

Exercício 12

Temática

- ☐ Variáveis
- ☐ Estruturas de decisão (simples e múltiplas)
- ☐ Áudio e Vídeo
- ☐ Eventos (teclado e rato)

Na sua pasta de trabalho, crie uma nova pasta com o nome “exercicio_12”. Nesta pasta devem ser guardados os ficheiros desenvolvidos nesta aula.

Todo o código *JavaScript* deverá ser colocado em **ficheiros .js externos**, ou seja, **não deve** proceder a qualquer alteração nos ficheiros HTML e de CSS.

Problema:

Pretende-se criar três cenários de interação independentes: (1º) Jardim: Ao entrar no jardim, reproduzir uma mensagem de áudio e, através do teclado, interagir com o piano; (2º) Sala: Controlar a televisão através de cliques no comando visível no ecrã; (3º) Cozinha: Passar com o rato sobre os elementos de forma a interagir com o cenário.

Versão resolvida: <https://labmm.clients.ua.pt/LM3/LM3-p/ex12/jardim.html>

Com base nas páginas de cada um destes cenários, desenvolva o código *JavaScript* necessário para implementar as seguintes alíneas¹.

Parte 1 – Jardim com piano (jardim.html)

No ficheiro de JavaScript correspondente, prepare a página `jardim.html`.

1. Assim que o utilizador entra neste cenário, deve:
 - 1.1. Carregar o áudio `intro.mp3`;
 - 1.2. Reproduzir esse áudio ao clique do elemento `jsTrigger`;
 - 1.3. Após reproduzir o áudio, esconder esse elemento.
2. Ao clique das teclas: `a`, `s`, `d`, `f`, `g`, `h` e `j`, reproduzir os áudios correspondentes às notas dó, ré, mi, fá, sol, lá e si, respetivamente, que se encontram na pasta `assets/media/audio`.

Parte 2 – Sala com televisão

No ficheiro de JavaScript correspondente, prepare a página `sala.html`.

1. Ao clique de cada um dos canais do comando (i.e. 1, 2 e 3)²:
 - 1.1. Associar o conteúdo `c[número do canal].mp4` como *source* do elemento `canal` (p. ex. se clicar no elemento `c2`, deve associar o vídeo `c2.mp4`);
 - 1.2. Reproduzir o vídeo de acordo com o canal selecionado.

¹ **Nota:** Todos os sons e vídeos encontram-se na pasta `assets/media`.

² **Consultar** <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/load>

2. Ao clique do elemento `pausa`:
 - 2.1. Verificar se o vídeo está a ser reproduzido³;
 - 2.2. Reproduzir ou pausar o vídeo consoante o seu estado atual, que é verificado na alínea anterior.
3. Controlar o volume do elemento `canal`:
 - 3.1. O elemento `mute` deve estar escondido por defeito;
 - 3.2. Ao clique do elemento `aumenta`:
 - 3.2.1. Esconder o elemento `mute`;
 - 3.2.2. Se o volume do vídeo for superior a 0 ou inferior a 1, aumentá-lo em 0.1.
 - 3.3. Ao clique do elemento `diminui`:
 - 3.3.1. Se o volume do vídeo for superior a 0.1, diminuí-lo em 0.1;
 - 3.3.2. Assim que o volume do vídeo for zero, mostrar o elemento `mute`.

Parte 3 – Cozinha

No ficheiro de JavaScript correspondente, prepare a página `cozinha.html`.

1. Ao clicar no elemento `coluna`, a música deve manter-se sempre a tocar:
 - 1.1. Associar o áudio `coluna.mp3` como fonte do elemento `musica`;
 - 1.2. Reproduzir esse áudio⁴;
 - 1.3. Estabelecer o volume deste áudio para o valor de 0.6;
 - 1.4. Esconder o elemento `jsInfo`.
2. Ao passar o rato sobre o elemento `gato` ou `cafe`:
 - 2.1. Criar um novo áudio;
 - 2.2. Associar como fonte o áudio correspondente ao elemento sobre o qual passou o rato (p. ex. se o rato passou por cima do café, deve associar o áudio `cafe.mp3`);
 - 2.3. Reproduzir o áudio.
3. Ao passar o rato sobre o elemento `jornal`:
 - 3.1. Associar o áudio `jornal.mp3` como fonte do áudio criado a alínea 1.1;
 - 3.2. Reproduzir o áudio **apenas** se o som do café já tiver sido reproduzido.

Avançado – Funcionamento da televisão

Vamos colocar a televisão a funcionar corretamente. Aprimore o seu algoritmo de forma que:

1. Quando o utilizador chegue à sala, a televisão esteja desligada;
2. O utilizador apenas possa mudar de canal se a televisão estiver ligada;
3. Quando a televisão for ligada, seja reproduzido o vídeo correspondente ao canal 1.

³ Consultar <https://developer.mozilla.org/en-US/docs/Web/API/HTMLMediaElement/paused>

⁴ Ler sobre política de *autoplay* do Chrome <https://developer.chrome.com/blog/autoplay/>