



プロトタイピングのための ビジュアルプログラミング入門

A GUIDE TO VISUAL PROGRAMMING
FOR PROTOTYPING

伊東 実
ITO MINORU
星 卓哉
HOSHI TAKUYA

インタラクション / 3Dアニメーション / リアルタイムグラフィックス / VJ /
マルチスクリーンシステム / ジェネラティブアート / サウンドプログラミング /
データビジュアライゼーション / GPU演算 / プロジェクションマッピング /
コンピュータビジョン / フィジカルコンピューティング / 物理シミュレーション……

表現のアイデアを、すばやく形に——

あらゆるメディアコンテンツを飲み込む
雑食系ツールキット「vvvv」の入門解説書

vvvv
BOOK

BNN

vvvvook

vvvbook – A Guide to Visual Programming for Prototyping

by Minoru Ito, Takuya Hoshi

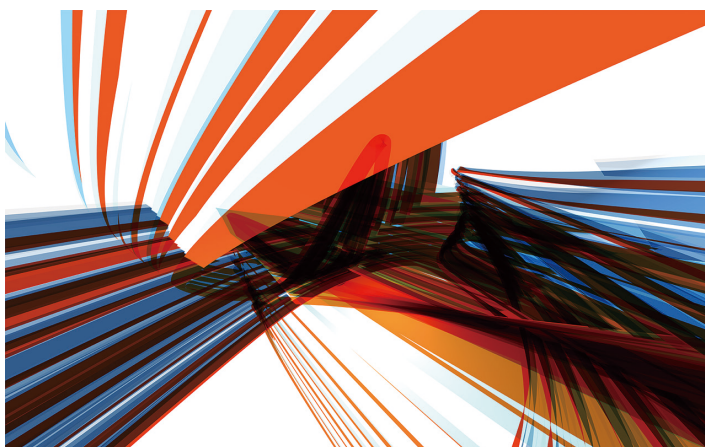
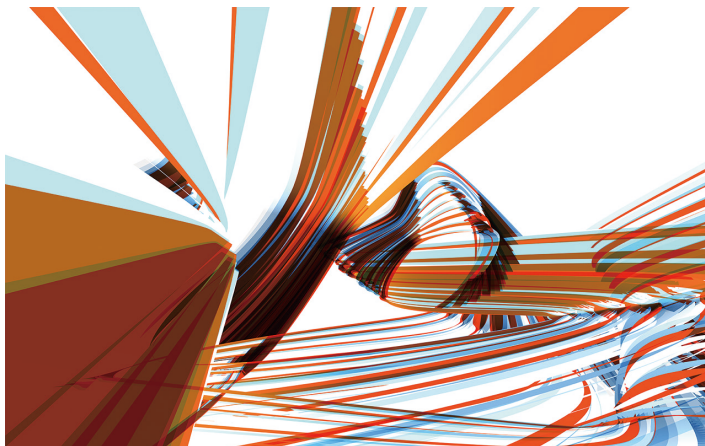
Copyright © 2014 Minoru Ito, Takuya Hoshi

Published in 2014 by BNN, Inc.

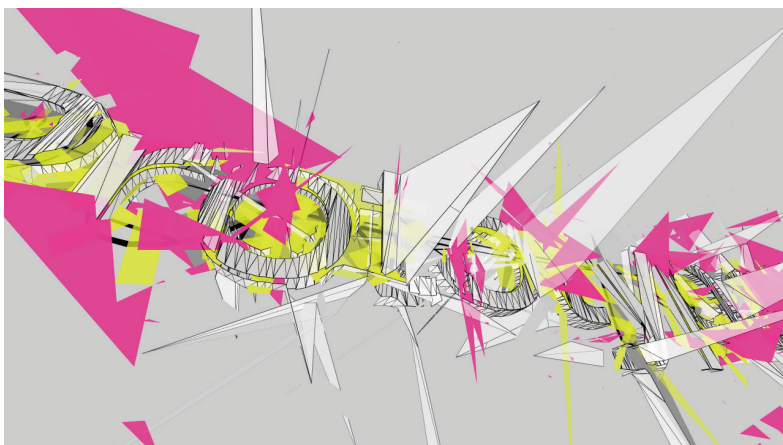
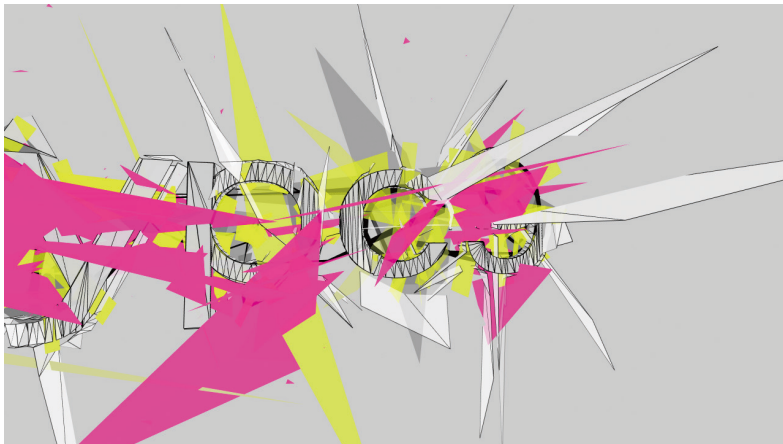
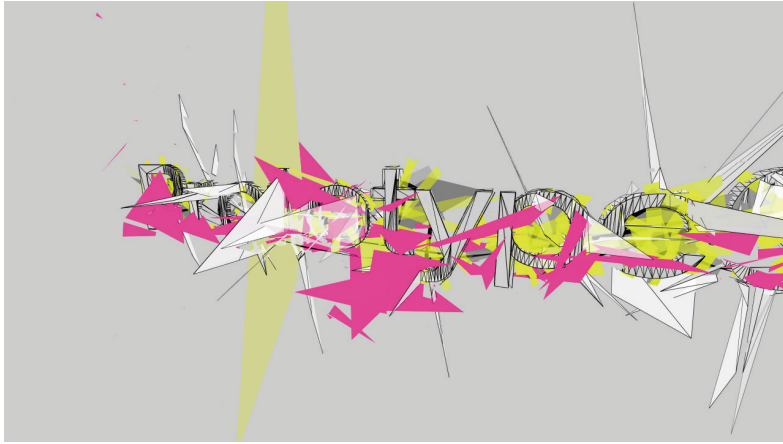
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system, without prior permission in writing from the publisher.



[sweet vectors] — Pedro Mari (defetto), 2009



[lines] — Pedro Mari (defetto), 2008



「proto type」— Pedro Mari (defetto), 2009

はじめに

伊東 実
著者を代表して
2014年1月

本書『vvvvook—プロトタイピングのためのビジュアルプログラミング入門』は、デスクトップアプリケーション「vvvv」の日本初の解説書です。vvvvというアプリケーションを通じて、「表現ツールとしてのコンピュータ」という考え方を知ってもらうことを目的にしています。そのため、プログラミングの入門書というよりは、異国の地を紹介する旅行のガイドブックを読むような気持ちでページをめくってもらえたらと思います。「こんなことができるんだ」「こういうやり方があったのか」といった小さな発見が散りばめられているような本を目指して書きました。なぜなら、私自身がvvvvを使ってそのような発見を何度もしてきたからです。その経験を少しでもみなさんと共有できたらと思っています。

vvvvは、クリエイターが自分たちに必要な表現ツールを自分たちで作るところからスタートしました。どんな環境でも使えるWebアプリケーションとは異なり、コンピュータの性能を最大限に発揮できるように設計されています。そのため、同じような思いを持つユーザーに支持され、個人の研究から著名なアーティストのライブパフォーマンスに至るまで、数多くの現場で利用されています。

似たような背景のアプリケーションやフレームワークは他にもありますが、vvvvはその一風変わった見た目や使い勝手から、特別な存在感を持ったアプリケーションとして地位を築いています。ビジュアルプログラミング言語であること、ドイツ製であること、Windowsでしか使えないこと、ウィンドウには×ボタンすらないこと、日本で使っている人をほとんど見かけないこと…… 私がvvvvに惹かれたのも、そういったある種の「一匹狼」のような特徴が理由だったのかもしれませんが。また、vvvvにはところどころでユーモアのある表現が使われており、ユーザーコミュニティでもみんなが遊び心を持って利用している雰囲気を感じることができます。

実際にvvvvを使っていると、これほど素晴らしく可能性に満ちたアプリケーションがあったのか、と実感するようになります。マウス操作とキーボードショートカットだけでいとも簡単に3Dグラフィックスを生成でき、しかも全てがリアルタイムに制御できます。また、様々なデバイスと連携して大規模なインタラクティブコンテンツを作ることができます。vvvvで作るプログラムには、まるで自分の脳の中を視覚化しているかのような錯覚すら覚えます。

「展示会で体験したようなインタラクティブ・アートを作ってみたい」「Webサイトで見かけたカッコいいグラフィックを描いてみたい」と思っている人には、自信を持ってvvvvをお勧めします。vvvvのアプリケーション、ドキュメント、ユーザーコミュニティは、そんなあなたをきっと応援してくれることでしょう。

また、テキストのソースコードを書くプログラミングに挫折した人、難しいと感じている人にも試してもらえればと思います。徹底的に装飾を取り除いたミニマルなビジュアルプログラミング環境は、あれこれ試しているうちに慣れていき、あっという間にサクサクとプログラムを組めるようになります。事実、テキストベースのプログラミングができなくてもvvvvのビジュアルプログラミングを使いこなすアーティストが数多く活躍しています。

とはいえ、はじめにお断りしておきますが、この本を読めば誰でも一流のクリエイターになれるわけではありません。vvvvはあくまでもツールであり、それを使って何をしたいのが重要であって、実現するためには努力が必要です。また、vvvvで扱う各種メディアやプロトコルなど、それぞれの専門分野の知識も求められます。ガイドブックの例えを借りるならば、vvvvという快適な乗り物に乗って入口までご案内しますが、その先に何があるかはみなさんの目で確かめてもらう、といったところでしょうか。

日本におけるvvvvユーザーのコミュニティは小さく、活用事例もまだそれほど多くありません。だからこそ、「我こそは」とチャレンジする人の登場を期待していますし、一緒に日本からvvvvを盛り上げていきたいと思っています。

さあ、それでは一緒にvvvvの旅に出かけましょう!

本書について

[本書の作業環境]

本書の解説は、Microsoft Windows 7+vvvv_45beta31.2の環境下で行っています。また、3ボタンマウスの使用を推奨します。

[サンプルパッチ]

本書で解説するサンプルパッチは、以下のURLからダウンロードできます。

●<http://www.bnn.co.jp/dl/vvvvook/>

[ご注意]

- ・サンプルパッチは、本書購入者の学習目的にのみご利用になれます。
 - ・サンプルパッチを実行した結果については、著者や出版社のいずれも一切の責任を負いかねます。ご自身の責任においてご利用ください。
 - ・本書に記載されたURL、バージョン等は予告なく変更される場合があります。
 - ・本書に記載されている商品名、会社名等は、それぞれの帰属者の所有物です。
-

日本のユーザーへ——vvvv groupからのメッセージ

vvvv group

joreg,
Max Wolf,
Sebastian Gregor,
Sebastian Oschatz

vvvvの歴史は、1997年頃から始まります。3Dレンダリング用の拡張ビデオカードが一般的になり、ドイツのフランクフルトにあるデザインスタジオMESOは商業見本市や美術館に向けてインタラクティブな展示や空間を開発していました。プログラミングは複雑で、その当時最先端のビデオカードが大きな可能性を提供しても、実験の促進にはなかなか繋がりませんでした。この状況は、オーディオプログラミングでは違っていました。というのは、ビジュアルプログラミング環境であるMaxとPdは実験的な音響生成を簡単に楽しいものにしたのです。しかし、MaxとPdはビジュアルを出力する機能が限られていたため、グラフィック出力専用のソフトウェアモジュールは、一般的なプログラミング言語で開発されていました。

様々な理由から、MESOに参加した多くの開発者たちは、具体的で明確な課題の実装以上のことに常に強い関心がありました。私たちは直面した問題を一般化し、可能な限り普遍的に適用可能な解決策を探りたかったのです。結局のところ、また今までと異なるようで似通った問題に直面することはわかっていたし、何度も同じような仕事をしなければならない状況は望んでいませんでした。だからそれぞれのプロジェクトを前例のないものであるかのように取り組む代わりに、それらを最小限のパーツに分解し、独立した解決策を考案したのです。

あとでわかったことですが、MESOの驚くほど多くの人びとがビジュアルプログラミング、つまり特徴のある「視覚的な」魅力という長所によってモジュール的な思考を促進させるアイデアにも興味を持っていました。小さな問題に対する個別の解決策は閉じたボックス(vvvvでは「ノード」と呼ばれるもの)に収めることで、より大きな問題を解決するために様々な方法で組み合わせることが可能となります。

既存のコンセプトとソフトウェアモジュールを集め、私たち独自のアイデアを提供することでその範囲を拡張し、さらにはユーザーが自身のツールを開発できることを目的としたツールを私たちが実装する——、そのような終わりのないミッションを自分たちに課すという明確な意義をもたらしたのです。こうしてvvvvは、MESOでの取り組みから生まれました。2006年のはじめ、vvvvはこの世界の問題のさらに具体的ではない観点を探して、その能力を発揮し飛び立ちました。10年以上の開発を経た現在、vvvvは巨人の肩に飛び乗り、その有利な視点から、コンピューティングにおける人気のないはみ出し者であるビジュアルプログラミングに関して、今まで以上に多くの潜在能力が見えてきました。この確信に従って、私たちはvvvvを他の人たちが頼れるような巨大なプラットフォームに変化させるよう取り組んでいます。

最初はビジュアル出力に明確に焦点を当てた「多目的のツールキット」としてデ

ザインされていましたが、現在 vvvv は汎用プログラミング言語として知られ利用されることを目的としています。何よりもビジュアルプログラミングは、私たちがあらゆる分野のソフトウェア開発者に楽しんでもらいたい多くの優位点を提供しています。

私たちはドイツで vvvv を開発しています。その効率性の高さ、偏見的にはエスプリの欠如で知られている国なので、私たちの問題解決のアイデアが他の文化圏でも流行っていることを非常に喜んでいます。この本は vvvv に関する書籍のうち、2番目に出版されるものです。もしヨーロッパ人が持つ日本人へのステレオタイプなイメージ——完全主義者で革新者であるというもの——が事実であるなら、おそらく我々は良い仲間なのではないでしょうか。

vvvv の開発者一同、読者の皆さんがこの本で学び、プログラミングを楽しんでくれることを願っています。

To vvvvook

The history of vvvv begins around 1997. Video cards with accelerated 3D rendering were becoming popular, and the design studio MESO (Frankfurt, Germany) was developing interactive exhibits and spaces for trade shows and museums. Programming was complicated, and although state-of-the-art video cards offered immense possibilities, they hardly encouraged experimentation. The situation was different in audio programming: the visual programming environments MAX and PD made experimental sound generation easy and fun. But they had limited visual output capabilities, and so dedicated software modules for graphical output were being developed in conventional programming languages.

For a wide range of reasons, all developers involved with MESO had always been strongly interested in going beyond the implementation of concrete and defined projects. We wanted to generalize problems we encountered and explore potential solutions that would be as universally applicable as possible. We knew, after all, that we'd soon face a different but similar problem, and then we wouldn't want to have to do much of the same work all over again. So instead of approaching our projects as though each were unprecedented, we broke them down into their smallest parts, for which we then devised standalone solutions.

As it turned out, a surprisingly large number of people at MESO were also interested in the idea of visual programming, which promotes such modular thinking by virtue of its distinctive—visual—appeal. Independent solutions to small problems can be wrapped up in closed boxes (in vvvv, they're called "nodes") and then combined in different ways to solve larger problems.

So it made obvious sense to bring existing concepts and software modules together, expand their range by contributing our own ideas, and then apply ourselves to the interminable mission of implement our own tool designed to enable the user to develop his own tools in turn. That's how vvvv grew out of the work done at MESO. In early 2006, vvvv spread its wings and flew off, looking for even more un-concrete perspectives on the problems of this world. Today, after over a decade of development, vvvv stands on the shoulders of giants, and from this vantage, we see more potential than ever before in visual programming, that unpopular stepchild of computing. True to this conviction, we are working to turn vvvv into a giant platform for others to stand on.

Originally designed as a "multipurpose toolkit" with a clear focus on visual output, vvvv now aims to be seen and used as a general-purpose programming language. After all, visual programming offers many advantages that we want software developers in all fields to enjoy.

We are developing vvvv in Germany, a country noted for its efficiency and stereotypical lack of esprit, and so we are delighted to see that our ideas about how to solve problems are catching on in other cultures as well. This is only the second book to be published about vvvv, and if there is any truth to Europeans' stereotypical view of the Japanese—they're thought to be perfectionists and innovators—then we're probably in good company here.

We at vvvv hope you'll enjoy this book, and we sure hope you'll have fun programming in the future.

日本のユーザーへ

—vvvvグループからのメッセージ	006
はじめに	008
本書について	010

CHAPTER1

Introduction 014

1-1. vvvv?	016
1-1-1. vvvvとは	016
1-1-2. vvvvの歴史	016
1-1-3. 本書の対象読者	016
1-2. vvvvを使用した作品例	018
1-3. vvvvの情報を入手するには	028
1-4. vvvvのダウンロードとインストール	030
1-4-1. ライセンスについて	030
1-4-2. 開発環境について	030
1-4-3. ダウンロードとインストール	031
1-4-4. デモプログラムを起動する	033
1-5. サンプルプログラムの紹介	034

INTERVIEW 01 Julien Vulliet	036
-----------------------------	-----

CHAPTER2

Basics 038

2-1. vvvvとビジュアルプログラミング	040
2-1-1. プログラミングとは	040
2-1-2. ビジュアルプログラミング	040
2-1-3. vvvvのプログラミングスタイル	041
2-1-4. マウス操作について	041
2-2. vvvvのGUIと基本操作	044
2-2-1. パッチ	044
2-2-2. メインメニュー	044
2-2-3. ノード	045
2-2-4. ヘルプパッチ	050
2-2-5. インスペクタ	051
2-2-6. IOBox	052
2-2-7. Renderer	056
2-3. vvvvプログラミング	060
2-3-1. グラフィックを表示する	060
2-3-2. 色を使う	063
2-3-3. アニメーション	064
2-3-4. スプレッド	067
2-3-5. 条件分岐	071
2-3-6. インタラクション	073
2-3-7. サブパッチ	075
2-4. チュートリアル	080
2-4-1. ジェネラティブ・アートを作ってみよう	080
2-4-2. パッチを準備する	081
2-4-3. 線を描く	082
2-4-4. 色を変える	084
2-4-5. アニメーションを作る	085
2-4-6. 動きの軌跡を表示する	087
2-4-7. インタラクションを加える	089
2-4-8. 仕上げ	091
2-4-9. 作品を書き出す	093

INTERVIEW 02 Takuma Nakata (中田拓馬)	094
-----------------------------------	-----

CHAPTER3		
Rapid Prototyping	098	
3-1. ラピッドプロトタイピングとは	100	
3-2. メディアを扱う	102	
3-2-1. 画像を扱う	102	
3-2-2. ムービーを扱う	105	
3-2-3. 音を扱う	107	
3-2-4. 3D空間を扱う	113	
3-2-5. フォントを扱う	118	
3-3. カメラの利用と映像解析	120	
3-3-1. カメラ映像の表示	120	
3-3-2. ビデオエフェクト	121	
3-3-3. 映像解析	127	
3-4. IO	130	
3-4-1. プロトコル	130	
3-4-2. vvvvのパッチ同士で通信する	131	
3-4-3. ソフトウェア連携	134	
3-4-4. vvvvとiPhoneで通信する	135	
3-4-5. ハードウェア連携	138	
3-4-6. Kinectを利用する	139	
3-4-7. Arduinoと連携する	144	
INTERVIEW 03 Elliot Woods	150	
CHAPTER4		
Case Study	154	
4-1. 音に同期した映像	156	
4-1-1. 解析	157	
4-1-2. 表現	158	
4-1-3. 外部アプリケーションとの連携	163	
4-2. インターネットのデータを利用する	166	
4-2-1. データ収集	167	
4-2-2. データ解析	168	
4-2-3. データ操作	172	
4-2-4. データ変換	173	
4-2-5. 表現	175	
4-2-6. 表示モードの選択	178	
4-2-7. まとめ	180	
CHAPTER5		
Advanced	182	
5-1. vvvvの上級者向け機能	184	
5-2. v4pファイルの正体	186	
5-2-1. XMLファイルとしての活用方法	187	
5-2-2. パッチのバックアップファイル	189	
5-3. シェーダプログラミング	190	
5-3-1. シェーダプログラミングについて	190	
5-3-2. vvvvのシェーダプログラミング	194	
5-3-3. GPUによるパーティクル表現	199	
5-4. プラグイン	208	
5-4-1. プラグインについて	208	
5-4-2. ダイナミックプラグインの開発	209	
5-4-3. プラグインの公開	216	
5-4. これからのvvvv	218	
イメージクレジット	220	
索引	222	



CHAPTER

1

Introduction

この章では、これからvvvvを使い始める人のために、
vvvvの特徴や、vvvvを使った作品を紹介します。
そして、vvvvをダウンロードして開発環境を整える手順について説明します。

伊東 実
ITO MINORU

1-1.

vvvv?

1-1-1. vvvvとは

本書は、「vvvv」というユニークなアプリケーションについて解説しています。vvvvは、プログラミング言語であり、統合開発環境であり、実行環境でもあります。ここではまずはじめに、vvvvとは何なのか、どんなことに使われているのかをお話します。

今日、クリエイティブな作品制作に特化したフリーウェアは数多く存在しています。ProcessingやPure Data、Blenderといった開発環境は、多くのクリエイターやアーティストたちを虜にしました。さらに、プログラミング経験の少ない人にとっては敷居の高かったC++言語も、openFrameworksやCinderなどの登場によって、高度な表現のツールとして使われるようになりました。

そんな中、ヨーロッパを中心とした世界中で使われているものの、日本ではそれほど普及していないツールとしてvvvvがあります。

そもそも、「vvvv」は何と読むのでしょうか？ 開発元であるドイツでは、ドイツ語のvを表す「ファウ」を4回続けて「ファウファウファウファウ」と発音したり、「4つのv」という意味で「フィーア・ファウ」と呼んだりします。日本語の場合、「ブイブイブイ」「ブイフォー」「フォーブイ」などいくつかの呼び方が存在します。筆者の知る限り、正式な発音は特に決められていません。みなさんの好きなように呼んでください(ちなみに筆者自身は「ブイフォー」と呼んでいます)。

vvvvは、「ビジュアルプログラミング」と呼ばれるスタイルで開発を行います。ビジュアルプログラミングによる開発環境には、Max/MSP、Pure Data、Quartz Composerなどがあります。openFrameworksやProcessingなどで扱うC++やJavaといったプログラミング言語では、テキストを記述してプログラムを作りますが、それに対してビジュアルプログラミングでは、その名の通りプログラムをビジュアル(視覚的)に作ります。プログラムの構造や流れがグラフィカルな図形や線で表されるので、テキストに比べて理解しやすいという特徴があります。

また、vvvvはテキストで記述するプログラミングもサポートしています。C#(シーシャープ)というプログラミング言語を使って、Windowsの.NET Frameworkという開発プラットフォームを利用できます。SDK(ソフトウェア開発キット)も公開されているので、Visual StudioやSharpDevelopなどのIDE(統合開発環境)でプラグインを開発し、世界中のvvvvユーザーに配布することも可能です。さらに、クオリティの高い3Dグラフィックスを作るのに欠かせないシェーダプログラムも、専用のエディタで記述できます。

vvvvのもう1つの大きな特徴は、全ての処理がリアルタイムに行われるということです。一般的な開発環境では、プログラムの編集と実行は明確に区別されています。作成したプログラムをビルドして、実行結果を確認したら再び編集作業に戻る、という流れを何度も繰り返しながら開発を進めます。vvvvは、編集と

実行が同時に行われるので、プログラムを変更した結果は即座に反映されます。そのため、画面に表示されたグラフィックを見ながら、細かいチューニングをリアルタイムに行うことも簡単にできます。

vvvvには、2D/3Dグラフィックス、アニメーション、サウンド、ビデオ、ネットワークなどのメディアを扱う機能が豊富に用意されています。画像解析や物理エンジンといった高度な処理もすぐに利用できます。また、複数のコンピュータによる分散コンピューティングやマルチプロジェクションも標準でサポートしており、本格的なライブ演出にも使われています。様々な外部デバイスと連動させることもできるので、アイデア次第でどんなことでもできると言っても過言ではありません。

1-1-2. vvvvの歴史

vvvvは、Sebastian OschatzとMax Wolfによって、1998年に開発がスタートしました。彼らは、MESOというドイツのフランクフルトにあるデザインオフィスで働いており、最先端のメディアインストール向けのプロトタイピングや開発を行うために高性能なマルチメディアツールを必要としていました。その後、Sebastian Gregorが加わり、中核となるアルゴリズムが数多く生み出されました。2000年にはjoregが、彼の卒業論文のテーマとしてvvvvのGUIをセットアップするために、MESOで開発に加わりました。

当初は社内用の開発ツールとして使われていましたが、2002年12月に最初のバージョンが公開されました。そして2006年、vvvvの開発は、joreg、Sebastian Oschatz、Sebastian Gregor、Max Wolfの4人が「vvvv group」として引き継ぐかたちとなりました。現在は、joregとSebastian Gregorが中心となってvvvvの開発を続けており、年に数回ずつ最新版がリリースされています。

vvvvはオープンソースではありませんが、拡張機能を開発するためのSDKは無料で公開されています。そのため、vvvvユーザーによって数多くの機能が開発されています。また、Webサイトの運営や世界各地でのイベントなど、たくさんの人が様々なかたちでvvvvの発展に貢献しています。

1-1-3. 本書の対象読者

本書は、主に次のような読者を対象として構成されています。

- ・プログラミング経験はないが、コンピュータを使った表現や作品制作に興味を持っている人
- ・メディアやデバイス、各種ライブラリを使ったプロトタイピングを手軽にやってみたい人
- ・普段からFlashやProcessingを使っており、新しい開発環境を試してみたい人

プログラミングに慣れている人にとっては当たり前の話が出てきたり、逆に説明が十分でないと感じるところもあったりすると思います。また、ビジュアルプログラミングよりも、コードをゴリゴリと書くスタイルの方が良いと思う人もいるでしょう。その通りです。vvvvの他にも便利なツールはたくさんあるので、自分に合ったものを使うに越したことはありません。ただ、筆者はvvvvの力を信じています。これほど簡単に、どんなことでも実現できるツールが知られていないのは残念でなりません。この本が、1人でも多くのvvvvユーザーを生み出すきっかけになることを願っています。

1-2.

vvvvを使用した作品例

それでは、実際に vvvv を使用した作品にはどんなものがあるか、見ていきましょう。多くの例が動画で紹介されているので、vvvv でどんなことができるのか是非確かめてください。

『MINI Bodypaint』 — MESO Digital Interiors

● <http://www.meso.net/MINI%20Bodypaint>

vvvv の生みの親でもある MESO Digital Interiors によるインタラクティブ展示です。壁面ディスプレイに映し出された MINI の車体は活き活きと動き、来場者はスプレー型のデバイスで車体にペインティングできます。スプレーを傾けて塗り方を変えたり、バケツにかざして色を変えたりできるようになっています。また、完成した車体の写真はタッチ端末からシェアすることもできます。



『Airbus “Flying Green Exhibition”』 — wirmachenbunt & Mutabor Design

● <http://www.wirmachenbunt.de/exhibitions/airbus-fge>

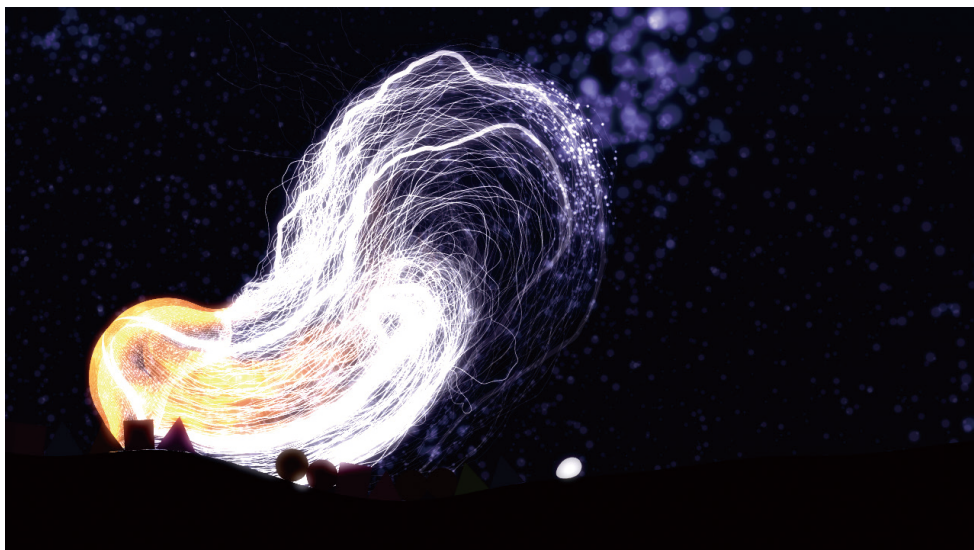
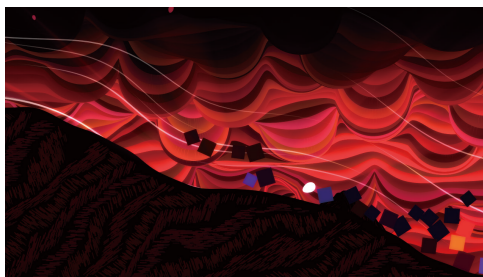
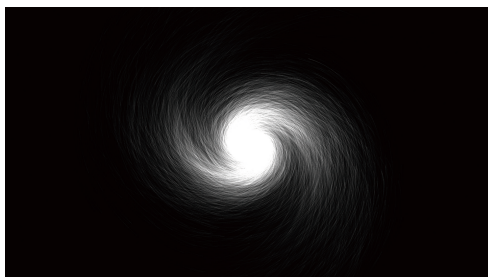
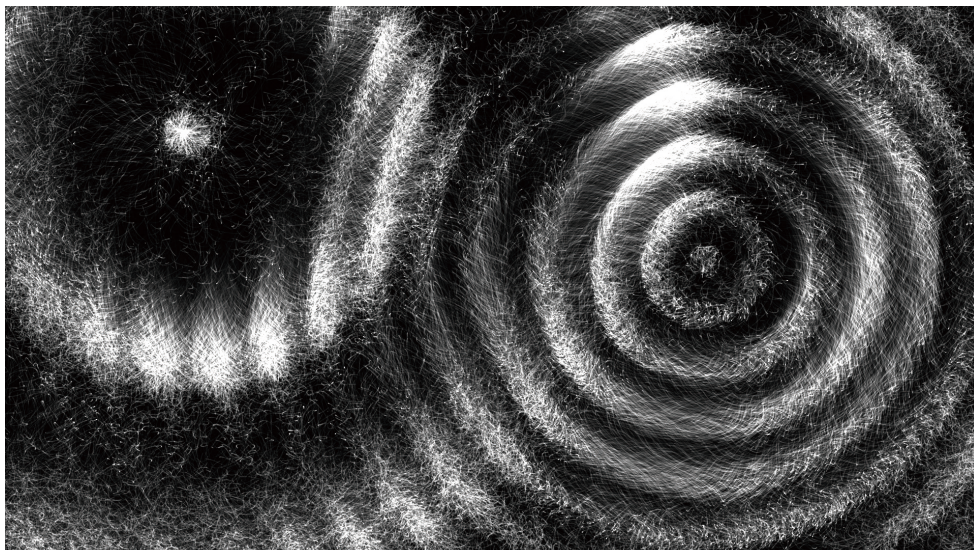
世界の経済やグリーンテクノロジーへの取り組みに焦点を当てて、航空機の将来について紹介する施設の展示物です。タッチパネルやセンサー等のハードウェア制御、飛行機のリアルタイムレンダリング、統計ツール、テキスト管理システムなどを統合して開発しています。

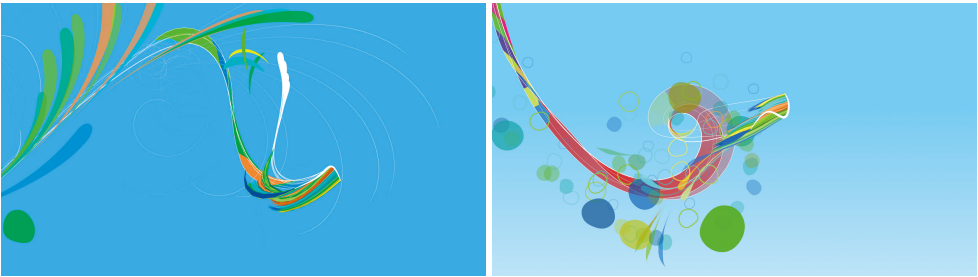


『Genesis』 — Abstract Birds

● <http://www.abstractbirds.com/34019/332756/projects/genesis>

Genesisは、Pedro Mari (defetto) と Natan Sinigaglia (dottore) による、Abstract Birdsのライブオーディオビジュアルパフォーマンスです。管楽器とピアノの入力に反応するオブジェクトをリアルタイムに生成し、物理シミュレーションを用いた高度なアニメーションを展開しています。





『FLUIDIC - sculpture in motion』 — WHITEvoid interactive art & design

● http://www.whitevoid.com/#/main/art_technology/fluidic

ベルリンを拠点とするデザインスタジオ WHITEvoid によるインスタレーションです。12,000 個の球体の群れはワイヤー制御によって有機的に変形し、8 台のレーザープロジェクタによって全ての球体を照らし出すことができます。立体的な光のパターンは見る人の動きに追従して動きます。人の動きは 6 台の Kinect センサーでトラッキングされ、ソフトウェアに vvvv、サウンド生成には Max for Live が使われています。

