Consider possible multi-client issues. What unexpected/undesired behaviors are possible? Can you actually make any of them happen? Where and why do you use synchronized methods; will they handle everything? Write a short (approximately a paragraph) response.

Possible multi-client issues:

- Two users moving the same shape at the same time -> This could result in the local editors having different shape positions than the central sketch.
- Two users deleting the same shape at the same time -> This might result in a null pointer exception when one editor tries to delete a shape that's already been deleted.
- Two users recoloring a shape with different colors at the same time -> This might result in the wrong color being stored in either the local or central sketch.
- Two users creating a shape at the same time -> Could result in two shapes with the same ID.

All of these issues are avoidable given adequate precautions. By forcing the delete, add, move, and recolor functions to remain atomic across the communicator classes, we can prevent two threads of the client from interfering with each other. Instead, the processes will run one after the other. Adding conditional statements to avoid null pointer exceptions and using Java's UUID class to ensure consistently unique ids can also help avoid these issues.