

Atividade Avaliativa 02

Programação de Computadores

- Felipe Natali Amaral – 24792566
- José Vitor Rodrigues Nucci – 24024423
- Lucas Leal Ibrahim - 24014012
- Pedro Pimentel Rodrigues – 24023362

Proposta:

- Este projeto tem como objetivo de criar um jogo da vida usando uma matriz 10X10, em que cada célula pode estar viva (1) ou morta (0), e sua evolução segue regras baseadas na quantidade de vizinhos vivos. O programa deve inicializar o tabuleiro com células vivas e mortas distribuídas aleatoriamente, simular um número de gerações definido pelo usuário e exibir o tabuleiro após cada iteração, permitindo acompanhar a transformação da população ao longo do tempo.

- **Informações adicionais:**

- Uma célula viva continua viva se tiver 2 ou 3 vizinhos vivos. Caso contrário, morre por solidão ou superpopulação.
- Uma célula morta torna-se viva se tiver exatamente 3 vizinhos vivos.

Apresentação do projeto:

- **Construção do tabuleiro:**

- Para a aleatoriedade foi utilizado o “srand(time(NULL))” que pega o número de segundos do momento que foi iniciado o programa e após isso pegamos esse número e fizemos variar de somente até 10 com o comando “rand() %10” e por fim pegamos o resto desse número final por 2 e comparamos se ele é par ou ímpar. No nosso caso quando for par irá imprimir 0 e quando for ímpar irá imprimir 1.

- **Construção do projeto:**

- No projeto 2 utilizamos somente a biblioteca “<time.h>” que usada para pegar o número de segundos desde 1970 e poder utilizar o “srand” e o “rand”. Além disso nós utilizamos funções int para fazer a lógica das regras e void para imprimir as matrizes.

- **Regras:**

- Para a construir as regras do jogo no projeto foi utilizado uma função int que recebe a matriz e os valores de sua coluna e linha (j e i) para assim saber a posição da célula que vamos analisar. Após isso criamos variáveis novas para colunas e linhas para assim não mexer nas variáveis i e j originais. Com isso utilizamos 2 for para que percorresse todos os vizinhos e quando percorresse a

própria célula que está sendo analisada o programa não soma faz nada. Assim quando um vizinho é 1 ele soma em uma variável vizinhos e assim até percorrer todos eles. Por fim ele volta para o main e vai comparar o número de vizinhos para assim saber o que vai acontecer com a célula analisada.

Referências bibliográficas:

- Aula 09 matrizes
- Aula 12 modularização