# Battleship Game (1D Version)

A simplified, one-dimensional implementation of the classic Battleship game in Python, created for the *Introduction to Computer Science (CS-UH 1001)* course at NYU Abu Dhabi, Spring 2024.

---

## 🎯 Objective

The player attempts to sink a hidden two-cell ship by guessing its position on a **1D board**. The game gives feedback on hits and misses, with radar guidance for near misses. You get **three attempts** to win.

---

## 🧩 Features

- 1D board with a default length of 15 (adjustable via a global constant).
- Ship of length 2 is randomly placed on the board, but not shown.
- Player inputs a letter (e.g., `C`) for each attempt.
- Program provides:
    - **Hit** (0) – ship part found
    - **Miss** (X) – nothing there, radar reports how close
- Invalid inputs prompt retry.
- Game ends after player sinks ship or uses all attempts.
- Uses a simple terminal interface.

---

## 🛠️ Game Setup

**Board Size**: Default is 15
**Ship Length**: 2
**Attempts**: 3

The board is represented as a one-dimensional list of 15 spaces (`"  "`). Column labels are letters `A` to `O`.

The ship is randomly placed using:

```
import random
random.randint(0, BOARD_SIZE - SHIP_LENGTH)
```

---

## 🚀 How to Run

1. Open terminal.
2. Navigate to the folder containing the script.
3. Run:

```
python3 battleship.py
```

---

## 🧪 Game Flow Example

1. **Initial Board**:
   ```
   A B C D E F G H I J K L M N O
   - - - - - - - - - - - - - - -
   ```
2. **Player Guess**: E
   - If it's a hit → mark as 0
   - If a miss → mark as X, show distance from ship
3. **Next Turn**: Guess again unless ship is sunk or attempts run out.

---

## ⚙️ Customization

Modify the following constants at the top of your script:

```
BOARD_SIZE = 15
SHIP_LENGTH = 2
MAX_ATTEMPTS = 3
```