

## Assignment 3: Code Review

### Scavenger.java

#### Code Smells:

1. Unused getter methods
2. Unused setter methods: setX(), setY(), setDirection()
3. Long draw() method, there's a section of code that can be put in it's own method and be called from the draw() method
4. Long update() method, there's a bit of code that is used to track the player. This could be put into its own method.
5. Long update() method, there's a section of code at the bottom of the method that animates the Scavenger. This can be placed in its own method.
6. We add javadoc and comment when necessary.

#### Refactor:

Code Smell 1: In Scavenger class, we remove the getX(), getY(), getDirection() methods because these methods were not used. Instead we have an update method to get the coordinates and accessible variables from the superclass, Entity.

Code Smell 2 : In Scavenger class, we also remove the setX(), setY(), setDirection() methods because these methods were not used.

Code Smell 3 : The original draw() method in Scavenger class was too long. However, because we thought the switch cases were necessary for the method, we didn't change that part. Instead we extracted the code at the bottom of the code (commit 13ce3655) and placed it into its own method called drawCurrentScreen().

Code Smell 4: The update() method in Scavenger is too long as well. We refactor the code chunk that tracks the player into its own method which is to be called in the update method. (commit fe22dab7)

Code Smell 5: For the update() method in Scavenger, we refactor the code chunk that animates the scavenger into its own method and be called from the update() method. (commit 49505077)

Code Smell 6: We add javadoc and comments throughout the code to help people understand and maintain the code more easily.

#### Conclusion:

We understand the importance of reviewing code and trying to make sure the code quality is at its best. In the previous phase, we continuously tried to improve our code. At this point in time,

most of the codes assigned to us had been refactored, so, in our opinion, there wasn't much to change. For maximum efficiency, one person was incharge of refactoring and pushing the code and the other was in charge of documenting the changes and such. We always make sure the changed code implements the correct functionality and pass the tests before we commit and push to the git repository.