Overview

Exercise\_14\_8 provides a guessing game UI and accompanying classes for a user to guess a number between 1-1000 and be given clues to help guide him.

### PROCESSING LOGIC

UI Logic:

1. Create GuessingGame instance
2. Monitor if Return key is pressed or the EnterButton is pressed
3. Get the text from the GuessingTextBox and send it to UserGuess(int).
4. If Answer.getCorrect()
   1. Then disable everything except Try Again button, make back ground Green and tell the user “Congrats!”
5. else
   1. set the ComputerVoiceLabel text to Answer.getClue()
   2. Check Answer.getWarmOrCold(), true set background color to red, false set to blue
   3. Reset GuessingTestBox.Text to blank
6. If Try Again button is pressed, then create a new GuessingGame instance, reset ComputerVoiceLabel, GuessingTextBox, and EnterButton

GuessingGame Logic:

1. On construction, create a new Random int between 0 and 1000 and set numberToGuess to that int.
2. On UserGuess(int), take the guess, figure out if it is closer to the numberToGuess than the last guess, and if it is higher or lower than the numberToGuess, and return an Answer with the correct variables.

### DATA (INPUT/OUTPUT)

Input: int: GuessingTextBox

Output: string: “Too high” or “Too low” or “Correct! Congrats!”

colors: Red, Blue, or Green

### COMPONENTS (SOURCE CODE NAMES, CLASSES, METHODS)

|  |
| --- |
| **GuessingGameUI** |
|  |
| -button2\_Click(object, EventArgs): void  -TryAgainButton\_Click(object, EventArgs): void  -GuessingTextBox\_KeyUp(object, KeyEventArgs): void |

|  |
| --- |
| **GuessingGame** |
| -numberToGuess: int  -lastGuess: int |
| +GuessingGame(): constructor  +UserGuess(int): Answer |

|  |
| --- |
| **Answer** |
| -m\_clue: string  -m\_correct: bool  -m\_WarmOrCold: bool |
| +Answer(string, bool, bool): constructor  +getWarmOrCold(): bool  +getCorrect(): bool  +getClue(): string |

### TESTING

Scenario 1 – Play Game: Binary Search Pattern Play Through

Steps to test:

1. Start program
2. Play game using a Binary Search Pattern to play.
3. Test Try again button, play again using Binary Search Pattern.
4. Exit program

Expected reaction:

For the game to accurately tell the player whether he is warm or cold and too high or too low based off red or blue and text saying too high or too low. Once the user gets the right answer, textbox and EnterButton should be disabled, background should be green, and the label should say Congrats. Try Again button should reset the game with a new number to guess.

Actual result:

Expected reaction was actual result. Program works.

##### 