Analysis Paper

1. Before starting to optimize code, firstly is to use the tools to check the functions calls, the execute times for each cpp files and methods, by doing this we figure out the draw and update method is the main target that we should works on optimizations.
2. After we do the performance profiler, we start from main file to see the process order of each files and methods that we will call outs.
3. From what we learn from classes and some martials online, we know the easiest way to increase the performance
    a. First it's the type, normally we will use float instead of double, since float will be slightly faster than double, it will estimate save some times based on the size file and numbers of double used in this function. Based on observations, we find all the types we using in the project is double, most are inside of matrix and vector files, and notice that some GL calls is called double so we also need to consider change the GL calls to float as well.
    b. Secondly, we will to avoid define some typed variable in a while or for loop, this will make compiler to redefine a variable for millions of times, by the observations, we find out that there are a lot of matrix defined in loops. So we need to get rid of it first.
    c. Thirdly is to reduce number of constructor calls, we need optimizing by RVO rules, which is the return value optimizations
4. Then will consider doing SIMD to increase our performance, it will generally increase performance at least 10 percent, by using the intel intrinsic library, we could find out millions of SSE methods to boost the performance
5. After that I will consider checking those functions which is used in project, maybe some functions are useless or not used at all, we need to remove those functions to avoid slow downing the executions process.
6. By observations we find out that its' all about matrix and vector calculations, we need think about the calculations that we will simplify the calculation methods, there are lots of ways to simplify the calculations of Matrix
7. And then I will consider the if else loop. By observe the behavior of program, we should think about which condition for if else loop has been used most, then put it in front of other conditions. And rest order by the possibility.
8. Then. In the main file, it's 5 matrix multiplications, then we might be using proxies to do MM, MMM, MMMM, MMMMM inline functions to increase the speed, it's normally faster for multiple matrix multiplications,
9. We also find out there are some STL list in the program, it's normally much slower than user defined list, so we need to do a linked list for Particle list and do the same works as STL list.
10. After that, we will to avoid too many tmp defined variable and eliminate temporary variables.

11. At last, I will consider to change the compiler setting, this is the easiest way to improve the performance by simple clicks.