PROJECT **1.1.6**

Chatting with Magpie

INTRODUCTION

The Magpie is a species of bird well known for being able to mimic sounds, such as ring tones, whistling, and even human speech. For this project, you will experiment with a Magpie program in BlueJ and Android™ that mimics simple human conversation. Programs such as these are called "chatbots". While your Magpie program is a very simple chatbot, more complex and sophisticated chatbots can appear to be nearly human in their ability to hold conversations.

Materials

- Computer with BlueJ and Android™ Studio
- Android™ tablet and USB cable, or a device emulator

RESOURCES

Lesson 1.1 Reference Card for Basic Constructs
Resources available online

AP Computer Science: A Magpie Chatbot Lab Student Guide
Resources available online

Procedure

Part I: The Magpie Chatbot

The College Board provides programming labs for students taking AP CS-A. To get started with this project, you will interact with the College Board's Magpie Chatbot lab to learn how it behaves, or how its algorithm works. Then, you will change the chatbot's behavior by writing new Java code in BlueJ and Android Studio.

1. Complete Magpie Student Guide *Activity 1: Getting Acquainted with Chatbots* to learn what chatbots and how they work. Use the chatbots documented in **Activity 1: Exploring Chatbots**.
   a. Record your results of your explorations.
   b. Can you identify keywords to which your chatbot responds?

c. What are several keywords and responses they might cause?

2 Open the *AP Computer Science A Magpie Chatbot Lab Student Guide* and complete *Activity 2: Introduction to the Magpie Class.* As you complete the exercises, the guide will pose questions. Record your answers below.

a. Copy or extract the project files in *1.1.6MagpieActivity2_StarterCode_BlueJ* to your `BlueJProjects` folder. Open the project in BlueJ and compile.

b. Answer questions from the activity here. How does it respond to:

- My mother and I talked last night.
- I said no!
- The weather is nice.
- Do you know my brother?

c. Complete the **Exploration**.

d. Complete the **Exercises**, altering your code as instructed. Fill in the keywords and responses below.

Keyword	Response

e. What happens when more than one keyword appears in another word? Consider the string “My mother has a daughter but no cat periods”. Explain how to prioritize responses in the reply method.

f. What happens when a keyword is included in another word? Consider the string “I know all the state capitals” and “I like vegetables smothered in cheese”. Explain the problem with the responses to these statements.

3. Continuing in the *Magpie Lab Student Guide*, complete *Activity 3: Better Keyword Detection*.

a. Copy or extract the project files in *1.1.6MagpieActivity3_StarterCode_BlueJ* to your *`BlueJProjects`* folder. Open the project in BlueJ and compile.

b. Work through the section titled **Exploration: Using the API**.

c. Work through the section titled **Exploration: Understand the new method** and run it using the two statements in the previous step 3f.

d. As instructed in the guide, read through the `findKeyword` method and trace the following method call: `findKeyword("She's my sister", "sister", 0);`

Iteration	psn	before	after

e. Trace: `findKeyword("Brother Tom is helpful", "brother", 0);`

Iteration	psn	before	after

f. Trace: `findKeyword("I can't catch wild cats.", "cat", 0);`

Iteration	psn	before	after

g. Trace: `findKeyword("I know nothing about snow plows.", "no", 0);`

Iteration	psn	before	after

h. Complete **Exercise: Use the new method** to implement the more "intelligent" algorithm when responding to questions about pets and your teacher's name.

i. **Prepare for the next activity** according to the Student Guide.

4. Complete *Magpie Student Guide Activity 4: Responses that Transform Statements*.

   a. Copy or extract the project files in *1.1.6MagpieActivity4_StarterCode_BlueJ* to your `BlueJProjects` folder. Open the project in BlueJ and compile.

   b. Following the guide, use a statement like, "I want to have pizza for lunch", substituting your favorite food for "pizza".

   c. Complete the activity according to the Student Guide.

   d. Explain your improvement to the algorithm.

Part II: Ok Magpie ...

The quality of speech-to-text converters has improved significantly in the last few decades. With that improvement in quality, both speech-to-text and text-to-speech have become much more common. In this part of the activity, you will insert your Magpie code into an Android app, test the results, and if you have time, continue modifying your chatbot.

5. If you have not opened Android Studio before, refer to Part III in *Activity 1.1.1 Introduction to Android Development* to launch Android Studio for the first time.

6. Create an `OkMagpie` folder in your `AndroidProjects` folder.

7. Get a copy of the *1.1.6OkMagpieApp* from your teacher. Copy or extract the files to the `OkMagpie` folder in your `AndroidProjects` folder.

8. In Android Studio, import the OkMagpie project: Select **File > New > Import Project...**

9. A dialog appears showing your file structure. Navigate to your `AndroidProjects` folder and then navigate to the location where you copied or extracted the OkMagpie files. In the `OkMagpie` file structure, select a file named `build.gradle`. It will be in the `OkMagpie` folder, not in a subfolder. Click **OK**.

10. There is a missing file in this project. Create a copy of ***Magpie4.java*** from your BlueJ project in your OkMagpie Android app:
    a. In your ***BlueJProjects*** folder, copy ***Magpie4.java***. The easiest way to do this is to right-click on the file and select **Copy**.
    b. In the Project panel of Android Studio, right-click on ***org.pltw.examples.okmagpie*** and select **Paste**.
    c. A dialog appears to confirm the copy. Click **OK**.
11. Open your favorite emulator—or connect a device that is running on API 22 or later—and run the OkMagpie project.
12. Test the functionality of the buttons and text field in the app. Describe their functions.

Part III: How Intelligent Can It Get?

In March 2016, Microsoft released an artificial intelligence (AI) chatbot called "Tay" on Twitter. But very quickly, it had to withdraw Tay due to racist and misogynist remarks it had started to tweet. Microsoft did not program the offensive language. Rather, Tay *learned* it from other Twitter posts. Tay had been programmed to incorporate new responses into its algorithm over time as it interacted with people on Twitter. In a sense, Tay learned new behavior from people's inappropriate and offensive remarks.

This part of the project encourages you to consider artificially intelligent machines, and to form your own opinions about them.

13. Discuss with your partner:
    a. Can a chatbot be a cyberbully or harass someone?
    b. If an AI entity *does* offend someone, who is responsible?
    c. If you knew you were being bullied by an AI entity, would it make it more or less painful than if you were bullied by an actual person?
    d. Do you think it is possible to create an intelligent chatbot that can learn not to be offensive?

14. Present your opinion to another team, and summarize your discussions below. Be sure to include both how you agree and disagree on the issues.

Part IV: Make Magpie Smarter

Now that you have a working chatbot app, you will switch to Android Studio where you will modify the artificial intelligence of Magpie.

15. If you need to, you can always copy your code back into BlueJ and debug it there. When you want to debug your app in Android Studio, you may want to create output messages to troubleshoot your logic. To do so, you will insert statements like the following:

```
Log.d(TAG, "Your debugging message here. Some_variable=" +
someVariable);
```

or

```
System.out.println("Your debugging message here. Some_
variable=" + someVariable);
```

When you run an app in the emulator or on a physical device connected to your computer, the Android Dalvik Debug Monitor Server (DDMS) panel appears in Android Studio at the bottom of the window. The logcat tab of that panel is where the messages appear whenever your code writes output using the `Log` methods (such as `Log.d` or `System.out.println`.)

16. Use **the official Android developer reference** to find what the parameters `tag` and `msg` of `Log.d` represent and record the information.
17. Form pairs as directed by your instructor.
18. Discuss with your partner what functionality you'd like to add to Magpie's AI.
19. Have your instructor approve your plans for augmenting Magpie.
20. Implement your improvements to Magpie.

CONCLUSION

1. In general terms, explain how the algorithm for a chatbot might (or might not) differentiate between the phrases "please post this", "fencepost", "I like your post", and "post-apocalyptic".