Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-005-F2024/it114-module-3-number-guesser-4/grade/kh465

Course: IT114-005-F2024
Assigment: [IT114] Module 3 Number Guesser 4
Student: Keven H. (kh465)

Submissions:

Submission Selection

1 Submission [submitted] 9/29/2024 2:39:12 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:

Option 1: To add a hint displaying if the user needs to guess higher or lower, I rewrote and added code to the method processGuess. To add the hints, I added two else if blocks to the method to check if the number that the user chose is lower or higher than the number generated. Depending on the number guessed, a System.out.println is used to display "higher" if the user guessed too low and "lower" if the user guessed too high, with strikes still being added regardless of the hint.

Sub-Task

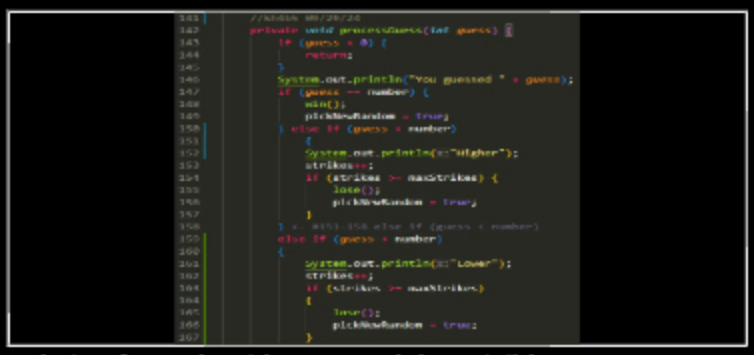
100%

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



```
    //bfkev 09/20/24
    private void processGuess(int guess) {
        if (guess < 0) {
            return;
        }
        System.out.println("You guessed " + guess);
        if (guess == number) {
            win();
            pickNewRandom = true;
        } else if (guess < number)
            {
            System.out.println("Higher");
            strikes++;
            if (strikes >= maxStrikes) {
                lose();
                pickNewRandom = true;
            }
        } else if (guess > number)
        {
            System.out.println("Lower");
            strikes++;
            if (strikes >= maxStrikes)
            {
                lose();
                pickNewRandom = true;
            }
```

Solution for code with UCID and date visible.

Caption(s) (required) ✓

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

Sub-Task

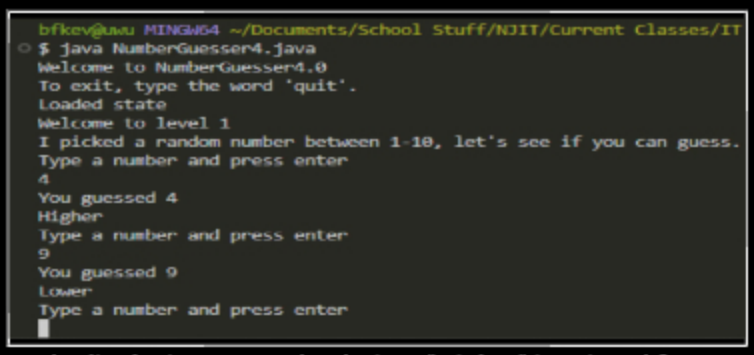
100%

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



```
bfkev@uwu MINGW64 ~/Documents/School Stuff/NJIT/Current Classes/IT
$ java NumberGuesser4.java
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
4
You guessed 4
Higher
Type a number and press enter
9
You guessed 9
Lower
Type a number and press enter
```

Code displaying created solution. "Higher" is printed for a low guess, "Lower" is printed for a high guess.

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: To add a difficulty selection to each level, a new method called difficulty was added. This is a void method with an if statement checking if strikes equals zero (this assumes a new level has started). If this is true, a Scanner is created, a System.out.println prompts the user for their choice in difficulty, and sets the private int maxStrikes to a predefined amount based on the user choice captured by the Scanner. This method is called in the method start() in the first if block so it prompts the user when a new level is started. This can be modified to only run at the beginning of the game (level 1) by adding an AND statement to the if block in difficulty (if (level == 1 && strikes == 0)). This is originally how I had the code but I was unsure of what option 3 wanted so I included this just in case

Sub-Task

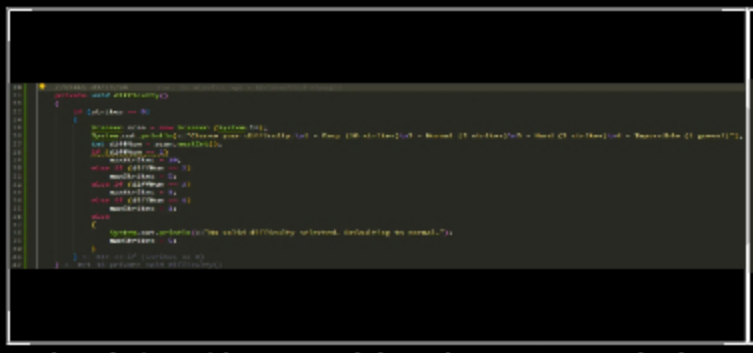
100%

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



Code solution with UCID and date shown. New method difficulty()

Method difficulty() called in method start(). UCID and date are also included.

Caption(s) (required) ✓

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

Sub-Task

100%

Group: Implementation 2

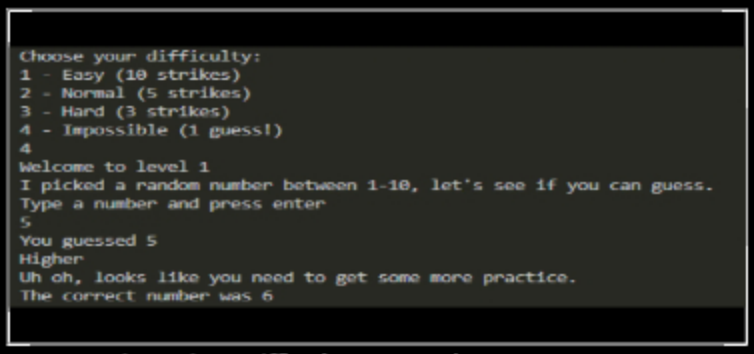
Task #1: Implementation Evidence

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



Program in action. Difficulty 4 was chosen, 1 guess was attempted, which loses since maxStrikes has been set to 1.

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:

Only one minor issue was faced while developing a solution to this problem, but this was an issue with myself rather than the program. I originally tried to overwrite maxStrikes in my difficulty() method using "==" instead of "=" and could not figure out why it was not working. It was when I actually looked closely at my method that I realised I was using an equality operator "==" instead of an assignment operator "=". The rest of the program was fairly straightforward.

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/kh465/kh465-IT114-005/pull/8

URL
https://github.com/kh465/kh465-IT114-005/pull/

End of Task 2

Task

Group: Misc
Task #3: Waka Time (or related) Screenshot

100%

Weight: ~33%
Points: ~0.67

COLLAPSE

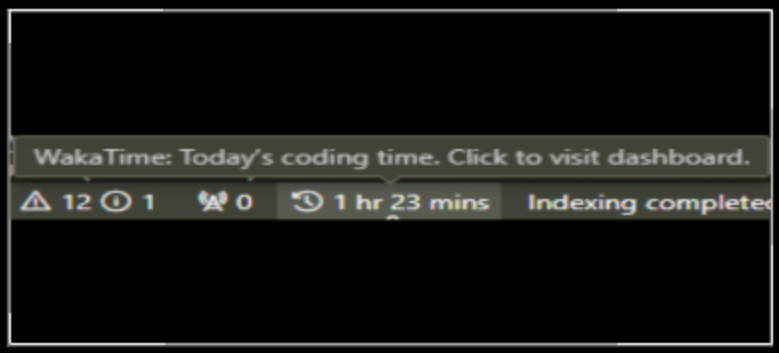
Checklist *The checkboxes are for your own tracking

#	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



Time spent on problem according to WakaTime.

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment