

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-number-guesser-4/grade/mbh3>

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

Submissions:

Submission Selection

1 Submission [active] 2/12/2024 11:22:02 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

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:

1. I picked the following two options

1. Display Higher or Lower Hint
2. Display a Cold, Warm, Hot Indicator

2. For the "higher or lower" hint, the logic was straightforward. After each incorrect guess, the game evaluates if the guessed number is greater than or less than the secret number. If the guess is higher, the game advises the player to guess lower next time, and vice versa. This simple feedback loop helps narrow down the range of possible numbers through elimination, making the guessing process more strategic and less about random luck.

The "cold, warm, hot" indicator adds another layer of feedback based on the proximity of the player's guess to the secret number. By calculating the absolute difference between the guessed number and the secret number, the game categorizes this distance into three zones: "cold" for guesses far from the target (more than 10 numbers away), "warm" for closer guesses (5 to 10 numbers away), and "hot" for very close guesses (less than 5 numbers away). This system provides a playful hint about how close the player is to the correct answer without giving it away, adding suspense and excitement to the game.

Both features utilize conditional statements to analyze the player's guess in relation to the secret number, and then provide feedback accordingly. By integrating these hints, the game becomes more interactive and engaging, encouraging players to think more critically about their guesses and learn from their mistakes, all while keeping the gameplay experience light and enjoyable.

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

#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



Checklist Items (1)

#4 A comment with the UCID/date is visible near the code change(s)

code

ucid



Checklist Items (2)

#1 Show implementation working by running the program

#2 Clearly caption the screenshot of what you're showing

output running

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:

1. I picked the following two options

1. Display Higher or Lower Hint
2. Display a Cold, Warm, Hot Indicator

2. For the "higher or lower" hint, the logic was straightforward. After each incorrect guess, the game evaluates if the guessed number is greater than or less than the secret number. If the guess is higher, the game advises the player to guess lower next time, and vice versa. This simple feedback loop helps narrow down the range of possible numbers

through elimination, making the guessing process more strategic and less about random luck.

The "cold, warm, hot" indicator adds another layer of feedback based on the proximity of the player's guess to the secret number. By calculating the absolute difference between the guessed number and the secret number, the game categorizes this distance into three zones: "cold" for guesses far from the target (more than 10 numbers away), "warm" for closer guesses (5 to 10 numbers away), and "hot" for very close guesses (less than 5 numbers away). This system provides a playful hint about how close the player is to the correct answer without giving it away, adding suspense and excitement to the game.

Both features utilize conditional statements to analyze the player's guess in relation to the secret number, and then provide feedback accordingly. By integrating these hints, the game becomes more interactive and engaging, encouraging players to think more critically about their guesses and learn from their mistakes, all while keeping the gameplay experience light and enjoyable.



^ COLLAPSE ^

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

code



Checklist Items (1)

#2 Clearly caption the screenshot of what you're showing

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:

While integrating the "higher or lower" and "cold, warm, hot" indicators into the NumberGuesser4 game, I encountered the challenge of balancing helpful feedback with maintaining the game's difficulty. The key was to provide enough information to guide the player without making the game too easy. Implementing the "higher or lower" hint required careful consideration of how to present the feedback without giving away too much information. I decided to trigger this hint only after incorrect guesses, ensuring it aids learning through trial and error.

The "cold, warm, hot" indicator presented a different challenge. Determining the thresholds for "cold," "warm," and "hot" required testing to find a balance that felt intuitive to players. I experimented with different ranges before settling on values that seemed to offer the best feedback without simplifying the game too much.

To overcome these issues, I focused on iterative testing and refinement. By playing through the game multiple times and adjusting the feedback mechanisms based on experience, I was able to find a balance that enhanced the gameplay without diminishing the challenge. This process underscored the importance of user experience in game design and the subtle art of providing feedback that encourages but doesn't overwhelm.



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/Mohammadh222/mbh3-IT114-004/compare/M3-NumberGuesser-4?expand=1>



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 with the grade for this item)

waka time ss

End of Assignment