

体験指向ドローソフトの提案とその開発環境の試作

秋山 博紀¹ 増井 俊之² 安村 通晃²

概要：ソフトウェアづくりの楽しさのひとつは、頒布して実際にユーザーに使ってもらうことであるが、完成させて頒布までたどり着くのは難しい。そこで本研究では、ソフトウェアづくりの楽しさを体験することの敷居を下げることを目的に、特定のジャンルに特化することで気軽にソフトウェアづくりの楽しさを体験してもらえる開発環境 Expresso を試作した。Expresso は、筆者が新たに定義した「体験指向ドローソフト」を開発の対象とする。体験指向ドローソフトは、絵を描く行為そのものを目的としており、成果物の完成度を高めることは目的としない。Expresso の利用者は、ペン・ボード・オーラという3つの要素を設定して世界観を構築することで、独自性のある体験指向ドローソフトをつくることができる。Expresso が基本的な構造を提供するため、ユーザーは独自性に直接関わる部分の設定に集中でき、ソフトウェアづくりにおける敷居を下げていく。

1. はじめに

日曜プログラマーやホリデープログラマーといった言葉が存在するように、ソフトウェアをつくるプログラミングは趣味のひとつとして扱われている。既に大量のソフトウェアがアマチュアプログラマーによって製作されているが、ソフトウェアづくりという行為そのものは学習して身につけることが必要で、万人が気軽に楽しめるというものではない。

同じ創造行為を考えると、例えば写真撮影は使い切りカメラやプリントシール機、スマートフォンのカメラアプリケーションのように、様々なツールが存在し、それによって気軽に誰でも写真を撮れる。しかし、コーディングを主体としたプログラミングは知識と経験がなければ扱いづらく、写真撮影のように誰でも気軽に試して、成果物を公開できるものではない。

安村は現在のプログラミングの分かりづらさの原因のひとつとして、記憶再生型の構文指向に問題があるとし、ユーザーにとって扱いやすい意味指向のプログラミングとして Programming2.0 を提唱している [1]。

本研究では、Programming 2.0 を踏まえ、ソフトウェアづくりを気軽に体験できる環境を提案し、試作システムの実装について報告する。

2. 既存のソフトウェア製作の課題

ソフトウェアづくりの手段としてのプログラミングは、完成までに障壁があり、ユーザーにとって敷居が高い。敷居が高い原因として、3つの問題が考えられる。

2.1 ソフトウェアづくりの障壁

(1) 作り始めるまでの障壁

増井は、プログラミングの敷居の高さの原因として、市販の開発環境は値段が高い一方、フリーの開発環境はインストールに大きな手間がかかる点を指摘している [2]。ソフトウェアをつくらうと思っても、開発環境を用意してプログラミングを始めるまでの敷居が高さがある。

また、そもそもプログラミングによってどのようなソフトウェアが生成できるのかが分かりづらい。例えばスマートフォンアプリケーションであれば、どのアプリを使って写真撮影をしたのかが事前に分かることがあり、それを手がかりにどの撮影環境でどのような写真が撮影できるのかわかる。一方、どのようなアプリケーションでも作れる開発環境は、何をどのように作ればよいかわからないという問題がある。

(2) 作る障壁

プログラミングによりソフトウェアをつくるには、構文や関数の使い方を学習し、事前に知識を持っていなければならない。また、自分の知識でどのようなソフトウェアが作れるのかといった感覚や、どのような手順で開発を進めればよいのかといった、開発の経験も必要になる。

¹ 慶應義塾大学政策・メディア研究科
Graduate School of Media and Governance, Keio University

² 慶應義塾大学環境情報学部
Faculty of Environment and Information Studies, Keio University

ソフトウェアを完成させるまでには手間がかかり、ユーザーは完成を前に諦めてしまう可能性がある。この点について宮下は、プリセットで派手な効果のある処理が行える命令を用意することで、ユーザーのモチベーションを保つことに成功している [3]。このように、簡単に見栄えのするソフトウェアをつくれるようにすることが、気軽なソフトウェアづくりに求められる。

また、プログラミングをより簡単にする試みとして例示プログラミングやビジュアルプログラミングがある。例示プログラミングは実行結果や動作例をユーザーが例示することでプログラミングが行えるというプログラミング手法である。ビジュアルプログラミングは、図やグラフといった要素を用いてプログラミングを行うプログラミング手法である。例示プログラミングやビジュアルプログラミングの手法を使ったプログラミング環境に原田の Viscuit がある [4]。Viscuit はアニメーションを記述するための言語で、柔らかい書き換えという原理に基づいて図を配置することで図の動作を生成でき、ワークショップを通じて児童がプログラミングを遊びのひとつとして楽しく遊んだことが確認されている [5]。このように、例示プログラミングやビジュアルプログラミングといった、コーディングに依らないプログラミング手法を用いることが、気軽なソフトウェアづくりに求められる。

(3) 配る障壁

ソフトウェアの完成後は、インストーラーの生成や Web サイトへの登録といったリリース作業を行う必要がある。インストーラーの生成アプリケーションの使い方を学習する必要があり、Web ページを作ったりバイナリデータの登録といった作業が求められる。

また、リリース後もバグの修正やセキュリティ問題への対処といった作業があり、開発者の手間が多い。

2.2 気軽なプログラミングの可能性

このようにプログラミングを行い、ソフトを頒布するまでには、障壁がある。これらの問題を解決することで、プログラミングをより気軽にし、誰でも楽しさを体験できる環境をつくることのできる可能性がある。

作り始めるまでの障壁のひとつである、何を作ればいいのか分からないという問題を解消するため、本研究は開発できるソフトウェアのジャンルを限定することで、どのようなアプリケーションを作れるのかユーザーが分かるようにした。

つくれるソフトウェアのジャンルを限定すると、限定の範囲によっては生成されるソフトウェアが類似してしまい、ユーザーの独自性が発揮できないことがある。ユーザーの独自性を発揮できるよう、本研究で試作する開発環境の開発対象のジャンルを、新たに定義する「体験指向ドロソフト」に限定した。

3. 体験指向ドロソフト

本研究で試作する開発環境は、著者が新たに定義した「体験指向ドロソフト」を開発の対象とする。体験指向ドロソフトは、絵を描く行為そのものを目的としており、成果物の完成度を高めることは目的としない。Microsoft ペイントや Adobe Photoshop のような、絵を完成させて成果物を得ることを目的としたドロソフトを「結果指向ドロソフト」と定義する。この結果指向ドロソフトの対義語として、成果物そのものを目的とせずに描いている過程そのものを重視しているドロソフトを「体験指向ドロソフト」と定義する。結果指向ドロソフトと体験指向ドロソフトという概念は、どちらか一方をとる固定的な概念ではなく、結果指向な要素を持つ体験指向ドロソフトも存在する連続的な概念である。

体験指向ドロソフトは描く過程そのものをユーザーがコンテンツとして楽しめるよう、結果指向ドロソフトと比較するといくつかの相違点が見られる。本研究においては、体験指向ドロソフトを、(1) 世界観の設定、(2) 世界観の逸脱を抑えるユーザーインタフェース、(3) 描く行為の支援を目的としない装飾、といった 3 つの特徴を持つドロソフトであると定義する。

3.1 特徴

(1) 世界観の設定

結果指向ドロソフトは成果物が重要で、描く過程が効率的になるように工夫された画面設計が用いられる。体験指向ドロソフトでは、画面の中にドロソフトの制作者による世界観が展開され、ユーザーはその世界の中で絵を描く。成果物を最適なものにすることは目的ではなく、描いている際の楽しさや面白さを制作者の表現した世界観を通じて体験することが目的となる。例えば、黒板を再現してチョークの粉が落ちるといった演出を含む黒板ドロソフトや、砂浜をテーマに波打ち際の演出を再現した砂浜ドロソフト、といった事例がある。

(2) 世界観を反映したインタフェース

結果指向ドロソフトは成果物を効率的に構築できることを目的に、WIMP やボタンなどの OS 標準のインタフェースやそれに似せたインタフェースが効果的に用いられている。一方、体験指向ドロソフトでは、インタフェースに世界観を反映するために外観や動作に独自性のあるインタフェースを用いる。

例えば、結果指向ドロソフトにおいて描画ツールの切り替えをボタンを使って行うのに対し、体験指向ドロソフトにおいては、黒板ドロソフトなら粉受けの上のチョーク、砂浜においてはスコップや熊手といった、アイコンとして提示されている事例がある。

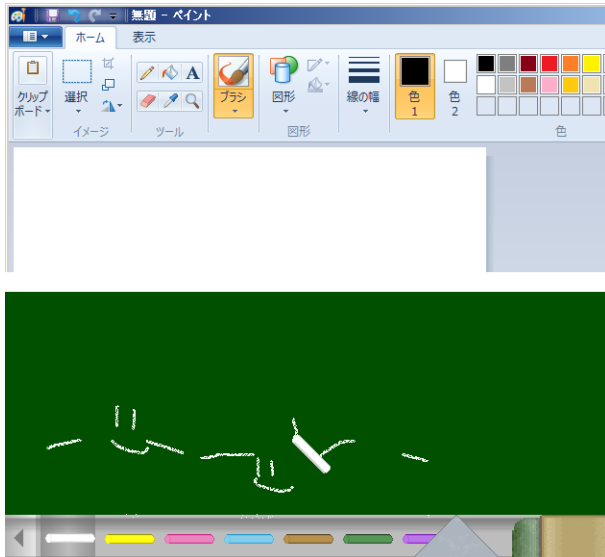


図 1 WIMP を活用したインタフェースの例 (上) と、世界観が反映されたインタフェースの例 (下)

(3) 描く行為の支援を目的としない装飾

結果指向ドローソフトは、カーソルの座標を示すルーラーや透過色を示す市松模様といった、描く行為を支援してよりよい成果物をつくるための装飾を持つ。一方、体験指向ドローソフトは絵を描く過程の体験を重視するため、体験の強化を目的に、成果物の完成度を高めることを目的としない装飾を持つ。



図 2 描く行為を支援する装飾の例 (ルーラーと透過を示す市松模様)

体験を強化する演出には、静的なものと動的なものがある。静的な演出は、画面の構成要素によって表される。例えば、黑板ドローソフトであればツールパレットを粉受けの画像で表現したり、砂場ドローソフトであればキャンバスにあたる砂場の周りに柵を設けたり、砂をいじるためのスコップや熊手をツール選択のインタフェースにしたりする、といった演出である。動的な演出とは、画像のアニメーションや効果音の再生といった演出により実現されるものである。例えば、黑板ドローソフトにおいて線描開始時にチョークが黑板にぶつかる音を鳴らして線描中にチョークの粉を落したり、砂浜ドローソフトにおいて打ち寄せる波をアニメーションと効果音で再現したりする事例がある。

3.2 事例

システムを構成する要素を探るため、体験指向ドローソフトの事例を収集した。本稿では、一般に配布されている3つの事例を紹介する。

3.2.1 AKI 黑板

AKI 黑板 [6] は、秋山によって開発され、2001 年よりフリーソフトとして公開されている黑板シミュレータである。

このソフトは「黑板」という題材をモチーフにした世界観が設定されている。また、粉受け上に置かれたチョークを選択すると、チョークの色を変更でき、チョークの粉が落ちるといった描く行為の支援とは無関係の装飾がある。そのため、体験指向ドローソフトであると分類できる。図 3 は AKI 黑板を利用して絵を描いている画面のスクリーンショットである。

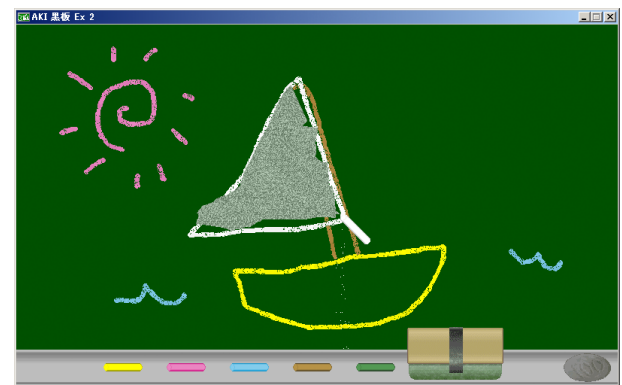


図 3 AKI 黑板 [6] で絵を描いた例

3.2.2 版画・彫る銅板レリーフ

「版画」[7] と「彫る銅板レリーフ」[8] は株式会社フィジオスにより公開されているドローソフトである。木の板や銅板といったベースとなる素材に対して掘削を行う。版画の場合は掘削結果を紙に転写して作品を仕上げる。銅板レリーフの場合は、掘削後の銅板に対してインクを付着させることで陰影をより目立たせることで作品として仕上げる。

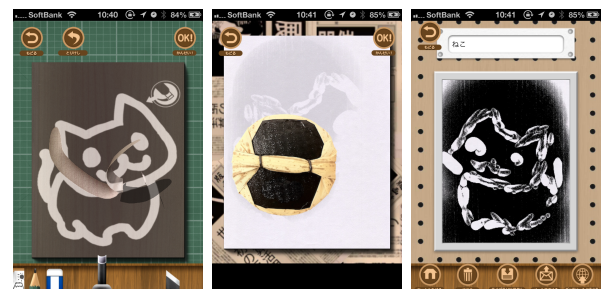


図 4 版画 [7] の彫る画面 (左) と刷る画面 (中央) と完成図の画面 (右) (画面の一部を加工)

3.2.3 Sand Art - Simulator Based Drawing(iOS)

“Sand Art”[9]は、指や掃除機を使ってガラス板にまいた砂に絵を描くことができる、サンドアート体験ソフトウェアである。物理エンジンを搭載し、あたかも本物の砂のように描き味を楽しむことができる。図5は、Sand Artによって描いた絵の例である。

このソフトは「砂」という現実世界のモチーフを利用した世界観があり、木目を基調にした独自のインターフェースを持つ。また、Sand Art独自の物理エンジンにより砂がリアルに動いてアニメーションするといった描画を支援しない演出が組み込まれている。そのため、体験指向ドロソフトと分類できる。



図5 Sand Art[9]で絵を描いた例

3.3 体験指向ドロソフトを構成する3つの要素

事例収集を元に、体験指向ドロソフトをつくるための開発環境を試作するにあたり、体験指向ドロソフトを構成する要素を3種類設定した。ペンとボードとオーラである。

ペン

描く時に用いられる、道具となる要素を構成する。機能を切り替えるアイコン、描画中のカーソル、描画されるブラシのテクスチャといった要素をペンが持つ。例えば、黒板風ドロソフトの場合、チョークと、チョークによって描かれる軌跡を示す。

ボード

描かれる対象となる平面を指す。背景画像や、描画範囲に関する情報を持つ。例えば黒板風ドロソフトの場合、黒板の板そのものを示す。

オーラ

描画時に発生する動的な演出。パーティクルや、音響効果を持つ。例えば、黒板風ドロソフトの場合、チョークの粉やチョークで描いた際の音を示す。

4. 体験指向ドロソフトの開発環境

体験指向ドロソフトの製作と頒布を通じて、ソフトウェアづくりを気軽に体験することができる開発・実行環境として、Expressoを試作した。Expressoは、3つの要素を設定して体験指向ドロソフトの世界観を構築することで、独自性のある体験指向ドロソフトをつくることができる。Expressoが基本的なソフトウェアとしての構造を提供するため、ユーザーは独自性に直接関わる部分の開発に集中でき、ソフトウェアづくりにおける敷居を下げている。

4.1 設計指針

Expressoを使ってソフトウェアづくりを気軽に体験できるようにするために、(1) 開発範囲の限定、(2) オンライン開発の導入、(3) 例示プログラミングの導入、(4) メニュー型のウィザードの導入の4つの指針を設定した。

(1) 開発範囲の限定

Expressoは、体験指向ドロソフトの開発のみを対象とした開発環境である。このように開発対象を限定することで、アプリケーションとしての目的がユーザーにとって分かりやすくなるようにした。また、開発対象を限定することで、開発しなければならない要素を限定できるようになった。Expressoはペン・ボード・オーラの3つに開発対象を限定することで、ユーザーはどこまでも開発を続けることなく、明確な開発目標の中で開発できる。

Expressoを使って生成されるアプリケーションの範囲には一定の限りがあるものの、HTMLやJavaScriptといったWeb標準の技術を用いたアプリケーションが生成される。そのため、ユーザーは生成されたアプリケーションを自由に拡張できる。

(2) オンライン開発の導入

開発された体験指向ドロソフトはサーバー上に登録され、実行時に体験指向ドロソフトとしてHTMLやJavaScriptのコードが生成される。

これにより、即座に他のユーザーが利用できるURLが発行されるため、インストーラの生成やパッケージングといった頒布時の手間が軽減される。また、セキュリティ対策はサーバーサイドで対応可能なので、開発するユーザーはセキュリティ対策やバグの修正といった煩雑な問題を考慮する必要がない。

(3) 例示プログラミングの導入

Expressoによって開発できる体験指向ドロソフトの構成要素の1つである「オーラ」は、開発された体験指向ドロソフト上で、線描に合わせて動きを持つ。この動きの変化が複雑であっても簡単に記述できるよう、例示プログラミングを導入した。

(4) メニュー型のウィザードの導入

Expresso は、メニューをたどって開発する、メニュー駆動型の開発環境である。これは、安村の提唱する Programming 2.0 に則ったものである。

Expresso の開発環境を起動すると、画面上部に現在制作中のドローソフトのプレビューが表示され、画面下部にはボードとペンの生成画面に移動するメニューが表示される。ボードを生成する画面では、写真撮影か写真のインポートが行える。ペンを生成する画面では、写真撮影か写真のインポートによりキャプチャした画像を元にテクスチャの範囲を指定できる。また、オーラを指定する画面への遷移はペンを生成する画面にあり、ペンとオーラが1対1で対応している。

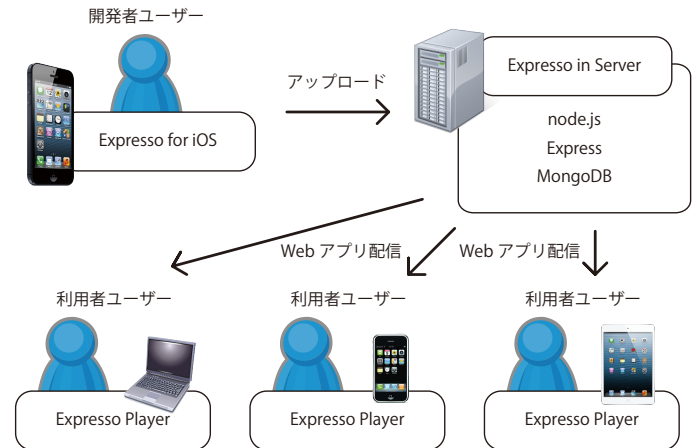


図 7 Expresso システム構成図



図 6 Expresso のメインメニュー

4.2 システム構成

Expresso における体験指向ドローソフトの開発環境は、Apple 社の iPhone 上で動作する。一般的なアプリケーションと同様の方法でインストール可能で、パスを通すといった面倒な設定を必要とせずに体験指向ドローソフトの開発を始められる。iPhone 上で開発されたデータはすべて、Web サーバー上に設置されたサーバーにアップロードされる。

Expresso によって開発された体験指向ドローソフトは、利用ユーザーには HTML と JavaScript と CSS の組み合わせによる Web アプリケーションとして配信される。配信後、JavaScript によって画像は適切な加工を受けて利用される。

4.3 機能

4.3.1 写真撮影による素材収集

体験指向ドローソフトを構成する画像素材を、スマートフォンのカメラ機能とギャラリーからのインポート機能を用いてすぐに収集できるようにした。これにより、素材を作り込まずに、気になったテクスチャを収集してドローソフトを構成できる。

ボード用にキャプチャされた画像は、そのままボードとして利用される。ペンのテクスチャとしてキャプチャされた画像は、必要な範囲を選択することで描画時のテクスチャとして利用できる。

4.3.2 なぞりビュー

オーラの動きを簡単に指定できるようにするため、オーラの動きをなぞって例示することで挙動をプログラミングできる「なぞりビュー」を導入した(図 8)。これは、画面をタップしたままなぞって指を画面から離すまでを 1 ストロークとし、なぞるたびにストローク情報をスタックする。実行時には、スタックされたストロークの中からランダムにストロークが選択され、挙動が再現される。

なぞるだけで動きが指定できるため、細かい操作を必要とせず、スマートフォンのような画面の小さい端末でも簡単に操作できる。また、ストロークをランダムに選択するため、大量のストロークを指定すれば複雑さが増す一方、すべてのオーラに同じ動きを持たせる場合には、1つのストロークだけを指定すればよい。

この仕組みを利用することで、例えば下に落ちるような動きを何度か入れてランダムなチョークの粉の動きを再現したり、上に昇る動きを何度か入れることで、煙を再現することができる。また、その場で左右に揺らすことで漫画表現における輝きを動的に再現することもできる。

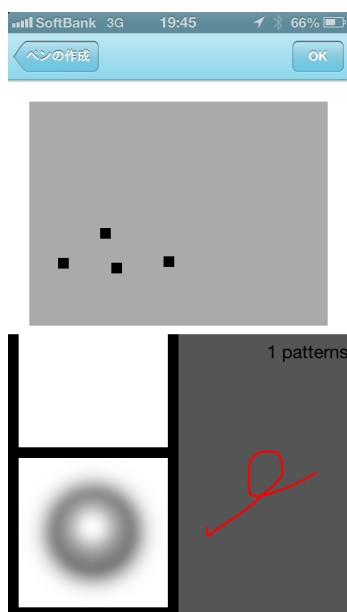


図 8 なぞりビュー

4.4 開発事例

実際に Expresso を使い、いくつかの体験指向ドロースフトを試作した。Expresso によってつくられた体験指向ドロースフトの例を紹介する。

図 9 は、黒板風ドロースフトをつくった例である。黒板の写真を元にボードをつくり、チョークで描いた線の写真を元にペンをつくり、プリセットで用意されている白いドットをオーラとして下方にゆらゆらと落ちる様を設定した。



図 9 Expresso でつくった黒板風ドロースフト

図 10 は、紙を半田ごてで焦がすドロースフトをつくった例である。紙の写真の元にボードをつくり、机の木目を小さくキャプチャすることで焦がした様を描くペンをつくり、プリセットで用意している煙をつかって、上方にゆらゆらと昇る様を設定することでオーラとした。

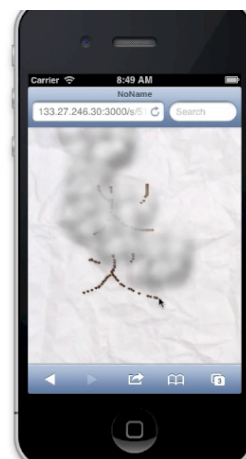


図 10 Expresso でつくった紙を焦がすドロースフト

図 11 は、砂浜風ドロースフトをつくった例である。一枚の砂浜の写真から、ボードをそのまま設定し、ペンを砂浜の中に写り込んでいる落書きを元に生成した。ボードとペンのみで十分雰囲気を用意できたため、オーラは設定していない。

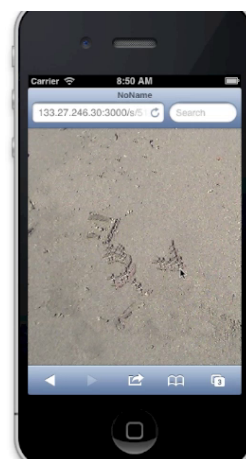


図 11 Expresso でつくった砂浜風ドロースフト

5. 考察

本研究では、体験指向ドロースフトを新たに定義し、ジャンルを限定した気軽なソフトウェアづくりが体験できる開発環境 Expresso を試作した。体験指向ドロースフトと気軽なソフトウェアづくりの環境について、考察したい。

5.1 体験指向ドロースフトとゲームの類似性

慶應義塾大学 SFC 研究所による Open Research Forum 2012 にて本研究で試作した Expresso を展示したところ、来場者から体験指向ドロースフトはゲームに類似しているという旨を指摘された。体験指向ドロースフトは、絵を作り出すことは目的ではなく、絵を描く過程そのものの体験を強化することが目的のソフトウェアである。体験指向ドロースフトに類似した、体験を強化することに重きをおく

ソフトウェアのジャンルに「ゲーム」がある。コンピュータゲームの多くは、コンピュータという特性を活かしてエフェクトやサウンドといった演出が取り込まれている。この類似性に基づいて、本研究で定義した体験指向ドロソフトがゲームであるかどうかについて検討したい。

ゲームの定義について Juul は、過去の 7 つのゲームの定義を踏まえた上で、以下の 6 つの条件を満たすものとして定義している [10]。

- (1) ルール: ゲームはルールに基づく
- (2) 可変で定量化できる結果: ゲームは可変で定量化できる結果を持つ
- (3) 実行可能な結果に対する評価の割り当て: 異なる結果は異なる評価がある、よい評価もわるい評価もある
- (4) プレイヤーの努力: プレイヤーは結果に影響を与えようと努力する
- (5) プレイヤーは結果とつながる: ゲームの結果が勝てば “happy” 負ければ “unhappy” という感覚によるつながり
- (6) 交渉可能な結果: 同じルール下のゲームは現実世界への影響に関係なくプレイできる

体験指向ドロソフトについて考えると、使える画材に限られるという点で一定のルールがある点や、現実世界への影響に関係なくプレイできるという点は、一致している。一方、その他の条件は合致せず、Juul の条件のうち 2 点しか満たしていない。そのため、体験指向ドロソフトはゲームではないと言える。

5.2 開発フェーズの分離

本研究で試作した Espresso を使って体験指向ドロソフトをつくるには、画像素材を集め、それを元にドロソフトをつくる必要がある。Espresso では、素材の収集フェーズとプログラミングするフェーズが混合していたが、収集フェーズとプログラミングフェーズを分けて常に素材を入れ替え可能なようにすることで、開発者ユーザーのドロソフトの開発順序をより一層自由にできる可能性がある。

本研究において試作した Espresso はスマートフォン上で動作するため、スマートフォンを持っていればどこでも開発が可能である。もし開発のフェーズを分けることができれば、例えば外出先の公園で素材の収集だけ行い、帰宅途中の電車でプログラミングを行う、といったような開発スタイルを実現でき、より気軽にプログラミングを始められるようになる可能性がある。

5.3 気軽なプログラミング環境をつくること

通常のソフトウェア開発ではつくるソフトウェアそのものに注力すればよいが、Espresso のような開発環境を開発するには、柔軟性のある開発環境を開発する必要があるため、ソフトウェア開発者に技量が必要であると感じた。

しかし、例えば作ったソフトウェアのデータを保存する過程や、完成したソフトウェアのデータをアップロードして公開するといった処理は、どのようなソフトウェアであっても共通する点が多い。また、その共通する点はセキュリティ上のリスクがあったり、サーバー運営の知識が必要であるため、開発環境自体の開発の難しさにつながっている。気軽な開発環境そのものが増えるには、開発環境を作るためのプログラミング用のフレームワークが必要だと考えられる。

6. まとめと今後の課題

本研究は、ソフトウェアづくりの楽しさを体験することの敷居を下げることを目的に、特定のジャンルに特化することで気軽にソフトウェアづくりの楽しさを体験してもらえる開発環境 Espresso を試作した。Espresso は、筆者が新たに定義した「体験指向ドロソフト」を開発の対象とする。今後は試作したシステムを公開し、試作したシステムによって開発されたアプリケーションの登場をもってソフトウェアづくりの楽しさを体験することの敷居を下げられるか検証したい。

参考文献

- [1] 安村通晃: Programming2.0: ユーザ指向のプログラミング, 情報処理学会夏のプログラミング・シンポジウム報告集, Vol. 2006, pp. 115-122 (オンライン), 入手先 <http://ci.nii.ac.jp/naid/80018468190/> (2007).
- [2] 増井俊之, 塚田浩二: 全世界プログラミング, 情報処理学会夏のプログラミング・シンポジウム報告集.
- [3] 宮下芳明: プログラミングに対するモチベーションを向上させる新言語 HMMML の開発, 第 51 回プログラミング・シンポジウム論文集, pp. 57-64 (オンライン), 入手先 <http://ci.nii.ac.jp/naid/80020832725/> (2010).
- [4] 原田康徳: 子供向けビジュアル言語 Viscuit とそのインタフェース (セッション 2: 子供向けインタフェース, <特集> 高齢者/子どものためのインタフェースおよび一般), 情報処理学会研究報告, HI, ヒューマンインタフェース研究会報告, Vol. 2005, No. 114, pp. 41-48 (オンライン), 入手先 <http://ci.nii.ac.jp/naid/110002973386/> (2005).
- [5] 原田康徳: ペタ語義: 生徒・児童によるプログラミング-ビスケットの挑戦-, 情報処理, Vol. 53, No. 1, pp. 60-64 (オンライン), 入手先 <http://ci.nii.ac.jp/naid/110008719198/> (2011).
- [6] 秋山博紀: AKI 黒板シリーズ, <http://akiroom.com/freeware/>.
- [7] 株式会社フィジオス: 版画, <http://hanga.phyzios.jp/>.
- [8] 株式会社フィジオス: 彫る銅板レリーフ, <http://www.facebook.com/pages/Carve-the-Copper-Relief/彫る-銅板レリーフ/276517302414823>.
- [9] 株式会社ピー・ソフトハウス: Sand Art for iPad / iPhone / iPod touch - サンドアート Simulator Based Drawing -, <http://psftmobile.net/jp/sandart.html>.
- [10] Jesper, J.: The Game, the Player, the World: Looking for a Heart of Gameness, *Level Up Conference Proceedings*, pp. 30-45 (online), available from <http://www.jesperjuul.net/text/gameplayerworld/> (2003).