

Chemistry Learning In Progress

Testing Recap and Results

Table of Contents

| | |
|------------------------------|-----------|
| 1. Testing Recap | 1 |
| 2. Online Testing | 5 |
| 2.1. Tester #1 | 5 |
| 2.2. Tester #2 | 7 |
| 2.3. Tester #3 | 13 |
| 2.4. Tester #4 | 18 |
| 3. Acceptance Testing | 20 |
| 3.1. Assignment #1 Results | 20 |
| 3.2. Assignment #2 Results | 37 |
| 3.3. Assignment #3 Results | 54 |

Testing Recap

Our team began the spring semester with several stages of testing in mind. These stages were divided into two categories: team-oriented and user-oriented. The team-oriented stages were run in a cyclical manner consisting of module testing, integration testing, and system testing. On the other hand, the user-oriented testing was done concurrently online and in the human-computer interaction lab.

Our module testing consisted of each team member working and improving the latest approved version of the system. Each member would be assigned to work on a particular feature, where each feature was mapped to a particular module. However, there was often overlap between communication interfaces (namely the User Interface Module and User Interaction) requiring that individuals working on those modules to appropriately comment their work in that area to make integration much easier. Therefore, each member handled their own testing of the feature (s) they were working on, making sure to appropriately comment their work. It should be noted that module testing done in this manner worked quite effectively despite the lack of a CVS Repository, but required much more caution during the latter part of the project after features were implemented and only required improvements.

The integration testing worked out fairly well because even though we were working concurrently on our respective modules, our work was divided up to allow related modules to end up at similar times so that integration would happen in smaller chunks. Obviously, this would have been easier with a CVS Repository but since the project was so small, we were able to manually manage the code integration.

After each major integration, we proceeded with system testing. This involved continually testing the latest stable version of the application until the next integration. Therefore, we produced as a result, a list of ongoing bug fixes that would be handled as we went along. The result of system testing was to produce what we called a milestone or stable version that was free of all known major and minor bugs.

Since we were working on an evolutionary prototyping lifecycle, the aforementioned stages were iterated over, starting at first with long gaps between testing and eventually shortening the gaps as we approached and went through online and user testing, so to make sure we were using the latest stable version.

After reaching a version that was both stable and contained improved functionality, we proceeded with online testing. This consisted of having users visit our usability-testing page of the CLIP website. Once there, they would be briefed on the purpose and goals of the testing and then would proceed with completing any of the four tasks. The four tasks consisted of familiarizing the user with the application, creating a Tile Arrangement, using Playback, and creating and modifying a Tile Set. Our intention was to get feedback from faculty at their own leisure, especially those using a Mac OS. Testing did not go as planned since the number of users was low. However, despite the low turnout, the users who tried the application were using Mac OS, which was a big deal for us since for most of this project we were unable to get a hold of a Mac. Our users told us that it worked and looked well on the Mac, which was definitely encouraging. Moreover, the preparation work done for the online testing laid the foundation for the user testing, which I will discuss next.

We did the user testing in the Human-Computer Interaction Lab with several students from our client's classes. These students had varying academic backgrounds and most had not used the current paper system. This created some initial difficulties with testing because we had to revise the versions of the tasks from the online testing to be careful not to confuse them with terminology that they would be unaware of. We complimented this with a brief introduction of the current system for every user. We ran each test session for 45 minutes, having the users start with the Tile Arrangement task, then do the playback task, and end with creating a Tile Set.

We found that throughout the testing, doing the Tile Arrangement was the easiest task for users to do. The biggest issue we had was making the user aware of

blank tiles and the ability to manipulate the use of the Tile Bin. We approached both issues by adding a context menu that is accessible on the Grid and the Tile Bin, since many users were clicking in both areas hoping to be able to do something. In task 2, most users had very little trouble completing the task and understanding both how to use the controls and the information being displayed. Obviously, the look and feel of interface for the controls and how they worked evolved as users made suggestions. The last task – Creating Tile Sets – was by far the most confusing and difficult task to complete. This worried us. Many of the users were often trying to save the Tile Set before adding tiles because they had no idea how to do that. In addition, no single user (including our client) noticed the scrollbar on the Rules tab because of the default look and feel in Java made the bars difficult to notice. The last major issue we ran into was making the user aware of how to set a tile as the Blank Tile. This is why paying close attention and detailing the user comments was very important.

Foremost, we dealt with the first issue by re-labeling the Tile Editor tab to the Tiles tab. Furthermore, if a user tries to save without adding any tiles, we switch automatically to the tiles tab and pop up a message indicating what they need to do. Next, the issue with the scrollbar was resolved by moving the information on the Rules tab closer together, such that part of the hidden information is visible so that they know to look for the scrollbar. Finally, we resolved the blank tiles issue by having multiple ways to set blank tiles. We have a separate button called Set as Blank, a context menu with the option that appears in the tiles listing, and a combo box at the bottom of the Tiles tab that both displays the currently selected blank tile and allows you to select a blank tile.

In summary, our testing ran for the most part as intended, with the exception of the online testing. However, all testing that was done provided important feedback for improving the CLIP application's interface and features.

| Version # | Date | Author | Description |
|------------------|----------------|----------------------|-------------------------|
| 1.0 | 4/28/06 | Brian Navarro | Initial Creation |
| | | | |
| | | | |