Team Gouda

HFID Final Report

Ben Hill, Kathy King, Sean McBride
December 17, 2005
Abstract

This is the final report for Team Gouda from the Fall 2005 section of Human Factors Interaction Design at Franklin W. Olin College Of Engineering. Our chosen project for the semester was the design of a digital music player. In this report, we discuss the design process as well as the final product, the Blender.

Problem statement

Although this project began as an effort to design a more effective way to create music playlists, in the end we wound up eliminating them altogether in favor of a completely new metaphor for playing music.

ORIGINAL MOTIVATION:
Playing music on a computer is easy. Finding, organizing, and playing a desired mix of music is inconvenient. As we found in a small, informal survey, some Olin students, despite frequently listening to music on their computers, very seldom create playlists. These users either listen to one album at a time or randomly shuffle their entire collection. However, these same students would very much like to have some way of easily hearing "good" mixes of their music.

FINAL DESIGN GOAL:
Design a digital music player interface that makes the process of selecting and playing songs as convenient and intuitive as possible.

Solution overview

We created a music player nicknamed the Blender, after the way that it mixes music. Some of its key features include:

- The Blender separates the organization and storage of music from the playing of music. This serves to clear up confusion present in some players about exactly which music out of an entire collection is actually playing.
- Music is stored in the “fridge”, where tabs expand to allow users to access all types of music. Here they can see the audio files saved on their own computers music, as well as radio stations, CDs, and friends’ music. Within the tabs, flexible sorting and display options simplify the process of selecting music.
- Care was taken to ensure consistency in the “fridge”. Each of its tabs has similar appearance and functionality. In addition, each type of music (be it a local file, radio station, CD, or something else) is treated equally and can be manipulated in the same way as other types of music.
• The music flow is separated into three main sections. Selected songs or groups of songs can then be sent directly to Now Playing or the Coming Up Next list, or they may be sent to the Current Blend. Songs in the Current Blend are mixed together and then dropped into the Coming Up window. This way, the player does all of the work of shuffling, but the user still has the opportunity to look at upcoming songs and veto any that she doesn’t want to hear.

• The Blender also learns users’ tastes by observing their listening habits and through their feedback about songs. Because the player keeps track of users’ preferences, it is possible to “autofill” the blender – generate a mix of music that begins playing immediately with a single click.

Personas

After interviewing users of existing digital music players, we created a set of personas. These fictitious characters are amalgams of the people we interviewed, and serve to guide the design process. Each has their own story, goals, needs and desires.

Jordan is a 22 year old college senior majoring in graphic design. He enjoys music and the ability it has to entertain and to make other people happy. He takes pride in his extensive music collection, and is always interested in new and eclectic music selections.

Juliet is 36, a full-time stockbroker with two small children: she has next to no time to play around with digital music. Aside from a few favorite CDs, most of her music comes from the radio. She values time above all, but musical variety is also important to her.

John, at 25, has just started working on his MBA full time. He is a regular digital music listener with a fairly extensive collection on his computer. The two most important things to him are making sure that his music stays organized, and that it doesn’t interfere with his other tasks.

Jamie is an 18 year old college freshman studying liberal arts. She listens to digital music regularly, mostly with the help of her more technical friends. While she is quite familiar with her system, she has no desire (and usually no need) to muck with technical details. All she wants is to hear music that she likes.
We chose Jamie as our primary persona, the main focus of our design efforts. Her goals are simple: she wants to hear music she likes and be entertained. If we succeed in designing a player that pleases Jamie, it will also satisfy the other personas. We also took the goals of the other personas into account as secondary considerations while designing our player.

**Scenarios**

The next step in the design process was the creation of scenarios – different situations in which our personas would use the music player. These scenarios are important in understanding what types of interactions any design will have to provide. In the interest of time, we decided to narrow our design space to focus on the music playing interaction, and deemphasize the other aspects of the player. We developed four scenarios, listed below.

1. Play a single song.
2. Play several particular songs.
3. Play a blend of music.
4. Let the player select songs for you, based on your earlier preferences.

We used these scenarios to create storyboards of the interactions between our personas and prototypes of the player.

**Final interface design**

The Blender as it appears when first opened. Note the help overlay.
The Blender in action, with the Library tab and the Recently Played list expanded. Notice the several different groups of music in the Current Blend and the corresponding mix in the Coming Up Next window.

**FUNCTIONALITY**

The final prototype of the Blender as implemented has nearly all of the functionality of an actual music player. You can view a collection of music in the Library (the only tab actually implemented in our design) and play actual clips of those songs with functional play controls.

The flow of music through the player all occurs in the blender portion, on the left. At the top there is the Current Blend section, which can hold different groupings of music (e.g., a particular artist, genre, or album, or a particular song). Songs from these groupings are then blended together and drop down to the Coming Up Next window. Also in this section are controls for working with blends. You can select The Usual, which fills the Current Blend with music you usually like to hear (as determined by the player), or Surprise Me, which fills the Current Blend with music you may not have heard in a while. You can also save and recall blends that you make.

Music from the Current Blend drops into the Coming Up Next window. Here, the next five songs to be played are shown. This allows the user to delete upcoming songs that they do not want to hear.

The Now Playing window is where the play controls for the currently playing song reside. There is a section with information about the song, including a working time slide bar. In this window, there are also Love It and Hate It buttons, which influence future selections that the player makes. The standard Play/Pause, skip, and volume controls are functional. This section also holds general controls, such as a
button to recall the help overlay, undo/redo buttons (non-functional), and a Start Over button to reset the player.

The Recently Played section can be opened to display the history of songs played. These songs can be deleted from the list, if the user so chooses.

Most of the browsing features of the Library tab are implemented. There is a box on the left side of the Library with different groupings of music (e.g. Artist, Genre). Selecting an item from one of these groupings narrows the Library to show all the music that has that particular attribute (e.g., all the music by Lynard Skynard). The library section can be sorted by any one of its columns in ascending or descending order.

After selecting a song or group of songs, you can then click one of the three color coded buttons to send those songs to the corresponding place in the play flow. In the actual design, this would usually be accomplished with drag-and-drop, but this was not implemented.

**INTERACTION FLOW**

The above diagram shows the main interaction patterns for the Blender. The music flow is separate from the music storage, and music collections can be moved around to any part of the player. Different actions are possible at each step of the way, as noted in the diagram.
Prototyping Tools

We created this prototype using Macromedia Flash. We chose to use Flash because of the highly visual nature of our interface. A simple HTML mock-up would not be convincing as a standalone application, nor would it permit the interactions that are possible with our interface. In Flash, we can create an interface that looks exactly as we envision it.

There were some difficulties with our choice. The initial problem was that none of us knew how to develop in Flash. The initial learning curve for making things more complicated than simple buttons was a little bit steep, but we were pleased to find that once we had some more experience, working in Flash became fairly quick. After getting started, there were still a few major problems. It is easy to create messy, scattered code in Flash that is difficult to follow, and impossible to develop in parallel. But the biggest problems were the limited interactions possible. Using standard Flash classes, we could not implement either double-clicking or drag-and-drop functionality over most of the player. These were intended to be the most common ways for manipulating music collections, with the buttons serving mostly to assist users with dysclicksia. Although it would be theoretically possible to mimic most of the desired interactions in Flash, it would require far more customization than would be feasible for this initial prototype. Consequently, we were not able to implement the full range of interactions that would be possible with our design.

Still Unimplemented

Since we made the decision to narrow the scope of our design to the music playing functionality, there is still much to be done before the player can be considered complete. But just within the music playing parts, there are still some things we would like to eventually add. Most of what is missing was given considerable attention in the design process, but was not possible to implement.

Chief among the omissions is drag-and-drop and double-click functionality throughout the player. As mentioned above, this is a limitation of Flash. The other tabs in the Music Storage area were designed, but not implemented due to lack of time. In the Library area, we would like to add search functionality, as well as to allow more than one of the groupings on the left side to be open at once. This latter omission is a limitation of the Flash class used in the implementation. If it were possible, having multiple groupings open would allow us to implement our drag-and-drop approach to tagging music by categories.

Design evolution

Design Goals

We began our design process by identifying our each of our persona’s individual goals. Since these personal goals guided our design decisions throughout the process, we feel that it is important to reiterate them here. We focused in particular on Jamie, our primary persona. For Jamie, a simple, intuitive interface is important. Also important is the ability to easily access all of her music and her friends’ music in one place, since she hates to spend time digging around in her computer. Playlisting is not important, but she does like to be able to play different mixes of music to suit her various activities.
Jamie’s goals guided our design process, but we also wanted to satisfy the rest of our users. For instance, we noticed that all of the people we interviewed spend most of their time listening to some subset of their music on shuffle, so we wanted to make some sort of shuffling functionality integral to the player. We also found that, even among experienced users, problems often arise when the user tries to listen to music and sort through their collection at the same time, so we wanted to create a clear separation between music-playing space and music-storing space. Additionally, we wanted to include several other nice features that would make our other personas particularly happy: a means of playing music with a single click (for Juliet); a way of training the player to avoid songs that the user dislikes or discover songs that haven’t been played recently (for Jordan); and an easy method for tagging and organizing songs (for John).

**INITIAL DESIGNS AND PAPER PROTOTYPES**

We sketched out three initial designs: KISS (Keep It Simple Stupid), the Music Menu, and the Conspiracy Shuffle. Each of these designs focused on one of our design goals. KISS tried to make all of the basic functions as simple and straightforward as possible without introducing anything extraneous. The Music Menu extended the metaphor of a restaurant, laying out your choices clearly in a menu format and remembering your prior preferences. The Conspiracy Shuffle was optimized to play music in a shuffled order. The more you play, the more it learns from you without any direction, although you could also provide suggestions, vetoes, and feedback.

We then combined these ideas into our first paper prototype. The primary innovations here were the separation of music playing and music storage and the invention of blending to replace shuffling. We tried to keep the interface simple as in KISS, while adding in the sliding menus of the Music Menu and keeping some of the personalization features of the Conspiracy Shuffle. The player looked fairly similar to currently available music players, with added features, like blending and displaying lists, kept out of the way by tabs. We were very excited to test this player in our first user interview.
The player’s debut was not a success. Its similar appearance to other players made the user assume that it would work like other players. The user did not understand the separation of music playing from organizing and didn’t notice the tabs to extend the Current Blend and Coming Up windows for quite some time, despite their bright colors. Trying to use our player was an extremely frustrating experience.

So, we went back to the drawing board. First, we decided to make the options for what to do with selected songs in music storage more apparent. Also, we made the Blend and Coming Up windows remain visible on the default screen at all times. Most importantly, we decided we needed to give the users some strong visual cues that this player was very different from other currently available music players. Extending our metaphor of blending music together, we chose to create a player shaped like an actual blender, with music stored in an adjacent refrigerator. This also gave a clear visual separation between the play flow and music storage.

Our new paper prototype received a much better response. Users figured out the new functionality much more quickly, although they still had some trouble initially. As a quick solution to this, we created a help overlay that would give users a brief introduction to the player when they first opened it, and that could be reopened at any time. The addition of this simple overlay had dramatically improved users’ understanding of the player.
Overall, paper prototyping was the most useful stage for our group. The high level of flexibility allowed us to make both drastic and minor changes to our prototype quickly. We were not limited by implementation realities, so functionality like drag-and-drop was easy to design. In general, direct user testing was always very helpful to us throughout the process. Observing users in action and hearing their offhand remarks gave us the most insight into what they really thought and felt about our design.

**INTERACTIVE PROTOTYPES**

For our first interactive prototype, we tried to implement enough of the player’s functionality to give users a feel for what it would be like to use this player. The first thing that we did was design a sleek-looking interface in Fireworks, in response to user feedback that our garishly-colored paper prototype did not look much like a blender. Another change we made was adding a “Blend It” button that would let users open or close a gate between the Blender and the Coming Up window. We did this because users became confused when the “Clear” button for the Coming Up window did not actually clear the window, but rather refilled it from the blender.
Heuristic evaluation told us that our hasty addition of the “Blend It” button was a very bad idea. It added a mode to our player, and actually increased confusion. Our evaluators helped us come up with a better solution to the problem, which was to simply change the name of the “Clear” button to say “Refill” when its function was to refill the Coming Up window. Our evaluators also suggested that we allow users to save their favorite blends, which we implemented in our second prototype.

Our first interactive prototype, with the Library tab expanded

We tested the second prototype with users, who responded positively. They still had some trouble figuring out the Current Blend functionality initially. To correct this we added a help overlay as in the second paper prototype. Users have also had difficulty discovering that it is possible to send groups of songs (e.g., all of the songs by one artist) together to the blender by selecting the group heading (e.g., the artist’s name). We would like to graphically indicate that group headings represent groups of songs by giving them icons with multiple musical notes and giving single songs icons with single musical notes, but this is not easily possible using the List class in Flash. However, we were able to ease the confusion caused by the verb-noun pattern of the “Send Selection To” buttons and by the direction of music flow in the Coming Up list. To correct the latter, we have numbered the songs in the Coming Up window, increasing from bottom to top, to indicate that the next song to play is at the bottom. In the final prototype, we have changed the names of the “Selection” buttons to a less confusing noun-verb construction, and we have added different colored backgrounds to the three sections of the play flow and their corresponding buttons.
Conclusions

We were able to design a completely new metaphor for a digital music player, based upon how people actually listen to their music. We learned a lot from the design process, as the changes to the player concept suggest. Initial feedback indicates that the Blender concept is both an interesting and effective solution. The design is also sufficiently novel to be exciting to users. Although there is still significant work to be done before the player could be considered complete, the Blender design seems promising.