日々と表現を束ねて流通する「短いコード」

――デイリーコーディングからライフコーディングの実践へ向けて――

高 尾 俊 介

The“Short Code”that Binds Together and Circulates Daily and Expressions: From Dailycoding to Lifecoding

TAKAO Shunsuke

Abstract: In general, in the software development process, including the front end, most of the products are produced and delivered through planned processes such as requirement definition, design, development and testing. In addition to the waterfall development methods, such as agile and XP, which set multiple development periods in a short period of time, many of the considerations are based on the division of labor and collaboration, such as the elimination of dependency, as a useful method in the complex and large-scale software development of today. It is the establishment and practice of uniform and efficient development methodologies for enabling product development. In this paper, we focus on creative coding, which is a current trend in programming that is mainly focused on expression, not on functionality and purposefulness, but on the value of negative aspects in software engineering, such as subjective and improvisational judgments, code writing style, shortness, and illegibility. The possibilities are discussed through the trend of the domestic creative coding community and the author's practice of“daily coding”.

Key Words: creative coding, poetic computing, new media art

要旨：一般に，フロントエンドをはじめとしたソフトウェア開発では，多くの製作物は要件定義や設計，開発やテストといった計画的な工程を経て製作・納品される。このようなウォーターフォール型の開発手法以外にも，開発期間の単位を短く複数回設けるアジャイルや XP のような開発手法にせよ，考慮されることの多くは，複雑で大規模化する現代のソフトウェア開発の現場において有益な手法としての，属人性の排除を代表とした，分業・協働を可能にするプロダクト開発のための画一的，効率的開発手法の確立と実践である。本稿では表現を主軸とするプログラミングの潮流であるクリエイティブコーディングにおいて，機能性や合目的性を主眼とせず，代わりに個人の主観的，即興的判断やコードの書きぶり，短さ，難読性といった，ソフトウェア工学的価値観においてネガティブな側面を逆に価値とする表現の可能性について，国内のクリエイティブコーディングコミュニティの動向と筆者の実践「デイリーコーディング」を通じて論じる。

キーワード：creative coding, poetic computing, new media art

1. はじめに

一般に，フロントエンドをはじめとしたソフトウェア開発では，多くの製作物は要件定義や設計，開発やテストといった計画的な工程を経て製作・納品される。このようなウォーターフォール型の開発手法以外にも，開発

期間の単位を短く複数回設けるアジャイルや XP のような開発手法がある。いずれにせよ，複雑で大規模化するソフトウェア開発の現場では，コード内に含まれる属人性の排除を代表とした，分業，協働を優先する画一的で効率的な開発手法の確立と実践が求められている。本稿では表現を主軸とするプログラミングの潮流であるクリエイティブコーディングにおいて，機能性や合目的性を主眼とせず，その代わりに個人の主観的，即興的判断やコードの書きぶり，短さや難読性といった，ソフトウェア工学的価値観におけるネガティブな側面を逆に価値とする表現の可能性について，国内のクリエイティブコーディングコミュニティの動向と筆者の実践「デイリーコーディング」を通じて論じる。

なお，本稿で扱うプログラミング関連の用語を予め定義する。「コード」はコンピュータによって実行されることを前提とした記述されたプログラムそのものを指す。また「フロントエンド開発」は，組込みシステムやミドルウェア開発のようなソフトウェア開発の一つの領域であり，ブラウザで実行される Web アプリケーションの開発を指す。

2.「短いコード」を擁護する

2018 年に研究者の久保田晃弘は『短いコードを擁護する　In Defence of the Short Code』というテキストを書いている。これは，2009 年にアーティストで理論家のヒト・シュタイエルによって書かれたテキスト『貧しい画像を擁護する』を援用したものである。このテキストで久保田は BASIC で書かれた 1 行コードのプログラム「10 PRINT CHR＄（205.5＋RND（1））：：GOTO 10」や「超絶技巧プログラミング」とともに，コードがコード自身を実行結果として表示するクワインのような創作例を挙げながら，以下のように述べている。

> *しかし，こうした単機能の「短いコード」には，ブラックボックス化した API（アプリケーション・プログラム・インターフェイス）を組み合わせた，「詳しくはわからないけれど，とにかく動いている」今日的な実用プログラミングにはない，（不）完全さ，（未）完成さ，そしてそこから生まれる単純性と多様性，そして何より詩的跳躍がある。*
>
> *久保田晃弘・畠中実（2018）．メディア・アート原論　あなたは，いったい何を探し求めているのか？，フィルムアート社．（Kindle の位置 No.2122-2125）．Kindle 版.*

また，以下のようにも論じている。

> *「短いコード」による実験コーディングは，（中略）今の時代の主流の価値観では顕在することのない，多様な表現の可能性を孕んでいる。それらはいずれも，プログラムの長さや実行速度，効率や結果の正しさや実用性だけを評価する，近代の実利主義のコーディングとはまったく違った意味や思想に根ざしている。*
>
> *久保田晃弘・畠中実（2018）．メディア・アート原論　あなたは，いったい何を探し求めているのか？，フィルムアート社．（（Kindle の位置 No.2150-2154）．Kindle 版.*

ここで久保田は，実利主義的観点から開発された巨大なコードと対置するものとして，「短いコード」を論じている。そして短いゆえに，逆説的にアルゴリズムの単純性や多様性が再発見されるとともに，制約の中で書き手の意識下において発現する創造性について言及している。またそういった「短いコード」から生まれる文化的な意味を捉え直す必要性を説いている。

議論の前提となるシュタイエルによるテキストでは，「短いコード」の参照元である「貧しい画像」が議論の中心に置かれている。この「貧しい画像」とは画質が粗く低解像度のイメージのことであり，それは高解像度でクリアなイメージに明瞭さや資料的価値においては劣っていると一般的には考えられてきた，とシュタイエルは論じている。その一方で，圧縮され軽量である貧しい画像は，インターネット上では軽量であるがゆえにどこまで

も流通していく。そしてシュタイエルはインターネットでは潜在的に貧しい画像が求められていることとともに，画質と容量に起因するイメージがもつ価値の不等号は，サイバースペースにおいてはその流通性によって逆転することを主張している。

本稿では，シュタイエルの主張した流通性を，「短いコード」に適用して思考することを試みる。そしてその現在の日本国内におけるクリエイティブコーディングコミュニティの情勢を論じていく。

3. つぶやき Processing, dwitter,（centiscript）

久保田とシュタイエルのテキストから出発して，インターネット上での流通性と「短いコード」の関係について考えていく。流通する「短いコード」の代表例としては勝倉一博が Twitter 上で開始した「つぶやき Processing」がある。つぶやき Processing は，1 ツイートの 280 文字に収まる範囲で Processing 関連のコードを投稿する活動である。2019 年 8 月から開始し 2020 年 8 月末迄の 1 年間で，170 名の参加，約 2,700 作品投稿があった。その他，ブラジルのクリエイティブコミュニティが主催するイベント「Noite de Processing」でつぶやき Processing をテーマとするセッションが開催されるなど，国境を超えて，コードを通じた表現手段として伝播しつつある。

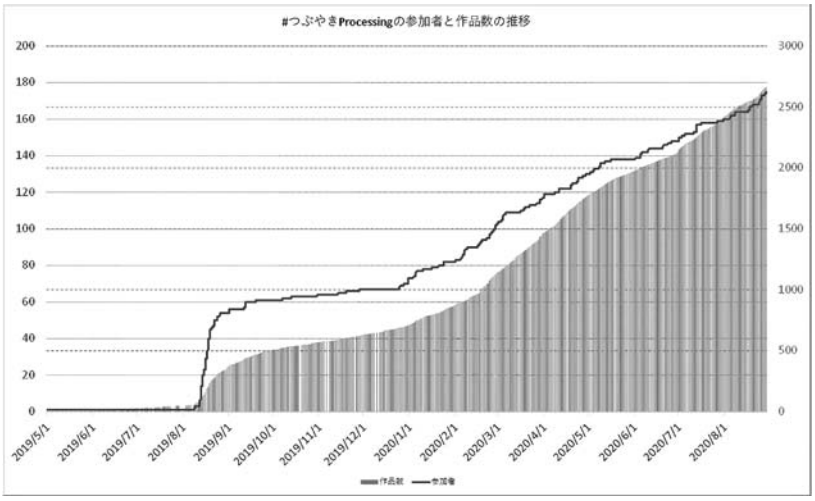

図 1　つぶやき Processing の参加者と作品数の推移

図 2　著者による#つぶやき Processing の投稿

つぶやき Processing と類似した，SNS 上での「短いコード」が展開していく例としては「dwitter」や「(centiscript)」が挙げられる。Andreas Selvik が開始した dwitter は，140 文字以内でグラフィックを描くチャレンジである。2016 年に始まった dwitter はつぶやき Processing と同様 Twitter の文字数制限をルールとして内在している一方で，ウェブサイト（dwitter.net）ではコードの技術要素についてディスカッションがなされるなどプラットフォーム上でクリエイティブコーダー同士の交流が生まれている。2014 年に堀井哲史によって開発された（centiscript）は，「ジェネラティブなグラフィックを，思考するよりも早く実践することを目的に作成した言語」であり，独自関数に省略記法が組み込まれるなど，分かりやすさ以上に演奏するようにコーディングするライブコーディングのような即興性を意識した実装的な特徴が見られる。

dwitter と（centiscript），両者はともに Javascript と W3C（World Wide Web Consortium）が策定した Canvas API をベースとしている。このブラウザ上で編集・実行が可能である点は Processing から派生した p 5.js も同様であり，「短いコード」を端としたコードの流通性の高まりとブラウザベースでのクリエイティブコーディング表現の隆盛には，河川と水源に類似した関係性を見て取ることが出来る。

このような「短いコード」に関連した研究としては，2014 年に書かれた宮代と宮下による「140 文字 Processing プログラミング」がある。ここでは Twitter 上の文字数制限内で行われたコードゴルフやショートコーディングのような競技的活動に触れている。2017 年に Twitter の文字数制限が 140 文字から 280 文字に変更されたことは，後発のつぶやき Processing や dwitter のような「短いコード」が流通した要因として非常に大きいと考えられ

る。Processing のような初期化（setup）とメインループ（draw）の 2 つの関数を起点とする多くのクリエイティブコーディング環境では，書き始めに最低限必要とされるオーバーヘッドが占める割合が高く，当初の 140 文字の制限下においては，自由に記述可能な文字数が予め大きく制限されていたためである。

4.「短いコード」とともに在る見えない長いコード

アーティストの稗田直人はウェブ上で公開した「ポスト・コーディング」という題のテキストで「短いコード」のバックグラウンドで動作する「長いコード」について以下のように言及している。

> *つぶやき Processing という 1 ツイート（280 文字）にコードを詰め込むフォーマットがあり，私もたまに投稿している。しかしその度に，実際には 93039 行ほどのコードが裏では動いていること，そしてその制約の中で「短いコード」を書いていることを思い出す（p 5.js バージョン 1.1.9 の場合。そしてもちろん JavaScript のインタプリタやブラウザ，OS に支えられている）*
>
> *ポスト・コーディング | naotohieda.com（最終閲覧日：2020 年 10 月 28 日）*
> *http://naotohieda.com/blog/post-coding-ja/*

p 5.js は Processing の系譜上にあるクリエイティブコーディング環境である。Processing のシンタックスを継承しながら，Canvas API を簡便な記法として移植・翻訳するような形で実装されている。ソフトウェア工学的見地に立ったとき，描画速度や計算コストを考えると直接 API にアクセスするメリットもあるが，一方で簡便で短い記法に包まれた API と書き手の瞬発的な思考の結びつきからときにジェネラティブなアウトプットが導き出される。

一般に行われるソフトウェア開発のように，安定したプロダクトを開発することを優先してコードを書く場合，一連の設計にまつわるプロセスを手順として踏むことが安全である。一方で堅牢性や安定性を重視する以上に目の前のコードを読み書きし，コンピュータと書き手の応答の中で実行されることだけを目指すコードにも，プロダクトにはない別種の創造的な価値を見出すことは出来る。コードをスケッチと呼ぶ，素描のようにコードを書いて実行するクリエイティブコーディングのような活動の多くは，在る種の適当さや場当たり的な対処という意味での tinkering（修理・改良のために小さな変更を加えること，いじくること）のような発想と試行錯誤の連続によって生み出されている。合目的性を一時的に手放すことで，書き手とコードの間に主観的で内省的な関係性を生じさせることができる。

クリエイティブコーディングやジェネラティブアートの特徴としてある，柔軟な発想と実装は，低レベルな API や冗長性を持った複雑なコードからはなかなかに生まれづらい。シンタックスが許容することで暗黙的に導かれる論理的思考は，プログラミング言語の開発者が持つ思想的な背景と見えづらい形ではあるが，強固に結びついている。「短いコード」は背後で動作する長いコードと接続する形で予め選択肢を示されている。同様に，コードにおける開発や実装と言った機能主義的な視点は不可分であり，それらを完全に排除することはできない。

5. 国内クリエイティブコーディングコミュニティの醸成

これまで述べてきた「短いコード」は SNS での使用可能文字数が増加したことのみが要因で流通が急速に広まったわけではなく，その下地には国内クリエイティブコーディングコミュニティの醸成があった。ここでは Processing に関連したコミュニティの活動を紹介しながら「短いコード」と関連する，日常におけるコーディング表現の広がりについて論じる。

クリエイティブコーディング環境の Processing は 2001 年に誕生して以来，非専門家のためのコーディング環境として広く知られており，安定的に更新されている一方で，登場してから 20 年近くが経過し，後発のクリエイティブコーディング環境として様々なものが登場してきた。そのような中，国内コミュニティは 2019 年から活動を

開始し，過去 2 回「Processing Community Day Tokyo（PCD Tokyo）」を開催している。PCD Tokyo では，日本における Processing を始めとしたクリエイティブコーディングコミュニティの活動が紹介され，またメンバー同士の交流の場が設けられた。

他の国内クリエイティブコーディングコミュニティの活動例として TDSW（Tokyo Developer Study Weekend）の活動がある。TDSW は TouchDesigner を主とした，広義のクリエイティブコーディング表現のコミュニティであり，近年はオンラインイベントを中心としてアーティストやクリエイティブコーダーといった活躍する有識者によるトークやワークショップを展開している。TDSW で行われている技術情報の共有と交換が，デジタルアートを含めたインタラクティブなメディア表現分野の振興に広く貢献していることは，コミュニティの規模の成長からも伺える。また，TDSW は YouTube チャンネル上でビデオチュートリアルを多数公開している。それは特に時系列順で GUI の操作を行う必要があるノードベースのプログラミングにおいては，文書による指南書よりもわかりやすく，多数のチュートリアルのアーカイブがコミュニティの活発化を支える柱となっている。

6. 日常的にコードで表現する，実践としての「デイリーコーディング」

筆者は，日々の連続性の中で短いコードを書き続けていく活動を「デイリーコーディング」と提唱し，2018 年から現在まで活動を継続してきた。こういった活動を踏まえ，前述のコミュニティイベント「Processing Community Day Tokyo」ではトークセッション「日常的にコードで表現する」で活動の広がりや，その個人的・社会的意義について勝倉氏らの実践者とともにディスカッションを行った。

図 3　トークセッション「日常的にコードで表現する」会場風景，撮影：福島シオン

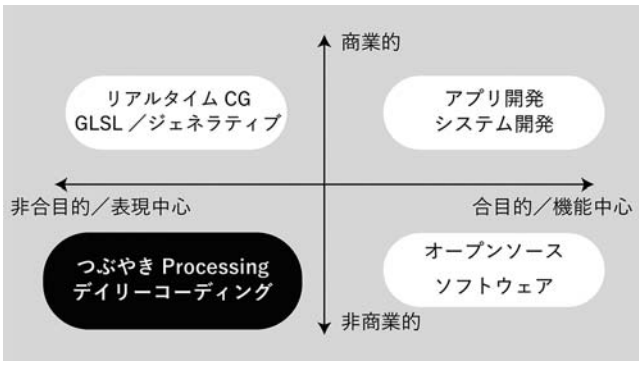

図 4　「短いコード」の位置

デイリーコーディングが，いわゆる課題解決や特定分野への指向性を持つプロジェクトベースやミッションオリエンテッドなコーディングと大きく異なっている点は，開発期間の中でコーディングを行うのではなく，年単位やそれ以上の中長期的で連続性をもった時間の中で実践される点である。時間的連続の中でコードで表現することそのものに思考を巡らすことは，自ずと日常生活の中にコードに付随する表現的側面を取り入れ，位置づけ，再解釈することへとつながっていく。そしてそれはコードの原理を日常に，生活の原理をコードに持ち込むことに他ならない。祝祭の装飾のような「ハレ」のためのコードではなく，連続する日常での変化を機微としてとらえるような「ケ」を含んだコードをどのように書くことができるか。コンピュータと人間，両者の結びつきをコードを通じて触れ，見つめ，聴き，嗅ぎ，味わうような，ディスプレイから届く視覚的な情報を超えて，様々な感情や経験といった私的記憶をどうコードで表現できるか，コンピュータと人間の関係性を人文学的な見地から考えていくことが日常的にコードで表現していくことの本質であり，コードの複雑性や技巧的な優位性は背景となり，コードの書きぶりや，即興性，主観的な連続性が前景化する。

7.「デイリーコーディング」から「ライフコーディング」へ

筆者が公開しているデイリーコーディングの多くは，100 行にも満たないごくごく短いコードである。それら一つ一つを平易な日本語でまとめたアルゴリズムとともにクリエイティブ・コモンズ・ライセンス表示－非営利－継承 4.0　国際（CC BY-NC-SA 4.0）で公開している。コードは形や配置，色，質感にまつわる表現技法と，その組み合わせによってできており，他者が読み書きして別のコードへと展開することも可能である。

「短いコード」の多くは，静止画を出力するようなグラフィックのための「動かないコード」である。動かないコードは，鑑賞者の振る舞いや操作に反応するインタラクティブな表現や，アニメーション，コンピュテーションを活用した生成的な表現が特徴のクリエイティブコーディングの文脈に置いて，「短いコード」と同様に貧しさを持ったコードである。しかし，コードを書くフェーズにおいては，動かないがゆえの利点もある。アニメーションするコードは，コーディング時に動きや全体を制御するためには実時間がかかる。1 分のアニメーションを確認・調整するには 1 分を端折ることなく確認する必要がある。他方で「動かないコード」は，結果が実行後即座に固定されるため，素早く実行結果の確認と書き換えを繰り返すことができる。このようにコーディング時の書き手の視点に立ったとき，「動かないコード」の単位時間あたりのコーディングで得られる発見と発想力の広がりには，他とは単純に比較できない価値がある。

短く，そして動かないコードは，生活の中で時に場当たり的に書き連ねられ，軽量で単純であるがゆえにビジュアルとともに流通していく。同時に，書き手の思考も，なぜコーディングするのか，コーディングと生きることの意味の接続，あるいは社会的なコーディングへと意識が巡っていく。

ステージ上である定められた時間の中で演奏するようにコードを公開しながら書くパフォーマンス，ライブコーディングのように即興性と場当たり的な処置をコードに組み込みながら，年月のような長く細い時間の中でコーディングとそれにまつわる思考を紐解き，束ねることは，その意味で「ライフコーディング」である。ライフコーディングは個人の生や日常生活とコードを結びつけた表現を目指す試みであり，非専門家を含む全ての人々の多様な目的へとクリエイティブコーディングを拓くものである。それは同時にコードをエディタ画面の外側，生活の内側へと持ち込むことであり，私的で詩的な表現の可能性につながっていく。ソフトウェア開発的な速度や効率，再利用性を重視する以上に，個人的の視点とコーディングやプログラミングに関する一般的規範や社会通念から，ときに自由であることを重視する「ライフスタイルとしてのコーディング」は，コードをリファクタリングするように，生活と生のあり方をも軽やかに書き換えていくものになるだろう。

参考文献

1　久保田晃弘・畠中実，メディア・アート原論あなたは，いったい何を探し求めているのか？，フィルムアート社，2018.
2　Hito Steyerl, “In Defense of the Poor Image”. e-flux Journal #10, 2009.11，閲覧日 2020-10-25, https://www.e-flux.com/journal/10/61362/in-defense-of-the-poor-image/
3　#つぶやき Processing まとめ－はぅ君プロジェクト，閲覧日 2020-10-25, https://haukun.projectroom.jp/archives/588
4　Noite de Processing - Garoa Hacker Clube，閲覧日 2020-10-25, https://garoa.net.br/wiki/Noite_de_Processing
5　About | Dwitter，閲覧日 2020-10-25, https://www.dwitter.net/about
6　（centiscript）| Working Log，閲覧日 2020-10-25, http://satcy.net/wlog/?p=922
7　宮代理弘，宮下芳明，140 文字 Processing プログラミング，エンタテインメントコンピューティングシンポジウム 2014 論文集，Vol.2014.
8　ポスト・コーディング| naotohieda.com，閲覧日 2020-10-25, http://naotohieda.com/blog/post-coding-ja/