

El Dilema del Prisionero (DP).

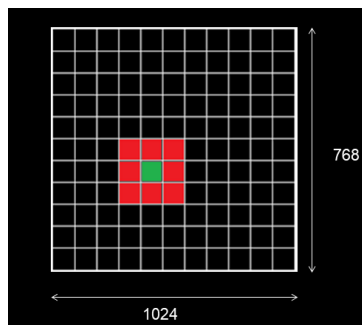
La clase pasada dimos un breve enunciado de este problema fundamental en teoría de los juegos que se podía esquematizar con esta matriz:

Prisoner A Prisoner A \ Prisoner B	Prisoner B stays silent (cooperates)	Prisoner B betrays (defects)
Prisoner A stays silent (cooperates)	Each serves 1 year	Prisoner A: 3 years Prisoner B: goes free
Prisoner A betrays (defects)	Prisoner A: goes free Prisoner B: 3 years	Each serves 2 years

Y vimos una extensión a una grilla de individuos que jugarían al DP con cada uno de sus vecinos modificando su estrategia en función de la ganancia obtenida.

Ahora nos proponemos implementar una simulación basada en GPU que cumpla con los siguientes:

1- Nuestra red esta compuesta de una grilla de 1024 x 768 individuos que juegan al DP con sus 8 vecinos mas cercanos.



2- La estrategia de cada jugador puede ser C="callar" (Prisoner stay silent) o T = "traicionar" (Prisoner betrays)



3- La Matriz de Pago esta dada por las variables $T > R > C > 0$,

		EL OTRO	
		CALLA	TRAICIONA
YO			
CALLO		R	0
TRAICIONO		T	C

Donde

const float T = 2.7;

// Tentacion x traicionar

const float R = 1.2;

// Recompensa por callar ambos

const float C = 0.1;

// castigo por traición mutal

4- En la simulación todos los prisioneros juegan con sus vecinos siguiendo su estrategia y calculan sus pagos sumando el resultado del DP con cada uno de los mismos. Una vez que todos han obtenido su pago, cada prisionero revisa su estrategia y si encuentra que alguno de sus vecinos tuvo mejor pago, copia la estrategia de dicho vecino, con la cual este logro dicho pago. Luego que todos los prisioneros actualizaron su estrategia el ciclo se reanuda y vuelven a jugar, actualizar su estrategia, etc etc y así indefinidamente.

5- La simulación arranca con un solo Traidor en el centro de la grilla. El objetivo del ejercicio es lograr que emerge esta figura fractal que tiende luego a expandirse para ocupar toda la grilla disponible:

