

Um procedimento de cada vez (TELL e os que estiverem «à mão de semear»)

Fernando Nunes

Qualquer *logoista* pode, tal como Schiller ou Rossini, sentir a curiosidade de recriar um episódio baseado no célebre desafio protagonizado pelo arqueiro suíço Guilherme Tell. Contando com várias tartarugas que representarão o arqueiro, o seu filho, a seta e a maçã, pode ser construído um drama num ou mais actos. Para «apimentar» um pouco mais, podem ser colocados ao acaso (ver o LOGO.MAT do n.º 9), o arqueiro e a maçã, surgindo a seta com uma orientação também aleatória, o que obrigará o utilizador a estimar a rotação necessária para que a maçã possa ser atingida.

Por sorte, algumas versões de Logo apresentam a possibilidade de trabalhar, simultaneamente, com mais que uma tartaruga. No caso do Logo Writer são quatro as tartarugas disponíveis.

Para quem já fez programas em que utilizou a animação, por exemplo a corrida da lebre e da tartaruga ou o trânsito numa estrada, notou com certeza a vantagem tirada do facto de se poderem usar várias tartarugas. O mesmo deve ter acontecido a quem fez a simulação do lançamento simultâneo de vários dados ou a da perseguição de um gato a um rato. Jogos que necessitem do movimento de mais do que um objecto, também podem ser programados com tartarugas múltiplas.

Logo após a introdução do programa, apenas está activa — pronta para executar instruções — a tartaruga 0 colocada no ponto [0 0]. As tartarugas 1, 2 e 3 estão invisíveis e inactivas nas suas «casas», respectivamente [0 -40], [-40 -40] e [-40 0].

É com o comando **tell** que as diversas tartarugas são activadas. Este comando aceita como entrada (input) um número de 0 a 3, uma lista composta por quaisquer destes números ou ainda o relator **all**, equivalente à lista [0 1 2 3]. Todas as instruções seguintes estão sintaticamente correctas:

```
tell 2
tell [1 2]
tell all
```

Uma das particularidades deste comando (não única em Logo, recordemos por exemplo os comandos relativos ao estado da caneta), reside no facto de não serem perceptíveis as suas consequências logo após a sua execução. As tartarugas nomeadas no último **tell** obedecerão a todas as instruções posteriores, só alterando este estado um novo **tell** que modifique o conjunto das tar-

tarugas activas ou o comando **rg** que, além de vários outros efeitos, restaura as condições iniciais, ficando visível e activa apenas a tartaruga 0. Existe a possibilidade de nos informarmos de quais as tartarugas estão activas nesse momento, usando o relator **who** que fornece o número ou a lista de números referentes a essas tartarugas.

Existe no entanto uma forma de, temporariamente, apenas alguma(s) das tartarugas (activas ou não) executarem instruções sem ter que se recorrer a **tell**:

ask 2 [fd 50]

Apenas a tartaruga 2 se irá deslocar 50 unidades para a frente, qualquer que seja o conjunto das tartarugas activas, que não é alterado pelo procedimento primitivo **ask**. A sua sintaxe obriga à indicação de dois inputs. O primeiro deve ser um número ou uma lista de números, referentes às tartarugas que irão executar as instruções contidas na lista do segundo input.

Outro comando, associado à utilização de várias tartarugas e nos permite a execução sequenciada, portanto não simultânea, por parte das tartarugas activas é o procedimento primitivo **each**.

tell all
each [fd 50]

As instruções acima referidas têm como resultado final exactamente o mesmo que obteríamos com **tell all fd 50**. A diferença reside na forma como as tartarugas as executam: ordenadamente (primeiro a 0 até à 3) se usarmos **each**, simultaneamente no caso contrário.

Em Logo, como em quase tudo, há muitas maneiras de alcançar o mesmo resultado o que, diga-se em jeito de parênteses, pode valorizar os meios usados para atingir os fins pretendidos. O célebre quadrado, normalmente desenhado só com uma tartaruga, pode ser obtido desta maneira:

```
tell all
each [seth (180 + 90 * who)]
fd 40
```

Será de notar que o relator **who** vai tomando sucessivamente os valores 0, 1, 2 e 3 devido ao uso de **each**, além de que as tartarugas devem estar todas nas suas «casas».

Estes seis procedimentos primitivos aqui analisados (**tell**, **ask**, **each**, **rg**, **all** e **who**) podem ser associados a outros, de modo a possibilitar a exploração de situações problemáticas que apresentam interesse para o ensino da Matemática.