

# Water Project

By Marcel Zama(C00260146)

	1 Thread	2 Threads	4 Threads	8 Threads
1 Run	699930	753741	669988	718457
2 Run	690784	773831	674803	691257
3 Run	758214	596033	706789	676476
4 Run	690068	732567	699580	733612
5 Run	695433	681425	613036	693173
Average	706,885.8	707,519.4	672,839.2	702,595

These iterations were taken with given variables:

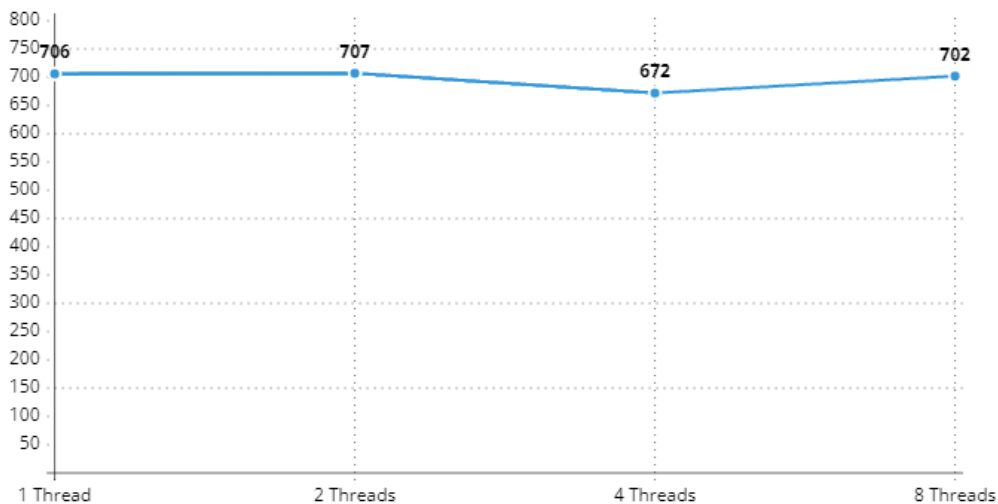
```
const int NumShark = 10000; // Initial number of sharks
const int NumFish = 15000; // Initial number of fishes
```

```
const int FishBreed = 2; // Fish breeding age
const int FishEnergyValue = 2; // Amount of energy fish gives to a shark when eaten
```

```
const int SharkBreed = 8; // Shark breeding age // number of cells shark needs to move before
it multiplies.
const int Starve = 6; // Shark starvation time // number of times the shark is accessed before
it dies // initiated to 6
```

```
const int GridSize = 1000; // Size of the grid (GridSize x GridSize)
```

## Graph



## Conclusion

In this 1000\*1000 graph, either the number of threads had no effect, or more likely my application of parallelism was done incorrectly. The execution time as observed varies, so the difference between 2 and 8 threads is not that great and could easily be interpreted as 0. I've heard that this confusion may be caused by the fact that the function I've been using, rand(), is not multithreaded friendly. This has been a very interesting and challenging Project which taught me and sharpened my programming skills for sure.