

TI-Basic 89 Programming/A Basic Program

Preface

This page is designed to give you a little glimpse into the future of exactly what TI-89 BASIC is capable of performing. The following program is a game that displays a number on the screen and moves it around for 10 seconds. The object of the game is to memorize that number and then input it.

At first, the program in its entirety will be given (so that you have a chance to copy it to your calculator and test out what it does for yourself). Then, each block of code will be explained as to what it does so that you can get an idea of what it does. Before delving into the details of how it works, however, it may be best to finish reading the rest of this book so that you understand how the syntax and structure works.

The program

This program is called "memory." If you wish to create it in your own calculator, go into the program editor, which can be reached by the following key presses: **APPS**+**7**+**3**. Make sure that the drop down is set to "program" and name it "memory" (a-lock should already be on, so you do not need to press the alpha button).

Also, this program was designed for the TI-89 and TI-89 Titanium. TI-92, TI-92+, and Voyager 200 may have different screen sizes or key mappings, and as such that will prevent this program from working correctly. If you would like to test this program and you have the TI-92, TI-92+, or Voyager 200, please leave a message on the talk page with the screen resolution in pixels (horizontal x vertical) and what number the **ENTER** (or equivalent) key maps to.

Just a quick reference on some of the characters you will encounter here (again for the TI-89 and TI-89 Titanium):

- © can be created by pressing **♦** + **)**
- & can be created by pressing **♦** + **×** (the multiplication key)
- ≤ can be created by pressing **♦** + **<**
- ≥ can be created by pressing **♦** + **>**
- Text can be copied by pressing **♦** + **↑** (the shift key to capitalize text)
- Text can be pasted by pressing **♦** + **ESC**

The spaces after the colons are for readability only, it does not affect anything if you remove them.

```
:memory(m)
:Prgm
:©NUMBER
:FnOff 1,2,3,4,5
:If m>12 Then
:  Text "Number too large"
:  Stop
:EndIf
:If m<1 Then
:  Text "Number too small"
:  Stop
:EndIf
```

```

: int(m)→n
: Lbl a
: ClrIO
: ClrDraw
: ClrGraph
: setGraph("Coordinates","OFF")
: setGraph("Grid","OFF")
: setGraph("Axes","OFF")
: int(rand()*10^n)→x
: While x≤10^(n-1)
:   int(rand()*10^n)→x
: EndWhile
: int(rand()*50)→z
: rand(3)-2→a
: rand(3)-2→b
: 0→c
: rand(56)+10→d
: rand(95-5*(n-8))+5→e
: PxlText string(x),d,e
: PxlHorz 76
: 0→p
: For y,0,50
:   getKey()→k
:   If k=13 Then
:     Exit
:   EndIf
:   PxlOff 76,p
:   p+1→p
:   PxlOff 76,p
:   p+1→p
:   PxlOff 76,p
:   p+1→p
:   If mod(y,10)=0 Then
:     PxlOff 76,p
:     p+1→p
:   EndIf
:   If c=z Then
:     int(rand()*50)→z
:     0→c
:     rand(3)-2→a
:     rand(3)-2→b
:   EndIf
:   c+1→c
:   d+a→d
:   If d<10 Then
:     10→d
:     z→c
:   EndIf
:   If d>60 Then
:     60→d
:     z→c
:   EndIf
:   e+b→e
:   If e<5 Then
:     5→e
:     z→c
:   EndIf
:   If e>95-5*(n-8)+5 Then
:     95-5*(n-8)+5→e
:     z→c
:   EndIf
:   PxlText " ",d-8,e-4
:   PxlText " "&string(x)&" ",d,e-4
:   PxlText " ",d+8,e-4
: EndFor

```

```

:ClrDraw
:Input "What is the number?",y
:If x=y Then
:  Disp "Good Job!"
:Else
:  Disp "Sorry! The number was",string(x)
:EndIf
:Input "Again? (0=no 1=yes)",x
:If x=1 Then
:  Goto a
:Else
:  DispHome
:EndIf
:EndPrgm

```

The explanation

This section details how the above program works, broken down piece by piece. It may be best to read the rest of this book before reading this section, as basic structures and functionality will not be explained thoroughly here.

```

:memory(m)
:Prgm
:@NUMBER

```

This defines the name of the program, what parameters it takes when called, specifies that it is a Program (not a Function), and tells the catalog listing help that the first parameter is a NUMBER.

```

:FnOff 1,2,3,4,5
:If m>12 Then
:  Text "Number too large"
:  Stop
:EndIf
:If m<1 Then
:  Text "Number too small"
:  Stop
:EndIf

```

This turns off the first five functions on the graph screen, so that they will not be re-drawn while the program is running. Then, if the given number is too large (above 12) or too small (below 1), an error text box is displayed and the program execution is stopped

```

:int(m)→n
:Lbl a
:ClrIO
:ClrDraw
:ClrGraph
:setGraph("Coordinates","OFF")
:setGraph("Grid","OFF")
:setGraph("Axes","OFF")

```

This takes the floor of the given number (makes in an integer), so that only a whole amount of numbers gets displayed on screen. Then, it clears the I/O screen and Graph of all text/functions/drawings, and turns off the Coordinates, Grid, and Axes on the Graph so that the Graph screen is completely blank.

```

: int(rand()*10^n)→x
: While x≤10^(n-1)
:   int(rand()*10^n)→x
: EndWhile
: int(rand()*50)→z
: rand(3)-2→a
: rand(3)-2→b
: 0→c
: rand(56)+10→d
: rand(95-5*(n-8))+5→e

```

This initializes all of the variables. x is the number that will be displayed on-screen, so the first four lines (ending with the EndWhile) are a loop to make sure that x is of the correct length. z is how many pixels that the text will move in any one direction. a and b specify the direction the text is moving. a specifies vertical movement, so a value of -1 means it moves upwards, 0 means it does not move vertically, and 1 means it moves downwards. b specifies horizontal movement, so a value of -1 means it moves left, 0 means it does not move horizontally, and 1 means it moves right. c is a counter that increments by 1 every time the number moves one pixel. d and e are the vertical and horizontal starting points of the number (respectively).

```

: PxlText string(x),d,e
: PxlHorz 76

```

This puts the number on the screen at the d and e coordinates from the above snippet. Then, it draws a horizontal line at the bottom of the screen. This line is the status bar for how much time the player has remaining to memorize the number being displayed.

```

: 0→p
: For y,0,50
:   getKey()→k
:   If k=13 Then
:     Exit
:   EndIf
:   PxlOff 76,p
:   p+1→p
:   PxlOff 76,p
:   p+1→p
:   PxlOff 76,p
:   p+1→p
:   If mod(y,10)=0 Then
:     PxlOff 76,p
:     p+1→p
:   EndIf

```

This sets p at 0 (p is being used as a counter to see how to progress the status bar), then starts a loop that runs 51 times. At the beginning of the loop, a check is made for a keypress on the calculator's key pad. If the "ENTER" key is pressed, the loop is cut short and the part of the program where the user must type in the memorized number executes. Otherwise, it removes 3 or 4 pixels from the end of the status bar (3 normally, 4 if y -- the counter for the loop -- is evenly divisible by 10).

```

: If c=z Then
:   int(rand()*50)→z
:   0→c
:   rand(3)-2→a

```

```

:   rand(3)-2→b
:   EndIf
:   c+1→c

```

This checks if the text has moved z pixels in the current direction, and if it has the text direction changes.

```

:   d+a→d
:   If d<10 Then
:       10→d
:       z→c
:   EndIf
:   If d>60 Then
:       60→d
:       z→c
:   EndIf
:   e+b→e
:   If e<5 Then
:       5→e
:       z→c
:   EndIf
:   If e>95-5*(n-8)+5 Then
:       95-5*(n-8)+5→e
:       z→c
:   EndIf
:   PxlText "           ",d-8,e-4
:   PxlText " "&string(x)&" ",d,e-4
:   PxlText "           ",d+8,e-4
:EndFor

```

The beginning of this code is what actually "moves" the text. Two checks are made for both d (the vertical coordinate) and e (the horizontal coordinate) to make sure that the text hasn't moved to a spot where it would no longer be visible in its entirety. Then, it displays the text padded by spaces so that the previous text gets overwritten (instead of leaving a trail behind). After it displays the text, it either continues executing the loop or stops if the loop counter is 50.

```

:ClrDraw
:Input "What is the number?",y
:If x=y Then
:   Disp "Good Job!"
:Else
:   Disp "Sorry! The number was",string(x)
:EndIf
:Input "Again? (0=no 1=yes)",x
:If x=1 Then
:   Goto a
:Else
:   DispHome
:EndIf
:EndPrgm

```

This wipes the graph screen (so one cannot simply toggle back to it to see the number), then asks on the I/O screen for the number. Different text will display if the player inputs it correctly or incorrectly. Then, it asks if the player would like to try again. If they agree, it jumps back to the line saying "Lbl a". If they do not agree, it displays the home screen then stops program execution

Retrieved from "<http://en.wikibooks.org/w/index.php?title=TI->

Basic_89_Programming/A_Basic_Program&oldid=2351075"

- This page was last modified on 21 May 2012, at 23:13.
- Text is available under the Creative Commons Attribution/Share-Alike License; additional terms may apply.
By using this site, you agree to the Terms of Use and Privacy Policy.