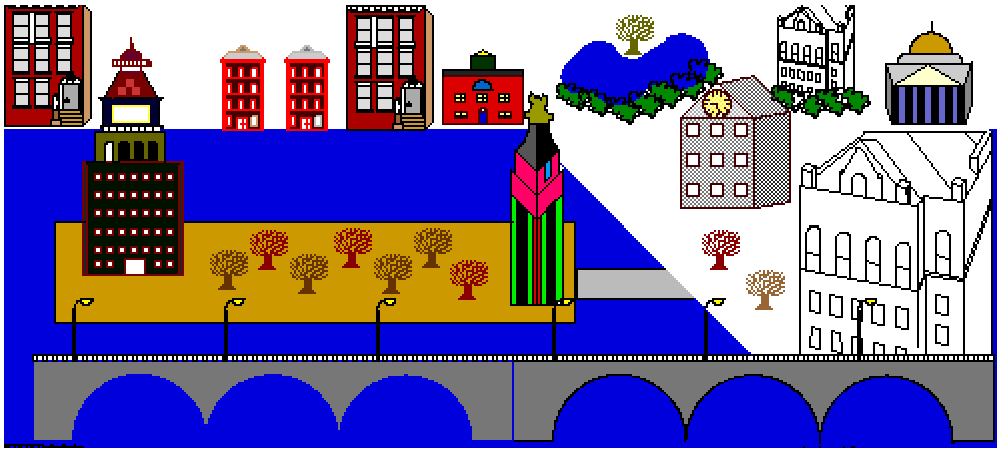
www.logofoundation.org

Cityscapes in Logo

by

Laura Allen



Director of the Computer Program
Computer Teacher, Grades 3-9
The Buckley School
New York, NY

Acknowledgments

Thanks to Seymour Papert, Michael Tempel, Eadie Adamson, Tom Trocco, and Linda Bluestone for their interest and enthusiasm about this project.

Special Thanks to the following boys who have worked on the Cityscapes project: Jeff Armstrong, Gardiner Anderson, Brad Aston, Naim Brown, Byron Chen, Teddy Dembowski, Charles Durkin, Sean Flynn, Griffy Foxley, Brian Gillespie, Jay Hallen, Nicholas Hobbs, John Lehman, Denny Lewis, John Lofberg, Rommy Martinez, Dan Paduano, Laurent Pelletier, Nick Sheets, Teddy Stephenson, Alexis Theodoracopulos, Loukas Zoumas.

Thanks to Mrs. Hawes' Fifth Grade class 1992-32 for their ideas and work on the stained glass windows.

Introduction

What could I do to create a palpable link between what my eighth graders were learning in Logo and their world away from school? In answer to this challenge, I developed a project that helped me grow as a teacher and excited my students about their own work and that of their classmates. My project idea was for each student to create an image of a city comprised of drawings of buildings created in Logo by his classmates.

I am one of two computer teachers at The Buckley School, a traditional, independent boys school in New York City. During this project we used the LCSI version of Logo, called LogoWriter. We met three times a week for 45 minutes and students had the option of coming to work every afternoon for an additional hour. Although each student worked at his own computer throughout the project, they often shared, compared, and discussed ideas. The first time I taught this project, I worked with nine boys in a computer lab equipped with 11 IBM compatible computers with CGA graphics. The second time I taught this project, we used Macintosh LC II computers with Macintosh LogoWriter.

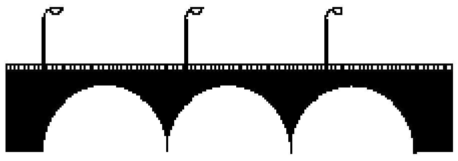
The project had two phases. First, each student designed and drew one building (first on paper, then using Logo). During the second phase of the project, the individual building programs were put together onto one LogoWriter page. Then each student created a "cityscape" using his building along with those of his classmates.

An important aspect of the project grew out of the need to alter the sizes of the various buildings when assembling them into the cityscape. This was accomplished by introducing variables into the programs so that each building could be scaled.

Starting the Project

To begin with, each boy drew an outline on paper of the building he wanted to create. As my school is located in the middle of New York City, I encouraged my students to look around and get ideas from buildings they see everyday. I also had picture books of buildings for them to look at. My goal was to encourage them to do a thoughtful drawing which incorporated elements from photos, or to develop their own types of building details and ornamentation.

Once at the computer and using Logo, each boy began to recreate the drawing of his building (one part at a time) onto the screen. They seemed to thoroughly enjoy the process. Some boys figured out their steps in the command center, wrote it down on paper, and then put it on to the flip side. Some used cut and paste to bring their developing procedures from the command center to the flip side. I found them suggesting design elements and offering help to one another. Boys who finished early had the option of designing a second building or city element; one boy designed a bridge (see below) and another designed a news stand.



Adding Variables

The next stage was the introduction of variables to enable the students to size each building according to its desired place in their cityscape. Don't panic! If you have had little or no experience with variables, it is still possible to understand enough to do this project. In fact, it may be a logical and easy way to understand how variables can operate.

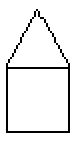
Each boy began with a finished building program that did not contain any variables. Then they added variables to each sub procedure. The following are procedures for a typical Logo house, with two sub-procedures: a box for the frame, and a triangle for the roof. Although the programs created by my students were considerably longer, there were no elements more complicated in their work, than in this example.

This program always draws a house of the same size.

```
to house
frame
fd 30
right 30
roof
end

to frame
repeat 4 [forward 30 right 90]
end

to roof
repeat 3 [forward 30 right 120]
end
```



In this version, the program draws houses of different sizes:

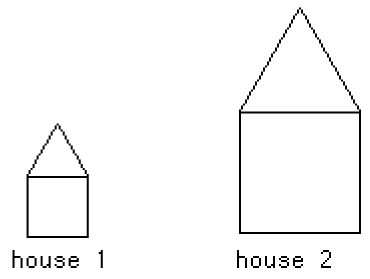
```
to house :scale
frame :scale
fd :scale * 30
right 30
roof :scale
end

to frame :s
repeat 4 [forward :s * 30 right 90]
end

to roof :s
repeat 3 [forward :s * 30 right 120]
end
```

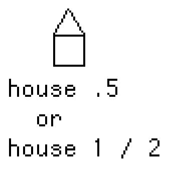
House 1 draws a house of the original size, with sides of 30 turtle steps. This is because when **:s** is 1, **:s * 30** = 30.

House 2 draws a house that is twice as big, with sides of 60 steps. When **:s** is 2, **:s * 30** = 60.



A house smaller than the original may be drawn by using fractional or decimal inputs:

House .5 or **House 1 / 2** draws a house with sides of 15 steps, because **.5 * 30** =15 (or **1/2 * 30** =15).



The approach I actually took with my students was somewhat different, and avoided the need to use fractional inputs to reduce the sizes of buildings. The procedures were rewritten so that an input of 10 would produce a drawing of the original size. An input of 20 would double it. An input of 5 would produce a drawing half the original size. The example above would come out like this:

```
to house :scale
frame :scale
fd 3 * :scale
right 30
roof :scale
end

to frame :s
repeat 4 [forward :s * 3 right 90]
end

to roof :s
repeat 3 [forward :s * 3 right 120]
end
```

The original forward distance of 30 is changed to 3, one-tenth of the original. In practice, things were not quite that easy. The forward distances were generally not multiples of ten. If the frame procedure had originally been

```
to frame
repeat 4 [forward 37 right 90]
end
```

it would have become

```
to frame :s
repeat 4 [forward :s * 3.7 right 90]
end
```

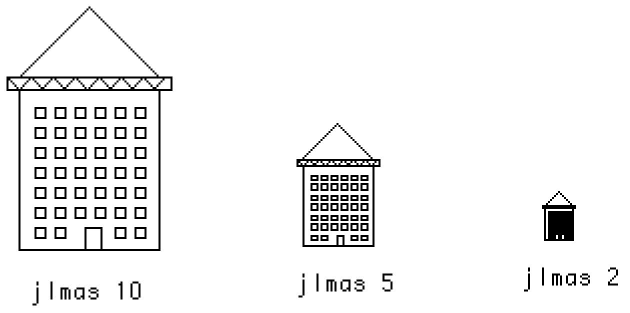
This use of decimals worked, but I am not sure that my students always understood the logic. Although they understood the concept of having a building change size; it was a bit more tricky for them to understand the mechanics of adding variables. I suggested they think of the variable (**:s**) as equal to 10. With **:s** = to 10, their goal was to get **:s * fd** to equal the original value. About half understood immediately; about a quarter followed by rote, then as they did it more, understood what they were doing. The rest followed my directions, but I don't think they completely "got it".

Here are some examples of student work. In the procedure **JLMAS**, JL stands for one boy's initials, MAS stands for "master", meaning a "master procedure" which represents a complete project.

Here is his super procedure for the complete building (If you would like to see the subprocedures that make up this building, see **Appendix II**):

```
to jlmas
jlsq
jlroof
jlroof1
jlmove1
repeat 6[jlrow jlmove2 ]
jlrow1
jlmove3
end
```

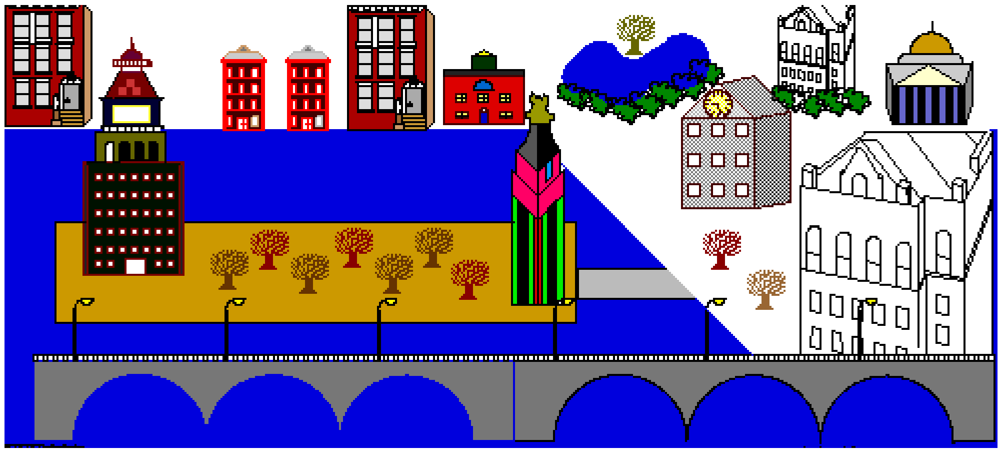
This is the building in three sizes. **JLMAS 10** is the same size as the original **JLMAS** prior to the introduction of variables.

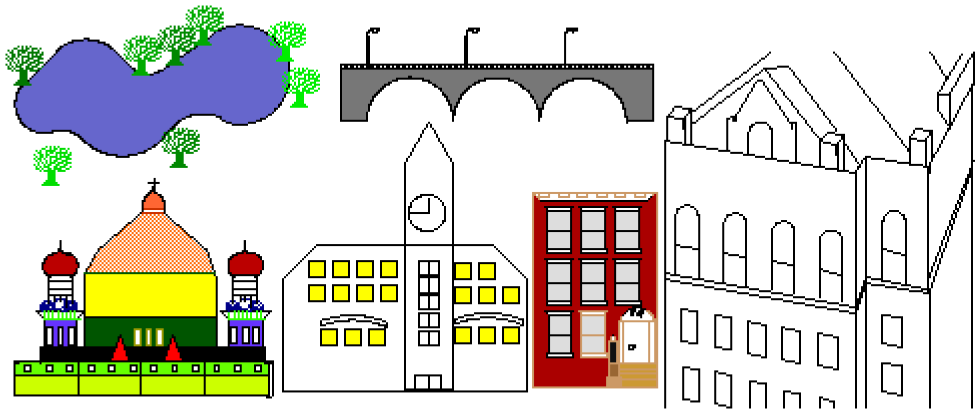


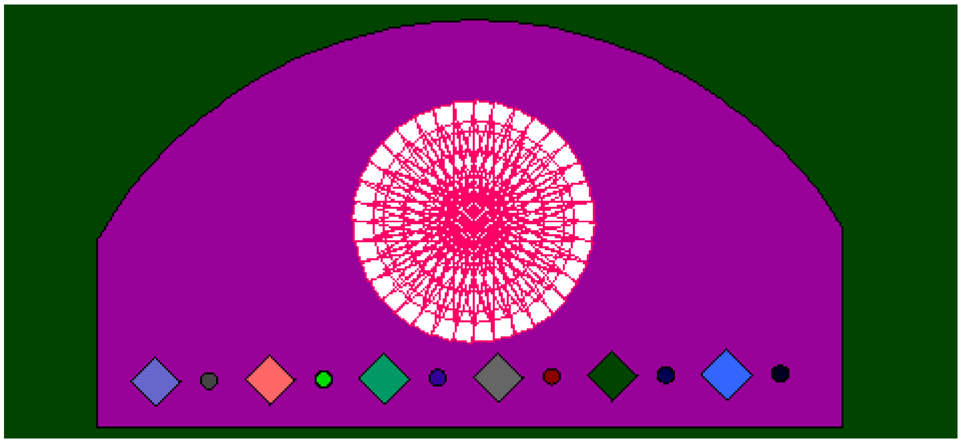
In this next example, the master procedure is **BGMAS** (If you would like to see the subprocedures that made up **BGMAS**, see **Appendix III**.)

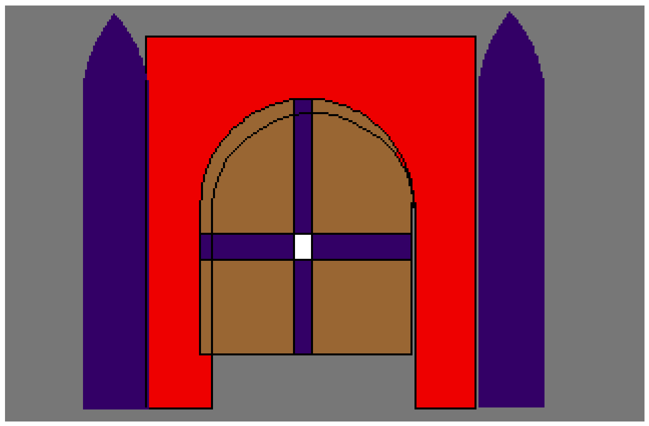
```
to bgmas :x
bgframe :x
bgmove1 :x
bgdemensions :x
bgdemensions2 :x
bgwind :x
bgwindows :x
bgwindow :x
bgrules :x
bgcircle :x
end
```

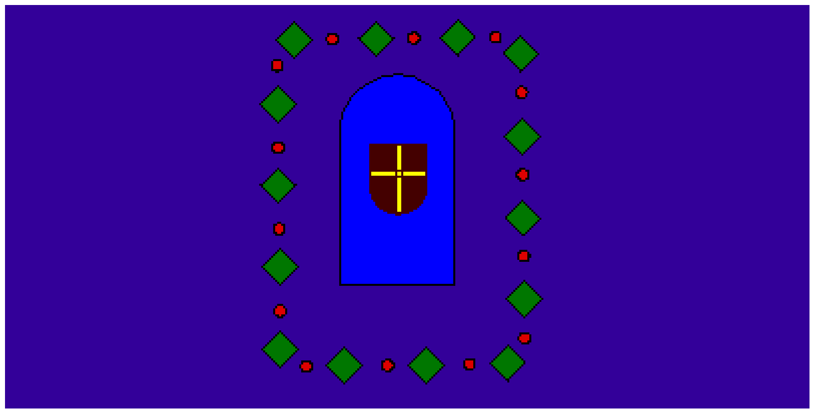
Cityscapes

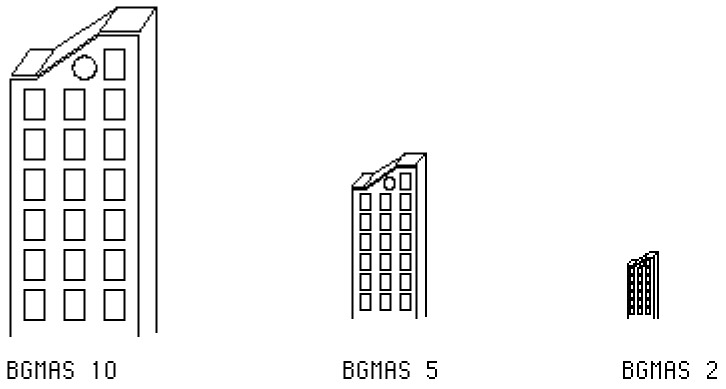












Although the mechanics of adding variables seemed difficult for some; most students quickly understood the potential flexibility and creative options using variables would afford them and became very excited. It was a big surprise to me that they grasped the larger concepts of using variables so quickly. Although I had created other projects which used variables, such as designing a simple house (like the example) and sizing it, ultimately it was not a very satisfying project: I never felt it engaged my students enough and it left me feeling unsure of how to explore the excitement of variables. But the Cityscape project seemed to concretize the concepts; making them much more easily understood and soon giving the boys a sense of ownership of the concepts involved.

Within about two classes each student had modified their procedures to include variables. I suggested they save the page first, so that if they made mistakes adding variables, they could use their pre-variable copy. Occasionally my students had used **home** or **setpos** (they were told not to, but had forgotten) and usually this was discovered at this stage. **Home** and **setpos** do not work well because the placement and sizing of buildings becomes limited. If they had used either of these commands, they had to rework their procedures to get rid of them.

Building the Cities

When each boy's building was complete I created a LogoWriter page that included everyone's work and called it CITY. This was a very, very large page. I copied it onto each boy's disk. There were two ways for the students to load it; as a normal LogoWriter page or as a tools page. I taught them to work with the page as a tools page, because there are many advantages to this method. (See Appendix I)

Using IBM compatible computers and IBM LogoWriter had advantages and disadvantages. The greatest advantage of the IBM was the amount of space each page had. We could create large tools pages containing twenty different buildings without running out of space. This was not the case when working with Macintosh LogoWriter: we found that we needed to change the allocation of memory to 3000K, but still ran into problems creating cities with more than eight buildings. The advantage of Macintosh is in coloring the buildings and in creating shapes. Macintosh LogoWriter has 256 colors and the ability to create and edit shapes is much more developed than in the IBM.

With the page loaded, the students were ready for the actual construction of cities. What was especially rewarding about this part of the project was seeing how differently each boy approached the task. Each cityscape was completely different from the others: Some boys approached the project with a lot of visual planning to produce a specific feeling. Some wanted a sense of traditional perspective - large buildings in the front, smaller buildings in the back. One boy wanted to make a road disappear in one point perspective, with the buildings along side of the road get bigger as they got closer. Some added rivers, lakes, and clouds. Others, less concerned with visual planning, basically scattered buildings on the screen.

The modular nature of this project and the necessity of using other students' work in the final phase created a new spirit of cooperation and helpfulness between my students. There was a sense of non-competitive interest and involvement in each other's work that was wonderful. Boys who generally did not relate to each other freely discussed ideas about the project.

At the completion of this phase, some students had time to add animation; people walking down the street, planes flying overhead, or helicopters landing on the top of buildings. These enhancements cannot be adequately conveyed on the printed page.

Extensions

Historical Buildings

This project could be expanded to tie in with the History curriculum. For example, if a class were studying ancient Greece, they could build the Parthenon and other actual structures and then design original buildings using the same elements.

Each student could be put in charge of designing a specific building part. For example, one student would be the column designer, another the window designer, another the door designer. The project could be even more specific; fluted column designers; non-fluted column designers; designers of Doric, Ionic, or Corinthian capitals. Putting the buildings together would be similar to "mad libs," but they would be visual.

Gardens

A similar project could revolve around the construction of a Logo garden. Each student would design a flower or plant. The class could then combine them into gardens. Variables could be used to size each plant in the final phase.

Stained Glass Windows

A few months ago the Fifth Grade History teacher asked if there was a way to coordinate the study of Medieval History with Logo. We developed a project for designing stained glass windows. Many of the concepts and methods in this project were the same as the Cityscape project. Each student drew a shape on paper, recreated it on the computer, then added variables. Then I made a page consisting of everyone's shapes and created a few "window panes" for them to use. They created a stained glass window using the shapes they had created. Some of the windows are shown on page seven.

Reflections

This project has given me insight into the importance of setting up situations where cooperation is not only necessary, but valued. It has also given me ideas about how such project could be integrated into the larger curriculum.

Once, while boys were working during their free time, the art teacher wandered in. She was impressed by their work and began thinking about possible collaborative efforts.

All in all this has been very enjoyable project to teach. I enjoyed dealing with issues that brought my students together in a cooperative, non-competitive manner. I liked the types of discussions the student had. I liked having each student work on his own part which was also part of a larger class project. I look forward to creating other projects that are as engaging to my students as this one was.

Appendix I

Initially I didn't how to work with tools pages, so the first time I did this project, each student loaded in a page called "CITY" which included everyone's work. The flip side was huge! It contained approximately 50 pages of procedures. To add procedures for their cityscape, each student had to work above or below all of this. Although tedious, it worked for a while, until we ran into a problem. An error message appeared on the screen

```
"out of space"
```

We couldn't type anymore on the flip side and we couldn't save. What now?

I learned the advantages of loading a page as a tools page:

- You save a lot of space.
- When you go to the flip side, you don't see any of the procedures, but when you type the procedure names, they work.

They are hidden, and operate as if they are primitive procedures. This solved the space problem and also added a different perspective to the "Cityscape" stage. Now that each building was in a hidden but usable form, we felt as though we had added to the list of Logo primitives. Although it was not a huge difference, the mystery of not seeing the procedures any more changed my students attitudes. This was a new way of using Logo, which elevated the experience, making all of us view the "Cityscape" stage with more excitement and engagement.

Appendix II

Below are the procedures that made up JLMAS

```
to jlsq
repeat 2[fd 80 rt 90 fd 70 rt 90]
end

to jlroof
fd 80
lt 90
fd 6
rt 90
fd 6
rt 90
fd 82
rt 90
fd 6
rt 90
fd 6
end

to jlroof1
rt 90
pu
fd 6
pd
lt 45
fd 50
lt 90
fd 50
end

to jlmove1
seth 90
pu
fd 1
seth 180
fd 15
seth 90
fd 8
end

to jlwindow
repeat 4[fd 5 rt 90]
end
```

```
to jlrow
pd
repeat 6[jlwindow pu fd 10 pd]
end

to jlmove2
pu
bk 60
rt 90
fd 10
lt 90
end

to jlrow1
pd
repeat 2[jlwindow pu fd 10 pd]
pu
fd 5
seth 0
pd
bk 10
fd 10
rt 90
fd 7.5
rt 90
fd 10
bk 10
seth 90
pu
fd 7.5
pd
repeat 2[jlwindow pu fd 10 pd]
end
```

```
to jlmove3
pu
fd 2
lt 90
fd 69
lt 90
fd 76
rt 90
fd 6
rt 180
lt 45
pd
repeat 7[fd 8 lt 90 fd 8 rt 90]
end
```

Now here is the same program with variables introduced:

```
to jlmas :a
jlsq :a
jlroof :a
jlroof1 :a
jlmove1 :a
repeat 6[jlrow :a jlmove2 :a]
jlrow1 :a
jlmove3 :a
end

to jlsq :a
repeat 2[fd :a * 8 rt 90 fd :a * 7 rt 90]
end

to jlroof :a
fd :a * 8
lt 90
fd :a * .6
rt 90
fd :a * .6
rt 90
fd :a * 8.2
rt 90
fd :a * .6
rt 90
fd :a * .6
end
```

```
to jlroof1 :a
rt 90
pu
fd :a * .6
pd
lt 45
fd :a * 5
lt 90
fd :a * 5
end

to jlmove1 :a
seth 90
pu
fd :a * .1
seth 180
fd :a * 1.5
seth 90
fd :a * .8
end

to jlwindow :a
repeat 4[fd :a * .5 rt 90]
end
to jlrow :a
pd
repeat 6[jlwindow :a * 1 pu fd :a * 1 pd]
end

to jlmove2 :a
pu
bk :a * 6
rt 90
fd :a * 1
lt 90
end
```

```
to jlrow1 :a
pd
repeat 2[jlwindow :a * 1 pu fd :a * 1 pd]
pu
fd :a * .5
seth 0
pd
bk :a * 1
fd :a * 1
rt 90
fd :a * .75
rt 90
fd :a * 1
bk :a * 1
seth 90
pu
fd :a * .75
pd
repeat 2[jlwindow :a * 1 pu fd :a * 1 pd]
end

to jlmove3 :a
pu
fd :a * .2
lt 90
fd :a * 6.9
lt 90
fd :a * 7.6
rt 90
fd :a * .6
rt 180
lt 45
pd
repeat 7[fd :a * .8 lt 90 fd :a * .8 rt 90]
end
```

Appendix III

Below are the procedures that made up BGMAS

```
to bgmas :x
bgframe :x
bgmove1 :x
bgdemensions :x
bgdemensions2 :x
bgwind :x
bgwindows :x
bgwindow :x
bgrules :x
bgcircle :x
end

to bgframe :x
ht
fd :x * 12.8
rt 90
fd :x * 1.2
lt 35
fd :x * 3.9
seth 90
fd :x * 2
rt 90
fd :x * 15
end

to bgmove1 :x
pu
rt 90
fd :x * 6.3
rt 90
fd :x * 12.925
pd
end
```

```
to bgdemensions :x
rt 35
fd :x * 1.6
bk :x * 1.6
seth 90
fd :x * 1.2
seth 35
fd :x * 1.7
bk :x * 1.7
rt 20
fd :x * 3.9
seth 35
fd :x * 1.4
bk :x * 1.4
seth 90
fd :x * 2.0
seth 35
fd :x * 1.5
end

to bgdemensions2 :x
seth 180
fd :x * 16.2
bk :x * 16.2
seth 270
fd :x * 2
seth 237
fd :x * 3.9
seth 270
fd :x * 1
end

to bgwind :x
pu
seth 180
fd :x * 13.5
seth 90
fd :x * 4.65
lt 90
pd
end

to bgwindow :x
repeat 2[fd :x * 1.5 lt 90 fd :x * 1.0 lt 90]
end
```

```
to bgtowindow :x
pu
lt 90 fd :x * 2.0 rt 90
pd
end

to bgrow :x
repeat 3[bgwindow :x bgtowindow :x]
end

to bgrules :x
pu
lt 90
fd :x * 2.0
rt 90
fd :x * .01
lt 90
pd
end

to bgbegrow :x
pu
fd :x * 2.0
rt 90
fd :x * 6.0
lt 90
pd
end

to bgwindows :x
repeat 6[bgrow :x bgbegrow :x]
end

to bgcircle :x
repeat 90[ fd :x * .04 rt 4]
end
```