

CHAPTER 12

EXTRA RESOURCES

Additional Resources

1. “Small Basic Reference Documentation: Controls Object” (<http://tiny.cc/controlsubject/>): Peruse the reference documentation to learn more about the Controls object.
2. “Small Basic: Controls” (<http://tiny.cc/sbcontrols/>): Browse some examples of different controls built with Small Basic.
3. “Controls Are Shapes” (<http://tiny.cc/areshapes/>): Check out this fun example of playing with your controls with the Shapes object.
4. “Share Your Simulation!” (<http://tiny.cc/sharesimulation/>): Share the simulation you made in Try It Out 12-3 with the community!
5. “Small Basic Reference Documentation: Flickr Object” (<http://tiny.cc/flickrobject/>): Read the Flickr object’s reference documentation.
6. Flickr Tutorial (<http://tiny.cc/flickrtutorial/>): Use this tutorial to get more comfortable with Flickr.

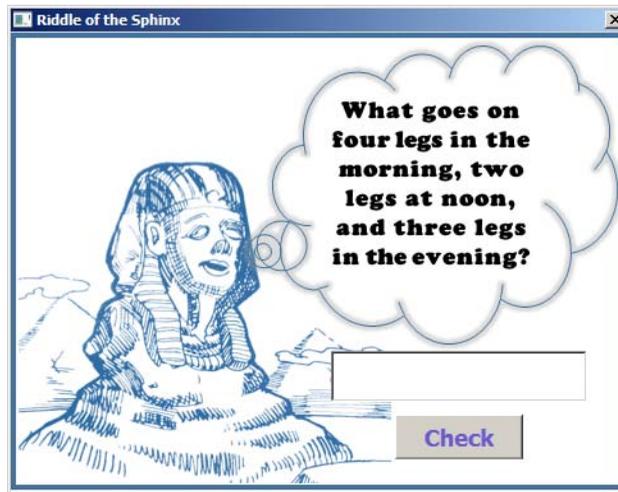
Review Questions

1. What does GUI stand for, and what are the features of a GUI?
2. What are five methods of the Controls object, and what do they do?
3. What's the difference between a text box and a multiline text box?
4. When you have multiple buttons, how can you tell which button a user clicked?
5. How does the Flickr object work?

Practice Exercises

1. The Sphinx (in Greek mythology) was a creature who possessed the head of a woman, body of a lion, and wings of an eagle. This creature guarded the gates of Thebes, challenging anyone attempting to enter with a riddle.

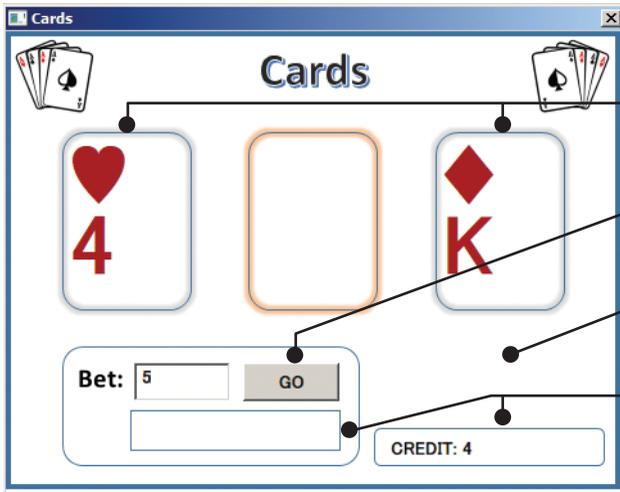
Open the file *Sphinx_Incomplete.sb* in this chapter's folder. When you run the application, you'll see the following interface.



Complete the program so that it check the user's answer when they click the Check button. Use the `ShowMessage()` method to tell the user their destiny. Accept human, man, or woman as correct answers to the riddle.

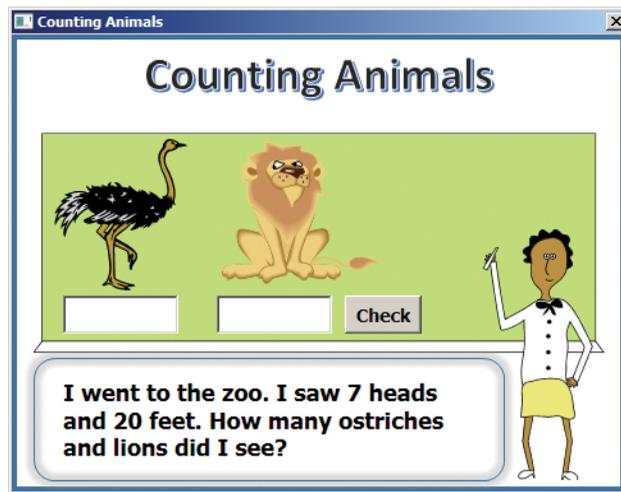
2. This program implements a card game in which three cards are dealt; one is face down (hidden). A player then bets on whether the value of the hidden card is between the values of the other two (see the following figure). If the value of the hidden card is between the other two, the player wins the bet; otherwise, they lose the bet. To make a bet, they enter the bet amount, and then click the GO button. Valid bets are 0, 5,

10, and 20. If the player bets 0, they lose \$2 and the game automatically starts another round. Open the file *Cards.sb* in this chapter's folder and play the game to understand how it works. Then study the code and come up with some ideas to improve the game.



- 1 The game shows two cards drawn at random.
- 2 The player makes his bet and clicks GO.
- 3 The DEAL button appears after each round. The player clicks the DEAL button to start another round.
- 4 The game shows the result of this round and the player's total credit. It then hides the GO button and shows the DEAL button.

3. Create a program that asks the user to solve math puzzles. Open the file *CountingAnimals_Incomplete.sb* in this chapter's folder. When you start the application, a window similar to the following figure appears (the numbers in the problem are selected at random).



Complete the application so it checks the user's answers when they click the Check button. Give the user a feedback message using the `ShowMessage()` method. If they guessed correctly, show them a new problem.

4. Use the *Dice_Incomplete.sb* file in this chapter's folder to create a dice game (see the following figure). A player bets on a number from 1 to 6, enters his bet amount, and then clicks the ROLL button. The computer rolls three dice. The player wins the amount of his bet for each match or loses his bet if there were no matches. Follow the comments shown in the application's source code to write the missing code and complete the application.



5. Write a horse race game (see the following figure). Open the file *HorseRace_Incomplete.sb* in this chapter's folder and follow the comments shown in the application's source code to write the missing code and complete the application.



6. Turtle Boy needs to learn how to type. So his dad, Turtle Man, wants you to create a program that teaches Turtle Boy how to type! Display a random letter in a text box, and then provide a text box for Turtle Boy to enter the matching letter. Add a button for him to click to move on to the next letter. After ten letters, end the program and let Turtle Boy know how many letters he got right! Make sure your UI text explains how to use your program. For an example of how to make this game, go to tiny.cc/typinggame. Leave a comment there with a link to your program (to get your URL, click **Publish** in the Toolbar). See if you can improve on Ray's version.

