

MATH
ART

using SeeLogo

Summer 2009

Math Art Using SeeLogo, Summer 2009

Overview.....	3
Section I: Getting Started and Using the Mouse Tool Bar.....	5
SeeLogo Start-Up Screen.....	5
Drawing Pictures with the Mouse Tool Bar.....	6
Transforming Pictures with the Mouse Tool Bar.....	9
Section II: Samples of Art Tools.....	13
The Spiral of Life	13
Writing on the Screen.....	17
Tiling in SeeLogo	19
Creating a Mandala.....	20
Playing with Colors (HSB).....	22
Section III: Drawing Simple Pictures.....	25
Drawing Closed Shapes.....	26
Drawing Lines.....	28
Drawing Arcs and Curves.....	33
Section IV: Transforming Pictures.....	36
REPEAT Command.....	39
SPIN Command.....	40
SPINR and SPINL Commands.....	42
MIRROR Command.....	44
GROWBY Command, Part 1	47
GROWBY Command, Part 2.....	48
Transformation Commands and Combinations.....	49
Section V: SeeLogo Programming and Dynamic Art.....	50
Introduction to SeeLogo Programming.....	50
Functions and Time in SeeLogo.....	53
Periodic Functions in SeeLogo.....	54
Exploring Dynamic Art in SeeLogo	56
Section VI: Quizzes.....	60
Closed Shapes Quiz.....	61
REPEAT Quiz 1.....	62
REPEAT Quiz 2.....	66
SPIN Quiz.....	69
SPINRL Quiz.....	73
MIRROR Quiz.....	76
GROWBY and GROWGBY Quiz	78
Single Transformation Commands Quiz.....	80
Section VII: Project Ideas.....	87

Overview

SeeLogo is a computer language that connects words and pictures. It is possible to go from words to a picture. You type special words, called commands, on the command line or in the editor and the commands then generate a picture on the screen. But it is also possible to go the other way, from a picture to words. You use the mouse to draw a picture on the screen and the editor then generates the words, the commands, that can later be used to regenerate the picture. By seeing the pictures and the words at the same time, you will come to understand mathematical ideas in a holistic and meaningful way.

The book is divided into six sections. Each section is divided into units.

Section I. Getting Started and Using the Mouse Tool Bar

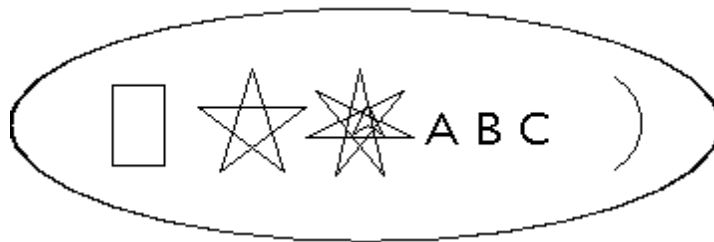
The first unit in this section provides a general introduction to the start-up screen. The remaining two units deal with the mouse tool bar user interface. In the second unit, you will learn how to draw pictures by using the drawing tools in the mouse tool bar. In the third and last unit of this section, you will learn how to transform one picture into other pictures by using the transformation tools in the mouse tool bar.

Section II. Samples of Art Tools

In this section, you will learn, by using, different features of SeeLogo, which can later be used in art projects.

Section III. Drawing Simple Pictures

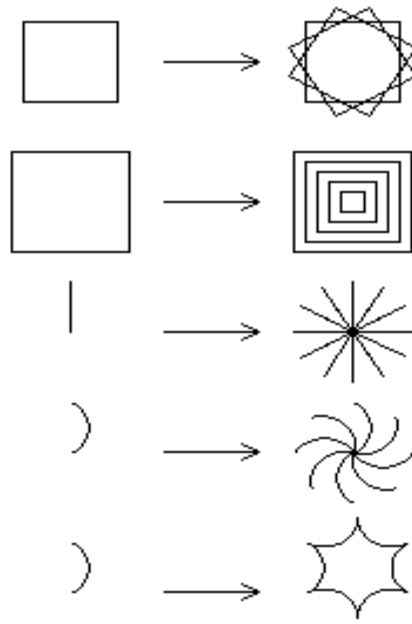
In this section, you will be engaged in drawing basic shapes and simple pictures, such as lines, arcs, boxes, circles and stars.



SeeLogo uses a cursor, called the “turtle,” to draw pictures. The turtle is represented on the screen by a shape like an arrowhead, glider or dart, although it is sometimes (incorrectly) called a triangle. The turtle has a location, the x and y coordinates of which are displayed under the drawing screen as tx and ty. It is also has a direction, the angle of which appears as tA. For example, typing the command CIRCLE 50 causes a circle of diameter 50 to be drawn with its center at the current location of the turtle. Typing the two commands JT 100 60 CIRCLE 40 draws a circle of diameter 40 centered at the location (100, 60).

Section IV. Transforming Pictures

In this section, you will learn how to transform a simple picture into a more complex picture. Once a basic design is created, it can be used to create a symmetric pattern by using a transformation. The following picture illustrates several transformations.



Section V. SeeLogo Programming and Dynamic Art

This section is more technical in nature. It uses mathematical functions and concepts from computer programming. The results are fascinating.

Section VI. Quizzes

This section is comprised of quizzes that will help you evaluate your learning.

Section VII. Project Ideas

The last section contains suggestions for projects.

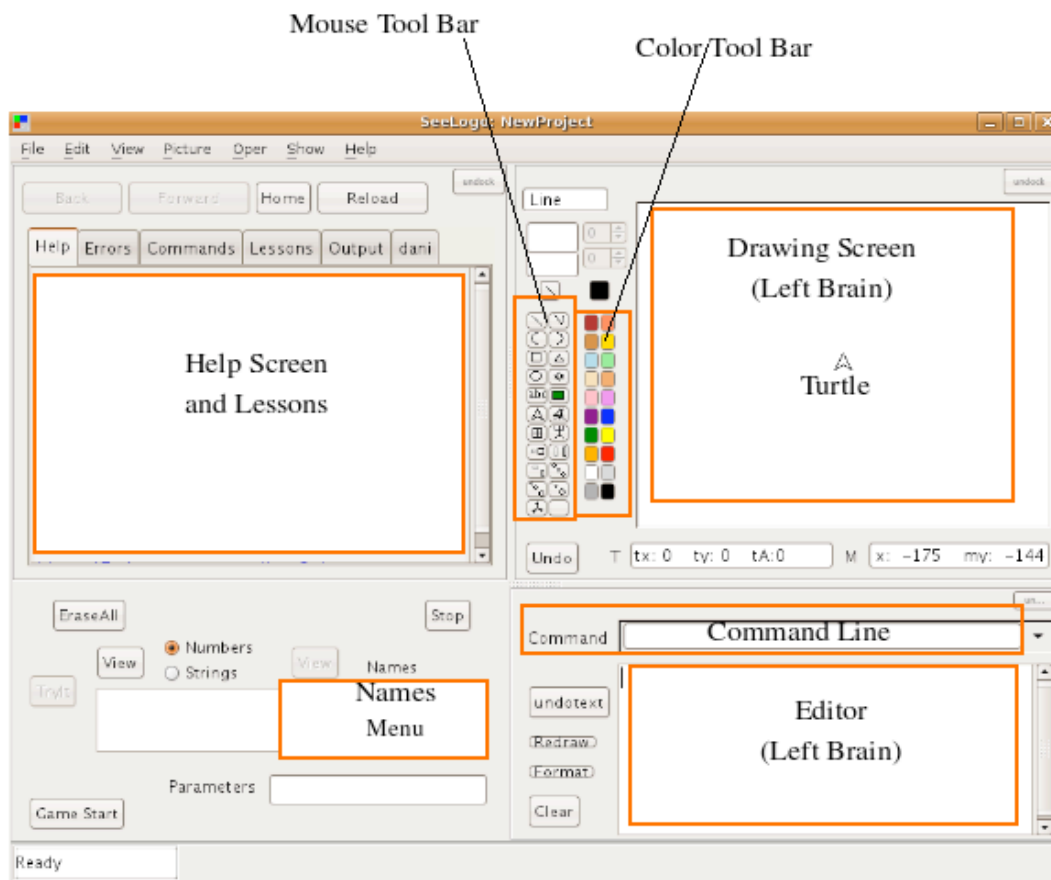
Section I: Getting Started and Using the Mouse Tool Bar

SeeLogo is a mathematical computer language. The user can either operate the mouse or type commands that tell the computer what to do. Moving the mouse and clicking on different tools automatically generate commands that appear in the editor. Alternatively, you can type commands directly in the command line or in the editor and the picture appears on the screen. The language uses the metaphor of a turtle, which is represented by the small arrowhead, glider or dart shape that appears in the middle of the screen when SeeLogo starts. By typing commands in the command line, you tell the turtle what to do. For example, if you type FD 50, the turtle will move 50 steps forward from its current location. By typing CIRCLE 50, the turtle will draw a small circle of diameter 50 around itself, and so on.

SeeLogo Start-Up Screen

Purpose: To learn how to start using SeeLogo.

Overview: The SeeLogo user interface contains three main windows, the **help window**, the **drawing window** and the **editor window**. The drawing window contains the **drawing screen**, which appeals to the right brain. The editor window contains the **editor**, which appeals to the left brain. Pictures may be created either by using the **mouse tool bar** in the drawing window or by typing commands into either the **command line** or the editor in the editor window.



When you type commands into the command line in the editor window and press the Enter key, you will see a picture in the drawing screen. Note that the text is automatically entered into the editor.

The help window contains the **help screen**, where you can read helpful information on many topics, explanations of error messages, summaries of all the SeeLogo commands and the lessons for this and other courses.

As we have mentioned, you can also use the mouse tool bar to draw pictures. The corresponding program is automatically entered into the editor.

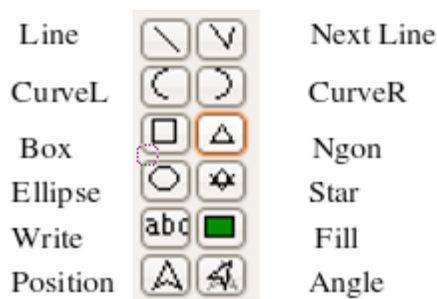
Activities

- Click the upper left button in the mouse tool bar.
- Use the mouse to draw some lines.
- Notice the text change in the editor.
- Press the clear button.
- Type the command CIRCLE 80 in the command line press the Enter key.
- Watch what happens in the drawing screen and the editor.

Drawing Pictures with the Mouse Tool Bar

Purpose: To learn to use the shape drawing tools from the mouse tool bar.

Overview: The tools in the upper part of the **mouse tool bar** are used to draw shapes. In this unit, we will explore these twelve shape drawing tools. The following diagram shows the upper mouse tool bar with the names of its twelve tools.



When you use a tool from the mouse tool bar, the corresponding program is automatically entered into the editor. Each time you use a tool, it creates text in the editor, spelling out one or more commands along with their numerical parameters. As you experiment, be sure to look both at the text in the editor, appealing to the right brain, and at the picture on the drawing screen, appealing to the left brain. Always try to understand how the program in the editor and the picture on the screen are related. Change some of the numbers in the editor and press the Redraw button (or F9) to see the effect of the change and to understand better the meaning of each command and its parameters.

For each of the tools, we show below an example of the commands that appear in the editor, but the numbers will vary, of course. We also provide a brief explanation. Many of the texts that are generated in the editor start with the command JT, meaning **Jump To**, which is followed by two numerical parameters. These numbers specify the location to which the turtle jumps, that is, moves without drawing. The first number is called the **x coordinate** and specifies the turtle's horizontal position in

pixels. The second number, called the **y coordinate**, specifies its vertical position in pixels.

- The Line and the Next Line tools are used to draw lines. The Line tool specifies both a initial point and a final point. Example: JT -62 58 MT 67 35. The two numbers that follow the JT command specify the x and y coordinates of the initial point of the line, the point from which the turtle draws the line. The command MT means Move To. The two numbers that follow the MT command specify the x and y coordinates of the final point, the point to which the turtle draws the line. The Next Line tool specifies only a final point. The initial point of the line is the final point of the last line drawn by the Line or Next Line tool. Example: MT -26 57.
- The CurveR and the CurveL tools are used to draw curves. You use the mouse to draw a straight line, called the chord, that points from the beginning to the end of the curve. The CURVER command draws a curve that bends to the right of the chord and, of course, CURVEL draws a curve that bends to the left of the chord. You may also set a number labeled “angle” that specifies the angle of the arc of the curve in degrees. Example: JT -137 -86 COLOR green [LT 300 CURVEL 120 133 RT 300]. The number 300 here specifies the direction angle of the chord in degrees. The number 120 specifies the arc angle of the curve in degrees and the number 133 specifies the length of the chord in pixels.
- The Box tool is used to draw rectangles. Example: JT 52 38 COLOR purple [BOX 42 48]. The numbers 42 and 48 define the width and height of the box.
- The Ngon tool is used to draw regular polygons. You can set a number labeled “sides” to specify the number of its sides. Example: JT -112 -73 COLOR orange [SIZE 93% [NGON 3]]. The number 93%, which may also be written 0.93 or 93/100, specifies the size of the polygon. The number that follows the NGON command specifies number of sides or of vertex points of the polygon. Here, the number 3 specifies a 3-sided regular polygon, that is, an equilateral triangle. If you change this number to 4, you get a square. If you set it to 5, you draw a regular pentagon, and so on.
- The Ellipse tool is used to draw circles and ovals. If the number labeled “circle?” is set to 1, it draws circles. Example: JT -23 -30 COLOR green [CIRCLE 41]. The number that follows the command CIRCLE specifies the diameter of the circle in pixels. If the “circle?” number is set to 0, you may draw ellipses instead.
- The Star tool is used to draw star-like shapes. You set the “points” and “skip” numbers to determine the shape of the star. Example: JT -72 67 COLOR red [SIZE 87% [STAR 5]] results from setting “skip” to 2 and “points” to 5. You can draw even more interesting pictures by setting these parameter to larger numbers. Example: STARSK 3 7 results in a star with 7 points and lines are drawn between every third point. The two numbers that follow the STARSK command, 3 and 7, specify the “skip” and “points” parameters, respectively.
- The Write tool is used to write text on the drawing screen. Example: COLOR lightblue [GT -54 -99 STRETCH 100% [SIZE 196% [WR “Hi there”]]]. The WR command is followed by a quoted string, which is printed on the drawing screen. The GT command is similar to the JT command.
- The Fill tool is used to fill enclosed areas with color. You can choose colors from the **color tool bar** between the drawing window and the mouse tool bar. But for many more colors, click on the currently selected color box above the color tool bar to access the color selection window. After selecting the Fill tool and choosing a color, click on an enclosed area, and it will be filled with that color. Example: JT -116 -78 FILL blue.

There are two more tools in the upper mouse tool bar. The Position and Angle tools are used to determine the location and orientation of the Turtle.

Activities

Play with the tools in the upper tool bar. Choose different colors from the color tool bar or the color selection window. Before starting a new picture, press the Clear Button to erase the screen.

- Use the Line and Next Line tools to draw some colorful straight lines.
- Use the CurveR and CurveL tools to draw some colorful curved lines. How are the curve and its chord related?
- Use the Box and Ellipse tools to draw a rectangle with a circle or an ellipse inside it or overlapping it. Fill these shapes with different colors. Use the Ngon tool to draw polygons with different numbers of sides.
- Use the Star tool to draw stars with different numbers of sides and skips. What does the “skip” number mean?
- Use the Write tool to write some words in different colors.

Project

Draw a picture that includes all the shapes we have discussed so far. Use all ten tools in the upper mouse tool bar.

In case there are labeled parameter boxes, be sure to change the numbers in the boxes. The sizes and locations of the shapes are not important. Try to draw as many shapes as you can. The following picture is provided as an example.



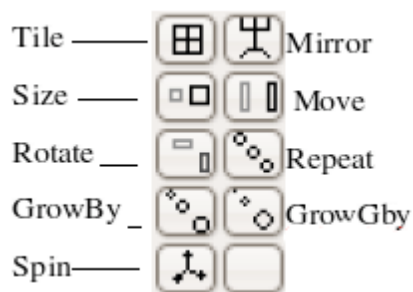
Transforming Pictures with the Mouse Tool Bar

Purpose: To transform pictures using the transformation tools in the lower mouse tool bar.

Overview: Once a picture has been drawn, it can be transformed in several different ways to create new pictures. It can be copied, reflected, rotated, scaled and changed in other ways. The picture that we start with we will refer to as the seed picture or, if the picture is of a single basic shape, the seed shape. In this unit, we will learn how to transform seed pictures using the transformation tools in the lower

mouse tool bar.

Currently, there are nine transformation tools located in the lower mouse tool bar. The names of these tools are given in the following diagram of the lower mouse tool bar.



In this unit, we will explain briefly the functions of some of these tools – what they do – and suggest some activities and projects – what to do with them. Start a new picture each time.

Tile Tool

Function

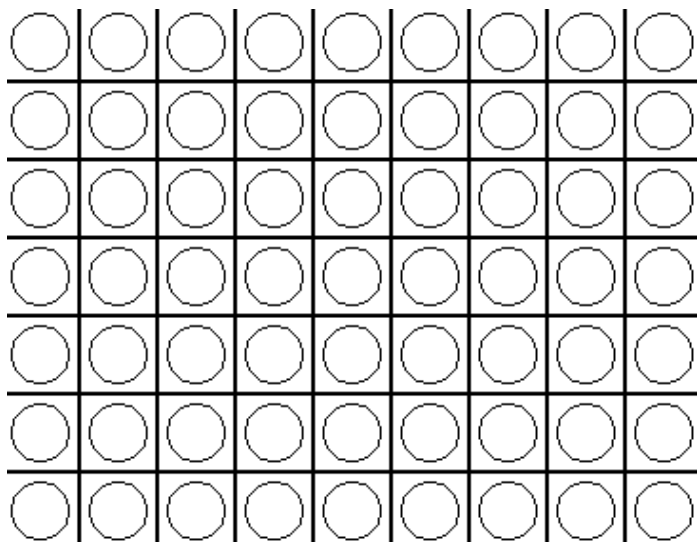
The Tile tool is used draw a two-dimensional, rectangular array of copies of a given seed picture, much like a floor tile or wallpaper design.

Activity

Use the box and circle tool to draw a seed picture, like this picture of a circle inside a square.



Select the Tile tool and click on the screen to draw a rectangular array of copies of the seed picture.



Mirror Tool

Function

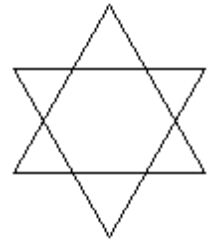
The Mirror tool is used to reflect a seed picture through a line, much as a mirror reflects an object through a plane.

Activity

Create a basic shape for your seed picture.

Use the Mirror tool to reflect this seed picture.

Now, experiment more: repeat this activity with different seed pictures and different locations and orientations of the mirror line.



Project

Use the Ngon shape tool and the Mirror transformation tool to create this picture.

Repeat Tool

Function

The Repeat tool is used to draw a one-dimensional, linear array of copies of a given seed picture. You specify the number of copies to be drawn, the direction of the line of copies and their spacing.

Activity

- Draw a seed picture, say, a box, by using the shape tools or by typing commands.
- Click on the Repeat transformation tool.
- Click and drag the mouse. If the “count” parameter is set to 6, you will see six small circles in a line. The mouse controls the distance between the circles and the direction in which the line of circles points.
- Release the mouse and observe the effect.
- Experiment: repeat this activity with different seed pictures, different “count” parameters, different directions of the line of copies and different spacings of the copies.

Growby Tool

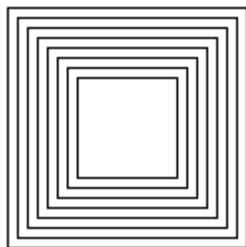
Function

Like the Repeat tool, the Growby tool draws multiple copies of a given picture, but the copies grow linearly in size: each copy is larger than the previous copy by a fixed amount.

Activity

- Draw a seed picture, say, a box, by using the shape tools or by typing commands.
- Click on the Growby transformation tool.
- Set the “count” and the growth rate or “%” parameters.
- Click and drag the mouse. You will see the “count” number of small circles in a line. Drag the mouse to the center of the screen, so that all the small circles come together.
- Release the mouse and observe the effect.
- Experiment more: repeat this activity with other seed pictures, other values of the “count” and growth rate or “%” parameters, other directions of the line of copies and other spacings of the copies.

When the basic picture is a box at the center of the screen, the “count” parameter is set to 8 and the “%” parameter is set to 20, you can get a picture like this.



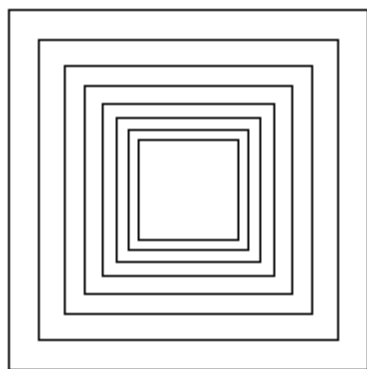
GrowGby Tool

Function

Like the Growby tool, the GrowGby tool draws multiple copies of a given picture that grow in size, except that the copies grow geometrically in size: each copy is larger than the previous copy by an amount which is a fixed proportion of its size.

Activity

Repeat the Growby tool activity with the GrowGby tool. With the same seed shape and the same parameter settings as before, but now with the GrowGby tool, you can get this picture.



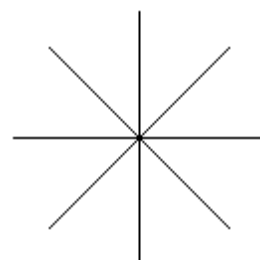
Spin Tool

Function

The Spin tool is used to draw multiple copies of a seed picture rotated by equal angles in a full circle around a given point.

Activity

- Use the Line tool to make a line that starts from the center of the screen.
- Select the Spin tool.
- Set the “count” parameter to 8.
- Drag the mouse, so that all the small circles converge to a single circle.
- Release the mouse.
- You get the picture shown here.
- Experiment more: use the Spin tool with other seed pictures and other settings of the “count” parameter.



Transformation Tools

Project

Create a single SeeLogo project file with six different picture files, each containing a picture made by one of the six transformation tools (Tile, Mirror, Repeat, Growby, GrowGby and Spin) that you have learned to use in this unit.

Section II: Samples of Art Tools

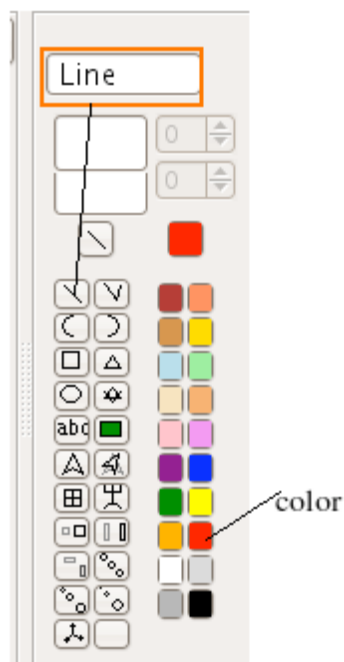
In the first unit, you will learn to make a picture that changes with time and that is called **The Spiral of Life**. In the next two units, you will learn how to put words on the screen and how to control colors. The last two units will focus on tiling the screen (wall paper designs) and on creating circular designs called mandalas.

The Spiral of Life

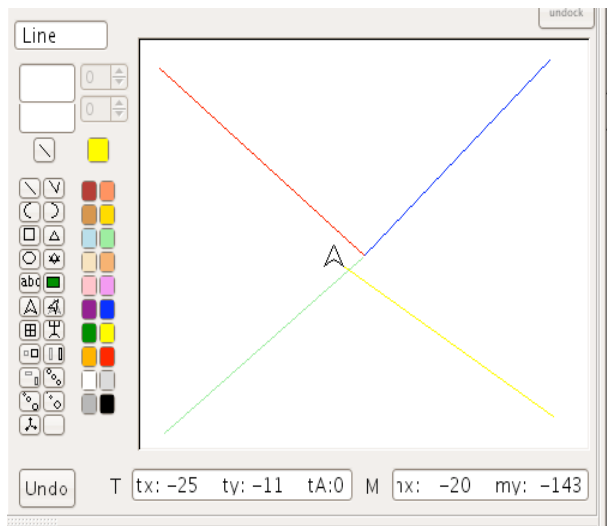
Purpose: To create a spiral of life and modify it.

Overview: A spiral of life is a picture that changes with time. You start by creating some colorful lines. This basic design is called the seed. Next, turn the lines many times around a central point. The last two steps will repeat that process while turning the design. In addition, the design will become bigger and bigger. The result is enchanting. Each time it will be different.

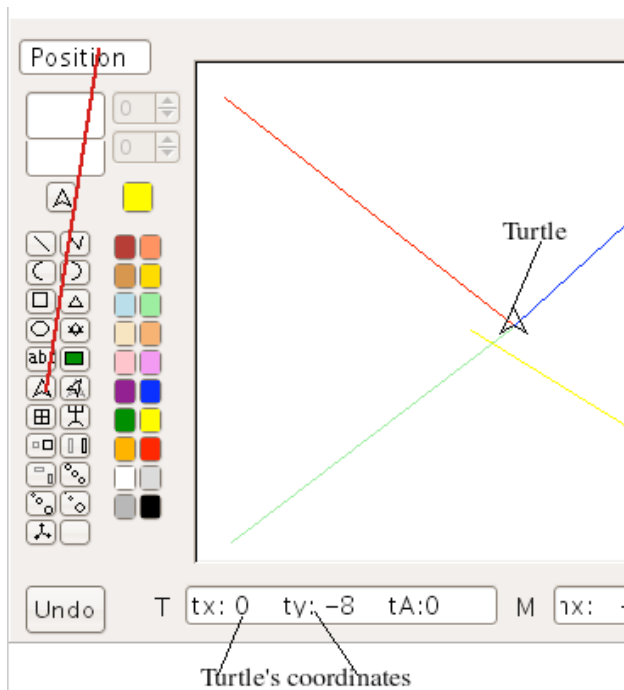
Step 1: Make sure that the line tool is selected. Then, select a color.



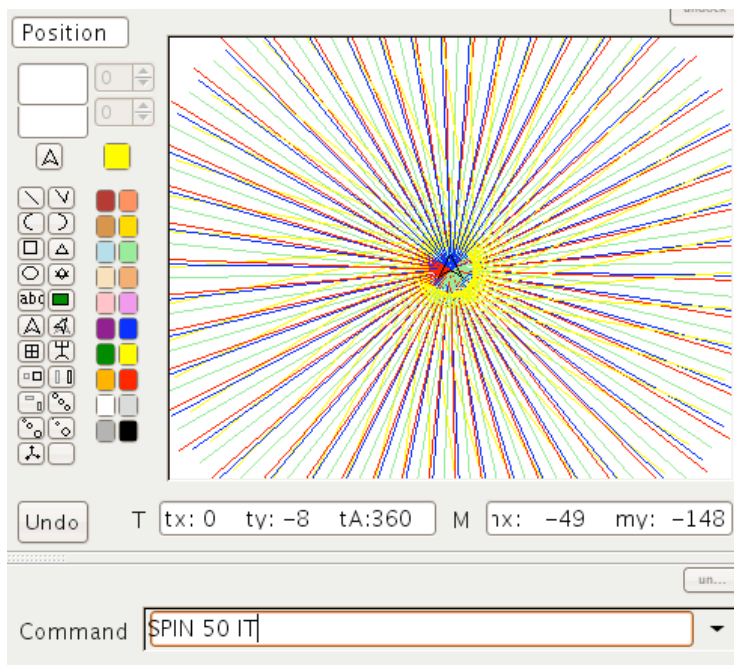
Step 2: Make a line as you drag the mouse toward the center of the screen. Continue to make lines, selecting a new color each time.



Step 3: Choose the Position Tool and place the mouse near the center of the screen. Left click and notice that the turtle moves to that point. The coordinates tx and ty should not both be exactly 0 but still be close.

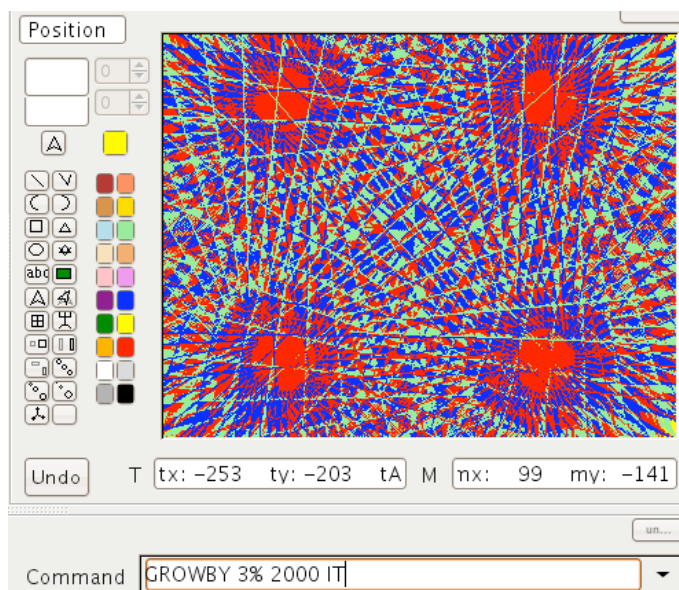


Step 4: In the command line, type SPIN 50 IT (instead of 50 you can type any other number). SPIN IT will rotate the picture that number of times:



Step 5: In the command line, type ROT 90 IT (instead of 90, of course, you can type any other number).

Step 6: In the command line, type GROWBY 3% 2000 IT (the number can be different). The result will be a picture that is dynamically changing. To stop the drawing, press the <esc> key. To redraw, press F9.



Program Analysis:

The program that created the picture is displayed in the editor and looks something like the following. Yours will look slightly different depending on the choice of colors and numbers.

```
GROWBY 3% 2000 [ROT 90 [SPIN 50 [  
JT -170 126 COLOR red [MT 0 -8]  
JT 155 132 COLOR blue [MT 1 -8]  
JT -166 -136 COLOR lightgreen [MT -1 -10]  
JT 158 -124 COLOR yellow [MT -25 -11]  
JT 0 -8]]]
```

Each line in the program reflects something you did when actually building the program. You can analyze your own program or paste the above program into the SeeLogo editor as follows:

- Type in the command line "NEW example" <enter>.
- Copy the text above.
- Click in the editor and paste the example.

Here are some things to explore:

- The choice of colors is reflected by the color names. Change the names to names of other colors and press F9 to see the effect.
- One line in the program has the instruction ROT 90 (in English this means “Rotate the picture by 90 degrees”). Change the number 90 to another number, and examine the effect by pressing F9 each time. After playing with it for a while, you may discover a pattern.
- Another line in the program is SPIN 60 (which means to turn the picture around the turtle 60 times). Change the number 60 to another number, and observe the effect.
- Another line in the program has the words "GROWBY 3% 2000" (the number 3% means the rate to increase the size of the picture, and the number 2000 means how many times to repeat the drawing). Change the number 2000 to a small number like 2 or 10 or to a larger number like 100, and press F9 each time. Then change it back to 2000, and change the percentage 3% to 1%, 0% -1%, 10% and other percentages to get a feel for what the number means.
- Another line in the program is JT 0 -8 (change the two numbers to 0 0 and press F9). Notice the effect on the picture.

Activities

- Create a spiral of life.
- Change some numbers or words in the editor and click Redraw (or press F9). Notice the effect on the picture.
- Change the two numbers that come after the last JT command to 0 0 (JT 0 0) and click Redraw. The picture you see is called "The Crystal of Life."
- Press the Undock button. The Undock button is on the right above the drawing screen. To bring it back to the small screen press Dock.

Project: Make a beautiful Spiral of Life.

Creating an Art Gallery (Naming Pictures)

Purpose: In this activity, you will learn to create an art gallery based on individual pictures.

Overview: In a single SeeLogo session, you type NEW Name1 <enter> to make a picture. Keep doing it (NEW Name 2 <enter> make a picture etc.). When you are finished making the picture, type New Show1 (or New Exhibit1 or any other name) and type the name you defined with the commands WAIT 2 JT 0 0 BG WHITE in between.

Example:

Type NEW P1 <enter> and make a spiral of life.

NEW P2 <enter> and make another spiral of life.

Keep doing this 3 or 4 more times and type NEW Exhibit <enter>

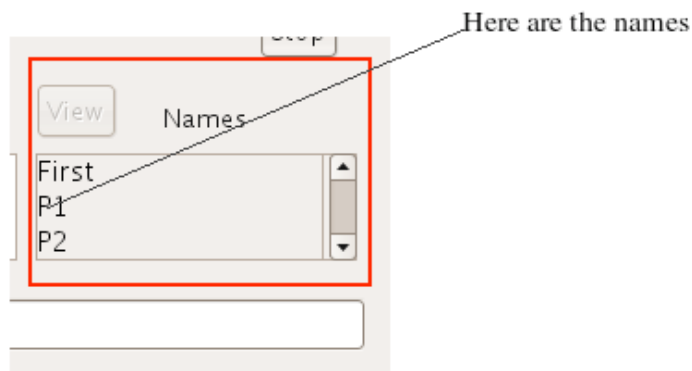
P1 WAIT 2 JT 0 0 BG WHITE

P2 WAIT 2 JT 0 0 BG WHITE

continue...

Explanation:

WAIT 2 means to wait 2 seconds and the number can change. JT 0 0 ensures that the next picture will start from the center. BG WHITE creates a white background. You can also experiment with other colors.

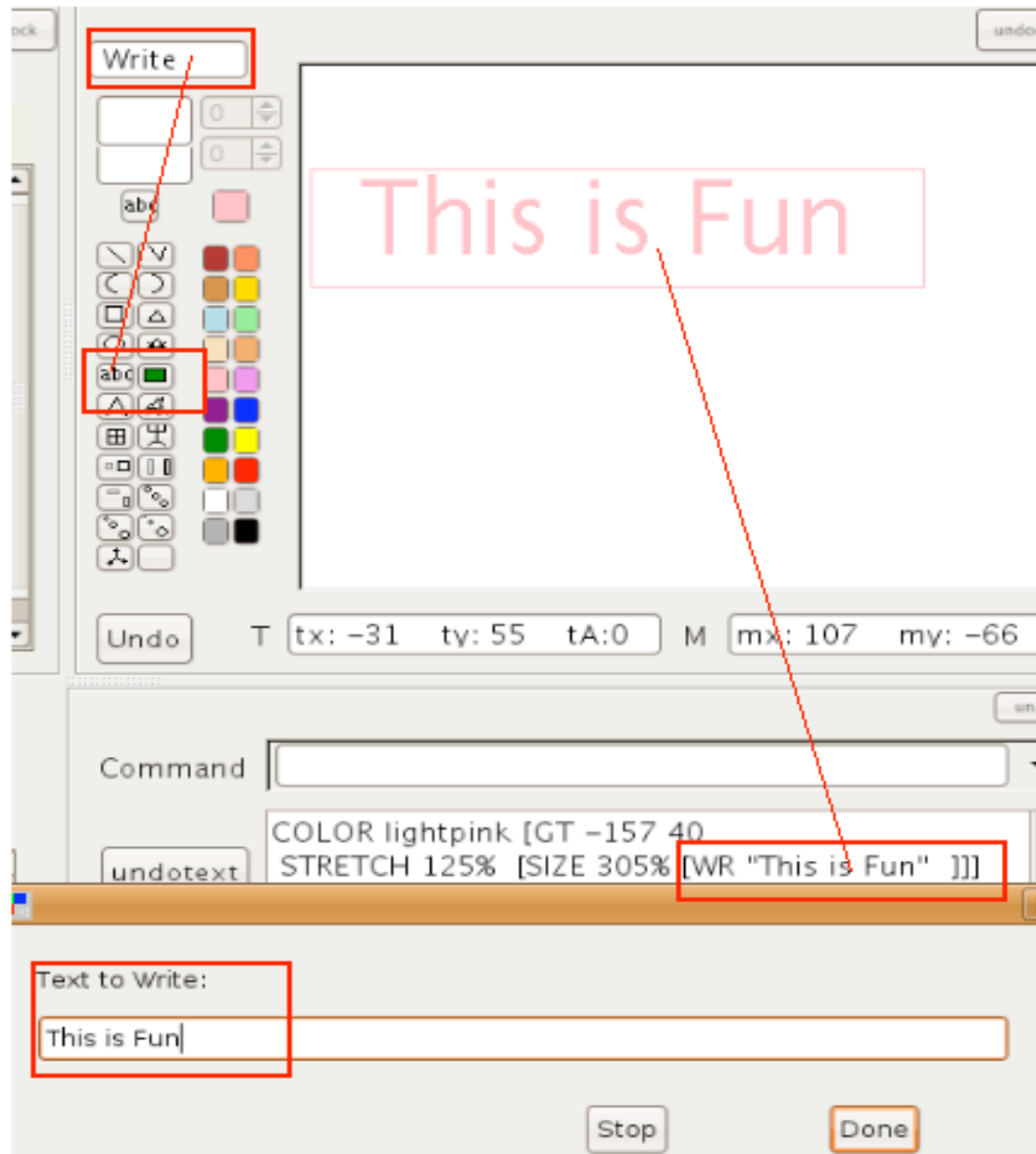


Project: Make an exhibit that includes at least 5 different names.

Writing on the Screen

Purpose: To find different ways to write text on the screen and create art.

Overview and Example: You can use the mouse tool bar to write text on the screen. The program is entered into the editor where it can be changed to create different effects. This picture explains what to do:



Explanation: In this example, the content of the editor reflected what the user did with the mouse:

```
COLOR lightpink [GT -150 100 STRETCH 125% [SIZE 150% [WR "This is Fun"]]]
```

Every word and all the numbers in the editor have meaning. You can experiment by changing numbers and words and pressing F9 to see the effect. The word “lightpink” defines the color of the text. The numbers after “GT” define the coordinates where the text is placed. The word “GT” is an abbreviation of “Go To.” The number after the word “STRETCH” defines the ratio of width to height, and the number after “SIZE” defines how big the letters are.

Activity:

- ☒ Use the “write tool” to write something on the screen.
- ☒ Change a number in the editor and click Redraw. Notice how the picture changed. Repeat this with different numbers.
- ☒ Write a sentence and then double the size by doubling the numbers after “Size.” Experiment.

- ☒ Write a sentence. Then, change the location by changing the two numbers after “GT.” These numbers are called the x and y coordinates.
- ☒ Change the color of the sentence by changing the name of the color in the editor and click Redraw.
- ☒ Change the number after the word “Stretch” and notice the effect.
- ☒ Change the sentence inside the quotes (after the word “WR”) and Redraw.

Project:

- Write a sentence on the screen.
- Copy the contents of the editor once and paste it 5 times so that there will be five lines in the editor.
- For each line, change the second number after the word “GT” to other numbers so that the pictures will appear under each other.
- For each line, change the numbers that come after “Size” to make the sentences smaller and smaller in an even way. For example:

This is Fun

This is Fun

This is Fun

This is Fun

This is Fun

Tiling in SeeLogo

Purpose: To create two-dimensional patterns using the TILE command, for repetitive patterns are beautiful!

Overview: Once you see a picture on the screen, you can type TILE IT and the whole screen will be filled with the same repetitive pattern, much like wall paper. The computer computes the dimensions of the picture and translates it to both horizontal and vertical directions to fill the screen.

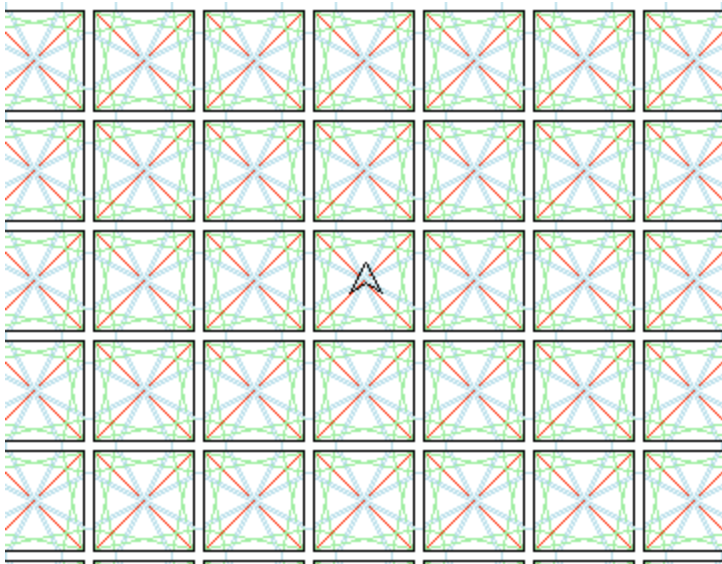
Examples:

- ☒ TILE [BOX 50 50] just fills the screen with a square tile.
- ☒ TILE [BOX 50 50 SOLID BLUE [CIRCLE 40]] tiles the screen with square tiles with circles in the middle.
- ☒ TILE [SPIN 3 [BOX 50 50]] creates an interesting design based on a rotated box.

A More Extensive Example:

- BOX 50 50 <enter>
- Use the line tool to make some colored lines inside the box that touch the boundary.
- JT 0 0 <enter>
- MIRROR IT <enter>
- SPIN 4 IT <enter>
- TILE IT <enter>

The results are most beautiful!



Creating a Mandala

Purpose: To learn to create a circular design known as a Mandala.

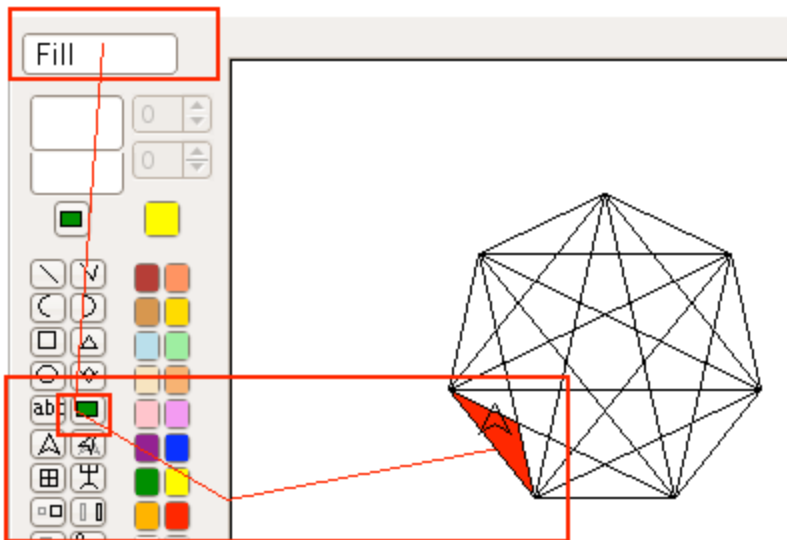
Overview: A mandala can be thought of as a design that reflects infinity in the finite world. It contains an infinite number of copies of itself with unending variations. The SeeLogo program can be used to create special types of mandalas that are called *self similar*. The technique goes like this: You create a picture (named, say, A). Then, use it together with other commands to make another picture (say, B). Then you use B to create another picture named C and so on.

Example: Type New A and create a picture (with or without the mouse). For example:
STARSK 1 7 STARSK 2 7 STARSK 3 7 <enter>

You can make the picture bigger by applying the SIZE command. Type SIZE 2 IT <enter>, and you will see:

SIZE 2 [STARSK 1 7 STARSK 2 7 STARSK 3 7]

You can also use the FILL tool to paint different areas of the picture with colors.

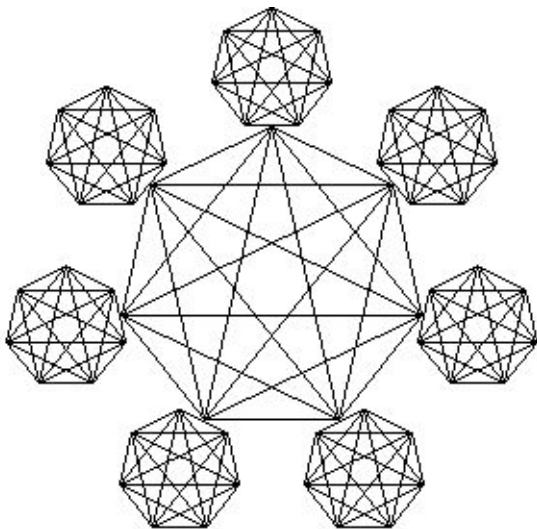


Once you are satisfied, type:

- NEW B <enter>
- JT 0 110 SIZE 0.4 A <enter>
- JT 0 0 <enter>
- SPIN 7 IT <enter>
- A <enter>

In the editor, you will see the SeeLogo program, which you can adjust by changing the numbers.

```
SPIN 7 [JT 0 110 SIZE 0.4 A JT 0 0 ]
```



Explanation: When creating a mandala, you use the NEW command several times. Once you have created the whole thing, you can adjust the different parts by clicking on the name and playing

with the program that is in the editor (or making additional adjustments using the mouse).

Further Exploration: You can create a slide show that shows how the Mandala was built as follows. Assume that the names that were used to build the design were:

A, B, C etc.:

Type NEW show

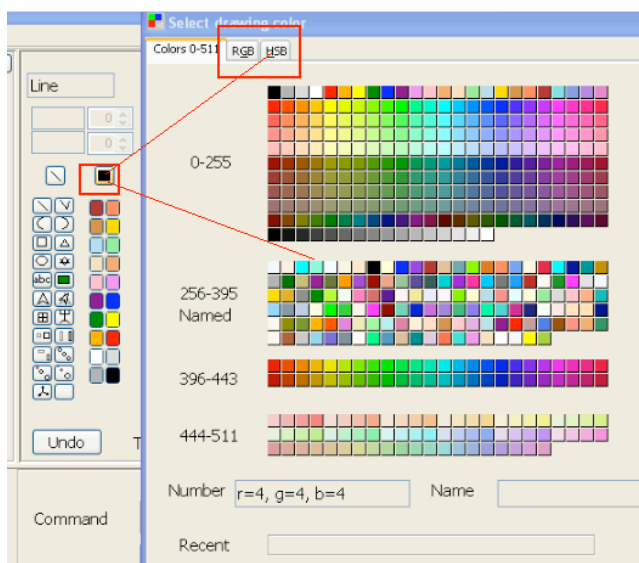
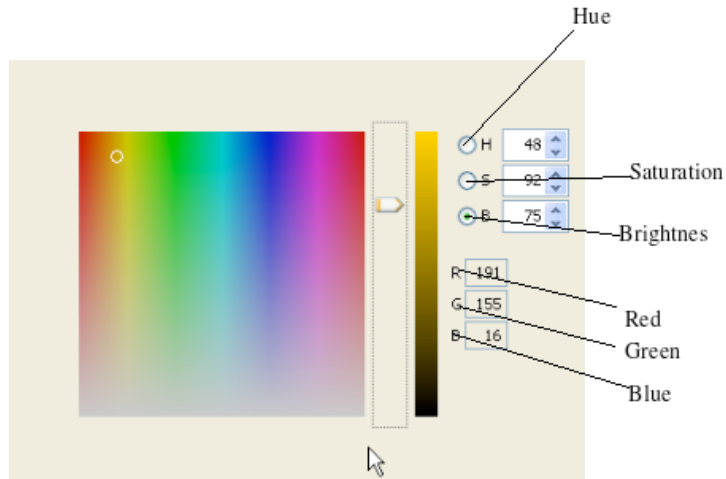
A JT 0 0 WAIT 3 BG white

B JT 0 0 WAIT 3 BG white etc...

Project: Make a beautiful Mandala.

Playing with Colors (HSB)

Purpose: To learn some features of SeeLogo involving colors. The picture below explains how to activate the color pallet. Once there, you can select one of the first 511 colors or play with the HSB tool to make up your own colors.

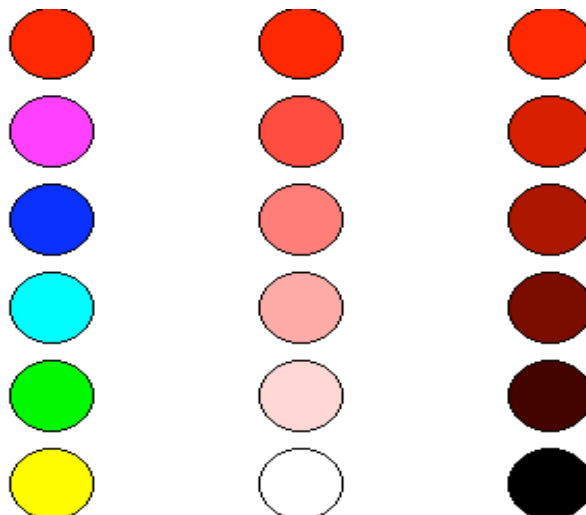


Overview: Some of the commands in SeeLogo that use colors are COLOR and SOLID. For example, SOLID RED [CIRCLE 50] makes a circle filled with red, while COLOR RED [CIRCLE 50] only paints the boundary of the circle with the color. The HSB command allows the user to define any possible color that is available on the computer. For example, the command: SOLID HSB 0 100 100 [CIRCLE 200] creates a red circle. In order to understand that concept, you must understand the HSB command. The first number (0 in our case) defines the hue. This is similar to the place of the color on a rainbow. Furthermore, the number can go from 0 to 360, which is again 0. The second number is the saturation and it goes from 0 to 100 (that tells how much "color" there is). The smaller the number, the closer to gray it is. The third number defines the brightness of the color (0 is black and 100 is fully bright).

Activity: Paste the following program in SeeLogo and study it:

```
JT 0 125  
SOLID HSB 360 100 100 [CIRCLE 40] GOBK 50  
SOLID HSB 300 100 100 [CIRCLE 40] GOBK 50  
SOLID HSB 240 100 100 [CIRCLE 40] GOBK 50  
SOLID HSB 180 100 100 [CIRCLE 40] GOBK 50  
SOLID HSB 120 100 100 [CIRCLE 40] GOBK 50  
SOLID HSB 60 100 100 [CIRCLE 40] GOBK 50
```

In this program the only thing that changed was the hue. To understand the color concept more and get a better feel for it, change the program so that the saturation will change and then write another program that changes the brightness.



Hue

Saturation

Brightness

Section III: Drawing Simple Pictures

A picture is composed of one or more basic shapes. A basic shape is either a closed shape, such as a polygon or an ellipse, or it is an open shape, such as a line (a segment of a straight line) or an arc (a segment of a circle).

This section contains three units. In the first unit, you will learn how to use SeeLogo to draw basic closed shapes, such as a BOX, a CIRCLE or a STAR. In the second unit, you will learn how to draw lines and to control the state of the turtle, its location and direction. In the third unit, you will learn how to draw ARCs and CURVEs, which are two different ways of drawing segments of circles.

Each unit comes with SeeLogo exercises. Here are the links to the Math Art SeeLogo exercise files.

For the first unit, on Drawing Closed Shapes, the link is:

http://www.ithaca.edu/seelogo/Math_Art/shapes.sl

For the second unit, on Drawing Lines, there are two links:

http://www.ithaca.edu/seelogo/Math_Art/lines1.sl

http://www.ithaca.edu/seelogo/Math_Art/lines2.sl

For the third unit, on Drawing Arcs and Curves, this is the link:

http://www.ithaca.edu/seelogo/Math_Art/curves.sl

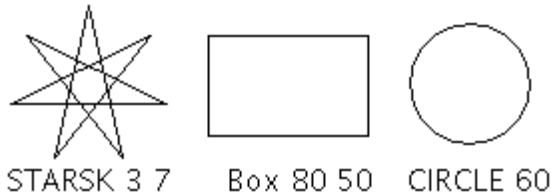
Here is how to download a SeeLogo Math Art exercise file.

1. Copy one of the above links.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

Drawing Closed Shapes

Rectangles, Circles, Polygons and Stars

In the first unit, you will learn how to draw basic closed shapes, such as BOXes, CIRCLES and STARS. Let us begin with a few examples.



Explanation: The BOX 80 50 command draws a box of 80 pixels width and 50 pixels height. The CIRCLE 60 command draws a circle of diameter 60. The STARSK 3 7 command draws seven points evenly spaced around a circle first and then connects every third point. All closed shapes are centered on the present location of the turtle.

Commands Used to Draw Closed Shapes

The SeeLogo language includes a several commands that draw closed shapes centered on the turtle's present location. The simplest closed shape is a CIRCLE. The CIRCLE command takes one parameter, one number to specify the diameter of the circle in pixels. Thus, the command CIRCLE 60 draws a circle of diameter 60 pixels.

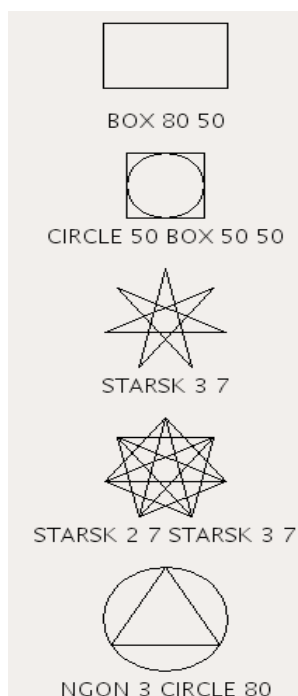
The second closed shape command, named BOX, draws a rectangle the center of which is the location of the turtle. The BOX command takes two parameters, two numbers that specify the dimensions of the rectangle. Thus, if the turtle is pointing up or north, as it does after the JT command, the command BOX 80 50 creates a rectangle of width 80 pixels and height 50 pixels. Now, observe how the direction in which the turtle points affects the drawing. The commands RT 90 BOX 80 50 draw a rectangle that is the same as that drawn by BOX 50 80, with the two parameters transposed. In fact, the BOX command draws a rectangle along axes that are perpendicular and parallel to the direction that the turtle points. The first parameter is the dimension perpendicular to that direction; the second parameter is the dimension parallel to that direction.

The SeeLogo command that draws regular polygons is called NGON. By “regular” polygon, we mean that all its sides are of equal length and all its internal angles are equal. Thus, the command NGON 3 draws an equilateral triangle, NGON 4 draws a square, and so on. The single parameter or number that comes after the command NGON specifies the number of sides of the polygon. Thus, NGON 6 creates a hexagon with six equal sides and six equal angles. The size of the polygon drawn is such that it fits perfectly in a circle of diameter 80 pixels.

Of the SeeLogo commands that draw closed shapes, the last to discuss here is the STARSK command. This command creates shapes that are related to polygons, but differ, because the lines of a STARSK shape may intersect. The STARSK command takes two parameters, two numerical inputs. The second number tells how many vertices or points the shape should have and the first number tells how to connect the vertices with edges or lines. Let us say, for example, that the second parameter, the “points” number, is 8. If the first parameter, the “skip” number, is 0, then just 8 points are drawn, but they are so small you may miss them. If the skip number is 1, a regular octagon is drawn. If the skip number is 2,

two braided squares are drawn. What happens when the skip number is set to 3? To 4? To 5?

Here are some examples of the use of these SeeLogo closed shape commands. Try them out for yourself. Be sure to vary the values of the numerical parameters to understand them better.



In order to work the following exercise, you will need to use the JT command, which tells the turtle to Jump To a specific location specified by its Cartesian coordinates. For example, to draw a circle of radius 40 centered at the point whose coordinates are $x = 30$ and $y = 50$, type the commands JT 30 50 CIRCLE 40.

Exercise

There is one SeeLogo exercise file for this unit, to which the link is:

http://www.ithaca.edu/seelogo/Math_Art/shapes.sl

Here is how to download a SeeLogo Math Art exercise file.

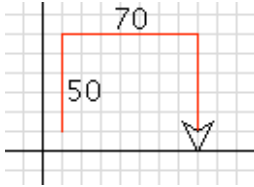
1. Copy the above link.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

Each exercise consists of several questions. Each question includes a red drawing. The goal in each question is to turn the red drawing from red to black by drawing over it using SeeLogo commands. In some questions, the final position of the turtle is shown in red, so you will have to turn it black, too. In many cases, the last command in your answer will be JT 0 0. Each exercise begins with question q1, to which you provide answer a1. When you are done with that, click on answer a2 and provide the answer to question q2, on so on. Many of the questions have more than one correct answer. Note that when you first download an exercise, if the drawing on the screen is not correct, just press the Redraw button.

Complete this exercise, before moving on to the next unit on Drawing Lines.

Drawing Lines

In this unit, you will learn how to draw straight lines and to control the state of the turtle, its location and direction. Let us begin with a couple of examples.



```
JT 10 10  
FD 50 RT 90 FD 70 RT 90 FD 50
```

Explanation: The command JT 10 10 tells the turtle to Jump To the location $x = 10$ and $y = 10$, without drawing a line, and turns it to point in the upward or north direction. The command FD 50 tells the turtle to go ForwarD 50 pixels. The command RT 90 tells it to change direction and turn 90 degrees to the right.



```
JT 20 30 MT 80 10
```

Explanation: You already know what the JT 20 30 command does. The MT 80 10 command tells the turtle to move to the specified coordinates, while drawing a line. Note that the MT command does not draw in the direction that the turtle is pointing and that it does not change the direction the turtle is pointing: the turtle is still pointing north.

Commands Used to Draw Lines

The commands used to draw lines are FD, BK, RT, LT, JT, MT, GO, GOFD and GOBK.

Part 1: Drawing Lines using FD, GOFD, BK, GOBK, RT and LT

The command FD (ForwarD) tells the turtle to move a specific number of steps (pixels) in the direction that it is pointing, drawing a line as it moves. The command RT (Right Turn) tells it to turn to the right a specific number of degrees. The commands BK (Back) and LT (Left Turn) cause the turtle to move or turn in the opposite direction. If you prepend the word GO before the word FD or BK, thus, GOFD or GOBK, the turtle moves, but does not draw a line.

Here again is the link to the lines1.sl SeeLogo exercise file to use for this part of this unit.
http://www.ithaca.edu/seelogo/Math_Art/lines1.sl

Activities

In each activity, you are creating a SeeLogo picture program. You may type the program line by line in the command window, pressing the Enter key after each line. Or you may type the program in the editor, clicking the Redraw button to see the result. Start each activity by giving it a NEW name. Here, we call the programs a1, a2, and so on. You can give the programs names of your own choice, but it is best to keep the names in alphanumeric order.

Activity 1

In the command window, type and enter:

NEW a1

In the command window or in the editor, type and enter: FD 50 RT 90

Copy and paste this line three more times, so that there will be a total of four such lines in the editor. Press Redraw to see the resulting 50 x 50 square.

Change two numbers in the program, so that it draws this 100 x 50 rectangle.



Repeat this activity, until you have fully mastered it.

Activity 2

In the command window, type and enter:

NEW a2

In the command window or in the editor, type and enter:

RT 90

FD 20 GOFD 10

Copy and paste the last line three more times, so that there are four such lines in the editor.

Press Redraw to see the resulting dashed line.



Activity 3

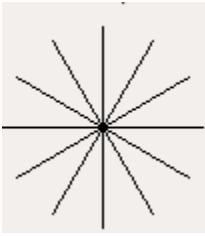
In the command window, type and enter:

NEW a3

In the command window or in the editor, type and enter:

FD 70 BK 70 RT 30

Copy and paste this line several times, until, when you click on the Redraw button, you see the following picture.



Experiment with this program. Can you draw a clock face with just five equally spaced rays?

Exercise

There is one SeeLogo exercise file for this part of this unit, to which the link is:

http://www.ithaca.edu/seelogo/Math_Art/lines1.sl

Here is how to download a SeeLogo Math Art exercise file.

1. Copy the above link.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

It is probably best for you to complete the lines1.sl SeeLogo exercise first, before moving on to the second part of this unit on drawing lines.

Part 2: Drawing Lines Using Commands MT and JT

The commands JT and MT are followed by two parameters or numbers that tell the turtle where to go next. The first number, called the x coordinate, specifies horizontal location, relative to the center of the screen. Similarly, the second number, called the y coordinate, specifies the vertical location.

There are two differences that distinguish the commands: First, the MT (Move To) command causes the turtle to draw a line to the new point, but the JT (Jump To) command does not draw a line. Second, the JT command causes the turtle to point north, but the MT command does not change the direction in which the turtle is pointing.

Here again is the link to the lines2.sl SeeLogo exercise file to use for this part of this unit.

http://www.ithaca.edu/seelogo/Math_Art/lines2.sl

3.

Activity 4

In the command window, type and enter:

NEW a4

In the command widow or in the editor, type and enter:

CIRCLE 20 JT 20 0

CIRCLE 20 JT 40 0

Keep typing commands, or copying, pasting and editing, until you have drawn five circles in a row, like this.



Activity 5

In the command window, type and enter:

NEW a5

In the command widow or in the editor, type and enter:

JT 50 70

MT 100 70

MT 100 20

Keep typing commands, until you have drawn a square. What is the length of its sides?

Remark: There is another command that works similarly to MT called MV. The only difference is that the coordinates are relative to the current position of the turtle and NOT to the center of the screen. The command GOMV is the same as MV, except that no lines are drawn. Try to make a square or a rectangle using the MV command to get a better feel for it.

Exercise

There is one SeeLogo exercise file for this part of this unit, to which the link is:

http://www.ithaca.edu/seelogo/Math_Art/lines2.sl

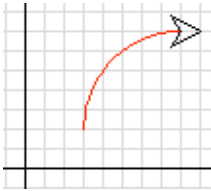
Here is how to download a SeeLogo Math Art exercise file.

1. Copy the above link.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

Complete this exercise, before moving on to the next unit on Drawing Arcs and Curves.

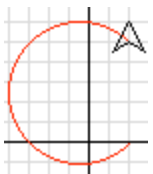
Drawing Arcs and Curves

In SeeLogo, there are two different ways to draw a segment of a circle, by using either an ARC command or a CURVE command. In this unit, you will learn how to draw ARCs and CURVES. Here are examples of each type.



```
JT 30 20 ARCR 90 100
```

Explanation: The command JT 30 20 places the turtle at the location $x = 30$ and $y = 20$ and turns it to point in the upward or north direction. The command ARCR 90 100 tells the turtle to move, while drawing, along a circle 100 pixels in diameter, while turning gradually to the right by 90 degrees in total, one-quarter of a full turn. Note that, after the ARC is completed, the turtle is pointing 90 degrees to the right of north.



```
JT 20 0 CURVEL 270 50
```

Explanation: You already know the effects of the command JT 20 0: it puts the turtle at the specified location and sets its direction to point north. The command CURVEL 270 50 tells the turtle to draw an arc totaling 270 degrees, while moving forward 50 pixels, along what is called the chord of the curve. The CURVEL command draws the curve to the left of the chord. Note that, after the CURVE is completed, the turtle is still pointing north.

ARC and CURVE Commands

There are four commands for drawing segments of circles: ARCR (ARC to the Right) and CURVER (CURVE to the Right) and their symmetrical twins ARCL and CURVEL that exchange left for right.

The ARCR and ARCL commands require two parameters, two numbers that come after the command. The first parameter is an angle, the number of degrees of the circle to draw. The second is the diameter of the circle in pixels. An ARC traces the turtle's path. The turtle's direction changes as it traces out an ARC. The turtle always points in a direction tangential to the ARC. The initial segment of an ARC points in the same the direction that the turtle initially points. The turtle finally points in the same direction as the final segment of the ARC. ARC commands are useful for creating spiraling shapes.

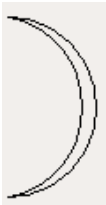
The CURVER and CURVEL commands also require two parameters. The first parameter is an angle, the number of degrees of the circle to draw. The second parameter is the length of the chord of the CURVE in pixels, where the chord of a CURVE is the straight line from its initial point to its final point. The direction of the turtle does not change as it draws a CURVE. The turtle points in the same direction after drawing a CURVE as it did before. The path of the turtle is a straight line, the chord of the CURVE, which is drawn to either the right side (CURVER) or the left side (CURVEL) of the chord. This command is useful for creating shapes like flower petals or bananas.

Activities

Each activity is a SeeLogo program, which you may type line by line in the command window or in the editor. Start each activity by giving it a NEW name. Here, we call the programs E1, E2, and so on.

Activity 1

```
NEW E1
CURVER 180 100
JT 0 0
CURVER 160 100
```



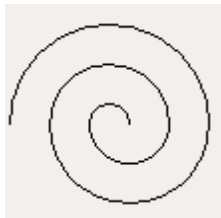
Continue to experiment with this program, until you understand what the two parameters of CURVER mean. For example, in the editor, change the number 100 after the first CURVER to 70, so the line reads CURVER 180 70. Can you explain what happened and why?



Activity 2

```
NEW E2
ARCR 180 100
ARCR 180 80
ARCR 180 60
```

Keep on typing until you complete the following spiral.



Activity 3

NEW E3

ARCR 120 50 ARCL 120 50 RT 90

Copy and paste the same line three more times, to get four identical lines in all.

Look at the shape and recognize that it is a kind of a "curved square."



Change the number 90 to another number, so that the picture will become a curved triangle. This time you will need only three identical lines instead of four. What is the number that you used instead of 90?



Exercise

There is one SeeLogo exercise file for this unit, to which the link is:

http://www.ithaca.edu/seelogo/Math_Art/curves.sl

Here is how to download a SeeLogo Math Art exercise file.

1. Copy the above link.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

Complete this exercise, before moving on to the next Section on Transforming Pictures.

Section IV: Transforming Pictures

One creative aspect of SeeLogo is the capacity to create new shapes by transforming old ones. But be warned: you may have so much fun that it can become addictive. Essentially, there are two types of transformations. Transformations of the first type transform a shape just once or twice: SIZE, FLIP, TRANS, GLIDERF and MIRROR. Those of the second type transform a shape a specified number of times: SPIN, SPINR, SPINL, GROWBY, GROWGBY and REPEAT.

Now, it is important to note that the state of the turtle affects the way a picture is transformed. While it may not be obvious, when we see a picture on the screen, that the state of the turtle is actually a part of the picture. But we have already met this concept, when we drew shapes. For example, the command BOX 70 20 draws a level rectangle but the command sequence RT 45 BOX 70 20 draws a tilted rectangle. The same principle applies, when we use transformation operators.

Let us start by examining the REPEAT command. This command takes two parameters or inputs, the first a number and the second a picture. For example, what results from the command REPEAT 3 [CIRCLE 30] is a single circle, because the three identical circles are superimposed, drawn right on top of one another, because the turtle did not move between drawings. However, the command REPEAT 3 [CIRCLE 30 JT 30 0] will draw three circles side by side, because the turtle moved 30 pixels horizontally between drawings.



Here is another way to accomplish this task by creating and calling a separate SeeLogo picture program. One SeeLogo picture program can use other picture programs as building blocks.

Start a new SeeLogo picture program, p1

. NEW p1

Create the program p1.

CIRCLE 30

JT 30 0

Start another SeeLogo picture program, p2.

NEW p2

Create the program p2.

REPEAT 3 p1

Note that the second program, p2, calls the first program, p1.

We will refer to the first picture, drawn here by the program p1, as the “seed picture” and the second picture, drawn by the program p2, as the “transformed seed picture” or simply the “transformed picture.” Note that if you had defined the picture program p1 as “CIRCLE 30 JT 0 30”, instead of “... 30 0”, the circles would be stacked vertically.

This example demonstrates how the REPEAT command is used. In a succinct form, we say that the syntax of the REPEAT command is REPEAT number [name]. Here, “number” is the number of copies to be drawn and “name” is either the name of a predefined picture program, which may be enclosed in square brackets, or a list of commands, which must be enclosed in square brackets. The program p2 is the same as REPEAT 3 [CIRCLE 30 JT 30 0], which one could type directly into the editor.

The following is a table of the syntax of all the transformation commands or operators.

TILE [name]

MIRROR [name]

DOUBLE [name]

SIZE number [name]

REPEAT number [name]

Note: Another command called LOOP works very similarly to REPEAT, but, since it is used only in programming, will be introduced later.

SPIN number [name]

SPINR number [name]

SPINL number [name]

ROT number [name]

COLOR number [name]

SOLID number [name]

GROWBY number1 number2 [name]

GROWGBY number1 number2 [name]

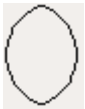
GLIDERE number1 number2 [name]

TRANS number1 number2 [name]

Here are some examples of the use of operators.

NEW p1

MIRROR [CURVER 120 50]



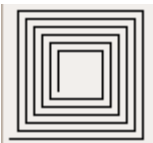
NEW p2

SPIN 8 [ARCR 180 40 JT 0 0]



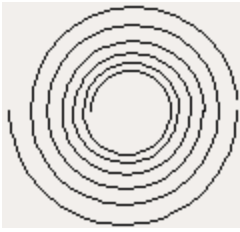
NEW p3

GROWBY 10% 24 [FD 20 RT 90]



NEW p4

GROWGBY 10% 12 [ARCR 180 40]



NEW p5
SPINR 3 [ARCR 120 40 ARCL 120 40]



Once an operator is applied to a picture, other operators, including the original one, can be re-applied. Beautiful pictures will emerge! For example, try this.

SPIN 5 [p1 JT 0 0]
MIRROR [p2]
MIRROR [p4]
SPIN 6 [p5]

Exercises

Here is a list of the units in this section with the links to the exercises for each unit.

REPEAT Command, Part 1

http://www.ithaca.edu/seelogo/Math_Art/repeat1.sl

REPEAT Command, Part 2

http://www.ithaca.edu/seelogo/Math_Art/repeat2.sl

SPIN Command http://www.ithaca.edu/seelogo/Math_Art/spin.sl

SPINR and SPINL Commands http://www.ithaca.edu/seelogo/Math_Art/spinrl.sl

MIRROR Command http://www.ithaca.edu/seelogo/Math_Art/mirror.sl

GROWBY Command, Part 1 http://www.ithaca.edu/seelogo/Math_Art/growby.sl

GROWBY Command, Part 2

http://www.ithaca.edu/seelogo/Math_Art/growby2.sl

Transformation Commands and Combinations

http://www.ithaca.edu/seelogo/Math_Art/mixed1.sl

http://www.ithaca.edu/seelogo/Math_Art/mixed2.sl

http://www.ithaca.edu/seelogo/Math_Art/mixed_level2.sl

Here is how to download a SeeLogo Math Art exercise file.

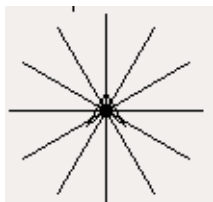
1. Copy one of the above links.
2. In the SeeLogo program, go to File > Open from web.
3. Paste in the link and click the Done button or press the Enter key.

REPEAT Command

The main purpose of this unit is to learn how to use the REPEAT command, obviously. But even more important is to

understand how the turtle's position after drawing the seed picture affects the drawing of the transformed picture. In the exercises, the small blue circle indicates the turtle's starting point and the small arrowhead, glider or dart shape indicates its ending point and direction. This lesson is very important, for many other lessons will follow the same principle.

Of the syntax of the REPEAT command, recall that after you type the word REPEAT, you must type its two parameters. The first is a number, the number of copies of the seed picture to draw. The second parameter is either 1) a list of commands enclosed in square brackets or 2) the name of a seed picture program, which may be enclosed in square brackets, or 3) the word IT, but only if you type it in the command line. For example, there are at least three different ways to draw this picture in SeeLogo.



Method 1: One Fell Swoop. After naming the new program, type the full program in one line in the command window.

NEW Way1

REPEAT 12 [FD 50 BK 50 RT 30]

Method 2: Call a separate seed picture program. First, name the new seed picture program whatever name you want and type only the seed program into it.

NEW OneRay

FD 50 BK 50 RT 30

Then, name the new program that will transform the seed picture. Finally, call the seed program from it. You can actually omit the square brackets in this case.

NEW Way2

REPEAT 12 [OneRay]

Method 3: Use the IT word. This method works only if you type everything in the command line.

NEW Way3:

FD 50 BK 50 RT 30

REPEAT 12 IT

Exercises

There are two SeeLogo exercise files for this unit, to which the links are:

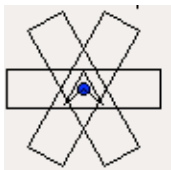
http://www.ithaca.edu/seelogo/Math_Art/repeat1.sl

http://www.ithaca.edu/seelogo/Math_Art/repeat2.sl

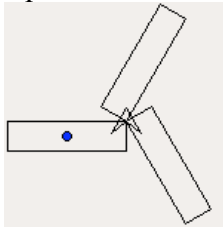
Complete both these exercises, before moving on to the next unit, on the SPIN command.

SPIN Command

The SPIN command allows you to turn a picture around a point. We shall refer to the picture which is rotated as the seed picture and to the point around which the seed picture is rotated as the turning point. The turning point is the turtle's ending point, after it draws the seed picture. Recall that the turtle's ending point is indicated by the small arrowhead, glider or dart shape, while its starting point is indicated by the small blue dot. For example, the command SPIN 3 [BOX 80 20] turns a rectangle around its center, because the turtle's ending point, as well as its starting point, is there.



On the other hand, the command SPIN 3 [BOX 80 20 JT 40 10] turns the rectangle around its top right corner, because this is where turtle ended up, upon completion of the seed picture program, enclosed in square brackets.



The syntax of the SPIN command is the same as that of the REPEAT command. After the command SPIN, the first parameter you must specify is a number, the number of copies of the seed picture to

array evenly about the turning point. The second parameter to specify is the seed picture, as a list of commands in square brackets or the name of a predefined seed picture program or the word IT.

Exercise

There is one SeeLogo exercise file for this unit, to which the link is:

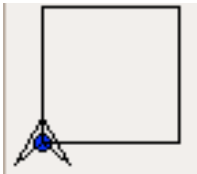
http://www.ithaca.edu/seelogo/Math_Art/spin.sl

Complete this exercise, before moving on to the next unit, on the SPINR and SPINL commands.

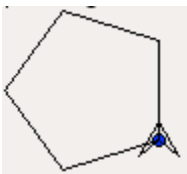
SPINR and SPINL Commands

In this unit, you will learn to use the SPINR and SPINL commands. The SPINR and SPINL commands are similar to the SPIN command, but with one significant difference. With SPIN, there is a single turning point for all copies, which is the turtle's ending point for the seed picture. But with SPINR and SPINL, there is a different the turning point for each copy, which is the turtle's ending point for the preceding copy, if the ending point differs from the starting point. With SPINR, the copies rotate to the right and with SPINL, obviously, they rotate to the left. By how much do they rotate? Well, if N is the first parameter, the number of copies to draw, each copy rotates by $360/N$ degrees from the preceding copy, so that drawing the N copies completes a full revolution, 360 degrees.

Now, for some examples. The command SPINR 4 [FD 50] makes a square turning to the right or clockwise.

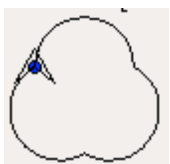


The command SPINL 5 [FD 50] makes a pentagon turning to the left or counterclockwise.

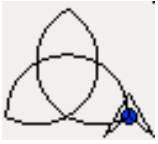


Using these commands with arcs creates beautiful results.

SPINR 3 [ARCR 180 50]



SPINR 3 [ARCL 180 50]



Note that if there is no difference between the ending point and the starting point, then there is no difference between the commands SPIN, SPINR and SPINL. For example, you get the same picture, a rectangle turning about its center, whether you issue the command SPIN 3 [BOX 80 20] or SPINR 3 [BOX 80 20] or SPINL 3 [BOX 80 20], because the turtle's ending point is the same as its starting point, when it draws a BOX.

There is one SeeLogo exercise file for this unit, to which the link is

http://www.ithaca.edu/seelogo/Math_Art/spinrl.sl

Complete this exercise, before moving on to the next unit, on the MIRROR command.

MIRROR Command

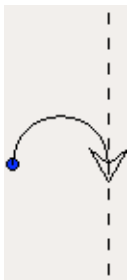
In this unit, you will learn how to use the MIRROR command.

The MIRROR command causes the seed picture to be reflected across a line, the mirror line or reflection line, which is defined by the ending position and direction of the turtle. The mirror line passes through the turtle's final location and is parallel to its final direction. Recall that the turtle's ending position and direction are indicated by the small arrowhead, glider or dart shape.

The following examples demonstrate the use of the MIRROR command. The location of the mirror line is indicated by a dashed line.

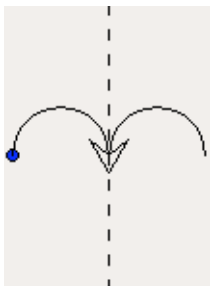
Draw a simple picture with different starting and ending points, say, an ARC.

ARCR 180 50



To reflect this picture across the dashed line using Method 3, type in the command line

MIRROR IT



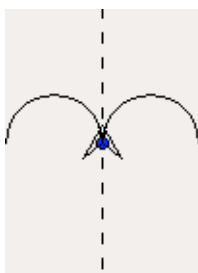
You could achieve the same result using Method 1 by issuing the command `MIRROR [ARCR 180 50]`. You could also use Method 2. Explain how you could do that.

If you change the final position or direction of the turtle before applying the mirror command, the result will be different. Check out these examples.

Change the final direction. `MIRROR [ARCR 180 50 LT 45]`



Change the final position.
`MIRROR [ARCR 180 50 JT 0 0]`



Change both the final position and final direction.
`MIRROR [ARCR 180 50 JT 25 25 RT 90]`



If the mirror line is a line of reflection symmetry of the seed picture, the `MIRROR` command does not change the appearance of the picture.

`MIRROR [ARCR 180 50 JT 25 25]`



Exercise

There is one SeeLogo exercise file for this unit, to which the link is

http://www.ithaca.edu/seelogo/Math_Art/mirror.sl

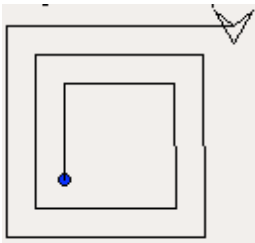
Complete this exercise, before moving on to the next unit, on the `GROWBY` command.

GROWBY Command, Part 1

This is the first of two units on the GROWBY command.

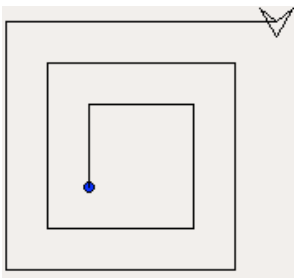
The GROWBY command is similar to the REPEAT command, except that the sizes of the copies of the seed picture change, increasing or decreasing by the same amount, with each drawing of a copy. If the size of a copy of a picture differs from the size of the picture, we shall say that it is a scaled copy. If the change in size from copy to copy is by the same amount, we shall say that the scaling or the growth is linear. The GROWBY command generates linear growth. The copies that the GROWBY command generates are scaled linearly.

For example, to draw a kind of square spiral, issue the command
GROWBY 15% 10 [FD 50 RT 90]



Change the growth rate from 15% to 25% and you get the same kind of spiral, except that the distance between the lines is greater.

GROWBY 25% 10 [FD 50 RT 90]



The syntax of the GROWBY command is a little more complicated than that of the REPEAT command. The GROWBY command requires, not two, but three parameters, the first two of which are numbers and the third, a picture program. The first number specifies the growth rate, for example, 15% or 0.15. The second number specifies how many scaled copies of the seed picture to draw, for example, 10 or 20. The third parameter specifies the seed picture, either as a list of commands, which must be enclosed in square brackets, or as the name of a predefined picture program, which may or may not be

enclosed in square brackets, or the word IT.

The following example is similar to the first one except, that there are 20 lines instead of 10.
GROWBY 15% 20 [FD 50 RT 90]

Exercise

There is one SeeLogo exercise file for this unit, to which the link is

http://www.ithaca.edu/seelogo/Math_Art/growby.sl

Complete this exercise, before moving on to the next unit, also on the GROWBY command.

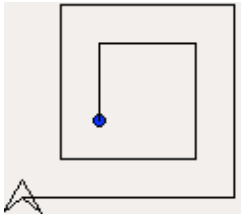
GROWBY Command, Part 2

In this unit, you will learn more about the mathematics of linear growth, which the GROWBY command generates.

An easy way to explain the GROWBY command is by giving an example. The command GROWBY 10% 5 [CIRCLE 50] draws five circles: CIRCLE 50 CIRCLE 55 CIRCLE 60 CIRCLE 65 CIRCLE 70. Note that the diameter of the circles increases by 5 pixels from one copy to the next. How did GROWBY calculate this amount? The growth increment, the amount by which each copy is larger than the preceding copy, is given by the product of the growth rate, 10%, and the diameter of the first circle, 50 pixels. Thus, the growth increment is $10\% * 50 = 0.10 * 50 = 5$ (pixels). So the second circle's diameter is 55 pixels, which is 5 pixels or 10% more than the first circle's, because $50 + 5 = 55$ or $(1.00 + 0.10) * 50 = 1.1 * 50 = 55$. The diameters of the other circles grow by the same amount, but not by the same percent. When a quantity grows by the same amount each time, its growth is described as uniform growth or linear growth or growth at a constant rate. The GROWBY command generates growth at a constant rate, linear growth or uniform growth.

In the same way, GROWBY 20% 4 [FD 50 RT 90] generates the sequence FD 50 RT 90 FD 60 RT 90 FD 70 RT 90 FD 80 RT 90. Notice that the angles stay the same at 90 degrees, but that the lengths grow linearly or uniformly, the growth increment being $20\% * 50 \text{ pixels} = 10 \text{ pixels}$.

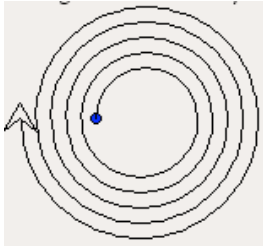
For example, the lengths of the lines in the picture GROWBY 25% 8 [FD 40 RT 90] are 40, 50, 60, 70, 80, 90, 100, 110, because the growth increment is $25\% * 40 = 10$.



On the other hand, the diameters of the CIRCLES in the drawing GROWBY 50% 7 [CIRCLE 30] are 30, 45, 60, 75, 90, 105, 120, because the growth increment is $50\% * 30 = 15$.



The command GROWBY 15% 10 [ARCR 180 60] generates 10 semicircular arcs. The first semicircle has a diameter of 60. Because the growth increment is $15\% * 60 = 9$, the second semicircle has a diameter of $60 + 9 = 69$. The diameter of each succeeding semicircle is larger than that of its predecessor by 9 pixels. The diameter of the tenth semicircle is $60 + (10 - 1) * 9 = 141$ (pixels).



Exercise

There is one SeeLogo exercise file for this unit, to which the link is

http://www.ithaca.edu/seelogo/Math_Art/growby2.sl

Complete this exercise, before moving on to the next unit, which includes three mixed exercises.

Transformation Commands and Combinations

This last unit in this section contains three exercises. In the first two exercises, mixed1.sl and mixed2.sl, the solutions to all questions require you to use one of the SeeLogo transformation commands that you have learned so far. In the third exercise, mixed_level2.sl, the solutions all require at least two such commands.

Exercises

There are three SeeLogo exercise files for this unit, to which the links are

http://www.ithaca.edu/seelogo/Math_Art/mixed1.sl

http://www.ithaca.edu/seelogo/Math_Art/mixed2.sl

http://www.ithaca.edu/seelogo/Math_Art/mixed_level2.sl

Complete these exercises, before moving on to the next section on SeeLogo programming.

Section V: SeeLogo Programming and Dynamic Art

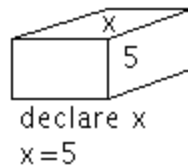
This section is more sophisticated than the previous sections and introduces programming concepts. It will help you to develop your own art projects later in the course.

Introduction to SeeLogo Programming

Let us start with a SeeLogo experiment: If you type `FD 50`, you will see a line drawn but if you type something like `FD x` the computer will object, for the computer "does not know" what `x` is. But if you type `DECLARE x` in the command line you and then type `FD x`, the computer will actually do something but you will not see anything new on the screen because we did not specify what `x` is.

Now if you type `x=5` then `x` actually becomes 5.

In computer terminology, `x` is called a "variable". It can be thought of as a small box that has a label. The label is called `x`. The command `x=5` puts the number 5 inside the box with label `x`. We say that the value of the variable `x` is 5.



If we type `FD x`, a line of length 5 will be drawn. The value of `x` can be now used in any SeeLogo command just as any number would. For example `CIRCLE x` will draw a circle of diameter `x`. The command `Starsk 2 x` will draw a 5 pointed star etc.

So there are three stages when using variables:

1. Creating a variable: the command `DECLARE x` creates a variable named `x`. The command `DECLARE giraffe` creates a variable named `giraffe` etc. Declaring a variable is analogous to creating a box inside the brain of the computer and labeling it with a chosen name.
2. Defining the value of the variable: `x=8` defines `x` as 8 and it is like putting the number 8 in the box.
3. Using the variable: once a variable is defined, it can be used in any situation as if it were a number.

In addition to defining the initial value of a variable, you can change the value if it was already defined. For example, if you type `x = 10` and then type `x=x+7`, the value of `x` will be 17. This is true because "`x=x+7`" means "Take the variable `x` (10 in this case) and add 7 to its value."

Let us stop here and ask four practice questions. Answers are on the following page:

1. Imagine you typed
 `declare elephant`
 `elephant = 3000`
 `elephant = elephant + 300`
 What is the value of the variable `elephant`?

2. Imagine you typed

```
declare x y t
```

```
t=10
```

```
x=100 + 2*t
```

```
y = 2*x
```

What is the value of the variable y?

3. Assume that $x=6$ $y=x/2$. What is the value of y?

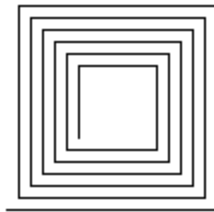
4. Assume that $x=3$ $x=x*2$ $y=x^2$. What is the value of y?

Variables can be used to draw pictures. For example, examine the program:

```
NEW A
```

```
x=60
```

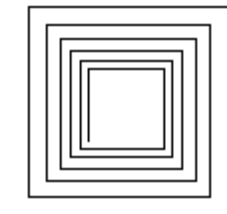
```
Loop 24 [FD x RT 90 x = x +5] creates a square spiral picture:
```



The sequence of numbers that corresponds to this picture is 60,65,70,75... The difference between a number and the one before it is the same (5 in this case). This type of sequence is called an Arithmetic Sequence.

The command "loop" mean is similar to the command "Repeat." The program A starts by defining the value of x to 5 and then using it to make a line with a right turn. Next, the value becomes 10 and the whole process is repeated 24 times: "Move Forward x and turn Right. Add 5 to x and do it again and again 24 times...."

Remark: This particular example is analogous to the GROWBY command. If instead of adding 5 to the value of x ($x=x+5$), we multiply it by a fixed amount (5% for example), we get a geometrical spiral. The program will become:



```
New B
```

```
x=60 loop 24 [FD x RT 90 x = x * 1.05]
```

The sequence of numbers that corresponds to this picture is 60, $60*1.05$, $60*1.05^2$,... The quotient between a number and the one before it is the same (1.05 in the case). This type of sequence is called a

Geometric Sequence.

Answers to practice questions:

1. elephant = 3300 (since $3000 + 300 = 3300$)
2. 240 (since $x=100 + 20 = 120$ and $y = 2*x = 2*120 = 240$)
3. 3 (since $6/2 = 3$)
4. 36 (since $x=2*3 = 6$ and $y = 6^2=6*6 = 36$)

More practice questions:

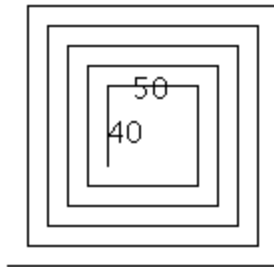
For each of the following commands, draw the picture you think you will see. After you see the answer, go to SeeLogo, type the commands and compare.

```
new A
x=10 y = x*x
JT 50 0 BOX y y
```

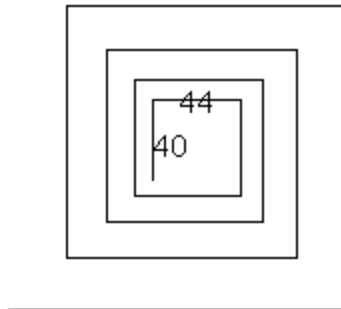
```
New B
x=20
JT x/2 0 circle x
JT (-x/2) 0 circle x
```

Exercises:

1. Write a program using variables that draw the following picture. The picture represents an arithmetic sequence.



2. Write a program using variables that draw the following picture. The picture represents a geometric sequence.



3. What is the value of x and y after the program has actually happened (in computer terminology we say "after the program is executed")?

A. $x=0$ loop 5 [$x=x+1$] $y=x*x$ $x=$ _____ $y=$ _____

B. $x=150$ loop 8 [FD x RT 90 $x = x - 10$]

C. $x=0$ repeat 100 [$x=x+0.01$ $y = x + 3$]

Functions and Time in SeeLogo

One way to create dynamic art in SeeLogo is by using functions. A function can be seen as a rule or distilled pattern. The functions we will define and use have just one variable that we will call t and interpret as time. We will also always let the time range from 0 to 1. This means that when $t=0$, the animation starts and when $t=1$, the animation ends. To achieve this in SeeLogo, we will use the following kind of programming construct:

$t=0$

loop 1000 [$t=t+0.001$ do something (using t)]

If we want to slow down the animation 10 fold, we will simply increase the loop to 10000.

loop 10000 [$t= t + 1/10000$ do something]

The variable t is called a parameter. The "do something" part can take many forms. For example, we can have two dots move from the side of the screen to the center:

Do something can then be something like

$x=100 - 100*t$ JT x 0 circle x

or we can spin this one line to make it into 8 lines moving towards the center using the spin command and making sure the center of spin is the center of the screen.

Here is the whole program (assuming you start SeeLogo from scratch):

declare t x

$t = 0$

loop 10000 [$t = t+1/10000$ $x=200 - 200*t$ spin 8 [JT x 0 circle 3 JT 0 0]]

Depending on your background, this may be difficult or easy but it should be motivating because the program we just wrote is so short and does not use sophisticated mathematics or programming. All you need to know is how spin works. Once you understand this program, try to reproduce it completely from scratch. You can change some numbers or even the way the program behaves.

Remark: Another way to define variables is by using the command "local" instead of "declare." Typing "local x y z" at the beginning of a program will define the variables x, y and z. It will only allow you to use them for that program. The advantage of using "local" is that it defines everything within the program itself. For example, if you start a fresh SeeLogo session and paste the program:

```
local x y t
t = 0
loop 10000 [t = t+1/10000
x=200 - 200*t y= 40 - 40*t spin 8 [JT x 0 circle y JT 0 0 ]]
```

Once you execute the program, you will see 8 circles converging to the origin, starting with diameter 40 and ending with diameter 0.

Exercises:

Modify the previous program so that 12 circles will be drawn. Each circle is of radius 5.

Modify the program above so that the circles will change diameter size from 20 to 10. You can accomplish this by using another variable called y. Circle y will then create a circle of diameter y, and all you need is to ensure that y starts with 20 and ends with 10 (while t ranges from 0 to 1)

Modify the program above.

Periodic Functions in SeeLogo

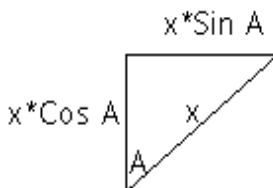
The Sin function:

One of the most useful functions in application of mathematical art are the Sin (and Cos) Functions.

If you paste the following program into SeeLogo,

```
LOCAL A x
x=100 A=50
RT A FD x BK x LT A
FD x*COS A RT 90 FD x*SIN A
```

you will see a picture like this:



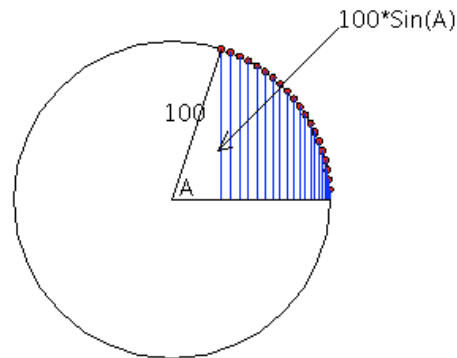
You can even modify the second line $x=100$ $A=50$ to make it random. To accomplish this, you replace that line with: $x= 80 + \text{rnd}$

70 $A=40 + \text{rnd } 30$ (this means that x ranges randomly between 80 to 150 and A ranges between 40 and

70). We suggest that you study the simple program above and play with it for some time until you develop a "feel" for the meaning of the Sin (and Cos) functions.

The useful thing about these functions (and from now on we will focus just on SIN) is that they are periodic with period 360. That means that $\text{Sin}(360 + x)$ is the same as $\text{Sin } x$.

The following program demonstrates the Sin function (the blue line is of length $100 \cdot \text{Sin}(A)$ while $A = 360 \cdot t$ where t ranges from 0 to 1. If we change A to $360 \cdot 2 \cdot t$, we get 2 periods, etc.



local t A y

circle 200

t=0

loop 100 [t=t+1/100 A=360*t

JT 100* Cos(A) 100* Sin(A)

solid red [circle 4] wait 0.1

color blue [BK 100 * Sin (A)]]

The answers to the following practice questions are at the end. We recommend that after you look at the answer, you go back and try to answer again without looking until you understand fully.

In all these questions and throughout the rest of this section, you may assume that the variable t (that means "time") ranges from 0 to 1.

1. Define a periodic function x with two cycles that ranges from -100 to 100.
2. Define a function y with 5 cycles that ranges from 0 to 400. Give four answers.
3. Make a program that changes the hue of a solid circle 7 times for all the possible ranges of hue. Make the program run slowly enough so that you can see the transition. Keep the saturation and

brightness at their maximum (100).

1. $x=100*\sin(360*2*t)$

2. A. $y = 400*(\sin(360*5*t))^2$
 B. $y= 200 + 200*\sin(360*5*t)$
 C. $y = 400*(\sin(360*5*t))^4$
 D. $y = 400*(\sin(360*5*t*t))^2$

3. local h t

```
t=0
loop 1000000 [t=t+1/1000000 h=360*(Sin(7*360*t))^2
solid hsb h 100 100 [circle 250]]
```

Exercises:

Remember that the domain for the independent variable t is [0,1].

1. Define a periodic function x which cycles 5 times and ranges from -150 to 150.
2. Define 3 different periodic functions that range from 0 to 100.
3. Write a program called A that moves a circle of diameter y from the point (-100,0) to the point (100,0). ranging from 0 to 40 in 4 cycles.
4. Modify the program you created in the previous question so that 8 circles will do the same thing symmetrically around the center of the screen (use spin).
5. Modify the program A so that the circles will move hence and force 3 cycles.
6. Modify the program A so that the brightness (in hsb) will cycle from minimum (0) to maximum (100) 7 times for the length of the animation.

Exploring Dynamic Art in SeeLogo

This is more difficult than the first part that we studied in the course and I will explain it in class or on the phone. If you understand this, you can use the ideas for the final project.

To get the SeeLogo program, do the following in SeeLogo:

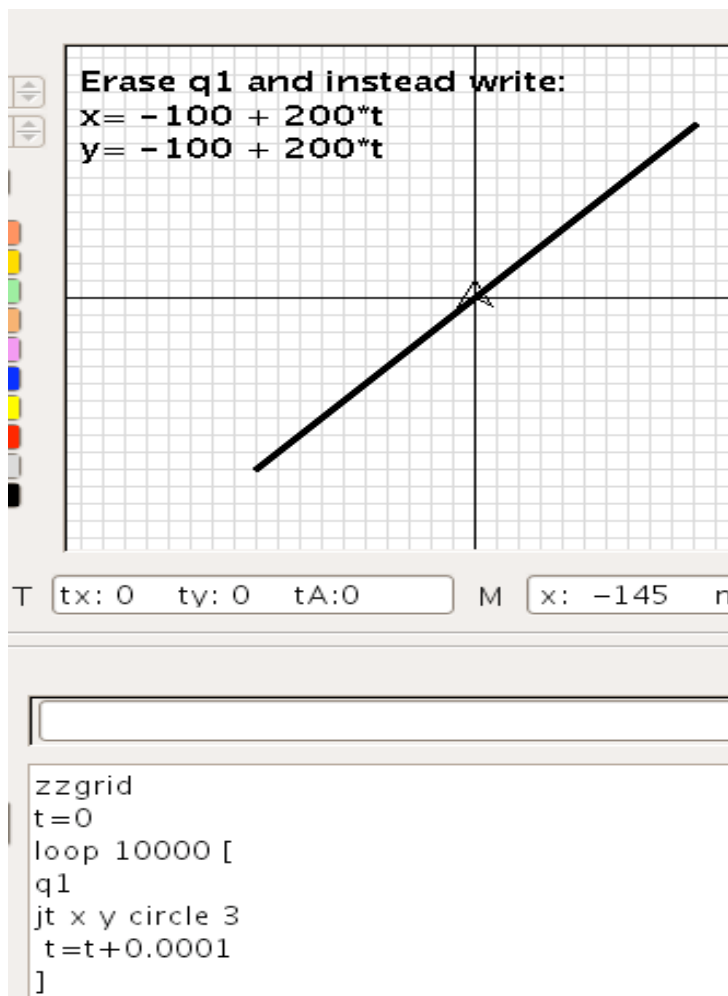
1. Copy the following link
2. In SeeLogo, go to File Open from Web
3. Paste the link and <enter>

http://www.ithaca.edu/seelogo/Math_Art/dynamic_art/template_v1.sl

Important comment: When you use the "-" sign to subtract, put a space after it (3-2 does not work but 3 - 2 does).

This part of the course is different than the first part and involves more programming and technical skills. It may be a good idea to share skills and learn from each other.

When you first load the training program, you will see this:



Our first task is to understand the connection between the picture and the program below. Let us do it step by step:

- ☒ The first command is zzgrid. This just makes the grid show on the screen and is not essential.
 - ☒ The second line is t=0. This means that t is a name of a variable and it gets the value 0.
- The next command is loop 10000 [something t=t+0.0001] (In this case something = q1 JT x y circle 3)

Explanation: we will deal with *something* in a moment, but t=t+0.0001 means in English "add 1/10000 to the number t" and because this is in a loop, 10000 mean the following in English: "Do the following 10000 times":

1. Do "something" (which actually means q1 JT x y circle 3)
2. Add 1/10000 to the value of t

Now this may take some time to absorb and learn, and it is okay if you do not yet understand it fully, but gradually you will understand more.

Let us do an experiment.

While SeeLogo is still open:

1. Copy the program in the editor.
2. Type NEW play1 and paste it there and hit F9 (Redraw). You should see the same picture without the words this time. Now put two slashes in front of zzgrid and you will see just the line without anything else. Then erase the two slashes so you can have a grid while you are learning.

The program is:

```
// zzgrid
t=0
loop 10000 [ q1 JT x y circle 3 t=t+0.0001 ]
3. Now replace the word q1 with the line and click F9
x=-100 + 200*t
y=x
```

Now it is time to think. What is happening here? Note that $0 \leq t \leq 1$ always. The number t (which we call time) is always between 0 and 1. Change the 10000 to 1000 and 0.0001 to 0.001 and you will see that the picture moves 10 times faster. Think why. Note that these two numbers are always reciprocal, second = 1/first. Change the line y=x to y=-x and see what happens. (Do you remember the function y=x?) Now try changing x to x=100 - 100*t y=x. Note that t if t=0 x=100 and if t=1 x=0

So now the program looks like:

```
//zzgrid
t=0
loop 10000 [
x=100 - 100*t
y=x
JT x y circle 3 JT 0 0
t=t+0.0001
]
```

Now, once you start understanding this and are getting a little knowledge and experience with

functions, you can make amazing pictures. Here are some examples:

You can change the speed of the line by changing 10000 --> 3000 and 0.0001--> 1/3000 etc.

By spinning the picture around (0,0) you get a powerful effect. Don't forget to type JT 0 0 at the end of the square brackets.

For example:

```
//zzgrid  
t=0  
loop 4000 [ spin 8 [ x= 100- 100*t  
y=x  
JT x y circle 3 JT 0 0] t=t+1/4000]
```

You can add other or

```
//zzgrid  
t=0  
loop 4000 [ mirror [ x= 100- 100*t  
y=x  
JT x y circle 3 JT 0 0] t=t+1/4000]
```

or you can change the circle to any other shape and change the colors. There are infinite possibilities and I am looking forward to seeing your work.

Remark: There is another version that has most of the answers plus many other examples for learning purposes.

To get this version, do the following in SeeLogo:

1. Copy the following link
2. In SeeLogo, go to File Open from Web
3. Paste the link and <enter>
http://www.ithaca.edu/seelogo/Math_Art/dynamic_art/template_v2.sl

Section VI: Quizzes

General Instructions

All of the following Quizzes follow the same format. In each Quiz, you are asked to create a set of from three to seven pictures using SeeLogo. Make your picture programs as simple as you can.

Put all your responses for each quiz in a separate SeeLogo project file. Begin each Quiz by starting a new SeeLogo project: click on the EraseAll button or select the EraseAll option of the File menu. Use the Save As option of the File menu to give each Quiz project file an appropriate file name, for example, Quiz01Shapes.sl. As you work, save your file regularly, and when you are done creating all your pictures, save the final version of your Quiz project file.

In each Quiz, give the pictures names in alphanumerical order. We suggest that you name the pictures B1, B2, B3, and so on. In the command window, type NEW B1 <Enter> to make space for a new picture named B1. Then, enter the commands to draw picture B1. When you are done drawing picture B1 and are ready to begin drawing picture B2, type NEW B2 <Enter>, and so on.

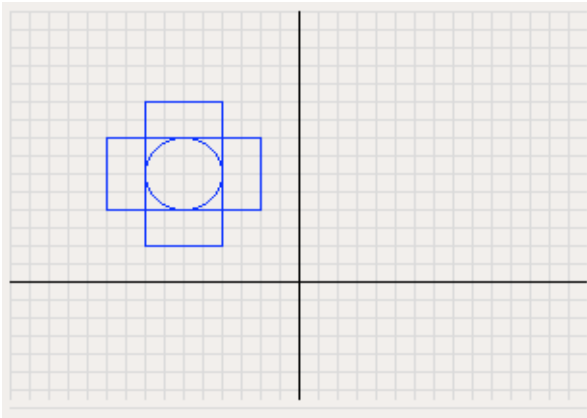
In many of the pictures, note the blue dot that indicates the center of the screen and the small arrowhead, glider or dart shape that indicates the final location and orientation of the turtle.

Closed Shapes Quiz

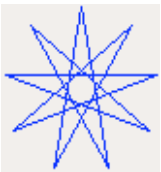
Instructions

Review the General Instructions for Quizzes.

Create the following three pictures in SeeLogo using the commands JT, BOX, CIRCLE and STARSK.



Picture B1



Picture B2



Picture B3

REPEAT Quiz 1

Instructions

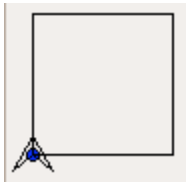
Review the General Instructions for Quizzes.

Create the following seven pictures in SeeLogo using the commands JT, FD, RT, LT and REPEAT.

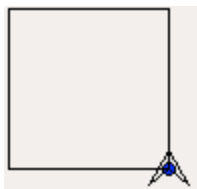
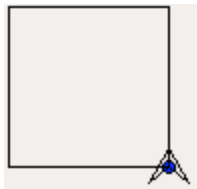


Picture B1:

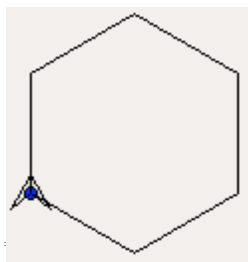
The length of each line is 50 pixels and the distance between lines is 20 pixels.



Picture B2: The length of each line is 70 pixels.



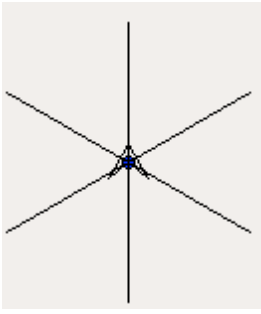
Picture B3: The length of each line is 80 pixels.



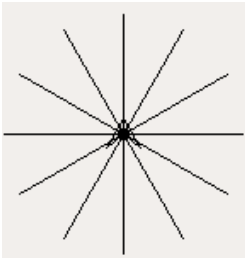
Picture B4: The length of each line is 60 pixels.

(Continued)

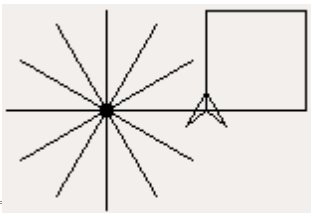




Picture B5: The length of each line is 70 pixels.



Picture B6: The length of each line is 70 pixels.



Picture B7: The length of each line is 50 pixels.

REPEAT Quiz 2

Instructions

Review the General Instructions for Quizzes.

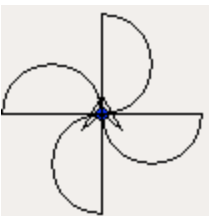
Create the following six pictures in SeeLogo using the commands JT, FD, BK, RT, LT, ARCR, ARCL, CURVER, CURVEL and REPEAT.



Picture B1

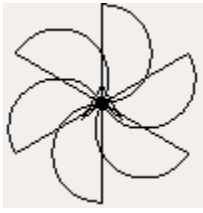


Picture B2: The length of each line is 40 pixels.



Picture B3: The length of each line is 50 pixels.

(Continued)



Picture B4: The length of each line is 50 pixels.



Picture B5: The length of each line is 70 pixels.

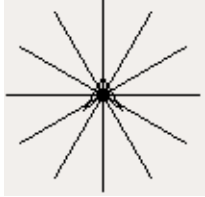


Picture B6

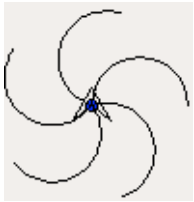
SPIN Quiz

Instructions

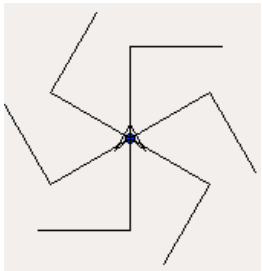
Review the General Instructions for Quizzes. Create the following six pictures in SeeLogo using the commands JT, FD, BK, RT, LT, ARCR, ARCL, BOX and SPIN.



Picture B1

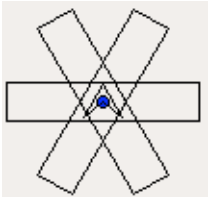


Picture B2

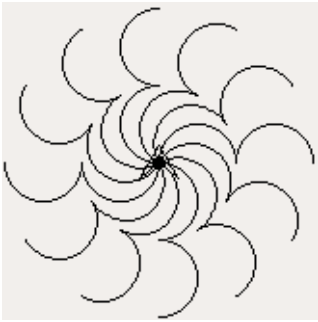


Picture B3

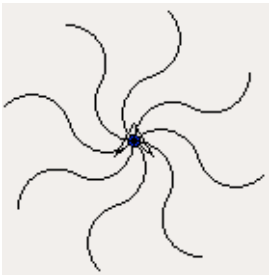
(Continued)



Picture B4: The ratio of width to height of the box is 5:1.



Picture B5



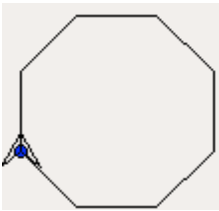
Picture B6

SPINRL Quiz

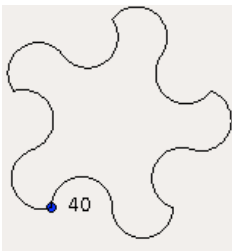
Instructions

Review the General Instructions for Quizzes.

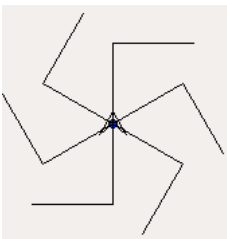
Create the following seven pictures in SeeLogo using the commands JT, FD, BK, RT, LT, ARCR, ARCL, SPINL, SPINR and SPIN.



Picture B1

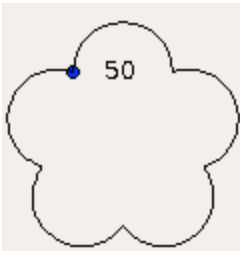


Picture B2

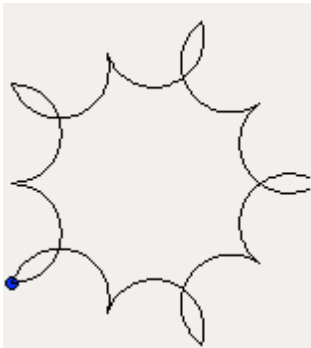


Picture B3

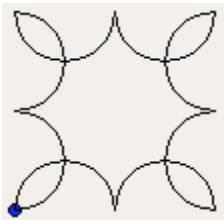
(Continued)



Picture B4



Picture B5



Picture B6



Picture B7

MIRROR Quiz

Instructions

Review the General Instructions for Quizzes.

Create the following six pictures in SeeLogo using the MIRROR command and any other commands you may need. In each picture, identify the mirror or reflection plane. Then, create half of the picture. Next, adjust the turtle's position and direction. Finally, apply the MIRROR command.



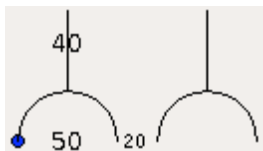
Picture B1:

The diameter of the circle is 50 pixels.



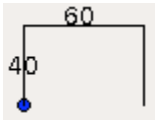
Picture B2:

The length of the line is also 50 pixels.

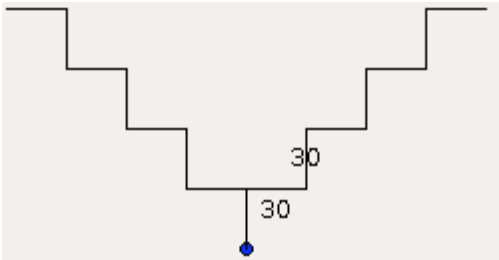


Picture B3

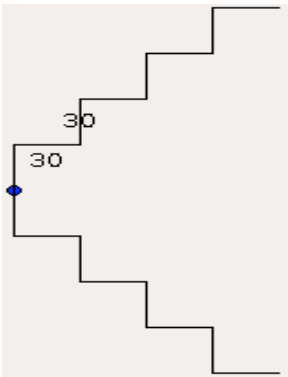
(Continued)



Picture B4



Picture B5



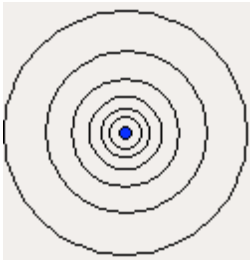
Picture B6

GROWBY and GROWGBY Quiz

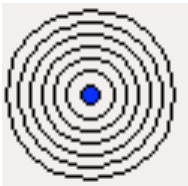
Instructions

Review the General Instructions for Quizzes.

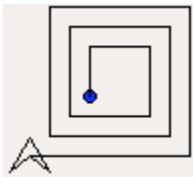
Create the following five pictures in SeeLogo using the either GROWBY or the GROWGBY command and any other commands you may need.



Picture B1: The diameters of the circles are 16, 24, and so on.

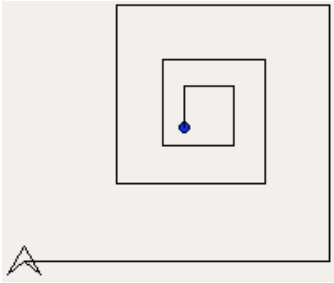


Picture B2: The diameters of the circles are 16, 24, and so on.

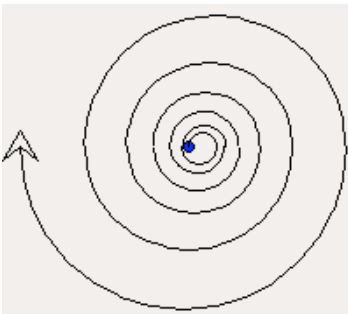


Picture B3: The lengths of the lines are 25, 30, and so on.

(Continued)



Picture B4: The lengths of the lines are 25, 30, and so on.



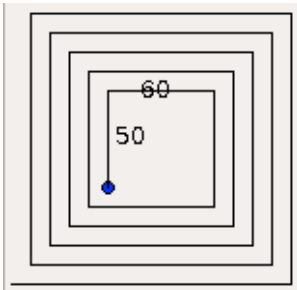
Picture B5: The diameters of the arcs are 16, 20, and so on.

Single Transformation Commands Quiz

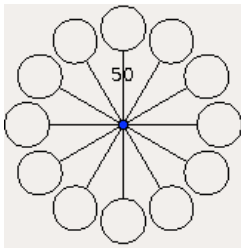
Instructions

Review the General Instructions for Quizzes.

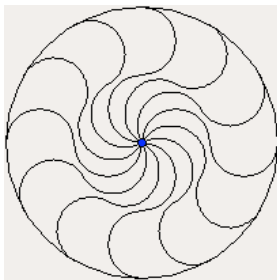
Create the following five pictures in SeeLogo, in each case using one of the transformation commands that you have learned so far.



Picture B1

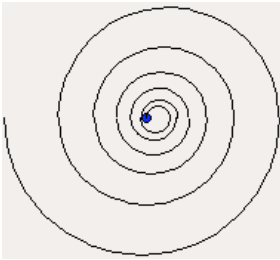


Picture B2: The diameter of the small circles is 30 pixels.

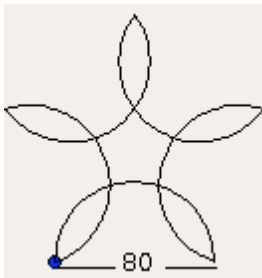


Picture B3: The diameter of the big circle is 200 pixels.

(Continued)



Picture B4: The diameters of the circular arcs is 16, 20, and so on.



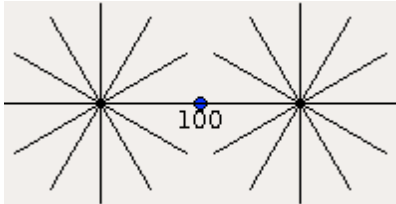
Picture B5

Multiple Transformation Commands Quiz

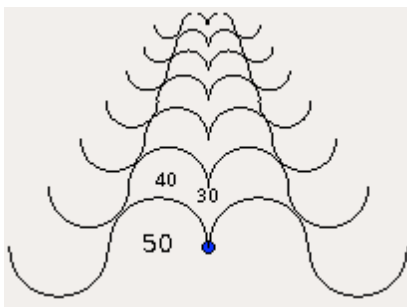
Instructions

Review the General Instructions for Quizzes.

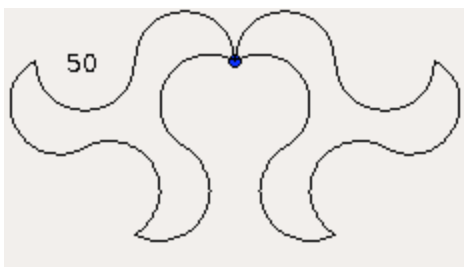
Create the following six pictures in SeeLogo, in each case using two or more of the transformation commands that you have learned so far.



Picture B1

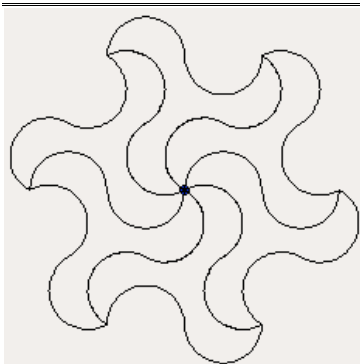
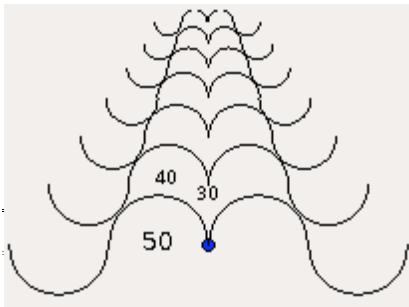


Picture B2

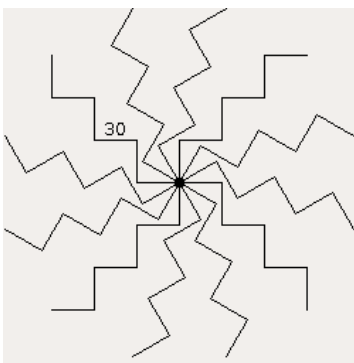


Picture B3

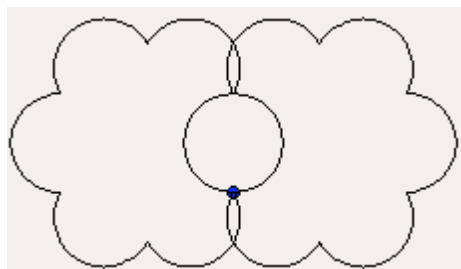
(Continued)



Picture B4



Picture B5



Picture B6

Section VII: Project Ideas

Here are some ideas of things you can do. We will be happy to explore ideas with you and to be your technical assistant if asked.

You can search the web and find a topic that is connected with symmetry and make an art show of that topic—either with still pictures or with pictures that change with time. Some good examples include: flags of different countries, wall paper designs (also Islamic art).

You can explore symmetry in nature either by actually observing it or by exploring the web. You can then use SeeLogo to make art pieces based on these designs.

You can create dynamic art shows based on the examples and ideas we explained in the last chapter. For example: You can have two (or any number) of spirals of life move towards each other and "explode in colors" in the middle of the screen. You can change sizes and colors of pictures while they move, etc. The parameters that you can control are location, size, and color. You can explore mathematical functions and see how they affect the design.

Here are some “seed examples.” Each seed example can turn into a unique and creative art project. The programs are actually very short and simple (after you give some thought to them). You can copy and paste them directly.

NEW A

```
local t x y //this makes a circle
t=0
loop 100000 [t=t+1/100000
x=100*cos(360*t) y = 100*sin(360*t)
JT x y circle 3]
```

NEW B

```
local t x y r //this makes a spiral
t=0
loop 100000 [t=t+1/100000 r=100*t
x=r*cos(360*t*5) y = r*sin(360*t*5)
JT x y circle 3]
```

NEW C

```
local t x y //this makes a very interesting shape
t=0
loop 100000 [t=t+1/100000
x=100*cos(360*t*3) y = 100*sin(360*t*5)
JT x y circle 3]
```

Note that the Cos cycles 3 times and the Sin cycles 5 times; instead of 3 and 5 you can test and play with infinitely many other variations. Each example above can be enriched and become an art piece once you start adding colors and symmetry. For example, look at the spinning addition (note that JT 0 0

was added. Study the example and try to reproduce it.

```
local t x y //note that very few things changed. Spin 4 was added and JT 0 0 was added.  
t=0  
loop 100000 [t=t+1/100000  
spin 4 [x=100*cos(360*t*3) y = 100*sin(360*t*5)  
JT x y circle 3 JT 0 0]]
```

Now, we can also make the sizes change, say, from 10 to 1 $s=20 - 19*t$
local t x y s//note that the function **s** was defined and used in **circle s**

```
t=0  
loop 100000 [t=t+1/100000 s= 20 - 19*t  
spin 4 [x=100*cos(360*t*3) y = 100*sin(360*t*5)  
JT x y circle s JT 0 0]]
```

Finally, you can change the hue saturation and brightness to anything you want, and you can play with the changes all at once.

```
local t x y s h1 s1 b1 //note that the function s was defined and used in circle s  
t=0  
loop 100000 [t=t+1/100000 s= 20 - 19*t  
spin 4 [x=100*cos(360*t*3) y = 100*sin(360*t*5) h1=360*sin(360*4*t)  
JT x y color hsb h1 100 100 [circle s] JT 0 0]]
```

Isn't this picture wonderful? It was all based on applications of mathematical thinking, and even though it may look complicated at first, it really is not. I hope you will give it a chance.