

審 アライコウ



# **『はじめに』**

携帯電話が世間に広まって十数年、人々の生活は大きく変化しました。それ はゲームの世界にも波及し、携帯電話の高性能化に伴って、アクションゲーム、 RPG、シミュレーションゲーム、パズルゲームなど、様々な作品が楽しめるよ うになりました。

文章を読み、選択肢を選ぶことで進行するノベルゲームも、たくさんの作品 が世に出るようになりました。しかし、携帯電話ゲームを制作するには、プロ グラムの知識がなければいけません。これはアマチュアクリエイターにとって 大きな壁となってきました。

そんな中で生まれたのが Novel-Press です。docomo、au、SoftBank の従来 型携帯、そして従来型携帯に代わり市場を席巻しつつあるスマートフォンの Android 向けにノベルゲームを作れるツールです。Android の普及速度は目を 見張るものがあり、今後は新規契約、機種変更ともに間違いなく多くのシェア を占めていくと思いますが、従来型携帯を使い続けたいという人のニーズにも 応えているのがポイントです。

「累計1億台以上の携帯電話市場に向けて、自分の作品を発表できる」

Novel-Press があれば、これが実現できるのです。

特殊な技術は必要ありません。ゲーム制作がまったく初めての人、あるいは パソコン向けで活動してきた人、誰でも使いこなせます。もちろん本書も、可 能な限りわかりやすいよう心がけて執筆しました。読み終えれば、すぐに開発 に取りかかれるでしょう。

読者の方々のクリエイター魂を刺激し、創作の日々を充実させる。本書がそ の助けになれば幸いです。手の平に収まる携帯電話の上に、ぜひ豊かなストー リーを築いてください。

 $\mathbf{2}$ 

# 『目次』

はじめに	2
第1章 Novel-Press とは?	6
モバイル向けのノベルゲームを作れる	7
Novel-Press の機能	9
機能紹介アプリをダウンロード	11
第2章 各プラットフォームの特徴を把握しよう	14
docomo の従来型携帯の特徴	15
au の従来型携帯の特徴	17
SoftBank の従来型携帯の特徴	19
Android 携帯の特徴	20
コラム1・今後の携帯市場はどうなるか	22
第3章 制作準備をしよう	23
Novel-Press のルールを確認する	24
制作環境を整える	28
作業用フォルダを作る	31
タグの書き方を覚える	33
第4章 ノベルゲームを作ろう	35
テキストを表示する	36
テキストを装飾する	40
背景画像を表示する	43
キャラクター画像を表示する	45
描画バッファリングを使う	

メッセージウィンドウを変更する	.52
画像に特殊効果をかける	.55
サウンドを再生する	.57
ウェイト機能を使う	.61
画面を揺らす	.64
画面をフラッシュさせる	.66
バイブレーションさせる	.68
選択肢とジャンプを使う	.70
フラグ機能を使う	.76
プレイヤーに文字列を入力させる	. 80
セーブ機能に制限をつける	. 82
ブラウザを起動させる	.85
各種リストに追加する	.87
コメントでシナリオを見やすくする	.91
スタッフファイルを制作する	.94

## 第5章 アプリを組み立てよう \_\_\_\_\_\_98

開発環境をインストールする	
Java ソースコードをダウンロードする	108
config ファイルを作成する	110
アイコンを作成する	113
i アプリの組み立て方法	114
オープンアプリの組み立て方法	118
EZ アプリ(J)の組み立て方法	122
S!アプリの組み立て方法	126
Android アプリの組み立て方法	130

コラム2・	WEB 版の Novel-Press	139
	· · · · · · · · · · · · · · · · · · ·	

第6章 Java ソースコードを書き換えよう。	140
より細かくシステム調節	
ボタンのテキストを変更する	
タイトル画面を変更する	
メニュー画面を変更する	
セーブ&ロード画面を変更する	
回想画面を変更する	
選択肢の表示方法を変更する	

第7章 作品をリリースする	169
ウェブサイトにアップロードする	
プレイヤーにダウンロードしてもらう	
アプリゲットに登録する	
d マーケット(i モード)に登録する	
Ameroad に登録する	
Google Play に登録する	

タグリファレンス	18	8'	7	1
----------	----	----	---	---

# 第1章

# Novel-Press とは?

# 『モバイル向けのノベルゲームを作れる』

#### ●数少ないモバイル向けのゲーム制作ツール

プログラムを知らない人でも簡単にゲームを作れるようになる。それがゲーム制作ツールで、『RPG ツクール』(開発:エンターブレイン)などが有名です。 パソコンの世界においては、そういった制作ツールがたくさんあり、数多くの アマチュアクリエイターが利用しています。

パソコンユーザーの 9 割は Windows を使用しています。この共通システム のおかげで、たくさんの人に使ってもらえるツールを開発することができたの です。ところが携帯電話は、いろいろな会社が仕様の違う製品を作っており、 どの携帯電話でも動く……そういったアプリケーションを開発するのが難しい のです。したがって、ゲームの制作ツールもパソコンほどには発展してきませ んでした。

しかし、まったく存在しないわけではなく、その数少ないツールのひとつが Novel-Press です。Novel-Press は docomo、au、SoftBank の従来型携帯、そ して Android 端末で動くノベルゲームを制作することができます。どうしても モバイル向けのノベルゲームを作りたい、しかも特定のプラットフォームに限 定されることなく……そういうクリエイターにとっては、まさに待ち望まれて いたツールなのです。

Novel-Press は開発者の公式サイトから無料で入手できます。実際に制作す るとなると、誰かにキャラクターグラフィックや BGM を外注する、素材を購 入するなどして費用がかかることもあると思いますが、基本的な制作コストは ゼロです。

この本を手に取った方は、すでにパソコンといずれかのキャリアの携帯電話 を所持していることでしょう。このふたつだけあれば、最初の準備は完了して いるのです。

7

### ●大規模なマーケットに参入できるように

Novel-Press 作品が動作する従来型携帯と Android 端末は、国内で最大のマ ーケットを形成しています。どれほどかというと、各携帯電話会社の公式発表 をもとに計算すると、ざっと1億台を超えています。あくまでも契約台数なの で、人数換算するとそれよりも下がりますが、国内最大級であることに変わり はありません。Novel-Press を使用すれば、これほどの規模を持つマーケット に自分の作品を発表でき、多くの人々に知ってもらえるのです。

特に現在、Android がかなりの人気を博しており、新規契約数では従来型を 上回ると言われています。この勢いある Android は、従来型携帯と違って世界 で使用されています。その気になれば海外展開もできるということで、なおさ ら魅力的だと思います。

#### ●デメリットは比較的少ない

Novel-Press で不可能なこと、難しいことについても知っておきたい。そう 考える人もいると思いますので、説明しておきます。

#### 【システムに制限がある】

Novel-Press は表現の自由度よりも、簡単に作れることに重点を置いていま すので、多くの点でデフォルト設定が強制されています。次項で説明しますが、 プログラム本体に改造を加えない限りは、大胆なシステム拡張は不可能です。

#### 【デザインに制限がある】

これも同じ理由で、オープニング画面やセーブ・ロード画面を凝ってみたい と思っていても、あまり大幅なカスタマイズは難しくなっています。たとえば オリジナル画像をボタンに使う、などはできません。

デメリットと言えるのは、おおまかに以上の2点です。とはいえ、手軽にモ バイルノベルゲームを作りたい人にとっては、さほど気にならないと思います。

# 『Novel-Press の機能』

### ●多彩な機能の数々

Novel-Press には以下に挙げるような多彩な機能があります。無料で使えて プログラムの知識がいらないのはありがたいけれど、やっぱり機能が第一…… と考える人も、安心できるはずです。

#### 【Novel-Press で可能なこと】

- ・背景画像の表示
- ・キャラクター画像の表示
- BGM、効果音の再生
- ・メッセージの一時消去
- ・メッセージの自動読み進み
- ・メッセージの読み返し
- ・ウェイトの使用
- ・画面を揺らす機能の使用
- ・バイブレーションの使用
- ・選択肢の使用
- ・時間制限付き選択肢の使用
- ・シーンジャンプの使用
- ・フラグ機能の使用
- ・条件分岐の使用
- ・プレイヤーに文字列を入力させる
- ・ブラウザの起動
- ・CGリストの使用
- ・BGM リストの使用
- ・エンディングリストの使用

モバイルゲームはパソコンゲームと違って、いろいろと制約があるものです が、これだけの機能があれば、充分に面白い作品が作れるでしょう。恋愛、ホ ラー、ミステリー……どんなジャンルのシナリオを書くにしろ、あとは作者の 力量次第です。

#### ●シンプルな文法

パソコンのノベルゲーム制作ツールでは、タグと呼ばれるものでシナリオを 制御するのがスタンダードです。タグとは主に英数字の組み合わせによる命令 で、このシーンで背景画像を表示する、キャラクター画像の位置を変える、BGM を流すという風に指定をするのです。

Novel-Press も同様に、タグによってシナリオファイルを作っていきます。 タグの書き方は制作ツールによって大きく異なりますが、Novel-Press の文法 は非常に簡単な部類に入ります。詳しいことは第3章で解説していきますが、 英語を覚えたての中学生でもすぐに理解できるというほどのシンプルさです。

### ●改造ができる

Novel-Press の特色は何より簡単に作れることですが、それだけでなく、プログラム部分を書き換えることで、細かい部分の調節も可能になります。

Novel-Press の本体は、Java というプログラミング言語で書かれています。 この本体部分はオープンソース(一般公開されているソース)となっており、 自由に改造して構わないとされています。

つまり、好みでシステムや画面レイアウトを変更することも可能なのです。 「それって Java の知識がないと難しいのでは?」と思われるでしょうが、数 値をちょっと書き換える程度であれば、特に難しいことはありません。

もちろん、自分やサークルメンバーがプログラムに詳しいということがあれ ば、さらに大胆な拡張をすることもできるでしょう。

# 『機能紹介アプリをダウンロードする』

### ●公式サイトへアクセス

Novel-Press のメリットを理解していただけたでしょうか。それでは実際に Novel-Press を試してみましょう。

まずはパソコンで Novel-Press 公式サイトにアクセスします。

#### 【Novel-Press 公式サイト】

http://houser-home.net/novel-press/top\_top.html

トップページで更新履歴が確認できます。そこから先に進むと、[製作手順] [タグ一覧][備考][掲示板]と4つのメニューがあります。一通り見て回り ましょう。

「Novel-Press のアプリケーションはどこにあるの?」

と迷った方がいるかと思いますが、Novel-Press とはプログラム、ソースコ ードの名称であり、アプリケーションではないのです。アプリケーション実行 型のパソコンゲーム制作ツールに慣れている人は、最初は戸惑うかもしれませ ん。Novel-Press はソースコードをもとにして作品を組み立てるのですが、こ れについては第5章で説明していきます。

#### ●Novel-Pressの機能を試す

メニューの [製作手順] をクリックしてください。[その他] に [サンプルア プリ] とあります。このページから、Novel-Press で使える機能を紹介したア プリをダウンロードできます。

表示されている QR コードを、携帯電話のバーコードリーダーで読み取りま しょう。[DOWNLOAD] のリンクを選ぶと、ダウンロードが開始します。こ のアプリを起動すると、シンプルな緑色の画面が出てきます。そして 4 つのメ ニューがあります。



#### [START]

ゲームをスタートします。

#### [LOAD]

セーブされたデータをロードします。

#### [MENU]

システム設定、CG リスト、エンディングリスト、BGM リストがあります。 【STAFF】

制作スタッフを見られます。

[START]を選ぶと、機能のサンプルを選べます。選択肢で[文字関係][画 面関係][その他]の3種類があり、さらにそこから細かく機能が分かれてい るので、すべて試してみましょう。その過程でCGやBGM、エンディングが 登録されます。何もせずとも各種リストが付いてくるのは、Novel-Pressの特 徴のひとつです。

操作方法については公式サイトの[備考]→[操作方法]で確認できます。 キー入力式の従来型携帯とタッチパネル式のAndroidスマートフォンで操作方 法は異なりますが、いずれも特に迷うことはないと思います。

#### ●メニュー画面

メニュー画面では、次の項目を設定できます。

#### 【文字の表示速度】

10段階で設定。10は瞬間表示。

## 【オートプレイのウェイト】

10段階で設定。10がもっとも長く待つ。

#### 【文字枠の透明度】

10 段階で設定。10 が透明。

#### 【音量】

10 段階で設定。

#### 【バイブレーション】

ONとOFFで切り替えられる。

## 【クリック(タップ)待ちの表示】

ONとOFFで切り替えられる。

#### 【改頁待ちの表示】

ONとOFFで切り替えられる。

# 第2章

# 各プラットフォームの特徴を把握しよう

# 『docomo の従来型携帯の特徴』

## ●i アプリがプレイできる FOMA シリーズ

ゲームを作る前に、各プラットフォームのことをよく知っておかなければな りません。本章の内容をきちんと把握しておけば、「自分の携帯でプレイでき る?」というプレイヤーの質問にも対応できると思います。

docomo の従来型携帯でプレイできるアプリは「i アプリ」といい、これをダ ウンロードできるのは i モードを搭載した FOMA (Freedom Of Mobile multimedia Access)端末です。この端末であれば Novel-Press 作品をプレイ できます。

FOMA シリーズでプレイできる i アプリは、メガアプリと呼ばれています。 容量が 1MB であることから、この名が付いています。メガアプリに対応する ようになったのは、2006 年から 2007 年にかけてで、それ以前の FOMA 端末 は、i アプリの容量上限が 500KB とかなり制限が厳しいものでした。

そして Novel-Press で作れる i アプリの容量上限は、1MB となっています。 この 1MB という容量は、GB が当たり前になっているパソコンゲームなどに比 べるといかにも小さいですが、そこはシナリオや演出でカバーして、面白い作 品を目指しましょう。むしろ制限のある中で、どうすれば容量を削減できるか を考えるのが、クリエイターにとって腕の見せどころと言えます。

### ●Doja プロファイルと Star プロファイル

「自分の持っている端末は、2MB以上のiアプリが OK みたいなんだけど?」 と疑問を持った人がいると思います。

iアプリの実行環境、つまりiアプリを動作させるためのソフトウェアの仕様 は2種類あり、それぞれ「DoJa プロファイル」「Star プロファイル」と呼ばれ ます。2008年頃までの FOMA 端末では DoJa が搭載されていましたが、それ 以降は Star が搭載されるようになりました。2MB 以上が OK というのは、こ の Star なのです。 DoJa はアプリの容量上限が 1MB です。そして Novel-Press は DoJa の仕様 に沿って製作されているので、それ以上の容量の作品を作ることはできません。 Star で制作された作品と比べて、DoJa は容量面で劣るわけですが、2008 年以 前にリリースされた機種でも動作させられるメリットがあります。2008 年以降 のリリースでも、DoJa しか搭載されていない機種が一部にありますので、よ り多くのユーザーにプレイさせたいなら、依然として DoJa が有利です。

Star が搭載される端末には、DoJa も平行して搭載されているため、 Novel-Press 作品がプレイできないという心配はありません。いつかは Doja が搭載されなくなる日が来るかもしれませんが、そうなるとしても相当先の話 のはずです。

# 『au の従来型携帯の特徴』

### ●オープンアプリと EZ アプリ (J)

auの従来型携帯で一般クリエイターがリリースできる Java アプリは「オー プンアプリ」と「EZ アプリ(J)」の2種類があります。

2007 年に登場したオープンアプリは「オープンアプリプレイヤー」というア プリ上で動作します。しかし 2010 年頃になると、このオープンアプリプレイ ヤーを搭載した機種はリリースされないようになり、代わりに 2011 年から、 上位互換に当たる EZ アプリ (J) 対応機種がリリースされました。仕組みがほ とんど同じなので、EZ アプリ (J) 対応機種でも、オープンアプリは問題なく プレイできるのです。

つまりオープンアプリは旧式で、EZ アプリ(J)は新式です。au 携帯向け にリリースしたいと考えるなら、この両者の違いをよく把握しておく必要があ ります。

#### ●容量と動作速度の違い

オープンアプリと EZ アプリ (J) の違いは、主に容量です。オープンアプリ の容量上限は 300KB までとなっており、かつての docomo 携帯のメガアプリ 非対応機種よりもさらに小さくなっています。

300KB 以内でゲームを制作するとなると、グラフィックの画質を落とす、 BGM の数を減らすなど、かなりの削減努力をしなければなりません。それで も容量を削減しきれない場合は、前編後編などに分割してリリースすることが 必要になってきます。

一方、EZ アプリ(J)は上記の制約が取り払われています。容量上限は 2MB で、DoJa で制作された i アプリの 2 倍もの容量です。これだけあれば、充分に クオリティの高い作品が作れるでしょう。また、プログラムの処理速度もオー プンアプリより速くなっていて、より快適にプレイできます。オープンアプリ の欠点がおおむね解消されているのです。 つまり選択肢として、

- 300KB 以内に抑えてオープンアプリ対応機種、EZ アプリ(J)対応機種 の両方をターゲットにする
- ② 容量の大きい EZ アプリ(J)対応機種のみをターゲットにする

この2種類があります。①の場合は一定のユーザー数を見込める代わりに制 限が厳しく、②の場合は容量に余裕がありますが、ユーザー数が少なくなりま す。一長一短なので、好みで決めましょう。

# 『SoftBank の従来型携帯の特徴』

## ●S!アプリがプレイできる SoftBank 3G シリーズ

SoftBank の従来型携帯で動作するアプリは「S!アプリ」といい、現在は SoftBank 3G と呼ばれるシリーズで利用できます。

容量は 1MB で、Doja で制作された i アプリと同じです。かつては機種によって 50KB、100KB、256KB といった制限がありましたが、現在はすべて 1MB となっています。i アプリとして制作したものをほぼそのまま S!アプリとして もリリースするという例が、商業アマチュア問わず見られます。

#### ●一般の制作者は自分のサイトで公開できない

S!アプリの最大の特徴は、自分で用意したウェブサイトへの公開ができない ことです。docomo でも au でもそういった制限はないのですが、これは SoftBank が決めたセキュリティの問題であり、仕方がないものと納得するし かありません。

企業ならば SoftBank が事業者向けのサーバーを用意していますが、一般の 制作者はどうやってアプリを公開すればよいのかというと、SoftBank が認定 したコンテンツアグリゲータ(コンテンツを集めてユーザーに提供する事業者) のサーバーにアップロードします。コンテンツアグリゲータのサービスにアカ ウント登録をして、審査を経て、ようやく公開できるという流れです。コンテ ンツアグリゲータの代表としてはアプリゲットというサイトがあり、第7章で 登録方法を解説します。

自分のサイトにアップロードできないということは、実機でのテストプレイ も逐一コンテンツアグリゲータを通さなければいけません。そのため時間がか かってしまい、i アプリと平行して制作していても S!アプリのみリリースが遅 れることがしばしばあります。

# 『Android 携帯の特徴』

### ●どの携帯会社からもリリースされている

Android 携帯は docomo、au、SoftBank の各キャリアからリリースされてい るため、その分ユーザー数が急増しています。iPhone と並び、今後もっとも契 約数の伸びが期待されるスマートフォンです。

Google が運営する Android アプリマーケット・Google Play でのアップロー ド容量制限は 50MB ですので、Novel-Press で制作する Android アプリの容量 もまた 50MB 以内になります。

ダウンロードに少々手間取るというデメリットはありますが、従来型携帯で は落とさざるを得なかったグラフィックの画質も、パソコンゲーム並みの綺麗 さで使うことが可能です。ボリュームある内容にしたいというなら、Android 向けのみで制作するとよいでしょう。

#### ●画面が縦向きでも横向きでも表示される

端末を持つ方向が縦なら縦画面、横なら横画面で表示される。これがスマートフォンの特徴のひとつですが、Novel-Press で作る Android アプリは、どちらの方向でも表示できます。

具体的な例を挙げると、画面を縦長に設定している場合、端末を縦方向に持 っていると通常表示されます。そこで端末を横方向にすると、画面に収まるよ うに全体が縮小されて表示されます。画面を横長に設定していて、縦方向に持 った場合も同様です。もし画面を正方形に設定していれば、縦方向でも横方向 でも同じように表示されるわけです。

## ●世界中でダウンロードしてもらえる可能性

従来型携帯は日本国内だけで通用する独自仕様のため、ガラパゴス携帯(ガ ラケー)などと呼称されることがあるのですが、Android はグローバル展開し ている製品です。また Google Play も世界中で利用されています。 つまり Novel-Press で Android アプリを作り、Google Play にアップロード すれば、世界の人々にプレイしてもらえる可能性があるのです。これは大変素 晴らしいことです。もちろんグローバル展開するつもりなら、シナリオを翻訳 する手間があるのですが、逆に言えばそれだけで海外のユーザーにプレイして もらえます。

# 『コラム 1・今後の携帯市場はどうなるか』

2012 年 5 月、docomo の夏モデル発表会が行われましたが、多くの人が驚き ました。iモード対応の従来型携帯が、1 機種も発表されなかったのです。

1999 年の登場以来、i モードで日本の携帯電話市場を牽引してきた docomo ですが、スマートフォンの勢いには抗えず、戦略的に仕方のないことなのでし ょう。今後一切リリースしないということではないと思いますが、au も SoftBank も、従来型携帯よりもスマートフォンにシフトしていることが明確 になっています。

したがって、従来型携帯のユーザーは今や増加を見込めないということなの ですが、これを変わらず使い続けるというユーザーも確実にいます。もしもあ なたがゲーム会社の人ならば、従来型携帯のユーザーは切り捨てるという選択 肢もあるでしょう。しかし本書の読者の多くは、趣味でゲーム制作をしている はずです。商売でやるのではないのですから、少数派になりつつある従来型携 帯のユーザーにも目を配り、楽しませるべきではないかと考えます。

筆者自身の話をしますと、ホラーアドベンチャーゲームの『BAD END』と いう作品をリリースしました。これは有料なのですが、iアプリと S!アプリの 合計ダウンロード数は約 500 になります。無料体験版も合わせれば、ざっと 2500 以上はダウンロードされており、めでたく制作費の回収程度は達成するこ とができました。たいして名前の売れていない個人の作品としては、充分な数 字だと思っています。

22

# 第3章

# 制作準備をしよう

# 『Novel-Press のルールを確認する』

### ●利用規約を確認

公式サイトには利用規約が書かれていますので、重要事項を以下にまとめま す。

- Novel-Press はフリーウェアで、ユーザーはいかなる対価も開発者に支払 う義務はない。ただし開発者もバグ修正などの義務を一切負わない。
- ② 個人/法人、無料/有料配信の別に関わらず、Novel-Press を利用したことを報告する義務はない。
- ③ 作品内容に関する制限はなし。年齢制限のある作品、二次創作など、いかなる内容でも作れる。
- ④ Novel-Press を利用して作られた作品は、それぞれの作者が著作権を持つが、Novel-Press 自体、つまりプログラムの著作権は開発者が持つ。
- ⑤ Novel-Press ソースコードの利用により、利用者がいかなる損害を受けた 場合にも、開発者は一切の責任を負わない。

ゲーム制作ツールは、だいたいどれもこういった規約があります。

「自由に作っていいけれど、プログラムまで自作したとウソをついたらダメ。 何かあっても自己責任」

ということです。特にややこしい規約ということもないかと思います。これ らの規約は予告なしに変更されることがあるということなので、たまにはチェ ックしておきましょう。

なお、Novel-Pressの利用規約とは別に、作品に用いる画像やサウンドについては、フリー素材を用いる場合、各素材サイトの利用規約に違反しないように気をつける必要があります。

### ●使用できるファイルのフォーマット

Novel-Press で使用できるファイルのフォーマットは、プラットフォームご とに異なっています。

プラットフォーム	テキスト	画像	サウンド
docomo(IAP 版)		jpg, gif	mld, mid
au(OAP 版)	++	jpg, png	mid, wav
SoftBank(SAP版)	LXL	jpg, png	spf、mmf
Android(ADR 版)		jpg、 gif、 png	mp3、ogg

#### 【シナリオファイル】

もっとも一般的な txt 形式で、特に説明の必要はないかと思います。

#### 【画像ファイル】

表のとおり、全プラットフォーム共通で jpg が使用できます。主に背景画像 でこの jpg を使いますが、従来型携帯の場合、サイズは横 240×縦 240 の正方 形にする必要があります。Android は機種によってディスプレイサイズが違う のですが、横 800×縦 480 ピクセルだと多くの機種でしっくり表示されます。 縦長表示を前提にするなら、横 480×縦 800 ピクセルでも構いません。

キャラクター画像は docomo の場合 gif、他のプラットフォームでは png に なります。gif や png をキャラクター画像に用いる場合は、背景を透過させな ければなりません。ちなみにウェブサイトで用いる gif ファイルはアニメーシ ョンさせることができますが、Novel-Press では不可能となっています。

#### 【サウンドファイル】

docomo では mld と mid です。mld は docomo 携帯の着メロで使われるフォ ーマットなのですが、自作するのが難しく、フリー素材としてもあまり出回っ ていないので、事実上 mid の一択となるでしょう。mid はパソコンでもよく使 われる MIDI ファイルのことで、こちらはフリー素材がたくさんあります。

au では mid の他に、wav を使用できます。wav は効果音に最適ですが、容 量制限を考えるとあまり多用はできません。 SoftBank では spf、mmf を使用します。spf は最大 4 和音までです。mmf は wav から変換したもののみ使用でき、かつ最大 16 和音までで用意しなけれ ばなりません。docomo の mld と同じく、いずれもフリー素材としてはほとん ど出回っていないので、mid を spf に変換したものを使うのが現実的な方法に なります。

Android では従来型携帯では使えない mp3 と ogg ファイルが使えます。容 量制限に余裕があるので、BGM も効果音も必要なだけ取り入れましょう。

#### ●ファイル名は2桁の番号

Novel-Press で使用するファイルは、ファイル名を作者の自由にできないという特徴があります。2桁の番号で管理されるのです。

たとえばシナリオファイルなら「00.txt」、キャラクター画像なら「11.gif」 といった具合です。シナリオ、背景画像、キャラクター画像、サウンドのいず れも、最大 100 ファイルまで(00 から 99 まで)用意することができます。番 号は連番である必要はなく、途中に空きがあってもかまいません。A というヒ ロインは 00 番台、B というヒロインは 10 番台という風にすれば、管理しやす くなると思います。

なお、背景画像の 00 番はタイトル画面、サウンドの 00 番はタイトル曲に自 動的に割り当てられます。

#### ●拡張子の扱い

各素材ファイルの拡張子は小文字3文字である必要があります。たとえば背 景画像に「00.JPEG」を使おうとしても、表示されません。

パソコンの設定で拡張子を非表示にしていると、大文字になっていることに 気づかないということがたまにあります。もしも表示されないと思ったら、こ こをチェックしてみましょう。

### ●アダルト・パロディ作品を作る場合

同人の世界で人気を博すアダルト・パロディ作品ですが、Novel-Press でも これらを作ることができます。 ただしアダルト作品に関しては、レンタルサーバーの規約でそれを公開する ことを認めているというのが条件になります。アダルト OK のレンタルサーバ ーはそれほど多くなく、ほとんどが有料ですが、どうしてもアダルト作品を作 りたい人は検索して調べてみましょう。

また。第7章で紹介するdマーケットやアプリゲット、Google Playのよう な公式サイトでは、アダルトは不可能となります。しかし Android の場合、 Google Play とは別に、アダルト専門のマーケットがありますので、興味があ るなら調べてみましょう。

## 『制作環境を整える』

### ●ソフトライブラリのベクターを利用する

工場が優れた機械を導入して生産性をアップさせるように、スムーズにゲー ムを制作するためには、高機能のソフトが必要になります。有料のソフトでも いいのですが、できるだけコストを減らすために、フリーソフトを使用するこ とをおすすめします。

インターネットでは、多くのプログラマーが自作のフリーソフトを公開して います。そういったプログラマーのウェブサイトをひとつひとつ訪れて、目当 てのソフトを探すのは大変な手間です。そこで、開発者にソフトを登録しても らって一斉公開しているダウンロードサイトを利用しましょう。

この手のサイトで一番有名なのはベクターで、ゲーム制作に役立つあらゆる ソフトが登録されています。人気順の上位にあるものなら、まず間違いはない でしょう。

#### 【ベクター】

http://www.vector.co.jp/

### ●テキストエディタ

シナリオファイルは txt 形式ですから、これを編集できるテキストエディタ が必要です。Windows にはメモ帳というものが標準で付属していますが、最低 限の機能しかなく、これを使っているクリエイターはまずいません。

ベクターで入手できるものなら、どんなテキストエディタでもよいですが、 ぜひ欲しい機能としては、グローバル検索・変換があります。そのファイルだ けではなく、指定したフォルダ内にあるファイルの語句を一括検索、変換でき る機能です。たとえば主人公の名前を「一郎」から「二郎」に変更したい時、 いちいちすべてのファイルを開かず、グローバルでやってしまえばものの数秒 で済んでしまうわけです。

### ●グラフィック編集・変換アプリケーション

従来型携帯で使用する背景画像のサイズは 240×240 ですが、素材サイトか らダウンロードできる画像は、たいていがパソコンゲーム用の横長サイズとな っています。グラフィック編集ソフトには、サイズ調整の操作をしやすいもの を選びましょう。

そして、キャラクター画像の作成のために、背景の透過した gif や png ファ イルを扱える機能が必須になります。他には、色や質感をワンタッチで加工で きるレタッチ機能が、あればあるほど便利です。

また従来型携帯の場合、画像の容量をどう抑えるかが一番気を遣うところで す。そのため、画質を一括して変換できるようなソフトがあれば、効率よい作 業に繋がるでしょう。

#### ●サウンド再生・変換アプリケーション

Novel-Press で再生できるサウンドファイルのうち、Windows に標準で付属 しているメディアプレイヤーでは mid、mp3、wav を聞くことができます。 Android 向けに制作する場合で、mp3 を使いたくないなら、ogg を聞けるサウ ンドプレイヤーを用意する必要があります。この 4 つのフォーマットは、アマ チュアクリエイター向けの素材サイトを回れば、かなりのファイルが無料で手 に入れられるでしょう。

一番扱いが難しいのは、SAP版で用いる spfです。携帯電話向けのフォーマットなので、これを聞けるサウンドプレイヤーはほとんどありません。その数少ないうちのひとつが、Yamaha SMAF GLOBAL が公開している「ATS-SMAFPhraseL2」です。これは mid ファイルを spf ファイルに変換する機能を備えており、S!アプリ制作に欠かせないものです。[ツール]のページからダウンロードしてください。

#### [Yamaha SMAF GLOBAL]

http://smaf-yamaha.com/jp/index.html

なお、サウンドファイル全般に言えることですが、フォーマットを変換する 際は、素材の配布元がそのことを許可していることを確認してください。中に はそうした変換を一切認めていないサイトもあります。

## ●ロゴ作成ソフト

最初に表示されるタイトル画面には、見栄えのするロゴが欲しいものです。 絵心のない人だとそういうのを作るのは苦労しますが、簡単に作れるソフトが あります。テキストを入力したら、縁の色や質感を指定するだけで、立派なロ ゴができあがるのです。

# 『作業用フォルダを作る』

### ●規定のフォルダ名にする

必要なソフトを揃えたら、作業用フォルダの作成に入ります。これは厳密に ルールが決められているので、間違いのないようにしてください。

まずはデスクトップあたりに、新規のフォルダを作ってください。テスト段 階なら、フォルダ名は IAP 版なら「i アプリ」、OAP 版なら「EZ アプリ」、SAP 版なら「S!アプリ」、ADR 版なら「Android アプリ」、あるいは「Novel-Press」 などです。本格的に作品を作る段階になったら、作品のタイトル名をつけると よいでしょう。

新規のフォルダを作ったら、さらにその中に4つのフォルダを新たに作りま す。フォルダ名は次のようにします。

#### [imageB]

背景画像やイベント CG を入れます。

#### [imageF]

前景画像(キャラクター画像や、それ以外の画像をまとめてこう呼びます) を入れます。

#### [sound]

BGM、効果音のファイルを入れます。

### [txt]

シナリオファイルと、制作者情報などを記載した STAFF ファイルを入れます。

これは全プラットフォーム共通です。後々にアプリを組み立てる際に、この フォルダ名と構成になっていないと、正しく組み立てられません。「imageb」 など、大文字と小文字を間違えてもいけませんので、確実にこのようにしてく ださい。

# 『タグの書き方を覚えよう』

### ●タグ名とパラメータ

次章からシナリオファイルの作成に入りますが、その前に Novel-Press にお けるタグの書き方をマスターしましょう。

画像表示やサウンド再生を指定するもの、それがタグだと第1章で書きました。タグはすべて、次のように[]で囲って書きます。

[タグ名,パラメータ]

タグ名には機能を端的に表した名称がつけられています。サウンドを再生す るタグは[bgmon]という風に。

パラメータは、タグの補助をする役割を持つ値です。画像を表示するタグで あれば、何番のファイルを表示するのか、どの位置に表示するのかといったこ とを、パラメータで指定するのです。パラメータの数はタグによって異なり、 おおむね0個から5個です。

タグ名とパラメータの間は【,】(コンマ)で区切ります。タグは通常、1行 で書くのですが、複数行に分けて書くこともできます。

[タグ名,	
パラメータ]	

一部のタグでは、このように複数行で書いたほうが見やすくなる場合があり ます。また、タグでない文字列、たとえば[あいうえお]などと記述しても、一 切無視されます。これを利用したテクニックがあるのですが、それは後ほど解 説します。

### ●エスケープシーケンス

Novel-Press の文法では、【[】で開始する文字列はすべてタグとして処理されます。それでは【[】という記号自体を表示させたい場合は、どうすればいいでしょうか。

これにはエスケープシーケンスというものを使います。規定された文字の並 びによって、ある文字を表現する手法で、コンピューターの世界ではよく使わ れます。

Novel-Press で【[]を表示したい時は、その直前に【¥】を挿入します。

[エスケープシーケンス]

これではタグとして処理されてしまうので、

¥[エスケープシーケンス]

こうすると画面上では【[エスケープシーケンス]】と表示されます。

もしも【¥】という文字自体を表示させたい時は、その直前に【¥】を挿入します。つまり連続で【¥】を記述するのです。

この品物は¥¥100だ。

これで画面上では【この品物は¥100だ。】と表示されます。

# 第4章

# ノベルゲームを作ろう

# 『テキストを表示する』

## ●プレイヤーの操作を待つ[c]タグ

まずはノベルゲームの基礎であるテキストの表示方法からです。それほど難 しいことはありません。txt フォルダに 00.txt を作成したら、さっそく書いて みましょう。

テキスト表示に関係するタグは複数ありますが、テキストを止めてプレイヤ ーの操作を待つのが[c]タグです。

テキストを停止してプレイヤーの操作を待ちます。[c]テキストを再開しました。また停止します。[c]このように適度に区切ることで、読みやすくしましょう。[c]

これを使わない場合、テキストが垂れ流し状態になるので、特に画面全体に テキストを表示するタイプのゲームでは必須のタグになります。逆に、画面下 に数行しか表示しないタイプだと、[c]タグはなくてもよいかもしれません。

## ●改行をする[1]タグ

打ち込んだテキストは設定された表示区域の端まで行くと、自動的に改行されるのですが、任意の箇所で改行したい場合は[1]タグを使用します。

何もしなくても端まで行くと自動的に改行されます。[c][l]
次に強制的に改行します。[c][l]
改行しました。[c][l]
でも、やたらと[c][l]
改行するのも[c][1]
見にくくなると思います。[c]


[1]タグは単独で使用するのではなく、この例のように[c]タグも一緒に記述して、 プレイヤーの操作を待ってから改行するというのが基本となるでしょう。

次に一行空けます。[c][l]
[1]
ー行空けました。次は二行空けます。[c][l]
[1]
[1]
二行空けました。次は三行空けます。[c][l]
[1]
[1]
[1]
三行空けました。[c]

[1]タグは空白行を作るのにも使えます。複数[1]タグを挿入すれば、その分だけ空白ができるのです。

### ●改ページをする[p]タグ

ページを切り替えたい時は[p]タグを使用します。画面上のテキストが全消去 されて新しいページが始まる、という処理になります。

次に改ページします。[p] 改ページしました。[p]

[p]タグは[c]タグのように、プレイヤーの操作を待ちますが、操作を待たずに改 ページさせたい時は、大文字で[P]と記述してください。

プレイヤーの操作を待たないで改ページします。[P] 改ページしました。[p]

大文字の[P]タグは、テキストを止めてしまうと演出上よくない、たとえばホ ラーゲームで、恐怖のメッセージをとめどなく流す……などのシーンで使える でしょう。

## ●メッセージウィンドウとテキストの表示区域を変更

シナリオ内でメッセージウィンドウやテキストの表示区域を変更したい時が あります。これには[boxpos]タグと[txtpos]タグを使用します。

[boxpos,0,165,240,75] [txtpos,15,170,210,65] メッセージウィンドウの表示区域は X 座標が 0、Y 座標が 165、横幅が 240 ピ クセル、縦幅が 75 ピクセルです。[p] テキストの表示区域は X 座標が 15、Y 座標が 170、横幅が 210 ピクセル、縦 幅が 65 ピクセルです。[p] メッセージウィンドウの表示区域を指定するのが[boxpos]タグです。4 つの パラメータは左から順に X 座標、Y 座標、横幅、縦幅となります。

同様に[txtpos]タグは、テキストの表示区域を指定します。テキストは当然、 メッセージウィンドウの中に収めなければいけませんので、パラメータもその ように指定する必要があります。

これらのタグを用いれば、通常はアドベンチャーゲームらしく画面下部に表示したいが、あるシーンでは全画面で表示するという演出が可能になります。

#### ●禁則処理はできない

行頭や行末にあっては見栄えが悪くなる文字が、前後の行に表示されるよう に調整すること、これを禁則処理といいます。ワープロソフトだと【。】【、】【「】 【」】といった句読点や括弧は、禁則処理が行われます。

Novel-Press ではこの機能がありませんので、テキストのレイアウトは手動 で行う必要があります。

放っておいてもプレイに支障はないのですが、画像を見ればわかるように、 やはり見栄えが悪いですので、調節したほうがよいかと思います。ノベルゲー ムは文章で魅せるジャンルですから、こうしたことにも気を遣いたいものです。



# 『テキストを装飾する』

## ●文字色を変更する[txtcolor]タグ

単にテキストを表示するだけでは、場合によっては物足りなくなります。ち ょっとした工夫でゲームを演出できる、テキストの装飾方法を覚えましょう。 もっとも代表的な方法のひとつは、シナリオ中で文字の色を変更することで す。これには[txtcolor]というタグを使用します。

デフォルトでは文字色は白です。次に色を変えます。[p]

[txtcolor,255,0,0]

文字色を赤にしました。[p]

[txtcolor,0,0,255]

文字色を青にしました。[p]

数字のパラメータが 3 つ出てきましたが、これは RGB 値というもので指定 しています。

コンピューター上の色は、R(赤)、G(緑)、B(青)の組み合わせで表現し ます。3種類がそれぞれ0~255の256段階あるので、256×256×256で約1677 万通りの色があるわけです。

RGB 値はゲーム制作で頻繁に出てくるので、代表的な値は暗記しておきましょう。ネット検索すると RGB 値をすぐに調べられるサイトが出てくるので、 ブックマークしておくと便利です。

RGB 値と代表的な色の対応表					
黒	0,0,0	赤	255,0,0	黄色	255, 255, 0
白	255, 255, 255	緑	0,128,0	紫色	128,0,128
灰色	128,128,128	青	0,0,255	水色	0,255,255
銀色	192,192,192	濃青	0,0,128	赤紫	255,0,255

なお[txtcolor]タグで文字色を変更すると、もう一度変更しない限りずっとそのままです。デフォルトの色は白なので、元に戻したい場合は [txtcolor,255,255,255]とします。

## ●文字の大きさを変更する[txtsize]タグ

文字の大きさを変えるには[txtsize]タグを使用します。

次に文字の大きさを変えます。[p] [txtsize,16] 文字を大きくしました。[p] [txtsize,24] もっと文字を大きくしました。[p] [txtsize,126]

文字の大きさを最大にしました。[p]

このタグにはパラメータが1個だけあり、文字の大きさをピクセル数で指定 します。デフォルトでは12になっていて、最大126まで大きくできます。な お、奇数を指定すると自動的に1引かれた数値になります。 IAP版、OAP版、SAP版は12~30、ADR版は30程度が推奨値です。あま り多用するタグではありませんが、キャラクターが大声を出す時の演出など、 ここぞという時に使うと効果的です。[txtcolor]タグと同様に変更したらしっぱ なしなので、必要がなくなったら元に戻す記述をしてください。

## ●文字の表示速度を変更する[txtspeed]タグ

文字の表示速度はアプリの組み立て時に設定しますが、シナリオ中に変えた い場合は[txtspeed]タグを使用します。

次に文字の表示速度を変えます。[p]

[txtspeed,10]

文字の表示速度を最速にしました。[p]

[txtspeed,0]

文字の表示速度をもっとも遅くしました。[p]

これもパラメータは1個で、0~10の範囲で速度を指定します。数字が大き いほど速く、10が瞬間表示になります。主人公が殺されてゲームオーバーとい ったシーンでは、速度をゆっくりにすると、雰囲気が出るでしょう。

必要がなくなったら元に戻すタグを記述しなければならないのは、[txtsize] や[txtspeed]タグと変わりません。

## 『背景画像を表示する』

### ●背景画像を表示する[bi]タグ

背景画像は IAP 版、OAP 版、SAP 版では 240×240 のサイズで、ADR 版で は自由です。ファイル名が 00 の背景画像は、自動的にタイトル画面に割り当 てられますが、ADR 版ではこのタイトル画面のサイズが、そのままゲーム画面 のサイズになります。

ではシナリオ本編で表示するために、名前が「01」のファイルを用意して、 次の記述をしてみます。

[bi,01]

背景画像を表示しました。[p]



背景画像が瞬間表示されます。[bi]タグにはパラメータが1個あり、imageB フォルダ内の00~99 までのファイルを指定します。タイトル画面に割り当て られる00も、このタグで表示させることが可能です。

ちなみに、背景画像に限らないのですが、拡張子を含めて記述する必要はあ りません。

### ●背景を一色で塗りつぶす

[bi]タグの応用として、画面全体を一色に塗りつぶすことができます。

[bi,m,255,0,0]

背景を赤く塗りつぶしました。[p]

[bi,m,255,255,255]

背景を白く塗りつぶしました。[p]

固定パラメータ1個と通常パラメータ3個からなる[bi,m]タグです。2~4つ めのパラメータには RGB の値を指定します。

通常の背景表示と同じく、瞬間表示されます。ライトを向けられた時は白く、 ナイフで刺されて血が出た時は赤く……という感じに、シンプルながら効果的 な演出が期待できるでしょう。また、時間の経過や場所の移動などでブラック アウトさせる場合、このタグが頼りになります。わざわざ真っ黒な画像ファイ ルを使わなくてよく、容量を節約できるのが利点です。

44

## 『キャラクター画像を表示する』

### ●キャラクター画像を表示する[fi]タグ

背景以上に欠かせない画像が、キャラクター画像です。背景画像と比べると、 少しだけ覚えることが多くなります。

キャラクター画像を表示させるには、[fi]タグを使用します。

[fi,00,1]

キャラクター画像 00 を 1 の場所に表示しました。[p]

パラメータは 2 個あり、1 つめはファイル番号です。2 つめは画像を表示す る場所を 1~10 の範囲で指定します。

Novel-Press ではキャラクター画像を表示できる場所が 10 個用意されてお り、たとえば左に表示するのは 1、右に表示するのは 2 という風に自由に決め ていいのです。数字が大きいほど前面に表示されます。



### ●位置を指定する[fixy]タグ

ただ表示するだけでは、もちろん充分ではありません。[fixy]タグで座標を指 定して、好きな位置に表示できるようにしましょう。

[fi,01,2] [fixy,2,30,0]

2の場所に表示しているキャラクター画像 01の座標を、X=30、Y=0 と指定し ました。[p]

パラメータは3個あり、順番にキャラクターの表示場所、X座標、Y座標で す。キャラクター画像を表示する時は、基本的にこのタグと[fi]タグをセットで 使用することになります。

これらのタグで表示できるのはキャラクター画像に限りません。第3章で書 いたように、キャラクターとそれ以外の画像をまとめて前景画像と呼びます。 カットインやアイテム画像など、自由に表示させてみましょう。

### ●消去する[fioff]タグ

キャラクター画像はいつでも好きな時に消せなければいけません。これには [fioff]タグを使用します。

[fioff,1]

1の場所に表示していたキャラクター画像を消去しました。[p]

パラメータは1個で、表示している場所を1~10の範囲で指定します。ファ イル名を指定するのではないことに注意してください。 [fioff] すべての場所のキャラクター画像を消去しました。[p]

このようにパラメータを省略すると、すべての場所のキャラクター画像が同時に消去されます。

# 『描画バッファリングを使う』

## ●[lock]タグと[unlock]タグ

これまでに解説してきたタグでは、画像を瞬間表示することしかできません。 何かしらの効果をつけて表示したいものですが、これも簡単なタグで行うこと ができます。まずは次の例をご覧ください。

時間と効果を指定して背景画像を切り替えます。[p]

[lock]

[bi,01]

[unlock,1000,f]

背景画像を1秒のフェードで表示しました。[p]

[lock]、そして[unlock]というタグが出てきました。この2つの間に、[bi]や[fi]、 [fixy]、[fioff]といった画像に関わるタグを挟むことで、いろいろな効果をつけ ながら表示、消去することができます。この機能を描画バッファリングといい ます。

- ① [lock]タグで描画バッファリングの開始を宣言する
- ② 画像関係のタグを記述する(複数記述可能)
- ③ 最後に[unlock]タグを記述する

流れは以上のようになります。[unlock]タグの1つめのパラメータは、画像 切り替えの時間です。ミリ秒という単位で指定し、1000なら1秒となります。 2つめのパラメータは効果の種類で、fとはフェードの意味です。 効果オプションをまとめると、次の表のようになります。タイプと方向の組 み合わせです。

効果オプション一覧				
f	効果タイプ	フェード		
b		ブラインド		
s		スクロール		
р		ページめくり		
q		クアドラングル		
u	効果方向	上方向へ		
r		右方向へ		
d		下方向へ		
1		左方向へ		
t	画面切り替え中にテキストを描画する			

たとえば「br」とすると、右方向へのブラインド効果です。同様に「su」な ら上方向へのスクロールです。方向を指定しなくてもよい効果があり、「f」の フェードと、「q」のクアドラングルです。クアドラングルとは、画面を 20×20 に分割して、モザイク状にランダムに切り替わるというものです。これは処理 がやや重めで、機種によってはカクカクするかもしれません。

t はテキストを表示したまま画像を切り替えたい時に使用します。ただしこのオプションは OAP 版、SAP 版では使用できません。

このテキストを表示したまま背景画像を切り替えます。[c][l] [lock] [bi,02] [unlock,1000,ft] 切り替えました。[p]



ブラインド

スクロール



ページめくり

クアドラングル

この画像で見るとスクロールとページめくりは似ていますが、前者は画像が 画面外から出現するのに対し、後者はまさにページをめくったように裏側から 出現します。

実際にどう使うかですが、場所が変わる時はフェードインかブラインド、またはページめくりがよいでしょう。スクロールは二分割したイベント CG の表示に威力を発揮します。たとえば最初は転んだ主人公に手が差し伸べられていて、上にスクロールすると、女の子の笑顔が……という風に。

キャラクターをクアドラングルで表示&消去すると、なんだか SF チックに なったりします。クリエイターの発想次第でいくらでも効果的な演出ができま すので、いろいろ試してみましょう。

### ●連続で切り替えたい時

描画バッファリングを連続して使いたい場合は、[lock]タグと[unlock]タグの セットを、必要な分だけ記述します。

背景画像 03、04、05 を連続で切り替えます。[p]
[lock]
[bi,03]
[unlock,1000,f]
[lock]
[bi,04]
[unlock,1000,br]
[lock]
[bi,05]
[fi,00,1]
[fixy,1,30,0]
[unlock,1000,f]
最後にはキャラクター画像も一緒に表示しました。[p]

[lock]タグと[unlock]タグの間には、画像タグを好きなだけ挟んで構いませんが、 画像以外のタグは挟んではいけないことに注意してください。

# 『メッセージウィンドウを変更する』

### ●メッセージウィンドウの色を変更する

メッセージウィンドウを工夫したい……そんな希望もタグひとつで叶えるこ とができます。デフォルトでもいいですが、ぜひ作品に合ったメッセージウィ ンドウにしてみましょう。

メッセージウィンドウの色は自由に変えることができ、[boxcolor]タグを使用 します。

メッセージウィンドウの色を変更します。[p]

[boxcolor,0,0,255]

青色に変更しました。[p]

[boxcolor,0,128,0]

緑色に変更しました。[p]

3つのパラメータは、おなじみの RGB 値です。キャラクターの視点が変わる ようなシナリオでは、メッセージウィンドウの色を変えると簡単に見分けが付 くようになるでしょう。

ほとんどのノベルゲームでは、メッセージウィンドウは暗色系となっていま す。それ以外はあまり目に優しくないというか、ちょっと見づらくなります。 個性を出したいからといって、奇抜な色にするのはやめたほうが賢明でしょう。

### ●メッセージウィンドウの透明度を変更する

通常は省略してよいのですが、[boxcolor]には 4 つめのパラメータがあります。

メッセージウィンドウの透明度を変更します。[p]

[boxcolor,0,0,0,10]

透明度を変更しました。[p]

透明度は設定画面と同じく、0から10の間で指定できます。10にすると完 全に透明です。

### ●画像をメッセージウィンドウに使う

通常は半透明の枠でしかないメッセージウィンドウを、画像にする方法もあ ります。

[fi,99,10]

[fixy,10,0,165]

メッセージウィンドウに画像を使用しています。[p]

imageFフォルダに置いた 99 というファイルを使用します。ファイル名はどの数字でもよいのですが、99 だとわかりやすくなります。

大事なのは、ウィンドウ画像の表示場所を 10 とすることです。こうすれば 他のどのキャラクター画像も、ウィンドウ画像の後ろに表示されるようになり ます。

従来のメッセージウィンドウが消えるわけではありませんので、アプリの組 み立て時にはメッセージウィンドウの透明度を 10 として、さらにプレイヤー が透明度を調節できない設定にする必要があります。 なお、IAP 版、OAP 版、SAP 版では半透明の画像を使えないので、ウィンドウ画像も当然ながら半透明のものを表示させることはできません。



# 『画像に特殊効果をかける』

## ●3 種類の特殊効果を加える[effect]タグ

もっと画像を工夫できないものか? そう考える人にうってつけの、画像に 特殊効果をかける[effect]タグがあります。これを使えば、いっそう画像演出が 豊かになることでしょう。

画像に特殊効果をかけます。[p]

[effect,mono]

色調をモノクロにしました。[p]

パラメータには次の表にある効果を指定します。

	特殊効果一覧
mono	色調をモノクロにする
sepia	色調をセピアにする
crev	色調を反転する
none	特殊効果を解除する

mono は灰色で、殺伐としたシーンや憂鬱なシーンなどに使えます。

sepia は昔の写真のような古びた感じの色合いで、回想シーンなどに役立つ でしょう。

crevの色調反転とは、文章ではちょっと説明がしにくいので、次の画像をご 覧ください。



白い部分が黒い部分になるなど、画像の印象がガラッと変わります。アニメ などでも、強いショックを受けるシーンではこういった色調反転が使われるこ とがあります。

特殊効果の必要がなくなったら、noneと指定して、通常の画像に戻しましょう。なお、特殊効果は画面全体にかかりますので、キャラクター画像などもまとめて変更されます。したがって、画像をメッセージウィンドウに使用している場合は、それにも特殊効果がかかってしまうので注意しましょう。

## ●描画バッファリングとの組み合わせ

[effect]タグは画像系のタグなので、描画バッファリングによって効果をつけ ながら表示させることができます。

[lock] [effect,sepia] [unlock,1000,f]

こうすると、フェードをかけながらセピア調にできます。

# 『サウンドを再生する』

## ●再生する[bgmon]タグ

Novel-Press においては、BGM も効果音も[bgmon]というタグで処理するこ とになります。mid などのサウンドファイルを sound フォルダに入れたら、次 の記述例を試してください。

これから BGM を再生します。

[bgmon,01]

BGM を再生しました。

パラメータにファイル名を指定します。00 はタイトル画面専用の曲なので、 01 以上を指定しましょう。サウンドはループ再生されますが、一回限り再生す る効果音は次のようにします。

これから効果音を再生します。

[BGMON,41]

効果音を再生しました。

大文字で[BGMON]とすると、一度だけの再生になります。BGM は[bgmon] で、効果音は[BGMON]と覚えてください。サウンドは IAP 版と ADR 版が 5 つまで、OAP 版と SAP 版が 4 つまで同時再生ができますが、機種によっては それ以下となる場合もあります。

## ●停止する[bgmoff]タグ

サウンドを停止するには、[bgmoff]タグを使用します。

[bgmon,01]

再生している BGM を停止します。

[bgmoff,01,1000]

BGM をフェードアウトで停止しました。

1 つめのパラメータに停止したいファイルの番号を指定します。2 つめのパ ラメータはフェードアウトにかける時間です。これを省略すると、即時停止と なります。即時停止だとブツ切り感が強いので、基本的にはフェードアウトさ せるのが好ましいです。時間は1秒から2秒程度が最適でしょう。

また、1 つめのパラメータに「-1」と指定すると、すべてのサウンドが停止 します。

[bgmon,01] [BGMON,41]

再生している BGM と効果音を停止します。[p]

[bgmoff,-1]

すべてのサウンドを停止しました。

#### ●SAP 版でのサウンド再生

すでに述べたように、SAP版でのサウンドは、spf形式のファイルを使用す るのが現実的です(wavファイルから変換したもののみ使用できる mmf はサ イズが重く、使いづらいです)。

spf ファイルは最大で 4 和音 (4 つの音色で構成されている) しか使えないの で、音数が多いサウンドは「分解して同時発音させる」というテクニックが必 要になります。

まず、mid ファイルを spf 形式に変換するために「ATS-SMAFPhraseL2」 を起動します。

🔡 S I	AAF/F	hra	ase L2	Auth	oring 7	ool C:¥iDKDoJa5.1¥apps¥b	adend¥res¥so	und¥09.mid - [Score Window] 📃 🗖 🔀
🗖 Ei	ile <u>E</u> d	lit	⊻iew	Window	Optio	n <u>H</u> elp		_ 8 ×
K		•	Rp	00.0	0.000	TB 1.250		
AT N	vlode	L2(N	1A-5)	- Con	tents Mo	ode FM16 💌		
Сору	Status		Edit F	🗸 Save	🗖 Tra	ans. Edit Status 🔽 Edit 🔽 Sa	ive 🔽 Trans.	
P1RS	(B):120	)		P2RS[B	]:72	P3RS[B]:0 P4	RS[B]:0	
<b> </b> 😂	🛎 💲		0		C I	7 歩 3 5 6 6		
AD[E	3/s]:116	;	MD[	3/s]:268		1DP[ms]:0		
Pian	o Roll	] [	Event Li	st				
Ch	Mute	С	BM	BL	Pch	Name	Cv O	
1			124	1	8	Celesta	1	
2			124	1	40	Violin	1	
3			124	1	48	Strings1	1	
4			124	1	42	Cello	1	
5			124	1	52	ChoirAah	2	
6								
7								
8								
9								
10			125	0	2	Drum User1	2	
11								
12								
13								
14								
15								
16				I	L			
Ready						TL[ms]:45600	SZ[B]:	Copyright(C) 2003-2004 YAMAHA CORPORATION

メニューの「File」→「Import from SMF」で mid ファイルを読み込むと、 そのファイルに使われている音色が表示されます。「Cv」という欄に 1 と 2 が ありますが、これは spf 形式に変換した時のグループになるのです。

この状態で「File」→「Save」を選んでみましょう。グループの数だけ spf ファイルが保存されます。

そしてこれらの spf ファイルを「sound」フォルダに入れて、同時に再生す るようにタグを記述します。

[bgmon,01][bgmon,02]

spfファイルを再生しました。[p]

以上が SAP 版でのサウンド再生の流れになります。他のプラットフォームと 比べると、ややファイルの扱いが煩雑です。しかも複数ファイルに分かれてい る楽曲は、BGM リストで再生させることはできなくなるので(BGM リストで は同時再生不可能)、注意が必要です。

# 『ウェイト機能を使う』

### ●指定した秒数だけ進行停止する

ただテキストを流すだけでは、メリハリがつかないことがあります。 適切な ウェイトをかけることによって、上手に演出してみましょう。

テキストを停止します。[c][1]

[wait,1000]

1秒間テキストを停止しました。[p]

テキストを一定時間停止する機能を持つ[wait]タグです。パラメータにはミ リ秒単位で数値を指定します。ウェイトしている間にセレクトキーを押すなど、 読み進みの操作をしたら、ウェイトは解除されます。

プレイヤーの操作に関係なく指定した時間ウェイトさせたい場合は、次のよ うにします。

テキストを停止します。強制的なウェイトです。[c][l]

[WAIT,1000]

1秒間テキストを停止しました。[p]

大文字で[WAIT]とすれば、指定した時間が経過する以外では先に進まないようになります。シーンによって適切に使い分けましょう。

## ●背景切り替え時にウェイトする

実際にどのようなシーンで[wait]タグを使用すればよいでしょうか。まずは 画面切り替わりの間に挟むということが考えられます。

背景画像を連続で切り替えます。[p]

 [lock]

 [bi,01]

 [unlock,1000,f]

 [wait,500]

 [lock]

 [bi,02]

 [unlock,1000,f]

 [wait,500]

 [lock]

 [bi,03]

 [unlock,1000,f]

 [wait,500]

 [lock]

 [bi,03]

 [unlock,1000,f]

描画バッファリングによる背景切り替えですが、普通に切り替えたのでは味 気ないことがあります。そこでウェイトを挿入して、ゆったり感を持たせてい ます。

この例では 0.5 秒のウェイトですが、日時が変わるなど、長い時間変化のシ ーンでは、思い切って数秒ウェイトさせてみましょう。時間が経過したことが プレイヤーにも強く印象づけられます。

この際、強制的にウェイトするか飛ばせるようにするかは悩みどころですが、 あまり強制が多すぎるのも考えものです。

## ●サウンド再生時にウェイトする

Novel-Press のサウンド再生は鳴り終わるまで待つことなく進行します。待たせたい場合は次のようにします。

効果音を再生します。[c][l]

[BGMON,41]

[WAIT,1000]

再生している間はテキストを停止しました。[p]

効果音の再生時間を1秒と仮定して、大文字の[WAIT]タグに1秒と指定する ことで、効果音が鳴り終わるまで強制ウェイトをしています。ただ再生するの と比べてかなり雰囲気が違いますので、状況に応じて使い分けましょう。

## 『画面を揺らす』

### ●衝撃を演出する画面揺らし

何かにぶつかった、殴った、殴られた……そんなシーンでは画面を揺らす演 出をセットにすると、より雰囲気が出ます。

これから画面を揺らします。[p]

[shake, 10, 1000]

画面を揺らしました。[p]

画面全体を揺らす[shake]タグです。1 つめのパラメータにはピクセル単位で 揺らす強さを指定します。この例では 10 ピクセルです。そして揺らす時間を ミリ秒で指定します。

この画面揺らしは、あまり強くしたり頻繁に使ったりすると、プレイヤーは 気分が悪くなってしまいます。程度と頻度をよく考えて使いましょう。

ちなみにこのタグは、画面が揺れ終わるまで自動的に待ちます。そして読み 進める操作でスキップすることはできません。つまり効果音のようにウェイト を併用する必要はないのです。

### ●効果音と一緒に揺らす

先ほど例に挙げた「殴った」「殴られた」というシーンを想定して、タグを書 いてみましょう。効果音を同時に鳴らすのです。

大事なのは、[BGMON]タグを[shake]タグよりも先に記述することです。そ うしないと揺れ終わってから再生されることになります。 [BGMON,41] [shake,20,1000] 俺は友人を殴った。[p] [BGMON,41] [shake,30,1000] するとそれよりも強く殴られた。[p]

揺れの強さはいつも同じにするのではなく、この例のように微妙に調整する ことで、シーンにあった演出が実現できます。

# 『画面をフラッシュさせる』

## ●一定時間だけ背景を塗りつぶす

背景画像を一色に塗りつぶすタグに[bi,m]がありましたが、一定時間だけフ ラッシュのような効果を実現したい場合はどうでしょうか。

[bi,01]

¥[bi,m]タグを用いて画面をフラッシュさせます。[p]

[bi,m,255,0,0]

[WAIT,1000]

[bi,01]

画面を赤くフラッシュさせました。[p]

このように、ウェイトと元の背景画像に戻す記述が必要になり、少々煩雑で す。そこで指定した時間だけ塗りつぶす[flash]タグを使用します。

[bi,01]

¥[bi,m]タグを用いず画面をフラッシュさせます。[p]

[flash,1000,255,0,0]

画面を赤くフラッシュさせました。[p]

かなり記述がシンプルになりました。パラメータは時間指定とRGB 値です。 フェード効果などはなく、瞬間的に画面が塗りつぶされ、指定した時間が経て ば元の背景に戻ります。

なお、[bi,m]タグを使った場合は塗りつぶす対象が背景なので、メッセージ ウィンドウはそのまま表示されるのですが、[flash]タグは画面全体が対象にな ります。つまりこのタグで塗りつぶしている間は、メッセージウィンドウが非 表示になるのです。そのあたりも踏まえて、使い分けるとよいでしょう。

#### ●画面を点滅させる

フラッシュとウェイトを同時に使うことで、点滅のような効果を出すことが できます。

[flash,1000,255,255,255] [WAIT,100] [flash,1000,255,255,255] [WAIT,100] [flash,1000,255,255,255] [WAIT,100]

画面を白く点滅させました。[p]

フラッシュしたあとに元の背景に戻り、わずかな時間ウェイトさせ、またフ ラッシュ……という流れです。立て続けに何かが光っている、などのシーンで 使えるでしょう。

# 『バイブレーションさせる』

#### ●携帯電話ならではの振動演出

パソコンゲームでは不可能な、モバイルゲームならではの機能にバイブレー ションがあります。端末そのものを振動させて、ショックを演出したりできま す。

バイブレーションを開始します。[p] [vibon] バイブレーションしています。次に終了させます。[p] [viboff] バイブレーションを終了しました。[p]

バイブレーションを開始するのが[vibon]タグ、終了するのが[viboff]タグです。 両方ともパラメータはありません。

注意点は、バイブレーションを開始すると、終了を指定しない限りずっと振動し続けることです。このため、[viboff]タグの書き忘れはかなり致命的なバグになります。

なお、設定画面でバイブレーションをオフにしていると、当然ながらこのタ グは機能しなくなります。

「どうしてもこの振動演出はオフにさせたくない」

という場合は、プレイヤーにバイブレーション機能をいじれないように初期 設定してください。

## ●小刻みに振動させる

開始するとずっと振動し続ける[vibon]タグは、やはりウェイトを併用することで効果的に使用できます。

[vibon]
[WAIT,500]
[viboff]
[WAIT,100]
[vibon]
[WAIT,500]
[viboff]
[WAIT,100]
[vibon]
[WAIT,500]
[viboff]
小刻みに振動させました。[p]

0.5 秒振動させて、0.1 秒の間隔を挟み、また 0.5 秒振動させる……という流 れになります。

# 『選択肢とジャンプを使う』

### ●選択肢でシナリオを分岐する

ノベル・アドベンチャーゲームは選択肢がなければ、挿絵とサウンドがつい た電子小説に留まってしまいます。適切に選択肢を用いることでゲーム性を加 えましょう。

選択肢を使用するには、行き先となる着地点を設定する必要があります。こ れをラベルといいます。

 $[@ \dot{\sim} - \checkmark A]$ 

ラベルを設定するには[@~]と記述します。ラベルの名前には【[】【]】【,】以 外なら何でもつけることができます。例のように[@シーン〇〇]とするのが、一 番わかりやすいでしょう。

これに選択肢の記述を加えてみます。

```
[select,
A:西へ,@シーン 02,
B:東へ,@シーン 03]
[@シーン 02]
西へ向かった。[p]
[@シーン 03]
東へ向かった。[p]
```



西へ向かうとシーン 02 に、東へ向かうとシーン 03 に飛びます。選択肢表示 の[select]タグは、[select,選択肢の文字列,ラベル名]という構成で、選択肢の数 だけセットで記述します。しかし一行にすると長くなって見た目がわかりづら いので、このように改行しています。

選択肢はこれで完成ではありません。シーン 02 に飛ぶと、すぐ次にシーン 03 があるため、何もしないとそれも表示されてしまうのです。選択肢のあとは 「共通のシナリオにジャンプする」という記述が必要になります。

[select, A:西へ,@シーン 02, B:東へ,@シーン 03]
[@シーン 02]
西へ向かった。[p]
[jump,@シーン 04]

$[@ \not\sim -  earrow 03]$
東へ向かった。[p]
$[i_1, \dots, n_{n-1}] \in [0, 1]$
[Jump,@/~/04]
$[@ \not\sim - \checkmark 04]$
今日もいい天気だ。[p]

シーン移動の[jump]タグで、同じようにラベル名を指定します。選択肢でどれを選んでも最終的にはシーン 04 に進むことになるわけです。

ちなみに[select]タグにせよ[jump]タグにせよ、存在しないラベル名を指定していたら、それは無視されて直後のシナリオに進んでしまいますので、記述ミスがあればすぐにわかります。

### ●タイトル画面に戻る

選択肢によってはバッドエンドになるというシナリオを書く場合、タイトル 画面に戻す記述が必要です。これには[jump]ではなく別のタグを用います。

どれかひとつの道だけが正解だ。[p] [select, A: 左へ,@シーン 02, B: 正面へ,@シーン 03, C: 右へ,@シーン 04] [@シーン 02] 無事に通過した。[p] [jump,@シーン 05]
[@シーン 03] 正面は失敗の道だった。バッドエンド……。[p]
[z]
[@シーン 04] 右は失敗の道だった。バッドエンド……。[p]
[z]
[@シーン 05] さあ、探索を続けよう。[p]

正面と右を選んだ先には[z]タグを記述しています。これだけでタイトル画面 に戻すことができるのです。何のエフェクトもなく戻るので、バッドエンドの 余韻を持たせたいなら、直前にフェードなどで暗転させるとよいでしょう。

## ●時間制限つきの選択肢

何も選ばなくてもやがてジャンプの処理が開始する「時間制限つきの選択肢」 を作ることもできます。

リンゴを取るかバナナを取るか。側でサルが狙っているぞ。[p]

[select,

A:リンゴを取る,@シーン 02, B:バナナを取る,@シーン 03, timeout,5000,@時間切れ]

[@シーン 02] リンゴを取った。[p]
[z]
[@シーン 03] バナナを取った。[p]
[z]
[@時間切れ] 悩んでいるうちにどちらもサルに取られてしまった。[p]
[z]



通常、選択肢が表示される画面ではシナリオが停止するのですが、[select] タグの最後に[timeout,ミリ秒,@ラベル名]を加えると、その選択肢に時間制限 が設けられます。 ー番上の選択肢部分にメーターが表示され、この例では5秒間何もしないで いると、時間切れというラベルにジャンプします。主人公にピンチが迫ってい る……というシーンでは、この機能がとても有効です。

なお、時間制限つきの選択肢が表示されている状態でもセーブすることがで き、ロードすると再びメーターが最初から減少していきます。しかしこれでは せっかくの時間制限演出が台無しで緊迫感が薄れてしまうので、後述するセー ブ機能オフを併用すると効果的です。

## 『フラグ機能と条件分岐を使う』

## ●条件によってシナリオに変化をつける

特定のイベントを起こすために必要なプログラム処理を、ゲームの世界では 「フラグを立てる」といいます。恋愛ゲームでは目当ての女性キャラを攻略す るため、デートを重ねたりするわけですが、正しい選択肢を選んでいってデー トを成功させたならフラグを立てる……という流れです。

[flagon,1]

彼女とのデートを成功させた。[p]

フラグを立てる[flagon]タグです。パラメータには1から256までの数字を 指定します。この数字がフラグ番号で、1から127まではファイルフラグ、128 から256まではシステムフラグと呼ばれます。

ゲームをクリアした。[p] [flagon,128] [z]

ファイルフラグはセーブデータごとに保存されるもので、最初の例のデート 成功など、その回のゲームの攻略に必要なフラグを管理します。対してシステ ムフラグは、作品そのものに保存されるフラグです。グッドエンドに辿り着い たら、新しいルートのシナリオを遊べるようになる……といった用途に使いま す。 この使い分けを間違えると、深刻なバグの原因になります。たとえばデート 成功をシステムフラグとして管理してしまうと、次回以降のプレイで失敗の選 択肢を選んでしまったとしても、デートは成功したと判定されてしまいます。 逆も同様で、グッドエンドを迎えたことをファイルフラグとして管理すると、 アプリを終了した瞬間に元通りになってしまいます。

### ●フラグによって条件分岐する

フラグは立てただけでは、何の役にも立ちません。 「もしこのフラグが立っていれば、このラベルにジャンプする」 という条件分岐処理を行いましょう。

どちらの宝箱を開けようか?[p]

[select,

A: 左の宝箱,@シーン 02,

B:右の宝箱,@シーン 03]

[@シーン 02] お宝を見つけた。[p]

[flagon,1]

[jump,  $\dot{\nu} - \nu 04$ ]

[@シーン 03] 中身は空っぽだった。[p]

[jump,  $\dot{\nu} - \nu 04$ ]

[@シーン 04] 村に帰ってきた。[p]
[if,1,@グッドエンド]
宝が見つけらなかったので、村人たちは残念そうだ。[p]
[z]
[@グッドエンド]
宝が見つかったので、村人たちは大喜びだ。[p]
[z]

シーン 04 にて、条件分岐の[if]タグを記述しています。フラグ番号、ジャン プ先のラベル名を順に記述するだけです。

(省略)

[if,1,2,3,4,@グッドエンド]

宝が見つけらなかったので、村人たちは残念そうだ。[p]

[z]

[@グッドエンド] 宝が見つかったので、村人たちは大喜びだ。[p]

[z]

このように、フラグ番号は1つだけでなくいくつも指定することが可能です。 指定したフラグがすべて ON の場合のみ、ラベルにジャンプします。

難易度を高めようとするほど、フラグの数は多くなりますので、どのイベントをどの番号で管理しているかは、頭の中で覚えるよりもメモを作っておくの が得策です。

## ●フラグを消去する

必要がなくなったフラグは、任意に OFF にすることができます。

[flagoff,1]
[flagoff,2]
[flagoff,3]
[flagoff,11]
[flagoff,21]
フラグ 1、2、3、11、21 を OFF にしました。[p]

立てる時と同じ要領で[flagoff]タグにフラグ番号を指定します。キャラクターの機嫌を損ねて攻略不可になった……などのシーンで使いましょう。

また、作品内で使用しているシステムフラグをすべて OFF にすれば、初期 化という処理が実現できます。もう一度インストール時の状態から遊びたいと いうプレイヤーの要求に応えられるのです。

# 『プレイヤーに文字列を入力させる』

## ●キャラクターの名前をつける

恋愛ゲームでは、プレイヤーの感情移入を強めるために主人公の名前を自分 で決められることがしばしばあります。Novel-Press でもこういった機能をサ ポートしています。

主人公の名前を入力してください。[p]

[input,1]

主人公の名前は[output,1]です。[p]



まず、[input]タグは文字入力画面を表示します。保存できる文字数は全角で も半角でも 10 文字までとなっています。それより長い文字を入力すると、11 文字目から先は自動で切り捨てられます。パラメータには保存するスロットを 1から 20 の数字で指定します。

そして保存した文字列を表示するには[output]タグを使用します。[input]タ グで指定したスロットを同じように指定すると、画面上に表示されます。

#### ●デフォルトの文字列を入力しておく

あらかじめ主人公のデフォルト名が用意されていて、プレイヤーは任意に名 前を変更できるというのも恋愛ゲームのスタンダードです。

[input,1,山田太郎]

主人公の名前を入力してください。[p]

[input,1]

主人公の名前は[output,1]です。[p]

最初の[input]タグの2つめのパラメータに「山田太郎」と指定しています。 2つめのパラメータを指定している場合、文字入力画面は表示されずに、その ままスロットに格納されます。これでデフォルト名を設定するのです。

次にあらためて[input]タグを記述し、今度は文字入力画面が出るようにする のですが、すでに「山田太郎」が入力されています。このままでいいか、それ とも変更するかをプレイヤーは選ぶことができます。

## 『セーブ機能に制限をつける』

### ●難易度の調整に役立つセーブ不可

ノベル・アドベンチャーゲームはいつでもセーブ可能というのが当たり前で す。しかし、あえてこの機能に制限をつけてみるとどうでしょうか。たとえば ミステリーやホラーといったジャンルでは、失敗できないという心理をプレイ ヤーに植えつけ、さらなる緊張感の増加をもたらすことができます。

[saveoff]

現在セーブできません。再びセーブできるようにします。[p]

[saveon]

セーブできるようになりました。[p]

[SAVE]	[LOAD]
File.01	
File.02	
File.03	
File.04	
File.05	
File.06	
File.07	
File.08	
File.09	
File.10	

対になるタグ[saveon]と[saveoff]です。パラメータはありません。[saveoff] と記述すると、栞画面を呼び出してもセーブを選ぶことができなくなります。 不可能になるのはセーブだけで、ロードは可能です。そして[saveon]と記述す ると、再びセーブ可能になります。

## ●フラグと併用する

単にセーブ不可能とするだけでなく、フラグと併用してみましょう。

セーブ機能を OFF にしますか? [p] [select, A: OFF にしない,@セーブ機能 ON, B: OFF にする,@セーブ機能 OFF] [@セーブ機能 ON] セーブ機能は ON のままです。[p] [jump,@オープニング] [@セーブ機能 OFF] [saveoff] [flagon,1] セーブ機能を OFF にしました。[p] [jump,@オープニング]

ゲーム開始直後に、選択肢でセーブ機能を OFF にするかしないかをプレイ ヤーに選ばせます。OFF にするとフラグが立って、以後はセーブができなくな ります。

これにより、セーブ不可能状態でグッドエンドに辿り着くと、プレイ評価が 変わる……といったシステム構築ができるのです。もちろん長編作品でセーブ 不可能というのは厳しいですが、推理力を試す短編ミステリーなどでは充分に 使えるでしょう。

# 『ブラウザを起動させる』

### ●直接ウェブサイトに接続する

パソコンゲームでは、システムメニューから URL クリックで公式サイトに 接続できるゲームがよく見られます。

Novel-Press でも基本機能として、タイトル画面から指定のウェブサイトへの接続ができるのですが、シナリオ内から接続させることもできるのです。これにはパラメータに URL を指定した[url]タグを使用します。

Google のサイトに接続します。[p]	
[url,http://m.google.co.jp]	
接続しませんでした。[p]	
URL: http://m.google.co.jp サイト接続しますか? 【はい】 しいえ	

これはエミュレーターによる画像なので、実際は機種によって表示形式が違ってきますが、[url]タグを通過すると確認が入ります。「はい」を選べばサイトに接続し、「いいえ」を選べばシナリオが継続します。

また、機種によってブラウザ起動と同時にアプリが終了する場合と、終了しない場合があります。これは Novel-Press では制御できません。

#### ●どんなケースで使用するか

ゲームからウェブサイトへの接続というのは、そう頻度の高いものではあり ませんし、クリエイターの公式サイトに接続する場合もタイトル画面からさせ ればよいので、[url]タグは使用する機会がかなり少ないことになります。

ではどんな活用方法があるかというと、その作品が有料ゲームの体験版で、 興味を持った人に有料版の案内をしたいというケースが挙げられます。

有料版を購入しますか? dマーケットに接続します。[p]

[select,

A:有料版を購入する,@有料版購入,

B:タイトル画面に戻る,@タイトル画面へ]

[@有料版購入]

[url,http://appli.docomomarket.ne.jp/]

[@タイトル画面へ]

#### [z]

有料版を購入するかタイトル画面に戻るかを選ばせます。これで有料版購入 までの手間が、だいぶ省けるでしょう。

# 『各種リストに追加する』

### ●CG リストへの追加

Novel-Press では、何もせずとも各種リストが自動的に用意されています。 まずは CG リストへの追加方法を見ていきましょう。

[cg,1,01]

CG リストの No.1 に、背景画像 01 を追加しました。[p]

[cg, 2, 02]

[cg, 3, 03]

[cg, 4, 04]

CG リストの No.2、3、4 に、背景画像 02、03、04 を追加しました。[p]



この[cg]タグだけで、設定画面の CG リストに追加できます。1 つめのパラメ ータはリストのナンバーで、2 つめは imageB フォルダに入っている背景画像 の指定になります。CG リストは 1 から 100 まで指定できます。つまり背景画 像すべてを追加することも可能ということです。

画像をリストに追加するタイミングですが、これは完全に自由です。グッド エンディングを迎えたあとに一括登録するのもいいですし、その画像が表示さ れたらすぐに登録するのもありです。

imageF フォルダに入っている画像は追加できないわけですが、キャラクター画像も CG リストで見せたいなら、imageB フォルダにコピーすれば可能です。ただしその分、全体の容量が増えてしまいます。

#### ●エンディングリストへの追加

エンディングもほぼ同じ要領でリストに追加することができます。

エンディングリストの No.1 に追加し、タイトル画面に戻ります。[p]

[z,1,エンディング1]



『選択肢とジャンプを使う』の項で解説した[z]タグには、タイトル画面に戻る だけではなく、エンディングリストに追加するための機能もあります。1 つめ のパラメータにはリストのナンバー、2 つめのパラメータはエンディング名を 指定します。

リストに追加できるエンディングは、最大で 100 個までです。エンディング 名は括弧の【]】以外の文字は、何でも使用することができます。字数制限も ありません。しかし、あまり字数が多いと画面からはみ出てしまい、折り返し て表示する機能もありませんので、事実上制限はあります。ほどほどの長さに しましょう。

また、エンディングリストに追加はしたいけどタイトル画面に戻したくない という場合は、次のように記述します。

エンディングリストの No.1 に追加し、ゲームを続けます。[p]

[Z,1,エンディング1]

ゲームを続けました。[p]

大文字で[Z]とすると、タイトル画面には戻らずそのまま進行します。エンデ ィングを迎えたが、後日談のシナリオが続く……というケースで使えると思い ます。

この機能を応用すれば、エンディング以外にもシナリオ内で出てきたヒント や暗号などを登録することができます。プレイヤーはいつでもその内容を確認 できるのです。ミステリーやホラー作品で採用できるでしょう。こうした工夫 で他の Novel-Press 作品と差別化を図るのも面白いはずです。

ただしその場合にはエンディングリストとの併用はできなくなります。加え て、設定画面の「ED-List」というデフォルトの項目名を変更する必要があり ますが、これは第6章で解説します。

### ●BGM リストへの追加

BGM もやはり同じようにリストに追加できます。使用するのは[bgm]タグで、 [bgmon]と似ていますが、まったく別のものです。

BGM リストの No.1 に、サウンドファイル 01 を追加しました。[p] 【設定】【CG-List】【ED-List】【BGM-List】 No.1 BGM01 No.2 No.3 No.4 No.5
【設定】【CG-List】【ED-List】【BGM-List】 No.1 BGM01 No.2 No.3 No.4 No.5
No.1 BGM01 No.2 No.3 No.4 No.5
No.3 No.4 No.5

パラメータは順にリスト番号、サウンドファイル番号、リストに追加する BGM 名になります。CG やエンディング同様に、100 個まで追加することがで きます。そして追加された BGM は、いつでも聞くことができるのです。

▶ BGM01

## 『コメントでソースを見やすくする』

#### ●シナリオ作業の効率化を図る

シナリオが長くなると、作っているクリエイター自身も「ここはどこのシーンだっけ」などと悩んでしまうケースがあります。

そういったことを防ぐために、画面上には表示されないコメント機能の活用 が、ゲーム制作においては一般的です。Novel-Press でのコメントは、次のよ うにします。

[■これはコメントです]

上のコメントは表示されません。[p]

第3章でも軽く触れましたが、【[]で開始する文字列はすべてタグであると 認識されます。当然ながら、存在しないタグを書いても機能しないのですが、 これを利用してタグでない文字列をコメントとすることができるのです。

上のコメントは表示されません。[p]

コメントには何を書いても構わないので、できるだけマークを派手にすると よいでしょう。すぐにコメントだとわかるようになります。

具体的にどうコメントを使うかですが、各シーンの冒頭で「シーン〇〇」と 書いたり、フラグのオンや条件分岐のシーンで活躍します。

[※※※※※シーン 01※※※※※]  $[@ \sim - ~ 01]$ どちらへ行こう?[p] [select. A: 左へ行く,@シーン 02, B:右へ行く,@シーン 03] [※※※※※シーン 02※※※※※]  $[@ \sim - ~ 02]$ 罠にはまってケガをしてしまった。[p] [flagon,1] [jump,@シーン 04] [※※※※※シーン 03※※※※※]  $[@ \sim - \sim 03]$ 何も起こらなかった。[p] [jump,@シーン 04] [※※※※※シーン 04※※※※※]  $[@ \sim - ~ 04]$ 

さて、何があるのか……。[p]
[☆★☆★☆フラグ1がオンならバッドエンド1へ☆★☆★☆]
[if,1,@バッドエンド1]
無事にクリアした。[p]
[z]
[※※※※バッドエンド1※※※※]
[@バッドエンド1]
ケガが悪化して倒れた。俺の運命はここまでだ……。[p]
BAD END[p]

何もコメントがない状態と比べると、格段にわかりやすくなっていると思い ます。用途によってコメントのマークを使い分けていることにも注目してくだ さい。

## <u>『スタッ</u>フファイルを作成する』

### ●スタッフを記載するファイル

制作の仕上げとして、スタッフファイルを作成します。タイトル画面の「STAFF」を選ぶと表示されるのがこのファイルです。

まずは txt フォルダに staff.txt を新規作成してください。その中に、以下の ように記述します。

```
[txtsize.12]
[align,l]
■シナリオ[1]
[align,c]
Aさん[1]
[align,l]
■グラフィック[1]
[align,c]
Bさん[1]
[align,l]
■音楽[1]
[align,c]
Cさん[1]
[align,l]
■制作[1]
[align,c]
ABCプロジェクト
```

このように、通常のシナリオファイルとほぼ同じ文法です。スタッフファイ ルで使用できるタグは次のとおりです。

	スタッフファイルで使用できるタグ
[1]	シナリオファイルで使用するのと同じ機能です。
[txtcolor]	シナリオファイルで使用するのと同じ機能です。
[txtsize]	シナリオファイルで使用するのと同じ機能です。
[align]	文字の表示位置を変更するタグです。

スタッフファイル専用のタグとして[align]があります。パラメータが1つあ り、「l」と指定すると左揃えになります。「c」だと中央揃え、「r」だと右揃え です。スタッフリストを見やすくするためには、[align]タグで役職と名前の位 置を分けるというのが基本になります。あるいはサイズやカラーを工夫しても よいでしょう。

	(STAFF)
<ul> <li>■シナリオ</li> <li>■グラフィック</li> <li>■音楽</li> <li>■制作</li> </ul>	Aさん Bさん Cさん
AB	Cプロジェクト

スタッフ画面には自動折り返し機能がないので、一行があまり長くなっては いけません。適当な位置で[1]タグを挿入する必要があります。縦の長さに関し ては、一画面に収まらなくなった場合は上下にスクロールできます。

フリー素材を使用している場合、「作品内にアドレスを明記すること」を使用 条件にしていることもありますので、ここに記載するとよいでしょう。

### ●スタッフ以外の内容を記述する

スタッフファイルという名称ですが、何もスタッフ名しか記述できないわけ ではありません。プレイにあたっての注意事項や操作方法など、内容は自由な のです。スタッフロールはゲームの最後に流す、という考えの人も多いでしょ うから、むしろスタッフ以外の内容を記述するほうが有効活用できるといえま す。

[align,c]
[txtsize,12]
■読み進める■[1]
決定、右、下キー[1]
[1]
■バックログ■[l]
左、上キー[1]
[1]
■セーブ&ロード■[l]
右ソフトキー[1]
[1]
■メニュー画面■[1]
左ソフトキー[1]
[1]
■オートプレイ■[1]
#キー[1]
[1]
■テキストの一時消去■[1]
上記以外のキー



これは従来型携帯における操作方法を解説したものです。操作方法は作品リ リースの際に自分のサイトに掲載すればよいのですが、外部のアプリマーケッ トからダウンロードした人は、制作者のサイトをあまり見ないケースもありま す。そこでこのようにすれば、必ず目に留まるでしょう。

なお、スタッフ画面上部の文字列は、タイトル画面と連動しています。この 例では「MANUAL」ですが、タイトル画面設定をそのようにしたので、スタ ッフ画面にも自動的に反映されているのです。

# 第5章

# アプリを組み立てよう

## 『開発環境をインストールする』

## ●Java アプリケーション開発に必須の JDK

ゲームが実際にどのように表示されるかは、それぞれのプラットフォームに 合わせた開発環境をインストールして、エミュレーターで確認することになり ます。

本章ではリリースを予定しているプラットフォーム以外の解説は読み飛ばし ても構いませんが、まずは全プラットフォームで必要な Java Development Kit (以降、JDK)というものをインストールしなければなりません。

JDK は Novel-Press 作品に限らず、Java アプリケーションを開発するなら 必須になります。Oracle 社のウェブサイトにアクセスして、ダウンロードペー ジから最新版を手に入れてください。

#### [Oracle]

http://www.oracle.com/

Java SE の JDK を選び、ライセンスに同意するとチェックを入れたら、 [Windows x86] の欄にあるファイルを選択します。ダウンロードが終わった ら、指示に従ってインストールしてください。JavaFX というのが同梱されて いますが、これはインストールしなくても構いません。

JDK はしばしばバージョンアップしますので、その都度アップデートしたほうがよいでしょう。

#### Java SE Development Kit 7u3

You must accept the <u>Oracle Binary Code License Agreement for Java SE</u> to download this software.

Product / File Description	File Size	Download
Linux x86	63.65 MB	差 jdk-7u3-linux-i586.rpm
Linux x86	78.66 MB	差 jdk-7u3-linux-i586.tar.gz
Linux x64	64.53 MB	差 jdk-7u3-linux-x64.rpm
Linux x64	77.3 MB	idk-7u3-linux-x64.tar.gz
Solaris x86	135.96 MB	idk-7u3-solaris-i586.tar.Z
Solaris x86	81.4 MB	差 jdk-7u3-solaris-i586.tar.gz
Solaris SPARC	138.92 MB	差 <u>idk-7u3-solaris-sparc.tar.Z</u>
Solaris SPARC	86.07 MB	差 <u>idk-7u3-solaris-sparc.tar.gz</u>
Solaris SPARC 64-bit	16.14 MB	idk-7u3-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	12.31 MB	差 idk-7u3-solaris-sparcv9.tar.gz
Solaris x64	14.46 MB	idk-7u3-solaris-x64.tar.Z
Solaris x64	9.25 MB	差 <u>idk-7u3-solaris-x64.tar.qz</u>
Windows x86	84.12 MB	idk-7u3-windows-i586.exe
Windows x64	87.41 MB	idk-7u3-windows-x64.exe

#### O Accept License Agreement O Decline License Agreement

### ●i アプリの開発ツール

i アプリの開発環境を整えるには、NTT docomo のサイトから専用のツール を手に入れる必要があります。

#### 【DoJa-5.1 プロファイル向けiアプリ開発ツール】

<u>http://www.nttdocomo.co.jp/service/developer/make/content/iappli/tool/doj</u> <u>a/doja51/index.html</u>

ここにある「i  $\alpha$  ppli Development Kit for DoJa-5.1」(以降、iappliTool)を、 使用許諾契約書に同意してダウンロードしたら、指示に従ってインストールし ます。デフォルトでは C ドライブ直下にインストールする設定です。特に理由 がないのなら、変更はしないでください。 iαppli Development Kit for DoJa-5.1のダウンロード

iαppli Development Kit for DoJa-5.1をダウンロードできます。

※ iappli Development Kit for Do Ja-5.1のダウンロードには使用許諾契約書への同意が必要となります。
※ 各ツールの操作方法については、同梱されている各操作マニュアルをご参照ください。

≥ iαppli Development Kit for DoJa-5.1のダウンロードへ

他のプラットフォームの開発環境もそうですが、JDK がない状態ではインス トールに失敗するので、必ず先に JDK をインストールしておいてください。

#### ●オープンアプリの開発ツール

au のオープンアプリ対応機種と EZ アプリ(J)対応機種の両方でプレイできる、300KB までの作品をリリースしたい場合は、「Sun Java Wireless Toolkit 2.5.2」(以降、WTK) というツールを使用します。

#### [Sun Java Wireless Toolkit 2.5.2 for CLDC]

http://www.oracle.com/technetwork/java/download-135801.html

このページには「WTK は Java ME SDK に統合されたので、よければそち らを」と書かれています。Java ME SDK は WTK よりも新しいモバイルアプ リの開発ツールですが、Novel-Press では WTK のほうが推奨されていますの で、本書でもそのように解説を進めていきます。

下のほうにダウンロードボタンがありますので、それをクリックすればファ イル置き場にジャンプします。JDK と同じく、ライセンスに同意してダウンロ ードしてください。

インストール中には JDK を指定する工程があります。そのため、JDK をバ ージョンアップ・ダウンすると WTK が起動できなくなることがあります。そ の場合には WTK を再インストールしてください。

#### Sun Java Wireless Toolkit 2.5.2 01

You must accept the <u>Sun Java Wireless Toolkit License Agreement</u> to download this software.

> O Accept License Agreement O Decline License Agreement

Product / File Description	File Size Download
Sun Java Wireless Toolkit 2.5.2_01 for	37.95 差 <u>sun java wireless toolkit-2.5.2_01-</u>
CLDC	MB <u>linuxi486.bin.sh</u>
Sun Java Wireless Toolkit 2.5.2_01 for	37.36 👱 sun java wireless toolkit-2.5.2 01-win.exe
CLDC	MB
	<u>Back to top</u>

## ●EZ アプリ(J)の開発ツール

auの2MBまで容量を使えるEZアプリ(J)対応機種向けにリリースするには、 2種類のツールが必要です。ひとつは「Eclipse」という統合開発環境です。統 合開発環境とは、簡単にいえばプログラミングに必要なものをひとつにまとめ たツールのことで、多くのゲーム開発者が利用しています。プログラムをよく 知らない人でも、慣れればそう難しいことはありません。

#### [Eclipse]

http://www.eclipse.org/

ダウンロードページに様々なファイルがありますが、「Eclipse IDE for Java Developers」を選択します。

Eclipse にはインストーラーがないので、ダウンロードして解凍したフォル ダを、好きな場所に移動してください。本書ではCドライブ直下に移動させた という前提で解説していきます。どこに移動するにせよ、実行ファイルのショ ートカットを作成すると便利です。アンインストールの際は、解凍したフォル ダをそのまま削除すれば完了です。

Compare Packages Older Versions	Eclipse Indigo (3.7.1) Packages for Windows 🛛 💌
Comparison of the second state of the second s	Windows 32 Bit Windows 64 Bit
Eclipse IDE for Java EE Developers, 212 Downloaded 2,696,301 Times Details	MB Windows 32 Bit Windows 64 Bit
Eclipse Classic 3.7.1, 174 MB     Downloaded 1,241,158 Times Details Other Dow	mloads Windows 32 Bit Windows 64 Bit
C+ Eclipse IDE for C/C++ Developers (inclu 107 MB Downloaded 799,446 Times Details	ides Incubating components), Windows 32 Bit Windows 64 Bit

また、Eclipse は有志が日本語化パッチを公開していますので、そちらのほうがいいという人は検索して入手しておきましょう。本書では英語の設定のままで解説していきますので、日本語化した人はメニュー用語などを適宜読み替えてください。

もうひとつの必要なツールは「EZ アプリ(J)SDK」です。「EZ アプリ(J)エミ ュレータ」と「EZ アプリ(J) Eclipse Plugin」の 2 つのツールの総称で、au の 開発者向けサイト EZfactory から入手します。

#### [EZfactory]

http://www.au.kddi.com/ezfactory/index.html

動作環境を確認して、使用許諾契約書の同意にチェックしたら、ダウンロー ドしてください。

これをインストールすることで、Eclipse において EZ アプリ(J)の開発とエ ミュレーターの起動が可能になります。 「EZアプリ(J)SDK」の動作推奨環境

	<i>م</i>	
ピアプリ(のエミュレー	3	
必要なソフトウエア	<ul> <li>Windows XP Professional version 2002 Service Pack 3(※)以上</li> <li>日本語版のみに対応</li> <li>動作確認は以下のバージョンでおこなっています。</li> <li>Windows XP Professional version 2002 Service Pack 3 日本語版</li> <li>Windows Vista Business Upgrade Service Pack 2 日本版</li> <li>Windows 7 Professional 日本語版</li> </ul>	
必要なハードウエア	上記「必要なソフトウエア」で説明しているソフトウエアが問題なく動作するPC	
EZアプリ(J)Eclipse P	lugin	
必要なソフトウエア	・Microsoft Windows XP Professional 以降 Microsoft Windows XP Professional operating system Service Pack 3 で 動作 確認済 ・Celipse パージョン 3.4 以降 Eclipse 3.6(Herios)に日本語化プラグイン(Pleiades)1.3を適用した環境で動作確 認済 J2SE Runtime Environment (JRE(TM))パージョン 5 以降 JRE 6 で動作確認済	
必要なハードウエア	ードウエア 上記「必要なソフトウエア」で説明しているソフトウエアが問題なく動作するPC	
EZアプリ(J)SDK	10.4MB ២ <u>ダウンロード</u>	
	エ このページの先頭へ	

### ●S!アプリの開発ツール

EZ アプリ(J)と同様に、S!アプリは Eclipse で開発します。必要なツールは ソフトバンクの開発者向けサイト Mobile Creation で入手します。

#### [Mobile Creation]

 $\underline{http://creation.mb.softbank.jp/mc/home/index.html}$ 

トップページから [技術仕様] → [S!アプリ] → [ドキュメント/ツール] と 移動してください。ここで「S!アプリ開発ツール MEXA SDK」と「Eclipse 向けプラグイン for MEXA SDK」をそれぞれダウンロードしてください。「S! アプリ開発ツール MEXA SDK」のほうがエミュレーターになります。

ツール名	内容	Ver.	作成日
<u>S!アプリ開発ツール</u> <u>MEXA SDK</u>	SIアブリの動作をPC上で確認するための開発支援ツールです。	2.3.1	2009/10/8

ツール名	内容	Ver.	作成日
<u>Eclipse向けプラグイン</u> for MEXA SDK	統合開発環境eclipse上でのS!アプリ開発を支援するブラグインです。	1.0	2007/2/9

### ●Android アプリの開発ツール

Android 版もまた、Eclipse で開発することになります。必要なものは「Android SDK」と「ADT プラグイン」です。Android SDK は Google のサ ービス Android Developers から入手します。

#### [Android Developers]

http://developer.android.com/index.html

メニューの [SDK] から installer\_r16-windows.exe をダウンロードしてく ださい。

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r16-windows.zip	29562413 bytes	6b926d0c0a871f1a946e65259984701a
	installer_r16-windows.exe (Recommended)	29561554 bytes	3521dda4904886b05980590f83cf3469
Mac OS X (intel)	android-sdk_r16-macosx.zip	26158334 bytes	d1dc2b6f13eed5e3ce5cf26c4e4c47aa
Linux (i386)	android-sdk_r16-linux.tgz	22048174 bytes	3ba457f731d51da3741c29c8830a4583

これをインストールしたら、Eclipse を起動して [Window] → [Preferences] を選択してください。この左のリストで [Android] を選択して、インストールした SDK のフォルダを指定します。そうしたら [OK] を押して、第一の準備は完了です。

次は ADT プラグインのインストールです。Eclipse のメニュー [Help] → [Install New Software] を選択します。そうしたら右上のボタン [Add] を 押して、以下のように入力します。

#### Name.....ADT Plugin

#### Location.....https://dl-ssl.google.com/android/eclipse/

これで [Work with] に [ADT Plugin] を指定できるようになりました。表示された [Developer Tools] にチェックを入れましょう。あとは先に進んでラ イセンスの内容を確認し、[Finish] を押せばダウンロードとインストールが行 われます。最後に再起動を求めるダイアログが表示されるので、それに従って 再起動します。

メニューの [Window] に [Android SDK Manager] というのが新たに出て いるはずですので、これを起動します。Android アプリ開発に必要なツールを 組み込むための作業を行うのですが、Novel-Press 作品では Android2.1 以上で 動作しますので、[Android2.1 (API7)] の [SDK Platform] を選択してくだ さい。他にチェックが入っているものは外して、[Install packages] のボタン を押します。

🖞 Android SDK Manager							
Packages Tools							
SDK Path: C:¥android-sdk-windows¥							
- Packages							
🟺 Name	API	Rev.	Status				
e Cools							
- Android SDK Tools		16	installed 🕘				
🔄 📑 📸 Android SDK Platform-tools		10	ay Installed				
Endroid 2.1 (API /)	7	2	🌦 Installed				
	r	0					
Show: Updates/New VInstalled Obsolete Select New	or <u>Upda</u>	ates		Install packages			
Sort by:  API level O Repository Deselect A	Ш			Delete packages			
Done loading packages.							

最後に、実機でデバッグするために、各 Android 端末に合わせた USB ドラ イバをインストールしなければなりません。「端末名 実機 デバッグ」といった キーワードで検索して入手しましょう。

準備はこれで完了です。こうして見ると Android での開発は準備だけでもかなり工程が多いですが、何回かやっていくうちに慣れてくるはずです。

## 『Java ソースコードをダウンロードする』

#### ●Novel-Press の根幹となるエンジン

開発環境のインストールが終わったら、Novel-Press 公式サイトから Java ソースコードを入手します。言わば Novel-Press の根幹となるエンジン、最重 要のファイルです。

メニューの [製作手順] から [Java ソースコード] のページに移動して、リ リースを予定しているプラットフォームのソースコードをダウンロードしてく ださい。過去のバージョンも置いてありますが、基本的には最新版だけで充分 です。

フォルダを解凍すると、プラットフォームによって微妙に中身は異なります。 java と拡張子のついたファイルがソースコードです。

#### 【IAP 版】

ソースファイル5種類

#### 【OAP 版】

ソースファイル 4 種類、HTACCESS ファイル、Download.xhtml

#### 【SAP 版】

ソースファイル4種類

#### 【ADR 版】

ソースファイル 16 種類、マニフェストファイル

OAP 版と ADR 版は、ソースコード以外のファイルもありますが、後々解説 していきます。

さてソースコードを見てみると、プログラムの羅列なので、知識のない人に はさっぱりわからないと思います。第6章ではこれを少し改造する方法を解説 しますが、文法などを覚える必要はありません。
### ●ソースコードを新しいものに代える時は

ゲームを制作中、バグ修正などで Novel-Press がバージョンアップしたら、 できる限りソースコードを新しいものに変えましょう。その際には、開発ツー ルの保存情報を初期化する必要があります。

初期化をしないと、動作に不具合が出るおそれがありますので、必ず実行し てください。

各開発ツールの保存情報初期化方法					
iアプリ	iappliTool のメニュー [ツール] → [スクラッチパッドを				
	全て初期化]を実行				
オープンアプリ	WTK $\mathcal{O} \not\prec = \neg - [File] \rightarrow [Utilitys] \rightarrow [Clean Database]$				
	を実行				
EZ アプリ(J)	EZ アプリ(J)エミュレータを直接起動して、[電源キー]→				
	[決定キー] → [アンインストール] → [作品名] → [は				
	い] → [OK] の順に実行				
S!アプリ	MEXA エミュレーターを直接起動して、メニューの [S!ア				
	プリ]→[電源キー]を選び、端末画面が出たら[決定キ				
	ー] → [Uninstall] → [作品名] の順に実行				
Android アプリ	SD カードの [パッケージ名] フォルダを削除				

## 『config ファイルを作成する』

### ●作品の基本設定をするファイル

config ファイルは文字の大きさや表示速度、音量といった、作品の基本設定 を行うファイルです。公式サイトの[製作手順]→[config ファイル]→[config ファイル生成ページ]へと移動してください。このページのフォームに各項目 を入力して生成ボタンを押すだけで、config ファイルが作られデスクトップに 保存されます。

決められる項目は以下のとおりです。

作品のタイトル					
作品のタイトルです。					
サイトの URL					
タイトル画面からアクセスできる URL を入力します。					
タイトルメニュー					
メニューの項目	デフォルトでは				
	START (シナリオを開始する)				
	LOAD(データを読み込む)				
	MENU (メニュー画面を表示する)				
	STAFF (スタッフ画面を表示する)				
	となっています。				
	この名称を自由に変えられ、文字揃えを中央揃え、左				
	揃え、右揃えから選べます。				
始点 X	タイトルメニューの左上の X 座標を指定します。				
始点 Y	タイトルメニューの左上の Y 座標を指定します。				
文字の大きさ	項目の文字の大きさです。2から126の間の偶数を指				
	定します。				
余白	項目と項目の間の間隔です。				

タイトル画面での BGM 演奏						
ここで「演奏する」をチェックすると、タイトル画面でサウンドファイル 00						
が読み込まれます。また、フェードインの時間を設定できます。						
コンフィグの初期設定						
文字の表示速度	第1章で解説したとおりです。					
オートプレイのウェイト プレイヤーが自由に変更できるかを設定すること						
文字枠の透明度	ができます。					
音量						
バイブレーション						
クリック待ちの表示						
改頁待ちの表示						
	CGリスト					
チェックを入れると、作中	で使用した背景画像を閲覧できます。					
数は1から100まで指定できます。						
EDリスト						
チェックを入れると、到達したエンディングのリストを閲覧できます。						
数は1から100まで指定できます。						
BGM リスト						
チェックを入れると、作中で使用した BGM を視聴できます。						
数は1から100まで指定で	きます。					
	オートプレイ機能					
オートプレイ機能の ON、	OFF を設定します。					
既読選択肢表示機能						
その選択肢を選んだことがあるか、視覚的にわかりやすくする機能の ON、OFF						
を設定します。						
縦書き機能						
テキストの縦書き機能の ON、OFF を設定します。(IAP 版のみ対応)						
セーブファイルの数						
セーブスロットを 1~15 の範囲で指定します。						

システム画面の配色				
背景の色	RGB 値で配色を設定します。			
カーソルの色	文字の色はここで指定した「濃い色」と、背景色			
文字の色	との中間の「薄い色」があります。			
エラー詳細情報の表示機能				

プレイ中にエラー詳細情報を表示する機能の ON、OFF を設定します。

### ●ファイルはあとで編集できる

各項目の設定を変えたくなった場合、config ファイルをあらためて作成し直 す必要はありません。純粋なテキストファイルなので、直接開いて編集するこ とができます。

config ファイルの中では、項目名はすべて半角のアルファベットになってい て、「作品のタイトル」なら「appName」という具合です。生成ページの項目 名と照らし合わせて、どれがどれに対応しているのか把握しましょう。

機能の ON か OFF を設定する項目では、ON の場合は「on」としますが、 OFF の場合は何も書かないことになっています。

ciMarkC =;	
ciMarkP =on;	

これはクリック待ちのマークが OFF で、改頁待ちのマークが ON という意味になります。OFF にする場合でも、行末のセミコロン【;】を削除してはいけません。

112

## 『アイコンを作成する』

### ●作品を区別するアイコン

端末にアプリをダウンロードすると、一覧にアプリが並ぶわけですが、それ ぞれのアイコンが表示されます。Novel-Press で制作するアプリでもアイコン を使うことができます。

各プラットフォームに対応するアイコンのサイズ、ファイル形式、ファイル 名は以下のようになります。

プラットフォーム	サイズ	ファイル名
:アプリ	$48 \times 48$	icon48.gif
1///	$160 \times 160$	icon160.gif
オープンアプリ	$16 \times 16 \text{ or } 20 \times 20 \text{ or } 24 \times 24$	icon.png
	$32\! imes\!32$ or $64\! imes\!64$	iconS.png
EZ アプリ(J)	$48 \times 48$ or $96 \times 96$	iconM.png
	$192\! imes\!192$ or $384\! imes\!384$	iconL.png
S!アプリ	$24 \times 24$	icon.png
	36×36	
Android	48×48	icon.png
	72×72	

アイコンがなくてもアプリをビルドすることはできますが、そうするとデフ オルトのアイコンになっていかにも個性がないので、理由がない限りはアイコ ンを用意しましょう。

なお、オープンアプリとして制作したアプリを EZ アプリ(J)対応機種でダウ ンロードした場合、例外なくアイコンは表示されず、デフォルトのものになっ てしまいます。これは回避できません。

## 『i アプリの組み立て方法』

### ●プロジェクトを新規作成する

iappliToolを起動すると、[プロジェクト新規作成]というボタンがあります。 これを選ぶと、新規作成ダイアログが表示されます。

新規作成	×
プロジェクト格納先: 参照(R)	
C:¥iDKDoJa5.1¥apps	
プロジェクト名 test	
□ 引き続きソースファイルを作成する	
作成 取消	

プロジェクト格納先はそのままにして、プロジェクト名を決めましょう。こ れがアプリ名にも使われるので、スペースなしの半角英数字で決めてください。 ソースコードはすでに用意しているので、[引き続きソースファイルを作成す る]のチェックは外します。もしチェックを外し忘れたまま作成したら、作ら れた java ファイルは削除してください。

プロジェクトが新規作成されたら、その構成を見てみましょう。4 つのフォ ルダが生成されています。他に 2 つのファイルがありますが、これは無視して 構いません。

	プロジェクトフォルダの中身
bin	ここにアプリが生成されます。プロジェクトを作成した直後は
	jam ファイルのみがあります。
res	素材ファイル、configファイル、アイコンを置きます。
$\mathbf{sp}$	データを格納するスクラッチパッドが生成されます。
src	ソースコードを置きます。

まずは各素材をresフォルダに入れます。つまりresフォルダの中にimageB、 imageF、sound、txtの4つのフォルダがある状態です。それに加えて config ファイルと icon48.gif、icon160.gifの2つのアイコンを入れます。

そしてソースコードを src フォルダに入れたら、組み込みの準備は完了です。 [ビルド]のボタンを押してみましょう。

ADF [LaunchApp],[AccessUserInfo],[UseBrowser],[UseDTV] または [TrustedAPID]の設定が必要です。

トラステッド [ネイティブメーラ利用の許可] の設定が必要です(ADF [TrustedAPID] を設定する場合)。

コンソールにこのようなメッセージが出ます。ADF (Application Descriptor File)とはiアプリの情報を設定するものです。

#### ●ADF を設定する

[ADF 設定]ボタンを押して ADF を編集します。必要なのは次の項目です。

- ① [AppName] に作品のタイトルを入力します。
- ② [AppVer] にアプリのバージョンを入力します。
- ③ [SPsize] に「3000」と指定します。スクラッチパッドで使用する領域サ イズです。
- ④ [UseBrowser] のチェックボックス [launch] にチェックを入れます。
   ブラウザ機能を利用するかどうかの設定です。
- ⑤ [DrawArea] に「240×240」と指定します。アプリの画面サイズです。
- ⑥ [AppIcon] に「icon48.gif」と指定します。小さいアイコンです。
- ⑦ [AppMainTitle] に「icon160.gif」と指定します。大きいアイコンです。

また、AppSize という項目はアプリのサイズです。これを 1024KB(1MB) 以内に抑える必要があります。これを越えていると、ADF のチェックでエラー が出ます。 ADF の設定が終わったら、もう一度ビルドしましょう。設定した情報は bin フォルダの jam ファイルに保存されます。そして jad ファイルがリリースする アプリ本体になります。

### ●エミュレーターでプレイ

ビルドしたアプリは、エミュレーターで実機とほぼ同じようにプレイするこ とができます。[起動]からエミュレーターを起動してください。



タイトル画面が表示されたら、メニューやスタッフ画面、サイト接続を確認 し、ゲーム本編をスタートしましょう。エミュレーターを終了させたい時は、 電源キーを押してください。モバイル特有の機能であるバイブレーションはど うなっているかというと、「デバイス ON/OFF」の欄にバイブレーションマー クが表示されます。

最終的には実機での検証が必須ですが、テストプレイ以前の開発段階ではこ のエミュレーターで作業します。シナリオや素材の更新があった場合、その都 度ビルドし直してバグやミスを修正していくのです。

セーブ機能も実機同様に使えるので、ゲーム終盤などをチェックする際には、 わざわざ最初から読み進めていくことはありません。 適当な場所でセーブして おけば、いつでもロードしてそこをチェックできます。

確認していく中で動作がおかしいと思ったら、スクラッチパッドを初期化し て再度試してみてください。ただしその場合はセーブデータやフラグも消去さ れるので注意しましょう。

なお [ツール] → [スクリーンショット] から画面を保存できるので、ウェ ブサイトでのサンプル画像公開に役立てましょう。画像が自動生成されるわけ ではないので、別途ペイントソフトを起動して貼り付ける必要があります。

# 『オープンアプリの組み立て方法』

### ●プロジェクトを新規作成する

WTK を起動したら [New Project] のボタンを選択します。[Project Name] はスペースなしの半角英数字で入力し、[MIDlet Class Name]は「NovelPress」 と指定します。

自動的に API (Application Programming Interface) 設定画面が起動するの で、下記のとおり設定してください。

🐻 Settings for p	roje	ect "test"	$\overline{\mathbf{X}}$		
e 10 - 10 - 10 - 10 - 10 - 10 - 10 - 10		API Selection			
API Selection					
ē		Target Platform JTWI 💌			
Required		JTWI-			
₹Ô		Profiles MIDP 2.0	Configurations		
Optional			⊙ CLDC 1.1		
User Defined		Required Wireless Messaging API 1.1 (JSR 120)			
<b>F</b>	=	Mobile Media API (JSR 135)			
MIDlets	-	Additional APIs			
		Wireless Messaging API 2.0 (JSR 205)	Web Services API (JSR 172)		
Push Registry		JAXP XML Parser (JSR 172)	PDA Profile for J2ME (JSR 75)		
		Bluetooth/OBEX for J2ME (JSR 82)	Mobile 3D Graphics for J2ME (JSR 184)		
		Location API for J2ME (JSR 179)	SATSA-APDU (JSR 177)		
External APIs		SATSA-JCRMI (JSR 177)	SATSA-PKI (JSR 177)		
		SATSA-CRYPTO (JSR 177)	Content Handler API (JSR 211)		
		Mobile Internationalization API (JSR 238)	Payment API (JSR 229)		
Permissions		SIP API (JSR 180)	Advanced Multimedia Supplements (JSR 234)		
		Scalable 2D Vector Graphics API (JSR 226)	Java Binding for OpenGL ES (JSR 239)		
Content Handlers					
	<b>•</b>		<u>QK</u> <u>Q</u> ancel		

- ① [Target Platform] で「JTWI」を選択します。
- ② [Configurations] で「CLDC 1.1」を選択します。
- [Optional] [Additional APIs] ですべてのチェックボックスのチェック を外します。

プロジェクトが新規作成されたら、その構成を見てみましょう。4 つのフォ ルダが生成されています。project.properties というファイルがありますが、こ れは無視して構いません。

	プロジェクトフォルダの中身
bin	ここにアプリが生成されます。プロジェクトを作成した直後は
	マニフェストファイルと jad ファイルがあります。
lib	何も入っておらず、無視して構いません。
res	素材ファイル、configファイル、アイコンを置きます。
src	ソースコードを置きます。

imageB、imageF、sound、txtの4つのフォルダ、configファイル、icon.png を res フォルダに入れます。そして src フォルダに Java ソースコードを入れま す。最後に [Edit] メニューから [Preferences] を選び、[Storage] の項目で サイズを 32 と指定します。

### ●エミュレーターでプレイ

設定が終わったら [Build] からアプリをビルドしましょう。次に [Run] で エミュレーターを起動します。あとはテキストや画像の調整を繰り返して、納 得いく作品に仕上げてください。

スクリーンショットを撮る機能はありませんので、別途キャプチャソフトを 用意するとよいでしょう。



### ● jad ファイルの編集とパッケージ作成

端末にダウンロードされた時の情報表示に必要な、jad ファイルの編集を行います。まず [Settings]  $\rightarrow$  [Required] を選び、下記のように設定します。

- [MIDlet-Name] に作品名を指定します。
- ② [MIDlet-Vendor] に作者名を指定します。

次に [Optional] の項目で、下記のように設定します。

- ① [MIDlet-Data-Size] に「32768」と指定します。
- ② [MIDlet-Icon] に「/icon.png」と指定します。

設定が終わったら、いよいよ [Project] のメニューから [Package] → [Create Package] と選びます。これで bin フォルダに jad ファイルと jar ファイルが 作成されます。

# 『EZ アプリ(J)の組み立て方法』

### ●プロジェクトを新規作成する

EZ アプリ(J)は、ソースコードはオープンアプリと同じものを使用しますが、 組み立て方はまったく異なります。Eclipse の起動後、[File] → [New] → [Project] → [EZ アプリ(J)] → [EZ アプリ(J)プロジェクト] の順で選んで いきましょう。

🌲 新規 EZアプリ(J) プロジェクト 📃 📃	
EZアプリ(J) エミュレータの設定 EZアプリ(J) エミュレータ実行パスおよびビルド用クラスパスの設定(オプショナル)	
▽実行パスを指定する⊗	
EZアプリ(J) エミュレータ実行パス(E) C:¥Program Files¥EZAPLI_J_EMULATOR 参照	(R)
ビルド用クラスパスを回接指定するセン ビルド用クラスパス:©) C¥Program Files¥EZAPLI_J_EMULATOR¥lib¥stubclasses.zi Browse	
□ テンプレートを利用する(1)	
Cance Next > Einish Cance	el

まず [Project name] はスペースなしの半角英数字で入力し、[Use Default Location] のチェックはそのままにして、[Next] を選びます。

次の画面では[実行パスを指定する]と[ビルド用クラスパスを直接指定する]にチェックを入れます。[EZアプリ(J)エミュレータ実行パス]と[ビルド 用クラスパス]の欄は、すでにデフォルトで指定されていると思います。され ていない場合は、例の画像のように指定してください。最後に[Finish]を押 します。

以上で「Eclipse のインストール先/workspace」内に、プロジェクトフォル ダが作られました。いくつかのフォルダとファイルが作成されていますが、オ ープンアプリと同じ要領で imageB、imageF、sound、txt の 4 つのフォルダ、 config ファイル、iconS.png、iconM.png、iconL.png を res フォルダに入れま す。そして src フォルダに Java ソースコードを入れます。

### ●アプリのビルドと jad ファイルの編集

画面左の Package Explorer でプロジェクト名を選択し、その状態で F5 キー を押してください。右クリックから [Refresh] でも構いません。これでアプ リがビルドされます。

そうしたら [プロジェクト名.jad] を右クリックして [Open With]  $\rightarrow$  [EZ マークの付いている JadEditor] を選択してください。ここで jad ファイルを 編集します。

- [必須属性]の[MIDlet-Name]に作品名を、[MIDlet-Vendor]に作者名 を、[MIDlet-Version] にバージョンを入れます。
- [起動 MIDlet の選択]の[使用]にチェックを入れ、[MIDlet 名]に作 品名を入れます。
- ③ [EZ アプリ(J)属性]の[OAP-Icon-S]に「iconS.png」を入力します。
   以下同様に[OAP-Icon-M]に「iconM.png」を、[OAP-Icon-L]に「iconL.png」を入力します。
- ④ [ユーザー定義属性]の欄に「MIDlet-Data-Size: 5000」という一行を追加します。
- ⑤ [File] メニューの [Save All] を選択します。

🔁 ezappj_badend.jad 🛛				
既存マニフェストファイル:				<u>B</u> rowse
読み込む				
必須属性				
MIDlet-Name BAD END				
MIDlet-Vendor アライコウ				
MIDlet-Version 1.00				
起動MIDletの選択				
使用 クラス名		MIDlet名	アイコン	
NovelPress		BAD END		
記動MIDletとして登録したいものにチェックを	を入れ、MIDlet名の	必須)を指定してください。	、アイコンの指定は任意で	です。)
EZアブリ(J) 属性				
+-	値			
OAP-Icon-S	iconS.png			
OAP-Icon-M	iconM.png			
OAP-Screen-Size	ICONE.pris			
ユーザー定義属性(名前: 値)				
MIDlet-Data-Size: 5000				~
				~
				>

これでテストプレイの準備ができました。

### ●エミュレーターでプレイ

エミュレーターの起動は [プロジェクト名.jad] の右クリックから [Run As] → [EZ アプリ(J)エミュレータ] で行います。Soft1 ボタンがメニュー画面で、 Soft2 ボタンが栞画面になります。バイブレーションの時は、エミュレーター 画面全体が揺れるようになっています。

こちらもスクリーンショットを撮る機能はありませんので、別途キャプチャ ソフトを用意するとよいでしょう。



## 『S!アプリの組み立て方法』

### ●プロジェクトを新規作成する

Eclipse のメニューを [File] → [New] → [Project] → [MEXA] → [MEXA プロジェクト] と選択していきます。

EZ アプリ(J)と違うのが、[Project name] を入力し [Use Default Location] のチェックはそのままにしたら、[Next] ではなく [Finish] を選ぶことです。

プロジェクトフォルダが「Eclipse をインストールしたフォルダ/workspace」 に作成されましたが、まず res フォルダと src フォルダを自分で新規作成しな ければなりません。そうしたら res フォルダの中に imageB、imageF、sound、 txt の 4 つのフォルダ、config ファイル、icon.png を入れ、src フォルダに Java ソースコードを入れます。この状態で一度、画面左の Package Explorer でプ ロジェクト名を選択し、F5 キーを押してください。

### ●パスを編集する

次に Package Explorer でプロジェクトを右クリックし、[Properties] を選 択してください。手順は以下のとおりです。

- ① [MEXA Emulator 設定]の[MEXA Emulator 実行パス]欄に、MEXA SDK をインストールしたフォルダ(「C:¥Program Files¥SOFTBANK\_MEXA\_EMULATOR23」など)を指定します。
- [JavaBuild Path]の[Source] タブで [Add Folder] を選択し、res フ オルダにチェックを入れて [OK] を押します。
- ③ [JavaBuild Path]の[Libraries]タブで[Edit]を選択し、表示される stubslasses.zipを指定します。見つからない場合は「MEXA SDK をインストールしたフォルダ/lib」を探してください。



## ● jad ファイルを編集する

パスの編集が済んだら[プロジェクト名.jad]を右クリックして[Open With] → [SoftBank マークの付いている JadEditor] を選択してください。

- [必須属性]の[MIDlet-Name]に作品名を、[MIDlet-Vendor]に作者 名を、[MIDlet-Version]にバージョンを指定します。
- ② [起動 MIDlet の選択]の[使用] にチェックを入れ、[アプリケーション名] に作品名を指定して、[アイコン] のプルダウンメニューで「/icon.png」を選択します。
- ③ [ユーザー定義属性]の欄に「MIDxlet-API: MEXA」「MIDlet-Data-Size:
   5000」「MIDxlet-ScreenSize: 240,260」「MIDlet-Icon: icon.png」の4行 を入力します。

= s_badend.jad	X			
既存マニフェストフ	Pr1.1k		<u>B</u> rowse	
読み込む				
必須属性				
MIDlet-Name	BAD END			
MIDlet-Vendor	לברק			
MIDlet-Version	1.0			
起動MIDletの選	択			
使用   クラス	(名	アプリケーション名	アイコン	
Nove	IPress	BAD END	£	
アプリケーションとし	」 して登録したいものにチェックを入れ、名前(必	 須)とアイコン(オプショナル)を指定してください。		
ユーザー定義属性	生(名前: 値)			
MIDxlet-API MEXA				^
MIDx1et-Size: 2000 MIDx1et-ScreenSize: 240,260				
INIT Net-TCOU: IC(	on.png			
				~
<			>	

## ●エミュレーターでプレイ

エミュレーターの起動は [プロジェクト名.jad] の右クリックから [Run As] → [MEXA Emulator] で行います。ダイアログが出るので [Untrusted でイ ンストール] にチェックが入っていることを確認して、[OK] をクリックして ください。

操作はキーパッドで行い、終了時には赤い電話のボタンをクリックします。 [Quit?]と聞かれるので、終了するなら[Yes]を、元の画面に戻るなら[No] をクリックします。

このエミュレーターにもスクリーンショットを撮る機能はありません。なの で、iアプリでリリースする予定がない場合でも、スクリーンショットを撮る ためだけにiアプリの開発環境を整えるという手もあります。



## 『Android アプリの組み立て方法』

### ●プロジェクトを新規作成する

Eclipse のメニューを [File] → [New] → [Project] → [Android] → [Android] Project] と選択していきます。

まずは [Project name] をスペースなしの半角英数字で入力し、[Create new project in workspace] と [Use default location] にチェックが入っていることを確認して、[Next] を押します。

次に [Build Target] に [Android2.1] を選択し、 [Next] を押します。

最後に [Application Info] の設定です。[Application name] に作品名を、 [Package name] に任意のパッケージ名を入れ、[Create Actvity] のチェッ クを外します。

2 つめのパッケージ名は、スマートフォンではスタンダードな仕組みで、作 品名とは違い、アプリケーションを識別するための名前です。内容は異なって いても世界のどこかに同じタイトルのアプリが存在することを考慮し、混同さ れないように独自のパッケージ名をつける必要があるのです。

パッケージ名の付け方は、自分のサイトのドメインを逆にして、それにアプ リのタイトルをつけるのが一般的です。

例:net.projectlips.test

👄 New Android Project 📃 🗔 🔯		
Application Info Configure the new A	Android Project	
Application Name:	Test	
Package Name:	net.projectlips.test	
Create Activity:	TestActivity	
Minimum SDK:	7	
Create a Test Pr	roject	
Test Project Name:	test-test	
Test Application:	TestTest	
Test Package:	test-test	
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish Canc	el

#### ●従来型携帯と違う素材構成

Android アプリを作る場合の素材構成は、従来型携帯と少し違います。プロ ジェクトフォルダの中に assets というフォルダがありますが、ここに imageB、 imageF、sound、txt の 4 つのフォルダ、config ファイルを入れてください。

アイコンを置くのは res フォルダです。drawable-ldpi フォルダに icon.png (小)、drawable-mdpi フォルダに icon.png (中)、drawable-hdpi フォルダに icon.png (大) をそれぞれ入れます。デフォルトのアイコンは削除してくださ い。

そして Java ソースコードは、ダウンロードして解凍した際に生成された novel フォルダを、そのまま src フォルダに入れます。そうしたら Package Explorer でプロジェクト名を選択し、F5 キーを押します。

あとは、以下のとおりに設定してください。

- [プロジェクト名]→[src]→[パッケージ名]を右クリックし、[Delete]
   を選択、[OK]を押します。
- ② [プロジェクト名]→[src]→[novel.press]を右クリックし、[Refactor]
   → [Rename] を押します。
- ③ [New name] にパッケージ名を入れ、[OK] を押します。途中でダイア ログが出ますが構わずに [Continue] を押してください。

🚔 Rename Package 📃 🗔 🔀		
New na <u>m</u> e: net.projectlips.test		
Update references		
Rename <u>s</u> ubpackages		
Update textual occurrences in comments and strings (forces preview)		
Update fully gualified names in non-Java text files (forces preview)		
File name patterns: *		
The patterns are separated by commas (# = any string, ? = any character)		
Previe <u>w</u> > OK Cancel		

### ●マニフェストファイルの編集

ダウンロードした Java ソースコードには、AndroidManifest というマニフ ェストファイルがついています。これは Android アプリの設計書のようなもの です。

これをプロジェクトフォルダにすでにあるファイルに上書きしたら、テキス トファイルで開いてください。3行目に「package="novel.press"」とあるので、 「novel.press」の部分をパッケージ名に変更します。

### ●実機でテストプレイ

従来型携帯と違い、Android アプリは実機でテストプレイすることになりま す。上部メニューの [Run]  $\rightarrow$  [Run Configurations] を選択したら、[Android Application] をダブルクリックします。あとは以下のとおりに設定します。

- [Name] に作品名を記入します。
- ② [Android] タブの [Project] 欄で、プロジェクトを選択します。
- ③ [Target] タブで [Manual] を選択します。
- ④ [Apply] を押してから [Close] を押します。

😂 Run Configurations	$\overline{\mathbf{X}}$
Create, manage, and run configu Android Application	rations
Image: Second Secon	New Configuration  Ardroid Tareet Common  Project android test  Lauch Action  Cauch Activity  Cauch:  Do Nothine
Filter matched 12 of 20 items	Apply
?	<u>B</u> un Close

いよいよ Android 端末をパソコンに USB 接続したら、[Run] → [RunConfigurations] と移動し、左画面にある作品名を選択します。そして [Run]を押し、接続している端末を選んで [OK] を押せば、端末にアプリが インストールされ、自動的に起動します。

2回目以降は設定の必要はなく、[Run]からすぐに起動させることができます。

#### ●キーストアファイルの作成

実機でのテストプレイが完了したら、Google Play などのアプリマーケット で配布するための apk ファイルの作成をします。

apkファイルの作成にはキーストアファイルが必要です。キーストアファイ ルとは、そのアプリを作ったのが確かに自分であることを証明するための、文 字通り鍵となるファイルです。

まずは Windows のスタートメニュー [ファイル名を指定して実行]から、「cmd」と入力してコマンドプロンプトを起動してください。「C:¥Documents and Settings¥ユーザー名」と表示されていますが、「cd C:¥」と入れてエンターを押すと、次の行が「C:¥」から始まるようになります。

この状態で、キーストアファイル作成のためのファイルを、パスを入力して 実行することになります。

"(JDK のインストール先)¥keytool" -genkey -keyalg RSA -keystore ADRkey.keystore -alias keyalias -validity 10000

これをあらかじめテキストファイルに入力しておき、コピー&ペーストして ください。筆者の場合、次のようになっています。

"Program Files¥Java¥jdk1.7.0\_03¥bin¥keytool" -genkey -keyalg RSA -keystore ADRkey.keystore -alias keyalias -validity 10000

以下、必要な項目を順に入力していきます。

- キーストアファイルのパスワードを入力します。入力しても画面には表示 されません。
- ② キーストアファイルのパスワードを再入力します。
- ③ 姓名を入力します。
- ④ 組織単位名を入力します。個人なら「none」とすればよいでしょう。
- 5 組織名を入力します。サークル名です。

- ⑥ 都市名または地域名を入力します。市区町村名です。
- ⑦ 州名または地方名を入力します。都道府県名です。
- ⑧ この単位に該当する2文字の国番号を入力します。日本を表すjpとします。
- 確認があるので「y」と入力します。
- ・鍵パスワードを入力します。①で入力したのとは別のパスワードになりますが、同じにしても構いません。その場合はエンターキーを押します。
- 鍵のパスワードを再入力します。



これで、C ドライブ直下に ADRkey.keystore というキーストアファイルが生 成されました。キーストアファイルは複数のアプリで使い回しができるので、 適当な場所に保管しておきましょう。

#### ●配布用の apk ファイルを作成する

いよいよキーストアファイルをアプリに組み込む作業をします。

- Package Explorer でプロジェクトを右クリックし、[Android Tools] → [Export Signed Application Package]を選択します。
- [Project] にプロジェクト名が入っていることを確認して [Next] を押し ます。
- ③ キーストア選択画面で[Use existing keystore]を選択し、[Location]に キーストアファイルの場所を、[Password]にキーストアファイルのパス ワードを入れて[Next]を押します。
- ④ キーエイリアス選択画面で [Use exising key] を選択し、[Alias] に [keyalias] を、[Password] にキーストアファイルの鍵のパスワードを 入れて [Next] を押します。
- ⑤ apk ファイルを出力する場所を指定して [Finish] を押します。

以上、長い説明でしたが配布用の apk ファイルが完成しました。従来型携帯 と比べるとなかなかややこしい手順ですが、慣れればどうということはないと 思います。

😂 Export	Android Application	- 🗆 🗙
Keystore	selection	0
💿 Use exi	sting keystore	
O Create	new keystore	
Location:	ents and Settings¥新井耕治¥workspace¥ADRkey.keystore	Browse
Password:	******	
Confirm		
?	< <u>B</u> ack <u>N</u> ext > <u>Einish</u>	Cancel

## 『コラム 2・WEB 版の Novel-Press』

Novel-Press はモバイル向けのノベルゲーム制作ツールですが、ウェブブラ ウザ向けも用意されています。以下、WEB版といいます。

WEB 版はモバイル版と比べて操作方法が異なり、一部機能が使用できませんが、WEB 版ならではのメリットもあります。

- ・組み立てがとても簡単
- 事実上、容量制限がない
- ・画面サイズが自由

このような特徴があるので、ほんのアクセサリー的な形で自分のサイトのど こかに設置すると、なかなか面白いのではないかと思います。

具体的な組み立て方法は、ここでは取り上げませんので、公式サイトをご覧 ください。なお開発者によれば「機能的に未熟なのでβ公開としていますが、 正式公開になることはたぶんないと思います」とのことです。

## 第6章

# Java ソースコードを書き換えよう

## 『より細かくシステム調節』

#### ●config ファイルでは不可能な部分の調整

各プラットフォームともアプリ組み立ての過程で、src フォルダに Java ソー スコードを入れました。このソースコードの一部を書き換えることで、config ファイルでは調整できない部分をいじることができます。

たとえばボタンのテキスト、メニュー画面やセーブ&ロード画面の文字列、 回想画面のレイアウトといったことを、細かく調整できます。

必要不可欠というほどのことではありませんが、たとえば和風の作品にカタ カナやアルファベットのテキストは似合いません。すべて漢字にすると、雰囲 気が出るでしょう。そういった細かな作り込みが、作品全体のクオリティを引 き上げることに役立つことは、間違いないはずです。

#### ●値を書き換えるだけ

プログラムの知識がない人にとっては、ソースコードを開くだけでもちょっ とした怖さがあります。かくいう筆者も、プログラムはほとんどわかりません。 しかしそんな人でも、ちょっと値を書き換える程度は可能です。どういうこと かというと、

#### XXXX = "YYYY";

ソースコードはこんな記述が大半を占めていますが、YYYY を別の値に書き 換えます。それだけで変更部分がアプリに反映されるのです。

Novel-Press のソースコードは自由に改造できるというのは、すでに述べた とおりです。その気になれば抜本的な改造も可能ですが、本章でまず気軽な書 き換えを覚えていきましょう。ちなみに OAP 版と SAP 版は、NovelPress.java 以外は同一のコードになっていますので、書き換え作業も同じになります。

# 『ボタンのテキストを変更する』

### ●もっとも簡単な書き換え箇所

従来型携帯では画面の左下と右下、Android 端末ならメニューキーを押すこ とで表示されるシステムボタンは、いずれもデフォルトのテキストが設定され ていますが、手始めにこれを書き換えてみましょう。

#### IAP 版の場合

IAP版の場合は NPCan.java ファイルを開き、下記の部分を検索で探してください。

public static final S	tring
LABEL_MENU	= "MENU",
LABEL_SAVE	= "栞",
LABEL_RECO	= "回想",
LABEL_TITLE	= "タイトル",
LABEL_BACK	= "戻る",
LABEL_END	= "終了",
LABEL_SITE	= "サイト",
LABEL_YES	= "YES",
LABEL_NO	= "NO";

これらの式の右側の値を、好きなように変更してください。

例として「終了」を「END」に、「サイト」を「WEB」にしてみます。非常 に簡単でしょう。



### ② OAP 版&SAP 版の場合

IAP 版と同じです。

### ③ ADR 版の場合

ADR 版は従来型携帯と違い、NovelPress.java のファイルを開きます。先頭からすぐのところに、下記の記述があります。

public class NovelPress extends Activity	{
//文字列定数	
protected static final String	
STR_MENULBL_END	= "終了",
STR_MENULBL_SITE	= "サイト",
STR_MENULBL_MENU	= "メニュー",
STR_MENULBL_BOOKMARK	= "栞",
STR_MENULBL_READBACK	= "回想",

STR_MENULBL_AUTOREAD	= "オート",
STR_MENULBL_TXTOFF	= "テキスト消去",
STR_MENULBL_TITLE	="タイトル",
STR_MENUTOP_CONFIG	= "設定",
STR_MENUTOP_CGLIST	= "CG-List",
STR_MENUTOP_EDLIST	= "ED-List",
STR_MENUTOP_BGMLIST	= "BGM-List",
STR_NUMIND_BOOKMARK	= "File.",
STR_NUMIND_CGLIST	= "No.",
STR_NUMIND_EDLIST	= "No.",
STR_NUMIND_BGMLIST	= "No.",
STR_OFFLBL_BOOKMARK	= "",
STR_OFFLBL_EDLIST	= "",
STR_OFFLBL_BGMLIST	= "",
STR_BTNLBL_BOOKMARK_SA	AVE = "SAVE",
STR_BTNLBL_BOOKMARK_L	OAD = "LOAD",
STR_BTNLBL_BGMLIST_PLAY	Y = " ♪ ";

ADR版では、あとで説明するメニュー画面などのテキストもここで変更できるようになっています。
# 『タイトル画面を変更する』

## ●画面のフェードインをなくす

タイトル画面も、変更できる箇所はいろいろあります。タイトル画面はデフ オルトではフェードインするようになっていますが、瞬間表示させることが可 能です。

#### IAP 版の場合

NPCan.java を開いてください。「private void scene\_title()」に含まれる下 記の部分を変更します。

```
//フェードイン
image = getImage(true,0);
if(image!=null) {
    for(int i=200; i>0; i-=5) {
        g.lock();
        g.clearRect(0,0,getWidth(),getHeight());
        g.drawImage(image,0,0);
        g.setColor(Graphics.getColorOfRGB(0,0,0,i));
        g.fillRect(0,0,getWidth(),getHeight());
        g.unlock(true);
        }
}
```

この中の「g.setColor(Graphics.getColorOfRGB(0,0,0,i));」が、フェードインの根幹となる透明度変化の記述です。徐々に透明度が下がっていき、やがて 完全に不透明になるという処理をしています。 この一連の処理を、まとめてスルーしてしまえます。「if(image!=null)」となっている部分を、次のように書き換えてしまうのです。

if(false)

これでフェードイン処理が起こらないようになります。

# ② OAP 版&SAP 版の場合

同じく「private void scene\_title()」ですが、かなり記述が異なっています。

```
//フェードイン
Image image = NPImage._createImage(true,0);
if(image==null) {
    int[] rgbs = new int[_getWidth()*_getHeight()];
    for(int i=0; i<rgbs.length; i++) rgbs[i] = this.bgColor;
    image = Image.createRGBImage(rgbs,_getWidth(),_getHeight(),false);
}else {
    long anc = System.currentTimeMillis();
    while(System.currentTimeMillis()<anc+1000) {
        drawImage(drawFrame(image,0,0,_getWidth(),_getHeight(),0,(int)(
256-256*(System.currentTimeMillis()-anc)/1000)),0,0);
        flushGraphics();
    }
}</pre>
```

フェードイン処理は「while(System.currentTimeMillis()<anc+1000)」の部 分です。次のように変更します。

while(System.currentTimeMillis()<anc)

## ③ ADR 版の場合

やはり「private void scene\_title()」に含まれる下記の部分を変更します。

```
//フェードイン
long timeAnc = System.currentTimeMillis();
while(true) {
    int alpha = 255-255*((int)(System.currentTimeMillis()-timeAnc))/1000;
    if(alpha<=0) break;
    //描画
    lock();
    fillRect(NovelPress._getColor(NovelPress._sysColor_back));
    if(image!=null) {
        drawImage(image,0,0);
    }
    fillRect(NovelPress._getColor(0x000000,alpha));
    unlock();
}</pre>
```

5行目の「if(alpha<=0) break;」以降をスルーするように書き換えます。

```
if(true) break;
```

# ●メニューのウィンドウ

config ファイルで設定した座標にメニューのウィンドウが表示されますが、 色や透明度を変えるにはソースコードをいじる必要があります。

# IAP 版の場合

「private void scene\_title()」に含まれる下記の部分を変更します。

//描画
<pre>setFont(this.titleMenuSize);</pre>
g.lock();
g.clearRect(0,0,getWidth(),getHeight());
if(image!=null) g.drawImage(image,0,0);
g.setColor(getColor(this.sysColor_cur,150));

変更部分は「g.setColor(getColor(this.sysColor\_cur,150));」です。メニュー のウィンドウは config ファイルで設定したシステムカラーが反映されるので すが、ここを直接書き換えることで、この部分のみ色と透明度を変更できます。

g.setColor(Graphics.getColorOfRGB(0,0,255,255));

第4章でも解説した RGB カラーと透明度です。こう記述すると、色は青で、 まったく不透明になります。4つめのパラメータは255を上限として、値が大 きいほど不透明になるのです。



image =

フェードイン処理のすぐ下に、次の記述があります。

.....

drawFrame(image,titleMenuX,titleMenuY,titleMenuW,titleMenuH,this.sys Color\_cur,150);

これがウィンドウの色と透明度を指定する部分です。やはり config ファイル で指定した色が反映されていますが、好みの色を RGB 値で指定します。

# image = drawFrame(image,titleMenuX,titleMenuY,titleMenuW,titleMenuH,0x0000f f,255);

0x0000ff という書き方は初めて出てきましたが、これも RGB 値の記述方法 です。16 進数という数字の数え方なのですが、詳細は検索などして調べてみて ください。これまでの書き方に直すと「0,0,255」、つまり青になります。

#### ③ ADR 版の場合

「private void scene\_title()」に含まれる下記の部分を変更します。

//描画

setFontSize(NovelPress.\_titleMenuSize);

lock();

fillRect(NovelPress.\_getColor(NovelPress.\_sysColor\_back));

if(image!=null) drawImage(image,0,0);

 $fill Rect (Novel Press\_title Menu X, Novel Press\_title Menu Y, title Menu W, title M$ 

MenuH,NovelPress.\_getColor(NovelPress.\_sysColor\_cur,150));

最後の行の「NovelPress.\_getColor(NovelPress.\_sysColor\_cur,150)」を、 OAP版、SAP版と同じ要領で指定します。

fillRect(NovelPress.\_titleMenuX,NovelPress.\_titleMenuY,titleMenuW,title MenuH,NovelPress.\_getColor(0x0000ff,255));

# ●メニューテキストのシャドウ

テキストにうっすらと描画されているシャドウを書き換えられます。

# ① IAP 版の場合

「private void scene\_title()」に含まれる下記の部分を変更します。

```
for(int i=0; i<titleMenu.length; i++) {</pre>
    if(titleMenu[i]!=null) {
        int sX = 0;
        if(titleMenuPos==0) sX = titleMenuX+titleMenuMgn;
        if(titleMenuPos==1) sX =
titleMenuX+(titleMenuW-stringWidth(titleMenu[i]))/2;
        if(titleMenuPos==2) sX =
titleMenuX+titleMenuW-titleMenuMgn-stringWidth(titleMenu[i]);
        int sY = titleMenuY+(titleMenuMgn+getFontH())*(i+1);
        drawString(titleMenu[i],sX+1,sY+1,0,shadeColor,0);
        if(i==com) {
             drawString(titleMenu[i].sX-1,sY-1,0,this.sysColor font,0);
        }else {
             drawString(titleMenu[i],sX,sY,0,fontColor2,0);
        }
    }
```

注目するのは「sX+1」「sY+1」「sX-1」「sY-1」となっている部分です。これ が通常のテキストと座標をずらしてシャドウを描画している処理になります。

```
drawString(titleMenu[i],sX,sY,0,shadeColor,0);
if(i==com) {
    drawString(titleMenu[i],sX,sY,0,this.sysColor_font,0);
}else {
    drawString(titleMenu[i],sX,sY,0,fontColor2,0);
```

}

このように+1や-1になっている部分を削除してしまえばよいのです。

#### ② OAP 版&SAP 版の場合

IAP版と同じです。

#### ③ ADR 版の場合

「private void scene\_title()」に含まれる下記の部分を変更します。

//描画

drawString(NovelPress\_\_titleMenu[i],sX+1,sY+1,NovelPress\_\_getColor(sha deColor));

if(onTouch) {

drawString(NovelPress.\_titleMenu[i],sX-1,sY-1,NovelPress.\_getColor(
NovelPress.\_sysColor\_font));

}else {

 $drawString (NovelPress\_titleMenu[i], sX, sY, NovelPress\_getColor(font Color2));$ 

}

こちらも「sX+1」「sY+1」「sX-1」「sY-1」となっている部分を書き換えます。

# ●終了する時のテキスト

終了ボタンを押した時の確認メッセージを変更することができます。

## IAP 版の場合

「private void scene\_title()」に含まれる下記の部分を変更します。

```
//実行
com = 0;
keyStateClear();
while(true) {
    int[] ks = getKeyState();
    if(ks[Display.KEY_SOFT1]==1) {
        if(confirm("終了しますか?")) {
        IApplication.getCurrentApp().terminate();
        }
```

これは文字列を書き換えるだけなので簡単です。



IAP 版と同じです。

# ③ ADR 版の場合

ADR 版は確認メッセージが出ないようになっていますので、変更はできません。

# 『メニュー画面を変更する』

# ●各メニュー名のテキスト

メニュー画面の「設定」「CG-List」などとなっているメニュー名を変更でき ます。これも NPCan.java を開いてください。

## IAP 版の場合

「private void scene\_menu(boolean fromTitle)」に含まれる下記の部分を書 き換えます。あまり長い文字列だと画面からはみ出るので注意してください。

```
private void scene_menu(boolean fromTitle) {
    int com = 0;
    String[] top = {"【設定】","【CG-List】","【ED-List】","【BGM-List】"};
    int[] topNo;
    int[] cX;
```

## ② OAP 版&SAP 版の場合

「private void pre\_init()」に含まれる下記の部分を変更します。

#### //menuTop

```
Vector vec = new Vector();
```

vec.addElement("【設定】");

if(this.modeCGList ) vec.addElement(" [CG-List] ");

if(this.modeEDList ) vec.addElement(" [ED-List] ");

if(this.modeBGMList) vec.addElement(" [BGM-List] ");

this.menuTop = new String[vec.size()];

vec.copyInto(this.menuTop);

IAP 版と記述が異なりますが、単純に書き換えるだけなのは変わりません。

## ③ ADR 版の場合

「ボタンのテキストを変更する」の項を参照してください。

#### ●設定画面のテキスト

設定画面のテキスト変更も、非常に簡単です。

#### IAP 版の場合

NPCan.java の「private void scene\_menu\_config(boolean directAcs)」に含 まれる下記の部分を変更します。

```
//設定画面
```

```
private void scene_menu_config(boolean directAcs) {
```

```
int i,j;
```

```
int com = 0;
```

```
int nooItem;
```

```
int volumeNo = -1;
```

```
String[][] top = {
```

```
{"文字の表示速度"."←遅い 速い→"}.
```

```
{"オートフ<sup>°</sup>レイのウェイト","←短い 長い→"},
{"文字枠の透明度","←不透明 透明→ "},
{"音量","←小さい 大きい→"},
{"ハ<sup>°</sup> イフ<sup>°</sup> レーション",null},
```

{"クリック待ちの表示".null}.

{"改頁待ちの表示",null}};

「null」というのは何も入っていないという意味なので、ここはいじらないで ください。

IAP版と同じです。

#### ③ ADR 版の場合

NPCan.java ではなく、AConfig.java を編集します。

//テキスト速度 if(NovelPress.\_ciTxtSpeed) { this.txtSpeedBase = createRow(); this.txtSpeedCom = createComView("文字の表示速度"); this.txtSpeedBar = createBarView(10,\_txtSpeed); this.txtSpeedInd = createIndView("速");

これはテキスト速度の例ですが、下のほうにオートウェイトなどの処理も記 述されていますので、同様に書き換えてください。

# ●タイトル画面に戻る時のテキスト

確認メッセージは、タイトル画面に戻る時にも出ます。終了する時と同じ要 領で書き換えてみましょう。

## ① IAP 版の場合

NPCan.java に次の処理が2箇所あるので、検索してください。

```
if(confirm("タイトル画面へ戻りますか?")) {
    gotoTitle = true;
    return;
}
```

IAP 版と同じです。

# ③ ADR 版の場合

ADR 版は確認メッセージが出ないようになっていますので、変更はできません。

# 『セーブ&ロード画面を変更する』

# ●セーブ&ロード画面名のテキスト

デフォルトでは「SAVE」「LOAD」となっています。ほとんどの場合はその ままで構わないと思いますが、「記憶する」「思い出す」などにすると、雰囲気 が増すことがあるかもしれません。

#### IAP 版の場合

NPCan.java の「private boolean scene\_record(boolean fromTitle)」に含 まれる下記の部分を変更します。

//描画

g.lock();

```
g.clearRect(0,0,getWidth(),getHeight());
```

int cW = stringWidth(" [SAVE] ");

```
int gap = (getWidth()-cW^{*}2)/3;
```

drawString(" [SAVE]

",gap,24-(24-getFontH())/2,0,save\_load?this.sysColor\_font:this.sysColor\_fon t2,0);

drawString(" [LOAD]

```
",gap*2+cW,24-(24-getFontH())/2,0,save_load?this.sysColor_font2:this.sysColor_font,0);
```

#### ② OAP 版&SAP 版の場合

「private boolean scene\_record(boolean fromTitle)」に含まれる下記の部分 を変更します。IAP版とやや記述が異なりますが、要領は同じです。 if(reDraw) {
 clearRect();
 drawMenuTop(new String[]{" [SAVE] "," [LOAD] "},save\_load?0:1);
 this.gra.setColor(this.sysColor\_cur);
 this.gra.fillRect(0,24+12+getFontH()\*com+1,\_getWidth(),getFontH()-2)
;

## ③ ADR 版の場合

「ボタンのテキストを変更する」の項を参照してください。

## ●セーブスロットのテキスト

デフォルトでは「File」となっているセーブスロットのテキストですが、これもまたゲームの雰囲気に合わせて書き換えたいものです。

#### IAP 版の場合

画面名のすぐ下に、以下の記述があります。「---」は、セーブデータがない 状態でのテキストです。「No Data」「データがありません」といった書き換え が考えられるでしょう。

```
for(int i=0; i<nooFile; i++) {</pre>
```

int cY = 24+12+getFontH()\*(i+1);

drawString("File." +

FUtil.\_formatNum((i+1),(nooFile>=10)?2:1),12,cY,1,this.sysColor\_font,this. sysColor\_font2);

if(time[i]==0L)

drawString("---",100,cY,0,this.sysColor\_font2,0);

【記憶する】	【思い出す】
記:1601	06/07 18:20
記憶02 記憶03 記憶04 記憶05 記憶06 記憶07 記憶08 記憶09 記憶10	記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記記

IAP版と同じです。

#### ③ ADR 版の場合

「ボタンのテキストを変更する」の項を参照してください。

# ●セーブ&ロードする時のテキスト

セーブ&ロードをする時の確認メッセージを書き換えられます。

IAP 版の場合

画面名のすぐ上に、以下の記述があります。初回セーブ時、上書き時、ロー ド時のそれぞれのテキストを変更できます。

また、セーブスロットの「File.」を書き換えていた場合、これも合わせて書 き換える必要があります。

```
if(save_load==false && time[com]==0L) continue;
String str = "File."+(com+1)+" ";
if(save_load) {
    if(time[com]==0L) str += "にセーブしますか?";
    else str += "に上書きしますか?";
    if(confirm(str)) {
        io_saveFile(com);
        break;
    }
}else {
    str += "をロードしますか?";
    if(confirm(str)) {
        io_loadFile(com);
        return true;
    }
}
```

IAP版と同じです。

#### ③ ADR 版の場合

ADR 版は確認メッセージが出ないようになっていますので、変更はできません。

# 『回想画面を変更する』

# ●回想テキストのサイズ

回想画面のテキストサイズは固定されていますが、自由な大きさにすること ができます。

#### IAP 版の場合

NPCan.java の「private void scene\_readback()」に含まれる下記の部分を 変更します。

#### //分割

setFont(12);

ここの数字をゲーム本編と合わせれば、しっくりくるでしょう。

#### ② OAP版&SAP版

IAP版とほとんど同じです。

//分割

setFontSize(12);

#### ③ ADR 版の場合

AReadback.java の 45 行目、「this.txt.setPadding(6,6,6,6);」となっていま す。この次に一行を追加します。

\_\_\_\_\_

this.txt.setTextSize(24);

# ●回想テキストの色

回想テキストの色は config ファイルの設定が反映されますが、これを変更することができます。

#### IAP 版の場合

NPCan.java の「private void scene\_readback()」に含まれる下記の部分を 変更します。

if(this.isVert) {

drawStringVert(lines[lineNo],getWidth()-(gap+getFontH()\*(index+1)),g
ap,0,this.sysColor\_font,0);

}else {

drawString(lines[lineNo],gap,gap+getFontH()\*(index+1),0,this.sysColo
r\_font,0);

書き換えるのは「this.sysColor\_font」です。好みの RGB 値にしてください。 ただし、あまり奇抜な色では読みにくくなるだけです。背景が黒なら、白か黄 色あたりがよいでしょう。

if(this.isVert) {

drawStringVert(lines[lineNo],getWidth()-(gap+getFontH()\*(index+1)),g
ap,0,0xffff00,0);

}else {

drawString(lines[lineNo],gap,gap+getFontH()\*(index+1),0,0xffff00,0);

ここで「if(this.isVert)」というのは「縦書きだったら」という意味なので、 横書き作品ならば、書き換えるのは下の方だけで構いません。



IAP 版と若干記述が違いますが、ほとんど同じです。縦書きがないので、1 つだけある「this.sysColor\_font」を書き換えてください。

if(lines[lineNo]!=null) {

drawString(lines[lineNo],gap,gap+getFontH()\*(index+1),0,this.sysColo
r\_font,0);
}

# ③ ADR 版の場合

ADR 版は AReadback.java の下記の部分を変更します。

this.txt.setTextColor(NovelPress.\_getColor(NovelPress.\_sysColor\_font));

ここの「NovelPress.\_sysColor\_font」を、やはり RGB 値に書き換えます。

# ●回想画面の余白を変更する

ゲーム本編と同様の余白で回想テキストを表示させたいことがあると思いま す。その場合もソースコードをいじりましょう。

#### IAP 版の場合

NPCan.java の「private void scene\_readback()」に含まれる下記の部分を 変更します。

```
private void scene_readback() {
    int startNo;
    int gap = 6;
    String[] lines;
```

3行目の「int gap」が回想画面の余白です。デフォルトでは上下左右が6ピ クセルで固定されています。この数字を書き換えるだけでOKです。

#### ② OAP 版&SAP 版の場合

IAP版と同じです。

#### ③ ADR 版の場合

ADR 版は AReadback.java の下記の部分を変更します。

this.txt.setPadding(6,6,6,6);

従来型携帯と違い、上下左右を個別に設定することができます。

# 『選択肢の表示方法を変更する』

# ●選択肢のフォーカス

選択肢を選ぶ時は、矩形のフォーカスが表示されます。この色を変えてみま しょう。

#### IAP 版の場合

NPCan.java の「private int scene\_select(String[] top,boolean[] log,int timer)」に含まれる下記の部分を変更します。

if(i==com) {
g.setColor(getColor(fileNow.fontColor,100));
//通常選択肢
if(timer<=0) {
g.fillRect(x-1,fileNow.txtareaY-1,getFontH()+2,maxW+2);
}

2行目の「fileNow.fontColor,100」が色と透明度です。

g.setColor(getColor(0xff0000,255));

これでまったく不透明な赤の矩形になりました。こうした色合いは、ホラー ゲームなどにいいかもしれません。

横書きの場合と縦書きの場合の2種類があるので、状況に応じてどちらかの み変更してください。



IAP 版と同じ要領で書き換えます。

```
//描画
if(timer>0 || reDraw) {
    draw();
    this.gra.setColor(curColor);
    //通常の選択肢の場合
    if(timer<=0) {
        int y = cY+(getFontH()+fileNow.fontSpaceH)*com-1;
        if(cX>fileNow.txtareaX) y += getFontH();
        this.gra.fillRect(fileNow.txtareaX-1,y,maxW+2,getFontH()+2);
    }
```

4行目の「this.gra.setColor(curColor);」が変更箇所になります。なお、OAP 版&SAP 版では透明度の変更はできません。

this.gra.setColor(0xff0000);

# ③ ADR 版の場合

こちらも NPCan.java の「private int scene\_select(String[] top,boolean[] log,int timer)」に含まれる下記の部分を変更します。

if(onFocus) {
<pre>int color = NovelPressgetColor(fileNow.fontColor,100);</pre>
//通常選択肢
if(timer<=0) {
fillRect(fileNow.txtareaX-1,y-1,maxW+2,this.fontSize+2,color);
}

書き換えるのは IAP 版と同じく「fileNow.fontColor,100」です。

# 第7章

# 作品をリリースする

# **『**ウェブサイトにアップロードする』

# ●集客力に自信があるなら

完成したアプリをいよいよアップロードしていくのですが、この方法にはお おまかに分けて2種類があります。ひとつは各社のアプリマーケットにアップ ロードすること、もうひとつは自分の運営するウェブサイトにアップロードす ることです。

自分のウェブサイトへのアップロードのほうが、はるかに簡単です。ただし、 自分のサイトのアクセス数が少なければダウンロードしてもらえないですし、 S!アプリは SoftBank の規約により、ダウンロードすること自体が不可能とな っています。

明確なメリットは、

- 18 禁作品のリリースが可能
- アプリマーケットに登録する際のわずらわしい手続きがない
- ③ 登録料などのコストがかからない
- ④ 差し替えが楽になる

これくらいでしょうか。逆に明確なデメリットは、ダウンロード数の把握が 困難なことです。その点、アプリマーケットならばダウンロード数も含めて、 様々な解析ができるようになっています。それと有料アプリのリリースが困難 であること。自分でシステムを構築すればできないこともないと思いますが、 そんないらない苦労をすることもないと思います。

ともあれ大前提として、アクセス数にある程度自信がある人でないと、厳し いのではないかというのが筆者の見解です。ちなみに筆者は、ほとんどをアプ リマーケットに登録しています。

# ●i アプリをアップロードする

プロジェクトフォルダの bin フォルダに jam ファイル、jar ファイル、そし て Download.html があります。Download.html をテキストファイルで開いて みましょう。

<html></html>
<head></head>
<title>Download Page</title>
<body></body>
<object <="" declare="" id="test" td=""></object>
data="test.jam"
type="application/x-jam">
<a href="notapplicable.html" ijam="#test">DOWNLOAD</a>

プロジェクト名が「test」の場合、このようになります。

ブラウザでアクセスすると、DOWNLOAD というテキストリンクのみがある 状態です。このままでもいいですが、多少デザインを調整したり、ゲーム紹介 ページに戻るリンクを追加してもよいでしょう。

FTP クライアントなどで、jam ファイル、jar ファイル、Download.htmlの3 つのファイルを同じ階層にアップロードすれば、それで作業は完了になります。

# ●オープンアプリ&EZ アプリ(J)をアップロードする

プロジェクトフォルダの bin フォルダに jad ファイル、jar ファイルがありま す。一方、Java ソースコードをダウンロードした際に、Download.xhtml と htaccess ファイルが同梱されていました。リリースに必要なファイルはこの 4 つです。

まず htaccess ファイルですが、これは一種の設定ファイルです。この htaccess ファイルは、契約しているレンタルサーバーによっては置くことがで きません。その場合リリースすることができないので、あらかじめ規約を確認 しておきましょう。

次に Download.xhtml を編集します。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<title>Download</title>
</head>
<body bgcolor="#ffffff" text="#000000">
<a href="device:jam?●●●">ダウンロード</a><br />
</body>
</html>
```

●●●には絶対パスの jad ファイルを指定します。絶対パスとは「http://~」 ではじまる URL です。もしファイルを『http://aaa/bbb/app/』の階層に置くな ら、jad ファイルまでの URL は『http://aaa/bbb/app/プロジェクト名.jad』と なります。編集を済ませたら、4 つのファイルをアップロードしましょう。

# ●Android アプリをアップロードする

Android アプリを自分のサイトにアップロードしている人はあまりいません。 なぜかというと、Google Play でリリースすることで得られるメリットが圧倒 的で、加えて Google Play ではないサイトに置かれている Android アプリは俗 に野良アプリと呼ばれ、信頼性といった面で不利だからです。どうしても野良 アプリとしてリリースしたいなら、このことを頭に入れておきましょう。

方法は簡単で、i アプリやオープンアプリ&EZ アプリ(J)と違い、リンクを設 定した html ファイルと一緒に apk ファイルをアップロードするだけです。

# 『プレイヤーにダウンロードしてもらう』

### ●QR コードを作成する

プレイヤーはモバイル用に最適化されたウェブサイトにアクセスするわけで すが、QRコードを読み取らせてアクセスしてもらうのが手っ取り早いです。

QR コードを作成できるサイトはいろいろありますが、筆者は「QR のススメ」 をいつも利用しています。

#### 【QRのススメ】

http://gr.quel.jp/

メニューの [作ってみよう!] → [URL] を選んでください。入力フォーム がありますが、URL だけを入力して先に進みましょう。

作成された QR コードが表示されるので、さっそく自分の携帯で読み取り、 ダウンロードしてください。従来型携帯ではこうやって実機でのテストを行い ます。QR コードはパソコンに保存しておけば、再度作成する手間を省けます。

# ●Android アプリをダウンロードする場合

野良アプリとして apk ファイルを自分のサイトでダウンロードしてもらう場合は、プレイヤーにも設定変更をしてもらわなければなりません。

Android 端末のアプリケーション設定で、「提供元不明のアプリ」という項目 があります。これにチェックを入れなければ、野良アプリをダウンロードする ことはできません。その旨をサイト上で説明しておきましょう。



# **『アプリゲットに登録する』**

# ●まずはクリエイター登録

アプリゲットはスパイシーソフト株式会社が運営する、携帯アプリの黎明期 から存在する最大手のサイトです。数えきれないほどのアプリが登録されてお り、筆者もお世話になっています。

たくさんのユーザーがいますので、自分のサイトで公開するのでないのなら、 ここに登録するのが選択肢のひとつです。何より、S!アプリを公開するのなら 事実上ここに登録するしかありません。

まずはクリエイター登録の手続きを行います。

#### 【アプリゲット クリエイター向けサイト】

http://creator.appget.com/

アクセスしたら[無料アプリ作家登録]とあるボタンをクリックしましょう。 クリエイター登録のためのメールアドレスを求められますので、フリーメー ル以外のメールアドレス(ただし GMail と Yahoo メールは可)を入力してく ださい。次にクリエイター情報を入力します。その他、任意でアンケート回答 などを求められますが、これだけで手続きは完了です。

## ●内部配信アプリの登録

アプリゲットで配信するアプリは、内部配信と外部配信の 2 つがあります。 内部配信とは、アプリゲットのサーバー内にアプリを置いて、直接ダウンロー ドしてもらうことです。

メニューの [アプリ配信] から [内部配信アプリを新規に公開する] に進ん でください。プラットフォームの選択をします。i アプリ、オープンアプリ、 S!アプリ、Android アプリの 4 種類を選べますが、Google Play があるので Android アプリをここで公開することはおすすめしません。



公開するファイルを選択し、作品情報を入力して、あとはアプリゲット側の 審査を待ちます。審査を通過したら、基本情報内に QR コードやダウンロード URL が表示されているので、自分のサイトに公開しましょう。

アプリ情報やファイルの更新、一時公開停止、削除も可能ですので、必要な 時に行ってください。

なお、EZ アプリ(J)については、残念ながら内部配信、外部配信ともに現在 未対応です。

#### ●外部配信アプリの登録

作品情報のみを登録して、後述する d マーケットや自分のサイト内に置いて あるアプリをダウンロードしてもらうよう、ユーザーを誘導するのが外部配信 です。

メニューの[アプリ配信]から[外部配信アプリを新規に公開する]に進ん でください。手順は内部配信とほぼ同じで、各プラットフォームごとに誘導先 URLを入力します。

なお、無料アプリを外部配信する場合は、誘導先は自身が運営されているサ イトでなければいけません。d マーケットなどを指定していても、受理されないのです。 誘導先URL(docomo)

docomo端末を誘導するURLを記載してください。(http://形式で入力して下さい。)

#### 誘導先URL(softbank)

softbankの誘導先のURLをご入力ください。(http://形式で入力して下さい。)

#### 誘導先URL(au)

auの誘導先のURLをご入力ください。(http://形式で入力して下さい。)

#### 誘導先URL(android)

アンドロイドアプリを配布するURLをご入力ください。(http://若しくは、market://から始まるURLを記載してください。)

# ●レジアプリの登録

有料の S!アプリをリリースしたい場合は、アプリゲットの兄弟サービス「ア プリレジ」に別途レジクリエイターとして登録する必要があります。このサー ビスでリリースするアプリをレジアプリと呼びます。

クリエイター向けサイトの [アプリの配信方法] ページから [アプリ★レジ とは?] に進んでください。規約を確認したら申込書をダウンロードして、ス パイシーソフトに郵送します。作者登録されたら、アプリレジ専用のサイトに アクセスしてログインしてください。

#### 【アプリゲット 作者サポートサイト】

http://ac-admin.appget.com/



このサイトで重要な機能が、アプリのホスティングです。SoftBank の規約 でS!アプリは一般サイトでの公開ができないのですが、ホスティング機能を使 えばここからダウンロードでき、実機でのプレイが可能になります。したがっ て無料アプリの場合でも、エミュレータでのテストプレイだけでは不十分と考 えるなら、レジクリエイター登録の必要があります。

まずは [S!アプリダウンロード通知設定] で、アプリアップロード時に URL を通知する SoftBank 携帯のメールアドレスを入力します。次に [S!アプリア ップロード] で jad ファイルと jar ファイルをアップロードします。この時の jad ファイルへのパスが、レジアプリ申請時のガイド URL になります。

あとはレジアプリ申請から、作品を登録してください。対応機種は 3GC 型 端末にチェックを入れます。現時点で不具合が発見されている「931SC」以外 の機種を対応としてください。

作品が受理されたら、クリエイター向けサイトで外部配信登録をします。[登録レジアプリ閲覧]のページに記載されている購入 URL を外部誘導の URL に設定し、登録してください。以上でレジアプリの公開手続きは終了です。

また、各作品ごとにユーザー向けのお知らせ、詳細情報を記載できる機能が あります。状況に応じて利用しましょう。

# 『d マーケット(i モード)に登録する』

## ●i アプリの公式マーケット

dマーケットは NTT docomo が運営する公式マーケットです。iモード向け とスマートフォン向けがあるのですが、個人制作者が利用できるのはiモード 向けのみです。利用料は無料となっており、とりあえずiアプリだけを制作す るという場合は、こちらが最有力候補になるでしょう。

#### 【マーケット管理システム】

https://www.market-box.jp/

最初に上記サイトよりアカウント登録してください。本人確認書類の提出を しなければなりませんので、一式を用意しておきましょう。

滞りなく手続きが済めば、ログイン情報が郵送されてきますので、それでロ グインしてください。

#### ●作品を登録する

ログインしたら[アプリ追加]のボタンからアプリの新規登録をします。コ ンテンツ名とはゲームのタイトルではなく、IDのようなものです。iappliTool でのプロジェクト名でよいでしょう。アプリ種別はもちろん Doja です。

ここでちょっと戸惑うのがアプリ基本情報です。この情報に従って、ADFを 追加で編集する必要があります。追加するのは下記の項目です。

PackageURL

TrustedAPID

PackageURL はそのままですが、TrustedAPID は DXID の欄にある数字を 入力してください。そしてビルドし直します。

180
#### アプリ詳細

公開ステータス	開発中
公開希望日時	
軽廃止予約日付	

#### ■ アブリ基本情報 DXID付与機能編集

※1度でもアプリを申請して承認されるとDXID付与	機能の変更はできなくなります。	■ アフリダウンロート
コンテンツ ID	1000008504	
コンテンツ名	badend	問登用に登録された端末のみアクセス可能
DXID	1000008504	※開発者端末として登録されたFOMAチップをいれた端末では
アプリ種別	Doja	dマーケット商用サイトの「アプリ詳細」ページをご覧頂くことが
ダウンロード用HTMLURL	http://appli.docomomarket.ne.jp/detail/BRW00 1_PM1_001?uid=NULLGWDOCOMO&guid=on&ci d=10000008504	でさません。 ご覧になる場合は、「開発者端末」の削除を実施するか、開発用に 登録されていないFOMAチップから商用サイトにアクセスして 下れ、」、
jamURL	http://appli.docomomarket.ne.jp/detail/BUY00 2_A1_001?uid=NULLGWDOCOMO&guid=on&cid =10000008504&urlRandomValue=h8D3lu3gwuV FL7dYI3UI	
PackageURL	http://appli.docomomarket.ne.jp/detail/BUY00 2_A2_001?uid=NULLGWDOCOMO&guid=on&cid =10000008504&ur/RandomValue=h8D3lu3gwuV FL7dYI3UI	

DXID 付与機能というのは、Novel-Press 作品では使うことはありません。 ここの設定はあまり深く考えなくても大丈夫です。

各情報の入力を終えたら、「バイナリ追加」から jam ファイルと jar ファイ ルをアップロードします。対応機種は「N704iµ」「N703iD」「N703iµ」を除 いたすべての DoJa-5.0 以上の端末にチェックを入れてください。

最後に[商用登録]にチェックを入れて、[コンテンツ公開申請]へと進みま す。公開日時を決め、チェック項目すべてをチェックし、[申請確認]→[確定] とクリックすれば申請が終了します。あとは審査結果のメールが来るのを待ち ましょう。

アプリを差し替えたり価格変更する場合も、同じ要領で申請し直すことにな ります。時折 d マーケット公式による 20%OFF キャンペーンなどが行われま すが、情報を編集しただけでストアに反映されるわけではありません。忘れず に再度の申請をしましょう。

# 『Ameroad に登録する』

## ●なんでも売れるソーシャル決済サービス

Ameroad は、個人がデジタルコンテンツをより売りやすくするためにと始ま ったサービスです。ソーシャル決済サービスと呼ばれ、Twitter のアカウント があれば、今すぐに始めることができます。

## [Ameroad]

http://ameroad.net/



アプリマーケットではなく Ameroad を利用する必然性は何かというと、オ ープンアプリまたは EZ アプリ(J)の作品を有料販売できる場だからです。i ア プリには d マーケットが、S!アプリにはアプリレジが、Android アプリには Google Play がありますが、オープンアプリと EZ アプリ(J)に関しては、自分 でシステムを作らない限りはありませんでした。それが、Ameroad で実現でき るのです。

## ●アップロードするのは QR コード

まずは Twitter アカウントでログインしてみましょう。そうしたら [売って みる] のページに進みます。

商品名、ファイル、価格、カテゴリ、商品説明、サムネイル画像を入力して いくのですが、ここでアップロードするのはアプリ本体ではなくアプリにアク セスするための QR コードです。つまりアプリ自体は、自分のウェブサイトに アップロードしておくのです。作品の登録が終わると、販売ページの URL が 表示されます。

ユーザーが QR コードを購入したら、それを au 携帯にデフォルトで付属し ているバーコードリーダーで読み込ませるのです。端末に保存した QR コード は「画像データを読込」から読み取ることができます。このとき、QR コード のファイル形式が jpg 以外だと、「画像データを読込」で読み取れない可能性が ありますので注意してください。

なお、作品説明のところで注意点があります。ひとつはその作品が EZ アプ リ(J)対応機種のみプレイできる場合、必ずその旨を記載しておくこと。もうひ とつは、販売するデータはアプリではなく QR コードであり、バーコードリー ダーから読み込ませてダウンロードする形式であることです。普通のアプリマ ーケットとは違うので、そうした説明はなるべく詳しく書いてください。参考 までに、筆者がリリースしているアプリのページを紹介します。

#### 【BAD END EZ アプリ(J)版】

http://ameroad.net/l/dIR

携帯からアクセスすると、次のページの画像のようになっています。

ところでこの方法では、QR コードの画像がネット上にばらまかれたら、タ ダでアプリが入手できてしまうことになります。そういうことがどうしても気 になってしまうなら、利用は避けたほうがよいでしょう。





by.<u>@araicreate</u>

【概要】 ホラーアドベンチャーゲーム auのEZアブリ(J)対応機種のみ ブレイできます。 対応機種:T006、T007、T008、 F001、CA007、S007、K009、URBAN O AFFARE

アブリにアクセスするためのQR コードを販売するという形になっ ています。携帯からのアクセスの 場合、QRコードを購入したら、バ ーコードリーダーの「画像データ を読込」からアクセスしてくださ い。\_

容量:958KB

## 『Google Play に登録する』

## ●Google 公式の Android アプリマーケット

Android アプリは基本的に、Google 公式である Google Play で公開すること になります。Google アカウントを持っていないと利用できませんので、持って いない人はまずアカウント作成をしてください。

#### 【Google アカウント】

https://accounts.google.com/

Google Play への登録は、初回に 25 ドルを支払う必要があります(支払い方 法はクレジットカードなので、持っていない方は利用できません)。SDK をダ ウンロードした Android Developers に再びアクセスしましょう。

#### [Android Developers]

http://developer.android.com/

手続きは[Publish]から行います。Google アカウントでサインインしてから、開発者情報を入力します。

ここでちょっとややこしいのが電話番号です。国番号を含めて入力しなけれ ばなりません。日本の電話番号の場合は先頭に「81」を付けます。さらに市外 局番の先頭に「0」が付く場合は、それを省いて入力します。

03-1234-5678という電話番号なら、次のようになります。

#### $+81 \cdot 3 \cdot 1234 \cdot 5678$

次にクレジットカード情報を入力して、決済画面で注文してください。

## ●アプリの登録

開発者登録が完了したら、以後は Google Play デベロッパーコンソールにて アプリの登録や情報の編集をすることになります。

#### 【Google Play デベロッパーコンソール】

https://play.google.com/apps/publish/

まずは[アプリケーションのアップロード]を選びます。apk ファイルを選 んでアップロードしましょう。

アプリケーションの編集

			公園保存
商品の詳細	APK ファイル		
アセットのアップロード			
<b>スクリーンショット</b> 2 個以上	スクリーンジョットを追加: 【 <b>ファイルを選択</b> 】 遅沢されていません	<u>(アップロード)</u>	スクリーンショット: 320x400、480x800、480x854、 1280x720、1280x300 24 ビット PNG または JPEG 形式(アルファ なし) フルブリード、枠なし スグリーンショットは横向ぎでアップロードで ぎます。サムネイルは回転して表示されます が、元の画像と向きは保持されます。
高解像度アブリケーション アイ コン 詳細	高解像度アブリケーション アイコンを追加: ファイルを選択) 遅択されていません	アップロード	高解像度アプリケーション アイコン: 512 × 612 32ビット PNG または JPEG 形式 上限: 1024 KB
<b>プロモーション画像</b> 省略可	ブロモーション画像を追加: 	アップロード	<b>プロモーション画像:</b> 検 180 x 縦 120 24 ビット PNG または JPEG 形式(アルファ ない) 画像は枠なし
宣伝用画像 省略可 詳細	宣伝用画像を追加: - ファイルを選択) 選択されていません	7970-6	<b>宣伝用画像:</b> 1024 x 500 24 ビット PNG または JPEG 形式(アルファ ない) ミニまたはマイクロにサイズダウン

次は作品情報の入力です。スクリーンショットや宣伝用画像なども必要なの で、あらかじめ用意しておきましょう。特に悩む箇所はないと思うので、必要 事項を入力したら[公開]をクリックしてください。

しばらくしたら、パソコンサイトと Android 端末の両方で、自分のアプリが Google Play に掲載されているのを確認できます。審査で待たされることがな いのが Google Play の素晴らしいところです。

# タグリファレンス

#### ■テキストタグ

## [c]

テキストを一時停止して、プレイヤーの操作を待ちます。

## [1]

テキストを改行します。

## [p] [P]

改ページをします。

[p](小文字)はプレイヤーの操作を待ってから改ページし、[P](大文字)はプレ イヤーの操作を待たずに改ページします。

## [txtpos]

文字の表示される区域を指定します。

パラメータ1

表示区域の始点の X 座標を指定します。

パラメータ2

表示区域の始点のY座標を指定します。

パラメータ3

表示区域の幅を指定します。

パラメータ4

表示区域の高さを指定します。

## [txtcolor]

文字の色を変更します。

一度変更すると、再び変更し直さない限り、その色のままになります。

#### パラメータ1

RGB カラーの R の値を、0~255 の範囲で指定します。

#### パラメータ 2

RGB カラーの G の値を、0~255 の範囲で指定します。

#### パラメータ3

RGB カラーの B の値を、0~255 の範囲で指定します。

## [txtsize]

文字の大きさを変更します。

一度変更すると、再び変更し直さない限り、その大きさのままになります。

#### パラメータ1

文字の大きさを 2~126 の範囲で指定します。

## [txtspeed]

文字の表示速度を変更します。

一度変更すると、再び変更し直さない限り、その速度のままになります。

#### パラメータ 1

文字の表示速度を 0~10 の範囲で指定します。

数字が大きいほど速くなり、10は瞬間表示になります。

## [txtspace]

文字間(文字と文字の間)と行間(行と行の間)の大きさを変更します。

一度変更すると、再び変更し直さない限り、その大きさのままになります。

#### パラメータ1

文字間を-127~128の範囲で指定します。

#### パラメータ2

行間を-127~128の範囲で指定します。

## ■グラフィックタグ

## [bi]

背景画像を表示します。

パラメータ1

表示する画像の番号を指定します。

## [bi,m]

背景を一色で塗りつぶします。

パラメータ1

RGB カラーの R の値を、0~255 の範囲で指定します。

パラメータ2

RGB カラーの G の値を、0~255 の範囲で指定します。

パラメータ3

RGB カラーの B の値を、0~255 の範囲で指定します。

## [fi]

前景画像を表示します。

パラメータ1

表示する画像の番号を指定します。

パラメータ2

表示する画像の場所を、1~10の範囲で指定します。

## [fioff]

前景画像を消去します。

パラメータ1

消去する画像の場所を、1~10の範囲で指定します。

省略した場合はすべての場所の画像が消去されます。

## [fixy]

前景画像の表示位置を変更します。

パラメータ1

位置を変更する画像の場所を、1~10の範囲で指定します。

パラメータ2

画像の x 座標を指定します。

パラメータ3

画像の y 座標を指定します。

[lock]

描画バッファリングを開始します。

## [unlock]

描画バッファリングを終了します。

[lock]タグとこのタグの間に[bi]などのグラフィックタグを挟んで使用します。

パラメータ1

画像を表示する時間をミリ秒単位で指定します。1秒なら1000です。

## パラメータ 2

効果オプションを指定します。「br」なら、右方向へのブラインド効果です。

f		フェードイン
b		ブラインド
s	効果タイプ	スクロール
р		ページめくり
q		クアドラングル
u		上方向へ
r	林田士白	右方向へ
d	%)未万问	下方向へ
1		左方向へ
t	画面切り替え中にテキストを描画する	

## [effect]

描画されているすべての背景画像・前景画像に対して、特殊効果を加えます。背景 か前景、個別には指定できません。

パラメータ1

効果を指定します。

mono	色調をモノクロにする
sepia	色調をセピアにする
crev	色調を反転する
none	特殊効果を解除する

[novbo9]
----------

メッセージウィンドウの表示区域を変更します。
パラメータ 1
表示区域の始点の X 座標を指定します。
パラメータ 2
表示区域の始点のY座標を指定します。
パラメータ3
表示区域の幅を指定します。
パラメータ 4
表示区域の高さを指定します。

## [boxcolor]

メッセージウィンドウの色を変更します。

#### パラメータ1

RGB カラーの R の値を、0~255 の範囲で指定します。

パラメータ2

RGB カフーの G の値を、0~28	55 の範囲で指定します。
---------------------	---------------

#### パラメータ3

RGB カラーの B の値を、0~255 の範囲で指定します。

## パラメータ 4

メッセージウィンドウの透明度を、1~10の範囲で指定します。

数字が大きいほど透明になります。このパラメータは省略可能です。

## [cg]

画像を CG リストに記録します。

パラメータ1

画像を記録するリストの番号を指定します。

パラメータ2

リストに記録する画像の番号を指定します。

## [shake]

画面を揺らします。

パラメータ1

揺らす強さをピクセル数で指定します。

パラメータ2

揺らす時間をミリ秒単位で指定します。

## [flash]

画面をフラッシュさせます。

パラメータ 1
フラッシュさせる時間をミリ秒単位で指定します。
パラメータ 2
RGB カラーの R の値を、0~255 の範囲で指定します。
パラメータ 3
RGB カラーの G の値を、0~255 の範囲で指定します。
パラメータ 4
RGB カラーの B の値を、0~255 の範囲で指定します。

## ■サウンドタグ

## [bgmon] [BGMON]

サウンドを再生します。

[bgmon] (小文字) はループ再生し、[BGMON] (大文字) は一度だけの再生となり ます。プラットフォーム、機種によって最大同時再生数が異なります。

## パラメータ1

再生するサウンドの番号を指定します。

## [bgmoff]

サウンドを停止します。

#### パラメータ1

停止するサウンドの番号を指定します。

省略した場合はすべてのサウンドが停止します。

## [bgm]

BGM リストにサウンドを記録します。

#### パラメータ1

サウンドを記録するリストの番号を指定します。

#### パラメータ 2

リストに記録するサウンドの番号を指定します。

#### パラメータ3

リストに表示する曲名を指定します。

## ■システムタグ

## [@]

[jump]タグや[select]タグの行き先となるラベルです。

[@ラベル名]のように記述します。

## [jump]

任意の場所にジャンプします。

#### パラメータ 1

ジャンプ先となるラベル名です。そのラベルが存在しなかった場合は無視されます。

## [select]

選択肢の表示に使用します。

パラメータ1と2を1セットとして、表示させる選択肢の内容と同じ数のセットを指 定します。

時間制限付き選択肢にする場合、最後の3つのパラメータを「timeout」「制限時間(ミリ秒単位)」「タイムアウト時に飛ぶラベル名」と指定します。

#### パラメータ1

表示させる選択肢の内容となる文字列です。

文字数に制限はありませんが、画面からはみ出ないように注意する必要があります。

#### パラメータ2

その選択肢を選んだ際にジャンプするラベル名です。

## [flagon]

フラグを on にします。

#### パラメータ 1

on にするフラグ番号を指定します。

1~128 まではセーブデータごとに保存されるファイルフラグ、

129~256まではゲーム全体に保存されるシステムフラグです。

## [flagoff]

フラグを off にします。

#### パラメータ1

offにするフラグ番号を指定します。他は[flagon]タグと同様です。

## [if]

指定したフラグ番号が on の時にのみ、指定したラベルヘジャンプさせます。

そのラベルが存在していなかった場合、無視されます。

#### パラメータ1

フラグ番号を指定します。このパラメータは複数指定でき、

[if,1,2,3,4,@ラベル名]

この場合は1から4がすべて on ならジャンプします。

#### パラメータ2

ジャンプするラベル名です。

## [z] [Z]

[z] (小文字) はゲームを終了してタイトル画面に戻し、同時にエンディングリストに 記録します。パラメータを省略した場合は、エンディングリストに記録せずにタイト ル画面に戻します。

[Z] (大文字) はエンディングリストに記録したあと、タイトル画面に戻さずにゲームを継続させます。

#### パラメータ1

エンディングを記録するリストの番号を指定します。

#### パラメータ2

リストに表示する文字列を指定します。

文字数制限はありませんが、画面からはみ出ないように注意する必要があります。

#### ■その他のタグ

## [wait] [WAIT]

テキストを一定時間、停止します。

[wait] (小文字) は、プレイヤーが読み進みの操作をすると、ウェイトが途中で解除 されます。[WAIT] (大文字) は、プレイヤーの操作にかかわらず指定した時間ウェ イトします。

#### パラメータ1

停止する時間をミリ秒で指定します。

## [vibon]

バイブレーションを開始します。

## [viboff]

バイブレーションを終了します。

このタグを使わないと、ずっと振動することになります。

## [input]

文字列入力画面を表示し、入力した内容を保存します。

全部で 20 個を保存できます。保存できる文字数は全角半角問わず 10 文字までで、

10 文字を超えて入力された場合、11 文字目以降は無視して保存されます。

#### パラメータ1

文字列を保存するスロット番号を、1~20の範囲で指定します。

#### パラメータ2

ここで指定した文字列が、スロット番号に保存されます。

途中でスロットの中身を変更したい場合や、入力画面でデフォルトの名前を入れてお く場合に使用します。

## [output]

[input]タグで保存した文字列を表示します。

パラメータ1

表示するスロット番号を、1~20の範囲で指定します。

## [saveon]

セーブ機能をオンにします。

## [saveoff]

セーブ機能をオフにします。

## [url]

ブラウザを起動します。

パラメータ1

接続する URL を指定します。

## 【著者略歴】

アライコウ

1981 年生まれ。大学卒業後、ライターとして活動開始。 パソコン系の雑誌への寄稿や、ゲームシナリオ執筆などを行っている。 著書に『XNovel でつくる iPhone ノベルゲーム』『もしものための iPhone サバ イバル術』がある。

## Novel-Press でつくるモバイルノベルゲーム

発行日: 2012 年 6 月 13 日 第 1 刷発行

著者:アライコウ

カバーイラスト:マヤマ

発売元:同人ゲーム制作支援 net http://doujingame-shien.net/

連絡先: info@doujingame-shien.net

本書の全部、または一部について、ウェブ上などに 無断コピー・転載することは固く禁じます。