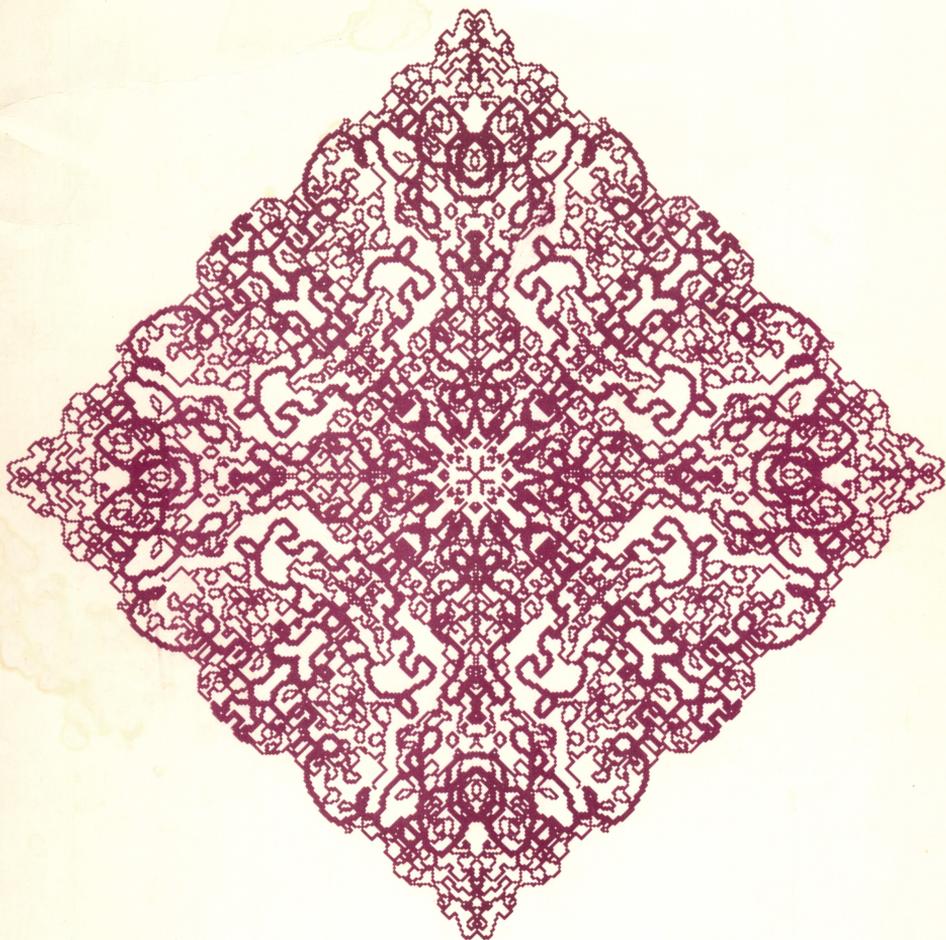


SpaceLace™

An Interactive Kaleidoscope



Requires Apple IIGS, II+, IIe, IIfx



Great Wave
Software™

Software Excellence For All the Ages

SpaceLace

An Interactive Kaleidoscope

by

Lucia Grossberger and Bob Bishop



Great Wave
Software™

*For Samuel and Camela Grossberger
With sincere gratitude for all their
help and support*

Warranty

Great Wave Software's sole warranty is that the program media will be free from defects in materials and workmanship under normal use for a 90 day period from the date of receipt by the customer and that the program operation will substantially conform to the description in this manual during that 90 day period. The warranty and remedies set forth above are exclusive and in lieu of all others, express or implied.

GREAT WAVE SOFTWARE EXPRESSLY EXCLUDES ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE EXCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Great Wave Software reserves the right to make any changes, additions, or improvements to this software or documentation at any time without notice to any person or organization.

Trademarks

Great Wave and SpaceLace are trademarks of Great Wave Software. Apple II and Apple IIGS are trademarks of Apple Computer Inc.

Copyright

This software and manual are protected by copyright laws. The purchasers of this software are granted a limited license to use the copyrighted programs only for their own personal use. The manual may not be copied in whole or in part without written consent from Great Wave Software. Backup copies of the program disk may be made provided that they are only for personal use by the purchaser of this product.

At the request of Apple Computer Inc, we have included the following statement: The system files are copyrighted programs of Apple Computer , Inc. licensed to Great Wave Software to distribute only in combination with SpaceLace. Apple Software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of SpaceLace. When SpaceLace has completed execution, Apple Software shall not be used by any other program.

Package and Cover Art

Copyright © 1987 Great Wave Software - All Rights Reserved

SpaceLace Software

Copyrighted © 1987 Lucia Grossberger and Bob Bishop - All Rights Reserved

SpaceLace Manual - First Edition

Copyright © 1987 Lucia Grossberger and Bob Bishop - All Rights Reserved

Table of Contents

<i>Introduction</i>	<i>v</i>
<i>Chapter 1: Creating a Traditional Kaleidoscope</i>	1
<i>Chapter 2: Computer Kaleidoscopes</i>	3
The <i>SpaceLace</i> Diskette	5
Exploring a Computer Kaleidoscope	7
The Parameters in KALEIDOKEY	9
Peeking in on other Computer Kaleidoscopes	11
<i>Chapter 3: KALEIDOMAKERS</i>	15
Hexadecimal Numbers	15
Changing the Parameters in KALEIDOMAKER1	17
Changing the Parameters in KALEIDOMAKER2	21
The Parameters in KALEIDOMAKER1 and KALEIDOMAKER2	21

<i>Chapter 4: Creating Your Own Computer Kaleidoscope</i>	25
A Hi-Res Kaleidoscope	31
<i>Appendix 1: Answers to Kaleidoscope Questions</i>	33
<i>Appendix 2: Reference to KALEIDOKEY1 and KALEIDOKEY2</i>	34
<i>Index</i>	39

Introduction

SpaceLace is a visually exciting series of interactive kaleidoscopes that will run on any Apple II computer. Computer kaleidoscopes offer some very exciting possibilities. Not only are they captivating as visual displays, but they can also be fun and rewarding exercises for beginning to intermediate programmers.

SpaceLace is divided into four chapters. The following is a brief description of what is covered in each of the chapters.

Chapter 1

"Creating a Traditional Kaleidoscope", describes how the most common type of traditional kaleidoscope is created.

Chapter 2

"Computer Kaleidoscopes", describes the nine different computer kaleidoscopes provided on the *SpaceLace* diskette. These kaleidoscopes are very interactive and easy to use. Most often only a keystroke is required to create some dramatic changes. Many of the kaleidoscopes included in this chapter have been enjoyed by the very young and the very old in art and science museums around the country. In this chapter there is also a fun hands-on tutorial exploring some of the different approaches and problems involved in creating a computer kaleidoscope. This chapter introduces some very important computer science concepts in a fun, visual way.

Chapter 3

"KALEIDOMAKERS", describes how to use the KALEIDOMAKER Programs on the diskette. Using these programs, you can define your own computer kaleidoscope.

Chapter 4

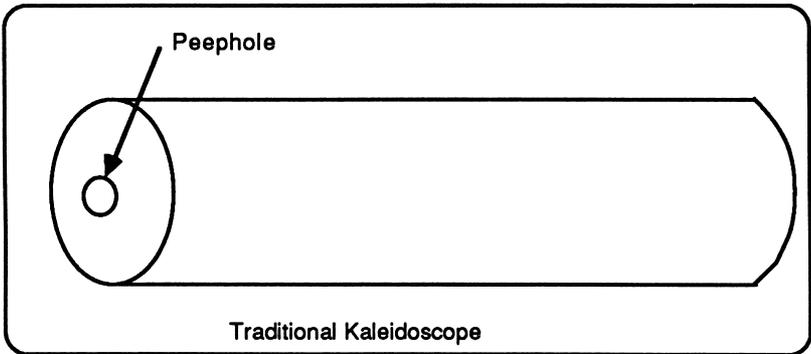
"Creating Your Own Computer Kaleidoscope", is an introduction to programming kaleidoscopes in AppleSoft BASIC. AppleSoft BASIC is included with every Apple II

computer. The computer science concepts that were introduced in Chapter 2, are now implemented in well documented simple programs that create some visually exciting computer kaleidoscopes. The programs provided in this chapter, are excellent exercises for beginning and intermediate programmers.

Chapter 1

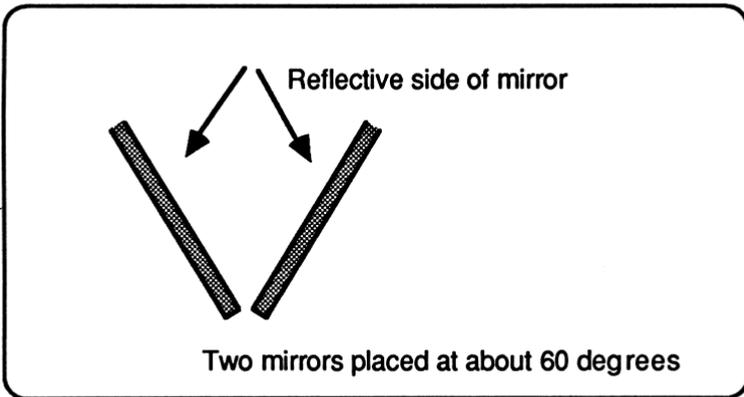
Creating a Traditional Kaleidoscope

A traditional kaleidoscope is a tube with a peephole at one end and small pieces of glass, or other such things, at the other end. As you look through the kaleidoscope's peephole you see a symmetrical pattern that changes as you turn the tube.



A traditional kaleidoscope is easy to create - it's all done with mirrors. The following steps can be used for creating a traditional kaleidoscope.

- Take two rectangular mirrors (each about five inches long and one and a half inches wide) and a sheet of paper that contains clear simple designs
- Place the two mirrors together, on top of the paper so that they meet at about a 60 degree angle surrounding one of the designs.



- If you look at the space between the two mirrors, instead of one design, you should see six designs. One is the actual design and the other five are mirror reflections.

If you want to change the number of designs that you see, change the angle of the two mirrors. The smaller the angle between the two mirrors, the greater the number of designs. The larger the angle the fewer number of designs you will see. Notice also that the designs will not all be facing the same direction.

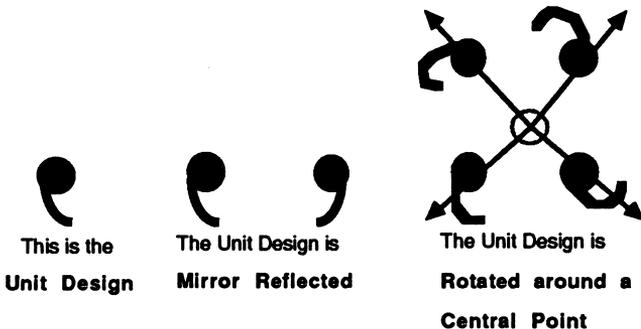
Chapter 2

Computer Kaleidoscopes

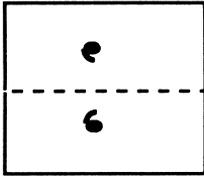
How would a computer kaleidoscope look? First of all, it would be quite an elaborate process to put a computer kaleidoscope in a tube! In fact the computer kaleidoscopes you will be exploring and creating will not even be round. Instead, these computer kaleidoscopes will fit into the rectangle of your T.V. (or monitor) screen.

As mentioned before, a kaleidoscope is a symmetrical pattern. On a computer, some symmetrical patterns are easier to create than others. Because of the resolution of the Apple and the layout of its pixels on its X,Y grid, it is impossible to accurately display symmetrical patterns involving multiples of three (3, 6, 9, etc.). But some multiples of two (2, 4, and 8) are simple to create and fit very nicely on the Apple's screen.

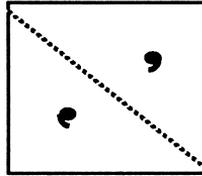
The **unit design** is the initial design. In the figure below, the unit design is the circle with a directional tail. The unit design can be repeated in two ways. The first way that the design can be repeated is that it can be reflected. The second way it can be repeated is that it can be rotated along a circular path. The figure below shows how both rotation and reflection are achieved.



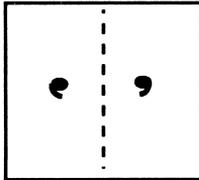
The figure below shows the symmetries that can easily be created on the Apple's screen.



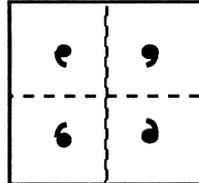
Pattern 1



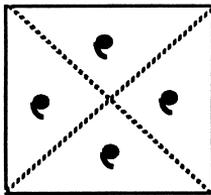
Pattern 2



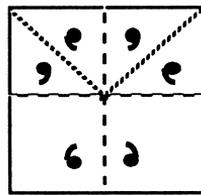
Pattern 3



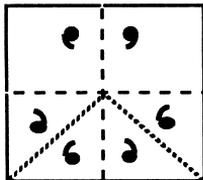
Pattern 4



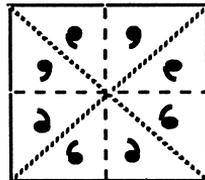
Pattern 5



Pattern 6



Pattern 7



Pattern 8

Several of these symmetrical patterns could not be created with mirrors. Designs 5, 6, and 7 could not be achieved with mirrors. Design 5 rotates the design through the center of the coordinate system. It is called a rotational symmetry. Designs 5 and 6 would also be impossible using mirrors, yet they are very rich patterns. More than any other symmetry, they resemble the bilateral symmetry of faces. Using KALEIDOMAKERS in *SpaceLace*, it is possible to explore other symmetries and asymmetries.

The *SpaceLace* Diskette

To begin, boot up the disk, a **MAIN MENU** should appear which contains the following information:

```
** SPACELACE **
KALEIDOSCOPIES
KALEIDOMAKERS
EXIT
```

**USE THE <I> AND <M> KEYS TO MOVE.
USE THE <RETURN> KEY TO SELECT.**

At this point you can enter the different sections of the program, or you can quit the program. Using the <I> key to move up, the <M> key to move down, and the <RETURN> key to select, you can select either section.

If you select KALEIDOSCOPIES, the program will display the KALEIDOSCOPIES MENU:

```
** KALEIDOSCOPIES MENU **

KALEIDOKEY
SOUTHWEST
KALEIDOMUSIC
KALEIDOTWISTER
KALEIDOKEY2
DOILIES
SERENITY
```

ELECTRIC WARMTH

USE THE <I> AND <M> KEYS TO MOVE.
USE THE <RETURN> KEY TO SELECT.

On pages 10 - 13 there is a more complete description of each of these kaleidoscopes. All of these kaleidoscopes, except KALEIDOTWISTER, are interactive.

Again in this menu, use the <I> and <M> keys to move, and the <RETURN> Key to select an item. If you want to leave the KALEIDOSCOPIES MENU, and return to the MAIN MENU, press <CTRL> <RESET>. Whenever you are in any of the kaleidoscope programs. you can quit that program by pressing <CTRL> <RESET> and the program will return to the MAIN MENU.

If you select KALEIDOMAKERS from the MAIN MENU, the following menu will be displayed:

```

** KALEIDOMAKER **
  KALEIDOMAKER1
  KALEIDOMAKER2

```

USE THE <I> AND <M> KEYS TO MOVE.
USE THE <RETURN> KEY TO SELECT.

Using the KALEIDOMAKERS you can design your own kaleidoscopes. On page 14, there is a description of these programs. Appendix 2 contains directions on how to create kaleidoscopes with the KALEIDOMAKER Programs. As in the other menus, use the <I> and <M> keys to move and the <RETURN> Key to select. If you want to leave the KALEIDOMAKERS MENU and return to the MAIN MENU, press <CTRL> <RESET>. Whenever you are in either of the KALEIDOMAKER programs, you can quit pressing <CTRL> <RESET> and the program will return to the MAIN MENU.

To quit *SpaceLace* you must be in the MAIN MENU. From there select EXIT and you will be in AppleSoft BASIC.

Exploring a Computer Kaleidoscope

The following is a hands-on exploration to introduce you to KALEIDOKEY. KALEIDOKEY is related to many of the kaleidoscopes in this package, so what you learn in this exercise will be applicable to other kaleidoscopes.

At this point, from the KALEIDOSCOPIES MENU, select KALEIDOKEY. In KALEIDOKEY whenever you press a letter key a different kaleidoscope will appear on the screen. Each of the kaleidoscopes will be named according to its corresponding key. For example the kaleidoscope that appears when you press the <P> key will be called the <P> Kaleidoscope.

Answer as many of the questions as you can. Don't be frustrated if the answers are a little difficult. If you don't know the answer just go on to the next question. The answers to these questions are in Appendix 1. The concepts explored in these questions will be discussed in the next section. To quit the KALEIDOKEY program and return to the MAIN MENU, press <CTRL> <RESET>.

1. Watch the <A> Kaleidoscope. (Remember that this is the kaleidoscope that comes up when you press the <A> key.) Now, watch the Kaleidoscope. What is the difference between the <A> Kaleidoscope and the Kaleidoscope?
2. Watch the <C> Kaleidoscope. Now, watch the <D> Kaleidoscope. What is the difference between the two kaleidoscopes?
3. How many times is the unit design repeated in the <E> Kaleidoscope?
4. How many times is the unit design repeated in the <F> Kaleidoscope?

5. How many times is the unit design repeated in the <G> Kaleidoscope?
6. How many times is the unit design repeated in the <H> Kaleidoscope?
7. How would you describe the type of symmetry in the <I> Kaleidoscope?
8. Look at the <J> Kaleidoscope and then look at the <K> Kaleidoscope. Can you see any difference between the two kaleidoscopes?
9. This is a somewhat difficult question, so be observant! Press the <L> key and notice the very first pixel. Keep watching that pixel. What happens to that pixel? Pressing the <L> key again will restart the kaleidoscope.
10. This question is related to question number 9. The <L> Kaleidoscope does not have very many pixels on the screen, yet it is always adding pixels. What is happening to the pixels?
11. Watch the <M> Kaleidoscope and notice the pattern it creates. Now watch the <N> Kaleidoscope. What is the difference between the two Kaleidoscopes?
12. Now this is an easy one! What is the difference between the <O> Kaleidoscope and the <P> Kaleidoscope?
13. Watch the <Q> Kaleidoscope. Now watch the <R> Kaleidoscope. They both use the same colors. What is the difference?

The Parameters in KALEIDOKEY

In the computer program **KALEIDOKEY**, each of the letter kaleidoscopes is defined by seven different settings or parameters. The questions in the exercise above explored the different effects of changing the quantities contained in these parameters. Can you guess what some of the parameters are?

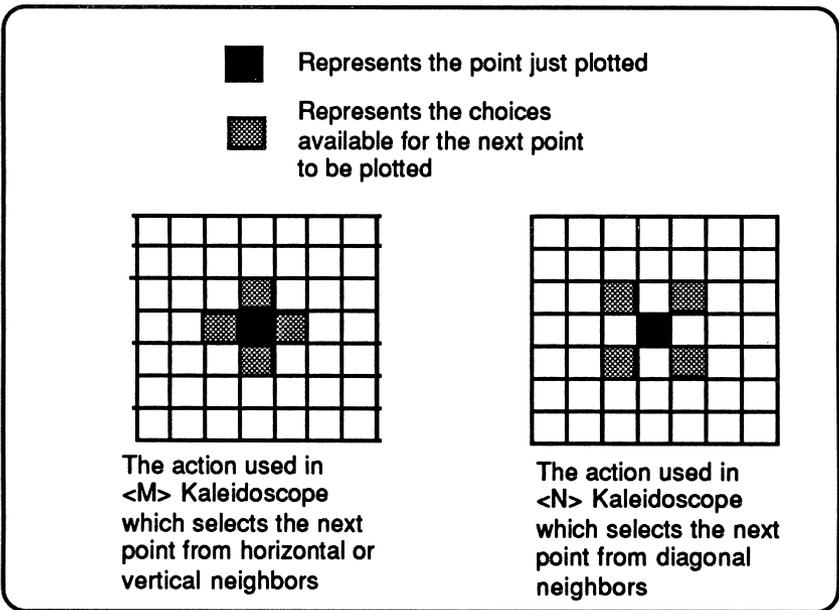
Four of the parameters are fairly obvious. They are:

- the different graphics modes (Lo-Res Graphics and Hi-Res Graphics)
- the different colors
- the different symmetries
- the different speeds

These parameters are the simple ones. The other three are much more difficult and subtle.

In question number 8 you were asked to find the difference between the <J> Kaleidoscope and the <K> Kaleidoscope. The answer is that one had a lot more pixels than the other. And, in question number 9 you saw that after a pixel was on the screen for a while that pixel was removed from the screen. The process of removing the pixel from the screen was accomplished with a data structure called a que. Ques are very useful concepts in computer programming. Essentially, a que is nothing more than a list in which new information gets inserted or placed at one end, while old information gets removed from the other end. It's very similar to standing in line at a movie theatre to buy a ticket to see a movie. To get in line you simply stand behind the last person in the line and anybody who comes to join the line after you will stand behind you, and so on. Meanwhile at the head of the line, the clerk selling tickets will process the first person in line followed by the second person in the line and so on until it is your turn. Then, after you are finished, anybody standing behind you will subsequently get their turn. As a matter of fact the English use the phrase "standing in que". In other words, a que is a structure which has the property that whatever goes in first comes out first and whatever goes in last comes out last.

The structure of the patterns created by the <M> Kaleidoscope and the <N> Kaleidoscope were different. The patterns of the <M> Kaleidoscope look as if though they were created on a checkerboard. When creating a kaleidoscope program an important question to ask is: "How will the program decide where to place the next pixel?" In the case of the <M> Kaleidoscope the position of the next dot to be plotted was randomly selected from one of its horizontal or vertical neighbors. In the <N> Kaleidoscope the next pixel to be plotted was randomly selected from one of its diagonal neighbors. The figure below illustrates this:



Another important question to ask is; "What color should be used to plot the new pixel?" The final parameter specifies how long the program will continue plotting in the current color. For example, in the <O> Kaleidoscope the color remained constant for a long period of time. In the <P> Kaleidoscope the color changed fairly quickly. Notice that in the <O> Kaleidoscope there are clusters of color while in the <P> Kaleidoscope the colors are scattered.

Peeking in on other Computer Kaleidoscopes

KALEIDOKEY does not include all of the possible parameters that could be included in a computer kaleidoscope program. There are many, many different kinds of kaleidoscope programs that could be written. In addition to **KALEIDOKEY**, the diskette contains several more examples of kaleidoscope programs. **KALEIDOKEY2**, **DOILIES**, **SERENITY**, **ELECTRIC**, and **WARMTH** all use the same parameters as **KALEIDOKEY**. **KALEIDOMUSIC**, **SOUTHWEST**, and **KALEIDOTWISTER** use different types of parameters.

The following is a description of the other kaleidoscopes included on the diskette. The description will tell you which graphics mode is required. If you try to run a kaleidoscope with a graphics mode that is not available on your particular Apple II computer, it will crash the system and you will have to reboot.

SOUTHWEST:

SOUTHWEST is a Hi-Res kaleidoscope. The number keys from 1 to 7 will change the speed. Number 1 is the slowest and number 7 is the fastest. Instead of plotting single pixels, **SOUTHWEST** plots short line segments. This kaleidoscope will run on any Apple II computer.

KALEIDOMUSIC:

Plug in a tape recorder into the cassette input jack of your Apple II+, or Apple IIe, play a tape and watch what happens to the kaleidoscope. (It is also possible to use a microphone with amplifier.) This kaleidoscope will run only on an Apple II+ and Apple IIe (the Apple IIc and the Apple IIGS do not have a cassette port.)

KALEIDOTWISTER:

This Hi-Res kaleidoscope uses a special "scribble algorithm" to select the next pixel. In other words, instead of randomly choosing a next pixel, the program starts out choosing pixels in a straight line. It then randomly starts curving this line either to the left or to the right. KALEIDOTWISTER will run on any Apple II computer.

KALEIDOKEY2:

KALEIDOKEY2 is a collection of Lo-Res and Hi-Res Kaleidoscopes. Each letter and number key contains a different set of parameters. It will run on any Apple II computer. Just press the letter and number keys to see the different Kaleidoscopes.

DOILIES:

DOILIES is a collection of Double Hi-Res Kaleidoscopes. Each letter and number key contains a different set of parameters. It requires Double Hi-Res.

SERENITY:

SERENITY, as well as ELECTRIC and WARMTH all require Apple IIGS Super Hi-Res mode. The interaction is different than the interaction in DOILIES, KALEIDOKEY and KALEIDOKEY2 in order to take advantage of the different colors available. The keys on the top two rows of the keyboard define the different parameters for the kaleidoscopes, except for the colors. The keys on the bottom two rows of the keyboard specifies different color sets. When the Kaleidoscope begins, it defaults to the set of parameters defined by the <A> Kaleidoscope and the standard color set. Select any of the kaleidoscopes on the top two rows and then select any of the colors in the bottom two rows.

In going from one color set to another the program blends between the two colors. The colors defined for SERENITY are designed to be serene and tranquil.

ELECTRIC:

ELECTRIC incorporates the same design as **SERENITY**, except that the colors are exciting and striking.

WARMTH:

In this Kaleidoscope the colors are warm and natural. It also functions just like **SERENITY AND ELECTRIC**.

Chapter 3

KALEIDOMAKERS

On the MAIN MENU in addition to the category of KALEIDOSCOPIES, there is a category KALEIDOMAKERS. By selecting KALEIDOMAKERS you will be in KALEIDOMAKERS MENU, where there are two programs, KALEIDOMAKER1 and KALEIDOMAKER2. Using either of these two programs, you can set the parameters and create your own kaleidoscope. (These parameters include: GRAPHICS Mode, RATE (or Speed), ACTION, QUEUE Size, COLOR, SYMMETRY, and DURATION.)

The major difference between the two KALEIDOMAKERS is that KALEIDOMAKER1 uses Graphics modes Lo-Res, Hi-Res, Double Lo-Res, and Double Hi-Res. KALEIDOMAKER2 takes advantage of Super Hi-Res Graphics. Setting the parameters in KALEIDOMAKER1 will be discussed on page 16. Setting the parameters in KALEIDOMAKER2 will be discussed on page 20. The section on page 20 - 23 describes all of the parameters. Appendix 2 is a quick reference to the KALEIDOMAKER Programs.

Hexadecimal Numbers

The parameters in the two KALEIDOMAKER programs are specified by using hexadecimal numbers. The reason hexadecimal was chosen is that it is well suited to the task. (After all, the computer at the most basic level really uses binary, or base 2. Binary is often represented as hexadecimal, because any four binary digits can be expressed with one hexadecimal digit.) When selecting colors in Lo-Res, Double Lo-Res, Double Hi-Res and Super Hi-Res there are sixteen possible colors available. Hexadecimal, or base 16 has sixteen different symbols. Therefore any of the 16 colors can be selected with one hexadecimal digit.

The following chart shows the difference in notation between decimal, or base 10 and hexadecimal or base 16:

Notice that in hexadecimal the letters, A,B,C,D,E and F are used

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 16	\$0	\$1	\$2	\$3	\$4	\$5	\$6	\$7	\$8	\$9	\$A	\$B	\$C	\$D	\$E	\$F

to represent the decimal values 10 through 15 respectively. Using hexadecimal numbers, one numeral can represent 16 different values (remember the zero). When referring to hexadecimal numbers a dollar sign (\$) is used in front of the numerals to signify that hexadecimal versus decimal is being used.

For this program, except when selecting colors (where \$F is the largest number), it is not important that you know how to convert base 16 numbers to base 10 numbers, or visa versa. It is only important that you have a sense of the number. For example, when selecting QUE, it is only important that you have a sense of the hexadecimal number you are choosing, but you need not know the exact value. Even when you use decimal, often times you only have a sense of the number, but you don't really know exactly what that number is. For example, when you encounter a number such as 187, you know that there are 7 items in the ones position, 8 units in tens position, and 1 unit in the hundredths position. But most of us don't have an exact picture of 187, as compared to 186, or 188. Likewise, in hexadecimal, you have a sense that \$A5 is larger than 100 decimal, but you don't need to know what the exact number is. The following is a table of a few hexadecimal numbers and their decimal equivalents:

Base 16	Base10
\$F	15
\$10	16
\$FF	255
\$100	256
\$FFF	4095

Changing the Parameters in KALEIDOMAKER1

At this point, using the <I> key to move up, the <M> key to move down, and the <RETURN> key, select an item. Select KALEIDOMAKERS from the MAIN MENU. From the KALEIDOMAKERS MENU select KALEIDOMAKER1.

Once KALEIDOMAKER1 is loaded into memory, the following menu will appear:

PARAMETERS			

GRAPHICS	-----		\$1
RATE	-----		\$20
ACTION	-----		\$2
QUE	-----		\$150
COLOR	1001	0000	1100 1111
SYMMETRY	-----	1001	0110
DURATION	-----		\$A1

Note that the numbers following each of the parameters will probably be different on your screen than those shown above. If you have forgotten what is meant by each of the different parameters, refer to the section titled "The Parameters of KALEIDOMAKER" on page 20.

Notice that the first letter of each of the word is in inverse video. To change a given parameter; first type the number(s) that you want to change the parameters to, and then type <CTRL> key along with the highlighted letter of the parameter you are changing. For example, suppose you want to change the Graphics Mode to Lo-Res Graphics. Look at Appendix 2 and

notice that Lo-Res is Graphics Mode 1. Therefore, type: <1> and then type <CTRL> <G>. If you decide to change the Que size to \$100, first type the number 100 (<1><0><0>) then type <CTRL> <Q>. If you select a number that is not acceptable for that particular parameter, the software will give you an error beep. If you get an error beep, just make sure you have typed the correct number and try again.

Changing colors and symmetry is a little different. On the PARAMETERS MENU next to COLOR are a series of sixteen ones and zeros divided up into four groups. These represent the sixteen available colors in Lo-Res, Double Lo-Res and Double Hi-Res. In Hi-Res, the second set of eight positions functions exactly like the first set. These "1"s and "0"s next to the COLOR parameter are switches. That means that if they have a value of "1", then they are "on". If they have a value of "0", they are "off".

In Appendix 2, there are charts for all of the colors available in each of the graphics modes. The following chart shows the colors for Lo-Res and Double Lo-Res:

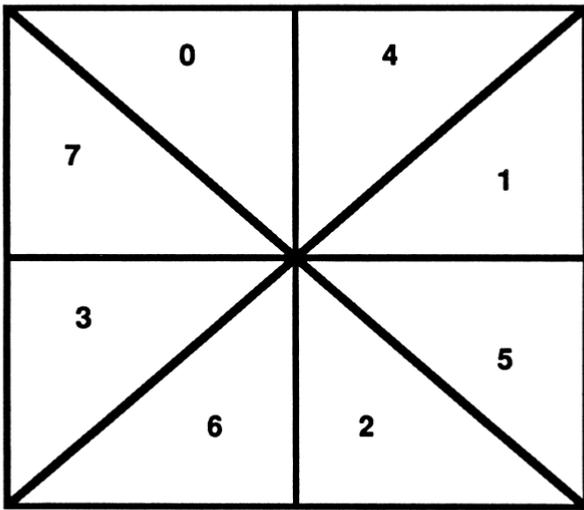
Lo-Res and Double Lo-Res	
\$0 = Black	\$8 = Brown
\$1 = Magenta	\$9 = Orange
\$2 = Dark Blue	\$A = Gray2
\$3 = Purple	\$B = Pink
\$4 = Dark Green	\$C = Light Green
\$5 = Gray1	\$D = Yellow
\$6 = Medium Blue	\$E = Aquamarine
\$7 = Light Blue	\$F = White

The following chart shows what value each of the positions indicates.

COLOR	1	0	0	1	0	0	0	0	1	1	0	0	1	1	1	0
POSITION	\$F	\$E	\$D	\$C	\$B	\$A	\$9	\$8	\$7	\$6	\$5	\$4	\$3	\$2	\$1	\$0

For example, in the setting on the previous page, the colors, Magenta (\$1), Dark Blue (\$2), Purple (\$3), Medium Blue (\$6), Light Blue (\$7), Light Green (\$C), and White (\$F) are switched on (set to "1"). The other colors are switched off (set to "0"). Suppose that you want to switch Magenta off. To do this, type the number <1> and then type <CTRL> <C>. To turn on the color Aquamarine, type the hexadecimal number <E> and then type <CTRL> <C>. This will switch on the Aquamarine color.

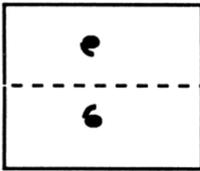
The following figure shows the different symmetry points that are available:



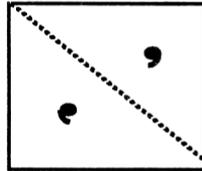
Using this system it is possible to create a wide variety of different symmetrical and asymmetrical patterns. On the PARAMETERS MENU next to SYMMETRY, there are eight switches. Each of these switches is represented in the diagram. For example, one way to get two point diagonal symmetry, would be to activate section 1 and section 7. That means that the switches for the those sections must contain "1"s, while all the other switches must contain "0"s. The following diagram shows how this would look on the PARAMETERS MENU:

SYMMETRY	1	0	1	0	0	1	0	1
Region	7	6	5	4	3	2	1	0

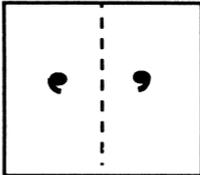
The following list shows the pattern of the switches of some of the different symmetries:



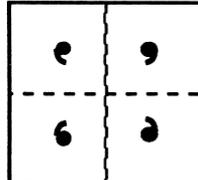
0010 0010



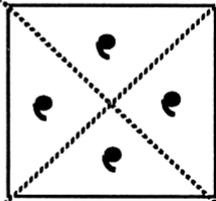
0100 0010



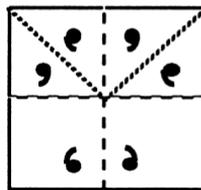
0010 0100



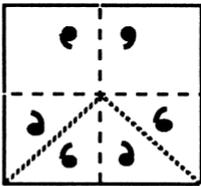
1010 0101 or 0101 1010



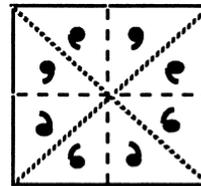
1111 0000 or 0000 1111



1101 1011



0111 1110



1111 1111

Once the parameter settings are the way you want them, all you have to do is press the <ESC> key to see your kaleidoscope. It is possible to change the kaleidoscope while it is running. For example, if the speed is set to \$20 and you want it to go faster, all you do is to type (remember the leading "0") <0>, then <1>, followed by a <CTRL> <R>. The kaleidoscope will now continue, but at a faster rate. All of the parameters can be changed while the kaleidoscope is running. To get back to the PARAMETERS MENU, type <CTRL> <P>. The PARAMETERS MENU will reflect any changes that you made while the kaleidoscope was running.

To leave KALEIDOMAKER1 and return to the MAIN MENU, type <CTRL> <RESET>.

Changing the Parameters in KALEIDOMAKER2

The KALEIDOMAKER2 PARAMETER MENU looks very similar to the KALEIDOMAKER1 PARAMETER MENU with a couple of differences.

The first obvious difference is that instead of a number after the Graphics Mode there is the word SUPER HI-RES. When using KALEIDOMAKER2 you can not switch GRAPHICS modes. The program is always in Super Hi-Res. The other difference is in the QUE. In KALEIDOMAKER1 the QUE was defined by a three digit hexadecimal number. In KALEIDOMAKER2, the QUE can go up to values of \$1FFF. If you try to put in a value higher than \$1FFF, you will get an error beep.

Refer to Appendix 2 for a complete listing of all the values available for each of the parameters.

The Parameters in KALEIDOMAKER1 and KALEIDOMAKER2

The following is a review of all the parameters in the KALEIDOMAKER Programs.

- **GRAPHICS Mode**

A GRAPHICS Mode is defined by the total number of pixels that are available on the screen and the number of colors that can be displayed. There are five different GRAPHICS Modes that the Apple IIGS can display. The Apple II+ can only display two of these modes, Lo-Res and Hi-Res. The Apple IIe with a 80 column card with extended graphics and the Apple IIc can also display Double Lo-Res and Double Hi-Res. (It is possible to upgrade the Apple IIe to display Super Hi-Res.)

Using KALEIDOMAKER1, it is possible to create kaleidoscopes using Lo-Res (GRAPHICS 1), Hi-Res (GRAPHICS2), Double Hi-Res (GRAPHICS3) and Double Lo-Res (GRAPHICS4). Using KALEIDOMAKER2, the GRAPHICS mode is set to Super Hi-Res and there is no way to change the graphics mode.

- **RATE (or Speed)**

The RATE determines how fast the Kaleidoscope will run. There are 255 (\$01 to \$FF) different speeds available. The rate can be changed on both KALEIDOMAKER programs. When the kaleidoscope is set to Lo-Res or Double Lo-Res, it appears that it is running at a faster rate. The most important reason for this is that the pixels are larger.

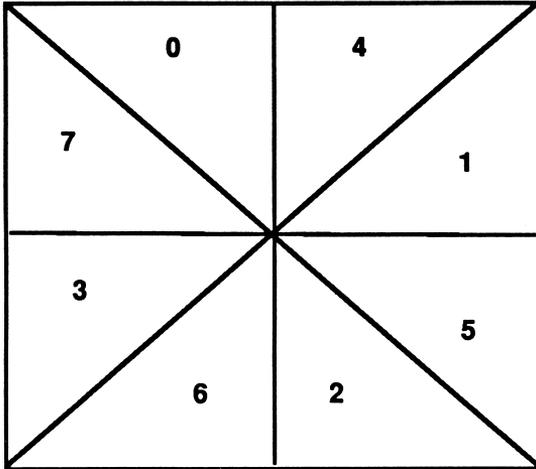
- **ACTION**

ACTION is the parameter that determines how the Kaleidoscope pattern moves from the most recently plotted point to the next point to be plotted. The four ACTIONs include:

- **ACTION 0-** This action is controlled by the movement of the joystick. Whatever actions you take with the joystick will be reflected according to the set symmetry.
 - **ACTION 1-** Using this action the next dot (pixel) to be plotted will be selected from one of the four horizontal and vertical neighbors.
 - **ACTION 2-** Using this action the next dot (pixel) to be plotted will be selected from one of the four diagonal neighbors.
 - **ACTION 3-** Using this action the next dot (pixel) to be plotted will be selected from either the horizontal, vertical or diagonal neighbors.
- **QUE**
If the QUE Size is small then the Kaleidoscope will generate a very sparse design. If the QUE Size is large then the design will be very dense. Remember that the pixels in Lo-Res (and Double Lo-Res) are much larger than the pixels in Hi-Res therefore if the QUE Size is 100 it will look very different in Lo-Res than in higher resolutions.

The QUE in KALEIDOMAKER1 can contain numbers from \$001 to \$FFF. Numbers beyond \$800 will create an interesting effect. In KALEIDOMAKER2, the QUE can range in values from \$0001 to \$1FFF. If you put in a value that is over \$1FFF, you will get an error beep.

- **SYMMETRY**
The screen is divided into eight different regions. If all of the regions are switched on, then the program repeats the unit design eight different times.



The regions have been given numbers. The unit pattern in regions 1, 3, 5, and 7 are rotated. The unit patterns in regions 0, 2, 4, and 6, are also rotated, but they are mirror reflections of the unit pattern 1,3, 5, and 7.

- **COLOR**

Each of the five graphics modes has a set of colors that are available. The Reference Card contains color tables for each of the different GRAPHICs Modes.

- **DURATION**

DURATION determines how many pixels will be plotted in a particular color before the program starts to plot in a different color. It is possible to have a **DURATION** of 1 to 255 (\$1 to \$FF). If the **DURATION** is 1 then the program will plot one color and then select a new color. If the **DURATION** is set at 100 then the program will plot 100 pixels in the current color and then go on to another color.

Chapter 4

Creating your own Computer Kaleidoscope

In this section you will learn to write kaleidoscope programs in AppleSoft BASIC. The programs in this section will incorporate many of the concepts that were explored in the last two sections.

There are no mirrors in a computer so the program must simulate, or pretend, that there are mirrors. It turns out that this is not a very difficult thing to do. The following is an AppleSoft BASIC program that produces mirror reflections of a point X,Y and places it on the Lo-Res graphics screen:

Program #1

```

10 HOME : GR
20 X = 3: Y = 5: Z = 13
30 GOSUB 1000
40 END
1000 COLOR = Z
2000 PLOT X,Y
2010 PLOT 39 - X, Y
2020 PLOT X, 39 - Y
2030 PLOT 39 - X, 39 - Y
2100 PLOT Y, X
2110 PLOT 39 - Y, X
2120 PLOT Y, 39 - X
2130 PLOT 39 - Y, 39 - X
2140 RETURN

```

Notice that the program consists of a very short main section (lines 10 through 40), followed by the subroutine for plotting the eight symmetry points for the specified X and Y coordinate. This subroutine, which starts with line number 1000, has three calling parameters. They are: X, Y, and Z. Parameters X and Y specify the coordinates of the "principal point", and Z specifies

the color in which the point will be plotted. The rest of the subroutine does nothing more than plots all the symmetry points and then returns back to the main program. You might find it instructive to use the grid paper and pretend that you are a computer and follow the steps in the program. Often times, the best way to understand something is to carry out the steps yourself the way a computer would.

Notice that lines 2000 through 2030 are identical to lines 2100 through 2130, except that the occurrences of X and Y are exchanged. This suggests that you might shorten our subroutine to only four plot statements. You could then invoke this four plot statement subroutine, swap the parameters X and Y, and then call the subroutine again. This could be accomplished with the following program:

Program #2

```

10 HOME : GR
20 X = 3: Y = 5: Z = 13
30 GOSUB 1000
40 END
1000 COLOR = Z
1010 GOSUB 2000
1020 GOSUB 2000
1030 RETURN
2000 PLOT X,Y
2010 PLOT 39 - X, Y
2020 PLOT X, 39 - Y
2030 PLOT 39 - X, 39 - Y
2040 T = X: X = T: Y = T
2050 RETURN

```

In the above program, notice that in line 2040 the values of X and Y are exchanged just before it returns. Therefore, when the subroutine is first invoked (in line 1010) four of the eight symmetry points are plotted. Then, in line 2040, X and Y coordinates are exchanged. When the subroutine is again invoked (in line 1020) the values of X and Y will have just been exchanged as a result of the previous GOSUB. Therefore, line 1020 plots the remaining four symmetry points before returning

the values of X and Y to their original states. Finally, in line 1030, the flow returns back to the calling program. But this program can still be further simplified. Notice that lines 1020 and 1030 consist of a GOSUB followed by a RETURN.

Whenever such a combination occurs, the two statements can be replaced by a single GOTO statement (which will allow the RETURN statement in line 2050 to be used instead of the RETURN statement in line 1030). However, the statement the program would be going to is simply 2000, the very next statement in the listing. Therefore, the program doesn't even need the GOTO statement! The following is the final compact version of the program:

Program #3

```

10 HOME : GR
20 X = 3: Y = 5: Z = 13
30 GOSUB 1000
40 END
1000 COLOR = Z
1010 GOSUB 2000
2000 PLOT X,Y
2010 PLOT 39 - X, Y
2020 PLOT X, 39 - Y
2030 PLOT 39 - X, 39 - Y
2040 T = X: X = T: Y = T
2050 RETURN

```

The main program consists of lines 10 through 40 just as before. In line 30 the program executes a GOSUB 1000 to the subroutine located at 1000. The first thing that subroutine 1000 does is set the COLOR equal to the value specified by the parameter Z. Next, it performs a GOSUB to another subroutine at 2000 (to plot the four major symmetry points). Before subroutine 2000 returns it swaps the values of X and Y. It then returns to the statement immediately following the GOSUB statement that called it. But this statement is just the beginning of the subroutine itself! The subroutine, therefore, plots the remaining four symmetry points, swaps X and Y back to their original values, and executes a RETURN. But to which statement does it return to? At this point the program returns to the statement following line 30 since line

30 contains the last referenced GOSUB.

The previous program plotted eight symmetrical stationary points. To create a "moving" kaleidoscope, you will need some systematic way of "moving" the points. There are many different algorithms that can be used to create movement depending on the effect you wish to produce. One simple way would be to replace lines 30 and 40. The program will then look like this:

Program #4

```

10 HOME: GR
20 X = 3: Y = 5: Z = 13
100 X = X + 1: IF X > 39 THEN X = 0
110 Y = Y + INT(2 * RND (1)): IF Y > 39
    THEN Y = 0
120 Z = Z + 1:
130 GOSUB 1000
140 GOTO 100
1000 COLOR = Z
1010 GOSUB 2000
2000 PLOT X,Y
2010 PLOT 39 - X, Y
2020 PLOT X, 39 - Y
2030 PLOT 39 - X, 39 - Y
2040 T = X: X = T: Y = T
2050 RETURN

```

Line 100 causes the X parameter to increment each time through the loop. The IF statement simply keeps the plotted values of X from exceeding the right hand side of the screen. Line 110 has the effect of causing Y to increment by either 0 or 1, randomly. Then the following IF statement on that line keeps the plotted values of Y from exceeding the bounds of the screen.

The kaleidoscope that this program produces, while somewhat interesting, suffers from some real problems. One of the problems is that after the program runs for a period of time, the whole screen gets filled up with color pixels and it becomes hard to see the kaleidoscope very well. In other words, it starts off

looking rather interesting when the screen is mostly black. But, as the screen starts to fill up, you get lost in the confusion of colored pixels. The second problem, which is really related to the first one, is the fact that the screen consists of randomly scattered pixels as opposed to colored regions (or something with a structure to it). To keep the color of the pixels from changing too often, the program must have a way of maintaining a pixels color for a period of time before it changes.

In addition, to keep the screen from becoming completely filled up with pixels, there needs to be some scheme for erasing what has previously been plotted so that the screen doesn't become too "busy". This would also help maintain a good proportion of colored pixels (foreground) to black pixels (background). One way to accomplish this is to keep track of all the points that have been plotted prior to the current time. Every time a new point is plotted, its coordinates are put into a que so that later the program will know which points should be erased. (As you recall, a que is a data structure in which new information is inserted at one end while old information is extracted from the other end.) Normally, one thinks of a que as being a straight line of objects with one end of the line representing the front of the que and other end representing the rear. However, if the que is bent around into the shape of a circle the beginning and end of the que will both have the same boundary. Now, instead of having two separate que pointers (one to the beginning, and one to the end), only one pointer is needed to indicate the boundary between the beginning and the end.

To apply these concepts of ques to kaleidoscopes look at the following program listing:

Program #5

```

10 HOME : GR
20 QS = 75
30 DIM XQUE (QS), YQUE (QS)
40 FOR K = 1 TO QS
50 XQUE (K) = 0: YQUE (K) = 0
60 NEXT K
70 XX = 0: YY = 0: ZZ = 8: QPT = 1

```

```

100 X = XQUE (QPT): Y = YQUE (QPT)
110 Z = 0: GOSUB 1000
150 X = XX: Y = YY
160 Z = ZZ / 8: GOSUB 1000
170 XQUE (QPT) = XX: YQUE (QPT) = YY
180 QPT = QPT + 1: IF QPT > QS THEN QP
    = 1
200 XX = XX + 1: IF XX > 39 THEN XX = 0
210 YY = YY + INT (2 * RND (1)): IF YY
    > 39 THEN YY = 0
220 ZZ = ZZ + 1: IF ZZ > 127 THEN ZZ = 8
250 GOTO 100
1000 COLOR = Z
1010 GOSUB 2000
2000 PLOT X,Y
2010 PLOT 39 - X, Y
2020 PLOT X, 39 - Y
2030 PLOT 39 - X, 39 - Y
2040 T = X: X = T: Y = T
2050 RETURN

```

Notice that the same symmetrical point plotting subroutine (at 1000) has been incorporated as in the other kaleidoscope programs. Lines 10 through 250 represent the new part of the program. Line 20 in the program sets the que size to 75. This means it will plot 75 new points before it starts to erase the old ones. (You might try changing the value of this parameter to see what effect it produces.)

Initially, the contents of both of the ques are set to zero (within the FOR NEXT loop located at lines 40 through 60). In line 70, XX and YY are initialized to zero. (XX and YY represent the running values of the X and Y coordinates.) In addition, ZZ is set equal to 8. ZZ will be used later to select the color to be plotted. On that same line, QPT is set equal to zero. QPT is the que pointer which points to the boundary between the beginning and the end of the que. This program uses a data structure known as a "circular que", so that the beginning of the que and the end of the que meet like a donut.

The main loop of the program starts in line 100. Here, we extract the oldest points from the que and return them to the X and Y parameters. In line 110 we then set Z equal to zero (the value for the color black), and invoke the subroutine at 1000 for plotting the eight symmetry points (i.e. we erase the point). In line 150, we select the running parameters XX and YY to specify the next point to be plotted. In line 160 we divide ZZ by 8 and use the result as the new color. (This keeps the color constant for eight consecutive pixels.)

In line 170, we insert the running parameters XX and YY into the que. Then, in line 180 we increment the que pointer and make sure that we haven't exceeded the value specified by QS. In line 200 and 210, we increment the running parameters XX and YY, and make sure that they don't exceed the screen boundary. In line 220 we increment the color specifier ZZ (which as you recall in line 160 was first divided by eight in order to specify the actual color). Finally, in line 250, we simply jump back to the beginning of the loop at line 100.

A Hi-Res Kaleidoscope

The next program, **LINE KALEIDOSCOPE**, uses high resolution graphics. Again, there are two ques, XQ and YQ, for storing coordinates that were plotted on the Hi-Res screen so that they can later be erased, as in the Lo-Res example. The main differences between the this program and Lo-Res program is that, instead of mirroring points the program mirrors line segments.

This kaleidoscope looks very different when it is running, but you will notice that the program itself is not that different than the last program. The following is the listing:

Program #6

```

10 N = 10
20 DIM XQ (N), YQ (N)
30 FOR K = 1 TO N
40 XQ (K) = 0: YQ (K) = 0

```

```
50 NEXT K
60 IN = 1: OUT = 3
70 X = 0: Y = 0
80 HOME: HGR2
100 X = X + 50 * RND (1) - 25
110 Y = Y + 50 * RND (1) - 25
120 IF X < 0 THEN X = - X
130 IF Y < 0 THEN Y = - Y
140 IF X > 95 THEN X = 190 - X
150 IF Y > 95 THEN Y = 190 - Y
200 XQ (IN) = X: YQ (IN) = Y
210 Z = IN: HCOLOR = 3: GOSUB 1000
220 Z = OUT: HCOLOR = 0: GOSUB 1000
300 IN = IN + 1: IF IN > N THEN IN = 1
310 OUT = OUT + 1: IF OUT > N THEN
    OUT = 1
500 GOTO 100
1000 ZZ = Z - 1: IF ZZ < 1 THEN ZZ = N
1010 HPLOT 140 + XQ (ZZ), 96 + YQ (ZZ)
    TO 140 + XQ (Z), 96 + YQ (Z)
1020 HPLOT 140 - XQ (ZZ), 96 + YQ (ZZ)
    TO 140 - XQ (Z), 96 + YQ (Z)
1030 HPLOT 140 - XQ (ZZ), 96 - YQ (ZZ)
    TO 140 - XQ (Z), 96 - YQ (Z)
1040 HPLOT 140 + XQ (ZZ), 96 - YQ (ZZ)
    TO 140 + XQ (Z), 96 - YQ (Z)
1050 HPLOT 140 + YQ (ZZ), 96 + XQ (ZZ)
    TO 140 + YQ (Z), 96 + XQ (Z)
1060 HPLOT 140 - YQ (ZZ), 96 + XQ (ZZ)
    TO 140 - YQ (Z), 96 + XQ (Z)
1070 HPLOT 140 - YQ (ZZ), 96 - XQ (ZZ)
    TO 140 - YQ (Z), 96 - XQ (Z)
1080 HPLOT 140 + YQ (ZZ), 96 - XQ (ZZ)
    TO 140 + YQ (Z), 96 - XQ (Z)
1100 RETURN
```

Appendix #1

Answers to Kaleidoscope Questions

1. The <A> kaleidoscope moves much faster than the kaleidoscope.
2. The pixels that are used in the <C> Kaleidoscope are bigger (Lo-Res) while the pixels that that are used in the <D> Kaleidoscope are smaller (Hi-Res). There are actually the same number of pixels on the screen for both of these kaleidoscopes.
3. In the <E> Kaleidoscope the unit design is repeated twice.
4. The <F> Kaleidoscope the unit design is repeated four times.
5. The <G> Kaleidoscope the unit design is repeated twice.
6. The <H> Kaleidoscope the unit design is repeated eight times.
7. The symmetry of the <I> Kaleidoscope is called rotational symmetry.
8. The <J> Kaleidoscope eventually contains more pixels than the <K> Kaleidoscope.
9. The pixel disappears.
10. New pixels are being added while old pixels are being erased.
11. The pattern created by the pixels on the <N> Kaleidoscope looks like a checkerboard pattern.
12. The colors are different. The <P> Kaleidoscope contains a lot more colors than the <O> Kaleidoscope.
13. The colors in the <Q> Kaleidoscope cluster together, while in the <R> Kaleidoscope, the colors look more random.

Appendix #2

KALEIDOMAKERS Reference

GRAPICHS Mode

In KALEIDOMAKER1, select the GRAPHICs mode you desire.

<1> = Lo-Res <3> = Double Hi-Res
<2> = Hi-Res <4> = Double Lo-Res

Press the corresponding number.
Then press <CTRL> <G>

In KALEIDOMAKER2, the GRAPHICS Mode is set to Super Hi-Res and can not be changed.

RATE or Speed

Select a RATE between \$1 - \$FF. Where \$1 is the fastest and \$FF (255) is the slowest..

Then press <CTRL> <R>

ACTION or Movement

Select the ACTION you desire:

<0> = To be used with a Joy Stick
<1> = Selects only horizontal and vertical neighbors
<2> = Selects only diagonal neighbors
<3> = Selects either horizontal, vertical or diagonal neighbors.

Press the number key corresponding to that action.
Then press <CTRL> <A>

QUE Size

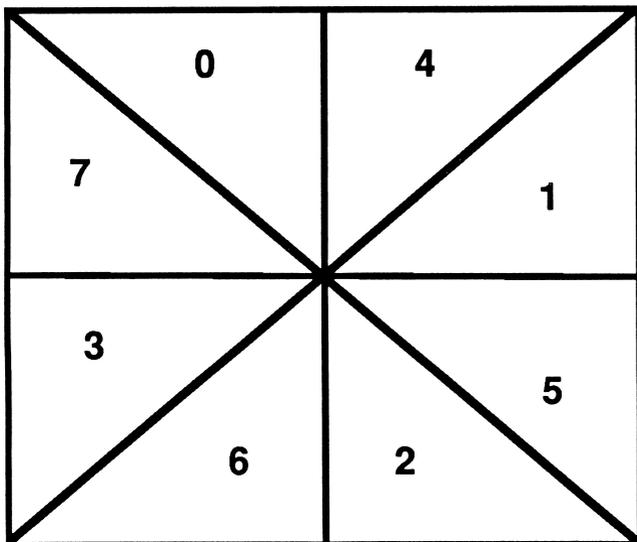
In KALEIDOMAKER1, select a QUE Size between \$001 and \$FFF. Any size over \$800 will give an interesting effect.

In KALEIDOMAKER2, select a QUE Size between \$0001 and \$1FFF.

Then press <CTRL> <Q>

SYMMETRY

The following diagram shows the eight regions:



The following chart shows how the regions relate to the SYMMETRY positions on the PARAMETERS Menu:

SYMMETRY	1	0	1	0	0	1	1	0
Region	7	6	5	4	3	2	1	0

If there is a "1" in the position, then that region is switched "on".
If there is a "0" in the position then that region is switched "off".

Select the area you want to switch. Press the number key that relates to that area and then press <CTRL> <S>.

On page 19 there are formulas for some of the common symmetries.

COLORS

On the PARAMETERS MENU next to the COLOR are a series of sixteen ones and zeros divided up into four groups. These are switches, and they represent the available colors in the different GRAPHICS Modes. If the switch is turned "on", has a value of one, then that color is switched on. If the switch is turned "off", has a value of zero, then that color is switched off.

The following chart shows what value each of the positions on the COLOR Parameter indicates.

COLOR	1	0	0	1	0	0	0	0	1	1	0	0	1	1	1	0
POSITION	\$F	\$E	\$D	\$C	\$B	\$A	\$9	\$8	\$7	\$6	\$5	\$4	\$3	\$2	\$1	\$0

Below and on the next page are COLOR Charts for all of the different GRAPHICS Modes.

To switch a particular color "on" or "off":

- Find the GRAPHICS Mode,
- Find the color on the corresponding COLOR Chart ,
- Type the hexadecimal number corresponding to that color and then type <CTRL> <C>.

COLOR Charts for GRAPHICS Modes

LoRes and Double LoRes	
\$0 = Black	\$8 = Brown
\$1 = Magenta	\$9 = Orange
\$2 = Dark Blue	\$A = Gray 2
\$3 = Purple	\$B = Pink
\$4 = Dark Green	\$C = Light Green
\$5 = Gray 1	\$D = Yellow
\$6 = Medium Blue	\$E = Aquamarine
\$7 = Light Blue	\$F = White

COLOR Charts for GRAPHICS Modes

HiRes	
\$0 = Black 1	\$8 = Black 1
\$1 = Green	\$9 = Green
\$2 = Violet	\$A = Violet
\$3 = White 1	\$B = White 1
\$4 = Black 2	\$C = Black 2
\$5 = Orange	\$D = Orange
\$6 = Blue	\$E = Blue
\$7 = White 2	\$F = White 2

Double HiRes	
\$0 = Black	\$8 = Red
\$1 = Blue	\$9 = Magenta
\$2 = Green	\$A = Dark Gray
\$3 = Purple	\$B = Light Blue
\$4 = Brown	\$C = Orange
\$5 = Gray	\$D = Light Orange
\$6 = Bright Green	\$E = Yellow
\$7 = Light Green	\$F = White

Super Hi-Res	
\$0 = Black	\$8 = Light Blue
\$1 = Gray	\$9 = Blue
\$2 = Red	\$A = Dark Blue
\$3 = Brown	\$B = Purple
\$4 = Mustard	\$C = Pink
\$5 = Yellow	\$D = Magenta
\$6 = Dark Green	\$E = White
\$7 = Bright Green	\$F = Light Gray

DURATION

DURATION determines how many pixels will be plotted in a particular color before the program starts to plot in a different color. The values can range between \$01 to \$FF.

To change the DURATION, type in the number you desire, and then press <CTRL> <D>.

To stop the kaleidoscope press the SPACE BAR. Pressing the SPACE BAR again will restart the kaleidoscope.

To return to the PARAMETERS MENU, press <CTRL> <P>.

From the PARAMETERS MENU, to start the kaleidoscope press the <ESC> Key.

To EXIT KALEIDOMAKERS, press <CTRL> <RESET>.

Index

ACTION	10, 14, 16, 22
AppleSoft BASIC	6, 25
Base 10	16, 25
Base 16	16
Binary Numbers	145
COLOR	15, 16, 18, 19
DOILIES	5, 11, 12
Double Hi-Res	14, 16, 21
Double Lo-Res	15, 17, 18, 22
DURATION	15, 17, 24
ELECTRIC	5, 11, 12, 13
GRAPHICS Mode	9, 14, 17, 21
Hexadecimal Numbers	15, 16
Hi-Res	9, 15, 21
KALEIDOKEY	5, 7, 11, 12
KALEIDOKEY2	11, 12
KALEIDOMAKER	5, 10, 16, 17
KALEIDOMAKER1	6, 15, 17, 21, 22, 23
KALEIDOMAKER2	6, 15, 21, 23
KALEIDOMUSIC	5, 11
KALEIDOSCOPIES	5, 7, 8, 10
KALEIDOTWISTER	5, 11
Lo-Res	9, 15, 17, 18, 21
MAIN MENU	5, 15, 17
Mirror Reflections	2, 3, 24
Parameters	8, 17, 19, 20, 21, 25
Que (QUE)	9, 16, 17, 21, 23, 29
RATE	15, 17, 22
Rotation	3, 24
SERENITY	5, 11, 12, 13
SOUTHWEST	5, 11
SpaceLace Diskette	5
Super Hi-Res	15, 21, 22
Switches	18, 19, 20
Subroutine	25, 26
Symmetry (SYMMETRY)	4, 8, 9, 15, 17, 19, 20, 23
Symmetry Regions	19, 20, 23

Traditional Kaleidoscope	1
Unit Design	3, 7, 8, 24
WARMTH	6, 11, 12, 13



Great Wave
Software™

5353 Scotts Valley Drive
Scotts Valley, CA 95066
(408) 438-1990