Sebastian Cruz

CMSI 4072 Assignment #2

5.1

* Component-based architecture regards the pieces of the system as a collection of loosely coupled components that provide services for each other
* Service-oriented architecture is similar to component-based architecture except that the pieces of the system are implemented as self-contained programs (services) that run on its own, providing service for its clients
  + Often running on separate computers that communicate across a network
* Main difference is how separated the pieces are in the architecture (service-oriented much more separated)

5.2

* Monolithic architecture would work well because the application is fairly simple and is meant to be self-contained (since there’s no external database for scores to be saved on)
  + Would work best because there are features in the application that can work with other types of architectures
    - Data-centric/rule-based for building the table of moves and choosing the best response given a game state
    - Event-driven for the UI, to respond to user events
    - Distributed for the AI to explore different movesets at the same time

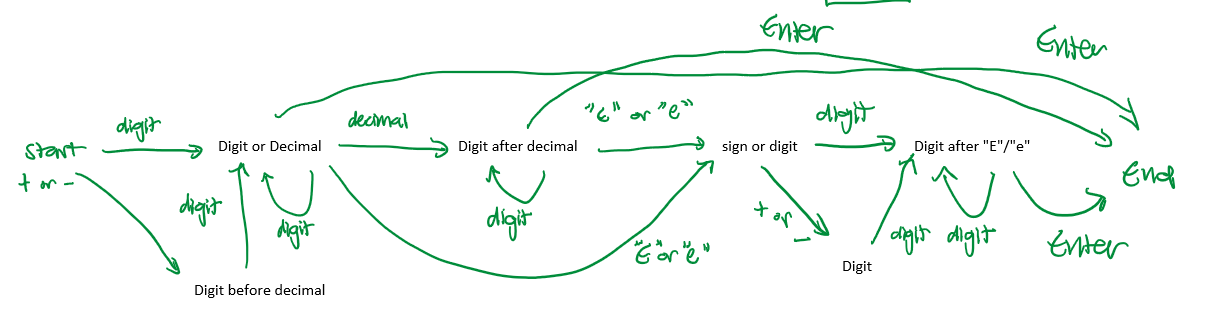
5.4

* With the addition of multiplayer over the Internet, the changes to a Chess game vs. a bot would be to allow the program to exchange data across the Internet with another instance of the application and remove the computer bot
* Because there is less focus on the AI, the application would become a monolithic service-oriented application

5.6

* A database isn’t really needed because drawings can be stored on separate files
  + OS’s has functions that allow the user to manage files; letting them delete old files, make backups, etc.
* For maintenance, the application should create a temporary file while the user is drawing so that the file can be restored in the case that the program crashes or is accidentally closed

5.8

* 

6.1

* All share
  + Color
  + Width
  + Height
  + Position
* Do not share
  + Only exclusive to Text
    - Font
    - String
  + Only exclusive to Star
    - Number of points
* Shared by some
  + Line, Rectangle, Ellipse, Star
    - Line style
    - Line thickness
* Shared properties should be stores in an overarching class while the non-shared properties should be stored in individual class

6.2

* 