

行動 to Code：日常生活の複数の身体行動を通したプログラムの要素・処理の体験アプリケーション

中村幸樹^{†1} 橋田朋子^{†1}

概要：本研究では、プログラミング的思考が日常の行動の分析に役立つ事やその面白さをより多くの人に知ってもらう事を目指し、日常の行動がプログラムの要素・処理で説明できる事を直感的に体験可能とする仕組みを実現する。具体的には、ユーザの身体で完結する簡易な行動をするように自然言語で指示し、ユーザがその行動をし始めると、含まれるプログラムの要素・処理を自動検出し、インタラクティブにプログラミング言語（命令とコメント）でフィードバックする仕組み“行動 to Code”を提案する。これにより自身の行動が細分化した Code で捉え直せる事を体感できる。提案システムは iPhone、PC の内蔵センサ・カメラを利用して、それらのブラウザからアクセスできるアプリケーションとして実装する。さらに本システムを用いることで、日常の行動に含まれるプログラムの要素・処理に気づき、プログラミング的思考が身近に感じられるかを明らかにする評価実験も行う。

1. はじめに

プログラミング教育が小学校の授業に導入されるなど、今、プログラミング的思考[1]と呼ばれる、対象を理解し表現するために、どのような動きに分解して組み合わせるべきかを論理的に考える力が注目されている。筆者らはプログラミングに親しみプログラミング的思考を自然と行う中で、特にこの思考がコンピュータ内の動作に限らず日常生活に適用できることや、それにより自分自身の考えや行動が整理・客観視しやすくなることに可能性を感じている。既に、プログラミング的思考の学びの一環として、日常の身近な動作や行動をプログラムの要素・処理に分解する試みは行われている。が、このような分析は基本的には自分の頭で考えてプログラムの要素・処理を自然言語で書き出す作業にとどまっている。実際にそれらが自動的に「実行」される事や、実行の可否を示す終了ステータス、エラー・ログメッセージなどのフィードバックはないため、分解した動作の粒度や処理の種類が適切であったのかを即時に知る事はできない。つまり現状の自然言語を用いたプログラムの要素・処理の分析は、本来のプログラミングではあったはずの多くの手掛かりなしに思考を進める難易度の高い作業であり、導入としては障壁が高い。

筆者らは、プログラミング的思考が日常の行動の分析に役立つ事やその面白さをより多くの人に知ってもらう事を目指し、そのためには日常の行動がプログラムの要素・処理で説明できる事を一度でも直感的に体験する事が重要であると考えた。そのためやや変則的ではあるが、本研究ではユーザの身体で完結する簡易な行動を対象として、まずは自然言語でユーザにある行動を指示し、ユーザがその行動をし始めると、そこに含まれるプログラムの要素・処理を逐一センサで自動検出し、即時にプログラミング言語（命令とコメント）でフィードバックする手法“行動 to Code”

を提案する（図1）。この手法では、身体で完結する行動を対象にすることで、通常の実行とは異なるがユーザが自然言語を読んでその行動をする事による「ジッコウ」が可能になる。さらにインタラクティブなフィードバックにより一つの行動に思えるものがより細かい身体動作（プログラムの要素）や処理から成る事をわかりやすく示す。自然言語で命令してフィードバックをプログラミング言語にすることで前者の曖昧さと後者の明確さを対比的に見せる事も可能である。これらの特徴によりユーザは普段自分が何気なく「実行」している行動が、細分化した Code によって捉え直せる事を直感的に体感できると考えた。本稿では提案システムの詳細と評価実験について報告する。

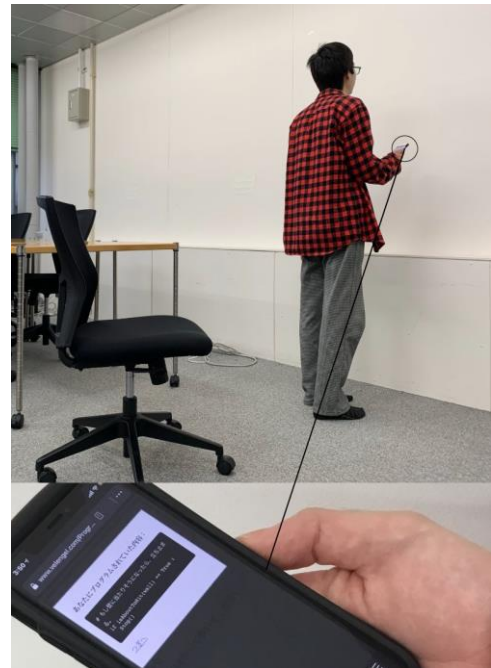


図1 システム体験時（壁に向かって歩く行動）の様子（上）
行動検出後にフィードバック画面が表示される様子（下）

2. 関連研究・事例

2.1 ソースコードを鑑賞や表現の対象とした研究・事例

ソースコードやプログラム自体が鑑賞や表現の対象となりうる事を示した先駆的な研究・事例として以下のものが挙げられる。日比野ら[2]は、JavaScript をベースにした独自言語を用いて様々な事象の表現を行い、その作品へのフィードバックを通してソースコードの持つ表現の可能性を示している。また、50 . Shades of Grey[3]は50 諧調の黒から白のグラデーションでできたまったく同じ画像を制作する6種類の言語によるプログラムの作品である。これらの研究・事例は作者が作成したソースコードそのものを鑑賞してもらうことに重点を置いているのに対し、本研究は日常の行動がプログラムの要素・処理に分解できることを示すことに重点を置き、動作に対して動的にソースコードを表示している点が異なる。

2.2 プログラミング的思考を身近にする目的の事例

プログラミング的思考を身近にする目的の事例として、テキシコー[4]がある。これは、コンピュータを使わずにプログラミング的思考を育むことを目的とした映像番組である。また、ピタゴラススイッチ[5]には、ミニコーナーであるアルゴリズムたいそう・アルゴリズムこうしんがある。これは、1人では意味のない身体動作を2人で組み合わせることによって意味を持つ、アルゴリズム（物を解くための手順）をテーマにした体操（行進）である。これらの先行事例がプログラミング的思考を映像で伝えるのに対し、本研究はユーザの身体動作に対してインタラクティブにフィードバックを返すアプリケーションという形をとる点で異なる。

2.3 携帯端末の内蔵センサを動作検出に使用する研究

本研究では身体動作を検出する身近なセンサとしてスマートフォンの内蔵センサを使用するが、同様の研究は数多くある。大内[6]のシステムでは、スマートフォンの加速度センサとマイクを用いて、家の中の行動と外の行動の連続的なモニタリングができる。また、西村ら[7]は、加速度センサとマイクを用いてユーザの周辺の混雑情報を推定する手法を提案している。これらの研究はユーザの生活の利便性の向上を目的とし、携帯している端末から一方的に値を取得しているのみであるのに対し、本研究ではプログラミング的思考が日常の身体動作に適用可能であると示すことを目的にしている点や、端末を手を持った状態のセンサの値を取得した上でユーザに即座にフィードバックしている点が異なる。

3. 提案：行動 to Code の設計

提案システム“行動 to Code”は、身体で完結する簡易な行動を自然言語でユーザに指示し、ユーザがその行動をする事を「ジッコウ」として捉え、それがより細かい身体動作（プログラムの要素）や処理の組み合わせによって行

われていることを、自動検出して即時かつインタラクティブにフィードバックする仕組みである。以下に(1)簡易な身体行動(2)センシングとフィードバックの装置(3)フィードバック表現の設計指針について詳しく説明する。

まず、身体で完結する簡易な行動として日常生活で何気なく行う「立ち上がる」「椅子に座る」などの9つの行動を取り上げる。これらの行動は表1に挙げる5つのプログラムの要素・処理を少なくとも1つ含むという観点から選択した。なお、本稿におけるプログラムの要素・処理とは、構造化プログラミング[8]における三要素である「順次（接続）」、「分岐（選択）」、「反復（繰り返し）」と、プログラミングにおける「変数（パラメータ）」、「関数（サブルーチン）」とする。

次に行動のセンシング及びフィードバック装置としては特殊なセンサ・デバイスではなく、ユーザが日常的に持ち歩いているものが好ましい。そこで、スマートフォン(iPhone)、とPCに着目して内蔵センサ・カメラ・ディスプレイを用いる。提案システムはこれらのブラウザからアクセスできるアプリケーションとして作成する。

フィードバック表現として、自然言語ではなくプログラミング言語（命令とコメント）を用いる。これは、プログラミング言語では、一連の動作は細かい処理・ステップに分解して一行ずつ記述するという特徴があるため、指示した行動に含まれていた詳細なプログラムの要素・処理を構造的・直感的に見せるのに適していると考えたためである。

表 1 各プログラムの要素・処理とその説明

プログラムの要素・処理	説明
順次	上から順番に処理を実行する
分岐	条件の真偽に応じて、次の処理を選択する
反復	条件が真である間、一連の処理を繰り返す
変数	あるデータを保持、読み書きする領域
関数	一連の処理のまとめ

4. 実装

4.1 システム構成

本システムのハードウェアとしてスマートフォンとPCを用いる。スマートフォンでは、内蔵の加速度センサ・ジャイロセンサを、PCではカメラを行動検出のセンサとして用いる。また、フィードバックの際のプログラミング言語はPythonの文法に則っている。提案システムでは、相互にゆるく関係する3つの行動をユーザに順々に指示してフィードバックを返す事を3回繰り返す。この一連の流れを手続き型人生体験ツアー（以下ツアー）と呼ぶ。ツアーは3種類ある。

各ツアーはHTML/CSS及びJavaScriptを用いてユーザの端末上のブラウザ(Chrome, Firefox)で動作するよう開発した。ツアー1,2はiPhoneで、ツアー3はPC(Windows, Mac)で動作する。ツアー1,2については、ユーザは図1(上)

のように iPhone を手に持ち、基本は画面を見ながら指示通りに行動をする。ツアー3については、ユーザはPCの内蔵カメラの前に座って、画面を見ながら指示通りに行動する。各ツアーの大まかな流れは以下の通りである。

1. ユーザに画面を操作してもらい、画面上で自然言語により指示(命令)されたとおりに行動をしてもらう。この命令は各画面につき1文だけ表示される(図3(左))。最初の画面(図2)のみ、センサ類許可処理を兼ねている
2. 行動を細分化した動作をスマートフォン内蔵センサもしくはPCのカメラで検知し、閾値を超えると動作をしたと判断する
3. 判断したら、「あなたにプログラムされていた内容」として、プログラムの要素・処理が図3(右)のようにユーザに即座かつインタラクティブに明示される
4. 表示されたウィンドウの「つぎへ」をタップすると、次の命令の画面が表示される
5. 3つの命令について1~4の処理・遷移が行われた後、3つの行動全体を表したCodeのフィードバック画面(図4)が表示される

以下の節では、各ツアーの具体的な命令とその動作に含まれるプログラムの要素について述べる。

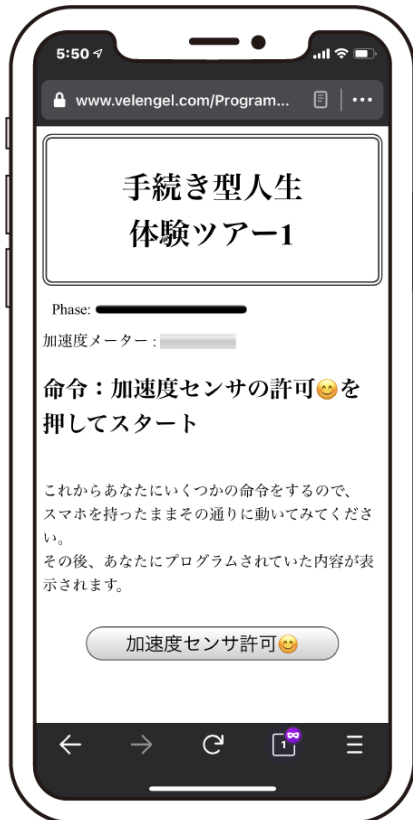


図2 ツアー1 最初の画面



図3 ツアー1 命令1(左)

命令1 プログラム的要素表示ウィンドウ画面(右)



図4 3つの行動全体を表したフィードバック画面

4.2 ツアー1 分岐と順次処理

4.2.1 命令とプログラムの要素・処理

ツアー1で自然言語により指示した行動は「立ち上がってください」「壁に向かって歩いてください」「椅子に座ってください」の3つである。これらの行動に含まれるプログラムの要素・処理は「分岐」と「順次」である。両者の対応関係は表2に示す。

また、ツアー1の最後に表示される3つの行動全体を示したプログラム(Code)のうち、命令2の部分を図5に示す。図5に示したように、「壁に向かって歩いてください」には、if文とコメントによって「もし壁に当たりそうになったら、立ち止まる」というプログラムの処理(分岐)が含まれていることがユーザに明示される。また、はじめに図1において示した行動はこの命令2である。

表2 ツアー1の各命令とそのプログラムの要素・処理

命令番号	文言	含まれるプログラムの要素・処理
1	立ち上がってください	分岐
2	壁に向かって歩いてください	分岐
3	椅子に座ってください	順次

```
# 命令2：壁に向かって歩いてください。
# もし壁に当たりそうになったら、立ち止まる。
if isAboutToHit(wall) == True :
    Stop()
```

図5 ツアー1の命令2のフィードバック画面

4.2.2 加速度の閾値の判定方法

命令1~3は、いずれも加速度センサの値を取得し、閾値に応じてユーザの行動判定をする。

iPhoneにおける加速度センサの軸の方向は左右がx、上下がy、表裏がzであり[9]、それぞれ左、下、裏がx,y,z軸正の向きである。

x,y,z方向の加速度をax,ay,azとし、これらの二乗和平方根をAとすると、Aは式(1)で示される。Aの値を50msごとに取得して配列に保存する。配列のn(n>0)番目に格納されているAをAnとする。さらに、AnとAn-1の差をBnとして式(2)より求める。

$$A = \sqrt{ax^2 + ay^2 + az^2} \dots\dots (1)$$

$$B_n = A_n - A_{n-1} (n > 0) \dots\dots (2)$$

Bn<-1.2となった時、命令1のポップアップウィンドウが表示される。同様に、Bn<-2、Bn>2.7となった時、それぞれ命令2,3のポップアップウィンドウが表示される。この閾値は、筆者自身が何度か試した上で一番最適な値に設定した。

4.3 ツアー2 関数と変数

4.3.1 命令とプログラムの要素・処理

ツアー2で自然言語により指示した行動は「後ろを向いてください」「10秒間なにもしないでください」「元の向きに戻ってください」の3つである。これらの行動に含まれるプログラムの要素・処理は「関数」「変数」「順次」である。具体的な対応関係は表3に示す。

ツアー2の最後に表示される3つの行動全体を示したプログラム(Code)のうち、命令1,3の部分を抜粋して図6に示す。図6に示したように、命令1「後ろを向いてくだ

さい」と命令3「元の向きに戻ってください」は、関数「TurnAround()」によって同じプログラムの要素・処理(関数)で表せることがユーザに明示される。

表3 ツアー2の各命令とそのプログラムの要素・処理

命令番号	文言	含まれるプログラムの要素・処理
1	後ろを向いてください	関数
2	10秒間なにもしないでください	変数、順次
3	元の向きに戻ってください	関数

```
# 後ろを向く関数
def TurnAround():
# スマホを見ながら180°回転する
    LookAt(SmartPhone)
    Turn(Math.PI)

def main():
# 命令1：後ろを向いてください。
    TurnAround()
```

```
# 命令3：元の向きに戻ってください。
# 命令1と同じ
    TurnAround()
```

図6 ツアー2の命令1と3のフィードバック画面

4.3.2 ジャイロ(角度)の閾値の判定方法

命令1,3は、どちらもiPhoneのジャイロセンサの値を取得し、閾値に応じてユーザの行動判定をする。

iPhoneの表裏の軸であるz軸まわりの回転角のみを考える。ユーザが最初にアプリケーションを起動したときの角度が真上(0°]=0[rad]になるため、3.14±0.3[rad]に入った時命令1のポップアップウィンドウが表示される。同様に、0±0.3[rad]に入った時に命令3のポップアップウィンドウが表示される。この閾値は、筆者自身が何度か試した上で一番最適な値に設定した。

4.4 ツアー3 反復と順次処理

4.4.1 命令とプログラムの要素・処理

ツアー3で自然言語により指示した行動は「まばたきをしてください」「画面をじっと見てください」「笑顔になってください」の3つである。これらの行動に含まれるプログラムの要素・処理は「反復」「順次」「変数」である。具体的な対応関係は表4に示す。

ツアー3の最後に表示される3つの行動全体を示したプログラム(Code)のうち、命令2の部分を抜粋して図7に示す。図7に示したように、命令2「画面をじっと見てください」にはウィンドウが出るまで「while True:」によって「反復」していたり、まばたきをする「Blink()」や、他のことを考える「Think(something)」という「順次処理」のようなプログラムの要素・処理が含まれていたことがユーザに明示される。

表 4 ツアー3の各命令とそのプログラムの要素・処理

命令番号	文言	含まれるプログラムの要素・処理
1	まばたきをしてください。	反復
2	画面をじっと見てください。	反復、順次
3	笑顔になってください。	反復、変数

```
# 命令2: 画面をじっと見てください。
while True:
    # 画面を見る
    LookAt(display)
    # まばたきする
    Blink()
    # 他のことを考える
    Think(something)
    # 周りを見る
    LookAround()
    # もしウィンドウが出たら、見るのをやめる
    if find(window) == True:
        break
```

図 7 ツアー3の命令2のフィードバック画面

4.4.2 表情認識の判定方法

表情認識のライブラリとして `clmtrackr[10]` を利用した。命令 1 のまばたきの判定に関しては、[11]の 2.瞬き検出を参考に実装した。また、命令 3 の笑顔判定に関しては、[12]の 3.表情認識を参考に、`emotion` 配列中の 5 番目 (0-indexed) に格納されている `happy` の値 > 0.9 ($\max 1$) を閾値とした。

4.5 動作確認

各ツアーについての動作確認を行った。指定した各画面において、正しく加速度・ジャイロセンサ、及びカメラが機能し、閾値に応じて画面が遷移することを確認した。

5. 評価実験

5.1 実験概要

提案システムを通して、日常生活の行動に含まれるプログラムの要素・処理に気づき、プログラミング的思考を身近に感じられるようになったかを明らかにするため評価実験を行った。実験参加者は 21~23 歳の男女 12 名であり、プログラミング歴は全員 3 年以上の熟達者であった。被験者をプログラミング熟達者に絞った理由は、プログラムの要素・処理を知識として知っているが、日常生活にまでは適用できていない人々を対象とするためである。手順としては、プログラミング的思考に関する実験であることは被験者に伏せた上で、3 つのツアーを順番に体験してもらい、終了後にアンケートを実施した。アンケートの評価項目は、「①アプリの体験を通して、自分の日常の行動の中に無意識にやっている複数の動作がある事に気づくことができた」、「②アプリの体験を通して、日常の行動の中に潜むプログラムの要素 (処理) に気づくことができた」、「③3 つのツアーを通して気づいたプログラムの要素 (処理) をできる限り書いてください (自由記述)」、「④本アプリのように、日常でプログラムの要素 (動作, 処理) をインタラクティブに体感させる仕組みは何に有用か (自由記述)」

「⑤一連の体験を通して、プログラミングの思考が身近に感じられるようになった」とした。このうち、①②⑤の項目に関しては「とてもそう思う」を 5, 「全くそう思わない」を 1 として 5 段階評価を行った。尚、各ツアーの順番はランダム化した。

5.2 結果

評価項目の内、①②⑤の結果を図 8 に示す。図 8 に示したとおり、評価項目①②⑤の平均はそれぞれ 3.75, 4.08, 4.25 であり、標準誤差はそれぞれ 0.28, 0.19, 0.22 であった。また、③についてはアプリケーション中で明示したプログラムの要素・処理全てを書き出せた人はいなかったが、1 つ以上は書いていた。他にも、「歩くときには、その場所が歩ける場所であることを確認しながら歩いていた」や、「ボタンを押す、リンクを入力する、文字を読む」、「動作と動作の間に顔を触る (自分の癖)」など、被験者ごとに様々なプログラムの要素 (処理・動作) に関する記述があった。また、④についても、「自身の癖を発見する」、「統計をとることによって特徴を抽出して、例えば嘘をついている人はこういう動作をしている、スポーツ選手はこういう動作を日々行っている、のように分類する」のように、様々な応用例に関する記述があった。

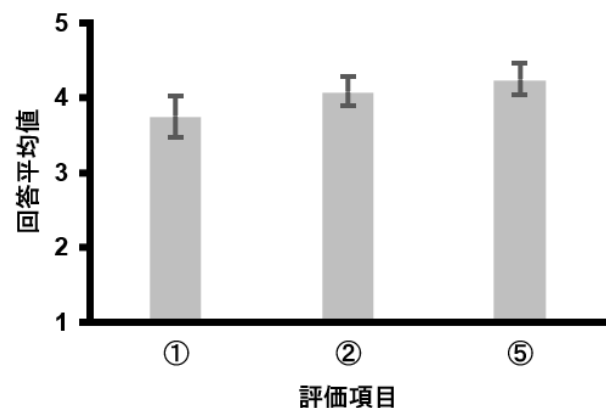


図 8 評価実験の結果 (項目①②⑤)

5.3 議論

評価項目①②⑤に関しては、5 段階評価で 3 以上の比較的高い平均値を得ることができた。しかし、図 4 に示すように最後のフィードバック画面で答えの一部である「無意識にやっている動作や、条件分岐」という言葉を出してしまっていたため、評価実験の結果としてはあまり有意なものではない。評価項目③に関しては、こちらがアプリ中に明示したものを全て書き出せた人はいなかったものの、それ以外の場面で、被験者がそれぞれプログラムの要素・処理に気づいていたことがわかる。特に、「アプリに正常に認識されているかをチェックする条件分岐」や、「無意識のうち何が無視するのか決める処理」、などは興味深かった。各々の自室の様子や、プログラミング経験によって、プロ

グラムの要素・処理に対する捉え方が異なるものの、普段一連の動作だと捉えているものを分解する考え方としてプログラミング的思考を利用できていた点で、本来の目的を達成できたと言える。評価項目④に関しては、想定していた自己の思考や行動の整理、見直し、の他にも多数の応用例について記述があった。特に、自身の癖を発見する、姿勢改善、職業による動作の特徴抽出などは非常に興味深かった。他にも、身近なプログラムの要素としてプログラミング初心者向けに例示する、などプログラミング教育への応用に関する記述もみられた。また、項目の最後に全体の感想（自由記述）を設けたところ、スマホの加速度センサの精度の悪さに関する記述がみられた。スマホを持って立ち上がる動作は個人差が大きいため、加速度のブレも大きかったことが想定される。

6. まとめと今後の展望

本研究では、プログラミング的思考が日常の行動の分析にも役立つ事やその面白さをより多くの人に知ってもらう事を目指し、簡易な身体行動を自然言語でユーザに指示し、そのプログラムの要素・処理による構成を、自動検出して即時にフィードバックすることで、日常の行動が Code によって捉え直せる事を直感的に体感させるシステムを実装した。加えて、システムを体験した後日常生活の行動に含まれるプログラムの要素・処理に気づき、プログラミング的思考が身近に感じられたかをユーザスタディを行い検証した。

今後の展望として、ユーザの行動に応じたプログラムの要素・処理の表示の分岐や、より細かい行動の分解、及びバグや例外処理、エラーなどの他のプログラムの要素と行動の対応付けなどが考えられる。具体的には、加速度センサ・ジャイロセンサの変化の大きさに応じて、フィードバックのプログラム表現が変わるようにし、閾値を大きく超えた行動に関してはエラーとして表示する、などが考えられる。また、提案手法で題材にしたプログラムの要素は基礎的なものに留まっていたが、より発展的な概念やデータ構造なども取り入れて、将来的には日常生活の様々な物事に応用可能であることを多くの人と共有するとともに、実際のプログラミング的思考の粒度や処理の適切さを考え直す一体験としての価値を高めていく。

参考文献

- [1] 小学校段階におけるプログラミング教育の在り方について(議論の取りまとめ)。
https://www.next.go.jp/b_menu/shingi/chukyo/chukyo3/053/siryo/icsFiles/afieldfile/2016/07/08/1373901_12.pdf
(参照 2020-12-13)。
- [2] 日比野 光紘, 山田 晃嗣. プログラム・ソースコードによる表現の可能性. 情報処理学会 インタラクション 2019 1A-08.
- [3] 50 . Shades of Grey - 文化庁メディア芸術祭 歴代受賞作品
http://archive.j-mediaarts.jp/festival/2015/art/works/19a_50_shades_of_grey/ (参照 2020-12-09)。
- [4] テキシコー | NHK for School.

- <https://www.nhk.or.jp/school/sougou/texico/> (参照 2020-12-09)。
- [5] ピタゴラスイッチ - NHK.
<https://www.nhk.jp/p/pitagora/ts/WLQ76PGNW2/> (参照 2020-12-09)。
- [6] 大内一成. 実世界に広がる装着型センサを用いた行動センシングとその応用: 4. スマートフォンを用いた生活行動認識 - 家の中も外もスマホで行動認識- 情報処理, Vol154, No.6, pp.578-581 (2013)。
- [7] 西村 友洋, 樋口 雄大, 山口 弘純, 東野 輝夫. スマートフォンを活用した屋内環境における混雑センシング. 情報処理学会論文誌, Vol.55, No.12, pp.2511-2523 (2014)。
- [8] Linger, Richard C.; Mills, Harlan D.; and Witt, Bernard I., "Structured Programming: Theory and Practice" (1979). The Harlan D. Mills Collection.
- [9] スマホの加速度センサを使う - こくぶん研究室.
<https://kkblab.com/make/javascript/acc.html> (参照 2020-12-13)。
- [10] GitHub - auduno/clmtrackr: Javascript library for precise tracking of facial features via Constrained Local Models.
<https://github.com/auduno/clmtrackr> (参照 2020-12-13)。
- [11] 兜森仁志, 安彦智史, 長谷川大, 佐久田博司. 2015. web カメラを用いた瞬き検出による集中度評価. 第 77 回全国大会講演論文集, pp.931-932. (2015)。
- [12] 表情認識で SNOW 的なアプリ - こくぶん研究室.
<https://kkblab.com/make/javascript/face.html> (参照 2020-12-13)。