CS 162 Assignment #1 Go Bowling

Design Document due: Sunday, 4/3/2022, 11:59 p.m. (Canvas)

Assignment due: Sunday, 4/10/2022, 11:59 p.m. (TEACH)

Demo due: 4/22/2022 (without penalties)

Goals:

- Practice good software engineering design principles:
  - Design your solution before writing the program
  - Develop test cases before writing the program
- Review conditionals, loops, functions, and array
- Practice dynamic memory usage with pointers
- Use functions to modularize code to increase readability and reduce repeated code

---

Problem Statement:

You have just been tasked with writing the software for your local bowling alley. You must simulate bowling with random numbers and make sure your software can keep score for different players in a game.

Bowling Rules:

- The game can have 1-5 players.
- There are 10 total frames in a game.
- Alternate players after each frame.
- For each frame, a player is given up to 2 rolls (chances) to knock down 10 total pins.
- With the first roll of any frame, a player can knock down 0-10 pins.
- With the second roll, a player can knock down 0 pins up to the number of pins remaining from the first roll.
- If the player knocks down all 10 pins in their first roll of a frame, then that is called a **strike(X),** and the player doesn't get a second roll/chance to knock down any remaining pins in that frame.
- If a player knocks down all 10 pins in a frame using two rolls, then it is called a **spare (/).**
- If a player knocks down 0 pins, this is called a **gutter (-)** ball.
- If a strike or spare is received in the 10^{th} frame, then the player is given a 3^{rd} roll/chance in that frame.
- The player who has the most points at the end of 10 frames is the winner.

Scoring rules:

- The score for a frame is determined by the total number of pins knocked down
  - A frame without a strike or spare is called an open frame. An open frame score is the total pins knocked down with two rolls in the current frame.
  - A frame score with a strike is 10 plus the number of pins knocked down with the next two rolls. This can give a player 10-30 points for a frame with a strike.
  - A frame score with a spare is 10 plus the number of pins knocked down with the next roll. This can give a player 10-20 points for a frame with a spare.
  - Three consecutive strikes in frame 10 gives 30 points.
- A player's total score is the sum of all 10 frame scores.

If you are not familiar with bowling, then you can read these instructions on how to play the game and keep score: http://www.fryes4fun.com/Bowling/scoring.htm

You can also try yourself here: https://www.bowlinggenius.com/

Implementation Requirements:

- The number pins knocked down in each roll is determined by a random number.
  - First roll: 0 to 10
  - Second roll: 0 to (10 – first roll)
- For each player, you must keep an array of pins knocked down with each roll for each frame, and display this information after each roll.
- You must keep an array of total scores for all players that is updated and displayed after each bowl or frame.
- After each roll, you must print the game scoresheet.
  - The game scoresheet consists of frame information and total score for all players.
  - The frame information consists of the number of pins knocked down with each roll in each frame and the sum of current and prior frame scores for each frame
  - Denote the strike with an X.
  - Denote the gutter ball with a dash, -.
  - Denote the spare with a forward slash, /.
- After each frame, you must print the total frame scores and a total score, if they can be calculated at that time.
- The game must print the winner(s) at the end of a game
- The game allows users to play again with a different number of players at the end of a game.
- Your program must handle all errors, such as not entering the correct input for number of players or option for bowling again.

Example Output:

```
How many players (1-5)? 2

Frame 1...
Player 1, press enter to roll.
Awe, you got a gutter ball, 0 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | -   |     |     |     |     |     |     |     |     |     | 0
            |     |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     |     |     |     |     |     |     |     |     |     |     |
            |     |     |     |     |     |     |     |     |     |     |
```

```
Player 1, press enter to roll.
You knocked down 7 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 |     |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     |     |     |     |     |     |     |     |     |     |     |
            |     |     |     |     |     |     |     |     |     |     |

Player 2, press enter to roll.
You knocked down 5 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 |     |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5   |     |     |     |     |     |     |     |     |     | 5
            |     |     |     |     |     |     |     |     |     |     |

Player 2, press enter to roll.
You knocked down 4 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 |     |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 |     |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |

Frame 2 ...
Player 1, press enter to roll.
You got a strike, 10 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 |     |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |


Player 2, press enter to roll.
You knocked down 5 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 | 5   |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |
```

```
Player 2, press enter to roll.
You knocked down 5 pins, spare.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   |     |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 | 5 / |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |


Frame 3...
Player 1, press enter to roll.
You knocked down 6 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   | 6   |     |     |     |     |     |     |     | 7
            |   7 |     |     |     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 | 5 / |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |

Player 1, press enter to roll.
You knocked down 2 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   | 6 2 |     |     |     |     |     |     |     | 33
            |   7 |   25|   33|     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 | 5 / |     |     |     |     |     |     |     |     | 9
            |   9 |     |     |     |     |     |     |     |     |     |

Player 2, press enter to roll.
You got a strike, 10 pins.

Name        |  1  |  2  |  3  |  4  |  5  |  6  |  7  |  8  |  9  |  10 |Total
-------------------------------------------------------------------------------
Player1     | - 7 | X   | 6 2 |     |     |     |     |     |     |     | 33
            |   7 |   25|   33|     |     |     |     |     |     |     |
-------------------------------------------------------------------------------
Player2     | 5 4 | 5 / | X   |     |     |     |     |     |     |     | 29
            |   9 |   29|     |     |     |     |     |     |     |     |
......
```

(10 pts) Extra Credit: Dynamic Arrays

Instead of supporting 1-5 players, modify the program so it can support N players using dynamic arrays allocated on the heap!!! You must not have memory leaks to get the full 10 points. Make sure you use valgrind!

Programming Style/Comments

In your implementation, make sure that you include a program header. Also ensure that you use proper indentation/spacing and include comments! Below is an example header to include. Make sure you review the style guidelines for this class , and begin trying to follow them, i.e. don't align everything on the left or put everything on one line!

```
/*********************************************************
** Program: bowling.cpp
** Author: Your Name
** Date: 03/30/2022
** Description:
** Input:
** Output:
*********************************************************/
```

Design Document – Due Sunday 4/3/2022, 11:59pm on Canvas
Refer to the Example Design Document – Example_Design_Doc.pdf

Understanding the Problem/Problem Analysis:

- What are the user inputs, program outputs, etc.?
- What assumptions are you making?
- What are all the tasks and subtasks in this problem?

Program Design:

- What does the overall big picture of each function look like? (Flowchart or pseudocode)
    - What variables do you need to create, when you read input from the user?
    - How to read from a file, and store information into your program?
    - How to write information from your program to a file?
    - What are the decisions that need to be made in this program?
    - What tasks are repeated?
    - How would you modularize the program, how many functions are you going to create, and what are they?
- What kind of bad input are you going to handle?

Based on your answers above, list the **specific steps or provide a flowchart** of what is needed to create. Be very explicit!!!

Program Testing:
Create a test plan with the test cases (bad, good, and edge cases). What do you hope to be the expected results?

- What are the good, bad, and edge cases for ALL input in the program? Make sure to provide enough of each and for all different inputs you get from the user.

Electronically submit your Design Doc (.pdf file!!!) by the design due date, on Canvas.

Program Code – Due Sunday, 4/10/2022, 11:59pm on TEACH

Additional Implementation Requirements:

- Your user interface must provide clear instructions for the user and information about the data being presented
- Use of array is required.
- Your program must catch all required errors and recover from them.
- You are not allowed to use libraries that are not introduced in class.
- Your program should be properly decomposed into tasks and subtasks using functions. To help you with this, use the following:
  - Make each function do one thing and one thing only.
  - No more than 15 lines inside the curly braces of any function, including main(). Whitespace, variable declarations, single curly braces, vertical spacing, comments, and function headers do not count.
  - Functions over 15 lines need justification in comments.
  - Do not put multiple statements into one line.
- No global variables allowed (those declared outside of many or any other function, global constants are allowed).
- No goto function allowed
- No vectors are allowed
- You program should not have any runtime error, e.g. segmentation fault
- Make sure you follow the style guidelines, have a program header and function headers with appropriate comments, and be consistent.

Compile and submit your assignment

When you compile your code, it is acceptable to use C++11 functionality in your program. In order to support this, consider the following approach (note the inclusion of **`-std=c++11`**):

```
g++ -std=c++11 <other flags and parameters>
```

Electronically submit your C++ program (.cpp file, not your executable!!!) by the code due date, on TEACH.

Remember to sign up with a TA to demo your assignment. The deadline of demoing this assignment without penalties is 4/22/2022.