```
package Battleship;

import java.util.Scanner;

public class actiongame extends gameboard {

	Scanner input = new Scanner(System.in);

	// Initialized game (Creates ships, PLayer Board, and Sets Turns).
	public void StartGame() {
		SetupGame();
	}

	// Execute game. End once turns = 0
	public void PlayGame() {

		StartGame();

		Player player = PlayerArray[currentPlayerIndex];
		for (;player.turns_remaining > 0; player.EndTurn()) {
			PrintBoard(player, player.Player_Board);
			int[] inputs = Move();
			int row = inputs[0];
			int col = inputs[1];
			UpdateBoard(player, row, col, player.Player_Board,
player.Ship_Board);
			if (Winner(player) == true)
				break;
			if (Loser(player) == true)
				break;

		}
	}

	// Get User's move.
	public int[] Move() {

		int row = 0;
		int col = 0;
		boolean notValid = false;
		String letter = "";

		do {

			if (notValid) {
				// output error message for the next loop
				System.out.println("'" + letter
						+ "' is not a valid choice, pick
again");
			}
			System.out.print("Enter coordinates (example a1): ");
			letter = input.nextLine();
			letter = letter.toUpperCase();
```

```
                row = ((int) letter.charAt(0)) - (int) 'A' + 1;

                // getting weird character output when integer is pass
9, got help
                // outside of book
                // col = ((int)letter.charAt(1)) - (int)'1' +1; // was
my code

                try {
                      col = Integer.parseInt(letter.substring(1));
                } catch (NumberFormatException exception) {
                      col = -1;
                }

                notValid = this.OutOfBounds(row) ||
this.OutOfBounds(col);

            } while (notValid);
            return new int[] { row, col };
      }

      // Update Player_Board.
      public String[][] UpdateBoard(Player player, int row, int col,
                String[][] Player_Board, String[][] Ship_Board) {

            // Previously hit space
            if (Player_Board[row][col] != SPACE_EMPTY)
                  System.out.println("\nYou already targeted this spot
dummy");

            // Miss
            if (Ship_Board[row][col] == SPACE_EMPTY
                        && Player_Board[row][col] == SPACE_EMPTY) {
                  Player_Board[row][col] = SPACE_MISS;
                  System.out.println("\nMISS!");
            }

            for (int i = 0; i < 5; i++) {

                  // Hit
                  if (Ship_Board[row][col] == player.ShipsArray[i].icon
                              && Player_Board[row][col] !=
player.ShipsArray[i].icon) {
                        player.ShipsArray[i].status += 1;
                        player.hits++;
                        if (player.ShipsArray[i].status != 0)
                              System.out.println("\nYou hit my " +
player.ShipsArray[i].name
                                          + "!\n");
                        else
                        // Sunk Ship
                        {
                              System.out.print("\nYou sunk my " +
player.ShipsArray[i].name
```

```
                                        + "! ");
                                player.ships_remaining--;
                                if (player.ships_remaining == 1) {
                                      System.out
                                                  .println("I still have 1
ship remaining though.\n");
                                } else if (player.ships_remaining > 1) {
                                      System.out.println("I still have "
                                                  + player.ships_remaining
                                                  + " ships remaining
though.\n");
                                } else {
                                      System.out.println("I have no ships
remaining.\n");
                                }
                          }

                          Player_Board[row][col] = Ship_Board[row][col];
                    }
              }
              return Player_Board;
        }

        // Player Wins
        public boolean Winner(Player player) {

              if (player.ships_remaining == 0) {
                    System.out.println("\n\nYou won!");

                    System.out.print("\n\t\tBATTLESHIP:\n" + "\tTotal Turns:
\t\t"
                                + player.GetTurns() + "\n" + "\tRemaining
Turns: \t"
                                + player.GetRem() + "\n" + "\tShips
Remaining: \t"
                                + player.ships_remaining + "\n" +
"\tAccuracy: \t\t");
                    System.out.printf("%2.2f%%", player.GetAcc());
                    System.out.println("\n");
                    return true;
              } else
                    return false;
        }

        // Player Loses
        public boolean Loser(Player player) {
              if (player.turns_remaining == 1 && player.ships_remaining !=
0) {
                    System.out.println("\n\n\n\nYou Lose!");
                    return true;
              } else
                    return false;
        }
}
```