



UNIVERSIDADE FEDERAL DE VIÇOSA
Campus FLORESTAL
DISCIPLINA: CCF 330 - PROJETO E ANÁLISE DE ALGORITMOS
PROFESSOR: DANIEL MENDES BARBOSA
GUILHERME FERNANDES CASTRO DE OLIVEIRA - 3398

Gerador de Obras de Arte NFT

Sumário

1	Introdução	2
2	Desenvolvimento	4
3	Conclusão	6
4	Referências	6

1 Introdução

Esta documentação é referente ao Trabalho Prático 0 da disciplina CCF 330 - Projeto e Análise de Algoritmo. O trabalho consiste em criar um gerador de obras de arte totalmente randômico, por esse motivo, denominado de Gerador de Obras de Artes NFT. O objetivo deste trabalho também é relembrar os conceitos de programação em C, para assim realizar os próximos trabalhos práticos da Disciplina.

Para atacar o problema principal, pensei na obra de arte como uma matriz em \mathbb{R}^2 e assim preenchendo toda ela com espaço menos as bordas, e na hora de construir alguma figura, verificar se a coordenada tem algo preenchido.

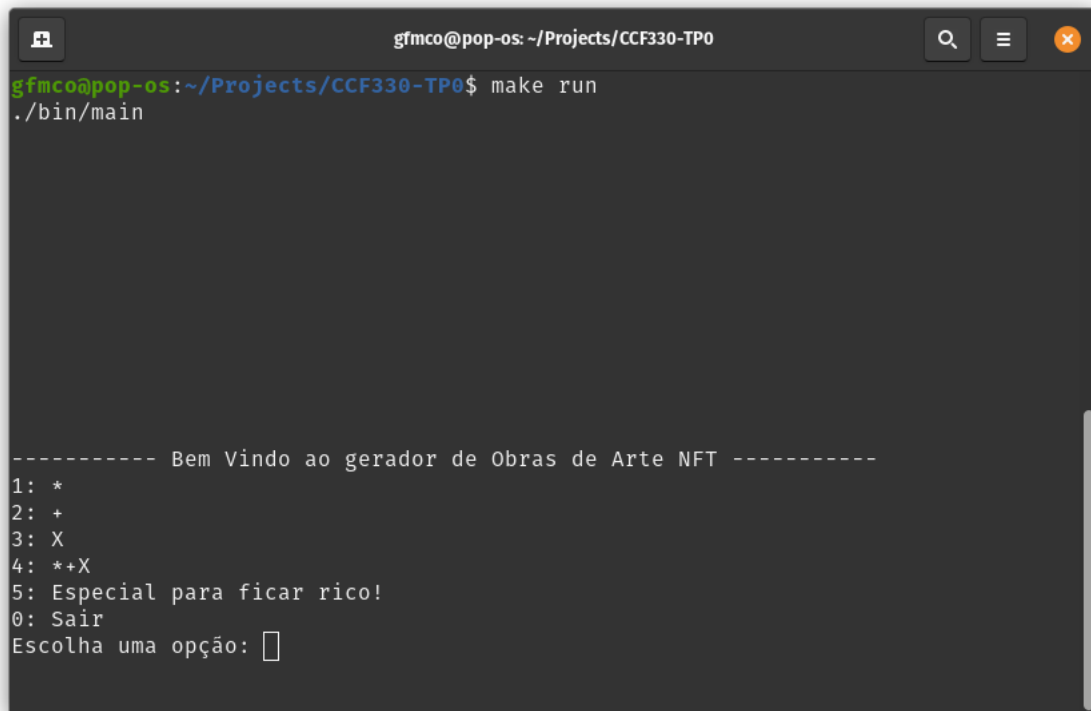
Para utilizar o programa, é necessário ter o Make, acessando a pasta via terminal, para rodar o programa é só utilizar os comandos

```
make clear
```

```
make
```

```
make run
```

Dessa forma irá cair no Menu principal que pode ser visto na imagem abaixo.



```
gfmco@pop-os: ~/Projects/CCF330-TP0
gfmco@pop-os:~/Projects/CCF330-TP0$ make run
./bin/main

----- Bem Vindo ao gerador de Obras de Arte NFT -----
1: *
2: +
3: X
4: **X
5: Especial para ficar rico!
0: Sair
Escolha uma opção: 
```

As figuras possíveis são:

- 1: Asterisco Simples
- 2: Sinal de adição feito com asteriscos
- 3: X feito com asteriscos
- 4: Aleatório com 1, 2 e 3
- 5: Especial, inventado por mim

O Especial consiste em, ele irá começar de uma coordenada, e depois decidir aleatoriamente para uma direção e irá continuar a figura com limite máximo de tamanho 10 ou até colidir com algo sem ser um espaço.

Após a escolha da opção, ele irá perguntar a quantidade de figuras que irá conter na obra de arte.

```
gfmco@pop-os: ~/Projects/CCF330-TP0
gfmco@pop-os:~/Projects/CCF330-TP0$ make run
./bin/main

----- Bem Vindo ao gerador de Obras de Arte NFT -----
1: *
2: +
3: X
4: **X
5: Especial para ficar rico!
0: Sair
Escolha uma opção: 5
Informe o número de símbolos: 
```

Após a escolha do número de figuras, ele irá exibir a obra de arte pedindo confirmação para continuar. Esta figura abaixo foi feita com a opção número 5.

```
gfmco@pop-os: ~/Projects/CCF330-TP0

----- Bem Vindo ao gerador de Obras de Arte NFT -----
1: *
2: +
3: X
4: **X
5: Especial para ficar rico!
0: Sair
Escolha uma opção: 5
Informe o número de símbolos: 5

Pressione uma tecla para continuar!

```

Após pressionar uma tecla para continuar, ele irá perguntar se deseja fazer uma nova obra de arte com

as mesmas propriedades, caso sim, ele irá exibir uma nova obra de arte, caso não, o programa retornará ao menu.

2 Desenvolvimento

O primeiro passo foi criar a função que construiria a matriz para a geração da obra de arte.

```
Frame *newFrame() {
    Frame* f = (Frame*)malloc(sizeof(Frame));
    for (int c = 0; c < LENGTH_C; c++) {
        f->frame[0][c] = '-';
        f->frame[LENGTH_L-1][c] = '-';
    }
    for (int l = 1; l < LENGTH_L-1; l++) {
        f->frame[l][0] = '|';
        f->frame[l][LENGTH_C-1] = '|';
    }
    for (int l = 1; l < LENGTH_L-1; l++) {
        for (int c = 1; c < LENGTH_C-1; c++) {
            f->frame[l][c] = ' ';
        }
    }
    return f;
}
```

Com a matriz criada, a próxima função foi para exibir a obra de arte, para verificar se tudo estava funcionando corretamente. Uma das principais funções utilizadas foi a de selecionar as coordenadas aleatoriamente, pois ela foi utilizada em todos os casos.

```
void getCord(int cord[2]) {
    cord[0] = 0;
    cord[1] = 0;
    while (cord[0] == 0) cord[0] = rand() % LENGTH_L;
    while (cord[1] == 0) cord[1] = rand() % LENGTH_C;
}
```

Então partindo para o caso mais básico que seria a do asterisco simples como figura, facilitou para entender os próximos casos que são bem parecidos.

```

void asterisco() {
    int n = getN();
    int cord[2];
    Frame *f;
    do
    {
        f = newFrame();
        for (int i = 0; i < n; i++) {
            getCord(cord);
            while (f->frame[cord[0]][cord[1]] != ' ') getCord(cord);
            f->frame[cord[0]][cord[1]] = '*';
        }
        printFrame(f);
    } while (continuar());
}

```

Após feito os casos 1, 2, 3 e 4, fiz o menu para poder os diferentes casos de forma mais prática. Tudo funcionando perfeitamente, partir para o caso principal 5, que precisava de criatividade e pensar o que seria melhor.

Primeiramente este caso irá escolher um número aleatório de 0 a 7, que vão representar as 8 direções possíveis, com esse valor ele entrará no bloco do switch e irá adicionar a figura na coordenada inicial até a condição de parada.

```

void especial() {
    int n = getN();
    int cord[2];
    Frame *f;
    int aux, cont;
    do
    {
        f = newFrame();
        for (int i = 0; i < n; i++) {
            getCord(cord);
            cont = 0;
            while (f->frame[cord[0]][cord[1]] != ' ') getCord(cord);
            // f->frame[cord[0]][cord[1]] = '*';
            aux = rand() % 8;
            switch (aux)
            {
                case 0:
                    while (f->frame[cord[0]+cont][cord[1]+cont] == ' ' && cont < 11) {
                        f->frame[cord[0]+cont][cord[1]+cont] = '\\';
                        cont++;
                    }
                    break;
                case 1:
                    while (f->frame[cord[0]-cont][cord[1]+cont] == ' ' && cont < 11) {
                        f->frame[cord[0]-cont][cord[1]+cont] = '/';
                        cont++;
                    }
                    break;
                case 2:
                    while (f->frame[cord[0]-cont][cord[1]-cont] == ' ' && cont < 11) {
                        f->frame[cord[0]-cont][cord[1]-cont] = '\\';
                        cont++;
                    }
                    break;
                case 3:
                    while (f->frame[cord[0]+cont][cord[1]-cont] == ' ' && cont < 11) {
                        f->frame[cord[0]+cont][cord[1]-cont] = '/';
                        cont++;
                    }
                    break;
            }
        }
    } while (1);
}

```

3 Conclusão

Com este trabalho prático foi possível relembrar conceitos básicos de C, como trabalhar com *strings*, com números randômicos, ponteiros, etc. Sendo assim, muito importante para o decorrer da disciplina, além de ter sido muito divertido imaginar e criar Obras de Artes únicas.

4 Referências

<https://www.cprogressivo.net/2012/11/A-funcao-printf-Caracteres-Especais.html> <https://www.cprogressivo.net/2013/03/Como-gerar-numeros-aleatorios-em-C-com-a-rand-srand-e-seed.html>