Keegan Mangahas

CSCI 345

Design Patterns in Deadwood

MVC

* GUI show information (view) , Deadwood.java runs methods to do what the player wants (controller), the Player, Set, Board, etc. classes get their data updated (model)

Abstract class

* Room class – Set/Trailer/Office
* The Set class is used to create all the different sets on the board, but can also be manipulated to create the unique Trailer and Office ‘sets’

Cohesion

* Communicational
  + The Board class interacts with the Set class, which interacts with the Scene class. The Set and Scene classes both contain Role classes. Where the communication starts (Ex. Board -> Set, Scene -> Role) depends on what needs to be updated
* Sequential
  + Because of the nature of an action-based program, events in the Deadwood class happen in certain orders depending on what is happening in the game

Coupling

* Content
  + The Gamemaster class modifies data in the Player, Board, Set, etc. classes and relies on them in order for the game to run
* External
  + Data needed to create the Set, Role, and Scene classes are gathered from an external XML file
* Message
  + When updating information or when a player does something, the game will send a message to the player to make sure they know what is happening

Singleton

* The XMLParser class is only needed once in the beginning of the program to collect its data and create other classes.
* The Player creation method is only used once in the beginning to create X players

Adapter

* An adapter is used to help make the Set class usable for the Trailer and Office ‘sets’

Observer

* Updating a scene card changes displayed information (wrapped, location, money, new day, etc)