20th March, 2024

Add more to the pseudo code. Moved the input command calculation to its own function for readability. Began looking at how score for a player should be kept before cashing points in case of a farkle.

**Global**

INITIALIZE Players array

INITIALIZE current turn = -1 *//position in array of players, -1 means game just started*

INITIALIZE current roll array || size of 6

**Players Object**

INITIALIZE player name

INITIALIZE player score = 0

INITIALIZE winner FALSE

**Constructor**

SET player name

**On Scoreboard Function**

IF score >= 500

RETURN TRUE

**Add Score Function**

Player score += points

IF player score >= 10000

SET winner TRUE

**Loop function:**

LOOP while input is not close

PRINT What is your command?

PRINT commands (roll / roll again, drop out, quit application, score dice)

READ input command

CALCULATE FUNC input command

**Calculate Input Function:**

CALCULATE input command

IF quit application

EXIT switch

ELSE IF roll / roll again

IF has score

Roll again

ELSE

Roll initial

ELSE IF score dice

IF can score dice

Add dice score to player score

ELSE IF drop out

PRINT Player quits the game

REMOVE player from game list

EXIT switch

ELSE IF quit application

PRINT application closing…

EXIT switch

END IF

**Roll Dice Initial Function**

INITIALIZE gotPoints to FALSE

PRINT Roll Dice

WAIT for keypress

FOR 6 dice

CALCULATE dice rolls

SET dice roll in array

IF roll is point AND got points is FALSE

SET got points TRUE

IF got points FALSE

RETURN farkle *//probably will be an empty array?*

ELSE

RETURN points as array *//having each point be in an array will help to determine how many dice contain points and how many can be rerolled.*

22nd March, 2024

Began implementing some stub code to get a further idea of how to structure the code and to see if some parts of the pseudo are feasible or not - IE can not use a switch statement with a string as the parameter to test against, had to use a if-else chain instead, *sad C# syntax noises ensue*. Beginning to implement some of the code has helped me to realize more input commands that will be needed, such as a command to view the scoreboard. Need to look into if a player can join the game part way through - thinking not but may be a possibility.

Having to declare functions and define them separately keeps throwing me off, or ordering them in a specific way so that every other function that uses said function is below said function… Declaration is easier to do but once again C# syntax and rules keep throwing me off.



| /\* Name : Nathan Hallam Course : IT-132 Date : --- Project NO : --- File Name : FarkleMain.cpp Description: --- \*/  #**include** <cstdlib> #**include** <iostream> #**include** <iomanip> **using** **namespace** std;  **int** currentTurn = -1; //What player currently has control **int** currentRoll[6] = { 0 }; //The scoring sides of the dice roll **int** tempScore = 0; //Used for when the current player rolls more than 6 dice //Function Declarations **void** **PrintCommands**();   **class** **Player** {  string \_name; //Alias for the player  **int** \_score = 0; //The amount of points the player has cashed  **bool** canWin = false; //If the player has enough points to win   //Constructor **public**:  Player(string name) {  **this**->\_name = name;  }   //Returns true if the player can be placed on the scoreboard,  // otherwise false  **bool** **OnScoreboard**() {  **if** (\_score >= 500) {  **return** true;  }   **return** false;  }   //Adds points to the scoreboard given the sum of points  **void** **AddScore**(**int** points) {  \_score += points;   **if** (\_score >= 10000) {  canWin = true;  }  } };  **void** **CalculateInput**(string command) {  //Quit program  **if** (command == "Q" || command == "q") {  **return**;  }  //Roll  **else** **if** (command == "R" || command == "r") {  **if** (tempScore == 0) {  //Roll Initial  cout << "STUB - Roll Dice Initial" << endl;  **return**;  }   //Roll Again  cout << "STUB - Roll Dice Again" << endl;  }  //Cash dice  **else** **if** (command == "C" || command == "c") {  cout << "STUB - Cash Dice" << endl;  //Loop through current roll and add score dice to temp score  //Add temp score to current players score  }  //View Scoreboard  **else** **if** (command == "S" || command == "s") {  cout << "STUB - Scoreboard" << endl;  }  //Drop out  **else** **if** (command == "D" || command == "d") {  cout << "STUB - Drop Out" << endl;  }  //Default  **else** {  cout << "Command not recognized..." << endl << endl;  } }  **void** **PrintCommands**() {  cout << "STUB - Print Commands" << endl; }  **void** **InitializeGame**() {  **int** numOfPlayers = -1;  **bool** set = false;   **do**  {  **if** (numOfPlayers < 2 && set) {  cout << "Farkle requires at least 2 players to start..." << endl;  }   cout << "How many players will be playing today? : ";  cin >> numOfPlayers;  set = true;  } **while** (numOfPlayers < 2); }  **void** **main**() {  string input = "";   InitializeGame();   **do** {  PrintCommands();   cout << "What is your command? ";  cin >> input;   //STUB - If current player is a winner, end game   CalculateInput(input);   } **while** (input != "Q"); } |
| --- |