Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-module-3-number-guesser-4/grade/rra23

Course: IT114-003-F2024
Assigment: [IT114] Module 3 Number Guesser 4
Student: Rahid A. (rra23)

Submissions:

Submission Selection

1 Submission [submitted] 9/30/2024 6:39:42 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

Group: Implementation 1
Tasks: 1
Points: 4

COLLAPSE

Task

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

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:

Option 1: Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level)

I created a method called higherOrLower. This method takes in the guess and if the guess is less than the number, it returns "Higher" and if the guess is greater than the number it returns "Lower". If the guess is equal to the number it returns an empty string. After writing this method, I wrote an if statement in the processGuess method. The if statement checks if the current strikes is not equal to the max strikes, and if it is not then it calls the higherOrLower

method. I chose to check if it is not equal to max strikes because if it is equal that means that the level gets rolled back and it should not run.

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

```
            System.out.println("That's wrong");
            strikes++;

            if (strikes != maxStrikes){
                System.out.println(higherOrLower(guess)); //rezik 9-30-24
            }
        You, 1 second ago • Uncommitted changes
            if (strikes >= maxStrikes) {
                lose();
                pickNewRandom = true;
            }
        }
        saveState();
    }

    private String higherOrLower(int guess){
        if (guess < number){
            return "Higher";
        }
        else if (guess > number){
            return "Lower";
        }
        else return "";
    }
```

higherOrLower code

Caption(s) (required) ✓

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

Sub-Task

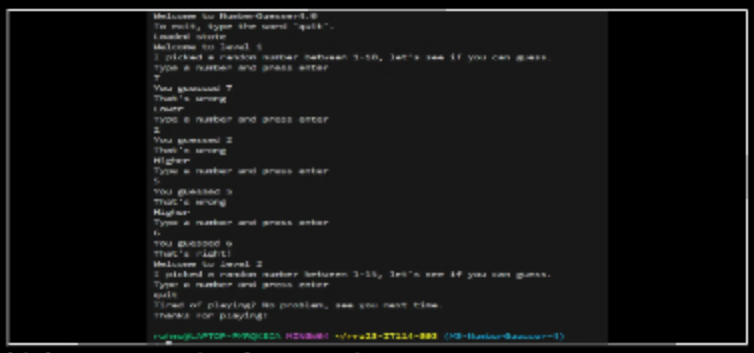
100%

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



```
Welcome to Number-Guesser-1.0
To exit, type the word "quit".
Loading 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
Lower
Type a number and press enter
2
You guessed 2
That's wrong
Higher
Type a number and press enter
5
You guessed 5
That's wrong
Higher
Type a number and press enter
6
You guessed 6
That's right!
Welcome to level 2
I picked a random number between 1-15, let's see if you can guess.
Type a number and press enter
quit
Tired of playing? No problem, see you next time.
Thanks for playing!
```

higherOrLower implementation

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:

Option 3: Add a difficulty selector that adjusts the max strikes per level (i.e., "easy" 10 strikes, "medium" 5 strikes, "hard" 3 strikes)

I created a method called difficulty that takes in a string. The method uses if statements to check whether the user entered easy, medium, or hard. Depending on what the user entered, the maximum number of strikes per level is changed to match the difficulty. I got the user input in the start method before the game starts using the Scanner object.

Sub-Task

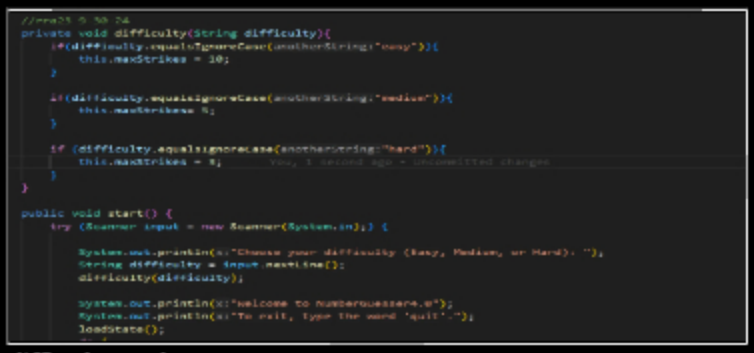
100%

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



difficulty code

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

difficulty implementation

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:

A challenge that I faced was determining where to call the higherOrLower method. Since the question said not to display it if it is on a guess that rolls back the level.I was having a hard time thinking of how to make sure that it does not get called if the level gets changed. Eventually, I used an if statement to check if the current strikes was equal to the max strikes, which determined if the method is called or not.

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/Rahid-Ahmed/rra23-IT114-003/pull/5

URL
https://github.com/Rahid-Ahmed/rra23-IT114-003

End of Task 2

Task

100%

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

Points: ~0.67

COLLAPSE

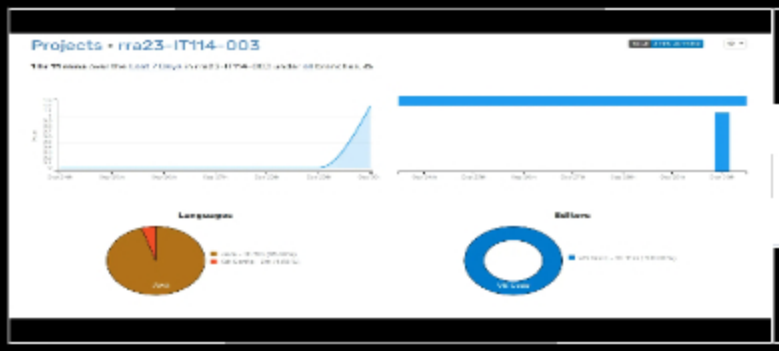
Checklist *The checkboxes are for your own tracking

#	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



Wakatime

Wakatime

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment