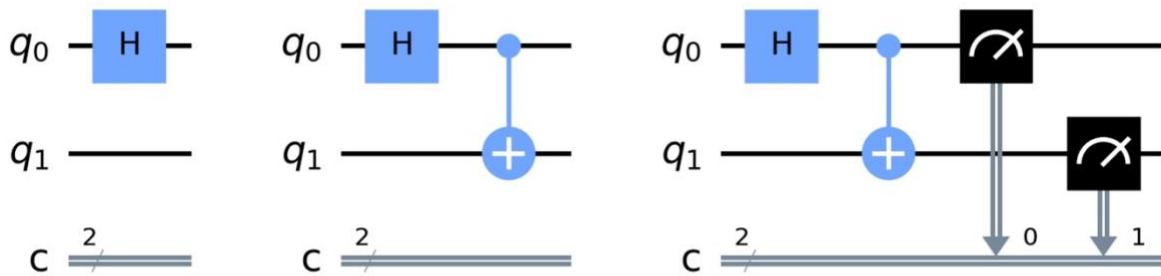


Figure 17 - a quantum circuit *a*, *b*, and *c*

One basic tenet, and a general expectation, is that QC is not to replace classical computing as a whole, but to provide an additional means for computation. As a matter of fact, there is even the proposed definition of what a quantum circuit is as a "computational routine consisting of coherent quantum operations on quantum data, such as qubits, and concurrent realtime classical computation" to compensate for the instability of quantum states (Asfaw et al. 2020a).²⁷

All the information defined in the construction of the quantum circuit is written using a classical computer. This information is converted into a series of pulses on the quantum computer that are fine tuned to change the state of the qubits. The state of the qubits can then turn into some superposition of both 1 and 0. But, when measurement takes place, they can only be read as being in the state of either 0 or 1. And this information that we receive back is, again, classical.

In sum, a quantum computer needs to work jointly with a classical computer. We send classical information, it makes use of quantum information, and we retrieve classical information. Classical numbers are what we are bound to use. The question that QC researchers around the world

²⁷ Though, defining what a quantum circuit is, hasn't come without its fair trade of multiple interpretations (Javadi 2020)

are facing now is, then: how to formulate the problems in a way that can be mapped onto a quantum computer and make use of its advantages. And this is, of course, accompanied with the quest to understand how QC works.

Following what we send to the quantum computer, QC literature and software frameworks have provided us different ways to help understand what is going on inside the quantum computer. One of them is the Bloch sphere, that represents the $|0\rangle$ state²⁸ at the top of z axis, and $|1\rangle$ state at the bottom (see figure 18). All the other points in the sphere represent a superposition of $|0\rangle$ and $|1\rangle$. The three degrees of freedom here, with the ability to rotate around the z axis, allows it to represent the state of the qubit including its phase.

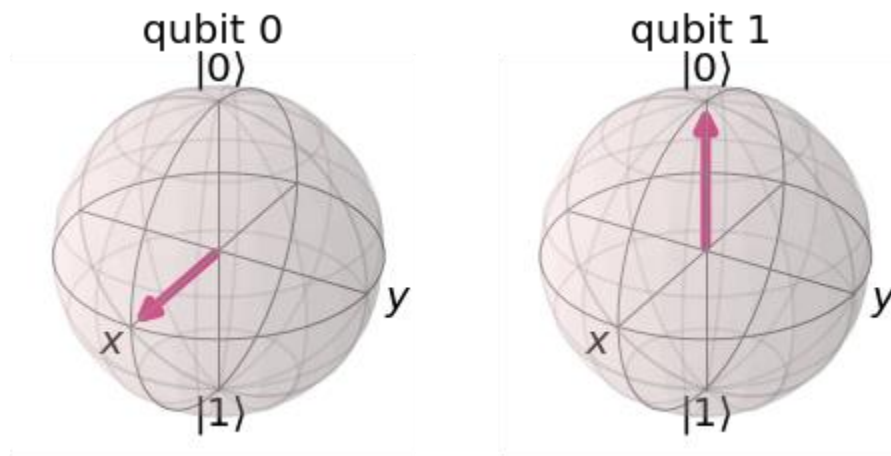


Figure 18 - Bloch spheres representing quantum circuit *a*

However helpful this representation is, it can only account for one qubit at a time, and it can in no way effectively represent qubits that are in an entangled state.²⁹ Another recent statevector

²⁸ $|x\rangle$ is the correct notation to represent a state x in QC, using Dirac's or Bra-ket notation. $|0\rangle$ is read as "zero state". Another useful way to write a state is by using matrix notation.

²⁹ Two particles in entanglement implies that the state of one cannot be independently described from the other.

visualization that can account for multiple qubits is the qsphere, invented by the Qiskit team. In it, the surface of the sphere is divided by the multiple possible states that a qubit register can take. And the phase here is represented by the use of color. This is a visualization technique that is also included in the QAC Max package as a jsui prototype called **och.qsphere** (see figure 19). It can be used in conjunction with **och.microqiskit**, if we request it to output a statevector using the *get_statevector* method.

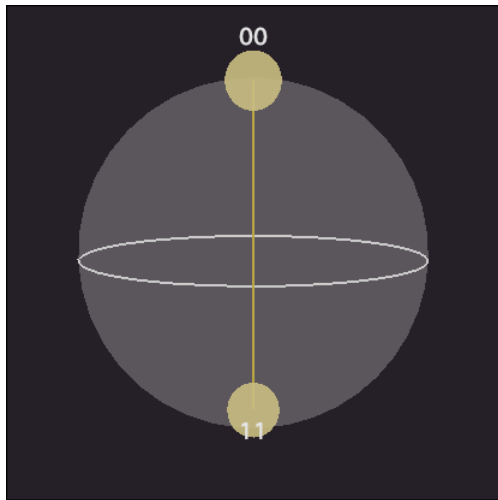


Figure 19 - och.qsphere representing quantum circuit *b*

These statevectors can be calculated mathematically, but not really measured. Instead, since the measurement process is bound to give us only 0 or 1, for each qubit, in each execution (or shot), we can only see an approximation to what the quantum state is, after collecting the results of a series of shots. For example, if a qubit is in a state of equal superposition, then its results should show that half of the time it is 0 and the other half is 1. Additionally, in the example displayed in figure 20, we can see that the effect of the entanglement via the cx gate here makes the result of the second qubit be consistently the same as the first qubit - we only get either 00 or 11.

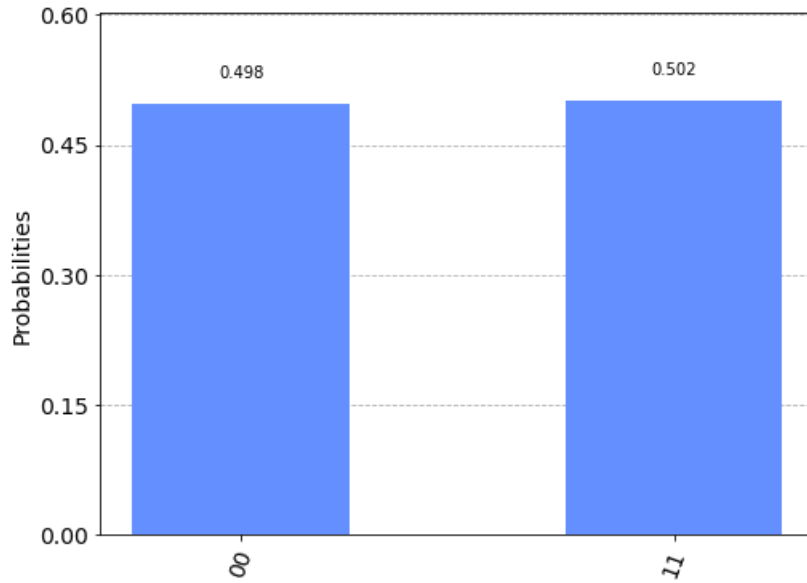


Figure 20 - histogram retrieved from running quantum circuit c

In this last example, we used quantum circuit c that includes the measurement gates, required to retrieve execution results. Even more importantly, this was run on a simulator. A simulator is a classical machine simulating the way a quantum computer works. This is not *real* quantum computation or, at the very least, is a very inefficient way to do it.³⁰ Running this example on a current superconducting real QPU retrieves the results shown in figure 21.

³⁰ Simply put, a classical machine needs extra resources to simulate the behavior of a quantum computer.

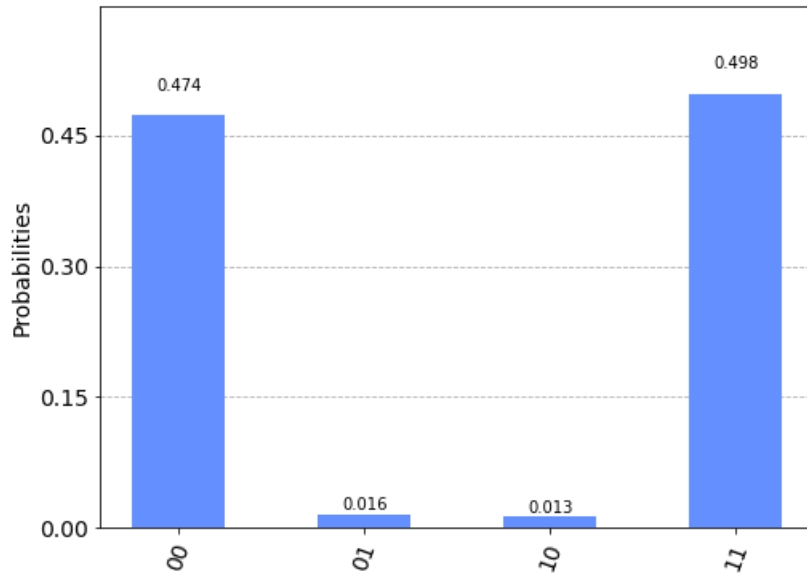


Figure 21 - histogram from running quantum circuit c on ibmq_athens real QPU device

Even though we can mostly see the same output here, there are also some unwanted results. The state $|01\rangle$ and $|10\rangle$ should not be possible to obtain from the quantum circuit submitted. But these results still happen due to noise in the machine. Noise here can be simply understood as the interaction of the qubits with the environment, which may cause them to decohere by dephasing, or have energy relaxation. At the time of this writing, this is unavoidable. And the study of noise in QC is a big part of the research being done in the field. Towards that end, some QC simulation in classical machines can use noise models to have its results resemble more closely the output of real QPUs.

Between inputs and outputs, and what the state of a quantum computer is - which again, includes quantum circuits, statevectors, simulators, simulators with noise, and real machines - creative artists can help explore the possibilities and build an understanding of this emerging technology.

Advantages of QC-enabled Art

Having revisited the inputs and outputs of QC, I would like to address the advantages that QC might bring to creative practices, in this current stage of Noisy Intermedia Scale Quantum machines, or NISQ for short (Preskill 2018).

Computers offer us yet another way to explore compositional ideas or build instruments for creative expression. But, as different as they may be from other instruments, they can be more efficient or even the only way to accomplish certain tasks. Sever Tipei observed the following about the use of classical computers in art:

[The computer's power] also provides [artists and scientists] with the means of scrutinizing their own way of thinking. Flaws in the concept behind the work or in the way the rules of the game are devised and applied will be revealed quickly, because the computer's choices depend only on the instructions contained in the code, not on implicit criteria inferred from alien considerations. [...]

In this context, the use of randomness in the arts provides a realistic way of experimenting that includes the possibility of accidents, of deviations from an abstract norm. Another computer skill that people can hardly match is the ability to mass produce variants of the same work or, in other words, to realize a large number of pieces belonging to the same class of compositions. (Tipei 1989)

Computational power and complexity

A computer as a machine to execute coherent logical operations is an unmatched instrument, to the point that it offers itself as a good collaborator to scrutinize one's own set of logical rules. One of its strengths is in its ability to store and retrieve information, with consistent faithful reproduction: a *groundhog-day* machine.³¹ On the other hand, a computer's ability to deliver random numbers much better than any human could do, makes it also the perfect unbiased collaborator: an *eternal-sunshine* machine.³²

³¹ An early 90's movie where the main character is living the same day over and over again (Ramis 1993).

³² A movie from early 2000's where the main characters erase the memory of each other and get to meet again for the first time (Gondry 2004).

QC may offer advantages in both cases. There are algorithms that can search through an unsorted database exponentially faster, like Grover's (1996). And, unlike its classical counterpart, it can offer true randomness, based on the physics of QM, as opposed to pseudo randomness, based on complex, but deterministic, algorithms. These two cases highlight two characteristics that I believe to be the reason for artists and researchers to want to devote time to learn and integrate QC: speed and complexity.

One can understand how the difference in speed can come into effect just by simply calculating the number of classical bits required to simulate n qubits in equal superposition: 2^n . The difference is exponential. Simulating a quantum computer with 300 qubits would require more classical bits than the number of atoms in the universe (Cavill 2015). That's the number of calculations that a quantum computer would be able to perform in one go, thanks to the capacity to perform calculations with multiple states at the same time.

In turn, complexity underlies the way a quantum computer works, and is also what allows it to produce true random numbers. To this point, the knowledge about a system's inner workings plays an important role, something that QC is protected from because of the laws of QM. On this topic, one of my favorite examples from Chris Dobrian's *Max Cookbook* that can serve as a good illustration is the one about sequences of pseudo-random numbers appearing random, where 2 number sequences are shown as the output of 2 processes whose inner workings are hidden, and the user must discover which one uses randomness (Dobrian n.d.). Of course, numbers can be turned into sequences or strings of numbers, but they can also be turned into music or painting. But even with music and painting, only the composer, who chooses to use true or pseudo randomness, might be able to tell the difference. For an audience member, or someone who was not in charge of the choice of method, knowing the difference might be impossible. As Christine Corbett Moran articulated:

[...] with the two numerical series of numbers a person might not be able to differentiate but mathematically you could tell the difference. So, it may be that actually to tell the difference between a quantum composed piece of music and a classically composed piece of music you might need an actual computer to do that. (Moran, Christine Corbett, conversation with the author, June 2, 2020)

Thinking differently

The current NISQ era isn't capable of delivering a universal quantum computer with 100+ fault tolerant qubits. The implications of this is that it makes it all the more difficult to prove the computational advantage of QC over classical computers. Given this, there is currently no creative artistic expression using QC that can be proven to not be achievable by any other means.³³

The difference then, of creating QC-enabled Art, relies on the creative process. To think differently is the first and main purpose of the tools that I've developed. In so far as QC requires one to approach computation from a new perspective, the creative artist using QC may also be required to change the thinking and compositional process. The expectation is, of course, that by thinking differently one will be able to end up in different and new places. Perhaps, this will lead to, or reinforce, a change of paradigm that Lukas Ligeti observed:

[...] I have noticed that several students represent an intriguing paradigm shift in the way they conceive and notate musical ideas. [...] They think of the idea at its most basic level as an algorithm, not as a set of instructions for a musician to perform the music, but as a set of instructions for a computer to generate it on-demand. (Ligeti 2021)

In my own practice I can recognize how the use of the computer plays an important role in the way I approach both research, writing, performance and production of my artistic work - even if only by admitting the amount of time that I spend with it. And the advent of these new QC tools that I've developed have also already informed both compositional strategies, instrument design, and

³³ Due to the current nature of this field, where there's more research being done in 1 year than in the previous 10 years, I need to highlight the fact that my statement is based on the current state of the art that I've been able to research and find, at the time of this writing.

visual production of my recent creative work. Like Tipei mentioned for classical computers, these tools may also help to achieve new variations, and simply by exploiting the noise in the real machines.

Encoding images and sounds in QC

Similar to QAC, the proposed QAD acronym stems from its classical counterpart CAD (Computer Aided Design), and refers to a set of tools and techniques for Quantum Computing Aided Design. Before I address what efforts might fit under this umbrella, and how it has been used in my work, I'd like to briefly recognize the main difference of working with digital images and sounds, from the computational cost perspective.

On the one hand, dealing with digital images is less constrained by time than when dealing with audio. A still image or a video has less images per second than the number of samples per second required to have a faithful and perceptible sound reproduction. On the other hand, the amount of information to be processed in video is usually far superior. Think of the following example: if we are using audio with a sample rate of 48,000 and a buffer size of 128, it means that we are processing 48,000 values per second in chunks of 128 values, every 2.7 milliseconds. In contrast, if we are using video at 60 frames per second and a resolution of 1280x720, and using 1 channel instead of the usual 4 (red, green, blue, and alpha), for the sake of comparison with a single channel audio, we are processing 55,296,000 values per second in chunks of 921,600 values, every 16.7 milliseconds.

One can easily see that even processing just one of these chunks, as in retrieving just a single image instead of a moving picture, is still far above the amount of information required for a couple seconds worth of audio. And if we take into account the range of numbers or values used to describe each quantum of digital information, which can usually be 256 values (8 bit) for a pixel per channel in image, or 65536 values (16 bit) for a sample per channel in audio, we are looking at a proportional relation of 9 to 2 - as in, video takes 4.5 times more information than audio. This alone is the reason

why we have GPUs (graphical processing units) for computers - a processing unit that is optimized for throughput rather than speed.

Alain Lioret, mentioned in Chapter 1, pointed at “Andrew Glassner's Other Notebook” as the first reference to using QC in computer graphics. In this book, Glassner gives an introduction to QC and mentions the possible advantages for z-buffer, radiosity, and data transmission (Glassner 2002). Another reference Lioret mentions is Lanzagorta and Uhlmann’s Siggraph paper and presentation, where they first mentioned quantum computational geometry, to solve computational geometry problems like collision detection, Convex Hull Determination, and rendering processes like Z-Buffering, ray casting, and radiosity (Lanzagorta and Uhlmann 2005a; 2005b).

A last reference that Lioret provides is the work of Caraiman and Manta, where they propose a new quantum algorithm for the RANdom SAmple Consensus (RANSAC) voting scheme, which as “an algorithm for robust fitting of models in the presence of many data outliers [that] has proved to be a very important result for both computer graphics and vision” (Caraiman and Manta 2009). It is interesting to note that Caraiman and Manta were also probably one of the early adopters of Ömer’s QCL programming language, mentioned also in Chapter 1 (Caraiman and Manta 2008).

Also interesting to note is that, contrary to what Kandinsky was facing in his own time, i.e. noting that music had advanced much more than art (Kandinsky 1946, 35), now it is the visual domain that is several years ahead of the sonic domain. At this point, there have been numerous attempts to encode images in QC. These include, as listed by Jian Wang (2016):

qubit lattice (Salvador E. Venegas-Andraca and Bose 2003), real ket (Latorre 2005), entangled image (S. E. Venegas-Andraca and Ball 2010), FRQI (Le, Dong, and Hirota 2011), MCQI (Sun et al. 2013), NEQR (Zhang, Lu, Gao, and Wang 2013), QUALPI (Zhang, Lu, Gao, and Xu 2013), QSMC & QSNC (H.-S. Li et al. 2013), NAQSS (H.-S. Li et al. 2014), GQIR (N. Jiang, Wang, and Mu 2015)

A more organized survey of quantum image representations can be seen in (Yan, Ilyasu, and Venegas-Andraca 2016). But one interesting thing to note is that, according to Abdullah M. Ilyasu (2019), the start of Quantum Image processing can be traced back to Alexander Vlasov's 1997 paper (Vlasov 1997).

In terms of encoding sound in QC, there are 3 main efforts. A first one, called QRDA (Quantum Representation of Digital Audio) that stores sample value and sample position in qubit sequences (Wang 2016). The second, FRQA (flexible representation of quantum audio), is an expansion of the former, making it more suitable for (quantum) audio signal processing, by enabling the representation of a bipolar signal, as opposed to unipolar (Yan et al. 2017). Finally, QRDS (quantum representation of digital signals) further expands FRQA by trying to overcome its limitation of using only integers and not processing decimals, and proposing circuits for simple arithmetic operations like signal addition, subtraction, multiplication, division, and circular convolution (P. Li et al. 2018).

It is important to note that these different encoding techniques only serve as theoretical models, with no current applications with NISQ era QPUs, since they don't take yet into account the effects of noise/decoherence. For that reason, despite these many attempts to encode images and sound in QC, there is still room for new methods. Perhaps a different approach to representing signals that embraces noise will first be motivated by artistic concerns, similar to the work that is still being done, for example, by Bovermann and Sanfilippo with their one-bit stream classical representation of audio (Bovermann and Sanfilippo 2020).

QAD

Some recent works that explore the generation and manipulation of images using QC, and accounting for the impact of noise in the real machines, can be seen as the first manifestations of what QAD can encompass. One of these works is the interesting discovery by the researcher and software

engineer Eric Johnston who developed a qubit-based image supersampling technique (Johnston, Harrigan, and Gimeno-Segovia 2019, chap. 11). This technique not only shows advantages when compared to the conventional Monte Carlo approach, with a lower mean pixel error percentage and a higher error-free pixel count, but it also delivers a resulting image whose noise (garbage pixels) in some ways might even be easier to identify (see figure 22). More importantly, I'd argue that such noisy results carry an identifiable characteristic, or flavor, that is interesting for the creative artist to explore.

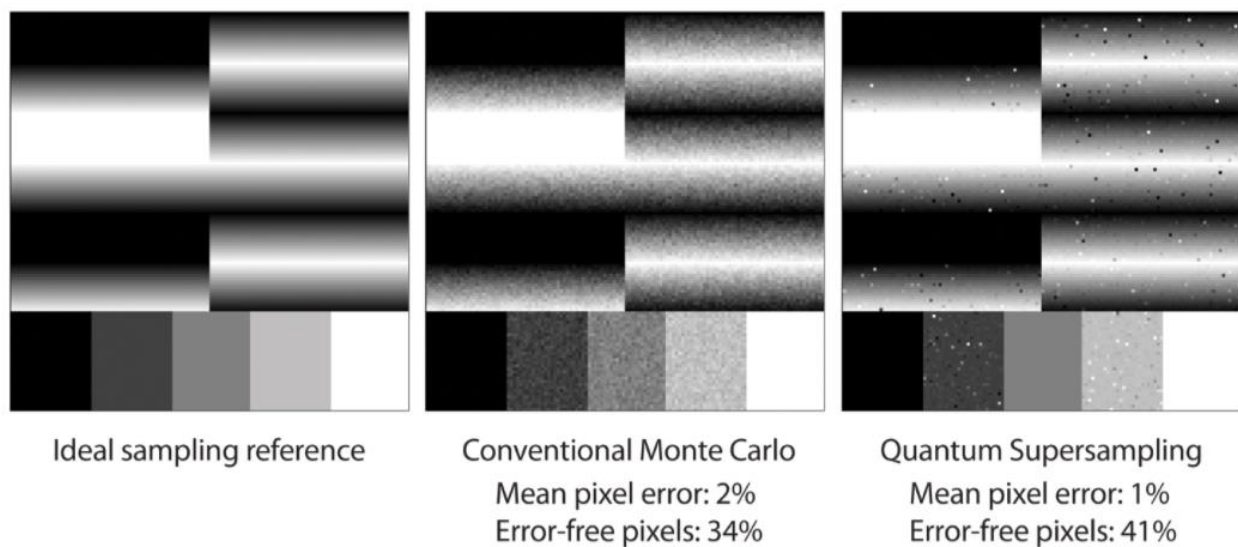


Figure 22 - comparison of QSS with Monte Carlo. Reproduced by permission from Eric R. Johnston (2019, fig. 11.1)

Another relevant recent work is James Wootton's "QuantumBlur". This is a tool that converts an image into a group of quantum circuits, applies a partial x rotation to every qubit on each circuit, and converts the results of running these circuits back into an image (J. Wootton [2020] 2021). Changing the value for the qubit rotation fraction allows for control of the amount of effect that is applied to the image. This is a value ranging from 0 to 2π , with maximum effect at π , after which point the effect seems to reverse. Saving multiple executions of this process with a progressively increasing

fractional rotation value can be equivalent to collecting frames of a video. In fact, the title screen for *The Gedanken Room* film-recital, discussed in the next chapter, was created using this technique.

Like Johnston’s technique, the output of Wootton’s “QuantumBlur” carries an identifiable characteristic that is very inviting for exploration by creative artists. The tools they use for carrying these techniques, however, are based on the regular QC tools that scientists and programmers use. Similar to the process described in the previous chapter, I attempted to create an integration of these innovative techniques with a realtime visual programming environment aimed at creative artists.

qc.VFX

In October 2020, during the Qiskit Hackathon Global, I started a project with the codename *qc.VFX* that integrated Qiskit and Microqiskit in TouchDesigner (Hamido, Woehr, and Chen [2020] 2020; “Derivative” n.d.). The goal, similar to the QAC Max Package, was to provide an integrated solution for working with QC in a realtime video art making environment. TouchDesigner (TD) is a visual programming language mainly dedicated to work with realtime video. Unlike Max, however, TD supports running python scripts using its built-in console or by executing the contents of a DAT operator.³⁴

The *qc.VFX* project was focused on implementing the “QuantumBlur” technique in TD as a tox component.³⁵ The team was composed of me, Chris (Jielun) Chen, and Jack Woehr. We were able to have James Wootton as the mentor for this hackathon project, which was helpful to understand some

³⁴ TD has six main types of operators. DATs (Data operators) for processing text, TOPs (Texture operators) for processing images, SOPs (Surface operators) for processing geometry, MATs (Material operators) for processing materials, CHOPs (channel operators) for processing signals, and COMPs (components) that can contain their own network.

³⁵ A tox component is a TD COMP that can be saved as its own file (.tox file), independent from a TD specific project (.toe file). This makes it easier to share just that individual part of the program.

of the inner workings for the image to circuit conversion part. Several texture adjustments had to be made in TD and part of the process had to be recreated in the tox component network. We ended up relying on a Script TOP to help convert the textures to a numpy array (see figure 23).³⁶

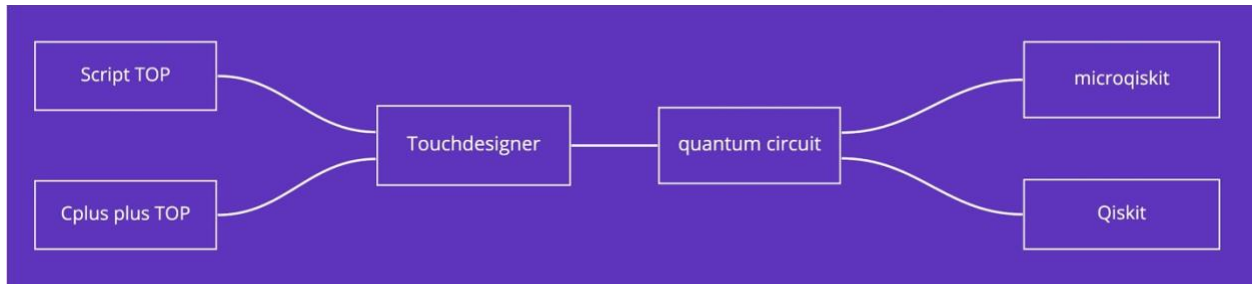


Figure 23 - framework for the implementation of qc.VFX

Soon enough we realized the main challenge we faced with this project: TD, as a realtime video environment, runs at 60 frames per second, and the “QuantumBlur” code cannot generate new frames anywhere near that speed. We had to spend considerable time exploring threading, in order to avoid TD freezing or simply crashing while running this process.

Because TD needs something to cook each frame,³⁷ we opted for not driving the main animation with the resulting textures from the QC process, and use these instead as a displacement map. For the purpose of the proof of concept demo, we applied these textures to a rotating plane, and interpolated between non-affected texture and the processed texture with an lfo. This worked very well demonstrating the result of the “QuantumBlur” effect as a change in geometry (see figure 24).

³⁶ See [Appendix A](#) and [B](#) for a screenshot of the TD project and “quantumblur” component.

³⁷ Cooking in TD refers to the process of re-computing the output of the operators. Each new frame triggers the cooking of all the operators required to render it.

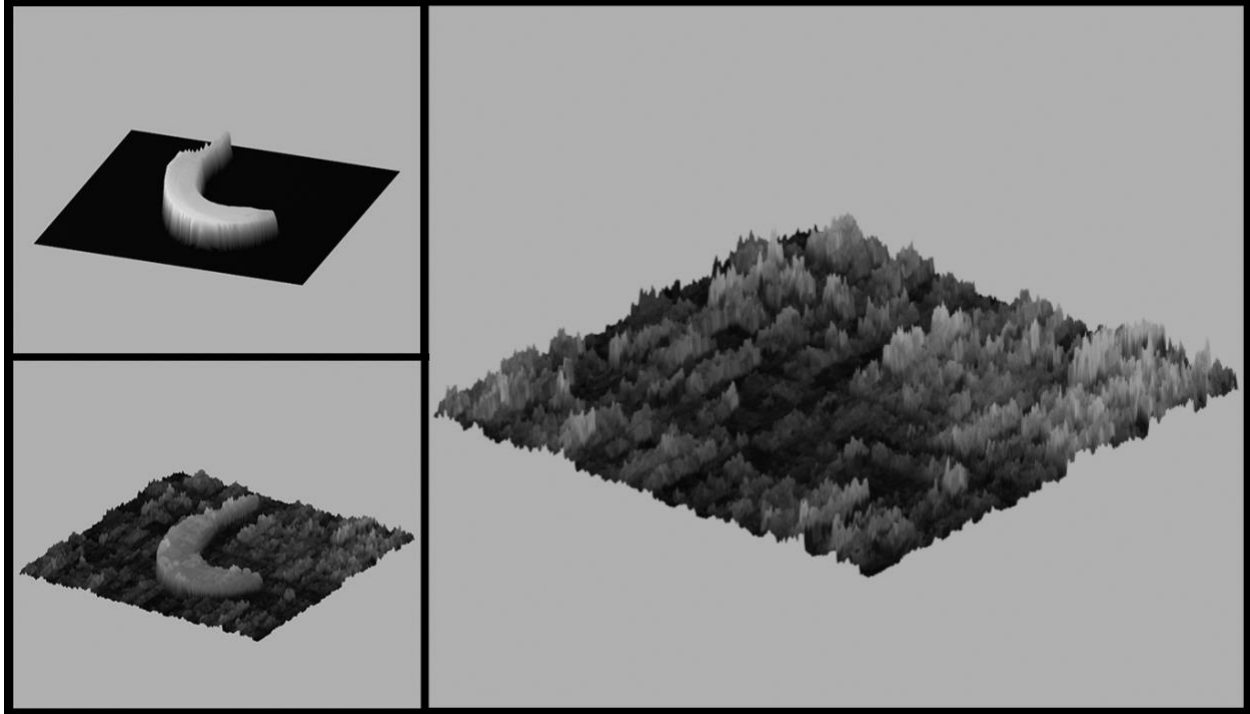


Figure 24 - frames of the interactive interpolation of QuantumBlur in qc.VFX

QuantumSynth reforged

While qc.VFX as a project that explores QC-enabled Visual FX is a good example of QAD, some QC-enabled sound design tools and visual interfaces may also be relevant for consideration here. The new version of *QuantumSynth*, used in the *Decoherence* piece discussed in the next chapter, makes use of QC to draw wavetables and define its controls.

This version of *QuantumSynth* is a wavetable synthesizer running pre-built circuits on a statevector or simulator backend using up to 8 qubits and 4096 shots. Each circuit creates a different wavetable that is drawn in the visual interface of this device. The main controls that modify the selected wavetable are the horizontal and vertical sliders, next to the wavetable display, that change the number of qubits and number of shots to use, respectively.

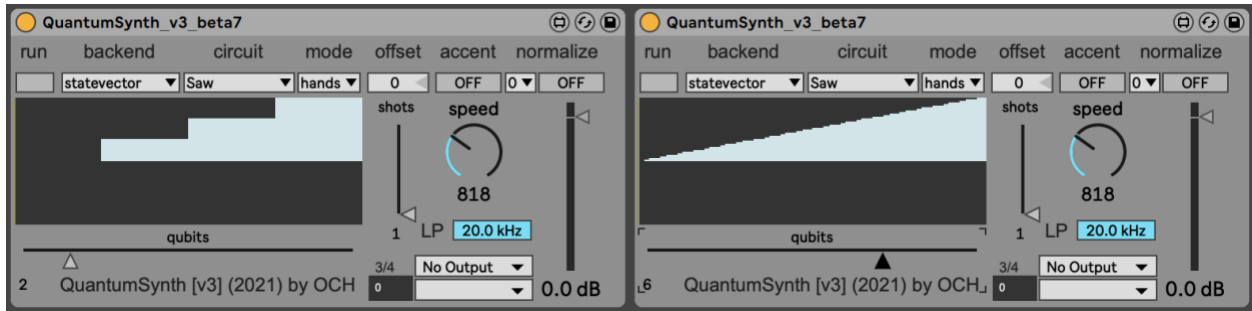


Figure 25 - *QuantumSynth* v3 saw circuit using statevector backend with 2 and 6 qubits

The wavetables are drawn based either on the calculated state probabilities, when using the statevector backend, or the counts from executing the circuit - similar to the histogram mentioned earlier in this chapter. The qubit count control was set as an horizontal slider since it controls the horizontal resolution of the depicted wavetable. The number of columns is equal to the number of possible states, that is $2^{(\text{number of qubits})}$. In figure 25 we can see the difference of representing a sawtooth wavetable using 2 and 6 qubits. Since this is using a statevector backend, the number of shots doesn't produce any effect.

Figure 26, on the other hand, shows the effect of different numbers of shots to represent this wavetable when using the simulator backend. In this case, a very low number of shots may only return results containing several single instances of different states. However, when using a higher number of shots, the shape of the wavetable becomes more clear, as it aggregates more counts that, collectively, are able to more closely represent the underlying statevector probability distribution. The simulator results are always normalized, so the number of shots controls the vertical resolution.

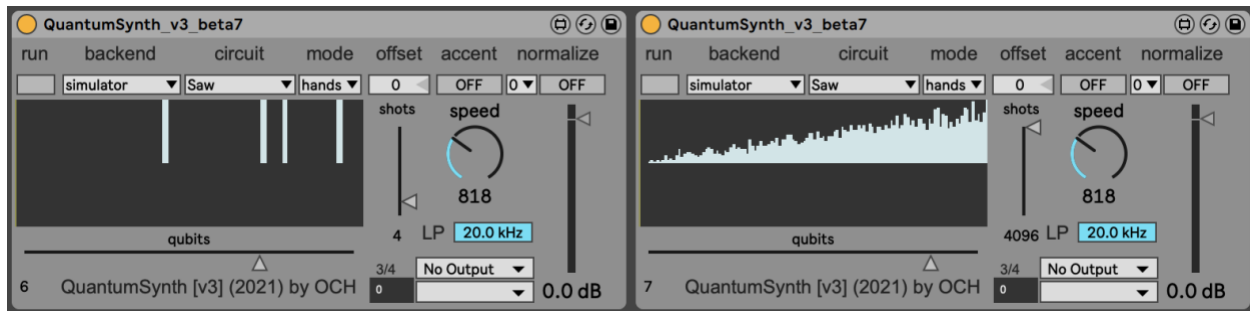


Figure 26 - *QuantumSynth* v3 saw circuit using simulator backend with 4 and 4096 shots

The control parameters and the visual feedback of the *QuantumSynth* as well as the inputs and controls used in qc.VFX can be seen as macroscopic formal controls, as defined by Chris Dobrian on his paper about stochastic realtime music (Dobrian 2012). More importantly, I believe these projects can demonstrate the advantages of QAC and QAD, in the way they make use of complexity in QC and are able to exploit noise to create variation, as previously mentioned in "[Advantages of QC-enabled Art.](#)" As these projects enabled me to design both images and sounds that I wouldn't think of, or wouldn't know how to create otherwise, I believe these may also stimulate creative artists into thinking differently and finding new creative approaches.

A QC-enabled Avant-Garde

In this new field of Quantum Computing Aided Composition and Design, and within the given constraints of the NISQ era, it is hard to define what exactly is, or will be, the QC-enabled artistic avant-garde. Beyond the current practical explorations and reflections presented in these chapters, there have been a couple attempts to theorize music in the age of quantum advantage. One of them is a paper I wrote with my colleague Scott Oshiro, in which we proposed a quantum-classical network framework for a telematic music scenario, with beat-making performers and audience members connected in this hybrid network (Oshiro and Hamido 2020).

This paper was aimed at a community of researchers in network music arts and included the description of some of the inner workings of this framework, as well as a preview of the prototype we were working on. Needless to say, there is currently no real quantum internet that we can use to test this, and simulated quantum networks are inefficient. Though our proposal might bring about an important change to this field, so far its implementation is impractical.

Another work very much focused on proving quantum advantage is Alexis Kirke's paper on "MIq," a multi-agent interactive system that he claims to be "the first attempt at a realtime interactive quantum computer music algorithm that utilizes the quantum advantage," that "provides a quadratic speed-up over classical methods" (Kirke 2020). It is a very complex system, for complex machines, but in the end he recognizes the limitations to putting his framework to practical use on real QPUs. Furthermore, the musical examples he presented are very simple, a fact which makes it difficult to say with confidence that the work in its current form achieved his stated goal of "using the underlying hardware in a uniquely useful way, or to do something new" (Kirke 2020).

As important as these attempts may be, their considerable impracticality raises the question of whether these tools, in their current form, can inspire artists to create new work. We may need to first take smaller steps, like those presented in the previous chapter with the toolkit integration and speed optimization found with **och.microqiskit**³⁸ and **och.qisjob**. The tension persists, however, between the theoretical and the experimental, and between forward-thinking frameworks and practical applications. My own creative work, of which I give an account in the next chapter, reflects the challenges of navigating these same tensions. I have to constantly ask myself how this work is helping me to find new creative directions, and remember the interdisciplinary connections that help

³⁸ On its own, it may not be able to run real quantum computation (only possible when paired with **och.qisjob**), but the increased speed might be just good enough to make it already useful for practical realtime creative work, and help creative artists in their explorations.

me understand the universe in a new light, a position articulated eloquently by the pioneering composer Pauline Oliveros:

So what is a new idea of Avant Garde?

Perhaps creating and being at the center of a community of practitioners?

Perhaps living in solitude, creating but never sharing any art created?

Perhaps swimming and exchanging in vast cyber oceans with numerous different schools of fish?

I feel myself existing as an atom of world culture. I bond and break apart with other atoms, to form molecules, entities and ever changing networks. (Oliveros 2019)

Chapter 4: The Gedanken Room

[...] every musical composition justifiably may be regarded as an experiment [...] in that Gedankenexperiment which is the mental act of composition [...] (Babbitt 1962, 49)

In a sense, artists are always constructing imaginary worlds, “parallel” or “aesthetic” realities, testing ideas and their implications through the creative process. (Tipei 1989)

In this chapter I address the musical and creative work that accompanies this document. The theme of the dissertation film-recital is explained in “[The Gedanken Room](#).”³⁹ The sections that follow will explain each of the four original works that are part of it — “[4 Disklavier Preludes](#),” “[Telematic Entanglements](#),” “[Second Cornerstone](#),” and “[Decoherence](#)” — as well as a section on “[Interludes and the larger narrative](#).” To end this chapter, the last section, “[In retrospect](#),” gives an account of the possibilities and limitations faced when pursuing the recording and post-production of this work.

The Gedanken Room

All the research and software development, presented in the previous chapters, has greatly contributed to, and informed, my own recent creative practice. Even though the mindset can be a little different for reading papers, solving programming challenges, and writing/performing new music, I do regard them all as being part of the same process. They don’t always end up in published papers, released software, or finished compositions, but they are, collectively, what I would like to call my *Adventures in Quantumland*.

³⁹ *The Gedanken Room* (2021), available at quantumland.art and public streaming platforms.

In turn, the dissertation film-recital, that accompanies the culmination of the research and creative work presented here, was composed of four different pieces and titled *The Gedanken Room*. The origin of this title lies in the concept *gedankenexperiment* - from German, *gedanke* meaning *thought*, and *experiment*. The use of thought experiments was central to the origins of the field of quantum mechanics.⁴⁰ Here I am creating a metaphor for each creative work as a thought experiment taking place in this black box theater. And at the same time, I also drew inspiration from this passage in Robert Gilmore's allegory of Quantum Physics, when Alice is led by the Classical Mechanic to a room where he can explain some quantum phenomena:

This is the gedanken room, which means a "thinking room." [...] In here one's thoughts can take on substance, so that anyone can look at them. It allows us to do thought experiments. These allow us to work out what would happen in various physical cases, and they are much cheaper than real experiments of course. (Gilmore 1995, 33)

Because of reasons that I will review later in this chapter, each piece had to be recorded individually, and this led to a non-traditional music recital, where it is usual to have a program of pieces that are played back-to-back. Instead, I had the opportunity to explore transforming the black box space for each piece individually, and this enhanced the idea of depicting different thought experiments.

This recital includes 4 main pieces: *4 Disklavier Preludes*, *Telematic Entanglements*, *Second Cornerstone*, and *Decoherence*. Between each piece there is an interlude, and the *4 Disklavier Preludes* are not presented together, but instead each Prelude is alternated with the rest of the repertoire. This decision was made with a whole narrative in mind. So much so, that the production of this recital became more of a film production, and it was released as such (Hamido 2021b).⁴¹

⁴⁰ For a taxonomy and history of thought experiments see (Brown and Fehige 2019; "Thought Experiment" 2021).

⁴¹ See storyboard in [Appendix C](#).

The whole recital is, like the dissertation overall, an exploration of the implications of Quantum Computing for Music Composition and Improvisation. Each piece explores different ways to use QC in my creative practice. In the following sections, I'll give an account of these different ways and the narratives that were constructed.

4 Disklavier Preludes

The *4 Disklavier Preludes* are 4 short pieces, with different instrumentation, but all having in common using the Disklavier instrument (Acker 2016), which is being controlled by the computer. In each Prelude the computer is sitting on the music rack, running different combinations of Max for Live devices, some of them making use of the tools presented in the previous chapters. Here, the control and the role of QC is in the generation and transformation of new MIDI note events, or clips, that are sent to the Disklavier, via MIDI interface, to activate the keys and pedals of the instrument.

An overall source of inspiration, on all four of the Preludes, for working with the Disklavier and, more specifically, writing for piano, is the Russian performer-composer Alexander Scriabin (1872-1915). This is a composer whose work I've been enjoying for over a decade, and whose musical expressivity and brightness has always puzzled me. Recently, while reading more about him, I found an interesting connection with the creative work that I am currently pursuing, as well as the conceptual discussion presented in [Chapter 1](#):

Scriabin arrived at his harmonic revolution by many means, and via diverse paths. He worked simultaneously in a number of concrete and intangible areas. He had invented inside himself an astral world. Now, how to actuate it on the outside? His friend the electrical engineer Alexander Mozer, introduced Scriabin to Einstein's theories, physics, and quantum mechanics - sciences which Scriabin felt he needed to complete his intellectual grasp of the world. (Bowers 1973, 135)

Disklavier Prelude #1



Figure 27 - Still frame of *Disklavier Prelude #1*. From (Hamido 2021b)

The first Prelude is set as a study session, and it has 3 sections. In the first section, the performer-improviser is progressively exploring a harmonic space that is constructed with the help of a program that responds to notes played, through the Disklavier. In the second section, the performer improvises with short and fast notes testing the speed of reaction of the Disklavier. And in the third section, a simulated quantum circuit decides which notes to play on the Disklavier while the performer creates a minimalist accompaniment on the fly.

The very first four notes played, C4 Bb4 F#4 and D5, denote the essence of the most iconic harmonic trait of Alexander Scriabin's music: the mystic chord. These four notes allow one to construct the variants of the mystic chord since, in the derived dominant articulation, C4 and F#4, as

tonic and #4, trade places, while Bb4 as the b7 will give us the major third, and D5 as the 9 will give us the #5, which is part of the whole-tone-mystic variant (Goldwire 1984, 30), as illustrated in figure 28. In effect, this will also be the genesis of all the harmonic content for all the 4 *Disklavier Preludes*.



Figure 28 - initial notes and harmony in *Disklavier Prelude #1*

The initial exploration section after these four notes uses a program that makes the Disklavier play the inverse of the notes played by the performer, with the axis on the first note of each group. In practice, the first group of notes spell a C9 chord, ending with a 9 two octaves above (16th). This results in a Bb9 chord on the inverted side that, together with the top half spells a Bb13#11 - a super spread and self-standing dominant chord, with the notes from the regular mystic chord (see figure 29).

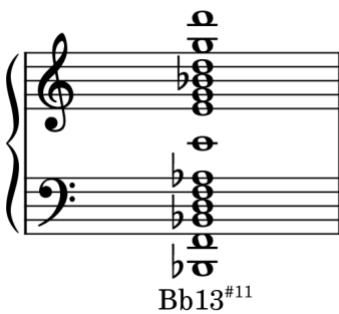


Figure 29 - Super spread self-standing dominant chord in *Disklavier Prelude #1*

In *Disklavier Prelude #1*, I am using a new Max for Live device, **och.disk**, very similar to the old **och.qc-circ-2q**, but using the new **och.microqiskit** external under the hood. It still reads the name of the playing MIDI clip as a minified circuit notation, but in this case, the clip names must start with an integer that sets the number of qubits to use. This variable qubit count circuit creation on the fly allows the device to represent a larger range of values that are useful for the generation of MIDI notes - while 2 qubits had a very limited range, 7 qubits can already express all MIDI pitch values (0-127). And instead of using dots "." to connect the symbols of a gate, the notation can now use spaces, similar to how the user writes the messages in Max, but only needing to add several gates together using the plus symbol.⁴²

In the last section of this first Prelude, by combining different clip lengths and different circuits using some superposition with **och.disk** I was able to have the Disklavier play some seemingly random faster notes followed by a greater probability of returning to the initial note.

⁴² E.g. "7+h 0+h 1+h 2+h 3+h 4+h 5+h 6" means create a circuit with 7 qubits and add an h gate to each qubit.

Disklavier Prelude #2



Figure 30 - Still frame of *Disklavier Prelude #2*. From (Hamido 2021b)

Prelude #2 starts completely in the dark, only hearing the voice of the Google Voice Kit⁴³ reading a passage about Heisenberg's episode on the night of June 7, 1925, that led to his insight about the way electrons behave in the atom. Similarly, one light, and then another, are turned on in sync with the text and some Disklavier melodic movements. This Prelude, in particular, is the only piece in the program that is completely fixed, with no space for improvisation, and it is also the shortest one, as well as the only one without a human performer on stage. It so happens that Scriabin

⁴³ See more in the discussion in [Interludes and the larger narrative](#).

himself was actually recorded by some of the early player-piano machines (the predecessor of the modern Disklavier): the Hupfeld Phonola and the Welte-Mignon (Leikin 2011).



Figure 31 - the harmonic progression in *Disklavier Prelude #2*

As the Prelude continues, the Disklavier is playing first an harmonic accompaniment to the reading, that later will take a lead role (see figure 31). The harmonic movement progressively comes to a point where it plays 4 bars of a MIDI transcription of Scriabin's *Waltz in A-flat major Op. 38* from 1904, making a clear direct connection with this reference (see figure 32). At the end of this excerpt, it becomes clear that the initial progression is slowly evolving, via parallel and contrary motion, from the same exact chord that ends this movement. The artistic intention in this piece was both to illustrate Heisenberg's episode, and depict the thought process and the rush of energy that is achieved upon gaining a new insight.



Figure 32 - excerpt from Scriabin's "Waltz in A-flat major Op. 38" from 1904

Disklavier Prelude #3



Figure 33 - Still frame of *Disklavier Prelude #3*. From (Hamido 2021b)

The narrative for *Disklavier Prelude #3* is distinct from the other Preludes, as it includes the participation of an actor and a dancer; the last role being taken by Waeli Wang. In it, we can see the dancer sitting still on a piano bench while the actor comes into the scene wearing a lab coat. The actor, here playing the role of a scientist, grabs the lamp shade and places it on the dancer's head. This *activates* the dancer who starts moving in a semi-mechanical way. Part of the dancer's movement instructions is to go back and forth, the same way the Disklavier chords are being arpeggiated up and down. This is in thematic reference to the idea of particles and virtual particles that are supposed to go in opposite directions.

This piece also offers a strong critique of modern society and its relation with technology. In particular, it critiques the blind belief that social media and virtual reality are inherently positive for individuals and society, and help to connect people without significant downsides. In 2016, I was struck by this episode at the Samsung Galaxy Unpacked event when Facebook's CEO walked to the stage between the crowd immersed into the virtual reality demo (McCormick 2016). On stage, and with everyone's attention focused on him, he preached that "One day, you're going to be able to put on a headset, and that's going to change the way that you live, work, and communicate [...] VR is going to be the most social platform" (*Galaxy Unpacked 2016* 2016). In direct reference to this episode, the electroacoustic intro includes a manipulation of a small speech sample, using reversed audio, an allusion to this Prelude's theme of virtual particles.



Figure 34 - Still frame of *Disklavier Prelude #3*. From (Hamido 2021b)

While working on this piece, I was also inspired by Gesu No Kiwami Otome's single *Tatakatte Shimau yo* (and its very provocative cover) containing lyrics that address the addictiveness of a mobile game, to a point where the user is "being played by the game" instead (Leicester 2018; Ronald 2018). This is the reason why the dancer begins the choreography with swiping and tapping on an invisible phone. The actual real phone attached to the wrist of the dancer is only used as a light source, but its color is changing in synchrony with the table lamp. In turn, the lampshade is a metaphor for the VR headset. And here there is another interesting analogy in Robert Gilmore's story, when the State Agent is teaching Alice about virtual particles:

"Here it is," he said triumphantly, "a marvel of modern technology. Just put this on, and you will see the world of virtual particles." Alice felt a little nervous as she contemplated the helmet. It was large, and it looked very complicated and even, she felt, a little sinister. (Gilmore 1995, 97)

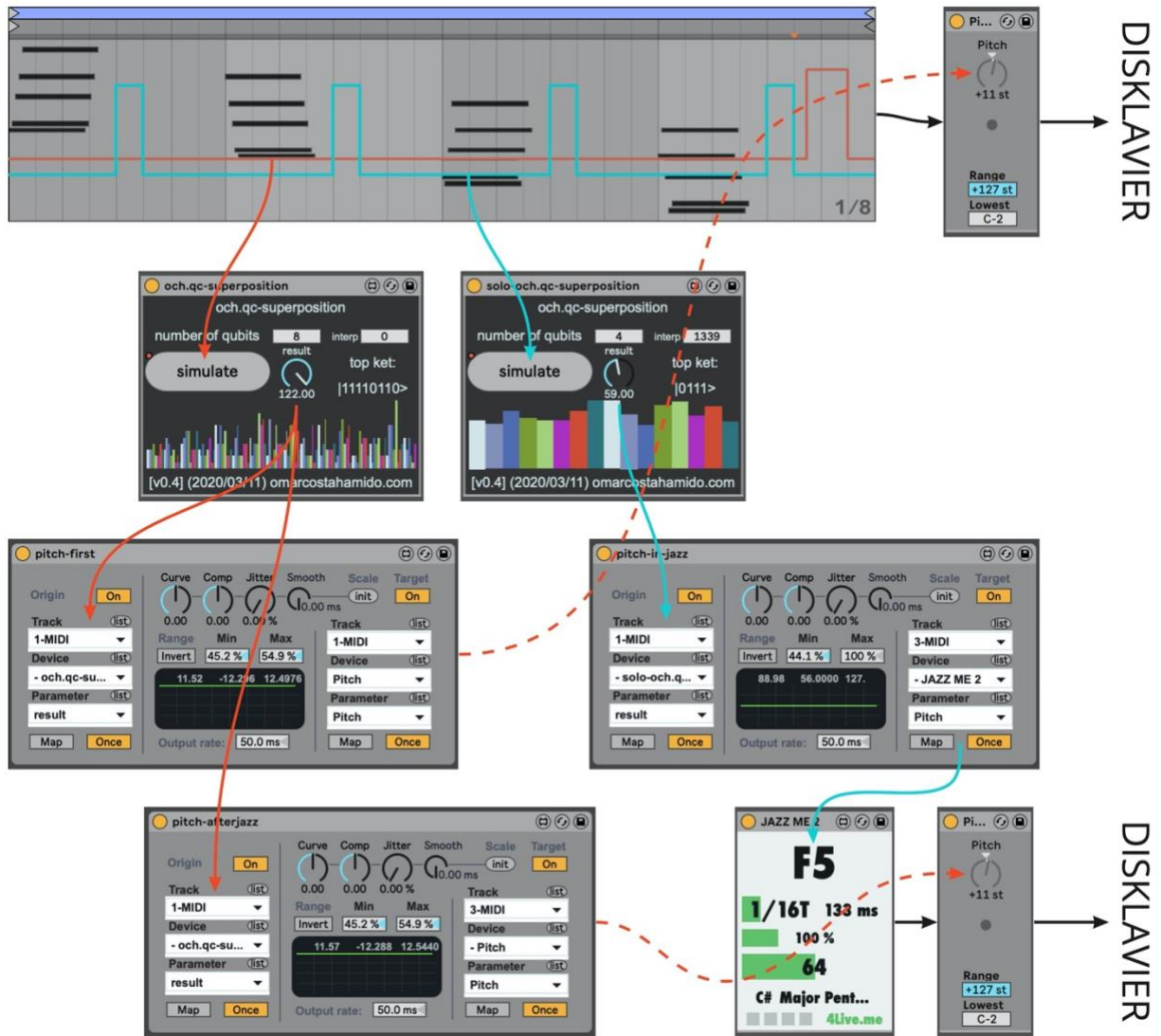


Figure 35 - One automation line triggers och.qc-superposition to control transposition, the other automation line triggers another och.qc-superposition to control a stream of notes on *Jazz me*

The choreography continues with the Disklavier accompaniment, until the lampshade falls from the dancer's head. At this point the dancer becomes aware of the surroundings and the controlling light, and the Disklavier adds a melodic line to the arpeggiated chords. The Disklavier plays maj7 chords all the way through, and these are transposed up and down based on the result of

the **och.qc-superposition** Max for Live device.⁴⁴ The melodic line that appears to be trying to communicate with the dancer, after the lampshade is lost, is composed of streams of notes from a pentatonic scale being transposed together with the chords, and being driven by another **och.qc-superposition** device.

In figure 35, we can see the MIDI clip with the arpeggiated maj7 chords and the automation lanes that trigger each of the **och.qc-superposition** device instances. The first automation lane only triggers a new simulation at the end of the clip. The **och.qc-superposition** instance here is configured with a high qubit count because I needed to have finer resolution while controlling the transposition; interpolation is set to zero because those changes should take effect immediately. The transposition then is achieved simply by mapping the result knob to the pitch knob, on a “Pitch” device, that transposes the MIDI information being sent to the Disklavier.

The second automation lane is enabled later in the piece, after the lampshade falls out, and it is responsible for controlling the melodic stream of notes. This automation triggers, every bar, a second **och.qc-superposition** device configured with a lower qubit count and a much higher interpolation time. The result knob then slowly transitions between results every time it is triggered, and this movement is mapped to an instance of Yehezkel Raz’s *Jazz me* device that generates a stream of notes in a major pentatonic mode. These notes are guaranteed to stay in tune with the arpeggiated chords because they are also being transposed by the first automation lane chain.

This complex network of interdependencies, including in the synchronized lights, allowed to express a very controlled scenario of a seemingly unknown outcome. Since the QC process is relying on a simple circuit with qubits in equal superposition, any possible value is equally likely. In turn, the

⁴⁴ See more about this device in the discussion in [Second Cornerstone](#) below.

repeating rhythmic motif was also very important to be able to articulate with the dancer, by allowing her to express the constraints, as well as break free from them, and freely explore this environment.



Figure 36 - Still frame of *Disklavier Prelude #3*. From (Hamido 2021b)

Disklavier Prelude #4



Figure 37 - Still frame of *Disklavier Prelude #4*. From (Hamido 2021b)

The last Prelude, which is also the last piece in *The Gedanken Room*, is a duet for Disklavier and saxophone. The Disklavier is playing and transposing whole-tone-mystic chords, articulated as a series of tritones and major thirds - e.g. C4 F#4 Bb4 E5 G#5 D6. The saxophone starts the Prelude by playing once again the main notes of the mystic chord, but here substituting the 9 for the #5, stating that this piece will clearly have the whole-tone sonority. During the series of transpositions that follows, the saxophone player is improvising, trying to anticipate and respond to the harmonic changes imposed by the Disklavier.

Prelude #4 also completes the overall movement drawn by all Preludes of playing melodically with an increasing number of notes while, at the same time, reducing the number of possible

harmonic combinations and possibilities too. The intended emphasis was on 4 notes in Prelude #1, 5 notes, or pentatonic scale, in Prelude #2 and #3, and six notes, or whole-tone scale, in Prelude #4. In this last piece, the repeating rhythmic pattern allows the saxophone player to freely improvise and easily anticipate the possible voice leadings between harmonies.

Consistent with the rest of the pieces, this is being videotaped with a moving camera. In this piece in particular, this movement allowed for the exploration of another level of analogy with QC/QM. Depending on the rotation and perspective, sometimes the dancer appears. When not in the shot, the dancer is actually controlling the RGB light attached to the Disklavier using a remote controller. The idea is to express an uncertainty relation between the dancer's exact position and movement, where the movement is expressed by the light color change at the same time that the exact position remains invisible, that is, unknown.

Telematic Entanglements



Figure 38 - Still frame of *Telematic Entanglements*. From (Hamido 2021b)

Between *Disklavier Prelude #1* and *Disklavier Prelude #2*, and after the first Interlude, I present *Telematic Entanglements*. This is a piece that, as the title suggests, deals with telematic music and QC, and builds on my work in recent years with telematic music composition, web and networked media technologies.⁴⁵ For this piece, I was able to work with New York based jazz saxophonist Will Vinson, now based in Australia, and drummer and CSU's director of Jazz Studies Tina Raymond. These are two musicians that I never thought I would be able to work with, but a small window of opportunity that appeared in the middle of the pandemic made it possible.

⁴⁵ A list of tools for networked media, that I decided to start as an open source project, in the beginning of the pandemic in 2020, can be found in (Hamido et al. [2020] 2021).



Figure 39 - Still frame of *Telematic Entanglements*. From (Hamido 2021b)

This work is also trying to draw important bridges between more traditional jazz musicians and new experimental music practices. Though networked music with computers can be traced back at least to the League of Automatic Composers in the 1970's (Bischoff, Gold, and Horton 1978), Telematic music as we know it today has been around for well over a decade now, too. However, the adoption of this medium was not widespread until very recently, when everyone was forced to connect over the internet. This technology has the potential to greatly influence the music itself, if the composers and performers using it intend not to replace a regular shared space scenario but, instead, try to make new music for this medium specifically. Such is the approach with Michael Dessen's scorestreams (Dessen 2010).

Though not part of *The QAC toolkit* Max package, the **och.scorestream** device, presented at the Ircam Forum Workshops 2019,⁴⁶ is a response to the work done by Dessen, Georg Hajdu (Hajdu n.d.), the Grame group (Fober et al. n.d.), and Joseph Branciforte (Branciforte 2017), among others. It is a simple-to-use Max for Live device that generates a list of MIDI clips for each given image inside a folder that the user specifies. These images are served over the internet, to any browser-enabled device, and can be triggered simply by firing the MIDI clips, also taking advantage of all the sequencing and algorithmic music capabilities that are built into Ableton Live (see figure 40).

⁴⁶ For reference see (“Ircam Forum Workshops - Paris 2019” n.d.; Hamido 2018).

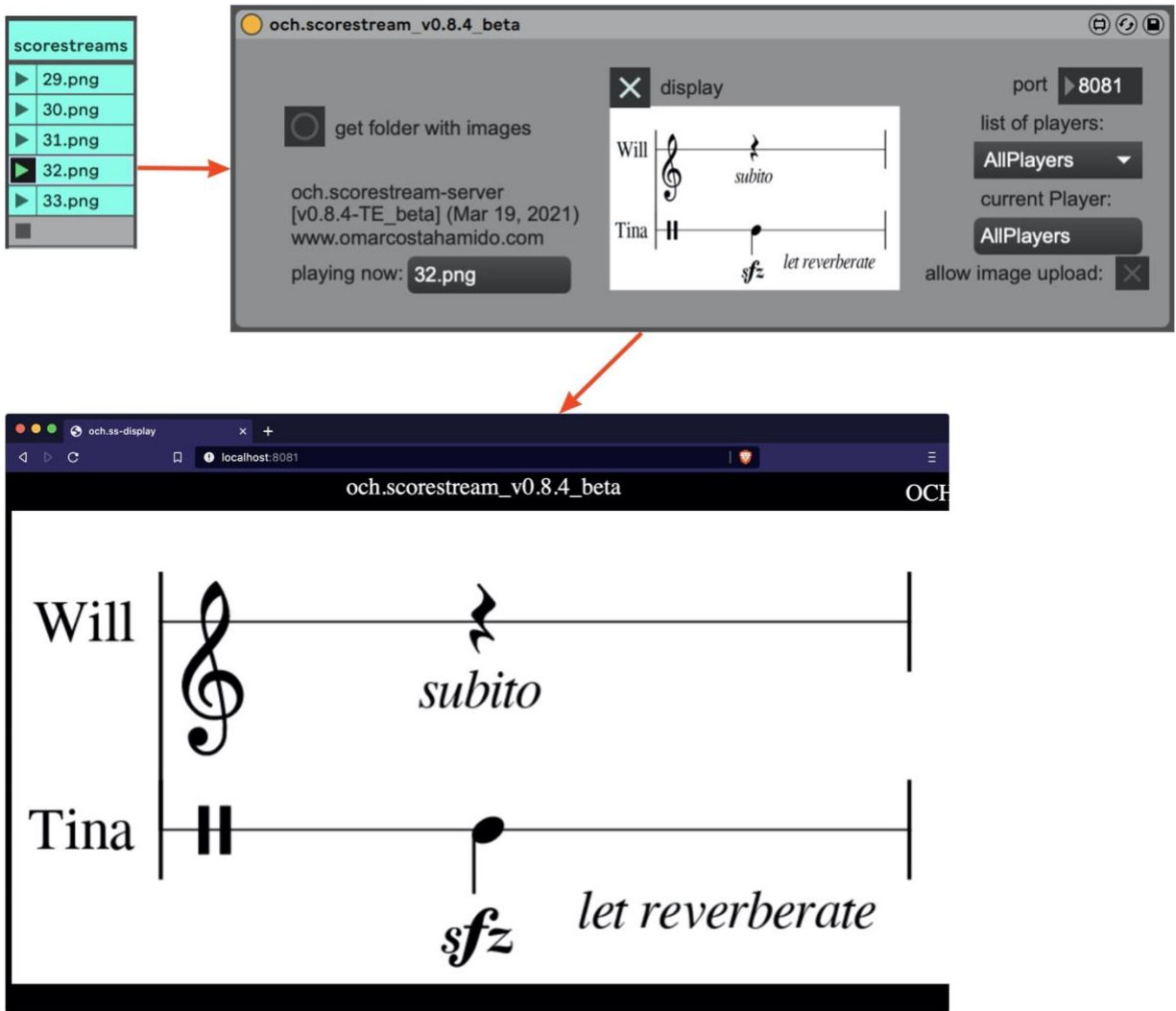


Figure 40 - och.scorestream uses MIDI clips to trigger the change of slides that are sent to performers connected with any browser enabled device

Telematic Entanglements uses **och.scorestream** to send different sections of a score to both performers in L.A., California, and Adelaide, Australia. In different sections of this piece, there is a small Max for Live device using **och.microqiskit** that selects which order of events to play next: in effect, firing MIDI clips automatically, and momentarily replacing the built-in sequencing functions in Live. Unlike any other piece in the program, this piece was first performed live, and recorded, before the filming of *The Gedanken Room* took place. Each performer was at their home, and I was

also in my home, connecting and recording them, and taking the role of electronicist/conductor. I was in charge of making sure the recording was happening, but also activating the QC process and performing live digital signal processing.

Unlike the audio routing, the scorestreams setup was rather simple. One track contained all the different slides that served as guidelines for Will and Tina's improvisation. Each slide was either triggered by me or by **och.qc-circ**, sitting on a parallel track, that runs a circuit and triggers one of the possible slides in the scorestreams track, as illustrated in figure 41.

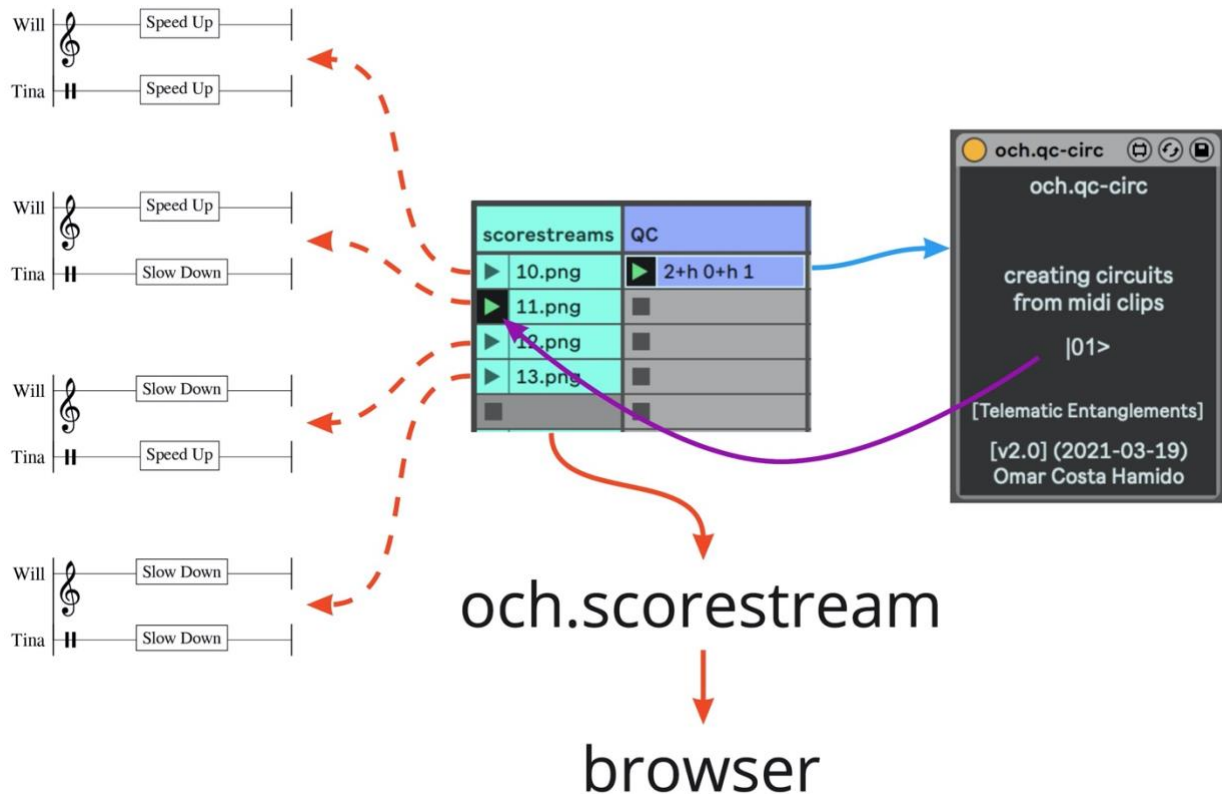


Figure 41 - och.qc-circ uses MIDI clips to generate quantum circuits whose result triggers one of the MIDI clips in the track on the left. In turn these MIDI clips trigger a change in och.scorestream

For this piece I am drawing on the fact that both Will and Tina are very talented musicians that can make use of a large vocabulary in this highly improvisatory and collaborative piece. The

concept of entanglement here is both suggested metaphorically, with the performers playing together at a distance, as well as more practically explored, with the score that *changes the state* of both performers at the same time. Most of the score slides contained written instructions that the performers had to react to instantaneously and connect to what they were doing.⁴⁷ Some of these instructions, like “Slow down” and “Speed up,” worked much better than the original ones I had, with precise metric modulations, due to the unavoidable latency of exchanging high quality audio across different continents. By purposely sending limited information to the performers, I was able to send it faster; and the simpler, more open notation allowed them to more easily articulate and integrate the new instructions, and move forward in the performance.

In terms of the practical integration of QC, this piece uses a similar implementation to the one explored in *Disklavier Prelude #1*, except that here I am triggering MIDI clips instead of generating MIDI notes. Besides the exact structure sequence of this performance, that was unknown each time we performed the piece, since it was being decided by the **och.qc-circ** device, another important artistic consideration was given to the definition of the sonic space. I wanted to create a space that allowed for something more than the sum of their parts to exist. I decided then to include live audio processing of their signals but, unlike the different score slides they got to practice and would eventually be able to predict what options would be possible at which point, I never told them when or exactly what effects I would dial in. In turn, this helped both creating a new space and enhancing the score. For example, when Will is playing over a progression of maj7#11 chords, Tina’s steady groove drums are passing through a series of resonators tuned to the pitches of the same chord. This was especially important in order to exploit the telematic medium not only as replacement of a physical space, but as a new environment altogether.

⁴⁷ see list of slides in [Appendix D](#).

With the pre-recordings done, I and my cinematographer for *The Gedanken Room*, Grant Speich, decided to play them on 3 vertical screens back to back in the center of the room. The vertical screens give each remote performer a physical presence in the room that, together, form this sculptural structure that emerges in the center of the room - making it consistent with the theme of *The Gedanken Room*. The fact that the screens are back to back makes it so that only the image of one screen can be seen at a time. This stimulated an exploration of the re-recording footage as a choreography of the moving camera. Several takes were needed in order to synchronize the camera movement with arriving at each screen at relevant moments. By coming very close to the screen there is an intention to both highlight smaller details and fill the screen with more abstract colors and light. Incidentally, each performer's own video recording displays only half of their bodies, in another layer of visual counterpoint to the concept of entanglement.⁴⁸ Like every other piece in this film-recital, this was shot in a single continuous take.

⁴⁸ Though not clearly explored in this piece, this creative exploration of entanglement, where each performer is *half of something else*, also draws inspiration from the "myth of the orange" shared by Aristophanes in Plato's Symposium (aka The Banquet).

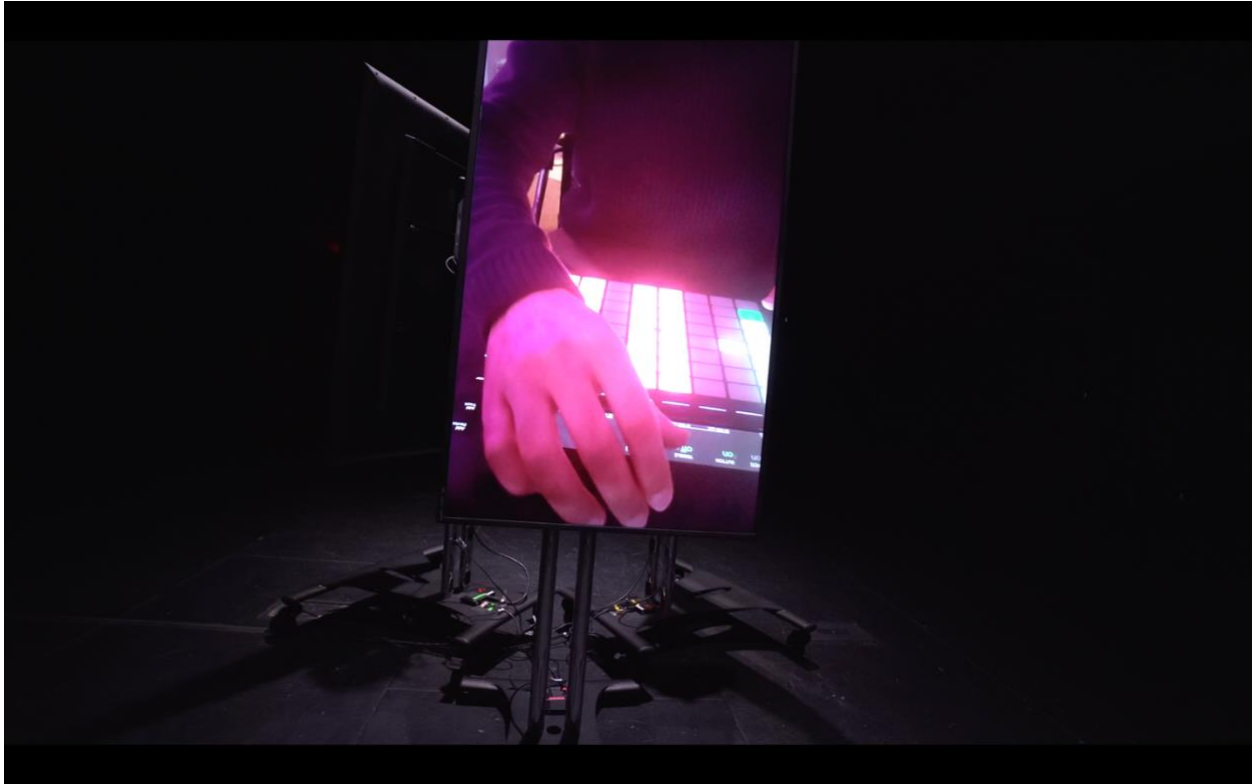


Figure 42 - Still frame of *Telematic Entanglements*. From (Hamido 2021b)

Second Cornerstone



Figure 43 - Still frame of *Second Cornerstone*. From (Hamido 2021b)

The third piece in the program, presented between *Disklavier Prelude #2* and *Disklavier Prelude #3*, is *Second Cornerstone*. This is the oldest of the four pieces, written in early 2020 before the pandemic began. This piece is inspired by a passage of Carlo Rovelli's book, titled *Reality is not what it seems*, where he explains the second cornerstone of QM as:

This is a second cornerstone of quantum mechanics, its hardest key: the relational aspect of things. Electrons don't always exist. They exist when they interact. They materialize in a place when they collide with something else. The "quantum leaps" from one orbit to another constitute their way of being real: an electron is a combination of leaps from one interaction to another. When nothing disturbs it, an electron does not exist in any place. (Rovelli 119)

The orchestration of this piece is a metaphor for the idea Rovelli elaborates here. On the one hand there is the kamâncheh, a Persian string instrument with more than 1000 years of history and repertoire, and part of the UNESCO Intangible Cultural Heritage list⁴⁹; on the other hand there are the music stands, which are not a proper musical instrument, let alone have any history or repertoire as such. My point is precisely to illustrate that music is also relational, and that objects become musical instruments upon a certain relational context. Furthermore, music stands, as merely supporting and utilitarian accessories, carry an invisible presence in the music concert setting. Like the electrons that “don’t exist” until they “collide with something else,” the music stands here only become “visible” when the performer engages with them as a musical instrument.

The score for this piece includes mostly timing and dynamics information, and instrumental techniques to use.⁵⁰ The music stands are struck with yarn mallets, both with the head and the handle. In order to extract different frequency ranges and timbres, the stands are also bent and rotated. The kamâncheh, performed by my colleague Niloufar Shiri, is bowed above and below the bridge, in *ordinario* and *col legno*, strumming and using hand *pizzicatos* on the strings and tuning box, which is also slightly struck with the bow, exploring both continuous sound and short attacks. This piece, similar to all others in this program, becomes alive in the human interaction it enables.

⁴⁹ See (“Art of Crafting and Playing with Kamantcheh/Kamancha, a Bowed String Musical Instrument - Intangible Heritage - Culture Sector - UNESCO” n.d.).

⁵⁰ See the score in [Appendix E](#).



Figure 44 - Still frame of *Second Cornerstone*. From (Hamido 2021b)

During the execution of this piece there are a couple moments with audio effects being applied to the acoustic sound of the duo. The parameters of these effects are being controlled by the **och.qc-superposition** Max for Live device, which creates a simple circuit with a Hadamard gate and a measurement gate applied to each qubit, effectively putting all qubits in a state of equal superposition. The process of writing the actual circuit is abstracted away from the user who, instead, only has to define the number of qubits (and classical bits) to allocate. This number influences the total number of possible state results, and serves as a way to control the resolution of the one mappable knob, with its range divided in $2^{(\text{number of qubits})}$ parts.

The first noticeable audio effect being applied to the sound of the duo is an oscillating reverb. After the sound is routed to a reverb processor, the reverberated sound is split into single left and

right mono channels whose panning is driven by an LFO. In effect, these left and right channels trade places, and with the help of a “Max Api Param2Param” control device it guarantees that the right channel’s balance value is always opposite to the left channel’s balance value (see figure 45). The idea here was to refer to the basis axis of measurement used in QC, and also to confuse the listener about the acoustic space. This also served well to bring together this piece with *Disklavier Prelude #2* and *Interlude #3*, as will be explained later in this chapter, in [Interludes and the larger narrative](#).

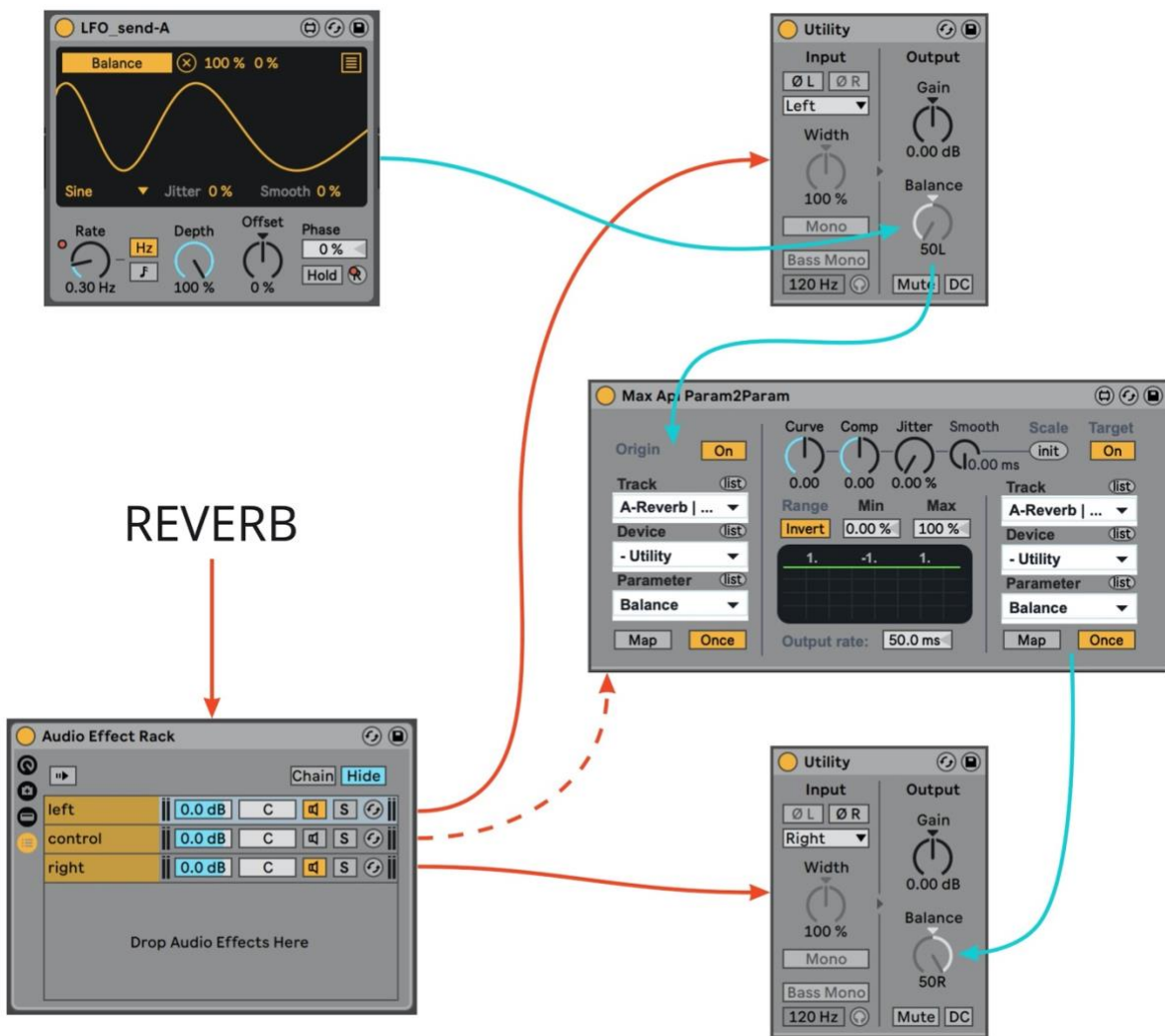


Figure 45 - Reverb effect in *Second Cornerstone*

For each other section where the effects are active, there is an automation that repeatedly triggers the *simulate* button of a **och.qc-superposition** device, and enables the controlled audio effect to change over time. The interpolation time is also automatable, affecting the *result* dial change, whenever it receives a new value. This device is also used, with a different mapping, and with a different role, in *Disklavier Prelude #3*. Here, this process is being used to control either panning position or the ring modulator's fine tuning (see figure 46).

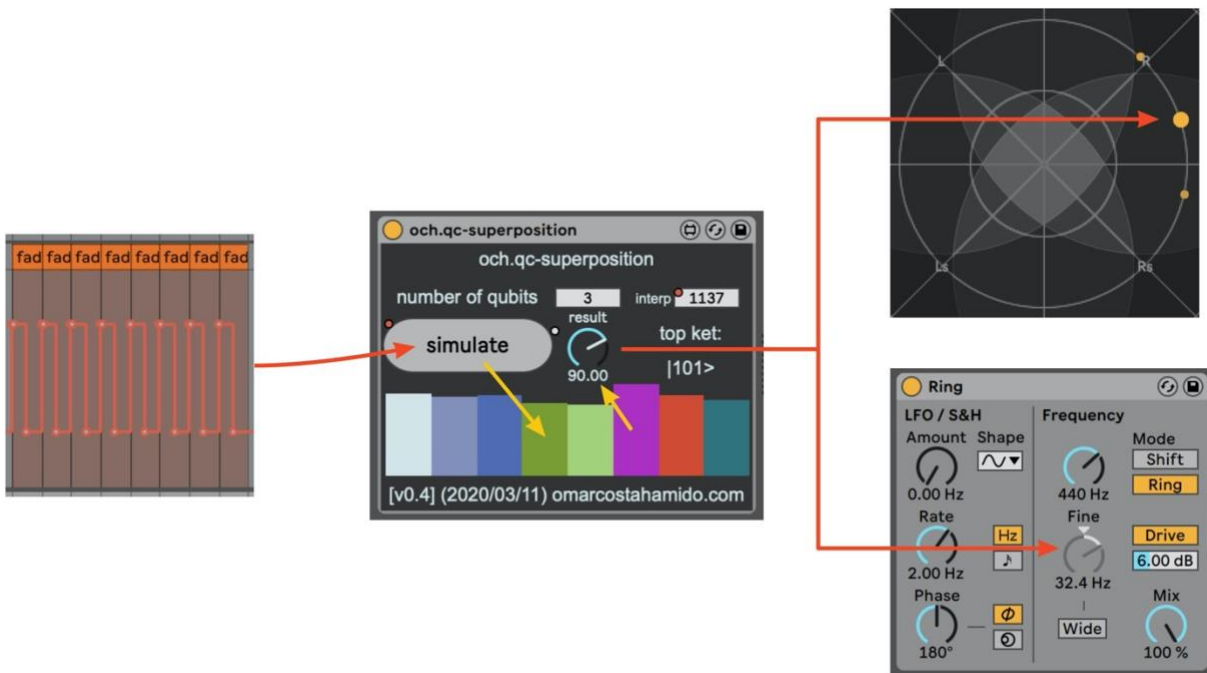


Figure 46 - och.qc-superposition controlling FX in *Second Cornerstone*

Decoherence



Figure 47 - Still frame of *Decoherence*. From (Hamido 2021b)

The last piece in the program is *Decoherence*, a piece for electronics, saxophone, and video. This piece uses a new version of the Quantum Synthesizer, mentioned in the previous chapter. I am playing the saxophones, and JNTHN STEIN (Jonathan Stein), electronic musician and Push controller virtuoso, is playing the *QuantumSynth*. In terms of the narrative sequence, this piece appears after *Interlude #5*, where the presenter is turning on a small analog TV that displays static noise, and the viewer is getting *inside* the TV. This *dive into the TV* is the third and final direct reference to Robert Gilmore's book, and also references the original story by Lewis Carroll, when the looking glass melts away and Alice traverses to Wonderland.

This piece takes place, metaphorically speaking, inside the TV, and it is an exploration of noise, both visually and sonically, including the quantum noise on the *QuantumSynth*. The title of the piece, *Decoherence*, is a reference to the Quantum Decoherence effect that is a cause of information loss in QC. The back wall of the room is displaying static noise as if the screen of TV, seen from the inside. However, the small TV still appears on the stage connected to the electronics setup. It was hacked, with the help of my colleague and hardware expert Chris Hadley, to work as an oscilloscope. Here it is drawing the waveshape of the *QuantumSynth* output, either only vertically or horizontally, or both. I take another momentary acting role playing with these options, and amplification as well, at the beginning of this piece. At the same time, Jonathan is playing with the *QuantumSynth*, starting from a very noisy sound, and moving to a more pitch-defined sound.

Working with Jonathan involved first a relationship different from composer and performer, but more like instrument designer and instrumentalist. I have known JNTHN STEIN's work since mid 2010's when I first found the music of several members from Team Supreme (Kleinermanns 2016) and became interested in working with the Push controller. In line with the envisioned collaborations at the end of chapter 2, I consider this work with JNTHN STEIN an important step to create a dialog between different musical creative practices and environments. It was a really interesting experience teaching Jonathan about the basics of QC, and how its outputs are being used in the *QuantumSynth*, as well as seeing how this different way to think about the parameters of a synthesizer informed his approach to explore, and to help me improve, the expressiveness of this instrument.

In the two days that we had to work together, we spent considerable time working on the *QuantumSynth*, turning it into an interesting instrument to engage with and working well with Jonathan's setup. I had prepared a version 2 of the Quantum Synthesizer that had several options for running the circuits, including on the real QPU devices. But the initial feedback that I received from Jonathan was that there were too many options, and it was hard to control it. Another interesting comment was that he wished for it to "do something on its own", to show more of its personality

without the user having to press several buttons. This feedback, as well as a conversation with him on being conscious about the body, led me to quickly rewrite a new version that was simpler.

The new version 3 of the Quantum Synthesizer was then born overnight: a wavetable synthesizer running pre-built circuits on a statevector or simulator backend with up to 8 qubits and 4096 shots. The main difference was that it was restructured to run in 3 different modes: *body*, *hands*, and *fingers*. Each mode determines how the instrument will react, or *act on its own*. In *body* mode, it executes the circuit in regular time intervals determined by the large *speed* knob - the idea here being that it acts as long as there is a human body present. The *hands* mode executes the circuit and keeps its result as long as the user keeps at least one hand in connection with the controller. And the *fingers* mode executes the circuit for every new finger coming into contact with the controller. With each new execution there is, in simulation mode of course, a new timbre variation being generated.

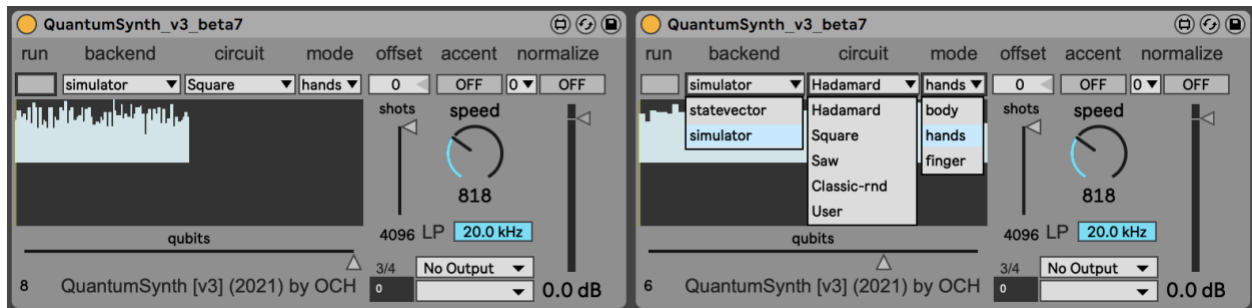


Figure 48 - *QuantumSynth* v3 (left) a simulated square circuit and (right) interface options

Building this instrument, in a highly collaborative process with Jonathan, required the development of multiple iterations that could be easily shared and quickly put to the test. The recording of the *Decoherence* piece itself required several takes to make sure the different sections were articulated. For the first section, where Jonathan needs to start with a very noisy sound, he used a very high qubit count and number of shots, with a very fast speed. This implies a very complex

waveform that is changing very fast, and it worked well to control the noisiness quality of the sound. Following this section, there is an ethereal entry of the soprano saxophone with an improvisation.

After some point in this piece, the performers switch roles and have the electronics doing a solo. Later the alto saxophone enters, but playing mostly noisy sounds. The two instruments in this piece draw opposite trajectories between noise and more pitch-defined sounds, and a counterpoint between frequency registers. The performers also exchange the acting roles, as the electronicist resets the initial TV configuration to playing back static noise, while the saxophone then plays a more white noise-like sound to end the piece.



Figure 49 - Still frame of *Decoherence*. From (Hamido 2021b)

Interludes and the larger narrative



Figure 50 - Still frame of *Interlude #2*. From (Hamido 2021b)

As mentioned above, all the pieces in *The Gedanken Room* are alternated with an interlude. These interludes do not contain any music but serve the purpose of giving context to the pieces, and helping articulate them, both in terms of narrative and the visual transitions, that were envisioned to make the whole recital-turned-film a more continuous experience.

The first interlude, taking place after the *Disklavier Prelude #1* and the title screen, includes a small introduction welcoming the viewer to *The Gedanken Room*, explaining that I am exploring the implications of QC for Music Composition and Improvisation. In the background, revealed while the narrator is walking, there are two flatscreens displaying alternate realities for what *Disklavier Prelude #1* could have been, or looked like. It then concludes with an invitation to imagine a telematic

music context with QC involved in generating a stream of information that can work as a musical guideline, a segue to *Telematic Entanglements*.

The second interlude includes only a pedestal with a Google Voice Kit DIY box that has been programmed to respond to my inquiry about Heisenberg's Uncertainty (see Figure 50). This is the same box that I've programmed for my interactive installation piece *Duet for Human and Machine*.⁵¹ For this interlude, the box is prepared to reply by citing the Carlo Rovelli passage mentioned earlier, used in *Disklavier Prelude #2*. This section is also accompanied by a camera zoom and dolly into the dark front holes of the box, setting the scene for the piece that follows.

After *Disklavier Prelude #2* comes the third interlude, in which the presenter continues reading, from a physical copy of Rovelli's book, the story of Heisenberg's insight which relates to the theme of *Second Cornerstone*. After *Second Cornerstone*, the viewer is drawn out of the box in the fourth interlude, with the box citing the last passage from the book that concludes the definition of QM's second cornerstone.

This narrative constructed by the second and fourth interludes places all the events for *Disklavier Prelude #2*, *Interlude #3*, and *Second Cornerstone*, inside the box. This "Inception"⁵² or "Matryoshka doll" effect, of having a room inside a room, is one of the artistic choices that makes this recital a cinematic experience. Symbolically speaking, the matryoshka metaphor still holds from the point of view where the box is one of my former creative offspring that contains the core elements for engaging in an interaction capable of inspiring another generation of creative works. Special attention was given to the spatialization and other sound effects between *Interlude #2* and *Interlude*

⁵¹ A collaboration with playwright, Andrea Sloan Pink, that was presented inside Richard Serra's *Connector* sculpture in 2019 (Hamido and Pink n.d.).

⁵² In reference to Christopher Nolan's film (Nolan 2010).

#4. When inside the box, listening to its voice from the inside, its timbre is different, and so is the room resonance during *Interlude #3* also different.

The fifth and sixth interludes both feature the analog TV, on top of the lone pedestal in the middle of the dark room again, as it is turned on and off, respectively. Accompanying this action, the camera dolly shot zooms and moves into and out of the TV, in a similar fashion to what was used for interlude 2 and 4. This camera work suggests that the events of *Decoherence* take place *inside the TV*, and that the static noise in the background of *Decoherence* symbolizes the TV screen, seen from the inside. These camera movements, including the movement choreographies like the one described in *Telematic Entanglements*, were studied and practiced in order to achieve the continuous experience envisioned.

All the sections and pieces of *The Gedanken Room* were recorded as full takes, with multiple simultaneous microphone techniques engineered by G. Blake Harrison-Lane, and then mixed and mastered. The main reason for that was to enable the consistency and uniqueness of each take and each improvisation moment they contained. The only piece completely fixed was *Disklavier Prelude #2*.

In terms of video, I had decided early on that I wanted a single moving camera, an approach that I've been exploring for the video recording of previous recent works like *The Space Between the Strings* (Hamido 2020a), and Alex Lough's *Assemblage no 4* (Lough 2020). My cinematographer collaborator for this film-recital, Grant Speich, used a Sony 4k super 35mm, which is a digital cinema camera, mounted on a gimbal. The choice of the 35 mm sensor allows for a shallow depth of field, along with a wider field of view, to create a more cinematic experience. In effect, the shallow depth of field makes it focus on closer subjects and lose detail on the background, which worked very well as an advantage here to create the illusion of a borderless room, with its thought experiment subjects taking shape in the middle of the void.

The poster for *The Gedanken Room* includes just the pedestal on its own, in the empty dark room (see [Appendix F](#)). This image captures the central idea of the *thought experiment*, where one is expecting the thoughts to embody the object to be on display, which is seemingly absent at first, on the empty pedestal. It also references the idea of QC as a probabilistic means of computation where we often want to maximize the number of results with the desired outcome state. In traditional QC programming environments this is represented as a column on a histogram, as described in the previous chapter; here, it is depicted as a column emerging from the floor.

In retrospect

The preparation and production of the dissertation recital was incredibly challenging for several reasons. It included very different pieces, with very different instrumentation, which meant collaborating both with colleagues and outside performers, some of whom I had never worked with before. The recital was projected to take place in the Experimental Media Performance Lab (xMPL) in the Contemporary Arts Center, at the Claire Trevor School of the Arts. However, due to the current pandemic, no audience was allowed to attend a public presentation of the creative work in person. Producing a film-recital was something that was already in the back of my mind, given my affinity for cinematographic experiences. As soon as I was aware that the pandemic constraints were not going to be lifted any time soon (this was Summer 2020), I started writing the script and storyboard.

The constraints imposed by the pandemic also affected the number of performers allowed in the space, and the access time to each room. I made the executive and logistic decision to only do a recording, and no livestream, splitting the recording of each piece to different days within the 2 week time window that I was allowed access to the space, between March 29 and April 9, 2021. This greatly contributed to keeping the number of people in the shared space at any given time to an absolute minimum. And, I believe, it was one of the reasons my proposal was accepted.

The core team was then composed of me, as the director-composer-performer, Grant Speich, as the cinematographer-camera and light operator, and G. Blake Harrison-Lane, as the audio recording engineer. The total number of performers was 6. Besides me, Niloufar Shiri, Waeli Wang, and JNTHN STEIN also came to the xMPL in person, while Will Vinson and Tina Raymond were recorded telematically 2 weeks before the start of the xMPL recording sessions.

In retrospect, I believe that the core team was small for the amount of work required. As my friend Grant Speich said on the last day: “we just did the work of 10 people.” Wrapping up the recording of a new section or piece each day, and being able to preview it, gave the team a sense of accomplishment and motivation to continue. Working in the space with the performers was also an incredibly enriching experience, and it is where the work really comes alive. The choice of the performers was long premeditated. My work tends to feature the creative contributions of the performers for whom I am writing, and the way they react and interpret my scores, instructions, directions, or challenges.

The only performer I was able to rehearse with beforehand more consistently was Niloufar Shiri with the *Second Cornerstone* piece, in early 2020, before we lost complete access to the labs and shared spaces. The *Telematic Entanglements* piece was an exception, as it was set to use the telematic medium, and I was able to get 2 sessions and a recording between September 2020 and March 2021. Interestingly enough, I had previously collaborated with Waeli Wang, back in 2017, on a piece that also involved working with a phone on her wrist, displaying colored light and sending back 9dof sensor data to an electroacoustic setup.

With the various creative adventures in *The Gedanken Room*, I was able to realize that working with QC requires most of the time to be expecting the unknown. Given the nature of this new computational paradigm, as a probabilistic means of computation, shorter computations, with a reduced number of shots (which are faster to simulate), tend to return more chaotic results, while

longer computations, with a higher number of shots, tend to output more faithful and predictable results. In the pieces present in *The Gedanken Room*, I tried to play around with this tension by including the QC process in the middle of a larger network of classical information processing, that was able to sometimes smooth out or interpolate between faster and more dispersed results.

As it is clear from this and previous chapters, my work was very technology intensive, and considerable time leading up to the recording sessions was spent making sure the tools would work. All the way through my PhD I felt that switching between roles was a constant, trying to address different expectations. And coming up with new creative work integrated with technological research tends to be a very lonely process. This project was no exception. Oftentimes it was hard to relate to and establish a dialog with my peers and even professors. I can blame this on the specificity or *uniqueness* of the work, or simply on my inability to present it in a more interesting or approachable light. If anything, I wish there had been more time for writing and playing music, for more discussions, for more collaborations.

From piece to piece, interlude to interlude, *The Gedanken Room* sets the stage for the exploratory creative process that tries to operate emerging paradigms in Quantum Computing, both metaphorically and experimentally. In this sense, I think this was a meaningful and successful artistic research project, and a very important milestone, not only in this PhD process but also in my path as a creative artist and researcher.

Conclusions

Quantum computing is on the verge of sparking a paradigm shift. Software reliant on this nascent technology, one rooted in the physical laws of nature, could soon revolutionize computing forever. [...] We think we can get there, but we can't take this leap alone. (Wehden, Faro, and Gambetta 2021)

My objective in this research was to contribute to identifying and understanding some of the implications of Quantum Computing's emerging paradigm for artistic creation. In doing so, I took on the perspective of my own creative practice, and developed an open exploratory process *with no end in sight*. This is *how I got here* (for now).

The current work was strongly informed by years of readings, skill development, and research in many different subjects, as well as multiple and varied collaborations, several of them mentioned throughout this document. There was a constant tension between spending time developing the software, developing the creative work, the writing, and the outside world. Sometimes I wished that the disconcerting idea of a mind-dependent world would be possible (at the large scale), such that, when I'm not looking, it wouldn't exist or just pause for a moment.

The implications of QC for artistic creation, as formulated in the research questions, are thought of as reciprocal, doing justice to the non-linear and complex character of the knowledge that serves as a source of inspiration. Not only did I question (Q1) what innovations in the field of composition and improvisation could happen under the influence of QC, but I also questioned (Q2) how creative realtime practices themselves could stimulate the development of new QC based tools.⁵³

⁵³ Please refer to the "[Introduction](#)" chapter.

In chapters 3 and 4, as well as in the creative work presented, I addressed Q1 by demonstrating and analyzing how my practice, both as a computer music programmer, composer, performer, and director, was influenced by the close study and integration of QC. The artistic choices, while using QC-enabled tools, illustrate this and how the compositional process changed. Ultimately, this describes how working with QC has led me to think about the compositional process itself in a new way, writing quantum circuits instead of notes on a paper, and using these tools to generate content and make structural decisions in realtime.

With the main focus on the composer's perspective, and the compositional and improvisational process, the advantages of QC for art making include the computational power and complexity, the advent of faster computation and a truly unbiased collaborator, and the motivation to think differently. While understanding how to do this, to rearticulate problems in a way that makes use of QC and its potential advantages, artist-researchers will also be engaging with a larger community of researchers and promoting a dialogue between science and art. I argue this will be the case even when exploring seemingly different directions. For example, when some want to minimize the quantum noise while others try to embrace it, both are engaged intellectually in understanding the same phenomena and how it affects their practice.

While the overall narrative for *The Gedanken Room* was constructed as one continuous cinematic experience, each piece tried to explore the integration of QC in very different ways. In *4 Disklavier Preludes*, QC is used to generate and manipulate MIDI notes; in *Telematic Entanglements*, QC is used to determine the structure and direction of the performance; in *Second Cornerstone*, QC is used to control live audio FX; and in *Decoherence*, QC is used to generate wavetables, i.e. different timbres for the synthesizer. The way QC was integrated to achieve all this relied heavily on *The QAC toolkit*.

I have addressed Q2 with my creative work and in chapters 2 and 4. The toolkit development process, built with the creative artist and their realtime environments in mind, shows several examples of possible improvements for the available QC tools. The improved speed in simulation while using **och.microqiskit** is a relevant example, the finding of which was motivated by the intended artistic practice use. The several devices and interfaces built with this object and its companion, **och.qisjob**, also exemplify other approaches the QC community can adopt and explore.

Personally, building my own simulator was a great way not only to learn about QC but also to contribute to this community. At the time of this writing, among the several MicroQiskit versions, the one I've worked on is the only one to include a ch and ox gate, and the 3 qubit gates ccx, oox, ocx, cch, ooh, and och. And it is interesting to note that if the ch and ccx gates are accepted by Qiskit, the other gates are not yet directly available, but my MicroQiskit version is capable of translating these exclusive gates into other known and acceptable gates. This is, in effect, what happens when a user calls the *get_qiskit* or *get_qasm* method on **och.microqiskit**.

If my initial attempts at integrating QC in my creative compositional practice were operationalized in building purpose-specific devices, like the old synthesizer and **och.qc-circ-2q**, the emphasis of the work presented in chapter 2 is on developing general purpose tools, well integrated in creative programming environments, that allow other users to develop their own patches and devices. This will continue to be developed, as I receive feedback from other users, and to accompany some other updates in the QC field. In the near future I will also work on optimizing **och.qisjob**, in order to get execution times closer to regular Qiskit, and also to allow it to connect to different backends from different providers.

Keeping this low-level access while making it easier to develop a higher level of abstraction, from the underlying QC circuits and processes, seems to also be a future direction of development. In *The Gedanken Room*, there were many cases where the interface with QC, in the software used for

each piece, still often relied on the direct description of the underlying QC circuits. The only exception to this was on the software used in *Decoherence*, since it was the only piece where I was not the person operating the QC software. Since Jonathan was the person operating the QC process here, by playing with the *QuantumSynth*, special attention was given to the construction of an interface with the minimum QC jargon possible. But even then, having an idea of the underlying process proved to be very important to fully explore the instrument.

It will be interesting to see other future developments of this work including the integration or expansion of *The QAC toolkit* to work with a score engraver like Musescore (Froment et al. [2012] 2021), or even to see it being used in a totally different context, like engineers building a control feedback system. There is much space to explore in this field, both experimentally as in the research and artistic work presented, as well as theoretically, as in the (impractical) explorations mentioned in "[A QC-enabled Avant-Garde](#)", for example.

Practically every single day there is exciting new research being done in the field of QM and QC, so much that it is difficult to keep track of. For example, I can't wait to see the implications of very recently reported macroscopic entanglement experiments that are said to serve as "building blocks for CV quantum information processing and quantum communication" (Kotler et al. 2021).

In my limited capacity as a performer-composer-technologist, I can only hope for my work to be well received in the diverse human and creative landscape outside of the *gedanken room* living in my head. Writing this dissertation, producing this film-recital, and releasing this software is one attempt at that. But I consider equally important all the direct human connections established, in discussion and collaboration, developing a tangible sense of thinking and moving forward. It is my hope that, for others too, these tools, writings, and works, will inform new compositional processes, and inspire a new generation of composers who think differently.

Bibliography

- Ableton Live*. 2021. Ableton. <https://www.ableton.com/>.
- Acker, Anne Beetem. 2016. "Disklavier." Grove Music Online. May 25, 2016. <https://www.oxfordmusiconline.com/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-4002294057>.
- Agostini, Andrea, and Daniele Ghisi. n.d. "Bach - Computer-Aided Composition in Max." Bach - Automated Composer's Helper. Accessed February 24, 2021. <https://www.bachproject.net/>.
- Akama, Seiki. 2015. *Elements of Quantum Computing: History, Theories and Engineering Applications*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-08284-4>.
- Alexander, Thomas, Naoki Kanazawa, Daniel J. Egger, Lauren Capelluto, Christopher J. Wood, Ali Javadi-Abhari, and David McKay. 2020. "Qiskit Pulse: Programming Quantum Computers Through the Cloud with Pulses." *ArXiv:2004.06755 [Quant-Ph]*, April. <http://arxiv.org/abs/2004.06755>.
- Ames, Charles. 1987. "Automated Composition in Retrospect: 1956-1986." *Leonardo* 20 (2): 169–85. <https://doi.org/10.2307/1578334>.
- "Art of Crafting and Playing with Kamantcheh/Kamancheh, a Bowed String Musical Instrument - Intangible Heritage - Culture Sector - UNESCO." n.d. Accessed May 17, 2020. <https://ich.unesco.org/en/RL/art-of-crafting-and-playing-with-kamantcheh-kamancheh-a-bowed-string-musical-instrument-01286>.
- "Arts at CERN." n.d. Arts at CERN. Accessed May 3, 2020. <https://arts.cern/>.
- Asfaw, Abraham, Luciano Bello, Yael Ben-Haim, Sergey Bravyi, Lauren Capelluto, Almudena Carrera Vazquez, Jack Ceroni, et al. 2020a. "Defining Quantum Circuits." In *Learn Quantum Computation Using Qiskit*. <https://community.qiskit.org/textbook/ch-algorithms/defining-quantum-circuits.html>.
- . 2020b. "Learn Quantum Computation Using Qiskit." <http://community.qiskit.org/textbook>.
- Babbitt, Milton. 1962. "Twelve-Tone Rhythmic Structure and the Electronic Medium." *Perspectives of New Music* 1 (1): 49–79. <https://doi.org/10.2307/832179>.
- Bača, Trevor, Josiah Wolf Oberholtzer, Victor Adan, Andrew Yoon, Tiago Antao, Chris Swithinbank, Gregory Rowland Evans, et al. (2013) 2021. *Abjad* (version v3.2). Python. Abjad. <https://github.com/Abjad/abjad>.
- Bell, Adam. 2019. "Quantum Music – Towards a Unified Aesthetic." Thesis, Brunel University London. <http://bura.brunel.ac.uk/handle/2438/18286>.
- Bernhardt, Chris. 2019. *Quantum Computing for Everyone*. Cambridge, Massachusetts: The MIT Press.
- Bernstein, Ethan, and Umesh Vazirani. 1993. "Quantum Complexity Theory." In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, 11–20. STOC '93. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/167088.167097>.
- Bischoff, John, Rich Gold, and Jim Horton. 1978. "Music for an Interactive Network of Microcomputers." *Computer Music Journal* 2 (3): 24–29. <https://doi.org/10.2307/3679453>.
- Bouche, Dimitri, Jérôme Nika, Alex Chechile, and Jean Bresson. 2017. "Computer-Aided Composition of Musical Processes." *Journal of New Music Research* 46 (1): 3–14. <https://doi.org/10.1080/09298215.2016.1230136>.
- Bovermann, Till, and Dario Sanfilippo. 2020. "Creative Use of Bit-Stream DSP in Faust." In *Proceedings of the 2nd International Faust Conference (IFC-20)*. <https://ifc20.sciencesconf.org/332745/document>.

- Bowers, Faubion. 1973. *The New Scriabin; Enigma and Answers*. New York: StMartin's Press.
- Branciforte, Joseph. 2017. "From the Machine: Realtime Networked Notation." NewMusicBox. October 16, 2017. <https://nmbx.newmusicusa.org/from-the-machine-realtime-networked-notation/>.
- Bresson, Jean, Anders Vinjar, Karim Haddad, Geof Holbrook, Dimitri Bouche, Marco Stroppa, Raphael Yancey, and David Stephen Grant. (2018) 2021. *OpenMusic* (version v6.18). Common Lisp. OpenMusic. <https://github.com/openmusic-project/openmusic>.
- Brown, James Robert, and Yiftach Fehige. 2019. "Thought Experiments." In *The Stanford Encyclopedia of Philosophy*, edited by Edward N. Zalta, Winter 2019. Metaphysics Research Lab, Stanford University. <https://plato.stanford.edu/archives/win2019/entries/hought-experiment/>.
- Caraiman, Simona, and Vasile I. Manta. 2008. "QCL Implementation Of Quantum Search Algorithms." <http://www.mcq.tuiasi.ro/articole/II-16.pdf>.
- . 2009. "New Applications of Quantum Algorithms to Computer Graphics: The Quantum Random Sample Consensus Algorithm." In *Proceedings of the 6th ACM Conference on Computing Frontiers*, 81–88. CF '09. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1531743.1531757>.
- Carroll, Lewis. 1872. *Through the Looking Glass, and What Alice Found There*. London: Macmillan and Co. <http://archive.org/details/throughlooking00carr>.
- Cavill, Steve. 2015. "Number of Atoms in the Universe." Oxford Education Blog. November 24, 2015. <https://educationblog.oup.com/secondary/maths/numbers-of-atoms-in-the-universe>.
- "Computer." 2020. In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=Computer&oldid=954337387>.
- "Computer Music." 2020. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Computer_music&oldid=952707812.
- Cross, Andrew W., Lev S. Bishop, John A. Smolin, and Jay M. Gambetta. 2017. "Open Quantum Assembly Language." *ArXiv:1707.03429 [Quant-Ph]*, July. <http://arxiv.org/abs/1707.03429>.
- "Csound Community." n.d. Accessed February 23, 2021. <https://csound.com/>.
- Culp, Laura, and Greg Wroblewski. 2014. "Quantum Computing Playground." 2014. <https://www.quantumplayground.net/#/home>.
- Cuthbert, Michael Scott, and Christopher Ariza. 2010. "Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data." *Michael Cuthbert*, August. <https://dspace.mit.edu/handle/1721.1/84963>.
- Daniels, Trevor, Dan Eble, Jonas Hahnfeld, Phil Holmes, David Kastrup, Werner Lemberg, James Lowe, et al. 2021. *LilyPond* (version 2.22.0). <https://lilypond.org/>.
- Dean, R. T. 2003. *Hyperimprovisation: Computer-Interactive Sound Improvisation*. The Computer Music and Digital Audio Series ; v. 19. Middleton, Wis: A-R Editions.
- . 2011. *The Oxford Handbook of Computer Music. The Oxford Handbook of Computer Music*. Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199792030.001.0001>.
- "Derivative." n.d. Derivative. Accessed February 24, 2021. <https://derivative.ca/>.
- Dessen, Michael. 2010. "New Polyphonies: Score Streams, Improvisation and Telepresence." *Leonardo Music Journal* 20 (December): 21–23. https://doi.org/10.1162/LMJ_a_00007.
- Deutsch, David. 1985. "Quantum Theory, the Church–Turing Principle and the Universal Quantum Computer." *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400 (1818): 97–117. <https://doi.org/10.1098/rspa.1985.0070>.
- Deutsch, David Elieser. 1989. "Quantum Computational Networks." *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425 (1868): 73–90. <https://doi.org/10.1098/rspa.1989.0099>.
- Dobrian, Chris. 2012. "Realtime Stochastic Decision Making for Music Composition and Improvisation." In *Xenakis Matters*, edited by Sharon Kanach, 325–40. Pendragon Press. <https://music.arts.uci.edu/dobrian/AP2013/XenakisMattersDobrianChapter.pdf>.
- . n.d. "Pseudo-Random Numbers Appear Random | Max Cookbook." Accessed May 6, 2021. <https://music.arts.uci.edu/dobrian/maxcookbook/pseudo-random-numbers-appear-random>.

- “Electronic Musical Instrument.” 2020. In *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=Electronic_musical_instrument&oldid=952709061.
- Evanstein, Marc. 2020. *Cooking up Some Music in SCAMP!*
<https://www.youtube.com/watch?v=vpv686Rasds>.
- . 2021. *SCAMP*. Python. <https://github.com/MarcTheSpark/scamp>.
- “Festival Della Tecnologia - Musica Con Un Computer Quantistico.” 2019.
<https://www.festivaltecnologia.it/sessioni/musica-con-un-computer-quantistico>.
- Fiebrink, Rebecca. (2015) 2020. *Fiebrink1/Wekinator*. HTML. <https://github.com/fiebrink1/wekinator>.
- “First Quantum Music Composition Unveiled.” 2015. MIT Technology Review. April 15, 2015.
<https://www.technologyreview.com/2015/04/15/168638/first-quantum-music-composition-unveiled/>.
- Fober, Dominique, Gabriel Lepetit-Aimon, Julien Aldon, Yann Orlarey, and Thibaud Keller. n.d.
 “INScore.” Accessed May 3, 2021. <http://inscore.sourceforge.net/>.
- Fourment, Mathieu, and Michael R Gillings. 2008. “A Comparison of Common Programming Languages Used in Bioinformatics.” *BMC Bioinformatics* 9 (February): 82. <https://doi.org/10.1186/1471-2105-9-82>.
- Froment, Nicolas, Werner Schweer, anatoly-os, Joachim Schmitz, Marc Sabatella, Dmitri Ovodok, Andrey M. Tokarev, et al. (2012) 2021. *Musescore/MuseScore*. C++. MuseScore.
<https://github.com/musescore/MuseScore>.
- Gabor, Dennis. 1947. “Acoustical Quanta and the Theory of Hearing.” *Nature* 159 (4044): 591–94.
<https://doi.org/10.1038/159591a0>.
- Galaxy Unpacked 2016*. 2016. <https://www.youtube.com/watch?v=Ulo12bZNx6k>.
- Gambetta, Jay, and Jerry Chow. 2016. “Quantum Computing: It’s Time to Build a Quantum Community.” IBM Research Blog. May 3, 2016.
<https://www.ibm.com/blogs/research/2016/05/quantum-computing-time-build-quantum-community/>.
- Gershenfeld, Neil A., and Isaac L. Chuang. 1997. “Bulk Spin-Resonance Quantum Computation.” *Science* 275 (5298): 350–56. <https://doi.org/10.1126/science.275.5298.350>.
- Gidney, Craig, Dave Bacon, Doug Strain, Balint Pato, Kevin J. Sung, Matthew Neeley, Adam Zalcman, et al. (2017) 2021. *Cirq*. Python. The Cirq Developers. <https://github.com/quantumlib/Cirq>.
- Gil, Dario. 2016. “The Dawn of Quantum Computing Is Upon Us.” THINK Blog. May 4, 2016.
<https://www.ibm.com/blogs/think/2016/05/the-quantum-age-of-computing-is-here/>.
- Gilmore, Robert. 1995. *Alice in Quantumland: An Allegory of Quantum Physics*. New York: Copernicus.
- Gimzewski, James. 2020. “Workshop: Music and Quantum Mechanics.” In Kepler’s Gardens. September 11, 2020. <https://ars.electronica.art/keplersgardens/en/workshop-music/>.
- Glassner, Andrew. 2002. “Quantum Computing.” In *Andrew Glassner’s Other Notebook*, 1st ed. CRC Press. <http://www.crcnetbase.com/isbn/9781482279146>.
- Goldwire, Betsylynn Dunn. 1984. “Harmonic Evolution in the Piano Poems of Alexander Scriabin.” Thesis DMA--University of Texas at Austin.
- Gondry, Michel. 2004. *Eternal Sunshine of the Spotless Mind*. Drama, Romance, Sci-Fi. Focus Features, Anonymous Content, This Is That Productions.
- Grosse, Darwin. 2019. “Node For Max Intro – Let’s Get Started!” January 8, 2019.
<https://cycling74.com/articles/node-for-max-intro-%E2%80%93-let%E2%80%99s-get-started>.
- Grover, Lov K. 1996. “A Fast Quantum Mechanical Algorithm for Database Search.” In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, 212–19. STOC ’96. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/237814.237866>.
- “Hackathon | Quantum Information for Developers.” n.d. Accessed March 26, 2021.
<https://qid.ethz.ch/hackathon/>.
- Hajdu, Georg. n.d. “Quintet.Net.” Accessed May 3, 2021. <http://georghajdu.de/6-2/quintet-net/>.

- Hamido, Omar Costa. 2013. “<|> Estudo sobre as relações entre Música e Pintura e processos composicionais.” Porto, Portugal: ESMAE-IPP. http://recipp.ipp.pt/bitstream/10400.22/8868/1/DM_OmarHamido_2013.pdf.
- . 2015. “Miguel Zenón’s Music Is like a Spinning Electron!” Medium. December 13, 2015. <https://medium.com/@omarcostahamido/miguel-zen%C3%B3n-s-music-is-like-a-spinning-electron-6dfca1ce8bc2>.
- . 2018. *Och.Scorestream* (version 0.8.1). Max. <https://www.maxforlive.com/library/device/5120/och-scorestream>.
- . 2020a. *The Space Between the Strings*. Short, Music. <https://www.imdb.com/title/tt13575414/>.
- . 2020b. *KeyFinder for Live - Key Detection for Ableton*. Max for Live. Beat Lab Academy. <https://beatlabacademy.com/keyfinder-for-live/>.
- . 2021a. “I’m a Musician, and Here’s Why I’m Learning Quantum Computing.” Medium Qiskit. January 27, 2021. <https://medium.com/qiskit/im-a-musician-and-here-s-why-i-m-learning-quantum-computing-db28c2aba7ac>.
- . 2021b. *The Gedanken Room*. Music. <https://www.imdb.com/title/tt14443510/>.
- Hamido, Omar Costa, Giovanni Amedeo Cirillo, and Edoardo Giusto. 2020. “Quantum Synth: A Quantum-Computing-Based Synthesizer.” In *Proceedings of the 15th International Conference on Audio Mostly*, 265–68. AM ’20. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3411109.3411135>.
- Hamido, Omar Costa, Michael Dessen, Guillaume Pellerin, and Jules Penuchot. (2020) 2021. *Awesome Networked Media*. TeX. <https://github.com/omarcostahamido/awesome-networked-media>.
- Hamido, Omar Costa, and Andrea Sloan Pink. n.d. “Duet for Human and Machine.” Accessed May 3, 2021. <https://sites.google.com/uci.edu/duetforhumanandmachine/home>.
- Hamido, Omar Costa, Jack J. Woehr, and Jielun (Chris) Chen. (2020) 2020. *Qc.VFX*. Python. <https://github.com/omarcostahamido/qc.VFX>.
- Hamido, Omar Costa, and James Wootton. 2021. *MicroQiskit C++*. C++. <https://github.com/qiskit-community/MicroQiskit/tree/master/versions/C%2B%2B>.
- Heaney, Libby. 2019. “Quantum Computing and Complexity in Art.” *Leonardo* 52 (3): 230–35. https://doi.org/10.1162/leon_a_01572.
- . 2020. *Quantum Computing: Art And Entanglements*. Berlin Science Week. <https://www.youtube.com/watch?v=V1dN0d7D5JQ>.
- Helweg, Kim. 2018. “Composing with Quantum Information: Aspects of Quantum Music in Theory and Practice.” *Muzikologija*, no. 24: 61–77.
- Herbert, Nick. 1985. *Quantum Reality: Beyond the New Physics*. Garden City, N.Y: Anchor Press/Doubleday.
- . 2014. “QUANTUM TANTRA: Bell’s Theorem Blues.” *QUANTUM TANTRA* (blog). November 7, 2014. <http://quantumtantra.blogspot.com/2014/11/bells-theorem-blues.html>.
- Holloway, Christopher L., Matthew T. Simons, Abdulaziz H. Haddab, Carl J. Williams, and Maxwell W. Holloway. 2019. “A ‘Real-Time’ Guitar Recording Using Rydberg Atoms and Electromagnetically Induced Transparency: Quantum Physics Meets Music.” *AIP Advances* 9 (6): 065110. <https://doi.org/10.1063/1.5099036>.
- Huang, Junye. (2019) 2021. *Awesome-Quantum-Games*. <https://github.com/HuangJunye/Awesome-Quantum-Games>.
- Huang, Junye, Omar Costa Hamido, and Aurélien Da Campo. (2019) 2020. *Wolfiverse*. HTML. <https://github.com/HuangJunye/Wolfiverse>.
- Huffman, Russell. 2021. “Quantum Computer Meets Pen Plotter.” Medium. February 10, 2021. <https://medium.com/qiskit/quantum-computer-meets-pen-plotter-21c0f0c136d2>.
- IBM Corp. 2019a. “Qiskit Camp Europe.” September 2019. <https://qiskit.org/events/europe/>.
- . 2019b. “Qiskit Camp Africa.” December 2019. <https://qiskit.org/events/africa/>.
- . n.d. “Qiskit Camp.” Accessed March 26, 2021. <https://qiskit.org/events/camp-2020/>.

- “IBM Quantum Experience.” n.d. IBM Quantum Experience. Accessed February 22, 2021.
<https://quantum-computing.ibm.com/>.
- Iliyasu, A. M. 2019. “Roadmap to Talking Quantum Movies: A Contingent Inquiry.” *IEEE Access* 7: 23864–913. <https://doi.org/10.1109/ACCESS.2018.2871107>.
- “IQuHACK 2021.” n.d. [www.IQuISE.Mit.Edu](http://www.iquise.mit.edu). Accessed March 26, 2021.
<https://www.iquise.mit.edu/iQuHACK/>.
- “Ircam Forum Workshops - Paris 2019.” n.d. Accessed May 3, 2021.
<https://forum.ircam.fr/collections/detail/les-ateliers-du-forum-paris-2019/>.
- Ismael, Jenann T. 2016. *How Physics Makes Us Free. How Physics Makes Us Free*. Oxford University Press.
<https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780190269449.001.0001/acprof-9780190269449>.
- Jain, Siddhi B., Sanjana G. Sonar, Sakshi S. Jain, Priyanka Daga, and Ritik S. Jain. 2020. “Review on Comparison of Different Programming Language by Observing It’s Advantages and Disadvantages.” *Research Journal of Engineering and Technology* 11 (3): 133–37.
<http://dx.doi.org/10.5958/2321-581X.2020.00023.9>.
- Javadi, Ali. 2020. “The Quantum Circuit Political Spectrum.” Twitter. April 15, 2020.
<https://twitter.com/jaygambetta/status/1250557658094022656>.
- Jay Gambetta, Matthew Treinish, Paul Nation, Paul Kassebaum, Diego M. Rodríguez, Salvador de la Puente González, qiskit-bot, et al. 2021. *Qiskit/Qiskit: Qiskit 0.23.6*. Zenodo.
<https://doi.org/10.5281/zenodo.4549740>.
- Jiang, Nan, Jian Wang, and Yue Mu. 2015. “Quantum Image Scaling up Based on Nearest-Neighbor Interpolation with Integer Scaling Ratio.” *Quantum Information Processing* 14 (11): 4001–26.
<https://doi.org/10.1007/s1128-015-1099-5>.
- Jiang, Xue, Chengzhi Shi, Zhenglu Li, Siqi Wang, Yuan Wang, Sui Yang, Steven G. Louie, and Xiang Zhang. 2020. “Direct Observation of Klein Tunneling in Phononic Crystals.” *Science* 370 (6523): 1447–50. <https://doi.org/10.1126/science.abe2011>.
- Johnston, Eric R., Nic Harrigan, and Mercedes Gimeno-Segovia. 2019. *Programming Quantum Computers: Essential Algorithms and Code Samples*. Sebastopol: O’Reilly Media, Incorporated.
- Kandinsky, Wassily. 1946. *On the Spiritual in Art : First Complete English Translation, with Four Full Colour Page Reproductions, Woodcuts and Half Tones*. Edited by Hilla Rebay. New York : Solomon R. Guggenheim Foundation, for the Museum of Non-Objective Painting.
<http://archive.org/details/onspiritualinart00kand>.
- . 1982. *Kandinsky, Complete Writings on Art*. The Documents of Twentieth Century Art. Boston, Mass: GKHall.
- Karalekas, Peter, Matthew Harrigan, Mark Skilbeck, Will Zeng, Nikolas Tezak, Eric Peterson, Lauren Capelluto, et al. (2017) 2021. *PyQuil: Quantum Programming in Python*. Python. Rigetti Computing. <https://github.com/rigetti/pyquil>.
- “KIM-1.” 2020. In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=KIM-1&oldid=953885191>.
- Kirke, Alexis. 2018. “Programming Gate-Based Hardware Quantum Computers for Music.” *Muzikologija*, no. 24: 21–37.
- . 2020. “Testing a Hybrid Hardware Quantum Multi-Agent System Architecture That Utilizes the Quantum Speed Advantage for Interactive Computer Music.” *Journal of New Music Research* 49 (3): 209–30. <https://doi.org/10.1080/09298215.2020.1749672>.
- Kleinermanns, Ralf. 2016. *Team Supreme: Strength in Numbers*.
<https://www.youtube.com/watch?v=10ZEtTb2qQw>.
- Koch, Daniel, Laura Wessing, and Paul M. Alsing. 2019. “Introduction to Coding Quantum Algorithms: A Tutorial Series Using Qiskit.” *ArXiv:1903.04359 [Quant-Ph]*, March.
<http://arxiv.org/abs/1903.04359>.

- Kotler, Shlomi, Gabriel A. Peterson, Ezad Shojaee, Florent Lecocq, Katarina Cicak, Alex Kwiatkowski, Shawn Geller, et al. 2021. “Direct Observation of Deterministic Macroscopic Entanglement.” *Science* 372 (6542): 622–25. <https://doi.org/10.1126/science.abf2998>.
- Kuchera-Morin, JoAnn, Lance Putnam, Luca Peliti, Dennis Adderton, Andres Cabrera, Kon Hyong Kim, Gustavo Rincon, et al. 2017. “PROBABLY/POSSIBLY? An Immersive Interactive Visual/Sonic Quantum Composition and Synthesizer.” In *Proceedings of the 25th ACM International Conference on Multimedia*, 559–61. MM ’17. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3123266.3129324>.
- Kultima, Annakaisa. n.d. “Quantum Games Course – At Aalto University.” Accessed March 1, 2021. <https://quantumgames.aalto.fi/>.
- Lanzagorta, Marco, and Jeffrey Uhlmann. 2005a. “Quantum Rendering: An Introduction to Quantum Computing, Quantum Algorithms and Their Applications to Computer Graphics.” In *ACM SIGGRAPH 2005 Courses*, 1-es. SIGGRAPH ’05. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1198555.1198722>.
- Lanzagorta, Marco, and Jeffrey K. Uhlmann. 2005b. “Hybrid Quantum-Classical Computing with Applications to Computer Graphics.” In *ACM SIGGRAPH 2005 Courses*, 2-es. SIGGRAPH ’05. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/1198555.1198723>.
- Latorre, Jose I. 2005. “Image Compression and Entanglement.” *ArXiv:Quant-Ph/0510031*, October. <http://arxiv.org/abs/quant-ph/0510031>.
- Le, Phuc Q., Fangyan Dong, and Kaoru Hirota. 2011. “A Flexible Representation of Quantum Images for Polynomial Preparation, Image Compression, and Processing Operations.” *Quantum Information Processing* 10 (1): 63–84. <https://doi.org/10.1007/s11128-010-0177-y>.
- Leicester. 2018. “REVIEW | Gesu No Kiwami Otome Game On With ‘Tatakatte Shimau Yo.’” *OH! Press* (blog). June 2, 2018. <https://blog.onehallyu.com/single-review-gesu-no-kiwami-otome-tatakatte-shimau-yo/>.
- Leikin, Anatole. 2011. *The Performing Style of Alexander Scriabin*. Farnham ; Ashgate Pub.
- Lewis, George E. 2000. “Too Many Notes: Computers, Complexity and Culture in Voyager.” *Leonardo Music Journal* 10 (1): 33–39.
- Li, Hai-Sheng, Zhu Qingxin, Song Lan, Chen-Yi Shen, Rigui Zhou, and Jia Mo. 2013. “Image Storage, Retrieval, Compression and Segmentation in a Quantum System.” *Quantum Information Processing* 12 (6): 2269–90. <https://doi.org/10.1007/s11128-012-0521-5>.
- Li, Hai-Sheng, Qingxin Zhu, Ri-Gui Zhou, Lan Song, and Xing-jiang Yang. 2014. “Multi-Dimensional Color Image Storage and Retrieval for a Normal Arbitrary Quantum Superposition State.” *Quantum Information Processing* 13 (4): 991–1011. <https://doi.org/10.1007/s11128-013-0705-7>.
- Li, Panchi, Bing Wang, Hong Xiao, and Xiande Liu. 2018. “Quantum Representation and Basic Operations of Digital Signals.” *International Journal of Theoretical Physics* 57 (10): 3242–70. <https://doi.org/10.1007/s10773-018-3841-0>.
- Ligeti, Lukas. 2021. “Artistic Innovation through African Concepts: Education for Art-Music Composers Based on African Traditions.” *Journal of the Musical Arts in Africa*.
- Lioret, Alain. 2014. “Quantum Computing for Art Exploration and Creation.” In *ACM SIGGRAPH 2014 Studio*, 1. SIGGRAPH ’14. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/2619195.2656283>.
- . 2016. *Quantum Art*. The Eurographics Association. <https://doi.org/10.2312/exp.20161072>.
- “List of Quantum Processors.” 2019. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=List_of_quantum_processors&oldid=880435754.
- Lončar, Sonja, and Andrija Pavlović. 2018. “Hybrid Duo.” *Muzikologija*, no. 24: 111–21.
- Lough, Alex. 2020. *Alex Lough: Assemblage #4 (Flicker)*. Short, Music. <https://www.imdb.com/title/tt14577748/>.
- Magenta/Magenta-Studio*. (2018) 2020. Max. Magenta. <https://github.com/magenta/magenta-studio>.

- Malone, Thomas W., Robert Laubacher, and Tammy Johns. 2011. "The Big Idea: The Age of Hyperspecialization." *Harvard Business Review*, July 1, 2011. <https://hbr.org/2011/07/the-big-idea-the-age-of-hyperspecialization>.
- Marletto, Chiara. 2018. "Musings on Quantum Music: Can Quantum Music Bring Us Closer to Objective Beauty?" *Muzikologija*, no. 24: 123–28.
- "Max for Live | Ableton." n.d. Accessed February 23, 2021. <https://www.ableton.com/en/live/max-for-live/>.
- "Max Mathews." 2019. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Max_Mathews&oldid=917912570.
- "Max (Software)." 2020. In *Wikipedia*. [https://en.wikipedia.org/w/index.php?title=Max_\(software\)&oldid=944421166](https://en.wikipedia.org/w/index.php?title=Max_(software)&oldid=944421166).
- Max (version 8.1.10). 2021. Cycling '74. <https://cycling74.com/>.
- McCormick, Rich. 2016. "This Image of Mark Zuckerberg Says so Much about Our Future." *The Verge*. February 22, 2016. <https://www.theverge.com/2016/2/22/11087890/mark-zuckerberg-mwc-picture-future-samsung>.
- Medić, Ivana, and Jelena Janković-Beguš. 2018. "What Does Quantum Music Sound like and What Would Pierre Boulez Think of It? Super Position (Many Worlds) by Kim Helweg (2017)." *Muzikologija*, no. 24: 79–93.
- Miranda, Eduardo R. 2020a. "Creative Quantum Computing: Inverse FFT, Sound Synthesis, Adaptive Sequencing and Musical Composition." *ArXiv:2005.05832 [Cs, Eess]*, May. <http://arxiv.org/abs/2005.05832>.
- . 2020b. "Quantum Computer: Hello, Music!" *ArXiv:2006.13849 [Quant-Ph]*, June. <http://arxiv.org/abs/2006.13849>.
- Moran, Christine Corbett. 2016a. "5-Qubit Quantum Computing Simulator." *CodeXgalactic* (blog). May 21, 2016. <https://codexgalactic.com/2016/05/21/5-qubit-quantum-computing-simulator/>.
- . 2016b. "Quintuple: A Python 5-Qubit Quantum Computer Simulator to Facilitate Cloud Quantum Computing." *ArXiv:1606.09225 [Quant-Ph]*, June. <http://arxiv.org/abs/1606.09225>.
- . 2019. *Mastering Quantum Computing with IBM QX: Explore the World of Quantum Computing Using the Quantum Composer and Qiskit*. Birmingham: Packt Publishing Ltd.
- "MSP: Introduction." n.d. Accessed May 19, 2021. https://docs.cycling74.com/max7/tutorials/01_mspintro.
- "MuseScore." n.d. Accessed February 22, 2021. <https://musescore.org/en>.
- Nolan, Christopher. 2010. *Inception*. Action, Adventure, Sci-Fi, Thriller. Warner Bros., Legendary Entertainment, Syncopy. <https://www.imdb.com/title/tt1375666/>.
- Novković, Dragan, Marko Peljević, and Mateja Malinović. 2018. "Synthesis and Analysis of Sounds Developed from the Bose-Einstein Condensate: Theory and Experimental Results." *Muzikologija*, no. 24: 95–109.
- Oliveros, Pauline. 2019. "Quantum Avant Garde." In *The Idea of the Avant Garde. Volume 2, and What It Means Today*, edited by Marc James Léger, 68. Bristol: Intellect Ltd.
- Ömer, Bernhard. 2014. "QCL - A Programming Language for Quantum Computers." 2014. <http://tph.tuwien.ac.at/~oemer/qcl.html>.
- "Open Call - Beyond Quantum Music." 2020. *Beyond Quantum Music* (blog). May 24, 2020. <http://quantummusic.org/open-call/>.
- "OpenFrameworks." n.d. Accessed February 23, 2021. <https://openframeworks.cc/>.
- Open-Source Quantum Software Projects*. (2018) 2021. Quantum Open Source Foundation. <https://github.com/qosf/awesome-quantum-software>.
- "Opusmodus." n.d. Accessed February 24, 2021. <https://opusmodus.com/>.
- Oshiro, Scott, and Omar Costa Hamido. 2020. "A Quantum-Classical Network for Beat-Making Performance." *Journal of Network Music and Arts* 2 (1). <https://commons.library.stonybrook.edu/jonma/vol2/iss1/4>.

- Outwater, Edwin. 2012. “Quantum: Music On The Frontier Of Science.” February 2012. <http://www.edwinoutwater.com/projects/quantum-music-on-the-frontier-of-science-with-the-iqc-computing/>.
- Prechelt, L. 2000. “An Empirical Comparison of Seven Programming Languages.” *Computer* 33 (10): 23–29. <https://doi.org/10.1109/2.876288>.
- Preskill, John. 2018. “Quantum Computing in the NISQ Era and Beyond,” January. <https://doi.org/10.22331/q-2018-08-06-79>.
- Puckette, Miller. 2017. “Four Surprises of Electronic Music.” *Lumina* 11 (2): 126–38. <https://doi.org/10.34019/1981-4070.2017.v11.21445>.
- . n.d. “Pure Data (Pd): Real-Time Music and Multimedia Environment.” Accessed February 22, 2021. <http://msp.ucsd.edu/software.html>.
- Putz, Volkmar, and Karl Svozil. 2017. “Quantum Music.” *Soft Computing* 21 (6): 1467–71. <https://doi.org/10.1007/s00500-015-1835-x>.
- “PWGL.” n.d. Accessed January 2, 2019. <http://www2.siba.fi/PWGL/>.
- “QHack.” n.d. Accessed March 26, 2021. <https://qhack.ai>.
- Qiskit. n.d. “Qiskit Advocates.” Accessed March 8, 2021. <https://qiskit.org/advocates>.
- “Quadrivium.” 2020. In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=Quadrivium&oldid=942556617>.
- “Quantum Computing.” 2000. MIT Technology Review. May 1, 2000. <https://www.technologyreview.com/2000/05/01/236302/quantum-computing/>.
- “Quantum Computing.” 2020. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Quantum_computing&oldid=956061878.
- “Quantum Computing Is Now Available on AWS through Amazon Braket.” 2020. Amazon Web Services, Inc. August 13, 2020. <https://aws.amazon.com/about-aws/whats-new/2020/08/quantum-computing-available-aws-through-amazon-braket/>.
- “Quantum Futures Hackathon.” n.d. Indico. Accessed March 26, 2021. <https://indico.cern.ch/event/838035/>.
- “Quantum in the Cloud.” 2013. 2013: Quantum in the Cloud | News and Features | University of Bristol. University of Bristol. September 6, 2013. <https://www.bristol.ac.uk/news/2013/9720.html>.
- “Quantum Programming Studio.” 2018. 2018. <https://quantum-circuit.com/>.
- Ramis, Harold. 1993. *Groundhog Day*. Comedy, Fantasy, Romance. Columbia Pictures. <https://www.imdb.com/title/tt0107048/>.
- Ronald. 2018. “Gesu No Kiwami Otome. to Release Their First Single in Over 2 Years.” ARAMA! JAPAN. January 12, 2018. <https://aramajapan.com/news/newrelease/gesu-no-kiwami-otome-release-first-single-2-years/84089/>.
- Rubio, Jesús, Daniel Bevenius, Omar Costa Hamido, Jorge Carballo, Abdón Rodríguez Davila, Juan Cruz-Benito, and Wei Hu. (2017) 2019. *Qiskit.Js* (version v0.9.0). JavaScript. Qiskit Community. <https://github.com/qiskit-community/qiskit-js>.
- Simonite, Tom. 2017. “The Quantum Computer Factory That’s Taking on Google and IBM.” *Wired*, June 20, 2017. <https://www.wired.com/story/quantum-computing-factory-taking-on-google-ibm/>.
- Stein, Norbert, Christoph Haberer, Nikolao Valiensi, and Michael Heupel. 2010. *Silent Sitting Bulls*. CD. Pata Music. <https://www.discogs.com/Norbert-Stein-Pata-Horns-Drums-Silent-Sitting-Bulls/release/10610185>.
- Stone, Lynden. 2013. “Re-Visioning Reality: Quantum Superposition in Visual Art.” *Leonardo* 46 (5): 449–54.
- “Strangeworks | Quantum Computing Ecosystem.” 2021. Strangeworks. 2021. <https://strangeworks.com/>.
- Sun, Bo, Abdullah Iliyasa, Fei Yan, Fangyan Dong, and Kaoru Hirota. 2013. “An RGB Multi-Channel Representation for Images on Quantum Computers.” *Journal of Advanced Computational Intelligence and Intelligent Informatics* 17 (May): 404–17.
- “SuperCollider.” n.d. Accessed February 23, 2021. <https://supercollider.github.io/>.

- “Thought Experiment.” 2021. In *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=Thought_experiment&oldid=1016947735.
- “Timeline of Machine Learning.” 2020. In *Wikipedia*.
https://en.wikipedia.org/w/index.php?title=Timeline_of_machine_learning&oldid=950506236.
- Tipei, Sever. 1989. “The Computer: A Composer’s Collaborator.” *Leonardo* 22 (2): 189–95.
<https://doi.org/10.2307/1575227>.
- Turner, Frederick. n.d. “Quantum Theory and the Arts.” Accessed May 3, 2020.
<http://www.nccsc.net/essays/quantum-theory-and-arts>.
- Tymoczko, Dmitri. 2013. “Review of Michael Cuthbert, *Music21: A Toolkit for Computer-Aided Musicology* (http://Web.Mit.Edu/Music21/).” *Music Theory Online* 19 (3). <https://mtosmt.org/issues/mto.13.19.3/mto.13.19.3.tymoczko.html>.
- Venegas-Andraca, S. E., and J. L. Ball. 2010. “Processing Images in Entangled Quantum Systems.” *Quantum Information Processing* 9 (1): 1–11. <https://doi.org/10.1007/s11128-009-0123-z>.
- Venegas-Andraca, Salvador E., and Sougato Bose. 2003. “Storing, Processing, and Retrieving an Image Using Quantum Mechanics.” In *Quantum Information and Computation*, 5105:137–47. International Society for Optics and Photonics. <https://doi.org/10.1117/12.485960>.
- Vlasov, Alexander Yu. 1997. “Quantum Computations and Images Recognition.” *ArXiv:Quant-Ph/9703010*, March. <http://arxiv.org/abs/quant-ph/9703010>.
- Vogt-Lee, Desiree. (2018) 2021. *Awesome Quantum Computing*. <https://github.com/desireevl/awesome-quantum-computing>.
- Wang, Jian. 2016. “QRDA: Quantum Representation of Digital Audio.” *International Journal of Theoretical Physics* 55 (3): 1622–41. <https://doi.org/10.1007/s10773-015-2800-2>.
- Weaver, James. (2018) 2018. *Quantum Music Composer for IBM Quantum Computers*. JavaScript. <https://github.com/JavaFXpert/quantum-toy-piano-ibmq>.
- . 2018. “Jamming with a Quantum Computer.” Medium. August 22, 2018. <https://medium.com/rigetti/jamming-with-a-quantum-computer-bed05550a0e8>.
- Wehden, Karl, Ismael Faro, and Jay Gambetta. 2021. “IBM’s Roadmap for Building an Open Quantum Software Ecosystem.” IBM Research Blog. February 4, 2021. <https://www.ibm.com/blogs/research/2021/02/quantum-development-roadmap/>.
- Weimer, Hendrik. 2010. “Listen to Quantum Computer Music.” April 13, 2010. <http://www.quantenblog.net/physics/quantum-computer-music>.
- Weimer, Hendrik, Markus Müller, Igor Lesanovsky, Peter Zoller, and Hans Peter Büchler. 2010. “A Rydberg Quantum Simulator.” *Nature Physics* 6 (5): 382–88. <https://doi.org/10.1038/nphys1614>.
- “Wekinator | Software for Real-Time, Interactive Machine Learning.” n.d. Accessed May 12, 2020. <http://www.wekinator.org/>.
- “Welcome to Magenta!” n.d. Magenta. Accessed May 11, 2020. <https://magenta.tensorflow.org/blog/2016/06/01/welcome-to-magenta/>.
- Williams, Anthony. 2019. *C++ Concurrency in Action, Second Edition, Video Edition*. 1st edition. Manning Publications.
- Woehr, Jack, and Omar Costa Hamido. 2021. *QisJob*. Zenodo. <https://doi.org/10.5281/zenodo.4554481>.
- Wootton, James. 2019. *MicroQiskit for Python*. Python. <https://github.com/qiskit-community/MicroQiskit>.
- . (2020) 2021. *QuantumBlur*. Jupyter Notebook. Qiskit Community. <https://github.com/qiskit-community/QuantumBlur>.
- Wootton, James R. 2020a. “Procedural Generation Using Quantum Computation,” July. <https://doi.org/10.1145/3402942.3409600>.
- . 2020b. “A Quantum Procedure for Map Generation.” *2020 IEEE Conference on Games (CoG)*, August, 73–80. <https://doi.org/10.1109/CoG47356.2020.9231571>.
- Yan, Fei, Yiming Guo, Abdullah M. Ilyasu, and Huamin Yang. 2017. “Flexible Representation and Manipulation of Audio Signals on Quantum Computers.” *ArXiv:1701.01291 [Quant-Ph]*, January. <http://arxiv.org/abs/1701.01291>.

- Yan, Fei, Abdullah M. Ilyasu, and Salvador E. Venegas-Andraca. 2016. "A Survey of Quantum Image Representations." *Quantum Information Processing* 15 (1): 1–35. <https://doi.org/10.1007/s11128-015-1195-6>.
- "YQI Artist: Spencer Topel." 2019. 2019. <http://art.quantuminstitute.yale.edu/Pages/Spencer.html>.
- Zhang, Yi, Kai Lu, Yinghui Gao, and Mo Wang. 2013. "NEQR: A Novel Enhanced Quantum Representation of Digital Images." *Quantum Information Processing* 12 (8): 2833–60. <https://doi.org/10.1007/s11128-013-0567-z>.
- Zhang, Yi, Kai Lu, Yinghui Gao, and Kai Xu. 2013. "A Novel Quantum Representation for Log-Polar Images." *Quantum Information Processing* 12 (9): 3103–26. <https://doi.org/10.1007/s11128-013-0587-8>.
- Zicarelli, David. 1998. "An Extensible Real-Time Signal Processing Environment for Max." In *ICMC*. <http://hdl.handle.net/2027/spo.bbp2372.1998.274>.

APPENDIX B: qc.VFX's quantumblur tox

The screenshot displays a Pure Data patch titled "EXTERNAL TOX" for "qc.VFX's quantumblur tox". The patch is a complex network of objects including:

- Math and Signal Processing:** `math1`, `top1o1`, `in1`, `script2`, `script2_callbaoks`, `moviefilestrip`, `in2`, `null1`, `null2`, `blurconstant`, `chopexec1`, `chopexec2`.
- Video and Image Processing:** `src`, `rgbhs1`, `script1`, `out1`.
- Text and File Handling:** `text_requirements`, `loadbang`, `text_template`, `text_paths`, `eval1`.
- Control and Utility:** `iparexpython`, `script_quantumblur`, `storing`, `me_parec`, `me_parec`, `me_parec`.

A code editor window titled "cooking_blur" shows the following Python script:

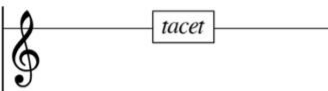
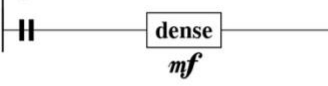

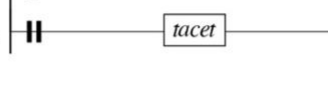

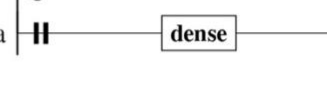
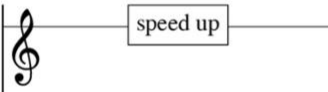
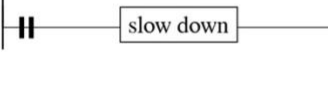
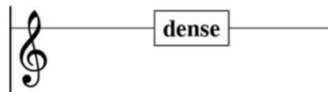

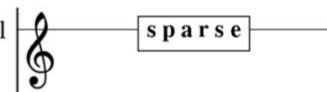
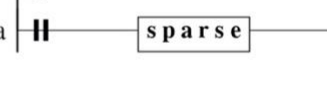
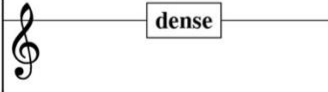
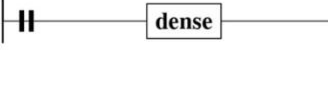
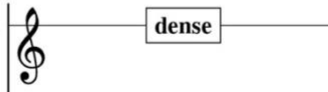
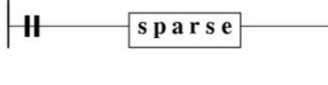
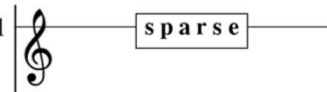
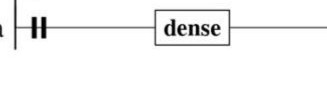
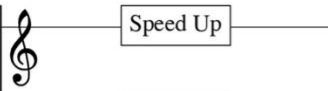
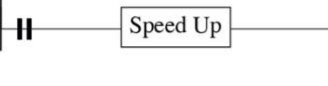
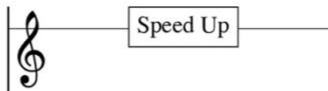
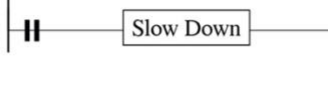
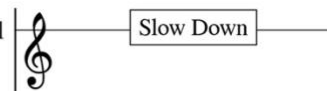
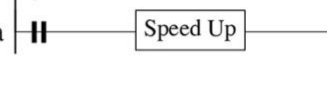
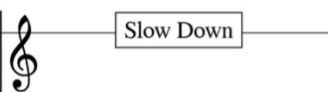
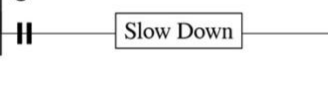
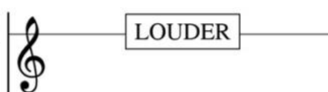

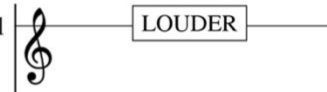
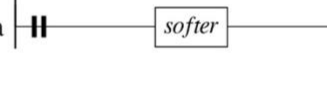
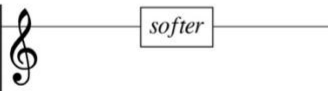
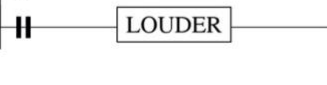
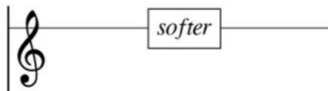
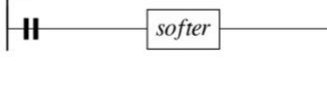


```

1 def onCook(scriptOp):
2     qc = me_fetch(qc)
3     if qc is None:
4         print("everyt
5         print("blurre
6         scriptOp.copy/
7
8
9
10
11
12
13
14

```

The interface includes a menu bar (File, Edit, Dialogs, Help), a toolbar with various icons, and a status bar at the bottom showing: Start: 1, End: 600, RStart: 1, REnd: 600, FPS: 60.0, Tempo: 120.0, Reset: 1. The time display shows 00:00:06.51 and 412. The range limit is set to Loop.

APPENDIX D: Slides for *Telematic Entanglements*

<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>
<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>
<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>
<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>
<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>
<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>	<p>Will </p> <p>Tina </p>

Will		Will		Will	
Tina		Tina		Tina	

Will		Will		Will	
Tina		Tina		Tina	

Will		Will		Will	
Tina		Tina		Tina	

Will		Will		Will	
Tina		Tina		Tina	

Will		Will		Will	
Tina		Tina		Tina	

APPENDIX E: Score for *Second Cornerstone*

Second cornerstone of QM

OCH

♩ = 60 A B

Kamanche x4

Music stand 1 x4

Music stand 2

Music stand 3

Music stand 4

setup your instrument and make sure every noise is intentional in itself...

medium hard yarn mallets

cresc...

pizz normal playing

mallet-strike mallet-strike

4 x11

Kmch.

MS1 x11

MS2

MS3

MS4

normal playing pizz *normal playing* underbridge pizz

#1 #6 #9: ***ff*** ***mf*** *last time only*

#2 #7 #8 #10: ***f*** ***mp***

#3 #4 #11: ***mp*** ***pp***

#5: ***mf*** ***f***

mallet-strike mallet-strike mallet-strike

every odd repetition

6 **C** CUE Reverb FX (fade in)

Kmch. *mp col legno* *mf*

MS1 *mp* *mf*

MS2 *mp* *mf*

MS3 *mp* *mf*

MS4 *mf*

10 CUE Reverb FX (fade out)

Kmch. *col legno* *mp*

MS1 *col legno* *mp*

MS2 *mp*

MS3 *mp*

MS4 *mp*

14 **D**

Kmch. *x4* *x8*

MS1 *x4* *x8*

MS2 *x4* *x8*

MS3 *x4* *x8*

MS4 *x8*

16

Kmch. *f* rubbing skin

MS1 *f*

MS2

MS3 *f* mallet handle on stand

MS4 *f* mallet handle on stand

18

Kmch. **E** normal playing extreme overpressure

MS1 *sfz* tilt

MS2 *sfz*

MS3

MS4 tilt

27

Kmch. **F** bowing underbridge extreme overpressure bowing underbridge

MS1 *ff* *mf*

MS2 tilt

MS3 *ff* tilt

MS4

G CUE Panning FX

36

Kmch. LH: strumming with nails
RH: tuning box pizz
RH: circular bowing
col legno

MS1

MS2 rotate

MS3

MS4

40

Kmch. circular bowing
circular bowing under and over bridge
any technique combination
last time pp

MS1

MS2 rotate

MS3

MS4 rotate

last time pp

x5

x5

H CUE Ringmod FX

46

Kmch. glissando up
glissando down

MS1 bend & mallet strike
 mf

MS2 bend & mallet strike

MS3 bend & mallet strike

MS4

x4

x5

x4 mf

x5

x6

APPENDIX F: Poster for *The Gedanken Room*

