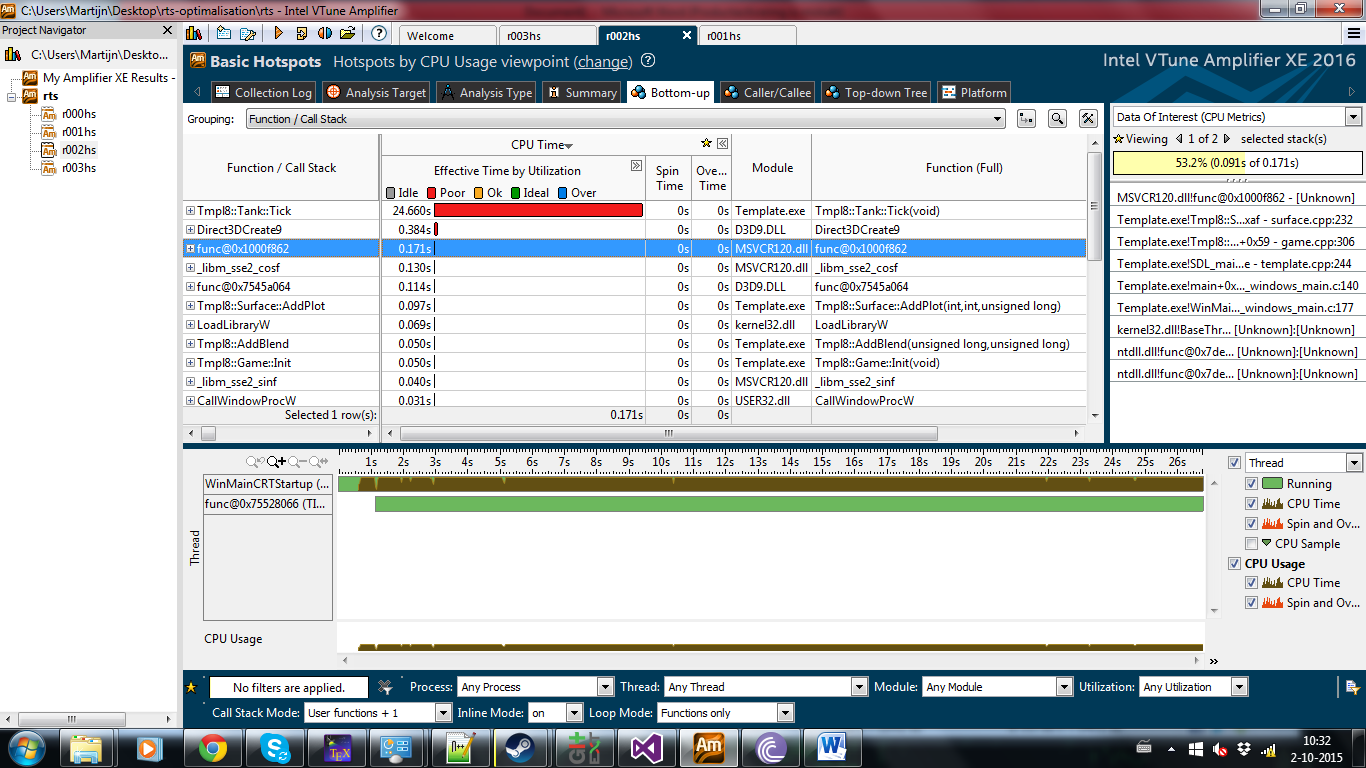
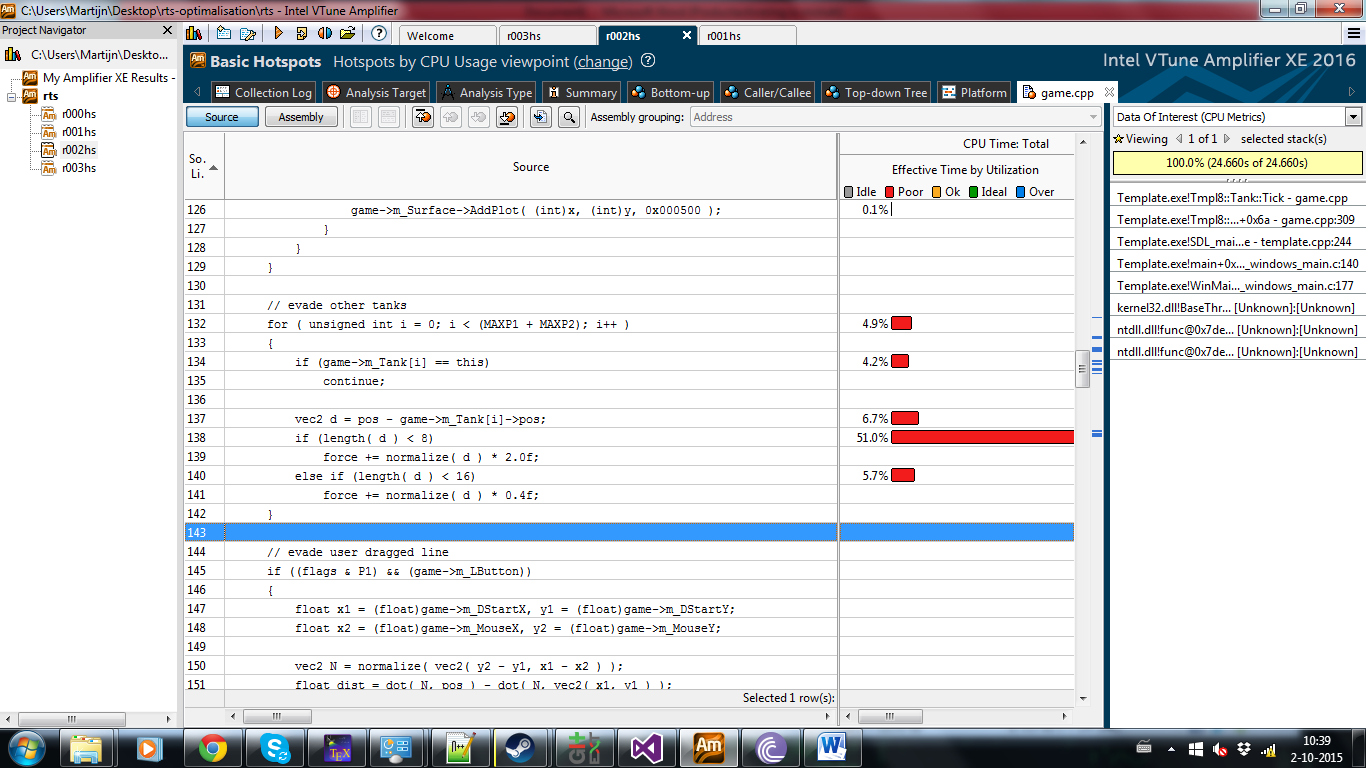
In order to obtain a good starting position for the optimisation process we increased the army size to one thousand. We also added and printed a framerate counter that at this stages showed 3 fps. We can now use both the profiler and the fps counter to see possible improvements.

After this we run the programme using the vtune profiler. This gave us these results:   
and when looking into the tank::tick function:

From this information we determined that the evade other tanks loop in the tank::tick function was the main bottleneck of the application. We then looked at the code and after some readability improvements we determined the speed of this “algorithm” to be O(n2) (the tank::tick() funtion loops over all tanks and is called #tank times per frame). A high level optimalisation for this could be to implement a grid into the program such that only one or a few cells need to be checked for other tanks to evade. This would reduce the number of tanks that need to be checked by each tank to a constant number (since there can at maximum be a constant number of tanks in the influence area). This would increase the speed of the algorithm to O(n). This could significantly increase performance and scalability of the application.