Sistemas Concurrentes y Distribuidos

Prueba prácticas I y II

Instrucciones

Lee atentamente las siguientes instrucciones e indica tu solución en un archivo .cpp. Se deberá adjuntar también un archivo PDF con la captura mostrando el resultado del ejercicio compilado y ejecutado. Los ficheros contendrán tu nombre y apellidos con el siguiente formato:

`<apellidos>_<nombre>.[cpp|pdf]`

Ejemplo:

`BacaRuiz_Luis.cpp y BacaRuiz_Luis.pdf`

En el archivo .cpp se habrá de indicar el nombre, apellidos, DNI y grupo. Seguidamente se indicará la solución al ejercicio. El código fuente deberá estar documentado.

El PDF contendrá dos capturas sin recortar del resultado del ejercicio compilado y ejecutado. La no compilación del código supondrá la no evaluación de la prueba.

La primera captura indicará el inicio de la ejecución. La segunda el final de la ejecución. En ambas capturas debe aparecer el nombre y apellido de aquel que la está ejecutando.

Es obligatorio que el código compile para ser evaluado.

Enunciado

Se pide implementar la simulación de un juego de Super Mario: Rugby. El juego consiste en permanecer en el campo evitando lo máximo posible que el enemigo nos golpee. A continuación, se ilustra un ejemplo:

En nuestro caso contaremos con las siguientes restricciones:

1. Habrá hasta 4 jugadores, con un mínimo de 2. Número de jugadores: n.
2. El campo será una matriz de $n \; x \; n$.
3. Los jugadores se posicionarán en una casilla del campo. Y no podrá haber dos jugadores en la misma casilla.
    a. Se ha de asegurar el acceso correcto al tablero.
4. Los jugadores podrán moverse libremente por el campo en cualquier momento.
5. Cada jugador tendrá hasta 3 puntos de vida.
6. Habrá un enemigo que aleatoriamente recorrerá una de las líneas (horizontal o vertical). Si encuentra algún jugador, lo golpeará restándole 1 punto de vida y quitándolo del tablero.
    a. Se ha de asegurar que cuando el enemigo está haciendo el barrido, ningún jugador se puede mover.
7. Un jugador golpeado no se podrá mover ni volver a ponerse en el tablero hasta que todos los jugadores hayan sido golpeados.
    a. Cuando esto sucede, todos los jugadores vuelven al tablero. En caso contrario habrán de esperarse a que el resto de jugadores sean golpeados.

Se pide implementar dicho juego haciendo uso de los mecanismos de sincronización vistos hasta ahora en las prácticas 1 y 2 (variables atómicas, cerrojos, semáforos o monitores). Cada alumno podrá seleccionar las que considere más oportunas exceptuando el mecanismo de señalar y espera urgente.

Puntuación:

- Justificación de la solución. (1 pt)
- Documentación del código. (1 pt)
- Utilización de los mecanismos de sincronización. (2 pts).
- Implementación de los jugadores. (3 pts)
- Implementación del enemigo. (3 pts)

Cabecera del fichero .cpp (`BacaRuiz_Luis.cpp`)

```
/**
 * Nombre:
 * Apellidos:
 * Grupo:
 * DNI:
 */
/*
    Breve justificación de la solución
    ----------------------------------

*/
```

Ejemplo de salida

```
  0   1   2   3
  .   .   .   .
  .   .   .   .
  .   .   .   .
J3: me moví a [3, 1]
  0   1   2   .
  .   .   .   .
  .   .   .   .
  .   3   .   .
J2: me moví a [2, 1]
  0   1   .   .
  .   .   .   .
  .   2   .   .
  .   3   .   .
(…)
Vidas: 3 3 3 3
E: hay 4 vivos, golpeados 0
E: Ataque a la columna 0
Golpeado jugador 1 en [2, 0]
  .   .   .   .
  .   .   .   0
  .   2   .   .
  .   3   .   .
 Vidas: 3 2 3 3
J1: he sido golpeado.
(…)
Golpeado jugador 0 en [3, 3]
 Vidas: 0 1 2 2
J0: he sido eliminado.
(…)
J3: ¡he salido con vida!
J1: ¡he salido con vida!
J2: ¡he salido con vida!
```