

Universidade Federal do Amazonas - UFAM  
Instituto de Computação - Icomp

**Trabalho Prático de AED 2**  
Relatório - Problema 2: labirinto

Equipe:  
Alexandre da Fonseca Vieira  
Teodório Ferreira Neto  
Thiago de Sá Queiroz

Manaus - AM  
2023

### **Descrição do problema escolhido:**

Implementar um gerador de labirinto aleatório que possui apenas uma saída, junto com uma interface que possibilite o usuário andar pelo labirinto criado. Elaborar uma funcionalidade dentro da interface do jogo que permite mostrar a solução do labirinto através de uma estrutura de fila de prioridades.

### **Explicação do procedimento de busca implementado:**

A ideia da nossa estratégia de busca é ligar o algoritmo do gerador de labirintos com o algoritmo A\*, que foi informado no enunciado. A intenção é repassar a lista com as transições válidas que é produzida dentro do labirinto para o algoritmo A\*, para então ele encontrar a saída. O conteúdo da solução produzida pode ser acessado através do botão presente na interface que informa ao usuário os passos para solucionar o labirinto.

### **Discussão sobre as decisões de projeto para elaboração de interface:**

Foi escolhido utilizar o pygame, pois o mesmo apresenta ser mais simples e intuitivo quando comparado ao tkinter. Ele também apresenta ser o mais ideal para o problema escolhido pelo grupo.

O código responsável por criar a interface do jogo inicialmente vai receber uma variável que contém a matriz resultado produzida em um código separado chamado “geraLabirinto\_13x15.py”, para em seguida mapear graficamente o conteúdo da matriz. Como também vai receber uma variável que guarda a posição do primeiro campo definido para posicionar o ponto de partida do jogador, e uma outra variável para guardar a posição da saída, ambas variáveis auxiliam na interface.

Para desenvolver a interface, foi utilizado a seguinte legenda presente na matriz do labirinto como base:

- O número 0 representa os locais inacessíveis do labirinto;
  - Na interface, os locais inacessíveis são representados por quadrados pretos.
- O número 2 também representa os locais inacessíveis do labirinto, porém também ilustram a borda e auxiliam na validação de posições próximas do limite do labirinto;
  - Na interface, as bordas são representadas por quadrados cinzas.
- O número 1 representa os locais acessíveis do labirinto.
  - Na interface, os locais acessíveis são representados por quadrados brancos.

Utilização de comandos pré-definidos do pygame, pois eles facilitam o acesso a teclas do teclado e mouse, o que agiliza o desenvolvimento de controles. A parte gráfica em pygame também é simples e intuitiva, permitindo exportação de imagens ou utilizar formatos geométricos simples.

Foi implementado o botão “Desistir” para mostrar a solução do labirinto (essa solução não foi implementada).

### **Instruções de instalação e uso do aplicativo desenvolvido:**

Atenção: é preciso instalar o pygame para poder rodar o jogo.

Para iniciar o jogo basta escrever no terminal ‘python Game.py’ ou ‘python3 Game.py’, que o jogo inicializará automaticamente.

O movimento do usuário é através das setas do teclado. O jogador poderá mover-se nos quadrados brancos.

Algumas observações:

- Durante a execução do código, o labirinto gerado pode não funcionar, pois o código dele está instável, então pode ser necessário rodar novamente o código até gerar o comportamento esperado.
- No repositório estão presentes 5 arquivos com código, porém só 3 estão sendo utilizados para produzir o resultado final do trabalho. Portanto, o “FilaPrioridade.py” e “algoritmoEstrela.py” não foram implementados, mas as suas ideias estão presentes nos seus respectivos arquivos.