Profiling Report

We compiled our program using the -pg flag that we inserted through the .pro file of the Qt project to ensure -pg is inserted during compilation. We then used gprof to use the executable and the gmon.out file to generate an analysis text file and understand where our program spends most of its time to identify where the bottlenecks are and where it is most useful to optimize.

We ran into issues with deallocating widgets as we were transitioning from window to window which caused issues when it came to exiting the program without running into any runtime errors. This caused our program to exit abnormally with a segmentation fault every time our program would exit from a transitioned widget and prevent a gmon.out file to be generated. This led us to fix the memory leak bugs to deallocate widgets properly to generate the gmon.out file and perform the analyses.

We suspect the update loop of the KillCovid part of our program to be the biggest bottleneck in our application. Our program uses a timer that repeats every 10 milliseconds or so to simulate the animations of the syringe and viruses, moving and falling in a smooth manner. This update functions houses the core logic of the game, where the graphics items’ positions are updated, boundary collisions are detected and handled, labels of scores and prompts are updated, and missing and hitting viruses is handled as well. Given that this function also requires a timer to trigger its call every 10 milliseconds, this function is called 100 times every second of the game’s runtime, which means it’s an important area for optimizing.

Other parts of the program that may be good candidates for optimization are the Othello game’s updating process. Unlike KillCovid, this process only relies on mouse events to update its UI and thus does not require a repeating timer. This makes it much less in need of optimization than the KillCovid update loop but an important are for optimization nonetheless. It is called 60 times in total due to how the game restricts the total number of moves to 60.

Other parts of the program like the login, signup and title widgets were very light on resources and barely took up any total time due to their simplistic nature.