



Universidade Federal de Viçosa – Campus Florestal
Projeto e análise de algoritmos
Vitor Hugo de Oliveira – 3049

Documentação- Trabalho pratico 0

PROGRAMA GERADOR DE OBRA DE ARTE



Universidade Federal de Viçosa – Campus Florestal
Projeto e análise de algoritmos
Vitor Hugo de Oliveira – 3049

- Sumário

- 1- Introdução
- 2- Desenvolvimento
- 3- Conclusão



1- Introdução

A proposta do TP0 era fazer um programa, do qual esse deveria ter opções de figuras formadas por asteriscos, para assim preencher de forma aleatória um quadro de 80x20.

As figuras escolhidas foram um simples asterisco, soma, X e tri force.

```
PROGRAMA GERADOR DE OBRA DE ARTE
-----
1 - simples: *
2 - soma: *
          ***
          *
3 - X:  *  *
        *
        *  *
4 - Figuras aleatorias
5 - TriForce:  *
               ***
               *  *
               *** **
Digite o tipo de figura desejada: _
```


1- Figuras.

Após a escolha da figura desejada, é possível selecionar a quantidade desejada dessa figura a aparecer de forma aleatória no quadrado.



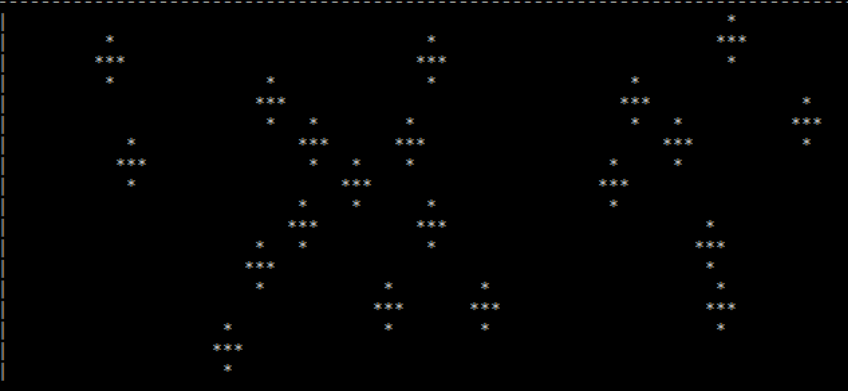
Universidade Federal de Viçosa – Campus Florestal
Projeto e análise de algoritmos
Vitor Hugo de Oliveira – 3049

```
4 - Figuras aleatorias
5 - TriForce:  *
               ***
               *  *
               *** **
Digite o tipo de figura desejada: 1
Digite qual será a quantidade a desenhar (entre 1 e 100): 10
```

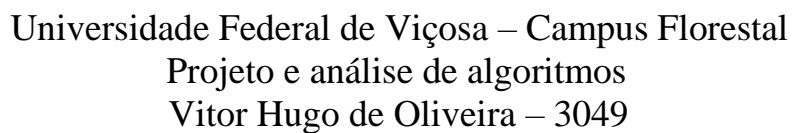


2- Obra de arte opção 1, 10 figuras.

```
4 - Figuras aleatorias
5 - TriForce:  *
               ***
               *  *
               *** **
Digite o tipo de figura desejada: 2
Digite qual será a quantidade a desenhar (entre 1 e 100): 20
```

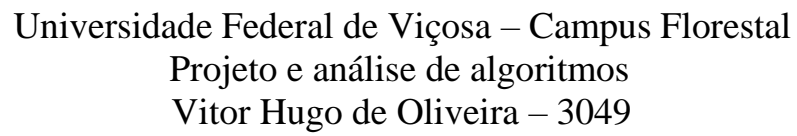


3- Obra de arte opção 2, 20 figuras.



4- Obra de arte opção 3, 50 figuras.

5- Obra de arte opção 4, 70 figuras.



6- Obra de arte opção 5, 100 figuras.

7- Opções ao final do programa.



2 – Desenvolvimento

O trabalho foi desenvolvido usando a linguagem C e seus artifícios, foram importado 2 bibliotecas a mais que o convencional para melhor construção e apresentação do programa, sendo estas `time.h`, possibilitando a criação de um seed para função `rand` (presente na biblioteca `stdlib.h`), para que a mesma no momento de gerar um número aleatório pegue como parâmetro o tempo do programa, possibilitando assim uma maior aleatoriedade e sem repetições após o rest do programa. Também, fora usado a `locale.h`, afim de importar o português para o console, de forma que o programa esteja de forma mais apresentável ao usuário.

Outro ponto importante de se notar no desenvolvimento foi a matriz 80x20, esta que comporta a moldura do quadro e as diferentes figuras e suas combinações. Foi necessário adaptar o espaço ocupado por cada figura dentro da matriz, para que assim esta não tome espaço da moldura do quadro.

Além disso, foram criadas 2 funções `comparacao` e `tricomparacao` em linhas gerais ambas as funções possuem o trabalho de retornar verdadeiro ou false, caso perceba a colisão de figuras dentro) um mesmo quadro, outro ponto em comum foi a criação de uma flag de 1501 (pontos livres na matriz) para evitar que o while que faz as comparações a procura de novos pontos vazios entre em um loop em finito devido o total preenchimento dos pontos disponíveis.

```
int comparacao(int linha, int coluna, char matriz[][80]) { // Faz comparacao para não haver sobreposição nas opções 2,3 e 4
    return (matriz[linha][coluna] == '*' || matriz[linha - 1][coluna + 1] == '*' ||
            matriz[linha + 1][coluna - 1] == '*' || matriz[linha - 1][coluna - 1] == '*' ||
            matriz[linha + 1][coluna + 1] == '*' || matriz[linha - 1][coluna] == '*' ||
            matriz[linha][coluna - 1] == '*' || matriz[linha][coluna + 1] == '*' ||
            matriz[linha + 1][coluna] == '*');
}

int tricomparacao(int linha, int coluna, char matriz[][80]) { // Faz comparacao para não haver sobreposição na opção 5
    return (
        matriz[linha - 2][coluna - 3] == '*' || matriz[linha - 2][coluna - 2] == '*' ||
        matriz[linha - 2][coluna - 1] == '*' || matriz[linha - 2][coluna] == '*' ||
        matriz[linha - 2][coluna + 1] == '*' || matriz[linha - 2][coluna + 2] == '*' ||
        matriz[linha - 2][coluna + 3] == '*' ||

        matriz[linha - 1][coluna - 3] == '*' || matriz[linha - 1][coluna - 2] == '*' ||
        matriz[linha - 1][coluna - 1] == '*' || matriz[linha - 1][coluna] == '*' ||
        matriz[linha - 1][coluna + 1] == '*' || matriz[linha - 1][coluna + 2] == '*' ||
        matriz[linha - 1][coluna + 3] == '*' ||

        matriz[linha][coluna - 3] == '*' || matriz[linha][coluna - 2] == '*' || matriz[linha][coluna - 1] == '*' ||
        matriz[linha][coluna] == '*' || matriz[linha][coluna + 1] == '*' || matriz[linha][coluna + 2] == '*' ||
        matriz[linha][coluna + 3] == '*' ||

        matriz[linha + 1][coluna - 3] == '*' || matriz[linha + 1][coluna - 2] == '*' ||
        matriz[linha + 1][coluna - 1] == '*' || matriz[linha + 1][coluna] == '*' ||
        matriz[linha + 1][coluna + 1] == '*' || matriz[linha + 1][coluna + 2] == '*' ||
        matriz[linha + 1][coluna + 3] == '*');
}
```



3 – Conclusão

Durante o trabalho pratico houve um grande ganho de logica e de conhecimento da linguagem. Sendo intrigante a forma como houve o trabalho com a matriz para gerar a moldura e, principalmente, as figuras criando um leque de diversas possibilidades de novas figuras.

Além disso, sobre a linguagem obtive o conhecimento da função rand e suas utilizações sendo muito útil para diversas aplicações cotidianas. Por fim, outro conhecimento que extrair desse TP, foi como passar matriz como parâmetro para uma função.