

IchigoJam ゲームの森 プログラム集

http://153.127.198.12/Hana/ichigo_book/

IchigoJam ゲームの森 1

ボードゲーム系

ルーレットマシン
ダイヤを探せ!!
イチゴを見つけよう
数当てゲーム
3つの絵をそろえよう
旗をあげろ
コンピュータとじゃんけん
ボタンでイチゴ(1,5)がり
ダイヤを見つけろ
算数の達人
算数の名人 3つの数のたし算
3匹のネコを探せ
神経すいじゃく
数当てゲーム(改)
モグラたたき(改)
旗あげ(改)
消えたイチゴを探せ(改)
上か下か、右か左か消えた敵を撃て(改)
走る男を探せ
UFOを探せ!!
石取りゲーム
五目ならべ

スクロールゲーム系

滝のぼり
コースを突っ走れ
ブロック拾い
ネコをつかまえて!!
ジャイアントスラローム
ブロックを集めろ
ネコ、まりを拾って!!

シューティングゲーム系

敵を撃ち落とせ
ターゲットに矢は当たるか
矢を放って敵を討て!!
ヘリだけをげきついせよ
消えた怪獣を撃て!!
Wシューティング
怪獣退治
蚊取り線香
宇宙バトル:敵を撃ち落とせ
ターゲットに矢は当たるか(改)
がんばって拾おう
宇宙船を落とせ!!

IchigoJam ゲームの森 2

迷路・探索・アクション系ゲーム

レスキュー、人々を救え!!
猫のフルーツがり
なわとびさっちゃん風ゲーム
すごろくバトル!
お手玉
アップ&ダウン
もぐらたたき
UFOキャッチャー
おにごっこ
ハードルを跳び越えろ!!
くねくね道ダッシュ!!
出口で会おう
出口へゴー!!マリオ風
ネコのイチゴがり
見えない敵をやっつけろ
山を登って下ってダイヤを拾う

ハートをつかめ!!
くらげ退治
がんばって走れ!!
ブロック集め
ゲートを抜けろ!!
散歩しながらイチゴを食べる
暗黒の世界:暗闇バトル
飛び越えながらつかみ取れ!!
同じ数字に積み
ネコ、イチゴを取りに行く
逃げる敵を捕まえろ!!
隠されたダイヤを発掘せよ!!

ボールゲーム系

サッカー:パスを受け取れ
ゴルフ
新版・ゴルフを楽しもう
一人でピンポン
一人でピンポン・改良ネット有り
ボーリング
バウンドボール

要点

初期化: 画面をクリアし、XとYの座標を設定。

RはYの初期値(11)を取り、以降のループで利用。

20-40: ランダム性を使って表示する文字のコードCを決定・表示。

乱数次第でCは0か236~239のいずれかに設定され、該当文字を画面上のランダムな位置に描く。

50-70: Y位相の周囲の3点(Y-1, Y, Y+1)にそれぞれ異なる文字コード(153, 146, 155)を描画。視覚的な「縦ライン」または物体の一部を構成している可能性。

80-90: 入力待ちと現状のクロック取得。キー入力があるかを検知準備。

110: Yの上下移動をキー入力(K=30, K=31)に応じて更新。ただし範囲は1~21の制限付き。

120: SCROLL RIGHTにより画面全体を右へスクロール。視覚的な動作の連続性を作る。

130: ループの先頭に戻り、再び乱数で新しい表示を作成。

```

10 CLS:X=31:Y=11:R=Y
20 IF RND(3)≠0 C=0:GOTO 40
30 A=RND(4):C=A+236
40 LC 0,RND(21)+1:CHR$(C)
50 LC X,Y-1:CHR$(153);
60 LC X,Y:CHR$(146);
70 LC X,Y+1:CHR$(155);
80 WAIT 8
90 K=INKEY()
100 CLK
110 Y=Y-(K=30)*(Y>1)+(K=31)*(Y<21)
120 SCROLL RIGHT
130 GOTO 20
    
```



補足

使用されているCHR\$(数値)は、該当の数値に対応するキャラクターを画面に出力します。

数値がどの文字になるかは、使用環境のコード表に依存します

(例: 153、146、155は特定の図形や線、ブロックなどを表すことが多い)。

行30-40の流れは、Cの値を決定して、画面上の任意の位置にランダムな文字を描く仕組み。

行110の式は、Kの値に応じてYを上下させるブロックですが、Kの値はINKEYの戻り値次第。実機環境によってKの数値は変わる場合があります(例えばキーコードの割り当て次第)。

10 CLS:X=31:Y=11:R=Y

CLS: 画面をクリア（クリアな表示状態へ）。

X を 31、Y を 11 に設定。

R を Y の値（11）に設定。

20 IF RND(3)≠0 C=0:GOTO 40

RND(3) は 0 以上 2 以下の乱数を返す想定。RND(3) ≠ 0 の場合、C を 0 にして 40 行へジャンプ。
つまり乱数が 0 以外なら C=0、40 へ。

30 A=RND(4):C=A+236

RND(4) は 0 ~ 3 の乱数。A にその値を格納。

C は A+236 に設定。したがって C は 236 ~ 239 のいずれか。

40 LC 0,RND(21)+1:CHR\$(C)

LC 0, RND(21)+1 の位置にカーソル移動して、CHR\$(C)（コードに対応する文字）を表示。

RND(21)+1 は 1 ~ 21 の乱数。多分背景のキャラクター描画位置のランダム化。

50 LC X,Y-1:CHR\$(153);

座標 (X, Y-1) にカーソル移動して、CHR\$(153) を表示。末尾にセミicolonは改行を抑制。

60 LC X,Y:CHR\$(146);

座標 (X, Y) にカーソル移動して、CHR\$(146) を表示。末尾は改行なし。

70 LC X,Y+1:CHR\$(155);

座標 (X, Y+1) にカーソル移動して、CHR\$(155) を表示。

80 WAIT 8

8 単位待機。処理を一時的に止めて表示を安定させる。

90 K=INKEY()

ユーザーのキー入力を K に代入。INKEY は非同期キー検出。

100 CLK

CLK は現在の時刻やクロック値を取得する命令。副作用はなし、デバッグ用か待機の合図。

110 Y=Y-(K=30)(Y>1)+(K=31)(Y<21)

Y の更新。

(K=30) は K が 30 の時に真 (1)、そうでなければ 0。

(Y>1) は Y が 1 より大きいときに 1、そうでなければ 0。

同様に (K=31) は K が 31 のとき、(Y<21) は Y が 21 未満のときに 1。

つまり:

矢印キーの制御を想定。K が 30（左キーなど）なら Y を 1 減少させる、Y>1 条件つき。

K が 31（右キーなど）なら Y を 1 増やす、Y<21 条件つき。

結果として Y が 1 ~ 21 の範囲内で上下移動可能。

120 SCROLL RIGHT

画面を右にスクロール（またはキャラクターの横移動の準備）。

130 GOTO 20

20 行へ戻ってループ開始。

初期化後、画面に小さな点/記号を描画して基準位置を設定。

ランダムに 3x3 の方向パターンを決めて、次の座標 (U, V) を算出して描画。

ユーザーがキーを入力するのを待ち、入力が W と一致すれば次のステップへ（正解扱い）。

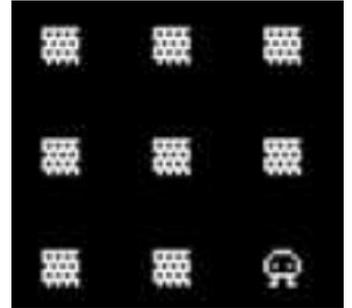
入力が正しくない場合は待機して再挑戦。

このサイクルを繰り返し、ゲーム的なミニ操作を実現している。

```

10 CLS:X=15:Y=11:H=3
20 I=-1:J=-1
30 LC X+I*H,Y+J*H:?CHR$(2)
40 I=I+1:IF I<2 GOTO 30
50 I=-1:J=J+1:IF J<2 GOTO 30
60 W=RND(9)+1
70 A=W%3:B=(W-1)/3
80 IF A=1 R=-H
90 IF A=2 R=0
100 IF A=0 R=H
110 IF B=2 L=H
120 IF B=1 L=0
130 IF B=0 L=-H
140 U=X+R:V=Y+L
150 LC U,V:?CHR$(237):WAIT 3
160 LC U,V:?CHR$(2)
170 K=INKEY():IF K=0 GOTO 170
180 IF K-ASC("0") !=W GOTO 200
190 LC U,V:?CHR$(237)
200 WAIT 60:GOTO 20

```



CHR\$() は特定のキャラクタを画面に描くためのコード

WAIT は一定時間待機

INKEY() はキー入力の取得

ASC は文字コード

RND(n) は0以上n未満の乱数

W は1~9の乱数、

% は剰余演算、**/** は除算、***** は乗算、**+** は加算、**-** は減算を意味します。

画面をクリアして初期設定を行い、乱数で X を決定
 5 回のループで何らかの文字表示と座標操作を行い、表示色を変更
 ユーザーに質問「ダイヤはどこ？」を表示し、Y の入力を受け取る
 Y に応じて A の値を計算し、X と Y が一致するかを判定
 一致すれば「見つけた」メッセージと特定文字を表示して終了、そうでなければ「ミス」を表示して再度 80 へ戻る
 補足と注意

CHR\$(...) による文字列は環境依存で、特定の絵文字や全角文字を表示している可能性があります。

LC の意味は色と位置の設定、CHR\$ の表示文字列によってゲーム性（ダイヤを探すミニゲーム風）を表現していると推測されます。

行番号は古い BASIC のパターンで、実行時は行番号に従って順次解釈されますが、GOTO と IF の組み合わせでループと分岐を実現しています。

```

10 CLS:T=1:A=5:B=10
20 X=RND(5)+1
30 FOR I=1 TO 5
40 LC A,B:?CHR$(1,1,1)
50 LC A+1,B+1:?!
60 A=A+4
70 NEXT
80 LC 0,12:?`ダ`イ`ハト`コ?`
90 INPUT Y
100 A=5:A=A+(Y-1)*4
110 IF X=Y LC 5,15:?T,"カime, ミツタ":LC A,B:?CHR$(231,231,231):END
120 LC 5,15:?T,"カime, ミス"
130 LC A,B:?CHR$(2,2,2)
140 T=T+1:GOTO 80
  
```

変数: T, A, B, X, I, Y などを使用

LC は画面のセルへ色指定と文字描画を行う想定のコマンド。

引数は (色A, 色B) のような意味合いで使われ、続く ?CHR\$(...) で文字を表示する

? は画面へ文字を表示

CHR\$(...) は複数の文字コードを連結して1つの文字列を作る関数。ここでは絵文字や特殊文字、文字セットを表現している

RND(5) は 0 以上 5 未満の乱数を返し、+1 で 1~5 の整数にする

PRINT 相当の出力は ? で行う

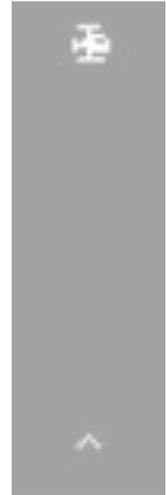
END, GOTO, IF, FOR, NEXT, INPUT, LET/代入などの基本的な構文を想定

画面クリアとカーソル操作を使い、点滅するマーカー（^）を縦方向に動かしながら、条件を満たしたときに「ゲット」を表示する仕掛け。
乱数によって得られるPの値は、文字表示とゲット条件に絡む要素として使われている。ユーザーはスペースキーを押すまで待機し、一定時間経過でリセットされるループ構造になっている。

```

10 CLS
20 X=15:Y=11:A=1
30 LC X,Y:?"^"
40 P=RND(6)+240
50 LC 15,0:?CHR$(P)
60 WAIT 2:LC 15,0:?" ":WAIT 2
70 A=A+1:IF A<=5 GOTO 50
80 CLT
90 K=INKEY()
100 IF TICK(>)>60 GOTO 10
110 IF K!=32 GOTO 90
120 LC X,Y:?" "
130 WAIT 1:Y=Y-1
140 LC X,Y:?"^"
150 IF Y=0 AND P=243 LC 10,22:?"ゲ ット"
160 IF Y>0 GOTO 120
170 WAIT 60:GOTO 10

```



用語

CLS：画面クリア

X, Y：画面上の座標

LC x,y：座標(x,y)へカーソルを移動

?：文字列の出力（画面へ表示）

CHR\$(n)：数値nを対応する文字に変換して返す

RND(n)：0以上n未満の乱数を生成

WAIT t：t秒待機（または一定の時間待つ命令）

A：カウンタ

IF ... GOTO ...：分岐とジャンプ

K=INKEY()：キー入力を取得し、文字コードを返す（何も押されていなければ0等）

TICK()：経過時間（メニューなどでよく使われるタイマー）

IF TICK(>)>60 GOTO 10のような表現：60単位の経過後にループを戻す

P：乱数で得た値を格納

P=243などの比較：特定の条件分岐

「ゲ ット」：表示される日本文字（全角カタカナ風の文字列）

コードの動作を段階ごとに解説

行 10: CLS

画面をクリアして新しい描画を始めます。

行 20: X=15:Y=11:A=1

座標系の初期設定。X=15、Y=11 位置、A=1 というカウンタを用意。

行 30: LC X,Y:?"^"

座標(15,11)へカーソルを移動させ、^ (キャレット) を表示。おそらく初期表示として「^」を描く。

行 40: P=RND(6)+240

0~5の乱数を生成して、それに240を加算。Pは240~245の範囲の値になる。

行 50: LC 15,0:?CHR\$(P)

座標を(15,0)に移動し、Pの値を対応する文字として表示。CHR\$(P)はASCII/文字コードに対応する文字を表示します。Pが240~245なので、特殊文字や日本語には適さない値ですが、環境によってはブロック文字等が表示される可能性。

行 60: WAIT 2:LC 15,0:?" ":WAIT 2

2単位待機。次に(15,0)に移動してスペースを表示(上の文字を消したり、点滅のための空白表示を行う)。再度2単位待機。

行 70: A=A+1:IF A<=5 GOTO 50

Aを1増やす。もしAが5以下なら再度50行へ戻り、同じ乱数を表示するループを5回行います。計5回、Pの文字が表示され続けます。

行 80: CLT

画面をクリアします(おそらくクリア直後の描画リセット)。

行 90: K=INKEY()

キーボード入力を取得します。何も押されていないければ0または空の値が返ることがあります。

行 100: IF TICK()>60 GOTO 10

タイマーが60を超えた場合、行10に戻って最初から画面をクリアしてリセットするループ。つまり経過時間に応じてリセットを強制します。

行 110: IF K!=32 GOTO 90

キーがスペースキー(コード32)以外なら再び90行へ。スペースキーが押されるまで待機します。

行 120: LC X,Y:?" "

現在の座標へ移動して空白を表示。これで現在の「^」の点滅表示を消す操作のよう。

行 130: WAIT 1:Y=Y-1

1単位待機の後、Y座標を1減らします。縦方向に移動させる動作。

行 140: LC X,Y:?"^"

座標(X,Y)に移動して^ (キャレット) を再表示。点滅を続ける演出。

行 150: IF Y=0 AND P=243 LC 10,22:?"ゲット"

Yが0かつPが243であれば、(10,22)に移動して「ゲット」という文字を表示。取得(ゲット)条件を満たしたことを通知。

行 160: IF Y>0 GOTO 120

Yがまだ正の値なら、再び120行へ戻って点滅と移動を続けます。

行 170: WAIT 60:GOTO 10

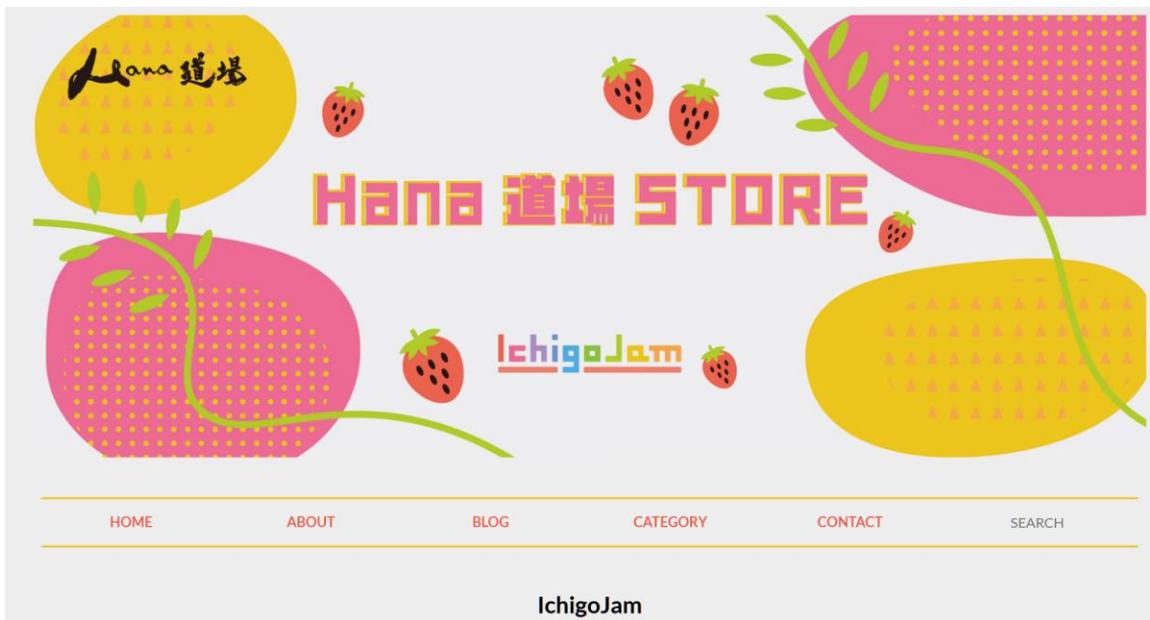
60単位待機してから、行10へ戻り、画面をクリアして最初の状態へリセットします。

行 180 以降はコードが現れていませんが、全体の流れとしては以下のようなミニゲーム風の挙動と読めます。

^の点滅とキャラクターの表示/消去を繰り返し、Y座標を徐々に下げていく。

ある条件(Y=0かつP=243)を満たすと「ゲット」と表示して成功を知らせる。

60単位のタイマーでリセットされ、スペースキーが押されるまで待機するループ構造。



<https://hanadojo.official.ec/categories/748154>

HOT

お得なゲームの森セット【送料
無料】
¥4,900

HOT

¥4,900

※こちらの価格には消費税が含まれています。
※この商品は1点までのご注文とさせていただきます。
※別途送料がかかります。送料を確保する
※¥20,000以上のご注文で国内送料が無料になります。

数量
1

カートに入れる

IchigoJamゲームの森①
¥1,650

IchigoJamゲームの森②
¥1,650

IchigoJamゲームの森③
¥1,650