

フリック入力に基づく2本の押し込み機能付きジョイスティックを用いたテキスト入力手法

横山 海青^{*1} 高倉 礼^{*1} 志築 文太郎^{*1} 川口 一画^{*1}

Text Entry Method based on Flick Input using Two Joysticks with Push

Kaisei Yokoyama^{*2}, Rei Takakura^{*2}, Buntarou Shizuki^{*2}, Ikkaku Kawaguchi^{*2}

Abstract – A Japanese software keyboard for game consoles has a problem with the entry speed and widget size. In this article, we show JoyFlick that is a text entry method for an improved Japanese software keyboard. We evaluated the text entry speed and accuracy of JoyFlick. Also, we conducted an analysis of the user's operations to study the tendency of the operations and verify the design of JoyFlick. The results showed the following tendencies: the user tends to tilt the stick to select the keys on the outer edge; the user tends to restore the stick to the neutral position between selections; the user tends to restore both sticks to the neutral position at the same time ($M = 15.5 \times 10^{-2}s$) at the entry of a character. Also, the results showed that the design of JoyFlick, which restricts the order of consonant and vowel selection prevents a part of mistaken entries. Moreover, we show the outlook of the application and the necessity of study in actual and several environments.

Keywords : text entry, entertainment, game controller

1 はじめに

ゲームパッドのみを用いる文字入力手法(以降、ソフトウェアキーボード)を提供することにより、ゲーム機のユーザはハードウェアキーボードを別途調達およびセットアップすることなく、文字を入力できる。

Nintendo Switch および PlayStation 4 は、日本語のかな文字表を模したキー配置を持つソフトウェアキーボード(以下、50音キーボード)をユーザに提供しているが、このキーボードには入力にかかるユーザの負担の高さおよび画面占有率の問題がある。50音キーボードは、かな文字表を模した配列に沿って並べられた多数のかな文字キーおよび1つのカーソルを持つ。ユーザは文字を入力したのち、次に入力したい文字に対応するキーまでカーソルを移動させる。その際、カーソルを動かす距離は「直前に入力した文字」と「次に入力したい文字」の組み合わせによって決まる。カーソル移動の距離が大きくなるほど、「次に入力したい文字」を入力するために必要なユーザの負担は増加する。さらに、50音キーボードはかな文字表を模しているため、同時に表示されるキー数が多い。同時に表示されるキー数が多い場合、文字の見やすさを高めるためにキーのサイズを大きくするとウィジェットの画面占有率が大きく上昇する。また逆に、コンテンツを見せるためにウィジェットのサイズを小さくすると、キーの

サイズが大きく下降する。これらの問題は、ユーザ体験を損なうと同時に、ゲーム内の表現に制限をかけていると考えられる。

我々は、入力に必要な負担および画面占有率の低い文字入力手法を開発するため、2本のジョイスティックを用いた、フリック入力に基づくかな文字入力手法 JoyFlick^{[1],[2]} を作製した。本論文において、まず JoyFlick の設計を述べる。続いて、JoyFlick を用いたかな文字入力の速度および精度の評価実験について述べる。その後、JoyFlick を用いてかな文字入力を行うユーザの操作の分析について述べ、JoyFlick の設計の有用性を検証する。最後に、JoyFlick の実用性をさらに示すための調査について述べる。

2 関連研究

JoyFlick はフリック入力を始めとしたかな文字入力手法およびジョイスティックを用いた文字入力手法に関連している。また、JoyFlick と同様に、これら2つの手法の両方に関連する手法が提案されている。

2.1 かな文字入力手法

本節において、JoyFlick に関連のあるかな文字入力手法を述べ、これらの手法が JoyFlick の設計指針にどのように関連しているのかを説明する。

2.1.1 QWERTY キーボード

Nintendo Switch および PlayStation 4 など、現代の据え置き型のゲーム機の多くに対して、ユーザは QWERTY キーボードを用いてかな文字を入力でき

*1: 筑波大学

*2: University of Tsukuba

る。熟練したユーザによる QWERTY キーボードを用いた入力は高速である。一方、Nintendo Switch および PlayStation 4 などのゲーム機に QWERTY キーボードは付属していない。したがって、QWERTY キーボードを用いて文字入力するためには、ユーザはゲーム機とは別にこれを調達する必要がある。それに加え、ユーザは文字入力の前後に、QWERTY キーボードとゲームパッドを持ち変える必要がある。さらに、ユーザの姿勢に対して、キーボードに手が届くこと、という制限が加わる。これらのデメリットを避けるため、JoyFlick はゲームパッドのみを用いてかな文字を入力できるよう設計された。また、ゲーム機への入力においても QWERTY キーボードを使用できることは、QWERTY キーボードに慣れているユーザにとって、ゲーム機専用の文字入力手法に習熟する必要がないというメリットがある。しかし、日本人の若者の多くは QWERTY キーボードによる入力を苦手としている^[3]。これらの特徴を踏まえ、JoyFlick は習得が容易な手法となるよう設計された。

2.1.2 50音キーボード

50音キーボードは、ゲームパッドにおける一般的なかな文字入力手法である。Nintendo Switch および PlayStation 4 など、現代の据え置き型のゲーム機の多くが 50音キーボードによるかな文字入力をユーザに提供している。50音キーボードは、かな文字表のキーレイアウトを持つ選択キーボードである。入力できる文字がすべて画面に表示されており、また操作が単純なため、ユーザはひと目で使い方を理解できると考えられる。また、50音表に似た見目をしているため、ユーザは入力したい文字に対応するキーがどこにあるのかをすぐに把握できると考えられる。

一方、50音キーボードは 50 前後のキーを持ち、ユーザはこれらを 1 つのカーソルで扱わなければならない。入力する文字の並びによってはカーソルの移動距離が大きくなるため、ユーザの負担が増加する。この特徴を踏まえ、JoyFlick は入力にかかるユーザの負担が少なくなるよう設計された。さらに、50音キーボードは大量のキーを持つため、キーボードの見やすさを優先してキーの文字のサイズを大きくするとウィジェットが大きくなり、コンテンツを表示することを優先してウィジェットを小さくすると、キーの文字が小さくなる。これらの特徴は、ユーザ体験を損なう恐れがある。これを踏まえ、JoyFlick は同時に表示するキーの数を減らすことにより、ウィジェットが小さくなるよう設計された。

2.1.3 フリック入力

フリック入力は、タッチスクリーンを搭載したモバイル機器で広く用いられているかな文字入力手法である。日本人の若者の 80% がスマートフォンにおいてフ

リック入力を使用している^[4]。ユーザはまず、入力したい文字の子音に対応するキーを押して子音を選択し、次に入力したい文字の母音に対応する方向にフリックすることにより、母音の選択と文字の入力を行う。これを踏まえ、学習コストを減らすため、JoyFlick のキー配置はフリック入力に倣っている。

2.1.4 フリック入力に基づくかな文字入力手法

Blossom^{[5],[6]}、井川らの手法^[7] および No-Look Flick^[8] は、フリック入力に基づくかな文字入力手法である。これらの手法はタッチパネルを持つ端末向けに示されたが、JoyFlick はゲームパッド向けに設計された。

2.2 スティックを用いた文字入力手法

EdgeWrite^[9] は 1 ストロークのジェスチャを描くことにより、アルファベット 1 文字を入力する手法である。EdgeWrite はジョイスティックを含むさまざまな入力機器に対応している。EdgeWrite はジェスチャ入力の一種である一方、JoyFlick は選択キーボードの一種である。

PizzaText^[10] および TwoStick^[11] は、2 本のジョイスティックを用いたアルファベット入力手法である。ユーザは片方のスティックを用いて「文字のグループ」を選び、もう片方のスティックを用いて「文字のグループ」の中から「入力する文字」を選ぶ。また、キーの配列はアルファベット順に沿って決められている。JoyFlick も同様の手順に沿ってかな文字を入力する。一方、JoyFlick のキーの配列は、ユーザの多くが慣れているであろうフリック入力のキー配置に基づいて決められている。

Sandes ら^[12] は、QWERTY キーボードのユーザ向けに、QWERTY 配列に基づいたゲームパッド向けの入力手法を提案した。ユーザは左のジョイスティックが QWERTY キーボードの「D」と「F」の間に、右のジョイスティックが「J」と「K」の間にあると想像し、QWERTY キーボードにおいて入力したい文字に対応するキーが存在する方向へとジョイスティックを倒す。QWERTY キーボードの左半分に存在する文字は左スティックを、右半分に存在する文字は右スティックを用いて入力される。また、ユーザのジョイスティックへの操作は言語モデルを用いた処理によって文字へと変換される。JoyFlick もユーザの多くが慣れているであろうフリック入力に基づいて設計された。一方、JoyFlick のキー配置はキーと要素（母音あるいは子音）が一对一に対応しており、文字入力に際して言語モデルによる補正を必要としない。

また、JoyFlick はこれらの手法とは違い、かな文字入力への適用を念頭に設計された。

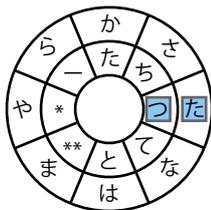


図1 EGConvertを用いて、ユーザが子音「た」および母音「う」を選択している様子。

2.3 ジョイスティックなどのポインティングデバイスを用いたかな文字入力手法

Goら^[13]は、EdgeWrite^[9]をカタカナの入力に適用する Katakana EdgeWrite を提案した。Katakana EdgeWrite は EdgeWrite と同様にジェスチャ入力的一种であり、一方、JoyFlick は選択キーボードの一種である。

中村らの手法^[14]は、2つのトラックボールマウスを用いた選択キーボードである。中村らの手法において、最適なキー配置の検討は今後の課題とされていた^[14]。一方、JoyFlick は学習コストを抑えるためにフリック入力に基づくキー配置を用いている。

今里の手法^[15]、EGConvert^[16]およびIToNe^[17]は、2本のジョイスティックを用いた選択キーボードである。特に、EGConvert^[16]およびIToNe^[17]は、ゲームパッド向けに設計されている。

今里の手法^[15]は、2本のジョイスティックを用いて、2度の操作で文字を入力できる手法である。ユーザはジョイスティックに対し、5種類の方向の入力を行える。ユーザは2本のジョイスティックにより提供される10種類(5種類の方向×2本のジョイスティック)の操作を2度行うことにより、文字を入力する。1度目は入力したい文字の子音などを選ぶための操作であり、2度目は母音などを選ぶための操作である。一方、JoyFlick は子音選択を右スティック、母音選択を左スティックに対応させている。

EGConvert^[16]のウィジェットには中心を共有する2つのリングが表示され、外側のリングが子音、内側のリングが母音にそれぞれ対応している(図1)。ユーザは左スティックを用いて子音を、右スティックを用いて母音をそれぞれ選択する。2つのリングを1つの円に収める配置は画面占有率を下げるが、ユーザの認知的な負担を増加させる可能性がある、とGoらは指摘している^[17]。

IToNe^[17]のウィジェットには2本のリングが存在し、それぞれのリングは操作状態に応じて上半分または下半分のみが表示される(図2aおよびb)。ユーザは、片方のスティックを上倒すことにより子音を選択し、もう片方のスティックを下倒すことにより母音を選択する(図2b)。最後に、どちらかのスティック

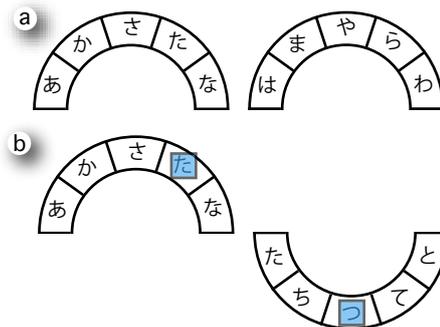


図2 IToNeのウィジェット。(a)無操作状態、(b)ユーザが子音「た」および母音「う」を選択しているとき。

をニュートラルポジションに戻すことにより、ユーザは文字を入力する。子音および母音の選択の順序は決まっておらず、ユーザに委ねられている。また、50音順に基づいた独自のキー配置を使用している。JoyFlickもIToNeと同様に、左右に分かれたウィジェットを持ち、左のリングは左のスティック、右のリングは右のスティックを用いて操作する。一方、JoyFlickのキー配置はスマートフォンのフリック入力に倣ったものを使用している。これはユーザの使い慣れたキー配置を用いることにより、学習コストを低下させるためである。また、IToNeのキー配置には重複が存在する(左右ともに、リングの下半分に同じキーが配置されている)が、JoyFlickのキー配置には重複がない。さらに、左右のスティックの押し込みと右スティックのニュートラルポジションにキーを配置したことにより、パイメニューのスライスの内角がIToNeよりも大きくなっている。これにより、ユーザに要求する操作の精度が低くなるため、ユーザにとって、慎重さよりも速度を優先することがより簡単になる。

3 JoyFlick

JoyFlick(図3)は押し込み機能付きのジョイスティック2つを用いる、フリック入力に基づくかな文字入力手法である^{[1],[2]}。関連研究とJoyFlickの比較を表1に示す。JoyFlickは以下の設計指針に沿って設計されている。

■高速なかな文字入力が可能であること

入力にかかる負担を減らし、高速な入力を実現することにより、頻繁な文字入力および長文の入力に対するユーザの負担が減少すると考えられる。これにより、ゲーム機における文字入力の更なる活用が容易になり、ユーザによる既存の文字入力を伴う機能のより頻繁な使用、ならびに開発者による文字入力を伴う新たな機能の実装が期待される。

表1 ゲーム機向けの既存のかな文字入力手法と JoyFlick の比較. 「キーボード」は QWERTY キーボードを指す. また, 本研究の長期実験にて記録された, 日時および個人ごとの入力速度のうち最も速い数値, または文献に報告されている数値のうち最も速い数値を最高入力速度とした.

	最高入力速度	追加のデバイス	キー配置	表示文字数
キーボード	70 CPM 以上 ¹	必要	PC 向けのキーボードと同一	(ウィジェット不要)
50音キーボード	53 CPM ²	不要	50音表に準拠	約50文字
EGConvert ^[16]	43 CPM ^{[17] 3}	不要	独自のキー配置	16文字
IToNe ^[17]	160 CPM ^{[17] 4}	不要	独自のキー配置	10文字
JoyFlick	93 CPM ²	不要	フリック入力に準拠	15文字

■ゲーム機に付属しているデバイスのみを用いること
 ゲーム機に付属しないデバイスを用いる手法は, ユーザに対して, デバイスの調達, デバイスの操作への習熟, およびデバイスの接続のための追加のコストを要求する. 一方, ゲームパッドのようなゲーム機に付属するデバイスは, ユーザがデバイスを新たに調達する必要および新たなデバイスの操作に慣れる必要がなく, あらかじめゲーム機にデバイスが接続されていることが多く, また, デバイスをゲーム機に接続する簡単な手法が提供されていることが多い. これらのコストを削減できることは, ユーザビリティの向上に貢献するだろう.

■習得が容易であること

ゲームパッドを用いて文字入力を行うユーザのほとんどは, すでに50音キーボードに慣れていると考えられる. そのため, 新たなかな文字入力手法の導入コストが高い場合, たとえ新たなかな文字入力手法を用いる方が高速な入力が可能であっても, ユーザは使い慣れた50音キーボードを使い続ける可能性がある. したがって, 新たなかな文字入力手法は, ユーザにとって習得が容易な手法である必要がある.

■ウィジェットが小さいこと

一度に表示する文字の数を減らすことにより, ソフトウェアキーボードのウィジェットを小型化できる. ウィジェットが小さくなると, ウィジェットに隠れるコンテンツの面積が少なくなり, 開発者はより自由に画面上のスペースを使えるようになるため(より大きくコンテンツを表示する, ソフトウェアキーボードに予測変換を実装するなど)コンテンツの表現の幅が広がると考えられる.

3.1 キー配置

JoyFlick のキー配置はフリック入力に倣っている(図3). このキー配置により, フリック入力のユーザは学習コストを抑えられると考えられる.

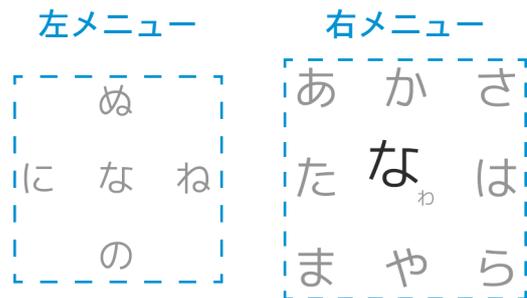


図3 2本のスティックがいずれもニュートラルポジションにあるときの JoyFlick のウィジェット. ウィジェットの左半分は左メニュー, 右半分は右メニューが存在する.

ただし, 「わ」キーの位置およびサイズは, 一般的なフリックキーボードとは異なる. これは, ニュートラルポジションの状態からスティックを押し込むことにより子音「わ」を選択できることをユーザにアフォードするためのものである. なお, 子音「わ」が選択されているときは, 図3右の状態から, 「わ」キーと「な」キーのサイズおよび位置が入れ替わる. 一方, 井川らの手法^[7]において, 「わ」キーは「な」キーと「や」キーの間に配置されている. このキー配置は, JoyFlick のキー配置よりもフリック入力のキー配置に近く, 学習コストを削減できる利点があると考えられる. 井川らの手法^[7]において, 「や」キーと「わ」キーの押し分けはフリックの距離に基づいて行われている. しかし, スティックを用いて距離のような連続値を入力するには, ユーザがスティックの倒し加減に気を配る必要がある. より少ない負担による文字入力の実現を目指すため, JoyFlick はこのキー配置を用いていない.

ユーザはスティックを倒す, 押し込む, あるいはニュートラルポジションにすることにより母音および子音を選択する. なお, 子音「わ」および母音「あ」は対応するスティックの押し込み, また子音「な」は対応するスティックをニュートラルポジションにすることにより選択する. また, 左スティックがニュートラルポジションにあるとき, 母音選択「なし」とする(図3左).

1: 日本情報処理検定協会の日本語ワープロ検定試験1級の試験基準 (https://www.goukaku.ne.jp/test_wordpro.html) より推定
 2: 本研究の長期実験にて記録された, 日時および個人ごとの入力速度のうち最も速い数値
 3: 1時間の練習の後に記録された数値
 4: 1日1時間の練習を1週間行った後に記録された数値

含まれるキーがより多いメニューは、ユーザにより精度の高い操作を要求する。多くのユーザが利き手を用いてキー数の多いメニューを操作できるよう、右スティックで操作する右メニューが子音選択に対応している。また、左スティックで操作する左メニューが母音選択に対応している。

3.2 入力手順

ユーザはまず子音選択を行い (図 4a)、次に母音選択を行う (図 4b および図 4c)。そして最後に、母音選択に用いた左スティックをニュートラルポジションに戻すことにより、1 文字が入力される (図 4d)。

試作の過程において、高速な文字入力を試みるユーザは、文字入力時、左スティックをニュートラルポジションに戻すと同時に、右スティックもニュートラルポジションに戻す傾向が見られた。その際、右スティックが左スティックよりも先にニュートラルポジションに戻った場合、子音が「な」に入れ替わる誤入力が発生することがあった (例:「つ」を入力するはずが、右スティックが先に戻ったために子音「な」が選択された結果、誤って「ぬ」を入力した)。この誤入力を防ぐため、JoyFlick を用いるユーザは、母音選択後から文字が入力されるまでの間、子音選択を行うことができないよう定めた (図 5)。

3.2.1 子音選択

ユーザはまず、ゲームパッドの右スティックを用いて子音を選択する (図 4a および図 5a)。右スティックをニュートラルポジションにすることにより子音「な」、押し込むことにより子音「わ」を選択できる。また、「な」および「わ」を除く 8 つの子音を、それぞれ右メニュー (図 3 右) の中心からの方向に右スティックを倒すことにより選択できる。

なお、母音選択を行うまで、ユーザは子音選択をやり直せる (図 5a)。

3.2.2 母音選択

ユーザは子音選択の次に、ゲームパッドの左スティックを用いて母音を選択する (図 4b および図 5b)。左スティックを押し込むことにより、母音「あ」を選択できる。また、「あ」を除く 4 つの母音を、それぞれ左メニュー (図 3 左) の中心からの方向に左スティックを倒すことにより選択できる。なお、左スティックがニュートラルポジションにある場合は、いずれの母音も選択されない。

なお、文字入力を行うまで、ユーザは母音選択をやり直せる (図 4c および図 5c)。

3.2.3 文字入力

ユーザは母音選択の次に、ゲームパッドの左スティックをニュートラルポジションに戻すことによって、文字を入力する (図 4d および図 5d)。

3.2.4 特殊入力

ユーザはゲームパッドのトリガボタンを押すことにより、最後に入力した文字を濁音、半濁音、捨て仮名、および清音に変換できる。最後に入力した文字に複数の変換が適用できる場合、濁音化、半濁音化、捨て仮名化、静音化の順に適用される (例:「は → ば → ぱ → は」、「つ → づ → っ → つ」)。

ゲームパッド前面の右下に存在するボタン⁵を押すことにより、最後に入力した文字を消去できる。

3.3 フィードバック

ウィジェットの右メニューには選択されている子音が、左メニューには選択されている子音および母音が常に反映される (図 4)。

今里らは、ジョイスティックにクリック感がないことが誤入力に繋がっている可能性を指摘した^[15]。これを踏まえ、選択されている子音または母音に変更された時、JoyFlick はゲームパッドのバイブレーション機能を用いてユーザへ振動フィードバックを与える。また、現代のゲームパッドには左半分および右半分に 1 つずつ、計 2 つの振動モータを内蔵し、これらを別個に動かせるものがある。この場合、より精密にクリック感を再現するため、対応するスティックにより近い振動モータを駆動する。

4 初期実験

我々はまず、JoyFlick を初めて扱うユーザの入力速度および精度を調べるための初期実験を行った^[2]。

4.1 実験参加者

日本語を母語とする、19–25 (M = 21.8, SD = 1.19) 歳の 24 名が実験に参加した。2 名は女性であり、残りは男性だった。1 名は左利きであり、残りは右利きだった。全ての実験参加者は JoyFlick を用いた文字入力の経験がなかった。一方、全ての実験参加者は 50 音キーボードを用いた文字入力の経験があった。18 名はスマートフォンにおいてフリック入力を用いており、残りは QWERTY 配列のソフトウェアキーボードを用いていた。初期実験において、フリック入力の経験を持つため、JoyFlick の学習コストを抑えられると考えられる前者を TARGET-USERS、そうでない後者を OTHERS と呼ぶ。

COVID-19 対策のため、12 名 (オンライン参加者) は Discord⁶ を用いたオンラインの実験に参加した。残り (オフライン参加者) は実験室においてオフラインの実験に参加した。オンライン参加者とオフライン参加者の入力速度および精度に有意差は見られなかった^[2]。



図4 JoyFlick を用いて文字「く」を入力する操作の例. まず, ユーザは右スティックを上倒して (a) 子音「か」を選択する. 続いて, (b) 誤って左スティックを左倒して母音「い」を選択してしまったため, (c) 左スティックを倒したまま, 倒す向きを変えて母音「う」を選択する. 最後に, (d) 左スティックをニュートラルポジションに戻して文字「く」を入力する.

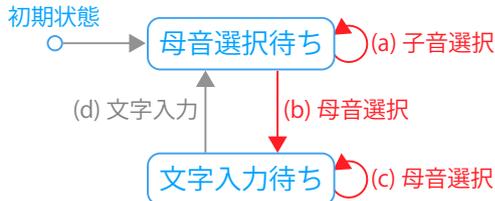


図5 JoyFlick の状態遷移図. 赤矢印は振動フィードバックを伴う遷移を示す.

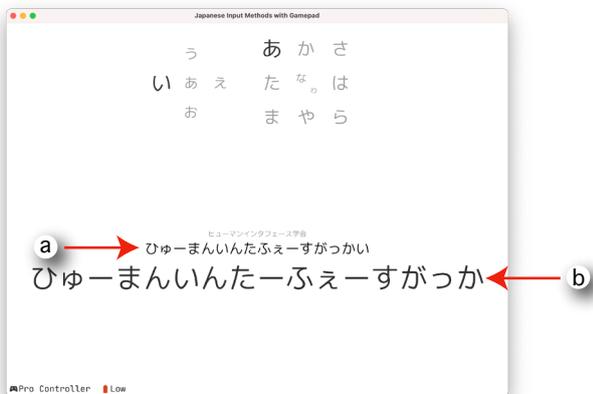


図6 実験用のアプリケーションの表示例. a: 参加者に入力するよう求められている文字列. b: 参加者が入力した文字列.

4.2 実験に用いた装置およびソフトウェア

我々は Rust 言語を用いて, 実験用のアプリケーションを Windows 10 および macOS 上に実装した (図 6). 実験にはゲームパッドとして Nintendo Switch Pro Controller を用い, ゲームパッドとコンピュータを Bluetooth を介して接続した. また, ゲームパッドとコンピュータの通信を joycon-rs^[18] を用いて制御した. オンラインの実験の参加者には, この実験用のアプリケーションを配布した.

4.3 タスク

タスクは, 表示された短文 (図 6a) を書き写す作業である. 参加者はなるべく高速かつ精確に入力するよう求められた. 削除ボタンを用いて入力誤りを訂正することができたが, 入力誤りを全て訂正するよう求められてはなかった. ゲームパッドのプラスボタンを押下することにより, 参加者は次の短文の書き写しに進むことができた.

4.4 実験手順

実験は 2 つのセッションによって構成されており, 1 つは JoyFlick, もう 1 つは 50 音キーボードを用いた入力を行うセッションだった. 各セッションにおいて, まず参加者は最初にセッションにて用いる文字入力手法の操作方法の説明を受けた. 次に参加者は, 練習として全ての練習文を一度ずつ入力した. 最後に参

⁵ Nintendo Switch Pro コントローラーにおいては B ボタン, また DualSense においては X ボタンと呼ばれる.

⁶ <https://discord.com/>

加者は、計測として全ての課題文を一度ずつ入力した。順序による効果を打ち消すため、セッションの順序、および練習文・課題文の順序のカウンターバランスを取った。実験に用いた練習文は 28 文であり、累計 289 文字だった。また、課題文は 18 文であり、累計 168 文字だった。課題文および練習文はかな文字の静音、濁音、半濁音、および捨てがなど、長音記号「ー」によって構成されていた。

実験の最初に、我々は参加者の属性を調べるためのアンケートを行った。続いて、2つのセッションを行った。

参加者 1 人につき、実験に要した時間は 1 時間以下だった。また、参加者 1 人につき、入力するよう求められた文字数は式 1 の通りだった。

$$2 \text{ セッション} \times (289 \text{ 文字} + 168 \text{ 文字}) = 914 \text{ 文字} \quad (1)$$

4.5 結果と議論

入力の速度および精度を解析対象とした。入力速度の指標として CPM (Characters Per Minute) を用いた。また、精度の指標としてトータルエラー率および MSD (Minimum String Distance) エラー率^[19]を用いた。

統計解析の独立変数となったのは、入力手法 (JoyFlick および 50 音キーボード) および参加者がスマートフォンにおいて普段使っている入力手法 (TARGET-USERS および OTHERS) の 2 つである。Shapiro-Wilk 検定がトータルエラー率および MSD エラー率のデータに正規性がないことを示したため、これらの指標の統計解析には Steel-Dwass 検定を用いた。また、CPM の統計解析には、二元配置分散分析を用いた後、Tukey の多重比較法を用いて事後検定を行った。検定結果のうち、主効果あるいは有意差が検出されたもののみを示す。なお、課題文に対する入力のみを検定の対象とした。

4.5.1 入力速度

JoyFlick および 50 音キーボードの平均入力速度を図 7 に示す。JoyFlick の平均入力速度は 42.8 CPM (SD = 8.98 CPM)、50 音キーボードの平均入力速度は 37.2 CPM (SD = 5.96 CPM) だった。また、この差は有意だった ($p < 0.05, F_{1,44} = 10.4, \eta^2 = 0.166$)。

JoyFlick を用いた TARGET-USERS および OTHERS の平均入力速度を図 8 に示す。JoyFlick を用いた TARGET-USERS の平均入力速度は 45.2 CPM (SD = 8.23 CPM)、OTHERS の平均入力速度は 35.6 CPM (SD = 7.52 CPM) だった。また、この差は有意だった ($p < 0.05, d = 1.16$)。

なお、TARGET-USERS の平均入力速度は JoyFlick を用いたとき 45.2 CPM (SD = 8.23 CPM)、50 音キー

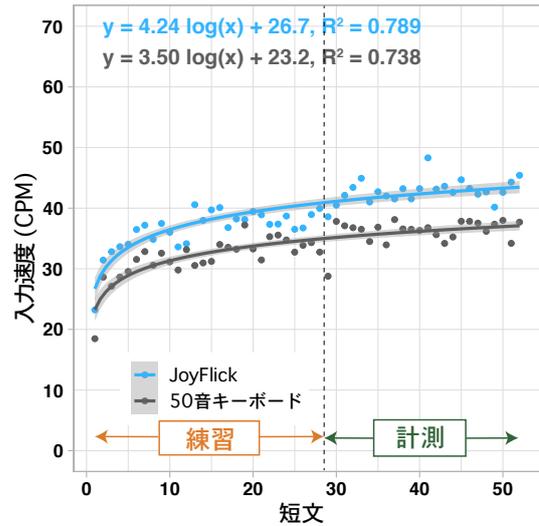


図 7 JoyFlick および 50 音キーボードの平均入力速度。灰色の帯は 95%信頼区間を表す。

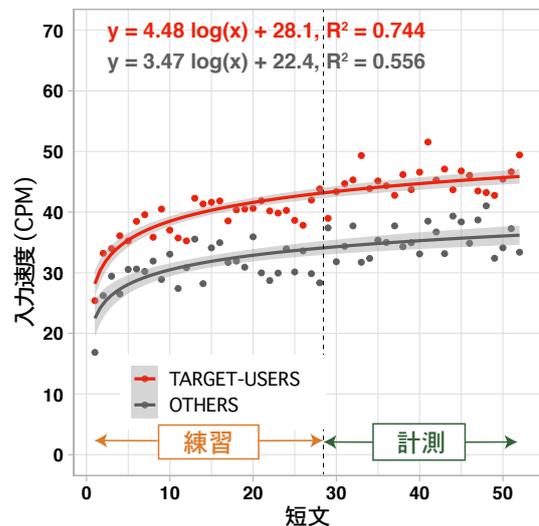


図 8 TARGET-USERS および OTHERS の、JoyFlick の平均入力速度。灰色の帯は 95%信頼区間を表す。

ボードを用いたとき 36.3 CPM (SD = 6.15 CPM) だった。また、この差は有意だった ($p < 0.05, d = 1.20$)。

これらの結果から、289 文字の練習の後、TARGET-USERS の JoyFlick を用いた入力は、50 音キーボードよりも高速であることがわかる。さらに、TARGET-USERS の JoyFlick を用いた入力が、常に OTHERS よりも高速である (図 8) ことから、TARGET-USERS の持つフリック入力の経験が、JoyFlick を用いた入力を高速にしていたと考えられる。

4.5.2 精度

平均トータルエラー率は、JoyFlick では 8.22% (SD = 3.90%)、50 音キーボードでは 5.29% (SD = 7.75%) だった。また、TARGET-USERS の JoyFlick

を用いた入力の平均トータルエラー率 (8.22%, SD = 3.46%) と, TARGET-USERS の 50 音キーボードを用いた入力の平均トータルエラー率 (6.05%, SD = 8.83%) に有意差が示された ($p < 0.05, d = 0.315$).

一方, 平均 MSD エラー率は, JoyFlick では 0.648% (SD = 0.965%), 50 音キーボードでは 0.749% (SD = 0.955%) だった.

すなわち, JoyFlick のトータルエラー率は 50 音キーボードに比べて高い傾向にあるが, 両者の MSD エラー率は同程度であった. この結果から, JoyFlick を用いた参加者は, 短文を書き写す過程においては 50 音キーボードよりも誤入力が多いものの, 最終的にはその誤入力を修正できていたと考えられる.

5 長期実験

JoyFlick を継続的に使用し始めたユーザの入力速度および精度を調べるため, 初期実験に続いて, 50 音キーボードとの比較を用いた 1 週間の長期実験を行った [2].

5.1 実験参加者

日本語を母語とする, 20–25 (M = 21.9, SD = 1.28) 歳の 15 名が実験に参加した. 全ての参加者が初期実験に参加していた. 全ての参加者は男性だった. 1 名は左利きであり, 残りは右利きだった. 11 名はスマートフォンにおいてフリック入力を用いており, 残りは QWERTY 配列のソフトウェアキーボードを用いていた. 以降, 長期実験において, フリック入力の経験を持つため, JoyFlick の学習コストを抑えられると考えられる前者を TARGET-USERS, そうでない後者を OTHERS と呼ぶ.

5.2 実験に用いた装置およびソフトウェア, タスクおよび実験手順

実験には初期実験と同じソフトウェアおよびタスクを用いた.

実験は 7 日間連続で行われた. 参加者は 1 日に, JoyFlick を用いた入力および 50 音キーボードを用いた入力, 合わせて 2 回の入力を行った. 1 回の入力において, 参加者は出題された短文の文字の合計が 140 文字を超える直前まで, 課題文を入力した. 1 回の計測において出題された課題文の数はおよそ 18 文であり, また出題された課題文の文字数は平均 135 文字だった. 順序による効果を打ち消すため, 入力手法の順序および課題文の順序のカウンターバランスを取った.

参加者 1 人につき, 1 日分の計測に要した時間は 15 分以下だった. また, 参加者 1 人につき, 入力するよう求められた累計の文字数はおよそ式 2 の通りだった.

$$7 \text{ 日} \times (135 \text{ 文字} \times 2 \text{ 回}) = 1890 \text{ 文字} \quad (2)$$

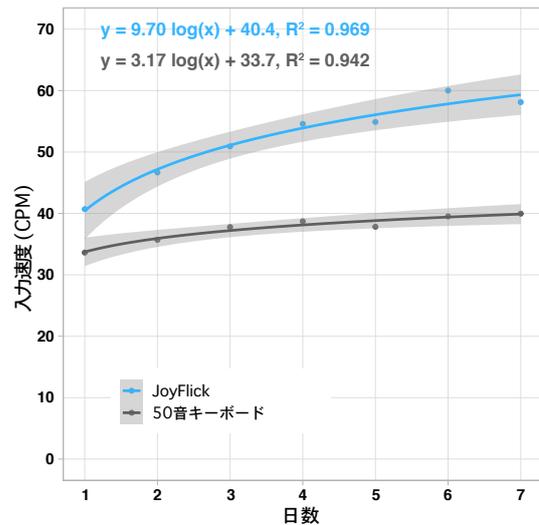


図9 JoyFlick および 50 音キーボードの平均入力速度. 灰色の帯は 95% 信頼区間を表す.

5.3 結果と議論

入力の速度および精度を解析対象とした. また, 入力速度の指標として CPM を, 精度の指標としてトータルエラー率および MSD エラー率を用いた.

統計解析の独立変数は, 日数 (7 日間), 入力手法 (JoyFlick および 50 音キーボード) および参加者がスマートフォンにおいて普段使っている入力手法 (TARGET-USERS および OTHERS) の 3 つである. ただし, Shapiro-Wilk 検定がトータルエラー率および MSD エラー率のデータに正規性がないことを示したため, これらの指標の最終日のデータのみを統計解析の対象とし, Steel-Dwass 検定を用いた. また, CPM のデータに対しては, 反復測定分散分析を行った後, Tukey の多重比較法を用いて事後検定を行った. 検定結果のうち, 主効果あるいは有意差が検出されたもののみを示す.

5.3.1 入力速度

JoyFlick および 50 音キーボードの平均入力速度を図 9 に示す. 統計解析の結果, 日数の主効果が検出された ($p < 0.005, F_{1,189} = 118, \eta^2 = 0.102$). また, JoyFlick と 50 音キーボード間に有意差が示された ($p < 0.005, d = 7.11$). さらに, 最終日, JoyFlick の平均入力速度は 58.1 CPM (SD = 11.2 CPM), 50 音キーボードの平均入力速度は 40.0 CPM (SD = 7.50 CPM) だった.

JoyFlick を用いた TARGET-USERS および OTHERS の平均入力速度を図 10 に示す. 両者の間に有意差が示された ($p < 0.005, d = 0.624$).

なお, TARGET-USERS の平均入力速度は, JoyFlick を用いたとき 52.3 CPM (SD = 12.7 CPM), 50 音キーボードを用いたとき 37.6 CPM (SD = 5.86 CPM) だっ

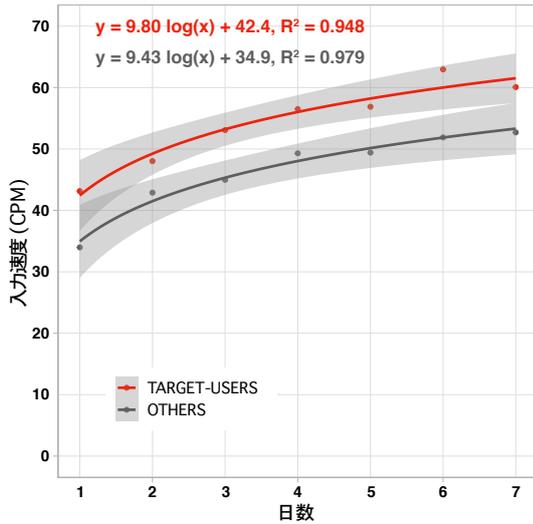


図 10 TARGET-USERS および OTHERS の, JoyFlick の平均入力速度. 灰色の帯は 95%信頼区間を表す.

た. また, 両者の差は有意だった ($p < 0.005, d = 2.03$).

最終日, JoyFlick の平均入力速度は 50 音キーボードの 1.45 倍に達した. この結果から, 初期実験における 457 文字および長期実験における 945 文字の入力によって, JoyFlick を用いた入力は, 50 音キーボードよりも高速になったことがわかる. さらに, TARGET-USERS の JoyFlick を用いた入力が, 常に OTHERS よりも高速である (図 10) ことから, 初期実験と同様に, TARGET-USERS の持つフリック入力の経験が, JoyFlick を用いた入力を高速にしていると考えられる.

5.3.2 精度

最終日の平均トータルエラー率は, JoyFlick では 11.2% (SD = 6.49%), 50 音キーボードでは 2.53% (SD = 12.2%) だった. TARGET-USERS が JoyFlick を用いたとき (11.3%, SD = 7.53%) と 50 音キーボードを用いたとき (2.35%, SD = 1.15%) の間に有意差が示された ($p < 0.005, d = 1.60$).

最終日の平均 MSD エラー率は, JoyFlick では 1.13% (SD = 0.831%), 50 音キーボードでは 0.244% (SD = 0.357%) だった. TARGET-USERS が JoyFlick を用いたとき (1.15%, SD = 0.705%) と 50 音キーボードを用いたとき (0.265%, SD = 0.368%) の間に有意差が示された ($p < 0.05, d = 1.51$).

JoyFlick の MSD エラー率およびトータルエラー率は, 50 音キーボードよりも高かった. ただし, 最終日において, JoyFlick の MSD エラー率は 1.13% であり, これは入力の過程で生じた誤入力のほとんどが修正されたことを示している.

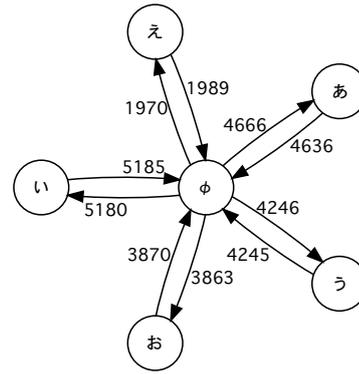


図 11 選択されている母音の遷移. ただし, ϕ はいずれの母音も選択されていない状態を示す. 記録回数が 200 回以下の遷移は省略.

6 スティックへの操作の分析

初期実験および長期実験の参加者が, JoyFlick を用いた入力においてどのような操作を行ったのかを調査することにより, JoyFlick の設計を評価した. 過去の研究 [2] において行われた初期実験および長期実験にて収集したデータが, 調査の対象となった. ただし, 初期実験にて行われた練習の際に収集したデータは, 調査の対象から除かれた.

6.1 母音選択時および子音選択時の操作

JoyFlick を用いるユーザがキーを選択する際, スティックへの操作は「倒す」「回す」「押し込む」の 3 つに分類できる. 「倒す」は, ニュートラルポジションにあるスティックを倒すことによりキーを選択する操作を指す. 「回す」は, スティックを倒したまま, 倒す向きを変えることによりキーを選択する操作を指す. 「押し込む」は, スティックを押し込むことによりキーを選択する操作を指す.

母音選択時および子音選択時, 実験参加者がこれらの操作をどのように組み合わせて選択を行ったのかを分析するため, 我々は選択されているキーの遷移に着目した.

6.1.1 母音選択時の操作

遷移のグラフを図 11 に示す. 図 11 において, ノード「 ϕ 」は他のすべてのノードとの間にエッジを持っている. さらに, 遷移のうち 99.0% が, 母音を選択していない状態から母音を選択する, または母音を選択した状態から母音を選択していない状態へ変わる際に記録された遷移だった. このことから, ユーザが JoyFlick を用いる際, 左スティックはニュートラルポジションを中心とする放射状の軌跡を描くものと考えられる. すなわち, 母音選択は, 左スティックを倒すまたは押し込むという 1 回の操作のみによって行われることが多いと考えられる.

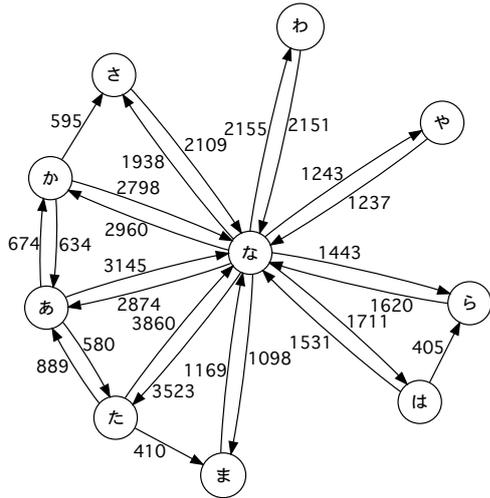


図 12 選択されている子音の遷移. 記録回数が 200 回以下の遷移は省略.

6.1.2 子音選択時の操作

記録されていた遷移のグラフを図 12 に示す. 図 12 において, ノード「な」は他のすべてのノードの間にエッジを持っている. さらに, 記録されていた遷移のうち 85.3%が, 子音「な」から他の子音を選択する際, または他の子音から子音「な」を選択する際に記録された遷移だった. このことから, ユーザが JoyFlick を用いる際, 右スティックはニュートラルポジションを中心とする放射状の軌跡を描くものと考えられる. すなわち, 子音選択は, 右スティックを倒すまたは押し込むという 1 回の操作のみにより行われることが多いと考えられる. また, 次の子音選択を行う際には, ユーザは一旦スティックをニュートラルポジションに戻すことが多いと考えられる.

6.1.3 考察

実験参加者は子音または母音を選択するために, スティックを倒す操作を使うことが多かった (図 11 および図 12). これは Fitts の法則^[20]を用いて説明できる可能性がある. スティックを倒してキーを選択するタスクをポインティングタスクとして考えたとき, パイメニュー周縁部に存在するキーは無限の幅を持っている^[12]. 一方, 倒されたスティックの方向を変えてキーを選択するタスクをポインティングタスクとして考えたとき, パイメニュー周縁部に存在するキーの幅は限られている. そのため, 高速かつ正確にかな文字を入力するよう求められた実験参加者にとって, 母音および子音の選択にスティックを倒す操作を用いるのは, 倒されたスティックの方向を変える操作を用いるよりも効率的であったと考えられる.

実験参加者は, あらかじめニュートラルポジションにスティックを戻しておき, スティックを倒すまたは押し込むという 1 回の操作のみによって子音選択を行

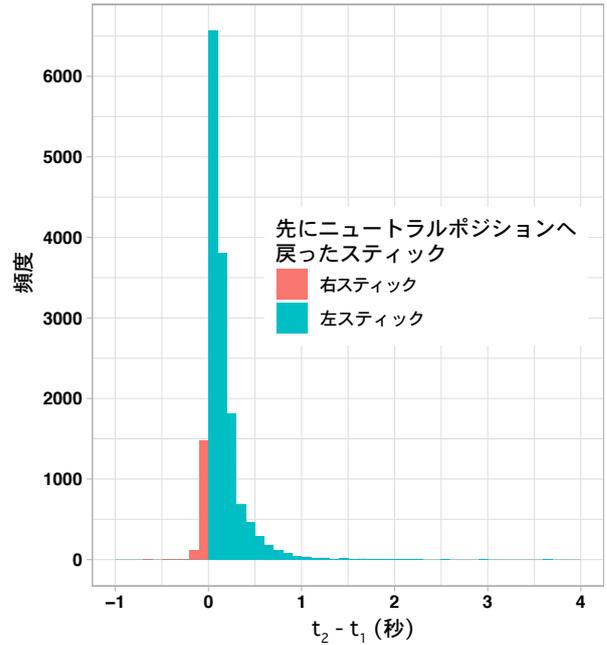


図 13 実験参加者が左右両方のスティックをニュートラルポジションに戻したときの, 左右のスティックの時間差のヒストグラム. 左スティックをニュートラルポジションに戻した時刻を t_1 , 右スティックを t_2 とする.

うことを好んだ. 仮に, ニュートラルポジションにスティックを戻さずに子音選択を行う場合, つまり, スティックを回して子音を選択する場合, 必要な操作は操作前のスティックの位置および目的のキーの 2 つの要素に依存する. 一方, ニュートラルポジションにスティックを戻してから子音選択を行う場合, 操作前のスティックは必ずニュートラルポジションに位置しているため, 実験参加者が子音を選択するのに必要な操作は目的のキーという 1 つの要素のみに依存する. 必要な操作を把握するための要素が少ないことは, 実験参加者の認知負荷の軽減および必要な操作を把握するための時間の短縮 (すなわち, 入力的高速さ) に寄与したと考えられる. そのため, 実験参加者は後者の入力方法を好んだと考えられる. また, スティックには指を離すと自動かつ素早くニュートラルポジションに戻るといった性質がある. スティックをニュートラルポジションに戻るのが容易であることは, 実験参加者がこの入力方法を好んだ理由の 1 つであると考えられる.

6.2 文字入力時の操作

JoyFlick を用いるユーザは, 母音を選択してから 1 文字の入力を完了するまで, 新たな子音選択が行えない. 試作の段階において, ユーザが文字入力の際に左右両方のスティックをほぼ同時にニュートラルポジションに戻す傾向が見られたため, この制限が導入された (3.2 節). この設計の有用性を確かめるため, 実験参加者の操作に同様の傾向があるか否かを調査した.

文字入力の際に、左右両方のスティックがニュートラルポジションに戻る確率は 79.1% だった。また、左右両方のスティックがニュートラルポジションに戻る時の、左右のスティックの時間差 ($M = 15.1 \times 10^{-2}$ 秒, $SD = 19.7 \times 10^{-2}$ 秒) の分布は図 13 のようになった。したがって、ユーザは文字入力時、左右のスティックをほぼ同時にニュートラルポジションに戻すことが多いと言える。

さらに、文字入力に際して、右スティックが左スティックよりも早くニュートラルポジションに戻る確率は、8.02% だった。したがって、仮に前述した制限を JoyFlick の設計から排した場合、ユーザの文字入力のうち 8% 程度が新たに誤入力となってしまうと考えられる。ゆえに、前述した制限は JoyFlick の設計に必要なものであったと言える。

7 今後の課題

今回分析対象とした実験において用いられた短文は、かな文字および長音記号によって構成されていた。しかし、実際にユーザが入力する文章には、カタカナおよび漢字などの変換機能を用いて入力する文字が含まれている。そのため、変換機能を用いる必要のある文を入力する実験を行うことにより、より実際の使用環境に近い環境における、JoyFlick を用いるユーザの入力速度および精度を調査する必要がある。

JoyFlick を用いたかな文字入力を行うには、2 本のスティックと 2 つ以上のボタンを持つデバイスが必要である。この条件に該当するデバイスとして、ゲームパッドの他に、一部の VR 用のコントローラが挙げられる。JoyFlick を用いたかな文字入力手法は、ハンドトラッキングを使う文字入力手法^[21] およびレイキャストを使う文字入力手法に比べて、ユーザの腕の姿勢に対する制限が小さいという利点があると考えられる。ユーザの入力速度、エラー率、疲労をこれらの手法と比較する実験を行うことにより、JoyFlick を VR 環境におけるかな文字入力に用いることのメリットおよびデメリットを調査する必要がある。

JoyFlick は 50 音キーボードに比べて高速かつ画面占有率低い手法であるため、ゲーム中のテキストチャットに用いるのにより適した手法であると言える。したがって、JoyFlick がユーザによる既存の文字入力を伴う機能の使用を促進しているかどうかを検証するため、ゲーム中のテキストチャットに 50 音キーボードを用いるユーザと JoyFlick を用いるユーザを比較する実験を行う必要がある。

8 まとめ

本論文において、フリック入力に基づく 2 本の押し込み機能付きジョイスティックを用いたテキスト入力

手法 JoyFlick の設計を述べ、入力速度および精度を評価する実験を行い、また設計を検証するための分析を行った。

JoyFlick を用いるユーザは、子音および母音を選択するため、スティックを倒す操作を使う傾向がある。また、選択と選択の合間に、スティックをニュートラルポジションに戻す傾向がある。さらに、文字入力の際に、左右両方のスティックをほぼ同時にニュートラルポジションに戻す傾向がある。JoyFlick は母音と子音の選択の順序を制限しているが、この設計が 3 つ目の傾向による誤入力を防いでいることがわかった。

JoyFlick の実用性をさらに示すためには、予測変換と組み合わせる、またゲームに組み込むなどして、実際の使用状況に近い環境において入力速度および精度を調査する必要がある。

参考文献

- [1] 横山海青, 高倉礼, 志築文太郎: JoyFlick: フリック入力に基づくゲームパッド向けかな文字入力手法, WISS2020, 日本ソフトウェア科学会, pp. 97–102 (オンライン), 入手先 (<https://www.wiss.org/WISS2020Proceedings/data/17.pdf>) (2020).
- [2] Yokoyama, K., Takakura, R. and Shizuki, B.: JoyFlick: Japanese Text Entry Using Dual Joysticks for Flick Input Users, *Proceedings of 18th IFIP TC13 Interaction Conference on Human-Computer Interaction (INTERACT 2021)*, Springer International Publishing (2021). 19 pages. (to appear).
- [3] Kojima, K.: On the Information Literacy of the Net Generation in the BYOD Era, *Journal of Nagoya Gakuin University; SOCIAL SCIENCES*, Vol. 52, No. 3, pp. 45–57 (online), available from (<https://doi.org/10.15012/00000619>) (2016). (in Japanese).
- [4] Nagasawa, N.: How Japanese University Students Type on Smartphone and PC, *Computer & Education*, Vol. 43, pp. 67–72 (online), available from (<https://doi.org/10.14949/konpyutariyoukyouiku.43.67>) (2017). (in Japanese).
- [5] Masui, T. and Sakurai, Y.: Blossom - Flick-based Japanese Input System on QWERTY keyboard, *IPSJ SIG Technical Report*, No. 5, pp. 1–4 (online), available from (<http://id.nii.ac.jp/1001/00094712/>) (2013). (in Japanese).
- [6] Sakurai, Y.: keroxp/Blossom-iOS: SoftwareKeyboard for iOS, <https://github.com/keroxp/Blossom-ios>. accessed: 2021-05-05.
- [7] Ikawa, Y. and Miyashita, H.: “Direction Only” Flick-Input Method for Shorthand with Eyes-Free, *Proceedings of Interaction 2013*, pp. 651–656 (online), available from (<http://www.interaction-ipsj.org/archives/paper2013/data/Interaction2013/interactive/data/pdf/3EXB-22.pdf>) (2013). accessed: 2021-05-05.
- [8] Fukatsu, Y., Shizuki, B. and Tanaka, J.: No-Look Flick: Single-Handed and Eyes-Free Japanese Text Input System on Touch Screens of Mobile Devices,

- Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '13, New York, NY, USA, Association for Computing Machinery, p. 161–170 (online), available from (<https://doi.org/10.1145/2493190.2493243>) (2013).
- [9] Wobbrock, J., Myers, B. and Aung, H.: Writing with a Joystick: A Comparison of Date Stamp, Selection Keyboard, and Edge Write., pp. 1–8 (online), available from (<http://faculty.washington.edu/wobbrock/pubs/gi-04.pdf>) (2004).
- [10] Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W. and Liang, H.: PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24, No. 11, pp. 2927–2935 (online), available from (<https://doi.org/10.1109/TVCG.2018.2868581>) (2018).
- [11] Költringer, T., Isokoski, P. and Grechenig, T.: TwoStick: Writing with a Game Controller, pp. 103–110 (online), available from (<https://doi.org/10.1145/1268517.1268536>) (2007).
- [12] Sandnes, F. E. and Aubert, A.: Bimanual text entry using game controllers: Relying on users' spatial familiarity with QWERTY, *Interacting with Computers*, Vol. 19, No. 2, pp. 140–150 (online), available from (<https://doi.org/10.1016/j.intcom.2006.08.003>) (2007). HCI Issues in Computer Games.
- [13] Go, K. and Kinoshita, Y.: Katakana EdgeWrite: An EdgeWrite Version for Japanese Text Entry, *IEICE Transactions on Information and Systems*, Vol. E97.D, No. 8, pp. 2053–2054 (online), available from (<https://doi.org/10.1587/transinf.E97.D.2053>) (2014).
- [14] Nakamura, S., Masahiko, T. and Shojiro, N.: A Text Input Method Using the Double-Mouse for Wearable Computing Environments, *IPSJ SIG Technical Report*, No. 97, pp. 63–68 (online), available from (<http://id.nii.ac.jp/1001/00037112/>) (1999). (in Japanese).
- [15] 今里育夫：ジョイスティック型片手操作入力機器の開発，ライフサポート， Vol. 16, No. 1, pp. 6–7 (オンライン)，入手先 (<https://doi.org/10.5136/lifesupport.16.6>) (2004).
- [16] 株式会社エルゴソフト：文字入力制御方法、プログラム、記録媒体及び文字入力装置，特開 2002-268818 (2002).
- [17] Go, K., Konishi, H. and Matsuura, Y.: IToNe: A Japanese Text Input Method for a Dual Joystick Game Controller, *Proceedings of CHI '08 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '08, pp. 3141–3146 (online), available from (<https://doi.org/10.1145/1358628.1358821>) (2008).
- [18] Yokoyama, K.: KaiseiYokoyama/joycon-rs: Joy-Con library for Rust, <https://github.com/KaiseiYokoyama/joycon-rs>. accessed: 2021-05-05.
- [19] Arif, A. S. and Stürzlinger, W.: Analysis of Text Entry Performance Metrics, *2009 IEEE Toronto International Conference Science and Technology for Humanity*, TIC-STH '09, pp. 100–105 (online), available from (<https://doi.org/10.1109/TIC-STH.2009.5444533>) (2009).
- [20] Fitts, P. M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement., *Journal of experimental psychology*, Vol. 47, No. 6 (online), available from (<https://doi.org/10.1037/0096-3445.121.3.262>) (1954).
- [21] Komiya, K. and Nakajima, T.: A Japanese Input Method Using Leap Motion in Virtual Reality, *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, Los Alamitos, CA, USA, IEEE Computer Society, pp. 1–2 (online), available from (<https://doi.org/10.23919/ICMU.2017.8330085>) (2017).

(2021年5月5日受付, 7月9日再受付)

著者紹介

横山 海青



1998年生。2021年筑波大学情報学群情報科学類卒業。現在、同大学院システム情報工学研究群情報理工学位プログラム博士前期課程在学中。ヒューマンインタフェースに関する研究に興味を持つ。

高倉 礼



2020年筑波大学情報学群情報メディア創成学類卒業。現在、同大学院システム情報工学研究群情報理工学位プログラム博士前期課程在学中。ヒューマンインタフェースに関する研究に興味を持つ。ACM, 情報処理学会学生会員。

志築 文太郎 (正会員)



1971年生。1994年東京工業大学理学部情報学科卒業。2000年同大学大学院情報理工学研究科数理・計算科学専攻博士課程単位取得退学。博士(理学)。現在、筑波大学システム情報系教授。ヒューマンインタフェースに関する研究に興味を持つ。日本ソフトウェア科学会, ACM, IEEE Computer Society, 情報処理学会, ヒューマンインタフェース学会各会員。

川口 一画 (正会員)



1986年生。2017年筑波大学大学院システム情報工学研究科知能機能システム専攻博士後期課程修了。博士(工学), 修士(感性科学)。現在、筑波大学システム情報系助教。平成27年度総務省異能vationプログラム本採択。CSCW, HRIの研究に従事。ヒューマンインタフェース学会, 情報処理学会, 感性工学会会員。

