

CS 499 Senior Project
Project Assignment
Dart Premier League

Rationale

The English Dart Premier League draws large crowds. It is difficult for the audience to see the dartboard and keep track of the status of the match. This scoreboard will allow the audience to see the match status and running match statistics. The system will keep all match statistics and allow for review of competitor statistics.

Targeted to provide real-time scores for the Premier League darts competition and management of the League <https://www.bbc.com/sport/darts>.

See an example match here: <https://www.dailymotion.com/video/x79pua4>.

The rules of darts are explained here: <https://www.youtube.com/watch?v=gXIXNshaVzk>

This is a four (4) group project.

Features

- Allow for league management and maintain competitor data. Maintain players and player information.
- Set-up for a match – Select competitors, scoring leg start value, number of legs, number of matches, initial running statistics to display, date of match, location of match, official names, layout for stadium scoreboard.
- Scoring for Leg starts at 801, 501, 301 and goes down to exactly 0 and the winning dart has to be a double.
- If the player is on track for the minimum number of throws to win the leg it will be displayed on the scoreboard. Once the player no longer can win in the minimum number of throws the display shall be removed.
- When there are 170 or less points remaining, the program is to calculate and display the winning throws. The largest double possible should be the suggested last throw. This is in case there is a miss there are many remaining doubles to try for.
- The scorer's interface will consist of a resizable dartboard that the scorer can use to mark the thrown dart location. Note there are bounce-outs, knock-out and fouls that the software must take into account.
- The stadium scoreboard is to be resizable and can be modified by the scorer during the match without having to bring the stadium scoreboard down.
- User selectable best of X legs (typically 14).
- During championship play the software may take in account matches. Best of X matches, typically 4.
- Data for each dart thrown are to be maintained which can be reviewed.
- A canned set of in-process match statistics can be selected by the scorer and displayed on the main scoreboard and changed during the match. For example: Average turn score, number of 180's in match, lowest turn score, etc.

CS 499 Senior Project
Project Assignment
Dart Premier League

- A canned set of player statistics can be selected by the scorer and displayed on the main scoreboard and changed during the match. For example: Current league rank, last match or championship win, average league score for the season or lifetime, number of 180's in current match or season, etc.
- The system should be designed in a manner so that scoring can be accomplished in real-time concurrent with the match. The dart scoring interface should be as efficient as possible. It is allowed to have a spotter for the official scorer that will call out the throws made by the player during match play.

Note: Data does not have to be stored in a “database”. This is an implementation or design decision that needs to be made by the team.

Constraints

The choice of programming language and database is left up to the design team.
Users must be able to run the program on a PC under Windows.

The system will be considered to be open source and in the public domain, therefore all code must be original and may not include any copyrighted material.

CS 499 Senior Project
Project Assignment
Dart Premier League

Student Feedback from previous semesters.

The following information was provided by students at the end of the term and should be considered as lessons learned. Some of this information may or may not be useful because the project requirements may have changed from previous terms.

Fall 2019

Work on drawing the dartboard first (plus the score zones), it's the major portion of the assignment.

I think that reviewing linear algebra (Atan functions, 2D linear distance formula, etc.) will help in the design of the dartboard interactions and determining where a dart hits. I would also recommend not using Java's Swing package for the GUI as it is limited when it comes to drawing and layout manipulation.

Know ALL of the rules before you start writing code, to save yourself the trouble of having to go back and fix code because it doesn't follow the rules.

Graphics are easily the most difficult thing in this project. If you do not tackle having a working, re-sizable, GUI early on it will sneak up on you later in the project.

Spring 2020

Start with the re-sizable dartboards and the database creation. While QT was overall a opportunity for our team it had a steep learning curve to it, so make sure you are comfortable with the tool before you begin the development in earnest if it is the environment you go with. Also it would help to give yourselves a reasonable amount of time to understand proper GUI design. The back end would probably be easier to implement with a more flexible environment/language. Be extremely familiar with the rules of darts its is easier to implement the rules once rather than having to go back and fix it later.

Carefully research your options. Many languages and tools will work, so consider what you and your group are comfortable with as well as the strengths and weaknesses of your language, IDE, database system, version control, et cetera. The dartboard itself is unquestionably the hardest part of the project, but remember it isn't all of the project, and you'll still have a lot of work ahead of you once it's done.

This project has been done successfully in C++ with QT and Java. Purportedly, it was a little easier to do in C++ with QT. We had a few issues with QT's reliability, though, so research carefully.

CS 499 Senior Project
Project Assignment
Dart Premier League

The four hardest things you're going to deal with are (in this order):

1. The clickable, resize-able dartboard.
2. The GUI
3. The database
4. Correctly implementing dart rules

Java will make you cry unless you have some serious skill with it coming into this. C++ was rough but doable for us. Find a tool that makes your GUI/Dartboard easier to make; we used QT. Find the correct dart rules and verify them with the professor before it's too late.

This project has a lot of interrelated parts that will need good communication to work with.

Fully researching the language, API, and IDE that you use are of paramount importance. You want to make a choice and stick with it. Look for a combination of those things above that allows for making a program that responds quickly to user input (across multiple windows), allows for resizing of the windows (which should resize the elements inside them), and is not too difficult because there will be a lot of code to write. We had over 55 files when it was all said and done. Also, consider how you will deal with the persistent data necessary to accomplish the league management side of the project. We created a local database file for that information. Much consideration should also be given to the medium your team will use to communicate (maybe Zoom or Discord), as well as what you will use for version control (maybe GitHub).

We suggest not using Java Swing or MFC. These options make implementation of the dartboard very difficult. Python could make the backend easy, but what about the dartboard? The dartboard is a very significant portion of this project and should not be neglected - in fact, take care of its implementation first (will you have to refresh your skills with polar coordinates?). Overall, the team must consider which side of the project to make easier - frontend or backend - while not letting either of them slip through incomplete. We split our 4-person team into two sub teams, one for frontend and one for backend. We all came together during integration and brought it all together.

Definitely give consideration to QT (look it up if you haven't heard of it - it's worth it) and it's IDE - QT Creator. If you use that combo, the QT mechanism known as slots and signals will be your best friend throughout this project to make any piece of the GUI responsive to user input. For the database, use your own preference, but definitely consider SQLite for localhost data storage.

Decide early on what your group's leadership style will be. Will you be like some groups we've seen and switch the team lead role every week or will you do what we did and have one team lead the whole time? If you have a team member with a strong vision and leadership potential, consider the latter; if you do not, consider the former. Whichever you choose, it is vitally important that you keep the same creative vision in-place from week-to-week. Do not change

CS 499 Senior Project
Project Assignment
Dart Premier League

your GUI design every week on a whim because of a new vision. Choose a design and stick with it - you will be glad you did. The most important thing in this project from an administrative standpoint (as with any project) is, after all, consistency and cohesiveness of implementation and design.

Your design will vary so drastically based on which tools and which language you choose, it will be unexpectedly difficult to make any of the early design documents (the whole first half of this course) until you make these crucial tool decisions. Consider assigning one person to be tools lead and make reports on various options and have the team read them and make a group decision everyone can agree with.

Good luck! Stay frosty and the dart league will be in great hands!

Spring 2021

My advice would be to make sure that you choose the tools you are working with very wisely. One of the big requirements is that everything be resizable - this would have been unnecessarily difficult to accomplish simply working with a prepackaged GUI library and it's exactly why we chose PySide2 (Python version of QT) to implement the GUI. Likewise, the necessity of managing players was exactly why we chose Django for our backend - it allowed us to focus on working with the controller and developing an API for the front-end to interface with the backend to implement the requirements of our software. My understanding is that past teams have struggled to implement this project, but for us it really came down to designing the project wisely upfront within the first week or two and then sitting down and doing the implementation, allowing the libraries and tools we chose to work with (in conjunction with the requirements) guide our development process. The MVC architecture really served our team well in this project.

Another point of advice - watch some darts!!! I'm not kidding, sit down and watch the premier league world championship or something similar. You have to understand the problem that you are trying to solve, and not watching darts or looking up the rules of darts is the most surefire way to create problems for yourself in your backend or API down the line. Do not assume you understand the requirements just from looking at them because there's a lot of minutia that gets left out. Also when you watch that darts match, look at what the scoreboard is displaying and take notes, it'll help you figure out things that you need to implement.

Lastly, set your timeline very optimistically. We planned to have our project done around the end of March and wound up finishing it only a week before it was due in April (which was still early) because of bugfixes that caused other bugs. Front-load your work schedule and appoint someone who is okay with being annoying to be a project manager (be that the team leader or someone else) who is in charge of making sure that everyone is doing the work they should be doing. Schedule hackathons on weekends where you churn out 8-10 hours of work. You'll thank yourself later for doing this so that you're not facing crunch time around finals and midterms.

CS 499 Senior Project
Project Assignment
Dart Premier League

The feedback from previous years that we received was greatly helpful with initial design decisions. The dartboard graphics were the hardest feature to implement and we based our GUI framework off of this. Using Python and QT provided an easy way to implement the dartboard and its collisions.

Also, we made a good decision not to pursue a web application design. After looking at the requirements, we decided that a desktop based application made the most sense for this project. With the requirements of a scorer and scoreboard application, it resembles a "Big Screen" application at a sports game where fans can see information about the game on a screen alongside their own personal viewing of the game. This form of implementation is simpler and does not require internet access for users to see the scoreboard.

Follow the advice of the other classes and use Python for the backend and the frontend with pyside2 etc.