Noah Golowich

Design:

All information stored by the app is stored in one of two SQLite databases, one of which holds players and the other of which holds reports on players. SQLite databases are well-suited for the categorized data that the app is intended to store (the only other option is to store data in files, which is much slower and more difficult to organize). Moreover, I decided to store all data on the device itself rather than in a central server in order to make queries to the database as as fast as possible.

The single table in the player database has an unique id for each player, which is used to associate each report in the report database with a single player. There are two tables in the report database. One table holds the “basic information” for each report, including a variety of ratings on each of a player's strokes, as well as a player id to associate the report with a unique player. The other table in the report database holds “advanced information” for each report, including a match score, comments, and a report id to associate this advanced information with a unique “basic information” report. I decided to separate these two tables in the report database since a user may not always want to enter information into or view information in the “advanced table”. Therefore, by keeping the “basic information” database smaller, SQL queries to that database should be faster.

Nearly every one of the project's activities either enters, updates, or reads information from one of these two databases. In particular, AddPlayer, EditPlayer, ViewPlayers, and PlayerDetails allow the user to add, edit, or view information in the players database. Rather than simply storing the name of each player, I decided to store information such as a player's hometown and year of birth to help to differentiate between people of the same name. All of this supplementary information on a player is displayed in a short phrase in PlayerDetails to make the user experience more pleasing.

The AddPlayerReport and ViewReport activities allow the user to enter information into and view information in the “basic information” table in the report database. All of this information is entered into RatingBars, which provide a convenient and simple method of comparing different aspects of a player's game. I decided to only allow 1-5 stars in the AddPlayerReport activity (and not 0 stars in any of the rating bars, as to indicate that a rating is not available) in order to force the user to completely analyze their opponent's game. By viewing either individual reports they have already entered or by choosing the “average reports” button, the user can access the data they have entered in the past on one of their opponent's strokes, so as to identify his/her weaknesses and strengths, which is the main goal of the app.

The user is not obliged to enter any information into the AdvancedReport activity; to reflect the fact that such information may not be available, this information is only viewable in the OverallViewAdvanced activity after selecting the appropriate list element in ViewReport.

The classes PlayerOpenHelper and PlayerReportOpenHelper extend SQLiteOpenHelper and specify how to create the databases when the app is first installed on an android device. They are used whenever a readable or writable database is needed.

The class Stroke is used to store basic information (e.g. power and consistency) on a player's stroke; it is written so that it can store overall information on a player (power, consistency, aggressiveness, and court speed) as a single stroke, rather than as a separate class. I made this decision in order to make StrokeAdapter, which extends ArrayAdapter<Stroke>, much simpler (having two separate classes would take up many more lines of code). StrokeAdapter is used to display an ArrayList<Stroke> as a list of RatingBars in a ListView.