

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-number-guesser-4/grade/msa224>

IT114-004-S2024 - [IT114] Number Guesser 4

Submissions:

Submission Selection

1 Submission [active] 2/13/2024 12:50:16 AM

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 ^

Task #1 - Points: 1

Text: Chosen Option and Details

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Mention which option you picked
<input type="checkbox"/> #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 chose the "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)" option.

What I did to achieve this was firstly I added the system print ln with whats your name to ask the user for their name. The next line is used to just read the users name input. Lastly we make a file for whatever the user name is, so for example if they add Miyan as their name it will be Miyan.txt and that will be the file pulled up when being saved or created.

Task #2 - Points: 1

Text: 2+ Screenshots of code and demo

Checklist

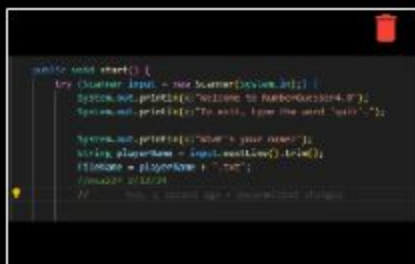
*The checkboxes are for your own tracking

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

Task Screenshots:



Large Gallery



Checklist Items (1)

#3 The code screenshot(s) clearly show the code specific to the feature



Checklist Items (1)

#1 Show implementation working by running the program

shows the newly added code

shows the ran code output with my name being loaded from a saved file and loading my level 2.

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 chose the "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" option.

To achieve this firstly I had to figure out a way to get the difference calculated so we know when to display which message, to do this we used the int difference line to find the difference by subtracting the guess from the actual number.

Then we had to figure which message to be sent and when, I edited it to do cold if 6 or more and warm when 3 or more, and hot otherwise. To do this we used if else statements.

We then store that hint based off the difference in a variable called hint, and we also add the print in that says your wrong and add whatever hint is necessary.

Lastly we make sure it doesn't display on the last strike so we add to the strike count using strikes++ each time the message is displayed.

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 (1)

#3 The code screenshot(s) clearly show the code specific to the feature

shows the code specific to the feature i picked



Checklist Items (1)

#1 Show implementation working by running the program

shows the code works after running. loads user level from name and shows hints depending on guess.

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:

Throughout this homework I did learn some new techniques when needed to do certain things within a java program. I didnt run into any big issues but just some small syntax errors here and there and needed to brush up on some things that helped me create my code. All in all it was simple what was being asked of us, just required some work and focus on our end.



^ 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/msa224/msa224-it114-004/pull/5>



^ 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 (1)

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



Checklist Items (1)

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