Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-number-guesser-4/grade/ms75

IT114-006-S2024 - [IT114] Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 2/12/2024 1:23:15 PM

Instructions

COLLAPSE

1. Create the below branch name
2. Implement the NumberGuess4 example from the lesson/slides
    1. https://gist.github.com/MattToegel/aced06400c812f13ad030db9518b399f
3. Add/commit the files as-is from the lesson material (this is the base template). You may want to push this commit so you can open the pull request and keep it open.
4. 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)
5. Fill in the below deliverables
6. Save changes and export PDF
7. Git add/commit/push your changes to the HW branch
8. Create a pull request to main
9. Complete the pull request (don't forget to locally checkout main and pull changes to prep for future work)
10. Upload the same PDF to Canvas

Branch name: M3-NumberGuesser-4

Tasks: 7 Points: 10.00

COLLAPSE

Implementation 1 (4 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Chosen Option and Details

Checklist — *The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention which option you picked
#2	1	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:

My first option I chose was to add a higher or lower text after each non level ending level. The first line of code checks which try of the level the user is at by comparing the variable strikes which keeps track of how many tries the user got wrong and the maxStrikes variable which is the level ending try and subtracts it by 1. Now that the program knows the try the user is on it compares the valuable guess which is entered by the user to the number generated by the number generator. If the guess is less than the number the system prints higher telling the user to guess higher, and the else is the opposite so if it is too high it will print telling the user to guess lower.

```
//Added an if statement to compare the value of the variable guess and the variable number to display highe
            if (strikes < maxStrikes - 1) {
                if (guess < number) {
                    System.out.println("Oops a little higher!\n");
                } else {
                    System.out.println("Oops a little lower!\n");
                }
```

COLLAPSE

Task #2 - Points: 1

Text: 2+ Screenshots of code and demo

Checklist — *The checkboxes are for your own tracking

#	Points	Details
#1	1	Show implementation working by running the program
#2	1	Clearly caption the screenshot of what you're showing
#3	1	The code screenshot(s) clearly show the code specific to the feature
#4	1	A comment with the UCID/date is visible near the code change(s)

Task Screenshots:

Large Gallery

Checklist Items (0)

This screenshot is showing my code on level 1 of the number guesser and at first i guessed 1 and the program said to go higher, so i guessed 10 and the program said lower.

COLLAPSE

Implementation 2 (4 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Chosen Option and Details

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Mention which option you picked
#2	1	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:

For this option I chose the fourth option of adding a choose difficulty selector for the user. I did this by first changing the variable initializer on the top of the given code
from: `private int maxStrikes = 5;` to: `private int maxStrikes;` to have no assigned value for the variable maxStrikes. I then created the method setDifficulty() which first asks the user to type out their desired difficulty level. Then i created a scanner object called scanner and in the following line i scanned the users input and used the .toLowerCase method to make all the letters lowercase in case the user typed them in caps. I then created a switch case which checked if the cases were "easy", "medium" or "hard". Depending on if the users entry matched the case or not the maxStrikes variable was set accordingly to which case it matched. I also added a default case if the user types something that doesn't match the cases and the default case sets the difficulty as medium and sets the maxStrikes to 5. The default case also prints out a message letting the user know the difficulty has been set to medium.

```
//setDifficulty method created to have user type in the difficulty level
//I also added a default case if the user types something wrong
    private void setDifficulty() {
        System.out.println("Choose a difficulty: (easy, medium, hard)");
        Scanner scan = new Scanner(System.in);
        String difficulty = scan.nextLine().toLowerCase();
```

```
        switch (difficulty) {
            case "easy":
                maxStrikes = 10;
                break;
            case "medium":
                maxStrikes = 5;
                break;
            case "hard":
                maxStrikes = 3;
                break;
            default:
                System.out.println("Invalid entry, defaulting level set to medium.");
                maxStrikes = 5;
                break;
        }
    }
```

Finally I just needed to call the new method I just created in the start method of the game by using the following line of code:

```
            setDifficulty(); // I added the method I created earlier to initiate in the start of the game
```

COLLAPSE

Task #2 - Points: 1

Text: 2+ Screenshots of code and demo

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show implementation working by running the program
#2	1	Clearly caption the screenshot of what you're showing
#3	1	The code screenshot(s) clearly show the code specific to the feature
#4	1	A comment with the UCID/date is visible near the code change(s)

Task Screenshots:

Large Gallery

Checklist Items (0)

In this screenshot I ran the program and when asked for difficulty I typed qwerty. Since qwerty is not a valid case the switch case chose the default case and set it to

medium. I then quit the game and restarted and it asked for difficulty again and I put hard and then after 3 wrong guesses it restarted the level. I tested the code for all 3 cases.

COLLAPSE

Misc (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Reflection

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	Example prompts: Learn anything new? Face any challenges? How did you overcome and issues?
#2	1	At least a few logical sentences related to the assignment.

Response:

This assignment was not too bad. I struggled on the switch case statement for a little as I forgot how to set it up. I was trying to configure the setDifficulty method so a user can choose a difficulty every time they advance a level or change the difficulty mid game without having to quit but it seemed a little too complicated.

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/mikes1302/MS75-IT114-006/pull/5

COLLAPSE

Task #3 - Points: 1

Text: Waka Time (or related) Screenshot

Checklist

*The checkboxes are for your own tracking

#	Points	Details
	1	Screenshot clearly shows what files/project were being worked on (the duration of time doesn't

#1		correlated with the grade for this item)

Task Screenshots:

Large Gallery

Checklist Items (1)

#1 Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated