U D A C I T Y

Score Keeper

REVIEW
CODE REVIEW 5
HISTORY

Meets Specifications

Congratulations on completing the project! 👏 💥 🎉

You did a great job to include all stats and all were working perfectly!
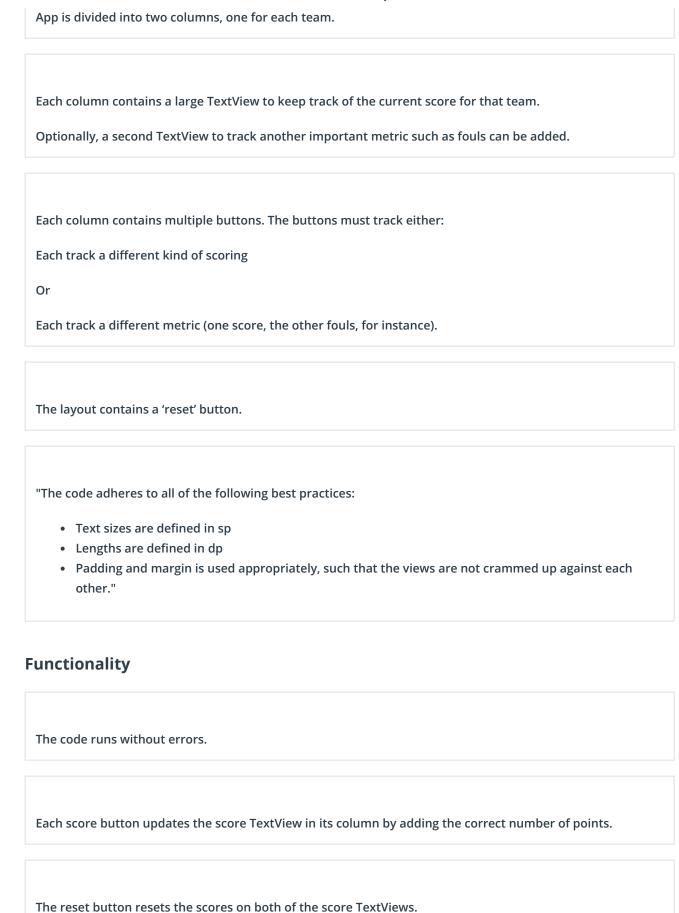
Best of luck for future! 👍

I LOOK FORWARD TO HEAR FROM YOU IN YOUR FEEDBACK.

Layout

> The chosen game has either multiple amounts of points that can be scored, as in american football, or multiple important metrics to track, such as fouls, outs, and innings in baseball.

App is divided into two columns, one for each team.

Each column contains a large TextView to keep track of the current score for that team.

Optionally, a second TextView to track another important metric such as fouls can be added.

Each column contains multiple buttons. The buttons must track either:

Each track a different kind of scoring

Or

Each track a different metric (one score, the other fouls, for instance).

The layout contains a 'reset' button.

"The code adheres to all of the following best practices:

- Text sizes are defined in sp
- Lengths are defined in dp
- Padding and margin is used appropriately, such that the views are not crammed up against each other."

Functionality

The code runs without errors.

Each score button updates the score TextView in its column by adding the correct number of points.

The reset button resets the scores on both of the score TextViews.

Code Readability

"Any classes are named after the object they represent.All variables are named by their intended contents.

All methods are named by their intended effect or in the style required by a callback interface."

"There are no unnecessary blank lines.One variable is declared per declaration line.

The code within a method is indented with respect to the method declaration line."

⬇ DOWNLOAD PROJECT

5   CODE REVIEW COMMENTS   ›

RETURN TO PATH

Rate this review