

アニメーションおよびソースコード表示を活用した ソートアルゴリズム学習アプリケーションの開発

渡辺大智[†] 石井幹大[‡] 伊藤一成[†]

[†] 青山学院大学社会情報学部 〒252-5258 神奈川県相模原市中央区淵野辺 5 丁目 10-1

[‡] 青山学院大学理工学部 〒252-5258 神奈川県相模原市中央区淵野辺 5 丁目 10-1

あらまし 我々は、ソートアルゴリズムを学習できる Web アプリケーション「人型ピクソートグラム」を提案している。人型ピクソートグラムでは、人型ピクソートグラムの体重をシーソーで比較する過程をアニメーションで再生できることが特徴の 1 つである。我々の先行研究における評価実験では、ソートアルゴリズムをアニメーションで解釈する点において有効性が認められたが、ソースコードによるソートアルゴリズムの学習が不十分であると懸念された。大学入学共通テストでは、2025 年度入試から教科「情報」が導入される。プログラミング分野では、Python 言語をベースとした共通テスト用プログラム表記が用いられる予定であり、共通テスト用プログラム表記のソースコードによるソートアルゴリズムの学習に対する需要があることが判明した。今回、人型ピクソートグラムにおいて、アニメーションに対応したソースコードを表示する機能、および動作中のソースコードの行をハイライトする機能を追加したので、本稿で報告する。

キーワード ソートアルゴリズム, 人型ピクソートグラム, 大学入学共通テスト, 共通テスト用プログラム表記, Python, 情報 I

1. はじめに

高等学校では、2022 年度から情報科において「情報 I」が必修教科目として運用されている[1]。情報 I の学習単元の一つである「コンピュータとプログラミング」では、Python, JavaScript, VBA, Scratch の利用が主に想定されている[2]。

さらに、大学入学共通テストでは、2025 年度入試から教科「情報」が導入される。プログラミングの問題では、Python 言語をベースとした共通テスト用プログラム表記が擬似言語仕様として用いられる予定であり、すでにサンプル問題や試作問題が公開されている[3][4]。

情報 I のコンピュータとプログラミングでは、代表的なアルゴリズムとして、ソートアルゴリズムが示されている。また、高等学校情報科「情報 I」教員研修用教材では、『問題の解決には様々な方法があり、それらを比較検討し、最善の方法を見つけることの重要性について理解させる』ための例として、ソートアルゴリズムが例示されている[2]。

プログラミング言語を使ってアルゴリズムを教える場合、プログラミング言語の基礎が不十分な生徒には理解が難しい場合がある。そのため、ソートアルゴリズムに関しても、プログラミング言語やコンピュータを用いない CS アンブレラグラムの技法を用いた教授法も提案されている[5][6]。

ピクソートグラムはコンテンツ表現の抽象度の高さから、それを見た人物が自分自身や本人に関わる人物事物などを想起させる効果があると言われている。有名な「非常口」ピクソートグラムのデザインの策定に関わった太田は“走る人型を囲む空間が見る人を包む空間とつながって走る人は見る人の投影になる”と述べ[7]、実際に避難中の人々が、出口へ向かって走る人型ピクソートグラムを見たときに、いかに自身をピクソートグラムに投影し、同一視するか、という観点にデザインの労力が払われた。

このピクソートグラムの特徴に着目し、我々の研究グループでは、情報デザインやプログラミングの学習[8]、データ活用に関する学習[9]など、各種学習用アプリケーションを開発している。ソートアルゴリズムの学習についても、アプリケーション「人型ピクソートグラム」¹ (以下、ピクソート) をインターネット上で公開している[10]。CS アンブレラグラドでは、天秤と重りを使い、重りを重さ順に並べる過程でソートアルゴリズムを学ぶ手法が提案されているが、ピクソートでは、重りを人型ピクソートグラム、天秤をシーソーに変更していることが特徴である。

本アプリケーションを用いて高校生を対象にした評価実験を行なった結果、ソートアルゴリズムをアニメーションで解釈する点において有効性が認められた[10]が、ソースコードによるソートアルゴリズムの学習が不十分であると懸念する声も聞かれた。

¹以下の URL から利用可能である。

<https://pictogramming.org/apps/humanpicsortgram/>

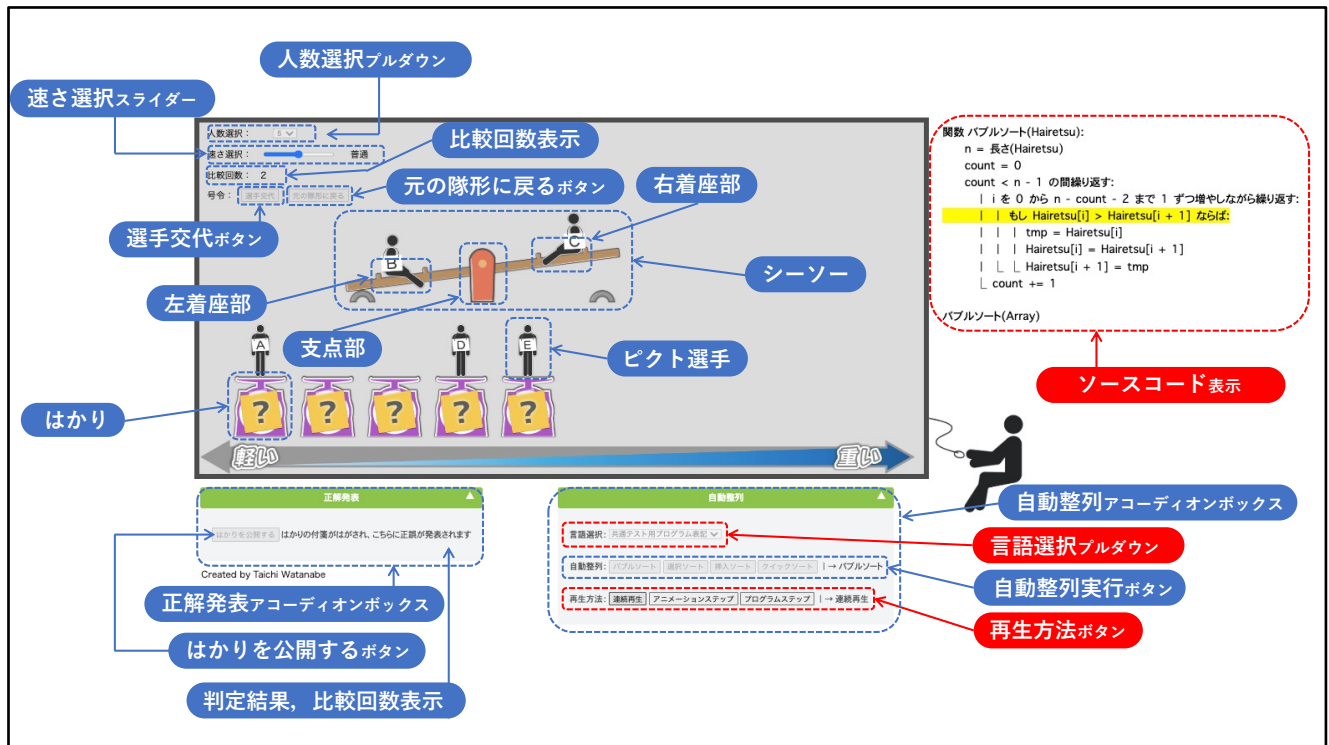


図1 「ピクソート」のスクリーンショット

大学入学共通テストのプログラム領域では、共通テスト用プログラム表記を用いた出題が予定されている。そのため、アニメーションによる学習に止まらず、共通テスト用プログラム表記のソースコードによるソートアルゴリズムの学習に対する需要があることが判明した。そこで、既存機能であるアニメーションに対応した、ソースコードを表示し、動作中のソースコードの行をハイライトする機能を追加したので報告する。

以下、2章でアプリケーションの概要を示し、3章では今回追加した機能について示す。4章ではまとめと今後の展望について述べる。

2. 人型ピクソートグラム

2.1 概要

「ピクソート」のスクリーンショットを図1に示す。図1中に赤色で示した部分は今回追加した機能であり、3章で解説する。ピクソートには、手動で人型ピクトグラム（以下、ピクト選手）を整列する「手動整列機能」と、自動でピクト選手を整列する過程をアニメーションで提示する「自動整列機能」を実装している。2.2節および2.3節にて、それぞれの機能について解説する。なお、ピクソートは、PC端末およびタブレット端末からの利用が可能である。

2.2 手動整列機能の操作手順

手動整列機能の操作は、「(1) 整列するピクト選手の人

数の決定」、「(2) シーソーによる重さの比較」、「(3) 並び替え」、「(4) 整列判定」の順に行なう。2.2.1項から2.2.4項でそれぞれの手順について示す。

2.2.1 (1) 整列するピクト選手の人数の決定

ピクソートの利用者（以下、利用者）が図1画面左上の「人数選択プルダウン」から人数を選択すると、人数分のはかりとピクト選手が横一列に表示される。人数は、「2」から「9」までの1刻みの値から選択する。人間のはかりの上に乗っているイメージを利用者に持たせるため、ピクト選手ははかりの直上に表示される。ピクト選手には一意の体重が設定され、その体重差をもとに整列が行なわれる。また、ピクト選手は一意のアルファベット1文字が書かれたプラカードを首に掛けており、ピクト選手それぞれを別の個体として識別できるようになっている。なお、人数選択プルダウンで人数を変更すると、現在表示されているピクト選手は画面外に退場し、新たに選択された値に応じてピクト選手が再度入場する。

2.2.2 (2) シーソーによる重さの比較

図1中のピクト選手とシーソーを用いて重さを比較する。利用者は、体重を比較したいピクト選手を選択し、シーソーのいずれかの着座部までドラッグアンドドロップ操作を行なうことで、ピクト選手をシーソーに着座させることができる。以降、シーソー左側の着座部を左着座部、シーソー右側の着座部を右着座部と呼称する。また、着座部にピクト選手が着座した状態を着座状態、着座部にピクト選手

が着座していない状態を空席状態とし、左着座部と右着座部がともに着座状態である状態を比較状態とする。

着座の際、ピクト選手の表示は直立状態から着座状態に遷移する。ピクト選手が着座すると、ピクト選手の体重に応じてシーソーが傾く。なお、支点部からの長さに関する思案を排除するため、ピクト選手の着座位置は左右それぞれで固定されている。また、着座しているピクト選手を着座部以外の場所へドロップすると、ピクト選手は直立状態に遷移する。このように、直立状態と着座状態を切り替えることで、ピクト選手の状態把握を容易に行なうことを可能としている。

ソートアルゴリズムの特徴である1対1の比較を再現するため、ピクト選手の体重の比較は1体対1体のみで行なう。2体対1体や2体対2体などの一方が多人数、また両端が多人数での比較はできない。そのため、着座状態の着座部に他のピクト選手を着座させることはできず、着座を試みたピクト選手は、乗っていたはかりへ自動で戻される。

図1の左上にある「比較回数」は、比較状態に遷移する度に「1」増加し、人数選択プルダウンの数値を変更する度に「0」となる。比較回数の表示のほか、画面左上には「選手交代ボタン」と「元の隊形に戻るボタン」も配置している。「選手交代ボタン」は、表示中のピクト選手を画面外へ退場させてピクト選手の選択人数は固定のまま体重値の異なるピクト選手を呼ぶ機能を有する。「元の隊形に戻るボタン」を押下すると、表示中のピクト選手がアルファベットの昇順ではかりの上に整列する。整列の際、ピクト選手に体重として設定されている数値は変更しない。「元の隊形に戻るボタン」の機能により、自動整列を見た後に、手動で同じようにソートの手順を模倣することができるため、ソートの理解を促進させることが期待される。さらに、自動整列のアニメーション中は「選手交代ボタン」と「元の隊形に戻るボタン」を無効化している。これにより、ソートの手順の一連の流れを最後まで見ることを促し、自動整列を中断することによるソート理解の誤りを防ぐことを期待している。左右どちらの号令ボタンを押しても、比較回数を「0」にリセットする。

2.2.3 (3) 並び替え

利用者は、ドラッグアンドドロップ操作により、はかり上のピクト選手を並べ替えることができる。「はかりにピクト選手を乗せ、並び替えた」ことを利用者に認識させるため、はかり周辺の一定の領域内でピクト選手をドロップすると、ピクト選手は当該のはかり上中央に自動配置される。

2.2.4 (4) 整列判定

全てのピクト選手がはかり上に配置されると、「はかり

を公開するボタン」を押下できるようになる。「はかりを公開するボタン」は、「正解発表アコーディオンボックス」を開くことで表示される。

「はかりを公開するボタン」を押下すると、はかりの付箋が全て剥がされてピクト選手の体重が公開され、ピクト選手が体重の昇順に整列できているかどうかの正誤判定、および比較に要した回数（以下、最終比較回数）が表示される。「はかりを公開するボタン」を押下する前後のはかりの表示例を図2に、正誤判定および最終比較回数の表示例を図3に、それぞれ示す。「はかりを公開するボタン」の押下後は、ピクト選手への操作を不能とし、ピクト選手の並び替えを終了する。

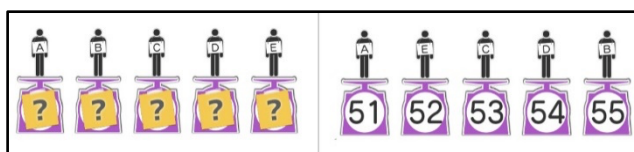


図2 はかりの表示例（左：体重公開前／右：体重公開後）



図3 正誤判定および比較回数の表示例
（左：正解の場合／右：不正解の場合）

2.3 自動整列機能

利用者が「自動整列アコーディオンボックス」を開き、バブルソート、選択ソート、挿入ソート、クイックソートのいずれかのボタンを押下すると、自動整列機能が実行される。自動整列機能では、ソートアルゴリズムに則ったアルゴリズムをアニメーション（以下、アルゴリズムアニメーション）で確認することができる。また、利用者が押下したソートアルゴリズムのボタン名を「自動整列アコーディオンボックス」内の右下に表示し、どのソートアルゴリズムを実行しているか、利用者に認識させる。自動整列機能実行時のスクリーンショットを図4に示し、各ソートアルゴリズムの説明を2.3.1項から2.3.4項で示す。

自動整列機能実行時は、アプリケーション画面の下部に表示した2つのアコーディオンボックスを除く操作範囲が、テレビ画面を模した枠（以下、テレビ画面）で囲まれる。テレビ画面の右側には、ソート対象に含まない人型ピクトグラム1体（以下、ピクト監督）を新たに配置し、ピクト監督がテレビ画面内のソート操作を行なっているかのように表示する。1章で述べたピクトグラムの特徴を利用し、ピクト監督を見た利用者が、自身をピクト監督に投影し、ピクト監督を模倣してソート操作を行なう効果を期待している。

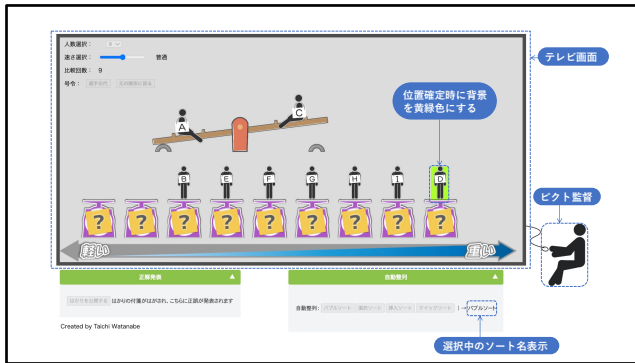


図4 自動整列機能実行時のスクリーンショット

さらに、アルゴリズムアニメーションによるソートアルゴリズムを利用者がより注意深く観察できるように、自動整列機能には2つの工夫を施した。1つ目は、はかりに乗せる位置が確定したピクト選手の背後色を黄緑色に変更することであり、ソートアルゴリズムによる整列決定順序を視覚的に認識させる。2つ目は、図1画面左上の「速さ選択スライダー」によってアルゴリズムアニメーションの速度を変更できることであり、利用者が観察したい速度でアルゴリズムアニメーションを観察することができる。アルゴリズムアニメーションの速度は「遅い、普通、速い」の3段階から選択可能であり、アルゴリズムアニメーションが実行中である場合も速度を変更することができる。

なお、ピクト選手に対して何らかの操作を行なった後に自動整列機能を実行すると、ピクト選手を元の隊形に戻し、比較回数を0回に戻した上で、自動整列機能が実行される。また、アルゴリズムアニメーションへの集中を促すため、自動整列機能実行時および終了後は、ピクト選手に対する操作および新たなアルゴリズムアニメーションの開始を不能としている。自動整列機能実行終了後は、画面左上の選手交代ボタンまたは元の隊形に戻るボタンの押下、あるいは人数選択プルダウンで新たな数値を選択することで、全ての操作が可能となる。

2.3.1 バブルソートによる自動整列機能

体重が重いピクト選手から順に、ソート順を決定する。隣り合う2体のピクト選手を順番に比較し、比較対象の2体のピクト選手の体重差から昇順にソートする。ソートを繰り返して最も体重の重いピクト選手を決定した後、決定したピクト選手を除いて同様にソートを繰り返し、ソート順を決定する。

2.3.2 選択ソートによる自動整列機能

バブルソートによる自動整列機能で説明したアルゴリズムと同様、体重の値が最も大きいピクト選手から決定するアルゴリズムである。バブルソートとの相違点は、確定したピクト選手を除いた横一列に並んだピクト選手の中で最

も体重の重いピクト選手が確定した時に、最も右に位置するピクト選手と交換する点にある。この手順を繰り返し、ソート順を決定する。

2.3.3 挿入ソートによる自動整列機能

隣り合う2人のピクト選手を比較し、比較対象の2体のピクト選手を体重差から昇順にソートする。ソート後、比較した2体のピクト選手のうち、体重の軽いピクト選手の左隣に別のピクト選手が存在する場合は、当該2体のピクト選手同士を続けて比較し、ソートする。この手順を繰り返すことで、ソート順を決定する。

2.3.4 クイックソートによる自動整列機能

中央に配置されたピクト選手（ただし、ピクト選手の人数が偶数の場合には、中央右側のピクト選手）を基準として、他のピクト選手を基準のピクト選手より重いグループと軽いグループに分類する。分類結果に基づき、基準のピクト選手のソート位置を決定する。続いて、分類したグループ内で同様に比較を繰り返すことで、ソート順を決定する。

2.4 自動整列ボタン非表示機能

自動整列ボタン非表示機能は、授業での活用を想定した機能であり、URLのクエリパラメータに「?hide」の文字列を追加することで、利用することができる。自動整列ボタン非表示機能では、「自動整列アコーディオンボックス」が表示されなくなり、手動整列機能のみが利用可能となる。

3. ソースコード表示機能

3.1 機能説明

自動整列機能に、選択したソートアルゴリズムのソースコードが表示され、リアルタイムでアルゴリズムアニメーションと対比して該当するソースコードの行をハイライトする機能（以下、ソースコード表示機能）を追加した。

2章で示した図1は、ソースコード表示機能にて、「言語選択：共通テスト用プログラム表記」、「自動整列：バブルソート」、「再生方法：連続再生」を選択した際のスクリーンショットである。本節では、図1において赤色で示した部分を解説する。ソースコード表示機能は、アルゴリズムアニメーションの右側に各ソートに対応するソースコードを表示する機能である。この機能は、URLのクエリパラメータに「?program」の文字列を追加することで、ソースコード表示機能に切り替えられる。ソースコード表示機能に切り替えると、「自動整列アコーディオンボックス」に表示言語を選択できるプルダウン（以下、言語選択プルダウン）と、再生方法に関するボタン（以下、再生方

法ボタン) が追加で表示される。再生方法ボタンは、「連続再生ボタン」、「アニメーションステップボタン」、「プログラムステップボタン」の3つである。連続再生ボタンを押下すると、一連のアルゴリズムアニメーションとソースコードのハイライトが実行される。一方、アニメーションステップボタンを押下すると、一定の動作ごとに区切ったアニメーション再生が、プログラムステップボタンを押下すると、ソースコード1行ずつの再生が、それぞれ実行される。言語選択プルダウンでは、共通テスト用プログラム表記あるいは Python 言語のいずれかを選択することができ、選択した言語でソースコードが表示される。選択言語の初期設定は、共通テスト用プログラム表記である。バブルソートを共通テスト用プログラム表記で表示したソースコードの表示例を図5に、Python 言語で表示したソースコードを図6に、それぞれ示す。

```

1 関数 バブルソート(Hairetsu):
2  n = 長さ(Hairetsu)
3  count = 0
4  count < n - 1 の間繰り返す:
5  | i を 0 から n - count - 2 まで 1 ずつ増やしながら繰り返す:
6  | | もし Hairetsu[i] > Hairetsu[i + 1] ならば:
7  | | | tmp = Hairetsu[i]
8  | | | Hairetsu[i] = Hairetsu[i + 1]
9  | | | Hairetsu[i + 1] = tmp
10 | | count += 1
11
12 バブルソート(Array)

```

図5 バブルソートを共通テスト用プログラム表記で表示したソースコード表示例

```

1 def bubble_sort(Hairetsu):
2     n = len(Hairetsu)
3     count = 0
4     while count < n - 1:
5         for i in range(n - count - 1):
6             if Hairetsu[i] > Hairetsu[i + 1]:
7                 tmp = Hairetsu[i]
8                 Hairetsu[i] = Hairetsu[i + 1]
9                 Hairetsu[i + 1] = tmp
10        count += 1
11
12 bubble_sort(Array)

```

図6 バブルソートを Python 言語で表示したソースコード表示例

操作手順を説明する。いずれかのソートアルゴリズムのボタンを押下すると、選択した言語に該当するソートアルゴリズムのソースコードが、アルゴリズムアニメーションの右側に表示される。連続再生ボタンを押下することで、アルゴリズムアニメーションと連動して、ソースコードのハイライトが最後まで実行される。これにより、アルゴリズムアニメーションの動作をどのようなソースコードで動作しているのかが見ることができる。アニメーションステップボタンを押下すると、アルゴリズムアニメーションでピクト選手の比較を一度だけ行なう処理が実行され、再生方法ボタンのいずれかが再度押されるまでその時点で停止する。プログラムステップボタンを押下すると、表示したソースコードのハイライトした1行の動作を行なう処理まで実行され、アニメーションステップボタンと同様、再生方

法ボタンのいずれかが再度押下されるまでその時点で停止する。連続再生の実行中に、アニメーションステップボタン、及びプログラムステップボタンを押下でき、連続再生中に再生方法を切り替えることができる。アニメーションステップ、及び、プログラムステップ実行中は他の再生方法ボタンを無効化している。

3.2 ソースコード表示機能の有効性の考察

本アプリケーションを利用した先行研究[10]では、アルゴリズムアニメーションにより自らの言葉でソートアルゴリズムの説明ができるようになった、という評価が得られている。今回、ソースコード表示機能を追加したことにより、アルゴリズムアニメーションとソースコードを対比しながらソートアルゴリズムの学習が可能になったことで、ソートアルゴリズムを体系立てて解釈できる効果が期待できる。

さらに、ソースコードの視覚的な確認が可能となったことで、利用者の目的や理解度に応じて、本アプリケーションを利用できるようになったと考えている。例えば、ソートアルゴリズムの学習を目的とした利用者は、自動整列機能、手動整列機能、ソースコード表示機能の順で段階的に学習を行なうことが可能であり、共通テスト用プログラム表記や Python 言語のソースコードを読み取ることを目的とした利用者は、ソースコード表示機能から学習を行なうことが可能である。したがって、本アプリケーションにおいて、ソースコード表示機能の必要性は高いと言える。

ソースコードの表示は、共通テスト用プログラム表記と Python 言語から選択できるように実装した。Python 言語を採用した理由は2つある。1つ目は、Python 言語と共通テスト用プログラム表記の相違点が少ないからである。共通テスト用プログラム表記は、Python 言語をベースに策定されているため、両者の比較を容易に行なうことができる。そのため、共通テスト用プログラム表記の学習後、Python 言語での学習に移行することも容易であると考えられ、本アプリケーションの利用を契機として、非擬似言語でのソートアルゴリズムの学習に繋がることを期待している。2つ目は、大学入学共通テストを目的としない利用者に対しても、ソースコードによるソートアルゴリズムの学習を促進したいからである。幅広い利用者層に活用されることを期待し、非擬似言語である Python 言語を採用した。

授業で学習するプログラミング言語によっては共通テスト用プログラム表記との相違が影響を及ぼすことが懸念されるが、井手は、情報Iで主に用いられる4言語に関して、どの言語を用いても共通テスト用プログラム表記で記述された問題の成績には大きくは影響しないことを実践から示している[11]。そのため、情報Iで利用されている Python

言語以外の言語によるソースコード表示も追加することでさらに需要が増加すると考えられる。

4. まとめと今後の展望

本研究では、人型ピクトグラムを用いたソートアルゴリズムを学ぶアプリケーション「人型ピクソートグラム」の機能の追加を行なった。我々が行なった先行研究における評価実験では、ソートアルゴリズムをアニメーションで解釈する点において有効性が認められたが、ソースコードによるソートアルゴリズムの学習が不十分であると懸念された。また、大学入学共通テストのプログラム領域では共通テスト用プログラム表記を用いた出題が予定されている。そのため、アニメーションによる学習に止まらず、共通テスト用プログラム表記のソースコードによる、ソートアルゴリズムの学習に対する需要があることが判明した。そこで、既存機能であるアニメーションに対応した、ソースコードを表示し、動作中のソースコードの行をハイライトする、ソースコード表示機能を追加した。今後は、追加した機能を活用した学習による有効性の検証や、ソースコード表示機能における Python 言語以外の非擬似言語の追加を行なっていく予定である。

謝辞

本研究は JSPS 科研費 21H03560 および 20K03160 の助成を受けたものです。また、東京都立南多摩中等教育学校情報科教員の御家雄一先生に助言をいただきました。

参 考 文 献

- [1] 平成 30 年告示高等学校学習指導要領
http://www.mext.go.jp/component/a_menu/education/micro_detail/_icsFiles/afieldfile/2018/07/11/1384661_6_1_2.pdf (2023 年 11 月 10 日閲覧)
- [2] 高等学校情報科「情報 I」 教員研修用教材(本編)
https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1416756.htm (2023 年 11 月 10 日閲覧)
- [3] 大学入学共通テスト「情報」 サンプル問題
https://www.mext.go.jp/content/20211014-mxt_daigakuc02-000018441_9.pdf (2023 年 11 月 10 日閲覧)
- [4] 大学入学共通テスト試作問題「情報 I」
https://www.dnc.ac.jp/albums/abm.php?d=511&f=abm00003277.pdf&n=6-2-1_%E8%A9%A6%E4%BD%9C%E5%95%8F%E9%A1%8C%E3%80%8E%E6%83%85%E5%A0%B1%E2%85%A0%E3%80%8F%E2%80%BB%E4%BB%A4%E5%92%8C%E5%B9%B412%E6%9C%8823%E6%97%A5%E4%B8%80%E9%83%A8%E4%BF%AE%E6%AD%A3.pdf (2023 年 11 月 10 日閲覧)
- [5] T. C. Bell, J. Alexander, I. Freeman, and M. Grimley. Computer Science Unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology, 13(1):20–29, 2009.
- [6] 兼宗 進 (監訳) : コンピュータを使わない情報教育 アンプラグド・コンピュータ・サイエンス, イーテキスト研究所(2007).
- [7] 太田幸夫: ユニバーサル・コミュニケーションデザインの認識と実践
<https://forum8.co.jp/topic/universal107.htm> (2023 年 11 月 10 日閲覧)
- [8] 伊藤一成: 複数のプログラミング言語で記述可能なピクトグラムコンテンツ作成環境の提案と実装, 情報処理学会論文誌 TCE, Vol.7, No.3, pp.1-11 (2021)
- [9] 高橋伶奈, 御家雄一, 伊藤一成 : 人型ピクトグラムを構成素とするピクトグラフ生成アプリケーション「Human Pictograph」の実装, 2022 年度情報処理学会関西支部支部大会, (2022)
- [10] 渡辺大智, 御家雄一, 伊藤一成 : 人型ピクトグラムを用いたソートアルゴリズムを学ぶアプリケーション「人型ピクソートグラム」の拡張と評価, 第 15 回データ工学と情報マネジメントに関するフォーラム DEIM2023, (2023.03)
- [11] 井手広康: 情報 I の教科書におけるプログラミング分野の比較と考察. 情報処理学会論文誌教育とコンピュータ (TCE), Vol.7, No.3, pp. 8–18 (2022)