Assignment 3

Group 4
Matin Keivanloo
Ana Premovic

Code Smell 1 - Confusing Class Hierarchy + Unused Variables
Commit ID: `e92a6f16`

> **Problem**: The class hierarchy of the reward objects is poorly structured. Class Reward extends class ImmovableObject. Then, classes RegularReward and BonusReward extend Reward. However, all the common functionality of RegularReward and BonusReward is already covered in ImmovableObject. On top of this, Reward contains the unused variable points, left over from early in our design.
>
> **Fix**: Delete class Reward including its unused variables and the getters associated with the variables. Update RegularReward and BonusReward to directly extend ImmovableObject. These changes will reduce redundancy, allowing for a more concise and readable class hierarchy.

Code Smell 2 - Duplicate Code
Commit ID: `358c85d8 + 2fb97e70`

> **Problem**: In class MainCharacter, when the player touches a lava tile, there needs to be three new carrots displayed to the screen. There is duplicate code to generate a new random position for the carrot to be displayed at. This code is copy pasted from method displayObjects() in class ImmovableObjectDisplay.
>
> **Fix**: In class ImmovableObjectDisplay, in method displayObjects(), extract code to generate a new random position for the object. Place this code in a new method called generateRandomPosition(). Now, instead of copy pasting this code, invoke the new method generateRandomPosition() in ImmovableObjectDisplay as well as MainCharacter.

Code Smell 3 - Long Method
Commit ID: 31aec9a7

> **Problem:** The displayMenu method seems to be doing multiple things, such as creating a window, setting the background image, creating buttons, adding buttons to the window.
>
> **Fix:** displayMenu was separated into smaller methods to make it more maintainable.

Code Smell 4 - Duplicate Code + Long Method
Commit ID: `5de6984d`

Problem: In class CollisionDetector, the method detectImmovableObject() is too long and complex, making it hard to understand and maintain. It also contains duplicate code in its switch statement.

Fix: First, remove the duplicate code from the switch statement. Place it after the switch statement, as it is called in every case. Then create two new methods: setCollidableAreaPositionsForCollision() and resetCollidableAreaPositions(). Extract the code from the beginning and end of the method, and place it into these two new methods. Now, the complexity has been broken down into multiple methods. The most important part of the method, the actual collision detection, is kept in the original method. These changes make the method a lot more readable by making its purpose clear.

Code Smell 5 - Poorly Structured Class Hierarchy + Duplicate Code
Commit ID: 8d4aa9a1

Problem: Both the winningmenu and gameovermenu had duplicate code. specifically, the duplicated code was responsible for rendering the background image on the menus
Fix: To fix the problem of having duplicate code in both GameoverMenu.java and WinningMenu.java, we created a new class called Menu.java. This class contains the code that was duplicated in both menus, such as displaying the background image. We then made GameoverMenu.java and WinningMenu.java extend the Menu.java class. By doing this, we were able to eliminate the duplicate code and make our code more organized and easier to maintain.

Code Smell 6 - Long Method
Commit ID: 663bf222

Problem: restart method was too long, makes it harder to maintain and update the code over time.
Fix: To solve this problem, we broke it down into smaller, more manageable pieces. This was achieved by dividing the code into logical sections or sub-tasks, each with its own method. For example, separated methods for resetting the game state, resetting the score, resetting the timer, resetting game objects, and creating a new game window.