Relatório Trabalho Prático LI2

Grupo 004

Francisco Rodrigues, José Luís Moura e André e Nunes

3 de Junho de 2018

Universidade do Minho

Índice

Descrição do problema

Esta tarefa, desenvolvida no âmbito da disciplina de Laboratórios de Informática 2, tem como objetivo o desenvolvimento de um programa "gerar" que recebe os seguintes argumentos: "gerar <dificuldade> <nº de linhas> <nº de colunas>". E imprime o tabuleiro gerado segundo a estrutura dos ficheiros.

A dificuldade deve ser 1 para fácil e 2 para difícil. E o tabuleiro impresso deve ter uma única solução e a dificuldade esperada.

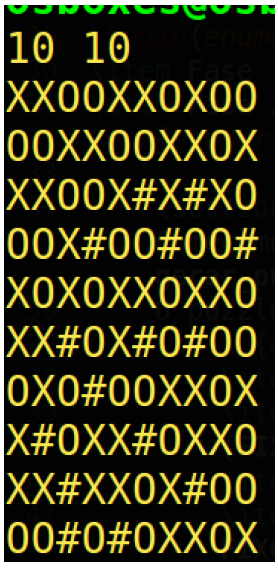
A nossa solução

Começamos por gerar um puzzle completo e válido de forma aleatória. Para isso usamos a função **gerar_puzzle_completo** que usa o input do user (nº de linhas e de colunas) para gerar um puzzle completo.

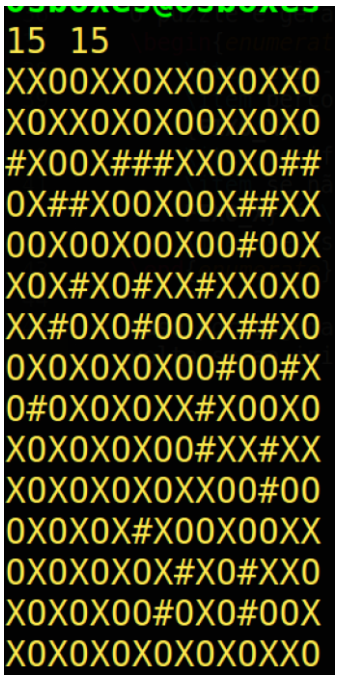
O puzzle é gerado da seguinte forma:

- cria-se o estado.
- percorre-se o puzzle preenchendo as casas vazias de forma aleatória com um **FIXO_O** e **FIXO_X**
- verifica-se se a peça é válida
- se não for põe-se a peça oposta (por exemplo: a peça contrária de **FIXO_O** é **FIXO_X**)
- se esta também não for válida preenche-se a casa com uma peça do tipo BLOQUEADA

Caso por alguma razão o nº de casas bloqueadas seja superior ao nº de linhas somado com o nº de colunas volta-se ao inicio do puzzle e o algoritmo recomeça desde o 2º ponto.



Puzzle completo 1



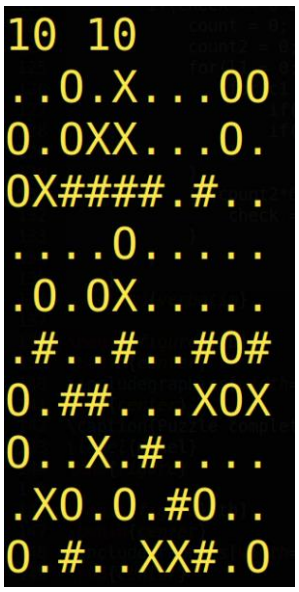
Puzzle Completo 2

Depois de ter um puzzle completo nós retiramos peças a esse puzzle enquanto este só tiver uma solução.

Caso o nível de dificuldade do puzzle pedido pelo utilizador seja 1 só se retiram peças enquanto o puzzle poça ser resolvido pelas ajudas, ou seja, enquanto seja possível resolver o puzzle recorrendo a padrões de duas peças iguais consecutivas ou intercaladas.

Caso o nível de dificuldade pedido seja 2 o programa continua a retirar peças ao puzzle até que este não possa ser resolvido recorrendo exclusivamente às ajudas.

Para evitar que o programa entre em ciclo infinito a função **apagar_pecas** tem um nº máximo de ciclos que pode fazer. Caso chegue ao fim de todos os ciclos sem produzir uma solução satisfatória (por exemplo: um puzzle fácil quando é pedido um difícil) é gerado um novo puzzle completo e recomeça todo o processo.



Puzzle Final 1

Para além disso para assegurar a dificuldade de alguns puzzles o programa obriga puzzles com mais de 5 linhas e 5 colunas e com um nível de dificuldade 2 a ter pelo menos 1/6 das casas vazias não pede ser resolvida usando hints.

Conclusão

Em conclusão nós acreditamos ter cumprido os requisitos pedidos para este projeto. Contudo a *performance* poderia ser melhor especialmente em para puzzles maiores. Apesar disto estamos satisfeitos com o resultado do trabalho.