

プチコンでロールプレイングゲームを作る (2)

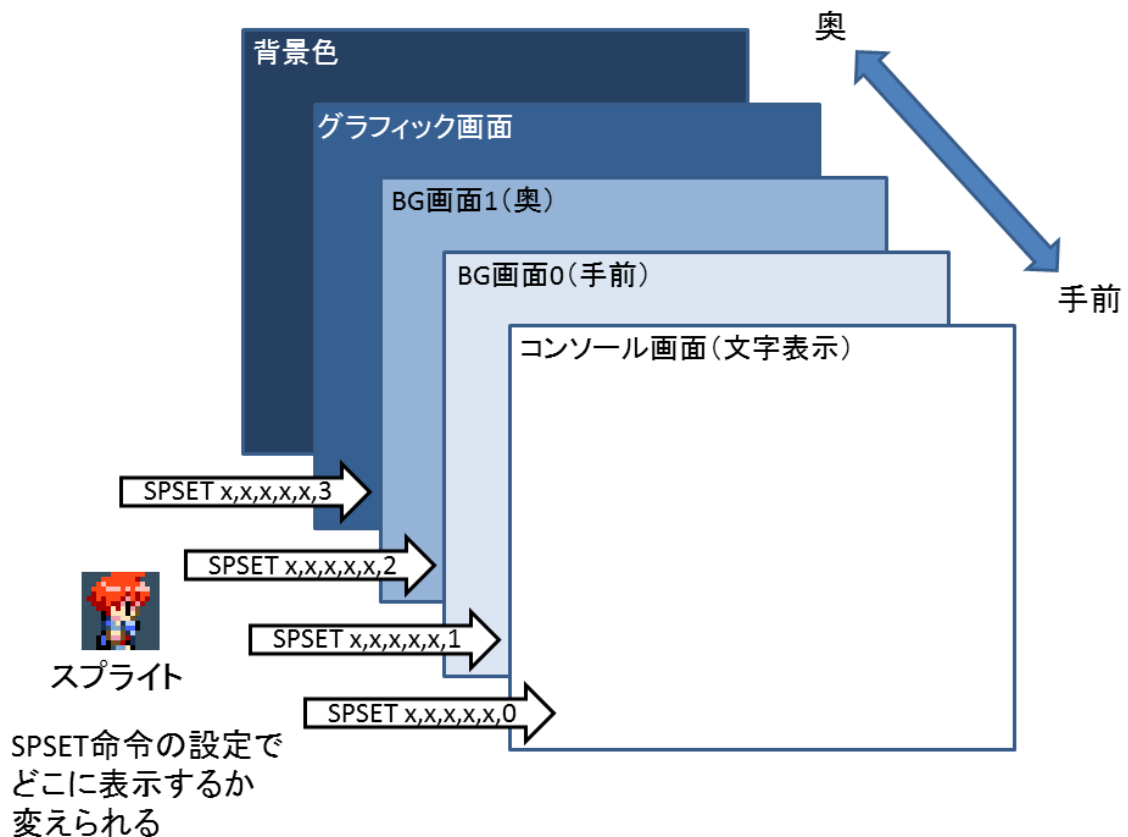
プチコンで作るロールプレイングゲームに、背景や他のキャラクターを追加してみましょう。

●背景を表示する

これまでは真っ暗な画面にキャラクターを表示していましたが、背景を表示してみましょう。

★BG 画面を使う

プチコンの画面には、背景用の「BG 画面」が 2 枚あります。



BG 画面を使うには、まず BGPAGE (ページページ) 命令で、3DS の上画面／下画面のどちらで BG 画面を操作するかを設定します。

BGPAGE 0 上画面の BG 画面を操作

BGPAGE 1 下画面の BG 画面を操作

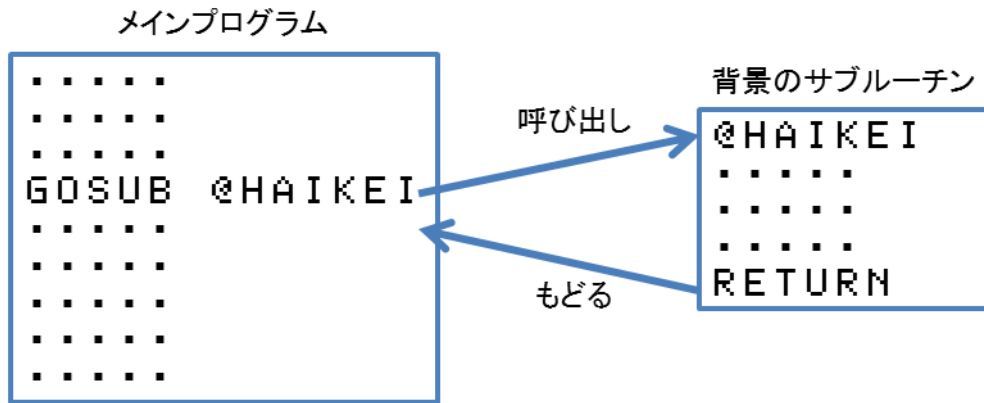
```

1 '*** RPG4 ***
2
3 ACLS
4 BGPAGE 0
5
6 'サイショ
  
```

今回は上画面で使うので、ACLS で画面をクリアした後に BGPAGE 0 を入れましょう。

★サブルーチンの呼び出し

背景の表示は後で何度か使うので、メインのプログラムとは別の「サブルーチン」にして、何度も呼び出せるようにします。



プログラムの中で何度も同じ処理をする場合、その部分をサブルーチンにして、使いたい時に呼び出すようにすると、同じプログラムを何度も書かなくて済みます。

まず、キャラクターを表示し始める前に、背景を表示するサブルーチンを呼び出します。サブルーチンを呼び出すのは「GOSUB」(ゴーサブ)命令を使います。GOSUB 命令の文法は以下のとおりです。

```
GOSUB    @HAIKEI
          ジャンプ先ラベル
```

GOTO 命令と同じように、ジャンプする先は「ラベル」で指定します。ラベルは、「@」(アットマーク)に続けてアルファベットや数字で指定します。

```

6  ' サイショ
7  X=0
8  Y=0
9  SPSET 0,64,2,0,0,2
10 SPANIM 0,4,10
11 SPOFS 0,X,Y
12
13 GOSUB @HAIKEI ←
14
15 ' --- アルク ---
```

★背景を表示するサブルーチン

プログラムの最後に、背景を表示するサブルーチンを追加します。

まず、コメント行と、GOSUB 命令で呼んだラベルを書きます。

```
36 ' --- ハイウェイ ---
37 @HAIKEI
```

次に、背景を表示します。

今回は、全体を芝生(BG キャラクター32番)にしてみました。



BG 画面を一面にうめるには「BGFILL」(ビージーフィル)命令を使います。

BGFILL 命令の文法は以下のとおりです。

```
BGFILL 1 ,0 ,0 ,31 ,23 ,32 ,8 ,0 ,0
        BG  始点  始点  終点  終点  キャラ パレット  横  縦
        画面  x    y    x    y    番号  番号  反転  反転
```

BG 画面	表示する BG 画面のレイヤー。0=手前、1=奥。
始点 x	BG キャラクターを表示するエリアの左上の x 座標。0～63。
始点 y	BG キャラクターを表示するエリアの左上の y 座標。0～63。
終点 x	BG キャラクターを表示するエリアの右下の x 座標。0～63。
終点 y	BG キャラクターを表示するエリアの右下の y 座標。0～63。
キャラ番号	表示する BG キャラクター。0～1023。
パレット番号	表示に使う色パレット。0～15。
横反転	0=なし、1=反転。
縦反転	0=なし、1=反転。

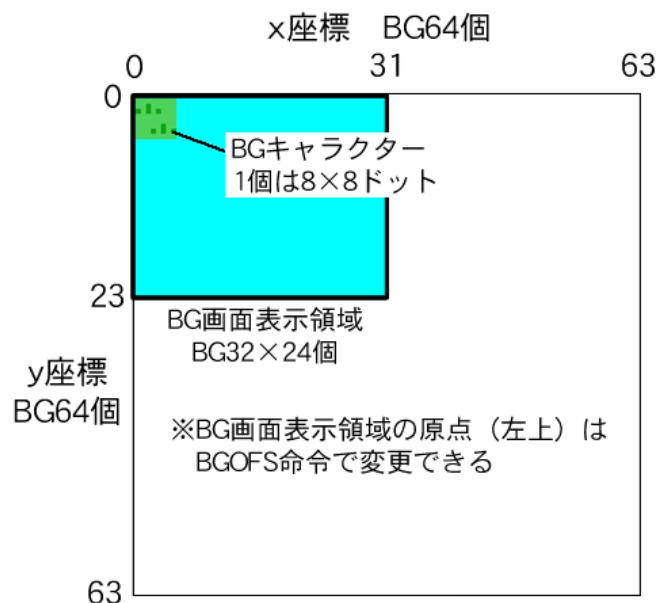
BG キャラクターは、画面の1文字分(8×8ドット)の大きさです。

始点・終点の座標は、BG キャラクターの単位で指定します。

画面に表示できる BG 領域は、横(x 方向)が BG32 個、縦(y 方向)が BG24 個分です。

なので、始点(画面左上)は「x=0, y=0」、終点(画面右下)は「x=31,y=23」になります。

BG 画面は、実際に表示する画面の 4 倍以上の大きさがあります。



背景表示サブルーチンに、BGFILL 命令を書きます。

```
36 ' --- ハイキ ---  
37 @HAIKEI  
38  
39 BGFILL 1, 0, 0, 31, 23, 32, 8, 0, 0  
40  
41 RETURN
```

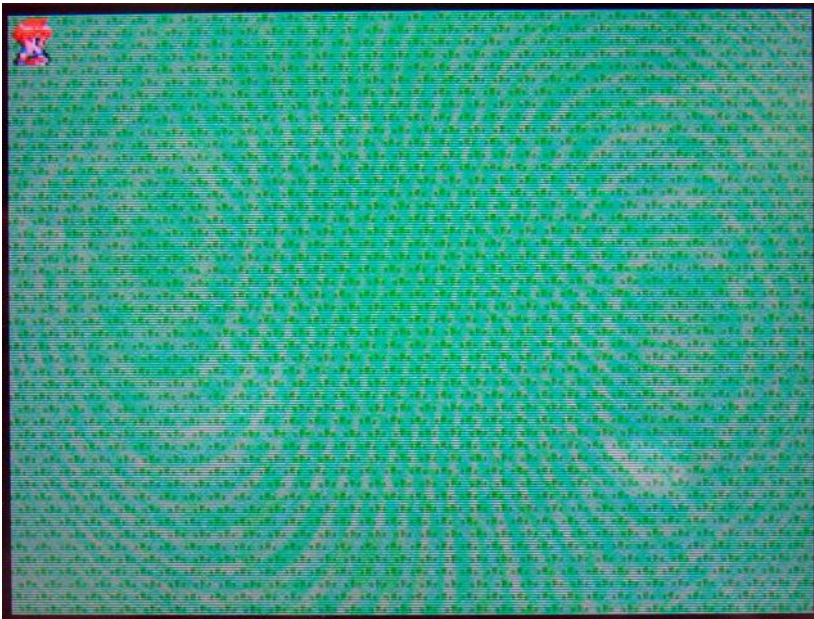
サブルーチンの最後には、「RETURN」(リターン)命令を書きます。

RETURN 命令は、サブルーチンの終わりを表します。

プログラムの実行は、このサブルーチンを呼び出した「GOSUB @HAIKEI」の次の命令へ戻ります。

プログラムを実行すると、背景に一面の芝生が表示されます。

だいぶゲーム画面らしくなりました。



★背景に他の物を表示する

一面の芝生だけではおもしろくないので、他の物を表示してみましょう。
BG キャラクターの一覧表に、大きめの草があるので、それを表示します。
BG キャラクター4 個で、1 つの草になります。



まず、この草を1 個表示してみましょう。

指定した BG キャラクターを1 個表示するには、「BGPOT」(ビージーポット) 命令を使います。
BGPOT 命令の文法は以下のとおりです。

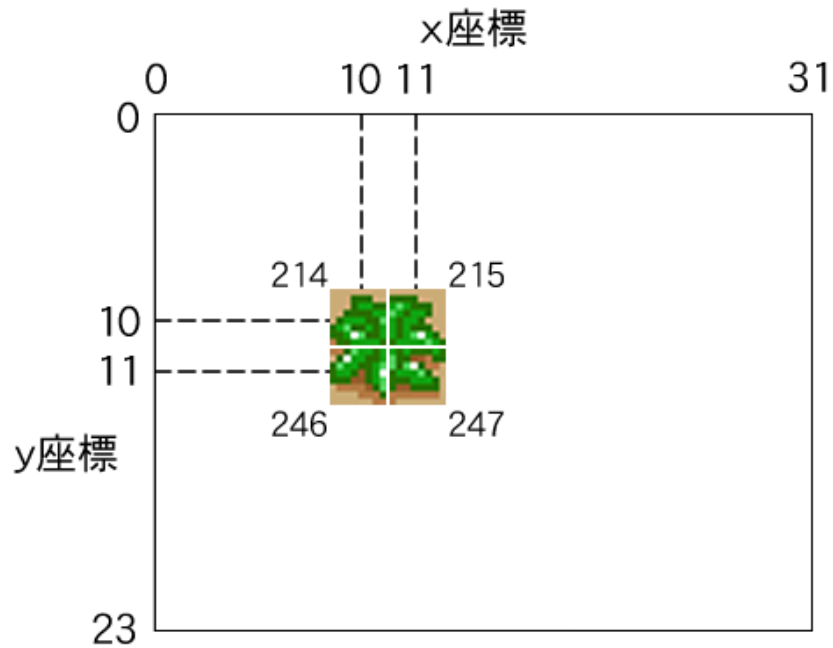
```
BGPOT   1      ,10    ,10    ,214    ,8     ,0     ,0
         BG   x座標  y座標  キャラ  パレット  横     縦
         画面                                番号   番号   反転   反転
```

BG 画面	表示する BG 画面のレイヤー。0=手前、1=奥。
x 座標	BG キャラクターを表示する x 座標。0~63。
y 座標	BG キャラクターを表示する y 座標。0~63。
キャラ番号	表示する BG キャラクター。0~1023。
パレット番号	表示に使う色パレット。0~15。
横反転	0=なし、1=反転。
縦反転	0=なし、1=反転。

芝生を表示する BGFILL 命令の後に、草を表示する BGPOT 命令を入れます。

```
36 ' --- ハイケイ ---
37 @HAIKEI
38
39 BGFILL 1, 0, 0, 31, 23, 32, 8, 0, 0
40
41 ' 草
42 BGPOT 1, 10, 10, 214, 8, 0, 0
43 BGPOT 1, 11, 10, 215, 8, 0, 0
44 BGPOT 1, 10, 11, 246, 8, 0, 0
45 BGPOT 1, 11, 11, 247, 8, 0, 0
46
47 RETURN
```

このプログラムでは、BG 画面の、x 座標 = 10、y 座標 = 10 の位置から、草のキャラクターを 4 個並べて表示します。



実行すると、芝生の中に草が表示されます。



x 座標・y 座標の数字を変えて、草の位置をいろいろ変えてみましょう。

★BG キャラクターをランダムな位置に表示する

さっきのプログラムでは、決まった位置に草を 1 個表示しました。

実際のゲームでは、いつも決まった位置に表示されるとおもしろくないので、表示位置(x 座標、y 座標)を毎回変えてみましょう。

位置を決めるのに、乱数(らんすう、毎回違うランダムな数字)を使います。

乱数を使うには、「RND」(ランダム)関数を使います。

RND 関数の文法は以下のとおりです。

```

      A=      RND (      32      )
      値を受け取る      最大値
      変数
  
```

上のプログラムで、「0,1,2...31(最大値-1)」までの値が、変数 A に入ります。

この RND 関数を使って、草の x 座標、y 座標をランダムに決めて、表示してみましょう。

```

36 ' --- ハイケイ ---
37 @HAIKEI
38
39 BGFILL 1, 0, 0, 31, 23, 32, 8, 0, 0
40
41 ' クサ
42 KX=RND(31)
43 KY=RND(23)
44 BGPOT 1, KX , KY , 214, 8, 0, 0
45 BGPOT 1, KX+1, KY , 215, 8, 0, 0
46 BGPOT 1, KX , KY+1, 246, 8, 0, 0
47 BGPOT 1, KX+1, KY+1, 247, 8, 0, 0
48
49 RETURN
  
```

RND 関数で、草の x 座標(0~30)を変数 KX に、y 座標(0~22)を変数 KY に入れて、BGPOT 命令で表示しています。

(BG 画面の x 座標は 0~31 ですが、草の幅が 2 個分あるので、乱数の範囲は 0~30 にしています。y 座標も同様です)

4 行の BGPOT 命令の KX,KY の辺りは、わざとスペース(空白文字)を入れて、文字位置をそろえています。こうするとプログラムが見やすくなります。

プログラムを実行してみましょう。実行する度に、いろいろな位置に草が表示されます。

さらに、草が1個だけだとさびしいので、FOR~NEXTのくり返しではさんで、5個表示してみましょう。

```
36 ' --- ハイイ ---  
37 @HAIKEI  
38  
39 BGFILL 1, 0, 0, 31, 23, 32, 8, 0, 0  
40  
41 ' 草  
42 FOR K=1 TO 5  
43 KX=RND(31)  
43 KY=RND(23)  
44 BGPOT 1, KX , KY , 214, 8, 0, 0  
45 BGPOT 1, KX+1, KY , 215, 8, 0, 0  
46 BGPOT 1, KX , KY+1, 246, 8, 0, 0  
47 BGPOT 1, KX+1, KY+1, 247, 8, 0, 0  
48 NEXT K  
49  
50 RETURN
```

実行すると、草が5個表示されます。



同じ方法で、いろいろなBGキャラクターを背景に表示することができます。
BGキャラクター一覧表を見ながら、試してみましょう。

●村人を表示する

プレイヤー以外のキャラクターとして、村人を登場させてみましょう。
背景表示と同じく、村人表示用のサブルーチンを作ります。



GOSUB 命令で背景表示サブルーチンを呼んだ後に、村人表示サブルーチンを呼びます。

```

13 GOSUB @HAIKEI
14 GOSUB @HITO
15
16 ' ---   アルク   ---

```

プログラムの一番最後に、村人表示サブルーチンを追加します。
まず、コメント行とラベルを書きましょう。

```

52 RETURN
53
54 ' ---   うひゝト   ---
55 @HITO

```

背景で草を表示した時と同じように、RND 関数でグラフィック座標の x 座標、y 座標を決めて、村人をスプライト 1 番に設定して、表示します。

```

54 ' ---   うひゝト   ---
55 @HITO
56
57 HX=RND(241)
58 HY=RND(177)
59 SPSET 1, 100, 2, 0, 0, 2
60 SPOFS 1, HX, HY
61 RETURN

```

スプライトは 16×16ドットなので、画面サイズ(x=0~255, y=0~191)に収めるには、x=0~240, y=0~176 の範囲になるように RND 関数を設定します。

プログラムを実行してみましょう。
画面に村人が表示されます。



★村人と出会って話す

画面に村人を表示しましたが、今は表示しただけなので、プレイヤーが村人の所へ行ってもすり抜けるだけで何も起きません。

村人に出会ったら話をするようにしてみましょう。

プチコンには、スプライト同士が重なったかどうか調べる、「SPHITSP」(エスピー・ヒット・エスピー)関数があります。それを使います。

SPHITSP 関数の文法は、以下のとおりです。

A=	SPHITSP (0	,	1)
値を受け取る		スプライト		スプライト	
変数		番号1		番号2	

SPHITSP 関数が返す値は、スプライト番号1と2の 2 つのスプライトが重なっている時は「TRUE」(トゥルー、真)、重なっていない時は「FALSE」(フォールス、偽)になります。

IF命令と組み合わせて使うといいです。

プレイヤーが歩くプログラムの中で、SPHITSP 関数で村人と出会ったか調べて、出会っていたら村人と話すサブルーチンを呼ぶようにします。

```

32 'コンカイノ キーヲ ホソクン
33 B0=B
34
35 IF SPHITSP(0,1) THEN GOSUB @HANASU
36
37 GOTO @ARUKU

```

プレイヤーのスプライトが0番、村人のスプライトが1番なので、SPHITSP(0,1)で2つのスプライトが重なったかどうかを調べています。

次に、村人と話すサブルーチンを作ります。

話の内容は、下画面(パネル)に表示するようにします。

パネルにキーボードの代わりに文字を表示するには、プログラムの最初で「PNLTYPE」(パネルタイプ)命令を使って、パネルの表示モードを切り替えます。

```

3 ACLS
4 PNLTYPE "OFF"
5 BGPAGE 0

```

PNLTYPE 命令の文法は、以下のとおりです。

PNLTYPE "OFF"
 設定文字列

設定文字列	パネルに表示する内容を設定する。 "OFF" キーボードやバーを表示しない "PNL" バーだけ表示する "KYA" 英語のキーボードを表示 "KYM" 記号のキーボードを表示 "KYK" カナのキーボードを表示
-------	---------------------------------------------------------------------------------------------------------------------------

今回は村人との話をパネルに表示したいので、PNLTYPE "OFF"で何も表示しないようにします。

何も表示しないと、プログラムを途中で止める「停止」ボタンが押せないので、止めたい時はSELECT ボタンを押してください。

そして、プログラムの最後に、村人と話すサブルーチンを追加します。

```

66 ' --- ムラビトト ハナス ---
67 @HANASU
68
69 PNLSTR 0,0,"コンニチハ"
70 RETURN
  
```

パネルに文字を表示するには、「PNLSTR」(パネルistring)命令を使います。

(上画面のように PRINT 命令は使えません)

PNLSTR 命令の文法は、以下のとおりです。

PNLSTR 0 ,0 ,"コンニチハ"
 x 座標 y 座標 表示文字列

プログラムを実行します。

プレイヤーを操作して、村人に出会わせると、下のパネルに「コンニチハ」と表示されます。



このままだと、村人と離れた後も「コンニチハ」と表示されっぱなしになってしまいます。「コンニチハ」と表示した後、A ボタンを押したら表示が消えるようにしてみましょう。

```

66 ' --- ムラビト ト ハナス ---
67 @HANASU
68
69 PNLSTR 0,0,"コンニチハ @"
70 @HANASU2
71 IF BUTTON() != 16 THEN GOTO @HANASU2
72 PNLSTR 0,0," "*32
73 RETURN

```

A ボタンを押したら次へ進むので、ボタン表示を追加

PNLSTR 命令で「コンニチハ」を表示した後、IF 命令と BUTTON 関数で、A ボタンが押されたかどうかを調べています。

A ボタンが押されると BUTTON 関数の値は 16 になるはずなので、16 と等しくない(!=)時には GOTO 命令で @HANASU2 へ戻して、ループさせています。

A ボタンが押された時は、次の行の PNLSTR 命令へ移り、表示された文字を消しています。「 "*32」は、スペース(空白文字)が 32 文字分です。

これでプログラムを実行してみましょう。

プレイヤーを操作して、村人と出会わせると、パネルに「コンニチハ」と表示されます。

A ボタンを押すと……文字が消えるのですが、プレイヤーと村人のスプライトが重なったままなので、再び村人と出会ったことになって、また「コンニチハ」と表示されてしまいます。

これではおもしろくないので、会話した後に、村人が別の場所へ移動するようにしましょう。

パネルの文字を消した後に、村人を表示するサブルーチン呼び出せば、違う場所に表示されます。

```

66 ' --- ムラビト ト ハナス ---
67 @HANASU
68
69 PNLSTR 0,0,"コンニチハ @"
70 @HANASU2
71 IF BUTTON() != 16 THEN GOTO @HANASU2
72 PNLSTR 0,0," "*32
73 GOSUB @HITO
74 RETURN

```

村人との話の文字列を変えるなどして、いろいろ会話してみてください。

● 宝箱を登場させる

村人の次は、宝箱を登場させてみましょう。(キャラクター57番)



プログラムの最初の、GOSUB 命令で村人を表示する所で、宝箱を表示するサブルーチンを呼ぶ GOSUB 命令を追加します。

```

14 GOSUB @HAIKEI
15 GOSUB @HITO
16 GOSUB @TAKARA
17
18 ' --- 宝箱 ---

```

次に、プログラムの最後に、宝箱を表示するサブルーチンを追加します。

村人を表示するサブルーチンと同じように、乱数で位置を決めて、宝箱のSpriteを表示します。

```

77 ' --- タカラハコ ---
78 @TAKARA
79
80 TX=RND(241)
81 TY=RND(177)
82 SPSET 2, 57, 2, 0, 0, 2
83 SPOFS 2, TX, TY
84 RETURN

```

Sprite番号は2番に割り当てています。

プログラムを実行すると、画面上に宝箱が表示されます。



★宝箱を開ける

プレイヤーが宝箱に出会ったら、箱を開けて何かを見つけるようにしてみましょう。
ストーリーはいろいろ考えられますが、今回は宝箱から薬草を見つけて、プレイヤーの体力(HP)が回復する、という設定にします。

まず、プログラムの最初で、プレイヤーの体力(HP)の初期設定をしましょう。
変数 HP を用意して、初期値を 50 にします。

```

7  ' サイショ
8  X=0
9  Y=0
10 SPSET 0,64,2,0,0,2
11 SPANIM 0,4,10
12 SPOFS 0,X,Y
13 HP=50

```

次に、プレイヤーを移動するループの中で、宝箱の sprite(2番)との重なりを調べます。
村人との重なりと同じように、IF 命令と SPHITSP 関数を使います。

```

35 ' コンカイノ キーヲ ホツゲン
36 B0=B
37
38 IF SPHITSP(0,1) THEN GOSUB @HANASU
39 IF SPHITSP(0,2) THEN GOSUB @AKERU
40
41 GOTO @ARUKU

```

宝箱を開けるサブルーチンを、プログラムの最後に追加します。
まず、コメント行とラベルを書きましょう。

```

88 ' --- タカラハダ コヲ アケル ---
89 @AKERU

```

せっかくなので、宝箱を開けたスプライトを表示してみましょう。

すでにあるスプライトのキャラクターを変更するには、「SPCHR」(エスピーキャラクター)命令を使います。

```

88 ' --- タカラハコ コマ アケル ---
89 @AKERU
90
91 SPCHR 2,56
92 RETURN

```

プログラムを実行してみましょう。

プレイヤーを移動させて、宝箱に重ねると、宝箱が開きます。



SPCHR 命令の文法は、以下のとおりです。

SPCHR	2	,56	,2	,0	,0	,0
	スプライト 管理番号	キャラクター 番号	パレット 番号	横 反転	縦 反転	優先 順位

スプライト管理番号	変更したいスプライトの管理番号(0~99)。
キャラクター番号	新しく設定するキャラクター番号(0~1023)。
[パレット番号]	表示に使う色パレット。0~15。
[横反転]	0=なし、1=反転。
[縦反転]	0=なし、1=反転。
[優先順位]	スプライトを表示する優先順位 0=コンソール(文字表示)の前 1=手前の BG 面(背景表示面)より前 2=2 枚の BG 面の間 3=奥の BG 面の後ろ

※[]内の数値は省略可能。

今回の「宝箱を開ける」以外にも、例えば「モンスターが変身して別のモンスターになる」、「敵機が撃たれて爆発する」など、その場でスプライトを変更したい時に使えます。

★薬草を見つけて HP を増やす

宝箱を開けた後に、薬草を見つけて HP を増やすプログラムを追加しましょう。

基本は、村人と話すプログラムと同じです。

パネルに、薬草を見つけたこと、HP が増えたことを表示するようにします。

```

88 ' --- タカラハコヲ アケル ---
89 @AKERU
90
91 SPCHR 2, 56
92 HP=HP+10
93 PNLSTR 0, 0, "ヤクソウヲ ミツケタ! HP=" + STR$(H
P) + " 0"
94 @AKERU2
95 IF BUTTON() != 16 THEN GOTO @AKERU2
96 PNLSTR 0, 0, " " * 32
97 GOSUB @TAKARA
98 RETURN

```

まず、「HP=HP+10」で、変数 HP の値を 10 増やします。

次の行で、「薬草を見つけた!」と、増やした HP の値を表示します。

HP の値を表示する所で、「STR\$」(STRING) 関数を使って、数値変数 HP を文字列に変換しています。PNLSTR 命令は文字しか表示できないので、数値は文字列に変換しないと表示できません。また、文字列同士は「+」でつながられます。

その後は、村人に出会った時のプログラムと同じように、A ボタンが押されるのを待って、パネルの文字を消しています。

宝箱の位置がそのままだと、また開けてしまうので、GOSUB 命令で宝箱表示サブルーチンを呼んで、位置を変えています。

プログラムを実行してみましょう。

宝箱に出会うと、宝箱が開いて、下のパネルにメッセージが表示されます。

