**Object Oriented Programming Assessment 2**

**Die object**

Text

Description automatically generatedThis is the simplest of the objects as all it really needed was a method to “roll” the die, however I did add a way to create dice with more or less than 6 sides also.

Text

Description automatically generated

**Player object**

Text

Description automatically generatedThe player class handles a lot of the functionality, it has two attributes: name and score

The main method is the Turn() method, which rolls the die 5 times and then adds those rolls to a list of all the rolls that turn, it then calls the printRolls() method and the calculateScore() method.

Text

Description automatically generated

This is the printRolls() method which is private due to only being used within its own object and takes the list of rolls and iterates through it adding them to a string which is printed to the console.

Text

Description automatically generatedText

Description automatically generatedThis is the calculateScore() method which is private due to only being used within the object itself, it first finds out the modal roll and also how many times it appears in the list. It then uses this to give the player a score based on how many of a kind they got using a switch case statement. If the player gets a two of a kind, they are given the opportunity to take 1 point or to reroll the remaining dice to try and get a better score, however if they do not get any rolls the same, they are awarded no points.

Text

Description automatically generated

This is the Reroll() method which is called when the player gets a two of a kind and chooses to reroll the remaining dice; it is private as it only needs to be called from within the object. Firstly, it creates a new list to store the new rolls and then it rolls twice. If neither of the two new rolls are equal to the mode then the two previous rolls are added to the front of the list and the player is awarded no points, however if at least one of the new rolls is equal to the mode, then it will roll once more and insert the previous rolls to the beginning of the list and calculate the score once more with the new set of rolls.

Text

Description automatically generated**Game object**

This is an abstract object used as a parent for the two game types (two and three players) which initialises a few attributes here: the private goal (only used in this object and not to be changed by child objects), protected n\_of\_players and playing (they don’t need to be accessed outside the object but to need to be changed by child objects), and the public player\_list and the die.

Text

Description automatically generatedText

Description automatically generatedThis object contains three methods with functionality. addPlayers() is called at the start of the game loop and it adds an amount of players to player\_list based on the n\_of\_players while asking the user for the names of said players. checkWin() is used to check if a player has won at the end of every turn and printScores() is called after every turn to print all players’ current scores to the console. They are all public in this screenshot but I changed them to protected as they do not need to be accessed from outside the object but are used by child objects. There is also an empty Play() method which will be changed in child objects.

Text

Description automatically generated**TwoPlayerGame and ThreePlayerGame objects**

Text

Description automatically generatedThese are both child objects of the Game object, they polymorph the original method to add individual functionality for the Play() method. The Play() method here initialises some attributes and then adds players to the player\_list, after this it starts a game loop. The game loop uses a switch case statement to call turn on the player who’s turn it is and then switch the turn number to the next player. It also prints the scores and checks for a win after each turn. The ThreePlayerGame is similar but with more turns.

**Program object and main method**

The main method contains all the options for the player, for example whether they want to play a game, how many sides the dice will have and how many players they want to play with, all of which have error handling in the case of erroneous input.

**Testing Video**

<https://youtu.be/pssytkItGwQ>

**Checklist**

Table

Description automatically generatedTable

Description automatically generatedTable

Description automatically generatedTable

Description automatically generated