

## Tarea 2: Tanques booteable.

### 1. Motivación.

Un cargador booteable (boot loader) es un programa especial que es ejecutado desde algún dispositivo, en el arranque del computador, el cual es muy específico al hardware y arquitectura del CPU, por lo que su funcionalidad es muy particular. Un proceso booteable es necesario en cualquier sistema operativo para que pueda iniciar el mismo, por ende, es de suma importancia que un ingeniero en computadores esté relacionado con este tipo de aplicación. La idea fundamental es realizar una pequeña aplicación programada en ASM x86, la cual sea ejecutada en el booteo del ordenador.

### 2. Objetivo.

Comprender los conceptos y flujo de información de los procesos en el booteo de una computadora mediante la implementación de un programa realizado en ensamblador.

### 3. Información general

- Valor del proyecto: 5%
- Nombre del programa: Tanques booteable.
- Este proyecto se puede implementar en grupos de 4 estudiantes.
- Cualquier fraude detectado significará una nota de cero y su proceso respectivo.

### 4. Requerimientos funcionales.

Para la implementación de este proyecto será necesario considerar lo siguiente:

- ✓ Se implementará un programa (boot loader) realizado en ASM x86 el cual sea capaz de ejecutar una aplicación llamada "Tanques boteable".

#### **A. Tanques boteable.**

La idea fundamental de este juego es crear una serie de laberintos o obstáculos por los cuales una cierta cantidad de tanques buscarán matar un águila la cual está protegida por paredes, las cuales se desvanecerán si reciben balan por parte de la casa.

El juego contará con un tanque especial, el cual es dirigido por el usuario y su función es derribar a los otros tanques para resguardar el águila. Este tanque posee balas infinitas para derrotar a los otros.

El juego tendrá tres niveles de dificultad, los cuales varían en la velocidad con la que se mueven los tanques enemigos y en añadirle

fortaleza a lo mismos, es decir, que tengan que recibir más de un disparo para ser eliminados, además los obstáculos deben de variar de posición (tanto la velocidad como el laberinto quedarán a creatividad del estudiante). Para pasar de nivel, es necesario derrotar al menos 4 tanques por nivel si el águila recibe un disparo se acaba el juego.

El juego deberá proveer en todo momento la cantidad de tanques derrotados y el nivel de juego (principiante-medio-avanzado) que se está llevando a cabo. También es importante que se muestren los comandos en pantalla como parte de la guía para el usuario.

Cuando el jugador gana el tercer nivel, el juego permitirá la opción de reiniciar o salir, además debe de desplegar un mensaje en pantalla con alguna animación (movimiento) de felicitaciones por haber ganado.

Se podrá reiniciar el juego en cualquier instante, así como finalizarlo. Cabe resaltar que cuando se inicia el juego el usuario debe de aceptar una confirmación para empezar a jugar.

El tanque se moverá en las direcciones que le indique el usuario, los posibles comandos son:

- Flecha derecha: El tanque se moverá hacia la derecha.
- Flecha izquierda: El tanque se moverá hacia la izquierda.
- Flecha abajo: El tanque se moverá hacia abajo.
- Flecha arriba: El tanque se moverá hacia arriba.
- Espacio: El tanque disparará.
- L: Pausa el juego y lo reanuda de la misma manera.
- R: Reinicia el juego.

#### **B. Boot loader.**

Corresponde al programa de la primera sección que levanta el juego en sí, debe de dar la bienvenida.

### **5. Requerimientos técnicos.**

- ✓ Este proyecto se debe realizar en el lenguaje de programación ASM x86.
- ✓ El programa debe ser implementado en Linux, debe realizar un *makefile* para los comandos que se necesiten.

**Tarea 2.**

**6. Documentación. (máximo 5 páginas estilo IEEE trans)**

- ✓ Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- ✓ Ambiente de desarrollo: Todos los detalles de implementación y herramienta durante el desarrollo del proyecto.
- ✓ Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto (documento aparte).
- ✓ Detalles del diseño del programa desarrollo, tanto del software como del hardware: Diagramas UML, diagramas de arquitectura, diagramas de dominio, diagrama de funcionalidades, imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto.
- ✓ Instrucciones de cómo se utiliza el proyecto.
- ✓ Coevaluación (Archivo aparte).
- ✓ Tabla de actividades por cada estudiante: bitácora.
- ✓ Conclusiones
- ✓ Sugerencias y recomendaciones.
- ✓ Referencias
- ✓ El estilo debe ser IEEE- trans, y tiene un máximo de 5 páginas.

**7. Entregables**

- ✓ Código fuente con documentación interna.
- ✓ Documentación.
- ✓ Archivos necesarios para ejecutar el programa.

**8. Evaluación**

- ✓ BootLoader: 15% (Estrictamente necesario)
- ✓ JuegoBooteable: 20% (Debe ser el juego no otro programa)
- ✓ Tanques: 40%
- ✓ Makefiles: 5%
- ✓ Documentación: 20%

**9. Fecha de entrega.**

- ✓ 9 de Junio 11:55 pm por tecdigital, por ningún otro medio.

**10. Otros aspectos administrativos.**

- ✓ Para la revisión del proyecto se debe de entregar tanto la documentación como la implementación del software.
- ✓ No se reciben trabajos después de la hora indicada.
- ✓ En la revisión del proyecto pueden estar presentes el coordinador y asistente.

Tarea 2.

- ✓ Es responsabilidad del estudiante proveer los medios para poder revisar la funcionalidad del software, por ejemplo, si no se realiza la interfaz, se debe de proporcionar otro medio para la verificación, de lo contrario la nota será cero en los rubros correspondientes a la funcionalidad faltante.
- ✓ **Si el juego no es bootable la nota será 0.**

11. Ejemplo de interfaz del juego.

