

JoyKey: One-handed Hardware Keyboard with 4x3 Grid Slide Keys

Ryosuke Takada^(✉), Buntarou Shizuki, and Shin Takahashi

University of Tsukuba, Japan
{rtakada,shizuki,shin}@iplab.cs.tsukuba.ac.jp

Abstract. In this paper, we show a one-handed hardware keyboard with 4×3 grid slide keys, called a JoyKey. It can be used for Japanese text entry based on the flick input method. We conducted a user study to compare the accuracy and input speed of the JoyKey and a software keyboard on a touch screen under two conditions (eyes-free and non-eyes-free). Results show that the software keyboard cannot be used under the eyes-free condition because users cannot feel tactile feedback on a touch screen. In contrast, using the JoyKey, users can enter text under both the conditions with relative ease.

Keywords: Text entry · KANA input · Flick input · Joystick · Multi input · Mouse · Game controller

1 Introduction

The flick input method, which originated from Apple Newton and was adopted into Apple iPhone, is now one of the de facto standard Japanese input methods in small mobile devices with touch screens. This method is designed for entering Japanese KANA letters as shown in Fig. 1. The keyboard consists of twelve keys aligned into a 4×3 grid: 10 consonant letter keys and punctuation key. A Japanese KANA letter is transcribed into one consonant and one vowel as shown in Table 1. There are ten consonants and five vowels in Japanese. With the flick input method, a user touches one of the consonant keys to enter a consonant. In response, a menu appears with five items corresponding to the five vowels. To enter a vowel, the user selects an item by simply releasing the finger or performing a flick gesture (upward, downward, rightward, or leftward).

In contrast, we developed a one-handed hardware flick text entry keyboard, called a “JoyKey” as shown in Fig. 2. It has twelve keys aligned into a 4×3 grid, each of which is composed of a joystick and a tact switch that can detect X and Y axis movements. Using JoyKey, the user can enter Japanese KANA letters in an eyes-free manner because each key provides a tactile feedback. In contrast to the one that Google Japan released on April 1st, 2016 [4], whose size is large (240 mm × 180 mm × 50 mm) and must be placed on a desk to be used, the user can use JoyKey with one hand due to its small size (59 mm × 85 mm × 24.5 mm). Moreover, JoyKey can be used as a keyboard, mouse, and video game controller as well, as shown in Fig. 3.

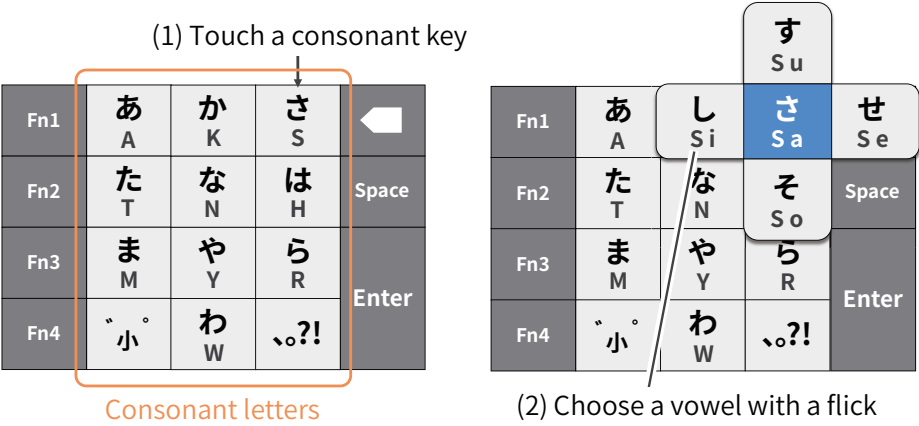


Fig. 1. Flick input

Vowels	Consonants											
	Basic letter											
	A	K	S	T	N	H	M	Y	R	W	-	
a	あ	か	さ	た	な	は	ま	や	ら	わ		が
i	い	き	し	ち	に	ひ	み		り			ぎ
u	う	く	す	つ	ぬ	ふ	む	ゆる				ぐ
e	え	け	せ	て	ね	へ	め	れ				げ
o	お	こ	そ	と	の	ほ	も	よ	ろ	を		ご
-											ん	
-											ー	

Table 1. Japanese syllabary (‘S*’ can be transcribed into the combination of consonant ‘S’ and symbol ‘*’, and can be transcribed as ‘G’ in a phonetic alphabet. The same applies to other voiced letters and p-sound letters.)

In addition, we conducted a user study in order to compare the accuracy and input speed of the software keyboard on a touch screen and JoyKey under two conditions (non-eyes-free and eyes-free). We will show the procedure and discussion of the user study in subsequent sections.

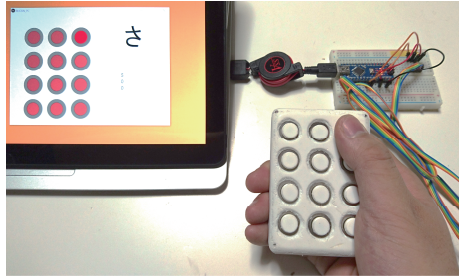
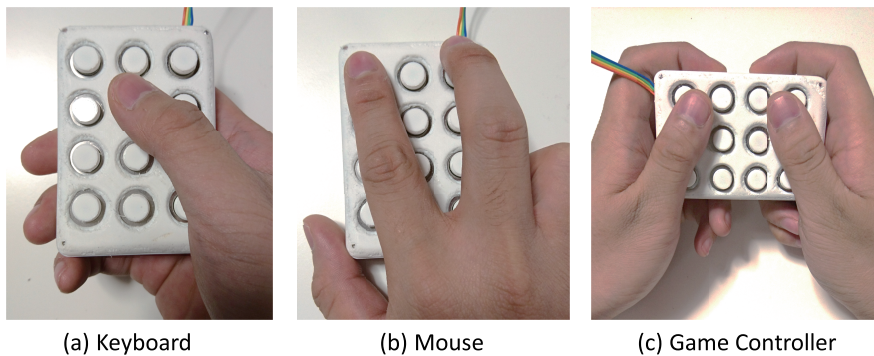


Fig. 2. JoyKey



(a) Keyboard

(b) Mouse

(c) Game Controller

Fig. 3. Different uses of a JoyKey

2 Related Work

JoyKey is based on the following previous works.

2.1 Japanese KANA Letters Entry

Since JoyKey is a Japanese KANA letter entry system based on flick input method, we will describe three conventional methods of entering Japanese KANA letters.

QWERTY

Japanese speakers use QWERTY keyboard to enter KANA letters when they use desktop and laptop computers. Japanese QWERTY keyboards have both KANA and Roman letters printed on key tops as shown in Fig. 4. Japanese speakers can enter KANA letters by KANA or Roman input modes. KANA input mode is used to directly input a KANA letter. The KANA letters are also arranged in a consistent way across different keyboards. For example, the ‘Q’, ‘W’, ‘E’, ‘R’, ‘T’, and ‘Y’ keys correspond to “た”, “て”, “い”, “す”, “か”, and “ん”, respectively. In Roman input mode, a number of Japanese input method editors allow Japanese text to be entered using Roman letters, which can then be converted to KANA, based on the Japanese syllabary as shown in Table 1. This method does not require the use of a Japanese keyboard with KANA letters.



Fig. 4. Japanese QWERTY keyboard

Multi-tap Entry

On Japanese cell phones with no touch screens, hardware keyboards are usually used. Fig. 5 outlines a typical key layout for such keyboards. This key layout consists of 12 keys, including 10 consonant keys, that are arranged in a 4×3 grid. Users enter a KANA letter by pressing a consonant key one to five times (i.e., multi-tap) to enter a vowel. A special KANA letter can be input using the other two keys. For example, “ぎ”, which is transcribed into ‘Gi*’ is entered by pressing the ‘2’ key twice, and then pressing the ‘*’ key once.

Pocket Bell Entry

In Japan, pagers called *pocket bells* were widely used as personal communication tools before mobile phones became popular. To enter a letter on this device, users used an input method called *pocket bell entry*. In this method, a letter is coded with two digits as shown in Table 2. Therefore, users can enter a KANA letter by pressing two numeric keys sequentially. For example, users input “し” by pressing the ‘3’ key followed by the ‘2’ key.

1 あいうえお	2 かきくけこ	3 さしすせそ
4 たちつてと	5 なにぬねの	6 はひふへほ
7 まみむめも	8 やゆよ	9 らりるれろ
* ° ° ~	0 わをんー	#

Fig. 5. Key layout for multi-tap entry

First input	Second input									
	1	2	3	4	5	6	7	8	9	0
1	あ	い	う	え	お	A	B	C	D	E
2	か	き	く	け	こ	F	G	H	I	J
3	さ	し	す	せ	そ	K	L	M	N	O
4	た	ち	つ	て	と	P	Q	R	S	T
5	な	に	ぬ	ね	の	U	V	W	X	Y
6	は	ひ	ふ	へ	ほ	Z	?	!	-	/
7	ま	み	む	め	も	¥	&			
8	や		ゆ		よ	*	#			
9	ら	り	る	れ	ろ	1	2	3	4	5
0	わ		を		ん	6	7	8	9	0

Table 2. Letter correspondence table for pocket bell entry

2.2 Multi Input on a Hardware Keyboard

There are many research projects of gestures on a keyboard to improve the input performance. FlickBoard [12] uses a silicon keyboard cover that contains a capacitive sensing grid for the detection of pointing and gestures. Keyboard Clawing [9] is an input technique that can detect keytop clawing gestures using acoustic sensing. Dietz et al. [1] proposed a technique to sense the force level on membrane keyboards. GestKeyboard [16] is a technique to detect gestures by using intervals between key presses on an ordinary unmodified keyboard. Type-Hover-Swipe in 96 Bytes [11] detects mid-air gestures on a keyboard using 64 photo reflectors. Surfboard [7] recognizes swipe left/right gestures by using a built-in microphone on the hardware keyboard.

2.3 Text Entry using Joystick

Various text entry methods using joystick have been proposed in the past. Wobbrock et al. [14] proposed text entry using joystick based on EdgeWrite [15]. TwoStick [8] is an alphabet text input method using a game controller's two joysticks. It divides the input area into 3×3 zones, each of which has 3×3 grids. Each grid represents one letter, hence there are total 81 different letters that can be entered. The user can select a letter by first selecting a zone with one joystick and then selecting a grid within that zone with the other joystick. IToNe [2] is a Japanese text input method using a game controller's two joysticks. It uses one joystick to select a consonant, while the other to select a vowel. Wilson and Agrawala [13] used a game controller's two joysticks to select letters from two halves of a split software keyboard. Isokoski and Raisamo [6] reported a text entry rate using two joysticks and Quikwriting [10].

3 Implementation of JoyKey

JoyKey is a hardware keyboard based on flick input. It is designed to be used with one hand. JoyKey is connected to a PC via Arduino Leonardo board.

3.1 JoyKey

Joykey ($81 \text{ mm} \times 59 \text{ mm} \times 24.5 \text{ mm}$) is a hardware keyboard that has 12 keys aligned into a 4×3 grid. Fig. 6 shows the construction of a JoyKey. Each key is composed of a tact switch (Switronic Industrial Corp., 1273) and a joystick (Top-Up Industry Corp., JT8P-3.2T) that can detect X and Y axis movements. To detect simultaneous key presses and reduce wiring, each key has a microcontroller (Atmel, ATTiny85) connected to the joystick and a tact switch, and this microcontroller is assigned to a unique I²C address. These microcontrollers are connected to Arduino Leonardo by I²C. When a user presses or flicks the key, he/she can feel tactile feedback from the key's joystick and tact switch.

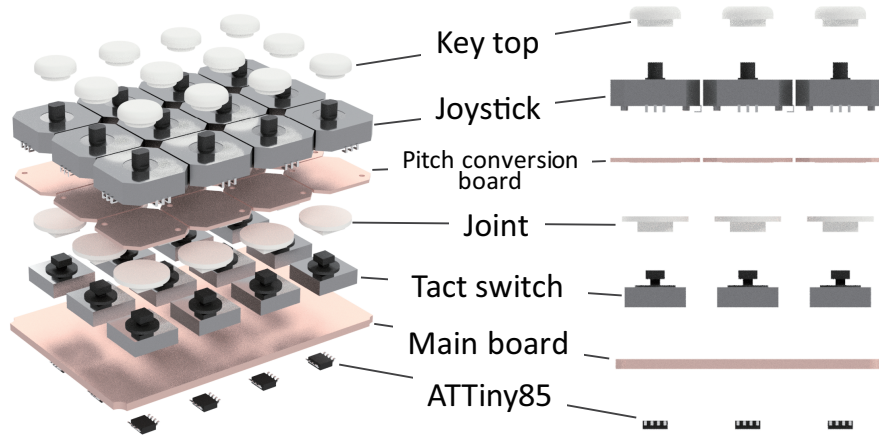


Fig. 6. Construction of a JoyKey

3.2 Key Layout

The user holds the JoyKey in one hand and types a letter with his/her thumb as shown in Fig. 3a. Fig. 7 shows JoyKey’s 4×3 keys, which correspond to the ten consonants, comma/period, and marks. Moreover, in JoyKey, there is no “Enter”, “Space”, “Next”, or “Backspace” key, which differs from the keyboard for flick input method on touch screens. Instead, we assigned simultaneous pressing of these special letters such as pressing ‘S’ and ‘!’ keys, using the pad and the ball of the thumb, to “Enter” and left flick of the ‘S’ key, without pressing, to “Backspace”. This is because if the button to input keys are added, its size increases and thus it becomes difficult to grasp. its size increases and it becomes difficult to grasp.

Note that, for flick input on touch screens, the bottom row keys cannot be flicked downwards because in such a scenario, the finger would leave the input area. On the other hand, JoyKey is able to detect downward flicks on the bottom row keys, which can then be assigned to other key inputs, such as “Shift”, “Tab”, Kana/Roman/Number input switch, etc.

4 User Study

We conducted a user study to compare the accuracy and input speed of JoyKey to that of a software keyboard. This user study was conducted under two conditions: non-eyes-free and eyes-free.

4.1 Participants

The participants were six university graduates in our laboratory (23–24 years old; male; right handed). They have been using smartphones from 24 to 106 months

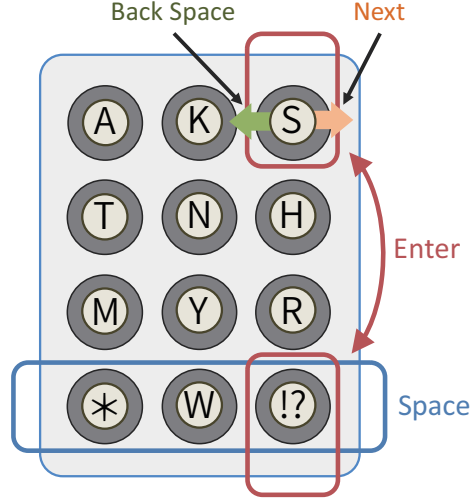


Fig. 7. Key layout of JoyKey

($M = 67.17$). All participants were familiar with the flick input keyboard on smartphones. We instructed them to use flick input with their right hand.

4.2 Devices and Software

The main device is the JoyKey (81 mm \times 59 mm \times 24.5 mm) connected to a PC. As the software keyboard, we used Google Japanese input [3] on Xperia Z5 Compact E5823 (Android OS 6.0.1; 4.6-inch screen with a resolution of 720 px \times 1280 px; 127 mm \times 65 mm \times 8.9 mm). Flick input of the smartphone sends keyboard events to the PC via Wi-Fi using the Intel Remote Keyboard application [5]. Fig. 8 and Fig. 9 show the input area sizes of the two keyboards.

For investigating the accuracy and time of text entry, we implemented a typing game as shown in Fig. 10. The typing game was implemented in Processing 3.1.1, and running on VAIO Z VJZ13A1 (OS: Windows 10 Pro 64bit, CPU: Intel Core i7-5557U 3.10 GHz, memory: 16.0 GB). It randomly displays a KANA word and measures the time, number of backspaces, and error inputs until the user presses the enter key when he/she finishes. For the KANA words, some Japanese daily life words that consists of 3–7 letters were used, such as greetings, days of the weeks, months, family relationships, subjects of study, fruits, foods, drinks, animals, flowers, places, vehicles, and clothings.

4.3 Procedure and Task

The user study was conducted in a quiet office environment. First, the purpose of the user study was explained to the participants.



Fig. 8. Software keyboard size

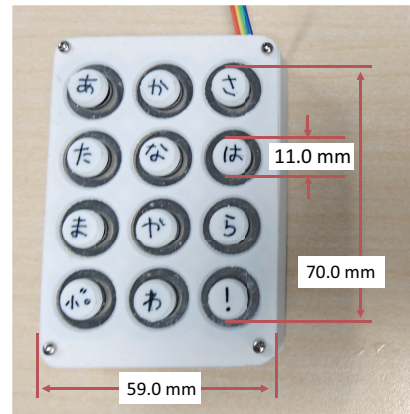


Fig. 9. JoyKey size

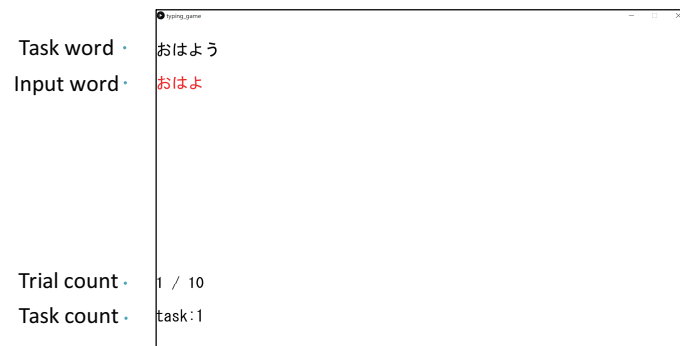


Fig. 10. Typing game

We divided the participants into two groups: one group performed the tasks with JoyKey first; the other group performed the tasks with the software keyboard first. Both the groups performed the tasks in the order of non-eyes-free condition to eyes-free condition.

Participants first practiced a keyboard until they got familiar with it. In each trial, the participants entered a word displayed in the typing game. One task consisted of 10 trials with five tasks in each session. We conducted one session for each keyboard as shown in Fig. 11. We also gave a break longer than a minute to avoid fatigue between the sessions. If the participants felt tired, they could freely take a break between the tasks.

In summary, the user study design involved:

2 keyboards \times
 2 conditions \times
 5 tasks \times
 10 trials
 = 200 trials per participant.

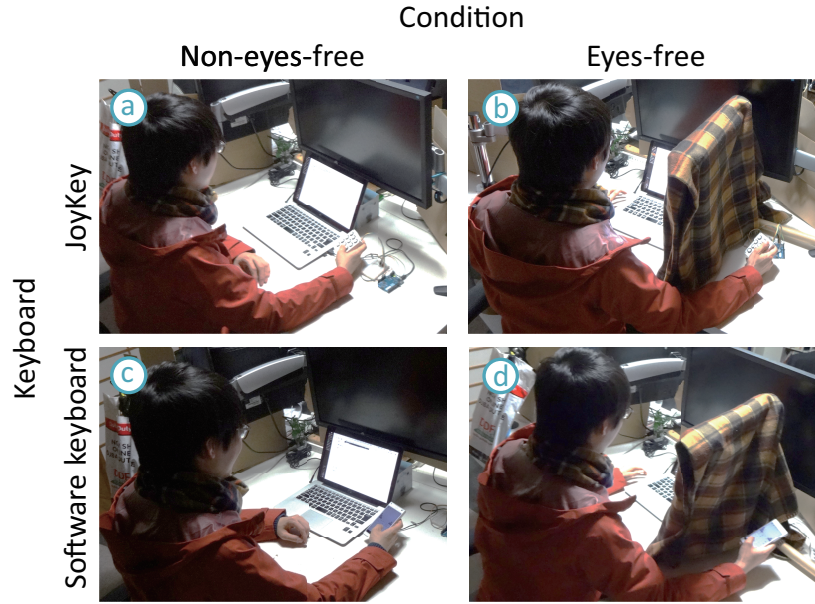


Fig. 11. Experimental setup for each condition (a: JoyKey and non-eyes-free condition, b: JoyKey and eyes-free condition, c: software keyboard and non-eyes-free condition, d: software keyboard and eyes-free condition).

After finishing the sessions, we asked them to answer the questionnaire with the following four questions with five-point Likert scales (1: easy – 5: difficult).

- Is it easy to operate JoyKey with non-eyes-free?
- Is it easy to operate JoyKey with eyes-free?
- Is it easy to operate software keyboard with non-eyes-free?
- Is it easy to operate software keyboard with eyes-free?

The user study took about one hour for each participant.

5 Result and Discussion

While the text entry speed is described in words per minute (wpm) for Roman letters, we use KANAs per minute (kpm) to describe a text entry speed of Japanese KANA letters. Fig. 12 shows kpm under each condition (error bars show standard deviations). The mean text entry speed of JoyKey in kpm in the two conditions are 24.95 (SD = 2.23) and 25.67 (SD = 3.80), respectively; The mean text entry speed of the software keyboard in kpm are 55.52 (SD = 8.06) and 20.71 (SD = 5.69), respectively. Fig. 13 shows backspace count per letter under each condition (error bars show standard deviations). The mean backspace count per letter of JoyKey under the two conditions are 0.269 (SD = 0.116) and 0.299 (SD = 0.121), respectively; the mean backspace count per letter of the software keyboard are 0.048 (SD = 0.036) and 1.208 (SD = 0.772), respectively.

Since Levene’s test showed homogeneity of variances ($p = 0.092 > 0.05$) in the result of the text entry speed, we ran a two-way factorial ANOVA. As we observed a significant interaction effect between the kind of keyboard and two conditions ($p = 0.000 < 0.05$), we ran a Scheffe’s test for posthoc analysis. The result shows a significant difference between the non-eyes-free condition with the software keyboard and the other conditions, especially between the two conditions with the software keyboard.

We ran a Kruskal-Wallis test to examine the result of backspace count per letter, since Levene’s test showed heterogeneity of variances ($p = 0.019 < 0.05$) in the data. The result shows a significant difference between the two conditions with the software keyboard ($\chi^2 = 18.42$, $df = 3$, $p = 0.000$).

The above results revealed that the eyes-free condition is significantly worse than the non-eyes-free condition with the software keyboard. This means that the user cannot enter letters under the eyes-free condition correctly with the software keyboard. During the user study, we observed that the participants accidentally tapped the ”prediction area” of the software keyboard frequently. As this result of this, the users had to delete those mistakes by using the Backspace key. In contrast, with JoyKey, the results of text entry speed and backspace count both did not change significantly between the eyes-free and non-eyes-free conditions.

Fig. 14 shows the result of the questionnaire. All participants answered that the eyes-free condition was very difficult, and the non-eyes-free condition was very easy with the software keyboard. This would be because there was no haptic feedback on software keyboard; thus the participant could not know the touch position on the smartphone. On the other hand, for the JoyKey, the results were similar for both conditions.

In summary, these results suggested that the user can use JoyKey to input letters efficiently under the eyes-free condition, because the users can sense each key by the tactile feedback.

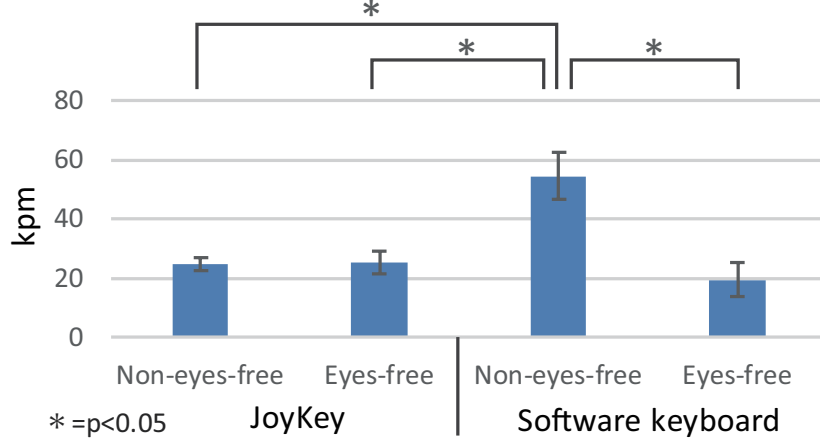


Fig. 12. Text entry speed in kpm under each condition (error bars show standard deviations)

6 Different uses of a JoyKey

Since JoyKey has twelve keys, each of which consists of a joystick and a tact switch, it can provide various input modes. Currently, JoyKey can be used in three modes: flick input mode described above and the two modes described in this session. The user can switch between these three modes by pressing two keys simultaneously.

6.1 Mouse

In this mode, the user places his/her hand on the JoyKey as shown in Fig. 3b. The user can move the cursor using the joystick of the top-left key. Furthermore, the user can perform left/right clicks and scroll operations using the top row keys as shown in Fig. 15 (left picture). Typically, Ctrl + scroll is used for scaling. In JoyKey, the user can perform pinch in/out gestures for scaling using the two keys as shown in Fig. 15 (right picture).

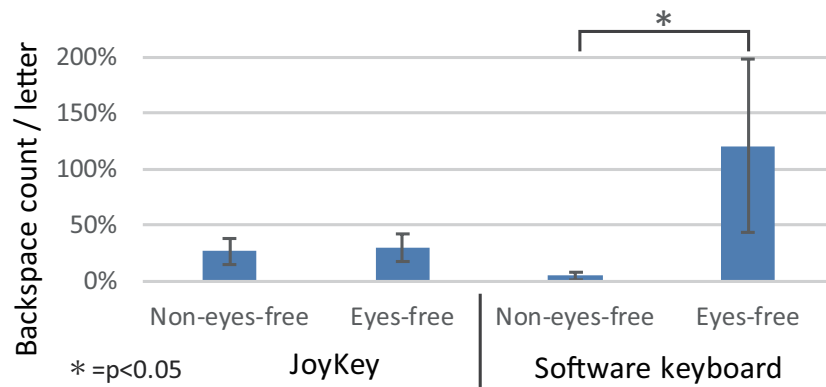


Fig. 13. Backspace count per letter under each condition (error bars show standard deviations)

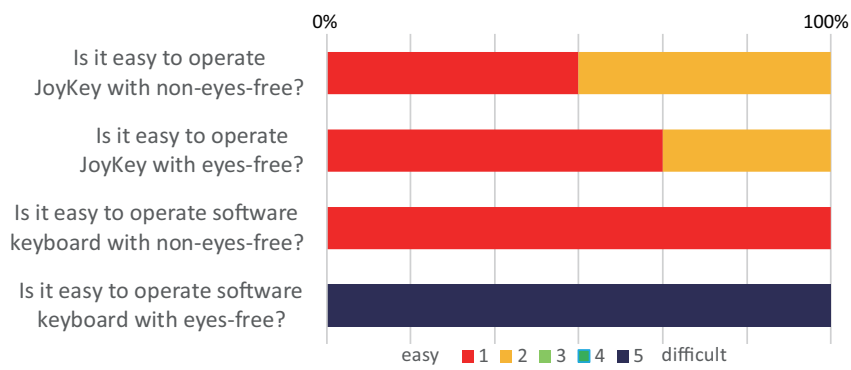


Fig. 14. Result of the questionnaire

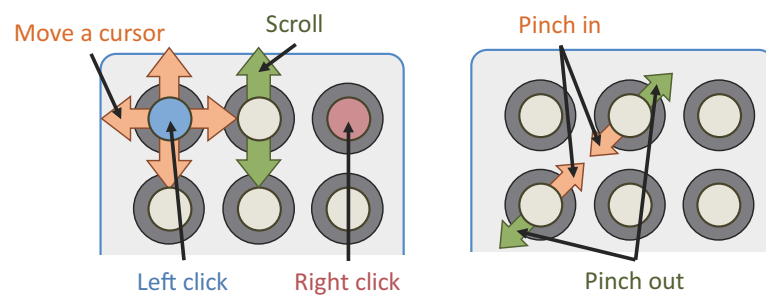


Fig. 15. Mouse mode

6.2 Game Controller

In this mode, the user holds the JoyKey with both hands as shown in Fig. 3c. The user can input commands of a video game as shown in Fig. 16. Game creators can design a suitable key assignment for each game.

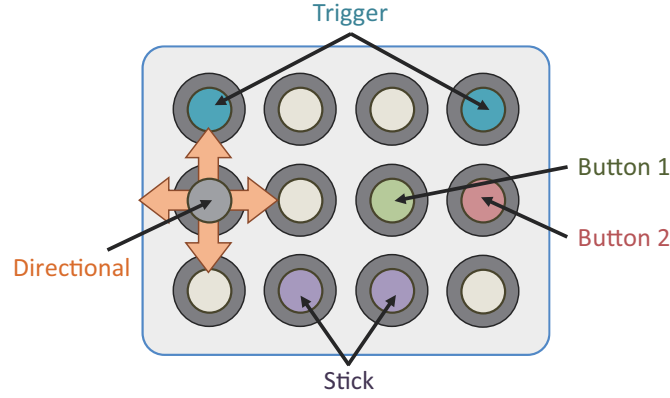


Fig. 16. Game controller mode (an example)

7 Conclusion

In this paper, we presented the JoyKey, which is a one-handed hardware keyboard for Japanese KANA input based on flick input method. We conducted a user study to compare the accuracy and input speed between JoyKey and software keyboard. Results show that the participants using the software keyboard cannot feel where they were touching and were unable to enter letters correctly under the eyes-free condition. In contrast, the participants could enter letters correctly under the eyes-free condition with the JoyKey because they could feel tactile feedback.

References

1. Dietz, P.H., Eidelson, B., Westhues, J., Bathiche, S.: A practical pressure sensitive computer keyboard. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology. pp. 55–58. UIST '09, ACM, New York, NY, USA (2009)
2. Go, K., Konishi, H., Matsuura, Y.: IToNe: A Japanese text input method for a dual joystick game controller. In: CHI '08 Extended Abstracts on Human Factors in Computing Systems. pp. 3141–3146. CHI EA '08, ACM, New York, NY, USA (2008)

3. Google Japan Inc.: Google Japanese text entry, <https://www.google.co.jp/ime/>, (in Japanese, retrieved February 10, 2017)
4. Google Japan Inc.: Google Japanese text entry hardware flick version, <https://www.google.co.jp/ime/furikku/>, (in Japanese, retrieved February 10, 2017)
5. Intel Corporation: Introducing the Intel Remote Keyboard, <http://www.intel.com/content/www/us/en/compute-stick/intel-remote-keyboard.html>, (retrieved February 10, 2017)
6. Isokoski, P., Raisamo, R.: Quikwriting as a multi-device text entry method. In: Proceedings of the Third Nordic Conference on Human-computer Interaction. pp. 105–108. NordiCHI '04, ACM, New York, NY, USA (2004)
7. Kato, J., Sakamoto, D., Igarashi, T.: Surfboard: Keyboard with microphone as a low-cost interactive surface. In: Adjunct Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology. pp. 387–388. UIST '10, ACM, New York, NY, USA (2010)
8. Költringer, T., Isokoski, P., Grechenig, T.: TwoStick: Writing with a game controller. In: Proceedings of Graphics Interface 2007. pp. 103–110. GI '07, ACM, New York, NY, USA (2007)
9. Kurosawa, T., Shizuki, B., Tanaka, J.: Keyboard Clawing: Input method by clawing key tops. In: Proceedings of 15th International Conference on Human-Computer Interaction. pp. 272–280. HCII '13, ACM, Nevada (2013)
10. Perlin, K.: Quikwriting: Continuous stylus-based text entry. In: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology. pp. 215–216. UIST '98, ACM, New York, NY, USA (1998)
11. Taylor, S., Keskin, C., Hilliges, O., Izadi, S., Helmes, J.: Type-Hover-Swipe in 96 Bytes: A motion sensing mechanical keyboard. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems. pp. 1695–1704. CHI '14, ACM, New York, NY, USA (2014)
12. Tung, Y.C., Cheng, T.Y., Yu, N.H., Wang, C., Chen, M.Y.: FlickBoard: Enabling trackpad interaction with automatic mode switching on a capacitive-sensing keyboard. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. pp. 1847–1850. CHI '15, ACM, New York, NY, USA (2015)
13. Wilson, A.D., Agrawala, M.: Text entry using a dual joystick game controller. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 475–478. CHI '06, ACM, New York, NY, USA (2006)
14. Wobbrock, J.O., Myers, B.A., Aung, H.H.: Joystick text entry with date stamp, selection keyboard, and EdgeWrite. In: CHI '04 Extended Abstracts on Human Factors in Computing Systems. pp. 1550–1550. CHI EA '04, ACM, New York, NY, USA (2004)
15. Wobbrock, J.O., Myers, B.A., Kembel, J.A.: EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology. pp. 61–70. UIST '03, ACM, New York, NY, USA (2003)
16. Zhang, H., Li, Y.: GestKeyboard: Enabling gesture-based interaction on ordinary physical keyboard. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems. pp. 1675–1684. CHI '14, ACM, New York, NY, USA (2014)