

EDUSCRATCH

PROGRAMANDO COM O



CORRIDA

Interactividade entre sprites

(Out-2010)

CORRIDA

Comecei a fazer este projecto, trabalhando por desporto e sem ter um plano muito definido do que pretendia fazer; mas como o resultado até ficou giro, resolvi partilhar convosco a execução do projecto.

Inicialmente, aproveitei a ideia do “Perseguição” para fazer uma bola circular pelo ecrã; mas logo decidi que devia ter cinco bolas, uma das quais perseguiria as outras, passando a ser perseguida pela que conseguisse apanhar. Algo assim como o jogo da “Apanhada” ou o “Toca e Foge”

Desenhei cinco bolas numeradas de 1 a 5, cada uma com um traje vermelho e outro azul; e comecei por animá-las, dando-lhes uma direcção inicial sorteada e fazendo-as mover-se 5 passos de cada vez, tabelando nos limites do ecrã.

Depois, achei que estavam lentas e resolvi experimentar outros valores de passos; mas como dava muito trabalho alterar os passos de todas as bolas, defini a variável “X” que coloquei no lugar dos passos e fui alterando o valor de “X” e experimentando até assentar em X=10.

A seguir, acrescentei um “muda para o traje livre” em 4 das bolas e um “muda para o traje fica” em apenas uma delas, para que ficassem quatro azuis (livre) e uma vermelha (fica).

Após cada movimento, acrescentei um teste para definir o que deve acontecer quando uma bola azul tocar na bola vermelha (Como a cada momento só há uma bola vermelha e essa não se testa consigo própria, este teste é o suficiente para marcar o toque entre uma bola azul e a vermelha, ignorando todos os toques entre bolas azuis).

Tenho de confessar que, neste ponto, andei um pouco à deriva porque comecei por fazer os dois teste: “a tocar no vermelho” e “a tocar no azul” para que, com um segundo teste a confirmar que traje tinha cada um, mudar de traje azul para vermelho e vice-versa. Porém, ao fim de algumas experiências, compreendi que na altura em que duas bolas de cor diferente se encontravam, havia uma que mudava primeiro, ficando da cor da outra, pelo que o segundo teste já não surtia efeito. Coisas da programação...

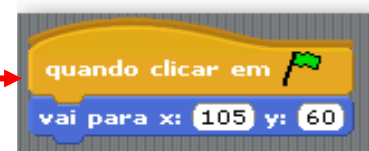
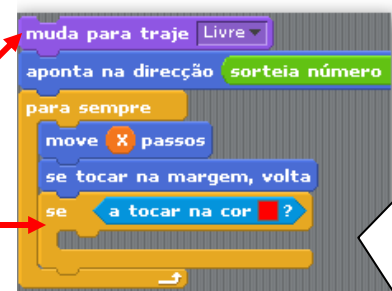
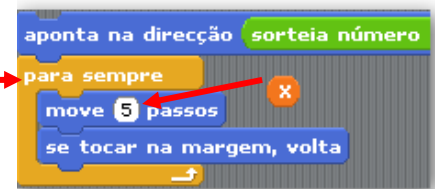
Dentro do “se a tocar na cor vermelha” coloquei, (um de cada vez e experimentado sempre), estes quatro comandos:

- O “anuncia” para que se soubesse que uma bola azul tinha chocado com a vermelha.
- O “faz fogo = #” para que se soubesse que bola tinha chocado com a vermelha. (Fogo é uma variável inicializada com zero e que toma valores 1 a 5 conforme a bola que tocou na vermelha).
- O “espera” porque, depois das bolas trocarem de cor, como estavam em contacto, continuavam a trocar entre si. O “espere” permite que uma delas se afaste e só a que muda para vermelha é que se detém 1 segundo. (Em casos de choque frontal, em que a sobreposição das bolas é muito grande, isto ainda acontece).
- O “toca som pop” é apenas um adorno.

Em cada uma das bolas há, então, um “quando receber” para mudar de traje, (azul para vermelho e vice-versa). Como o número da bola atingida pela vermelha foi guardado em “Fogo”, cada bola testa esta variável e, se coincidir consigo, muda para “fica” (vermelho).

No caso de não coincidir, (só muda para vermelho a bola número “fogo”), a mudança de traje é para o “livre”, o que atinge as outras quatro, incluindo a antiga vermelha.

No que respeita às bolas, falta só dizer que, com o clique na bandeira verde, retomam as posições iniciais, (tal como se vê na figura junta). Na página seguinte, vamos ver a continuação do



projecto e veremos também que cabeça pertence ao primeiro bloco das bolas, que foi reproduzido nesta página, sem cabeça.

Nesta altura em que as bolas já estavam a comportar-se como eu previra, achei que não tinha graça ficar a assistir às suas voltinhas sem qualquer acção do utilizador; e lembrei-me de arranjar cinco carros, também numerados de 1 a 5, para deslocar cada um no ecrã, quando a respectiva bola estivesse vermelha.

Desenhei os 5 carros, coloquei-os nos respectivos pontos de partida, com um bloco com a bandeira e criei um bloco para mover o carro, para sempre, enquanto o valor de “Fogo” coincidir com o número do carro.

O avanço é de 5 passos e, quando a coordenada X do carro ultrapassar 200, terminam todos os movimentos, dos carros e das bolas, dizendo “Vitória” e reiniciando as variáveis “Carro” e “Fogo”, antes de terminar o programa.

Talvez faça aqui falta um som vitorioso; mas convém não demorar muito a terminar o jogo, senão, entretanto, outro carro poderá atingir a meta.

Tive necessidade de criar a variável “Carro” porque o jogo continuava a não solicitar a participação do utilizador; e a melhor ideia que me ocorreu, foi envolvê-lo na escolha de um carro que pode vir a ganhar ou não a corrida.

Assim, o movimento só começa quando o utilizador clicar num dos cinco carros e este fizer o “anuncia Corrida”.

É importante evitar que o utilizador possa escolher outro carro depois da corrida começar, senão ele irá mudando para o que estiver mais avançado e... ganha sempre.

Por isso, “Carro” é inicializado a zero e só quando for zero é que se pode escolher um carro, mudando-se o valor de “Carro” para o número do carro escolhido. Desta forma, não é permitida a mudança enquanto “Carro” não voltar a ser zero.

Pela mesma razão é que só depois de clicar um carro é que se “anuncia Corrida”, que é também o sinal para que as bolas comecem a girar.

“Quando receber Corrida” é, então a cabeça que faz falta no bloco das bolas que, na página anterior, ficou sem cabeça.

O comando “termina o programa” é indispensável para que todos os sprites parem os seus movimentos e terminem os blocos “para sempre” que os controlam.

Finalmente, para assinalar o carro escolhido, criei o sprite “marca” cujo papel é “melgar” o carro escolhido. Assim, começa por estar escondido mas, logo que recebe o anúncio “Corrida” vê que carro foi escolhido e “vai para ele”, para sempre.

Para repetir a corrida, não é preciso clicar na bandeira verde pois, se as bolas arrancam de onde estão, os carros voltam ao princípio e as variáveis são inicializadas, logo, é um jogo novo.

Parece que se conseguiu aqui um jogo já com alguma complexidade, mas contendo uma lógica de programação relativamente fácil, não é?

Frederico

Nota: O projeto “Corrida-Racing” que apoia este documento está publicado em

<http://kids.sapo.pt/scratch/projects/EduScratch/2586>

