BlochOut: Quantum Escape Game

A game-oriented introduction to quantum computing

Unnati Akhouri

Abstract

This guide accompanies the BlochOut quantum escape game, providing detailed explanations of quantum computing concepts, solutions to puzzles, and theoretical background for students learning quantum mechanics. The game introduces fundamental quantum gates, the Bloch sphere representation, entanglement, and multi-qubit operations through an engaging narrative format.

Contents

1 Introduction to Quantum Computing

1.1 What is computing and why is quantum interesting?

Computing is the act of calculating something. Our brains do it all the time—how much time until dinner, how many dumplings should I eat for dinner (the answer is always more), how much milk to water should I add for a good cup of tea. These seemingly simple questions represent sophisticated computational processes that our minds do throughout the day.

But what exactly makes something "computational"? At its heart, computing isn't just about crunching numbers—it's about processing information systematically to solve problems and make decisions.

A key component of computing is framing the question in a mathematical language that, based on a set of rules, returns reasonable and verifiable answers to us. This translation from everyday problems to mathematical representations is perhaps the most crucial step in all of computing.

Let's take a concrete example. If you associate the symbol "1" for every interval of time in which a sand clock needs to be restarted, then the problem of "how long until lunch?" can be re-framed into "how many symbols accumulate for the time to make dumplings"—let's say 1111. We can then recognize that these symbols can be added by a set rule for addition: 1+1+1+1 = 4.

This might seem trivial, but this truly is at the heart of computing: transforming a real-world question into abstract symbols, applying systematic rules to those symbols, and getting back a meaningful answer.

Therefore, by having symbols that store information and operations on that information, we perform computation. This is at the crux of anything that computes—your brain, your phone, your laptop, and yes, even your Roomba bumping around your living room.

Every computational system, regardless of its complexity, follows this same basic pattern:

- Representation: Encode information as symbols
- Manipulation: Apply rules to transform those symbols
- Interpretation: Translate the results back into meaningful answers

Realizing and utilizing this fact was behind the computing revolution. We learned that if we pre-code the symbols and the set of rules—that is, the actions we can perform on them—then a machine can execute these operations repeatedly, as many times as the resources allow for. This revelation was profound because it meant we could expand the type of calculations we could do far beyond simple arithmetic. Once the rules are set, the computer can be given gigantic computing tasks to work on—things that would take a human their entire lifetime to complete. Weather prediction, protein folding, analyzing the movement of galaxies—all of these became possible when we could automate the process of symbol manipulation.

It should be noted, though, that computers can perform very specific tasks for which they are trained—that is, provided with appropriate symbols and rules. When it comes to tasks that require abstract thinking, creativity, and nuanced understanding—thoughts, social movements, stories—the human brain remains remarkably capable. Dealing with ambiguity,

reading between the lines, and making leaps of intuition cannot (at present) be reduced to symbols and rules.
This need for storing information in the form of symbols is also what fundamentally limits computers. At the most basic level, computers are built from switches that can be either on or off, corresponding to the binary digits 0 or 1. These binary digits—or "bits"—are the atoms of digital information. Different combinations of these symbols correspond to different numbers. For example, with two switches or bits, we can represent four different numbers:

$$00 \longrightarrow 0 \tag{1}$$
$$01 \longrightarrow 1 \tag{2}$$
$$10 \longrightarrow 2 \tag{3}$$
$$11 \longrightarrow 3 \tag{4}$$
$$\tag{5}$$

Similarly, with three bits, we can represent 8 numbers (from 000 to 111, or 0 to 7 in decimal). Therefore with n bits, we can represent 2^n different values. This exponential relationship is both the power and the constraint of digital computing. Want to represent larger numbers or more complex information? You need more bits. Want to perform more sophisticated calculations? You need more bits to store the intermediate results.

This is where quantum computing enters the scene.

But before we dive into the quantum realm, it's worth appreciating just how far we've come with classical computing. The device you're reading this on contains billions of those simple on/off switches synced to render text, images, and interactive experiences [1].
Quantum computing offers the possibility of speeding up these calculations and more. In fact, certain quantum computing algorithms–calculators meant to perform certain tasks–have the potential to improve our ability to allocate flight routes and availability more efficiently in the airplane industry. Some algorithms are specifically designed to explore a range of mathematical outputs and pick the desired one. This can be helpful when analyzing risks or even in the discovery of molecules that perform a certain way such that they can be used in the pharmaceutical industry. Neat, isn't it! [2]

1.2 What is a Qubit?

Qubits are the fundamental building blocks of a quantum computer. Just like how classical computers are built of numerous bits–switches that can be in an *on* (1) or *off* (0) state–quantum computers are built of qubits. The difference is that while a classical bit can ever

[1] The figures in this text, except the introductory figure of the quantum cat, were generated using Anthropic Claude's svg code.

[2] It is important to add that currently quantum computers do not beat classical computers at computational tasks. However it is proposed that soon they may be able to compete. It is also important to add that over-hyping is dangerous and we should keep in mind that quantum computers are not a magic wand that will shoot us into space and create wizardy of sorts. All science in incremental and, in my **opinion**, celebrating it's journey for the sake of curiosity is probably more rewarding that obsessing over a profitable end.

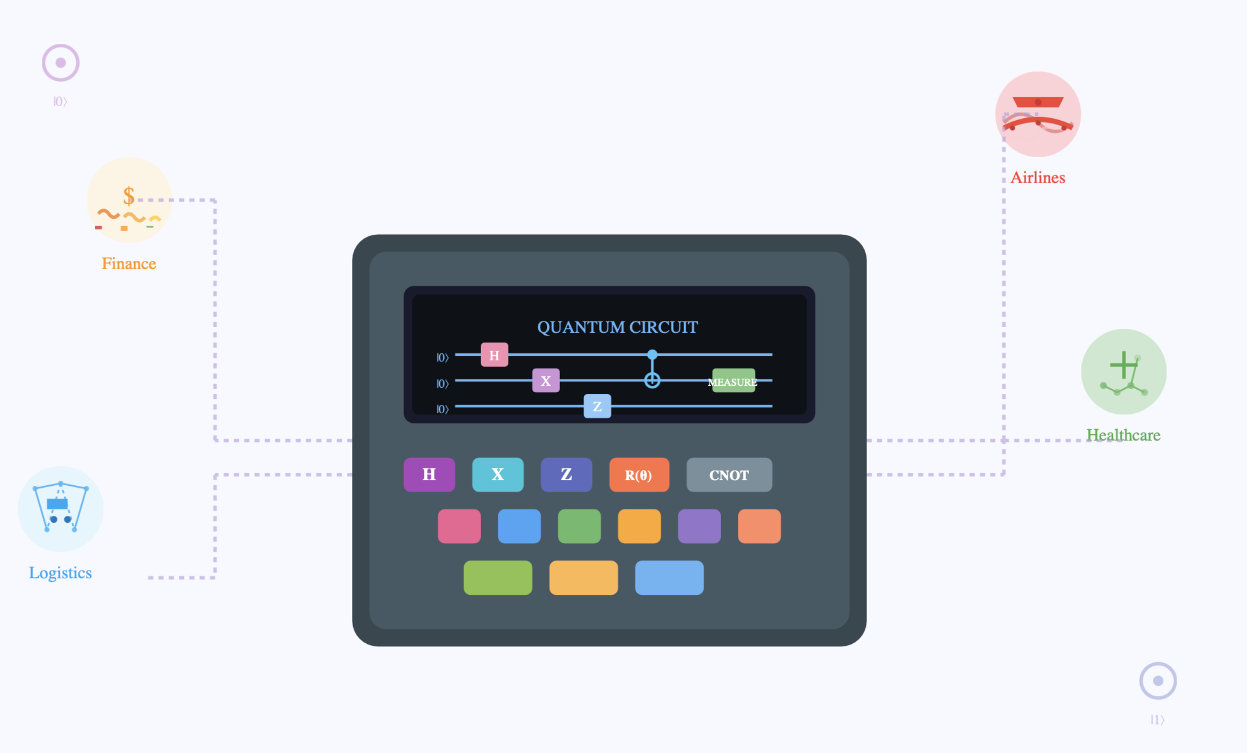


Figure 1: Quantum computing transforms complex real-world challenges through elegant circuit programming. While classical computers would require astronomical amounts of data—7.1 billion times more than all global digital storage by 2025—to simulate even simple molecules like Naphthalene, quantum systems accomplish this with 116 qubits. This quantum advantage unlocks revolutionary applications: airlines optimize intricate route networks in real-time, pharmaceutical researchers model molecular interactions for rapid drug discovery, financial analysts navigate multidimensional risk scenarios with unprecedented precision, and logistics networks achieve seamless global coordination. Algorithms on quantum computer represent computational leaps that can redefine how we perform computations and their limitations.

be in one of the two states, a qubit can be in a superposition.
But what does "superposition" actually mean, and why should we care?
Think of a coin spinning in the air—while it's spinning, it's neither definitively heads nor tails, but somehow "both" until it lands. Classical bits are like coins that have land and are either 1 or 0. Qubits are like coins that can remain spinning, existing in a combination of both states simultaneously.

The Quantum Bit

A **qubit** (quantum bit) is the fundamental unit of quantum information. Unlike classical bits that can only be 0 or 1, a qubit can exist in a *superposition* of both states simultaneously.
Mathematically, a qubit state is written as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{6}$$

where α and β are complex numbers called *probability amplitudes*, and $|\alpha|^2 + |\beta|^2 = 1$.

So, if you have 3 classical bits, you can represent one number at a time (like 101 representing 5). To explore all 8 possible 3-bit numbers, you'd need 8 separate calculations. But 3 qubits in superposition can represent all 8 numbers simultaneously—it's like exploring every possible path instead of trying each route one by one.
Take the example of searching through a phone book with 10,000 unsorted names. A classical computer might need to check up to 10,000 entries sequentially (you could speed up by using clever algorithms but only so much). A quantum computer using superposition can search through all entries simultaneously and provide a potential for dramatic speedup.
Manipulating the states of qubits is what makes quantum computers interesting, but also incredibly challenging. Superposition is fragile—any disturbance can cause qubits to "collapse" into definite 0s and 1s, which is why quantum computers are sensitive to error and noise and need serene environments to work.
In this tutorial we won't worry about making a quantum computer. Instead we'll focus on the building block–the qubit– after all, you are trapped in there!

1.3 The Bloch Sphere

The Bloch sphere is a way to visualize all possible states of a single qubit using a 3D sphere. Think of it as a map where every point on the sphere's surface represents a different quantum state that a qubit can be in.
The sphere has some key landmarks. The North pole represents the 0⟩ state (like a classical bit being "off"), while the South pole represents the 1⟩ state (like a classical bit being "on"). The equator is where the magic happens - these points represent superposition states, where the qubit is in a combination of both 0⟩ and 1⟩ simultaneously. What makes the Bloch sphere particularly useful is that it shows us quantum states exist on a continuum. Unlike classical bits that are strictly 0 or 1, qubits can exist at any point on this sphere's surface, giving us an infinite number of possible quantum states to work with.
Most generally a qubit state can be written in the matrix form as

$$\rho = \frac{1}{2}\begin{pmatrix} 1 + r_z & r_x + ir_y \\ r_x - ir_y & 1 - r_z \end{pmatrix} \tag{7}$$

The Bloch sphere is the 3-D sphere with radius 1 and any qubit state is a vector with coordinates (r_x, r_y, r_z).

Visualizing Quantum States

The **Bloch sphere** is a geometric representation of qubit states. Every point on the sphere's surface corresponds to a possible quantum state:

- North pole: $|0\rangle$ state
- South pole: $|1\rangle$ state
- Equator: Superposition states like $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- Interior points: Mixed states (not pure quantum states)

Any pure qubit state can be written as:

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{8}$$

where θ is the polar angle and ϕ is the azimuthal angle on the Bloch sphere.

2 Level 1 & 2: Basic Quantum Gates

But having infinite number of states to access is only interesting if we can move between these states. Mathematical operations allow us to do just that. Operations take a state as an input and map it to another state as the output. In the context of qubit systems, these operations are also called **gates**. Quantum gates are mathematical operations that act on one, two or few qubit systems and transform their states.

Since transformation is key to manipulating systems, gates are at the heart of quantum computing ecosystem.

This game intends to teach you about these few qubit gates that unlock endless mysteries of the quantum world.

A pro tip: Start each level by experimenting! Apply each gate and watch how your position changes on the Bloch sphere before trying to solve riddles.

2.1 The Fundamental Gates

The game begins with an ominous sounding message that appears on the screen:

You wake up and find yourself quantized. One moment you were human, the next—poof!—you are a qubit...the simplest quantum system possible. And what's more, the quantumland has trapped you in a Bloch sphere, a perfect geometric prison. The curved surface stretches in all directions, its translucent blue walls pulsing with vague symbols that look like matrices.

But wait a second, you see a door on the sphere—surely an anomaly that shouldn't exist in this geometrically perfect prison—and you wonder maybe it will get you out? But how to reach the door? You try to move, but the rules are different here. Linear motion is impossible. You can only rotate, superpose, dephase....

Suddenly, you hear a voice that seems to come from everywhere and nowhere at once. 'Ah, a new consciousness enters the quantum realm,' the voice resonates, its tone both amused and curious. 'Lost between states, are we? How delightfully uncertain!' The voice presents you with riddles and says 'finally a spin to my tale! If you wish to get back to your hooman form, you must solve some puzzles first!'

A control panel materializes before you—gates labeled with strange symbols: X, Y, Z, H, S, T. You recognize them somehow as quantum gates—tools to manipulate your very state of being. 'Choose wisely,' the voice purrs. 'The wrong transformation might scatter your consciousness across multiple universes. But the right one...' The voice trails off into a chuckle that sounds like static interference.

But before you can learn how top navigate this world, you must first learn what each of the moves do! Solve the riddles by finding the correct gate match! But remember, it is important to figure out how many times you must apply a gate to come back to yourself! You realize this will surely help you later on even if you applied the wrong gate! You can undo the action by knowing precisely how many times you must apply the same gate!

It would seem that the Quantum Cat has trapped your consciousness in a qubit state inside a Bloch sphere. To escape the Bloch sphere, you must learn how to move around the sphere. The following gates will help you do just that!

Pauli Gates

The Pauli gates are the most fundamental single-qubit operations:

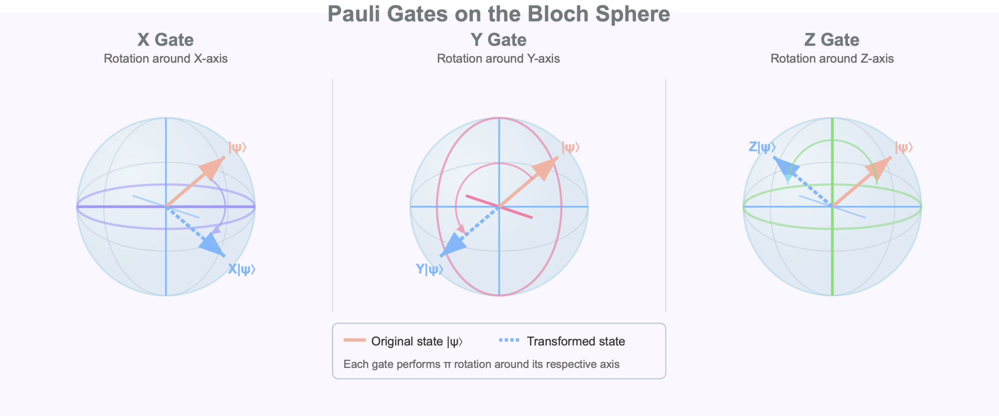
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{(Bit flip)} \tag{9}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \text{(Bit and phase flip)} \tag{10}$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{(Phase flip)} \tag{11}$$

To apply any of these gates, just press the button in the game. However, if you are interested in the mathematics, the evolution is given by

$$\rho_f = G\rho_i G^\dagger \tag{12}$$



where G is a gate, ρ_i is the initial state and ρ_f is the final state.
Instead of memorizing matrices, we can visualize gates as rotations on the Bloch sphere:

- X gate: 180° flip around the X-axis (like flipping a coin)
- Y gate: 180° flip around the Y-axis (flip + phase change)
- Z gate: 180° rotation around Z-axis (phase change only)
- H gate: Creates superposition by rotating to the equator

Gate Properties

Each Pauli gate is:

- **Hermitian**: $X^\dagger = X$, $Y^\dagger = Y$, $Z^\dagger = Z$
- **Unitary**: $XX^\dagger = I$, etc.
- **Involutory**: $X^2 = Y^2 = Z^2 = I$ (applying twice gives identity)

The game also introduces the player to some additional single qubit gates that are often used in quantum computing. These include:

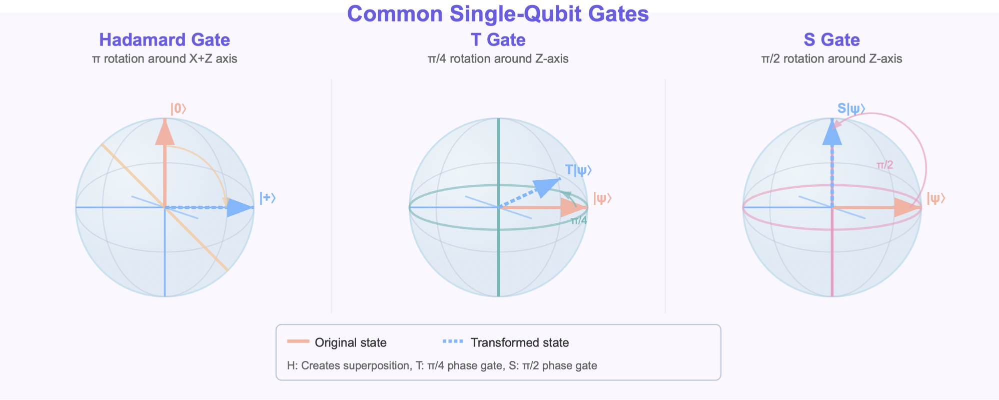
Hadamard Gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{13}$$

The Hadamard gate is used to create an superposition of the basis states:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \tag{14}$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \tag{15}$$


Common Single-Qubit Gates
Hadamard Gate
π rotation around X+Z axis
|0⟩
|+⟩
T Gate
π/4 rotation around Z-axis
T|ψ⟩
|ψ⟩
π/4
S Gate
π/2 rotation around Z-axis
S|ψ⟩
π/2
|ψ⟩
Original state
Transformed state
H: Creates superposition, T: π/4 phase gate, S: π/2 phase gate

Phase Gates

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \quad \text{(Quarter phase, } S^4 = I\text{)} \tag{16}$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \quad \text{(Eighth phase, } T^8 = I\text{)} \tag{17}$$

The S and T gates that can add a phase shift between the basis states $|0\rangle$ and $|1\rangle$. Equivalently, it rotates the Bloch vetor around the z-axis by $\theta = \frac{\pi}{2}$ for S and by $\theta = \frac{\pi}{4}$ for the T gate.

The first level of the game teaches the player that these gates can be applied a certain number of times such that their action becomes identity i.e. as if nothing happened to your state. In the world of quantum computing, knowing this is helpful because it tells us when states stay unaffected by certain actions and if an erroneous operation occurs, how to retrieve back the original state.

Did you know

- The Hadamard (H) gate and the S gate are part of a set called Clifford group gates [a]. These are a set of gates that can be classically simulable. This means that a classical computer can perform all algorithms and tasks that are based entirely of these computations. Effectively it means that if your computation or *algorithm* was entirely made up of these gates then they could be run on a classical computer with no need for quantum speedup!

- While Clifford gates are great in that they make some quantum algorithms classical simulable and therefore easy to study, they lack in that you can reach all possible quantum states using just those gates!

- T gate is an example of a gate whose transformation cannot be mimicked by S or H gate since $T = \sqrt{S}$ which is hard to perform!

[a] You will meet the final member of the Clifford group later in this pdf!

2.2 Riddle Solutions

Level 1 Riddles and Solutions

1. *"Flip me over, flip me back, apply me twice and there's no lack."*
   Answer: X gate - The bit flip gate returns to identity when applied twice.

2. *"Around the Y-axis I'll rotate, apply me twice and seal your fate."*
   Answer: Y gate - Rotation around Y-axis, $Y^2 = I$.

3. *"I change your phase but not your chance."*
   Answer: Z gate - Changes phase of $|1\rangle$ but preserves probabilities.

4. *"I'm equal parts X, Y, and Z, apply me twice and you will see."*
   Answer: H gate - Creates equal superposition, $H^2 = I$.

5. *"A quarter turn around the Z, apply me four times, if you please."*
   Answer: S gate - $S^4 = I$.

6. *"One-eighth of a full rotation, apply me eight times for restoration."*
   Answer: T gate - $T^8 = I$.

Learning objective: Understanding that gates have specific orders (how many times you apply them to return to identity).

Single qubit gates are the simplest yet essential building blocks of quantum circuits. Since they act on a single qubit, at most they can only affect the state of a qubit locally and therefore cannot generate entanglement between qubits [3]. Still they are important as they allow for local manipulation of qubit states.
After being acquainted with the 1 qubit gates, the players must now solve the riddle to get out of the Bloch sphere. The quantum cat has the following message

> The voice returns, this time with a hint of mischief, 'Ah, now that you know how to move around, lets see you try to get out!'
> 'See that gate, don't you? Lets see of you can escape this Bloch I have created for you!'
> 'Oh and don't you worry,' the voice adds, 'there will be puzzles guiding you all along the way...'
> 'But remember, any mistake you make you must undo the actions by coming back to the starting state else you may lose yourself in the infinite possibility mess!'

The crucial lesson in this level is to recognize that if we make a wrong guess to a gate, remember to apply it as many times until you nullify the action of that gate!

[3] Which as we will see later is what really makes quantum stuff powerful

Level 2 Riddles and Solutions

1. X Gate - "Turn 1 into 0 and 0 into 1"
2. Y Gate - "Now lets rotate about the Y"
3. Z Gate - "Without changing your chance, lets do the phase dance"
4. S Gate - "Pass through the gate that takes half the amount of times as the longest gate" (T takes 8, so S takes 4, which is half)

Target State: The final state after applying X → Y → Z → S to the initial state.
Learning Objectives: Apply the gates learned in Level 1 in the correct sequence

3 Level 3: Rotation Gates and Universality

You may have noticed that the gates above allowed only discrete operations...it is as if you were allowed to only jump two steps forward everytime you move. What an awful life when the dumpling you want to eat is one step away but all you can ever do is take two steps... Well, not in this quantum land. The quantum cat returns to teach you about continuous rotations.

Having mastered the basic quantum gates, you face a new challenge. The voice speaks again: 'Well done, quantum traveler. But discrete operations can only take you so far...'
You notice the control panel has transformed. The discrete gate buttons have been replaced by continuous sliders. 'Welcome to the world of parameterized quantum rotations,' the voice continues.
These rotation channels allow you to apply transformations with varying strength, unlike the fixed rotations of the basic gates. The angle parameter gives you precise control over your quantum state.
'Use the Rx, Ry, and Rz channels to navigate the quantum realm with greater finesse. Adjust the sliders to control the rotation angle, then apply the channel to see its effect on your qubit state.'

3.1 Parameterized Rotation Gates

Continuous Rotations

Unlike the discrete Pauli gates, rotation gates allow continuous control:

$$R_x(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{18}$$

$$R_y(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{19}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{20}$$

Continuous rotations can be thought of as the equivalent of how we move around our when we walk. Rather than taking discrete steps, we can take any continuous value. Similarly, in the quantum world, this allows us to move the Bloch vector in any direction on the sphere.

Level 3 Riddles and Solutions

Use the three rotation gates Rx, Ry, and Rz gates with precise angle control. The target state requires specific rotation angles. So experiment with different angle combinations to reach the target.

1. $Rx(\theta)$: Rotation around X-axis by angle
2. $Ry(\theta)$: Rotation around Y-axis by angle
3. $Rz(\theta)$: Rotation around Z-axis by angle

Learning objective: Observe the target Bloch vector position. Use Ry to adjust latitude. Use Rz to adjust longitude and finally fine-tune with Rx if needed

3.2 Universal Gate Sets

Wouldn't it be great if instead of all these gates we could have just a few handful ones to do all the movement we want? Well, we're in luck because that's exactly what universal gate sets do! For a given qubit system size they correspond to the set of gates that can perform any movement! Here's the exact definition:

Quantum Universality

A set of gates is **universal** if any quantum computation can be approximated to arbitrary precision using only those gates.
One universal set consists of:

- $R_y(\theta)$ - Rotation around Y-axis
- $R_z(\phi)$ - Rotation around Z-axis
- Phase gate: $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{pmatrix}$

The voice returns, this time with a different tone—more instructive, yet somehow more cryptic. If you must move alone in this Bloch sphere, it intones, you shall not need all these operations. All you need is the universal set of gates.
"With these alone," the voice continues, "any transformation can be achieved. Any state can be reached. Any door... unlocked." You reach toward them, but the voice interrupts with a warning that sends a chill through your quantum state.
"However," it whispers, "the parameters must be precise. The angles, the phases, the timing—all must be exact. If you do not use the correct variables to build your gate, you may find yourself trapped in an infinite composition."

Universal Decomposition

Any single-qubit unitary can be written as:

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) \tag{21}$$

You can check this mathematically by writing the composition of the gates on the right side of the equation and comparing it with a general single qubit gate given by:

(22)

Level 4 Solution Hint

The game asks you to recreate a target state using the universal gates. The key insight is that you need exactly the sequence $e^{i\phi}R_z(\gamma)R_y(\beta)R_z(\alpha)$ where the parameters are multiples of π.
Try values like:

- $\alpha = 0.4\pi$
- $\beta = 0.3\pi$
- $\gamma = 0.2\pi$
- $\phi = 0.1\pi$

Learning Objective: Try tuning the gate parameters until you reach the target state so that you may exit the sphere. In real world, fine-tuning parameters is often the an important job of a scientist!

4 Levels 4 and 5: Two-Qubit Gates and Entanglement

You input the final sequence of gates, and the door on the Bloch sphere slides open with a resonant hum. Relief washes over your quantum state as you rush through the opening.
But something isn't right. As your consciousness settles, you realize you're still in a Bloch sphere—just a different one. And there, across the curved quantum landscape, is Sonalika, your friend who disappeared a week before you did. Their quantum signature flickers with recognition when they see you.

Now, you may think that your job is done but alas, quantum-ness is nothing without a little entanglement. And for entanglement–the ability of one system to affect the evolution of a spatially separated system simply because they share correlations–you NEED two qubit gates.
And so the cat returns to teach you about the two-qubit gates.

"Impressive progress," it says, "but did you really think escape would be so simple? You and your friend are now trapped in entangled Bloch spheres.... "In this realm, the single qubit gates are too simplistic. Now, you must harness the power of entanglement and multi-qubit gates."
"You made it!" Sonalika calls out, their voice carrying across the quantum void. "I've been waiting for—" They stop mid-sentence as both of you suddenly lurch sideways. When you shifted your position, Sonalika moved too—perfectly mirroring your rotation but in the opposite direction.
You try to move toward Sonalika, but the more you struggle to approach, the further they seem to drift away. When you rotate clockwise, they rotate counterclockwise. When you try to shift your phase, theirs shifts in complementary patterns.
"The only way out," the voice continues, "is to master the gates of entanglement. The PSWAP to exchange your positions, the CNOT to flip states conditionally, and the CR for controlled rotations. Only by working together can you break free of this quantum prison."
Sonalika looks at you with determination in their eyes. "We can figure this out," they say. After all, rotation gates are universal—they can create any single-qubit transformation with the right sequence and parameters. But we'll need to coordinate our actions perfectly. When I apply my gate, you'll need to apply yours at exactly the right moment."
You notice three new controls have appeared on your quantum interface: PSWAP, CNOT, and CR, each with parameters that need precise calibration. A wrong move could entangle you both more deeply, perhaps irreversibly.
"Ready to try?" Sonalika asks, hovering a finger over their control panel. The true test has only just begun.The challenge is clear: use rotation gates with the right parameters, along with two-qubit entangling operations, to reach the target state that will finally unlock your escape from this quantum puzzle.

4.1 Two-Qubit Systems

A two-qubit system has four basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$.

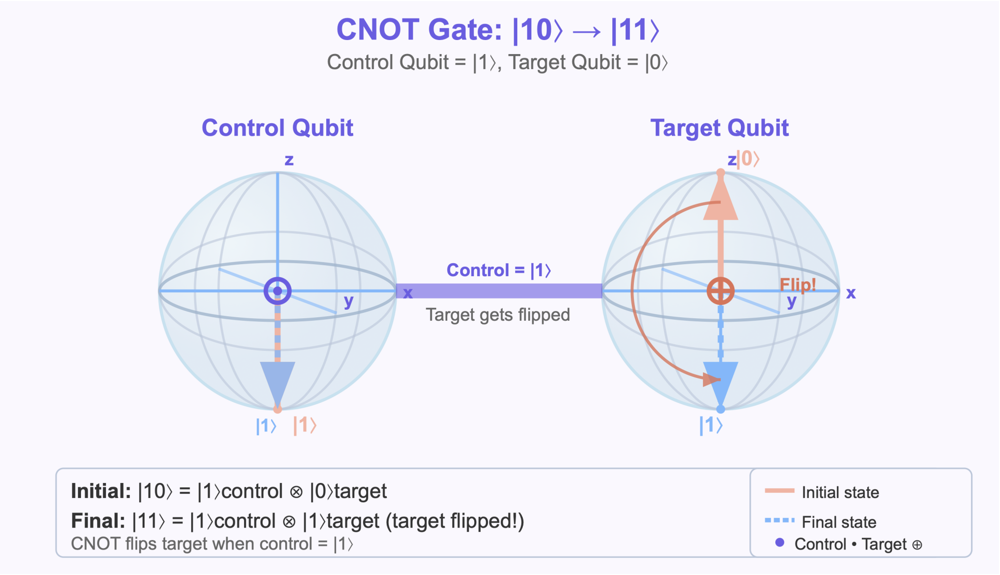
The general state is:

$$|\psi\rangle = \alpha_{00}\,|00\rangle + \alpha_{01}\,|01\rangle + \alpha_{10}\,|10\rangle + \alpha_{11}\,|11\rangle \tag{23}$$

4.2 Entangling Gates

CNOT Gate

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{24}$$



CNOT Gate: |10⟩ → |11⟩

Control Qubit = |1⟩, Target Qubit = |0⟩

Initial: |10⟩ = |1⟩control ⊗ |0⟩target

Final: |11⟩ = |1⟩control ⊗ |1⟩target (target flipped!)

CNOT flips target when control = |1⟩

The CNOT gate flips the target qubit if the control qubit is $|1\rangle$:

$$\text{CNOT}\,|00\rangle = |00\rangle \tag{25}$$

$$\text{CNOT}\,|01\rangle = |01\rangle \tag{26}$$

$$\text{CNOT}\,|10\rangle = |11\rangle \tag{27}$$

$$\text{CNOT}\,|11\rangle = |10\rangle \tag{28}$$

Did you know

The CNOT gate along with the H S and T single qubit gates forms a universal gate set implying that any two qubit rotation can be written as a combination of these gates.

You input the final sequence of gates, and the door on the Bloch sphere slides open with a resonant hum. Relief washes over your quantum state as you rush through the opening.

But something isn't right. As your consciousness settles, you realize you're still in a Bloch sphere—just a different one. And there, across the curved quantum landscape, is Sonalika, your friend who disappeared a week before you did. Their quantum signature flickers with recognition when they see you.

"You made it!" Sonalika calls out, their voice carrying across the quantum void. "I've been waiting for—" They stop mid-sentence as both of you suddenly lurch sideways. When you shifted your position, Sonalika moved too—perfectly mirroring your rotation but in the opposite direction.

The mysterious voice returns, now sounding amused. "Congratulations on solving the first puzzle," it says. "But did you really think escape would be so simple? You and your friend are now trapped in entangled Bloch spheres. Every action one takes affects the other."

You try to move toward Sonalika, but the more you struggle to approach, the further they seem to drift away. When you rotate clockwise, they rotate counterclockwise. When you try to shift your phase, theirs shifts in complementary patterns.

"The only way out," the voice continues, "is to master the gates of entanglement. The PSWAP to exchange your positions, the CNOT to flip states conditionally, and the CR for controlled rotations. Only by working together can you break free of this quantum prison."

Sonalika looks at you with determination in their eyes. "We can figure this out," they say. "But we'll need to coordinate our actions perfectly. When I apply my gate, you'll need to apply yours at exactly the right moment."

You notice three new controls have appeared on your quantum interface: PSWAP, CNOT, and CR, each with parameters that need precise calibration. A wrong move could entangle you both more deeply, perhaps irreversibly.

Parameterized SWAP (PSWAP)

$$\mathrm{PSWAP}(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & i\sin\phi & 0 \\ 0 & i\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{29}$$

This gate partially swaps two qubits with a phase factor. This gate can be used to generate or destroy entanglement. Starting with a pure state of two qubits where one qubit is in $|1\rangle$ state and the other in $|0\rangle$ state, the gate can generate bell state. Conversely, starting from entangled state, the qubits would have ended up en-entangled.

Controlled Rotation (CR)

$$\mathrm{CR}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix} \tag{30}$$

Applies rotation $e^{i\theta}$ to the target qubit only when the control is $|1\rangle$.

4.3 Bell States and Maximum Entanglement

Bell States

The four maximally entangled two-qubit states:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{31}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{32}$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{33}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{34}$$

These states cannot be written as a product of single-qubit states - they are *entangled*.

4.4 Creating Bell States

Bell State Circuit

To create the Bell state $|\Phi^+\rangle$ from $|00\rangle$:

1. Apply Hadamard to first qubit: $H \otimes I\,|00\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$
2. Apply CNOT: $\mathrm{CNOT}\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

5 Density Matrices and Mixed States

5.1 Pure vs Mixed States

Density Matrix Formalism

For a pure state $|\psi\rangle$, the density matrix is:

$$\rho = |\psi\rangle \langle\psi| \tag{35}$$

For mixed states (statistical mixtures), we have:

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| \tag{36}$$

where p_i are classical probabilities.

5.2 Reduced Density Matrices

For entangled systems, we often look at subsystems using the **partial trace**:

If we have a two-qubit state ρ_{AB}, the reduced density matrix for qubit A is:

$$\rho_A = \mathrm{Tr}_B(\rho_{AB}) \tag{37}$$

Important Note

Even if the full two-qubit state is pure (represented by a point on the Bloch sphere), the individual qubits may be in mixed states when entangled. This is visualized in the game as the individual Bloch spheres showing mixed states when the qubits are entangled.

Levels 5+: Entanglement Challenges

- Coordinate actions between both qubits
- Use CNOT to create initial entanglement
- Apply CR gates for controlled rotations
- Use PSWAP for partial state exchange
- Remember: individual qubit states become mixed when entangled

5.3 Advanced Strategy Tips

Expert Tips

1. **Plan your sequence**: Write down the target state and work backwards
2. **Use symmetry**: Many quantum operations have elegant symmetries
3. **Check progress**: Monitor the distance to target state continuously
4. **Undo mistakes**: Remember which gates are self-inverse
5. **Coordinate timing**: In two-qubit levels, timing of operations matters

6 Mathematical Appendix

6.1 Matrix Representations

Rotation Gates

$$R_x(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X \tag{38}$$

$$R_y(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y \tag{39}$$

$$R_z(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z \tag{40}$$

Single-Qubit Gate Cheat Sheet

6.2 Acknowledgments

I thank Sonalika Purkayastha, Jackson Henry, Shio Sakon and Anthony Katona for their comments in the process of developing the code.

Gate	**Matrix**	**Action**	**Order**
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Bit flip	$X^2 = I$
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	Bit+phase flip	$Y^2 = I$
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Phase flip	$Z^2 = I$
H	$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	Superposition	$H^2 = I$
S	$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	Quarter phase	$S^4 = I$
T	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	Eighth phase	$T^8 = I$

Table 1: Single-qubit gate reference