PROJECT REPORT

AI-POWERED TETRIS GAME USING PYTHON

TEAM MEMBERS

SUBHAN (22K-4316)

SADIQ (22K-4303)

MUHAMMAD UMER (22K-4160)

MUHAMMAD JUNAID ASIF (22K-4369)

AI-Powered Tetris Game Using Python and PyQt5

Submitted By: Subhan (22K-4316)

Course: Artificial Intelligence (AI)

Instructor: Miss Syeda Ravia Ejaz , Miss Alina Arshad

Submission Date: May 15, 2025

1. Executive Summary

● Project Overview:

This project involves the development of a classic Tetris game using Python, enhanced with a graphical interface and a built-in AI capable of playing the game autonomously. The goal was to recreate the core gameplay of Tetris and implement an AI agent that could analyze game states and make optimal decisions in real-time. The AI is rule-based and uses heuristic evaluation functions to simulate and choose the best possible move for each falling tetromino. The project showcases real-time decision-making and serves as an introduction to AI in dynamic, interactive environments.

2. Introduction

● Background:

Tetris is a widely recognized and historically significant puzzle game that involves arranging falling tetrominoes to form complete horizontal lines. The game was chosen for this project due to its balance of simplicity and complexity—it offers straightforward mechanics yet presents significant AI design challenges due to its rapidly changing environment and long-term consequences of early moves.

● Objectives of the Project:

- To develop a fully functional Tetris game in Python.
- To implement a heuristic-based AI capable of playing the game automatically.
- To allow toggling between manual and AI gameplay.
- To explore the decision-making process of AI in real-time, spatial reasoning tasks.

3. Game Description

● Original Game Rules:

In traditional Tetris:
- Blocks of various shapes fall from the top of the board.
- Players must rotate and move these blocks to complete horizontal lines.
- Completing a line clears it and awards points.
- The game ends when blocks stack up to the top of the board.

● Innovations and Modifications:

- Added a fully automated AI using rule-based decision making.
- Integrated a real-time graphical interface using PyQt5 instead of using text or console-based visualization.
- Included functionality to toggle between AI mode and manual mode.
- Adjustable speed to allow tuning for human players or fast AI runs.

4. AI Approach and Methodology

● AI Techniques Used:

The project uses a heuristic evaluation approach. Instead of classic Minimax, it evaluates all possible rotations and horizontal positions for each shape and scores them based on certain board characteristics.

● Algorithm and Heuristic Design:

The AI simulates each possible move and evaluates it based on a set of heuristics:
- Lines completed
- Holes created
- Surface bumpiness and height
- Column height deviation

The AI picks the move with the highest overall score and applies it in real-time.

● AI Performance Evaluation:

Performance was observed by enabling the AI and letting it play continuously. The AI successfully avoids early game-over states, prioritizes line clearing, and makes intelligent decisions that mimic human strategies. It consistently outperforms novice human players and maintains survival for long durations.

5. Game Mechanics and Rules

● Modified Game Rules:

- AI can automatically control tetromino movement, rotation, and dropping.
- Game speed is adjustable to accommodate AI decision time or human reflexes.
- Game starts with a random shape and shows the upcoming shape in a preview panel.

● Turn-based Mechanics:

- The game runs in real-time but follows a timer-driven loop.
- On each tick, the game processes a downward movement, checks for line completion, and draws the board.
- If AI mode is enabled, it calculates the next move before the piece drops.

● Winning Conditions:

There is no final win state—consistent with classic Tetris, the game is endless and ends when the board fills up. The goal is to achieve the highest possible score before a game-over.

6. Implementation and Development

● Development Process:

The project was implemented in modular components:
1. `tetris_model.py`: Created game rules and data structures.
2. `tetris_ai.py`: Developed the AI class and scoring mechanism.
3. `tetris_game.py`: Implemented GUI, game loop, keyboard controls, and AI integration.

● Programming Languages and Tools:

- Programming Language: Python 3
- Libraries: PyQt5 (GUI), NumPy (AI board simulations)
- Tools: Visual Studio Code, Git for version control

● Challenges Encountered:

- Designing a scoring function that balances short-term and long-term consequences.
- Integrating AI decisions smoothly within a real-time GUI without lag.
- Managing shape collisions and rotations within the confines of the grid boundaries.
- Adjusting speed for AI performance while maintaining a playable experience for humans.

7. Team Contributions

Muhammad Umer: Responsible for AI algorithm development, including heuristic design and evaluation logic.

Sadiq: Handled the core game logic, including piece rotations, line clearing, and board data management.

Junaid: Implemented the graphical user interface using PyQt5 and integrated user inputs.

Subhan: Conducted performance testing, debugging, and assisted in AI integration with the main game loop.

8. Results and Discussion

● AI Performance:

The AI was evaluated based on its ability to survive and score points efficiently:
- Average survival time: Over 10 minutes in standard mode
- Decision-making time: Less than 100ms per move (real-time compatible)
- Line clearing: Prioritizes multiple line clears and avoids risky placements
- Win condition: Game does not end with a win, but AI can play significantly longer than a typical human without making major mistakes

9. References

- Tetris Official Game Mechanics: https://tetris.com
- PyQt5 Documentation: https://doc.qt.io/qtforpython/
- NumPy Documentation: https://numpy.org/doc/
- AI Heuristic References:
  - https://codemyroad.wordpress.com/2013/04/14/tetris-ai-the-near-perfect-player/
  - https://www.cs.cornell.edu/~hp583/tetris.pdf