



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Verificación y Validación

Curso 2024-2025

Dpto. Sistemas Informáticos



ETS I SISTEMAS
INFORMÁTICOS

SPACE INVADERS

Objetivo de la práctica

En esta práctica, vais a realizar el proceso de análisis, depuración y testing del código del juego “Space Invaders”. Para ello, se plantea un proceso por niveles, similar a como funcionaría en un videojuego. Estos niveles son incrementales, es decir, es necesario haber superado el anterior para poder realizar correctamente el siguiente, ya que de lo contrario se irán arrastrando los errores y será cada vez más complicado. Para facilitar el proceso de entrega y corrección, los niveles se han dividido en dos bloques. El bloque 1, que abarca los niveles 1 y 2; y el bloque 2, que abarca los niveles 3, 4 y 5. En la siguiente tabla se recoge la repartición de niveles por bloques, así como las fechas de entrega de estos:

Bloque 1 Fecha límite de entrega: 15 de noviembre	Nivel 1. Análisis dinámico – Pruebas de caja negra
	Nivel 2. Análisis dinámico – Pruebas de caja blanca
Bloque 2 Fecha límite de entrega: 13 de diciembre	Nivel 3. Depuración de código
	Nivel 4. Pruebas de integración
	Nivel 5. Pruebas de sistema

Al final de la práctica, el juego debería funcionar correctamente, y todos los fallos de código deberían ser detectados y corregidos. Además, se debería de aportar un conjunto de pruebas realizado que sea suficientemente extenso como para permitir verificar de la manera más completa posible la funcionalidad del juego.

La superación de cada nivel llevará aparejada la asignación de una insignia en Moodle. Además, se otorgarán premios especiales a aquellos grupos que obtengan mejores resultados en cada uno de los niveles.

Instrucciones generales

La práctica se realizará de acuerdo con las siguientes indicaciones generales:

- **Los grupos serán de, como máximo, 5 personas.** Se pueden hacer grupos de menos personas, pero no se modificará la carga del trabajo ni el contenido de la práctica para adaptarse a las circunstancias de cada grupo.
- **La práctica se realizará en Java, utilizando Maven para la construcción del entorno.** Se aportará un fichero pom.xml para la generación del entorno. Se debe utilizar este fichero para asegurar la correcta ejecución del código y de los posteriores ficheros de prueba.
- **La práctica se entregará en un único fichero .zip,** donde se incluirán todos los ficheros desarrollados y necesarios para su corrección.



POLITÉCNICA

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Verificación y Validación

Curso 2024-2025






Dpto. Sistemas Informáticos



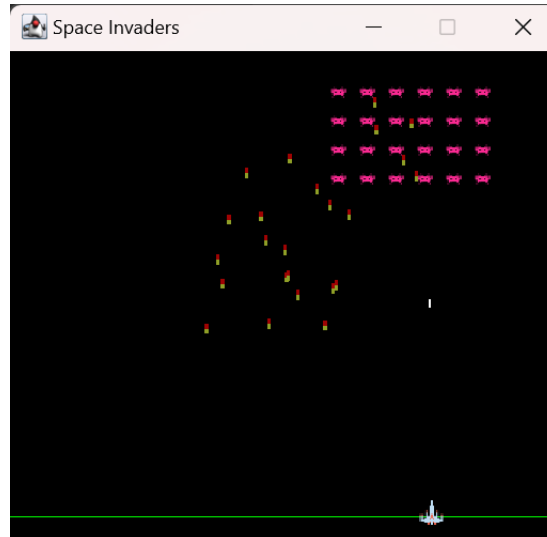
ETSÍ SISTEMAS
INFORMÁTICOS

Contexto de la práctica

En esta práctica, trabajaréis sobre el popular juego de arcade Space Invaders. Este juego consta de los siguientes elementos:

Icono	Nombre	Descripción
	Alien	Los alienígenas son los enemigos del juego. Su objetivo es destruir al jugador mediante disparos. Si los alienígenas llegan hasta la nave, ganarán el juego.
	Bombas	Los alienígenas pueden lanzar bombas al jugador para intentar derrotarlo. Si le alcanzan, el jugador es eliminado.
	Jugador	El jugador se representa mediante una nave espacial. Su objetivo es disparar y destruir a todos los alienígenas al tiempo que evita que le alcancen las bombas.
	Disparos	La nave espacial puede disparar proyectiles para destruir a los alienígenas.
	Explosión	Cuando un alienígena o un jugador es alcanzado, se produce una explosión que indica que ha sido eliminado.

A continuación, se muestra una captura de pantalla del juego donde se observan los elementos:



El objetivo principal del juego es destruir todos los alienígenas antes de que estos destruyan la nave o alcancen el suelo, es decir, completen la invasión. La nave se puede mover de izquierda a derecha y puede disparar un proyectil cada vez. Por el contrario, los alienígenas además de moverse hacia los lados se van moviendo hacia abajo, acercándose cada vez más al suelo. Los alienígenas pueden disparar bombas con el fin de eliminar al jugador.

La mecánica del juego de manera detallada es la siguiente:

1. **Movimiento de la nave:** La nave sólo puede moverse de izquierda a derecha, dentro de los límites del tablero (es decir, una vez alcance los bordes izquierdo y derecho del tablero se quedará quieta).
2. **Mecánica de disparos:** La nave puede disparar únicamente hacia arriba. Sólo se puede tener un disparo por pantalla cada vez, es decir, hasta que el disparo actual no desaparezca no se podrán disparar más.

proyectiles. El disparo desaparece si: 1) desaparece por la parte superior de la pantalla, o 2) alcanza a alguno de los alienígenas o de sus bombas.

3. **Movimiento de los alienígenas:** Los alienígenas se desplazan en formación, moviéndose de izquierda a derecha a la vez que avanzan hacia la parte inferior de la pantalla. Cuantos más alienígenas son destruidos, más rápido avanzarán hacia el inferior de la pantalla. Si los alienígenas alcanzan el inferior de la pantalla, el jugador pierde la partida.
4. **Bombas:** Los alienígenas disparan bombas para eliminar al jugador. Si alguna de estas bombas alcanza al jugador, el jugador quedará eliminado. Por el contrario, si uno de los disparos del jugador alcanza la bomba, la bomba se destruirá.
5. **Finalización:** El jugador ganará el juego cuando todos los alienígenas hayan sido eliminados.

El código de la práctica consta de la siguiente distribución de ficheros:

Directorio	Fichero	Descripción
\main	<i>Board.java</i>	Crea la pantalla de juego, la configuración y coloca los elementos
	<i>Commons.java</i>	Indica las constantes del juego: tamaños, posiciones, etc.
	<i>Main.java</i>	Fichero main del juego (Se aporta para el segundo bloque)
\sprites	<i>Alien.java</i>	Construcción y métodos de los alienígenas y sus bombas
	<i>Player.java</i>	Construcción y métodos del jugador
	<i>Shot.java</i>	Construcción y métodos de los disparos
	<i>Sprite.java</i>	Clase de la que heredan las clases Alien, Player y Shot. Establece métodos comunes usados por todos los elementos del juego, como los “getters” y “setters” de sus posiciones en pantalla.

En total, hay 3 ficheros afectados por defectos de código, habiendo un total de 10 defectos de código repartidos por los diferentes ficheros. Puede haber más de un defecto en un mismo método.