

NScripter の応用

アドベンチャーゲームなどを作れるソフト「NScripter」(エヌスクリプター)の応用です。

●表示効果

背景やキャラクターの立ち絵の画像を表示する時、これまではパッと一瞬で表示していました。これを「だんだん現れる」などの効果をつけて表示することができます。

★表示効果の番号を設定する

テストプログラムに、表示効果をつけてみましょう。

```
*define
caption "テスト"
deletemenu
game

*start
setwindow 16, 350, 25, 4, 24, 24, 0, 6, 20, 1, 1, #999999, 8, 345, 631, 471
bg "img¥haikei. jpg", 1
ld l, "img¥sarutobi. bmp", 1
ぼく、SARUTOBI くん¥
cl l, 1
end
```

表示効果を使うには、まず初期設定部分(「*define」～「game」の間)で、「effect」(エフェクト)命令で呼び出し番号を設定します。

例えば、以下の設定を入れてみてください。

```
*define
caption "テスト"
deletemenu
effect 2, 10, 500
game
```

effect 2, 10, 500

効果呼び出し
番号

エフェクト番号
10 はクロスフェード
(だんだん現れる)

効果秒数(ミリ秒)
1000 分の1秒単位
1000 ミリ秒=1秒

※呼び出し番号は、「1」は瞬間表示に使われているので、「2」より大きい数字をつけます。

★背景や立ち絵の表示に効果をつける

「effect」命令で設定ができれば、背景表示に効果をつけてみましょう。

背景を表示する「bg」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

背景がゆっくり表示されましたか？

```
bg "img¥haikei.jpg", 2
```

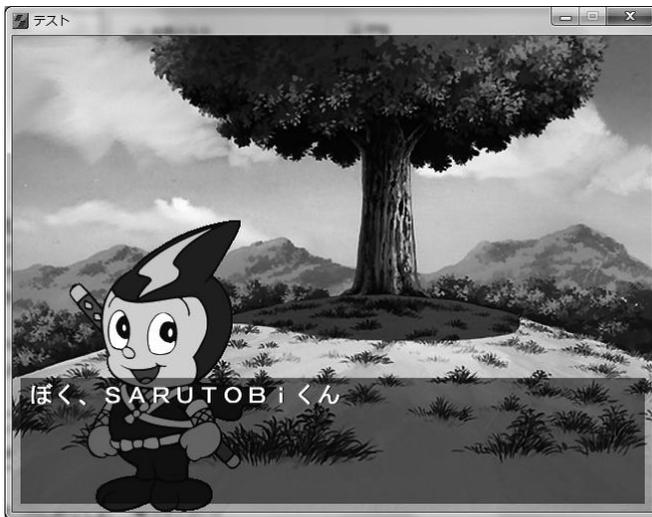
同じように、キャラクターの立ち絵の表示にも、効果をつけてみましょう。

立ち絵を表示する「ld」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

キャラクターがゆっくり表示されましたか？

```
ld 1, "img¥sarutobi.bmp", 2
```



キャラクターの立ち絵を消す時にも、効果がつけられます。

立ち絵を消す「cl」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

キャラクターがゆっくり消えましたか？

```
cl 1, 2
```

このように、効果呼び出し番号を最初に設定すると、何度もその番号で呼び出すことができます。いろいろな表示効果を「effect」命令で設定して、呼び出してみてください。

(例)

```
effect 3, 14, 500 ;呼び出し番号 3 を設定、効果=下からスクロール
```

```
effect 4, 4, 500 ;呼び出し番号 4 を設定、効果=上シャッター
```

```
bg "img¥haikei.jpg", 3
```

```
ld 1, "img¥sarutobi.bmp", 4
```

「effect」命令の2番目の数値で指定する「エフェクト番号」は、以下の 0～18 があります。

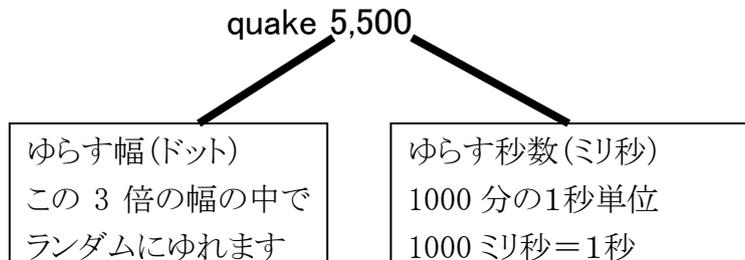
| 番号 | 効果 |
|----|---|
| 0 | 表示効果なし。標準で呼び出し番号「0」に設定されています。 0番をつけた画像は、その場では表示されません。 次に0番以外の番号をつけた画像を表示すると、一緒に表示されます。 立ち絵を左右2枚同時に表示したい時などに使います。 |
| 1 | 瞬間表示。標準で呼び出し番号「1」に設定されています。 |
| 2 | 左シャッター。 |
| 3 | 右シャッター。 |
| 4 | 上シャッター。 |
| 5 | 下シャッター。 2～5のシャッター系は、画像を複数のたんざく形にカットして表示します。 それぞれ左右上下方向に向けて表示されます。 |
| 6 | 左カーテン。 |
| 7 | 右カーテン。 |
| 8 | 上カーテン。 |
| 9 | 下カーテン。 6～9のカーテン系は、シャッターよりも大き目のたんざく形にカットして表示します。 |
| 10 | クロスフェード。透明→不透明にだんだんと表示します。 |
| 11 | 左からスクロール。 |
| 12 | 右からスクロール。 |
| 13 | 上からスクロール。 |
| 14 | 下からスクロール。 11～14のスクロール系は、画面全体がスクロールする効果です。 ただし、背景も立ち絵も一緒にスクロールしてしまうので、背景だけが表示されている時、色背景+立ち絵の時などに使うといいでしょう。 色背景+立ち絵で、立ち絵表示の「ld」命令にこの効果をつけると、キャラクターが動いて登場するようになります。 |
| 15 | マスクパターンによるフェード。難しいので省略。 |
| 16 | モザイクアウト。画像がだんだん大きなモザイクに変化します。背景も立ち絵も一緒にモザイクになります。 「cl」命令で使うと、キャラクターがおかしな消え方をするように見えます。 |
| 17 | モザイクイン。画像がだんだん小さなモザイクになって表示されます。背景も立ち絵も一緒にモザイクになります。 「bg」や「ld」命令で使うと、だんだんレンズの焦点が合う、キャラクターがワープして出現する、などの表現ができます。 |
| 18 | マスクパターンによるクロスフェード。難しいので省略。 |

●画面をゆらす

表示されている画面をゆらすことができます。

キャラクターが何かにぶつかった、バトルで攻撃を受けた、などの効果として使えます。

画面をゆらすには「quake」(クエイク)という命令を使います。



SARUTOBi さんの立ち絵を表示した後に、画面をゆらしてみましよう。

SARUTOBi さんが何かにぶつかったような効果がつけられます。

ゆらす幅や時間をいろいろ変えて、試してみてください。

```
ld l, "img¥sarutobi.bmp", 2
ぼく、SARUTOBi くん¥
quake 5, 500
いてて、ぶつかった¥
```

「quake」命令は上下左右ランダムにゆれますが、左右方向だけ、あるいは上下方向だけゆらすこともできます。

quakex 回数,時間 ←指定した回数と時間だけ左右にゆれる

quakey 回数,時間 ←指定した回数と時間だけ上下にゆれる

例えば、「quakey」命令で上下に1回ゆらすとキャラクターがうなずいたように(Yes)、「quakex」命令で左右に1回ゆらすとキャラクターが首を振ったように(No)見えます。

回数や時間をいろいろ変えて、試してみてください。

```
ld l, "img¥sarutobi.bmp", 2
ぼく、SARUTOBi くん¥
quakey 1, 1000
うん、わかった¥
quakex 1, 1000
ううん、そうじゃない¥
```

●音を出す

BGM や効果音などの音を出す方法です。音が出ると、ゲームにとっても臨場感が出ます。

★BGM を流す

「bgm」(बीジーエム) 命令を使って、ゲームの背景に流れる音楽を設定します。

`bgm "ファイル名"`

BGM のファイルは、
「bgm」フォルダに入っ
ています。

```
*start
setwindow
16, 350, 25, 4, 24, 24, 0, 6, 20, 1, 1, #999999, 8, 345, 631, 471
bgm "bgm¥bgm01. wav" ←
bg "img¥haikei. jpg", 1
```

ゲームがスタートする所で、「bgm」命令を入れてみましょう。

「bgm」命令は、曲が終わるとまた最初からくり返して流れます。(ループ再生)

1 回だけ流して終わりにしたい時は、「bgmonce」(बीジーエム・ワンス) 命令を使います。

`bgmonce "ファイル名"`

流れている BGM を止めるには、「bgmstop」(ビージーエム・ストップ) 命令を使います。

`bgmstop`

BGM の音量(ボリューム) 調整は、「bgmvol」(ビージーエム・ボリューム) 命令を使います。

`bgmvol` 音量(0~100) ←0が最小、100 が最大

BGM の音量が大きいと、他の効果音やせりふが聞きづらいので、音量は 50 くらいにした方が良いでしょう。

BGM を変えたり、命令を変えたりして、いろいろ試してみてください。

★効果音を出す

キャラクターの動き、場面転換などに合わせて、効果音を出すことができます。
効果音を出すには、「**dwave**」(ディーウェーブ)命令を使います。

dwave チャンネル番号, "ファイル名"

「チャンネル番号」は 0~49 の数字を指定します。
キャラクターのせりふはチャンネル 0、それ以外の効果音はチャンネル 1~49 にします。

効果音のファイルは、「sound」フォルダに入っています。

SARUTOBi くんが登場する所で、「dwave」命令を入れて、効果音をつけてみましょう。

```
dwave 1, "sound¥sound01.wav" ←
ld l, "img¥sarutobi.bmp", 1
ぼく、SARUTOBi くん¥
```

「dwave」命令は、1 回だけ効果音を再生します。
くり返し再生したい時は、「**dwaveloop**」(ディーウェーブ・ループ)命令を使います。

dwaveloop チャンネル番号, "ファイル名"

効果音を途中で止めたい時は、「**dwavestop**」(ディーウェーブ・ストップ)命令を使います。

dwavestop チャンネル番号

チャンネルを変えると、2つ以上の効果音を同時に再生できます。

```
dwave 1, "sound¥sound01.wav"
dwave 2, "sound¥sound02.wav"
```

この場合、2 つの効果音が重なって聞こえます。

1 番目の効果音が終わってから 2 番目の効果音を出したい時は、間にクリック待ちのセリフをはさむといいでしょう。

```
dwave 1, "sound¥sound01.wav"
ぼく、SARUTOBi くん¥           ;←クリック待ちの間に 1 番目の効果音が終わる
dwave 2, "sound¥sound02.wav"
```

●変数を使う

例えば、ゲームのプログラムで、最初にプレイヤーの名前やレベル、体力、所持金などを設定して、その後の場面で使うことがあります。

これを実現するには、数値や文字を記憶しておく「変数」(へんすう)を使います。変数は、数値や文字を入れる箱のようなものです。

まず、プレイヤーの名前と所持金を設定してみましょう。

ゲームが開始する時に、変数の値を設定します。

★文字変数

文字が入る「文字変数」は、「\$」に続けて数字を書きます。

文字変数「\$1」を作って、プレイヤーの名前を設定してみましょう。

変数に値をセットするには「mov」(ムーブ)命令を使います。



指定する文字は、「"」(ダブルクォーテーション)で囲みます。

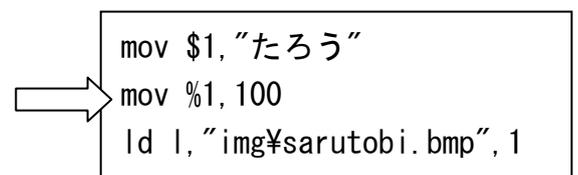
★数値変数

数値が入る「数値変数」は、「%」に続けて数字を書きます。

数値変数「%1」を作って、プレイヤーの所持金を設定してみましょう。

mov %1,100

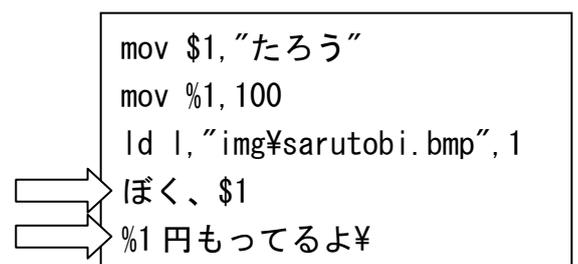
なお、変数の番号は、0~199 までが使えます。



変数に設定した名前と所持金を、画面に表示してみましょう。

表示する文章の中に変数名を書くと、一緒に表示されます。

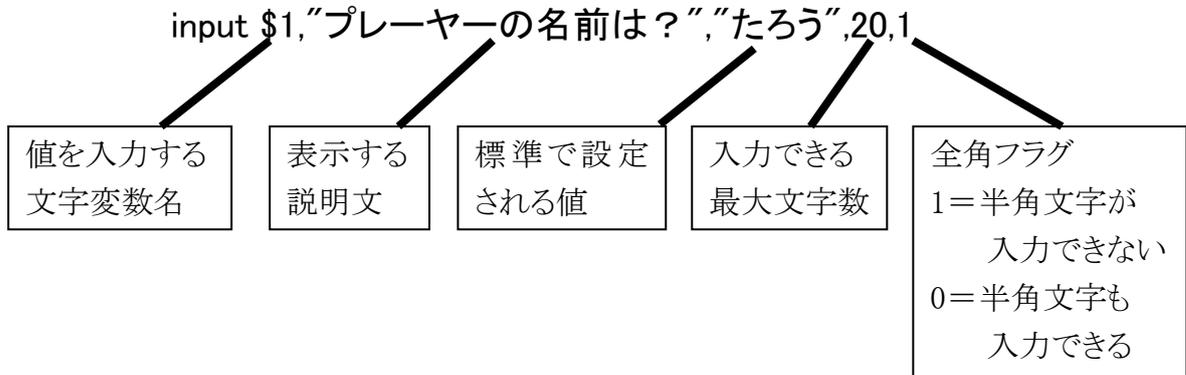
名前や所持金の値をいろいろ変えて、試してみてください。



★文字を入力する

プレイヤーの名前を、キーボードから入力できるようにしてみましょう。

「mov」命令で文字変数「\$1」を指定している行を削除して、代わりに「input」命令を書きます。



⇒

```
input $1,"プレイヤーの名前は?","たろう",20,1
mov %1,100
ld l,"img¥sarutobi.bmp",1
```

実行すると、文字列を入力するためのウインドウが開きます。

名前を入力して「決定」ボタンを押すと、その名前がプレイヤーの名前として表示されます。

いろいろな名前を入れてみてください。



なお、「input」命令の最後の「全角フラグ」を「0」にして、半角文字も入力できるようにすると、名前を画面に表示した時に表示がおかしくなります。通常は「1」にして、半角文字が入力できないようにしてください。

★数値を入力する

次にプレイヤーの所持金の金額を、キーボードから入力できるようにしてみましょう。

「mov」命令で数値変数「%1」を指定している行を削除して、代わりに「input」命令を書きます。

```
input $2,"所持金は？(半角文字で)","100",10,0
```

「input」命令は文字変数にしか入力できないので、新しく文字変数「\$2」を作って、そこに金額を入力します。数字なので、半角文字で入力するようにします。

そして、文字変数「\$2」の値を、数値変数「%1」へ変換します。

変換には「atoi」(エー・トゥー・アイ)という命令を使います。

```
atoi %1,$2 ;文字変数$2 に入っている数字を変換して%1 へ入れる
```

```
input $1,"プレイヤーの名前は？","たろう",20,1
input $2,"所持金は？(半角文字で)","100",10,0
atoi %1,$2
ld l,"img¥sarutobi.bmp",1
```

実行すると、名前の次に所持金を聞かれます。

半角の数字で金額を入力すると、その金額が所持金として表示されます。



★数値変数の計算

プレイヤーの所持金で、お金の計算をしてみましょう。

SARUTOBiくんの後にタヌキおやじを登場させて、買い物をすることにします。

まず、タヌキおやじを登場させてセリフをしゃべらせませす。注意ですが、「100円」の数字は必ず全角文字で入れてください。

```
ld r,"img¥tanuki.bmp",1
いらっしやい
おにぎり100円だよ¥
```

次に、SARUTOBiくんがおにぎりを買って100円使ったとします。

所持金の数値変数「%1」から100を引き算します。

先にやった「mov」命令で、引き算ができます。

```
mov %1,%1-100 ;%1の数値から100を引いて%1へ入れる
```

これを入れて、SARUTOBiくんがタヌキおやじからおにぎりを買ったストーリーを作ってみましょう。



```
いらっしやい
おにぎり100円だよ¥
じゃあ、ください¥
毎度あり!¥
mov %1,%1-100
所持金が%1円になりました¥
```

数値変数では、足し算、引き算、かけ算、割り算などの計算ができます。

```
mov %1,%1+100 ;%1の数値に100を足して%1へ入れる
mov %1,%1-100 ;%1の数値から100を引いて%1へ入れる
mov %1,%1*100 ;%1の数値に100をかけて%1へ入れる
mov %1,%1/100 ;%1の数値を100で割って%1へ入れる
```

なお、それぞれの計算をする専用の命令もあります。

```
add %1,100 ;(アッド) %1の数値に100を足して%1へ入れる
sub %1,100 ;(サブ、subtractの略) %1の数値から100を引いて%1へ入れる
mul %1,100 ;(マル、multiplyの略) %1の数値に100をかけて%1へ入れる
div %1,100 ;(ディブ、divideの略) %1の数値を100で割って%1へ入れる
```

計算の行を書き換えて、いろいろな計算を試してみてください。

★数値変数で条件分岐(じょうけんぶんき)

おにぎりを買った後、残ったお金がある／ないで、ストーリーを変えてみましょう。

(もし～なら～、「条件分岐」)

変数「%1」の値の大小で、行き先を変えます。

条件分岐には「if」(イフ) 命令を使います。



この例は、「変数%1 が 0 より大きい時は、ラベル*aru へジャンプする」という意味になります。これを使って、お金がある時と無い時で、表示する文章を変えてみましょう。

右に書いたプログラムを見て下さい。

SARUTOBi くんが登場する前に、新しいラベル「*kaimono」を入れています。

```

→ *kaimono
ld l, "img¥sarutobi.bmp", 1
ぼく、$1
今%1 円もってるよ¥
ld r, "img¥tanuki.bmp", 1
いらっしやい
おにぎり 100円だよ¥
じゃあ、ください。¥
毎度あり!¥
mov %1, %1-100
所持金が%1 円になりました¥
if %1 > 0 goto *aru

*nai
お金がもうありません¥
end

*aru
まだお金があります¥
goto *kaimono
  
```

買い物をした後に、「if」命令を入れます。

所持金の変数「%1」が 0 より大きい時は、ラベル「*aru」へジャンプして、「まだお金があります」と表示して、買い物の最初「*kaimono」へもどります。

「%1」が 0 かマイナスの時は、ラベル「*nai」以下を実行して、「お金がもうありません」と表示して、ゲームを終了します。

これで、「買い物ストーリー」になりました。

最初に入力する所持金の金額をいろいろ変えて、試してみてください。

「if」命令で使う条件式は、「等しい」「大きい」「小さい」など、いろいろな条件が使えます。

```
if %1=10 goto *hitoshii    ;%1 が 10 と等しかったら*hitoshii へジャンプ

if %1<10 goto *chiisai    ;%1 が 10 より小さかったら*chiisai へジャンプ
                        (10 はふくまれない)

if %1>10 goto *ookii      ;%1 が 10 より大きかったら*ookii へジャンプ
                        (10 はふくまれない)

if %1<=10 goto *ika       ;%1 が 10 以下だったら*ika へジャンプ
                        (10 がふくまれる)

if %1>=10 goto *ijou      ;%1 が 10 以上だったら*ijou へジャンプ
                        (10 がふくまれる)

if %1<>10 goto *nai        ;%1 が 10 と等しくなかったら*nai へジャンプ
```

後ろに書く命令は、「goto」以外も使えます。

例えば、所持金「%1」の値によって表示する文章を変えるプログラムは、下のようによく書くことができます。

ラベルを減らせるので、かんたんになります。

```
mov $2, "まだお金があります"
if %1<=0 mov $2, "お金がもうありません"
$2¥          ;←$2 の内容を画面に表示、クリック待ち
if %1>0 goto *kaimono
end
```

また、入門編で、「select」命令を使って選択肢(せんたくし)を作りました。
これを使って、おにぎり以外のメニューを選べるようにしてみましょう。

タヌキおやじが登場して買い物をする所で、「select」命令でメニューを作ります。
「select」の飛び先で、買う物の金額用の数値変数「%2」に、金額を入れます。
金額を入れたら、「*maido」ラベルにジャンプして、所持金「%1」から代金「%2」を引きます。

```
mov %1,%1-%2
```

ファイルを保存して、実行しましょう。

これで「タヌキおやじ商店」のストーリーになりました。

最初に入力する所持金をいろいろ変えて、買い物してみてください。

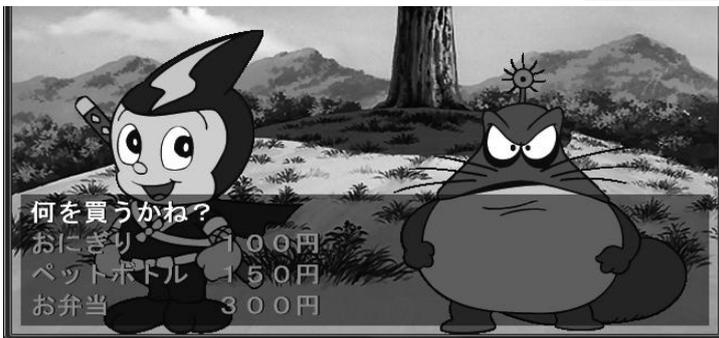
```
ld r, "img\tanuki. bmp", 1
いらっしやい¥
何を買うかね?
select
"おにぎり      100円", *onigiri,
"ペットボトル  150円", *pet,
"お弁当        300円", *obento

*onigiri
mov %2, 100
goto *maido

*pet
mov %2, 150
goto *maido

*obento
mov %2, 300

*maido
毎度あり! ¥
mov %1, %1-%2
所持金が%1 円になりました¥
```



実はこのプログラムだと、所持金以上の値段の買い物をする時、所持金がマイナスになってしまいます。

それを防ぐには、メニューを選んだ後で、もし買う物の値段が所持金より高かったら、メニューにもどって選びなおすようにするといいですね。

プログラムを考えてみてください。