

Submarine!

... a game for

Have you ever wondered how random number generators work or how to write a program for one? Well, Pete Stark discusses the "how to" in the following article which describes a game called "Submarine."

The material applies to both the SR-52 programmable calculator and computers. As a matter of fact, Pete has provided a flow chart so that the program can be written for a computer. ...or another calculator.—John.

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

Location	Comments	Instructions
000	Label A	Enter X value for depth charge
002	STO 05	Store it in register 5
005	Halt	Stop and wait for Y value
006	Label B	Enter Y value for depth charge
008	- RCL 02	
012	$x^2 \sqrt{x} =$	
015	x^2	
016	$+(RCL 05 - RCL 01$	
025	$x^2 \sqrt{x})$	
028	$x^2 =$	
030	\sqrt{x}	Compute distance by which the depth charge missed
031	STO 04	
034	INV Ifzero =	
037	0 1/x Halt	If zero flash display & stop
040	Label =	If not zero . . .
042	- 5 =	. . . compare with 5
045	Ifpos +	If less than 5 subtract 1 from
047	INV dsz C	minor hit counter and go to C if counter has reached 0; otherwise continue
050	Label +	
052	Subr 1 '	Get a small random number . . .
054	SUM 01	. . . and move sub sideways
057	Subr 1 '	Get another small random number . . .
059	SUM 02	. . . and move sub up/down
062	RCL 04	Display miss distance
065	Halt	And stop.
066	Label C	Reset
068	Subr 8 '	Get a big random number . . .
070	STO 01	. . . for sub's X location
073	Subr 8 '	Get another big random number . . .
075	STO 02	. . . for sub's Y location
078	5 STO 00	Reset minor hit counter to 5
082	Halt	And stop
083	Label 1 '	Start of subroutine to get a
085	5 +/-	small number between -5
087	+ .10 X	and +5
092	Label 8 '	Start of subroutine to get a
094	100 X ((7 y ^x 9	big random number between
103	x RCL 03	0 and 100
107	x 5 +/- INV Log)	
113	-(RCL - .5)	Shuffle numbers around to make
120	Fix 0 D.MS INV Fix)	them seem random
126	STO 03=	
130	Fix 0 D.MS INV Fix	
135	Return	End of both subroutines

Program A.

When you're not using your computer or calculator for something useful you can always program it to play games. Many games have been programmed for various computers, including hangman, tic-tac-toe, three-dimensional tic-tac-toe, battleship, and a lunar landing simulation. Some of the games like three-dimensional tic-tac-toe are complicated enough to require a good-sized computer; others are so simple that they can be played with a programmable calculator. Here is a game of "Submarine" for the SR-52 calculator; it is a slight variation on the "Battleship" game included in the Applications Manual for the Texas Instruments SR-56 calculator. In addition to the program, we give here a flowchart so that you can reprogram the game for other calculators or computers as well.

An enemy submarine has been spotted near one of your ports, its exact location unknown. To destroy it, you take a map of the suspected area and place a 100 x 100 grid of graph paper over it as shown above. With this

the SR-52

grid you drop depth charges at specified points, using the X and Y coordinates to keep track of their location. Each time you fire, sonar and other classified equipment pick up an echo from the submarine and tell you how far you missed by. Of course, as soon as you start firing, the submarine starts to zigzag, trying to escape. It can go anywhere in the square as well as up or to the right of it, though it cannot go left or down — into negative X and Y coordinates — since this would bring it too close to shore.

The submarine can be put out of commission only by a direct hit (signalled by flashing lights on the calculator). If you miss by a distance of 5 or less, you only inflict minor damage; the submarine can tolerate up to four minor hits, but at the fifth minor hit it puts on a great burst of speed to get to a new location so repairs can be made.

The SR-52 program is in Program A. To play the game, proceed as follows:

1. Push the C button to start the game.
2. Enter the X coordinate for your depth charge; push A.
3. Enter the Y coordinate for your depth charge; push B.
4. If you hit the sub, the display will flash. Otherwise the display will indicate the distance by which you missed.

5. Go back to step 2 for more shots.

Starting points A, B, and C in the flowchart correspond to the A, B, and C keys on the calculator. The flowchart is fairly straightforward, with the *random number genera-*

tor, represented by the six-sided boxes, being the heart of the game.

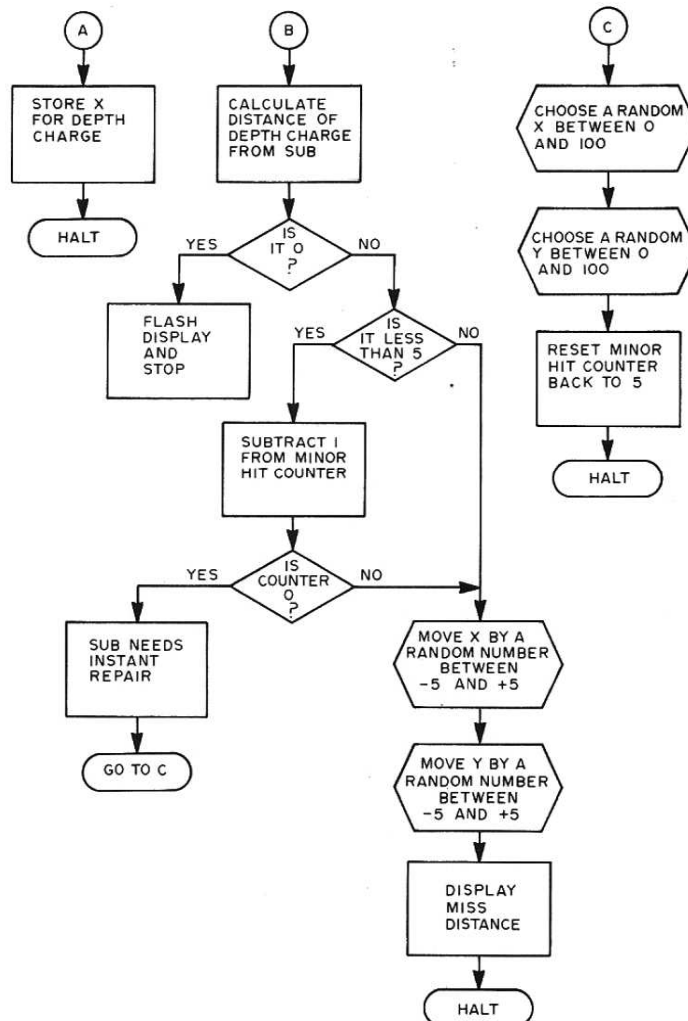
A random number generator is a short program which generates a series of supposedly random numbers. These random numbers aren't really random, since if you know how the program works you can always predict what the next number will be if you know which numbers came before. Nevertheless, a good random generator program will put out a wide variety of numbers which can be used as if they really were random — they are often called *pseudo random*.

In this program we have two random number generators. One generates a random number between 0 and 100. By using the same program twice, we get two different numbers which are used to pick a starting point for the submarine whenever we push the C key to start a new

game, or whenever the sub goes to a new location after the fifth minor hit.

The other random number generator generates a number between -5 and +5 to move the sub up to five spaces after every depth charge. This generator is also used twice, once to get a movement in the X direction, and once to get a movement in the Y direction. This second generator actually uses the first one and then modifies the answer to make it smaller. The first random number generator puts out a number between 0 and 100; this number is multiplied by 0.1 to make it between 0 and 10, and then subtracting a 5 makes the output between -5 and +5.

Random number generators usually start with the previous number, which is called the *seed*. They manipulate this seed in some unusual way, such as multiplying it by a very large number



Submarine game flowchart.

and then dividing it by some other large number, throw away a few of the digits and interchange others, and thus come up with a new "random" number which is also the new seed. In our program (Program A) the seed is kept in register 03, and the main random number generator is in locations 092 through the end. Though the actual instructions are difficult to follow unless you are very familiar with the SR-52, the program works roughly like this:

1. Take the seed and multiply it by 7^9
2. Take the result and multiply by .00001
3. Throw away all digits to the left of the decimal point
4. Save the result as the seed; it is between 0 and 1
5. Multiply the seed by 100 to give a random number between 0 and 100, and round to nearest integer.

For example, if you start with a seed of 0.4, the calculations work out like this:

$$7^9 = 40,353,607$$

$$0.4 \times 7^9 = 16,141,442.8$$

$$16,141,442.8 \times .00001 = 161.414428$$

Eliminate 161, so the seed is 0.414428.

Using this random number generator, the calculator will always go through the same sequence if you start with the same seed; with a starting seed of 0.4, it would always position the sub at X=41 and Y=24 for the first move (these are the first two numbers produced by the random number generator). The program automatically starts with a different seed from 0.4, but it always starts the first game with the same seed. If you want to make it really random so even you cannot anticipate where the sub is next, all you have to do is put a different seed into register 03.

Random number generators are usually optimized for the particular computer or calculator to be used, and the above method may not work well on other machines. The purpose of using a different method is to achieve the longest *period*. Every such random number generator will eventually start to repeat itself; the length of the sequence of numbers before numbers start to repeat is called the period, and so we want the longest possible period to make sure that the numbers stay random for at least as long as we need them.

Random number simulator
 — Not just a game
 — Check practical applications

Multiply by 100 to get 41.4428

Round to nearest integer to get 41.

If you keep repeating this over and over, the seeds and random numbers work out like this:

Seed	Random Number
0.414428	41
0.236645	24
0.494876	49
0.700157	70
0.538579	54
0.336272	34
0.697946	70
0.646506	65
0.888556	89

This lengthy discussion of random number generators has been included mainly to give you some thoughts as to how they might be of use in other games and programs as well. They are often used by scientists and statisticians as well as by games addicts to simulate random events. They can be used not only in playing submarine or blackjack, but also in such serious jobs as simulating the random arrival of customers in a store to try to calculate how long a customer will have to wait in line with a given number of salesclerks. ■