

Final Report: Quantum Storm -

Educational Quantum Computing Puzzle Game

Student: Feng Wu (a1879322)

Project Group: KM5b

1. Project Aims

1.1 Background

In recent years, quantum computing has gradually moved from theoretical research to practical applications, and is considered to be the next generation of computing paradigm after classical computers. With its exponential processing power in specific problems, it is expected to bring breakthroughs in cryptography, drug discovery and development, optimisation problems, financial modelling, and so on (Tunçel, 2024). Therefore, learning about quantum computing, or understanding the related concepts, will help people keep up with and understand this future technology.

1.2 Key Problems

The basic unit of quantum computing, unlike traditional bits that can only be 0 or 1, quantum bits, or called qubit in short, can be in a superposition of 0 and 1, which is commonly represented as a matrix in mathematics. Furthermore, the entanglement states between multiple qubits, as well as quantum gates, can be represented by the corresponding more complex matrix arithmetic. Since matrices and their computations are complex advanced mathematics, and the traditional teaching of quantum computation is based on matrices and formulas, these are great challenges for students without a good foundation in mathematics, even for some undergraduate and graduate students.

1.3 Project aims

Our main goal is to use Unity to design a game to transform the difficult abstract concepts of quantum computation into an easy-to-understand, visual and interactive way, so that players can interact, learn, understand and apply these concepts in a low-barrier method, and thus master the knowledge of quantum computation in the process of playing the game.

- The specific teaching objectives are:
 - To lower the threshold of the game to the level of elementary or high school students, i.e., without any knowledge of higher mathematics related to matrices, players can play the game and understand the concepts of quantum computing at the same time.
 - To enable players to grasp the basic quantum computing concepts, including quantum bits, superposition states, collapsed states, quantum measurement, quantum entanglement, quantum gates, and so on.

- After players understand the concepts, they have the opportunity to use them to solve level puzzles, gain a sense of accomplishment, stimulate learning interest, and create positive feedback.

2. Approach

2.1 From abstract concept to a game

To transform quantum computing into a game, the first step is to gamify or visually represent its abstract concepts—that is, to express these ideas in intuitive, non-mathematical ways.

After studying the fundamentals of quantum computing in depth, we found that the Bloch Sphere offers an intuitive visualisation of a qubit's state, including classical states $|0\rangle$ and $|1\rangle$, as well as superposition states (see Figure 1). The Bloch Sphere is a widely used representation in quantum computing, introduced by Felix Bloch and commonly used in quantum mechanics education (Nielsen & Chuang, 2010). When the arrow on the Bloch Sphere points upward (along the Z axis), it represents the $|0\rangle$ state; when it points downward (along the $-Z$ axis), it represents the $|1\rangle$ state. Positions in between indicate a superposition of $|0\rangle$ and $|1\rangle$, with the arrow's proximity to the top or bottom reflecting the corresponding measurement probability. After measurement, the arrow collapses in either the upward or downward direction, signifying that the qubit has collapsed to a definite state.



Figure 1. A Bloch Sphere illustrating the state of a qubit.

Next is the representation of quantum gates. We observed that single-qubit gates' function can be visualised on the Bloch Sphere as rotations of the arrow around the X, Y, or Z axes (see Appendix 1 - Bloch Sphere Rotations Corresponding to Quantum Gates). For example, the X gate corresponds to a 180-degree rotation around the X-axis, the Y gate around the Y-axis, and the H (Hadamard) gate around an axis between X and Z. Once these transformation rules are summarised, the Bloch Sphere can be effectively used to visualise how a qubit's state changes when acted upon by different gates.

For multi-qubit gates like the CX (controlled-X) gate, an additional concept, quantum entanglement, must be introduced. Entangled qubits influence one another's measurement outcomes—for instance, some entangled qubits will always yield correlated or opposite results upon measurement. However, since the Bloch Sphere is only capable of representing the state of an individual qubit, it cannot describe the joint probabilities of entangled qubit pairs. In fact, fully representing these relationships requires matrix algebra, which contradicts our project's aim of avoiding complex mathematics.

Through further exploration, I found a feasible compromise: treat each qubit as an independent entity while visually highlighting entangled pairs using identical-coloured outlines (see Figure 2). The Bloch Sphere continues to show individual qubit states, but the underlying code tracks entangled states. When one qubit is measured, its entangled counterpart is automatically updated to reflect the entangled outcome. As players can clearly see which qubits are

entangled via the outline, the resulting simultaneous change in states appears intuitive and natural within gameplay.

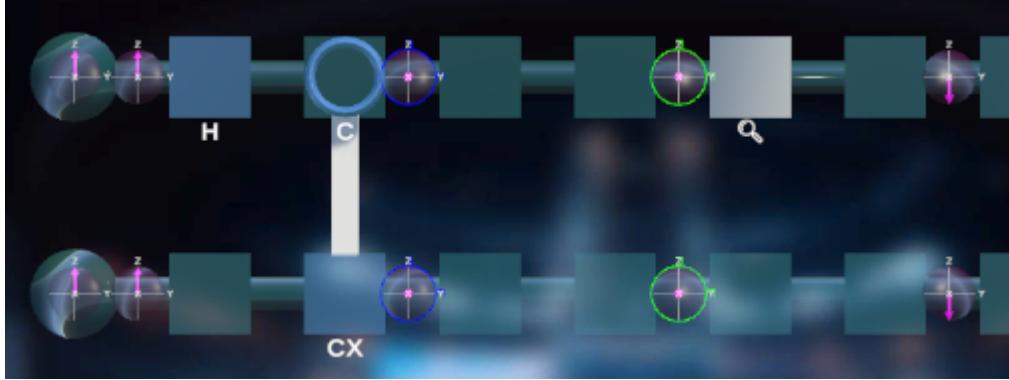


Figure 2. Visualisation of a CX (controlled-X) gate acting on two qubits. Using identical-coloured outlines to highlight entangled qubit pairs.

2.2 Game story and level design

The storyline and puzzle levels are the two most important elements of our game. A well-crafted story allows players to immerse themselves in the experience. Puzzle-based level design, on the other hand, provides a gradual and engaging way for players to encounter, learn, and apply new concepts. Combined, these two components make the game enjoyable, appealing, and intellectually stimulating.

The narrative is set in a futuristic space station, beginning with a space accident that creates a sense of urgency. Players must complete quantum computing challenges to save the station. The dramatic storyline, integrated with puzzle-solving gameplay, allows players to engage without perceiving the experience as purely educational—making the game genuinely enjoyable. Simultaneously, the narrative encourages reflection and adds imaginative depth that goes beyond the mere acquisition of knowledge.

In terms of gameplay, we designed a sequence of levels. Starting with the use of a measurement gate to observe qubit states, players are gradually introduced to individual qubits, single quantum gates, and eventually more complex circuits composed of multiple gates. By establishing distinct goals and puzzle mechanics for each level, the game maintains variety and interest. In later stages, the free-form quantum circuit building mode allows for more than one solutions for players to explore, increasing the challenge while enhancing the sense of discovery and achievement.

2.3 Technical Implementation

- Game Engine: Unity, version 6000.0.45f1
- Programming Language: C#
- version control & code management tool: Github

3. Progress

3.1 Team collaboration

We held meetings twice a week, on Mondays and Wednesdays. Monday meetings were for internal team discussions and coordination, while Wednesday meetings were dedicated to reporting project progress and receiving feedback from the project client (Kamal Mammadov).

3.2 Sprint Progress Overview

- **Weeks 1–2: Studied quantum computing fundamentals.**

During this period, we deeply studied key concepts in quantum computing and summarised them—such as the table shown in Appendix 1 - Bloch Sphere Rotations Corresponding to Quantum Gates. We then began brainstorming ideas for game levels and the storyline. For level design, we referred to IBM Composer, a quantum computing simulation platform (<https://quantum.ibm.com/composer/files/new>), that allows users to build quantum circuits by dragging and dropping gates. This visual and interactive approach was well suited to game development. As for the storyline, we explored two directions: one involving a quantum computing crisis on a space station, and the other telling the life story of a quantum scientist. The former was simpler and easier to implement, while the latter offered more narrative depth but posed greater challenges. Since we couldn't decide quickly, we initially developed both options in parallel.

- **Weeks 3–4: Completed Agile Sprint 1. Completed conceptual design for levels and selected the narrative direction.**

For level design, we created initial sketches (see Appendix 2 - Concept Sketch of Game Level Design) and began early development in Unity. We modelled the qubit, gate, and qubit machine, implemented the drag-and-drop function, and developed a basic main menu. For the story, we decided to go with the space station setting and chose a card-based dialogue system for narrative delivery. Compared to an interactive map-exploration format with character movement, this option was simpler and more aligned with our goals. Once the narrative direction was confirmed, we started to use AI-generated imagery to support the development of the game's visual style and character concepts.

- **Weeks 5–6: Completed Agile Sprint 2. Developed initial levels and narrative.**

In Unity, we completed the 3D model with two visualisation modes: simple mode and Bloch Sphere mode. We also implemented several single-qubit gates, including X, Y, Z, X/2, Y/2, Z/2, H, and M gates. Except for the M gate, these gates work by rotating the qubit arrow by 180 or 90 degrees around a specific axis, which could be implemented with a unified logic, allowing for quick development. We also began designing specific game levels and produced sketches for Levels 1–3. On the narrative side, we completed the script for the game's opening and Story 1, and began building story scene templates in Unity while sourcing sound effects.

- **Weeks 7–8: Completed Agile Sprint 3. Developed Levels 1–3 and Stories 1–2, resulting in a basic game prototype.**

For the level design, we implemented the CX gate—a two-qubit gate involving up to 16 possible states and complex quantum entanglement. Entanglement also affects the

implementation of other gates, such as the M gate, making this a major challenge. I spent significant time conducting in-depth research and simulations using IBM Composer, eventually extracting consistent rules and implementation logic to successfully build the CX gate. Additionally, we completed the Main Menu, Levels 1–3, and Stories 1–2, which together formed a functional game prototype ready for demonstration.

- **Weeks 9–10: Completed Agile Sprint 4. Developed Levels 3–6 and Stories 3–4; began testing and bug fixing.**

We designed and implemented Levels 4–6 and Stories 3–4. During internal meetings, we tested the game from the beginning, identified gamming issues, bugs and transition problems, recorded them in a to-do list, and updated or fixed them one by one during the sprint. We also added several features: gate instruction panel feature by right-click-and-hold, hint functionality for each level, and a skip button in the story scenes to improve gameplay flow.

- **Weeks 11–12: Completed Agile Sprint 5. Finalised full game development.**

We completed Game Level 7, all remaining story segments (Stories 5–8), and the ending scene. At this point, the game was fully developed. We continued testing, refining gameplay details, and fixing bugs based on the test results. Enhancements included assigning different background music to levels, adding a developer menu in the main menu for quick level access, embedding videos into the gate instruction panel, and fixing visual glitches when players continue playing the game.

4. Result

4.1 Final game deliverables

We developed an educational puzzle game on quantum computing that features a compelling narrative and challenging level design. The game includes seven levels, eight story segments, a main menu, a loading scene, and an ending scene. Below is a detailed description of each scene (see Appendix 3 - Game Scene Screenshots (Ordered by Gameplay Flow)):

1. **Starting Scene:** Displays the game title, *Quantum Storm – Crisis on the Space Station*, with a backdrop of astronauts, a space station, and the vastness of space. Pressing any key leads to the main menu.
2. **Main Menu:** Set against a space-themed background of planets and spaceships, accompanied by mysterious, rhythmic music to evoke a sense of the unknown. The menu includes:
 - Start Game: Begins the game, leading to Story 0.
 - Continue: Allows players to resume from the last saved story segment.
 - Lab: A sandbox lab containing all qubits and gates for free exploration and experimentation.
 - Setting: Adjusts overall game volume.
 - Quit Game: Exits the game.

- Dev: Developer mode for jumping directly to any level.
3. **Loading Scene:** Features a rotating quantum icon and uses asynchronous loading to improve efficiency and reduce wait times, ensuring a smoother gameplay experience.
4. **Story 0:** An opening cutscene using text and video to introduce the game's backstory, followed by a transition to Story 1.
5. **Story 1:** Introduces the protagonists, Erik and Daph, who describe the space station's emergency through dialogue and guide the player into Level 1. A skip button at the top-right allows players to bypass the story.
6. **Level 1:** Introduces the concept of qubit superposition. Players must use the measurement gate to collapse qubits and complete the level. Key features include:
- Speed Slider: Adjusts the movement speed of qubits within the machine, allowing for either careful observation or faster completion.
 - Simple Mode/Sphere Mode: Switches between numerical and Bloch Sphere visualisation of qubit states.
 - Hint Panel: Provides game level goal, requirements and control instructions.
 - Settings Button: Adjusts volume.
 - Exit Button: Returns to the main menu.
 - Launch Button: Starts the qubit machine, which moves the qubit through gates to the output area.
 - Qubit Enlargement: Activated on click, this feature enlarges the qubit for closer inspection and enables the player to rotate the viewing angle for a more detailed perspective.
 - Gate Instruction Panel: Triggered by right-click and hold, displaying gate instructions and video explanations.
7. **Story 2:** Continues Erik and Daph's journey through various areas of the space station, accompanied by changing scenes and background music. They arrive at the lab, which requires completing Level 2 to enter.
8. **Level 2:** Introduces the X gate, similar to the NOT gate in classical computing. It rotates the qubit 180 degrees around the X-axis. Players must achieve three qubits in the $|1\rangle$ state to clear the level.
9. **Story 3:** The protagonists acquire the Hadamard module. Due to an unexpected event, they are trapped in the lab and must use the H gate to solve Levels 3 and 4.
10. **Level 3:** Hadamard (H) gate is introduced in this level. It performs a 180-degree rotation of the qubit around the axis between the X and Z axes, transforming the basis state $|0\rangle$ into a superposition state.
11. **Level 4:** Requires strategic use of both X and H gates. A key concept is that applying two H gates in a row returns a qubit to its original state.

12. **Story 4:** With the lab crisis resolved, the protagonists proceed to the core chamber to repair the quantum core. The difficulty increases slightly, reflecting the core's importance. Players must solve a puzzle to gain entry.
13. **Level 5:** CX gate is introduced in this level. This two-qubit gate creates entanglement if the control qubit is in superposition. This level is simple for players to learn it.
14. **Story 5:** The protagonists begin repairing the quantum core, the "brain" of the station, as gameplay difficulty continues to rise in accordance with the storyline.
15. **Level 6:** Players apply previously learned X and CX gates. The challenge is to use four X and four CX gates across four machines to produce the output 1011. This is a new gamming machenics, making the game level more interesting, and deepen the understanding of the knowledges players just learned.
16. **Story 6:** With the core restored, the crisis seems resolved. Erik ventures outside the station alone for a final inspection, which requires completing Level 7.
17. **Level 7:** The final challenge. Players, now familiar with all concepts, encounter gates that rotate qubits by various angles around the X, Y, and Z axes. They must use each gate at least once to produce the output 1111. Instructions are accessed through right-clicks for them to learn these gate by themselves. These new gates are easy to understand, but this puzzle is not easy to solve.
18. **Story 7:** Outside the station, Erik discovers that the core was never actually damaged. The crisis was orchestrated by Daph, who sought to demonstrate that the super AI within the core remains under human control. Her intent was not malicious but rather to ease humanity's distrust of AI.
19. **Story 8:** Instead of accusing Daph, Erik helps her pass inspections, sparing her from destruction as a perceived threat. The story ends on a philosophical note about technology and humanity's future.
20. **Ending Scene:** listing all team members and their respective contributions, along with a message 'thanks for playing'.

5. Contribution

5.1 Educational impact

Our game successfully achieved its project goals by delivering a complete educational game on quantum computing. It significantly lowers the barrier to understanding complex quantum concepts and establishes a smooth learning curve. By simply following the story and progressing through the levels, players can grasp the fundamental ideas without needing a background in advanced mathematics. As a result, individuals without strong mathematical foundations can still learn and understand what would typically be considered difficult content. Furthermore, our game simulates commonly used quantum gates, making it a useful visual tool even for students newly introduced to quantum computing, enabling them to experiment with basic quantum circuits.

5.2 Personal Contributions

As one of the core members of the development team, I played a central role in both the conceptual and technical aspects of the project. My contributions spanned from the initial design and idea formulation to hands-on implementation in Unity and project coordination. The following outlines my specific responsibilities and achievements throughout the development process.

Project Concept:

- Independently conceptualised the entire transition from quantum computing principles to game level design (see Appendix 2 - Concept Sketch of Game Level Design), including the representation of qubits, the operational logic and visualisation of quantum gates, and the functional design of the qubit machine.
- Designed most of the game levels, including Levels 1–5 and Level 7, while Level 6 was co-developed with Zhenlu Shan (see Appendix 4 - Game Level Design Table).

Unity Development:

- Independently implemented all core functionalities for Levels 1–7 in Unity, including:
 - Modelling of qubits, quantum gates, and the qubit machine, with dual visual modes—Simple Mode and Sphere Mode—using two separate model sets.
 - Functional development of all qubits, gates, and the qubit machine, including drag-and-drop mechanics and algorithmic implementation of all nine quantum gates.
 - Programming of the qubit machine system: cloning and moving qubits from input after launch, optimising asynchronous qubit motion, collecting results at output, and resetting machine states upon shutdown.
 - Speed control of qubit motion via a slider bar.
 - Success/failure judgement logic for each level, hint functions, post-puzzle Continue button, and loading next scene function.
 - Independently created the background image, found and applied all background music, and sound effects for the levels.
- Co-developed the following features with Kunrui Wang:
 - Gate instruction panel by right-click-and-hold.
 - UI layout for the levels.
- While the storyline was primarily written by Mu Sha and supported by Lirui Zhang, I implemented the skip button in the story scenes.

Project Management:

- Served as the Scrum Master, ensuring weekly project progress.
- Responsible for preparing weekly meeting agendas and maintaining the to-do list.
- Delivered weekly progress presentation during online meetings with Kamal Mammadov every Wednesday.

In addition, for the whole project, the major contributors to the project Unity development were Kunrui Wang, Mu Sha, and myself. I focused on level implementation, Kunrui on menus, loading screens, and testing, and Mu Sha on the storyline. Additionally, Jiaqi managed most of the project's documentation, including meeting minutes and reports. Lirui Zhang supported Mu Sha in narrative development, and Zhenlu Shan contributed to the design of the levels.

6. Conclusion

This project successfully transformed abstract quantum computing concepts into an interactive and accessible educational game through the Unity platform. By combining Bloch Sphere visualisation, narrative storytelling, and progressively challenging puzzles, we created an experience that makes learning quantum computing understandable for a broad audience. Our teamwork, driven by iterative development and effective project management, enabled us to deliver a complete, playable prototype that achieves both educational and entertainment objectives. This work demonstrates the potential of gamified learning to bridge the gap between complex scientific theories and public understanding.

Reference

Tunçel, K. (2024). Exploration and Development of Quantum Computing Algorithms for Optimization, Cryptography, and Machine Learning Applications. *Human Computer Interaction*. <https://doi.org/10.62802/1dwq2w96>.

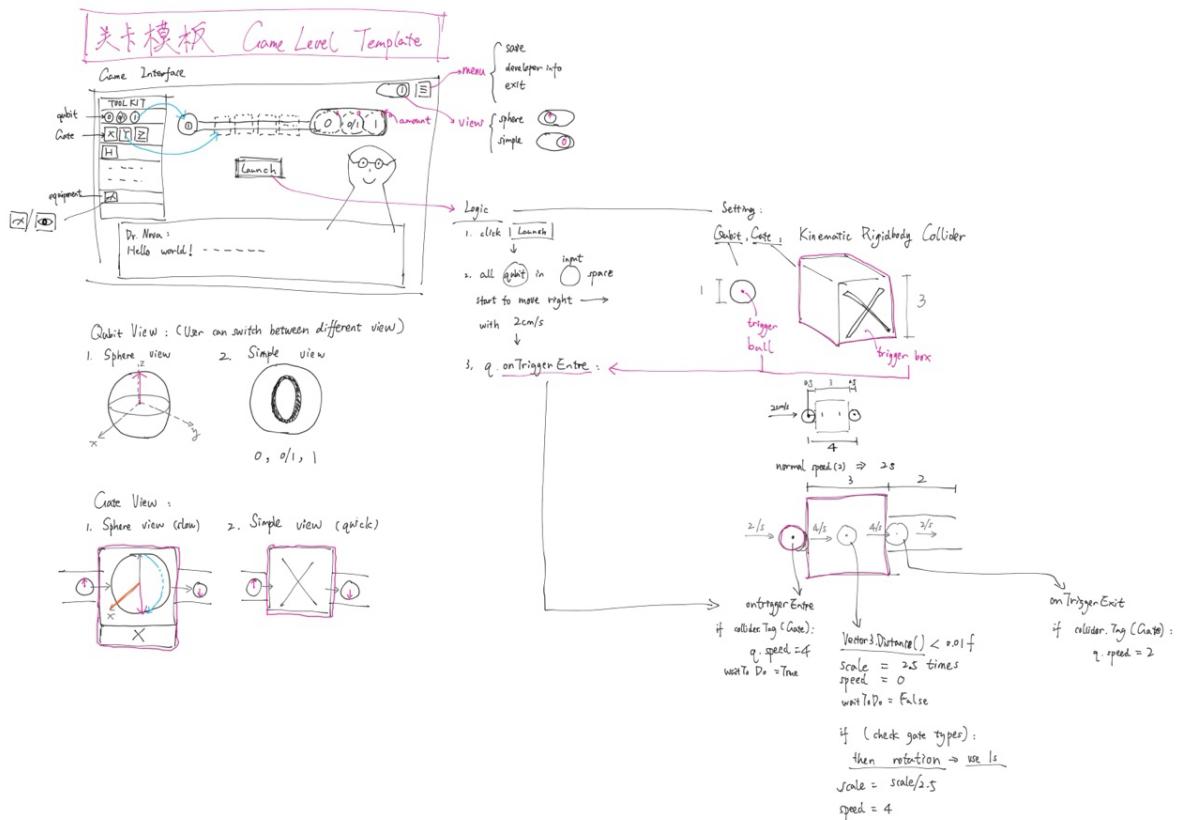
Nielsen, M. A., & Chuang, I. L. (2010). *Quantum computation and quantum information*. Cambridge university press.

Appendix

Appendix 1 - Bloch Sphere Rotations Corresponding to Quantum Gates

Gate Name	Bloch Sphere Rotation or Action
X (NOT gate)	About X-axis by 180 degrees
Y	About Y-axis by 180 degrees
Z	About Z-axis by 180 degrees
X/2	About X-axis by 90 degrees
Y/2	About Y-axis by 90 degrees
Z/2	About Z-axis by 90 degrees
H (Hadamard gate)	About axis between X and Z by 180 degrees

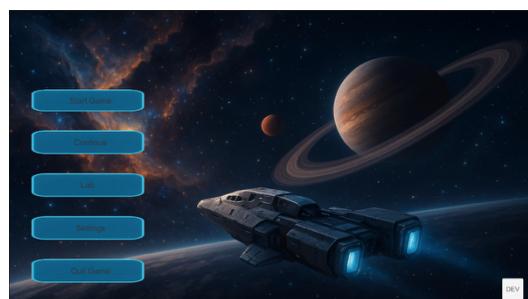
Appendix 2 - Concept Sketch of Game Level Design



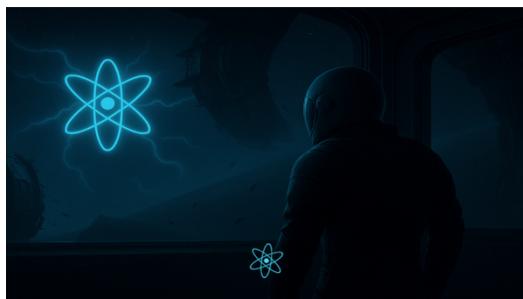
Appendix 3 - Game Scene Screenshots (Ordered by Gameplay Flow)



1 - starting



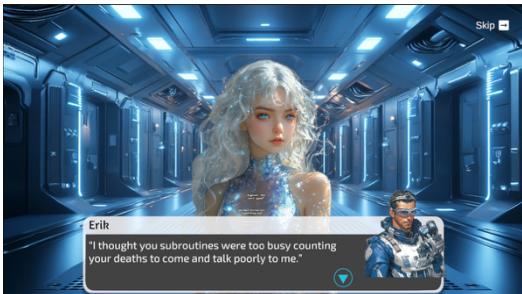
2 - main menu



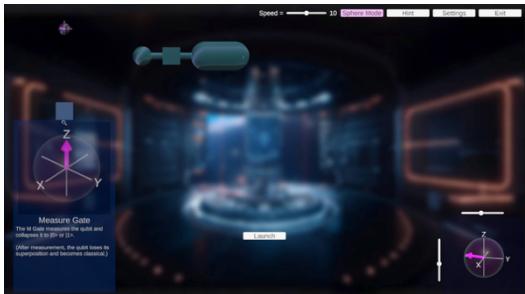
3 - loading



4 - story 0



5 - story 1



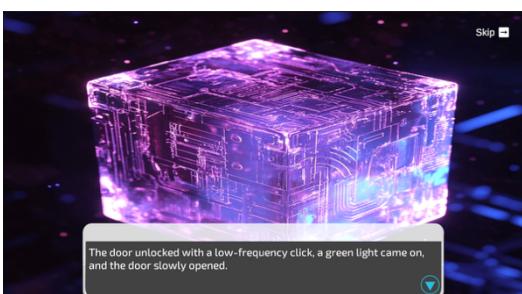
6 - level 1



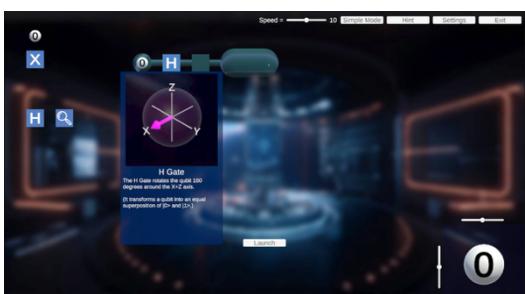
7 - story 2



8 - level 2



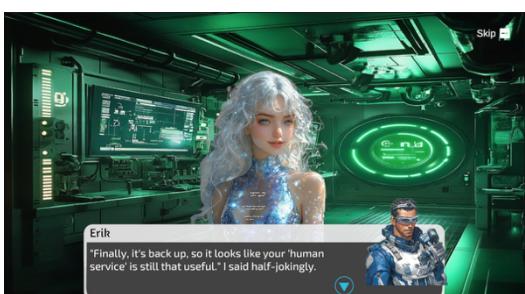
9 - story 3



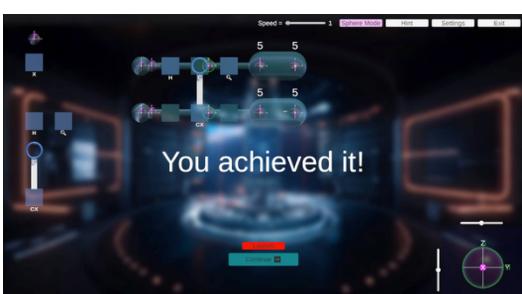
10 - level 3



11 - level 4



12 - story 4



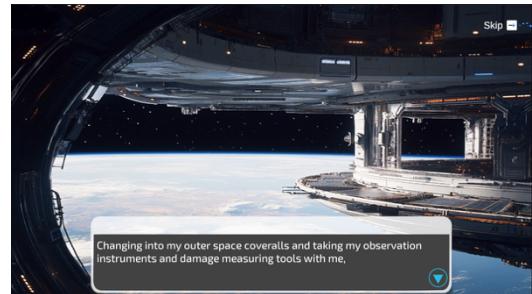
13 - level 5



14 - story 5



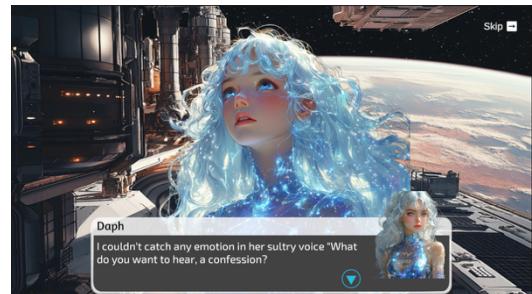
15 - level 6



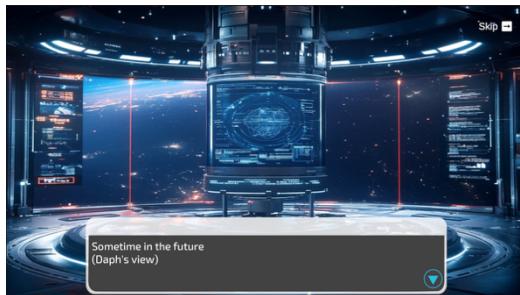
16 - story 6



17 - level 7



18 - story 7



19 - story 8



20 - ending

Appendix 4 - Game Level Design Table

Aa Name	Title	Toolkit	Rewards	# Machine	# Gate	Goal	Solution	Mission?	Difficulty
L1	Qubit Measurement	Qubit=2 M/		1	1	get 3x q0 + 3x q1, no q-2	q-2 - M → q0+q1	machine output accepts q0 or q1 only, please fix	
L2	Qubit Rotation X	Qubit=0 X		1	1	get 3x q1	q0 - X → q1	machine output accepts q1 only, pls fix	
L3	Qubit Superposition	Qubit=0 H		1	2	get 3x q0, 3q1, no q-2	q0/q1 - H - M → q0&q1	machine output need 3 q0 and 3 q1	
L4	Gate Problem	Qubit=0		1	3	H locked in middle gate, get 3x 1, no others	q0 - H - [H] - X → q1	broken machine, output accepts q1 only	
L5	Qubit Entanglement	Qubit=0 CX		2	3	get 3x 00, 3x 11, no -2	q0 - H - CX_c - M → q0 q0 - - CX_t	using quantum entanglement, update machine to keep 2 output always the same with results q0 & q1	
L6	X and CX	Qubit=0		4	4	use 4 X gates & 4 CX gates, to get result of 1011.		use 4 X and 4 CX gate	
L7	All gates	Qubit=0		4	8	use all gates for at least 1 time, to get result of 1111.			