Synth Dolphin  
CS 330 Programming Language Concepts  
Project #1  
Date Handed – In: February 26, 2018

**Design Document**

**Introduction:**

The game of bowling has a total of ten frames with a possibility of a total of 300 points. If two people bowls one game of ten frames each, then they have bowled a total of two games. The first nine frames in which the player has two chances per frame to knock down all the tens pins. If the player knocks down all the ten pins in those two turns in the tenth frame, an extra chance will be awarded.

Bowling.cpp will be a real-world gaming program written in C++ that will keep track of the scores in this game. The program will ask the user to input the score for each bowl per frame in the terminal window. As the user inputs the scores, the corresponding score of that frame along with the total score will be displayed on the terminal window.

**Data Structures:**

The program will use only one data structure. Array data structure will be used in this program. The array balls [] will take score of each bowls in the bowling game. A separate function will be used to check whether the input scores are valid. If the case that the input scores are not valid, the program will loop until the valid score is inputted. As the program receives a valid input, the total score of each bowl, and the score of entire frames that has been attempted will be displayed on the screen.

**Functions:**

Six other functions other than the main() function will be used in the development of this bowling program. The function getInput() will be used to used to accept the scores until the ninth frame. The function tenthFrame() will be used separately to handle the scores of the tenth frame. Function getInput() will accept the input from the user and the function checkInput() will check whether the input is valid. A while loop will be used to check if the input is less than 0 or greater than 10. If this is the case, the program will loop until the valid score is inputted. When the valid score is received, the user will be proceeded to enter the score for the next ball with the help of the function getNextBalls(). The function printScores() will continue to calculate the total score for each bowl and the total score for each frame in the terminal window. This process will continue until the ninth frame.

Function tenthFrame() will be used to keep track of the scores for the tenth frame. As there is a possibility of getting to bowl three times in the tenth frame given that the first two bowls knocks down all ten pins, three separate variables roll1, roll2, and roll3 have been declared within the function. The first bowl of the tenth frame will be ball [18], the second bowl will be ball [19] and the third bowl will be ball [20]. The user will first input the score for the first two bowls in the tenth frame. If the user knocks down all ten pins in these first two bowls, then the third chance id awarded. If the user fails to knock down all tens pins in the first two bowl, the game ends and the program will display the total points for the game in the terminal window.

**Main Program:**

The main() function will be used to process all ten frames. At first, the first nine frames are processed in this block using a while loop. This block of code will continue to run until the ninth frame is completed. The tenth frames is handled separately on a different function tenthFrame() and is simply being called in the main() function for further processing.

**Code List**

|  |
| --- |
|  |
|  |  |  |
|  |  | #include <iostream>  using namespace std;  const int tenthframe = 19; |
|  |  | -int checkInput(int ball, int score, int balls[]); //NEEDS WORK. gets input and checks if input is valid, loops until valid input |
|  |  | -void printScores(int b, int balls[]); |
|  |  | +int getInput(int ball, int balls[]); //gets input and checks if input is valid, loops until valid input |
|  |  | +void printScores(int ball, int balls[]); |
|  |  | int getNextBalls(int i, int balls[]);//returns the value of the next 2 balls |
|  |  | +void tenthFrame(int balls[]); |
|  |  |  |
|  |  | int main() |
|  |  | { |
|  | | @@ -21,39 +22,56 @@ int main() |
|  |  | while(ball < tenthframe) |
|  |  | { |
|  |  | score = 0; |
|  |  | - cout<<"Next Ball: "; |
|  |  | - cin>>score; |
|  |  | - score = checkInput(ball, score, balls);//needs work |
|  |  | + score = getInput(ball, balls); |
|  |  | balls[ball-1]=score; |
|  |  | if(ball % 2 != 0 && score == 10)//is strike... |
|  |  | ++ball; //skip next ball |
|  |  | ++ball; //increment ball |
|  |  | printScores(ball, balls); |
|  |  | } |
|  |  |  |
|  |  | - score = 0; |
|  |  | - //begin special case for frame 10 |
|  |  | + tenthFrame(balls); |
|  |  |  |
|  |  | return 0; |
|  |  | } |
|  |  |  |
|  |  | -void printScores(int b, int balls[]) |
|  |  | +void printScores(int ball, int balls[]) |
|  |  | { |
|  |  | int total=0; |
|  |  | - for(int i=0; i<19; i=i+2) |
|  |  | + |
|  |  | + //print ball scores |
|  |  | + for(int i=0; i<18; i=i+2) |
|  |  | + cout<<"|"<<balls[i]<<" "<<balls[i+1]; |
|  |  | + cout<<"|"<<balls[18]<<" "<<balls[19]<<" "<<balls[20]<<"|"<<endl; |
|  |  | + |
|  |  | + //print frame totals |
|  |  | + for(int i=0; i<ball-1; i=i+2) |
|  |  | { |
|  |  | if(balls[i]!=10) //if not strike... |
|  |  | { //print normally |
|  |  | total += balls[i]+balls[i+1]; |
|  |  | - cout<<balls[i]<<" "<<balls[i+1]<<" Frame Total: "<<balls[i]+balls[i+1]<<endl; |
|  |  | + cout<<"| "<<balls[i]+balls[i+1]<<" "; |
|  |  | } |
|  |  | else //is strike... |
|  |  | { //print special |
|  |  | - total += balls[i]+balls[i+1]+getNextBalls(i+2, balls); |
|  |  | - cout<<balls[i]<<" "<<balls[i+1]<<" Frame Total: "<<(balls[i]+balls[i+1]+getNextBalls(i+2, balls))<<endl; |
|  |  | + if(i!=16)//not in 9th frame |
|  |  | + { |
|  |  | + total += balls[i]+balls[i+1]+getNextBalls(i+2, balls); |
|  |  | + cout<<"| "<<balls[i]+balls[i+1]+getNextBalls(i+2, balls)<<" "; |
|  |  | + } |
|  |  | + else//in 9th frame |
|  |  | + { |
|  |  | + total += balls[i]+balls[18]+balls[19]; |
|  |  | + cout<<"| "<<balls[i]+balls[18]+balls[19]<<" "; |
|  |  | + } |
|  |  | } |
|  |  | } |
|  |  | - cout<<"GRAND TOTAL: "<<total<<endl; |
|  |  | + if(ball >= tenthframe) |
|  |  | + { |
|  |  | + total += balls[18]+balls[19]+balls[20]; |
|  |  | + cout<<"| "<<balls[18]+balls[19]+balls[20]<<" "; |
|  |  | + } |
|  |  | + cout<<"|\nGRAND TOTAL: "<<total<<endl; |
|  |  | } |
|  |  |  |
|  |  | //returns value of next 2 balls |
|  | | @@ -71,18 +89,104 @@ int getNextBalls(int i, int balls[]) |
|  |  | } |
|  |  |  |
|  |  |  |
|  |  | -int checkInput(int ball, int score, int balls[]) |
|  |  | +int getInput(int ball, int balls[]) |
|  |  | { |
|  |  | - while((score < 0 || score > 10) || (ball%2 == 0 && (balls[ball-1] + score > 10))) |
|  |  | + int score = 0; |
|  |  | + bool valid = false; |
|  |  | + bool notInt = false; |
|  |  | + |
|  |  | + cout<<"Next Ball: "; |
|  |  | + cin>>score; |
|  |  | + |
|  |  | + if(cin.fail())//not an int type |
|  |  | + { |
|  |  | + notInt = true; |
|  |  | + cin.clear(); |
|  |  | + } |
|  |  | + |
|  |  | + while(notInt) |
|  |  | + { |
|  |  | + cin.clear(); |
|  |  | + cout<<"Invalid Input1. Try again:"; |
|  |  | + cin.ignore(); |
|  |  | + cin>>score; |
|  |  | + if(cin.good()) |
|  |  | + notInt=false; |
|  |  | + } |
|  |  | + |
|  |  | + if(ball < tenthframe)//not tenth frame |
|  |  | { |
|  |  | - cout<<"Invalid Input. Try again."<<endl; |
|  |  | - cin>> score; |
|  |  | + while( !valid ) |
|  |  | + { |
|  |  | + if( score >= 0 && score <=10 )// score is in range |
|  |  | + { |
|  |  | + if(ball%2 == 0 && balls[ball-2]+score <= 10)//on 2nd ball and ball1 + input <= 10 |
|  |  | + valid = true; |
|  |  | + else if(ball%2 != 0) |
|  |  | + valid = true; |
|  |  | + } |
|  |  | + |
|  |  | + if( !valid )//input not valid |
|  |  | + { |
|  |  | + cout<<"Invalid Input2. Try again:"; |
|  |  | + cin>> score; |
|  |  | + } |
|  |  | + } |
|  |  | + } |
|  |  | + else //is tenth frame |
|  |  | + { |
|  |  | + while( !valid ) |
|  |  | + { |
|  |  | + if( score >= 0 && score <= 10 )//score is in range |
|  |  | + { |
|  |  | + if( balls[18]+balls[19]+balls[20]+score <= 30)//10th frame total <= 30 |
|  |  | + valid = true; |
|  |  | + } |
|  |  | + |
|  |  | + if( !valid )//invalid input |
|  |  | + { |
|  |  | + cout<<"Invalid Input3. Try again."<<endl; |
|  |  | + cin>>score; |
|  |  | + } |
|  |  | + } |
|  |  | } |
|  |  | return score; |
|  |  | } |
|  |  |  |
|  |  |  |
|  |  | +void tenthFrame(int balls[]) |
|  |  | +{ |
|  |  | + int roll1, roll2, roll3; //3 rolls of the 10th frame |
|  |  | + int total\_score = 0; //total score for the 10th frame |
|  |  | + int tenthTotal = 0; |
|  |  | + |
|  |  | + //get roll1 |
|  |  | + roll1 = getInput(tenthframe, balls); |
|  |  | + balls[18]=roll1; |
|  |  | + |
|  |  | + //print scores |
|  |  | + printScores(tenthframe, balls); |
|  |  | + |
|  |  | + //validate roll2 |
|  |  | + roll2 = getInput(tenthframe, balls); |
|  |  | + balls[19]=roll2; |
|  |  | + |
|  |  | + //print scores |
|  |  | + printScores(tenthframe, balls); |
|  |  | + |
|  |  | + //begin roll 3 if stike or spare |
|  |  | + if(roll1+roll2 >= 10) |
|  |  | + { |
|  |  | + //get roll3 |
|  |  | + roll3 = getInput(tenthframe, balls); |
|  |  | + balls[20]=roll3; |
|  |  | + |
|  |  | + //print scores |
|  |  | + printScores(tenthframe, balls); |
|  |  |  |
|  |  | + } |
|  |  | + //end of game... |
|  |  | +} |
|  |  |  |
|  |  |  |
|  |  |  |

**Test Data Plan**

|  |  |  |
| --- | --- | --- |
| **Valid Input** |  |  |
|  |  |  |
|  |  |  |
| **Boundary Input** |  |  |
| **Invalid Input** |  |  |
|  |  |  |