

Exercício 16 – Chuckie Egg

Temática

- Variáveis
- Estruturas de decisão
- Estruturas de repetição
- Manipulação da DOM
- Colisão de objetos
- Timers
- Sons
- Eventos

Na sua pasta de trabalho crie uma nova pasta com o nome “ex_16”. Nesta pasta devem ser guardados os ficheiros desenvolvidos nesta aula.

Problema:

Pretende-se criar um jogo *online*, onde o jogador terá como objetivo apanhar todos os ovos que vão aparecendo. Deverá controlar os movimentos da personagem, recorrendo ao teclado, para que apanhe os ovos que se encontram espalhados pela área de jogo.

Nível 1 – Single Player

Dentro da pasta “nivel1_SinglePlayer” e com base na página ‘index.html’ e no ficheiro ‘script.js’, desenvolva o código javascript necessário para implementar as alíneas seguintes.

Pode consultar a versão resolvida em:

http://labmm.clients.ua.pt/LM3/LM3-p/ex17/nivel1_singleplayer/

Siga os seguintes passos para orientar uma possível solução para este jogo. Nesta primeira versão do jogo vamos construir uma versão simplificada, onde serão distribuídos aleatoriamente os ovos pela área de jogo e o controlo do movimento da personagem.

1. Desenvolver a função **carrega_elementos** na qual deve:
 - 1.1. Ser invocada quando todos os elementos da página HTML estiverem carregados;
 - 1.2. Adicionar seis imagens ovo.png à div `main`, recorrendo a estruturas de repetição;
 - 1.3. Adicionar a imagem jogador.png à div `main`.
 - 1.4. Associar a função **comeca_jogar** [especificada na alínea seguinte] ao evento de clique no botão “Jogar” (`btn_jogar`).
2. Desenvolver a função **comeca_jogar** na qual deve:
 - 2.1. Esconder a div `ajuda`;
 - 2.2. Mostrar em `txt_ovos` o número de ovos recolhidos.

3. Desenvolver a função **posiciona_elemento** na qual para já deve:
 - 3.1. Ter os seguintes parâmetros de entrada: *id*, *limiteEsquerdo*, *limiteDireito*, *posicaoVertical*.
4. Ainda na função **comeca_jogar** esta deve:
 - 4.1. Invocar a função **posiciona_elemento** [especificada na alínea seguinte] para posicionar o jogador:
 - horizontalmente entre 0 e 20% do comprimento da plataforma (750px)
 - verticalmente na posição resultante da diferença da distância ao topo da plataforma (300px) e a altura do jogador
 - 4.2. Invocar a função **posiciona_elemento** para posicionar cada uma das 6 imagens dos ovos:
 - horizontalmente entre 20% do comprimento da plataforma e o limite direito da plataforma
 - verticalmente na posição resultante da diferença da distância ao topo da plataforma e a altura dos ovos
5. Voltando à função **posiciona_elemento** esta deve:
 - 5.1. Para o elemento com o id especificado no parâmetro de entrada, deve posicioná-lo horizontalmente aleatoriamente entre os valores *limite_esquerdo* e *limite_direito* (também estes, parâmetros de entrada) e verticalmente com o parâmetro *posicao_vertical*.
6. Desenvolver a função **processa_tecla** na qual deve:
 - 6.1. Ser invocada sempre que ocorrer o evento de tecla pressionada;
 - 6.2. O movimento da personagem deve ser efetuado pela invocação da função **move_objecto** [especificada de seguida], sendo a velocidade 15px;
 - 6.3. Se a tecla pressionada for:
 - a** – deve movimentar a personagem para a esquerda;
 - d** – deve movimentar a personagem para a direita;
7. Desenvolver a função **move_objecto** na qual deve:
 - 7.1. Ter os seguintes parâmetros de entrada: *id*, *vel*, *sent*
 - 7.2. Deslocar o objeto (jogador) de acordo com a velocidade e sentido do deslocamento;
 - 7.3. Atualizar a imagem da personagem¹ de acordo com o sentido do movimento;
 - 7.4. Garantir que o movimento só ocorre na área da plataforma;
8. Na função **processa_tecla** após o movimento da personagem deve:

¹ <https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/rotateY>

- 8.1. Detetar a colisão entre o jogador e cada um dos ovos, invocando a função **deteta_colisao** [especificada de seguida].
- 8.2. Se o valor devolvido pela função **deteta_colisao** for `true`, devem ser efetuadas as seguintes ações:
 - Esconder o ovo e aumentar o contador de ovos `txt_ovos`;
 - Verificar se foram apanhados todos os ovos. Nesse caso, o jogo deve parar através da invocação da função **fim_jogo** [especificada de seguida].
9. Desenvolver a função **deteta_colisao** na qual deve:
 - 9.1. Ter os seguintes parâmetros de entrada: `obj1_id`, `obj1_largura`, `obj2_id`, `obj2_largura`;
 - 9.2. Verificar se há colisão entre os objetos, de acordo com os parâmetros de entrada;
 - 9.3. Retornar o resultado da colisão (`true` ou `false`).
10. Desenvolver a função **fim_jogo** na qual deve:
 - 10.1. Mostrar a `div_ajuda`.

Nível 1 – Multi Player

Pretende-se agora implementar a versão do jogo com 2 jogadores em simultâneo. Para tal, copie o ficheiro `script.js` para a pasta de trabalho “nível1_multiplayer” e efetue os passos seguintes.

Pode consultar a versão resolvida em:

http://labmm.clients.ua.pt/LM3/LM3-p/ex17/nivel1_multiplayer/

11. Para tal deverá alterar a função **carrega_elementos** para ser adicionado o jogador 2.
12. Deve acrescentar à função **processa_tecla** mais duas teclas:
 - o - deve movimentar a personagem 2 para a esquerda;
 - p - deve movimentar a personagem 2 para a direita;
13. Ainda na função **processa_tecla**, deverá ainda contar os ovos apanhados pelo jogador 1 e jogador 2 e mostrar o contador dos ovos apanhados para cada jogador.
14. Deve ainda adicionar os seguintes sons:
 - 14.1. “`tune.mp3`” quando o jogo inicia;
 - 14.2. “`egg.mp3`” quando a personagem apanha um ovo.
 - 14.3. “`leftright.mp3`” para os movimentos esquerda e direita.

Nível 2

No nível 2, o objetivo é acrescentar mais uma plataforma ao jogo e permitir que o jogador se movimente por duas plataformas. Para tal, copie o ficheiro script.js para a pasta de trabalho “nível 2” e efetue os passos seguintes.

Pode consultar a versão resolvida em: <http://labmm.clients.ua.pt/LM3/LM3-p/ex16/nivel2>

15. Na função **comeca_jogar**, deve acrescentar as duas plataformas.
 - 15.1. Deve acrescentar uma escada (escada1) a 550px da sua posição esquerda. A escada tem a largura de 50px.
 - 15.2. Deve posicionar o jogador na plataforma 1 (a de baixo).
 - 15.3. Modifique a função **posiciona_elemento** de forma a posicionar 6 ovos nas duas plataformas.
 - 15.4. Deverá ainda implementar um contador do tempo de jogo (deve invocar a função **tempo_jogo** a criar, que por sua vez deve invocar a função **fim_jogo**, e terminar o jogo ao fim de 10 segundos).
16. Deve ainda adicionar as seguintes funcionalidades na função **processa_tecla**:
 - 16.1. Deve identificar mais duas teclas a pressionar:
 - w** – deve movimentar o agricultor para cima;
 - s** – deve movimentar o agricultor para baixo;
 - 16.2. Garantir que o jogador **apenas** se movimenta horizontalmente nas plataformas e verticalmente nas escadas;
 - A subida/descida das escadas só deve ser possível quando o jogador se encontra “dentro” dos limites das escadas;
 - Quando o jogador estiver a subir/descer não deve ser possível deslocar-se horizontalmente;
 - 16.3. Caso o jogador esteja a subir a escada deve alterar a imagem do mesmo para a imagem **subir.png** e efetuar a deslocação à mesma velocidade dos movimentos horizontais (15px).
 - 16.4. Repita a operação para o jogador descer a escada.
 - 16.5. Quer no caso do jogador subir ou descer a escada deve ser reproduzido o **som_sobe** e parado o **som_caminha**.
17. Implemente na função **deteta_colisao** o código necessário para esta funcionar com a deteção dos ovos nas duas plataformas.
18. Na função **fim_jogo**, deverá ser mostrado novamente a **div_ajuda**, deve ser invocada a função **para_sons** [especificada de seguida] e deve ser parado o **timer** criado.
19. Crie a função **para_sons** que deverá parar todos os sons em execução.

Nível 3

No nível 3, o objetivo é acrescentar mais uma plataforma ao jogo e permitir que o jogador se movimente pelas três plataformas. Para tal, copie o ficheiro script.js para a pasta de trabalho “nível 3” e efetue os passos seguintes.

Pode consultar a versão resolvida em: <http://labmm.clients.ua.pt/LM3/LM3-p/ex16/nivel3>

20. Neste nível deverá alterar todas as funções referidas na alínea anterior de forma a que seja acrescentado ao jogo mais uma plataforma e mais uma escada (escada 2) que terá a sua posição esquerda a 300px e 35px de largura.