

洋ゲー風、拾得アイテム強化プログラム

概要

探索で拾得できるアイテムに自動でマジック効果が” 付属したように見える ” プログラムです。
TRPG などの洋ゲー風に「凡庸な剣 +1」「凡庸な剣 +2」みたよう名称変更されます。

改造

1. ファイルをダウンロードします

.

ファイルの内容

```
# 設定部分 (下記参照)
sub add_magic {
    if ($_[1] = /$dny_idt/ || ($_[1] ! /$add_iak/ && $_[1] ! /$add_idd/ && $_[1] ! /$add_iad/))
    { return($_[0],$_[1]) }

    $add = $add_idt[int(rand($#add_idt + 1));

    if ($add == 0) { return($inm,$idt) }

    $pc = int(substr($_[1],0,4));
    $iak = int(substr($_[1],4,2));
    $idd = int(substr($_[1],6,2));
    $sts = substr($_[1],8,length($_[1]));

    $new_inm = "$_[0]" . '+' . "$add"; # 名前変更部分
    $pc = $add * $pc_rate + $pc;

    if ($_[1] = /$add_iak/ || $_[1] = /$add_iad/) { $iak += $add }
    if ($_[1] = /$add_idd/ || $_[1] = /$add_iad/) { $add += $add }

    $pc = sprintf("%04d",$pc);
    $iak = sprintf("%02d",$iak);
    $idd = sprintf("%02d",$idd);
    $new_idt = "$pc$iak$idd$sts";

    return($new_inm,$new_idt);
}
1;
```

2. 設定の変更

DL したファイルを編集し、設定項目を任意に変更します。

デフォルトでは以下のようにになっています

```
$dny_idt = 'Li'; # 強化を禁止する略字

$add_iak = 'Ax|Bw|Gv|Kn|Kt|Mi|Sw|Wd'; # $iak を強化する略字
$add_idd = 'Ar|Sh'; # $idd を強化する略字
$add_iad = 'Jw'; # $iak,$idd を強化する略字

$pc_rate = 10; # 強化値に対する価格変動 +n 倍率 (この場合 +1 強化で +10G の付加価値)
@add_idt = (0,0,0,0,1,1,1,2,2,3); # 強化値 (この場合 +3 が 10%、+2 が 20%、+1 が 30% の確立になります、分率は適当で構いません)
```

3. # Sub Find Item # の修正

```
if ($idt ! /GI/) { push(@items,"$inm $idt"); &joinitem('i') }
```

の部分に一行追加します。

```
if ($idt ! /GI/) { require './add_magic.pl'; ($nm,$idt) = &add_magic("$nm","$idt") } # この行
if ($idt ! /GI/) { push(@items,"$nm $idt"); &joinitem('i') }
```

以上で導入は終了です。

至極簡単ですが、リクワイアのファイルパスを間違えないよう注意が必要です。

機能拡張

食料も強化しちゃう

\$add_iak に「Fd」を追加するだけです、チキン +1 の完成です：p

マイナス方向へ強化したい

@add_idt ヘマイナスの数値を設定するだけです。

やりすぎると最低価格がマイナスになります、その辺は良い塩梅に調節してください：)

+5 以上でレアアイテムに

WIZ#2KOD のロングソード +5 ですか、そうですか：)

```
$pc = $add * $pc_rate + $pc;
```

の下に

```
$sts .= 'Li' if $add >= 5;
```

Ultima Online 風に名称変更

```
名称設定
@add_nm_w = ('', ' of Ruin', ' of Might', ' of Force', ' of Power', ' of Vanquishing');
# 武器
@add_nm_a = ('', ' of Defence', ' of Guarding', ' of Harding', ' of Fortification', ' of
Invulnerability'); # 防具
```

名称設定を追加します。

```
$new_inm = "$_[0]" . '+' . "$add";
```

の下へ 2 行追加します。

```
$new_inm = "$_[0]" . '+' . "$add"; # 2 行
$new_inm = "$_[0]" . "$add_nm_w[$add]" if $_[1] = /$add_iak/; # $iak 強化を武器限定とした場合
$new_inm = "$_[0]" . "$add_nm_a[$add]" if $_[1] = /$add_idd/; # $idd 強化を防具限定とした場合
```

マイナス方向へ強化し、かつ名称変更を行う場合、浮動的に演算を行うことで対処しましょう。

```
@add_nm_w = ('-2 の名前 ', '-1', '0', '1', '2', '3');
```

このような場合、最低値である \$add = -2 を、名称付加時のみ \$add = 0 と考えます。

```
$add_ = $add + 2;  
$new_inm = "$_[0]" . "$add_nm_w[$add_]" if $_[1] = /$add_iak/;
```

このようになります。

- ・ Script Of Saga II Worlds Apart でのみ動作確認を行いました
- ・ 機能拡張の方は動作未確認です..... * smile *

おまけ

折角ですので少しお勉強してみましょう。

サブルーチン 1 行目の意味

```
if ($_[1] = /$dny_idt/ || ($_[1] ! /$add_iak/ && $_[1] ! /$add_idd/ && $_[1] ! /$add_iad/)) {  
return($inm,$idt) }
```

これは強化を禁止する略字が入っていた場合、もしくは強化を行うべき略字が入っていなかった場合の双方において真になる記述です。

```
return($inm,$idt);
```

条件式が真の場合に上記 return を行い、戻り値を返してサブルーチン終了になります。

このサブルーチンは通貨以外のアイテム発見時に毎回行われるものですので、上記処理が行われています。

動作的にはリソースの無駄ですが、改造配布という観点で言えば導入が至極容易になります。実際これは 1 行をコピーするだけですからね。

サブルーチン 3 ~ 5 行目の意味

```
$add = $add_idt[int(rand($#add_idt + 1))];  
if ($add == 0) { return($inm,$idt) }
```

@add_idt の配列からランダムに一つ選択し、その値が真であればその値を \$add に代入します。また \$add が 0 であれば return で戻り値を返し、ルーチンを抜けます。\$add が 0 であった場合、強化は行われず、それ以降の処理は無駄になるためです。

サブルーチン 7 行目 ~、int と substr に関するお話

```
$pc = int(substr($_[1],0,4));  
$iak = int(substr($_[1],4,2));  
$idd = int(substr($_[1],6,2));
```

```
$sts = substr($_[1],8,length($_[1]));
```

\$_[1] は引数（これは区別のため ” ひきすう ” と特殊な読みをします）として \$idt が代入されたものです。

\$idt は発見されたアイテムのステータス値ですね。

```
$_[0] = '00010101';
```

とした場合。

```
$pc = int(substr($_[1],0,4));  
$pc = 1;
```

となります。

int 関数で整数化しているためですね。

さて、ここで少し大事なことがあります。

```
$a = int(substr($_[1],0,4));  
$c = $a + 10;  
  
$b = substr($_[1],0,4);  
$d = $b + 10;
```

この場合

```
$c = 11;  
$d = 11;
```

と結果は同じ数値です。

しかし

```
$a = 1; # 数値  
$b = 01; # 文字列
```

と、中途ではこのような差異が出ます。

ここで大事なことは整数化を行っていない方は数値ではなく文字と認識されているということです。

SOSII ではアイテム使用などに関して整数化を行っていない場合が殆どですが、これは整数化を行ってなくても問題が無いからです。

何が問題かといいますと、Perl 演算子は文字列用のもの、数値用のものと区別されているからです。

ここで数値用の演算子、++(インクリメント)などを文字列に対して行っても有効になりません。

これを知っていなければ、自分でルーチンを組んだ時にハマる恐れがあります。

同じように

```
$a = 0;  
$b = 1;  
undef $c;
```

これらの変数はすべて偽と判断されます。

しかし、\$a は数値としての 0、それ以外は文字列としての null です。

ループブロック内で ++ 等のカウントを行っている際、undef などでは初期化すると、それ以降はカウントされなくなります、注意しましょう。

サブルーチン 12 行目の意味

```
$new_inm = "$_[0]" . '+' . "$add"; # 名前変更部分
```

この部分ですね。

右辺中央は顔文字ではありません.....

.(ドット、ピリオド)は文字列同士を連結するのに使います。

またダブルクォーテーションに囲われた文字列は変数の展開やエスケープを行い、シングルクォーテーションに囲われた文字列は行いません。

行うものを行わないものを混同したい場合、上記のように連結すればスマートになるという具合です。

+ をクォートで囲っているのは、これは加算演算子ではなく、ただの文字と解釈させるためです。

```
$new_inm = "$_[0]" . '+' . "$add";  
$new_inm = "$_[0]+$add";
```

残念ながら二つとも意味は全く同じです。

ただし、変数や使用する文字によっては前者で表記せざるをえなくなる場合があります、覚えておいても損は無いでしょう。

このページのコメント

- ・ ちなみに他の改造と非干渉だお - Pse (2007 年 07 月 13 日 22 時 00 分 22 秒)
- ・ おまけを追加したお、知識の足しになれば幸いです ~ - Pse (2007 年 07 月 15 日 01 時 24 分 33 秒)