Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-module-3-number-guesser-4/grade/yh68

IT114-450-M2024 - [IT114] Module 3 Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 6/9/2024 7:14:49 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

Tasks: 5 Points: 10.00

Implementation 1 (4 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Implementation Evidence

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.

#1) Mention which option you picked and how you solved it

Explanation (required) ✓

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

PREVIEW RESPONSE

I chose the first option "Display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level)". I implemented this using an else statement where it checks **if** the answer of the user was less than the number, **else** it is greater than it. After it checks whether it is greater or less than it, it prints the hint higher or lower.

#2) Add screenshots of the coded solution (ucid/date must be visible)

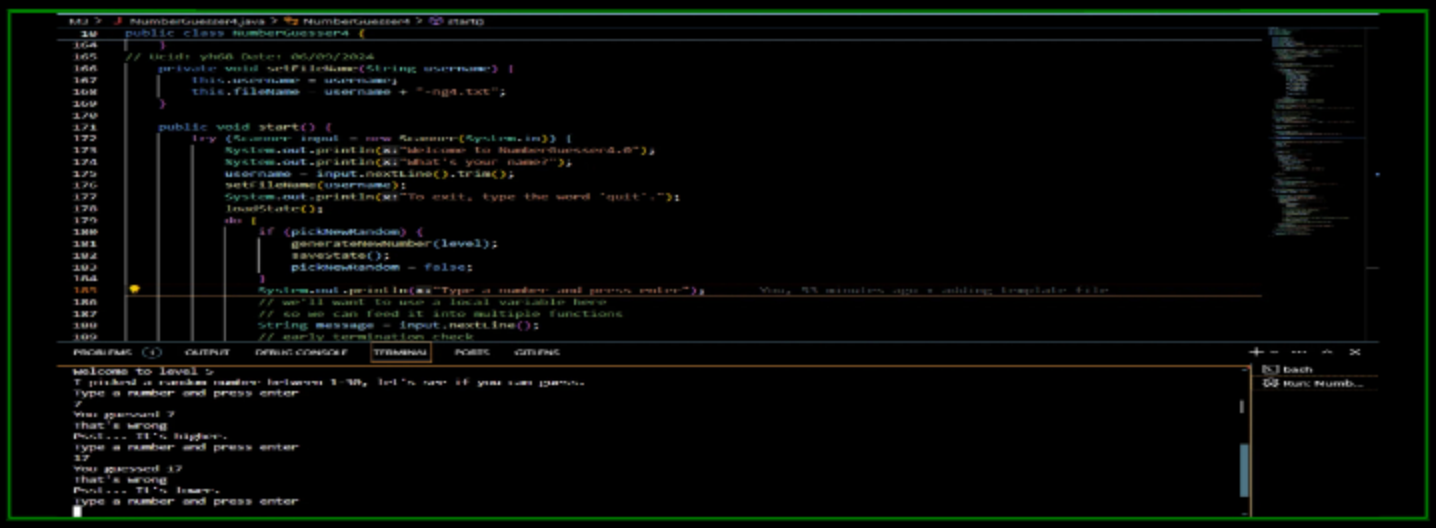
```
            win();
            pickNewRandom = true;
        } else {
            System.out.println(x:"That's wrong");
            strikes++;
            if (strikes >= maxStrikes) {
                lose();
                pickNewRandom = true;
            }
            // Ucid: yhG8 Date: 06/09/2024
            else {
                if (guess < number) {
                    System.out.println(x:"Psst... It's higher.");
                } else {
                    System.out.println(x:"Psst... It's lower.");
                }
            }
        }
        saveState();
    }
```

Caption (required) ✓

Describe/highlight what's being shown

Showing the implementation of the higher or lower hint.

#3) Show implementation working by running the program



Caption (required) ✓

Describe/highlight what's being shown

Showing the successful implementation of this prompt when running the code.

Implementation 2 (4 pts.)

^COLLAPSE^

^COLLAPSE^

Task #1 - Points: 1

Text: Implementation Evidence

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.

#1) Mention which option you picked and how you solved it

Explanation (required) ✓

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

PREVIEW RESPONSE

I chose the last option for my second choice, "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)". I did this by first making an empty string variable called username. I then made the function setFileName() which takes the string variable username as a parameter, inside the function it simply sets the username to the one being inputted by the user and changes the filename to the username being place before the name of the file. In the start function the setFileName() function is called after prompting the user to input their name and assigning the name the input to the username and subsequently the filename. I implemented this method using the Scanner variable that was already declared in the start function, "input".

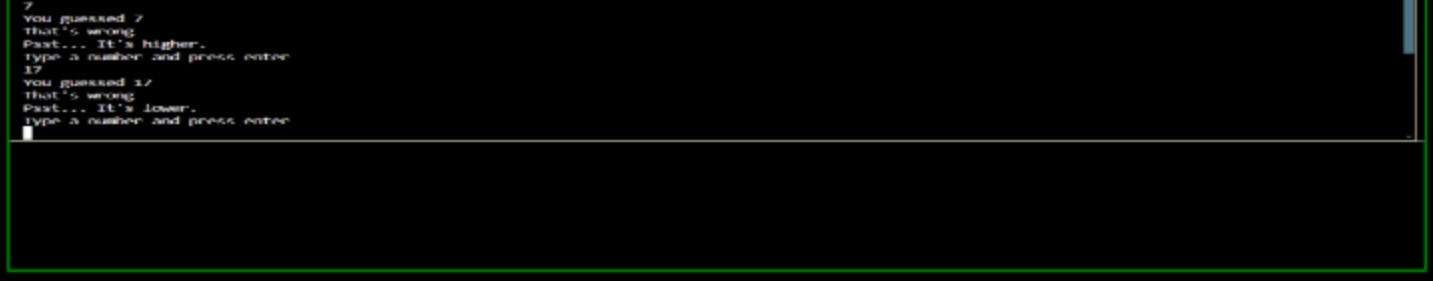
#2) Add screenshots of the coded solution (ucid/date must be visible)

Caption (required) ✓

Describe/highlight what's being shown

Showing the implementation of the last prompt in which we are to implement separate save files.

#3) Show implementation working by running the program



Caption (required) ✓

Describe/highlight what's being shown

Showing the successful implementation of the prompt.

Misc (2 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Reflection

#1) Learn anything new? Face any challenges? How did you overcome any issues?

Explanation (required) ✓

Provide at least a few logical sentences

PREVIEW RESPONSE

I got to brush up on some skills and I learned things about java file handling in which I did not know before. No issues, I've been getting accustomed to gitbash and the professors videos are a great tool to help me understand whatever is not too clear from the prompts.

COLLAPSE

Task #2 - Points: 1

Text: Pull Request URL

Details:

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

URL #1

https://github.com/FreePalestine7/yh68-it114-450/pull/10

COLLAPSE

Task #3 - Points: 1

Text: Waka Time (or related) Screenshot

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	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: Large View

Small Medium Large

Showing overview and individual file times.

Checklist Items (1)

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

End of Assignment