

このページはNYSDLで公開しています（転載可）

## ゲーム製作用ツール

ゲームやゲーム用素材などに便利なツール  
基本的に情報募集中  
いまんとこ、Windowsが中心です（求む他OS情報）

## ライブラリ・コンパイラなど

<http://gamdev.org/w/>  
ここが詳しいです。

### Gamemaker

2Dゲーム専用の統合開発環境。Click&Createのように、命令ブロックを組み合わせてプログラムを組みます。TonyuSystemのようにキャラクタを並列処理しますが比較的高速です。無料版はアルファブレードなど拡張機能を使うことができません。

### YGS2K

Cっぽい記述でDirectXを使えるスクリプト。制作環境がコンパクトで可搬性があります。2Dゲーム開発に、VCが使えるればDLLを作成するか、本体ソースも公開されているので本体の改造によって機能の拡張が可能です。BulletMLも使用可能になりました。開発は終了しているのでバージョンアップは自分でするしかありません。公式サポートサイトがあります。

<http://ygs2k.xrea.jp/>

### YaneuraoGameSDK.NET

やねうらお氏開発の.NET向けゲームライブラリです。

<http://yansdkdotnet.sourceforge.jp/>

### JAVA

C++に似た(派生した)言語。C++であるような面倒なメモリ管理が不要。絵や音を制御するAPIがすべて本家から提供されています。webやxmlといった新しい技術を比較的簡単に利用できます。エディタはEclipseなど。

### JavaApplet

ブラウザ内で動く、埋め込みオブジェクトの形です。

### JavaApprication

単独で動くアプリケーションで動く形です。配布が面倒らしいです。作ったこと無いので誰かよろしくー

## Doja

Docom用携帯電話iアプリの開発環境。無償で配布されており、個人での配布も可能。  
kenmo氏作成のDoja用ワークフレームiHellがあります。  
<http://gamehell.g.hatena.ne.jp/keyword/iHell>

## C/C++関連

C/C++は現状最速でもっとも高性能な低レベル言語です。基本機能しかないのので、絵や音、ジョイスティック、マウスを使ったりウインドウを出したりするには、WindowsAPIを使うかライブラリを使う必要があります。また、同じC/C++言語でも、開発環境/コンパイラによって使い勝手はグッと変わります。この下はそのライブラリや開発環境について。

## VisualC++ + DirectX

マイクロソフトが提供する開発環境/コンパイラ。DirectXはゲーム専用のWindowsAPI郡です。最新のGPU機能やマシンパワーをフルに使いたい人向け。

## GCC + SDL

GNU compiler collectionことLinuxの標準コンパイラgcc。Windows環境下ではCygwinなどで利用することができます。基本的にコンパイラだけなので、ソースはテキストエディタ等を使って書きます。インストールが面倒。

Simple DirectMedia LayerことSDL。最大の特徴は同じソースでいろいろな環境で動かせることです。が、シンプルにするため、本当に最小限の部分しかなく、いろいろと自作する必要があります。

LinuxやMacや他の環境へ移植できるよ。

## BCC + Luna

C++でDirectXを使うとき、3Dゲームの作成も可能です。

## Delphi

オブジェクトPascal。personalは無料。

「Delphi6.0 Personal」を使ったときの感想ですが、VCL（GUIの部品）が充実しており、GUIツールを作るのがかなり便利でした。あと、コンパイルが爆速。ただ、VSと比べて、RADツールとしての使い心地があまりよろしくなかったのので、別のテキストエディタでコード書いてました。インストールしてないのですが、「Turbo Delphi」が無料で使えるみたいです。

<http://dn.codegear.com/jp>

ちなみにアンディーメンテのゲームは、DelphiとC++Builderで作られているそうです。DirectXを呼ぶライブラリも作られているそうです。

「Quadruple D」<http://www-fu.magma.ne.jp/~hayase/dddd/>

## Flash

ブラウザで動作するため、プレイの敷居が一番低いのはこれじゃないですかねー。ActionScript3となると、記述力や文法チェックが大幅に向上しています。ただしFlash9を要求する。

<http://www.adobe.com/jp/products/flex/>

コンパイラのみであれば無料。Eclipseプラグインは有料。

## RPGツクール2000

Rubyスクリプトを組むことを前提としているRPGツクールXPに比べて、GUIから出来ることが多いので、ゲーム作成初心者におすすめ。廉価版がでており5000円前後で購入できる。また、完成されたシステムは、RPGのプログラム設計の参考にもなるので、プログラムがゴリゴリ組める人にもおすすめ。実行にはランタイムが必要です。

## RPGツクールXP

有名なRPG作成ツールの最新版。素材もシステムもそろっています。Rubyでスクリプトを組むことも可能で、RPGの枠を越えるゲームの作成が可能です。実行にはランタイムを要します。

## HSP

BASICっぽい記述で使えるインタプリタ。開発環境が付属しており、インストールしたらすぐにプログラムが始められます。でかい作品を作るのはちょっときついです。簡単な3Dゲームを作ることもできるようになりました。

<http://www.onionsoft.net/hsp/>

## proce55ing

JAVAアプレットが生成可能なスクリプト。高度な画像処理関数を備えています。3Dを扱う場合はOpenGLを使います。エディタ付き。ちょっと重いです。

<http://processing.jp/>

## DxLib

DirectXで2Dゲームを作るならそれなりにお手軽なライブラリ。VCとセットで使う。Windowsの面倒な所はかなりフォローされています。

ライブラリ群も充実。Windowsベッタリなので移植性は低いです。あと、グラフィックの扱いとか一部に特殊な点とかありますが、スプライト/BGマシンに慣れ親しんだ方にはそんなに障壁ではないかと（狭！）。

ライブラリの更新も頻繁で、バグ報告に対する対応も丁寧です。

<http://homepage2.nifty.com/natupaji/DxLib/>

## Tonyu System

マップエディタ機能のある統合環境がついてきます。

『アクションゲームの作成に適した』と書かれていますが、他のジャンルのゲームも普通に作れます。

Cなどの言語が「1フレーム（1コマ）毎に行う動作」を記述するのに対して、キャラクタの振る舞いを個別に記述し並列処理するので、「キャラクタが生まれてから、死ぬまでの動作」を自然に記述できます。

MP3が使えない、動作が基本的に重い、3Dゲームの作成は難しいなどの欠点があります。M-Twoという波形メモリ音源の曲データを扱えます。

<http://tonyu.kake.info.waseda.ac.jp/>

## Ruby/SDL

<http://www.kmc.gr.jp/~ohai/rubysdl.html>

## Python+Pygame

「日本Pythonユーザ会」 <http://www.python.jp/Zope/>

「Pygame」 <http://www.pygame.org/news.html>

## D言語

<http://www.kmonos.net/alang/d/>

D+SDL, D+SDL+OpenGLなど。

## dHell2

Kenmoさん謹製のD言語用ゲームライブラリ。

<http://www5.atwiki.jp/yaruhara/pages/80.html>

## iPhoneSDK

Objective-C + iPhoneOS, 下位にOpenGL, OpenALを含む。まじめにゲーム作るならOpenGLの知識は欲しいところ。

機器はiPodTouchとiPhoneのみ。480x320の液晶、固定パイプラインのGPU、ARMプロセッサ、膨大なストレージ(8G以上)、独特の入力デバイス(タッチパッド、加速度センサ、GPS、カメラ)など。

Obj-CやiPhoneOS独特の仕様に対応する必要がある。よくも悪くも特殊な環境。

実機転送と配布が有料(年間99ドル)、開発環境(Xcode+iPhoneSDK、シミュレータ)は無料(要登録)

## マップエディター

### Platinum

汎用マップエディター

<http://www.hyperdevice.net/>

## 音楽

### データ制作

シーケンサというのが必要になります。あと気力と時間

### ゲーム内で音楽を使う

- ・ データ容量が限られている場合

まずはフリー作品も多く公開されているmidiを使うのがいいでしょう。

データは大きくしたくないけど、音質にこだわりたい場合

特にテクノ系だとmidiじゃ冴えないですよ、  
そんな時はサンプルベースのフォーマットを使うと良いです。  
このタイプの特徴として、自分の使いたい音色をWAVで用意すれば  
その音が曲でそのまま使えます。  
modやアルトラマリン、  
最近では、制作環境の整った「ピストンコラージュ」も登場しました。  
(ピストンコラージュは、デフォルトでファミコンっぽい音が収録されています。)  
<http://piscolla.hp.infoseek.co.jp/>

#### ・ ファミコンやFM音源に思い入れがある場合

TSSのような昔の音源を再現できるソフト音源や、  
PMD等の昔のフォーマットのエミュレーションができるDLLを使う方法があります。  
特にTSSは導入しやすい(HSPでの実装例もある)のと、  
TSSCPによって周辺環境が整ってきているという利点があります。

#### ・ データ容量に制限が無い場合

mp3やoggなどを躊躇無く使ってしまいましょう。  
最近ではフラッシュ向けのフリー素材も多くあります。  
ただし、GAMEMAKER等のツールではつなぎ目の無いループが出来ない場合がありますので、  
短いループを流したい時はWAVを使いましょう。

mp3,oggは1分で1メガくらいの容量が必要になります。  
(圧縮率にもよりますが)  
圧縮率を上げると、高音がつぶれたりする事がありますので注意して下さい。  
圧縮に関しては不可逆なので、JPEGのようなものと考えて下さい。  
mp3よりoggの方が高音再現に有利な傾向にあるようです。

HSPではHGIMG3のベータ版からOGG形式の音声ファイルを扱うことができます。

#### ・ ライブラリ

FMOD sound system、BASS.DLLあたりが有名です。(ogg,modなどに対応)

## グラフィック

### 2DCG

一枚絵などには自分の好きなツールを使えば良いと思いますが、  
[SAI](#)や[PIXIA](#)あたりがおすすめです。

レタッチツールはPhotoshopが標準的ですが、フリーのものにJTrimやPhotoFiltreがあります。サイズ調整、減色には[Yukari](#)がおすすめです。

## 3DCG

- [Metasequoia Ver2.3.4](#)フリーで使うにしてもシェアウェア版（機能制限）の方が良い。

x形式が必要であれば別途ツールで変換。

ボーンを仕込む場合には、LightWaveや[Mikoto](#)などのツールを使う必要があります。

## ドット絵

[Alfar](#)

[EDGE](#)

[D-pixed](#)

[GraphicsGale](#)

[IDraw3](#)

等フリーでたくさん出ていますので

とりあえず全部落として片っ端から試し、気の合ったツールを使いましょう。

フルカラーに対応しているのはGraphicsGale、シェアウェアのEDGE2などです。

## 爆発

[発色弾](#)

## 効果音

[KanaWave](#)

擬音を文字入力するとそれを元にそれっぽい音を生成します。

ゲームボーイのような懐かしい感じの音になります。

[Beam2002](#)

アナログシンセなので敷居はちょっと高いですが

ランダム生成もついているので、知識がなくても結構遊べます。

## 音楽・効果音編集

[SoundEngine1](#)、[2](#)

効果音の前後の空白を消去して容量を削減したり、

音量の調節・フェードインフェードアウトなどの機能が一通りそろっています。

## 誰が何を使っているか（敬称略）

意味あるかどうか分かりませんが一応。

なるべく発表作品に基づいて載せます

なお、分からないからといって作者に直接問い合わせるのは避けましょう。

ABA：C,D,OpenGL,SDL,Java,ACID,Proce55ing

( ° ▽ ° )ノ：D,C/C++,VC++,HSP等,KanaWave,SoundEngine

井：Gmemaker,HSP,KanaWave,Beam2002,OrionPro,Alfar

本気屋：Gmemaker

D.K : C+SDL,Doja,YGS2000  
Keim : VC++,SDL,Cubase,TSS  
kenmo : D,Doja,Eclipse+Pydev+Python+Pygame  
8086 : D,C+SDL

## ゲームデザイン資料

オンラインで読める資料。

[クロフォードのゲームデザイン論](#)

[カティ・ザーレン、エリック チュメルマン、ルールオブプレイ](#)

[チュートリアル：よいゲームとは？](#)

[OKINIIRI : アーケードゲーム \(レトロゲーム分析系Blog\)](#)

- SDL の名称を間違えていたので、勝手に編集しました…。 -- 8086 (2006-01-21 02:36:38)
- ゲームデザイン資料に「チュートリアル：よいゲームとは？」を追加しておきました。 -- kenmo (2006-03-30 15:28:51)
- [http://web.archive.org/web/20041011141554/www.hh.ij4u.or.jp/~peto/Monster/monster\\_top.html](http://web.archive.org/web/20041011141554/www.hh.ij4u.or.jp/~peto/Monster/monster_top.html)  
モンスタープログラマー列伝。 -- CRS (2007-01-25 18:42:21)