

別紙 原告主張

本件パッチに創作性が認められることを明らかにするため、本件色切替パッチに焦点を当て、その条件設定や動作設定を説明することでその創作性を述べることにする。

(1) 部分①

```
$82:95E0 B9 00 30    LDA $3000, y[$40:3200]
```

メモリ[\$3000, y]に入っているモビルスーツの登場機体に付された番号を、CPUに呼び出す指令である。

(2) 部分②

```
$80:0000 C9 E7 00    CMP #$00E7
$80:0000 D0 07      BNE $07
$80:0000 A9 50 96    LDA #$9650
$80:0000 5C CC F1 02 JMP $02:F1CC
karasu
$80:0000 C9 9C 00    CMP #$009C
$80:0000 D0 04      BNE $04
$80:0000 5C E0 E9 25 JMP $25:E9E0
```

当該部分は、登場機体を角システムとは別のプログラム（F 9 1 というモビルスーツの分身に関するもの）に移行させるかどうかのオンオフにかかるフラグ（コンピュータにおいて、処理結果を真（True）/偽（False）いずれかの値で保持するレジスタ（コンピュータの記憶回路）または変数のこと。甲 3 5）であり、角システムを構成するものではないため、説明は省略する。

(3) 部分③

```
$80:0000 AA      TAX      ①
$80:0000 BF 1A FF 31 LDA $31FF1A, x[$31:FF**] ②
$80:0000 29 FF 00  AND #$00FF ③
$80:0000 D0 06    BNE $06
*0 Flag*
$80:0000 8A      TXA
$80:0000 3A      DEC A
$80:0000 5C C3 F1 02 JMP $02:F1C3 ④
```

まず、登場機体番号をCPUの「X」（Xレジスタ）というテーブルに移動させる指令をする（①）。

※本件ゲームの対応ハードウェアであるスーパーファミコンのCPUには「A」「X」「Y」という3つのテーブルと、「スタック」というメモ領域が存在するところ、パッチプログラムにおいては、「A」「X」「Y」にメモ領域から数値を出し入れして計算をしてプログラムを処理させるのである。

その上で、原告が作成した角システムデータベースのシート1（甲36の1）の登場機体番号（シート1のA列）のデータを「A」のテーブルに呼び出す指令をする（②）。呼び出した登場機体番号の「pフラグ」（角システムデータベースのシート1のC列）が「00」の場合には、「0 Flag」以下の行に移行させ（③）、バンダイプログラムに戻す旨の指令を与える（＝角プログラムを終了させる）（④）。

一方、「pフラグ」が「00」以外の場合には、「Flag」以下の行（下記（4））に移行させる（③）。

(4) 部分④

Flag

\$80:0000 3A	DEC	
\$80:0000 85 B6	STA \$B6	①
\$80:0000 84 B8	STY \$B8	
\$80:0000 68	PLA	②
\$80:0000 7A	PLY	
\$80:0000 FA	PLX	
\$80:0000 DA	PHX	
\$80:0000 5A	PHY	
\$80:0000 48	PHA	

ここでは、まず、「A」や「Y」の各テーブルについて、[\$00:30B6] [\$00:30B8] という空きメモリ容量に保存する旨指令している (①)。データを出し入れして処理するメモリ領域は、バンダイプログラムの初期設定のままでも実行は可能である。ただ、原告プログラムにより追加される処理もその場で行うとなると、コーディング (記述方法) も複雑となり、負荷も大きくなるため、動作が遅くなったりエラーが起きる等、動作が不安定となる恐れがある。その点、原告プログラムではそれを [\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更することにより、以降のコーディングを最小限とすることができ、負荷も小さくなった。この創意工夫によるコーディングの結果、原告プログラム適用時の処理の高速化と動作の安定性の確保を実現することができたものである。

また、ここでは、CPUのメモ領域(スタック)に記録されている値を「A」「X」「Y」という各テーブルに読み出した上で、再びメモ領域に戻すという指令を与えている(②)。

すなわち、「PL」は「A」「X」「Y」と組合せて用いるところ、メモ領域に記録されている値をCPUの「A」「X」「Y」に呼び出す旨の指令である。また、「PH」は「A」「X」「Y」と組合せて用いるところ、「A」「X」「Y」のテーブルの値をメモ領域に戻す旨の指令である。ここで出し入れする値は、バンダイプログラムの値であり、具体的には各登場機体に紐づく1～7のユニット番号である(本件ゲームのバトルにおいては最大7機体が参加できる)。以降の原告プログラムを適用していくに先立ち、これを一旦呼び出した上で指定したメモリ領域内に保存(後記(5))しておくことで、バンダイプログラムにおける登場機体との紐付けがなされ、以降の原告プログラムの適用による色の切り替えが、バンダイプログラム上(ゲーム画面上)に表れることになる。

一旦呼び出した値を再びメモ領域に戻しているのは、原告プログラムがあくまでパッチプログラムであり、それへの適用はいわば「う回路」であるため、以降の処理の流れを本流(バンダイプログラム)に戻す必要があるからである。ただ、出し入れの順番について様々な組み方がある中で、原告が「A→Y→X→X→Y→A」の順で組んでいるのは(4～9行目)、この順序によることで出し入れ前後でバンダイプログラムのメモ領域の値が変わってしまうことを回避でき、バンダイプログラムに悪影響を及ぼしてバグが生じてしまうことを回避することができるかと原告が判断したことによる(原告は幾度となく失敗しながらバグ解消の手法に行きついた)。係る部分は、原告が、試行錯誤の末バグを回避するために記述したものである。

(5) 部分⑤

\$80:0000 A4 B8	LDY \$B8
\$80:0000 86 B8	STX \$B8
\$80:0000 A5 B6	LDA \$B6

部分④で取り出したバンドイプログラムの値を，[\$00:30B6] [\$00:30B8] という空きメモリ容量に保存したり（ST），出したり（LD）する指令である（前記（4）における「A」と「Y」と同様，ここで「X」についても [\$00:30B6] [\$00:30B8] という空きメモリ容量に保存する旨指令している）。

前記（4）と同様，データを出し入れして処理するメモリ領域は，バンドイプログラムの初期設定のままでも実行は可能であるが，原告プログラムではそれを [\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更することにより，以降のコーディングを最小限とすることができ，負荷も小さくなった。

また，ここでは，後述の「条件a」「条件o1」「条件o2」や，角システムデータベースのシート2（甲36の2）の「paletteアドレス」（T列～AA列まで。以下「パレットアドレス」という）で使用するために保存していた「X」や「A」のテーブルの値を「LD」で呼び出して準備（一部の処理。下ごしらえのようなもの）している。

各テーブル上での本格的な処理に先立ち，この段階で一部の処理（下ごしらえ）を先行して行っておくことにより，後の本格的な処理の際に負荷を抑えることが可能となった。

これらの創意工夫によるコーディングの結果，原告プログラム適用時の処理の高速化と動作の安定性の確保を実現することができたものである。

(6) 部分⑥

```
$80:0000 0A      ASL ①  
$80:0000 0A      ASL  
$80:0000 0A      ASL  
$80:0000 48      PHA ②      ;p dataMatrix y
```

部分⑤最終行の「LDA \$B6」と組み合わせて意味を有する部分であるが、パレットアドレスは1機体につき8個の数値を使用するため（角システムデータベースのシート2のT列～AA列（甲36の2））、「A」テーブルの値を8倍している（①。SLは「倍」という意味であり、倍×倍×倍＝8倍）。そして、そのように計算した「A」テーブルの値（パレットアドレスを呼び出す値）をメモ領域（スタック）に記録する指令を与えている（②）。

なお、前記（4）及び（5）のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。

(7) 部分⑦

```
$80:0000 A5 B6    LDA $B6 ①  
$80:0000 0A      ASL  
$80:0000 18      CLC  
$80:0000 65 B6    ADC $B6  
$80:0000 0A      ASL  
$80:0000 0A      ASL  
$89:F795 48      PHA ②      ;p 属性Matrix dataMatrix y
```

角システムデータベースのシート 2 (甲 3 6 の 2) 記載の「所属」と「レベル」は、1 機体につき 1 2 個の数値を使用するため (F 列～Q 列)、「A」テーブルの値を 1 2 倍している (①)。そして、そのように計算した「A」テーブルの値 (所属とレベルを呼び出す値) をメモ領域 (スタック) に記録する指令を与えている (②)。

なお、前記 (4) 及び (5) のとおり原告プログラムでは各テーブルを [\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。

(8) 部分⑧

```

$80:0000 B9 1A 30 LDA $301A, y[$40:3*1A] ①
$80:0000 29 FF 00 AND #$00FF ②
$80:0000 0A ASL A
$80:0000 AA TAX
$80:0000 BF 0F 60 40 LDA $40600F, x[$40:600F] ③
$80:0000 85 B6 STA $B6 [$00:30B6] ;*player 所属
④

```

「301A」というメモリにはプレイヤーナンバーが記録されているところ、この数値を「A」テーブルに移す (①)。そして、一定の計算を行った上で、「X」テーブルに移す (②)。次いで、「所属」のパラメータを記録しているメモリ「\$40600F, x」を「A」テーブルに呼び出す (③)。そして、「A」テーブルから空きメモリ「B6」に記録をする (=「所属に関する変数」の登録) (④)。

※所属に関する変数は、後述の「条件 a」「条件o1」「条件o2」での突
合作業の際に用いることとなる

なお、前記（4）及び（5）のとおり原告プログラムでは各テーブルを [\$0
0:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したこと
により、ここでのコーディングもシンプルな記述とすることが可能となった
ものである。

（9）部分⑨

```
$80:0000 A6 B8      LDX $B8      ①  
$80:0000 BF 9F 34 00 LDA $00349F, x ②  
$80:0000 85 B8      STA $B8      [$00:30B8] ; unit Lv  
                    ③
```

部分⑤の「STX \$B8」でバンダイプログラムから取り出して記録した空きメ
モリ容量「B8」の中のユニット番号の値（1～7）を「X」に移し（①）、
メモリ容量「00349F」に記録されている登場機体の「レベル」のパラメータ
を「A」テーブルに呼び出し（②）、「A」テーブルから空きメモリ容量「B
8」に記録する（＝「レベルに関する変数」の登録）（③）。

※レベルに関する変数は、後述の「条件 a」「条件o1」「条件o2」での
突合作業の際に用いることとなる

なお、前記（4）及び（5）のとおり原告プログラムでは各テーブルを [\$0
0:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したこと
により、ここでのコーディングもシンプルな記述とすることが可能となった
ものである。

(10) 部分⑩

```
$8A:FE4B FA          PLX          ①          ;p dataMatrix y  
*条件a**  
$80:0000 BF 51 E5 25 LDA $25E551, x ②  
$80:0000 C5 B6       CMP $B6      ③  
$80:0000 D0 16       BNE $16      ;条件o1 ④
```

原告は、色の切り替えを実行する際の条件として「条件 a」「条件o1」「条件o2」の3本立て構成を採用したところ、この部分は「条件 a」についての記述である。

まず、⑦の部分でメモ領域に記録していた「条件」を呼び出す所属とレベルに関する値を「X」テーブルに呼び出す指令を発する(①)。その上で、「\$25E551」(=角システムデータベースのシート2における「所属」(F・Gの列)データを意味する)を「A」テーブルに呼び出し(②)、部分⑧で定義したモバイルスーツの所属に関する変数と突合する(③)。そして、数値が合致しなければコンピュータ処理を「条件o1」(後記(13))に移行させ、合致すれば次の行(下記(11))へ移行させる(④)。

なお、前記(4)及び(5)のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。

(11) 部分⑪

```
$80:0000 BF 53 E5 25 LDA $25E553, x ①
$80:0000 29 FF 00   AND #$00FF
$80:0000 C5 B8     CMP $B8 ②
$80:0000 F0 4B     BEQ $4B      ;DATA1 ③
$80:0000 90 49     BCC $49      ;DATA1
```

「\$25E553」（=角システムデータベースのシート2における「レベル」（H列）データを意味する）を「A」テーブルに呼び出し（①），部分⑨で定義したモビルスーツのレベルに関する変数と突合する（②）。そして，モビルスーツのレベルが，「レベル」に関するデータベースで定められた値よりも大きいか，または同等であれば（すなわち，BEQとBCCの両方で「以上」の意味を有する），コンピュータ処理をDATA1（後記（19））に移行させ，未満であれば次の行（下記（12））へ移行させる（③）。

なお，前記（4）及び（5）のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより，ここでのコーディングもシンプルな記述とすることが可能となったものである。

(12) 部分⑫

```
$80:0000 BF 54 E5 25 LDA $25E554, x ①
$80:0000 29 FF 00   AND #$00FF
$80:0000 D0 51     BNE $51      ;DATA4 ②
```

「\$25E554」（＝角システムデータベースのシート2における「f l a g」（I列）データを意味する）を「A」テーブルに呼び出し（①），当該データベースにおける f l a g の値が「0 1」であれば，コンピュータ処理をDATA4に移行させ，「0 1」でなければ次の行（下記（1 3））へ移行させる。

なお，前記（4）及び（5）のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより，ここでのコーディングもシンプルな記述とすることが可能となったものである。

(13) 部分⑬

条件o1

```
$80:0000 BF 55 E5 25 LDA $25E555, x ①  
$80:0000 25 B6 AND $B6 ②  
$80:0000 F0 16 BEQ $16 ;条件o2 ③
```

原告は，色の切り替えを実行する際の条件として「条件a」「条件o1」「条件o2」の3本立て構成を採用したところ，この部分は「条件o1」についての記述である。

まず，「\$25E555」（＝角システムデータベースのシート2における「所属」（J・K列）データを意味する）を「A」テーブルに呼び出し（①），部分⑧で定義したモビルスーツの所属に関する変数と重なる部分があるかを検証する（②）。そして，数値が重ならなければコンピュータ処理を「条件o2」（後記（1 6））に移行させ，重なれば次の行（下記（1 4））へ移行させる（③）。

なお、前記（４）及び（５）のとおり原告プログラムでは各テーブルを〔\$00:30B6〕〔\$00:30B8〕という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。

（14）部分⑭

\$80:0000 BF 57 E5 25	LDA \$25E557, x	①
\$80:0000 29 FF 00	AND #\$00FF	
\$80:0000 C5 B8	CMP \$B8	②
\$80:0000 F0 30	BEQ \$30 ;DATA2	③
\$80:0000 90 2E	BCC \$2E ;DATA2	

「\$25E557」（＝角システムデータベースのシート２における「レベル」（L列）データを意味する）を「A」テーブルに呼び出し（①）、部分⑨で定義したモビルスーツのレベルに関する変数と突合する（②）。そして、モビルスーツのレベルが、「レベル」に関するデータベースで定められた値よりも大きいか、または同等であれば、コンピュータ処理をDATA2（後記（19））に移行させ、未満であれば次の行（下記（15））へ移行させる（③）。

なお、前記（４）及び（５）のとおり原告プログラムでは各テーブルを〔\$00:30B6〕〔\$00:30B8〕という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。

(15) 部分⑮

```
$80:0000 BF 58 E5 25 LDA $25E558, x ①  
$80:0000 29 FF 00 AND #$00FF ②  
$80:0000 D0 33 BNE $33 ;DATA4
```

「\$25E558」（=角システムデータベースのシート2における「f l a g」（M列）データを意味する）を「A」テーブルに呼び出し（①），当該データベースにおける f l a g の値が「0 1」であれば，コンピュータ処理をDATA4に移行させ，「0 1」でなければ次の行（下記（16））へ移行させる（②）。

なお，前記（4）及び（5）のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより，ここでのコーディングもシンプルな記述とすることが可能となったものである。

(16) 部分⑯

条件o2

```
$80:0000 BF 59 E5 25 LDA $25E559, x ①  
$80:0000 25 B6 AND $B6 ②  
$80:0000 F0 2B BEQ $2B ;DATA4 ③
```

原告は，色の切り替えを実行する際の条件として「条件a」「条件o1」「条件o2」の3本立て構成を採用したところ，この部分は「条件o2」についての記述である。

まず，「\$25E559」（=角システムデータベースのシート2における「所属」

(N・O列) データを意味する) を「A」テーブルに呼び出し (①) , 部分⑧で定義したモビルスーツの所属に関する変数と重なる部分があるかを検証する (②) 。そして, 数値が重ならなければコンピュータ処理をDATA4に移行させ, 重なれば次の行 (下記 (17)) へ移行させる (③) 。

なお, 前記 (4) 及び (5) のとおり原告プログラムでは各テーブルを [\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより, ここでのコーディングもシンプルな記述とすることが可能となったものである。

(17) 部分⑰

```
$80:0000 BF 5B E5 25 LDA $25E55B, x ①
$80:0000 29 FF 00   AND #$00FF
$80:0000 C5 B8     CMP $B8 ②
$80:0000 F0 19     BEQ $19      ;DATA3 ③
$80:0000 90 17     BCC $17      ;DATA3
```

「\$25E55B」 (=角システムデータベースのシート2における「レベル」 (P列) データを意味する) を「A」テーブルに呼び出し (①) , 部分⑨で定義したモビルスーツのレベルに関する変数と突合する (②) 。そして, モビルスーツのレベルが, 「レベル」に関するデータベースで定められた値よりも大きいか, または同等であれば, コンピュータ処理をDATA3に移行させ, 未満であれば次の行 (下記 (18)) へ移行させる (③) 。

なお, 前記 (4) 及び (5) のとおり原告プログラムでは各テーブルを [\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより, ここでのコーディングもシンプルな記述とすることが可能となった

ものである。

(18) 部分⑱

```
$80:0000 BF 5C E5 25 LDA $25E55C, x ①
$80:0000 29 FF 00   AND #$00FF
$80:0000 C5 B8     CMP $B8 ②
$80:0000 F0 13     BEQ $13      ;DATA4 ③
$80:0000 90 11     BCC $11      ;DATA4
```

「\$25E55C」（=角システムデータベースのシート2における「レベル」（Q列）データを意味する）を「A」テーブルに呼び出し（①），部分⑨で定義したモビルスーツのレベルに関する変数と突合する（②）。そして，モビルスーツのレベルが，「レベル」に関するデータベースで定められた値よりも大きいか，または同等であれば，コンピュータ処理をDATA4に移行させ，未満であればDATA1（下記（19））へ移行させる（③）。

なお，前記（4）及び（5）のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより，ここでのコーディングもシンプルな記述とすることが可能となったものである。

(19) 部分⑲

```
**DATA1*
$80:0000 68     PLA ①
$80:0000 80 13   BRA $13
**DATA2*
```

\$80:0000 68	PLA	
\$80:0000 18	CLC	
\$80:0000 69 02 00	ADC # \$0002	①
\$80:0000 80 0C	BRA \$0C	
**DATA3*		
\$80:0000 68	PLA	
\$80:0000 18	CLC	
\$80:0000 69 04 00	ADC # \$0004	
\$80:0000 80 05	BRA \$05	
**DATA4*		
\$80:0000 68	PLA	
\$80:0000 18	CLC	
\$80:0000 69 06 00	ADC # \$0006	
\$80:0000 AA	TAX	②

DATA 1は、角システムデータベースのシート2における「paletteアドレス」(T・U列)、DATA 2は同(V・W列)、DATA 3は同(X・Y列)、DATA 4は同(Z・AA列)をそれぞれ意味する。ここでは、どのパレットアドレスを使用するかを選択する作業が行われる。

これまでの条件への突合結果をもとに割り振られたDATA 1～4において、それぞれが指定するメモ領域(スタック)の値を「A」テーブルに呼び出し(①)、この値を「X」のテーブルに移行させる(②)。

なお、前記(4)及び(5)のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8]という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となった

ものである。

(20) 部分②

\$80:0000 BF E0 FB 34 LDA \$34FB E0, x

\$80:0000 D0 08 BNE \$08

不採用

\$80:0000 B9 00 30 LDA \$3000, y[\$40:3*00] ①

\$80:0000 3A DEC A

\$80:0000 5C C3 F1 02 JMP \$02:F1C3

*採用***

\$80:0000 0A ASL A ②

\$80:0000 0A ASL A

\$80:0000 0A ASL A

\$80:0000 0A ASL A

\$80:0000 0A ASL A

\$80:0000 18 CLC

\$80:0000 69 E0 F2 ADC #\$F2E0

\$80:0000 5C CC F1 02 JMP \$02:F1CC

「パレットアドレス」の値が「00 00」の場合は角システム不採用となり、バンダイプログラムの方式でパレットアドレスを作成する。すなわち、色の切り替え動作は生じない (①)。

採用になった場合は、原告作成のパレットデータを読み込むための完全なアドレス (=色のデータベースを呼び出すプログラム) を生成する。すなわ

ち、色の切り替え動作が生じる (②)

なお、前記(4)及び(5)のとおり原告プログラムでは各テーブルを[\$00:30B6] [\$00:30B8] という空きメモリ容量にて処理する形に変更したことにより、ここでのコーディングもシンプルな記述とすることが可能となったものである。