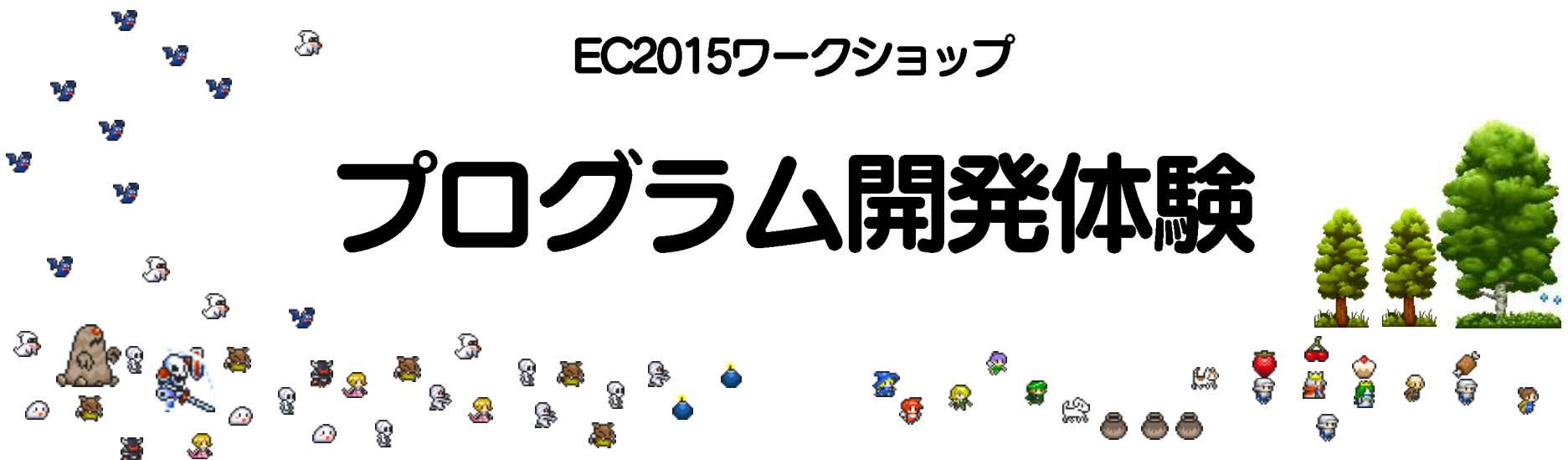


EC2015ワークショップ

プログラム開発体験



 SmileBoom Co.Ltd.

ニンテンドー3DSとPCHコン3号を使ったプログラムの開発

※資料内の商品名や社名は一般に各社の商標または登録商標です

本日のワークショップの内容

- これからニンテンドー3DSを使ってプログラムを作ります
- プログラムを作るソフトは「プチコン3号SmileBASIC」です
- 普段遊んでいるゲームもプログラムで動いていることを少しでも体験してみましよう

このワークショップは、次のような人を対象としています

- プログラムのことは、まったくわかりません
- 英語は習っていないので読めないし意味もわかりません
- 絵がうまく描けません
- 楽譜を読んだり楽器を演奏できません

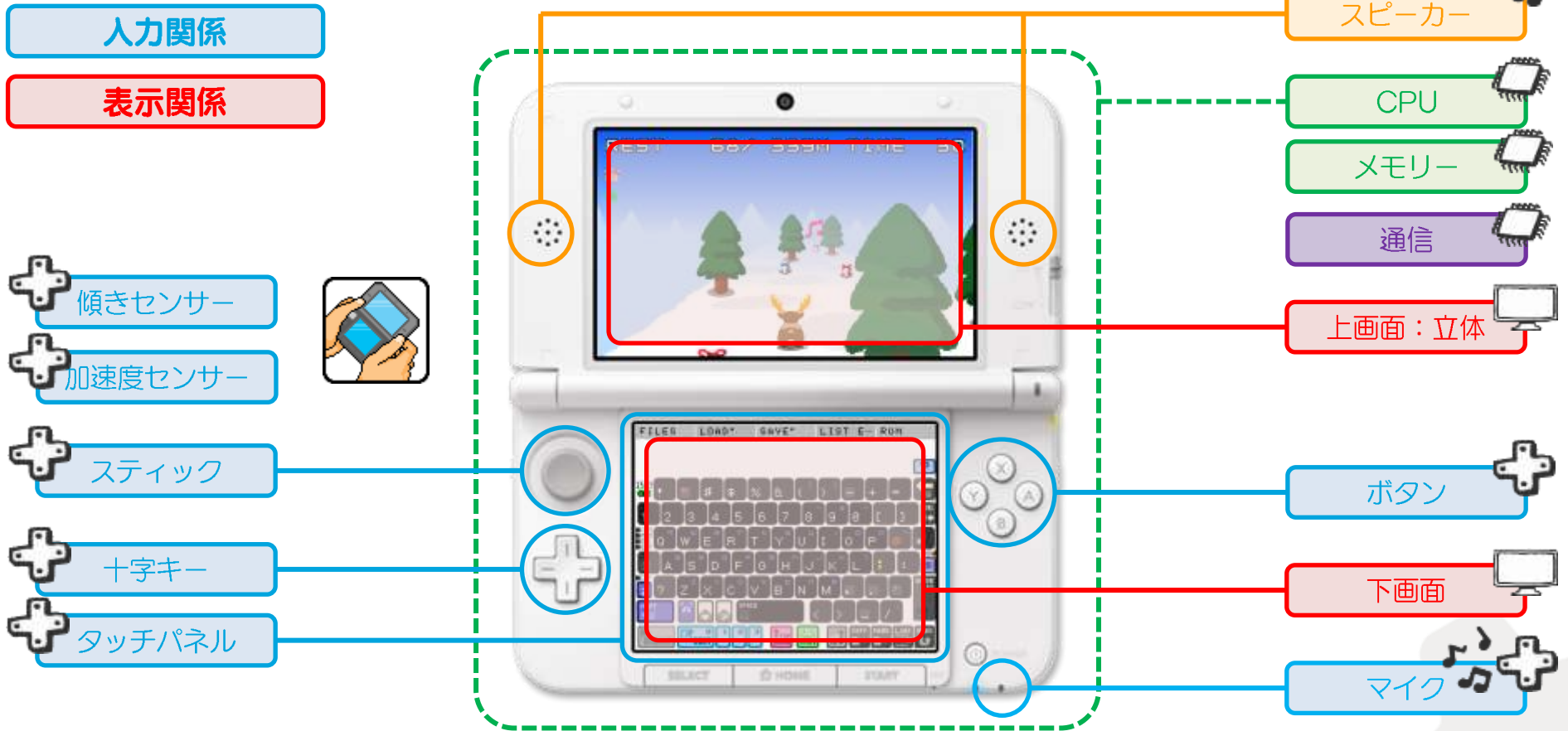
分からないことや気になることがあれば、いつでも声をかけてください！

ニンテンドー3DSとゲームの仕事

- どんな機能（部品）があればゲームを作ることができるのか？

ニンテンドー3DSにはどんな部品がありますか？


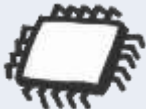
- ・ニンテンドー3DSに搭載されている部品を調べてみましょう



「表示機能」「入力機能」「音を鳴らす機能」「通信機能」「CPU」5つの要素が組み込まれている
※PlayStation、Xbox、WiiU、パソコン、スマホなどはどうだろう？

機能ごとのゲームの仕事（職種）

- それぞれ専門の仕事を担当する人達が集まって1つのゲームを作っています

部品	仕事の名前	仕事の内容	説明
表示機能 	グラフィックデザイナー	絵を描く仕事	ゲームの世界設定に合わせた原画を元にドット画像や3Dモデルを作る
人力機能 	プランナまたはゲームデザイナー	ルールを考える仕事	操作方法やルールを考えて企画書や仕様書を作成する
音を鳴らす機能 	サウンドコンポーザ	作曲したり効果音を作る仕事	ゲームの世界設定に合わせた原画やグラフィック画像を元に効果音やBGMを作る
通信機能 	サーバープログラマ	ネット用のプログラムを作る仕事	インターネット経由でサーバーとやり取りをするプログラムを作る
CPUやメモリー 	プログラマ	表示や操作や制御のプログラムを作る仕事	用意された仕様書を元にゲーム全体のプログラムを作る

はじめての **プチコン3号**

SMILE BASIC

- TOPMENU (トップメニュー)
- キーボードの使い方



アイコンを
タッチ!

取扱説明書の目次

- プチコン3号起動中にHOMEボタンを押すと「取扱説明書」を見ることができます
- プログラムの作り方やBASIC入門など役立つ情報が含まれています
- 最初にひと通り目を通しておくとプチコンに関する基本的な知識が得られます



はじめにお読みください

- 1 ご使用になる前に

安全にお使いいただくために

- 2 パレンタルコントロール
- 3 拡張スライドパッドについて
- 4 ユーザーコンテンツについて

通信とインターネット

- 5 インターネットについて
- 6 いつの間通信
- 7 通信プレイについて
- 8 ゴールド会員権の購入(有料)

はじめてのプチコン

- 9 プチコンについて
- 10 操作方法
- 11 BASICについて
- 12 TOP MENUについて
- 13 作品公開とダウンロード
- 14 プロジェクトとファイルの管理
- 15 オプション

プログラムを作る

- 16 BASICを起動する
- 17 キーボードの使い方
- 18 DIRECTモードで命令しよう
- 19 EDITモードでプログラム入力
- 20 EDITモードの編集機能
- 21 プログラムの保存と読み込み
- 22 サンプルプログラムについて
- 23 HELPの使い方
- 24 SMILEツールの使い方

BASIC入門

- 25 PRINTと変数
- 26 変数の使い方
- 27 入力と条件判断
- 28 コンピュータの色(RGB)
- 29 グラフィック命令
- 30 サウンド命令

詳細情報

- 31 BASIC基本仕様
- 32 立体視
- 33 画面構成
- 34 BG
- 35 SPRITE
- 36 お問い合わせ先

TOPMENU (トップメニュー) の機能

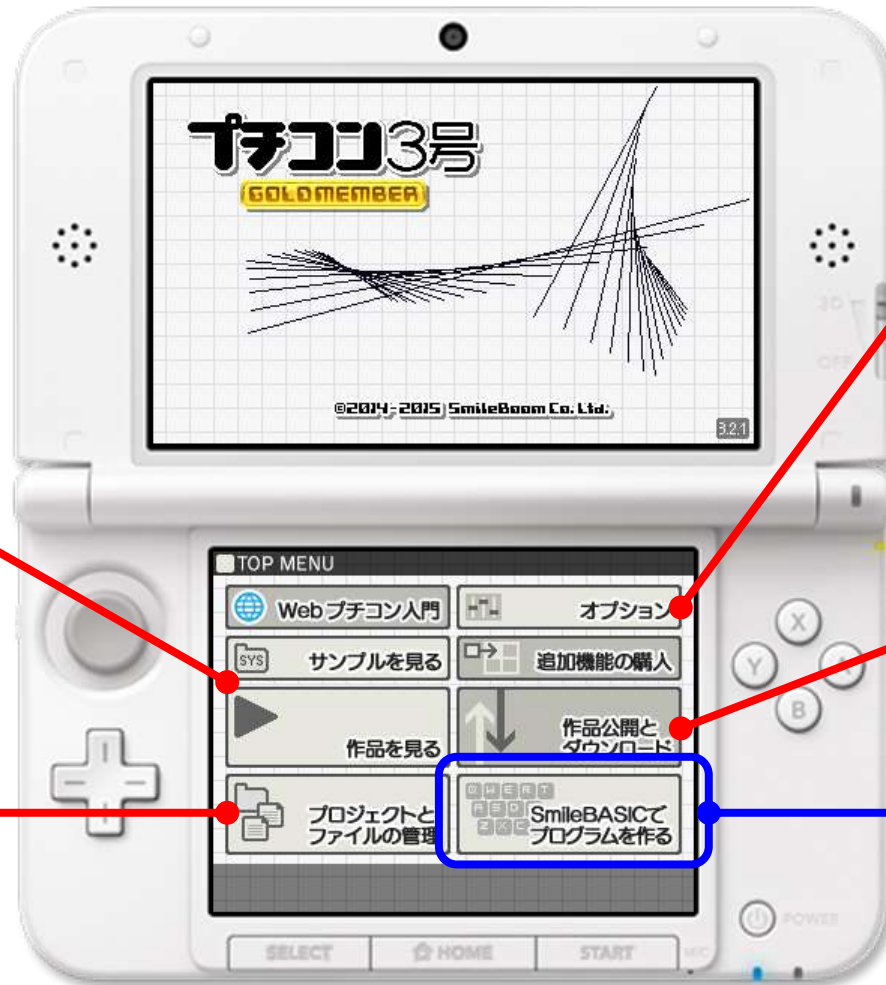
- ・ 起動すると必ず最初に表示される画面 (やりたい作業にあわせてボタンをタッチ)

- Webプチコン入門
 - ・ 公式初心者ページへ
- 追加機能の購入
 - ・ 拡張サービスの購入

サンプルや作品を見る



ファイルの管理機能



色や設定の調整



公開とサーバー管理機能

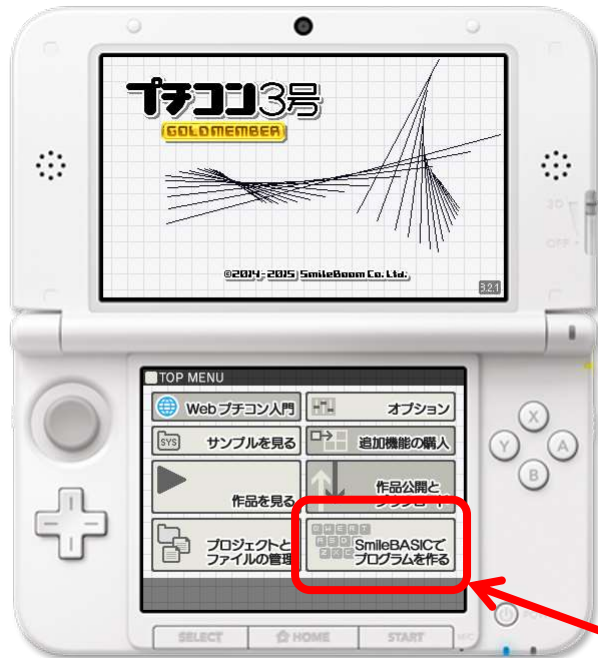


SmileBASICでプログラム作成



プログラム開発中の画面の切り替え

- プログラム作成中は、 **TOPMENU** と SmileBASIC の間はそれほど切り替えない
- SmileBASIC 内での **DIRECT** と **EDIT** の切り替えは良く使う



TOPMENU との行き来

SmileBASICでプログラム開発中の切り替え

DIRECT モードでプログラムを実行



DIRECT と **EDIT** の切り替え

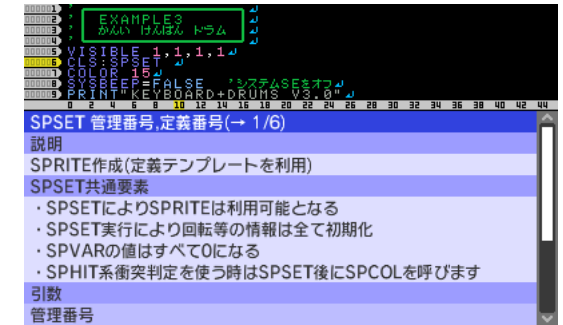
EDIT モードでテキストを編集



ツールへ

キーボードについて

- SmileBASICの作業中は下画面にキーボードが表示されます
 - DIRECTモードとEDITモードで共通部分



- ### 命令候補エリア
- 通常は命令候補を表示
 - #で始まる定数の確認
 - 長い命令の入力回避
 - 文字を増やし候補を絞る

- ### ●CAP
- SHIFTキーを押したままの状態にする

- ### 文字種類の変更
- 英数字
 - 記号 (欧州文字)
 - ひらがな/カタカナ

- ### スマイルボタン
- ツールを実行する

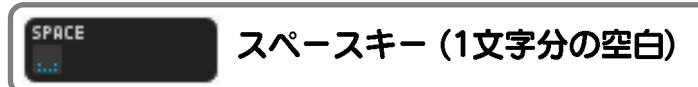
- ### ファンクションキー
- KEY命令で再定義可能なボタン

- ### ヘルプ表示モード切替


- ### Yボタン
- 1文字削除(BSキー相当)

- ### Aボタン
- 改行(エンターキー相当)

- ### INS
- 挿入と上書きの切り替え



ENTERキー (改行)

本資料内で、 という記号がある場合は、ENTERキーを入力すると解釈してください。

その他のキーとステータス表示部品

- 主にテキスト入力時に利用します（SHIFTを押すと変化するボタンがあります）

ファンクションキー（KEY命令で定義可能なボタン）

FILES LOAD" SAVE" LIST E... RUN

SHIFTを押すと、SLOTのSAVEとLOADボタンとなる

SAVE LOAD *EX3KEYDRUM



バックスペースキー（カーソルの前を1文字消す）



SHIFTを押しながら押すと1行削除



デリートキー（カーソル位置の文字消す）



SHIFTを押しながら押すとカーソルから行末まで削除

DIRECTモードとEDITモードの切り替え（SLOT変更）



- COM（通信中）
- MOT（センサー利用中）
- MIC（マイク利用中）
- EXP（拡張スライドパッド利用中）



範囲選択（カーソルのある位置を始点とする）



コピー（選択範囲を取り込む、範囲が無い時1行）



ペースト（取り込まれた内容を貼り付ける）



1行挿入（カーソルの下の行に1行追加する）



やり直し（直前の操作を元に戻す）



カット（選択範囲を切り取る、範囲が無い時1行）



ペースト（取り込まれた内容を貼り付ける）



1行削除（カーソル行を1行削除する）



取り消し（UNDOを取り消す）

※CUT以降は、SHIFTキーを押した時に切り替わります



TOPMENUへ戻る



検索と置換モードへ

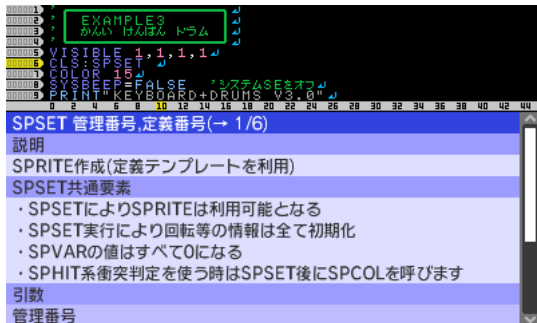
12:54



時計と電池残量

ヘルプの使い方

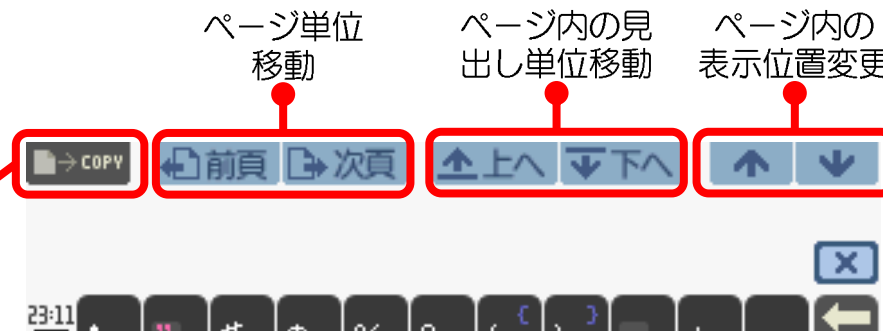
- 調べたい単語にカーソルを合わせて **?** を押します (SPSETの表示例)



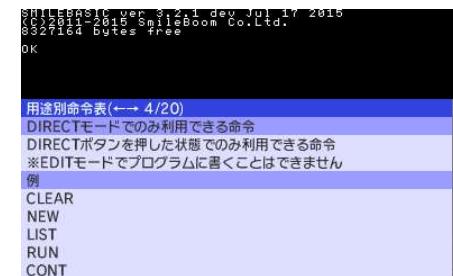
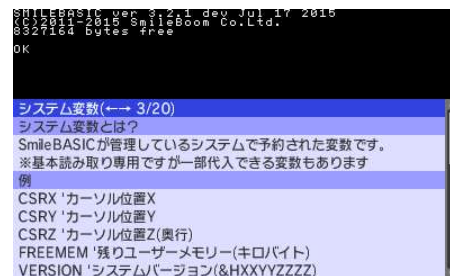
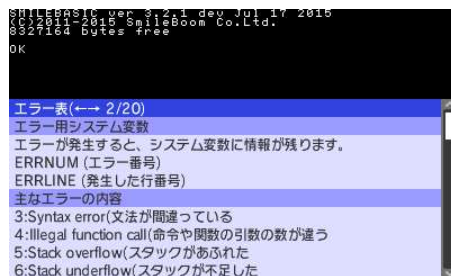
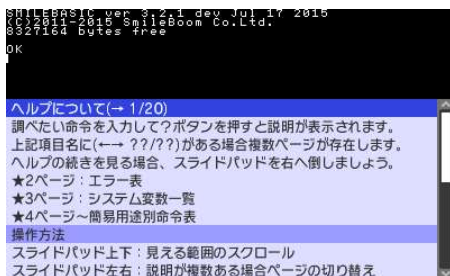
上画面にヘルプが表示されているときは、スティックの上下で説明を見渡すことができます。下画面のファンクション部分はヘルプ操作専用に変化しているのを見出し単位等の操作が可能です。

ヘルプ内の「例」の内容を内部へ記録。

編集画面で **PASTE** を押して貼り付けます



- 何も単語を選んでいる状態で押した場合はヘルプ自身の説明となります
 - スティックを右に倒すと、エラー表・システム変数・用途別命令がページごとに登録されています



ダイレクト

DIRECTモードについて

- TOPMENUからSmileBASICに入ってください
- DIRECTモードでコンピュータに直接指示

ボタンを
タッチ!



DIRECT (ダイレクト) モードとは?

- 3DSの中に入っているコンピュータに直接指示を出すモードです
- 命令を1行分入力して実行することができます

DIRECT モードでプログラムを実行する場合の表記 (1行単位)

```
ACLS ↵
```

スティック

- 左右で行頭行末移動

十字ボタン

- 上下で入力履歴の再利用
- 左右でカーソル移動



画面に1+3の答えを表示する「PRINT」

- 単純な計算結果を表示させます

```
PRINT 1+3 ↵
```

```
4  
OK
```

まるで電卓のように計算結果を表示します（数学関数も搭載しているので関数電卓代わりにも使えます）

- ひき算 PRINT 5-2 ↵
- かけ算 PRINT 2*4 ↵
- わり算 PRINT 6/3 ↵
- わった余り PRINT 6 MOD 5 ↵
- 複雑な計算 PRINT (1+3)*5-2/4 ↵
- 小数を含む計算 PRINT 1.25*2.5+5.77 ↵
- 四捨五入 PRINT ROUND(0.25*63.99)+15 ↵
- 0~9までの乱数 PRINT RND(10) ↵

PRINT は、？(クエスチョン・はてな)記号で省略することが可能です

```
? 1+3 ↵
```

入力された数字の合計を表示「INPUT」

- ”:”(コロン)で区切ることで2つの命令を指示することができます
- INPUT 後に変数を”,”(カンマ)で区切って複数入れると複数入力が可能です

```
INPUT A, B:PRINT A+B
```

上記命令を実行すると、画面に?が表示されます (これはINPUT命令が数値入力を待っている状態です)

```
?
```

ここで適当な数字を1つ入力してエンターキーを押すと警告 (?Redo from start) が出ます

```
?53  
?Redo from start
```

INPUT A, Bと書いたのが2個分の数値入力が必要です (カンマで区切って数字を2個人力します)

```
53, 67  
120  
OK
```

正しい結果が表示されました (120を表示しているのは、PRINT A+Bの命令です)

画面に線や円を描く

- 順番に実行すると画面に線や円が描画されます

```
ACLS ↵
```

まず、表示関係を初期状態に戻します（画面の文字などが消えます）

```
GLINE 0, 0, 399, 239, RGB(255, 0, 0) ↵
```

左上から右下へ赤い線を引きます

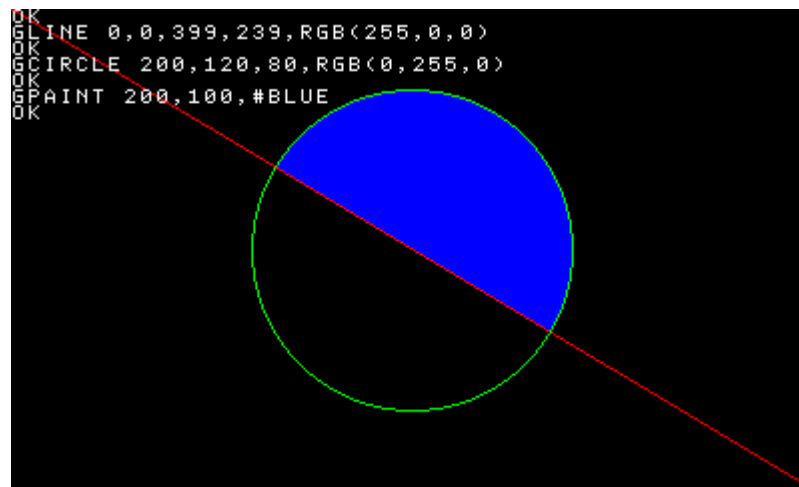
```
GCIRCLE 200, 120, 80, RGB(0, 255, 0) ↵
```

画面中央に半径80ドットの緑の円を描きます

```
GPAINT 200, 100, #BLUE ↵
```

最後に画面中央より少し上から青で塗ります

#BLUEは、定数と呼ばれるシステムが予約した名前です



音を出す (効果音・BGM・声)

- どんな効果音だったか忘れたときに簡単に確認できます (134種類)

BEEP 8 ↓

- 標準搭載されているBGMについても確認できます (43種類)


BGMPLAY 41 ↓

- BGM演奏を止めたい時は以下のように入力して下さい

BGMSTOP ↓

- 音声合成によって会話を喋らせることもできます

TALK "オハヨウゴザイマス" ↓

- プチコン3号に搭載されている音声合成は日本語しか喋れません
- ひらがなカタカナの入力は、キーボードの  を押してキーの種類を切り替えてください

作ったプログラムを SAVE 命令で保存する

- 3DSの電池が切れるとせっかく作ったプログラムは消えてしまいます
- プログラムを書いたらこまめに保存する癖をつけましょう
 - 何か大きな変更を行う前には保存しておくのと元に戻すのが楽です
 - 大きな変更を行う前にはバックアップとして別の名前を付けて保存すると安心です
- ファイル名として使える文字はアルファベットの大文字A~Zと数字および _ です

```
SAVE" SAMPLE" ↵
```

- SLOTごとのSAVE専用の機能 (SHIFTを押しながらファンクションキー)

SAVE LOAD *EX3KEYDRUM

- SAVE用のダイアログが開きファイル名を入力することができます
- DIRECTにSAVEコマンドを入力するよりも楽に保存できます
- 現在作業対象となっているSLOTの内容が保存されます



保存されたプログラムを LOAD 命令で読み込む

- 前回の続きからプログラムを作る場合はSAVEで保存されたファイルを読み込みます
- 作業中のプログラムがある状態でLOADすると作業中のものは消えてしまいます
 - 作業中のプログラムを後で利用する可能性がある場合は名前を付けてSAVEしておきましょう

```
LOAD" SAMPLE" ↓
```

- SLOTごとのLOAD専用の機能（SHIFTを押しながらファンクションキー）

SAVE **LOAD** *EX3KEYDRUM

- LOAD用のダイアログが開きファイルを選択することができます
- 現在の作業SLOTに読み込むファイルを選択し「決定」を押します
- 現在読み込まれているファイル名は **青い枠部分** に表示されます



エディット

EDITモードについて

- まとめて命令を書くためのテキスト入力画面

ボタンを
タッチ！



EDIT (エディット) モードとは？

- コンピュータへ指示する内容をテキストとして登録するモードです
- 命令をたくさん並べて書くことで連続して指示を出すことができます
 - 編集用の行番号は実行時には意味がありません
- プログラム実行は **DIRECT** モードから
 - RUN ↓

EDIT モードでテキストを編集する場合の表記 (複数行)

```
000001 ACLS ↓  
000002 PRINT 1+2 ↓
```

実行するためには **DIRECT** モードで RUN (またはファンクション5)

```
RUN ↓
```

スティック

- 上下で先頭と最後尾
- 左右で行頭行末移動

十字ボタン

- 上下左右でカーソル移動

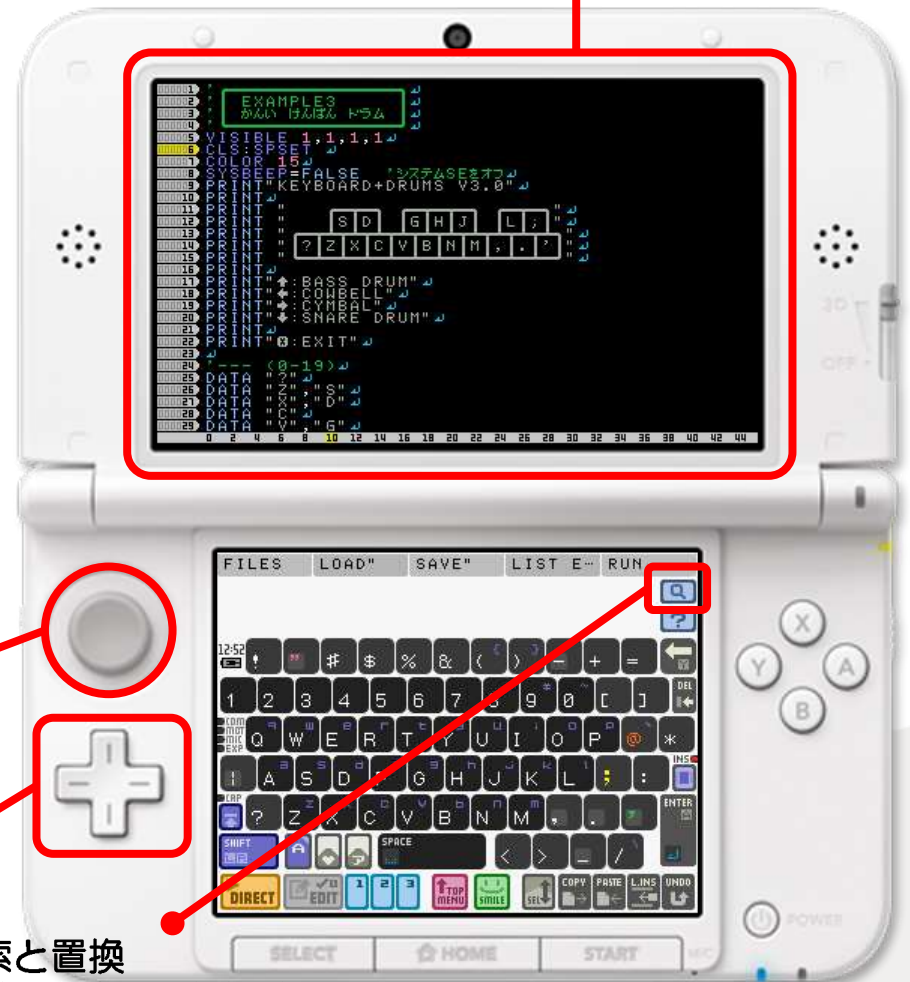
SHIFTを押しながら操作するとスティックと同じ動作となる

検索と置換

- テキスト内から文字を探すときや置き換えるときに利用します

編集画面

- カーソルのある位置に文字が入力されます

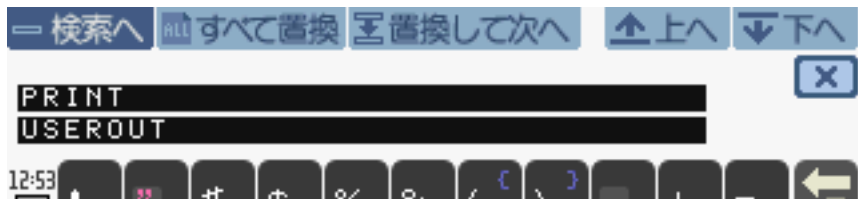


検索と置換の使い方


- 調べたい文字列を文字列入力エリアへ入力すると検索が始まります
- 前の候補や次の候補へ進めたいときは **↑上へ** **↓下へ** ボタンを押します
- 文字列を置き換えたい場合は **≡置換へ** を押して置換モードに切り替えます



- 置換モードに切り替わると文字列入力エリアが1段増えます
- 変更後の文字列を入力して **⏪すべて置換** を押すとプログラム内の対象がすべて変化
- 変換するかどうかを確認しながら進める場合は **≡置換して次へ** を押してください
- 検索に戻る場合は **←検索へ** を押します



新しいプログラムを作る準備

- 新しいプログラムを入力する前に今まで入力したプログラムをすべて消します
-  DIRECTボタンを押して実行モードに切り替えます
- キーボードから「NEW」と入力してENTERを押して下さい

NEW ↓

- 編集作業対象が **SLOT0** になっていることを確認してください
 - プチコンには上級者向けに作業用SLOTが4個用意されていますが今回はSLOT1～SLOT3は使いません

このボタンが押されて
いればOKです！




単純な画面表示のプログラム入力

- いよいよ本格的なプログラム作業の始まりです
- ①～⑤までの5種類のプログラムを入力します
- 最初に画面に「イチゴ」と「円」を描いてみましょう
- EDITモードで25行分のプログラムを入力します

①

画面にイチゴと円を表示するプログラムを入力

-  EDITボタンを押して編集画面を開きます
 - 編集画面から以下のプログラムを入力してみましょう

```
000001  '  ↓
000002  '  ■がくしゅうサンプル 2015.9 ↓
000003  '  ↓
000004  OPTION STRICT ↓
000005  ACLS:XSCREEN 2,256,2 ↓
000006  '  --- ↓
000007  PRINT "#へんすうしょきが" ↓
000008  VAR SFT,BTN ↓
000009  VAR TX,TY,TS ↓
000010  VAR AX,AY,AZ ↓
000011  VAR OX=0,OY=0,SW=0 ↓
000012  VAR I,X,Y ↓
000013  '  --- ↓
000014  PRINT "#いちご(DISPLAY 0/うえがめん)" ↓
000015  DISPLAY 0 ↓
000016  SPSET 0,0 ↓
000017  SPOFS 0,200,120,-100 ↓
000018  SPHOME 0,8,8 ↓
000019  SPSCALE 0,4,4 ↓
000020  '  --- ↓
000021  PRINT "#えんをかく" ↓
000022  FOR I=1 TO 5 ↓
000023  GCIRCLE 200,120,(2<<I)*8,RGB(64,64,64) ↓
000024  NEXT ↓
000025  '  --- ↓
```

基本的な画面構成の初期化

このプログラム全体で利用する変数の初期化

スプライト機能を使ってイチゴを上画面に表示

SPSETでスプライトを1つ確保し、SPOFSで位置調整、SPHOMEで表示原点調整、SPSCALEで拡大率を設定

グラフィック画面に円を5回表示

変数名	利用方法
SFT, BTN	ボタンの入力情報を管理
TX, TY, TS	タッチ座標とタッチ時間を管理
AX, AY, AZ	傾きセンサーの値を管理
OX, OY, SW	描画関係の座標と状態を管理
I, X, Y	汎用の変数

画面にイチゴと円を表示するプログラムを実行

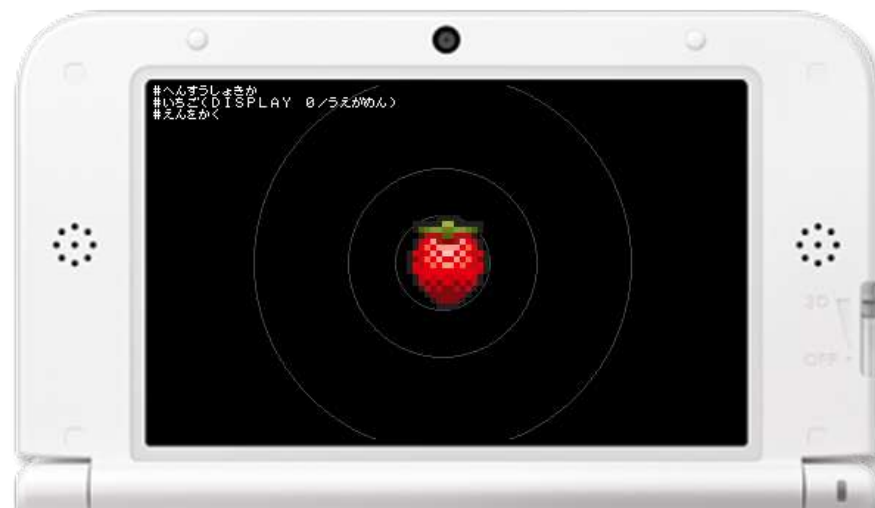
-  DIRECTボタンを押して実行モードに切り替えます

RUN ↓

- キーボードから、**RUN** ↓ と入力します (START/SELECTでもRUNできます)
- 上画面に「イチゴ」が表示されましたか？
- 上画面に「灰色の円」が表示されていますか？

動かない時は、次のページ「エラーが出てしまった人へ」を参考に修正しましょう！！

使った命令	利用方法
OPTION STRICT	宣言前の変数を使えなくする
ACLS	表示関係を初期設定
XSCREEN	上下画面の管理方法変更
PRINT	画面に文字や数字を表示
VAR	変数を宣言
DISPLAY	対象画面切替 (0=上、1=下)
SPSET	スプライトを確保
SPOFS	スプライトの座標変更
SPHOME	スプライトの表示原点変更
SPSCALE	スプライトの拡大縮小
FOR ~ NEXT	指定回数繰り返し
GCIRCLE	グラフィック画面に円を描く
RGB	赤、青、緑の成分から色情報変換




エラーが出てしまった人へ

- RUNした結果、エラーが出てしまった！！

`Syntax error in 0:##` ※##の部分はエラーが出ている行の番号が表示されます

- ##行で書式に間違いがあるようです

-  EDITモードに切り替えて、編集画面から該当行を見てみましょう
- キーボードの上の方にあるファンクションボタンの右から2番目を押すとエラーのある行に飛んでくれます



- その行の中に必ずエラーがあります。元になっているプログラムと良く見比べてみてください
- その他のエラーが出た場合も、まずは編集画面でエラーが出ている行を見てみましょう
- 必ず何か入力間違いがあるはずです
 - 似ている記号'アポストロフィーと、カンマや、ゼロとオーなどが良く間違えます
 - 緑色のコメント（注釈）としてメモが書かれている行はアポストロフィーで始まっています

プログラムを作るという仕事は、

「プログラムを書いて」→「実行」→「エラーがあれば修正」→「再び実行」・・・
という作業を正しい動きになるまで繰り返す作業となります。

★ただしく動いたら「**タッチで絵を描く**」へ進みましょう！

タッチで絵を描くプログラムを追加

- ・タッチペンを使って下画面に小さい四角を描いてみましょう

②

お絵かき：メインループ

- プログラムのメインとなるループ部分を入力します
 - 編集画面から以下のプログラムを入力して下さい

```
000021 PRINT "#えんをかく" ↵
000022 FOR I=1 TO 5 ↵
000023 GCIRCLE 200,120,(2<<I)*8,RGB(64,64,64) ↵
000024 NEXT ↵
000025 ' --- ↵
000026 DIM COL[8],VOICE#[8] ↵
000027 VAR CL=7,R,G,B ↵
000028 COL[CL]=RGB(255,255,255) ↵
000029 ' ===== ↵
000030 ' ↵
000031 ' ■メインループ ↵
000032 ' ↵
000033 PRINT "#メインループ" ↵
000034 @LOOP ↵
000035 BTN=BUTTON(0):SFT=BTN AND &H300 ↵
000036 TOUCH OUT TS,TX,TY ↵
000037 ' --- ↵
000038 DRAWSUB TS,TX,TY ↵
000039 ' --- ↵
000040 VSYNC 1 ↵
000041 GOTO @LOOP ↵
000042 ↵
```

すでに入力済み（そのまま使います）

最初の色を設定。RGB(255,255,255)は白を意味する

■メインとなるループ構造

@LOOPでループの先頭位置を定義、
ボタンとタッチ情報を取得し、
DRAWSUB（線を引くプログラム）を呼び出し、
VSYNC命令で描画周期との同期を取り、
@LOOPへ強制分岐（無限ループ）

お絵かき：線の描画

- 下画面をタッチして絵を描くプログラムを追加します
 - 編集画面から以下のプログラムを入力して下さい

```
000040 VSYNC 1↵
000041 GOTO @LOOP↵
000042 ↵
000043 ' =====↵
000044 ' ↵
000045 ' ■せんをひく↵
000046 ' ↵
000047 DEF DRAWSUB S, X, Y↵
000048 IF S==0 THEN SW=0↵
000049 IF S==1 THEN OX=X:OY=Y:SW=1↵
000050 IF SW==0 THEN RETURN↵
000051 ' --- せんをひく↵
000052 DISPLAY 1↵
000053 GCLIP 1, 0, 0, 319, 219↵
000054 IF SFT && S==1 THEN↵
000055   GPAINT X, Y, COL[CL]↵
000056 ELSE↵
000057   GFILL X-4, Y-4, X+4, Y+4, COL[CL]↵
000058   BEEP 27, 0, 20↵
000059 ENDIF↵
000060 ' ---↵
000061 OX=X:OY=Y↵
000062 END↵
000063 ↵
```

すでに入力済み（そのまま使います）


線を引くプログラム

変数Sで受け取ったタッチ時間の情報が0だったら放されていると判断し、Sが1の時は初めてタッチ、Sが1以上の場合タッチし続けていると判断する。

初めてタッチされた時は変数X,Yで受け取ったタッチ座標にGFILLで四角を塗る（ただし、L/Rボタンが押された際はGPAINTで塗りつぶしを行う）。

変数OX,OYに座標の記録を残す。

お絵かき：プログラム実行

-  DIRECTボタンで実行モードに切り替えてプログラムを実行します

RUN ↓

– 下画面をタッチすると白い■が描けますか？

動かない時は「エラーが出てしまった人へ」のページを参考に修正！！

早く動かすとツブツブが表示されますが、
ゆっくり描くと線のようにつなげて描けます。



使った命令	利用方法
DIM	配列の利用宣言
@LOOP	@から始まる名前はラベルとして扱う
BUTTON	ABXY十字キーなどのボタン情報取得
AND	ビット同士の掛け算
TOUCH	タッチ座標と時間情報取得
OUT	複数の戻り値を返す命令用
VSYNC	画面の描画周期と同期をとる
GOTO	指定行へ処理を切り替える
DEF	ユーザーによる命令の定義
IF THEN ENDIF	条件判断 (ELSEは不成立時用)
RETURN	ユーザー命令やGOSUBから戻る
GPAINT	グラフィック画面の塗りつぶし
GFILL	グラフィック画面を矩形で塗りつぶす
BEEP	効果音を鳴らす
END	ユーザー定義命令やプログラムの終了

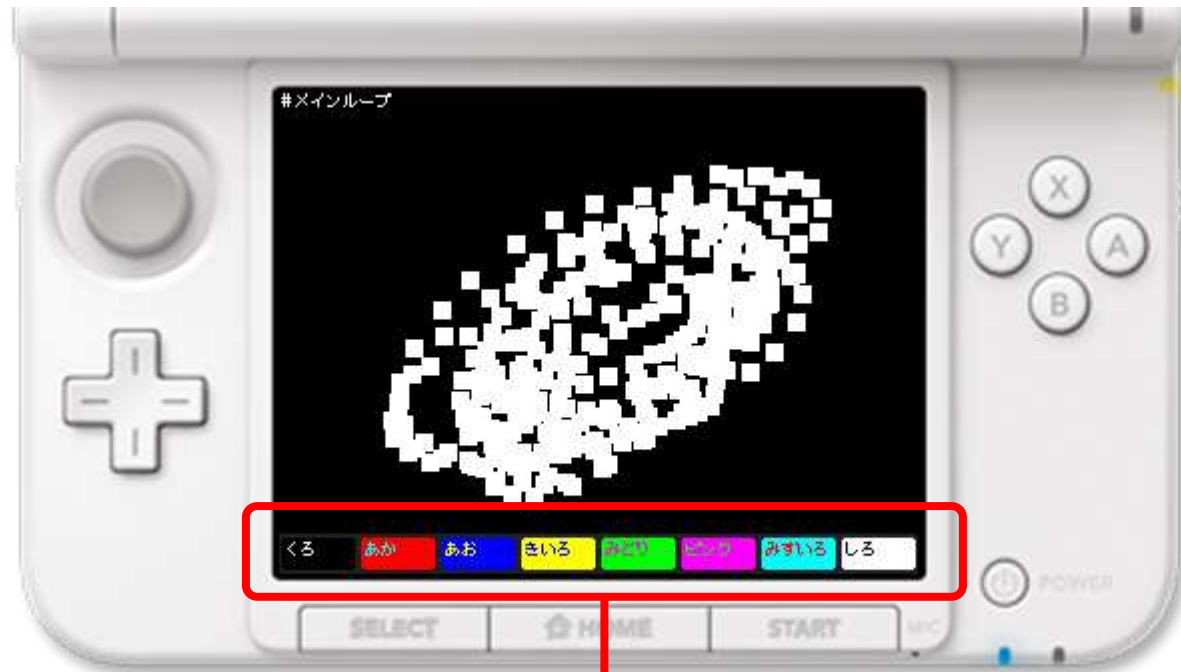
★正しく動いたら「色の選択パネル」を追加してみましょう！

色選択パネルのプログラムを追加

- 線の色を変更するためのプログラムを追加します
- 色選択用のパネルをタッチすることで色が切り替わります
- ついでに色の名前をコンピュータに喋らせます

色選択パネルについて


- お絵かきプログラムの表現力を高めるため色の選択パネルを追加します



色選択パネル

- 8種類の色をタッチで選択
- 選択すると色の名前をコンピュータが喋ります

色選択パネル：色選択プログラム呼び出し部分を2か所挿入

- 下画面に色選択パネルを用意して色を変えるプログラムを追加します
 - 今回はプログラムの上の方に追加（挿入）する部分があります
 - `COL[CL]=RGB(255, 255, 255)` の行にカーソルを移動し  を押すと下に1行追加されます
 - 挿入された行に、`GOSUB @INIT_COLOR` を追記してください
 - 同様に `DRAWSUB TS, TX, TY` の行にカーソルを移動して挿入し `SELCOLOR TS, TX, TY` を追記します

```
000021 PRINT "#えんをかく"
000022 FOR I=1 TO 5
000023 GCIRCLE 200,120,(2<<I)*8,RGB(64,64,64)
000024 NEXT
000025 ' ---
000026 DIM COL[8],VOICE#[8]
000027 VAR CL=7,R,G,B
000028 COL[CL]=RGB(255,255,255)
000029 GOSUB @INIT_COLOR
000030 ' =====
000031 '
000032 ' ■メインループ
000033 '
000034 PRINT "#メインループ"
000035 @LOOP
000036 BTN=BUTTON(0):SFT=BTN AND &H300
000037 TOUCH OUT TS, TX, TY
000038 ' ---
000039 DRAWSUB TS, TX, TY
000040 SELCOLOR TS, TX, TY
000041 ' ---
000042 VSYNC 1
000043 GOTO @LOOP
000044
```

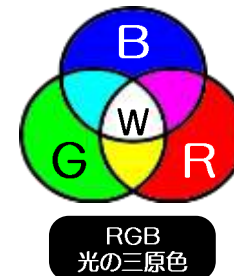
色選択パネル初期化プログラム呼び出し

色選択パネル本体プログラムの呼び出し

行番号は目安なので多少ずれても問題ありません（前後の行のプログラムから挿入場所を判断してください）

色選択パネル：色選択パネルを初期化するプログラム

- 8色分の色情報を下画面に表示して、タッチで選択するためのプログラムです
 - プログラムの一番下の行以降に以下のプログラムを追加します



```
000066 ? =====
000067 ?
000068 ? ■いろしよりしょきが
000069 ?
000070 @INIT_COLOR
000071 DISPLAY 1
000072 ? --- R G B
000073 DATA 0, 0, 0, "くる" '0
000074 DATA 255, 0, 0, "あか" '1
000075 DATA 0, 0, 255, "あお" '2
000076 DATA 255, 255, 0, "きいろ" '3
000077 DATA 0, 255, 0, "みどり" '4
000078 DATA 255, 0, 255, "ピンク" '5
000079 DATA 0, 255, 255, "みすいろ" '6
000080 DATA 255, 255, 255, "しろ" '7
000081 ? ---
000082 GFILL 0, 220, 399, 239, RGB(32, 32, 32)
000083 ? ---
000084 FOR I=0 TO 7
000085 READ R, G, B, VOICE#[I]
000086 COL[I]=RGB(R, G, B)
000087 X=I*40+2:Y=222
000088 GFILL X+1, Y, X+35, Y+15, COL[I]
000089 GFILL X, Y+1, X+36, Y+14, COL[I]
000090 R=RGB(255-R, 255-G, 255-B)
000091 GPUTCHR X+2, Y+2, VOICE#[I], R
000092 NEXT
000093 ? ---
000094 GFILL 0, 0, 319, 219, COL[0]
000095 RETURN
```

作業画面を下画面に変更

色のデータ

R (RED:赤)、G (GREEN:緑)、B (BLUE:青)、“色の名前文字列”の順番で色情報が8種類分並んでいます。

パネルの背景を塗りつぶす

色のパネルを8枚表示

変数I (アイ) を使って0から7までの8回分FOR命令でループし、色の情報をREAD命令でDATAから取り出して下画面にGFILLで色を塗り、GPUTCHRで色の名前を文字としてグラフィック画面へ表示させています。

お絵かきエリアを0番の色で塗りつぶす

色選択パネル：タッチで色を選択するプログラム


- 8色分の色情報を下画面に表示して、タッチで選択するためのプログラムです
 - プログラムの一番下の行以降に以下のプログラムを追加します

```
000093) ? --- ↓
000094) GFILL 0, 0, 319, 219, COL[0] ↓
000095) RETURN ↓
000096) ↓
000097) ? ↓
000098) ? ■いろせんたく ↓
000099) ? ↓
000100) DEF SELCOLOR S, X, Y ↓
000101) IF S!=1 THEN RETURN ↓
000102) IF Y<220 THEN RETURN ↓
000103) ? --- ↓
000104) VAR I=FLOOR(X/40) ↓
000105) IF I==CL THEN RETURN ↓
000106) ? --- ↓
000107) CL=I:TALK VOICE#[CL] ↓
000108) END ↓
000109) ↓
```

色選択プログラム

下画面に表示された8種類の色選択パネルをタッチすると色を変更してTALK命令で色の名前を喋ります。描画用の色を管理する変数は、CLに入ります。

色選択パネル：プログラム実行

-  DIRECTボタンで実行モードに切り替えてプログラムを実行します

RUN ↓

- 色選択部分をタッチすると色が変わりますか？
- 色を変えた時コンピュータがしゃべりますか？

動かない時は「エラーが出てしまった人へ」のページを参考に修正！！

使った命令	利用方法
GOSUB	サブルーチンの呼び出し
DATA	数値や文字列をデータとして定義
READ	DATAから要素を取り出す
GPUTCHR	グラフィック画面へ文字を表示
FLOOR	数値から整数部分を取り出す
TALK	文字列を喋らせる



★ただしく動いたら「使いにくい消しゴム」を追加してみましょう！

使いにくい消しゴムプログラムを追加

- 傾きセンサーを使ってみましょう
- センサーの入力値は不安定なので消しにくい消しゴムです

ニンテンドー3DSに入っているモーションセンサーについて

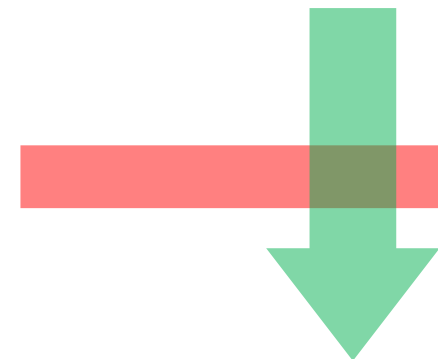
- ニンテンドー3DSには加速度と傾き(角度)を検知するセンサーが搭載されています
 - プチコン3号ではACCEL命令とGYRO系の命令で値を取得することができます
 - 消しゴムプログラムで利用するACCEL X,Y,Z で得られる値は以下の情報です

Xの値として入ってくる情報

- 平らな場所に置いた時の数値は0
- 軸↑を中心に右に傾けるとプラス値
- 軸↑を中心に左に傾けるとマイナス値

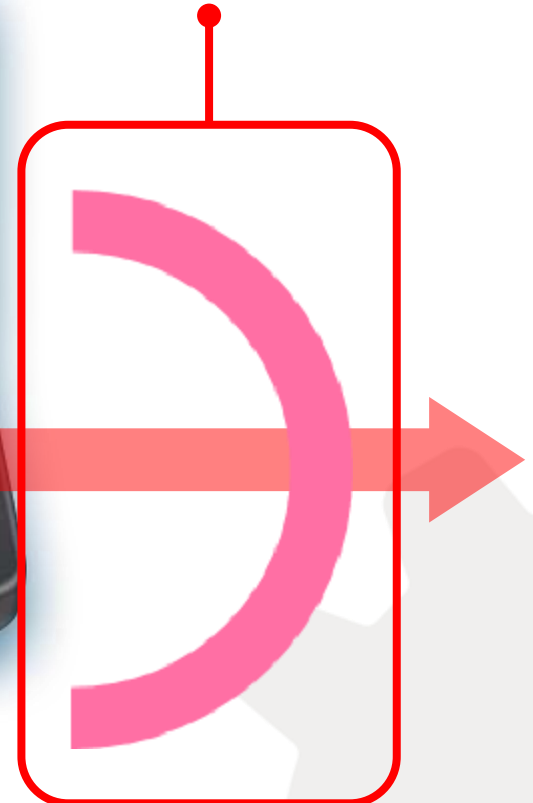
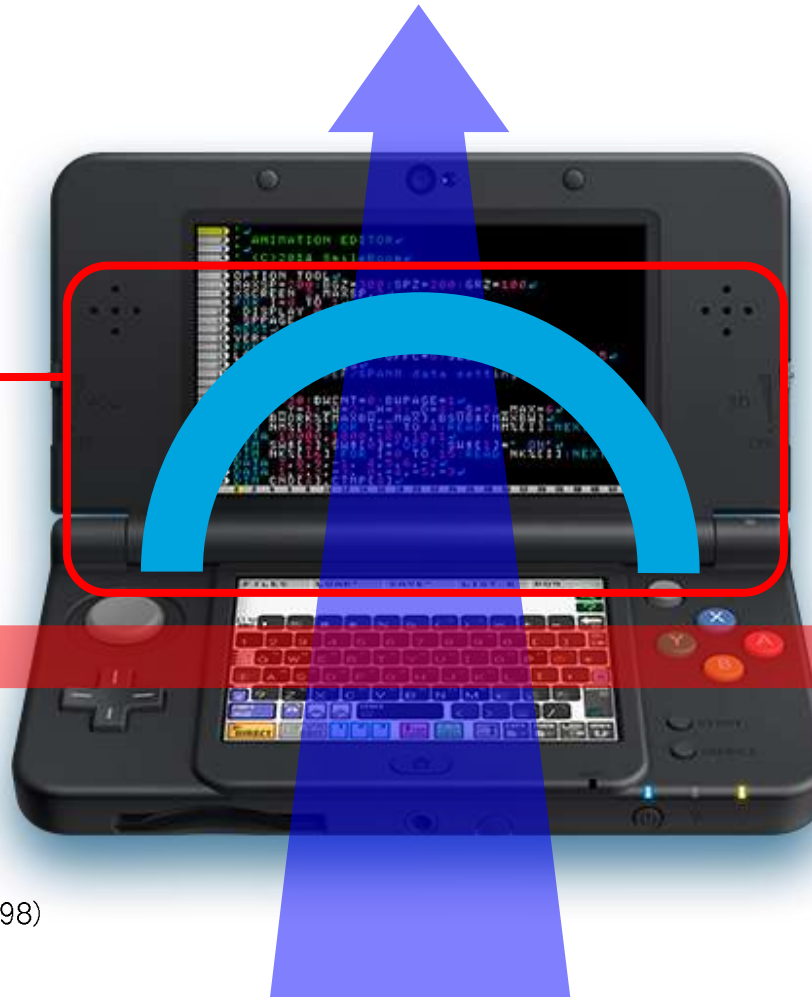
Yの値として入ってくる情報

- 平らな場所に置いた時の数値は0
- 軸→を中心に奥に倒すとプラス値
- 軸→を中心に手前に傾けるとマイナス値



Zの値として入ってくる情報

- 平らな場所に置いた時の数値は1G (約0.98)
- Zの値は消しゴムでは使いません

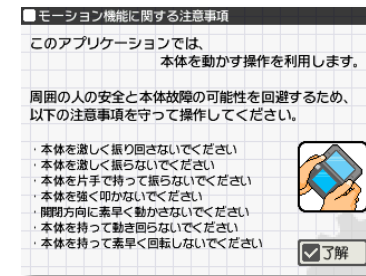


使いにくい消しゴム：新しい初期化等のプログラムを挿入

- プログラム先頭の方にモーションセンサーを使う宣言をする命令を追加

```
000001 ?  ↵
000002 ?  █がくしゅうサンプル 2015.9 ↵
000003 ?  ↵
000004 OPTION STRICT ↵
000005 ACLS:XSCREEN 2,256,2 ↵
000006 XON MOTION 'モーションつかう' ↵
000007 ?  --- ↵
000008 PRINT "#へんすうしょきが" ↵
```

モーションセンサーを使うための
おまじない（1回だけ注意事項表示）



- メインループ近くに初期化、加速度情報取得、消しゴムプログラムの呼び出しを追加

```
000031 ?  --- ↵
000032 VAR HD=0,HTOP=16,SMAX=32-HTOP ↵
000033 GOSUB @INIT_KESIGOMU ↵
000034 ?  ===== ↵
000035 ?  ↵
000036 ?  █×インループ ↵
000037 ?  ↵
000038 PRINT "#×インループ" ↵
000039 @LOOP ↵
000040 BTN=BUTTON(0):SFT=BTN AND &H300 ↵
000041 TOUCH OUT TS, TX, TY ↵
000042 ACCEL OUT AX, AY, AZ ↵
000043 ?  --- ↵
000044 DRAWSUB TS, TX, TY ↵
000045 KESIGOMU ↵
000046 SELCOLOR TS, TX, TY ↵
000047 ?  --- ↵
000048 VSYNC 1 ↵
000049 GOTO @LOOP ↵
000050 ↵
```

消しゴムの初期化プログラム呼び出し

加速度センサーからの情報を取得

消しゴムプログラムの呼び出し

使いにくい消しゴム：初期化

- スプライトを使って消しゴムを表示させます
 - 無駄に残像を表示して余計な演出表現を行います

```
000112) * --- ↓
000113) CL=I:TALK VOICE#[CL] ↓
000114) END ↓
000115) ↓
000116) * ===== ↓
000117) * ↓
000118) * ■ケシゴムしゅりしょきか ↓
000119) * ↓
000120) @INIT_KESIGOMU ↓
000121) PRINT " #けしごむ(DISPLAY 1/したがめん)" ↓
000122) DISPLAY 1 ↓
000123) SPSET 0, 212 ↓
000124) SPOFS 0, 160, 120, -100 ↓
000125) SPHOME 0, 8, 8 ↓
000126) SPSCALE 0, 2, 2 ↓
000127) * --- ↓
000128) PRINT " #ざんぞう" ↓
000129) FOR I=0 TO SMAX-1 ↓
000130) SPSET I+HTOP, 212 ↓
000131) SPHIDE I+HTOP ↓
000132) SPOFS I+HTOP, 160, 120 ↓
000133) SPHOME I+HTOP, 8, 8 ↓
000134) SPSCALE I+HTOP, 2, 2 ↓
000135) SPCOLOR I+HTOP, RGB(63, 255, 255, 255) ↓
000136) NEXT ↓
000137) RETURN ↓
000138) ↓
```

消しゴム本体用のスプライト初期化
消しゴム本体の表示用スプライト(0番)を初期化して、表示位置を下画面の中央(160,120)、奥行きを-100(手前に飛び出る)、表示原点を8,8へ変更し、スケールを2倍に設定。

消しゴムの残像用スプライト初期化
残像を記録するためのスプライト領域を初期化し、非表示状態としている。SPCOLORを使って透明度を64に変更。

使いにくい消しゴム：消しゴム処理本体

- 傾きセンサーの値を使って消しゴムを動かします
 - 動いたら消しゴムの後ろを黒い四角で塗りつぶします（絵を消す）

```
000138 '  ↓
000140 '  ■かたむきケシゴム ↓
000141 '  ↓
000142 DEF KESIGOMU ↓
000143 DISPLAY 1 ↓
000144 VAR I=0, X, Y ↓
000145 ' --- ころがる ↓
000146 SPOFS I OUT X, Y ↓
000147 X=X+(AX*16) ↓
000148 IF X<8 THEN X=8 ↓
000149 IF X>311 THEN X=311 ↓
000150 Y=Y+(AY*16) ↓
000151 IF Y<8 THEN Y=8 ↓
000152 IF Y>219 THEN Y=219 ↓
000153 SPOFS I, X, Y, 0 ↓
000154 ' --- けす ↓
000155 GCLIP 1, 0, 0, 319, 219 ↓
000156 I=12 ↓
000157 GFILL X-I, Y-I, X+I, Y+I, COL[0] ↓
000158 ' --- ざんぞうきる ↓
000159 SPSHOW HD+HTOP ↓
000160 SPOFS HD+HTOP, X, Y, 10 ↓
000161 HD=(HD+1) MOD SMAX ↓
000162 END ↓
000163 ↓
```

加速度センサーの値から消しゴムを動かす

AX,AYに格納されている加速度センサーの値を16倍して消しゴム座標に加算、描画エリアの少し内側で座標がはみ出ないように調整し新しい座標として消しゴム本体用のスプライト座標を更新。

X,Y座標を中心として、I(アイ)で指定されたサイズを12ドット分のエリアをGFILL命令を使って0番カラーで塗りつぶす。

今回のX,Y座標を新しい残像情報として記録

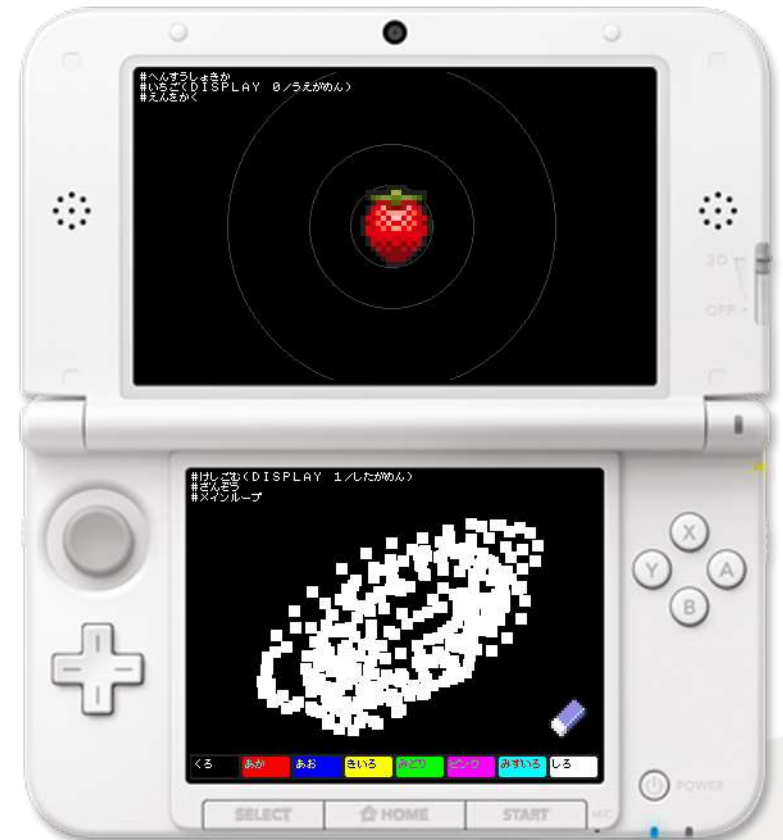
使いにくい消しゴム：プログラム実行

-  DIRECTボタンを押して実行モードに切り替えます

RUN ↓

- 本体を傾けると消しゴムが移動しますか？
- 消しゴムが通過した部分が黒く消えますか？

動かない時は「エラーが出てしまった人へ」のページを参考に修正！！



使った命令	利用方法
XON MOTION	センサーの利用宣言
ACCEL	センサーの情報取得
SPHIDE	スプライトを見えなくする
SPCOLOR	スプライトの色情報変更
GCLIP	グラフィック画面の描きこみ範囲制限
SPSHOW	スプライトを見えるようにする
MOD	割り算の余りを求める

★時間が残っていたら「上画面を派手にする」を追加してみましょう！

上画面を派手にするプログラムを追加

- 時間があったら入力してみよう！
- 上画面のイチゴの周りに派手な装飾を表示させます
- このような表現をパーティクルと呼びます

⑤

上画面を派手にする：星くずプログラムの呼び出しを2か所追加

- メインループ内に星くずを消すためのプログラムを挿入

```
000043) ? --- ↓  
000044) DRAWSUB TS, TX, TY ↓  
000045) HOSIDEAD ↓  
000046) KESIGOMU ↓  
000047) SELCOLOR TS, TX, TY ↓  
000048) ? --- ↓  
000049) VSYNC 1 ↓  
000050) GOTO @LOOP ↓  
000051) ↓
```

星くず消去プログラムの呼び出し

- 線を引くプログラム内に星くずを生成するプログラムを挿入
 - タッチで線を引いた方向と反対側へ星くずが生成されます

```
000060) ? --- せんをひく ↓  
000061) DISPLAY 1 ↓  
000062) GCLIP 1, 0, 0, 319, 219 ↓  
000063) IF SFT && S==1 THEN ↓  
000064) GPAINT X, Y, COL[CL] ↓  
000065) ELSE ↓  
000066) GFILL X-4, Y-4, X+4, Y+4, COL[CL] ↓  
000067) BEEP 27, 0, 20 ↓  
000068) ENDIF ↓  
000069) ? --- ↓  
000070) HOSIKUZU X, Y ↓  
000071) ? --- ↓  
000072) OX=X:OY=Y ↓  
000073) END ↓  
000074) ↓
```

星くずを1つ発生させるプログラムの呼び出し

上画面を派手にする：星くずの生成プログラム

- 1回で17個の星くずを発生させます
 - プログラムの一番最後に以下を追加してください

```
000167 ' =====
000168 '
000169 ' ■ほしくずせいせい
000170 '
000171 DEF HOSIKUZU X, Y
000172 VAR I
000173 ' --- へんかりょうをけいさん
000174 VAR VX=X-OX, VY=Y-OY
000175 IF ABS(VX)<2 THEN VX=0
000176 IF ABS(VY)<2 THEN VY=0
000177 IF VX==0 && VY==0 THEN RETURN
000178 ' --- うえのいちご
000179 VAR R=90+DEG(ATAN(VY, VX))
000180 DISPLAY 0:SPROT 0, R
000181 ' --- ほしくずはっしや
000182 HOSI1 R+90, X, Y, 90
000183 FOR I=1 TO 8
000184 HOSI1 R+90+I*2, X, Y, RND(60)+30
000185 HOSI1 R+90-I*2, X, Y, RND(60)+30
000186 NEXT I
000187 END
000188
```

直前の座用OX,OYと新しい座標X,Yの差分から最新の移動方向をVX,VYに求める。差分が少ない場合は生成させない。

上画面のイチゴの角度を変化させる

基準となる星くずを1つだけ生成

基準となる星くずの角度から±方向に少しずつ広げながら8×2個分の星くずを生成する。

上画面を派手にする：星くず1つ分の生成プログラム

- 発生した星くずが画面外に出たら消す処理を登録
 - 続けてプログラムの最後の行以降に以下を入力します

```
000188 ' ↵
000190 ' ■ほしくずを1つせいせい↵
000191 ' ↵
000192 DEF HOSI1 R, X, Y, S↵
000193 VAR I, U, V, W, H, A↵
000194 ' ---↵
000195 DISPLAY 0↵
000196 SPSET 1, 255, RND(1024) OUT I '1361↵
000197 IF I == -1 THEN RETURN↵
000198 ' ---↵
000199 SPCHR I OUT U, V, W, H, A↵
000200 SPCHR I, U, V, W, H, A OR &H20↵
000201 ' ---↵
000202 X=X+(400-320)/2↵
000203 SPOFS I, X, Y, 512↵
000204 SPHOME I, W/2, H/2↵
000205 ' ---↵
000206 R=RAD(R)↵
000207 X=X+COS(R)*512↵
000208 Y=Y+SIN(R)*512↵
000209 SPROT I, R↵
000210 SPANIM I, "XY", -S, X, Y↵
000211 SPANIM I, "Z", -30, -200↵
000212 SPANIM I, "C", -S, RGB(0, 255, 255, 255)↵
000213 SPANIM I, "R", -S, R+360↵
000214 SPANIM I, "S", -S, 3, 3↵
000215 END↵
000216 ↵
```

スプライトに空きが無い時は生成しないで終了

生成したスプライトの情報を取得しアトリビュートに加算合成を追加

生成したスプライトの座標を上画面用に調節し基準点を中心へ

与えられた角度から三角関数を使って移動量を計算し、
値を512倍することで画面外の座標に変換しX, Yに格納

求めた目的座標X, YへSPANIMで変数Sで受け取ったフレーム時間でなめらかに移動させる設定。ついでに同じ時間で色は透明へ、角度は360度回転、スケールは3倍にするアニメを設定する。

上画面を派手にする：星くずの消去プログラム

- 発生した星くずが画面外に出たら消す処理を登録
 - 続けてプログラムの最後の行以降に以下を入力します

```
000217  '  '
000218  ' ■がゆんがいへてたけす  '
000219  '  '
000220  DEF HOSIDEAD  '
000221  DISPLAY 0  '
000222  VAR I, X, Y  '
000223  ' ---  '
000224  FOR I=0 TO 255  '
000225  IF SPUSED(I) THEN  '
000226  SPOFS I OUT X, Y  '
000227  IF X<0 || X>399 || Y<0 || Y>239 THEN  '
000228  SPCLR I 'がゆんがいのときけす  '
000229  ENDIF  '
000230  ENDIF  '
000231  NEXT  '
000232  END  '
000233  '  '

```

画面外へはみ出たスプライトを検査

256個のスプライトを順番に検査し、使われているスプライトが画面外に出ている場合、SPCLRで未使用状態に開放する。

無駄に上画面を派手にする：プログラム実行

-  DIRECTボタンを押して実行モードに切り替えます

RUN ↓

– 線を描くと上画面に派手な演出が表示されましたか？

動かない時は「エラーが出てしまった人へ」のページを参考に修正！！

使った命令	利用方法
ABS	絶対値の取得
DEG	ラジアンから角度への変換
ATAN (VY, VX)	移動量から角度(ラジアン)を求める
SPROT	スプライトの角度を変更
RND	ランダムな数値を得る
SPCHR	スプライトの表示情報を設定
RAD	角度からラジアンへの変換
COS	角度(ラジアン)からコサイン値を得る
SIN	角度(ラジアン)からサイン値を得る
SPANIM	スプライトにアニメーションを登録
SPUSED	スプライトの利用状況を確認
SPCLR	スプライトを消去



★おつかれさまでした！

付録：プチコン3号資料

- TOPMENU関係の機能紹介
- プチコン3号の仕様

どんな作品が作れるかサンプルを見てみよう（調査&研究）

- TOPMENU「サンプルを見る」および「作品を見る」で作品を見ることが出来ます
- ファイル選択画面から気になるファイルを選んで「決定」を押すと実行します

– 簡単なプログラム

- EX1TEXT、EX2CALC、EX3KEDRUM、EX4NUMGAME、EX5BIORHYTHM、EX6SEQUENCER、EX7ALIEN

– すべての命令をテストするプログラム

- EX8TECHDEMO

– ゲーム

- GAME1DOTRC、GAME2RPG、GAME3JUMP、GAME4SHOOTER、GAME5VS、GAME6TALK、GAME7EXPAD

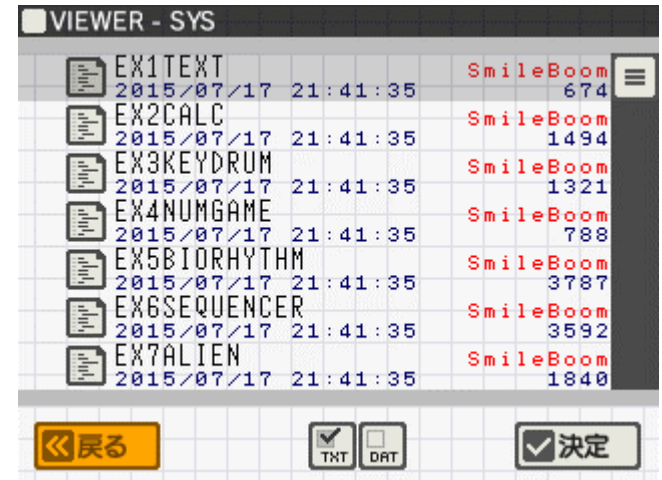
– 漢字の文字サンプル（実行できません）

- KANJITBL

– ツール

- SBANM、SBGED、SBSMILE、SBWAV

– STAFFROLL




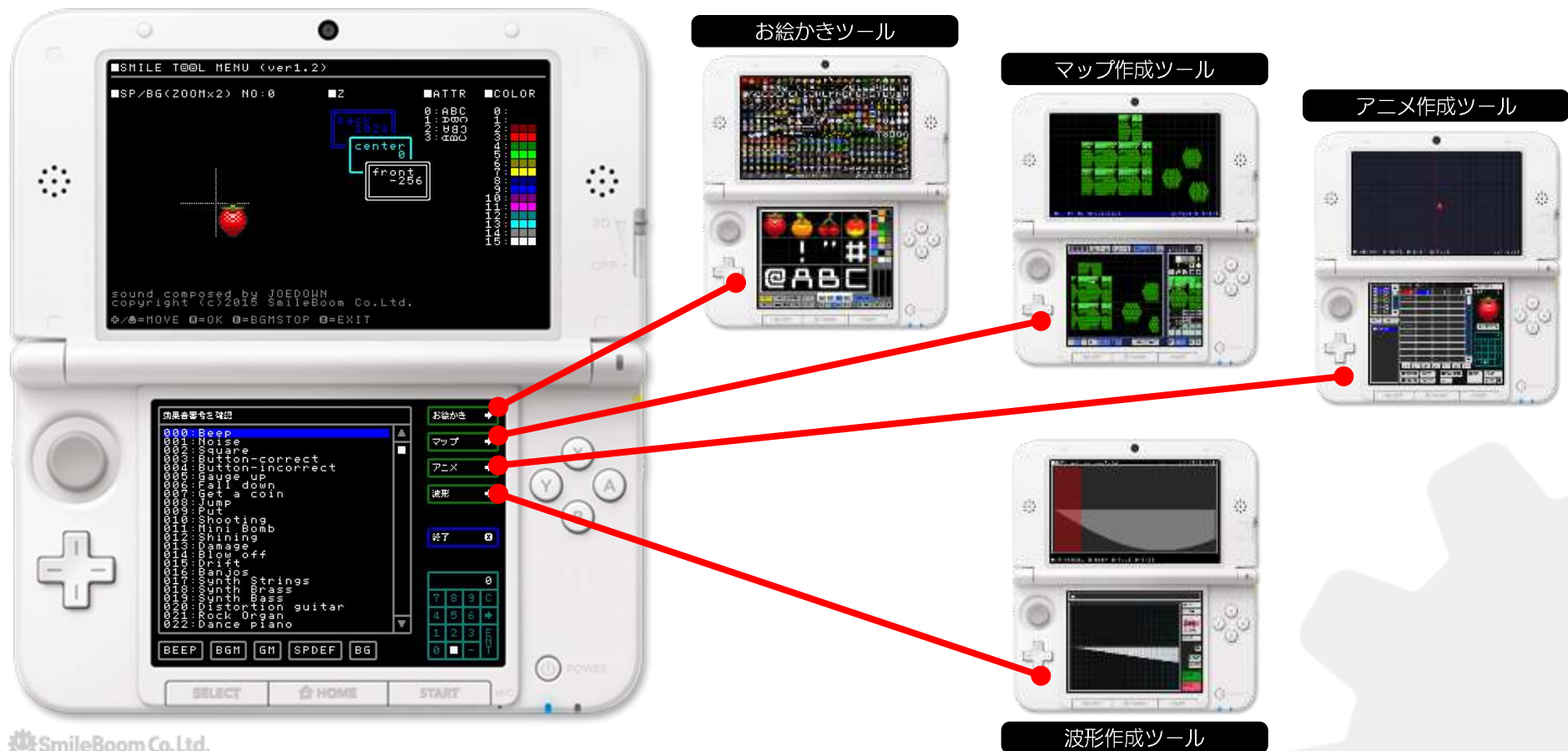
実行直後のサンプルプログラムを見る方法

1. TOPMENU「SmileBASICでプログラムを作る」からBASICへ
2. EDITボタンを押して直前まで実行されていたプログラムを表示
3. DIRECTモードからRUNで実行できます

※作品によってはプログラム実行時に画像データ等を読み込むものがあります。実行時にエラーが出る場合、TOPMENUの「プロジェクトとファイルの管理」からカレントプロジェクトを変更してください。

SMILEボタンの役割 (BEEP・BGM・SPDEF確認、ツール)

- キーボード上の  SMILEボタンを押すことで標準ツールが起動します
- 効果音やBGMやスプライトの番号を確認することができます
- お絵かき、マップ、アニメ、波形作成ツールを呼び出すこともできます
- ツールを実行しても作業中のプログラムは消されずに残ったままとなります



公開されているサンプルを手に入れよう（調査&研究）

インターネットに接続できる環境であれば公開キーで作品を手に入れます

- Miiverse上の作品
 - 公式キャラ「はかせ」の倉庫 <https://miiverse.nintendo.net/users/smileDr/posts>
- 公式サイトにある公開キー置き場
 - 大喜利の作品 <http://smileboom.com/special/ptcm3/ogiri/>
 - トランプライブラリ <http://smileboom.com/special/ptcm3/publickey/>
- プチコンユーザーによる情報サイト
 - ぷちこん3号まとめWiki <http://wiki.hosiken.jp/petc3gou/>

Twitter や WEB検索等から「プチコン」 「#petitcom」 「#smilebasic」 などで検索するとプチコン関係の情報を入手できます。

プチコン3号の画面を構成する要素

- 役割の異なる5つの描画面が重なって表示されています
- 画面上での実際の重なり方は各面のZ座標で決まります (Z座標：手前-256～奥1024)



定数

- システムで予約された利用頻度の高い数値に名前を付けたものを「定数」と呼びます
- 定数は内部で整数値に変換されます（利用者は値を気にする必要はない）
 - 例えば、ボタンのビットを表す #RIGHT は、&H0008 に置き換わります
 - DIRECTモードで "# " を入力してヘルプボタンを押すと定数の一覧と割り当てられた数値が確認できます

文字の色	グラフィック色	ボタンのビット	スプライト属性	アニメの状態	BG属性	一般
#TBLACK #TMAROON #TRED #TGREEN #TLIME #TOLIVE #TYELLOW #TNAVY #TBLUE #TPURPLE #TMAGENTA #TTEAL #TCYAN #TGRAY #TWHITE	#AQUA #BLACK #BLUE #CYAN #FUCHSIA #GRAY #GREEN #LIME #MAGENTA #MAROON #NAVY #OLIVE #PURPLE #RED #SLIVER #TEAL #WHITE #YELLOW	#UP #DOWN #LEFT #RIGHT #A #B #X #Y #L #R #ZL #ZR	#SPSHOW #SPROT0 #SPROT90 #SPROT180 #SPROT270 #SPREVV #SPREVV #SPADD	#CHKXY #CHKZ #CHKUV #CHKI #CHKR #CHKS #CHKC #CHKV	#BGSHOW #BGROT0 #BGROT90 #BGROT180 #BGROT270 #BGREVV #BGREVV	#ON #YES #TRUE #OFF #NO #FALSE

効果音の種類

1. Beep
2. Noise
3. Square
4. Button-correct
5. Button-incorrect
6. Gauge up
7. Fall down
8. Get a coin
9. Jump
10. Put
11. Shooting
12. Mini Bomb
13. Shining
14. Damage
15. Blow off
16. Drift
17. Banjos
18. Synth Strings
19. Synth Brass
20. Synth Bass
21. Distortion guitar
22. Rock Organ
23. Dance piano
24. High Tom
25. Low Tom
26. Crash Cymbal
27. DancedrumKit OpenHiHat
28. DancedrumKit ClosedHiHat
29. Clap
30. Snare Rim
31. Dancedrum KitSnare
32. Dancedrum KitKick
33. Button-Clear1
34. Button-Tsudumi
35. GOUKA KENRAN
36. Electricity
37. Wipe-up
38. Broken piece
39. Warp jump
40. Banjos-phrase
41. Scratch
42. Distortion guitar-phrase
43. Rock Organ-phrase
44. Dance piano-phrase
45. Car pass-synth
46. Count up
47. REC Noise
48. Synth Tom
49. Synth conga
50. Metronome normal
51. Metronome accent
52. Conga
53. Dancedrum KitKick2
54. Dancedrum KitSnare2
55. DancedrumKit OpenHiHat2
56. Orchestra Hit1
57. Timbals
58. China Cymbal
59. Chappa Cymbal
60. Shaker
61. Bell tree
62. Wadaiko
63. Synth Hit
64. Cuckoo 'Dove
65. Puff-Puff horn
66. Shinobue
67. Voicepercussion BOON
68. Voicepercussion Ah
69. Dog
70. Cat
71. Girl's Voice-OK
72. Girl's Voice-Yattane!
73. Girl's Voice-Omedetou!
74. Girl's Voice-ByeBye
75. Girl's Voice-Iyan
76. Girl's Voice-Kya!
77. Girl's Voice-Uwaaan
78. Girl's Voice-WAO!
79. Girl's Voice-Yahho
80. Waterdrop
81. Flame
82. Whip
83. Rock break
84. Raven
85. Gull
86. Stream
87. Baseball-Hit
88. Baseball-Catch
89. Audience-Dejection
90. Audience-Cheer
91. Applause
92. Badminton-Smash
93. Soccer-Shoot
94. Fan noise light
95. Fan noise heavy
96. Dig
97. Whistle-short
98. Whistle-long
99. Frog
100. Door
101. Ignition
102. Steam
103. Faint away
104. Slash
105. Flap
106. Funny Bomb
107. Button-Clear2
108. Up&Down
109. Large explosion
110. Dance synth-phrase
111. Mini Drill
112. Drill spin
113. Finger Snap
114. Result Jingle-Synth
115. Result Jingle-Gothic
116. Vanish
117. Button-start
118. Button-usually2
119. Item get-power up
120. Item get-status up
121. Cannon-synth
122. Alert
123. wobblebass-down
124. wobblebass-up
125. Machine crash
126. Burner boost
127. Robot-moving
128. Robot-shining eyes
129. Robot-wakeup
130. Vocorder-a
131. Vocorder-i
132. Vocorder-u
133. Vocorder-e
134. Vocorder-o

BGMの種類

1. Kung-Fu POP
2. With stealthy steps
3. Flat out run
4. Nostalgia TECHNO
5. Feel easy
6. Have a good time
7. Relief
8. Exciting days
9. Skipping march
10. Valiant departure
11. Important thing
12. Chasing at 'Ooedo'
13. Funny land
14. Step on the accelerator
15. Experiment
16. New discovery
17. Thinking time
18. Mischievous boy
19. Float
20. Sound of the surf
21. Sound of the surf2
22. Spy movie
23. Calculating
24. Take Off!
25. The evening moon.
26. Sensibility
27. Pure water
28. Strategy
29. cure
30. Intense battle
31. Keen competition
32. Heat uuuup!!
33. Rise with force
34. Bright blue
35. Storyteller
36. Return trip
37. High spirits
38. Welcome to the party
39. Funky claps
40. Night surfer
41. Ready to FLY
42. We are heroes
43. Pure water2

付録：困ったときのアドバイス

- プチコンを使っていると必ず起こる問題の解決方法
- プチコンに関する情報の集め方

MMLを使って自分で作った曲を演奏させる

- プチコン3号にはMML (MusicMacroLanguage) が搭載されています
- この機能を使うと自分で作った曲をBGMや効果音として利用できます
 - 以下を実行するとピアノとドラムが同時に演奏されます

```
BGMPLAY " : 0CDEFGAB<C2:1@12902L8[CEFG]4" ↓
```

• 主なMMLコマンド

■曲全体を制御するコマンド

チャンネル指定 :0~:15(コロソ:に続けて番号)
テンポ指定 T1~T512

■音の長さに関するコマンド

デフォルト音長指定 L1~L192
音長個別指定 音程直後に数値(1/2/4/8/16)
奏法_(前後の音をつなぐ)
ゲート時間 Q0~Q8

■音程(音の高さ)に関するコマンド

C(ド), D(レ), E(ミ), F(ファ)
G(ソ), A(ラ), B(シ)
半音上げる C# D# E# F# G# A# B#
半音下げる C- D- E- F- G- A- B-
休符 R
オクターブ指定 O0~O8
オクターブを1つ上げる <
オクターブを1つ下げる >
オクターブ指定の反転!
キーを数値指定 N0~N127 ※O4C=60

■音量や定位に関するコマンド

音量指定 V0~V127
音量を1つ上げる ()
音量を1つ下げる ()
パンポット P0~P127(P64=中央)
エンベロープ設定 @E数値A,D,S,R
A(Attack time):0~127
D(Decay time):0~127
S(Sustain level):0~127
R(Release time):0~127
※各timeは小さいほど遅い

■音色変更コマンド

楽器音変更 @0~@127(GM音源相当)
標準ドラム @128
電子ドラム @129
PSG音源 @144~@150
ノイズ音源 @151
ユーザー定義波形 @224~@255

■特殊な演奏コマンド

リピート開始 []
リピート終了]回数 ※回数省略時無限

音が何だか変です

- 鳴りっぱなしのBGMを止めたい

```
BGMSTOP ↓
```

- 洞窟？風呂場？のような凄いエコーがかかったような聞こえ方をする

```
EFCOFF ↓
```

- キーボード入力時やボタンを押したときにクリック音が鳴らなくなった

```
SYSBEEP=1 ↓
```

– SYSBEEPは、システムが鳴らす効果音を制御するためのシステム変数です

なにも見えませーん！そんな時は「ACLS」

- いろいろさわっていると画面がゴチャゴチャになることがあります
 - 他の人のプログラムを実行したらフォントが壊れた
 - グラフィックやスプライトで画面が埋め尽くされて見えなくなった
 - 画面が一色で何も見えなくなった
 - 文字がスプライトなどのうしろ側に表示された
- このような状況になったら以下の命令をダイレクトモードから入力してください

ACLS」

- そもそも文字が見えない状態になっている場合・・・
 - 見えない状態のまま根性でACLSを入力
 - スマイルボタンを押して「終了」を押す（グラフィック、スプライト、BG、FADEなどの状態は元に戻ります）

まちがってSAVEで上書きしてしまった！！

- 直前のファイルは、@BACKUP.PRG というファイルに残っています
- RENAME 命令を使って復活させましょう

```
RENAME "@BACKUP.PRG", "REBORN" ↵
```

- これで、上書きする前のファイルが "REBORN" という名前で復活しました。

保存されているファイル内容を確認

- これまでに保存されているファイルの一覧を表示

```
FILES ↓
```

– ABC、XYZ、BBBという3個のファイルが保存されていた場合の結果（残り容量は各自で異なる）

```
FILES
*ABC
*BBB
*XYZ
-- 14874144 kbytes free --
OK
```

改造のヒント

- 時間が残っている人は以下の改造方法を考えて変更してみてください
 1. 色を変更したらBGMが変化する機能 (BGMPLAY)
 2. 四角の代わりに円を描画させる (GCIRCLE)
 3. 下画面にエフェクトが出るようにしてみる (DISPLAY)
 4. 描いたときに画面の左右で聞こえ方が変わるように音を鳴らす (BEEP)
 5. 消しゴムを使いやすくする (STICK)
 6. ランダムに変化する色を作る (RND、RGB、BUTTON)
 7. 描いた線がなめらかにつながるようにする (GLINE、直前の座標記憶)
 8. 全部消すボタンの追加 (BUTTON)
 9. ときどき飛んでくるハエをつぶす (RND、SPSET、SPOFS、SPHIT)

などなど・・好きなように改造して面白いソフトに変化させましょう！



発行：株式会社スマイルブーム (C)2015 SmileBoom Co. Ltd.

不具合やご質問はメールにてご連絡ください info@smileboom.com