Abstract
This paper presents aspects of a study on usability of a voice warm-up application for singers. The study examined singer’s behaviour at the time of asking them to use the software. We focus on the interaction between the computer and the user with special attention to a set of usability requirements.
We conducted the evaluation with a high fidelity prototype that allowed us to explore non-typical interaction with a computer: sounds and images, like a piano or a staff, in application software.

To conclude we offer a summary of the lessons learned over the course of the iterative design process and some key ideas to improve in future projects.

Author Keywords
Human-computer interaction, HCI, sound, music, singer, warm-up

ACM Classification Keywords
H.5.5 Sound and Music Computing

Introduction
Technology has penetrated in all sectors of the society. People use these new tools in many different ways depending on the task they want to accomplish, their education, their environment or even their physical conditions, amongst others.

Musicians use plenty of specialized devices that help them do efficiently everyday tasks of their field. Some questions arise, like "What kind of special needs do they have?" or "Does technology help or bother a singer?".

We open this paper with a description of the problem we tackle. We then discuss the final design of the application and the alternatives we considered, high-level implementation issues, and the procedure and results of the evaluation. The final section of this paper presents some lessons on HCI, like the convenience of having a participatory design and a practical use of Nielsen’s heuristics [1] and design principles.

Problem
There are two main problems when it comes to warming up singers. One, most singers do not know how to properly warm up, and two, you almost always need a piano around to run the warm-ups. When you do not warm up, or do not warm up properly, singing will put a lot of pressure on your vocal cords causing damage. The problem with always needing a piano is that pianos are large. If you live in a dorm or small apartment you won’t have the space to keep one. If you’re in college there aren’t many places where practice rooms with pianos are available.

The target user for this application would be a singer, primarily aimed at a student in college, but also for use by any singer of any experience level.

We want to create an app that you can use to warm-up your voice with. An app that will act as a general piano for warming up purposes, as well as supplying additional features to the user. It will have a section for browsing pre-loaded vocal exercises and selecting them to warm up. Each exercise will have an explanation with it detailing what it does and how to properly perform it. In each exercise you will be able to play / pause it, as well as speed up and slow down the exercise. There will be a favorites feature too in which exercises that the user performs frequently can be added to a list of favorites. These exercises will appear on the home screen for quick access.
Design

Final high fidelity prototype

The application is separated into a home screen and three different sections: "Discover Voice Type", "Perform Exercise", and "Tone". The home screen contains links to the three sections as well as a list of all the exercises favored by the user. If none are favored, the favorite box displays "You don't have any favorite exercise yet."

The "Discover Voice Type" section allows the user to find their voice part by harmonizing with a piano and selecting their highest and lowest notes. The main section of the screen is a large keyboard. It is one octave and shows the octave you are in above it. There are arrows on either side of it to change the octave up or down. Below the keyboard are two buttons. On the left is the lowest note selector and on the right is the highest note selector. In between is a screen which shows you the last note played. If no note has been played then it will say "Play a Note". At the bottom of the screen, the selected range of the user is shown with a final accept button. When finished the users voice part will appear on the home screen as well as a challenge button where the user can challenge their old range to see if they have improved or not.

The "Perform Exercise" section contains screens for the list of exercises as well as the screen showing the exercise running. The first screen that you are taken to is the list of exercises. There is a clickable star next to each exercise to have it favored. A gold star means the exercise is favored and can be clicked again to reset it. To the right of the list is a box containing the description of the exercise that the user has last clicked on. Under the description is a button allowing the user to perform the selected exercise. When perform is selected a new screen is opened. On the top of the screen the title of the exercise you are performing is displayed. In the main section of the screen is a treble clef with the notated exercise written on it. Underneath is the controls to play, pause, rewind, and fast forward the exercise. At the bottom is another tool which allows the user to speed up and slow down the entire warm-up. The finish button is in the bottom right-hand corner of the screen exits the exercise and returns you to the browse screen.

The last section is the "Tone" section. This screen is mainly for the user to have an easy way to get a note, for example if they are learning a song and need the starting pitch. The screen is an octave keyboard which is identical to the one from "Discover Voice Type" to keep with internal consistency. It displays the last played note underneath the keyboard as to help the user if they are not proficient with a keyboard.

Design revisions

Many design revisions were made throughout the process of building this application. We realized as we went through revisions that many of our broad categories could be condensed into one thing. For example, learning a warm-up and performing a warm-up became part of the same portion of the application. We originally considered these different sections to separate new from experienced users. The new user would be presented with instructions and a speed slider that the experienced user may see as a nuisance. As we went thought designs we realized how we could keep the instructions out of the way enough so that it wouldn’t become an annoyance to more experienced users.

Once we felt we had a suitable app that users would like, we went to paper prototyping. We gave the prototype to several different test users and realized that maybe the application was not as intuitive as we thought. Although the prototype was only paper at this stage, many users had problems using the keyboard widget we had designed for the Discover Voice Type functions. In addition, some of the cosmetic choices we made at this stage either confused users, or caused them to make slow decisions while using the app. Specifically, the favorites system, in which exercises can be selected to be shown on the main screen for easy playback, was not as easy for users to learn as we had originally thought. This was deemed to be a problem with the affordance of the
star icons we used to represent favorites selection, as well as the visual limitations of the prototype itself. (Especially the difficulty in making an element of the paper prototype look "clickable", which is easier to show consistently with Java Swing widgets.)

Another revision we did at this time was getting rid of the results screen in the "Discover Voice Part" section of the application. We decided to get rid of this page because all the information the page was displaying was already being displayed somewhere else. The vocal range was already seen on the page the user was on before, and the voice part is always shown on the home page. All the page seemed to add was one more click to get through to get back to the home page.

We then received some heuristic evaluations from 3 classmates with 20 issues each. They pointed out that some elements of our interface did not conform with some of Nielsen’s usability heuristics. The most prominent of these problems were code-based, but some were interface design decisions that needed to be changed. These changes include adding error recovery options in the form of confirmations dialogues, ensuring the user would not exit the exercise prematurely. In addition, widget and button sizes were standardized among forms of the application in order to be consistent visually.

In our third and final testing we used a high fidelity prototype with test users. Again we realized, with the feedback from our test users, that there were still many fixes we could make to the application. We again had problems with favorites. This time with the star icon itself. Many testers thought the stars were just bullet points next to each exercise name. This was surprising but because many users had this problem we realized it was a very important fix. We made the icon larger and more apparent, making it
look more like an interaction and less of part of the background. Many users said that they thought the star was just a bullet point and not something clickable. Another piece of feedback we got was on the keyboard the user had no way of knowing what octave they were on. This caused problems with the users trying to harmonized with the keyboard only to find that the keyboard had jumped two octaves. We added a simple text display showing the octave the user was in on top of the keyboard to remedy this problem.

Implementation

Functionality

The prototype implements enough functionality to accomplish the tasks given to the users. These have the following views available:

- Main screen with favorited exercises
- Browse exercises

Technology

In order to test if the functional requirements and the look and feel was acceptable, we decided to make some prototypes. This way we could check the usefulness of the application, validate requirements and see if our metaphores suited well the needs of our users.

Two kinds of prototypes were implemented: low fidelity and high fidelity. We built in all the necessary functions to test a certain set of tasks (vertical prototype), so in every new version of the prototype we made design changes, but did not necessarily add new features. Not being committed to one implementation or idea let us apply changes rapidly and go back to the users to test it.

The first implementation was a cardboard prototype with a few detachable parts. The size was adjusted so that it could be used with fingers instead of a mouse. Because it had a clunky look, users gave us feedback about their intentions, actions and expected results instead of non-interesting aspects in this stage, like colors or typeface. A member of this team played computer and changed all detachable parts and other small pieces so as to have changes in the data displayed that are consistent with the user’s action. Be-
cause a real system would generate music for the user to sing along, the person playing computer also sang.

The high-fidelity prototype is coded in Java 6 SE and needs to be run from Netbeans on any operating system with a Java VM from Oracle. Ideally it should run in a browser as an applet but the Java browser plug-in does not allow certain actions for security reasons. The sound is generated with MIDI (javax.sound.midi.*).

The first experiments with sound generation made the UI hang. As expected, CPU intense work needs to run in a worker thread to prevent the UI thread from blocking. Otherwise the application is not responsive and this affects dramatically the user experience. In any case, sometimes the application needs to be restarted after using the sound system. Probably the sound device is not marked free so the next time the prototype requires sound, it can not gain access to it. This issue does not interrupt any of the tasks assigned to the user.

Because this is a prototype without any back-end logic, not having a data model made it harder to implement the favorites feature. Users could not test it in the first iteration. It became available in the next versions of the prototype in order to get feedback for this feature.

Custom elements on screen, like the piano and the favorites icon, are not widgets for simplicity. They are images in a label with OnClick callbacks. Their logic is encapsulated in a function and a very simple system of asynchronous messages solves the issue of making data consistent across multiple windows.

Future work and issues

Extra on-screen cues, like pointing the current note in a staff while performing an exercise, are out of the scope of the project at the time of writing this report. However, implementing this feature is trivial. We suggest having a vector of variations of the image that contains the staff or, maybe more efficiently, overlaying another Swing component that behaves like an arrow.

Icons belong to the Tango Project. Because of a lack of graphic design skills, some images might not be the best choice and affect visibility and mapping of actions. Text next to them should compensate this bias. These icons are widely used in the Gnome project. However, they are not common in Microsoft Windows or MacOS. This clearly affects integration with the platform and, therefore, external consistency.
Evaluation

Method

The target user is a singer, most likely in college. They can range in age from 13 - 50. Their ethnicity, or education should not make a difference on using this application. They do not need to have extensive knowledge of computers. They can be at any level of singing, from beginner to professional. This application is not intended to be used by people who are deaf or blind.

We did two sets of user testing for this application with 5 participants each time. The first using a paper-prototype, the second using a high-fidelity prototype. We scheduled and met with each user in a private room and had them preform the tasks given. One person welcomed and explained them the procedure and the others observed and took notes. In the low-fidelity prototype, one team member played the computer. We then conducted a semi-structured interview to get comments and general feedback. When all the user testing was complete, notes were synthesized and a list of changes to the prototype was proposed.

Users

The users who participated in the evaluation were all singers between the ages of 18 and 23. Both men and women were tested. All users were from various a-cappella groups from around campus who gratefully agreed to help out. These users had varying experience with computers and applications, which allowed us to test users with similar singing experience but different computer backgrounds. The only demographic which we would have ideally tested would be a group of older, less technologically inclined singers. Due to the difficulty involved in setting up a test session with professional singers, however, we decided that a cross-section of users from singing groups on campus represented the majority of our primary stakeholders, and geared our design for the use of these singers.

In-lab Procedure

We started each section with a briefing describing who we were, what we were trying to do, and what we needed of them. We let them know who the people were in the room and how they would just be taking notes as the tester was performing the tasks. We explained to them how in no way was anything their fault in performing the tests and that they should just try to do the given task to the best of their abilities. We let them know that none of their personal information would be given out.

We gave each user three tasks to complete while using the application:

1. Discover their voice part and return to the main screen to view the results.
2. Find and perform the given exercise.
3. Favorite an exercise so that is shows up on the main page, for future use.

After they completed all the tasks we asked for feedback and asked them what they were thinking as they went through the different tasks.

Incidents

Throughout the process we isolated multiple design problems from user testing.

Many users had problems finding out how to favorite exercises. When told that to favorite an exercise they must click the star icon, many replied by saying that they thought it was only a bullet point in the list of exercises. To remedy this and to make the star icon stand out more against the list, we made it look more like a click-able part of the interface.

Users also had problems using the discovering voice section of the application. It would usually take the user a couple seconds before they clicked a piano key and determined how it worked. We realized that there was not enough instruction on what exactly the user was suppose to do. In the future, a different widget should be designed with clear cues.

Sometimes instead of clicking the browse button, users would click the "tone" button and get lost. We realized that the label "tone" might not be the most intuitive label. To fix this we re-labeled it "piano", which users found better represented the functionality of that part of the application, as well as avoiding confusion with the exercises.

We also found problems with the code itself during testing. We had issues earlier in which the descriptions between the different exercises would change due to a bug. We fixed the code so the prototype did not do this anymore. We also had problems with text overlapping and getting squashed together. These issues were formatting problems in Java and were easily addressed to make text appear correctly.

Conclusion

Throughout the course of the project, we have attempted to create an application which can perform
all of the warm-up tasks a vocalist usually carries out with a piano. In doing this, we had to successfully translate a physical instrument into an interface on the computer that replicated the tasks that singers are used to carrying out, while adding all of the advantages of being a piece of software.

Some of our most difficult tasks in the iterative design process for this project arose from the challenges of condensing the piano into a screen sized application that provided the same affordances of the instrument. Combining our knowledge of computer science, music, and interface design, we created several different design ideas for each feature of the program.

By subjecting each stage of our interface to user testing, as well as soliciting the evaluations of our peers, we were able to isolate usability problems that we had not considered. We found, through these evaluations, that users desire to transition to the app for their every day warm-up tasks was highly dependent on initial learnability and efficiency of the application. Considering this, our focus in developing the interface was to combine these two metrics by creating an interface that was simple and minimalist in design, while adding features to promote efficiency for the learned user. We focused the application on the discovery of voice type, and the performance of exercises. The browse dialog in particular is an example of this: new users can look through the entire catalog of exercises while an experienced user can favorite certain exercises, so they can be played in seconds from the main screen.

In today’s climate of portable electronics and lightweight utility applications, we attempted to create an application which allowed singers in any location to, with a smartphone or other device, warm up their voice without requiring bulky instruments or equipment. In the same family as guitar tuning applications or recording applications, we hoped to use the creation of such an application to examine the unique challenges of designing software for use in the fine arts. Iterative usability testing with singers and vocalists has demonstrated both the viability of the niche market for music applications of this nature, as well as the vital importance of user feedback in the design of such applications.

Acknowledgements

The work of this paper has been supervised by Timothy Bickmore and Lazlo Ring.

Heuristic Evaluators: Matthew Tebaldi, Ryan Williams, Benjamin Doyle, Deniz Ozkaynak.

Testers: Members of NU Acapella groups, especially the UniSons.

References


