Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-number-guesser-4/grade/mth39

IT114-002-S2024 - [IT114] Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 2/10/2024 12:07:32 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)
10Upload 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:

The option I picked was to display higher or lower as a hint after a wrong guess (only after a wrong guess that doesn't roll back the level). 'if (guess < 0) ' checks if the guess is less than 0. If it is, it returns early, assuming that negative guesses are invalid. It then prints out the guess made by the user. 'if (guess == number) {win();pickNewRandom = true;' checks if the guess matches the randomly generated number (number). If it does, it calls the win() method, indicating that the user guessed correctly, and sets pickNewRandom to true, meaning a new random number should be generated for the next round. 'if (guess > number){hint = "Select higher value";}' checks if the guess is greater than the random number. If so, it assigns the hint "Select lower value" to the hint variable. 'else if (guess < number){hint = "Select higher value";}' assigns the hint "Select higher value" to the hint variable. 'System.out.println("That's wrong. Hint: " +hint+")");strikes++;' prints out a message indicating that the guess is incorrect along with the hint, if any and increments the strikes counter. ' if (strikes >= maxStrikes) {lose();pickNewRandom = true;' checks if the number of strikes exceeds or equals the maximum allowed strikes (maxStrikes). If it does, it calls the lose() method, indicating that the user has lost the game, and sets pickNewRandom to true to generate a new random number for the next round.

●

^ 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:

Task Screenshots:

Large Gallery

Checklist Items (0)

Shows completed option of offering higher/lower hint code to users

Checklist Items (0)

Shows terminal output of code of higher/lower hint code

Implementation 2 (4 pts.)

COLLAPSE

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:

The option I picked was to add a difficulty selector that adjusts the max strikes per level. If increase is true, indicating that the user wants to increase the difficulty level, it checks if the current maximum strikes allowed (maxStrikes) is greater than 1. If it is, it decrements maxStrikes by 1. If the maximum strikes are successfully decreased, it prints a message indicating that the difficulty level has increased, along with the updated maximum strikes allowed. If increase is false, indicating that the user wants to decrease the difficulty level, it checks if the current maximum strikes allowed (maxStrikes) is less than 5. If it is, it increments maxStrikes by 1. If the maximum strikes are successfully increased, it prints a message indicating that the difficulty level has decreased, along with the updated maximum strikes allowed.

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

#	Points	Details
☐ #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)

Checklist Items (0)

Shows completed code for changing difficulty option	Shows terminal output of changing difficulty code

Misc (2 pts.)

COLLAPSE

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:

Missing Response

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

Missing URL

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:

Large Gallery

Checklist Items (0)

Checklist Items (0)