## Bug report: #1:

**Title:** Incorrect Win/Loss Record Update

**Description:**

The game incorrectly records every match as a loss for the player, regardless of the actual outcome of the match.

**Steps to Reproduce:**

1. Start a new game.
2. Choose bots for the player's duo.
3. Engage in a duel.
4. Win the duel.

**Expected Result:**

The win count in the player's record should increment by one after winning a match.

**Actual Result:**

The loss count in the player's record increments by one after winning a match, indicating that the game was lost even though it was won.

**Impact:**

This bug impacts the player's experience negatively as it does not correctly acknowledge the player's wins, making it seem as if the player is constantly losing.

**Recommended Fix:**

In the /api/duel endpoint, within the condition block which determines the result of the duel, the player's win count should be incremented if the player's health is equal to or greater than the computer's health. Specifically, the line `playerRecord.losses += 1;` should be changed to `playerRecord.wins += 1;` in the else clause of the conditional statement.

**Environment**: Node.js server, Express.js application

Bug report #2:

**Title:** Unhandled ReferenceError - "botsArr is not defined"

**Description:**

The application throws a ReferenceError: "botsArr is not defined" when a GET request is made to the "/api/robots" endpoint, such as when the user clicks the "See all bots" button.

**Steps to Reproduce:**

1. Launch the application.
2. Click on the "See all bots" button.
3. Expected Result:

The application should display all the available bots without any errors.

**Actual Result:**

The application throws a ReferenceError: "botsArr is not defined" and does not display the bots.

**Impact:**

This bug prevents users from viewing all available bots in the application, hindering the overall user experience.

**Recommended Fix:**

In the "/api/robots" endpoint, replace botsArr with bots (or the correct variable that contains the bots' data).

**Environment**: Node.js server, Express.js application

Test case:

**Test Case Description:**

Test case for the chooseBot function:

This test case is designed to validate the functionality of the chooseBot function. It checks whether the function correctly adds a bot to the playerDuo array, removes the chosen bot from the choices array, and updates the display accordingly.

**Preconditions:**

* The application should be running, and the initial setup should be completed (bots should be fetched and displayed in the choices).
* The playerDuo array should have less than 2 bots.

**Test Steps:**

* Note the initial number of bots in the choices and playerDuo arrays.
* Note the id of a bot in the choices array.
* Call the chooseBot function with the id of the chosen bot.
* Check the choices and playerDuo arrays after the function call.

**Postconditions:**

* The playerDuo array should include the bot with the chosen id.
* The number of bots in the playerDuo array should be one more than the initial count.
* The choices array should not include the bot with the chosen id.
* The number of bots in the choices array should be one less than the initial count.
* If the playerDuo array has 2 bots after the function call, the duel button should be visible.
* The displayed bots for the player should match the playerDuo array.