JProfiler

Los resultados de JProfiler desde la penúltima iteración mejoraron significativamente. El caso más emblemático es el CPU Load, el cual disminuyó desde ~35% a ~2%. La alta carga impuesta a la CPU se debía a que cada vez que una entidad se movía, se solicitaba que el display se actualice. Considerando que son cientos de entidades, y que cada una se mueve constantemente, esto significaba una gran carga. Para mejorar esto, se decidió crear un nuevo thread que se preocupa de actualizar el display cada cierto tiempo (predefinido, como por ejemplo cada 75ms), dando resultados visuales bastante parecidos a los anteriores, y disminuyendo considerablemente el consumo de la CPU. En ambas imágenes que hacen referencia al CPU Load (adjuntas en la carpeta JProfiler), se nota que la tendencia a ocupar la misma carga del CPU se mantiene aún ya pasados varios minutos desde que se inició la simulación. Al inicio del programa se mantiene una baja CPU Load porque consiste en escoger parámetros, y luego hay un salto porque se posicionan las entidades (se crean todos los threads). Después de este momento se mantiene en niveles constantes.

Con respecto al consumo de memoria, la imagen adjunta también certifica que se mantiene constante (no hay memory leaks, a diferencia de las entregas anteriores).