

Data de Entrega: 27/06/2011

A nota do trabalho corresponderá à 20% da nota final

A cada dia de atraso serão descontados 20% da nota do trabalho, ou seja, com 5 dias de atraso a nota é zero.

Serão aceitos somente trabalhos entregues pessoalmente, na sala de aula ou na minha sala.

As questões teóricas deverão ser entregues impressas ou manuscritas;

Questões copiadas de outros colegas, os dois levam zero naquela questão.

Trabalho Prático

1. Implementar um jogo Sudoku em Lazarus. (Valor 70 pontos)

O objetivo do jogo é a colocação de números de 1 a 9 em cada uma das células vazias numa grade de 9×9, constituída por 3×3 subgrades 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 (3x3) só pode ter um número de cada um dos 1 a 9. Resolver o problema requer apenas raciocínio lógico e algum tempo. (Fonte: Wikipédia)

Exemplo de aplicação:

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

Figura 1 Um problema Sudoku difícil...

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

Figura 2 ... e sua solução, marcada em vermelho

Tutorial

Introdução

Sudoku é um jogo de raciocínio e lógica. Apesar de ser bastante simples, é divertido e viciante.

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

Cada jogo dura de 10 a 40 minutos, dependendo do nível de dificuldade e da experiência do jogador.

Objetivo

O objetivo do jogo é completar todos os quadrados utilizando números de 1 a 9. Para completá-los, seguiremos a seguinte regra: Não podem haver números repetidos nas linhas horizontais e verticais, assim como nos quadrados grandes.

1
2
3
4
5
6
7
8
9

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

1	2	3
4	5	6
7	8	9

Níveis de dificuldade

Ao iniciar um jogo, você poderá escolher entre 3 níveis de dificuldade: fácil, medio, difícil. A principal diferença destes níveis está na quantidade de quadrados preenchidos e na disposição dos números. De modo que no nível difícil, por exemplo, você terá que analisar a disposição dos números de uma maneira mais global para poder encontrar os números que restam.

O jogo

Analisando uma jogada errada

Após fazer uma jogada, há a possibilidade de verificar se ela foi correta ou não clicando no botão "Como estou indo?". Se a jogada estiver incorreta, o local da jogada ficara em vermelho indicando o erro. Veja as imagens que apresentam os possíveis erros:

Número repetido em um quadrado grande:

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

Número repetido na linha vertical:

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

Número repetido na linha horizontal:

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

Para saber mais:

<http://pt.wikipedia.org/wiki/Sudoku>

Para ver o funcionamento, jogue:

<http://sudoku.net.br/>

2. Implemente o jogo “Resta 1” no Lazarus. (Valor 30 pontos)

O jogo Resta 1 é jogado em um tabuleiro como o mostrado na Figura abaixo (a). O objetivo do jogo é eliminar as peças de forma que reste apenas uma. As peças são eliminadas pulando uma sobre a outra nas direções vertical e horizontal, como mostra a Figura abaixo(b).

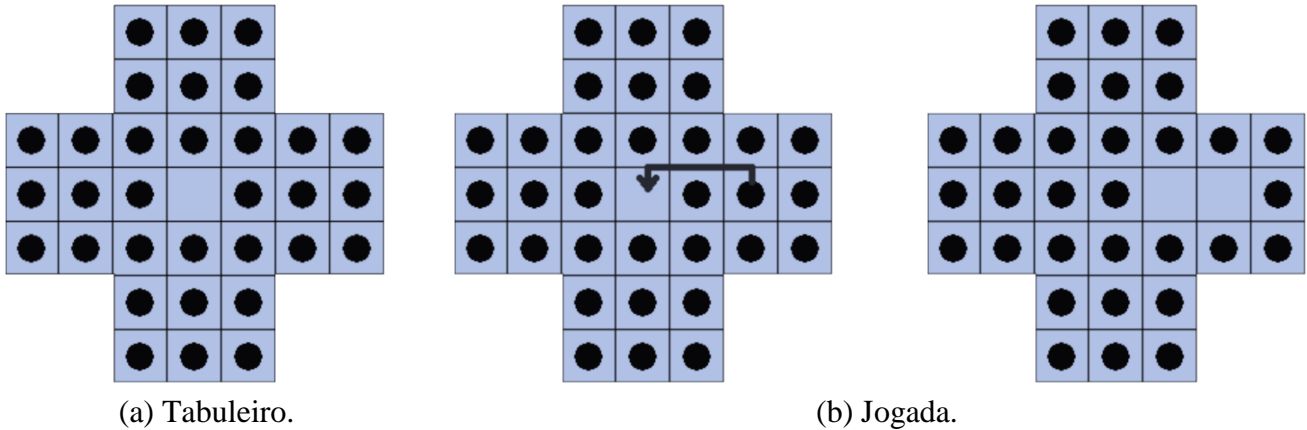


Figura: Jogo Resta 1.



Figura: Programa Jogo Resta 1

Projete um formulário como mostrado na Figura 5

- O botão “Sair” deve fechar o formulário;
- Crie um procedimento **NovoJogo()** para inicializar o tabuleiro como mostrado na Figura 5(b). Este procedimento deve ser chamado na execução do programa e pelo botão “Novo Jogo”. As peças são representadas com “O” (o maiúsculo) e a posição central fica com um “.” (ponto) indicando que está vazia. O procedimento deve alterar o **Caption** do formulário para “Resta 1 (Jogando)”. Dica: Preencha as três colunas, as três linhas e depois o ponto central;

- c) Para realizar uma jogada o usuário precisa clicar em uma posição inicial e em uma posição final. Portanto a cada clique é preciso saber se a jogada está começando ou terminando. Para fazer isto, declare as variáveis globais **JogadaIniciada**, **LinhaInicial** e **ColunaInicial**.
- d) Modifique o procedimento **NovoJogo()** para que **JogadaIniciada** seja **false**, pois quando o jogo começa o usuário não iniciou nenhuma jogada;
- e) Crie o procedimento **Jogada(L, C: Integer)** que exiba apenas uma mensagem dizendo se a jogada esta começando ou terminando (por enquanto, não precisa mexer com as peças). Dica: utilize a variável global **JogadaIniciada**;
- f) Codifique o evento necessário para que o procedimento **Jogada()** seja chamado quando o usuário clicar na grade. Passe **GridResta1.Selection.Top** como parâmetro para **L** e **GridResta1.Selection.Left** como parâmetro para **C**;
- g) Modifique o procedimento **Jogada()** para que uma jogada seja iniciada apenas se o usuário clicar sobre uma peça, ou seja, quando houver um clique em uma célula com “O”. Neste caso o procedimento deve armazenar em **LinhaInicial** e **ColunaInicial** a linha (**L**) e a coluna (**C**) clicadas, respectivamente;
- h) Agora modifique o procedimento **Jogada()** para finalizar uma jogada corretamente. Uma jogada pode ser finalizada se a posição original possui peça, a posição de destino está vazia e a posição do meio possui peça. Existem quatro possibilidades:
- Jogada para cima: **C = ColunaInicial**, **L = LinhaInicial - 2**, mostrado na Figura 6(a);
 - Jogada para baixo: **C = ColunaInicial**, **L = LinhaInicial + 2**, mostrado na Figura 6(b);
 - Jogada para a esquerda: **L = LinhaInicial**, **C = ColunaInicial - 2**, mostrado na Figura 6(c);
 - Jogada para a direita: **L = LinhaInicial**, **C = ColunaInicial + 2**, mostrado na Figura 6(d).

.
O
O

(a) Jogada para cima

O
O
.

(b) Jogada para baixo

.	O	O
---	---	---

(c) Jogada para a esquerda

O	O	.
---	---	---

(d) Jogada para a direita

Se uma das possibilidades for atendida, as posições de origem e do meio devem ficar vazias (“.”) e a posição de destino preenchida (“O”).

- i) Crie a função **FimJogo(): Boolean** para verificar se o jogo terminou. Esta função deve percorrer a grade e verificar se ainda existem jogadas a serem realizadas. Se houverem possíveis jogadas a função deve retornar **false**, senão deve retornar **true**. Dica: percorra a grade da segunda até a penúltima linha e da segunda até a penúltima coluna;
- j) Modifique o procedimento **Jogada()** chamando a função **FimJogo()** no seu final. Se **FimJogo()** retornar **true** mostre uma mensagem para o usuário mostrando que o jogo terminou e modifique o **Caption** do formulário para “Resta 1 (Fim de jogo)”;
- k) Crie a função **ContaPecas():Integer** que conte o número de peças na grade. Modifique novamente o procedimento **Jogada()** para que sejam exibidas o número de peças restantes no final do jogo.

Para saber mais:

http://pt.wikipedia.org/wiki/Resta_um

Para ver o funcionamento, jogue:

<http://www.divertudo.com.br/restaum/restaum.html>