

LOGO . MAT

Um procedimento de cada vez Diálogos com a tartaruga?!

É com certeza uma perspectiva interessante. Mas como fazê-lo? Bom, será isso que, de uma forma resumida, iremos ver. Isto é, falaremos das primitivas que estabelecem comunicação entre o computador e o utilizador, permitindo-lhe introduzir novos dados no decorrer de um programa.

1. readchar, readlist, readlistcc

Experimente o procedimento:

```
to experiência1
print [Carregue numa tecla]
print readchar
end
```

readchar lê o caractere correspondente a uma tecla premida e dá como resultado esse caractere. Se nenhuma tecla tiver sido premida, **readchar** espera que tal aconteça.

readchar não faz aparecer o caractere no ecrã pelo que, se o quisermos ver, temos que o mandar escrever.

Experimente mais:

```
to experiência2
ct
print [Escreve o teu nome e carrega em enter]
make "nome first readlist
(print "Olá :nome)
end
```

readlist (rl) lê uma sequência de palavras escrita através do teclado, e dá como resultado a lista constituída por essa(s) palavra(s).

readlist espera que carreguemos na tecla *enter*, sinal de que a sequência de palavras acabou.

Uma vez que **readlist** forma uma lista com a(s) palavra(s) por nós escritas, torna-se necessário encontrar um processo para aceder ao conteúdo dessa lista, permitindo a sua futura utilização. No caso da lista ter um único elemento, a primitiva **first** resolve o problema. Noutros casos... bom, temos que deixar algumas coisas para vocês pensarem.

Se quisermos que as nossas mensagens apareçam na zona de comandos (o que poderá ser conveniente quando não pretendemos misturar gráficos com texto, como por exemplo num jogo) utilizamos a primitiva **readlistcc** (rlcc).

Construa um procedimento substituindo, no procedimento **experiência2**, **ct** por **cc**, **print** por **show** e **readlist** por **readlistcc**, e execute-o. Observe as diferenças.

E que podemos nós fazer com tudo isto? Como diz um conceituado logoísta nosso amigo, não há como os

quadrados para iniciar o Logo. Então voltemos aos quadrados, mas que sejam quadrados sofisticados...

Suponhamos que pretendíamos que a simpática tartaruga nos perguntasse qual a medida com que deveria desenhar o quadrado:

```
to quadrado
ct rg
insert [Escreve a medida que queres para o lado
do quadrado e carrega em]
insert char 32
settc 2 print "enter
settc 1
print [ ]
make "lado first readlist
repeat 4 [fd :lado rt 90]
....
```

já agora, poderia calcular a área do quadrado, uma vez que a medida do lado está guardada na variável de nome **lado**:

```
repeat 10 [print [ ]
(print [A área do quadrado é] :lado * :lado)
....
```

pode, ainda, perguntar ao utilizador se quer repetir o programa:

```
print [ ]
print [Queres fazer outro quadrado? (S/N)]
make "resp readchar
ifelse :resp = "s
[quadrado]
[ct rg pr [Até à próxima!] wait 50 ct stop]
end
```

Aproveitámos para introduzir alguns "truques" que podem ser úteis neste tipo de projectos, como seja mudar o texto de cor (**settc** — só no Logowriter 2.0), deixar espaços em branco (**insert char 32**) e deixar linhas em branco (**print []**).

2. Key?

Experimente:

```
show key?
```

```
wait 50 show key?
```

carregue numa tecla imediatamente após ter carregado em enter

Key? dá como resultado "false" (falso) se não carregarmos numa tecla, caso contrário dá como resultado "true" (verdade).

Para que **key?** dê como resultado verdade, o caractere não pode ter sido lido por nenhuma das primitivas referidas em 1.

Para que poderá servir esta primitiva?

Suponhamos que queríamos que a tartaruga desenhasse uma circunferência (outra das aprendizagens fundamentais num bom curso de Logo), mas pudéssemos

interromper e continuar o desenho sempre que quiséssemos:

```
to circunferência
repeat 360 [fd 1 rt 1 parar]
end

to parar
if key? [repeat 2 [make "lixo readchar]]
end
```

Afinal o que faz o procedimento **parar**? Não é elementar, mas vejamos se nos entendemos.

Se, durante o desenho da circunferência, não carregarmos em nenhuma tecla, **parar** não faz nada (uma vez que o resultado de **key?** é "false"). Pelo contrário, se premirmos uma tecla, tornamos o valor de saída de **key?** "true" e **readchar** interrompe a execução do procedimento, até que uma tecla volte a ser premeida. Mas tal só acontece na segunda vez que a instrução **make "lixo readchar** é executada, pois na primeira vez **readchar** lê o caractere correspondente à tecla já premeida e portanto interrompe o traçado da circunferência.

Por outro lado, uma vez que **readchar** leu os caracteres correspondentes às teclas em que carregámos, **key?** dá falso no fim do procedimento **parar**, pelo que, se voltarmos a carregar numa tecla a tartaruga volta a parar e assim sucessivamente.

Não é demais salientar que os exemplos referidos pretendem ser, apenas, uma primeira abordagem do que podem ser projectos utilizando estas primitivas. Fica para vocês os ampliarem, modificarem, etc.

Para terminar, uma pequena nota de carácter pedagógico. Pelo que nos foi dado observar, a utilização destas primitivas, embora não sendo elementar, suscita grande adesão por parte dos alunos que trabalham com o Logo, talvez porque ao permitir-lhes uma relação mais pessoal com a máquina, aumenta a sua criatividade e a originalidade do seu trabalho. Além de que, tal como acontece com professores, há alguns alunos para quem os bonecos da tartaruga não têm uma graça por aí além, aderindo melhor a projectos deste tipo.

Margarida Junqueiro
Sérgio Valente

Um mágico quadrado mágico (conclusão)

Debatendo-se entre o panteísmo germânico e o idealismo do Renascimento, Dürer assume um pessimismo filosófico que exprimiu em diversas gravuras: *Nemesis* (1503), *o Cavaleiro, a Morte e o Diabo* (1513) e *a Melancolia* (1514), símbolo da inutilidade das ciências e das obras humanas.

Onde se fala, finalmente, do quadrado mágico

Depois de contemplar a última gravura referida, que se reproduz na página seguinte, concentre-se no quadrado mágico que aí figura. Cada linha, cada coluna, cada diagonal conduzem, sempre, à soma 34.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Quadrado mágico de Dürer

Mas trata-se, de facto, de um mágico quadrado mágico! A gravura data de 1514, como pode ler-se nas células centrais da última linha.

Adicione os números representados nos quatro cantos do quadrado principal. A soma é, ainda, 34.

Adicione, agora, os números que figuram nas quatro células centrais. Ainda a soma 34.

Experimente, agora, com as células interiores da primeira e da última linhas. Ainda 34.

O mesmo com as células interiores da primeira e da quarta colunas. Ainda e sempre 34.

Adicione, também, as duas primeiras células da coluna um com as duas últimas da coluna quatro. Não me diga que ficou admirado(a) ao encontrar a soma 34...

Calcule a soma das duas diagonais de cada um dos quatro quadrados de 2×2 , com um dos vértices coincidente com um vértice do quadrado principal. Já não se espantou, não é verdade?

Pois bem, descubra todos os quadriláteros cujos vértices têm por "soma" 34. Mande as soluções para a revista que nós prometemos dar-lhe os parabéns se acertar.

Leonora Moreira

Bibliografia

Bazin, G. (1953). *Histoire de l'art — de la Pré-histoire à nos jours*. Paris: Garamond.

Classici dell'arte Rizzoli (1968). *L'opera completa di Dürer*. Milão: Rizzoli.

Wills III, Herbert. (1989). *Magic with Magic Squares*. *Arithmetic Teacher*, vol. 36, n.º 8, 44-49.