# *Using Earcons to Improve the Usability of a Graphics Package*


## Stephen Brewster

*Glasgow Interactive Systems Group, Department of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK.*

*Tel: +44 (0)141 330 4966,*
*Email: stephen@dcs.gla.ac.uk,*
*Web: http://www.dcs.gla.ac.uk/~stephen/*



**This paper describes how non–speech sounds can be used to improve the usability of a graphics package. Sound was specifically used to aid problems with tool palettes and finding the current mouse coordinates when drawing. Tool palettes have usability problems because users need to see the information they present but they are often outside the area of visual focus. An experiment was conducted to investigate the effectiveness of adding sound to tool palettes. Earcons were used to indicate the current tool and when tool changes occurred. Results showed a significant reduction in the number of tasks performed with the wrong tool. Therefore users knew what the current tool was and did not try to perform tasks with the wrong tool. All of this was not at the expense of making the interface any more annoying to use.**



**Keywords:** Earcons, sonically–enhanced widgets, sound, interface sonification.


## 1 Introduction

This paper describes how the usability of a graphics package can be improved by the addition of non–speech sound. It might not be immediately obvious how sounds could be used in a graphics package which is, of course, highly visual by its nature. One problem with modern graphical displays is that they are very visually demanding; all information is presented graphically. As has been demonstrated, this can cause users to become overloaded and to miss important information (Brewster, 1997). One reason is that our eyes cannot do everything. Our visual sense has a small area of high acuity. In highly complex graphical displays users must concentrate on one part of the display to perceive the graphical feedback, so that feedback from another part may be missed as it is outside the area of visual focus (Brewster & Crease, 1997). This problem is worse for partially sighted users whose area of acuity may be reduced by problems such as tunnel vision. Sound also does not take up any screen space and so does not obscure parts of the display, it is good at getting our attention whilst we are looking at something else and it does not disrupt our visual focus.

As an example, imagine you are working on your computer creating a drawing and are also monitoring several on–going tasks such as a compilation, a print job and downloading files over the Internet. The drawing task will take up all of your visual attention because you must concentrate on what you are doing. In order to check when your printout is done, the compilation has finished or the files have downloaded you must move your visual attention away from your picture and look at these other tasks. This causes the interface to intrude into the task you are trying to perform. It is suggested here that some information should be presented in sound. This would allow you to continue looking at your drawing but to hear information on the other tasks that would otherwise not be seen (or would not be seen unless you moved your visual attention away from the area of interest, so interrupting the task you are trying to perform). Sound and graphics can be used together to exploit the advantages of each. In the above example, you could be looking at the drawing you are creating but hear progress information on the other tasks in sound. To find out how the file download was progressing you could just listen to the download sound without moving your visual attention from the drawing task.

In this paper the use of non-speech sound, and not speech, is suggested. Speech is slow and serial (Slowiaczek & Nusbaum, 1985); to get information the user must hear it from beginning to end. Speech is similar to text, whereas non-speech sounds are similar to graphical icons. An icon, or non-speech sound, can concisely represent a concept whereas many words may be needed to describe it. The non-speech sounds described later in the paper are very short (the longest was less than 0.5 sec.) and can communicate their meaning quickly, but speech sounds would take longer. Speech is also language dependent, whereas non-speech sound is universal. For these reasons this paper suggests the use of non-speech sound in this case.

Even though sound has benefits to offer it is not clear how best to use it in combination with graphical output. The use of sound in computer displays is still in its infancy, there is little research to show the best ways of combining these different media (Alty, 1995, Brewster *et al.*, 1994). This means sounds are sometimes added in *ad hoc* and ineffective ways by individual designers (Barfield *et al.*, 1991, Portigal, 1994). This paper describes the addition of sound to a graphics package (see Figure 1) as an example of how it can be used effectively. Such packages are visually demanding and users may become visually overloaded when they have to look at the drawing they are working on and the interface to the package. Sounds were added to tool palettes to stop tool mis-selection errors, and for indicating the cursor position when drawing.

## 2 Previous uses of sound

One of the first attempts to use sound in a graphical interface, in this case the Macintosh Finder, was Gaver with the *SonicFinder* (Gaver, 1986, Gaver, 1989). The SonicFinder added sounds for selecting different types of items, dragging and copying. The extra sonic feedback gave users information that they could not see, for example file size and type when selecting. The same approach was taken in sonifying the graphics package; sounds were to give users information that they could not see without taking their eyes off the task in which they were involved. However, one thing that Gaver did not do was formally experimentally test his sonic enhancements to see if they actually improved usability. His results were more anecdotal. The addition of sounds to the graphics package had to be fully tested to ensure they improved usability.

Brewster and colleagues have successfully improved the usability of buttons, scrollbars and menus with non-speech sound (Brewster, 1997, Brewster & Crease, 1997 and Brewster *et al.*, 1994). Sounds were added to help users in situations where they could not see graphical feedback they needed. For example, sounds were added to pull–down menus because users may slip off the item they want on to one above or below (Brewster & Crease, 1997). This is often not noticed as their visual attention is elsewhere. Sonic enhancements reduced time taken to recover from errors, time taken to complete tasks and workload without any increase in subjective annoyance. Beaudouin-Lafon & Conversy (1996) added sound to overcome usability problems in scrollbars. They used an auditory illusion called Shepard–Risset tones which increase (or decrease) in pitch indefinitely (similar to the Escher drawing of an endless staircase). When the user was scrolling down a continuously decreasing tone was used, when scrolling up an increasing one. If scrolling errors occurred then the user would hear tones moving in

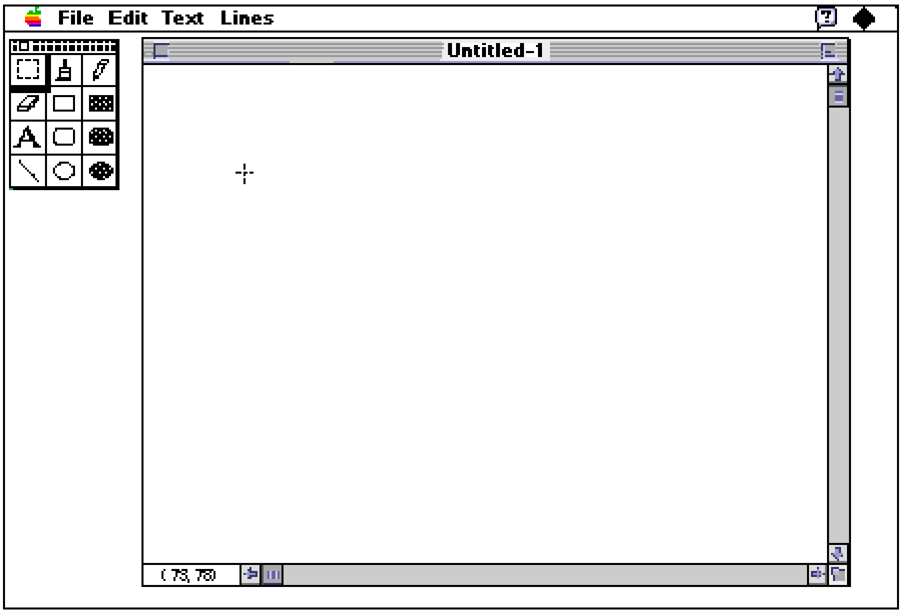


**Figure 1**: Screen–shot of the graphics package used.

the wrong direction. Results from these earlier experiments suggested that sound would be effective in overcoming the problems in a graphics package. Therefore, the same approach was used here.

Rigas & Alty (1997) used non-speech, musical sounds in the interface to a graphics package for blind people. They were able to use sounds to present the layout and structure of simple graphical images. To represent coordinate locations they used a *note-sequence* technique: notes were used to represent each of the points between the origin and the current location; the sequence of notes was then played to indicate the location. Results were favourable within a small (40x40) grid. In the graphics package described in this paper there could be a very large grid (up to 1024x768). Therefore it would not be possible to play a sequence of notes as Rigas & Alty had done because this would take too long. However, Rigas & Alty showed that it was possible, suggesting that, with some adaptation, cursor location could be presented in sound.

The commercial graphics application KidPix™ by Brøderbund Software also uses sound to make it more engaging (Kramer, 1994) for its users (who are children). The sound effects it uses are generally added for amusement but do indicate the drawing tool that is currently being used. For example the pencil tool sounds similar to a real pencil when used (making a scratching sound). These sounds were not an attempt to improve usability but to make the interface more engaging. It is hoped that the graphics package described in this paper can take advantage of the added engagement provided by sound but also improve usability.

One of the problems with graphics packages is that they often have modes (see section 4 for more details). In early work, Monk (1986) suggested that non-speech sounds could be used to overcome mode errors. In an experiment he tested the use of keying-contingent sounds to indicate different modes in a simple chemical plant simulation. Participants had to type in codes to transfer oxygen to reactors to keep the plant running. Errors occurred when switching between column-identifier mode and oxygen-addition mode. His results showed that with sounds one third less mode errors occurred, indicating that sound could provide valuable feedback on mode state. The research described in this paper builds on this earlier work.

## 3 Overall structure of the sounds used

The sounds used were based around structured non–speech musical messages called *Earcons* (Blattner *et al.*, 1989, Brewster *et al.*, 1993). Earcons are abstract, musical sounds that can be used in structured combinations to create audio messages to represent parts of an interface. The earcons were created using the earcon guidelines proposed by Brewster *et al.* (1995).

The widgets in any application form a hierarchy. For example, a simple application might bring up a window as in Figure 2. This window is made up from a frame which contains a menu bar, a listbox and a scrollbar. The menu bar contains two menus and, in turn, these menus will contain menu items. The windowing system will represent this as a hierarchy of widgets. This hierarchical structure was used to define the sounds for the graphics package.

The overall structure of the sounds was as follows: The application had its own spatial location (via stereo position) as a base for all of its sounds (just as an application has a base frame for all of its widgets). All widgets within the application used this and added to it by changing timbre, rhythm, pitch, etc.

This approach to allocating sounds is the same as for graphical widgets: A graphical application has a base frame for its widgets. This frame has a spatial location and any widget drawn within the frame uses the base spatial location to position itself. If the frame is moved then all of the widgets move accordingly. In terms of the sounds, if the application is moved all of the sounds play from a new stereo location.

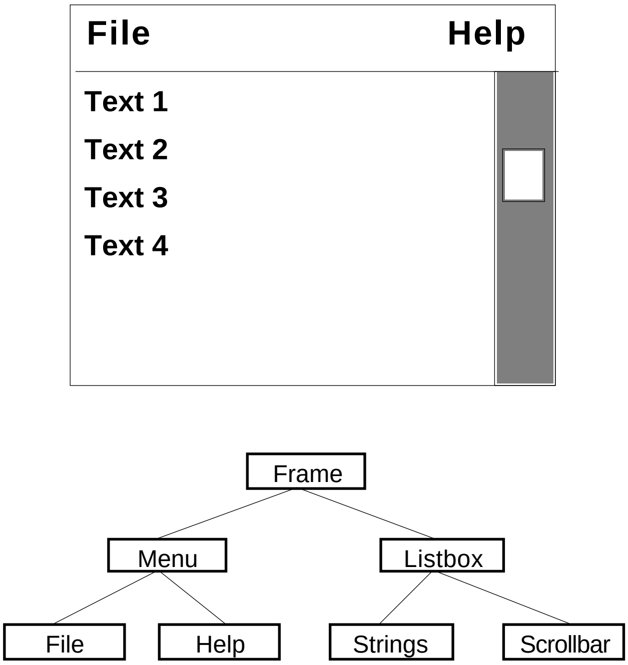


**Figure 2**: A simple window with several widgets and the hierarchy used to represent them.

Each of the main components was given a different timbre, as suggested by the earcon construction guidelines (Brewster *et al.*, 1995). For example, the selection tool was given a marimba timbre and the other drawing tools a trumpet (for more detail on this see below). For drawing or selecting horizontally a higher pitched sound was used than for drawing/selecting vertically (see below).

The overall structure of the sounds has now been described. In the next sections the specific uses of sound and the usability problems to be addressed will be dealt with. In particular, the use of earcons to correct problems with tool palettes and drawing will be discussed.

## 4 Problems with tool palettes

Tool (or button) palettes are a common feature of most graphical interfaces and especially graphics packages. One reason for this is that they allow the user easy access to a set of tools and indicate which tool is currently active (see (a) in Figure 3). Put another way, palettes are mode indicators; they allow the user to set the mode and then indicate what the current mode is (Dix *et al.*, 1993). Figure 3(a) shows a set of standard tools from a graphics package (shown in

Figure 1). The currently selected tool (in this case the rectangle drawing tool) is highlighted by changing its border.

In some systems (for example the Microsoft Word drawing package) after one rectangle has been drawn the system will change back to the default tool, often the selection tool (the dotted square at the top left of (a) in Figure 3). In other systems (for example Adobe Illustrator) the tool will remain selected until the user changes it (the tools in some packages use a mixture of both of these methods). There is a hybrid of these two (for example ClarisDraw) where the user can single–click a tool for it to be selected once or double–click for it to remain permanently selected. This method has the advantage that users can choose whether they want to stay in a drawing tool or revert back to the selection tool – it is more flexible. Figure 3(a) shows an example of the different feedback indicating a single click and Figure 3(b) shows a double click on a tool icon.

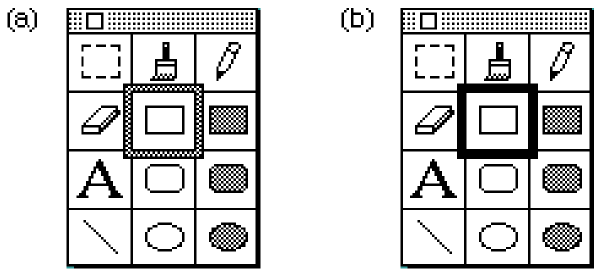


**Figure 3:** Rectangle tool selection by (a) single click, and (b) double click.

Interaction problems occur because users may not notice the currently active tool. In a graphics package users will be occupied with the drawing task they are doing (perhaps drawing a series of rectangles) which will require their full visual attention. This means that they will not be looking at the palette to see the current tool. If the system switches back to a default tool users may try to draw another rectangle but end up using the selection tool by mistake. If, on the other hand, the system remains in the current tool they may draw another rectangle by mistake when they really wanted to position the rectangle just drawn. These problems are exacerbated by the hybrid system because it is less predictable as the user may not remember if the current tool was single or double clicked.

### *4.1 Overcoming the problems of tool palettes*

In order to solve the problems of tool mis–selection users must get the right feedback to ensure that they know what is going on

(Reason, 1990). This paper suggests using auditory feedback to solve the problems. Why use sound, why not just use extra graphical feedback? It is difficult to solve these problems with extra graphics. Graphics displayed on the palette will not be seen by users because their attention will be on the drawing task they are engaged in. The visual system has a narrow area of focus which means that users cannot look at the palette as well as their main task. Information could be displayed at the mouse location – often the shape of the cursor is changed to reflect the current tool. This has some effect but if the different cursors are too big they will obscure the drawing underneath or if they are too small they will be too subtle to be noticed by users who are concentrating on the drawing they are creating and not the cursor. Sellen *et al.* (1992) have also demonstrated that the use of even gross graphical feedback (changing the colour of the entire screen) can be ineffective at indicating modes. Therefore, a different approach was needed. If we give tool information in sound then it will not obscure the display, it will be noticeable and we do not need to know where users are looking in order for them to perceive it. If users must look at the palette then it forces them to stop what they are doing for their main task and causes the interface to intrude; sound does not have such drawbacks.

### *4.2 Earcons for the tool palette*

Earcons were needed to indicate the currently active tool (the hybrid system of single and double clicking of tools will be used as an example). The main problems with the tool palette occur when switching from one tool to another. If the user does not know a switch has occurred (or conversely does not know that the same tool is still active) then errors will result as the wrong tool will be used.

An earcon was played when a tool was chosen. This occurred when (a) the user clicked on a new tool or (b) after he/she had finished drawing. In (a) this could be a single or double–click sound. In (b) if no tool change occurred (i.e. the user had doubled–clicked a tool) the same tool earcon was played again to reinforce that the tool had not changed, otherwise a sound indicating a switch back to the default tool was played.

The earcons were created using the earcon guidelines proposed by Brewster *et al.* (1995). The default selection tool was given a marimba timbre and the other tools a trumpet timbre. Only two instruments were needed because any automatic tool changes would always be from a drawing tool to the default tool. This situation was where errors were likely to occur so these changes had to be made salient to the user. The difference between these instruments when the earcons were played would grab the users' attention (Brewster *et al.*, 1995).

For a single-click selection one 100 msec. note at pitch $C_3$ (261Hz) was played. When a tool was selected by a double click the user heard the single-click earcon, to indicate a change in tool, and then two 100 msec. notes at a higher pitch, $C_2$ (523Hz), to indicate a double-click selection. These sounds were played in the timbre of the tool selected.

The intensity of sound was not used to make the earcons grab the users' attention to alert them to a tool change. Instead, a change in the earcon's instrument and pitch were used. In general, intensity changes should be saved for the most important events that must be communicated (for example, serious errors) because intensity has the most potential for annoyance (Berglund *et al.*, 1990, Brewster *et al.*, 1995).

The advantage of earcons used in this way is that they can stop errors happening. In previous experiments Brewster *et al.* used sound to help users recover from errors more quickly (Brewster & Crease, 1997, Brewster *et al.*, 1994) but did not stop the errors from occurring in the first place. In this case, the sounds alert users to the next tool that will be used and therefore if that is the wrong tool they can choose the correct one before beginning to draw, so avoiding the error in the first place.

## 5 Problems when drawing

The other (related) problem is that of coordinate display when drawing. Many graphics packages give cursor coordinates to the user in the form of a small box at the bottom edge of the drawing area (see Figure 1) or by rulers around the edge of the drawing area. When users are drawing they can look across at the ruler and see where they are, but this means they must take their eyes off the drawing as the ruler is outside the area of visual focus.

When precisely drawing or positioning objects it is common to see users with their noses up close to the screen concentrating hard so that they can position an object exactly where they want it. In this situation they cannot easily look at the coordinates or the ruler – they have to take their eyes off the drawing area and the task they are performing to get the information they need (this is a similar problem to the tool palettes above – the interface is intruding into the task the user is trying to perform).

If users must look at the ruler then they must take their eyes off the objects being positioned. By the time users have moved their eyes from the object to the ruler, or back from the ruler to the object, they may inadvertently move their hand slightly on the mouse and so move the object being positioned. They may not notice this movement (and so position the object incorrectly) or may then have to look back to the ruler again to check if the position is correct (with the potential of the same error happening again).

For example, suppose you wanted to position a label five pixels to the left of the top of a rectangle. You would move to the approximate location and then have to look at the ruler or coordinates to check if you were in right location. You would then move and check until the position was correct. When doing this your hand may move slightly so moving the label. This is particularly important when doing very fine positioning tasks.

One way to solve the problem is to make the position information easier to get at. The coordinates could be presented graphically at the mouse location so that users would not have to move their visual attention away from their drawing task, but then the coordinates would obscure the drawing underneath (in a similar way to the tool palette described above). An alternative solution is to use sound. This would not require users to take their eyes off their drawings, so making it less likely that they would accidentally move the mouse when positioning items (also the drawing will not be obscured by extra graphics).

### *5.1 Earcons for drawing*

Earcons were needed to indicate the coordinate location. An earcon was played if the mouse was moved one pixel with the mouse button down. This meant that if a user wanted to move five pixels he/she could listen to five sounds, making the task simpler as he/she would not have to look away from the drawing at the ruler or coordinates.

In the example just described, if the user moved to the approximate location, he/she could look once at the coordinates to find out the current location and then move to the correct location by listening to the sounds. If he/she had moved to within five pixels of the correct location of the label he/she could move the five by listening to the sounds without needing to look away from the drawing task. The chance of moving the mouse inadvertently would be removed as the user would be looking at the object being positioned all of the time.

The hierarchy of earcons used is shown in Figure 4. The timbre of the earcon played depended on which tool was being used. This also acted to reinforce the current tool in the users mind, helping with the problems described for the tool palette. For drawing horizontally the earcons were played at pitch $C_2$ and for vertically at $C_3$, the timbre was based on the current tool.

**Drawing**
Default tool: Marimba
Other tools : Trumpet
Stereo position defined by the application

**Move 1 pixel**
Vertical: Pitch $C_3$ base note
Horizontal: Pitch $C_2$ base note
Single tone, duration 100 msec.

**Move 10 pixels**
Pitch E above base note

**Move 50 pixels**
Pitch G above base note

**Figure 4:** Hierarchy of earcons used for drawing.

It was decided that the sounds could be made to emulate a ruler more closely by providing different earcons to represent 1, 10 and 50 pixels. On a ruler it is common to see the units represent by small tick marks, the tens represented by longer marks and the fifties by the longest marks. The same was done with earcons.

When users moved 1 pixel the 1 pixel earcon was played. When the mouse was moved over a 10 pixel boundary (as would be shown by a longer tick mark on a graphical ruler) the 10 pixel earcon was played in addition (see Figure 4). This made a chord of C and E (the pitch was defined by whether the movement was horizontal or vertical, the timbre was defined by the current tool). When the user moved over a 50 pixel boundary the 50 pixel earcon was played. This added another note (G) into the chord. By doing this users could drag objects around and get much of the information in sound that was given visually by the ruler without having to take their eyes off their drawing.

However, if a sound was heard for every pixel all of the time then it could become annoying. To avoid this problem the speed of movement of the mouse was used to determine when earcons should be played. If the user was moving fast (X>90 pixels/sec.) then only the 50 pixel earcon was played. If the user was moving more slowly (15 pixels/sec.<X<90 pixels/sec.) then the 10 pixel earcon was played. Finally, if the user was moving very slowly (X<15 pixels/sec.) then the one pixel earcon was played (as described in the previous paragraph). Slow movements usually mean that fine positioning tasks are being undertaken whereas fast movements mean positioning is more general. When making fast movements users do not need to know about each pixel they move over. The values for the speeds of movement were defined by some simple trials with users.

## 6 Experiment

An experiment was needed to investigate if the addition of sound to the graphics package would increase the usability. Due to time constraints it was only possible to test the tool palette to investigate if sound could solve the problems of tool mis–selection. Twelve participants were used. They were undergraduate students from the University of Glasgow with three years of experience with graphical interfaces and tool palettes. Participants had to be familiar with graphics packages so that they could concentrate on the drawing tasks they were given and use the tool palette as they would in their everyday interactions.

### *6.1 Hypotheses*

The extra auditory feedback heard by participants should make the task easier because they will be able to tell they have made errors and recover from them more readily. This should result in an overall reduction in subjective workload.

There should be no increase in annoyance due to the sounds as they will be providing information that the participants need to overcome usability problems.

The number of tasks performed with the wrong tool should be reduced when sound is present as users will know which tool is currently active; sonic presentation of the current tool is the most effective method. This will be indicated by a decrease in the number of tasks performed with the wrong tool.

### *6.2 Task*

Participants were required to perform drawing tasks set by the experimenter in a simple graphics package. Figure 1 shows a screen–shot of the graphics package, the tool palette used is shown in Figure 3. The package was a standard one with standard tools. It was based on ArtClass, a demonstration program supplied with Symantec Think Pascal for the Macintosh. The hybrid method for tool selections was added to the package as this has the potential for the most usability problems. However, the results would also show how errors with the other types of palettes could be solved.

In the standard ArtClass package different tools were indicated graphically by a change in cursor shape. Some of the cursor shapes are shown in Figure 5 (these are similar to the ones used in the Microsoft Word drawing editor and are standard shapes used on the Macintosh). These were left in for the experiment as many packages do use cursor shape to indicate the tools. The hypothesis here is that these are not salient enough to indicate the current tool to the user.

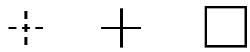

**Figure 5**: Examples of the cursor shapes used in the graphics package. The first is for the selection tool, the second for the line/rectangle tools and the final one for the eraser.

The drawing tasks performed involved users drawing a simple car, tree and sun. The eight car–drawing tasks were described step–by–step, the final two tasks were left more open (so that the users could do them as they liked). The tasks were designed to mimic the standard drawing tasks a user might perform. The tasks also gave participants the opportunity of double and single clicking the tools in the palette.

### *6.3 Experimental design and procedure*

The experiment was a two–condition, within–groups design. The order of presentation was counterbalanced to avoid any learning effects. One group of six participants performed the auditory tool palette condition first and the other used the standard visual palette first (see Table 1). Ten minutes of training was given before each condition to enable the participants to become familiar with the system, the sounds and the types of drawing tasks they would be required to perform. Each condition lasted approximately 20-25 minutes. During each condition the participants had to perform the standard drawing tasks set by the experimenter. Instructions were read from a prepared script.

To get a full measurement of usability combined measures of error rates and subjective workload were used. Such qualitative and quantitative tests give a good measure of usability (Bevan & Macleod, 1994). The standard six–factor NASA Task Load Index (TLX) was used for estimating workload (Hart & Staveland, 1988). To this was added a seventh factor: annoyance. One of the concerns of potential users of auditory interfaces is annoyance due to sound pollution. In the experiment described here the annoyance was measured to find out if it was indeed a problem. It is important when evaluating systems that use sound to test a full range of subjective factors, including annoyance. The area is still in its infancy and it is not yet clear how to design sonically–enhanced interfaces that work well. Just evaluating performance time or errors may not be enough to bring up all of the potential problems. Participants were also asked to indicate overall preference, i.e. which of the two interfaces they felt made the task easiest.

<table>
<tr><th>Participants</th><th>Condition 1</th><th></th><th>Condition 2</th><th></th></tr>
<tr><td>Six Participants Æ</td><td>Sonically-Enhanced Palette Train & Test</td><td rowspan="2">Workload Test</td><td>Visual Palette Train & Test</td><td rowspan="2">Workload Test</td></tr>
<tr><td>Six Participants Æ</td><td>Visual Palette Train & Test</td><td>Sonically-Enhanced Palette Train & Test</td></tr>
</table>

**Table 1:** Format of the Experiment.

## 7 Workload results

The average TLX workload scores for each category were calculated and are shown in Figure 6. They were scored in the range 0–20. The average raw workload was 9.29 in the auditory condition and 9.61 in the visual. There was no significant difference in the workload between conditions ($T_{11}$=0.26, p=0.79).

There was no significant difference in terms of annoyance between the conditions ($T_{11}$=0.24, p=0.81). Six of the participants felt the visual condition was more annoying, five felt the auditory more annoying and one felt them equal.

The average scores for overall preference was 13.67 for the auditory condition and 10.67 for the visual condition. Again, this was not significantly different ($T_{11}$=1.70, p=0.12). Nine participants preferred the tool palette with sounds, and three participants preferred it without.

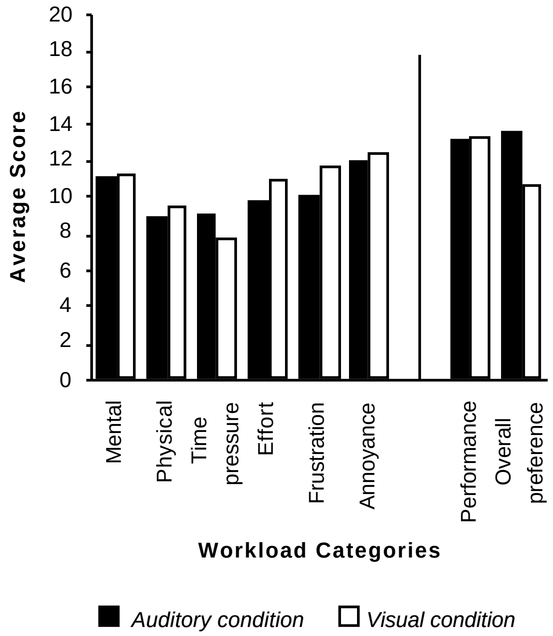


**Figure 6:** Average workload scores for the two conditions. In the first six categories higher scores mean higher workload. The final two categories are separated as higher scores mean lower workload.

## 8 Error results

Figure 7 shows the number of tasks performed with the wrong tool (i.e. the participant used the wrong tool and then had to change to the correct one. This could happen either because an automatic tool change occurred and was not noticed or the tool did not change). There was a significant reduction in the number of such tasks in the auditory condition ($T_{11}$=3.08, p=0.01). The average number of tasks performed with the wrong tool fell from 3.25 in the visual condition to 0.83 in the auditory. This indicated that the earcons did help participants remember the tool they were using. In total, eight participants never used a wrong tool in the auditory condition with only three not making such errors in the visual.

### *8.1 Discussion*

The workload analysis showed no significant differences in terms of workload. Even though the error results showed that sound reduced the number of times the wrong tools were used, workload was not reduced (as had been hypothesised). One explanation for this is that users were asked to rate the workload of the task as a whole rather than for the tool palette specifically. They may have considered the selection of tools a small part of the whole task so any differences in the workload attributable to tool selection were lost amongst the data for the whole task.

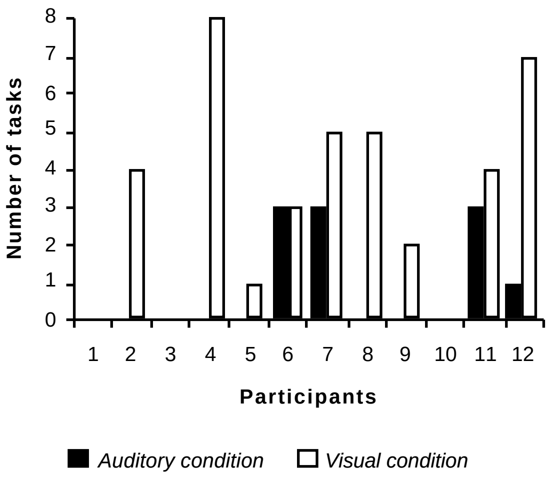


**Figure 7:** Number of tasks performed with the wrong tool.

The results show no difference in the annoyance experienced by users. This indicates that if care is taken in the design of the earcons, and they solve specific usability problems, users will find them useful and not annoying.

Analysis of the errors made by participants showed that there were significantly fewer tasks performed with the wrong tool in the auditory condition (as hypothesised). This meant that the earcons were successfully indicating to the participants what tool, or mode, they were in. In fact only four of the twelve participants tried to perform any of the tasks with the wrong tools in the auditory condition. When tool errors did occur there were two possibilities: The user could try to draw another object but end up using the selection tool (as a tool change occurred and was not noticed) or the user could try and position an object but end up drawing another (as the tool did not change). Analysis showed that 86% of all of the errors that occurred in both conditions were of the former type. Even though the earcons were effective at reducing the number of errors, further development of this work should concentrate on making such tool changes as salient as possible to users.

The tool palette evaluated in the experiment was the hybrid type. However, the sounds needed for the other types mentioned above would be very similar. For the type in which the system changed back to the default tool after every use the changes must be made salient to the user. This could be done, as here, by playing a sound that indicated a change had occurred.

For the type where the current tool stays selected until the user changes it, the system must make it clear to the user that the tool has not changed. This could be done, as in this experiment, by playing the same earcon again.

## 9 Conclusions

Results from the experiment described here showed that sonic–enhancement of a tool palette could significantly increase the usability, and therefore productivity, of a graphics package without making it more annoying to use. The package used here was very simple, with only one tool palette. In more complex applications (for example, large CAD tools) there may be many tool palettes conveying many different types of information to the user. Users are likely to miss information displayed in palettes, and the more palettes there are the worse this problem will be. The results given in this paper show that with the simple enhancement of the existing interface many of these problems can be removed. Designers of graphics packages can use this work to enhance their products and allow their users to increase their productivity.

## Acknowledgements

Thanks to Catherine Clarke who helped in this research as part of her undergraduate project. Thanks also to Michel Beaudoin–Lafon for discussions on this work. Part of this work was supported by EPSRC Grant GR/L79212.

## References

References by Brewster are available from http://www.dcs.gla.ac.uk /~stephen/

Alty, J.L. (1995). Can we use music in human-computer interaction? In Kirby, Dix & Finlay (Eds.), *Proceedings of HCI'95*, (pp. 409-423), Huddersfield, UK: Cambridge University Press.

Barfield, W., Rosenberg, C. & Levasseur, G. (1991). The use of icons, earcons and commands in the design of an online hierarchical menu. *IEEE Transactions on Professional Communication*, **34**, 101-108.

Beaudouin-Lafon, M. & Conversy, S. (1996). Auditory illusions for audio feedback. In Tauber (Ed.), *ACM CHI'96 Conference Companion*, (pp. 299-300), Vancouver, Canada: ACM Press, Addison-Wesley.

Berglund, B., Preis, A. & Rankin, K. (1990). Relationship between loudness and annoyance for ten community sounds. *Environment International*, **16**, 523-531.

Bevan, N. & Macleod, M. (1994). Usability measurement in context. *International Journal of Man-Machine Studies*, **13**, 123-145.

Blattner, M., Sumikawa, D. & Greenberg, R. (1989). Earcons and icons: Their structure and common design principles. *Human Computer Interaction*, **4**, 11-44.

Brewster, S.A. (1997). Using Non-Speech Sound to Overcome Information Overload. *Displays*, **17**, 179-189.

Brewster, S.A. & Crease, M.G. (1997). Making Menus Musical. In Howard, Hammond & Lindgaard (Eds.), *Proceedings of IFIP Interact'97*, (pp. 389-396), Sydney, Australia: Chapman & Hall.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1993). An evaluation of earcons for use in auditory human-computer interfaces. In Ashlund, Mullet, Henderson, Hollnagel & White (Eds.), *Proceedings of ACM/IFIP INTERCHI'93*, (pp. 222-227), Amsterdam, Holland: ACM Press, Addison-Wesley.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1994). The design and evaluation of an auditory-enhanced scrollbar. In Adelson, Dumais & Olson (Eds.), *Proceedings of ACM CHI'94*, (pp. 173-179), Boston, MA: ACM Press, Addison-Wesley.

Brewster, S.A., Wright, P.C. & Edwards, A.D.N. (1995). Experimentally derived guidelines for the creation of earcons. In Kirby, Dix & Finlay (Eds.), *Adjunct Proceedings of BCS HCI'95*, (pp. 155-159), Huddersfield, UK

Dix, A., Finlay, J., Abowd, G. & Beale, R. (1993). *Human-Computer Interaction*. London: Prentice-Hall.

Gaver, W. (1986). Auditory Icons: Using sound in computer interfaces. *Human Computer Interaction*, **2**, 167-177.

Gaver, W. (1989). The SonicFinder: An interface that uses auditory icons. *Human Computer Interaction*, **4**, 67-94.

Hart, S. & Staveland, L. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research., In Hancock & Meshkati, Eds., *Human mental workload*. Amsterdam: North Holland B.V.

Kramer, G. (1994). An introduction to auditory display, In Kramer, Ed., *Auditory Display.* Reading, MA: Addison-Wesley.

Monk, A. (1986). Mode Errors: A user-centered analysis and some preventative measures using keying-contingent sound. *International Journal of Man-Machine Studies*, **24**, 313-327.

Portigal, S. (1994) *Auralization of document structure*. MSc. Thesis, The University of Guelph, Canada.

Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.

Rigas, D.I. & Alty, J.L. (1997). The use of music in a graphical interface for the visually impaired. In Howard, Hammond & Lindgaard (Eds.), *Proceedings of IFIP Interact'97*, (pp. 228-235), Sydney, Australia: Chapman & Hall.

Sellen, A., Kurtenbach, G. & Buxton, W. (1992). The prevention of mode errors through sensory feedback. *Human Computer Interaction*, **7**, 141-164.

Slowiaczek, L.M. & Nusbaum, H.C. (1985). Effects of speech rate and pitch contour on the perception of synthetic speech. *Human Factors*, **27**, 701-712.