ELECTRONIC SCOUTING SYSTEM

DONALD PINCKNEY

SEASON OF 2013-2014

Introduction

Brief History

Last year I published a white paper of team 1678's 2012-2013 season scouting system, available here:
http://www.chiefdelphi.com/forums/showthread.php?t=118980&highlight=electronic+scouting

In short, scouts used Android tablets to collect data during matches, and then upload the data later. We had an iPhone app that our drive team could use to view scouting data on robots to strategize for matches.

At the close of the 2013 season we identified some goals for 2014: Improve Internet connectivity on the tablets so that we can upload data reliably after every match, create visual graphs of robot performance to help communicate robot abilities, and integrate high quality photos of robots to help identify what sort of role each robot could play on an alliance.

For the 2014 season we reworked many parts of our system: we kept the fundamentals the same, but primarily improved the mechanism for uploading data, and the reliability of the processing of that data. Specifically, we have six scouts with Android tablets recording objective data (# of shots made, etc.), and two "superscouts" subjectively evaluating relative performance within and alliance and match (how fast each robot is, etc.) All of this data collected on the tablets was uploaded to a server, which performed various calculations and pushed the results to an app for iPhone and Android. The calculations implemented on the server include our pick-list calculations, and also a detailed breakdown of the stats of a robot, including data such as: auto shooting, mobility, goalie blocks, teleop shooting, catching, trussing, ground receiving, and human player receiving. We then display succinct summaries of the offensive and defensive strengths of robots for each match we play in, with easy hyperlinks to view the more detailed data on each team in a match.

Teams can be easily sorted by first and second pick rankings (calculated on the server) and are updated throughout the competition. We develop the calculations for the pick lists based off of quantitative analysis of previous competitions. Ultimately, we supplement the raw pick list calculations with some subjective observations as well.

Advantages of The System

The goal of Citrus Circuits's scouting system was to allow for seamless input of data by scouts in the stands into a central database. The immediate benefit is being able to much more effectively develop alliance pick lists. However, the system also lends itself as a critical component of developing match strategy. Clearly, Aerial Assist was a game in which strategic preparation was key, and we believe we benefited by creating detailed match strategy beforehand, all powered by the data available from scouting app. Concretely, using the system we were able to go undefeated during qualification matches on the Newton Division, even though we were predicted to loose 2 matches by OPR data. Following our success in qualifications, we were able to pick team 1640, an accomplished offensive robot with a great swerve drive, as our 3rd robot from 1st seed position: an achievement that highlights the competitive success of our system.

Logistical Overview

Materials:
ASUS Memo Pad HD 7-inch tablet: ($150.00) x 6 = **$900**
http://www.amazon.com/HD-7-Inch-Tablet-Green-ME173X-A1-GN/dp/B00E0FDYKY/ref=sr_1_4?s=electronics&ie=UTF8&qid=1399163828&sr=1-4&keywords=ASUS+MeMOPad+HD+7-Inch

Nexus 7 + T-Mobile SIM ($350) x 2 = **$700**
https://play.google.com/store/devices/details/Nexus_7_32GB_Black_Wi_Fi_Mobile_data_Unlocked_T_Mo?id=nexus_7_32gb_2013_lte_tmo

Virtual Private Server (~ $7.00/month) = **$100**/year
http://www.chicagovps.net/index.html

MacBook Pro ($1,200 new, but we bought much cheaper via Craigslist) = **$1200**
http://store.apple.com/us/buy-mac/macbook-pro

iOS Developer Program (necessary for programming iPhone apps) = **$100**/year
https://developer.apple.com/programs/ios/

Approximate total = **$3000**

These materials do not include the smartphone necessary for viewing the data, because smartphones are popular enough to probably be owned by someone on the drive team. Note that both the MacBook Pro and the iOS Developer Program are needed to program iPhone apps. Implementing the system with only Android would reduce the cost, but at least one dedicated programming laptop would still be necessary.

Experience necessary:
Android and iOS programming experience. If lacking experience, follow this excellent class by Stanford on programming iOS apps: https://itunes.apple.com/us/course/developing-ios-7-apps-for/id733644550
I don't know as much about resources to learn Android programming, but I am sure you can find some online.

The programming abilities demanded by iPhone and Android app programming should be sufficient for any other programming in the system.

Technical Overview

The scouting system consists of 5 main parts:

- An **Android input app** running on eight tablets, which scouts sitting in the stands input data into.
- A **server** programmed with Node.js, which receives data from the Android app, and using it calculates statistics on teams and pick lists.
- An **iPhone viewing app,** which lets team members view:
    - The qualification match schedule with predicted and final scores
    - 2nd and 3rd robot pick lists.
    - Detailed statistics on each team, including graphs of their performance over time.
    - High quality photos of robots.
- An **Android photo-taking app,** which a trained media student uses to take high quality photos of all robots at the competition. These photos are automatically uploaded into the scouting system and then displayed in the viewing app.
- **Printed sheets** for each robot: Right before we make pick lists, we print a massive PDF file which has a full 8.5'' x 11'' page for each robot, including a photo and many stats. The iPhone-viewing app is programmed to automatically generate this file. We implemented this idea after the Sacramento Regional, taking inspiration from team 971.

Implementation Details

Internet Problem – Bluetooth!

To provide real-time data to the iOS app requires the Android tablets in the stands to have a constant Internet connection. However, rule T4 in the FRC Game Manual states that "Teams may not set up their own 802.11a/b/g/n/ac (2.4GHz or 5GHz) wireless communication (e.g. access points or ad-hoc networks) in the venue."

This rule made it significantly difficult to provide a constant Internet connection to all eight Android tablets, as this specifically outlaws a simple Wifi hotspot. We were not able to finance eight 3G tablets and data plans, and we were committed to building a fully legal system, so we had to get creative.

We concluded the 2013 season with a system that used a messy network of USB cables to transmit data from tablets to a Raspberry Pi via USB, which would then upload to the server. However, this system left a lot to be desired, particularly because the Raspberry Pi only supports a maximum of six USB devices, not our required eight.

This year, we decided to take another attempt at the Internet issue using Bluetooth. We created our system so that two of our eight tablets had cellular data from T-Mobile, and the remaining six would transmit their scouting data over Bluetooth to one of those two tablets, which would then upload it to the server. In our system, the two tablets with cellular data are the "superscout" tablets, but could be and of the tablets.

Android Input App: Native App, Programmed in Java

Key Functions:

Automatic scheduling: The Android app receives the match schedule from the server, and using that automatically keeps track of the schedule. It tells the scout using the tablet which match it currently is, and what robot to watch. The schedule is transmitted as **JSON** data, and only needs to be transferred to the tablets once, often the night before qualification matches.

Simple objective data collection: We believe that having scouts collect simple objective data is the best way to obtain the most accurate data for pick lists. The Android input app is about as simple as possible: it contains a +/- stepper for each action a robot can perform (high shot make, high shot miss, truss, etc), or in the case of the "superscouts", a +/- stepper to assign relative rankings to robots.

Data collection format: We create 1 **JSON** file for each robot in a match, which contains all of that robot's stats. An example file looks like this:

```
{
  "teamNumber": 1678,
  "matchNumber": "Q5",
  "isRed": false,
  "scoutName": "ko",
  "auto": {
    "Make Low Cold": 0,
    "Miss Low": 0,
    "Make High Cold": 1,
    "Miss High": 1,
    "G. Receive": 1,
    "Mobility": 1,
    "Make High Hot": 1,
    "Make Low Hot": 0,
    "Eject": 0
  },
  "tele": {
    "Make High": 1,
    "Miss Low": 0,
    "Miss High": 0,
    "G. Receive": 6,
    "HP Receive": 0,
    "Catch": 0,
    "Truss": 4,
    "Make Low": 0,
    "HP Receive Fail": 0,
    "Eject": 1,
    "Goalie Block": 0
  }
}
```

Data uploading: As mentioned above in the Internet Problem section, the Android app transmits the above JSON file via Bluetooth to a tablet that has cellular data. Once the tablet with cellular data receives the JSON data, it uploads it to the server via Dropbox. We use the Dropbox developer API to have the tablet just save the JSON file into our team Dropbox account, in a specific folder structure. This way, we don't have to concern ourselves with making sure the data uploading code is reliable: we simply use Dropbox, which is essentially the most reliable system in existence for uploading arbitrary files. Also, using Dropbox detaches the server from being responsible for the uploading: if the server were to fail, then all of our files would still upload to Dropbox just fine, and no data would be lost.

Server: Programmed in Node.js

In general, the server in our system this year is very transparent: it is critical to the system, but you almost never interact with it except for programming.

Key Functions:

Receiving uploaded data: As mentioned above, all of the data is uploaded to Dropbox by the Android tablets. To receive and process this data, the server needs to be able to sync with the team's Dropbox account. For our server we use a Linux VPS, and luckily Dropbox supports Linux and includes a command-line interface. For the server to receive all the uploaded data, all we had to was install Dropbox on it, and let it sync!

Storing match schedule: The server is programmed to automatically download the match schedule from FIRST via a URL like this: http://www2.usfirst.org/2014comp/events/Newton/schedulequal.html and cache the downloaded match schedule in a JSON file on the server.

Perform detailed, flexible calculations for the iOS app: The calculations and statistics that appear in the iOS app are all calculated on the server.

iOS App: Native App, Programmed in Objective-C

Key Functions:

Minimizes the need for the FRC Spyder or FRC Tracker app in competition. The app consolidates scouting data, match schedule, and seeding all into one app. The app features both predicted match scores and official match scores.

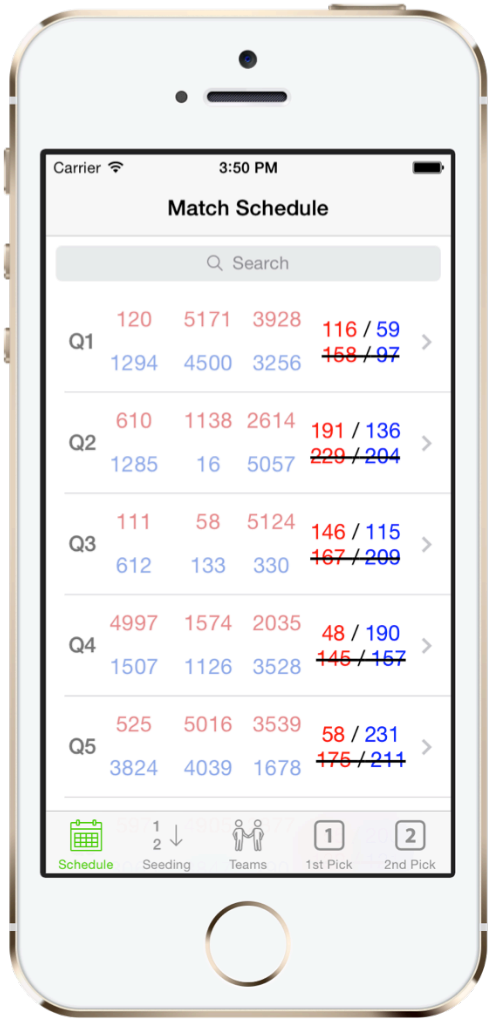
Alliance pick list output: The app displays our 1st and 2nd robot pick lists (calculated on the server) to our team leaders and mentors. The fast and automated calculation of these lists are invaluable to our ability to pick good alliance partners.

Provides a quick overview of the abilities of other robots in upcoming qualification matches. Strategists and our drive team can quickly see what the best offensive and defensive robots are in upcoming qualification matches, analyze what team is best at what role, and communicate with alliance partners far in advance.

Provides more detailed calculations for each team. This is all calculated on the server, and has dynamically generating UI based on what the server calculates. Example calculations include autonomous average and maximum scores, teleop accuracy, human player receiving data, etc.

Graphs of data over time: The app renders clear and understandable graphs over time of any of the statistics on a team. For example, the image below shows teleop high shots of team 16 over all their matches: red indicates a miss; blue is a make.

Robot photos: The includes easy access to high quality robot photos taken by a student with the photo taking app (see next section). The photos help us to easily identify what role a robot is most capable of playing.



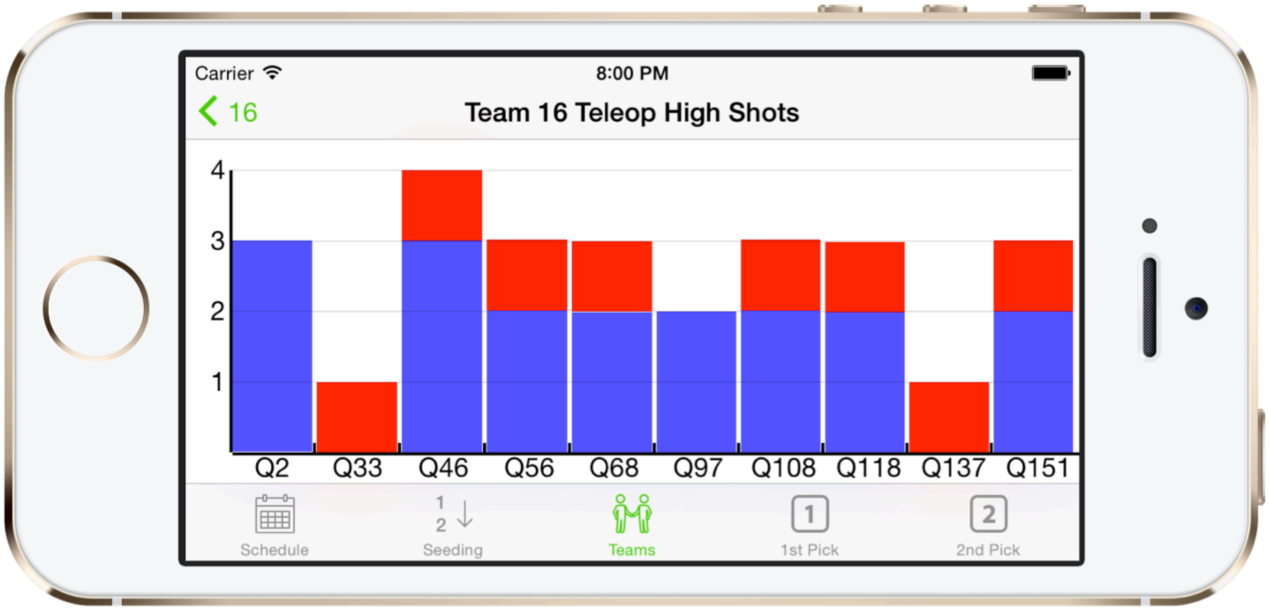
Carrier 3:50 PM

Match Schedule

Search

Q1	120	5171	3928	116 / 59
	1294	4500	3256	~~158 / 97~~
Q2	610	1138	2614	191 / 136
	1285	16	5057	~~220 / 204~~
Q3	111	58	5124	146 / 115
	612	133	330	~~167 / 209~~
Q4	4997	1574	2035	48 / 190
	1507	1126	3528	~~145 / 157~~
Q5	525	5016	3539	58 / 231
	3824	4039	1678	~~175 / 211~~

Schedule Seeding Teams 1st Pick 2nd Pick

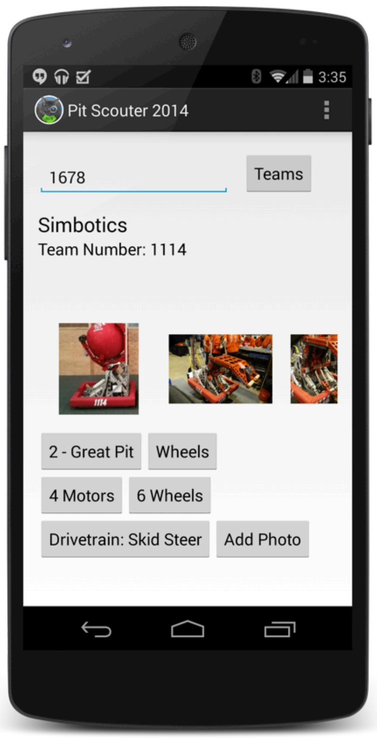


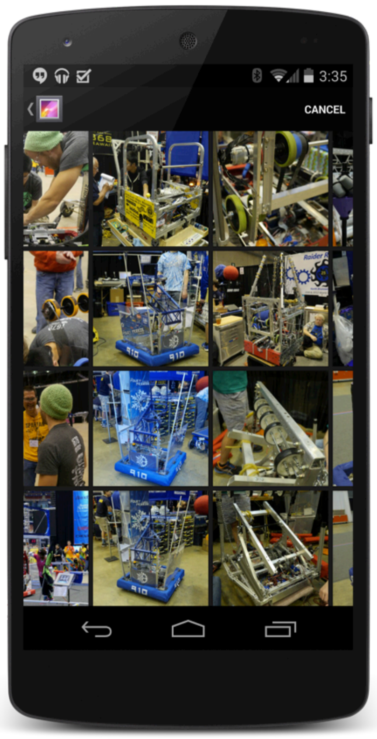
Android Photo Taking App: Native App, Programmed in Java

Key Functions:

Upload photos for a team: Using the app, the photographer can simply choose a photo from his photo gallery and then it will be automatically uploaded for the chosen team. No manual sorting of files is necessary. To upload photos, once again we use the Android Dropbox API, so in actuality the app just syncs each photo into the correct folder in the team's Dropbox

Pit scouting: The student photographing teams is also responsible for some preliminary pit scouting: he or she takes simple data such as type of drive train. Once the data is inputted into the app, it is automatically synced with the server.



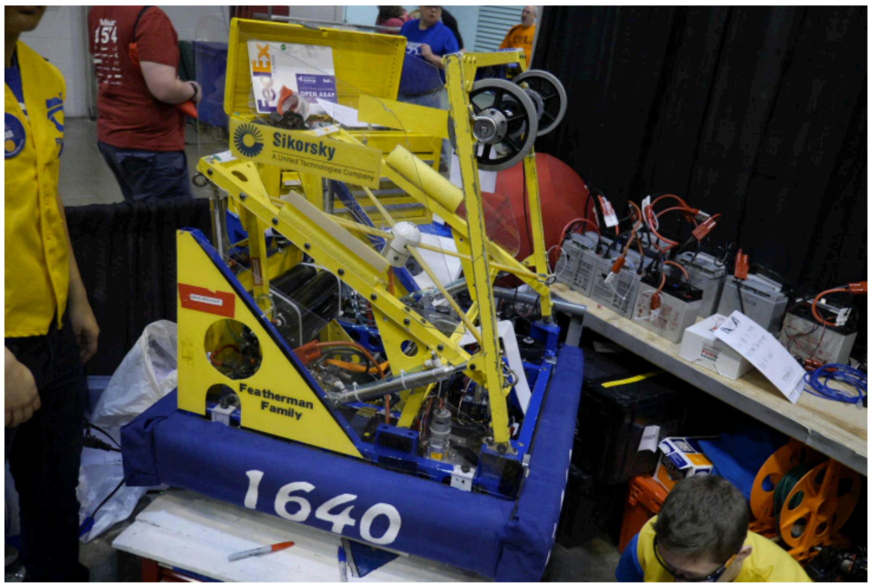


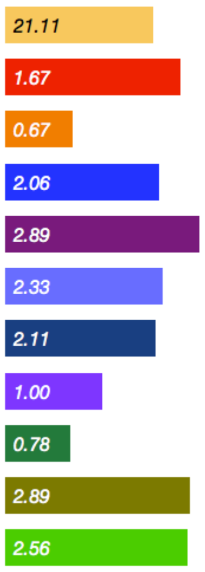
Printed Sheets

To make our pick list process much easier, we took inspiration from Team 971's format of printed sheets for each robot. Ours feature a large color photo of each robot (taken with the app above), and selected pieces of data that are most relevant to the selection process. We also include graphs of some of the data indicating relative strength to help more quickly compare robots. When we make our pick lists, we put these printed pages up on the wall, and then manually reorder them. Having the large photos of each robot really helps us to keep mental track of each robot.

The robot sheets are automatically created by the iPhone app: it renders the photo and text into a massive PDF document, and supports saving to Dropbox and wireless printing to FedEx Office. Below is one of the 100 pages that we printed at Championships.

1640





Offense:

● **34th** *(21.11)* Auto Points
24th *(89%)* Auto Accuracy

● **9th** *(1.67)* Tele High Shots
12th *(94%)* Tele High Accuracy
29th *(0.67)* Tele Low Shots
● **43rd** *(0.67)* Trusses
45th *(0.00)* Catches

Defense:

● **37th** *(2.06)* Overall:

● **14th** *(2.89)* Evasion
● **32nd** *(2.33)* Speed
● **39th** *(2.11)* Torque
● **71st** *(1.00)* Blocking

Swerve Drivetrain

Passing:

● **68th** *(0.78)* HP Receives

89th *(0.00)* HP Receive Fails
● **18th** *(2.89)* G. Receives
● **25th** *(2.56)* Ball Control

Notes:

3 Build quality
3 Acquisition zone
18 second Cycle time
Notes:
Swerve, kiss pass

Future Goals

Strategy App

This year was the first year where we put a significant effort into pre-match strategizing for qualification matches, and it certainly contributed to us finishing first in qualification matches in Newton and advancing to Einstein. In addition, we would observe all the other opponent alliances during eliminations, and propose the best strategies to stop their cycles. However, the strategizing took place on paper, which made it difficult to organize and difficult to communicate to the drive team. An app that allows people in the stands to suggest and diagram strategy and automatically send it to a smartphone or tablet in the pits would be both a fun and useful goal for next year.

Contact Us

We published this white paper hoping that by indicating the usefulness of the system and hinting at how to program it, we will encourage other teams to program their own scouting system. We are all in favor of helping other teams, but it is our philosophy that other teams will benefit much more in experience by programming their own custom scouting app rather than if we were to directly open-source ours and give it out to teams. However, we will do all that we can to lend a helping hand to other teams that are creating a scouting app, as we want other to be as successful as they can in the process. If you have any questions at all, feel free to email us at:

frc1678@gmail.com