UNIVERSIDADE FEDERAL DE VIÇOSA – CAMPUS FLORESTAL

DANIEL FREITAS MARTINS – 2304

Documentação referente ao Trabalho Prático 0 — Aquecimento na Linguagem C — Gerador de Obra de Arte

Florestal

DANIEL FREITAS MARTINS – 2304

Documentação referente ao Trabalho Prático 0 – Aquecimento na Linguagem C – Gerador de Obra de Arte

Documentação apresentada à disciplina CCF 330F — Projeto e Análise de Algoritmos do curso de Bacharelado em Ciência da Computação da Universidade Federal de Viçosa — *Campus* Florestal.

Orientador: Daniel Mendes Barbosa

Florestal

2018

SUMÁRIO

1.	Ir	ntrodução	3
		Metodologia	
		Desenvolvimento	
		Obra de arte The Village	
	3.2.	Menu secundário	6
	3.3.	Principais resultados	
4.	C	Conclusão	10
5.	R	Referências bibliográficas	1

1. Introdução

O objetivo deste trabalho foi a criação de um programa capaz de gerar obras de arte aleatórias em um quadro delimitado por 20 linhas e 80 colunas, com bordas prédefinidas nas primeiras e últimas linhas ('-') e colunas ('|').

As figuras básicas solicitadas podem ser vistas na Figura 1 abaixo:

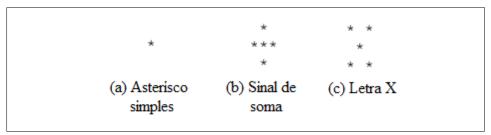


Figura 1: Figuras básicas disponíveis para gerar o quadro: (a) Asterisco simples; (b) Sinal de soma feito com asteriscos; (c) Letra X, feita com asteriscos.

O menu permite o usuário escolher qual tipo de figura básica será utilizada e a quantidade de figuras (máx. 100) a serem geradas em posições aleatórias. Ao informar um número negativo de figuras, um valor aleatório será gerado entre 1 e 100. O menu pode ser visto na Figura 2:

Figura 2: Menu principal. As cinco primeiras opções se referem ao que foi pedido; as demais foram implementações extras.

A opção 4 considera apenas as figuras das três primeiras opções para a aleatoriedade dos elementos. Para compensar a aleatoriedade com a obra The Village (opção 5), está implementada a opção 6.

As opções 5 e 7 do menu da Figura 2 serão explicadas na seção de Desenvolvimento.

2. Metodologia

Com a finalidade de simplificar as operações e ao mesmo tempo deixar o código mais flexível em relação a adição ou modificação de conteúdo, optei pela criação de dois TADs, sendo um relacionado ao quadro (ou tela) e o outro à figura de um elemento

a ser desenhado. Vale citar também a construção de uma estrutura para as coordenadas *x* e *y*, que foi de grande ajuda.

O *TADFigura* é responsável pelo desenho de uma figura qualquer em uma área previamente delimitada por constantes no respectivo arquivo de cabeçalho, e foram escolhidas de modo a formarem um espaço 7x7 do tipo *char* (Fig. 3). Este tamanho foi escolhido para conseguir comportar figuras maiores do que as figuras básicas (3x3), permitindo assim uma maior flexibilidade na escolha e montagem de desenhos. Apesar disso, foram fixados quatro tamanhos de figura: Pontual (1x1), Pequeno (3x3), Médio (5x5) e Grande (7x7). Estes tamanhos foram escolhidos desta forma por conveniência e para a utilização de simetria das figuras a serem modeladas, reduzindo assim o trabalho de calcular a validade de um desenho. Vale ressaltar que o desenho feito por este TAD não leva em consideração outras restrições. Em outras palavras, o problema de desenhar uma figura fica restrita a este TAD.

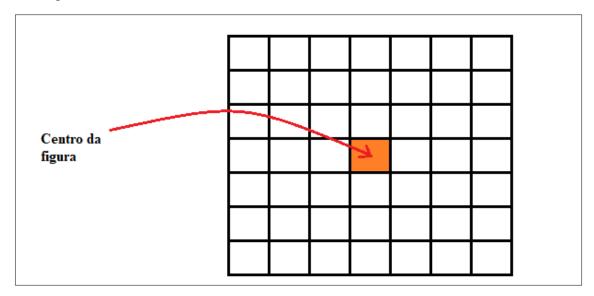


Figura 3: Representação da matriz de caracteres 7x7 de *TADFigura*. O centro da matriz, em destaque, é utilizado para tratar questões de simetria que simplificam problemas na consideração de uma figura válida (i.e. que não ultrapassam as bordas delimitadoras). Observe também que quando o tamanho da figura é Pontual, ela preenche apenas o centro da figura, em laranja.

O *TADQuadro* é responsável por posicionar uma figura na tela e realizar demais operações referentes a impressão e checagem de conflitos. Este posicionamento consiste de coordenadas aleatórias do centro da figura levando em consideração o tamanho da mesma. Para o tratamento de conflitos, são realizadas no máximo cem mil tentativas de reposicionamento; caso os conflitos não tenham sido resolvidos, o desenho é interrompido e é mostrado na tela em sequência.

Deste modo, veja que o *Main* fica restrito apenas a operações da interface do usuário. Apesar da linguagem *C* não ser orientada a objetos, tentei implementar uma forma de encapsulamento dos campos, facilitando assim a manutenção. Observe também que a criação de uma nova figura fica bastante facilitada devido a essa divisão de responsabilidades.

3. Desenvolvimento

3.1. Obra de arte The Village

A figura escolhida por mim havia sido um boneco, inicialmente. Entretanto, querendo explorar a capacidade da flexibilidade do código como um todo, implementei mais cinco figuras além da original e disponibilizei os três tipos de tamanhos para as figuras padrões (i.e. as figuras pedidas, exceto a figura pontual Asterisco Simples). A motivação para a criação da obra The Village foi devido aos jogos de estilo RPG, especificamente aqueles de Super Nintendo, como *Chrono Trigger*, *Final Fantasy* e *Secret of Mana*.

→ Observação: Para que se tenha o efeito inicialmente solicitado, com todos os itens cabendo na tela levando em consideração a quantidade máxima de itens gerados, é necessário selecionar o tamanho da figura como sendo Não Variável (veja abaixo a Figura 4). O tamanho padrão para todas as figuras a serem geradas será o tamanho Pequeno (3x3).

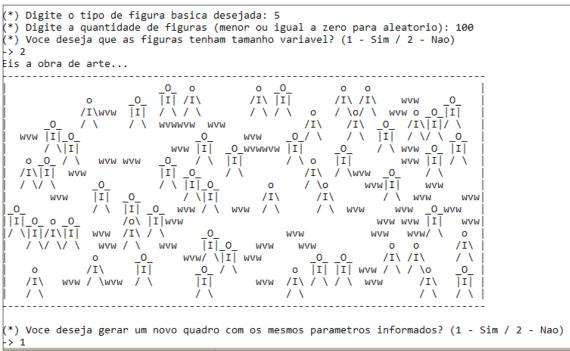


Figura 4: O tipo de figura básica selecionada foi a número 5 (The Village); a quantidade de figuras a serem desenhadas é de 100 itens; as figuras não possuem a condição de tamanho variável, sendo portanto fixadas como Pequeno (3x3). Na figura é possível observar três das seis figuras disponíveis para a obra The Village: dois tipos de bonecos, sendo um com ombreiras e o outro apenas um boneco comum, e uma grama, formada pelas letras 'w' e 'v'. Perceba que todas as 100 figuras couberam no quadro.

A figura 4 mostra que sempre que um quadro é mostrado, o usuário é perguntado se ele quer repetir os parâmetros que o gerou para computar outro quadro diferente deste.

O nome The Village faz mais sentido quando se atribui tamanho variável para as figuras da figura básica de número 5. É possível observar isso na Figura 5 abaixo:

```
*) Voce deseja que as figuras tenham tamanho variavel? (1 - Sim / 2 - Nao)
.> 100000 tentativas para encontrar uma posicao valida para uma das figuras nao foram suficientes...
> 63 figuras foram desenhadas no quadro.
> Interrompendo o desenho...
*) Pressione ENTER para continuar...
Eis a obra de arte...
        WWVWW
WWVWW
      VWWVWWV
พพงพพพพงพพ
                      ***
              VWWVWWV
                                               /o\ o /
                                               /I\/I\
 |I|
      0
                                                              /||o\
                                                                          0
  0
     II
 /I\
                                                 o
     wwvww
                               0
                                 0
   WWVWW
                   VWWVWWV*** /I\/I\
       |I|
(*) Voce deseja gerar um novo quadro com os mesmos parametros informados? (1 - Sim / 2 - Nao)
> 1
```

Figura 5: Foi solicitado o desenho de 100 figuras para este quadro, não sendo possível, entretanto, após cem mil tentativas de reposicionamento de uma destas figuras. Isto aconteceu pois o tamanho variável foi solicitado, considerando assim as demais figuras que complementam a figura The Village. São elas: dois tipos de árvores, sendo um deles um pinheiro (a grama na base da figura faz parte dela) e o outro uma árvore mais comum, e uma casa de dois andares.

Observando a Figura 5, vemos que o reposicionamento se faz necessário para que não haja conflito entre imagens anteriormente colocadas. A checagem de conflitos cuida disso e é eficiente, gerando a cada tentativa novas coordenadas para o centro da figura.

3.2. Menu secundário

A opção 7 do menu principal (Fig. 2) abre um menu secundário, que foi criado para explorar ainda mais a flexibilidade dos códigos. Veja a Figura 6 abaixo:

Figura 6: Menu secundário: a primeira opção lista todas as figuras disponíveis: aquelas que possuem tamanho variável são indicadas em seus respectivos cabeçalhos; a segunda opção mostra o último quadro gerado; a terceira opção remove todas as figuras inseridas no quadro; a quarta opção permite inserir uma das figuras disponíveis em uma posição aleatória, mantendo a ideia do programa.

O quadro gerado por este menu permanece até que uma outra figura seja gerada pelas opções do menu principal ou que o usuário informe a opção de limpar o quadro no menu da Figura 6. A Figura 7 mostra um desenho feito considerando apenas elementos de The Village (entretanto note que pode ser incluído qualquer uma das figuras utilizadas neste trabalho):

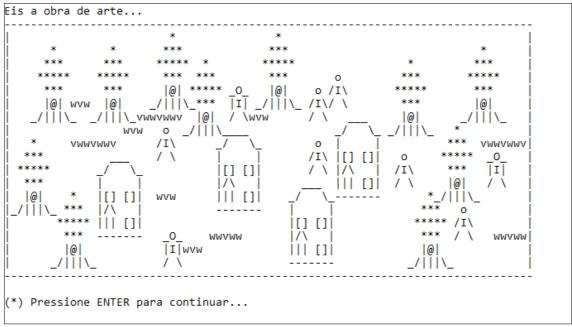
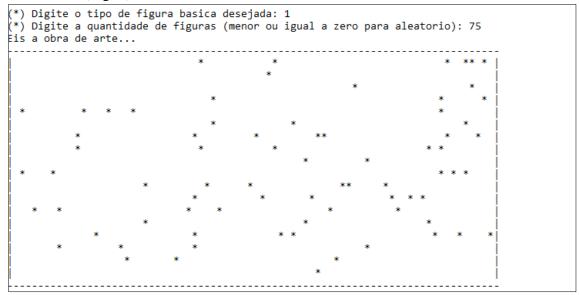


Figura 7: Quadro criado pelo menu secundário (Fig. 6). As figuras utilizadas foram apenas as figuras da obra The Village, mas qualquer uma das figuras utilizadas para este trabalho poderiam ser utilizadas para a construção do quadro. Veja que o cenário remete a um cenário de uma vila em um jogo clássico de RPG.

3.3. Principais resultados

Abaixo seguem os resultados que eram esperados para as figuras básicas pedidas. Apenas para fins de ilustração, 75 figuras do tipo selecionado serão impressas em posições aleatórias (Fig. 8).



(a) Asteriscos simples

(b) Sinal de soma feito com asteriscos.

(c) Letra X feita com asteriscos.

```
Digite o tipo de figura basica desejada: 5
*) Digite a quantidade de figuras (menor ou igual a zero para aleatorio): 75
*) Voce deseja que as figuras tenham tamanho variavel? (1 - Sim / 2 - Nao)
> 2
Eis a obra de arte...
                          wvw/I\
                                                       |I|
 /I\
                  /I\
                             /o\
           wvw
                                          0
                                                                                   wvw
                             /I\
           0
                                         /I\
                                                              wvw O
                                                                         /I\
       o
          /I\
                o
                             /o\ wvw
                                                                 | I | _ O
      /I\
            \ /I\
                             /I\
                                                      0
                                                                   \II
 o
                                                                                  wvw
                                                0
/I\
                                                     /I\
                                                                         wvw
                  0
                                             /I\|I|
                                                                   0
                                                                              II
                               0
                                                   /I\
                                                                  /I\
  0
                                   0
                              ĪI /I\
                                                             |I|
 /I\
           owvw 0
                                                wvw/
                                                                         WVWWVW WVW
          /I\
                |I|
                              / \ / \wvw
                                                                          0
                                                   o
                                                                         II
                                                       o
              0_
           0
                  /I\
                          /I\
                                                  /I\ /I\
                                                                            /I\wvw
                                    0
                                                          WVW
                                                                            0/\
       wvw|I||I|
                                    /I\
                                                                     o
   /I\
                                    /o\
                                                                           /I\wvw
            \/
                                                     0
                  /I\
                                   /I\
                                                    /I\
          WVW
                                   (d) The Village
```

Figura 8: Quadros gerados a partir das figuras básicas, considerando tamanho não variável.

Para ilustrar o tamanho variável de todas as figuras em apenas uma, veja a Figura 9 abaixo, que corresponde a opção 6 do menu principal (Fig. 2):

Figura 9: Quadro gerado a partir de figuras aleatórias, considerando todo tipo de figura disponível no programa.

4. Conclusão

Este trabalho propiciou um bom treinamento da linguagem C e também de técnicas de programação. Por ser um programa interessante e legal de se usar, diversas ideias de manipulações adicionais surgiram, desafiando o modo de como os dados deveriam ser armazenados para uma melhor representação e facilidade de uso para a apresentação final. Os números randômicos foram explorados para gerar tanto posições como figuras aleatórias, como é o caso da obra de arte The Village e as opções de criação 4 e 6 do menu principal (Fig. 2).

Esta foi uma boa aplicação prática da geração de números randômicos, uma vez que muitas das vezes esse conceito não é muito visto em sala de aula. Isto enfatiza a importância dos números randômicos, que são utilizados para gerar até mesmo mapas de jogos, como acontece com o jogo *Minecraft*.

5. Referências bibliográficas

[1] Generate random number in range [min,max]. Disponível em https://stackoverflow.com/questions/29381843/generate-random-number-in-range-min-max. Acesso em 18 ago. 2018.