CMPT 276 Assignment 4 Report

Rodrigo Arce Diaz, Caleb Hengeveld

Code Review:
Upon doing a manual code review, we identified different kinds of smells that could benefit from being refactored. Working primarily in the rewards and entities directories, many of the smells we identified were of two main types, those being duplicated code smells and long method smells.

In the enemy and character classes we identified a duplicated code smell. We were able to move certain getters and setters into the parent class entity.

- *Moved direction, getters and setters, and constructors for enemy and character into Entity.java*: 15-11-2024, commit a2ae7789534a998d8b36c38e801986482436e4bf
    - Smell type: duplicated code

In the moving enemies class calculations defined by find_player were refactored and made into its own function to reduce the clutter and make it easier to understand.

- *Refactored distance calculations in find_player into its own findPlayerCalculations method*: 30-11-2024, commit 75e47c0326a98d8e86ab3d8789499b3390c259e2
    - Smell type: long method

Another smell in the moving enemies class was another long method for determining the movement selection on the y axis.

- *Refactored movement selection if-else statements for when moving along the y axis is preferred in find_player into its own wantsToMoveAlongY method*: 30-11-2024, commit bd6f4a1a3297a2624304bb050190ee5848ef756b
    - Smell type: long method

Similarly the movement selection on the x axis could also be refactored as it was also a long method.

- *Refactored movement selection if-else statements for when moving along the x axis is preferred in find_player into its own wantsToMoveAlongX method*: 30-11-2024, commit 4113bf73883041a93caa236620862a284a901cb0
    - Smell type: long method

There was a duplicated code smell in the entity class that was refactored by using a function to check if the entity can move.

- *similar code for each case in Entity's direction method refactored into private method*: 29-11-2024, commit 6df7cd4355f192e9d7d9282520a8c7cb89bc544e
    - Smell type: duplicated code

Going over the previous refactor fix, we identified that we could further simplify our code by using more functions as well as using the ternary operator to clean up the if statements.

- *Refactored movement selection making it conditional and using separate functions for repeated code*: 30-11-2024, commit 22fbbf1ee2541324ab812c664d8a319c7c0b7b8b
    - Smell type: duplicated code

Moving on to the rewards and punishments, we noticed that they had lots of duplicated code so we could refactor it by creating a Collectible class and making rewards and punishments extensions of said class.

- *Refactored rewards, punishments, and their normal subclasses by taking repeated code, putting it in the Collectable class, and making rewards and punishments extensions of said class*: 01-12-2024, commit f048a266a13c17b239e652942c6e4f69ae6eef0e
    - Smell type: duplicated code

In the board class, the way we set up the board was very messy and cluttered. We refactor this by creating helper methods to clean up the code.

- *Refactoring Board's createBoard method into several smaller private methods*: 01-12-2024, commit dff15aeb7b54c700e19e46eaf3c5e5b9db33f846
    - Smell type: long method

The same kind of logic for generating new bonuses could be reduced in the same way the creation of the board was made. These changes were implemented to clean up the code

- *Refactored genNewBonus to use createAllBonusRewards and createAllBonusPunishments to reduce duplicated code*: 01-12-2024, commit 7b208336b35982d73961b8a8e3c525fab6db4a8e
    - Smell type: duplicated code

Some extra refactoring in the moving enemies class function to reduce clutter and improve clarity was done by extracting logic from the created function into smaller private methods.

- *Extracted the logic from findPlayerCalculations into a helper*: 01-12-2024, commit acbd5b79e6b87adb86bd5cb7e60e431e1832034c
    - Smell type: long method