



Inteligência Artificial

Universidade de Aveiro

Diogo Aguiar 81020, Rafael Pereira 98354

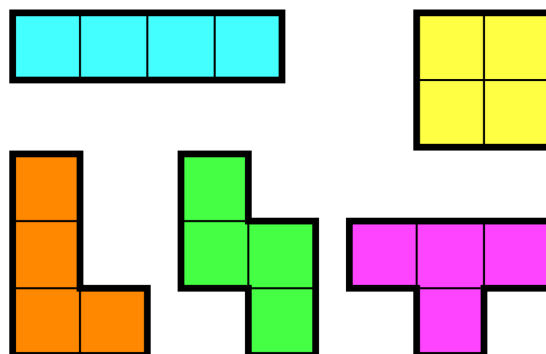
Conteúdo

1	Introdução	1
2	Agente	2
2.1	Criação do agente	2
2.2	Descrição do algoritmo	2
2.2.1	Game field	2
2.2.2	Final piece	2
2.2.3	Heuristic	2
2.2.4	Choice	3
2.2.5	Run AI	3
3	Resultados e conclusões	4

Capítulo 1

Introdução

Como intuito da avaliação em grupo na cadeira de Inteligência Artificial foi-nos proposto a criação de um agente capaz jogar de forma inteligente e autónoma o jogo Tetris. O jogo consiste em empilhar as peças (tetrominos) que vão sendo introduzidas no jogo de forma a completarem linhas horizontais que se desintegram quando se encontram "cheias" e consequentemente o jogo remove a linha e é atribuído 1 *ponto* ao jogador. Quantas mais linhas completas de uma só vez, maior a quantidade de pontos atribuída ao jogador. Conforme o tempo de jogo vai aumentando, também vai aumentando a velocidade e dificuldade do jogador em colocar as peças. Quando a pilha das peças chega ao topo da tela de jogo, o jogo dá-se por terminado.



Capítulo 2

Agente

2.1 Criação do agente

Para a criação do agente baseamo-nos em criar funções para determinarem o campo de jogo, a altura, a quantidade de buracos, quantidade de linhas completas e instabilidade das colunas. Para a escolha da melhor maneira de colocar uma peça criamos uma função que determina-se um score que ajuda-se assim o agente a escolher a sua melhor jogada.

2.2 Descrição do algoritmo

2.2.1 Game field

Começamos por desenvolver uma função que cria uma matriz (de tamanho 9x29) que demonstra aquilo que se esta a passar no jogo. A matriz baseia-se em 1's e 0's onde os 1's correspondem a um fragmento de uma peça e 0's correspondem a um espaço vazio.

2.2.2 Final piece

Para sabermos todas as posições e rotações possíveis de cada peça começamos por coloca-la encostada à esquerda e fomos "descendo" a peça ate encontrar uma que já estivesse no jogo. Depois de encontrar uma peça, incrementamos o valor do x para andar com a peça para a direita. Posto isto rodamos a peça e fizemos o mesmo e no fim guardamos todos os fields possíveis para as nossas peças.

2.2.3 Heuristic

Esta função cria um score para cada movimento possível de uma peça e seleciona o movimento com melhor pontuação para a sua próxima jogada. O score para cada movimento é escolhido com base na escolha do melhor mapa criado

para cada peça e consequente rotação. A escolha do score é feita com base em 4 heurísticas: *height – field*, *bumpiness*, *holescomplete* e *lines* e para cada uma delas o agente irá tentar maximizar ou minimizar o seu valor.

Height-field

Um dos objetivos no tetris é manter o jogo a uma altura "baixa". Para isso criamos a função *height – field* que retorna a soma das alturas de cada coluna. O nosso objetivo para o score é minimizar o valor da altura portanto atribuímos-lhe um valor negativo.

Bumpiness-field

O objetivo desta função é tentar meter as colunas na mesma altura, de forma a não criar buracos entre colunas para que então o agente seja capaz de concluir melhor os objetivos. A função consiste em somar a subtração em modulo da altura das colunas adjacentes. Para o score o objetivo é minimizar as diferenças de altura, ou seja, minimizar o nosso bumpiness atribuindo-lhe valor negativo.

Holes-field

Outro dos grandes objetivos do tetris é conseguir colocar uma peça de maneira a que não se criem buracos no jogo. Para isso criamos a função *holes – field* que retorna o numero de buracos no jogo. Consequentemente, o objetivo do agente é minimizar a quantidade dos buracos portanto atribuímos-lhe também um valor negativo.

Complete-lines

Provavelmente o objetivo mais importante do jogo, completar linhas. Criamos para isso a função *complete – lines* que nos retorna o numero de linhas completas em jogo. Quantas mais linhas completas, maior a pontuação que o agente irá fazer e assim, consequentemente, atribuímos um valor positivo para que o mesmo tente criar o maior numero de linhas completas possível.

2.2.4 Choice

Para a escolha da melhor maneira de colocar a peça, criamos uma função que com base na função *Heuristic* cria um dicionário *best – move* onde coloca o melhor score, a melhor rotação e qual o game-field correspondente. A peça (e consequente rotação) que estamos a usar. No final a função retorna o dicionário.

2.2.5 Run AI

É nesta função que atribuímos ao agente qual o movimento que irá fazer com base na função *choice*.

Capítulo 3

Resultados e conclusões

Com esta implementação o nosso agente é capaz de jogar até uma certa velocidade, mas sabemos que é passível de melhoramento. A tabela seguinte indica a media da pontuação do nosso agente em 10 jogos realizados. Uma das possíveis implementações para melhorar os resultados obtidos pelo nosso agente, seria calcular a heurística usando a peça em jogo e as próximas três next pieces, para que assim o agente conseguisse colocar as peças no melhor sítio possível obtendo a melhor pontuação.

Jogos	Pontuação
Jogo 1	363
Jogo 2	115
Jogo 3	69
Jogo 4	100
Jogo 5	153
Jogo 6	173
Jogo 7	551
Jogo 8	199
Jogo 9	513
Jogo 10	220

Media de pontuação em 10 jogos
245.6