

- [フィールドマップの仕組み](#)
 - [概要](#)
 - [全体マップ \(256*256\)](#)
 - [マップパターン \(4*4\)](#)
 - [マップパーツ \(1*1\)](#)
 - [屋根の仕組み](#)
 - [マップ書き換えのために](#)
-

フィールドマップの仕組み

概要

- マップは3段階に分かれて構成されている
 - 全体マップ (256*256): マップパターン(4*4)を64*64組み合わせることで構成
 - マップパターン (4*4): マップパーツ(1*1)を4*4組み合わせることで構成
 - マップパーツ (1*1): 左上・右上・左下・右下の画像の組み合わせで構成
- また、各パーツ毎に地形属性が設定されている
- たぶん3と6は同じ仕組みなので、参照アドレスは変わるだろうが、6への応用も可能と思われる。

全体マップ (256*256)

- 地上(便宜上命名)とアレフガルドに分かれる
 - 他に火山の火口付近、浅瀬の祠付近、虹の雫の橋付近など、フラグによって、上書きされる小マップがあるが、今回はオミット
 - 並び順は、左 右、上 下
 - 左上を1番目とすると、右上は64番目
 - 左下は4033番目、右下は4096番目
- \$7F0000-\$7F1FFFにマップパターン番号が展開される。
- マップパターンは、同じものが繰り返し使われる場合がある。
 - 特に全面海(地上の場合、マップパターン番号0x0001)に関しては、イヤと言うほど出てくる。が、固有の地形を書いている箇所は、ほとんど繰り返されない。だから、展開された情報はこんな感じになる。

例：上から6行目の展開結果

```
01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00
0D 00 0E 00 0F 00 10 00 11 00 12 00 13 00 14 00   ユーラシア
15 00 01 00 16 00 17 00 01 00 01 00 01 00 01 00
01 00 18 00 19 00 1A 00 1B 00 01 00 01 00 01 00
01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00
01 00 01 00 01 00 01 00 1C 00 1D 00 1D 00   アメリカ
1D 00 1E 00 1F 00 20 00 21 00 01 00 01 00 01 00
01 00 01 00 01 00 01 00 01 00 01 00 01 00 01 00
```

- マップパターンの集合は、圧縮された状態で格納されている
 - 地上は2D8A00-2D9BE2、アレフガルドは2D9BFF-2DA475に格納されている

- 圧縮データの展開方法

圧縮データを展開するワーク用のRAM領域(\$7FF3C6-\$7FF7C6, サイズ:\$0400)が存在する。展開前は全て0がセットされている。展開開始時のインデックスは\$03BEから開始し、\$0400までいくと\$0000に戻る。

マップデータは可変長だが、以下の構造の繰り返しで構成されている。

1byte目 データ構成情報 8ビットそれぞれが以降のデータの種類を規定しており、0ビット目が1番目、8ビット目が8番目の種類を表す。
0:圧縮データ(2バイト) 1:実データ(1バイト)

実データの場合は1バイトずつ読み、ワーク領域のindexの値をインクリメントする。2バイトごとにマップパターン領域(\$7F0000-\$71FFFF)にコピーしていく(このあたりの処理の順番はいい加減)。

圧縮データの場合は以下の処理を行う。

- 1)1byte目 + 2byte目 & 0xC0 <<2 展開するデータのindexを得る。
- 2)2byte目 & 0x3F + 3 から展開するbyte数を得る
- 3)1)のindexを開始点として、2)のbyte数だけ現在のindexの位置を開始点にしてデータをコピーする

わかりにくいと思うので地上マップデータ(\$2D8A00-\$2D9BE2)の1番目のレコードを例にとって説明する。

例1 : 01 01 BD FF FF FF 41 3F 83 3F C5 3F 07 7F 0F 44 という並びの時の解釈

- 1)1byte目(\$2D8A00)は データ構成情報 0x01(2進表記:00000001)なので、\$2D8A01のみが実データで\$2D8A02-\$2D8A0Fは7個の圧縮データが格納されていることがわかる。
 - 2)\$2D8A01は実データなのでワーク領域にコピーし、indexを進める(\$03BF)
 - 3)\$2D8A02-3は圧縮データ 展開先の開始index=\$03BD, 展開するbyte数は\$42byte
 - 4)index=\$03BDのデータはまだなにもセットされていないので\$00を\$03BFにコピーする 2種類のindexをととも1進め、展開するbyte数を1減らす(\$03C0, \$03BE, \$41)
 - 5)index=\$03BEのデータは2)でセットした\$01である。これを現在のindex=\$03C0にコピーし、4)と同様にindexを進め、残り展開byte数を減らす(\$03C1, \$3BBF, \$40)
- あとは繰り返し、インデックスが\$0400になったら\$0000にリセットする。

例2 F7,06,00,07,0F,78,08,00,09,00 という並びの時の解釈

F7 (命令 11101111 上位bitから実行される)

06 1番目の命令(データの読み込み)

- 00 2 番目の命令
- 07 3 番目の命令
- 0F 78 4 番目の命令 (繰り返しに関する何か)
- 08 5 番目の命令
- 00 6 番目の命令
- 09 7 番目の命令
- 00 8 番目の命令

圧縮を考えなければ (要はbit1の命令しか使わなければ)、マップパターンの集合は容易に書き換えることができる。

- 例えば、上に挙げた例 2 の部分を無圧縮で書こうとすれば、

```
FF,01,00,01,00,01,00,01,00,
FF,01,00,01,00,01,00,01,00,
FF,0D,00,0E,00,0F,00,10,00,
FF,11,00,12,00,13,00,14,00, (以下略)
とすればよい。
```

(参考)

000541D8-00054231 フィールドマップデータ情報 (9byte*10rec)

- 0: フィールドマップパターン数・横
 - 1: フィールドマップパターン数・縦
 - 2: フィールドマップパターン表示開始位置・横
 - 3: フィールドマップパターン表示開始位置・縦
 - 4: マップパターン番号 (下位8bit)
 - 5: マップパターン番号 (上位2bit)
 - 6-8: フィールドマップ情報格納アドレス
(マップパターンの集合が書かれたアドレス)
- エントリの 1 番目が地上、6 番目がアレフガルド (他は小マップ)

マップパターン (4*4)

- マップパーツの組み合わせで構成される

(マップパーツは、1 タイルの画像と地形情報)

- 並び順は、左 右、上 下

1 番目, 2 番目, 3 番目, 4 番目,
5 番目, 6 番目, 7 番目, 8 番目,
9 番目, 10 番目, 11 番目, 12 番目,
13 番目, 14 番目, 15 番目, 16 番目,
という感じになる

- マップパターンにおける、マップパーツ番号の情報は分散している

1 番目:2DA49C, 2 番目:2DAED9, 3 番目:2DB916, 4 番目:2DC353
5 番目:2DCD90, 6 番目:2DD7CD, 7 番目:2DE20A, 8 番目:2DEC47

9番目:2DF684, 10番目:2E00C1, 11番目:2E0AFE, 12番目:2E153B
13番目:2E1F78, 14番目:2E29B5, 15番目:2E33F2, 16番目:2E3E2F
これにマップパターン番号を加算したアドレスから読み出す

- マップを改造するにはここがネック。この格納形態では、情報が把握しづらいので書き換えが大変だし、何より空きエントリがない。プログラムを書き換えて参照箇所を変えること（しかしこれを参照している箇所は17箇所もある）、ある程度のエリアの参照領域を確保すること、が必要と思われる
- 参照しているマップパーツは、地上とアレフガルドで異なる
 - 地上はマップパーツ群情報0x04
 - アレフガルドはマップパーツ群情報0x05
- で指定されるマップパーツを用いる
- 全体マップにおいてマップパターン番号を展開する際、アレフガルドは0x0703を加算して展開されている。すなわち、ROM上のマップパターンの集合において、アレフガルドで0x0001が指定されていた場合は、参照するマップパターン番号は0x0704となる

(参考)

00054232-00054258 フィールドマップパターン情報 (4byte*10rec)

(マップパターン番号順)

0: マップパーツ群情報 (下位8bit)

1: 2b: マップパーツ群情報 (上位2bit)

6b: マップパターン開始エントリ (下位6bit)

2: マップパターン開始エントリ (中位8bit)

3: 2b: マップパターン開始エントリ (上位2bit)

マップパターン開始エントリを展開したフィールドマップ情報に加算して設定
(アレフガルドは0x0703を加算)

マップパーツ (1*1)

- マップパーツへのアクセスは、先に参考資料を出した方がわかりやすいか

(参考)

00051450-0005184F マップパーツ群情報(8byte*128rec)

0: グラフィック読み込み情報番号 (下位8bit)

1: 2b: グラフィック読み込み情報番号 (上位2bit)

6b: マップパーツ開始エントリ (下位6bit)

2: マップパーツ開始エントリ (中位8bit)

3: 2b: マップパーツ開始エントリ (上位2bit)

6b: マップパーツ数 (下位6bit)

4: 4b: マップパーツ数 (上位4bit)

5-7: 地形情報の参照先アドレス (24bit) ->マップパーツと対応して並んでいる

002545A0-002838D7 マップパーツグラフィック (8byte*24167rec)

1 マスの画像 8byteで構成 (2byte*4)

0-1: グラフィック左上

2-3: グラフィック右上

4-5: グラフィック左下

6-7: グラフィック右下

地上はエントリ0x0173-0x025F(237record)

アレフガルドはエントリ0x0260-0x030F(176record)を用いている

- 先述したように、地上はマップパーツ群情報0x04、アレフガルドは0x05

0x04 グラフィック読み込み情報番号:0x00F

マップパーツ開始エントリ :0x0173

マップパーツ数 :0x0ED

地形情報の参照先アドレス :\$E836D8 ->ROM上2836D8

0x05 グラフィック読み込み情報番号:0x010

マップパーツ開始エントリ :0x0260

マップパーツ数 :0x0B0

地形情報の参照先アドレス :\$E837C5 ->ROM上2837C5

- マップパーツ番号 + マップパーツ開始エントリの位置にある<マップパーツグラフィック>を取得する

- 実際のグラフィックやパレットの取得は、さらにグラフィック読み込み情報番号を用いて行っている(この辺はもうキャラクターグラフィックの取得などと同じルーチン)が、そこまでは書ききれない

(一応参考)

000510A6-0005144F グラフィック読み込み情報 (7byte*134rec)

0-1: グラフィック読み込み開始エントリ

2: グラフィック読み込み数 (下位8bit)

3: 2b: グラフィック読み込み数 (上位2bit)

6b: 未使用

4-6: グラフィックの上位4bitとして付加する情報のアドレス
(ラーミア飛行時に使用している)

エントリ0x02: キャラクターグラフィック 通常

エントリ0x03: キャラクターグラフィック 棺桶

エントリ0x04: キャラクターグラフィック 特殊

エントリ0x0B: キャラクターグラフィック 変化の杖

エントリ0x0F: フィールド・地上

エントリ0x10: フィールド・アレフガルド

(参考ここまで)

- 地形情報は1byteの情報で、マップパーツ番号に対応して並んでいる

0: 2b: 不明

2b: 0:不明 1:通常 2:海 3:侵入不可

4b: 0:草原 2:砂漠 3:森林 4:毒沼 5:山地 6:海 7:岩山 9:施設 A:氷原

(この情報はエンカウント率や、戦闘背景の情報を取得する際に使われている)

- マップパーツ内には、街への移動などの条件は書かれていない。
- 街への移動や、イベントが発生する座標などの情報は、別途定義されている

屋根の仕組み

マップ書き換えのために

- ダンジョンにはよく先が見えなくなっている部分があり、そこへ行くと表示されているマップ部分が切り替わってさらにその瞬間に戦闘に突入しやすいのはおなじみである。もし屋根があるマップを書き換えたいなら屋根のデータも変更する必要があるが、この屋根は固定長データで管理されていない(先発のDQ6では固定長データである)。
- 屋根を配置しているのは各フロアでの以下のサブルーチンである。

22 AD 91 C6 aa bb cc dd ee:屋根を配置する

(aa=屋根の開始X座標, bb=屋根の横幅, cc=屋根の開始Y座標, dd=屋根の縦幅, ee=屋根の番号)

22 34 92 C6 aa bb cc dd ee:屋根切り替え地点を配置する

(aa=通行, bb=左端X座標, cc=Y座標, dd=右端X座標, ee=消す屋根の番号)

22 8E 92 C6 aa bb cc dd ee:屋根切り替え地点を配置する

(aa=通行, bb=X座標, cc=上端Y座標, dd=下端Y座標, ee=消す屋根の番号)

- フロアでこれらの処理をしている場所を探すには、アドレス 00804A3-00808CE からのフロアごとの配置処理をたどっていくか、フロアに入るときに上の3つのサブルーチンの位置にブレークポイントをしかけてかかったら step out を1回押すかのどちらかをすればよい。
- もし階段などの移動地点に屋根をかぶせたときは、各移動データの4bit長の部分が屋根番号になっているのでその値も変更しておこう。