

このページは<http://www.smogon.com/forums/showpost.php?p=1448482&postcount=1>からの引用です

この文書は英語が苦手な人にも本スレで議論されている内容が分かるように翻訳しているものです。

対訳版、査読中、

Translation with the original version. It is reviewing.

項目番号が合わないのはpart 1 の番号が変更されているからです。

著者: X-Act (Revision: 2008/12/1版)

6. More Things affected by the PIDポケモンIDに影響を受ける他の要素

6.1 How the Game Makes a Pokemon Shiny

6.1 どのように色違いのポケモンが生成されるのか

Very rarely, you might encounter a Pokemon having a different coloration than usual. This is called a **shiny Pokemon**. The game's criterion for making a shiny Pokemon depends in part on its PID, but also depends on two other numbers.

ごくまれに通常の色と異なるポケモンに遭遇することがある。これは色違い(shiny)ポケモンと呼ばれる。ゲームで色違いのポケモンが生成される基準はポケモンIDに一部依存している。また他に2個の関係する数値がある。

One of these numbers is visible in the game. It is called the **Trainer ID**. This is the number you have when you look at your Trainer card, and is a number between 0 and 65535 (between [0000] and [FFFF] in hexadecimal). The other number is also a number between 0 and 65535 but is unfortunately invisible to you. It is called the **Secret ID**. Currently, there's no way you can know your Secret ID without looking at your save file or without using cheat codes.

2つのうちの一つはゲーム内で知ることが出来る。それはトレーナーIDである。この番号はトレーナーカードを見ることにより知ることが出来、0から65535の値(十六進数で[0000]から[FFFF])を取る(訳者註:日本語版ではIDNo.と記述されている)。もう一つの番号も0から65535の値をとるが、残念なことに通常知ることが出来ない。この番号は秘密ID(Secret ID, SID)と呼ばれる。現在のところ秘密IDを知る方法はセーブファイルを直接見るかチートコードを利用することである。

So how does the game determine that a Pokemon is shiny? Here's how.

では、どのようにポケモンの色違いが決まるのか?これは次のようになっている。

Split the PID in two 16-bit numbers. That is, if the PID was [465DB901], it would split it in two as [465D] and [B901]. We shall call the first 4 hexadecimal digits the **HID**(high ID) and the last 4 hexadecimal digits the **LID**(Low ID). Now convert the HID, LID, Trainer ID (which we shall denote by **TID**for short) and Secret ID (denoted by **SID**for short) to binary, and compare the first 13 bits of all four, like so:

ポケモンIDを2個の16ビットの数値に分解する。ポケモンIDが[465DB901]だとすると[465D]と[B901]となる。最初の4桁をHID(high ID)、残りの4桁をLID(low ID)と呼ぶことにする。そしてHID,LID,トレーナーID(以降TIDと表記する)、秘密ID(以降SIDと表記する)を二進数に変換し次のようにすべての上位13ビットを抽出する。

Code:

```
HID: xxxxxxxxxxxxxx | xxx
LID: xxxxxxxxxxxxxx | xxx
TID: xxxxxxxxxxxxxx | xxx
SID: xxxxxxxxxxxxxx | xxx
```

Here we're counting the number of 1's in the first bit of the HID, LID, TID and SID. If there are either 1 or 3 ones among those bits, the Pokemon is automatically **not shiny**. If there aren't, the game counts the second bits. Again, if there are either 1 or 3 ones among those bits, the Pokemon is automatically not shiny. The game checks the bits of the HID, LID, TID and SID up to their thirteenth one, and all of them would need to have 0, 2 or 4 ones for the Pokemon to be shiny! This is a one in 8192 (2 to the power of 13) chance of happening

.ここで、HID,LID,TID,SIDの先頭ビットに存在する1の数を数える。もし1の数が1または3の場合、ポケモンは自動的に色違いにならない。もし1の数が、0,2,4の場合、上位2ビット目を選択する。そして同じ事を行い、1の数が1か3の場合にはポケモンは自動的に色違いとならない。ゲームではHID,LID,TID,SIDの値の確認を上位13ビット目まで繰り返し行い、すべて1の数が0,2,4の場合にポケモンが色違いとなる。これは8192(2の13乗)分の1の確率である。

(訳者註: 要するに、上位13ビットの排他的論理和 (XOR) をとってそれが0だったら色違いということ。)

Let's have an example. Suppose the Trainer ID of the player is 43288, or [A918] in hexadecimal, and the Secret ID is 6075, or [17BB] in hexadecimal. Suppose also that the PID of the Pokemon is [B58F0B2A]. We split the PID in two, the HID [B58F] and the LID [0B2A], and convert the HID, LID, Trainer ID and Secret ID to binary. Thus we would have:

例を見てみよう。プレイヤーのトレーナーIDが43288(十六進数で[A918])、秘密IDが6075(十六進数で[17BB])であったとする。そしてポケモンのポケモンIDが[B58F0B2A]であったとする。ポケモンIDを2個に分解すると、HIDが[B58F]、LIDが[0B2A]となる。HID,LID,TID,SIDを二進数に変換すると次のような結果が得られる。

Code:

```
HID: 1011010110001 | 111
LID: 0000101100101 | 010
TID: 1010100100011 | 000
SID: 0001011110111 | 011
```

We now count the number of bits in the first bit, second bit, third bit ... up to the thirteenth bit of each of these four 16-bit numbers.

そして4個の16ビットの数から先頭ビット、2番目のビット、3番目のビット...の順に13番目のビットまで数えていく

Code:

```
HID: 1011010110001 | 111
LID: 0000101100101 | 010
TID: 1010100100011 | 000
SID: 0001011110111 | 011
Ones: 2022222420224
```


Combine these 8 bits together to form an 8-bit number, and convert it to decimal. If this number is 28 or more, subtract 28 from this number until it is a number between 0 and 27. This number would then correspond to the Unown symbol according to the following table:

これらの8ビットの値を順に組み合わせ、8ビットの数値を生成した後、十進数に変換する。得られた値を28で割った余りを次の表に当てはめたものがアンノーンの形である。

Code:

Number	Unown Shape
0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H
8	I
9	J
10	K
11	L
12	M
13	N
14	O
15	P
16	Q
17	R
18	S
19	T
20	U
21	V
22	W
23	X
24	Y
25	Z
26	?
27	!

Let's give an example. Suppose an Unown has PID [4C07DE71]. We first convert this PID to binary, getting

例を見てみよう。アンノーンのポケモンIDが[4C07DE71]であったとする。最初にポケモンIDを二進数に変換すると次の値が得られる。

Code:

01001100000001111101111001110001

Combine the underlined bits together, getting 00111001, and convert it to decimal, getting 57. 57 is larger than 27, so it is subtracted by 28, becoming 29. 29 is again larger than 27, so it is subtracted by 28, getting 1. 1 corresponds to B, and hence the Unown would have the shape of a**B**.

下線を引いたビットを順に組み合わせると00111001という値が得られる。この値を十進数に変換すると57となる。57は27より大きいので28で割った余りを求めると1となる。よってアンノーンの形はBとなる。

(訳者註: ポケットモンスター情報センター2号館にある出現リストによってアンノーンの形が決まっただけからどのようにポケモンIDが生成されるかは分かっていない。エメラルドのシンクロ性格一致のように単に形が一致するまで乱数が進むだけかもしれない。要検証)

6.4 How Spinda's spots are determined

6.4 パッチールの点の決定方法

Spinda's spots are the most useless thing ever implemented in the history of computer implementations, but, for the sake of completeness, we explain exactly how Spinda's spots are placed on its rubbery body, because the PID, again, is the culprit.

パッチールの点はコンピューターの実装の歴史上もっとも意味の無いものである。しかし、完全性を期すためにここでパッチールの点がああゴムの体の上のように描画されるのかを説明する。もちろんポケモンIDが元凶である。

Look at Spinda's PID in hexadecimal. Convert each hexadecimal digit into decimal in turn. So, for example, if Spinda's PID is [65492BDA], we would have 6 = 6, 5 = 5, 4 = 4, 9 = 9, 2 = 2, B = 11, D = 13 and A = 10. From these numbers, the co-ordinates (5, 6), (9, 4), (11, 2) and (10, 13) are thus constructed. These correspond to the coordinates of the upper-left hand corner of Spinda's **left face spot**, its **right face spot**, its **left ear spot** and its **right ear spot** respectively. (Spinda's left spots appear on the right in its front sprite and vice-versa.) All Spinda have exactly 4 spots. If some Spinda appear to have less than four spots, it is either because some spots are superimposed on each other or because the coordinate for some spots does not fall completely on Spinda's sprite.

パッチールのポケモンIDを十六進数で見る。その十六進数の各桁を十進数で表記する。例えばもし、パッチールのポケモンIDが[65492BDA]であるとすると6=6, 5=5, 4=4, 9=9, 2=2, B=11, D=13, A=10という値が得られる。これらの値から、座標系 (5, 6), (9, 4), (11, 2), (10, 13) を作成する。これらの値がそれぞれ順にパッチールの「顔の左側の点」、「顔の右側の点」、「左耳の点」、「右耳の点」の左上の点から数えた座標系の位置である(左の点と右の点の表示場所はパッチールのスプライト上でお互い重なっている。他もそうである。)。全てのパッチールは厳密に4個の点を持っている。もしパッチールの点の数が4個より少なかった場合は、いくつかの点重なっているか、パッチールのスプライト上で全く表示されない位置にあるかである。

To put them in place, the game adds 18 to the x-coordinate and 19 to the y-coordinate of the left face spot. It also adds 6 to the x-coordinate and 18 to the y-coordinate of the right face spot. Lastly, it adds 24 to the x-coordinate and 1 to the y-coordinate of the left ear spot. The right ear spot's coordinates are left unchanged.

これらを表示するために、ゲーム内では左の顔の点については(18, 19)を加えている。右の顔の点では(6, 18)、左耳の点では(24, 1)を加えている。右耳の点に変更していない。

(訳者註: プレイ時にはパッチールと向かい合っているので左右が逆転している。)

(訳者註: すなわち得られた値を(18, 19), (6, 18), (24, 1), (0, 0)の相対座標系で表示するということ)

Thus we have:

これらにより次の結果が得られる。

Code:

```
Left face spot upper-left hand coordinates: (23, 25)
Right face spot upper-left hand coordinates: (15, 22)
Left ear spot upper-left hand coordinates: (35, 3)
Right ear spot upper-left hand coordinates: (10, 13)
```

(訳者註: フォーラムのスレッド内でこの項目の結果に基づいてパッチールを表示するプログラムが次

のURLで公開されている<http://www.freewebs.com/gatorshark/Spinda%20Painter.htm>)

7. The PID of Chained Shiny Pokemon

7. 連鎖色違いポケモンのポケモンID

A new feature in Diamond and Pearl is the PokeRadar, which allows you to chain Pokemon. Chained Pokemon, besides having the advantage of always catching Pokemon of the same species, have a greater probability for them to appear shiny. This section explains in detail how the game determines whether or not a chained Pokemon is shiny, and how it constructs its PID so that it conforms to the criterion that appears in the previous section entitled 踐ow the game makes a Pokemon shiny?

ダイヤモンドとパールの新しい機能としてポケトレ(PokeRadar)があり、ポケトレ連鎖が出来るようになった。ポケトレ連鎖によって、同じ種類のポケモンを捕まえられるという利点が得られるようになる上、色違いポケモンの出現率が高くなる。この項目ではポケトレ連鎖で色違いポケモンが出現するか否かがゲーム内で決定される方法と、それらから前の項目で説明した基準を満たしたポケモンIDがどのように生成されるかを説明する。

7.1 The Probability of a Chained Pokemon to appear Shiny

7.1 ポケトレ連鎖での色違いポケモンの出現率

This depends on the chain number you're attained, which cannot be greater than 40. Suppose your chain number is c . The probability that a chained Pokemon is shiny is governed by the following formula:

ポケトレ連鎖での色違いポケモンの出現率は連鎖数に依存しており、40連鎖の時で最大となった以降一定となる。連鎖数を c とするとポケトレ連鎖で色違いが出現する確率は次の式で表される。

Code:

```
Probability = floor((14747 - 40 × c) ÷ (1640 - 40 × c)) ÷ 65536
```

where floor(x) is x rounded down.

floor(x) は小数点以下切り捨て。

Hereunder is the probability of catching a shiny Pokemon knowing your chain length:

これよりポケトレ連鎖数による色違い出現率は次のようになる。

Code:

Chain Length	Probability of catching a Shiny Pokemon
0	1 in 8192
1	1 in 7282
2	1 in 7282
3	1 in 7282
4	1 in 7282
5	1 in 6554
6	1 in 6554
7	1 in 6554
8	1 in 6554
9	1 in 5958
10	1 in 5958
11	1 in 5958
12	1 in 5461
13	1 in 5461
14	1 in 5041
15	1 in 5041
16	1 in 4681
17	1 in 4681
18	1 in 4369
19	1 in 4369
20	1 in 4096
21	1 in 3855
22	1 in 3641
23	1 in 3449
24	1 in 3277
25	1 in 3121
26	1 in 2979
27	1 in 2731
28	1 in 2521
29	1 in 2341
30	1 in 2185
31	1 in 1986
32	1 in 1771
33	1 in 1598
34	1 in 1394
35	1 in 1192
36	1 in 993
37	1 in 799
38	1 in 596
39	1 in 400
40 or more	1 in 200

Notice that the probabilities are not very good unless your chain's length becomes 30 or more, so aim for as long a chain as possible.

ここでの留意点は30連鎖以上にならないと確率はあまり良くないので、可能な限り連鎖数をあげることが求められるということである。

(訳者註: 実際の色違いの入手方法については「色違いポケモン総合スレまとめ」Wikiを見た方が具体的で良い)

7.2 How the PID and IVs of a Chained Shiny Pokemon is created

7.2 連鎖色違いポケモンのポケモンIDと個体値の生成方法

First, the game calls the RNG twice to create the PID as usual. Remember that the first number generated would be the LID (last 16 bits) of the PID, while the second one would be the HID (first 16 bits) of the PID. The game now needs to fix the PID such that the criterion for the Pokemon to appear shiny is attained. This is done as follows.

最初にゲーム内でポケモンIDを作成するために通常通り2回乱数生成器が実行される。ここで、乱数生成器が最初に作成した数値がポケモンIDのLID(下位16ビット)となり、次に生成した数値がポケモンIDのHID(上位16ビット)になることに注意する。ゲーム内では、このポケモンIDを色違いとなるように「修正」しなければならない。これは次のように行われる。
(訳者註:ここは本当にそうなのか要調査。エメラルドとダイパで変わっている?)

Convert the Trainer ID and Secret ID to binary, and look at their thirteenth, twelfth, eleventh, etc. bits of both, until we reach their first bit. Suppose we're looking at the i 'th bit. For each of these pairs of bits, the RNG is called once and the resulting random number's last bit is noted.

トレーナーIDと秘密IDを二進数に変換し、13,12,11等のビット順に先頭ビットまで見てゆく。 i 番目のビットの処理を考えると、これらのビットの組に対し、乱数生成器が一回ずつ実行され、その結果より得られた最下位ビットを記録する。

If the pair of bits at the i 'th position of TID and SID are the same (i.e. either both are 0 or both are 1), the HID and LID's i 'th bits are **both set to the last bit of the random number generated**. If the pair of bits at the i 'th position of TID and SID are different (i.e. one is 0 and the other one is 1), **the LID's i 'th bit is set to the last bit of the random number generated, and the HID's i 'th bit is set to the reverse bit**(1 if it was 0, 0 if it was 1) of the last bit of the random number generated.

TIDとSIDの i 番目の位置にあるビットのペアが同じ値だった場合(すなわち両方とも0または両方とも1だった場合)、HIDとLIDの i 番目のビットは両方とも乱数生成器が生成した最下位のビットに設定される。もしTIDとSIDの i 番目の位置にあるビットのペアが異なる場合(すなわち片一方が0でもう片方が1の場合)、LIDの i 番目のビットは乱数生成器が生成した乱数の最下位ビットとなり、HIDの i 番目のビットはその逆の値となる(1であった場合には0, 0であった場合には1)。

Following this procedure, this would ensure that the number of 1's in the i 'th position of the TID, SID, HID and LID is either 0, 2 or 4. Since this would be true for all bits between the first and the thirteenth, the Pokemon would be forced to appear shiny.

この手法に従い、TID, SID, HID, LIDの i 番目のビットに含まれる1の数が0,2,4になることが保証される。先頭ビットから13ビット目まで全てのビットでこの手法が成立することによって、ポケモンは「強制的」に色違いになる。

After this is done, there are two subsequent calls to the RNG to create the IVs as usual. Thus, the RNG is called **17 times** in all to generate a chained shiny Pokemon: twice to generate the initial PID, 13 times to fix the PID, and twice to generate the IVs.

これらが実行された後、乱数生成器が2回実行され、通常通り個体値が生成される。連鎖色違いポケモンを生成する為に、乱数生成器は17回実行される。2回は最初のポケモンIDを生成するのに使われる。13回はポケモンIDを修正するのに使われる。2回は個体値を生成するのに使われる。

7.3 An Example of generating the PID of a Chained Shiny Pokemon

7.3 連鎖色違いポケモンポケモンの生成例

Suppose the player encounters a chained shiny Pokemon. We suppose the player has Trainer ID (TID) 38710 and Secret ID (SID) 13099. We also suppose that the current RNG seed is [69AC550F]. We call the RNG twice, getting the numbers [18E5] and [1DD6]. Hence, the LID is [18E5] and the HID is [1DD6]. The numbers in binary are:

プレイヤーが連鎖色違いポケモンに遭遇したとする。トレーナーID(TID)が38710,秘密ID(SID)が13099であったとする。また、現在の乱数の種が[69AC550F]であったとする。乱数生成器が2回実行さ

れ、[18E5], [1DD6]という値が得られる。よってLIDが[18E5], HIDが[1DD6]となる。二進数での値は次のようになる。

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 0001110111010110
LID: 0001100011100101
```

We now need to fix the HID and LID such that the Pokemon is shiny. We start looking from the thirteenth bit.

次にHIDとLIDを色違いポケモンになるように修正する。最初に13ビット目を注目する。

We call the RNG, getting [A013]. When converted to binary, its last digit is 1. (To know at a glance if the last bit of a hexadecimal number is 0 or 1, look at the last hexadecimal digit. If it is 0, 2, 4, 6, 8, A, C or E, the last bit would be 0, otherwise it would be 1.)

乱数生成器を実行すると[A013]という値が得られる。二進数に変換して最下位ビットを取ると1となる(簡単に十六進数の最下位ビットが0であるか1であるかを知るためには、16進数の下一桁を見ればよい。0,2,4,6,8,A,C,Eの場合最下位ビットが0であり、それ以外は1である)。

The TID and SID's thirteenth bits are 0 and 1 respectively, hence they are different. Thus, we make the thirteenth bit of LID as 1 (last digit of the last random number generated) and the first bit of HID as 0. So we have:

TIDとSIDの13ビット目はそれぞれ0と1であり、値が異なる。これから、LIDの13ビット目は1(直前に生成された乱数の最下位ビット)であり、HIDのビットは0となる。これらより次の値が得られる。(訳者註: first bit of HID as 0となっているのは生成した最初のビット i.e. 13ビット目という意味)

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 0001110111010110
LID: 000110001110101
```

Next we go to the twelfth bit. We generate another random number, getting [A4AE], whose last bit is 0. TID's and SID's twelfth bits are 1 and 0 respectively. Hence the twelfth bit of the LID is made to be 0 and the HID's twelfth bit is made to be 1.

次に12番目のビットに移動する。もう一つ乱数生成器を実行すると[A4AE]という値が得られ、最下位ビットは0である。TIDとSIDの12ビット目はそれぞれ1と0である。これよりLIDの12ビット目は0となり、HIDの12番目のビットは1となる。

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 0001110111010110
LID: 000110001110101
```

Moving on to the eleventh bit, we call the RNG again, getting [77AD] whose last bit is 1. The TID and SID's eleventh bit are both 1. Thus we make both the HID and the LID's eleventh bits to be 1.

11番目のビットに移動し、乱数生成器を再度実行すると[77AD]という値が得られる。最下位ビットは1である。TIDとSIDの11番目のビットは両方とも1である。これよりHIDとLIDの11番目のビットは1と

なる。

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 0001110111110110
LID: 0001100011101101
```

For the tenth bit, we again call the RNG, getting [B237] whose last bit is 1. The TID and SID's tenth bits are both 0. Thus we make the fourth bit of LID and HID both 1.

10番目のビットでも乱数生成器を実行し[B237]という値が得られる。この最下位ビットは1である。TIDとSIDの10番目のビットは両方とも0である。これよりLIDとHIDの10番目のビットは両方とも1となる。

(訳者註: fourth となっているが、4回目に生成したビット i.e. 10番目のビットという意味)

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 0001110111110110
LID: 0001100011101101
```

Continuing doing this until we reach the first bit, we get:

先頭ビットまでこれを繰り返すことにより次の値が得られる。

Code:

```
TID: 1001011100110110
SID: 0011001100101011
HID: 1100001101110110
LID: 0110011101101101
Ones: 2222024402422
```

The last bit of each of the subsequent random numbers generated was 0, 1, 1, 1, 0, 0, 1, 1, 0 for the ninth, eighth, seventh, ... first bit respectively. As you can see, the HID and LID now satisfy the criterion for the Pokemon to appear shiny.

乱数生成器が生成した乱数の最下位ビット列は9番目、8番目... 1番目の順に0, 1, 1, 1, 0, 0, 1, 1, 0となる。この結果はHIDとLIDは色違いとなる基準を満たしている。

Thus the PID of the chained shiny Pokemon would be

よって連鎖色違いポケモンのポケモンIDは次のようになる。

Code:

```
11000011011101100110011101101101
```

which is equivalent to [C376676D] in hexadecimal, or 3279316845 in decimal. Two subsequent RNG calls would then be made to create the Pokemon's IVs normally.

この値は十六進数では[C376676D]であり十進数では3279316845となる。最後に乱数生成器が2回実行されポケモンの個体値が通常通り生成される。

Last edited by X-Act : December 1st, 2008 at 05:30 AM. Reason: Fixed RNG calls