

Introdução ao StarLogo

1. O programa StarLogo

1.1. O que é o StarLogo?

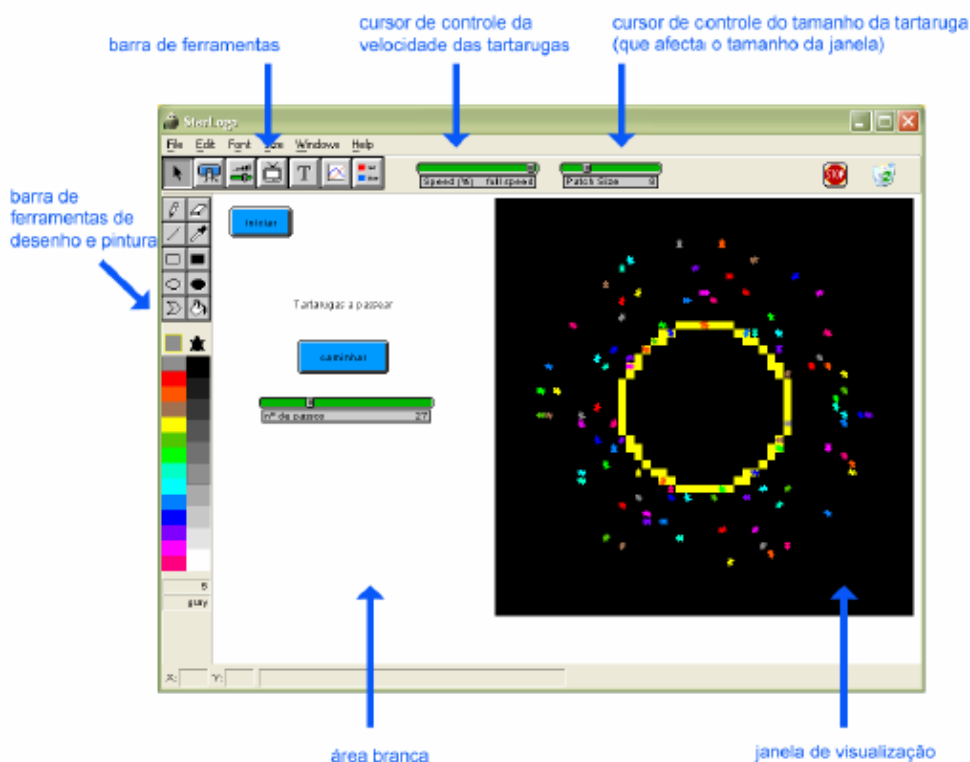
StarLogo é um programa que permite criar desenhos e animações através de comandos dados a uma tartaruga (turtle) no ecrã do computador.

Com a linguagem Logo podemos construir figuras geométricas por meio de um conjunto simples de regras que definem a chamada «Geometria da tartaruga» e o StarLogo amplia as possibilidades do Logo, permitindo trabalhar simultaneamente com muitas tartarugas (milhares!) e atribuindo-lhes propriedades e comportamentos diferenciados. O conjunto de instruções foi também consideravelmente ampliado. Em particular, o StarLogo permite programar simulações com alguma facilidade. Há comandos (instruções) que já estão definidos e outros que podemos acrescentar, usando os chamados *procedimentos*.

1.2. Interface

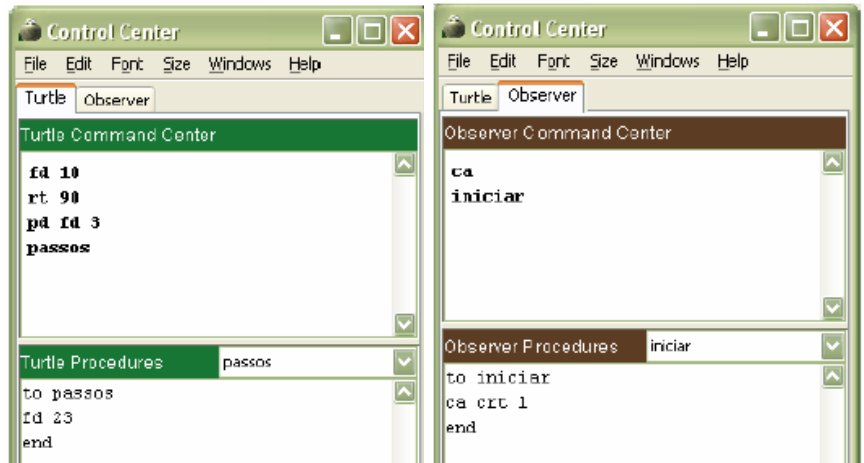
Aberto o programa, surgem 2 janelas: a Janela StarLogo e o Centro de Controle.

Janela StarLogo



Centro de Controle

Janela onde o utilizador executa os comandos e/ou constrói os procedimentos. Clicando em «Turtle» (= «tartaruga») temos uma janela com barras verdes (imagem da esquerda) onde são escritos os comandos das tartarugas. Clicando em «Observer» (= «observador») temos uma janela com barras castanhas onde são escritos os comandos do observador.



2. Exemplos e sugestões de projectos

Seguem-se exemplos de projectos com o StarLogo. Para os concretizar basta seguir as instruções dadas. Caso tenha dúvidas acerca dos comandos utilizados, consulte o anexo «Comandos».

Exemplo 1 - Construir um segmento de comprimento 10 unidades.

Pode ver uma tartaruga no centro da Janela de Visualização. Na verdade, não é só uma que lá está, mas muitas, umas em cima das outras. Precisamos apenas de uma tartaruga. Vamos apagar as que lá existem e criar uma única que ficará virada para cima:

a) No **Centro de controle** clique em **Observer** e na área **Observer Command Center** execute:

`ca crt 1`

Nota: São apagadas todas as tartarugas (`ca` = *clear all* = apaga tudo) e no centro da **Janela de Visualização** é criada uma única tartaruga (`crt 1` = *create 1* = cria 1 tartaruga).

A tartaruga traz uma caneta que pode baixar, deixando um traço por onde passa, ou levantar, deslocando-se sem deixar rasto. Para desenhar o segmento tem de baixar a caneta e deslocar a tartaruga.

b) No **Centro de controle** clique em **Turtle** e na área **Turtle Command Center** execute:

`pd fd 10`

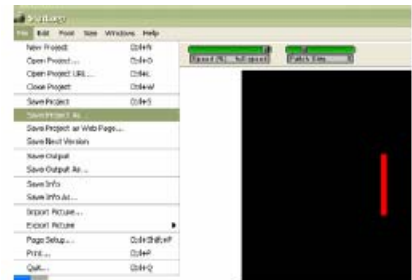
Nota: A tartaruga baixa a sua caneta (`pd` = *pen down* = caneta para baixo) e dá 10 passos (`fd 10` = *forward 10* = para a frente 10).

Tem assim o segmento pretendido.



c) Grave na sua pasta o exemplo que acabou de construir: no menu *File*, seleccione *Save Project As...* e chame-lhe *exemplo1*.

Se quiser guardar apenas a imagem, poderá fazer *Export Picture*, *Interface Window* e depois *exemplo1*.



Exemplo 2 - Construir um quadrado de comprimento 10 unidades.

Vamos agora desenhar um quadrado, partindo do exemplo anterior.

a) Grave o mesmo projecto, mas com outro nome:

b) No **Centro de controle** clique em **Turtle** e na área **Turtle Command Center** execute:

`rt 90 fd 10`

Nota 1: A tartaruga roda 90° para a direita (`rt 90 = right turn 90° = roda 90° para a direita`) e dá 10 passos em frente.



Só lhe resta desenhar mais 2 lados.

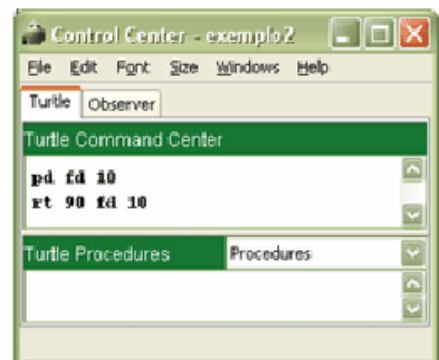
c) Execute a última linha de comandos

`rt 90 fd 10`

duas vezes. Tem assim o quadrado pretendido.



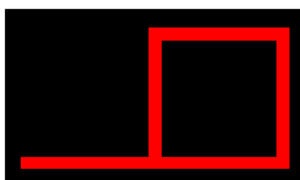
d) Grave o seu ficheiro.



Algumas sugestões de trabalho

Construa projectos que permitam criar as seguintes figuras:

S1



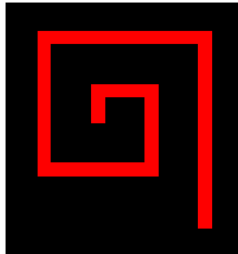
Tanto o segmento como o lado do quadrado medem 10 unidades

S2



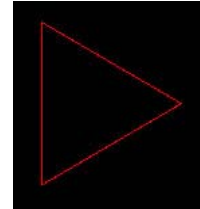
Rectângulo 20 X 10 unidades

S3



Os comprimentos dos segmentos são 2, 4, 6, 8, 10, 12 e 14 unid.

S4



Triângulo equilátero

Comece por alterar o *Patch Size* para 1 e aumentar a Janela de Visualização (ver observação abaixo)

Observação – Como tornar os traços mais finos...

O traço deixado pelo rasto da tartaruga é bastante grosso. Podemos torná-lo mais fino diminuindo o tamanho da tartaruga. Para isso basta diminuir o valor do **Patch Size** com o cursor.



Diminuindo o *Patch Size*, a Janela de Visualização também diminui. No entanto, podemos aumentá-la (sem aumentar o tamanho da tartaruga) do seguinte modo:

- seleccione a Janela de Visualização (clique sobre a janela, enquanto prime a tecla Ctrl). Confirme que está seleccionada: em cada vértice da janela devem aparecer uns pequenos quadrados.

- arraste com o rato um desses quadrados. Se quiser que a Janela de Visualização seja quadrada, carregue na tecla Ctrl enquanto arrasta o rato.

Exemplo 3 - Construir dois quadrados de comprimento 10 unidades e de cores diferentes.

O objectivo é construir a figura ao lado.

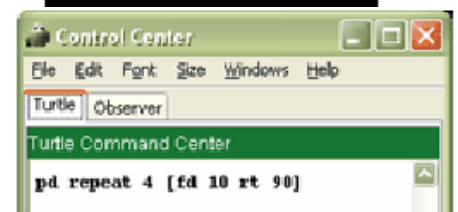
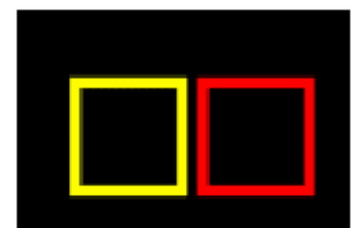
Abra um novo projecto e crie 1 tartaruga. Vamos agora desenhar um quadrado mas de forma mais eficiente:

a) Comece por baixar a caneta. A seguir, em vez de seguir os passos b) e c) do exemplo anterior, execute:

```
pd repeat 4 [fd 10 rt 90]
```

Nota: A tartaruga baixa a caneta. A seguir, repete 4 vezes a seguinte acção: «dar 10 passos e rodar 90° para a direita»

(repeat 4 [lista de comandos] = repete 4 vezes os comandos da lista dentro de []).



Vamos agora mudar a cor da tartaruga para amarelo (*yellow*).

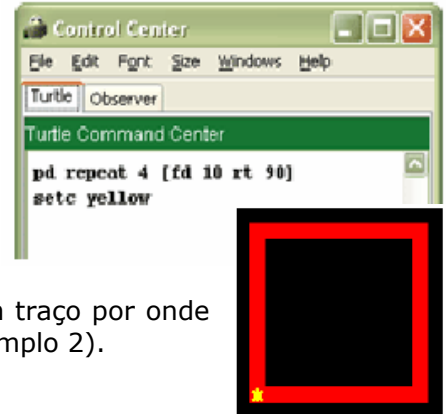
b) Execute:

```
setc yellow
```

Nota: A cor do traço desenhado pela tartaruga é a sua própria cor. Podemos, com o comando **setc** (= *set color* = escolher cor) alterar a cor da tartaruga.

Como pode observar, a tartaruga está voltada para cima, exactamente no mesmo sítio de onde partiu.

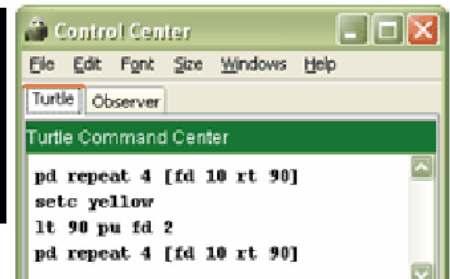
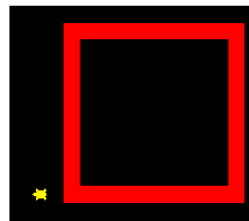
Para desenhar o quadrado amarelo, rode a tartaruga 90° para a esquerda, levante a caneta (para que não deixe um traço por onde passa) e faça com que a tartaruga ande alguns passos (por exemplo 2).



c) Execute:

```
lt 90 pu fd 2
```

Nota: A tartaruga roda 90 graus para a esquerda (*lt 90* = *left turn 90°* = roda 90° para a esquerda), levanta a caneta (*pu* = *pen up* = caneta para cima) e dá 2 passos.



A tartaruga já não está sobre o primeiro quadrado e pode então desenhar o segundo.

d) Execute:

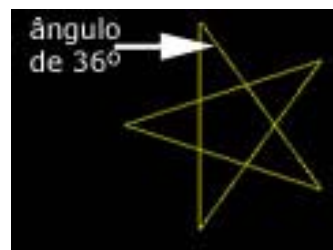
```
pd repeat 4 [fd 10 rt 90]
```

A tartaruga desenha um novo quadrado.

Sugestão de trabalho

Construa o seguinte projecto

S5



Pentagrama, lado=80 unidades

Utilizando o comando *repeat* construa o pentagrama (amarelo). Comece por alterar o *Patch Size* para 1 e aumentar o tamanho da Janela de Visualização.

Exemplo 4 – Construir um procedimento que faça com que a tartaruga desenhe um quadrado de lado 10 unidades.

Os comandos do StarLogo dividem-se, basicamente, em comandos primitivos, que já vêm implementados na linguagem, e em outros que são construídos a partir de **procedimentos**, escritos pelo utilizador e que podem ser executados como os comandos primitivos.

Imagine que quer desenhar vários quadrados. Se definir um **procedimento** que desenhe um quadrado de lado 10, evita repetir, para cada quadrado, toda a linha de comandos.

Depois de abrir um novo projecto e de criar 1 tartaruga, vamos definir um procedimento que desenhe um quadrado de lado 10.

Para definir um procedimento precisa de 2 novos comandos: **to <nome>**, que indica o princípio do procedimento e **end** para indicar o fim.

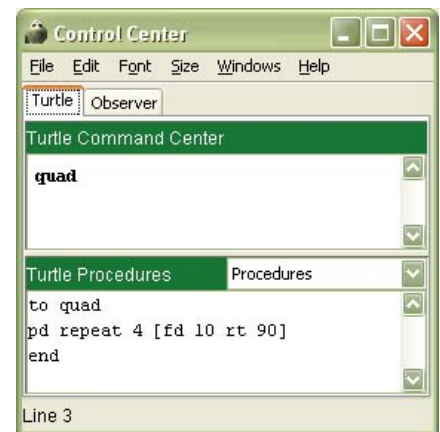
a) Escreva o procedimento no **Centro de controle**, na área da tartaruga, em *Turtle Procedures*:

`to quad` -> *Inicia o procedimento a que vamos chamar quad.*

`pd repeat 4 [fd 10 rt 90]` -> *Comandos a serem executados.*

`end` -> *Finaliza o procedimento.*

Depois de definido o procedimento, se escrever a palavra `quad` no **Centro de controle** na área da tartaruga (em *Turtle Command Center*) e se clicar na tecla enter, a tartaruga executará todos os comandos descritos no procedimento `quad`.



Algumas sugestões de trabalho

Utilizando a definição de **procedimentos** construa projectos que permitam criar as figuras abaixo.

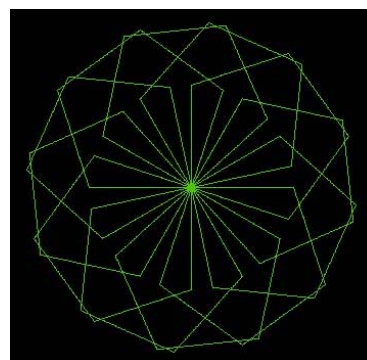
S6



Pentágono regular, lado=50 unidades

Nota: O *Patch Size* deve ter valor 1

S7



Utilizando o procedimento que escreveu em S6, construa a imagem acima.

Nota: esta imagem pode ser vista como uma reunião de 12 pentágonos regulares, cada um rodado do anterior por um ângulo de 30° (em torno do centro da imagem).

Também é possível definir **procedimentos** com variáveis:

Exemplo 5 - Definir um procedimento que faça com que a tartaruga desenhe um quadrado de qualquer lado

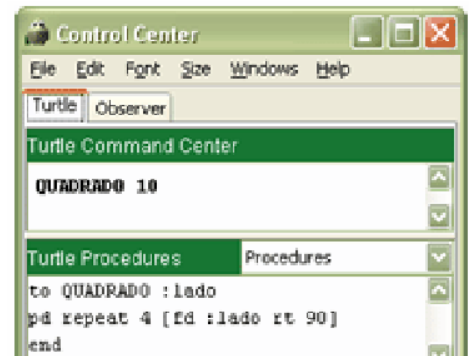
Depois de abrir um novo projecto e de criar 1 tartaruga, vamos definir um procedimento que desenhe um quadrado de qualquer lado. Chamemos **QUADRADO** a este procedimento.

Este procedimento é idêntico ao definido no Exemplo anterior, mas agora o nome do procedimento está acompanhado da variável `:lado` (é necessário colocar 2 pontos antes do nome da variável). Note-se ainda que sempre que a variável apareça ao longo do procedimento também deve ser precedida pelos 2 pontos.

Assim, na área da tartaruga no **Centro de controle**, em *Turtle Procedures*, escreva:

```
to QUADRADO :lado
pd repeat 4 [fd :lado rt 90]
end
```

Podemos agora fazer algumas experiências: execute na área da tartaruga no **Centro de controle** (em *Turtle Command Center*) **QUADRADO 10**. Experimente **QUADRADO 5**, **QUADRADO 3**, **QUADRADO 12**, etc. Verifique que a tartaruga desenha um quadrado de lado 10, 5, 3, 12, ..., ou qualquer outro número que escolha.



Algumas sugestões de trabalho

Utilizando a definição de procedimento com variáveis construa os seguintes projectos:

S8:

a) Defina um procedimento **poligono** com variável **n** que desenhe um polígono regular de **n** lados. Nota: O comprimento de cada lado deverá ser 50 e o *Patch Size* deverá ter valor 1.

Execute o procedimento definido em S8, para vários valores de **n**: 3, 5, 7, 10, ...

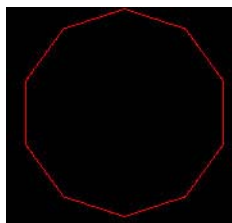


Imagem obtida pela execução de **poligono 10**

b) Defina um procedimento análogo, mas com perímetro (e não lado) constante (= 500)

S9: Utilizando o procedimento que escreveu em **S8 a)**, defina um procedimento **rosacea** com variável **n** que desenhe uma rosácea constituída por 12 polígonos regulares de **n** lados (de comprimento 50), cada um rodado do anterior 30° em torno do centro da imagem, (este procedimento é análogo ao da sugestão **S7**, com a hipótese de escolha do polígono "base").

Execute o procedimento definido em S9, para vários valores de **n**: 3, 5, 7, 10, ...

Associação Atrator

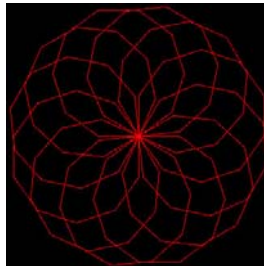


Imagem obtida pela execução de **rosacea 10**

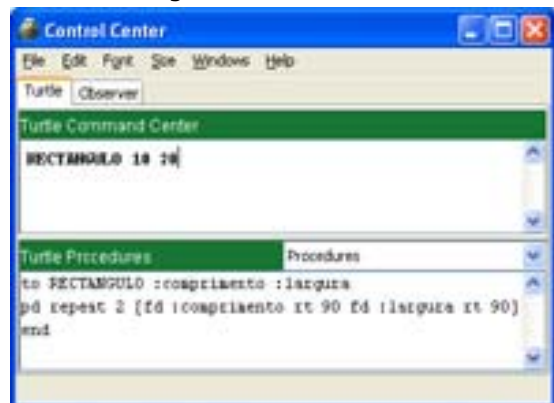
Exemplo 6 – Definir um procedimento que faça com que a tartaruga desene um rectângulo comprimento e largura arbitrários (utilização de duas variáveis).

As tarefas a realizar serão muito parecidas com as do exemplo anterior, mas agora utilizaremos duas variáveis: o comprimento e a largura.

a) Abra um novo projecto, apague tudo e crie uma tartaruga.

b) No **Centro de controle da tartaruga**, defina o procedimento RECTANGULO:

```
to RECTANGULO :comprimento :largura
pd repeat 2 [fd :comprimento rt 90 fd :largura rt 90]
end
```




Faça algumas experiências: execute RECTANGULO 10 20 . Experimente ainda RECTANGULO 20 10 , RECTANGULO 15 15 , RECTANGULO 5 20 , ..., ou qualquer outro par de números.

Exemplo 7 – Criar um *applet* que faça com que a tartaruga desene um rectângulo de qualquer comprimento e largura. Estas medidas são controladas por cursores.

O StarLogo permite criar ambientes de fácil manuseamento, acessível a qualquer tipo de utilizador, mesmo àqueles que não sabem programar. Estas interfaces são constituídas por botões que executam conjuntos de comandos e cursores que permitem atribuir valores a variáveis.

Começaremos por criar o botão "iniciar", que recomeça o projecto. Para isso,



1) Clique em , que está na barra de ferramentas na parte superior da Janela StarLogo. Depois clique na parte branca, numa zona vazia da mesma janela.

Abrir-se-á uma janela como a do lado.



2) Para poder dar um nome ao botão, no rectângulo branco por baixo de «Name», apague a palavra «button1» e escreva a palavra «iniciar».

3) No rectângulo branco por baixo de «StartLogo Instruction», escreva:

ca crt 1

4) No rectângulo branco por baixo de «Tooltip», escreva: «Apaga tudo e cria uma tartaruga.»

Nota: Esta última frase (opcional) é uma descrição do que o botão irá fazer.

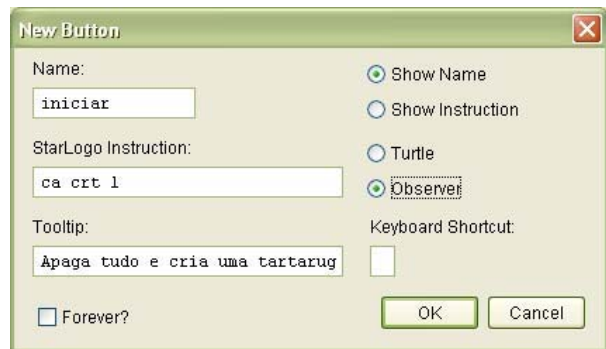
5) Seleccione a opção «Observer»

Nota: as instruções «ca crt 1» são do Observador (e, por isso, tem de escolher a opção «Observer»).

6) Seleccione a opção «Show Name» (=«Mostra Nome»).

Nota: esta instrução faz com que o nome do botão fique visível.


Deve obter o que está na figura ao lado.



7) Carregue em OK. Na janela StarLogo deve aparecer este botão:

Vamos agora criar o botão «rectangulo», que desenhará um rectângulo de qualquer comprimento e largura:



8) Clique em . Depois clique na parte branca, numa zona vazia da mesma janela. Abrir-se-á uma nova janela.

9) Chame a este botão «rectangulo». Para isso, no rectângulo branco por baixo de «Name», apague a palavra «button2» e escreva a palavra «rectangulo».

10) Em «StartLogo Instruction», vamos utilizar o procedimento RECTANGULO que definiu anteriormente.

Portanto, no rectângulo branco por baixo de «StartLogo Instruction», escreva:

RECTANGULO comprimento largura.

11) No rectângulo branco por baixo de «Tooltip», escreva, por exemplo:

Rect. comprimento x largura

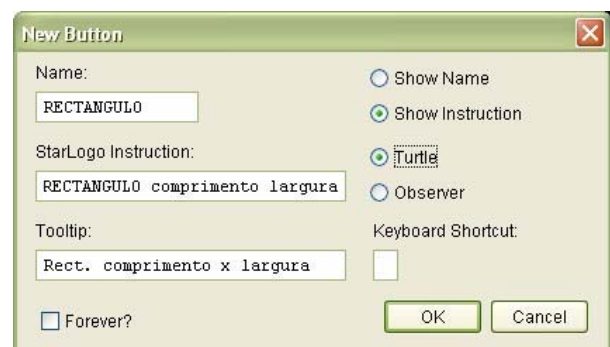
ou escreva outra frase que ache mais apropriada.

Deve obter o que está na figura ao lado.

12) Seleccione a opção «Show Name»

13) Seleccione a opção «Turtle».

Nota: a instrução «rectangulo» é uma instrução da tartaruga.





14) Carregue em OK. Na janela StarLogo deve aparecer o botão ; neste momento, o rectângulo ainda não está completamente definido. É necessário atribuir valores a «comprimento» e «largura». Vamos começar por atribuir um valor ao comprimento, utilizando um cursor:



15) Clique em , que está na barra de ferramentas na parte superior da Janela StarLogo. Depois clique na parte branca, numa zona vazia da mesma janela. Abrir-se-á uma janela como a do lado.

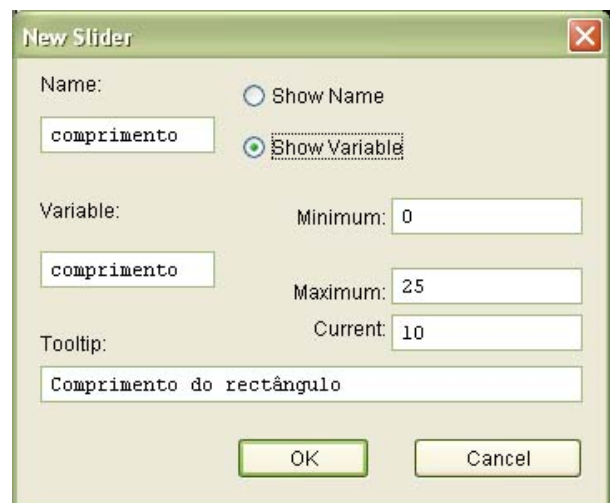
16) Em *Variable* escreva a palavra «comprimento».

17) Dê o valor 0 ao mínimo (*Minimum*), 25 ao máximo (*Maximum*) e 10 ao valor inicial (*Current*) de «comprimento».

18) Tal como nos botões, podemos dar um nome ao cursor. Deve escrevê-lo em "Name". Se quiser que esse nome fique visível deve seleccionar «*Show Name*».

19) Tal como nos botões, nos cursores também podemos colocar uma descrição em «*Tooltip*».

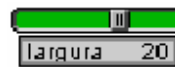
Deve obter algo parecido com o que está na figura ao lado.



20) Carregue em OK. Na janela StarLogo deve aparecer o seguinte cursor:



Falta atribuir um valor à largura. À semelhança do que fez anteriormente construa então o cursor "largura". Sugerimos-lhe que atribua 0 ao valor mínimo, 25 ao máximo e 20 ao valor inicial.



21) Na janela StarLogo deve aparecer este cursor:

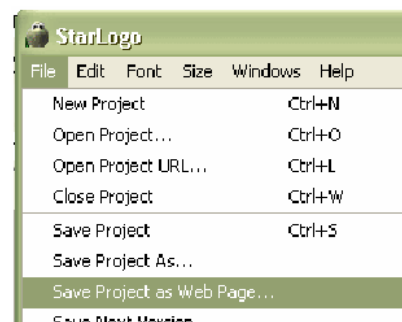
22) Teste os botões e os dois cursores que criou na janela StarLogo: clique em cada um deles e veja o que acontece...

Agora, grave um *applet* com o StarLogo. Para construir um *applet*:

23) Na janela StarLogo clique em *File* e depois em *Save Project as Web Page...*

24) Grave o seu projecto na pasta com o nome que desejar.

Fica assim criado um *applet*. O *applet* permite que o projecto fique visível numa página da Internet, logo que alojado num site e desde que o ficheiro *starlogo.jar* esteja presente.

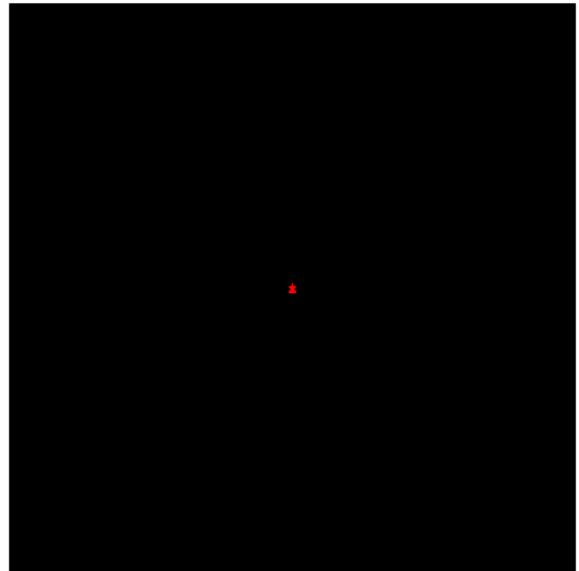
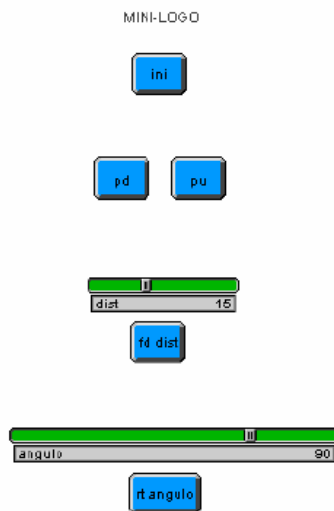


25) Abra, com um browser (internet explorer ou outro), a página que acabou de criar e teste o *applet* que construiu.

Sugestões de trabalho

S10: Crie um "Mini-Logo".

Construa um *applet* que permita executar as acções mais simples do StarLogo – apagar tudo e criar uma tartaruga, baixar a caneta, levantar a caneta, andar um número de passos e rodar um determinado ângulo - sem que seja necessário conhecer os comandos da linguagem. Para isso, crie botões e cursores que executem tarefas e permitam a escolha de valores.



Teste o *applet* que construiu.

S11:

- Reduza o *Patch Size* para 1 e aumente a Janela de Visualização
- Crie um procedimento **poligono :n :comp** que permita desenhar um polígono regular de **n** lados de comprimento **comp**.

Nota: basta efectuar as devidas alterações no procedimento definido em **S8 a)**

- Inclua dois cursores que permitam ao utilizador escolher os valores das variáveis **n** e **comp**.

c) Crie dois botões – **iniciar** e **poligono**. O botão **iniciar** recomeça o projecto e o botão **poligono** constrói um polígono regular com **n** lados de comprimento **comp**, sendo os valores de **n** e **comp** dados pelos cursores correspondentes.

d) Crie um *applet* com o projecto que acabou de construir. Teste-o.

S12:

- Reduza o *Patch Size* para 1 e aumente a Janela de Visualização.
- Crie um procedimento **circ** que desenhe uma "circunferência" de raio 50 e centro no ponto onde se encontra a tartaruga. No final a tartaruga deve reocupar a sua posição inicial virada no mesmo sentido (inicial).

NOTA: Na realidade, a curva traçada não é uma circunferência, mas sim um polígono regular de muitos lados de pequeno comprimento...

c) Utilizando o procedimento anterior, construa um procedimento **poli-circ :n** que permita desenhar a circunferência acima referida e um polígono regular de **n** lados inscrito nessa circunferência.



Imagem obtida pela execução de **circ**

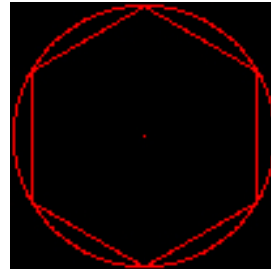



Imagem obtida pela execução de **poli-circ 6**

d) Inclua um cursor que controle o número **n** de lados do polígono. Crie dois botões – **iniciar** e **poli-circ** - o botão **iniciar** recomeça o projecto e o botão **poli-circ** executa o procedimento **poli-circ :n** .

e) Crie uma janela de *Output* que indique ao utilizador o perímetro do polígono inscrito na referida circunferência. Para tal, clique em , que está na barra de ferramentas na parte superior da Janela StarLogo. Depois clique na parte branca, numa zona vazia da mesma janela e siga as instruções.

Exemplo 8 – Criar um projecto onde seja definido um procedimento que desenhe a espiral de S3 usando iteração

a) Abra um novo projecto, apague tudo e crie uma tartaruga.

b) Na área da tartaruga, no **Centro de controle**, defina o procedimento ESPIRAL:

```
to ESPIRAL
```

let [:s 2] -> Define uma variável local ("interna" ao procedimento), sendo 2 o primeiro valor que a variável "s" toma.

```
pd
```

```
repeat 7 [ fd :s rt 90 let [:s :s + 2]] -> ver NOTA abaixo.
```

```
end
```

NOTA: Repete 7 vezes o conjunto de comandos: «a tartaruga dá s passos, roda 90° e a variável s é aumentada de 2 unidades».

Se executar **ESPIRAL** deve obter a imagem apresentada em S3

Observação: Na verdade podemos definir o procedimento com uma variável que permita obter um número qualquer de iterações e outra que determine o acréscimo de cada segmento da espiral. Bastam umas ligeiras alterações no procedimento definido acima:

```
to ESPIRAL :k :s0
let [:s :s0]
pd
repeat :k [ fd :s rt 90 let [:s :s + :s0]]
end
```

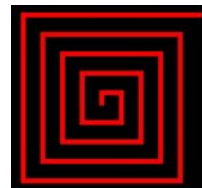


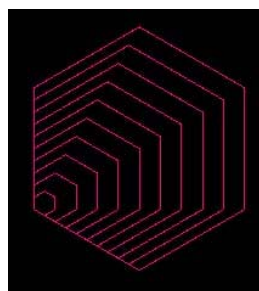
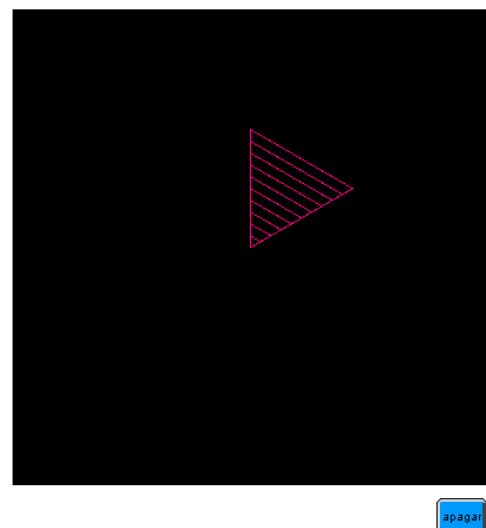
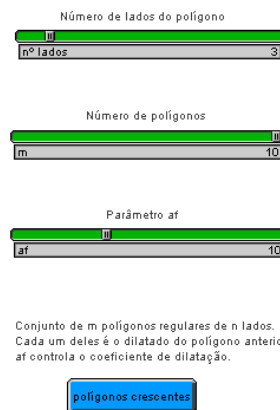
Imagem obtida pela execução ESPIRAL 10 2

c) Crie botões e cursores, e grave um *applet* com o projecto que acabou de realizar.

Sugestões de trabalho

S13: Construa um *applet* com o qual seja possível desenhar m polígonos regulares de n lados, cada um homotético do primeiro, relativamente a um vértice comum a todos, e com coeficientes de homotetia af , $2af$, $3af$, Os valores m , n e af devem poder ser controlados através de cursores.

Comece por diminuir o *Patch Size* para 1 e aumentar o tamanho da Janela de Visualização.

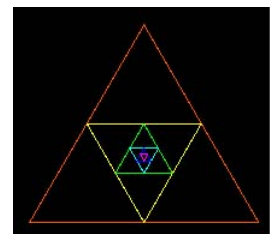


10 hexágonos homotéticos

S14 Crie um procedimento que construa N triângulos equiláteros por iteração como na figura ao lado.

Os triângulos devem ter cores diferentes (ver **Nota**)

O triângulo maior deve ser o primeiro a ser construído. O segundo triângulo tem os vértices nos pontos médios dos lados do primeiro triângulo. A construção dos restantes triângulos deve seguir a mesma regra.



Associação Atrator

Devem ainda existir dois cursores e dois botões:

- um botão que apaga tudo e cria 1 tartaruga;
- um cursor para controlar o número N de triângulos a serem construídos;
- um cursor que permite a escolha do comprimento do lado do triângulo maior;
- um botão que cria a imagem propriamente dita.

Nota: As cores podem ser identificadas por números:

Preto	0	Cinzento	5
Branco	9	Vermelho	15
Cor-de-laranja	25	Castanho	35
Amarelo	45	Verde	55
Verde limão	65	Turquesa	75
Cyan	85	Azul céu	95
Azul	105	Violeta	115
Magenta	125	Cor-de-rosa	135

Todos os projectos apresentados e sugeridos envolviam uma única tartaruga. Sugerimos-lhe, agora, que construa uma pequena simulação com mais do que uma tartaruga:

Exemplo 9 – Criar uma simulação:

Nesta simulação pretende-se que existam dois grupos de tartarugas distinguidos pela cor, que representam as “presas” e os “predadores”. O número de elementos de cada grupo é controlado por um cursor.

Todas as tartarugas passeiam aleatoriamente pela Janela de Visualização e os “predadores” “comem” as “presas” sempre que as encontram.

Pretende-se ainda que o programa permita ao utilizador estar permanentemente informado do número de presas patentes na Janela de Visualização.

a) Abra um novo projecto e apague tudo.

O StarLogo permite criar diferentes “espécies” de tartarugas, utilizando o comando **breeds**. Neste exemplo, utilizaremos esta potencialidade do StarLogo, distinguindo duas “espécies” – as presas e os predadores. Para isso,

b) em Turtle Procedures, escreva:

```
breeds [presas predadores]
```

O StarLogo reconhece, agora, a existência de dois tipos distintos de tartarugas.

Em seguida, vamos criar um procedimento **ini :n :m** que apague tudo, crie **n** presas e **m** predadores, que os coloque aleatoriamente no ecrã, mude a cor dos predadores para vermelho e o das presas para amarelo.

Dado que, para apagar tudo, vamos utilizar o comando **ca** (=clear all), que é um comando de Observador, o procedimento que se segue terá de ser escrito em **Observer Procedure**. Assim,

c) em Observer Procedures, escreva:

```
to ini :n :m
ca
```

Vamos agora criar as **n** presas e os **m** predadores. Para tal, utilizaremos o comando primitivo do StarLogo **create-breed k**, que cria k tartarugas da espécie "breed".

d) Escreva:

```
create-presas :n
create-predadores :m
```

Seguidamente, "espalharemos" as tartarugas no ecrã. Para tal, utilizaremos o comando primitivo do StarLogo **random n**, que fornece um valor aleatório entre 0 e n-1.

Vamos agora utilizar comandos da tartaruga (rt, fd, ...) e há um problema, pois estamos a trabalhar na área do observador, que não reconhece (directamente) comandos da tartaruga. Resolve-se este problema, invocando o comando do StarLogo **ask-turtles**, que permite, por assim dizer, chamar um comando da tartaruga a partir da janela dos comandos do observador.

e) Escreva:

```
ask-turtles [
```

rt random 360 jump random 35 -> as tartarugas rodam um ângulo arbitrário (entre 0º e 359º) e "saltam" (jump) um número arbitrário de passos (inferior a 35), ficando elas espalhadas no ecrã.

Por fim, atribuiremos a cor vermelha aos predadores e amarelo às presas. Para tal, utilizaremos o comando **if condição [comando]**, que verifica se **condição** é verdadeira e, nesse caso, executa **comando**.

f) Escreva ainda:

if breed = predadores [setc red] -> atribui cor vermelha às tartarugas da espécie predador

if breed = presas [setc yellow] -> atribui cor amarela às tartarugas da espécie presa

] -> fecha os parêntesis abertos em "ask-turtles"

```
end
```

Deverá obter o que está na figura ao lado.

g) Crie dois cursores relativos às variáveis **n** e **m**.

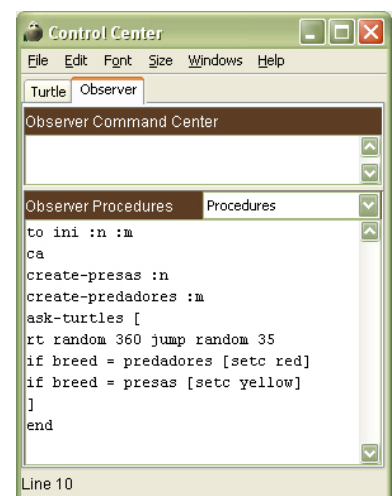
h) Crie um botão **ini** que execute o procedimento que definiu e teste-o.

Vamos agora construir um procedimento **avancar** que faça com que todas as tartarugas se desloquem aleatoriamente e de tal modo que os predadores "comam" as presas sempre que as encontrem.

i) Em Turtle Procedures, escreva (por baixo da instrução **breeds [presas predadores]**):

```
to avancar
```

rt random 360 fd 1 -> as tartarugas rodam um ângulo arbitrário (entre 0º e 359º) e deslocam-se um passo, caminhando assim de uma forma "aleatória".



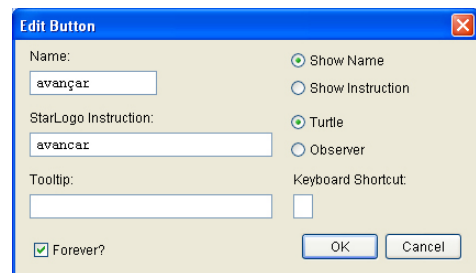
j) Escreva:

```
if breed = presas [if count-predadores-here > 0 [die] ]
end
```

Assim, se a tartaruga que o StarLogo estiver a analisar for uma presa (`if breed = presas`), o StarLogo irá contar o número de predadores que se encontram sobre esta tartaruga (utilizando o comando primitivo do StarLogo `count-predadores-here`) e, no caso de haver pelo menos um predador nesta situação (`if count-predadores-here > 0`), a tartaruga em causa é "morta" (utilizando o comando primitivo do StarLogo `die`), isto é, é apagada do StarLogo.


l) Crie um botão **avancar** que execute o procedimento **avancar**.

Nota: Ao introduzir os dados do botão, seleccione a opção "Forever" (ver imagem ao lado). Esta opção permite que o movimento das tartarugas continue indefinidamente.

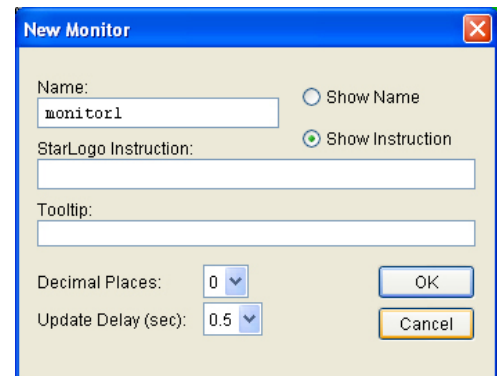


Vamos agora criar uma janela de *output* que permita ao utilizador estar permanentemente informado do número de presas presentes na Janela de Visualização.



m) Clique em , que está na barra de ferramentas na parte superior da Janela StarLogo. Depois clique na parte branca, numa zona vazia da mesma janela.

Abrir-se-á uma janela como a do lado.



n) Para poder dar um nome à janela, apague a palavra «*monitor1*» e escreva a palavra «nº de presas».

o) Em «*StarLogo Instruction*», escreva:

`count-presas` -> a instrução **count-breed** indica o número de tartarugas da espécie **breed**.

p) Tal como nos botões, também podemos colocar uma descrição em «*Tooltip*».

q) Se quiser que o nome da janela de output fique visível, seleccione «*Show Name*».

Deve obter algo como o que está na figura ao lado.



Apêndice 1 – Comandos

Existem comandos que são específicos das tartarugas, do observador ou das manchas.

Comando de observador

Comando de tartarugas

Comando de manchas (*patches*)

Outros há que são comuns a todos ou a algum par deles:

Comando de observador, tartarugas e manchas

Comando de observador e tartarugas

Comando de observador e manchas

Comando de tartarugas e manchas

Os **comandos de observador** têm de ser escritos na área do observador.

Os **comandos de tartarugas** podem ser escritos directamente na área das tartarugas ou chamados a partir da área do observador (com o comando ask-turtles ou variantes deste).

Os **comandos de manchas** podem ser chamados de qualquer umas das áreas (usando comandos como ask-patches ou variantes deste).

Abaixo estão alguns exemplos de comandos do StarLogo.

Comando	Comando de...	Efeito produzido	Abreviatura de...
ca	observador	Apaga tudo: tartarugas e manchas	Clearall (apaga tudo)
crt 4	observador	Cria 4 tartarugas sobrepostas, de cores diferentes e direcções igualmente repartidas	Create-turtles 4 (cria 4 tartarugas)
fd 5	tartarugas	Tartarugas avançam 5 passos	Forward 5 (para a frente 5 passos)
bk 6	tartarugas	Tartarugas recuam 6 passos	Back 6 (6 passos para trás)
pd	tartarugas	Tartarugas põem caneta para baixo (quando se movem deixam rasto)	Pendown (caneta para baixo)
pu	tartarugas	Tartarugas põem caneta para cima (quando se movem não deixam rasto)	Penup (caneta para cima)

setc green	tartarugas	Tartarugas mudam a sua cor para verde	SetColor green (põe cor verde)
rt 90	tartarugas	Tartarugas rodam 90° para a direita	Right 90 (para a direita 90°)
lt 45	tartarugas	Tartarugas rodam 45° para a esquerda	Left 45 (para a esquerda 45°)
repeat 4 [fd 2 rt 90]	tartarugas observador manchas		Repeat 4[forward 2 right 90] (repete 4 vezes os comandos entre [])
let [:x 2 :y 1]	tartarugas observador manchas	Indica as variáveis locais :x e :y e atribui-lhes um valor (2 e 1, respectivamente)	