

Code Review: Caleb Presto → Rahim Mirzada

Model: Cell

1. **Type Checking** – Cell access control tied to instanceof checks

Symptoms: Cell gatekeeps entry and occupancy with instanceof checks for Player and Enemy (isEnterableFor, addOccupant, removeOccupant), embedding character knowledge in the tile.

Project impact: Adding new character categories or changing rules requires modifying Cell, increasing coupling and risking invariant breaches across call sites.

Solution: Delegate entry/occupancy decisions to character interfaces or strategy objects so Cell enforces capacity via polymorphic contracts instead of explicit instanceof checks.

Relevant commits:

- [12ea61a](#)

Model: Position

2. **Cryptic Naming** – Position hides column/row intent behind x/y accessors

Symptoms: Position stores coordinates as x/y fields with x() and y() getters, obscuring that they represent columns and rows; callers mirror the cryptic naming when comparing positions or building offsets, spreading ambiguous terminology.

Project impact: The mismatch between grid semantics (rows/cols) and x/y labels makes navigating code harder to read and onboard, increasing the risk of axis confusion when adding movement rules or board generators.

Solution: Rename Position's fields and accessors to col()/row() and propagate the clearer terminology through board, generator, helper, and enemy logic.

Relevant commits:

- [9eed977](#)

3. **Dead Code** – Dead helpers: neighbors4 / manhattanTo / translate / hashCode

Symptoms: Position.neighbors4 constructs adjacent coordinates but has no callers, leaving unused logic that may diverge from movement rules. This is the same for the other unused methods.

Project impact: Stale helper behavior could mislead contributors and adds noise to the API surface.

Solution: Remove the unused helper or integrate it into movement/pathfinding code with tests to ensure correctness.

Relevant commits:

- [9786356](#)

Model: Board

4. **Duplicated Workflow** – Repeated registration logic for board occupants

Symptoms: Four register methods each append to lists and update cells with nearly identical code for rewards, bonuses, punishments, and enemies.

Project impact: Fixes to registration rules risk being missed in one path, and the duplication increases maintenance overhead.

Solution: Extract a shared registration helper that handles list updates and cell mutation, parameterized by occupant type, to eliminate repetition.

Relevant commits:

- [f085d97](#)

Model: Characters – Enemies

5. **Self-Use**: MovingEnemy.tick calls super.tick().

Symptoms: MovingEnemy.tick calls its non-abstract superclass method version that can be overridden.

Project impact: this could potentially break encapsulation

Solution: make the superclass method abstract and let MovingEnemy provide the full concrete implementation of it.

Relevant commits:

- [f085d97](#)

UI: GamePanel

6. **Duplicated Code:** GamePanel.paintComponents
7. **Method Too Long:** GamePanel.paintComponents
8. **God Method (Method Too Many Responsibilities) :** GamePanel.paintComponents
Symptoms: paintComponents(...) had duplicated code from the old drawSprites(...) function. This made it so that drawSprites was never called. paintComponents also had too many responsibilities and was too long.

Solution: first, drawSprites was deleted because we needed three total “draw” functions.
The following functions were made:

- drawSpritesForImages
- drawSpritesForSymbols
- drawBanner

These functions were then called in paintComponents.
paintComponents used to be ~300 lines long. Now it is ~45 lines.

Relevant commits:

- [61f2dd7](#)
- [1adbee4](#)
- [dd9283b](#)
- [1d2aa5a](#)
- [1a83d07](#)

9. Data Clump – Data clump of rendering constants

Symptoms: Multiple colors are scattered as primitives throughout GamePanel instead of grouped.

Project impact: Global tuning is error-prone, requiring edits in several places with risk of inconsistent values.

Solution: Group related rendering constants into either

- a) a configuration object or enum, then reference those centralized values, or
- b) a section of the GamePanel initialization parts, and reference those values

Relevant commits:

- [48a7970](#)