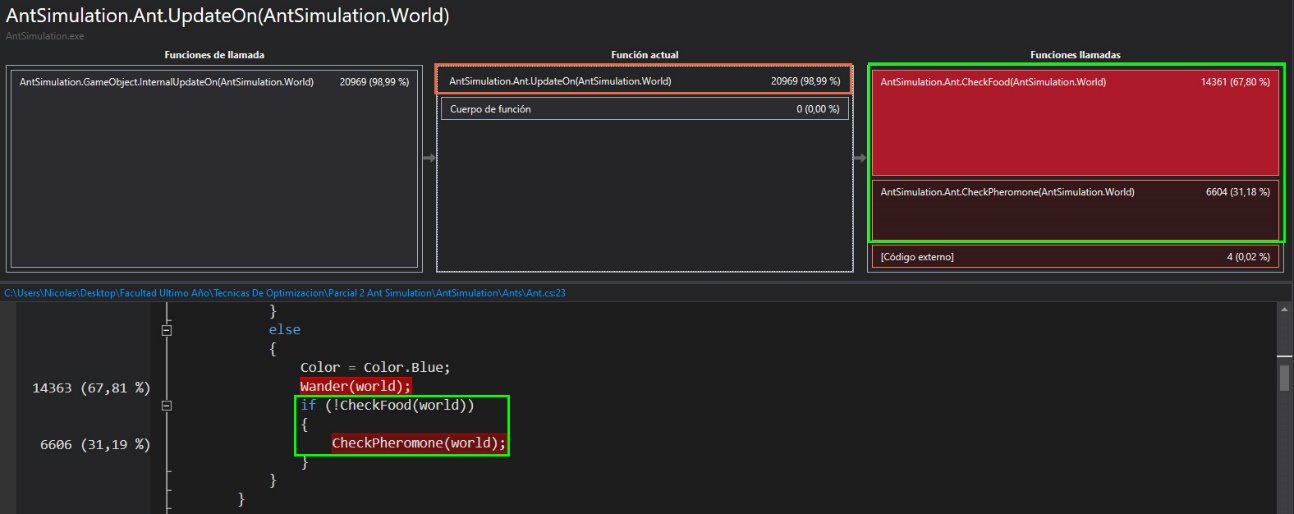
**Parcial 2 Documentación**

Nicolas Touceda

HIPOTESIS1

Mediciones:



Hipotesis:

Como se puede ver en las mediciones, en el cuadro naranja, el método UpdateOn() de las Hormigas llama a 2 métodos que están retrasando el programa, a CheckFood() y a CheckPheromone(), estos 2 métodos, a su vez, llaman al método FindNear().

Este método calcula el rango que tiene la Hormiga y luego llama a GameObjectsNear() donde se recorre la lista de todos los objetos para encontrar el Food que puede comer.

El problema de esto es que ocurre con cada hormiga, entonces se esta recorriendo la lista de objetos constantemente buscando el Food, esto ocurre en los 2 métodos.

Solución Propuesta:

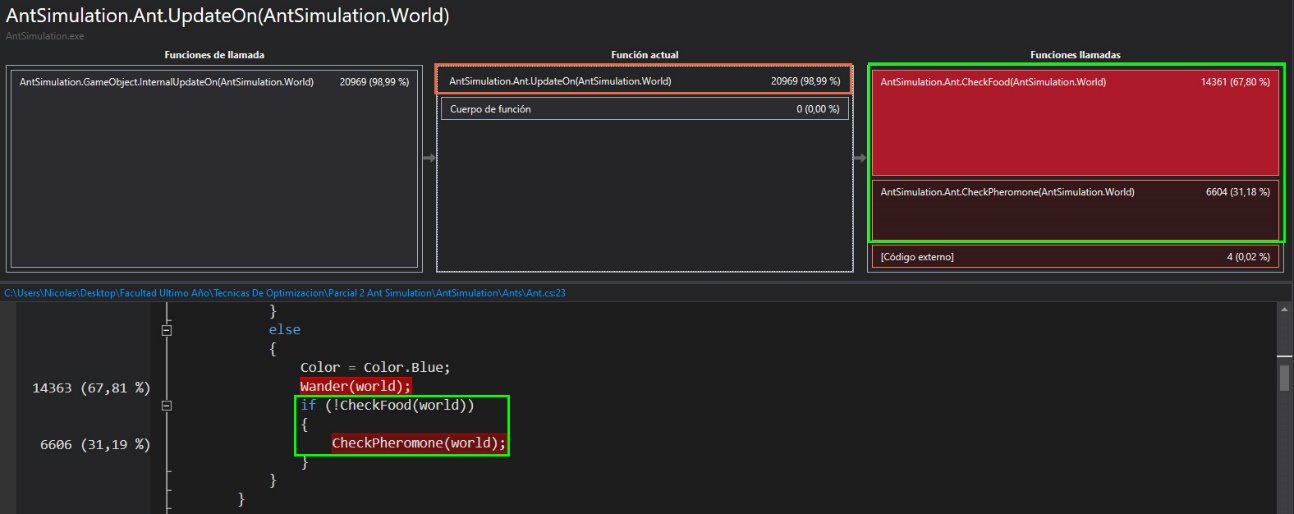
La solución que propongo es almacenar la Food en una matriz según su posición para que las hormigas puedan acceder de manera directa a la misma.

Resolución:

Realice un revert del código, porque no logre implementar la solución de forma correcta, logre que el programa arrancara más rápido, pero al segundo tira un error de tipo “System.IndexOutOfRangeException.”

HIPOTESIS2

Mediciones:



Hipotesis:

Utilizando las mismas mediciones de la hipótesis 1 encuentro un problema en el método GameObjectsNear() donde se recorre la lista de todos los objetos, para encontrar todos los Food que la hormiga pueda comer.

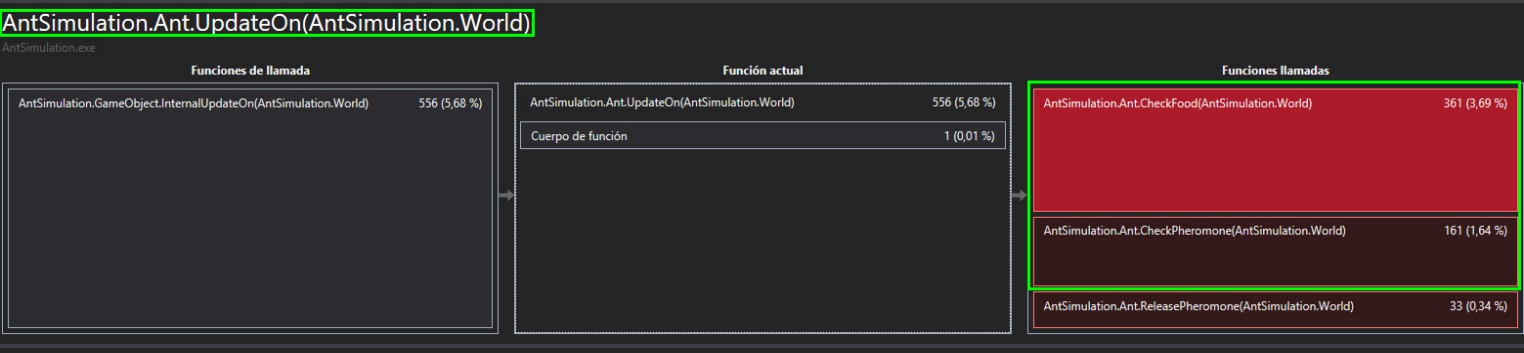
Cada ves que la hormiga calcula un nuevo Point (PosX, PosY), ejecuta el método GameObjectsNear().

Solución Propuesta:

La solución que propongo es almacenar la Food en una matriz según su posición en el World.

A partir de ahora cada ves que se calcule el nuevo Point(PosX, PosY) de la hormiga (según su rango), voy a entrar el método GameObjectsNear() pasándole la PosX e PosY para acceder a la Food directamente.

Resolución:

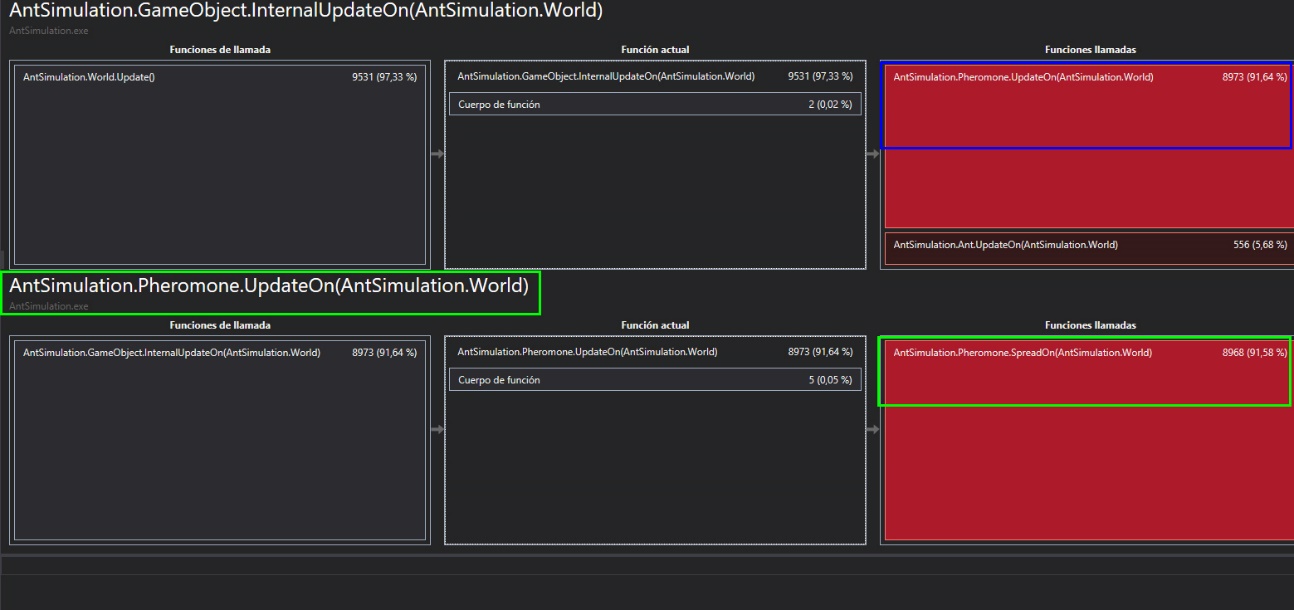


Como se puede ver en los cuadros verde de las imágenes, los métodos CheckFood() y CheckPheromone() pasaron a consumir un 67%, 30% de Total CPU a un 6,69%, 1,64%.

Por otro lado, el programa inicia mucho mas rápido, los cambios fueron Positivos.

HIPOTESIS 3

Mediciones:



Hipotesis:

En el cuadro azul de la medición, podemos ver que el método UpdateOn() de las Pheromone están consumiendo el 91,64% de CPU.

En los cuadros verdes vemos que el UpdateOn() llama al método SpreadOn() donde está ocurriendo este problema.

Solución Propuesta:

Resolución:

HIPOTESIS 1

Mediciones:

Hipotesis:

Solución Propuesta:

Resolución: