Chen, Ethan
November 15, 2020
CS10, 20F, Pierson

Discussion of PS6

Some potential multi-client issues include:

- If two clients try to add or delete the same object at the same time
- If two clients attempt to move the same object at the same time
- If a client tries to move an object that another client is currently creating

The first one is an issue with regard to the synchronization problem we investigated in class on November 6th (https://www.cs.dartmouth.edu/~cs10/notes23.html). If two users try to create an item at the same time, they may accidentally grab the same id for their object since both programs would be running the same method that deals with the same static variable at the same time. In this case, one of the objects would overwrite the other in the "shapes" map, and therefore, one of the objects would not be created. Therefore, the method in which a key is attained, or in my program's case "getKey()", must be synchronized so that the keys attained are always different even if run at the same time. This is the same logic as making the process request synchronized. If two clients attempt to move the same object at the same time, then if the methods run concurrently only one will ultimately be processed. However, this does not fully solve this problem as I will explain below. The third problem is easily solved by only requesting to create the object after it is dropped. During the creation process, the drag-out should only be visible on the client's screen who is making the shape, while it only appears, and thus is valid to drag once the original client drops it.

Just making the "getKey()" and "processRequest()" methods synchronized solve the vast majority of the issues. However, if both clients try to move an object, then each of their requests will likely be sent one after another. Therefore, if two clients try to drag in opposite directions, the object will likely jitter in place but never move towards either client's cursor. If it will be first come first serve to service an entire movement, some other synchronization must be required, or the entire movement must be contained only on a single editor and processed all at once similar to drawing.