City Science and Sound

A Digital Exhibit Interface for use by the Patrons of the EcoTarium Museum

An Interactive Qualifying Project submitted to the faculty of

Worcester Polytechnic Institute

In partial fulfillment of the requirements for the Degree of Bachelor of Science

By

David Keeley-DeBonis

Eric Breault

Alex Margiott

Dennis Chen

February 24, 2013

Report Submitted to:

Dominic Golding
Worcester Polytechnic Institute

Abstract:

Our team designed an interactive exhibit to allow children and families to explore the concept of soundscapes for the EcoTarium, an environmental museum located in Worcester. We designed and thoroughly tested a Java-based digital interface for the museum. Our exhibit will become a part of the larger City Science and Sound exhibition, with the eventual goal of giving children and families a better appreciation for and understanding of the urban environment. We provided the EcoTarium with the interface itself and supplementary information concerning the use of the interface to allow them to expand upon or replicate the design.

Authorship Page:

Through each team member's research, effort, expertise, and dedication, we were able to successfully generate a working interface for the EcoTarium. Each group member had equal participation in the initial research, design, testing methodology, and field testing of the interface. Eric Breault and Alex Margiott were primarily responsible for writing of the final report. David Keeley-DeBonis was responsible for programming the application itself. Dennis Chen was primarily responsible for the collection and modification of sounds. The final interface and report were made possible by all of our group members working to the best of their respective abilities.

Acknowledgements:

Our team would like to thank our advisors, Dominic Golding and Anne Ogilvie at Worcester Polytechnic Institute, for their constant support of our efforts and their timely and constructive feedback. We could also like to thank Betsy Loring, Alexander Goldowsky, Alice Promisel, and Eric Zago at the EcoTarium for their support, advice, and guidance throughout every phase of our project. Our final design would not have been possible without their regular encouragement and counsel.

Executive Summary

The EcoTarium, a science and nature museum in Worcester, Massachusetts, is creating a new exhibition, City Science and Sound, which will encourage visitors to examine the nature of the urban environment through a series of hands-on, interactive exhibit elements that highlight key scientific concepts. The EcoTarium approached our IQP team to develop and test a prototype exhibit with the tag line “Create the Sounds of Your Neighborhood,” that would be part of this larger exhibition and enable visitors to explore urban soundscapes.

The EcoTarium envisioned an exhibit that would allow museum visitors to create soundscapes by overlaying sounds from a sound bank and placing them on a timeline in an interactive virtual environment on a touch-screen interface. The sounds would play back to the user, playing the soundscape they had created. The interface would allow for an iterative editing process, during which users would develop their soundscape beyond their initial design. The design would need to be intuitive enough that visitors who were less familiar with digital programs could still interact with and learn from the exhibit.

Following several stages of research and iterative testing, we developed a Java-based, touch screen interface which allowed users to generate and listen to their own soundscapes by selecting from a library of possible sounds. We also developed a prototype of the surrounding educational materials to guide museum guests in their use of the interface.

An example of an interface-created soundscape

The interface, seen above, allows users to move icons from the top section of the screen onto the canvas at the bottom, where they automatically play and combine to generate a soundscape. Key features of this design include drag-and-drop functionality, a clean and cartoon-like visual appearance, a 'Clear All' button, and the ability to easily move sounds within that canvas. These features were driven by user needs for clarity, simplicity, and perception of control that our evaluations revealed. Simplicity became a dominant design feature as we realized the wide differences in technological familiarity among different museum patrons, particularly the distinctions between younger visitors and older guests.

Based on our experience developing and evaluating the exhibit prototype we draw several conclusions concerning design for the museum environment. On the design level, there is enormous value in ensuring consistent and frequent feedback to the user and that visual cleanliness is central to effective interface design. Concerning the evaluation process in a museum setting, we concluded that simple data collection techniques are often more powerful and productive than complex techniques and that polite interaction with visitors in the museum setting is an acquired skill. Finally, we conclude that enabling the process of exploration to encourage learning is the most important attribute of an exhibit.

Table of Contents

Table of Figures

Table of Tables

1. Introduction

Museums exist as establishments for the education of the general public. As such they must constantly evolve to meet the meet the many needs and expectations of their audience. This presents a unique challenge; as technology and society evolve, museums must constantly adapt to engage visitors. A modern children's science museum exhibit must confront these challenges in order to achieve its goal of providing an engaging and educational experience to the museum patron. It must compete for attention with an increasing number of alternative entertainment options, provide an environment which a museum patron will spend a prolonged amount of time at, and it must be able to serve as a point of discussion between children and their guardians to encourage further exploration of the subject (Allen & Gutwill, 2005).

Ideally, a children's museum would provide a large number of highly engaging, unique, and educational experiences for its patrons. However, the difficulties of creating engaging interaction, prolonged engagement, and educational experiences relevant to various learning styles make this difficult to achieve (Yilmaz-Soylu & Akkoyunlu, 2009), (Lord, 2007), (Allen and Gutwill, 2005). Current museum exhibits face these problems as they attempt to transition their role from one of simply displaying information and using text as a primary educational tool to a role which combines voluntary and enthusiastic interaction with education. In particular, they must best determine the proper use of emerging technologies in museum exhibits (Falk, Dierking, & Foutz, 2007).

The EcoTarium, a Worcester children's environmental museum, faces these challenges with their decision to create an exhibit on soundscapes as part of a series of exhibits on the urban environment. The EcoTarium approached our IQP group to help them design an exhibit that would overcome these challenges and provide a learning environment to their visitors. Our project was to generate a highly interactive exhibit concerning the idea of soundscapes, which would likely serve as ambient noise to other, surrounding exhibits (See Appendix A).

The purpose of our project was to generate a prototype interactive exhibit to educate and engage children in learning about the role of soundscapes in their daily lives. The EcoTarium wished to create an exhibit to provide an educational experience concerning city soundscapes based around the concept that users would be instructed to "create the soundscape of your neighborhood." This was to be accomplished by way of an interactive touch screen interface that would facilitate the active design of a soundscape through the overlay and playing of sounds. One goal of this interaction would be to show the juxtaposition of natural and technological sounds to illustrate the concept of noise pollution. By using the exhibit, users would become more aware of the sounds of their neighborhood and would ideally carry this knowledge into being more conscientious of their sound environment.

The EcoTarium staff also had a specific design vision that they hoped to implement. The idea put forth was a touch screen environment where users could select sounds from a bank and place them on a timeline to be played back to the user. The playback would hopefully be in real time and would allow for iteration and improvement towards a desired sonic environment. The user should also be able to have visual feedback to accompany their soundscape in order to facilitate the interactive experience, meaning that imagery should drive the museums functions which in turn intuitively output sounds.

To accomplish this, our group researched previous implementations of similar museum exhibits, talked extensively with EcoTarium staff and professionals with skills related to the child education or relevant technological fields, and examined current interactive exhibits available at the EcoTarium. We then used this information to develop a Java-based initial prototype exhibit. This prototype went through seven successive rounds of iterative testing with patrons and staff at the EcoTarium. Through this research and testing, we generated what we believe to be an intuitive, educational interface and supporting educational materials for a future soundscape exhibit. With our prototype we hope to give the EcoTarium a tested solution to their design challenge that is both intuitive and functional, but also engaging and educational so that visitors to the EcoTarium can ultimately leave more informed about the sounds that surround them in their environment.

2. Literature Review

The three primary roles of modern museums are to educate, maintain collections and conduct research. The ability to carry out each role is limited by resources. Many museums serve these roles to different degrees; however, each has the responsibility to educate (Hein, 1998). The development and incorporation of technology has changed the way that museums are able to achieve their educational goals. Museums are now able to serve their educational role though a variety of highly engaging, interactive exhibits, supplemented by online discovery activities, school field trips, and other programs (Alexander & Alexander, 2007). The approach to education has also changed as educational models have shifted. Current models replace older ones which emphasized bland and static exhibits that were typically directive and heavily text-based with ones that are designed to be engaging, interactive, and entail open-ended discovery. Visitors are now encouraged to explore and experiment with exhibits capable of giving and receiving multisensory output and input. Evaluations of visitor interaction with dynamic exhibits have reinforced the shift in the educational approach (Tallon & Walker, 2008). These evaluations focus on the learning needs of different audiences and have provided insight as to how to appeal to each learning style. By understanding how their audience learns through a combination of front-end, formative, and summative evaluations, museums are able better design exhibits and create the ideal learning environment. As noted by Beverly Serrell,“Unobtrusive observations of visitors as they move around an exhibition—interacting with each other and with the exhibit elements—give important information about what visitors do, especially how much time they spend in the exhibition and with what parts of the exhibition they become engaged”(Serrell, 2012). Evaluations reveal that the visitor experience has been greatly enhanced through the incorporation of interactive exhibits. The use of digital technology has made it far easier for museums to successfully create interactive and educational exhibits, particularly with respect to nonvisual senses. A digital exhibit may be the ideal medium for creating a highly interactive soundscape (Pekarik et al, 2002).

2.1 The Museum Environment

2.1.1 Museums as Educators

While originally established as a means of displaying and preserving art and scholarly works, museums have adapted to promote education through discovery and provide their visitors an engaging learning experience. George Hein, renowned for his work analyzing the implementation of educational theory in museums, states that as long as museums have been public, they have existed for the edification and entertainment of the public. The theory driving the need for an educational component in the museum environment has existed ever since governments have possessed the perceived role of responsibility for the social well-being of the public (Hein, 1998). The emphasis has been placed on exhibits being both attractive to the public eye and constructive to the public's education. Museums are expected serve a number of roles in society. Emphasis on the development of educational theory is prevalent in many museums, while remaining particularly significant in science museums. Educational theory in science is driven by analysis of different types of theory, including theories of learning and knowledge. By gaining an understanding of these theories, the ideal educational environment can be obtained (Falk, Dierking, & Foutz 2007). Museums must interpret and adapt these theories of learning, taking into account the expectations and desired explorations of their visitors. They often must design environments tailored to the public's inquiries. In order to facilitate the elucidation of their visitors and fulfill public expectations, museums have undergone a rather recent and drastic change (Hein, 1998).

2.1.2 The Technological Revolution in Museums

As computers have become increasingly ubiquitous, museums have started to incorporate interactive technology as a means of further engaging their audiences. This increase in social acceptance of technology, in tandem with an unprecedented recognition of the overwhelming benefits that technology can bring to a learning environment, has been a powerful tool towards the application of educational theories in museum settings. The implementation of digital interfaces has also served to increase visitor understanding of various forms of art. Technologically integrated exhibits are often embraced as a learning tool because they allow for novel presentation of information and may be more adaptable than physical exhibits, allowing exhibits to more easily change along with public interests. The move towards integration of digital technology has also been driven by the trend towards "personal relevance and interpretations, interactivity and easy access and control of content to shape the twenty-first-century museum visitor's experience" and Loic Tallon and Kevin Walker, in their book Digital Technologies and the Museum Experience, identify "todays museum visitors [as] less audience than they are author" (Tallon & Walker, 2008, xiv). Nina Simon expands on this idea in her book *The Participatory Museum*, outlining techniques which museums may use to have their exhibits driven by user generated content (Simon, 2010). Technology is used as a means of "customization in the public service arena of museums [and] draws on the features of the business-world definition but also takes into account the distinctive features of museums and their role in communities and the larger society" (Falk, Dierking, & Foutz, 2007, 80). An environment that provides multisensory input and output promotes a sense of connectivity between the exhibit and the audience. This is the fundamental idea of interactivity, a major contributing factor to learning in many museum guests.

2.1.3 The Audience

Along with an increased focus on the educational process, museums have also more closely studied the demographics which make up their audience. In recent years, many science museums have come to recognize that their primary audience is composed of family groups. As museums make efforts to increase their appeal among both their core demographics and wider audiences, they are increasingly recognizing the utility and educational value of the incorporation of digital technology. Museums have become an important destination for school field trips, families, and visitors of all ages. In 2008, The Center for the Future of Museums (CFM) was launched by the American Association of Museums. CFM identified the changes in the demographical, cultural, technological and economic state of the museum community. Based on survey data, the CFM predicted the future state of the museum community. They found that, "…museum audiences are radically less diverse than the American public…" (Betty & Medvedeva, 2010, 1-15). The report goes on to highlight that today's science museums, the visitor community only consists of around nine percent minorities. This is projected to change substantially as minority populations grow, relative to the overall population. Museums are challenged to develop exhibits that appeal to their patrons despite shifting and diverse demographics. Possession of different backgrounds alters guests' perception of and ability to interact with an exhibit. An exhibit that does not facilitate multilingual interactivity will have decreased educational value and may result in unmet learning goals. It is vital to consider local ethnic demographics, as many museums are located in areas with sizeable minority ethnic groups. Accommodating and promoting an environment that is educational to all increases the museum's visitation and overall contribution to the surrounding community (Betty & Medvedeva, 2010, 15-21). In order to provide the optimal learning experience and maintain an engaging environment, it is important for the exhibit designer to understand how different groups learn and the role technology can play in their education.

2.2 Learning in Museums

2.2.1 Methods for Learning

Museums attract a wide range of visitors, including school children and families, who attend for a variety of reasons ranging from cultural enrichment, to simply keeping children occupied. Whatever the reason, museums try to engage their visitors and provide them with information in a way that will promote learning. Learning involves using the five senses to differentiate between things and make connections, but individuals will apply these senses and interact with their environment in numerous ways. The senses most commonly used to learn in museums are touch, taste, and sight (Hilke, 1988). Hilke separates the strategies individuals use to learn new information broadly into first-hand learning (direct experience) and second-hand learning (indirect experience). The strategies used for first hand-learning include gazing, looking intently, manipulating objects, move-on looking, and touch (Table 1). For second-hand learning, people either read text or listen. By observing visitors on the museum floor, Hilke found that 82% of visitors use first-hand learning strategies to acquire their information as opposed to just 18% using second-hand strategies.

Learn First-Hand	Learn Second-Hand
Gaze 33% (495)	Look/read text 17% (241)
Manipulate 24% (358)	
Move-on-looking 16% (241)	Listen 1%(17)
Touch 6% (94)	
Look intently 3% (49)	
Total: 82% (1237)	**Total: 18% (259)**

Table 1- Strategies of First-Hand Learning (Hilke, 1988, 128)

Another way that people learn is by asking questions, both actively and subconsciously. When visitors, especially those who are part of a family, visit to an exhibit, they focus on finding anything that is useful, interesting, and or engaging (Hilke, 1988). They will then subconsciously ask themselves a series of questions that will help differentiate the type of information they want to obtain. These may include questions such as:

- What looks interesting?
- What do I not understand?
- What do I not recognize?
- How is the stuff here related to the stuff I know?
- Is there anything to do here?

(Hilke 1988, 124)

These questions essentially reflect the mindset of any museum visitor, especially family members, who want an interesting learning experience for their children. Besides these techniques, different people learn in different ways. The four most common learning styles are:

- Accommodator: learns best when they are taught using active experimentation, such as implementing decisions or setting goals, and concrete experience, such as participation and communication
- Diverger: learns best when they are taught using concrete experience and reflexive observation, such as gathering data or information and listening

- Assimilator: learns best when they are taught using reflective observation and abstract conceptualization, such as testing or working with theories or concepts
- Converger: learns best when they are taught using abstract conceptualization and active experiments
(Yilmaz-Soylu & Akkoyunlu, 2009)

2.2.2 Age and learning

Differing learning styles are not the only significant difference between museum visitors. As museum visitors range in age from toddlers to senior citizens, it is important to identify the target audience in order to "help narrow the broad band of possibilities and allow the museum to make best use of its resources" (Lord, 2007, 173); not only will each group utilize different learning techniques, but each will bring with them varying degrees of education. For example, younger children are more likely to move around more, and tend to experience everything first hand, most likely by touching or manipulating items. For these children, an ideal learning environment might "include lots of opportunity to do things- more complex puzzles, discovery boxes with magnifying glasses and other simple tools, working with clay or plasticine, or turning a big crank to make a music box work," (Lord, 2007, 174) whereas adults are better aware of their surroundings and are better able to think logically. As their thinking is more complex, adults can "conceptualize the abstract, and to hypothesize and come to conclusions" (Lord, 2007, 176). Thus, the way adults learn is more complex than how children learn; they might sign up for lectures, films or classes, and may also volunteer at the museum. By using second hand learning, adults are able to acquire information that they would not otherwise have been able to acquire first hand.

In museums, children will usually be under adult supervision, whether by parents or by other chaperones. These adults serve as teachers in order to translate the information from the exhibits to the children as best they can, even though the information taken in is based on perception. To children, "experience is external, something that happens to them; to adults personal experience has defined their individual identity. As adults have a richer foundation of experience than children, new material they learn takes on heightened meaning as it relates to past experiences" (Jensen, 1999, 112). Thus, in order to create a family friendly exhibit, the

Philadelphia-Camden Informal Science Education Collaborative (PISEC) conducted three phases of research and development to determine what qualities make an exhibit family friendly. These exhibits would be multi-faceted, multi-user, accessible, multi-outcome, multi-model, readable, and relevant (Borun, 2010). Many exhibits targeting different ages and learning styles have been created using these PISEC characteristics. Exhibits are also made interesting and inviting using the APE (Active Prolonged Engagement) evaluation tool, in which the exhibit is designed to keep the visitor actively engaging and interested for longer periods of time (Humphrey and Gutwill, 2005). This is so the visitor can draw out the exhibit's full potential, while staying actively engaged.

2.2.3 Learning Goals and Outcomes

While at a museum, visitors typically want to either learn something new or to make a new connection to existing knowledge. Depending on the target demographic, the goals of learning may be different. For children, they might wish to learn all about dinosaur bones; adults may wish to learn how much their children enjoyed the exhibit and what information they took away. Museums conduct visitor studies "because [they] are interested in finding out what visitors think and how they feel about their visits" (Hein, 1998, 100). Hein identifies three ways to observe visitors and record their movements and actions:

- observing what people do;
- listening to conversations; and,
- examining products of human activity, such as wear patterns on the floor or nose-prints on display cases.
  (Hein, 1988)

These encompass many methods which can be used to directly observe visitors. Another method that does not follow these categories, but can provide a rich source of information, is the comment card. As Hein stated, "Many museums collect comment cards from visitors, but some never use them. They can provide a rich source of ideas about visitor concerns, and used comparatively they can provide excellent information about visitor response to exhibitions and program" (Hein 1998). Although some people will not fill the cards out, the visitors that do take the time to fill out these cards may provide insightful information concerning the effectiveness of the exhibit.

While being able to evaluate and revise an exhibit as necessary important in ensuring that an exhibit fulfills its learning outcomes, the primary tool for accomplishing these remains the initial design of the exhibit. The design intent of any exhibit is directly tied to its eventual performance.

2.3 Exhibit Design and Development

Most exhibits have the same eventual goals, closely correlating with those of the APE and PISEC schemata. However, the design and iterative testing of an exhibit often serve as the means by which an exhibit may achieve these design goals. A variety of organizations and individuals have looked into the question of how one can develop an ideal exhibit.

2.3.1 Interactive Exhibits

"At the heart of interactivity is reciprocity of action, where a visitor acts on the exhibit and the exhibit reacts in some way" (Allen and Gutwill, 2004, 2). An interactive exhibit provides a unique opportunity for enticing and engaging museum visitors that static displays lack. Visitors tend to prefer exhibits that are interactive in some way, as it provides them the opportunity to physically involve themselves in their learning and exploration (Hein, 1998; Allen and Gutwill, 2004). Using interactive exhibits allow users to explore, in addition to simply viewing, which can ultimately lead to the excitement of discovery that an unresponsive display often cannot. Interactive exhibits actively use the input of the user to stimulate engagement and interest in a topic. Digital technology is on the forefront of many new interactive exhibits and allows for new elements in the museum environment.

Digital interactive technologies enable "visitors to customize their experience", extend "the experience beyond the...boundaries of the museum", and layer "multisensory elements within the experience" (Tallon and Walker, 2008, 27-28). These choices and dynamics create feedback between the exhibit and user which drive the learning aspect and goal of the museum.

Interactive exhibits emulate a small portion of the scientific method and allow users to experiment and try different options to achieve different results. These results need not be explicit; often users will learn about a topic simply through observation, while other topics will require more in depth feedback to the user. Multisensory and multidimensional exhibits create multimodal learning experiences which incorporate multiple senses, including sight and hearing (Fritsch, 2007).

Even so, these methods of engaging visitors are not unique to digital exhibits, and in fact are present in many non-digital exhibits. Digital exhibits are not necessarily superior to their physical counterparts, they simply differ in the capabilities offered and means of interaction. Both models have benefits and hindrances to visitor interaction, and creating an exhibit that engages visitors of a particular demographic becomes a key decision point.

2.3.2 Audience Attraction and Engagement

Based on research done by Falk and Dierking on learning in museums, several factors contribute to visitor learning and engagement at a museum:

Personal Context	● Visit motivation and expectations ● Prior knowledge and experience ● Prior interests ● Choice and control
Sociocultural Context	● Cultural background ● Within group social mediation ● Mediation by others outside the immediate social group
Physical Context	● Advance organizers ● Orientation to physical space ● Architecture and macroscale environmental factors ● Design of exhibitions, programs, and technology ● Subsequent reinforcing events and experiences outside the museum

Table 2- Factors Contributing to Visitor Learning (Tallon and Walker, 2008, 24)

These contexts shape how visitors interpret the museum, its exhibits, and the overall experience. Museums must take these diverse personal motivations and backgrounds into consideration when designing exhibits. An exhibit design that fails to account for these contexts runs the risk of not being able to engage users for any appreciable amount of time, ensuring that no learning will take place. Only through understanding the specific audiences of the museum can contextual improvements be achieved. Different demographics will find different aspects of an exhibit appealing, and a successful exhibit generally must attempt to interest as many visitors as possible. Universal appeal is an ideal that, unfortunately, cannot be realistically met. Appealing to a majority, or "the 51%," as Beverly Serrell calls it, becomes a necessary compromise. By focusing on a more stable average of visitors, an exhibit's success can be measured, disregarding outlying users who stay abnormal amounts of time at an exhibit or to who to the exhibit has no appeal (Serrell, 1998, 2-3).

Serrell also makes a direct connection between dwell times and learning, which is the ultimate goal of any museum. "Thorough use," as she calls it, represents "time well spent," and is a quantifiable method for measuring the success and appeal of an exhibit. Those who stay and participate at an exhibit longer generally learn more. The longer visitors remain at an exhibit, the more likely they are to remember the exhibit and what they learned as compared to exhibits where they spent less time (Serrell, 1998, 5-7).

In order for an exhibit to be successful, it must have both initial and prolonged engagement. Possessing both allows an exhibit to draw in visitors and keep them once they begin to interact. According to research conducted at the Detroit Institute of Art, museum staff were significantly over-optimistic and unrealistic of how long visitors would stay at exhibits.

Hopes	20 minutes
Expectations	15
Actual (mean)	4:16
Actual (median)	3:20

Table 3- A comparison of staff's hopes and expectations for exhibit view time as compared to actual values (Allen and Gutwill, 2004, 19).

These numbers for actual time spent are fairly typical, and indicate that any exhibit has only a very limited timeframe in which to draw in and engage visitors. In the time frame of two to four minutes, a specific exhibit must interest visitors enough that they stay for a more extended period. Users are generally drawn in by both the visual aspects of the exhibit and a potential interest in the subject matter. Different demographics are attracted by different design elements, and creating something that appeals universally is virtually impossible. For example, children may be attracted to exhibits by features which appeal to short attention spans, while adults may skip over these aspects. Literacy might also be an issue for some age groups or regional demographics. Appealing to larger groups of people demands an exhibit be multidimensional and approach an objective from multiple angles, in order for everyone who uses it to come away with some new knowledge or experience.

Museums have constantly researched how to increase dwell times at exhibits in order to maximize learning time. Research conducted by the Exploratorium (e.g., Allen and Gutwill, 2004) shows that learning is enhanced by exhibits that are able to promote active prolonged engagement (APE). APE focuses on creating exhibits that "provide visitors with opportunities to engage in their own scientific investigations, to question, wonder, and hypothesize" (Tisdal, 2004, 2, *Going*). In other words, APE exhibits differentiate themselves from other interactive exhibits in that they allow the user to create their own experience. APE exhibits, rather than guide in an often restrictive setting, allow for much more user control that is intended to lead to discovery. The critical aspect of these exhibits is that they have been proven to increase dwell times and user engagement across the board (Tisdal, 2004, *Going*). While it is not essential that an exhibit use APE characteristics, the undeniable evidence of its success should drive any interactive exhibit to use its interactive elements in order to increase its success.

Children are a high profile target audience and computer interactive elements in an exhibit often attract children more so than static exhibits simply because they are familiar with and attracted to computer games. One easy way to capitalize on this is to create an interactive environment that simulates aspects seen in video games and other similar applications. The issue with this approach is that it detracts from the learning environment. Interactive elements are usually enough to provide the initial interest in an exhibit, but may not be sufficient to ensure immersion in the exhibit (Cairns & Haywood, 2005). Only once the child is drawn in and is actively engaging and interacting for some time will the child begin to learn. It must then be realized that the simple existence of interactive elements at an exhibit is not enough to maintain engagement. Another drawback to an interactive approach is that it can often alienate the older audience if the interactivity displays an overtly game-like atmosphere. The challenge is to create an application environment that is simultaneously attractive to children and has enough educational substance and merit to be interesting to adults, allowing both to share a prolonged period of engagement with the exhibit. This can largely be avoided through parent-child group participation in the exhibit.

Engaging the audience beyond an initial draw is what allows an exhibit or experience to ultimately teach visitors. Through this engagement they become interested in the topics provided and stay at the exhibit longer. One frequently found phenomenon is that "individuals are much more likely to describe the outcomes of their museum experiences as strengthening rather than changing their existing knowledge structures" (Tallon and Walker, 2008, 26). Exhibits therefore must make a point to build on the existing knowledge-base of the users. While specific interest in a topic might draw knowledgeable people in, it is not often the case that museum visitors, especially children, are fully informed on a specific topic before arriving. Therefore, in order to draw users to an exhibit they have to make the exhibit appealing to those with little or no knowledge of the topic while still tapping into what preexisting knowledge they do have. By understanding its visitor demographics, a museum can more accurately and meaningfully engage visitors in a way that takes their general knowledge and builds it into an understanding of the topic at hand.

2.3.3 Visitor Technology Capability

The capabilities of the target audience must be taken into consideration when designing an exhibit. Whether interactive or not, if a visitor does not have the skills necessary to learn from the exhibit, then the museum has largely failed in its goals. In the specific case of interactive exhibits, users must possess the ability to interact on a multi-sensory level in order to gain the full experience and potential of the exhibit. "Strong support is found particularly among visitors to science centers and science museums, where digital exhibits have been shown to be immensely popular, with high attracting and holding power" (Tallon and Walker, 2008, 38). Ultimately, an exhibit, despite its potential for education, must teach visitors something, and if they cannot understand the interface or design of the exhibit they will have no chance of learning what the exhibit had to offer.

One of the major technological barriers is between younger and older museum patrons. Any technology implemented would need to be easily understood not only by young children, but also by an older generation of parents who may not be as technologically oriented. This can be difficult, but it can be mitigated by using simple interfaces that allow users to immediately jump in without having to spend time learning how to use the exhibit.

The use of touch screens allows for relatively simple clicking and dragging to be the primary functions that users perform. With a touch screen, users would not have to deal with mechanical functions, and interaction would be much more intuitive to both younger and older audiences. In creating a soundscape exhibit specifically, users must be able to understand how to select and mix sounds. The design of any buttons on the screen cannot get overly complicated with text, or even flashy images, as users may be distracted or unable to understand the functions of the exhibit. Drag and drop or touch to select methods are the simplest features to both implement and use. By using pictures as the primary button display while still including small text descriptors on buttons or on auxiliary materials, we can appeal to the visually minded younger audience as well as the older, more literate audience. It is also essential that images selected accurately represent the sound being played (Stone et al, 2010).

2.3.4 Creating the Interactive Experience

Interactive experiences, by definition, require an exchange of information between users and the interface. In order for users to willingly participate, the design must be simple enough to use, but also powerful enough to engage users and increase learning potential. Experiments conducted at the Science Museum in London showed many successes and shortcomings of technological exhibit designs that attempt to facilitate smooth use and interaction. They found that content and instruction must be much faster paced than typical game or home computer interaction. Visitors lose interest quickly and often leave if they cannot find information quickly, if the system fails to have a rapid response, or if there is any form of lag. Educational material must also be embedded throughout the interaction, rather than saved for the end when users want to move on. Touchscreens specifically are useful, as they allow for simple manipulation. However due to a lack of tactility, it can be difficult to determine what is touchable and what is not (Tallon and Walker, 2007, 45-46).

One critical element to creating an interactive exhibit specifically for children is that an exhibit must provide a connection to the real world. Children often use fantasy to connect what they see and learn to the real world, so in order to help make that connection an exhibit must not only be grounded, but also open ended enough to allow for unrestricted creative thought (Cairns Haywood, 2005; Tisdal, 2004, *Going*). If an exhibit is too guided and restrictive, it can actually stifle a child's desire to continue to interact and explore. Educational aspects of an exhibit should also follow this model as it allows children to make cognitive connections that are applicable to their lives.

Specifically in the case of digital soundscapes, an exhibit will be most applicable and relevant to visitors if it allows for connections to their daily lives. By allowing visitors to create and guide their own experiences, they will be more engaged and interested as they will be able to make useful connections while at the same time learning more about community soundscapes and their features and impacts. With a museum's ultimate goal of education, any interactive exhibit involving soundscapes must foster creativity and control or the overarching learning objectives will be lost amongst and unappealing, structured environment.

2.4 Soundscapes

2.4.1 Defining Soundscapes

Soundscapes are a shared experience about which few people are consciously aware (Schafer, 1994). This may make soundscapes ideal as interactive experiences to which all visitors can connect. The notion of a soundscape was first formalized by R. Murray Schafer in his 1981 book, *The Tuning of the World* (Wrightson, 2000). It is the subject of the scientific field of Acoustic Ecology (Wrightson, 2000). As put forth by Schafer, soundscapes consist of a combination of high-fidelity (Hi-Fi) signals, those that are highly distinct and can be heard clearly, and low-fidelity (Lo-Fi) signals, which are less discernible (Schafer, 1994). Pre-industrial areas and rural environments tend to have higher fidelity signals, where each sound occupies its own niche of frequencies and can be distinguished from other, simultaneous sounds (Wrightson, 2000). In more urbanized areas, however, constant noise due to traffic, technological sounds, and close human proximity results in a Lo-Fi soundscape containing little useful information for humans. This problem is compounded by the effect of "the noise generator," individuals using music or other sounds as a mask to try and block out annoying sounds, resulting in an even louder and more muddled overall soundscape (Wrightson, 2000). However, these localized sounds may contribute to the unique identity of a location, an example of which has been observed among shopkeepers in Osaka, Japan (Kreutzfeldt, 2010).

2.4.2 Classifying Soundscapes

There is no universally accepted scheme to classify soundscapes. According to research by Axelsson, Nilsson and Berglund, soundscapes may be categorized on the basis of the three qualities of Pleasantness, Eventfulness, and Familiarity (Axelsson et al, 2010). Technological sounds were found to, in general, be unpleasant. Even in soundscapes in which natural sounds dominate an otherwise natural soundscape with a low frequency technological sound in the background, such as distant traffic, was rated as being less pleasant than natural sounds. The excitement of the particular soundscape was largely independent of the perceived pleasantness. Other research supports the claim of nature being closely associated with pleasant sounds (Marry, 2010).

An alternative scheme of soundscape classification is a sound by sound approach, with the physical characteristics of each individual sound noted (Schafer, 1994). Other classifications define the soundscape in terms of the origin of each component sound, but this method can produce an enormous number of categories (Schafer, 1994). Experienced individuals may classify sounds by aesthetic qualities, but this is one of the most difficult methods of classification, and cultural differences drive vastly different perceptions of certain sounds or entire soundscapes (Schafer, 1994).

Certain models of soundscape definition forgo the idea of an objective soundscape, understanding individual perception to be a vital feature of the environment. The World Soundscape Project, an international effort to classify soundscapes and improve sound design, uses a model which evaluates soundscapes from an ecological perspective, using interdisciplinary and subjective tools (Paquette, 2004). This model, however, struggles significantly with the evaluation of technological sounds within the soundscape (Paquette, 2004).

The model of soundscapes put forth by Traux, a variation of the model used by the World Soundscape Project, envisions the listener as part of the system, in a continuous interaction with his environment which is mediated by sounds (Paquette, 2004). The approach of CRESSON, (the French Research Centre on Sonic Space and the Urban Environment), referenced by Paquette, forgoes the narrow definition of sound object and the broad definition of soundscape in favor of the intermediate sound effect, a "particular sound perception as a result of specific physical conditions" (Paquette, 2004, 10). Certain models of sonic perception seek to preserve the information of orientation of the sounds with respect to the listener, through either description of the location of the sound or a visual aid such as a sonic mind map (Schafer, 1994; Paquette, 2004).

Soundscapes are only sometimes consciously noticed. In his study of sonic ambience, Marry observes that visual information is generally prioritized over sonic information by the mind (Marry, 2010). Some research indicated that a more open visual field can increase the positive perception of a soundscape (Marry, 2010).

2.4.3 Urban Soundscapes

Urban soundscapes may be classified by its "overpopulation of sounds" (Schafer, 1994). Distinct features of the urban soundscape include the low fidelity of its sonic information, the prevalence of technological noises, prolonged and constant high volume sound objects, and the generation of intentional noise as a mark of power (Schafer, 1994).

In his article *Discreet Mapping of Urban Soundscapes*, Balaÿ notes some of the difficulties inherent in defining which parts of an urban soundscape may be considered noise, any unwanted sound. The chief problem is that any sound may be subjectively considered positive or negative, depending on the qualities and mindset of the listener (Balaÿ, 2004). This research was performed in the wake of Directive 2002/49/EC of the European Union, which required member states to develop a noise map of their major cities in order to try to reduce levels of noise pollution (Council Directive 2002/49/EC of 25 June 2002 relating to the assessment and management of environmental noise). While not always even consciously noted, environmental noise has many effects on the listener, and awareness of these effects is a primary goal of many educators (Schafer, 1994).

2.4.4 Previous Implementation of Teaching about Soundscapes

Michael Cumberland, a middle school teacher of music, describes a method of teaching awareness of soundscapes, using modified activities from R. Murray Schafer's *A Sound Education*. The activity that he describes has students remain quiet and listen for one minute, noting the type and time of every sound that they hear. After, the classroom develops a "Soundscape Composition," a graph which displays a representation of the soundscape, but which sacrifices directional information about the sounds (Cumberland, 2001).

A simple previous implementation of an interactive soundscape is the Soundscape Constructor, which was created as part of the online resources of the Exploratorium, a science museum in San Francisco, CA. The online resource allows users to turn on and off typical sounds which one would experience in a number of locations (Soundscape Constructor).

One unique previous implementation of a soundscape exhibit, created by Hartwig Hochmair, characterized the city of Vienna by allowing users to take a virtual walk along predetermined paths through the city. As the exhibit virtually transitioned from one area to another, the sights and sounds surrounding the user would change to match recordings which had been taken in that area and which characterize that part of the city (Hochmair, 2004).

A tool allowing users to generate their own soundscapes, created by Zach Poff and NB Aldrich, is the Interactive Soundscapes project (sic), "an opportunity for communities to develop a profile, or self-portrait, in audio recordings" (Poff & Aldrich).They created a piece of software called the Interactive Soundscape Designer which invites users to recreate a local soundscape using a graphical interface (Figure 1).

2.5 Research Conclusions

The role of the museum, the scientific understanding of soundscapes, and the best methods of education are all points on which different writers disagree and are all topics which are in flux. Our goal as designers was to create an exhibit which satisfies the greatest reasonable consensus among the literature, while also being informed by the expectations and experience of the knowledgeable staff at the EcoTarium. We were also aware of the fact that our understanding of the realities of the situation was sure, at that point, to be incomplete. There is rarely a substitute for thorough testing of an engineering design.

3. Methods

After initial research, we began to form the methods a design a plan for the creation and testing of our prototype. Our goal was to take our findings and develop them alongside the EcoTarium in order to make a prototype exhibit that not only met the design specifications laid out by the museum, but also successfully engaged museum visitors on a level where they effectively learned from their experience. The execution of this process was not trivial, and required many changes from our initial process and expectations. Most notably, despite our research and discussions with the museum, there existed a significant disconnect between our early expectations on the design and testing process, and the actual environment in which we worked. As such, as we developed a prototype, our implementation of it and how we tested it changed just as much and just as often as the prototype itself. Ultimately, this made for a more effective development cycle.

Our initial intent was to investigate the process by which we would be testing the exhibit and the standards we would hold it to, all based off of our research. Parallel to this, we investigated the various options for the programming environment and language that we would need to create the program. Once settled, we set out to test the early versions of our prototype. From early October to late February, we developed and iterated our design and testing methods.

3.1 Exhibit Evaluation Criteria

We developed a set of design criteria based on our initial research and in consultation with the EcoTarium staff. In order to guide the initial design process and testing, we focused on adapting our research and the experiences of the EcoTarium staff to create a testing environment indicative of our evaluation needs.

The goal of the design criteria was to not only generate a prototype of an interactive exhibit that allows young users to become engaged in the process of recreating their own personal soundscape, but also to create a family friendly environment. The prototype was based on a set of design criteria, which in turn were based on both conversations with the EcoTarium staff and a review of the pertinent literature and design characteristics. Our objectives are largely modified from the goals of the PISEC (Philadelphia-Camden Informal Science Education

Collaborative) and APE (Active Prolonged Engagement) evaluation schemata with the end goal of educating users on city/neighborhood soundscapes.

The EcoTarium provided us with several of the criteria which they utilize in their observations and evaluations for visitor studies. Their methods emphasize family learning, open-ended investigation, and goals of the visitors, while taking into account additional variables such as the gender of the children using the exhibit and the role played by the parent. Family learning observation examines the interactions between parents or guardians and children. Open-ended observation, one of our evaluation tools, focuses on the behaviors of the subjects while interacting in any meaningful way with our exhibit.

3.1.1 Characteristics of Family Learning and Open-Ended Investigation

The EcoTarium staff identified several schemata that represent the ‘cutting edge’ in exhibit design and evaluation. The soundscape exhibit will incorporate the best elements of multiple schemata. PISEC is an evaluation process that utilizes the three learning levels of identification, description, and interpretation/application, to score families at test exhibits and, ultimately, describe the effectiveness of the exhibit, and was also used initially in the design of the exhibit (Borun et al, 1998). Based on earlier observations and findings, PISEC created a list describing the seven characteristics of family-friendly exhibits:

- Multi-sided: a family group can cluster around the exhibit.
- Multi-user: the interaction allows for several sets of hands.
- Accessible: the exhibit can comfortably be used by children and adults.
- Multi-outcome: the observation and interaction are sufficiently complex to foster group discussion.
- Multi-modal: the exhibit appeals to different learning styles and levels of knowledge.
- Readable: the text is arranged in easily-understood segments.
- Relevant: the content provides cognitive links to visitors’ existing knowledge and experience.
  (Borun et al, 1998, 1)

We designed our prototype with the goal to address each of these key characteristics. Our surveys will then be able to determine successful and unsuccessful features in greater detail.

APE is a set of goals which focuses on keeping the visitor actively engaged at a given exhibit while remaining consistently engaged and entertained (Humphrey and Gutwill, 2005). One study identified several factors that were noticed most commonly at exhibits that were considered to have a "high capacity for APE engagement" (Tisdal, 2004, *Active*). These factors include:

1. Exploration of exhibit phenomena
2. Visitors asked to think scientifically
3. Immediate physical engagement by visitors of all ages
4. Providing directions for exhibit
5. Engagement at multiple levels of knowledge
6. Rewarding new ideas and behavior
7. Participation is free from external influence
8. Allowing for group work
9. Multiple stations to prevent interference
10. Tasks that require optimal time range for engagement
11. Providing resources for extended engagement
    (Tisdal, 2004, *Active*)

Studies also examined how visitors utilized APE exhibits in their own scientific investigations, questions, and hypotheses. The interactions were categorized into four different types of engagement: physical, intellectual, social and emotional.

Physical engagement is defined as the different ways in which visitors physically interact with an exhibit. This includes the amount of time they spend, the labels they read, where they sit or stand, and what buttons they push. It also includes the sequence of activities in which they participate. By analyzing physical engagement, we can understand whether an interaction was primarily guided by the exhibit design or if visitors engaged in self-directed exploration (Tisdal, 2004, *Active*).

Intellectual engagement is defined as the various ways in which visitors engage with the exhibit on a mental level. Intellectual engagement is often referred to as "minds-on" in contrast to hands-on. It includes the connections visitors make to existing knowledge during their interaction, conceptual understandings, and the questions visitors have. One aspect of intellectual engagement that was noticed, particularly in young children, was the awareness of learning experiences encountered at both APE and non-APE exhibits and how it affected the decision to use the exhibit (Tisdal, 2004, *Active*).

Social engagement is defined by how visitors influence each other's experiences at exhibits. It includes conversations that might guide what an individual does or understands during the interaction. Social engagement also includes directions, observation, guidance, assistance, cooperation, and competition among visitors using an exhibit at the same time, as well as deliberate teaching/learning behaviors, such as a parent asking a child a question to get the youngster engaged in the exhibit, or one person explaining something to another visitor. Other members of a social group can also impact the respondent, regardless of where they are during the engagement (Tisdal, 2004, *Active*).

Emotional engagement involves both the nature and intensity of the effect exhibited by visitors during the engagement and immediately after. The nature of the emotional engagement may be positive (fun, awe, pleasure, enjoyment, caring) or negative (embarrassment, confusion, disdain, humiliation). The effect of the exhibit on the respondent can indicate the things that individuals value over others (Tisdal, 2004, *Active*).

3.1.2 Implementation of User Learning Objectives

Considering our goal of educating museum visitors, we compiled our research on the subject and determined that our design objective would concentrate on learning. Rather than designing our prototype around teaching, we instead decided to design it around exploration. In this way, through an engaging environment, users would learn based on voluntary interest and discovery, rather than through more formal methods of direct teaching. This design principle was adopted in an effort to keep users interested longer and draw their attention away from the fact that they are in an educational environment, and instead focus them on delving deeper into the more creative functions of the exhibit. By keeping users engaged longer and giving them the opportunity to explore, visitors would hopefully discover the themes and concepts for themselves, coming away from the exhibit with an understanding and appreciation of city soundscapes.

3.2 Software Selection

The EcoTarium staff provided a clear set of objectives and goals for the progressive set of prototypes that were developed. Their main interest was the development and testing of the software that would be incorporated into the final exhibit. They asked that all prototyping be done from a laptop and specified that the physical exhibit would be created separately from our application. This meant that our prototypes would not be tested on a touch screen, and the bulk of the physical exhibit would be designed after the software has been created. The software was required to be something that could be tested on a mobile object like a laptop for the purposes of evaluation. The final product developed, based on the culmination of all our prototypes and evaluations would need to include the functionality for a user to dynamically select, play, and mix sounds. These basic requirements required the software to overlay and mix sounds in real time, a non-trivial stipulation.

3.2.1 Software Comparison

In determining a software choice, we needed to evaluate the pros and cons of using a commercial product versus developing or using a set of programming libraries. Products like Audacity and GarageBand provide astounding music editing and mixing capabilities. However, these, like the majority of other large music editing software packages, were not designed to produce a derivative application. Both include a large set of functions, but do not possess the precise functionality that our prototype required. In order to make something like Audacity or GarageBand viable, the simplistic functionality would have required extensive additional scripting (Audacity, 2012). This process would have been far more difficult than writing the software from scratch. In addition to this obstacle, neither program has an intuitive graphical user interface (GUI). We believed that users would have significant trouble interacting without spending significant time trying to interpret the GUI (Audacity, 2012). In order to eliminate this learning curve, a separate interface would have had to be designed, interacting with the software behind the scenes. As the difficulties of adapting existing software in a user friendly way are enormous, we instead designed and implemented our own software, which supports all of our required functionality.

There are several well-known programming languages that provide significant resources for building software that manipulates sound. Among these viable options, the notable ones include C++, Java, Objective-C, and html. Objective-C was immediately ruled out, despite its significant support for synthesizing sound, because it is primarily used for designing Mac Software. An html web page may have been viable, but would not have enabled us to test in all possible situations, as it continuously requires a stable internet connection. We would have had to accept the additional risks of the host server going down or internet being for any reason being interrupted. This would have imposed significant additional requirements for the museum exhibit itself. When examining Java and C++, it was determined that both languages provide nearly identical capabilities with respect to the manipulation and playing of sound; however, Java provides far simpler GUI building support (wxWidgets, 2012). The deciding factors in favor of Java were substantial existing functionality for playing, overlaying and mixing sound in real time, the customizability that these libraries provide, and the built in GUI creation templates. A Java application was also appealing because, once designed, it requires no maintenance (The Eclipse Foundation, 2012). The EcoTarium staff would not necessarily require a programmer to maintain or fix issues with the application. If the application breaks or for some reason becomes corrupted, it may simply be reinstalled.

3.2.2 Java as a Design Choice

All prototypes generated were Java applications. In order to have a platform where we could quickly solicit user feedback on sound selection and associated imagery, the first prototype had only the functionality for a user to select a set number of sounds to play. Subsequent iterations began to incorporate the other desired capabilities. Touch screen capability, while central to the final user interaction, was considered a final step. It was not initially considered necessary for testing purposes, and a much simpler mouse interface provided the same basic functionality. Our priority was to create an engaging and responsive user interface that would be intuitive to as many age groups as possible.

3.2.3 Java Capabilities and Methodology

The Eclipse WindowBuilder, a drag and drop Java Graphical User Interface (GUI) designer, made a Java application a very attractive choice. This plugin possesses the design and layout tools necessary to produce a layout without actually writing substantial code. This made the process of designing a GUI relatively fast. This capability proved invaluable; as the GUI regularly changed in response to users provide regular feedback through our evaluation process.

Java libraries also provide a set of Application Programing Interfaces (APIs) that simplify the set of code required to overlay sound in real time. The Java TM 2 Platform javax.sound package provides a set of interfaces and classes that capture, process and playback sampled audio data. A class is a component of a program that defines objects, in our case sounds, and the operations that that can be performed on those objects (Appendix H). This sampled data can be in the form of WAV files, the format that we will be using for all our sound samples. This powerful API also includes Musical Instrument Digital Interface (MIDI) data capabilities (The Eclipse Foundation, 2012). MIDI is an electronic musical specification that allows computers to communicate with one another and its' set of commands guarantees compatibility between them. Though we do not require this functionality, the efficient sound engine that supports the API produces high quality audio mixing and MIDI synthesis capabilities.

The Java TM 2 Platform API also includes a class called Thread. A thread is a single sequential flow of control within a program. Using a thread enables programmers to execute multiple tasks at the same time. In the context of our prototype designs, this enabled us to play multiple sounds simultaneously without overlaying or mixing them in real time, a task that would otherwise have required the development of an extensive algorithm which merged multiple byte arrays. By extending the Java Thread class in a separate class that holds a single WAV file and then invoking the object's run() method on multiple instances of this class, we were able to play all user selected sounds simultaneously as if we were playing a single sound file. This is because when a thread is created it is permanently bound to an instance of itself with a run() method. Instantiating this object creates both an object and a thread capable of executing the sequential flow outlined by the run() method of the object (The Eclipse Foundation, 2012). Our prototypes utilized the aforementioned Java functionality. While we anticipated that revisions may have needed to incorporate other software, the Java TM 2 Platform allowed us to rapidly create an application with all required functionality. In turn, this enabled us to spend more time on the frequent user evaluation stages and ultimately produce an exhibit to further user learning and is better tailored to their interests.

3.3 Pretesting and Formative Evaluation of Prototypes at the EcoTarium

Various levels of testing and prototyping were used in order to progressively improve upon not only our prototype, but also our methods of evaluation themselves. Our testing began with a more informal prototype review process, but eventually moved towards floor testing at the EcoTarium, where we evaluated our progress and made a number of substantial modifications (Appendix G).

3.3.1 Prototype Review Process

The role of the initial prototype review was intended to determine and rectify some of the problems that would inevitably come up during our initial testing at the EcoTarium. The two areas that we felt could be evaluated before our testing at the museum were the regcognizability of our sound clips and the regcognizability of our images. Our early plans for this review phase were, in brief, a combination of a convenience and a volunteer sample, utilizing both volunteers and targeted individuals who may have specific expertise in a given area, particularly the EcoTarium staff (Appendix F).

We, however, perceived many potential problems with this phase of testing. Ultimately these led us to drop this phase and move straight onto floor testing at the EcoTarium. These included:

- Our survey subjects in this phase would be largely college students, rather than young children;
- Our results would be biased by our friend groups and those who are willing to take the survey;
- There was the potential for this test group to have no intuitive understanding of images, sounds, or questions that younger children have little trouble understanding, and vice versa; and
- We did not yet have a comprehensive understanding of our own image and sound requirements.

The chief advantage and goal of this pretesting phase, however, was that it would theoretically make our first rounds of testing more informative. We feared that without this phase, the first several iterations of our prototype would have largely concerned making our interface more intuitive, rather than having been able to address the issue of whether the interface was successful in meeting its desired metrics.

After consulting with the EcoTarium staff, however, and listening to their experiences concerning testing new exhibits, we came to the conclusion that skipping an initial phase and proceeding directly to floor testing was, in actuality, a more desirable option. Even if we had an incomplete and poorly functioning prototype (Prototype 1), in the EcoTarium staff's experience, testing that in the real environment generally yielded better results than potentially wasting time on features that did not reflect the feedback of users. As a result, we brought our initial Prototype 1 to test at the museum, which ultimately yielded valuable results that impacted the final version of our prototype (see Section 4.2.1 for details).

3.3.2 Participant Selection in Formative Evaluation

We planned on having participant selection be an integral part of ensuring our prototype and our testing methods were properly representative. Our initial goal was to create a general visitor profile that would drive us towards creating an exhibit that appealed to all users. Like our original testing plan, this intent changed substantially once we were faced with the reality of the museum environment.

For generalized participant selection, we planned to take every effort to ensure a maximum number of visitors interact with the exhibit. By creating a large testing pool, we had hoped to ensure a more random and generalized sample of the EcoTarium visitor profile, thus lessening any potential effects of selection bias on our part. On light days at the museum, this generally would entail actively seeking out visitors and inviting them to the exhibit; on busy days it was more prudent to allow visitors to actively choose to try the exhibit and only invite visitors if necessary. Regardless of the museum crowding, we would have attempted to create as diverse a visitor profile as possible that included selecting participants by demographic.

However, we were quickly informed by the EcoTarium staff and discovered for ourselves that exhibit testing rarely requires large sample sizes. While on some specific days we received enormous amounts of data, days during which we received few test results proved almost as informative. With each of our prototypes, there were generally a small number of critical stumbling blocks that almost all users came across. Throughout our process of testing, our focus shifted from obtaining a large number of samples to obtaining a smaller sample filled with more qualitative information.

One key interest we had was the feedback of specific demographics (particularly age, sex, and cultural background). We still felt that it was important to survey as many demographics as possible, such as age or culture, in order to create a broadly appealing and engaging exhibit. While it is difficult or impossible to achieve universal appeal, only by surveying as many demographics as possible could we begin to create an exhibit that addresses the nuances of various cultures and ages.

We found, however, that testing according to culture was effectively impossible. On any given day of testing, representatives from a given culture will almost certainly be absent, and even if present it is both impossible to identify a cultural group based on visual cues and highly impolite in many cases to inquire. Age, however, did turn out to be a variable that we were able to roughly judge and consider in our iterations. Age became our most important indicator of the intuitiveness of our prototype. The goal was to create a testing environment that appealed to children familiar with touch screen interfaces, inquisitive teenagers, and older patrons, all at the same time. Daunting as it was, by documenting their interactions we were able to draw many connections between the age demographics that were fairly common and ultimately allowed for a more universally acceptable prototype.

3.3.3 Obtaining Consent for Observation and Testing

Prior to any participant testing or using any prototype, consent was obtained from a user or, as necessary, the parent of a user, to allow for both observation and follow up questioning. Prior to a visitor approaching our exhibit, they passed signs indicating that they would be interacting in an observed evaluation environment, that participation is entirely voluntary, that they may end their interaction at any point during the process, that personally identifying information is not being recorded, and that they are not obliged to answer every question in the evaluation. This allowed visitors to refrain from interaction with the exhibit should they choose. Once a user had chosen to use our exhibit prototype, consent was required and gathered in the form of a simple verbal agreement asked before the participant interacted with the prototype or were asked any other questions. No names or other personally identifiable information were collected at any time, as it was not needed for the benefit of the exhibit evaluation. Participants were informed that the scope of the evaluation would include both observation and interview questions. While during early stages of testing these were highly segregated periods of our interaction with users, in later testing the two phases became less distinct.

If the evaluation subject was either a child or a group that contains children, consent was received from the parents or other adult in charge, rather than the child. Children were allowed and encouraged to participate in the exhibit, as they are the primary demographic and target audience of the EcoTarium. Children were also questioned along with their parents about their experience, but the parents or guardians were of course allowed to remove a child from the interview or observation at their own discretion.

We never encountered an individual or parent who denied consent to be monitored or asked questions. However, had consent to observation and interview not been granted, the participant would not have been questioned or recorded in any way. They would have been allowed to use the exhibit until the arrival of a consenting group or individual.

3.3.4 Observation Criteria, Aids, and Procedure

Evaluation of visitors in the exhibit fell into two general categories: passive observation and active interview. This allowed for us to observe both the users' real time interaction with the exhibit and the after action thoughts of the visitors.

For passive observation, participants' actions were initially recorded without input by us in order to determine user patterns. However, if the user asked for help or we felt that providing aid would give us more useful information on the prototype, prompts were sometimes given to guide users, particularly younger ones. Evaluation was guided by an evaluation sheet which categorized user actions and reactions while using the exhibit. This allowed for standardized categorization of user feedback that we could then synthesize into a general user profile (see Appendix C). Initial iterations of our prototype observations revolved primarily around improving our methods, learning what behaviors to look for, and determining the best way to record data. On successive prototypes, this developed into a less formalized but more problem oriented method, leading to more informative data, which ultimately resulted in a more effective exhibit. Observation was the primary method of initial data collection, with interview style questioning developing later as a direct result of questions formed from common (or uncommon) observations.

When questions were integrated during the period of user interaction, they took the form of both an after action interview and questions asked while the user was interacting with the exhibit. Questions asked after the fact primarily involved users' personal opinion of the exhibit and an attempt to determine if the visitors effectively learned about soundscapes (Appendix C). This oral feedback also attempted to determine the success of the digital technology implementation. Only by asking these questions were we able to determine if our prototype exhibit was an effective learning tool. While some questions were reserved until the end of user interaction with the exhibit, some questions were also asked of users as they interacted with the exhibit. These questions targeted specific user actions and filled in the knowledge gaps that pure observation could not. Active intermediary questions evaluated misunderstandings and miscommunications between the user and the exhibit. By evaluating common user difficulties and challenges, we were better able to improve our prototype, as in the final version where no museum staff will be present to actively guide visitors in how to use and interact with the exhibit.

3.4 Repeated and Revised Evaluation and Improved Prototypes

Once feedback had been gathered after each iteration of testing, we consulted with museum staff, decided what modifications might be appropriate, and updated our prototype accordingly (see section 4.2). The goal was to incrementally improve upon our prototype in light of the design goals and requirements of the EcoTarium. Initial revisions were focused on user comprehension of the exhibit interface and general usability. Once these aspects of the interface had been improved upon, the intent was for our focus to shift towards more intricate parts of the exhibit that were not necessarily as apparent to the user. This involved updating and increasing the number of sounds used, as well as modifying the inner workings of the program to add more functionality.

As our prototype evolved in accordance with user feedback, so too did our testing methods. With each testing phase, we revised our questioning techniques, our questionnaire, and once our entire approach in order to address shortcomings in our prototype evaluation and adapt our approach to the available audience at each phase. We eliminated unnecessary questions and added any that were needed to fill gaps in our research and evaluation (Appendix D). Ultimately, our evaluation methods developed along with our prototype, allowing for each to benefit from the other's improvement.

3.4.1 Evaluation of User Interaction with Technology

It was imperative that users be able to interact with the exhibit. If there existed a technological barrier, then users would never be able to gain an appropriate understanding of soundscapes. The focus of technological improvements was largely in revisions to the GUI, with the exception of a major overhaul of the back-end code later in the iterative process. By iterating the GUI, we arrived at a functioning display that facilitated efficient, intuitive, and fast interaction with the exhibit. User difficulties in interacting with the device from our testing were synthesized and used to revise images, text, and sounds to be more understandable and facilitating of a learning environment.

3.4.2 Evaluation of User Learning Achievements

With one of our major design goals being the education of visitors, it is imperative that users leave the exhibit better educated in soundscapes. If they did not, the exhibit was generally considered as having failed, regardless of how long a visitor stayed at the exhibit or how engaged they were. Part of the interview questionnaire would address this question, and how testers responded would determine how we revised or reinforced our prototype. In reality, however, there was rarely a clear distinction between difficulty with interaction and difficulty with learning.

The barrier to learning was often the technology itself, driving many of our changes. General usability problems directly hindered user education. This was substantially improved over the course of the several prototypes. As our prototype developed in functionality, the shortcomings in user education became clearer. As a result, we refrained from directly addressing the issue of education until we had a clearer idea of what specific areas needed to be addressed.

In order to determine the extent of user education, we tailored a portion of our evaluation survey to gauge what users took away from their experience. Analyzing what they learned (or more often did not learn) afforded us the opportunity to further develop the prototype in order to better address the issues and help facilitate better education in the future.

Towards the end of the prototype iterations, we concluded that the program alone would not lead to adequate learning. Accordingly, we designed an "Education Panel," which included supplemental educational information as well as question prompts in order to guide user exploration and promote learning.

3.5 Prototype Evaluation Changes

This section details the substantial changes to our methodology that occurred with each prototype iteration.

3.5.1 Prototype 1

In our first iteration, we immediately found areas in which our initial methodology was lacking. With younger users in particular, the technology was not intuitive (one test subject had never used a mouse before, for example), requiring us to in cases explicitly guide the users through the soundscape process.

3.5.2 Prototype 2

In the second iteration, we revised our data collection sheets to reflect what we determined to be either redundant or trivial information from the first iteration. We also made the decision that from this iteration onwards, only two of our group members should be present for testing at any time. We replaced using embedded laptop speakers with external, powered speakers. We became far less selective about the ages and demographics of our users. Finally, we modified our signage to appear more professional and less intimidating (Appendix I).

3.5.3 Prototype 3

No significant changes to our methodology were implemented.

3.5.4 Prototype 4

With iteration four, the need for “prompting on use” of the interface significantly decreased due to the implementation of drag and drop functionality.

3.5.5 Prototype 5

With iteration five, we began asking directly for feedback on which sounds users wished that they had access to, rather than only recording things that users said indirectly.

3.5.6 Prototype 6

Prototype 6 was a review by the EcoTarium staff rather than by museum visitors. This allowed us to touch base with their initial expectations and design goals, and provide feedback as to the direction of the prototype.

3.5.7 Floor Model (Prototype 7)

The seventh and final iteration of our prototype is the compilation of all user feedback, EcoTarium staff input, and our own ideas. It represents the most user friendly and technically functional edition. This model was also put on a touch screen as per the initial design expectations laid out by the EcoTarium staff. Included in this model was the Education Panel, which stood as a supplement to the interface.

3.6 Outcomes of our Methodology

Through use of our methodology, we completed seven rounds of iterative testing. From this testing, we were able to gather sufficient feedback in order to make informed changes to our exhibit. This allowed us to better fulfill the EcoTarium's expectations with regards to the interface and to meet our own evaluation metrics.

4. Design Findings and Analysis

4.1 General Prototyping Process

As described in the previous section, we conducted a series of prototype evaluations (formative evaluation) between October 2012 and February 2013 to generate and refine our prototype design in an iterative fashion. Knowing that our evaluation approach would change with experience and that the implementation of our formative evaluation would evolve, we planned on adapting the process itself along with the testing of the actual prototype. Once testing began, we substantially changed our procedures so we could efficiently gather more useful data. This evolution allowed us to better develop our prototype based on more targeted feedback from testers as the prototyping progressed.

Data were collected from a variety of sources, but mainly from EcoTarium visitors. EcoTarium staff also provided substantial information concerning their expectations for the prototype, their vision of the larger purpose of the exhibition, and experiences with exhibit development; they also provided direct feedback after interaction with the prototype. Significant testing was also conducted within our team to ensure that the prototype functioned smoothly, and to ensure that we were able to guide visitors appropriately.

Information was gathered from each iteration of our testing, which we assessed to find trends and patterns in the user experience. We paid particular attention to the unique problems encountered by specific users or demographics so that we could design the prototype to be accessible and appealing to as many visitors as possible. Common problems that arose generally revolved around the absence of features that we planned to incorporate, but had not yet executed due to time constraints. While this led to some frustration on our own part and on the part of our users, our data collection and focus on unique problems was ultimately manageable and resulted in significant new design changes.

4.2 Prototype Iteration and Evaluation

In total, our prototype went through seven substantial revisions, culminating with the floor model in late February. Each iteration was modified based on user interaction and feedback, as well the gradual implementation of our planned features. Based on feedback from the EcoTarium and our own discussion, we decided that it was better to test each iteration on the floor rather than waiting until we had a fully refined version. This allowed us to receive immediate feedback that often raised expected concerns, but also provided us with information that allowed us to fix additional problems early in the development cycle. In the sections that follow we discuss each of the prototypes in turn as they evolved. For each prototype, we present:

- a basic description of the main design elements and how they changed from the previous version;
- a discussion of the problems visitors encountered in using the prototype that indicated necessary design changes for the next prototype; and,
- a discussion of the problems we encountered in conducting the evaluations that indicated how our evaluation protocols and procedures should be modified.

A more detailed discussion of the design elements of each prototype is presented in Appendix J.

4.2.1 Proof of Concept (Prototype 1)

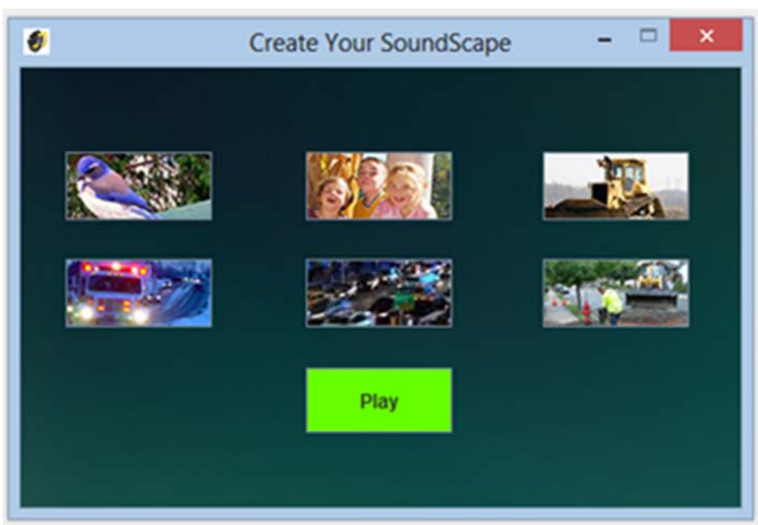


Figure 1- The interface of Prototype 1

The first iteration functioned as a simple proof of concept. The purpose of this version was to evaluate both for ourselves and for the EcoTarium that designing the exhibit would be possible on the Java platform. While not fully functional in any regard, the simple design allowed us to test basic sound selection (via button) and sound playback in an overlapping format, as these were the most fundamental functions that the exhibit would need to perform. If it proved too difficult or time consuming, we would need to attempt the design in another program or environment. Little effort was put into the layout or design, as this version was never planned for public display.

4.2.1.1 Prototype Findings

Despite the crude format of this initial interface, it yielded important information on how to proceed. First, it proved that the overlay of dynamically selected sounds was possible and achievable within the timeframe allotted for our project. Second, the program provided us a simple environment, within which we could test potential sounds that might be added to our sound library. The prototype's simplicity allowed us to mix test sounds clips and evaluate if they would adequately overlay while remaining largely identifiable. The code for this prototype also served as the basic architecture for the code of our next several interfaces.

4.4.1.2 Evaluation Process Findings

While we knew that the interface would achieve few or none of the goals which we hoped our final exhibit would fulfill, the testing in this phase served as an evaluation of our testing procedures as much as it was an evaluation of the prototype. We learned that there was a great deal of redundant or irrelevant information on our original testing sheets, driving the need to create new design metrics. We also learned that we needed to provide far louder speakers than were available on a laptop. The EcoTarium testing floor is an extremely acoustically bright environment, and the noise forced us into a dark but quiet corner of the museum floor, where few visitors congregated and where we and our prototype testing station seemed much less approachable (see Appendix K). This made recruiting visitors to test the prototype more difficult. We also realized that our information sheet informing users of our ethical testing practices could be made to look far more inviting than black, intimidating text on a white background.

4.2.2 Prototype 2

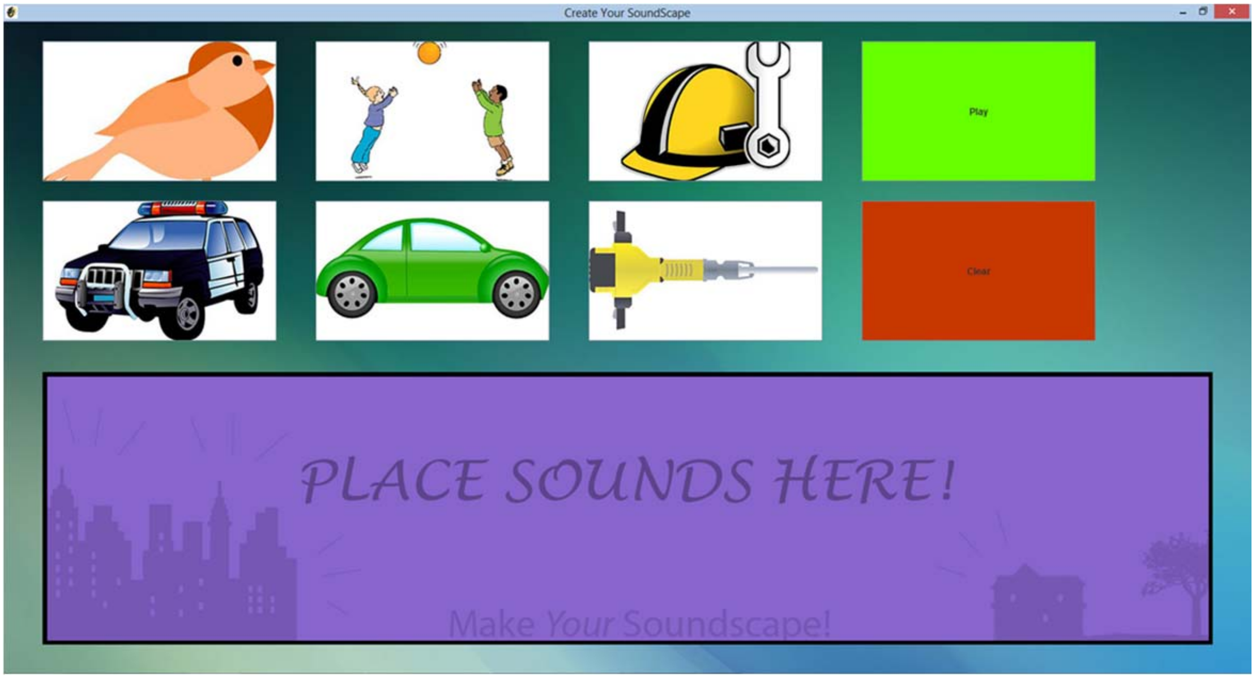


Figure 2- The interface of Prototype 2

While Prototype 2 was ultimately very simple, it laid the groundwork for future versions. This version included just basic functionality, not too far advanced from the proof of concept design. Also included was the first attempt at creating an aesthetically pleasing visual environment with which the user would interact and play sounds.

The prototype was built around a simple layout: buttons, consisting of images represented the sound they would play. The buttons were laid out in rows on the top of the screen, and a large block (the “canvas”) at the bottom, which was the field in which sounds could be placed.

The user interacted with the exhibit by clicking on the button they wanted, and then clicked on the canvas where they wanted to place the sound. This would make a thumbnail image appear. The horizontal-axis of the canvas served as a timeline for sound playback, such that the sounds farther to the left would begin playing before those to the right. This allowed the vertical dimension to be used for potential overlapping of sounds. The thumbnails were different images than the buttons in an attempt to add variety.

In order to play the sounds, the user would press the large play button, which would cause the program to run through the canvas and play the placed sounds. The soundscape would play once, and then stop; each user had to successively hit play in order to replay. A large “clear all” button was also present, which gave us the ability to remove all thumbnails from the image simultaneously. No functionality existed to individually remove sounds.

4.2.2.1 Prototype Findings

Ultimately, this was one of our most informative testing experiences. It immediately became clear that there were not nearly enough visual clues to indicate which tasks the user was supposed to perform. No user we tested had any intuition as to how to place sounds, or even that placing sounds was the intent of the exhibit. As a result, we had to prompt users directly in order for them to understand both what to do and why they were doing it. Many of the testers found that the system of both clicking the button and then clicking the canvas to place the sound was unintuitive and clumsy. This was an expected but unfortunate side effect of us planning on adopting drag and drop functionality later in the design process, due to the complexity of implementation. Our first tester, a 5 year old girl with her mother, immediately upon sitting in front of our prototype attempted to physically touch the laptop monitor, assuming it was a touch screen. This was a clear indication of the technological disparity between age groups; younger users are more inclined to expect touch screen interfaces due to tablets and similar devices, while older individuals are generally more familiar with mouse-based interfaces. Even so, drag and drop functionality, regardless of input device was the most requested feature (until this functionality was introduced with Prototype 4). Nevertheless, once users were instructed on the mouse clicking procedure of Prototype 2, sound placement went relatively smoothly.

Users became frustrated with many of the visual cues, or lack thereof, in the prototype. They universally brought up the fact that placed thumbnails did not share the same image as the original button. This caused confusion as to what they were actually placing on the canvas. Additionally, the buttons were cartoon-like images, while the thumbnails were life-like images, causing visual dissonance for the users. Another visual shortcoming was the lack of visual feedback indicating that the thumbnail represented the sound coming out of the speaker, because there was no indication of what sound was currently playing on the canvas. If no thumbnail was currently being played, many users would become confused as to why no sounds were being played, despite them having pressed a button labeled “play”. The canvas remained static once the sounds were placed. Users were also frustrated that they could not delete a single sound once placed. Instead they had to clear the entire canvas, which was described as annoying if many sounds had already been placed.

We used six sounds in this prototype: birds, children playing, construction, sirens, traffic, and the sound of a jackhammer. The jackhammer was by far the most popular with children, due to its loud and distinct noise. In general, individuals did not have trouble distinguishing what the sounds were supposed to be based on the button images. The exception was "construction", which was the most abstract of the sounds.

4.2.2.2 Evaluation Process Findings

Immediately, we identified several shortcomings in our planned evaluation methods. The most obvious was the limited number of visitors available to serve as testers given the times during the week when we were able to conduct evaluations at the museum, and that we were testing during the 'off-season' for visitation. This forced us to change our expectation of being able to 'select' our testers. Instead, we accepted virtually any visitor who was willing to try our prototype. We also had to actively ask visitors to visit our table, as most visitors generally never gave us more than a passing glance unless we engaged them. Unlike Prototype 1, however, we had a significantly better testing location for this prototype (and all others thereafter). We were no longer in a removed corridor, but rather, more central to the main museum floor.

Once a user sat down, we found that we had to split up out team, with one member engaging the tester, explaining the functionality of the prototype, while the other recorded notes. This negatively impacted the evaluation of the users because our influence and input was necessary for them to understand what to do, as well as being somewhat intimidating to them.

Additionally, our designed surveys proved to be largely useless in their current form. Most of the questions were left unused, and empty space became the commodity as the recorder attempted to record offhand comments and suggestions from the users. We saw the need for a single page that included only our most common questions, organized in such a way that note taking flowed with the exploration of the exhibit.

4.2.3 Prototype 3

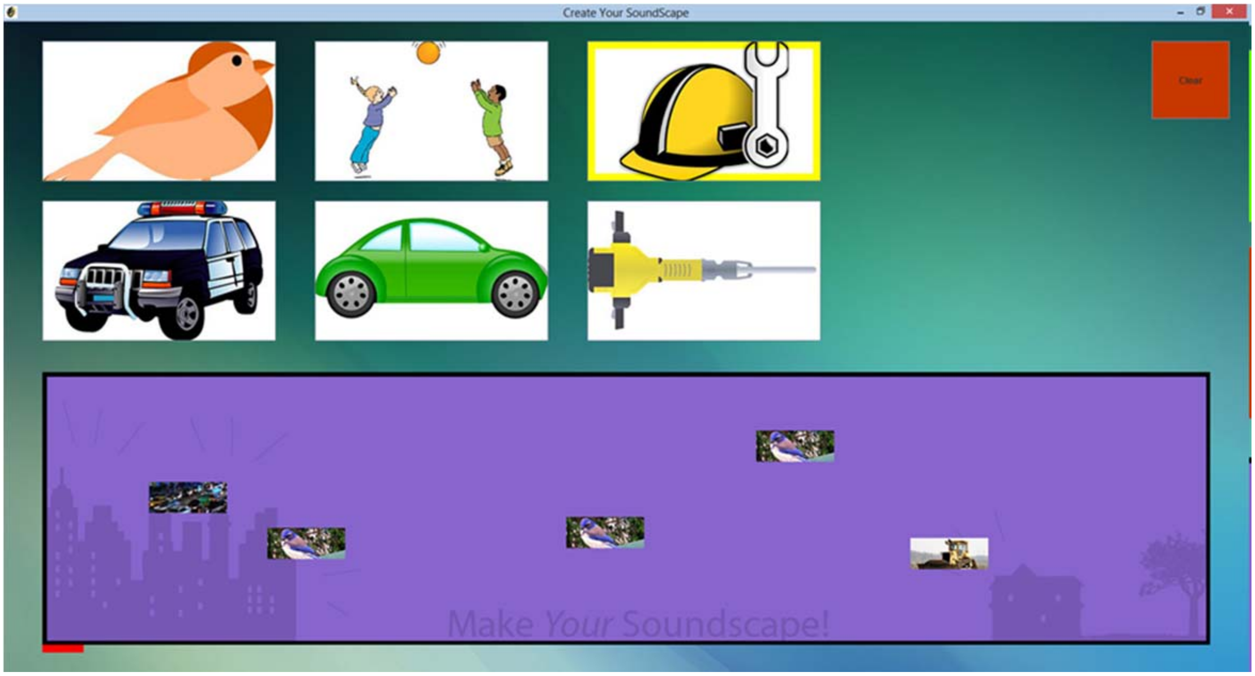


Figure 3- The interface of Prototype 3

Prototype 3 was in development before our first day of testing with Prototype 2, and was finished shortly thereafter, though not in time to test during Prototype 2's testing period. As a result, most of the feedback that we received from testing Prototype 2 did not have time to be implemented and applied to Prototype 3. The new features present in Prototype 3 reflect instead many of the planned features which had not yet been implemented into our earlier prototypes.

The largest change to Prototype 3 was the method by which sounds were played. The play button was removed and replaced instead by a continuously playing canvas. New sounds would automatically play when a timer, looping from left to right on the canvas, intersected with them. This loop was 40 seconds in length and would continue indefinitely. This new play style was visually facilitated by a new time-bar, which tracked along the bottom of the canvas. Reminiscent of any standard internet video player, this bar served as a visual indicator as to where on the canvas the sound loop currently was, allowing users to gauge where to place sounds and when sounds would play.

Another significant advance was the inclusion of the ability to delete single thumbnails, rather than being limited to a single clear-all button (which was also still present, however). Placement of sounds remained identical, but if a user wanted to delete a particular sound, all they had to do was click the placed sounds and it would be removed from the canvas. There were, however, no visual indicators or instructions about how to use this feature.

4.2.3.1 Prototype Findings

Despite Prototype 3 not having any major features resulting from Prototype 2's feedback due to a short turnaround time before our next testing block, the features we did add had a positive impact on user interaction and engagement. The largest impact was due to the inclusion of the scrolling time-bar. When provided with visual feedback, we found that users had a clearer understanding of how the prototype worked. Users generally associated the canvas with sounds playing and perceived it as being a dynamic changeable environment. Prompts were still needed to engage the tester and give them an initial understanding of both what the exhibit was supposed to do and how it functioned. However, users tended to understand the basic functionality far more quickly than in the previous prototypes.

The fact that sounds played continuously in a loop also had an impact on user reaction. Previously, when a play button had been present, the user would create a soundscape then press the button. Once pressed, the user would sit idly until it finished. Now, with continuous playback, users could hear a constant feedback of what they were making. This resulted in many users dynamically adapting the sound of their soundscape to make it a better representation of their neighborhoods. We found this new reaction to be a very positive change. Prototype 3 better facilitated constant and lasting engagement. When users had used Prototype 2, they had often only created one soundscape, listened to it, and then moved on to other exhibits, unless specifically prompted by us to edit their original creation. Now, users were more attentive to the exhibit and often updated their soundscape to get desired results.

4.2.3.2 Evaluation Process Findings

No significant updates were made to our methods while testing Prototype 3. Instead, we used our experience from testing Prototype 2 to improve our interaction with EcoTarium visitors. We tried to sound more professional and knowledgeable by being more confident with museum visitors due to our experience.

4.2.4 Prototype 4

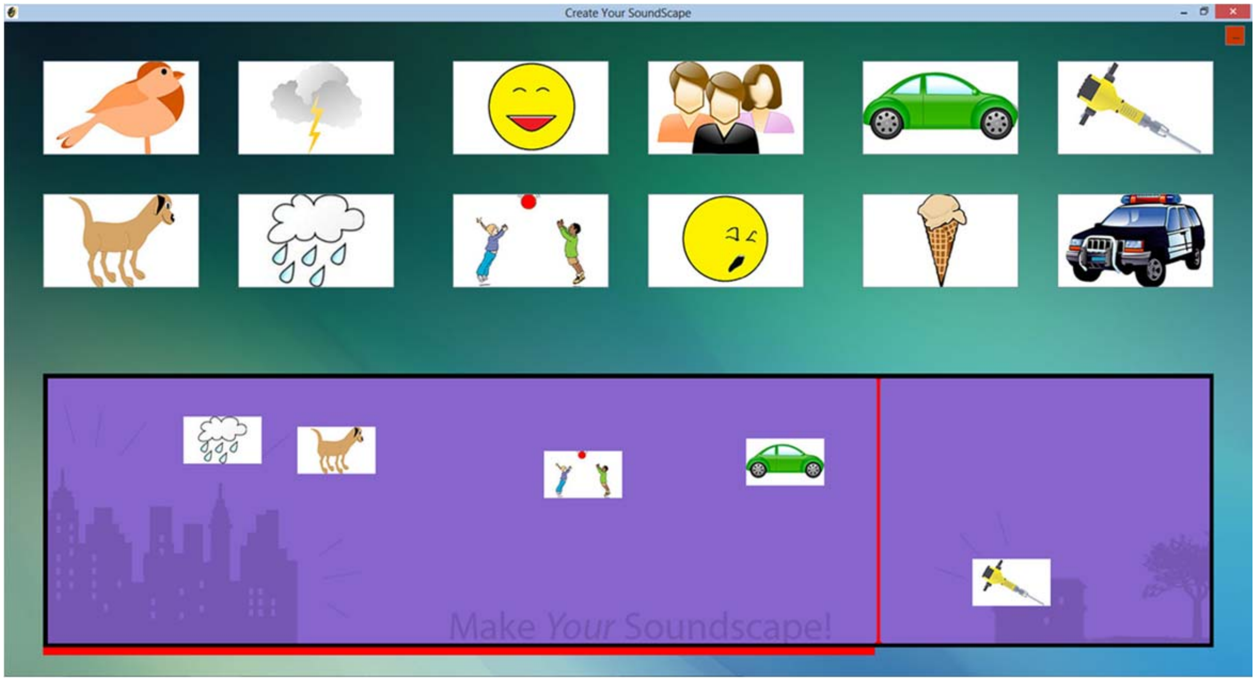


Figure 4- The interface of Prototype 4

Prototype 4 was our first opportunity to significantly implement changes to the prototype as a direct result of user feedback. This iteration implemented many of our planned features, as well as suggestions and needs from the testing of our three previous prototypes. The prototype continued its gradual advancement towards our design goals and was intended to be easier for users to comprehend and interact with. The main aspect we aimed to test was the new drag and drop feature. We also expanded the sound library and improved the layout of the screen.

Drag and drop was one of the most requested features during previous implementations, and one we knew we needed to implement if the exhibit was to function successfully on the museum floor. It was an extremely beneficial change. Drag and drop had always been the natural tendency for the vast majority of users, and matching peoples' expectations with reality kept users interested and engaged, as well as keeping them from becoming frustrated by a clumsy interface. This iteration of the drag and drop function worked by replacing the mouse icon with a smaller thumbnail of the selected sound to signify that it had been picked up. The sound would then be placed when the user released their drag in the desired location. We did not allow overlapping of thumbnails in order to avoid confusion and complexity; if a sound was dropped in an invalid location (off the canvas or on top of another sound) the sound would not be placed and would instead disappear.

To accommodate this more fluid placement and the possibility of invalid locations, the method for deleting individual thumbnails was changed. In Prototype 3 the method of clicking a sound to delete it caused some frustration with many visitors due to accidental clicks and a slight time delay between a click and a placement. We decided to remove this feature and utilize our new drag and drop functionality to create a new method. In Prototype 4, if a user wanted to delete a single sound, all they would have to do was drag and drop the placed sound anywhere off the canvas. The item would then be removed. This served two purposes. First, it changed the act of individual deletion to a less frustrating alternative. Second, it gave users the ability to move already placed sounds to a new location. If a user decided to move a sound, all they need do was drag and drop it to the desired location (if that location was invalid, but still on the canvas, it would snap back to its old position). This was done ultimately to give the user more intuitive and simple control over their soundscape creation.

This prototype also featured significant changes to the sound capabilities. In this iteration, we doubled our sound library from six to twelve sounds. With more sounds at our disposal, we then visually divided them into three thematic groups: natural sounds, human generated sounds, and technological sounds. The addition of more sounds was requested by users to give them the ability to create more accurate and diverse soundscapes. The variety of sounds available facilitated generation of more faithful soundscapes. When granted more options, users were generally more engaged and interested as they interacted with the prototype.

With the new sounds came new images, similar in visual style to the images of Prototype 3, and we wished to test them for sound association. Additionally, the thumbnails and buttons throughout the prototype now all matched their parent image. The placed thumbnail icon no longer differed from the original selected button. This avoided confusion about what a user had already placed on the canvas. A cartoon look was chosen in order to be both more appealing to children and also more easily recognizable on the smaller image sizes. Another visual change was the addition of a vertical time bar (in addition to the older time-bar). This feature was included to draw users' attention to the fact that sounds were being actively played as the vertical bar passed over them. It also gave a clear visual indicator of when a sound was going to play, so that they could better arrange sounds to their liking.

With the inclusion of twelve sound buttons, the layout of the screen was changed to facilitate ease of use. The large “clear all” button was replaced by a small button in the corner. This was left in largely for our use in testing; we believed that the main method of deletion should be the drag and drop method. The sounds were arrayed in two rows to allow for easy access with drag and drop, and we made them as large as possible to utilize the space.

4.2.4.1 Prototype Findings

Feedback concerning this prototype was significantly more positive than previous iterations. While still flawed and incomplete, users were generally happy with the drag and drop feature, the look of the canvas, and the visual feedback as to sounds being played. We experienced a total drop off in comments concerning image disparity and the annoyance of deletion. As all testers for this program were unique and had not seen previous versions, they offered us a distinct viewpoint to give us feedback on the true functionality of the exhibit, as opposed to our relative measures. While many of the major issues had been resolved, other less obvious, but equally important, problems came to the forefront.

One of the major issues that became readily apparent was an image-sound disparity. Our assumptions of what we thought were clear images to represent the sounds were often quickly proven wrong when users misattributed the sounds and became confused as to the sources of elements of their soundscapes. In general, the more abstract the idea, the larger the disparity. Human-generated sounds were most problematic because we could not easily display them in an intuitive, static cartoon form. For example, human sounds, such as coughing, were almost always met with a “what is that image?” response; only after hearing the sound did they make the connection. Other icons, such as using an ice cream cone to represent the sounds of an ice cream truck, caused confusion, because ice cream cones don’t ordinarily make sounds.

Another issue was that of sound balance. Certain sounds, like the jackhammer, were intended to be loud, while others, like bird song, were intended to be quieter. Issues arose when certain sounds jarringly stood out as loud or invasive. Users felt some sounds, such as coughing, were distracting from the rest of the exhibit, as they unrealistically drowned out background noise.

4.2.4.2 Evaluation Process Findings

This round of testing was accompanied by an overhaul of many of our testing procedures. Having gained experience from previous rounds of testing, we returned this time to the EcoTarium with a modified recording sheet and a scripted plan for interacting with visitors.

The new survey was designed to complement, rather than guide, our interaction with the users. The survey attempted to balance common questions and inquiries with our own adaptive observations of users. This allowed for a significantly smoother testing process on our end, which in turn afforded us the opportunity to more closely watch the user interaction and record data. This data sheet was successful and useful enough that we used it unchanged for the remaining iterations of our testing.

Additionally, our use of a standard user interaction plan allowed us to be consistent and effective in our interaction with users. By asking common questions and giving the same prompts to every user, we were able to standardize and normalize our testing to obtain a better understanding of just how well they understood the prototype (Appendix E). We realized that some individuals were less willing to engage with the soundscape prototype without more explicit instructions on how to use it. We concluded that a supplemental text panel would need to be integrated into the final version before it could stand alone as an exhibit on the museum floor in order to give optional instructions and provide off screen prompting for some visitors. It might also include educational prompts as supplementary learning material. In the meantime, until this was designed and created, we continued use of our standard prompt.

4.2.5 Prototype 5

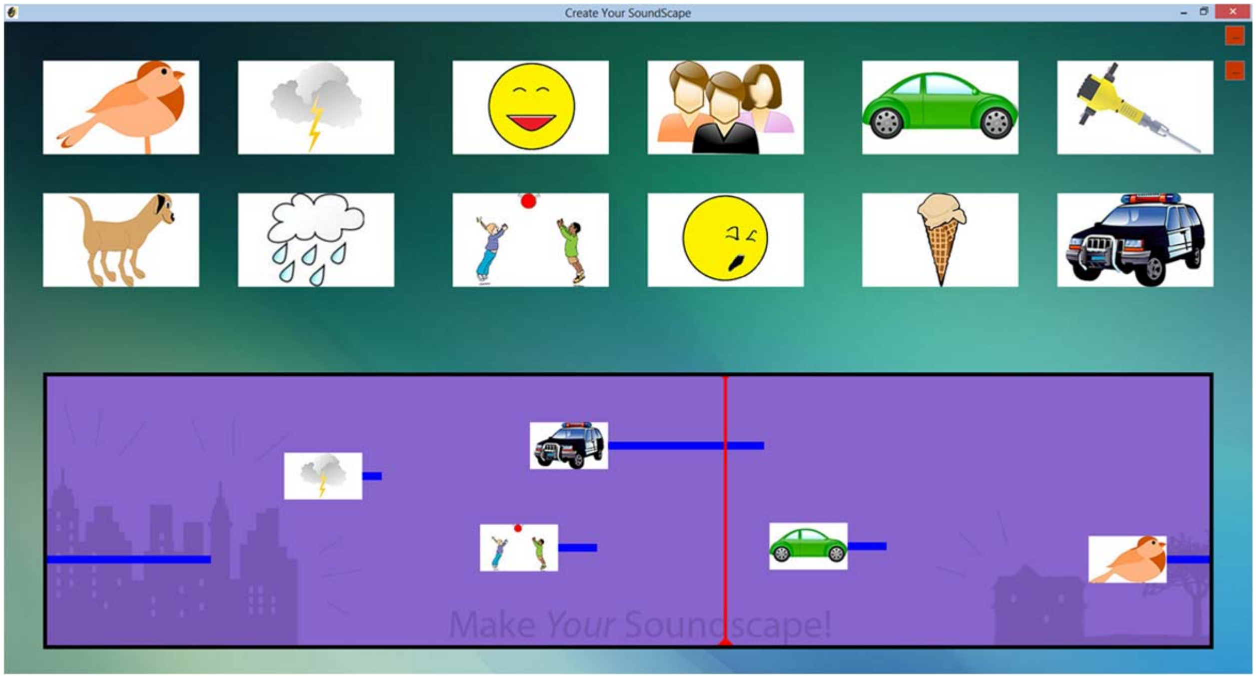


Figure 5- The interface of Prototype 5

Prototype 5, like Prototype 3, was in development prior to Prototype 4 being tested, due to our timeline. Many of its new features are those we intended to implement but had not yet had time to design, rather than changes which were a direct result of user feedback. The major modifications in this iteration were either cosmetic or behind the scenes coding to make the program easier to understand for the EcoTarium's technical staff and to reduce delays between user input and response. The new visual aspects had an immediate impact on the user experience.

The single largest change was the addition of visual bars appended to each thumbnail, representing the length of the sound clip. The lengths of the sounds were not standardized, in order to make them sound as realistic and natural as possible. As a result, users had no indications of how closely to place successive sounds in order to get a realistic or desired mix. With the addition of individual time bars, testers could now see how long a sound, such as bird song, was and either avoid unintended overlap or to overlap for a desired effect. If a time-bar ran over the right edge of the canvas, it automatically displayed the remaining length on the other side of the canvas, so as to mimic the looping functionality and the reality of how the sound played. Overall, this provided users with more perceived control and offered them less frustration when placing sounds.

Another visual change was the removal of the original scrolling time bar at the bottom of the canvas. We found it redundant and unnecessary with the addition of the vertical time bar, and it made the overall appearance less cluttered.

Two small buttons remained in the upper right hand corner. Both of these were used by us for testing purposes. One was the clear-all button. The second was an experimental preset button. We initially wanted to test if it was feasible to implement and worked well, and we wished to see if switching to a preset before user interaction began changed their actions.

4.2.5.1 Prototype Findings

When testing this prototype, there were no significant differences in feedback between this from Prototype 4. However, our entire sampled population for this iteration was high school students, due to a high school class attending a program entirely meant to test prototype exhibits, ours included. They would likely not have encountered the exact same issues as children or families. Users were generally pleased with the time bars attached to the thumbnails when prompted. We did notice more conscientious sound placement in this iteration, but overall users raised old issues we had not yet had time to address.

4.2.5.2 Evaluation Process Findings

No changes were made to our testing procedure beyond those made for prototype.

4.2.6 Prototype 6

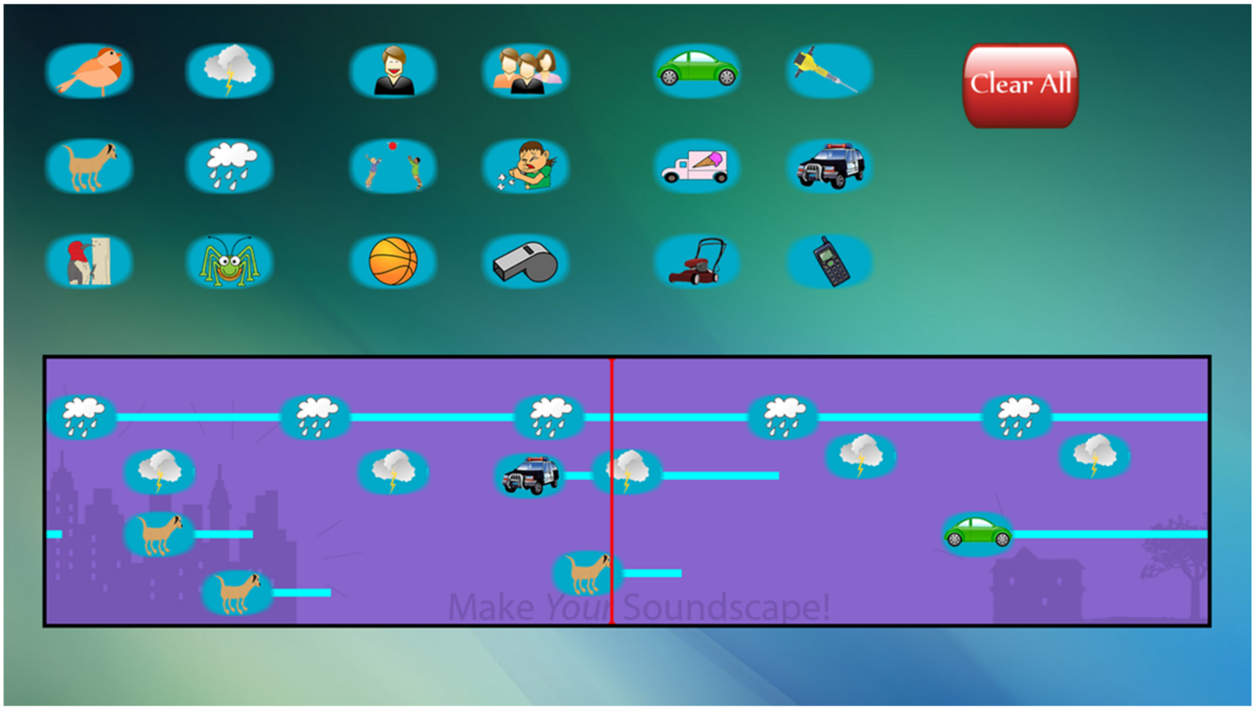


Figure 6- The interface of Prototype 6

Prototype 6 was our final major overhaul of the exhibit. Available features remained largely the same as in Prototype 5, but radical changes were made on the visuals as well as the back-end code. The program as a whole underwent extensive behind-the-scenes overhauls in order to streamline the program, make it run more smoothly, make it less bug-afflicted, and stabilize it.

We spent a considerable amount of time developing this iteration updating aspects of the user experience besides functionality. We once again expanded the sound library with an additional 6 sounds, bringing the total to 18. Additionally, each sound button now selected randomly between two available sound clips each time it was played. This added an enormous amount of variety, and made it statistically improbable that an identical soundscape will ever sound the same twice. The increased number of sounds and randomization between two options not only makes every user's experience unique, but also makes each loop unique for a user even if they do not change anything on the canvas. This decreases repetition and increased the immersion.

In designing the new buttons for the new sounds, we decided to completely overhaul the icon visuals. About 50% of the old sounds received new icons, and all of the images were standardized to give a more professional and polished feel. The goal of this was twofold. First, we were attempting to reduce or eliminate the image-sound disparity that many users had complained about. Second, by refining our imagery, our interface gained a more professional style and appearance. This had the effect of making our exhibit much more appealing to visitors who felt it looked much more like a floor-ready exhibit than a prototype.

Other cosmetic updates included new time bar colors and changing the visuals of dragged items. Now, when a user clicked a button to drag to the canvas, they carried a glowing/highlighted thumbnail to provide a visual full sized icon with which they could place on the canvas. This dynamically showed users what they were doing, afforded them the opportunity to see where exactly they were placing a thumbnail. While simple, and almost expected, it is a continued advancement in making the interface easier and more intuitive for users.

The one major functionality change was the full implementation of three preset buttons. When pressed, these buttons populated the canvas with a preset/pre-made soundscape that would give the users an example and starting point from which to modify and develop their own. We felt that giving users this option would help stimulate ideas and attract less inquisitive and experimental visitors. It was also hoped that should a user choose not to dedicate the time necessary to create a soundscape that they would still have the opportunity to experience the exhibit and learn about soundscapes.

4.2.6.1 Prototype Findings

Due to our timeline and scheduling, this prototype was not tested on the EcoTarium floor. Instead, feedback came from the EcoTarium staff. We presented the exhibit to them, and based on their original design goals as well as their ideas on Prototype 6, they provided us with feedback on how to proceed to a floor ready model.

This version of the prototype was similar to Prototype 4 in that it involved a significant overhaul of many parts of the exhibit. While no major functions were changed, the cosmetic changes were well received, with the general consensus that it made the prototype look and feel more professional. This was ultimately the direction the EcoTarium wanted to go, as the staff looked to put our prototype on the floor for an extended, unaccompanied tour.

Several concerns were raised on some of our design decisions. The major criticism dealt with our implementation of default soundscapes the user could easily select, which we termed 'presets.' The EcoTarium staff felt that the availability of the presets detracted from the experience of exploration and creation; with presets, users were no longer required to put significant thought into their neighborhood soundscape, but rather could just click a single button and move on. Several solutions to this were discussed, and ultimately we decided upon removing the preset buttons from the screen. Presets would remain in some capacity, but in a different implementation in the next iteration of the prototype.

Other concerns were raised on some of the sound clip choices that were made. Multiple sounds still had unbalanced relative volume levels, and others did not effectively integrate into the environment. Major revisions on the sound clips were planned for the next iteration in preparation for a floor model.

In addition, one idea put forth by the EcoTarium staff was the possible inclusion of varying the volume of a placed icon based on its location or another control. We considered implementing this design, but it quickly became apparent in internal discussions and in discussion with the EcoTarium technical staff that such a design would be too cumbersome when added to our already unique interface (Eric Zago, personal communication, February 1, 2013).

4.2.6.2 Evaluation Process Findings

Because we did not test this version on the EcoTarium floor, our methods process did not change. All feedback came from either the EcoTarium staff or our own observation.

4.2.7 Floor Model (Prototype 7)

Figure 7- The interface of the floor model

Prototype 7 is the deliverable package that was made available to the EcoTarium on February 18, 2013. Most of the improvement and modifications that were introduced in Prototype 6 are still present in Prototype 7. Prior iterations had no knowledge or sense of user interaction and time. The Prototype now maintains a global clock and is aware of user interaction. Added behind-the-scenes functionality enabled the addition of behavior in the absence of user interaction. If a user has not interacted with the interface for a certain amount of time, the interface generates a soundscape using the preset functionality that was introduced in earlier prototypes. A more user-friendly form of drag and drop was introduced in this iteration. Prior prototypes required that users drag or move the sound precisely to a valid location. This is no longer the case. The prototype now allows for a significant margin of error by looking for a nearby location to place the Sound Icon. Functionality that is not essential to the experience was hidden from the user. A user can no longer move, resize, minimize or close the application. Additional error checking was added that enabled the interface to recover from any unforeseen errors that may occur. All of the code was revisited and refactored. Full documentation was also added (see Appendices M and N for the XML document and Javadoc respectively). Prototype 7 was successfully installed on an EcoTarium provided touch screen for floor testing.

Prototype 7 is also the first prototype which does not require our immediate presence to run. Instead, we gave our "Education Panel" (See Appendix K) to the EcoTarium. It contains information detailing the basic functionality of the interface and brief educational material.

4.2.7.1 Prototype Findings

We saw great success with this prototype. The largest concern that we faced was that the computer used for this stage of testing was provided by the EcoTarium, and was significantly slower than the computers which we had used for our design and testing. This caused problems with the smoothness of the moving time-bar and caused a slight delay when images were dragged. There also seemed to be issues with the sensitivity of the touch screen provided by the EcoTarium, in that it struggled to detect dragging motions produced by small fingers. This makes it difficult for a child to have a productive experience with the interface.

However, we predict that the addition of the Education Panel as a reference guide to the use of the device will be a major benefit to the user experience. A significant issue during our testing was that users would simply not be aware that functionality existed, particularly the functionality to remove individual sounds from their soundscape.

4.2.7.2 Evaluation Process Findings

We did not evaluate this exhibit, but it was evaluated as part of a suite of new exhibits at the EcoTarium. However, we predict that there will be significant issues isolating problems with this iteration. The simultaneous addition of the Education Panel, a slower computer, and the questionably sensitive touch screen will make it almost impossible to determine which factors are most directly contributing the success or to the failure of the interface.

4.2.7.3 Interface Documentation

Upon completion of this iteration of the prototype, we created documentation of both the internal code (the Javadoc) and the editable functions (the XML file). The documents were provided to the EcoTarium. See Appendices L and M for these documents.

4.3 Visitor Education Analysis – The Education Panel

As stated in our Methods section, we found it difficult to isolate learning objective data while significant user interface functionality was still being implemented. As a result, the vast majority of data collected on learning came towards the end of the iteration cycle, specifically during Prototypes 6 and 7. With near-complete functionality, these versions allowed us to better determine if a user had learned about soundscapes uninhibited by usability complications. After observing users, we found that our education stemmed from two general sources: the iterative soundscape creation and the supplemental education panel.

4.3.1 Learning through the Interface

Embedded in our design philosophy is the concept that users will systematically generate and evolve their soundscape in real time as it is played back to them. This process allows for a constant feedback loop between the user and program, which leads to a final product that has been tested and verified by the user. This process may not have been conscious, but it developed in most users. Through this process, users discovered the concept of a soundscape and the overlaying of ambient sounds. They gained some understanding of how the sounds of their relative neighborhoods are put together, and would often detail their sounds back to us describing the dynamics between them (such as how traffic would distract them from other natural sounds). What often lacked from this half of the learning process was the vocabulary. Users would often describe a soundscape without using the exact word, or would come at the concepts from a roundabout view. Additionally, some users (especially the youngest children) were unable to make the initial connections to arrive at an understanding of soundscapes.

4.3.2 The Education Panel

In order to alleviate some of the fundamental learning problems, we began the development of an "Education Panel." This panel would be akin to the supplemental data that surrounds most museum exhibits. While originally not in our design requirements or objectives, we decided that this aspect of the exhibit was critical not only to education, but also to a general understanding of the exhibit's functions. The Education Panel consisted of simple instructions and examples on how the exhibit worked, and educational information and questions that would facilitate understanding of soundscapes. Having the Education Panel as a separate, physical object allowed us to keep the interface itself visually cleaner and more intuitive.

When the Education Panel was included, we found that many of the problems previously experienced in user learning were mitigated. The largest was the prominent information concerning soundscapes that allowed users to put a word to a concept. It also guided user's thought processes and directed their attention to the concepts of sound overlay and the mixing of natural, human, and technological sounds. In addition to the direct contributions to learning, the Education Panel also helped by further increasing the general usability of the program. By having the basic function instructions present but off screen, the user could more quickly jump straight into creating the sound of their neighborhood. This reduced confusion and led to an increase in time spent at the exhibit, which in turn increased learning potential. In many cases, some younger children would be too young to read the panel and often required their parents or other adult to help them use the exhibit. Even in this scenario, it allowed for adults to quickly determine the function of the exhibit and guide the child in its use and educational significance.

5. Conclusions and Recommendations:

Through our research, design, and iterative testing process, our project group generated a successful digital interface which enabled the exploration of the concept of soundscapes by EcoTarium visitors.

The EcoTarium required an exhibit which allowed user investigation and generation of soundscapes, as part of a larger City Science exhibition. Our overarching goal was to develop and test a prototype exhibit with the tag line "Create the Sounds of Your Neighborhood," that would be part of a larger exhibition and enable visitors to explore urban soundscapes (Appendix A).

Working with various members of the EcoTarium staff, we designed and developed an interactive, touch screen based exhibit which would allow guests of all ages, particularly children, to recreate the sounds of their neighborhood.

Conclusions Concerning Engagement

We have learned a great deal about the proper construction of interfaces, particularly in the museum setting. The most basic and important conclusion which we determined is that effective designs require strong, intuitive feedback between the interface and the user. As the user receives signals which confirm that they are performing the correct or incorrect task, they are better able to adapt and understand their soundscape. The feedback mechanism may be as simple as highlighting and color coding elements, or as complex as our time-bar indicators, which visually demonstrate the duration of sound clips.

Additionally, our group determined the greatest barrier to learning in the museum is if a visitor simply does not interact with the exhibit. If it is not completely and immediately self-evident how any given exhibit functions, the fear of failure will prevent some individuals from attempting to use the interface. To alleviate this, there are two key elements which the designer must consider: the design must have a minimum of extraneous visual elements, and it must have supporting material for patrons who require it.

Our first prototypes suffered significantly due to user confusion. Without drag and drop functionality and ease-of-use features we later implemented, testers found our initial prototypes too challenging. Users quickly became frustrated and eager to move on. Later prototypes drew greater interest as we refined our interface, visuals, and functionality. Appealing to visitors' expectations and biases allowed us to effectively engage them for longer periods of time, increasing their exposure to and exploration of the idea of soundscapes.

Conclusions Concerning Testing

The process of evaluating user interactions with any exhibit is a vital supplement to the engineering design process. Without feedback from users, any changes to a prototype have the potential to be blind or arbitrary. Our extensive research into exhibit design, education, and learning models was used to inform our prototype design. When determining planned features, we applied this knowledge to create intuitive functionality and facilitate a simple user experience. What we found when our various interfaces were tested was our research alone did not provide us sufficient data in order to create a user friendly interface. We required supplemental data obtained from monitored user interaction in order to substantially improve our prototype for the EcoTarium floor. We found that although one may try to design an exhibit based purely on evaluation schemata and insight found in literature, direct interaction with the target audience provides the only feedback that is directly relevant to that same audience.

We experimented with different evaluation methods over the course of our prototype iterations to find more effective testing protocols and procedures. Our initial surveys were largely useless for the rate of data collection and types of information that we attempted to gather. We adapted our survey, added verbal prompts, and generally streamlined our evaluation methods over the course of our testing. By Prototype 4, we had identified an efficient and informative system for data collection. Our modified system was simpler and open-ended, but yielded more relevant information. We conclude that while intuition might lead one to seek complex answers to complex problems, a simple, targeted approach is often superior to a widely comprehensive approach.

Our personal and professional approach to interaction with museum visitors improved greatly over the course of the project. We found that visitors generally do not volunteer for testing without some sort of encouragement. Instead, we often had to directly request that guests test our exhibit. Determining the balance between professionalism and approachability is vital, particularly during interactions with children, who are often shy to interact with unfamiliar persons. Finding ways to engage and guide them in case of confusion was difficult at the outset of the project. Eventually, we developed a set script which allowed us not only to be consistent, but also allowed us to interact with users in a way that enhanced their experience and allowed us to inquire directly about their valuable opinions and concerns.

Conclusions Concerning Education

In the course of our research, we examined the various roles served by museums in society and the objectives of museum organizations with reference to education. Museums attempt to serve multiple roles to individuals spanning virtually all demographics. To evaluate if their exhibits are successful in such an environment, they often rely upon evaluation schemata similar to APE and PISEC. We conclude that designers must consider their evaluation schemata at all times, as they are the metrics which determine success or failure.

Central to the design process for a museum setting is the need to ensure that learning occurs as a result of interaction with an exhibit. An interactive exhibit needs to serve a role greater than that of simple entertainment or a game. Exhibits have minimal value to the museum without the ability to convey information to individuals who possess distinct and diverse learning styles, or if the information that it conveys is too distinct from the original objective. On examination of our final exhibit, we conclude that we have fulfilled the requirements set by ourselves and by the EcoTarium, and have generated a valuable addition to the museum's learning environment.

Recommendations

To assist with their ongoing use of this interface, we recommend that the EcoTarium take a number of steps.

First, the museum should use a computer with greater processing power. The delays caused are significant enough to impact user experiences with the exhibit. Particularly, the museum should avoid using the interface on a computer which requires a network connection unless steps are taken to ensure that the computer has the requisite processing power to overcome inherent limitations.

Second, the EcoTarium should consider use of a different touch screen. Although our experience with the model that they are using is limited, it was difficult for users to take intended actions, even those as simple as sliding an icon. This sort of negative feedback will quickly become frustrating to many users and will make them significantly less able to explore the concept of soundscapes.

Third, we do not recommend that the EcoTarium attempt to use this prototype as the basis for other exhibits. While it may be tempting to use the existing architecture for other applications, it would be very challenging to adapt this highly specialized application to other purposes.

Fourth, we recommend continuation of iterative testing on both the interface itself and on the Education Panel. We are aware that the exhibit will at least be edited to come more into line with the visual themes of the rest of the museum, and those revisions provide for another round of iterative testing which may ensure that the project meets its desired goal of providing an immersive exploratory environment.

Fifth, we recommend that the EcoTarium continue its excellent practices and protocols with regards to exhibit evaluation. Steps should be taken to ensure that effective training of staff in these techniques continues into the future.

Bibliography

Alexander, E., & Alexander, M. (2007). Museums in motion: An Introduction to the History and Functions of Museums. Altamira Pr.

Allen, S., & Gutwill, J. (2004). Designing Science Museum Exhibits with Multiple Interactive Features: Five Common Pitfalls. California: California Academy of Sciences, 2-19.

Anderson. (1968). Noseprints on the Glass.

Audacity. (2012, August). Scripting. Retrieved from http://manual.audacityteam.org/man/Scripting

Axelsson, Ö, Nilsson, M, & Berglund, B. (2010). A principal components model of soundscape perception. Journal of the Acoustical Society of America, 2836-2846.

Balaÿ, O. (2004). Discrete mapping of urban soundscapes. Soundscape , 13-14.

Betty, F., & Mededeva, M. (2010). Demographic Transformation and the Future of Museums. Washington, DC: The AAM Press, American Association of Museums .

Borun, M. (2010). Visitor Evaluation Project - Rethinking Exhibitions. Retrieved September 24, 2012, from Family Learning: http://www.familylearningforum.org/rethinking-exhibitions/designing-exhibits/intro-pisec.htm

Cairns, P., & Haywood, N. (2005). Engagement with an Interactive Museum Exhibit. London: UCL Interaction Centre , 13-14.

Council Directive 2002/49/EC of 25 June 2002 relating to the assessment and management of environmental noise. (2002). Retrieved from http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:189:0012:0025:EN:PDF

Cumberland, M. (2001). Bringing Soundscapes Into the Everyday Classroom. Soundscape , 16-20.

Din, H., & Hech, P. (2007). The Digital Museum : A Think Guide. Washington, D.C.: American Association of Museums , 24.

Falk, J. H., Dierking, D. L., & Foutz, S. (2007). In Principle, In Practice: Museums as Learning Institutions. Lanham: AltaMira Press.

Eclipse Foundation, The. (2012). WindowBuilder. Retrieved from http://www.eclipse.org/windowbuilder/

Fritsch J. (2007). "Thinking about Bringing Web Communities into Galleries and How it Might Transform Perceptions of Learning in Museums" (paper presented at the Museum of Social Laboratory: Enhancing the Object to Facilitate Social Engagement and Inclusion in Museums and Galleries seminar, Arts and Humanities Research Council Seminar Series, March 2007).

Hein, G. E. (1998). Learning in the Museum. Retrieved from http://site.ebrary.com/lib/wpi/Doc?id=10056149

Hilke, D. (1988). Chapter 12: Strategies for Family Learning in Museums. Retrieved from http://informalscience.org/researches/VSA-a0a1o2-a_5730.pdf

Hochmair, H. (2004). Vienna Soundwalk. Soundscape , 26-27.

Humphrey, T., & Gutwill, J. (2005). Fostering Active Prolonged Engagement. Walnut Creek, CA.

Smithsonian Institute, The. (1999). Publications. Retrieved from The Peale Family Papers: http://www.npg.si.edu/exh/peale/index-pubs.html

Pekarik, Andrew et al. (2002, May). Developing Interactive Exhibitions at the Smithsonian. Retrieved from http://www.si.edu/Content/opanda/docs/Rpts2002/02.05.InteractiveExhibitions.Final.pdf

J., F. (2007, March). Thinking about Bringing Web Communities into Galleries and How it Might Transform Perceptions of Learning in Museums. paper presented at the Museum of Social Laboratory: Enhancing the Object to Facilitate Social Engagement and Inclusion in Museums and Galleries seminar, Arts and Humanities Research Council Seminar Series .

Jensen, N. (1999). Children, teenagers, and adults in museums. London: Routledge.

Kreutzfeldt, J. (2010). Acoustic Territoriality and the Politics of Urban Noise. Soundscape , 14-17.

Lord, B. (2007). The Manual of Museum Learning. 173-174.

Marry, S. (2010). Spatial and Sonic Evaluation of Urban Ambiances. Soundscape , 18-22.

Paquette, D. (2004). Describing the Contemporary Sound Environment: An Analysis of Three Approaches, their Synthesis, and a Case Study of Commercial Drive, Vancouver, B.C. (Master's thesis). Retrieved from http://www.sharawadji.org/thesis/files/page0_1.pdf

Poff, Z., & Aldrich, N. (n.d.). Retrieved from Interactive Soundscapes: http://www.interactivesoundscapes.org/index.html

Schafer, R. M. (1994). The Soundscape. Rochester: VT: Destiny Books.

Serrell, B. (2012). Center for Advancement of Informal Science Education. Retrieved from Paying More Attention to Paying Attention: http://caise.insci.org/news/96/51/Paying-More-Attention-to-Paying-Attention/d,resources-page-item-detail

Simon, N. (2010). The Participatory Museum. Retrieved February 8, 2013, from http://www.participatorymuseum.org/read/

Soundscape Constructor. (n.d.). Retrieved from Exploratorium: http://www.exploratorium.edu/listen/activities/soundscapes/deploy/activity_soundscapes.php

Stone, D., Jarret, C., Wodroffe, M., & Minocha, S. (2005). User Interface Design and Evaluation. United States of America: Elsevier, Inc.

Tallon, L., & Walker, K. (2008). Digital Technologies and the Museum Experience. Lanham: AltaMira Press.

The Tech Museum. (2012). The Tech Museum. Retrieved from Spirit of Silicon Valley: http//www.thetech.org

Tisdal, C. (2004). Active Prolonged Engagement at the Exploratorium. Selinda Research Associates, Inc.

Tisdal, C. (2004). Going APE! Chicago: Selinda Research Associates, Inc.

Wrightson, K. (2000). An Introduction to Acoustic Ecology. Soundscape , 10-13.

wxWidgets. (2012). Cross-Platform GUI Library. Retrieved from wxWidgets: http://www.wxwidgets.org/

Yilmaz-Soylu, M., & Akkoyunlu, B. (2009). The effect of learning styles on achievement in different learning environments. Turkish Online Journal of Educational Technology , 43-50.

6. Appendices

Appendix A: Original EcoTarium Documentation

This Appendix contains the first explicit design goals for our project, provided to us by the EcoTarium. While vague, these documents gave us a basic understanding the intent of the museum with regards to this project.

) Neighborhood Soundscape. Visitors drag different soundscape elements (traffic, birds in trees, playground, dog barking, different types of music, etc) onto a "soundscape" interface. This soundscape is played through multiple speakers sounding the station, and also becomes a soundtrack for this area of the exhibition. Some sounds provide a continual background (traffic), while others are more episodic (dog barking). The whole soundscape loops, perhaps every 30 - 45 seconds.

Considerations/Discussion:
--are visitors dragging elements via touch screen? Or is it a real physical installation, using something like a turntable to create the space?
Is there a way to control volume and/or duration of each sound element you add?
Physical interface, while an attractive idea, would probably have serious issues in terms of reset, and the tendency to add in all the pieces at once.

-- As with the neighborhood area, ideally there would be a way to save your soundscape, perhaps with comments. Perhaps there is a way to download to your phone.

-- If no one was using the exhibit it should cycle though some saved soundscapes as an attract loop/exhibit sound track.

-- Should the exhibit be primarily designed for one family group to be using the interface to create there neighborhood, or a larger group activity with different people controlling, say, nature sounds, vehicles sounds, and people sounds? The former will get people thinking more about neighborhood differences, the latter is arguably more analogous to a city with the many contributions to the soundscape. Another alternative is to have two interfaces – each a Neighborhood, with the natural overlap area between them.

3) Noise Pollution. A collaborative activity between two visitors. One visitor puts his/her hands or fingers on a device that measures stress level, (heart rate or skin conductivity) while the other visitor, on the other side of a table, selects audio components. while also seeing a read out of the monitor. Some sounds will promote calm, others stress or excitement. Some prompts may cause different reaction in different people. The challenge: can you make the person calm or not? What do you consider noise pollution? Does everyone react the same way? Some sort of mechanism would be provided to record visitors' reactions to different stimuli – i.e. how many people consider a given sound nice or noise pollution. This could be a physical graph, or other sort of record.

Considerations/Discussion:
-- The physiological monitoring software is available from Vernier (we have used there systems in exhibits before), PASCO or other microcomputer based lab systems.
- The sound interface may want to have some connection to the soundscape interactive so it feels familiar, if only reuse of the same icons.
- Sounds would need to be played over a directional speaker right above the exhibit. Since this exhibit wants to be close to the other sound exhibit (conceptually) but also needs some sound isolation we may need to build a partial surround to mitigate other noise.

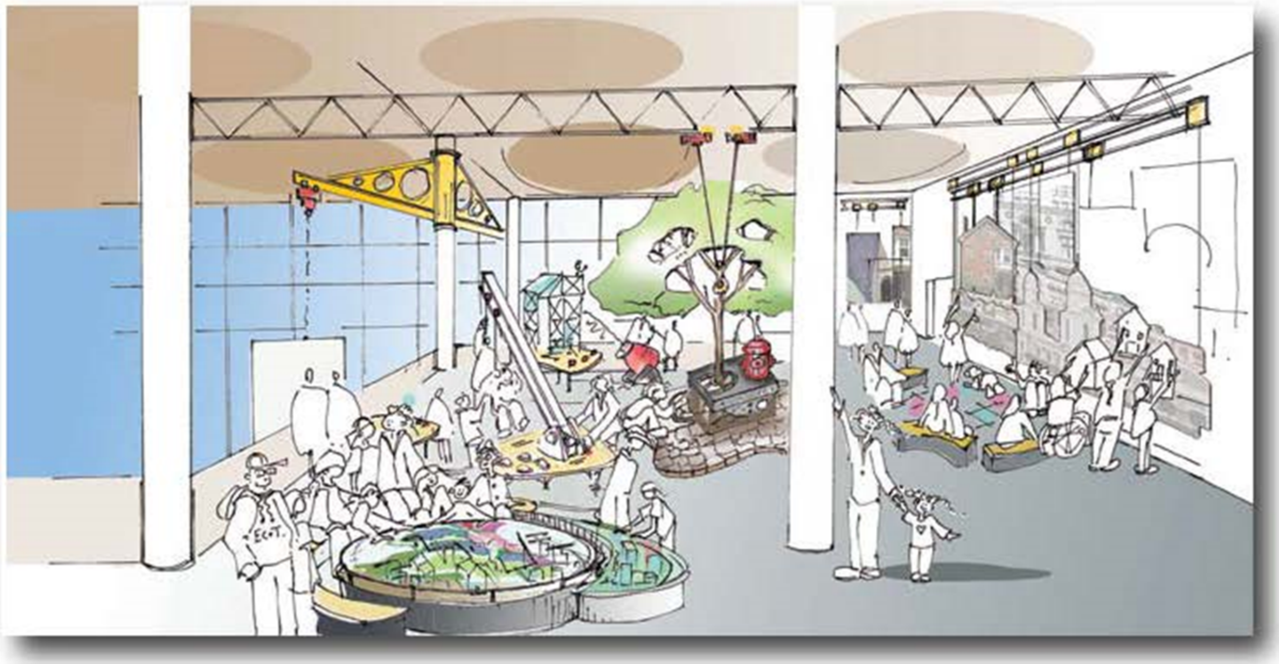
The original design specifications laid out by the EcoTarium.

CITY SCIENCE

Conceptual Design

[EcoTarium Exhibit Master Plan, Middle Level]

CITY SCIENCE will be an interactive look at the science we encounter in our every day lives, but rarely stop to consider. What keeps our buildings and bridges standing? What is the hidden infrastructure that brings water in and out of our houses? Can you design a better solution for a problem intersection? How do trees affect temperature -- and your air conditioning bill? Hands on experiment stations will allow guests will explore topics from civil engineering to city planning and issues of urban ecology, pollution and health impacts. The exhibit will explore the people, plants and animals that call the city home. Located on the middle level of the museum, the dramatic two-story atrium space will include a moving city skyline, and even a construction crane.

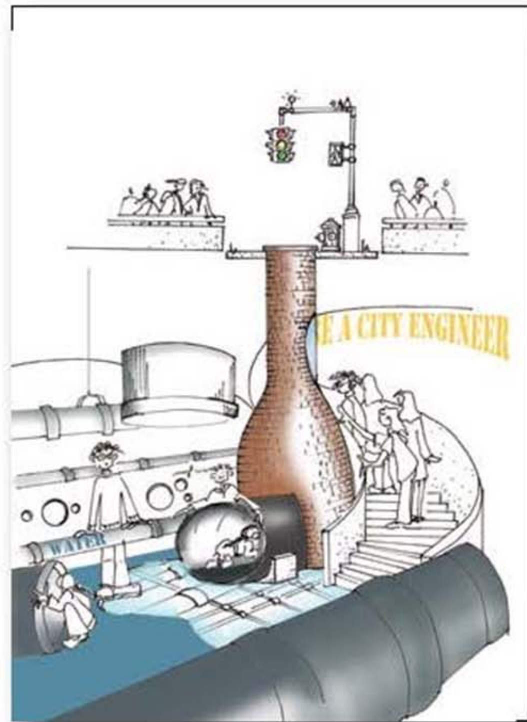


HIDDEN CITY NETWORKS: Guests will encounter their first glimpse of the City Science exhibit from the top floor: a traffic light and manhole cover emerging from the stairwell. Descending the stairs, visitors will find out what happens below the manhole, including a video from Worcester's sewer cam robot. Interactive exhibits will let visitors investigate the systems and networks that keep the city running. A traffic simulator will challenge visitors to improve traffic flow and keep vehicles moving smoothly through intersections.

THE HUB: Viewing the city from above shows cars streaming along the interstate, pedestrians on the streets, trains coming and going, and unloading containers – a complex system analogous to a living organism. A multi-touch computer interactive combines live projection of Worcester with GIS data layers allowing visitors to see infrastructure systems, or even historical views showing the location of the Blackstone canal hidden under the streets, or development of the rail system over time.

ENGINEERING LAB: Combining large scale interactives with table top engineering challenges, visitors in this area can operate the overhead jib crane or hang on an I-beam to see how much it deflects under their weight. Using an air cannon visitors can test the movement of a model of the EcoTarium's window wall and examine why structures are designed to flex. Visitors will try their hand at being an engineer, testing materials and choosing the best materials to meet different challenges. Visitors can examine traditional building materials and new ones such as carbon fiber composites and metal foams. Before leaving the Engineering Lab, visitors can create their own city skyline using theatrical rigging and controls to move images of houses, buildings, towers and trees to create varied cityscapes on the walls.

NEIGHBORHOODS: Create the soundscape of your neighborhood. Test your body's response to different sounds. What is relaxing? What is noise pollution? Be a city planner and build your ideal neighborhood and record your creation for others to see. At the Block Party children engage in a hands-on building activity while parents can discuss the challenge and share ways to make their community better.



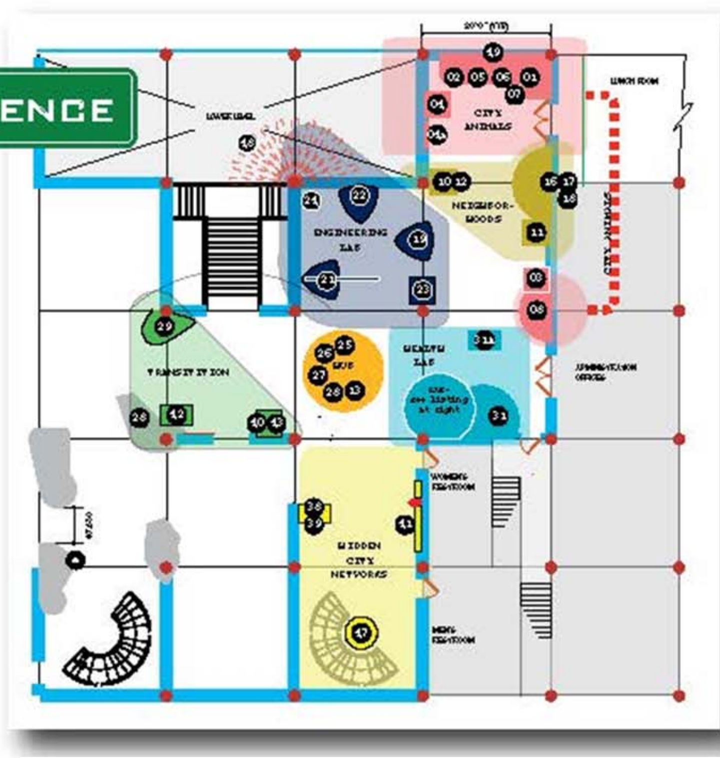
CITY ANIMALS: People are not the only inhabitants of the city. What comes out at night? A live skunk exhibit will provide a close up view of one common city resident. Images from camera traps on our property will let visitors explore other urban wildlife. Visitors can view and identify birds at our birdfeeders and see live pigeons and their chicks – a familiar bird with amazing behaviors (did you know pigeons produce a sort of milk to feed their chicks?). An elevated cross-section provides a peek beneath a city sidewalk and tree showing the miniature ecosystems in sidewalk cracks.

HEALTH LAB: How does your environment effect you? What is in the air you breath and how do you measure it? Visitors can experiment with an infrared camera to see how cities can trap and radiate heat. By manipulating models of buildings, roads, and trees, visitors can see how heat islands are formed and ways to mitigate their effect.

TRANSITION: Linking the City Science exhibit with the forest exhibits on the middle floor, the transition exhibits explore the changing landscape. Live wood turtles show the effects of habitat fragmentation. Visitors will be challenged to solve the Disappearing Plant Mystery and examine how we can design our streets and lawns to improve water quality.

CITY SCIENCE

Preliminary Exhibit Floor Plan



EcoTarium

science.nature.explore.connect.®

The EcoTarium's City Science Exhibit Original Conceptual Design (page 2)

Appendix B: Research Aims

This Appendix shows our initial research goals, based on a very early understanding of our project.

Overall Research Question	**Overall Project Goal**
What type of design (visual, audible, interactive, and presentation) is best designed for engaging and ultimately teaching visitors to the EcoTarium about city soundscapes?	The goal of this project is to evaluate and design an exhibit that offers visitors to the EcoTarium the ability to experience and learn about city soundscapes in a manner that is engaging and promotes interest in the topic.
Subsidiary Research Questions	**Project objectives:**
1. What is the historical context of museum exhibits, and how do modern changes affect our exhibit? 2. What are the characteristics of an interactive space? 3. What are the types of museum exhibits? What are their differences? 4. What is the audience and the target demographic? 5. What stimulates interest and learning in a topic? How does this vary by demographic? 6. How do different demographics learn, and how do we create an exhibit that appeals to as many demographics as possible?	(1) Determine the characteristics of an engaging museum exhibit. What types of interactivity promote learning and interest in a topic?
1. What is the overall purpose of the exhibit? 2. What are the museum staff's expectations or requirements for the design of the exhibit? 3. How do other related exhibits compare and perform? 4. How do we measure the success of achieving the goals of the museum staff?	(2) Determine the specific goals of the museum staff in order to create an exhibit that meets their expectations as well as provides a unique and successful experience to visitors.

1. What draws a person into an exhibit? What makes them stay? 2. What type of interactivity (or lack thereof) empowers a person to remain at an exhibit? 3. How do you teach visitors about soundscapes in a way that is interesting and inspires continued exploration?	(3) Design an exhibit that engages visitors to stay and participate long enough to learn about soundscapes.
1. What is the layout of the exhibit? 2. How will sound and visuals be projected to visitors? 3. What types of interactivity will be present and how will they be implemented? 4. How do we create a complete learning experience and present it in a way that attracts different people and interests?	(4) Design and model an exhibit that can be tested and improved upon.

Appendix C: Initial Evaluation Forms

This Appendix contains the first evaluation forms that we used as port of our EcoTarium testing. The former consists of observed behaviors while the user experienced the exhibit, while the latter consists of a questionnaire, intended to be asked after the user's testing experience.

Focused Observation: Open Ended Investigation

(Modified from EcoTarium Materials)

Date: ___________

Start Time: ________

End Time: _________

Age:	Adult					
F						
M						

Group Type (circle one): Family School Group Other: _________________

EXHIBIT BEHAVIORS:

☐ Did the child volunteer generate a sound-scape?

☐ Did the child volunteer generate more than one soundscape?

☐ Were there later soundscapes that were modifications of an earlier soundscape?

☐ Did any of the generated soundscapes use at least three different sounds?

☐ Did an adult physically touch the interface while the child volunteer was creating their soundscape?

☐ Did more than one child volunteer work to make the same soundscape?

MISCELANEOUS NOTES

Which sounds did the child use in each iteration of their soundscape (first box is first iteration).

Siren

☐☐☐☐☐

Traffic

☐☐☐☐☐

Music

☐☐☐☐☐

Wind

☐☐☐☐☐

Birds Chirping

☐☐☐☐☐

Children Playing

☐☐☐☐☐

☐ The child volunteer generated more than five soundscapes

City Soundscapes

EcoTarium Prototype Exhibit Evaluation

Date:____________ Time:___________ Surveyor:______________________________

We are a group of students at Worcester Polytechnic Institute. We are gathering information on a prototype exhibit in order to improve and create a new exhibit for the EcoTarium. Will you please take 5 minutes of your time to answer the following questions?

GROUP COMPOSITION:

MALE	FEMALE	OTHER	|	# of ADULTS(+18)	# of CHILDREN (<18)
_______	_______	_______	|	_______	_______

ENLGISH KNOWLEDGE: FLUENT SOME LITTLE NONE

1) How often do you visit the EcoTarium in a year?

   0-1 2-3 4-6 7-10 10+

2) What age group do you generally fall into (if a family, please select the age of the adults)?

   <18 yrs. 18-22 yrs. 22-39 yrs. 40-59 yrs. 60+ yrs. No Answer

3) Why did you visit the EcoTarium today?

   ____

   ____

4) How do you feel about the use of digital technology in the EcoTarium? Does it help or hinder your experience?

   ____

   ____

5) If you could change one particular thing about the technology in this exhibit, what would it be? Why?

   ____

   ____

6) Please name one thing you learned from this exhibit.

   ____

   ____

7) Did you have any trouble understanding or using any part of this exhibit? If so, what?

   ____

   ____

8) How technologically proficient would you consider yourself?

   1 2 3 4 5 6 7 8 9 10

   Not Proficient Very Proficient

Appendix D: Modified Evaluation Form

This Appendix contains the evaluation form that we used for later testing.

Focused Observation

Number: ________

Evaluator: ________

Date: ______

Start Time: ______

End Time: ______

Group Type:

Family School Group

Other: ____________________

Sound	Used	Didn't Understand Image	Didn't Understand Sound
Bird			
Rain			
Thunder			
Dog			
Coughing			
Laughter			
Crowd			
Children			
Car			
Jackhammer			
Siren			
Ice Cream Truck			

Adults:

Mom Dad Teacher Guardian

Children:

Age			
Number			

Notes:

What improvements did the adult or child suggest?

What actions did the parents take?

What challenges did the child experience with the interface?

Misc.

Appendix E: User Interaction Script

This Appendix details the major interactions that we had with visitors prior to and during our testing at the EcoTarium. Please note that these are not necessarily verbatim, and if we felt that a user was not having a productive experience we were always willing to go off-script.

<u>Initial Visitor Interaction Script</u>

> Hello, my name is __________ . My group and I are from Worcester Polytechnic Institute. We are working with the EcoTarium to test a new prototype exhibit to hopefully be part of the museum one day.

> Would you be willing to help us test this new exhibit by trying out our prototype today?

> You can stay for as long as you want.

> Your feedback is important so that we can improve our prototype and make it more user-friendly.

<u>Testing User Prompt Script</u>

> The goal of this exhibit is to “Create the Sounds of Your Neighborhood.”

> You can drag sounds down from the top onto the purple canvas.

> These sounds will then play as the time-bar passes over them.

> Give it a try.

> Mix the sounds to make it sound like your neighborhood.

Appendix F: Initial Plan for Casual Pretesting

This Appendix is the proposed methodology for our initial plans to pretest our images and sounds. It was never implemented.

Casual Pretesting for Image Recognizability

1) The tester will prepare a PowerPoint presentation, with each slide containing an image he wishes to test. This image should be small (not more than two inches in any direction) when the slideshow is made to be full screen. In the notes section of the PowerPoint presentation, the tester will write the sound that that particular image is expected to correspond with. This presentation should not be more than 50 slides in length. Multiple representations of the same expected sound may be used (for example, both a police car and a fire truck being used to represent the sound "siren,") but the same image should not be used more than once.

2) The tester will print himself a "handout" copy of this PowerPoint, with the answers to each slide displayed

3) The tester will present the slides, one by one, to the volunteer. Upon viewing each, the volunteer will state "what sound he believes is best represented by the image."

4) The tester will record yes or no if, in his estimation, the volunteer gave the correct answer.

5) After the presentation, the tester will inquire about and informally record any comments that the volunteer has with reference to the regcognizability of the images.

Casual Pretesting for Sound Recognizability

1) The tester will prepare a file containing a series of 15 second sound clips, each titled as what the clip is intended to represent.

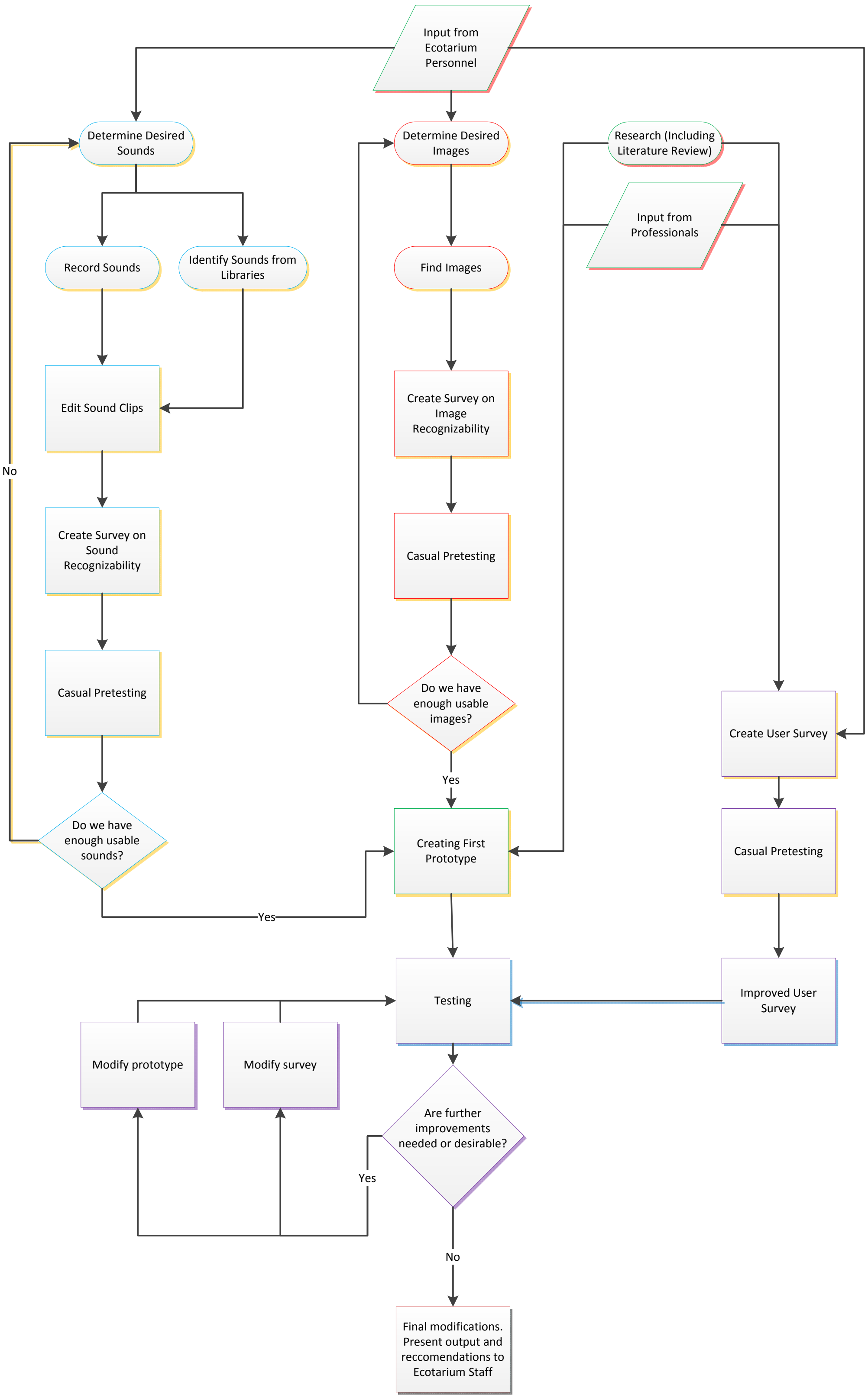
2) The tester will play the sounds, one by one, to the volunteer. Upon listening to each, the volunteer will state "what generated" each of this sounds.

3) The tester will record yes or no if, in his estimation, the volunteer gave the correct answer.

4) After the presentation, the tester will inquire about and informally record any comments that the volunteer has with reference to the regcognizability of the sound clips.

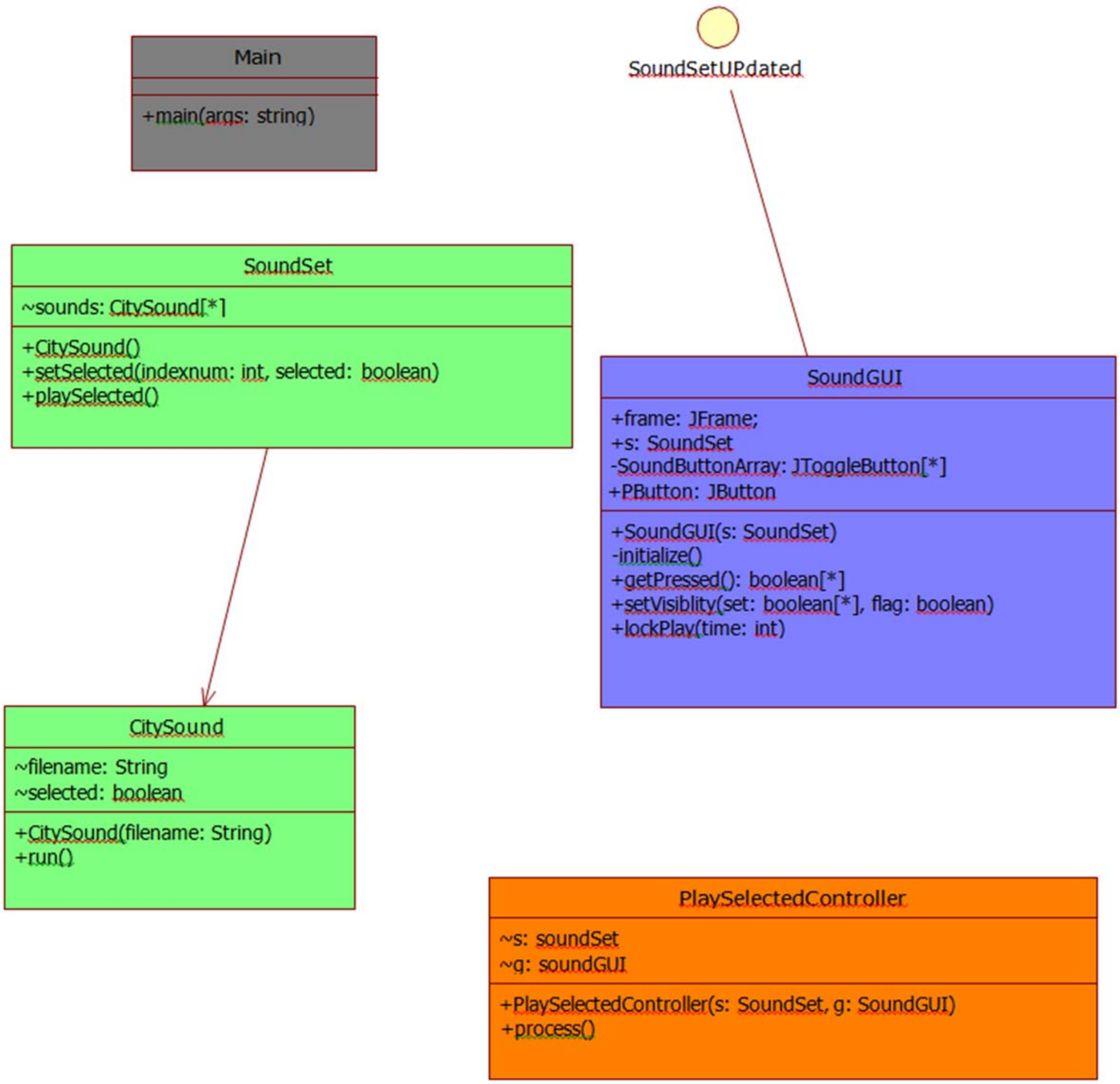
Appendix G: Original Proposed Workflow

This Appendix details our original planned workflow. It was generated concurrently with our research.



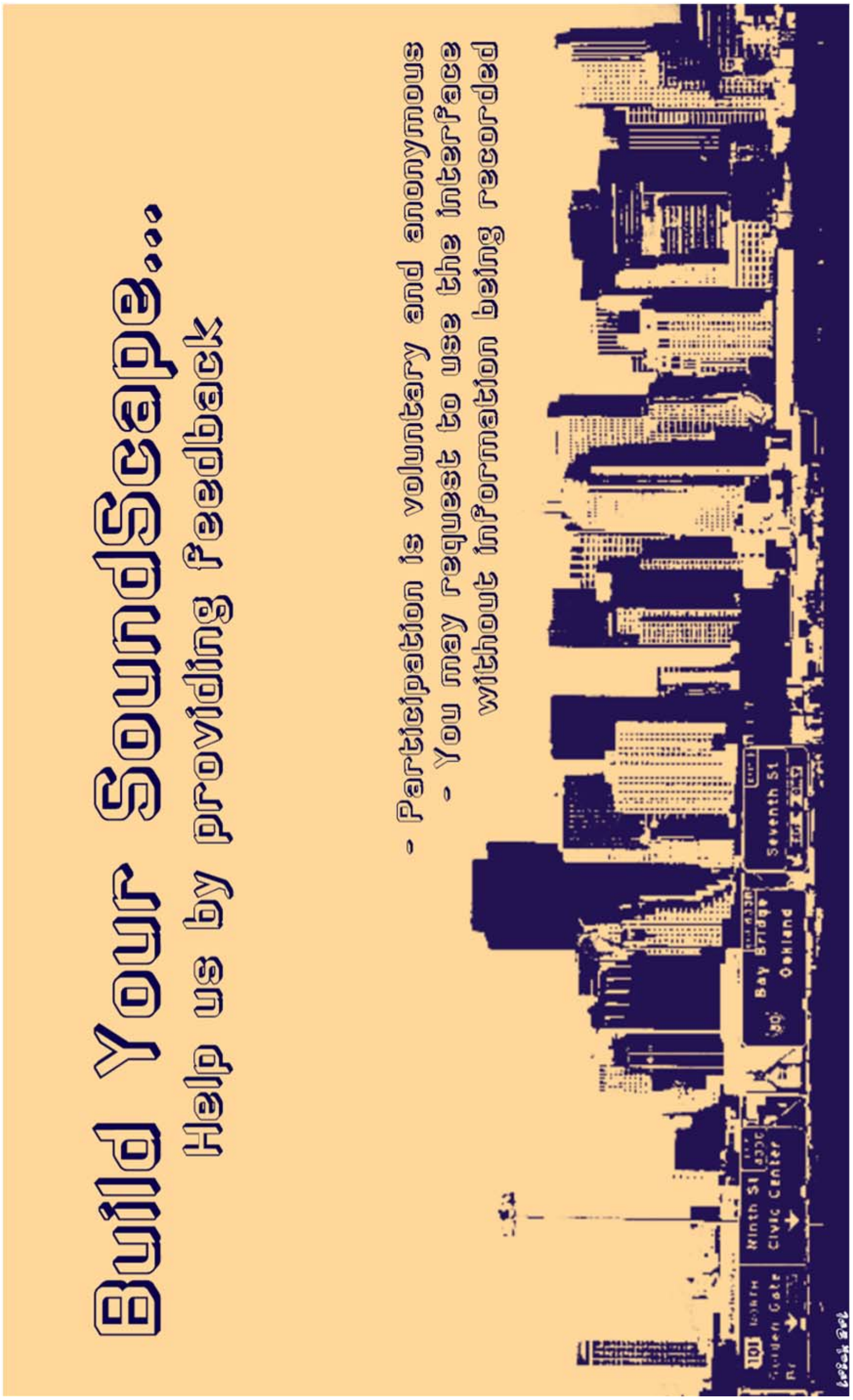
Appendix H: UML Diagram

This Appendix details the initial design layout of the JAVA program code structure. It indicates the planned flow of information and execution of the anticipated functions.



Appendix I: Legal Information Signage

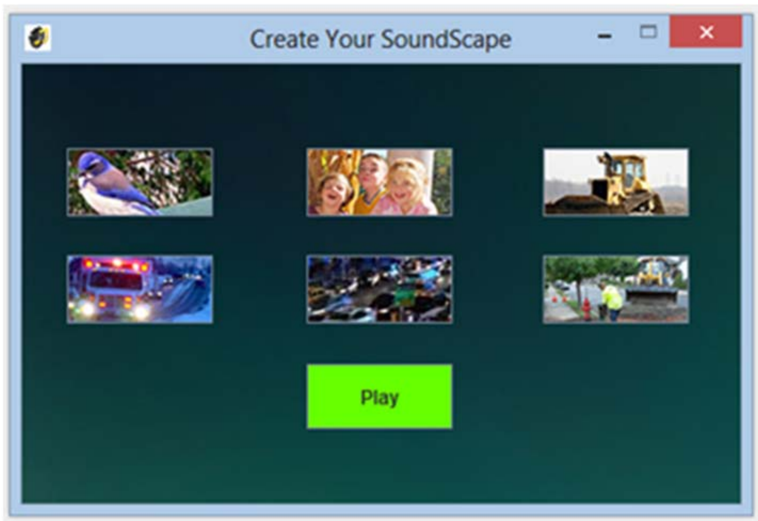
This appendix shows the later signage that we used to inform and attract visitors to our testing location in later iterations of the prototype.



Appendix J: Prototype Changes and Iterations

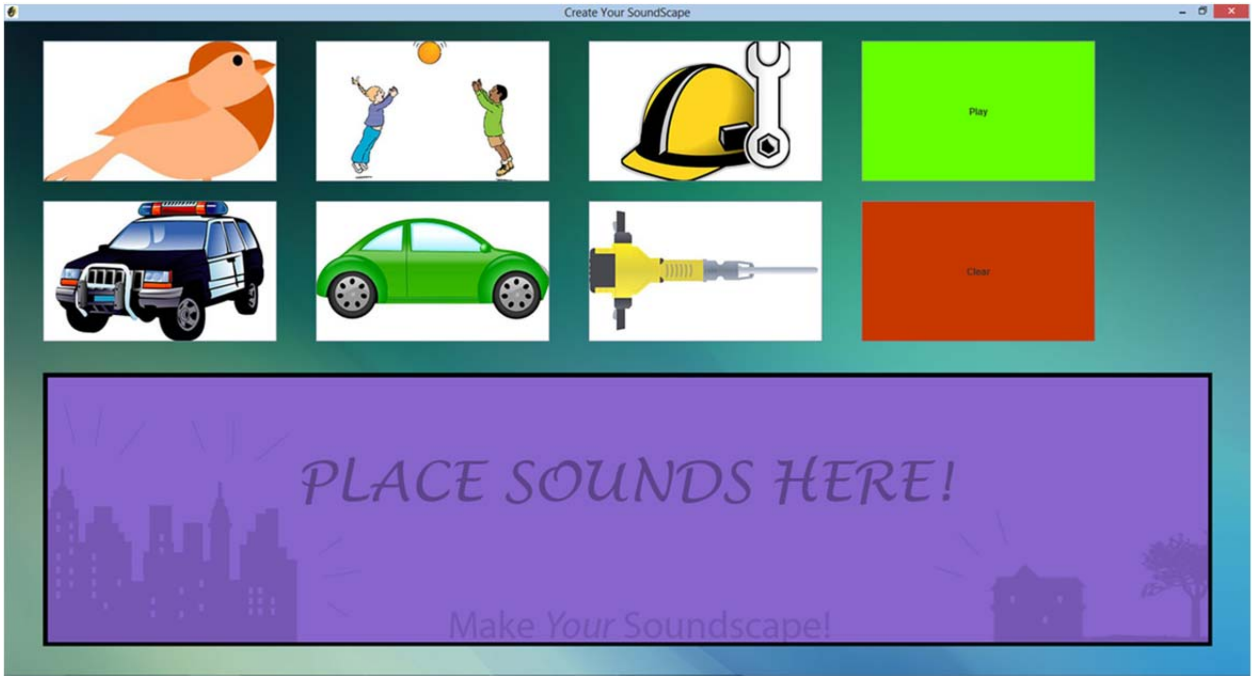
This Appendix documents the major changes to each iteration of the prototype interface.

Prototype 1



- First Iteration
  - → The iteration existed as a proof of concept for the Java platform and the possibility of sound overlay
- Clicking a button selected the sound
  - → The buttons highlighted when clicked
- Selected sounds would play once the "Play" button was pressed
  - → Sounds would overlay and play together creating a soundscape

Prototype 2



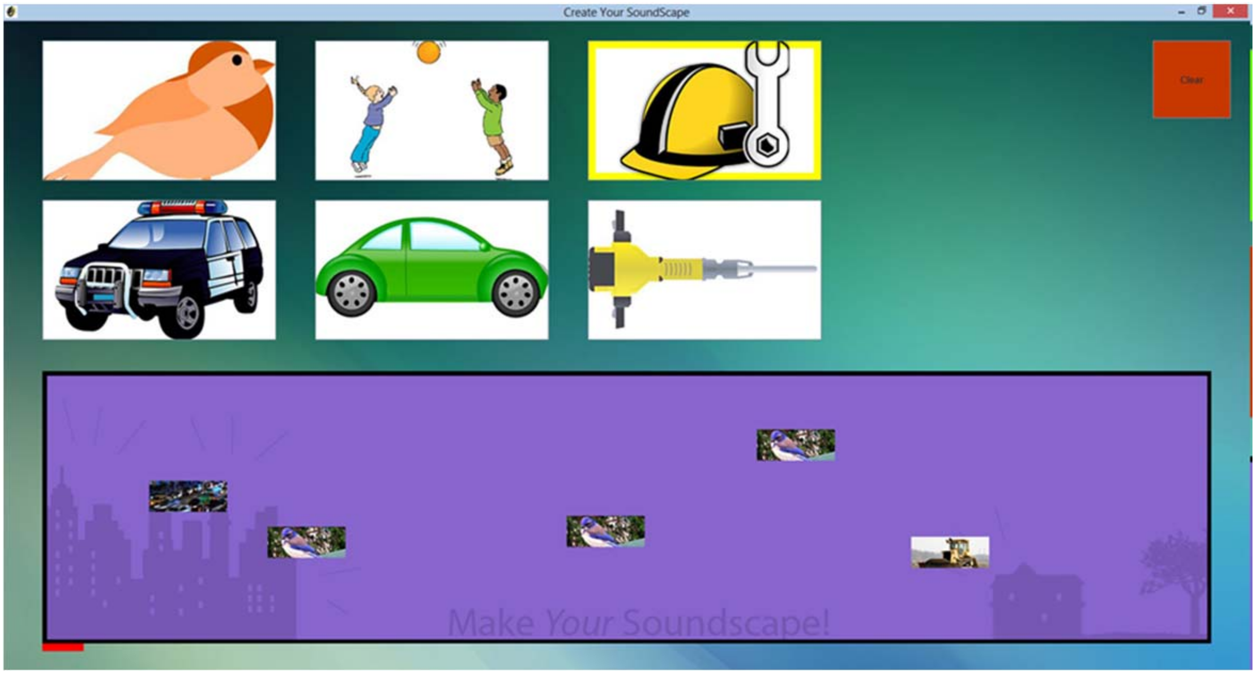
- First floor tested model
  - → This iteration was the first featuring functionality and capability resembling the final product and design philosophy
- Initial Layout Design
  - → Buttons for sound selection laid our across the top
  - → “Play” and “Clear” buttons aligned to the right
  - → Large Canvas placed along the bottom section where sounds could be placed for playback
- Creation of the Visual Design Theme
  - → Focus on mellow blues hues
  - → Cartoon images for the buttons
  - → Placed thumbnails retained the photorealistic images from Prototype 1
  - → A scrolling time bar would scroll across the bottom of the canvas when “Play” was pressed to give an indication of how much playback time remained
- Sound Profile
  - → All sounds were played for a standard duration of 15 seconds
- Initial Functionality
  - → Sounds could only be placed on the canvas by first clicking on the desired button and then clicking on the canvas where the sound should be placed
    - Buttons highlighted when selected

    - This was considered a placeholder function until the implementation of click and drag

→ The "Play" button could be pressed to play the placed sounds on the canvas

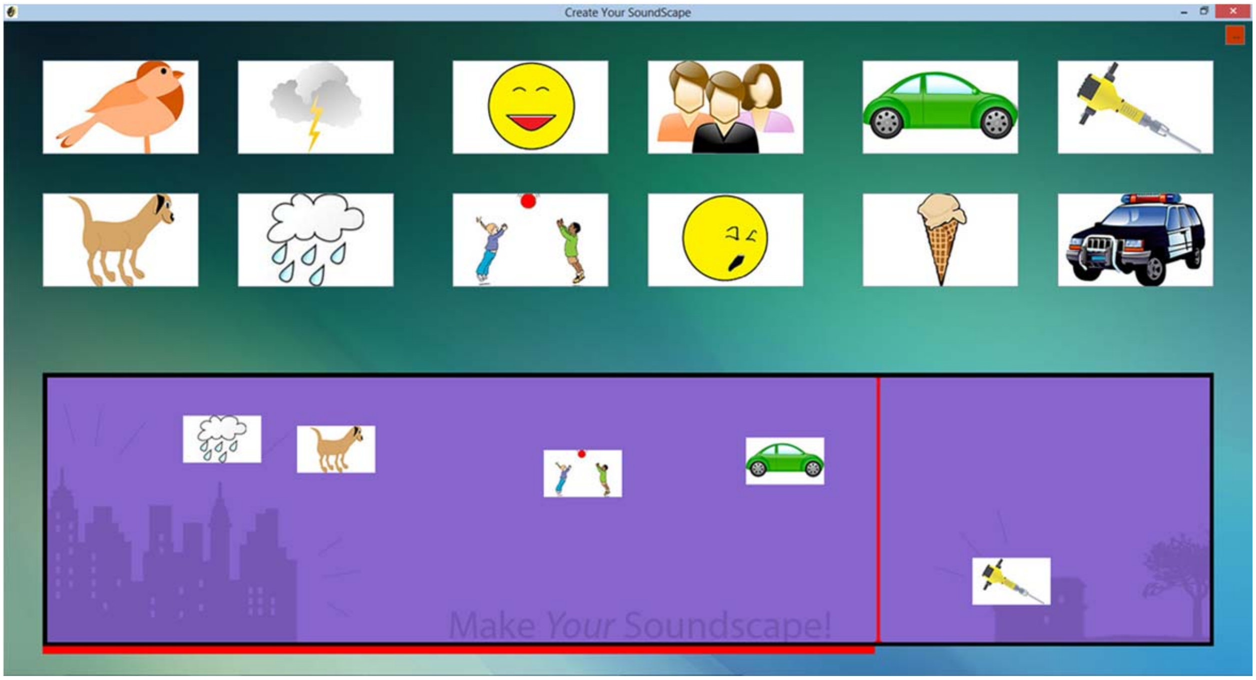
    - No action could be taken during sound playback
    - The playback lasted 15 seconds, the same length as the sound clip lengths

Prototype 3



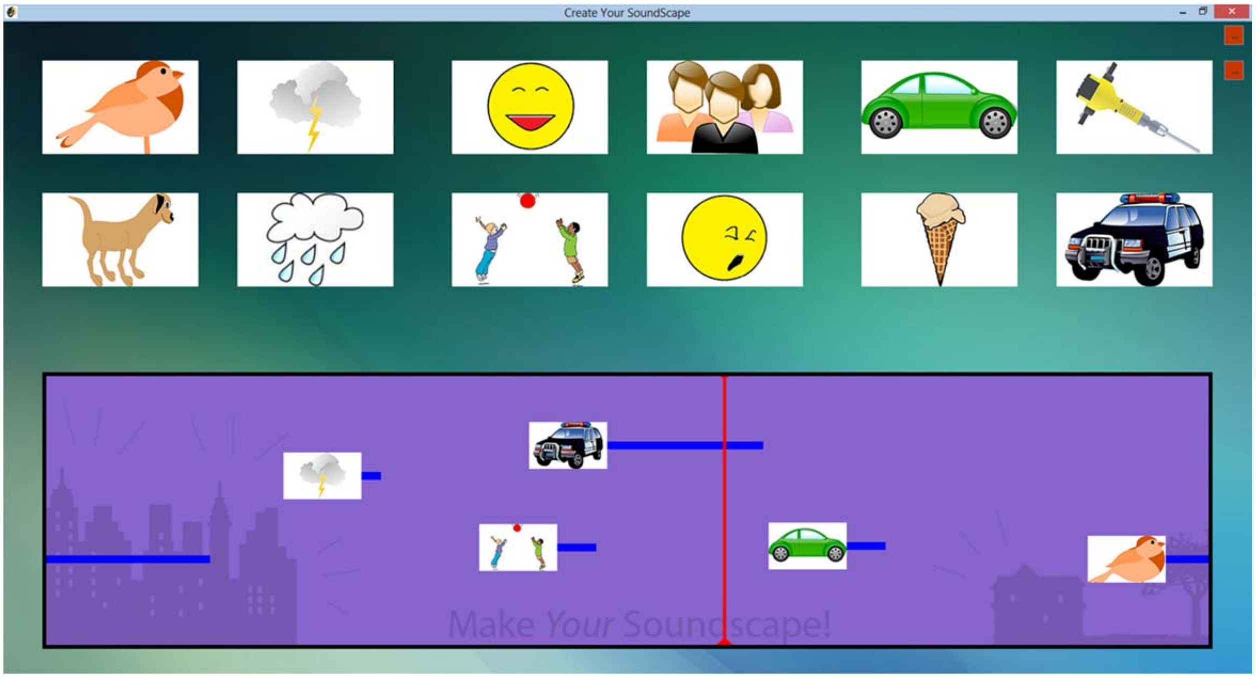
- Layout Changes
  - → The "Play" button was removed
  - → The "Clear" button was shrunk and moved into the top right
  - → Room was made available for more sound buttons for future iterations when more sounds were implemented
- Sound Playback
  - → Playback was now continuous and indicated by the scrolling time-bar at the bottom of the canvas
    - Playback was automatically looped, replacing the need for a "Play" button
    - Full functionality was allowed at all times, and placed sounds would be played as the time-bar reached them
    - One loop was 30 seconds long (each sound clip was 15 seconds)
- Functionality
  - → Deletion of individual sounds was added
    - Simply clicking on a placed sound would remove it from the canvas

Prototype 4



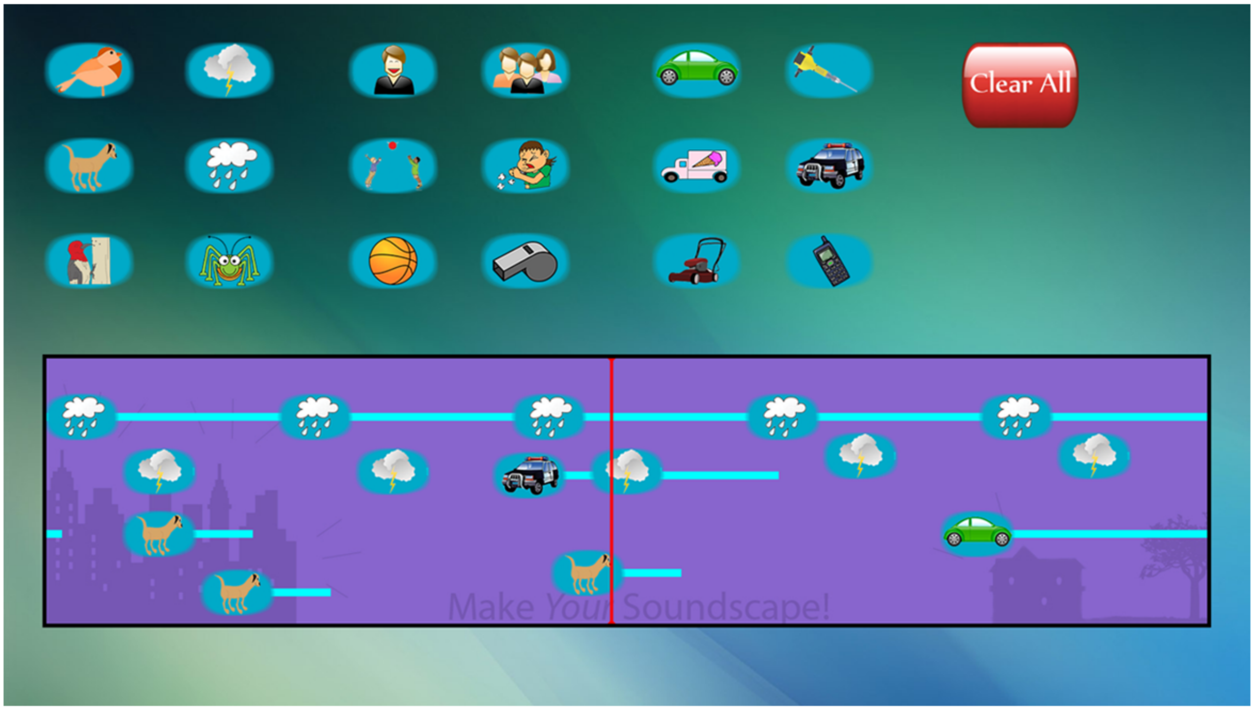
- Layout Changes
  - → The "Clear" button was shrunk further
  - → Buttons were shrunk in order to make room for a total of 12 sounds
  - → Buttons were organized by the type of sound they represented
    - Natural, Human, and Technological (Machine) sounds from left to right, respectively
- Visual Design Changes
  - → Placed thumbnails now visually matched their parent buttons to avoid discrepancies
  - → A vertical scrolling time-bar was added in addition to the original for more clarity
- Drag and Drop Functionality was added
  - → Sounds were now placed by clicking and dragging them to the desired location
  - → Drag and Drop sounds off the canvas in order to delete individual sounds
- Sound Changes
  - → Sounds no longer have a universal length
    - The length of the sound matches the expected length of the source

Prototype 5



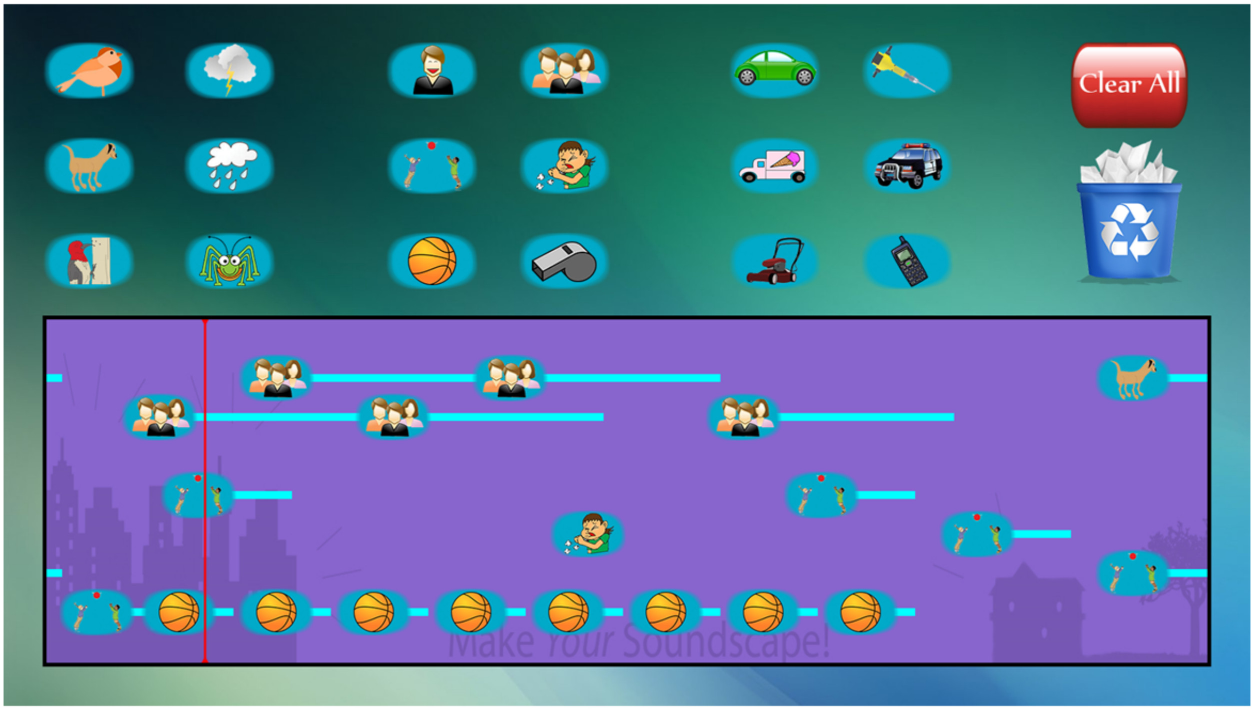
- Layout Changes
  - → A new preset button was added underneath the clear button
    - This button allowed us to quickly load a preset in order to populate the canvas during testing, as needed.
- Visual Design Changes
  - → Blue bars were added each thumbnail as a visual representation of the length of the sound
  - → The sound would start playing when the time-bar reached the thumbnail and would conclude playing when the time-bar reached the end of the new bar

Prototype 6



- Layout Changes
  - → Increased the number of sounds bringing the total to 18
    - Shrunk the button size accordingly to fit the layout
  - → Returned the “Clear” button to its original larger size and position
  - → Visually separated the sound categories to further differentiate them
- Visual Design Changes
  - → Overhauled the button and thumbnail images
    - The new design is consistent for all buttons, removing the visually jarring white space
  - → The images themselves were changed to be more clear as to the sound they represented
  - → Changed the color of the length bars to a lighter blue to be more consistent with our visual theme
- Sound Playback Changes
  - → When a sound is reached by the time-bar, the program randomly selects between two sounds
    - Each button has two sounds associated with it to add variety to each loop through the canvas

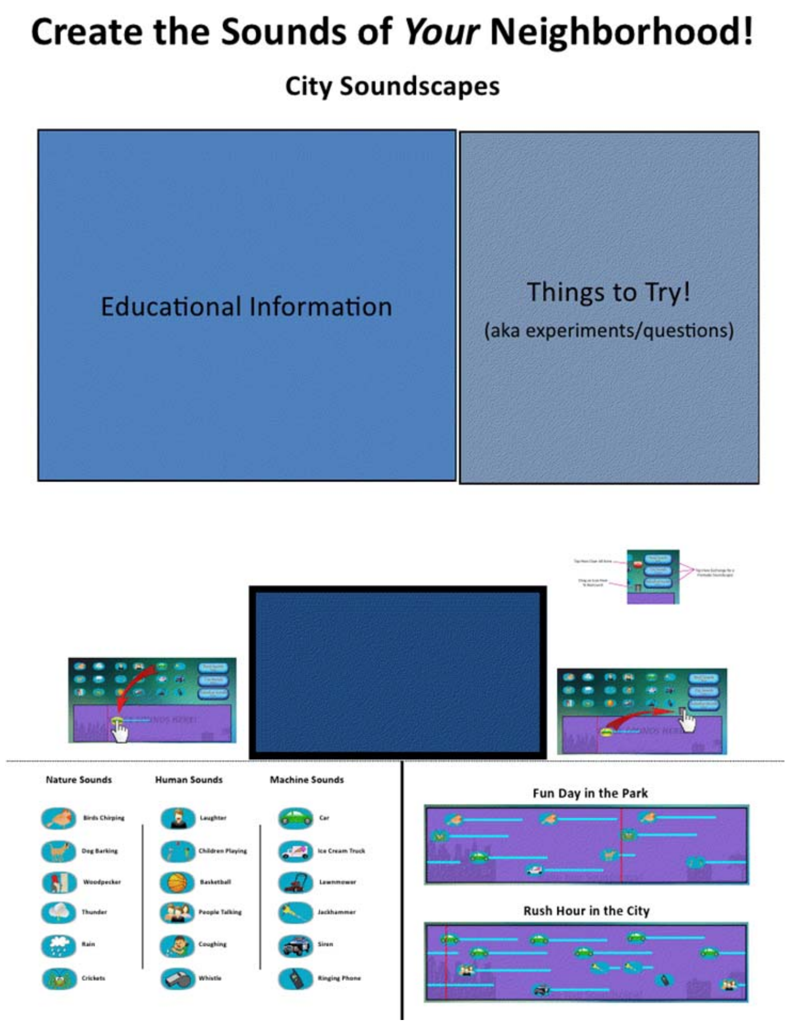
Prototype 7 / Floor Model



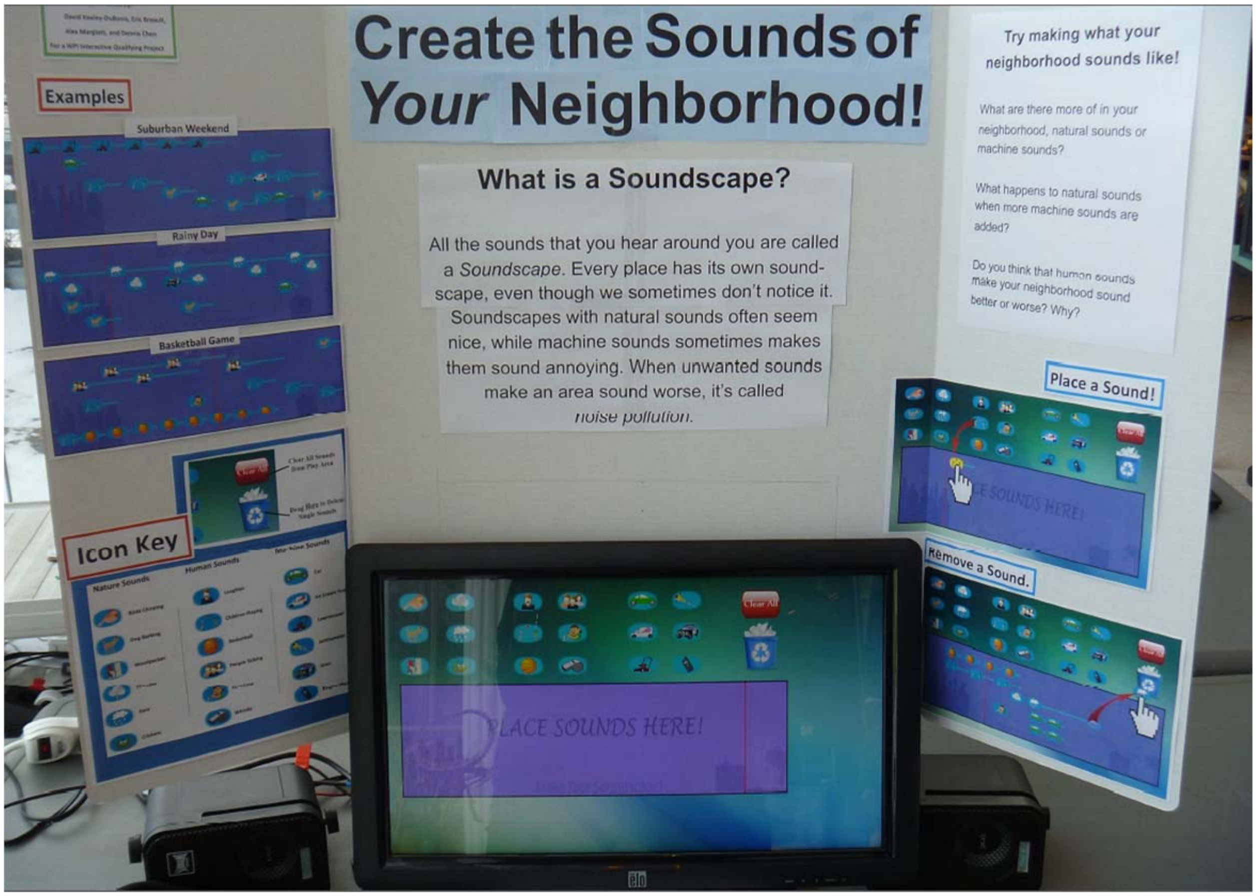
- Layout Changes
  - → Finalized the layout of the buttons at the top of the program
  - → Added a recycle bin icon to facilitate the deletion of individual sounds
  - → Increased the size of the canvas to the largest size possible to allow for the most room to place sounds
- Sound Playback Changes
  - → Many of the sounds themselves changed due to user feedback and replaced the old ones
- Functionality
  - → Deletion of individual sounds remained the same, however a recycle bin icon was added to make the process more intuitive and give users something to aim for

Appendix K: Education Panel Modification Summary

This Appendix details the evolution of the "Education Panel," the auxiliary information surrounding the interface designed to inform visitors of how to operate it and to educate them.



Early work on the Education Panel envisioned the screen in a central field, surrounded by all sorts of supporting material. Our testing had clearly shown that some sort of auxiliary material was necessary for proper operation of the exhibit. Included in the design were an icon key, two simple examples, and basic functionality demonstrations, including the use of how to recreate our preselected soundscapes.



Our final panel was somewhat reorganized, placing functional information closer to the bottom of the screen (at child height) and information at the top (at adult height). This grouping of directly relevant information together was intended to reduce confusion. We also removed instruction on how to use the preselected soundscapes (on recommendation of the EcoTarium) and emphasized the means of removing sounds, which appears in two different locations on the Panel.

Appendix L: XML Files

This Appendix is the XML file that was included with the final Prototype. It allows for easy changes to various parts of the program without the need of major changes to the back-end code. Changes beyond those allowed in this XML file would potentially require extensive editing of the code base.


```xml
<!--
 This xml file contains the file extentions for all files used by the SoundScape Exhibit
-->
<!--
 Editing this file will effect the appearence and sound of the interface but not its functionality.
-->
<!--
 Absolute addressing is not advised and will most likly result in an error.  After switching a file name, the file must be added to the
-->
<!--
 White space anywhere can result in an error. <AFile>file1.jpg</AFile> is good. <AFile> file1.jpg </AFile> will error.
-->
<!--
 All sounds should be have the .wav extention.  Other file types will work but are not expected and therefor treated as invalid file ext
-->
<!--
 All sound associated images should be png with opaque backgrounds
-->
<EcoTariumSoundScape>
  <!--
   The background image for the application.  1600 X 900 is ideal
  -->
  <Background>background.jpg</Background>
  <!--  The taskbar icon for the application. -->
  <Icon>icon.jpg</Icon>
  <!--
    The number of sounds that the application uses.  This number must be 18. To support a dynamic number of sounds will require
     aditional changes to code.
  -->
  <NumSounds>18</NumSounds>
  <!-- The image associated with the trash can -->
  <TrashIcon>trash can.png</TrashIcon>
  <!-- The clear button image -->
  <ClearImage>clear all_button.png</ClearImage>
  <!-- The set of sounds. -->
  <Sounds>
    <!-- The files associated with the first sound.  -->
    <Sound>
      <!--
       Name of the sound.  This currently has not effect on the SoundScape
      -->
      <Name>Bird</Name>
      <!--
       The number of the Sound.  The first sound should be 0 because all arrays index from 0
      -->
      <Num>0</Num>
      <!--
       The sound that is played is randomly selected from Clip1 and Clip2.  If both of these have
             the same sound file extention, there will be no randomnness.
      -->
      <Clip1>birds1.wav</Clip1>
      <Clip2>birds2.wav</Clip2>
      <!--
       The image that corresponds with the sound.  This is what is displayed to the users in the table that they may select from
      -->
      <TablePic>CartBird120x72.png</TablePic>
      <!-- The  -->
      <HighlightedPic>CartBird120x72Bordered.png</HighlightedPic>
      <PlayPic>CartBird120x72small.png</PlayPic>
      <!--
       The duration of the .wav files.  If a sound file is changed, this should be updated aswell
      -->
      <Duration>14</Duration>
    </Sound>
    <Sound>
      <Name>Thunder</Name>
      <Num>1</Num>
      <Clip1>thunder1.wav</Clip1>
      <Clip2>thunder2.wav</Clip2>
      <TablePic>CartThunder120x72.png</TablePic>
      <HighlightedPic>CartThunder120x72Bordered.png</HighlightedPic>
      <PlayPic>CartThunder120x72small.png</PlayPic>
      <Duration>4</Duration>
    </Sound>
    <Sound>
      <Name>Laughter</Name>
      <Num>2</Num>
      <Clip1>laughter1.wav</Clip1>
      <Clip2>laughter2.wav</Clip2>
      <TablePic>CartLaughing120x72.png</TablePic>
      <HighlightedPic>CartLaughing120x72Bordered.png</HighlightedPic>
      <PlayPic>CartLaughing120x72small.png</PlayPic>
      <Duration>4</Duration>
    </Sound>
    <Sound>
      <Name>Crowd</Name>
      <Num>3</Num>
      <Clip1>crowd1.wav</Clip1>
```

```xml
    <Clip2>crowd2.wav</Clip2>
    <TablePic>CartCrowd120x72.png</TablePic>
    <HighlightedPic>CartCrowd120x72Bordered.png</HighlightedPic>
    <PlayPic>CartCrowd120x72small.png</PlayPic>
    <Duration>13</Duration>
  </Sound>
  <Sound>
    <Name>Car</Name>
    <Num>4</Num>
    <Clip1>car1.wav</Clip1>
    <Clip2>car2.wav</Clip2>
    <TablePic>CartCar120x72.png</TablePic>
    <HighlightedPic>CartCar120x72Bordered.png</HighlightedPic>
    <PlayPic>CartCar120x72small.png</PlayPic>
    <Duration>5</Duration>
  </Sound>
  <Sound>
    <Name>Jackhammer</Name>
    <Num>5</Num>
    <Clip1>jackhammer1.wav</Clip1>
    <Clip2>jackhammer2.wav</Clip2>
    <TablePic>CartJackhammer120x72.png</TablePic>
    <HighlightedPic>CartJackhammer120x72Bordered.png</HighlightedPic>
    <PlayPic>CartJackhammer120x72small.png</PlayPic>
    <Duration>6</Duration>
  </Sound>
  <Sound>
    <Name>Dog</Name>
    <Num>6</Num>
    <Clip1>dogs1.wav</Clip1>
    <Clip2>dogs2.wav</Clip2>
    <TablePic>CartDog120x72.png</TablePic>
    <HighlightedPic>CartDog120x72Bordered.png</HighlightedPic>
    <PlayPic>CartDog120x72small.png</PlayPic>
    <Duration>7</Duration>
  </Sound>
  <Sound>
    <Name>Rain</Name>
    <Num>7</Num>
    <Clip1>rain1.wav</Clip1>
    <Clip2>rain1.wav</Clip2>
    <TablePic>CartRain120x72.png</TablePic>
    <HighlightedPic>CartRain120x72Bordered.png</HighlightedPic>
    <PlayPic>CartRain120x72small.png</PlayPic>
    <Duration>15</Duration>
  </Sound>
  <Sound>
    <Name>Children</Name>
    <Num>8</Num>
    <Clip1>children1.wav</Clip1>
    <Clip2>children2.wav</Clip2>
    <TablePic>CartChildren120x72.png</TablePic>
    <HighlightedPic>CartChildren120x72Bordered.png</HighlightedPic>
    <PlayPic>CartChildren120x72small.png</PlayPic>
    <Duration>7</Duration>
  </Sound>
  <Sound>
    <Name>Cough</Name>
    <Num>9</Num>
    <Clip1>cough1.wav</Clip1>
    <Clip2>cough2.wav</Clip2>
    <TablePic>CartCoughing120x72.png</TablePic>
    <HighlightedPic>CartCoughing120x72Bordered.png</HighlightedPic>
    <PlayPic>CartCoughing120x72small.png</PlayPic>
    <Duration>1</Duration>
  </Sound>
  <Sound>
    <Name>IceCreamTruck</Name>
    <Num>10</Num>
    <Clip1>icecreamtruck1.wav</Clip1>
    <Clip2>icecreamtruck2.wav</Clip2>
    <TablePic>CartIcecreamtruck120x72.png</TablePic>
    <HighlightedPic>CartIcecreamtruck120x72Bordered.png</HighlightedPic>
    <PlayPic>CartIcecreamtruck120x72small.png</PlayPic>
    <Duration>15</Duration>
  </Sound>
  <Sound>
    <Name>Siren</Name>
    <Num>11</Num>
    <Clip1>siren1.wav</Clip1>
    <Clip2>siren2.wav</Clip2>
    <TablePic>CartSiren120x72.png</TablePic>
    <HighlightedPic>CartSiren120x72Bordered.png</HighlightedPic>
    <PlayPic>CartSiren120x72small.png</PlayPic>
    <Duration>9</Duration>
  </Sound>
  <Sound>
    <Name>Woodpecker</Name>
    <Num>12</Num>
    <Clip1>woodpecker1.wav</Clip1>
    <Clip2>woodpecker2.wav</Clip2>
```

```xml
      <TablePic>CartWoodpecker120x72.png</TablePic>
      <HighlightedPic>CartWoodpecker120x72Bordered.png</HighlightedPic>
      <PlayPic>CartWoodpecker120x72small.png</PlayPic>
      <Duration>8</Duration>
    </Sound>
    <Sound>
      <Name>Crickets</Name>
      <Num>13</Num>
      <Clip1>cricket1.wav</Clip1>
      <Clip2>cricket2.wav</Clip2>
      <TablePic>CartCricket120x72.png</TablePic>
      <HighlightedPic>CartCricket120x72Bordered.png</HighlightedPic>
      <PlayPic>CartCricket120x72small.png</PlayPic>
      <Duration>15</Duration>
    </Sound>
    <Sound>
      <Name>Basketball</Name>
      <Num>14</Num>
      <Clip1>basketball1.wav</Clip1>
      <Clip2>basketball2.wav</Clip2>
      <TablePic>CartBasketball120x72.png</TablePic>
      <HighlightedPic>CartBasketball120x72Bordered.png</HighlightedPic>
      <PlayPic>CartBasketball120x72small.png</PlayPic>
      <Duration>5</Duration>
    </Sound>
    <Sound>
      <Name>Whistle</Name>
      <Num>15</Num>
      <Clip1>whistle1.wav</Clip1>
      <Clip2>whistle2.wav</Clip2>
      <TablePic>CartWhistle120x72.png</TablePic>
      <HighlightedPic>CartWhistle120x72Bordered.png</HighlightedPic>
      <PlayPic>CartWhistle120x72small.png</PlayPic>
      <Duration>1</Duration>
    </Sound>
    <Sound>
      <Name>Lawnmower</Name>
      <Num>16</Num>
      <Clip1>lawnmower1.wav</Clip1>
      <Clip2>lawnmower2.wav</Clip2>
      <TablePic>CartLawnmower120x72.png</TablePic>
      <HighlightedPic>CartLawnmower120x72Bordered.png</HighlightedPic>
      <PlayPic>CartLawnmower120x72small.png</PlayPic>
      <Duration>8</Duration>
    </Sound>
    <Sound>
      <Name>Cellphone</Name>
      <Num>17</Num>
      <Clip1>cell1.wav</Clip1>
      <Clip2>cell2.wav</Clip2>
      <TablePic>CartCellPhone120x72.png</TablePic>
      <HighlightedPic>CartCellPhone120x72Bordered.png</HighlightedPic>
      <PlayPic>CartCellPhone120x72small.png</PlayPic>
      <Duration>1</Duration>
    </Sound>
  </Sounds>
</EcoTariumSoundScape>
```

Appendix M: JavaDoc

This Appendix contains the complete documentation of the workings of our Java-based interface.

Class SoundscapeMain

java.lang.Object
  SoundscapeMain

```
public class SoundscapeMain
extends java.lang.Object
```

Version:

$Revision: 1.0 $

Author:

David Keeley-DeBonis

Constructor Summary

Constructors

Constructor and Description
`SoundscapeMain()`

Method Summary

Methods

Modifier and Type	Method and Description
`static void`	`main(java.lang.String[] args)` Creates an instance of the SoundSet based on the the file data.txt and passes this to an instance of SoundGUI.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

SoundscapeMain

```
public SoundscapeMain()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
```

Creates an instance of the SoundSet based on the the file data.txt and passes this to an instance of SoundGUI. SoundGUI is the interface that the user interacts with.

Parameters:

`args` -

soundscape.controller

Class AddSoundController

java.lang.Object
    soundscape.controller.AddSoundController

```
public class AddSoundController
extends java.lang.Object
```

Adds a sound to the playlist.

Author:

David Keeley-DeBonis

Constructor Summary

Constructors

Constructor and Description
`AddSoundController(SoundSet s, SoundGUI g, int index, int x, int y)` Creates an instance of a class that adds a sound to the playlist at the specified location if it is a valid location.

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`process()` Creates an instance of a class that adds a sound to the playlist at the specified location if it is a valid location.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

AddSoundController

```
public AddSoundController(SoundSet s,
                          SoundGUI g,
                          int index,
                          int x,
                          int y)
```

Creates an instance of a class that adds a sound to the playlist at the specified location if it is a valid location. If the location is invalid, this searches for a nearby valid location. If no location is found, no sound is added to the playlist

Parameters:

`s` - The current SoundSet used for this instance of the application

`g` - The current GUI displayed to the user

`index` - The index of the CitySound that is to be added

`x` - The x coordinate to attempt to add the sound to

`y` - The y coordinate to attempt to add the sound to

Method Detail

process

```
public void process()
```

Creates an instance of a class that adds a sound to the playlist at the specified location if it is a valid location. If the location is invalid, this searches for a nearby valid location. If no location is found, no sound is added to the playlist

soundscape.controller

Class ClearSelectedController

java.lang.Object
    soundscape.controller.ClearSelectedController

```
public class ClearSelectedController
extends java.lang.Object
```

Controller that clears the playlist

Author:

David Keeley-DeBonis

Constructor Summary

Constructors

Constructor and Description
`ClearSelectedController(SoundSet s, SoundGUI g)` Creates an instance of a controller that clears the playlist

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`process()` Removes all CitySounds from the playlist

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

ClearSelectedController

```
public ClearSelectedController(SoundSet s,
                               SoundGUI g)
```

Creates an instance of a controller that clears the playlist

Parameters:

s - The current SoundSet used for this instance of the application

g - The current GUI displayed to the user

Method Detail

process

```
public void process()
```

Removes all CitySounds from the playlist

soundscape.controller

Class EndMoveController

java.lang.Object
    soundscape.controller.EndMoveController

```
public class EndMoveController
extends java.lang.Object
```

The controller that handles end move conditions

Author:

David Keeley-DeBonis

Constructor Summary

Constructors

Constructor and Description
`EndMoveController(SoundSet s, SoundGUI g, int x, int y)` Creates an instance of the controller completed the movement of an CitySound in the playlist

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`process()` Moves the sound to the specified location.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

EndMoveController

```
public EndMoveController(SoundSet s,
                         SoundGUI g,
                         int x,
                         int y)
```

Creates an instance of the controller completed the movement of an CitySound in the playlist

Parameters:

- `s` - The current SoundSet used for this instance of the application
- `g` - The current GUI displayed to the user
- `x` - The potential x coordinate at which to move the CitySound to
- `y` - The potential y coordinate at which to move the CitySound to

Method Detail

process

```
public void process()
```

Moves the sound to the specified location. If the location is invalid, this searches for a nearby valid location. If no location is found, then the CitySound is returned to its old location in the playlist. If the sound is moved to a location off of the playzone, it is removed.

soundscape.controller

Class PlaySelectedController

java.lang.Object
    java.lang.Thread
        soundscape.controller.PlaySelectedController

All Implemented Interfaces:

java.lang.Runnable

```
public class PlaySelectedController
extends java.lang.Thread
```

The thread that controls the timers for the entire application. Only one instance of this class should be created

Author:

David Keeley-DeBonis

Nested Class Summary

Nested classes/interfaces inherited from class java.lang.Thread

`java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler`

Field Summary

Fields inherited from class java.lang.Thread

`MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY`

Constructor Summary

Constructors

Constructor and Description
`PlaySelectedController(SoundSet s, SoundGUI g)` Creates a non-expiring thread that maintains the playtime for the application.

Method Summary

Methods

Modifier and Type	Method and Description
void	**run**() Updates the timers of all graphic components of the GUI.

Methods inherited from class java.lang.Thread

activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

PlaySelectedController

```
public PlaySelectedController(SoundSet s,
                              SoundGUI g)
```

Creates a non-expiring thread that maintains the playtime for the application. the loop time for the playzone, and the monitors the total runtime.

Parameters:

s - The current SoundSet used for this instance of the application

g - The current GUI displayed to the user

Method Detail

run

```
public void run()
```

Updates the timers of all graphic components of the GUI. Monitors user interaction. If a user has not interacted with the interface for a certain amount of time, the GUI creates a new SoundSet with a random preset playlist.

Specified by:

`run` in interface `java.lang.Runnable`

Overrides:

`run` in class `java.lang.Thread`

soundscape.controller

Class StartAddController

java.lang.Object
  soundscape.controller.StartAddController

```
public class StartAddController
extends java.lang.Object
```

Controller that handles the adding of a CitySound to the playlist.

Author:

David Keeley-DeBonis

Constructor Summary

Constructors

Constructor and Description
`StartAddController(SoundSet s, SoundGUI g, int sindex)` Creates a controller that will handle the addition process of a single CitySound to the playlist

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`process()` Updates the state with the index of the CitySound that is being added.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

StartAddController

```
public StartAddController(SoundSet s,
                          SoundGUI g,
                          int sindex)
```

Creates a controller that will handle the addition process of a single CitySound to the playlist

Parameters:

`s` - The current SoundSet used for this instance of the application

`g` - The current GUI displayed to the user

`sindex` - The index of the CitySound to add from the SoundSet

Method Detail

process

```
public void process()
```

Updates the state with the index of the CitySound that is being added.

soundscape.controller

Class StartMoveController

java.lang.Object
    soundscape.controller.StartMoveController

```
public class StartMoveController
extends java.lang.Object
```

The controller that handles the initial movement process of a CitySound

Constructor Summary

Constructors

Constructor and Description
`StartMoveController(SoundSet s, SoundGUI g, int x, int y)`

Method Summary

Methods

Modifier and Type	Method and Description
`void`	`process()` If a CitySound is located at the x and y coordinates of a mouse click, the sound is removed from the playlist and a dragging animation is displayed until the mouse is released.

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

StartMoveController

```
public StartMoveController(SoundSet s,
```

```
                              SoundGUI g,
                              int x,
                              int y)
```

Parameters:

s - The current SoundSet used for this instance of the application

g - The current GUI displayed to the user

x - The x coordinate to move the CitySound to

y - The y coordinate to move the CitySound to

Method Detail

process

```
public void process()
```

If a CitySound is located at the x and y coordinates of a mouse click, the sound is removed from the playlist and a dragging animation is displayed until the mouse is released.

soundscape.model

Class CitySound

java.lang.Object
    java.lang.Thread
        soundscape.model.CitySound

All Implemented Interfaces:

java.lang.Runnable

```
public class CitySound
extends java.lang.Thread
```

This class contains all sound files and image files associated with a single sound. All sound files should be in the .wav format and not exceed 128Kb in size. The image files can either be in the .jpeg or .png format. A .png file format is recommended for visual appeal.

Author:

David Keeley-DeBonis

Nested Class Summary

Nested classes/interfaces inherited from class java.lang.Thread

`java.lang.Thread.State, java.lang.Thread.UncaughtExceptionHandler`

Field Summary

Fields inherited from class java.lang.Thread

`MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY`

Constructor Summary

Constructors

Constructor and Description
`CitySound(java.lang.String soundfile,`

```
java.lang.String soundfile2, java.lang.String imagefile,
java.lang.String imageborderedfile,
java.lang.String imageplayfile, int x, int y, int d)
```

Creates an instance of a CitySound

Method Summary

Methods

Modifier and Type	Method and Description
int	**getDuration**()
java.lang.String	**getImageborderedfile**()
java.lang.String	**getImagefile**()
java.lang.String	**getImageplayfile**()
java.lang.String	**getSoundfile**()
java.lang.String	**getSoundfile2**()
int	**getX**()
int	**getY**()
void	**run**() Plays the sound file of this sound.
void	**setX**(int x)
void	**setY**(int y)

Methods inherited from class java.lang.Thread

activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

CitySound

```
public CitySound(java.lang.String soundfile,
          java.lang.String soundfile2,
```

```
          java.lang.String imagefile,
          java.lang.String imageborderedfile,
          java.lang.String imageplayfile,
          int x,
          int y,
          int d)
```

Creates an instance of a CitySound

Parameters:

`soundfile` - The .wav file associated with the first sound clip for this CitySound

`soundfile2` - The .wav file associated with the second sound clip for this CitySound

`imagefile` - The default .png file for the CitySound

`imageborderedfile` - The .png file that indicates that the CitySound is selected

`imageplayfile` - The .png file that represents the CitySound once it has been placed in the play zone

`x` - The x coordinate of the CitySound if it has been placed in the play zone

`y` - The y coordinate of the CitySound if it has been placed in the play zone

`d` - The duration in seconds of the sound clips for the CitySound

Method Detail

run

```
public void run()
```

Plays the sound file of this sound. This is a thread. Attempting to execute this method again before the thread expires will result in an error

Specified by:

`run` in interface `java.lang.Runnable`

Overrides:

`run` in class `java.lang.Thread`

getImagefile

```
public java.lang.String getImagefile()
```

Returns:

the imagefile

getImageborderedfile

```
public java.lang.String getImageborderedfile()
```

Returns:

getX

```
public int getX()
```

Returns:

the x

setX

```
public void setX(int x)
```

Parameters:

`x` - the x to set

getY

```
public int getY()
```

Returns:

the y

setY

```
public void setY(int y)
```

Parameters:

`y` - the y to set

getSoundfile

```
public java.lang.String getSoundfile()
```

Returns:

the soundfile

getSoundfile2

```
public java.lang.String getSoundfile2()
```

Returns:

the soundfile2

getImageplayfile

```
public java.lang.String getImageplayfile()
```

Returns:

the imageplayfile

getDuration

```
public int getDuration()
```

Returns:

the duration

soundscape.model

Class SoundSet

java.lang.Object
    soundscape.model.SoundSet

```
public class SoundSet
extends java.lang.Object
```

Holds the State of the application.

Author:

David

Constructor Summary

Constructors

Constructor and Description
SoundSet(java.lang.String dataset, boolean verbose) Creates an instance of SoundSet.

Method Summary

Methods

Modifier and Type	Method and Description
int	**checkPlace**(int x, int y) Compares the x and y location of the sound to be placed against every sound in the playlist to ensure that there is no overlap
void	**generatePresetList**() Creates three preset playlists and stores these lists in a list of presets.
int	**getAddingselected**()
java.lang.String	**getBackground**()
java.lang.String	**getClearbutton**()
java.lang.String	**getIcon**()
int	**getInsertIndex**(int x) Compares the x coordinate a CitySound against the playlist and determines the proper insertion index.
CitySound	**getMovingselected**()
java.util.List<**CitySound**>	**getPlaylist**()
java.util.List<java.util.List<**CitySound**>>	**getPresetlist**()
CitySound[]	**getSounds**()
java.lang.String	**getTrashcan**()
boolean	**isCurrentlyAdding**()
boolean	**isCurrentlyMoving**()
boolean	**isVerbose**()
void	**playValidPlaylist**(double d) Plays all valid sounds in the playlist.
void	**regeneratePlayIndex**(int index) Creates a new instance of a CitySound
void	**regeneratePlayList**() Creates a new instance of every CitySound in a playlist.
void	**setAddingselected**(int addingselected)
void	**setBackground**(java.lang.String background)
void	**setClearbutton**(java.lang.String clearbutton)
void	**setCurrentlyAdding**(boolean currentlyAdding)
void	**setCurrentlyMoving**(boolean currentlyMoving)
void	**setIcon**(java.lang.String icon)
void	**setMovingselected**(**CitySound** movingselected)
void	**setPlaylist**(java.util.List<**CitySound**> playlist)

void	**setPresetlist** (java.util.List<java.util.List<**CitySound**>> presetlist)
void	**setTrashcan**(java.lang.String trashcan)
void	**setVerbose**(boolean verbose)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

SoundSet

```
public SoundSet(java.lang.String dataset,
        boolean verbose)
```

Creates an instance of SoundSet. Adds all sound clips and associated images to an array of CitySounds. The initial playlist is selected randomly form an initial set of preset options.

Parameters:

dataset - The name of the file that contains the file names of all images and sound clips used by the application.

verbose - For Debugging purposes. If verbose is true, print statements are enabled.

Method Detail

getSounds

```
public CitySound[] getSounds()
```

regeneratePlayIndex

```
public void regeneratePlayIndex(int index)
```

Creates a new instance of a CitySound

Parameters:

index - The index of the CitySound to be regenerated.

regeneratePlayList

```
public void regeneratePlayList()
```

Creates a new instance of every CitySound in a playlist.

playValidPlaylist

```
public void playValidPlaylist(double d)
```

Plays all valid sounds in the playlist. Validity is determined by d. If the x coordinate location of a sound in the playlist is within a certain distance of d, then the sound will play.

Parameters:

d - The current x value of the play bar. Use to determine what sounds should play

checkPlace

```
public int checkPlace(int x,
            int y)
```

Compares the x and y location of the sound to be placed against every sound in the playlist to ensure that there is no overlap

Parameters:

x - The x coordinate of the CitySound to be placed `y` - The y coordinate of the CitySound to be placed

Returns:

-1 if the desired location is available for placement, -2 if the desired location is unavailable for placement, positive index of correct location to insert into playlist if desired location is available for placement

getInsertIndex

```
public int getInsertIndex(int x)
```

Compares the x coordinate a CitySound against the playlist and determines the proper insertion index.

Parameters:

`x` - The x coordinate of the CitySound to compare with the x values of all CitySounds in the playlist

Returns:

The index to insert the CitySound at

generatePresetList

```
public void generatePresetList()
```

Creates three preset playlists and stores these lists in a list of presets. This creates the same presets every time.

isVerbose

```
public boolean isVerbose()
```

Returns:

the verbose

setVerbose

```
public void setVerbose(boolean verbose)
```

Parameters:

`verbose` - the verbose to set

getPlaylist

```
public java.util.List<CitySound> getPlaylist()
```

Returns:

the playlist

setPlaylist

```
public void setPlaylist(java.util.List<CitySound> playlist)
```

Parameters:

`playlist` - the playlist to set

getPresetlist

```
public java.util.List<java.util.List<CitySound>> getPresetlist()
```

Returns:

the presetlist

setPresetlist

```
public void setPresetlist(java.util.List<java.util.List<CitySound>> presetlist)
```

Parameters:

`presetlist` - the presetlist to set

isCurrentlyAdding

`public boolean isCurrentlyAdding()`

Returns:

the currentlyAdding

setCurrentlyAdding

`public void setCurrentlyAdding(boolean currentlyAdding)`

Parameters:

`currentlyAdding` - the currentlyAdding to set

getMovingselected

`public CitySound getMovingselected()`

Returns:

the movingselected

setMovingselected

`public void setMovingselected(CitySound movingselected)`

Parameters:

`movingselected` - the movingselected to set

isCurrentlyMoving

`public boolean isCurrentlyMoving()`

Returns:

the currentlyMoving

setCurrentlyMoving

`public void setCurrentlyMoving(boolean currentlyMoving)`

Parameters:

`currentlyMoving` - the currentlyMoving to set

getAddingselected

`public int getAddingselected()`

Returns:

the addingselected

setAddingselected

`public void setAddingselected(int addingselected)`

Parameters:

`addingselected` - the addingselected to set

getBackground

```
public java.lang.String getBackground()
```

Returns:

the background

setBackground

```
public void setBackground(java.lang.String background)
```

Parameters:

`background` - the background to set

getIcon

```
public java.lang.String getIcon()
```

Returns:

the icon

setIcon

```
public void setIcon(java.lang.String icon)
```

Parameters:

`icon` - the icon to set

getClearbutton

```
public java.lang.String getClearbutton()
```

Returns:

the clearbutton

setClearbutton

```
public void setClearbutton(java.lang.String clearbutton)
```

Parameters:

`clearbutton` - the clearbutton to set

getTrashcan

```
public java.lang.String getTrashcan()
```

Returns:

the trashcan

setTrashcan

```
public void setTrashcan(java.lang.String trashcan)
```

Parameters:

`trashcan` - the trashcan to set

soundscape.view

Class GraphicCanvas

java.lang.Object
    java.awt.Component
        java.awt.Container
            javax.swing.JComponent
                soundscape.view.GraphicCanvas

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable

---

```
public class GraphicCanvas
extends javax.swing.JComponent
```

The graphics component that displays the playzone to the user

Author:

David Keeley-Debonis

See Also:

Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.BaselineResizeBehavior`

Field Summary

Fields inherited from class javax.swing.JComponent

`TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED,`

WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

Constructors

Constructor and Description
GraphicCanvas(**SoundSet** set, **SoundGUI** gui) Create an instance of Graphics Canvas

Method Summary

Methods

Modifier and Type	Method and Description
void	**drawPlayZoneBorder**(java.awt.Graphics g) Paints the border for the playzone
java.awt.Point	**getDragLocation**()
java.awt.Rectangle	**getPlayzone**()
SoundSet	**getSet**()
void	**paint**(java.awt.Graphics g) The paint method for this component.
void	**paintAdd**(java.awt.Graphics g) The paint method that draws the drag add animation
void	**paintMove**(java.awt.Graphics g) The paint method that draws the drag move animation
void	**paintPlayZone**(java.awt.Graphics g) Paints the playzone graphics.
void	**paintSoundTable**(java.awt.Graphics g) Paint the interface buttons
void	**setBounds**(int x, int y, int width, int height) Sets the bounds of the component and updates the size and position of the playzone
void	**setPlayzone**(java.awt.Rectangle playzone)
void	**setSet**(**SoundSet** set)
void	**updateTime**(double t)

Updates the time.

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled, isValidateRoot, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, remove, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout, transferFocusDownCycle, validate

Methods inherited from class java.awt.Component

`action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, contains, createImage, createImage, createVolatileImage, createVolatileImage, dispatchEvent, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, remove, removeComponentListener, removeFocusListener, removeHierarchyBoundsListener, removeHierarchyListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeMouseWheelListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, resize, resize, setBounds, setComponentOrientation, setCursor, setDropTarget, setFocusable, setFocusTraversalKeysEnabled, setIgnoreRepaint, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus, transferFocusBackward, transferFocusUpCycle`

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Detail

GraphicCanvas

```
public GraphicCanvas(SoundSet set,
                     SoundGUI gui)
```

Create an instance of Graphics Canvas

Parameters:

`set` - The current SoundSet of the interface

`gui` - The current instance of the interface that this component belongs to

Method Detail

setBounds

```
public void setBounds(int x,
                      int y,
                      int width,
                      int height)
```

Sets the bounds of the component and updates the size and position of the playzone

Overrides:

`setBounds` in class `java.awt.Component`

Parameters:

`x` - The x position of this component

`y` - The y position of this component

`width` - The width of this component

`height` - The height of this component

paint

```
public void paint(java.awt.Graphics g)
```

The paint method for this component. Updates the current display.

Overrides:

`paint` in class `javax.swing.JComponent`

Parameters:

`g` - The graphics

paintAdd

```
public void paintAdd(java.awt.Graphics g)
```

The paint method that draws the drag add animation

Parameters:

`g` - The graphics

paintMove

```
public void paintMove(java.awt.Graphics g)
```

The paint method that draws the drag move animation

Parameters:

`g` - The graphics

paintSoundTable

```
public void paintSoundTable(java.awt.Graphics g)
```

Paint the interface buttons

Parameters:

`g` - The graphics

paintPlayZone

```
public void paintPlayZone(java.awt.Graphics g)
```

Paints the playzone graphics. This includes all placed sounds

Parameters:

`g` - The graphics

drawPlayZoneBorder

```
public void drawPlayZoneBorder(java.awt.Graphics g)
```

Paints the border for the playzone

Parameters:

`g` - The graphics

updateTime

```
public void updateTime(double t)
```

Updates the time. The playbar position is based on this. Repaints the canvas every time the time changes

Parameters:

`t` - The time in seconds that has passed in this loop

getDragLocation

```
public java.awt.Point getDragLocation()
```

Returns:

The current mouse location of the mouse with respect to the applications position on screen and the location of the component with respect to the application.

getSet

```
public SoundSet getSet()
```

Returns:

the set

setSet

```
public void setSet(SoundSet set)
```

Parameters:

`set` - the set to set

getPlayzone

```
public java.awt.Rectangle getPlayzone()
```

Returns:

the playzone

setPlayzone

```
public void setPlayzone(java.awt.Rectangle playzone)
```

Parameters:

`playzone` - the playzone to set

soundscape.view

Class ImagePanel

java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        soundscape.view.ImagePanel

All Implemented Interfaces:

java.awt.image.ImageObserver, java.awt.MenuContainer, java.io.Serializable

---

```
public class ImagePanel
extends javax.swing.JComponent
```

An panel with the desired background

Author:

David Keeley-DeBonis

See Also:

Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class javax.swing.JComponent

`javax.swing.JComponent.AccessibleJComponent`

Nested classes/interfaces inherited from class java.awt.Component

`java.awt.Component.BaselineResizeBehavior`

Field Summary

Fields inherited from class javax.swing.JComponent

`TOOL_TIP_TEXT_KEY, UNDEFINED_CONDITION, WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED,`

WHEN_IN_FOCUSED_WINDOW

Fields inherited from class java.awt.Component

BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Fields inherited from interface java.awt.image.ImageObserver

ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH

Constructor Summary

Constructors

Constructor and Description
ImagePanel(java.awt.Image image)

Method Summary

Methods inherited from class javax.swing.JComponent

addAncestorListener, addNotify, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, disable, enable, firePropertyChange, firePropertyChange, firePropertyChange, getAccessibleContext, getActionForKeyStroke, getActionMap, getAlignmentX, getAlignmentY, getAncestorListeners, getAutoscrolls, getBaseline, getBaselineResizeBehavior, getBorder, getBounds, getClientProperty, getComponentPopupMenu, getConditionForKeyStroke, getDebugGraphicsOptions, getDefaultLocale, getFontMetrics, getGraphics, getHeight, getInheritsPopupMenu, getInputMap, getInputMap, getInputVerifier, getInsets, getInsets, getListeners, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPopupLocation, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getToolTipText, getTopLevelAncestor, getTransferHandler, getUIClassID, getVerifyInputWhenFocusTarget, getVetoableChangeListeners, getVisibleRect, getWidth, getX, getY, grabFocus, isDoubleBuffered, isLightweightComponent, isManagingFocus, isOpaque, isOptimizedDrawingEnabled, isPaintingForPrint, isPaintingTile, isRequestFocusEnabled, isValidateRoot, paint, paintImmediately, paintImmediately, print, printAll, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, requestFocus, requestFocusInWindow, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setActionMap, setAlignmentX, setAlignmentY, setAutoscrolls, setBackground, setBorder, setComponentPopupMenu, setDebugGraphicsOptions, setDefaultLocale, setDoubleBuffered, setEnabled, setFocusTraversalKeys, setFont, setForeground, setInheritsPopupMenu, setInputMap, setInputVerifier, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setTransferHandler, setVerifyInputWhenFocusTarget, setVisible, unregisterKeyboardAction, update, updateUI

Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addPropertyChangeListener, addPropertyChangeListener, applyComponentOrientation, areFocusTraversalKeysSet, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getComponentZOrder, getContainerListeners, getFocusTraversalKeys, getFocusTraversalPolicy, getLayout, getMousePosition, insets, invalidate, isAncestorOf, isFocusCycleRoot, isFocusCycleRoot, isFocusTraversalPolicyProvider, isFocusTraversalPolicySet, layout, list, list, locate, minimumSize, paintComponents, preferredSize, printComponents, remove, remove, removeAll, removeContainerListener, setComponentZOrder, setFocusCycleRoot, setFocusTraversalPolicy, setFocusTraversalPolicyProvider, setLayout, transferFocusDownCycle, validate

Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addHierarchyBoundsListener, addHierarchyListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addMouseWheelListener, bounds, checkImage, checkImage, contains, createImage, createImage, createVolatileImage, createVolatileImage, dispatchEvent, enable, enableInputMethods, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, getBackground, getBounds, getColorModel, getComponentListeners, getComponentOrientation, getCursor, getDropTarget, getFocusCycleRootAncestor, getFocusListeners, getFocusTraversalKeysEnabled, getFont, getForeground, getGraphicsConfiguration, getHierarchyBoundsListeners, getHierarchyListeners, getIgnoreRepaint, getInputContext, getInputMethodListeners, getInputMethodRequests, getKeyListeners, getLocale, getLocation, getLocationOnScreen, getMouseListeners, getMouseMotionListeners, getMousePosition, getMouseWheelListeners, getName, getParent, getPeer, getPropertyChangeListeners, getPropertyChangeListeners, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, isBackgroundSet, isCursorSet, isDisplayable, isEnabled, isFocusable, isFocusOwner, isFocusTraversable, isFontSet, isForegroundSet, isLightweight, isMaximumSizeSet, isMinimumSizeSet, isPreferredSizeSet, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, remove, removeComponentListener, removeFocusListener,

```
removeHierarchyBoundsListener, removeHierarchyListener,
removeInputMethodListener, removeKeyListener,
removeMouseListener, removeMouseMotionListener,
removeMouseWheelListener, removePropertyChangeListener,
removePropertyChangeListener, repaint, repaint, repaint, resize,
resize, setBounds, setBounds, setComponentOrientation, setCursor,
setDropTarget, setFocusable, setFocusTraversalKeysEnabled,
setIgnoreRepaint, setLocale, setLocation, setLocation, setName,
setSize, setSize, show, show, size, toString, transferFocus,
transferFocusBackward, transferFocusUpCycle
```

Methods inherited from class java.lang.Object

```
equals, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

Constructor Detail

ImagePanel

```
public ImagePanel(java.awt.Image image)
```

Parameters:

`image` - The image to use as background