CSSE220: Example Design Problem

For maximum benefit, I encourage you to attempt to solve these problems yourself before peeking at the solutions document.

This document includes example problems and space for you to write in solutions. In every case, the instructions are:

1. Identify the problems with Solution A & B using your design principles
2. Design a new solution that solves all problems

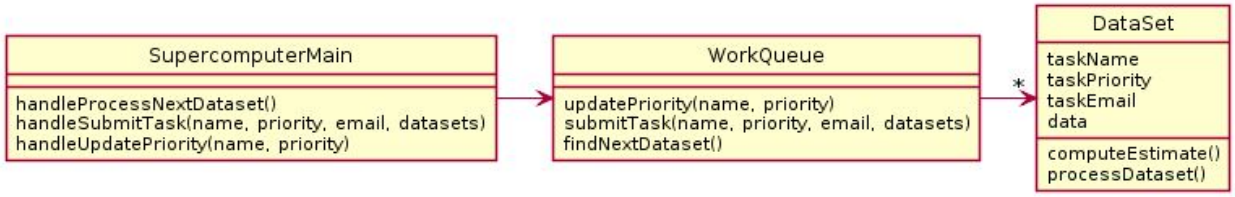
List of Design Problems:

1. Supercomputer
2. Announcements
3. State Hospitals
4. VideoGame

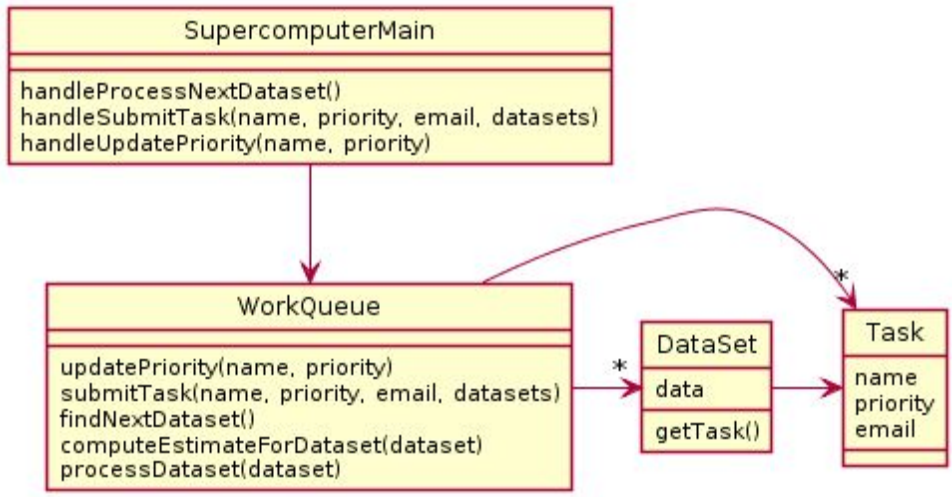
Supercomputer

The astronomy department maintains a supercomputer that everyone wants to use. Astronomers submit tasks to the supercomputer that consist of a series of datasets that must be processed independently. Each task includes a name, priority, and an email address where the results should be sent. Each dataset just has data. Given a dataset, it is possible to compute an estimate of how long it will take to run. The department agrees that the supercomputer should process datasets in priority order, and when priority is equal the supercomputer should select the dataset with the smallest estimate runtime. However, it must be possible to change a task's priority after it has been submitted.

Solution A



Solution B



Your Solution: Problems With A

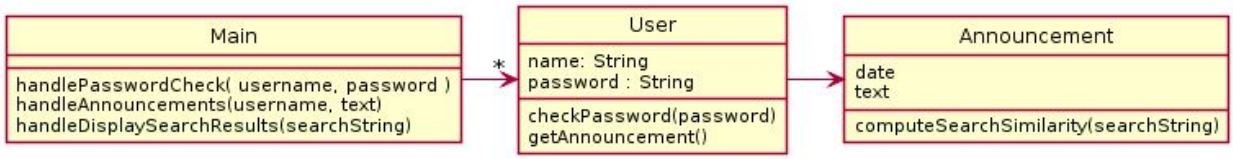
Your Solution: Problems With B

Your Final Solution

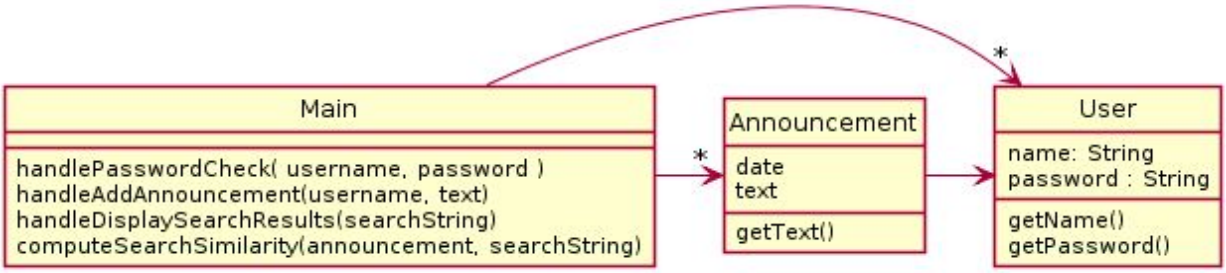
Announcements

On a particular website, users log on with a username and password. Once logged on, they can make announcements that have their username, the current date and some text associated with them. The website also has a feature where you can search for announcements. Given a search string and announcement, an algorithm can compute a similarity rank in the range of 0-100. The results are then sorted in similarity rank order.

Solution A



Solution B



Your Solution: Problems With A

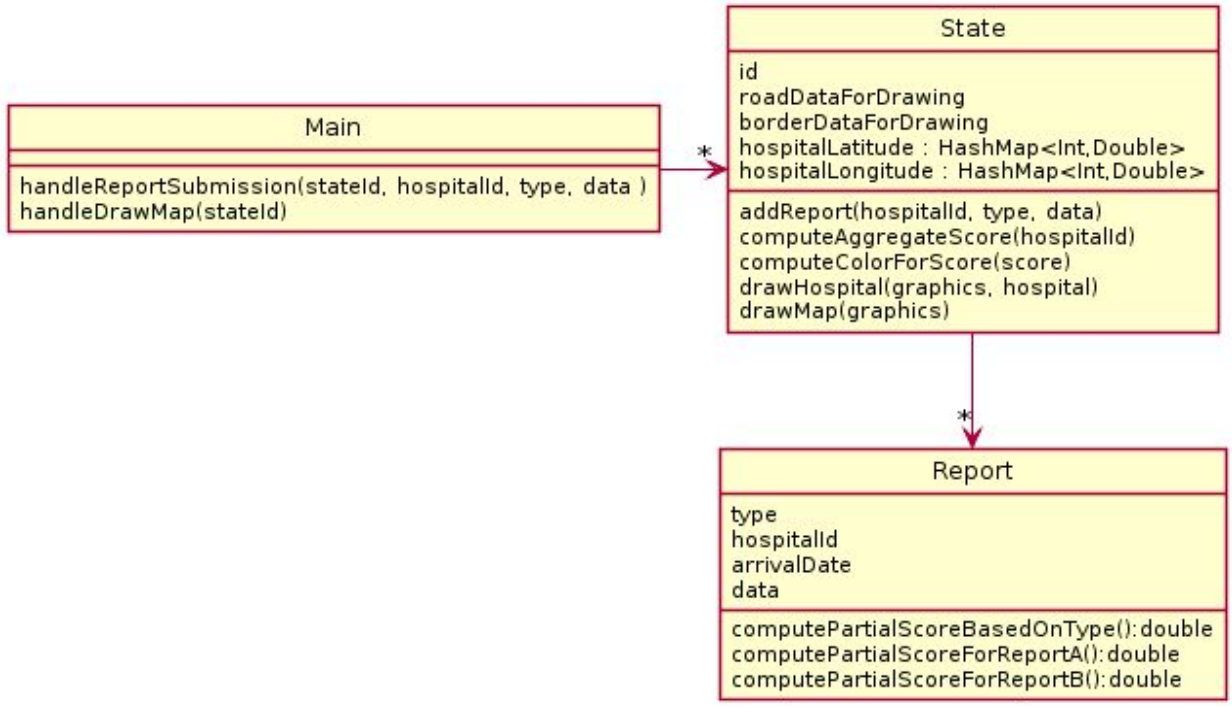
Your Solution: Problems With B

Your Final Solution

State Hospitals

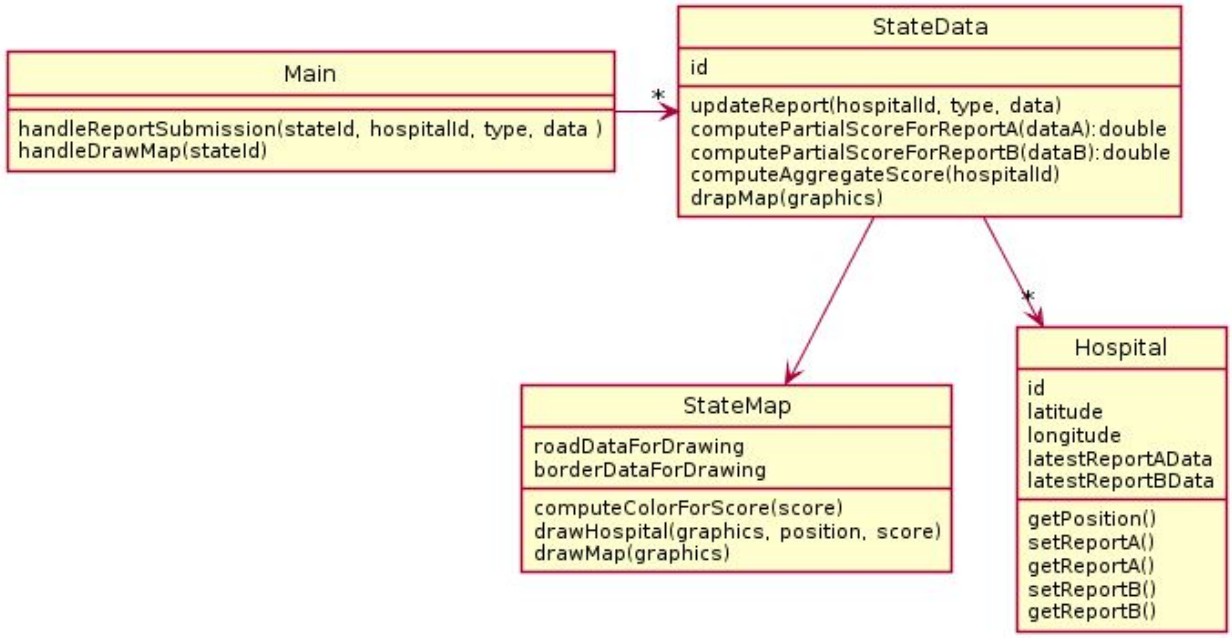
A government agency tracks hospital data to make a monthly report. The highlight of the report is a set of colorful maps that show every hospital in a particular state with colors indicating the hospital's overall status. To make this report, hospitals submit two different kinds of reports (reportA, reportB) that come in at different times. Each of these reports is analyzed differently and produces a partial score. The final map color is produced by combining these two scores into an aggregate score, using the most up-to-date version of each report available at that time. Color on the map is positioned according the the hospital's specific latitude and longitude which is stored in the system and does not change. The drawn map also includes the states borders and major roads.

Solution A



Your Solution: Problems with A

Solution B



Your Solution: Problems with B

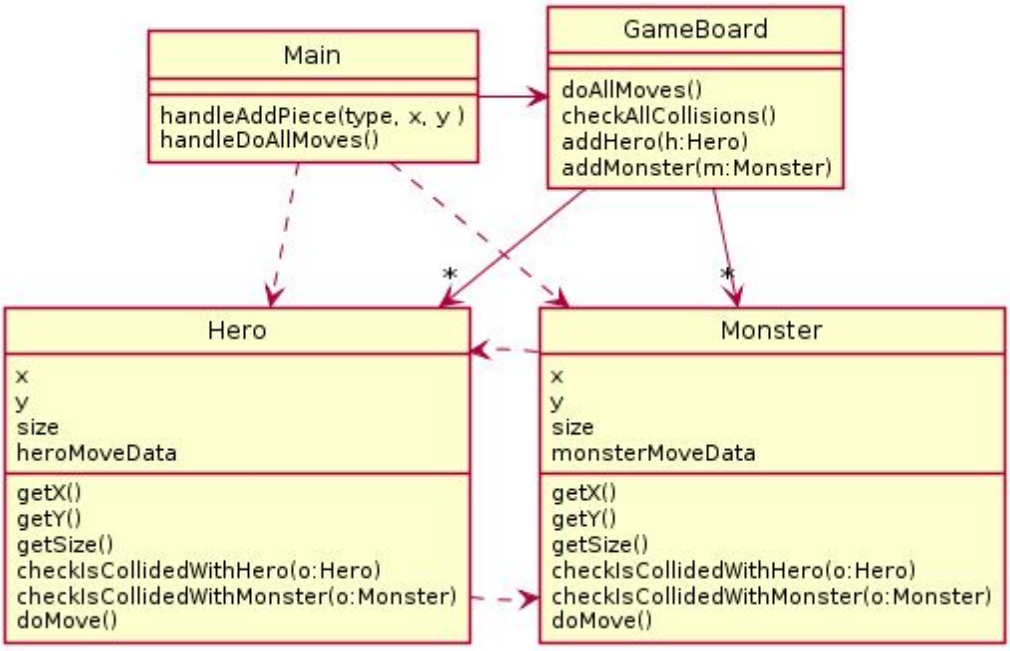
Your Final solution

VideoGame

<u>This problem uses interfaces & inheritance, so don't bother looking at if you haven't gotten there in the class yet.</u>

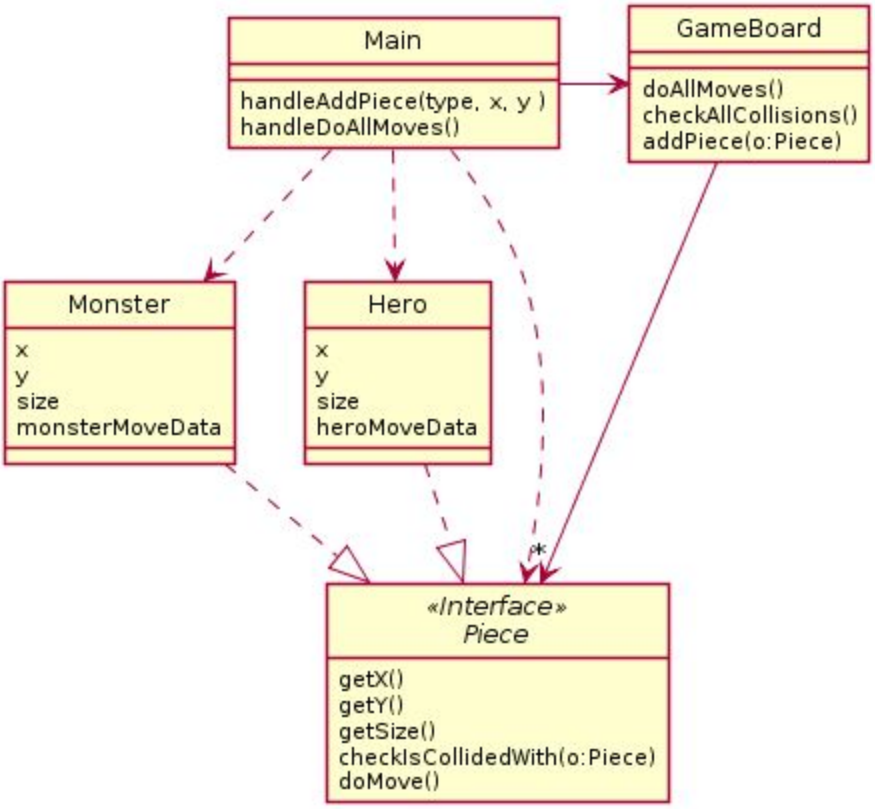
In a particular video game, there are pieces called heros and monsters. Heros and monsters move differently from each other. The move is complex determination which requires specific data that is updated over the monster and hero's lifetime. Heros and monsters store different kind of move data. It is also necessary to detect if one piece is colliding with another piece - this functionality is pretty much the same regardless of if you are talking about heroes or monsters. The game must support: giving each piece the opportunity to move and update their position, and adding a new piece (either hero) or monster to the board.

Solution A
For this one, I only want you to identify problems in A ***that are fixed by Solution B.***



Your Solution: Problems with A ***...that are fixed by B:***

Solution B



Your Solution: Problems with B

Final Solution