Full Stack challange QA  
Nemanja Bogojevic

:

## Core Gameplay Bugs

# 1. Double Score Increment

**Issue:** Score increased twice per correct answer

**Cause:** Double function call to **handle\_correct\_guess()**

**Solution:** Removed duplicate function call

## 2. Mode Confusion Bug

**Issue:** Hard mode features appearing in normal mode

**Cause:** Missing mode state tracking

**Solution:** Added **self.is\_hard\_mode = False flag**

**Impact:** Properly separated game modes

# 3. Streak Limitation Bug

**Issue:** Hard mode streak capped at 5

**Cause:** Single variable tracking both streak types

**Solution:**

* Added **self.lives\_streak = 0**
* Separated **consecutive\_correct** for score streak
* Separated **lives\_streak** for bonus lives tracking
* **Prevention:** Added streak boundary testing

# 4. UI Height Display Bug

**Issue:** Height indicator stuck at last valid position **Cause:** Missing out-of-bounds reset

**Solution:** Added default reset

**def** update\_hover**(**self**,** mouse\_pos**):**

x**,** y **=** mouse\_pos

**if** y **>** self**.**header\_height **and** x **<** WIDTH**:**

row**,** col **=** self**.**get\_tile\_pos**(**mouse\_pos**)**

**if** 0 **<=** row **<** ROWS **and** 0 **<=** col **<** COLS**:**

self**.**game\_logic**.**update\_hover**(**row**,** col**)**

self**.**ui\_manager**.**update\_hover\_height**(**self**.**height\_matrix**[**row**][**col**])**

**else:**

self**.**ui\_manager**.**update\_hover\_height**(**0**)**

# 5. Score Reset Bug

**Issue:** Score not resetting properly between games

**Solution:**

**def** reset\_game**(**self**):**

self**.**score **=** 0

self**.**consecutive\_correct **=** 0

self**.**lives\_streak **=** 0

self**.**show\_name\_input **=** **False**

self**.**player\_name **=** ""

self**.**score\_submitted **=** **False**

**:**

**return** **False**

# 6. State Management Bug

**Issue:** Game state inconsistency after mode switching

**Solution:**

**def** set\_game\_mode**(**self**,** is\_hard**):**

self**.**is\_hard\_mode **=** is\_hard

**if** **hasattr(**self**,** 'game\_logic'**):**

self**.**game\_logic**.**set\_game\_mode**(**is\_hard**)**

**print(**f"Game mode set to: {'Hard' **if** is\_hard **else** 'Normal'}"**)**

### Ways you would test the projects of other contestants ?

If I were testing other contestants projects, I would adopt a variety of actions to identify potential issues. This would include r**apidly clicking all buttons**, **clicking outside valid areas**, **attempting to interact with multiple modes or islands simultaneously**, **repeatedly clicking on water areas**, **and interacting near border regions to observe how the application handles edge cases**.

For projects with an API for submitting new scores, my testing would focus on ensuring robust input validation, preventing **SQL injection attacks**, and addressing other security vulnerabilities.

### Improvements/features you would make if you had a magic wand ?

If I had a magic wand to enhance the project, I would introduce dynamic features like **volcanic islands that change height over time**, **weather effects that impact visibility**, **and time-based height variations to add depth to the gameplay**. Additional enhancements could include **power-ups**, a real-time **competitive mode**, a **3D visualization option** for more immersive gameplay, and **mobile touch suppor**t to broaden accessibility and engagement, I would also implement **random map generation.**

### Think of how some factors could affect your solution?

The **map size** plays a significant role in game balance and scaling. Larger maps would require players to have more lives since locating islands becomes more challenging, adding pressure to both navigation and time constraints. To address this, a minimap could be essential for efficient navigation. However, performance could suffer due to the sheer number of tiles on larger maps, necessitating optimization to avoid lag or crashes.

The **height range** of islands directly influences gameplay difficulty. A wider range, such as 0–2000, increases the challenge of guessing island heights and affects how easily players can interpret color gradients. Logarithmic scaling might help improve visibility and balance. These changes could also shift the difficulty curve, requiring careful adjustments to ensure fairness.

The **lives system** has implications for game pacing and strategy. Allowing more lives can lead to longer games and a less punishing experience, particularly in easier modes. However, in harder modes, the availability of extra lives could become a critical part of the strategy. The number of lives might need to scale with the map size or difficulty level to maintain consistency and balance.

**Performance scaling** introduces challenges with increasing complexity. As the number of islands grows, so does the frequency of collision checks and hovering calculations, potentially impacting responsiveness. Water animations and UI responsiveness might also degrade, requiring optimized algorithms and efficient rendering techniques.