CHAPTER 14
EXTRA RESOURCES

Additional Resources

1. "While Loop Statement" (*http://tiny.cc/whileloop/*): See the Control Statements reference documentation for more context.
2. "Three Loop Counters" (*http://tiny.cc/loopcounters/*): Compare the different ways you can construct a loop counter.
3. "Three Ways to Validate Input" (*http://tiny.cc/validateinput/*): Read more about all three ways that you can check whether the input is what you're asking for.
4. "Example Loop" (*http://tiny.cc/exampleloop/*): Follow this example `While` loop for more context.
5. "The 14 Keywords" (*http://tiny.cc/14keywords/*): Read this contextual list of all 14 keywords.
6. GetRandomNumber Method (*http://tiny.cc/getrandomnumber/*): See this method explained in the context of all the `Math` object's methods.

7. "Wood Chuck: How Much Wood Could a Woodchuck Chuck?" (*http://tiny.cc/woodchuck/*): Check out the code from Try It Out 14-1, and share how you improved the program.
8. "Enter the Sentinel" (*http://tiny.cc/sentinel/*): Learn how to use a sentinel to control and end a loop.
9. GraphicsWindow Events (*http://tiny.cc/graphicswindowevents/*): See the `MouseDown` method in context with the other `GraphicsWindow` events.
10. "Event Basics" (*http://tiny.cc/eventbasics/*): Build on what you learned in Chapter 11 with more event examples.

Review Questions

1. What do you call the statements enclosed between the `While` and `EndWhile` keywords?
2. Which loop should you use when you don't know the number of repetitions in advance?
3. When a `While` loop checks the condition, what happens if the condition is true? What happens if the condition isn't true?
4. What does it mean to validate your user's input?
5. What is an example of a program in which you would want to use an infinite loop?
6. What kind of statement can you use to instantly exit a deeply nested loop?

Practice Exercises

1. What is the output of the following code?

```
count = 0
While (count < 100)
  TextWindow.WriteLine(count)
  count = count + 1
EndWhile
```

2. What is the output of the next code block?

```
n = 9
While (n  > 1)
  TextWindow.WriteLine(n + ", " + n * n)
  n = n -2
EndWhile
```

3. What does the following program do?

```
number = TextWindow.ReadNumber()
max = number
While (number <> 0)
  number = TextWindow.ReadNumber()
  If (number > max) Then
    max = number
  EndIf
EndWhile
TextWindow.WriteLine("max is " + max)
```

4. This program is supposed to display all the odd integers from 1 to 15. Correct the error in this program:

```
' Displays odd integers from 1 to 15
While (n  <= 15)
  n = 1
  TextWindow.WriteLine(n)
  n = n + 2
EndWhile
```

5. Bill Gates wants to hire you to develop the latest cloud services. You start earning a penny on the first day, but your salary doubles each day. Write a program to find out how many days you have to work to earn at least a million dollars. Here is an example of the program's output:

```
Day     Salary($)      Earnings($)
----    ---------      -----------
1       0.01           0.01
2       0.02           0.03
3       0.04           0.07
4       0.08           0.15
...
```

6. A principal amount of $100 is deposited with an annual interest rate of 5% compounded quarterly. Write a program to find how many years it will take to triple the deposited principal.

7. Three sailors were trapped on a desert island. On the first day, the sailors collected a pile of coconuts and agreed to divide it among them early the next day. During the night, the first sailor woke up, divided the pile into three equal parts, and found one coconut left over. He hid his share (putting the other sailors' shares back into a single pile), gave the leftover to a monkey, and went back to sleep. The second sailor then woke up and did the same thing as the first sailor. The third sailor then repeated it as well. In the morning, the three sailors divided the pile into three equal parts and found one left over, which they gave to a monkey. Write a program to find the smallest integer that can represent

the number of coconuts in the original pile. (Hint: try different integers inside a `While` loop, and stop when you find the first integer that satisfies the described division.)

8. An aging snail is trying to cross to the other side of a 4-meter road. It moves one meter on day 1, 1/2 meter on day 2, 1/3 meter on day 3, 1/4 on day 4, and so on. How many days will it take the snail to cross the road? (Hint: find the sum 1 + 1/2 + 1/3 + 1/4 + . . . until the sum reaches (or exceeds) 4.)

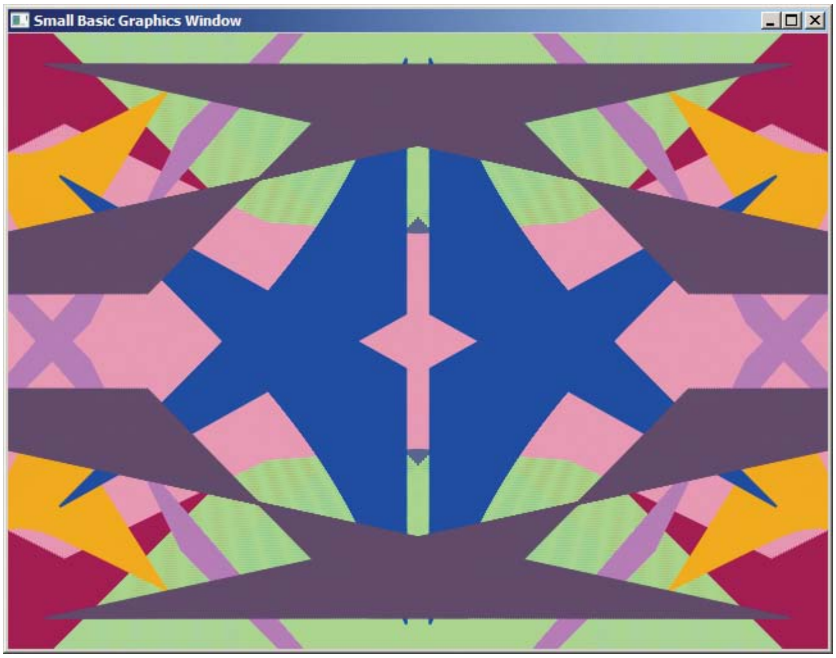
9. A conjecture is a statement that might be true but hasn't been proven. Test this simple conjecture: start with any positive integer, and if it's even, divide it by 2; otherwise, multiply it by 3 and add 1 to the result. If you continue to repeat this process, the final result should always equal 1. Here's an example that starts with 11:

$$\begin{tikzcd}
\boxed{11} \times 3 + 1 = 34 \arrow[d] \\
34 \div 2 = 17 \arrow[d] \\
17 \times 3 + 1 = 52 \arrow[d] \\
52 \div 2 = 26 \arrow[d] \\
26 \div 2 = 13 \arrow[d] \\
13 \times 3 + 1 = 40 \arrow[d] \\
40 \div 2 = 20 \arrow[d] \\
20 \div 2 = 10 \arrow[d] \\
10 \div 2 = 5 \arrow[d] \\
5 \times 3 + 1 = 16 \arrow[d] \\
16 \div 2 = 8 \arrow[d] \\
8 \div 2 = 4 \arrow[d] \\
4 \div 2 = 2 \arrow[d] \\
2 \div 2 = \boxed{1}
\end{tikzcd}$$

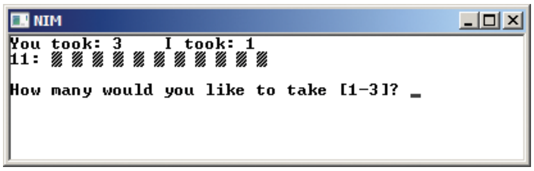
Write a program that reads the starting integer from the user and then shows how this integer converges to 1.

10. Open the file *EtchASketch.sb* from this chapter's folder. Use the arrow keys to draw on the screen. Brainstorm some ways to improve this game, and then see if you can code your suggestions.

11. Open the file *Kaleidoscope.sb* from this chapter's folder and run it. The following is a sample output of the program. Study the program and explain how it works.



12. Open *Nim.sb* from this chapter's folder. This program lets a user play a game of Nim against the computer. When the game starts, it selects a number of objects (between 15 and 23) at random. The player and the computer then take turns removing 1, 2, or 3 objects. The player who takes the last object loses. The following figure shows the game.



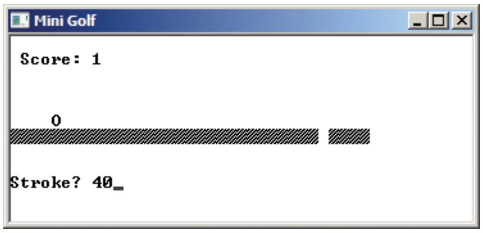
```
You took: 3    I took: 1
11: ▨ ▨ ▨ ▨ ▨ ▨ ▨ ▨ ▨ ▨ ▨

How many would you like to take [1-3]? _
```

Play it several times to understand how it works. Change the game so it asks the user if they want to play another round at the end of the game.

13. Open the file *MemoryTest.sb* from this chapter's folder. The program implements a simple memory test. It displays a random number for a short time and then asks you to enter that number. The number gets larger (that is, more digits are added) with each round of the game. The display time also gets slightly longer to give you more time to remember the number. Run the program to see how it works. Find some ways to improve this game.

14. If you like golf, here's your chance to practice! Open the file *MiniGolf.sb* from this chapter's folder. Play the game and read the code to understand how it works. The following figure shows this game in action.



```
Score: 1


    o
▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨▨ ▨▨▨▨

Stroke? 40_
```

The game ends if the player overshoots the hole. Update the code so the player can move the ball in the opposite direction when they overshoot.

15. Do you think you can read a computer's mind? Open the file *Predict.sb* from this chapter's folder and play the game. The computer asks you to predict the next number (between 1 and 9) it will generate. The smaller your score the better. Think of some ways to improve the game and try to make those changes. The game ends when you correctly guess the computer's number. Your score is how many guesses you took to reach the correct answer.