CMPT 276

Assignment 3 : Code Review

Pavel Jordanov & Tai Png

For clarification, the both of us together worked on the 5 refactorings on the code written by Pavel followed by working together on the 6 refactorings of the code written by Tai.

Work on Pavel's Code:

1. Inside the MovingCharacter.java class we changed the name of spriteNum to WilburFoot. we made this change because the spriteNum naming was confusing and did not reflect well on what the variable was responsible for. More specifically, spriteNum was holding multiple images of Wilbur with his feet at different positions. Alternating between these images makes it look like the Wilbur character is walking because his left foot and right foot will alternate. Therefore, we changed the variable to WilburFoot to reflect which foot is in front. For example, WilburFoot = 1 will show Wilbur with his left foot forward and WilburFoot = 2 will show Wilbur with his right foot forward.
    a. Commit: 005f4be233573853d12c1e40734fa633fa272b61
2. spriteCounter was removed as it was meant to serve as a check for whether Wilbur's feet should change. However, at the start of the stepChanges method in Board.java the counter was incremented, and at the end the counter was decremented, therefore serving no purpose.
    a. Commit: 1db36e3afb5f85417f259c9654116f8091abbc40
3. Deletion of multiple methods and variables were done as they were unused in the program. These unused methods and variables included:
    a. void setPosition(int x, int y) in the MovingCharacter.java class
        i. Commit: 316ca9b007f610c3ad558dcd18e32eccef2d1e59
    b. All the getters and setters for WilburFoot and WilburFootCounter in the MovingCharacter.java class
        i. Commit: 270e45da9bc1abf42adbd873e4c50428442688ef
    c. Speed variable, getSpeed method and the default constructor for MovingCharacter which initialized the speed variable to 1.
        i. Commit: ba93b28c5099b53dbd7b9353593a396707d9a410
4. Removed code duplication inside the Wilbur.java class. In the initPlayerImages() method there were originally four 'for' loops to store the various images of each direction of Wilbur (Up, Down, Left, Right). There was originally a 'for' loop for each direction because a direction may have more images than another direction. However, in the end each directional image of Wilbur has exactly 3 images, therefore the four 'for' loops were combined into one. The upper bound for the 'for' loop was replaced to a final variable called 'wilburDirectionImageLength' since we know with absolute certainty that each direction has 3 images, this guarantees no other method could change this variable by mistake, resulting in erroneous direction image initialization.
    a. Commit: e4a00846971d2de244f4b3017c34daf83a0ce50d
5. A reworking on documentation was also performed to provide a clearer explanation of what the class/method/variables did. For example in the stepChanges method, the prior documentation stated that the method changed the sprite image drawn on the board, which is true. However, this was replaced with an explanation that the Wilbur character would have his sprite change by having its feet alternate, giving the illusion that the Pig character was walking. This is a much better reflection on what the method did.
    a. Commit: f9ab89f43085549890a466b05b708333472a245c
    b. Commit: 9e42d534a8c989dfc63a5afac86d4d5ebb9db399

After performing each refactor, we tested the game out multiple times to make sure there were no bugs produced due to the refactoring. JUnit tests were also conducted and we ensured that all of the tests passed with no errors.

Work on Tai's Code:

1. The first smell we noticed was that there was an unused variable in the StationaryModifier class:
   public static final int BUTCHER_DAMAGE = 10;
   This variable was completely unused in the entire code. To test its importance and use, we commented the line out first, then we ran all of the related tests and they all passed. Lastly we ran the game and it was working perfectly fine. To refactor the StationaryModifier class, we simply deleted the line of code which creates and initializes this variable.

2. The second smell we noticed was that the SationaryModifier class had a variable that had an inappropriate name. The line of code is:
   public static final int BEAR_TRAP_DAMAGE = 10;
   We initially had it like this because we believed early on in the project that the traps should cause damage but they just rather decrease your points. To fix this we simply renamed the variable to a more appropriate name and the line of code is now:
   public static final int BEAR_TRAP_POINTS = 10;
   So now the variable refers to the point changes caused by the bear trap.

3. The third smell we noticed was that the StationaryModifier class had a piece of dead code which was a dead method. This method was the randomCoordinate method. It was meant to create a random coordinate for the items on the board. This was eventually phased out since our JSON file initializes all of the items on the board, effectively rendering this method useless despite it still taking up memory. To fix this, we commented out its uses in the StationaryModifier's subclasses, removing all uses in the process. Afterwards, we ran all related tests that had been written and they all passed, followed by testing the game itself which was unaffected by this change. As a result, we deduced that we could delete this method and its usages in the entire project to refactor Tai's code. Following this, we had to remove the now unused java.util.Random import.

4. The fourth smell we notices was that there was two more unused variables which were supposed to be related to the randomCoordinate method in the Stationary Modifier class. Noticing that these were unused, we simply performed a safe delete to refactor the code.

   The link to the four refactorings is the commit:154006ed79b4a66c2c2eac4d7d16e38b28952894

5. The fifth smell we noticed was that there were 3 completely useless methods in the Stationary Modifier class. These methods were used as setters to set the x and y coordinates of the items on the board. The purpose of these disappeared since the JSON file takes care of this and they were never even used in the first place. To fix this and refactor, we simply performed a safe removal of these 3 methods.

   The link to this refactoring is the commit: ae8bd21016adbff26b24178b6c4e24131de929fd

6. The sixth smell, we noticed that the ItemType enumeration did not have any documentation. Since all methods and classes require documentation we added Javadocs to the enumeration.

   Link to this refactoring is the commit: 67408340b91b7a68218383933fb0c29b1dd60840