**1. What can you test with Selenium?**

With Selenium you can test web applications. This is done by essentially automating the use of buttons, input fields, menus, etc. This can be automated to the extent desired by the tester, with which they believe they have covered all features relevant within the web app. These tests, however, can only test basic input/output of the web page being used; no backend processes can be tested; therefore, it always has a somewhat Blackbox approach in its testing.

**2. What can you test with Selenium and JUnit?**

With Selenium and Junit, you can test web applications under multiple sessions, with asserted expectations and conditions. This allows more extensive tests to be made, especially tests with expected outcomes, and tests requiring more than one user.

**3. What can you not test with Selenium by itself?**

With Selenium by itself, you cannot test anything that isn’t a web application. It also does not have any built-in reporting capabilities, so in order to report any tests, you need some other plugin. It can only test the system at the frontend, so anything hidden behind all the web elements cannot be tested. And although I have not experienced this myself (yet), I have heard that testing web apps ran within Internet Explorer is a nightmare.

**4. What can you not test with Selenium combined with JUnit?**

With Selenium combined with Junit, you still cannot test anything that isn’t a web application.

**5. How you would go about using Selenium and other testing tools to test your game’s interface AND your game’s logic in a systematic way?**

To go about testing the games interface and logic, I would essentially have tests following the game’s logic, with backend testing tools looking over the server processes, and once the server logic reaches a point where it sends a command to one or more clients, Selenium (with the help of some reporting plugin) would then cover testing of the client side interface, after which the server would then receive a response and the backend testing would continue. I would continuously utilize Selenium with other testing tools like this, back and forth in a scenario which tests a full run through of a web-based game. They would only be separated in test scenarios that apply to smaller aspects of the game (tests that apply to only one piece of the logic or interface), in which only Selenium or other testing tools would be relevant to the testing required.

**6. (Actually part 4 of the assignment pdf) What you would have liked to test in this game that you could not using Selenium?**

I would like to have tested the game in a more robust way. Primarily testing that interface was loaded and updated correctly, if the game was displaying things that should never be displayed, if all the web elements responded correctly, or even responded at all, the communication between a client and the server, and obviously anything at all that has to do with the games core logic (backend/server side).