Sgt. Kenneth E. Wigger
125 North 1st St.
Wood River IL 62095

Ready on the Firing Line?

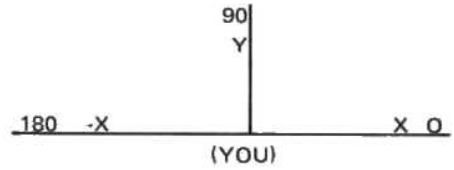
"Artillery Practice" for the SR-56

After reading Herman DeMonstoy's article, "Artillery Practice," in *Kilobaud* (June 77), I wondered if possibly it could be adapted to a small programmable calculator. Lacking 8K BASIC and 10K of memory, I admit that adaptations of computer programs for programmable calculators can be very trying. On several occasions, I have found that I have been able to adapt a good program only to find that the 100-step memory was overrun by 30 or 40 steps. For those who are unaware of a programmable calculator user's definition of frustration, I will relate it to you here.

Frustration is successfully adapting a program only to find yourself with a 101-step program and a 100-step memory. In the true tradition of Murphy's Law, you innocently assume that all you have to do is go back over your program and throw out an unnecessary step. After your first few confident minutes, you slowly realize that every last one of those steps was needed. Still somewhat composed, you accept that you might have to redo the program a little. Forty sheets of paper later, you find that you have a longer program than you started with and cannot find your original program. At this point, your mind doesn't make sense out of things like CMs, EXC, int, bst or subr anyhow. My advice is to give up and try tomorrow or the next day. I am not going to admit how many sheets of paper I used to adapt this program, or why my old analytic geometry book from college is still under my pillow.

The instructions for the game are typed out on a sheet of paper since only a few letters really look like the symbols on my Texas Instruments SR-56.

I would like to give credit to Peter Stark for the ran-



Example 1.

Example 2.

dom-number-generator subroutine ("Submarine!" *Kilobaud*, Feb. 77), which I used in my adaptation of DeMonstoy's "Artillery Practice." Stark's subroutine enabled me to program games that require random numbers without unnecessarily using up my very limited program step memory.

Basic game instructions and an SR-56 user's-instructions form are included to facilitate programming.

Game Instructions

Artillery Practice is a computer simulation game that simulates range-firing of artillery. The simulated range in which you will be firing is shown in Example 1. You will be given only the X and Y coordinates of your target. You must determine the proper azimuth and elevation angle needed to hit it. The maximum range of your artillery piece is 4648.9 feet, and this is only possible if you fire at an elevation angle of 45 degrees. Any elevation angle above or below 45 degrees will reduce proportionally the range of your weapon. After you fire your artillery piece, the distance by which you missed will be displayed. Now that you understand the game, go to the SR-56 user's instructions and start playing. The trajectory is shown in Example 2. ■

Step	Procedure	Enter	Press				Display
1	Prepare to store data in registers.	N/A	CLR	RST	2nd	CMs	0
2	Store initial data to set up game for play.	385.7	STO	0			385.7
		.00001	STO	4			.00001
	(Insert initial random number seed between 0 and 1)	.9	STO	5			.9
		2600	STO	6			2600
	(Example game)						
1	Start game and display initial X of target.	N/A	R/S				-2139.0 (X)
2	Display Y of target.	N/A	X→T				974.4 (Y)
3	Enter elevation angle of shot.	23	X→T				-2139.0
4	Enter azimuth of shot.	160					160
5	Fire and find out distance by which you missed.	N/A	R/S				1017.6
1	Display new X of target.	N/A	R/S				719.5 (X)
2	Display new Y of target.	N/A	X→T				1512.1 (Y)
3	Enter elevation angle of shot.	10	X→T				719.5
4	Enter azimuth of shot.	75					75
5	Fire and find out distance by which you missed	N/A	R/S				308.9

Each new play is continued in this same fashion.

User's instructions.

Loc.	Code	Key
00	49	* fix
01	01	1
02	57	* subr
03	07	7
04	04	4
05	33	STO
06	01	1
07	92	.
08	05	5
09	32	X→T
10	57	* subr
11	07	7
12	04	4
13	34	RCL
14	05	5
15	17	* INV
16	47	X≥T
17	02	2
18	02	2
19	01	1
20	93	±
21	64	X
22	34	RCL
23	07	7
24	94	=
25	33	STO
26	02	2
27	34	RCL
28	01	1
29	32	X→T
30	34	RCL
31	02	2
32	41	R/S
33	33	STO
34	03	3
35	32	X→T
36	64	X
37	02	2
38	94	=
39	23	sin
40	64	X
41	34	RCL
42	00	0
43	43	X^2
44	54	/
45	03	3
46	02	2
47	94	=
48	32	X→T
49	34	RCL
50	03	3
51	26	*F(n)
52	02	P→R
53	52	(
54	34	RCL
55	09	9
56	74	−
57	34	RCL
58	02	2
59	53	)
60	43	X^2
61	84	+
62	52	(
63	34	RCL
64	08	8
65	74	−
66	34	RCL
67	01	1
68	53	)
69	43	X^2
70	94	=
71	48	* $\sqrt{X}$
72	41	R/S
73	42	RST
74	34	RCL
75	05	5
76	64	X
77	07	7
78	45	Y^x
79	09	9
80	64	X
81	34	RCL
82	04	4
83	94	=
84	17	* INV
85	29	int
86	33	STO
87	05	5
88	64	X
89	34	RCL
90	06	6
91	84	+
92	05	5
93	00	0
94	00	0
95	94	=
96	33	STO
97	07	7
98	58	* rtn
99		

Registers

0	385.7
1	Y of target
2	X of target
3	Azimuth
4	.00001
5	Random #
6	2600
7	Random coordinate
8	Y of shot
9	X of shot

Notes

* denotes 2nd

Program listing.