【A-0】IchigoJam でシューティングゲームを作る

●今回の目標

IchigoJam で動くシューティングゲ ームを作ります。 自機を左右に動かして、ビームを 発射して、上空の UFO を打ち落 とします。

0		
	۵	
	I.	
	0	

●ゲームの画面を表示する

まず、プログラムの初期設定をした後、画面にスコア(点数)を表示します。

10	/*SHOOTING*	コメントで、プログラムのタイトルを入れる
20	CLS:CLV	画面をクリア、変数をクリア
30	LC 0,0	カーソルを画面左上へ移動
40	?S	スコアを表示

入力できたら、「RUN」で実行してみましょう。 画面がクリアされて、左上に「0」が表示され ます。



プログラムの内容を説明します。

10行:コメントで、プログラムのタイトルを入れています。

先頭に「'」(アポストロフィ、キーボードでは Shift キーを押しながら「7」を押す)を付けると、その行はコメントとなり、何も実行されません。ですから、「'」のあとは好きな文字を書くことができます。

プログラムを後で見た時にわかりやすくするために、プログラムにいろいろコメントを 入れるといいでしょう。 20行:CLS命令で、画面をクリアします。

その後の CLV(シーエルブイ)命令は、変数(へんすう)をクリアする命令です。 変数とは、数字を入れる入れ物(箱)と思ってください。 小学校の算数でやる「□」(四角)、中学校の数学でやる「x」と考えればいいです。 今回は、ゲームの最初に、全ての変数をクリアしています。

この行では、2 つの命令を「:」(コロン)で続けて書いています。「:」を使うと、複数の命令を続けて書くことができます。

30 行:LC(LOCATE、ローケート)命令は、画面に文字を表示する位置(カーソル位置)を設 定する命令です。

LC 0 ,0

x座標 y座標

x 座標	画面の x 座標(0~31)。
y 座標	画面の y 座標(0~23)。

IchigoJamの画面サイズは、横32文字×縦24 行になっています。 横(x 方向)の座標は 0~31、縦(y 方向)の座

標は0~23になっています。

今回は「LC 0,0」として、画面左上の座標を指 定しています。



40行:「?S」で、画面にスコア変数Sの値を表示しています。

(?は PRINT 命令の略です)

最初に CLV 命令で変数をクリアしていて、S の値も0 なので、「0」と表示されます。 「?"S"」と S をダブルクォーテーションで囲むと、文字「S」を表示しますが、「?S」だと、 「変数 S の値を表示する」という意味になります。

LC 命令の座標の数字を変えると、スコア「0」の表示位置が変わります。 いろいろ変えて試してみましょう。

●自機を表示する

スコアと同じように、自機を表示してみましょう。 以下のプログラムを追加します。



プログラムを実行してみましょう。 画面中央下に「A」が表示されます。

追加したプログラムの内容を説明します。

50 行:自機の横座標の変数 X、縦座標の 変数 Y に値を入れています。 「X=15」は、「X と 15 が等しい」とい う意味ではなく、「X に 15 を入れる」 (代入する)という意味です。

0	
	АОК

60 行:LC 命令で、カーソルを(X,Y)の座標へ移動します。 このように、座標を変数で指定することもできます。

70 行:?(PRINT)命令で、自機「A」を表示しています。

ここでは「?"A";」と最後に「;」(セミコロン)を付けています。 PRINT 命令で文字を表示した後は、通常は自動的に改行してカーソルが次の行へ 移るのですが、「;」を付けると、改行せずに次の文字位置にカーソルがとどまります。 (※改行するとあとで表示がおかしくなるので、こうしています)

今の所は、「A」を表示した後にプログラムが終了して、続いて「OK」を表示してしまいます。 これから続きのプログラムを作っていくので、今回は気にしないことにします。

50 行の変数 X,Y の値を変えれば、自機の表示位置が変わります。 いろいろ変えて試してみましょう。

●自機を動かす

自機をカーソルキー(矢印キー)で左右に動かせるようにしてみましょう。 以下のプログラムを追加します。



プログラムを実行してみましょう。 カーソルキーの左と右を押すと、自機が左 右に移動します。 このままだとプログラムがいつまでも終わら

ないので、ESC キーを押して止めてください。



追加したプログラムの内容を説明します。

80 行:ここからゲームの処理をするプログラムになるので、コメントを入れています。 ループしてくり返すので、「GAMELOOP」としています。

90~100行:自機を一度消します。

110行:自機を左へ動かす処理をします。

BTN(ボタン)関数を使って、左矢印キーが押されているかどうかを判断します。 BTN 関数の文法は以下のとおりです。

BTN(28)

キー指定

キー指定	28(LEFT)…左矢印キー(←)
	29(RIGHT)…右矢印キー(→)
	30(UP)…上矢印キー(↑)
	31(DOWN)…下矢印キー(↓)
	32(SPACE)…スペースキー
返り値	キーが押されている=1
	キーが押されていない=0

指定したキーが押されているとBTN 関数の値が1、押されていないと0になります。 その BTN 関数の値を、IF(イフ)命令で判断します。

IF BTN(28)=1 THEN X=X-1 ELSE ~ 条件式 条件が成り立 条件が成

つ時に実行

条件が成 り立たない 時に実行

条件式	条件を判断する式。
THEN(ゼン)	条件が成り立つ時に実行するプログ
の後	ラム。
ELSE(エルス)	条件が成り立たない時に実行するプ
の後	ログラム。ELSE 以下は省略可能。

この行は、「もしBTN(28)が1だったら(=左矢印キーが押されていたら)、変数 X を 1減らす」という意味になります。

なお「X=X-1」は、「Xから1を引いて、それをXに入れる」という意味です。「XとX-1が等しい」という意味ではないので注意してください。

120行:110行と同じように、自機を右へ動かす(x座標を1増やす)処理をします。

130~140行:自機を新しい座標に表示します。

150行:続けて自機を動かすため、GOTO(ゴートゥー)命令で前へ戻しています。

GOTO 80 行番号 指定した行番号へ実行を移します。 このプログラムだと、以下の問題があります。

- ▶ 自機が画面左はじへ行っても左矢印キーを押し続けると、自機は動かないが、次に 右へ動かそうとすると、左矢印キーを押していたのと同じ時間だけ右矢印キーを押 し続けないと、右へ動かない。
- ▶ 右はじでも同様。

これは、自機の横座標 X の範囲を考えずに増やしたり減らしたりしてしまい、X がマイナスの 値になってしまったりするからです。 【IchigoJam画面:32文字×24行】





IF 命令の条件式に、「AND」(アンド)でつないで、2つの条件を入れます。

110 IF BTN(28)=1 AND X>0 THEN X=X-1

この条件式は、「BTN(28)が1と等しい、かつ、Xが0より大きい」という意味になります。 両方の条件が成り立った時だけ、「THEN」以下の命令が実行されます。 画面の左端に来ると、Xが0になるので、条件が成り立たなくなり、それ以上左へ行きません。

同じように、画面の右はじでもはみ出さないように、AND 条件式を入れます。

120 IF BTN(29)=1 AND X<31 THEN X=X+1

●UFO を表示する

自機の次は、敵の UFO を表示しましょう。 まず、UFO の最初の設定をするプログラムを追加します。

10	/*SHOOTING*	
20	CLS:CLV	
30	LC 0,0	
40	?S	
50	X=15:Y=22	
60	LC X,Y	
70	?"A";	
72	U = R N D (32) : V = 0	縦座標Vは0にする
74	LC U,V	
76	?CHR\$(241);	UFO を画面に表示
80	/*GAMELOOP	
90	LC X,Y	

70 行と80 行の間に行を追加するので、行番号を72~76 にしています。



プログラムを実行してみましょう。 UFO が画面に表示されます。 プログラムを実行するたびに、UFO の 表示位置が変わります。

次ページから、プログラムの内容を説明します。

72 行: UFO の横座標 U・縦座標 V を指定します。毎回同じ場所に出てくると面白くないの で、Uは乱数(らんすう)で指定します。乱数とは、毎回違うでたらめな数字です。 乱数を取り出すには、RND(ランダム)関数を使います。

> 32) RND(乱数の最大値

乱数の最大値 0~最大値-1の乱数が出てきます。

「RND(32)」と指定すると、0~31の乱数が出てくるので、UFOの横座標Uが画面の 左はじから右はじのどこかになります。 縦座標 V は、一番上(0)にします。

74~76行: UFO を画面に表示します。ここでは UFO の文字を CHR\$(シー・エッチ・アール・ ドル、キャラクター)関数で指定しています。

CHR\$(241) 文字コード

ます。

文字コード | 0~255の数字で文字コードを指定



CHR\$関数の文字コードを変えると、表示されるキャラクターが変わります。試してみましょう。 自機の「A」を CHR\$関数に変えて、自機のキャラクターをいろいろ変えるのも面白いでしょう。

140 ?CHR\$(240); 例

●UFO を動かす

表示した UFO を動かしてみましょう。

自機と UFO を同時に動かさないとゲームにならないので、自機を動かしているループのプログラムから、UFO を動かすプログラムをサブルーチンとして呼び出す形にします。

…(前略)…
80 /*GAMELOOP
90 LC X,Y
100 ?" ";
110 IF BTN(28)=1 AND X>0 THEN X=X-1
120 IF BTN(29)=1 AND X<31 THEN X=X+1
130 LC X,Y
140 ?"A";
145 GSB 160 UFOを動かすサブルーチンを呼び出す
150 GOTO 80
160 「×UFO UFOを動かすサブルーチン
170 LC U, V UFO をいったん消す
180 ?" ";
190 U=U+RND(3)-1 UFO の横座標・縦座標を乱数で変化
200 V=V+RND(3)-1
210 LC U,V
220 ?CHR\$(241);
230 RTN メインプログラムへもどる

プログラムを実行してみましょう。UFO がランダムに動き回ります。 それでは、プログラムの内容を説明します。

145行:「GSB」(GOSUB、ゴーサブ)命令で、UFOを動かすサブルーチンを呼び出します。



メインプログラムからは「GSB」命令でサブルーチンへジャンプし、サブルーチンからは「RTN」(RETURN、リターン)命令でもどります。もどった後は、GSB 命令の続きへ プログラムの処理が移ります。

サブルーチンに分けると、プログラムがすっきりしてわかりやすくなります。また、何度 も同じサブルーチンを呼び出して使うことができます。

160 行~: UFO を動かすサブルーチンです。

170~180行: UFO をいったん消します。



今のプログラムだと、UFO が画面の外へはみ出して消えてしまいます。 自機の時と同じように、はみ出さないようにプログラムを改造します。

190	U=U+RND(3)-1	
192	IF U<0 THEN U=0	Uが0より小さくなったら、0にもどす
194	IF U>31 THEN U=31	U が 31 より大きくなったら、31 にもどす
200	V = V + R N D (3) - 1	
202	IF V<0 THEN V=0	Vが0より小さくなったら、0にもどす
204	IF V>22 THEN V=22	V が 22 より大きくなったら、22 にもどす
210	LC U,V	

プログラムを実行してみましょう。今度は UFO が画面からはみださなくなります。

190 行・200 行の RND 関数の最大値や引き算する値を変えると、UFO の動きが変わります。

●ビームを発射する

自機からビームを発射してみましょう。

まず、ゲームのメインループのプログラムで、「スペースキーが押されていたら、ビームのサブ ルーチンを呼ぶ」ようにします。



次に、プログラムの最後に、ビームのサブルーチンを追加します。



プログラムを実行してみましょう。 スペースキーを押すと、ビームが自 機から UFO の高さまで飛びます。



プログラムの内容を説明します。

240行:ここからビームのサブルーチンが始まります。

250行:ビームの横座標 Bを、自機の横座標Xにします。

260 行:ビームを自機から UFO まで飛ばすため、FOR(フォー)命令と NEXT(ネクスト)命令 を使って、ビームの縦座標 C を変化させてくり返します。

FOR	C=Y-1	ТΟ	V	STEP	-1
	変数の		変数の		変数の
	初期値		終値		増分

変数の初期値	ループ変数の最初の値
変数の終値	ループ変数の最後の値
変数の増分	ループ変数をどのくらい変化させるかの値
	STEP 以下を省略すると1ずつ増やす

ビームの動きを考えると、自機の1 つ上で発射されて、UFO まで上向 きに飛んでいきます。

ですから、ビームの縦座標 C を、 自機の1つ上(Y-1)から、UFO の 縦座標(V)まで、1 つずつ減らして いけばいいことになります。

この FOR 命令から、310 行の NEXT 命令までのプログラムを、C の値を Y-1 から V まで変化させて くりかえします。



270~280 行:ビームを表示します。「」」の文字は、Shift キーを押しながら「¥」を押します。

290~300行:ビームを消します。

310行:くり返しの終わりを示す NEXT 命令です。

320行:RTN 命令で、メインプログラムへもどります。

このままだと、ビームの動きが速すぎてよく見えません。時間待ちの命令を入れて、動きを遅くしてみましょう。

24	0 ′*B6	EAM			
25	$\Theta = B = X$				
26	0 FOR	C = Y = 1	TO V 🔅	STEP	-1
27	0 LC 1	B,C			
28	0 ?" '	";			
28	5 WAI	T 2 時間	寺ちをする		
29	0 LC)	B,C			
30	0 ?" '	", ;			
		-			
31	0 NEX	Г			

プログラムを実行してみましょう。

今度はビームの動きが遅くなるので、UFO の高さまで飛ぶのがよくわかります。

時間待ちをするために、WAIT (ウェイト)命令を使います。

WAIT 2 待ち時間

待ち時間 60分の1秒単位で指定する。「60」で1秒。

WAIT の待ち時間を変えると、ビームの移動速度が変わります。 いろいろ変えて試してみましょう。

●ビームとUFO の当たり判定

今のプログラムだと、ビームが UFO に当たっても何も起きません。

ビームが UFO に当たったら、爆発するようにしましょう。



プログラムを実行してみましょう。 ビームが UFO に当たると、命中音が出て爆発 して、UFO が消えます。

320行:ビームの横座標Bと、UFOの横座標U が違っていれば、ビームが当たってい ないので、RTN命令でメインプログラム へもどります。 「B!=U」は「BとUが等しくない」という意 味の条件式です。



330 行:ビームが当たった時のプログラムが、ここから始まります。

340行:BEEP(ビープ)命令で、命中音を出します。BEEP命令の文法は以下のとおりです。

BEEP	30	,30
	音の高さ	音の長さ

音の高さ	1~255 で指定する。省略可能。
音の長さ	60分の1秒単位で指定する。「60」で1秒。省略可能。

数字を変えると、音の高さや長さが変わります。いろいろ変えて試してみましょう。

350~360行: UFO の座標に、爆発「*」を表示します。

370行:爆発を表示した後に、しばらく時間待ちをします。

380~390行:爆発表示を消します。

400 行: RTN 命令で、メインプログラムへもどります。

このままだと、UFO をうち落としても点数が入らないので、スコアを加算します。



プログラムを実行してみましょう。 UFO をうち落とすと、スコアが増えていくよ うになります。

362 行:スコア変数 Sを 10 点加算します。

364~366行:画面左上にスコアを表示しま す。



362 行の加算する数字を変えると、UFO をうった時の点数が変わります。 いろいろ変えて試してみましょう。 UFO をうち落とした後、同じ場所にまた出てくると、次にうつのが簡単になってしまいます。 UFO が上にもどって、違う場所に表示されるようにしましょう。

330	′*HIT
340	BEEP
350	LC U,V
360	?"*";
362	S=S+10
364	LC 0,0
366	?S
370	WAIT 20
380	LC U,V
390	? " " ; UFO の位置を再設定する
395	U=RND(32): V=0
400	RTN

最初の設定と同じように、UFO の位置を乱数で決めます。

プログラムを実行してみましょう。

UFO をうち落とすと、違う場所に出てくるので、次にうつのが難しくなります。

●UFO の侵略

今のままだと、UFO が一方的にやられるだけなので、ゲームとして面白くありません。 UFO が自機の段まで降りてきたら、ゲームオーバーになるように改造しましょう。

160	1 * UF 0
170	LC U,V
180	?"";
190	U=U+RND(3)-1
192	IF U<0 THEN U=0
194	IF U>31 THEN U=31 $10 \oplus 01 \oplus 000$
200	V = V + R N D (3) - 1 UFO の縦座標 Vを1 増やす
201	IF RND(10)=0 THEN $V=V+1$
202	IF V<0 THEN V=0
204	IF V>22 THEN V=22
210	LC U,V
220	?CHR\$(241);
230	RTN

「RND(10)」で乱数「0,1,2,…9」が出るので、それが0と等しい場合は(=10分の1の確率)、 UFOの縦座標Vを1増やします。これでだんだん下へ降りてくるようになります。

そして UFO が一番下まで降りると、縦座標 V が 22 になるので、その場合はゲームオーバー になるようにします。メインループのプログラムを改造します。



プログラムを実行してみましょう。 UFO が自機の段まで来ると、ゲーム オーバーになります。



150 行:これまで「GOTO 80」と無条件に前へもどっていた行を、IF 命令を使って「V が 22 より小さかったらもどる」ようにします。

152 行: BEEP 命令で、ゲームオーバーの音を出します。

154~156 行: 画面の中央に「GAME OVER」と表示します。

158 行:CLK(シーエルケー)命令でキー入力をクリアして、END(エンド)命令でプログラムを 終了します。

201 行の乱数の範囲を変えると、UFO が下に降りてくる速度が変わります。 いろいろ変えて試してみましょう。

201 IF RND(10)=0 THEN V=V+1