

Application name: BBALL_SCOREIT
Application target: Android 2.3
Team Name: White_WVU
Team Members: Steven White

I chose to build a CBB scoring application for the Android platform. I received notification of this competition from a professor and thought it sounded fun as I have always been a huge basketball fan and a small amount of previous Android experience.

I. INSTALLATION

The apk file BBall_ScoreIt1-0.apk can be installed to an emulator or mobile device running Android 2.3 or higher. The simplest way to install is by terminal via the command:

```
adb install <path-to-apk>/BBall_ScoreIt1-0.apk
```

II. UI DESCRIPTION

II-A. LOGIN SCREEN/GAME LISTING SCREEN

The application as it stands now initially launches a login screen, though my credentials have been hard-coded into the app for ease of testing. Once logged in, a list of games is displayed (the same range available through the espn_alps website, this can be easily changed). The user then selects a game to score and the scoring portion of the app begins.

II-B. GAME SCORING SCREEN

On the left side of the screen, five basketballs are displayed with an abbreviated last name and jersey number overlaid on top. These represent the on-court players for the away team, and the same can be found on the right side of the screen to represent the home team.

The top of the screen contains a centered piece of text to indicate the current part of game, such as pregame, start of 1st, end 1st, etc. To the left of this is the away team name, with an action button below, and a larger piece of text representing the current score of the away team. This setup is mirrored on the right portion of the screen for values representing the home team.

In the center of this screen is an image of a basketball court, with a basketball overlaid on top initially at center court. This basketball is used to indicate the position of events that require a location. The user can simple touch or drag on the court and the ball will follow.

II-C. STATS SCREEN

The stats screen can be accessed from the game scoring screen by pressing the menu button followed by selecting the "Show Stats" menu item. The current stats for all players from each team are then displayed in an easy to read format.

III. SCORING EVENTS

Once a game is selected to score, the application asks the user to select five starters for the away team, followed by the home team. These players are placed on their respective basketballs on the screen as described previously.

III-A. PLAYER EVENTS

All player events (rebound, made shot, missed shot, turnover, foul, substitution) can be accessed by the basketball for the specific player. This will bring up a dialog with a list of potential actions, and touch each action leads to different subsequent dialogs until enough information has been gathered to submit the request to the server. For example, if a user wished to send an offensive rebound request, they would touch the player who grabbed the rebounds basketball, touch rebound, and touch offensive. At this point the application will show a progress dialog indicating to the user that the request is being sent. Once a response has been received, this dialog will automatically be closed.

III-B. TEAM EVENTS

All team events (team rebound, team timeout, team technical foul) can be accessed by touching the ACTION button located directly below the corresponding team's name. The logic behind sending an event is similar to that described in PLAYER EVENTS. The same progress dialog will be shown while submitting a request until a response has been received.

III-C. OFFICIAL EVENTS

All official events (jump ball, period start, period end, official timeout, media timeout) can be accessed by touching the menu button followed by the "Official" menu item. These events also follow a similar logic to that described in PLAYER EVENTS and display the same progress dialog.

IV. PROGRESS TO DATE

The approach I took to develop this application was to first code up support for sending game events followed by attempting to make a decent UI. The major deficiency of the app in its current state is the complete lack of game logic. For example, there is currently no requirement that the game starts (sending period start document) before a player misses a shot, though I believe that this would not be a difficult thing to implement in a future version, I simply ran out of time at the April 9 deadline.

A separate area where the app is lacking is checking for previous events entered when receiving the document from the getgamedata method. On a similar note, the user is also unable to change previous events. This is lacking for reasons similar to those listed previously.

V. CONCLUSION

I had a really good time developing this app, and tested it out on the Miami OKC game this past Wednesday where I had no problems keeping up with the action, despite pretending LeBron was Mark Refstien. I think applications of this nature present a new way to watch games from home as well as in person.

I also would like to thank the ESPN alps team for conducting this competition and would be very interested in participating in future events. Thanks for your consideration.