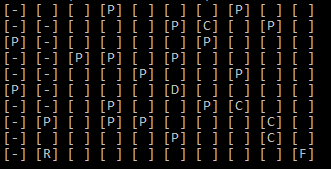
Aluno: André Luiz Pizani  
Disciplina de Inteligência Artificial - Sistemas de Informação 2017-1  
Professor: Aran Morales  
Disponível em: https://github.com/andrepizani/TrabalhoIA

Linguagem Utilizada: Java  
  
Legenda:  
R –Robô  
C –Vida de 5  
D –Vida de 10  
P – Parede  
- – Caminho fechado



O programa escrito em Java define **randomicamente** as posições das Paredes [P], das Vidas com valor +5 [C] e das Vidas com valor +10 [D] no labirinto onde o Robô [R] caminha por meio de Busca em Largura ou Algoritmo A\*.

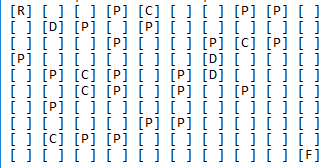
O Robô sempre nasce na posição (0,0) e só consegue “enxergar” as posições vizinhas a que se encontrar e não pode andar na diagonal, seu objetivo final é chegar na posição (9,9).

Caso o Robô fique em uma posição sem saída, seja cercado por paredes e/ou posições que já estão setadas como “fechadas”, o Robô morre, exibindo mensagem de “Game Over”

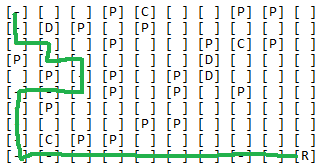
O Robô tem uma vida de valor 50, a cada posição que caminha ele perde -1 de vida, em algumas posições há vidas de valores 5 e 10, o qual aumenta sua vida referente ao valor da posição, caso a vida do Robô chegue a 0, o Robô morre, exibindo mensagem de “Game Over”

* Busca em Largura
  + Sentido: Esquerda, Baixo, Direita, Cima;
  + Sempre verificará na ordem acima se a posição existe ou está disponível;
  + Uma posição é definida como não disponível quando o robô já passou por ela ou é uma Parede [P].
  + Exemplo:

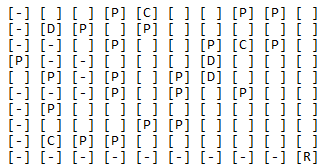
**Estado Inicial:**

****

**Trajetória:**

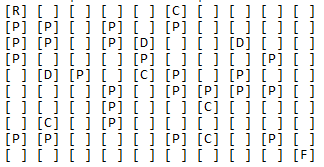
****

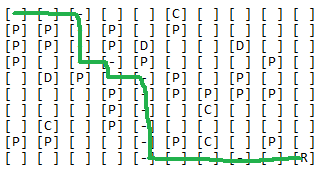
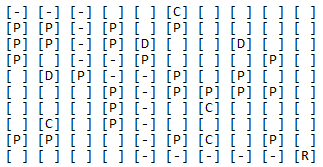
**Estado final:**

****

* Algoritmo A\*
  + Identifica o melhor caminho por meio do menor valor de movimentação dos vizinhos;
  + O valor de movimentação (menorValor) é composto pelo Número de Passos Realizados + número de casas até a posição final (9,9)
  + Caso duas posições vizinhas tenham o mesmo de valor de movimentação, ele optará pela seguinte ordem: Baixo, Direita, Cima, Esquerda.
  + Exemplo:

**Estado Inicial:**

**Trajetória**

**Estado Final:**

