PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Mini Project Report On Ludo Game

Course Code : CSE3080 **Course Title :** Quantum Computing

Submitted by

S.No	Roll.No	Names
1.	20201CAI0220	Jonnagiri Aishwarya
2.	20201CAI0222	Madhu P Bannigol
3.	20201CAI0042	Y Sai Yashwanth Reddy
4.	20201CAI0011	Kuruba Yashwanth

In partial fulfillment for the requirement of 6th

Under the guidance of: **Date of Submission:**

Dr. Jayakumar V 01-06-2023

School Of Computer Science Engineering

Presidency University,Bangalore.

Introduction:

Quantum computing is an emerging field that leverages the principles of quantum mechanics to perform complex computations. In this report, we present a simplified implementation of a Ludo game using quantum computing with Qiskit, a popular Python library for quantum computing.

Game Overview:

Ludo is a popular board game that involves rolling dice and moving game pieces across a board to reach a specific destination. Our implementation focuses on simulating a single player's turn using a quantum circuit to generate random dice rolls. The game continues until a player reaches or exceeds position 100, indicating a win.

Implementation Details:

1. Quantum Circuit Construction: We use the Qiskit library to construct a quantum circuit representing a player's turn. The circuit consists of two qubits and two classical bits. It starts by applying a Hadamard gate (H-gate) to the first qubit to put it into a superposition of states. Then, a controlled-X gate (CNOT) is applied between the first and second qubits to entangle them. Finally, measurements are performed on both qubits, storing the results in the classical bits.
2. Simulating a Player's Turn: To simulate a player's turn, we create an instance of the player's turn quantum circuit and execute it using a simulator backend provided by Qiskit. The simulator performs a single shot measurement of the circuit, providing us with a binary string representing the measurement outcome.
3. Dice Roll and Position Update: The measurement outcome from the quantum circuit represents the dice roll for the player's turn. We convert the binary string to an integer and add 1 to obtain the dice value. The player's position is then updated by adding the dice value. If a player's position reaches or exceeds 100, the game ends, and the player is declared the winner.

Code :

```
from qiskit import QuantumCircuit, execute, Aer
from qiskit.visualization import plot_bloch_multivector, plot_histogram

# Create the quantum circuit for a single player's turn
def create_player_turn_circuit():
    circuit = QuantumCircuit(2, 2)
    circuit.h(0)
    circuit.cx(0, 1)
    circuit.measure([0, 1], [0, 1])
    return circuit

# Simulate a single player's turn
def simulate_player_turn():
    circuit = create_player_turn_circuit()
    backend = Aer.get_backend('qasm_simulator')
    job = execute(circuit, backend, shots=1)
    result = job.result()
    counts = result.get_counts()
    return list(counts.keys())[0]
```

```
# Main game loop
def play_ludo_game():
    num_players = 4
    player_positions = [0] * num_players

    while True:
    for player in range(num_players):
    print(f"Player {player+1}, it's your turn!")
    dice_roll = simulate_player_turn()
    dice_value = int(dice_roll, 2) + 1
    print(f"You rolled a {dice_value}!")
    player_positions[player] += dice_value
    print(f"Your position: {player_positions[player]}")

    # Check for winning condition
    if player_positions[player] >= 100:
        print(f"Player {player+1} wins!")
        return

# Start the game
play_ludo_game()
```

Results and Observations:

Running the code in a Jupyter Notebook with Qiskit installed allows us to play a simplified version of Ludo using quantum computing. Each player takes turns, rolling the dice using the quantum circuit, and updating their positions accordingly. The game continues until a player reaches or exceeds position 100.

Limitations and Future Improvements:

1. Simplified Gameplay: The implemented version of Ludo is highly simplified and lacks many of the rules and mechanics present in the actual game. For a more comprehensive implementation, additional features such as player turns, game board management, and player interactions need to be incorporated.
2. Quantum Dice Roll: While the quantum circuit provides a random outcome, it is important to note that it does not necessarily generate truly random numbers. Quantum circuits are probabilistic, and the measurement outcomes are determined by the underlying quantum state and measurement basis. Further research and more complex quantum algorithms may be required to achieve true randomness.
3. Performance Considerations: The current implementation utilizes a simulator backend, which is suitable for small-scale simulations. However, for larger-scale simulations or actual quantum devices, factors such as noise, error correction, and optimization techniques need to be considered to ensure accurate and reliable results.

Conclusion:

In this report, we presented a simplified implementation of a Ludo game using quantum computing with Qiskit. The use of a quantum circuit to generate random dice rolls adds an interesting dimension to the gameplay. Although the implemented version is basic, it serves as a starting point to explore the intersection of quantum computing and traditional board games. Further enhancements and optimizations can be made to improve the gameplay experience and incorporate more advanced quantum computing concepts.

By leveraging the power of quantum computing, it is possible to explore new possibilities in game design and computation, opening doors for innovative applications and research opportunities in various domains.

Github Link :

https://github.com/madhubannigol/Quantum-mini_-Project