

Atari 800XL - TurboBASIC XL Source

```
'          FLTSIM2D
'
'Longitudinal (2-D: Z & X) flight simulator
'
'for Atari 800XL 2017 8-bit BASIC 10-Liner
'
'Jeff Piepmeier
'March 4, 2017
'http://jeffpiepmeier.blogspot.com/
'http://github.com/jeffpiep/
'
'Parsed with TurboBASIC XL Parser Tool
'http://github.com/dmsc/tbxl-parser
$options +optimize, optimize=-
convert_percent-const_replace,
optimize=+const_folding
'Tested on Altirra
'http://www.virtualdub.org/altirra.html
```

```
DIM K$(1)
DT = .1 : REM (S) TIME STEP

CLS : REM CLEAR THE SCREEN
?

?,"FLIGHT SIMULATOR 2D"
POKE 752,1 : REM TURN OFF CURSOR
```

TBXL Parser Ouput

run on Altirra 800XL with 65C816 @ 7 MHz

```
DIMA$(1)
A=.1

CLS
?

?,"FLIGHT SIMULATOR 2D"
POKE752,1
```

BBC BASIC IV translation
run on BeebEm BBC Model B
with 2nd 65C02 coprocessor

```
A=.1

CLS

@%=&0002010A
P., "FLIGHT SIMULATOR 2D"
VDU 23;8202;0;0;0;
C=0
D=0
F=0
G=0
I=0
J=0
N=0
```

REPEAT	REP.	O=0 P=0 U=0 W=0 X=0 16REP.
BASETIME = TIME	B=TIME	FOR ZZ=1 TO 2 B=TIME
'ATMOSPHERE		
SIGMA=(1-Z*8.0E-05) : REM LINEAR APPROX FOR RELATIVE AIR DENSITY	C=1-D*8e-5	C=1-D*8E-5
STALL = ((U-2*FLAPS)>29) ! (Z<1) : REM DETERMINE IF NOT STALLED	E=(F-2*G>29) ! (D<1)	E=- ((F-2*G>29) OR (D<1))
QSW = 8.1*(U*U+W*W)*1.225*SIGMA : REM DYNAMIC PRESSURE * WING AREA	H=(F*F+I*I)*8.1*1.225*C	H=(F*F+I*I)*8.1*1.225*C
'ANGLE OF ATTACK(S)		
UNOSING = 1/(U + (U=0)) : REM U != 0	1J=1/((F=0)+F)	J=1/(F-(F=0))
SLOPE = W*UNOSING : REM SLOPE OF WIND	K=I*J	K=I*J
SLOPE = -(SLOPE<=-1) + (SLOPE>-1)*SLOPE : REM LIMIT SLOPE TO <+/-1	K=- (K<=-1) + (K>-1)*K	K=(K<=-1) - (K>-1)*K
SLOPE = (SLOPE>=1) + (SLOPE<1)*SLOPE : REM LIMIT SLOPE TO <+/-1	K=(K>=1) + (K<1)*K	K=- (K>=1) - (K<1)*K
ALPHA = SLOPE - .22 * SLOPE^3 : REM WING ANGLE OF ATTACK WRT WIND	L=K-K^3*.22	L=K-K^3*.22
ALPHAT = ALPHA + OMEGA*UNOSING*4.3 + 8.33E- 3*DLTA + .0863*CL - .0873 : REM TAILPLANE AOA. INCLUDES ELEVATOR AND DOWNWASH TERMS	M=N*J*4.3+L+833e-5*O+.0863*P-.0873	M=N*J*4.3+L+833E-5*O+.0863*P-.0873
'LIFT & DRAG COEFFICIENTS		
CLT = .4*ALPHAT -.24*ALPHAT^3 : REM TAILPLANE LIFT COEFFICIENT WITH AREA RATIO. POLYNOMIAL APPROX TO SINE	Q=.4*M-M^3*.24	Q=.4*M-M^3*.24
CL = 0.3+0.16*FLAPS+4.8*ALPHA+12*ALPHA*ABS(ALPHA) -46*ALPHA^3 : REM WING LIFT COEFFICIENT. CUBIC FIT TO GET "DOUBLE HOOKS"	2P=.16*G+.3+4.8*L+12*L*ABS(L) -L^3*46	P=.16*G+.3+4.8*L+12*L*ABS(L) -L^3*46

CL = CL * STALL : REM IF STALLED NO WING LIFT	P=P*E	P=P*E
CLL = CL+CLT : REM TOTAL A/C LIFT COEFFICIENT	R=P+Q	R=P+Q
CD = .025 + .0575*CLL*CLL : REM DRAG COEFFICIENT USING OSWALD EFFICIENCY	S=.0575*R*R+.025	S=.0575*R*R+.025
'ENGINE WITH DROPOFF DUE TO ALTITUDE THRUST=(SIGMA-.05)*THROTTLE*UNOSING*1100 : REM INVERT PROPULSIVE POWER EQUATION	T=(C-.05)*U*J*1100	T=(C-.05)*U*J*1100
THRUST=THRUST*(THRUST<=2000)+2000*(THRUST>2000) : REM LIMIT THRUST TO 2000 N	T=(T<=2e3)*T+(T>2e3)*2e3	T=- (T<=2E3)*T- (T>2E3)*2E3
'TIME STEP EQUATIONS OF MOTION MY = -QSW*(.0308 + CL*(.28-0.1*FLAPS) + CLT*4.3) : REM COMPUTE PITCHING MOMENT, + IS NOSE UP	3V=((.28-.1*G)*P+.0308+Q*4.3)*-H	V=((.28-.1*G)*P+.0308+Q*4.3)*-H
OMEGA = OMEGA + MY*5.48e-5 : REM UPDATE PITCHING RATE WITH PITCHING MOMENT AND IYY	N=V*548e-7+N	N=V*548E-7+N
OMEGA = OMEGA * ((Z>.01) ! (OMEGA>0)) : REM NO ROTATION UNLESS POSITIVE WHEN ON GROUND	N=(D>.01)!(N>0)*N	N=- ((D>.01) OR (N>0))*N
U = U + ((THRUST-QSW*CD)*1E-3 - SINHT*9.81 - OMEGA*W) * DT : REM UPDATE LONGITUDINAL VELOCITY	F=((T-H*S)*1e-3-W*9.81-N*I)*A+F	F=((T-H*S)*1E-3-W*9.81-N*I)*A+F
W = W + (-QSW*CLL*1E-3 + COSTH*9.81 + OMEGA*U) * DT : REM UPDATE TRANSVERSE VELOCITY	4I=(-H*R*1e-3+X*9.81+N*F)*A+I	I=(-H*R*1E-3+X*9.81+N*F)*A+I
W = W * ((Z>.01) ! (W<0)) : REM NO VERTICAL VELOCITY IF ON GROUND, UNLESS ITS UP	I=(D>.01)!(I<0)*I	I=- ((D>.01)OR(I<0))*I
'INERTIAL FRAME UPDATE T=T+DT : REM STEP TIME FORWARD	Y=Y+A	Y=Y+A
THETA = THETA + OMEGA*DT : REM UPDATE PITCH ANGLE	Z=N*A+Z	Z=N*A+Z
COSTH = 1 - 0.49*THETA*THETA : REM QUADRATIC APPROX TO COSINE	X=1-.49*Z*Z	X=1-.49*Z*Z

SINTH = THETA -.15*THETA^3 : REM CUBIC APPROX TO SINE	W=Z-Z^3*.15	W=Z-Z^3*.15
VX = (U*COSTH + W*SINTH) : REM UPDATE VELOCITY OVER GROUND	_ =F*X+I*W	_ =F*X+I*W
VZ = (U*SINTH - W*COSTH) : REM UPDATE VERTICAL RATE	A0=F*W-I*X	A0=F*W-I*X
X = X + VX*DT : REM UPDATE GROUND TRACK POSITION	A1=_*A+A1	A1=_*A+A1
Z = Z + VZ*DT : REMO UPDATE ALTITUDE	5D=A0*A+D	D=A0*A+D
Z = Z * (Z>0) : REM RESTRICT Z TO ABOVE GROUND	D=(D>0)*D	D=- (D>0)*D
 OKTOLAND = OKTOLAND ! (Z>9) : REM BE A LITTLE FORGIVING ON THE TAKEOFF	 A2=(D>9)!A2	 A2=((D>9)ORA2)
'PILOT INPUT. USE KEY INPUTS SAME AS SUBLOGIC FLIGHT SIM II		NEXT ZZ
K\$=INKEY\$	A\$=INKEY\$	A\$=INK.(0)
THROTTLE = THROTTLE + 5*((K\$="\1F")*(THROTTLE<100)- (K\$="\1E")*(THROTTLE>0))	U=(A\$="")*(U<100)-A\$="")*(U>0))*5+U	U=((A\$=".")*(U<100)-(A\$=",")*(U>0))*5+U
DLTA = DLTA + 0.5*((K\$="T")*(DLTA<23)- (K\$="B")*(DLTA>-28)) : REM ELEVATOR UP/DOWN	O=(A\$="T")*(O<23)-A\$="B")*(O>-28))*5+O	O=((A\$="T")*(O<23)-(A\$="B")*(O>-28))*5+O
 FLPIN = FLPIN + ((K\$="N")*(FLPIN<3)- (K\$="Y")*(FLPIN>0)) : REM FLAPS IN/OUT	6A3=A\$="N")*(A3<3)-A\$="Y")*(A3>0)+A3	A3=(A\$="N")*(A3<3)-(A\$="Y")*(A3>0)+A3
FLAPS = 0.05*FLPIN+0.95*FLAPS : REM A LITTLE IIR EXPONENTIAL RESPONSE TO MODEL THE FLAP DRIVE MOTOR	G=.05*A3+.95*G	G=.05*A3+.95*G
 POSITION 2,3	POS.2,3	V.31,0,3
? "KTAS",INT(U*1.94)," " : REM KNOTS	? "KTAS",INT(F*1.94)," "	P."KTAS",F*1.94
? "PITCH",INT(THETA*573)*.1," " : REM DEGREES WITH 1 DECIMAL PLACE	? "PITCH",INT(Z*573)*.1," "	P."PITCH",Z*57.3
? "ALT.",INT(Z*3.28)," " : REM FEET	7? "ALT.",INT(D*3.28)," "	P."ALT.",D*3.28
? "VRATE",INT(VZ*197)," " : REM FEET/MINUTE	? "VRATE",INT(A0*197)," "	P."VRATE",A0*197
 ?	 ?	P.
? "ELEV.", DLTA," " : REM ELEVATOR POSITION	? "ELEV.",O," "	P."ELEV.",O

<pre>? "POWER", THROTTLE, " " : REM % POWER ? "FLAPS ", INT (FLAPS*10+0.5), " " : REM FLAP POSITION ? ? "STALL ", CHR\$ (161-116*STALL); : REM STALL INDICATOR ? CHR\$ (253-221*STALL) : REM BEEP IF STALLING! ? ? "DIST.", INT (X*1.09), " " : REM DISTANCE TRAVELED OVER GROUND ? "TIME", T, , INT (600/ (TIME-BASETIME)); "% " : REM TIME AND SIMULATOR RATE RELATIVE TO REAL TIME UNTIL OKTOLAND & (Z=0) : REM END GAME IF BACK ON GROUND ? IF VZ >-1 : REM CHECK TO SEE IF VERTICAL RATE IS SLOW ENOUGH ? "TOUCHDOWN" ELSE ? "YOU CRASHED!\FD\FD\FD" : REM OH NO! VERTICAL TOO FAST! ENDIF</pre>	<pre>? "POWER", U, " " ? "FLAPS ", INT (G*10+.5), " " ? ? "STALL ", CHR\$ (161-116*E); ? CHR\$ (253-221*E) ? ? "DIST.", INT (A1*1.09), " " ? "TIME", Y, , INT (600/ (TIME-B)); "% " U. (D=0) & A2 ? IF A0 >-1 ? "TOUCHDOWN" EL. ? "YOU CRASHED! ýýý" END.</pre>	<pre>P. "POWER", U P. "FLAPS ", G*10 P. P. "STALL ", NOT-E V. -7* (NOT-E) P. P. "DIST.", A1*1.09 P. "TIME", Y, 2000/ (TIME-B); "% " U. (D=0) AND A2 P. IF A0 >-1 THEN P. "TOUCHDOWN" ELSE P. "YOU CRASHED!"</pre>
---	--	---