EDUSCRATCH

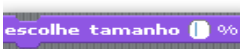
PROGRAMANDO COM O

JOGO DE PERSEGUIÇÃO

Controlo de sprites pelo teclado

(Out-2010)

Plano: Pretende-se que uma bola persiga o gato e que termine o jogo quando o alcançar. A bola deve percorrer todo o ecrã a uma velocidade moderada. O gato será comandado pelas setas do teclado.

Sugestão: Num projecto novo, desenhe um sprite(1) circular vermelho. Se o tamanho do sprite não lhe agradar, use o comando 

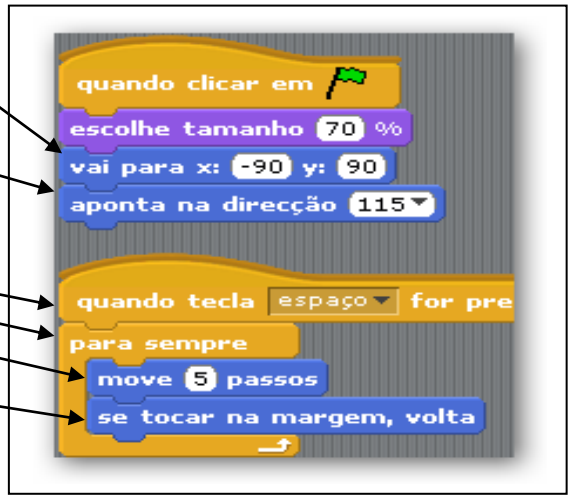
logo a seguir à cabeça 

e o mesmo pode fazer no sprite "Gato" para o harmonizar com o tamanho escolhido para a "Bola". Já agora, por uma questão de elegância, baptize os sprites(2) com nomes portugueses.

Passemos então aos blocos da "Bola":

Precisamos de colocar a bola numa dada **posição** do ecrã, dando-lhe um **declive** não ortogonal.

(Só entre nós: o Scratch conta os ângulos a partir do Norte, como se fossem azimutes, mas com valores negativos quando a medição é feita contra os ponteiros do relógio. Não sei como se conseguirá "vender" isto a alunos a quem se ensina que a origem dos ângulos é a Leste.). O 2º bloco executa quando a tecla "espaço" é premida e inicia um ciclo infinito, (que adiante iremos parar), movendo 5 passos de cada vez e testando se o limite do ecrã foi atingido, para reflectir-se nele. É boa altura para reparar no tracinho azul do cabeçalho e na variação dos valores de X, Y e Direcção (bolas rosa no último quadro desta página). Clique a bandeira e prima "espaço" para ver se está tudo a correr bem.



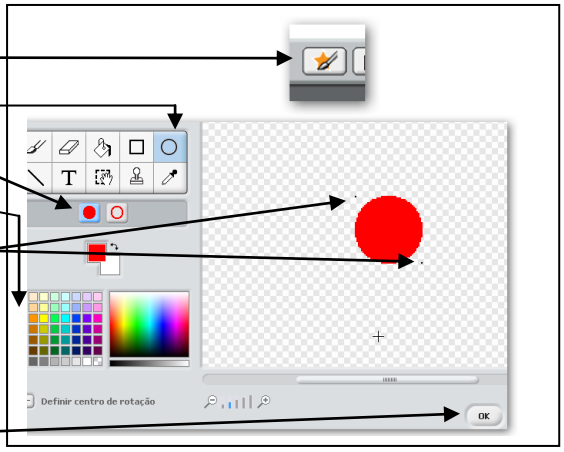
---

(1) Desenhar um sprite circular vermelho

Clique neste ícone para abrir a paleta de desenho
Clique no círculo,
(repare que está seleccionado "cheio")
Escolha a cor

Clique num ponto da paleta e arraste o rato para abrir um quadrilátero onde se inscreverá o círculo. É, mais ou menos, nos locais assinalados com as pintas pretas que eu coloquei para orientação; mas, se quiser um círculo perfeito, prima a tecla "shift" enquanto desloca o rato. Tem 11 quadradinhos de diâmetro, (+/-)

Termine com "OK"



(2) Baptizar um sprite

Seleccione o sprite e, na zona do cabeçalho dos blocos clique no nome para o editar. (Ainda havemos de falar sobre as outras informações do cabeçalho).

Aproveitando a presença do cabeçalho, podemos ficar a saber que o cadeado fechado inibe o sprite de ser arrastado pelo rato quando se trabalha em "modo de apresentação", localmente ou 'online'. Clicando-lhe, ele abre e fecha, alternadamente. Pode testar-se isto passando ao modo de apresentação e tentando arrastar o sprite.

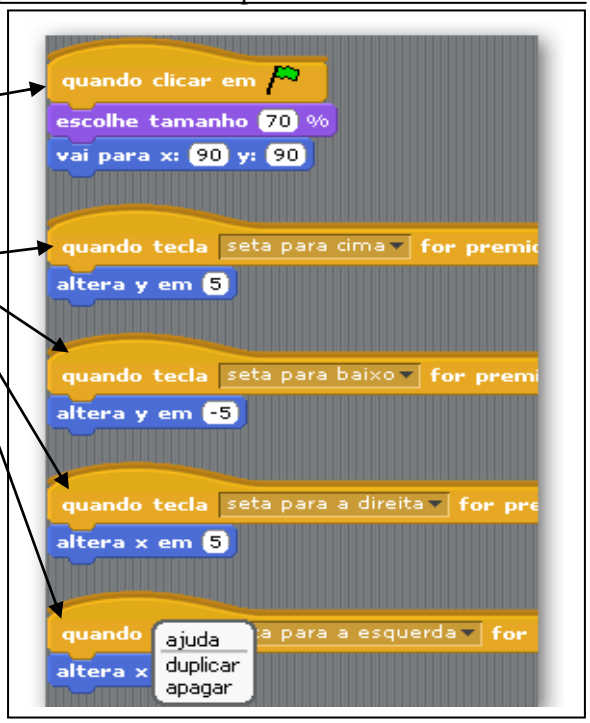
Os 3 ícones à esquerda definem o modo como os trajes seguem a inclinação dada ao sprite:
a volta completa significa que o traje obedece à inclinação,
a dupla seta significa que o traje só assume as direcções +/-90º
o ponto significa que o traje só assume a direcção +90º

} é pena que não se possa mudar isto em 'runtime', porque daria muito jeito.

Os separadores, abrem as áreas dos blocos de comandos, dos trajes e dos sons como depois veremos.

Plano: Pretende-se que uma bola persiga o gato e que termine o jogo quando o alcançar. A bola deve percorrer todo o ecrã a uma velocidade moderada. O gato será comandado pelas setas do teclado.

Já temos a bola a movimentar-se pelo ecrã. Ainda temos que lhe explicar o que deve fazer quando tocar no gato; mas agora vamos tratar do sprite "Gato":



Começamos por colocá-lo numa determinada posição, com um determinado tamanho. Não é preciso que ele obedeça à tecla "espaço", mas sim às setas do teclado. Assim, fazem falta estes quatro blocos:

Cliquei no último bloco, com o botão direito do rato para abrir uma janela que permite duplicar os blocos de comandos, (a partir do comando em que se dá o clique até ao fim do bloco), o que dá jeito em muitos casos, como este; mas também se pode ver a "Ajuda Online" sobre o comando clicado; e podem apagar-se um ou mais comandos, desde aquele em que se clica, até ao último do bloco.

Se o apagamento for involuntário, há ainda uma hipótese de recuperar o último apagamento (**e só o último**), clicando no menu "Editar" no topo do ecrã e escolhendo "Recuperar o apagado".

O valor da alteração de X que a janelinha tapa é, obviamente, **-5**; e agora vamos clicar a bandeira e premir a tecla espaço, para treinar, sem perigo, a evasão do gato às arremetidas da bola, manobrando o bichano com as setas do teclado e ganhando prática porque as coisas vão ter de se complicar a partir daqui.

Para já, ensaiemos uma primeira personalização do projecto. Embora eu vá continuar com a bola e o gato, sugiro que cada um desenhe ou importe(3) os bonecos que achar mais adequados. Até sugiro que se use um touro e um toureiro(4) para dar mais realismo ao projecto.

Quem achar que está tudo muito lento, pode aumentar os passos que cada sprite dá de cada vez. Se alguém tem alguma ideia de como gerir o choque da bola com o gato, deve avançar scratchando. Também se pode alterar o ângulo inicial da bola, para ver como isso pode influir na acção.

---

(3) Para importar um sprite:

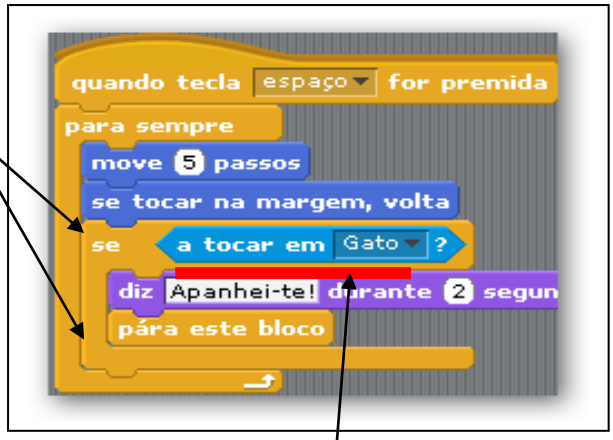
Pode clicar num destes três ícones. O da direita traz da biblioteca um sprite qualquer, à sorte. O do meio permite que se navegue pelas pastas de sprites e se escolha um deles. O da esquerda abre o editor de imagens, onde se pode clicar no botão "Importar" para se escolher um traje. Quando se importa um sprite completo, (e não apenas um traje), podem vir vários trajes e comandos com ele.

Também se pode duplicar um dos nossos sprites, clicando-o com o botão direito do rato e escolhendo a opção duplicar na janelinha que se abre. A função apagar também funciona e, se for activada sem querer, pode recuperar-se o último sprite apagado.

Se podemos exportar um sprite, também podemos importá-lo, com o botão do meio, como se viesse da biblioteca do Scratch. É uma boa maneira de passar sprites de um projecto para outro.

(4) Desenhei o toureiro Escamilho, a partir de um boneco Lego, no meu projecto "Carmen-1". É apenas um que podem usar se não encontrarem melhor.

Plano: Pretende-se que uma bola persiga o gato e que termine o jogo quando o alcançar. A bola deve percorrer todo o ecrã a uma velocidade moderada. O gato será comandado pelas setas do teclado.



A maneira mais simples de cumprir o plano acima enunciado é acrescentar uma condição para que, em cada movimento da bola se verifique se está a tocar no gato; e, em caso afirmativo, termina o programa.

O "diz apanhei-te!" é um adorno, como adorno seria adição de um som(5) a marcar este momento.

Para terminar o programa existe o comando

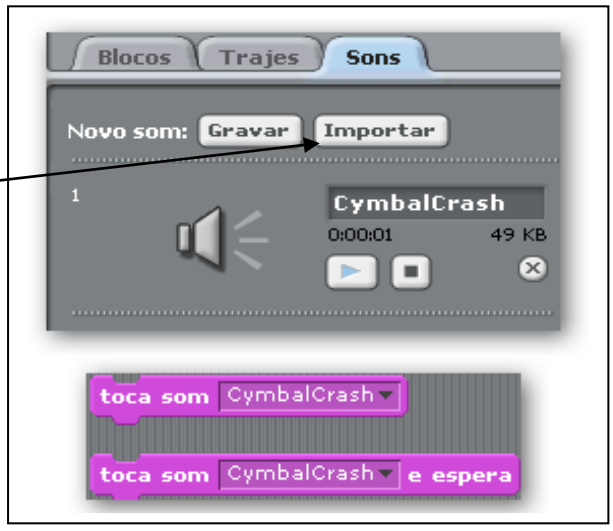
que faz cessar a execução de todos os blocos que, na altura, ainda estejam em actividade. É muito útil para quando temos necessidade de lançar vários ciclos "para sempre" sem controlo de paragem; mas neste caso, o único bloco com "para sempre" é o da bola. Então basta que o paremos com "pára este bloco", até para ilustrar como devemos proceder para cessar a execução de um bloco, sem interromper o trabalho do resto do projecto.

Também é bom que se diga que usar este comando no fim de um bloco é "chover no molhado" já que o bloco termina a sua execução quando cumpre o último comando, (desde que não seja um ciclo "para sempre", claro).

Se quiser adicionar um som quando o gato é apanhado, insira o comando aqui.

---

(5) Para adicionar um som



Para que um efeito sonoro ou gravação de voz seja adicionado às instruções de um sprite é preciso que exista nos seus recursos.

Uma hipótese de adquirir um som é, depois de abrir o separador dos "Sons", clicar no botão "Importar", o que nos leva às bibliotecas de sons do Scratch onde podemos ouvir cada som se o clicarmos uma vez ou importá-lo se o clicarmos duas vezes (ou usando o botão OK depois de o seleccionar com um clique). (É válido para eventuais bibliotecas de som do utilizador, em MP3 ou não comprimidos: WAV, AIF e AU, de 8 ou 16 bits).

Depois do som existir nos recursos, pode ser "tocado" com os comandos da caixa de "Sons". A diferença de comportamento entre estes comandos permite que se use o 1º quando desejamos que o som toque enquanto os próximos comandos são executados, ao passo que no 2º a acção fica parada até que o som cesse. No nosso caso, convém usar o 1º para que toque dentro dos 2 segundos que dura a fala "Apanhei-te".

Sob o nome de cada recurso sonoro existem 3 botões que servem para ensaiar o som ou apagá-lo. Em caso de apagamento involuntário, é possível recuperar o último apagado, clicando no menu "Editar"

Os recursos de um sprite podem ser copiados para outros, arrastando-os da área dos sons para cima da representação do sprite, na área inferior direita do ecrã.

Sugiro que usem o "CymbalCrash" para quando o gato é apanhado e o som de passos ou um "pop" em cada deslocamento da bola, embora isso vá retardar a velocidade dela. Também no gato se podem usar uns "miaus" de cada vez que se move. Enfim, o colorido sonoro é do gosto pessoal de cada utilizador.

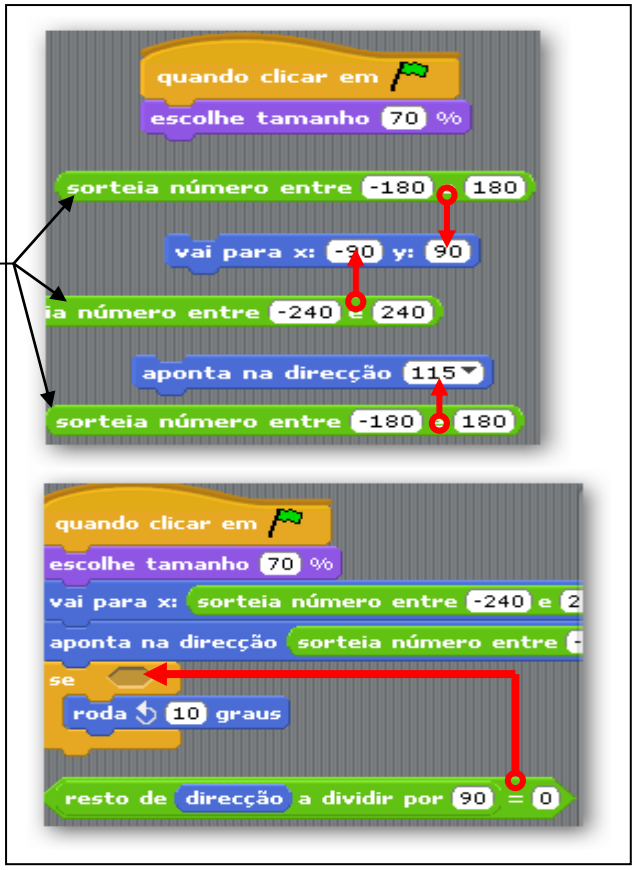
Também se podem fazer gravações, directamente no projecto, desde que se possua um microfone e se clique no botão "Gravar" para abrir a janela de gravação e usar os botões de "Rec", "Play" e "Stop".

Há ainda a hipótese de usar notas musicais de um enorme acervo de instrumentos, incluindo os de percussão; mas este estudo fica para outra oportunidade, se, e quando, fizermos um projecto musical.

Plano: Pretende-se que uma bola persiga o gato e que termine o jogo quando o alcançar. A bola deve percorrer todo o ecrã a uma velocidade moderada. O gato será comandado pelas setas do teclado.

Cumpriu-se o plano; mas pode sempre complicar-se o que é simples. É o que vamos fazer nesta quarta página dedicada ao jogo "Perseguição"

Por que razão é que colocamos os dois sprites nas mesmas posições, quando se clica na bandeira verde? Talvez seja mais desafiante se as posições variarem aleatoriamente em cada início do jogo; e o mesmo se aplica à inclinação inicial.



Os blocos verdes da imagem devem ser colocados nos sítios indicados pelas **setas**, fazendo variar não só a posição de partida da bola, mas também a sua inclinação, para aumentar a insegurança do gato.

Por seu lado, também o gato pode ser colocado aleatoriamente, fazendo com que, em dia de azar, calhe mesmo em cima da bola e comece logo a perder; se quiserem ser imparciais, então limitem as zonas onde cada sprite deve iniciar a execução. As coordenadas do ecrã variam desde (-240,-180) até (240,180). Basta atribuir um quadrante a cada sprite para evitar que se sobreponham inicialmente.

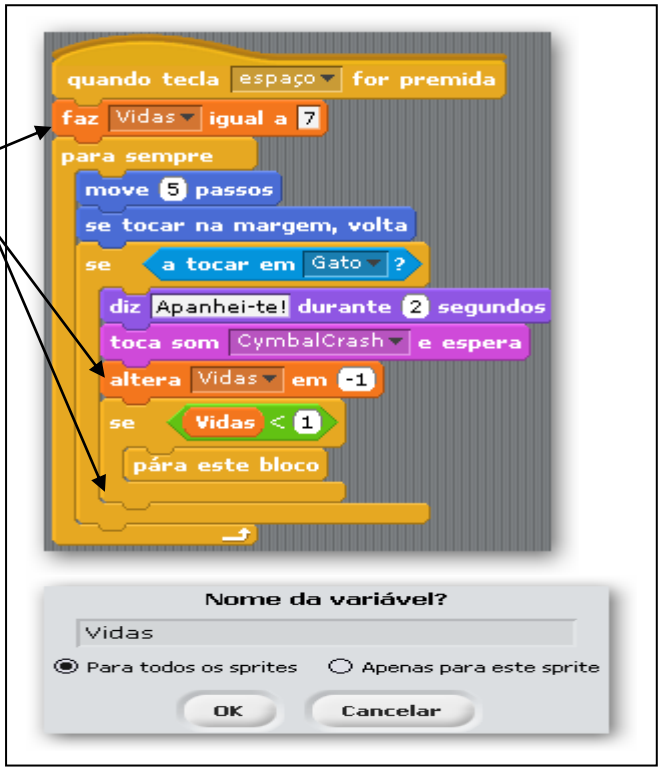
A inclinação(azimute), como já tínhamos visto, varia de -180º até 180º, ambos virados a Sul; mas os ângulos múltiplos de 90º provocam movimentos pouco interessantes, pelo que temos de os evitar com uma condição colocada logo a seguir ao sorteio da direcção: se o resto da divisão inteira da direcção por 90 for zero, alteramo-la 10º e não se fala mais nisso. (A condição bicuda deve ser colocada no local indicado pela seta vermelha).

Isto dá-os o ensejo de falar da caixa de comandos "Números", onde há operações aritméticas e lógicas, bem como algumas funções trigonométricas e logarítmicas, raiz quadrada, arredondamento e estas duas que acabámos de usar: sorteio e resto da divisão inteira.

Por outro lado, talvez seja desmotivador perder logo no primeiro choque dos sprites. Se passássemos a respeitar as sete vidas do gato, o jogo ficaria mais interessante, não é?

Criamos então uma variável(6), inicializada a 7, e substituímos o "pára este bloco" por este conjunto de comandos que subtrai uma vida de cada vez que o gato é tocado e pára o bloco quando "vida" chega a 0

A variável criada fica patente no ecrã e pode assumir três formatos: normal, largo e cursor. Neste último é possível definir valores mínimo e máximo.



---

(6) Para criar uma variável

Abrir a caixa das variáveis e clicar no botão "criar uma variável". Dar nome à variável e terminar com um clique no botão OK.

Na página seguinte falaremos dos "radiobuttons" e também das variáveis-lista. Por agora, convém testar se o projecto está a correr como esperamos. É preciso fugir com o gato enquanto a bola se vangloria pois em caso contrário, as vidas vão-se esvaindo.

Plano: onde é que já vai o plano.....

Temos que falar mais um pouco sobre variáveis antes de complicarmos mais o projecto. Uma variável é um pedacinho de memória do computador, onde podemos gravar um determinado valor, numérico ou alfabético, durante algum tempo, para ser usado onde a oportunidade o requerer.

É como a memória temporária das calculadoras, que se apaga quando ela é desligada. Chama-se variável porque o pedacinho de memória varia de acordo com os nossos desejos.

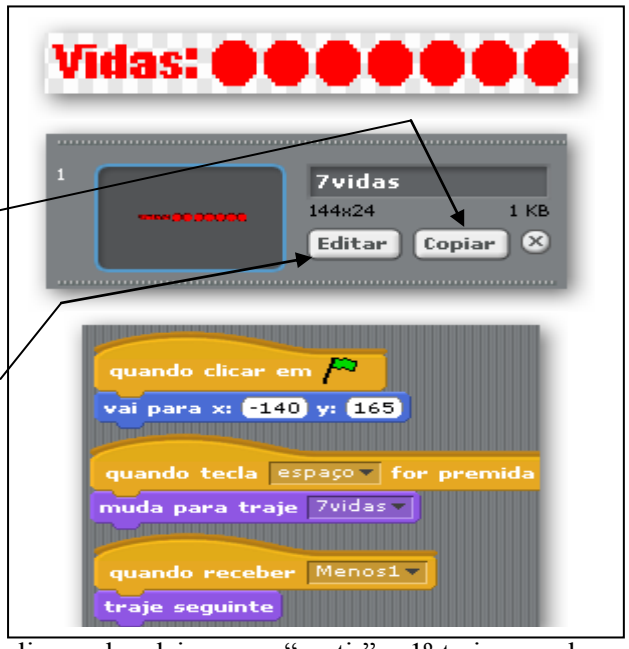
Pode ser um contador, como a "Vidas"; pode ser um sinal, para que se tome alguma decisão conforme o seu valor for negativo ou positivo; pode servir para guardar o resultado de operações aritméticas; enfim, não se pode viver sem elas. Também pode ser uma lista, (ou array), que é um conjunto de variáveis com o mesmo nome, apenas distinguidas por um índice que as numera da 1ª até à última.

Quando se cria uma variável é preciso dar-lhe um nome e um âmbito onde ela é reconhecida: só para o sprite em que está ser criada ou para todos os sprites. A vantagem do egoísmo é que podemos ter nomes iguais de variáveis em cada sprite e, às vezes, isso dá jeito. A vantagem do altruísmo é que há variáveis úteis para todos os sprites, como é o caso da "vidas" com que vamos agora comandar um sprite curioso.

Ter a variável "vidas" exposta é interessante, mas se tivermos uma espécie de semáforo com 7 luzes a apagar-se por cada vida perdida, fica mais engraçado.

Já sabemos fazer um novo sprite:(fizemos a bola). Agora vamos abrir a paleta de desenho e desenhar uma bateria de 7 bolas, como na figura junta. (Cada bola mede 3 quadradinhos de diâmetro. O texto é "arial black", tamanho 12).

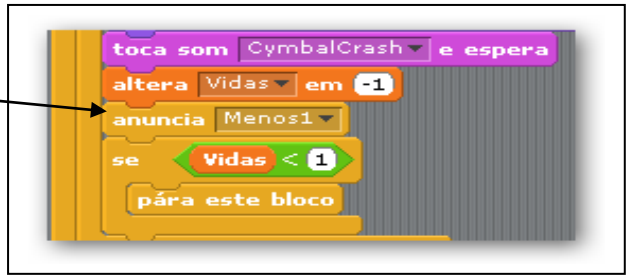
Chamemos "7vidas" a este traje e usemos o botão "Copiar" para obter mais 7 cópias autênticas deste traje, a que chamaremos "6vidas", "5vidas"...etc até "0vidas". Os nomes são pouco importantes, mas ajudam-nos a não nos distrairmos, porque agora vamos "Editar" cada um destes trajes para escurecer as bolas, da direita para a esquerda, até ficarem todas pretas, resultando daqui que, quando mudarmos de um traje para outro, vai parecer que uma bola se apaga em cada mudança.



Arraste a exibição da variável "vidas" para fora do ecrã, para a apagar e coloque o novo sprite nesse local, abrindo agora a zona dos blocos para introduzir alguns comandos: para a colocação inicial quando se clica na bandeira; para "vestir" o 1º traje quando se prime a tecla espaço; e para mudar de traje cada vez que receber o aviso "menos1".

Então: onde é que temos de inserir o comando 

para avisar este sprite que deve mudar de traje?Parece indicado que fique junto do comando que diminui 1 à variável "vidas", na bola.



Chegado aqui, não vou complicar mais o projecto. Deixo nas vossas mãos e no "feed back" que me derem a decisão do caminho a seguir; mas, como já disse: um projecto nunca está definitivamente pronto, podendo ser melhorado e complicado até onde chegar a nossa necessidade e a nossa imaginação.

Como meras sugestões, cito as seguintes possibilidades: definir níveis de dificuldade, fazendo variar os passos da bola e do gato, isto é, a velocidade de cada um; variar o seu tamanho,o que diminui o espaço de fuga; aumentar o número de bolas em doida circulação pelo ecrã; distinguir a letalidade de cada bola com cores e subtraindo ora 1 ora 2 vidas em cada toque; mudar a consistência colorida do gato, a cada toque, por forma a que se torne transparente quando perder as 7 vidas; ou, simplesmente, iniciar outro projecto.

Fico aguardando as vossas perguntas e decisões.