

COMPUTING AND SYSTEMS DEVELOPMENT (VIDEOGAMES PROGRAMMING)	1VGP	Profesor titular / Asesor	<b>Federico Sanjuan</b>
		Fecha asignación: / Starting date:	21/03/18
<b>METODOLOGIA DE PROGRAMACION</b>		Fecha entrega: / End date:	Convocatoria Mayo 2018
<b>Práctica 2: ASTEROIDS</b>			

<b>ALUMNO / STUDENT :</b>	
---------------------------	--

<b>PROFESOR / ASSESOR :</b>	Federico Sanjuan
-----------------------------	------------------

ENUNCIADO / ASSIGNMENT BRIEF	CHECKING		EVALUACIÓN / MARK	R	P	M	D
------------------------------	----------	--	----------------------	---	---	---	---

### Propósito de la práctica:

Implementar una versión en C, del clásico juego de Atari **Asteroids**.

### Escenario de la práctica:

Acabas de ser contratado como Programador Junior por la empresa de videojuegos "Divide et impera, S.L." La empresa se dedica al outsourcing para clientes de diversos tamaños. Estás en un periodo de pruebas de 3 semanas en el que debes demostrar tu capacidad de producción y desarrollo independiente de código en C, utilizando librerías gráficas desarrolladas por otros programadores de la empresa.

### Requisitos generales de la práctica:

La gestión de usuarios deberá implementarse en una aplicación independiente que se manejará a modo de administrador y con una gestión completa de registros de usuario para permitir jugar al Asteroids, recabando la siguiente información en el registro:

Nombre y apellidos, fecha de nacimiento, provincia, país, usuario y contraseña, créditos de partidas y dirección de email.

Las opciones en la administración de usuarios deberán permitir obtener un listado de todos los usuarios registrados, dar de alta, baja o modificación cualquier usuario, atendiendo a poder modificar cualquier información del mismo, así como su crédito de partidas necesario para poder jugar.

El juego inicialmente deberá presentar la posibilidad de identificarse (usuarios ya registrados) o bien registrarse, debiendo cumplimentarse todos los datos necesarios en registro y asignando por defecto un crédito de 10 partidas en cada alta.

En caso que el usuario ya esté registrado, deberá identificarse con su usuario y password, permitiendo que este usuario juegue siempre que disponga crédito para una nueva partida (1 crédito = 1 partida).

Una vez superado el proceso de validación se mostrará una pantalla con un listado de los últimos 10 mejores jugadores, fecha de la partida y su puntuación de forma ordenada, de mayor puntuación a menor y se iniciará el juego.

Una vez finalizada la partida, en caso de obtener una puntuación que se registre en el listado de los 10 mejores, se le informará al usuario y se le concederán 5 créditos más que deberán añadirse a su cuenta.

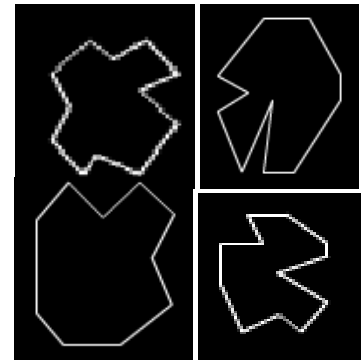


El juego programado deberá estar basado en la versión original de este arcade vectorial lanzado por la empresa Atari en 1979.

El juego mostrará una nave en pantalla que responderá a un control de aceleración (Cursor Arriba) y un control de fuego (Barra espaciadora), para avanzar será necesario ir en cualquier dirección izquierda (Cursor Izquierda) o derecha (Cursor Derecha) y acelerar produciendo un desplazamiento en la dirección apuntada por la nave.

Los bordes de la pantalla no son límites, por lo tanto cuando se alcancen, la nave deberá aparecer por el lado contrario y dispondremos de un botón de hiperespacio (Tecla G) con el que conseguiremos que la nave desaparezca de su posición actual y vuelva a aparecer en una posición aleatoria dentro de la pantalla, con la posibilidad de explotar si aparece ubicada dentro de un asteroide.

Disponemos de asteroides en pantalla de cuatro tipos y con un ángulo de presentación en pantalla distinto: con tamaño grande (20 puntos), tamaño medio (50 puntos) y tamaño pequeño (100 puntos), que darán su puntuación al ser impactados por los disparos de la nave.



Cuando sean impactados se dividirán en otros dos asteroides de tipo distinto, del tamaño inferior siguiente y que mostrarán en pantalla una trayectoria y velocidad distinta.

Los asteroides siguen un movimiento rectilíneo con aceleración constante, de forma que, cada uno de ellos mostrará una trayectoria fija, salvado los bordes de la pantalla de igual forma que nuestra nave hasta que sean destruidos.

Los asteroides en pantalla no colisionan entre ellos permitiendo en la partida que se superpongan en pantalla, siguiendo cada uno con su trayectoria y velocidad, lo que provocará que al impactar con dos asteroides superpuestos se destruyan ambos.



De forma periódica, aparecen dos tipos de ovnis, uno de tamaño mayor (200 puntos) que dispara en dirección aleatoria, con independencia de la posición de la nave, y otro con un tamaño pequeño (1000 puntos) que persigue a la nave con sus disparos. Ambos tipos de ovnis deberán aparecer cuando tenemos pocos asteroides en pantalla o bien cuando la nave pasa un periodo de tiempo sin disparar y sus disparos sí impactan con los asteroides provocan su ruptura sin añadir puntuación al jugador, al igual que si impacta contra un asteroide se destruye sin añadir puntuación.

El juego comienza con tres naves en reserva y una nave en la posición central de la pantalla y cuatro asteroides tamaño grande, y cada vez que se destruyen todos los asteroides, se limpia la pantalla y se aumenta un nivel de forma que aparecen más asteroides (2 más por nivel) hasta alcanzar un máximo de 12 asteroides grandes en pantalla.

Cuando la nave es alcanzada por un asteroide o por el impacto de un ovni se destruye y la partida sigue mientras se dispongan de naves de reserva.

Valoraciones a tener presentes en el desarrollo de la práctica final.

PASS. Juego operativo con gestión de usuarios para un jugador con ovnis, pero sin niveles.

MERIT: Requerimientos anteriores con niveles de partida.

DISTINCTION: Requerimientos anteriores con posibilidad de dos jugadores simultáneos de forma alterna y/o guardar partida en disco y recuperar partida.

## Tareas de la práctica:

### Tarea 1

Aplicación de la metodología a la hora de plantear y analizar la implementación a realizar con los recursos del entorno gráfico disponible y repositorio de imágenes gráficas.

Diseño de las estructuras de datos necesarias para la implementación de la solución diseñada.

### Tarea 2

Diseño e implementación de los procesos básicos para la carga de los elementos gráficos para un entorno del juego correcto y ajuste de las mecánicas básicas de animación dentro del entorno gráfico utilizado.

### Tarea 3

Implementación completa de la mecánica del juego para una ejecución correcta, según el nivel de desarrollo alcanzado en la entrega de la práctica.

### Tarea 4

Testeo, depuración y corrección de errores para la entrega de una aplicación lo más fiel al juego original, libre de errores.

### Tarea 5

La documentación a entregar en la práctica deberá contener la siguiente información:

1. **CODIGO FUENTE DEBIDAMENTE COMENTADO.** Los comentarios deberán estar incluidos en el código correspondiente, a elección del programador y donde se considere necesario la inserción de la explicación correspondiente de la estructura de control, expresiones o rutina utilizada en cada caso, siempre como documento necesario para una posible revisión del código posteriormente o para la ampliación del mismo por terceras personas.
2. **CODIGO OBJETO.** Programa ejecutable, **LIBRE DE ERRORES DE EJECUCIÓN**, que deberá poder ejecutarse en una máquina con SO Windows y bajo los requisitos mínimos indicados por el autor.
3. **MANUAL DE USUARIO.** Documentación gráfica, debidamente comentada, de todas las posibilidades de ejecución de nuestra aplicación, comentando en cada caso lo que se está ejecutando y completando una ejecución completa de todo el proceso de forma correcta y otra de todos aquellos posibles errores de ejecución que nuestra aplicación es capaz de capturar y darle el tratamiento de errores debido. Recordar que debería mostrarse en este manual todo aquello que es capaz de realizar mi aplicación, sin necesidad de ejecutarla.
4. **ESTRUCTURAS DE DATOS.** Detalle de todas aquellas estructuras de datos utilizadas en nuestra aplicación (matriz, vector, cadena, etc) con indicación de su punto de declaración, tipo de datos que contienen, justificación de su utilización y ámbito global o local, declarado para las mismas.

5. **MANUAL DE PROCEDIMIENTOS Y FUNCIONES.** Informe completo de los prototipos de los procedimientos y funciones creados por el programador y en su caso, parámetros de cada una de ellas y tipo devuelto, así como una descripción completa de la ubicación donde se han declarado y los puntos del código donde se realizan las llamadas estas.
6. **INSTRUCCIONES DE EJECUCIÓN E INSTALACION.** Deberá incluir de forma detalla los requisitos software mínimos y los pasos a seguir para ejecutar en un entorno de forma correcta y segura la aplicación.

Toda esta documentación deberá entregarse dentro de una carpeta que se grabará en una ubicación determinada por el profesor de la asignatura y que se habilitará para tal efecto el día de la entrega de prácticas.

La carpeta deberá nombrarse con el NIF del alumno, seguido de las iniciales correspondientes a su nombre y apellidos.

El formato de entrega será un documento Word, debidamente maquetado, con su índice de contenidos, numeración de páginas y listo para ser impreso si fuera necesario. (Recordad el tema de NO ENCONTRAR faltas de ortografía en la documentación redactada).

## Resumen de tareas

Tarea	Resumen de entregables
Tarea 1	Análisis previo de recursos, repositorios disponibles de entorno gráfico, y planteamiento del desarrollo del código a implementar utilizando programación procedural, Diseño de las estructuras de datos necesarias para la implementación de la solución diseñada.
Tarea 2	Carga de los elementos gráficos para un entorno correcto e implementación de las mecánicas básicas de animación
Tarea 3	Implementación completa de la mecánica del juego
Tarea 4	Todo el código fuente de la demo técnica, compilando correctamente y libre de errores.
Tarea 5	La documentación de la práctica y el manual de usuario, correctamente redactados,, siguiendo el esquema solicitado y citando todas las fuentes externas de información

## Feedback al alumno

<u>Fecha</u>	<u>Objetivo</u>	<u>Observaciones</u>
	Prueba de validación	
	Tarea 1	
	Tarea 2	
	Tarea 3	
	Tarea 4	
	Tarea 5	
	M1	
	M2	
	M3	
	D1	
	D2	
	D3	

## Criterios de evaluación

### PASS

<b>P1</b>	La asistencia obligatoria a todas aquellas clases marcadas como clases de desarrollo de la práctica.
<b>P2</b>	La entrega de la práctica dentro de los límites de plazo de entrega.
<b>P3</b>	La correcta realización de los objetivos básicos de la práctica según valoraciones del enunciado PASS

**Adicionalmente a los criterios anteriores de PASS, esta práctica da la oportunidad de entregar evidencias para conseguir las calificaciones de MERIT y DISTINCTION**

Descripción de la calificación	Contextualización
<b>M1 Identificar y aplicar estrategias para encontrar las soluciones apropiadas</b>	Para conseguir M1 debes demostrar que eres capaz de escoger suficientes ejemplos y estrategias para solucionar el problema planteado, especificando por qué escogiste dichos ejemplos de manera que se demuestre tu conocimiento en profundidad sobre el problema. Debes usar múltiples fuentes de conocimiento y claramente identificar los orígenes de tus ejemplos para mostrar tu razonamiento. Todo esto dentro de los plazos de tiempo establecidos por este documento de práctica.
<b>M2 Seleccionar, diseñar y aplicar los métodos y técnicas apropiadas</b>	Para conseguir M2 debes demostrar fehacientemente que has obtenido información de un abanico de fuentes y has seleccionado la más apropiada para formar parte de la solución a los problemas de programación de videojuegos planteados. Esta demostración puede incluir una explicación sobre por qué decidiste que dichos recursos eran apropiados y qué criterios has seguido para seleccionarlos. Todo esto dentro de los plazos de tiempo establecidos por este documento de práctica.
<b>M3 Presentar y comunicar los resultados de manera apropiada</b>	Para conseguir M3 debes presentar una memoria/presentación bien estructurada que sea fácil de entender tanto para usuarios técnicos como no-técnicos, que sea interesante e informativa. Debes usar un lenguaje correcto y apropiado, así como hacer buen uso de imágenes, tablas, etc... La memoria debe estar realizada dentro de los plazos de tiempo establecidos por este documento de práctica.
<b>D1 Usar una reflexión crítica para evaluar el trabajo propio y justificar conclusiones válidas</b>	Para conseguir D1 debes entregar una reflexión crítica en la que: hayas llegado a conclusiones a través de síntesis de ideas y de manera justificada; la validez de los resultados haya sido evaluada usando criterios standard; se incluya un análisis crítico de posibles mejoras que aportar al trabajo para mejorar su calidad de manera realista.
<b>D2 Comportamiento responsable a la hora de gestionar y organizar tareas</b>	Para conseguir D2 debes demostrar tu propia autonomía/independencia a la hora de desenvolverte y organizarte en tareas sustanciales, tareas de investigación planificadas y tu capacidad de reacción ante imprevistos.
<b>D3 Demostrar pensamiento lateral, convergente o creativo</b>	Para conseguir D3 debes demostrar que has generado tus propias ideas y tomado tus propias decisiones en la práctica, que has llevado a cabo una auto-evaluación, que has aplicado pensamiento lateral o convergente, que te has mostrado receptivo ante nuevas ideas y que has demostrado desenvolverte con comodidad y rapidez en un contexto nuevo.

## Objetivos del módulo / Unit Outcomes

### PASS

✓	<b>Aplicar análisis y técnicas de diseño al proceso de desarrollo del software</b>	<p>Análisis: Obtener los requisitos y datos necesarios de una especificación de programa</p> <p>Diseño: Desarrollo de una solución para la especificación de un problema siguiendo la fase de análisis, técnicas, pseudocódigo, y representación de diagramas</p>
✓	<b>Desarrollar un alto nivel de código usando un apropiado lenguaje de programación</b>	<p>Lenguaje de programación: desarrollo en un lenguaje de programación, desarrollo de texto, facilidades y herramientas de un lenguaje de programación, pull-down, menús, técnicas de edición, guardar ficheros, compilación de código, ayudas; buena práctica de programación (comentarios, corrección de código, identificadores...)</p> <p>Representación de datos: tipos de datos, enteros, reales, boolean, caracteres y cadenas; declaración de constantes y variables con un alto nivel de programación</p> <p>Desarrollo del programa: estructuras simples de programación, reglas sintácticas, operadores, sentencias de entrada/salida, implementar sentencias lógicas (sentencias de selección, condicionales, bucles, criterios de salida de un bucle)</p>
✓		Programas: propósito, módulos, ejemplos de diseño procedural, diseño simple de programas, implementación simple de programas (sintaxis, paso de parámetros, invocación, declaración de datos públicos y privados), funciones
✓	<b>Producir una apropiada documentación para una determinada solicitud del programa</b>	<p>Presentación de documentación: utilizar procesador de textos o gráficos, análisis, diseño e implementación de documentación, estándares profesionales, necesidades de la empresa</p> <p>Documentación de usuario: para especificar el desarrollo del programa y su utilización</p>
✓		Documentación de programa: documentación que cubra los aspectos técnicos que contiene la aplicación, incluyendo algoritmos, datos, tablas, estructuras de control, estructuras utilizadas, uso de interface
✓	<b>Utilizar métodos de testeo para aplicar correcciones en cada parte del código</b>	Método de testeo: planes de testeo y casos de testeo, métodos de caja blanca y caja negra, métodos de análisis de valores de variables, trazas.

### MERIT

✓	Se han utilizado otros métodos y técnicas para recopilar, analizar y procesar la información
✓	Se han usado técnicas y teorías relevantes en el análisis del conocimiento
✓	Se ha presentado coherentemente el trabajo utilizando un lenguaje técnico apropiado

### DISTINCTION

✓	Se ha validado/contrastado la información recopilada
✓	Se han evaluado y resumido las técnicas y teorías usadas para generar las conclusiones/solución
✓	Se ha presentado coherentemente el trabajo utilizando un lenguaje técnico de forma fluida

Material de apoyo
<p><u>Software:</u></p> <p>Compilador modo consola Visual Studio</p> <p>Editor de código Atom, notepad++, o similares a elección del alumno.</p> <p>Librería gráfica ESAT mínimo rev. 218, facilitada en clase para la implementación de ejercicios propuestos.</p>
<p><u>Bibliografía:</u></p> <p>Apuntes de clase y documentación correspondiente a la operativa Visual Studio</p>
<p><u>Web:</u></p>