

Matéria Adiantada dia 22/10

Semana 28 - Aula 1

Tópico Principal da Aula: Vetores e Matrizes: Criação de programas: prática.

Subtítulo/Tema Específico: O Conceito de Matrizes Aplicado ao Jogo 2048

Código da aula: [SIS]ANO1C1B4S28A1

Objetivos da Aula:

- Compreender de forma prática os conceitos relacionados a vetores e matrizes no contexto de desenvolvimento de software.
- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.
- Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes.

Recursos Adicionais:

- Caderno para anotações;
- Acesso ao laboratório e a computadores para resolução das atividades.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - O conceito de matrizes

- **Definição:** Uma matriz é uma coleção de números, símbolos ou expressões organizados em uma estrutura retangular de linhas e colunas. Cada item na matriz é denominado "elemento". Matrizes são ferramentas matemáticas fundamentais, frequentemente usadas para resolver sistemas de equações lineares.
- **Aprofundamento/Complemento:** Em programação, uma matriz é comumente implementada como uma estrutura de dados bidimensional, ou um "array de arrays". Ela é ideal para armazenar dados que possuem uma relação espacial ou tabular, como um tabuleiro de jogo, uma imagem (onde cada elemento é um pixel) ou dados de uma planilha. Acessamos seus elementos por meio de dois índices: um para a linha e outro para a coluna, geralmente no formato `matriz[linha][coluna]`.
- **Exemplo Prático:** Em Python, uma matriz pode ser representada por uma lista de listas. Uma matriz 3x3 contendo números de 1 a 9 seria:
- Python

```
matriz = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
]
```

```
# Para acessar o elemento '6' (linha 1, coluna 2 - lembrando que a contagem começa em 0)  
elemento = matriz[1][2]  
print(elemento) # Saída: 6
```

-
-
- **Vídeos Sugeridos:**
 - O que são Matrizes? (Khan Academy): https://www.youtube.com/watch?v=v_l6M_StH5E
 - Matrizes em Python (Programação Dinâmica): <https://www.youtube.com/watch?v=15s0GtrH-0g>

Referência do Slide: Slide 07/08/09/10 - Operações com matrizes no jogo 2048

- **Definição:** O jogo 2048 utiliza um tabuleiro que pode ser visto como uma matriz. As peças do jogo são os elementos da matriz. Quando o jogador move as peças (cima, baixo, esquerda, direita), todos os elementos da matriz se deslocam na direção escolhida.¹² Se duas peças com o mesmo número se encontram, elas se somam, criando uma nova peça com o dobro do valor. Isso é análogo a uma operação de adição entre elementos da matriz. Após cada movimento, um novo número (2 ou 4) é inserido em uma posição vazia da matriz.
- **Aprofundamento/Complemento:** A lógica de movimentação no jogo 2048 é um excelente exemplo de manipulação de matrizes. Para um movimento "para a esquerda", por exemplo, o algoritmo percorre cada linha da matriz. Para cada linha, ele primeiro "compacta" os números, movendo todos os valores diferentes de zero para a esquerda. Em seguida, ele verifica se há peças adjacentes com o mesmo valor para somá-las. Finalmente, ele compacta a linha novamente. Esse processo se repete para todas as linhas.
- **Exemplo Prático:** Considere uma linha do jogo 2048 representada por um vetor (uma linha de uma matriz): [2, 0, 2, 4].
 - **Movimento para a esquerda:**
 - **Compactação inicial:** [2, 2, 4, 0] (o zero foi movido para o final).
 - **Soma de adjacentes:** O primeiro '2' e o segundo '2' são iguais e adjacentes. Eles se somam: [4, 0, 4, 0].
 - **Compactação final:** [4, 4, 0, 0]. O resultado do movimento nessa linha seria [4, 4, 0, 0].
- **Vídeos Sugeridos:**
 - Como Programar o Jogo 2048 (Code Bullet): https://www.youtube.com/watch?v=b4K_s1-B4nE
 - Lógica de Matrizes e Listas Bidimensionais (Curso em Vídeo): <https://www.youtube.com/watch?v=b4p-a2jA-Uo>

Semana 28 - Aula 2

Tópico Principal da Aula: Vetores e Matrizes: Criação de programas: prática.

Subtítulo/Tema Específico: Aplicações Práticas de Matrizes em Python

Código da aula: [SIS]ANO1C1B4S28A2

Objetivos da Aula:

- Compreender de forma prática os conceitos relacionados a vetores e matrizes no contexto de desenvolvimento de software.
- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.
- 19 Resolver problemas computacionais com estratégias criativas.

Recursos Adicionais:

- Caderno para anotações;
- Acesso ao laboratório e a computadores para resolução das atividades.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Representação e Acesso a Elementos da Matriz

- **Definição:** Uma matriz é uma coleção de elementos dispostos em linhas e colunas.
23 Em Python, ela pode ser representada como uma lista de listas. O acesso a um elemento é feito por meio de índices que indicam sua posição, lembrando que a indexação em Python começa em zero.
- **Aprofundamento/Complemento:** O acesso via índice duplo (ex: `matriz[linha][coluna]`) é a forma mais comum de interagir com um elemento específico. O primeiro índice `[linha]` seleciona a lista interna (a linha), e o segundo índice `[coluna]` seleciona o elemento dentro dessa lista. Tentar acessar um índice que não existe (fora dos limites da matriz) resultará em um erro `IndexError`.
- **Exemplo Prático:** Para uma matriz de placar de jogo, onde as linhas representam jogadores e as colunas representam pontuações em diferentes fases:
- Python

```
placar = [  
    [150, 200, 180], # Jogador 1  
    [170, 210, 190], # Jogador 2  
    [160, 195, 205] # Jogador 3  
]  
# Acessar a pontuação do Jogador 2 (linha 1) na Fase 3 (coluna 2)  
pontuacao_jogador2_fase3 = placar[1][2]  
print(pontuacao_jogador2_fase3) # Saída: 190
```

-
-
- **Vídeos Sugeridos:**
 - Listas dentro de Listas (Matrizes) em Python (Otávio Miranda):
https://www.youtube.com/watch?v=35-5H_B4a4Y

Referência do Slide: Slide 07 - Iteração e Modificação de Matrizes

- **Definição:** Iterar sobre uma matriz significa percorrer seus elementos, o que é comumente feito com laços de repetição (como `for`) para realizar operações em cada elemento. A modificação de um elemento ocorre após acessá-lo; seu valor pode ser alterado diretamente. Em muitos programas, o conteúdo da matriz precisa ser alterado dinamicamente durante a execução.
- **Aprofundamento/Complemento:** Para percorrer todos os elementos de uma matriz, são necessários laços aninhados. O laço externo itera sobre as linhas, e o laço interno itera sobre os elementos (colunas) de cada linha.
- **Exemplo Prático:** Imagine um mapa de um jogo representado por uma matriz de zeros, onde '1' representa a localização de um tesouro. Vamos colocar o tesouro e depois imprimir o mapa:
- Python

```
# Criando um mapa 5x5 preenchido com 0
mapa = [[0 for _ in range(5)] for _ in range(5)]
```

```
# Modificando um elemento para adicionar o tesouro na posição (linha 3, coluna 4)
mapa[3][4] = 1
```

```
# Iterando sobre a matriz para exibir o mapa
```

```
for linha in mapa:
```

```
    print(linha)
```

```
# Saída:
```

```
# [0, 0, 0, 0, 0]
```

```
# [0, 0, 0, 0, 0]
```

```
# [0, 0, 0, 0, 0]
```

```
# [0, 0, 0, 0, 1]
```

```
# [0, 0, 0, 0, 0]
```

```
•
```

```
•
```

- **Vídeos Sugeridos:**

- Laços Aninhados com Matrizes em Python (Curso em Vídeo):

https://www.youtube.com/watch?v=h2x1n_h_tMs

- Como Manipular Matrizes em Python com NumPy (Didática Tech):

<https://www.youtube.com/watch?v=pFR164y99dY>

Semana 28 - Aula 3

Tópico Principal da Aula: Vetores e Matrizes: Criação de programas: prática.

Subtítulo/Tema Específico: Depuração no Jogo "Encontre o Tesouro"

Código da aula: [SIS]ANO1C1B4S28A3

Objetivos da Aula:

- Compreender de forma prática os conceitos relacionados a vetores e matrizes.
- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.
- Migrar sistemas, implementando rotinas e estruturas de dados mais eficazes.

Recursos Adicionais:

- Caderno para anotações;
- Acesso ao laboratório e a computadores para resolução das atividades.

Exposição do Conteúdo:

Referência do Slide: Slide 06 - Depuração no jogo Encontre o Tesouro

- **Definição:** O slide apresenta um cenário de depuração (debugging) para o jogo "Encontre o Tesouro".³⁶ Um erro foi introduzido no código, fazendo com que o jogo encerre antes da hora ou não registre as tentativas do jogador corretamente, embora a lógica principal seja permitir múltiplas tentativas até encontrar o tesouro.³⁷
- **Aprofundamento/Complemento:** Depuração é o processo de encontrar e corrigir erros (bugs) em um software. Problemas em laços de repetição são muito comuns. Um erro como "encerrar prematuramente" em um jogo de tentativas geralmente está ligado à condição de parada do laço (`while` ou `for`). Pode ser que a variável de controle não esteja sendo atualizada corretamente, ou a condição lógica para continuar o jogo esteja sendo avaliada de forma errada. Não registrar tentativas pode indicar que o contador de tentativas está dentro de um escopo incorreto ou não está sendo incrementado a cada palpite.
- **Exemplo Prático:** Vamos analisar um código com o bug descrito.
- Python

```
# Código com BUG
tesouro_encontrado = False
tentativas = 0
posicao_tesouro = (2, 2)

while not tesouro_encontrado:
    print(f"Tentativa #{tentativas + 1}")
    linha = int(input("Digite a linha: "))
    coluna = int(input("Digite a coluna: "))

    if (linha, coluna) == posicao_tesouro:
        print("Você encontrou o tesouro!")
        tesouro_encontrado = True
    else:
        print("Tente novamente.")
        # BUG: o jogo encerra aqui, pois o laço termina.
        tesouro_encontrado = True # Esta linha está errada!
```

tentativas += 1

-
- **Correção:** O erro é a linha `tesouro_encontrado = True` no bloco `else`. Ela faz o laço parar mesmo que o jogador erre. A correção é simplesmente remover essa linha do `else`.
- **Vídeos Sugeridos:**
 - Depuração (Debugging) em Python no VS Code (Otávio Miranda): https://www.youtube.com/watch?v=big_a40s434
 - Entendendo Erros Comuns em Python (CodeWithMosh): <https://www.youtube.com/watch?v=6y64215C5v8>

Semana 28 - Aula 4

Tópico Principal da Aula: Vetores e Matrizes: Criação de programas: prática.

Subtítulo/Tema Específico: Jogo Caça ao Zero

Código da aula: [SIS]ANO1C1B4S28A4

Objetivos da Aula:

- Compreender de forma prática os conceitos de vetores e matrizes.
- Desenvolver sistemas computacionais utilizando ambiente de desenvolvimento.
- Resolver problemas computacionais com estratégias criativas.

Recursos Adicionais:

- Caderno para anotações;
- Acesso ao laboratório e a computadores para resolução das atividades.

Exposição do Conteúdo:

Referência do Slide: Slide 06/07/08 - Jogo Caça ao Zero

- **Definição:** O "Caça ao Zero" é um jogo simples onde o jogador precisa adivinhar a posição de um único zero escondido em uma matriz (o slide sugere ou) preenchida com o número 1. O jogo utiliza conceitos de:
 - **Representação de matriz:** Uma matriz bidimensional é usada para o tabuleiro.
 - **Acesso a elementos:** O jogador fornece coordenadas (linha e coluna) para verificar se encontrou o zero.
 - **Iteração:** Para mostrar o tabuleiro ao jogador, o programa percorre (itera sobre) a matriz.
- **Aprofundamento/Complemento:** A criação do jogo envolve alguns passos lógicos:
 - **Inicialização:** Criar uma matriz e preenchê-la completamente com o número 1.

- **Posicionamento do Zero:** Escolher uma posição aleatória (linha e coluna) e modificar o elemento naquela posição para 0.
- Loop do Jogo: Iniciar um laço que continue até o jogador acertar. Dentro do laço:
 - a. Pedir ao jogador para inserir uma linha e uma coluna.
 - b. Acessar o elemento da matriz na posição fornecida.
 - c. Verificar se o elemento é 0. Se for, o jogador vence e o laço termina. Se não, informa que ele errou e o laço continua.
- **Exemplo Prático:** Um trecho do código em Python para a lógica do jogo.
- Python

```
import random
```

```
# 1. Inicialização da matriz 3x3 com o número 1
tabuleiro = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
```

```
# 2. Posicionamento do Zero em um local aleatório
linha_zero = random.randint(0, 2)
coluna_zero = random.randint(0, 2)
tabuleiro[linha_zero][coluna_zero] = 0
```

```
# 3. Loop do Jogo
acertou = False
while not acertou:
    print("Tente adivinhar onde está o ZERO!")
    linha = int(input("Digite a linha (0 a 2): "))
    coluna = int(input("Digite a coluna (0 a 2): "))
```

```
if tabuleiro[linha][coluna] == 0:
    print("Parabéns, você encontrou o ZERO!")
    acertou = True
else:
    print("Errado! Tente novamente.")
```

-
-

- **Vídeos Sugeridos:**

- Gerando Números Aleatórios em Python (Curso em Vídeo): https://www.youtube.com/watch?v=kchC5KLZf_0
- Como Criar um Jogo de Batalha Naval em Python (Tech With Tim): <https://www.youtube.com/watch?v=tF1WimxSP7c> (A lógica é muito similar à do Caça ao Zero)