

TypeSinger: 歌うように“弾く”歌詞タイピングゲーム

内田 千尋^{1,a)} 橋田 光代^{1,b)}

概要: 音楽の歌詞を入力題材としたタッチタイピングゲームは、文字入力を演奏に合わせることで自体に楽しさがある。しかし、音楽演奏ゲームの一種として捉えた場合、入力文字の正確さや速度を追求するだけで音楽的要素について考慮されず、演奏とタイピング入力との間に聴覚的な関連がない。本研究では、タイピングを楽器演奏の一種とみなし、歌唱の発音と同じタイミングで各文字を入力させることで、カラオケのようにリズムカルなタイピングスキルを身につけられる歌詞タイピングゲームを提案する。

TypeSinger: A Lyrics Typing Game with ‘Playing’ Melody

UCHIDA CHIHIRO^{1,a)} HASHIDA MITSUYO^{1,b)}

Abstract: There are lots of touch-typing games which types characters of lyrics along with music playing. However, those games require a players only to pursue correctness and speed of typing and don't considered with typing rhythm as a music performance. In this paper, we describes a musical typing game, which considered with rhythm of a melody on typing characters of lyrics.

1. はじめに

タイピングゲームの主たる目的は、文字を正確かつ素早く入力することである。出題例として人気が高い題材のひとつに音楽の歌詞がある。歌詞入力によるタイピングゲームは、その楽曲の演奏を聴きながらタイピングすることで、プレイヤーのモチベーションを上げつつ、タイピングスキルの向上と、好きな楽曲の歌詞を覚えるという2つのメリットを享受している。

しかし従来の歌詞タイピングゲームは、実際は単に文節レベルの長さのフレーズ歌詞を入力するだけであり、入力行為そのもののリズムについては考慮されていない。すなわち、演奏に合わせるといっても事実上は演奏を無視した高速文字入力が競われており、音楽を採り入れた意義があまりないのが現状である。

本研究では、歌詞タイピングを音楽演奏に関連づけ、正確かつ高速な文字入力を目指すと同時に、音楽演奏としてのタイピングを考慮したカラオケ感覚の歌詞タイピング

ゲームを提案する。

2. 関連研究

2.1 音楽に合わせたタイピングゲーム

音楽演奏に沿ってタイピングスコアを競うタイプのゲームは、プログラミング学習の題材としても人気が高く、すでに多数存在する。

SongTyping や Vocaloida^{*1}, TypingTube[1], TYPING-MANIA[2] などがあり、楽譜や演奏動画に対して歌詞情報を独自に付与し、再生に合わせて歌詞を入力する。このタイプのゲームでは、あくまで歌詞の文字判定が重視されており、音楽演奏のリズムそのものは採点対象とならない。

一方、typebeats[3] のように、いわゆる「音ゲー」とタイピングを組み合わせたものは、演奏のリズムに合わせたキー入力がメインとなる。しかしこの場合は、入力時刻においてキー入力がひとつだけとなる。すなわち、単語入力よりも一字ずつのランダムなアルファベット入力が主流であり、歌詞について考慮されないという状況である。これでは、「音ゲー」の代表格ともいえる beatmania[4] や太鼓

¹ 相愛大学音楽学部 音楽マネジメント学科
Soai University, Chuo-ku, Osaka 541-0053, Japan

a) M1211004@s.soai.ac.jp

b) hashida@soai.ac.jp

^{*1} SongTyping, Vocaloida とともに現在はサイトが閉鎖されている。SongTyping については <https://youtu.be/Il10VH6uD-Q> などプレイ動画が確認できる。

の達人 [5] などと本質的な面で相違がない。

音楽演奏，しかも歌唱を題材にしたタイピングゲームとしては，歌詞入力と演奏リズムの双方を同時に競えるシステムデザインが望ましい．この観点においては，竹本らの MikuType[6] がその理想形のひとつに数えられる．ただし，MikuType では PC キーボード入力に加えて，鼻歌による微細なピッチ操作まで可能としている．初音ミクを題材として，人間性の高いリアルタイム歌唱の実現を目指しているため，音楽的にタイピングスキルの習得を目指す本研究とは趣旨が異なる．

2.2 語学学習としての歌詞タイピング

語学学習という観点において，音楽は，そもそもの言語習得の動機であったり，学習意欲を高める題材であったりする．対象言語で書かれた歌の歌詞を覚えるという学習法がその代表例であり，(a) 好きな映画やドラマのセリフをシャドウイングしたり，(b) 好きな音楽の歌詞を覚えて原語で歌うなどが実施されている．いずれも，積極的に発話を促進し，身体的，聴覚的側面から言語習得を促進するものである．

Lyrics Training[7] は，歌唱曲を聴きながら歌詞中の一部の単語を聴き取り，その単語が歌われるまでに入力して演奏とタイピングを進めるリスニング学習システムである．聴取対象となる単語は歌詞の中からランダムに抽出されるが，その抽出率が，プレイモードによって4段階にレベルデザインされている．音符ひとつに文字ひとつが基本の日本語曲に比べ，英語を始めとする外国語曲においては，演奏速度に合わせてすべての歌詞をタイピングするのは極めて難易度が高い．そのことにうまく対処した例である．

歌詞に対する単語抽出は，別の見方をすると「難しい楽曲において一部の歌詞を間引く」という操作である．音楽演奏においても，原曲で細かい音符が連続する速いパッセージ（音符列）が含まれる場合，初級者向けの楽譜には，音符数を減らしてリズムを単純にするという編曲が行われている．本研究では今後の課題とするが，歌詞タイピングのレベルデザインを考察するうえで，Lyrics Training の手法は参考にしたい．

3. 音楽演奏としてのタイピング

第3章では，タイピング行為を楽器演奏の一種と位置づけ，より音楽的な側面から自然なタイピングスキルを習得したり楽しめるゲームデザインについて述べる．

タイピングが直接的に音楽と紐付いた事例としては，R. アンダーソンの作曲による「タイプライター」[8]がある．本物のタイプライターを用いて，その入力音をリズムに沿って鳴らしていくというユニークな楽曲である．カタカタという文字キーの入力音だけでなく，改行時のレバー操作音，印刷部のシリンダーが右にある程度進むと鳴るベル

音が効果的に用いている．この作品に見られるように，タイプライターは (1) 継続的な文章入力の表現と，(2) 文章入力に伴う複数の音色の表現を同時に操作可能であることがわかる．

そこで，本研究でも，通常の文章入力に伴う特殊キーの操作も，歌詞演奏の一環として導入し，より音楽的な歌詞タイピングができる仕組みを構築する．

3.1 歌詞入力

本システムに対応する楽曲は，主に日本語と英語の歌詞を想定する．本稿では日本語楽曲について説明する．

表示される歌詞は漢字を交えた文章表記であるが，外国語の歌詞入力にも対応できるよう，日本語入力にはローマ字入力による仮名を用いる．仮名のローマ字表記には，訓令式，ヘボン式を筆頭とする複数の方式が存在するが，どれをどのように用いるかは個人差がある．

文字の入力方式は，楽器演奏における運指法に該当する．王道に器楽学習を目指すのであれば，運指法はある程度遵守すべきものであるが，一方で「演奏自体を楽しむ」ことを重視すると，運指の正確さは必ずしも厳格でなくとも良くなる．この考え方に倣い，本システムの文字入力に関しては，1文字に対する複数の入力方式を許容し，多くのプレイヤーがスムーズにタイピングできるようにする．

3.2 歌としての歌詞のリズム

従来の歌詞タイピングゲームの大きな特徴として，以下の2点がある．

(1) 誤字を打つと判定が中断し，正しい入力が入るまで次の文字に進むことができない．この仕様のまま音楽演奏が導入されると，誤字によって入力できなくなるにも関わらず，演奏は先に進んでしまい，その差がどんどん開いてしまうことになる．その対策として，Lyrics Training[7] は，誤字によって音楽を消音し，再開可能な状態になると，中断時刻の少し前から再び演奏を始めている．

(2) 音楽演奏を伴うにも関わらず，タイピング自体はメロディの進行に関係なく入力可能である．この仕様は，(1)のように，文字入力に遅延が生じた場合に，以降の入力速度を上げて「追いつく」といった対策を取るのに都合が良い．一方で，プレイヤーのタイピングスキルが十分に高い場合，演奏を無視してただ高速に文字入力を進めることも可能となる．

本研究においては，音楽演奏に沿って歌詞を進めるというのは，カラオケ歌唱と同等であると考えられる．カラオケのシステムには，多くの場合，共通して次の3点の特徴がある．

(1) 採点機能について，歌詞自体の成否や発音の正しさ，滑舌などは採点基準に含まれない．すなわち，子音と母音のセットからなるローマ字仮名表記において，子

音の正確さを考慮しないのと同義と考える。

- (2) 音高やリズムを間違えた際も、演奏は中断することなく最後まで再生される。このことは、楽器演奏を上手にこなすための必須事項のひとつに数えられる「途中で弾き間違えても中断せず進み続けること」と対応する。
- (3) プレイヤーの歌唱スキルや感覚に原曲が合わない場合、テンポや調を自由に変更できる。このような操作は楽器演奏においても見られる。すなわち、難しい曲に挑戦する際にいきなりオリジナルの速さで演奏できるとは限らない。そのような場合、まずはテンポを落としてゆっくり弾くことから始めて、慣れたら本来の速さに戻していくという練習プロセスが実施される。そこで、音楽演奏に合わせることに重点を置くために、歌詞入力については「後戻り禁止」とし、誤字は無視して次の文字を入力できるようにする。

3.3 フレーズの区切り

通常の文章入力において、Enter、Space、Backspace (Macの場合 delete) の各キーは、改行・漢字変換の確定 (Enter)、漢字変換・空白文字 (Space)、誤字入力時の一字削除 (Backspace) など、文字入力を円滑に進める役割を担う。特殊操作であるため、従来の歌詞タイピングでは無視されるキー群である。

しかし、これらの補助入力は、単語や文節単位で文章を確定させ、次の文章に意識を向けるという重要な「区切り」の役割をも担っている。この役割は、音楽演奏でいうならば息継ぎや合いの手に相当するであろう。息継ぎや合いの手は、フレーズの強弱や抑揚を効果的に制御し、音楽全体のノリを良くするための上級テクニックとして工夫が必要なものである。つまり、より質の高い演奏を目指すならば、これらはいつでも好き勝手に入れるのではなく、入れるタイミングをよく見計らう必要がある。

本研究では、これら入力補助キーについても歌詞タイピングの一環とみなし、主に合いの手のサウンドに変換することでゲーム要素に組み込む。

4. システム概要

TypeSinger は Max 6 で動作するデスクトップアプリケーションとして実装されている。システムの概要を図 1 に示す。

4.1 基本操作

楽曲を選択し、再生ボタンを押すと演奏が開始される。画面上には、メロディ譜と歌詞が表示され、演奏に沿って入力バーが楽譜上をスクロールする。プレイヤーは、演奏を聴きながら、入力バーが音符に乗るタイミングに合わせて該当歌詞を逐次入力する。入力を始めた段階で随時スコ



図 1 TypeSinger の操作

アが更新、表示される。

4.2 入力キー

入力可能なキーとして、下記の 2 種類を用いる。

- 1) アルファベット (a-z) 歌詞入力。日本語の場合はローマ字仮名入力とする。入力方式については、3.1 節で述べた通り、1 文字について複数の入力方法に対応する。

- 2) Enter, space, Backspace 「合いの手」のサウンドを鳴らす (3 種)。演奏中はいつでも入力できる。上級者モード (後述) でのみ採点対象とする。

その他のキーについては、歌詞入力および採点には影響しない。

4.3 レベル設定

歌詞タイピングの難易度は、(1) メロディーに含まれる音符数、すなわち、フレーズ中の歌詞の文字数と、(2) 楽曲の再生速度に依存する。

楽器演奏の場合、プレイヤーが弾きこなせない難しいメロディーに対しては、テンポを遅くしてゆっくり弾くという練習方法が採られる。一部のフレーズのみが弾けない場合で全体の通し演奏をする際も、作曲上、あるいは演奏表現としての意図と無関係に演奏途中でテンポを急激に変動するのは好ましくない。

そこで、全体的に原曲の速さでの歌詞入力が困難な場合は、プレイヤーが自分の意思で演奏速度を指定させることで、ほぼ一定の演奏テンポでリアルタイム入力が可能なレベルに演奏を合わせるようにする。

4.4 演奏データ

題材となる楽曲の演奏データには WAV、AIFF 形式の音響信号ファイルを用いる。ユーザ設定によりテンポ補正がかかる場合、楽曲全体の音響信号に対して演奏時間伸縮と周波数シフトを組み合わせたタイムストレッチ処理をかけることで、原曲のキーを維持した上でのテンポ変更に柔軟に対応する。

4.5 採点方法

タイピングゲームにおける採点は、基本的に、課題となる文字数における入力文字の正答率やその入力速度に基づいて算出される。しかし、音楽演奏に合わせた歌詞タイピングの場合は、文字入力のみを採点対象にすると、演奏の進行を無視して高速に歌詞入力をしても高得点を得ることができてしまい、「演奏に合わせたタイピング」にならない。そこで、メロディに沿ったリズムカルな入力要素についても採点対象とすることで、プレイヤーにタイピングと演奏リズムの両方を同時に意識させることを狙う。

TypeSinger におけるユーザ入力の正否判定には、初級者モードと上級者モードの二つがあり、以下の要領で実施される。

歌詞の文字判定 入力方式に関係なく、一文字ずつに対する成否判定を行い、正しければ加点する。

歌詞のリズム判定 母音の入力タイミングが、メロディの拍節の範囲内に収まったかどうかで、Best, Good, OK, Bad の4段階の点数を加算する。

「合いの手」のリズム判定 上級者モードでのみ採点対象とする。Enter, space, Backspace の打鍵について、拍節に沿った入力ができただどうかを判定する。拍節の解像度については、楽曲のテンポにもよるが、最大でBPM=60における三十二分音符（およそ125ミリ秒間隔）相当までとする。

4.6 スコア算出

演奏および歌詞入力が始まると、以下の方式に沿って、演奏時刻 t までの文字正答率 $S_L(t)$ 、歌詞リズム正答率 $S_R(t)$ 、合いの手のリズム正答率 $S_C(t)$ が、採点モード M に沿って合算され、それにテンポ補正率 W_T がをかけて100点満点に正規化されたスコア S_t がリアルタイムに算出されていく。

$$S_t = \left(\frac{S_L(t) + S_R(t) + S_C(t) \times M}{2 + M} \right) \times W_T \times 100$$
$$S_L(t) = \left(\frac{\text{正答字数 } L_i(t)}{\text{楽曲全体の文字数 } L_s(t)} \right)$$
$$S_R(t) = \left(\frac{\text{正答音符数 } M_i(t)}{\text{楽曲全体のメロディ音符数 } M_s(t)} \right)$$
$$S_C(t) = \left(\frac{\text{拍節正解数 } B_i(t)}{\text{合いの手の入力数 } B_s(t)} \right)$$
$$M = \begin{cases} 0 & (\text{初級者モード}) \\ 1 & (\text{上級者モード}) \end{cases}$$

ここでは、選択した楽曲のオリジナルの速さで歌唱できれば十分であるという考えから、原曲通りの速さで完全な歌詞入力できれば100点満点となるようにする。なお、原曲よりテンポを速めてのセッションを行った場合に限り、 W_T が100%を超えるため、最高点が100点を超える可能性があるが、それについては特殊扱いとする。

5. まとめ

本稿では、音楽演奏を題材とし、歌詞入力と演奏リズムの双方を考慮したカラオケ感覚の歌詞タイピングゲーム TypeSinger について述べた。誤字を打っても演奏に合わせて「歌い」続けられること、しかし演奏リズムに乗って正確なタイピングを目指す必要があることに加え、上級者向けに、「合いの手」を加点対象とすることで、ある程度のタイピングスキルがあるプレイヤーを対象としたレベルデザインを構築することができた。

今後は、タッチタイピングの初級者にもプレイ可能なレベルデザインについて検討する。タイピング習得に必須となる文字の位置を覚える機構や、2.2節で述べたような「歌詞を間引く」機構などを検証した上で、より多くのプレイヤーが音楽的なタイピング練習ができるシステムの構築を目指す予定である。

参考文献

- [1] Typing Tube, <http://typing-tube.net/>.
- [2] TYPINGMANIA, <http://www.sightseekerstudio.com/typingmania/>.
- [3] typebeats cloud, <http://cloud.typebeats.com/>.
- [4] コナミデジタルエンタテインメント: beatmania, <http://www.konami.jp/am/bm/>.
- [5] パンダイナムコエンターテインメント: 太鼓の達人, <http://taiko.namco-ch.net/taiko/>.
- [6] 竹本拓真, 馬場 隆, 片寄晴弘: MIKUTYPE: リアルタイムで初音ミクに歌わせるタイプソングシステム, エンターテインメントコンピューティング(EC)2013, pp. 330-331 (2013).
- [7] Lyrics Training, <http://lyricstraining.com/>.
- [8] Anderson, R.: Typewriter, (1950).
演奏例: <https://youtu.be/nW8dGwa2zRw>