Introduction

Rainbow Boys Progress Tracker is a software used to track progress for gymnasts at Gymnastics World, a gymnastics training center in Brecksville, Ohio. The way the Rainbow Program works at Gymnastics World is that gymnasts learn skills in order from a list, and every three skills they learn, they earn a ribbon. These ribbons are colored in the order of the rainbow, then after the rainbow is finished, they earn a bronze, silver, and gold medal, hence the Rainbow Program. For each gymnast there are six events where ribbons can be earned, which means there are 9 different ribbons, so 27 skills per event. That is 162 skills per gymnast that need to be tracked. Combine this with hundreds of gymnasts each year and the need for tracking software becomes clear. That is the purpose of Rainbow Boys Progress Tracker.

The primary features of this software are to sort gymnasts into classes, be able to effectively search for a gymnast, and track their progress on each event. Also, new gymnasts and classes can be easily added to the system. The software also supports adding coaches to the system and associating them with any number of classes. This manual should help coaches with admin duties use this software to best utilize its capabilities and effectively track gymnast progress.

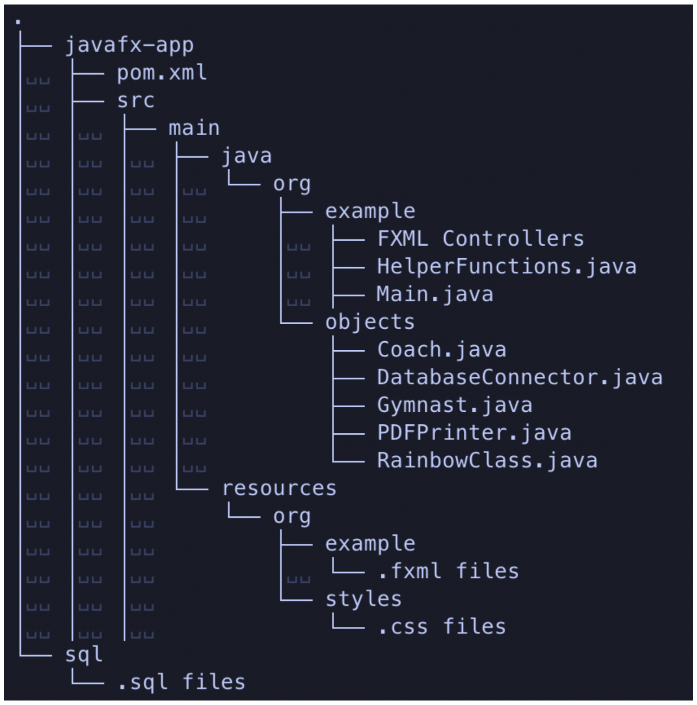
Getting Started

To create a development environment for this software, you will need a couple of other pieces of software to begin. This is a Java application, so a JDK is required for development. Along with that, Maven is used as a build tool to put the application together, grab dependencies, and execute the software.

After you have Maven, a JDK, and the files for this project on GitHub, you begin by changing to the directory inside the project directory called javafx-app that has the pom.xml file. After this, just execute the command `mvn clean javafx:run` and the software should start running.

Project Structure

Below is an image of the relevant file structure of the project. There are other directories created by maven, but these should be of little relevance to development at its current state.



```
.
├── javafx-app
│   ├── pom.xml
│   ├── src
│   │   ├── main
│   │   │   ├── java
│   │   │   │   └── org
│   │   │   │       ├── example
│   │   │   │       │   ├── FXML Controllers
│   │   │   │       │   ├── HelperFunctions.java
│   │   │   │       │   ├── Main.java
│   │   │   │       └── objects
│   │   │   │           ├── Coach.java
│   │   │   │           ├── DatabaseConnector.java
│   │   │   │           ├── Gymnast.java
│   │   │   │           ├── PDFPrinter.java
│   │   │   │           └── RainbowClass.java
│   │   │   └── resources
│   │   │       └── org
│   │   │           ├── example
│   │   │           │   └── .fxml files
│   │   │           └── styles
│   │   │               └── .css files
└── sql
    └── .sql files
```

The most important directories are java/org/example and resources/org/example. The controllers for each fxml file are in java/org/example, which provide make the windows behave how they are supposed to. Any tine a new window is created, a controller file should be attached to the window to give it functionality. As such, there should be as many fxml files as there are controllers. In resources/org/example the fxml files reside. It is these that display every window (or the tabs in the main window). The controllers for a given fxml file should have a similar name, like GymnastScreenController.java for gymnast_screen.fxml.

Aside from these, there are objects in java/org/objects. Here the objects for gymnasts, coaches and classes lie. To give the appearance to the windows, there is resources/org/styles for css files. Aside from that there are a couple classes with static methods that prove useful across all controllers including HelperFunctions.java and DatabaseConnector.java.

Contributing to the Project

Some decisions have been made for how the data should come from the database to the java code. When you use JDBC to execute a query, a ResultSet is returned that has column names associated with values. To make things standardized, this ResultSet is passed into the constructor of a class to give the object values directly from the ResultSet. The way that all queries and updates are done to the database are through stored procedures. This is to ensure minimal SQL is written in the DatabaseController.java file. So, to create a new query, a new stored procedure or function must be added to the database, then a static method in DatabaseConnector.java must call that stored procedure or function. To follow suit with the constructors for the project, entire rows of the database from the gymnasts or classes tables are returned from every function.

Apart from this, there aren't many significant coding conventions used for this project. More should be enforced in the future as better project organization rolls out.

References

Here is a list of some useful documentation that covers some of the dependencies used in this project.

JavaFX CSS Style Guide

JDBC Docs

JavaFX Docs

Useful JavaFX Video Series

Introductory Maven Information