Introducing QRogue: Teaching Quantum Computing Using a Rogue-like Game Concept

Michael Artner
Johannes Kepler University Linz
Linz, Austria
michael.artner@jku.at

Günter Wallner
Johannes Kepler University Linz
Linz, Austria
guenter.wallner@jku.at

Robert Wille
Technical University of Munich
Munich, Germany
robert.wille@tum.de

ABSTRACT

Recent years have shown that we are steadily getting closer to industrial applications of quantum computing. As such it is important to teach quantum computing concepts to users to allow them to incorporate quantum computing into their toolbox. As educational research has shown the potential of game-based learning in the past years, we are thus proposing *QRogue*, an educational game with Rogue-like elements targeted at computer science students. The game's goal is to teach the math behind quantum computing in a playful environment with analogies to this technology's counter-intuitive fundamentals. To gather first feedback and stir the further development, we conducted a user study – involving playtesting and a post-experience survey – with eight students showing that the game was positively received but requires further tuning of the onboarding process and of the in-game feedback provided to players.

CCS CONCEPTS

• Applied computing → Computer games; **E-learning**; **• Hardware → Quantum computation**.

KEYWORDS

educational games, quantum computing, serious games

ACM Reference Format:
Michael Artner, Günter Wallner, and Robert Wille. 2023. Introducing QRogue: Teaching Quantum Computing Using a Rogue-like Game Concept. In *Foundations of Digital Games 2023 (FDG 2023), April 12–14, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3582437.3587177

1 INTRODUCTION

Quantum computing [9] is an emerging technology with promising applications in domains such as cryptography and machine learning. This is because it is believed that quantum computers solve certain problems in these domains more efficiently than conventional computers. Examples include integer factorization based on Shor's algorithm [12] as well as quantum singular value decomposition [3]. The latter can be used as a subroutine for a variety of important data analysis tasks such as principal component analysis [1] to decrease the size of a matrix without loosing too much information.

While being a purely fundamental research topic in the past decades, recent accomplishments in quantum computing (e.g., quantum supremacy [11]) triggered more and more practical interest and are moving the topic towards applied and even industrial research. This can, e.g., be seen through "big players" such as IBM, Google, Intel, etc. developing own quantum computers as well as comprehensive ecosystems around that [4]—aimed at a future practical utilization of this technology.

However, being able to effectively use quantum computers requires specific expertise of the underlying concepts. While conventional computing also required expertise, it is particularly challenging for quantum computing as users do not only have to learn and understand quantum-mechanical properties such as superposition and entanglement, but also corresponding reversible computing paradigms [15], the probabilistic nature of quantum states, and, most importantly, quantum operations that are vastly different from "common" operations [9]. Accordingly, educating and familiarizing future users with those new concepts and paradigms is key for the success of this promising technology.

Educational games have shown to be able to facilitate learner's knowledge in various domains, including mathematics (cf. [8]) which constitutes fundamental knowledge in quantum computing. Indeed, educational games such as *Quantum tic-tac-toe* [5], *Quantum Minesweeper* [6], or *Quantum Odyssey* [10] (see Section 2) have already successfully shown how to provide an intuition for the key concepts of quantum computing. However, a more detailed treatment of how quantum computations work (i.e., how quantum operations can change quantum states and, eventually, how quantum algorithms are executed) seems to be missing so far.

Thus, in this paper, we are proposing *QRogue* [1] —a Rogue-like game that is supposed to aid users in learning and understanding quantum computations in a playful fashion. Target audience hereby are mainly computer science students that already know matrix-vector multiplication and are familiar with digital circuits. To this end, the next section first reviews existing games featuring quantum computing as well as briefly reviews the math behind it. Section 3 introduces the game and its core mechanics, followed by the results of *QRogue*'s initial user study in Section 4. Finally, Section 5 concludes the paper.

2 BACKGROUND AND MOTIVATION

This section first briefly reviews selected games featuring quantum computing. Afterwards, we review an aspect which is the focus of *QRogue* (i.e., how actual quantum computations are carried out).

FDG 2023, April 12–14, 2023, Lisbon, Portugal
2023. ACM ISBN 978-1-4503-9855-8/23/04.
https://doi.org/10.1145/3582437.3587177

[1] *QRogue* is available at https://www.cda.cit.tum.de/app/qrogue

2.1 Educational Quantum Computing Games

Quantum computing-related educational games have been around for over a decade. Examples include quantum-extended versions of classics such as *Quantum tic-tac-toe* [5] or *Quantum Minesweeper* [6], as well as dedicated new games such as *Quantum Odyssey* [10]. Their basic ideas and goals are very similar: provide a playful environment to showcase the otherworldly properties and concepts of quantum computing. More precisely, they mainly revolve around providing an intuition for working with qubits (i.e. quantum bits). This includes *superposition* (being in the classical binary states 0 and 1 at the same time), *measurement* (collapsing a qubit's state to either 0 or 1) and *entanglement* (entangled qubits share their state, i.e., every operation or measurement conducted on one of them equally influences the others).

Quantum tic-tac-toe [5] simply combines two classical 3x3 grids and lets one of the players decide a measurement's outcome if a field is in superposition (e.g., when both players placed their symbol at the same field—creating a state which is X and O at the same time). This way the game can be played with pen and paper and neither of the players need to conduct any calculations.

Similarly, *Quantum Minesweeper* [6] also comes with multiple classical boards combined to one where fields might be in superposition (at least one board containing a mine while at least one other board does not). Hence, revealing a field yields a probability of containing a mine instead. Because of this, it also introduces different ways to measure (i.e., reveal) a field. However, in this game it is crucial to understand probabilities to find a save path between entangled mines.

Quantum Odyssey [10], on the other hand, works with quantum circuits (i.e., a sequence of quantum gates representing quantum operations) and, hence, provides a more technical approach as well as the possibility to "code" simple quantum algorithms within the game. However, since a gate's functionality is solely described visually (i.e., branching and combining input with output lines and animating a "qubit flow" through the circuit), the learned technicalities do not include mathematical descriptions and, hence, do not provide detailed insights in how the algorithm itself is executed.

2.2 Quantum Computations

In this work, we are proposing a game which should provide the user with a more detailed understanding about how quantum computations work. To this end, we are using a formalism which actually is not that complex: In fact, every quantum state and every quantum operation can be represented by a corresponding (complex-valued) state vector and a (unitary) operation matrix [9]. Determining the output state of an operation applied to an input state can then easily be done by matrix-vector multiplication.

For example, consider the basis state $|0\rangle$ as input q_{in} and the so-called Hadamard operation H defined as

$$q_{in} = |0\rangle = \begin{pmatrix}1\\0\end{pmatrix}, H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}.$$

Then, the output state which results when applying operation H to state q_{in} can be determined by

$$q_{out} = H \times q_{in} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}\frac{1}{\sqrt{2}}\\\frac{1}{\sqrt{2}}\end{pmatrix}.$$

This quantum computation represents the process of transforming a basis state into a state in superposition. This can be seen by the complex values of the vectors where the upper one (denoted α_0) provides the amplitude for 0 and the bottom one (denoted α_1) for 1. The results of $|\alpha_0|^2$ and $|\alpha_1|^2$ give the probability of the corresponding state evaluating to 0 or 1, respectively [9]. As can be seen, the input state evaluates to 0 with $|1|^2 = 1 \hat{=} 100\%$ probability while, after applying the H-operation, the output state evaluates to 0 or 1 with $|\frac{1}{\sqrt{2}}|^2 = 0.5 \hat{=} 50\%$ probability.

Using this formalism, *any* quantum operation and, hence, quantum algorithm can be evaluated. Based on this, the main idea of the proposed game is to illustrate this concept in an intuitive fashion also for more complex quantum operations and algorithms—eventually allowing users to learn and understand them in detail.

3 GAME IDEA AND MECHANICS

QRogue is an educational puzzle game where the player has to transform a given input to a target output state by using different quantum operations. While initial puzzles are as simple as the example provided in Section 2.2, later puzzles cover states with more qubits and a broader spectrum of quantum operations. Over time, this allows the player to learn and understand complex quantum computations [2]. In order to do that in a playful fashion, the corresponding puzzles are embedded in a Rogue-like game concept. This section first describes the core mechanic of the game, i.e., the puzzles. Afterwards, the environment, i.e., the Rogue-like game concept is described.

3.1 Computation Puzzles

As mentioned before, quantum computations are described by matrix-vector multiplications. In the game, these are embedded in puzzles as shown in Fig. 1. While input *b*) and target state *d*) are pre-determined by the puzzle, the circuit matrix *a*) can be manipulated by the player to transform the input state *b*) to the output state *c*). Once the output state has been transformed to be equal to the target state, the puzzle is solved. To achieve this, the player has to use their different quantum operations that *QRogue* models as gates in a circuit (hence, *a*) is called circuit matrix). Similar to classical circuits, Fig. 1 *e*) shows that the gates' operations are conducted from left ("In") to right ("Out"). The rows hereby specify the gates' targeted qubits while the column determines their execution order. Thus, depending on which gate (e.g., H gate applies Hadamard operation) is placed where (in terms of rows and columns), the corresponding circuit matrix will be different and, therefore, transforming the input to a different output state. Hence, in order to solve a given puzzle (i.e., determine a sequence of gates corresponding to the required circuit matrix), the players have to familiarize themselves with the corresponding vector descriptions for states, matrix descriptions for operations, and their overall application.

To this end, we are providing the player with an interface as shown in Fig. 1 *f*). In case the player decides to edit their circuit, a list of available gates is presented that can be placed in a grid-based fashion onto the circuit. Other options include removing placed gates, a "Gate Guide" to recall descriptions of available gates, "Reset"

[2]These computations are implemented using IBM's quantum SDK Qiskit [14] making it possible to run them on actual Quantum Hardware in a future update.

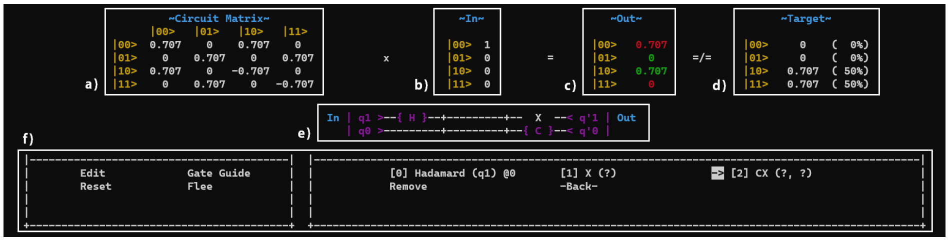

Figure 1: Screenshot of a computation puzzle in *QRogue*: The player is currently placing their `CX` gate. The bottom shows the UI (editing the circuit, guide for the different gates, reset to clear the circuit, flee to abort and solve the puzzle later). In the middle the manipulated circuit is displayed (colored gates are placed, while the white `X` is currently being placed). Lastly, the top shows the underlying matrix-vector multiplication with the output being colored based on its equality to the target.

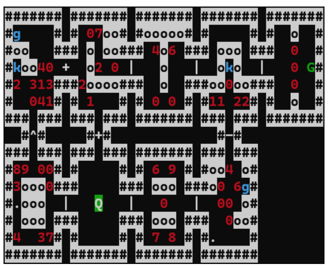

Figure 2: Screenshot of an exemplary level of *QRogue*. It includes multiple rooms, the player's Qubot (`Q` with green background), multiple puzzles (red *digits*), collectibles (`k`, `g` in blue), messages/hints (`.` in white), walls (`#`, `o` with white background), doors (`-`, `|`, `^`, `+` in white) and the goal (green `G`).

to remove all placed gates at once, as well as "Flee" to abort the current puzzle and try again later.

As the game progresses, the player will gather more gates and increase the size of the circuit. Naturally, this allows us to confront them with more complex puzzles and gives them the opportunity to implement quantum algorithms.

3.2 Rogue-like Environment

To support the core mechanic and embed it in a playful environment, the computation puzzles have been integrated in a Rogue-like game concept as shown in Fig. 2. Here, the player navigates a robot (called Qubot and denoted by a `Q` with green background) through different rooms which are displayed in a 2-dimensional, top-down, and ASCII-style fashion–similar to the genre inspiring game *Rogue* [2]. The computation puzzles are denoted by digits in red color and represent static enemies which, e.g., block your path or the entrance to the next room. In order to defeat an enemy (i.e., remove it from the map) the player has to solve its puzzle (as described in Sec. 3.1). In case the player fails and decides to flee, it will remain in its place and keep blocking the way.

However, some enemies will instead flee from you, immediately vanishing from the map without the need of solving a puzzle. This is chance-based depending on the digit the enemy is represented by (e.g., `1` corresponds to a chance of 10%, `9` to 90%, etc.). This mechanic describes the quantum computing concept of *measurement*. Whenever something non-quantum (the player's robot) interacts (moves onto) with a qubit (an enemy) its state collapses (it either flees or confronts the player with a puzzle). Additionally, within a room all enemies represented by the same digit are *entangled*. This means that either all of them instantly flee or all of them will confront the player with a puzzle. Therefore, `0`s have a special niche. Because all of them always confront the player with a puzzle, they are used in the first few levels to showcase the mandatory basics.

Besides that, collectibles (denoted by characters in blue color) are distributed across the rooms. These can be keys or gates needed to solve the puzzles. The occasional impassable door (e.g., locked with a key or blocked until a certain event triggers) breaks linearity and should encourage exploration.

Lastly, after completing the first few levels the player unlocks so-called "Expeditions". These are randomly generated levels where the player has a random choice of unlocked gates at their disposal and can find new gates to unlock by solving a boss-like puzzle.

4 USER STUDY

To gather first feedback and steer the further development of the game we conducted a small-scale user study.

Procedure. The study took about 20 minutes and consisted of a) filling an informed consent form as well as a brief demographics questionnaire, b) playtesting of the game, and c) a post-experience survey. Demographics included gender (following [13]) and current semester the student is in, prior knowledge regarding quantum computing (on a 5-point scale from 1=no knowledge to 5=professional knowledge), as well as interest in quantum computing and attitude towards educational games. The latter were phrased as agreement items and measured on a bipolar 7-point scale (-3 = strongly disagree to +3 = strongly agree). Afterwards, the participants were asked to play the game itself for around 12-13 minutes. Once the playtest was completed, the post-experience survey gathered feedback on the game itself through the miniPXI [7] and open-ended qualitative questions asking what they liked/disliked as well as when participants got discouraged, become motivated, or felt they learned the most. The latter three were adopted from [10]. In addition, we included questions to inquiry about the participants' perception whether they feel they learned something (-3 = strongly disagree to +3 = strongly agree). In total, the evaluation took about 20 minutes and was conducted in-person. Participants were recruited from the local computer science student population (as these are considered the main target audience for the game) via convenience sampling, i.e. the study was announced in class and students could voluntarily take part in it.

Participants. In total, eight students took part in the study, of which five were male and two were female (one participant preferred not to disclose). Six were within their first 5 semesters, one was in the 13th semester, and one did not disclose it. All agreed or strongly agreed that educational games have potential and that they might be able to help them personally to better understand complex topics. All but one (with substantial knowledge) participants indicated to have no or limited knowledge of quantum computing.

Results. Within the available time participants could mostly finish two lessons (of five) with the more knowledgeable participant reaching the middle of Lesson 4. Asked about if they felt that they have learned something about quantum computing, responses were mixed, with four answers falling on the agreement and three on disagreement side. One participant was undecided. Asked whether the game has increased their interest in quantum computing participants rather agreed (N=5) than disagreed (N=2). Again, one participant was undecided.

Asked about what they liked about the game, four participants highlighted the visual style of the game and two participants appreciated the simple controls. In contrast, the main drawback of the game as expressed by five participants was that it has a too steep learning curve and could benefit from including examples of how the formulas work or providing brief explanations about basic matrix calculations. This was also the main cause for the participants feeling discouraged at certain times, with six participants mentioning unclear instructions. Participants' responses to when they were motivated and felt they learned the most were varied but mainly related to the puzzles themselves (being able to solve them without trial-and-error, having multiple attempts at and being able to skip puzzles, getting keys and proceeding to the next level) which suggests that on overall the participants appreciated the embedding of the puzzles within a Rogue-like framework. Based on the results, future developments will focus on easing the onboarding of the players with additional background information and tutorials.

With respect to the miniPXI, participants on average agreed that the goals ($M = 2.0$, $STD = 1.31$) and progress feedback ($M = 2.25$, $STD = 1.04$) were clear, with ease of control scoring slightly lower ($M = 1.5$, $STD = 1.77$). Challenge ($M = 0.5$, $STD = 1.60$) and Audiovisual Appeal ($M = 1.25$, $STD = 1.83$) scored lowest, but still on the positive spectrum, across the functional constructs. The low score on challenge is likely caused by the steep learning curve as mentioned by the participants. As noted above, while the visual style was liked by five participants, two rated it negatively on the miniPXI. With respect to psychosocial constructs, curiosity ($M = 2.13$, $STD = 0.64$), immersion ($M = 2.50$, $STD = 0.53$), and autonomy ($M = 1.63$, $STD = 0.92$) scored highest, pointing to participants appreciating the embedding of the learning content in a Rogue-like game which offers different possibilities to proceed and explore the levels. Meaning ($M = 1.25$, $STD = 1.39$) scored lower which could be explained by the topic not directly relating to their studies. Mastery scored the lowest ($M = 0.00$, $STD = 1.51$). In addition to changes to the learning curve we will thus focus on providing further encouraging feedback to ensure continued engagement with the game and avoid player churn early on.

5 CONCLUSION

In this work, we introduced *QRogue*, an educational quantum computing game mainly for computer science students. Unlike existing games (e.g., *Quantum tic-tac-toe*, *Quantum Minesweeper*, and *Quantum Odyssey*), the proposed game can not only provide an intuition for the fundamentals of quantum computing but also accurately describes the underlying math. We showed how this is implemented by using puzzles and how the environment surrounding them can support *QRogue*'s teaching process with analogies to superposition, measurement, and entanglement. This way it can help to provide an intuition and aid users in designing algorithms mathematically.

Although our initial user study of the game showed mixed results in the effectiveness of its teaching capabilities, embedding it into a Rogue-like environment was overall well received. As a next step, further development will focus on flattening the learning curve, e.g., by extending existing tutorials and providing helping material for the assumed prior knowledge (i.e., basics of matrix-vector multiplication), and improving player feedback.

ACKNOWLEDGMENTS

This work received funding from the University of Applied Sciences, from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101001318), was part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and has been supported by the BMWK on the basis of a decision by the German Bundestag through project QuaST.

REFERENCES

[1] Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics* 2, 4 (2010), 433–459.
[2] A.I. Design. 1980. *Rogue.* Game [Multiplatform]. Epyx, San Francisco, California.
[3] Armando Bellante, Alessandro Luongo, and Stefano Zanero. 2022. Quantum algorithms for SVD-based data representation and analysis. *Quantum Machine Intelligence* 4, 2 (2022), 1–23.
[4] Gregory T. Byrd and Yongshan Ding. 2023. Quantum Computing: Progress and Innovation. *Computer* 56, 1 (2023), 20–29.
[5] Allan Goff. 2006. Quantum tic-tac-toe: A teaching metaphor for superposition in quantum mechanics. *American Journal of Physics* 74, 11 (2006), 962–973.
[6] Michal Gordon and Goren Gordon. 2010. Quantum computer games: quantum minesweeper. *Physics Education* 45, 4 (2010), 372.
[7] Aqeel Haider, Casper Harteveld, Daniel Johnson, Max V Birk, Regan L Mandryk, Magy Seif El-Nasr, Lennart E Nacke, Kathrin Gerling, and Vero Vanden Abeele. 2022. miniPXI: Development and Validation of an Eleven-Item Measure of the Player Experience Inventory. *ACM Hum.-Comput. Interact.* 6, CHI PLAY (2022), 1–26.
[8] Gulsah Kacmaz and Adam K. Dubé. 2022. Examining pedagogical approaches and types of mathematics knowledge in educational games: A meta-analysis and critical review. *Educational Research Review* 35 (2022), 100428.
[9] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information.* Cambridge University Press.
[10] Laurentiu Nita, Nicholas Chancellor, Laura Mazzoli Smith, Helen Cramman, and Gulsah Dost. 2021. Inclusive learning for quantum computing: supporting the aims of quantum literacy using the puzzle game Quantum Odyssey. *arXiv preprint arXiv:2106.07077* (2021).
[11] Subhasree Patro and Alvaro Piedrafita. 2020. An overview of quantum algorithms: From quantum supremacy to shor factorization. In *IEEE International Symposium on Circuits and Systems.* 1–5.
[12] Peter W Shor. 1994. Polynomial-time algorithms for prime factorization and discrete logarithms. In *Annual Symposium on Foundations of Computer Science.* 124–134.
[13] Katta Spiel, Oliver L. Haimson, and Danielle Lottridge. 2019. How to Do Better with Gender on Surveys: A Guide for HCI Researchers. *Interactions* 26, 4 (2019), 62–65.
[14] Robert Wille, Rod Van Meter, and Yehuda Naveh. 2019. IBM's Qiskit tool chain: Working with and developing for real quantum computers. In *Design, Automation & Test in Europe Conference (DATE).* 1234–1240.
[15] Alwin Zulehner and Robert Wille. 2017. Make it reversible: Efficient embedding of non-reversible functions. In *Design, Automation & Test in Europe Conference (DATE), 2017.* 458–463.