

Text Entry Using a Dual Joystick Game Controller

Andrew D. Wilson¹

¹Microsoft Research
One Microsoft Way
Redmond, WA 98052
awilson@microsoft.com

Maneesh Agrawala^{1,2}

²University of California, Berkeley
615 Soda Hall #1776
Berkeley, CA 94720-1776
maneesh@cs.berkeley.edu

ABSTRACT

We present a new bimanual text entry technique designed for today's dual-joystick game controllers. The left and right joysticks are used to independently select characters from the corresponding (left/right) half of an on-screen selection keyboard. Our dual-stick approach is analogous to typing on a standard keyboard, where each hand (left/right) presses keys on the corresponding side of the keyboard. We conducted a user study showing that our technique supports keyboarding skills transfer and is thereby readily learnable. Our technique increases entry speed significantly compared to the status quo single stick selection keyboard technique.

Author Keywords

Text entry, dual joystick, dual control pad, game controller, game console

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Input devices and strategies.

INTRODUCTION

Today's video game consoles are powerful computers that provide access to all forms of electronic media, including music, movies and, of course, video games. Already these consoles connect to the Internet through services like Xbox Live, and software for next generation consoles is likely to enable digital video recording, photo browsing, web surfing, instant messaging (IM) and email. An effective text entry technique would greatly enhance all of these applications and it is a fundamental requirement for extended use of IM and email. Even video games, especially networked games, often require text entry for registration, character setup and in-game chat.

But a significant limitation of current and next generation video game consoles is the lack of an effective mechanism for entering text. Keyboards, if available, cost extra and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22-28, 2006, Montréal, Québec, Canada.
Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

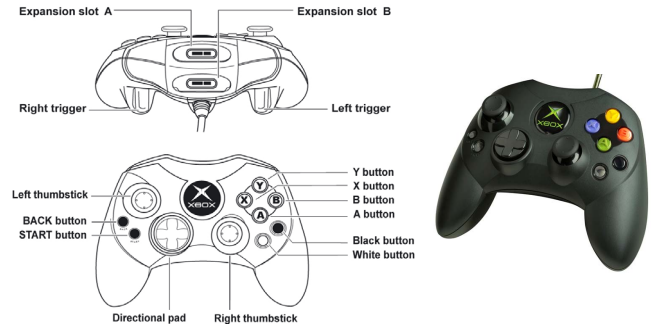


Figure 1. The Xbox controller used in the present work is typical of today's console game controllers.

therefore many users must enter text through a game controller. The most common approach allows users to select characters from an onscreen selection keyboard using a single joystick. The advantage of this approach is that first-time users can immediately enter text without having to learn a complicated input procedure. However, entering lots of text this way can be very slow and tedious [13]. While alternative key layouts [5, 9] might reduce selection time in the long run, users initially have to spend time visually searching for and learning the location of each character.

In this paper, we propose a different approach for improving the speed of selection keyboards. Instead of selecting characters with a single joystick, users drive our system using the two joysticks commonly found on a game controller (see Figure 1). We maintain the QWERTY layout of a standard keyboard to promote skills transfer between typing and entering text with our system. Because our approach is bimanual, it is a closer analog to typing on a keyboard than selecting characters with a single joystick. A key feature of our system is that it requires almost no learning and is therefore more likely to be picked up by casual users. User studies show that, even with first-time users, our technique increases entry speed by 22% over the single joystick selection keyboard.

RELATED WORK

Text entry is a very old interface problem and as a result there is a vast body of work on the topic. Here, we briefly touch on earlier work concerning joystick-based text entry.

Gesture-based text entry techniques allow users to enter characters by drawing gestures. However, accurately drawing curved gestures with a joystick is very difficult. Therefore, these techniques often limit the gesture set to contain

only straight line segments [2, 7]. Wobbrock et al.'s Joy-Stick EdgeWrite [13] is particularly interesting because it offers straight line gestures that resemble Roman characters. As a result it is easier to learn the mapping between the EdgeWrite gestures and characters. Nevertheless, EdgeWrite requires some practice to learn the gestures. All of these techniques are designed for single joystick input.

Radial marking menus have also been used for joystick based text entry [1, 11]. Weegie [3] is notable because it is a dual-stick technique with a separate marking menu available through each stick. XNav [4] is a zone-based technique that builds on Perlin's Quikwriting system [10]. A major drawback of all of these techniques is that users must learn the location of characters within the menu hierarchy. Transfer of typing skills to joystick text entry is impossible with these techniques.

DUAL JOYSTICK TEXT ENTRY

Our dual-stick text entry method uses an onscreen keyboard that is split into two halves (see Figure 2). Each half separately maintains a currently selected key, indicated graphically by highlighting. Moving the left or right joystick on the gamepad moves the corresponding selection. Both the left and right selections may be changed simultaneously. To enter the character users must hit the corresponding left or right trigger on the controller.

We anticipate that this two stick selection keyboard will improve typing speed due to the reduction in distance (work) required to reach a given key, compared to a similar single stick approach.

The most frequently used character in many typing tasks is 'space'. In our system hitting both triggers simultaneously enters a 'space'. We chose this approach rather than including a separate spacebar key in the keyboard because it facilitates quick entry. To enter the second most frequent character 'backspace', users must move both sticks to the left (as far left as possible, to distinguish from simultaneously moving both selection highlights left). A 'sticky' shift mode is entered by pushing either stick down into the case of the controller. Shifted mode is automatically disabled upon selection of a key. Finally, the top row of the usual QWERTY keyboard, including numbers and many symbols, as well as some symbols missing from the layout in Figure 2, can be reached by switching to a new key layout (not shown) by simultaneously moving both sticks up.

Our technique takes advantage of the typical user's familiarity with typing on a QWERTY keyboard. Users do not have to visually search for a desired key because they know the keyboard layout. Moreover, users who type on a QWERTY keyboard regularly build motor memory of the keyboard layout within each hand. The left hand strikes keys on the left half of the keyboard while the right hand strikes keys on the right half. Our two-stick selection keyboard is designed to exploit such motor memory as the left

esc	q	w	e	r	t	y	u	i	o	p	\
tab	a	s	d	f	g	h	j	k	l	;	'
ctrl	z	x	c	v	b	n	m	,	.	/	alt

Figure 2. The dual stick/QWERTY onscreen keyboard. The left and right highlighted characters indicate the character to be entered upon hitting the left and right triggers on the controller, respectively.

and right hands are responsible for the corresponding halves of the keyboard.

The onscreen keyboard can either remain fixed in one part of the screen or move continuously so that it appears just below the current insertion point. We included the continuously moving option to reduce the divided attention problem that can occur when users must look at two different locations that are far away from one another.

LABORATORY USER STUDY

To evaluate the performance of our text entry technique, we conducted a user study in which subjects were asked to type text phrases as quickly as they could using our technique. Besides demonstrating that naïve users are able to quickly learn the technique, we were interested in isolating the effects of two of its salient features: the use of two sticks and QWERTY layout. In addition to the dual stick/QWERTY (DQ) prototype discussed above, we had our subjects use variants of our prototype that used a single stick with a QWERTY layout (SQ), single stick with an alphanumeric layout (SA), and two sticks with an alphanumeric layout (DA). Figure 3 illustrates all configurations in the study.

In the single stick configurations, the left stick alone is used to change the single highlighted keystroke before it is selected by pulling the left trigger. To enter a space, the X button is depressed with the right hand, while hitting either trigger. Similarly, to enter a backspace, the X button is depressed while moving the stick to the left. In this way the X button takes on the role of the right trigger of our 2 joystick design. Shift mode is entered as before, by pressing the left stick into the case.

Hypotheses

In our study we wished to test the following hypotheses: H1--Users will obtain higher text entry speeds with QWERTY layout; H2--Users will obtain higher text entry speeds with dual stick configuration; and H3--The greatest text entry speed will be obtained with QWERTY layout and dual stick configuration (condition DQ).

Informally, we also wished to demonstrate that novice subjects with no prior experience entering text through a gamepad could achieve a level of performance comparable with alternative gamepad-based text entry methods during a single experimental session.

Single alpha (SA)

esc	a	b	c	d	e	f	g	h	i	j	\
tab	k	l	m	n	o	p	q	r	s	t	'
ctrl	u	v	w	x	y	z	,	.	/	;	alt
space: left trigger + X						backspace: ← + X					
shift: press left stick down						enter: Y					

Single QWERTY (SQ)

esc	q	w	e	r	t	y	u	i	o	p	\
tab	a	s	d	f	g	h	j	k	l	;	'
ctrl	z	x	c	v	b	n	m	,	.	/	alt
space: left trigger + X						backspace: ← + X					
shift: press left stick down						enter: Y					

Dual alpha (DA)

esc	a	b	c	d	e	p	q	r	s	t	\
tab	f	g	h	i	j	u	v	w	x	y	'
ctrl	k	l	m	n	o	z	,	.	/	;	alt
space: both triggers						backspace: ← ←					
shift: press either stick down						enter: Y					

Dual QWERTY (DQ)

esc	q	w	e	r	t	y	u	i	o	p	\
tab	a	s	d	f	g	h	j	k	l	;	'
ctrl	z	x	c	v	b	n	m	,	.	/	alt
space: both triggers						backspace: ← ←					
shift: press either stick down						enter: Y					

Figure 3. Onscreen keyboards for each of the 4 conditions in the user study. For the study, the onscreen interface was augmented with instructions on entering space, backspace, shift and enter.

Participants

We recruited 14 right-handed participants between the ages of 23 and 53 (average age 40). All had experience using computers, but none regularly played games. Nine subjects rated themselves as ‘non touch-typists’, where ‘touch typing’ referred to whether they needed to look at the keyboard while they typed.

Method

Throughout the experiment, a trial consisted of the subject reproducing a phrase displayed on screen by entering text in a text-edit box under the phrase (see Figure 4). The software described in [12] was used to present the target phrases and log all keystrokes and timing information. The phrases were drawn in a random order from the standard phrase set developed by [8]. The total time to enter each phrase was calculated as the duration between the first and last keystrokes made during the trial. Subjects advanced to the next trial by hitting the ‘enter’ key (the Y button in all of our conditions). Data such as words per minute (WPM, where a word is taken as 5 characters, including errors) and text entry errors were extracted from the log file by software also graciously provided by the authors of [12].

To familiarize subjects with the experimental procedure, we first administered a practice block in which they used a standard QWERTY keyboard to enter text.

We used a 2×2 within-subjects design with hands (single stick, dual stick) and key layout (QWERTY, alphanumeric) as factors. Blocks of 10 trials were preceded by 3 or more practice trials in which the experimenter coached the subject on the use of the particular game pad text entry technique. Blocks were fully counterbalanced, but because of the possibility of subjects’ confusion in the change of the implementation of space and backspace between single and dual stick conditions, both single stick conditions and both

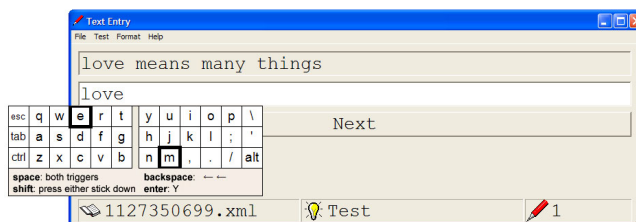


Figure 4. In each user study trial, the subject types the displayed phrase just below it. The onscreen keyboard is moved to follow the insertion point.

dual stick conditions were grouped together. Phrases during the practice trials were chosen to ensure that at least one capital letter appeared, so that subjects were familiar with the use of the ‘shift’ mode functionality during test trials. At the session’s conclusion, subjects answered a questionnaire in which they were required to rank their preference of the 4 text entry techniques.

Results

Measures of words per minute (WPM) were averaged across all trials and all subjects for each of the 4 conditions. These means, displayed in Table 1, indicate that users performed about 22% better in the DQ condition than the SA condition. These results obtained with first-time users are comparable to those of joystick EdgeWrite obtained after a longer practice session, and with much younger subjects (average age 21.5).

We performed a 2×2 Repeated Measures ANOVA on average WPM and found a significant effect for hands ($F(1,13) = 19.0, p < 0.001$) as well as key layout ($F(1,13) = 56.2, p < 0.001$). There were no significant interactions. Subsequent paired samples t-tests confirmed that DQ was faster than SQ ($t = 3.063, p < 0.009$) and that DQ was faster than DA ($t = 3.88, p < 0.002$).

Variability in performance among subjects was high. For example, the fastest subject averaged 9.1 WPM across all trials, while the slowest averaged 3.6 WPM. This variability inflates the standard error in Table 1. To remove some of this variability we normalized the WPM measurements for each subject by subtracting their overall average WPM, computed across all conditions (see Figure 5), which clearly illustrates that in terms of text entry speed, $DQ \gg DA, SQ \gg SA$, but $SQ \approx DA$. Furthermore, the increase associated in moving to a QWERTY layout was approximately equal, and additive, to the increase associated with the use of dual

Table 1. Mean WPM and error rate

WPM, error rate	alpha layout	QWERTY layout	Total
single stick	5.79 (± 0.37), 0.031 (± 0.0067)	6.48 (± 0.43), 0.044 (± 0.010)	6.14 (± 0.41), 0.037 (± 0.0087)
dual stick	6.42 (± 0.43), 0.064 (± 0.011)	7.08 (± 0.46), 0.054 (± 0.011)	6.75 (± 0.46), 0.059 (± 0.011)
Total	6.10 (± 0.43), 0.047 (± 0.010)	6.78 (± 0.44), 0.049 (± 0.010)	6.44 (± 0.44), 0.048 (± 0.010)

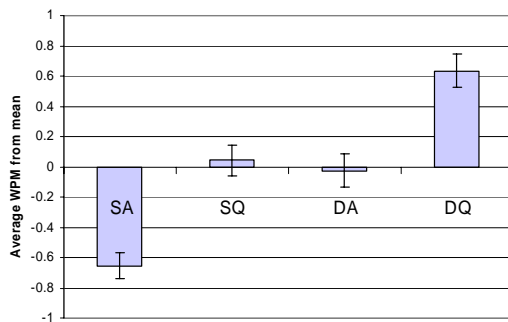


Figure 5. Mean-normalized WPM

sticks. We conclude that we can accept all three hypotheses.

While 9 subjects declared that they were not touch typists, in the regular typing practice block before the main experiment we observed only two subjects that we would judge as non-touch typists, and even they seemed to be familiar with the QWERTY layout. Since their performance did not trend differently than other subjects, we did not pursue further analysis regarding touch typing ability.

Errant keystrokes, including those corrected by the subject, were logged to generate an average total error rate for each subject and condition, listed in Table 1. We performed a 2 × 2 Repeated Measures ANOVA on the error rate to show that there was a significant effect for hands ($F(1,13) = 17.3$, $p < 0.001$) and no significant effect from key layout, indicating that there were more errors in the dual stick condition (see Table 1). We also confirmed that while subjects make more errors in the DA and DQ conditions, subjects were equally conscientious in correcting typos across all conditions, with corrected error rates showing no significant difference across conditions. Informally, we observed that in the DA and DQ conditions, many errors were caused by pulling the incorrect trigger, a behavior that is reminiscent of homologous errors in keyboarding [6]. We note that the error rate obtained in the fastest condition (DQ) is approximately half that of joystick EdgeWrite.

Friedman's χ^2 test on subjects' ranking of the four conditions revealed that subjects had a significant preference in ordering ($\chi^2(3, n=13) = 12.2$, $DQ = 1.77$, $SQ = 2.00$, $DA = 3.00$, $SA = 3.23$, $p < 0.007$). Subsequent Wilcoxon Signed Rank tests revealed that the difference in ranking between DQ and SA was significant ($Z = -2.3$, $p < 0.02$), while the difference in ranking DQ and SQ was not. In conclusion, even though subjects' preference for DQ was only very weak, their text entry performance gains with DQ were much greater than their preferences indicate.

FUTURE WORK AND CONCLUSION

While the present study demonstrates that naïve users are able to pick up the dual joystick text entry technique relatively quickly, it does not explore the effect that experience may have on performance of the technique. One of the authors, for example, has been able to achieve more than 13 WPM on the main condition with very little practice. Future

work may consider the effect of practice, experience with gaming and game controllers, and so on.

A drawback of the approach presented here is that onscreen keyboards in general tend to divide the user's attention between the onscreen keyboard itself and the typed text, and they can obscure parts of the interface. Future work should address making the technique more 'eyes-free'. One idea is to display only the half of the keyboard corresponding to the stick most recently in motion. It may be possible to further reduce the onscreen presentation by exploiting familiarity with QWERTY within each hand. For example, each of the three rows of the main part of the keyboard maps onto the 'up', 'center', and 'down' states of the stick, such that it may be possible to select at least the row in an eyes-free fashion assuming familiarity with QWERTY.

ACKNOWLEDGMENTS

We thank Edward Cutrell for assistance with the user study.

REFERENCES

1. KeyStick home page, <http://www.n-e-ware.com/KeyStick.htm>.
2. myText home page, <http://www.my-text.com>.
3. Weegie home page, <http://weegie.sourceforge.net>.
4. XNav: Gesture-based text input and device navigation, <http://www.microsoft.com/mscorp/ip/ventures/technologies/XNav.asp>.
5. Bellman, T., and MacKenzie, I. S., A probabilistic character layout strategy for mobile text entry. in *Proc. Graphics Interface (GI '98)*, (1998), 168-176.
6. Grudin, J. Error patterns in novice and skilled transcription typing. in Cooper, W.E. ed. *Cognitive aspects of skilled typewriting*, 1983, 121-143.
7. Isokoski, P., and Raisamo, R., Device independent text input: A rationale and an example. in *Proc. of the Working Conference on Advanced Visual Interfaces (AVI 2000)*, (2000), 76-83.
8. MacKenzie, I.S., and Soukoreff, R. W., Phrase sets for evaluating text entry techniques. in *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems (CHI 2003)*, (2003), 745-755.
9. MacKenzie, I.S., and Zhang, S. X., The design and evaluation of a high-performance soft keyboard. in *Proc. ACM Conference on Human Factors in Computing Systems (CHI '99)*, (1999), 25-31.
10. Perlin, K., Quikwriting: Continuous stylus-based text entry. in *ACM UIST '98*, (1998).
11. Venolia, G., and Neiberg, F., T-Cube: A fast, self-disclosing pen-based alphabet. in *Proc. CHI 1994*, (1994).
12. Wobbrock, J.O., and Myers, B. A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Trans. on CHI (submitted)*.
13. Wobbrock, J.O., Myers, B. A., and Aung, H. H., Writing with a joystick: A comparison of date stamp, selection keyboard and EdgeWrite. in *Proc. Graphics Interface (GI '04)*, (2004), 1-8.