



Resolução do Elo Maluco - 2º SEMESTRE/2024

INTRODUÇÃO:

Conforme apresentado pelo vídeo¹ e a Figura 1, o elo maluco é um jogo bastante popular na década de 80. Como projeto de desenvolvimento para a disciplina de PAA, o time deverá propor e desenvolver um algoritmo utilizando alguma estratégia de programação, como por exemplo “*Backtracking*”, que seja capaz de encontrar qual o caminho de movimentos que resolva o jogo do elo maluco. Este deverá ser implementado utilizando a linguagem de programação C/C++, e as bibliotecas que se fizerem necessárias.



Figura 1: Brinquedo Elo Maluco.

Conforme solicitado, para realizar o desenvolvimento do algoritmo de solução do brinquedo, o programa a ser desenvolvido deverá possuir as seguintes características e funcionalidades:

- ENTRADA:

- O software deverá ler um arquivo em XML contendo o estado inicial do jogo que, a partir deste, irá encontrar o caminho para a solução;
 - O software possibilitará a geração aleatória de um dado estado inicial do jogo que, a partir deste, irá encontrar o caminho para a solução;

- PROCESSAMENTO:

- O algoritmo para solucionar o jogo poderá utilizar alguma técnica de programação vista em sala de aula;
 - O software deverá possuir uma funcionalidade de apresentar qual o estado atual do jogo, isto é, a cada passo da busca pela solução, poderá haver uma opção de apresentar o estado atual do jogo;

¹<https://www.youtube.com/watch?v=1bPjr2fmY-k> Acessado em 12/08/2024.

- SAÍDA:

- O algoritmo irá finalizar quando todas as possibilidades foram verificadas ou então quando o estado atual do jogo for uma solução encontrada, solucionada;
- Uma vez encontrada a solução do jogo, esta deverá ser apresentada em tela;
- E também gerada um arquivo contendo o estado inicial do jogo, bem como a sequência de passos realizados para se chegar na resolução do jogo.
- Para elaboração do relatório, deverá ser apresentado o resultado do jogo para no mínimo 3 instâncias de entrada.

Formato do arquivo de entrada (contém um estado do Elo Maluco)

O arquivo será em formato XML contendo uma descrição da seguinte forma, a primeira linha representa as 4 faces superiores do Elo Maluco, separadas pelas colunas, e assim por diante. As faces serão representadas pelos seguintes códigos:

Duas letras indicando a cor seguidas de uma letra indicando a parte do Elo, como se segue na Tabela 1:

Tabela 1: Codificação do jogo.

Cor	Código	Parte do Elo	Código
Vermelho	vm	Superior	s
Verde	vr	Intermediário	m
Amarelo	am	Inferior	i
Branco	br		

A face vazia será indicada pelas letras - vzo.

- *Observações:*

A corretude no formato do arquivo é de total responsabilidade do jogador. Segue um exemplo de arquivo de entrada para um dado estado do jogo, conforme observado pela Figura 2.

```
<?xml version="1.0" encoding="UTF-8" ?>
<EloMaluco>
  <EstadoAtual>
    <row>
      <col>vms</col>
      <col>ams</col>
      <col>vds</col>
      <col>brs</col>
    </row>
    <row>
      <col>vmm</col>
      <col>amm</col>
      <col>vdm</col>
      <col>brm</col>
    </row>
    <row>
      <col>vmm</col>
      <col>amm</col>
      <col>vdm</col>
      <col>vzo</col>
    </row>
    <row>
      <col>vmi</col>
      <col>ami</col>
      <col>vdi</col>
      <col>bri</col>
    </row>
  </EstadoAtual>
</EloMaluco>
```

Figura 2: Arquivo XML com o formato de entrada para o estado do jogo. Este arquivo de exemplo mostra o estado do jogo solucionado.

Formato do arquivo utilizado para exibição de estados passo a passo

O arquivo conterá nas primeiras linhas uma descrição de estado inicial como do arquivo citado anteriormente. Em seguida em uma só linha, separados por espaço, estarão as ações sobre o elo, de acordo com a codificação dada pela Tabela 2. A Figura 3 apresenta as informações do arquivo de entrada contendo os dados iniciais do jogo e os passo que levam para a solução.

Tabela 2: Ações do jogo.

Ação	Código
Rotacionar a parte superior para a direita	rsd
Rotacionar a parte superior para a esquerda	rse
Rotacionar a parte inferior para a direita	rid
Rotacionar a parte inferior para a esquerda	rie
Mover face abaixo da vazia para cima	mfc
Mover face acima da vazia para baixo	mfb

```

<?xml version="1.0" encoding="UTF-8" ?>
<EloMaluco>
    <EstadoAtual>
        <row>
            <col>vms</col>
            <col>vds</col>
            <col>brs</col>
            <col>vms</col>
        </row>
        <row>
            <col>vmm</col>
            <col>amm</col>
            <col>vdm</col>
            <col>brm</col>
        </row>
        <row>
            <col>vzo</col>
            <col>amm</col>
            <col>vdm</col>
            <col>bri</col>
        </row>
        <row>
            <col>vmi</col>
            <col>ami</col>
            <col>vdi</col>
            <col>vmm</col>
        </row>
    </EstadoAtual>
    <Acoes>
        <acao>rid</acao>
        <acao>mfc</acao>
        <acao>rsd</acao>
        <acao>rie</acao>
        <acao>mfb</acao>
    <Acoes>
</EloMaluco>

```

Figura 3: Arquivo XML com o formato de entrada para o estado inicial do jogo e as ações que levam à solução do jogo.

ENTREGAS:

O projeto deverá ser entregue seguindo as seguintes diretrizes.

- A equipe deverá ser composta de no **máximo 3 integrantes**.
- Deverá ser confeccionado um relatório na forma de artigo seguindo um template que será disponibilizado no SIGAA, não ultrapassando 6 páginas.
- Deverá ser gravado um vídeo explicativo da solução desenvolvida, detalhando o que foi desenvolvido na entrada, processamento e saída. O vídeo não poderá ultrapassar 10 minutos.

- A codificação deverá ser feita usando a **Programação Orientada a Objetos em C/C++**. É disponibilizado pelo SIGAA um código de base para realização do trabalho. Todas as características solicitadas anteriormente serão pontuadas no processo avaliativo.
- O código deverá ser bem documentado.
- **ENTREGA DO PROJETO 04/12/2024**

AVALIAÇÃO:

Atividade	Pesos
Apresentação Video	15%
Codificação	50%
Relatório	30%
Documentação	05%