Proc. of the 5th International Conference on Entertainment Computing (ICEC2006),
pp. 322--325 (Sep. 2006).

TEMPEST: A Text Input System for Musical Performers

Yoshinari TAKEGAWA[1], Tsutomo TERADA[1], and Shojiro NISHIO[1]

Grad. Sch. of Information Sci. and Tech. Osaka Univ. Osaka, JAPAN
{takegawa, tsutomu, nishio}@ist.osaka-u.ac.jp

Abstract. Recently, due to the widespread use of computers, text-based communication methods, such as e-mail, blog, and chat, have attracted a great deal of attention. Moreover, various communication media, such as picture and movie, are used for expressing own emotion. If pianists can apply their musical expressions to the text input, they can input texts with their own emotion richly. Therefore, the goal of our study is to construct TEMPEST (TExt input and Musical PErforming SysTem) that is a text input system to input texts with various musical expressions by using a MIDI keyboard. Since this system provides text input methods considering musical artistry, a performer can input texts like playing music. As the result of a questionnaire evaluation, it proves that proposed methods have high effectiveness for expressing emotion in inputting texts.

1 Introduction

Recently, due to the widespread use of computers, text-based communication methods, such as e-mail, blog, and chat, have attracted a great deal of attention. Moreover, various communication media, such as pictures and movies, are used for expressing own emotion, and various new interfaces for such purposes are studied and developed[2, 4, 6].

Here, the best device for pianists to express their emotion is obviously the piano. A pianist creates a marvelous sense of his emotion using a piano. If pianists can apply their musical expressions to the text input, they can input texts with their own emotion richly. Therefore, the goal of our study is to construct TEMPEST (TExt input and Musical PErforming SysTem) that is a text input system to input texts with various musical expressions by using a MIDI keyboard. Since this system provides several text input methods considering musical artistry, a performer can input texts like playing music.

The remainder of this paper is organized as follows. Section 2 explains the design of TEMPEST, and Section 3 describes the implementation. Section 4 describes an evaluation of TEMPEST. Finally, Section 5 presents conclusions and our planned future work.

2 Design of TEMPEST

TEMPEST is not only a text input interface but also a musical instrument. That is to say, TEMPEST should have functions not only for the speed and the

Table 1. The basic kana characters

Consonant \ Vowel	a	i	u	e	o
Null	あ [a]	い [i]	う [u]	え [e]	お [o]
k	か [ka]	き [ki]	く [ku]	け [ke]	こ [ko]
s	さ [sa]	し[si]	す [su]	せ [se]	そ [so]
t	た [ta]	ち [ti]	つ [tu]	て [te]	と [to]
n	な [na]	に [ni]	ぬ [nu]	ね [ne]	の [no]
h	は [ha]	ひ [hi]	ふ [hu]	へ [he]	ほ [ho]
m	ま [ma]	み [mi]	む [mu]	め [me]	も [mo]
y	や [ya]		ゆ [yu]		よ [yo]
r	ら [ra]	り [ri]	る [ru]	れ [re]	ろ [ro]
w	わ [wa]				を [wo]
n	ん [n]				

accuracy of text input but also for expressing emotions. Therefore, we design TEMPEST that it always outputs sound at a character input and it makes a sequence of output sounds artistic. We design TEMPEST under the following policies:

Elements used for text-input
In playing a MIDI keyboard, the keyboard can get various information, such as note number, velocity, fingering, time value, and rhythm. Since TEMPEST is a text input system, we employ stable and strict information for text input, which is only the note number. This is because other information, such as velocity and rhythm, needs calibration and clustering to adjust to individuals.

Hardware
We assume that performers use TEMPEST anytime and anywhere. Therefore, we suppose any MIDI keyboards whose numbers of keys are at least 2 octaves or more, to use all functions of TEMPEST.

Text Input Type
Using TEMPEST, a user can input kana characters, alphameric characters, and special characters such as “Shift” and “Ctrl”. Here, kana is the syllabic Japanese scripts used for constructing words. Table 1 shows the kana characters, and the neighbor symbol of a kana character expresses pronunciation in English. In this paper we use that symbol (“[**]”) instead of kana characters. By way of exception, the “[a], [i], [u], [e], [o]” consists of only one vowel, and “[n]” consists of only one consonant. The number of consonants/vowels is ten/five. Japanese people are familiar with this relationship between consonants and vowels.

There are various text input methods such as QWERTY method, Multi-tap method, Two-key method, and Chording method. Most common keyboards for PC employ the QWERTY method, each key represents one character. Multi-tap method is currently the most common text input method for mobile phones. In this method, a user presses a key one or more times to specify a character. In Two-key method, the user presses the combination of two keys to specify

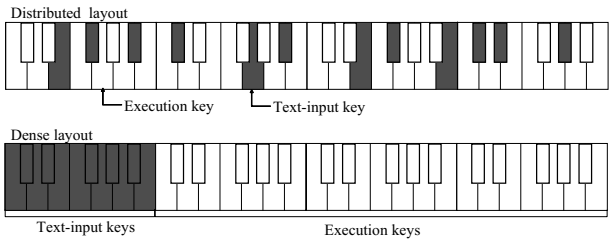


Fig. 1. Layouts of Text-input/Execution keys

one character. In Chording method, multiple keys are pressed simultaneously to specify a character, instead of pressing keys in sequence.

Although the QWERTY method is the simplest, it needs many keys. Therefore, it is not applicable to small keyboards whose numbers of keys are few. Moreover, since a user has to press the same key more than once in Multi-tap method, it does not look like a piano performance. As a result, TEMPEST employs Two-key method and Chording method to input texts.

Key layout

As we described, TEMPEST has three types of characters, and a user changes modes to select the type of characters. The maximum number of symbols for one mode is 46, which is for the kana characters. Two-key method needs at least seven keys to input all kana characters, and Chording method, which consists of two- or three-note chords, required seven keys (${}_7C_3+{}_7C_2= 56$ possible chords).

All keys in a keyboard are divided into two types in our method; Text-input keys and Execution keys. The former is used for inputting characters with sounds, and the latter is used for only playing sound. Moreover, there are two layout methods for Text-input keys as shown in Figure 1; the distributed layout and the dense layout. In the distributed layout, performers often need a big movement of the hands to specify a character. Moreover, the layout of Text-input keys changes according to the number of keys in the keyboard. Therefore, we employ the dense layout.

We call the area that includes Text-input/Execution keys Text-input area/ Execution area. TEMPEST can layout multiple Text-input areas with different text input methods as shown in Figure 2.

2.1 Text Input Methods

We propose three types of text input method; Monophony method, Polyphony method, and Accompaniment method. The first one is based on Two-key method, and the others are based on Chording method. We design the text input method to input kana characters easily.

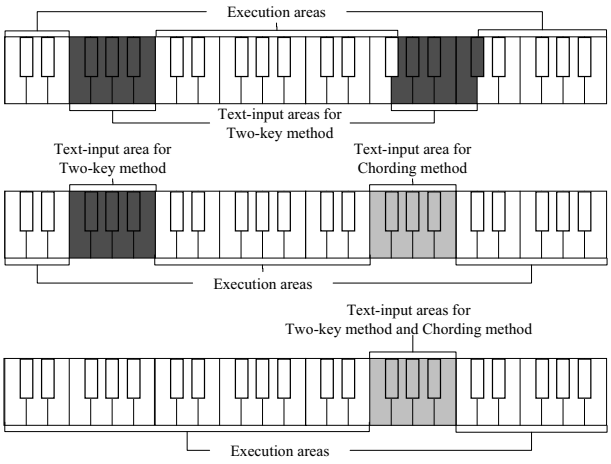


Fig. 2. Layouts of Text-input/Execution area

Monophony method

In Monophony method, that is the based on Two-key method, a user presses two keys successively to specify the input characters. Table 2 (a) shows the character-code chart in Monophony method. It uses 10 keys from C note to A note. Users can input kana characters easily, because the first key corresponds to the consonant, and the second key corresponds to the bowel. To generate "[ku]", the user presses C# key followed by D key.

The default input mode of Monophony method is kana mode. If a user wants to change the mode, the user inputs "MS (Mode Shift)". When the user selects "IC (Input Conversion)", the last entered character is changed. To be more accurate, when the last entered alphabetical character is lower-case/upper-case character, the character is changed upper-case/lower-case. Moreover, when the last entered character is kana character, the system changes the entered kana character into other signages of the kana character.

The meanings of "MS" and "IC" do not change in the other methods.

Polyphony method

Instead of pressing keys in sequence to specify the input character, multiple keys are pressed simultaneously in Polyphony method. Table 2 (b) shows the character-code chart in Polyphony method. This method uses 11 keys from C note to A# note chords that have two notes and three notes. A user can input kana characters easily, because the white keys correspond to the consonant, and the black keys correspond to the bowel. To generate "[ku]", the user presses C, D, F# keys simultaneously. Since this method and Monophony method are independent. They can be used at the same time as shown in the bottom of

Table 2. The character code chart in three text input methods

(a) Monophony method

Kana

First \ Second	C	C#	D	D#	E	F
C	[a]	[i]	[u]	[e]	[o]	Enter
C#	[ka]	[ki]	[ku]	[ke]	[ko]	Space
D	[sa]	[si]	[su]	[se]	[so]	Back
D#	[ta]	[ti]	[tu]	[te]	[to]	**MS**
E	[na]	[ni]	[nu]	[ne]	[no]	**IC**
F	[ha]	[hi]	[hu]	[he]	[ho]	
F#	[ma]	[mi]	[mu]	[me]	[mo]	
G	[ya]	,	[yu]	.	[yo]	
G#	[ra]	[ri]	[ru]	[re]	[ro]	
A	[wa]	[wo]	[n]	!	?	

Alphameric

First \ Second	C	C#	D	D#	E	F
C	a	b	c	d	e	Enter
C#	f	g	h	i	j	Space
D	k	l	m	n	o	Back
D#	p	q	r	s	t	**MS**
E	u	v	w	x	y	**IC**
F	z	!	?	.	,	
F#	1	2	3	4	5	
G	6	7	8	9	0	
G#	+	−	*	/	=	
A	(	)	{	}	^	

Special

First \ Second	C	C#	D	D#	E	F
C	Esc	Tab	Ctrl	Shift	Alt	Enter
C#	PrSc	Del	PgUp	PgDn	Ins	Space
D	ScrLk	Pause	Home	End	\	Back
D#	[	]	#	$	%	**MS**
E	&	|	~	_	@	**IC**
F	<	>	;	:		
F#	Left	Up		Down	Right	
G	F1	F2	F3	F4	F5	
G#	F6	F7	F8	F9	F10	
A	F11	F12	'	`	"	

(b) Polyphony method

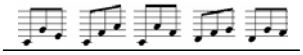
Kana	C#	D#	F#	G#	A#
C	[a]	[i]	[u]	[e]	[o]
C, D	[ka]	[ki]	[ku]	[ke]	[ko]
D	[sa]	[si]	[su]	[se]	[so]
D, E	[ta]	[ti]	[tu]	[te]	[to]
E	[na]	[ni]	[nu]	[ne]	[no]
E, F	[ha]	[hi]	[hu]	[he]	[ho]
F	[ma]	[mi]	[mu]	[me]	[mo]
F, G	[ya]	,	[yu]	.	[yo]
G	[ra]	[ri]	[ru]	[re]	[ro]
G, A	[wa]	[wo]	[n]	!	?
A	Enter	Space	Back	**MS**	**IC**

Alphameric	C#	D#	F#	G#	A#
C	a	b	c	d	e
C, D	f	g	h	i	j
D	k	l	m	n	o
D, E	p	q	r	s	t
E	u	v	w	x	y
E, F	z	!	?	.	,
F	1	2	3	4	5
F, G	6	7	8	9	0
G	+	−	*	/	=
G, A	(	)	{	}	^
A	Enter	Space	Back	**MS**	**IC**

Special	C#	D#	F#	G#	A#
C	Esc	Tab	Ctrl	Shift	Alt
C, D	PrSc	Del	PgUp	PgDn	Ins
D	ScrLk	Pause	Home	End	\
D, E	[	]	#	$	%
E	&	|	~	_	@
E, F	<	>	;	:	
F	Left	Up		Down	Right
F, G	F1	F2	F3	F4	F5
G	F6	F7	F8	F9	F10
G, A	F11	F12	'	`	"
A	Enter	Space	Back	**MS**	**IC**

(c) Accompaniment method

Kana



[a]	[i]	[u]	[e]	[o]
C, G, E	C, F, A	C, A, F	D, F, G	D, G, F
[ka]	[ki]	[ku]	[ke]	[ko]
C, G, E	C, F, A	C, A, F	D, F, G	D, G, F
[sa]	[si]	[su]	[se]	[so]
D, A, F	D, G, H	D, H, G	E, G, A	E, A, G
[ta]	[ti]	[tu]	[te]	[to]
D, A, F	D, G, H	D, H, G	E, G, A	E, A, G
[na]	[ni]	[nu]	[ne]	[no]
E, H, G	E, A, C"	E, C", A	F, A, H	F, H, A
[ha]	[hi]	[hu]	[he]	[ho]
E, H, G	E, A, C"	E, C", A	F, A, H	F, H, A
[ma]	[mi]	[mu]	[me]	[mo]
F, C", A	F, H, D"	F, D", H	G, H, C"	G, C", H
[ya]	,	[yu]	.	[yo]
F, C", A	F, H, D"	F, D", H	G, H, C"	G, C", H
[ra]	[ri]	[ru]	[re]	[ro]
G, D", H	G, C", E"	G, E", C"	A, C", D"	A, D", C"
[wa]	[wo]	[n]	!	?
G, D", H	G, C", E"	G, E", C"	A, C", D"	A, D", C"
Enter	Space	Back	**MS**	**IC**
A, E", C"	A, D", F"	A, F", D"	H, D", E"	H, E", D"

MS : Mode Shift

IC : Input Conversion

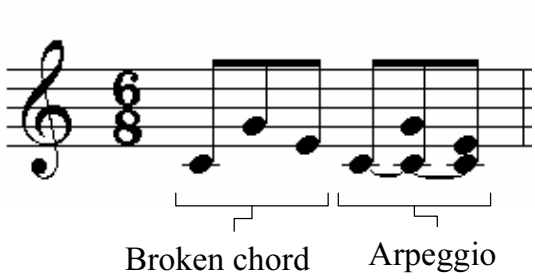


Fig. 3. An example of chords

Figure 2.

Accompaniment method

Table 2 (c) shows the character-code chart in Accompaniment method. As space is limited, we show only the chart for kana characters. Each character has the

corresponding chord as shown under the character in the table. This method uses two kinds of chords which are named *broken chord* and *arpeggio* shown in Figure 3 to specify the input character. The underlined chords in Table 2 (c) must be played as arpeggio. Additionally, the double quotation mark means the note up an octave. Users can input kana characters easily, because the forms shown in Table 2 (c) correspond to the vowel.

Charactristics of the text input methods
Each text input method has own characteristics, and performers can use multiple text input methods simultaneously. The combination of text input methods enables to use various musical expressions. Although Monophony and Polyphony method lack the artistry because the output sounds are fixed to the input character, Accompaniment method dose not cause any discordance because the method composes of only consonance.

On the other hand, Accompaniment method needs more Text-input keys than the other methods. Moreover, since this method uses only white keys, the tonality of the playing music is restricted to only the C major.

2.2 Musical artistry

To dissolve the lack of artistry in output sounds for Monophony method and Polyphony method, our system translates the output sounds of Text-input keys according to output sounds in Execution areas. Concretely, the system extracts note-numbers that are harmonious with note-numbers played in Execution area, and it replaces note-numbers of Text-input keys by the extracted note-numbers according to the proposed algorithm.

Replacement methods of note names

1. **Random method**
   When a Text-input key is pressed, the system randomly picks out and uses one note name played in Execution areas. For example, in the situation as shown in Figure 4, when C3, E3, and G3 in Execution area are played, the system randomly allocates one of them for Text-input keys.
2. **Direct allocation method**
   The system extracts note names played in Execution areas. Then, the system allocates the extracted note names in order for Text-input keys. For example, in the situation as shown in Figure 5, when C3, E3, and G3 in Execution area are played, the system allocates C, E, G in order for Text-input keys as shown in Figure 5.
3. **Chord extraction method**
   The system extracts possible chords that include note names played in Execution areas and then it selects one chord from them randomly. Next, it replaces the note name of Text-input keys by constituents of the extracted chord in order of the key. For example, in the situation as shown in Figure 6,

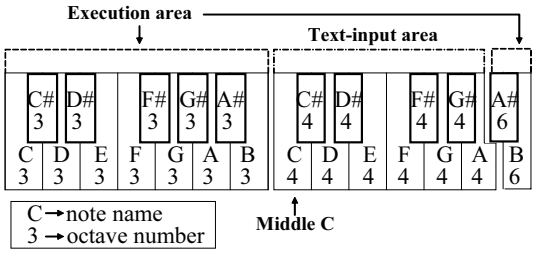


Fig. 4. An example of assigned note-numbers

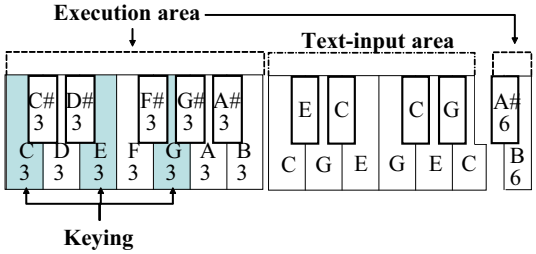


Fig. 5. The example of Direct allocation method

when C3, E3, and G3 in Execution area are played, the possible chords extracted by the system are C (C major triad), CMaj7 (C major seventh), C7 (C dominant seventh), C6 (C major sixth), and the system randomly selects one of them. Finally, the system allocates the note names of the selected chord in order for Text input keys as shown in Figure 6.

In Random method, since the system replaces the note name of the key whenever a Text-input key is played, the user enjoys unpredictable sounds. On the other hand, he cannot control output sounds. In Direct allocation method, if the note names played in Execution area is same, all keys in Text-input area are assigned the same note names. Therefore, users can control output sounds, at the same time, performances tend to be monotonous. Moreover, in Random and Direct allocation method, in the case that the number of keys pressed in Execution areas is one or two, the system reluctantly allocates the same note name for each Text-input key. On the other hand, Chord extraction method extracts possible chords to resolve this problem. This method has the most unpredictable quality in the proposed three methods, because the system may replace the note names of Text-input keys by the note names not played in Execution areas.

The methods of note names have advantage and disadvantage. Therefore, users can configure avairable methods on the keyboard.

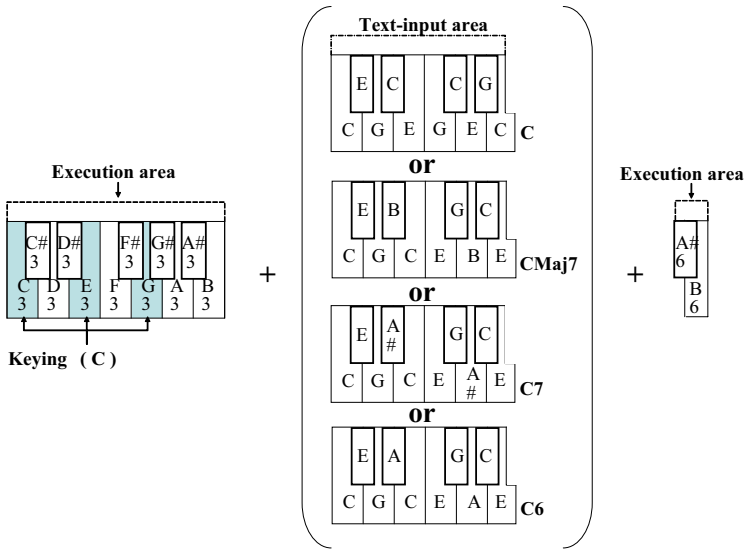


Fig. 6. The example of Chord extraction method

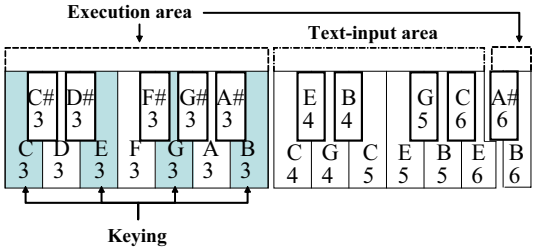


Fig. 7. The example of the note replacement algorithm

Replacement methods of octave numbers

In the general keyboards, the righter key has the higher pitch. According to this rule, the system allocates the octave number as shown in Figure 7.

In concrete terms, the system allocates the octave numbers that the leftmost/rightmost key in the Text-input area is higher/lower than the rightmost/leftmost key in the left/right side Execution area. For example, when the rightmost key in the left side Execution area is “B3”, and the leftmost key in the right side Execution area is “A#6”, the keys in a Text-input area are as shown in Figure 7.

Fig. 8. A snapshot of TEMPEST

3 Implementation

We implemented a prototype of TEMPEST using Microsoft Visual C++ 6.0 on Windows XP. Figure 8 shows a prototype and a player.

For example of using our system, Figure 9 shows a piece to enter "apple" artistically using Monophony method. The figure includes two scores; the upper one shows the case of not applying the note replacement algorithm, and the bottom one shows the case of applying the algorithm. In this example, we use a two octave keyboard, the area setting, note names as shown in Figure 4. Using these scores, finally "apple" phrase is entered to the PC. Obviously the upper score makes unmusical sounds including many inharmonic tones. On the other hand, using Direct allocation method as shown in the bottom score, it makes harmonic tones.

The first author of this paper has entered many texts using TEMPEST. He has been played the piano for twenty years over. He did not have uncomfortable feeling from misalignment between keying position and its output sounds. Moreover, after he mastered the proposed text input methods, he could concentrate on his execution and could control expressions such as dynamics, articulations, and agogics.

4 Evaluation and Consideration

4.1 Evaluation

We have evaluated the effectiveness of the proposed algorithms by the subjective evaluation. In this evaluation, we use IBM Think Pad X30 whose platform is Windows XP as a PC, SC-8820 of Roland as a MIDI sound generator, MM-1 of Bose as a speaker, OXYGEN8 as the keyboard and we project entered text onto a 40-inches screen. The performer entered "apple" phrase according to the piece of Figure 9. Examinees heard and watched the demonstration, and they compared the case of applying algorithm with the case of not applying it.

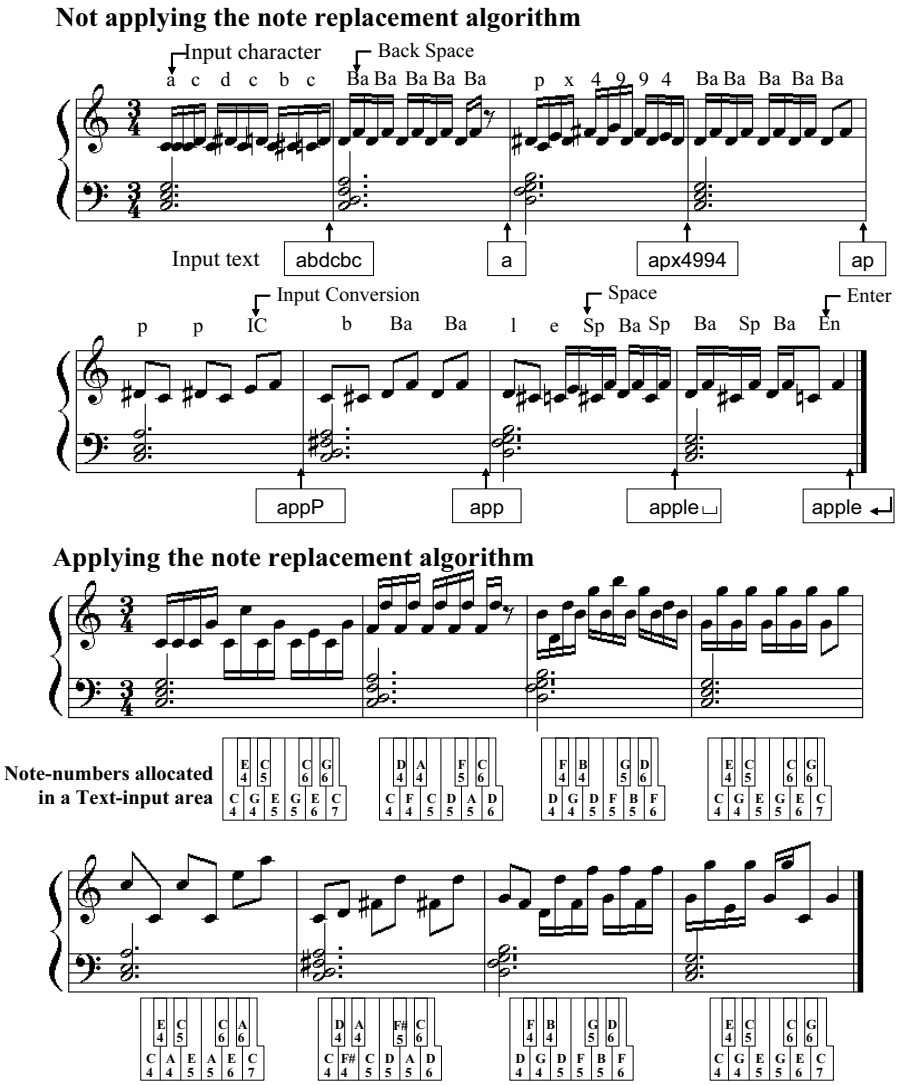


Fig. 9. The pieces to enter "apple"

48 examinees evaluated the system by ranking (1: worst, 5: best) the questions as shown in Table 3. The table shows the aggregate statistics of the questionnaires. Note that they did not answer Questions 3 in the play not applied the algorithm.

From the result of Question 1-2, we prove the effectiveness of proposed algorithm for the improvement of the artistry. From the result of Question 3,

Table 3. Aggregate statistics of the questionnaire

Question	Not Applying	Applying
1. This performance has musical artistry.	2.12	4.35
2. I am impressed by input characters with musical artistry.	1.56	3.83
3. I am impressed by the total performance.	–	4.17

TEMPEST has enough capability for performance art and an entertainment system. The most interesting point is that the input of useless characters and the deletion of them are effective to show the artistry for examinees.

4.2 Related work

There are OBOE[1] and Twiddler[3] that are interfaces inspired musical instruments. OBOE is like an oboe and a user can enter text fastly by using Braille input method. Twiddler is a character input device for wearable computing, which inputs a character like a guitar playing. These interfaces are designed to input text with high speed and high accuracy. On the other hand, our study aims to input texts as musical performances.

CiP[5] is a system for helping performers to freely demonstrate their musical expressions. Even if a user plays a wrong note, CiP always outputs the correct sound according to the playing piece. Therefore, the user concentrates on musical expressions such as the keying timing and the velocity. However, users cannot apply their technical skill to other applications by using CiP.

5 Conclusions

In this study, we have constructed TEMPEST that a user can input texts with musical artistry. We propose the several text input methods to improve the artistry of the text input, and evaluate the effectiveness of them.

In future, we have a plan to evaluate our system from the point of player's view. Moreover, we will construct text input interfaces using other instruments such as the guitar, the violin, and the maraca. Additionally, we will construct a musical performance chat system as an application.

Acknowledgments

This research was supported in part by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment."

References

1. T. Amemiya, K. Hirota, and M. Hirose. Oboe: Oboe-like braille interface for outdoor environment. pages 498–505. ICCHP 2004, July 2004.
2. Y. Itoh, A. Miyajima, and T. Wanatabe:. 'tsunagari' communication: fostering a feeling of connection between family members. pages 810–811. CHI 2002, April 2002.
3. K. Lyons, D. Plaisted, and T. Starner:. Expert typing using the twiddler one handed chording text entry on the twiddler one-handed keyboard. pages 94–101. ISWC 2004, October 2004.
4. E. Mynatt, J. Rowan, S. Graighill, and A. Jacobs. Digital family portraits: providing peace of mind for extended family members. pages 333–340. CHI 2001, April 2001.
5. C. Oshima, K. Nishimoto, Y. Miyagawa, and T. Shirosaki. Coloring-in piano: Indiscrete musical elements are essential for performers. pages 21–23. ICAD 2002 Rencon Workshop - Performance Rendering Systems, July 2002.
6. T. Yoshino, T. Muta, and J. Munemori. Namba: location-aware collaboration system for shopping and meeting. *IEEE Trans. on Consumer Electronics*, 48(3):470–477, August 2002.