

Programación Concurrente

Trabajo Práctico 1: Concur Battles

El objetivo de este trabajo es realizar una simulación de un juego de dos bandos. El tablero o mapa está compuesto por dos castillos, varias ciudades y caminos que comunican a las ciudades y los castillos. Al inicio del juego cada bando cuenta con uno de los castillos y una unidad en él. Cada vez que un bando conquista una nueva ciudad genera una nueva unidad posesionada en su castillo. El objetivo del juego es capturar el castillo enemigo. Si esto sucede el juego termina inmediatamente. Las unidades pueden moverse a través de los caminos y capturar una ciudad simplemente llegando a ella. Las unidades deben poder moverse simultáneamente (e independientes las unas de las otras). Cuando dos unidades llegan a la misma ciudad o se encuentran atravesando el mismo camino se produce una batalla.

Movimiento: Una unidad en una ciudad puede elegir moverse a través de cualquier camino que llegue a esa ciudad. El movimiento no es necesariamente instantáneo, pero la simulación debe funcionar correctamente independientemente del tiempo que tome esta tarea.

Batallas: Las batallas ocurren entre dos unidades de bandos opuestos. En este trabajo consideraremos un modelo simplificado de batallas que sólo tiene en cuenta el nivel de una unidad. Las unidades comienzan en nivel 1 y ganan un nivel cada vez que ganan un número de combates igual al número de la sucesión de Fibonacci. Por ejemplo

$[combates\ ganados \rightarrow nivel] : [1 \rightarrow 1], [2 \rightarrow 2], [3 \rightarrow 3], [5 \rightarrow 4], [8 \rightarrow 5], \dots$

En el combate influyen además factores aleatorios que hacen que la probabilidad que el atacante gane sea igual a:

$$nivelDelAtacante / (nivelDelAtacante + nivelDelDefensor)$$

Cuando una unidad es derrotada es destruida. Si la unidad tenían un nivel superior a 1 su bando puede generar una nueva unidad en su castillo.