
QUANTUM (Q)-KIT

Quantum Computing:
Laboratory for Secondary Level Students
Student Manual



WOMEN SUPPORTING
WOMEN IN THE SCIENCES

Meet a Scientist



Celine
Awino
Omondi

*Lecturer, Masinde Muliro
University of Science and
Technology and WS2 Partner
& Lab Kit Design Co-Lead,
Quantum Computing*

About me: I am a Kenyan scientist and educator with a passion for renewable energy, quantum science, and mentorship. I was born and raised in Mauna village in rural western Kenya. I spent my early childhood with my grandmother, attending primary school barefoot in grass-thatched classrooms that we helped maintain ourselves. These humble beginnings instilled in me a strong sense of resilience, discipline, and determination. I hold a Master of Science in Physics from Pune University (India) and a PhD in Renewable Energy (Solar Energy) from the Technical University of Berlin. My science interests sit at the nexus of renewable energy, materials physics, and quantum science. My career reflects a deep passion for understanding matter at both the macroscopic and quantum scales—and leveraging that knowledge to design sustainable technologies like photovoltaics for real-world impact.

My advice for students interested in science: Stay curious and persistent. Science is not just about getting the right answers – it's also about asking the right questions. Believe that your voice and ideas matter. Science thrives on diversity, creativity, and courage.

Mission Statement

The mission of this laboratory is to teach secondary level students (ages ~12-18) about quantum computing concepts through experiments related to quantum mechanics.

Contents

1. Introduction to WS2 Laboratory Kits	5
1.1. Information about WS2	5
1.2. Information about this Kit.....	5
1.3. Key Vocabulary	6
1.4. Key Questions	6
1.5. Purpose.....	7
2. Background on Main Topics	7
2.1. Quantum bits (Qubits) and Quantum Computers	7
2.2. Supplies List.....	9
2.3. Safety Information.....	10
3. Experiments.....	10
3.1. Part I. Quantum Coin Flip.....	10
3.1.1. Pre-Experiment Questions	10
3.1.2. Materials.....	10
3.1.3. Procedure (work in groups of 2-4)	11
3.1.4. Results.....	11
3.1.5. Post-Experiment Questions	12
3.2. Part II. Quantum Cryptography	13
3.2.1. Additional Background.....	13
3.2.2. Pre-Activity Questions.....	15
3.2.3. Materials.....	15
3.2.4. Procedure (work in groups of 2-4)	16

3.2.5.	Results.....	16
3.2.6.	Post-Activity Questions.....	17
3.3.	Part III. Quantum Computer Gates.....	18
3.3.1.	Additional Background.....	18
3.3.2.	Materials (for each group)	20
3.3.3.	Procedure & Analysis (work in groups of 2-4)	20
3.3.4.	Results.....	23
3.3.5.	Post-Activity Questions.....	23
4.	Design Challenge	25
4.1	Design Questions	25
4.2	Design Sketch	26
5.	Supplemental Activity: Quantum Computing with Python and Qiskit.....	27
5.1	Additional Background	27
5.2	Procedure.....	27
6.	Sources	29

1. Introduction to WS2 Laboratory Kits

1.1. Information about WS2

Women Supporting Women in the Sciences (WS2), an international organization unifying and supporting graduate and professional-level women and allies in science, technology, engineering, and mathematics (STEM), was awarded an American Physical Society (APS) Innovation Fund in 2020 to form international teams to design and distribute low-cost physics and materials science lab kits to primary and secondary school students, predominantly in eastern Africa. The lab kits utilized local resources and included topics that are especially relevant to young girls in order to spur their interest in STEM subjects. From 2020-2023, over 5100 students from eastern Africa at over 40 school sites engaged with our lab kits, with 62% being girls.

WS2 was awarded their second APS Innovation Fund in 2025 to support another Lab Kit Initiative, though this time with a focus on quantum topics. For more information about WS2, please visit our website at ws2global.org.

WS2 is sponsored by the APS Innovation Fund, APS Forum on Education, Northwestern University Materials Research Science and Engineering Center, and Northwestern University Multicultural Student Affairs. WS2 is extremely grateful to the lab kit design volunteers for their hard work (John Bakayana, Celline Omondi, Alice Flarend, Elvira Khwatenge, Babra Mwimali, and Sserugo Enock) and external consultants (SciBridge and Projekt Inspire) for their advising. WS2 also thanks and acknowledges PhysicsQuest (<https://www.aps.org/initiatives/physics-education/physicsquest>) and Quantum Explorations Student Toolbox (QuEST) for example experiments that were used as foundation for the lab kit content.

1.2. Information about this Kit

Welcome to the exciting world of quantum science—a realm where particles can be in multiple places at once, light behaves like both a wave and a particle, and distant objects can influence each other instantly. Sounds like science fiction, but it is not!

This manual is designed to guide you through simple, hands-on experiments that bring some of the most fascinating concepts of quantum physics to life. Using everyday materials, you'll explore big ideas like superposition, interference, and entanglement—concepts that challenge our classical understanding of how the universe works.

Don't worry if these terms are new or sound complicated. Each experiment includes:

- Clear purpose statement to indicate what you are learning
- Pre-lab questions to get you thinking
- Step-by-step procedures you can follow with ease
- Simple tools you likely already have at home or in class
- Observation tables for recording your results
- Thoughtful post-experiment questions to help connect what you've seen to the mysteries of quantum science

These experiments are meant to spark curiosity, creativity, and deeper thinking. You don't need to be a physicist (yet!) to enjoy them—just bring an open mind, a sense of wonder, and a willingness to explore ideas that stretch the limits of what we think is possible. As you work through these activities, remember: even the most famous scientists started by asking simple questions and trying small experiments. Who knows what discoveries your curiosity might lead to?

1.3. Key Vocabulary

- Bit (classical): the 0s and 1s that traditional classical computers use
- Qubit (quantum bit): fundamental unit of quantum information in quantum computing that can exist as 0 or 1 simultaneously
- Measurement: the process of collapsing the superposition state of a qubit into a (0/1) definite state
- Superposition: a system (like a qubit) existing in multiple states at once until measured
- Entanglement: the linking of states in objects no matter how far apart they are
- Quantum gate: a device that changes the quantum state of a qubit
- Quantum cryptography: a method of secure communication that encodes messages in qubits

1.4. Key Questions

- What is a qubit, and how does it differ from a classical bit?

- What is superposition?

- How does a quantum computer differ from a classical computer that uses binary 0s and 1s?

1.5. Purpose

The purpose of this lab kit manual is to learn about quantum computing through experiments and activities. Students will learn about the difference between classical bits and qubits and the concepts of superposition and entanglement. Students will also engage in activities that teach them about quantum cryptography and quantum computing through games and analogies.

2. Background on Main Topics

2.1. Quantum bits (Qubits) and Quantum Computers

Quantum computing, which is computing that utilizes quantum mechanical properties, is sought after because it can perform computations that are impossible for classical computers. How does a traditional computer work? Traditional computers process information using fundamental units of data called classical bits (either 0s or 1s). These bits are manipulated through logic operations which allow the computer to complete simple and complex tasks. How is a quantum computer different? Unlike classical bits that represent 0 or 1, quantum bits, or qubits, in quantum computers can exist in

superposition, which is a state in which the bit exists both as 0 and 1 simultaneously (see Figure 1). To collapse a superposition state, the qubit is measured, which means the qubit is no longer 0 and 1 simultaneously, but instead is either a definite 0 or 1.

What is another way to think about superposition? Imagine a light switch. It can be either ON (1) or OFF (0). This is like a traditional computer bit. Now, imagine a special light switch that can be ON, OFF, or both at the same time! That would be like a qubit. This "both at once" superposition ability is what gives quantum computers their power. While traditional computers can only process traditional bits (0 or 1), quantum computers can process 0, 1, or both, which means they can explore multiple possibilities at once.

Like traditional computers, quantum computers follow instruction sets called algorithms that employ logic gates to manipulate bits in a specific order. Instead of using classical gates (like "AND", "OR"), quantum computers use quantum logic gates, which are devices that manipulate qubits using quantum mechanics. For example, a Hadamard gate puts a qubit into a superposition state (both 0 and 1).

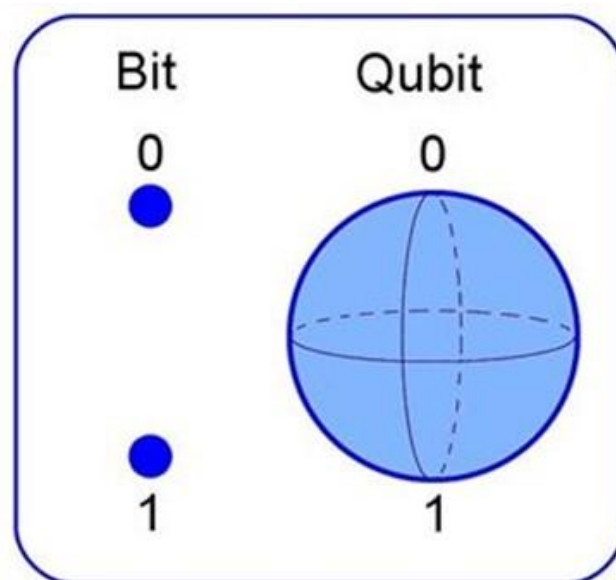


Figure 1. Illustration of classical bit (left) and qubit (right). The classical bit can either be a 0 or 1, and the qubit can exist in superposition as both a 0 and 1 simultaneously. [This Photo](#) by Unknown Author is licensed under [CC BY](#)

What are other quantum mechanics phenomena that quantum computers utilize? Entanglement, which is the linking of states (like qubits) no matter how far apart they are, is used in quantum computers to create powerful quantum algorithms and communication methods. If you think back to the light switch analogy from

superposition, now imagine two entangled light switches. If you flip one, the other instantly flips too, even if they are far apart! That is one spooky connection!

How could quantum computers be useful in our lives? Because quantum computers can explore multiple possibilities at once with qubits, they can work faster and more efficiently than traditional computers. This could be extremely useful in a variety of fields including drug discovery, materials science, and artificial intelligence. Quantum computers could also be used in cryptography, which is the way information is secured from unauthorized access. Classical cryptography uses math to scramble messages. Quantum cryptography uses qubits and quantum mechanics to create unbreakable codes. If someone tries to eavesdrop on a quantum message, the spy will disturb the qubits, and the sender will know someone is listening!

2.2. Supplies List

- Coins (metal with two distinct sides)
- Opaque cups
- Paper
- Wrapped candies (or small pieces of paper)
- Buckets or jars
- Markers
- Colored tape (red and green) (2 other colors may also be used)
- Save Schrödinger's Cat game pieces for each group:
 - 8 blue cat/red cat tokens to act as qubits
 - 8 yellow cat/green cat tokens to act as qubits
 - 2 X gates
 - 2 Y gates
 - 2 X gates
 - 2 S gates
 - 2 H gates
 - 1 CNOT gate
 - 1 qubit gate phase change table
 - 1 qubit Interference table
- Scissors
- Clear tape

3.1.3. Procedure (work in groups of 2-4)

1. Flip a coin onto a table or other hard surface and place the cup over it without revealing if it is heads or tails.
2. Predict whether the coin is heads or tails.
3. Measure the coin by lifting the cup and record the results.
4. Repeat 20 times.

3.1.4. Results

<u>Coin flip</u>	<u>Heads or Tails?</u>
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

3.1.5. Post-Experiment Questions

1. What was the percentage of times you measured heads? Tails? How did this vary in your classroom?
2. How do these percentages relate to the expected probability of heads versus tails?
3. What could you do to get your measured percentages to approach the expected probability of heads versus tails?
4. If we consider the coin to represent a qubit, what state is the qubit in when it is under the cup prior to measurement? What about after measurement?

5. How do we collapse the superposition state of the coin?

3.2. Part II. Quantum Cryptography

3.2.1. Additional Background

Quantum key distribution (QKD) is a way to exchange cryptographic codes (keys) securely using quantum physics. Here, light is used to pass information between two parties. Imagine that Alice wants to send Bob a message, but they do not want anyone else to read it. Alice sends Bob a secret code (a key) using tiny particles of light called photons. Here is the quantum twist: if someone tries to listen in (like Eve), the person will disturb the photons, and Alice and Bob will know. It is like having a secret alarm system for your message. Think of information you want to keep safe – phone messages, bank account passwords, and PINs. These are all areas in which quantum cryptography could be helpful!

There are many ways quantum cryptography can protect our information, but what about situations in which quantum computers are being used to try to steal our information? In this case, we use quantum physics to combat these attacks. Specifically, post-quantum cryptography (PQC) is a sub-field of quantum cryptography that develops algorithms to protect against quantum computers (see Figure 2), As quantum

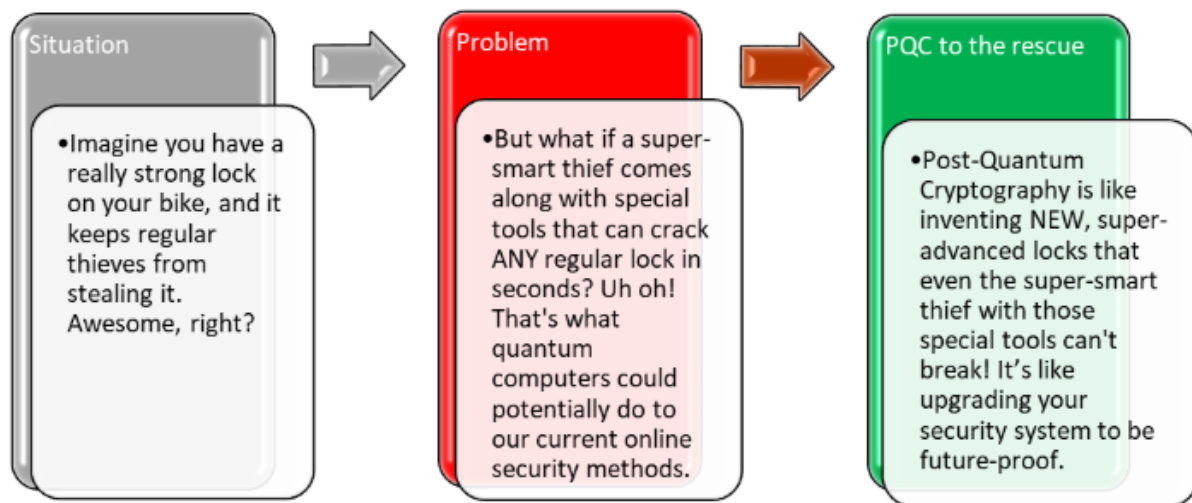


Figure 2. How can we think about post-quantum cryptography? Consider the analogy in which you have a bike (left) but there is a thief (middle) and PQC serves as a special lock to protect your bike (right).

computers get more powerful, the ways we protect our data now might not be enough. PQC encryption is designed to be super resistant to attacks, even from quantum computers.

In this activity, we are going to simulate a quantum cryptography protocol – the BB84 protocol – to send a message between two people (Alice and Bob) that will create a secret key they can use in the future to decode messages. In the real-life BB84 protocol, photons with different polarizations are sent across a communication channel using filters. Think of it like developing a secret handshake with light! Polarization refers to the orientation of the photon's oscillations – think of it like the direction a photon is pointing its electric field. Filters are like tiny gates that only let photons oscillating in a certain direction pass through. The receiver of the message uses similar polarization filters to guess in which direction the sender sent the photons. After sending a long string of photons, Alice and Bob can publicly share their filters to decide which bits to keep (when the filters match) and which to discard (when the filters do not match) for their secret key. An interesting feature of this protocol is that it has eavesdropping detection, as any effort to spy on the photons changes them. So, if Alice and Bob measure an error rate (the rate at which their filters do not match) that is higher than random, they can guess their communication is being eavesdropped and they do not use their communication to form their shared secret key.

How can we simulate the BB84 protocol? We will need something to represent the photons and how we filter them. Here we will use two types of wrapped candies: one type will have 0s or 1s written on them and one type will have red or green markings. Alice will select one of each randomly to assign her photon bit and color and record this information. Think of these two descriptors (bit and color) together as an analogy for the photon's polarization. Alice will then pass the bit candy to Bob, and Bob will randomly select a color candy to “view” the bit. Bob will record the bit and color information. This process will be repeated several times to represent a string of photons being passed through the quantum channel. Then, Alice and Bob will publicly compare their color information for the photons. If the colors match, they keep bits; otherwise, they discard the bits.

3.2.4. Procedure (work in groups of 2-4)

1. Select one person in your group to be Alice and one to be Bob. The other group members can act as Alice's and Bob's assistants and record the information about the "photons" that are passed during this activity.
2. Write 0 on 10 candies (or pieces of paper) and 1 on 10 candies (or pieces of paper) and place them in one jar. Shake them to ensure they are well mixed.
3. Put red tape on 10 candies (or pieces of paper) and green tape on 10 candies (or pieces of paper) and place them in a second jar. Shake them to ensure they are well mixed.
4. Alice selects one candy from each jar randomly and records the bit and color information.
5. Alice passes the bit candy only to Bob.
6. Bob selects one candy from the jar with different colors of candies and records the bit (from Alice) and color (from the jar) information.
7. Repeat steps 4-6 at least 5 times and up to 10 times.
8. Alice and Bob publicly share the color information for the "photons". If the colors match, they keep the information; otherwise, they discard the information.
9. Change roles in the group and repeat steps 4-8.

3.2.5. Results

Alice's results

Photon	Bit	Color	Keep or Discard?
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Bob's results

Photon	Bit	Color	Keep or Discard?
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

3.2.6. Post-Activity Questions

1. Describe your roles as Alice and Bob.

2. What did the candies and information on the candies represent in this activity?

The game Save Schrödinger's Cat will teach you about how the logic of quantum mechanical systems is fundamentally different from classical mechanics. The first key difference is the use of the phase of a qubit, which is a wave-like property that describes the timing of the wave relative to a reference position. Classical computer gates do not affect the phase of the bit, but many quantum gates do. This means there are additional tools to solve problems using quantum computers.

The second key difference comes from the interference of bits, which if we consider qubits as a wave-like object is basically wave interference. Wave interference is the addition of two or more waves. Interference is affected by the phase of the wave representing the qubit. A wave may begin at the height of zero or it may begin at its maximum height, or it may begin somewhere in between. It can even be negative! This means that when two waves are added together, the result can vary. For example, the maximum heights of the waves may align so the addition gives a larger height. This is called constructive interference. Or the resulting height may be zero if the waves are the same height, but one begins in the positive and the other in the negative. This is called destructive interference. Figure 3 shows examples of wave interference.

Wave interference plays an important role in the quantum gate called the Hadamard gate. As discussed earlier, this gate puts the qubit in a state of superposition, meaning the bit is in two or more states at once. This can be thought of as being able to run logic gates on the two states at the same time instead of one at a time in classical computers. Superposition is a property that only exists in a quantum state and, therefore, cannot be used in classical computers. For the purpose of this game, the zero (0) state is the cat being asleep and the one (1) state is the cat being awake. The phases of the qubit are represented as colors.

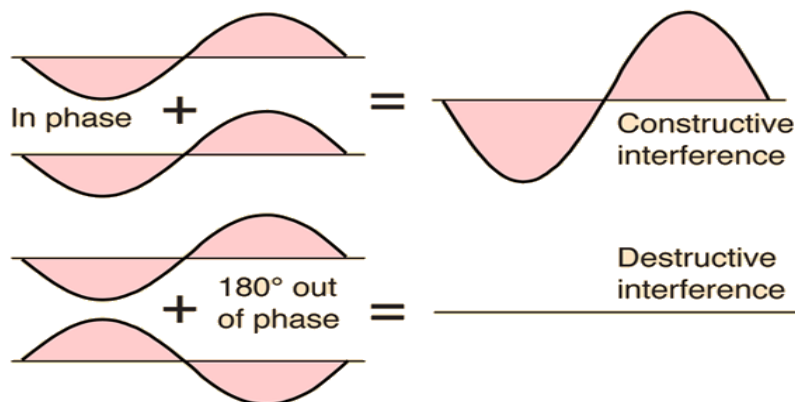


Figure 3. Constructive interference is when two waves add together to give a larger total height (top). Destructive interference is when two waves out of phase add together to give a wave with zero height (bottom). In quantum computing, quantum gates can impact the phase of the qubit. This is unique to quantum computing, as classical gates do not impact the phase of the bit. [This Photo](#) by Unknown Author is licensed under [CC BY-NC](#).

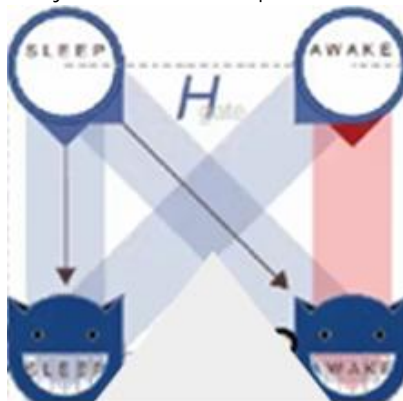
3.3.2. Materials (for each group)

- 8 blue cat/red cat tokens to act as qubits
- 8 yellow cat/green cat tokens to act as qubits
- 2 X gates
- 2 Y gates
- 2 X gates
- 2 S gates
- 2 H gates
- 1 CNOT gate
- 1 qubit gate phase change table
- 1 qubit Interference table

3.3.3. Procedure & Analysis (work in groups of 2-4)

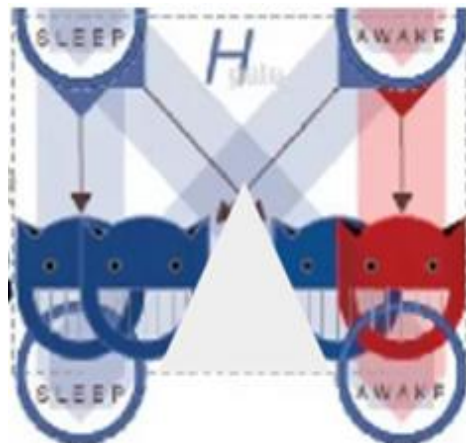
1. Lay out your QubitCats according to color.
2. Place an X gate where everyone can see it. Your first challenge is to wake up a blue cat. Place a blue cat token on the Sleep circle.
3. Notice that the blue path crosses over to the Awake side of the gate. Use your QubitCat phase change table to determine if a blue cat traveling on a blue path changes color. If the color does not change, place a blue cat on the Awake circle. If it does, then place the correct color of cat on the circle. Record this result in your table.
4. Repeat with the red, green, and yellow cats being in the initial state of Sleep. Record your results for the other colors.
 - a. Can you use an X gate to wake up any colored cat?
5. Your next challenge is to change an awake red cat into an awake blue cat.
 - a. Can you use an X gate to do this?
6. We will try other gates. Lay out the Z gate and place a red cat token on the Awake circle. Use your QubitCat phase change table to determine the effect of the red path on a red awake cat.

- a. Did you change the phase of the red awake cat to a blue cat? Note this in your results table.
 - b. Can you change an awake green cat into an awake yellow cat using the Z gate? Note this in your results table.
7. There are two more gates to work with, called the Y gate and the S gate. Try them out to see which one will change an asleep red cat into an awake yellow cat. Note the results in your table.
 8. There are more to quantum gates than just these four you have tested. Quantum computers can also place qubits into superposition. This is called preparing the state. Quantum computers use Hadamard (H) gates for this. Lay out your H gate and trace the two paths from Sleep and the two paths from Awake. Look for phase changes in any of the paths. Start a new challenge by placing a single blue cat on the Sleep circle in an H gate. This will result in a cat in two different states (Sleep and Awake), as shown in the picture below. Note that there is still only a single cat, but it is in a state of superposition (both Sleep and Awake), which you learned about in your earlier experiment with the coins.



9. Now, when we apply another gate, we have both states to work with. Use the superposition blue cat (two tokens) as the input for a Z gate. This means you “apply” a Z gate and move the asleep blue cat from the H gate to the Z gate Sleep circle and also the awake blue cat from the H gate to the Z gate Awake circle. Be sure to check for phase changes as you apply the Z gate.

10. To obtain a final result, we must make a measurement by applying a second Hadamard (H) gate. Unlike the first time you used the H gate, now there are tokens in both the awake and asleep states. This will result in four cat tokens on the bottom of the second Hadamard (H) gate, with (two on the Sleep circle and two on the Awake circle). There is an example of what this looks like below (though this is not for this challenge).



11. When two cat tokens occupy the same space, they interfere with each other and may cancel out. When used properly, there will be only one cat left because the other pair will interfere to cancel each other out. Use your QubitCat interference table to determine the final result of your cat.
- Did you wake your blue cat up? Record the result in the table.
12. Here is another challenge: start with a red sleeping cat. Apply an H gate and then a Y gate. Finish by taking a measurement by applying the second H gate.
- Do you have an awake cat?
13. There is one more very special gate unique to quantum computing, called a CNOT gate. The "C" stands for controlled, and two qubits are needed. The qubit at the top of the gate is the control bit and the target qubit is in the bottom part of the gate. If the control qubit is in the Sleep state, the target qubit follows the path on the left starting in the same state as the control qubit (here, Sleep). Examine the path to see if the target qubit has a change of state. If, instead, the control qubit is in the Awake state, the path on the right is followed (again, the target qubit follows the path on the right in the same state as the control qubit (here, Sleep). Examine that path and note if the state or phase of the target qubit is changed. As you can see, the state of the target qubit is dependent on

the state of the control qubit. This is an example of entanglement. Entanglement is a property of the two-qubit system and is a unique quantum phenomenon.

- Fill in your result table to show the state of an awake blue cat with an awake control cat in the CNOT gate. Repeat with an asleep blue cat and an asleep control cat.

3.3.4. Results

Gate	Starting		Ending	
	Color (phase)	State (bit)	Color (phase)	State (bit)
X	Blue	Sleep		
X	Red	Sleep		
X	Green	Sleep		
X	Yellow	Sleep		
Z	Red	Awake		
Z	Green	Awake		
Y	Red	Sleep		
S	Red	Sleep		
HZH	Blue	Sleep		
HYH	Red	Sleep		
CNOT (awake control)	Blue	Awake		
CNOT (asleep control)	Blue	Sleep		

3.3.5. Post-Activity Questions

- What did the cats and their colors represent in this game?

2. What is wave interference? How did this show up analogously in this game?

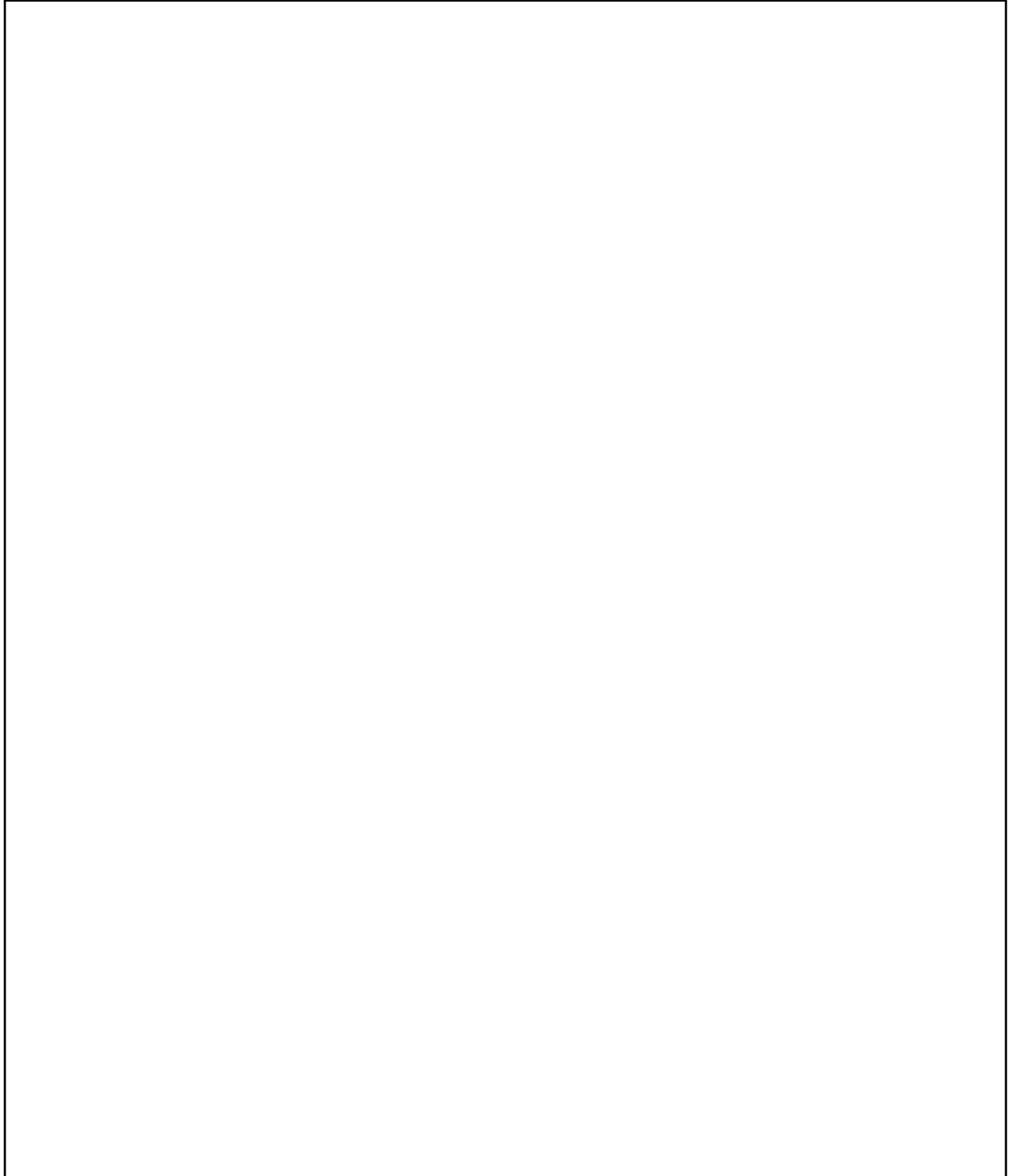
3. What is superposition? How did this show up analogously in this game?

4. What is entanglement? How did this show up analogously in this game?

5. What did you enjoy about this game?

4.2 Design Sketch

Sketch the designs of your cat qubit change below. Write the challenge of the change using: Take a (insert color) Cat that is (asleep/awake) and change it into a Cat that is (insert color) (insert state).



5. Supplemental Activity: Quantum Computing with Python and Qiskit

5.1 Additional Background

Quantum computing is a new paradigm of computation that leverages the principles of quantum mechanics to solve complex problems that are intractable for classical computers. Unlike classical computers that use bits to represent information as 0 or 1, quantum computers use qubits, which can exist in a superposition of both states simultaneously. This, along with other quantum phenomena like entanglement and interference, allows quantum computers to perform certain calculations much faster than classical computers.

In a nutshell two bits of a classical computer can be in four possible states (00, 01, 10, or 11), but only one of them at any time. This limits the computer to processing one input at a time (like trying one corridor in the maze). In a quantum computer, two qubits can also represent the exact same four states (00, 01, 10, or 11) at the same time. Quantum computing is the idea that we can use this quantum rule-breaking to process information in a new way - one that's totally different from how regular computers work.

To start programming quantum computers, you will use Python and Qiskit, an open-source quantum computing framework developed by IBM. Qiskit provides tools for creating, manipulating, and simulating quantum circuits.

5.2 Procedure

1. Before you begin, you need to install Python, Qiskit, and Jupyter Lab. Follow these steps to set up your environment:
 - a. Download Python from the official website (<https://www.python.org/downloads/>) and install it. Make sure to add Python to your system's PATH during installation.
 - b. Open your terminal or command prompt.
 - c. Use pip, the Python package installer, to install Qiskit ('pip install qiskit').
 - d. Use pip, to install a tool to visualize Qiskit ('pip install pylatexenc')
 - e. Use pip (or another method: <https://jupyter.org/install>), to install Jupyter Lab ('pip install jupyterlab')

- f. Open a Python interpreter by typing python in your terminal. Import Qiskit modules to verify the installation ('import qiskit' then 'print(qiskit.__version__)').
 2. Create simple quantum circuits to generate a superposition state and an entanglement state, which is a fundamental concept in quantum computing. Here's how you can do it in a Jupyter notebook:
 - a. Import the required Python packages.
 - i. `import qiskit as q`
 - ii. `from qiskit import quantum_info as qi`
 - iii. `from qiskit import QuantumCircuit`
 - b. Instantiate the quantum circuit class.
 - i. `qc = QuantumCircuit(2) # the argument (2) represents 2 qubits we're going to work with`
 - ii. `qc.draw(output = 'mpl')`
 - c. Instantiate the quantum circuit class.
 - i. `state_0 = qi.Statevector(qc)`
 - ii. `state_0.draw('bloch')`
 - d. Apply different gates on these qubits using different methods such as x, h, etc.
 - i. `qc.x(0) # x gate is applied on the first qubit`
 - ii. `qc.h(1) # Hadamard gate on second qubit`
 - iii. `qc.draw(output = 'mpl')`
 - e. Visualize the output of these circuits. (#to find out the output vector at the output of the circuit. let's name the state vector state_1)
 - i. `state_1 = qi.Statevector(qc)`
 - ii. `state_1.draw(output='bloch')`
 - f. Apply a CNOT gate on the first qubit.
 - i. `qc2 = QuantumCircuit(2)`
 - ii. `qc2.x(0) # x gate on the first qubit ...`
 - iii. `qc2.cx(0, 1) # CNOT gate on first qubit ...`
 - iv. `qc2.draw(output = 'mpl')`
 - v. `state_3 = qi.Statevector(qc2)`
 - vi. `state_3.draw('bloch')`

6. Sources

Quantum computing background

- <https://www.ibm.com/think/topics/quantum-cryptography>
- <https://risingwave.com/blog/beginners-guide-to-quantum-computing-for-dummies/>
- <https://www.youtube.com/watch?v=tsbCSkvHhMo>

BB84 protocol

- <https://arxiv.org/pdf/2110.01402>

Save Schrodinger's Cat

- https://www.youtube.com/watch?v=1OEjGWOUhM&ab_channel=PhysicsCentral
- <https://www.aps.org/learning-resources/save-schroedingers-cat>