Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-451-M2024/it114-module-3-number-guesser-4/grade/st278

IT114-451-M2024 - [IT114] Module 3 Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 6/11/2024 2:44:00 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

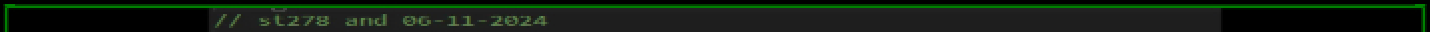
Explanation (required) ✓

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

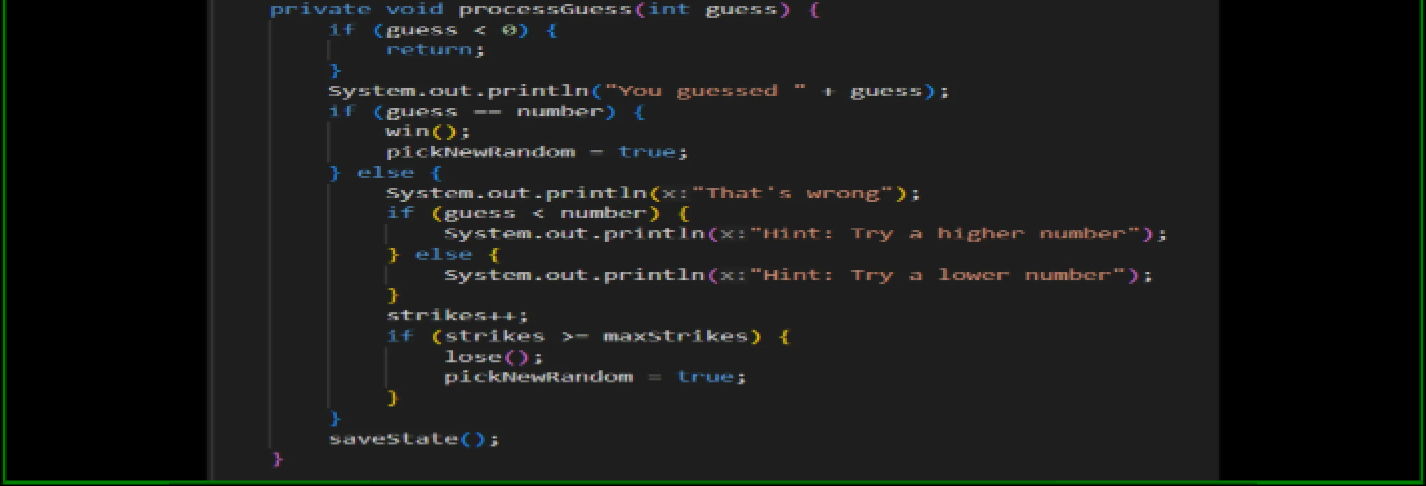
PREVIEW RESPONSE

The option I picked for my first implementation is 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).
The player's guess is compared with the target number (number).
If the guess is correct (guess == number), the player wins, advances to the next level, and the strikes counter is reset.
If the guess is incorrect (guess != number), the game provides feedback and a hint:
If the guess is lower than the target number, the hint "Try a higher number" is displayed.
If the guess is higher than the target number, the hint "Try a lower number" is displayed.

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



```
// st278 and 06-11-2024
```



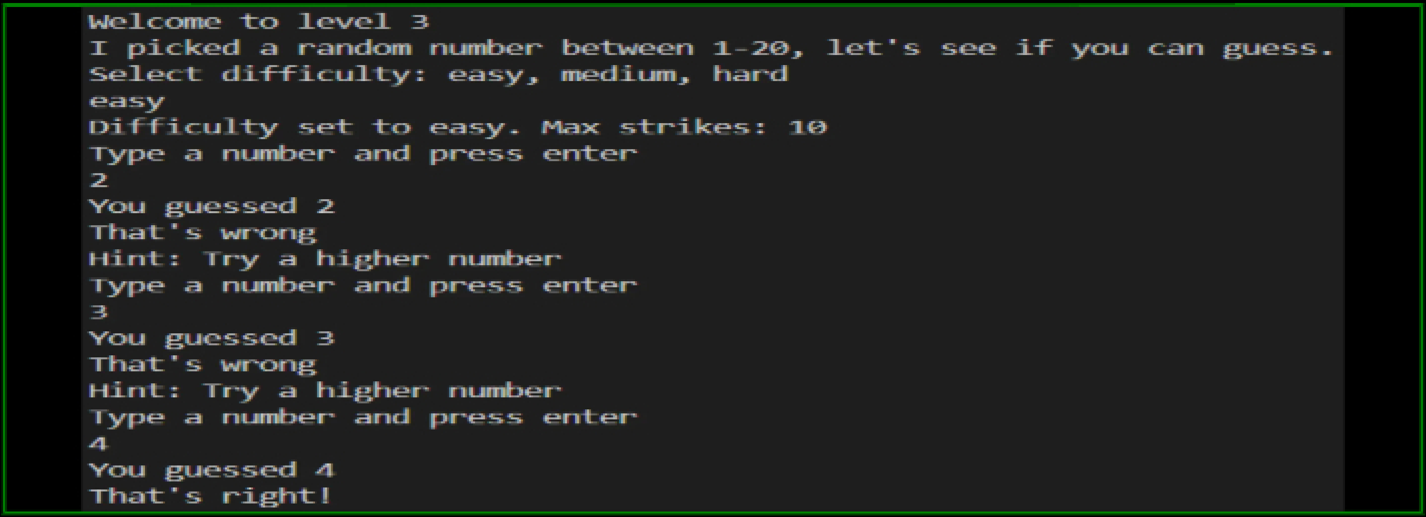
```
private void processGuess(int guess) {
    if (guess < 0) {
        return;
    }
    System.out.println("You guessed " + guess);
    if (guess == number) {
        win();
        pickNewRandom = true;
    } else {
        System.out.println(x:"That's wrong");
        if (guess < number) {
            System.out.println(x:"Hint: Try a higher number");
        } else {
            System.out.println(x:"Hint: Try a lower number");
        }
        strikes++;
        if (strikes >= maxStrikes) {
            lose();
            pickNewRandom = true;
        }
    }
    saveState();
}
```

Caption (required) ✓

Describe/highlight what's being shown

The coded solution is shown.

#3) Show implementation working by running the program



```
Welcome to level 3
I picked a random number between 1-20, let's see if you can guess.
Select difficulty: easy, medium, hard
easy
Difficulty set to easy. Max strikes: 10
Type a number and press enter
2
You guessed 2
That's wrong
Hint: Try a higher number
Type a number and press enter
3
You guessed 3
That's wrong
Hint: Try a higher number
Type a number and press enter
4
You guessed 4
That's right!
```

Caption (required) ✓

Describe/highlight what's being shown

The output is shown.

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

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

The option I picked for my second implementation is option 3: Add a difficulty selector that adjusts the max strikes per level (i.e., "easy" 10 strikes, "medium" 5 strikes, "hard" 3 strikes).

The player is prompted to select a difficulty level: "easy", "medium", or "hard".

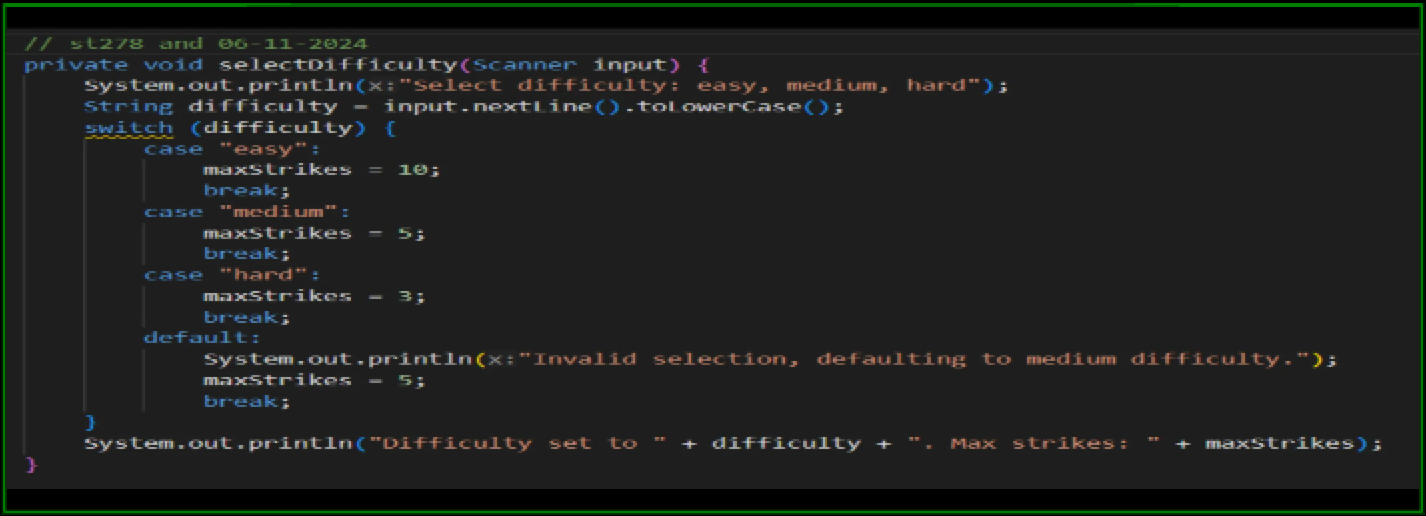
Based on the player's input, the maxStrikes variable is set:

"easy" -> 10 strikes

"medium" -> 5 strikes

"hard" -> 3 strikes

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



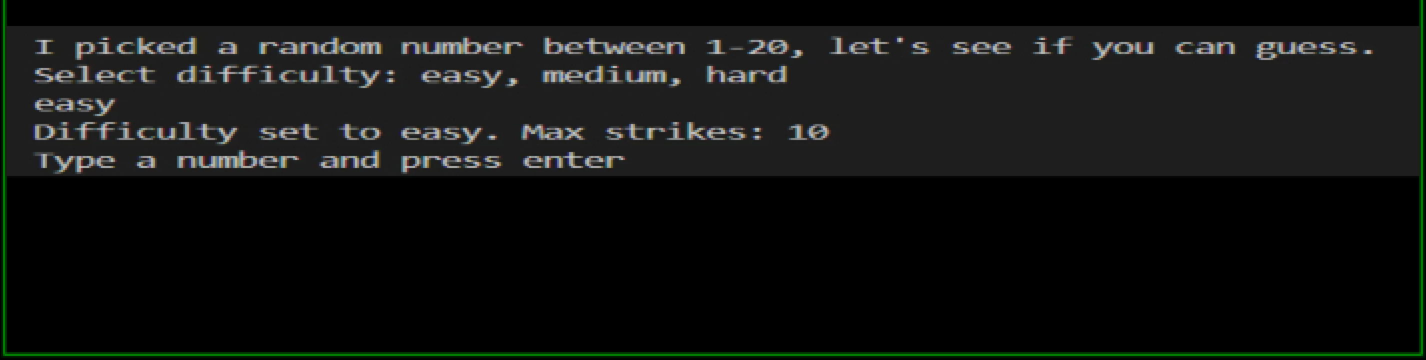
```java
// st278 and 06-11-2024
private void selectDifficulty(Scanner input) {
    System.out.println(x:"Select difficulty: easy, medium, hard");
    String difficulty = input.nextLine().toLowerCase();
    switch (difficulty) {
        case "easy":
            maxStrikes = 10;
            break;
        case "medium":
            maxStrikes = 5;
            break;
        case "hard":
            maxStrikes = 3;
            break;
        default:
            System.out.println(x:"Invalid selection, defaulting to medium difficulty.");
            maxStrikes = 5;
            break;
    }
    System.out.println("Difficulty set to " + difficulty + ". Max strikes: " + maxStrikes);
}
```

Caption (required) ✓

Describe/highlight what's being shown

The coded solution.

#3) Show implementation working by running the program



```
I picked a random number between 1-20, let's see if you can guess.
Select difficulty: easy, medium, hard
easy
Difficulty set to easy. Max strikes: 10
Type a number and press enter
```

Caption (required) ✓

Describe/highlight what's being shown

The output works well.

● 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

The challenge I faced was handling the NoSuchElementException error, which occurs when the Scanner tries to read input that isn't available, was addressed by ensuring the Scanner checks for input availability using hasNextLine() before attempting to read with nextLine(). This approach effectively prevents the NoSuchElementException from being thrown, ensuring smooth and error-free user input handling.

● ^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/st278/st278-IT114-M2024/pull/5

●
^COLLAPSE ^

Task #3 - Points: 1

Text: Waka Time (or related) Screenshot

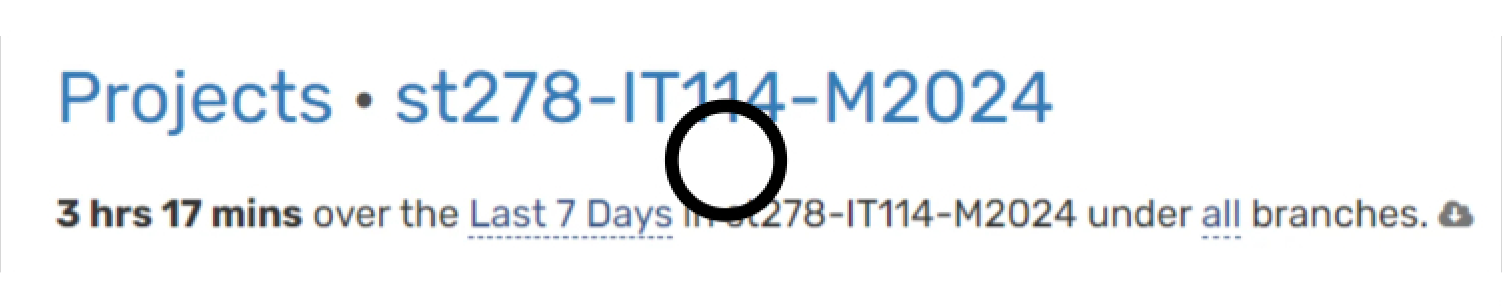
Checklist *The checkboxes are for your own tracking

#	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

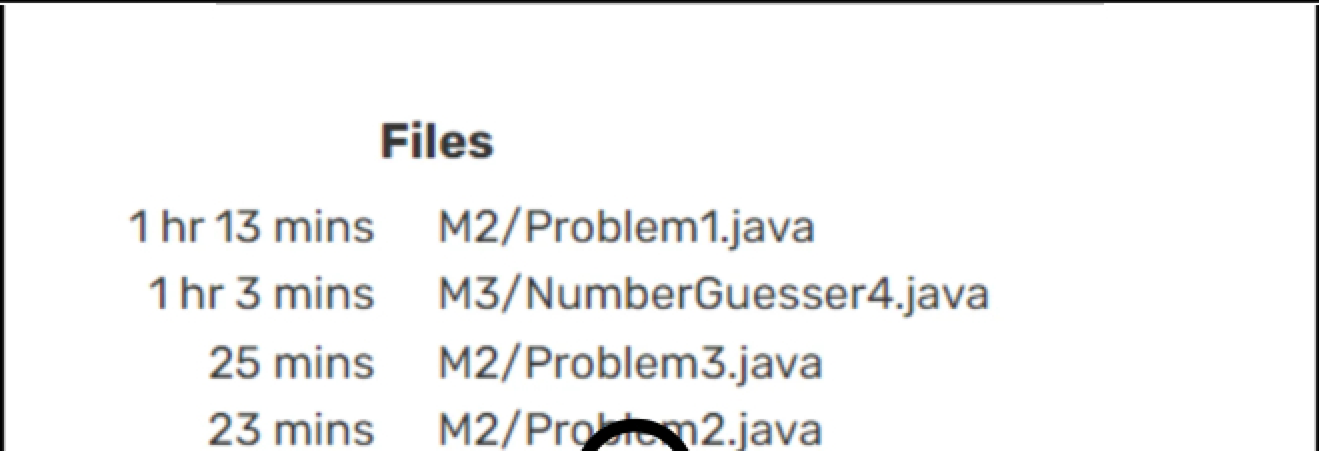


Projects • st278-IT114-M2024

3 hrs 17 mins over the Last 7 Days in st278-IT114-M2024 under all branches.

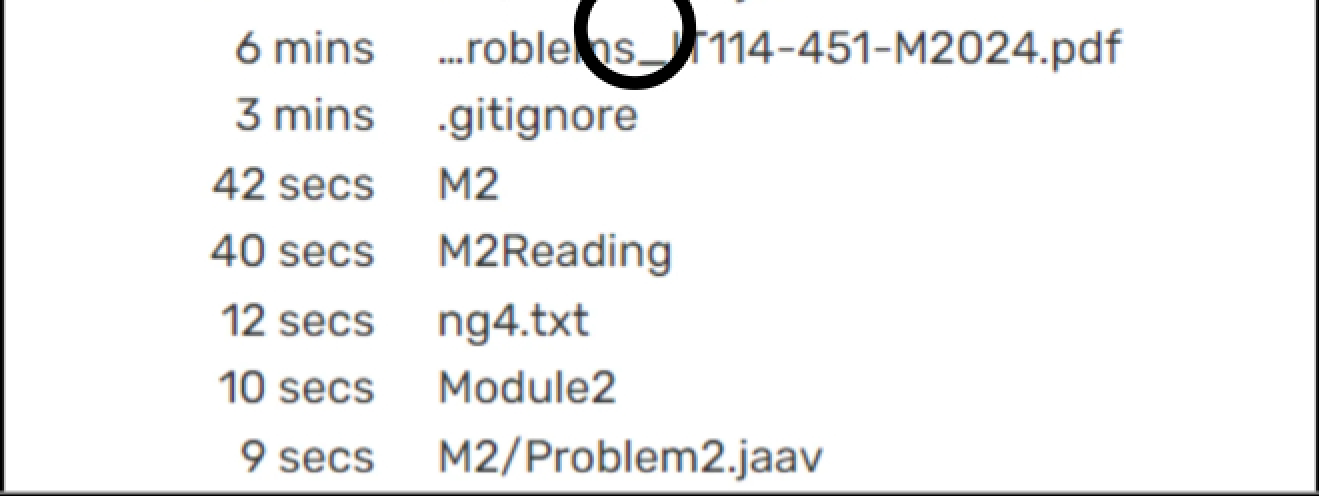
Overview of the wakatime utilized.

Checklist Items (0)



Files

1 hr 13 mins	M2/Problem1.java
1 hr 3 mins	M3/NumberGuesser4.java
25 mins	M2/Problem3.java
23 mins	M2/Problem2.java



6 mins	…roblems_IT114-451-M2024.pdf
3 mins	.gitignore
42 secs	M2
40 secs	M2Reading
12 secs	ng4.txt
10 secs	Module2
9 secs	M2/Problem2.jaav

Individual files.

Checklist Items (0)

End of Assignment