Department of Computer Science
The City College of CUNY

CSc 22100 [P 34271]: Software Design Laboratory [Fall 2021]

Assignment 3

A report uploaded on the Blackboard's course page for the section showing [1] *the problem,* [2] *solution methods,* [3] *codes developed, and* [4] *outputs produced for the assignment indicated is due by* 11:00 *pm on Tuesday,* 9 *November* 2021. ***The deadline is strictly observed.***

In-person demonstration of your application is required. The day and time of the demonstration will be announced on the Blackboard.

1- Amend the **MyShape** class hierarchy in Assignment 2 as follows:

MyArc *extends* **MyShape**;

Class **MyArc** inherits class MyShape. The **MyArc** object is a segment of the boundary of a **MyOval** object, defined by the endpoints $\mathbf{p}_1(x_1, y_1)$ and $\mathbf{p}_2(x_2, y_2)$, or their corresponding angles, on the **MyOval** boundary. The **MyArc** object may be filled with any color of **MyColor** *enum* reference type. The class includes appropriate class constructors and methods, including methods that perform the following operations:

a. *toString*— returns a string representation of a **MyArc** object;
b. *draw*— draws a **MyArc** object.

2- Implement a Java class **MyPieChart** that displays a *circular* pie chart of the probabilities of the *n* most frequent occurrences of an event to be specified in part 5 of the Assignment. The probability of event is given by:

$$Probability\ of\ event = \frac{Frequency\ of\ event}{\sum\ Frequencies\ of\ all\ events}$$

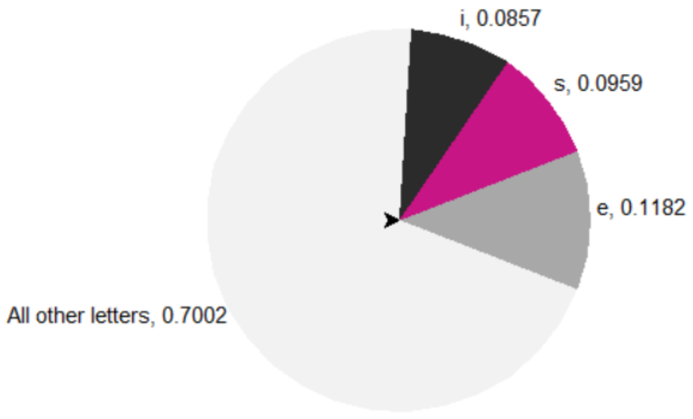
In the pie chart:

i. Each event is represented by a slice of the pie chart. The area of the slice is proportional to the probability of the corresponding event:

$$Probability\ of\ event = \frac{Central\ angle\ of\ segment}{2\pi}$$

ii. Each slice has a different color of your choice of type *enum* **MyColor**;
iii. Each slice has a legend showing the corresponding event and its probability;
iv. The slices are displayed in order of decreasing probability;

v. The last slice represents “All Other Events” and their cumulative probability. As an example, in the graph below where the event is the occurrence of a letter in a text: $n = 3$, and the probability of All Other Events is *one* minus the sum of the probabilities of events *e*, *s*, and *i*;



3- Class **MyPieChart** utilizes class **Slice**, and includes appropriate constructors and a method *draw* that draws the pie chart. The **MyPieChart** class utilizes a **Map** collection. The drawing canvas may include appropriate GUI components to input the number of events, *n* (variable), and display the pie chart together with the events and their corresponding probabilities.

4- Class **Slice** includes appropriate constructors and methods, including methods that perform the following operations:

a. *toString*— returns a string representation of a **Slice** object;
b. *draw*— draws a **Slice** object.

5- Implement a Java class **HistogramAlphaBet** that calculates the *frequencies* of the alphabet characters in “Alice in Wonderland” (file *Alice in Wonderland.txt*) and their *probabilities*. The **HistogramAlphaBet** class utilizes a **Map** collection for statistical calculations and the drawing canvas above to draw a pie chart of the probabilities. It also includes the **MyPieChart** class as an inner class.

6- You may only use JavaFX graphics and your own classes and methods for the operations included. Further,

a. The code is applicable to canvases of variable height and width;
b. The size of the pie chart is proportional to the smallest dimension of the canvas;

Hesham A Auda
25 October 2021