

Universidade Metodista de Piracicaba

Trabalho de Inteligência Artificial

LABIRINTO

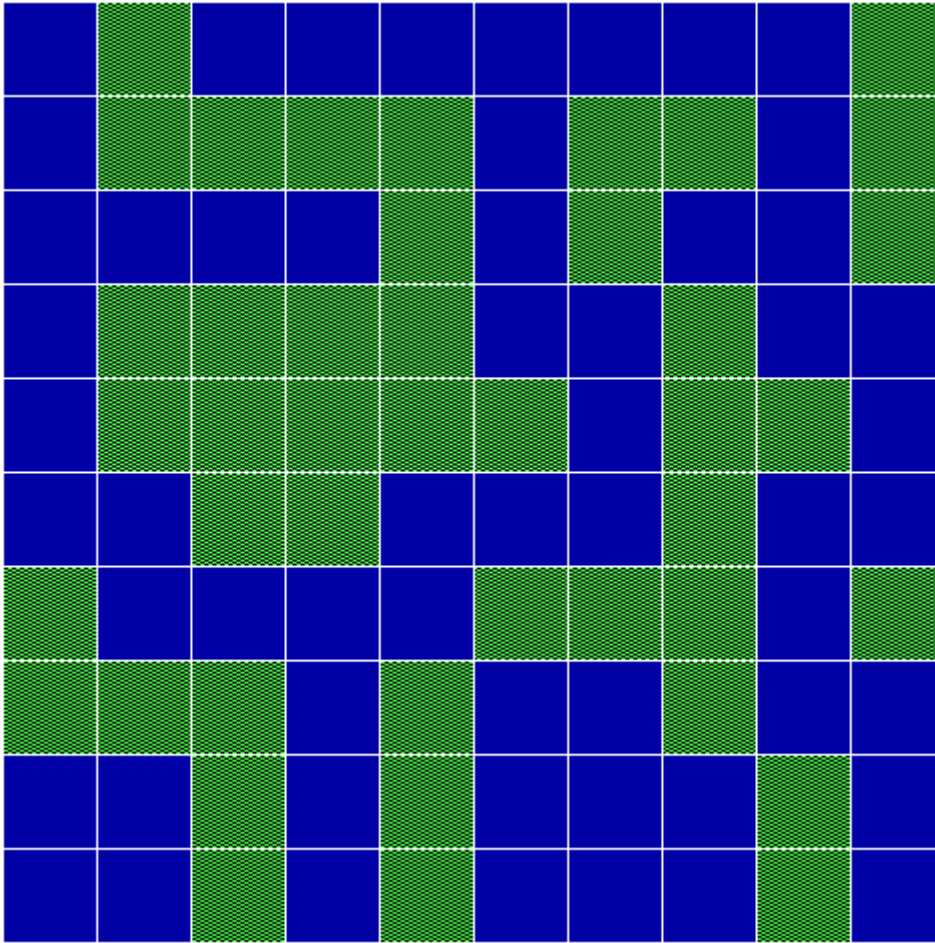
**Adriano Gheller Bruschi
Fabricio Aparecido Breve
Luis Gustavo Giordano**

Descrição do Universo

Um robô deve caminhar por um labirinto até encontrar a saída.

Ambiente: o labirinto fica localizado em uma sala que está dividida em 100 quadros (10x10). Cada espaço pode estar livre (o robô pode passar), ou bloqueado (o robô não pode passar).

Exemplo:



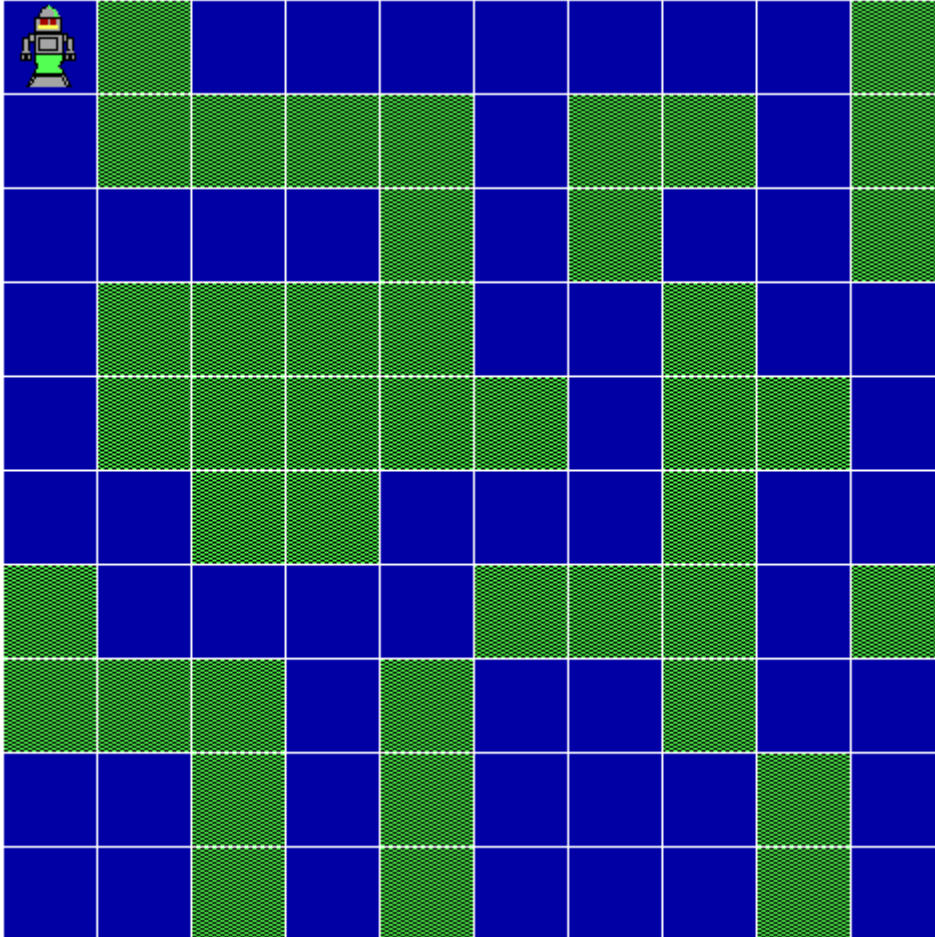
Agente: o agente é o robô, ele deve caminhar pelo labirinto até encontrar a saída ou até constatar que o labirinto não tem saída, nesse caso ele deve voltar para a entrada.

Conhecimento: o robô sabe apenas que o labirinto tem 10x10 e que o Espaço 10,10 é a saída. Os caminhos são aleatórios, portanto ele não sabe qual o caminho que leva até a saída, e também não sabe se existe um caminho que leva até a saída, pois pode não existir uma maneira de chegar até lá ou o bloco da saída pode estar fechado.

Objetivo: Chegar na saída caso exista um caminho. Ou voltar para a entrada caso chegue a conclusão de que não existe uma saída ou não existe uma maneira de chegar na saída.

Estado Inicial:

- O robô sempre começa no espaço 1,1 (superior esquerdo)



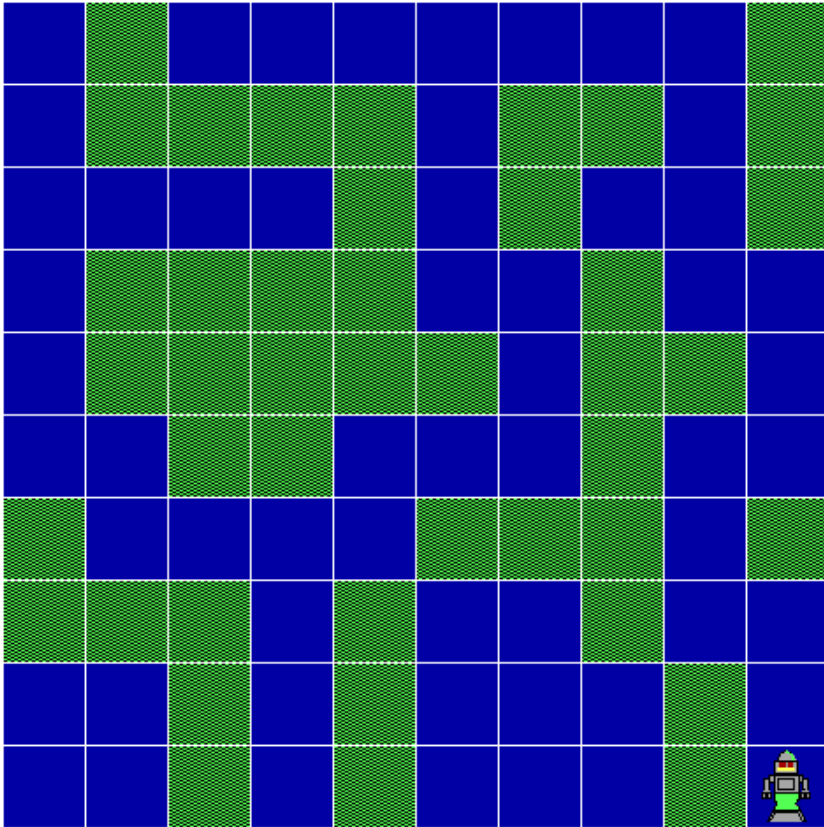
Obs: os espaços bloqueados e livres dependem do labirinto selecionado pelo usuário. Existem alguns labirintos predefinidos e o usuário também pode criar o seu próprio labirinto. O espaço inicial (1,1) deve sempre estar livre.

Estado Final:

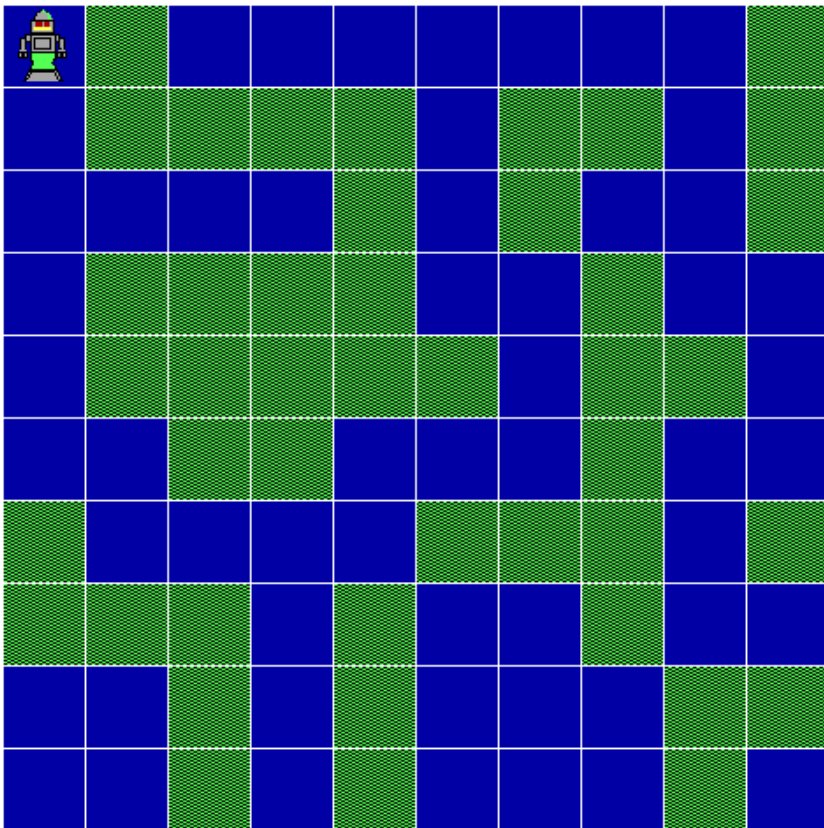
Existem 2 tipos de estados finais possíveis:

- No labirinto com saída, o robô estará na posição de saída (10,10).
- Em um labirinto sem saída, o robô volta ao início do labirinto, na posição (1,1).

Os espaços bloqueados e livres também dependerão do labirinto selecionado pelo usuário.



Labirinto com saída



Labirinto sem saída

Ações:

- Andar para cima
- Andar para baixo
- Andar para a esquerda
- Andar para a direita

O custo de cada ação é 1

Percepções:

O robô “enxerga” apenas os quadros imediatamente acima, abaixo, a direita e a esquerda do local onde ele está. Assim ele pode perceber se estes quadros estão bloqueados ou livres.

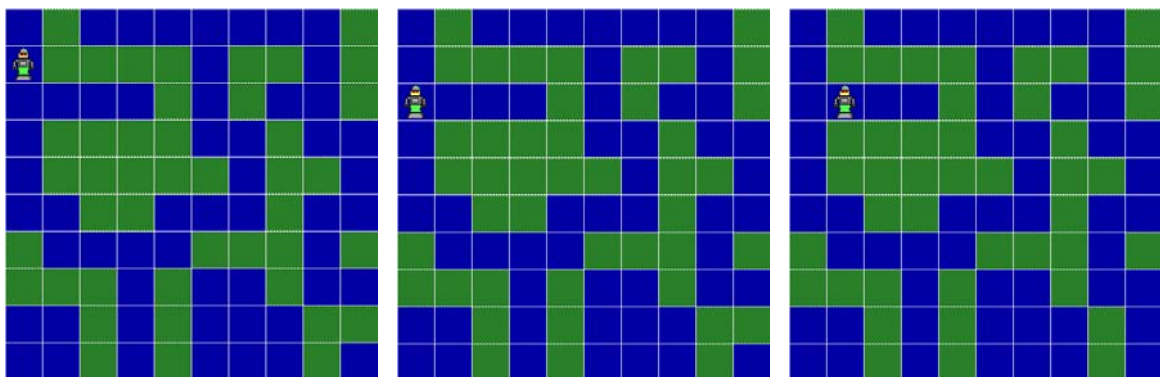
Método de Busca Utilizado:

Foi escolhido o método de busca heurística “Hill Climbing”, a seleção do caminho à ser seguido é baseada em uma lista de prioridades. Neste caso, a primeira prioridade é andar à direita, a segunda prioridade é andar para baixo, a terceira andar à esquerda, e a quarta e última é andar para cima. As prioridades se devem ao fato de o Agente saber que a saída (se existir) estará no canto inferior direito. Ele também nunca deve escolher um caminho pelo qual já tenha passado anteriormente independentemente da prioridade, se ele o fizesse ficaria andando em círculos.

Exemplo de Estados intermediários (Exemplo de execução do algoritmo)

Não é possível representar todos os estados intermediários possíveis, pois são muitos, com uma matriz 10x10 é possível criar até 2^{99} labirintos diferentes, ou seja 633.825.300.114.114.700.748.351.602.688 labirintos diferentes. E para cada labirinto o robô pode dar de 0 a 196 passos.

Representamos aqui alguns estados intermediários do labirinto já utilizado como exemplo anteriormente:

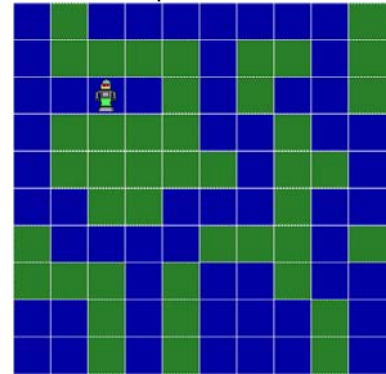
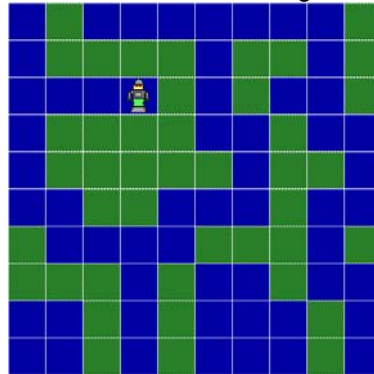
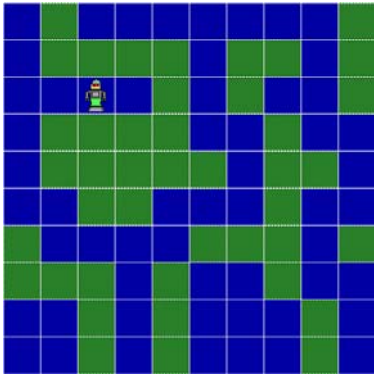


O robô começa a caminhar

Neste ponto o robô encontra

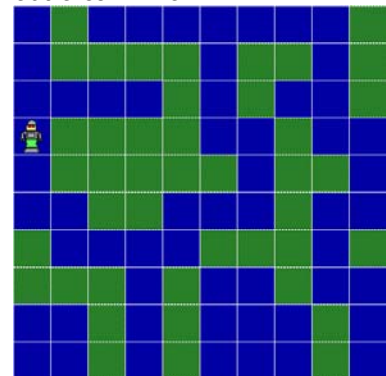
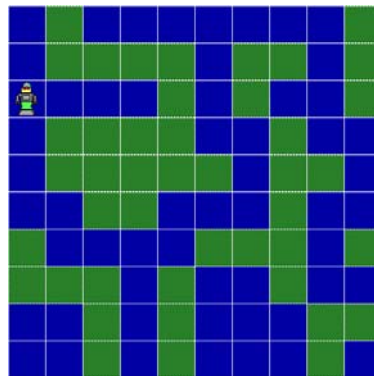
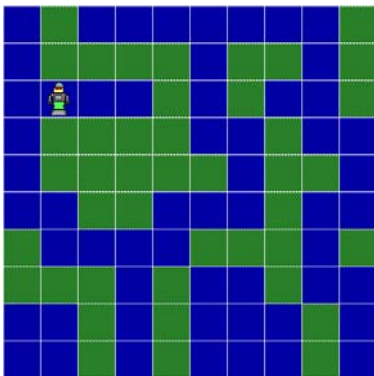
O robô escolhe entrar pelo

pele labirinto a procura da saída uma bifurcação e deve escolher o caminho a seguir. caminho a direita baseado na sua lista de prioridades.



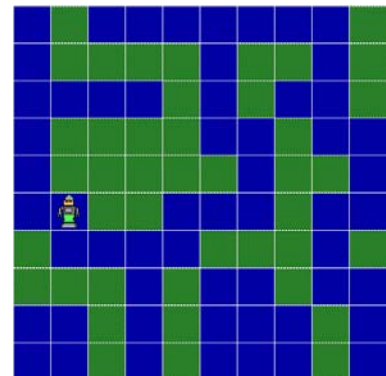
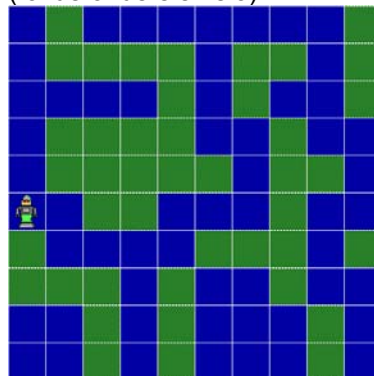
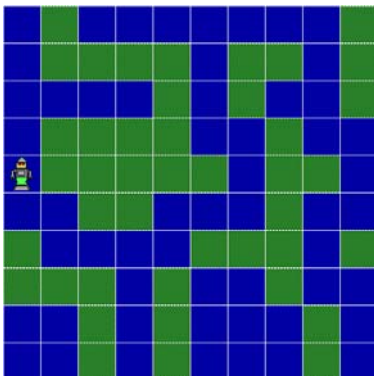
O robô percebe que escolheu um caminho sem saída.

Agora o robô deve voltar até a última bifurcação e tomar um outro caminho



O robô já sabe que pelo caminho de cima ele já passou (foi de onde ele veio)

Portanto ele decide ir para baixo agora.



E o robô continua caminhando desta forma, escolhendo um caminho em cada bifurcação de acordo com sua prioridade. E no caso de chegar em um beco sem saída, ele volta e escolhe outro caminho, até que ele consiga chegar ao final do labirinto.

Se o robô andar por todos os caminhos possíveis e não encontrar a saída, ele conclui que o labirinto não tem saída, e retorna para o início.

Algoritmo

Variáveis:

- Matriz 10x10 para armazenar onde o robô já passou. (0 = não passou, 1 = passou)
- Estrutura Bloco
 - Inteiro Linha, Coluna
 - Ponteiros: cima, baixo, esquerda, direita

Método de Busca:

```
função caminha(ponteiro bloco atual) {  
    marca espaço na matriz como já caminhado;  
    se (linha = 9 e coluna = 9) então achou saída;  
    senão {  
        se (bloco a direita não bloqueado e ainda não caminhou por ele) {  
            caminha(bloco da direita);  
        }  
        se (bloco abaixo não bloqueado e ainda não caminhou por ele)  
            caminha(bloco abaixo);  
        }  
        se (bloco a esquerda não bloqueado e ainda não caminhou por ele) {  
            caminha(bloco da esquerda);  
        }  
        se (bloco acima não bloqueado e ainda não caminhou por ele) {  
            caminha(bloco acima);  
        }  
    }  
}
```

É montada dinamicamente uma estrutura, com variáveis dinâmicas sendo alocadas para cada novo bloco visitado. Cada variável armazena a linha e coluna de cada bloco, e contém ponteiro para as variáveis dos blocos imediatamente acima, abaixo, a direita e a esquerda.

Instruções do programa:

Instalação:

Os seguintes arquivos devem estar em um mesmo diretório:

- LAB.EXE
- EGAVGA.BGI
- ROBO.TXT
- LAB1.TXT
- LAB2.TXT
- LAB3.TXT
- LAB4.TXT

Mostrando o caminhamento em um labirinto:

Basta escolher a opção desejada:

- **1** – Labirinto com saída simples
- **2** – Labirinto com saída complexo
- **3** – Labirinto sem saída
- **4** – Labirinto criado pelo usuário
- **0** – Para sair do programa

Após selecionar o labirinto desejado o modo gráfico é acionado com o mapa e o robô, pressione ENTER para iniciar a simulação.

Após o término da simulação pressione ENTER para voltar ao menu

Caso selecione a opção 4, digite o nome do arquivo do mapa. Ele será carregado e a simulação será iniciada.

Obs: Acompanha o programa um labirinto de usuário com o nome *lab4.txt*, é um labirinto interessante pois mostra como o robô não entra em caminhos já percorridos evitando ficar andando em círculos infinitamente.

Criando labirintos de usuário:

O labirinto de usuário deve estar no formato texto (.txt) e conter 10 linhas com 10 números cada (sem espaços). O número 1 representa um espaço bloqueado e o número 0 representa um espaço livre.

Exemplo:

```
0100000001
0111101101
0000101001
0111100100
0111110110
0011000100
1000011101
1110100100
0010100010
0010100010
```

Estes arquivos podem ser criados em qualquer editor de texto que não utilize formatação (Edit do MS-DOS, Windows Notepad, etc) e devem estar sempre no mesmo diretório do programa.