Steven Le and Colton Hagan

February 23rd, 2021

CSCI 345

Cohesion and Coupling Report

Our program follows the MVC model which means there is high cohesion and low coupling. We have 8 java files to store data and to act as Models: Board, CastingOffice, Gamestate, PlayerModel, Role, Room, Scenecard, and Scenes. We have 4 java files that take user input and XML files to modify this data and act as Controllers: DeadwoodController, GameStateController, Systems, and parseData. We have one java file that communicates with controller to display data for the user and act as a View: DeadwoodView.

By employing this strategy, we ensure low coupling. All eight models will only rely on either DeadwoodController or GameStateController to manipulate their data, which means the coupling degree is low. Thus, if there are any errors for one Model it will either be in that model, models which it will use, or it's controller. We can also easily modify models as we see fit since we only need to change its associated controller to fix any errors. Model's that contain instances of other models won't be affected much since they only store data instead of manipulating it. Also, we can easily reuse models as needed. The view will only receive instructions to display to user from the same two Controllers.

This strategy also ensures high cohesion. Each model will only hold data for its respective task, which is overall small for each of these models. For instance, Role only needs to know the attributes for the given role: rank, usedBy, isExtra, name, and tagline. It is a similar case for the Controllers: DeadwoodController is responsible for player actions like acting, rehearsing, moving, upgrading, and taking a role. GameStateController on the other hand

controls setting up the game, playing the game, bonus payments on scene end, and day/scenecard/game endings. In other words, code that handles similar functionality and is used together will be kept in the same java file. Changing aspects of one of these Models or Controllers would not require changes in other files.