PROJECT 5 MAIN SPEC - Music Preference Visualization

Project 5 Music Preference Visualization (Team Project)

Team Project Logistics

Throughout the remainder of the semester, you will work on a group project in which you will systematically develop a medium-scale program that incorporates various aspects of the topics we are covering in this course. This project is in addition to and intertwined with, your individual programming projects, homework, and lab work. There will be intermediate deliverables due at specific points between now and the deadline of the project as outlined below. The GTAs will be reading through and executing your final submission.

Take ownership of your group and its responsibilities. Make key design decisions as a group, and assign well-defined individual tasks to each team member. Set up specific class interfaces so that each part can rely on the functionality of the others. Meet every few days or at least weekly to integrate member's code together testing it and to plan future project work. Each member is responsible for keeping copies of ALL project work, not just their own work. Do not let the external deadlines of the course be the only motivating force. Do not be the person who lets your group down – this will come to light through the individual contribution reports if not otherwise and will negatively affect your grade. Be sure to start the project early, if you have concerns about your groups functioning during the design phase let us know. After the design phase check in with us if your group is dysfunctional, **April 9 is the deadline for group adjustments**. After the project you will be assessing each other, documenting your timelines and contributions is wise.

You will develop JUnit tests for every non-GUI class except the Input class or ENUMs. Remember to test as you develop!

Project Specifications

The project will be a visualization of the music survey data we collected for two sections of CS 2114. Information Visualization allows users to view multidimensional data from different perspectives and can give insight into data patterns and relationships. You can read an overview at **Wikipedia's data visualization page. (https://en.wikipedia.org/wiki/Data_visualization)**

Here are some generic examples of information visualization:

- **Google Public Data Explorer (http://www.google.com/publicdata/directory)**
- **Dear DATA (http://www.dear-data.com/theproject)**

In this project there are specific expectations for how to display the provided data. There are some implementation guidelines, but the backend design is largely up to your team. You will need to determine how to design your classes and distribute the coding responsibilities. We encourage you to use tools such as GitHub (for version control) and Gliffy (for UML diagrams), however it is imperative that your work is kept private, either with a free student account or paid private account. It is an Honor Code Violation to have any of your solution publicly available online.

Output

"In the context of data visualization, a glyph is the visual representation of a piece of data where the attributes of a graphical entity are dictated by one or more attributes of a data record." — Matthew O. Ward , Multivariate data glyphs: Principles and practice, In: Handbook of data visualization (2008), p.180.

The above visualization is based of the data from the Spring 2016 surveys, your display should represent the data from the following surveys (**SongList.csv (https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** and **MusicSurveyData20182018.csv (https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)**). Each mini graph is called a glyph. Each glyph represents a song and its vertical axis represents a student attribute(hobby, major or region). The glyphs are color coded, (use only these colors), which is represented in the legend. The left side of the glyph is the percentage of those students who have *heard* the song and the right side is the percentage of those students who *like* the song. So, if there are 100 students whose hobby is reading and 90 of those students have heard the song, that's 90%. If there are only 30 students whose hobby is art and 27 of them have heard the song, that's also 90%.

The user can navigate through the data with `prev` and `next` buttons with 9 glyphs showing at a time.

The user can select whether to represent the student survey responses by:

- hobby (read, art, sports, music)

- major (Computer Science, Other Engineering, Math or CMDA, Other)
- state (Northeast US, Southeast US, the rest of US, outside the US)

The user can select whether to sort the song glyphs on the screen by:

- artist name (strings are sorted alphabetically, and are **not** case-sensitive i.e. a == A and A < b)
- song title
- genre
- date (ascending)

We do not require you to sub-sort at all. We will not grade for this. For example, if you sort by genre then the songs that have the same genre are not required to be in a specific order.

Sorting of strings should consider NOT case. So "ABBA" < "a-ha" < "Tom Petty" for artist. But for Genre it does : "R&B" < alternative

Input

There will be two input files. The music input file, **SongList2018.csv (https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** , is a list of the songs in the survey. For each song there is a column for Song Title, Artist, Year and Genre. The survey input file, **MusicSurveyData2018.csv (https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)** , contains the survey data for all the participating students. Each line contains a data record from one student survey. There is a column for each of the various student attributes: Person ID, Date of Survey, Major, Region and Hobby, then columns for responses to the song questions. For each song there are two columns with a possible "Yes", "No" or blank response. The first column is the response for whether the student has heard the song, and the second is for whether the student likes the song. The song question columns in the survey input file are in the same order as the songs are listed in the rows of the music input file.

3 example music input file lines:

```
Uma Thurman,Fall Out Boy,2015,rock
Upside Down,Diana Ross,1980,disco
Watching the Detectives,Elvis Costello,1977,punk
```

3 example survey data input file lines

There is a single newline character at the end of each record. Here the records are displayed with wrapped lines and an extra line between each record to make them easier to read.

```
190,10/19/15 14:45,Math or CMDA,Southeast,music,No,No,Yes,Yes,Yes,Yes,No,No,Yes,Yes,Yes,No,Yes,
No,No,No,No,,,Yes,Yes,No,No,No,No,Yes,Yes,Yes,No,No,No,Yes,Yes,Yes,Yes,No,No,Yes,Yes,Yes,Yes,
Yes,Yes,No,No,Yes,No,Yes,Yes,Yes,Yes,No,No,No,No,Yes,No,Yes,Yes,No,No,No,No,Yes,No,Yes,Yes,No,No,
Yes,Yes,No,No,Yes,Yes,No,No,No,No,Yes,No,Yes,No,No,No,Yes,No,Yes,No,Yes,Yes,Yes,Yes,Yes,Yes,
```

```
No,No,Yes,Yes,Yes,Yes,Yes,No,Yes,No,Yes,Yes,No,No,No,No,No,No,No,Yes,Yes

191,10/19/15 14:59,Computer Science,United States (other than Southeast or Northwest),,No,,No,,Yes,
Yes,No,Yes,Yes,Yes,Yes,Yes,No,,No,,No,,Yes,No,No,,Yes,Yes,,,Yes,Yes,Yes,Yes,
No,,Yes,Yes,Yes,Yes,No,No,Yes,Yes,,,No,,No,,,,Yes,No,,,,,,,Yes,Yes,No,,No,,No,,No,,
No,No,No,,Yes,Yes,No,,Yes,Yes,No,,No,,No,,Yes,Yes,No,,No,,Yes,Yes,No,,Yes,Yes,No,,
No,,No,,Yes,Yes,Yes,Yes,Yes,No,No,,No,,No,,No,No,Yes,Yes,No,

192,10/19/15 15:01,Math or CMDA,Southeast,sports,No,No,Yes,Yes,Yes,Yes,No,No,Yes,Yes,
Yes,Yes,Yes,Yes,No,No,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,No,No,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,No,No,Yes,Yes,Yes,,Yes,Yes,Yes,No,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,No,No,Yes,Yes,
Yes,Yes,No,No,No,No,Yes,Yes,Yes,No,No,No,Yes,Yes,Yes,No,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,No,Yes,No,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,No,No,No,No,Yes,No,Yes,Yes,Yes,Yes,
Yes,Yes,No,No,No,No,No,Yes,Yes,Yes,Yes
```

There will be some blank data fields in the input. If a student attribute is blank (hobby, major, region), then exclude that student from your data. If a song response is blank, then do not count it. So, you can tally the "Yes" responses and "No" responses and the percentage of those that are "Yes" responses is what will determine the length of the bar.

It is a good idea to test in small pieces. Download the the two survey files, **SongList.csv (https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981305/download?wrap=1)** and **MusicSurveyData20182018.csv (https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)** **(https://canvas.vt.edu/courses/81975/files/8981302/download?wrap=1)** . You should also create some smaller test files for testing.

Implementation Guidelines

- Create a GUI that has all the functionality listed in the **Output** section and demonstrated in the video.
- Include the PIDs of your group members (separated by spaces) in the title of your GUI window.
- Implement a Linked List and an Iterator.
- Your Linked List must provide sort functionality. You may implement an insertion sort similar to the one provided in your textbook.
- Use Linked List(s) to store and sort the data.
- Have separate classes for front-end and back-end responsibilities. Name front-end class with the prefix GUI for our grading purposes.
- You may only use class and Java standard APIs (e.g. Scanner, ArrayList).
- Name ENUMS with the suffix Enum(such as HobbyEnum) for our grading purposes.

Schedule

Reference **Submissions** sections for details on each Milestone.

Milestone	Due
Project Group of 4 (or we will group you)	Friday, March 22
Project design with UML	Sunday, April 7

Milestone	Due
Intermediate submission	Tuesday, April 16
Final Submission DUE	Tuesday, April 30
Individual Contribution Response	Sunday, May 5

Submissions and Grading

Submissions will be set up on both canvas and WebCat. *Late submissions will only be accepted within 24 hours of the deadline and for a 10 point deduction, this is different than for individual projects!*

Project Description and UML (15%) – Submit to Canvas

- UML Diagram and details in accompanying prose
- Front end(GUI) and Back end(Model) distinguishable
- Linked List Implementation (what's your approach, what's needed?)
- Linked List use
- Adequate data fields for all data from input files
- Adequate data fields for display
- Behavior to handle various display changes based on buttons pushed
- Refer to Implementation Guidelines above

Intermediate Solution (20%) – Upload to both WebCAT and Canvas

- Updated UML Diagram
- Screen shot that shows widgets of the display. This display does not have to be connected to the backend yet
  - all the buttons
  - at least one default glyph, you can still be using constant values for the glyph bar sizes.
  - a legend
  - the order of bars in the glyph(s) and legend should correspond to video
- Generate text output of the data for 2 views
  - hobby, sorted by song title
  - hobby, sorted by genre
- The code uploaded to WebCAT is expected to have the above functionality and testing with full code coverage except for the front-end. Name front-end class with the prefix GUI for our grading purposes.
- You should list all the members of your group as partners, so various people can submit and see the feedback and receive the grade.

Final Submission (20%) - Upload to WebCAT and Canvas

- Submit your final code to WebCAT.
- You should list all the members of your group as partners, so various people can submit and see the feedback and receive the grade.
- Upload your final UML to Canvas.

We will be following regular project code grading standards, this is the first time you will be scored on the class design portion: **Project Grading Rubric. (https://canvas.vt.edu/courses/81975/pages/project-grading-rubric-fall-2018)**

Program Execution (40%)

The GTA will pull your submission down from Web CAT and run it. To avoid a deduction, have the course input files in your project and the file names hard coded for your final submission. The GTAs should just be able to run your code as an Application without entering or looking for files. **Your code should accept command line arguments as was needed for the intermediate submission. If there are no arguments your code should input SongList2018.csv and MusicSurveyData2018.csv which should be saved in your project. This is similar to how "input.txt" was specified in the space colonies project.**

- 20% Properly display class data set for the following views:
  - hobby, sorted by song title
  - major, sorted by artist
  - region, sorted by year

- 20% All buttons, any properly formatted input file and code should be able to run with various input files.
  - We may additionally run it with our test files
  - Every button should work
  - The code uploaded to WebCAT is expected to have the above functionality and testing with full code coverage except for the front-end. Name front-end class with the prefix GUI for our grading purposes.

Self and Team Evaluation (5%) - Complete form on Canvas

(https://canvas.vt.edu/courses/81975/files/8981318/download?wrap=1)