AI-Driven Tetris Game Using Heuristic Search Algorithm

TEAM MEMBERS:

SUBHAN

MUHAMMAD SADIQ

JUNAID ASIF

MUHAMMAD UMER

Course: Artificial Intelligence

Instructor: Miss Syeda Ravia Ejaz

Submission Date: 18/03/25

AI-Driven Tetris Game Project

1. Project Overview

Project Topic

This project focuses on developing an AI-driven Tetris game where an intelligent agent strategically analyzes the board state and evaluates optimal piece placements using heuristic-based search algorithms. The AI aims to maximize efficiency by optimizing moves, clearing lines, and minimizing gaps to enhance gameplay. **Objective**

The primary objective is to develop a high-performing AI agent capable of playing Tetris autonomously. The AI will leverage heuristic evaluation to assess potential moves and select the best one based on a scoring system. This project demonstrates the application of state evaluation functions combined with search techniques to create intelligent game-playing agents.

2. Game Description

Original Game Background

Tetris is a classic puzzle game in which players manipulate falling tetromino shapes to form complete horizontal lines on a grid. Completed lines are cleared, and the game continues until the board is full. The challenge increases as the falling speed accelerates over time.

Innovations Introduced

- **AI Integration:** An autonomous AI agent will play Tetris using heuristic-based search techniques.
- **Scoring System:** The AI evaluates board states by optimizing line clears while minimizing holes and block heights.
- **Dynamic Strategy:** The AI adapts its decision-making based on the current and next

Tetris pieces.

- **Performance Optimization:** The AI employs optimized search techniques to explore potential moves efficiently, adapting to increasing complexity.

These innovations enhance gameplay by creating a challenging AI opponent capable of adapting to dynamic environments.

---

3. AI Approach and Methodology

AI Techniques to be Used

- **Heuristic Search Algorithm:** Evaluates each potential move based on multiple weighted features.
- **Greedy Search (One-Ply Lookahead):** Examines the next game state and selects the move with the best score.
- **Feature-Based Scoring:** Uses weighted heuristics to evaluate board states. **Heuristic Design**

The AI evaluates board states based on the following weighted features:

- **Lines Cleared:** Maximizing completed lines for higher scores.
- **Hole Penalty:** Minimizing empty spaces beneath placed blocks.
- **Height Penalty:** Reducing the maximum column height to avoid losing the game.
- **Surface Roughness:** Minimizing differences between column heights for better stability.
- **Vertical Block Density:** Evaluating how many blocks are above existing holes.

Complexity Analysis

- **Time Complexity:** O(b × r), where *b* is the board width and *r* is the number of rotations explored. The greedy search reduces complexity by only looking one step ahead.
- **Challenge:** Managing real-time piece placements and optimizing decisions under time constraints.

---

4. Game Rules and Mechanics

Modified Rules

- **Autonomous AI Play:** The AI will control all Tetris piece movements automatically.
- **Real-Time Decisions:** The AI must react dynamically to falling speeds.
- **Scoring Optimization:** The AI’s goal is to maximize line clears while minimizing penalties.

Winning Conditions

- The AI plays until the board is full, at which point the score is recorded. The objective is to achieve the highest possible score.

Turn Sequence

1. Identify all valid moves for the current Tetris piece.
2. Evaluate each move using the heuristic scoring function.
3. Select the move with the highest score.
4. Execute the move and repeat the process for subsequent pieces.

---

5. Implementation Plan

Programming Language

- **Python**

Libraries and Tools

- **NumPy:** For efficient data manipulation and board state evaluation.
- **Pygame:** For game visualization and user interaction.
- **Datetime:** For performance measurement during AI decision-making.

Milestones and Timeline

- **Week 1-2:** Study Tetris game structure and design AI heuristics.
- **Week 3-4:** Implement board evaluation function and search algorithm.
- **Week 5-6:** Integrate AI decision-making into Tetris gameplay.
- **Week 7:** Optimize AI performance and test game stability.
- **Week 8:** Finalize implementation, conduct evaluation, and prepare the report.