

The design of my program follows the model view controller idea. Where the the "Game" class responsible to display the entire game, that is the view of the program and will update the view as the controller requires. The "GameController" is the controller which translate user actions into operations on the model, and the other classes such as "Room" "RoomPart" "Player" those are the model where the data is. For example, if the user wanted to upgrade, the payer will click the level their wanted to upgrade and the "Game" class will pass it to "GameController" then "GamController" do what it needs, such as getting the cost and setting the cost and level to "Level" "player" which are model, then once it is updated then pass back the "GameController" ,and "GameController" will pass it back to "Game".

For coupling, I use data coupling for getting the xml data, XmlBoardReader class , XmlCardReader class and the classes that are called the reader which is the broad class, and saving in room class card class part class...etc. Also External Coupling i use java fx.

For Cohesion , I used Procedural Cohesion for the level up function. The functions include LevelUP,dollarLevelCalculation, creditLevelCalculation,afterLevelUp,failUpgrade, they are all working for Procedural Cohesion. All it is doing is allow the player to choose what kind of level up , then calculate the cost and apply the cost. The last is showing that if it is successful the upgrade, and showing failure , if the player doesn't have enough money. I also use Communicational Cohesion for all the display functions, showAllPlayerDice ,DisplayAllPlayer, getPlayerInfoFormats, those are all Functions performed on the same data which is based on the player data. So that it is easy to keep checking what is needed to fix or implement

Behavioral Patterns, I used a lot between the game class where getting the rooms, there are alot of getters and setters and showing all the near rooms as an option , showing all the oncardjob options, setting the offcardjob option..ect. I think it is a good idea to use it since it is more affection to have it processed that way i wanted.