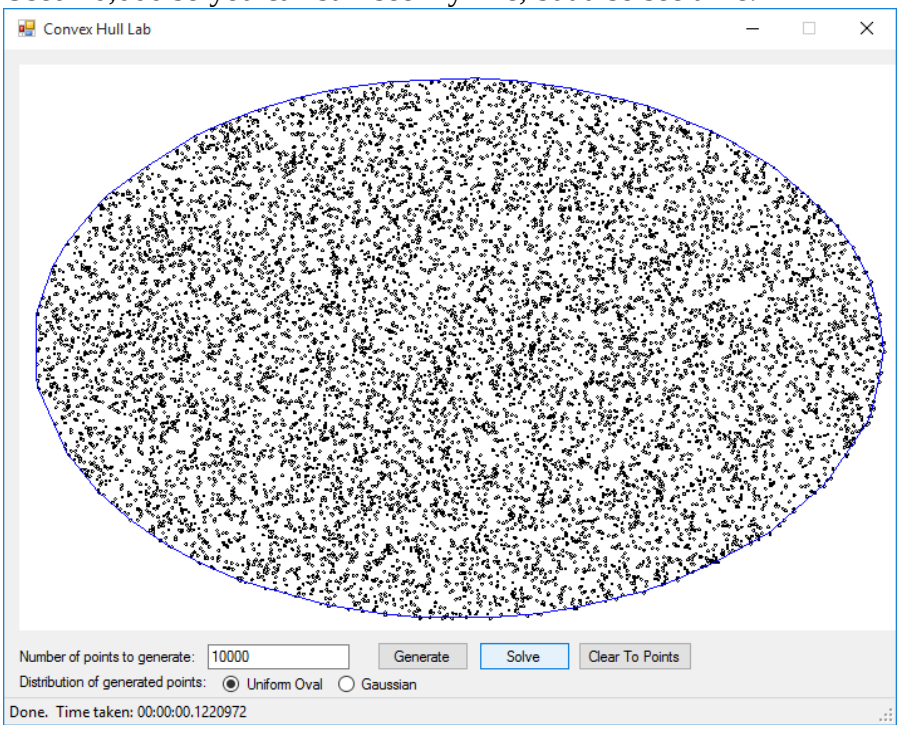
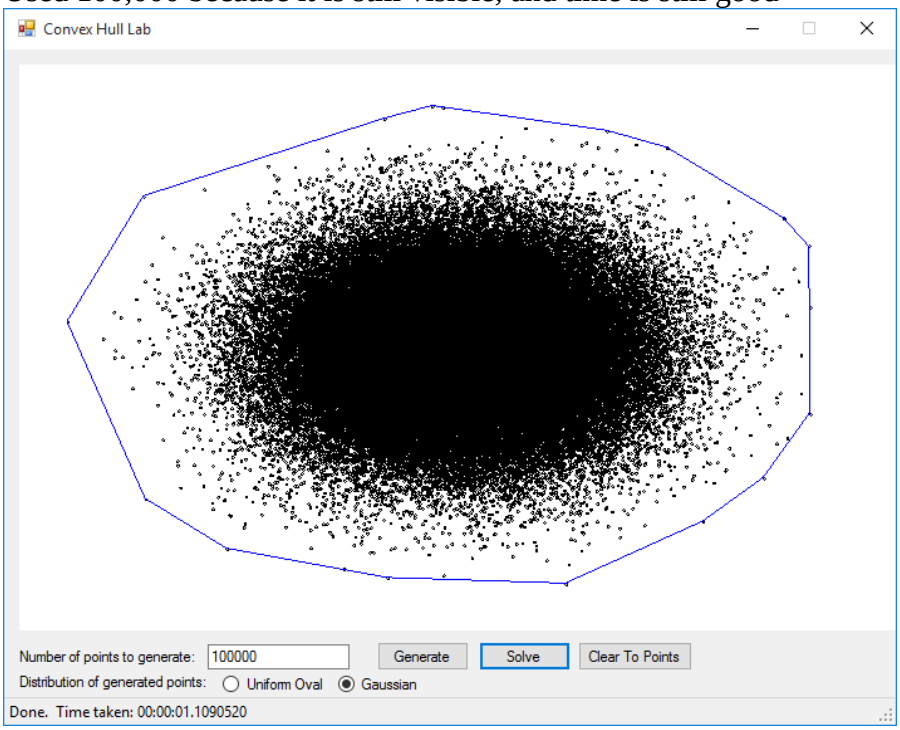
Anthony Constantino
Project 2 Write up

Used 10,000 so you can still see my line, but also see time.

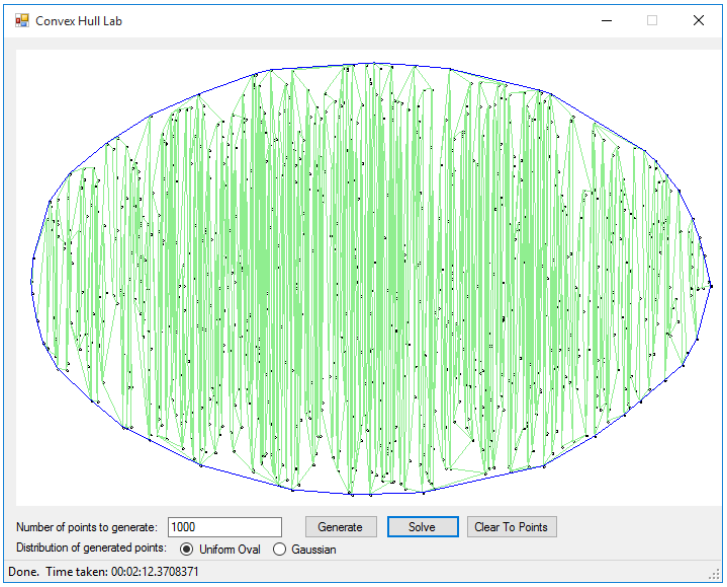


Used 100,000 because it is still visible, and time is still good



So in solving this problem, we start by implementing the framework for a divide and conquer algorithm. We organize all the points by their x-coordinates. After having the list organized, we begin dividing the list into left and right halves, using recursive calls downward, and continued doing so until the set of points was small enough (sizes of 2/3 points). By breaking the list in half each time we end up creating smaller problems of size n/2.

This brings me to the second portion of the algorithm, coming upward out of the recursion. The next step to this process is to merge these lists together. For me the most logical way to store these points in to something that can be merged together is through a Circular Doubly Linked List. This way we can simply manipulate the pointers to create a continuous list only including the outer points in the list, and ignore the points that fall inside the polygon easily.



As we do this, because there were many points being dropped off the list as we merge, our new n (consisting of the combination of points from left and right halves of the list) was decreasing and less the original n because each left and right set is now smaller than n/2. Significantly reducing the amount of time it to merge, because we are only dealing with necessary points. Allowing us to work with much smaller subset. Giving a much better performance time around O(nlogn). Allowing us to solve large numbers such as 10.000, 100.000 or even more (as seen at the beginning of this report) in relatively fast times.

From here the process will continuously repeat the merging process, until we escape the recursive function, and by this point we will have set of points only containing the points which form the outer boundaries of the polygon which encompasses all the points on the canvas, thus solving the Convex Hull Problem.