

Sudoku em Python – Eduardo Schultz de Oliveira

O relatório a seguir é sobre o trabalho de programação: um Sudoku feito em Python. Nele, explicarei passo a passo a criação do programa.

Inicialmente, criei o menu inicial, onde adicionei uma “ASCII Art” — ideia sugerida pelo próprio professor. Ainda não havia adicionado os níveis de dificuldade do Sudoku, então havia apenas um botão para iniciar o jogo.

Criar os tabuleiros foi um pouco difícil, pois eu ainda não havia decidido como faria as linhas e colunas. Mas logo encontrei uma solução: dividi o tabuleiro em filas, onde cada fila é uma lista com 9 valores diferentes entre si. Por fim, essas filas se uniriam em uma variável chamada matriz. Assim, criei 5 tabuleiros com valores diferentes, para adicionar um fator de jogabilidade.

Minha próxima ideia foi criar uma função com o objetivo de remover um número fixo de valores desse tabuleiro, deixando-o incompleto, de modo que o usuário tivesse que descobrir os números faltantes. Para isso, o computador escolhia dois números aleatórios de 1 a 9 e os utilizava para selecionar uma posição aleatória da matriz. Se o valor ainda não tivesse sido removido, ele era substituído por um ponto (.).

Um dos problemas que enfrentei foi imprimir o tabuleiro, pois precisei encontrar uma forma de adicionar os valores da matriz a um tabuleiro pré-formatado. Para isso, utilizei o método de formatação `format`, que insere os valores nos lugares corretos. A aparência do tabuleiro foi feita com símbolos do teclado, como traços e barras, além de letras e números para identificar as posições onde o usuário faria suas tentativas.

A última função que criei foi a principal, responsável por permitir ao jogador jogar. Nela, era necessário que o jogador escolhesse uma posição e um valor para preenchê-la. Em seguida, o programa verificava se o número escolhido estava correto e, se estivesse, validava a jogada. Primeiro, defini como localizar a posição escolhida pelo usuário: para as colunas (1 a 9), converti o valor para inteiro e subtraí 1, obtendo assim o índice correto. Para as linhas, utilizei o método `.index`, que retorna a primeira ocorrência de um valor em uma lista — neste caso, uma lista com as letras de A a I.

Enfrentei muitos problemas ao comparar as jogadas. Procurando online, encontrei em um fórum uma sugestão sobre o módulo `copy`, que permite copiar exatamente os valores do tabuleiro original antes de sua alteração. Assim, os valores no tabuleiro antigo poderiam ser alterados sem afetar a cópia original.

Para prevenir erros, adicionei alguns parâmetros para verificar se a resposta do usuário estava correta. Dessa forma, o código não quebraria em caso de entrada inválida. Também adicionei o sistema de dificuldade, que, de acordo com a escolha do usuário, altera a quantidade de remoções no tabuleiro. Por fim, utilizei o módulo `os` para limpar o terminal, facilitando a compreensão por parte do usuário.

Nesse trabalho, utilizei como forma de pesquisa principalmente o site Stack Overflow.