

Diciembre 2017

# PROYECTO 2

## Mecanismos de memoria compartida Space Invaders

### Objetivos

- Implementar el algoritmo de Dekker para hacer uso de memoria compartida.
- Implementar semáforos del sistema operativos GNU/Linux.
- Poner en práctica los conocimientos adquiridos en el curso y laboratorio de sistemas operativos 1.

### Descripción

El proyecto consiste en la creación de un juego similar a "Space Invaders" en lenguajes C. El juego será desarrollado para dos jugadores en simultaneo (modo versus). Para ello se deben utilizar dos instancias diferentes del mismo programa. El cual se podrá jugar utilizando conexión SSH y usuario local.

El juego consiste en bando defensor y bando invasor y contara con un sistema de puntuación para definir el bando ganador.

La comunicación entre ambas instancias del programa se realizará por medio de memoria compartida, deben implementar la versión 5 del algoritmo de Dekker. También se debe hacer uso de semáforos en ambas instancias del juego.

### Características de la aplicación

- El juego se debe desarrollar en lenguajes C y debe implementar graficas basadas en texto que hagan del juego más llamativo. (se recomienda el uso de librería Ncurses).

- La aplicación debe funcionar con memoria compartida ya que no debe mostrarse diferencias visuales en ambas instancias (punteo, posición, tiempo, etc.).
- El juego debe iniciar cuando los dos jugadores estén listos.
- El juego terminará cuando uno de los dos jugadores se quede sin vidas o el score del defensor sea suficiente.

## Lógica del Juego

El juego cuenta con dos bandos: defensor e invasor.

**Defensor:** es el encargado de derribar la línea de naves invasoras y comandante.

**Invasor:** es el comandante de las líneas de naves invasoras y es el encargado de indicar que nave debe disparar.

### Inicio

El juego inicia cuando ambos jugadores se encuentran listos.

Las líneas invasoras inician con un total de 20 naves. Estas naves se mueven de forma síncrona (al mismo tiempo, distancia y dirección). El defensor inicia con 0 puntos de score. Ambos jugadores inician con 5 vidas.

### Punteo

Cada vez que el defensor logre darle a una nave invasora se le sumaran 10 o 15 puntos, según el tipo de nave que será definido más adelante).

Si el invasor logra disparar y darle al defensor se le restará una vida al defensor.

### Tiempo

El tiempo se debe mostrar en ambos jugadores, y debe ser continuo, no se debe reiniciar durante la ejecución del juego.

### Invasores

Las naves invasoras se estarán moviendo de forma horizontal, y en ambas instancias del juego deben moverse igual. El movimiento de los invasores es automático, es decir no es controlado por el jugador invasor.

Las naves invasoras desaparecen al momento de hacer contacto con una bala.

Existen dos tipos de invasores comunes y fuertes.

**Invasores comunes:** al derribarlos suman 10 puntos al score del defensor. Deben existir 16 naves de este tipo.

Forma: \-.-/

**Invasores fuertes:** al derribarlos suman 15 puntos al score del defensor. Estos son los únicos que pueden disparar en contra del defensor.

Deben existir 4 naves de este tipo y estarán identificadas por un número el cual será utilizado por el comandante invasor para indicar que nave será la que debe disparar.

Forma: (/1-\) ( el número de nave debe ir al centro)

Existen cuatro líneas de naves invasoras, en cada línea debe estar una nave fuerte, la distribución de las naves queda a discreción del estudiante.

### **Puntos de vida**

Cada vez que el defensor sea golpeado por una bala enemiga perderá un punto de vida.

Cada vez que el comandante invasor sea golpeado por una bala del defensor perderá un punto de vida.

Los disparos están limitados a un máximo de 10, es decir, solo puede haber diez balas del defensor en pantalla, para poder disparar otra debe esperar que al menos una de las balas salga de pantalla, al igual con las naves invasoras, solo puede haber diez balas en el campo a la vez.

### **Fin de juego**

El juego termina cuando algunos de los dos jugadores tengan 0 vidas.

O cuando el jugador defensor logre un score de 100 puntos.

## Mockups

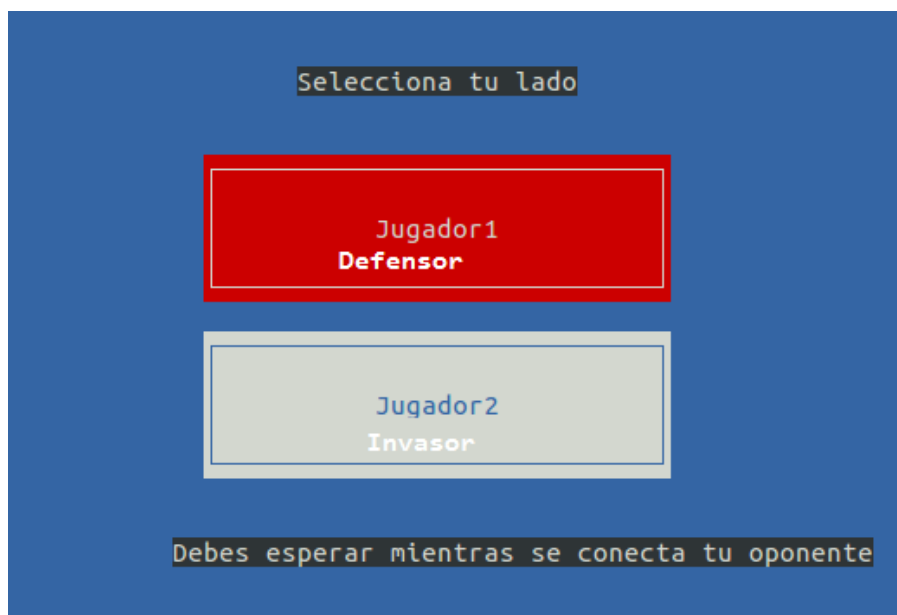
### Pantalla de bienvenida

Se utiliza para mostrar un mensaje inicial, en esta ventana debe aparecer su nombre y carne. Para avanzar a la siguiente pantalla se debe presionar enter.



### Pantalla de selección de bando

En esta sección el usuario debe seleccionar un bando presionando la tecla 1 para ser defensor o 2 para ser invasor.



## Pantalla de espera de oponente

Se debe mostrar una ventana de espera indicando el rol hasta que el otro jugador seleccione el bando opuesto, **esta sección debe ser implementada utilizando semáforos.**

Seleccionaste: 2
Defensor
Espera mientras se conecta invasor....

## Pantalla de juego

Debe mostrar la información compartida por medio de Dekker, la información compartida mínima es:

1. Tiempo del juego (es el mismo para ambos jugadores)
2. Posición del invasor y defensor
3. Posición de las balas del campo
4. Posición de los invasores
5. Punteo
6. Vidas
7. Cualquier dato que considere necesario.

Untitled Window	
<pre>* * *           &lt;-----&gt;        \.-/ \.-/ (/--\ \.-/ \.-/ \.-/ \.-/ (/--\ \.-/ \.-/       \.-/ \.-/ (/--\ \.-/ \.-/ \.-/ \.-/ (/--\ \.-/ \.-/        *       *       *       *       *           &lt;-----&gt;</pre>	<div>Vida:     defensor</div> <div>Tiempo 00:43</div> <div>invasor Puntos: 0 Vida:     </div>

## Pantalla de fin de juego

Debe mostrarse en ambas instancias de la aplicación un resumen del juego con la siguiente información:

Mensaje de fin de juego

Ganador, en caso de ganar el defensor mostrar el score  
tiempo

Al presionar Enter el jugador es dirigido nuevamente a la ventana de bienvenida para jugar una vez más.

## RESTRICCIONES

- Debe implementarse en lenguaje de programación C.
- No se calificará de ningún IDE.
- **La implementación de la memoria compartida con algoritmo de Dekker y semáforos tendrá una ponderación del 50% de la nota obtenida. Para comprobar funcionamiento de ambos mecanismos se evaluará: para el semáforo las pantallas de espera de jugador y para Dekker el cronometro y el moviendo de las naves invasoras.**
- Es requisito implementar Dekker y semáforos por lo que la nota minita es de 50 puntos.
- Debe implementarse en una instancia EC2 con sistema operativo Ubuntu.
- Debe realizarse manual técnico y de usuario.
- **Copias totales o parciales, tendrán una nota de 0 puntos y será reportado a escuela de sistemas.**

**En el caso de no cumplir con alguna de las indicaciones antes mencionadas, NO se calificará el proyecto; por lo cual, se tendrá una nota de cero puntos.**

## **ENTREGABLES**

- Cadena de conexión ssh hacia la instancia EC2.
- Ruta en la que se encuentra el archivo ejecutable del juego.
- Link de repositorio GitHub
- Manual técnico y de usuario en formato PDF
- Llave ssh de instancia EC2 para poder acceder al momento de la calificación.

Asunto: [SO1]Proyecto2\_Carnet

Correo: [lenguajes2017@gmail.com](mailto:lenguajes2017@gmail.com)

**Fecha de entrega: domingo 7 de enero 2018 11:59 pm**  
**Calificación Lunes 8 de enero**