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.

A problem could occur two separate users are operating on the same shape at the same time before it is updated on their end.

We used synchronized methods in my sketch class because they need to be called “at the same time” in all editors. This allows all the editors to have the same version of the master sketch maintained by the server and communicators. The synchronization allows the shapes to be fully drawn first before switching to the next client/editor.

Synchronization can’t handle everything, though. For example, if someone removes an object and in the middle of that happening, someone else tries to move the same object, this would be a problem because the object would no longer exist when the first person’s command finishes executing. The code still needs to check for certain exceptions to make sure that such problems don’t occur.

We couldn’t make any of these errors occur. Because one of us is not on campus, we could only do local-based testing and only had one mouse. We tried using each other’s IP addresses, but one of our firewalls would not let us do so Therefore, it was impossible for us to send simultaneous requests on the same object.

Also, we don’t have to worry about the starvation problem that we discussed in class. Since each editor is working and sending requests independent of the other editors, this problem would not occur.