

Fecha: Martes 7 de Octubre de 2025

Plazo: 3 semanas

Para esta tarea tendrá que diseñar e implementar un programa de simulación sincronizado que sea una versión a menor escala del clásico videojuego Doom¹. Para su simulación usted tendrá que implementar un *grid*, en donde el jugador comenzará en un par de coordenadas (x,y), para luego moverse en una ruta predefinida hacia otro par de coordenadas (w,z). Sin embargo, la simulación no trata solo de empezar en un punto para llegar a otro (sería bien fome de ser el caso, ¿no?), así que deberá agregar monstruos, los cuales atacarán al personaje principal cuando este entra en su rango de visión. Por lo anterior, es necesario explicar las distintas variables del juego:

1. Personaje Principal:

- Vida (HP): es un número entero positivo que representa la vida del personaje, si llega a cero el personaje muere.
- Daño de ataque (ATTACK_DAMAGE): nuestro personaje tiene un daño de ataque predefinido $A > 0$, con el que ante cada ataque baja una determinada cantidad de vida a los monstruos.
- Rango de ataque (ATTACK_RANGE): el personaje tiene un rango de ataque $R > 0$ con el que puede empezar a atacar a los monstruos para darles matarile.
- Coordenadas de inicio (START): el personaje inicia en unas coordenadas de inicio (x,y).
- Ruta a la meta (PATH): el personaje no se mueve de manera aleatoria, ni en tiempo real controlado por un jugador, tiene una ruta predefinida en un archivo de configuración, dicha ruta es una secuencia de coordenadas (x,y) que el personaje sigue para llegar al punto final.

2. Monstruos

- Vida(HP): tienen una vida.
- Daño de ataque (ATTACK_DAMAGE): tienen un daño de ataque A.
- Rango de visión (VISION_RANGE): tienen un rango de visión V con el cual al divisar al héroe del juego, **alertan** a todos los otros monstruos en su rango de visión. En el momento que un monstruo divisa al héroe, primero alerta a todos los monstruos en su rango de visión, una vez hayan despertado a los monstruos a su alrededor, se acercarán al héroe con la ruta más corta en el grid (distancia de Manhattan).
- Rango de ataque (ATTACK_RANGE): los monstruos tienen un rango de ataque R, con el que pueden atacar al héroe desde una distancia R.
- Coordenadas de inicio (START) x e y.

¹ [https://en.wikipedia.org/wiki/Doom_\(1993_video_game\)](https://en.wikipedia.org/wiki/Doom_(1993_video_game))

Los monstruos en la simulación están en estado pasivo, esto cambia cuando ya se divisan al jugador en su rango de visión o cuando son alertados por otro monstruo para empezar el ataque. Para simplificar la simulación, puede asumir que cuando el héroe empieza a atacar a los monstruos deja de moverse hasta matar a todos los monstruos que tenga alrededor, las heurísticas para evadir a los monstruos podrían requerir técnicas un poco más avanzadas, así que asuma que el personaje se queda parado en su posición hasta ya sea matar a los monstruos en su rango de ataque o morir.

En la simulación habrá 1 héroe (1 thread) y M monstruos (M threads), donde cada uno se mueve y ataca de manera concurrente. Recuerde que los monstruos antes de lanzarse al ataque alertan a los monstruos en su rango de visión.

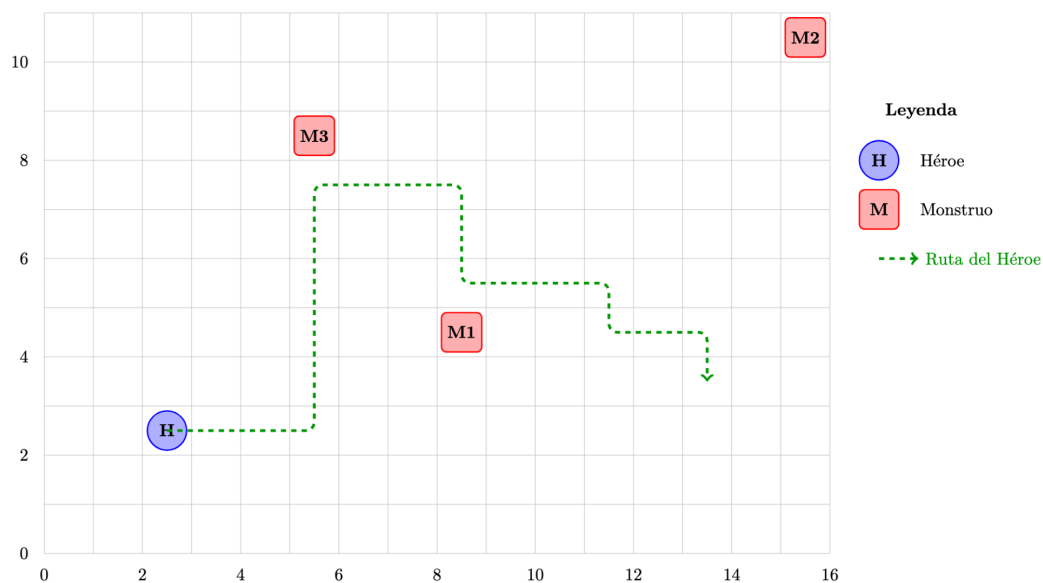
Para la simulación y posterior revisión su programa debe recibir un archivo de configuración de entrada, este archivo de configuración tiene una especificación bien definida y su programa **debe** ser capaz de parsearlo para empezar la simulación.

Una vez haya implementado la simulación cumpliendo con lo anterior **(45 Puntos)**:

1. Un héroe con las variables descritas, además de su ruta definida a la meta, además del movimiento y ataque concurrente del héroe.
2. Monstruos con las variables descritas, con la alerta, además del posterior movimiento y ataque hacia el héroe. Los monstruos también tienen coordenadas x,y en el grid. Recuerde que los monstruos se mueven hacia el héroe siguiendo la ruta más corta hacia su posición.

El paso final será extender su simulación, a una en la cual existan N héroes (N threads) en el mapa, cada uno con su ruta independiente. Estos héroes tienen las mismas variables que el héroe del punto anterior. **(15 puntos)**

Visualización del Escenario de Simulación



Importante

Algunos trabajos, seleccionados de forma aleatoria, podrán ser objeto de una interrogación oral. En tales casos, la calificación podrá ser modificada en función del desempeño en dicha instancia. Si el trabajo no es seleccionado, la nota original se mantendrá.

Condiciones de entrega

Se indican las siguientes condiciones para la entrega de la tarea:

- La tarea se desarrolla **individual o en parejas**.
 - **Debe** incluir un archivo **README** que explique cómo se debe ejecutar su código, además de una breve explicación de cómo funciona. Si incluye un Makefile que permita la compilación y ejecución todavía mejor.
 - Es **requisito indispensable** que los códigos compilen y sean desarrolladas en un ambiente UNIX. De no cumplirse esta condición, la tarea se evaluará con la nota mínima..
 - La tarea se entregará vía el Canvas del curso. La fecha y hora límite para la entrega es el día Jueves 30 de Octubre a las 23.59. Cualquier entrega atrasada no se considerará y será calificada con la nota mínima de inmediato. **No habrá extensión de plazo.**
-

Ejemplo archivo de configuración

```
GRID_SIZE 30 20
HERO_HP 150
HERO_ATTACK_DAMAGE 20
HERO_ATTACK_RANGE 3
HERO_START 2 2
HERO_PATH (3,2) (4,2) (5,2) (5,3) (5,4) (6,4)
```

```
MONSTER_COUNT 3
MONSTER_1_HP 50
MONSTER_1_ATTACK_DAMAGE 10
MONSTER_1_VISION_RANGE 5
MONSTER_1_ATTACK_RANGE 1
MONSTER_1_COORDS 8 4
```

```
MONSTER_2_HP 50
MONSTER_2_ATTACK_DAMAGE 10
MONSTER_2_VISION_RANGE 5
MONSTER_2_ATTACK_RANGE 1
MONSTER_2_COORDS 15 10
```

```
MONSTER_3_HP 80
MONSTER_3_ATTACK_DAMAGE 15
MONSTER_3_VISION_RANGE 4
MONSTER_3_ATTACK_RANGE 2
MONSTER_3_COORDS 5 8
```