

FIRST STOS GAME FOR NEWBIES

Well, you've opened the STOS manual. You've read it and seen all the clever commands and accessories you can use. You've read and heard that all this can enable you to write your very own masterpiece. But how do you start, what do you do, how do you tell STOS to do this? These are the questions the beginner asks. In this article, I will show you how to get started and even write a simple game.

How do we start? Well, first we need to learn some commands. A program is just a group of commands that tell STOS what to do. We start with simple commands and we type out the examples in the STOS manual. This helps to give us a good idea of commanding STOS and telling it what to do.

Let's say we wanted STOS to print something on the screen over and over again. We can use the PRINT and GOTO commands like this...

```
10 print "STOSSER"
```

```
20 goto 10
```

Here we have a simple program. First, it prints the word STOSSER onto the screen, and then it goes onto the next line that tells STOS to go back to line 10 and print STOSSER on the screen again.

So, as you can see, we have grouped two commands to make a small program a small program like this is better known as a routine. A program such as a game is just a large group of routines. Let's try a routine that counts up to ten.

```
10 for X=1 to 10
```

```
20 home : print X : wait 5
```

```
30 next X
```

What we have here is a loop that adds one to the variable X, prints it on the screen, then stops when X reaches 10. The HOME command is used to tell STOS to keep the count printing on the same part of the screen, and the WAIT command stops STOS from printing too fast.

So once we have learned the commands we can easily group them. Once you've been learning them, typing out examples, and looking at other routines, you'll find it all comes clearer.

Let's program a small game. First, we have to decide what the program is going to be, and what it does. The game in question is a Guess the Number game where STOS will first choose a random number then ask the player to guess it. If the player's guess is wrong then STOS will tell him so. First...let's set up the screen.

```
10 key off : hide : mode 0
```

Rem Now to start printing the information we need on screen.

```
20 locate 0,1 : centre"NUMBER GUESSER"
```

Rem Now we need to tell STOS to choose a random number, like this.

```
30 RN=rnd(10) : if RN=0 then goto 30
```

This line will choose a different number from 1 to 10 every time STOS comes across it. With programming we are sometimes faced with problems, in this case, the RND command will sometimes choose nought as a random number. The problem is that we only want numbers from 1 to 10. So, therefore, we add an IF THEN statement to tell STOS that if it chooses nought...then go back and try again.

When STOS has chosen a number between 1 and 10, it needs to ask the player to guess the number.

```
40 locate 2,4 : print"I am thinking of a number between 1 and 10?"
```

```
50 locate 2,5 : print"Can you guess what it is?"
```

Next we need a way of letting the player enter his guess.

```
60 input A
```

This command waits for the player to enter a number and press return. Now so far, we have the RN variable which holds the random number chosen by the game, and the A variable which holds the number chosen by the player. But what if the player has typed in a number which is out of range? We can check for this with these two lines.

```
70 if A<1 then print"Your guess is too low, try again" : goto 60 80 if A>10  
then print"Your guess is too high, try again" : goto 60
```

So, if the number in the A variable is out of range, then the game informs the player to try again then goes back to the input at line 60. Now we need a line to check if the player has guessed the right number.

```
90 if A<>RN then print "That's not it, try again" : goto 60
```

This line checks if the A variable contains the same number as the variable chosen by the game. Here we are telling the game that if the player's guess is other than the number the game has chosen then to give them a chance to try again.

Finally, we need to check if the player has guessed the right number.

```
100 if A=RN then print "Well done, you've guessed the number" : stop
```

So there we have it. A small game produced by just a group of routines. We can easily improve it, such as getting the game to loop back to the start after the player wins. We could also add extras such as a score by simply adding points to a variable and printing it on the screen like this.

```
110 SC=SC+10 : print "Your score is now:";SC
```

```
120 wait 50 : cls : goto 20
```

I recommend you download [The Beginner Guide to STOS Basic](#) by MT Software as this is much more detailed.