Justin Schuyler

Mod 7.3 Sandbox Usability Evaluation

Purpose

The purpose of this usability evaluation was to build perspective. To provide the best user experience available, it was important to identify strengths and weaknesses in the system through thorough testing. Feedback from target users will help to demonstrate how well design intentions coalesce with actual practice. Framing that feedback within the contexts of usefulness, learnability, efficiency, and satisfaction will produce a clear strategy for improvements to the system as a whole.

Introduction

Sandbox is an integrated development environment targeted at open web technologies and designed to facilitate rapid, cooperative software development. The overarching goal is to provide developers, teachers, and students with a rich, interactive platform for learning and working through constant experimentation. This evaluation focuses on a set of low fidelity prototypes that illustrate a component of Sandbox called multiplayer coding. Multiplayer coding features a mixture of community and development tools, allowing people to connect, communicate, and code together. Code is saved, executed, and managed by version control automatically. Just like a real sandbox, users are encouraged to play. Their changes are reflected in real-time, enabling them to test new ideas and easily recover from mistakes. Sandbox couples this exploration with community based support to foster creativity and simplify the process of troubleshooting.

User Profiles

Matthew D.

Matthew is a former SISLT Masters student and currently working on a PhD in Computer Science. He is also the System Administrator for MU's Information Technology Program and handles web technologies on a daily basis.

Elliott S.

Elliott is an Electrical Engineering undergraduate at MU. His programming perspective is technically-oriented and less focused on interaction design. That makes his feedback particularly intriguing because he understands the importance of technical requirements and functionality with respect to usability. He has experience using Visual BASIC, Java, and MATLAB.

Testing

The evaluation was split between Matthew and Elliott into two procedures: direct observation and situated cognition. Matthew was observed directly in a structured setting with a clear set of directions, while Elliott evaluated Sandbox individually and at his convenience. My reasoning behind this was that there are costs and benefits to each method. The benefit of direct observation is that it provides an extraordinarily detailed look at set of tasks; however, the laboratory setting doesn't lend itself to experimentation. Even when the user is assured that they are not being tested, they may still be reluctant or cautious with their interactions. Removing the user from that setting loses the control and detail of direct observation, but it offers extra insight by allowing the user to explore and play with the interface on their own.

Debrief

I explained the purpose of this evaluation to both users and provided a short description of the project as an introduction. I described the level of detail included in the prototype and their tasks in evaluating it. I encouraged them to relax and not view the evaluation as a test of their aptitude. Task completion and incompletion are equally valuable pieces of usability information. In the direct observation, I asked Matthew to "think aloud" and express his immediate reactions, both good and bad. With the situated cognition session, I instructed Elliott to simultaneously take notes reflecting his ideas. After each session, we held a discussion to review the findings and walk through a few follow-up questions. To conclude, I thanked Matthew and Elliott for their participation.

Evaluation Summary

Good Designs

- Overall, the system is easy to use and logical.
- The layout works well and does a great job of encapsulating content and other features.
- The ability to code and easily communicate with people while seeing their progress updated in real-time is very helpful.
- The chat-drag option is convenient and promising. Expanding the chat-drag option could motivate people to try creative things with other people's work. For example, consider a person who is coding with a friend and is having trouble getting started on a certain task. That person could observe what their friend is working on and through their observations they might become motivated to approach the task from an angle they wouldn't have considered alone.
- The fading feature on the chat window is a nice aesthetic effect and it also has an immediate practical benefit. Some people have wandering minds and maybe that subtle fading of the chat text will allow extra focus on the code while still providing the comfort of knowing the chat conversation is one click away. Comfort leads to relaxation, which leads to productivity.
- The interface is user friendly, uncluttered, accessible, and intuitive.

Room for Improvement

- Version control should be more detailed and offer a way to view a history of edits.
- When are changes committed to the real-time interface? This is an internal detail that would need to be ironed out.

Recommendations

- Perhaps a tab bar could be added to the top of the interface to handle browser refreshes and the execution of code, much like the controls on a traditional web browser.
- If I were working in a development environment like this, I think I would benefit from some kind of "paint" interaction option. What I mean by that is that I think it could greatly enhance the efficiency of communication if you could draw a circle around some feature on the webpage that is being displayed and send that through the chat menu like you can with the code.