Vlad Bugayev
Natthapong Choeypant
CSCI 345
8/1/19

Deadwood Report

In our design of deadwood, we had to use multiple design patterns to create a playable, GUI based version. The main and obvious design patterns are MVC (Model – View – Controller) as well as the observer pattern. MVC was used to separate and coordinate the communications between the model (which was completed for assignment 2) and the view, which we had to create for this assignment. The interactions between these two systems was accomplished using a controller, which had to have references to both the model and the view.

The observer pattern was used to update the GUI according to changes that the controller made in the model via user interactions. Without the observer pattern, polling loops would have to be used to accomplish the same task which would be a waste of resources. We only had to use 2 observers, one for the main GUI and one for the player box upon game startup. The interaction between the MVC and observer patterns was accomplish using action listeners as well as item listeners. This was the hardest part of the assignment.

The other design pattern we used was the singleton in the design of the die rolling system, as there is no need for multiple dice – we can simply reuse the same object multiple times for multiple scenarios.

Given more time, we would have liked to use the builder pattern in creating all of the cards, instead of hard coding parameters which can be time consuming and also confusing. We did not know we had access to an .xml, which could have made things a lot easier.

Because we only had two weeks to complete the entire game, we had to work quickly and speed up our design process, therefore the coupling between modules is very high, although the classes themselves have a cohesive set of responsibilities. We learned a lot through the development of this game about object-oriented programming, and this skillset will surely be useful as we continue our programming careers.