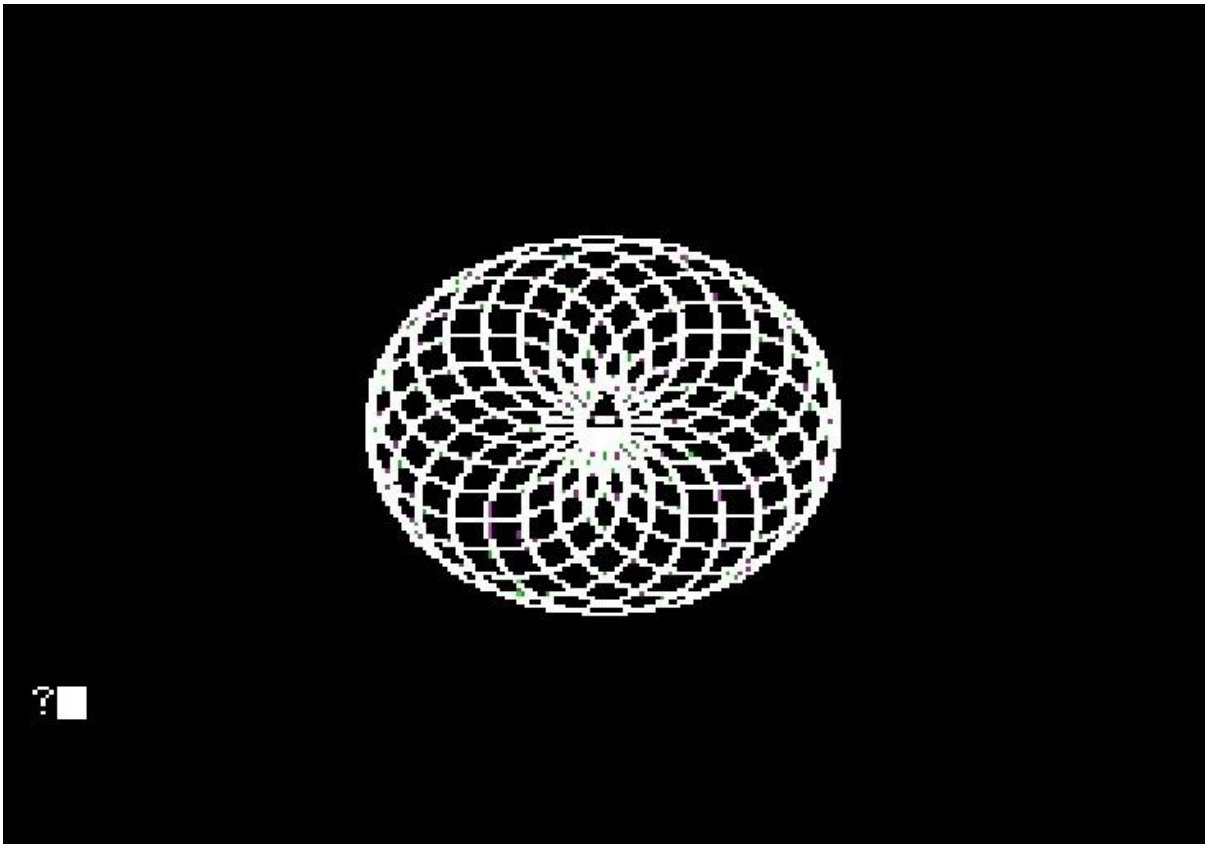
Program in Logo on an Apple II

Created by Matthew Goodrich



https://learn.adafruit.com/program-logo-on-an-apple-ii

Last updated on 2021-11-15 07:42:55 PM EST

Table of Contents

Overview

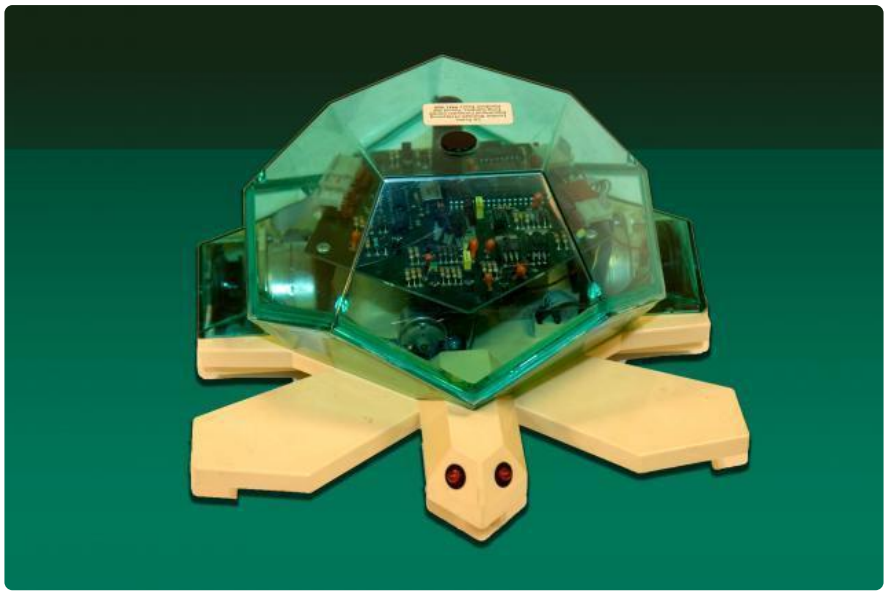
You may have seen the turtle graphics library that ladyada ported to CircuitPython (https://adafru.it/Fck) and thought, "Wow, that's cool! But can I do that on 30+ year old hardware?" Or if you're above a certain age you may remember doing something similar in school on an Apple II.

Let's set up an Apple II emulator and run old school Logo to make our own turtle graphics!

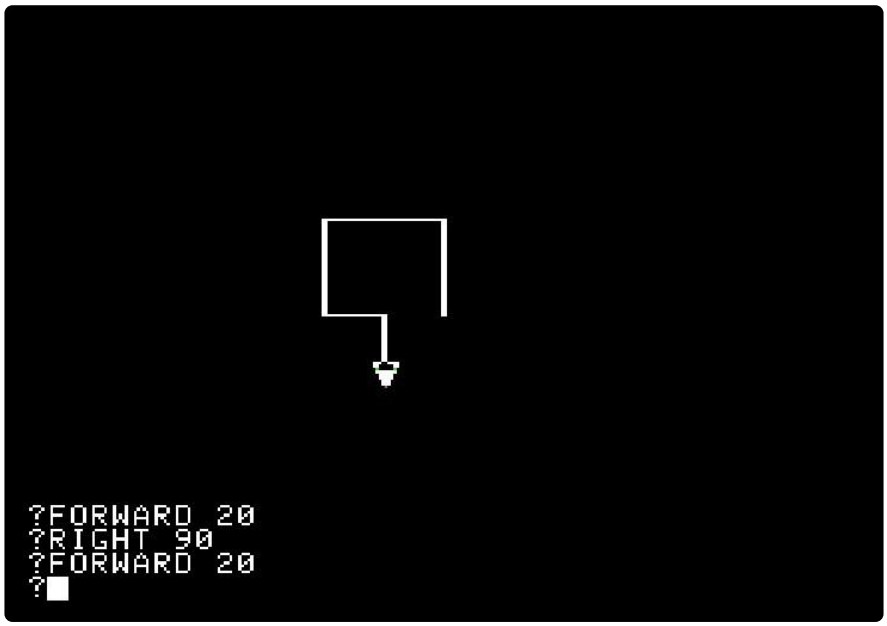
What is Logo?

Logo is an educational programming language designed to teach basic concepts.

Logo goes back much earlier than the Apple II, it was developed in 1967 by Cynthia Solomon, Wally Feurzig, and Seymour Papert. That's five years before C and 24 years before Python! The three worked at Bolt, Beranek, and Newman (BBN), famous for all kinds of other computing history. BBN built the first Interface Message Processors (early routers) in 1968 for the ARPANET, which would evolve into the modern internet.



Logo includes commands used for drawing on the screen using a cursor known as a Turtle. Small robots called Turtles had been developed as far back as the 1940s to teach mechanical engineering and computer science. Logo uses a virtual turtle on the screen and can be linked to a physical turtle which moves in response to commands. When people remember Logo they usually talk about the Turtle graphics.



Enough with the history lesson, let's get started!

Choose Your Emulator

The easiest way to get started is to use a Javascript Apple II emulator (shown in an image above). This one even has Logo ready to go: https://www.scullinsteel.com/apple//e#logo (https://adafru.it/Fcl)

If you'd like to set up a native emulator on your system, keep reading. Otherwise you can use the Javascript one above and skip ahead to the next section.

Windows, Mac, Linux

Windows

AppleWin is a good choice for Windows systems and it's open source! Get the latest version here: https://github.com/AppleWin/AppleWin/ (https://adafru.it/Fcm)

Just extract the zip file and it's ready to go.

Linux

For Linux you can run LinApple (https://adafru.it/Fcn), a port of AppleWin.

All Systems

There's also EPPLE II (https://adafru.it/Fco) which runs on Windows, Mac, and Linux, and is also open source. It's a little harder to get set up and running. You'll probably need DskToWoz2 (https://adafru.it/Fcp) to convert disk images to the Woz format that EPPLE II uses.

Raspberry Pi

RetroPie (https://adafru.it/tjB) comes with an Apple II emulator.

LinApple has also been optimized for the Pi, check out linapple-pie (https://adafru.it/Fcr).

Disk Images of Software

For these emulators, you'll need the disk images for Logo, those can be found at Apple2Online (https://adafru.it/Fcq).

Hello, Turtle!

When you first start Logo you'll reach a ? prompt. From here you can enter Logo commands.

We could write Hello, World, but that's boring. Let's get straight to the good stuff and draw some Turtle graphics!

Type in this line and watch the Turtle go! It will take quite a while to finish if your emulator is running at normal speed:

REPEAT 20 [REPEAT 180 [FD 1 RT 2] RT 18]

You can see the results at the bottom of the page.

Now that we've done something cool, let's unpack it and see what the commands did. REPEAT takes a number and a list, then it runs the list that many times. With the first one we're repeating whatever is in the list 20 times. A list is contained in square brackets and is made up of Logo objects, in this case we have two commands: another REPEAT with its own list and RT 18.

The inner REPEAT has its own list, [FD 1 RT 2]. These two commands are shorthand versions of turtle commands FORWARD and RIGHT. It tells the turtle to move forward one unit, then turn right 2 degrees. Repeat this 180 times and you get a circle (2 degrees * 180 = 360)! After completing the REPEAT there's another RT, so once the circle is done we turn right 18 degrees. See where this is going? Repeat that 20 times and you get another circle (20 * 18 = 360). Now we've got a circle of circles!

You can clear the screen by typing CLEARSCREEN. Try doing REPEAT 180 [FD 1 RT 2] and you'll see one component of the larger drawing that we just made.

Commands

Here's an overview of some turtle commands and their shorthand equivalents:

CLEARSCREEN - Clear the screen. (Shorthand: CS)

HIDETURTLE - Don't show the turtle cursor. (HT)

SHOWTURTLE - Show the turtle cursor. (ST)

HOME - Move back to the home position.

FORWARD steps - Move forward steps. (FD)

BACK steps - Move back steps. (BK)

LEFT degrees - Turn left this many degrees. Negative degrees work too, they'll turn it right. (LT)

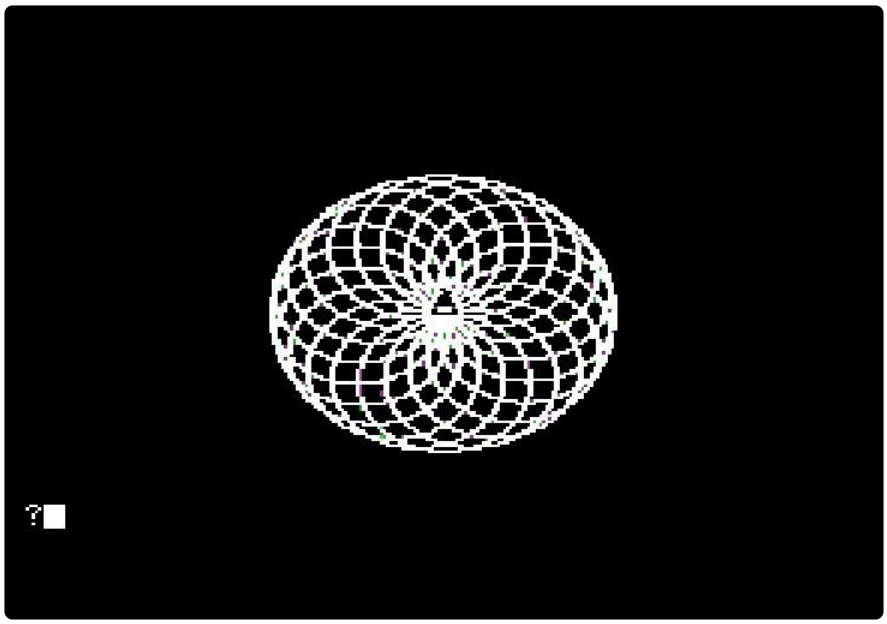
RIGHT degrees - Turn right this many degrees. (RT)

SETHEADING degrees - Turn to an absolute heading of degrees. (SETH)

SETPOS [x y] - Set the position to x, y coordinates. These are Cartesian, so 0,0 is the middle of the screen.

SETX x - Set the horizontal position to x.

SETY y - Set the vertical position to y.



More Logo

Logo has some of the basic language features you'd expect - variables, functions, and conditionals.

Variables

Define a variable using MAKE name value. Logo normally interprets any word as a procedure call, so to use a literal word you can prefix it with a double quote.

```
MAKE "A 8
```

You can use a variable by prefixing the name with a colon:

```
PRINT :A
8
```

Variables created with MAKE are global. The only local variables in Logo are procedure arguments.

Functions (Procedures)

Logo functions are called procedures, and they're defined like this:

```
TO HELLO
PRINT "HI
END
```

TO specifies the name of the function, the lines after are the function body, and it's all closed out with END. You can call a procedure by typing its name. You can also call a procedure from within another procedure. Logo handles recursion, so a procedure can call itself!

To run it, just type the procedure name:

```
?HELLO
HI
```

Procedure arguments are defined by listing them after the procedure name:

```
TO HEY "NAME
PRINT "HI
PRINT :NAME
END
```

If you want to delete a procedure, use the ERASE command.

```
ERASE "HEY
```

Conditionals

Logo's `IF` statement is straightforward: `IF test list1`

If test is true, run list1. You can add another list to the end (`IF test list1 list2`) and list2 will be run if test is false.

You can break this down a bit using TEST, IFTRUE, and IFFALSE. TEST test will save the result of test and use it for any calls to IFTRUE and IFFALSE.

These are equivalent:

```
IF :A=8 [PR [A IS 8]] [PR [A IS NOT 8]]
```

```
TEST :A=8
IFTRUE [PR [A IS 8]]
IFFALSE [PR [A IS NOT 8]]
```

Example

Let's put these together and draw something. First we'll define a square function:

```
TO SQUARE "LEN
REPEAT 4 [FD :LEN RT 90]
END
```

Try it out! SQUARE 50

How about a hexagon?

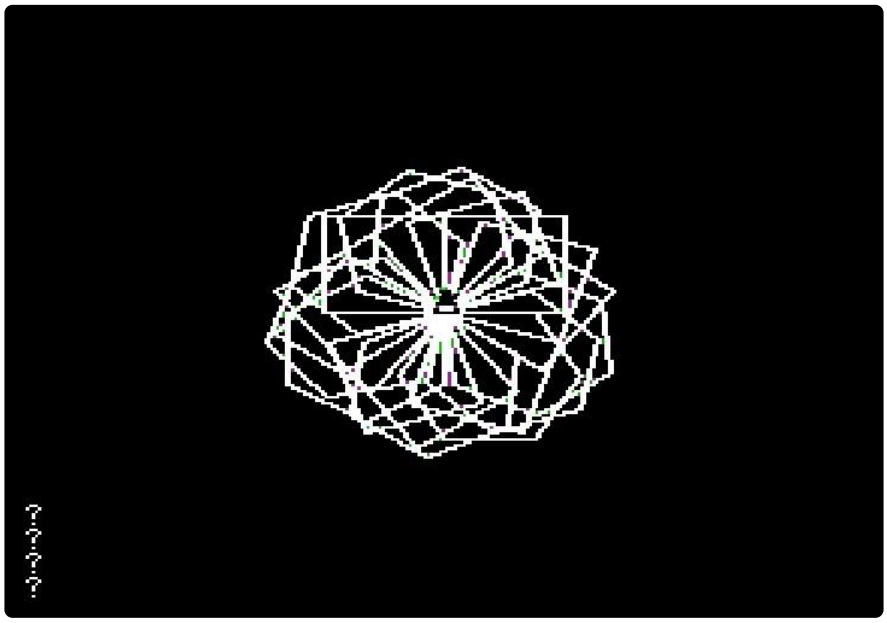
```
TO HEXAGON "LEN
REPEAT 6 [FD :LEN RT 60]
END
```

Now let's put them together and draw something.

```
REPEAT 20 [IF RANDOM 2=1 [SQUARE 40] [HEXAGON 30] RT 18]
```

RANDOM num will output a random number from 0 to num. Here we use it to decide whether to draw a square or a hexagon.

That's a pretty simple example, see what you can do with Logo! There are a lot more features and we can't cover them all here, check out the Logo manuals in Resources for a full reference.



Resources

Emulators

- Scullin Steel Apple IIe Javascript Emulator (https://adafru.it/Fcl)
- AppleWin (https://adafru.it/Fcm)
- LinApple (https://adafru.it/Fcs)
- LinApple-Pie (https://adafru.it/Fcr)
- Epple II (https://adafru.it/Fco)
- DskToWoz2 (https://adafru.it/Fcp)
- RetroPie (https://adafru.it/tjB)

Apple II Software

- Apple2Online (https://adafru.it/Fcq)

History

- CircuitPython Turtle Graphics (https://adafru.it/Fck) - for a modern take on the Turtle.
- Logo Foundation at MIT (https://adafru.it/Fct)

Reference

Logo books from the excellent Internet Archive:

Apple Logo II Reference Manual (https://adafru.it/Fcu) - Logo II was a later version, but most of the manual is still applicable to the original one that we're using.

Introducing Logo for the Apple II, TI 99/4A, and Tandy Color Computer (https://adafru.it/Fcv)