Draft 2: Design Part 2 Course Work:

(New success criteria)

Edit the success criteria from analysis to include more of the features you implemented to give yourself top merit

Optional Further Problem Analysis:

Here I intend to explore the problems that must be tackles in a more technical sense to flesh out the specific nature of the problems to be tackled:

Minimax:

* British Museum not possible
* Use heuristic to make the problem tractable

Design Objectives:

These should be achievable and measurable goals and technical requirements of the solution you devise in order to meet your success criteria

Better to have many singular and specific

Proposed Solution Description:

Here I will describe the proposed solution by use of a very macro decomposition of design objectives. I will describe what general components are needed and how they will interact.

I will describe the benefits of this macro decomposition and I will designate how each macro component will solve a subset of the design specifications.

I will also describe the responsibilities of each of these components in terms of INPUTS, PROCESSES and OUTPUTS.

Diagrams of propose solution:

* Data flow diagram
* Menu flow diagram
* Limitations of solution proposed (hardware and software)

Decomposition of Chess Engine (not algorithms just functions, classes and data structures)

Link back to design requirements

Decomposition of GUI logic and API client server connection:

Link back to design requirements

Chess Engine All major algorithms

Discuss the use of layers of abstraction in the use of classes

Emphasize:

* Board state class (layers of abstraction)
* Game class (use of database)
* Minimax function (optimisations)
* Add flowcharts and pseudocode
* Ensure you explain how this helps meet a design specification

GUI logic and API client server connection:

\* Discuss validation

\* Discuss how data is transferred (diagrams of API etc)

\* Use of database

* Add flowcharts and pseudocode
* Ensure you explain how this helps meet a design specification

Describe database use (save games and minimax cache) and other persistent storage:

* Include UML diagram and ensure structure is normalised

Variables and key data structures used in these key algorithms:

Create a table of key data structures and variables and include there data type as well any key validation

Complete trace tables and dry runs of key algorithms:

Try and trace through these key algorithms:

* Some vector calculations
* Piece movement: Trace through the algorithm for a specific piece on a specific board states
* Then trace though check function assuming that piece movement functions are all working
* Then trace though generate legal moves, assuming check function and piece movement working
* Then trace through game over
* Try to trace though with alpha beta pruning, a basic chess board that it towards the end to show the optimal move

Test Plans

Functional white box testing during development:

Describe tests that were completed during each iteration

Post development testing against development criteria (black box alpha testing):

Acceptance testing for success criteria (beta testing):