

プチコンでパズルバトルゲームを作る (4)

●BGM を鳴らす

今は静かにゲームが進んでいますが、音楽(BGM)を鳴らしてみましょう。

まず、プログラムのタイトルを「PUZZLE11」に変えましょう。

```
1 '*** PUZZLE11 ***
2
3 '=== しょき せってい ===
```

BGM を鳴らすには、BGMPLAY(ビージーエムプレイ)命令を使います。バトルが開始する時に、BGM を鳴らしてみましょう。

```
15 '=== バトルかいし ===
16 @BATTLESTART
17
18 ACLS
19 XSCREEN 2
20 BGMPLAY 2
21
22 ML=ML+1
```

2 番の BGM を鳴らす

プログラムを実行してみましょう。バトルが始まると、BGM が流れます。

同じように、ゲームオーバーの BGM が鳴るようにしてみましょう。ゲームオーバーのプログラムで、BGMPLAY 命令を追加します。

```
89 '=== ゲーム オーバー ===
90
91 DISPLAY 0
92 LOCATE 0, 25
93 COLOR 3
94 PRINT "xxx ゲーム オーバー xxx"
95 BGMPLAY 6
96 VSYNC 360
97 COLOR 15
98 END
```

6 番の BGM を鳴らす

待ち時間をのばす

プレイヤーが勝利した時のファンファーレを鳴らしてみましょう。

```

185 '=== プレイヤー-しょうり ===
186 @PLAYERWIN
187
188 DISPLAY 0
189 LOCATE 0,25
190 COLOR 13
191 PRINT "●●● ";MNAME$;"をたおした! ●●●"
192 SPHIDE 100
193 BGMPLAY 5
194 VSYNC 180
195 RETURN

```

5 番の BGM を鳴らす

BGMPLAY 命令の文法は、以下のとおりです。

```

BGMPLAY      0      ,2      ,127
              トラック番号  曲番号      音量
              (0~7)      (0~42)    (0~127)
              ※省略可      ※省略可

```

曲番号は以下のとおりです。スマイルメニュー(SMILE→BGM)でも確認できます。

0	Kung-Fu POP 軽快な曲	11	Chasing at 'Ooedo' 時代劇ゲーム風	22	Calculating SCI	33	Bright blue
1	With stealthy steps 湿った暗い感じの曲	12	Funny land 軽快なマーチバンド風	23	Take Off! シューティング	34	Storytellar
2	Flat out run 緊張感高まる曲	13	Step on the accelerato 激しいロック調	24	The evening moon. パッド	35	Return trip
3	Nostalgia TECHNO 激しくアップテンポな曲	14	Experiment 軽快な曲 その2	25	Sensibility SEN	36	High spirits
4	Feel easy スタートジングル	15	New discovery WOND	26	Pure water ピュア	37	Welcome to party
5	Have a good time クリアジングル	16	Thinking time 考え中	27	Strategy ROA	38	Funky claps
6	Relief ゲームオーバー	17	Mischievous boy WOND2	28	Cure CUR	39	Night surfer
7	Exciting days メニューセレクト	18	Float 未来系	29	Intense battle FIG	40	Ready to FLY
8	Skipping march 結果発表	19	Sound of the surf BAL	30	Keen competition	41	We are heros
9	Valiant departure スタッフロール	20	Sound of the surf2 BAL_2	31	Heat uuuup!	42	Pure water2
10	Important thing スタッフロールその2	21	Spy movie スパイ系	32	Rise with force		

BGM を止めたい時は、BGMSTOP(ビージーエムストップ)命令を実行してください。

●効果音を鳴らす

BGM の次は、プレイヤーとモンスターの攻撃の時に、効果音が出るようにします。
効果音を出すには、「BEEP」(ビーブ)命令を使います。

```
157 '=== プレイヤーのこうげき ===
158 @ATTACK
159
160 DISPLAY 0
161 LOCATE 0, 25
162 COLOR 13
163 PRINT "◆◆◆ プレイヤーのこうげき! ◆◆◆"
164 BEEP 3
165 FOR I=1 TO 3
166   GFILL 168, 68, 231, 131, RGB(255, 0, 0)
167   VSYNC 5
168   GFILL 168, 68, 231, 131, RGB(0, 0, 0)
169   VSYNC 5
170 NEXT
```

3 番の効果音を鳴らす

```
198 '=== モンスターのこうげき ===
199 @MONATTACK
200
201 DISPLAY 0
202 LOCATE 0, 25
203 COLOR 11
204 PRINT "xxx "; MNAME$; "のこうげき! xxx"
205 BEEP 13
206 AP=RND(21)
207 PRINT "ダメージ: "; AP
```

13 番の効果音を鳴らす

●アイテム「コイン」を追加する

数字パネルに、いろいろなアイテムを追加してみましょう。

まずは「コイン」を追加して、「コインをふくめてパネルを選ぶと、攻撃ポイントが 2 倍になる」ルールにします。



まず、プログラムのタイトルを「PUZZLE12」に変えましょう。

```
1 '*** PUZZLE12 ***
2
3 '=== しょき せってい ===
```

パネルをセットするサブルーチンで、乱数でコインを発生させるようにします。

これまでパネル配列変数 P[x,y] に 0~9 の数字を設定していますが、コインを「10」にします。

```
100 '=== パネル セット ===
101 @SETPANEL
102
103 FOR Y=0 TO 4
104   FOR X=0 TO 4
105     P[X, Y]=RND(10)
106     IF RND(100)<5 THEN P[X, Y]=10
107     T[X, Y]=0
108   NEXT
109 NEXT
110 RETURN
```

5%の確率で、P[x,y]をコインにする

パネルを落とすサブルーチンで、新しいパネルを作る部分も同様に改造します。

```
234 '=== パネル おとす ===
235 @FALLPANEL
(中略)
243 'いちばんうえに しんパネル
244   P[FX, 0]=RND(10)
245   IF RND(100)<5 THEN P[FX, 0]=10
246   T[FX, 0]=0
```

5%の確率で、P[x,y]をコインにする

パネルを表示するサブルーチンを改造して、パネル配列変数 P[x,y]が 10 だったらコインを表示するようにします。

```

112 '=== パネル ひょうじ ===
113 @PRTPANEL
114
115 DISPLAY 1
116 FOR Y=0 TO 4
117   FOR X=0 TO 4
118     N=P[X,Y]
119     S=X+Y*5
120     GX=80+32*X
121     GY=32+32*Y
122     SPSET S, 48+N
123     IF N==10 THEN SPSET S, 230
124     SPOFS S, GX, GY
125     SPSCALE S, 2, 2

```

N(=P[x,y])が10だったら
スプライトをコインにする

プログラムを実行してみましょう。パネルにコインが表示されます。

(ランダムなので、表示されない時もあります)

まだ攻撃ポイント計算のプログラムを変えていないので、コインをタッチすると10ポイントの攻撃になるだけです。



スプライトの番号(230)を変えると、アイテムのキャラクターが変わります。

いろいろ試してみましょう。

メインループの、パネルの合計判定で、コインを消した時の処理を追加します。

```

36 '=== メイン ループ ===
37 @MAINLOOP

(中略)

55 'ごうけい はんてい
56 GP=0
57 GS=0
58 AT=0
59 GCOIN=0   コイン合計変数 GCOIN を 0 にリセット
60
61 FOR Y=0 TO 4
62   FOR X=0 TO 4
63     IF T[X, Y]==1 THEN
64       GP=GP+1
65       N=P[X, Y]   パネルの数値が 10 より小さかったら、合計数に加算
66       IF N<10 THEN GS=GS+P[X, Y]
67       IF N==10 THEN GCOIN=GCOIN+1 :
BEEP 7   パネルの数値が 10 だったら
        コインとカウントして効果音を出す
68     ENDIF
69   NEXT
70 NEXT

```

プレイヤーの攻撃サブルーチンで、コインがあった時のポイント計算を変更します。

```

162 '=== プレイヤーのごうげき ===
163 @ATTACK

(中略)

177 AP=GS+(GP-1)*10
178 IF GCOIN>0 THEN AP=AP*2   コインがあったら
                               攻撃ポイントを2倍にする
179 MHP=MHP-AP
180 LOCATE 0, 26
181 COLOR 15
182 PRINT "ダメージ:"; AP; "ポイント"

```

プログラムを実行してみましょう。

コインをふくめてパネルを消すと、攻撃ポイントが2倍になります。



ただこのままだと、コインを2枚以上消しても、攻撃ポイントは2倍にしかありません。

コイン1枚→2倍、コイン2枚→4倍、コイン3枚→8倍…とポイントが倍々に増えていくとおもしろいですね。

ポイントの倍率計算を変えてみましょう。

```

162 '=== プレイヤーのごうげき ===
163 @ATTACK
(中略)
177 AP=GS+(GP-1)*10
178 IF GCOIN>0 THEN AP=AP*POW(2,GCOIN)
179 MHP=MHP-AP
180 LOCATE 0,26
181 COLOR 15
182 PRINT "ダメージ:";AP;"ポイント"

```

コインがあつたら
攻撃ポイントを2のべき乗倍にする

POW(パワー)関数は「べき乗」を計算します。「POW(2,X)」で 2^x (2のx乗、2をx回かける)になります。X=1なら2、X=2なら $2 \times 2 = 4$ 、X=3なら $2 \times 2 \times 2 = 8$ …になります。

(※べき乗は高校の数学で習います)

プログラムを「PUZZLE12」の名前で保存(SAVE)しましょう。

●アイテム「岩」を追加する

2番目のアイテムとして、「岩」を追加します。
パネルを選択できなくする、お邪魔アイテムです。



まず、プログラムのタイトルを「PUZZLE13」に変えましょう。

```
1 '*** PUZZLE13 ***
2
3 '=== しょき せってい ===
```

パネルをセットするサブルーチンで、乱数で岩を発生させるようにします。
パネル配列変数 P[x,y]を「11」にします。

```
100 '=== パネル セット ===
101 @SETPANEL
102
103 FOR Y=0 TO 4
104   FOR X=0 TO 4
105     P[X,Y]=RND(10)
106     IF RND(100)<5 THEN P[X,Y]=10
107     IF RND(100)<10 THEN P[X,Y]=11
108     T[X,Y]=0
109   NEXT
110 NEXT
111 RETURN
```

10%の確率で、P[x,y]を岩にする

パネルを落とすサブルーチンで、新しいパネルを作る部分も同様に改造します。

```
235 '=== パネル おとす ===
236 @FALLPANEL
(中略)
252   'いちばんうえに しんパネル
253   P[FX,0]=RND(10)
254   IF RND(100)<5 THEN P[FX,0]=10
255   IF RND(100)<10 THEN P[FX,0]=11
256   T[X,Y]=0
```

10%の確率で、P[x,y]を岩にする

パネルを表示するサブルーチンを改造して、パネル配列変数 P[x,y]が 11 だったら岩を表示するようにします。

```

116 '=== パネル ひょうじ ===
117 @PRTPANEL
118
119 DISPLAY 1
120 FOR Y=0 TO 4
121   FOR X=0 TO 4
122     N=P[X, Y]
123     S=X+Y*5
124     GX=80+32*X
125     GY=32+32*Y
126     SPSET S, 48+N
127     IF N==10 THEN SPSET S, 230
128     IF N==11 THEN SPSET S, 262
129     SPOFS S, GX, GY
130     SPSCALE S, 2, 2

```

N(=P[x,y])が 11 だったら
スプライトを岩にする

プログラムを実行してみましょう。パネルに岩が表示されます。
(ランダムなので、表示されない時もあります)

まだ他のプログラムを変えていないので、岩をタッチしても何も起こりません。



岩のルールとして、「岩にタッチするとタッチ終了になってしまう」(攻撃に使えない)とします。メインループのタッチのプログラムで、岩にタッチしたらタッチ終了になるようにします。

```

36 '=== メイン ループ ===
37 @MAINLOOP
38
39 TOUCH OUT STTM, TX, TY
40 IF STTM==0 THEN @MAINLOOP
41
42 'タッチ ちゅう
43 WHILE STTM>0
44   PX=(TX-80)/32
45   PY=(TY-32)/32
46   PX=FLOOR(PX)
47   PY=FLOOR(PY)
48   IF PX<0 OR PX>4 THEN BREAK
49   IF PY<0 OR PY>4 THEN BREAK
50   IF P[PX, PY]==11 THEN BREAK
51   T[PX, PY]=1
52   GOSUB @PRTPANEL
53   TOUCH OUT STTM, TX, TY
54 WEND

```

タッチしたパネルが岩だったら
タッチ終了にする

プログラムを実行してみましょう。

岩にタッチすると、そこでタッチ終了になってしまいます。攻撃するパネルとして使えないので邪魔になり、合計を10の倍数にするのが難しくなります。

ただ、このままだと、岩だけをタッチすると「-10ポイントの攻撃」になってしまいます。(パネル0枚・合計数0で攻撃になるため)。それはおかしいので、プログラムを修正しましょう。

```

73 'こうげき
74 IF GS MOD 10==0 AND GP>0 THEN AT=
1:GOSUB @ATTACK

```

合計数が10の倍数、かつ、パネル数が0より大きければ、攻撃する

プログラムを「PUZZLE13」の名前で保存(SAVE)しましょう。

●アイテム「スター」を追加する

3 番目のアイテムとして、「スター」(星)を追加します。

スターをふくめてパネルを消すと、パネル全体がクリアされて新しいパネルになります。



まず、プログラムのタイトルを「PUZZLE14」に変えましょう。

```
1 '*** PUZZLE14 ***
2
3 '=== しょき せってい ===
```

これまでのアイテムと同じように、パネルセットのサブルーチンと、パネルを落とすサブルーチンで、乱数でスターを発生させます。パネル配列変数 P[x,y]は「12」にします。

```
104 '=== パネル セット ===
105 @SETPANEL
106
107 FOR Y=0 TO 4
108   FOR X=0 TO 4
109     P[X, Y]=RND(10)
110     IF RND(100)<5 THEN P[X, Y]=10
111     IF RND(100)<10 THEN P[X, Y]=11
112     IF RND(100)<3 THEN P[X, Y]=12
113     T[X, Y]=0
114   NEXT
115 NEXT
116 RETURN
```

3%の確率で、P[x,y]をスターにする

```
238 '=== パネル おとす ===
239 @FALLPANEL
(中略)
255 'いちばんうえに しんパネル
256   P[FX, 0]=RND(10)
257   IF RND(100)<5 THEN P[FX, 0]=10
258   IF RND(100)<10 THEN P[FX, 0]=11
259   IF RND(100)<3 THEN P[FX, 0]=12
260   T[X, Y]=0
```

3%の確率で、P[x,y]をスターにする

パネルを表示するサブルーチンを改造して、パネル配列変数 P[x,y]が 12 だったらスターを表示するようにします。

```

118 '=== パネル ひょうじ ===
119 @PRTPANEL
120
121 DISPLAY 1
122 FOR Y=0 TO 4
123   FOR X=0 TO 4
124     N=P[X, Y]
125     S=X+Y*5
126     GX=80+32*X
127     GY=32+32*Y
128     SPSET S, 48+N
129     IF N==10 THEN SPSET S, 230
130     IF N==11 THEN SPSET S, 262
131     IF N==12 THEN SPSET S, 226
132   SPOFS S, GX, GY
133   SPSCALE S, 2, 2

```

N(=P[x,y])が 12 だったら
スプライトをスターにする

プログラムを実行してみましょう。パネルにスターが表示されます。

(ランダムなので、表示されない時もあります)

まだ他のプログラムを変えていないので、スターをタッチしても何も起こりません。



メインループの、パネルの合計判定で、スターを消した時の処理を追加します。

```

36  '=== メイン ループ ===
37  @MAINLOOP
    (中略)
56  'ごうけい はんてい
57  GP=0
58  GS=0
59  AT=0
60  GCOIN=0
61  GSTAR=0
62
63  FOR Y=0 TO 4
64    FOR X=0 TO 4
65      IF T[X, Y]==1 THEN
66        GP=GP+1
67        N=P[X, Y]
68        IF N<10 THEN GS=GS+P[X, Y]
69        IF N==10 THEN GCOIN=GCOIN+1 :
        BEEP 7
70      IF N==12 THEN GSTAR=1
71    ENDIF
72  NEXT
73 NEXT

```

スター変数 GSTAR を 0 にリセット

パネルの数値が 12 だったら
スター変数を 1 にする

同じくメインループ内の、タッチ状態をクリアした後で、スターを消していたらパネルをクリアするサブルーチン呼びます。

```

87  'パネルをおとす
88  IF AT==1 AND GSTAR==0 THEN GOSUB
    @FALLPANEL
89
90  'スター ゲット
91  IF AT==1 AND GSTAR==1 THEN GOSUB
    @CLRPANEL
92
93  'モンスターのこうげき
94  GOSUB @MONATTACK

```

スターが無ければ、パネルを落とす。(スターがある時は、この後で
パネル全体がクリアされるので、落とす必要は無い)

スターがあれば、パネルクリアサブルーチンと呼ぶ

プログラムの最後(モンスターデータの前)に、パネルをクリアするサブルーチンを追加します。

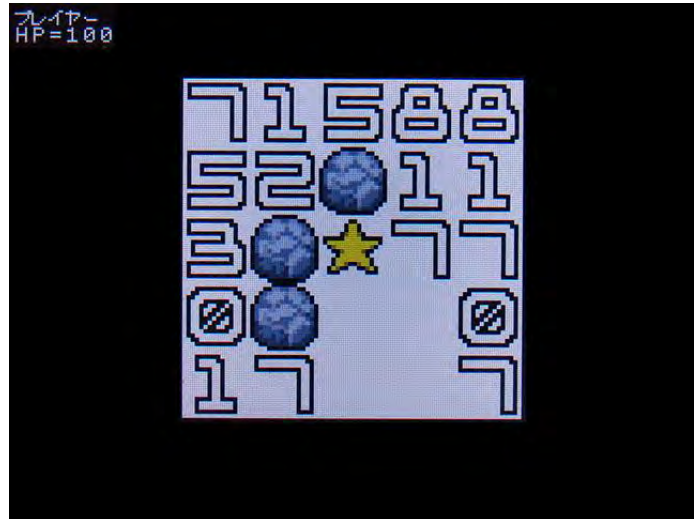
```

244 '=== パネル おどす ===
245 @FALLPANEL
    (中略)
274 NEXT
275 NEXT
276
277 RETURN
278
279 '=== パネル クリア ===
280 @CLRPNEL
281
282 DISPLAY 0
283 LOCATE 0, 25
284 COLOR 13           「パネルクリア」の表示
285 PRINT "☆☆☆ パネル クリア! ☆☆☆"
286 BEEP 5             クリアの効果音を出す
287
288 DISPLAY 1
289 FOR I=1 TO 3       パズルに白の長方形を3回フラッシュ表示
290   GFILL 80, 32, 239, 191, RGB(255, 255,
291     255)
291   VSYNC 10
292   GFILL 80, 32, 239, 191, RGB(0, 0, 0)
293   VSYNC 10
294 NEXT
295
296 GOSUB @SETPANEL    パネルを再設定
297 GOSUB @PRTPANEL    パネルを再表示
298 DISPLAY 0
299 LOCATE 0, 25
300 PRINT " " * 50     「パネルクリア」の表示を消す
301 VSYNC 30
302
303 RETURN
304
305 '=== モンスター データ ===

```

プログラムを実行してみましょう。
 スターを消すと、パズル全体がクリアされて新しいパズルになります。
 例えば岩が増えて手詰まりになった時に、スターでクリアして打開することができます。

ちなみにスターだけをタッチすると、0ポイントの攻撃になってしまいますが、パズルはクリアされます。



プログラムを「PUZZLE14」の名前で保存(SAVE)しましょう。

【できる人は】

自分のアイデアで、いろいろなアイテムを追加してみましょう。以下は一例です。

- 薬びん……………プレイヤーの体力(HP)が回復する。
- 盾(シールド)…直後のモンスターの攻撃をはね返せる。
- ワイルドカード…パネルの合計数が10の倍数でなくても攻撃できる。

など

●体力(HP)をバーで表示する

あとは細かい演出です。

プレイヤーとモンスターの体力(HP)を、これまでは数字で表示していますが、バーで表示してみましょう。

体力がどのくらい残っているか、視覚的にわかりやすくなります。



まず、プログラムのタイトルを「PUZZLE15」に変えましょう。

```
1 '*** PUZZLE15 ***
2
3 '=== しょき せってい ===
```

プレイヤーの HP の設定の設定のプログラムで、別の変数に HP の最初の値をコピーしておきます。

```
15 '=== バトルかいし ===
16 @BATTLESTART
17
18 ACLS
19 XSCREEN 2
20 BGMPLAY 2
21
22 ML=ML+1
23 HP=80+20*ML
24 HP0=HP
25
```

HP の最初の値を HP0 にコピーしておく

その直後で、HP を表示している所で、バーをグラフィックで表示します。

```

26 DISPLAY 1
27 LOCATE 0,0
28 COLOR 15
29 PRINT "プレイヤー"
30 PRINT "HP=" ; HP
31 GFILL 0,16,299,23,RGB(0,255,255)
32
33 GOSUB @BATTLESTART

```

300ドットの長さのバーを表示

プログラムを実行すると、下画面にプレイヤーの HP がバー表示されます。最初は体力が 100%として、横 300ドットの長さのバーを表示しています。



同じように、モンスターの HP もバーで表示してみましょう。

モンスターの設定をするサブルーチンで、HP の最初の値をコピーしておきます。

```

151 '=== モンスター セット ===
152 @SETMONSTER
153
154 M=(ML-1) MOD 5
155 MNAME$=MNA$(M)
156 MS=MSP[M]
157 MHP=100*ML
158 MHP0=MHP
159 RETURN

```

モンスターの HP の最初の値を MHP0 にコピー

モンスターを表示するサブルーチンで、100%のバーを表示します。

```

161 '=== モンスター ひょうじ ===
162 @PRTMONSTER
163
164 DISPLAY 0
165 SPSET 100, MS
166 SPSCALE 100, 4, 4
167 SPOFS 100, 168, 68
168 SPSHOW 100
169 LOCATE 0, 0
170 COLOR 15
171 PRINT "レベル"; ML; " : "; MNAME$
172 PRINT "HP="; MHP
173 GFILL 0, 16, 299, 23, RGB(0, 255, 255)
174 RETURN

```

300ドットの長さのバーを表示

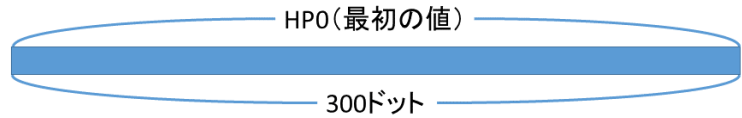
プログラムを実行してみましょう。
上画面にモンスターの HP がバー表示されます。



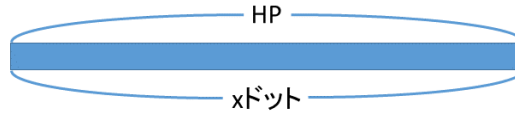
さて、ここからバトルが始まった後、減っていく HP をどうやって表示するかです。
HP が減れば、その分バーを短くして表示しないといけないのですが、どんな長さにすればいいでしょうか？

これは小学校の算数でやる「比例」の問題です。まずプレイヤーの HP で考えましょう。

プレイヤーの HP が最初の値 (HP0) の時は、バーは 300ドットの長さです。



HP が減った時のバーの長さを x ドットとすると、HP の値とドット数の間には、比例関係が成り立つはずですが、



つまり、

$$HP0: HP = 300: x$$

(HP の値の比は、ドット数の比と等しい)

比例の式はそのままわり算に直せますから、

$$HP0 \div HP = 300 \div x$$

両辺に x をかけると、

$$HP0 \div HP \times x = 300$$

両辺に HP をかけると、

$$HP0 \times x = 300 \times HP$$

両辺を HP0 で割ると、

$$x = 300 \times HP \div HP0$$

これで、体力の値が HP の時のバーの長さ(ドット数) x が計算できます。

モンスターから攻撃を受けた時の、プレイヤーの HP 表示に、バー表示を追加します。

```

218 ' === モンスターのこうげき ===
219 @MONATTACK
(中略)
229 DISPLAY 1
230 LOCATE 0, 1
231 COLOR 15
232 PRINT "HP=" ; HP ; "  "
233 X=300*HP/HP0   バーの長さ x を計算
234 GFILL 0, 16, X-1, 23, RGB(0, 255, 255)
235 GFILL X, 16, 299, 23, RGB(0, 0, 0)
236   水色のバーを xドット分表示
237 FOR I=1 TO 3   残りは黒色のバーを表示して消す

```

プログラムを実行してみましょう。
 モンスターからの攻撃を受けると、プレイヤーの HP のバーが減っていきます。



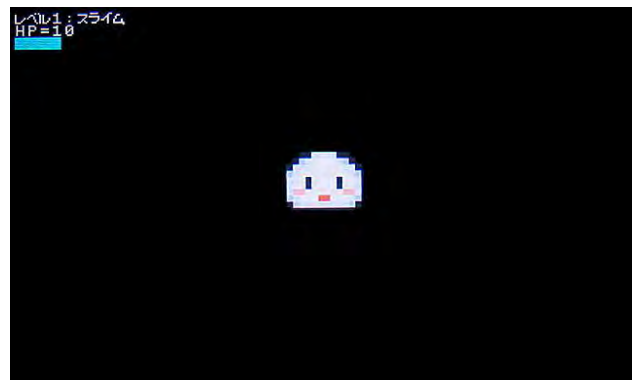
同じように、モンスターの HP のバーも減らしてみましょう。
 プレイヤーの攻撃サブルーチンを改造します。

```

176 '=== プレイヤーのこうげき ===
177 @ATTACK
    (中略)
197 LOCATE 0,1
198 PRINT "HP=";MHP;" "
199 X=300*MHP/MHP0  バーの長さ x を計算
200 GFILL 0,16,X-1,23,RGB(0,255,255)
201 GFILL X,16,299,23,RGB(0,0,0)
202 VSYNC 120
  
```

水色のバーを x ドット分表示
 残りは黒色のバーを表示して消す

プログラムを実行してみましょう。
 プレイヤーが攻撃すると、モンスターの HP のバーが減っていきます。



せっかくなので、HP が減ったらバーの色が変わるようにしてみましょう。
 HP が 20%以下になったら黄色、10%以下になったら赤色で表示してみます。
 バーを表示する GFILL 命令のカラーコードを変数にして、HP の値によって色を変えるようにします。

プレイヤーの攻撃サブルーチンを改造します。

```

176 '=== プレイヤーのこうげき ===
177 @ATTACK
(中略)
197 LOCATE 0,1
198 PRINT "HP=";MHP;" "
199 X=300*MHP/MHP0
200 C=RGB(0,255,255)
201 IF MHP/MHP0<=0.2 THEN C=RGB(255,255,0)
202 IF MHP/MHP0<=0.1 THEN C=RGB(255,0,0)
203 GFILL 0,16,X-1,23,C
204 GFILL X,16,299,23,RGB(0,0,0)
205 VSYNC 120

```

変数 C にカラーコードを設定

MHP が 20%以下なら C を黄色に

MHP が 10%以下なら C を赤色に

カラーコード C でバーを表示

モンスターの攻撃サブルーチンも改造します。

```

224 '=== モンスターのこうげき ===
225 @MONATTACK
(中略)
235 DISPLAY 1
236 LOCATE 0,1
237 COLOR 15
238 PRINT "HP=";HP;" "
239 X=300*HP/HP0
240 C=RGB(0,255,255)
241 IF HP/HP0<=0.2 THEN C=RGB(255,255,0)
242 IF HP/HP0<=0.1 THEN C=RGB(255,0,0)
243 GFILL 0,16,X-1,23,C
244 GFILL X,16,299,23,RGB(0,0,0)
245
246 FOR I=1 TO 3

```

変数 C にカラーコードを設定

HP が 20%以下なら C を黄色に

HP が 10%以下なら C を赤色に

これで、プレイヤーとモンスターの HP のバーが、色が変わるようになります。

このバー表示は、バトルゲームの HP 表示、シューティングゲームのシールドやエネルギー表示、アクションゲームのパワー表示など、いろいろな場面で使えるテクニックです。

プログラムを「PUZZLE15」の名前で保存(SAVE)しましょう。