Programación de Videojuegos en Lenguajes Interpretados

Grado en Desarrollo de Videojuegos

Examen Convocatoria Extraordinaria Curso 23-24

18 de junio de 2024

► ***Lee detenidamente el enunciado completo antes de comenzar el examen.***

Índice

1 Descripción de la tarea del examen

Se implementará la base para un juego de ritmo. Hoy en día existen multitud de juegos de ritmo, y en algunos casos de mezcla con otros géneros. En este caso vamos a imitar algunas mecánicas de la saga Theatrhythm Final Fantasy.

Figura 1: Theatrhythm Final Fantasy.

En el examen implementaremos una versión muy básica de nivel, donde además habrá un menú y un nivel de juego (Figura 2). En el nivel tendremos al jugador y varios enemigos que lanzan ataque a un ritmo constante. El jugador debe pulsar los botones en el momento adecuado para evitar el daño y vencer a los enemigos.

Tenéis un ejemplo interactivo del ***resultado completo del examen aquí***.

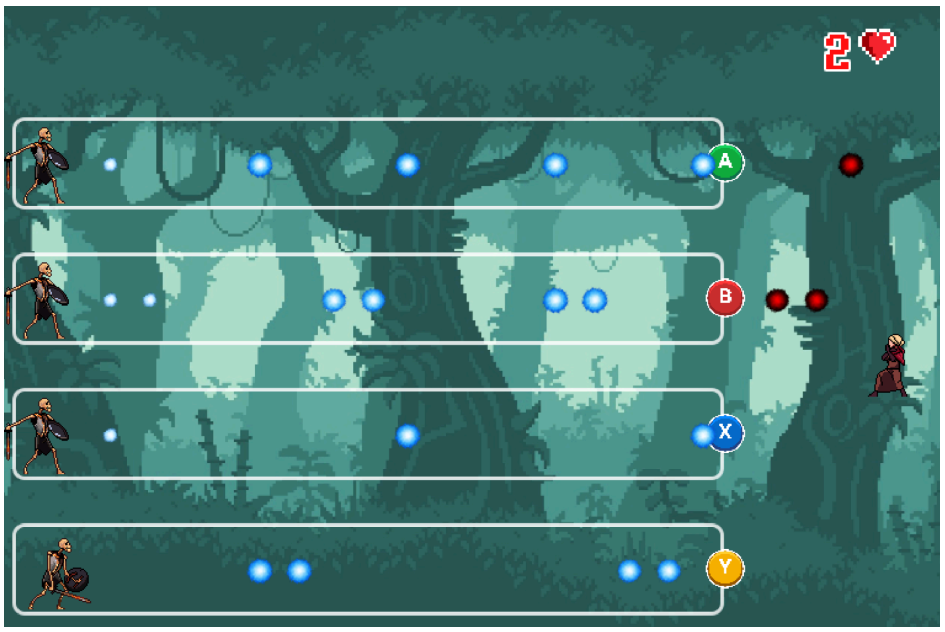

Figura 2: Pantallazo de nuestro juego de ritmo.

Todo el material (sprites, sonidos, etc.) así como un esqueleto del HTML y el archivo `game.js` están disponibles en el Campus Virtual.

IMPORTANTE: Independientemente de la suma de puntos, el examen no estará aprobado si no hay una **aproximación funcional del juego donde el jugador pueda ganar o perder y se vuelva al menú**.

El exámen puede **aprobarse** siempre que el **código** sea **correcto y** haya una **aproximación** mínimamente **funcional** a las mecánicas fundamentales, es decir:

- Un menú de juego con, al menos, un botón para entrar a la escena principal de juego.
- Un enemigo que genere bolas de energía a ritmo constante.
- La mecánica de detener los ataques o ser golpeado.
- Perder o ganar después según si se vencen a los enemigos del nivel o si el jugador se queda sin vida.
- Pantalla de final de juego

Los apartados que aparecen a continuación no representan el orden de desarrollo del examen por lo que te recomendamos que los hagas de la manera que consideres más adecuada, teniendo en cuenta los mínimos que se piden.

Así mismo, te recomendamos que hagas uso de un repositorio de Git **local** en el que puedas ir **guardando versiones intermedias** de la solución del examen. El objetivo de esto es **que podáis deshacer cambios fácilmente** ya que es importante que **lo que entreguéis se ejecute**. Crear copias intermedias en local, aunque más rudimentario, también os puede servir.

1.1 Menú principal (1 punto)

Cuando se empieza la partida, hay un menú con el título del juego en rojo con borde blanco. En este menú se puede elegir la dificultad y empezar el juego. La bola (frame 1 de energy.png) indica la opción elegida, pulsar “S” y “W” permite cambiar entre opciones. El nivel de juego en sí, empieza al pulsar la tecla de “espacio”, y se jugará con el modo de juego que esté seleccionado por la bola en el momento de pulsar la tecla.

La fuente empleada en todo el juego se llama “*bitdragon*” y ya está cargada desde la CSS.

Figura 3. Menú de juego

La diferencia entre el nivel normal y el difícil será la velocidad de movimiento de las bolas de energía generadas en los ataques.

1.2 Escena de juego (0.5 punto)

Crea una escena con el fondo de juego y las zonas donde aparecen los enemigos. La escena de juego tiene además una música de fondo (*/sounds/forestMusic.mp3*).

Figura 4. Escena de juego.

Las zonas que delimitan donde aparecerán los enemigos y las bolas de energía se crean con el recurso “area.png”.

1.3 Enemigo

1.3.1 Enemigos ataque (2.5 puntos)

Los enemigos (esqueletos) se encuentran a la izquierda del escenario, con una animación de reposo (Figura 6) y además un pequeño movimiento en el eje Y.

Figura 5. Esqueleto en reposo (SK_Idle.png)

Los enemigos lanzan bolas de energía (Figura 7) al atacar realizando su animación de ataque (Figura 6). Las bolas de energía se mueven en el eje X a una velocidad constante y tienen un *collider* circular.

Figura 6. Esqueleto atacando (SK_Attack.png)

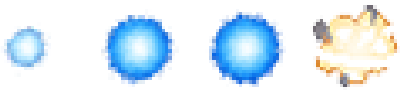

Figura 7. Bola de energía (energy.png)

Los ataques de los esqueletos son constantes y con tiempos diferentes entre cada esqueleto. Además, algunos esqueletos pueden generar 2 bolas de energía consecutivas (Figura 8).

Figura 8. Esqueleto que genera una bola de energía al atacar y esqueleto que genera 2 bolas de energía al atacar.

1.3.2 Zona de botones (2 puntos)

A la derecha del escenario se encuentra la zona de botones (Figura 9). El jugador debe pulsar el botón correspondiente cuando las bolas de Energía pasan por encima para evitar que estás golpeen a su personaje. Los botones, de arriba a abajo, corresponden a las teclas “A”, “S”, “D” y “F”.

Cuando la tecla correspondiente es pulsada, el botón en pantalla se hace más grande (Figura 9, botón “A”).

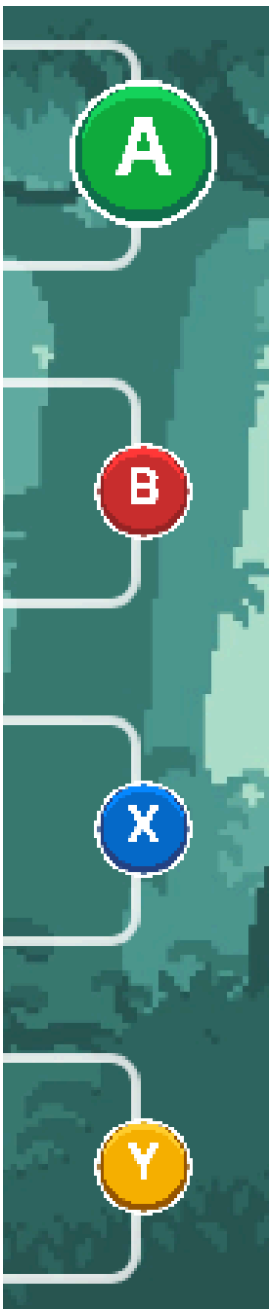

Figura 9. Área de botones (el botón “A” es más grande porque está siendo pulsado).

Si un botón es pulsado y hay una bola de energía en su área circular de colisión, la bola de energía será destruida.

Figura 10. Bola de energía siendo destruida porque se ha pulsado la tecla correspondiente al botón A.

1.3.3 Enemigos y su muerte (1 puntos)

Los enemigos tienen 2 puntos de vida, cada vez que una bola de energía es destruida por pulsar un botón, el enemigo que generó el ataque perderá un punto de vida. Cuando los puntos de vina del enemigo llegan a 0, se lanza su animación de muerte (Figura 11) y desaparece del escenario al completar dicha animación.

Figura 11. Muerte del esqueleto (SK_Dead.png).

Cuando un enemigo muere, todas las bolas de energía que ha generado, y no han dañado al jugador, se destruyen.

1.4 El jugador

1.4.1 Avatar y zona de daño (2 puntos)

El jugador es una maga de luz que aparece a la derecha del escenario con una animación de reposo (Figura 12), el jugador tiene 5 puntos de vida al comenzar el nivel. Cuando bola de energía no es destruida y llega a la zona de daño el jugador verá reducidos sus puntos de vida en 1.

Figura 12. Avatar del jugador (CH_Idle.png)

La zona de daño es una zona que va desde la parte superior del escenario a la parte inferior (Figura 13). Cuando una bola de energía toca esta zona, se vuelve roja, indicando que el

jugador ha sido golpeado por esa bola (También se puede ver en la Figura 13). Además en ese momento, el avatar del jugador ejecutará la animación de ser herido (Figura 14).

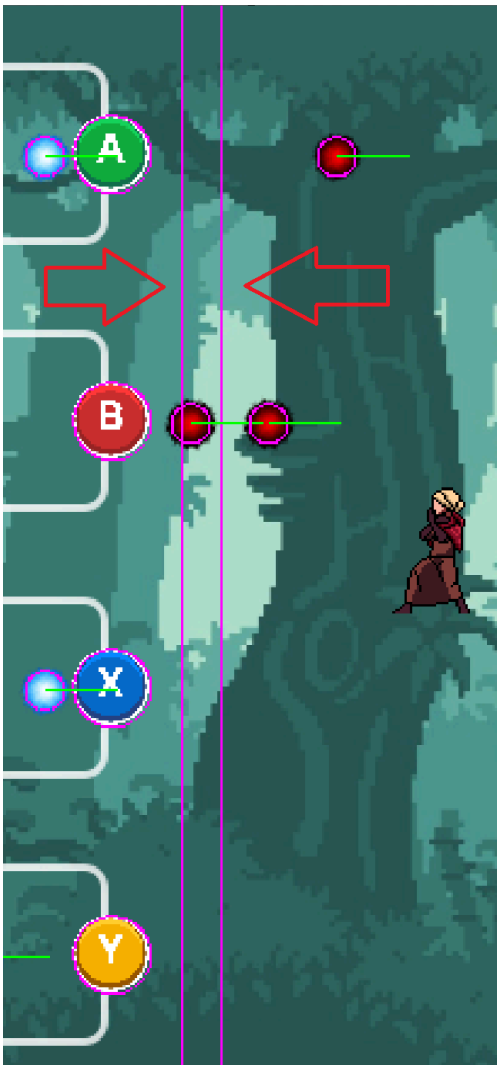

Figura 13. Zona de daño (se ve marcada entre dos flechas rojas en la imagen superior)

Figura 14. Animación de jugador recibiendo daño (CH_Hurt.png)

1.4.2 Indicadores de vida (1 puntos)

En la parte superior derecha se ve en todo momento la cantidad de vidas que le queda al jugador (Figura 15).

Figura 15. Animación de jugador recibiendo daño (se encuentra formado por texto y por el archivo heart.png)

1.5 Victoria y derrota (1 puntos)

Un partido del juego termina si se cumple una de las 2 condiciones siguientes:

- El jugador pierde sus 5 vidas (condición de derrota).
- Todos los enemigos son eliminados (condición de victoria.

Cuando el juego termina se detiene la música de fondo, se ejecuta la animación de muerte del jugador (Figura 16) y después de 2s se lanza una nueva escena con el mensaje que corresponda y un botón “menu”. Si se pulsa espacio se volverá a la pantalla de título.

La figura 17 muestra las escena de victoria/derrota.

Figura 17. Animación de muerte del jugador (CH_Dead.png)

Figura 17. Mensaje de Victoria/Derrota

2 Ayuda

Prioriza conseguir todas las mecánicas básicas (requisitos mínimos) para aprobar y mejora a partir de ahí.

Por ejemplo, un flujo de trabajo puede ser el de empezar por tener un menú simple con un solo botón que permita empezar el juego. Después hacer un enemigo con dos vidas y capaz de generar bolas de energía cada X tiempo. Después generar la zona de botones y la posibilidad de que pulsando la tecla correspondiente se destruya la bola de energía que pasa por encima, quitando así vida al enemigo que generó la bola. Lo siguiente sería hacer la zona de daño y la opción de derrota si pasan 5 bolas por la zona. Por último desarrollaría la condición de victoria si el esqueleto se queda sin contadores de vida.

Después sería abordar los detalles (botón más grande, colorear las bolas de energía al dañar al jugador, el indicador de vida, y las animaciones

PD: El ejemplo del juego que se proporciona contiene varios trucos para facilitaros comprobar las mecánicas. Implementar **estos trucos no deben implementarse en el examen**, aunque puede ayudaros a vosotros mismos a comprobar el resultado conseguido. Las **teclas y trucos que podéis ejecutar** son los siguientes:

- “C” → Activa/Desactiva el modo debug para ver/ocultar los colliders.

3 Evaluación

El código se ejecutará igual que se ha hecho durante el curso, abriendo un servidor web local en la raíz del proyecto (http-server o live-server, por ejemplo). Se probará con la última versión disponible de Google Chrome a fecha del examen, exclusivamente.

El examen tendrá una nota de 0 a 10 (aunque puede sacarse hasta 11 puntos), siendo necesario un 5 para aprobar. Si el código del examen no se ejecuta (error de sintaxis), el juego se queda colgado, o la solución es muy lejana a lo pedido, el examen estará suspenso.

Cada apartado recibirá, como máximo, el valor indicado. Y se valorará el estilo, el uso correcto de construcciones y se tendrá en cuenta la solución en general (no sólo los apartados independientemente). Es decir, a partir de una versión que funcione y cumpla los mínimos pedidos, se tendrá en cuenta la calidad del código, tanto la arquitectura de clases como la corrección de la implementación, de las funciones y de los métodos.

4 Entrega

La entrega se hará a través del Campus Virtual, en la entrega habilitada para tal propósito.

Se debe subir un proyecto completo como archivo comprimido llamado ***"rythmic.zip"***. El proyecto deberá tener un archivo ***"README.txt"*** con el **nombre** del alumno y su ***DNI***. Un proyecto sin este archivo no será evaluado. La entrega es individual.

Una vez hecha la entrega (subida la solución al campus), descárgala y comprueba que contiene la última versión de tu examen (**asegúrate de que has subido la versión correcta**).

5 Materiales

Todo el material para realizar el examen (esqueleto de HTML, `game.js`, sprites, sonidos…) está disponible en el Campus Virtual. Aunque este material es suficiente para hacer el examen, se puede modificar si se considera necesario.

Además el material contiene un archivo ***sizes.txt*** con el tamaño de las imágenes y de los frames de cada spritesheet.

Así mismo, se pueden usar todos los materiales disponibles (Internet, documentación, apuntes), pero **no se puede establecer ningún tipo de comunicación con otros compañeros o personas externas**. **Tampoco está permitido el uso de IAs generativas**. Cualquier de estas acciones se considerará copia.

6 Copia

Cualquier intento, fructuoso o infructuoso, de copia o uso de IAs generativas supondrá la aplicación de la normativa de la asignatura y el suspenso de la asignatura en todas las convocatorias restantes del curso actual.