

Sudoku

O jogo começa com uma tela inicial, mostrando o título e qual dificuldade o jogador quer selecionar



-----> Selecione a dificuldade do jogo <-----

A

Fácil

B

Médio

C

Difícil

Dependendo da dificuldade, o número de quadrados em branco será maior ou menor, aumentando ou diminuindo a dificuldade.
O jogador também pode colocar um valor numérico, ele será igual ao número de valores excluídos no início do jogo.

Tabuleiro no modo fácil:

3	4	5	9	1	2	6	7	8
6			3		5	9	1	2
9	1	2	6	7	8		4	5
2	3	4	8	9		5	6	
5	6	7	2	3	4	8		1
8	9		5	6	7	2	3	4
1		3	7	8	9	4	5	6
4	5	6		2	3	7	8	9
7	8	9	4	5	6	1	2	3

Tabuleiro no modo difícil:

		3		9	
9					
3	4				
5		2			1
8					
2	3				
4		6			
7	8		6		2
1		3	8		5

Iniciando o jogo, o jogador será introduzido ao tabuleiro.

Você pode reescrever valores se achou que errou!

0 para apagar um valor

	A	B	C	D	E	F	G	H	I
1	8	9	1	5	6	7	2	3	4
2	2	3	4	8	9	1	5	6	7
3		6			3		8		1
4	7	8	9	4	5	6	1	2	
5	1	2	3	7	8	9	4	5	6
6	4	5		1	2	3	7	8	9
7	6	7	8	3	4	5	9	1	2
8	9	1	2	6		8			5
9	3	4	5	9	1	2	6	7	8

LETRA, NUMERO da casa e o valor ('casa':A1 'valor':2)

O jogador deverá colocar os valores em estilo notação de Xadrez, primeiro a letra e depois o número, para o local, seguido do valor que quer colocar.

As regras são simples, você deve preencher o tabuleiro com valores que:

O valor **não** esteja na mesma linha do tabuleiro

O valor **não** esteja na mesma coluna do tabuleiro

O valor **não** esteja no mesmo quadrante do tabuleiro

Se nenhum desses parâmetros forem cumpridos, o jogador poderá colocar o valor que ele quer no quadrado desejado. Junto com a confirmação de vitória.

Se o jogador achar que errou na jogada e quiser voltar atrás, ele poderá escrever a casa e o valor 0, para excluir um valor.

Quando o jogador conseguir preencher todos os quadrados sem ter nenhum erro, o jogador receberá o tabuleiro verificado, e a confirmação de vitória.

	A	B	C	D	E	F	G	H	I
1	4	5	6	1	2	3	7	8	9
2	7	8	9	4	5	6	1	2	3
3	1	2	3	7	8	9	4	5	6
4	3	4	5	9	1	2	6	7	8
5	6	7	8	3	4	5	9	1	2
6	9	1	2	6	7	8	3	4	5
7	2	3	4	8	9	1	5	6	7
8	5	6	7	2	3	4	8	9	1
9	8	9	1	5	6	7	2	3	4

O jogo acabou! Você ganhou!

Deseja continuar jogando ou acabar? (c/a)

Aqui o jogador é pedido se ele quer continuar jogando, ou fechar o programa.

O código

O código é constituído de 4 funções de lógica e 2 de print.

`criarBoard()`

A função principal, utilizada para criar o tabuleiro do jogo. Criando 9 linhas, cada linha com 9 objetos '0', assim, criando um tabuleiro nove por nove.

`printBoard()`

Função utilizada para imprimir o tabuleiro de uma forma bonita e não bruta.

Ela passa por todos os valores, se o valor for 0, ele irá imprimir um espaço vazio, não deixando poluído o tabuleiro.

Também colocando customizações de borda, como as letras e números de orientação.

Esta função tem um parâmetro opcional de cor do texto. Colocando '\x1b[(valores)' consegue alterar as cores do texto, utilizado quando o jogador ganha a partida.

Podendo mudar: A cor do texto, estilo do texto e cor de fundo do texto.

`printTitle()`

Função para imprimir o título "Sudoku" em arte ASCII

`criarValores()`

A função principal da criação dos tabuleiros randomizados.

Ele usa um algoritmo de criação de tabuleiros de sudoku em que:

Começa no primeiro quadrado, com o valor 1, indo até o final do quadrante aumentando os valores. Neste primeiro, sendo do 1 até o 9

No próximo quadrante, o programa move para o quadrante de baixo, e aumenta a posição inicial em 1, então começa em: 0 1 ... 8. E no início, colocar o valor que faltou no final, então ficando 9 1 ... 8.

Indo assim até o final.

Essa mudança de posição é chamada de **Offset**.

Quando essa função é chamada, o offset é randomizado do 0 até o 8, criando assim um tabuleiro customizado toda sessão de jogo.

`excluirValores(dificuldade: int)`

Esta função é para excluir valores dependendo da dificuldade, que é um número inteiro.

A função pega a dificuldade e exclui 'dificuldade' valores. Randomizando o passo da exclusão dos valores, para ter uma aleatoriedade toda partida.

```
checkLinha(linha, coluna, valor)
```

Função para verificar se a jogada que o jogador quer é válida. A função passa pela linha que o jogador jogou, verifica se a jogada é válida. Se for inválida, ele retorna um `False` e pede outro valor para jogar. Mostrando que o problema é na linha. Depois a função verifica a coluna e também retorna o problema, se tiver algum.

E no fim a função passa pelo quadrante da jogada.

Loop principal

No loop principal do jogo, ele irá entrar no menu. Printando o título 'Sudoku' com as opções de dificuldade.

Se o jogador digitar a letra no quadrado da dificuldade, o nome da dificuldade ou a primeira letra da dificuldade, ele irá pegar a tal dificuldade. Então para fácil pode escrever "A", "fácil" ou "f", todos são válidos.

O jogador também pode escolher um valor numérico, que irá ser igual ao número de quadrados excluídos.

Após selecionar a dificuldade, é chamado às funções `criarValores()` e `excluirValores(dificuldade)`.

Assim, entrando no próximo Loop,

O Jogo

De princípio, irá imprimir o tabuleiro, e as instruções de como jogar, e pedindo para o usuário o movimento.

Assim que o jogador confirma os valores, o programa verifica se os valores são válidos. Se não forem fala pro jogador escrever de novo.

Então o programa chama `checkLinha(linha, coluna, valor)` com os valores das posições e o valor que o jogador escolheu.

Se retornar `True`, o programa irá mudar os valores da posição, limpar o terminal e chamar `printBoard()`

Então, irá passar por todas as linhas do tabuleiro. Se alguém tiver o valor zero, essa verificação irá parar e não fazer nada.

Se não tiver zeros na tabela, quer dizer que todos os espaços foram preenchidos, assim, o jogo ganhou, quebrando o loop de jogo, limpando o terminal e imprimindo o tabuleiro com a cor verde usando `printBoard('\x1b[1;32;1m')`, e pedindo para o jogador se ele quer continuar jogando, que levará ele para o menu novamente, ou se ele quer acabar o jogo, assim fechando o programa.

Problemas que podem acontecer fora do meu controle

Para limpar o terminal, estou usando a biblioteca OS. No windows o comando para limpar é **system('cls')**, mas em outros OS é **system('clear')**. E sem limpar o terminal o código responsável pela mudança de cor é corrompido ou algo do tipo.

Outro problema é a mudança de cor do texto. Eu acho que isso é uma função do terminal do Windows, não sei se funciona em outros OS, não tenho como testar, então fica na dúvida. O esperado é como está nas prints acima.