Code Review Group 2

By Shawn Xie, Sriyathavan

Below are some bad smells we noticed within our code and some of the refactoring strategies we utilized. To reference a commit, we used the first 7 digits of it's SHA:

Long Method:

Reason: Inside Board class the updateTerrain() method way too long because it contains all the code to generate map

Refactoring: Broke down the updateTerrain() to 2 methods. updateTerrain() is now just responsible for maze generation using recursive backtracking and spawnRoom() generates room. Reference: c36095d <- main

Dead Code

Reason: Inside the board class there was unused method printMap(). Was originally used for testing by seeing a map inside the console. Inside the GamePanel class there was unused CollisionChecker object initialization.

Refactoring: Got rid of the printMap() in Board class and got rid of CollisionChecker inside GamePanel class. Reference: 55f0dbf <- main, a1717db<-main

Lack of Documentation

Reason: GamePanel and Board class lacked javadoc comments for code that provides very little context. Since we associate entities with numbers, it is vital we indicate that to future developers.

Refactoring: Provided context to what object a number maps to through javadoc comments. Reference: ea4cd7f <- main

Poorly Structured Code

Reason: There was misleading code inside TileManager class. It contains a method loadMap() which copies a map from Board class into a private variable which is an extra step and can cause developers to bounce around. Can also be an example of useless variables.

Refactoring: Used the Board class getter function for map to generate world from TileManager. Reference: d01a92b<- develop

Unnecessary Comments

The character class had unnecessary comments within the code. They were located in the updateLocation() function, and the code block was comments aimed at debugging the reward and punishment behaviours when detecting collisions. This inadvertently could impede readability and hinder the debugging process in the future. Since these comments are now obsolete, we deleted them from the codebase. This proactive measure ensures a more streamlined and comprehensible codebase, ensuring smoother future maintenance and debugging efforts. The commit can be found at: 72ddc1c on the main branch.

Speculative Generalization

In the updateLocation() function within the character class, there was extra logic in case the implementation needed to be changed/reverted back in the future. Specifically, the code was designed to enable collisionless movement in case there were bugs in the implementation. Since the logic for collisions is now fully tested and works smoothly, we have decided to remove the additional functionalities that were initially incorporated. The commit can be found at: fd348ff on the main branch.

Temporary Field

In the MovingEnemy class there were some lines of code that had unnecessary temporary fields in the move() function. Instead of using temporary fields, the values were inserted directly into parameters and used to update variables. While having temporary variables may look nice for understanding what's going on, they simply add to clutter making it harder to debug in the future. Removing that extra step keeps the code easy to maintain and refactor. The commit can be found at: eb76b34 on the main branch.

Unused Variable

In the Character class there is a variable named "direction" that is unused. Not only is this adding clutter to the code, it also adds confusion because there is a local variable in the updateLocation() class, which is also in Character, named "direction" as well. Additionally, the MovingEnemy class that extends Character also has variables for direction logic. Removing the variable "direction" in the Character class, is simply put, a step in the right direction. The commit can be found at: 3ff8666 on the main branch.