April 1st, 2024

Rewrote the RollDice function as suggested by the professor.

| **void** **RollDice**(**int** diceRoll[6], **int** diceToRoll) {  **for** (**int** i = diceToRoll - 1; i >= 0; i--)  {  diceRoll[i] = rand() % 6 + 1;  } } |
| --- |

Added functionality to the PrintScoreboard function and implemented a comparer function:

| //Prints the scoreboard to the console **void** **PrintScoreboard**() {  cout << "Scoreboard:" << endl;   //Create temp array to sort players by score while keeping play order  Player\* temp = **new** Player[numOfPlayers];   //Copy each object to the new array without copying address  **for** (**int** i = 0; i < numOfPlayers; i++)  {  temp[i] = players[i];  }   sort(temp, temp + numOfPlayers, ScoreComparer);   //for each player  **for** (**int** i = 0; i < numOfPlayers; i++)  {  //If they are on the board, print position and score  **if** (temp[i].OnScoreboard())  cout << "[" << i + 1 << "] : " << temp[i].GetName() << " : " << temp[i].GetScore() << endl;  //Otherwise print name without position  **else**  cout << "[-] : " << temp[i].GetName() << endl;  }   **delete**[] temp; //Clear memory } |
| --- |

| //Used by the scoreboard to determine who has the higher score **bool** **ScoreComparer**(Player playerOne, Player playerTwo) {  **return** playerOne.GetScore() > playerTwo.GetScore(); } |
| --- |

April 3rd, 2024

Moved the player class to a separate header file. Moved function declarations to a separate header file and put definitions on a new FarkleCore.cpp file. FarkleMain.cpp contains the main() function to run the file.

Added print to the PrintCommands() function so it is no longer a stub function.

April 6th, 2024

Added functionality to the Dropout() function so it is no longer a stub function.

| //Removes the current player from the array of players and starts the next turn **void** FarkleCore::Dropout() **const** {  //Decrement number of players  numOfPlayers--;   **int** p = 0; //placeholder  Player\* temp = **new** Player[numOfPlayers]; //Temp dynamic array   //loop through the array and copy the elements  **for** (std::**size\_t** i = 0; i < numOfPlayers; i++) {  //If on the player that wants to drop out, increment placeholder  **if** (i == currentTurn) p++;   temp[i] = players[p];  p++;  }    **delete**[] players; //free memory  players = temp;   //If under the minimum number of players to play, end early  **if** (numOfPlayers < 2) **return**;   NextTurn(); } |
| --- |