

Tangible Code：物理センサーでコードの因果関係を体感するインタフェース

小芝 美由紀^{1,a)}

概要：プログラムにおける入力と出力の因果関係を直感的に理解することは容易ではない。本研究では、物理センサーを用いてコードのパラメータを操作し、リアルタイムに視覚的フィードバックを返すシステム Tangible Code を提案する。コードを印刷したパネルにセンサーを埋め込むことで、センサー（入力）→コード（処理）→ビジュアル（出力）という因果関係を身体的に体感できる。本稿では、この体験を設計するための3つの指針（コードとセンサーの物理的統合、コードの簡潔性と変化の明瞭性、常時動作状態の維持）を示す。展示での観察を通じて、物理センサーが因果関係への関心を引き出すこと、幅広い年齢層が直感的に体験できることを確認した。

1. はじめに

プログラムは入力を受け取り、処理を行い、出力を返す。この因果関係を理解することは、プログラミングの基礎である。しかし、コードは抽象的であり、入力と出力の対応を直感的に把握することは容易ではない。

既存のアプローチとして、ビジュアルプログラミングやタンジブルプログラミングがある。これらはエラーを排除し、直感的な操作でプログラムを構築できる。しかし、これらの目的はプログラムの構築であり、既存のコードにおける入力と出力の因果関係を体感するという用途には対応していない。

本研究では、物理センサーを用いてコードのパラメータを操作し、リアルタイムに視覚的フィードバックを返すシステム Tangible Code を提案する。ユーザーは物理センサーを操作することで、入力（センサー値）が処理（コード）を経て出力（ビジュアル）に変換される過程を身体的に体感できる。本研究の貢献は、この体験を設計するための3つの指針と、プロトタイプの評価を通じて得られた知見である。

2. 関連研究

2.1 ビジュアルプログラミング

Scratch に代表されるビジュアルプログラミングは、ブロックの組み合わせでプログラムを構築する [1]。構文エラーが発生せず、即時に実行結果を確認できる。本研究と

共通して、エラー排除と即時フィードバックを重視している。しかし、ビジュアルプログラミングの目的はプログラムの構築であり、既存コードの因果関係の体感ではない。

2.2 タンジブルプログラミング

物理オブジェクトを通じてデジタル情報を操作するタンジブルインターフェースの概念は Ishii & Ullmer に遡る [2]。この概念をプログラミング教育に応用したものがタンジブルプログラミングである。Horn & Jacob は木製ブロックを連結してロボットを制御するシステムを提案した [3]。藤田らはテーブル上で物理的なロボットを直接操作してイベント駆動プログラミングを行う RoboTable を開発した [4]。これらは物理操作による直感的なプログラミングを実現しているが、目的はプログラムの構築である。本研究は物理操作を用いる点で共通するが、既存コードの因果関係を体感することを目的とする点が異なる。

2.3 プログラム可視化

Python Tutor はコードの実行過程をステップごとに可視化するツールである [5]。変数の状態やスタックフレームを視覚的に表示し、実行フローの理解を支援する。本研究と共通して既存コードを対象としている。しかし、Python Tutor は実行フローの追跡に焦点を当てており、パラメータを変えて結果の変化を体感するという体験は提供していない。

2.4 本研究の位置づけ

以上の関連研究を踏まえ、本研究の位置づけを整理す

¹ 無所属

^{a)} ksbmyk@gmail.com

る。ビジュアルプログラミングとタンジブルプログラミングは、プログラムの構築を目的としている。一方、プログラム可視化は既存コードを対象とするが、実行フローの追跡に焦点を当て、ユーザーは観察する立場にある。本研究は、既存コードを対象とする点でプログラム可視化と共通し、物理操作を用いる点でタンジブルプログラミングと共通する。しかし、本研究の目的は、プログラムの構築でも実行フローの追跡でもなく、パラメータ操作を通じて入力と出力の因果関係を体感することにある。物理センサーを用いることで、タンジブルインターフェースの身体性を取り入れ、この因果関係をより直接的に体感できることを目指す。

3. Tangible Code

3.1 コンセプト

Tangible Code は、物理センサーを用いてコードのパラメータを操作し、リアルタイムに視覚的フィードバックを返すシステムである。ユーザーはセンサーを操作することで、コード中の変数値を変更し、その結果としてビジュアルが変化する様子を観察できる。センサー（入力）→コード（処理）→ビジュアル（出力）という因果関係を、環境構築やエラーなしに体感することを目指す。

3.2 コードとセンサーの物理的統合

コードを印刷したパネルにセンサーを埋め込み、変数とセンサーを物理的に一体化させる。パネルに印刷されたコードは、センサー値を定数に置き換えれば実際に動作するコードであり、描画ロジックそのものである。コードの表示にはエディタのシンタックスハイライトと似た配色を用い、コードを編集しているという認識を促す。これにより、変数を直接操作している感覚を生み出す。物理センサーの手応え、重さ、カチッという感触が、直感的な操作を可能にする。

3.3 コードの簡潔性と変化の明瞭性

コードの分量は、ユーザーが内容を把握しようとする意欲を維持できる範囲に抑える。過度に複雑なコードは、操作前の段階で関心を失わせる要因となる。一方、パラメータ変更に対する視覚的变化は明瞭にし、操作と結果の対応関係を知覚しやすくする。色、大きさ、個数など複数のパラメータを用意することで、組み合わせによる多様な表現を可能にし、継続的な探索行動を促す。

3.4 常時動作状態の維持

展示環境において、来場者が体験を開始するまでの心理的障壁は大きい。本システムでは待機中もビジュアルが継続的に動作する設計とした。動きによって注意を引きつけ、操作開始時の急激な状態変化を避けることで、来場者

がインタラクションを開始しやすい状態を維持する。

4. 実装

4.1 作品の構成

本システムでは、2種類の円が中央を中心に回転するビジュアルを題材とした。ユーザーは物理センサーを操作することで、色、大きさ、回転速度、中心からの距離、円の個数といった描画パラメータを変更できる。これらのパラメータは、コード中の変数として定義されており、センサー入力が即時に描画結果へ反映される。連続的な値を扱う可変抵抗器と、状態の切り替えを行うトグルスイッチを組み合わせることで、異なる種類の入力が描画に与える影響を観察できる構成とした。

4.2 センサーの選定

センサーは、操作対象となる変数の性質に応じて選定した。色相、透明度、回転速度、円の個数、中心からの距離、円の大きさといった連続的に変化する数値には可変抵抗器を用いた。可変抵抗器は、つまみを回すことで値を連続的に増減でき、数値の変化を直感的に操作できる。一方、背景色と描画モードの切り替えには、トグルスイッチを用いた。トグルスイッチは ON/OFF の 2 状態を切り替えるデバイスであり、コード中の Boolean 値に対応する。

4.3 パネルのデザイン

パネルには、実際に動作するコードをシンタックスハイライト風にカラー印刷した。キーワード、変数名、数値などを色分けすることで、コードエディタで見慣れた表示に近づけている。センサーは、コード中の変数に値を代入する箇所に直接配置した。たとえば `hueValue =` の右側にセンサーを埋め込むことで、センサーを操作すると変数の値が変わるという対応関係を視覚的に示している。

4.4 システム構成

システムは Arduino と Processing から構成される（図 1）。Arduino は物理センサーから入力値を取得し、シリアル通信を介して PC に送信する。Processing 側では、受信した値をコード中のパラメータに代入し、描画処理に反映する。この構成により、物理センサーによる入力が、コードを介して視覚的出力へ変換される一連の処理過程をリアルタイムに観察できる。



図 1 システム構成の概要

5. 観察と設計上の示唆

5.1 開発プロセス

本システムは、プロトタイプ制作と展示を通じた観察を反復しながら設計を進めた。本節では、各段階で得られた観察結果と、そこから得られた設計上の示唆を整理する。

5.2 Web 版プロトタイプでの課題

最初に、ブラウザ上でスライダーやキーボードを用いてパラメータを操作する Web 版プロトタイプを作成した。コードを表示し、パラメータを変更するとビジュアルがリアルタイムに変化する構成であったが、「図形を作るアプリに見える」という指摘が得られた。このことから、画面上の汎用的な入力操作では、コードとの対応関係が十分に意識されにくい可能性が示唆された。この観察を踏まえ、物理センサーを用いた展示版の制作に進んだ。

5.3 展示を通じた観察

物理センサーを用いた展示版を作成し、2025 年 3 月の武蔵野美術大学卒業制作展において予備的な観察を行った(図 2)。展示期間 4 日間のうち 3 日間、会場において来場者の行動を観察した。来場者は子どもから大人まで幅広い年齢層であった。

3.2 節で述べたコードとセンサーの物理的統合に関しては、センサーの手応えに対する反応が多く観察された。来場者の関心は、操作とビジュアルの対応関係に向けられる傾向が見られ、入力と出力の因果関係に注意を向ける体験が生じていることが示唆された。一方で、手元のパネルと前方の画面に視線が分散する様子も観察された。

3.3 節で述べたコードの簡潔性と変化の明瞭性に関しては、複数のパラメータを組み合わせる探索的な行動が見られた。自分好みの状態を見つけて止める来場者や、一度体験した後に再び戻ってくる来場者も観察された。

3.4 節で述べた常時動作状態の維持に関しては、待機状態から自然に操作へ移行する様子が多く観察され、体験開始のハードルを下げる設計として機能している可能性が示唆された。

5.4 改良版における観察

展示版で観察された視線の分散という課題を踏まえ、画面上にコードとセンサー値をリアルタイムに表示する改良を行った(図 3)。RubyWorld Conference 2025 における発表では、センサー操作に応じて変数値が変化するように注目する反応が見られ、操作とコードの対応関係を把握しやすくする表現として有効である可能性が示唆された。

5.5 考察

Web 版プロトタイプでは「図形を作るアプリに見える」



図 2 展示版デモの設置例

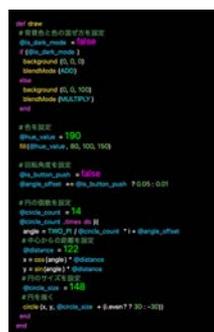


図 3 改良版デモにおける画面表示

という課題があったが、物理センサー版ではこの指摘は見られなかった。この違いは、操作対象の認識に起因すると考えられる。画面上のスライダーやキーボードは汎用的な入力デバイスであり、何を操作しているかが曖昧になりやすい。一方、物理センサーをコードが印刷されたパネルに埋め込むことで、入力デバイスとコードの関連が明示される。展示での観察では、来場者の関心は「センサーとビジュアルがどう繋がっているか」という仕組みに向いていた。これは、本システムが入力と出力の因果関係を体感させるという目的を達成していることを示唆する。幅広い年齢層がこの体験に興味を持ったことは、プログラミングの前提知識がなくても因果関係を直感的に把握できることを示している。3.4 節で述べた常時動作状態の維持は、体験開始のハードルを下げる効果があった。展示では、動いているビジュアルに注意を引かれて立ち止まり、そのまま操作を始める来場者が多く観察された。常に動いているため、前の来場者が操作した状態がそのまま残っており、リセットや初期化といった準備が不要である。また、触ったときの変化もわかりやすい。開始・終了の操作がないこと

でコードもシンプルになり、読む気が起きる量に抑えることに貢献している。

5.6 デモ展示

本デモでは、展示版のパネル方式（コードを印刷したパネルにセンサーを埋め込む構成）を基本としつつ、改良版で得られた知見を反映する。具体的には、画面上にもコードとセンサー値をリアルタイムに表示し、センサーと変数の対応関係を色で明示する。センサーには可変抵抗器とトグルスイッチを用いる。この構成により、パネル操作の身体性と、画面での視覚的な対応関係の確認を両立させることを意図している。本デモでは、来場者に実際にシステムを操作してもらい、物理センサーを用いてコードの因果関係を体感するというアプローチについて意見を伺いたい。既存のタンジブルプログラミングがプログラムの構築を目的とするのに対し、本システムは既存コードの因果関係を体感することを目的としている。このような試みが、コードに興味を持つきっかけとなりうるか、また他にどのような応用や展開が考えられるかについて、議論の場としたい。

6. おわりに

本稿では、物理センサーを用いてコードのパラメータを操作し、その結果をリアルタイムに視覚的フィードバックとして返すインタフェース Tangible Code を提案した。本システムは、センサー（入力）→コード（処理）→ビジュアル（出力）という因果関係を、身体的な操作を通じて体感することを目的としている。本稿では、この体験を成立させるための3つの設計指針を整理した。展示での観察では、来場者が操作とビジュアルの対応関係に関心に向け、探索的に試行する様子が確認された。本デモでは、これらの設計が来場者にどのように受け取られるかを共有し、本アプローチの可能性と展開を検討する。

参考文献

- [1] Resnick, M., et al.: Scratch: Programming for All, Communications of the ACM, Vol.52, No.11, pp.60-67 (2009).
- [2] Ishii, H. and Ullmer, B.: Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms, Proc. CHI '97, pp.234-241 (1997).
- [3] Horn, M.S. and Jacob, R.J.K.: Designing Tangible Programming Languages for Classroom Use, Proc. TEI '07, pp.159-162 (2007).
- [4] 藤田真浩, 高橋 治久, 酒造正樹: タンジブルプログラミングツール RoboTable の開発, インタラクシオン 2011 論文集, pp.107-114 (2011).
- [5] Guo, P.J.: Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education, Proc. SIGCSE '13, pp.579-584 (2013).