

Desenvolvimento do Jogo *Sudoku* em Linguagem de Programação C

Thiago Medeiros de Souza¹

¹Curso de Ciência da Computação – Faculdade Anglo-Americano (FAA)
Avenida Paraná, 5661 – Foz do Iguaçu – PR – Brazil

thsouzza@gmail.com

Resumo. *Este artigo trata sobre o desenvolvimento de um jogo Sudoku na linguagem de programação C, com a finalidade de aplicar e aprimorar os conhecimentos adquiridos durante as aulas, utilizando funções, ponteiros, arquivos assim como outras técnicas de programação para a geração de números aleatórios e leitura de arquivos texto. O artigo é parte integrante do sistema de avaliação bimestral da disciplina de Programação de Computadores, ministrada pela Professora Alessandra Bussador no curso de Ciência da Computação das Faculdades Anglo-Americano Foz do Iguaçu.*

1. Jogo *Sudoku*

Sudoku é um quebra-cabeça baseado na colocação lógica de números. O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias em uma matriz 9x9, constituída por uma matriz 3x3 chamadas regiões. O quebra-cabeça contém algumas pistas iniciais, que são números inseridos em algumas células, de maneira a permitir uma indução ou dedução dos números em células que estejam vazias. Cada coluna, linha e região só pode ter um número de cada um dos 1 a 9. Para resolver o problema requer apenas raciocínio lógico e algum tempo. Os problemas são normalmente classificados em relação à sua realização. O aspecto do *Sudoku* lembra outros quebra-cabeças de jornal. Foi criado por Howard Garns, um projetista e arquiteto de 74 anos aposentado.

2. Estratégias e Análises Para o Desenvolvimento

Inicialmente a estratégia adotada para o desenvolvimento do jogo foi criar uma matriz 9x9 vazia e em seguida adicionar valores aleatórios em cada linha, sendo que esses valores não poderiam existir na mesma coluna através de uma condição *if*. Porém não foi possível obter êxito durante os momentos de programação dessa ideia. Através de algumas pesquisas na *Internet* e esboços *brainstorm* surgiu a estratégia de inicializar o programa com uma matriz 9x9 como modelo. Essa matriz obrigatoriamente possui valores corretos em relação as regras do quebra-cabeça.

Com uma matriz modelo existente é criada então uma segunda matriz temporária que copiará os valores da matriz modelo. Essa matriz temporária irá embaralhar os valores, levando em conta as regras do jogo e mantendo-se a condição de não repetir valores nas mesmas linhas ou colunas, através de várias funções, que, para embaralhar os valores irá substituir posição de linhas e colunas assim como inverter a ordem das mesmas. Foi utilizado a geração de números aleatórios de acordo com o tempo (segundos) para chamar cada condição para essas mudanças. Dessa forma então é possível sempre obter um *Sudoku* completo e novo cada vez que o programa invocar a função *inicializar*. Um arquivo texto com a solução será gerado nesse momento.

Para obter os valores vazios foi criado uma função que vai ler o arquivo texto gerado com o *Sudoku* completo e gerar uma nova matriz 9x9 temporária. Essa matriz temporária irá substituir alguns valores aleatórios por um espaço vazio e então gerar um novo arquivo texto sem solução.

Para a interação homem-máquina foi desenvolvido um ambiente de opções com apenas três opções: gerar um novo *Sudoku*; solução do *Sudoku* gerado; fechar o programa. Após a conclusão do jogo três arquivos texto serão gerados, um para a matriz com solução, outro para a matriz sem solução (ambas exatamente igual as que são apresentadas durante a interação com o *software*) e um outro arquivo com valores seguidos da matriz modelo, sem formatação.

3. Funções

Uma parte das funções foram desenvolvidas para embaralhar a matriz modelo, enquanto outras funções tem o objetivo de gerar os arquivos texto e de forma aleatória criar os espaços vazios.

As funções *trocaLinhaPeq* e *trocaColunaPeq* recebem informações por parâmetro de linhas e colunas e trocam uma por outra. *trocaLinhaGrande* e *trocaColunaGrande* tem a mesma função porém chamam as outras duas funções para troca dentro delas, realizando essa substituição mais de uma vez. Para alterar as posições da matriz o programa chama as funções *giraVertical*, *giraHorizontal*, *inverteMatriz*, *inverteMatriz2*, *trocaColLinha* e *substitui*.

Para gerar o arquivo com a solução do *Sudoku* é utilizado a função *geraArquivoSudokuCompleto* enquanto a *imprimeSudokuCompleto* apenas mostra no momento de execução do programa a solução da matriz. *geraArquivoImprimeSudoku* e *geraArquivoSudokuModelo* também criam arquivos textos um apenas para o modelo, pois não existe formatação e outro com o jogo *Sudoku* sem resposta.

No momento da interação com o usuário é chamado a função *geraSudoku* que chamará então as demais funções apresentadas e também a função que inicializará a matriz padrão, *inicializar*.

3.1. Lista de Funções

- *inicializar()*;
- *imprimeSudokuCompleto()*;
- *geraArquivoImprimeSudoku()*;
- *geraArquivoSudokuCompleto()*;
- *geraArquivoSudokuModelo()*;
- *trocaLinhaPeq(int linha1, int linha2)*;
- *trocaLinhaGrande(int lgrande1, int lgrande2)*;
- *trocaColunaPeq(int col1, int col2)*;
- *trocaColunaGrande(int cgrande1, cgrande2)*;
- *giraVertical()*;
- *giraHorizontal()*;
- *trocaColLinha()*;
- *inverteMatriz()*;
- *inverteMatriz2()*;

- substitui(int a, int b);
- geraSudoku();

4. Funcionamento

O programa dispõe de uma *interface* extremamente simples com apenas três opções para o usuário, conforme a figura 1. Após a escolha da primeira opção um *Sudoku* será gerado, assim como o arquivo texto no mesmo diretório do código fonte. Caso o usuário queira verificar a solução pode executar a segunda opção que irá mostrar a solução na tela, apagando o *Sudoku* em branco. Essa opção também gerará um arquivo texto com a solução do jogo.

As opções sempre serão apresentadas na parte de baixo do jogo, podendo ser escolhidas a qualquer momento do programa. Conforme aparece no *display* para sair do programa é necessário pressionar a tecla 'X' ou 'x' seguido da tecla 'ENTER'.



```
Terminal - ./sudoku
File Edit View Terminal Tabs Help

-----
----- SUDOKU -----
-----
Thiago Medeiros

Entre com uma opção:
1. Gerar sudoku (arquivo será criado automaticamente).
2. Solução sudoku (arquivo será criado automaticamente).

---
X' - SAIR
█
```

Figura 1. Interface sudoku.c

5. Conclusão

Programa foi desenvolvido e elaborado com a finalidade de aprimorar e praticar as funcionalidades de técnicas de programação, tais como o uso de ponteiros, funções, criação e manipulação de arquivos. Foram realizadas algumas pesquisas para saber como gerar números aleatórios através da função *rand() % num* e utilização de sementes aleatórias para obter o número de acordo com a variação do tempo em segundos. O jogo foi programado em um ambiente *GNU/Linux* sendo necessário então compilar o código fonte em outros sistemas operacionais.