U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO	Mestrado Integrado em Eng. Informática e Computação **Inteligência Artificial**	2018/2019 (3º Ano) 2º Sem
DOCENTES: Luís Paulo Reis, Ana Paula Rocha		

Tema do Trabalho Nº1

Métodos de Pesquisa para Resolução de Problemas utilizando Heurísticas/Conhecimento (Solitário/Puzzle para um Jogador)

Resumo

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, como sejam pesquisa em largura, pesquisa em profundidade, aprofundamento progressivo, pesquisa de custo uniforme, pesquisa gulosa e algoritmo A*, com configurações diversas (diferentes modos de representação do problema e diferentes heurísticas). 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

Descrição

Um jogo do tipo solitário caracteriza-se pelo tipo de tabuleiro e de peças, pelas regras de movimentação das peças (operadores/jogadas possíveis) e pelas condições de terminação do jogo com derrota (impossibilidade de resolver, número máximo de movimentos foi atingido, tempo limite foi atingido) ou vitória (solitário resolvido) e com a respetiva pontuação. Tipicamente, no caso de vitória é atribuída uma pontuação dependendo do número de movimentos, recursos gastos, bónus recolhidos e/ou tempo despendido.

Pretende-se desenvolver uma aplicação para jogar um destes jogos. A aplicação deve ter uma visualização em modo de texto ou gráfico (embora isto não seja o essencial do trabalho) para mostrar a evolução do tabuleiro e realizar a comunicação com o utilizador/jogador. Deve permitir um modo de jogo em que o PC resolve o solitário sozinho utilizando o método e sua configuração selecionado pelo utilizador. Poderá também permitir, um modo de jogo Humano em que o utilizador pode resolver o jogo, eventualmente pedindo "dicas" ao PC. O modo humano não é requerido, mas o modo PC e a implementação dos algoritmos de resolução de problemas respetivos são o essencial do trabalho.

Linguagem de Programação

Pode ser utilizada qualquer linguagem de programação e sistema de desenvolvimento, incluindo, a nível de linguagens, entre outras: C++, Java, C#, Python, Prolog ou LISP. A escolha da linguagem e do ambiente de desenvolvimento a utilizar é da inteira responsabilidade dos estudantes.

Constituição dos Grupos

Os grupos devem ser compostos por 2 ou 3 estudantes. Não se aceitam grupos individuais ou compostos por 4 estudantes. Os grupos podem ser compostos por estudantes de turmas diferentes (no máximo de duas turmas diferentes) mas todos os estudantes têm de estar

presentes nas sessões de acompanhamento e apresentação/demonstração do trabalho. De notar que embora a constituição de grupos compostos por estudantes de diferentes turmas seja permitida, não é aconselhada, dadas as dificuldades logísticas de realização de trabalho que pode provocar aos estudantes.

Entrega Final

Cada grupo deve submeter no Moodle dois ficheiros: um artigo em formato IEEE (relatório do trabalho) em formato PDF seguindo o modelo a disponibilizar no Moodle da disciplina, e o código implementado, devidamente comentado, incluindo um ficheiro "readme.txt" com instruções sobre como o compilar, executar e utilizar. Os estudantes devem também fazer uma demonstração da aplicação (cerca10 min) do trabalho, na respetiva aula prática, ou em outro período a designar pelos docentes da disciplina.

Estrutura do Artigo/Relatório

Sugere-se que o artigo/relatório contenha as seguintes partes:

Título: Título apropriado ao trabalho realizado.

Cabeçalho: Incluindo os elementos de identificação do trabalho, do grupo e dos estudantes.

Resumo: Descrição sumária do trabalho (dois a três parágrafos).

1. Introdução: Descrição do enquadramento, objetivos, motivação e estrutura do trabalho.

2. Descrição do Problema: Descrevendo sucintamente o puzzle/solitário, a sua história (caso esteja disponível) e as suas regras utilizando imagens apropriadas construídas/adaptadas pelo grupo e texto escrito pelos elementos do grupo. Esta secção deve ser independente da resolução do solitário e facilmente compreensível por alguém sem conhecimentos de IA ou de programação.

3. Formulação do Problema: Descrevendo a formulação do problema como um problema de pesquisa, ou seja: Representação do estado, Estado(s) Inicial(is), Teste(s) Objetivo, Operadores e respetivos Nomes, Pré-condições, Efeitos e Custo(s)). Esta formulação deve ser independente da linguagem de programação utilizada para a sua implementação.

4. Implementação do Jogo: 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).

5. 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 **c**álculo 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).

6. 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.

7. Conclusões e Perspetivas de Desenvolvimento: Sumário do trabalho 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: 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.

Anexos: Outros elementos úteis que não sejam essenciais ao artigo/relatório.

Problemas (Puzzle/Solitários) Sugeridos

Os jogos a implementar são jogos de tabuleiro (puzzles/solitários) para um jogador em que não exista a influência do fator sorte e em que todas as peças/informação esteja visível, sempre, no ecrã. Deve ser criado um ficheiro com puzzles de diferentes níveis que é lido pelo programa. Os puzzles devem poder ser resolvidos através de uma sequência de passos. Os jogos sugeridos e que possuem as características desejadas, são:

1. Klotski (https://play.google.com/store/apps/details?id=com.alcamasoft.juegos.klotski.android)
2. Break the Ice – Snow World (https://play.google.com/store/apps/details?id=com.bitmango.breaktheice)
3. Unblock Me - Free (https://play.google.com/store/apps/details?id=com.kiragames.unblockmefree)
4. Cohesion - Free (https://play.google.com/store/apps/details?id=com.NeatWits.CohesionFree)
5. Roll the Block (https://play.google.com/store/apps/details?id=com.bitmango.go.bloxorzpuzzle)
6. Birdy Jumper – (https://play.google.com/store/apps/details?id=softkos.birdyjumper&hl=en)
7. Labyrinth Robots – (https://play.google.com/store/apps/details?id=com.FridellGames.LabyrinthRobotsGame)

Outros jogos podem ser sugeridos, pelos estudantes, mas devem ser discutidos, com os docentes, nas aulas práticas, de modo a confirmar que possuem todas as caraterísticas adequadas.