

# Dart Scoring App

---

*Neal S. Keating*

*The George Washington University*

*CSCI 4237 – Handheld Devices*

*December 13, 2010*

## Table of Contents

Introduction .....	3
Background of Cricket.....	3
DSA Architecture.....	3
Cricket Class .....	3
Menu Class.....	4
Scoreboard Class.....	4
Board Class.....	4
Challenges Encountered .....	5
Scoreboard Class / Live Mode.....	5
Board Class / Digital Mode.....	5
Conclusion.....	6

## Introduction

The Dart Scoring App (DSA) is a software application programmed in Java 2 Micro Edition (J2ME) and utilizes Connected Limited Device Configuration 1.1 and Mobile Information Device Profile (MIDP) 2.1. The application gives the user two distinct ways to enjoy the classic dart game Cricket: Live mode and Digital mode.

Live mode enables the user to score along with a real life dart game by registering the scores input by the user. DSA registers the users touch on the scoreboard, calculates the correct score, and then checks to see if there is a winner. The user is able to rest the scoreboard at any time.

Digital mode is a two player game that allows the user to “throw” digital darts at a board background. The user slides their finger from the bottom of the screen towards the board and DSA registers the power and line of the throw, calculates the point of impact, and places a dart in the correct spot on the board. Player’s turns and correct number of darts are enforced by DSA. The scoreboard function described in Live mode above also scores this mode.

## Background of Cricket

Cricket is a two player dart game that utilizes a standard 20 number dartboard with triple and double rings. The goal of the game is the “close” each of the targets and have the highest score. One closes targets by hitting three “marks” on that target. A mark is denoted by a successfully hit in the target region. If a player has already closed a target and hits that target again before the opponent successfully closes the target, then the player receives the number of points of the region (i.e. if you hit the 15 you get 15 points, if you hit the 19 you get 19 points). The players alternate three dart turns until both aforementioned goals are accomplished simultaneously.

## DSA Architecture

DSA was developed using the Sun J2ME Software Development Kit 3.0 and tested exclusively using the J2ME Wireless Toolkit (WTK) emulators, specifically DefaultFxTouchPhone1. The application is a collection of one MIDP class and three runnable GameCanvas classes. The folder containing the pictures for the sprites is the only resource added to the project.

## Cricket Class

The Cricket Class is the main MIDP class of DSA. Cricket instantiates each of the runnable GameCanvas classes then passes control to the GameCanvas threads to do most of the heavy lifting. While the GameCanvas classes are running, Cricket is continuously waiting for a Boolean indication to destroy the app. When the Boolean comparison is triggered the app is destroyed. Cricket also has three public static void methods to switch between the active GameCanvas threads.

## Menu Class

This class controls the app architecture and menu structure of DSA. When the Cricket class is finished with the instantiation of the GameCanvas classes, it passes control to the menu class which displays the main menu screen. The main menu screen has four options: Exit, Live, Digital, and Options. Each of the commands are handled using the `pointerPressed()` method. Pressing the Exit option will trigger the Boolean that the Cricket class is waiting for, therefore causing DSA to exit. Pressing the Live option will transfer control to the active instance of the Scoreboard class via `Cricket.callScoreboard()`. Pressing the Digital option will transfer control to the active instance of the Board class via `Cricket.callBoard()`.

Selecting Options will change the mode of the menu class (cleverly called a “bottle” instead of mode because there just so happened to be an empty Crown Royal bottle in front of my while I was coding). Each bottle is essentially a full screen sprite that I set to visible when I need and hide when I don’t. Each bottle also has a unique set of `pointerPressed()` commands. This is also how I implemented the About and Help features.

## Scoreboard Class

The Scoreboard Class encompasses the entirety of my project as envisioned at the time of the proposal. This class runs the scoring function for DSA and once again uses the `pointerPressed()` methods to capture events on the device’s touch screen. When control of DSA is transferred to the Scoreboard class, it starts listening for touch events. When an event occurs, the class determines the proper score based on the (x,y) coordinates of the event. Arrays are used to tally marks for each player and total score is tracked. After each event the `checkForWinner()` method is called and if someone meets the criteria a winner is declared. All of the changes to the mark total and score are transferred to the screen via a series of sprites.

## Board Class

This class is something I added to DSA after the proposal to enhance to functionality of the app and hopefully my knowledge of J2ME. This is also the class that gave me the most trouble. This class allows two players the virtually throw darts at a board background. When control is transferred to the Board class, it starts listening for touch events. This time the class listens via both the `pointerPressed()` and `pointerReleased()` methods. When the user throws a dart, the class uses the vertical distance between the points to determine the power of the throw and uses the horizontal distance variance to determine the lineup of the throw. Once power and line are determined, the impact (x,y) is calculated and a dart is placed at the impact coordinates. The hard part is then determining where the impact coordinates are on the board and what score to assign the impact. After some fun math, the score is determined then passed to the Scoreboard class which scores this game the same as normal Scoreboard only game.

## Challenges Encountered

I encountered no shortage of challenges on my endeavor to complete this project, especially after I decided to add the Board class/Digital mode to DSA. This section discusses a few of the more important and time consuming ones, but is by no means an exhaustive list.

### Scoreboard Class / Live Mode

The only notable problem I encountered in when implementing the original Scoreboard/Live Mode functionality was creating multiple concurrent threads and passing control of the application back and forth between the threads in an orderly fashion. Up until this point in the class i had only created programs with a single thread. At first i encountered problems when trying to nest tread creation within GameCanvas classes and I eventually settled on instantiating all threads in Cricket and creating public static methods in Cricket to handle passing the control of the MIDlet.

### Board Class / Digital Mode

Most of the problems I encountered while designing the Board/Digital Mode centered around two main things: determining where(and if) the dart hit the board and if so, what score should be assigned to the dart.

In order to determine if the dart hit the board, DSA must first determine the impact (x,y) coordinate of where the dart will impact the board. I attempted to use the power and line attributes captured from the User Interface (UI) and calculate trajectory using the formula for the height of a projectile at an arbitrary distance:  $y = y_0 + x \tan \theta - \frac{gx^2}{2(v \cos \theta)^2}$ . While I believe I implemented the equation correctly, I was unable to get consistently accurate values. Therefore I settled for a more simplistic approach. I used the power attribute as a percent of 100 and divided the board into 10 bands. Each of the bands where then broken down into 10 subsections and the attribute was directly mapped to the board. A similar approach was then taken for the line, using a direct mapping methodology to determine the variance from the centerline.

Once the impact coordinate is determined, DSA then has to figure out if that coordinate is on the game board. This is done by calculating the distance from the center of the board using the Pythagorean theorem. Knowing the impact coordinate, we can find the opposite and adjacent sides of a right triangle formed with the x-axis. If the length of the hypotenuse is less than the radius of the circle then the dart hit the board.

Once DSA has determined that the dart hit the board, then the biggest problem of all occurs: what score gets assigned to this dart? To answer this question I went back to my trusted principles of trigonometry and determined I would use the same right triangles I created to determine a board hit. I can use the arctan of the interior angle and compare with an axis and determine the slice of the board associated with the impact. Unfortunately, J2ME's Math class only implements traditional trigonometric functions and not their inverses. After multiple attempts to create my own arctan class or find other suitable trigonometric functions to accomplish the task, I decided to utilize a reusable arctan library from Richard Carless' mMath.java Package. One Additional challenge in this was finding out that actions in each of the four quadrants will have to be handled differently (at first, my darts only landed in Quadrant II).

## **Conclusion**

Overall I am very pleased with the final outcome of my project. The App accomplishes much more than I outlined in my proposal, including an entire new mode of operation. DSA is fully functional and achieves all of my initial goals, but going forward I intended to improve the graphics and port it to Android and potentially IOS.