Lesson 03.06 - PROG

Number Guessing Game

In this lesson we will write a number guessing game program:

- Player clicks PLAY button to call **playGame** function
- **playGame** function hides the PLAY button and
- shows the input box for entering guess
- shows the GUESS button for submitting guess
- shows an h2 for displaying feedback
- Program generates a mystery random integer from 1-100
- Player inputs a number by typing or using stepper arrows
- Player clicks GUESS button to call **evalGuess** function
- **evalGuess** function compares guess to mystery number
- feedback is displayed: Too High or Too Low or Congrats
- function keeps track of total guesses
- Once the mystery number is guessed:
- the score (GamesPplayed and Guess Avg) is updated
- the **resetGame** function is called automatically

resetGame function:

- updates the score
- shows playBtn; hides guessBox, guessBtn and feedback
- We are ready for a new game: click PLAY AGAIN

1. Declare game play variables. These need to be in the global scope, since the vars are used by more than one function.

```
let r = guess = guesses = games = avg = 0;
```

The above shorthand requires that the variables have the same initial value.

2. Get the PLAY and GUESS buttons (GUESS is hidden on page load):

```
const playBtn = document.getElementById('play-btn');
const guessBtn = document.getElementById('guess-btn');
```

3. Have the buttons call their respective functions:

```
playBtn.addEventListener('click', playGame);
guessBtn.addEventListener('click', evalGuess);
```

4. Get the guess input box and feedback h2, both of which are also hidden on page load:

```
const guessBox = document.querySelector('input');
const feedback = document.getElementById('feedback');
```

5. Get the footer spans for keeping score:

```
const gamesSpan = document.getElementById("games-span");
const avgSpan = document.getElementById("avg-span");
```

The playGame function runs when the PLAY button is clicked.

6. Declare the playGame function:

```
function playGame() {
```

   7. Generate a random mystery number from 1-100:

   ```
   r = Math.ceil(Math.random() * 100);
   console.log('mystery number:', r);
   ```

   8. Hide the PLAY button

   ```
   this.style.display = "none";
   ```

   9. Show the guess input box, GUESS button and feedback h2:

   ```
   guessBox.style.display = "inline";
   guessBtn.style.display = "inline";
   feedback.style.display = "inline-block";
   ```

   10. Output a prompt to the feedback h2:

```
    feedback.textContent = "Guess the mystery number from 1-100";

} // end function playGame()
```

The evalGuess function runs when GUESS button is clicked.

- if the guess is too low, output: "Guess is too LOW"
- else if the guess is too high, output: "Guess is too HIGH"
- else guess is correct, so "Congrats!" (Game Over)
- the resetGame() function is called when the game is over

11. Declare the playGame function:

    ```
    function evalGuess() {
    ```

    12. Increase guesses by 1

        ```
        guesses++;
        ```

    13. Get the player's guess from the input box and convert it to an actual number:

        ```
        guess = Number(guessBox.value);
        ```

    14. If the guess is less than the mystery number:

        ```
        if(guess < r) {
            feedback.textContent = "Guess is too LOW!";
        ```

    15. If the guess is greater than the mystery number:

        ```
        } else if(guess > r) {
            feedback.textContent = "Guess is too HIGH!";
        ```

    16. If neither too high nor too low, the guess is correct:

        ```
        } else {
            feedback.innerHTML = `Congrats! You guessed<br>the mystery number: ${r}`;
        ```

    17. Game's over, so call the resetGame function:

    ```
            resetGame();
        }

    } // end function evalGuess()
    ```

The resetGame function runs automatically when the player guesses correctly.

18. Declare the playGame function:

```
function resetGame() {
```

19. Reset the player guess and guess input box to 0:

```
guess = 0;
guessBox.value = 0;
```

20. Increment total games played by 1

```
games++;
```

21. Hide the guess input box and guess button:

```
guessBox.style.display = "none";
guessBtn.style.display = "none";
```

22. Show playBtn; this time have it say PLAY AGAIN:

```
playBtn.style.display = "inline-block";
playBtn.textContent = "PLAY AGAIN";
```

23. Update the score average (guesses per game):

```
avg = guesses / games;
```

24. Update the score Games Played and Guess Avg:

```
    gamesSpan.textContent = games;
    avgSpan.textContent = avg.toFixed(2);

} // end function resetGame()
```