GRADO EN DISEÑO Y DESARROLLO DE VIDEOJUEGOS
VJ1217: DISEÑO Y DESARROLLO DE JUEGOS WEB
Prueba de programación
4 de junio de 2019

Usuario examen: **usuario-de-examen**
Contraseña: **la-del-examen**

Nombre y apellidos: _____

DNI: **dni-de-examen**

Laboratorio: ☐ LA1 ☐ LA2 ☐ LA3

Previo. Descarga en tu ordenador el paquete ExamenJunio.zip de *Aula Virtual*. Está en la sección **Exams**. Descomprímelo y desempaquétalo para obtener la carpeta ExamenJunio, que contiene tres subcarpetas dedicadas a cada una de las siguientes secciones. **Muy importante**. Después de descomprimirlo y desempaquetarlo *correctamente* **debes borrar** el archivo ExamenJunio.zip.

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/DOM (2,5 ptos.)

Abre en VS Code la carpeta HTML-CSS. Al abrirla, podrás encontrar una sencilla página HTML con una serie de elementos. Fíjate en la estructura de carpetas dentro de este proyecto, ya que también encontrarás un fichero CSS, un fichero de tipo de letra y una imagen.

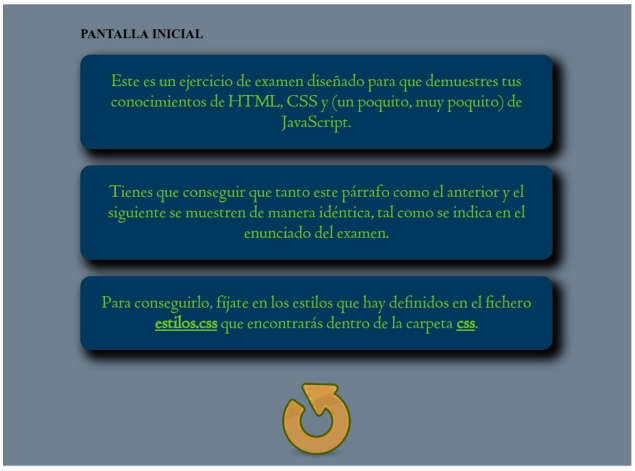
1. (0,2 ptos.) Aplica el estilo definido en el fichero CSS (estilos.css) que consideres necesario a los elementos `div` pertinentes del fichero index.html con el fin de que tengan un aspecto similar al que se muestra en las Figuras 1(a) y 1(b).

2. (0,5 ptos.) Define un estilo para una *clase* en el fichero CSS y aplícalo a los dos elementos `span` que hay en el fichero index.html de tal manera que su contenido aparezca en negrita y subrayado.

3. (0,5 ptos.) Añade la imagen que se encuentra en el fichero change.png al final de la página web de tal manera que deje un margen con respecto al elemento `div` anterior de 3 espacios (`em`), que la imagen aparezca centrada horizontalmente en la página y que tenga unas dimensiones de 100 píxeles de ancho y 110 píxeles de alto. No modifiques el fichero CSS: si tienes que usar un estilo, aplícalo *inline*.

4. (0,5 ptos.) Añade un identificador a cada uno de los dos grandes bloques `div` que hay en index.html: el primero —el que comprende el rótulo "PANTALLA INICIAL"— deberá llamarse begin y el segundo —el que comprende el rótulo "PANTALLA FINAL"— se llamará end. A continuación, añade las reglas necesarias al fichero CSS (estilos.css) de tal manera que el segundo bloque —identificado mediante la etiqueta end— no se muestre en la ventana del navegador al cargar la página.

5. (0,8 ptos.) Crea la carpeta js en VS Code y, dentro, el fichero toggle.js. En él, escribe el código JavaScript para que cada vez que se haga clic en la imagen que insertaste en el apartado 3 cambie la visualización del bloque `div` correspondiente. Tras hacer el ejercicio anterior se muestra el bloque identificado con la etiqueta begin y no el identificado con end. Cuando se hace clic en la imagen se debe mostrar el bloque



(a) Bloque `div` begin

(b) Bloque `div` end

Figura 1: Página web tras completar todos los ejercicios de este bloque. Inicialmente se muestra el primer bloque `div` y el segundo permanece oculto. Tras hacer clic en la imagen se muestra el segundo y se oculta el primero. Con otro clic volvemos a la situación inicial, y así sucesivamente.

end y ocultar el bloque begin. Si se vuelve a hacer clic ocurrirá lo contrario, y así sucesivamente. **Nota:** también tendrás que hacer modificaciones en el fichero index.html para resolver este ejercicio.

Phaser (2,5 ptos.)

Abre en VS Code la carpeta Phaser. Al abrirla, encontrarás una estructura de carpetas y ficheros que conforman un juego similar al Shoot 'em up que estudiamos en la práctica 5 —es exactamente dicho juego, al que se le ha eliminado el tercer estado, el que mostraba el *Hall of Fame* al final—.

6. (1,5 ptos.) Supón que incorporamos conceptualmente a este juego unas naves enemigas situadas fuera del escenario, por encima de su borde superior. Dichas naves disparan plasma dirigido a la nave del jugador que entra en el escenario por los bordes superior y laterales. Cuando dicho plasma alcanza a la nave del jugador, solo inutiliza sus disparos durante 2 segundos. Para ello, resuelve estos dos ejercicios:

   a) (1,0 pto.) Crea la infraestructura necesaria para producir y gestionar los disparos de plasma: un grupo en Phaser cuyos 40 miembros usen la imagen assets/imgs/plasma.png, tengan el sistema físico *Arcade* habilitado, comprueben los bordes del escenario y se desactiven al salir del mismo.

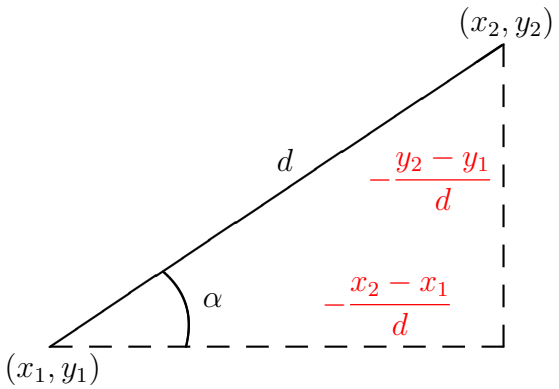
   Con frecuencia fija (`TIMER_RHYTHM`), y en función de una probabilidad tambien fija $(0, 1)$, deben activarse los disparos de plasma, originándolos aleatoriamente en la franja ilustrada en la Figura 2(a), cuya longitud es la suma de la anchura y la altura del escenario y que se alinea horizontalmente al centro con él. La altura de la franja es la de la imagen assets/imgs/plasma.png.

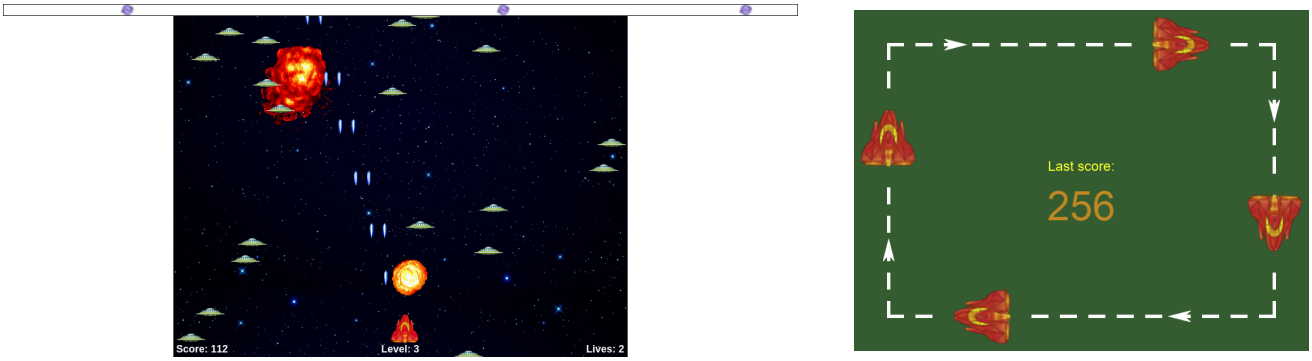
   Cada disparo de plasma debe dirigirse a la posición de la nave del jugador en el momento de dicho disparo. Para ello, ten en cuenta las siguientes fórmulas para calcular las velocidades necesarias.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$v_x = -\cos\alpha \cdot v = -\frac{x_2 - x_1}{d} \cdot v$$

$$v_y = -\sin\alpha \cdot v = -\frac{y_2 - y_1}{d} \cdot v$$





(a) (b)

Figura 2: (a) Origen aleatorio de las nubes de plasma fuera del escenario para entrar en él por los bordes superior o laterales. (b) Estado final mostrando una puntuación obtenida en el juego y la trayectoria de la animación solicitada.

Así, los disparos de plasma entrarán en el escenario tanto por el borde superior como por los laterales. La velocidad v, fija también, establécela en 100.

b) (0,5 ptos.) Implementa el comportamiento de inhibición de sus disparos de láser durante 2 segundos, cuando una nube de plasma alcance a la nave del jugador. Fíjate en que solo has de eliminar la nube de plasma del escenario e impedir el disparo de los lásers por 2 segundos, tanto mediante la barra espaciadora como mediante el botón izquierdo del ratón. En lugar de intentar deshabilitar mediante Phaser solo el botón izquierdo del ratón y la barra espaciadora, tienes que implementar un mecanismo basado en código JavaScript para gestionar la habilitación/deshabilitación simultánea de ambos.

7. (1,0 pto.) Implementa un nuevo estado final al que transite el juego cuando el jugador pierda sus tres vidas. Escribe todo el código relacionado con este estado en los ficheros ya implementados y, especialmente, en end.js: debe presentar la puntuación obtenida en el juego y realizar una animación de la nave del jugador dando vueltas por el interior del escenario, en líneas rectas, cerca de los bordes y con el morro en vanguardia en el sentido del movimiento, girando en las esquinas (Figura 2(b)).

JavaScript (5,0 ptos.)

Abre en VS Code la carpeta JavaScript. Al abrirla, encontrarás una estructura de carpetas y ficheros que conforman un juego similar al Infinity Square que vimos en la práctica 3. Observa que tenemos una página web, un fichero CSS dentro de la carpeta css y un fichero con código JavaScript dentro de la carpeta js.

Nota. No necesitas entender todas y cada una de las líneas del código de partida que se proporciona. Intenta realizar una inspección somera del código y hacerte una idea aproximada y general de cómo funciona. A continuación, céntrate en el código que es relevante para resolver cada una de las preguntas planteadas. **En Aula Virtual hay un vídeo que muestra el funcionamiento del juego tras haber completado todos los ejercicios de este bloque.**

8. (0,5 ptos.) Al ejecutar el juego verás que no funciona, ya que es necesario asignar los valores adecuados a las variables `canvas`, `canvasWidth` y `canvasHeight` en las líneas 111, 112 y 113 del fichero juego.js. Estos valores deben obtenerse a partir del elemento `canvas`, identificado como `canvasGame`, en el fichero index.html. Usa las funciones adecuadas del DOM para ello. Una vez resuelto, el juego ya debería de funcionar: usa las teclas de los cursores para mover el rectángulo granate (la nave) y esquivar los rectángulos oscuros (ovnis). **Si no sabes resolver este ejercicio**, y con el fin de que puedas continuar con el resto, dile al profesor que está cuidando el examen que te proporcione la respuesta.

9. (1,0 ptos.) Modifica la clase `Ship`:

   - Modifica el constructor para que admita un parámetro más, `img`, que será *opcional*, con valor por defecto `null`. Su valor se almacenará en un atributo de la clase con el mismo nombre.

- Añádele dos métodos: `setX` y `setY` que permitirán cambiar la posición de cualquier instancia de la clase en los ejes X e Y respectivamente.
- Finalmente, modifica el método `render` para que, si el valor almacenado en el atributo `img` es distinto de `null`, dibuje la imagen correspondiente en el canvas. Si fuese `null` el método debe dibujar el rectángulo correspondiente tal como venía haciendo hasta ahora.

Usa la imagen del fichero spaceship.png que hay dentro de la carpeta imgs para representar la nave en lugar del rectángulo granate, considerando las modificaciones a la clase `Ship` que acabas de realizar.

10. (1,2 ptos.) Añade un nuevo tipo de ovni al juego: se dibujarán con la imagen nave_alien.png que encontrarás en la carpeta imgs, aparecerán cada 3 segundos con una probabilidad del $60\,\%$, tendrán una posición aleatoria en el eje Y comprendida entre *una quinta* y *cuatro quintas partes* de la altura del canvas, su velocidad en X y en Y estará comprendida entre 6 y 15 y deben *rebotar* cuando alcancen cualquiera de los límites superior e inferior del canvas. La velocidad en Y podrá ser positiva o negativa con la misma probabilidad ($50\,\%$). El resto de características de este nuevo tipo de ovnis serán las mismas que tienen los originales. Almacénalos en el mismo vector en el que se almacenan los originales.

11. (2,3 ptos.) Utiliza el elemento `div` identificado mediante la etiqueta `livesleft` para llevar un contador de las vidas del jugador. Tienes que modificar el juego para que permita *tres* vidas en total. Además, tras cada "muerte" de la nave —excepto la última, ya que el juego se acabará— el juego se pausará durante 4 segundos y se reanudará con la nave situada en su posición original y todos los ovnis que pudieran haber en pantalla eliminados. Para ello será necesario, *entre otras cosas*, que realices las siguientes acciones:

    - En primer lugar añade las reglas de estilo necesarias en el fichero ejercicio3.css que hay dentro de la carpeta css para que este elemento tenga un margen izquierdo de 2 espacios (em), un margen superior de un espacio, un espacio de relleno para sus elementos (*pad*) de $0{,}2$ espacios, alineación vertical *middle* y modo de visualización *inline-block*.
    - Ya en el fichero juego.js, añade una constante para el número máximo de vidas. También tendrás que almacenar en una variable el elemento `livesleft` del DOM, tal como hiciste al principio de este bloque con el canvas y pasárselo en el constructor a la clase `GameCanvas` para que lo guarde (tal como ocurre con el canvas).
    - Añade un método a la clase `GameCanvas`, `renderLives`, que se encargará de dibujar las vidas que le quedan al jugador: para ello, utilizará tantas imágenes de la nave (spaceship.png) como sean necesarias para indicar las vidas restantes. Estas imágenes tendrán aquí unas dimensiones de 82 de ancho por 39 de alto —define constantes para almacenar estos valores—. Este método se invocará tanto al inicio del juego como cuando la nave "muera", es decir, colisione con un ovni.
    - Añade otro método a esta misma clase al que llamarás `removeAllUfos`. Este método simplemente dejará vacío el vector `ufos` de la clase.

Entrega. Una vez hayas concluido, comprime en un nuevo archivo ZIP la carpeta ExamenJunio con todos los contenidos de los ejercicios que has estado desarrollando y súbelo a la tarea correspondiente de *Aula Virtual* **según el grupo de laboratorio en el que estés matriculado**. Si no lo subes donde corresponde, tu examen **no se corregirá**.

Para facilitar la corrección, por favor **marca** el número de cada cuestión que crees que has **resuelto correctamente**:

1 ☐ **2** ☐ **3** ☐ **4** ☐ **5** ☐ **6a** ☐ **6b** ☐ **7** ☐ **8** ☐ **9** ☐ **10** ☐ **11** ☐