International Islamic University Chittagong

Project Report ON

Intelligent 8-Queen Puzzle Game with Adaptive Difficulty and Rewards

Team Members

C201279 Ummay Samiun Basri

Course Code & Name

CSE-3636

Artificial Intelligence Lab

Course Teacher

Sara Karim

Assistant Lecturer, Dept of CSE, IIUC.

1. Introduction

This project implements an interactive, intelligent version of the classical 8-Queens puzzle using **Python** and the **Pygame** library. The game introduces multiple difficulty levels, obstacles, reward mechanisms, and visual effects to enhance user experience and introduce AI-based logical reasoning in game dynamics.

2. Problem Statement

The traditional 8-Queens puzzle requires placing eight queens on a chessboard such that no two queens attack each other. This project reimagines the puzzle as a level-based game, integrating obstacles, limited moves, time constraints, and rewards — simulating decision-making under constraints, a common theme in AI.

3. Objectives

- Develop a playable game that challenges players with the 8-Queens puzzle.
- Introduce AI-inspired mechanics like valid move checks and hints.
- Provide a progressive difficulty structure (8 levels).
- Implement a reward system encouraging strategic thinking.
- Use animations and visual cues to make the experience engaging.

4. Tools and Technologies

- **Language:** Python
- **Library:** Pygame
- **IDE:** Visual Studio Code
- **Platform:** Windows 64-bit

5. Game Features

a. Level-Based Design

- 8 levels of increasing difficulty.
- Time and move constraints per level.
- Each level adds more obstacles on the board.

b. AI Logic

- Valid queen placement is checked using logic that ensures no attacks (row, column, diagonal).
- The board dynamically prevents invalid moves and notifies the player.

c. Obstacles

- Randomly generated unplaceable cells that simulate real-world constraints.

d. Rewards

- After level completion, players may receive:
  - Extra moves
  - Hints
  - Undo options
  - New board themes (colors)

e. Hints and Undo

- Hints highlight valid placements (limited in number).
- Undo removes the last queen placed (limited as well).

f. Visual Effects

- Fireworks and dancing queens celebrate victories.
- Real-time updates for score, time, level, and move count.

6. Code Highlights

a. Validity Checking (AI Logic):

```
python
CopyEdit
def is_valid_placement(x, y, board_size):
    for qx, qy in queens:
        if x == qx or y == qy or (x - y) == (qx - qy) or (x + y) == (qx + qy):
            return False
    if (x, y) in obstacles:
        return False
    return True
```

b. Hint System:

```
python
CopyEdit
def show_hint(board_size):
```

```
    for x in range(board_size):
        for y in range(board_size):
            if is_valid_placement(x, y, board_size):
                pygame.draw.rect(screen, BLUE, (x * TILE_SIZE, y * TILE_SIZE,
TILE_SIZE, TILE_SIZE))
                pygame.display.flip()
                time.sleep(3)
                return
```

c. Reward Logic:

```
python
CopyEdit
def give_reward():
    reward = random.choice(["extra_moves", "hint", "undo_move", "new_theme"])
    ...
```

7. Challenges Faced

- Dynamically managing screen size and tile rendering across devices.
- Ensuring the hint system does not compromise puzzle integrity.
- Balancing difficulty and rewards to maintain user engagement.

8. Learning Outcomes

- Applied **AI concepts** in a practical game scenario.
- Gained proficiency in **Pygame** and dynamic rendering.
- Improved understanding of **constraint satisfaction problems**.
- Learned how to simulate intelligent behavior and interactivity.

9. Future Enhancements

- Integrate AI solver to suggest optimal queen placements.
- Introduce multiplayer or competitive mode.
- Save progress and allow resume functionality.
- Add sound effects and background music.

10. Conclusion

This project successfully combines **AI principles** with **game development**, transforming a classic problem into an engaging, interactive application. Through adaptive difficulty, logical reasoning, and player rewards, it reflects the essence of AI-based intelligent systems.