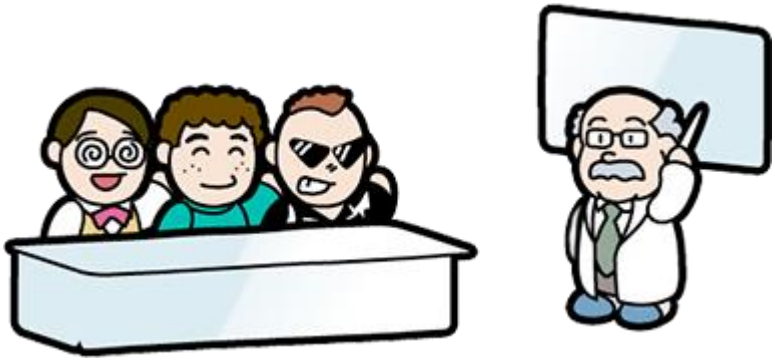
プチコン4
SMILE BASIC®

ジャンプアクションの
プログラムを作る

version 2018.8

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作ります。

SmileBoom



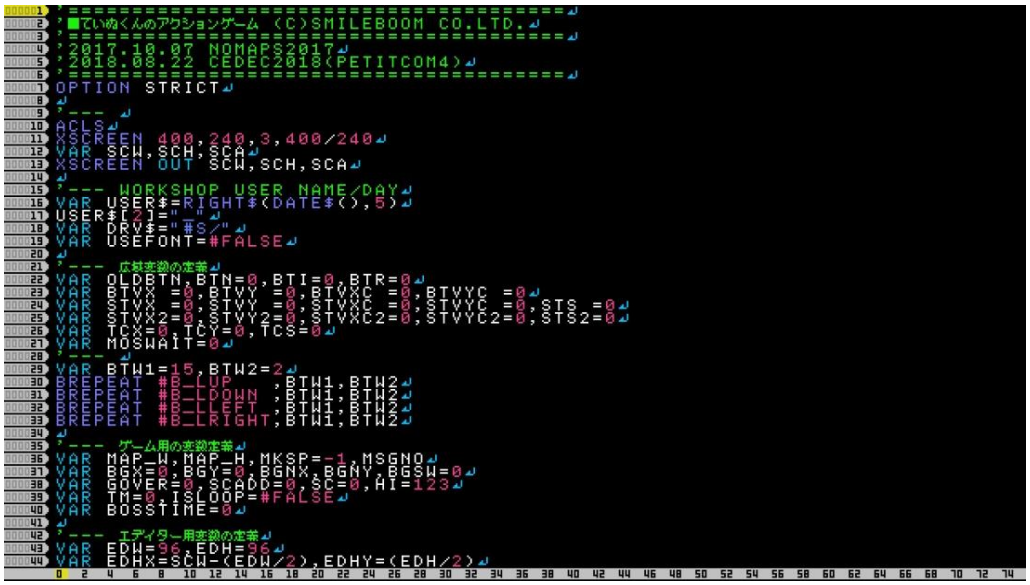
テンプレートの読み込み（LOAD）

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 ↵
LOAD "#S/L00AUG22" ↵
```

これでプログラムが読み込まれます。

F9 キー を押して編集画面を開きプログラムをのぞいて見ましょう。



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 行ぐらいの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。

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、「ていぬくん」を操作して敵に触れないように右へ進むジャンプアクションゲームです。

F5 キー を押してください。現在読み込まれているプログラムが実行されます。

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。
このワークショップで少しずつプログラムを追加して「ていぬくん」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、 F9 キー 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。

L01：マップをスクロールさせる

プレイヤーキャラクター（ていぬくん）が動いた時にマップをスクロールさせるためのプログラムを追加します。追加するのは、L01 のコメントがある行です。1 行しか追加しないので間違わずに入力してみましょう。

単語を検索する場合は、 F3 キー 押します。
探したい文字列を入力してエンターキーを押すと見つけた行へジャンプします。

◎追加する前の状態

”L01”で検索して修正する場所へカーソルを移動します。

```
'---- メインループ
ISLOOP=1
WHILE ISLOOP
 GET_BUTTON #CID
 CALL SPRITE
 '←L01A（マップスクロール
 '←L02A（スコア
 VSYNC 1
WEND
```

◎追加した後の状態（124 行目付近）

赤枠内の、`'←L01A（マップスクロール` の前に `MAPSCROLL BGX,BGY` を追加します。

```
'---- メインループ
ISLOOP=1
WHILE ISLOOP
 GET_BUTTON #CID
 CALL SPRITE
 MAPSCROLL BGX,BGY '←L01A（マップスクロール
 '←L02A（スコア
 VSYNC 1
WEND
```

◎テストプレイ（ F5 キー ）

F5 キー を押すことでプログラムを実行することができますが・・

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しょう。（プレイヤーが動くようになればスクロールも始まります）

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。
L01と同じように1行だけ書き込んでください。L01で入力した行の下です。

◎追加する前の状態

”L02”で検索して修正する場所へカーソルを移動します。

```
'---- メインループ
ISLOOP=1
WHILE ISLOOP
 GET_BUTTON #CID
 CALL SPRITE
 MAPSCROLL BGX,BGY 'ﾍL01A(マップスクロール
 'ﾍL02A(スコア
 VSYNC 1
WEND
```

◎追加した後の状態（125 行目付近）

赤枠内の、`'ﾍL02A(スコア` の前に `SCORE_PUT` を追加します。

```
'---- メインループ
ISLOOP=1
WHILE ISLOOP
 GET_BUTTON #CID
 CALL SPRITE
 MAPSCROLL BGX,BGY 'ﾍL01A(マップスクロール
 SCORE_PUT 'ﾍL02A(スコア
 VSYNC 1
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。

SC	スコア
HI	ハイスコア
SCADD	スコアに加算する値

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。

L03：十字キーで左右に歩く

ここからは、変更後の状態だけを記載します。
修正する部分（”L03”）を F3キー で検索して入力します。

◎追加した後の状態（309 行目付近）

赤枠内の、’←L03A(左右　の前に以下のようにプログラムを追加します。
入力に関係のない行は、青色にしています。

```
TEVX=V
'--- 横向きの壁チェック
IF MAPHIT(TEPX+TEVX,TEPY,TEW) THEN  '←L03A(左右
 '--- 壁にぶつかった
 TEPX=(TEPX DIV 32)*32  '←L03B(左右
 TEPX=TEPX+(31*(TEDIR==1))  '←L03C(左右
 TEPX=TEPX-(TEDIR*(TEW/2))  '←L03D(左右
ELSE  '←L03E(左右
 '--- 壁なし(進める)
 TEPX=TEPX+TEVX  '←L03F(左右
ENDIF  '←L03G(左右
'--- マップの終わりチェック(横)
IF TEPX<TEW THEN TEPX=TEW
```

◎テストプレイ（ F5 キー ）

実行すると十字キーで「ていぬくん」を左右に移動する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
・右端まで進むとスクロールが止まります
・左側に行くと「ていねくん」が消えます

TEPX , TEPY	ていぬくんの座標
TEVX, TEVY	ていぬくんの移動量
TEDIR	ていぬくんの向き （1＝右、-1＝左）
TEW, TEH	ていぬくんの横幅と高さ
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
	SPANIMの角度

L04：Y ボタンで加速する

修正する部分（"L04"）を F3キー で検索して入力します。
（前のページで入力した部分の少し上にあります）

◎追加した後の状態（294 行目付近）

赤枠内の、'←L04A(ダッシュ　の前に以下のようにプログラムを追加します。
入力に関係のない行は、**青色**にしています。

```
'--- 押されている間ダッシュ
VAR VM=1.5,A=30*TEDIR
IF TESTOP==0 THEN '←L04A(ダッシュ
 IF BUTTON(0,#B_DUSH) THEN '←L04B(ダッシュ
  VM=TEVXM:IF V THEN TENM$="RUN" '←L04C(ダッシュ
  IF STVX THEN SPANIM #SP_TEINU,"R",-15,A,1 '←L04D(ダッシュ
 ENDIF '←L04E(ダッシュ
ENDIF '←L04F(ダッシュ
'--- 放されているとき角度を戻す
IF BUTTON(#CID,#B_DUSH)==0 THEN '←L04G(ダッシュ
 SPANIM #SP_TEINU,"R",-8,0,1 '←L04H(ダッシュ
ENDIF '←L04I(ダッシュ
IF V<-VM THEN V=-VM
IF V> VM THEN V= VM
TEVX=V
'--- 横向きの壁チェック
```

◎テストプレイ（ F5 キー ）

実行すると Y ボタンを押しながら左右十字キーで「ていぬくん」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
TENM$	ていぬ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_TEINU	ていぬ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
実験要素	VM、Aの初期値
	SPANIMの角度

L05：フルーツを取ってスコアアップ

修正する部分（"L05"）を F3キー で検索して入力します。
（前のページで入力した部分の少し下にあります）

◎追加した後の状態（323 行目付近）

赤枠内の、'←L05A(フルーツ の前に以下のようにプログラムを追加します。

```
'--- マップの終わりチェック(横)
IF TEPX<TEW THEN TEPX=TEW
IF TEPX>MAP_W*32-200 THEN BGSW=1
IF TEPX>MAP_W*32-TEW THEN TEPX=MAP_W*32-TEW
'--- 敵やフルーツに当ったか?
VAR H=SPHITSP(#SP_TEINU) '←L05A(フルーツ
IF H!=-1 && H!=#SP_MOUSE THEN '←L05B(フルーツ
 IF SPVAR(H,"HP") THEN '←L05C(フルーツ
  '--- 相手の上に落下?
  IF TEVY>0 THEN '←L05D(フルーツ
   OBJDEAD H '←L05E(フルーツ
  ELSE '←L05F(フルーツ
   IF TEOLD$!="MISS" THEN BEEP 64 '←L05G(フルーツ
   TENM$="MISS":TEDMG=8 '←L05H(フルーツ
  ENDIF '←L05I(フルーツ
 ELSE '←L05J(フルーツ
  SCADD=SCADD+SPVAR(H,"SCORE") '←L05K(フルーツ
  SPCLR H:BEEP 7 '←L05L(フルーツ
 ENDIF '←L05M(フルーツ
ENDIF '←L05N(フルーツ
END
```

◎テストプレイ（ F5 キー ）

実行して移動するとリンゴに触ったときにリンゴが消えてスコアが加算されます。

TEDMG	ダメージを受けた時の停止時間
TEOLD$	直前に設定されたていぬくんアニメ名
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

BEEP命令で簡単に効果音を鳴らすことができます。BEEPの後に続く数字を適当な数字に変更すると変な音が鳴って面白くなるかもしれません。BEEPに指定できる音の番号は、0～133までです。さらに続けて値を指定すると、周波数やパンポットなどの値も調整できます。詳しくはインラインヘルプをご覧ください。

L06：B ボタンでジャンプする

修正する部分（”L06”）を F3キー で検索して入力します。

◎追加した後の状態（224 行目付近）

赤枠内の、’◆Ｌ０６Ａ（ジャンプ　の前に以下のようにプログラムを追加します。
ちょっと多いですね。

```
'◎ていねくん：ジャンプと落下のプログラム
'---------------------------------------
DEF JUMP_TEINU
'--- ジャンプしてない？
IF TEJUMP==0 && TESTOP==0 THEN '◆L06A（ジャンプ
 '--- ジャンプボタンが押されたら
 IF BUTTON(#CID,#B_JUMP,1) THEN '◆L06B（ジャンプ
  TEJUMP=1:TENM$="JUMP":BEEP 8 '◆L06C（ジャンプ
  TECNT=16 '◆L06D（ジャンプ
  TEVY=-4 '◆L06E（ジャンプ
 ENDIF '◆L06F（ジャンプ
ENDIF '◆L06G（ジャンプ
'--- ジャンプ中？
IF TEJUMP==1 THEN '◆L06H（ジャンプ
 '--- 押してる時間ジャンプし続けている？
 IF TECNT==0 THEN '◆L06I（ジャンプ
  '--- 普通のジャンプ中
  TEVY=TEVY+0.5 '◆L06J（ジャンプ
  IF TEVY>0 THEN TENM$="FALL" '◆L06K（ジャンプ
 ELSE '◆L06L（ジャンプ
  '--- 押している間上昇
  TECNT=TECNT-1 '◆L06M（ジャンプ
  IF BUTTON(#CID,#B_JUMP)==0 THEN TECNT=0 '◆L06N（ジャンプ
 ENDIF '◆L06O（ジャンプ
ENDIF '◆L06P（ジャンプ
'--- 上下移動
IF TEVY THEN TEPY=TEPY+TEVY
```

◎テストプレイ（ F5 キー ）

実行して B ボタンを押すとその場でジャンプします。押し続けている時間で高さが変化する処理にも対応しています。Y ボタンで加速しながらジャンプすると飛距離が伸びます。

TEJUMP	0=ジャンプ前、１＝ジャンプ中
TECNT	押してる時間で高さを調整するカウンタ
TESTOP	1＝ていねくん強制停止
実験要素	TECNT、TEVYの初期値
	TEVYジャンプ中の加算値

L07：カニを動かす

修正する部分（"L07"）を F3キー で検索して入力します。

◎追加した後の状態（460 行目付近）

赤枠内の、'←L07A(カニ　の前に以下のようにプログラムを追加します。

```
ELSEIF MD==1 THEN
 '--- 足元チェック(床の上?)
 C=MAPHIT(X+VX,Y+1,8) '←L07A(カニ
 IF C==0 THEN '←L07B(カニ
  VX=-VX '←L07C(カニ
 ELSE '←L07D(カニ
  '--- 横壁チェック
  C=MAPHIT(X+VX,Y,24) '←L07E(カニ
  IF C THEN '←L07F(カニ
   '--- 壁にぶつかった
   X=(X DIV 32)*32 '←L07G(カニ
   X=X+(31*(VX==1)) '←L07H(カニ
   X=X-(VX*(24/2)) '←L07I(カニ
   VX=-VX '←L07J(カニ
  ELSE '←L07K(カニ
   '--- 壁なし(進める)
   X=X+VX '←L07L(カニ
  ENDIF '←L07M(カニ
 ENDIF '←L07N(カニ
ENDIF
'---
```

◎テストプレイ（ F5 キー ）

カニのいる場所まで移動してみましょう。カニは左右に動き出したかな？

X,Y	座標管理
C	マップ情報
VX	横の移動量

L08：マップを長くする

修正する部分（”L08”）を F3キー で検索します。

◎追加した後の状態（115 行目付近）

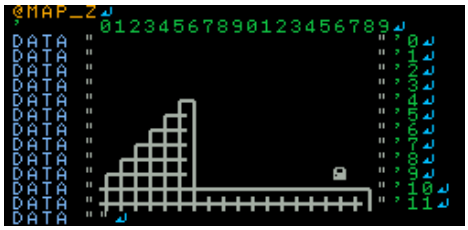
ていぬくんのマップデータはマップを小さい単位で区切ってマップ部品とし、それらを並べることで横に長いマップを実現しています。この行にある "AAABACADAAEBAAZ" は、マップ部品ごとに付けられたアルファベットA～Zまでの記号を文字列として並べることでマップを作っています。Zは特別に最後のボスが出る場所として定義されています。

以下のようにA～E、およびZを使って適当に文字列を登録してください。

```
'---- ステージ初期化
MAPSET "AAABACADAAFEGBAAZ"  'マップ
BGMPLAY 12
SETMSG 34 'スタート
GOSUB @INIT_TEINU
```

◎マップの構造（790 行目付近、”@MAP_A”で検索）

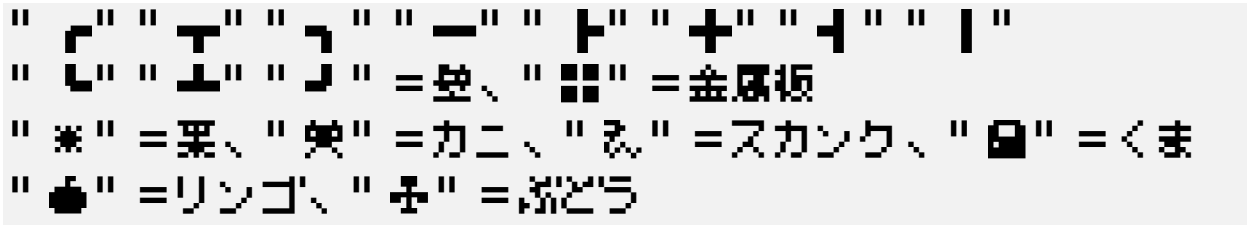
マップ部品（A～Zの名前を付けて管理）は以下のように文字の組み合わせで地形を表現しています。現在のプログラム内には、A～EおよびZが登録されています。この部品は任意に増やすことができます。



L09：マップにフルーツや敵を配置

修正する部分（”L09”）を F3キー で検索します。

マップ全体の長さ（マップ部品の並び順）は115行目付近の MAPSET 命令に渡す引数で指定しますが、マップ上に物や壁などを配置する時は、マップ部品そのものを変更します。
これらの記号は、キーボードから記号を選択して入力することができます。



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。

```
@MAP_A
'       0123456789012345
DATA "                "'0
DATA "                "'1
DATA "                "'2
DATA "                "'3
DATA "                "'4
DATA "                "'5
DATA "                "'6
DATA "                "'7
DATA "                "'8
DATA "   ●  ●  ●    "'9
DATA "━━━━━━━━"'10
DATA "++++++++"'11
DATA ""
```

```
@MAP_B
'       0123456789012345
DATA "                "'0
DATA "     ■  ■       "'1
DATA "     ■  ■       "'2
DATA "        ●╋      "'3
DATA "       ■■■■  ■   "'4
DATA "             ■  "'5
DATA "   ■         ■  "'6
DATA "       ※       "'7
DATA "      ┌┐  ┌┐    "'8
DATA "      |+|  |+|┌┐ "'9
DATA "━━━━━++|  |++┼┼━"'10
DATA "+++++++|  |++++++"'11
DATA ""
```

※あくまでも例として配置しているだけなので、好きなように好きなものを置いてもかまいません

◎テストプレイ（ F5 キー ）

実行すると、L08 で指定したマップ部品がマップとして利用されていることが確認できます。また、L09 でマップ部品に敵やフルーツを追加している場合は、それらも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。

これでアクションゲームの要素は一通り組み込まれました。

L88：タイトル画面の表示

修正する部分（”L88”）を F3キー で検索して入力します。

◎追加した後の状態（112 行目付近）

赤枠内の、’◆L88A(タイトル　の前に以下のようにプログラムを追加します。

```
@LOOP
RESET
'--- タイトル
TITLE '◆L88A(タイトル
'◆L99A(ツール
'--- ステージ初期化(◆L08A
MAPSET "AAABACADAAFEGBAAZ" 'マップ
BGMPLAY 12
SETMSG 34 'スタート
GOSUB @INIT_TEINU
'--- メインループ
ISLOOP=1
WHILE ISLOOP
```

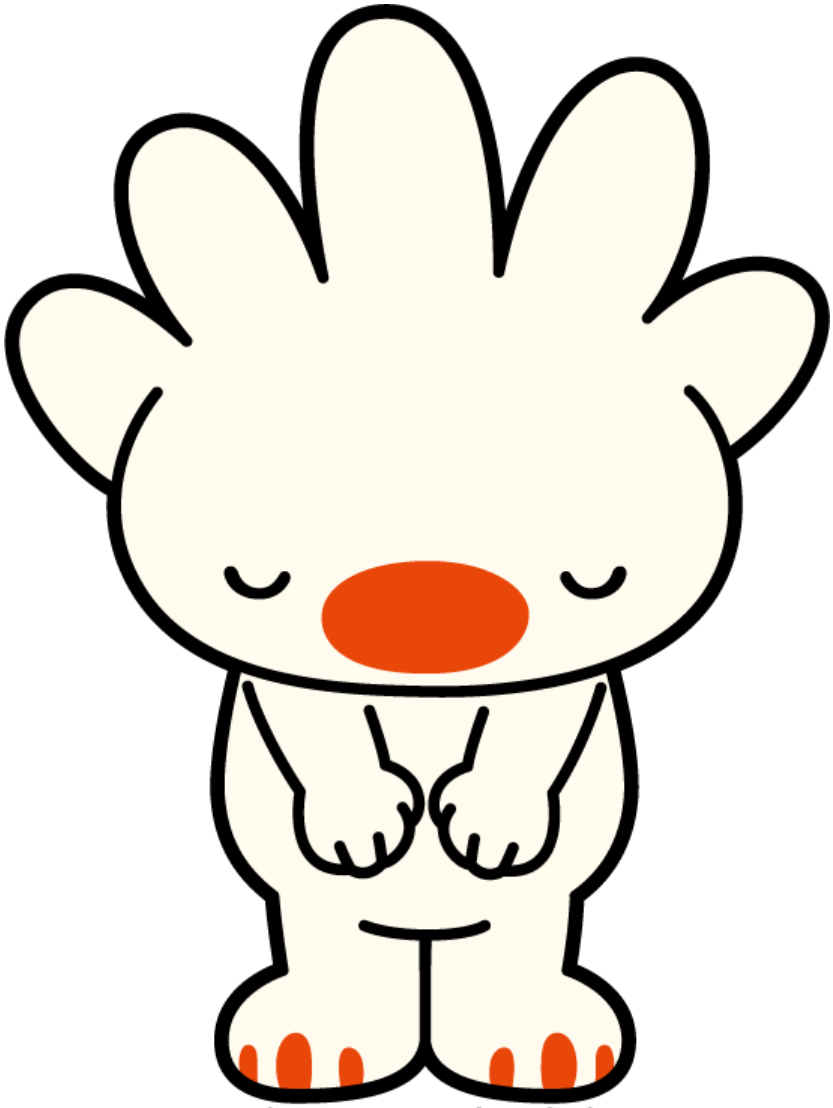
◎テストプレイ（ F5 キー ）

実行するとゲームのマップが表示される前にタイトル画面が表示されます。
ここで A ボタンを押すとゲームが始まります。

これですべての作業が終わりです。

でも・・なんだかボスには勝てないですよね・・
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。

おつかれさまでした！



ていぬくんは、札幌市手稲区のマスコットキャラクターです。
イベント用に利用申請を行い画像データをお借りしました。

ご協力ありがとうございます。