Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-005-F2024/it114-module-3-number-guesser-4/grade/el286

Course: IT114-005-F2024
Assigment: [IT114] Module 3 Number Guesser 4
Student: Erik L. (el286)

Submissions:

Submission Selection

1 Submission [submitted] 10/4/2024 9:41:28 PM

Instructions

COLLAPSE

Overview Video: https://youtu.be/ej6lWrg9XjE

1. Create the below branch name
2. Implement the NumberGuess4 example from the lesson/slides
    1. https://gist.github.com/MattToegel/aced06400c812f13ad030db9518b399f
    2. Add/commit the files as-is from the lesson material (this is the base template).
    3. Push the changes to the HW branch and create a pull request to keep open until this assignment is done
3. Pick two (2) of the following options to implement
    1. Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level)
    2. Implement anti-data tampering of the save file data (reject user direct edits)
    3. Add a difficulty selector that adjusts the max strikes per level (i.e., "easy" 10 strikes, "medium" 5 strikes, "hard" 3 strikes)
    4. Display a cold, warm, hot indicator based on how close to the correct value the guess is (example, 10 numbers away is cold, 5 numbers away is warm, 2 numbers away is hot; adjust these per your preference) Only display this when the wrong guess doesn't roll back the level
    5. Add a hint command that can be used once per level and only after 2 strikes have been used that reduces the range around the correct number (i.e., number is 5 and range is initially 1-15, new range could be 3-8 as a hint)
    6. Implement separate save files based on a "What's your name?" prompt at the start of the game (each person gets their own save file based on user's name)
4. Fill in the below deliverables
5. Save changes and export PDF

6. Git add/commit/push your changes to the HW branch
7. Create a pull request to main (if not done so before)
8. Complete the pull request (don't forget to locally checkout main and pull changes to prep for future work)
9. Upload the same PDF to Canvas

Branch name: M3-NumberGuesser-4

Group

100%

Group: Implementation 1
Tasks: 1
Points: 4

COLLAPSE

Task

100%

Group: Implementation 1
Task #1: Implementation Evidence
Weight: ~100%
Points: ~4.00

COLLAPSE

Details:
Code screenshots must have ucid/date shown as a comment in the code.

Explanations must be your own words describing the logic and how the solution code solves the problem.

Columns: 1

Sub-Task

100%

Group: Implementation 1
Task #1: Implementation Evidence
Sub Task #1: Mention which option you picked and how you solved it

Task Response Prompt

Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

Response:

I chose option 4 where you inform the user if the number they inputed is either cold, warm, or hot from the original random number. I used an if statement to determine if the amount of strikes is less than the max which is 5 and would get difference by subtracting the random number with the user's guess into a variable. Now that I have the difference, I used more if statements to determine if the difference fits the range that I assigned. If the number was greater than or equal to 10, then it'll display "You're cold". If not then It'll then check if it's between 4 and 9 and will display "You're warm". I used the && in order to get more a accurate approach for the number than just leaving it to be something like "4<= difference". The last if will be the range between 1 and 3 and will display "You're Hot!". Lastly

if the strikes surpass the maximum of strikes, then it'll run the lose method and reassign the pickNewRandom

Sub-Task 100%

Group: Implementation 1
Task #1: Implementation Evidence
Sub Task #2: Add screenshots of the coded solution (ucid/date must be visible)

Task Screenshots

Gallery Style: 2 Columns

4 2 1

```
        // c1286
        // 10/04/24
        if (strikes < maxstrikes){
            int difference = Math.abs(number -
            guess);

            if(difference >= 10){
                System.out.println("You're
                cold!");
            } else if(4 <= difference &&
            difference< 10){
                System.out.println("You're
                Warm!");
            } else if(1 <= difference &&
            difference< 4){
                System.out.println("You're
                Hot!");
            }
        } else{
            lose();
            pickNewRandom = true;
        }
    }
    savestate();
}
```

#4 Coded solution

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

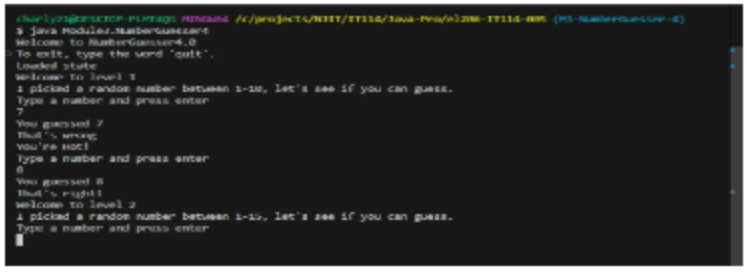
Sub-Task 100%

Group: Implementation 1
Task #1: Implementation Evidence
Sub Task #3: Show implementation working by running the program

Task Screenshots

Gallery Style: 2 Columns

4 2 1



```
$ java Module1.NumberGuesser
Welcome to NumberGuesser4.0
To exit, type the word 'quit'.
Loaded state
Welcome to level 1
I picked a random number between 1-10, let's see if you can guess.
Type a number and press enter
7
You guessed 7
That's wrong
You're Hot!
Type a number and press enter
8
You guessed 8
That's right!
Welcome to level 2
I picked a random number between 1-10, let's see if you can guess.
Type a number and press enter
```

#4 Implementation running

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

End of Task 1

End of Group: Implementation 1

Task Status: 1/1

Group

100%

Group: Implementation 2
Tasks: 1
Points: 4

COLLAPSE

Task

100%

Group: Implementation 2
Task #1: Implementation Evidence
Weight: ~100%
Points: ~4.00

COLLAPSE

Details:
Code screenshots must have ucid/date shown as a comment in the code.

Explanations must be your own words describing the logic and how the solution code solves the problem.

Columns: 1

Sub-Task

100%

Group: Implementation 2
Task #1: Implementation Evidence
Sub Task #1: Mention which option you picked and how you solved it

Task Response Prompt

Explain the logic of how you solved/implemented the chosen option (concrete details). Explain how the code works, don't just paste code snippets

Response:

I chose option 3 where the user would choose the difficulty level and depending on the difficulty, it'll minimize the strikes. I created a method where it asks the user's difficulty level and based on the user's input that's stored in a variable, it'll determine what difficulty it'll be set. For this to work, I had to use scanner to get the user's input and used a swtich case statement to where if the user's input matches with the case such as ("1", "2", or"3"). In this case it's a string in order to function properly. If a something else is inputed other than the difficulty level, then it'll set the default difficulty to medium which is 5 strikes.

Sub-Task

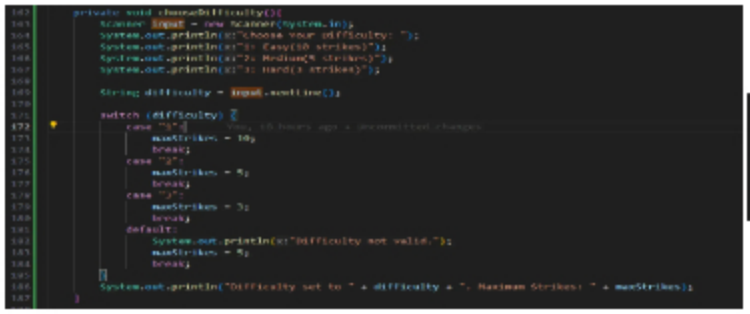
100%

Group: Implementation 2
Task #1: Implementation Evidence
Sub Task #2: Add screenshots of the coded solution (ucid/date must be visible)

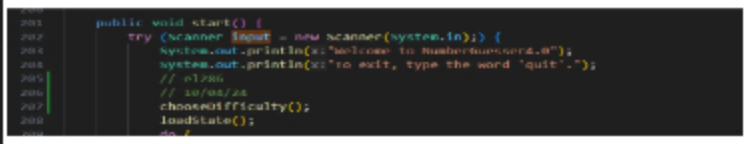
Task Screenshots

Gallery Style: 2 Columns

4 2 1



#3 coded solution



#3 2nd coded solution

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Implementation 2

Task #1: Implementation Evidence

Sub Task #3: Show implementation working by running the program

Task Screenshots

Gallery Style: 2 Columns

4 2 1

#3 Implementation running

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

End of Task 1

End of Group: Implementation 2

Task Status: 1/1

Group

100%

Group: Misc

Tasks: 3

Points: 2

COLLAPSE

Task

100%

Group: Misc
Task #1: Reflection
Weight: ~33%
Points: ~0.67

COLLAPSE

Sub-Task

100%

Group: Misc
Task #1: Reflection
Sub Task #1: Learn anything new? Face any challenges? How did you overcome any issues?

Task Response Prompt

Provide at least a few logical sentences

Response:

I was mainly stuck in implementation #4 where I had to get the number to be in a range where if it hot,cold, or warm. In my mind I thought I had to dome more math than just getting its difference in order to determine the such rannge if it 10 numbers away or some sort. Completely forgot I could use the if (<,>) in order to see if it's less than or more than.

End of Task 1

Task

100%

Group: Misc
Task #2: Pull Request URL
Weight: ~33%
Points: ~0.67

COLLAPSE

Details:
URL should end with /pull/# where the # is the actual pull request number.

Task URLs

URL #1
https://github.com/ElLopez21/el286-IT114-005/pull/5

URL
https://github.com/ElLopez21/el286-IT114-005/p

End of Task 2

Task

100%

Group: Misc
Task #3: Waka Time (or related) Screenshot
Weight: ~33%
Points: ~0.67

COLLAPSE

Checklist

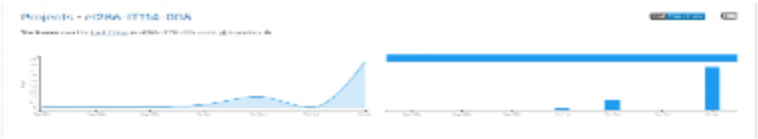
*The checkboxes are for your own tracking

#	Details
☐ #1	Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



waka time

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment