Resolução do jogo Break the Ice utilizando métodos de pesquisa em linguagem Java (Tema 2 Grupo 5)

1st Eduardo Silva (up201603135)
MIEIC
FEUP
Porto, Portugal
up201603135@fe.up.pt

2nd Mariana Costa (up201604414)
MIEIC
FEUP
Porto, Portugal
up201604414@fe.up.pt

3rd Tiago Castro (up201606186)
MIEIC
FEUP
Porto, Portugal
up201606186@fe.up.pt

***Resumo*—O presente artigo tem como principal objetivo delinear os contornos de desenvolvimento da resolução do jogo Break the Ice utilizando métodos de pesquisa em linguagem Java. Será realizada um descrição sucinta do jogo em questão, seguida da formulação do mesmo como problema de pesquisa bem como a exposição de outros trabalhos semelhantes considerados úteis e uma breve conclusão a destacar as perspetivas de desenvolvimento.**

***Index Terms*—Inteligência Artificial, Pesquisa, Jogo, Break the Ice**

I. Introdução

O trabalho aqui exposto insere-se no âmbito da unidade curricular de Inteligência Artificial do MIEIC. É proposta a resolução do Break the Ice, um jogo do tipo solitário, utilizando métodos de pesquisa adequados em linguagem Java. Este deverá disponibilizar um modo de visualização textual ou gráfica, de forma a mostrar com clareza a evolução do tabuleiro e possibilitar a comunicação entre a máquina e o utilizador/jogador. Deverá igualmente permitir um modo de jogo no qual o computador resolve o solitário autonomamente, com recurso a um método e configuração selecionados previamente pelo utilizador.

II. Descrição do problema

Break The Ice: Snow World [1] é um jogo de azulejos no qual o objetivo principal consiste em mover as peças em jogo (azulejos) para um espaço vazio ou trocá-las com peças adjacentes de modo a criar uma cadeia vertical ou horizontal de três ou mais peças. De cada vez que estas cadeias são criadas, as peças a elas pertencentes são quebradas, consequentemente desaparecendo do tabuleiro de jogo. O nível é ganho quando todas as peças deste são quebradas, ou seja, quando não restar nenhuma peça. No entanto, é de notar que há um número limitado de jogadas (movimentos) por nível, sendo que o jogador perde se ultrapassar este limite.

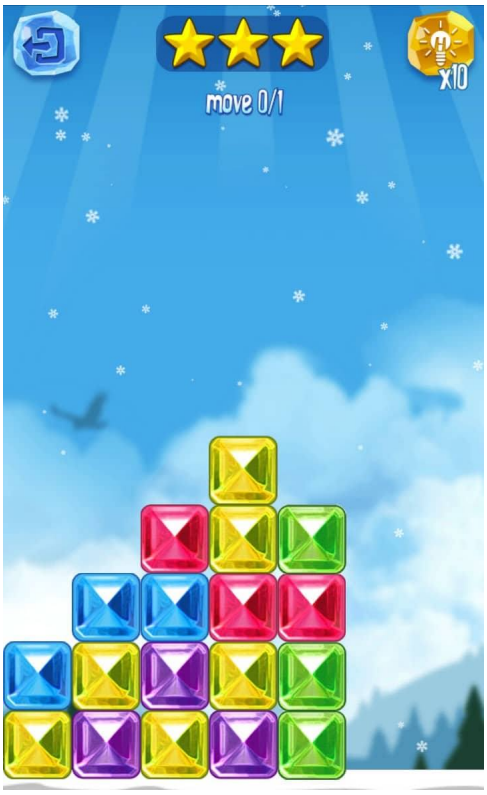

Figura 1. Ecrã inicial de um puzzle

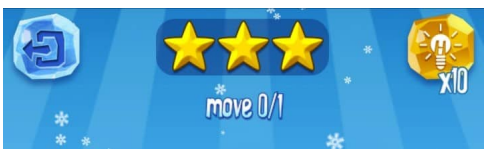

Figura 2. Sistema de custo por jogada

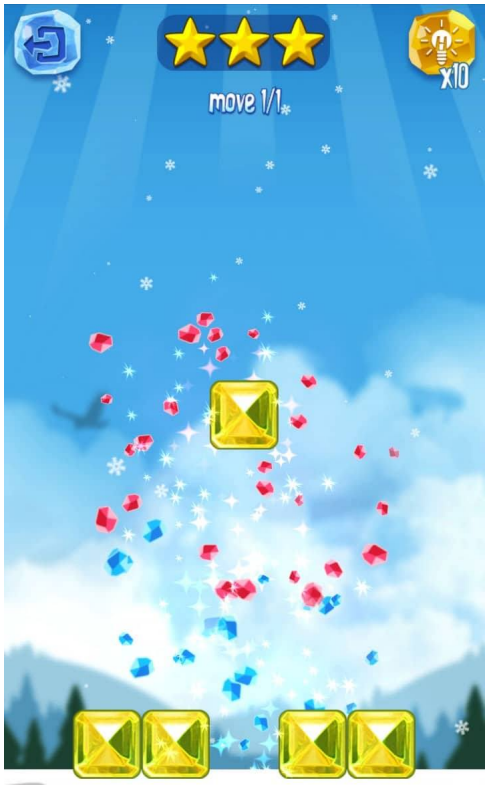

Figura 3. Movimento das peças

III. Formulação do problema

A. Representação do estado

Estados representados por uma matriz 2D de carateres em que cada elemento representa um espaço vazio ou um elemento (peça). A peças de diferentes cores irão corresponder carateres diferentes, assim como aos espaços vazios. As peças podem ser roxas, laranjas, rosas, azuis, verdes e amarelas.

B. Estados iniciais

Peças distribuídas pela matriz, sendo que não há nenhum conjunto de 3 ou mais peças da mesma cor alinhadas ortogonalmente.

C. Teste objetivo

Para um nível ser considerado completo, todas as peças têm que ter sido eliminadas, ou seja, a matriz contém apenas espaços vazios.

D. Operadores

- *Mover peça (esquerda, direita)*
  - **Pré-condições**: Existir um peça na posição selecionada e a posição de destino não corresponder a uma posição fora da matriz de jogo.
  - **Efeitos**: A peça move-se uma posição para a esquerda ou direita. Caso a peça esteja inicialmente numa posição superior à última linha da matriz e na posição para a qual se tenha movido não haja outro elemento na posição imediatamente inferior, a peça percorre as linhas da matriz por ordem crescente de índice até encontrar outro elemento ou até atingir a última linha na matriz.
  - **Custo**: 1 jogada
- *Trocar peças (esquerda, direita, cima, baixo)*
  - **Pré-condições**: As peças a trocar devem estar em posições adjacentes ortogonais.
  - **Efeitos**: As peças trocam de posição.
  - **Custo**: 1 jogada

E. Efeitos gerais

Se após uma destas operações houver pelo menos 3 peças da mesma cor em linha (ortogonalmente), todas as peças dessa cor que estejam em contacto com eles irão desaparecer (para além do conjunto inicial).

F. Custo da solução

Cada movimento tem um custo de 1 jogada, sendo o custo total da solução o número de movimentos realizados para resolver o problema.

IV. Trabalho relacionado

Durante a pesquisa para este trabalho, foi encontrado um projeto [2] com um objetivo semelhante num contexto bastante parecido ao presente. Este projeto retrata um jogador AI para uma versão clone do popular jogo Candy Crush. Este jogo é, em si, bastante parecido ao puzzle central deste trabalho. Além disso, esse projeto foi realizado na mesma linguagem que o presente trabalho (Java), portanto poderá haver módulos úteis em que nos possamos inspirar para a conclusão deste.
Foi encontrado também um relatório [3] que analisa multiplos métodos de busca para resolução de puzzles do jogo Candy Crush sendo alguns deles a pesquisa em largura e a pesquisa gulosa.

V. Conclusões e perspetivas de desenvolvimento

Após análise do trabalho a desenvolver concluímos que o método de estruturação do projeto previamente apresentado é o melhor para obter um bom resultado no mesmo. Assim, implementaremos, em Java, os operadores designados procedendo seguidamente a testar os vários métodos de pesquisa de forma a otimizar a obtenção de uma resolução do puzzle. Até à data de conclusão do presente relatório, encontram-se implementados todos os operadores e a estrutura de representação visual do estado.

Referências

[1] https://play.google.com/store/apps/details?id=com.bitmango.breaktheicehl=en
[2] https://github.com/dkarageo/ThmmyCrushCompetition
[3] http://www.cs.huji.ac.il/ ai/projects/2014/crushingCandyCrush/report.pdf