PianoText: Redesigning the Piano Keyboard for Text Entry

Anna Maria Feit[1]
afeit@mpi-inf.mpg.de

Antti Oulasvirta[1,2]
oantti@mpi-inf.mpg.de

[1]Max Planck Institute for Informatics & Saarland University
[2]Aalto University, School of Electrical Engineering

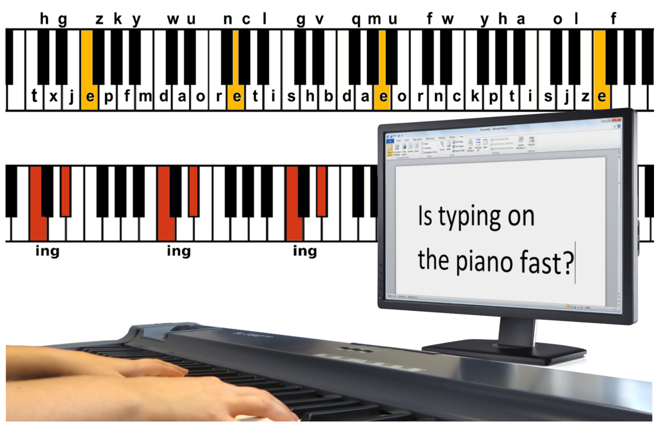

Figure 1: PianoText allows text entry on any MIDI capable piano. A computationally designed mapping offers multiple keys for one letter (yellow) and assigns chords (red) to whole letter sequences in English.

DIS '14, Jun 21-25 2014, Vancouver, BC, Canada ACM
978-1-4503-2903-3/14/06. http://dx.doi.org/10.1145/2598784.2602800

Abstract
Inspired by the high keying rates of skilled pianists we study the design of piano keyboards for rapid text entry. We present *PianoText*, a computationally designed mapping that assigns letter sequences of English to frequent note transitions in music. The design is based on four concepts: 1) redundancy, 2) chords, 3) sound and 4) skill transfer. It allows fast text entry of over 80 wpm on any MIDI enabled keyboard. At the demonstration, visitors can explore the benefits of these concepts by typing on *PianoText–Mini*, a device that allows for piano-based typing at a portable form factor.

Author Keywords
Text entry; the piano keyboard.

ACM Classification Keywords
H.5.2. [User-Interfaces]: Input devices and strategies

Introduction
We present *PianoText*, a novel text entry method for MIDI-capable piano keyboards (Figure 1). In its design we exploit the motor skill of pianists and combine features of music and text entry research. *PianoText* allows to enter text with a typing rate of expert Qwerty typists requiring only a fraction of the training time.

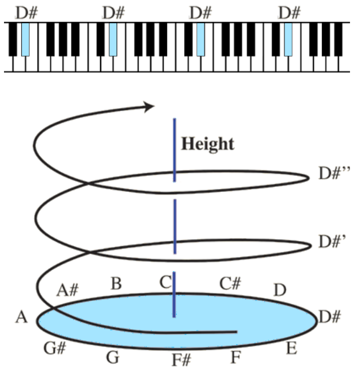

Figure 2: The 12 pitch classes repeat circularly with increasing tone height over the octaves of the piano. This gave inspiration to the concept of redundancy. The linear and circular dimension of pitch can be illustrated as a helix [3].

Underlying our research is a marvel at the performance of skilled pianists and the desire to comprehend if such capacity could be matched in a text entry task. For example, a pianist playing the Flight of the Bumblebee at regular speed enters 17 notes per second, which would translate to an astonishing 204 words per minute (wpm). PianoText is essentially a *mapping* that translates key presses on the piano to letters and letter sequences (n-grams) in the English language. Its design is based on four main concepts:

1. **Redundancy**: leveraging the large number of keys to assign one letter to many piano keys.
2. **Chords**: pressing 2 or more keys at once to enter frequent and long n-grams with only one stroke.
3. **Sound**: utilizing the auditory feedback of the instrument that identifies the pressed key .
4. **Skill transfer:** respecting frequently played note transitions in piano music, assuming that those are familiar to pianists and thus quick to type.

Their combination in one text entry method makes this device unique and, although learnable by anyone, allows pianists to adopt the method much quicker.

The demonstration seeks to convey the benefits of these concepts and their relevance to the design of other interactive systems. Visitors can type on PianoText–Mini, which is of a smaller form factor comparable to a regular Qwerty keyboard. nevertheless allows for similar input rates after training [4]. They will be introduced to the use of chords and redundant keys and can experience the rich sound feedback of the device. Audience members experienced in piano playing can make use of their musical skill and transcribe text presented in form of sheet music.

Design Concepts: from Music to PianoText

The four concepts underlying the design of PianoText result from the analysis of the piano keyboard as an input device, done in [4]. In the following we discuss their musical background, their implementation in PianoText and their exploration in other systems.

Redundancy:
Redundancy describes the fact that different input actions lead to the same result. It is inspired by the periodicity of the pitch classes over the octaves of the piano (see e.g. [3] and Figure 2). We implement this in PianoText by mapping one letter to several keys (see Figure 1). This creates "islands" of key clusters where frequent n-grams are close together. It shortens the hand-travel distance and provides more opportunities for hand alternation. Replicating structures is commonly used in the design of user interfaces (e.g. saving a document via toolbar or menu). However, only few text entry devices make use of this concept, such as the KALQ [6], a two-thumb keyboard which offers two buttons to enter a space.

Chords:
Pressing multiple keys at once to produce simultaneously sounding tones is a key concept in piano music. It forms the harmonic basis up on which the melody can be built. Chords are highly practiced by pianists and can be executed very fast. Musical chords gave inspiration to many text entry devices as discussed in [2] and more recently to chording gestures on multi-touch displays [5]. They are often used to account for a small number of keys. However, we utilize chords as a shortcut for frequent n-grams, similar to shorthand machines like stenographs. Following the periodicity of music, we replicate chords in every octave, minimizing the hand-travel distance for pressing a chord to max. 5 keys (see Figure 1).

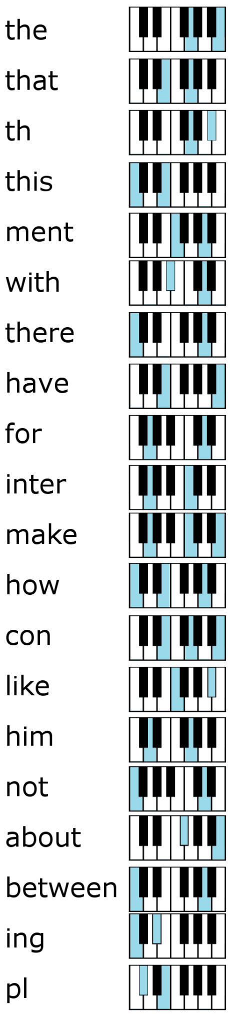

Figure 3: An extract of the chord mapping. PianoText implements 57 chords covering 34% of the English language.

Sound:
While the pianist relies on the sound feedback, Qwerty typists have much simpler auditory feedback of button presses. In contrast to the piano, the sound only serves as a confirmation that a key was pressed. However, this already suffices to improve performance in mobile text entry [1]. The complex musical feedback of the piano may further benefit training time, speed and accuracy.

Skill transfer:
In our design we want to leverage the motor skill of pianists. We make use of finger trajectories that are daily practiced by the pianist and common in music. we hypothesize that this will allow for higher input speed as well as shorten the learning process. Thus, for computing a mapping we assign common events in music to frequent n-grams in English, based on the frequency distributions of 1- and 2-grams.

Mapping Approach and Outcome

The construction of the mapping follows previous keyboard optimization methods to address the NP-complete problem of assigning letters to key slots [8]. We reformulate the problem for a 1D array instead of a 2D grid. Unfortunately, there are no quantitative predictive models for aimed movements in piano playing. Instead we build on the hypothesis that the most frequently encountered notes and note transitions will also be the fastest to respond. Therefore we extend our objective function to consider *two* frequency distributions: music (source) and language (target). We implement a greedy algorithm with additional constraints that ensure that the mapping favors frequently played notes, avoids infrequent intervals and minimizes hand-travel distances. For a discussion of the algorithm and a more detailed description of the approach see [4].

PianoText
The mapping from letters to piano keys, as computed by the algorithm described in [4], can be seen in Figure 1. It utilizes 55 keys and maps most of the letters at least twice. In addition, it assigns 57 chords to frequent n-grams (Figure 3). They cover 34% of the English language and enter on average 3.6 characters. The space is mapped to the *pedal* of the piano, controlled by the foot. A further analysis of PianoText with respect to the design concepts can be found in [4]. A long-term study described in [4] shows the learnability of the mapping despite its apparent complexity. After 140 hours of training, a hobby pianist memorized the mapping and reached a text entry rate of 81 wpm in a regular text transcription task. This compares favorably to *expert* Qwerty typists, who need a training time of 200 hours to reach a typing rate of 50 wpm, as cited in [7].

PianoText–Mini
To understand how our findings scale down to the form factor required by present-day computing devices, we implement a reduced version. PianoText–Mini uses a 37-key piano which is inexpensive, lightweight and similar in size to a regular physical keyboard (Figure 4). The mapping is obtained by cutting PianoText at the 37th key. This contains the whole alphabet and covers 90.15% of key presses performed on the full PianoText [4]. The new design also implements error correction. N-grams are deleted in the same unit they were entered by pressing a designated 2-key-chord forming the interval of a small second. The dissonant sound distinguishes it from the other chords and underlines the consequence of the action. The implementation of error correction and the smaller form factor increases the applicability of piano-based typing, while allowing for similarly high input rates [4].

Demonstration Requirements

Space:	3 x 3 m, optimal
Equipment we bring:	1 laptop 1 37-key midi-piano 1 tablet tapes and wires as needed
Equipment needed:	2 tables for displays, laptop, keyboard 1 chair to sit when playing the piano 2 large displays to show entered text and videos (24") 1 set of speakers Power cord for ca. 6 plugs Poster stand
Power for:	1 laptop 2 Displays 1 set of speakers 1 midi-piano 1 tablet charger

Table 1: Equipment needed to set up the demonstration. Description of what we will bring and what would have to be provided.

Demonstration

Visitors will explore the design concepts and get a hands-on experience on piano-based typing:

- *Non-pianists:* Visitors can try entering text on the labeled keys of PianoText–Mini.
- *Pianists:* Audience members skilled in piano playing can make immediate use of their expertise. We will provide music sheets from which they can enter sentences translated into notes.
- *Design concepts:* Visitors will be introduced to the use of chords and redundancy and can experience the effect of a complex sound feedback.
- *Expert performance:* We present videos of a trained piano-typist entering sentences with a speed of 70 to over 100 wpm.

Timing and Visitor Capacity

The demonstration allows for one visitor at a time to type on the piano, while waiting people will be shown videos of a trained piano typist and explained the design concepts by aid of a poster. There are no limits on time or the number of people visiting the demonstration.

Set-up and Requirements

Requirements are listed in Table 1. The demonstration does not require any special location concerning lighting or acoustics, however, it should be ensured that surrounding demonstrations are not disturbed by the piano sound.

Impact

We believe, that the exploited design concepts are highly relevant to the design of interactive systems in general. Our demonstration will introduce them to the visitors and inspire them to further investigate the application of redundancy, chords and sound feedback, and explore the exciting concept of skill transfer in other domains as well.

Figure 4: PianoText–MINI is implemented on a 37 key Midi-piano, similar in size to a physical Qwerty keyboard.

References

[1] Brewster, S. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing 6*, 3 (2002), 188–205.

[2] Buxton, B. Chord keyboards, 2012. `www.billbuxton.com/input06.ChordKeyboards.pdf`.

[3] Deutsch, D. Pitch circularity. `http://deutsch.ucsd.edu/psychology/pages.php?i=213`.

[4] Feit, A. M., and Oulasvirta, A. Pianotext: Redesigning the piano keyboard for text entry. In *Proc. DIS'14*, ACM (2014).

[5] Lepinski, G. J., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. In *Proc. CHI'10*, 2233–2242.

[6] Oulasvirta, e. a. Improving two-thumb text entry on touchscreen devices. In *Proc. of CHI'13*, 2765–2774.

[7] Schneider, W. Training high-performance skills: Fallacies and guidelines. *Human Factors 27*, 3.

[8] Zhai, S., Hunter, M., and Smith, B. Performance optimization of virtual keyboards. *Human–Computer Interaction 17*, 2-3 (2002), 229–269.