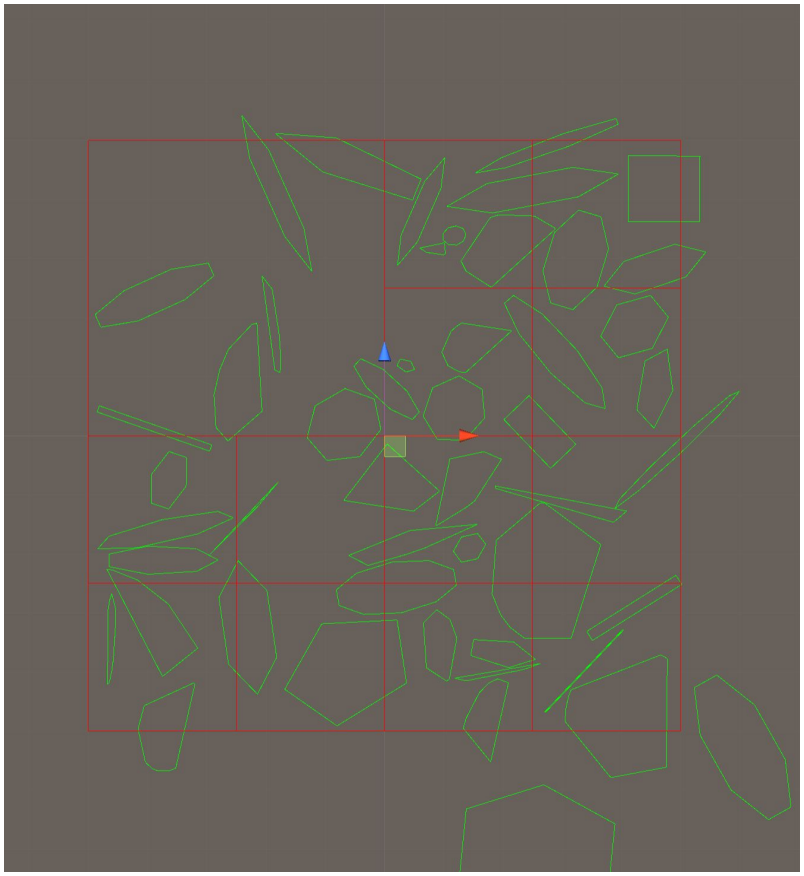
Link to github:
https://github.com/Darkat-X/Rutgers-CS423-528-Collisions

Link to video:
https://youtu.be/8z7r_36CJ5g

Write a concise paragraph explaining the efficiency of your approach. If you use a data structure, explain how to incorporate k-nearest neighbor queries into collision detection, i.e., answer (1) what point of a shape you are encoding into your data structure and (2) how you know when to stop querying neighbors of a given shape.
We use quadtree to do this part. We will separate the whole area to four parts which are represented by four nodes in the tree and input the prisms to the tree based on its positions. If the number of prisms in each node reach the maximum number of one node. Then, we will separate this node to four nodes again and do the same thing as before. If a prism is on the separating line, it will belong to the parent node of these areas. Now, when we try to check the collision, for one prism, we can select possible prisms based on the tree instead of all the prisms. The bigO should be O(nlogn)The red line shows the structure of quadtree in that situation.



The maximum number is 5 in the picture

Extra Credit:

a. We use quadtree, this data structure, to do part 4. We explained it in part 4.