victoryforphil / compsci-2018-lessons

Branch: master ▾ compsci-2018-lessons / src / projects / magpie / act3 / Magpie3.java Find file Copy path

victoryforphil Mag Pie Done 0313d3a 2 minutes ago

1 contributor

187 lines (173 sloc) 4.23 KB

```
/**
 * A program to carry on conversations with a human user.
 * This version:
 * <ul><li>
 *    Uses advanced search for keywords
 * </li></ul>
 *
 * @author Laurie White
 * @version April 2012
 */
public class Magpie3
{
	/**
	 * Get a default greeting
	 *
	 * @return a greeting
	 */
	public String getGreeting()
	{
		return "Hello, let's talk.";
	}

	/**
	 * Gives a response to a user statement
	 *
	 * @param statement
	 *            the user statement
	 * @return a response based on the rules given
	 */
	public String getResponse(String statement)
	{
		String response = "";
		if (statement.length() == 0)
		{
			response = "Say something, please.";
		}
		else if (findKeyword(statement, "no") >= 0)
		{
			response = "Why so negative?";
		}
		else if (findKeyword(statement, "mother") >= 0
				|| findKeyword(statement, "father") >= 0
				|| findKeyword(statement, "sister") >= 0
				|| findKeyword(statement, "brother") >= 0)
		{
			response = "Tell me more about your family.";
		}
		else if (findKeyword(statement, "swing") >= 0)
		{
			response = "She sounds greeeaaaatttttt!";
		}
		else if (findKeyword(statement, "robotics") >= 0)
		{
			response = "WHO ARE WE? 4RP!!!";
		}
```

```
56              else if (findKeyword(statement, "gths") >= 0)
57              {
58                      response = "why is asb so mean :(";
59              }
60              else if (findKeyword(statement, "java") >= 0)
61              {
62                      response = "Why u take my ram!";
63              }
64              else
65              {
66                      response = getRandomResponse();
67              }
68              return response;
69      }
70
71      /**
72       * Search for one word in phrase. The search is not case
73       * sensitive. This method will check that the given goal
74       * is not a substring of a longer string (so, for
75       * example, "I know" does not contain "no").
76       *
77       * @param statement
78       *            the string to search
79       * @param goal
80       *            the string to search for
81       * @param startPos
82       *            the character of the string to begin the
83       *            search at
84       * @return the index of the first occurrence of goal in
85       *         statement or -1 if it's not found
86       */
87      private int findKeyword(String statement, String goal,
88                      int startPos)
89      {
90              String phrase = statement.trim().toLowerCase();
91              goal = goal.toLowerCase();
92
93              // The only change to incorporate the startPos is in
94              // the line below
95              int psn = phrase.indexOf(goal, startPos);
96
97              // Refinement--make sure the goal isn't part of a
98              // word
99              while (psn >= 0)
100             {
101                     // Find the string of length 1 before and after
102                     // the word
103                     String before = " ", after = " ";
104                     if (psn > 0)
105                     {
106                             before = phrase.substring(psn - 1, psn);
107                     }
108                     if (psn + goal.length() < phrase.length())
109                     {
110                             after = phrase.substring(
111                                             psn + goal.length(),
112                                             psn + goal.length() + 1);
113                     }
114
115                     // If before and after aren't letters, we've
116                     // found the word
117                     if (((before.compareTo("a") < 0) || (before
118                                     .compareTo("z") > 0)) // before is not a
119                                                                             // letter
120                                     && ((after.compareTo("a") < 0) || (after
121                                                     .compareTo("z") > 0)))
122                     {
```

```
123                             return psn;
124                     }
125
126                     // The last position didn't work, so let's find
127                     // the next, if there is one.
128                     psn = phrase.indexOf(goal, psn + 1);
129
130             }
131
132             return -1;
133     }
134
135     /**
136      * Search for one word in phrase. The search is not case
137      * sensitive. This method will check that the given goal
138      * is not a substring of a longer string (so, for
139      * example, "I know" does not contain "no"). The search
140      * begins at the beginning of the string.
141      *
142      * @param statement
143      *            the string to search
144      * @param goal
145      *            the string to search for
146      * @return the index of the first occurrence of goal in
147      *         statement or -1 if it's not found
148      */
149     private int findKeyword(String statement, String goal)
150     {
151             return findKeyword(statement, goal, 0);
152     }
153
154     /**
155      * Pick a default response to use if nothing else fits.
156      *
157      * @return a non-committal string
158      */
159     private String getRandomResponse()
160     {
161             final int NUMBER_OF_RESPONSES = 4;
162             double r = Math.random();
163             int whichResponse = (int) (r * NUMBER_OF_RESPONSES);
164             String response = "";
165
166             if (whichResponse == 0)
167             {
168                     response = "Interesting, tell me more.";
169             }
170             else if (whichResponse == 1)
171             {
172                     response = "Hmmm.";
173             }
174             else if (whichResponse == 2)
175             {
176                     response = "Do you really think so?";
177             }
178             else if (whichResponse == 3)
179             {
180                     response = "You don't say.";
181             }
182
183             return response;
184     }
185
186 }
```