

Informe Trabajo Práctico N°2  
*Atari Asteroids*

Malena Díaz Falvo y Agustín Ormazábal

30 de Junio de 2019

## Introducción

El actual proyecto consta de la implementación de una reversión del juego Asteroids en lenguaje ISO-C99 utilizando la biblioteca gráfica SDL2. Para su recreación se implementó el concepto de TDA, entre otras funcionalidades incorporadas a lo largo del curso.

## Desarrollo

Para abordarlo, se utilizaron y modificaron algunos de los archivos brindados para el TP1 (reversión del Lunar Lander). Además, se diseñaron y utilizaron diferentes TDAs para los objetos del juego, obteniendo finalmente los siguientes .h:

- caracteres.h
- config.h
- dibujar.h
- diccionario.h
- movimiento.h
- nave.h
- asteroides.h
- disparos.h
- graficador.h
- iterador.h
- lista.h

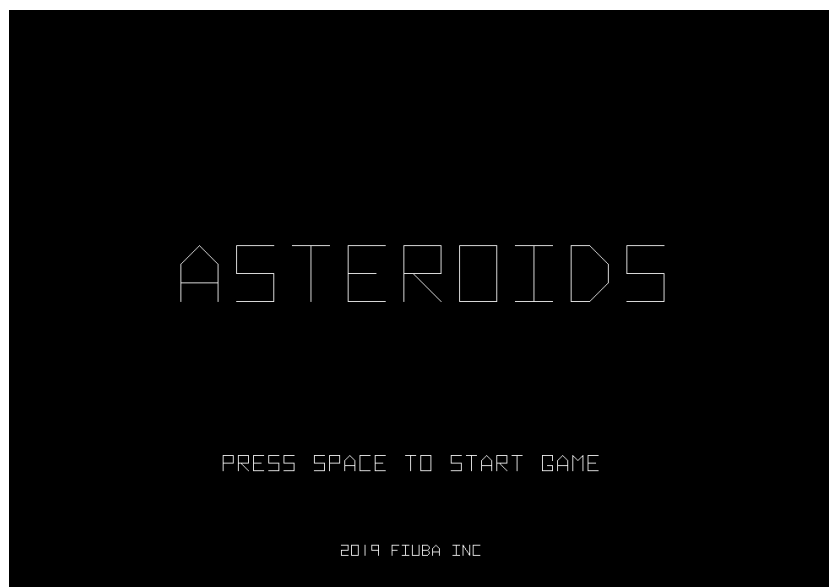


Figura 1: Pantalla de inicio del juego.

Se confeccionaron dos pantallas de inicio. Una para el comienzo del juego, donde se observa el nombre del mismo, junto con las instrucciones del inicio de la primera partida jugada (Figura 1). Al iniciar, se disponen de 4 vidas, las cuales se grafican en pantalla,

debajo del puntaje actual (Figura 2), para que esta finalice, se espera a que la o el usuario pierda las vidas iniciales.



Figura 2: Partida en transcurso.

Una vez finalizada, se crea una pantalla de fin de partida (Figura 3), donde se proyecta el lema **GAME OVER**, junto con el puntaje recién alcanzado, y el máximo puntaje obtenido desde la ejecución del juego. Además de la condición para comenzar una nueva partida.

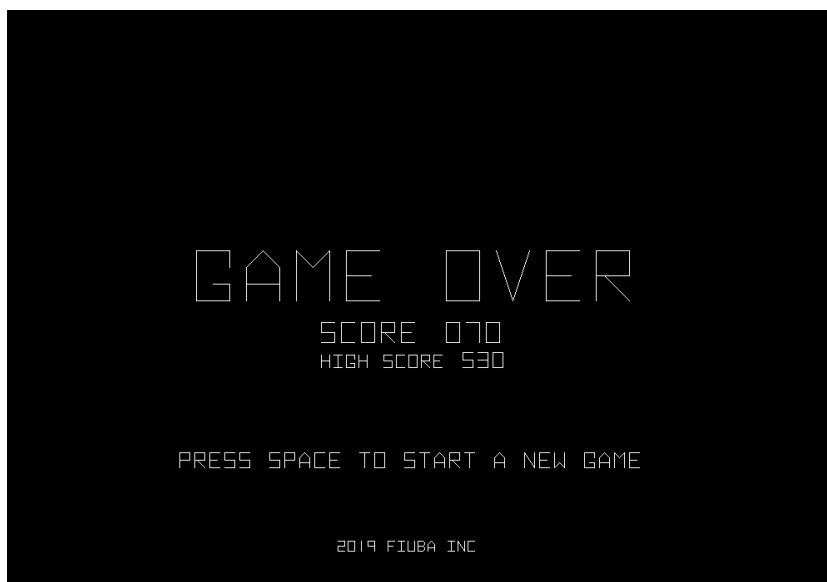


Figura 3: Pantalla de finalización de partida.

## Dificultades

Se han tenido diversas dificultades durante el desarrollo del juego. En un principio al implementar la función `bool graficador_inicializar(const char *fn, SDL_Renderer *r);`, encargada de levantar el archivo de sprites, se cometieron diversos errores respecto a su construcción. Para poder solucionarlo se necesitó de una profunda comprensión del manejo de archivos binarios. Una vez implementada, se utilizó con el fin de leer los distintos tipos de sprites contenidos dentro del archivo `sprites.bin` brindado por la cátedra, siendo posteriormente almacenados cada uno en una estructura.

En un principio, dentro de la función mencionada, se llamó a la función `realloc();` para tener como resultado un vector redimensionado el cual contenía las distintas estructuras de sprites, luego de evaluar las posibles implementaciones, optamos por adicionar los mismos a una lista local al módulo, sin la necesidad de utilizar un vector redimensionado.

Otra de las dificultades abordadas fue cómo encarar la modularización del proyecto, por malinterpretar cuestiones del enunciado, en un comienzo se utilizaron listas dentro de los TDAs Disparos y Asteroides, complejizando mucho más el desarrollo del código. Así que, luego de consultar las dudas, se replanteó y corrigió la estrategia de modularización.