

Assignment 4: Code Review

This report details the refactorings and identified code smells of our game.

Note: Since we are a team of three, we did this assignment all together.

Code smell: large class. `Map` class had a private `loadGrid()` method that was intricate enough to test.

Solution: Created a new class `MapGridLoader` that has `loadGrid()` as a public method.

commit: ecb547e3c53161b9d2ade4ec2cd227f393c6fbbe

Class: `Map.java`

Code smell: large class/long method. A lot of the movement and direction functionality in the `MainCharacter` class was in its `update()` method.

Solution: Created a new class `PlayerMovementHandler` that encapsulates all player movement/direction and key press inputs.

commit: 6c212e071e24c156d77c5b14e854a85331f9e98a

Class: `MainCharacter.java`

Code smell: dead code. A portion of the `collidingWithExit()` was never executed since the exit `Passage` cell can only be entered in one direction.

Solution: Removed cases that were not accessible.

commit: 30db4706c44a34d6b935621dba093f66552cac2c

Class: `CollisionHandler.java`

Code smell: dead code. The fields `mapWidth` and `mapHeight` are never used in `Map` or requested by any other class.

Solution: Removed the two fields and their setup in the constructor.

commit: 7e2d71a9a6cca019949dd09473073ba220c65668

Class: `Map.java`

Code smell: Confusing Class Hierarchy. All other `Entity` s have a class that spawns them in aside from `MainCharacter` and `Star` . There is only one `MainCharacter` , so spawning only one in is easy, but spawning multiple `Entity` s should be handled by a class. Solution: Created new class: `StarSpawner` to handle all the spawns for stars, similar to `TrapSpawner` , `SlimeSpawner` and `CoinSpawner` (Previously known as `RewardSpawner`), and removed the previous implementation from `Game` , replacing it with the new methods from `StarSpawner` . Commit: dc1c39477e2a47fff5df0f768e0008f7f9a88403 Class: `StarSpawner.java`

Code smell: Confusing Location of Method. `GUI` 's methods `drawStarShell` and `drawStarCollected` should be implemented in `Star` 's draw method since it's dependent on the `Star` 's variable. Solution: Moved `GUI` 's star UI implementation to `Star` 's draw method. Commit: dc1c39477e2a47fff5df0f768e0008f7f9a88403, 248e371eb857ec5757cdf6f8c333a3e5ee28b68d Class: `GUI.java` `Star.java`

Code smell: Bad Class/Method Names. Confusing Class Name. `RewardSpawner` was originally made to spawn all rewards, but ended up being used just for `Coin` s. Solution: Changed `RewardSpawner` to `CoinSpawner` , changing all the methods and variables from "Reward" to "Coin", and updated all the methods called in `Game` . Commit: 5e2e4083998c83ec584886a93be6e57a5e062192 Class: `CoinSpawner`

Code smell: Unused methods `displayWinScreen` and `displayLoseScreen` in `Screen` Solution: Removed the methods. Commit: e2641bfe437e49c9b50474b6e3af4f5542caf1ce Class: `Screen.java`

Code smell: Dead code. Classes `MainCharacter` , `Map` , `MovingEntity` , `Reward` all contained unused import statements. Solution: Removed the import statements. Commit: bfdd2d050c62f18162b6e5405b8b9802d47a6e14 Class: `MainCharacter` `Map` `MovingEntity` `Reward`

Code smell: Badly structured project. No packages (at all) to organize similar classes. Solution: Created 5 new packages to encapsulate similar class. Commit: 94f9adca53b0cd267c016f16c9fd4b02006947ad, e359f7904f6bdc36093a8299e5389b6ff3eb2284 Packages: `cell` `entity` `main` `map` `spawner`

Code smell: Unused methods. `draw` , `configMovement` Solution: Removed the methods. Commit: b81a42476e2f6ba4d7069ef18744fbe57151d96b Class: `Entity`, `Slime`