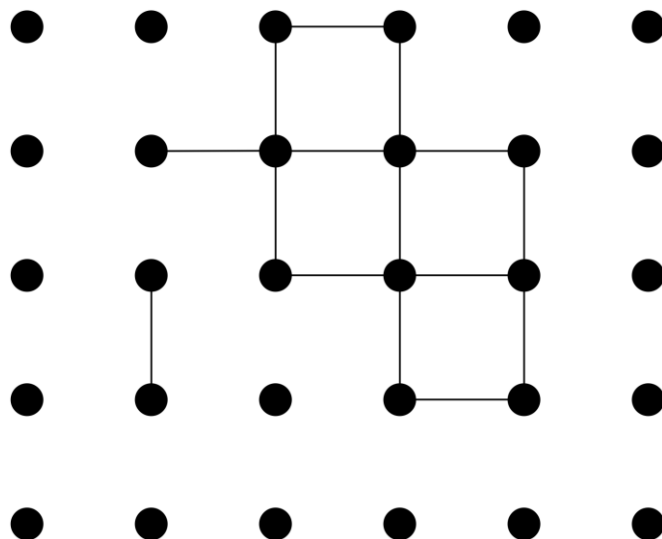


Manual



Jogo Puzzle dos Pontos e das Caixas

Data de criação: 27 janeiro de 2017

Docentes da disciplina

Joaquim Filipe

Cédric Grueau

Um trabalho realizado por

Artur Esteves 140221076 Engenharia Informática

Daniel Costa 120221058 Engenharia Informática

“Computers are like humans they do everything except think”

John von Neumann

Conteúdo

Sumário Executivo	4
Introdução	4
Descrição geral de funcionamento	4
Instalação do software LispWorks	5
Download e instalação	5
Execução do software	6
Perguntas Frequentes	9
Limitações da aplicação	9
Terminologia e Abreviaturas	10

Sumário Executivo

O manual de utilizador é um manual concebido com o propósito de ensinar o utilizador a trabalhar com o programa que gera o jogo designado **puzzle dos pontos e das caixas** e entender mais claramente que opções lhe são expostas facilitando, deste modo, a sua interação com o mesmo.

Este documento contém toda a informação que o utilizador necessita para explorar o programa de modo adequado a todos os níveis.

Uma vez visto todo o manual, o utilizador deverá ser capaz de executar a aplicação e poderá visualizar as estatísticas referentes ao jogo.

Introdução

Este programa visa conceder ao utilizador a hipótese de jogar o puzzle dos pontos e das caixas contra um adversário, sendo ele um humano ou uma máquina. Consiste num jogo tático para dois jogadores, e tem como objetivo de conseguir fechar o maior número de caixas através do acréscimo de arcos, entre dois pontos adjacentes tanto na horizontal como vertical, a partir de uma configuração inicial do tabuleiro. O jogador que ganha a partida é o jogador que tiver fechado o maior número de caixas.

Descrição geral de funcionamento

O puzzle é constituído por um tabuleiro de $n * m$ caixas, onde n é o número de arcos verticais e m o número de arcos horizontais, o que neste caso vamos lidar com um tabuleiro 7×7 caixas.

Cada caixa está delimitada por 4 pontos, $(n+1) * (m+1)$ pontos e $[(n+1) * m] + [n * (m+1)]$ arcos. Cada caixa está delimitada por 4 arcos, 2 horizontais e 2 verticais. Um tabuleiro é constituído por $(n+1) * (m+1)$ pontos e $[(n+1) * m] + [n * (m+1)]$ arcos.

O jogo possui as seguintes componentes:

- 1 Jogador
- 1 Problema
- 1 Objetivo

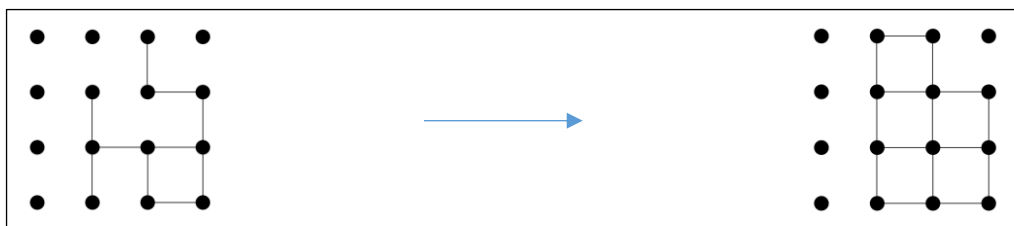


FIGURA 1 EXEMPLO DO JOGO

Este exemplo (Figura 1 Exemplo do jogo) é uma caixa com 9 caixas ($n=3$ e $m=3$), com 10 arcos conectados e 1 caixa fechada, sendo o objetivo deste problema fechar 5 caixas.

Instalação do software LispWorks

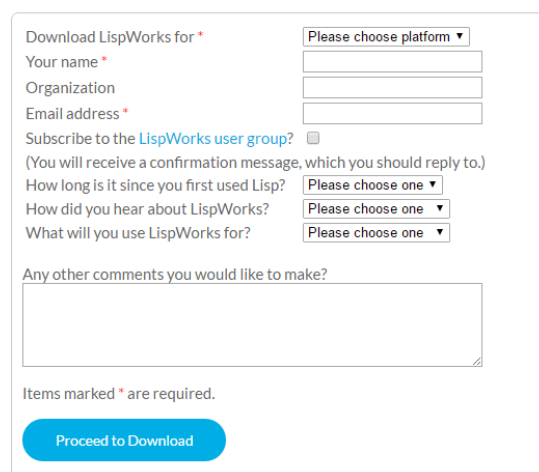
Os passos que se seguem, servem de apoio à instalação do software LispWorks, para o sistema operativo Windows, este é essencial para executar o projeto. No caso de possuir outro sistema operativo será necessário verificar as indicações fornecidas pelo site do software.

Download e instalação

O primeiro passo consiste na instalação da aplicação, na qual deverá se registar em <http://www.lispworks.com/downloads/index.html> (visto em 13-12-2015 pelas 11:54h). Depois de escolher o sistema operativo adequado e preencher os campos obrigatórios. Após o preenchimento desses, poderá descarregar o ficheiro de instalação, carregando no botão “Proceed to Download” (Figura 2 Instalação do software LispWorks).

Downloading LispWorks™ Personal Edition

Please provide us with some information about yourself.



The screenshot shows a web form for downloading LispWorks. It includes fields for 'Your name', 'Organization', and 'Email address', all marked with an asterisk. There is a dropdown menu for 'Please choose platform' and another for 'Please choose one' for 'How long is it since you first used Lisp?'. A checkbox is present for 'Subscribe to the LispWorks user group?'. Below these are more dropdowns for 'How did you hear about LispWorks?' and 'What will you use LispWorks for?'. A large text area is provided for 'Any other comments you would like to make?'. At the bottom, a blue button labeled 'Proceed to Download' is visible.

FIGURA 2 INSTALAÇÃO DO SOFTWARE LISPWORKS

Execução do software

Após ter instalado o software, é necessário seguir os passos seguintes:

1. Para executar o programa, é necessário seleccionar o ficheiro `jogo.lisp` através da opção: File->Open... (Figura 3 Seleccionar o ficheiro).

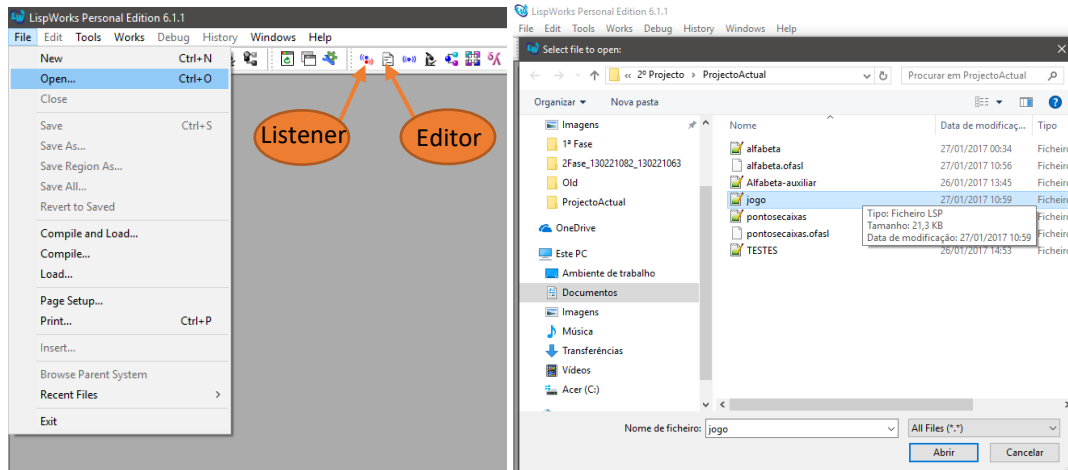


FIGURA 3 SELECIONAR O FICHEIRO

2. Após ter seleccionado e aberto o ficheiro, deverá compilar o ficheiro através da opção que mostra a seguinte figura (Figura 4 Compilar ficheiro).

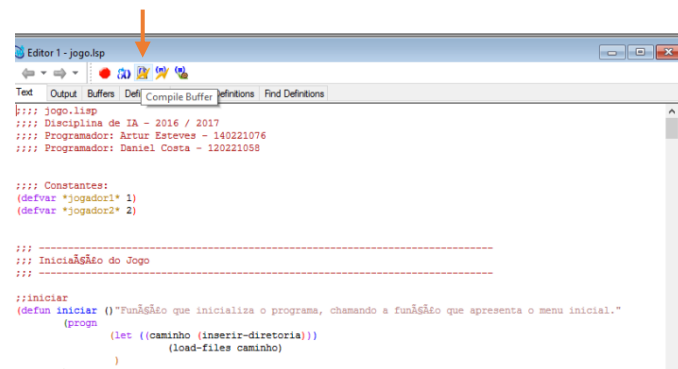


FIGURA 4 COMPILAR FICHEIRO

3. Após ter compilado o ficheiro, na janela (*Listener*), executa-se a função (**iniciar**) para dar início ao programa. Ele vai pedir a diretoria onde se encontram os restantes ficheiros pertencentes ao jogo. (Figura 5 Inserir diretoria dos ficheiros)

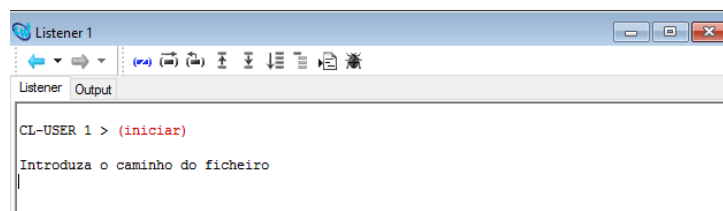


FIGURA 5 INSERIR DIRETORIA DOS FICHEIROS

- Após o utilizador ter inserido o caminho, o programa vai compilar e depois ler todos os ficheiros pertencentes ao seu funcionamento, e após executa-los, será apresentado o menu que permite resolver um puzzle. Neste menu existem 2 opções iniciais, a de iniciar a jogada, número 1, e para sair do jogo é o número 2. (Figura 6 Iniciar Jogo - Menu).

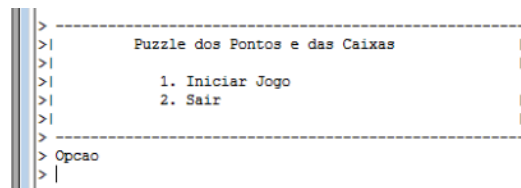


FIGURA 6 INICIAR JOGO - MENU

- Ao seleccionar 1 do menu anterior, será apresentado um outro menu onde divulga ao utilizador duas opções de jogo: Humano vs. Humano e Humano vs. Computador. (Erro! A origem da referência não foi encontrada.).

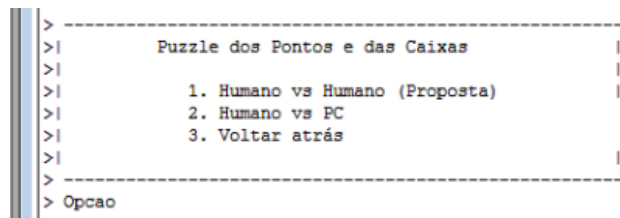


FIGURA 7 MENU PROBLEMAS

- Na opção 1 (Humano vs. Humano), será iniciado o jogo entre dois humanos (player1 e player2), na qual apresenta uma janela de seleção do jogador (Figura 8 Selecionar jogador). Caso o utilizador que vai iniciar a jogada, pretende ser o Jogador, então os arcos na qual executa a ação serão — (caso horizontal) e | (caso vertical). Caso contrário (ser jogador 2), os arcos serão --- (caso horizontal) e | (caso vertical).

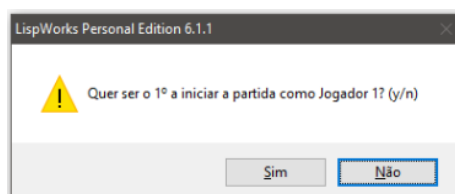


FIGURA 8 SELECIONAR JOGADOR

- Na opção 2 (Humano vs. Computador), é o jogo entre o humano contra a máquina (computador), na qual será apresentada a mesma janela de seleção do jogador (Figura 8 Selecionar jogador). Neste caso, se o for selecionado sim, então o primeiro a executar a jogada será o humano, caso contrário será a máquina a inicializar a jogada.

8. No caso da máquina a jogar, será mostrado o tabuleiro inicial, com a primeira jogada efetuada, e de seguida mostra um menu para o próximo jogador, ou seja, o humano, de modo a selecionar qual a operação que pretende executar, colocar um arco horizontalmente ou verticalmente. Após selecionar a operação, será mostrado as coordenadas que pretende colocar o arco, coordenada x e coordenada y, tendo ambas uma restrição fase ao limite do tabuleiro. (Figura 9 Jogada computador - menu arco). O mesmo processo será idêntico ao tipo de jogo (Humano vs. Humano), só que neste, não existe a jogada do computador, mas sim de outro jogador humano.

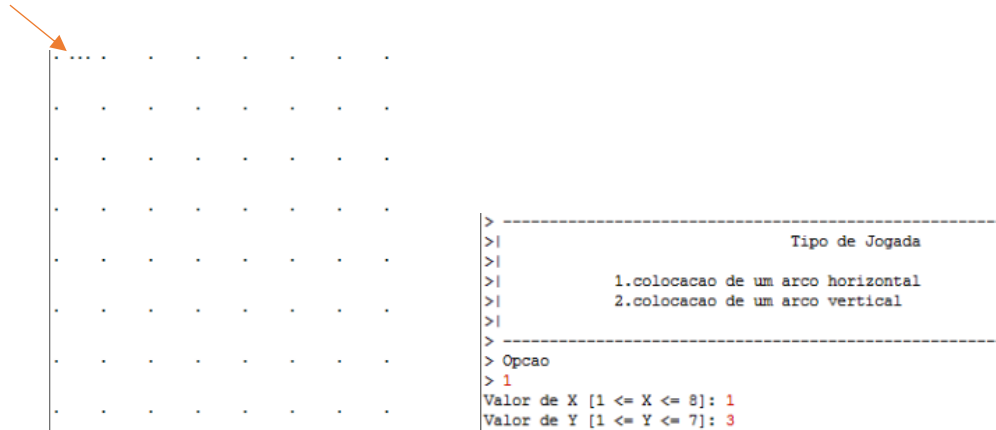


FIGURA 9 JOGADA COMPUTADOR - MENU ARCO

Nota

Caso a casa pretendida encontra-se ocupada será mostrado a seguinte mensagem:
"Esta casa já esta ocupada."

9. A jogada termina caso encontre um vencedor ou quando ocorre um empate. Neste exemplo podemos verificar o resultado final com a indicação do vencedor desta partida. (Figura 10 Vencedor da partida - Resultados). Os resultados mostrados na consola, também serão escritos num ficheiro LOG.dat, (ficheiro este que será guardado na mesma diretoria que se encontra o projeto).

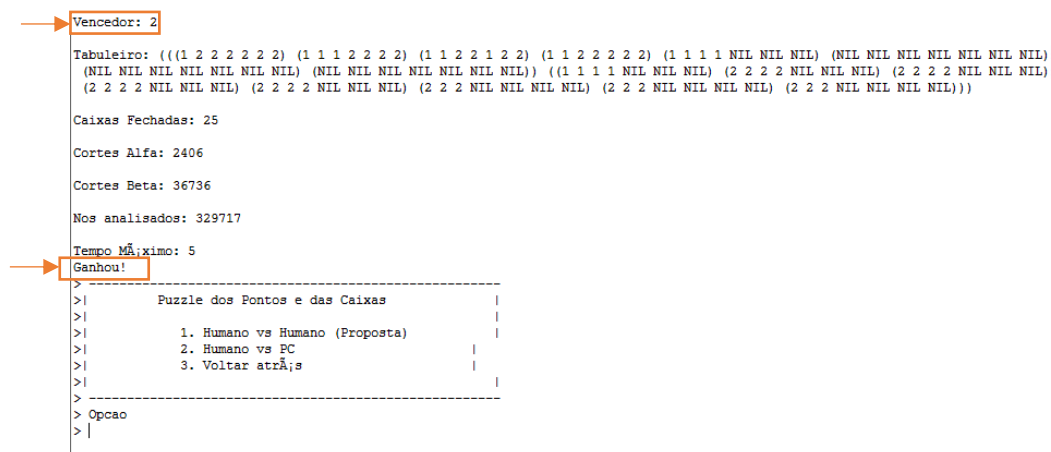


FIGURA 10 VENCEDOR DA PARTIDA - RESULTADOS

Perguntas Frequentes

- **O que faço quando o programa dá erro?**
 - Nestas ocasiões, se o programa não voltar ao menu que se encontrava, deverá voltar a escrever **(iniciar)** na consola. (Preferencialmente fazer *Abort* para não haver erro numa nova tentativa).

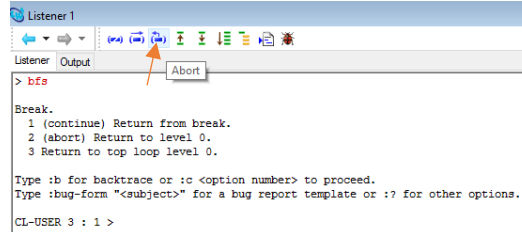


FIGURA 11 ABORT

- **O que acontece se eu não conseguir alcançar o objetivo do jogo?**
 - O objetivo consiste em fechar o maior numero de caixas possível, de modo a vencer o jogo. Caso não consiga e tentar uma nova jogada e optar por uma estratégia de jogo.
- **Este jogo aplica-se em outro dispositivo (ex: telemóvel)?**
 - Este projeto poderá ser aplicável para, por exemplo, dispositivos móveis, no entanto, limitámos o jogo apenas ao modo consola no computador, usando a linguagem *Common Lisp* e o IDE *LispWorks*.
- **Posso ter os ficheiros distribuídos pelo computador?**
 - Não. Quando o utilizador insere a diretoria, assim que escreve na consola **(iniciar)**, será necessário que todos os ficheiros pertencentes ao jogo estejam juntos, visto que o programa esta preparado para compilar e ler numa só vez.

Limitações da aplicação

Este projeto apresenta uma limitação que é a eficiência da jogada do computador. Este problema deve-se ao fato do computador analisar todas as possibilidades de colocar um arco, tendo em conta a jogada do humano. A função de utilidade utilizada neste projeto também não é melhor, isto significa uma diminuição no desempenho do algoritmo alfabeta. É de notar que estamos a lidar com um tabuleiro de dimensão 7 x 7, o que dá um total de 49 caixas, que também irá influenciar as jogadas que o utilizador fizer, isto porque, não existe qualquer elemento (por exemplo cor) que faça uma distinção vistosa entre jogador 1 e jogador 2.

Terminologia e Abreviaturas

Estrutura – Conjunto de relações entre os elementos de um sistema;

- **Menu** – Lista que aparece no ecrã do computador, apresentando as opções que determinado programa ou funções permitem;
- **Opção** – Ato de escolher algo;
- **Opção Inválida** – Decisão incorreta, o programa não permiti;
- **Sair** – Abandonar o programa;
- **Voltar** – Regressar ao menu anterior.
- **Algoritmo** - sequência finita de *instruções* bem definidas e não ambíguas, cada uma das quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finita.
- **Minimax** – algoritmo usado na Teoria da Decisão, Teoria dos Jogos, Estatística e Filosofia para minimizar a perda máxima possível. Determina a estratégia ótima para o MAX.
- **Alfa – Beta** – algoritmo utilizado para realizar a procura numa estrutura de árvore, recorrendo a cortes de modo a ser mais eficiente que o Minimax, isto é, sem analisar todos os nós da árvore.
- **Função utilidade** – função que permite calcular a utilidade de um estado sem explorar toda a subárvore respetiva.
- **MAX** – maximiza o valor da sua função de utilidade. Representa o jogador que toma decisões.
- **MIN** - representa o adversário. Minimiza o valor da sua função de utilidade.
- **Software** - Sequência de instruções, seguidas por regras, que são executadas ou manipuladas de um programa de um computador.