CMPT 276: Assignment 3

Kaleigh Toering & Henry Ruckman-Utting

CODE SMELL: In changeState() there were several variables created which represented certain keys being pressed down in the KeyHandler class. These variables were never changed later in the code, only referenced once each when checking the value of the boolean. The code could be simplified by removing these variables and checking if the keys are pressed down in the comparisons where the booleans were checked.
FILE: state.java
RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/54/diffs?commit_id=1731e335a750cc6d80730af933084d8ace058ba0
SOLUTION: The unnecessary variables were removed and their appearances in the code was replaced by keyhandler checks, the same way the variables were created.

CODE SMELL: The variable node[][] which represents the map in the game is poorly named because node does not accurately represent what the variable is.
FILE: pathfinding.java
RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/54/diffs?commit_id=e2d1fece5a4ff691fb9abb19184d4aad943749d2
SOLUTION: The variable names were changed from node to map to more accurately show what the variable represents.

CODE SMELL: In all of the update functions for the MovingEntity classes, there is repeated code which deals with updating the entities position with regard to it's direction, this code should be extracted and moved into it's own method or class to remove the repeated code.
FILE: movement.java, movingentity.java, monkey.java, zookeeper.java
RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/54/diffs?commit_id=ba87ea7b89af7accf9c3c0598aeccb054d15011f
SOLUTION: The creation of a new class called Movement which can perform the movement of these moving entities for them in a method called move(). This helps to clear up some repeated code in the update functions of the MovingEntity classes.

CODE SMELL: The method of assigning a random value to the lifecycle variable in Banana.java is not easily understandable, and provides little flexibility if the range of possible life cycle values needs to be changed in the future. (i.e. Poorly Structured code)
FILE: Banana.java

RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/65/diffs?commit_id=d2f4c27c3e26434617f0c62fa6e8ead89a1f01c0
SOLUTION: A private method was created called generaterandomLifecycle(), which takes in the shortest lifecycle value and the longest life cycle value wanted by the user. The function then generates a random integer within the range specified by the input parameters and sets the lifecycle variable to this generated value. The equation has been expanded to show the logic behind the number generation, to increase understanding in case it needs to be modified later.

CODE SMELL: The parameter variable, i, in the setFile() method of the sound class is not descriptive enough to provide context to the user about what the variable represents. (i.e. Bad/Confusing variable name)
FILE: Sound.java
RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/65/diffs?commit_id=3eb9a8bc4ba65e8a08ce9abe3dd3ca3aea7fd88e
SOLUTION: The setFile parameter variable was changed from “i” to “soundFileIndex” to more clearly describe what the variable represents, and what information should be passed into the method.

CODE SMELL: Both the FixedEntity and MovingEntity classes had x and y member variables, which held the x and y coordinates of the entity’s position on the map. (i.e. Data clumps)
FILE: Banana.java, FixedEntity.java, key.java, LionPit.java, Monkey.java, Movement.java, MovingEntity.java, Zookeeper,java, Collision.java
RELEVANT COMMIT:
https://csil-git1.cs.surrey.sfu.ca/cmpt276f22_group17/project/-/merge_requests/65/diffs?commit_id=dba952de9847219531f6a71f7497f845ef6c4c10
SOLUTION: Since we had previously created a Position class to encapsulate coordinates for another class in our project, we thought it was best to change the entities to have a Position member variable so we could group similar data together. During this refactoring, we also changed the coordinate information to be protected instead of public to increase the encapsulation of the entity classes, so getter/setter methods for each coordinate were also added.