due in 6day 6h

Submission Phase

1. Do assignment ☐ (/gameprogramming-001/human_grading/view/courses/971200/assessments/9/submissions)

Evaluation Phase

2. Evaluate peers 🔒 (/gameprogramming-001/human_grading/view/courses/971200/assessments/9/peerGradingSets)

Results Phase

3. See results 🔒 (/gameprogramming-001/human_grading/view/courses/971200/assessments/9/results/mine)

☐ In accordance with the Honor Code, I certify that my answers here are my own work, and that I have appropriately acknowledged all external sources (if any) that were used in this work.

Save draft | Submit for grading

ASSIGNMENT DESCRIPTION

Your job for this assignment is to develop a console application that plays a new version of War that uses a 13-sided die instead of cards. The rules of the game are simple. A game of War consists of 21 battles. Each battle consists of the two players each rolling a 13-sided die. Whichever player has the higher die value wins the battle. If there's a tie in a battle, then "WAR" is declared and the die are rolled again until one of the players wins the battle. Whoever wins the most battles wins the war. After the winner has been announced, the player may choose to play another game of War. Complete games of War are played until the player decides not to play again.

PEER GRADING

After submitting your work (described also in the two question parts below), you'll get the chance to grade the work of **five** of your peers. Your own work will also be assessed by your peers, from which we'll get your grade. Since you've worked hard on your submission and would like your peers to do a good job of assessing your submission, please take your time and do a good job of assessing your peer's submission in return.

HONOR CODE

Please remember that you have agreed to the Honor Code, and your submission should be entirely yours. Our definition of plagiarism follows from standard literature: passing off someone else's work as your own, whether from your peer or Wikipedia.

REQUIREMENTS

Start up the IDE and create a new Console Application project named ProgrammingAssignment5.

Your program must do the following (all in the Main method):

- Declare a variable for a random number generator and create a new Random object for that variable
- Declare other variables as necessary to keep track of each player's roll, each player's number of wins, and whether or not we should play another game of War
- Print a "welcome" message to the user telling them that the program will play games of War
- While the player wants to play another game
    - Set both player win counts to 0
    - Loop for 21 battles
        - Roll the die for player 1
        - Roll the die for player 2
        - While the rolls have the same value (a tie)
        - Print "WAR" and the die values for the two players
        - Roll the die for player 1
        - Roll the die for player 2
        - Print the die values for the two players
        - Print which player won and increment that player's win count
    - Print out which player won more battles
    - Prompt for and get whether the player wants to play another game

ASSUMPTIONS

- Users may enter any character when asked to play again. Accept only 'n' or 'N' to end the program. Any other entry plays another game of War.
- If you develop your solution to this problem using the appropriate programming constructs, you should have the following constructs in your application:
    - for loop: executes the 21 battles for each war
    - while loops:
        - rolls the die again whenever there is a tie [there can be multiple wars in a row!]
        - keeps playing games of war
    - if statements:
        - determines battle winner
        - determines war winner

EXAMPLE OUTPUT

Here's some example output. You don't have to match this exactly, but your output format should look pretty close to this (your player rolls will obviously be different). You definitely need to make sure the columns are lined up for the player rolls and winner messages.

```
Welcome to War!

BATTLE: P1:6    P2:8    P2 Wins!
BATTLE: P1:6    P2:10   P2 Wins!
BATTLE: P1:8    P2:3    P1 Wins!
BATTLE: P1:5    P2:3    P1 Wins!
BATTLE: P1:13   P2:6    P1 Wins!
BATTLE: P1:11   P2:3    P1 Wins!
BATTLE: P1:8    P2:9    P2 Wins!
BATTLE: P1:5    P2:4    P1 Wins!
BATTLE: P1:9    P2:5    P1 Wins!
BATTLE: P1:2    P2:10   P2 Wins!
```

```
BATTLE: P1:4    P2:5    P2 Wins!
BATTLE: P1:6    P2:3    P1 Wins!
BATTLE: P1:11   P2:12   P2 Wins!
BATTLE: P1:12   P2:1    P1 Wins!
BATTLE: P1:5    P2:2    P1 Wins!
BATTLE: P1:6    P2:11   P2 Wins!
BATTLE: P1:9    P2:13   P2 Wins!
BATTLE: P1:10   P2:6    P1 Wins!
BATTLE: P1:8    P2:7    P1 Wins!
BATTLE: P1:7    P2:8    P2 Wins!
   WAR! P1:4    P2:4
BATTLE: P1:1    P2:13   P2 Wins!

P1 is the overall Winner with 11 battles!

Do you want to play again (y/n)? Y

BATTLE: P1:10   P2:5    P1 Wins!
BATTLE: P1:7    P2:5    P1 Wins!
BATTLE: P1:1    P2:9    P2 Wins!
BATTLE: P1:12   P2:4    P1 Wins!
BATTLE: P1:5    P2:9    P2 Wins!
BATTLE: P1:3    P2:1    P1 Wins!
BATTLE: P1:2    P2:3    P2 Wins!
BATTLE: P1:12   P2:9    P1 Wins!
   WAR! P1:5    P2:5
BATTLE: P1:11   P2:12   P2 Wins!
   WAR! P1:1    P2:1
   WAR! P1:8    P2:8
   WAR! P1:12   P2:12
BATTLE: P1:9    P2:2    P1 Wins!
BATTLE: P1:11   P2:8    P1 Wins!
BATTLE: P1:6    P2:13   P2 Wins!
BATTLE: P1:4    P2:7    P2 Wins!
   WAR! P1:7    P2:7
BATTLE: P1:6    P2:7    P2 Wins!
BATTLE: P1:13   P2:10   P1 Wins!
BATTLE: P1:4    P2:2
P1 Wins!
BATTLE: P1:7    P2:13   P2 Wins!
BATTLE: P1:11   P2:9    P1 Wins!
   WAR! P1:5    P2:5
BATTLE: P1:8    P2:12   P2 Wins!
BATTLE: P1:3    P2:13   P2 Wins!
BATTLE: P1:7    P2:3    P1 Wins!

P1 is the overall Winner with 11 battles!

Do you want to play again (y/n)? n

Press any key to continue . . .
```

HELPFUL HINTS

Write 5 or fewer lines of code, save, compile, test, repeat! Don't try to write a whole block of code at once – implement one small chunk at a time.

You'll probably want to use the \t escape sequence (to add a tab) when you output each row above to line up the columns properly

This portion of the assignment is about using the appropriate coding style in your Program class. This portion of the assignment is worth 1 point.

Copy and paste your entire Program.cs file into the text box below. Clicking the < code > tab at the top of the text box and pasting inside the resulting grey box will keep all your spacing intact.

B | I | Link | <code> | <HTML>

Default Editor

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Evaluate the source code for the following 2 criteria:

1. Follows coding standards (descriptive names and proper capitalization for variable names, appropriate commenting and blank lines)
2. Code uses appropriate implementation approach to solve the problem (no unneeded variables, doesn't use significantly more code than it should to solve the problem)

This portion of the assignment is about correct program behavior. This portion of the assignment is worth 7 points.

Upload two screen shots of your console window -- one before you answer yes to play again after the first game of War and the other before you've pressed a key to close the program. Be sure to run using Ctrl-F5 and make sure your screen shots are saved as png files. Click the Browse button just below the text box below to upload your screen shots (one at a time); your screen shots will be added to the text box.

B | I | Link | <code> | <HTML>

Default Editor

Attach a file

Evaluation/feedback on the above work

Note: this section can only be filled out during the evaluation phase.

Evaluate the screen shots for the following 7 criteria:

1. Prints welcome message
2. Displays each battle, with ties shown as a WAR
3. Displays battle results with appropriate column formatting
4. Correctly handles ties by re-rolling die
5. Decides and displays battle winner properly
6. Decides and displays war winner properly
7. Plays another game or not correctly based on user input

Overall evaluation/feedback

Note: this section can only be filled out during the evaluation phase.

In accordance with the Honor Code, I certify that my answers here are my own work, and that I have appropriately acknowledged all external sources (if any) that were used in this work.

Save draft | Submit for grading