

Tecnología de la programación de Videojuegos II

Examen Final, 21 de mayo de 2025

Duración: **3 horas**

Puntuación máxima del examen: **6 puntos**

Instrucciones

- Descarga desde el campus virtual los archivos **enunciado.pdf**, **ExamenTPV2.zip** y tu entrega de la práctica 2 para el examen, y cópialos en la carpeta **hlocal** (es muy importante usar **hlocal**, porque el contenido de **Downloads** se borra automáticamente).
- Descomprime el archivo **ExamenTPV2.zip** en **hlocal**. Esto creará una carpeta llamada **ExamenTPV2** con el siguiente contenido:
 1. Un archivo **EXAMEN.txt** en el que debes escribir: 1) tu nombre completo y 2) para cada pregunta, una explicación muy breve de la solución.
 2. Varias carpetas **Pi** (p.ej., **P1**, **P2**, etc.). Cada una incluye ya un proyecto de **Visual Studio** que debes usar para la pregunta correspondiente (sólo para preguntas que no requieren la modificación de la práctica 2). *Separando las preguntas en proyectos es más cómodo para no tener problemas si la solución de una pregunta no compila, etc.*
 3. Una carpeta con el nombre **PacMan** en el que debes descomprimir tu práctica 2, y que debes usar para las preguntas que requieren la modificación de la práctica 2. **Si tu entrega no incluye todo el proyecto, puedes descargar la plantilla **TPV2.zip** y **ecs_5.zip** para montar el proyecto.** Es muy importante tener tu práctica en esa carpeta para que todo el exámen esté bajo la carpeta **ExamenTPV2**.
- **Para entregar tu examen:**
 1. Limpiar todos los proyectos de **Visual Studio** (usando **Build -> Clean Solution**).
 2. Borrar la carpeta **".vs"** de todos los proyectos **Visual Studio**.
 3. Descomprimir la carpeta **ExamenTPV2** en un archivo **TuNombre.zip**. El tamaño del archivo debe ser inferior a **50 MB**; si no, es que se te ha olvidado borrar algo.
 4. Haz doble clic en el icono del escritorio denominado **"EXÁMENES en LABs entregas"**, y dentro de la ventana que aparece, doble click en **"ALUMNOS entrega de prácticas y exámenes"**. Se abre otra ventana en la que debes seleccionar el archivo **zip** en el panel inferior izquierdo y arrastrarlo al panel inferior derecho (o utiliza el botón derecho del ratón y la opción Subir).
 5. Antes de abandonar el laboratorio debes pasar por el puesto del profesor para asegurarte de que lo que se ve en el puesto del profesor es lo que has entregado y firmar en la hoja de entregas.

Pregunta 1 [1.5 puntos]

Un *fantasma clonable* es un fantasma que puede clonarse durante el juego, mediante el siguiente mecanismo:

1. El *fantasma clonable* se puede clonar hasta N veces durante el juego. Donde N es un número aleatorio, entre 1 y 5, que se le asigna al fantasma en el momento en el que se crea. Los fantasmas clonables pueden tener valores diferentes para N .
2. La clonación se realiza M milisegundos después de su nacimiento o de su última clonación. El valor de M es aleatorio entre 1000 y 10000, y debe cambiarse cada vez que se clona.
3. La posición del clon debe ser la misma que la del fantasma del que se clonó.
4. El clon debe ser un fantasma normal, es decir, no clonable.

Modifica la práctica 2 para que cuando se crea un fantasma, con probabilidad P (p.ej., el 10%) se crea un fantasma clonable y con probabilidad $100\%-P$ se crea uno normal. Definir los rangos de N y M , y la probabilidad P como **constantes** para poder modificarlos fácilmente si es necesario depurar usando otros valores.

Pregunta 2 [2 puntos]

El *escudo* del PacMan consiste en 4 estrellas que se colocan alrededor del PacMan para protegerlo de los fantasmas. Cuando chocan con un fantasma, este desaparece (el fantasma). La imagen de la estrella está en la carpeta `P3/resources/icons`, si no la tienes ya.

Si p es la posición que corresponde al centro del PacMan, cuando el escudo está activo, las posiciones de las estrellas deben de ser: $p+(X,0)$, $p-(X,0)$, $p+(0,X)$, y $p-(0,X)$. Ajusta el valor de X y el tamaño de las estrellas, en función del tamaño del PacMan en tu práctica.

No te preocupes si alguna coordenada es negativa, es decir si alguna estrella sale de la ventana. Recuerda que las estrellas deben cambiar de posición cuando lo hace PacMan, es decir, siempre se mueven con él. Modifica la práctica 2 para que tenga el siguiente comportamiento:

1. El escudo se activa si se cumple la siguiente condición: el PacMan come N frutas en menos de M milisegundos, fuera del periodo de inmunidad, y el escudo no está activo. Conviene guardar los tiempos en los que se han comido las últimas N frutas para verificar esta condición (es decir, comprobar si se han comido cinco y si el más antiguo se ha comido antes de que pasen M milisegundos). Mandar mensaje de activar escudo
2. El escudo se desactiva si se cumple una de las siguientes condiciones: han pasado T milisegundos desde su activación, se activa el periodo de inmunidad, o acaba la ronda. Cuando se desactiva el escudo, la cuenta de las N frutas debe resetear a 0.

Declara N , M y X como **constantes**, con los valores que quieras para poder depurar el programa fácilmente. Una buena solución debe hacer los mínimos cambios en los sistemas existentes.

Pregunta 3 [1.5 puntos]

El código está en la carpeta P3.

El código de esta pregunta incluye un juego sencillo, en el que el **PacMan** puede hacer 2 acciones:

1. Se mueve hacia delante al pulsar la tecla «↑», y para cuando choca con el borde o si se pulsa la tecla «SPACE».
2. Se gira hacia la derecha o hacia la izquierda pulsando, respectivamente, «←» o «→», y para si se pulsa la tecla «SPACE».

Nótese que mientras se está realizando una acción no se puede hacer la otra, hasta que se pare.

El código es un poco complicado porque usa muchas instrucciones condicionales. Modifícalo para que utilice el patrón de diseño «**State**» (estado) y así simplifiques el flujo de control correspondiente. Las clases nuevas que crees se deben colocar en la carpeta P3/src/game.

Pregunta 4 [0.5 puntos]

El código está en la carpeta P4.

El código proporcionado representa un juego que consiste en una clase **Game** y una clase **GameObject**. No es un juego de verdad, es solo para simular 2 problemas.

1. Modifica el código para que no haya fugas de memoria dinámica sin alterar la «lógica» del juego. No se puede guardar todos los objetos creados en una lista para eliminarlos al final. Nótese que las instancias de **GameObject** pueden compartir recursos.
2. Modifica el código para que funcione correctamente si usamos un bucle **for-each** en lugar del **for** en el método **start** de la clase **Game**. (el código del bucle **for-each** está en comentarios). Esto se puede conseguir si los objetos que se añaden al juego en una iteración, entran al juego solo en la siguiente.

Pregunta 5 [0.5 puntos]

El código está en la carpeta P5.

Implementa los siguientes métodos en la clase **MyVector**: 1) constructor de copia; 2) constructor de movimiento; 3) operador **operator=** de copia; y 4) operador **operator=** de movimiento.

Nótese que el destructor de la clase **MyVector** está comentado, ya que puede causar errores si dichos métodos no se implementan correctamente. Por tanto, cuando hayas implementado dichos métodos, deberás descomentar el código del destructor.