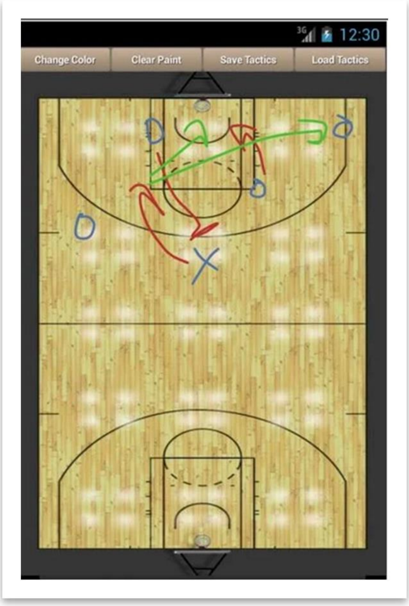
Dimitri Cavoulacos Thursday February 06, 2014

Computer Science 490 Project Description and Proposal

There are generally two approaches on how to choose a project topic for Computer Science 490: selling a topic to a professor or having a professor sell one to you. The reviews of the course strongly recommended the first approach, and stressed how important it was to choose a topic you were passionate about. I am passionate about basketball, so I chose to spend my senior project creating an application aimed at helping basketball coaches create, replay, edit and share pre-choreographed, set basketball plays. Ideally, this application would be used on a tablet device but given the time and scale allowed by a one-semester senior project, I decided to focus first on achieving full functionality on a personal computer.

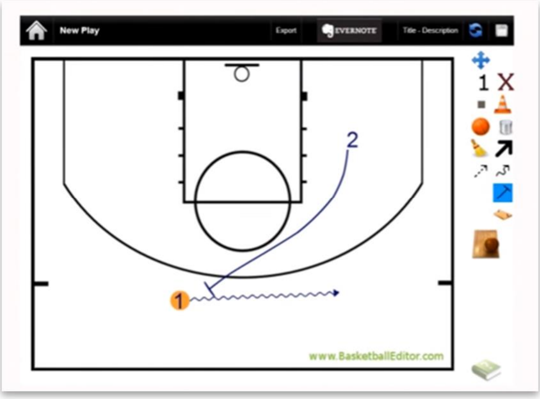
Currently, the overwhelming majority of basketball coaches across all levels of play use whiteboards and markers to draw up set plays during games. In addition to being the traditional medium of designing set plays, the whiteboard allows coaches to design and edit plays quickly as well as accurately. The goal of this application would be to replicate the ease with which coaches can currently draw their plays on whiteboard, and to add the element of saving, sharing and editing pre-existing sets. There are a few existing applications that do exactly this, notably Hoops Clipboard, Basketball Playbook Coach, Basketball Coach and Basketball Coach Board among other similarly named examples. These applications fall into two groups: the ñpure whiteboardò apps and the ñmenu barò apps. The ñpure whiteboardò apps simply display a basketball court and allows the user to draw plays in the same way one would on a classic whiteboard. Players are drawn, and notes can be written on-screen as well.



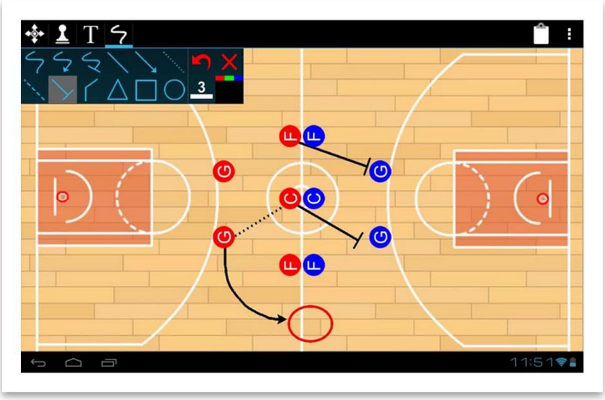
Screenshot of Basketball Coach Board[1]

Alternatively, ñmenu barò applications trade the classic drawing approach and its full control of the final display for greater accuracy and more manipulable board. The players are created and have explicit, changeable positions. The range of actions performed by any player (i.e. dribbling, running, passing, setting a screen) are always represented by different types of lines (i.e. squiggly for dribbling, straight for running, dashed for passing, ending in a ñTò for setting a screen). Traditionally, these are drawn explicitly, but ñmenu barò applications owe their nickname to their use of menu bars similar to Paintôs. These menu bars allow the user to select the type of action to be performed before drawing it out. This gives a cleaner final result, but the interface can feel cluttered or unprofessional.

[1] https://play.google.com/store/apps/details?id=com.coachboard.basketballcoachboard

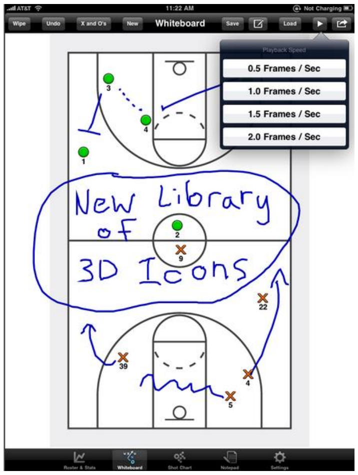


Screenshot of BasketBall Playbook Coach[2]



Screenshot of Basketball Dood[3]

One application, Hoops Clipboard, attempts to bridge the gap between ñpure whiteboardò and ñmenu barò by combining the latterôs use of player icons with a formerôs play-design approach. The difficulty here is translating the hand-drawn commands into computable commands to be carried out by the player icons. Given what currently exists, I determined that I would attempt to create a more intelligent application that would be more aware of the state of each player. For instance, if a player has possession of the basketball they can only dribble, pass or shoot. A player without the basketball can only run and set screens. This would allow for users to quickly move the player icons to their desired position and for the application to do all the work involved in drawing the correct line.



Screenshot of Hoops Clipboard[4]

In addition to creating and drawing plays, this application should also allow users to replay, edit and share preexisting set plays. The replay and editing are extremely important, as a strong design could completely change how coaching staffs work with their players while a weak design could hurt the application and greatly lower its utility. My proposal is to define a set play as a series of ñaction framesò, in which one or more actions take place, and any actions occurring in a single action frame are considered to be simultaneous. This structure allows for smooth replays as well as ñframe by frameò replays. Furthermore, using this action frame structure opens the doors for a wide range of features, such as a set play with multiple options in the form of a branched collection of action frames, or an intelligent defense feature that treats each action frame independently and moves the defensive players automatically. Sets with multiple options are more realistic a goal for this project than intelligent defense, but the primary directive would be to build the application such that these features could be incorporated smoothly at a later point in time.

[2] https://play.google.com/store/apps/details?id=air.basketballeditor
[3] https://play.google.com/store/apps/details?id=air.BasketballDoodFree
[4] http://www.hoopsclipboard.com/

Sharing a set play should be relatively straightforward, but it does open itself to interesting possibilities. The creator of a set play can always view and edit their own plays and initially, sharing would simply allow another user to view a play. The first interesting possibility is to allow a user to be associated to a player icon on the whiteboard. This would allow a coach to design a play, send it to his players through the application for them to study. The players can then link or connect to their corresponding player token and focus on their movements first and foremost. The inclusion of a visible timeline of a set play in the user interface would greatly help, and would keep doors open for more sophisticated features later on. Moreover, the ability to link user types and player icons would be very useful in the integration of statistics if/when the application reaches that point.

One big concern I encountered was the real-time, end-of-game uses of the application. In order for it to be considered successful, it must perform at least as well as a traditional whiteboard in any context, and the context of a 20 second timeout is extremely important. This is when play designing must work quickly, accurately and free of error. It is therefore extremely impractical for a coach to position all 5 players by hand and then have to click a button for ñnext action frameò after every sequence. This can be remedied by treating the entire set play as a single action frame and replaying it smoothly. It can then be saved and edited later, where the coach can determine where to place the action frame cut-offs, and this could be done similarly to how Lisp uses car and cdr. The problem of the positioning can be solved by creating a special type of action frame, ñinitialò or ñloadableò frames, which can be used as a starting point for a new play. These special action frames must be designable, easily accessible and must account for inbounding plays (where one player is out of bounds, passing the ball into play).

In order to create this application, I plan on using Ruby for the court interface and Rails for the application features as the initial goal is to achieve functionality on a personal computer. There is a ñ2D game development libraryò available online called Gosu[5] which I plan on using as well. Gosu is available for both Ruby and C++, the latter also being available for iPad, iPhone and iPod Touch. Given that this application would ultimately be used on a tablet device, Gosu could allow for an easier transition. Before such a transition occurs, a tablet emulator will be used to ensure a minimum usability. Should I come across a language, library or overall approach that is better suited to optimizing the application, I will of course shift the project to include it.

Given that there are many aspects to this project, it is important to have a progress timeline and deadlines for each step of building the application. The timeline I have decided on is the following, with each step completed by the end of its assigned week.

Week of February 24th

Allow creation of action frames possessing five players, each player having a position on the court and a Boolean value for ball possession. Players should be able to move to a new position *or* pass/receive the ball in one action frame.

Week of March 3rd

Allow creation of set plays composed of one or more action frames where each playerôs final position and Boolean value in one action frame is carried over to the next action frame as the

[5] http://libgosu.org/

playerôs initial position and Boolean. Display the initial state of the action frame and update the display after each action (move/pass) with the appropriate line: dotted for pass, normal for moving without the ball, squiggly line for moving with the ball.

Week of March 10th (spring break)

Include the ability for a player to shoot, with the conditions that only a player in possession of the ball can shoot and that shooting the ball automatically ends the set play. This also requires setting a hard limit of players in possession of the basketball to one.

Week of March 17th (spring break)

Include the ability for a player to set a screen. Only players without the ball can set a screen, and should be able to move *and* set a screen in a single action frame if the player moves first. However, if the player sets a screen without moving they cannot move until the next action frame.

Week of March 24th

Expand player action creation from manual input to graphical interface interaction. Start with simply moving with and without the ball by dragging and dropping player type icon on the display.

Week of March 31st

Expand player action creation further to include passing and setting screens. Setting a screen could be done by double clicking the player type icon, and passing could be done by dragging the player with the ball onto another player; the exact approach will be determined as the project advances.

Week of April 7th

Expand player movements beyond straight lines such that they follow the path along which the player type icon was dragged. Also begin animating player actions, moving the player type icons smoothly from initial to final positions instead of ñsnappingò from one to the other.

Week of April 14th

Finalize animation for all player actions types. Also implement various personalization settings such as different court types (high school, NCAA, NBA), team colors and the like.

Week of April 21st

Begin Computer Science 490 Special Project write up.

Week of April 28th (final week, ends Wednesday)

Complete write up and submit project.

My hopes for this project is to use my passion for all aspects of basketball to build a fully functioning, truly useful and well-designed application. I have virtually no experience working with graphics or graphical user interfaces so I expect that to be the largest hurdle I will have to face. Furthermore, there is a lot of room for complexity in this application, and the basketball fan in me wants to fit all of it in. I will have to plan many steps ahead when building in order to account for all the future functionality that I envision. This will be my first project that is wholly my own from inception to completion, and I think that will be the single greatest benefit of this Computer Science 490 Project.