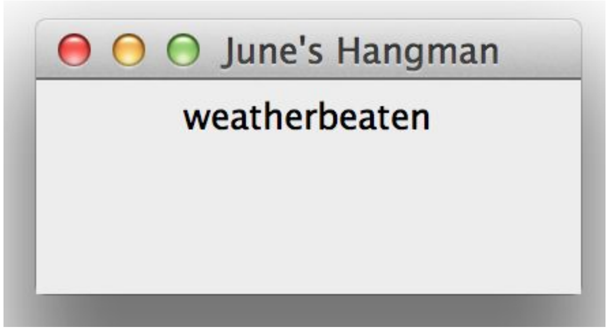
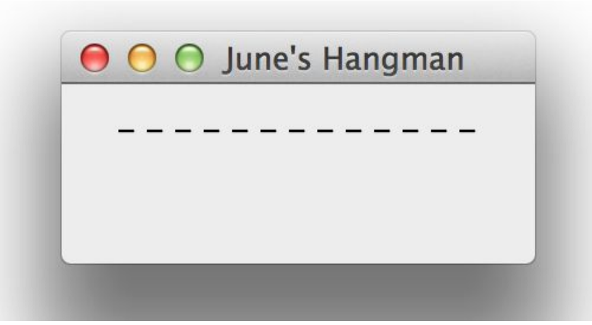
Create an interface that shows a new word (“puzzle”) when a key is pressed. Use a Stack to store the list of words.



Separate your main code into 2 methods… one that adds the puzzles to the stack (e.g. addPuzzles()) and one that creates the UI.

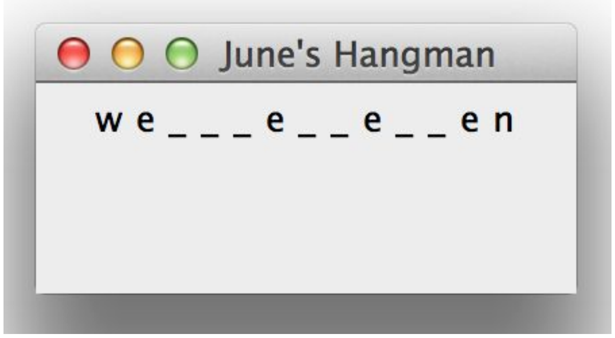
Stop showing the puzzle and replace it with a row of JLabels with text set to “_”. There should one JLabel for each letter in the puzzle.



If one of the keys pressed was contained in the puzzle, print “yay” to the console.

Update the JLabels so that they show the correct answers. To do this, create an ArrayList that holds all the JLabels, then call this method whenever a key is pressed. My ArrayList is called “boxes”.

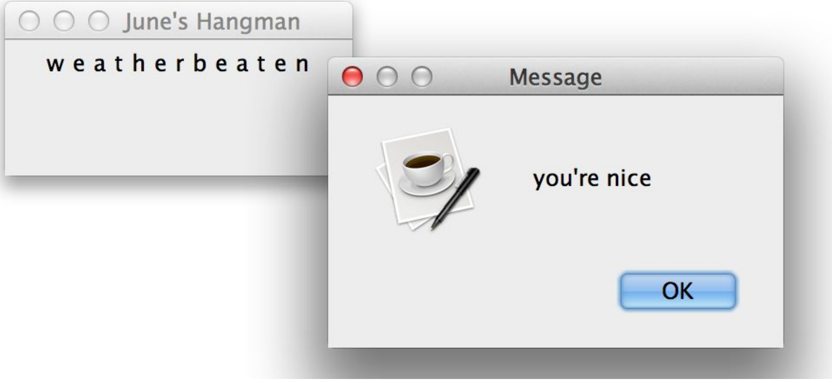
```
	private void updateSpacesWithUserInput(char keyChar) {
		for (int i = 0; i < puzzle.length(); i++) {
			if (puzzle.charAt(i) == keyChar) {
				boxes.get(i).setText("" + keyChar);
			}
		}
	}
```



```
w e _ _ _ e _ _ e _ _ e n
```

If they complete the puzzle, give them some love.

```
`     /* This method will tell you what is currently on the screen
      *  in the form of a String. */
      private String getCurrentAnswer() {
            StringBuffer answer = new StringBuffer();
            for (JLabel textBox : boxes) {
                  answer.append(textBox.getText());
            }
            return answer.toString();
      }
```

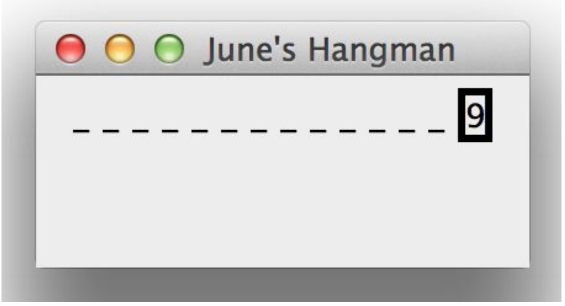


```
w e a t h e r b e a t e n
```

you're nice

If you’ve gotten this far, you are a rock star. Tidy up your code before we add hangings. e.g. You might want to extract the method that creates the JLabels (“boxes”);

Next we add punishment for incorrect guesses. Add a JLabel for number of lives and start it at 9.



I added a border to mine with:

```
  livesLabel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 3));
```

When they guess an incorrect letter, decrement the number of lives.

After they use up all their lives, and show the next puzzle. Extract this code into a method. For reference, mine looks like this:

```
      private void loadNextPuzzle() {
            removeBoxes();
            puzzle = puzzles.pop();
            System.out.println("puzzle is now " + puzzle);
            createBoxes();
      }
```

When they have no lives left, play the death knell.

```
      public void playDeathKnell() {
                  try {
                        AudioInputStream audioInputStream =
      AudioSystem.getAudioInputStream(new File("/Users/League/Google
      Drive/league-sounds/funeral-march.wav"));
                        Clip clip = AudioSystem.getClip();
                        clip.open(audioInputStream);
                        clip.start();
                        Thread.sleep(8400);
                  } catch (Exception ex) {
                        ex.printStackTrace();
                  }
            }
```

When you show the next puzzle, reset lives to 9.

Well now, aren't you clever?!