

ニンテンドー3DSで プログラム作り

NoMaps 2017 (2017年10月7日)

対象年齢	小学4年生以上 ※英字キーボードによる入力が必要なため保護者の方が付き添われる場合は低学年も受付可能です
時間	1回目 13:15～ 2回目 15:15～ 各回 60分 (状況により前後する可能性があります)
定員	各回 10名
参加料金	無料 ※ニンテンドー3DSは人数分ご用意しております
お申し込み	不要 ※当日、会場1階の受付にて整理券をお受け取りください (先着順)

目次

このワークショップの目的.....	3
◎用意するもの.....	3
◎ワークショップの内容に関するご質問について	3
◎ていぬくんについて	3
プチコンの使い方 (1/3) TOP MENU 編.....	4
プチコンの使い方 (2/3) BASIC 編.....	5
プチコンの使い方 (3/3) キーボード編	6
テンプレートの読み込み (LOAD)	7
作業用のファイルを保存 (SAVE)	8
テストプレイ	9
L01 : マップをスクロールさせる	10
◎追加する前の状態.....	10
◎追加した後の状態 (47 行目付近)	10
◎テストプレイ (START ボタン)	10
L02 : スコアを表示する	11
◎追加する前の状態.....	11
◎追加した後の状態 (48 行目付近)	11
◎テストプレイ (START ボタン)	11
L03 : 十字キーで左右に歩く.....	12
◎追加した後の状態 (206 行目付近)	12
◎テストプレイ (START ボタン)	12
L04 : Y ボタンで加速する	13
◎追加した後の状態 (194 行目付近)	13
◎テストプレイ (START ボタン)	13
L05 : フルーツを取ってスコアアップ.....	14
◎追加した後の状態 (222 行目付近)	14
◎テストプレイ (START ボタン)	14
L06 : B ボタンでジャンプする	15
◎追加した後の状態 (122 行目付近)	15
◎テストプレイ (START ボタン)	15
L07 : マップを長くする.....	16
◎追加した後の状態 (38 行目付近)	16
◎マップの構造 (672 行目付近、"@MAP_A"で検索)	16
L08 : マップにフルーツや敵を配置.....	17
◎テストプレイ (START ボタン)	17
L99 : タイトル画面の表示.....	18
◎追加した後の状態 (206 行目付近)	18
◎テストプレイ (START ボタン)	18
参考資料.....	19

このワークショップの目的

身近なゲーム専用機ニンテンドー3DS 上で動く「プチコン 3 号 SmileBASIC」を使って、ゲームのプログラムはどのように書かれているのかを実際のプログラムを見て改造しながら体験します。他の人のプログラムを見ることで、プログラムの書き方や考え方を学習します。

◎用意するもの

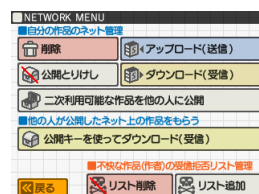
ワークショップ会場では以下の機材等は用意されています。

- ・ニンテンドー3DS 本体とダウンロードソフト「プチコン 3 号 SmileBASIC」
ニンテンドーeShop の検索から「プチコン」で調べると見つかります。
いくつか候補出てきますが「プチコン 3 号 SmileBASIC」を選択して購入してください。
価格は、税込み 500 円です。

- ・ワークショップ用に用意されたテンプレートプロジェクト

公開キー：**V 3 2 N Q 4 H M**

上記公開キーをプチコン 3 号の TOPMENU にある「作品公開とダウンロード」のボタンから NETWORK MENU に入り、「公開キーを使ってダウンロード（受信）」のボタンを押して、事前にダウンロードしておいてください。



◎ワークショップの内容に関するご質問について

弊社公式 WEB サイトのお問い合わせページよりご連絡ください。

<https://smileboom.com/contact/>

◎ていぬくんについて

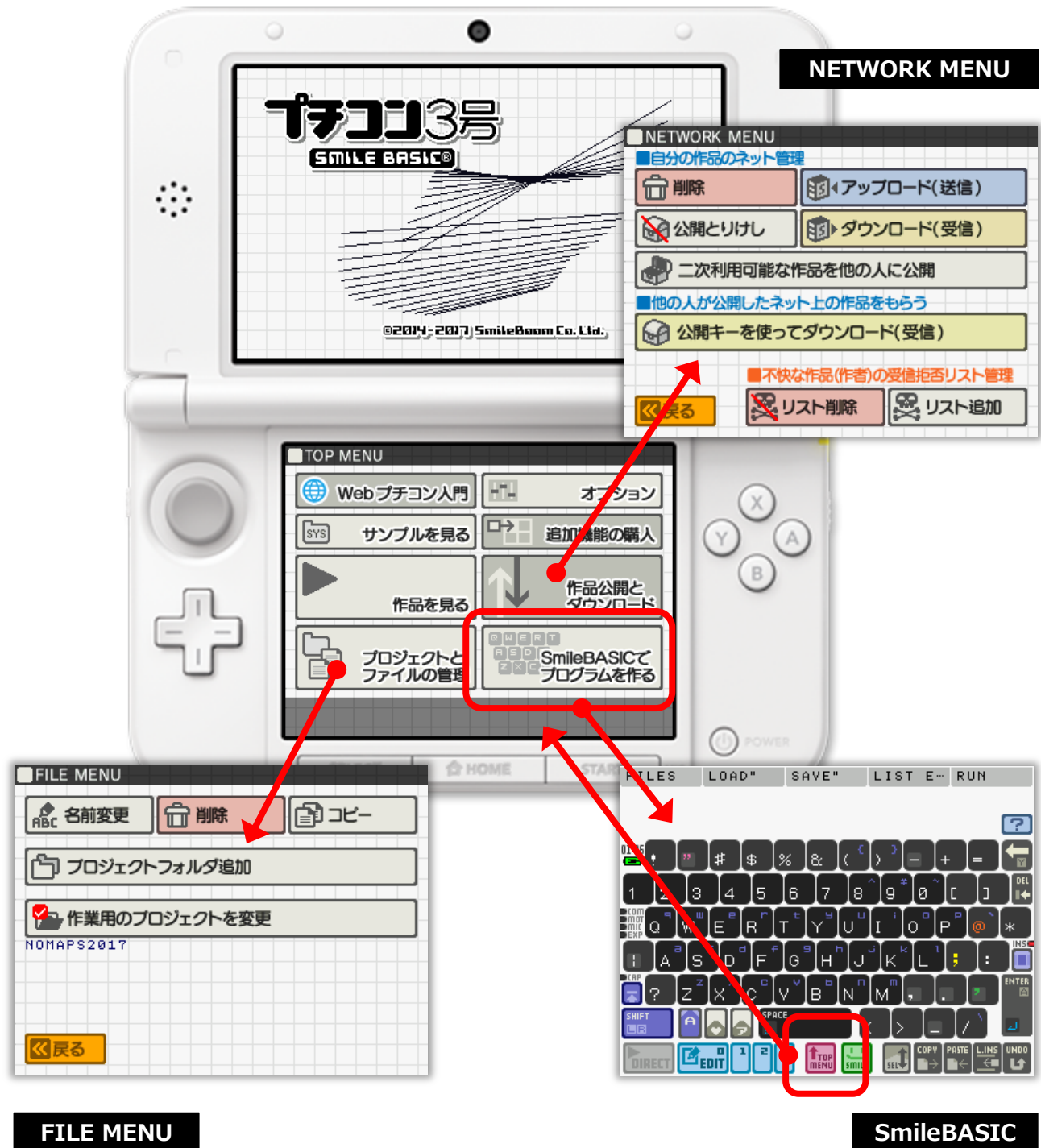
札幌市手稲区のマスコットキャラクターです。NoMaps のイベント用に利用申請を行い画像データをお借りました。画像データを再利用する場合は、手稲区の許可が必要になる可能性があります。

©TEINE WARD SAPPORO



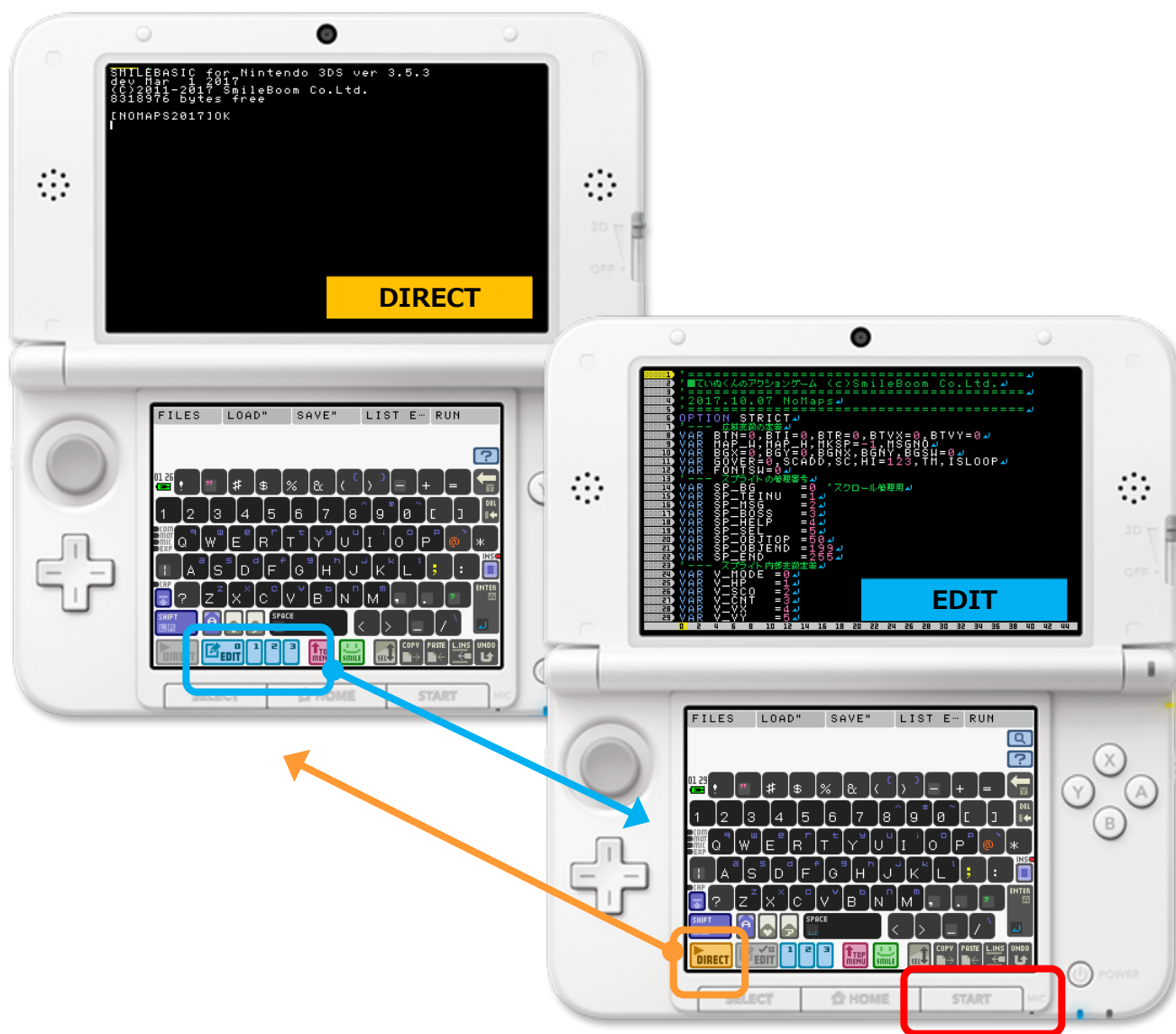
プチコンの使い方 (TOP MENU)

ニンテンドー3DSのHOMEメニューからプチコン3号 SmileBASIC を起動すると以下のような画面が表示されます。この画面をTOP MENU (トップメニュー) と呼びます。プログラムを作る場合には下画面右下にある「SmileBASIC でプログラムを作る」ボタンをタッチします。



プチコンの使い方 (BASIC)

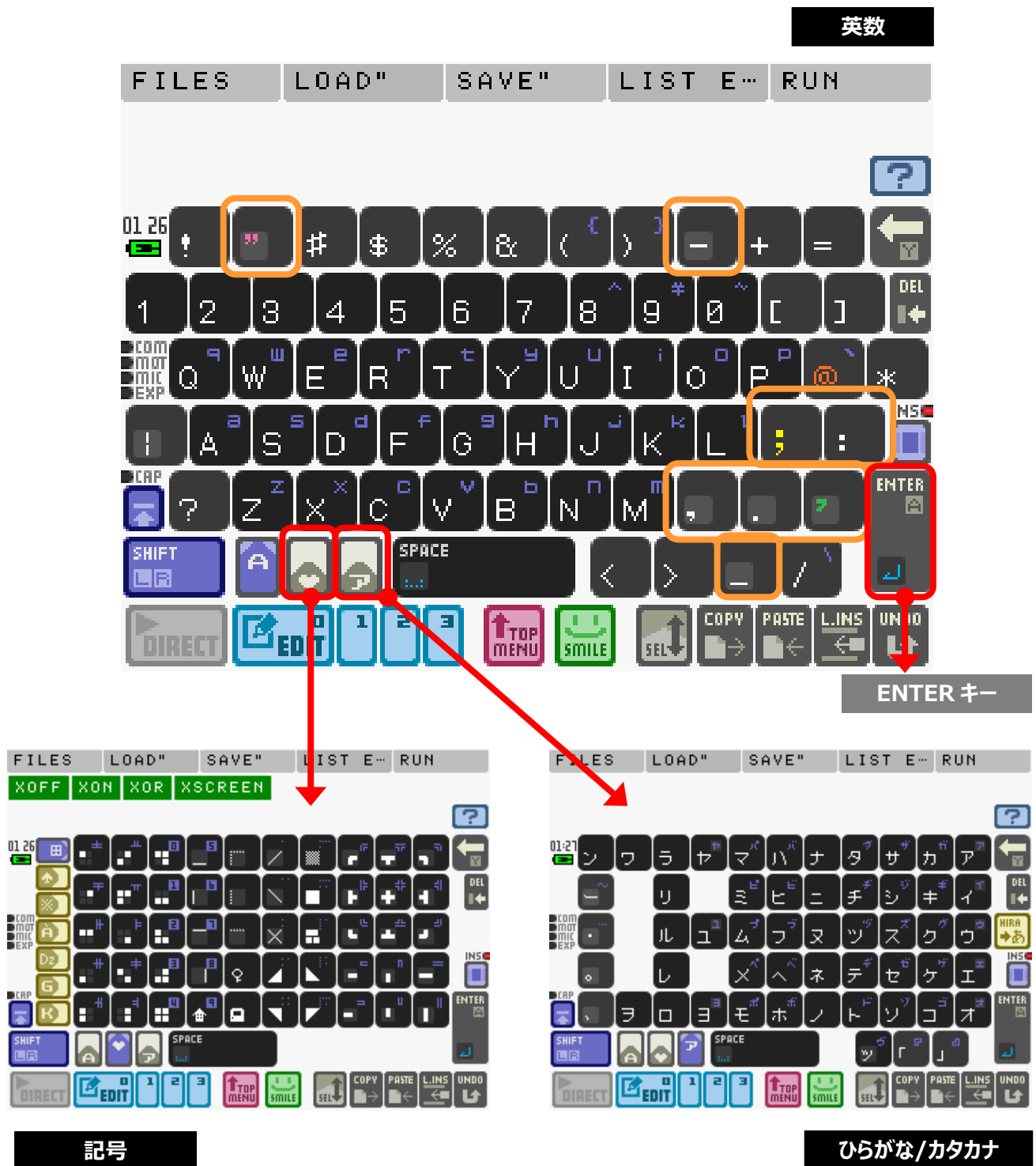
TOP MENU から SmileBASIC に入ると、下画面にキーボードが表示されます。上画面に編集画面が無い状態を **DIRECT** (ダイレクト) モードと呼び、上画面に編集画面が表示されている状態を **EDIT** (エディット) モードと呼びます。プログラムを実行するときは **DIRECT** モードから行き、編集するときは **EDIT** モードから行います。それぞれの切り替えは下画面キーボード左下の**オレンジ色のボタン**とその隣の**水色のボタン**から行います。



編集画面から START ボタンを押すことで直接プログラムを実行することができます。
プログラム実行中に START ボタンを押すとプログラムは途中で停止します。

プチコンの使い方（キーボード）

パソコンのようにキーボードが付いていないニンテンドー3DS でプログラムを書くため、プチコンには、ソフトウェアキーボードが搭載されています。プログラムで使われる記号には似ているものが多いため、プチコンのソフトウェアキーボードではキーを色分けして間違いにくくしています。適当に色々なボタンを押してみましょう。



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。DIRECT モードの画面から、以下のように入力してください。最後に ENTER キーを押してください。

```
LOAD "L01" ↵
```

これでプログラムが読み込まれます。

水色の EDIT ボタンを押して編集画面を開きプログラムを見て見ましょう。

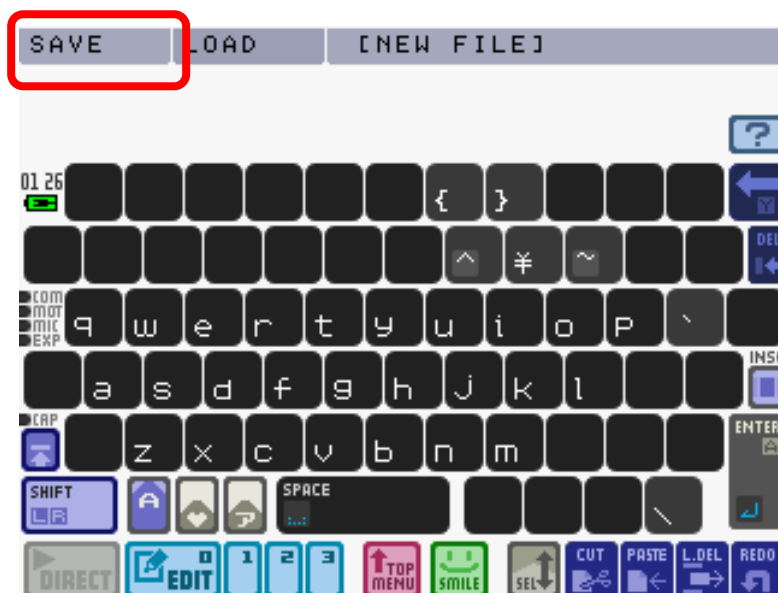
A screenshot of a text editor window with a black background and green text. The editor shows a large amount of code, including variable declarations and assignments. The code is organized into sections with labels like 'OPTION STRICT', 'スプライトの管理番号', and 'スクリール管理用'. The editor has a line number column on the left (000001 to 000029) and a column number indicator at the bottom (0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44).

編集画面で見るとプログラムが大量に並んでいます

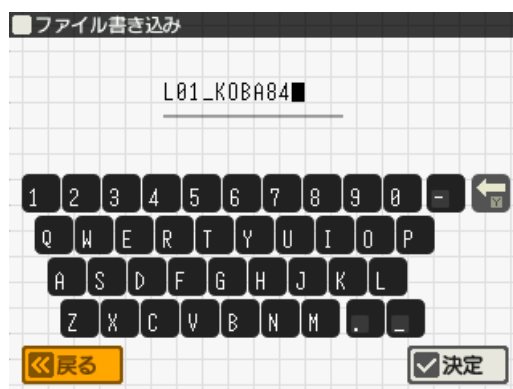
それなりに遊べる内容のゲームを作るためには、1000 行ぐらいのプログラムになってしまいます。これを最初から作るのは大変なので、今回のワークショップではキャラクターを動かす部分などの分かりやすい処理を実際に記述してもらおう方法としました。

作業用のファイルを保存（SAVE）

これから作業を行う前に、自分で作ったプログラムとして記録するため、名前を付けて SAVE（保存）しておきましょう。ソフトウェアキーボードの SHIFT ボタンを押すか、ニンテンドー3DS の L ボタンを押すと以下のように画面が変化します。



この画面に表示されている下画面左上の「SAVE」をタッチしてください。SAVE するためのダイアログが開いてファイル名を入力することができます。



L01 の直後に （アンダーライン）をつけて、その後ろに名前を英文字で入れます。文字数が限られているので、適当に短く省略した名前を使ってください。これで保存は完了です。この先、新しくプログラムを追加したら定期的に保存してください。

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。

ニンテンドー3DS のスタートボタンを押してください。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。

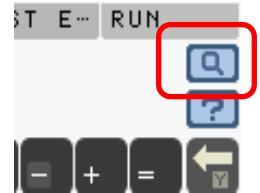
まずは、START ボタンを押して動いているプログラムを止めてください。

プログラムが止まったら、水色の EDIT ボタン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、下画面のキーボード上にある虫眼鏡アイコンを押します。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。



◎ 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
CALL SPRITE
? + L01<スクロール>
? + L02<マップ>
VSYNC 1
WEND
? ---
GOTO @LOOP
```

◎ 追加した後の状態（47 行目付近）

赤枠内の、? + L01<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
CALL SPRITE
MAPSCROLL BGX, BGY ? + L01<スクロール>
? + L02<マップ>
VSYNC 1
WEND
? ---
GOTO @LOOP
```

◎ テストプレイ（START ボタン）

START ボタン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
  CALL SPRITE
  MAPSCROLL BGX, BGY ? + L01(スクロール)
  ? + L02(スコア)
  VSYNC 1
WEND
```

◎追加した後の状態 (48 行目付近)

赤枠内の、 ? + L02(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
  CALL SPRITE
  MAPSCROLL BGX, BGY ? + L01(スクロール)
  SCORE_PUT ? + L02(スコア)
  ? + L02(スコア)
  VSYNC 1
WEND
```

◎テストプレイ (START ボタン)

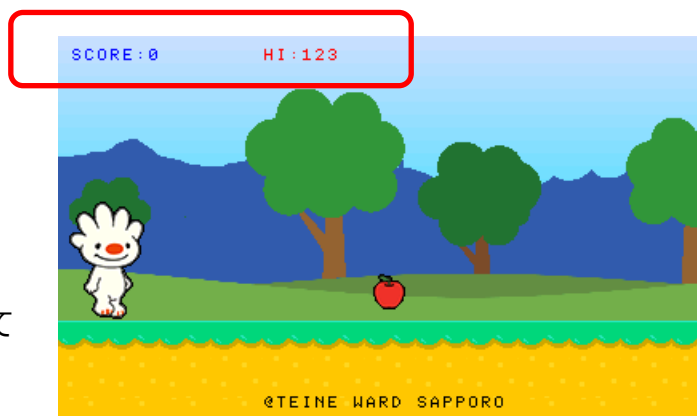
実行すると上画面にスコアなどが表示されました。

少しゲームっぽい画面になりました。

【変数メモ】

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”) を検索して入力します。

◎追加した後の状態 (206 行目付近)

赤枠内の、`'+ L03<左右移動`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。

```
IF V<-VM THEN V=-VM
IF V> VM THEN V= VM
TEVX=V
? ---- 横向き壁チェック
IF MAPHIT(TEPX+TEVX, TEPY, TEW) THEN `'+ L03<左右移動
? ---- 壁にぶつかった
TEPX=(TEPX DIV 32)*32 `'+ L03<左右移動
TEPX=TEPX+(31*(TEDIR==1)) `'+ L03<左右移動
TEPX=TEPX-(TEDIR*(TEW/2)) `'+ L03<左右移動
ELSE `'+ L03<左右移動
? ---- 壁なしに進める)
TEPX=TEPX+TEVX `'+ L03<左右移動
ENDIF `'+ L03<左右移動
? ---- 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
? ---- 敵やフルーツに触ったか?
```

◎テストプレイ (START ボタン)

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。

リングは出ていますが取ることはできません。

マップの右端まで進むとスクロールが止まります。

左側に行くと「ていぬくん」が見えなくなります。

【変数メモ】

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

【命令・関数メモ】

MAPHIT	足元や壁などのマップ状態を調べる命令
--------	--------------------



※もし、実行時に Syntax Error などのエラーが出て動かない場合は、直前に入力した部分をよく見て間違いを見つけて正しい命令に修正してください。

L04 : Y ボタンで加速する

修正する部分 ("L04") を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 (194 行目付近)

赤枠内の、'+ L04(タツシ) 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。

```
IF V>-RM && V<RM THEN V=0:TENM$="IDLE"
ENDIF
?---- Yが押されている間タツシ
VAR VM=1.5, A=30*TEDIR
IF TESTOP==0 THEN '+ L04(タツシ
  IF BTN AND #Y THEN '+ L04(タツシ
    VM=TEVXM:IF V THEN TENM$="RUN" '+ L04(タツシ
    IF BTVX THEN SPANIM SP_TEINU, "R", -15, A, 1 '+ L04(タツシ
  ENDIF '+ L04(タツシ
ENDIF '+ L04(タツシ
?---- Yが落されているとき角度を戻す
IF (BTN AND #Y)==0 THEN '+ L04(タツシ
  SPANIM SP_TEINU, "R", -8, 0, 1 '+ L04(タツシ
ENDIF '+ L04(タツシ
IF V< VM THEN V= VM
IF V> VM THEN V= VM
TEVX=V
?---- 横向きの壁チェック
IF MAPHIT(TEPX+TEVX, TEPY, TEW) THEN
```

◎テストプレイ (START ボタン)

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。

【変数メモ】

BTN	最新のボタンの押されている状態
TENM\$	ていぬくんのアニメーション名
BTVX	左右のボタンで作る移動量 (±1.0)
SP_TEINU	ていぬくん表示用のスプライト番号

【命令・関数メモ】

SPANIM	スプライトのアニメーションを指定する命令
--------	----------------------

【実験】

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 (222 行目付近)

赤枠内の、'+ L05<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
' --- 敵やフルーツに当たったか?
VAR H=SPHITSP(SP_TEINU) '+ L05<フルーツ>
IF H!=-1 THEN '+ L05<フルーツ>
IF SPVAR(H,V_HP) THEN '+ L05<フルーツ>
' --- 相手の上に落下?
IF TEVY>0 THEN '+ L05<フルーツ>
OBJDEAD H '+ L05<フルーツ>
ELSE '+ L05<フルーツ>
IF TEOLD#!="MISS" THEN BEEP 64 '+ L05<フルーツ>
TENM#="MISS":TEDMG=8 '+ L05<フルーツ>
ENDIF '+ L05<フルーツ>
ELSE '+ L05<フルーツ>
SCADD=SCADD+SPVAR(H,V_SCD) '+ L05<フルーツ>
SPCLR H:BEEP 7 '+ L05<フルーツ>
ENDIF '+ L05<フルーツ>
ENDIF '+ L05<フルーツ>
END
```

◎テストプレイ (START ボタン)

実行して移動するとリングに触ったときにリングが消えてスコアが加算されます。

【変数メモ】

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のアニメ名

【命令・関数メモ】

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") を検索して入力します。

◎追加した後の状態 (122 行目付近)

赤枠内の、'+ L06<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 '+ L06<ジャンプ>
?---- Bボタンが押されたらジャンプ
IF BTI AND #B THEN '+ L06<ジャンプ>
TEJUMP=1: TENM$="JUMP": BEEP 8 '+ L06<ジャンプ>
TECNT=16 '+ 押している時間でジャンプ高さ変化用のカウンタ
TEVY=-4 '+ ジャンプ初速
ENDIF '+ L06<ジャンプ>
ENDIF '+ L06<ジャンプ>
?---- ジャンプ中?
IF TEJUMP==1 THEN '+ L06<ジャンプ>
?---- 押してる時間ジャンプし続けている?
IF TECNT==0 THEN '+ L06<ジャンプ>
?---- 普通のジャンプ中
TEVY=TEVY+0.5 '+ L06<ジャンプ>
IF TEVY>0 THEN TENM$="FALL" '+ L06<ジャンプ>
ELSE '+ L06<ジャンプ>
?---- 押している間上昇
TECNT=TECNT-1 '+ L06<ジャンプ>
IF (BTI AND #B)==0 THEN TECNT=0 '+ L06<ジャンプ>
ENDIF '+ L06<ジャンプ>
ENDIF '+ L06<ジャンプ>
?---- 上下移動
IF TEVY THEN TEPY=TEPY+TEVY
?---- 上昇中か?
VAR C, Y
```

◎テストプレイ (START ボタン)

実行して B ボタンを押すとその場でジャンプします。伝統的な B ボタンを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。

【変数メモ】

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】

TECNT、TEVYの初期値、TEVYジャンプ中の加算値



L07 : マップを長くする

修正する部分 (“L07”) を検索します。

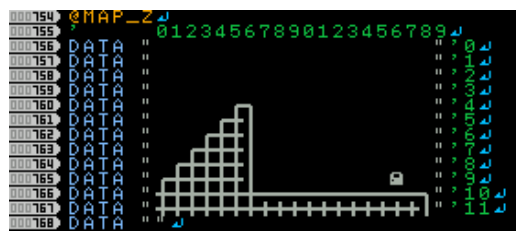
◎ 追加した後の状態 (38 行目付近)

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、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。

```
MAPSET " AAABACADAAEBAAZ" 'マップ'
GOSUB @INIT_TEINU
```

◎ マップの構造 (672 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。



L08 : マップにフルーツや敵を配置

修正する部分 ("@MAP_A") を検索します。

マップ全体の長さ (マップ部品の並び順) は38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。

```
" [ " " E " " ] " " - " " T " " + " " | " " | "
" [ " " E " " ] " = 壁、" ■ " = 金属板
" * " = 栗、" 兎 " = カニ、" 魚 " = スカンク、" 罟 " = くま
" 果 " = リンゴ、" 毒 " = 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。

```
@MAP_A                                     @MAP_B
'      0123456789012345                  '      0123456789012345
DATA "                                     DATA "                                     "' 0
DATA "                                     DATA "                                     "' 1
DATA "                                     DATA "          ■ ■          "' 2
DATA "                                     DATA "          ■ ■          "' 3
DATA "                                     DATA "          ■ ■          "' 4
DATA "                                     DATA "          ■ ■          "' 5
DATA "                                     DATA "          ■ ■          "' 6
DATA "                                     DATA "          ■ ■          "' 7
DATA "                                     DATA "          ■ ■          "' 8
DATA "          ● ● ●          "' 9
DATA "          ──────────          "'10
DATA "          ──────────          "'11
DATA "                                     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かまいません

◎テストプレイ (START ボタン)

実行すると、L07 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、L08 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。

L99 : タイトル画面の表示

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（"L99"）を検索して入力します。

◎追加した後の状態（206 行目付近）

赤枠内の、'+L99<タイトル>'の前に以下のようにプログラムを追加します。

```
' --- タイトル
@LOOP
BGMSTOP:SECLR
TITLE '+L99<タイトル>
' --- ステージ初期化
MAPSET "AAABACADAAEBAAZ"
GOSUB @INIT_TEINU
BGMPLAY 12 '+L99<タイトル>
SETMSG 34 'スタート '+L99<タイトル>
' --- メインループ
ISLOOP=1
WHILE ISLOOP
GET_BUTTON
```

◎テストプレイ（START ボタン）

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。おつかれさまでした！！

【 実験 】

FONTSWを0以外にする、BOSSの熊は絶対倒せないで倒せるように修正できるか？、スカクの尻の回数調整
気になる命令を見つけたら「？」ボタンを押してインラインヘルプの説明を見る

参考資料

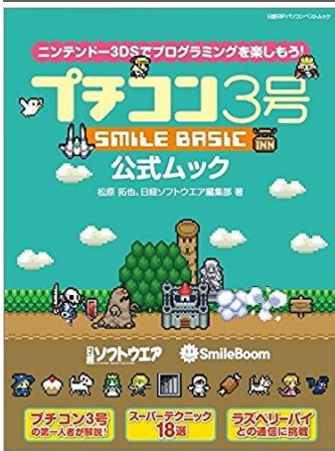
ニンテンドー3DS用の「プチコン 3号 SmileBASIC」および WiiU 用の「プチコン BIG」でプログラムを作る時の参考資料として以下の書籍が発売されています。

プチコン 3号+BIG 公式リファレンスブック : SMILEBASIC MAGAZINE SPECIAL



アンビット編集

プチコン 3号 SMILEBASIC 公式ムック



日経ソフトウェア編集部

◎ ツイッター上での情報のやり取りを検索する場合は、以下のハッシュタグ等をご利用ください。

ハッシュタグ

#petitcom、#sbkey

検索キーワード

プチコン OR SmileBASIC OR #petitcom

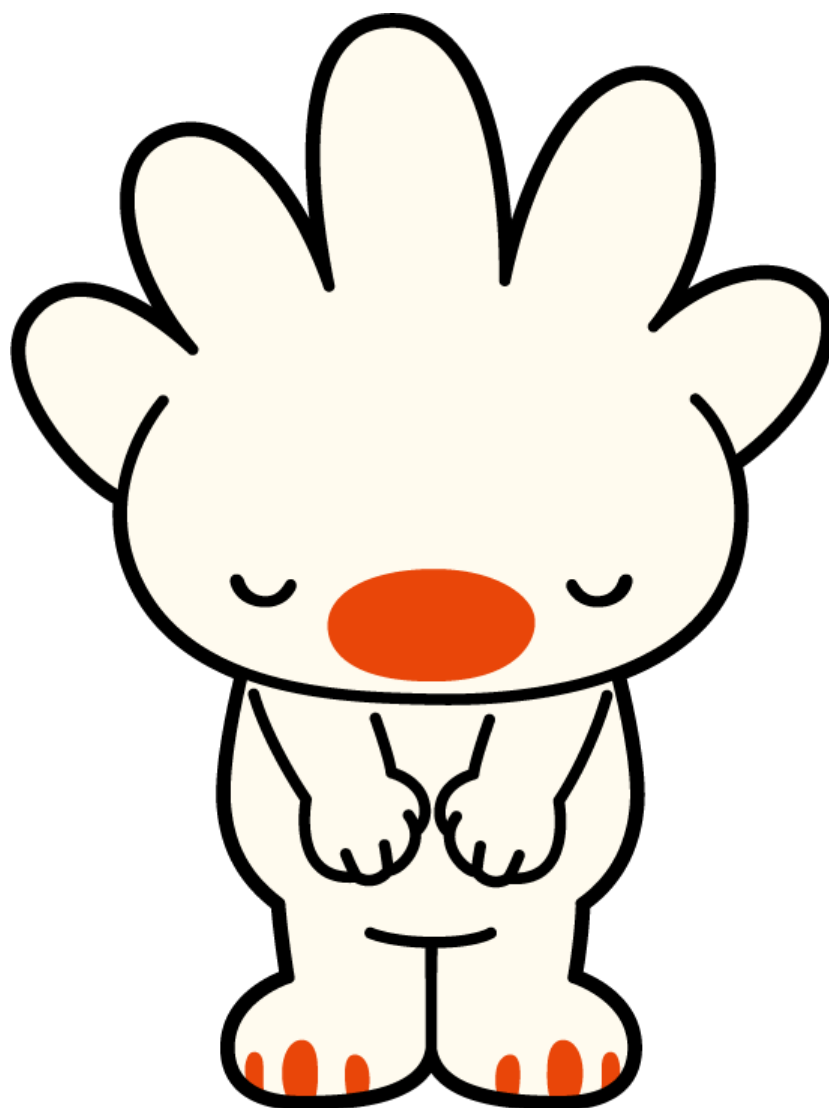
公式アカウント

@PetitComputer

◎ スマイルブーム公式ページ（他の人が作った作品や公式の素材等も並んでいます）

<http://smilebasic.com/>

おつかれさまでした！



©TEINE WARD SAPPORO

プチコンは株式会社スマイルブームの登録商標です。
ニンテンドー3DS・ニンテンドー3DS ダウンロードソフトは任天堂の登録商標または商標です。