

Programación de Videojuegos en Lenguajes Interpretados

Grado en Desarrollo de Videojuegos

Examen Convocatoria Extraordinaria Curso 22-23

16 de junio de 2023

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

Índice

[Índice](#)

[1 Descripción de la tarea del examen](#)

[1.1 Menú principal \(1 punto\)](#)

[1.2 Escena de juego \(0.5 puntos\)](#)

[1.3 Los jugadores - TwinBee y WinBee \(2 puntos\)](#)

[1.4 Los disparos \(2 puntos\)](#)

[1.5 Los enemigos \(1.5 puntos\)](#)

[1.6 Victoria y derrota \(1 punto\)](#)

[1.7 Power Ups \(2 puntos\)](#)

[2 Ayuda](#)

[3 Evaluación](#)

[4 Entrega](#)

[5 Materiales](#)

[6 Copia](#)

1 Descripción de la tarea del examen

Se implementará un juego con una serie de mecánicas parecidas a una de las fases del juego de 1985 TwinBee (Figura 1).

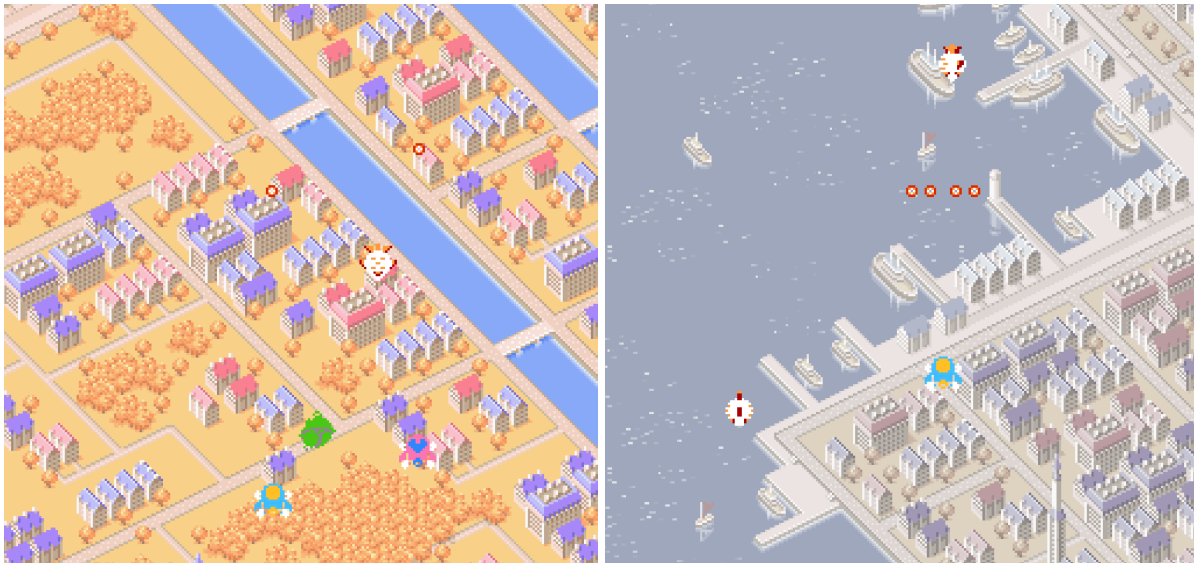


Figure 1: Nuestra versión de TwinBee

En el examen hay que implementar una escena de nivel, el movimiento y ataque del jugador, un tipo de enemigo y un tipo de power up.

Tenéis un ejemplo interactivo del **resultado completo del examen** [aquí](#).

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 y ataque, y que el juego vuelva a la pantalla de título una vez se gane o pierda.

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

- El menú de juego con un botón para entrar a la escena principal de juego.
- El jugador con movimiento y ataque.
- Un enemigo básico.
- Fin del juego al morir o llegar al final del recorrido.

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, aunque más rudimentario, también os puede servir.

1.1 Menú principal (1 punto)

Cuando se empieza la partida, hay un menú en el que se elige el número de jugadores y empieza el juego si pulsamos sobre ellos con el botón izquierdo del ratón.

El botón “1 player” hará que el juego empiece para un solo jugador mientras que “2 player” nos permitirá tener dos jugadores en pantalla.

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



Figura 2. Menú

Para maximizar la puntuación en este apartado es necesario que los textos tengan un borde de otro color y que el color del borde del título se genere de forma aleatoria.

1.2 Escena de juego (0.5 puntos)

Crea una escena con un fondo que será el que se mueva para dar sensación de que avanzamos en el nivel (la cámara es estática).

El fondo del escenario estará en movimiento **mientras aún no se haya mostrado entero**. Es decir, es la posición y de la imagen del fondo la que cambiará progresivamente hasta que se visualice la parte superior.

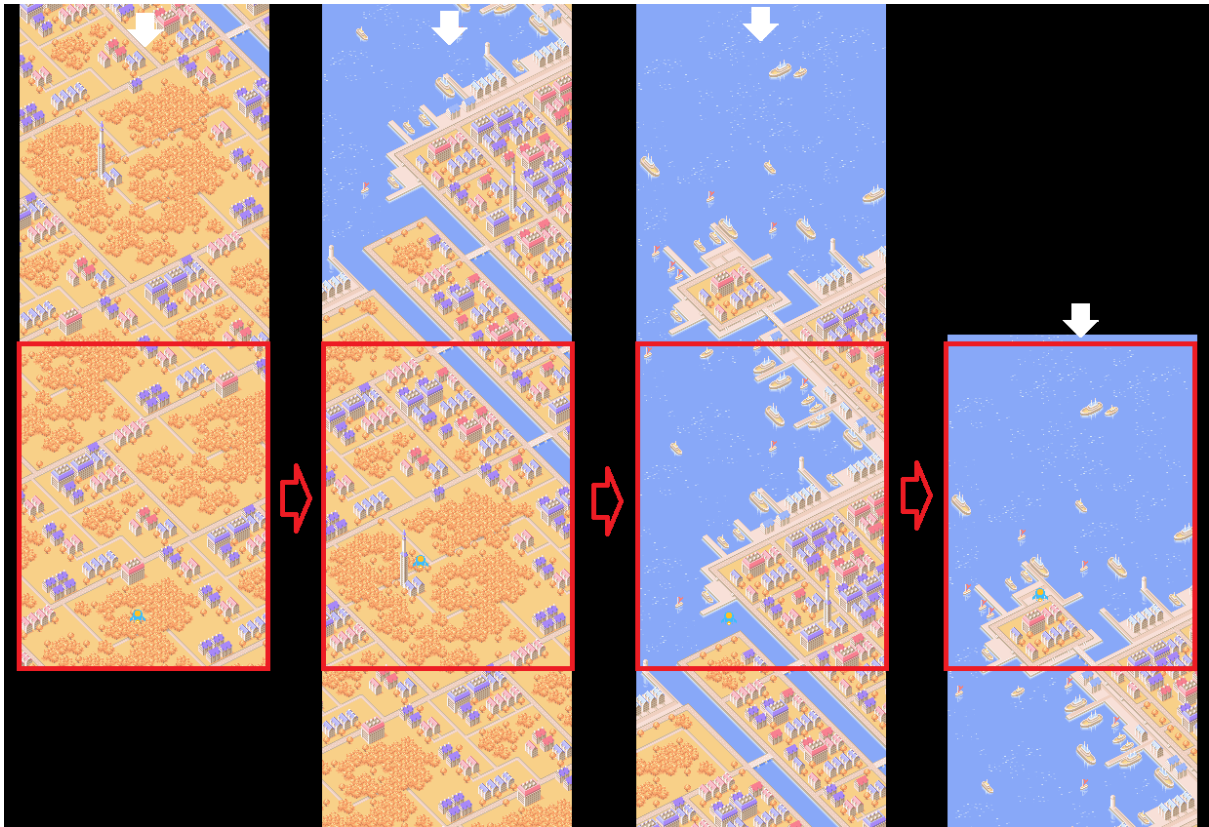


Figura 3. Escena de juego (se puede usar background.png ó background_hcontrast.png)

1.3 Los jugadores - TwinBee y WinBee (2 puntos)

El jugador está compuesto por un spritesheet de 4 frames (Figura 4). El jugador se mueve en 8 direcciones a una velocidad fija cuando se pulsan las teclas “W”, “A”, “S” y “D” y no debe salirse de la escena de juego. Además el jugador podrá disparar pulsando la tecla “espacio”, momento en el que se reproduce el archivo de sonido “shoot.wav” (ver sección 1.4 Los disparos).



Figura 4. Jugador 1 (twinbee.png)

El sprite del jugador cambia según la dirección en la que se mueve, levantando el brazo derecho (movimiento positivo en el eje X), el izquierdo (movimiento negativo en el eje X) o ninguno (si no hay movimiento o si únicamente existe movimiento en el eje Y).

Si sólo juega un jugador, este aparece centrado horizontalmente en la escena.

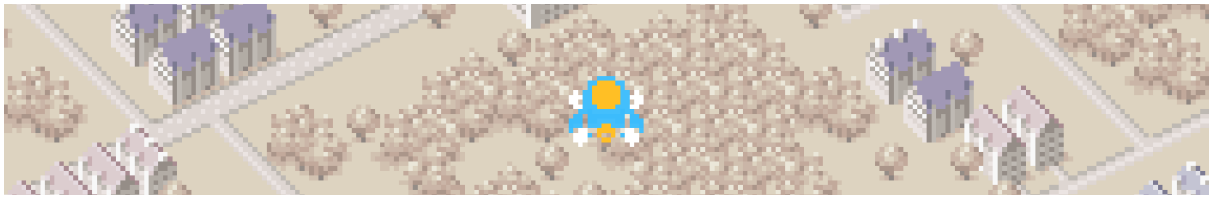


Figura 5. Twinbee en el centro de la pantalla de juego al comenzar el nivel

Cuando en el menú se elige la opción “2 players” debe aparecer un segundo personaje jugable en la escena de juego. Este 2º jugador también se mueve en 8 direcciones pero pulsando teclas diferentes, en este caso los cursores (“↑”, “←”, “↓” y “→”).

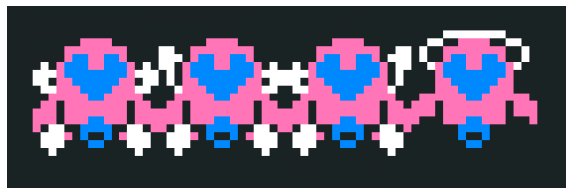


Figura 6. Jugador 2 (winbee.png)

El jugador 2 será capaz de disparar pulsando la tecla de “intro” (ver sección 1.4 Los disparos), momento en el que se reproduce el archivo de sonido “shoot.wav”. Y el sprite debe comportarse de la misma forma que el del jugador 1.

Además, en el caso de haber 2 jugadores en la escena de juegos, estos inicialmente aparecerán centrados horizontalmente en la primera y segunda mitad de la escena.

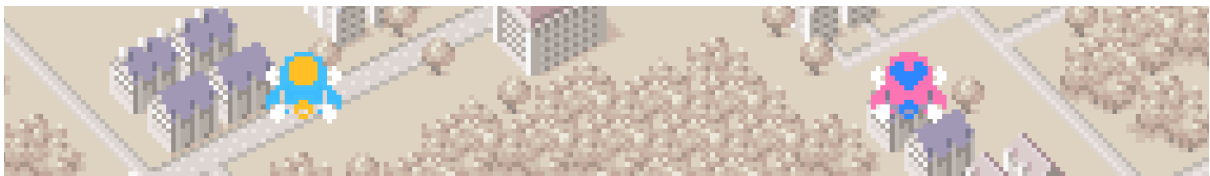


Figura 7. Twinbee y Winbee centrados en sus respectivas mitades de la pantalla de juego al comenzar el nivel

1.4 Los disparos (2 puntos)

Los jugadores pueden disparar pequeñas balas pulsando sus respectivas teclas con el objetivo de destruir a los enemigos. La velocidad inicial de disparo es de 1 bala por segundo.

Si un enemigo colisiona con una bala, éste será destruido y la bala “desaparecerá”. La bala también “desaparecerá” si sale fuera de la escena. Para conseguir la máxima puntuación en este apartado y para evitar que con cada disparo se creen y destruyan nueva balas, debe usarse una **pool** de objetos que contendrá una capacidad máxima de 100.



Figura 7. Disparo del jugador (bullet.png)

Hay diferentes tipos de disparo según la cantidad de power ups que consiga el jugador (ver sección 1.7 Power Ups).

**Conseguir los diferentes tipos de disparo y sus trayectorias será evaluado como parte de este apartado*

1.5 Los enemigos (1.5 puntos)

Los enemigos están compuestos por un spritesheet de 4 frames que crea una animación de rotación. Los enemigos son naves que aparecen en la parte superior de la escena en una posición X aleatoria y van hacia la parte de abajo del jugador formando un movimiento armónico simple.

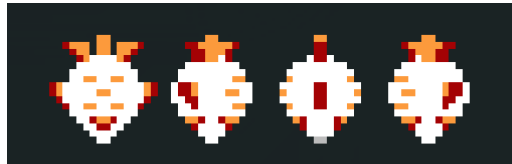


Figura 8. Nave enemiga (enemy.png)

Los enemigos colisionan con el jugador y con las balas que disparan. Si un enemigo colisiona con el jugador, el jugador será destruido. En caso de que el enemigo sea alcanzado con una bala, el enemigo será destruido formando una explosión y reproduciendo el archivo de sonido “explosion.wav”.

La animación de la explosión se repite 2 veces antes de desaparecer del mapa.

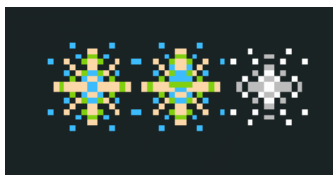


Figura 9. Explosión generada por un enemigo al ser golpeado por una bala del jugador (explosion.png)

1.6 Victoria y derrota (1 punto)

Cuando un jugador es destruido (colisiona con algún enemigo) se reproduce el sonido “dead.wav”. Si todos los jugadores en juego son destruidos entonces el/los jugador/er habrán perdido y se verá el texto “Defeat en pantalla” durante 3 segundos antes de volver a mostrar la escena de título.

Por otro lado, si algún jugador alcanza la parte superior del nivel, estos habrán ganado y se mostrará el texto “Victory” durante 3 segundos, a la vez que el/los jugador/es aún vivos salen por la parte superior de la pantalla.

Al igual que pasa con el título del juego en la escena del menú, los textos de victoria y derrota tienen un borde de un color generado aleatoriamente.



Figura 10. Vista del juego al ganar y perder

1.7 Power Ups (2 puntos)

A lo largo del recorrido los jugadores pueden conseguir power ups para mejorar sus disparos e incluso reducir la cadencia de disparo.

Un jugador consigue un power up si pasa por encima de este. Estos power ups se mueven a la velocidad del propio fondo haciendo que parezca que están estáticos en el nivel.



Figura 11. Power up (greenbell.png)

Los primeros 2 power ups que se consigan aumentarán las balas que dispara el jugador de la forma indicada en la figura 12. A partir de aquí cada power up reducirá la cadencia de disparo del jugador en 200ms. Además los power ups no son compartidos por varios jugadores, es decir, mientras que el jugador 1 puede estar disparando una única bala, el jugador 2 puede disparar 4 si ha conseguido 2 power ups.

Una vez un jugador consigue un power up, éste desaparece y se reproduce el archivo de sonido "lucky.wav". Un mismo power up no puede ser cogido por dos jugadores.

Por último, el sprite del power up tiene una animación generada por tween (y no por frames) en la que se cambia su ángulo para dar la sensación de que el cascabel se mueve.

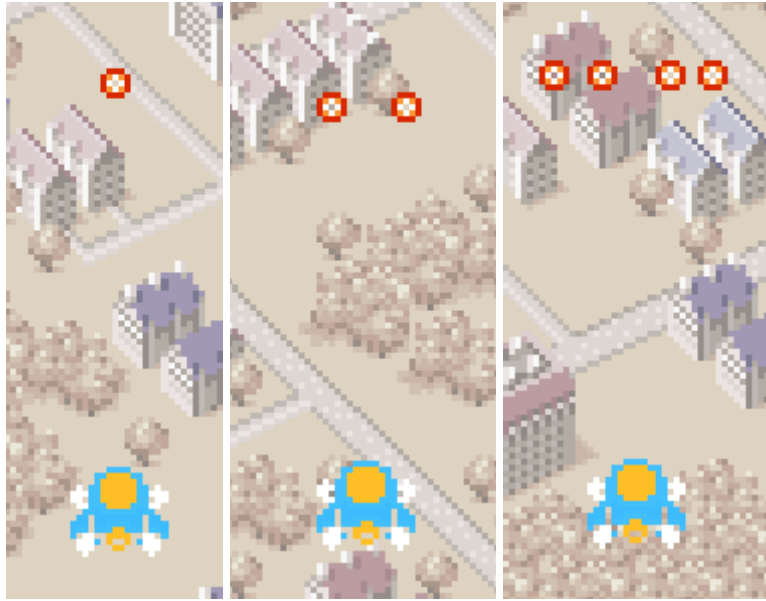


Figura 13. Tipo de disparo según la cantidad de power ups conseguidos por el jugador. (izquierda: sin powerUp, disparo normal compuesto por una bala con trayectoria recta en el eje Y; centro: 1 powerUp, disparo doble compuesto por dos balas con una pequeña desviación (1°); derecha: 2 powerUp, disparo avanzado compuesto por 4 balas, 2 de ellas, las centrales, con trayectoria recta en el eje Y y otras dos balas con desviación e 3°.

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 jugador capaz de moverse, de ganar, de disparar y matar un enemigo estático, y de morir si colisiona con él. Lo siguiente sería completar el menú de juego, añadiendo un segundo jugador, haciendo la generación de enemigos y su movimiento. Lo siguiente sería añadir la opción de conseguir mejoras para el ataque de los jugadores. Y por último vendría añadir el resto de sonidos, animar el power up y ajustar las velocidades de movimiento de los diferentes elementos de juego.

PD: El ejemplo del juego que se proporciona contiene varios trucos para facilitaros comprobar las mecánicas. Implementar **estos trucos no es necesario 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.
- “P” → Aumenta la velocidad de avance del nivel, permitiendo completarlo en mucho menos tiempo.
- “U” → Mejora el disparo del jugador 1 en un nivel. Después del nivel 3 disminuye la cadencia de disparo.

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, 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, 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 en un archivo comprimido llamado **“twinbee.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 **sprites.txt** con el tamaño de los frames en cada spritesheet.

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.

6 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.