

## **Descripción:**

Se debe programar un juego para computador cuya ambientación es el espacio estelar. En el mismo hay una nave espacial controlada por el usuario. La misma es de un color distintivo (seleccionado por Ud.), que se puede desplazar vertical y horizontalmente dentro de los límites de la pantalla de juego. Conforme pasa el tiempo, aparecen enemigos en forma aleatoria, en diferentes partes del escenario (bordes), cada uno con tiempo de juego también aleatorio y solo desaparecen después de haber disparado al menos una vez o al recibir un disparo del usuario.

## **I. Mecánica de juego**

La cantidad de balas o municiones son ilimitadas para el usuario. La nave puede cambiar de posición mediante un sistema de entrada de comandos de movimientos (Definido por Ud. Un ejemplo sería entrada de teclado, botones en pantalla, consola, etc). Existe un gatillo para lanzar la munición. La trayectoria de las balas va desde la posición donde está el cohete o la nave apuntando hasta el límite del tablero. Si en la trayectoria de las balas hay X cantidad de enemigos se obtienen  $X \times 2$  puntos por el disparo que acierte. Una vez sale la munición de la nave los enemigos podrían permanecer fijos hasta que la bala llega al límite del tablero en caso de que lo considere adecuado. La frecuencia de disparo debe tener un máximo establecido, es decir, entre cada disparo hay un tiempo que evita un comportamiento “láser” de la munición. Los enemigos también son capaces de disparar. Si dos balas de cañón chocan estas se anulan entre sí. La rapidez con que los vehículos enemigos aparecen va de acuerdo con el nivel de dificultad ajustado. Debe tener al menos 3 niveles de dificultad, el nivel más difícil otorga más puntaje por enemigo destruido. La nave tiene 3 puntos de vida. Al ser impactada por una bala de un enemigo, o colisionar con un enemigo, pierde un punto de vida. Al llegar a 0, se le comunica al jugador que ha perdido el juego.

Se debe desplegar una pantalla de presentación, donde se muestre el nombre del juego, las opciones que tiene y un botón de finalización del programa. Debe poseer un fondo musical agradable y con un volumen adecuado para no perturbar a los demás. En esta pantalla se captura el

nombre y podría establecerse el nivel de dificultad. Una vez seleccionado se debe retroalimentar al usuario informando cuál nivel seleccionó. Desde esta pantalla se debe poder acceder a la pantalla de juego, la pantalla de mejores puntajes y la ventana de información complementaria.

## **II. Dificultad**

Ud. Deberá definir qué aspectos cambia para demostrar un aumento de dificultad en el juego conforme pasan los niveles, además de la velocidad de movimiento de los enemigos. Ejemplos de esto son variación en tipo y comportamiento de los enemigos (trayectoria, vida, frecuencia de disparo) y restricciones al usuario (limitación de cantidad de munición, frecuencia de disparo permitida, velocidad de la nave). El cambio de nivel debe ser notable en jugabilidad. Si bien la dificultad puede ser retadora, el juego no debe ser imposible de ganar.

## **Pantallas**

### **I. Pantalla de escenario de batalla y botonera**

Es necesario retroalimentación para informar al usuario del estado del juego en todo momento: objetivos alcanzados, puntaje, cantidad de munición restante y toda la información necesaria para sentirse comfortable jugando. Los obstáculos bloquean los disparos, estos pueden ser fijos por nivel, aunque se puede definir que los mismos también aparezcan en forma aleatoria conforme avanza el nivel. La pantalla debe tener un fondo agradable que invite a jugar. La cantidad de enemigos en pantalla (un valor entre 1 y 9) debe mostrarse también en esta pantalla.

### **II. Pantalla de los Mejores Puntajes.**

Esta pantalla muestra los siete mejores puntajes. Se va actualizando una vez que se han agotado las municiones, se finaliza la sesión de juego o se quiere iniciar con otro jugador.

- Las funciones de recorrer la lista de objetivos o enemigos, la función de evasión, la colocación de obstáculos entre otras debe ser RECURSIVAS. Funciones de estas características y cuya naturaleza puede ser resuelta recursivamente, no se aceptará con iteración. Desviarse de esta directriz implicará la anulación de esa parte del código mediante comentarios en él.

- Cualquier clase de copia de código será sancionada de acuerdo con el reglamento vigente. Código adoptado para el manejo de interfaz debe especificarse claramente la fuente y reconocer los créditos. Está prohibida la copia de código que involucre la solución lógica general del algoritmo.