

# スクリーンリーダーで操作するジグソー・テキスト - アクセシブルな学習分析と Computer Based Testing

山口 琢<sup>1</sup> 大場 みち子<sup>2</sup>

**概要:** ジグソーテキストはテキストのピースをドラッグ&ドロップで並べ替えて完成させる、読解・作文のジグソー・パズルである。プログラミングについても同様なジグソー・コードがある。パソコンやスマートフォンや Voice User Interface (VUI) など、マルチモーダルに使える Web アプリを目指している。演習問題アプリとして、また並べ替え操作を測定する学習分析に利用されている。アクセシブルな学習分析と Computer Based Testing (CBT) を目指して、第一歩として、スクリーンリーダーで操作できる UI を試作した。移動するピースと移動先を順に選択する操作でピースを並べ替える。並べ替え操作を、ドラッグ&ドロップの UI と同様に測定・分析できた。スマートフォンのスクリーンリーダーでは、パズル中の空白文字の扱われ方に注意が必要と分かった。

**キーワード:** 並べ替え作文、並べ替えプログラミング、並べ替え操作の測定・分析、アクセシビリティ、学習分析、Computer Based Testing、インクルーシブ教育、スクリーンリーダー、空白文字

## Jigsaw Text with screen reader: toward accessible learning analytics and computer-based testing

TAKU YAMAGUCHI<sup>1</sup> MICHIKO OBA<sup>2</sup>

**Keywords:** reordering texts, reordering program codes, measurement and analysis of reordering operations, accessibility, ally, learning analytics, computer-based testing, electronic assessment, inclusive education, screen reader, space character

### 1. はじめに

Computer-based testing (CBT) の導入が進んでいる。試験にコンピュータを導入することで、紙ベースの試験 (Paper-based testing) ではできなかった試験が可能となる。「思考力・判断力・表現力」を評価できると期待され、例えば、プログラミングの技能を測る試験出題形式として期待されている [1][2][3]。そこでは、解答中に問題を動的・自動的に生成するといったことも行われるかもしれない。

2016 年 4 月、障害者差別解消法が施行された。「社会的

障壁の除去の実施についての必要かつ合理的な配慮」が義務・努力義務となった。

教材や試験のデジタル化は、アクセシビリティ向上の可能性を広げるチャンスでもある。例えば電子書籍は、読み上げ機能などによって「読書機会を大きく広げる可能性が期待される」[4]。

受験の時間・場所の自由度が増せば、BYOD (Bring Your Own Device) の可能性も広がるだろう。それに伴って、幅広いデバイス、スクリーンリーダー、ブラウザに対応する相互運用性の課題が解決されていくだろう。

学習分析は、学習者の読書や作文やプログラミングなどの過程を観察して、細かい粒度で測定・分析して、学習や指導に役立てるものである。学習分析は学習者にとって有益なものだから、障害を持つ学習者が他の学習者と平等に

<sup>1</sup> フリー

Independent Researcher

<sup>2</sup> 公立はこだて未来大学システム情報科学部

Faculty of Systems Information Science, Future University  
Hakodate

学習分析の対象となれるよう、合理的に配慮されるようになるだろう。

また障害を持つ研究者が学習分析で活躍するために、グラフィックスなどさまざまな可視化ツールが、アクセシブルになるよう工夫されるだろう。

## 2. 先行する取り組み

関連する取り組みをあげて考察する。学習や試験がコンピュータの支援でマルチモーダル化するとき、教材や問題というコンテンツの再構成が起きる可能性や、学習や受験というコミュニケーションのモードが異なっても同じ理解をしていると判断してしまう可能性を、先行する研究は示唆している。

### 2.1 VUI とマルチモーダルインターアクション

Google Home や Amazon Echo などスマートスピーカー (または AI スピーカー) や、スマートフォンの音声入力などに代表される VUI (Voice user interface) が普及しつつある。

Google Home を使った研究では、Moodle の小テストを VUI で受けられるプロトタイプを開発している。試作の結果、VUI を通した学習教材では、フレーズ (音声メッセージ) の長さや、教材全体の構成を再設計すべきではないかと示唆している [5]。

システムがユーザーに複数の UI を用意することをマルチモーダルインターアクション (Multimodal interaction) と呼んで、1990 年代～2000 年代に盛んに研究された [6]。スマートスピーカーなどの VUI は、商用化されたマルチモーダルインタフェースの事例と言えるだろう。

モード、モーダルということばは、パス、チャンネルといった意味で使い、マルチモーダルを、複数のチャンネルがあるという意味で使っていると考えられる。

学習分析では、同じ学習者を複数種類の信号で観察・測定して分析することをマルチモーダル学習分析 (Multimodal Learning Analytics, MMLA) と呼んでいる [7]。ここでは、同じ一人の学習者を複数のモードで測定・分析したとき、あるモードで推定される学習者の心的状態や理解状態が、別のモードで推定されるものと同じなのか、どう異なるのかが不明なことを、注意点として指摘している。そして、そのうえで学習者の心的状態と理解状態を関連づけることが課題であると指摘している。

### 2.2 アクセシブルな教材パズルアプリ

アクセシビリティ機能を持つ、クロスプラットフォームの UI フレームワークを使って、スマートフォン用にアクセシブルなパズルアプリケーションを開発した研究がある [13]。15 パズルや mn-1 パズルをアプリ化したもので、パズルのピースをタッチすると、スマートフォンのスク

リーンリーダーがそのピースを読み上げる。これは、スマートフォン用の数学教材アプリを、アクセシブルにできた事例である。

### 2.3 Web アクセシビリティ

JavaScript や CSS などによって Web ページに機能を追加し、一方通行の読み物からインターアクションするアプリケーションにできる。これをリッチインターネットアプリケーション (rich internet application, RIA) と呼ぶ。

HTML のタグのセマンティックに沿ってページを作ること、Web ページのアクセシビリティ (accessibility, a11y と略されることがある) は高まる。スクリーンリーダーは、Web ページを読み上げるだけでなく、リンクを選択したり、フォームのチェックボックスにチェックを入れたり、プルダウンメニューの項目を選択したり、など、ユーザからの入力も支援する。スマートフォンのスクリーンリーダーは、まずタップされた部分を読み上げる。続いてダブルタップすることで、今読み上げた部分を選択することになる。この振る舞いを使って、スクリーン上にあるものを指などで探って配置を把握して、選択・実行する。

スクリーンリーダーは、JavaScript や CSS などによって動的に更新される内容の扱いが苦手である。W3C の Web Accessibility Initiative Accessible Rich Internet Applications (WAI-ARIA)[8] が定めた属性を使うことで、内容の更新などをスクリーンリーダーに伝えることができる。具体例は本稿の「6 結果」で説明する。

普通のをアクセシブルにするという方針は重要である。Web サイトを PC のブラウザでもスマートフォンでも適切に表示されるようにするなど、コンテンツをマルチモーダルに管理したことがあれば、異なるモードのために異なるコンテンツを管理する大変さが分かるであろう。普通のコンテンツがアクセシブルになるように、システムの仕組みを設計することが、アクセシビリティ向上にとって重要であると考えられる。Web ページを HTML のタグのセマンティックに沿って作るとは、その例である。例えば、電子書籍の規格 EPUB は、そのように考えて設計されている。[9]。

### 2.4 ドラッグ&ドロップ

スマートフォンのネイティブなアプリ、例えば iPhone のミュージックのプレイリストや、リマインダーのリマインダーは、iOS の VoiceOver を使ってドラッグ&ドロップで並べ替えられる。ドロップすると、どこにドロップされたかを VoiceOver がしゃべる。しかし、ドラッグ中に、どこにいるかはしゃべらない。

Web アプリ、すなわち JavaScript や CSS を使って Web ページ内に実現したドラッグ&ドロップを、スクリーンリーダーで操作できる例を、筆者は見つけられなかった。

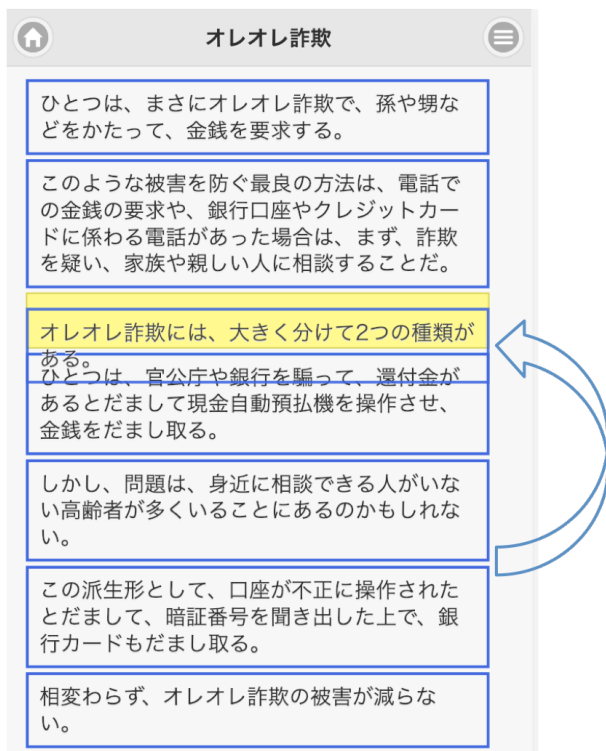


図 1 ドラッグ&ドロップで並べ替えるジグソー・テキスト

ドラッグ&ドロップは、何らかの情報を表現したオブジェクトを移動する手段の一例であるから、その移動を行う別の操作で代替できる場合があり、Web ページでの UI の実装例がある [11]。この実装例では、まず移動するオブジェクトを選択し、これがドラッグ開始に該当する。次に移動先を選択し、これがドロップに該当する。このような操作であれば、VUI での操作も可能であろう。

「このような実装が代替となる」と考えることは、ドラッグ&ドロップの「根底にある機能が、利用者の動作の端点だけでなく、動作による軌跡に依存した入力を要求し」ないとみなすことである [12] (“Success Criterion 2.1.1 Keyboard”)。「代替」を用意することは、インターアクションをマルチモーダル化することである。また、「動作による軌跡に依存」しないと見なして UI が設計されること、および、複数の学習者が異なるインターアクションのモードで学習することは、マルチモーダル学習分析で指摘された課題に影響するであろう。

### 3. われわれの取り組みと課題

ジグソーテキストはテキストのピースをドラッグ&ドロップで並べ替えて完成させる、読解・作文のジグソー・パズルである。プログラミングについても同様なジグソー・コードもある。パソコンやスマートフォンなどマルチモーダルに使える Web アプリを目指している。講義中の演習問題アプリとして使われている。

### 3.1 並べ替え操作の測定・分析

ジグソー・テキストは、プレイヤーの並べ替え操作を測定 (記録) している。測定データは 4 種類: パズル開始、ピースの移動開始、ピースの移動終了、パズル完了である。ドラッグ&ドロップの軌跡は測定していない。ジグソー・コードの測定データを使って、プログラミングの思考過程を分析する学習分析研究が行われている [14]。測定データを分析することで、どういう順番でどのピースに着目するかといった、プレイヤーの思考内容を推定する。

測定データを簡易的に可視化するアプリもある。ジグソー・テキスト分析は、最初に動かされたピース、各ピースが動かされた回数、最終結果の分布やある順序からの編集距離、ピース同士が前後して動かされる頻度 (操作の時間的な共起)、各プレイの操作の時系列などを集計して表示する。最終結果の分布は、演習でも使われていて、学習者がクラス全体の結果を共有してディスカッションすることに役立っている。

### 3.2 マルチモーダル

演習や研究の基盤として、ジグソー・テキストはマルチモーダル化することを目標としている。単純並べ替えの 1 ペイン版、選択肢と解答の 2 ペイン版、選択肢と解答の 2 ペインを画面を切り替えて使うスマホ版を開発してきた。スマホの 2 ペインでは、選択肢から選ぶ操作はドラッグ&ドロップとはならない。

操作の UI はマルチモーダルだが、測定・分析は共通で、パズル開始、ピースの移動開始、ピースの移動終了、パズル完了を記録している。モードが異なっても、同様に解釈できると判断して設計し、同様に測定するよう実装できた。例えば、1 つのピースをドラッグすることも、選択肢から 1 つのピースを選ぶことも、ピースの移動開始として記録している。

その上で、同じ問題データでも、モードの違いによって考え方が違うかどうかといった研究ができると期待している。例えば、PC とスマホで、正答率といった結果だけでなく、考え方というプロセスが変わるかどうかといった研究である。今回、モードに音声 (読み上げ) も含めた。見て読んで操作する場合と、聞いて読んで操作する場合で、解答にいたる過程は同じか?などを問う。さらに、考え方に違いがある場合、それを踏まえて、同じ問題データを使っても UI が異なるならば、それは同じ問題と言えるかといった研究である。

### 3.3 課題

ジグソー・テキストは 3 つのユーザインタフェースを持つ。先生が問題データを登録するインタフェース、学習者がパズルをプレイするインタフェース、先生や学習者や研究者が測定データを概観・分析するインタフェースである

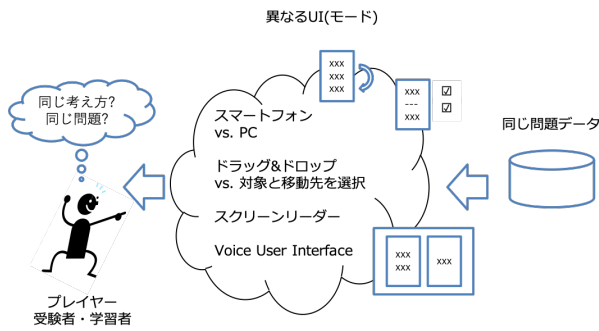


図 2 マルチモーダルなジグソー・テキスト

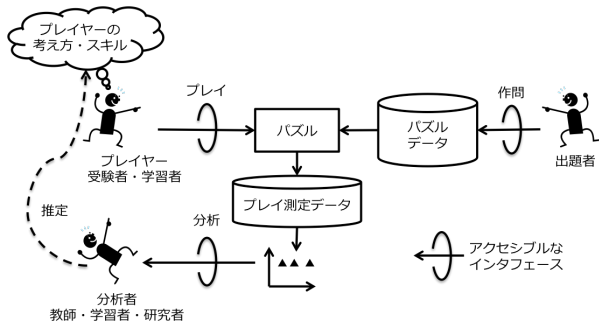


図 3 ジグソー・テキストの学習分析における 3つのアクセシビリティ

(図 3)。

パズルのアクセシビリティとは、プレイヤーがパズルをプレイ (操作) できることである。これまでは、スクリーンリーダーによって、画面に表示されているテキストは読み上げられるものの、ドラッグ&ドロップできず、並べ替え操作ができない。

測定データや分析のアクセシビリティとは、教師・学習者・研究者が、さまざまに可視化した傾向情報にアクセスできることである。共起行列や最終結果の分布表は HTML の table 要素である。ピース操作の時系列散布図や共起頻度のヒストグラムは、Scalable Vector Graphics (SVG) でタグ付けされたグラフィックである。傾向を端的につかむためのアプリであるが、見えなければ伝わらないだろう。このアクセシビリティが向上することは、例えば、障害者が学習分析の研究者として活躍するチャンスを広げるだろう。

#### 4. 目的と目標

学習分析と CBT をアクセシブルなものとするために、ジグソー・テキストのマルチモーダル化を進め、アクセシビリティを高める。

今回はパズルのプレイについて、スクリーンリーダーで操作できる UI を試作することを目標とする。その過程で気付いた課題があれば提示することとする。

対象をスマホとし、iPhone の VoiceOver[10]、Android の Talkback を使って、開発者自身が目をつぶって操作で

きることを確かめる。スマホのスクリーンリーダーはできることが限られ、初心者である筆者が操作を覚えやすいからである。必要があれば、PC のスクリーンリーダー、例えば NVDA で動作を確認する。

新たな並べ替え UI でも操作を測定できることも確認し、ドラッグ&ドロップ UI の場合と比較考察する。

なお、既に現時点で読み間違いがあると分かっているが、スクリーンリーダーによる読み間違いの改善には、今回は取り組まない。

#### 5. アプローチ

ドラッグ&ドロップとは別の UI を新たに実装して、スクリーンリーダーで操作できるようにする。文献 [11] の実装例に倣う。この方が技術的に容易かつ安定すると予想されるからである。ドラッグ&ドロップのままスクリーンリーダーで操作できるようにするアプローチを取らない。

引き続き Web 技術を使った Web アプリケーションとする。それが実現可能であれば、その方がクロスプラットフォームでの運用が容易だからである。企業などでのアドホックなライティング演習の場で、ネイティブアプリのインストールから始めるのはハードルが高い。

WAI-ARIA を採用すれば、WAI-ARIA ガイドラインが忠告するように、わずかな設定でスクリーンリーダーに対応できると見積もれる。ジグソー・テキストが採用している jQuery UI や jQuery Mobile といった JavaScript ライブラリが、WAI-ARIA に沿って実装されている。

#### 6. 結果

「移動するピースと移動先を順に選択する」という並べ替え UI を実装できた。パズルデータの読み込み中や、並べ替えの途中といった状態について、WAI-ARIA に沿って aria 属性を設定した。その結果、画面を見ずに並べ替え操作できた。測定データは、ドラッグ&ドロップによるものと同じで、ジグソー・テキスト分析アプリで分析できる。測定データを見ると、移動するピース選択 (ドラッグ開始) から移動終了 (ドロップ) までの間の時間が、ドラッグ&ドロップよりも長い。

##### 6.1 ドラッグ&ドロップに替わる並べ替え UI

この UI では、次のように並べ替え操作する。

まず、移動するピースを選択する。選択したピースの先頭に「移動するピース:」というテキストが挿入される。「移動するピース オレオレ詐欺には大きく 2 つの…」などと読み上げられる。また、各ピースの間に、「移動先 N」 (N は数字) というテキストが挿入される (図 4)。移動するピースを選択した状態では、全体としては、「…移動先 3、移動するピース、オレオレ詐欺には大きく 2 つの…、移動先 4…」のように読み上げられる。「移動先 N」と数字を挿入

するのは、次の段階の VUI を想定して、移動先を音声で区別できるようにするためである。

次に、移動先を選択する。ピースの前後に配置した「移動先」というラベルの付いたテキストを選択する。すると、移動するピースが、その移動先の直後に移動し、その後、「移動するピース:」や「移動先 N」というテキストは削除される。「移動を中止する」を選択すると、移動を中止する。

### 6.2 並べ替え状態と読み上げ: WAI-ARIA 属性

動的に変わるコンテンツをスクリーンリーダーに対応させるために、WAI-ARIA の属性を設定した。並べ替えられるエリア、すなわち「パズル」という見出しとピースが配置された div 要素に、aria-live、aria-atomic、aria-busy を設定した。このエリアは、最初にパズルの内容がネットから AJAX によって動的に読み込まれるエリアでもある。

並べ替え操作は、詳しくは次のように処理される。

ピースの移動が始まると、まず aria-busy 属性を true に設定し、「移動するピース:」や「移動先 N」といったテキストが動的に挿入される。その後、「パズル」という見出しからエリア全体が読み上げられる。そのとき、元々あったピースや「移動するピース:」や「移動先 N」も読み上げられる。

移動が終了すると、それら追加されたテキストを削除し、aria-busy 属性を false に設定する。それぞれのピース移動が終了したときに、エリア全体が「パズル」という見出しから読み上げられ、移動後のピース全体を把握できる。これが、パズルが完了するまで繰り返される。

このように、読み上げそのものについてはロジックを記述しなくても、その時々状態を属性に設定するだけで、適切に読み上げられた。

また、tabindex 属性を設定することで、後で述べるインライン表示であっても、スクリーンリーダーの「左右にスワイプ」するジェスチャーで、前後に進んで、ピースや「移動先 N」などを選択できる。

なお、PC では、このようには読み上げられなかった。あるピースの移動が始まると、新たに挿入された「移動先 N」などが読み上げられるが、元々のピースは読み上げられない。

### 6.3 測定データと分析

現時点で筆者=開発者のデータしかないが、測定データから、並べ替え操作を時系列散布図すると、ドラッグ&ドロップによる並べ替え操作のものが図 5、対象と移動先を順に選択する UI でのものが図 6 である。縦軸が移動対象のピースの ID、横軸が時間である。この 2 つを比較することは、図 2 で、同じパズルデータ、異なる UI(モード)で、同じプレイヤーによるプレイを比較する例となる。

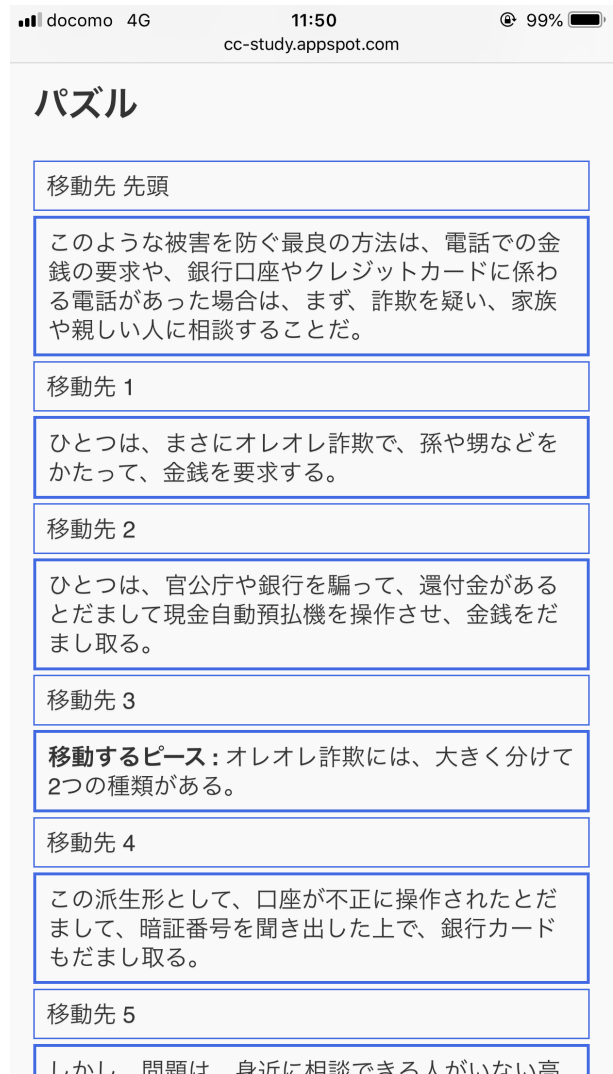


図 4 移動するピースを選択し、次に移動先を選択するという UI

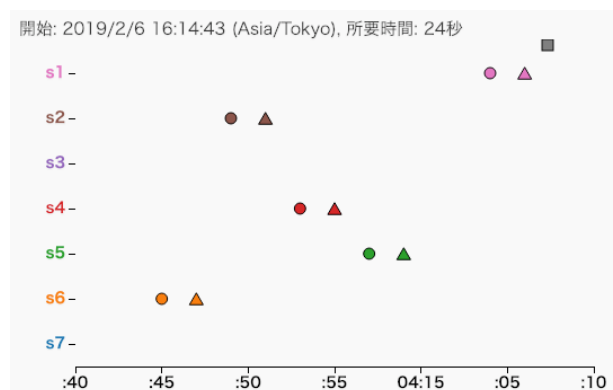


図 5 ドラッグ&ドロップによる並べ替え操作の時系列散布図

## 7. 考察

目標を達成したか、すなわち、スクリーンリーダーによる読み上げで並べ替え操作できたと言えるのか、ドラッグ&ドロップと同様に測定・分析できるのかを考察する。また、気付いた問題点を示す。



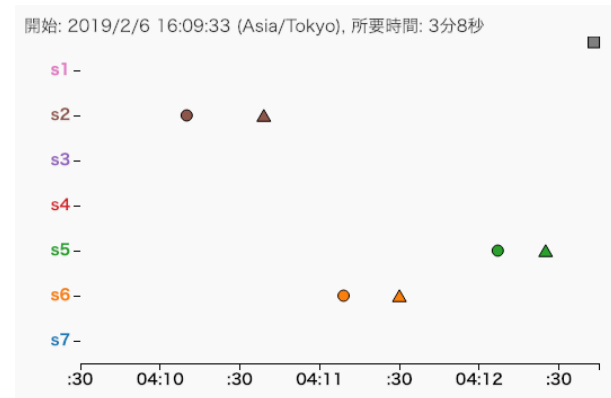


図 6 対象と移動先を順に選択する UI での並べ替え操作の時系列散布図

### 7.1 並べ替え操作と読み上げ

スクリーンリーダーに対応した並べ替え操作を実装できた。実際にスクリーンリーダーを使って、画面を見ずに操作できた。ドラッグ&ドロップの代替操作の実装例が既にあったこともあり、WAI-ARIA といった仕組みを利用することで、実装は容易であった。

PC では、目標通りに読み上げられなかったことについては、その理由が、実装が不十分だからなのか、スクリーンリーダーの問題なのか不明である。

### 7.2 並べ替え操作の測定・分析

図 5 と図 6 は同じ幅であるが、全体の操作時間が、ドラッグ&ドロップの場合 24 秒なのに対し、今回の UI の方が 3 分 8 秒と長い。図が同じ幅であるから、今回の UI では、移動対象のピースを選んでから移動先を選ぶまでの時間が、ドラッグ&ドロップに比べて長いことが分かる。この違いは、プレイヤーがパズル自体には慣れていることから、スクリーンリーダーの操作に不慣れなことが理由であろう。

他のデータも含めて、すべて筆者=開発者によるものにはあるが、どのピースの次にどのピースを動かすかといった手順については、違いを見いだせなかった。2つのモードの測定データを比較して見いだせることについて、UI への習熟度の差と、思考内容の差というものの違いの、シンプルな例と言えよう。

例えば、ピースを 1 つずつ前後に移動させられる UI も考えられる。プレイヤーは、移動するピースを選択して、1 つ前 (上) または後ろ (下) への移動させる。これを、意図した位置に達するまで繰り返す。シンプルな UI なので、実装もプレイも可能であろう。このような操作を測定して、前節と同様に「移動するピースを選択して、移動先へ移動させるまでの時間」を比較することを考える。このためには、連続する「1 つ前後への移動」をどこかで区切って、ピース移動の終了を判定する前処理が必要となる。

今回の UI は、このような前処理が不要な実装であった。

## パズル

```

if (n % i == 0) System.out.print(i + " ");
System.out.println("");
}
for (int i = 2; i <= n/2; i++) {

```

図 7 インデントした行を含むジグソー・コードのパズル

すなわち、ドラッグ&ドロップと同様に測定・分析できると言ってもよい。

### 7.3 問題点

気がついた点を報告する。

#### 7.3.1 空白文字、字下げ、インデント

iOS の Safari や Chrome では、VoiceOver は空白文字を読み上げない。パズル (問題) で、空白文字が意味を持つような場合は注意が必要であろう。例えば、プログラムコードの場合である。

図 7 は、ジグソー・コードのパズルである。プログラムの行がパズルのピースで、ピースがランダムに並べられてプレイヤーに提示される。この図で、1 行目の「if」の前のインデントは大きなヒントになっている。4 行のコードは「2 以上の整数  $n$  を引数に受け取って 1 以外の正の約数すべてを出力し、最後に改行を出力する」という Java のメソッドの中身である。このパズルは、「1 以外の正の約数」の意味や判定の仕方や Java の詳しい文法を知らなくても、プログラミング言語の一般的な構文、すなわちカッコやインデントの使われ方を知っていれば、正解にたどり着ける。

「if」ピースのインデントは、4 文字の空白文字「」が挿入されて実現されている。HTML の DOM データ上でも空白が 4 文字入っている。iOS の VoiceOver は、それを読み上げない。これは、「句読点および記号」をすべて読み上げる詳細度に設定しても変わらない。

スクリーンリーダーに 1 文字ずつ読み上げさせることもできる。iOS の VoiceOver は、「if」の前の 4 つの空白文字を読み上げないが、「if」と「(」の間の空白文字を読み上げる。

Windows の NVDA は、「句読点/記号読み辞書」で、空白文字の「最低読み上げレベル」を「文字」以外に設定し、「句読点/記号レベル」をそのレベル以上に設定すると、読み上げられる。Windows の NVDA は、空白文字の最低読み上げレベルを文字以外に設定していると、「よんすべーすいふすべーすかつこえぬ…」と読み上げる。

#### 7.3.2 インライン表示

ピース全体で 1 つの段落を構成するように表示するイン

## パズル

**移動先 先頭** オレオレ詐欺には、大きく分けて2つの種類がある。**移動先 1** このような被害を防ぐ最良の方法は、電話での金銭の要求や、銀行口座やクレジットカードに係わる電話があった場合は、まず、詐欺を疑い、家族や親しい人に相談することだ。**移動先 2 移動するピース**:ひとつは、まさにオレオレ詐欺で、孫や甥などをかたって、金銭を要求する。**移動先 3** ひとつは、官公庁や銀行を騙って、還付金があるとだまして現金自動預払機を操作させ、金銭をだまし取る。**移動先 4** この派生形として、口座が不正に操作されたとだまして、暗証番号を聞き出した上で、銀行カードもだまし取る。**移動先 5** しかし、問題は、身近に相談できる人がいない高齢者が多くいることにあるのかもしれない。**移動先 6** 相変わらず、オレオレ詐欺の被害が減らない。**移動先 最後尾**

図 8 ピースをインライン表示して並べ替える UI

ライン表示モードも併せて実装した。並べ替え操作は、移動対象のピースを選択し、移動先を選択するという、同様のものである。ドラッグ&ドロップでは、インライン表示で並べ替えるのは見づらいが、今回の UI では問題ない。

インライン表示でも、スクリーンリーダーにピースを個別に認識させて、ジェスチャーで前後に進むことができる。しかし、操作の最中に、ピースがインライン表示されてるのか、個別の段落のようにブロック表示されてるのかの区別がつかない。

パズル「オレオレ詐欺」では、「オレオレ詐欺には…2つある」、「ひとつは…」、「ひとつは…」がセットで動かされる傾向があると、ドラッグ&ドロップの UI について報告した [15]。図 4 のようなブロック表示では、先頭の「ひとつは」が目立つが、インライン表示ではそれほど目立たなく見える。この違いが、パズルの難易度に影響するかもしれない。

## 8. まとめ

ジグソー・テキストに、移動するピースと移動先を順に選択するという新たな並べ替え UI を開発して、スマートフォンのスクリーンリーダーを使って並べ替え操作できた。また、並べ替え操作を、ドラッグ&ドロップの UI と同様に測定・分析できた。スマートフォンのスクリーンリーダーでは、パズル中の空白文字の扱われ方に注意が必要と分かった。

## 9. 残された課題

課題は多い、分からないことだらけである。障害のある学習者でも操作できるかの検証、開発者以外の多数のプレイヤーによる検証、スクリーンリーダーとデバイスを組み合わせた使われ方の調査と、その組み合わせでの検証、スクリーンリーダーを使ったプログラミングの現状の調査、様々なパズル (問題) を使った検証、などが必要である。

キーボードやタッチスクリーンを全く使わず、VUI によって並べ替えるジグソー・テキストを開発すると、また新たな知見が得られるだろう。

ジグソー・テキスト分析のアクセシビリティ、すなわち図 3 の「分析」のアクセシビリティが手つかずである。

謝辞 本研究は JSPS 科研費 17K01085 の助成を受けたものです。

## 参考文献

- [1] 角田 博保, 久野 靖, 短冊型問題: プログラミングの技能を評価可能な試験出題形式, 夏のプログラミング・シンポジウム 2016「教育・学習」(2016)
- [2] 西田 知博, 植原 啓介, 角谷 良彦, 鈴木 貢, 中山 泰一, 香西省治, 高橋 尚子, 中西 通雄, 松浦 敏雄, 増澤 利光, 萩谷 昌己, 萩原 兼一, 「情報科」大学入学者選抜における CBT システムの研究開発, 情報教育シンポジウム論文集, 2017(28), 182-187 (2017-08-10)
- [3] 大学入試センター, 教科「情報」における CBT を活用した試験の開発に向けた問題素案の募集について, 2018-07-17 <https://www.dnc.ac.jp/news/20180717-01.html>
- [4] 松原 聡, 山口 翔, 岡山 将也, 池田 敬二, デジタル教科書プラットフォームの検討—アクセシビリティを中心に—, 東洋大学経済研究会 経済論集, 38(2), 185-199 (2013-03)
- [5] 喜多 敏博, 長岡 千香子, 平岡 齊士, スマートスピーカーを通じた LMS 上での学習活動, 情報処理学会 研究報告 教育学習支援情報システム (CLE), 2018-CLE-26(16), 1-5 (2018-11-30), 2188-8620
- [6] 三吉秀夫, マルチモーダルインタフェース, シヤープ技報 77, 43-47, 2000
- [7] 松居 辰則, ラーニングアナリティクス: 4. マルチモーダルラーニングアナリティクス, 情報処理, 59(9), 810-814 (2018-08-15)
- [8] Accessible Rich Internet Applications (WAI-ARIA) 1.1 <https://www.w3.org/TR/wai-aria-1.1/>
- [9] 村田 真, 電子書籍フォーマット EPUB と日本語組版 日本でメインストリームにいる人間は国際標準化の舞台ではまず勝てない, 情報管理, 2012, 55 巻, 1 号, p. 13-20, 公開日 2012/04/01
- [10] VoiceOver, iPhone ユーザガイド, <https://support.apple.com/ja-jp/guide/iphone/iph3e2e415f/12.0/ios/12.0>
- [11] Gez Lemon, Accessible Drag and Drop Using WAI-ARIA, Dev.Opera, 2009-07-08 <https://dev.opera.com/articles/accessible-drag-and-drop/>
- [12] Web Content Accessibility Guidelines (WCAG) 2.1 <https://www.w3.org/TR/WCAG21/>
- [13] 藤本 光史, スライドパズルにおける回転型操作とアクセシビリティ, 情報処理学会 研究報告アクセシビリティ (AAC), 2017-AAC-5(3), 1-6 (2017-12-01), 2432-2431

- [14] 川北紘正, 大場みち子, 山口琢, プログラミング思考過程に基づくプログラミング時の行動分析と傾向, 情報処理学会 第 81 回全国大会, 2019
- [15] 山口 琢, 大場 みち子, 高橋 慈子, 小林 龍生, ジグソー・テキストによる文並べ替え操作の測定, 情報処理学会 研究報告コンピュータと教育 (CE), 2017-CE-142(27), 1-6 (2017-12-01), 2188-8930