Submission Worksheet

CLICK TO GRADE

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

Submissions:

Submission Selection

1 Submission [active] 2/12/2024 5:37:57 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

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:

I added the higher lower function here. I put it in the add strikes part thinking it would only go off when there are strikes still left to be added (i was wrong). It just compares if the guess is less than the the random number, if it is it says higher. If it isn't it goes tot he else and prints lower.

^ 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)

Checklist Items (0)

added the higher lower mechanic after each guess

code running in git bash

Implementation 2 (4 pts.)

^ COLLAPSE ^

Task #1 - Points: 1
Text: Chosen Option and Details

^ COLLAPSE ^

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:

I did the hot cold add on here. At first I was going to make it a switch case but when thinking about it I decided and else if block would be easier. First it checks the absolute value of the the guess minus the number in each part of the block. If the result isn't equal to or greater than 10 it goes to the next else if followed by it checking if its greeter than or equal to 5 and then next block it compares the outcome to 2. if it goes through the whole thing then you are only 1 away and tell you you are on fire (very close).
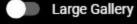
Task #2 - Points: 1
Text: 2+ Screenshots of code and demo

^ COLLAPSE ^

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)   🗑 Checklist Items (0)

● **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:

I had some issues having the code that I added so close to eachother. At first I tried to build them off each other but I had issues. I also stumbled into an issue when the Hot cold one was after the higher lower part but putting it before helped. I also sat there for a while trying to make the higher lower part a switch statement but decided on just making it an else if statement. Also i confused myself and had the absolute values of the guesses calculated not the absolute value of the guess minus the number and that was messing me up in testing.

● ∧ 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/Jomarrrrr/Oha2-it114-006/pull/3

● ∧ 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)