

## 53 プログラム実行中にソースコードの変更を反映させる技術の応用

ビジュアルインタフェース研究室 田嶋 航太

### 1. 背景と目的

一般的なソフトウェア開発ツールでは、プログラマーがソースコードの書き込みをした後、プログラムを実行し、その実行結果と想定していたプログラム動作との違いを確認しながらデバッグ作業を行う。この作業を繰り返し行うことでソフトウェアを開発していく。

この際、プログラム実行中にソースコードの変更が行われた場合、その変更内容は実行中のプログラムに反映されることはなく、一度プログラムを終了し、再度プログラムを実行させる必要がある。これは、微小な数値変更が多いコンピュータグラフィックスやゲームなどの制作においては、幾度もプログラムを実行させるため手間となる。

この問題点を解消する手立てを調べた結果、プログラム実行中にソースコードの変更を反映させる技術を用いている開発ツールを発見した。本研究ではその開発ツールに改良を加え、更なる改善を目的とする。

### 2. 関連技術

ビジュアルデザイン用開発ツールである Processing [1]は、Java 言語を基にして作られた独自のプログラム言語での記述により、静止画やアニメーションの描画を行う。Processing 3.0以降に追加された機能の1つである Tweak モードでは、マウスドラッグによる動作でプログラム実行中にソースコード内の図形や色など一部の数値データの変更（調整）が可能である（図 1）。この機能により、前述した微小な数値変化による問題点が解消される。また、Tweak モードを実際に触ってみた結果、ソー

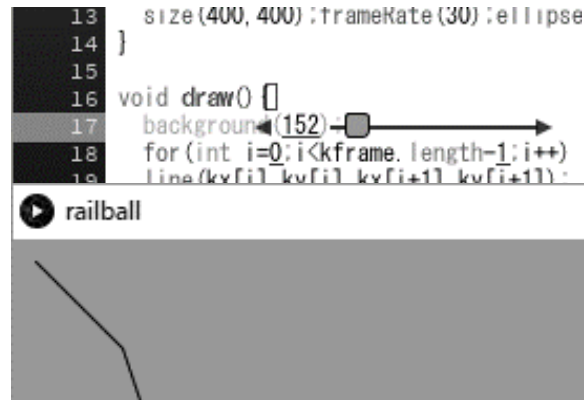


図 1 Tweak モードでの数値変更

ソースコード変更によって実際に変動するプログラムの実行結果を確認することで、ソースコードとプログラム構造の結びつきをより感じることができた。

### 3. 本研究の提案

本研究では、変更したソースコードがプログラムに与える動作をさらに理解しやすいものとする機能の実現を目指す。その提案として、Processing のソースコード変更前の実行画面と、Tweak モードによるソースコード変更中の実行画面とを同時に表示させることにより、ソースコード変更前のプログラム実行結果と比較しながらソースコードの変更を可能とする改良を加える。

### 4. 開発環境

本研究で改良を加える Processing は Java 言語で構成されているオープンソースプロジェクトであるため、プログラム解析をするための特殊なツールを用いらずに改良を行うことができた。また、統合開発環境の Eclipse を使用してプログラムの変更を行い、改良したプロ

グラムのビルドには Processing の公式指定であるビルドツールソフトウェア Apache Ant を使用した。

## 5. Tweak モードの解析

Processing に改良を加えるべく、Tweak モードにおける動作の解析を行った。Tweak モードを実行させる場合、通常のプログラム実行とは異なり、メニューバー上のスケッチタブから実行させる。メニューバーのプログラムコードは JavaEditor.java 内に記述されている。

Tweak モードが実行されると、あらかじめ TweakClient.java 内に記述されたコードを Processing で実行させているコードに付加し、さらにローカルホスト接続での UDP 通信により、Tweak モード時に変更した数値を自身に送り返し、再度記述、実行することでプログラム実行中でのソースコード変更の動作を可能にしていた。

## 6. 実行画面の同時表示の実現

Processing では、通常のプログラムの実行と Tweak モードでの実行のどちらかの操作を行った場合、もう片方の動作を停止してから実行させている。そのため、それらの実行を同時に行うことは不可能である。そこで、プログラム内のメニューバーにあたる JavaEditor.java の記述に変更を加え、Tweak モードでの実行と同時に通常のプログラム実行を行わせることで、実行画面の同時表示を実現させた(図 2)。

## 7. 本研究の課題

実行画面の同時表示を実現させたが、Tweak モード実行直後では 2 つの実行画面が重なって表示されているため、描画し始めた直後の実行画面の比較を行うことができない。また、コンピュータの稼働状況により、両画面の実行時

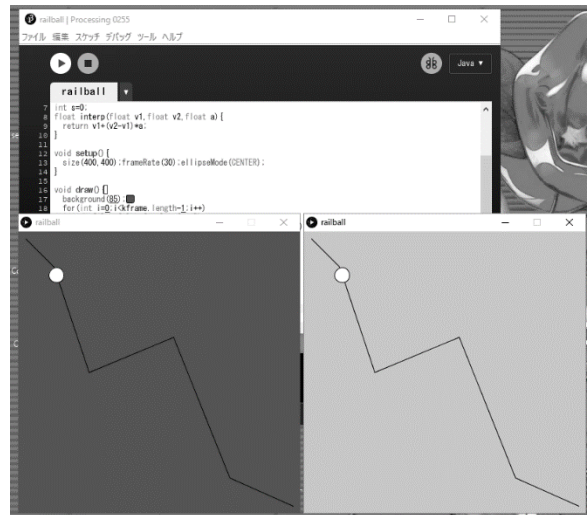


図 2 同時表示された実行画面

間に多少なりとも差が出てしまうことがある。これらは静止画を描画する際には問題ないが、アニメーションのプログラムを実行させた場合には 2 つの実行画面を比較しながらソースコードを変更するという用途とは合わなくなってしまうため、アニメーション制作においてはそれらを改善する必要がある。

## 8. まとめ

本研究では、度重なるソースコードの変更に對する問題点を解消するため、ソースコードの変更を実行中のプログラムへと直ちに反映させる開発ツール Processing を挙げた。

その上で、ソースコード変更によるプログラムの動作変化をプログラマーにより分かりやすく表示するため、ソースコード変更前と変更中の 2 つの実行画面の表示を実現させた。

本研究の目的に対して、アニメーション制作では不適合となる部分もあるが、静止画製作においては目的を果たしていると言える。

## 参考文献

- [1] C. Reas and B. Fry: Processing (version 3.0 以降) , <https://processing.org>