

Inteligência Artificial

Aplicações de IA em Tetris

1

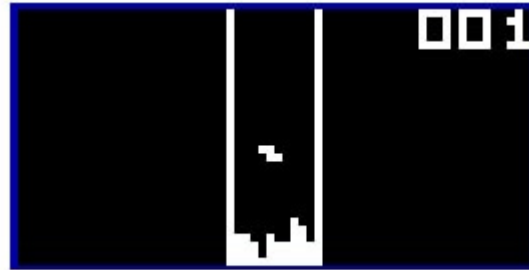
GRUPO:

FREDERICO SAMPAIO
GUILHERME SAMPAIO
GUSTAVO PANTUZA

História

2

- Em 1985, Alexey Pajitnov and Dmitry Pavlovsky programavam a primeira versão de Tetris em um Electronica 60.



Multiplicidade de implementações

3



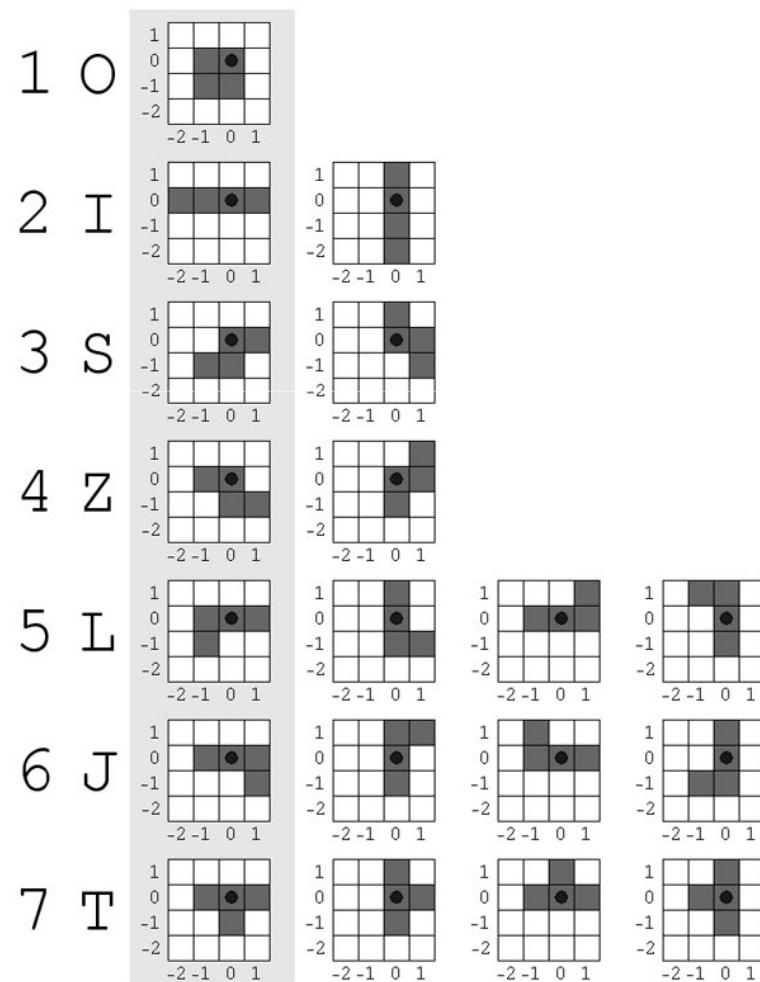
O jogo TETRIS

4

- O jogo não possui um estado de vitória.
- O objetivo é atrasar a perda (prolongar o jogo).
- É competitivo, ganha quem eliminar mais linhas.
- É dinâmico e estocástico.
- As peças são escolhidas aleatoriamente.
 - Dispersão aleatória: “uniforme” x “não uniforme”.
- Aumenta a dificuldade: aceleração da queda.
- Pode-se usar tabuleiros de diferentes dimensões .
- Pode-se mostrar (preview) as “n” próximas peças.

As peças do jogo original

5



Exemplos de aplicações

6

- Diversão.
- Campeonatos, inclusive profissionais.
- Estudo da percepção e reação humana.
- Estudos de Inteligência Artificial:
 - Agentes inteligentes.
 - ✦ Determinísticos.
 - ✦ Com aprendizagem.
 - Visão computacional.

Problema NP-Difícil

7

- “We prove that in the offline version of Tetris, it is NP-complete to maximize the number of cleared rows, maximize the number of tetrises (quadruples of rows simultaneously filled and cleared), minimize the maximum height of an occupied square, or maximize the number of pieces placed before the game ends.”

DEMAINE, Erik D.; HOHENBERGER, Susan;
LIBEN-NOWELL, David: **Tetris is Hard, Even to Approximate.** February 1, 2008.

Agentes inteligentes sem aprendizagem

8

- Algoritmo de **Pierre Dellacherie** (2003; France)
 - Baseado em uma probabilidade constante (p) do jogo terminar nos próximos (k) movimentos.
 - Considerado uma boa abordagem para o problema.
 - Citado em vários artigos.
 - Usado como base de treinamento de agentes com aprendizagem.
 - “The best one-piece, real-time Tetris algorithm to my knowledge was created by Pierre Dellacherie of France. His amazing algorithm sometimes fills more than 2 million rows! The average performance is on the order of 650000 rows. The algorithm takes very little memory, and runs at high speed (about 160 rows per second on my 800 MHz computer).”
(http://www.colinfahey.com/tetris/tetris_en.html)

Agentes com aprendizagem

9

- Agente com aprendizagem por imitação.
 - ZHANG, Dapeng; CAI, Zhongjie; NEBEL, Bernhard. **Playing Tetris Using Learning by Imitation**. University of Freiburg, 2009.
- Agente com aprendizagem por reforço.
 - CARR, Donald. **Applying reinforcement learning to Tetris**. Rhodes University. May 30, 2005.
- Agente de busca harmônica.
 - ROMERO; Victor II M.; TOMES, Leonel L.; YUSIONG, John Paul T. **Tetris Agent Optimization Using Harmony Search Algorithm**. University of the Philippines. 2010.

Agentes com aprendizagem

10

- **Agentes de Redes Neurais?**
 - Existem vários artigos na internet sobre inteligência artificial aplicada ao Tetris, mas poucos citam técnicas de Redes Neurais (RN), apesar de algumas técnicas encontradas serem claramente baseadas em RN.
 - “Applied to a Tetris-like game, the neural network performed very badly.” (<http://www.theroundpixel.com/Turner/NeuralNetwork/turnerneuralnetwork.html>)
- **Agentes genéticos.**
 - SHAHAR, Elad; WEST, Ross. **Evolutionary AI for Tetris.** University of Massachusetts, 2010.
 - A maioria das técnicas encontradas e aplicadas ao jogo de Tetris parecem ser baseadas em algoritmos genéticos.

Algoritmos genéticos para Tetris

11

- Os genes são fatores associados ao jogo (i).
- Função de avaliação, $f(i)$, avalia cada fator do jogo.
- O cromossomo é um vetor de pesos ($w[i]$) relacionados com os fatores (genes).
- A função de fitness é:
$$\sum_{i=\min}^{\max} w[i] * f(i)$$
- Um indivíduo é um jogo do início ao fim.
- As populações são os resultados de vários jogos.
- Seleção natural por gerações de jogos.

Seleção dos indivíduos

12

- Seleção dos mais adaptados para a próxima geração, baseada nos maiores valores da função de fitness dos indivíduos da atual geração.
- Mutação, por mudança aleatória de um gene de um cromossomo, de um dos indivíduos selecionados para a próxima geração.
- Cruzamento (crossover, ou combinação de genes) entre indivíduos selecionados para próxima geração; reprodução ou geração de novos indivíduos.

Exemplos de genes para o Tetris [SHAHAR]

13

- **Filled Spot Count** - The number of filled spots on the game board.
- **Weighted Filled Spot Count** - Similar to the above, but spots in row i counts i times as much as blocks in row 1 (the bottom).
- **Maximum Altitude** - The height of the tallest column on the game board.
- **Hole Count** - The number of unfilled spots that have at least one filled spot above them.
- **Lines Cleared** - The number of full horizontal rows on the game board.
- **Altitude Delta** - The difference in height between the tallest column and the shortest column.
- **Deepest Hole** - The depth of the deepest hole (a width-1 hole with filled spots on both sides).

Ref: [SHAHAR]

Exemplos de genes para o jogo Tetris

14

- **Sum of all Holes** - The total depth of all the holes on the game board.
- **Horizontal Roughness** - The number of times a spot alternates between an empty and a filled status, going by rows.
- **Vertical Roughness** - The number of times a spot alternates between an empty and a filled status, going by columns.
- **Well Count** - The number of holes that are 3 or more blocks deep.
- **Weighted Holes** - Same as the hole count, but rows are weighted by the row they're in.
- **Highest Hole** - The height of the highest hole on the board.
- **Game Status** - Based on the game status, 1 for a losing state, 0 otherwise.

Ref: [SHAHAR]

Arquitetura do software

15



Ferramentas

16

- Existem diversas implementações do jogo Tetris.
 - Mais provável em JAVA ou em Python.
 - C# e C++ como referência do algoritmo de Pierre Dellacherie.
- Existem diversas ferramentas (bibliotecas, pacotes de classes, frameworks, etc) para algoritmos genéticos.
 - JGAP (Java Genetic Algorithms Package).
 - aima-python: Python implementation of algorithms from Russell and Norvig's Artificial Intelligence: A Modern Approach.

Dados

17

- Os dados para a aprendizagem serão obtidos ao longo dos próprios jogos.
- Caso seja necessário aprendizagem por “ensinamento”, será usado o algoritmo de Pierre Dellacherie para sugerir movimentos.
- Serão coletados dados dos jogos para reproduzir e comparar resultados entre agentes.
- Os dados genéticos dos agentes serão guardados para análise e futuras evoluções.

Proposta e critérios de avaliação

18

- Avaliar a adequação do uso de algoritmos genéticos para agentes inteligentes capazes de jogar Tetris.
- Verificar e demonstrar a evolução dos indivíduos.
- Coletar dados genéticos (cromossomas) dos indivíduos e separá-los em nível de “expertise”.
- Comparar a performance dos agentes com o algoritmo de Pierre Dellacherie.
- Identificar fatores genéticos viáveis.
- Analisar a relevância dos fatores genéticos.
- Avaliar o uso de agentes com “preview” das peças.

Riscos do software

19

- A integração entre o módulo do agente e do jogo pode ser difícil e/ou complexa.
- Curva de aprendizagem, dificuldade ou características inesperadas durante o uso das ferramentas.
- O uso de “preview” pode alterar significativamente os algoritmos inteligentes.

Riscos do algoritmo genético

20

- A seleção de indivíduos por algoritmos genéticos pode demorar muito, inviabilizando a coleta de resultados.
- Fatores genéticos irrelevantes podem comprometer o resultado da aprendizagem.
- Mudança da função de avaliação podem demandar de executar novamente todo o processo de aprendizagem em gerações.
- Pode não ser possível avaliar as relevâncias dos fatores genéticos por falta de informações estatísticas.