Brett Plemons

IT 312 Software Development with C++.NET

5-1 Final Project Building Block Two: Pseudocode (Building the Final Project)

30 September 2023

**Liar’s Dice Game**

**IMPORT** Game

**MAIN** **FUNCTION**

// Show welcome message

**DISPLAY** “Welcome to Liar’s Dice!”

// Initialize the Game (which will read and display rules, set up players, and start the game loop)

**SET** game **TO** **NEW** Game()

// Goodbye Message

**DISPLAY** “Thank you for playing Liar’s Dice!”

**Player Class**

**CLASS** Player

**PROPERTIES:**

id **AS** **INTEGER**

dice **AS LIST TYPE** Dice

CONSTRUCTOR(id)

**SET** this.id **TO** id

**CALL** this.ROLL\_DICE()

**METHODS:**

**FUNCTION** ROLL\_DICE()

**SET** this.dice **TO** **EMPTY** **LIST**

**FOR** i **FROM** 1 **TO** 5

**SET** newDie **TO** **NEW** Dice() // Create a new Dice object

**APPEND** newDie **TO** this.dice

**ENDFOR**

**FUNCTION** DISPLAY\_DICE()

**SET** diceValues **TO** **EMPTY** **LIST**

**FOR** **EACH** die **IN** this.dice

**APPEND** die.GET\_FACE\_VALUE() **TO** diceValues

**ENDFOR**

**DISPLAY** “Player “ + this.id + “, your dice are: “ + diceValues

**FUNCTION** MAKE\_GUESS()

**DISPLAY** "Enter your guess in format (quantity, face\_value): "

**READ** new\_guess **FROM** **USER**

**RETURN** new\_guess

**FUNCTION** CALL\_LIAR()

**DISPLAY** "Do you want to call liar? (yes/no)"

**READ** call\_liar **FROM** **USER**

**RETURN** call\_liar

**END CLASS**

**Dice Class**

**CLASS** Dice

**PROPERTIES:**

face\_value **AS** **INTEGER**

CONSTRUCTOR()

**CALL** this.ROLL()

**METHODS**:

**FUNCTION** ROLL()

**SET** this.face\_value **TO** random\_number\_between(1, 6)

**FUNCTION** GET\_FACE\_VALUE()

**RETURN** this.face\_value

**END** **CLASS**

**Game Class**

**CLASS** Game

**PROPERTIES:**

players **AS** **LIST TYPE** Player

currentPlayerIndex **AS** **INTEGER**

lastGuess **AS** **TUPLE TYPE(INTEGER, INTEGER)** // (quantity, face\_value)

rulesText **AS STRING**

CONSTRUCTOR()

**SET** this.players **TO EMPTY LIST**

**SET** this.currentPlayerIndex **TO** 0

**SET** this.lastGuess **TO** (0, 0)

**SET** this.rulesText **TO** CALL READ\_RULES\_FROM\_FILE("rules.txt")

**DISPLAY** this.rulesText

**CALL** this.SETUP\_PLAYERS()

**CALL** this.PLAY\_GAME()

**METHODS:**

**FUNCTION** READ\_RULES\_FROM\_FILE(filename)

**SET** rulesContent **TO** **EMPTY\_STRING**

file\_handle = **OPEN\_FILE(**filename, ‘read’**)**

**IF** file\_handle **IS** **NONE**

**DISPLAY** “Error: Could not open rules file.”

**RETURN**

**ENDIF**

**WHILE NOT END\_OF\_FILE(**file\_handle**)**

**SET** line **TO READ\_LINE\_FROM\_FILE(**file\_handle**)**

**APPEND** rulesContent **WITH** line + **NEWLINE**

**ENDWHILE**

**CLOSE\_FILE(**file\_handle**)**

**RETURN** rulesContent

**FUNCTION** SETUP\_PLAYERS()

**DISPLAY** "Enter the number of players: "

**READ** num\_players **FROM** **USER**

**WHILE** num\_players < 2

**DISPLAY** "Please enter a number greater than 1: "

**READ** num\_players **FROM USER**

**ENDWHILE**

**FOR** i **FROM** 1 **TO** num\_players

**APPEND** **NEW** Player(i) **TO** this.players

**ENDFOR**

**FUNCTION** PLAY\_GAME()

**WHILE** true

**SET** currentPlayer **TO** this.players[this.currentPlayerIndex]

**CALL** currentPlayer.DISPLAY\_DICE()

**SET** guess **TO** currentPlayer.MAKE\_GUESS()

**IF** **CALL** VALIDATE\_GUESS(guess, this.lastGuess) EQUALS FALSE

**DISPLAY** “Invalid guess. Try again.”

**CONTINUE**

**ENDIF**

**SET** this.lastGuess **TO** guess

**SET** call\_liar **TO** currentPlayer.CALL\_LIAR()

**IF** call\_liar **EQUALS** “yes”

**SET** winner **TO** CHECK\_GUESS\_AGAINST\_DICE(this.lastGuess, this.players)

**DISPLAY** "The winner is Player " + winner

**BREAK**

**ENDIF**

**INCREMENT** this.currentPlayerIndex **BY** 1

**IF** this.currentPlayerIndex >= **LENGTH** **OF** this.players

**SET** this.currentPlayerIndex **TO** 0

**ENDIF**

**ENDWHILE**

**FUNCTION** VALIDATE\_GUESS(new\_guess, last\_guess)

**SET** isValid **TO** **FALSE**  
 **IF** last\_guess[0] EQUALS 0 AND last\_guess[1] EQUALS 0

**SET** isValid **TO** **TRUE**

**RETURN** isValid

**ENDIF**

**IF** new\_guess[0] > last\_guess[0] **OR**

(new\_guess[0] EQUALS last\_guess[0] **AND** new\_guess[1] > last\_guess[1]) **THEN**

**SET** isValid **TO** **TRUE**

**ENDIF**

**RETURN** isValid

**FUNCTION** CHECK\_GUESS\_AGAINST\_DICE(last\_guess, players)

**SET** counter **TO** 0

**FOR** **EACH** player **IN** players

**FOR** **EACH** die **IN** player.dice

**IF** die.GET\_FACE\_VALUE() **EQUALS** last\_guess[1]

**INCREMENT** counter **BY** 1

**ENDIF**

**ENDFOR**

**ENDFOR**

**IF** counter >= last\_guess[0]

**RETURN** “Guessing Player”

**ELSE**

**RETURN** “Calling Player”

**ENDIF**

**END CLASS**