# Quantum Path Navigator


| | |
|---|---|
| Author: | Jakub Vaniš |
| Date: | 16.5.2025 |
| Email: | jakub.vanis@proton.me |
| Institution: | Czech Technical University in Prague |


## Introduction

*Quantum Path Navigator* is a game built with Python's `pygame` library in which the player pilots a small spaceship through a branching tree of possible paths. At each decision point, the direction the ship takes is determined by the outcome of a quantum measurement.

## Running the Game

To run the game:

1. Make sure Python 3.7+ is installed.
2. Install dependencies with: `pip install pygame numpy`
3. Run the game using: `python final_game.py`

## World and Objective

The game world consists of a 6-level 2D binary tree, resulting in a total of 127 nodes. At each node, there's a right (green) and a wrong (red) path. Which path is taken at each node is determined by a measurement of a qubit. Taking the wrong path results in a "Game Over" message being displayed and ending the game. The right and wrong paths at each node are randomly assigned at the start of the game.

The player's goal is to successfully navigate from the starting point (left side) to any of the leaf nodes on the far right. If they reach the yellow goal zone at the end of the tree a "You Win!" message is displayed.

## The Qubit System

At the heart of the game is a simulated qubit, initialized at each node in a random pure state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle\,, \quad \text{with } |\alpha|^2 + |\beta|^2 = 1\,.$$

The player can manipulate this qubit before each measurement using quantum gates. The gates available are:

- **Hadamard (H):**

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- **Pauli-X (X):**

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- **Pauli-Z (Z):**

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- **Phase Gate (S):**
$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

The player uses the keyboard inputs in the brackets to apply each gate. The order of operations matters, and the most recent gate sequence is shown on screen.

## Measurement and Decision-Making

Once the player reaches a fork, the game performs a projective measurement of the qubit in the $X$-basis:
$$\left\{ |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \right\}.$$
If the outcome of the measurement is $|+\rangle$, the ship moves to the next node; if $|-\rangle$, the game ends in failure. The probability of measuring $|+\rangle$ is
$$P(+) = \left| \frac{\alpha + \beta}{\sqrt{2}} \right|^2.$$

## Visual Feedback

The game includes real-time visual feedback to help players understand how the quantum gates influence the measurement probabilities:

- The sequence of gates applied since the last measurement.
- Red and green vertical bars representing $P(+)$ and $P(-)$ respectively.
- The current quantum state in Dirac notation.
- A Bloch sphere projected in the $xz$-plane showing the qubit's orientation.

## Gameplay Loop

The game runs at 60 frames per second. Each frame, it:

1. Handles input for applying gates.
2. Moves the ship toward its current destination node.
3. Triggers measurements upon arrival.
4. Updates visuals, including state projections and probabilities.

The ship automatically moves towards the next node once the decision is made; the only control the player has is in preparing the qubit state before each node.

## Conclusion

This project offers an intuitive, interactive way to explore quantum measurements and gate operations. It's designed both as a game and as a teaching tool, giving players hands-on experience with how quantum systems behave under manipulation and measurement.