

Quantum "Mens Erger Je Niet!": A Board Game to Demonstrate Quantum Mechanics that can be run on a Quantum Computer

Kyrian Rahimatulla, Alexis Fimeyer, Emirhan Balban, Guo Chuen Liu, Sjoerd Terlouw^a

^a*Delft University of Technology,*

Abstract

A quantum board game is constructed using the classical board game "Mens erger je niet!" as a foundation. New game mechanics are added and the size of the game is reduced. The positions on the board and the winning positions are encoded using qubits. As moves are made, classical logic is used to update the quantum circuit by appending quantum gates. Four bases are used to measure the positions of the pawns. With these bases, the Bell test is recreated. The quantum circuit is optimized to remove unnecessary gates and idle qubits to remain under the 29-qubit limit. The smaller circuit is then transpiled and run on the IBM Sherbrooke quantum computer and a simulator. It is concluded that superposition, entanglement and measurement can be implemented into the game while keeping it playable for the general audience as an educational resource.

Keywords: Quantum game, Mens erger je niet!, quantum mechanics, quantum gates, qubits

1. Introduction

Quantum mechanics is a notoriously difficult topic to understand. Beyond the abundance of mathematics involved in the field, quantum mechanics describes phenomena that are fundamentally different from the macroscopic classical phenomena that humans are intuitively accustomed to. Things like quantum superposition and entanglement are typically not observed at human scales. This poses a challenge for quantum mechanics education, as students not only need to master the mathematics but also reshape their intuition about how the world works physically. To this end, numerous games have been developed that claim to make players accustomed to quantum phenomena, like superposition, entanglement, interference, and collapse, by incorporating them into the mechanics of the game. Tic-Tac-Toe [6], Minesweeper [7], Minecraft, Chess [1, 4], and Checkers have all gotten this treatment.

In quantum Tic-Tac-Toe the players can place marks in superpositions, such that a square can be occupied by several marks. At a certain point, the board collapses and a player has won or the game continues. The game, however, is not very complex and does not show actual quantum mechanics, since it can be played on paper. Quantum minesweeper consists of several minesweeper boards in superposition. The player performs different types of measurements to try and guess the superposition. Unfortunately, the game only shows superposition, entanglement and collapse, of which the last one is the only one that is influenced by the player.

Some of these games, as previously mentioned, like Quantum Tic Tac Toe and Quantum Minecraft, are not meant to be run on a quantum computer. Quantum Checkers and Cantwell's Quantum Chess are designed with quantum circuits in mind, but the practical considerations of running them on today's QPUs have not been taken into account. These games succeed to varying

degrees at demonstrating superposition, entanglement, interference, and collapse. There is, however, one missing piece in this puzzle: what happens when different measurement bases are used? Measuring in different bases is typically used to demonstrate the fundamental difference between quantum and classical uncertainty, like in the Bell test [2, 3], but no quantum board games were found that implemented this feature. The goal of this project is to create a quantum version of the German board game Mensch Ärgere Dich Nicht!¹ that demonstrates superposition, entanglement, and measurement in different bases, and can actually be run on a quantum computer and simulator. This goal will be achieved using the Python package Qiskit [9]. Then, it will be shown that this game is capable of replicating the Bell test.

2. Method and Results

2.1. Quantum Mens erger je niet! and its mechanics

In the quantum version, players do not throw one, but two dice when moving a pawn. The pawn moves forward by the value of each individual die simultaneously. Thus, the pawn is then in a superposition of having moved forward by different values, that both have half of the original probability. In the second turn, the player selects one of the two superpositions to split up further again. This way the number N of positions of a single pawn with s splits is given by $N = 1+s$. It is also possible that players throw the same value twice; in that case, the pawn that moves forward does not split into two new positions and is moved just like in a classical game. The movement described

¹If you are unfamiliar with this game, see the quick introduction on Wikipedia https://en.wikipedia.org/wiki/Mensch_%C3%A4rgere_Dich_nicht

above also works for pawns that are already in a superposition of different positions.

When a pawn, possibly in a superposition, lands on another pawn of a different colour (which can also be in a superposition), a capture is registered. The first pawn is called the **capturer** and the second pawn is called the **captive**. Since the pawns can be in a superposition, the captive could be captured and at the same time not be captured. Now there are two different pawns occupying the same tile. However, as explained in Section 2.2, each position has only one qubit associated with it, where the up state means free, and the down state means occupied. Therefore, measuring the circuit can not reveal which pawn occupies that spot. To avoid this problem, a new rule is introduced called the **no-double-occupancy rule** [4], which states that each position can only be occupied by a single pawn. During a capture move, the capturer becomes entangled with the captive, but the captive is not removed from the board. The capturer then lands on the next free spot in front of the captive, to satisfy the no-double-occupancy rule. All possible moves and the rules associated with them are given in Appendix A.

In the classical game, pawns can overshoot their winning space and move backwards, but in the quantum version, backwards movement is impossible. When a pawn sets foot in its winning space, a measurement will occur regardless of the moves remaining. If after measuring the pawn is indeed inside the winning space, then it will occupy the remaining free spot in the winning space. Another way a measurement can occur is when more than twenty pawns are on the board. A measurement must occur when this condition is met to stay below the 29-qubit limit, as the simulator only has access to 29 qubits. For m filled positions, roughly m qubits are needed, but with entanglement, the final number of qubits needed can be higher. Besides this, the limit is also set to make the game more clear. After a measurement, all superpositions will collapse and only after a measurement captured pawns can be removed from the board.

So, there are two situations which can cause a measurement to occur. When the measurement is triggered by exceeding the twenty positions limit, all the qubits are measured in the standard basis. In the other case, the four players are measured in different bases. There are four bases used for measurement. These are the Z-basis, X-basis, Q-basis and R-basis. The Z-basis and X-basis are already well known, while the Q-basis and R-basis are newly defined bases. Measurement in different bases is achieved by adding a gate in front of a measurement in the Z-basis. For each basis, one of the basis vectors represents a pawn being present, while the other vector is associated with the pawn not being present. This works in the same way as what $|0\rangle$ and $|1\rangle$ mean, where $|0\rangle$ means that no pawn is present and $|1\rangle$ means that there is a pawn present.

The gates added in front of the Z-measurement are described

by the following matrices in $|0\rangle, |1\rangle$ notation:

$$R(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}, \quad (1)$$

with $\theta = \pi/4$ for the Q-basis
and $\theta = 3\pi/4$ for the R-basis

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (2)$$

for the X-basis

The Z-basis does of course not get a gate in front, since this is the basis that the system is measured in. Which colour gets measured in which basis, depends on the type of pawn that triggered the measurement and is given in Table 1. From Table 1 it is clear that the colour which triggers the measurement is always measured in the Z-basis.

Table 1: The bases which are used to measure each colour, depending on the pawn that enters the winning space. Here Z, X, Q and R stand for the Z-basis, Q-basis, X-basis and R-basis respectively.

Pawn\Basis	Z	Q	X	R
Red 1	R	G	B	P
Red 2	R	P	B	G
Blue 1	B	P	R	G
Blue 2	B	G	R	P
Green 1	G	R	P	B
Green 2	G	B	P	R
Purple 1	P	B	G	R
Purple 2	P	R	G	B

The results of measuring all the colours in the standard basis are relatively easy to understand. Superpositions collapse to a single position, and if a capturer collapses to a position coming from the capture position, the captive is removed from the board. But measuring each colour in different bases gives rise to two strange side effects, which makes the game less intuitive (and even more quantum). The first side effect is that capturing becomes a bit stranger: a captured pawn can be removed even if the capturer collapses into a position which did not originate from the capture position. In other cases, the captured pawn is not removed, even though the capturer has collapsed into a position which originated from the capture position. The second side effect is that the players might gain or lose pawns. Consider a pawn with n points in its superposition. After measurement, it could happen that the pawn still occupies these n positions, hence the player has gained pawns after measurement. In the other case, it could happen that all n points have vanished and the pawn has thus disappeared from the board.

If the second side effect is left ignored, then this could very quickly bloat the board with totally new pawns. This comes with all sorts of problems. A player can win without having all their pawns in their winning positions, the board fills up too quickly and it is confusing for players to have several of the same pawns. Therefore whenever multiple positions of a pawn

remain on the board after measurement, only the furthest pawn remains and the rest is discarded. The uncaptured pawns that have disappeared from the board after measuring return to their home base outside the board.

With the ability to measure in different bases, it is possible to recreate the Bell test while playing the game. To achieve this, one has to create a reduced board state in the following sequence:

1. Start with $|110010\rangle$ with Green 1, Red 1 and Green 2 in the occupied positions respectively.
2. Then get $\frac{1}{\sqrt{2}}(|101010\rangle - |100001\rangle)$: Red 1 throws a 1 and a 3. This causes Red 1 and Green 2 to be entangled.
3. Finally $\frac{1}{\sqrt{2}}(|000110\rangle - |000101\rangle)$: Green 1 throws double 3, so it captures Red 1.

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}(|10\rangle - |01\rangle) \quad (3)$$

The last state can be separated into $\frac{1}{\sqrt{2}}(|0001\rangle \otimes (|10\rangle - |01\rangle))$, which is the Bell state shown in Equation 3. By triggering a measurement with a pawn at the winning position, the Bell state will be measured in different bases. For example, if Purple 1 triggers the measurement, the measurement bases will be R for Red 1 and X for Green 2. The Bell inequality is as follows:

$$|S| = |\langle A_1 B_1 \rangle - \langle A_1 B_2 \rangle + \langle A_2 B_1 \rangle + \langle A_2 B_2 \rangle| \leq 2. \quad (4)$$

By triggering the winning positions of Red 2 and Blue 2, the matrices become $A_1 = Q, A_2 = R, B_1 = X$ and $B_2 = Z$, which can be seen from Table 1. This will replicate the Bell test as can be seen in Figure 1.

To calculate the expected value of S , the individual expectation values need to be calculated. In general, the expectation value is given by the following derivation: define \mathbf{A}_i and \mathbf{B}_j as the gates in front of the measurement for basis $\langle A_i B_j \rangle$. Which gates these exactly are, is given by Equation 1 and 2. Now the state in question given by Equation 3 becomes becomes

$$|\psi_o\rangle = (\mathbf{A}_i \otimes \mathbf{B}_j) |\psi^-\rangle \quad (5)$$

The expectation in general is given by $\langle A \rangle = \text{Tr}(\rho A)$, with A as the observable and $\rho = |\psi\rangle\langle\psi|$. Since qiskit measures in the Z-basis, the expectation will become:

$$\begin{aligned} \langle A_i B_j \rangle &= \text{Tr}(\rho_o (Z \otimes Z)), \\ \text{with } \rho_o &= (\mathbf{A}_i \otimes \mathbf{B}_j) |\psi^-\rangle \langle \psi^-| (\mathbf{A}_i \otimes \mathbf{B}_j)^\dagger \end{aligned} \quad (6)$$

Evaluating this gives the following values²:

$$\begin{aligned} \langle ZR \rangle &= \langle XR \rangle = \langle XQ \rangle = \frac{1}{2} \sqrt{2} \\ \langle ZQ \rangle &= -\frac{1}{2} \sqrt{2} \end{aligned} \quad (7)$$

²The calculation itself is tedious, but quite straightforward. It can be found on https://github.com/SjdT1/QuantumEngineering/blob/main/Report/bells_inequality.ipynb

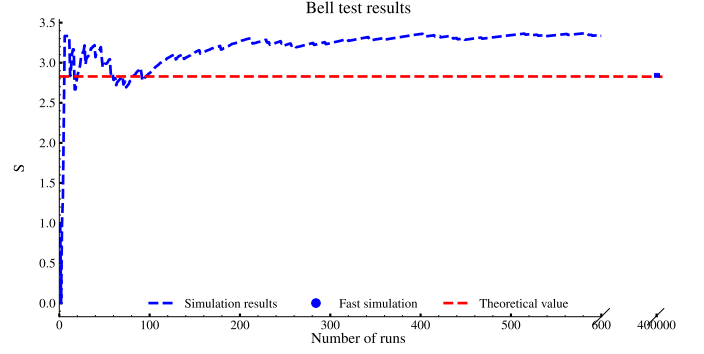


Figure 1: Results of running the bell test. The first few hundred runs show that the value of S is very slowly approaching $2\sqrt{2}$.

This then violates Bell's inequality exactly in the way that is found in the measurements described in figure 1:

$$S = \langle ZR \rangle + \langle XR \rangle + \langle XQ \rangle - \langle ZQ \rangle = 2\sqrt{2} \quad (8)$$

Besides the obvious differences between the classical and quantum game by introducing quantum effects, the game setup has also been changed. First, the board size has been reduced from 40 to 32 positions and the total amount of pawns for each colour has been reduced from 4 to 2. The reason for this is to reduce the complexity and the number of qubits needed to encode the positions of the pawns, such that it can run on a (simulated) quantum chip. This is because real quantum chips have a limited runtime and the local simulator has a limitation of about 29 qubits. More information about these limitations can be found in Section 2.3.

2.2. The quantum circuit

Each qubit represents a position on the board. The state $|1\rangle$ represents an occupied spot and $|0\rangle$ an unoccupied spot. The positions of the board are therefore represented by $|n_0\rangle \otimes |n_1\rangle \otimes \dots \otimes |n_{31}\rangle$, with $n_i \in \{0, 1\}$ and $i \in \{0, \dots, 31\}$. Mainly due to this system, there is a no-double-occupancy rule: the qubits do not keep track of the colour or type of pawn and therefore would be unable to differentiate two pawns in superposition on the same spot. After a throw, the classical code identifies the type of move. It then tells which gates to append to the quantum circuit for that type of move. This process is shown in Figure 2. Usually, these gates span three qubits, but with a capture, the gates usually span more qubits. More information on the specific gates used can be found in Appendix A.

For the winning positions of the pawns, more qubits have to be added. These qubits work the same as the regular qubits, except that only two are required since each player moves at most one pawn into a superposition of two pawns when arriving at a winning position. Due to following this up with a measurement, the eight winning positions can share two qubits between them.

A naive implementation would perhaps even assume that this amount can be brought down to one qubit since moving into the final positions with a superposition is the same as swapping

into the final positions. However, this does not take into account the possibility of one of the superpositions getting into the final positions while capturing another pawn at the same time.

Another possibility is to couple several qubits to a pawn instead of coupling each qubit to a position. Then, each pawn has a binary representation of its position on the board. In practice, this would mean that if red's first pawn is on position 16, its associated qubits would be in $|01000\rangle$. This would require $n \log_2 N = 40$ qubits for $n = 8$ pawns on a board of $N = 32$ positions, instead of just $N = 32$ qubits. The benefit of this method is that pawns could now occupy the same position, but at the cost of the implementation being less general; adding more pawns would increase the amount of necessary qubits way more instead of not influencing the amount of qubits. However, the main problem is that the gates are very complex and counter-intuitive. A standard move gate could span five qubits, and a capture up to ten qubits.

2.3. Running the circuit

The circuit can be either simulated or executed on a real quantum chip. The simulation is performed locally using the Aersimulator [8]. In version 0.16, the simulator can handle up to 29 non-idle qubits with noise on our machine. However, it struggles with circuits that involve a lot of entanglement, but the limits for that scenario are more difficult to define. To still be able to simulate the 34 qubits with noise, the circuit is first optimized to remove unnecessary gates and then the idle qubits are removed from the circuit. The new smaller circuit is then transpiled on a simulated IBM Sherbrooke backend and then simulated using the Aersimulator.

The game is run on the IBM Eagle quantum computing platform, which is the only platform from IBM that is available

for free³. This platform, like all quantum computers that contain more than 51 qubits (as of 2024) [5], does not have complete entanglement⁴, meaning that not every pair of qubits is entangled. In this implementation of Quantum Mens Erger Je Niet, entangled states between distant qubits are created liberally. This means that the generated circuit might try to entangle two qubits that can not be entangled. Fortunately, if the circuit is run on a QPU with a universal quantum gate set and enough qubits, it is always possible to remap the circuit to efficiently approximate it on the architecture [10]. This remapping does not have to be programmed from scratch, as it is already integrated in the Qiskit transpiler. The Qiskit transpiler takes the quantum circuit generated by the gameplay and modifies its topology in such a way that the specified backend (in this case IBM Eagle) can compute it. The various QPUs implementing the Eagle platform have median qubit readout errors on the order of 10^{-2} IBM QPUs. Measures are implemented to reduce the effect of these errors. Whenever a measurement is triggered, the circuit is executed and measured 1024 times. All outcomes that occur less than 0.2% of the time are discarded. Then the final result is selected from the remaining outcomes, taking their frequency into account in a classical weighted random selection.

3. Discussion

This quantum version of the classic “Mens Erger Je Niet!” game demonstrates that quantum-mechanical principles, such as superposition, entanglement, and measurement collapse can be successfully integrated into a board game while keeping

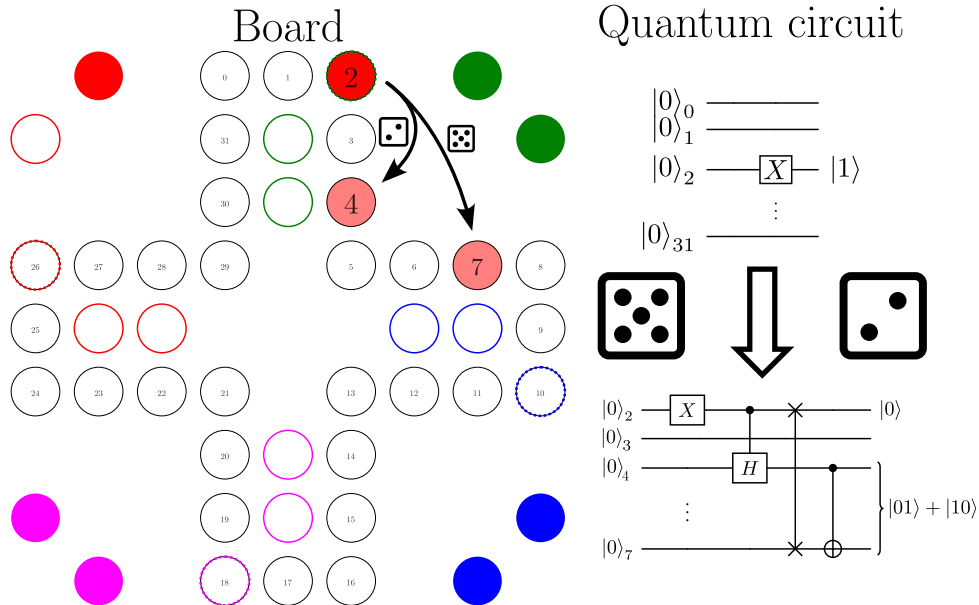


Figure 2: Each position is represented by a qubit. With each move the quantum circuit is updated. In this case, the qubit $|1\rangle_2$ representing the first red pawn becomes a superposition $|01\rangle_{47} + |10\rangle_{47}$.

its original playability. Implementing these quantum features required modifications to the rules, including the no-double-occupancy constraint and additional measurement and capture steps.

From a technical standpoint, encoding each position on the board as a single qubit gives a good balance between gate complexity and hardware limitations. This ensures that important phenomena like superposition and entanglement remain observable. At the same time, it shows trade-offs: the simplifications (such as reducing board size and the number of pawns) show the limited number of qubits and connectivity constraints in current quantum devices.

For further improvements, combining interference effects into the gameplay could showcase more quantum behaviours. Additionally, scalability remains a problem. Implementing richer configurations, potentially adding more pawns or a larger board, would require both hardware advances and algorithmic optimizations, such as more efficient gate placement and circuit transpilation.

Overall, this adaptation shows the potential of “gamified” approaches for exploring quantum mechanics. By showing phenomena that differ from classical intuition, such as superposition and entanglement. Players gain a sense of quantum mechanics’ unique features. Ongoing improvements in both hardware and circuit design will make increasingly more complex quantum board games feasible.

4. Summary and conclusions

This paper presents a quantum adaptation of the board game “Mens Erger Je Niet!” as a tool to demonstrate and explore quantum phenomena such as superposition, entanglement, and measurement collapse. By incorporating quantum mechanics into familiar game mechanics, the game serves as a bridge to introduce these principles to a wider audience. Although the implementation shows the feasibility of integrating quantum logic into gameplay, it also shows the challenges of balancing playability with the complexity of quantum logic. By performing the Bell test in game, it was proven that the game indeed follows quantum mechanical principles. The result of the Bell test is in line with what would be expected based on the theory. Further developments could focus on expanding the scope of quantum interactions, refining rule sets, and optimizing the game’s compatibility with real quantum hardware to further enhance its value and accessibility.

Acknowledgements

We would like to thank Menno Veldhorst for organizing the course and providing guidance throughout this project. Our gratitude also extends to Anasua Chatterjee, Stefano Bosco and Giordano Scappucci for their valuable feedback and support during our meetings. Their insights have been essential to the development and completion of this report.

Appendix A. Move circuits

Dedicating a whole section to explaining how a full game would play out, would become far too large as the game itself is rather complex. Instead, the five fundamental moves of which all possible moves consist out of are given. These moves are called swap, superposition move, new pawn, capture and measure. The type of move is detected using classical code, which then updates the UI and appends the corresponding quantum gates to the circuit. In the figures, the pawn movement and the corresponding circuit is given. It is assumed that the move is started from a mostly ‘clean’ circuit. The different pawns are represented by the patterns inside the positions.

Appendix A.1. New pawn

The simplest move is adding a new pawn to the game. This move can be made when at least one of the dice is a 6. The circuit is updated by just adding an X-gate to the relevant qubit.

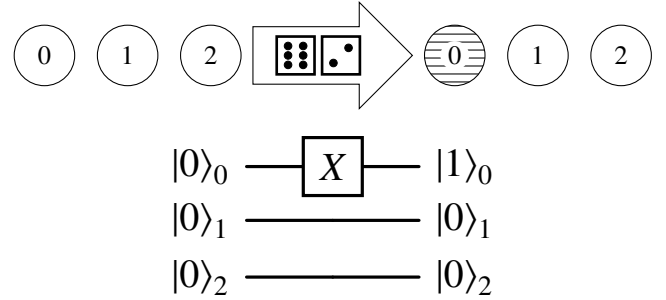


Figure A.3: Addition of a pawn to a position after throwing at least one six by appending an X-gate.

Appendix A.2. Swap

The swap, or direct move, happens when the same value is thrown twice. The pawn just switches positions and the circuit is a standard swap gate.

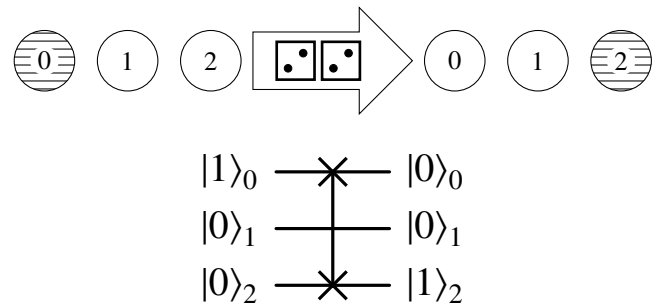


Figure A.4: Swapping a pawn from a position to another position by appending a swap gate.

Appendix A.3. Superposition move

When the dice have two different values, the pawn is brought in superposition. This can also be a pawn that is already in a superposition, it then splits up even more.

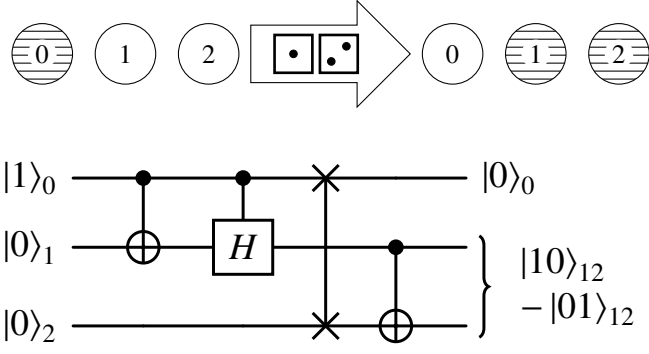


Figure A.5: A classical pawn that goes into superposition by appending the superposition circuit. The amplitudes are not normalized.

Appendix A.4. Capture

When a pawn lands on top of a pawn of a different colour, the original pawn is captured. The capturing circuit in itself is just a controlled not gate, where the controlling qubits is the capturing pawn *and* all the inverted qubits the captive is in superposition with.

The capturing pawn does not go to the position of the captive (due to the no double occupancies rule), but goes to the next available position. This could be the position right after the captive, or could be any amount of positions after it. It does not capture the pawns on the way of finding a new position. This is also included in the example of Figure A.6.

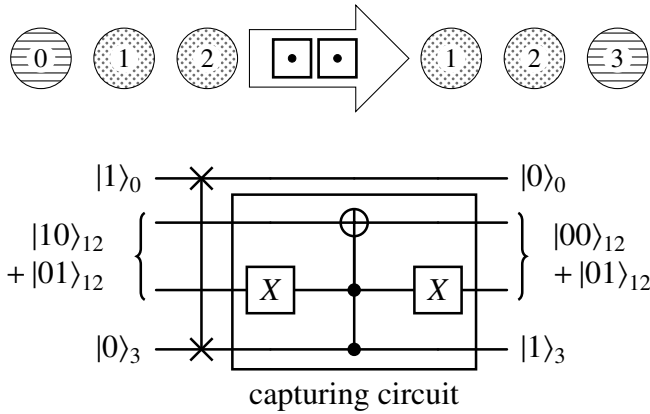


Figure A.6: Capturing a pawn in a superposition using a swap move. The swap gate is *not* part of the capturing circuit. Even though the moving pawn should go to position 1, it goes the next free position, which is position 3. It does not capture pawn 2 when moving to the next free position.

Appendix A.5. Measure

The circuit is measured when a pawn enters its final position, or when the total number of occupied positions (by any pawn) exceeds twenty. The measurements happen for all the pawns at the same time. The measurement bases are dependent on the colour of the pawn that reaches the winning position.

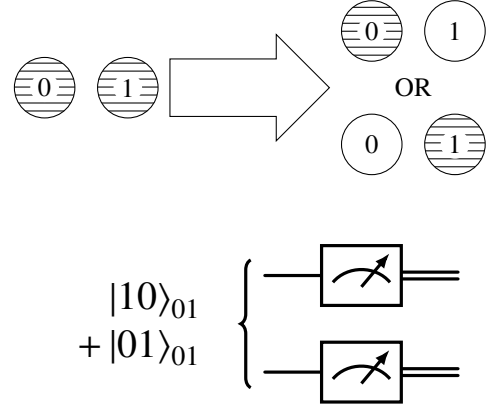


Figure A.7: Measuring a circuit in the Z-basis that describes one pawn in a single superposition, which updates the game accordingly: the pawn collapses into the first or the second position.

Appendix B. Code

For all code references, please visit the GitHub repository at [<https://github.com/SjdT1/QuantumEngineering>]. The README file contains contains instructions to run the game, high-level documentation and some example pictures of the game. A video walk-through of the game can also be found on [YouTube](#) [11].

References

- [1] Selim G. Akl. "On the Importance of Being Quantum". In: *Parallel Process. Lett.* 20 (2010), pp. 275–286. URL: <https://api.semanticscholar.org/CorpusID:207108826>.
- [2] J. S. Bell. "Bertlmann's socks and the nature of reality". In: *CERN* (1980). URL: <https://cds.cern.ch/record/142461/files/198009299.pdf>.
- [3] J. S. Bell. "On the Einstein Podolsky Rosen paradox". In: *Physics Physique Fizika* 1 (3 Nov. 1964), pp. 195–200. doi: 10.1103/PhysicsPhysiqueFizika.1.195. URL: <https://link.aps.org/doi/10.1103/PhysicsPhysiqueFizika.1.195>.
- [4] Christopher Cantwell. *Quantum Chess: Developing a Mathematical Framework and Design Methodology for Creating Quantum Games*. 2019. arXiv: 1906.05836 [quant-ph]. URL: <https://arxiv.org/abs/1906.05836>.
- [5] Sirui Cao et al. "Generation of genuine entanglement up to 51 superconducting qubits". In: *Nature* 619.7971 (2023), pp. 738–742.
- [6] Allan Goff. "Quantum tic-tac-toe: A teaching metaphor for superposition in quantum mechanics". In: *American Journal of Physics* 74.11 (Nov. 2006), pp. 962–973. ISSN: 0002-9505. doi: 10.1119/1.2213635. eprint: https://pubs.aip.org/aapt/ajp/article-pdf/74/11/962/13128226/962_1_1_online.pdf. URL: <https://doi.org/10.1119/1.2213635>.
- [7] Michal Gordon and Goren Gordon. "Quantum computer games: Quantum minesweeper". In: *Physics Education* 45 (June 2010), p. 372. doi: 10.1088/0031-9120/45/4/008.
- [8] Ali Javadi-Abhari et al. *Qiskit Aer Simulator*. 2024. URL: <https://github.com/Qiskit/qiskit-aer>.
- [9] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. doi: 10.48550/arXiv.2405.08810. arXiv: 2405.08810 [quant-ph].

- [10] A Yu Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), p. 1191.
- [11] Sjoerd Terlouw. *Quantum Engineering - Ludo*. Youtube. 2025. URL: <https://www.youtube.com/watch?v=1w0mJI2P1PA>.