ハピネスチャージ
プチコン

プチコン3ごうにゅうもん
ぜんぺん

誠司：ジャ～ン！　これが何だか分かるかな？

ひめ、めぐみ：それって、なに？なに？

誠司：これは「プチコン3号」といってスマイルブームから発売のニンテンドー3DS向けのダウンロード専用アプリなんだ。

ひめ：見たことない3DS用ゲームだ。それで、これってどんなゲームなの？

誠司：これはただのゲームではなくゲームを自分で作ることができるソフトなんだ。

めぐみ：「メイドイン俺」や「RPGツクール」みたいなソフト・・・？

誠司：それがちょっと違うんだな。プチコン3号は「SmileBASIC」というプログラミング言語を使って自分で自由なゲームを作れるんだ。もちろん、ゲームだけではなくどんなソフトだってできてしまう。

めぐみ：それはすごい！でも、難しそう・・・。

誠司：大丈夫。確かに最初は難しいかもしれないけど慣れたら簡単にできるようになるさ。

```
LOCATE 20,15,-256
PRINT "プチコン3ごう"
```

誠司：これをEDITモードで入力してDIRECTモードでRUNとしてみよう。(STARTやSELECTボタンでも実行できる)

ひめ、めぐみ：すご～い！「プチコン3ごう」って文字が飛び出して見える！

誠司：ＬＯＣＡＴＥの最初の２０が横の座標（X座標）、次の１５が縦の座標（Y座標）、最後の－２５６が奥行きの座標（Z座標）を示しているんだ。この値を変えることで表示する位置が変わったり飛び出す量が変わったりするんだ。

誠司：ちなみにプチコンの文字表示の座標はX座標が0～49、Y座標が0～29となっている。奥行きの座標は0が画面と同じ場所、マイナスだと画面より飛び出してプラスだと引っ込んで見える（指定できる範囲は-256～+1024）のでそれを変えるだけで自由に3DSの立体視表示が可能になるんだ。

めぐみ：たったこれだけで立体視ができるなんてすごい！

誠司：プチコン3号では左上の座標が(0,0)で縦方向は下に行くほど値が大きくなるということに注意が必要だな。数学でよく使われているグラフとは縦方向のプラスマイナスが逆になっている。

ひめ：でも、文字だけではつまんないよね。

誠司：大丈夫さ。プチコン3号ではキャラ用に使えるスプライト、背景用に使えるBGのデータが多数用意されているのでそれを使えば簡単に表示できるんだ。

※写真に記載のものはごく一部です

ひめ：すごくいろんなのがあるよ。

めぐみ：これを使えばいろいろなゲームができちゃいそう。

誠司：あと線や四角形や円を描く命令もあるのでそれを組み合わせて自由な絵を描くことも可能だ。

めぐみ：でも、いろいろあると使うだけでも大変そう・・・。
誠司：そんなことないさ。例えば塗りつぶされた四角形を描くにはこれだけでいいんだ。

```
GFILL 150,50,300,140,RGB(255,0,200)
```

めぐみ：たったこれだけでピンクの四角形ができるんだね。
誠司：ちなみにこのＧＦＩＬＬという塗りつぶし四角形の命令では最初のX座標150、Y座標80とX座標300、Y座標140を結んでできる四角形が描かれているんだ。文字とは違ってX座標は0～399、Y座標は0～239の範囲で指定できるんだ。（※下画面は小さいのでX座標は0～311の範囲になる）
誠司：色指定にはＲＧＢという関数を使用しているけどこれはパソコンとかでおなじみのR（赤）、G（緑）、B（青）をそれぞれ0～255の範囲で指定できるんだ。
ひめ：それはパソコンのお絵かきソフトでよく見かけるから知ってる！
めぐみ：この四角形は飛び出して見えないの？
誠司：この線や四角形などを表示するグラフィック面は面ごとにＧＲＰＲＩＯという命令で奥行きを指定できるんだ。飛び出して見えるようにするにはＧＲＰＲＩＯ －２５６を追加してみたらいい。塗りつぶし四角形以外にも塗りつぶさない四角形ＧＢＯＸ命令、線を描くＧＬＩＮＥ命令、点を描くＧＰＥＳＴ命令、円を描くＧＣＩＲＣＬＥ命令なんかがあるのでそれを組み合わせればいろいろな絵が描くことができるんだ。

ここでようやく　登場人物の紹介

誠司
前作のプチコンmkIIも使い込んでいる
文武両道の少年

NOW PRINTING

めぐみ
人助けが好きだけど少しお馬鹿な少女
プログラミング未経験

ひめ
おしゃれが大好きな学業優秀少女
プログラミング未経験

ゆうこ
ごはんがすべての食欲旺盛少女
プログラミング経験は不明

ひめ：さっきのスプライトの絵を使うにはどうしたらいいの？
誠司：これもプチコン3号ならとても簡単さ。例えば4番目（定義番号4）にあるリンゴを表示するにはまずはＳＰＳＥＴ ０，３とすればいい。スプライトの定義番号は1からではなく0から始まっているので4番目は3になっている。

```
SPSET 0,3
```

誠司：ちなみに最初の0は管理番号0番のスプライトを使用しているという意味なんだ。プチコン3号では0～511の最大512個のスプライトを自由に表示することができるので管理番号は0に限らず範囲内の好きな数でもいいけど小さい数字から順に使った方が未使用の番号が何かが分かりやすいぞ。
めぐみ：たったこれだけでリンゴのキャラが表示できるなんてすごすぎるよ。
ひめ：このリンゴのキャラを自分の好きな場所に表示するにはどうしたらいい？
誠司：それにはＳＰＯＦＳという命令を使うんだ。管理番号、X座標、Y座標、Z座標を記せばいい。

```
SPOFS 0,200,160,-200
```

誠司：これで、好きな位置や奥行きにスプライトキャラを表示できる方法が分かったよな？
ひめ：でも、このリンゴって小さいよね。もっと大きくできないの？
誠司：これもＳＰＳＣＡＬＥという命令を使えば簡単さ。

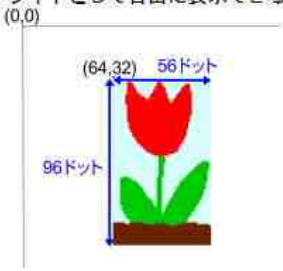
```
SPSCALE 0,5,5
```

ひめ：大きくなっちゃった！
誠司：たったこれだけで縦横5倍の大きさになる。ちなみに縦横の倍率は自由に設定可能だから縦長や横長にすることも可能なんだ。
めぐみ：スプライトのキャラで歩いているようなスプライトがあるけどこれって1つ1つ順番に表示していくいい方法はないのかな？
誠司：それならばＳＰＡＮＩＭという命令を使うといい。

```
SPSET 0,716
SPANIM 0,"I",6,716,6,717,6,718,6,719,0
```

誠司：これでメイドさんが歩いているアニメが簡単にできる。
めぐみ：すご～い！
誠司：簡単に仕組みを説明すると最初の"Ｉ"がスプライトの定義番号を変えるという意味なんだ。要するに違う定義番号のスプライトを表示してアニメーションさせているように見せているわけだ。他にも"ＸＹ"で自動的に上下左右に動かしたり、"Ｚ"で奥行き方向に動かしたり"Ｒ"で回転させたりなど様々な変化をさせることができるんだ。
めぐみ：へぇ～・・・。
誠司："Ｉ"の後の６，７１６というのは6フレームの時間、定義番号716のスプライトのキャラを表示するという意味なんだ。ここで「フレーム」というのは1/60秒のことで6フレームは6/60秒だから0.1秒だな。それと同様に6フレームの間717のキャラを表示、6フレームの間718のキャラを表示、6フレームの間719のキャラを表示という感じで4つのキャラを順番に表示していくことで動いているように見せているんだ。最後の数字はその一連のアニメの実行回数でこの部分を0にすると無限ループになりずっと繰り返してくれる。
めぐみ：つまり、1回だけ動いているように表示するならば最後の数字は1にすればいいわけだよね。
誠司：その通り。
ひめ：ねぇねぇ誠司。自分で描いた絵をスプライトとして表示することはできないの？
誠司：それも簡単さ。まずは、標準で入っているツール「SMILE TOOL」をスマイルボタンで起動してみよう。

誠司：そこから下画面の右上にある「PAINT」を選べばを選べば「GRAPHIC EDITOR」が起動するので下画面の下側中央にある「GO」を選んでグラフィック画面に絵を描いてみよう。描いたものはYボタンを押してメニューを開いてちゃんとセーブするのを忘れずにな。
誠司：例えばＨＩＭＥというファイル名で保存していた場合にはＬＯＡＤ "ＧＲＰ０：ＨＩＭＥ" で描いた絵を読み出すことが可能になる。（※GRP0～GRP5の間で読み出すページは自由にできる）
ひめ：ロードしたら私が描いた絵が表示されたよ。
誠司：描いた絵をGRP3にロードした場合にそれをスプライトで使用するにはSPPAGE 3としてやってそのページをスプライトで使用するように設定する必要がある。ちなみにプチコン3号には0～5の6つのグラフィックページが用意されていて標準設定では0が上画面用GRP、1が下画面用GRP、4がスプライト用、5がBG画面用に設定されているんだ。
誠司：これをスプライト用のページを設定したら次にＳＰＤＥＦという命令を使うんだ。これはＳＰＤＥＦ 定義番号,左上のX座標,Y座標,横幅のドット数,縦幅のドット数を順に書いていくといい。描いた絵の左上が(64,32)で横幅が56ドット、縦幅が96ドットとしたらＳＰＤＥＦ １，６４，３２，５６，９６とすればいい。これで管理番号1のスプライトがその描いた絵になるのであとはSPSETで定義すればそれをスプライトとして自由に表示できるようになるぞ。




```
SPDEF 1,64,32,56,96
```

ひめ：できた！
誠司：スプライトは1ページ分しか使えないので標準で入っているスプライトも併用したい場合にはそのスプライト用に用意しているページの一部を書き換えて使うといい。

誠司：これで、画面に表示するやり方はだいたい分かったと思うけどこれを使ってプログラムを作る場合にまず基本になるのは「変数」なんだ。
ひめ、めぐみ：へんすう！？
誠司：たとえば、ＨＩＭＥ＝１４というものがあるとしよう。これはどういう意味かというとHIMEという変数に14という値を代入しているんだ。**「HIMEと14が等しい」という意味ではない**ことに注意だ。
めぐみ：何だかややこしいので「＝」なんて記号を使わなければいいのに・・・。
誠司：まぁそう言うなって。慣れれば間違うことなんて無くなる。代入した直後はＨＩＭＥという変数には14という値が入っているのでＰＲＩＮＴ ＨＩＭＥやってみて確かめてみよう。変数にはプチコン3号で使われている命令名や関数名と被るもの以外は自由に英数字を組み合わせて付けることができるぞ。（※ただし、1文字目を数字にすることはできない）
めぐみ：つまり、ＭＥＧＵＭＩという変数もＳＥＩＪＩという変数もOKということよね。
誠司：その通り。自分が作るプログラムによって分かりやすい変数名を付けていくといい。
誠司：では、ＨＩＭＥ＝ＨＩＭＥ＋１０としたらどうなるだろう。
めぐみ：ＨＩＭＥにＨＩＭＥ＋１０という値を代入するわけだから・・・え～と・・・。
ひめ：今はＨＩＭＥには14という値が入っているから14+10という値がＨＩＭＥに入るのかな？
誠司：その通り！それを使えばこんな「10年後の年齢を表示するプログラム」も簡単にできる。

```
INPUT "あなたはなんさいですか？" ; AGE
AGE=AGE+10
PRINT "１０ねんごは、あなたは" ; AGE ; "さいです"
```

誠司：ＩＮＰＵＴというのはキーボードから文字を入力する命令なのでこれを使えば簡単なプログラムは作れるようになる・・・と言いたいけど条件判断も覚えておかないとダメだな。
めぐみ：条件判断？
誠司：条件判断にはＩＦ命令を使おう。年齢がマイナスになることはないのでＩＦ命令を使ってマイナスの値を入力したらプログラムを終了するようにしてみた。

```
INPUT "あなたはなんさいですか？" ; AGE
IF AGE<0 THEN END
AGE=AGE+10
PRINT "１０ねんごは、あなたは" ; AGE ; "さいです"
```

誠司：ＩＦ命令による条件判断はIFとTHENの間にある条件を満たしていたらＴＨＥＮ以下を実行し、それ以外の時はＴＨＥＮ以下は実行しない（ＥＬＳＥがある時はＥＬＳＥ以下を実行する）というシンプルなものだ。この場合はＡＧＥ＜０という条件つまりＡＧＥという変数の値が0より小さいときにＴＨＥＮ以下であるＥＮＤを実行するという意味になる。
誠司：ちなみにＡＧＥが0と等しいかどうかを判断したい場合には「ＩＦ ＡＧＥ＝０ ＴＨＥＮ ～」ではなく「ＩＦ ＡＧＥ＝＝０ ＴＨＥＮ ～」になることに注意が必要だ。どんなプログラムでも入力部分と表示部分と計算処理部分と条件判断部分を組み合わせることで作ることができるんだ。
めぐみ：そうなんだ・・・。
誠司：あとプチコン3号ではＲＵＮで実行した直後は変数の値は0になっているけどダイレクトモードで計算した場合にはそうならないので強制的にすべての変数の値を0にしたい場合はＣＬＥＡＲ命令を実行しよう。

誠司：ゲームを作る場合にINPUTでキャラを動かすというのは無理があるのでプチコンでできる入力方法をいくつか説明していくことにするぞ。十字ボタンやⒶⒷⒶⒷ ボタンなどを使う方法、スライドパッドを使う方法、タッチパネルを使う方法、モーションセンサーやジャイロセンサーなどを使う方法がある。
誠司：まずは一番応用範囲が広いボタンを使うやり方を説明しよう。どんなボタンが押されたかを調べるにはＢＵＴＴＯＮという関数を使えばいい。これは次のようになっている。

各ボタンを押した時のBUTTON関数の値

	上	下	左	右	A	B	X	Y	L	R
BUTTON関数の値	1	2	4	8	16	32	64	128	256	512

めぐみ：何だかバラバラだよね。1、2、3、4と順番に分かるようにしてくれたら簡単なのに・・・。
誠司：実はこれには理由があるんだ。ボタンは複数のボタンを同時に押すことが可能なのでＢＵＴＴＯＮ関数ではこのように2倍ずつ値が増えていくのは都合がいいんだ。例えば、1、2、3、4、・・・と順番に増えていく場合にはAボタンとBボタンを同時に押したかどうかを調べることは不可能だけどこのように2倍ずつ増えていくようにしておけばAボタンの16とBボタンの32を足してＢＵＴＴＯＮ関数の値は48になる。ボタンを複数押した場合に合計で「48」になるという組み合わせは16と32以外は無いので48という値からAボタンとBボタンが押されていることが分かるんだ。
めぐみ：なるほど！ 2倍ずつ増えていくのは都合がいいんだ。
誠司：実はこれは2進数で考えるととても簡単なんだ。Ⓐボタンの16は**0000010000**、Ⓑボタンの32は**0000100000**になる。
めぐみ：2進数って・・・余計分からないよ・・・。

誠司：これは先ほどのボタンと数字の対応表を「上」からではなく「R」から始まるようにしたものだけどそれで見てみれば簡単に分かるぞ。

	R	L	Y	X	B	A	右	左	下	上	※押した所のボタンが1になっている 2進数		10進数
Ⓐボタン を押した時	0	0	0	0	0	1	0	0	0	0	0000010000	→	16
Ⓑボタン を押した時	0	0	0	0	1	0	0	0	0	0	0000100000	→	32
ⒶⒷボタン を押した時	0	0	0	0	1	1	0	0	0	0	0000110000	→	48

※BUTTON関数の値

ひめ：これならば0000010000(=16)がⒶボタン、0000100000(=32)がⒷボタンというのも一目瞭然だよね。

誠司：では、Ⓐボタンが押されたら猫の鳴き声（BEEP 69）が鳴って終了するようにするにはどうしたらいいか分かるかな？

ひめ：簡単だよ。

```
B=BUTTON()
IF B==16THEN BEEP 69:END
```

ひめ：これを実行すると・・・あれ！？何も起きない・・・。

誠司：それはこのプログラムは実行してすぐに終了してしまうためなんだ。すぐに終了しないためにはGOTO命令を使って繰り返す必要がある。実は先ほど言い忘れていたけどこの繰り返し処理もプログラムを作る場合には必要になってくるんだ。

```
@LOOP
B=BUTTON()
IF B==16THEN BEEP 69:END
VSYNC
GOTO @LOOP
```

※プログラムを停止させるにはSTARTボタンかSELECTボタンを押す。

誠司：@で始まるのがラベルでGOTOではそのラベルのある行に一気にジャンプすることができるんだ。あとおまじないでVSYNCという命令を入れておいた。これは表示のタイミングを取る命令だけどボタン入力のタイミングにも必要になるのでこのプログラムでもちゃんとボタン判定が行えるように入れておくのがいいだろう。

ひめ：ではRUNをしてⒶボタンを押すと・・・やった成功だ！

誠司：確かにこのプログラムの場合はこれで問題はないけど十字ボタンを押しながらⒶボタンを使うようなゲームだとこれだとダメなんだ。

めぐみ：確かにこれだと他のボタンを押しているときはいくらⒶボタンを押しても反応しないよ。

ひめ：ウソ・・・。

誠司：そのためにあるのがビット演算子「AND」なんだ。PRINT 48 AND 16をダイレクトモードで実行してみよう。

ひめ：16と表示されたよ。

誠司：このANDを使えば一度にいくつボタンを押してもそれが押されたか分かるんだ。先ほどのプログラムのB==16を(B AND 16)==16に変えてみよう。

```
@LOOP
B=BUTTON()
IF (B AND 16)==16 THEN BEEP 69:END
VSYNC
GOTO @LOOP
```

めぐみ：すごい。他のボタンを押してもちゃんと判定が可能になったよ！
誠司：この**16**を48に変えれば**A**ボタンと**B**ボタンを同時に押しているときという条件にすることも可能だ。先ほどのように2進数で考えればANDを使えば各桁を比較して両方が1の時に1になるというのも分かるだろう。

```
    1100110000   例えばLRABボタンを押している場合        512+256+32+16=816（BUTTON関数の値）
AND 0000110000   ABボタンが両方押されているかを判断したい   32+16=48     B AND 48は48になる
    0000110000   48 → ABボタンと同じ値となっているので両方押されていることが分かる
```

※2進数で表記された各桁を比較して両方1ならば1になりそうでない場合は0になっている
誠司：少し応用的な考えになるけどＩＦ命令では条件式の値が**「0」**か**「0以外」**かでTHEN以下が実行するかどうかが決まるんだ。**条件式が成立しているときは式の値は1（不成立の時は0）になる**ためＩＦ（Ｂ ＡＮＤ １６）==１６ＴＨＥＮ ～というのは**A**ボタンを押している時は式の値が**1**だからＴＨＥＮ以下を実行している。Ｂ ＡＮＤ １６というビット演算は**A**ボタンを押しているときは**16**の値になり**「0以外」**であるため実はＩＦ Ｂ ＡＮＤ １６ ＴＨＥＮ ～と短く省略することもできる。
めぐみ：へぇ～そうなんだ。だったら**AB**ボタンを両方押しているかどうかを判定するにはＩＦ Ｂ ＡＮＤ ４８ ＴＨＥＮ ～と短くできるわけだよね。
誠司：めぐみ、それはできないんだ。
めぐみ：えっ！どうして！？
誠司：**A**ボタンのみを押している時はＢ ＡＮＤ ４８は**16**になるし**B**ボタンのみを押している時はＢ ＡＮＤ ４８は**32**になるのでどちらの場合も**「0以外」**だから**AB**ボタンを同時に押してない時でもＴＨＥＮ以下を実行してしまうため**なんだ。だから省略できるのは単独のボタンを押しているかを判定する場合のみなんだ。
めぐみ：そっか・・・。難しいなぁ・・・。
誠司：難しければ、省略をしなければいいだけだな。
めぐみ：理解できるまではそうするよ。
誠司：あと繰り返し処理はラベルとＧＯＴＯを使ったけどプチコン3号では様々なものが用意されていてＷＨＩＬＥ ～ ＷＥＮＤ、ＲＥＰＥＡＴ ～ ＵＮＴＩＬなどを使うこともできる。「ＷＨＩＬＥ 条件式～ＷＥＮＤ」で**条件式を満たしている間はＷＨＩＬＥとＷＥＮＤの間を実行でき**るけどこれも満たしているかはＩＦ命令と同じく**「0」**か**「0以外」**かで判断されているのでずっとループ（無限ループ）にしたい場合には ＷＨＩＬＥ １ ～ ＷＥＮＤ でOKなんだ。ちなみに「ＲＥＰＥＡＴ ～ ＵＮＴＩＬ 条件式」は**条件式を満たすまで実行する**のと**ループを抜ける時に判定が行われるため最低1回はＲＥＰＥＡＴ～ＵＮＴＩＬ内を実行するするというのがＷＨＩＬＥ～ＷＥＮＤとの違い**なので自分の用途に応じて使い分けるといいだろう。（※WHILE～WENDは入口チェックであるため最初から条件を満たしてない時はWHILE～WEND内は実行されない）

誠司：ゲームでよく使われる十字ボタンで4方向や8方向に移動させる方法だけど十字ボタンで8方向にキャラを動かす場合もビット演算を使えば簡単にできるんだ。

《十字ボタンでキャラの8方向移動　サンプルプログラム》

```
ACLS
SPX=200:SPY=120
SPSET 0,3
WHILE 1
  B=BUTTON()
```

```
  X=0:Y=0
  IF B AND 1 THEN Y=-1
  IF B AND 2 THEN X=-1
  IF B AND 4 THEN X= 1
  IF B AND 8 THEN Y= 1
  SPX=SPX+X
  SPY=SPY+Y
  SPOFS 0,SPX,SPY
  VSYNC
WEND
```

誠司：ACLSとすることで表示を初期化できるんだ。文字（コンソール）のみならばCLS、スプライトのみならばSPCLRで初期化ができる。先ほども言ったようにWHILE 1は常に条件を満たしている状態（1は0以外の値）なので無限ループになっている。あと**ループ内は字下げ（インデント）をするとリストが見やすくなるのでおすすめ**だ。
誠司：SPX=200：SPY=120とかは**初期設定**といってプログラムを開始するときに最初にどのようにしたいのかをメイン部分を実行する前に設定しておく必要があるんだ。このプログラムの場合はRUNをしたら常に画面中央付近にキャラがあるけどゲームの場合はステージやキャラの配置などはこの初期設定で行う必要がある。
ひめ：ねぇねぇ誠司～8方向の移動なのにIF命令による判定は4つでいいの？
誠司：ひめ、いいところに気づいたな。ビット演算を使っているのでこれで判定が可能なんだ。例えば斜め左上に行く場合には左と上を同時に押す必要があるけどこれはB AND 1が0以上の場合（上ボタンが押されている場合）とB AND 4が0以上の場合（左ボタンが押されている場合）で考える必要があるけど斜め左上のBUTTON関数の値は5だからIF B AND 1 THEN ～とIF B AND 4 THEN ～ という2つのIF命令を両方とも満たしているんだ。
ひめ：なるほど～。
めぐみ：何となく分かった。

誠司：次はスライドパッドだ。これにはSTICK OUTという命令を使用する。スライドパッドを使ってリンゴのスプライトキャラを動かすプログラムは次のようになる。

《スライドパッドでキャラ移動　サンプルプログラム》

```
ACLS
SPX=200:SPY=120
SPSET 0,3
WHILE 1
  STICK OUT X,Y
  SPX=SPX+X
  SPY=SPY-Y
  SPOFS 0,SPX,SPY
  VSYNC
WEND
```

誠司：STICK OUT X，YでX、Yには押した量に応じて-1～1の値が入る（実際は最大でも1になることはなく0.866程度）けど円形をイメージすれば分かるけどX=0.866、Y=0.866になることはなくXの絶対値が大きくなったらYが小さくなるため**X*X+Y*Yの値は0.75程度で固定**になっているので十字ボタンと併用するゲームだと斜め方向の移動量が小さくなる点に注意が必要だな。

誠司：スライドパッドの場合は**上向きに押したらY座標方向がプラス**になるので**プチコンの座標とはプラスマイナスが逆**になっていることに注意が必要だ。だからY座標方向に関してはＳＰＹ＝ＳＰＹ－Ｙのように**ＳＴＩＣＫ　ＯＵＴ**で得られる値をマイナスしなくてはならないんだ。
めぐみ：ちゃんとプラスマイナスを合わせてくれたら良かったのに・・・。

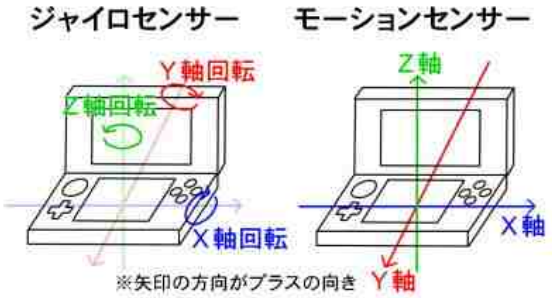
誠司：タッチパネルは**ＴＯＵＣＨ　ＯＵＴ**という命令を使用する。これは**ＴＯＵＣＨ　ＯＵＴ　ＴＭ，ＴＸ，ＴＹ**とすることで変数**ＴＭ**にはタッチしている時間（タッチしてない時は0）、**ＴＸ**にはタッチパネルのX座標、**ＴＹ**にはY座標が入る。ここで注意が必要なのは**プチコン3号の仕様上の問題でタッチパネルの周囲5ドットは正しい座標が取得できない**のでここにタッチが必要なものを置かないということとタッチの**X座標、Y座標はタッチしていないときも前回タッチした場所が取得されてしまう**ためタッチ時間と併用しないと正しく判定ができないということだな。これさえ注意しておけばタッチパネルを使ってキャラを動かすのは簡単にできる。

《タッチパネルでキャラ移動　サンプルプログラム》

```
ACLS
SPSET 0,3
WHILE 1
  TOUCH OUT TM,TX,TY
  IF TM THEN SPSHOW 0:SPOFS 0,TX,TY ELSE SPHIDE 0
  VSYNC
GOTO @LOOP
```

誠司：単にタッチした座標に表示するだけならば簡単なのでこのサンプルではタッチしている時は表示してタッチしてない時には非表示にしてみた。表示には**ＳＰＳＨＯＷ**、非表示には**ＳＰＨＩＤＥ**を使うといい。下画面をタッチしていて音がうるさいと思ったら**ＳＹＳＢＥＥＰ＝０**をプログラムに追加すればいい。**ＳＹＳＢＥＥＰ＝１**とすれば元にもどる。あと**ＸＳＣＲＥＥＮ**を使って上下画面をどのように使うかが設定できるけど今は必要ないので省略する。

誠司：次はジャイロセンサーとモーションセンサーの使い方について説明しよう。
めぐみ：そもそも、ジャイロセンサーとかモーションセンサーって何なの？
誠司：ジャイロセンサーは本体を傾けた角度が分かり、モーションセンサーは本体を動かしたときの加速度が分かるんだ。
めぐみ：う～ん・・・そう言われてもさっぱりわからないよ。
誠司：詳しくは次の図を見てみれば分かるぞ。



めぐみ：なるほど～。確かに違うね。
誠司：注意しなければならないのは**ＸＯＮ　ＭＯＴＩＯＮ**と書いておいてセンサーを使える状態にする必要があるんだ。あとジャイロセンサーは誤差が蓄積されやすいので**ＧＹＲＯＳＹＮＣ**を入れておくのがオススメだ。**ＧＹＲＯＡ　ＯＵＴ**で3方向の傾けた角度が分かるんだ。

《ジャイロセンサー使用　サンプルプログラム》

```
ACLS
XON MOTION
GYROSYNC
SPSET 0,3
WHILE 1
  GYROA OUT R,P,Y
  SPX=200+P*400/PI()
  SPY=120-R*240/PI()
  SPOFS 0,SPX,SPY
WEND
```

めぐみ：本体を傾けたらそれに応じてリンゴが動いてる！
誠司：注意しなくてはならないのは動かした角度はラジアンという単位になっているという点だ。
めぐみ：らじあん？
誠司：直角は90°だけどラジアンではπ/2（約1.57）になる。つまり、3.14だと約180°、つまり、半周分だけ回転させたことになる。この基準点はＧＹＲＯＳＹＮＣを実行したときの本体の向きだからその時点で本体が傾いていたらその傾いた状態から180°回転させたら3.14くらいの値になるんだ。
ひめ：つまり、ＧＹＲＯＳＹＮＣを実行するときは本体を水平にした方がいいってことかなぁ？
誠司：なかなかいい理解をしているな。
誠司：ちなみにこのサンプルでは本体を90°傾けたら画面の端くらいにリンゴが動くようなプログラムになっているんだ。（※中央を基準にしているため左右は200ドットだけどそれをπ/2で割ったらX座標は中心点から動く量が求められる）

誠司：次はモーションセンサーを使ってみよう。これはジャイロセンサーと同様にＸＯＮ ＭＯＴＩＯＮとして使える状態にしてＡＣＣＥＬ ＯＵＴで本体を動かした場合の3方向の加速度が分かるんだ。
めぐみ：さっきも思ったけど加速度ってどういうものなの？
誠司：速度がどんどん速く（遅く）なっていく時の値のことなんだ。例えば物を落下したときどんどん速くなっていくのは地球上では重力加速度があるためなんだ。
めぐみ：そうなんだ。

《モーションセンサー使用　サンプルプログラム》

```
ACLS
XON MOTION
SPSET 0,0
WHILE 1
  ACCEL OUT X,Y,Z
  SPX=200+X*200
  SPY=120-Y*120
  SPOFS 0,SPX,SPY
WEND
```

誠司：このモーションセンサーを使う場合に**注意しなくてはならないのは重力加速度まで検出されてしまうという点**だな。本体を地面に平行にして持った場合にはサンプルのZの値は-1くらいの値になる。つまり、**重力加速度と同じ1Gの加速度で絶対値が1くらいの値になる**わけなんだ。
ひめ：それだと無重力ならば0になっちゃう？
誠司：その通り。本体を自由落下させれば遊園地のフリーフォールと同じで無重力が体感できるので値はほぼ0になるぞ。
めぐみ：そんなことをしたら本体が壊れてしまうよ。

誠司：そうだな・・・ハハハ。（フリーフォールを体感するネタゲームを作った・・・なんてとても言い出せない状況だ）

《　プチコン3号用体感ゲーム「フリーフォール」　》

```
ACLS
XON MOTION
S=0
REPEAT
UNTIL FREE()<0.05
BEEP 6
REPEAT
  S=S+1
UNTIL FREE()>1
BEEP 71
PRINT "SCORE=";S
DEF FREE()
  VSYNC
  ACCEL OUT X,Y,Z
  RETURN ABS(X)+ABS(Y)+ABS(Z)
END
```

※自作関数DEFの解説については
後編で書く予定です

遊び方

RUNをして本体を開いたまま真上に投げるか
真下に落としてください。（自由落下させる
うまくキャッチできれば点数が入ります。
キャッチできなければ本体が壊れます。
（※注意　本体が壊れても責任取れません）

誠司：さて、そろそろゲームを作ってみようか。
ひめ：もう疲れたよ～・・・。
ゆうこ：それならば少し休憩ね。特製ハニーキャンディーを食べれば頭すっきりよ。
ひめ、**めぐみ**：やったー！！
ゆうこ：おおもりごはん特製の弁当もあるのでぜひ食べてね。
誠司：さて、食べながらでもいいので聞いて欲しいけどゲームだけどめぐみとひめはどんなものを作りたいのかな？
めぐみ：私はストリートファイターのような格闘アクションゲーム
ひめ：私はときメモのような恋愛シミュレーションゲーム
誠司：具体的なゲーム名を挙げて作りたいゲームを考えるのはいいけどさすがに今の二人には難しいかもしれないな。でも、**難しければより単純なものに変えてもいいだけだし、別の作りたいものを作ればいい。難しくて訳が分からず途中で作るのをやめるよりも簡単なものでも完成させる方が絶対にいいぞ。**
誠司：どんなゲームであっても今まで説明したように初期設定をして計算処理、条件判断、繰り返しを行うだけで作ることができる。ただし、**経験値を増やす命令とかダメージを受けたら体力が減る命令なんてないからそれを具体化してどの変数にどんな値を計算して入れていくかとかどうなったらいいのかという条件式を考えなくてはいけない。**これは実際にゲームを作りながら勉強していこうと思ったけどどうやら時間が無くなったみたいだ。
誠司：というわけで、続きはまた今度やるのでちゃんと復習しておくように！
ひめ、**めぐみ**：ムシャムチャ（食べることに集中して全く聞いてない）
誠司：・・・・・・

誠司：***それではみんな、後編で会おうぜ！***（・・・って俺は誰に向かって言ってるんだ！？）

ハピネスチャージ プチコン
プチコン3号入門　前編

2014年12月30日発行
制作・発行　おちゃめくらぶ
発行人　　御茶目菜子（おちゃめ）

おちゃめくらぶ　Webサイト　http://ochameclub.web.fc2.com/
おちゃめ　twitter　https://twitter.com/ochame_nako
おちゃめ　pixiv　http://pixiv.me/ochame_nako

おちゃめくらぶ 2014