



この文書は2011年11月5日に「文化庁メディア芸術祭 京都展」
エンターテインメント部門「ゲームってアートなの？——エンターテインメントのいま・これから」内
ワークショップ「ゲームはどうやって動くのか？——ニンテンドーDSと『プチコン』で遊ぼう！」にて
配布されたものに加筆・訂正を行なったものです。

2011年11月8日 株式会社スマイルブーム

ゲームはどうやって動くのか？ ——ニンテンドーDSと『プチコン』 で遊ぼう！

文化庁メディア芸術祭 京都展「パラレルワールド京都」

2011.11.5 ワークショップ (15:30~17:00)

ワークショップ方針

* 想定している受講者像

- * プログラムのことは知らないけど、ちょっと興味がある
- * BASIC（ベーシック）は見たことも触ったことも無い
- * ゲーム機やスマホや携帯などでゲームはする
- * DSは知っているし触ったこともある

* 説明方針

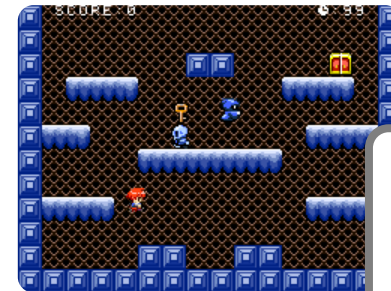
- * 派手な3Dやゲームっぽいキャラクタは使わずシンプルな表現にとどめます
- * 難しいことは詳しく説明しません（理論とか詳細仕様など・・・）
- * 資料から文字を入力してプログラムを動かすことを最優先とします
- * 動いた後に改造すると面白い要素について説明します

ワークショップ進行予定

- * 講師紹介 (15:30~)
- * ①コンピュータで遊ぶ (15:35~)
 - * 1-1 コンピュータとBASICとPチコン
 - * 1-2 コンピュータ (DSi) へ命令してみる
 - * 1-3 一度にたくさん命令する
 - * 1-4 画面と文字の関係
- * ②お絵かきツールの開発 (15:45~)
 - * 2-1 タッチ操作で線を引く
 - * 2-2 ボタン操作で機能を追加
- * ③無理やりゲームに改造 (16:15~)
 - * 3-1 大量に動くものを追加
 - * 3-2 初期化の追加
 - * 3-3 移動処理の追加
 - * 3-4 当たり判定を付ける
 - * 3-5 終了処理
- * おしまい

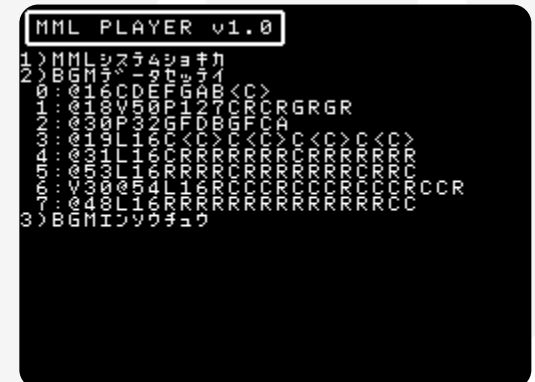


プチコンは3Dに対応していないため今回のワークショップでは3D表現のゲームは作りません

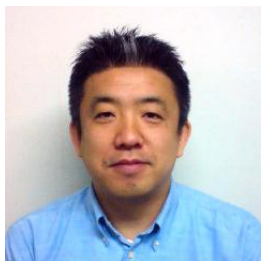


このゲームはプチコンで作られたキャラクターを使ったゲームですが、今回のワークショップではキャラクターも使いません。

コンピュータに命令することを実感してもらうため、今回のワークショップは基本的に文字中心で進める方針です。



講師紹介



20年以上前から札幌でゲームを作り続けています。
北海道でゲームを作る若者を増やすために活動中。

名前	小林貴樹（こばやしただかき）
所属	株式会社スマイルブーム
特技	面白い企画発案、簡単なプログラム作成
その他	CEDECアドバイザリーボードメンバー Microsoft MVP2011
関連作品	うっでいぽこ、ヴォルガード2、パワフル麻雀 俺の料理、ガチャろく、だれでもアソビ大全 アクションゲームツクール、プチコン

CEDEC2011
Computer Entertainment Developers Conference



コンピュータで遊ぶ

ニンテンドーDSもコンピュータです。

1

1-1 コンピュータとBASICとプチコン

* コンピュータは生活に溶け込んでいます

- * ゲーム専用機、携帯電話、スマートフォン
- * 家電製品、車、銀行のATMなど・・・

* コンピュータの役割

- * 数値の記憶と計算、条件比較と分岐
- * 入力操作受付、画面への表示
- * 0と1だけの機械語という命令で動く

* BASICの役割

- * 機械語を人間用にわかりやすくした命令言語

* プチコン

- * ニンテンドーDSi上で動くBASIC言語



1-2 コンピュータ (DSi) へ命令してみる

* 実行モード



- * BASIC命令を直接入力することができる画面
- * 実行モードからの入力を試してみよう
 1. 命令1-2Aの内容を下画面のキーボードから入力
 2. 入力が終わったら「Enter」キーを押す
 3. 上画面に円が描画される
 4. 続けて、命令1-2Bと命令1-2Cも試してみましよう

命令1-2A：画面に赤い円を描く

```
GCIRCLE 128, 96, 80, 2 ↵
```

命令1-2B：円の中をグレーで塗る

```
GPAINT 128, 96, 1 ↵
```

命令1-2C：図形画面を消す

```
GCLS ↵
```

```
PetitComputer フォン ver1.00  
SMILEBASIC 1048576 byte free  
(C)2011 SmileBoom Co.Ltd.  
  
composed by JOEDOWN  
developed by  
Rocket Studio, Inc.  
READY  
_
```

1 FILE. 2 LOAD. 3 SAVE. 4 CONT 5 RUN



全ての入力が終わったら「Enter」キーを押すと命令がコンピュータに伝わります。

1-3 一度にたくさん命令する

* 編集モード



- * コンピュータへの命令を並べて書く画面
- * 編集モードでの入力と実行を試してみよう
 1. 下画面にある「編集」ボタンを押す
 2. リスト1-3に書いてある通りに下画面のキーボードから入力
 3. 入力が終わったら下画面の「実行」ボタンを押す
 4. 命令1-3を入力して「Enter」キーを押す

リスト1-3：上画面に色見本表示

```
CLS
FOR Y=0 TO 15
  FOR X=0 TO 15
    LOCATE X+8, Y+4
    PRINT CHR$(Y*16+X)
  NEXT X
NEXT Y
```

行番号は気にしなくても良いです。

命令1-3：編集モードで入力した命令を実行する

RUN

```
101
102
103
104
105
106
107 @DISPINIT ?カ?メ?ン?シ?ョ?キ?カ?
108
109 CLS:COLOR 0
110
111
112
113
114 GPAGE 0:GCOLOR 0:GCLS
115
116
117
118 LOCATE 0,8
119 PRINT"
120 PRINT" CHARACTER EDITOR v1.0
121
122 GOSUB @CHRALL
123 LOCATE 0,23
124 PRINT "0=ファイルモード";
124
```



全ての入力が終わったら「実行」ボタンを押して実行モードに切り替えてください。

1-4 画面と文字の関係 (ついでに音も)

* 文字表示画面のしくみ

- * 横32文字、縦24文字表示できる
- * 文字の色は16色から選択可能

命令1-4A : 文字画面を消して右上の方にAを表示

```
CLS:LOCATE 28,5:PRINT"A" ↵
```

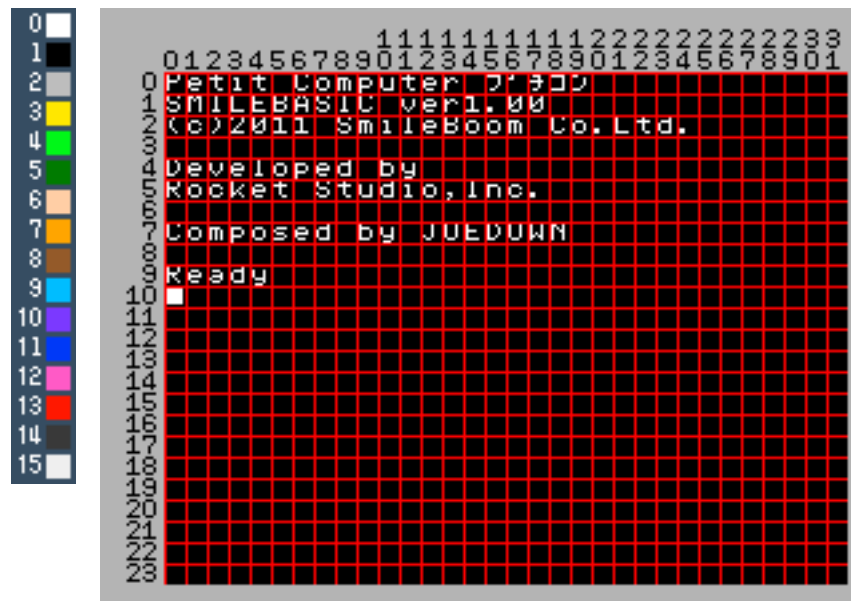
命令1-4B : 文字色を赤にしてNAMEを表示

```
COLOR 13:PRINT"NAME":COLOR 0 ↵
```

* 効果音を鳴らす

命令1-4C : 「ぽよ～ん」という音を鳴らす命令

```
BEEP 8 ↵
```



【 ヒント 】

- 音楽を演奏させる・・・ BGMPLAY 番号(0~29)
- 音楽を止める・・・ BGMSTOP
- 効果音の周波数変更・・・ BEEP 番号(0~69), 周波数(±8192)

お絵かきツールの開発

DSのタッチパネルを使ってお絵かきしてみよう！
プログラムの意味が分からなくても気にせず入力しましょう。

2

【 ヒント 】

危険な命令「NEW」を実行モードで入力すると編集モードに登録してあったプログラムがすべて消えてしまいます。

2-1 タッチ操作で線を引く

* 線を引くための予備知識

- * グラフィックは横256ドット、縦192ドット、256色のドットが打てます
- * DSiのタッチ情報は、TCHST、TCHX、TCHYの変数に入ってきます
- * タッチされるとTCHSTに1が入ります（通常は0）

* 線を引くための動きを考えておく

1. 新しくタッチされた場所を記憶しておく
2. 次のタッチ入力を少し間を空けて待つ（1/60秒）
3. 新しいタッチ位置と記憶された位置の間に線を引く
4. タッチが放されるまで2からを繰り返す
5. 次にタッチされたら1へ戻る

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	<input checked="" type="checkbox"/>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
1	<input checked="" type="checkbox"/>	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
3	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
4	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
5	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
6	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
7	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
8	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
9	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
A	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
B	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
D	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
E	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
F	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

【 ヒント 】

プログラムの中で色番号を管理している変数は、COLです。
COL=15で白、COL=2で赤、COL=4でピンク。上の色見本が
色番号の順番に並んでいます。左上が0番、右上が15番、1段下
がった左2段目が16番、右下が255番となっています。

リスト2-1：タッチ操作で線を引く

```
CLEAR
'-----
@RESET
GOSUB @DISPINIT
OX=0:OY=0:OT=0:COL=15
'-----
@LOOP
TX=TCHX:TY=TCHY
GOSUB @SETLINE
VSYNC 1
GOTO @LOOP
'-----
@SETLINE
IF TCHST==0 THEN OT=0:RETURN
IF OT==0 THEN OX=TX:OY=TY
'---
@LINE
GLINE OX,OY, TX, TY, COL
OX=TX:OY=TY:OT=1
RETURN
'-----
@DISPINIT
CLS
PNLTYPE"OFF"
GPAGE 1:GCLS 0
SPPAGE 1:SPCLR
RETURN
```

正しく入力されたら実行モードからRUNで実行。
下画面にタッチで白い線が描けるようになります。
終わるときは、**SELECTボタン**を押してください。

変数名	用途
TCHX,TCHY	システムが用意しているタッチ座標
TCHST	システムが用意しているタッチ状態 (0=押される前、1=押されている)
TX,TY	TCHX,TCHYの保管用(ただのコピー)
OX,OY	座標の記憶用
OT	初めて触られたかの管理用 (0=初めて、1=すでに触られている)
COL	色番号

ラベル	用途
@RESET	初期化開始位置
@LOOP	メインループ
@SETLINE	線を引く処理
@LINE	線を引く処理
@DISPINIT	画面関係の初期化処理

命令2-1：編集集中のプログラムを保存

```
SAVE "LIST201"
```

ファイル名に“-”が使えないので201とした。

2-2 ボタン操作で機能を追加（1/3）

* Xボタンが押されたら描いた絵を消去する機能を追加

- * ボタン情報を取得する処理を挿入

命令2-2A：挿入位置に近い部分を編集

```
LIST @LOOP ↵
```



リスト2-2A：ボタン受付処理を挿入

```
      :  
@LOOP ↵  
OLDBT=BT:BT=BUTTON( ) ↵  
TX=TCHX:TY=TCHY ↵  
GOSUB @SETLINE ↵  
      :
```

数値	対応するボタン
1	十字ボタンの上
2	十字ボタンの下
4	十字ボタンの左
8	十字ボタンの右
16	Aボタン
32	Bボタン
64	Xボタン
128	Yボタン
256	Lボタン
512	Rボタン



変数名	用途
BT	DSiのボタンの状態
OLDBT	1つ前のボタンの状態

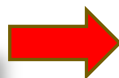
※次のページに続きます

2-2 ボタン操作で機能を追加 (2/3)

- * Xボタンに対応する処理を挿入

命令2-2B : 挿入する位置に近い部分を編集

```
LIST @SETLINE ↵
```



リスト2-2B : Xボタンの押下判断処理挿入

```
:
@SETLINE ↵
IF BT==64 THEN @CMDCLS ↵
IF TCHST==0 THEN OT=0:RETURN ↵
IF OT==0 THEN OX=TX:OY=TY ↵
' --- ↵
:
```

- * 画面を消去する機能を追加

命令2-2C : 挿入位置に近い部分を編集

```
LIST @DISPINIT ↵
```



リスト2-2C : 画面を消す処理追加

```
:
' --- ↵
@CMDCLS ↵
GCLS 0 ↵
RETURN ↵
' ----- ↵
@DISPINIT ↵
:
```

※次のページに続きます

ラベル

用途

@CMDCLS

画面消去

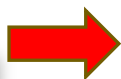
2-2 ボタン操作で機能を追加 (3/3)

* 十字ボタンで色を変更する機能を追加

- * ボタンに対応する色を設定

命令2-2D : 挿入位置に近い部分を編集

```
LIST @SETLINE ↵
```



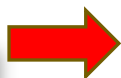
リスト2-2D : 十字ボタンの押下判断処理挿入

```
:
@SETLINE ↵
IF BT==1 THEN COL=2 ↵
IF BT==2 THEN COL=4 ↵
IF BT==4 THEN COL=11 ↵
IF BT==8 THEN COL=15 ↵
IF BT==64 THEN @CMDCLS ↵
:
```

- * 色の情報を上画面に表示

命令2-2E : 挿入位置に近い部分を編集

```
LIST @LOOP ↵
```



リスト2-2E : 色の番号表示

```
:
@LOOP ↵
LOCATE 0,0 ↵
PRINT "COLOR:";COL;" " ↵
OLDBT=BT:BT=BUTTON( ) ↵
TX=TCHX:TY=TCHY ↵
:
```

命令2-2F : 編集集中のプログラムを保存

```
SAVE "LIST202" ↵
```


おまけ 1

このページは少し余裕がある人向けのおまけです。
他の人の作業が終わるまで時間がある人は
入力して機能を追加してみましょう。

* 時間がある人は、他の機能にも挑戦

* Lボタンが押されてたら塗りつぶし

リスト2-2F : Lボタン対応と塗りつぶし処理挿入

```
      :  
@LINE↵  
IF BT AND 256 THEN @PAINT↵  
GLINE OX, OY, TX, TY, COL↵  
OX=TX:OY=TY:OT=1↵  
RETURN↵  
' ---↵  
@PAINT↵  
GPAINT TX, TY, COL↵  
RETURN↵  
      :
```

【訂正】

印刷された資料の64ではなく256が正しい値でした。
入力される方は64を256に置き換えてください。

無理やりゲームに改造

お絵かきツールを改造してゲームっぽくしてみよう！
かなり強引だけど、自分だけの勝手なルールが楽しい

3

【 ヒント 】

スプライトに関しては詳しい説明を省略しています。下画面で使えるキャラクタは右図のような画像が登録されています。



3-1 大量に動くものを追加

* 大量に物を動かすための予備知識

- * 画面に表示するための座標と移動量などを複数管理（配列という変数）
- * くりかえし処理で同じプログラムを回数分実行する
- * 移動させるのが楽なスプライトという表示機能を利用する

* 大量に物を動かすための動きを考えておく

1. 必要な回数分の配列型の変数を確保しておく
2. 必要な回数分の座標や移動量を初期化
3. メインループから物を動かす処理を呼び出す
4. 必要な回数分の座標を変化させて表示位置を変更

【 ヒント 】

配列を扱う命令として、DIM命令が用意されています。
DIM X(10), Y(10) のように書くと10個分の変数が確保されます。

3-2 初期化を追加

* 配列変数を確保を挿入

命令3-2A : 挿入位置に近い部分を編集

```
LIST @RESET
```

* 初期化処理を挿入

命令3-2B : 挿入位置に近い部分を編集

```
LIST @DISPINIT
```

変数名	用途
X(),Y()	動き回る物体の座標
VX(),VY()	動き回る物体の移動量
NO	番号管理用 (当たり判定で利用)
SC	スコア管理用 (当たり判定で利用)
ED	終了判定用 (当たり判定利用)

【 ヒント 】

0~最大値-1までの乱数生成 . . . RND(最大値)

リスト3-2A : 配列変数の確保、初期化の呼び出し

```
CLEAR
MAX=10
DIM X(MAX), Y(MAX)
DIM VX(MAX), VY(MAX)
'-----
@RESET
GOSUB @DISPINIT
GOSUB @ENEINIT
:
```

リスト3-2B : 初期化処理

```
'-----
@ENEINIT
FOR I=0 TO MAX-1
X(I)=RND(14)*16+16
Y(I)=RND(10)*16+16
GOSUB @GETV:VX(I)=V
GOSUB @GETV:VY(I)=V
SPSET I, 48+I, 0, 0, 0, 0
SPOFS I, X(I), Y(I)
NEXT I
NO=0:SC=0:ED=0
RETURN
'-----
@GETV
V=RND(3)*0.5+0.5
IF RND(2)==0 THEN V=-V
RETURN
'-----
@DISPINIT
:
```

3-3 移動処理の追加

* 移動処理の呼び出しを挿入

命令3-3A : 挿入位置に近い部分を編集

```
LIST @LOOP
```

リスト3-3A : 移動の呼び出し

```
@LOOP  
LOCATE 0,0  
PRINT "COLOR:";COL;" "  
OLDBT=BT:BT=BUTTON()  
TX=TCHX:TY=TCHY  
GOSUB @SETLINE  
GOSUB @ENEMOVE  
:
```

* 移動処理を挿入

命令3-3B : 挿入位置に近い部分を編集

```
LIST @DISPINIT
```

リスト3-3B : 移動処理

```
'-----  
@ENEMOVE  
FOR I=0 TO MAX-1  
IF X(I)=-1 THEN @ESKIP  
PX=X(I)+VX(I)  
PY=Y(I)+VY(I)  
GOSUB @CHKAREA  
X(I)=PX:Y(I)=PY  
SPOFS I,PX,PY  
SPSCALE I,200  
@ESKIP  
NEXT  
RETURN  
'---  
@CHKAREA  
V=VX(I):A=256-32  
IF PX<0 THEN PX=0:V=-V  
IF PX>A THEN PX=A:V=-V  
VX(I)=V  
V=VY(I):A=192-32  
IF PY<0 THEN Y=0:V=-V  
IF PY>A THEN Y=A:V=-V  
VY(I)=V  
RETURN  
'-----  
@DISPINIT  
:
```

3-4 当たり判定を付ける

* あたり処理の呼び出しを挿入

命令3-4A : 挿入位置に近い部分を編集

```
LIST @ESKIP ↵
```



リスト3-4A : 当たり判定の呼び出し

```
      :  
      SPOFS I, PX, PY ↵  
      SPSCALE I, 200 ↵  
      GOSUB @NEAR ↵  
      @ESKIP ↵  
      :
```

* 当たり判定処理を挿入

命令3-4B : 挿入位置に近い部分を編集

```
LIST @DISPINIT ↵
```



リスト3-4B : 当たり判定処理

```
' ----- ↵  
@NEAR ↵  
IF TCHST==0 THEN RETURN ↵  
AX=TX-X(I) ↵  
AY=TY-Y(I) ↵  
IF AX<0 THEN RETURN ↵  
IF AX>31 THEN RETURN ↵  
IF AY<0 THEN RETURN ↵  
IF AY>31 THEN RETURN ↵  
X(I)=-1 ↵  
SPOFS I, TX, -32, 8 ↵  
NO=NO+1 ↵  
SC=SC+NO*10:BEEP 37 ↵  
RETURN ↵  
' ----- ↵  
@DISPINIT ↵  
      :
```

3-5 終了処理

* 終了する処理を挿入

命令3-5A : 挿入位置に近い部分を編集

```
LIST @SETLINE
```



リスト3-5A : 命令置き換えと終了処理

```
      :  
GOSUB @ENEMOVE  
VSYNC 1  
ON ED GOTO @LOOP, @GOVER  
' ---  
@GCLEAR  
LOCATE 0, 20  
PRINT "CONGRATULATIONS!"  
BEEP 42  
END  
' ---  
@GOVER  
LOCATE 0, 20  
PRINT "GAMEOVER"  
BEEP 55  
END  
' -----  
@SETLINE  
      :
```

* 終了条件を挿入

命令3-5B : 挿入位置に近い部分を編集

```
LIST @DISPINIT
```



リスト3-5B : 終了条件判定の挿入

```
      :  
SC=SC+NO*10:BEEP 37  
IF NO==MAX THEN ED=2  
RETURN  
' -----  
@DISPINIT  
      :
```

おまけ 2

このページは少し余裕がある人向けのおまけです。
他の人の作業が終わるまで時間がある人は
プログラムを自分なりに改造してみましょう。

* 時間がある人は、他のルールにも挑戦

- * スコアを上画面に表示する
- * 順番に押さないとGAMEOVERになる
- * ハイスコアの追加
- * 制限時間によるボーナスポイント
- * 物体の動きの変化（加速？近づいてくる？）
- * 敵が画面を塗りつぶし始める



おしまい

support@smileboom.com

何か気になることがあれば、
件名に「京都パラレル」と書いてメールにてご質問ください。

ありがとうございました！