Seungmin Lee

Professor Sciore

Object Oriented Design

18 September 2015

Assignment 2

My code is largely divided into four sections: *Player, Dice, Pot* and *Game*. I designed it so that the *Game* handles the dice object, the pot object and a list of players. A player will usually not have a direct access to the dice nor the pot object. Only when it is their turn, they are given access to the two objects (through *playOneTurn* function and *collectThePot* function). For the *Dice* class, I did not make it to be 1 die per object, but instead, added a constructor called *numDice* that keeps track of how many dices one is rolling. I found it unnecessary to create a new Dice object every time a user is rolling another dice because the number of dices to roll are constantly going up and down. So the dice class handles arbitrary and dynamic number of dices that are used in the game. Under the abstract class *Player*, there are User player (the human) and Computer player. To create various strategy AI’s for the computer, I made the ComputerPlayer class an abstract class with no content inside, and made three different subclasses for different difficulties. I did this purely for organization issue as I wanted to keep different types of computers grouped up as a subclass.

The *Game* class largely handles the flow of the game from creating User and Computer object and setting the difficulty, to managing turns, dice and deciding the winner. All the players were kept in an arraylist called *players*. ComputerPlayer object were created into the game by a for loop and an if statement to check for what difficulty the user chose, and set created computers of that difficulty. This does, however, limit the user from creating various computers with different difficulties; but, as this was not part of the task I had to implement, I left it as is.

I have a central command processing function called “processCommand” in the Game class which takes in the output of the abstract function “MakeDecision()”of each player. For the human player this function will be a jOptionPane output, and for the computers, this function will take in the current pot value and dice counts and output a decision according to its difficulty. If command is 0, it will run “processOneTurn” which will make the player roll the dices. If command is 1, it will run “endTurn” which will make the player take whatever is in the pot.