

```

1 /WARPDAN 251231
10 CLS:CLV:D=8:Z=10
20 IF!RND(90)&&![C]GSB100
30 IF[C]GSB200
40 IF!ZGOTO500
50 P=K:GSB300:IFBTN()&&!FBEEP5:F=1:T=K
60 IFP<>KDRAW31,47,W,V,0:W=COS(K)*6/256+31:V=SIN(K)*6/256+47:DRAW3
1,47,W,V
70 LC15,23:ZS:;C=(C+1)%16:IF!(C%2)&&FGSB400
80 GOTO20
100 [C]=1:[C+16]=RND(10)
110 BEEP20:FORA=0TO1:DRAW32,-9,C*4+1,[C+16],!A:WAIT!A*3:NEXT:BEEP5
0:RTN
200 B=[C+16]:M=C*4
210 FORA=0TO1:DRAWM,B+A,M+2,B+2+A,A:DRAWM,B+2+A,M+2,B+A,A:DRAWM+1,
B+2+A,M+1,B+3+A,A:NEXT
220 IFB<>48[C+16]=B+1:RTN
230 [C]=0:Z=Z-1:LCM/2,11:Z:;BEEP40,8:FORA=0TO1:FORB=-1TO1:DRAWM+1,
47,M+1+B*2,25+ABS(B)*3,!A:NEXT:WAIT6:NEXT:LCM/2,11:ZCHR$(0);:RTN
300 K=(ANA(2)-22)/11*2+180:IFK<180K=180ELSEIFK>360K=360
310 RTN
400 DRAWX,Y,0:X=COS(T)*D/256+31:Y=SIN(T)*D/256+47
410 IFPOINT(X,Y)S=S+1:N=X/4:[N]=0:F=0:D=8ELSEGOTO430
420 FORA=0TO1:FORB=0TO330STEP30-A*26:BEEP8/15:DRAWX/4*4+1,[N+16]+1,
COS(B)/50+X,SIN(B)/50+Y,!A:NEXT:WAIT4*!A:NEXT:RTN
430 IFX<0ORX>63ORY<0F=0:D=8ELSED=D+1
440 DRAWX,Y:RTN
500 VIDEO2:BEEP20,50:WAIT50:VIDEO1:CLK
510 IFINKEY()RUNELSECONT

```

■ 変数 ■

A : 汎用 (FOR~NEXT ループなど)
B : 汎用 (FOR~NEXT ループ、代入用)
C : カウンター (ミサイルの号数) [C] : C 号ミサイル発射中フラグ [C+16] : C 号ミサイルの Y 座標
D : 自弾の砲台中心部からの距離
F : 自弾の発射中フラグ
K : 砲台の仰角
M : 代入用 (C 号ミサイルの X 座標)
N : 撃墜したミサイルの号数
P : 砲台仰角値の一時退避用
S : スコア
T : 自弾の進行角度
W、V : 砲身先端の座標
X、Y : プレイヤーの座標
Z : 被弾耐久回数 (初期値 10)

■ 処理の内訳 ■

10 行 : 初期設定
20~80 行 : メインループ
100~110 行 : ミサイル出現処理 (出現航跡描画)
200~230 行 : ミサイル描画、地面着弾判定
300~310 行 : 砲台仰角算出 (アナログコントローラー値読み込み)
400~440 行 : 自弾描画、ミサイル撃破 & ハズレ判定
500~510 行 : ゲームオーバー

★ プログラムは IchigoJam R (BASIC Ver.1.5 系) で記述しました。

IchigoJam S など 1.4 系でも動きはしますが、ゲームとしての動作がかなり遅くなってしまいます。

★ IchigoJam 本体とアナログコントローラーとの接続は VCC-VCC、GND-GND、IN2-OUT、BTN-ショットボタンで繋いでください。

※ 砲台のチラつき、バタつきを抑えたい場合、300 行、310 行を

```
300 K=(ANA(2)/50)-1:IFK<1K=0ELSEIFK>18K=18
```

```
310 K=K*10+180:RTN
```

に変更してみてください。[20260215]