Universiteit
Leiden
The Netherlands

Bachelor Computer Science & Economics

Quantum Games as a benchmark for quantum hardware
How suitable are they?

Michaël Kooter
s1826301

Supervisors:
Evert van Nieuwenburg & Jan Krzywda

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl
03/07/2024

Abstract

This thesis aims at answering the question “How suitable are quantum games as a benchmark for quantum hardware?”. In a literature review on benchmarking we found that relevance, reproducibility, fairness, verifiability, and usability are important properties for proper benchmarks. For quantum hardware, in general low-level metrics like number of qubits, fidelity and physical layout are used, while some benchmarks like Qpack, QUARK or Quantum Value have also been proposed. We analyzed the concepts of participatory science and gamification, since both have seen increasing popularity. Using game-related elements to increase motivation and involving the general public to help tackle tasks are becoming more common practices. Then we analyzed the concept of quantum games, ending with an analysis of “Quantum Magic”, a game we specifically developed for this thesis. In the game players aid a wizard by developing potions (quantum circuits) to fulfill specific tasks. We come to the conclusion that quantum games can indeed be used as a benchmark for quantum hardware. However, additional research might still lead to new or different insights, for example through a practical project where the real-world application is analyzed.

Contents

1 Introduction

The field of quantum computing is currently in an interesting spot. While the current era of "Noisy Intermediate-Scale Quantum" or NISQ devices is by some still referred to as "the infancy stage of the technology" [BKY18], an increasingly fast development of quantum devices and quantum computing principles can be seen. Unlike with other recent developments, like the emergence of artificial intelligence, the hardware used is completely new. This poses certain challenges, one of which is how the progress in hardware development can be monitored.

Traditionally, the developments of hardware can be monitored using benchmarks. But how does one set benchmarks for a technology that has not yet been fully developed and has no standardized development method yet? And to perform those benchmarks, does one require very deep technical knowledge? The content of this thesis is aimed at answering those questions. Specifically, the goal is to analyse the usability of quantum games in the benchmarking process for quantum hardware. Can quantum games be used to tackle this task? And if so, how suitable are they?

1.1 Research questions

The main question of this research is: *How suitable are quantum games as a benchmark for quantum hardware?* To help answering this question and give structure to this research, the following sub research questions will be used:

- Which properties of quantum hardware are currently used to create suitable metrics and benchmarks?

  What determines whether a benchmark is suitable?

  What properties determine the quality of quantum hardware and how are they currently being benchmarked?
- How can quantum games contribute to benchmarking quantum hardware?
- What insights can be gained from developing a quantum puzzle game on Quantum Inspires superconducting quantum processor?

Quantum Inspire is a quantum computing platform built by QuTech. It will further be introduced in section 4.1.

1.2 Thesis overview

This thesis is structured based on the above-mentioned research questions. The first two research questions are based on literature study. First we looked at the process of benchmarking and properties of proper benchmarks. Then, we made the step to quantum hardware and the process of benchmarking for quantum hardware. The second research question focuses on the contribution of games to the benchmarking process. To do this, first we analyzed the concepts of participatory science and games and gamification. After that, we transitioned to quantum games, and how the earlier analyzed concepts can contribute to their benchmarking process. The third research question comes with an "experiment" of developing a quantum game and focuses on the development process and substantiates choices based on theory. In that part, we first discussed Quantum Inspire and the hardware properties of their quantum device and finally elaborated on the game that we developed and the theory behind it. The end of the thesis contains a results section, the conclusion and limitations and some suggestions for further research.

2 Which properties of quantum hardware are currently used to create suitable metrics and benchmarks?

2.1 What determines whether a benchmark is suitable?

2.1.1 Benchmarking

Since the industrial revolution benchmarking has become increasingly important for business owners to analyse performance in comparison to competition. [Sta09] It saw a use in the military, where countries would try to capture state-of-the art equipment of others and reverse-engineer their own equipment based of that. [CMH10] In the second half of the last century, it became closely tied to Lean Six Sigma, which led to great use of benchmarking for efficiency analysis, especially in Japan. [GK09] When the classical computer became more prominent, benchmarks started to get used to measure performance of hardware. While many western hardware companies had to close down, because they lost the competition to Japanese firms, Xerox was able to successfully implement benchmarking to survive the competition. [Sta09] Robert Camp, who worked at Xerox wrote the book "Benchmarking: The Search for Industry Best Practices that Lead to Superior Performance". This work caused him to often be referred to as "The father of the modern performance benchmark". [Rob]

Nowadays benchmarking is among others used to set up investing portfolio's, create datasets for data science, measure performance of hardware, devices or infrastructure; or measure company performance. [Sta09]

In the Cambridge dictionary benchmarking is described as "the act of measuring the quality of something by comparing it with something else of an accepted standard". [Cam] Springer defines benchmarking as "evaluating or checking something by comparison with a standard." [Mic14]

Between these definitions, the words measure, compare, and standard stand out. A benchmark can be described as the best performing solution, based on a certain measurement for a specific task. Its usage is to set a standard, to which others can measure their own performance and to set goals to improve this performance both on an individual or an industry-wide level. This often creates a competitive element, where several parties try to improve a set benchmark.

The main goal of organisations when using benchmarking, is to monitor performance on a certain task. Often this is used to measure own performance compared to the market standards or to track technical progress to predict future developments. [KLK20] The insights gained can than be used to inform and steer policy makers or technology stakeholders. [vKAH+15] Example insights might be: "Should an investor go long or short in the stock of company X?", "Are all production lines working effectively, or should (parts of) the process be transformed?" or "How many years will it approximately take before Quantum Computers can be used to tackle everyday problems?"

2.1.2 Properties of proper hardware benchmarks

The Standard Performance Evaluation Corporation or SPEC is a non-profit organization that sets and maintains standardized benchmarks for the newest computing systems. [SPE] They facilitate a neutral stance to establish these benchmarks in fields like CPU's, virtualization, cloud systems, but also high-performance computing. This way, producers have access to a guaranteed fair benchmark to evaluate performance of their hardware and systems. To set a proper bar when establishing these benchmarks, SPEC handles the following criteria: relevance, reproducibility, fairness, verifiability, and usability. [vKAH+15] In his paper, "The art of building benchmarks", Huppler [Hup09] adds these criteria to a scale. On one side he puts relevance and views this as a collective term for among others usability, scalability, and representativeness. On the other side he puts reproducibility, fairness, and verifiability and adds economical. He uses the metaphor of a scale, since he claims that not every property can be present in every benchmark, but there should always be a balance between them.

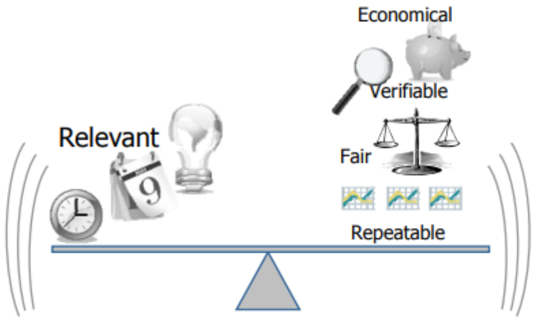

Figure 1: The scale of benchmark properties described by Huppler

To further analyse the criteria, they can be divided like on Hupplers scale. On the one side, are the relevance criteria that are necessary to make a benchmark useful, On the other side are the criteria to make a benchmark viable.

Relevance is essential for benchmarks. A good benchmark should represent a real-world task and measure important features. [Hup09] If these are not the case, there would be no use for people to use the benchmark at all. Another important factor that determines the usefulness of a benchmark is usability. Usability relates to both the technical aspect of a benchmark, as well as the application side. On one hand, benchmark should be able to fit to a system or environment to work. On the other hand, it is important that the method and metrics are understandable to the user, and give a proper representation.[DB19]

On the viability side, the criteria define whether a benchmark is scientifically sound and desirable to establish from a business perspective. From a scientific perspective, it is important to both be able to recreate results and be able to verify how results were generated. [BLW19] In 2016, it was discovered that large parts of computer science empirical research results could not be replicated. [CP16]This so called "replication-crisis", stressed the need for reproducability in scientific work. Besides that, fairness is important to ensure measurements are unbiased and give similar results when different systems are compared.[DB19]

2.2 What properties determine the quality of quantum hardware and how are they currently being benchmarked?

2.2.1 Quantum hardware

To understand the process of developing benchmarks for quantum hardware, one first has to have a basic understanding of quantum hardware itself. Quantum hardware is inherently different from classical computing hardware, due to it being based on quantum principles. While these principles enable great possibilities, like allowing different ways of transferring data or greatly speeding up calculations, they involve strict requirements for the design of quantum hardware.

Classical computation uses bits to store information. These classical bits can be either in state 0 or 1. In quantum computation special quantum bits, or qubits, are used to store information. They are always in a state that is neither 0 nor 1 until they are measured. Instead, they are in a superposition of the two states: $\big|\psi\big\rangle = \alpha\big|0\big\rangle + \beta\big|1\big\rangle$. The symbols $\big|0\big\rangle$ and $\big|1\big\rangle$ correspond with the classical values 0 and 1. The complex numbers α and β relate to the probability of measuring 0 and 1, where $|\alpha|^2$ is the probability of measuring 0 and $|\beta|^2$ the probability of measuring 1. Some physical properties of qubits are that they exist on an atomic level and they need to be strictly separated from any outside influences. Often qubits are created within systems in near absolute-zero temperatures.[Loc15]

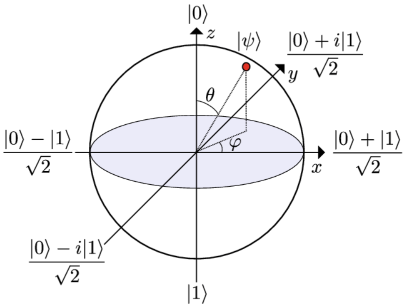

Figure 2: A qubit, represented by a Bloch Sphere.

Besides qubits and superposition, entanglement is another quantum principle that influences the design of quantum hardware systems. Where bits in classical systems are fully independent, qubits have the possibility to entangle with each other. When two qubits are entangled, the state of one qubit becomes directly related to the state of the other, regardless of the distance between them. [Loc15] A two-qubit state can be described by the state vector: $|\phi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle$. For all the subsets of states, where $|\phi'\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ holds, a measurement of the second qubit will give the same result as the measurement of the first qubit, independent of the measurement basis. Such correlation between two qubits is known as quantum entanglement, while specifically this state is known as one of the "Bell states".[NC10]

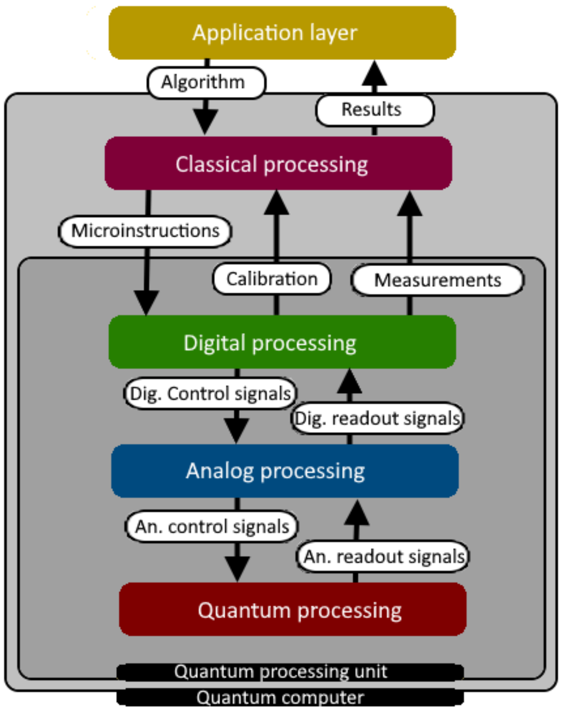

Figure 3: A possible structural layout of a quantum device

The combination of superposition and entanglement allows for the processing of many states at the same time. This is called parallelism. In classical computing, operations can only be performed sequential or to some extent in parallel when using several CPU cores for different tasks.[Loc15]

Finally, another important difference is with error correction. All computing devices, both classical and quantum, are prone to errors in computation. In classical computation, they occur rarely and relate to random bit-flips. Causes could for example be fluctuations in electricity levels or hardware defects. These can often be mitigated by assigning a few extra bits that flip if an error has occurred during computation. In quantum computing, errors are a much bigger problem. Since qubits exist on an atomic level and are often just isolated photons ore electrons, they are extremely sensitive to outside influences, also known as noise. Quantum error can influence systems in several different ways, and is currently one of the main difficulties in the process of developing quantum hardware.[RK21]

While classical computers are based around the usage of semi-conductors, there is no single defined way to construct quantum computers yet. To be able to trap and manipulate qubits in a way that all properties described before are present, specialized hardware is required. Di Vincenzo [DiV00] defines the following requirements for quantum hardware: Firstly, there needs to be some sort of scalable quantum register where quantum states can be stored. In the register, qubits need to be prepared in a basic state. The quantum system needs to remain coherent for long enough to perform logic operations and to do so, a high fidelity gate set is required. Finally, there must be a possibility for (a part of) the register to be read out.

In figure 3 a schematic drawing is shown, of how different elements within a quantum computer can be organized. (Based on [Ver20]) Currently there are many different ways, in which researchers try to apply these elements into a quantum computer with enough qubit capacity to make real-world calculations. Some of the most promising designs are: superconductors, ion traps, neutral atoms, photons, semiconductor spins or NV centers in diamond. [WDE+23]

To conclude with some of the main challenges of developing benchmarks for quantum hardware, one can take a look back at the matters covered in this paragraph:

- The diversity of quantum hardware makes it hard to build general benchmarks fit for every kind of system. This causes shortcomings to the usability and fairness properties of benchmarks.
- The current state of quantum hardware is not good enough to process calculations of real-world applications yet. Although the development is progressing rapidly, benchmarks on the tackling of actual computational problems is still hardly possible.
- The effects of hardware noise on quantum computation is not yet fully understood and its effects can not yet be completely mitigated or prevented. This causes issues with scalability and reproducability of benchmarks. [RK21]

2.2.2 IEEE Framework

In line with the difficulties mentioned in the last paragraph, Blume-Kohout and Young wrote the technical report “Metrics and Benchmarks for Quantum Processors: State of Play”, where they point out some problems of creating benchmarks in the current state of quantum computing. Since quantum computing is a technology that is at most in its infant stage, a lot is uncertain. This makes it impossible to create relevant and cost-effective benchmarks, at least until quantum supremacy has been convincingly proven. Besides that, benchmarks have to be created using very small hardware devices, but the big computers do not exist yet. For that reason it’s impossible to exactly know the characteristics of actual quantum computers. Simulations can predict them to a certain degree, but they have too little computing power to come to a real conclusion. Also because of the many different approaches for hardware, it is currently very hard if not impossible to develop benchmarks that are fair to use on any system. [BKY18]

The Institute of Electrical and Electronics Engineers, or IEEE is "a professional organization dedicated to advancing technology for the benefit of humanity". [IEE] Stimulated by research from the field, like that of Blume-Kohout and Young, they held a meeting with stakeholders from industry, academics and government in 2018. Here a framework was discussed with regard to benchmarks of quantum computing. The goal was to come up with guidelines of how to track technological progress, enable communication and stimulate the development of quantum computing from current technology into fault-tolerant, universal quantum computers.

In the framework, the IEEE supports the defining and developing of metrics and benchmarks for quantum computers. [Fra] The key message of the framework is to be careful around the establishment of benchmarks, since doing so in a too early stadium might limit the development of the technology. Instead it is proposed to publish metrics to characterize features of quantum computing devices. In doing so, a clear separation between technological layers (part of the quantum computing device or system) and use cases (categories of designed purposes for quantum computing devices) can be made. This gives better insights in the development levels of different aspects of the technology. This allows for focused development while also giving a more complete view of the current state of technology.

The sources used in this section are somewhat dated (2018) and developments to quantum computing have continued since. Nevertheless it is still included, as it gives a view on the development of the technology that is important to regard when examining benchmarks for quantum hardware.

2.2.3 Current metrics and benchmarks

Currently, there are several different ways in which companies track the progress of (their) quantum hardware. The first of which are general performance metrics.

Metrics are single aspects of a quantum device that get measured. These are low level known as "low-level benchmarks". Examples are qubit count, qubit stability, qubit coherence, and gate fidelity. While these metrics do give some insights on the performance of specific parts of a device, they are not suited to directly predict application performance. [WDE+23][FRH+22] They might be sufficient, depending on the purpose of the benchmarks. For an in depth analysis on performance on certain applications however, there are too many parameters that can vary and depend on one another.

This led to the development of more complex benchmarks, where for example a correlation between several metrics is regarded and/or more parameters are accounted for. IBM, one of the bigger companies in the sector of quantum computing, proposed the quantum volume metric. Instead of using details, this metric uses the effective error rate ϵ_{eff}. This is the equivalent per-gate error rate that causes the same overall error rate. The quantum volume, $V_Q = \max_{n' \leq n} min \left[n', \frac{1}{n' \epsilon_{eff}(n')}\right]^2$, is based on the lower limit of number of qubits n and the achievable circuit depth $d \simeq 1/(\epsilon_{eff})$ required to reach reasonable fidelity to the correct answer on an algorithm. It quantifies "the space-time volume occupied by a model circuit with random two-qubit gates that can be reliably executed on a given device." [BBC+17]

Li et al. developed a low-level open-source benchmark suite for current quantum devices. The QASMBench suite, contains many commonly used quantum routines and kernels from varying domains, devided into small-scale, medium-scale, and large-scale benchmarks, based on the number of qubits used. The metrics analysed in this benchmark are circuit width and depth, gate density, retention lifespan, measurement density, and entanglement variance. [LSKA22] The approach mainly focuses on hardware-related performance, while some application relevant circuits like Quantum Approximate Optimization Algorithms are included.[FRH+22]

QUARK is a proposed application-based benchmark based on calculation of robot paths and vehicle optimization. The developers claim, since these are real-world problems where quantum computers might come to use, analyzing the performance on these tasks gives good insight on relative performance on real-world tasks. The performance is analyzed based on the TTS, which stands for the end-to-end time required to obtain a solution. This calculated as $TTS = T_{mapping} + T_{solver} + T_{reverseMap} + T_{processSolution} + T_{validation} + T_{evaluation}$. The architecture, as can be seen in figure 4, is modular and based on the separation of concerns design. This makes the application (somewhat) universal, allowing it to run on quantum hardware devices of different designs.[FRH+22]

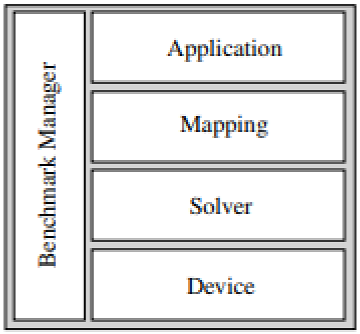

Figure 4: Quark mapping

A final example is the proposed Qpack benchmark. This benchmark is based on a combination of different metrics: maximum solvable problem size, required runtime and achieved accuracy. This benchmark analyses Quantum Approximate Optimization Algorithms for among other the max-cut problem and the traveling salesman problem.[MAAM22]

3 How can quantum games contribute to benchmarking quantum hardware?

3.1 Participatory Science

In participatory science (or citizen science), members of the public, regardless of situation or background, are motivated to actively participate in a research project. By doing so, large tasks that require human knowledge or cannot be fully automated can be completed in a reasonable time without requiring scientists to do all the hard work themselves. [VLZC+21][Hak15]

A similar concept that has grown large over the last decades is crowdsourcing. Here, the purpose is to collect data from participants out of the general public. [Waz17] CAPTCHA is a good example that everyone knows about, but maybe not in the sense of crowdsourcing application. While it is widely known as a security step when logging in, CAPTCHA also provides image annotations based on the input of users. This has greatly aided the image retrieval process and enabled the creation of classified image databases like ImageNet. [MK22]

Thanks to the rise of widespread availability of technology, both participatory science and crowdsourcing saw an increase in usage. Besides offering an easy way for users to participate, apps or websites allow data collection on a bigger scale and real-time data provision. [VLZC+21] An example of this, is the yearly Dutch bird count organized by the Dutch Bird Protection. [Voga] In 2020 the new record number of participants was set at 91.000. In 2021, it had to be organized fully online due to the Covid pandemic while also gaining more popularity thanks to social media. That year, the record was smashed by counting 198.000 participants, more than double the number of the year before. [Koo] In the following years the numbers remained higher than before the transition, with 170.000 and 140.000 respectively. [vK][Vogb]

According to Vohland et al. [VLZC+21] participants can provide data using several methods. Surveys can answer questions about specific topics; spotting allows contribution of (map-based) observations like in the bird count; sensing can be used to provide sensor data like heartrates; image and video classification contribute like with the CAPTCHA example; or gaming allows people to generate data by playing (competitive) games. This last method fits in the trend of gamification of certain tasks. The next paragraph will give a more in-depth explanation of this concept.

So, researchers or organizations can use participatory science to have a relatively cheap way to generate data, and to tackle otherwise time- and/or resource-consuming tasks. Besides that, it gives participants a sense of connection and creates public understanding. It also allows participants to learn about or gain experience with the tasks they are completing. [VLZC+21][Hak15]

On the other hand, there are some concerns. Since participation is voluntary, the sample "selects itself", so the generalizability might be questionable. Also, the intentions of people cannot be measured, so there is no way to stop malicious participants. Finally, the laws around protecting sensitive data have made the process harder, since better protection is required. [Waz17]

All in all, while having a background in social or medical sciences, participatory science seems to become more popular in other scientific fields as well. [Hak15] On the downside it has the issue of sensitive data protection or that the data generated might be inconsistent or limited. But participatory science can offer a cheap alternative to AI or help fill in its findings with definitions. Furthermore, it can be used on other tasks that require human intelligence and it can create more public understanding around research.

3.2 Gamification

Deterding et al. [DDKN11] define gamification as: “the use of game design elements in non-game contexts”. Over the last two decades, in different sectors of society, a trend can be witnessed where tasks or activities are made more interesting by implementing game-related elements. This can be loose elements like rating performance with a social-credit score, earning badges or rewards upon completing tasks or “levels” being added to business software. [DDKN11] However, sometimes tasks are even completely transformed into a game, like how city-escape puzzles are used to explore cities, or how Pokémon GO was used to collect millions of dollars’ worth of location data. [Jin17]

To analyse how gamification benefits the performance of tasks or the actors, first games themselves will be analysed. Games can be described as competitive environments with a general structure and prescribed rules, where players aim to reach a certain goal. Reeves and Read [RR09] describe ten “ingredients of great games”. They mention these ingredients to be: Competition; Reputation, ranks and levels; Time pressure; Teams; Environment; Narrative Context; Self-representation; Marketplaces or economics; and Feedback.

Of these “ingredients”, not all have to be present to make a good game. While most (if not all) of them also apply to real-world situations. This shows, there are no general game elements that explain the success of gamification. Deterding et al. [DDKN11] claim that game elements are the characteristics of a certain game. While they might be different for individual games, a few characteristics can be found in almost every game. While the implementation can greatly differ between games, it is these characteristics that can explain the success of gamification.

Competition – Whether through direct tests of strength with rivals or some ranking system measuring individual performance, competition motivates people to perform. Competition is a powerful method of motivating since it can trigger forms of intrinsic motivation like pride (when outperforming an opponent) or mastery (when practice causes a player to perform better than before) [RD00].

Rules or structure – Caillois [Cai58] claims that these distinguish “game” from “play”. The latter, he describes as uncertain and free but also unproductive and make-believe. While limiting freedom with constraints, rules and structure do help us be productive. [Cai58] Even the most sandbox games do follow certain rules or structure. For example, in Minecraft a player can find specific blocks in specific biomes (structure) and if the player’s health drops to zero, they die (rule). This on itself sets certain goals for the player and steers them to productivity. For example, to prevent dying the player needs food and shelter so they might start building a home or find a source to obtain food from.

A clear goal – Games cannot exist without a clear goal for the player. Through a goal a player knows what they need to do and this way they can look for a way to how to reach this goal. A clear goal sets direction, allowing the player to focus their efforts and divert resources accordingly. [LL02] On top of this a clear goal allows for measurement and feedback. These allow for setting benchmarks and are essential to improve performance. Also, a clear goal delivers a source of purpose for the player, which is also a form of intrinsic motivation [RD00].

Rewards – While being a source of extrinsic motivation, rewards are a great way to stimulate people, especially for the short-term.[RD00] Often, rewards are given based on reaching a goal or beating competition. The type of reward can differ greatly, from a funny animation to a cash price. Since humans like to earn things and to get confirmation they are doing well, rewards often motivate to keep on performing, or put in a little extra effort is that leads to a greater expected reward. [Hid15]

The extra motivation to tackle tasks, learn skills and improve results, that comes with these characteristics has driven the process of gamification. [Ham19] Combined with the ideas of participatory science, this leads to the field of citizen science games. These are games that allow the public to perform a certain task that is completely covered behind the mechanics of a standalone game. Seth Cooper [Coo14] writes about a successful example of this. “Foldit” is an experimental game-based approach for scientific discovery. Players of the game were challenged to solve complex protein-folding problems and eventually succeeded. After this, scientists themselves could use the tool to further work out the problems. Altogether, in gamification games or game elements can be used to stimulate participants to tackle tasks, learn skills and motivate them to give a little extra effort.

Figure 5: Search results for “Quantum” on Steam

3.3 Quantum Games

Quantum games can be described as games that are in some way based around quantum principles. There are several different types of quantum related games. Firstly, there are classical games using (some reference to) quantum principles in gameplay. As can be seen in figure 5, there are currently a lot of these games on modern gaming platforms. Of course, “quantum” is a popular name in pop-culture, but it seemed like most of these games actually made some use on quantum related principles in gameplay. Examples were the use of measurement, quantum gates, or coherence.

Secondly, there are quantum versions of classical games. Examples are Cat, Box, Scissors, Tiq Taq Toe, or Quantum Chess. Here, popular classical games get their gameplay expanded by the introduction of quantum principles in moves or game states. Often these games have increasing levels of “quantumness”, where each level adds more quantum elements into the game. Depending on their coding, these games might already be playable on actual quantum devices. [PPW+24]

In the example of Tiq Taq Toe (or Quantum Tic Tac Toe), on full quantumness adds three mechanics on top of the base game. Dragging a move over two empty squares causes a player to make a *superposition* move. The sign will exist in superposition on both squares, until the board is filled. When the board is filled, the field gets *measured* and all superposition states collapse, causing the sign to be placed in one of the two squares. Players can also make a superposition move on a square occupied by a (superposition) sign of the opponent to create an *entangled* state. When measured, one square of the entangled pair will get one sign and the other square gets the other sign. [VN19] These mechanics can be seen in figure 6.

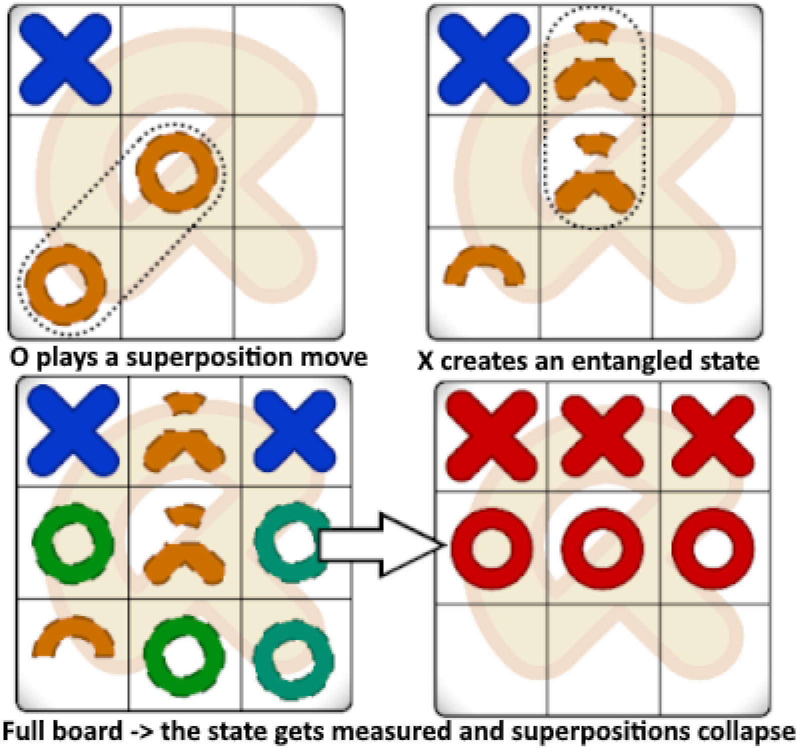

Figure 6: The quantum moves in a game of Quantum Tic Tac Toe

Finally, there are full quantum games, which are games that are completely based on quantum principles (and might or might not be designed to be exclusively playable on quantum devices. In recent years, the number of those games has increased greatly. Probable causes to this increase include increased attention through game jams or hackathons and the recent developments made in quantum devices. [PPW+24] Some examples include: “Blochduel” where two players “attack” each others qubit with gates, to try and set (and measure) its state to 0, while preventing this from happening to their own qubit. [Blo] “Quantum Escape”, where the player collects quantum gates and solves quantum circuits to solve a puzzle. [Quab] “The Photonic Trail”, which offers a quantum optics treasure hunt in a virtual quantum lab. [Pho]

Just like classical games, quantum games can further be classified based on their purpose. There are entertainment games, which have a purpose to purely entertain the user. Besides that, there are

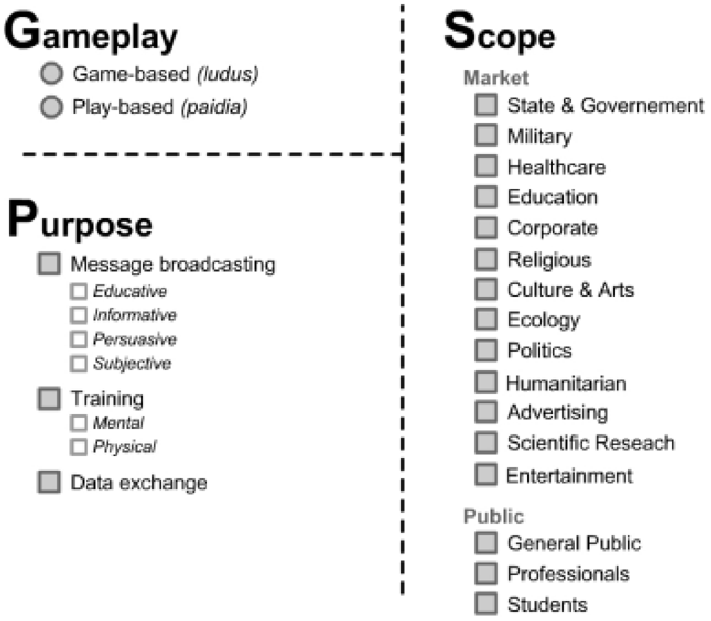

Figure 7: Representation of the G/P/S model

"serious games". Djaouti et al. [DAJ11] introduced the G/P/S model (Gameplay, Purpose & Scope) to classify these games (see figure 7). Games created for/as a result of gamification are serious games. Serious games are often well-suited for participatory science projects, because their purpose and scope can be matched upon the scope and goal of the project. Quantum games are often serious games as well, since those are often used for message broadcasting purposes or training purposes. [PPW+24]

Piispanen et al. [PPW+24] propose three analytical dimensions and define a list of existing quantum related games according to those dimensions. The perceivability of quantum physics measures whether or not a game is notably uses quantum physics in gameplay. The quantum technologies dimension measures if a game uses quantum software or hardware during the development or the gameplay itself. The scientific purpose dimension describes if the game serves a clear purpose (as in the G/P/S model), and if so, which purpose. With these dimensions, it can be defined whether a game is a quantum game, quantum games can further be characterized, and to a certain extend enable the discovery of the purpose of quantum games.

3.4 Relation to benchmarking

To start explaining how this theory can be applied to the benchmarking of quantum hardware, a short demonstration of an existing game will be discussed. In 2018, Dr James Wootton, researcher of quantum computers at IBM, published a document at Medium [Woo18] where he described a puzzle game to benchmark quantum devices. He developed his game "Quantum Awesomeness" to allow non-professionals to be able to compare quantum devices.

The game has a grid with pairs of dots and labels between them. The solution consists of

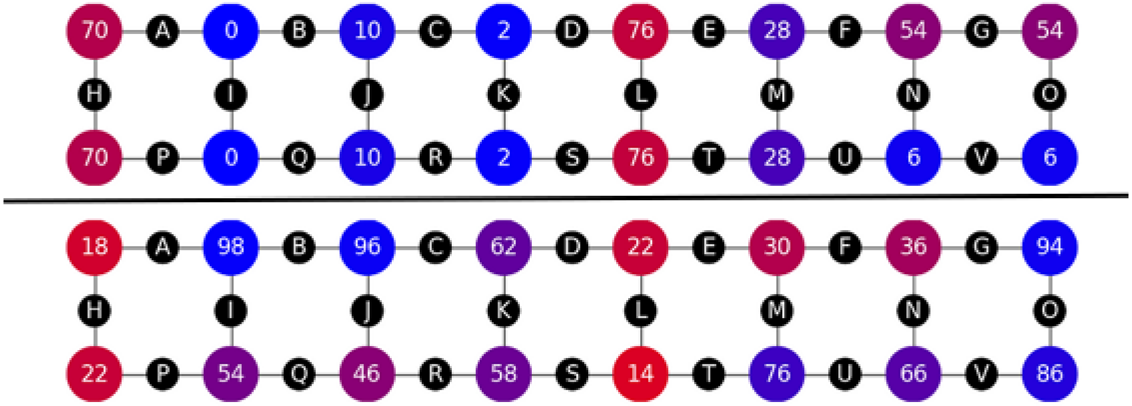

Figure 8: An early, obvious round of Quantum Awesomeness, compared to a later, fuzzier one

labeled pairs with dots of similar numbers and colours. After a few rounds 'fuzziness' causes numbers to drift away, and thus making the game harder. This difficulty is increased until the 'Game Over' condition is reached when it becomes too hard to solve any more rounds. As mistakes increase the amount of fuzziness, technically the skill of the player greatly influences the amount of rounds a player can finish.

The game uses a similar concept to random circuit sampling (RCS) to generate random slices of quantum circuit with an almost even probability distribution. By giving solutions, the player adds new pieces of circuit, while correctly completing a level will make a previous part undone. After enough mistakes, the quantum circuit will become too big, making it unable to be ran any further and creating the Game Over condition.

This simple game creates some interesting metrics to measure the quality of a quantum computer. Firstly, since each dot represents a qubit, bigger quantum processors can generate more complicated puzzles. So, a device that is able to generate sophisticated puzzles, will also be able to run more complex programs (metric = qubit count). Secondly, when quantum circuits get bigger, they are more prone to noise. More natural errors cause more fuzziness in the program. So, the better the system is at error mitigation, the further a player can come within the game (metrics = several fidelity measures).

Based on this example, several insights can be gained. As discussed earlier, benchmarking quantum hardware is often a complicated task that requires specific knowledge. The gamification of this task, allows for a much broader public to (subconsciously) help to tackle this task. Since playing the game automatically generates metrics of hardware performance (qubit count of a device and several fidelity measures), the game can on itself be classified as a simple, low-level benchmark. While the game is not necessarily coined as a citizen science project, the general public could theoretically play and, by doing so, aid with the purpose of generating performance metrics of quantum devices.

4 What insights can be gained from developing a quantum puzzle game on Quantum Inspires superconducting quantum processor?

This section focuses on the experimental part of this research, where we created a small puzzle game. The goals for developing this game were to deliver something that is actually a game, it should benchmark quantum hardware in some way, and ideally it should also be based upon quantum principles.

4.1 Quantum Inspire

Quantum Inspire is a quantum computing platform developed by QuTech. QuTech itself is a collaboration between the Delft Technical University and TNO, an independent statutory research organization in the Netherlands, where research in the fields of quantum computing and quantum internet is conducted. Through Quantum Inspire, QuTech aims to provide a way to let users perform quantum computations, give users insights in the principles of quantum computing, and give users access to a quantum computing community.[QuT18] Within Quantum Inspire there are several full-stack quantum systems available. Full-stack means that the systems contains all of the layers of quantum chip hardware, classical control electronics, quantum compiler and software front-end.

The software front-end is usable via a web interface, the QI online editor. Programming can also be done using the Quantum Inspire software development kit, which has a Python API interface and back-ends for both a ProjectQ and a QisKit framework. The programming language of the software front-end is cQASM, which is a proposed common syntax definition for QASM that aims at interoperability. [KGA+18]

For chip hardware, Quantum Inspire offers the Spin-2 and Starmon-5 quantum processors. The first is currently unavailable, because it is being upgraded from a 2-qubit system to a 4-qubit system. It is based on single electron spin qubits in a double quantum dot in isotopically purified silicon-28. The Starmon-5 QPU consists of five superconducting transmon qubits in a X-configuration. Besides the quantum hardware devices, Quantum Inspire also offers several QX-simulators. [QuT18]

4.2 Starmon-5 quantum processor

For the development of the quantum game the Starmon QPU was used. [DCD24] This processor is praised for its high qubit connectivity and fast two-qubit gates. All five transmon qubits have a 7-port connectivity. Each qubit has a microwave-control line to allow for single-qubit gates and a flux-control line to allow for two-qubit gates. Dedicated bus resonators connect each qubit to its nearest-neighbor pair, with Q2, the centre transmon being connected to all corner transmons. Every qubit has a readout resonator, and based on the chip layout Q0, Q2, Q3 and Q4 connect to a single readout feedline, and Q1 connects to a different readout feedline. Figure 9 shows the physical layout of the QPU.

The Starmon-5 QPU has a native gate set including 14 single-qubit gates and 1 two-qubit

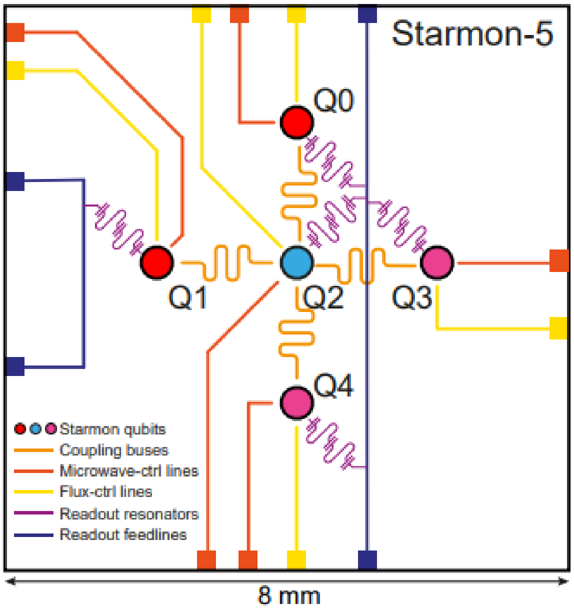

Figure 9: Physical design of the Starmon-5 quantum processing unit

gate. The native single-qubit gates are I, Z, S, S†, T, T†, X, X90, mX90, Y, Y90, mY90, $R_x(\theta)$ and $R_y(\theta)$. $R_z(\theta)$ and H are supported non-natively. $R_z(\theta)$ is realized as $R_y(-\pi/2)R_x(\theta)R_y(\pi/2)$ and H is realized as $R_x(\pi)R_y(\pi/2)$. The only native two-qubit gate is CZ. CNOT and SWAP are non-natively supported. Native measurement is in the computational Z basis. Readout in X and Y bases are supported non-natively using rotations.

In the web interface, a table is given with some low-level metrics for each of the qubits. These are:

- Initialization Fidelity (F_{init}) in %, given by the area ratio of the dominant gaussian of a double-gaussian fit from a histogram of analog outputs of single-shot readouts with the qubits initialized.
- Readout Fidelity ($F_{R/O}$) in %, given by the corrected assignment fidelity of single-shot readouts that determine the probability of properly declaring the right measurements outcome.
- Qubit Relaxation Time (T_1) in μs, extracted from a standard sliding-π pulse experiment.
- Qubit dephasing Time ($T_{2^{echo}}$) in μs, extracted from a Hahn-echo experiment.
- Single-Qubit Gate Fidelity (F_1Q) in %, obtained by taking the average error per native single-qubit gate, performing single-qubit Clifford randomized benchmarking.
- Two-Qubit Gate Fidelity (F_2Q) in %, obtained by extracting the average error per two-qubit Clifford gate, performing interleaved random benchmarking.

4.3 Quantum Magic

In order to fabricate some practical results in aid to answer the question “How suitable are quantum games as a benchmark for quantum hardware?” a little quantum puzzle game was created during research. This game, named Quantum Magic, is mostly intended to demonstrate possibilities of using an actual game as benchmark and gather insights in doing so. The code is written in Python, using the Pygame package. It uses the software development kit of quantum inspire and connects to their back-end API to make calculations on the Starmon-5 quantum processor.

Figure 10: Left: The introduction introduces the wizard and Nôdd-onh. Right: A still from the main menu.

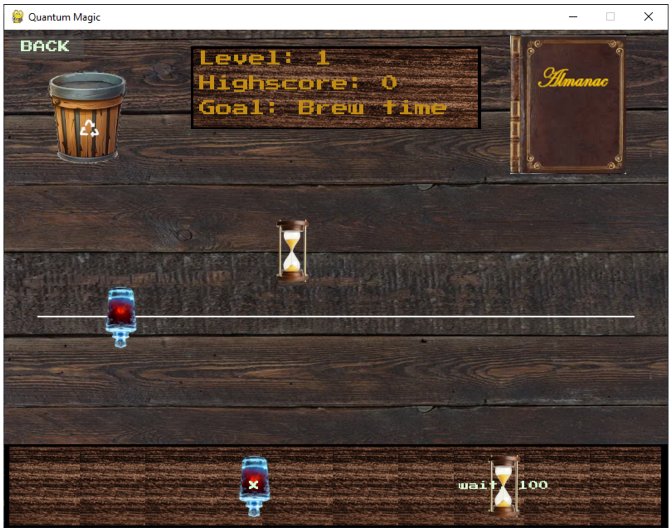

Figure 11: Preparation of a potion recipe in level 1

The story of the game starts with an unnamed wizard trying to stop the evil “Nôdd-Onh the Detangler”. It can be seen how the wizard attacks and kills Nôdd-Onh, but not before the latter uses a spell to destroy the wizards potion recipes. The player is then asked to help the wizard rediscover the recipes. In the main menu, the player has entered the wizards hut, and a boiling kettle and a table can be seen. These are used to run back-end calculations and prepare quantum circuits respectively. Besides that, there are buttons to open a menu and select a different level or quit the game.

In each level the player has a certain goal to reach by “brewing a potion of specific properties". At the recipe table, the player will have a specific collection of ingredients at his disposal, each representing a certain quantum gate. Further there are lines, the assembly area, where a player can prepare a recipe or develop a quantum circuit. *In the current state of the game the goals for the levels have to be communicated with the player better. Also, some feedback mechanism on the performance is missing.* When a player is content with his recipe, clicking the kettle in the main menu starts the brewing process. During this process the circuit is translated to cQASM, sent to the API of Quantum Inspire, and a score is calculated in accordance to the outcome of the circuit.

With each circuit measurement, 2048 shots are done. The result is an probability distribution of outcomes, returned in an ordered-dict. The keys represent the (binary) states that were reached, where the result of every qubit is represented by a bit in a bit-string. Each key is connected to a value, which represents the proportion of the shots that resulted in the state of that key. This documentation of the different shots, can show how often results diverge from the expected result on a certain calculation, and allow calculation with the level of divergence in specific states.

The idea behind the game is that each level focuses on a certain aspect of quantum computing that influences the performance of quantum hardware. The philosophy would be that players can both get to know these aspects themselves, while also producing metrics of the hardware device they are playing on. In the following paragraphs, for each level an overview will be given of the goal, the scientific aspect behind that goal, the theory behind the scoring of that level, and the contribution that level makes to benchmarking quantum hardware.

4.3.1 Level 1

The first two levels of the game focus on generating metrics related to decoherence. Since quantum systems are open systems, meaning these systems can easily couple to things outside of it, over time quantum information is lost from the quantum system to the environment. [Bac03] There are several different forces that cause decoherence. Firstly, due to energy exchanging with the environment, qubits tend to thermalize to equilibrium at the temperature of their surroundings (lattice). This process is known as transverse relaxation. More concretely, in this process the excitation of a qubit to $|1\rangle$ is lost, causing it to fall back into the ground state. The timescale at which this process happens is known as the transverse relaxation time, or T_1 time. [NC10]

In the first level, the goal for the player is to create a semi-excited state. This is the point where the interaction with the environment causes the proportion of shots measured to $|1\rangle$, of a (previously) excited state, to be equal to 0.5. This can be done by adding a Pauli-X (or X) gate and then wait for a certain time. The player will have to figure out the exact time by trial and error.

The X gate is represented by the matrix $\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and equates to a rotation around the x-axis of a bloch sphere by π radians. Since $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ or $\sigma_x |0\rangle = |1\rangle$, this puts the qubit in an exited state. The other gate the player has at his disposal is the wait-gate or idle-gate. This gate causes the quantum system to remain idle for a specified time, which currently is set to 2 microseconds in the game. The player can add several of these wait-gates, to wait for a desired

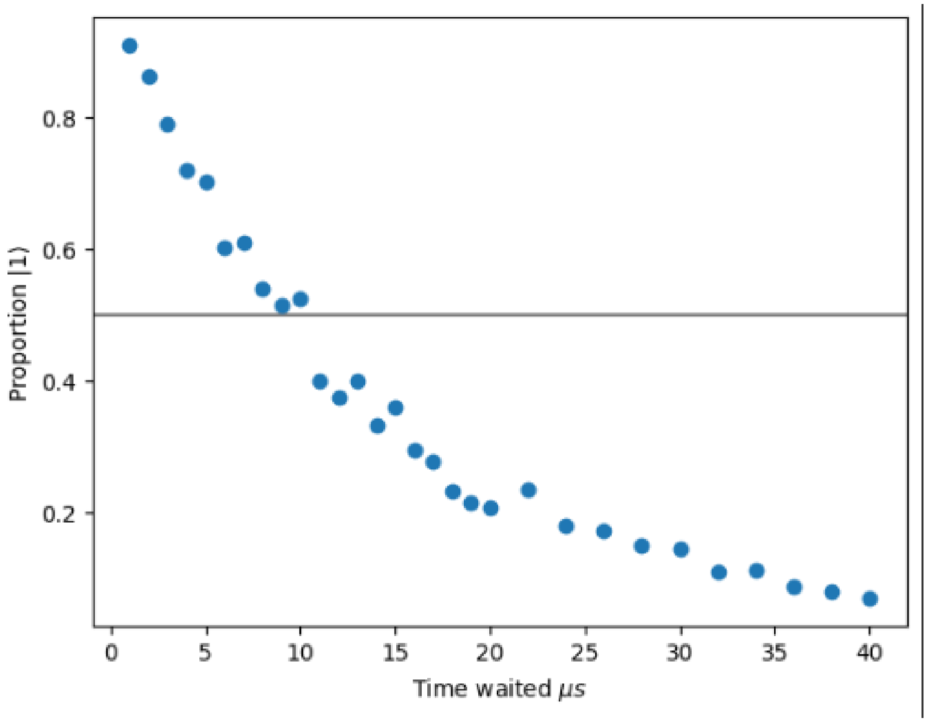

Figure 12: Visualizing T1-time

time τ. During this time, the transverse relaxation over time T_1 will cause an increasing proportion of the shots to measure to 0 instead of 1. Based on the T_1 time metric of the quantum device, the result of a measurement will become $M = \exp(-\tau/T_1)$. [NC10] [Loc15] In figure 12 a plot is made of the proportion of shots measuring $|1\rangle$ related to different waiting-times. The horizontal-line at 0.5 represents the goal of the player, the "semi-excited" state.

The score in this level is based on the closeness to the targeted semi-excited state The player receives an increasingly bigger score the closer the number of shots measuring $|1\rangle$ nears a proportion of 0.5. However, once the proportion of shots measuring $|1\rangle$ drops below 0.5, the score will become 0. In regard to the game, this adds a fail-condition to the level, to which the player will have to adjust its approach. The formula that is used for calculating the scores is: $score = 100000 * e^{-2*log_{100000}*(result-0.5)}$. The corresponding graph of the score distribution can be seen in figure 13.

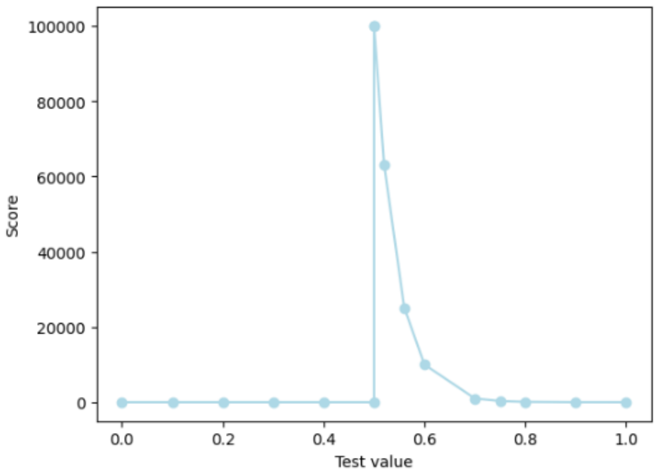

Figure 13: The score distribution in Level 1

In this level a metric of the T_1 time is generated for the quantum device used in the back-end. When a player repeatedly tries to improve the score on this level, a plot like in figure 12 can be generated. The quality of the hardware device is represented by the T_1 time. If a device has a better T_1 time, the player will be able to wait for longer, and might be able to generate more accurate measurements, resulting in values closer to 0.5.

4.3.2 Level 2

In the second level, another source of decoherence is used: the phase randomization or dephasing of states. [NC10] [Loc15] This refers to the decay of a superposition state on the equator of the block sphere. Dephasing, just like spin-relaxation, is a process that constantly causes quantum systems to lose some information, which is here encoded in an oscillating phase between probability amplitudes α and β in $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$. A natural cause for this is the uncontrolled evolution of magnetic and electric fields around the physical qubit. For example atomic level magnetic fields can flip their poles near the qubit. In a perfectly sealed environment, the phase of a qubit oscillates at a constant frequency. However, due to the influence of these external uncontrolled forces, the frequency of the oscillation is not constant, but has some dynamic fluctuations $\delta\omega$, which cause the qubit to lose its phase over time.

There are several forces at work that cause dephasing. Firstly, there are inhomogeneities or low-frequency noise. This causes the imperfect oscillation of a qubits phase. For example, if one would analyze the phase of a qubit from its equator by checking every nanosecond, 100 shots of observing the exact same state for some time, would result in a slightly different final state every time. The time until the phase is completely lost due to this effect is known as the dephasing time. It is called T_2^*-time when the cause of this dephasing is the change of oscillation frequency between shots. This effect is actually reversible, by applying a so called π-pulse (after waiting for time t, applying an X-gate, and waiting for time t again, would cause all observations to be at the starting point again after T_2^{echo}), as can also be seen in figure 14.

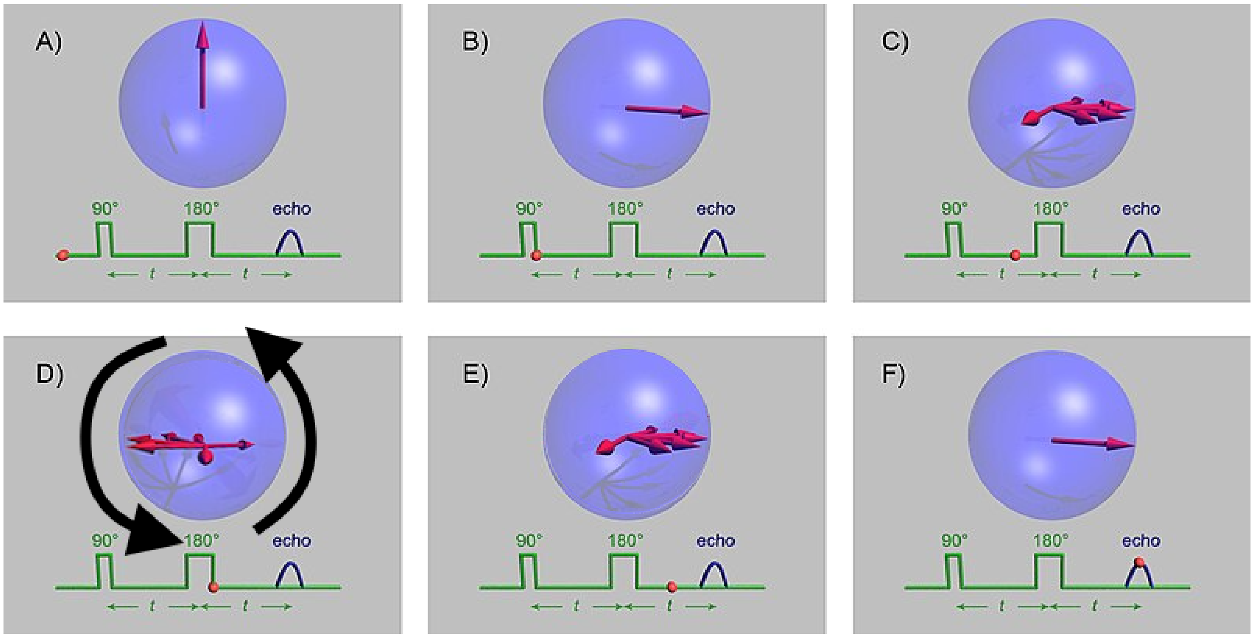

Figure 14: Applying a π-pulse on a dephasing state results in a refocused phase after T_2^{echo}

There is also high-frequency noise, an example hereof is phase randomization due to electric field fluctuations. High-frequency noise causes the frequency of the qubit to fluctuate when it is in superposition. Over time the effect of these fluctuations get stronger to the point where the phase will be lost. The time for this to happen is known as the T_2-time. This effect is irreversible and it is one of the main difficulties during the development of quantum hardware. Since the impact of the low-frequency noise causes quicker dephasing of the qubit, a general rule for computing T_2 and T_2^* is $T_2^* \leq T_2$.

The goal of the player will be to find the time where dephasing causes the qubit to "turn classical" or where the probability of measuring $|1\rangle$ in X-basis becomes 50%. In figure 15 the oscillation of a qubit is visualized using repeated measurements on Quantum Inspire. Over time the oscillations start to dampen, which shows the dephasing of the qubit. *An unexpected effect on Quantum Inspire was that the equilibrium neared a probability of measuring $|1\rangle$ of 25%. It would be expected that the equilibrium nears 50%, but some unknown bug in the hardware caused this result instead. Also different calibrations had strongly divergent T_2^{echo} times, but since the wait-gates have a fixed time, the level might not function as intended.* The goal of the user will be to wait as long as possible before the dampening of the oscillations cause the line to flatten.

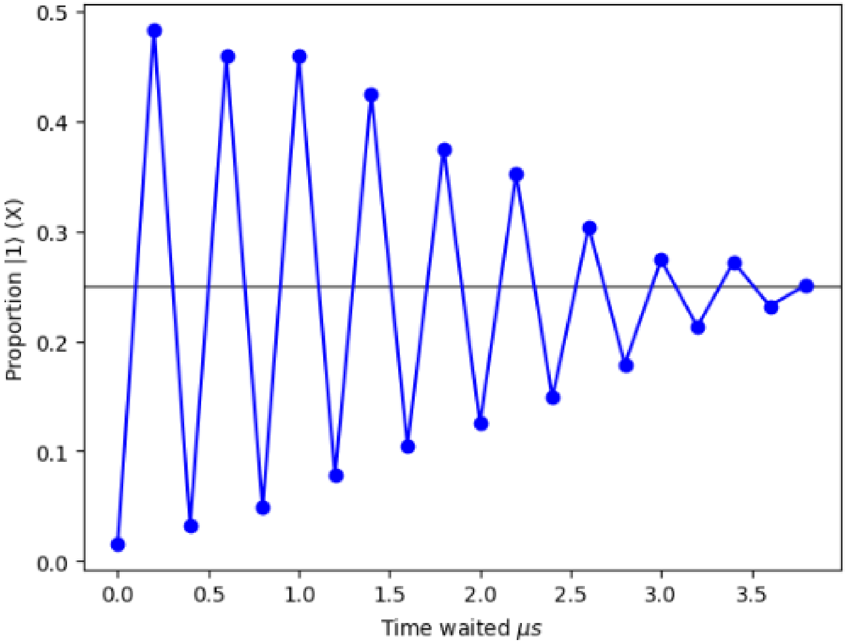

Figure 15: Visualizing the effect of dephasing using repeated measurements

To do this, the player can use a Hadamard gate on the qubit. The Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$ makes a rotation of π on the axis $\hat{x}+\hat{z}/\sqrt{2}$, changing a state $|0\rangle$ into $\frac{1}{\sqrt{2}}|0\rangle + |1\rangle$ or $|+\rangle$ (or when used on $|1\rangle$ it changes the state to $\frac{1}{\sqrt{2}}|0\rangle - |1\rangle$ or $|-\rangle$). $|+\rangle$ is a superposition state in which the phase starts to oscillate. From there, the player once again has the wait-gate at their disposal, to wait exactly long enough before the qubit turns classical. This time however, each individual wait gate only waits for 0.3μs, since the T_2-time is way shorter than the T_1-time.

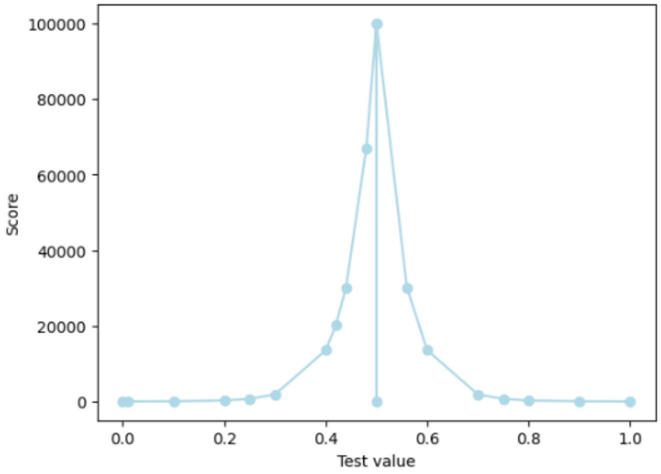

Figure 16: The score distribution in Level 2

In this level the score is based on the closeness of the measurement to probability 0.5 of measuring $|0\rangle + |1\rangle/\sqrt{2}$ (or $|+\rangle$). The player is rewarded if they manage to wait for such a time that the phase is minimal, but not yet classical. The score increases exponentially when nearing a probability of 0.5 for measuring $|+\rangle$ from either side. Would the phase be lost however, then the score will be 0. Once again this adds a fail-condition which the player should try to avoid. The formula used to calculate the score is: $score = 100000 * e^{-20*|result-0.5|}$. The conditional exception is 0.5, where no formula is used, but the score is set to 0. The distribution of scores can be seen in figure 16. *Last minute note: During final testing, it is discovered that the scoring* *mechanism does not account for "lucky guesses", so in practice the current scoring mechanism does not give a good representation of performance!*

The second level generates a metric of the T_2 time of the quantum device that gets used. Once again, a graph is generated based on previous results, so repeated tries result in a graph visualizing the dephasing of the device like in figure 15. The quality of the device is measured by the time a player can wait before the qubit reaches the classical state. In contrast to level 1, the probability in this level oscillates, which adds an additional challenge for the player. If the quality of the device used gets better, the player will be able to wait for longer and probably also allow more accurate measurements, resulting in higher scores.

4.3.3 Level 3

This is the first level that focuses on a two qubit state. [NC10] [OW23] The goal of the player will be to create one of the bell-states, Φ^+ also known as a maximally entangled state. The underlying goals of this level are to both provide a metric for two-qubit gate performance, as well as to introduce the player to entanglement. Since the target state of this level requires only two gates, it has two advantages: it is an easy introduction for the player; and due to the minimized number of gates, this circuit is less prone to quantum noise than longer circuits.

To start working towards the desired state, the player should once again start with a Hadamard gate ($H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$) on one of the lines, to put one qubit in the $|+\rangle$ state. From there, the player does not wish to just wait this time, since their goal is to create a two-qubit state where both qubits measure the same outcome. To do this, a controlled not gate can be used:

$$\text{CNOT} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$$

The CNOT has a similar working as the X-gate, which was shown earlier, but has a "control bit" that determines whether or not the X-gate should be applied. More precisely, on a two-qubit state $\psi = |AB\rangle$, a CNOT with the first bit (A) as control bit ($CNOT_{A\to B}|AB\rangle$) will give the following outcomes:

- If the control bit is 0, no NOT is applied: $CNOT_{A\to B}|00\rangle = |00\rangle$ and $CNOT_{A\to B}|01\rangle = |01\rangle$.
- If the control bit is 1, the NOT is applied on the second bit: $CNOT_{A\to B}|10\rangle = |11\rangle$ and $CNOT_{A\to B}|11\rangle = |10\rangle$

If the player applies a CNOT on the qubits, using the qubit where H was applied as control qubit, this gives the following result: $CNOT\frac{|0\rangle|0\rangle+|1\rangle|0\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. This results indeed in the bell-state Φ^+, which is equal to $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ as well.

The score for this level is calculated based on measurements in the X and the Z basis. This is one of the parts where the back-end is currently tailored to Quantum Inspire. Since the API of Quantum Inspire does not properly handle several measurements in one instruction, the circuit is currently sent to the back-end twice. First, a measurement in the Z-basis is made and the result is stored. Then a measurement in the X-basis is made. When calculating the score correlating to the result, the probabilities of measuring $|00\rangle$ and $|11\rangle$ are added for both the measurement in the X-basis and the measurement in the Z-basis. These sums are multiplied, resulting in the value x. After that the following formula is used to calculate the score: $100000 * \left(\frac{x-0.75}{0.25}\right)^{10}$. If the value of x is below 0.75 however, the score is set to 0. This results in higher scores if both measurements in the X- and the Z-basis have a high probability of obtaining the desired Φ^+ state. The distribution of scores for level 3 can be seen in figure 17.

This level generates a metric on the two-qubit gate fidelity of the quantum hardware device. Because the circuit to reach the desired state is quite simple, there is not much room for player error in the metrics that are created. The only way a player might influence the result is by running the level several times, and hoping for a more accurate calculation of the hardware. While this might obviously be somewhat frustrating for the player, it results in very clear metrics. Since the length of the circuit is only two, other noise will have a relatively low impact on the result. The score gives a clear representation of the performance of the one and two qubit gates on the hardware used. If the hardware delivers high fidelity gates, a high score can be reached. Because of the exponential score increase, the closer to perfection the two-qubit fidelity comes, the quicker the reachable scores increase.

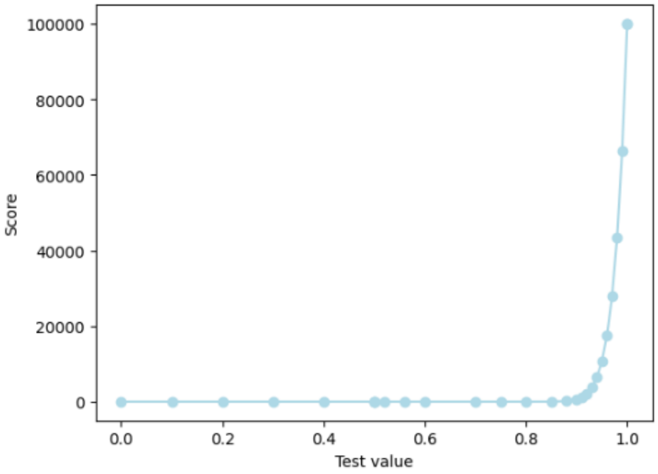

Figure 17: The score distribution in Level 3

4.3.4 Level 4

There has been a start with a fourth level of the game that focuses on quantum teleportation. [NC10] [OW23] [Loc15] While the code has been written for the level to be applied, its working has not been tested, as the back-end API to Quantum Inspire has been offline during the final stages of this thesis. This level would provide metrics for the measurement-fidelity, specifically for the appliance of mid-circuit measurements (MSM) [QuE]. Recent developments in quantum computing saw an increasing importance of MSM. MSM reduces the time qubits need to be coherent and the effects of decoherence because of the shorter time between initialization and measuerment. Besides that, MSM enables error detection and correction during execution; the need for certain quantum gates is removed; and through MSM qubits can be "recycled", since they can be re-initiated during execution after measurement. An important downside to MSM is that measurement might be one of the slowest operations within quantum computing. For that reason fast and reliable measurement is very important, and this level focuses on generating metrics for that.

Quantum teleportation is about the moving of information through the movement of quantum states. An important action to make teleportation work is for the sender to make a measurement amidst the circuit used. In figure 18 a circuit for quantum teleportation is shown. The sender controls qubits q_0 and q_1, where q_0 is the qubit containing the state ψ that will be teleported to the receiver. The first part of the circuit puts the qubits q_1 and q_2 in an entangled (Φ^+) state. Then, a Bell-measurement is made on q_0 and q_1, putting them in one of the four Bell-states before collapsing to one of them through the measurement.

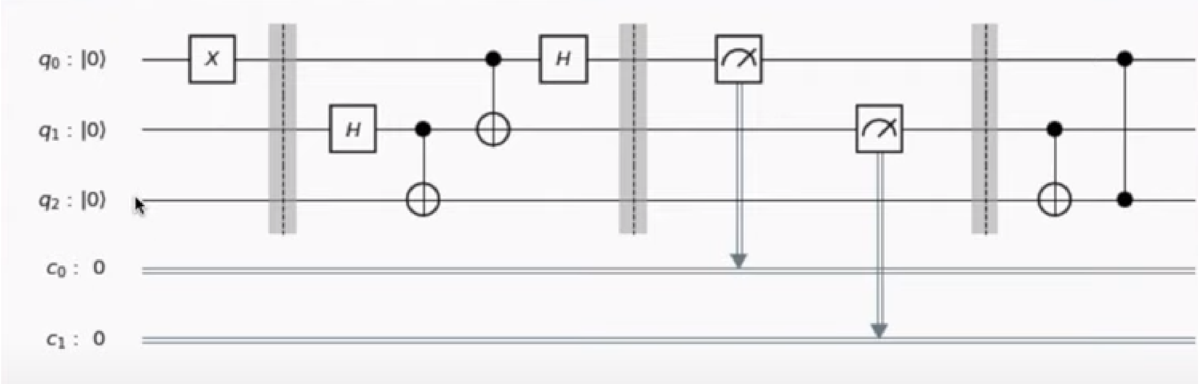

Figure 18: A circuit for quantum teleportation

After this there is only one qubit left: q_2, the receivers qubit. Since their qubit is in a state that depends on which of the four Bell-states was reached previously, some "correction" can be made based on the classical outcomes of the measurements from the sender. If q_0 measured to 1 a CZ-gate gets applied and/or if q_1 measured to 1 a CNOT gate gets applied. This will result in q_2 now containing the ψ state. Because of the earlier entanglement, the receiver only has to obtain 2 classical bits of information from the sender. Besides that no information about ψ is obtained during the circuit, and no other information needs to be transferred.

This philosophy for this level extends upon the last level, since an entangled state is required to be able to perform teleportation. It was however not possible to get this level working inside the game using Quantum Inspire as it did not provide the connectivity required to inter-entangle three different qubits.

The metrics generated would relate to the fidelity of measurement and general gate fidelity. The better the quantum hardware is at performing accurate measurements, the higher the score will be since the rewards increases exponentially when the probability distribution of measuring $|1\rangle$ nears 1. Furthermore, since measurements are currently quite a slow procedure, the quantum states can be vulnerable to the effects of quantum noise. Systems that allow for faster measurements, should therefor also score better since the system gets exposed to noise for a shorter time. The general performance on quantum teleportation can also be regarded as a metric itself. Because quantum teleportation occasionally gets applied in real-world problems, this would be the first level providing metric that is a little less low-level compared to fidelity and coherence times.

5 Results

To determine the results the experimental part, the game will be analyzed using the theory described before.

A classification of Quantum Magic can be made using the Gameplay/Purpose/Scope model.[DAJ11] On the gameplay axis, it falls into the category of game-based games. This is because it gives the users a concrete goal to complete and rules and structures to follow. On the Purpose axis, Quantum Magic could be classified within two purposes. Firstly, there is data exchange in the form of delivering benchmarking metrics. The second purpose is to a certain level to broadcast a message of educative nature. By playing the game, the player can actually gain some insights and knowledge on several aspects related to quantum computing on physical hardware. The scope of Quantum Magic was originally to be playable for the general public. However, there are currently no actual plans to release the game in any way.

Since the game is designed to be fitted for participatory science purposes, it could theoretically be used as a tool to have the general public play it to fulfill the task of generating metrics on quantum hardware, but that does still require some work. One important limiting factor is the general availability of quantum hardware to run on. Currently the game is fitted to one device, because that device was easily available during development. Some adjustments will have to be made for interoperability with multiple quantum devices. Besides that, to allow the general public to help owners of quantum computers acquire metrics of their quantum device, some way of connecting players to quantum devices would have to be realized.

Defining Quantum Magic according to Piispanens three dimensions, would give the following result [PPW+24]:

- Perceivability: yes
- Technical: yes
- Scope: benchmarking

This would qualify the game to truly be called a "quantum game". Besides Woottons game[Woo18] described earlier, this would be the second quantum game with a benchmarking purpose.

The game itself offers a benchmarking suite, that purely focuses on the hardware side of quantum computing. Each level produces a metric of the hardware the game is used on. The metrics on itself are low-level. It would be possible to extend the metrics generated, by adding new levels. These could be used to also generate higher-level benchmarks, by for example developing levels focused on specific algorithms like Shor's Algorithm or Quantum Approximate Optimization Algorithms.

Based on gamification the task of generating benchmarks can actually become a fun experience. There is a competitive element, as the user can try to beat high scores. If the game were to go public, this could be further extended upon by adding some online leader-board or score mechanism where people can share, compare and compete. Another advantage of wrapping the benchmarking process inside a game, like with Quantum Magic, is that there is a clear goal, and the steps can be discovered on the way. This idea of "learning by doing" can be yet another source of motivation to tackle the task, as well as a way to obtain new knowledge for the players.

To give an insight on the usefulness of Quantum Magic as a hardware benchmark, the properties from Hupplers scale [Hup09] can be analyzed. Firstly, the usability is currently low. It could be increased by adding interoperability options, but right now it only works on a single device. The scalability is only present in one way: the game can theoretically be used on noisy intermediate-scale quantum (NISQ) devices and also on bigger, actual quantum computers. However, by design it will still only measure one,two or three qubits, even on much bigger devices. This was designed to consider fairness and usability on different NISQ hardware designs. Further, the results generated are reproducible and verifiable. The design allows a fair competition between different devices. The economic viability is hard do determine because of the current state of quantum computing. All in all, there is plenty of room for improvement, and the usefulness of Quantum Magic as an actual benchmark could be labeled as "situational".

6 Conclusions & Limitations

6.1 Conclusion

The main question that was analyzed in this research is how suitable quantum games are as a benchmark for quantum hardware. Based on the experimental game, an existing game from a researcher, and analysis of theoretical research we conclude that quantum games can indeed be used as a benchmark for quantum hardware.

There are certain advantages of using quantum games for the benchmarking process. The usage of a game can stimulate motivation and ease the task of benchmarking. Besides that, it is a way of enabling participation to the process and to quantum technology. Combined, these aspects can lead to large(r) amount of data collected on a quantum device which might lead to new insights. An important disadvantage of using quantum games for benchmarking is that in the current state of quantum computing, it is hard to use games to come up with interesting performance data, other than low-level metrics. Further there are the hardships of interoperability between NISQ devices, the limited capacity of current quantum devices to support simultaneous players, and the requirement for connecting an audience to (parties owning) quantum devices.

All in all, the usage of quantum games as benchmarks is situational and depends on the purpose of the benchmarking process. Some situations where quantum games might be a good way to perform benchmarking are:

- Statistics: when a lot of statistics are required, or a lot of testing with a device should be done.
- Awareness: when it is desired to involve people with the technology of quantum computing.
- Education: when benchmarking needs to be performed by someone with limited knowledge, since in the process of gaming the goal is set and the process can have educational benefits.

On the other hand, quantum games might ideally not be used for benchmarking when specific data needs to be collected, for example of the hardware performance on a specific algorithm. Also quantum games might not be very economical as hardware benchmark when there is not a clear plan or purpose for it.

To conclude, it is a realistic option to use quantum games as a benchmark for quantum hardware. However, the true usefulness differs per case and is highly dependent on the purpose of the benchmarking. It is not likely that quantum games will become some sort of standard benchmark for quantum hardware. However, in specific cases the usage of quantum games might provide enough advantages to actually see them used for benchmarking in practice.

6.2 Limitations

Like in all research, this research knows certain limitations that have to be kept in mind.

Research First of all, this research is mostly based on the theory of previous research. While a small experiment was carried out to provide some practical insights, its conclusions are still highly theoretical. For instance, aspects like the time and costs for development of a game, upkeep costs or implementation strategies have all not been accounted for in this research.
Secondly, the research is based on theory thought to be most relevant to create a broad image on the subject. Due to this choice, certain aspects may have been missed, that otherwise might have changed (aspects of) the outcome of the research. Also, since quantum computing is a quickly developing research field, some actual developments might quickly change the findings of this research.
Finally, one small experiment was the basis for the practical insights. This experiment also has some shortcomings which are discussed in the following paragraph. This might definitely be of heavy influence of the outcome. With a different result of the experiment the conclusion of the research might have been significantly different.

Game The game, developed as an experiment for this research, also has certain limitations itself. The game was developed in such a way, that it fit to certain theoretical properties. This was done as it was developed as a method of supporting this research. As a result however, the quality is limited and it would not be suited for a practical use in the current shape. Besides that, it was made on and for one device. Because of this, it lacks interoperability options and its functioning and structure might have been completely different when designed with a different device.

7 Further Research

Based on the findings and the shortcomings of this research, the following ideas for future research on this topic came up:

- A further analysis of related theory can be made, to discover which aspects might have been missed in this research.

- A more practical research in relation to real-world application might be nice. An example might be a project in collaboration with a company which is interested in benchmarking their own hardware.
- This research could be expanded upon with extension of the created game or the creation of another game. It would be interesting to see a more universal game and mechanisms that focus on different benchmarks, for example on hardware performance on specific algorithms.

8 Code Availability

The code and assets for the game Quantum Magic can be obtained on GitHub [Quaa].

References

[Bac03] D Bacon. Decoherence, control, and symmetry in quantum computers, 2003.

[BBC$^+$17] L. S. Bishop, S. Bravyi, A. Cross, J. M Gambetta, and J. Smolin. Quantum volume. *Quantum Volume. Technical Report*, 2017.

[BKY18] Robin J. Blume-Kohout and Kevin Young. Metrics and benchmarks for quantum processors: State of play. *Office of Scientific and Technical Information*, 2018.

[Blo] Bloch duel. `https://gitlab.com/michalowski.prz/bloch-duel/-/tree/main`.

[BLW19] D. Beyer, S. Löwe, and P. Wendler. Reliable benchmarking: requirements and solutions. *International Journal on Software Tools for Technology Transfer*, 21:1–29, 2019.

[Cai58] R. Caillois. *Man, Play and Games*. University of Illinois Press, 1958.

[Cam] Benchmarking - english meaning. `https://dictionary.cambridge.org/dictionary/english/benchmarking`. Accessed: 25-04-2024.

[CMH10] S. K. Curtis, C. A. Mattson, and S. P. Harston. On barriers to reverse engineering mechanical components. *American Society of Mechenical Engineers*, 5:575–590, 2010.

[Coo14] S. Cooper. *A framework for scientific discovery through video games*. Morgan & Claypool, 2014.

[CP16] C. Collberg and T. A. Proebsting. Repeatability in computer systems research. *Communications of the ACM*, 59(3):62–69, 2016.

[DAJ11] D. Djaouti, J. Alvarez, and J. P. Jessel. *Classifying Serious Games: The G/P/S Mode*. IGI Global, 2011.

[DB19] W. Dai and D. Berleant. Benchmarking deep learning hardware and frameworks: Qualitative metrics. *CoRR*, abs/1907.03626, 2019.

[DCD24] G. R. Di Carlo and L. DiCarlo. Quantum inspire starmon-5 fact sheet, 2024. Retrieved from: `https://qutech.nl/wp-content/uploads/2024/03/Starmon5_FactSheet_240316.pdf`.

[DDKN11] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From game design elements to gamefulness: Defining gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*, 11:9–15, 2011.

[DiV00] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48:771–783, 2000.

[Fra] An ieee framework for metrics and benchmarks of quantum computing. `https://quantum.ieee.org/images/files/pdf/ieee-framework-for-metrics-and-benchmarks-of-quantum-computing.pdf`. Accessed: 03-05-2024.

[FRH^{+}22] J. R. Finzgar, P. Ross, L. Holscher, J. Klepsch, and A. Luckow. Quark: A framework for quantum computing application benchmarking. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022.

[GK09] A. Gurumurthy and R. Kodali. Application of benchmarking for assessing the lean manufacturing implementation. *Benchmarking: An International Journal*, 16:274–208, 2009.

[Hak15] M. Haklay. *Citizen Science and Policy: A European Perspective*. Woodrow Wilson Center, 2015.

[Ham19] J. Hamari. Gamification. *The Blackwell Encyclopedia of Sociology.*, 2019.

[Hid15] S. Hidi. Revisiting the role of rewards in motivation and learning: implications of neuroscientific research. *Educational Psychology Review*, 28:61–93, 2015.

[Hup09] K. Huppler. The art of building a good benchmark. *IBM*, 2009.

[IEE] Ieee at a glance. `www.ieee.org/about/at-a-glance.html`. Accessed: 11-06-2024.

[Jin17] Dal Yong Jin. Critical interpretation of the pokémon go phenomenon: The intensification of new capitalism and free labor. *Mobile Media & Communication*, 5(1):55–58, 2017.

[KGA^{+}18] N. Khammassi, G. G. Guerreschi, I. Ashraf, J. W. Hogaboam, C. G. Almudever, and K. Bertels. cqasm v1.0: Towards a common quantum assembly language, 2018.

[KLK20] S. Kounev, K. D. Lange, and J. Kistowski. *Benchmarking Basics*. Springer International Publishing, 2020.

[Koo] Marloes Koolmees. Nationale tuinvogeltelling 2021 verbreekt alle records – en dit is de top 10. `https://www.rootsmagazine.nl/vogels/van-pimpelmees-tot-kauw-en-ekster-tot-roodborst-deze-10-vogelsoorten-werden-het` Accessed: 11-06-2024.

[LL02] E. A. Locke and G. P. Latham. Building a practically useful theory of goal setting and task motivation: A 35-year odyssey. *American Psychologist*, 57(9):705–717, 2002.

[Loc15] Michael Loceff. *A Course in Quantum Computing (for the Community College) Volume 1*. Foothill College, 2015.

[LSKA22] A. Li, S. Stein, S. Krishnamoorthy, and J. Ang. Qasmbench: A low-level qasm benchmark suite for nisq evaluation and simulation, 2022.

[MAAM22] K. Mesman, Z. Al-Ars, and M. Möller. Qpack: Quantum approximate optimization algorithms as universal benchmark for quantum computers, 2022.

[Mic14] A. C. Michalos. *Encyclopedia of quality of life and well-being research*. SpringerReference, 2014.

[MK22] M. Moradi and M.R. Keyvanpour. Captcha for crowdsourced image annotation: directions and efficiency analysis. *Aslib Journal of Information Management,*, 74:522–548, 2022.

[NC10] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[OW23] M. Ozols and M. Walter. The quantum quest, 2023. Retrieved from `https://qi-rub.github.io/quantum-quest/2023/en/index.html`.

[Pho] Photonic trail. `https://qplaylearn.com/treasure-hunt`.

[PPW$^+$24] L Piispanen, M. Pfaffhauser, J. Wootton, J. Togelius, and A. Kultima. Defining quantum games, 2024.

[Quaa] Quantum magic. Available on `https://github.com/MJH-Kooter/Quantum-Magic/tree/main`.

[Quab] Quantum escape. `https://quantumgames.aalto.fi/game-quantum-escape/`.

[QuE] QuEra. Quera glossary. Retrieved from QuEra: `https://www.quera.com/glossary/mid-circuit-measurement`.

[QuT18] QuTech. Quantum inspire home., 2018. Retrieved from Quantum Inspire: `https://www.quantum-inspire.com/`.

[RD00] R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25:54–67, 2000.

[RK21] S. Resch and U. R. Karpuzcu. Benchmarking quantum computers and the impact of quantum noise. *ACM Computing Surveys,*, 54, 2021.

[Rob] Robert c's speaking profile. `https://www.expertbase.org/478`. Accessed: 12-06-2024.

[RR09] B. Reeves and L. Read. *Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete*. Harvard Business Press, 2009.

[SPE] Standard performance evaluation corporation. `https://spec.org`. Accessed: 12-06-2024.

[Sta09] T. Stapenhurst. *The Benchmarking Book*. Butterworth-Heinemann, 2009.

[Ver20] R. Versluis. Here's a blueprint for a practical quantum computer. *Quantum computers scale up*, 2020.

[vK] Adinda van Kuijk. Nationale tuinvogeltelling 2022: de merel laat zich steeds meer zien. https://www.rootsmagazine.nl/nieuws/nationale-tuinvogeltelling-2022-dit-is-de-top-10. Accessed: 11-06-2024.

[vKAH+15] J. von Kistowski, J. Arnold, K. Huppler, K. D. Lange, J. Henning, and P. Cao. How to build a benchmark. *ICPE*, 2015.

[VLZC+21] K. Vohland, A. Land-Zandstra, L. Ceccaroni, R. Lemmens, J. Perelló, M. Ponti, R. Samson, and K. Wagenknecht. *The science of citizen science evolves.* Springer, Singapore, 2021.

[VN19] E. Van Nieuwenburg. Introducing a new game: Quantum tiqtaqtoe. Quantum Frontiers, 2019.

[Voga] De nationale tuinvogeltelling. https://www.vogelbescherming.nl/tuinvogeltelling. Accessed: 11-06-2024.

[Vogb] Nationale tuinvogeltelling: ruim 2 miljoen vogels geteld. https://www.bnnvara.nl/vroegevogels/artikelen/resultaten-nationale-tuinvogeltelling. Accessed: 11-06-2024.

[Waz17] K. Wazny. Crowdsourcing ten years in: A review. *Journal of gloabal health*, 7, 2017.

[WDE+23] K. Wintersperger, F. Dommert, T. Ehmer, A. Hoursanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin, and S. Luber. Neutral atom quantum computing hardware: performance and end-user perspective. *EPJ Quantum Technology*, 10:32, 2023.

[Woo18] J. Wootton. Using a simple puzzle game to benchmark quantum computers. *Medium*, 2018.