



GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS

VJ1217: DISEÑO Y DESARROLLO DE JUEGOS WEB

Prueba de programación

14 de junio de 2021

Usuario examen: **usuario-de-examen**

Contraseña: **la-del-examen**

Nombre y apellidos: _____

DNI: **dni-de-examen**

Previo. Descarga en el Escritorio de tu ordenador el paquete **ExamenJunio.zip** de *Aula Virtual*. Está en la sección **Exams**. Descomprímelo y desempaquéalo para obtener la carpeta **ExamenJunio**, que contiene tres subcarpetas dedicadas a cada una de las siguientes secciones.

ATENCIÓN: A la hora de probar el resultado de los diferentes ejercicios en el navegador debes emplear la opción **Open with Live Server** de **VS Code** tal como hemos visto en las clases de prácticas.

HTML5/CSS3 (2,5 ptos.)

Abre en **VS Code** la carpeta **Bloque I - HTML-CSS** (da igual que esté vacía). A continuación, crea el fichero **ex1.html** teniendo en cuenta las siguientes indicaciones y los resultados mostrados en la Figura 1.

1. Muestra una lista con las estaciones del año.
2. Vincula el texto **Spring** con el URL: [https://en.wikipedia.org/wiki/Spring_\(season\)](https://en.wikipedia.org/wiki/Spring_(season)).
3. Mediante una regla de estilo CSS, cambia la fuente del texto a **sans-serif**, su color a verde, y las viñetas a cuadrados.
4. Añade una barra de desplazamiento (*slider*) con rango de valores entre 30 y 100 y valor inicial 50, que representarán porcentajes. Muestra ese valor inicial al lado de la barra.

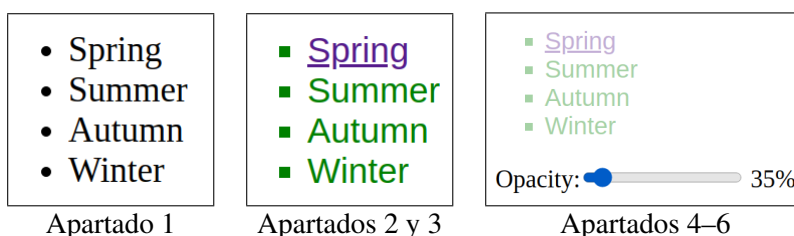


Figura 1: Resultados intermedios en el bloque de HTML5/CSS3

5. Se desea poder controlar interactivamente la opacidad de la lista según el valor del *slider*. Para ello:
 - asocia la función `changeVisibility(this)` al atributo adecuado del elemento correspondiente al *slider*; y
 - escribe dicha función.

Utiliza identificadores donde consideres oportuno. Recuerda que la opacidad puede tomar valores entre 0 (invisible, máxima transparencia) y 1 (máxima opacidad).

6. Completa la función `changeVisibility()` para actualizar el texto del valor actual del slider.

JavaScript (4,0 ptos.)

Abre en VS Code la carpeta **Bloque II - JavaScript**. Al abrirla, encontrarás una estructura de carpetas y ficheros que replica la del proyecto **CapShieldVsMjolnir** que desarrollamos en la práctica 2. Exactamente, encontrarás aquí el proyecto desarrollado hasta el final de la sección 8 (**Your turn 7**). Observa que dispones de una página web, un fichero CSS dentro de la carpeta **css** y un fichero con código JavaScript dentro de la carpeta **js**.

Solo hay una novedad aquí, que es la línea:

```
<p id="chrono" style="font: bold 18px Arial">Time to clash weapons: 00:00</p>
```

en el fichero **index.html**, para dar soporte a una cuenta de tiempo sobre la duración de la animación desarrollada. Ejecuta este proyecto, recuerda el funcionamiento de la animación y el código en JavaScript que la lleva a cabo, y responde a los ejercicios que se plantean a continuación.

7. Reemplazando repetidamente el texto del elemento HTML `p` incorporado ahora, implementa un cronómetro que permita visualizar segundo a segundo el tiempo de duración de la animación, desde su inicio hasta que acaba cuando chocan Mjolnir y el escudo del Capi.
8. Comenta o elimina mediante el editor de VS Code el elemento HTML `p` incorporado aquí en **index.html** y desarrolla el código necesario para sustituir completa y precisamente su función, código que has de basar en métodos de creación de elementos, atributos y contenidos del DOM.
9. En esta animación, Mjolnir y el escudo del Capi se mueven sistemáticamente el uno hacia el otro. Mjolnir, de derecha a izquierda, y el escudo del Capi, justo a la inversa, de izquierda a derecha. Modifica el comportamiento de esta animación para que al pulsar, y mientras se mantenga pulsada, la tecla `→`, ambas armas inviertan el sentido de su movimiento, separándose entre sí. Al soltar la tecla `→` debe recuperarse el sentido original del movimiento de las armas, tendente a acercarlas y, eventualmente, hacer que lleguen a chocar.

Cuando las armas choquen, igual que ocurría originalmente, la animación debe finalizar y no tener efecto alguno, a partir de ese momento, ni la pulsación ni la liberación de la tecla `→`.

Por otra parte, la pulsación persistente de la tecla `→` puede provocar que las armas salgan del escenario, por su derecha y por su izquierda. El comportamiento de las armas cuando no están a la vista debe ser idéntico a cuando están visibles en el escenario, tanto cuando está pulsada la tecla `→` como cuando deja de estarlo.

10. Introduce en el método `advanceWeapons` del objeto `clash` el código necesario para lograr que las armas no abandonen el escenario cuando se pulsa persistentemente la tecla `→`. Al colisionar los bordes correspondientes de las armas con los bordes izquierdo y derecho del escenario, estas debe detener su movimiento y aguantar pegadas a los bordes del escenario mientras se mantenga pulsada la tecla `→`. Al liberarla, las armas deben recuperar inmediatamente sus movimientos de acercamiento entre sí, desde su proximidad a los bordes del escenario.

Phaser (3,5 ptos.)

Abre en VS Code la carpeta **Bloque III - Phaser**. Al abrirla, encontrarás una estructura de carpetas y ficheros que conforman una versión simplificada del juego **Shoot 'em up** que estudiamos en la práctica 5: es

exactamente dicho juego, al que se le han eliminado tanto la pantalla de inicio como la del final, que mostraba el *Hall of Fame*, y se le ha suprimido la entrada de ratón (solo funciona con las teclas de los cursores izquierdo y derecho para mover la nave y con la barra espaciadora para disparar).

Se pretende introducir un nuevo tipo de nave enemiga (ovni). Para ello, en el juego se utiliza un grupo y se rellena de una manera muy parecida a la empleada con los ovnis originales: mira las líneas 11, 26–29, 61, 76 y las funciones `createNewUfos`, en las líneas 81–89, y `activateNewUfo`, en las líneas 147–156. La función `activateNewUfo` es la que debe invocarse para sacar uno de los nuevos ovnis del grupo correspondiente y mostrarlo en la pantalla.

11. Los valores de los parámetros para los nuevos ovnis, velocidad (`newUfosVelocity`) y rango de posiciones en el eje *X* (`newUfosMinX` y `newUfosMaxX`), que se usan en la función `activateNewUfo` se encuentran almacenados en el fichero JSON `enemy.json` que hay en la carpeta **assets**. Por otro lado, estos nuevos ovnis usan una *spritesheet* con 4 frames como imagen —fichero `newUfo.png` que se carga en la línea 61. Para completar el código del juego tienes que:
 - Escribir el código necesario para almacenar en las variables `newUfosVelocity`, `newUfosMinX` y `newUfosMaxX` los valores que hay establecidos en el fichero JSON. Tienes que hacerlo al final de la función `createNewUfos`. Si no sabes hacer este ejercicio, asigna estos valores manualmente.
 - Escribir el código necesario para añadir la animación del *spritesheet* mencionado a los elementos del grupo `newUfos` en `createNewUfos` —usa la etiqueta `newufo`: véase la línea 61 del código proporcionado— y el código requerido para reproducir la animación en bucle y con una velocidad de 10 fps en `activateNewUfo`.
 - Finalmente, escribe el código necesario para que se muestre en pantalla un nuevo ovni: debe activarse un nuevo ovni cada vez que el marcador del juego (`score`) sea un múltiplo de 10 (10, 20, 30...).
12. Escribe el código necesario en el juego para que cuando uno de estos nuevos ovnis choque con la nave el juego termine, es decir, el número de vidas se ponga a 0, además de producirse la correspondiente explosión. Fíjate en el código proporcionado y recuerda la práctica 5: tendrás que hacerlo de una manera muy parecida —aunque no exactamente igual— a como se hace con los ovnis originales.



Figura 2: Pantalla final del juego.

13. Cuando el juego termina, no hay ningún mensaje ni indicación al respecto. Escribe el código necesario para crear un nuevo estado en el juego al que se transite cuando éste termine, es decir, cuando el número de vidas ya no sea mayor que 0. Para ello, crea un nuevo fichero `.js` dentro de la carpeta **js**.

En dicho estado simplemente se mostrará una pantalla con un color de fondo gris (#b5b5b5) y los textos *independientes* “GAME OVER” y “FINAL SCORE: score” —sustituye “score” en el texto por el total de puntos obtenidos en el juego—. El color para ambos textos será rojizo (#eb1333) y con un tamaño de 32px. Además, al hacer *clic* en cualquiera de los dos textos se debe de reiniciar el juego. Puedes ubicar los textos en la posición que desees, siempre y cuando ambos aparezcan aproximadamente centrados en el eje X, el primero (“GAME OVER”) en la mitad superior de la pantalla y el segundo (“FINAL SCORE...”) en la mitad inferior de la misma. Fíjate en la Figura 2.

Entrega: Una vez hayas concluido, abre la carpeta **Espai** en disc personal que hay en el Escritorio de Windows y copia, dentro de ella, la carpeta **ExamenJunio**, que debe tener todos los contenidos de los ejercicios que has estado desarrollando. Si no copias esta carpeta en **Espai** en disc personal será como si no hubieses efectuado el examen.