

Apresentação

Algo que fascina crianças e adolescentes são os vídeo-jogos. Melhor do que jogá-los, é criá-los. A tarefa não é fácil. Requer tempo, dedicação, paciência, ... Nessa tarefa, muita coisa se pode aprender ou ter que estudar: técnicas de programação, lógica, psicologia, matemática, princípios físicos, etc. Nessa oficina, tenta-se explorar alguns princípios que devem ser levados em conta ao criar um jogo (seja ele pedagógico ou não):

- 1) Interatividade (via teclado ou mouse);
- 2) Multimídia (é importante fazer uso de imagem e som);
- 3) Elemento surpresa.

O elemento “surpresa” pode ajudar a evitar que um jogo seja repetitivo, o que levaria o usuário a cansar do mesmo, desistindo dele após pouco dias de uso. Alguns jogos são tão monótonos ou óbvios que o usuário sequer joga mais de uma vez. Finais alternativos, objetos escondidos no cenário (que mudam de lugar a cada novo jogo), mudanças de cenário e alternância nos diálogos são algumas das possibilidades a serem pensadas em um jogo.

Além dos princípios citados acima, considere que no processo de criação de um jogo, deve-se pensar o que se quer com o jogo (fazer um esboço, um roteiro) e qual o seu público alvo. É justamente aí que está o ponto de partida. Geralmente começo um projeto do seu básico. A parte estética e os princípios acima citados são deixados para pensar mais adiante. Afinal, de que adianta concentrarmos tanto em tudo isso sem termos antes feito o programa funcionar? Por exemplo, de que adianta começarmos a criar um jogo de adivinhações com cenário maravilhoso, músicas escolhidas a dedo, ... se não conseguimos pelo menos fazer com que o jogo sorteie um número para adivinharmos? É como se estressar com a decoração de uma casa antes mesmo de construir seus alicerces.

Interatividade com Teclado e Mouse

Faça o Megalogo ler o teclado:

```
atr "ABC leiatecla
```

O MegaLogo pára até que seja pressionada uma tecla.

Melhor ainda é deixar a tat fazer algo:

```
atr "ABC leiatecla
se :ABC = 97 [pf 45]
```

Para saber o número de uma tecla basta escrever **es leiatecla** na linha de comandos e pressionar uma tecla desejada.

No exemplo acima, a tat andar 45 passos se pressionarmos a tecla “A”.

Podemos sofisticar um pouco:

```
para andar
atr "ABC leiatecla
; leiatecla faz a tat ler a tecla
pressionada
se :ABC = -72 [pf 45]
se :ABC = -80 [pt 45]
se :ABC = -77 [gd 45]
se :ABC = -75 [ge 45]
se :ABC = 27 [pare]
se :ABC = -79 [pare]
andar
; é necessária a recursão para que
; a leitura do teclado seja contínua.
fim
```

Pode ser interessante dar a opção de parar o jogo. Como isso poderia ser feito? Que tal colocar um cenário no fundo?

Isso pode ser feito escolhendo uma imagem bitmap (BMP) no menu Arquivo

-> Carregar gráficos...ou ainda com o comando carregagráficos:

```
carregagráficos "ceu.bmp
```

O jogo Supertat (lab2.lgp) é um exemplo simples que utiliza os recursos mostrados até aqui (figura acima). Compare-o com o jogo do Labirinto (lab1.lgp).



O Megalogo permite interação com o mouse. Há dois comandos: rato (mouse) e estadorato (mousestate).

Vejamos um exemplo simples de uso:

```
para desenhar :in
se :in = -1 [fixapos mouse]
desenhar leiatecla
fim
```

Incrementar um projeto com imagens BMP. Elas podem ser feitas com programas tais como o Paint e o Drawing for Children. Se preferir, desenhe no Megapaint.

Escreva **desenhar leiatecla** na linha de comando e mova o mouse com o botão esquerdo pressionado.

Para incrementar um pouco mais, modifique o procedimento **desenhar**:

```
para desenhar :in
se :in = 0 [un fixapos mouse ul]
se :in = -1 [fixapos mouse]
se :in = 27 [pare]
desenhar leiatecla
fim
```

É uma boa idéia manter o arquivo ajuda profissional aberto para tirar dúvidas sobre comandos!

Agora, o comportamento do mouse é diferente e ainda podemos interromper o procedimento usando a tecla ESC (número 27).

Uso de Som e Video

Digite o comando toquewave:

```
toquewave [Opa.wav]
```

Dentro dos colchetes coloca-se o nome do arquivo de som que se deseja tocar. Não é necessária sua extensão (.wav).

O Megalogo permite usar comando mci. Isso significa que você pode usar poderosos recursos multimídia de seu computador. O arquivo ajuda do Megalogo não esclarece como usá-lo e foge aos propósitos dessa oficina. Para facilitar o trabalho, use o seguinte procedimento:

```
para toquemidia1 :c :titulo
ignore abrevideo :c "m1p [0 titulo]
mci [play m1p wait]
fechevideo "m1p
fim
```

A possibilidade de introduzir sons, imagens e mesmo vídeos, permitiu a elaboração de várias versões do jogo de forca, inclusive na língua de sinais no SuperLogo (outro programa de linguagem logo que também aceita comando MCI).

Ou, melhor ainda:

```
para toquemidia :c
ignore mci ( frase "open :c [type mpegvideo alias m1p] )
mci [play m1p wait]
mci [close m1p]
fim
```

O procedimento **toquemidia** permite tocar vídeos (mpeg, avi, ...), e ainda música (mp3, midi, ...). Tudo isso e muito mais. Por exemplo, escreva agora na linha de comando: **toquemidia "bomba.mpg**. Note que o arquivo bomba.mpg deve estar salvo na mesma pasta que salvou o projeto do procedimento **toquemidia**. Do contrário, você terá que indicar todo o caminho:

```
toquemidia [d:\\Sons\\applause.wav]
```

(é só um exemplo, você deve saber qual o caminho em seu computador). Este último exemplo tocaria o arquivo applause.wav que está na pasta Sons de um cd (drive "d:"). As barras duplas são necessárias. Em fim, o procedimento **toquemidia** é mais abrangente do que o **toquemidia1**.

Exemplos genéricos:

```
toquemidia [minhamusica.mp3]
```

```
toquemidia [meuvideo.mpg ]
```

Digite os comandos (um de cada vez):

```
toquemidia [Opa.wav ]
```

```
toquemidia [dimono.mid]
```

```
toquemidia [Vid1.avi ]
```

Que fique bem claro: se o seu micro não consegue tocar uma dessas mídias, não será o Megalogo que irá consegui-lo.

Para tornar mais divertido, podemos fazer

```
toquewave escolhe [Dimono.mid Legal.wav Gmgreen3.mid]
```

A linha **mci [play m1p wait]** pode ser substituída por algo como:

```
mci [play m1p from 100 to 200 wait]
```

Nesse caso, tocará somente um trecho da mídia (da posição 100 até 200).

Façam Suas Apostas!

Que tal fazer uma fézinha? Nesse jogo você só tem a ganhar.

```
para adivinha
atr "loto sorteie 40
atr "palpite pri ( leiacaixa [Adivinhe que número pensei!] [Só vale de 0 a 39] )
se :palpite = :loto [es "parabéns] ~
[es [Que pena!] es [O número sorteado foi] es :loto]
fim
```

Não esqueça de usar o "pri" (primeiro) antes de leiacaixa para que o Megalogo entenda o número escrito.

A última linha do procedimento adivinha poderia ser também:

```
es ( frase [Que pena!] [O número sorteado foi] :loto )
```

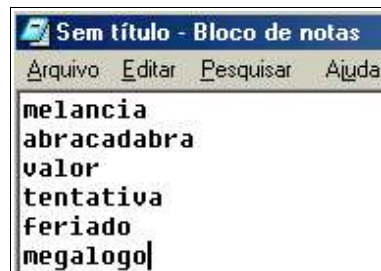
Banco de Dados

Crie uma lista de palavras no bloco de notas e salve com o nome “teste”. Lembre-se que, assim como no procedimento toquemídia, este arquivo deve ser salvo na mesma pasta que você salva seus projetos do Megalogo.

Veja o modelo na figura ao lado: uma palavra abaixo da outra.

Digite os comandos:

```
fixaparaleitura "teste.txt
es lêpalavra
[ melancia ]
es lêpalavra
[ abracadabra ]
fixaparaleitura [ ]
```



Quantas palavras tem nossa lista?

Vamos pedir para o Megalogo contar para nós.

<pre>para leitura fixaparaleitura "teste.txt atr "i 0 contar fixaparaleitura [] es :i fim</pre>	<pre>para contar es lêpalavra atr "i 1 + :i verifica fimleitura? sev [pare] sef [contar] fim</pre>
--	--

Uma forma mais elegante para o procedimento contar seria:

```
para contar
enquanto [não fimleitura?] [es lêpalavra atr "i 1 + :i]
fim
```

A idéia é a mesma só que mais objetiva: o arquivo teste.txt será lido até chegar a sua última linha.

Esse recurso é utilizado em jogos como da série Forca (que lêem banco de palavras para depois sortear) e no Asteroid Miner (para registrar os records).

Nosso exemplo pode ser sofisticado para fazer um jogo de adivinhações.

Como fazer um jogo de adivinhação de palavras?

Basta juntar essas idéias com a anterior do “Façam Suas Apostas!”.

<pre>para leitura fixaparaleitura "teste.txt atr "i 0 contar fixaparaleitura [] escolhepalavra fim</pre>	<pre>para contar es lêpalavra atr "i 1 + :i verifica fimleitura? sev [pare] sef [contar] fim</pre>
---	--

```
para escolhepalavra
atr "vezes 1 + sorteie :i
(local "pal "palpite)
fixaparaleitura "teste.txt
repita :vezes [atr "pal lêpalavra]
fixaparaleitura [ ]
atr "palpite pri ( leiacaixa [Adivinhe que palavra pensei!] [Adivinhe] )
se :palpite =:pal [es "parabéns ] ~
[ es [Que pena!] es [A palavra era ] es :pal ]
fim
```

A última linha do procedimento escolhepalavra poderia ser também [es [Que pena!] es (frase [A palavra era] :pal)].

Roda-Gigante

Algumas idéias sobre como trabalhar com mais de uma Tat.

```
para rodagigante
limpe ctat 1 [at] ctat 2 [at] ctat 3 [at]
pede [0 1 2 3] [fmf "verdadeiro fixaforma "icones fixafase 3]
pede 1 [repita 90 [gd 1 pf 1]]
pede 2 [repita 180 [gd 1 pf 1]]
pede 3 [repita 270 [gd 1 pf 1]]
pede [0 1 2 3] [repita 180 [pf 1 gd 1 espere 25] ]
fim
```

Porque não colocar uma musiquinha para a roda-gigante?

Usar o comando toquewave? Comandos mci?

Inserir um contador

Contadores são interessantes em jogos. Podem estimular o usuário e também servirem de controle de certas ações.

```
para contar
ctat "contador [at un] pede "contador [ fixapos [-420 250] ]
atr "contador 0 pede "contador [fixamodocaixa "verdadeiro]
; a variável e a "tartaruga caixa" devem ter o mesmo nome
; nesse exemplo elas receberam o nome de "contador"
fim
```

Em que contexto poderíamos inserir esse contador?

Há tantas possibilidades! Jogos para responder um cálculo ("Quanto é 2 + 3?"), clicar sobre a letra correta, ...

O jogo abaixo sorteia dois números de 0 a 10 e pede para o usuário responder quanto dará a soma dos dois:

```
para vai
( local "x" "y" "resposta )
atr "x sorteie 11 atr "y sorteie 11
atr "resposta pri ( lêcaixa ( frase [Quanto é :x "+" :y "?" ] [Calcule] )
se :resposta = :x + :y [acertou] [errou]
fim
```

```
para acertou
atr "contador 1 + :contador
caixa [Parabéns]
fim
```

```
para errou
caixa ( frase [O correto era:] :x + :y )
fim
```

Para jogar pela primeira vez, digita-se **contar** (para criar o contador marcando zero) e em seguida digita-se **vai** (que é a essência do jogo).

Note que o jogo em si é simples. Geralmente começa assim. Depois pode-se começar a pensar em incrementá-lo.

Uma idéia seria inserir uma personagem que fizesse uma carinha de triste para cada erro e mostrasse alegria para os acertos. Além da carinha, ela poderia dizer algo (usar som). É possível pedir participação das crianças, gravando suas vozes (gritos de alegria, tristeza, números que seriam pronunciados, ...).

Pega-pega

Que tal aproveitar as técnicas vistas até aqui e criar um jogo com certa inteligência (artificial?).

O objetivo do jogo é fazer com que uma tartaruga controlada pelo usuário seja perseguida por outra tartaruga controlada pelo computador.

Coisas para pensar antes de fazer o jogo:

- Que cenário usar?
- Como a tat do usuário será controlada?
- Como a tat do computador será controlada?
- Em que situações o jogo será encerrado?

Para os 4 elementos elencados acima, qual seria a sua ordem de prioridade? Ou seja, qual o elemento mais importante? E quais seriam os seguintes?

Abaixo, um possível caminho para começar a fazer o jogo:

```
para cenario
limpe ctat 1 [at]
pede [0 1] [un]
pede 0 [fixax -180 fixay -10 - sorteie 80 fixaforma "borbole]
pede 1 [fixax sorteie 80 fixay sorteie 80 fixaforma "setas_7]
carregagráficos "ceu.bmp
caixa [Ajude a borboleta a fugir do foguete]
caixa [Use as setas no teclado para mover a borboleta]
; marca quando iniciou o jogo ( em segundos ) :
atr "inicio 60 * ( elem 2 hora ) + elem 3 hora
; acrescenta 60 ao tempo inicial do jogo.
; 60 segundos será o tempo que vai durar o jogo
atr "inicio 60 + :inicio
fim
```

```
para andar
atr "ABC leiatecla
; leiatecla faz a tat ler a tecla pressionada
se :ABC = -72 [pf 1] ; seta para cima
se :ABC = -80 [pt 1] ; seta para baixo
se :ABC = -77 [gd 45] ; seta para direita
se :ABC = -75 [ge 45] ; seta para esquerda
fim
```

```

para ondesta
pede 0 [andar atr "xy0 arredonde pos]
pede 1 [atr "xy1 arredonde pos fixarumo direção :xy0 pf 1]
; deve - se arredondar xy0 e xy1 ou o logo não vai perceber que
; elas estão próximas, por exemplo, em posições [121,1 0] e [121,15 0]
;
se :ABC = 27 [pare] ; tecla ESC interrompe o jogo
se :ABC = -79 [pare] ; tecla END interrompe o jogo
; Verifica se terminou o tempo de jogo
atr "tempo 60 * ( elem 2 hora ) + elem 3 hora
se :tempo = :inicio [pare]
verifica igual? :xy1 :xy0
sev [ es [foi pega] pare ]
sef [ ondesta ]
fim

```

Uma estratégia usada no ensino da linguagem Logo é apresentar projetos incompletos ou com erros para que o aprendiz faça a depuração.

Após testar os procedimentos acima e depurá-los, cheguei à seguinte solução (que não é a única, nem a melhor possível):

```

para pegapega
cenario
; marca quando iniciou o jogo (em segundos) :
atr "inicio 60 * ( elem 2 hora ) + elem 3 hora
; 20 segundos será o tempo que vai durar o jogo
; então, acrescento 20 ao tempo inicial do jogo.
atr "inicio 20 + :inicio
andar ondesta
se :tempo > :inicio [caixa [Parabéns] ]
fim

```

```

para cenario
limpe ctat "foguete [at un]
ctat "borbo [at un]
pede "borbo [fixax -180 fixay -10 - sorteie 80 fixaforma "borbole]
pede "foguete [fixax sorteie 80 fixay sorteie 80 fixaforma "setas_7]
carregagráficos "ceu.bmp
caixa [Ajude a borboleta a fugir do foguete]
caixa [Use as setas no teclado para mover a borboleta]
fim

```

```

para ondesta
pede "foguete [atr "xyF arredonde pos fixarumo direção :xyB pf 0.1]
pede "borbo [atr "xyB arredonde pos]
se :ABC = 27 [pare] ; tecla ESC interrompe o jogo
se :ABC = -79 [pare] ; tecla END interrompe o jogo
; Verifica se terminou o tempo de jogo
atr "tempo 60 * ( elem 2 hora ) + elem 3 hora
verifica igual? :xyF :xyB
sev [ caixa [foi pega] pare ]
;
se :tempo > :inicio [pare]
se tecla? [pede "borbo [andar] ]
ondesta
fim

```

A tat "foguete deve andar mais devagar (pf 0.1, por exemplo) para que o usuário tenha tempo de pensar e agir.

```

para andar
atr "ABC leiatecla
; leiatecla faz a tat ler a tecla pressionada
se :ABC = -72 [pf 10] ; seta para cima
se :ABC = -80 [pt 10] ; seta para baixo
se :ABC = -77 [gd 45] ; seta para direita
se :ABC = -75 [ge 45] ; seta para esquerda
fim

```

Esse foi o pontapé inicial para um projeto que pode se tornar bem mais complexo, sofisticado (que ajustes seriam necessários?). Poderia ser dada uma mensagem ou emitir um som quando uma tartaruga pegar a outra; proibir que as tartarugas voadoras caiam num telhado ou usá-lo como "raia"; limitar o território do jogo ao cenário (não poder sair de cima dele), etc. Um projeto em logo não necessariamente acaba. Hoje podemos estar satisfeitos com o que fizemos ou paramos por aqui por não termos idéia de como prosseguir, talvez precisemos estudar mais técnicas ou a própria linguagem logo... Mais tarde, podemos retomá-lo ou dar-lhe um novo rumo. É isso que precisamos assimilar na linguagem logo: saber saborear as idéias.

A reprodução deste material (todo ou em parte) é permitida e desejada desde que citada a fonte. Proibida distribuição deste material visando lucro de qualquer espécie (contate o autor antes).

Todas as atividades criadas por Alexandre R. Soares.