Lab 3: Dice Roll

Thomas Bischoff

Original

Date: 2 / 26 / 18

Problem: Simulate a Dice Game where if the player rolls a 7 or 11 on their first roll they win, if they roll 2, 3, or 12 on their first roll they lose. If their sum is 4, 5, 6, 8, 9, or 10 that number becomes their point and they must re-roll that number to win but if they roll a 7 before then they lose. Once done ask the player if they would like to play again.

Pseudocode –

Level One: 1. Simulate the roll for a set of dice. 2. Check if the player has met the win or loss parameters on the first roll. A. If the player won tell them they won. B. If the player lost tell them they lost. C. If the player did not win or lose on the first roll, have the player roll the dice until they re-roll their original number indicating a win or a “7” indicating a loss. 3. Tell the player the numbers of wins and losses they have had thus far. 4. Ask the player if they would like to play again. A. If “Yes” restart the game. B. If “No” end the game saying goodbye to the player.

Level Two: int main() - 1. Start the Game. 2. Use the first\_dice\_roll() function. 3. Check if the player won or lost. A. If the player won tell them they won. B. If the player lost tell them they lost. C. Otherwise use the followup\_dice\_roll(int points) function. 4. Check if the player won or lost. A. If the player won tell them they won. B. If the player lost tell them they lost. 5. Show the player the number of games they have won and lost. 6. Use users\_choice()

Pseudocode –

Level Two: int first\_dice\_roll() - 1. Roll the first pair of dice. 2. Check if the player rolled a “7” or “11” indicating a win. A. Tell main() that the player won. 3. Check if the player rolled a “2”, “3”, or “12” indicating a loss. A. Tell main() that the player lost. B. Otherwise tell main() the numbers rolled by the player. int followup\_dice\_roll(int points) - 1. Start to roll pairs of dice. 2. Check if the sum of the dice rolled is equal to “points”. A. If true, stop rolling the dice and tell main() that the player won. 3. Check if the sum of the dice rolled is equal to “7”. A. If true, stop rolling the dice and tell main() that the player lost. B. If false, roll another pair of dice bool users\_choice() - 1. Ask the user if they would like to play again. 2. If the player answers “Yes”, stop asking them if they would like to play again and tell main() that they want to play again. 3. If they player answers “No”, stop asking them if they would like to play again, tell them “Goodbye.”, and tell main() to stop playing. 4. If the player answers with something not recognized re-ask the question.

Pseudocode –

Level Three: int main() - 1. Set the current time. 2. Initialize the number of wins to “0”. 3. Initialize the number of losses to “0”. 4. Initialize a statement as “True” to run a while loop. 5. Generate a while loop. A. Initialize a value for the first roll of dice. B. Set the first roll of dice equal to what is returned by first\_dice\_roll(). C. Check if first roll of dice is equal to “0”. + Display a message indicating that the player has won. + Add one to the number of wins D. Check if first roll of dice is equal to “1”. + Display a message indicating that the player has lost. + Add one to the number of losses. E. If first roll of dice was neither “0” or “1”. + Initialize a value for the follow up rolls. + Set follow up rolls to be equal to the value returned by followup\_dice\_rolls(int points) where points is equal to the value of first roll of dice. + Check if follow up rolls is equal to “0”. - Display a message that the player has won. - Add one to the number of wins. + Check if follow up rolls is equal to “1”. - Display a message that the player has lost. - Add one to the number of losses. F. Display the win and loss results. G. Set the statement to be equal to the value returned by users\_choice().

Pseudocode –

Level Three: int first\_dice\_roll() - 1. Initialize the pair of dice. 2. Have the first die set to a random number between “1” and “6”. 3. Have the second die set to a random number between “1” and “6”. 4. Initialize the sum of the dice rolled to be equal to the value of die one plus the value of die two. 5. Display the numbers on the dice and their sum. 6. Initialize a value to hold the points for the player. 7. Check if the sum of the dice rolled is equal to “7” or “11”. A. Set points to “0” to indicate a win. B. Return the points indicating a player win. 8. Check if the sum of the dice rolled is equal to “2”, “3”, or “12”. A. Set points to “1” to indicate a loss. B. Return the points indicating a player loss. 9. Otherwise. A. Set the points to be equal to the sum of the dice rolled. B. Return the sum of the dice as points. int followup\_dice\_roll(int points) - 1. Initialize win result. 2. Initialize a statement to be true. 3. Generate a while loop. A. Initialize the pair of dice. B. Set die one to be equal to a number between “1” and “6”. C. Set die two to be equal to a number between “1” and “6”. D. Initialize a variable for the sum of the dice to be equal to die one plus die two. E. Display the Numbers rolled and their sum.

Pseudocode –

Level Three: int followup\_dice\_roll(int points) - F. Check if the sum of the dice is equal to “points”. + Set the win result to be equal to “0” indicating a win. + Set the statement to be “False”. G. Check if the sum of the dice is equal to “7”. + Set the win results to be equal to “1” indicating a loss. + Set the statement to be “False”. 4. Return the win results. bool users\_choice() - 1. Initialize the user’s statement. 2. Initialize a statement as “True”. 3. Generate a while loop. A. Initialize the user’s response. B. Display a message asking if the player would like to play again. C. Set user’s response to be equal to the player’s input. D. Convert the user’s response to uppercase. E. Check if the player indicated that they want to continue. + Set user’s statement to be equal to “True”. + Set statement to be equal to “False”. F. Check if the player indicated that they don’t want to continue. + Display a “goodbye” message. + Set user’s statement to be “False” + Set statement to be “False”. G. Otherwise. + Display that the response entered was not recognized. 4. Return the user’s statement.

Code –

1. //######################################################################
2. // Program Header: dice\_roll.cpp
3. // The function of this program is to simulate the rolling of two six
4. // sided dice, where their sum will be calculated. If the player rolls
5. // a 7 or 11 on their first roll they win, if they roll 2, 3, or 12 on
6. // their first roll they lose. If their sum is 4, 5, 6, 8, 9, or 10 that
7. // number becomes their point and they must re-roll that number to win
8. // but if they roll a 7 before then they lose. Once done ask the player
9. // if they would like to play again.
10. // Author: Thomas Bischoff
11. // Date Created: 2/13/18
12. //######################################################################
14. #include <iostream>
15. #include <cstdlib>
16. #include <ctime>
17. #include <string>
19. **using** **namespace** std;
21. // Function Declarations
22. **int** first\_dice\_roll();
23. **int** followup\_dice\_roll(**int** points);
24. **bool** users\_choice();
26. **int** main()
27. {
28. // Set Current Time
29. srand(time(0));
30. // Initialize the Number of Wins
31. **int** wins = 0;
32. // Initialize the Number of Losses
33. **int** losses = 0;
34. // Initialize a Statement as True to Run a While Loop
35. **bool** run\_statement = **true**;
36. // Generate a While Loop
37. **while** (run\_statement == **true**)
38. {
39. // Initialize a Value for the First Roll of Dice
40. **int** first\_roll;
41. // Roll the First Pair of Dice
42. first\_roll = first\_dice\_roll();
43. // Check if the Player Won
44. **if** (first\_roll == 0)
45. {
46. // Display a Message Indicating that the Player Won
47. cout << "Congratulations, You Win!" << endl;
48. // Add to the Win Count
49. wins += 1;
50. }
51. // Check if the Player Lost
52. **else** **if** (first\_roll == 1)
53. {
54. // Display a Message Indicating that the Player Lost
55. cout << "Sorry, You Lost." << endl;
56. // Add to the Loss Count
57. losses += 1;
58. }
59. // Check if the Player Neither Won or Loss
60. **else**
61. {
62. // Initialize a Value for the Follow Up Rolls
63. **int** followup\_rolls;
64. // Perform the Follow Up Rolls
65. followup\_rolls = followup\_dice\_roll(first\_roll);
66. // Check if the Player Won
67. **if** (followup\_rolls == 0)
68. {
69. // Display a Message Indicating that the Player Won
70. cout << "Congratulations, You Win!" << endl;
71. // Add to the Win Count
72. wins += 1;
73. }
74. // Check if the Player Lost
75. **if** (followup\_rolls == 1)
76. {
77. // Display a Message Indicating that the Player Lost
78. cout << "Sorry, You Lost." << endl;
79. // Add to the Loss Count
80. losses += 1;
81. }
82. }
83. // Display the Win and Loss Results
84. cout << "Game Statistics:" << endl;
85. cout << "  Wins = " << wins << endl;
86. cout << "  Losses = " << losses << endl;
87. // Check if the User Want to Play Again
88. run\_statement = users\_choice();
89. }
90. }
92. //######################################################################
93. // Function 1: int first\_dice\_roll()
94. // Purpose: The purpose of this function is to "roll" the dice and
95. //          determine if the player has won or lost the game.
96. //######################################################################
98. **int** first\_dice\_roll()
99. {
100. // Initialize the Pair of Dice
101. **int** die\_one, die\_two;
102. // Roll the First Die
103. die\_one = rand()% 6 + 1;
104. // Roll the Second Die
105. die\_two = rand()% 6 + 1;
106. // Initialize the Sum of the Dice Rolled
107. **int** dice\_sum = die\_one + die\_two;
108. // Display the Numbers on the Dice and their Sum
109. cout << "You Rolled a " << die\_one << " and a " << die\_two << " = " << dice\_sum << endl;
110. // Initialize a Value to Hold the Points for the Player
111. **int** points;
112. // Check if the Player Won
113. **if** ((dice\_sum == 7) or (dice\_sum == 11))
114. {
115. // Set Points to 0 to Indicate a Win
116. points = 0;
117. // Return the Points Indicating the Player Won
118. **return** points;
119. }
120. // Check if the Player Lost
121. **if** ((dice\_sum == 2) or (dice\_sum == 3) or (dice\_sum == 12))
122. {
123. // Set Points to 1 to Indicate a Loss
124. points = 1;
125. // Return the Points Indicating the Player Lost
126. **return** points;
127. }
128. // Check if the Player Neither Won or Loss the First Roll
129. **else**
130. {
131. // Set the Points to be Equal to the Sum of the Dice Rolled
132. points = dice\_sum;
133. // Return the Sum of the Dice as Points
134. **return** points;
135. }
136. }
138. //######################################################################
139. // Function 2: int followup\_dice\_roll(int points)
140. // Purpose: The purpose of this function is "roll" a set of dice until
141. //          they land on the players point or reach a 7.
142. //######################################################################
144. **int** followup\_dice\_roll(**int** points)
145. {
146. // Initialize Win Result
147. **int** win\_result;
148. // Initialize a Statement to be True
149. **bool** repeat = **true**;
150. // Generate a While Loop
151. **while** (repeat == **true**)
152. {
153. // Initialize the Pair of Dice
154. **int** die\_one, die\_two;
155. // Find the Value "Rolled" by Die One
156. die\_one = rand()% 6 + 1;
157. // Find the Value "Rolled" by Die Two
158. die\_two = rand()% 6 + 1;
159. // Initialize a Variable for the Sum of the Two Dice
160. **int** dice\_sum;
161. // Calculate the Sum of the Dice Rolled
162. dice\_sum = die\_one + die\_two;
163. // Display the Numbers Rolled by the Player and their Sum
164. cout << "You Rolled a " << die\_one << " and a " << die\_two << " = " << dice\_sum << endl;
165. // Check if the Player Won
166. **if** (dice\_sum == points)
167. {
168. // Set the Win Results to 0 to Indicate a Win
169. win\_result = 0;
170. // Set Repeat to be False
171. repeat = **false**;
172. }
173. // Check if the Player Lost
174. **if** (dice\_sum == 7)
175. {
176. // Set the Win Results to 1 to Indicate a Loss
177. win\_result = 1;
178. // Set Repeat to be False
179. repeat = **false**;
180. }
181. }
182. // Return the Win Results
183. **return** win\_result;
184. }
186. //######################################################################
187. // Function 3: bool users\_choice()
188. // Purpose: The purpose of this function is return the players decision
189. //          if they want to continue playing or not
190. //######################################################################
192. **bool** users\_choice()
193. {
194. // Initialize the Users Statement
195. **bool** users\_statement;
196. // Initialize a Statement as True
197. **bool** repeat = **true**;
198. // Generate a while Loop
199. **while** (repeat == **true**)
200. {
201. // Initialize the User's Response
202. **char** users\_answer;
203. // Ask the Player if they Want to Play Again
204. cout << "Would You Like to Play Again (Y / N): ";
205. // Take the User's Answer
206. cin >> users\_answer;
207. // Convert User's Anser
208. users\_answer = toupper(users\_answer);
209. // Check if the Player Wants to Conitnue
210. **if** (users\_answer == 'Y')
211. {
212. // Set User's Statement to True
213. users\_statement = **true**;
214. // Set Repeat to False
215. repeat = **false**;
216. }
217. // Check if the Play Does Not Want to Continue
218. **else** **if** (users\_answer == 'N')
219. {
220. // Display Goodbye Message
221. cout << "Goodbye." << endl;
222. // Set User's Statement to False
223. users\_statement = **false**;
224. // Set Repeat to False
225. repeat = **false**;
226. }
227. // Check if the Player Made a Mistake with their Input
228. **else**
229. // Display that the Response they Entered Cannot be Read
230. cout << "Sorry, but the Response you Entered is Not Recognized. Please Try Again." << endl;
231. }
232. // Return the User's Statement
233. **return** users\_statement;
234. }

Sample Output –

You Rolled a 2 and a 2 = 4 You Rolled a 3 and a 6 = 9 You Rolled a 6 and a 6 = 12 You Rolled a 2 and a 5 = 7 Sorry, You Lost. Game Statistics: Wins = 0 Losses = 1 Would You Like to Play Again (Y / N): y You Rolled a 3 and a 6 = 9 You Rolled a 4 and a 2 = 6 You Rolled a 5 and a 2 = 7 Sorry, You Lost. Game Statistics: Wins = 0 Losses = 2 Would You Like to Play Again (Y / N): u Sorry, but the Response Entered is Not Recognized. Please Try Again. Would You Like to Play Again (Y / N): y You Rolled a 6 and a 4 = 10 You Rolled a 5 and a 3 = 8 You Rolled a 1 and a 3 = 4 You Rolled a 5 and a 4 = 9 You Rolled a 5 and a 2 = 7 Sorry, You Lost. Game Statistics: Wins = 0 Losses = 3 Would You Like to Play Again (Y / N): n Goodbye.