Basketball Event Tracking Interface:

The Final Report

Written by George Zhang, Andrew Forsman, Alex Vann, and Mike Dillon

Problem Domain and Root Concept

Our initial vision was to design a game recording interface for professional basketball that is intuitive, efficient, and quick. The software is used to update fans on play by play game changes over the Internet. We knew from the beginning that basketball is a fast paced game; many crucial events may occur within seconds of each other. The recorder must be able to keep up with the pace of the game or be overloaded with too many tasks. If this happens, important game information may not reach the fans in a timely manner. Additionally, the data that the user sees may not be accurate. The software must prevent this problem by eliminating repetitive tasks and by allowing the recorder to generate events quickly and accurately. We agreed that the interface should be kept as simple as possible while still maintaining the robustness that is needed to capture the essence of all the possible events that may occur during a standard professional basketball game. The entire project is structured around this root concept. Before arriving at this conclusion, however, we did consider several other ideas.

Several other root concepts were considered but ultimately discarded. One was to provide a very in depth interface that has a large number of menu options that encompass a large number of possible outcomes. Another idea was to assign a recorder to each individual player on each team. The personal recorder is responsible for logging all of the events that happen to his assigned player. We also considered designing a speech to text interface that generates a play by play report by looking for key words. Another project vision that we considered was to stream an audio broadcast of the game from the recorder's point of view. Finally, we considered using a simple keyboard to manually type all of the events of the game. Although these root concepts were all considered, they were all subsequently rejected for several reasons.

When coming up with the root concept, we looked at and analyzed several different factors of recording basketball games. The first and most important is the amount of information that needs to be logged. In order to give the online viewer a full experience, every event that occurs during the basketball game should be reported. A play by play breakdown is necessary for those who cannot watch the game on television. The reporter is tasked with tracking each of these events. Given the vast number of possibly exciting affairs that can happen in a single game of basketball, it is very tempting to simply let the reporter type up each event. This, however, is very tedious and the reporter would undoubtedly get tired after typing for an entire game. Also, the reporter would have to repeatedly type in the names of the player who is performing the action; this is a mindless task that could be easily automated by software. In addition, this method will not be effective since basketball games are very fast paced and it would be difficult for even the most skilled typist to keep up with the action.

The action packed nature of basketball means that many significant events can happen in the time frame of several seconds. This is the reason why the interface should be simple and quick to use. If the interface was very powerful and had many buttons to input a vast variety of events, the reporter might be over encumbered and be slow to react. With too many choices, it may not be immediately obvious which one is right. Suppose, for example, Player A passes the ball to Player B. This information should suffice; the reporter should not have to specify whether the pass was a bounce pass, regular pass, or overhead pass. This additional detail would only serve to clutter the interface and bog the reporter down in details.

If we only consider these two factors, an interface that allows the reporter to input information through speech might sound like a great idea. In reality, however, this idea is not practical. The reporter is usually situated right on the sideline in order to get a close view of the game. This ensures that he won't miss any of the action and will be able to see exactly what is going on. Because of the reporter's position, any sort of audio input would probably be unsuccessful. This is due to all of the noise that is generated by basketball games. The coach, the players, and the audience are all yelling at each other. In fact, due to the enclosed stadium, it is sometimes difficult to speak clearly to someone sitting right next to you.

Considering these three factors in addition to technological and staff limitations leads to an obvious conclusion. The interface that professional basketball recorders use needs to be simple, accurate, and quick to use. Most importantly, the interface must also be able to record any significant event that happens in a game of basketball.

Requirements Analysis

In order to gather critical data about the requirements for the interface, we conducted an interview with our client and distributed questionnaires among friends via Facebook. These data gathering methods provided valuable insight on the requirements our software would have to meet. In additional to the interview and the questionnaires, we also analyzed current basketball event tracking interfaces to evaluate their effectiveness and judge how closely they match up to HCI standards.

Our client was a basketball enthusiast. He regularly keeps up with professional basketball games and sometimes records the major events of each game. His experience provided us with a lot of knowledge on the inner workings of recording basketball. We asked him several questions related to the speed of the game. He confirmed our initial belief that events happen very fast and need to be recorded quickly. We also discussed the significance of recording errors and the need to be able to go back and redact mistakes. Finally, we learned that the length of games makes it very stressful for one person to record for the entire duration. In addition to the interview, we also circulated a questionnaire to help us better understand what kind of information the general public expects to see on live online updates for basketball games.

After we developed the questionnaire, each member of the team used Facebook to hand it out. In total, we had 30 participants. The information we learned from the survey is quite interesting. Eighty percent of the people surveyed claimed that they would go online to view the game if they could not watch it on the television. Out of all the typical events that happen during a basketball game, scoring, fouls, and blocks were the three most important. Finally, sixty percent of the participants responded that they would view the information on a laptop. In addition to the questionnaire and the interview, we also analyzed a few artifacts of current basketball event tracking interfaces.

There are currently several solutions available for those who want to track events during basketball games. We examined each of these artifacts to learn more about what is currently usable. We looked at a paper recording interface, a digital interface, and a command line interface. We realized that none of these interfaces are ideal for basketball recording; they are either unintuitive, contain too many options which may bog down the user, or extremely outdated. Finally, after gathering all the information for requirements analysis, we concluded that our software must be quick to use, intuitive, cover a broad range of events including fouls, blocks, and scoring, and feature an option for correcting mistakes.

User Interface Design

When creating the concept for our product several factors from the interaction and information design were incorporated to make sure that our product was the most comprehensible and intuitive product that could be presented to testing volunteers. Many of these design choices that will be discussed later overlap from one area to another typically in terms of context and actually implementation. A fortunate fact for our group was the agreement early on that we knew exactly what our product should look like. From there, we simply explored different technologies that would make our finished product close to what we had envisioned in our minds.

A primary focus of our information design phase was the need for a product that would not overwhelm the user. To do so we began to think of different ways that we could incorporate needed information as well has hiding other information that would be too much for a simple user. To begin we started by sending out a survey that would check to see what sort of information end users who would view basketball stats online are invested in learning about. From there we were able to form an action event list that would help the recorder by providing the most commonly seen actions as input, all the while maintaining simplicity in the overall design. As we started focusing on simplicity of the product, we realized that this focus would translate into improving the accuracy of the product itself as well.

To increase the simplicity and accuracy of the product at the same time we focused on different ideas of how to hide information that might be inappropriate to the user for a given task. To achieve this we implemented information hiding that would hide buttons and actions that would not compile into a sensible task given the current context. We also decided to just show the crucial points that end users would want to see. This helped simplify the display and make the program seem less cluttered.

Lastly, we thought about error correction and how to display the different menu options throughout the program. To achieve this, we provided a buffer of the last recorded tasks for which the recorder could change and modify if he or she made a mistake. To present the information we focused on a way that would implement both information hiding and simplicity at the same time. The conclusion that we as a group came to was to create some form of a tree structure that would expand and retract given certain bits of information or the context of the task being preformed. The reasoning for this is that next events would be located near preceding events making locating events simple, events that did not make sense to show would be hidden so that no mistakes would be made. Overall, the information design phase of our project helped us greatly for structuring our final design and our prototype as well.

To incorporate all the information design choice that we made into our interaction, we began by thinking of an appropriate medium for our product to actually be on. We concluded that some sort of tablet interaction would be our best bet. Once we began exploring the tablet idea, we were able to pinpoint our idea that an iPad application would be the best fit. An application would be the easiest form for disseminate the product to potential users and cut back on overhead and production costs. In physical terms, the iPad design worked for several reasons that will be discussed in the next paragraph.

First, the iPad coding technologies were open source and easy to create a working prototype. Creating a application for an iPad also meant that it would be easier to send out the product to different clients that could access it through the Apple App Store. Secondly, the iPad provided the portability that we believed would be crucial for the product's principals. The iPad's display was also large enough that a fair amount of information and interaction could take place on. Lastly, the iPad's multi-touch capabilities also provided us with a great amount of different ways we could take advantage of the user's interaction with the product. Multi-touch meant that we could certainly implement technology that would be able to do multiple actions at once (given a user's skill level of course).

But more important than the actual multi-touch aspect of the iPad was the actual fact that we could work with a touch screen. This opened many doors for us. We decided to make use of this fact by implementing a dragging motion to record tasks and events. A user would start on a player and then drag his or her finger through the appropriate movements to record actions. A side effect of this is that since we wanted to include some form of information hiding, we attempted to implement technology that would recognize when a user dragged their finger over a specific action, other actions that would be out of context to include as possible actions. We also figured that with our tree design, combining information hiding with touch screen would provide smooth and easy movements for the recorder. Lastly, having the touch screen also allowed us to pinpoint location were basketball players would shoot from on the court. To achieve this, we created and implemented a pie-menu that would pop up around the user's finger where they indicated the location of the shot. This would not only increase the speed with which the user could record the action but also the speed at which the user could return to recording other parts of the game.

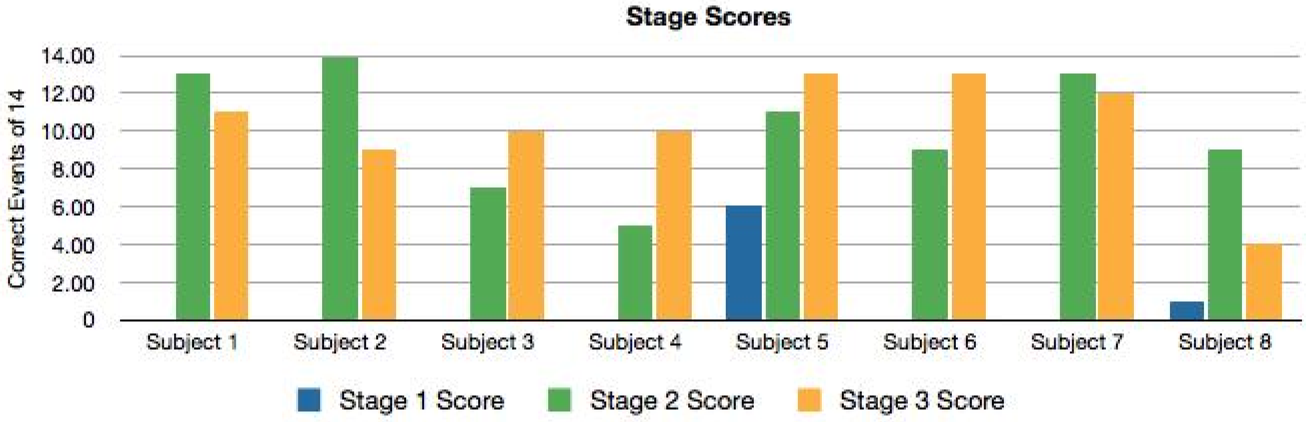
Evaluation and Redesign

The next phase of our project consisted of performing an evaluation of our system via our prototype. For this we chose to run an empirical study. In our study our participants were tasked with recording the events of a recorded basketball game using our system. In order to try and mitigate the effects of poor quality video recordings of games, we chose to make an audio recording of a game and have our participants record events based off our recording. This allowed us to closely monitor the participants accuracy and speed, however it somewhat reduced the practicality of the study.

The study we designed consisted of 3 stages: The first was what we dubbed the "exploratory" stage. In this stage the user was presented with a game recording and was simply told to record the events. The next stage was the "tutorial" stage, where we first gave the participant instructions and examples on the use of the system. In the third and final stage, we provided the user with a faster and more realistic recording. Each of these stages was designed to evaluate a particular aspect of the system. The first was intended to inform us of the intuitiveness of the system, observing the users ability to quickly learn the system on the fly. The second was to give the user experience with the system and evaluate how quickly the system could be learned. The final stage was designed to evaluate several other critical aspects of the system including the overall usability, the speed, reliability and ability to edit.

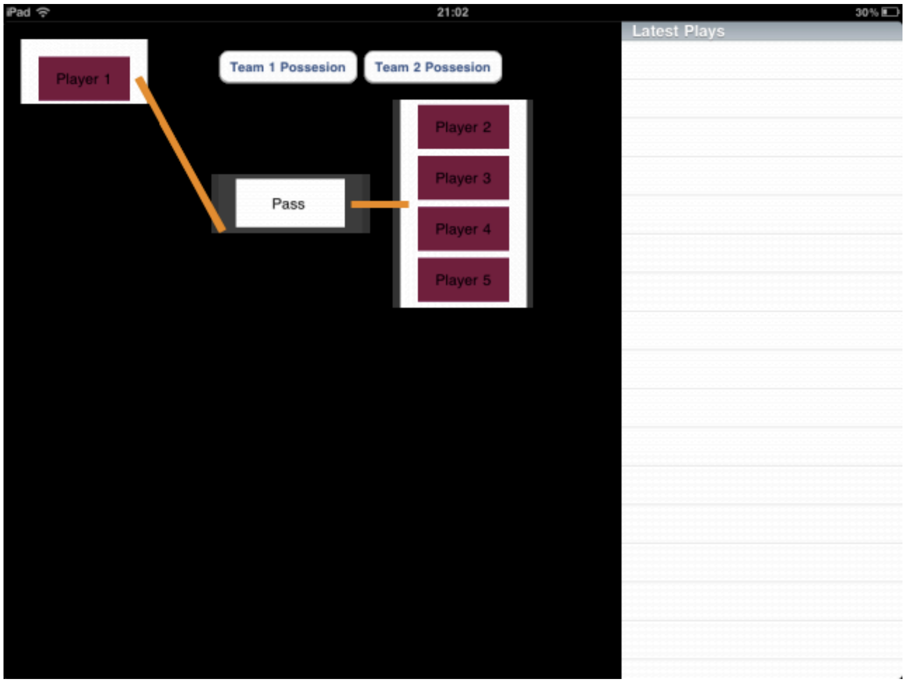
During each of the stages we recorded several metrics, including total completion time, successful event creations, uncorrected and corrected mistakes, and uses of edit. We chose these metrics because they allowed us to make inferences about and empirically evaluate our main goals . For example, if a user recorded a high completion time (still creating events even after the game recording was over) and also recorded no mistakes then this could (in combination with our post-surveys mentioned below) indicate things about the speed of our system. In addition to our metrics we also asked users to complete a pre- and post-survey. These surveys collected subjective data on the users' experience with the system as well as gave us relevant background on users to consider, particularly in the case of statistical outliers.

Our study was done with the help of 8 participants and proved to be very helpful and somewhat conflicting with our expectations. Most surprising to us was the lack of intuitiveness of our system. Only two of our 8 participants managed to create an event in the first stage (as can be seen in the graph below). Every participant first tried to use our system by tapping buttons. This fact is especially interesting when considering that most of our users had a moderate to high level of experience with touch screen interfaces and indicates that our "drag" selection mechanism clearly broke with many current and popular touch screen interaction patterns.



Many of our participants also noted in the free response section of our post-survey that they would often get lost midway through creating an event. For example, they sometimes thought they had selected player 1 when they had really selected player 2. Confusion as to what a participant had already selected was common amongst users and often lead to errors. The occurrence of this particular type of error exhibits a clear lack of feedback being provided to the user by the system.

After conducting our evaluation it became clear that we had several usability issues that we needed to amend. The first and foremost of these was changing the input gesture to a "tap" selection in input instead of a "drag" input. This change was our highest priority for our new prototype as it was greatly disliked by our study participants. The other changes we made to our prototype included information hiding, where options that did not make sense contextually were hidden from the user (i.e. if the steal button is pressed then only players from the same team as the first player are visible). We also created a tree-like menu structure where when one option is selected the next available options are displayed out to the right of the selected option, as seen below.



If more time had been available to us we would have liked to implement visual feedback features in our system. Such an example would be the highlighting of selected options as a user creates an event. We feel that adding this feature would produce a great addition in usabiloity of our prototype. Finally we would have liked to conduct an evaluation on this third iteration of the prototype using the metrics of our first evaluation. We feel that the third prototype version would have a great increase in usability and it would be interesting to see how it compares to our first.

Reflection on the Product

After much hard work, evaluation, and analysis, we came up with a product that is reflective of our original root concept. We set out to create an interface that lets recorders quickly, intuitively, and efficiently record play-by-play events as they happen during basketball games. Not only did we fulfill the goals of our initial vision, we also added more features based on the feedback we got from our client and our requirements analysis. Our final product, an application on the iPad, is very quick to use and intuitive. As seen by some of our evaluations, subjects were able to pick up the software and immediately start using it without any assistance. The test subjects also scored well on our tests which demonstrates that they can effectively carry out the tasks that they need to. Unfortunately, our evaluation did not fully test the usability of this system in the real world.

Although we met all of our initial objectives, the usability of our application in the real world is yet to be determined. In all of our evaluations, our tests were performed based on auditory input. In the real world, however, the recorder must be able to keep up with visual input. This makes a significant difference because the user now has to look at the game and at the software simultaneously. This makes the task at hand much more difficult. Even though this aspect of the software was not directly tests, we can deduce that a skilled user would have no problem keeping up with the pace of an actual basketball game. Before the software is released, however, we would have to perform tests to verify this assumption. In conclusion, we think that our application will be usable in the real world in the hands of an experienced user but we do not have conclusive evidence of this. Although can not confirm our claims on the real world usability of our application, we can confirm that it is usefulness.

We received much positive feedback from our client after giving him a demonstration of the final iteration of our application. He stated that it is a stark improvement over the other interfaces that are currently available. Unfortunately, the application is not connected with any kind of server that can manage live updates over the Internet. This means that the current iteration of our application is not very useful. On the other hand, when the application is finally finished, it will be a very useful tool for basketball event recorders. Our interface is a marked improvement over the current ones and allows the user to carry out his tasks much more easily than before. Additionally, our application does not require any complex or expensive hardware that needs to be set up and maintained for each game; it runs on the (relatively) inexpensive Apple iPad. There is no doubt that once the application is finished and fully functional, it will be a serious contender among other basketball event recording applications.

Although our application oozes with usefulness and practicality, there are several barriers that may hamper its adoption. First, we still have to finish a finalized version of the app that can interface with an online reporting system. Second, current event trackers may be hesitant to give up their old systems to put a new one in its place. No matter how many drastic improvements are made, it is sometimes difficult to fight against tradition. Lastly, much real world testing and evaluation needs to be done to fully confirm the usability of our application.

Our work on this interface is not yet complete. We still have work to do evaluating the usability of the application in the real world, but there is no doubt that we have designed a useful product. We have come a long way since we first developed our root concept. Looking back, it is obvious that we met our original goals and intentions. Overall, our work on this interface has been successful.

Reflection on the Process

For the entire duration of the project, we followed the scenario based design process to the best of our ability. We conformed very closely to the process and performed each step as it was described. We rarely, if ever, deviated from the scenario based design method. This process was a new concept to every member of the team. It is a significant contrast to the waterfall method that we are currently used to. These differences presented us with many new and interesting challenges. They also provided us with a lot of valuable insight from an HCI perspective.

One of the most significant “new” aspects of scenario based design was the inclusion of scenario writing and claims analysis. These stories and conclusions were very useful in determining the user’s tasks and potential problems he may face while using our software. The problems scenarios and claims helped us establish a solid connection with the users in a way that is not possible when strictly following the waterfall model. Additionally, the problem scenarios helped us highlight critical problems with the current situation which gave us a better understanding of what we need to fix. The claims that we wrote following each problem scenario also provided to be beneficial. They helped us categorize and objectively document the positive and negative features of the current interface. This enabled us to easily point out key aspects that should be kept consistent and flaws that should be fixed. Another important feature of the scenarios and claims was that i was pervasive throughout the entire process. This definitely kept us on track as we progressed through the various design phases. It also made us approach each situation with a much more user oriented perspective. Writing problem scenarios and claims analysis, an important element of scenario based design, was a great benefit to the project. With this in mind, all members of the team agreed that they would potentially use scenario based design in future work.

Scenario based design has many applications in the real world. According to our textbook, “The basic argument behind scenario-based methods is that descriptions of people using technology are essential in discussing and analyzing how the technology is (or could be) used to reshape their activities. A secondary advantage is that scenario descriptions can be created before a system is built and its impacts felt.” This quote from Rosson and Carroll is a good summation of why we are likely to use this process again for system design or usability engineering task at our jobs. In the real world, however, it is unlikely that strictly following the scenario-based design paradigm is the most practical approach. Indeed, in actuality, the best approach would probably be a combination of scenario-based design and some other method. It would draw out the best features of scenario-based design such as an user oriented approach and prototyping . In future work, we will definitely take out the best aspects of scenario-based design and inject it into our design process.

The use of scenario based design has definitely resulted in a better product. We had the user in mind for the duration of the process. This has resulted in an end product that is not only more usable, but also contains all the features that the users need in order to carry out their tasks successfully. In the end, following the scenario-based design process has taught us a lot about several fundamental principles of human-computer interaction and provided us with valuable knowledge that will be essential as we pursue our respective careers.