Break the Ice – Snow World (Grupo 34)

Francisco Ferreira (201605660)

Francisco Friande (201508213)

João Pedro Fidalgo (201605237)

Pretende-se neste trabalho implementar um jogo do tipo solitário para um jogador e resolver diferentes versões/quadros desse jogo, utilizando métodos de pesquisa adequados. Os métodos aplicados devem ser comparados a diversos níveis com ênfase para a qualidade da solução obtida, número de operações executadas e tempo despendido para obter a solução

Keywords— pesquisa em largura, pesquisa em profundidade, aprofundamento progressivo, pesquisa de custo uniforme, pesquisa gulosa e algoritmo A\*.

# Introdução

Este trabalho foi desenvolvido no âmbito da unidade curricular de Inteligência Artificial, do 3º ano do Mestrado Integrado em Engenharia Informática e Computação, cujo objetivo é aprofundar os conhecimentos previamente adquiridos tanto nas aulas teóricas, juntamente com a abordagem de problemas práticos, sendo que o grupo optou por utilizar a linguagem C++. Sendo assim, o grupo propôs-se a implementar o jogo de tabuleiro Break the Ice com os seguintes modos: resolução do computador e resolução do jogador, sendo que este pode pedir dicas sobre qual a melhor jogada possível.

O relatório está dividido nas seguintes secções: Introdução; Descrição do Problema; Formulação do Problema; Trabalho Relacionado; Implementação do Jogo; Algoritmos de Pesquisa; Experiências e Resultados; Conclusões e Perspetivas de Desenvolvimento; Referências Bibliográficas.

# Descrição do Problema

O jogo Break the Ice caracteriza-se por ser um jogo de tabuleiro do tipo solitário cujo objetivo é fazer com que todas as peças em jogo sejam eliminadas, resultando numa matriz vazia. Para que as peças possam ser eliminadas é necessário que estas formem uma linha(horizontal ou vertical) com um número de peças igual ou superiores a três com a restrição de que todas as peças dessa linha têm obrigatoriamente de possuir a mesma cor.

Para que uma peça seja movimentada tem que haver uma peça adjacente com diferente cor e a movimentação consiste na troca das suas posições.

Esta troca de posições tem repercussões, nomeadamente a formação de um grupo de três ou mais peças da mesma cor, eliminando por isso o grupo referido. Uma peça não pode estar em cima de um bloco vazio, ou seja, é necessário que esteja diretamente em contacto com o nível do solo (ultima linha da matriz) ou então que esteja a ser suportada por uma peça abaixo de si. Assim sendo, se uma peça que esteja suportar outra for eliminada, a segunda vai descer no tabuleiro até encontrar uma outra peça ou a base do tabuleiro.

# Formulação do Problema

1. Representação do estado

O estado do jogo vai ser representado por uma matriz de duas dimensões em que cada elemento representa um bloco de jogo, este pode estar vazio ou com uma peça/cubo. Cada peça pode ter 6 cores possíveis (roxo, laranja, rosa, azul, verde, amarelo).

1. Estado Inicial

As peças estarão distribuídas pelo fundo da matriz e, inicialmente, não há nenhum conjunto de 3 cubos alinhados, nem com espaços por baixo, visto que estes caem.

1. Teste objetivo

Não haver nenhuma peça em jogo, ou seja a matriz estar com todos os elementos vazios.

1. Operadores

* Mover Peças/Cubos horizontalmente:

Pré-condições - O elemento selecionado ser uma peça/cubo e este não pode ser movido contra os limites da matriz.,

Efeitos - A peça movimenta-se horizontalmente, conforme o selecionado, para a “casa” ao lado e caso a nova posição esteja sem peças por baixo, esta irá cair até encontrar o fundo ou outra peça.

Custo – 1

.

* Trocar Peças/Cubos horizontal e verticalmente:

Pré-condições - As peças têm de estar em posições adjacentes (horizontal ou verticalmente).

Efeitos - As peças trocam de posição uma com a outra.

Custo – 1

* Efeitos gerais:

Após a movimentação das peças, se houver 3 ou mais peças com a mesma cor em linha, na vertical ou horizontal, essas irão desaparecer e todas as que estão nas posições superiores a elas “cair”.

* Custo da solução:

Cada movimento custa 1, sendo o custo da solução o número de movimentos para resolver o puzzle.

# Trabalho Relacionado

O trabalho relacionado diz respeito à implementação do jogo/puzzle e aos métodos de pesquisa envolvidos na sua resolução. As hiperligações incluídas nesta secção incluem no seu conteúdo código fonte referente ao respetivo mecanismo a ser usado em C++., relativamente à criação do tabuleiro e jogabilidade[1]; e Métodos de Pesquisa usados[2][3][4].

# Implementação do Jogo

O jogo foi implementado em C++, sendo o display dos estados do tabuleiro feita através do terminal, tal como toda a jogabilidade e selecções de menu, através de impressões e leituras. TODO

Descrevendo o projeto e implementação, na linguagem selecionada, do jogo incluindo a forma de representação do estado do tabuleiro, operadores (verificação do cumprimento das regras do jogo) aplicáveis com determinadas pré-condições e que têm efeitos sobre o estado do jogo e um dado custo, teste objetivo (determinação do final do jogo). Entre outras devem ser implementadas funções: ler nível de ficheiro (lendo um dado nível/estado de um ficheiro de texto), visualizar em modo de texto/gráfico um dado estado, validar uma dada jogada/operador (tendo em conta as suas pré-condições), executar uma dada jogada/operador, num dado tabuleiro, tendo em conta os seus efeitos e gerando o respetivo estado sucessor, listar todas as jogadas/operadores disponíveis num dado tabuleiro, avaliar um dado estado (tendo em conta a sua “proximidade” à solução final), testar se um dado estado é solução (teste objetivo). Os métodos de pesquisa para cálculo das jogadas a realizar que permitam ao computador jogar sozinho e resolver os puzzles devem ser descritos na secção seguinte assim como o método geral para os chamar e resolver o puzzle (utilizando um dado método selecionado de entre os disponíveis).

# Algoritmos de Pesquisa

Descrevendo os vários algoritmos de pesquisa utilizados e a sua implementação de modo a calcular a próxima jogada do PC ou retornar a solução final (conjunto de operações para transformar o estado inicial no estado objetivo). Devem ser implementados algoritmos para cálculo da solução utilizando pesquisa em largura, pesquisa em profundidade (se aplicável), aprofundamento progressivo, custo uniforme (se aplicável), pesquisa gulosa e Algoritmo A\* (estes último método utilizando várias heurísticas).

# Experiências e Resultados

Descrevendo as experiências realizadas com os vários algoritmos para resolver diversos puzzles e os resultados obtidos a nível de tempo e custo da solução obtida em cada nível, por cada um dos métodos experimentados. Devem ser incluídas tabelas comparativas dos resultados obtidos na aplicação dos vários métodos aos vários puzzles (níveis do jogo) e discutidos os resultados.

# Conclusões e Perspetivas de Desenvolvimento

Sumário do trabalho realizado e conclusões que retira deste projeto. Análise crítica dos resultados obtidos em comparação com os resultados teóricos que seriam esperados. Trabalho futuro, ou seja, formas de melhorar o trabalho desenvolvido.

##### Referências Bibliográficas

1. Stuart Russel and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edtition, Pearson Education Inc., 2010, ISBN: 978-0-13-604259-4.
2. Stuart Russel and Peter Norvig, “AimaCode - Code for the Book Artificial Intelligence: A Modern Approach", 2019, [online], available at: <https://github.com/aimacode> , consulted on March 2019.

Livros, artigos e páginas Web utilizados para desenvolver o trabalho. Todos os elementos bibliográficos devem ser citados no texto do trabalho, incluindo qualquer código fonte adaptado de uma dada fonte para a realização do trabalho.