



プログラムの改造

version 4.3.0 対応版

穴埋め式プログラムの入力で本格的なジャンプアクションゲームを改造します。



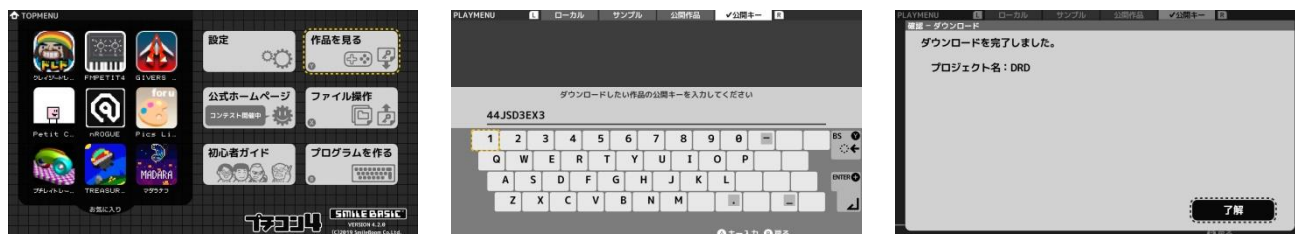
この資料で利用するプロジェクトの公開キー 44JSD3EX3

2020.5.7修正版（フォント：M+ SmileBoom）

準備

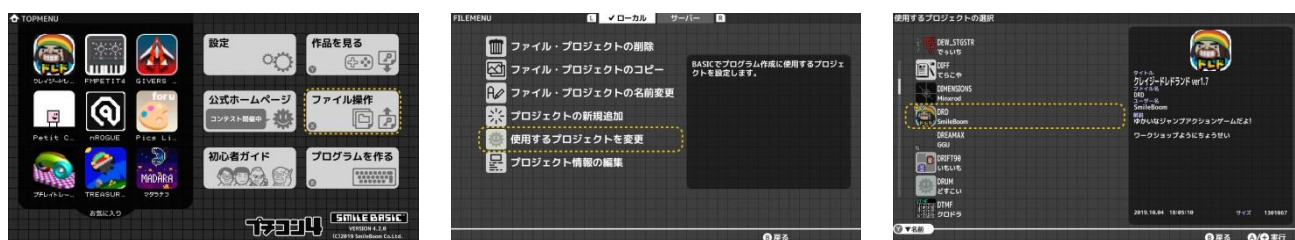
この資料を使ってプログラムの改造を行う前に以下の準備を終わらせてください。

1. 公開キーでテンプレートをダウンロード



TOPMENU → 作品を見る → 公開キー → ダウンロード

2. ダウンロードしたプロジェクトを作業プロジェクトに設定

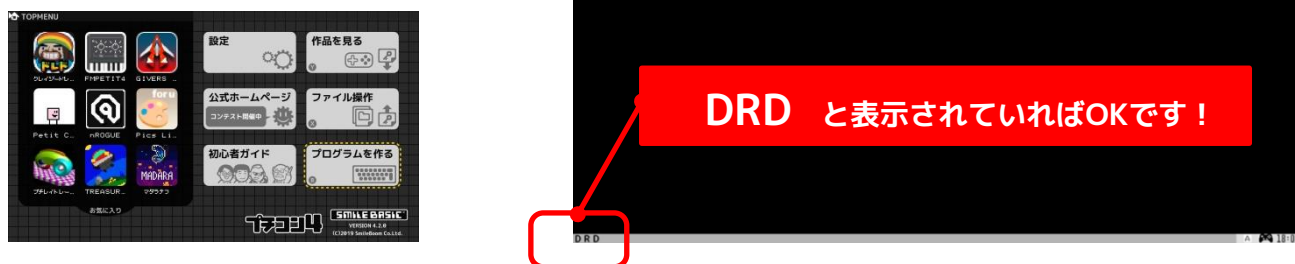


TOPMENU → ファイル操作 → 使用するプロジェクトを変更 → DRD を選択

3. BASICを起動

TOPMENU → プログラムを作る

これで準備は完了です！



テンプレートの読み込み (LOAD)

限られた時間内で作業を行うため、ゼロからすべてを組み込む時間が確保できません。あらかじめゲームのひな型（テンプレート）を用意しておきましたので、最初にこのファイルを読み込みます。

F8キー を押して、コンソール画面から、以下のように入力してください。最後にエンターキーを押してください。

NEW

確認画面が出たら「はい」を選んでください。

LOAD "L00.PRG" ↵

これでプログラムが読み込まれます。

F7キー を押して編集画面を開きプログラムをのぞいてみましょう。



編集画面で見るとプログラムが大量に並んでいます

それなりに遊べる内容のゲームを作るためには、1000行を超えるプログラムになってしまいます。これを最初から作るのは大変なので、今回のワークショップではキャラクターを動かす部分などの分かりやすい処理を実際に記述してもらう方法としました。

操作説明が複雑にならないよう外付けUSBキーボードを利用する前提で進めていますが、ソフトウェアキーボードを使っても利用可能です。

テストプレイ（まだ遊べません）

さきほど読み込んだワークショップ用の参考プログラムを実行してみましょう。

このプログラムは「ドレドくん」を操作して穴に落ちないように右へ進むジャンプアクションゲームです。敵に触ってもダメージを受けるだけで死にません。

F5キー（または+ボタン）を押すと読み込まれたプログラムを実行できます。



実行結果：マップとキャラクターは表示されますが操作できません

残念ながら、このプログラムはそのままではゲームとして遊ぶことはできません。少しずつプログラムを追加して「ドレドくん」が動き回れるように改造しましょう。

早速作業に入ります！

まずは、**F5キー**を押して、動いているプログラムを止めてください。
プログラムが止まったら、**F7キー**（またはⓧ）を押して編集画面を表示させます。

次のページから実際にプログラムを入力します。

この資料の進め方（検索＆追加）

追加するプログラムを入力する場所は検索機能で探します。

- 1) 検索機能ボタン **F3キー**（または ）を押して探す文字を入力しENTERを押す
- 2) 見つかった行の下へカーソルを↓矢印キーで移動し資料を見ながらプログラムを入力
- 3) 全部入力出来たら **F5キー** でプログラムを実行（エラーが出たら修正）

探す文字は、各ページ左上の **L01 :** のように書かれた **Lで始まる番号** です。

文字を探す検索用の文字入力は、 **F3キー** を押すと画面の下に表示されます。



```
000001  ■ドレドくんのアクションゲーム (C) SMILEBOOM CO.LTD.
000002  2017.10.07 NOMAPS2017 (PETITCOM3)
000003  2018.08.22 CEDEC2018 (PETITCOM4)
000004  2018.10.28 NOMAPS2018 (PETITCOM4)
000005  2019.05.14 ドレドくん用に処理修正 (高解像度化)
000006  2019.05.15 BGを使わずにマップ表示、直接数値指定を減らす
000007  2019.05.16 画像を置き換える
000008  2019.05.17 ステージ追加
000009  2019.10.03 CEDEC+SAPPORO向けに調整
000010  画面を初期状態にしたい時は、ACLS (ただし画像も元に戻る)
000011  音を止めたい時は、SNDSTOP
000012  カレントプロジェクトを変更してから作業すること
000013  OPTION STRICT
000014  CONST #DEB=0
000015  CONST #VER$="VER1.0"
000016  --- SCREEN
000017  ACLS #SCRN_W=1280, #SCRN_H=720
000018  XSCREEN #SCRN_W, #SCRN_H
000019  TSCREEN 4, 16, 16
000020  TSCREEN 0, 16, 16
000021  --- TPAGE
000022  CONST #TPAGE_INFO=0
000023  --- GPAGE
000024  CONST #GPAGE_GRP=0
000025  CONST #GPAGE_SP=4
000026  CONST #GPAGE_BG=2
000027  GTARGET #GPAGE_GRP
000028  SPPAGE #GPAGE_SP
000029  --- SPRITE LINK
000030  CONST #LINK_RIT=8R1111
000031  検索:
000032  ↓:次 ↑:前 Esc:終了
```

検索用の入力機能

```
0 2 4 6 8 10 12 14
検索: L01
PRGO L00.PRG
```

探したい文字列を入力してENTERキーを押すと
見つけた行へジャンプします。検索から抜ける
時は、 **ESCキー** を押します。

見つからない時は矢印キーの上や下を押してみてください。

```
000237 CALL SPRITE
000238 ' --- ↓L01A: マップスクロール
000239 '
000240 ' --- ↓L02A: スコア
000241 '
000242 WEND
```

L01：マップをスクロールさせる

最初に「ドレドくん」が動いた時にマップをスクロールさせるプログラムを追加します。

◎変更する場所を探す

F3キーを押して **L01** を入力しEnterキーで検索します。（239行目付近）

見つからない時は矢印キーの↑上や↓下を押してみてください。

<入力前>

```
0232  ' --- メインループ↓
0233  ISLOOP=#LP_GAMEMAIN↓
0234  WHILE ISLOOP==#LP_GAMEMAIN↓
0235      VSYNC 1↓
0236      GETBUTTON #CID↓
0237      CALL SPRITE↓
0238      ' --- ↓L01A:マップスクロール↓
0239      ↓ ← ここに入力
0240      ' --- ↓L02A:スコア↓
0241      ↓
0242  WEND↓
```

◎文字を追加する（239行目付近）

検索して発見された ' --- ↓L01A:マップスクロール の下の行にある、赤矢印 ← **ここに入力** の行に、MAPSCROLL BGX, BGY と書きます。

<入力後>

```
0238  ' --- ↓L01A:マップスクロール↓
0239  MAPSCROLL BGX, BGY ↓ ← 入力が終わった
```

◎テストプレイ **F5キー**

実行はできますが・・・

今回の追加ではボタンを押してもマップはスクロールしません。スクロールさせるためのプログラムを追加するという作業を体感してもらいました。あまり気にしないで次に進みます。（プレイヤーが動けばスクロールも始まります）



◎テキストのポイント

さきほど追加した MAPSCROLL BGX,BGY は、「BGXとBGYの数値だけ、マップをスクロール（MAPSCROLL）させる」という意味です。ですが、プチコン4にMAPSCROLLという命令が用意されているわけではありません。MAPSCROLL は“プログラムを組んで”作られた命令なのです。プログラムの先のほう、1479行目から1508行目までを見てください。

DEFという命令を使って、
「MAPSCROLLという名前の命令を作ります」
という宣言をしています

```
'=====
'■MAP SCROLL
'=====
DEF MAPSCROLL X,Y
VAR L,T=-16
'--- Calculate the visible part from given coordinates
IF SPCHK(#SP_BG) THEN RETURN
'---
X=MAX(-#CHR_W*3,MIN((MAP_W*#CHR_W)-#SCRN_W),X-(#SCRN_W/2))
Y=MAX(-#CHR_H*5,MIN((MAP_H*#CHR_H)-#SCRN_H,Y-(#SCRN_H/2))
IF Y>#MAP_H-#CHR_H THEN Y=#MAP_H-#CHR_H
Y=Y-(#MAP_H-(#CHR_H*2))
PUTMAP X,Y
SPANIM #SP_BG,"XY",T,-X,-Y,1

'### Background is an independent loop structure

VAR OX,OY
L=#SCRN_W/4
'--- Mountain
SPOFS #SP_MT OUT OX,OY
SPANIM #SP_MT,"XY",T,-(L+X*0.1),OY,1
'--- Grass (back)
SPOFS #SP_GR0 OUT OX,OY
SPANIM #SP_GR0,"XY",T,-(L+X*0.2),OY,1
'--- Waves (back)
SPOFS #SP_GR1 OUT OX,OY
SPANIM #SP_GR1,"XY",T,-(L+X*0.3),OY,1
END
```

ここは「MAPSCROLL」という命令の中身になります。
プログラム中、「MAPSCROLL」と書かれていたら、
コンピューターは指定されたXとYの値を使ってここに書かれた内容を実行します。

「マップをスクロールさせる」といった処理は何度もプログラムの中で使う可能性があります。そのたびに同じ内容を書いているとプログラムがどんどん長くなってしまいます。そのため、「MAPSCROLL」という命令を作って、その命令をプログラムの中で使うようにしています。

L02 : スコアを表示する

次にスコアを表示するプログラムの呼び出しを追加します。

L02 を **F3キー** で検索して  **ここに入力** の行を入力します。(241行目付近)

検索した ' ---  **L02A:スコア** の下に PUTSCORE を追加します。

<入力後>

```
0232 ' --- メインループ
0233 ISLOOP=#LP_GAMEMAIN
0234 WHILE ISLOOP==#LP_GAMEMAIN
0235     VSYNC 1
0236     GETBUTTON #CID
0237     CALL SPRITE
0238     ' ---  L01A:マップスクロール
0239     MAPSCROLL BGX,BGY
0240     ' ---  L02A:スコア
0241     PUTSCORE  ここに入力
0242 WEND
```

◎テストプレイ **F5キー**

実行すると上画面にスコアなどが表示されました。少しゲームっぽい画面になりました。



これ以降の作業も同じようにプログラムを追加してテストプレイを繰り返します。

SC	スコア
HI	ハイスコア
SCADD	スコアに加算する値

L03 : 左スティックで左右に歩く

いよいよ「ドレドくん」を移動するプログラムを追加します。

L03 を **F3キー** で検索して **ここに入力** の行を入力します。(434行目付近)

```

0434 ' --- ↓L03A: 横向き壁チェック
0435 IF MAPHIT(DRDPX+DRDVX, DRDPY, DRDSX) THEN
0436 ' --- ↓L03B: 壁にぶつかった
0437 DRDPX=(DRDPX DIV #CHR_W)*#CHR_W
0438 ' --- ↓L03C: 向きで補正するサイズを調整(左右方向で座標補正)
0439 DRDPX=DRDPX+(#CHR_W-1)*(DRDDIR==1)
0440 ' --- ↓L03D: 壁判定ココまで
0441 ELSE
0442 ' --- ↓L03E: 壁なく進む
0443 DRDPX=DRDPX+DRDVX
0444 ' --- ↓L03F: 横向き判定ココまで
0445 ENDIF
0446
  
```

◎テストプレイ **F5キー**

実行すると、左スティックで「ドレドくん」を左右に移動することができるようになり、端に近づくとスクロールも発生します。

- ・リンゴは出ていますが取れません
- ・障害物を越えられない



飛び出した床は、ジャンプがないと越えられないようです。

DRDPX, DRDPY	ドレドくんの座標
DRDVX, DRDVE	ドレドくんの移動量
DRDDIR	ドレドくんの向き (1=右, -1=左)
BGSW	1=マップの右端に到着した
MAPHIT	足元や壁などのマップ状態を調べる
ヒント	DRDPXにDRDVXを足す倍率

L04 : Yボタンで加速する

Yボタンを押した時に「ドレドくん」がダッシュするプログラムを追加します。

L04 を **F3キー** で検索して **ここに入力** の行を入力します。(418行目付近)

検索で見つからなかった場合は、矢印の↑上を押して他の場所を探してください。

```

0418 ' --- ↓L04A:ダッシュボタンが押されているか?
0419 IF BUTTON(0,#B_DUSH) THEN ←ここに入力
0420 ' --- ↓L04B:変化する場合は走るアニメへ
0421 VM=8.0:IF V THEN DRDNM$="RUN" ←ここに入力
0422 ' --- ↓L04C:速度に応じて傾ける
0423 IF STVX THEN SPANIM #SP_DRD,"R",-15,A,1 ←ここに入力
0424 ' --- ↓L04D:放されているとき角度を戻す
0425 ELSE ←ここに入力
0426 ' --- ↓L04E:角度を戻すためのアニメ設定
0427 SPANIM #SP_DRD,"R",-8,0,1 ←ここに入力
0428 ' --- ↓L04F:角度を戻すココまで
0429 ENDIF ←ここに入力

```

◎テストプレイ **F5キー**

Yボタンを押しながらスティック操作で傾きながら「ドレドくん」が走ります。

これで移動は速くなりました。

まだリンゴは取ることができません。



BUTTON	ボタンの取得
DRDNM\$	ドレドくんのアニメーション名
STVX	スティック左右の変化量 (±1.0)
#SP_DRD	ドレドくんのスプライト管理番号
SPANIM	スプライトのアニメ設定命令
ヒント	VM、Aの初期値

L05 : フルーツを取ってスコアアップ

「ドレドくん」がフルーツや敵に触れた時の当たり判定プログラムを追加します。

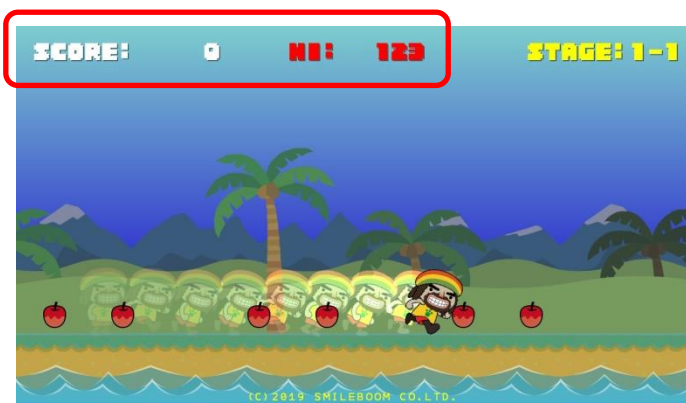
L05 を **F3キー** で検索して **ここに入力** の行を入力します。(468行目付近)

```

0468 ELSE
0469 ' --- ↓L05A: フルーツを取ったらスコア加算
0470 SCADD=SCADD+SPVAR(H, "SCORE")
0471 ' --- ↓L05B: フルーツを消して音を出す
0472 SPCLR H: BEEP 7
0473 ' --- 敵とフルーツの判定ここまで
  
```

◎テストプレイ **F5キー**

実行して移動するとリングに触った時にリングが消えてスコアが加算されます。



DRDDMG	ダメージを受けた時の停止時間
DRDOLD\$	直前に設定されたドレドくんアニメ名
SPHITSP	スプライト同士の当たり判定
SPVAR()	スプライトの内部変数を取得する命令
OBJDEAD	マップから生成した物体の消去
SPCLR	スプライトの消去
BEEP	効果音を鳴らす

L06 : Bボタンでジャンプする

Bボタンを押した時に「ドレドくん」がジャンプするプログラムを追加します。

L06 を **F3キー** で検索して **ここに入力** の行を入力します。(338行目付近)

```

0338 ' --- ↓L06A:ジャンプしていない?
0339 IF DRDJUMP==0 THEN ←ここに入力
0340 ' --- ↓L06B:ジャンプボタンが押されたら
0341 IF BUTTON(#CID,#B_JUMP,2) THEN ←ここに入力
0342 ' --- ↓L06C:ジャンプ開始
0343 DRDJUMP=1:DRDNM$="JUMP":BEEP 8 ←ここに入力
0344 ' --- ↓L06D:押している時間でジャンプ高さ変化用のカウンタ
0345 DRDCNT=26 ←ここに入力
0346 ' --- ↓L06E:ジャンプ初速
0347 DRDVY=-8 ←ここに入力
0348 ' --- ↓L06F:ジャンプ開始ココまで
0349 ENDIF ←ここに入力
0350 ' --- ↓L06G:ジャンプ判定ココまで
0351 ENDIF ←ここに入力
0352 ' --- ↓L06H:ジャンプ中?
0353 IF DRDJUMP==1 THEN ←ここに入力
0354 ' --- ↓L06I:押してる時間ジャンプし続けている?
0355 IF DRDCNT==0 THEN ←ここに入力
0356 ' --- ↓L06J:普通のジャンプ中
0357 DRDVY=DRDVY+0.5 ←ここに入力
0358 ' --- ↓L06K:頂点を越えた?
0359 IF DRDVY>0 THEN DRDNM$="FALL" ←ここに入力
0360 ' --- ↓L06L:ジャンプ中じゃない
0361 ELSE ←ここに入力
0362 ' --- ↓L06M:ジャンプが長くカウンタを減らす
0363 DRDCNT=DRDCNT-1 ←ここに入力
0364 ' --- ↓L06N:ジャンプボタンが離されていたら落下へ
0365 IF BUTTON(#CID,#B_JUMP)=0 THEN DRDCNT=0 ←ここに入力
0366 ' --- ↓L06O:落下判定ココまで
0367 ENDIF ←ここに入力
0368 ' --- ↓L06P:ジャンプココまで
0369 ENDIF ←ここに入力

```

◎テストプレイ **F5キー**

Bボタンでジャンプします。押し続けている時間で高さが変化し、Yボタンで加速しながらジャンプで飛距離が伸びます。

DRDJUMP	0=ジャンプ前、1=ジャンプ中
DRDCNT	押してる時間で高さを調整するカウンタ
DRDSTOP	1=ドレドくん強制停止
ヒント1	DRDCNT、DRDVYの初期値
ヒント2	DRDVYジャンプ中の加算値



L07 : カニを動かす

敵の1つ「カニ」の移動プログラムを追加します。

L07 を **F3キー** で検索して **ここに入力** の行を入力します。(723行目付近)

```

0123 ' --- ↓L07A:足元チェック(床の上?) ↓
0124 C=MAPHIT(X+VX,Y+1,8) ←ここに入力
0125 ' --- ↓L07B:床が無い? ↓
0126 IF C==0 THEN ←ここに入力
0127 ' --- ↓L07C:床が無い時は移動方向反転 ↓
0128 VX=-VX ←ここに入力
0129 ' --- ↓L07D:床判定ココまで ↓
0130 ELSE ←ここに入力
0131 ' --- ↓L07E:横壁チェック ↓
0132 C=MAPHIT(X+VX,Y,24) ←ここに入力
0133 ' --- ↓L07F:壁があるか? ↓
0134 IF C THEN ←ここに入力
0135 ' --- ↓L07G:壁にぶつかった ↓
0136 X=(X DIV #CHR_W)*#CHR_W ←ここに入力
0137 ' --- ↓L07H:移動方向でサイズを調整 ↓
0138 X=X+(#CHR_W-1)*(VX==1) ←ここに入力
0139 ' --- ↓L07I:座標補正 ↓
0140 X=X-(VX*(24/2)) ←ここに入力
0141 ' --- ↓L07J:移動方向反転 ↓
0142 VX=-VX ←ここに入力
0143 ' --- ↓L07K:壁判定ココまで ↓
0144 ELSE ←ここに入力
0145 ' --- ↓L07L:壁なしに進める ↓
0146 X=X+VX ←ここに入力
0147 ' --- ↓L07M:壁ナシ判定ココまで ↓
0148 ENDIF ←ここに入力
0149 ' --- ↓L07N:床ナシ判定ココまで ↓
0150 ENDIF ←ここに入力
0151 ' --- 移動処理ココまで ↓

```

◎テストプレイ (F5キー)

カニが左右に動いているはず・・・

カニがいない！！

カニを出したい時は、次のページを見てマップ上にカニを配置してください

X,Y	座標管理
C	マップ情報
VX	横の移動量



L08 : マップパーツにフルーツや敵を配置

修正する部分 **L08** を **F3キー** で検索します。

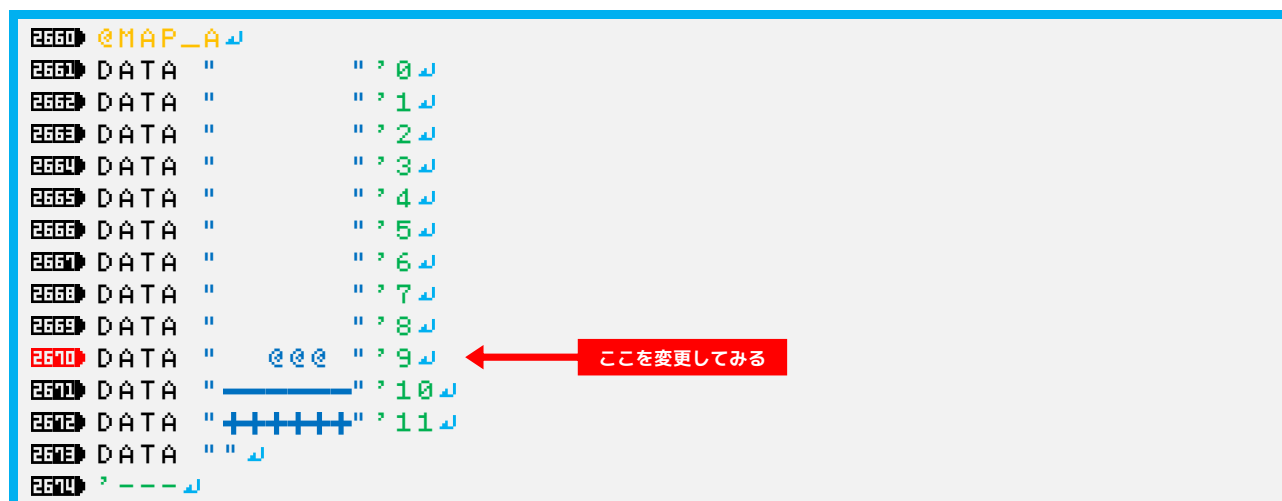
ドレドランドのマップは、縦に細いマップ（これをマップパーツと呼びます）を並べることで長いマップを表現しています。マップパーツはA～Zまで用意することができて、横方向は長めに伸ばせますが、高さは12段分しか持てません。文字や記号の組み合わせで作られているので、地面の形を変えたり、敵やアイテムの配置を調整することができます。

壁記号：  (金属板)

敵果物： K (カニ) S (スカンク) * (栗) @ (リンゴ) & (ぶどう)

試しに、以下のようにマップ部品へ記号を追加してみてください。

INSキー を使って上書きモードで入力すると文字列がずれないように入力できます。



※これは参考なので好きなように好きなものを置いてください

◎テストプレイ **F5キー**

実行すると、マップ配置した敵やフルーツが表示されるようになります。マップの最後に「部品Z」を配置して置けば、やる気のなさそうなボスとの戦いも楽しめます。

L09 : マップの長さの変更

修正する部分 **L09** を **F3キー** で検索します。

L08で改造したマップパーツを並べるためのデータがL09以降に並んでいます。ステージごとのマップの長さを調整したいときは、このデータを変更してください。

```
2810 ' =====
2811 ' ステージデータ (↓L09A)
2812 ' =====
2813 ' DATA BGMPLAY用の曲番号
2814 ' DATA "マップの順番を並べる文字列"
2815 ' =====
2816 @STAGE99_99
2817 DATA 33 'Bright blue(BGM)
2818 DATA "GAAF"
2819
2820 @STAGE0_0
2821 DATA 46 'WAKUWAKU :D adventure
2822 DATA "GAAAHY" ← ここを変更してみる
2823 @STAGE0_1
2824 DATA 46 'WAKUWAKU :D adventure
2825 DATA "GAAFGACAADBY"
2826 @STAGE0_2
2827 DATA 46 'WAKUWAKU :D adventure
2828 DATA "GAAABACADAFAFEGBAAY"
2829 @STAGE0_3
2830 DATA 48 'WAKUWAKU >X adventure
2831 DATA "GAAAZ"
```

◎ステージ1のマップの長さを増やす

```
2820 @STAGE0_0
2821 DATA -1
2822 DATA "GAAAGAAAGAAAGAAAY" ← 入力が終わった
```

2822行目の DATA "GAAAHY" がマップパーツを並べている部分です。マップパーツは A~Z までの部品を登録できるので、登録されているアルファベットを並べることでマップを長くできます。存在しない文字は入るとエラーになります。

この内容を、 DATA "GAAAGAAAGAAAGAAAY" とすれば、最初のマップが4倍ぐらい長いマップに変化します。

さらに改造するためのヒント！

これでプログラムの入力は一通り終わりました。ここからは他の人とは違うゲームにするためのチューニングに入ります。改造のためのヒントをここにまとめておきます。

◎ボスが絶対に勝つのはバグじゃないか？

どうやったら勝てるのかは、プログラムを調べてみましょう。怪しい部分がないかな？

```
1068 ' --- 6:最終結果↵
1069 IF MD==6 THEN↵
1070 ' --- あとだし↵
1071 IF DRDNM$=="GOO" THEN JAN$="PA"↵
1072 IF DRDNM$=="CHOKI" THEN JAN$="GOO"↵
1073 IF DRDNM$=="PA" THEN JAN$="CHOKI2"↵
1074 ' --- スリいプログラム↵
1075 IF ZURU==0 THEN JAN$=JTBL$[RND(3)]↵
1076 ' ---↵
1077 NX=SPVAR(SP,"PANEL")↵
1078 SPOFS NX OUT U,V↵
1079 SETANIM NX,"KUMA"+JAN$,U,V,#Z_BOSSPNL↵
1080 C=0↵
1081 IF DRDNM$=="GOO" && JAN$=="PA" THEN C=1↵
1082 IF DRDNM$=="CHOKI" && JAN$=="GOO" THEN C=1↵
1083 IF DRDNM$=="PA" && JAN$=="CHOKI2" THEN C=1↵
1084 IF INSTR(JAN$,DRDNM$)>-1 THEN C=1↵
1085 ' --- 俺様の勝ち？↵
```

◎常識を超えたスーパーマンにしたい

ジャンプの高さ、走る速度、走ってるときの傾き方・・・この辺りを変更すると速すぎて落ちまくる操作難易度の高い主人公も作れます。移動やジャンプのプログラムを書いている時に、色々な数字が見えていたと思います。その辺りの数値を適当に大きくしたり小さくしたりすると・・・どうなるかな？

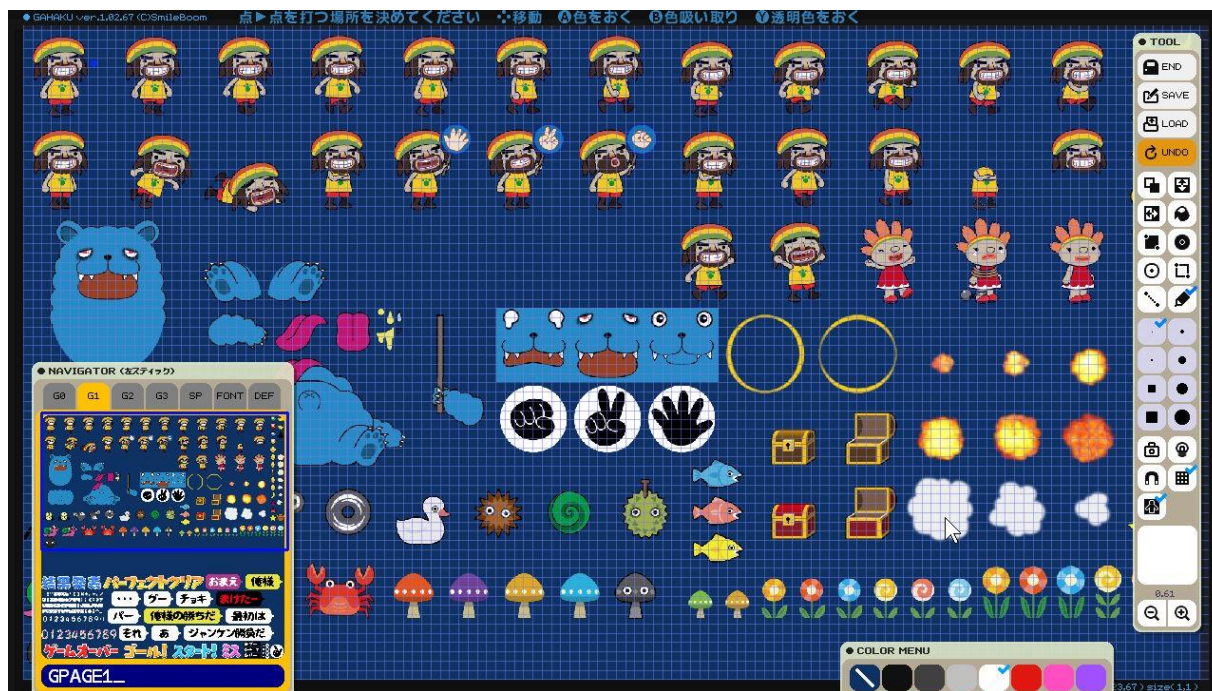
◎効果音やBGMを自分の好きなものに変更したい

効果音は、BEEPという命令で発声してます。音楽は、BGMPLAYという命令で演奏しています。検索を使えばどこで使われているかを探すのは簡単です。

F9キーを押すとプチコンに内蔵されている効果音や音楽の番号を調べるソフトが起動します。これを使ってよい音を探してみましょう。

GAHAKUを使って画像を改造する

クレイジードレランドを実行中に **F10キー** を押すとプチコンに内蔵されているお絵描きツール「GAHAKU」が起動します。このツールを使うと現在動いているゲームの中で使われている画像を直接書き替えることができます。（マウスで操作します）



- 1) NAVIGATORウィンドウのG1とG2のタブで画像を切り替えて絵を描く
- 2) 絵が描けたらG1かG2もタブを切り替える（切り替えることで画像が更新される）
- 3) SAVEボタンでファイル名を変えずに保存（両方描いたらそれぞれをSAVEする）
- 4) CTRL+C（または+ボタン）でGAHAKUを強制終了
- 4) ドレランドに戻る（描き替えた内容が反映されているはず・・・）

一度記録するとSAVEされた画像でドレランドが楽しめます。



この資料は、ここでおしまいです。

どんどん改造して、オリジナリティあふれる面白いゲームを作ってみましょう！

おつかれさまでした！