00853D

Program Description I

Program Title ORBITOR GAME

Contributor's Name Richard Brown

Address 277 Beacon St

City Boston **State** Mass. **Zip Code** 02116

Program Description, Equations, Variables You are the commander of an orbiting space craft. Your problem is to land at a specified target on the planet rotating below. You have a limited amount of fuel for your mission.

The space craft has one main engine, and two side thrust engines. The fuel for the main engine is in a separate chamber from the side thrust fuel chamber, so you may run out of one before the other.

Burning fuel in the main engine accelarates the craft, that is, the craft velocity increases in the direction the craft is pointing. Buring fuel in the "positive" side thrust engine increases the rate the craft is rotating counterclockwise; burning fuel in the "negative" side thrust engine increases the craft rotation in the clockwise direction. This means that a positive side burn of 30 will cnacel a negative side burn of 30. After executing both, your rate of rotation will unchanged

To summarize, a main burn changes velocity and a side burn changes rotational velocity.

The craft's position is specified by its location on the plane, and its orientation. The third dimension is ignored. If the orientation is 0°, the craft is pointing to the right; if the orientation is 90° it is pointing up. The craft's location can be expressed as one of the following:

Operating Limits and Warnings

1. Absolute cartesian coordinate. Point 0,0 is at the center of the planet. The X-axis is horizontal; the Y-axis is vertical. In the diagram, the craft's absolute rectangular coordinates are 0,8000.

2. Absolute polar coordinates. In the diagram, this coordinate is magnitude 8000, angle 90°.

3. Relative cartesian coordinates. In this coordinate system, the target is at the point 0,0. The X-axis is tangent to the surface at the target point, and the Y-axis is perpendicular to the surface at the target point. In the diagram the craft's relative rectangular coordinates are -5667, 3657.

Similarly, there are four ways to express velocity. Let's assume that the current velocity is -500 along the absolute X-axis (that is, the velocity is 500 to the left).

1. Absolute cartesian velocity. This is the change in absolute X and absolute Y coordinates. In the example, this would be -500,0.

2. Absolute polar velocity. This is the magnitude and angle of the velocity (not the change in magnitude and change in angle). In the example, the polar velocity is magnitude 500, angle 180°.

3. Relative cartesian velocity. The fact that the planet is rotating introduces an apparent velocity in the relative X-axis. The relative cartesian velocity includes this "ground effect." If the rotational velocity of the target is 1°, the relative cartesian velocity of the craft would be -319, -354.

4. Relative polar velocity. This velocity has the same relation to relative cartesian velocity that the absolute polar velocity has to the absolute cartesian velocity. In the example, the relative polar velocity has magnitude 476, and angle 228°.

The unit of time used to express velocities is a "move." That is, suppose the craft's rotation were known to be -10° per move and the current orientation were 100°. Then after this move the new orientation would be 90°. Suppose that nothing very important was expected in the next five moves. Since the calculator takes about seven seconds to make a move, we would like to be able to compress five moves into one.

Suppose that initially a move takes place every hour. If things become excited (for example, you are about to crash), you might like moves to occur every 10 minutes. That is, you want to change the rate at which moves occur so that 6 of the new moves occur for every one of the old moves. The orbitor game program allows such a clock change. For example, to change the move rate (or, equivalently, change the clock rate) so that 6 new moves occur for every old move, one would press [6][f][C] (see step 3 of user instructions). The original clock rate is retained, and the clock rate can be reset at any time.

As mentioned earlier, velocities are expressed in terms of "change per move." Similarly, fuel expendatures (burns) are specified in terms of "change in velocity (or rotational velocity for side burns)." This means that after changing the clock rate the amount of remaining fuel will appear to have changed.

PACKING THE DISPLAY

The orbitor computer usually displays two numbers at a time. These numbers are integers, separated by a decimal point, much as in the moon rocket lander game in the standard pac. However, since the orbitor computer needs to display negative values, the sign of the second integer is indicated by the presence of a 0 digit immediately following the decimal point. Examples:

8000.90	8000 and 90
-5667.3657	-5667 and 3657
12.015	12 and -15
-319.0354	-319 and -354

The two integers packed into the display by the orbitor computer can be read quickly and easily by remembering the simple rule "if a 0 follows the decimal point, the second integer is negative."

COMMANDS

Burn commands:

[A] -- continue. This is equivalent to a 0 main burn or a 0 side burn.

[B] -- main burn. This command specifies the amount of fuel to burn in the main engine. The amount of a burn does not need to be an integer. The craft's velocity will be increased by the amount of the burn in the direction of the current orientation. That is, suppose the craft's orientation is 45°, and a burn of 10 occurs ([1][0][B]). Ignoring the effect of gravity, the velocity will increase by 7.07 along the absolute X-axis and by 7.07 along the absolute Y-axis.

[fB] -- Side burn. This command specifies the amount of fuel to burn in the side engine. The amount of a burn need not be an integer. If the amount is <u>negative</u>, the fuel will be burned in the negative side thrust engine, producing rotation in a clockwise direction (or partially canceling counterclockwise rotation).

These three commands adjust the position, velocity, and orientation of the craft. After doing that, the new orientation and position will be displayed.

Orientation:

rotational velocity . current orientation

Position (one of the following):

abs X . Abs Y
rel X . rel Y
abs magnitude . abs angle
rel magnitude . rel angle

The first number displayed is the orientation. A positive rotational velocity is counterclockwise, negative is clockwise. The second number will be in one of the four foms above. Polar vs. rectangular is controlled by flag 0:

Flag 0 set -- use polar
Flag 0 clear -- use rectangular

Relative vs. absolute is controlled by which was the last requested:

[D] or [f][D] was last request -- use relative
[E] or [f][E] was last request -- use absolute

<u>Crash</u>. If you should crash (the new position of the craft is below the surface of the planet) the absolute polar coordinates are displayed <u>once</u>, and then the calculator enters a loop displaying first the relative velocity and then the relative position. You may use [R/S] to stop the loop.

Clock commands:

[fC] -- Change the clock rate. [6][f][C] specifies that 6 new moves occur for every old move. Similarly [.][5][f][C] specifies that one new move occur for every two old moves.

[fA] -- Reset clock rate to what it was when the game began.

. Information requests: (note: [R/S] may be used to stop any of the following commands at any time)

[f D] -- display relative position. The relative position will be in rectangular coordinates if the polar flag (flag 0) is cleared, and in polar coordinates if the flag is set.

rel X . rel Y if rectangular
rel magnitude . rel angle if polar

[D] -- Display relative velocity and position. The velocity will be in relative cartesian velocity if the polar flag (flag 0) is cleared, and in relative polar velocity form if the flag is set.

rel X velocity . rel Y velocity
rel velocity magnitude . relative velocity angle

[f E] -- display absolute position. Like [f D] except that absolute position is used.

[E] -- display absolute velocity and position. Like [D] except that absolute coordinates are used, and ground effect is ignored (see discussion of absolute velocity and relative velocity above).

[C] -- general information request. Four values are displayed:

1. remaining main fuel . remaining side fuel

2. rotational velocity . orientation

3. target position in polar coordinates
   magnitude . angle

4. The rate at which the planet is rotating. This is the only time a single number is displayed by the game. This number is displayed using [DSP][9].

RENDEZVOUS. Flag 1 controls whether the calculator checks for crashing. This flag can be cleared to create a variation of the orbitor game, rendezvous. In rendezvous the object is to catch up with an orbiting space station. The station is in a circular orbit.

RACE. The calculator can keep track of two crafts at the same time. This allows two people to race (number of moves) to the target. Both players may have identical initial conditions, or initial conditions may be different.

DATA CARD

The initial configuration is on a data card. Only one side of the card is needed for solo play. Racing will require both sides of the data card.

Registers R0 to R9 (and RS0 to RS9 for races) contain information about the current status of the craft and the target. Registers RA to RE contain more global constants.

Reg.	Description of contents	Example
0 (S0)	absolute X coordinate of craft	0
1 (S1)	absolute Y coordinate of craft	8000
2 (S2)	absolute X velocity of craft	-500
3 (S3)	absolute Y velocity of craft	0
4 (S4)	orientation of craft in degrees	135
5 (S5)	rotational velocity of craft in degrees	0
6 (S6)	radius of planet (or space station orbit radius for rendezvous)	2000
7 (S7)	angle of target point in degrees (registers 6 and 7 contain polar coordinate of target point)	45
8 (S8)	remaining main fuel	11428
9 (S9)	remaining side fuel	1140
A	gravitational constant	-8163265305
B	rate planet is rotating (not = 0)	1
C	rate planet is rotation (must be same as above)	1
D	(temporary)	0
E	constant	360

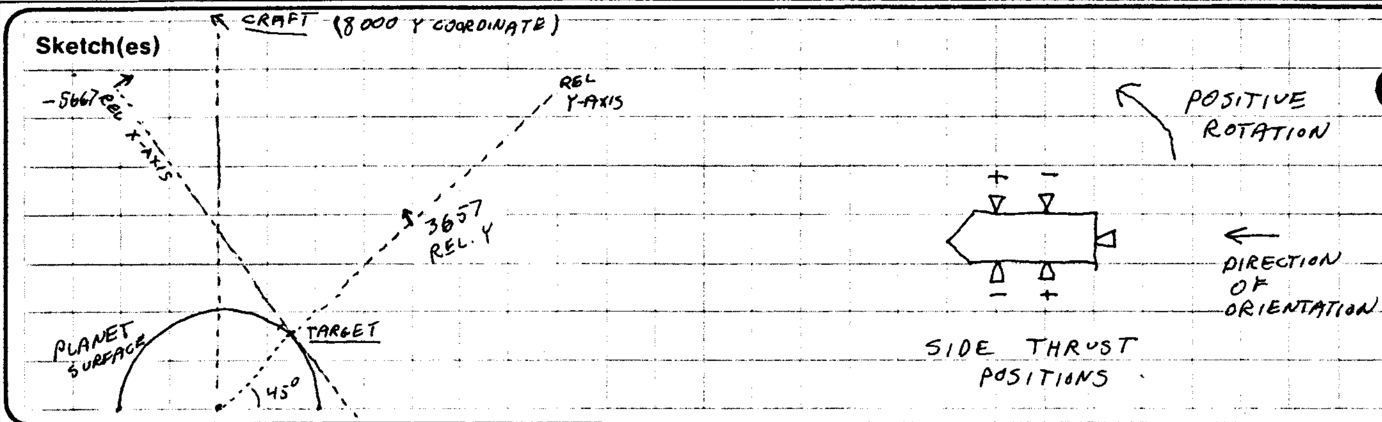
LIMITS AND WARNINGS. Orbits are only approximate due to discrete step size. One effect of this is that the orbit will precess.

There is a singularity at the center of the planet. If the craft should pass too near this singularity, error conditions may result.

The display packing routine may fail if the number of digits necessary to display both numbers is greater than 10. For example, 10250 and -20215 will be incorrectly displayed as 10250.02021.

Program Description II

Sketch(es)



Sample Problem(s) Using data from the example data card description given in the program description I, one might start as shown below:

input	display	explanation
[C]	11428.1140	main and side fuel remaining
	0.135	rotational velocity and orientation
	2000.45	target position in polar coordinates
	1.000000000	rotational rate of planet (one number)
2.5 [CHS] [f] [B]	-3.133	new rot. vel. and orientation (rounded off)
	-500.8000	new position in absolute rectangular coords.
[A]	-3.130	
	-992.7873	
[CF] [0]		we will now be using polar coordinates
[f] [E]	7935.97	absolute polar coordinate of position
[A]	-3.128	
	7758.101	new position in absolute polar coordinates
[A]	-3.125	
	7479.105	note that magnitude is decreasing.

Solution(s) A solution to the problem of landing in the situation above would spoil all the fun. However, noting that the magnitude of the absolute position is decreasing, we might think that the trajectory is either parabolic or hyperbolic (instead of elliptic). In fact, it is parabolic (as can be discovered by pressing [A] until a crash occurs); so we are not in orbit and will crash on the wrong side of the planet. We want to have more velocity in the negative X direction.

The first step towards a solution is to get the craft pointing in the proper direction (68 [f] [B]), and then accelarate (600 [E]).

Reference(s)

00853D

User Instructions

ORBITOR

F0 -- POLAR COORD. FLAG

	A	B	C	D	E	
1 (f)	RESET CLOCK	SIDE BURN	ΔCLOCK	REL. POS	ABS POS	2
	CONTINUE	MAIN BURN	F, θ, T, ΔT	REL VEL. + POS.	ABS VEL. + POS.	

STEP	INSTRUCTIONS	INPUT DATA/UNITS	KEYS		OUTPUT DATA/UNITS
1	LOAD BOTH SIDES OF PROGRAM CARD				
2	LOAD ONE SIDE OF DATA CARD (2 FOR RACE)				
3	OPTIONAL: SELECT CLOCK RATE	# NEW MOVES PER OLD MOVE	f	C	VELOCITY MULTIPLIER
4	OPTIONAL: SET POLAR COORDINATE FLAG		SF	0	
5	OPTIONAL: SET RECTANGULAR COORDINATE FLAG		CF	0	
6	OPTIONAL: SUPRESS CRASH CHECKING		CF	1	
7	OPTIONAL: ALLOW CRASH CHECKING		SF	1	
8	SELECT FUEL BURN *				
	MAIN	AMOUNT		B	ORIENTATION **
	SIDE (INCREASE Δθ) COUNTERCLOCKWISE	AMOUNT	f	B	ORIENTATION
	SIDE (DECREASE Δθ) CLOCKWISE	-AMOUNT	f	B	ORIENTATION
	CONTINUE (NO FUEL BURN)			A	ORIENTATION
9	OPTIONAL: REQUEST ADDITIONAL INFORMATION				
	REMAINING FUEL, ORIENTATION, TARGET POS.,				
	TARGET Δθ			C	INFO. REQUESTED
	RELATIVE POSITION		f	D	"
	RELATIVE VELOCITY + POSITION			D	"
	ABSOLUTE POSITION		f	E	"
	ABSOLUTE VELOCITY + POSITION			E	"
10	IF RACING, SWITCH IN OPPONENT'S DATA			P⇄S	
11	TO MAKE NEXT MOVE, GO TO STEP 3				
12	TO RESTART, GO TO STEP 2				
13	TO RESET THE CLOCK RATE TO ORIGINAL		f	A	
	* IF A CRASH IS DETECTED, A LOOP				
	IS ENTERED GIVING RELATIVE VELOCITY				
	AND POSITION				
	** ORIENTATION GIVEN IN DEGREES.				
	POSITION RETURNED BY STEP 8				
	IS EITHER RELATIVE OR ABSOLUTE,				
	DEPENDING ON LAST POSITION INFO.				
	REQUESTED IN STEP 9.				
	TO CREATE A DATA CARD:				
1	ENTER $R_0 - R_9$ AND $R_A - R_E$				
2	FOR RACE, ENTER $R_{S0} - R_{S9}$				
3	WRITE DATA CARD		f	W/DATA	

STEP	KEY ENTRY	KEY CODE	COMMENTS
001	*LBL 4	31 25 04	
	F? 0	35 71 00	
	GTO 4	22 04	
	GTO 9	22 09	
	*LBL 4	31 25 04	
	R→P	32 72	
	x⇄y	35 52	
	GSB 8	31 22 08	
	x⇄y	35 52	
010	*LBL 9	31 25 09	
	CF 3	35 61 03	
	DSP 0	23 00	
	x⇄y	35 52	
	RND	31 24	
	x<0	31 71	
	SF 3	35 51 03	
	ABS	35 64	
	ENT↑	41	
	x≠0	31 61	
020	LOG	31 53	
	INT	31 83	
	STI	35 33	
	R↓	35 53	
	ISZ	31 34	; NEVER SKIPS
	F? 3	35 71 03	
	ISZ	31 34	; NEVER SKIPS
	RCI	35 34	
	10^x	32 53	
	÷	81	
030	x⇄y	35 52	
	RND	31 24	
	x<0	31 71	
	SF 3	35 51 03	
	ABS	35 64	
	+	61	
	F? 3	35 71 03	
	CHS	42	
	DSP (I)	23 24	
	RTN	35 22	
040	*LBL 8	31 25 08	
	RCL E	34 15	
	÷	81	
	FRAC	32 83	
	RCL E	34 15	
	x	71	
	DSP 0	23 00	
	RND	31 24	
	x>0	31 81	
	RTN	35 22	
050	RCL E	34 15	
	+	61	
	RTN	35 22	
	*LBL B	31 25 12	
	8	08	
	GSB 5	31 22 05	
	RCL 4	34 04	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	x⇄y	35 52	
	R←P	31 72	
	STO + 2	33 61 02	
060	R↓	35 53	
	STO + 3	33 61 03	
	GTO A	22 11	
	*LBL fB	32 25 12	
	CF 3	35 61 03	
	x<0	31 71	
	SF 3	35 51 03	
	9	09	
	GSB 5	31 22 05	
	F? 3	35 71 03	
070	CHS	42	
	STO + 5	33 61 05	
	*LBL A	31 25 11	
	RCL B	34 12	
	STO + 7	33 61 07	
	RCL 3	34 03	
	STO + 1	33 61 01	
	RCL 2	34 02	
	STO + 0	33 61 00	
	RCL 1	34 01	
080	RCL 0	34 00	
	R→P	32 72	
	F? 1	35 71 01	
	GSB 7	31 22 07	
	x^2	32 54	
	1/x	35 62	
	RCL A	34 11	
	x	71	
	R←P	31 72	
	STO + 2	33 61 02	
090	R↓	35 53	
	STO + 3	33 61 03	
	RCL 5	34 05	
	STO + 4	33 61 04	
	GSB 1	31 22 01	
	F? 2	35 71 02	
	GTO fD	22 31 14	
	*LBL fE	32 25 15	
	CF 2	35 61 02	
	RCL 1	34 01	
100	RCL 0	34 00	
	GTO 4	22 04	
	*LBL 5	31 25 05	
	STI	35 33	
	R↓	35 53	
	ABS	35 64	
	RCL (I)	34 24	
	x⇄y	35 52	
	x>y	32 81	
	x⇄y	35 52	
110	STO − (I)	33 51 24	
	RTN	35 22	
	*LBL C	31 25 13	

REGISTERS

0	1	2	3	4	5	6	7	8	9
X	Y	ΔX	ΔY	θ	Δθ	TARGET RADIUS	TARGET θ	MAIN FUEL	SIDE FUEL
S0 "	S1 "	S2 "	S3 "	S4 "	S5 "	S6 "	S7 "	S8 "	S9 "

A	B	C	D	E	I
CONSTANT IN DISTANCE³/TIME²	TARGET Δθ (CURRENT)	TARGET Δθ (ORIGINAL)	(TEMP)	360	(TEMP)

Program Listing II

STEP	KEY ENTRY	KEY CODE	COMMENTS
	RCL 9	34 09	
	RCL 8	34 08	
	GSB 9	31 22 09	
	-x-	31 84	;REMAINING FUEL
	GSB 1	31 22 01	MAIN. SIDE
	RCL 7	34 07	
	GSB 8	31 22 08	
120	RCL 6	34 06	
	GSB 9	31 22 09	
	-x-	31 84	;TARGET r. θ
	RCL B	34 12	
	DSP 9	23 09	
	RTN	35 22	;TARGET Δθ
	*LBL 1	31 25 01	
	RCL 4	34 04	
	GSB 8	31 22 08	
	RCL 5	34 05	
130	GSB 9	31 22 09	
	-x-	31 84	; Δθ . θ
	RTN	35 22	
	*LBL fc	32 25 13	
	1/x	35 62	
	GSB 3	31 22 03	
	ST I	35 33	
	RCL B	34 12	
	x	71	
	STO B	33 12	
140	RC I	35 34	
	x^2	32 54	
	RCL A	34 11	
	x	71	
	STO A	33 11	
	RC I	35 34	
	*LBL 3	31 25 03	
	STO x 3	33 71 03	
	STO x 2	33 71 02	
	STO x 5	33 71 05	
150	STO x 8	33 71 08	
	STO x 9	33 71 09	
	P⇄S	31 42	
	RTN	35 22	
	*LBL 7	31 25 07	
	RCL 6	34 06	
	x>y	32 81	
	GTO 7	22 07	
	R↓	35 53	
	RTN	35 22	
160	*LBL 7	31 25 07	
	R↓	35 53	
	GSB 9	31 22 09	
	-x-	31 84	
	*LBL 2	31 25 02	;CRASH LOOP
	GSB D	31 22 14	
	-x-	31 84	
	GTO 2	22 02	
	*LBL D	31 25 14	

STEP	KEY ENTRY	KEY CODE	COMMENTS
	RCL 3	34 03	
170	RCL 2	34 02	
	R→P	32 72	
	x⇄y	35 52	
	GSB 6	31 22 06	
	—	51	
	x⇄y	35 52	
	R←P	31 72	
	RCL B	34 12	
	D→R	32 73	
	RCL 6	34 06	
180	x	71	
	+	61	
	GSB 4	31 22 04	
	-x-	31 84	; REL. VELOCITY Vx . VY
	*LBL fD	32 25 14	OR
	SF 2	35 51 02	MAGNITUDE. ANGLE
	RCL 7	34 07	
	RCL 6	34 06	
	R←P	31 72	
	CHS	42	
190	RCL 0	34 00	
	+	61	
	GSB 6	31 22 06	
	x⇄y	35 52	
	R←P	31 72	
	RCL 1	34 01	
	R↑	35 54	
	—	51	
	RCL D	34 14	
	x⇄y	35 52	
200	R←P	31 72	
	R↓	35 53	
	+	61	
	R↓	35 53	
	—	51	
	R↑	35 54	
	GTO 7	22 07	
	*LBL 6	31 25 06	
	RCL 7	34 07	
	9	09	
210	0	00	
	—	51	
	STO D	33 14	
	RTN	35 22	
	*LBL E	31 25 15	
	RCL 3	34 03	
	RCL 2	34 02	
	GSB 4	31 22 04	
	-x-	31 84	;ABS. VELOCITY Vx . VY
	GTO fE	22 31 15	
220	*LBL fA	32 25 11	OR
	RCL B	34 12	MAGNITUDE. ANGLE
	RCL C	34 13	
	÷	81	
	GTO f C	22 31 13	

LABELS

A CONTINUE	B MAIN BURN	C F, θ, T, ΔT	D REL V+P	E ABS V+P
a RESET CLOCK	b SIDE BURN	c Δ CLOCK	d REL P	e ABS P
0	1 USED	2 CRASH LOOP	3 USED	4 USED
5 UPDATE REMAINING FUEL	6 USED	7 CRASH CHECK	8 USED	9 DISPLAY PACKING

FLAGS

0	USE POLAR
1	CHECK CRASH
2	USE REL.
3	TEMP

SET STATUS

FLAGS	ON	OFF
0	☐	☒
1	☒	☐
2	☐	☒
3	☐	☒

TRIG	
DEG	☒
GRAD	☐
RAD	☐

DISP	
FIX	☒
SCI	☐
ENG	☐
n	0