

1 Efficiency

1.1 Profiling Analysis

In order to identify locations for performance improvement of our project, we use Unity Profiler, a tool that allows developers to gather data on the performance of applications in areas such as the CPU, memory, audio and renderer.

1.2 Performance Comparison between Avoidance Algorithms

As shown in Figure 1, Unity Profiler analysis shows that the RVO algorithm uses approximately four times more CPU than the Priority and Blended algorithms combining obstacle and character avoidance movements.

RVO achieves better performance at ensuring that there will never be any collisions since it takes into account the reactive behavior of the other agents by implicitly assuming that the other agents make a similar collision-avoidance reasoning.

This is a direct cause of the RVO algorithm design, when to decide the best sample for a character, the algorithm has to compute collisions with every other character and every other obstacle in order to calculate it's penalty, therefore causing some nested loops in the code implementation thus slowing down performance.

In our implementation, Priority and Blended movements still have minor collisions due to their algorithms, while RVO achieves zero collisions.

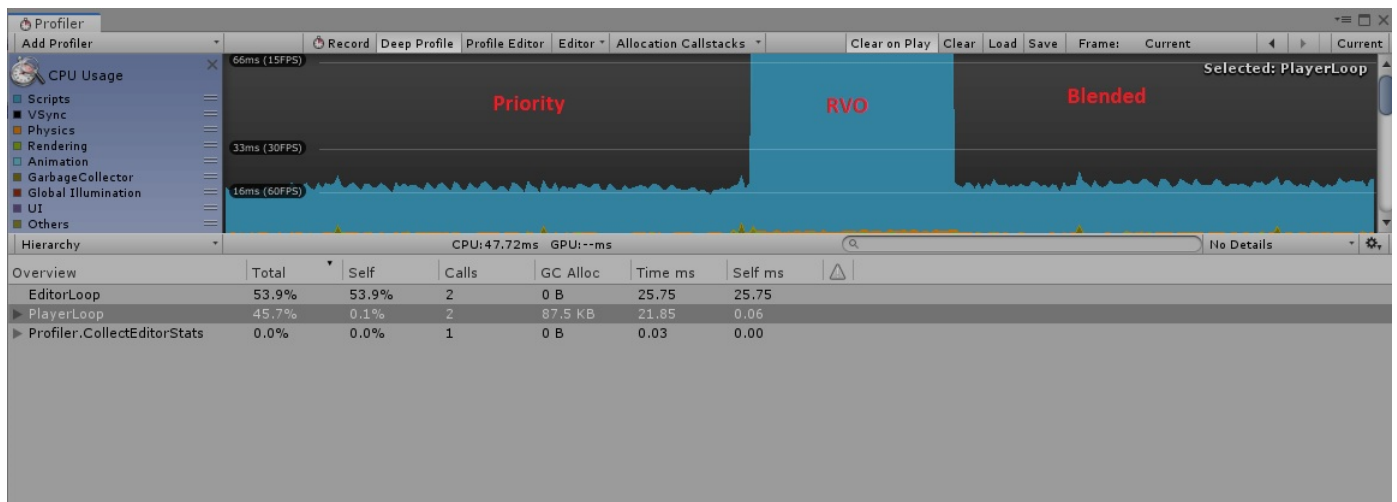


Figure 1: CPU usage for all Avoidance Algorithms with 50 characters.

1.3 RVO Algorithm Optimized

With some code optimizations in the RVO implementation, the cycles required to compute the best sample can be significantly improved.

The first optimization is to simply stop the sample analysis after the minimum penalty has been found, selecting the current sample and preventing useless iterations.

Another simple optimization can be achieved by preventing a full analysis of a sample. After the character collisions calculation, if the time penalty has already passed a threshold, the sample can be rejected, saving time analysing the collisions with the static objects.

As shown in Figure 2 after the optimizations, the RVO algorithm's CPU usage is cut to just over two times the usage of the other algorithms.

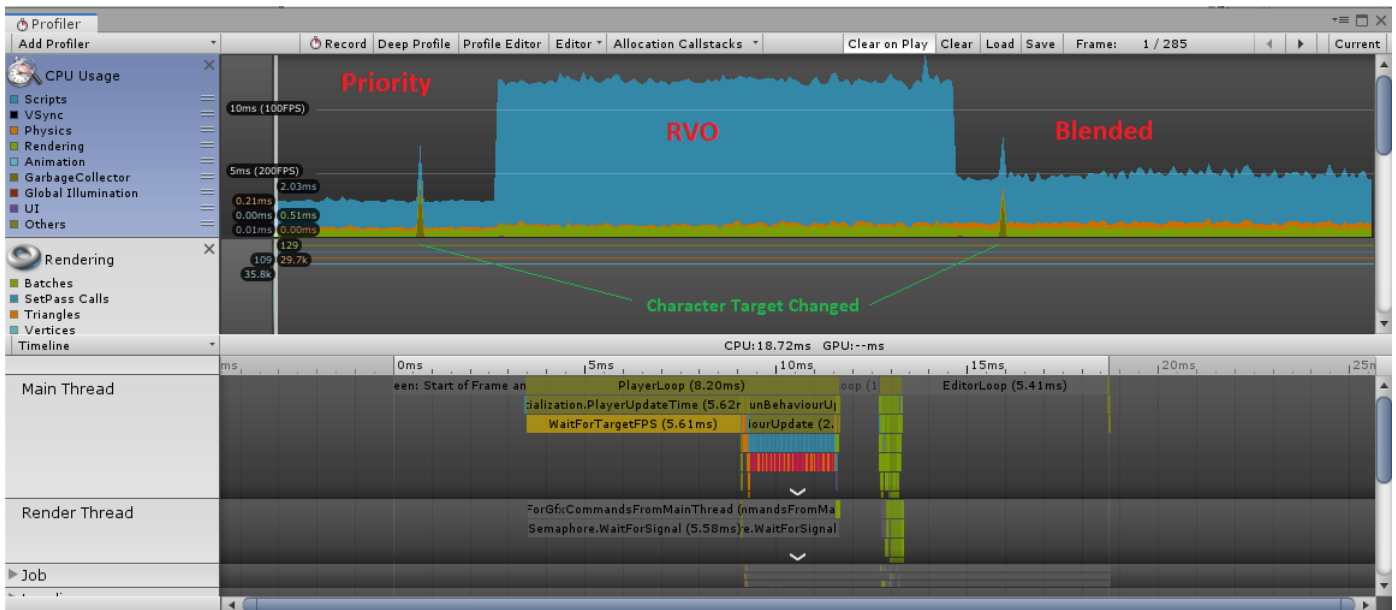


Figure 2: CPU usage for all Avoidance Algorithms after RVO optimization (with 50 characters).