

Project Name: Guess the Number

Overview

We are developing Guess the Number, a simple Java Swing-based game where the player must guess a randomly generated number within a specified range. The player has a limited number of attempts to guess the correct number. Our game provides feedback on whether the guessed number is too high or too low, helping the player narrow down their guesses. If the player guesses the correct number within the given attempts, they win; otherwise, they lose. The range can be set between 1 and 100; our application reveals the correct number if the player uses all attempts.

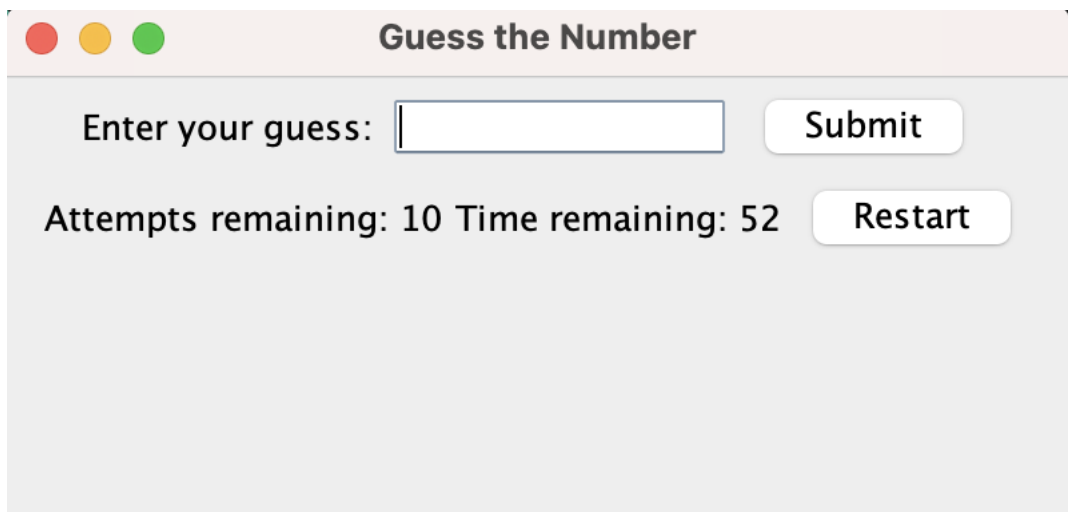
Our game UI features an input field for the player to enter their guessed number, a display for the remaining attempts, and feedback on whether the guess is too high or too low. Additionally, a timer displays the time remaining for the player to make their guesses. If the player fails to guess the correct number within the given attempts, the correct number is revealed.

The game starts with a random number generated within the specified range of 1 to 100. The player is given a fixed number of attempts to guess the number. The player enters a number in the input field and clicks the "Submit" button. Our game provides feedback on the guessed number. If the guess is too high, "Too High" is displayed. If the guess is too low, "Too Low" is displayed. If the guess is correct, "Correct!" is displayed, and the player wins. If the player uses all their attempts without guessing the correct number, the game ends, and the correct number is revealed. A message indicating the end of the game is displayed. The player can click the "Restart" button to start a new game. The input field and "Submit" button are re-enabled for the new game.

Functional Requirements:

1. FR1: The game initializes with a randomly generated number within the chosen range (1-100 or 1-1000).
2. FR2: The player is prompted to enter a guess.
3. FR3: After each guess, the game checks if the guessed number matches the generated number.
4. FR4: If the guessed number is correct, the player wins, and a winning message is displayed.
5. FR5: If the guessed number is incorrect, the game provides feedback:
 - If the guessed number is too high, the game displays "Too High."
 - If the guessed number is too low, the game displays "Too Low."
6. FR6: The game decrements the remaining attempts after each guess.
7. FR7: The game ends when the player runs out of attempts:
 - If the player has not guessed the correct number, a losing message is displayed along with the correct number.
8. FR8: The game displays the remaining time for the player to make their guesses.
9. FR9: The player can restart the game after it ends.

User Interface Requirements (UIRs)



The snapshot above shows the game display.

1. UIR1: The main window displays the input field for the player to enter their guessed number.
2. UIR2: A label shows the number of remaining attempts.
3. UIR3: A feedback label displays messages like "Too High," "Too Low," or "Correct!".
4. UIR4: A timer displays the remaining time for the player to make their guesses.
5. UIR5: A "Submit" button allows the player to submit their guessed number.
6. UIR6: A "Restart" button allows the player to start a new game after the current game ends.
7. UIR7: When the game ends, the input field and "Submit" button are disabled.
8. UIR8: The correct number is displayed if the player runs out of attempts.

Project Plan

| Requirement | Deliverable 1 | Deliverable2 | Deliverable 3 |
|-------------|---------------|--------------|---------------|
| FR1 | x | | |
| FR2 | x | | |
| FR3 | x | | |
| FR4 | | | x |
| FR5 | | x | |
| FR6 | x | | |
| FR7 | x | | |
| FR8 | | | x |

| | | | |
|------|---|---|---|
| FR9 | | | x |
| UIR1 | | x | |
| UIR2 | x | | |
| UIR3 | x | | |
| UIR4 | | x | |
| UIR5 | x | | |
| UIR6 | | x | |
| UIR7 | | | x |
| UIR8 | x | | |