Alex Verdin

CMSI 402

Prof. Johnson

February 12, 2018

Homework 2

* 1. The difference between component-based architecture and service-oriented architecture is that rather than having one program that is designed to interact with other components of the program (component-based), service-oriented uses services which are their own independent programs which handle their specific task.
  2. Since the game is not sharing information online, only locally; it is best to use a component-based architecture since all the information will be contained in one program and not over a network.
  3. The chess game should use a service-oriented architecture since it is required that it has online functionality, and a service-oriented architecture is needed to use the services that will send information to an independent server, then back to the players.
  4. The ClassyDraw application only needs to use a simple hierarchical database structure, meaning that it really only needs to save information based on a drawing that is saved. Multiple drawings that may be saved as different files will not have any relationship to one another. Perhaps ClassyDraw creates a new save file if an old drawing is edited, and this new version is still related only to the previous drawing in case the user wants to undo changes. In terms of maintenance, the database will need to get rid of old files that have already been changed at some point, but the database should not get rid of them too early in case the user wants to undo their changes.
  5. Diagram:
  6. These classes share components such as some sort of draw function, a size function, a position/rotation function, and a color/transparency function. However, the Line, Rectangle, Ellipse, and Star classes may also share line style properties that the Text class may not have. The Rectangle, Ellipse, and Star classes may also share a fill style property. On its own, the Text class will have its own unique properties such as font, bolding, and italicization. Although all classes share some draw property and size property, they should be implemented differently for each class. The Line, Rectangle, Ellipse, and Star classes could implement the line style property the same for each. The Rectangle, Ellipse, and Star classes could implement the fill style property the same (given that a fill style function was written to detect lines and draw within them). All the classes could implement color/transparency functions and position/rotation functions the same as well.
  7. Diagram: 
  8. Diagram: 
  9. You can combine Salaried, Manager, and Vice President by removing the managers property from Vice President and just keeping a universal Employees property. The name of the class would just be Salaried Employees and it would have boss, employees, office, and salary properties. It may also be useful to add an employee position property (position) to this class so individual positions can still be distinguished. The updated diagram would look like: 