プチコンでレースゲームを作る（４）

レースゲームに、BGM・効果音などの演出を追加してみましょう。

●BGM を鳴らす

今は静かにゲームが進んでいますが、音楽（BGM）を鳴らしてみましょう。

まず、プログラムのタイトルを「RACE8」に変えましょう。

```
1 '*** RACE8 ***
2
3 '--- サイショ ---
```

BGM を鳴らすには、**BGMPLAY**（ビージーエムプレイ）命令を使います。
ゲームがスタートする時に、BGM を鳴らしてみましょう。

```
37 'テキ
38 SPSET 12,130,0,0,0,1
39 SPOFS 12,0,-20
40 SPANIM 12,2,5
41 ITEKI=0
42
43 GM=0
44 BGMPLAY 14
45
```

14番のBGMを鳴らす

プログラムを実行してみましょう。スタートすると、BGM が流れます。

同じように、ミスした時にゲームオーバーの BGM が鳴るようにしてみましょう。
ミスのプログラムで、BGMPLAY 命令を追加します。

```
76 '--- ミス ---
77 @MISS
78
79 LOCATE 10,11
80 COLOR 13
81 PRINT "XXX MISS XXX"
82 BGMPLAY 6
83 VSYNC 400
84 END
```

6 番の BGM を鳴らす

待ち時間をのばす

BGMPLAY 命令の文法は、以下のとおりです。

```
BGMPLAY      0
             曲番号      曲番号は 0～29。
```

曲は 30 曲あります。一覧リストはこちら→。
曲番号を変えると、違う曲になります。
いろいろ試してみてください。

流れている BGM を止めたい時は、**BGMSTOP**（ビージーエムストップ）命令を実行すると止まります。

番号	説明
0	軽快な曲
1	湿った暗い感じの曲
2	緊張感高まる曲
3	激しくアップテンポな曲
4	スタートジングル
5	クリアジングル
6	ゲームオーバー
7	メニューセレクト
8	結果発表
9	スタッフロール
10	スタッフロール　その 2
11	時代劇ゲーム風
12	軽快なマーチバンド風
13	激しいロック調
14	軽快な曲　その 2
15	WOND
16	考え中
17	WOND2
18	未来系
19	BAL
20	BAL_2
21	スパイ系
22	SCI
23	シューティング
24	パッド
25	SEN
26	ピュア
27	ROA
28	CUR
29	FIG

●効果音を鳴らす

BGM の次は、チェックポイントを通過（ヒット）した時に、効果音が出るようにします。
効果音を出すには、「BEEP」（ビープ）命令を使います。

```
125 '--- チェックポイント ヒット ---
126 @POINTHIT
127
128 BEEP 7
129 SC=SC+100
130 LOCATE 0,0
```

7 番の効果音を出す

BEEP 命令の文法は右のとおりです。

```
BEEP  0     ,0     ,127   ,64
      波形  ピッチ 音量   パン
      番号                ポット
```

波形番号	0～69。詳しくは下の表を見てください。省略時は 0。
ピッチ	音の高さ。0＝標準。-8192＝2 オクターブ下、8192＝2 オクターブ上。
音量	0（無音）～127（最大）。
パンポット	ステレオスピーカーを使って、音を左右にふる。 0＝左、64＝中央、127＝右。

※それぞれの値は省略可能。

波形番号（0～69）は以下のとおりです。いろいろな効果音の他、楽器の音もあります。

0	BEEP	18	シンセブラス	36	AUTO	54	ダンス HH
1	ノイズ	19	シンセベース	37	キラ	55	ヒット
2	カーソル移動	20	ギター	38	ESC	56	ティンバレス
3	決定	21	オルガン	39	バンジョー2	57	チャイニーズシンバル
4	キャンセル	22	ピアノ	40	スクラッチ	58	ミニシンバル
5	上昇	23	カウベル	41	ギター2	59	シェーカー
6	下降	24	タム	42	オルガン 2	60	鈴
7	コイン	25	シンバル	43	ピアノ 2	61	太鼓
8	ジャンプ	26	オープンハイハット	44	PASS	62	シンセ
9	着地	27	クローズハイハット	45	UP2	63	かっこう
10	発射	28	ハンドクラップ	46	録音	64	パフ！
11	ダメージ	29	リムショット	47	シンセタム	65	nohkan
12	金属	30	スネアドラム	48	カウベル 2	66	humandr1
13	爆発	31	バスドラム	49	metro	67	humandr2
14	叫び声	32	OK2	50	tri	68	犬
15	ブレーキ	33	BALL	51	コンガ	69	猫
16	バンジョー	34	和風	52	ダンス BD		
17	シンセストリングス	35	VOLT	53	ダンス SD		

効果音の番号を変えて、いろいろ試してみましょう。

プログラムを「RACE8」の名前で保存（SAVE）しましょう。

●スタートの演出

これまでのプログラムは、RUN 命令で実行するとすぐにレースがスタートしていました。
レースらしく、スタートやカウントダウンの演出をつけてみましょう。

まず、プログラムのタイトルを「RACE9」に変えましょう。

```
1 '*** RACE9 ***
2
3 '--- サイショ ---
```

ゲームループに入る前に「スタート」の処理をするプログラムを追加します。
まず「スタート」の文字を画面の真ん中に表示します。

```
37 'テキ
38 SPSET 12,130,0,0,0,1
39 SPOFS 12,0,-20
40 SPANIM 12,2,5
41 ITEKI=0
42
43 '--- スタート ---
44
45 LOCATE 12,11
46 PRINT "*START!*"
47 BEEP 38
48 VSYNC 60
49 LOCATE 12,11
50 PRINT "       "
51
52 GM=0
53 BGMPLAY 14
54
55 '--- イドウ ---
```

「START」の文字を表示

38 番の効果音を鳴らして、1 秒待つ

「START」の文字を消す

プログラムを実行してみましょう。
「スタート」と表示されてから、レースが始まります。

「スタート」と表示する前に、「3」「2」「1」とカウントダウンするようにしましょう。

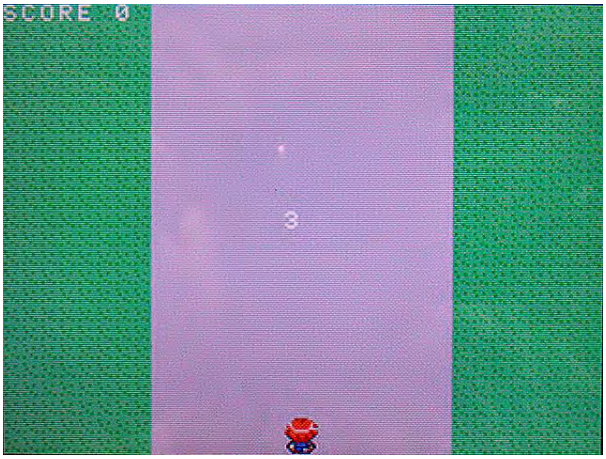
```
43 '--- スタート ---
44
45 FOR T=3 TO 1 STEP -1
46 LOCATE 15,11
47 PRINT T
48 BEEP
49 VSYNC 60
50 NEXT T
51
52 LOCATE 12,11
53 PRINT "*START!*"
54 BEEP 38
55 VSYNC 60
56 LOCATE 12,11
57 PRINT "          "
```

T を 3 から1まで変えてくり返し

画面中央に数字を表示

ビープ音を出して 1 秒待つ

プログラムを実行してみましょう。
カウントダウンしてからスタートするようになります。



プログラムを「RACE9」の名前で保存（SAVE）しましょう。

●コースアウトの演出

これまでのプログラムは、中央のコースを走っても、外側の芝生を走っても、何も変化しませんでした。
コースアウトして芝生を走ると、スピードが遅くなるようにしてみましょう。

まず、プログラムのタイトルを「RACE10」に変えましょう。

```
1 '*** RACE10 ***
2
3 '--- サイショ ---
```

芝生の上を走るとスピードが遅くなるようにするには、まずキャラクターがいる場所の BG（背景）が芝生かどうかを調べます。
ある場所の BG キャラクターを調べるには、BGREAD（ビージーリード）命令を使います。

```
59 GM=0
60 SP=1
61 BGMPLAY 14
62
63 '--- イドウ ---
64 @MOVE
65
66 FOR S=1 TO 2*SP
67
68 'ジュウジ キー
69 B=BUTTON()
70 IF B==4 AND X>0 THEN X=X-2
71 IF B==8 AND X<239 THEN X=X+2
72
73 BGREAD(1,X/8,Y/8),BG,P,H,V
74 IF BG==33 THEN SP=2 ELSE SP=1
75
76 SPOFS 0,X,Y
77 VSYNC 1
```

スピード変数 SP を1にセット

くり返し回数を SP の値で変化させる
SP=1→くり返しが S=1～2 で 2 回
SP=2→くり返しが S=1～4 で 4 回

プレイヤーがいる場所の BG を読み取り

BG が芝生だったら SP を 2 に、
それ以外は SP を1にする

プログラムを実行してみましょう。
プレイヤーがコースアウトして芝生に出ると、スピードが遅くなります。

背景の BG キャラクターを読み取る BGREAD 命令の文法は以下のとおりです。

BGREAD(	1	,X/8	,Y/8)	,BG	,P	,H	,V
	使う BG 面 0＝手前 1＝奥	x 座標 （0～63）	y 座標 （0～63）	BG 番号 読み出し変数	パレット 読み出し変数	横反転 読み出し	縦反転 読み出し

読み出したい BG 面の x 座標、y 座標を指定すると、そこにある BG キャラクターの番号が変数 BG に入ります。何も無ければ番号は「0」になります。
ここではプレイヤーのグラフィック座標 X,Y を 8 でわって、BG の座標を計算しています。

プログラムの 74 行目で、SP に設定する値を変えると、スピードの変化が変わります。
いろいろ数字を変えてためしてみましょう。

プログラムを「**RACE10**」の名前で保存（SAVE）しましょう。

●時間制限、ゴールの演出

これまでのプログラムは、スタートした後はずっと走り続けるだけで、何も変わりません。
時間制限を設けて、30 秒走ったらゴールして次のコースへ行くようにしましょう。

★レース時間（秒数）の表示

まず、プログラムのタイトルを「RACE11」に変えましょう。

```
1 '*** RACE11 ***
2
3 '--- サイショ ---
```

時間を計るには、TIME$（タイムドル）関数を使います。
まず実行画面で、かんたんに試してみましょう。

```
PRINT TIME$
13:28:40
OK
```

「PRINT TIME$」と入力して、ENTER キーを押して実行すると、現在の時刻（時:分:秒）が表示されます。何回も入力して実行すると、時刻がどんどん変わっているのがわかります。
実行画面では、十字キーの上（↑）を押すと、1つ前に打った命令を呼び出せます。

レースがスタートしてからの秒数を計るには、この TIME$関数の秒の部分（最後の 2 文字）を取り出せばよいです。
右側の文字を取り出すには、RIGHT$（ライトドル）関数を使います。
実行画面で以下の命令を打って実行してみましょう。

```
PRINT RIGHT$(TIME$,2)
38
OK
```

RIGHT$関数の文法は以下のとおりです。

```
RIGHT$(      A$          ,2)
          文字変数    右側から取り出す文字数
```

「PRINT RIGHT$(TIME$,2)」で、TIME$の右側から 2 文字が取り出されて表示されるので、現在の秒数が表示されます。

それでは編集画面に移って、レースゲームのプログラムを書きかえてみましょう。
まず、スタート時の時刻（秒）を記録します。

```
59 GM=0
60 SP=1
61 T0=VAL(RIGHT$(TIME$,2))
62 BGMPLAY 14
63
64 '--- ｲﾄﾞｳ ---
65 @MOVE
```

TIME$の右側 2 文字（秒）を取り出し、数字に変換して T0 へ入れる

RIGHT$関数で秒の文字を取り出した後、VAL（バリュー）関数で文字を数字に変換して、変数 T0 に入れています。（数字に変換しないと、後で時間を計るための引き算ができません）

次に、移動のループの中で、スタートしてからの時間を計って表示します。

```
64 '--- ｲﾄﾞｳ ---
65 @MOVE
（中略）
81 IF SPHITSP(0,11) THEN GOSUB @HOLLHIT
82 IF SPHITSP(0,12) THEN GOSUB @TEKIHIT
83
84 T1=VAL(RIGHT$(TIME$,2))
85 T2=T1-T0
86 LOCATE 16,0
87 PRINT "TIME ";T2;" "
88
89 NEXT S
```

現在時刻の秒を T1 に取り出す

T1 から T0 を引いて、経過秒数 T2 を出す

T2 を画面に表示

プログラムを実行してみましょう。
画面の上に秒数が表示されます。

ただ、時刻によってはマイナスの値が表示されてしまいます。

なぜこういうことが起こるか、考えてみましょう。
レーススタートの時刻と、スタートから 10 秒経った時のレース中時刻を例に考えます。

例 1:レースがスタートした時刻が「13:00:00」（13 時 00 分 00 秒）だった場合

スタート時刻　13:00:00	→秒だけ取り出すと	T0=0
↓10 秒後		
レース中時刻　13:00:10	→秒だけ取り出すと	T1=10

この場合は、T2=T1-T0=10-0=10　となるので、表示は「10」となり、特に問題はありません。

例 2:レースがスタートした時刻が「13:00:55」（13 時 00 分 55 秒）だった場合

スタート時刻　13:00:55	→秒だけ取り出すと	T0=55
↓10 秒後		
レース中時刻　13:01:05	→秒だけ取り出すと	T1=5

この場合は、T2=T1-T0=5-55=-50　となるので、表示が「-50」になってしまいます。

つまり、秒の位から分の位への「くり上がり」がある場合を考えないといけません。
「くり上がり」と考え出すと難しいのですが、プログラムで秒数表示だけを考えるなら、「**もし T2 が 0 より小さくなったら、60 を足す**」とすればいいです。
（1 分＝60 秒なので、その分足してもどしてやる）
上の例 2 の場合は、「-50」に 60 を足せば「10」になるので、正しく表示されます。

```
64 '--- ｲﾄﾞｳ ---
65 @MOVE
（中略）
81 IF SPHITSP(0,11) THEN GOSUB
@HOLLHIT
82 IF SPHITSP(0,12) THEN GOSUB
@TEKIHIT
83
84 T1=VAL(RIGHT$(TIME$,2))
85 T2=T1-T0
86 IF T2<0 THEN T2=T2+60
87 LOCATE 16,0
88 PRINT "TIME ";T2;" "
89
90 NEXT S
```

もし T2 が 0 より小さければ、60 を足す

プログラムを実行してみましょう。
今度はスタートからの秒数が正しく表示されます。

★時間制限・ゴールの設定

今はスタートからの秒数が表示されるだけですが、30 秒経ったらゴールするようにします。
プログラムの最初のコースの設定で、制限時間の変数 TLIMIT を 30 秒にします。

```
14 '--- コース ---
（中略）
21 COLOR 0
22 LOCATE 0,0
23 PRINT "SCORE ";SC
24 TLIMIT=30
25
26 '--- アイテム ---
```

制限時間を 30 秒に設定

移動ループで経過時間を計算する所で、もし経過時間 T2 が制限時間 TLIMIT に達したら、
ゲームモードを変更してゴールするようにします。

```
65 '--- イドウ ---
66 @MOVE
（中略）
85 T1=VAL(RIGHT$(TIME$,2))
86 T2=T1-T0
87 IF T2<0 THEN T2=T2+60
88 LOCATE 16,0
89 PRINT "TIME ";T2;" "
90 IF T2>=TLIMIT THEN GM=2
91
92 NEXT S
```

もし T2 が制限時間 TLIMIT を越えれば、
ゲームモード変数 GM を2にする。

ゲームモード変数 GM を前回作ったのですが、新たに「ゴール」のモードを追加します。

GM=0	通常モード
GM=1	ミスをした
GM=2	ゴールした

移動ループの最後でGMの値を見て、もしGM=2ならゴールのプログラムへジャンプします。
ミスのプログラムの後に、ゴールの処理をするプログラムを追加します。

```
100 GOSUB @TEKISET
101 GOSUB @TEKISCROLL
102
103 IF GM==0 THEN @MOVE
104 IF GM==2 THEN @GOAL
105
106 '--- ﾐｽ ---
107 @MISS
108
109 LOCATE 10,11
110 COLOR 13
111 PRINT "XXX MISS XXX"
112 BGMPLAY 6
113 VSYNC 400
114 END
115
116 '--- ｺﾞｰﾙ ---
117 @GOAL
118
119 LOCATE 10,11
120 COLOR 10
121 PRINT "★★★ GOAL ★★★"
122 BGMPLAY 4
123 VSYNC 300
124 END
125
126 '--- ｺｰｽ ｽｸﾛｰﾙ ---
```

もし GM が 2(ゴール)なら
@GOAL へジャンプ

画面中央に
「GOAL」表示

4番の BGM を流して、しばらく待つ

プログラムを終了

プログラムを実行してみましょう。
コースを 30 秒走りぬけると、ゴールします。
24 行の TLIMIT の値を変えると、制限時間が変わります。

レースゲームに限らず、時間制限があるゲームに、今回のプログラムテクニックは使えます。

★次のコースへ行く

このままだと、30 秒走り抜けたらゴールしてゲームが終わってしまうので、ゴールした後は次のコースへ行くようにしてみましょう。
プログラムの最初を、新しいコースで再スタートできるように書きかえます。

```
 1 '*** RACE11 ***
 2
 3 CLEAR
 4
 5 '--- ｻｲｼｮ ---
 6 @RESTART
 7
 8 ACLS
 9 X=120
10 Y=175
11 SPSET 0,76,2,0,0,2
12 SPANIM 0,4,5
13 SPOFS 0,X,Y
14 BGY=0
15
16 '--- ｺｰｽ ---
```

RUN 直後に変数をクリア。
クリアしないと、2 回目に RUN した時に途中のコースから始まってしまう。

再スタートのラベルを追加

「SC=0」の行を消す
（再スタートの時に、スコアがクリアされるとこまるので）

ゴールのプログラムを書き換えて、「END」で終了せずに、最初へもどるようにします。

```
120 '--- ｺﾞｰﾙ ---
121 @GOAL
122
123 LOCATE 10,11
124 COLOR 10
125 PRINT "★★★ GOAL ★★★"
126 BGMPLAY 4
127 VSYNC 300
128 GOTO @RESTART
129
130 '--- ｺｰｽ ｽｸﾛｰﾙ ---
```

@RESTART へジャンプ

プログラムを実行してみましょう。
ゴールすると、また最初からレースがスタートします。

このままだと、再スタートして次のコースになっても、全く同じコースなのでおもしろくありません。次のコースはレベルが上がって難しくなるといいですね。
次のコースへ来ると、コースの幅がせまくなるようにしてみましょう。

コースを設定するプログラムで、コース番号 CN を記録するようにします。
CN の値によって、コースの幅 CW を変化させて、コースを表示します。

```
16 '--- コース ---
17
18 CN=CN+1
19 BGPAGE 0
20 BGFILL 1,0,0,31,23,33,8,0,0
21 COX=16
22 CW=9-CN
23 IF CW<1 THEN CW=1
24 BGFILL 1,COX-CW,0,COX+CW-1,
23,13,0,0,0
25
26 COLOR 0
```

コース番号を1増やす。RUN した直後は CN=0 なので、CN=1 から始まり、2,3,4…と増えていく

コースの幅 CW を計算。コース 1 では 8、その後 7,6,5…と減る

CW が 0 だとコースがなくなるのでもどす

CW で幅を決めて、コースを表示

コースをスクロールさせるサブルーチンでも、CW で幅を決めて表示するようにします。

```
131 '--- コース スクロール ---
（中略）
139 IF COX<8 THEN COX=8
140 IF COX>24 THEN COX=24
141 BGFILL 1,COX-CW,BGY/8,COX+CW-1,BGY
/8,13,0,0,0
```

プログラムを実行してみましょう。
ゴールして次のコースへ行くと、
コースの幅がせまくなって、
コースをたどるのが難しくなります。

プログラムを「RACE11」の名前で保存（SAVE）しましょう。

●できる人は

★新しいルールやアイテム

ここまでで、レースゲームのプログラムは完成です。
できる人は、新しいルールやアイテムなどを追加してみましょう。以下は例です。

- スピードアップアイテム
  - ✧ 魔法、くつなど。これを取ると数秒間走るスピードが上がる。（「イドウ」ループの SP の値を下げればよい）。ただし操作はやりづらくなる。

- コースの氷、水たまり
  - ✧ 氷の上に乗ると、滑って操作が効かなくなる。（十字キーで左右に動かない）
  - ✧ 水たまりにはまると、体がぬれて、その後数秒間速度が落ちる。

- 体力の設定
  - ✧ 走って行くとだんだん体力が減って、0 になるとゲームオーバーになってしまう。
  - ✧ 0 になる前に体力回復アイテム（食べ物・薬など）を取ると、体力がもどる。
  - ✧ 体力が減っていくと走るスピードが落ちる、という演出もおもしろいでしょう。

- 穴（ホール）の設定変更
  - ✧ 今は穴にはまるとすぐにゲームオーバーですが、一定時間（3 秒間など）穴にはまった後に抜け出すようにする。（その分タイムロスになって、スコアが上がらない）

- ジャンプ
  - ✧ A ボタンを押すと、キャラクターが上へジャンプするようにする。（スプライトのy座標を少し上に変えてやる）。目の前に敵やホールがあっても飛び越えられる。
  - ✧ ただしずっとジャンプして空中にいるとゲームにならないので、0.5 秒くらいで着地するようにする。あるいは上に書いた体力の設定を入れて、ジャンプすると大幅に体力が減るようにする。（無制限にジャンプできない）

★全く違うゲームにする

今回はプレイヤーがコースを走って行くレースゲームを作りましたが、背景や設定を変えて、全く違うゲームにすることができます。
要は「画面が上から下へスクロールするゲーム」だったら、今回のレースゲームのプログラムを応用して作ることができます。いろいろアイデアを考えてみてください。

- スキー、障害物競走などのスポーツゲーム
- 背景を宇宙空間、自分を宇宙船にして、スペースバトルゲーム
- 上から落ちてくる音符を受けて、音楽を奏でる（いわゆる音ゲー）

など

スキーゲームに改造した例です。

- 2 本のポールの間をくぐるとボーナスポイント。ただし両側のポールに当たるとゲームオーバー。
- 次のコースへ進むと、ポールの間がせまくなって、くぐるのが難しくなる。
- 敵のキャラクターはそのまま。当たるとゲームオーバー。

プログラムの改造方法は、次のページにあります。

```
 1 '*** SKI ***
 2
 3 '--- ｻｲｼｮ ---
```

タイトルを変更

```
16 '--- ｺｰｽ ---
17
18 CN=CN+1
19 PW=5-CN
20 IF PW<1 THEN PW=1
21 BGPAGE 0
22 BGFILL 1,0,0,31,23,13,8,0,0
23
24 COLOR 11
25 LOCATE 0,0
26 PRINT "SCORE ";SC
27 TLIMIT=30
28
29 '--- ｱｲﾃﾑ ---
30
31 'ﾎﾟｰﾙ
32 SPSET 10,19,1,0,0,1
33 SPSET 11,19,1,0,0,1
34 SPOFS 10,0,-20
35 SPOFS 11,0,-20
36 IPOLL=0
37
38 'ﾃｷ
39 SPSET 12,130,0,0,0,1
40 SPOFS 12,0,-20
41 SPANIM 12,2,5
42 ITEKI=0
43
44 'ﾎﾟｰﾙ ﾂｳｶ
45 FOR P=0 TO 3
46 SPSET 20+P,12,3,0,0,1
47 SPOFS 20+P,0,-20
48 NEXT P
49
```

ポールの幅をコース番号から設定

コース背景を白い雪に

文字色を青に

左右のポールのスプライトを設定

左右のポール間のスプライトを設定

```
71 '--- ｲﾄﾞｳ ---
72 @MOVE
73
74 FOR S=1 TO 3*SP
（中略）
84 SPOFS 0,X,Y
85 VSYNC 1
86
87 'ﾎﾟｰﾙ ﾂｳｶ ﾁｪｯｸ
88 PHIT=0
89 FOR P=0 TO 3
90 IF SPHITSP(0,20+P) THEN PHIT=1
91 NEXT P
92 IF PHIT==1 THEN GOSUB @POINTHIT
93
94 'ﾎﾟｰﾙ､ﾃｷ ﾁｪｯｸ
95 IF SPHITSP(0,10) OR SPHITSP(0,1
1) THEN GOSUB @POLLHIT
96 IF SPHITSP(0,12) THEN GOSUB @TE
KIHIT
（中略）
107 GOSUB @CSCROLL
108
109 GOSUB @POLLSET
110 GOSUB @POLLSCROLL
111 GOSUB @TEKISET
112 GOSUB @TEKISCROLL
113
```

スクロールの速度を遅くする

左右のポール間を通過したかをチェック

ポールと敵に当たったかをチェック

ポールのセットとスクロールのサブルーチンを呼ぶ

```
137 '--- コース スクロール ---
138 @CSCROLL
139
140 BGY=BGY-8
141 IF BGY<0 THEN BGY=BGY+512
142 BGOFS 1,0,BGY
143 BGFILL 1,0,BGY/8,31,BGY/8,13,8,0,0
144
145 SC=SC+1
146 COLOR 11
147 LOCATE 0,0
148 PRINT "SCORE ";SC
149
150 RETURN
151
152 '--- ポール セット ---
153 @POLLSET
154
155 IF IPOLL==1 THEN RETURN
156 IF RND(100)>3 THEN RETURN
157 IPOLL=1
158 IPX=RND(223-PW*16)
159 IPY=0
160 SPOFS 10,IPX,IPY
161 SPOFS 11,IPX+PW*16+16,IPY
162 FOR P=0 TO PW-1
163 PX=IPX+16+P*16
164 SPOFS 20+P,PX,IPY
165 NEXT P
166 RETURN
167
```

コース背景を白い雪にする

文字色を青にする

チェックポイントをセットするプログラムを
ポールをセットするプログラムに改造

左右のポールを表示

左右のポールの間の
通過ポイントを表示

```
168 '--- ﾎﾟｰﾙ ｽｸﾛｰﾙ ---
169 @POLLSCROLL
170
171 IF IPOLL==0 THEN RETURN
172 IPY=IPY+8
173 IF IPY>191 THEN IPOLL=0:IPY=-20
174 SPOFS 10,IPX,IPY
175 SPOFS 11,IPX+PW*16+16,IPY
176 FOR P=0 TO PW-1
177 PX=IPX+16+P*16
178 SPOFS 20+P,PX,IPY
179 NEXT P
180 RETURN
181
182 '--- ﾎﾟｰﾙ ﾂｳｶ ---
183 @POINTHIT
184
185 BEEP 7
186 SC=SC+100
187 LOCATE 0,0
188 PRINT "SCORE ";SC
189 IPOLL=0
190 IPY=-20
191 SPOFS 10,IPX,IPY
192 SPOFS 11,IPX,IPY
193 FOR P=0 TO PW-1
194 SPOFS 20+P,IPX,IPY
195 NEXT P
196 RETURN
197
198 '--- ﾎﾟｰﾙ ﾋｯﾄ ---
199 @POLLHIT
200
201 GM=1
202 RETURN
```

チェックポイントをスクロールするプログラムを、ポールのスクロールに改造

チェックポイントヒットのプログラムを、ポール通過のプログラムに改造

ホール（穴）ヒットのプログラムを、左右のポールにヒットのプログラムに改造

ホールのセット／スクロールのプログラムは、いらないので削除します。