Sistemas de Informação — DECSI - UFOP
CSI105 Algoritmos e Estruturas de Dados III — 2023/2
Professor: Dr. George Henrique Godim da Fonseca — Data: 20/12/23

Aluno: Matrícula: Valor: 25,0 Nota:

Trabalho Prático II

1. Objetivos.
   - Desenvolver a habilidade de leitura e manipulação de dados.
   - Reforçar o aprendizado sobre estruturas de dados em grafos.
   - Aplicar os conhecimentos em grafos para resolver problemas reais.

2. Descrição.
   Esse trabalho contempla a continuação do Trabalho Prático I, adicionando características que aproximam o trabalho do problema real de deslocamento de equipamentos em ambiente 3D. Além dos requisitos do trabalho I, nesse trabalho será considerado:

   (a) que a área de *laydown* (destino) pode ocupar vários pixeis e pode ser em mais de uma localidade;

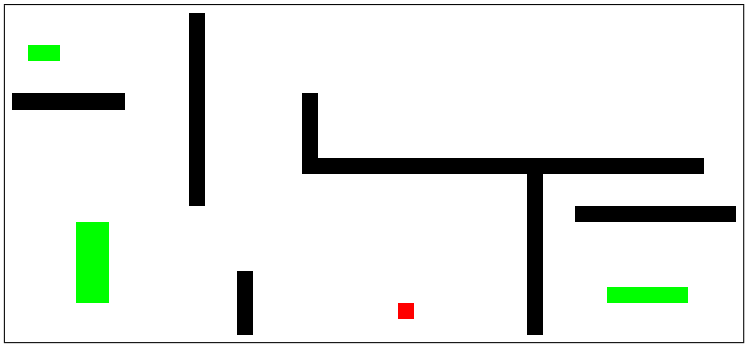
   (b) que o deslocamento deve, sempre que possível, passar mais distante possível dos objetos de colisão para maior segurança das operações;

   (c) a área de deslocamento é 3D, sendo possível o deslocamento de um andar imediatamente superior para um imediatamente inferior e vice-versa;

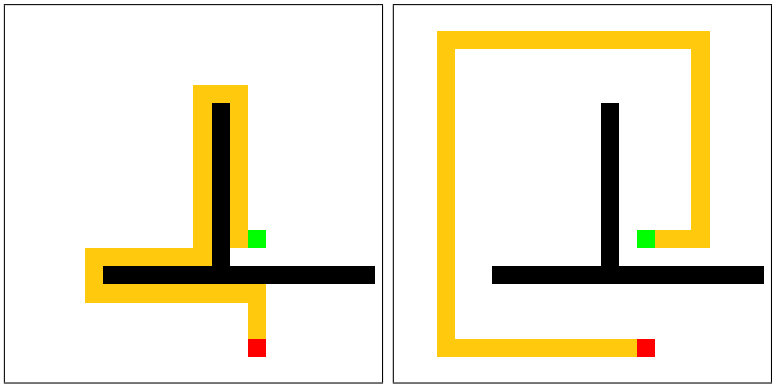
   (d) EXTRA o objeto a deslocar pode ocupar mais pixeis e não-necessariamente ter formato quadrado.

   É compreensível que nem todas as adequações sejam feitas, assim a nota será proporcional aos requisitos contemplados. Maiores detalhes de cada requisito são dados a seguir.

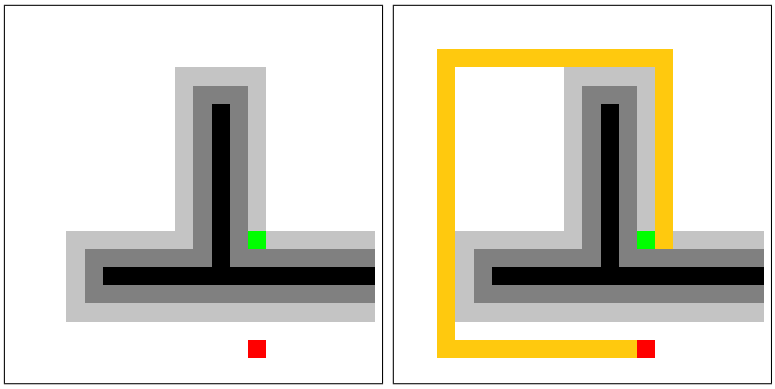
   Múltiplas áreas de *laydown* deve-se computar o melhor deslocamento do objeto para qualquer uma das áreas de laydown, como no exemplo a seguir, onde é mais interessante deslocar o objeto para o *laydown* da esquerda.



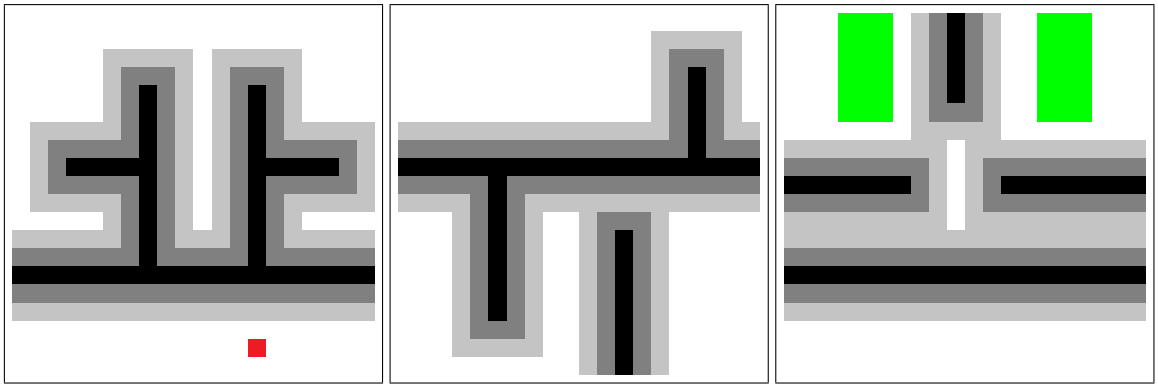
   Evitar passar próximo dos obstáculos no mundo real é preferível deslocar o objeto com uma margem de segurança dos obstáculos ao deslocamento para evitar conflitos e promover maior segurança às operações. Compare, por exemplo, a solução (em amarelo) à esquerda e a outra à direita. Qual delas é mais segura no mundo real?



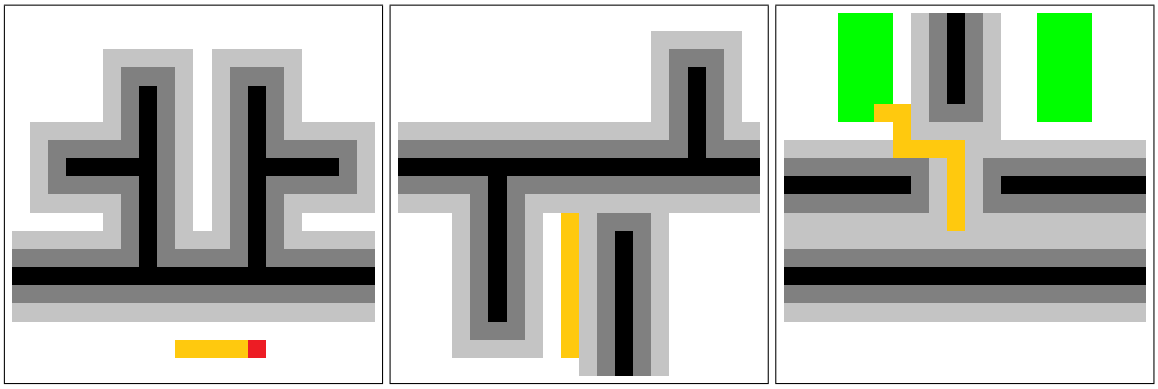
Para implementar esse requisito teremos que considerar que ligações para nós (pixeis) mais próximos aos pixeis pretos devem ter peso maior de modo a serem menos favorecidas no deslocamento. De modo a facilitar essa tarefa, as imagens de entrada agora podem contemplar pixeis cinza escuro (128, 128, 128) e cinza claro (196, 196, 196). Ligações para pixeis cinza escuro devem ter peso 2 e ligações para pixeis cinza claro devem ter peso 1,5 (lembrando que ligações para pixeis brancos têm peso 1). Naturalmente, a introdução de pesos implica na aplicação de algoritmos de caminho mínimo, como o Dijkstra, para resolver o problema. As figuras a seguir apresentam um exemplo contendo pixeis cinza e uma possível solução que evita passar por eles.



Deslocamento 3D seu algoritmo deve contemplar o planejamento 3D, ao permitir que o objeto desloque do nível (andar) atual para o nível imediatamente superior ou inferior caso não haja obstáculos ao deslocamento. Ou seja, o objeto pode ser mover não apenas para esquerda, direita, frente e trás, mas também para cima e para baixo. Para contemplar esse requisito, as entradas agora podem contemplar múltiplos andares, cujo número será informado no nome do arquivo, ex: `toy_0.bmp` representa o andar 0 e `toy_1.bmp` representa o andar 1. O deslocamento de um andar para o outro deve ser evitado, por ser mais dispendioso operacionalmente; assim, as ligações de nós de um andar para nós de outro devem ter peso 5. O objeto e a(s) área(s) de *laydown* podem não estar no mesmo andar, como no exemplo a seguir, de três andares:

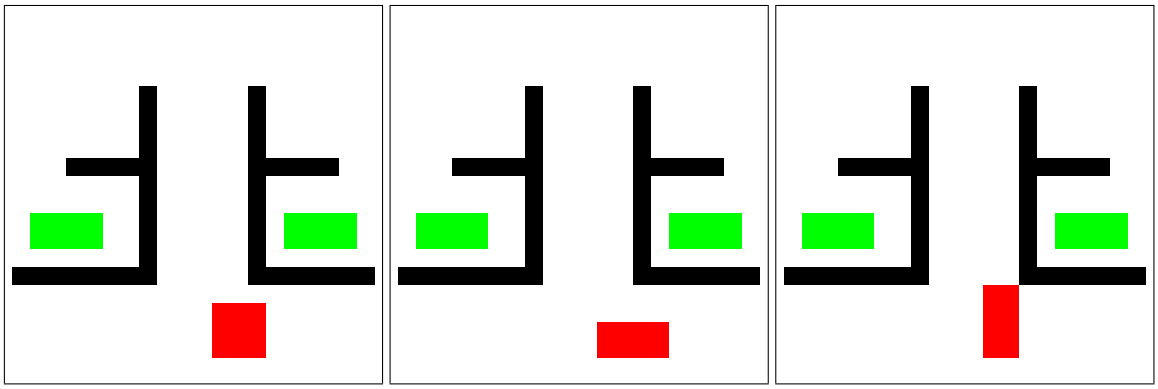


Vejamos uma possível solução para o exemplo:



Geometria do objeto (EXTRA - até 5 pontos) no mundo real, o objeto a ser deslocado usualmente ocupa mais de um píxel na imagem correspondente. Podemos considerar então que o ponto vermelho na verdade é uma caixa de pontos vermelhos, que pode, inclusive, ter qualquer formato como por exemplo retangular. Nesse sentido, ao deslocar o objeto, não só um ponto deve ser considerado, mas deve-se verificar se todo o seu corpo passa sem conflitos no deslocamento.

No exemplo abaixo não seria possível deslocar o objeto na figura do centro devido à sua geometria. Note que não é solicitado considerar a rotação do objeto, mas caso queira se desafiar, seria um exercício interessante.



3. Interação com o usuário

   A interação com o usuário deve ocorrer no arquivo `main` do seu programa. O mesmo deve solicitar ao usuário o nome da pasta onde os arquivos bitmap de entrada estão localizados (considerando o exemplo `toy_0.bmp`, `toy_1.bmp` e `toy_2.bmp`)e, após a execução, informar o caminho a ser seguido pelo equipamento. Segue um exemplo de interação com o programa:

```
Informe a pasta contendo o(s) arquivo(s) bitmap: <toy>
Processando...

É possivel deslocar o equipamento:
← ← ← ← ^ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑ ^ ↑ ↑ ↑ ↑ ↑ ← ← ← ↑ ↑ ←
```

4. Avaliação.

   O trabalho deverá ser feito individualmente ou em dupla e enviado via Moodle até as 23:59h do dia 16/02/23. Caso se tenha alguma dúvida com relação à autoria do trabalho o professor poderá solicitar uma apresentação presencial ao aluno (ou dupla). O uso de ferramentas de IA, como ChatGPT e Copilot, é permitido e incentivado, porém tentem sempre entender o que a(s) ferramenta(s) estão sugerindo para identificar eventuais bugs (que acontecem com frequência). O plágio de trabalhos de outrem será duramente punido com a perda total dos pontos em ambos os trabalhos.

5. Pontos extra.

   Serão atribuídos até 5 pontos extras para os alunos que contemplarem o requisito (d) do enunciado.

Bom trabalho!