

### Puredata(Pd)とは？

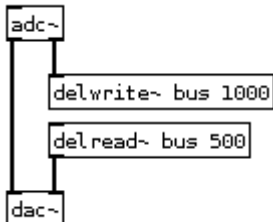
各種機能を持ったオブジェクトをパッチ線で繋ぐことにより、信号を变幻自在に処理できるアプリケーションです。主にオーディオ/MIDIの処理を行うよう設計されていますが、外部ライブラリを使用することにより3D-CGをリアルタイム合成したり、行列演算により画像処理を行ったりもできます。以下に、いくつか簡単な例をば。



[adc~](A/Dコンバータ。すなわちオーディオ入力)を[dac~](D/Aコンバータ。すなわちオーディオ出力)に繋ぐと、サウンドカードの入力がそのまま出力に流れます。



オーディオ入力のLチャンネルが、左右のチャンネルから出力されます。Rチャンネルへの入力はどこにも繋がっていないので、無視されます。



[delwrite~]は、ディレイラインを作成して、同時にそのラインに入力を流し込むオブジェクトです。ここでは“bus”という名前で、長さ1,000ms(1秒)のディレイラインを作成しました。対を成すdelread~は、指定した名前のディレイラインから、指定した時間だけ遅延した音声を取り出すオブジェクトです。左のパッチは、Lチャンネルの入力はスルーさせ、Rチャンネルだけ音声を500ms(0.5秒)遅らせて出力します。

### ~オブジェクトの作り方~



オブジェクトの新規作成時に表示されるのは、破線で描かれた空っぽの箱です。文字の入力待ちを表すカーソルが中にありますので…

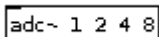


作成したいオブジェクトの名前を入力してEnterを押すと…



そのオブジェクトの入出力が自動的に作成されます。枠の上側に描かれるのがオブジェクトの入力、下側に描かれるのが出力です。

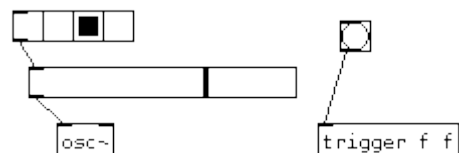
もし枠が破線のままであれば、存在しない名前のオブジェクトを指定した可能性があります。もう一度タイピングを確認してみてください。



オブジェクトの多くが、名前に続いて入力されたパラメータを受け取ります。左の例では、サウンドカードの1,2,4,8チャンネルから入力を受信するように指定しましたので、4チャンネル分の出力が自動的に作成されました。オブジェクトの出力からマウスをクリックしてドラッグすると直線が引けますので、同様に作成したオブジェクトの入力へと繋いでいきます。

各オブジェクトがどのようなパラメータを受け付けるかは、オブジェクトを右クリックすると現れるメニューから'Help'を選択して下さい。簡単にリファレンスを参照することができます。

右の図のように、ライブ・パフォーマンスに役立つスライダーやボタンを配置して、パラメータの変更に使用することも可能です。



## Pd で出来ること

- シンセサイザー、サンプラー、エフェクター etc の作成（オーディオ処理）
- MIDI ミドルウェアの作成（数値データとしての MIDI 信号の処理）
- オーディオに連動する MIDI、MIDI に連動するオーディオ(\*[Video.1 Kausolator](#))
- オーディオ入力に連動する映像出力、映像入力に連動するオーディオ出力 (\*[Video.2 MotionDrums](#))
- Vj ツール (\*[Video.3. PureData Gem](#))
- その他、各種入出力（RS-232, ネットワーク, OSC, etc, etc..）を処理するパッチを、手軽に作成・プロトタイピングを行う（UI などの外観はともかく、とりあえず短時間でアイデアを実現する）プラットフォームになる。(\*[Video.4 Solenoid Concert](#))
- あるいは、更に大規模なインスタレーションの中核として使用。
- すべてがリアルタイムで処理可能、連動可能  
→ 外部入力により、出力を任意に変化させることができる
- 作成したパッチを VST プラグインとして使用(PdVst)

## Pd では出来ないこと、不向きなこと

- CPU の速度が追いつかないほどの処理を要するパフォーマンス。
- FFT の結果を容易にリアルタイム加工するような手段は、現段階では提供されていない。
- 「狭義での西洋音楽」寄りの作法に準じた MIDI データ編集、オーディオファイルを加工して保存する作業などは、できないことはないものの専用アプリケーションを使用する方が合理的。

## Pd の概要

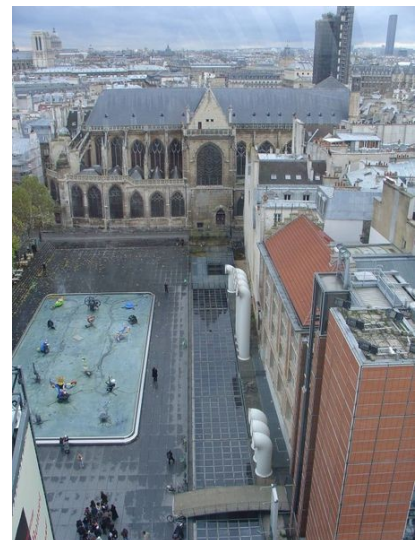
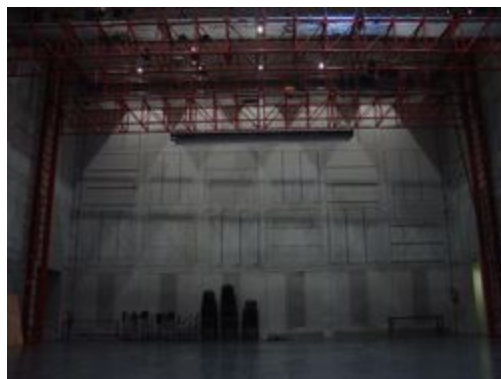
Miller Puckette が IRCAM 在籍中に開発に携わった'Max'の後継。（Max は後に Cycling'74 より製品化、現在に至る）コンセプトや概念は Max と重複する部分も多く、後述の差異を除けば相互に置き換えが可能とも言われる。

現在、Pd 本体(Vanilla Pd と呼ぶ) は Miller Puckette が一人でソースを管理しており、同様に入手可能な有志のライブラリをユーザが各自、追加インストールする形が一般的。

Pd 本体は BSD ライセンス。

オーディオと MIDI の入出力に関するハードは Pd 本体が、ネットワークや RS-232 等、その他の入出力はオブジェクトにより管理されている。(厳密には Pd 本体が、各 OS にネイティブなドライバとやりとりをする)

ちなみに Miller 氏は現在、UCSD（米国カリフォルニア州立大学・サンディエゴ校）にて教鞭を取っていらっしゃるそう。



IRCAM 内のホール（左）と、外観（右）。  
右手に見えるのが IRCAM

## 他のツールとの棲み分け

### Csound (<http://www.csound.com/>)

音声の処理ルーチンを Orchestra ファイルに、イベントが発生するタイミングを Score ファイルにテキストで記述する。敷居の高さと制約から、難易度は高め。現在でこそ目的に応じたツールを選択できるようになったものの、CSound は他者と一線を画すほど長い歴史を有する。また、後続の DSP 系ソフトの多くが CSound で培われた概念をなにかしら踏襲しており、音響生成ソフトのご先祖様とも言われる。MIT とベル研究所で開発。LGPL ライセンス。ほぼ全ての OS に対応するバイナリが入手可能。

### Max/MSP (<http://www.cameo.co.jp/products/cycling74/maxmsp/>)

puredata の遠縁に当たる。  
長い間 Mac 専用だったが、近年やっとこさ Windows 版が登場。  
Pd との違いについては後述する。

### ChuckK (<http://chuck.cs.princeton.edu/>)

イベントが発生するタイミングをテキスト・ファイルに記述。  
CSound などテキストベースの他の言語よりも癖が少なく、比較的取り組みやすい。  
GPL ライセンス。

### Super Collider (<http://www.audiosynth.com/>)

本格的なオブジェクト指向言語で記述。  
プログラミングのスキルは非常に高いが、one liner で無数のオシレータを作成できたりと、非常に強力。以前は Mac 専用のシェアウェアだったが、のちにフリーに。  
Windows、Linux に移植するプロジェクトも存在する。

### vvvv (<http://vvvv.org/tiki-index.php>)

リアルタイム処理が可能なビデオ・シンセサイザーソフト。  
モジュールを繋ぐので、インターフェースの見た目は puredata に近い。  
非商用の利用は無料。…しかし商用⇔非商用の判断基準はかなり曖昧？

### Proce55ing (<http://processing.org/>)

Java のサブセットをベースにした言語で記述する。音声も扱えるが、どちらかというと視覚効果に重点が置かれている。作品は簡単に Java アプレットに変換できるため、ウェブブラウザを対象にした作品を公開できることが他のツールにはない特徴の一つ。  
GPL ライセンス。



Miller Puckette 氏  
(サイトより。…いつの写真?)

Max/MSP と pd の比較 (いずれか一方を推奨するものでもありません…)

	Puredata	Max/MSP
対応 OS	Linux / WinXP / Mac	Windows / Mac
価格	無料	63,000 JPY (Jitter のバンドルは 109,200JPY)
メーカー・サポート	無	有
コミュニティの規模	地味	巨大
コミュニティの傾向	開発者寄り	ユーザ寄り
インターフェース	無骨 (シンプルとも言う)	華やか
変数	float	float, integer
パッチ線 (注*)	直線のみ	直線 / 曲線 / 折れ線
ソースコードの入手	可	不可
映像系の拡張	GEM, GridFlow, etc..	Jitter(別売)
アプリケーションの改変	可	不可
追加オブジェクトの開発	可	可

(注\*) 個人的にはどうでもいい事なのだが、つい先月曲線パッチを使用することの是非についてMLが熱くなっていたので一応記しておく。否定派の意見の多くは、曲線が必要になるようなパッチは、そもそもロジックかオブジェクトの配置に何かしら無駄があるに違いないという考えに基づく。

Pdのメリット

- フリーのため、配布が簡単、低コスト。LiveCD を使用すれば(同様の例に漏れず)、例えば教育機関などにおいて、自宅と学校の環境を統一させるのが容易。
- OS を選ばない
- マイナーなため、今ならオンリーワンになれる…かも。

Pdのデメリット

- 使い方によっては不安定な側面も。
- 日本語ドキュメントの欠如。
- ライブラリ製作者に、なぜか作ったら作りっぱなしの人が多い。  
→ライブラリに関するドキュメントの不足。最新の Pd 本体と整合性がとれないことも。
- 日本でのユーザベースが少ない。極端に少ない。あまりに少な過ぎる。
- アイドル的人物の不在  
→Super Collier のアノ人とか、Max/MSP のアノ人に相当するユーザは現在のところ Pd 界限においては確認できておりません。悪しからず。

## その他の有用と思われるリンク

Miller Puckette の個人サイト (Vanilla Pd の入手)

<http://crca.ucsd.edu/~msp/>

Pd-extended

<http://puredata.info/downloads>

Pd 本体に、主要なライブラリとインストーラをパッケージしたもの。  
各種 OS 用のバイナリが存在する。

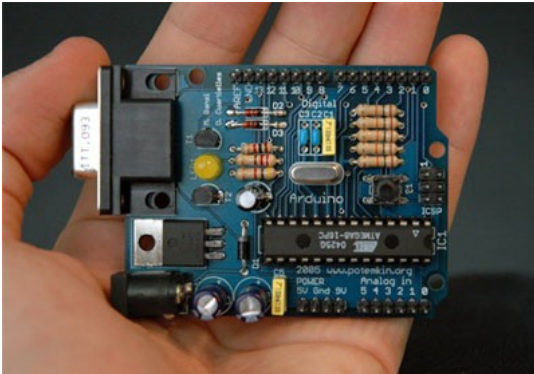
pd-list (本家メーリングリスト)

<http://lists.puredata.info/listinfo/pd-list>

次期バージョンの仕様についての談義、バグレポート、イベント告知など内容は多岐に渡る。  
初歩的な事項に関する質問にも丁寧な解答が多く、言語（英語）さえ障害にならなければ最も有用な情報源の一つ。

Arduino (<http://www.arduino.cc/>)

Atmel 社のプログラマブル IC “AVR” をベースにした、デジタル工作のプラットフォーム。Pd を外部機器と連動させる際に、数ある手段の中でも比較的容易なことから、好んで使用される。キットが市販されているものの、基本的にハード・ソフトともにオープン・ソースであり、Arduino 自体を自作することも可能。（ただし他の電子工作や IC プログラミングを行わない限り、自作する事が不毛と思えるほどコスト・パフォーマンスは高い）



Arduino



pdmtl ライブラリに含まれる、GEM をベースにした画像処理オブジェクトの一例。



## 独習法

Miller が自身のサイトで公開しているドキュメント “[Theory and Techniques of Electronic Music](#).” は、Pdの基本的な使用法と、デジタル音響生成の基礎知識を体得できるよう整備されている。(PDF, HTML バージョンがある)

既にデジタル・シンセシスの知識がある程度備わっているならば、pdのパッチとして提供されているマニュアルを順に追ってもよいと思われる。

マニュアルは大抵、以下の場所に設置されている。

<Pdのインストール・ディレクトリ>/doc/2.control.examples/

<Pdのインストール・ディレクトリ>/doc/3.audio.examples/

: (以下、数章に渡る)

## ヘルプ・ファイルについて

- オンラインのマニュアル中、または自分で作成したオブジェクトは、右クリックで開くメニュー中の “Help” から、簡単にヘルプを参照できる。
- ヘルプファイル自体がPdのパッチとなっており、サンプル内のパラメータを変更して反応を伺ったりすることもできる。(上書き保存しないように注意!)
- 同様に、ヘルプ内に見慣れないオブジェクトがあれば右クリックからヘルプを呼び出し、イモヅル式にオブジェクトの内容を学ぶことができる。
- ただし、外部ライブラリのヘルプは説明のいい加減なものが多い。

この場合にオブジェクトの用途を知る主な方法としては、

1. 自分で数値を放り込んで、出力を観察して推測
2. ソースコードを読む
3. 作者にコンタクトをとる

などが挙げられる。

なお、Pd-extended プロジェクトは有用そうなライブラリの他に、誰かれ問わず作成したドキュメント(ヘルプファイル…つまりPdパッチ)も積極的に取り入れている。一見単純そうなオブジェクトでも、ヘルプファイルの欠如しているものを見つけた場合は積極的に投稿するべし。大抵は大いに歓迎される。

## Pdを使用した大型プロジェクト

'Movable Type'

<http://www.nytimes.com/2007/10/25/arts/design/25vide.html>

New York Times の本社ロビーに展示中の大型作品。無数のディスプレイに、社内データベースから読み込まれた記事が表示される。組込 Linux 上で動作するPdが、コアを制御している。

上記サイトにて、動作の様子と作者インタビューを含むビデオが閲覧可能。

'Spore'

<http://www.spore.com/>

米エレクトリック・アーツ(EA)社のゲーム。オーディオエンジンにPdが使用されており、近年のPCMプレイバック型の音源とは一線を画した演出に利用されている。らしい。



Movable Type

**Pd の作業環境を構築するにあたり必要なもの、あると便利なもの**

- 容易さでは Windows + pd-extended、自由度と導入する敷居の低さのバランスでは Ubuntu + pd-extended がお勧め。（個人的に Mac を所有しておらず、こちらの対応状況はあまり把握しておりません…ゴメンナサイ）
- Audio I/F（できれば Linux での対応が確認されている、ASIO 対応のもの）
  - あらゆる OS において、安定性では RME Hammer Fall が「鉄板」らしい。
  - Firewire のインターフェースは、現段階の Linux では機器認識に手間がかかる。
  - 個人的には M-Audio の PCI 製品をしばらく使用して、現状納得している。
- MIDI I/F(Audio I/F に付属していない場合、大抵の USB MIDI I/F が Windows/Linux とともに安定して動作する)
- OpenGL 対応のビデオカード（逆にこれがないと、3D-CG に関する処理はすべて CPU が担うことになる。廉価なものでも、あるのと無いのとでは雲泥の差。GEM を扱うならば可能性が格段に広がる）

以下、やや主観に基づく OS の比較

	Windows	Linux
導入の容易さ	比較的容易	Linux の知識がなければ困難。 あれば、さほど問題なし。
インターフェース	ほとんど同じ	
オーディオ	レイテンシの改善に限度あり	最小のレイテンシ
RealTime モード	不可能ではないらしい…が？ (注*1)	RealTime カーネルパッチにて対応 (Ubuntu Studio には標準で導入済み)
MIDI	タイミングがシビアでなければ 問題無し (注*2)	問題無し
ライブラリ	FrameStein など一部の例外を除き、ほとんどのライブラリが *NIX 向けに書かれている。CVS から最新のソースを落として手軽に試したいなら、LINUX の方がお勧め。	
Pd 本体のビルド	鬼門	ややオーソドックス

高レイテンシでも問題ない（オーディオの発音タイミングがシビアでない）パッチを組んだり、Pd-extended に含まれるライブラリを使用する程度なら、Windows でも十分作業が可能です。

Pd をオーディオ・エフェクターとして使用するような、低レイテンシが要求される作業を行う場合、LINUX の導入をお勧めします。

幸いなことに、はじめから Pd を含んだ Live CD（CD から直接 Linux を起動する OS）が出揃いつつあります。（PureDyne, Regret, etc..）CD からの起動のために最初は時間がかかるものの、現在使用している PC のハードディスクに触れることなく環境を構築できます（作成したデータは USB メモリ等に保存すれば管理も簡単）

（注\*1）先日 ML にて、タスクマネージャからプロセスの優先順位を「リアルタイム」に変更すると安定性が向上したとの報告があった。ただし、これが Linux カーネルでいうところの RealTime と同義なのか、この設定がパフォーマンス向上に本当に貢献しているのかどうかについては明確な答えを目にしていな。

（注\*2）Pd が使用している MIDI ドライバは、Windows に付属の MS 社製。市販アプリケーションに比べて Windows + Pd の MIDI 入出力がタイミングの精度を欠くのは、多くのソフトメーカーが MS 社のドライバに愛想を尽かし実は独自制作のものを内部で使用しているのに対して、Pd は MS 社製のものをママ使用しているため。Pd の Windows バイナリを立ち上げた時に、「Win での MIDI 使用は大変危険だよ！」という旨のメッセージが表示されるのは、この辺りに由来する… という話を最近見かけた覚えがあるのだが、この注釈の内容自体がうろ覚えによる記述なので、外で吹聴する前に各自ウラを取るようお願いしたいところ。