Leone Learning Systems, Inc.
Wonder. Create. Grow.

Leone Learning Systems, Inc.
237 Custer Ave
Evanston, IL 60202
Email tj@leonelearningsystems.com

Phone 847 951 0127
Fax 847 733 8812

Mathematical Problem-Solving with Logo – Part 1

Beginning work on Application 1.3

TJ Leone

October 2004

Introduction

This guide accompanies the text *Approaching Precalculus Mathematics Discretely* by Philip G. Lewis.

In this unit, we begin a discussion of Application 1.3 on pages 11-14. There are different ways to approach this problem mathematically. Here we seek to clarify the particular approach taken by Lewis, and discuss strategies for problem-solving with Logo.

Preliminaries

This section reviews background knowledge that should have to be able to complete application 1.3.

Cartesian coordinates

It's assumed you know how to plot points on a plane using Cartesian coordinates (http://mathworld.wolfram.com/CartesianPlane.html). If you need more review on coordinates, check out http://www.langara.bc.ca/~acooper/mathlabs/cartesian/index.htm, Activity#1 – Getting Started. The activity is an applet, so you need to have java enabled on your browser. If you have java enabled and you still can't see the applet, try resizing your browser window and then temporarily obscuring that window (by covering it with another window or partially dragging it off screen) so that it has to refresh itself.

Review the section **Coordinatizing the Plane** on pages 6 and 7. Be sure you understand why

```
setpos [40 10]
```

places the turtle in the upper right quadrant of the screen (http://mathworld.wolfram.com/Quadrant.html) while

```
setpos [-50 -10]
```

places the turtle in the lower left quadrant.

Similar triangles

Similar triangles are triangles that have the same size but not necessarily the same shape. The following principles of similar triangles are important for application 1.3:

- Corresponding angles of similar triangles are congruent.
- Corresponding sides of similar triangles are in proportion.
- Two triangles are similar if tow angles of one triangle are congruent respectively to two angles of another.
- A line parallel to a side of a triangle cuts off a triangle similar to the given triangle.

Understand what is required

Before you try to solve a problem, it helps to understand what's required for a solution. In Application 1.3, there are four main tasks that must be accomplished:

1. Plot two points, A (the house) and B (the barn).
2. Reflect B over the *x* axis.
3. Draw a line from A to the reflected image of B (B′).
4. Determine the point P at which the line intersects the *x* axis.

If any of these tasks are unclear to you, review Reflections on Application 1.1, pages 4 and 5.

Start with a concrete example

In Application 1.3, an important thing to notice is that Lewis begins to attack the problem with a concrete example. In the general case, we don't know where points A or B are, so the problem is tough to visualize.

So Lewis picks two specific points. He puts point A at `[-30 50]` and puts point B at `[150 100]`. Notice what this does for us. First of all, it gives us an easier way to think about the problem. It's not too hard to think generally how you would plot two points (you can just use `setpos`), but it's a bit more work to think in general terms about completing tasks 2-4 in the section above.

On the other hand, if we know that B is `[150 100]` then it is pretty easy to see that B′ must be `[150 -100]`. This leads us to the general observation that the reflected image of any point [*x y*] over the x axis is [*x* –*y*].

In other words, our xreflect procedure should be:

```
to xreflect :pt
output list first :pt minus last :pt
end
```

Having a concrete example gives us another important advantage—we can now test our procedure to make sure that it works.

Type in your `xreflect` procedure and test it with our input for B:

```
show xreflect [150 100]
```


Test early and often

Another important strategy for problem-solving with computers is to test each procedure as you write it. In the section above, we tried out `xreflect` with one particular point. Did it work? If not, see if you can figure out what went wrong.

Once your procedure is working, how else can we test it? One strategy is to try to pick inputs that might give your procedure some trouble. For example, if your procedure sometimes divides its input by zero, you would want to provide zero as an input to make sure this doesn't cause problems.

Another strategy is to try a number of arbitrary inputs to see what happens. Logo has an operation called `random` for picking random numbers. Here's the entry for RANDOM in the MSWLogo Reference Manual (choose Help/Index from the main window, then look for the entry RANDOM):

> RANDOM
>
> num RANDOM num1
>
> Outputs a random nonnegative integer less than its input, which must be an integer.
>
> num:(INTEGER) Random number generated.
>
> num1:(INTEGER) Range for random number to be generated.
>
> Example:

```
repeat 5 [show random 10]
6
8
3
0
9
```

An important thing to understand about these manual entries is the way that inputs and outputs are shown. On the second line of the RANDOM entry you see:

num RANDOM num1

The first num indicates that the output of the random operation is a number. The num1 indicates that the input is also a number. Details about the output and input numbers are given on the lines that start with num: and num1: respectively.

Anyway, the reason I brought up the random operation is that you can use it to generate random points on the screen which you can then use to test `xreflect`.

Test early and often (continued)

Here's a procedure I wrote to generate random points:

```
to point :x :y
output list :x :y
end

to random.point.a
output point (minus random 100)  (random 100)
end

to random.point.b
output point (random 100)  (random 100)
end
```

Of course, this procedure also needs to be tested:

```
show point 20 20
[20 20]
show point 5
not enough inputs to point
show point 0 0
[0 0]
show random.point.a
[-18 72]
show random.point.a
[-3 96]
show random.point.b
[77 60]
show random.point.b
[35 27]
```

In which quadrant will point A always occur? In which quadrant will point B always occur? Do those restrictions fit the problem? Why or why not?

Text `xreflect` on some random points and make sure they work. In Part 2, we will talk about the use of figures like the one on page 14 to analyze problems.

The Author

TJ Leone owns and operates Leone Learning Systems, Inc., a private corporation that offers tutoring and educational software. He has a BA in Math and an MS in Computer Science, both from the City College of New York. He spent two years in graduate studies in education and computer science at Northwestern University, and six years developing educational software there. He is a former Montessori teacher and currently teaches gifted children on a part time basis at the Center for Talent Development at Northwestern University in addition to his tutoring and software development work. His web site is http://www.leonelearningsystems.com.