

Alec Harb and Beryn Staub-Waldenberg

March 12, 2020

Assignment 3 - Cohesion and Coupling

CSCI345

Going from the text-based version to the GUI-based version we of course added a “view” and our program now follows the model-view-controller pattern. Additionally, we moved all the player-based turns into the player class, rather than having them in our controller class. This created a more cohesive player class because all of the actions pertaining to a player were in the same class. There are a few places in our code, however, where our view class is coupled with the model. For example, after the act button is pressed, the view interacts with the player class to call its act() method. Conversely, the act() method in the player class will call a method in the view class in order to display the outcome of acting. Another example is when move is pressed, the player’s location is updated in the view rather than the model. There are definitely a few parts of our program that could be simplified and made a little more cohesive and a little less coupled. Given more time, we could improve those issues.

The controller (Deadwood) class does seem to have functional cohesion since it only contains methods and fields that pertain to the succession of the overall game, with the exception of the upgrade() method which arguably could be put in the Player or Office class but it’s a bit of a tossup. The view (Board) class however, only has logical cohesion since it does a few things that pertain to the model such as updating certain player information. The individual object classes (ie. Shot, Set, Role, etc.) all seem to have functional cohesion.