Keeping an eye on the game: eye gaze interaction with Massively Multiplayer Online Games and virtual communities for motor impaired users

S Vickers[1], H O Istance[1], A Hyrskykari[2], N Ali[2] and R Bates[1]

[1]Human-Computer Interaction Research Group, De Montfort University, The Gateway, Leicester, UK

[2]Human-Computer Interaction Unit (TAUCHI), Department of Computer Sciences FIN-33014 University of Tampere, FINLAND

svickers@dmu.ac.uk, *hoi@dmu.ac.uk*, *ah@cs.uta.fi*, *nazmieali@gmail.com*, *rbates@dmu.ac.uk*

ABSTRACT

Online virtual communities are becoming increasingly popular both within the able-bodied and disabled user communities. These games assume the use of keyboard and mouse as standard input devices, which in some cases is not appropriate for users with a disability. This paper explores gaze-based interaction methods and highlights the problems associated with gaze control of online virtual worlds. The paper then presents a novel 'Snap Clutch' software tool that addresses these problems and enables gaze control. The tool is tested with an experiment showing that effective gaze control is possible although task times are longer. Errors caused by gaze control are identified and potential methods for reducing these are discussed. Finally, the paper demonstrates that gaze driven locomotion can potentially achieve parity with mouse and keyboard driven locomotion, and shows that gaze is a viable modality for game based locomotion both for able-bodied and disabled users alike.

1. INTRODUCTION

In the last few years, the popularity of on-line games and virtual communities has grown enormously. The Massively Multiplayer Online Social Game (MMOSG) of Second Life, for example, has over 14 million members (Linden, 2008) and can support interest groups, cooperative activities between people as well as serious commercial activities. On the other hand, the Massively Multiplayer Online Role Playing Game (MMORPG) of World of Warcraft is more focused on character development achieved through completing game related goals. World of Warcraft has more than 10 million users worldwide (Blizzard, 2008). Profoundly disabled people can find great satisfaction and enjoyment from participation in such virtual communities (Stein, 2007) and can choose to reveal as much, or as little of their disability as they choose visually, by creating their own avatar.

Users with motor impairments often retain good control of their eye muscles when fine motor control of other muscle groups is lost. Eye movement can be used very effectively for interacting with computers, although most of the existing work on gaze based interaction for disabled users focuses on 2D desktop applications and text entry (Lankford, 2000; Majaranta & Raiha, 2002; Hornof & Cavender, 2005). However a number of general problems exist with gaze-based interaction and the migration to the control of 3D graphical worlds. If gaze is the only modality being used, then mouse clicks are often emulated by a slight stare, or dwell, at a location on the screen. The 'Midas Touch' problem arises when unintentional clicks are generated by the user looking naturally at the same place on the screen (Jacob, 1993). The most common solution to this is to use deliberately long dwell times. However, these can be fatiguing and can result in the gaze point moving off the intended target before the end of the dwell period. Another problem arises where the point of input and the point of feedback are spatially disparate, and the user has to look repeatedly between the two (Istance et al, 1996).

User interaction with these virtual worlds needs to accommodate many more tasks than before, such as navigation in 3D space. It also faces the added challenge of requiring fast real-time interaction if participation by disabled users in a world or game is to be on an equal footing as other able-bodied participants (Bates et al, 2008). The usual type of 'dwell-to-click' interaction is too limited for the extended range of tasks where a

variety of interaction techniques using mouse and keyboard are used in quick succession by able-bodied users.

2. EYE CONTROL IN MMOGS

2.1 Interaction

The typical method of interaction in MMOG environments is via a combination of mouse and keyboard; with avatar movement normally being performed using cursor keys (or the W, A, S, D cluster of keys). However, the mouse is often an option for movement control, both in point-to-navigate games and also those games that offer a movement interface. This is the case in Second Life; by clicking on directional arrows located on semi-transparent movement panels. Camera movement can also be performed by using a mouse with a similar panel. In fact, tasks in all categories can be performed using a mouse, with object manipulation and application control being performed using drop-down menus, transparent pie-menus (figure 1) and dialog boxes.

Figure 1. *Pie menu is Second Life that allows different actions to be performed on an object (Istance et al, 2008a).*

2.2 Problems with Basic Input Device Emulation

One approach to using eye gaze as an input device is by straight forward emulation. Mouse emulation can be implemented with the system cursor following the user's point-of-gaze on the screen. To perform a mouse click then either a secondary input device is used or an eye gaze based selection methods is used, such as dwell time. In addition to mouse emulation there are possibilities to emulate other input devices. There are several ways to perform joystick emulation such as, using an onscreen joystick with gaze friendly command buttons or by the cursor moving incrementally based on the user's point-of-gaze. A keyboard can be emulated by using an onscreen keyboard or by more novel methods such as those used by Dasher in which, text is entered by 'flying' into predicted text. Typical emulation problems include:

- Functionality – Gaze based emulation needs to represent all of the possible functions and have ways of switching between them with a minimal cognitive overhead. E.g. a mouse usually has at least two buttons and often a central wheel that is used for scrolling. In addition, multiple interaction techniques can be used on each such as click, double click and drag.
- Interface design – Most interfaces have not been considered for use with eye gaze and have often been designed to make full use of mouse interaction methods and the functions that they represent.
- Accuracy – Difficulties arise from eye tracker accuracy and these become apparent when pointing at small targets, such as those found in 2D interfaces. Selection of such targets can also be difficult due to 'cursor chasing', occurring when the cursor position and the point-of-gaze are offset due to the inaccuracy and also eye drift occurring during dwell selection.
- Freedom – Normal mouse use allows the user to look at one part of the screen whilst pointing the cursor at another or even removing the hand from the mouse altogether. The same applies to keyboard and joystick input.
- Latency – There is response latency caused by the detection of eye movement, its interpretation, the mapping to a system event such as a key press and finally any latency in the application responding to the event.

No matter what device is being emulated most applications have no knowledge of the input device being used and that the cursor movement and button events originate from an eye tracker.

Some of these problems found can be traced back to the Midas Touch paradigm that arises due to the eyes being an always on device; the eye is a perceptual organ designed for looking at objects rather than controlling them (Jacob, 1993). Due to the inherent inaccuracies of the eye tracking system it is common to move the system cursor over objects that are close to the desired object. The Midas Touch turns this inaccuracy into a selection error if a control action is applied.

Previously, we have investigated the feasibility of gaze based mouse emulation for interaction with online virtual environments (Bates et al, 2008a; Vickers et al, 2008). We found that due to the real-time requirements for interaction in such environments there was a need to switch rapidly between mouse emulation modes. There also needed to be a method of disengaging gaze control so as to observe the environment without fear of Midas Touch selection. Additionally, a similar method was required so as to avoid problems with distraction, e.g. during locomotion style tasks the participant "walks where they are looking", meaning that they are unable to observe the world in the periphery whilst walking.

2.3 Finding Solutions

Most gaze control systems have a facility to pause or suspend active control. However, this is often not used as the cognitive overhead in doing so is prohibitive. One element in a solution therefore is to incorporate a very lightweight process to turn gaze control off. Another is to support several modes of mouse, keyboard or joystick action simultaneously and to have a similarly light weight way of switching between these. One such method is by using glances and detecting when a glance or "flick" of the eye off the edge of the screen occurs. Each directional glance could represent a different mode of operation.

The use of modes in the user interface is not a new concept and was once considered bad practice due to the overhead of remembering which mode was currently in operation (Nievergelt & Weydert, 1987). There are also potential difficulties in the user remembering where each mode is located and how to switch between them. Therefore, for moded operation to be successful, there are a number of design requirements that should be met such as:

- Movement and actions needed to accomplish the task should be efficient.
- Low effort associated with using the mode and with changing between the modes.
- Easy to remember how the modes work and how to activate them.
- Clear feedback about the mode the user is currently in.

2.4 Snap Clutch: a Flexible Gaze Based Input Event Emulator

Snap Clutch (Istance et al, 2008a) is a gaze-based mouse emulator which has 4 modes of operation and uses gaze gestures in the form of off-screen glances to switch between modes, see figure 2. Feedback about the current mode is given by the shape and colour of the cursor.

The modes in our initial implementation of Snap Clutch were based on using *interaction technique* modes. In addition to left click and left drag, we also added a "turn gaze off" and a "park the cursor here mode". Initial trials showed that having the ability to rapidly switch between different mouse emulation modes, had the potential to offer a more real-time interaction experience than normal mouse emulation. Additionally, it offered one approach to resolving problems that can occur due to the Midas Touch.

Although these Snap Clutch modes offered an improvement in task time over normal mouse emulation, when compared to using a hand mouse times were still significantly longer.

3. AN EXPERIMENT

3.1 Task Based Modes

In order to improve on task performance the idea of *task based* modes is introduced, which are to be used together with a selection of the previous *interaction based* modes.

A 'locomotion task' mode provides a means of moving the avatar by introducing active in-screen regions which respond to the gaze position rather than to the system pointer position. This mode causes a stream of keystroke events to be generated which the application recognizes as movement control commands. The user is now free to look around while the avatar moves forward. Glancing to the far left and right regions of the

screen causes the avatar to rotate, see figure 2. Stopping the movement is achieved as before by switching the mode off (glancing off screen).

Figure 2. *Screenshots showing Extended Snap Clutch being used with World of Warcraft. The 'no control' mode with overlay zones for rotating the avatar left and right (left); the 'locomotion' mode with overlay zones for moving and rotating the avatar (right).*

The 'no-control' mode has been enhanced by making the avatar rotate when the user looks at the left or right edges of the visible world. This is now a natural response of the world to the users gaze behavior, see Fig. 2.

New mouse emulation modes have been developed to suit particular interaction techniques. Pie-menus are used extensively in Second Life, therefore, a suitable mode was designed and implemented for their use in which the first dwell generates a right button click and the second dwell generates a left button click.

The interface to the emulator enables interactive selection of the modes to be associated with each edge of the screen. This opens the way to having transparent overlays to enable the user to change the mode associated with each edge of the screen at run-time, see figure 3.

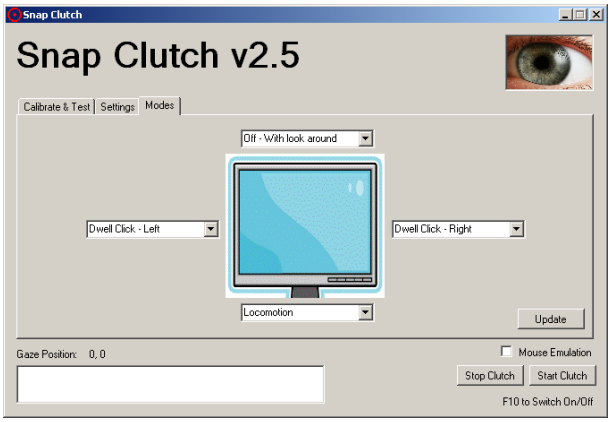

Figure 3. *Screenshot showing the Snap Clutch mode selection screen. The user can choose four modes to suit the application and assign them to an off screen glance of their choosing.*

It was our intention from the outset that these application developments would not be software specific and so far we have been successful. New developments are tested using other MMO style games for compatibility and also for generating new task modes; figure 2 shows Snap Clutch being used with World of Warcraft.

We performed an experiment (Istance et al, 2008b) where the participant used Second Life using two input conditions: keyboard and mouse and gaze interaction using Snap Clutch. We then analysed the error conditions in order to be able to focus further development efforts more efficiently. The four modes used can be chosen according to the actual application. In this experiment we used the following:

- Glance up – Unconstrained looking around
  - No action on dwell.
  - Rotate left when looking inside the left hand edge of the screen.
  - Rotate right when looking inside the right hand edge of the screen.

Proc. 7th ICDVRAT with ArtAbilitation, Maia, Portugal, 2008

- Glance left – Left button click
  - A gaze dwell causes a left button click.
- Glance right – Right button click
  - A gaze dwell causes a right button click.
- Glance down – Locomotion
  - No action on dwell.
  - Streaming of 'W' keystroke events when the user looks in the main part of the screen.
  - Streaming of 'A' and 'D' keystroke events when the user looks into small regions in the left and right hand sides of the screen causing the avatar to rotate left and right whilst walking forward.
  - Streaming of 'S' keystroke events when the user looks inside a thin strip at the bottom of the screen causing the avatar to walk backwards.

3.2 An Approach to User Performance Investigations

The initial pilot study demonstrated that using Second Life with our gaze-based interaction technique resulted in distinctively longer task completion times than when using conventional interaction techniques. In order to achieve parity of gaze interaction with normal keyboard and mouse, it is necessary to be able to identify the usability issues with gaze control. This will allow us to understand what influences the speed of interaction (time of task completion) and the type of errors that are made.

The partitioning of task time into productive time and error time has long been a feature of usability engineering (Glib, 1984). The time spent in a specific error condition represents the potential time to be saved in task completion if the cause of that error can be removed. This relative saving in task completion times by addressing each type of error represents a kind of cost saving benefit of redesigning different features of the user interface.

3.3 Participants and Apparatus

The study involved twelve participants. Ten of them were students and two were university lecturers, who were experienced users of gaze interaction. Ages varies from 20 to 56 with the average being 29. All participants were able bodied. The study was carries out using a Tobii T60 screen integrated eye tracker and all the task executions were recorded using TeamWork screen capture software.

3.4 Tasks

Two sets of three tasks were devised and carried out within a purpose built Second Life environment that represented the university computer science building. These were based on Hand's (Hand, 1997) proposed taxonomy of main control and manipulation areas present within virtual environments. They were as follows:

i) Locomotion – The participant was required to walk from the main entrance, up the staircase and go into a room where there were display panels about individual university modules, see figure 4. The task required the participant to retrieve a module code from a particular panel. Each participant set were required to retrieve a different module code from a different panel.

ii) Object manipulation – The participant was required to change a slide or a web page from within the main lecture theatre and accept the change using a dialog box. One subject set changed the presentation slide by 'touching' a button on a control panel located on the stage. The other participant set caused a web page to be displayed by 'touching' another button that was located near the stage. To achieve this task both sets of subjects were required to perform a right click to display a pie menu and then a left click to select the 'Touch' menu item.

iii) Application control – The participant was required to change the appearance of their avatar. One participant set was to remove the moustache and the other was to raise the height of the eyebrows. To achieve this, the participant was required to use a pie menu as described previously but with the selection of the 'Appearance' menu item. This caused a dialog box to appear and the subject then had to select a group of features to edit from a panel of vertical buttons. A horizontal slider was used to change the selected feature.

3.5 Procedure

The twelve participants were split into two groups of six. One half did one task set with the keyboard and mouse, followed by the other task set using gaze control. The other half started with gaze control followed by

keyboard and mouse. All subjects with the exception of the two experienced participants had not used Second Life before, nor did they have any previous experience of using gaze control. Each participant was given a fifteen minute introduction to Second Life followed by a fifteen minute introduction to using gaze control. This involved a series of simple training exercises.

Figure 4. *Screenshot of a subject performing the locomotion task.*

Each task set began with the avatar in the same place, standing at the entrance to the building. Tasks were completed in the same order for all subjects: locomotion; object manipulation; application control. The task was first explained and then the participant was asked to complete it. If required, the participant was reminded of the task during completion. All participants were asked to complete a brief questionnaire upon completion of the two task sets. They were advised that they could withdraw from the trial at any time. Each session took between 45-55 minutes to complete.

3.6 Analysis and Results

We identified four different categories of errors that the subjects made during the tasks. These were to be annotated upon the videos using an open source video annotation application called Elan. The video data from one subject was marked up by two people so that consistency of the outcomes could be checked. As a result there were minor adjustments made to the error categories and definitions but for the most part there was a high degree of agreement between the two analyses. The four main error categories were as follows:

i) Locomotion error – being one of the following,
- Unintentional motion backwards (the gaze point first moves through the 'move backwards' zone of the screen after glancing down to change into the locomotion mode).
- Unintentional rotation left or right (the subject means to glance off screen to change modes when in 'no control' mode, but rotates the avatar instead).
- Turn overshoot (the subject deliberately turns while in 'locomotion' mode but turns too far and has to correct).
- Walk overshoot (the subject tries to stop, but the change to 'no control' mode takes too long and the avatar walks to far and subsequently has to reverse).

ii) Mode change error – an unintentional change of the mode; a subject tries to rotate left or right, in 'no control' or 'locomotion' mode but changes mode by mistake by looking too far off screen.

iii) Accuracy error – a subject tries to select a target but misses due to inaccurate pointing.

iv) Misunderstanding error – a subject misunderstands / mishears / forgets what to do.

For each subject and for each task the total time spent in each error condition was subtracted from the total time, leaving us the non-error time for each trial. The outcome of the trials for the three tasks is as shown in figure 5 and table 1. Table 1 shows the tasks along with the average error-free time; the total error count; the total error time; the percentage of error time. This is also represented in figure 5 but with the error-time divided into their associated error categories.

One of the subjects had difficulty in calibrating the eye tracker. She was able to complete all of the tasks in the gaze condition but the number of accuracy errors was far greater than the other subjects with more than 3 standard deviations from the mean. Consequently all data from this subject was removed from the analysis.

Table 1. *Average task times, error counts, error time and error percentage for all tasks (kbm = keyboard and mouse)*

Task	Condition	Error free Time (s)	Error Count	Error Time (s)	Err %
Locomotion	*Gaze*	57.97	4.55	21.4	22.68
	KBM	47.13	0.18	1.32	2.54
Application	*Gaze*	79.25	3.09	41.75	29.91
	KBM	40.15	0.45	2.73	5.95
Object	*Gaze*	18.99	2.45	21.89	50.05
	KBM	7.84	0.18	1.05	6.1

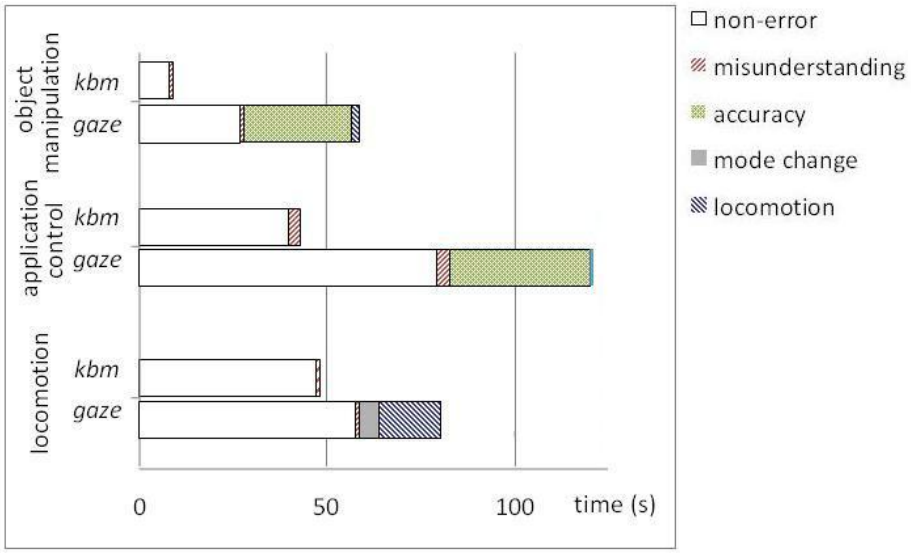

Figure 5. *Average task completion times partitioned into error times (in four types of errors) and non-error times*

The results show that all subjects were able to complete the three tasks using eye gaze. The non-error time enables comparison of task times if the cause of the errors can be removed through design changes. The non-error times for gaze is encouraging, especially those for the locomotion task. With only a short training session, subjects would be able to complete the locomotion task using gaze almost as quickly as they would using a keyboard, if the cause of the locomotion errors could be removed. The reasons for the locomotion errors were partly due to the speed of movement by the avatar in response to the key commands generated by Snap Clutch. This caused overshooting and undershooting of avatar movement that would need to be corrected. The processing pipeline in using a single computer is a significant contribution to this: eye tracker – emulator (Snap Clutch) – Second Life (additionally, in the experimental condition, the video capture software). There are also possible optimisations that can be made to improve the emulator software. Another source of locomotion error is the location of the backwards motion overlay zone at the bottom of the screen, see figure 3. The gaze position has to travel through this zone after changing into the locomotion mode and combined with the latency within the system an unwanted backwards movement results. These issues can be addressed within the implementation of the locomotion mode in addition to examining causes for response latency.

The biggest cause of errors in the application control and object manipulation tasks is the difficulty in hitting the small control objects within the dialog boxes. This was exacerbated by latency in generating click events probably due to the processing pipeline. One solution to reduce these types of errors is to incorporate some form of zoom interface that is common within 2D gaze driven interfaces.

4. CONCLUSIONS

In this paper, we have discussed some of the problems associated with gaze based mouse emulation and their effectiveness for use in online virtual environments such as Second Life. Our initial studies highlighted key

problem errors which, we were able to partly address in the implementation of emulator software. To improve further we moved from using *interaction technique* modes to *task based* modes.

The study has been successful in identifying the extent and causes of the difference in performance between the gaze and keyboard-mouse conditions. This has revealed specific design changes that address these differences and also gives an indication of the likely performance improvements. Importantly however, it has demonstrated the feasibility and potential for gaze based interaction with online virtual environments. In particular that of gaze based locomotion, in which there lies real possibility that eye gaze can achieve parity with that of mouse and keyboard when performing such tasks.

Acknowledgements: This work is supported by: Communication by Gaze Interaction (COGAIN) FP6 Network of Excellence, the Institute of Creative Technologies (IOCT) at De Montfort University and the Royal Academy of Engineering, London.

5. REFERENCES

R Bates, H O Istance and S Vickers (2008), Gaze Interaction with Virtual On-Line Communities: Levelling the Playing Field for Disabled Users, *Proceedings of the 4th Cambridge Workshop on Universal Access and Assistive Technology; CWUAAT 2008*.

Blizzard (2008, January), *World of Warcraft Reaches New Milestone: 10 Million Subscribers*, Retrieved June 2008, from Blizzard Entertainment: *http://www.blizzard.com/us/press/080122.html*

T Glib (1984), The "impact analysis table" applied to human factors design, *First IFIP Conference on Human-Computer Interaction*; *INTERACT 1984.*

C Hand (1997), A Survey of 3D Interaction Techniques, *Computer Graphics Forum , 16* (5), 269-282.

A Hornof, A Cavender and R Hoselton (2004), EyeDraw: A System for Drawing Pictures with Eye Movements, *Conference on human factors in computing systems; CHI 2005.*

H O Istance, C Spinner and P A Howarth (1996), Providing motor-impaired users with access to standard graphical user interface (GUI) software via eye-based interaction, *Proceedings of the 1st International Conference on Disability, Virtual Reality and Associated Technologies; ICDVRAT 1996.*

H O Istance, R Bates, A Hyrskykari and S Vickers (2008), Snap Clutch, a Moded Approach to Solving the Midas Touch Problem, *Eye Tracking Research & Applications; ETRA 2008.*

H O Istance, A Hyrskykari, S Vickers and N Ali (2008), User Performance of Gaze-based Interaction with On-line Virtual Communities, *Proceedings of the 4th Conference on Communication by Gaze Interaction; COGAIN 2008*

R Jacob (1993), Eye-movement-based human-computer interaction techniques: Toward non-command interfaces, In *Advances in Human-Computer Interaction* (Vol. 4, pp. 151-190), Ablex Publishing Corporation.

C Lankford (2000), Effective eye-gaze input into Windows, *Eye Tracking Research & Applications; ETRA 2000*.

Linden (2008, July), *Second Life | Economic Statistics*, Retrieved July 2008, from Second Life: *http://secondlife.com/whatis/economy_stats.php*

P Majaranta and K J Raiha (2002), Twenty Years of Eye Typing: Systems and Design Issues, *Eye Tracking Research & Applications; ETRA 2002.*

J Nievergelt and J Weydert (1987), Sites, modes, and trails: Telling the user of an interactive system where he is, what he can do, and how to get to places (excerpt), In *Human-computer interaction: a multidisciplinary approach* (pp. 438-441), San Francisco: Morgan Kaufmann Publishers Inc.

R Stein (2007, October), *Real Hope in a Virtual World: Online Identities Leave Limitations Behind*, Retrieved June 2008, from Washington Post: *http://www.washingtonpost.com/wp-dyn/content/story/2007/10/05/ST2007100502446.html?hpid=topnews*

S Vickers, R Bates and H O Istance (2008), Gazing into a Second Life: Gaze-Driven Adventures, Control Barriers, and the Need for Disability Privacy in an Online Virtual World, *Proceedings of the 7th International Conference on Disability, Virtual Reality and Associated Technologies (ICDVRAT), Sept. 8-11, 2008, Maia, Portugal.*