Programación de Videojuegos en Lenguajes Interpretados

Examen Convocatoria Ordinaria Curso 21-22

11 de enero de 2022

IMPORTANTE: Lee detenidamente el enunciado completo antes de comenzar el examen.

1 Descripción de la tarea del examen

Se implementará un juego con una serie de mecánicas parecidas a las del juego clásico JetPac (Figura 1)

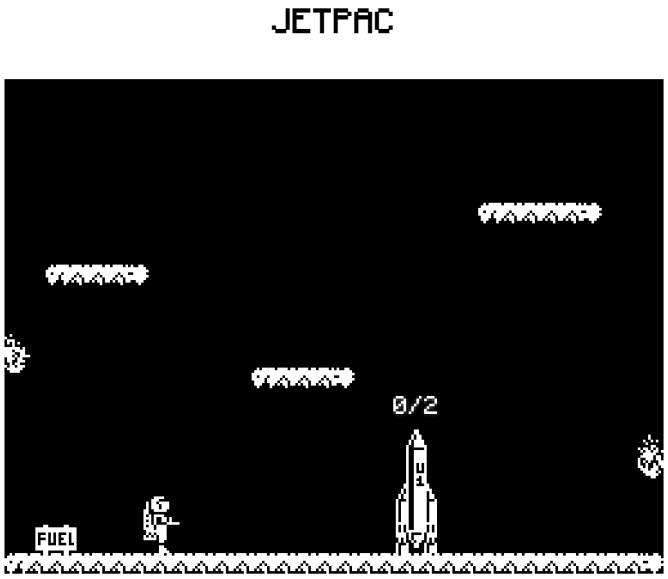

Figure 1: Nuestra versión de JetPac

En el examen hay que implementar el movimiento del jugador, la recogida de combustible para hacer que la nave abandone el planeta y un tipo de enemigo sencillo: los meteoritos. Tenéis un ejemplo interactivo del resultado completo del examen en https://uaj.fdi.ucm.es/pvli/.

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

IMPORTANTE: Independientemente de la suma de puntos, el examen no estará aprobado si no hay, al menos, un menú simple con un botón para entrar al juego, un personaje con movimiento completo, la opción de perder cuando un meteorito mate al jugador y la opción de ganar al recoger una unidad de combustible (no es necesario que el combustible se vea afectado por la gravedad) y dejarla en la nave. Es decir, **el examen no estará aprobado si no hay una aproximación mínimamente funcional a las mecánicas fundamentales**.

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.

1.1 Menu principal (1 punto)

Cuando se empieza la partida, hay una menú en el que se elige la dificultad con 3 botones, que nos llevan al juego al hacer clic con el botón izquierdo del ratón:

- Fácil: son necesarias 2 unidades de combustible para salir del planeta y aparecen nuevos meteoritos cada 2 segundos.
- Intermedio: son necesarias 3 unidades de combustible para salir del planeta y aparecen nuevos meteoritos cada 1 segundo.
- Difícil: son necesarias 5 unidades de combustible para salir del planeta y aparecen nuevos meteoritos cada 1/2 segundo.

La fuente empleada en todo el juego se llama Pixeled y ya está cargada desde la CSS (no es necesario que hagáis nada para utilizarla).

1.2 Escena (1 punto)

Crea una escena con un suelo y varias plataformas como aparece en la imagen al principio de este documento. La cámara queda fija en el centro de la escena. Los tiles empleados (`tileset.png`) son de 8x8 y el mapa en Tiled es de 32x24 tiles. La puntuación máxima de este apartado se conseguirá si el suelo y las plataformas son creadas desde Tiled.

1.3 Jugador (2 puntos)

El jugador se mueve de izquierda a derecha usando las flechas correspondientes de los cursores. El movimiento horizontal es *toroidal*, es decir, cuando el jugador sale por el lado izquierdo de la pantalla reaparece por el derecho y viceversa. El jugador colisiona con el suelo y las plataformas.

Cuando se pulsa el cursor de la flecha arriba, el jugador se propulsa en vertical. Cuando se deja de pulsar, cae por gravedad. No existe ningún límite por la parte superior de la pantalla.

Las animaciones de andar y volar del jugador están disponibles en la spritesheet llamada `jetpac.png`.

1.4 Recoger combustible (1,5 puntos)

La unidad de combustible (`fuel.png`) se crea en una posición aleatoria de la escena, tiene gravedad y colisionará con el suelo y las plataformas. Cuando el jugador pasa por encima de ella la recoge, es decir, la unidad de combustible aparecerá sobre el jugador y se trasladará con él. Cuando el jugador la recoge se emitirá un sonido (`pick.wav`).

1.5 Recargar combustible (2 puntos)

Crea una nave (`spaceship.png`) en una posición fija de la escena. Si el jugador ha recogido combustible y pasa por encima de ella, la nave recargará esta unidad de combustible. Al hacer la recarga se emitirá un sonido (`drop.wav`).

La nave requiere de un número de unidades de combustible para poder recargar completamente y salir del planeta. El número de unidades que ha recargado y el total de unidades que necesita aparecen en un HUD sobre la nave.

Si al recargar combustible la nave no está completamente recargada, aparecerá una nueva unidad de combustible en una posición aleatoria de la escena.

El número de unidades que se necesita para recargar completamente la nave depende del modo de juego, tal y como se ha explicado en la sección del Menú principal.

1.6 Meteoritos (1,5 puntos)

Los meteoritos se crean en una posición aleatoria en la parte superior de la escena y se mueven siempre con un ángulo aleatorio descendente. También tienen un movimiento horizontal toroidal. Los meteoritos están animados (spritesheet `meteor.png`).

Cuando un meteorito colisiona con el suelo o las plataformas, explota (usa `explosion.png` y `explosion.wav` para generar las explosiones).

Se genera un meteorito nuevo cada N segundos. Este número de segundos depende del modo de juego, tal y como se ha explicado en la sección del Menú principal.

1.7 Ganar o perder (1 puntos)

Si al recargar combustible la nave está completamente recargada, despegará con una animación y se activará un sonido de victoria (`win.wav`). Después se volverá al menú principal.

Si un meteorito colisiona con el jugador entonces explota, se activará un sonido de derrota (`lose.wav`) y se volverá al menú principal.

2 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, exclusivamente.

El examen tendrá una nota de 0 a 10, siendo necesario un 5 para aprobar. Si el código del examen no se ejecuta (error de sintaxis, el juego se queda colgado, solución muy lejana a lo pedido), estará suspenso. Cada apartado recibirá, como máximo, el valor indicado.

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, se tendrá en cuenta la calidad del código, tanto la arquitectura de clases como la corrección de la implementación de las funciones y métodos.

3 Entrega

La entrega se hará a través del Campus Virtual, en la entrega habilitada para tal propósito. Se debe subir un proyecto completo en un archivo comprimido. El proyecto deberá tener un archivo de texto con el nombre del alumno y su DNI. **Un proyecto sin este archivo no será evaluado**.

La entrega es individual.

4 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.

Así mismo, se pueden usar todos los materiales disponibles (Internet, apuntes), pero no se puede establecer ningún tipo de comunicación con otros compañeros o personas externas. Esto se considerará copia.

5 Copia

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