

Assignment 4: Code Review

**Vivien
Lovejeet**

- 1) Lack of proper exception handling for image loading and map loading
 - a) The original Enemy image loading only used general IOExceptions and just printed the stack trace. With custom exception handling, more context is provided when an error occurs. This change was made in commit 45e5a18.
 - b) Tile manager now throws and catches an image loading exception when attempting to load tile images potentially fails. It prints a specific message whether the image was not found or had failed to load. This change was made in commit e1cb139.
 - c) The load map method now throws a custom map loading exception which provides a message when the map fails to load correctly. This change was made in commit e1cb139.
- 2) Avoiding the use of “magic numbers” and values
 - a) In the original code, unnamed numerical constants were used throughout the Enemy class without any context. To improve the readability and quality of the code, these constants were replaced with descriptive variable names and declared at the top of the class for easier modification. This change can be found in commit 63b077c.
- 3) Method that is too long
 - a) The updateEnemy method had too many lines of code and was difficult to read, understand, and maintain. To solve this, the method was refactored with numerous smaller helper methods. Each method has a single responsibility and is called by updateEnemy. This change can be found in commit ec883b4.
 - b) The collisionChecker method originally handled both collisions for interactions between the player and objects as well as the player and barriers. This has been changed as the method checkObject method was moved into its own class called CollisionCheckerObject. This change was made in commit 9994986.
- 4) Got rid of unnecessary if/else
 - a) The alternateDirection method was refactored to avoid the use of unnecessary nested if/else statements. The nested statements were replaced with ternary operations that combined the condition-checking and made the code more concise. This change can be found in commit f93b419.
 - b) The draw method had redundant if/else checks for determining which sprite image to use. This was fixed with the addition of a helper method that used

ternary operations to determine the correct sprite image. This method is then called in the draw method, which also ensures the draw method is now only responsible for rendering the graphics. This change can be found in commit c761e78.

5) Refactored methods with code duplication

- a) The original implementation of `collisionWithEnemy` and `collisionWithPlayer` contained significant code duplication, as both methods checked for collisions and bounds of the rectangle representing the entity. Helper methods were created for `isCollision` and `getBounds` to reduce code duplication and ensure each method only had a single responsibility. This change can be found in commit 59c08d1.

6) Moved file paths into their own variables

- a) In the `TileManager` class the path for the map text file was initialized within the default constructor. The map path has been moved into a default `String` variable at the top of the class. Similarly, the tile image paths were removed from the `getTileImage` method and placed into their own default variables at the top of the class. This allows others to easily identify the file paths being used in the `TileManager` class, and change them or call multiple instances of the path. This change can be found in commit daa6765.