Assignment 1

Before going into the Usability Test my knowledge of what to expect was already quite high. Previously in the year I had already undertaken the same test with the same tester for the same application. The first time I did the Test I had no idea what to expect or what I was going to be doing. It felt very much like going into a room and being lead by the tester the whole time. This time however I had already done the routine so it wasn’t as important for the tester to lead me through what was going to happen as I already knew what to do.

She went through the same process again, making it formal which worked well. After explaining the tasks I was to perform it moved back to a very relaxed environment where she would let me explore what to do as opposed to lead me in doing tasks.  
From this it shows that doing a user test more than once makes the user require the tester to help them a lot less. It also shows that users who have no experience with the test going into it need to rely on the tester fully. For most user testing this would be great however, for some usability tests it would be disadvantageous as users would normally have to figure things out for themselves without being lead through the various features of the application.

During the study I was given breaks to write up about my experiences, thoughts and how I felt during the testing. Unlike the first time, this time I was able to reflect on what I had done. For this study there didn’t seem like much of a point in reflection as there wasn’t much to think about. It also stopped my mind from focussing on the task at hand. I had to switch between doing the User Test Study and doing my own assignment. The first time I did this study was much better as I was only focussing on one task.  
There are both pros and cons to this. There was time for me to reflect on what I had learnt so far and think about what I could do to improve my performances for the next exercise. The downside of this was that it broke my train of thought. It is much easier to work through an application without breaks, especially when it is a small program. Having the breaks and the change of thoughts made it harder to focus as I was splitting my knowledge into working with the application and reflecting, and working on my assignment and taking notes.  
The other note of interest between the two times partaking in the user study was that in the first one it was all about the application the whole time. There was nothing else going through my mind. No thoughts on how the tester was guiding me through or the pros and cons to what she was doing. In the second however there was much more going on. I had to focus on the tasks that she had set. I also had to note how she explained exercises, her presence, how she was conducting the testing. Other thoughts that were going through my mind were how easy the exercises are now that I was already exposed to them and how this would affect my views within this study. The end result was that there was much more going through my mind in the second one that user testing of the application was not a true representation of my abilities.

The hardware being used wasn’t the best that it could have been. The issue was that it wasn’t always reactive to my gestures. Occasionally it would respond but I would generally have to touch the screen multiple times before something would happen. This irritated me throughout the exercises making some tasks take longer than others. For one task I had to turn a page and it wouldn’t respond. I sat there continuously trying to get the right touch so that it would respond. In the end the tester actually had to point me to the solution that it was set to the wrong function. It was my own fault that it hadn’t worked properly. However due to the class of the hardware and because of its past issues, I had assumed that it was the reason it wasn’t working as opposed to being user error. From this it was a bad device to use as it irritated me when trying to complete tasks. When I go to do user testing it would be a good idea to make sure the actual hardware that the testers will be using works correctly and won’t confuse or irritate them while they are trying to perform tasks on the software. The other reason the hardware should work without irritating the user is that users may explore various parts of the program so as to get a good understanding of where everything is and what everything does. A trial and error approach is generally used by everyone and if they get irritated by the hardware because it’s not responsive or too slow, they will limit themselves in using the trial and error approach. As a result it may not give a true indication of how they would use the application, which skewers the results for the tester.

One thing that was good the first time was the level of formality. By keeping it formal and work related, the exercises got done and there was a clear distinction in who was in charge and who was to follow. However during the second one there was a lot of informality. Having a formal session broken up with informality caused a slow and somewhat scattered flow of events throughout the testing. It was understandable why it occurred in the second test however, due to the repeated process that both tester and I had to go through. When testing for usability it seems both more professional as well as more controlled if everything it kept at a formal level.

Lastly there were issues in the tasks laid out by the tester. During both times sitting the usability testing one exercise was to find a word in the book with the focus being on using the search bar to find it. However while doing exercises before this task, both times the pages were open to a page showing the word in question “snail”. The book being used was a book on snails so the probability of seeing the word without using the searcher was quite high. If the user wanted to see the word finder to be used they should have chosen a more vague word that occurs less in the book or even put the search task first so that the user couldn’t cheat by pointing to the word after skimming a page to find the answer.

The first time in doing the user study there was a exercise that I spent a good 10 minutes trying to solve only to have the tester figure out that the feature she was trying to test wasn’t even implemented in the application I was working on. This not only upsets the users who are being blindly lead by the tester, assuming the tester knows everything there is to know about their application, but also creates a form of distrust between user and tester. This trusting issue is vital in testing for if testers ask them to do something else; the user begins to doubt whether it’s possible and whether they will be wasting their time trying to achieve something else that doesn’t exist. It was great to see that during the second testing session that exercise no longer existed, it meant you could fully trust what the tester was doing. To conclude from these findings it would be a great idea to have tested the program yourself first and maybe even have a practise run through with someone before opening it up to user testing to make sure you can focus more on the usability rather than all the errors there are still left to fix.