

理想のマップエディタを求めて

文：古川 史幸

1 前置き

...困ったものです。

全然ゲームを作る気が起こらないのです。

でも、ツールを作る気は満々にあるのです。

というわけで、現在私が製作中のマップエディタ¹について、宣伝も兼ねているいろいろ語っていきたいと思います。

2 動機

ゲームプログラマーなら誰しも(?)が持つ感情の中に、「あ～、RPG 作りたいな～」という想いがあります²。しかし、実際にはRPGなんて一人で作れるもんじゃありませんし、一人でRPGを完成させられたならば、それは天才の成し得る技か人並みならぬ努力の結晶でしょう。そもそもRPGはやるが多すぎます。プログラム、キャラクタ、マップ、効果音、音楽、シナリオ、ストーリー、戦闘システム、...などなど、思いつくだけでこれだけあります。

さらに、データを用意するのもまた一苦労です。絵や音はともかく、マップやシナリオの記述などにテキストエディタやバイナリエディタを駆使することになるでしょう。しかし、バイナリエディタでの編集は何をやっているのかが一目でわからず、テキストエディタで編集されたデータは直接扱わず、何らかの変換を行って利用することがほとんどだと思われまふ。とすると、プログラマーはそのための変換ツールを—RPG 本体のプログラムとは別に—用意しなければなりません。そしてそのようなツールを作っているうちに、だんだんRPG本体のプログラムがおざなりになり結局制作を

きらめてしまったことがある人はプログラマーの数の4分の1ぐらいは居るでしょう³。

そこで私は考えました。

「そんなプログラマーの手助けになろう」と。

...とか書いてると、非常に聞こえがいいですが、どうも自分はあまりゲームを作るのに向かなくなっただんじゃないかなあと思ったのが本音でもあります⁴。それならツールでも作ってサークルの人の役に立てればいいかなあ、ツール作りをはじめたのです⁵。

3 目標

しかし、なぜ今更マップエディタなのか。

マップエディタと名のつくものなら既にフリーでいろいろ出回っているではないか。

...確かに、既にそのようなソフトは数多くあります。しかし、それらのマップエディタはいざ自分で使ってみようとする、

- フォーマットが固定されている
- マップサイズに制限がある
- 障害物を配置できない
- 描画速度が遅い
- 使い方が分かりにくい

等といった不満も少なからずあると思います。なかでもファイルフォーマットを自分で決められないのは、変換前後のデータファイル管理の点で圧倒的に不利になります。

¹ここで話に上る『マップエディタ』とは、初期ドラクエに代表されるRPGに使われる海とか地面とか城とかの並びを編集するツールと考えてください。『マップエディタ』という言葉には、HTMLのクリッカブル・マップを編集するツールとの意味もありますが、そちらではありません。

²もちろん自分でプログラムを組んでRPGを作りたいということです。RPG ツールに代表されるRPG コンストラクションツールを使う場合はここでは一切無視されます。

³4分の1の根拠は全くありませんが。

⁴ほら、去年だって開発途中のゲーム画面載せながら結局完成できなかったわけだし。

⁵ちなみに、ツールの実績は一応ありますよ。RPGのスク립トをバイナリに変換するツールを作りました。

そこで、私の作るマップエディタは既にあるようなマップエディタの模倣などではなく、「ファイルフォーマットを自由に決められるエディタにしよう」と考えたのです。もちろん、操作性も損なってはなりません。できれば、障害物の配置も行えるといいですね。そのようなことを考えて、私の作ろうとするマップエディタの目標はだいたい次のようになっています。

- 扱うファイルフォーマットを自由に決められるような仕組みを搭載する。
- 快適な動作のために必ず必要な、高速マップ表示を実現する。

現在一部の3年生を中心に、RPGを制作するプロジェクトが進んでいます。プロジェクトおよび制作ゲームについての説明は別ページを参照していただくとして、現在私の作っているマップエディタは最低でもそのプロジェクト向けにマップ編集ができるようになることが大きな目標です⁶。

4 構想

開発環境

今回は、VS.NET2003⁷でC#言語を用いることに決めました。C#言語を用いると、GUI周りの開発がVBライクに行えることと、オブジェクト指向を進化させたようなコンポーネント指向言語であり、C/C++言語以上に安全なプログラムが組めるものと判断したためです。もっとも、C#言語を用いるということは、実行環境に.NET Frameworkが必須になるというデメリットがありますが、現行のOSなら対応しているはずですのでそんなに大きな問題ではないでしょう。

⁶ま、ようするに、去年みたいに投げ出すことができないということですね。

⁷「Microsoft Visual Studio .NET 2003」の略。Windows プログラマにとっては有名すぎる開発環境ですね。なお、これをインストールするのに12GBのハードディスクでは足りず、秋葉原で40GBのハードディスクを購入して換装したのは、今でも正しいことだったと信じています（私のメインマシンはノートPCですよ）。

⁸ごめんなさい。良く調べてないのであまり正確ではないかもしれませんが、が、大きく外してはいないと思われます。

⁹Device Independent Bitmap：デバイス独立ビットマップ。出力装置に依存しないので、ピクセルベースの画像処理には向いています。ただし、画面への表示はあまり速くありません。

¹⁰Device Dependent Bitmap：デバイス依存ビットマップ。出力装置に依存するのでピクセル単位の細かな処理は出来ませんが、アクセラレーションが働く可能性がある矩形転送は非常に高速です。

¹¹私の環境（Pentium（700MHz）、192MB、Win2000+SP4）での感覚的な所要時間。この手の描画は遅くとも1秒以内で済まないと快適とは程遠いでしょう。ただ、これはGDI+のDrawImage関数が遅いのであって、DIB自体の転送がここまで遅いということではありません。

プラグイン

マップフォーマットを自由に決めるにはどうしたらよいか—その答えは「プラグイン」です。有名な画像ビューアにSusieというものがありますが、あれと似たような感じです。ただし、プラグインの作り方はおそらく大きく異なります。Susieプラグインはプログラムで決められた内部データ構造に合わせてデータを出力します⁸。私の作るマップエディタのプラグインは編集するデータ構造自体を記述します。オブジェクト指向が多少わかる人向けに書くと、「エディタで用意してあるマップファイルを表したクラスを派生して、独自のマップファイルを扱えるようなクラスを自分で作れ」ということになります。

このようなプラグインを実現するためには、マップエディタで扱うデータ構造を抽象化する必要があります。この抽象化は、出来る限り多くのファイルフォーマットに対応できるようにしないといけないので、多くの時間を費やして慎重に構造を決めました。

快適操作

快適操作のために絶対に必要なことは、マップ表示の高速化が重要な課題になります。そのために利用した技術として、「DIB⁹とDDB¹⁰の併用」および「マップイメージの再利用」をあげることが出来ます。

C#上（正確にはGDI+上）での画面表示は、基本的にDIBを用いて描画を行います。が、このDIB、描画速度が遅すぎて使い物にならないのです。32×32の小さなイメージでXGA(1024×768)の画面を埋めるのに平気で数秒から数十秒時間を取られます¹¹。これでは快適操作など考えられないですね。というわけで、次に思いつくのはDDBを用いた描画です。しかし、DDBは高速ですが、

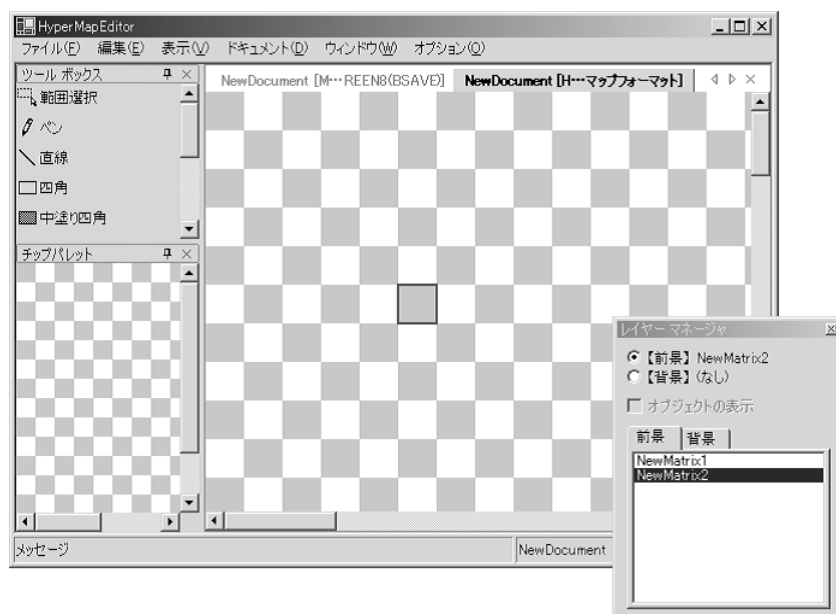


図 1: マップエディタの実行画面イメージ (画面は開発中のものです)

その代わり半透明処理などの複雑な画像効果を得ることが出来ません。複雑な画像効果を得るならばやはり DIB です。さあどうしましょう。

そういうわけで、画像効果は DIB で行い、画面表示は DDB を用いる方法を考えました。表示されるイメージに対して特殊な操作 (例えば少し暗くするなど) は DIB 上で行い、それを DDB に変換しておいていざ画面表示を行うときは DDB を用いればいいのです。ここで DIB → DDB への変換が行われていますが、これは元となる DIB に変更があったときのみ行われますので、パフォーマンスの点では問題ありません¹²。

複数レイヤーの重ねあわせ時に半透明処理が出来るると少しは便利になりそうですが、この際きっぱりとあきらめてしまいました¹³。その代わりとってはなんですが、単色カラーキーつき転送¹⁴は DDB でも速度を落とさず実現できますので、これを実装しました。

もう一つ、マップイメージの再利用については、まさにそのとおりで、マップをいったんバッファに描画しておき、画面に表示する時はバッファから切り出して表示する手法をとりました。そして、

画面がスクロールされると、バッファの再利用可能な部分をバッファ内で転送して余った部分にマップチップを描画するようにしました。このため、マップのスクロールも非常に高速になり、さらには画面のちらつきも隠すことが出来るようになりました。

5 現状

で、現状はこんな感じです (図 1)。とりあえずプラグインの管理、マップ表示、パレット表示、レイヤーの管理が大体出来るようになっていきます。まだマップチップをマップに置くという編集部分が未実装のままです。編集に関しては、アンドウや矩形転送の処理がネックになりそうです。

正直マップエディタにはあまり時間を取れなくなってきているので、この本が皆様の目に触れる時にはまだ完成してない可能性のほうが大きいです。それでも無事に完成するといいなあ...とこんなことを思ってこの文章をおしまいにします。最後までお読みいただきありがとうございます。

(2004/10/24 古川 史幸)

¹²メモリは 2 倍 (実際はマスク画像を別に保存しているので 3 倍) 必要ですが。

¹³これを「仕様」といいます (笑)。あ、バグみたいなのとは違うから、「制限」か。

¹⁴ある特定の色だけ転送しないということができる転送方法です。アニメのセル画のような効果が得られます。