Quantum Chess- Project Report

Submitted By:

- Ayan Mohammad Zakriya – K224728
- Hamza Yousuf – K224748
- Sami-ur-Rehman – K224673

BSCYS Students, FAST NUCES, Karachi Campus
Spring Semester, 2025

1. Introduction

Quantum Chess is a variant of traditional chess that incorporates concepts from quantum mechanics such as superposition, entanglement, and probabilistic collapse. The goal of this project is to provide a playable and educational experience that challenges conventional thinking about game mechanics and logic.

This project was developed as part of our Programming Fundamentals and Object-Oriented Design coursework. It aims to introduce basic quantum principles in a gamified and visual form, encouraging players to think in probabilistic and uncertain scenarios.

2. Objectives

- To implement a functional chess game using Python and Pygame.
- To add quantum mechanics elements to enhance complexity and strategic depth.
- To create an interactive and user-friendly interface.
- To demonstrate understanding of object-oriented programming and software architecture.
- To allow users to play against a computer with basic AI capabilities.

3. Technologies Used

- Programming Language: Python 3.8+
- Graphics Library: Pygame
- GUI Framework: GTK3 (via PyGObject) – used for advanced UI elements like the welcome screen
- SVG Rendering: CairoSVG
- Development Environment: VS Code / PyCharm
- Version Control: Git

4. System Design & Architecture

4.1. Code Structure

- main.py – Handles UI, game loop, event handling, and rendering.
- board.py – Manages game logic, board state, piece movement, and quantum behaviors.
- Assets/ – Contains piece and board images in SVG format.

4.2. Class Design

- Piece: Base class for all chess pieces.
- QuantumPiece: Inherits from Piece; adds superposition and entanglement properties.
- Board: Maintains the current state, move legality, and piece positions.
- GameState: Controls turn logic, win conditions, and history.
- AI: A basic opponent that selects from valid moves and applies quantum collapse.

5. Key Features

Feature	Description
Superposition	A piece can exist in two positions simultaneously with a 50/50 probability.
Entanglement	Two pieces can become linked; changes to one affect the other.
Collapse Mechanism	Upon interaction, a quantum piece collapses to a classical state randomly.
Player Selection	Choose White or Black side from a welcome screen.
Animated Interface	Smooth transitions and modern look using SVG graphics and Pygame.
Move Log Panel	Displays the latest moves to help players strategize.
Computer Player	Basic AI opponent implementing quantum-aware logic.

6. Testing & Evaluation

6.1. Test Cases

Component	Test Performed	Result
Piece Movement	Normal and quantum moves validated	Passed
Collapse Logic	Randomized and deterministic checks	Passed
UI Flow	From welcome screen to game loop	Passed
SVG Rendering	All assets loaded and scaled properly	Passed
AI Behavior	Legal move generation and execution	Passed

6.2. Bugs Resolved

- Initial overlap issues in entanglement rendering.
- Board refresh lag after quantum move.
- SVG loading path corrections.

7. Results

We successfully implemented a playable version of Quantum Chess, supporting all classical rules along with key quantum features. The user interface is intuitive, and the game introduces a novel strategy paradigm by incorporating uncertainty and probability into classic chess gameplay.

8. Challenges Faced

- SVG Rendering: Loading SVGs dynamically and scaling them to board size required using cairosvg and involved troubleshooting image clipping and positioning.
- Quantum Logic: Managing game states for superposed and entangled pieces involved complex data structures and event handling.
- UI Complexity: Creating a responsive UI with animations while keeping performance optimal was a technical challenge.

9. Conclusion

Quantum Chess represents a fusion of traditional gaming and modern science. Through this project, we enhanced our understanding of object-oriented programming, graphical game development, and the application of theoretical principles in software. The final product is not only functional but educational and fun.

We believe this project can be expanded further by implementing multiplayer functionality, stronger AI, and enhanced quantum rules. It serves as a strong foundation for further research and experimentation.

10. Future Work

- Multiplayer Support (Local & Online)
- Advanced AI using Minimax with Probabilistic Trees
- Enhanced Graphics with Sound Effects
- Tutorial Mode to Explain Quantum Rules
- Support for saving/loading games