The use of ambient audio to increase safety and immersion in location-based games

by

John Jason Kurczak

A thesis submitted to the
Graduate Program in the School of Computing
in conformity with the requirements for
the degree of Master of Science

Queen’s University
Kingston, Ontario, Canada
January 2012

Copyright © John Jason Kurczak, 2012
Abstract

The purpose of this thesis is to propose an alternative type of interface for mobile software being used while walking or running.

Our work addresses the problem of visual user interfaces for mobile software being potentially unsafe for pedestrians, and not being very immersive when used for location-based games. In addition, location-based games and applications can be difficult to develop when directly interfacing with the sensors used to track the user’s location.

These problems need to be addressed because portable computing devices are becoming a popular tool for navigation, playing games, and accessing the internet while walking. This poses a safety problem for mobile users, who may be paying too much attention to their device to notice and react to hazards in their environment. The difficulty of developing location-based games and other location-aware applications may significantly hinder the prevalence of applications that explore new interaction techniques for ubiquitous computing.

We created the TREC toolkit to address the issues with tracking sensors while developing location-based games and applications. We have developed functional location-based applications with TREC to demonstrate the amount of work that can be saved by using this toolkit.
In order to have a safer and more immersive alternative to visual interfaces, we have developed ambient audio interfaces for use with mobile applications. Ambient audio uses continuous streams of sound over headphones to present information to mobile users without distracting them from walking safely.

In order to test the effectiveness of ambient audio, we ran a study to compare ambient audio with handheld visual interfaces in a location-based game. We compared players’ ability to safely navigate the environment, their sense of immersion in the game, and their performance at the in-game tasks.

We found that ambient audio was able to significantly increase players’ safety and sense of immersion compared to a visual interface, while players performed significantly better at the game tasks when using the visual interface. This makes ambient audio a legitimate alternative to visual interfaces for mobile users when safety and immersion are a priority.
Acknowledgments

I would like to thank, first and foremost, my supervisor Nicholas Graham for his infallible patience, encouragement, guidance, and bottomless good humour.

I would also like to thank Regan Mandryk and Claire Joly. Their work and advice was instrumental in completing the work that forms the core of this thesis.

Jonathan Segel has helped guide my work from its inception, and I thank him for the time and effort he’s put into helping us push the limits of knowledge forward.

The members of the EQUIS lab, past and present, have been more than supportive in their ongoing efforts to hasten the completion of my degree. I would especially like to thank Tad Stach for all of the advice he gave me while writing my thesis.

I must thank my parents, Catherine Grodesky and John Kurczak. They have guided me throughout my academic progress, and have never failed to support me emotionally (and financially, when necessary). Finally, I am grateful for my girlfriend Alex Kay, who has always been beside me while I have been riding the roller coaster of completing a graduate degree, and who I can never thank enough for being there with me.
# Table of Contents

Abstract i

Acknowledgments iii

Table of Contents iv

List of Tables vi

List of Figures vii

Chapter 1:

**Introduction** ........................................ 1

1.1 Problems ............................................. 4

1.2 Solutions .............................................. 6

1.3 Hypotheses ........................................... 10

1.4 Evaluation ............................................ 11

1.5 Summary of contributions .............................. 13

1.6 Organization of thesis ................................ 14

Chapter 2:

**Related Work** ........................................ 16

2.1 Designing interfaces for use while mobile ............... 17

2.2 Simplifying the use of sensors in developing location-based games 27

2.3 Conclusion ............................................. 36

Chapter 3:

**Designing Ambient Audio Games** ......................... 38

3.1 Ambient audio .......................................... 38

3.2 Examples of ambient audio applications .................. 39

3.3 Ambient audio design guidelines ........................ 45

3.4 Implementation considerations .......................... 46

3.5 Potential limitations ................................... 50
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>Conclusion</td>
<td>51</td>
</tr>
<tr>
<td>Chapter 4:</td>
<td>Evaluation of Ambient Audio Interfaces</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Hypotheses</td>
<td>53</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental method</td>
<td>54</td>
</tr>
<tr>
<td>4.3</td>
<td>Results</td>
<td>63</td>
</tr>
<tr>
<td>4.4</td>
<td>Discussion</td>
<td>67</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>73</td>
</tr>
<tr>
<td>Chapter 5:</td>
<td>TREC Toolkit for Ubiquitous Games</td>
<td>75</td>
</tr>
<tr>
<td>5.1</td>
<td>Difficulties in developing location-based applications</td>
<td>76</td>
</tr>
<tr>
<td>5.2</td>
<td>The TREC three-layer architecture</td>
<td>78</td>
</tr>
<tr>
<td>5.3</td>
<td>Addressing the development difficulties with TREC</td>
<td>80</td>
</tr>
<tr>
<td>5.4</td>
<td>Applying TREC</td>
<td>81</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>86</td>
</tr>
<tr>
<td>Chapter 6:</td>
<td>Summary and Conclusions</td>
<td>87</td>
</tr>
<tr>
<td>6.1</td>
<td>Summary</td>
<td>87</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>89</td>
</tr>
<tr>
<td>6.3</td>
<td>Conclusion</td>
<td>90</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>Appendix A:</td>
<td>Custom Questionnaires</td>
<td>97</td>
</tr>
<tr>
<td>Appendix B:</td>
<td>Expertise Questionnaire</td>
<td>100</td>
</tr>
<tr>
<td>Appendix C:</td>
<td>Research Ethics Board Approval</td>
<td>102</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Performance measures between two conditions . . . . . . . . . . . . . 64
4.2 Performance measures between two groups . . . . . . . . . . . . . . 64
4.3 Presence measures between two conditions . . . . . . . . . . . . . . 65
4.4 Player preferences between two conditions (7-point Likert scale) . . . 67
# List of Figures

1.1 Using a visual display while mobile ........................................... 2
1.2 A prototype ambient audio system ........................................... 8
1.3 The indoor experimental setup for testing Growl Patrol ............... 12

2.1 A layered device hierarchy in VRPN ......................................... 32
2.2 The OpenInterface Workbench GUI ......................................... 33

3.1 Using the tourist application ................................................... 40
3.2 Overhead view ........................................................................ 41
3.3 Growl patrol gameplay: players hear animals overlaid on the physical world ........................................................................... 44
3.4 The equipment used to track a user’s location and orientation for an ambient audio interface .................................................. 49

4.1 Indoor experimental equipment for the audio condition ............... 55
4.2 Indoor experimental equipment for the visual condition ............... 56
4.3 An example of the radar map used in the visual condition. This map is shown on a display on the player’s right side. ....................... 57
4.4 A small rectangular radar map is shown in the upper right corner of the game Command and Conquer ........................................ 59

5.1 The TREC three-layer architecture ........................................... 79
5.2 Adding a new sensor to the TREC hierarchy ............................... 84

A.1 Ambient audio condition questionnaire ..................................... 98
A.2 Visual condition questionnaire ............................................... 99

B.1 Expertise questionnaire for all study participants ...................... 101

C.1 GREB approval for ambient audio study .................................... 103
Chapter 1

Introduction

Mobile phones and other portable computing devices are often used by people who are walking and driving, but are not always designed to be used this way. Applications for these mobile devices frequently require precise manual input and visual output that requires the user to focus on the device when using it, making it hard to attend to the environment around him. A user who is ignoring his environment may put himself at risk of an accident, as illustrated in figure 1.1. Schwebel et al. tested pedestrian safety while crossing streets, and found that pedestrians attempting to send SMS messages while crossing a virtual street were significantly more likely to be hit by a passing car than undistracted pedestrians [22].

Since a pedestrian’s attention to his environment can be so important to his safety, an application that actually necessitates walking around during use could put its users at risk. Location-based games are a fairly novel form of mobile application that make this demand, where control of the game depends on the player’s movement in the real world. Using a graphical interface for this type of game might be difficult, either requiring the player to stop for feedback or encouraging unsafe walking behaviour.
Ambient interfaces are designed to demand less attention than traditional graphical interfaces, allowing the user to perform a primary task while monitoring his application in the background. The ambient interface attempts to accomplish this by providing feedback at the periphery of a user’s attention, as described in the Wisneski, Ishii, et al. paper *Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information* [33]. Designing location-based games and other mobile applications to include ambient interfaces might permit safer use for pedestrians.

There is a second problem for players attempting to use a visual display for location-based games: the small handheld display might limit a player’s sense of immersion in the world. Players looking at a small map of locations and having to
imagine these places and entities overlaid on the real world may not feel very involved and immersed in the game world, especially when constantly glancing away from the game interface to focus on the real world.

In addition to facing challenges in providing feedback to users, mobile applications that need the user’s position may be somewhat difficult to develop. They gather the user’s position through a number of sensors (usually embedded in the mobile device). If the sensor data is noisy or inaccurate, the data can be analyzed from multiple sources and combined or “fused” to provide a more responsive and accurate view of user behaviour. The difficulty of gathering this sensor data and fusing it can provide significant barriers to a developer who must attend to all the other aspects of creating a fun game.

We have addressed both the interface problem and the sensor problem for mobile applications. We propose ambient audio interfaces as a way to effectively communicate information to mobile users safely, using streams of ambient sounds to convey information in a way that is not as distracting as visual interfaces can be. These ambient audio streams might also increase the immersion of mobile users, by being always on and appearing to emanate from the real world. We also propose our TREC sensor input toolkit as a way to make it easier to use sensor data when developing location-based games, by making sensor fusion algorithms easier to implement and making them available to any TREC-based application without modifications.

This thesis also presents applications designed using ambient audio and TREC, and describes an experiment comparing an ambient audio interface and a visual interface for a location-based game called Growl Patrol. In the experiment, ambient audio was shown to increase a player’s sense of immersion and safety when compared
to players using the visual interface, but these benefits came at the cost of reduced performance of tasks in the game.

1.1 Problems

Location-based games require a player to walk around the real world to accomplish goals in the game. Walking to a specific location can achieve the goal, or set off a game that is played on the device using the standard interface. An example of this game is PiNiZoRo [26], a game where players must travel to the next destination shown on a “treasure map” while being tracked by the mobile phone’s GPS. At this destination, a player triggers a puzzle game on the phone’s touchscreen that must be completed before he can move on.

This thesis addresses problems related to the design and development of applications meant to be used by pedestrians. The first problem is the potential lack of immersion that game players experience when using a handheld visual display to play location-based games, and the second is the inability for pedestrians to safely use applications that require handheld visual displays. The final problem is the difficulty for developers of using tracking sensor data to control location-based games or applications.

1.1.1 Interfaces for mobile applications

Mobile devices can be used in situations that software designers may not have experience accommodating: situations where the user must constantly ignore the main interface. In cases where the user is playing an ubiquitous game that requires rapid
CHAPTER 1. INTRODUCTION

movement within the environment, looking at the handheld screen may distract from the user’s goal of reaching a landmark or navigating through a crowd.

The use of a small handheld screen for visual output requires the player to briefly ignore his environment when he desires feedback, which may impede the user’s ability to navigate and complete his task.

An additional problem occurs when using small handheld screens to display a map that shows virtual entities’ locations in the real world. The display adds a layer of indirection between seeing an entity’s location on the map and being able to mentally calculate its real-world location. It may require additional conscious effort to perform this mapping from virtual world to real world, which could make it harder for players to feel immersed, “perceiving oneself to be enveloped by, included in, and interacting with an environment that provides a continuous stream of stimuli and experiences” [34].

1.1.2 Sensor limitations

The sensors being used to give mobile devices their position and orientation in the world can be noisy and inaccurate - for example, Micro Electro Mechanical System (MEMS) gyroscope sensors used to measure rotational rate have an upper limit to their measurement range, which in some sensors can be reached during normal use. For example, the Melexis MLX90609-E2 gyroscope we use for head tracking has a measurement range of ±150°/s, which we have often exceeded with rapid head movements during operation.

In other sensors, such as a digital compass, there can be significant fluctuations in the signal that would provide erratic head tracking behaviour if used directly. The
data from multiple sensors can be combined to reduce the errors in the location and orientation data, but doing so can be difficult and can require knowledge of sensor limitations and sources of outside interference.

In addition, writing code to interact with specific sensors at a low level, to handle the change between different sets of sensors, and to handle missing sensors consumes developer time that could be better spent building the application itself.

1.2 Solutions

We present two potential solutions that attempt to address the aforementioned problems with mobile applications. Ambient audio is proposed as a way to provide feedback to pedestrians in a more immersive and less distracting manner than handheld visual interfaces.

The TREC toolkit is presented as a tool for facilitating the use of location sensors while developing location-based games and applications, by reducing the work necessary to a) access sensor data and b) increase the data’s accuracy.

1.2.1 Ambient audio

To address the immersion and safety problems of handheld visual displays, we developed a solution that we call an ambient audio interface. An ambient audio interface uses continuous streams of sound to represent virtual entities such as a homing beacon to walk towards, a small animal to chase, or an enemy in pursuit.

We use spatial audio signal processing to modify the sound streams with the aural cues that humans use to perceive sound direction. Played over headphones, the sound
CHAPTER 1. INTRODUCTION

streams can then appear to emanate from any direction around a user. These sounds can be used to overlay information on real-world objects, or to represent completely virtual objects.

We argue that this type of interface is especially useful for applications that communicate a real-world location to a user, such as for navigation, to find friends, or to play location-based games. Certain mobile applications may also be able to use sound stream locations as metaphors to communicate non-locational information to the user, like using the clockwise rotation of a “ticking” sound to indicate the time remaining before an appointment.

Ambient audio may not be suitable for many traditional forms of application, however, such as web browsers or email, that necessarily must use text or images to communicate. Applications that display very complex or multifaceted data to the user may not be suitable either, since we have found that users cannot keep track of large numbers of sounds simultaneously. Ambient audio is intended for mobile applications that must keep a user continuously informed of data while moving in a busy environment.

We use headphones to present ambient audio, since they offer privacy, are commonly used and socially acceptable in public, and transmit an isolated channel of sound to each ear (which simplifies the processing that is necessary to hear spatial audio). A user with headphones and a mobile device can walk around hearing sounds seemingly embedded in the environment. To ensure that sound streams are placed in the correct directions, we use a sensor-based head tracking system that updates the sound environment to match the player’s current “perspective” of the real world. In
CHAPTER 1. INTRODUCTION

Figure 1.2: A prototype ambient audio system

Figure 1.2, a user is holding a mobile device and listening to ambient audio with earbud headphones, showing the current version of our prototype ambient audio system. The user is wearing a hat with a digital compass and gyroscope attached to track the user’s head movements, and the mobile device is using a GPS tracker to calculate the user’s location.

We developed two applications using this interface, Noisy Planet [11] and Growl Patrol [10], which will be described in detail in chapter three. To summarize, Growl Patrol is a location-based game played in a pre-defined outdoor area that gives players the chance to catch escaped pets (dogs, cats, and birds) and bring them back to the pet store. These animals independently roam around the game area, and the player
must identify the direction and distance of the barking, meowing, or tweeting to find an animal’s location. The player must get close to an animal in order to register a catch, at which point he may bring the animal back to the pet shop for one point. There is, however, a tiger hunting in the area who will attempt to steal pets from the player on his way to the shop, making the trip back a little more exciting.

Noisy Planet is a location-based application for navigating and exploring cities, able to provide an ambient audio summary of nearby attractions in addition to more detailed spoken audio and visual interfaces. In Noisy Planet, every point of interest that has been selected for display emits a continuous sound from its real-world location, played back over the headphones. Each sound identifies the place that it represents, such as flipping pages for a bookstore or dripping coffee and a whistling kettle for a cafe. This application can guide users to a destination, or can enable the discovery of points of interest that are out of the way and not immediately apparent to someone unfamiliar with the area.

1.2.2 TREC toolkit

The TREC toolkit attempts to solve some of the problems with developing applications that require knowledge of a person’s position and orientation. Specifically, it addresses five problems that can occur when trying to access sensors directly: 1) noisy or inaccurate sensor data will harm application behaviour, 2) sensor readings might be in a format that needs to be converted for use in the application, 3) differences in sensor interfaces might require recoding an application when changing sensors, 4) manually choosing which sensors to use in advance may cause applications to fail if different sensors are available, and 5) existing code can be difficult to extend to
support new sensors or new algorithms that reduce the errors in sensor data. TREC uses a three-layer architecture that can address these problems through abstraction. This architecture, and examples of how to apply TREC in application development are further described in chapter five.

1.3 Hypotheses

We tested four hypotheses to determine how well ambient audio works as a replacement for handheld visual interfaces. First, we hypothesize that ambient audio can increase the game player’s perceived level of immersion in the game compared to a visual display, thanks to the direct mapping of sound locations to real-world locations. Second, we hypothesize that ambient audio leads to increased safety over a visual display, due to ambient audio not requiring the user’s visual attention, as well as the potential reduction in overall attention required to interpret the interface.

The third hypothesis is that ambient audio leads to reduced performance in location-based games compared to a visual display, since the audio cues being used to display entity locations may prove more difficult to interpret accurately than judging precise locations with a visual map. Finally, we hypothesize that ambient audio’s positive effects will be more pronounced with players who do not have expertise using secondary map displays to navigate an area, such as the radar-map displays used in many video games.

We also have one hypothesis regarding the TREC toolkit. We hypothesize that TREC can reduce the time a developer must spend to handle sensor input when building a location-based game. In order to test this hypothesis, we have built applications using TREC to demonstrate whether TREC is functional and can save
developer time in certain situations. The next step to fully evaluating our hypothesis would involve rigorous evaluation of the quality of TREC’s sensor tracking, its compatibility with sensors, and its compatibility with noise and error reducing algorithms when compared to other sensor frameworks.

1.4 Evaluation

We evaluated ambient audio by comparing it with a handheld visual interface, implemented in a location-based game. The player’s immersion was measured using the Witmer and Singer Presence questionnaire, and was found to be significantly higher for ambient audio than the visual interface. We measured safety by the number of collisions with obstacles that a player experienced while playing, and ambient audio was found to induce significantly fewer collisions during the game.

We measured the player’s performance by his score attained over the course of a game session, where higher scores indicated increased performance. In this case, players using the visual interface tended to score significantly higher than players using ambient audio. Finally, players were given a questionnaire to assess their level of experience with games and radar-map displays. We found that expert radar-map users experienced the same benefit from the ambient audio interface as novice users did.

1.4.1 Growl Patrol experiment

In order to test our proposed ambient audio solution for mobile devices, an experiment was run to compare the effectiveness of an ambient audio interface against a handheld
visual equivalent. The test was run using a location-based game based on Growl Patrol, where players must attempt to catch as many runaway animals as possible in 5 minutes. In one condition of the experiment, the game was played using a visual interface showing a map, and in the other condition the game was played using ambient audio to display the locations of animals in the park.

In order to make the test more controlled and to reduce the safety risks to participants, the experimental trials were performed in a virtual park environment that players navigated using a recumbent bicycle and joystick, as shown in figure 1.3.

Analysis of the results showed that players did tend to achieve higher game scores in the visual condition. In the audio condition, an average of 8.33 animals were caught
per session, while in the visual condition, participants caught an average of 11.25.

Immersion scores were based on the Witmer and Singer Presence questionnaire, which showed that players felt that the audio condition was more immersive. In the audio condition, a mean score of 44.92 out of 56 was given, while in the visual condition, a mean score of 40.54 out of 56 was given.

The safety measure (number of collisions with obstacles) showed that the ambient audio interface was significantly safer. In the audio condition, there was an average of 10.13 collisions per session, while there was an average of 27.42 collisions in the visual condition.

Finally, both novice and expert participants reported experiencing a higher sense of presence in the audio condition, as well as finding it easier to avoid obstacles in the audio condition. Experts did, however, judge the visual condition easier to play than the novices did.

Overall, the first three hypotheses were confirmed, while the fourth was shown to be false. Safety and immersion are both higher with ambient audio, while radar map expertise did not hinder ambient audio’s benefits, indicating that ambient audio interfaces are a very good candidate for use in location-based games. These results should be followed up with further experimentation in the physical world to determine whether they hold as strongly in real conditions.

1.5 Summary of contributions

The work presented in this thesis makes two primary contributions to the area of mobile application development. The first contribution is the evaluation of ambient audio interfaces. We performed an experiment to directly compare how an ambient
audio interface can affect a mobile user’s task performance, immersion, and safety when compared to a handheld visual interface. This experiment demonstrated that ambient audio can provide better immersion and safety than a handheld visual interface for certain types of applications.

The second contribution is the development of the TREC toolkit. We designed TREC to reduce the development time needed to develop a location-based game or application. TREC addresses the five major problems we have identified in mobile application development, using the multiple levels of abstraction provided by its three-layer architecture. We have demonstrated TREC’s ability to save developer time by creating location-based applications with TREC.

1.6 Organization of thesis

This thesis presents two related solutions to problems encountered when developing and using location-based mobile applications. First, it can be difficult to take advantage of location and orientation sensors when developing a mobile game. Providing the user with a safe and unobtrusive display while he is on the move can also be difficult to achieve when using a handheld visual interface.

In chapter two, research related to this topic is described, including experiences with using ambient audio-like systems while mobile, and research examining the effectiveness of audio interfaces. There is also a summary of the many existing frameworks for handling sensor input to mobile applications, and how these solutions relate to TREC and its stated goals.

Chapter three describes in detail our ambient audio solution to mobile application interfaces, enumerating the features that make it potentially suitable for mobile use.
Two applications that were created using this type of interface will also be described, showing how ambient audio may be applied to real applications.

Chapter four presents an experiment that was run to evaluate the effectiveness of ambient audio in a mobile setting using a location-based game, when compared to the same game being played using a handheld visual interface. The results from this experiment are given, as well as discussion regarding the recommended uses for ambient audio based on these results.

Chapter five details how the TREC toolkit attempts to solve the development issues involved in using location sensors in mobile games. A thorough explanation of the unique TREC three layer architecture is given, in addition to demonstrations of how TREC can help both application developers as well as developers who are extending the toolkit.

Finally, chapter six discusses of the issues surrounding mobile development, giving an indication of the suitability of these solutions for current practical use, as well as looking to future developments.
Chapter 2

Related Work

The work in this thesis addresses two problems for software for mobile devices: handheld visual displays can be difficult to use while walking, and it can be time-consuming to develop an application that tracks a user’s location using tracking sensors.

The first problem involves the usability of mobile devices like smartphones. These devices can be difficult to use while walking because they often require the user to focus his visual attention on the handheld screen. This can be distracting to a pedestrian who must devote some of his visual attention to his environment in order to safely navigate.

The first part of this chapter addresses the problem of creating interfaces that are not distracting to a user who is trying to visually negotiate his environment while on foot. This includes research investigating the most effective ways to design ambient interfaces, how to use audio interfaces with location-based games and mobile applications, and testing the effectiveness of audio interfaces.

Developing mobile applications can be time-consuming if the applications make use of the user’s physical location and orientation. This location data can help to
build novel experiences or gameplay, but interfacing with the location and orientation sensors can be difficult for programmers. In this chapter we identify five issues that can make development more time consuming for these developers, including problems such as noisy tracking data, incorrect data formats, and inconsistencies between sensor interfaces.

The second part of this chapter addresses ways of making the development of location-based games and applications easier. We discuss how the five sensor interfacing problems can be addressed, in order to simplify the process of determining a user’s location and orientation using sensor input. Specifically, we discuss the use of sensor fusion to increase the overall accuracy of a calculated position or orientation. We also discuss the use of frameworks to read and interpret incoming sensor data, delivering the final location data directly to the developer.

2.1 Designing interfaces for use while mobile

Users who are walking in unpredictable environments risk having accidents if they are not able to pay attention to what is going on around them. Therefore, minimizing the distraction caused by the device interface is important when designing an application for mobile use. The following three areas of research contribute some insight into this challenge.

First is the research on ambient interfaces. Such interfaces help maintain awareness of some activity while performing some other task. An appropriately designed ambient interface for mobile applications could keep the user informed of the application’s state while minimizing the distraction from his primary task of walking safely.
Next is investigation of the design of audio interfaces for mobile use. Since navigating a potentially dangerous environment is primarily a visual task for most users, using a different modality for the application interface can help the user to keep his visual attention focused on his environment. Some mobile applications have been implemented with different methods of keeping the user informed with sound.

Finally, there is research into the effectiveness of using audio interfaces when attempting to perform another task. These experiments tested the effectiveness of audio interfaces for use while performing some primary task like walking or driving. The results showed, in nearly all cases, that users were able to perform better in their primary task while using the audio interface. They also demonstrated in some cases that users performed better in their secondary task involving the interface.

### 2.1.1 Designing ambient interfaces

Ambient interfaces are interfaces that display information at the periphery of a user’s attention. This helps the user to maintain awareness of a secondary activity or process while performing a different primary task. We will enumerate some guiding principles for building ambient interfaces in this section, but first we will more concretely illustrate the concept of an ambient interface with some examples.

**Ambient interface examples**

The AmbientROOM and Ambient Fixtures were first described in detail in the Wisneski, Ishii, et al. paper *Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information* [33]. The AmbientROOM is a cubicle-like office space, surrounding the user with displays of different modalities.
Water ripples projected on the ceiling represent the activity of a distant colleague (or office hamster in this case), while activity within the office’s digital whiteboard system is represented by the sound of dry-erase markers rubbing against a board. Another ambient fixture, a motorized pinwheel, might be used by an atmospheric scientist to reflect solar wind intensity. All of these interfaces are designed to rely on peripheral awareness to convey useful information while the user is focused on another primary task.

Another visual ambient display is AuraOrb [1], where the arrival of new email messages is indicated in a desk-mounted sphere by a gradually brightening blue glow, which waits until a user gazes at the sphere to display the subject line of the email.

In an office-like environment, it is easy to give visual and audio feedback to the user by installing physical fixtures or projected displays. Interface designers can have complete control over this environment. When mobile, however, displaying visual feedback in the user’s environment is much more difficult. A pedestrian is not typically in a small area that can be easily and inexpensively retrofitted with displays throughout the environment. Heads-up display (HUD) goggles might solve this problem, by requiring only a single portable display device that can implant graphics into the user’s view of the environment, but HUD goggles are currently impractical due to expense and physical inconvenience.

Rather than using a HUD for visual feedback, an audio-based interface may be a more practical alternative, since modern earbuds are portable, lightweight, and inexpensive. In addition to the practical advantages of using audio hardware while mobile, an audio interface may provide less distraction to pedestrians, due to the use of a different sensory modality from their primarily visual task of navigation. The
next example considers an audio interface that uses sound to help maintain awareness of a system, although it does not directly address mobile interfaces.

The ARKola simulation experiment [7] was run to test the ability of an audio interface to aid the monitoring of a soft drinks manufacturing plant. The user maintains and inspects the assembly line using a visual interface, while the entire assembly line can be monitored using the audio interface. Each component of the line is represented by a sound that indicates its state, producing a very busy sonic environment. The pattern of these sounds is modified by the change or absence of individual components, helping the user identify when the system is running abnormally.

**Ambient interface design guidelines**

The main goal of ambient interfaces, such as those presented in the last section, is to help the user maintain awareness of some system or process while performing other tasks. Two design guidelines for designing ambient interfaces are:

- Move information off the screen (or main interface) and into the user’s environment, to free up this main interface and rely on a user’s peripheral attention to communicate information.

- Take advantage of the senses the user is not employing in the primary task, using a complementary modality that may be less distracting from the primary task.

Both guidelines are central to the principles of ambient displays. The first principle was expressed in Weiser and Brown’s *Designing Calm Technology*, the first paper to address the concept of peripheral ambient displays:
Technologies encalm as they empower our periphery. This happens in two ways. First, as already mentioned, a calming technology may be one that easily moves from center to periphery and back. Second, a technology may enhance our peripheral reach by bringing more details into the periphery. An example is a video conference that, by comparison to a telephone conference, enables us to attune to nuances of body posture and facial expression that would otherwise be inaccessible. This is encalming when the enhanced peripheral reach increases our knowledge and so our ability to act without increasing information overload. [31]

While the ARKOLA simulation is based on the second principle of using alternate modalities to help users maintain awareness of systems they could not see, Wisneski et al.’s *Ambient Displays: Turning Architectural Space into an Interface between People and Digital Information* may have been the first to articulate the second guideline as it applies to ambient interfaces being used for a second unrelated task:

The choice of modality for the background media should be considered with the person’s foreground task in mind. For example, if a person is in his office performing a visually intensive foreground task - say, writing software - a visually based ambient display on the wall behind the person might not be as effective as an auditory display. Similarly, when a person is performing an intensive auditory task, like talking on the phone, ambient information might be better presented through non-auditory ways, or for example, in visual ways like shadows on a wall. [33]

In all, the research surrounding ambient interfaces provides useful starting guidelines for designing an interface that attempts to minimize its impact on a user’s
attention. In a mobile environment, the options for integrating displays into the environment are limited, but audio interfaces based on ambient principles could make use of the user’s peripheral attention while maintaining the necessary physical portability.

2.1.2 Ambient audio in location-based games

Ambient audio has been used in a small number of games, including SoundPark [17], Songs of North [5] and Viking Ghost Hunt [16]. These games use non-speech sound to inform players’ movements and actions while they’re walking outdoors, without requiring the player to look at a screen. As we shall see, these games show that ambient audio has promise, but may have limited appeal depending on implementation specifics (such as using external speakers for audio transmission).

SoundPark is a team game that is played outdoors, requiring players to “capture” sounds and use them to construct a song [17]. “Hunter” players use earphones to hear sounds spatially in the environment, and must gather them up and bring them back to home base in order to assemble them into a musical arrangement.

In Songs of North, players perform quests to find items and rescue characters. They kill monsters and cast spells using a virtual drum on the screen [5]. Players ambiently hear other characters, enemies’ attacks, and the drum spells of other players. The authors found that players were not likely to use the audio interface unless they were explicitly told of its importance. Songs of North uses external speakers with no volume control, making the game socially awkward to play.

In Viking Ghost Hunt, players find “paranormal” zones using audio cues triggered by GPS location [16]. Once a zone is found, players interact with the screen to decode secret messages. Ambient sound helped to create an immersive and emotionally
engaging experience, where 70% of players either agreed or strongly agreed with statements characterizing the game’s audio as immersive.

These three examples provide early indications that ambient audio can benefit mobile games. In all three games, the ambient audio helps players find desired locations in the physical world while maintaining awareness of their surroundings. The spatial audio cues used in SoundPark may permit even more efficient ambient navigation than the basic “proximity” feedback of the other games (since a target’s direction is provided in addition to its proximity) making this type of interface a good candidate for future ambient interfaces.

Songs of North and Viking Ghost Hunt also demonstrate that ambient audio can be easily used in tandem with a visual interface, where the audio is used while walking and the screen can be used to play mini-games while stationary at a waypoint or destination. This means that game designers can change between different interfaces when necessary to adapt to a user’s level of attention, without fear of overly confusing the player.

Viking Ghost Hunt goes on to show that ambient audio related to real-world locations can help to make gameplay feel immersive, as shown by the players’ feedback. Viking Ghost Hunt, however, always used ambient audio in combination with visual interfaces, and made no comparison between the two. It is not known how the immersion of ambient audio and visual interfaces compare when each interface is used on its own, so this remains to be investigated.
2.1.3 Ambient audio in other mobile applications

Ambient audio has been used in other applications intended for mobile use, with goals of reducing the demand on users’ visual attention or of increasing the application’s immersive experience.

Several applications apply ambient audio to navigation tasks, with the goal of allowing users to focus visual attention on a demanding foreground task (e.g., driving) while navigation instructions are conveyed via audio. Holland et al. created the AudioGPS system as a minimal-attention navigation system, using spatial audio cues to highlight waypoints of the path to be followed [8]. Initial informal evaluation of this system resulted in positive feedback.

Stahl later created and tested the ambient the Roaring Navigator, for identifying and navigating to landmarks in a zoo environment [25]. Representational sounds are played at the location of each landmark or animal in the zoo. All of the participants using the system were able to complete a wayfinding task around the zoo, and most rated the interface easy to use.

2.1.4 Effectiveness of audio interfaces

Ambient audio has the promise of helping users navigate more safely while increasing their immersion in a virtual world overlaid over the physical world. The above examples indicate that systems based on these principles can be practically built. So far, however, there has been little empirical evaluation of how effectively ambient audio conveys information, and whether, when compared to visual interfaces, an ambient audio interface would improve users’ safety, immersion and enjoyment. We now review those experiments that have been performed to-date.
Pirhenen et al. tested a user’s ability to control a touch-screen portable music player while walking around a set of obstacles [18]. Participants using gesture-based input with audio feedback experienced a smaller workload compared to those using the touch-screen GUI, and also tended to walk at a pace much closer to their preferred walking speed. In this experiment, the audio interface is enhancing the user’s performance in his primary (walking) task, and the lower workload hints that the walker’s safety might also be improved (although this was not tested).

Walker and Brewster compared the use of visual and spatial audio interfaces on a computer to monitor a background task while performing a demanding visual foreground transcribing task [29]. The background task required that users monitor file transfers that periodically required user intervention. Participants used either a standard visual progress bar or a novel spatial audio progress bar that could display transfer progress and rate. Users’ performance in both the foreground and background tasks was better in the audio condition; users were significantly more likely to correctly handle the file transfer task, and were also able to type at a faster rate. The higher performance in the foreground task mirrors the positive result of the music player test, where the interface is less distracting to the primary task than a visual interface. The increased performance of the background task with audio also shows that enhanced performance in the foreground task is not necessarily at the cost of ignoring the background task when using an ambient audio interface.

Takeuchi and Sugimoto compared audio and visual interfaces for navigating city streets [27]. CityVoyager is a system that analyzes behaviour while performing everyday shopping, and then guides users to recommended shops based on their shopping
CHAPTER 2. RELATED WORK

history. A simple beeping “metal-detector”-like interface was used to indicate recommended shops, and was played over a speaker. When users had access to both the visual and audio interfaces rather than the visual alone, they scrolled the map eight times less, and visited the recommended shops more often. When the audio cue beeps became very frequent near shops, however, users reported being more distracted by the interface than when using the screen alone. The results of this case complement those of the experiment above: when audio was available, users followed the device’s directions more often, possibly because the audio interface allowed them to be more attentive to the system while doing their shopping.

In addition to these studies on the effectiveness of ambient audio, there have been notable studies on the comparative effectiveness of spoken-word audio in navigation tasks. Walker et al. found that car drivers using a spoken-audio guidance system had fewer navigational errors than those using a visual mapping device, and did not slow down as much as in the visual case [30]. Here the driver’s foreground task performance (counting both movement speed and safety) is increased when using audio, as in the findings above by Pirhonen et al. and Walker and Brewster.

Sodnik et al. also tested drivers in a simulator, using steering-wheel mounted controls to navigate a menu in order to perform predetermined tasks while driving [24]. Drivers who had spoken-word representations of the menu system again drove more safely than those using visual displays, with 60% fewer driving penalties assigned on average during the tests.

Collectively, these studies show that ambient audio interfaces can improve users’ performance when they need to split their attention between two tasks. The results nearly all show better performance of the foreground task when using audio, and
some also show better performance in, or attention to, the background task as well.

They also show that audio in general can lead to lower error rates in the foreground task, which in many cases translates directly to increased safety in a potentially dangerous walking or driving environment.

Based on this work, ambient audio might be well suited to location-based games, thanks to the potential for increased performance and safety of the primary walking/running task. No study, however, comprehensively compares ambient audio versus visual displays in terms of their effects on performance, safety and immersiveness. In chapter four of this thesis, we take a first step towards filling this gap.

2.2 Simplifying the use of sensors in developing location-based games

The interface research being conducted in this thesis is applicable to mobile applications, and especially to location-based games that use location and orientation tracking to control the application. For this interface research to have any practical application in aiding users, location-based games and applications must see some use in the real world.

But these applications risk irrelevance if they are too difficult or time-consuming to develop. They require knowledge of the player’s location (and sometimes orientation) in the world to function, likely using some form of sensors built into the device. All of the supporting code necessary to use the sensor data can be difficult and time consuming to create from scratch. Based on the existing literature on location-based applications and sensor frameworks, and our own experience building location-based
applications, we have identified five time-consuming problems that can pull developer focus away from application development:

1. Noise problem - Sensor noise or other inaccuracy can cause inaccuracy and erratic behaviour in the user’s display of the virtual world if raw sensor data is used.

2. Format problem - Raw sensor readings aren’t always in the desired form (e.g., rotation rate instead of current direction).

3. Sensor inconsistency problem - When attempting to use a different sensor, application code may require rewriting due to differences between sensor interfaces.

4. Early configuration problem - Choosing which sensors to use in advance can lead to inoperability if different sensors are available.

5. Extensibility problem - It can be difficult to extend existing code to support new sensors or fusion algorithms.

In this section, we examine existing sensor frameworks that can save developer time by addressing some of these problems when creating location-based applications. We also examine why sensor fusion is an important feature of any framework that attempts to use inexpensive sensors for location tracking while outdoors.

2.2.1 Sensor fusion

Sensor fusion combines sensor data from multiple sources to obtain a more accurate representation of the environment or system being measured than any one sensor could provide [4]. A simple form of sensor fusion is the averaging of input from multiple
sensors that are measuring the same property, in order to average out the noise from individual sensors (e.g. averaging the measurements of multiple anemometers to ascertain wind speed).

More complex techniques take advantage of the known behaviours of different sensor types. For example, a gyroscope, magnetometer, and accelerometer might be used in tandem. Since the magnetometer needs to track its orientation with an accelerometer in order to calculate its magnetic bearing, it can give erroneous measurements when subjected to external forces, such as those experienced while walking. In order to compensate for these errors, a gyroscope (which is not sensitive to such forces) can be used to track changes in bearing while walking. When the accelerometer detects that the user is at rest, the magnetometer can then be used to calibrate the system with an absolute bearing.

A more advanced approach is to use a Kalman filter [32]. For each step, a linear model of the system is used to predict the future state, and then this prediction is averaged with data from multiple noisy sensors (which are weighted based on their estimated uncertainty) to finally arrive at a value that estimates the true state of the system [6].

In the space of mobile applications, noisy and inaccurate sensors like GPS receivers and magnetometers are used in order to allow freedom of movement around an unprepared environment. This leads to the noise problem stated above. Sensor fusion can be used to improve the quality of user tracking in mobile situations by combining redundant sources of sensor data to reduce the uncertainty in the property being sensed [13]. A sensor framework that supports sensor fusion in location-based games and other applications might improve the experience of users through better
tracking, and also allow developers to place more focus on the application itself.

\section*{2.2.2 Frameworks for handling location sensor data}

Above, we listed five time-consuming problems with developing location-based applications. One way to address many of these problems is to use a sensor framework that interfaces with the sensors and provides the tracking data to the developer. A sensor framework can save developers from re-inventing sensor-reading code and allowing them to spend more time on application development. These frameworks typically read in the raw sensor data, transform it in some way, and pass an abstracted form of the location data to the application. Transformations of the sensor data can improve accuracy through sensor fusion or other filtering, as well as put the data into a form that is better suited to the application domain. The abstraction of sensor data can also make it easier for the framework to gracefully handle the change or loss of sensors during operation without causing disruptions to the application’s behaviour.

The configuration of a sensor framework includes the selection of the sensors being used, and the specification of the transformation steps (e.g., filtering or sensor fusion) that are used during tracking. The way that this configuration is specified can vary between frameworks, depending on the framework’s focus: either the configuration must be specified manually by the developer, or the framework can automatically perform this configuration.

This can affect the degree to which a given framework is able to address the problems listed above. Automatic configuration frameworks might be able to switch between multiple sensors without intervention, even at runtime. In contrast, a manual configuration framework might require work by the developer or user to change the
sensors being used, and may preclude the ability to change sensors at runtime. The following are some currently used sensor frameworks for handling input to location-based and augmented reality applications.

Manual configuration frameworks

The Virtual-Reality Peripheral Network (VRPN) gives Virtual Reality (VR) applications access to data from input devices, using a protocol that is able to run over networks. This ability allows applications to access remote devices over a network, and allows the simultaneous use of a single device by multiple applications [28]. What makes VRPN interesting as an input framework is the ability to create layered devices, which take the signals from an existing device and modify it to filter, amalgamate, or otherwise transform the input.

This functionality allows developers to create a hierarchy of transformations that takes sensor, button, and other forms of input from multiple devices and transforms them into suitable input for a specific VR application. This can help to solve the format problem, since the data can be easily transformed into the desired format. Figure 2.1 demonstrates a potential VRPN hierarchy: a VirtualTracker layered device takes analog data from two Joystick devices and transforms them into a tracking format that emulates a real location and orientation tracking device, providing the result to an application for use when debugging.

By standardizing the data types that can be passed around (i.e. button press, analog value, tracking), VRPN effectively standardizes the interfaces for sensors or peripherals with the same functionality, addressing the sensor inconsistency problem. It does, however, require all the sensors being used to be specified in the hierarchy,
meaning it cannot solve the early configuration problem. VRPN is also extensible, permitting the creation of new devices and device types, while also supporting *layered devices* that let the developer program higher-level behaviour based on input from one or more sensors. These layered devices have the potential to allow the building of sensor fusion algorithms to improve tracking accuracy, addressing the noise problem.

*OpenTracker* supports tracking hardware with a flexible and customizable architecture [20]. It uses dataflow graphs to manage data being passed from sensors to applications. OpenTracker functions similarly to VRPN, except that the process of input, transformation, and output to an application is made explicit, rather than requiring the layering of many virtual devices to accomplish the same task. Here, “device drivers” act as source nodes that bring data into the system; “filter nodes” transform, merge, or otherwise modify passed data from source nodes, and “sink nodes” output the data to an application. This “hierarchy of filters” architecture again helps to address the format problem and the noise problem, while saving some time since OpenTracker includes a number of pre-defined filters to solve issues like
predicting future data values, merging data from multiple sources, and smoothing signals.

OpenTracker has a high level of configurability, using an XML schema to define the dataflow graph and supporting custom nodes, making it quite extensible. It also uses standardized data formats between nodes to help address the sensor inconsistency problem. Like VRPN, however, OpenTracker does not address the early configuration problem, requiring the developer to provide a configuration file that describes the exact dataflow graph and devices to use.

The OpenInterface Project [12] [23] uses a similar “hierarchy of filters” architecture to solve a related problem in human-computer interaction: developing multimodal interfaces to adapt a desktop application’s standard input methods to match the needs of users or to take advantage of advanced input devices. This can include
adaptations such as mapping head tracking to mouse input or using voice recognition to trigger application commands.

OpenInterface is an open source platform for rapidly prototyping these multimodal input interfaces for computer programs. Unlike VRPN or OpenTracker, the OpenInterface Workbench uses a graphical interface (shown in figure 2.2) to manipulate the hierarchy of devices and filters being used. In the graphical editing interface, OpenInterface uses modular transformation components to create the hierarchy of transformations, just as OpenTracker uses filter nodes to perform the same actions. As in VRPN and OpenTracker, OpenInterface does not support the automatic selection of sensors and fusion algorithms, requiring explicit configurations by the developer like VRPN and OpenTracker. With support for the necessary devices and transformation components, OpenInterface could potentially be used with location-based games, and demonstrates how an input framework could take advantage of a graphical interface for rapid prototyping situations.

Automatic configuration frameworks

Ubitrack, in contrast to the above mentioned frameworks, is designed for automatic configuration, and explicitly solves the early configuration problem by automatically using the most suitable sensors from all the available sensors at runtime [19]. It can make use of a network of heterogeneous sensors and embedded tracking systems to track a person or object within the network’s reach, automatically using the sensors and systems within tracking range of the entity.

Ubitrack uses Spatial Relationship Graphs (SRGs) to keep track of the absolute and relative positions of sensors within the network. An SRG is a graph where
each node is the coordinate frame of some object, and the edges between nodes represent the spatial relationship between the two objects. Based on the known relationships in the SRG, Ubitrack can calculate what sensors and transformations are necessary to track any object in the SRG. When an application demands to know the location of a specific object, Ubitrack builds a unique dataflow network of sensors and transformations based on the SRG to provide this location with the best accuracy possible.

The ability to gather and analyze information from multiple sensors automatically allows the use of sensor fusion and filtering before passing the output data to an application, solving the format and noise problems together. There does not appear to be any way to modify or extend the algorithm used to configure the dataflow networks, or to override this algorithm to use specific devices or configurations, making this framework not easily extensible.

Summary

This section has examined some of the available frameworks for handling player-tracking input to mobile applications, as well as noting the similarities and differences in their approaches. All of the frameworks use some form of dataflow network to handle the data: taking the raw input from tracking sensors, transforming it into the most appropriate format while reducing noise, and sending it to the application in a way that isolates it from the underlying hardware.

The major difference between the frameworks was their method of configuration; VRPN, OpenTracker, and OpenInterface all require manual configuration, while Ubitrack will perform all necessary configuration and adapt at runtime to changes in
available sensors. This means that all three manual configuration frameworks do not address the early configuration problem, giving an advantage to Ubitrack.

This seems to involve a tradeoff, however, as all three manual configuration networks are extensible, while Ubitrack appears to not support any simple form of extension. Another feature of the manual frameworks is that specific configurations can be used when the developer chooses to, should there be some advantage to using a very particular configuration with the given equipment and environment. There again does not seem to be any way to manually select a configuration with Ubitrack.

None of these toolkits is able to fully solve all five problems listed in this section. Our TREC framework has been designed specifically to fill this gap, and is described in chapter five.

2.3 Conclusion

This chapter has presented work addressing two problems that arise when developing location-based applications for mobile devices. We first considered the design of interfaces for mobile devices that allow a walking user to safely and efficiently comprehend feedback from an application. We then reviewed technologies that make it easier for developers to access and employ the user’s location and orientation data from the available sensors.

The work addressing the creation of safe interfaces for mobile users covered three areas of investigation: the use of ambient interfaces to minimize the interface’s impact on a user’s attention, the implementation of ambient audio interfaces in location-based games and applications, and testing the effectiveness and safety of audio interfaces
when compared to visual interfaces. This work collectively hints that ambient, audio-based interfaces would likely be a safer and more effective alternative to visual interfaces in location-based games, but this has not yet been demonstrated. The topic of designing location-based games with ambient audio is examined further in chapter three, while an experiment to test the safety and effectiveness of ambient audio is detailed in chapter four.

There has also been research into making development of location-based games easier, by alleviating five problems associated with accessing sensor data for location tracking: the noise problem, the format problem, the sensor inconsistency problem, the early configuration problem, and the extensibility problem.

Sensor fusion was seen to be a useful tool for improving the accuracy of sensor data when multiple sensors are available for analysis. Four sensor frameworks for handling sensor input were also examined, showing that there are frameworks to address most of these problems, but none attempts to solve all five. In chapter five, a framework designed to solve all five problems is described.
Chapter 3

Designing Ambient Audio Games

As we have discussed, people can experience difficulties in using hand-held displays while walking. First, the graphical interface of the application requires visual attention, distracting the user from his environment. This can result in compromised safety and reduced performance. Second, when using a map display, users need to translate the markers on screen into locations in the real world. This effort may decrease the user’s sense of immersion in the virtual environment being depicted. We have applied ambient audio to both of these problems, with the goal of improving the experiences of mobile users.

3.1 Ambient audio

An ambient audio interface is an audio interface that uses continuous streams of sound to convey information, using the principles of ambient interfaces described in chapter two. This involves making changes to the sound streams (e.g. changing the sound’s volume) that the user can notice and understand with little attention, so that
it will not distract the user from primary tasks like walking.

A simple navigation application, for example, could make a continuous sound stream appear in the direction of the user’s destination, increasing in volume as they get closer. The user would then follow the sound to reach his destination, without requiring complex instruction, while maintaining full visual awareness of his surroundings. Another application might allow users to hear when a friend is nearby, using a unique sound stream to identify the person, and playing this sound stream at the friend’s location.

We will now present two examples of ambient audio applications that we have developed, in order to illustrate how ambient audio can be used when creating more complex games and applications.

### 3.2 Examples of ambient audio applications

We have developed two location-based applications using ambient audio. Noisy Planet is a tourist guide application for finding interesting locations and navigating around a city, and Growl Patrol is a fast-paced location-based game of catching runaway animals in a park or other large open area.

Both of these applications use *spatial audio* processing to make sounds appear to originate from a specific location in the user’s environment, which is explained in detail in section 3.4.2. This spatial audio processing allows each application to indicate that an entity or phenomena is associated with a particular location by playing a sound stream in that location, which should be natural for users to comprehend.
3.2.1 Noisy Planet

Noisy Planet is a location-aware mobile tourist guide created to demonstrate the concept of ambient audio, and was developed using the TREC toolkit [11]. This application, shown in figures 3.1 and 3.2, allows a tourist staying in an unfamiliar city to navigate to proximate destinations on foot while also allowing serendipitous exploration of the area.

Noisy Planet uses an ambient audio interface to convey to users the position and distance of nearby points of interest. For example, in figure 3.1, the user “hears” that the Stauffer Library is behind him and to his right. Each point of interest is represented by an audio stream – for example, the sound of riffling pages represents the library; clinking glasses represent a restaurant, and jingling coins represent a bank. These audio streams are chosen to hint at the identity of the point of interest being displayed, while remaining ambient sounds that do not tug at the user’s attention.
A tourist using Noisy Planet might begin navigating to a popular landmark by pinning a sound stream to it and simply walking towards the sound. This sound gets louder as the tourist nears, just as in real life, but maintains a baseline minimum volume level in order to facilitate navigating from a distant location. The tourist might also turn on an "exploration" layer of sounds that correspond to his known interests. For a culturally-oriented tourist, all of the independent bookstores, cafes, and museums along the way would have sound streams that gradually become audible as they were approached. Since there is no direct path to follow to his destination, the tourist can easily meander around while discovering locations directly related to his interests. When it is time to continue on to the final destination, the tourist simply listens for the destination’s pinned sound stream and again heads towards it. Ambient audio makes it easy for the tourist to tune-in and tune-out of the audio streams whenever he chooses.
CHAPTER 3. DESIGNING AMBIENT AUDIO GAMES

3.2.2 Growl Patrol

For an example of how a more complex location-based game or application can benefit from the use of ambient audio, we now present Growl Patrol. The Growl Patrol game has been implemented using the TREC toolkit described in chapter five, and has been play-tested by tens of people [10].

Growl Patrol is played outside, using GPS to track the player’s real-world location. The premise of the game is that a number of small animals (cats, dogs, and birds) have cunningly escaped from the local pet shop, and the player is charged with bringing them back. These animals run around the play area until they are picked up, at which point the player can drop them off at the pet shop. There is also a hungry tiger nearby, who attempts to steal any animals that the rescuer is carrying. The player’s goal is to catch and drop off as many of the animals as possible in the time allowed, without losing any to the hungry tiger.

Growl Patrol uses an interface based on ambient audio. Players hear the animals through stereo headphones. Each animal is represented by a constantly repeating sound, or sound stream – birds twitter, cats meow, and dogs bark. In order to function well in the ambient audio interface, there are only short pauses between each sound in the animal’s audio stream. In addition, the volume levels of the original sound files are modified to be consistent over time. Since changes in volume are used in the game to indicate an animal’s distance, volume changes in the source files could be easily mislead users to believe that the animal was moving.

The player begins the game hearing an animal’s sound stream in the near distance, coming from a specific direction. While walking towards this animal, the sound stream gets louder, indicating that the player is getting closer. A short chase ensues, and
once the player manages to get close enough to the animal, it is put into his backpack and disappears from the playing field. Then the player pursues the next animal he hears.

Sound is used in Growl Patrol as an alternative to vision; the player uses sound for his primary task of navigating around a complex environment. This is based on one of the guiding ideas for implementing an ambient interface stated in chapter two, that an ambient interface should use an alternate modality from the primary task if possible.

Using spatial audio

In Growl Patrol, sound is spatialized on a 2D plane; if the animal is to the player’s left, the sound will appear to be coming from the left, and the closer the animal is to the player, the louder it will sound. Figure 3.3 shows a player outside playing Growl Patrol. He is wearing stereo headphones to “hear” the game. His position is tracked via a GPS sensor, and his head orientation is tracked with a gyroscope and compass attached to his hat. The figure shows that the player hears a bird tweeting to his right. The map in the top of figure 3.3 (not part of the game) illustrates what the player hears: the bird is close by to the right, and the tiger is in the distance to the left.

There are two advantages to using spatial audio for Growl Patrol: the player may feel more immersed in his environment compared to using a handheld display, and also avoid accidents while walking in the outdoor environment, as shown in chapter four.
Immersion and safety

Growl Patrol uses ambient audio to provide a virtual overlay on the real world, using sound to locate the animals and pet shop. This approach can improve immersion versus a hand-held visual display. Rather than having to look at a map and work out how it corresponds to the physical world, the animals’ sounds are overlaid directly onto the real world. As one player said, “it’s like you’re in a park trying to catch animals, as opposed to staring at the little map trying to catch animals.”
 CHAPTER 3. DESIGNING AMBIENT AUDIO GAMES

The use of an audio-based interface can also improve player safety versus a visual interface, as players’ vision is free to view the world around them. The use of ambient audio (versus alert-based or spoken audio) allows players to drop their attention from the game when the real world intrudes. When the player returns his attention to the game he can immediately interpret the game state, allowing the player to continue chasing the next animal.

3.3 Ambient audio design guidelines

In the course of developing and testing the two location-based applications described above, we have gained a better understanding of the strengths and limitations of ambient audio interfaces. Based on this experience, we have compiled the following set of guidelines for designing ambient audio applications.

First, only a small number of sound streams (up to 4 based on our experience) should be audible at any time in the application. Playing too many sound streams simultaneously will confuse the user and make the individual streams harder to distinguish, though the maximum acceptable number of streams may be depend on the accuracy of the spatial audio algorithms being used, and on each user’s individual hearing proficiency.

Next, we recommend using sound streams to represent phenomena that occur at a specific location in the physical world. By placing a sound stream at this specified location in the physical world, ambient audio simplifies the process of mapping locations in the application to real locations.

Ambient audio should not be used to represent information that is time-critical. If user must act immediately when certain information is transmitted, he will have to
devote constant attention to the interface to prepare for quick reaction. This would prevent the user from comfortably tuning out of the interface when necessary, and would prevent the application from being truly “ambient” and low-attention.

Similarly, ambient audio should not be used to represent discrete events in time. Since these events exist for only a short period of time, there is a chance that the user will miss them completely if he is momentarily tuned out of the interface. Ambient audio is better suited to representing the current state of some entity or continuing process, where the sound streams will seldom “appear” or “disappear” when a user is momentarily focused on something else.

Finally, ambient audio should only be used when communicating information that is low in detail. The state being reported to the user by a sound stream must be communicated using changes in the sound stream’s direction and volume, so only a small number of variables can be represented. For example, the content of an email message would be difficult to communicate with ambient audio, while the number of unread messages would be much easier to represent with a sound stream’s volume or direction.

3.4 Implementation considerations

Beyond these general guidelines, there are a number of technical issues with implementing ambient audio interfaces. In this section, we discuss how to select appropriate sounds for a sound stream, methods for modifying the sound stream to convey information to the user (including spatial audio processing), and the use of position tracking to ensure the user’s virtual audio environment matches up with the physical environment.
CHAPTER 3. DESIGNING AMBIENT AUDIO GAMES

3.4.1 Using sound streams

A sound stream should be a set of sounds played either continuously or with short pauses between sounds. During long silences in the sound stream, there would be no way to notice changes in the data until the next noise begins to play, and the user may mistake the pause as the ending of one stream and the beginning of another.

For very simple forms of data that are only one-dimensional (e.g., temperature, activity level, task progress) changes in the volume of a stream of audio are sufficient to indicate their current state. In order to make use of volume in this way, it is also important for the original audio files to have a uniform volume over time. If there are significant volume changes in the original sound, users might mistake these for live changes in the variable that they are monitoring.

In order to convey more complex data, further changes can be made to the sound streams. For each variable we wish to add, another sound filter or effect can be added to represent it. For example, in addition to using volume to represent the activity level of a colleague in a messaging application, reverberation might be added to the sound to give a sense of distance, representing how long it has been since the last conversation with that person.

There are additional sound modifications that can be used for ambient audio, such as filtering low or high frequencies from the original sound, adding echoes of the sound to the sound stream, or repeating sections of the sound in a loop. Many of these “musical” types of sound effects were used in Barrington et al’s study investigating the use of audio effects on music to ambiently communicate information [2]. In the case of location-based games, the communication of important locations can be critical to gameplay. Spatial audio was chosen as the primary method of encoding data.
in ambient audio, due to its ability to present a relative location to the user quite naturally.

### 3.4.2 Spatial audio algorithms

Spatial audio processing allow users to listen to headphones (or any source of audio where each channel is only heard by one ear) and perceive the sounds to be emanating from a specific location outside of the head. There are many cues that the brain uses to perceive sound directions, and spatial audio artificially introduces these cues into the sound to trick the brain into hearing it from a certain direction.

The most easily perceptible directional cue is called the Interaural Level Difference (ILD), where the ear closest to a sound source hears the sound more loudly than the far ear. The relative difference in volumes helps to identify the angle of an incoming sound. The brain also calculates direction using the Interaural Phase Difference (IPD), the difference in the phase of a sound wave being heard by both ears. This is produced when a sound wave reaches one ear first and then must travel farther around the head to reach the other ear, resulting in a milliseconds-long delay that puts the tone out of phase [15].

To perceive sound directions even more accurately, the outer ear and ear canal help the brain by physically modifying the sound with a directional signature as it enters the ear from different directions. The shape of the ear and ear canal will either increase or decrease the amplitude of the sound in narrow “notches” of frequencies. The shape and position of these notches can be interpreted by the brain to indicate the sound’s direction [15].

To take advantage of all three of these audio cues, we use a simplified Head-Related
Figure 3.4: The equipment used to track a user’s location and orientation for an ambient audio interface.

Transfer Function (HRTF) to perform the spatial audio processing. An HRTF is a mathematical model of the effects that a person’s head, ears and ear canal have on a sound before it reaches the eardrum. A generalized HRTF can use ILDs, IPDs, and notch filtering to approximate the average person’s perception of a real 3D sound environment [3]. We use a Creative Soundblaster X-Fi Go sound interface to perform the spatial audio processing for Noisy Planet and Growl Patrol, which includes a built-in generalized HRTF that can be controlled using OpenAL commands.

3.4.3 Position tracking for spatial audio accuracy

In order to use spatial audio to indicate locations in the real world, tracking devices need to be used on the player in order to know his location and head orientation. The head tracking is especially important for ambient audio, since all sounds are presented relative to a person’s head (or more specifically, ear) orientation.
In the prototype system that we use to test ambient audio, there are two devices being used to track head orientation: a 3-axis magnetometer and a single axis gyroscope. As explained in chapter two, using multiple sensors with sensor fusion algorithms can reduce the errors in sensor readings, and lead to more accurate tracking. In this case, the magnetometer alone would experience large deflections from its true direction when walking and running, so a gyroscope was added to provide a sensor that would be less affected by the jarring motion of walking. This combination is mounted on a baseball cap, as shown in figure 3.4.

### 3.5 Potential limitations

Ambient audio is limited both by humans’ ability to accurately localize a sound source, and by the inaccuracy of algorithmically spatialized sound.

The average human’s hearing accuracy is quite variable, depending on the sound’s location, duration, or bandwidth. In a study performed by Makous and Middlebrooks, errors in identifying a sound’s location varied from an average of 2 and 3.5 degrees of error in azimuth and elevation, respectively, to approximately 20 degrees of average error, depending on the direction of the sound [14]. This may lead to some difficulty in using spatial audio to identify sound locations compared to using a visual interface.

Growl Patrol requires players to chase progressively faster animals and to get within a 10 m radius in order to catch them. Specifically, sound is spatialized using a generalized approximation of a person’s true Head-Related Transfer Function, providing lower accuracy than real-life sound [3]. Additionally, we have observed in casual testing that players find it difficult to determine a sound’s distance based on volume.
alone. Therefore, while our anecdotal experience indicates that Growl Patrol’s ambient audio increases immersion and safety versus a hand-held display, it also indicates that these improvements come at the cost of increased difficulty in gameplay.

3.6 Conclusion

In this chapter, we presented our concept of ambient audio as streams of continuous sound that appear to emanate from a specific location, being used to highlight this location to a mobile user wearing headphones.

We also described the design Noisy Planet and Growl Patrol, two applications built to demonstrate and test ambient audio interfaces for use by pedestrians.

We hypothesize that using ambient audio interfaces in location-based games might have benefits over handheld visual interfaces, since they might increase players’ feelings of immersion in the virtual environment as well as freeing their visual attention to allow safer play outside. The next chapter presents an experiment that we performed to evaluate ambient audio, comparing it with a visual interface for use in the Growl Patrol game.
Chapter 4

Evaluation of Ambient Audio Interfaces

There has been little evaluation of ambient audio interfaces for use while walking or running. In order to determine whether ambient audio is a reasonable alternative to handheld visual interfaces for pedestrians, we performed an experiment testing these two interface styles in a location-based game. A simple game based on Growl Patrol (described in chapter three) was developed with two possible interfaces: one using only ambient audio for gameplay, and one using only a handheld visual display.

We evaluated participants’ performance while playing both versions of the game, in order to compare the two interfaces’ effects on the participants. Three primary effects were measured: how immersive each version of the game felt to the participants, how safe the participants’ behaviour was during play, and how far the participants were able to advance in the game. We also investigated the influence of participants’ expertise in playing games that require using a map to navigate while playing the game. This was done to establish whether visual map expertise would make players
experience less benefit from ambient audio.

### 4.1 Hypotheses

This chapter describes the experiment run to evaluate the use of ambient audio interfaces in mobile games, specifically by comparing this type of interface with a handheld visual interface. There were four hypotheses regarding the performance of the ambient audio interface in this case, compared to a visual interface:

1. Ambient audio increases immersion in location-based games

2. Ambient audio increases safety in location-based games

3. Ambient audio leads to reduced performance in location-based games

4. The positive effects for ambient audio are more pronounced among players who have little or no experience using secondary radar-map displays in games.

These four hypotheses were drawn from the differences between ambient audio and visual interfaces, and our anecdotal experience with both types of interfaces. First, ambient audio may be more immersive than a handheld visual display because ambient audio is always being displayed to the user, while the visual display can only be observed intermittently by the user while he glances back and forth between his environment and the screen. In addition, ambient audio overlays its sounds directly on the world (as perceived by the user) while a handheld visual map requires interpretation to relate locations on the display to the real world.

Ambient audio may also increase safety compared to a visual interface, because it allows users to always maintain their visual attention on their environment. Users
who must periodically glance at a screen may have more trouble reacting to dangerous situations in their environment during these glances.

We speculate that ambient audio may lead to worse in-game performance for players of location-based games, due to the relative imprecision of ambient audio compared to visual displays. The limitations of human hearing accuracy combined with the limitations of the spatial audio processing being used (as discussed in chapter three) may limit a player’s ability to accurately locate in-game entities compared to a visual interface.

Finally, players who are accustomed to using maps to navigate while playing video games may find it easier to use the handheld visual interface, and may have a more difficult time adapting to an ambient audio interface. We hypothesize that players without this expertise may see greater benefits when using ambient audio compared to the visual map experts.

In the next section, we describe the details of the experiment, and the measures that were used to evaluate our four hypotheses.

4.2 Experimental method

The location-based game being tested was inspired by the Growl Patrol game described in chapter three. Two different interfaces were developed for this game. One interface for the game uses only an ambient audio interface similar to that used in Growl Patrol, while the other interface uses only a visual handheld display. To remove some uncertainty from the experimental results, we elected to run the experiment inside the laboratory using a virtual outdoor environment.
Figure 4.1: Indoor experimental equipment for the audio condition

4.2.1 Indoor experimental setup

Testing Growl Patrol in an outdoor area on the Queen’s University campus would risk unpredictable interruptions to the game by pedestrians or vehicles, and would introduce exogenous factors such as inaccuracy of current localization sensors. Therefore, we used indoor equipment to create an experience analogous to playing outdoors. The experimental setup using a projection screen and stationary bicycle is shown in figure 4.1.

A large projector screen is used to display a 3D outdoor scene from a first-person perspective. This represents the “physical” world that a player would see if playing the game outside. The player uses a stationary bicycle to control forward and backward movement in the “physical” world. An Xbox 360 gamepad is used to turn the player left and right, and the sound is provided through open-type supra-aural headphones.

The displayed outdoor environment does not show the animals, since the experiment is replicating outdoor play where players need to rely on their ambient audio
Figure 4.2: Indoor experimental equipment for the visual condition

or visual display to find the animals. Instead, the position of the animals is only indicated using either spatial audio or a top-down map screen.

4.2.2 Conditions

We tested two conditions: an ambient audio interface and a visual interface. In the ambient audio condition, players hear animals through their headphones, as shown in figure 4.1. In the visual condition, they see animals on an overhead map provided on a side display, as shown in figure 4.2. Players pedal to navigate a representation of the physical world shown on a large screen. Players must chase and catch a series of animals. Three animals are used: cat, bird, dog, and then back to cat. The animals are not shown on the large screen, so players must locate them with the ambient audio or the side display map, depending on the condition.

Once a player has caught an animal, a “level up” tone sounds to indicate the catch, and the next animal appears in the game. Each successive animal moves faster than the previous one, progressively increasing the difficulty of the game.
Figure 4.3: An example of the radar map used in the visual condition. This map is shown on a display on the player’s right side.

The virtual world contains shrubs, trees and buildings, shown on the large display. If a player bumps into one of these obstacles, an “Ouch!” sound is played, and the screen blanks for 0.5 s.

Both versions of the game were instrumented to log the player’s position, score, collisions and speed.

**Ambient Audio Condition**

The player hears the animal to catch through stereo headphones. The animal’s position is spatialized in the 2D plane. The only screen available to the player is the main projector screen showing the virtual environment, as shown in figure 4.1.

The animal’s position is spatialized assuming that players look forward at the large display. This removes the possibly confounding factor of sensor error in head tracking.
CHAPTER 4. EVALUATION OF AMBIENT AUDIO INTERFACES

Visual Condition

This version of the game shows the animal’s location on a small screen located to the player’s right, as shown in figure 4.2. The virtual world is presented as a top-down map, similar to the radar maps in games such as World of Warcraft or Halo. An example of this map is shown in figure 4.3.

Each type of animal is represented by a different icon, and the map shows significant features like buildings and the perimeter of the play area. The player is represented by an arrow in the centre of the screen that rotates to point in the player’s direction of travel, while the map scrolls beneath this arrow, centered at the current location.

To mimic a hand-held device’s screen for the visual condition, a 17” flat-panel monitor with a 4:3 aspect ratio was used. It was placed 138 cm away from the player to approximate the size of a smartphone’s screen at a typical viewing distance. (Kato et al. report that users hold mobile phones on average 35 cm away from their eyes [9]. A 17” display at 138 cm from the user occupies the same visual angle as a 3.5” display at 35 cm.) The screen was located at eye-level, to the side of the main projector screen at an angle of 28 degrees to avoid obstruction of the projector screen.

4.2.3 Recruited participant groups

24 participants were recruited from the student population at Queen’s University. Participants’ ages ranged from 19 to 44, with a median age of 25. 12 “experts” and 12 “novices” were recruited. Experts had at least 25 hours of experience playing video games that use radar maps, or currently play such games at least three hours per week. Novices had fewer than 10 hours of lifetime experience with video games.
that use radar maps.

We expected that inexperienced players would see a more pronounced benefit from the ambient audio interface as compared to experts, because the experts’ experience with radar maps would give them a comparative advantage in the visual case. Glancing back and forth between a radar map (an overview display of the environment) and the main display of the game is a common action in many types of video games, like Real Time Strategy (RTS) or First-Person Shooter (FPS) games. This practiced skill may give an advantage to these players in the analogous situation presented in the visual condition of this experiment. An example of a radar map used in Electronic Arts’ Command and Conquer is shown in figure 4.4, where the main view shows a
number of vehicles and troops in a section of the play area, and in the upper right there is a map displaying an overview of the entire play area, including both friendly and enemy entities.

4.2.4 Measures

In this experiment we used three primary types of measures: player performance, player presence, and player preference. We compared these measures between the two conditions and the two levels of player expertise.

We measured player performance in the game using two ratings. First, a player’s score in the game was used to measure his advancement and general performance in the game. This score is equal to the number of animals that have been caught in the game. In addition, we measured each participant’s average speed while playing, which can be used to roughly indicate a player’s proficiency at navigating through the virtual environment.

Safety was evaluated by another measure of navigation performance in the virtual environment: the number of collisions with obstacles during a play session. A collision occurs when the player attempts to move too closely to an obstacle, which is measured by a specific “collision radius” for each type of obstacle in the game. The number of collisions in a play session directly measures the number of potentially dangerous accidents that the player experienced in the game.

Presence was measured using the Witmer and Singer Presence questionnaire [34]. This questionnaire was administered after each play session and provides four scores measuring different aspects of a player’s sense of presence: involvement, sensory fidelity, adaptation/immersion, and interface quality.
Throughout this thesis, we use the term “immersion” to refer to the sense of a virtual world seeming to surround and involve a player. Witmer and Singer use the word “presence” to refer to a related concept that counts immersion as a large factor. Their definition is focused on traditional virtual environments rather than situations mixing real and virtual elements, describing presence as “the subjective experience of being in one place or environment, while being physically situated in another” [34].

This does not actually exclude its use in an environment that mixes real and virtual components:

Experiencing presence ... in a [virtual environment] requires the ability to focus on one meaningfully coherent set of stimuli (in the VE) to the exclusion of unrelated stimuli (in the physical location). To the extent that the stimuli in the physical location fit in with the VE stimuli, they may be integrated to form a meaningful whole. [34]

The “adaptation/immersion” score given by the Presence questionnaire directly addresses the concept of immersion being used in this thesis. The other three scores are measures which, in addition to immersion, address the complete concept of presence, and do not directly concern our hypotheses.

We measured player preference with a custom questionnaire of nine questions, available in appendix A. These questions addressed topics like the perceived difficulty of play and whether the gameplay was fun.

4.2.5 Procedure

At the beginning of the trial, the participant completed the Witmer and Singer Immersive Tendencies questionnaire [34] and a custom questionnaire given to determine
participant expertise with video games and associated equipment.

Next, participants tested each of the two conditions. The order of conditions was chosen so that each condition was performed first by half of the participants, and so that an equal number of experts and novices experienced each condition order.

For the first condition, players were given a short explanation of the game rules, and a summary of either the sounds (in the audio condition) or graphics (in the visual condition) used to provide feedback in the game. Participants played for a five minute training period, during which they were free to ask questions to clarify the game rules and interface.

The five minute duration of training was chosen based on experiences in a pilot study, which had used a shorter training period and had shown significant ordering effects. To attempt to reduce the ordering effects, we decided to lengthen the training period. Anecdotally, we found that there was an upper limit of approximately five minutes of training before the process became monotonous for the player, and employed this maximum length in order to try to reduce the ordering effects.

Next, the participant played the game again for five minutes. The participant was not permitted to ask questions of the experimenters. On completion, he completed the Witmer and Singer Presence questionnaire, and a custom questionnaire regarding his experience with the game.

This process was repeated for the second condition.

After completing both conditions, the participant completed a final custom questionnaire comparing the two interfaces. Finally, they completed a semi-structured interview discussing these preferences between the interfaces.
4.3 Results

In this section, we present the results of the ambient audio experiment. The results were divided into three categories: player performance, based on participants’ behaviour in the game; player presence scores, based on the participants’ subjective experiences reported with the Witmer and Singer Presence questionnaire; and player preference, based on participants’ feedback using a custom 9-question questionnaire (provided in appendix A).

4.3.1 Data Analyses

This experiment had a single factor design, comparing audio and visual interfaces, with an additional between-subjects factor comparing participant expertise levels of novice and expert. We conducted a Repeated Measures Multivariate Analysis of Variance (MANOVA) on our results data. This was used for the Presence questionnaire data, the participants’ scores, speeds, and number of collisions, using interface as a within-subjects factor and expertise as a between-subjects factor. $\alpha$ was set at .05.

We also checked whether the order of presentation of the interfaces had any effect on the dependent measures, by conducting a MANOVA that included the condition order as one of the factors. This determines whether the condition order had any significant effect on the dependent variables, and we found that there were no significant effects due to ordering.

We were able to perform this test because the condition order was counterbalanced for participants, so that a randomly chosen half of the participants experienced the conditions in reverse order. This provides results for participants experiencing
Table 4.1: Performance measures between two conditions

<table>
<thead>
<tr>
<th>Measure</th>
<th>Audio mean</th>
<th>Audio SD</th>
<th>Visual mean</th>
<th>Visual SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>8.33</td>
<td>3.09</td>
<td>11.25</td>
<td>3.94</td>
</tr>
<tr>
<td>Collisions</td>
<td>10.13</td>
<td>11.35</td>
<td>27.42</td>
<td>18.55</td>
</tr>
<tr>
<td>Speed (m/s)</td>
<td>5.34</td>
<td>1.55</td>
<td>6.32</td>
<td>2.07</td>
</tr>
</tbody>
</table>

Table 4.2: Performance measures between two groups

<table>
<thead>
<tr>
<th>Measure</th>
<th>Novices mean</th>
<th>Novices SD</th>
<th>Experts mean</th>
<th>Experts SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>8.13</td>
<td>3.41</td>
<td>11.46</td>
<td>1.74</td>
</tr>
<tr>
<td>Collisions</td>
<td>19.04</td>
<td>12.23</td>
<td>18.5</td>
<td>14.62</td>
</tr>
<tr>
<td>Speed (m/s)</td>
<td>4.81</td>
<td>1.38</td>
<td>6.83</td>
<td>1.42</td>
</tr>
</tbody>
</table>

both condition orders, and allows analysis for the existence and size of ordering effects. If there had been significant ordering effects, the condition order would have to be considered as a confounding variable in the experiment, although in some cases counterbalancing can help to cancel out the ordering effects when analyzing the entire group of participants.

The repeated measures MANOVA analysis was used for the Presence questionnaire data, the participants’ scores, speeds, and number of collisions, using interface as a within-subjects factor and expertise as a between-subjects factor. \( \alpha \) was set at .05. The Wilcoxon signed ranks tests for 2-related samples were used to check for differences between the two conditions in the questionnaires’ non-parametric responses.

### 4.3.2 Player Performance

There were main effects of interface on the score \( (F_{1,22}=18.8, \ p\approx.000, \ \eta^2=.46) \), the average speed \( (F_{1,22}=14.1, \ p=.001, \ \eta^2=.39) \), and the number of collisions \( (F_{1,22}=28.9, \ p\approx.000, \ \eta^2=.57) \). In all cases the values were higher when using the visual interface
Table 4.3: Presence measures between two conditions

<table>
<thead>
<tr>
<th>Measure</th>
<th>Audio mean</th>
<th>Audio SD</th>
<th>Visual mean</th>
<th>Visual SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Involvement (/84)</td>
<td>58.25</td>
<td>12.13</td>
<td>55.42</td>
<td>10.74</td>
</tr>
<tr>
<td>Sensory Fidelity (/42)</td>
<td>27.38</td>
<td>4.79</td>
<td>22.71</td>
<td>7.78</td>
</tr>
<tr>
<td>Adaptation/immersion (/56)</td>
<td>44.92</td>
<td>7.34</td>
<td>40.54</td>
<td>7.55</td>
</tr>
<tr>
<td>Interface quality (/21)</td>
<td>6.42</td>
<td>2.67</td>
<td>7.33</td>
<td>2.76</td>
</tr>
</tbody>
</table>

than when using the audio interface (see table 4.1 for the means and variances).

Although we observed a significantly greater frequency of collisions when using the visual interface than when using the audio interface, this could be an artifact of the fact that players generally progressed further in the game when using the visual interface, therefore encountering more challenging gameplay. On average players progressed through 9.2 levels (range: 3-14) when using the audio interface and 11.9 levels (range: 4-19) when using the visual interface. If we balance the number of levels for each player, (i.e., remove data for levels completed in one condition, but not the other by player), a pairwise t-test reveals that there are still significantly more collisions when using the visual (mean=0.070, SD=0.055) interface than when using the audio interface (mean=0.030, SD=0.037) (t23=3.9, p=0.001).

There were main effects of expertise on score ($F_{1,22}=9.10$, $p=0.006$, $\eta^2=0.29$) and speed ($F_{1,22}=12.5$, $p=0.002$, $\eta^2=0.36$), but not on collisions ($F_{1,22}=0.1$, $p=0.922$, $\eta^2\approx0.000$). Experts achieved higher speeds and higher scores than novices (see table 4.2).

There were no significant interaction effects of expertise and interface on any of the performance-related dependent measures.
4.3.3 Player Presence

Presence is captured via four measures: involvement, sensory fidelity, adaptation/immersion, and interface quality. There was a significant difference in the measures of sensory fidelity ($F_{1,22}=12.3$, $p=.002$, $\eta^2=.36$), adaptation/immersion ($F_{1,22}=8.5$, $p=.008$, $\eta^2=.28$), and interface quality ($F_{1,22}=6.8$, $p=.016$, $\eta^2=.24$) depending on interface. Players felt that the audio interface was more immersive and provided higher sensory fidelity, whereas the visual interface had a higher interface quality (see table 4.3 for means and variances). There was no effect of interface on involvement ($F_{1,22}=3.5$, $p=.073$, $\eta^2=.14$).

There were no significant effects of expertise on any of the presence-related measures, and no significant interactions of expertise and interface on the presence measures.

4.3.4 Player Preference

In questionnaires administered after using each interface, we asked players nine questions related to their play experience (see table 4.4 for means and variances). Players reported that their experience using the audio interface was more fun than using the visual interface ($Z=2.8$, $p=.005$) and that they found it easier to avoid obstacles when using the audio interface ($Z=3.5$, $p=.000$). In addition, players noted that the visual interface made it more difficult to play ($Z=2.8$, $p=.010$), and that they stopped pedalling more often when using the visual interface ($Z=2.3$, $p=.023$). All other questions did not result in significant differences.

By considering the significant results for experts separately from those for novices, we found that both experts and novices reported that it was easier to avoid obstacles.
CHAPTER 4. EVALUATION OF AMBIENT AUDIO INTERFACES

Table 4.4: Player preferences between two conditions (7-point Likert scale)

<table>
<thead>
<tr>
<th>Question</th>
<th>Audio mean</th>
<th>Audio SD</th>
<th>Visual mean</th>
<th>Visual SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>When an animal was far away, I could easily tell which direction it was in</td>
<td>5.00</td>
<td>1.69</td>
<td>5.63</td>
<td>1.56</td>
</tr>
<tr>
<td>I often had to stop pedalling when (listening to the ambient audio) / (looking at the map)</td>
<td>4.13</td>
<td>2.09</td>
<td>5.21</td>
<td>1.44</td>
</tr>
<tr>
<td>I found it easy to avoid the obstacles in the world while locating the animals</td>
<td>4.58</td>
<td>1.77</td>
<td>2.79</td>
<td>1.28</td>
</tr>
<tr>
<td>I felt rushed during the game</td>
<td>3.13</td>
<td>1.75</td>
<td>3.54</td>
<td>1.84</td>
</tr>
<tr>
<td>It was hard to use the (ambient audio) / (map) to locate the animals</td>
<td>2.83</td>
<td>1.55</td>
<td>2.75</td>
<td>1.62</td>
</tr>
<tr>
<td>I thought that the game was fun to play</td>
<td>5.67</td>
<td>1.40</td>
<td>5.00</td>
<td>1.62</td>
</tr>
<tr>
<td>When I was close to an animal, I had no trouble finding it</td>
<td>4.17</td>
<td>1.52</td>
<td>4.75</td>
<td>1.45</td>
</tr>
<tr>
<td>There wasn’t enough time to finish the game</td>
<td>2.25</td>
<td>1.22</td>
<td>2.46</td>
<td>1.56</td>
</tr>
<tr>
<td>I thought that the game was hard to play</td>
<td>2.79</td>
<td>1.47</td>
<td>3.79</td>
<td>1.50</td>
</tr>
</tbody>
</table>

using the audio interface (experts: Z=2.4, p=.016; novices: Z=2.6, p=.010). And both experts and novices reported that the audio interface was more fun, although only marginally so for experts (experts: Z=1.9, p=.058; novices: Z=2.0, p=.041). Only novices reported that it was harder to play using the visual interface (Z=2.3, p=.022); experts did not report this difference (Z=1.3, p=.204). The same trend was true for results of stopping pedalling more often using the visual interface, only marginally so (experts: Z=1.4, p=.156; novices: Z=1.8, p=.064).

4.4 Discussion

The results of the experiment are consistent with our hypotheses: players received better scores in the visual condition, but felt a stronger sense of presence and fewer collisions in the audio condition. We now discuss these results.
4.4.1 Performance of ambient audio

Players’ performance was worse using the ambient audio interface. Players averaged 8.33 points per game when using ambient audio, while they scored an average of 11.25 points per game using the visual interface. They also moved faster with the visual interface, at an average speed of 6.32 m/s during play, while players using ambient audio averaged 5.34 m/s. Participants’ comments in interviews indicate that a primary cause of this is that spatialized audio is less accurate than a visual map for locating animals. One participant stated “In the map it was easier to track the animals and to anticipate where they’re going to. The current location was easy to find in the auditory game, but where they were going wasn’t, and so it was more a game of following, so it became harder to catch them in the auditory game.” Another said “[The map has] easier information to rely on than the audio, particularly when you’re close to the animal and it switches from right to left and it keeps moving around, you’re spinning around in circles looking for something you can’t see.”

An interesting secondary effect is that players reported feeling more stress to perform in the visual condition. One participant reported: “I think that with the visual map with the animals, there was kind of pressure on me, that I have to catch these animals.” Other participants stated “If I knew on the visual task that the animal is across the map, that I have to go all the way over there and run really fast, it made that more stressful.” and “When I could see how fast the animal was going... it was like, ‘well now I have to pedal faster’.” This increase in stress and sense of urgency may also have contributed to the higher pedaling speed observed, and hence higher score.
4.4.2 Safety

We consider the number of collisions with obstacles as an indicator of the safety of the interface. The results indicate that the number of collisions was significantly higher in the visual condition. This is consistent with our hypothesis that when players shift their attention to the secondary map display, they have more difficulty seeing obstacles, and therefore are more likely to hit them.

Feedback from participants is consistent with this hypothesis. One participant said of the audio condition, “I didn’t have to keep jumping my eyes off the map to see where I was, and with how tightly [spaced] the some of the obstacles were, it was easier for me to just focus on avoiding obstacles than jumping back and forth.” Another said, “I could focus all my visual attention on the main screen, so I knew where I was going all the time. The map on the side was a lot more distracting.”

As a secondary effect, it is possible that the additional stress that players reported in the visual condition (as discussed in section 4.4.1) may have resulted in higher pedalling speed, which could have contributed to increased collision rate.

4.4.3 Immersion

Players reported higher sense of immersion in the audio condition. We hypothesized that this was due to the audio being overlaid directly onto the world, whereas in the visual condition, the world view and map view are not directly connected. Players’ comments lend weight to this hypothesis. One participant said: “In the map version, I was tearing myself away from the real world screen to look at the map... it was more engaging to look at the main screen and think, ‘Hey I could actually be there’, as opposed to thinking ‘I need to find out where I am so I’ll look at this other
Another said that the audio version “involved me more personally in the environment”.

Interestingly, several players reported that immersion was reduced in the visual condition by the absence of any sound at all. For example, “hunting around looking for silent animals doesn’t seem as realistic”, and “If there was some random rustling of leaves in the background that was not related to the animals but just happened to be there, I probably would have found it more immersive.” This implies that even when the ambient audio is not required as the main navigational mechanism, it could be included regardless to contribute to players’ sense of presence in the virtual world.

### 4.4.4 Expertise

We hypothesized that prior expertise with games that use radar views could counteract the reduced immersion and safety in the visual condition. In fact, our results showed that both novices and experts experienced higher sense of presence with the audio version, and both found it easier to avoid obstacles in the audio version. This is a positive result, indicating that the immersion and safety benefits of ambient audio carry over to gamers who are expert with current interface styles.

Some differences were seen. Both novices and experts rated the audio condition as more fun, but novices more strongly. Perhaps most interestingly, novices found the visual condition harder to play, but experts did not. These findings indicate that experts may be more comfortable with the limitations of using a secondary display.
4.4.5 Limitations of Study

Our study considered only one game, the simplified version of Growl Patrol. We argue that since the game consisted solely of the task of locating and chasing a virtual animal, the results are likely generalizable to navigational tasks in any location-based game. Further studies with different games would be beneficial in confirming this claim of generality.

A second limitation is that participants had only a brief experience of using ambient audio in this study. Participants had five minutes of ambient audio training and five minutes of play, for a total of ten minutes experience with ambient audio. It may be that the novelty of ambient audio and the short duration of exposure had some effect on the participants’ subjective reports of enjoyment and preference for the interface, and it would be interesting to examine participant preferences after extended experience.

By carrying out the study indoors, we gained several benefits: we were able to monitor players and collect detailed data more easily; we removed the problem of accurately localizing players’ positions and head orientations, and we removed the possibility of external interruptions to the game. The use of this controlled environment provided us with considerably better confidence in our results than would have been possible with an unconstrained exterior environment.

Our experience is not limited to the indoor version of the game, however. We have implemented Growl Patrol for play outside (as was shown in figure 3.3). Anecdotally, we have found the results to be similar to those observed in our formal study; however, problems with locating animals in the virtual condition were exacerbated by inaccuracies in the positional sensors. The GPS used to monitor position is updated
only once per second, leading to appreciable error if the player is moving quickly. A compass and gyroscope were used to monitor head-orientation. The compass is inaccurate when the player is moving, and the gyroscope is inaccurate when the player rotates his head quickly. Sensor hardware is continually improving, but hardware limitations must be considered when implementing ambient audio today.

An important limitation of ambient audio not considered in this evaluation is that there is a limit to how many audio streams players can track. Further research is required to establish how many audio streams can be effectively distinguished. Anecdotally, we have found the number to be three or four with the sound spatialization technology used for our ambient audio platform and in this study. This implies that visual interfaces can convey more information than ambient audio interfaces. For example, in our study, the secondary visual display showed the locations of buildings and the extent of the play area, neither of which were available to players using the audio interface. Our results show that ambient audio is useful for navigation-style tasks and for maintaining situational awareness while mobile. However, some forms of information (such as the full details of the map) must necessarily be visual in form.

Ambient audio is most useful when the player is moving. For parts of the game where players are stationary, the full richness of a visual display can be used without putting the user in an unsafe situation. For example, the PiNiZoRo game [26] combines a navigational task (which could be supported by ambient audio), and a set of visual mini-games that are played while stationary.

While we found that ambient audio lends itself well to representing the locations of objects surrounding the user, we have not tested ambient audio for the representation of more abstract types of data, such as was tested in the Walker and Brewster audio
progress bar study [29] described earlier. While certain presentation techniques based on that study or the results of Schmandt’s Audio Hallway study [21] might aid in representing other types of data with ambient audio, and though we expect that ambient audio would offer similar safety benefits to those found in our experiment, we cannot draw any conclusions about its effectiveness in communicating abstract data to users.

Our core lessons for game designers are therefore that ambient audio games should not require overly precise navigation, should not require players to track large numbers of information sources, and should not require mobile players to access information that is not easily representable in ambient audio form. Despite these restrictions, ambient audio is directly applicable to any location-based game that relies on navigational tasks, and can help increase players sense of presence in the virtual world, while improving their safety.

### 4.5 Conclusion

In this chapter, we discussed an experiment that compared ambient audio interfaces and visual interfaces for use while mobile. Using a location-based game based on Growl Patrol, we tested the performance, immersion, and safety of players when using these two interfaces. We also tested whether players with expertise in using a secondary display while visually navigating would see the same benefits from ambient audio as unexperienced players. While the visual interface did promote significantly better performance in-game from the players, ambient audio was perceived to be significantly more immersive and prevented significantly more dangerous collisions with obstacles. There was no effect of expertise on the benefits of ambient audio.
Based on these results, we recommend the use of ambient audio for location-based games and other applications used by pedestrians, provided that these applications can adapt to the interface conventions of ambient audio.
Chapter 5

TREC Toolkit for Ubiquitous Games

In location-based games, the player navigates around the physical world to control gameplay. This can involve the player directly controlling an avatar’s movement, or allowing the game to provide new options to players when they are physically occupying certain locations.

Location-based games rely on sensing a player’s position in the real world in order to give players control of the game based on their movements. If head tracking is also used, it can allow the game interface to accurately layer game objects or information over the real world, as in Augmented Reality (AR) applications. It can be time-consuming and difficult for developers to interface with and use location and head orientation sensors, due to the inaccuracy of these sensors and the inconsistency between different sensors.

As a solution to these problems, we have developed the TRacking with Extensible
Components (TREC) toolkit to handle location and orientation input in mobile augmented reality applications and location-based games. TREC’s goals are to reduce the time required to develop interfaces for hardware sensors, and to reduce the difficulty of adapting AR applications to work with a particular collection of sensors. To achieve this goal, reads tracking data from any sensors that are available, increases the accuracy of this tracking data if possible, and makes it available to the developer in an abstracted form that is independent of the sensors being used.

5.1 Difficulties in developing location-based applications

As previously described in Chapter 2, developers face difficulties when writing location-based applications and games. We have identified five problems in total that can consume developer time when developing these applications. The following three problems can arise when developers attempt to interface directly with hardware tracking sensors:

1. *Noise problem* - The data being provided by the sensor hardware may be noisy and inaccurate. If this data is used directly as input, it could cause inaccuracy and erratic behaviour in the user’s display of the virtual world.

2. *Format problem* - The raw readings produced by a sensor may require transformation to provide the desired property, or they may be in an inconvenient form for the application domain. For example, if the desired property is direction, but only the current rate of rotation can be directly measured, we must
use the rotation rate and prior knowledge of the system to calculate the current direction before it can be used in the application. Similarly, GPS receivers measure location, but only in longitude and latitude. If the application uses a local cartesian coordinate system for location data, the longitude and latitude location data must be converted to local coordinates before it can be used in the application.

3. **Sensor inconsistency problem** - Using raw sensor readings may also make the application dependent on the hardware being used, creating a coupled system that may be difficult to adapt to different sensors. For example, it may be hard to replace an accelerometer with a gyroscope, or even to replace it with an accelerometer made by a different manufacturer.

Existing sensor frameworks address some of these problems. The following two issues, however, remain problems when using most existing frameworks, as well as when attempting to interface directly with the sensors hardware:

4. **Early configuration problem** - If the sensors being used for tracking must be chosen and configured before runtime, there is a risk that the application will be inoperable if different sensors are available at runtime. This makes it harder to distribute an application capable of running on different hardware platforms. It also makes it difficult to allow recovery from device failures at runtime, such as those cause by batteries running out.

5. **Extensibility problem** - It can be difficult to extend existing code with support for completely new sensors or new algorithms for mitigating sensor inaccuracy (e.g., basic filtering, sensor fusion). New sensors may introduce new types of
data or new formats, which may require extensive modifications in the sensor and application code when written from scratch. The introduction of new data types or formats may simply not be possible in some frameworks, limiting their extensibility.

The TREC toolkit seeks to solve all five of these problems, making it especially suitable for developing location-based applications. The novel feature of TREC that helps to solve each problem is its three-layer architecture.

### 5.2 The TREC three-layer architecture

The TREC framework is structured around a three-layer architecture: the Abstract Input Layer, the Abstract Device Layer, and the Device Layer.

The lowest layer, the *device layer*, contains objects that expose the data provided directly by device sensors. Devices must implement one or more abstract device interfaces (see below). For example, the OceanServer USB Compass provides both compass and accelerometer functionality. To add a new device to the framework, a developer needs to create a device-layer class for the device.

The *abstract device layer* groups standard types of sensor devices, and also provides virtual sensors by fusing the data from multiple concrete devices. For example, the *CompassOrientation* class provides orientation data from compass-like devices, such as OSCompass, while the *GyroscopeOrientation* class provides orientation data from gyroscope-like devices. Meanwhile, *FusedOrientation* provides orientation information by fusing data from a number of devices.

Finally, the *abstract input layer* provides interfaces for specific input data types.
In the current version of the framework, two types of input are defined: \textit{IOrientation} provides head orientation data and \textit{ILocation} provides position information. To access input, an application queries the TREC device manager for one of these types of input, and receives an object in return implementing the appropriate abstract input type. In this way, the TREC framework shields applications from the details of individual devices or fusion algorithms, and simplifies the work of the developer by providing a simple and uniform way to access input information.
5.3 Addressing the development difficulties with TREC

1. **Noise problem** - TREC attempts to ameliorate the noise problem by using filtering and supporting sensor fusion algorithms. TREC uses sensor fusion algorithms automatically when a supported set of sensors is available. This automatic use of sensor fusion algorithms is made possible by the layers of abstraction built into the TREC architecture, requiring no changes to the application code when changing input configurations.

2. **Format problem and Sensor inconsistency problem** - The format problem and the sensor inconsistency problem result from the sensor hardware providing its data in a fixed format, which might be inconvenient for the application domain, and can make the application overly dependent on the hardware being used. In order to make the application programmer’s job easier, TREC abstracts all sensor data into a high level representation of location and orientation. These are directly useable in a location-based mobile application and will not change if the hardware changes. TREC provides the application programmer with simple interfaces for these two properties in its abstract input layer.

4. **Early configuration problem** - The early configuration problem can be handled automatically by TREC. TREC configures the sensors once each time it is run by determining which of the connected sensors can be used to provide the best data to the application. Because the TREC layered architecture hides all differences between sensor hardware, it can provide the application plug-and-play compatibility with different sensors. This could also permit TREC
to change configurations multiple times while running, such as when sensors cease to function, although this functionality has not been programmed into the current implementation.

5. *Extensibility problem* - TREC’s architecture makes it easy to address the extensibility problem and extend the framework to support new fusion algorithms and sensors. It uses a three-layer hierarchy (see figure 5.1) to abstract device details. This allows newly added sensors in the *device layer* to work automatically with existing applications, and fusion algorithms in the *abstract device layer* can take advantage of the abstracted devices automatically. The framework is open and allows access and modification at any level, meaning low-level sensor data can always be accessed directly if necessary.

In summary, the three-layered architecture of TREC allows it to automatically provide less noisy and more accurate data to applications, and provide this data in a standard format that decouples applications from sensor hardware. TREC’s architecture also allows it to be easily extended to support new sensors or algorithms, which will then immediately work in existing applications.

### 5.4 Applying TREC

We now show, by example, how TREC can help both application programmers creating a mobile AR system, and systems programmers adding new functionality to TREC itself.
5.4.1 The Application Programmer’s Perspective

To illustrate how TREC helps in developing mobile AR applications, we examine how the Noisy Planet tourist application was implemented using the TREC toolkit. Users of the application carry a mobile device containing a GPS, and wear a gyroscope and compass for head tracking. These sensors are hidden behind TREC’s IOrientation and ILocation abstract input types.

To access the abstracted sensor data, the Noisy Planet code requests an appropriate data source from TREC’s Device Manager. E.g., when an orientation data source is requested, the Device Manager provides an object that adheres to the IOrientation interface, with data coming from either the OSCompass or WTGyroscope device. The application cannot tell which device is supplying this data, and does not need to know since the incoming data is in a standardized format. This allows TREC to provide the best available sensor, and allows new devices to be added to TREC without impacting application code. If the programmer for some reason preferred a compass, he/she could choose to access the CompassOrientation object in the abstract device layer to guarantee the use of a compass, or even directly access the OSCompass in the device layer to manipulate that specific device.

In addition to choosing from available sensors at runtime, TREC can automatically use sensor fusion algorithms when sufficient sensors are available. Noisy Planet actually receives a FusedOrientation object (from the abstract device layer) after the earlier request for an orientation object, if the required devices are available. This happens without any changes to or configuration of the application. TREC decides which device or combination of devices to use based on an internal rating mechanism, which is currently a hard-coded list of the known devices.
For comparison, we implemented a version of Noisy Planet without access to TREC, while providing the same functionality. This implementation required approximately 250-350 lines of code per sensor (GPS, compass, gyroscope) to handle serial input from these devices, in addition to another 80 lines to manage these sensors and perform the sensor fusion. The version using TREC requires just two method calls: one to get the current orientation, and another for the current location.

5.4.2 The Toolkit Developer’s Perspective

The TREC architecture is open, making it straightforward to add support for new devices. We now discuss the experience of adding new hardware and a new virtual device to TREC.

Adding a New Hardware Device

In order to add a hardware device, a class needs to be added to the device layer that can communicate with the hardware and make the data available with methods that adhere to the corresponding interface(s) in the abstract device layer.

For example, if we wanted to add a new combination accelerometer/gyroscope device from “ABC Sensor Corp” to the hierarchy, first a new class called ABCAccel-Gyro would be created at the device layer, as shown in figure 5.2. This object would contain all the custom code for communicating with the hardware, such as reading a serial connection and using a parser to capture the sensor readings and store them internally. When accessor methods are called on an object of this class, they provide the stored data in the correct format.
In order to match the IGyroscope and IAccelerometer interfaces, the ABCAccelGyro class would have to provide at least the AccX(), AccY(), AccZ(), and AngularVelocity() accessor methods. In addition, other methods can expose the unique data available from this sensor, such as Temperature().

Once this class is created and the DeviceManager is made aware of its existence (currently implemented using a text file that lists available devices) any application that uses TREC can immediately use the ABCAccelGyro. In addition, any fusion algorithm that uses an IGyroscope or IAccelerometer can also begin using the device
without any changes, like the FusedOrientation class in the abstract device layer shown in figure 5.2.

**Adding a Simple Sensor Fusion Algorithm**

TREC’s layered architecture allows easy integration of sensor fusion algorithms into applications, and allows the fusion code itself to be shielded from the details of the underlying devices. Sensor fusion algorithms are written as abstract device modules that adhere to a standard interface. Therefore, any program taking advantage of TREC’s abstract input modules (*IOrientation* and *ILocation*) can automatically use the new algorithm.

For example, to implement the *FusedOrientation* class of figure 5.1, a new abstract device class is created in the abstract device layer. This class implements the *IOrientation* interface, and therefore can be used whenever orientation information is requested by the application. Inside *FusedOrientation*, a gyroscope, compass and accelerometer are used (each of which are abstract devices), and some kind of location service is used (an abstract input.) Specifically, the gyroscope is used to determine orientation; the compass is used to calibrate the gyroscope from time to time, but not when the compass is moving (as determined via the accelerometer or from changes in location).

This example shows how the fusion algorithm leverages TREC’s layered architecture. When necessary, the fusion algorithm knows the kind of device that it is using (gyroscope, compass, etc.), but does not need to know exactly which device is in use. Furthermore, when it is not necessary to know the kind of device (i.e., for the location service), the appropriate abstract input can be used.
5.5 Conclusion

In this chapter we detailed the TREC toolkit and the way its architecture addresses the five major problems associated with using tracking sensors when developing location-based games and applications. TREC uses a three-layer hierarchy of increasingly abstract data sources to address these problems, by making the application independent of the specific hardware sensors and fusion algorithms being used for tracking.

We demonstrated the use of TREC in a simple tourist guide application called Noisy Planet, reducing many lines of sensor code to two method calls. The extensibility of TREC was also shown by walking through the steps to adding a new device and a new fusion algorithm.
Chapter 6

Summary and Conclusions

This thesis has presented work addressing two problems with applications for mobile devices: the difficulties of using handheld visual interfaces while walking, and the difficulties of developing applications that take advantage of tracking sensors to control the application. We now present a summary of the findings in this thesis.

6.1 Summary

Mobile applications present problems to both their users and developers. Pedestrians can have issues with the usability of application interfaces if they are presented using handheld, visual displays. Walking users may have a difficult time paying attention to their environment if they must repeatedly glance at a small screen. They also might find it hard to feel immersed in a location-based game if they must mentally calculate the locations of objects and entities around them, based on a map provided on the small handheld screen.
In addition, it can be time-consuming for developers to develop mobile applications and games that make use of the user’s location. The location tracking data must be interpreted from a set of sensors on or around the user, and it can require significant developer time to access this sensor data and use it in an application. We have identified five time-consuming problems with using sensor data that can pull developers away from application development, including noisy and inaccurate sensor readings, incorrect data formats, inconsistencies between sensor interfaces, difficulties with static configuration of the sensors, and the difficulties in extending existing code to support new sensors.

We have developed two potential solutions to address these problems with mobile application development and usability. To address the immersion and safety problems of using mobile applications while walking, we have developed an interface called ambient audio. Ambient audio uses streams of sound to provide feedback to users over headphones. These streams of sound can be used to represent virtual entities in the user’s environment, such as a homing beacon or a pursuing enemy.

Ambient audio is designed to be more immersive than a map-based visual display because it presents virtual entities in situ in the environment. Map users must mentally model the locations of these entities based on quick glances at the map, which may feel less immersive than hearing noises that appear to originate from an entity’s actual location. Ambient audio may also be safer to use than handheld visual displays, because the user’s visual attention is always free to focus on his environment.

We also developed the TREC toolkit to address the time-consuming problems of accessing location sensors when developing location-based games. TREC uses a three-layer hierarchy of devices to abstract away the differences between sensors, and
to help solve the five problems we have identified above. Using TREC, we have developed two fully functional location-based applications, showing that the TREC hierarchy is functional, and that it can save developer time during development.

We also ran an experiment to test ambient audio, comparing it to a handheld visual interface for use with a location-based game. In this experiment, we found that ambient audio was both significantly safer and more immersive to use than a visual interface, based on the number of dangerous collisions that participants experienced while playing the game, as well as their recorded perceptions of the immersiveness of the game. We did find, however, that players were significantly better able to perform the in-game tasks while using the visual interface, which may be a result of the increased accuracy and resolution of visual interfaces compared to our implementation of ambient audio.

6.2 Future Work

The experiment presented in this thesis has given a strong indication that ambient audio can be a practical alternative to visual interfaces for mobile devices, depending on the goals of the interface and the nature of the application. This has only been demonstrated indoors, however, using a simulated outdoor environment. A logical next step for ambient audio research would involve testing ambient audio outdoors in the real world, rather than the indoor simulation that was used for the experiment we presented here. There may be unknown factors that affect the safety, immersion, and performance of ambient audio when used outdoors, with less reliable user position tracking and with the potentially distracting visual and aural stimuli present in a typical public outdoor environment.
The TREC framework would benefit from further rigorous testing of its abilities. Experiments should be performed to test the quality of user tracking provided by TREC compared with other sensor frameworks. In order to provide a better measure of TREC’s usefulness, a test could be performed to measure how easily a number of popular sensor fusion algorithms can be implemented using TREC’s abstract sensor interfaces, and how the performance of these general algorithms varies from the performance of a custom implemented fusion algorithm that uses the same sensors.

6.3 Conclusion

Ambient audio and TREC have been shown to address two issues with mobile applications: that handheld visual interfaces can be unsafe and un-immersive to use for mobile devices, and that interfacing with tracking sensors can be time-consuming when developing a location-based application. We conclude that each solution shows promise for use in future mobile applications, and that there are clear steps to be taken in the future to further assess the quality of these two solutions.
Bibliography


Automatic configuration of pervasive sensor networks for augmented reality. 


[23] Marcos Serrano, Laurence Nigay, Jean-Yves L. Lawson, Andrew Ramsay, Rod-


Appendix A

Custom Questionnaires

Included in this appendix are the custom questionnaires used in the ambient audio experiment. The first questionnaire in figure A.1 was given to participants having completed the ambient audio condition, and the second in figure A.2 was given to participants having completed the visual condition.
POST-CONDITION QUESTIONNAIRE: AUDIO

This part provides statements about the game you just played. Read the following items and circle the number that best expresses how you feel about the statement. Please answer by circling off the following 7-point scale:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Disagree</td>
<td>Neutral</td>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>When an animal was far away, I could easily tell which direction it was in.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I often had to stop pedaling when listening to the ambient audio.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I found it easy to avoid the obstacles in the world while locating the animals.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I felt rushed during the game.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>It was hard to use the ambient audio to locate the animals.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I thought that the game was fun to play.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>When I was close to an animal, I had no trouble finding it.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>There wasn’t enough time to finish the game.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I thought that the game was hard to play.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>

Figure A.1: Ambient audio condition questionnaire
APPENDIX A. CUSTOM QUESTIONNAIRES

POST-CONDITION QUESTIONNAIRE: VISUAL

This part provides statements about the game you just played. Read the following items and circle the number that best expresses how you feel about the statement. Please answer by circling off the following 7-point scale:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Strongly Disagree</td>
<td>Neutral</td>
<td>Strongly Agree</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>When an animal was far away, I could easily tell which direction it was in.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I often had to stop pedaling when looking at the map.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I found it easy to avoid the obstacles in the world while locating the animals.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I felt rushed during the game.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>It was hard to use the map to locate the animals.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I thought that the game was fun to play.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>When I was close to an animal, I had no trouble finding it.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>There wasn’t enough time to finish the game.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>I thought that the game was hard to play.</td>
<td>1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>

Figure A.2: Visual condition questionnaire
Appendix B

Expertise Questionnaire

Included in this appendix is the Expertise Questionnaire that was used to evaluate a participant’s level of expertise in playing video games that include radar maps.
EXPERTISE IN VIDEO GAMES QUESTIONNAIRE

Read the following items and circle the number that best expresses how you feel about the statement. Please answer by circling the appropriate answer from the following 5-point scale:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>How often do you use computers?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you use visual car navigation systems?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you play computer/video/console games?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you play FPS (First Person Shooter) games?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you play a game pad?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you play using a bicycle accessory?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How often do you play games using a radar-view map?</td>
<td>1  2  3  4  5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statement</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>How much time do you spend using computers?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend using visual car navigation systems?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend playing computer/video/console games?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend playing FPS (First Person Shooter) games?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend playing using a game pad?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend playing using a bicycle accessory?</td>
<td>1  2  3  4  5</td>
</tr>
<tr>
<td>How much time do you spend playing games using a radar-view map?</td>
<td>1  2  3  4  5</td>
</tr>
</tbody>
</table>

Age: __________
Gender: M / F

Figure B.1: Expertise questionnaire for all study participants
Appendix C

Research Ethics Board Approval

Included in this appendix is a letter from the Queen’s General Research Ethics Board (GREB), indicating that the study described in chapter four complies with the Tri-Council Guidelines and the Queen’s ethics policies.
APPENDIX C. RESEARCH ETHICS BOARD APPROVAL

July 29, 2010

Mr. John Kurczak
Master's Student
School of Computing
Queen's University

GREB Ref #: GCISC-043-10
Title: “Ambient Audio Interfaces for Outdoor Gaming”

Dear Mr. Kurczak:

The General Research Ethics Board (GREB), by means of a delegated board review, has cleared your proposal entitled “Ambient Audio Interfaces for Outdoor Gaming” for ethical compliance with the Tri-Council Guidelines (TCPS) and Queen’s ethics policies. In accordance with the Tri-Council Guidelines (article D.1.6) and Senate Terms of Reference (article G), your project has been cleared for one year. At the end of each year, the GREB will ask if your project has been completed and if not, what changes have occurred or will occur in the next year.

You are reminded of your obligation to advise the GREB, with a copy to your unit REB; of any adverse event(s) that occur during this one year period (details available on webpage http://www.queensu.ca/or/researchethics/GeneralREB/forms.html – Adverse Event Report Form). An adverse event includes, but is not limited to, a complaint, a change or unexpected event that alters the level of risk for the researcher or participants or situation that requires a substantial change in approach to a participant(s). You are also advised that all adverse events must be reported to the GREB within 48 hours.

You are also reminded that all changes that might affect human participants must be cleared by the GREB. For example you must report changes in study procedures or implementations of new aspects into the study procedures on the Ethics Change Form that can be found at http://www.queensu.ca/or/researchethics/GeneralREB/forms.html - Research Ethics Change Form. These changes must be sent to the Ethics Coordinator, Gail Irving, at the Office of Research Services or irvingg@queensu.ca prior to implementation. Mrs. Irving will forward your request for protocol changes to the appropriate GREB reviewers and I or the GREB Chair.

On behalf of the General Research Ethics Board, I wish you continued success in your research.

Yours sincerely,

Joan Stevenson, PhD
Professor and Chair
General Research Ethics Board

c.c.: Dr. Nicholas Graham, Faculty Supervisor

JS/gi

think Research
think Queens

SHIPPED AUG 03 2010

Figure C.1: GREB approval for ambient audio study