

Tetris-AI - Triste

Framerica

MOLLY-MITTON Clément
College of Engineering
dept. Computer Science
Paris, France
clement.mollymitton@gmail.com

BENDAVID OUYOUSSEF Sarah
College of Engineering
dept. Computer Science
Paris, France
sarahbendavid@dartybox.com

HUBNER James
College of Arts and Sciences
dept. Criminal Justice
Alabama, United States
jhub@uab.edu

ARROUET LE BRIGNONEN Aubin
College of Engineering
dept. Computer Science
Paris, France
aubin.arrouet_le_brignonen@edu.devinci.fr

MOBLEY Erin
College of Engineering
dept. Computer Science
Florida, United States
erin.mobley@email.saintleo.edu

Abstract—This project proposal aims to delve deeply into the game Tetris, focusing on two essential aspects. Firstly, it involves faithfully recreating the game Tetris and conducting an in-depth analysis of its functioning, design, and history. Secondly, this project seeks to develop an Artificial Intelligence (AI) capable of mastering Tetris to the extent of never losing.

The first part of the project entails a meticulous analysis of Tetris, breaking down its mechanics, design principles, and historical evolution. This analysis will shed light on the elements that have contributed to making Tetris an enduring cultural phenomenon and explain its consistent appeal.

The second component of the project is equally fascinating, involving the development of an Artificial Intelligence (AI) system that can achieve exceptional mastery of Tetris, avoiding losses altogether. By creating an AI capable of surpassing human-like strategic thinking and adaptability within the Tetris environment, we aim to push the boundaries of AI capabilities and explore potential applications of this technology in the realms of gaming, problem-solving, and AI research.

ROLE ASSIGNMENTS

| Roles | Name | Task description |
|----------|-------|---|
| User | Erin | <ul style="list-style-type: none">- Proofreading documents: Erin, as a native English speaker, will review and edit project documents for language accuracy and clarity. She will ensure that the content is linguistically flawless.- User Interface Development: Erin will be responsible for designing and developing the user interface, ensuring a user-friendly and visually appealing experience. |
| Customer | Aubin | <ul style="list-style-type: none">- Resource Management: Aubin will oversee the allocation and utilization of project resources, ensuring efficient utilization of time and budget. |

| | | |
|---------------------|-----------------|---|
| | | <ul style="list-style-type: none">- Support: Aubin will also have the mission of helping the various members in their tasks if necessary. |
| Software Developer | Clément / James | <ul style="list-style-type: none">- Writing Code: James and Clément will be responsible for writing the code for the project, and implementing the game mechanics, AI algorithms, and system functionality.- AI Development: They will also focus on creating and fine-tuning the artificial intelligence (AI) component of the project, allowing it to excel at playing Tetris.- Documentation Assistance: In addition to coding, James and Clément will provide support in writing technical documentation, ensuring that the development process is well-documented. |
| Development Manager | Sarah | <ul style="list-style-type: none">- Organizing the Project: Sarah will take charge of project organization, setting deadlines, scheduling meetings, and ensuring that project milestones are met.- Coordination: She will facilitate communication and coordination among team members, ensuring that everyone is aligned and working effectively towards project goals.- Technical Documentation: Sarah will be responsible for crafting technical documents that detail the project's architecture, design, and development processes. |

I. INTRODUCTION

Motivation

Video games have always been a captivating realm of technological exploration and creativity, and among the most iconic titles in this industry, Tetris shines brightly. Created by Russian developer Alexey Pajitnov in 1984, Tetris has evolved into a global phenomenon, thanks to its addictively simple yet strategically rich gameplay. This puzzle game has enthralled entire generations of gamers, delivering a timeless gaming experience.

This is precisely why our interest was piqued by Tetris. While Tetris's rule set is very simple the multitude of strategies one can employ to strive for the highest score is particularly fascinating. Naturally, some strategies prove more effective than others. Given Tetris' universal recognition, we found it compelling to create an AI capable of playing the game endlessly while employing the most advanced strategies. To realize this project, we must rebuild Tetris from the ground up and design a sophisticated AI that excels in the game.

Developing an AI with the ability to indefinitely succeed at playing Tetris presents a complex problem-solving challenge. It necessitates critical thinking, the creation of innovative strategies, and the optimization of algorithms for maximum efficiency. Through this endeavor, we will not only employ and apply theoretical AI concepts but also enhance our software engineering skills.

Problem statement

The objective of Tetris is simple, Match the tiles to form a line so that it clears space in the board. Tetris was designed to be endless, However, there comes a point in the game where the player cannot continue. This happens when enough uncleared rows build up and there is no room left on the board to add the next generated game piece (tetrominoes).

With the inclusion of Artificial Intelligence, could the games be pushed to their limits, and a truly endless game of Tetris be achieved? How can we design a Tetris AI that not only places the tetrominoes efficiently but also can adjust to the gameplay mechanics of ever-increasing gravity with the randomness of block combinations, while still taking the player's inputs into account? The AI should still mimic the decision-making done by humans once it is aware of the upcoming tetrominoes, checking for the best placement, and executing its decision to allow the game to continue for as long as possible.

Research on any related software

Several AIs have been implemented in Tetris to cover all the different ways of playing the game. Indeed, Tetris has been implemented with different AIs over the years for several purposes: creating its computer opponents or creating its own recommendation system. Several videos have been posted on Youtube presenting AIs that could play Tetris, some indefinitely:

A. Code Bullet

The AI coded by Code Bullet was reported as being the first AI to beat the human world record. It managed to get a score of around 15,000 points, he also said if he continued training and improving it, the AI could possibly become invincible.

B. StackRabbit

The AI coded by Greg Cannon managed to break Tetris NES. It broke the game because when the game is played on the console, scores that are too high can cause the game to crash because of the way the score is calculated. After reaching a score of 102 million, the game crashed and so the AI broke the game and thus could be considered invincible.

C. Github

Making a Tetris-solving AI is a typical project for students, people who want to learn using AIs, and simply people who enjoy the game. A lot of people then posted their projects on Git Hub, consequently, there is a huge amount of GitHub projects for a Tetris AI.

II. REQUIREMENTS

A. Creation of TETRIS

1) *Platform:* The project will be developed using Visual Studio code.

2) *Game Engine:* In this project, we want to program everything by ourselves to have the maximum amount of control so we will develop our interface.

3) *Programming Language:* We will use Python for this project for its ease of use, multipurposeness, and the fact that it will be helpful for the AI part of the project.

4) *Tetromino Generation:* The tetromino generation will use a random generation algorithm. We would like, if possible, to use the exact random generation algorithm that the original Tetris uses to keep the maximum amount of fidelity possible.

5) *Movement and Rotation:* Development functions to handle the movement (left, right, down) and rotation of tetrominoes based on player input. Tetris also has more advanced mechanics in which when the player rotates the tetromino, this one stays in place and doesn't descend, this function should also be developed.

6) *Collision Detection:* Implement collision detection algorithms to determine when a tetromino reaches the bottom or collides with another tetromino, triggering actions like line clearing.

7) *Line Clearing:* Implement the logic to detect and clear complete lines when a tetromino fills a row, adjusting the game grid and influencing the score.

8) *Scoring System:* Design and implement a scoring system that rewards points based on the number of lines cleared simultaneously (single, double, triple, or Tetris) and the level of the game.

9) Game speed: The game speed of Tetris increases when the score increases. The game speed of Tetris increases by steps rather than linearly and is capped after reaching a certain score, the precise speeds and the scores at which they are used in the game should be the same as in the original Tetris. Some tutorials and forums provide the table in which the speeds are listed so it should be easy to get them.

10) Game Grid: Create a graphical representation of the game grid using appropriate rendering techniques, clearly displaying the current state of tetrominoes and placed blocks.

11) HUD Elements: Design and integrate HUD elements, including score display, level indicator, and upcoming tetromino preview, providing essential information to the player during gameplay. For the AI, these functions should also be inputs that are directly provided to the AI so it can play without having to use the graphical interface. This interface is simply used for the human users of the game.

12) Efficient Rendering: Optimize rendering by utilizing techniques like sprite batching, reducing draw calls, and managing textures efficiently to achieve optimal frame rates. Also, since the aesthetics of the game are not a priority, using low-resolution textures should help. Still, since Tetris is a 2D game with a low amount of elements, performance should not be an issue for the human players, and the AI won't need any graphical interface so this way we will be able to train it faster on the game.

13) Memory Management: Implement efficient memory management practices to minimize memory leaks and ensure the game runs smoothly without performance degradation during extended gameplay. This is extremely important for our project because we count on having the AI play the game for extended periods.

14) Code Optimization: Apply algorithmic and code-level optimizations to enhance efficiency, reducing processing time for critical game functions like collision detection and tetromino movement.

15) Unit Testing: Develop and conduct unit tests for critical game components, ensuring their functionality and correctness, and facilitating quick identification and resolution of bugs.

16) Integration Testing: Perform integration tests to validate the seamless integration and collaboration of various game components, including the user interface, game logic, and scoring system.

17) User Acceptance Testing: Conduct extensive playtesting with a diverse group of users to gather feedback and identify usability issues, ensuring the game meets the expectations and preferences of the target audience.

B. Creation of the AI

1) Data Collection: The AI system needs access to a comprehensive dataset of Tetris's gameplays from which it can learn. This dataset should encompass a variety of game scenarios, including different starting positions, levels, and player actions.

2) Input Processing: The AI should possess the ability to process the current game state, which includes the current Tetrimino, the matrix of filled blocks, and the current score.

3) Strategy Development: The AI system must develop a strategic approach to decision-making regarding the placement of Tetriminos. This strategy should take into account various factors such as maximizing score, clearing lines, avoiding stacking blocks, and ensuring long-term success.

4) Decision-Making Algorithm: An intelligent decision-making algorithm is crucial for the AI to determine the optimal move. This algorithm should consider various factors, including the current game state, upcoming pieces, and potential future moves, to calculate the best possible move.

5) Training Environment: The AI system requires a training environment in which it can interact with a game simulation. This environment allows the AI to observe, learn, and repeatedly play against itself or human players, thereby enhancing its performance.

6) Reinforcement Learning: The AI should utilize reinforcement learning techniques to continually improve its gameplay. Through receiving rewards or penalties based on its performance, the AI can adapt its decision-making algorithm to optimize its strategy.

7) Adaptability: The AI system must be designed to adapt to different levels of difficulty and game speeds. This adaptability ensures that the AI can effectively handle various game scenarios and continue to perform well as the game progresses.

8) Real-Time Responsiveness: The AI needs to be capable of making decisions in real time, considering the ever-changing game state. This requirement is essential to ensure that the AI can respond effectively to rapidly changing conditions and make timely moves.

9) User Interface: The AI system may include a user interface that allows users to interact with the AI, observe its decision-making process, and control certain aspects of the training process.

10) Performance Evaluation: To assess the effectiveness of the AI, performance evaluation metrics should be defined. These metrics may include the number of cleared lines, average score per game, average time played, and the ability to adapt to different game speeds.

By fulfilling these general requirements, it is possible to design and develop an AI system that can efficiently learn and play Tetris. However, it is important to note that the AI will require continuous refinement, debugging, and iteration to improve its performance and ensure its success in various game scenarios.