Develop Element

Within the game.start() method, once the player selects (within the postMoveOptions method), the Develop Element option, the developmentMenu() method is called within Game. This method calls the returnDevelopableElements(player) method with an argument of the current player. This method creates a List<Element> developableElements, and loops through each ElementSystem object, checking each element comprising the elementSystem to confirm if the player owns all Elements within the ElementSystem, adding said Elements to developableElements if so. The returned developableElements list is then checked via the noDevelopmentsToMakeChecker(player, developableElements) method. If developableElements.size==0, the developmentMenu is exited. If developableElements.size>0, each element's current development level is checked via the element.getDevLevel method, removing it from developableElement if devLevel==4. A loop then offers the player options to develop each of the final developableElements list. This loop will persist until the player chooses to exit, calling the developElement(player, element) method on valid user input for the selected element. The method checks if the player can afford the development, calling player.removeResources(element.priceToDevelop) and element.increaseDevLevel() if the player has sufficient resources, or displaying a message indicating a failed upgrade if not. Note that I have excluded the userInput class from the diagram to provide a less cluttered visualisation of the flow.

Random Event

Within Game, at the end of each loop of player turns, the generateRandomEvent(players) method of RandomEvents is called. The method generates a sequence of random numbers, playerNum, goodOrBad, goodEventNum, badEventNum, and fineNum. if goodOrBad==0, this denotes a 'bad' random event, and the method prints badList.get(badEventNum) and a dialogue indicating player fine of fees.get(fineNum). Resources are then removed from a random player via players.get(playerNum).removeResources(fees.get(fineNum)). Alternatively, if goodOrBad==1, goodList.get(goodEventNum) will be displayed along with a dialogue indicating player reward of fees.get(fineNum). Resources are then added to a random player's resources via players.get(playerNum).addResources(fees.get(fineNum)).