Sixes and Sevens Game Journal

Application Start: 15/04/2019

* Form designed and created.
* Die 1, Die 2, Running Score, Player 1 and Player 2 (Cumulative Scores) fields used.
* Roll Dice, Pass Turn and Exit Buttons created.
* Roll Dice button initialized containing code to roll 2 dice randomly.
* Die 1 and Die 2 fields display the result of each random die roll from when the Roll Dice button is clicked.
* At this point it was decided to have Die 1 roll 1-6 and Die 2 roll 7-12 to test achieving 2 different outcomes as having both dice use 1-6 randomly always ended in the exact same result making it not as random as needed.
* Running Score field contains the random result of the Roll Dice button added together and displays the new result after each time the Roll Dice button is clicked.
* Exit button initialized to close the application when clicked.

28/04/2019

* A slight redesign in the form by narrowing the whole window size
* A change was made to how the random function on Roll Dice button works.
* The range for the Random function was changed back to between 1-6
* One instance of "Random rRollDice = new Random();" is now used and each instance of iDie1 and iDie2 calls the single "Random rRollDice = new Random();" to get separate random results for each Dice, now resulting in more random rolls.
* A message box has been added to test the "if…else" function to check when a "7" is rolled displaying that the player has lost and needs to pass the turn.
* The next step will be to add a check for the current player to be highlighted as the "Active Player". This should hopefully enable the "Pass Turn" button to select the next player and clear up the text boxes and add the running score to cumulative score once clicked.

04/05/2019

* Today we set the Roll Dice event handler to check whether the result of iDie1 and iDie2 have 6 and 1 to show a message box displaying that the player has lost and needs to pass the turn.
* Included a 'Start Game' button that will be displayed on start up with the 'Roll Dice' and 'Pass Turn' buttons not displayed until the 'Start Game' button is pressed, setting the 'Roll Dice' and 'Pass Turn' buttons to visible true and the 'Start Game' button to visible false.
* When 'Start Game' button is pressed it also sets the 'Player 1' label's background colour to Red to establish the active first player.

9/5/2019

* Today we worked on getting the cumulative and turn switching functionalities. To start off with we got the back colours to change with the btnPassTurn OnClick event. This works with an if statement that checks the current labels back colour and if its red, it changes it to the default colour.
* Next, we set most of the game buttons to not visible and added a function to set them to visible in the Start Game button. After this we added 2 Boolean variables called p1Turn and p2Turn. Then we added various “If” statements to switch the players turns by changing the Booleans value to False. For example, if player 1 rolls a six, their Boolean value gets set to false and p2Turns value gets set to true.
* Each time the roll button gets clicked, both the running score and cumulative score get recalculated. Then the Player 1 and Player 2 cumulative score is adjusted depending on which Player is set to true.

13/5/2019

* Today was focused a lot on fixing minor things like setting some buttons to invisible until Start is clicked. Alexander and I also talked about making the scores display in a listbox instead of a textbox so that the players can see how much they added each turn.
* We also decided to add a message box every time it’s a new turn. We also realized a fault with the pass turn as it was setting P1Turn to false and P2Turn to true when both players select Pass Turn.

14/5/2019

* Today we added the dice display system so that when a player rolls the dice, there is a visual display for each die. So far it is only using draw ellipse because it is similar code compared to the dice drawing lab.

15/5/2019

* Today we added a message box for when a player reaches 50 points. When they reach 50 points we added a pop-up box to check if they wanted to play again. If they select yes, everything resets. When they select yes, we also added a label saying “Games Won: “, this shows how many games the player has beaten his opponent to 50.

20/05/2019

* Today we decided that all DrawEllipse Graphics functions would be changed to FillEllipse Graphics functions to have each die face show filled in black circles instead of empty circles.
* This was achieved by using SolidBrush brshBlack = new SolidBrush(Color.Black); within the btnRollDice\_Click event handler and then each time the button is clicked, the IF statements for iDie1 and iDie2 call it and create the filled in circle depending on the random result. This was done to replicate the actual look of a set of Dice for the user to recognize better.
* Graphics graPaper = picBxDie1.CreateGraphics(); and Graphics graPaper2 = picBxDie2.CreateGraphics(); were removed from each IF Statement and used before the IF Statements for each to be called once instead of during each If Statement check
* In depth comments added throughout application
* Partitioned code into Regions to better understand the working sections
* Reference - (ThreeFlagsDemo)

21/05/2019

* Today we added 2 List boxes lstBxPlayer1 and lstBxPlayer2 to replace the 2 text boxes txtBxPlayer1 and txtBxPlayer2 containing the Cumulative Scores
* This added the running score to the list box every time the Roll Dice button was clicked to keep track of each dice roll total
* The lstBxPlayer1 and lstBxPlayer2 fields were added to the Sixes and Sevens Initialization to have their visibility set to false when the application is run:
  + lstBxPlayer1.Visible = false;
  + lstBxPlayer2.Visible = false;
* Reference (Lec7Demo – ListboxDemo)
* A For Loop was added to add an iterative count when a player wins a round and 1 round is added to that players’ “Rounds Won” count.
* Eg:

p1RoundsWon = p1RoundsWon + 1;

for (int i = 1; i <= p1RoundsWon; i++)

{

lblP1Rounds.Text = Convert.ToString(p1RoundsWon);

}

* Reference (Lec5Demo – LoopTests)
* Made a few changes to the ListBoxes so that they show when a player score gets reset.
* Added text boxes before the user starts the game so that they can provide their own name.
* Had to change the Message Boxes so that they display the user’s entered name + “’s Turn”.

22/05/2019

* Added a text box to the start so that the user can provide the score limit. Changed the variable iGoalScore so that it is equal to the inputted score limit.
* Added a computer checkbox to the start of the game. This works by simulating the roll dice button using “.PerformClick”.
* Added System.Threading.Sleep so that the computer doesn’t run its turn instantly. The computer rolls the dice twice and then passes turn. But currently the dice scores are both going instantly.
* Added a Try Catch statement to the game start that checks if a number is entered.

25/05/2019

* Removed the Try Catch statement to replace it with an If…Else Statement to check the ScoreLimit input.
* Removed the this.Close() function after the Try Catch statement to give the user time to input the details into the required fields.
* Added a computerFillEllipse() method which contains the exact same drawing code as the btnRollDice\_Click event handler. This is now called during the computers turn when rolling the dice both times to display the dice graphics each time the computer rolls the dice.
* Fixed the PassTurn button to include passing the turn to the computer and it automatically taking its turn using the same code from the RollDice event handler if the computer check box is checked.