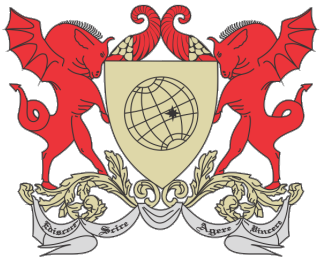

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS FLORESTAL

Trabalho Prático 0
Projeto e Análise de Algoritmos

Artur Papa - 3886

Trabalho Prático apresentado à disciplina de CCF 330 - Projeto e Análise de Algoritmos do curso de Ciência da Computação da Universidade Federal de Viçosa.

Florestal
Setembro de 2022

CCF 330 - Projeto e Análise de Algoritmos

Trabalho Prático 0

Artur Papa - 3886

30 de Setembro de 2022

Contents

1 Introdução

O Grito é a obra-prima do pintor norueguês Edvard Munch. Pintada pela primeira vez em 1893, a tela foi ganhando três novas versões com o passar do tempo. As obras de Munch são classificadas como precursoras do **expressionismo** (um importante movimento modernista da primeira parte do século XX).[1] Apesar de ainda existirem obras de arte pintadas a mão como antigamente, vemos hoje em dia um grande uso da tecnologia nesta área, seja com o uso de Inteligência Artificial para fazer quadros, ou de softwares para facilitar o desenho ou a modelagem a ser feita.

Dessa maneira, foi proposto neste trabalho, embora que de maneira lúdica, o uso de um algoritmo para gerar obras de arte, vale ressaltar que foi feito o uso apenas de caracteres da tabela ASCII e que não há grande riqueza de detalhes nas obras, se tratando apenas uma forma de "brincar" com o algoritmo gerando figuras randômicas em um quadro.

2 Desenvolvimento

2.1 Criando o quadro vazio

Primeiramente, temos que gerar um frame 20x80 que irá conter as imagens a serem criadas posteriormente, desta forma, teríamos o quadro representado na figura.

Figura 1: Frame vazio

Vale citar que essa criação do quadro foi feita através de uma função de inicialização que apenas gera as bordas do devido frame e o preenche com um ' ' em cada linha que seja diferente da borda.

2.2 Lógica para geração aleatória

Para a criação de figuras em posições aleatórias do quadro foi utilizada a mesma a lógica para todas as figuras. Primeiro, temos um ***for*** com uma quantidade **n** de iterações, que representa a quantidade de figuras que serão geradas dentro do frame, já no escopo do ***for*** temos um ***while*** em que será feita a escolha da posição das figuras na área desejada, sendo que é escolhido um número aleatório para linha e coluna, já que estamos tratando de uma matriz. Além disso, dentro deste ***while*** fazemos a verificação para conferir se não há nenhum conflito de espaço.

```c
void asterisk(char **frame, int n)
{
    int x;
    int y;

    for (int i = 0; i < n; i++)
    {
        while (1)
        {
            x = (rand() % ((LINE - 2))) + 1;
            y = (rand() % ((COL - 2))) + 1;

            if (frame[x][y] == ' ')
                break;
        }

        frame[x][y] = '*';
    }
}
```

Figura 2: Função para gerar um asterisco simples

Outrossim, vale fazer uma observação para os valores das posições a serem escolhidas (x e y), temos que as figuras a serem geradas não podem ultrapassar ou ficar no mesmo espaço que as bordas, desta forma, nosso valor gerado será limitado a **(QTDLINHAS - 2)** e **(QTDCOLUNAS - 2)** sendo que em ambos os casos será limitado que o valor a ser gerado comece de um pois zero ocuparia uma borda do quadro. Assim, em cada figura foi analisado o seu tamanho e qual o seu centro, pois caso isto não seja analisado, poderia ocorrer de termos um valor sendo gerado que ultrapassasse os limites do quadro, o que acabaria gerando um erro no código.

2.3 Matriz para armazenar as figuras

Para o armazenamento das figuras que foram geradas, foi feito uso de um ponteiro para **char**, assim, seria mais fácil para definir as posições em que as figuras seriam armazenadas de maneira a formar o desenho proposto.

```c
frame[x][y] = '*';
frame[x + 1][y] = '*';
frame[x - 1][y] = '*';
frame[x][y + 1] = '*';
frame[x][y - 1] = '*';
```

Figura 3: Posições para o símbolo de soma simples com asteriscos

```c
frame[x][y] = '*';
frame[x - 1][y - 1] = '*';
frame[x - 1][y + 1] = '*';
frame[x + 1][y - 1] = '*';
frame[x + 1][y + 1] = '*';
```

Figura 4: Posições para a letra X com asteriscos

2.4 Geração de figuras aleatórias

A priori, para a geração de figuras aleatórias foi utilizada a mesma lógica citada anteriormente, entretanto foi criada uma terceira variável que teria seu valor variando de zero à dois, com cada valor representando uma figura. Assim, para cada número gerado seria feita a devida verificação de acordo com a obra para que não houvesse conflito de espaço assim como feito nos outros casos e, após ter sido feita essa verificação, a matriz de char iria receber os asteriscos nas devidas posições de acordo com o valor da variável aleatória.

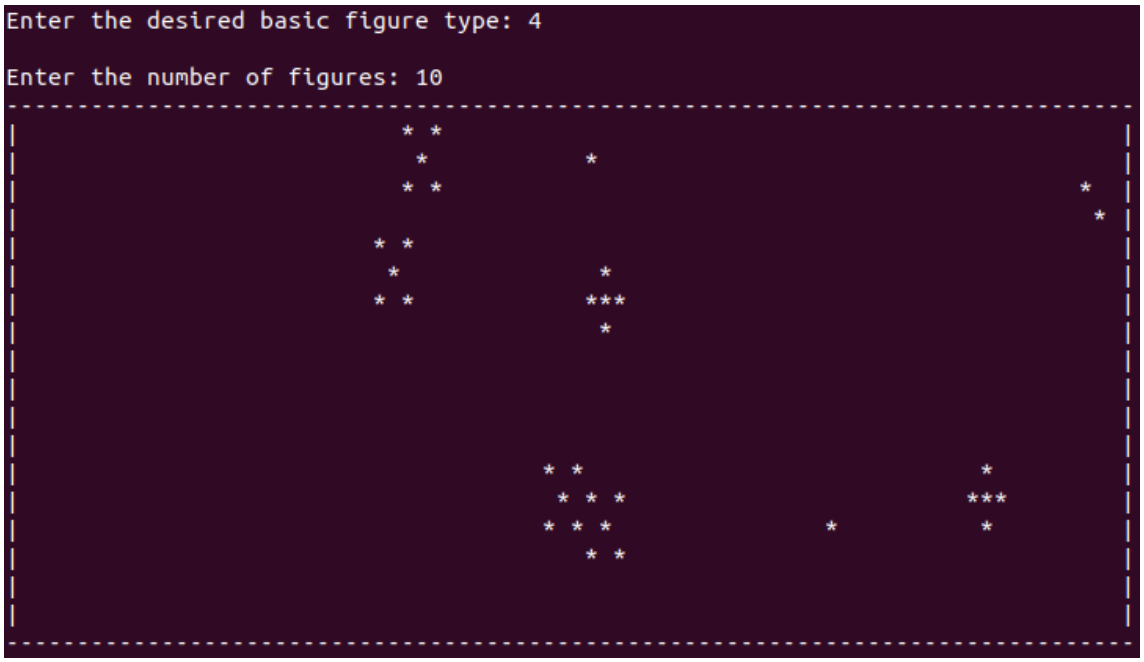

Figura 5: Figuras aleatórias

Obs.: As figuras geradas são apenas as propostas no trabalho prático, não incluindo assim a obra escolhida pelo aluno.

2.5 Obra de arte específica

Para a obra de arte específica foi feita a escolha de se fazer o jogador de basquete Michael Jordan e sua famosa logo da marca **Jordan** com caracteres da tabela ASCII, sendo que a obra foi feita tanto na posição original quanto com a figura na posição invertida.

Figura 6: Jumpman Logo

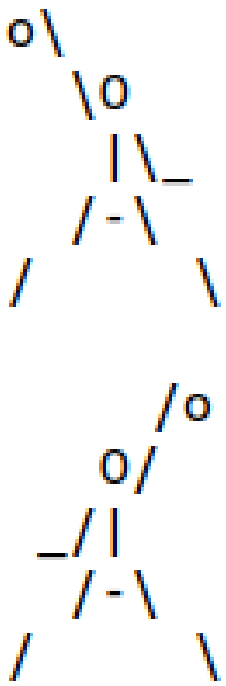

Figura 7: Jordan com caracteres ASCII

Além disso, vale ressaltar que por ser uma obra um pouco maior que as outras, a quantidade a ser gerada está limitada a cinquenta figuras por execução, assim, caso um valor maior que cinquenta seja digitado serão considerados apenas este limite.

Outrossim, tanto para esta, mas como para as outras obras também, caso seja digitado um valor menor ou igual à zero, serão geradas quantidades aleatórias de figuras dentro do limite estabelecido.

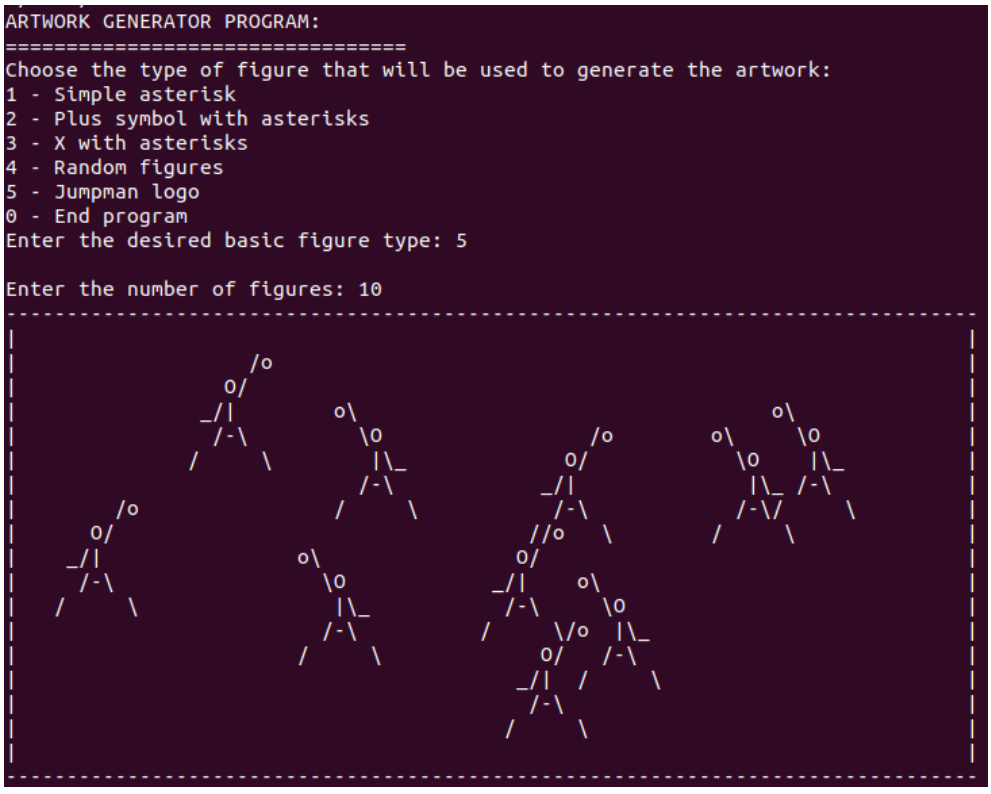

Figura 8: Menu e execução com o Jordan

2.6 Execução do trabalho

Para a execução do caminho de dados foi feita a escolha de adicionar um arquivo makefile. Assim, para compilar o programa basta realizar no terminal o comando ***make*** e, para executar usa-se o comando ***make run***.

3 Conclusão

Posto isso, conclui-se que o trabalho em questão foi desenvolvido conforme o esperado, atingindo as especificações requeridas na descrição do mesmo em implementar um programa para criar obras de arte aleatórias.

Por conseguinte pode-se ressaltar que o material apresentado na disciplina foi de suma importância para o desenvolvimento do projeto, haja vista que as explicações dadas pelo professor ajudou bastante a entender as etapas a serem executadas e na construção dos códigos.

Em adição, é válido dizer que apesar das dificuldades na implementação do código, o aluno em questão foi capaz de superar e corrigir quaisquer erros no desenvolvimento do algoritmo. Por fim, verificou-se a assertiva para o objetivo do projeto implementar um programa para criar obras de arte aleatórias, além de uma obra de arte própria feita à mercê do autor do projeto.

References

[1] O grito. https://www.culturagenial.com/quadro-o-grito-de-edvard-munch/. (Accessed on 30/09/2022).