

UNIVERSIDADE FEDERAL DE VIÇOSA
CAMPUS FLORESTAL

SAMUEL SENA - 3494

TRABALHO 0

FLORESTAL
2019

Sumário

Sumário2

Introdução3

Tipos de Artes7

Desenvolvimento 11

Conclusão 14

Introdução

O trabalho apresentado a seguir entrega um algoritmo executado em modo texto, com um menu interativo pelo teclado. O algoritmo é capaz de gerar 5 tipos de quadros composto por figuras despostas de maneira completamente aleatória.

Inicialmente para se executar o programa, é necessário realizar a compilação do código fonte em C. Para isso, em algum terminal Linux execute o “*makefile*” da seguinte forma:

Para compilar:

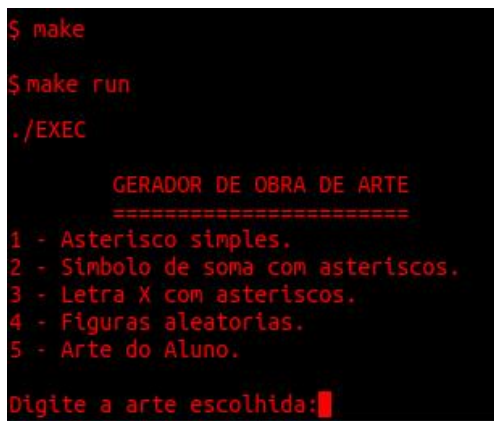
`$ make`

E para executar:

`$ make run`

A figura abaixo exemplifica o processo de compilação e execução pelo Linux:

Figura 1



```
$ make
$ make run
./EXEC

          GERADOR DE OBRA DE ARTE
          =====
1 - Asterisco simples.
2 - Simbolo de soma com asteriscos.
3 - Letra X com asteriscos.
4 - Figuras aleatorias.
5 - Arte do Aluno.

Digite a arte escolhida:
```

Fonte: Terminal Linux

Ou no caso de execução em um sistema Windows, execute o seguinte código no Prompt de comando devidamente navegado até a pasta contendo o arquivo “main.c”:

`> gcc main.c -o EXEC Sources/Gerador_Quadro.c`

E em seguida abra o executável gerado.

O programa inicialmente exibirá um menu com as seguintes opções:

Figura 2

```
./EXEC

      GERADOR DE OBRA DE ARTE
      =====
1 - Asterisco simples.
2 - Simbolo de soma com asteriscos.
3 - Letra X com asteriscos.
4 - Figuras aleatorias.
5 - Arte do Aluno.

Digite a arte escolhida:█
```

Fonte: Terminal Linux

Primeiramente, a escolha do tipo de arte deverá ser realizada (números de 1 a 5, caso maior que 5, automaticamente a opção 5 será selecionada). Em seguida a quantidade de elementos que deveram ser combinados na arte desejada será requisitada (valores entre 0 e 100 devem ser escolhidos preferencialmente. Caso maior que 100, este automaticamente este será convertido para 100. Caso menor ou igual a 0, automaticamente um valor aleatório de 1 a 100 será computado). Ao final da exibição do quadro, será perguntado se deseja visualizar um novo quadro com as mesmas configurações selecionadas anteriormente (0 para não, 1 para sim). A figura abaixo representa bem as situações:

[illegible]

Caso em algum momento, a opção 0 seja selecionada, outra pergunta será realizada pelo programa, porem agora, ela se refere a retornar ao menu principal para poder realizar uma nova configuração de uma arte, ou se deseja finalizar o programa. Como ilustrado a seguir nas figuras abaixo:

Figura 4

```
Deseja fazer outro quadro com o mesmo valor?
    0-Nao
    1-Sim
Entre:0

Deseja configurar uma nova arte?
    0-Nao
    1-Sim
Entre:1

        GERADOR DE OBRA DE ARTE
        =====
1 - Asterisco simples.
2 - Simbolo de soma com asteriscos.
3 - Letra X com asteriscos.
4 - Figuras aleatorias.
5 - Arte do Aluno.

Digite a arte escolhida:█
```

Fonte: Terminal Linux

Figura 5

```
Deseja fazer outro quadro com o mesmo valor?
    0-Nao
    1-Sim
Entre:0

Deseja configurar uma nova arte?
    0-Nao
    1-Sim
Entre:0

        Finalizando...
        =====
        Desenvolvedor:
        Samuel Sena - 3494
        OBRIGADO POR UTILIZAR O PROGRAMA!
```

Fonte: Terminal Linux

Tipos de Artes

Dependendo da escolha durante a configuração da arte desejada a ser gerada, diferentes desenhos serão formados. Entre os elementos básicos que compõem as possíveis artes geradas, estão: Asterisco simples; Símbolo de soma feito de asteriscos; Letra X feito de asteriscos; Mistura aleatória dos 3 anteriores; Arte pessoal do Aluno. Para cada tipo de elemento selecionável citado, as figuras (6 a 10) abaixo ilustram quadros gerados com 15 elementos para fins de demonstração:

Figura 6

```
Digite a arte escolhida:1
Digite a quantidade de figuras (menor ou igual a zero para aleatorio):15

      OBRA GERADA:
+-----+
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
|               |
+-----+

Deseja fazer outro quadro com o mesmo valor?
0-Nao
1-Sim
Entre:█
```

Fonte: Terminal Linux

Figura 7



Fonte: Terminal Linux

Figura 8



Fontes: Terminal Linux

Figura 9

```
Digite a arte escolhida:4
Digite a quantidade de figuras (menor ou igual a zero para aleatorio):15

Quantidades de cada figura:
* : 6
+ : 5
X : 4

OBRA GERADA:
```

```
*
          *
      *   +   *
    *   *   *   +
*   *   *   *   *       *
    *   *   *   *       *
        *   ***      *
            *   *
        ***
            *
                *
          *   *   ***
        *   *   *
          *   *   *
            *   ***
                *
                    *   *
                      *
                        *   *   *
```

```
Deseja fazer outro quadro com o mesmo valor?
0-Nao
1-Sim
Entre: 
```

Figura 10

```
Digite a arte escolhida:5  
Digite a quantidade de figuras (menor ou igual a zero para aleatorio):15
```

OBRA GERADA:

```
      *  
    *   *   *  
  * S&S *   * S&S *  
 *   *   *   *  
 *       *           * S&S *   *   *   *  
          *   *   *   *   *   *  
            * S&S ** S&S *  
              *   *   *   *  
                *   *   *   *  
                  * S&S *   * S&S *  
                    *   *   *   *  
                      *   *   *   *  
                        * S&S *   * S&S *  
                          *   *   *   *
```

Deseja fazer outro quadro com o mesmo valor?

- 0-Nao
- 1-Sim

Entre:

Fonte: Terminal Linux

Se atentando a figura 10, a qual ilustra o quadro com a arte pessoal do aluno, é possível perceber que a composição central de um elemento individual conta com a disposição dos caracteres “S&S”. Estes são em referência ao primeiro e último nome do aluno. Já os contornos desse elemento, por outro lado, foram escolhidos desta maneira por possuírem certa capacidade de se misturarem bem em casos que valor da quantidade de elementos é elevado. O quadro configurado para ser gerado com a arte do aluno tem capacidade máxima de executar com 55 elementos (Caso o valor entrado seja superior, o valor será convertido para 55). Este limite foi estabelecido após testes exaustivos para garantir que o programa não entre em “loop” caso fique sem espaço de encaixar novos elementos na arte em questão. Como exemplificado na figura abaixo:

Figura 11

```

Digite a quantidade de figuras (menor ou igual a zero para aleatorio):100

Quantidade excede capacidade do quadro!! Alternando
para o limite seguro da quantidade da figura -> 55

OBRA GERADA:

-----
*      *      *      *      *      *      *      *      *      *
*  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
**S&S** * S&S** * * * * * **S&S * * * * *S&S*** * * * S&S**S&S*** S&S *
**S&S** * * * * * S&S*** S&S** * * * * *S&S*** * * * * *S&S * *
* * * * * * * * * * * * * * *S&S** * **S&S** * * *
** **S&S * * * * * *S&S * * * * * **S&S** * * * *
**S&S** * * * * *S&S * * * * *S&S** * * * * *S&S *
* * * * * * * **S&S** * * S&S** **S&S*** * * * S&S** * *
* S&S** * * * **S&S** * * * * * * * *S&S**S&S * * * * *S&S *
* * * S&S**S&S** * * * * * * * * * *S&S** * * * * *S&S *
* * * * * * *S&S * * * * * * * *S&S * *S&S * * *
* S&S*** ** S&S * * * * **S&S** * * * *S&S** * * * * *
* * **S&S * * * * * * * * * *S&S*** * * S&S** **S&S *
**S&S** * * * S&S**S&S** * S&S** * * * *S&S** *
* * * * * * * * * * * * * *
*      *      *      *
-----

Deseja fazer outro quadro com o mesmo valor?
0-Nao
1-Sim
Entre:

```

Fonte: Terminal Linux

Desenvolvimento

O desenvolvimento do programa se consistiu basicamente na criação de uma matriz de caracteres de dimensões 20x80. Em seguida, quando a matriz é inicializada, o preenchimento com as laterais do quadro também já é realizado. Desta forma, apenas um percurso pela matriz é efetuado. A função responsável por isso é ilustrada a seguir:

Figura 12

```
void inicia_quadro(char Quadro[20][80]){
    int i,j;           //Função que inicializa
    for(i=0;i<20;i++){
        for(j=0;j<80;j++){
            if(i==0 || i==19)
                Quadro[i][j] = '-';
            else if(j==0 || j==79)
                Quadro[i][j] = '|';
            else
                Quadro[i][j] = ' ';
        }
    }
}
```

Fonte: Gerador_Quadro.c

Mais à frente, após as devidas entradas desejadas serem computadas, o algoritmo realiza um loop de tamanho igual a entrada do usuário, sendo que para cada tipo de elemento, posições geradas aleatoriamente são testadas e enquanto não for encontrado uma respectiva posição vazia, o algoritmo continua a gerar novas coordenadas aleatórias. A figura a seguir demonstra o funcionamento em um dos casos de geração da arte:

Figura 13

```
void plota_X(int Quant,char Quadro[20][80]){ //Função para criar
    int i,x,y;
    if(Quant>100){ //Limitando quantidade de elementos.
        printf("\n\tQuantidade excede 100!! Alternando para 100.\n");
        Quant = 100;
    }
    for(i=0;i<Quant;i++){
        x = 2+(rand()%16);
        y = 2+(rand()%76); //Inicialmente as coordenadas aleatorias
        while((Quadro[x][y] != ' ') || (Quadro[x+1][y+1] != ' ') || (Quadro[x-1][y-1] != ' ')){
            x = 2+(rand()%16); //Caso a posição aleatoria gerada atinja uma posição ocupada
            y = 2+(rand()%76);
        }
        Quadro[x][y]='*'; //Devidas posições disponiveis sendo preenchidas
        Quadro[x+1][y+1]='*';
        Quadro[x-1][y-1]='*';
        Quadro[x-1][y+1]='*';
        Quadro[x+1][y-1]='*';
    }
}
```

Fonte: Gerador_Quadro.c

Ao final deste loop, a matriz é exibida na tela, com o uso da seguinte função:

Figura 14

```
void imprime_quadro(char Quadro[20][80]){
    int i,j; //Função para imprimir matriz com arte.
    printf("\n\t\t\t\t\tOBRA GERADA:\n");
    for(i=0;i<20;i++){
        printf("\n");
        for(j=0;j<80;j++){
            printf("%c",Quadro[i][j]);
        }
    }
}
```

Fonte: Gerador_Quadro.c

A opção de arte que gera um quadro com três elementos em quantidades aleatórias tem como principal mecanismo para se definir as quantidades de cada elemento da seguinte forma apresentada abaixo:

Figura 15

```
Q_Aster = (rand()%Quant); //Inicial
Q_Soma = (rand()%(Quant-Q_Aster));
Q_x = Quant - Q_Aster - Q_Soma; //
```

Fonte: Gerador_Quadro.c

Sendo que “Q_Aster” se refere a quantidade de asteriscos simples, “Q_Soma” se refere a quantidade de símbolos de soma, e “Q_x” se refere a quantidade de X.

O arquivo principal do programa “main.c” é responsável por organizar as chamadas de cada subprograma e possibilitar o loop do programa controlado pelo usuário. A figura abaixo ilustra uma pequena parte do código fonte do arquivo “main.c”:

Figura 16

```
int main(int argc, char const *argv[])
{
    char Quadro[20][80]; //Matriz de caracteres para Obra de Arte.
    srand(time(NULL));
    while(1){
        int a,b,d=3; //Variaveis auxiliares.
        menu();
        printf("\nDigite a arte escolhida:"); //Menu interativo.
        scanf("%d",&a);
        printf("\nDigite a quantidade de figuras (menor ou igual a zero para aleatorio:");
        scanf("%d",&b);
        if(b<=0){ //Verificando se entrada deverá ser aleatoria.
            b=1+(rand()%99);
            printf("\n\t -> Quantidade aleatoria gerada: %d <-\n",b);
        }
        if(a==1){ //Arte de um Asterisco.
            while(1){
                int c=3; //Variavel auxiliar
                inicia_quadro(Quadro); //Incializando matriz de quadro.
                plota_asterisco(b,Quadro); //Criando arte.
                imprime_quadro(Quadro); //Imprimindo arte gerada na tela.
                printf("\nDeseja fazer outro quadro com o mesmo valor?");
                printf("\n\t0-Nao");
                printf("\n\t1-Sim");
                while(c!=1 && c!=0){ //Garantindo que entrada de opcao é valida.
                    printf("\nEntre:");
                    scanf("%d",&c);
                    if (c!=1 && c!=0) {
                        printf("\n\tValor invalido!\n");
                    }
                }
                if(c == 0){ //Caso o usuario opte por nao, o programa descarta valores.
                    break;
                }
            }
        }
    }
}
```

Fonte: main.c

Vale lembrar que todo o código fonte encontra-se comentado para a fácil compreensão e entendimento do seu funcionamento.

Conclusão

Sem dúvidas, o desenvolvimento de uma ideia abstrata para o devido cumprimento do comportamento desejado do programa requisitado pela descrição do trabalho foi a maior empecilho em comparação com os demais. A decisão de se trabalhar com uma matriz de caracteres serviu como uma espécie de “gatilho” mental para fazer com que os demais problemas que surgissem em seguida fossem facilmente contornados. Portanto, após longos testes de execução e verificações no código fonte, o programa se encontra executando da maneira desejada. E por último, mas não menos importante, é notável a suma importância com relação a aprendizagem e aperfeiçoamento de conceitos em programação, não ficando limitado apenas a *syntax* da linguagem, mas abrangendo também ao pensamento lógico que possibilita a correta programação.

Agradecimentos ao professor Daniel Mendes pela oportunidade de realização do trabalho e dúvidas sanadas.

Todo o desenvolvimento e distribuição do trabalho encontra-se hospedado na seguinte página do [GitHub](#).