Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-module-3-number-guesser-4/grade/rev

Course: IT114-003-F2024
Assigment: [IT114] Module 3 Number Guesser 4
Student: Ricardo V. (rev)

Submissions:

Submission Selection

1 Submission [submitted] 9/28/2024 6:43:56 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:

The first option is to display higher or lower as a hint after a wrong guess, The logic is added in the "processGuess()" method after checking if the guess was incorrect but before determining if the player has lost. After a wrong guess, if the strikes are still less than the maximum, the game now tells the player whether the correct number is higher or lower than their guess.

Sub-Task

Group: Implementation 1

100%

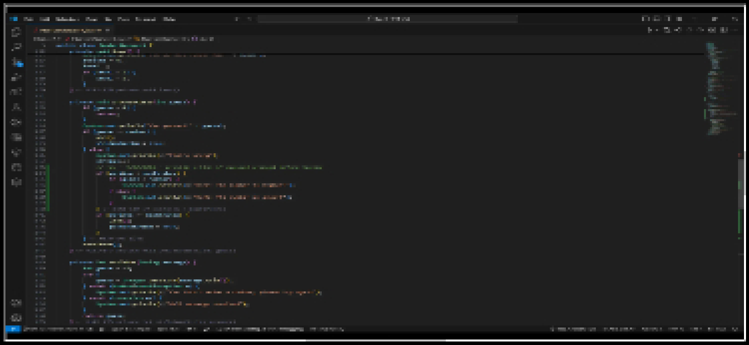
Task #1: Implementation Evidence

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



screenshots of the coded solution

Caption(s) (required) ✓

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

Sub-Task

100%

Group: Implementation 1

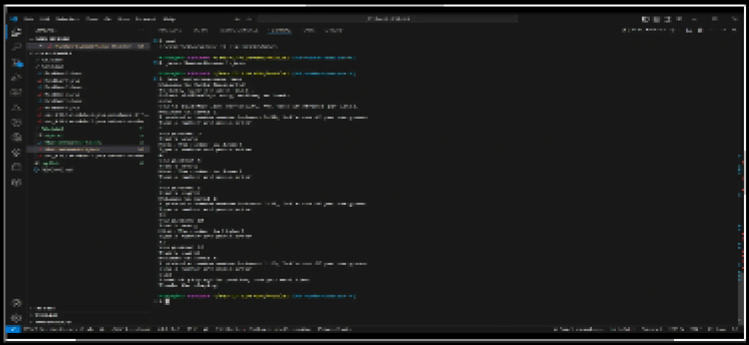
Task #1: Implementation Evidence

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



implementation working

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

Group: Implementation 2

100%

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:

The third option to add a difficulty selector that adjusts the max strikes per level. A new method, "selectDifficulty()", lets the player choose a difficulty level: "easy", "medium", or "hard", adjusting the maximum number of strikes allowed, with a default of 5 strikes for medium difficulty.

Sub-Task

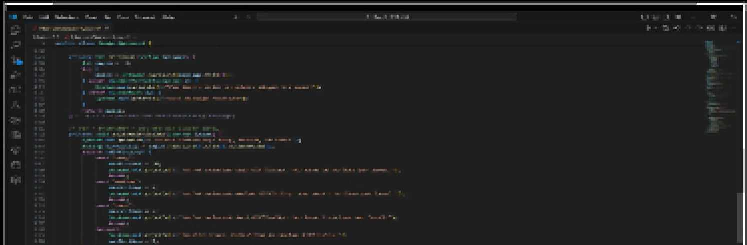
100%

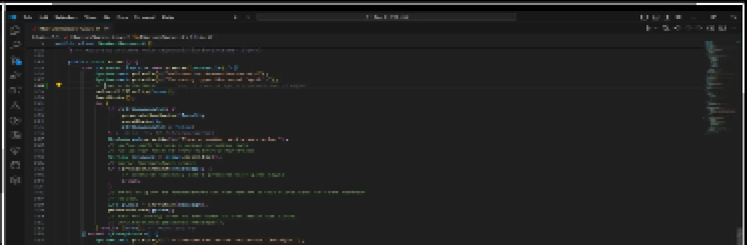
Group: Implementation 2
Task #1: Implementation Evidence
Sub Task #2: Add screenshots of the coded solution (ucid/date must be visible)

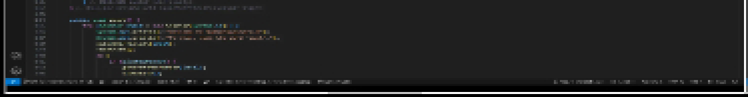
Task Screenshots

Gallery Style: 2 Columns

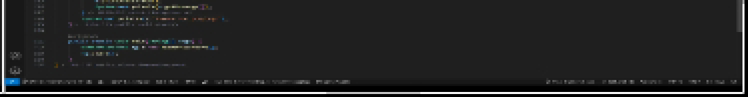
4 2 1







screenshots of the coded solution



coded solution

Caption(s) (required) ✓

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

Sub-Task

100%

Group: Implementation 2

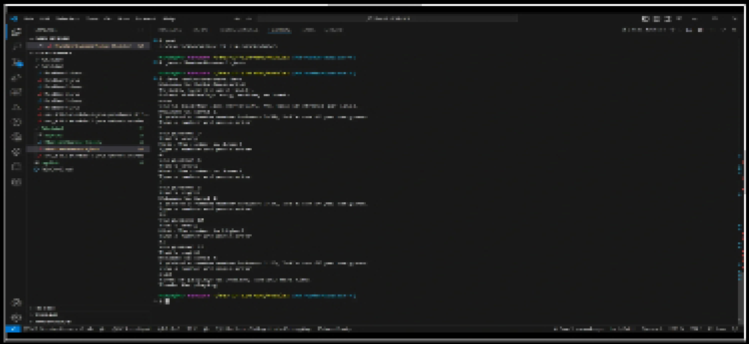
Task #1: Implementation Evidence

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

Task Screenshots

Gallery Style: 2 Columns

4 2 1



implementation working

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

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:

Setting new difficulties took me by surprise. I almost got stuck until a realize I forgot to add another line of code in the end. A silly mistake.

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/RicardoVas9991/Rev-IT-114-003/pull/7

URL
https://github.com/RicardoVas9991/Rev-IT-114-0

End of Task 2

Task

100%

Group: Misc
Task #3: Waka Time (or related) Screenshot
Weight: ~33%
Points: ~0.67

COLLAPSE

Checklist

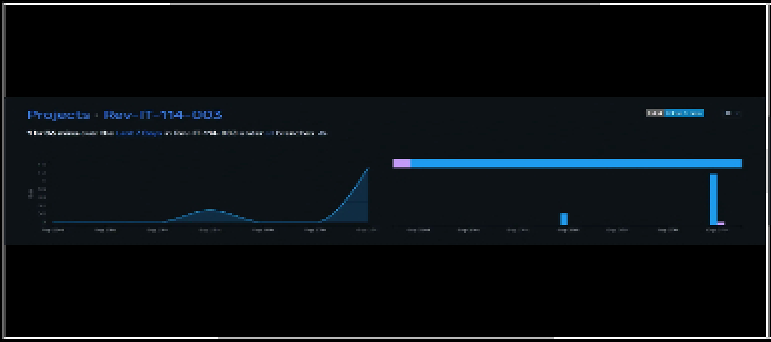
*The checkboxes are for your own tracking

#	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



Waka Time

Waka Time

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment