Thinking Too Classically: Research Topics in Human-Quantum Computer Interaction

Zahra Ashktorab
IBM Research AI
Yorktown Heights, NY, USA
zahra.ashktorab1@ibm.com

Justin D. Weisz
IBM Research AI
Yorktown Heights, NY, USA
jweisz@us.ibm.com

Maryam Ashoori
IBM Research AI
Yorktown Heights, NY, USA
maryam@us.ibm.com

ABSTRACT

Quantum computing is a fundamentally different way of performing computation than classical computing. Many problems that are considered hard for classical computers may have efficient solutions using quantum computers. Recently, technology companies including IBM, Microsoft, and Google have invested in developing both quantum computing hardware and software to explore the potential of quantum computing. Because of the radical shift in computing paradigms that quantum represents, we see an opportunity to study the unique needs people have when interacting with quantum systems, what we call Quantum HCI (QHCI). Based on interviews with experts in quantum computing, we identify four areas in which HCI researchers can contribute to the field of quantum computing. These areas include understanding current and future quantum users, tools for programming and debugging quantum algorithms, visualizations of quantum states, and educational materials to train the first generation of "quantum native" programmers.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; • **Computer systems organization →** *Quantum computing*;

KEYWORDS

Human-quantum computer interaction; QHCI; Quantum computing education

ACM Reference Format:
Zahra Ashktorab, Justin D. Weisz, and Maryam Ashoori. 2019. Thinking Too Classically: Research Topics in Human-Quantum Computer Interaction. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland Uk.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3290605.3300486

CHI 2019, May 4–9, 2019, Glasgow, Scotland Uk

ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300486

1 INTRODUCTION

Quantum computing [43], once considered purely a theoretical domain, is rapidly becoming a reality with the availability of publicly-accessible quantum computing systems, advanced software simulators, and domain-specific programming languages and SDKs. Quantum computing has the potential to solve problems considered intractable for classical computers. For example, scientists have used quantum computers to determine the ground-state energy for beryllium hydride (BeH_2), a complex inorganic molecule [27]. Performing these kinds of simulations has a tremendous computational cost – often intractable for classical computers – but the results of these simulations may lead to the discovery of new drugs and materials and bolster clean energy efforts [23].

Quantum computing has potential to tackle complex problems, such as those with exponential solution spaces, more efficiently than classical computers because it is theorized that the parallelism of quantum mechanics can solve NP-complete problems [5, 12]. Copious amounts of research demonstrate that the run time for problems considered difficult for classical computing can be significantly improved when simulated with quantum computing [11, 54, 55]. Fields that may benefit include materials sciences and chemistry, logistics, finance, and cryptography, but the true potential of quantum remains to be seen.

There are a number of institutions making significant investments in quantum computing technology across all levels of the technology stack. These range from low-level technologies for implementing qubits and their interconnections to high-level languages and SDKs for simulating quantum systems and programming quantum algorithms. Companies such as IBM, Microsoft, and Google have all released tools, SDKs, and educational materials for people to experiment with. IBM even made several of their actual quantum computers publicly available for anyone to program [37]. Startups like Rigetti Computing and D-Wave are developing new quantum computing hardware, and 1Qubit and QxBranch are focused on developing solutions for industries such as finance and insurance. Universities are also heavily involved in quantum computing by making learning materials available online [38] and training students in quantum information science. Even governments are getting involved in incubating quantum computing technologies, evidenced by the introduction of the National Quantum Initiative Act [56] in the United States, the funding of the Institute for Quantum Computing by the Canadian government [46], and Prime Minister Trudeau's viral explanation of quantum computing [42].

Because of the rapid onset of quantum computing technology, our aim is to motivate the HCI community to consider the study of a wide range of topics in quantum computing and how users of quantum computers interact with the technology. Based on our own observations and interviews with experts in the field, we identify a number of areas in which HCI researchers can have an impact and define a new practice of Quantum HCI:

- Understanding current and future users of quantum computers, including the problems they are trying to solve, the domains in which they work, and their workflows to identify gaps in their needs,
- Developing and evaluating educational materials and designing learning communities to train the first generation of "quantum native" programmers, as well as ease the learning curve for existing developers,
- Designing new tools for programming and debugging quantum computers, including creating and evaluating programming languages and SDKs for interacting with quantum computers and simulators, and
- Creating new visualizations of quantum state, including exploring the effectiveness of multiple visual representations on learning, and representing the extraordinarily high dimensionality of multi-qubit states.

2 A BRIEF PRIMER ON QUANTUM COMPUTING

In classical computing, information is encoded with bits that are in a state of either 0 or 1. In quantum computing, information is encoded with qubits, and as with classical bits, qubits may be observed in a classical state of 0 or 1. Qubits may also exist in a state of superposition, in which their true state is unknown until it is measured. In superposition, a qubit has a probability p of being observed as a 0 and $1 - p$ of being observed as a 1. Qubits may be entangled with each other, such that measuring the state of one qubit tells you the state of the other without measuring it. Similar to how classical bits can be manipulated with logic gates, qubits are manipulated using quantum gates. For example, applying the Hadamard (H) gate to a qubit in a classical state puts it into a state of superposition, and vice versa.

Quantum computing can be difficult to reason about and use for computation because information is represented by microscopic, isolated physical states that obey the laws of quantum mechanics. These laws can be hard to intuit as they cannot be directly observed in our daily lives, unlike the laws of classical physics (e.g. objects at rest remain at rest). For example, the no-cloning theorem states that arbitrary, unknown quantum states cannot be directly copied between qubits [68], as the measurement of a qubit destroys its quantum state.

Quantum states are also notoriously fragile to maintain. Physical qubits may experience errors when external noise or stray magnetic fields perturb their quantum state, and quantum states may decohere after very short timescales; as of this writing, quantum states can only be maintained for about 90 microseconds [25] before they decohere to the classical state of 0 (called the ground state). Quantum error correction seeks to overcome these challenges by simulating logical qubits using large numbers of physical qubits.

There are a number of tools available for programming quantum systems, both in simulation and with actual hardware. GUIs such as IBM's Composer [47] (part of the IBM Q Experience), Google's Quantum Computing Playground [16], and Quirk [14] enable people to program their own quantum circuits and visualize the results. Programming languages and SDKs such as IBM's Qiskit [49], Google's OpenFermion [3, 17], Rigetti's Forest [7], and Microsoft's LIQUi|> [34, 62] allow people to develop quantum algorithms and run them in simulation, or in some cases, on actual quantum hardware.

3 METHODOLOGY

Our interest in defining a new practice of Quantum HCI stemmed from our experiences in joining a team focused on building tools for quantum scientists. As HCI researchers new to quantum computing, we found ourselves spending a lot of time learning many new concepts and terminology, quite outside our comfort zone. From our own learning process, we identified significant gaps in the availability of engaging learning materials for complete beginners. During the course of developing such materials, we realized the tremendous impact that good HCI research and practice could have on the field, especially as it was a niche field just beginning to experience significant attention and investment [1].

In order to more rigorously formalize our observations on how HCI could impact quantum computing, we consulted with quantum information scientists, applied mathematicians, and theoretical and experimental physicists, all of

[1] We discuss some criticisms of the maturity and widespread appeal of quantum computing in Section 5.

whom currently work with quantum computers. We identified participants through a snowball method and conducted seven semi-structured interviews in which we asked them about their daily workflows with respect to quantum computers and their views on the utility of quantum computing in the future. In addition to these interviews, we consulted with other members of our quantum computing development team to understand their user research and future directions. These interviews and discussions, as well as our own observations and experiences, formed the basis of what we believe is a fruitful landscape for HCI researchers to explore. Table 1 describes the participants in our interviews and their background in quantum computing.

Table 1: Interview participants.

Participant	Institution	Background
P1	Industry	Experimental physicist, low temperatures
P2	Industry	Applied physicist, manager of quantum scientists
P3	University	PhD student, quantum computing
P4	Government	Theoretical quantum chemist
P5	University	Physicist, superconducting nanodevices
P6	Industry	Applied mathematician, optimization
P7	Industry	Theoretical physicist

Interview Topics

Our interviews were semi-structured and probed the following areas.

Education & Background. We asked participants about their education and background in quantum computing, as well as about what learning materials they found effective.

Applications & Algorithms. We discussed the kinds of problems participants used quantum computers to solve, how quantum computing differs from classical, and thoughts on the future of quantum computing.

Programming, Tools, & Workflow. Participants told us about which quantum computing programming languages, tools, and SDKs they used in their daily work, and how they collaborated and shared work with others.

Community. Some participants were highly active in online quantum computing communities and shared their experiences and ideas for how they would improve those communities.

Recruitment

We recruited a small batch of participants from our own organization through an email seeking individuals who conducted research in quantum computing. We then did snowball sampling by asking who they would recommend to speak with, both internal and external to our organization. We realize this recruitment strategy limited our observations on the diversity of quantum computing tools used. However, by interviewing individuals external to our organization, we aimed to capture opinions and skepticism we may not have seen inside our organization. Furthermore, at the current stage, most quantum users are experts and scientists in the field. Given that our agenda is to make quantum computing more user-friendly and accessible, additional studies are needed that observe and study the learning curve novices must overcome when learning about quantum.

4 AREAS OF INQUIRY IN QUANTUM HCI

From our discussions, we identified four main topic areas in which HCI researchers can make concrete, important contributions in order to actualize the field of Quantum HCI: understanding users and workflows, developing and evaluating educational materials, developing and evaluating tooling to aid programming and debugging of quantum systems, and designing new visualizations that can represent the ultra-high dimensionality of quantum systems.

In the rest of this paper, we discuss these areas and provide concrete examples of what the quantum community has done to support themselves in each area. We also highlight specific opportunities for HCI researchers to "move the needle" by conducting additional studies or building new kinds of tools. Furthermore, our interviewees shared observations on the future of quantum computing and its entry into the mainstream, which we discuss in Section 5.

Understanding Users & Workflows

Designing effective technologies for programming quantum systems requires an understanding of the different types of people who will use those systems. Our quantum development team had conducted early-stage user research and identified three groups of quantum users: quantum scientists, science enthusiasts, and developers. We describe each of these types below, but we note that this categorization does not represent a complete picture of the landscape of quantum computing learners, domain experts, and programmers (which we collectively refer to as "users"). As the popularity of the field grows, many new people are becoming involved, each with a different background, set of interests, and skills. Further research is needed to identify who these new users are and what their specific needs are in order to identify gaps in existing tooling, documentation, and learning materials.

Quantum Scientists. Quantum scientists include professors, postdocs, and graduate students in physics, quantum information science, computer science, and other related fields. They perform experiments using quantum simulators and on real quantum computers, such as IBM's quantum devices. Quantum scientists see value in being able to run experiments on actual hardware rather than just in simulation, and the availability of actual hardware is spurring interest in the field that has not previously been seen.

> *"I think we are currently in the gold rush of user experimentation. I'm getting phone calls from German companies to help them – companies that would not take my call two years ago." (P5, physicist)*

> *"None of the other quantum tools lets you run on an actual machine. No one else has opened up their machine to the public. No one else has the kind of support that I've seen [from IBM]." (P4, quantum chemist)*

We asked the quantum scientists what kinds of problems quantum computers are good for solving, and one scientist gave a clear, high-level description.

> *"A good problem for a quantum computer is a problem that can be succinctly described but can explode into an infinite number of possible solutions." (P2, physicist)*

We asked an experimental physicist to describe a specific problem quantum computers could solve that classical computers could not, and he described how quantum computers could be used to simulate quantum systems and calculate the energy states of molecules.

> *"One [problem] that we know will have an advantage is a simulation of quantum systems. If you take atoms and calculate possible energy states and bind them in molecules and try to calculate the equilibrium energy, the number of parameters in these molecules go up very quickly. When you try to simulate something like that in a computer you use numerical simulations and that could run forever. When you cut it you incur an error but for molecules like caffeine, it is complex and simulating it is nontrivial. So, if we can map the problem of complex molecules into the interactions that happen in which we have a chip we can run the problem there, and that will tell us about bonding length and reaction speed." (P1, experimental physicist)*

We also asked them what tools they use when conducting experiments. Many participants expressed a preference for programming in Python as they made heavy use of Jupyter notebooks [29] and Github to share code and results.

> *"Almost everything goes through Jupyter notebooks" (P2, physicist)*

> *"I put everything I was doing on Github so I could share the code when I write a paper." (P3, PhD student)*

One quantum scientist described her workflow, in which she performed some experiments in IBM's Composer UI (shown in Figure 1a) before switching to Qiskit and Juypter notebooks.

> *"I will start with the Composer and write out a simulation and see if it works... I'll use Qiskit if I want to run a bunch of things or if I want to make changes or massage the output... If I want to write a Jupyter notebook to explain what I've done then I will use Qiskit." (P4, quantum chemist)*

Other quantum scientists reported that graphical environments such as Composer were tedious to use and they preferred writing code in Python.

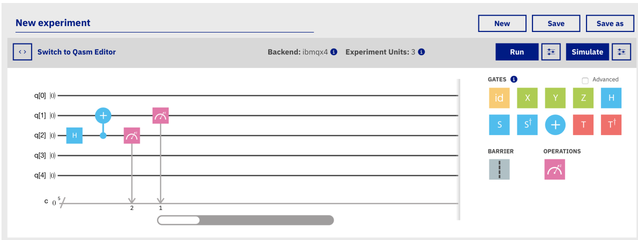
> *"The nice thing about Qiskit it gives you a free path to do whatever you like in terms of programming the computer... Every time you run a new thing you have to drag and drop in Composer. If you want to do some particular experiments that require looping, it's very tedious to do it in Composer. It's not what any programming environment looks like." (P1, experimental physicist)*

Science Enthusiasts. Science enthusiasts have an interest in quantum computing, but no formal background or training in it yet. These individuals include university students, academics, and industry professionals, all with varying levels of expertise in programming and quantum physics. Their primary needs are for educational materials and front-end tools that facilitate learning and easy interactions with quantum computers. We further discuss their needs in the next section.

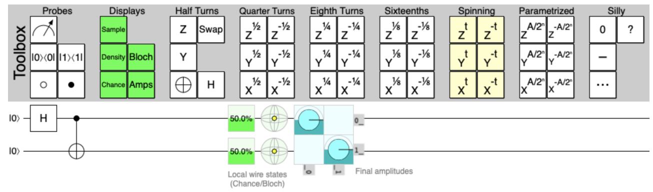
Developers. Although developers may have a deep understanding of classical computing, their knowledge of quantum computing is likely limited. For this audience, SDKs that provide high-level functionality while hiding lower-level technical details of the quantum implementation are desired.

> *"I think there is another layer of user that we will eventually connect with, which is: I want to use a quantum computer to accelerate a certain aspect of a certain problem, but I want it as a piece of a much bigger solution. The technical folks in banks and finance and chemical design already have environment(s) they are used to working inside and I think they will eventually take [quantum] and plug it in." (P2, applied physicist)*

In practice, this need may be met by providing developers with domain-specific APIs that solve problems from a higher-order specification, leveraging quantum on the back-end but not exposing it outright. For example, domain-specific APIs such as Google's OpenFermion [3, 17] (chemistry) and IBM's Qiskit Aqua [52, 58] (chemistry, AI, and optimization) enable developers to perform experiments on quantum computers and simulators without having knowledge of the underlying concepts of quantum computing.



(a) IBM Q Experience Composer [47]. The H and CNOT (+) gates are used to entangle two qubits. The measurement gates (purple) then measure the state of each qubit.



(b) Quirk [14]. The H and CNOT (+) gates are used to entangle two qubits. The subsequent visualizations show the measurement state of each qubit; note they are identical.

Figure 1: Graphical interfaces for programming quantum systems. Horizontal lines represent individual qubits. Boxes/circles on the lines represent quantum gates. Some gates operate on multiple qubits, represented by vertical lines between qubits.

Research opportunities. As discussed earlier, we have captured only a preliminary snapshot of quantum computing users. Additional research is needed to more comprehensively identify the people who are currently using quantum computers and those who have a desire to learn. For quantum scientists, we learned that the tools important to their workflows satisfy different needs: rapid experimentation, complex programming, and code sharing. However, the tools they mentioned using, and others not mentioned, are somewhat disconnected from each other; for example, code written in IBM's Composer cannot easily be shared with others in Github. Further research is needed to more comprehensively analyze the workflows of quantum scientists in order to understand the extent to which existing tools are truly satisfying their needs, especially for the collaborative experiment-code-share cycle discussed by our participants. Science enthusiasts and developers may also benefit from more integrated tooling, especially when strong tutorials and documentation are incorporated. Furthermore, we can leverage lessons learned from existing work on how software engineering practice is taken up by non-software-engineer scientific programmers, such as documentation, code sharing, and community management [61].

Developing and Evaluating Educational Materials

Quantum computing is a highly-technical subject, with a high barrier between understanding basic concepts such as superposition and entanglement and being able to craft quantum algorithms that perform useful computations. As one of our interviewees succinctly described, *"quantum computing is a esoteric subject." (P6, applied mathematician).*

That said, several notable efforts have aimed to make quantum computing more accessible, interesting, and fun for everyone. For beginners, these efforts include low-math or no-math introductions to quantum computing, metaphor-based approaches to teaching fundamental principles of quantum mechanics, and educational games that teach quantum computing concepts in fun ways. For quantum scientists, the efforts center around creating online communities that encourage information exchange and collaboration in the field.

Teaching Beginners with Guides and Metaphors. One of the main challenges in designing educational content for quantum computing is to identify an appropriate level of complexity for the audience. Over the last decade, there has been an emergence of introductory books [28, 70] that are intended for general audiences. Additionally, institutions including IBM [48], D-Wave [59], and Microsoft [35] provide introductory, low-math/no-math guides and tutorials for beginners to quantum computing.

Another approach to teach the principles of quantum computing is via metaphors. Many readers will be familiar with the metaphor of Schrödinger's cat [19], which explains the concept of superposition via thought experiment – a cat kept in a box with a bottle of poison is simultaneously dead and alive because it is unknown whether the cat drank the poison. Another metaphor for superposition uses a box with a ball inside [44]; when one door is opened, the ball is observed to be red, but when the other door is opened (the superposition door), we may observe the ball to be green. D-Wave uses a light switch game [59] to teach how quantum mechanics can be used to solve combinatorial optimization problems. Metaphors such as these can be more easily understood by those lacking a theoretical physics education, although their ability to convey accurate mental models for how quantum mechanics works is not well understood.

Quantum Games. Games have effectively been used for teaching educational concepts because they create situations of collaborative learning and positive interdependence amongst players [53]. Through the mechanics of game play, players learn the educational concepts embedded within the game.

Several noteworthy digital games have been designed to specifically teach about quantum computing. The Quantum Game [36] teaches quantum mechanics via physics-based puzzles. Quantum Tic-Tac-Toe [8] teaches concepts such as superposition and measurement. Meqanic [33] and Hello Quantum [45, 51] are iOS games that both require players to solve puzzles by manipulating quantum states using quantum gates, albeit with different visual representations. Meqanic, in particular, was developed with the explicit goal of teaching players how to create "quantum intuition" through trial-and-error by having players manipulate quantum circuits and see the results in a visualization [60] (Figure 2e). Quantum Moves [57] is a citizen-science game that helps physicists solve optimization problems in quantum physics.

Quantum computing has moved out of the digital realm and onto paper, as well. Entanglion [50] is a board game designed to teach fundamental concepts of quantum computing including qubits, quantum states, superposition, entanglement, measurement, and error. It also exposes players to some of the hardware and software components that make up a quantum computer. The game is a technically-accurate simulation of a 2-qubit system designed in concert with quantum scientists at IBM. An evaluation with young students and industry professionals showed that it was enjoyable and fun, and it gave players a positive impression of quantum computing [63].

Quantum Communities. Online Q&A communities have been quite successful for technical domains, and some of our participants reported being highly active in the IBM Q Experience Community [47]. Despite our understanding of strategies that help community members achieve successful individual-group interactions, such as asking questions using less complex language [1] or employing politeness [6], our participants expressed frustration and disengagement with the community. Many issues stemmed from how the platform did not distinguish between discussions amongst novices vs. experts. Our participants – experts in their field – expressed preferences for specialized communities that encouraged deep discussions and information exchange over those in which experts were tasked with answering novice questions. However, as experts in their field, our participants' opinions do not capture the needs or desires of novices and amateurs; further work is needed to capture their opinions.

> *"What I've noticed about the community is that there are few engaging questions to answer... there are a few questions that are a bit cumbersome to answer because people ask for particular instructions." (P7, theoretical physicist)*

There is a significant knowledge gap between quantum computing experts and novices just coming to the field. Some of our participants expressed frustration when novices expressed misconceptions about what quantum computers were capable of or when they veered off topic.

> *"When you go on the community you sometimes read questions from people who clearly have no idea what quantum computing is because they have high expectations of what it can do and that is not the case." (P6, applied mathematician)*

Another participant expressed a desire for question-askers to spend more time searching for answers themselves before engaging with the community, as well as spend more time deepening their knowledge of quantum computing. Our experts spent many years learning their field and although they generally do enjoy sharing their knowledge, they do not want to spend their energy on answering novice questions.

> *"For a while there were posts that I consider to be inappropriate but I think they are gone and those people have left... the posts need to be on subject. I mean they could vary a little bit, [but] people ask the same questions over and over again. They say 'I'm new help me' but it would be so much more helpful if they looked at the other thousands of posts and seen the answers there. I think one of the dangers is that because [quantum computing] is open to anyone, people get the feeling that it is easy to learn. It is not easy. You gotta put in the effort." (P4, quantum chemist)*

> *"It's really hard to try to teach someone across a web forum what people have tried to study for years... this costs energy to explain things." (P7, theoretical physicist)*

Research opportunities. Opportunities abound for HCI researchers in the space of creating and evaluating educational materials for beginners to quantum computing. First, although several low-math/no-math guides have been created to teach beginners the basics of quantum computing, the extent to which these guides are actually effective at helping beginners attain mastery is unknown. In addition, the use of metaphors in science education can be quite effective when the relevant attributes are identified and explored [2]. However, the creation of an *effective* metaphor can be a difficult process requiring a broad range of interdisciplinary skills – domain expertise, design, copy writing, evaluation – that the HCI community is highly qualified to orchestrate.

Other mechanisms for learning quantum, such as through games, also seem promising because of their ability to engage learners and create situations of collaborative learning and positive interdependence [53]. The effectiveness of these games should also be evaluated for their ability to scaffold learners through the process of understanding core quantum computing concepts to the point of programming quantum systems. In addition, it is unknown whether an intuitive, trial-and-error understanding of how quantum gates manipulate quantum states leads to an increased ability to learn quantum SDKs and program quantum algorithms.

Online communities for quantum scientists can be vastly improved by applying the lessons learned by HCI researchers in building successful online communities [31]. For example, topic modeling [22] and the creation of expertise profiles [32] can ensure questions are routed to the relevant set of people. Screening questions can be used to filter out or deprioritize novice questions, entry barriers can be used to build up the expertise of novices before they are able to participate, and identifying uniqueness in expertise can be used to motivate experts to answer questions [31].

Developing and Evaluating Quantum Tooling

The development of programming languages for quantum computers largely mirrored that of classical computers, albeit on a much more rapid timescale. Early quantum systems programming evolved from placing quantum gates directly in a circuit (as shown in Figure 1) to writing assembly-style code using the Open Quantum Assembly Language (OpenQASM). Newer SDKs such as Qiskit [49, 52] rely on Python to provide high-level imperative and functional programming semantics, which are then translated down to the level of QASM before executing on real quantum hardware or in simulation. LIQUi|> is another tool suite for quantum computing that includes both a programming language, Q#, as well as a set of quantum simulators. Circuits are expressed in the high-level Q# language, which are then translated into low-level machine instructions for a quantum device [62].

Our participants use a wide range of tools to conduct their research work, including pencil and paper, visual programming tools, and programming languages like Python and Matlab. One participant even described a workflow involving Microsoft Excel.

> *"The tool chain starts with an Excel spreadsheet. Excel spat out awk and awk wrote C and C invoked CTF [Cyclops Tensor Framework] on the super computer." (P2, physicist)*

Paper, GUIs, and Code. Despite the numerous tools available for programming quantum systems, low-fidelity tools have a strong foothold in the day-to-day workflows of quantum scientists. A theoretical physicist (P7) reported that the ideal quantum computing tools in many instances were a pen and paper. However, a quantum chemist (P4) reported the importance of GUIs and other quantum computing end-user tools. She remarked that visual tools allowed her to think in novel ways when approaching problems, and they made the learning process more enjoyable.

> *"There is a big difference between writing a circuit on a piece of paper and actually trying to implement it on a machine where something like a Toffoli gate requires numerous gates. It's like trying to learn computing just reading the theory of computing but never learning a programming language or getting anything to work or debug anything." (P4, quantum chemist)*

Visual tools like IBM's Composer [47] and Quirk [14] (Figure 1) use a circuit-like metaphor in which each qubit is represented by a horizontal line. Quantum gates are then placed along these lines indicating the order of operations. A physicist closely involved with the development of the first version of the Composer interface remarked on the ease of understanding this metaphor.

> *"The beauty of the original [Composer] GUI was that once you understood the representation of it, it was really easy to understand. There are only 5 qubits and time went from left to right, so I think it's a great on boarding mechanism for people because of the level of abstraction." (P2, physicist)*

Conversely, because each qubit uses a horizontal line of screen real estate, the circuit metaphor does not scale with the increasing numbers of qubits becoming available in newer quantum computers and the complexity of the problems being solved.

> *"This is no longer kindergarten so we moved from graphics to Qiskit... you can express in text form much more complex thoughts than graphically." (P2, physicist)*

As is the case for many other technical domains, our participants indicated that they prefer low fidelity tools such as paper & pencil as the initial step in their workflow. Only later do they use tools such as visual programming UIs and programming languages to formalize their ideas. Paper and pencil are preferred in other domains because of the unconstrained nature of sketching and the nature of moving a pen across paper with very fine tactile control [24, 66]. Systems for transcribing or digitizing sketches have been created for other domains, such as medical documentation in hospitals [66]. We posit that the math-heavy nature of quantum computing makes pen and paper desirable tools, but further research is needed to understand the difficulties quantum scientists have in translating their sketch work into digitized, sharable, and executable forms.

Documentation and Sharing. As more people delve into quantum computing, such as science enthusiasts and developers, a need arises for environments that better integrate documentation and sharing tools. Well-documented workflows are integral to the learning process. Research on transparency in Github reveals that users leverage transparency to improve collaboration and advance their technical skills [10].

One quantum scientist expressed the challenge of sharing knowledge about quantum.

> *"We realized it might be simple to write or replicate an algorithm but hard to explain to others what you are doing." (P2, physicist)*

Technologies like Jupyter notebooks allow users to showcase their quantum computing projects and experiments. However, in order to make quantum computing education more transparent and accessible, HCI researchers can consider how development and dissemination workflows for quantum differs from workflows in the classical computing world. For example, paired visualizations of quantum gates and states (e.g. Figure 1b) with code may not be as important in classical programming, but may be a boon to programming and debugging quantum systems.

> *"I think something between Composer and Qiskit – a visual aspect would be helpful. You could design a circuit using gates and your Jupyter notebook is integrated with Composer. A visual version of Qiskit would be more understandable." (P4, quantum chemist)*

This desire demonstrates the importance of designing environments that support transparency of workflows and furthermore represents a space for HCI researchers to ideate, iterate, and evaluate front-end tools to enhance and improve the user experience for quantum computing users.

Debugging Quantum Algorithms. As with any development workflow, debugging is a part of the quantum computing process. One participant described how she debugged her code by switching amongst different tools.

> *"When I am trying to figure out an algorithm or debug something, I will start with the Composer. If I want to make any changes I go over the QASM to change things there and I go back and forth between the two." (P4, quantum chemist)*

Another described how he used a quantum simulator to perform debugging, as the simulator enabled one to view the internal state of all of the qubits.

> *"To debug we used a C simulator for everything... out of the simulator you can get internal state of all qubits." (P2, physicist)*

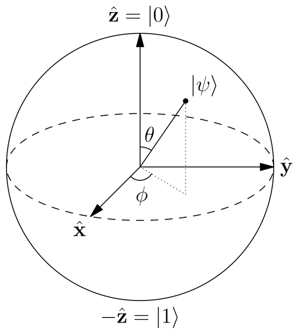
While we know that debugging is an important part of the development cycle, it is inefficient when people have to use multiple tools to debug or cannot debug on actual hardware. HCI researchers and practitioners can aid in this area. For example, novel debugging paradigms such as Whyline [30] have been shown to reduce debugging time for classical systems by enabling programmers to ask *why* and *why not* questions about their code's behavior. We have yet to see such novelty in debugging quantum systems. A deeper understanding of how quantum programmers conduct the debugging process would provide opportunities for developing new kinds of quantum debugging tools. This work should build on previous research that demonstrates the best practices for scientific computing, such as performing optimizations only after a program is correct [65].

"Quantum Native" Programming. Over the last few decades, programming languages have evolved into higher-level forms as technologies have abstracted low-level implementation details. Most programming languages today strive to capture the intent of the programmer, and special-purpose frameworks and compilers translate that intent down to the specific mechanical instructions that drive the CPU (or GPU). The field of quantum computing is experiencing a similar evolution, but more rapidly. Quantum programming has shifted from building low-level quantum circuits, to programming in quantum assembly, to programming in higher-level languages like Python and Q#. Yet, deep knowledge is still required to productively program quantum systems, even when using higher-level languages.

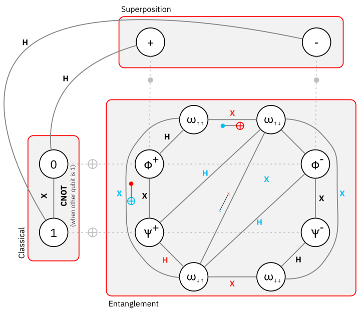
> *"Today you need a strong background in quantum computing. You need to understand quantum physics in order to work with a quantum program. This is how assembly programming language was in the 50s... In quantum there is no higher level of abstraction. We are waiting for that day." (P2, physicist)*

One opportunity for HCI researchers to consider is whether the abstractions used to program classical computers should be applied to programming quantum systems. How can traditional notions of "register," "variable," and "conditional statement" sufficiently capture the probabilistic state of a qubit, the accumulated error accrued by the depth of a quantum circuit, or the statistical nature of quantum algorithms? Might there be alternative models of programming that explicitly capture these qualities, and what would they look like? HCI researchers are ideally situated to ponder radical new "quantum native" programming interfaces and environments, due to expertise with methods for improving programming tools [40] and past contributions on the development and improvement of numerous programming paradigms (e.g. [39]), including pedagogical paradigms (e.g. [9]). Such development should include collaboration with research communities already dedicated to the design of quantum programming languages (e.g. [18]).

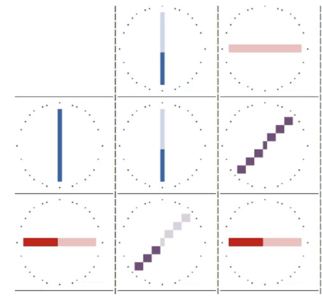
In addition to radical new designs, existing tooling may also be improved in collaboration with other scientific programming communities. For example, embedded systems programming has challenges similar to quantum programming around abstraction, running in simulation vs. hardware, debugging, and validating a program's correctness [4]. Interfaces for FPGA programming, such as Intel's Quartus Prime, use similar graphical programming metaphors as interfaces



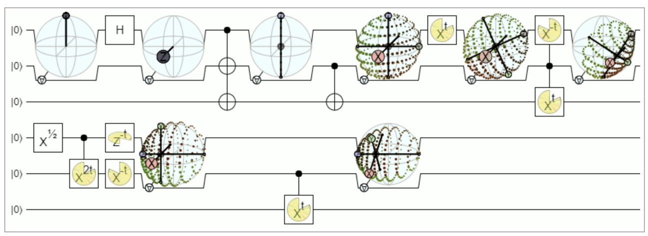
(a) The Bloch sphere [64]. The vector $|\phi\rangle$ represents the state of a qubit.



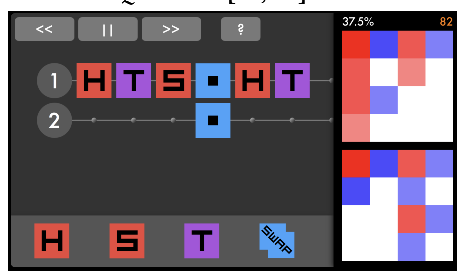
(b) Schematic representation of two-qubit states and transitions used in Entanglion [50].



(c) Schematic representation of two-qubit states used in Hello Quantum [45, 69].



(d) A visualization of two entangled qubits from Gidney [15].



(e) Meqanic [33] uses the Pauli matrix representation to show the state of two qubits.

Figure 2: Different visualizations of qubits.

for quantum programming. Thus, HCI researchers have an opportunity to refine and optimize quantum programming GUIs by leveraging work done to support other scientific communities.

Research Opportunities. Our understanding of the collaborative needs and practices of quantum programmers, beyond sharing Jupyter notebooks, is quite limited. A deeper understanding of existing quantum workflows – toolchains, collaboration, and debugging – is needed in order to identify ways in which they can be optimized. For example, collaborative debugging of quantum algorithms may prove challenging due to the constraints of quantum hardware; as quantum states cannot be cloned, quantum programmers may not be able to debug each others' work. In addition, the transition points between low-fidelity tools (e.g. pen and paper) to high-fidelity tools (e.g. Python) needs more study.

One specific area that could benefit science enthusiasts and developers is in the usability of quantum APIs. API usability studies have been conducted for programming classical computers [40, 41], and have led to improvements in autocomplete, documentation, and usage analysis. Usability studies of quantum APIs could improve their accessibility to audiences of developers without formal training in quantum computing, and develop new aids for creating novel, domain-specific quantum algorithms.

Visualizations of Quantum State

Visualizing qubit states is another challenging area to which HCI and visualization researchers can contribute. A common visual representation for a single qubit's state is the Bloch sphere (Figure 2a). This representation shows how a qubit can be in a superposition state between ground ($|0\rangle$) and excited ($|1\rangle$).

One major limitation of the Bloch sphere is that it is unable to show the entanglement of two qubits. Gidney [15] describes in great detail an attempt to use multiple, animated Bloch spheres to show two-qubit entanglement (Figure 2d), but the results seem difficult to interpret at-a-glance and no evaluation was performed on its efficacy.

Another approach for representing qubit state is using matricies of Pauli Expectation Values (PEVs) [20, 67]. This visual representation is the basis for the Meqanic [33, 60] and Hello Quantum [45] iOS games. Its defining characteristic is that qubits are arranged in a matrix, with colors denoting possible measurement outcomes. Wootton [69] gives an overview for how this visualization is derived and how it relates to the Bloch sphere.

Not all visualizations need to focus on a qubit's state, however. The two-qubit representation used by Entanglion [50] focuses on the *transitions* between qubit states (Figure 2b). This diagram clearly shows the effects of how different quantum gates act on a two-qubit system in a given state; for example, the top-leftmost transition line shows how a qubit in a classical $|1\rangle$ state transitions to the superposition $|-\rangle$ state via the H gate (and vice versa).

Research Opportunities. HCI researchers can utilize information visualization methods [13] to build more effective visualizations for quantum computing. Visualizations of qubit states are particularly difficult to create because the number of achievable quantum states is exponential with respect to the number of qubits. A quantum computer with n qubits can be in a superposition of up to 2^n different states simultaneously, whereas a classical computer can only be in one of those states at a time. This explosion of state space necessitates the development of scalable visualizations that can intuitively help quantum developers understand the states their systems are in, in order to facilitate the development, debugging, and validation of quantum algorithms. In addition, user studies exploring the interpretability of large-scale graphs and other large-scale visualizations can contribute to improved visualizations of quantum states. Synergies may exist with the deep learning community as they develop visualizations of complex models, such as GANs [26], which also have an immensely high dimensionality.

5 SKEPTICISM ABOUT QUANTUM COMPUTING

While this paper favorably suggests potential directions of expansion for quantum computing, we acknowledge that our optimism must be tempered with the realities of the field. As enthusiastic as our participants were to discuss quantum computing with us, they expressed uncertainty about its future and the need for ubiquitous quantum computers.

> *"It's possible that quantum computers work well for very certain tasks but these tasks are not ubiquitous tasks. They are specific. Sixty years ago, we didn't think we would have a universal Turing calculator in our pocket, but here we are. I don't see why an average person that doesn't do scientific computing would need a quantum computer." (P6, applied mathematician)*

> *"At the moment we struggle with the WIIGF problem - what is it good for? I think the key will be isolation of problem domains where quantum computers will be useful." (P2, physicist)*

> *"I'm generally optimistic, [but] not as much as what you read in the media... With regards to error correction my concern is that we won't ever be able to build this thing. Then, it's still not completely clear what we will use these things for. I'm hopeful, but I don't have too much." (P3, Ph.D. student)*

An Impending "Winter of Quantum"?

In addition to skepticism over the usefulness or ubiquity of quantum computing, some participants expressed trepidation over an impending "winter of quantum." They compared the current hype of quantum computing with the hype experienced by AI in decades past. In part due to a lack of significant breakthroughs, AI funding dried up, resulting in a "winter of artificial intelligence." [2]

> *"The same thing happened with Nano and AI. At one point, they get hyped because they make a few advances and then they don't deliver on their promises, and then they get trashed so then no one gives them attention. While they are being ignored, they slowly or maybe even incrementally improve so suddenly they have some major advances." (P4, quantum chemist)*

> *"We don't want the same thing with quantum [as with AI] – over-hyping it and then realizing that you can't accomplish what you want to accomplish and then abandoning it." (P6, applied mathematician)*

With any technology having such a large, but potentially unknown impact, people may have inflated expectations. When those expectations are not satisfied, they may become disillusioned. We argue that the uncertainties around quantum computing and the potential for disillusionment should not be reasons to avoid approaching the field. Instead, we see them as opportunities to show the strength that great HCI research and practice can bring. By making quantum computing easier to understand and conduct, we can in turn enable the revolutionary breakthroughs that are so desired.

6 CONCLUSION

We introduce the HCI community to the field of quantum computing and seek to formalize a new practice of Quantum HCI (QHCI). From interviews with experts in the field, we identified four areas central to this practice: understanding current and future quantum computing users, developing new tools for programming and debugging quantum systems, creating new educational materials to teach quantum computing, and designing visualizations that aid in understanding quantum states. In many ways, quantum computing is still in a state of infancy, but many feel that it has huge potential in fields such as chemistry, cryptography, and finance. We encourage the HCI community to think deeply about how best to support quantum computing users and further establish the field of human-quantum computer interaction.

[2] A historical summary of the AI winter is given by Grudin [21], with contrasts to the co-evolution of HCI.

REFERENCES

[1] Jaime Arguello, Brian S. Butler, Elisabeth Joyce, Robert Kraut, Kimberly S. Ling, Carolyn Rosé, and Xiaoqing Wang. 2006. Talk to Me: Foundations for Successful Individual-group Interactions in Online Communities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 959–968. https://doi.org/10.1145/1124772.1124916

[2] Peter J Aubusson, Allan G Harrison, and Stephen M Ritchie. 2006. *Metaphor and analogy in science education*. Vol. 30. Springer Science & Business Media.

[3] Ryan Babbush and Jarrod McClean. 2017. Announcing OpenFermion: The Open Source Chemistry Package for Quantum Computers. (Oct. 2017). https://ai.googleblog.com/2017/10/announcing-openfermion-open-source.html [Online; accessed 5-September-2018].

[4] David F Bacon, Rodric Rabbah, and Sunil Shukla. 2013. FPGA programming for the masses. *Commun. ACM* 56, 4 (2013), 56–63.

[5] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. 1997. Strengths and weaknesses of quantum computing. *SIAM journal on Computing* 26, 5 (1997), 1510–1523.

[6] Moira Burke and Robert Kraut. 2008. Mind Your Ps and Qs: The Impact of Politeness and Rudeness in Online Communities. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW '08)*. ACM, New York, NY, USA, 281–284. https://doi.org/10.1145/1460563.1460609

[7] Rigetti Computing. 2017. Forest SDK. https://www.rigetti.com/forest [Online; accessed 13-September-2018].

[8] Wikipedia contributors. 2017. Quantum tic-tac-toe — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Quantum_tic-tac-toe [Online; accessed 5-April-2018].

[9] Stephen Cooper, Wanda Dann, and Randy Pausch. 2000. Alice: a 3-D tool for introductory programming concepts. In *Journal of Computing Sciences in Colleges*, Vol. 15. Consortium for Computing Sciences in Colleges, 107–116.

[10] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*. ACM, 1277–1286.

[11] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. 2001. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* 292, 5516 (2001), 472–475.

[12] Michael H Freedman. 1998. P/NP, and the quantum field computer. *Proceedings of the National Academy of Sciences* 95, 1 (1998), 98–101.

[13] Carla MDS Freitas, Paulo RG Luzzardi, Ricardo A Cava, Marco Winckler, Marcelo S Pimenta, and Luciana P Nedel. 2002. On evaluating information visualization techniques. In *Proceedings of the working conference on Advanced Visual Interfaces*. ACM, 373–374.

[14] Craig Gidney. 2016. Quirk: Quantum Circuit Simulator. http://algassert.com/quirk [Online; accessed 13-September-2018].

[15] Craig Gidney. 2017. Visualizing 2-Qubit Entanglement. (2017). http://algassert.com/post/1716 [Online; accessed 5-September-2018].

[16] Google. 2014. Quantum Computing Playground. http://www.quantumplayground.net [Online; accessed 13-September-2018].

[17] Google. 2017. OpenFermion. http://www.openfermion.org [Online; accessed 5-September-2018].

[18] Alexander S Green, Peter LeFanu Lumsdaine, Neil J Ross, Peter Selinger, and Benoît Valiron. 2013. Quipper: a scalable quantum programming language. In *ACM SIGPLAN Notices*, Vol. 48. ACM, 333–342.

[19] John Gribbin. 2011. *In search of Schrodinger's cat: Quantum physics and reality*. Bantam.

[20] David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. 2010. Quantum state tomography via compressed sensing. *Physical review letters* 105, 15 (2010), 150401.

[21] Jonathan Grudin. 2009. AI and HCI: Two fields divided by a common focus. *AI Magazine* 30, 4 (2009), 48.

[22] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. 2008. Tapping on the Potential of Q&a Community by Recommending Answer Providers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 921–930. https://doi.org/10.1145/1458082.1458204

[23] Johannes Hachmann, Roberto Olivares-Amaya, Adrian Jinich, Anthony L Appleton, Martin A Blood-Forsythe, Laszlo R Seress, Carolina Roman-Salgado, Kai Trepte, Sule Atahan-Evrenk, Süleyman Er, et al. 2014. Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry–the Harvard Clean Energy Project. *Energy & Environmental Science* 7, 2 (2014), 698–704.

[24] Tomoko Hashida, Kohei NIshimura, and Takeshi Naemura. 2012. Handrewriting: automatic rewriting similar to natural handwriting. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*. ACM, 153–162.

[25] IBM. 2017. IBM Announces Advances to IBM Quantum Systems & Ecosystem. https://www-03.ibm.com/press/us/en/pressrelease/53374.wss [Online; accessed 5-April-2018].

[26] Minsuk Kahng, Nikhil Thorat, Duen Horng Polo Chau, Fernanda B Viégas, and Martin Wattenberg. 2019. GAN Lab: Understanding Complex Deep Generative Models using Interactive Visual Experimentation. *IEEE transactions on visualization and computer graphics* 25, 1 (2019), 310–320.

[27] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549, 7671 (2017), 242.

[28] Phillip Kaye, Raymond Laflamme, Michele Mosca, et al. 2007. *An introduction to quantum computing*. Oxford University Press.

[29] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, et al. 2016. Jupyter Notebooks-a publishing format for reproducible computational workflows.. In *ELPUB*. 87–90.

[30] Andrew J Ko and Brad A Myers. 2004. Designing the whyline: a debugging interface for asking questions about program behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 151–158.

[31] Robert E Kraut and Paul Resnick. 2012. *Building successful online communities: Evidence-based social design*. Mit Press.

[32] Jennifer Lai, Jie Lu, Shimei Pan, Danny Soroker, Mercan Topkara, Justin Weisz, Jeff Boston, and Jason Crawford. 2014. Expediting Expertise: Supporting Informal Social Learning in the Enterprise. In *Proceedings of the 19th International Conference on Intelligent User Interfaces (IUI '14)*. ACM, New York, NY, USA, 133–142. https://doi.org/10.1145/2557500.2557539

[33] Tareso, LLC. 2018. *Meqanic*. Game [iOS]. http://www.meqanic.com/ [Online; accessed 5-April-2018].

[34] Microsoft. 2016. LIQUi|>: The Language Integrated Quantum Operations Simulator. http://stationq.github.io/Liquid/ [Online; accessed 13-September-2018].

[35] Microsoft. 2018. A Beginner's Guide to Quantum Computing and Q#. https://blogs.msdn.microsoft.com/uk_faculty_connection/2018/02/06/a-beginners-guide-to-quantum-computing-and-q/ [Online; accessed 17-September-2018].

[36] Piotr Migdał, Patryk Hes, and Michal Krupiński. 2018. The Quantum Game. http://play.quantumgame.io [Online; accessed 5-September-2018].

[37] Ron Miller. 2017. IBM makes 20 qubit quantum computing machine available as a cloud service. (2017). http://tcrn.ch/2i13jH3 [Online; accessed 13-September-2018].

[38] MIT. 2018. MIT Quantum Computing Curriculum. https://quantumcurriculum.mit.edu [Online; accessed 13-September-2018].

[39] Brad A Myers. 1986. Visual programming, programming by example, and program visualization: a taxonomy. In *ACM sigchi bulletin*, Vol. 17. ACM, 59–66.

[40] Brad A Myers, A J Ko, T D LaToza, and Y S Yoon. 2016. Programmers are users too: Human-centered methods for improving programming tools. *Computer* 49, 7 (2016), 44–52.

[41] Brad A Myers and Jeffrey Stylos. 2016. Improving API usability. *Commun. ACM* 59, 6 (May 2016), 62–69.

[42] CBC News. 2018. Canadian Prime Minister Justin Trudeau schools reporter on quantum computing during press conference. https://youtu.be/Eak_ogYMprk [Online; accessed 13-September-2018].

[43] Michael A Nielsen and Isaac Chuang. 2002. Quantum computation and quantum information.

[44] John Preskill. 2013. Can we exploit the weirdness of quantum mechanics? *Physics World* 26, 10 (2013), 45–46.

[45] Howard Rabinowitz. 2018. Hello Quantum: The Making of a Seriously Fun Quantum Game. (2018). https://www.ibm.com/blogs/research/2018/07/hello-quantum/ [Online; accessed 5-September-2018].

[46] Kevin Resch. 2018. IQC receives $15M from Government of Canada. https://uwaterloo.ca/institute-for-quantum-computing/news/iqc-receives-15m-government-canada [Online; accessed 13-September-2018].

[47] IBM Research. 2016. IBM Q Experience. https://quantumexperience.ng.bluemix.net [Online; accessed 30-August-2018].

[48] IBM Research. 2017. IBM Q Full Guide. https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=full-user-guide [Online; accessed 30-August-2018].

[49] IBM Research. 2017. Qiskit. https://qiskit.org [Online; accessed 30-August-2018].

[50] IBM Research. 2018. Entanglion. http://entanglion.github.io [Online; accessed 5-September-2018].

[51] IBM Research. 2018. Hello Quantum. http://helloquantum.mybluemix.net [Online; accessed 5-September-2018].

[52] IBM Research. 2018. QISKit Aqua. https://qiskit.org/aqua [Online; accessed 5-September-2018].

[53] Margarida Romero, Mireia Usart, Michela Ott, Jeffrey Earp, and Sara de Freitas. 2012. Learning through playing for or against each other? Promoting collaborative learning in digital game based learning. *Learning* 5, 2012 (2012), 15–2012.

[54] Peter W Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. Ieee, 124–134.

[55] Peter W Shor. 1999. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* 41, 2 (1999), 303–332.

[56] Rep. Lamar Smith. 2018. H.R.6227 - National Quantum Initiative Act. https://www.congress.gov/bill/115th-congress/house-bill/6227 [Online; accessed 13-September-2018].

[57] Jens Jakob W H Sørensen, Mads Kock Pedersen, Michael Munch, Pinja Haikka, Jesper Halkjær Jensen, Tilo Planke, Morten Ginnerup Andreasen, Miroslav Gajdacz, Klaus Mølmer, Andreas Lieberoth, and Jacob F Sherson. 2016. Exploring the quantum speed limit with computer games. *Nature* 532, 7598 (April 2016), 210–213.

[58] Bob Sutor. 2018. Domain Experts, Welcome to Quantum: Introducing QISKit ACQUA. (June 2018). https://www.ibm.com/blogs/research/2018/06/domain-experts-qiskit-acqua/ [Online; accessed 5-September-2018].

[59] D-wave Systems. 2018. Quantum Computing Primer. https://www.dwavesys.com/tutorials/background-reading-series/quantum-computing-primer/#h2-2 [Online; accessed 13-September-2018].

[60] Charles Tahan. 2015. Quantum Mechanics is Intuitive or Why We Made a Game Called Meqanic. http://tahan.com/charlie/talks/2015MMquantumintuitiontalk.pdf [Online; accessed 18-September-2018].

[61] Erik H Trainer, Chalalai Chaihirunkarn, Arun Kalyanasundaram, and James D Herbsleb. 2015. From personal tool to community resource: What's the extra work and who will do it?. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 417–430.

[62] Dave Wecker and Krysta M Svore. 2014. LIQUi|>: A software design architecture and domain-specific language for quantum computing. *arXiv preprint arXiv:1402.4467* (2014).

[63] Justin D. Weisz, Maryam Ashoori, and Zahra Ashktorab. 2018. Entanglion: A Board Game for Teaching the Principles of Quantum Computing. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '18)*. ACM, New York, NY, USA, 523–534. https://doi.org/10.1145/3242671.3242696

[64] Wikipedia contributors. 2018. Bloch sphere — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Bloch_sphere&oldid=861907167 [Online; accessed 4-January-2019].

[65] Greg Wilson, Dhavide A Aruliah, C Titus Brown, Neil P Chue Hong, Matt Davis, Richard T Guy, Steven HD Haddock, Kathryn D Huff, Ian M Mitchell, Mark D Plumbley, et al. 2014. Best practices for scientific computing. *PLoS biology* 12, 1 (2014), e1001745.

[66] Christian Winkler, Julian Seifert, Christian Reinartz, Pascal Krahmer, and Enrico Rukzio. 2013. Penbook: Bringing pen+ paper interaction to a tablet device to facilitate paper-based workflows in the hospital domain. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. ACM, 283–286.

[67] HM Wiseman and GJ Milburn. 1993. Interpretation of quantum jump and diffusion processes illustrated on the Bloch sphere. *Physical Review A* 47, 3 (1993), 1652.

[68] William K Wootters and Wojciech H Zurek. 1982. A single quantum cannot be cloned. *Nature* 299, 5886 (1982), 802–803.

[69] James Wootton. 2018. Visualizing bits and qubits. (2018). https://medium.com/qiskit/visualizing-bits-and-qubits-9af287047b28 [Online; accessed 5-September-2018].

[70] Noson S Yanofsky, Mirco A Mannucci, and Mirco A Mannucci. 2008. *Quantum computing for computer scientists*. Vol. 20. Cambridge University Press Cambridge.