Jogo Soma-Sudoku

Letícia Gonçalves Souza

Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS) – Feira de Santana, BA

letigsouza03@gmail.com

Abstract. *This project reports the steps of building a virtual Soma-Sudoku game, played by two players, containing different levels and factors. Its development focused on the use of arrays to structure ordered data sets. All program operating requirements were met and well executed, resulting in a fair and well-structured competition platform.*

Resumo. *Este projeto relata os passos de construção de um jogo virtual de Soma-Sudoku, disputado por dois jogadores, contendo níveis e fatores diversos. Seu desenvolvimento teve foco no uso de matrizes para estruturar conjuntos de dados ordenados. Todos os requisitos de funcionamento do programa foram atendidos e bem executados, resultando em uma plataforma de competição justa e bem estruturada.*

1. Introdução

Os jogos fazem parte da interação humana desde a antiguidade. O espírito esportivo e o prazer pela competição estão presentes nas mais diversas culturas.

Assim, os jogos vêm fazendo o papel de suprir a diversão e incentivar o desenvolvimento, raciocínio logico e programação entre os jovens, principalmente aqueles que vivem em regiões de invernos rigorosos e precisam se divertir dentro de casa.

Portanto, os irmãos Lara e Liam que vivem sob esse contexto, solicitaram a turma de MI algoritmos para que desenvolvessem um jogo baseado na fusão do jogo das somas esquecidas e o famoso Sudoku [Teixeira 2014] através de linguagem Python. Fazendo utilização de matrizes, vetores e modularização resultando no Soma-Sudoku.

2. Metodologia

Com o objetivo de desenvolver o software, atendendo aos requisitos e regras, foram realizadas seções PBL. Como produto das seções PBL, foi solicitado o código-fonte de um jogo de Soma-Sudoku. O software foi desenvolvido em linguagem de programação Python.

2.1. Sessões Tutoriais

Foram realizadas reuniões para auxiliar no processo de entrega do produto. A leitura coletiva das regras, que deveriam ser alcançadas, teve grande relevância no esclarecimento de questões que não foram totalmente entendidas de início. O estabelecimento de metas semanais manteve o bom ritmo no avanço da codificação. Além

da troca de experiencias e ideais, que tiveram papel fundamental em determinadas partes da estrutura criada.

2.2. Requisitos

De acordo com o solicitante, o jogo Soma-Sudoku, deve ser disputado por dois jogadores, utilizando um único tabuleiro para ambos.

O usuário deve escolher qual será o nível da partida: Nível 1 ou Nível 2, representando qual será a dimensão do tabuleiro: 4x4 com 4 seções ou 9x9 com 9 seções, respectivamente. Cada uma das seções possui números aleatórios que não se repetem com valores de 1 a 4 para o nível 1 e valores de 1 a 9 para o nível 2.

Diferentemente da regra do Sudoku original, as linhas e colunas do tabuleiro podem ter valores repetidos, sendo a regra apenas aplicada para as seções.

Os valores do tabuleiro permanecem ocultos aos jogadores. Ao lado de cada linha e abaixo de cada coluna são apresentados os valores de suas somas respectivamente, além da diagonal principal do tabuleiro, considerada pontuação bônus, que deve ser revelada após o jogador obtê-la. Diferentemente, as somas das linhas e colunas devem permanecer sempre visíveis aos usuários.

A cada rodada, cada jogador escolher uma seção e um valor que deseje revelar do tabuleiro. Caso o valor revelado complete uma linha ou coluna o jogador acumulará as suas somas em sua pontuação. Caso o valor revelado complete uma linha e uma coluna simultaneamente ele acumulará as somas de ambas em sua pontuação. E por fim, caso o valor revelado complete a diagonal principal, o jogador acumulará a soma da diagonal em dobro em sua pontuação e revelando-a no tabuleiro.

Vence o jogador que tiver conquistado a maior pontuação após todo o tabuleiro ter sido completo.

2.3. Passos do Código

Inicialmente, foi feita a importação da biblioteca Random para sorteio dos valores do tabuleiro, em seguida foram criadas as funções para auxiliar no funcionamento do programa. Em seguida no programa principal foi feita a construção de um menu para pedido das entradas das informações essenciais para execução do jogo.

Após isso, foi elaborado o processo da partida com um tabuleiro para os dois jogadores. Paralelamente, foram pensadas as formas de resolver as seguintes questões: caso o usuário escolhe uma seção ou valor já revelados ele perderá a vez; a seção e valor escolhidos devem estar dentro do intervalo indicado por cada nível do jogo.

Assim que os principais requisitos e regras foram resolvidos e implementados, o programa passou a funcionar perfeitamente, acumulando as pontuações dos jogadores corretamente.

2.4. Descrição do Código

2.4.1. Funções presentes

As funções têm o propósito de deixar o programa o mais enxuto possível e deixar a realização das partidas mais simplificadas diminuindo o tamanho do código principal.

Tabela 01. Funções utilizadas e suas funcionalidades

Nome da Função	Funcionalidade
tabuleiro_4x4	Exibe o tabuleiro 4x4
sec_jogadas4x4	Substitui os valores do tabuleiro 4x4 oculto para o tabuleiro jogável 4x4
substitui4x4	Completa a função sec_jogadas4x4 e finaliza a substituição dos valores em sua respectiva seção
tabuleiro_9x9	Exibe o tabuleiro 9x9
sec_jogadas9x9	Substitui os valores do tabuleiro 9x9 oculto para o tabuleiro jogável 9x9
substitui9x9	Completa a função sec_jogadas9x9 e finaliza a substituição dos valores em sua respectiva seção

A primeira função exibe o tabuleiro 4x4 sempre que chamada, deixando assim mais simples para a execução do código principal.

A função sec_jogadas4x4 faz a substituição dos valores do tabuleiro oculto 4x4 no tabuleiro jogável 4x4 relacionando a entrada da escolha da seção e o valor do respectivo jogador.

A função substitui4x4 completa a função sec_jogadas4x4 relacionando-a com cada valor possível de ser revelado finalizando o processo de exibição do tabuleiro com os valores atualizados.

A função tabuleiro_9x9 funciona da mesma forma que a função tabuleiro_4x4, mas exibindo um tabuleiro de 9x9.

A função sec_jogadas9x9 e substitui9x9 funciona da mesma forma que a função sec_jogadas4x4 e substitui4x4 respectivamente, apenas adaptando os comandos para o tabuleiro 9x9.

2.4.2. Nível 1

No nível 1 são utilizados vetores para preenchimento do tabuleiro oculto e tabuleiro jogável, loops para realização das partidas e condicionais para a contabilização da pontuação dos jogadores.

Tabela 02. Vetores Utilizados

Vetores	Funcionalidade
sessão1_oc, sessão2_oc, sessão3_oc e sessão4_oc	Armazena valores iguais a 0 para ocultar os valores reais para as seções 1, 2, 3 e 4 respectivamente
sessão1, sessão2, sessão3 e sessão4	Armazena valores sorteados de 1 a 4 para as seções 1, 2, 3 e 4 ocultas respectivamente

Os vetores sessão1_oc, sessão2_oc, sessão3_oc e sessão4_oc são utilizados para preencher as seções do tabuleiro jogável, elas são preenchidas com valores 0 para que não seja revelado os valores aos jogadores antes das partidas.

Os vetores sessão1, sessão2, sessão3 e sessão4 são utilizados para guardar os valores sorteados das seções, ou seja, os valores que ficarão ocultos aos jogadores antes das partidas.

Após as declarações dos vetores e variáveis para o nível 1, o pedido dos nomes dos jogadores e a exibição do tabuleiro, foi implantado o loop para a realização das partidas. Inicializando com os pedidos da seção e valor desejados e implementação das condicionais para o acúmulo das pontuações.

2.4.3. Nível 2

No nível 2, assim como no nível 1 são utilizados os mesmos vetores e alguns mais para preenchimento do tabuleiro oculto e tabuleiro jogável, já que eles estão declarados dentro da condicional do seu respectivo nível, loops para realização das partidas e condicionais para a contabilização da pontuação dos jogadores.

Vetores	Funcionalidade
sessão1_oc, sessão2_oc, sessão3_oc, sessão4_oc, sessão5_oc, sessão6_oc, sessão7_oc, sessão8_oc e sessão9_oc	Armazena valores iguais a 0 para ocultar os valores reais para as seções 1, 2, 3, 4, 5, 6, 7, 8 e 9 respectivamente.
sessão1, sessão2, sessão3, sessão4, sessão5, sessão6, sessão7, sessão8 e sessão9	Armazena valores sorteados de 1 a 9 para as seções 1, 2, 3, 4, 5, 6, 7, 8 e 9 ocultas respectivamente.

Os vetores sessão1_oc e suas variações são implementados da mesma forma que no nível 1, facilitando o entendimento do código, alterando apenas na quantidade de valores e dimensão do tabuleiro 9x9.

Os vetores sessão1 e suas variações também são implementados da mesma forma que no nível 1, apenas adaptando-se a dimensão do tabuleiro 9x9.

Após esses vetores serem novamente implementados para o nível 2, o código segue para os loops para realização da partida e contabilização dos pontos dos jogadores, agora adaptados para o tabuleiro 9x9.

3. Resultados e Discussões

A seguir, as instruções para executar o programa e os relatórios contendo seu desempenho

3.1. Percorrendo o programa

No início do programa é apresentado o menu principal mostrando as 3 opções centrais do programa. O usuário escolhera qual o nível desejado e prosseguirá para a partida.

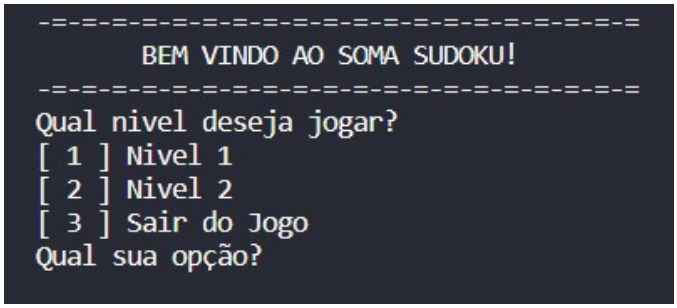

Figura 01. Menu principal

Após a escolha do nível é pedido como os jogadores querem ser chamados.

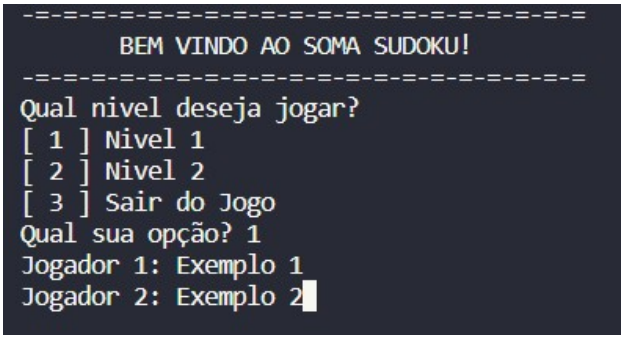

Figura 02. Pedido dos nomes dos jogadores

Ao ser inserido os nomes dos jogadores, o tabuleiro irá aparecer e iniciar a partida oficialmente.

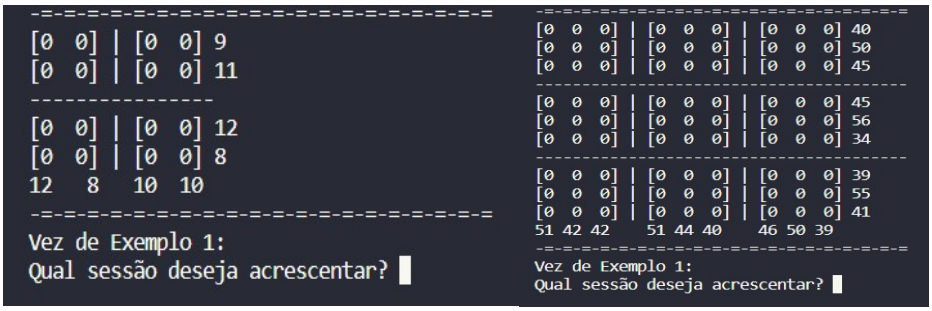

Figura 03. Tabuleiros

Os jogadores então, prosseguirão na partida até que todo o tabuleiro, do respectivo nível, seja preenchido.

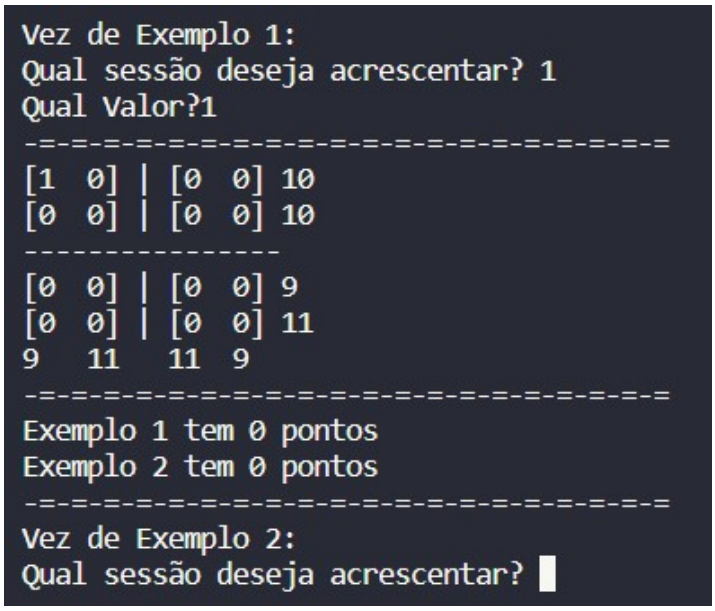

Figura 04. Sequência da partida

Dessa maneira, basta escolher a seção e o valor que queira inserir que será revelado no tabuleiro, a pontuação de cada jogador será atualizada em tempo real a cada rodada (Figura 04)

3.2. Entradas Válidas

As entradas da opção do menu não podem ter valores diferentes dos das opções (Figura 05).

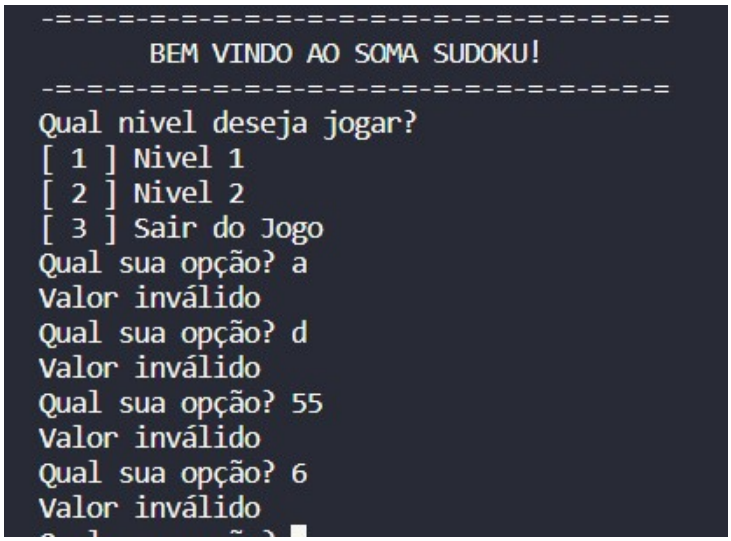

Figura 05. Validação do menu

O mesmo ocorre para a escolha do valor a ser revelado (Figura 06) onde não é permitido usar opções maiores ou menores que a quantidade do tabuleiro.

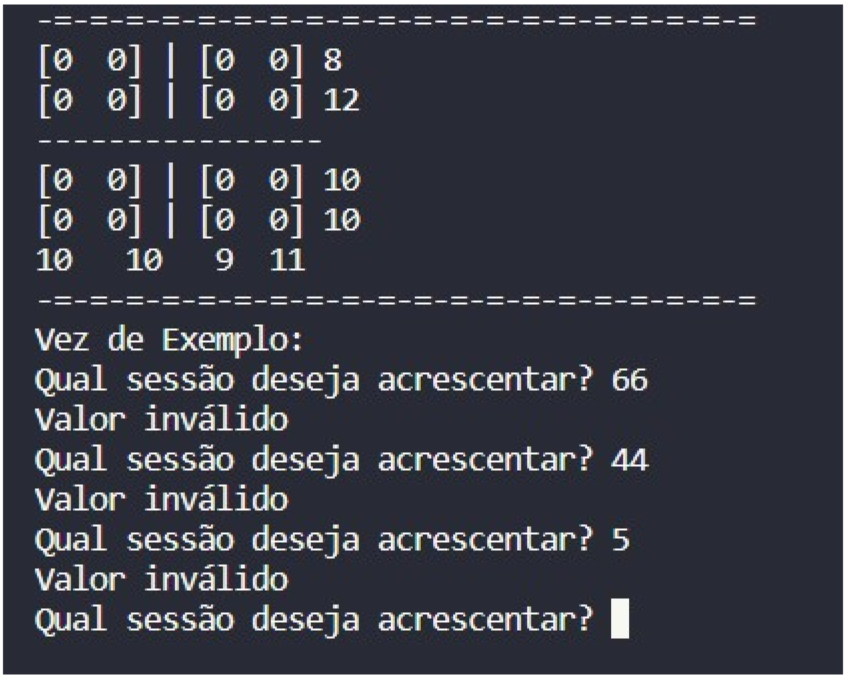

Figura 06. Validação dos pedidos de seção e valor

3.3. Erros e Bugs

Os erros encontrados foram apenas quando a sequência de validação é quebrada, por exemplo, ao usar a validação numérica, em seguida a de valores menores que zero e após isso querer utilizar a numérica novamente, o programa não irá processar a validação numérica após a de valores menores que zero, apresentando um bug. Mas apenas esse problema é apresentado, durante as outras entradas e saídas tudo ocorre de forma concisa e correta.

4. Conclusão

Conclui-se que todos os requisitos solicitados para a construção do programa foram atendidos. O programa funciona de maneira simples e direta sendo de fácil entendimento para o usuário.

Como melhorias para o programa, acrescentar um modo single player seria uma ótima sugestão para aqueles que queiram jogar sozinhos.

5. Referências

TEIXEIRA, Ricardo Emanuel Cunha. O Sudoku. **Tribuna das Ilhas**, p. 10-10, 2014.

MANZANO, José Augusto NG; DE OLIVEIRA, Jayr Figueiredo. **Algoritmos lógica para desenvolvimento de programação de computadores.** Saraiva Educação SA, 2000.