Aaron Lopez (301434253)
Jordan McKenzie (301551839)

1. **Unnecessary switch statement**
   Commit 49f8055 “Code Smell: Unnecessary Switch Statement Refactored”

   There was a switch statement for handling collisions between the player and entities, which I replaced with a function in the gameobject class. Now, when a player is on the same tile as an entity, instead of checking for the type and performing the correct actions, we just call the ‘onHit’ method of the entity.

2. **Unnecessary switch statement**
   Commit ad74c94 “Code Smells: Split classes into packages, removed unnecessary switch statement for displaying gui screens”

   There was a switch statement for handling what screen to display when switching between the various different gui screens (start menu, scoreboard menu, etc.) I replaced the switch statement with multiple new classes, which all inherit from a base “GUIScreen” class. Now, all we need to do is call the ‘display’ method on the current screen.

3. **Bad Project Structure**
   Commit ad74c94 “Code Smells: Split classes into packages, removed unnecessary switch statement for displaying gui screens”

   Currently, all of our classes were stored in the default package. I split the classes into two different packages; the GUI, and the Game. This provided a more organised file structure. I also had to fix the imports for all of the tests.

4. **God Class**
   Commit e1f6eba “Code smell: Moved gui screen functionality to appropriate classes”

   The functionality for the gui screens were all done in the gui manager class. Instead of having each screen class call a method from the gui manager, I moved the method functionality to each class’ respective display method. This reduced the amount of work done by the gui manager class.

5. **Bad/confusing name variables**
   *C*ommit ce7d74d “Code smell: Changed multiple variable names”

   Renamed a few variables as some were ambiguous or were not specific. Simply refactoring one variable name did not feel enough to consider a proper refactoring, so I decided to change as many variable names as I could. (That were even slightly confusing) I changed the variable “completionCount” to “collectedBatteries”, “score” to “finalScore”, “oxygen” to “currentOxygen”, and “oxygenRate” to “oxygenDecreaseRate”. These changes were all done within the GameManager class I additionally made a few variable name refactorings in my last comit/refactor (*Commit* 2b16be8*)*

6. **Unnecesary switch statement**
   *Co*mmit 70c2729 “Code smell: Split the levels into a package named Levels.”

   In the GameManager class, I placed the switch statement that initialized each level’s variables into a separate level package. I also had to change all the instances which used a numerical value to initialize a level. This led to a chain reaction of things that needed to be changed: imports, function calls, class structures, tests etc.

7. **Poorly structured code/Feature envy**
   *Com*mit 70c2729 “Code smell: Method that performed a task which should have also been moved with level initialization ”

   Because I moved all the variables that initialize each level to a separate class, the startLevel() method had only one purpose which was to create the respective level’s map. I decided that this method should ultimately be removed and the map creation should be with the level initialization classes. Note that this refactor was done simultaneously as refactor 6 as I realized almost instantly that this was an issue.

8. **Magic Number**
   Commit 2b16be8 “Code smell: A friend of mine was looking over my code and could not easily identify the purpose of constant values that were used.”

   Changed multiple classes that had constant values that could be named. This helps readers to understand the code better than to guess what the constant numbers do. For example a common constant used was 40. This number represented the conversion of oxygen (ranging 0-4000) to score (ranging 0-100). I then refactored the number to be a “private static final int” which also has a variable name “OXYGEN_CONVERSION_RATIO” and value “40”.
   However, upon refactoring the code, I also noticed two things: The scoring system was not correctly working and scores were not saving to the leaderboard. The first issue was due to incorrect values being used to calculate the score/oxygen score overflow. The issue regarding the leaderboard was due to a previous refactoring not correctly assigning a value to the level variable.