Group 21 December 8, 2022
Preet Dhadda & Jimmy Hui

REFACTORING REPORT

Identified Code Smells

Below are the code smells we identified in both of our code:

Number	Code Smell Type	Instance
1	Confusing variable names and variable encapsulation issue	• Located in Character class • characterLeft and characterRight variables sound like they describe the direction of the character, but they store Images • imageDirection variable does not need to be public
2	Duplicated code	• Located in loadCellImage() method in CellMap class • Lines for creating a new cell and updating its attributes are repeated several times
3	Long method	• Located in checkCell() method in detectCollision class • checkCell() is a very long method and as a result, its functionality is unclear
4	Low cohesion, code duplication, and feature envy	• Located in checkCell() method in detectCollision class • The checkCell() class is doing too many things besides checking cell (for example, it is also instantiating and deleting UI windows and resetting game variables) • Method regularly and excessively calls methods in other classes
5	Data clumps and code duplication	• Located in LosePanel, WinPanel, RulePanel, and TitlePanel • Button's calls and attributes are clumped together in different places • Similar code for replayBTN and continueBTN are found in multiple classes
6	Poorly Structure code and unnecessary use of unsafe or unsound constructs	• Located in Character class • There should only be one instance of the Character class throughout the whole game

Group 21 December 8, 2022
Preet Dhadda & Jimmy Hui

Refactors

Below are explanations of how we refactored each code smell:

Number	Commit Information	Solution
1	Commit ID: 06496953 Commit message: "Refactored variable names and accessibility in Character class"	• Changed variable titles from 'characterLeft' and 'characterRight' to 'characterLeftImage' and 'characterRightImage' to avoid confusion as to what the variables do • Changed imageDirection variable from being public to private
2	Commit ID: aec0cfb0 Commit message: "Refactored loadCellImage() in CellMap to use a helper method"	• Extracted the lines of code that set the cell's attribute values and locate the cell's image into its own helper method to reduce code duplication • New helper method is called createCell()
3	Commit ID: 705ba2c6 Commit message: "Refactored checkCell() in detectCollision to use a helper method"	• Extracted the beginning of the method that simulates the character's movement into its own helper method to make checkCell()'s purpose more clear • New helper method is called SimulateCharacterMove()
4	Commit ID: 003f283c Commit message: "Refactored checkCell() in detectCollision to increase cohesion"	• Moved the UI calls in checkCell() to the UI class to reduce feature envy • Improved code cohesion because now the checkCell() method only focuses on checking cells and not instantiating and resetting UI panels • New methods are ui.makeWinRewardsWindow(), ui.makeLoseRewardsWindow(), and ui.makeLoseScoreWindow() in the detectCollision Class
5	Commit ID: 83aac632 Commit message: "Refactored buttons by extracting JButton in Panel classes into a Buttons class"	• Removed data clumps and duplicate code in JPanel classes by moving them into their own object • Reduced code duplication from 5 lines to 1 line in the LosePanel, RulePanel, TitlePanel, and WinPanel classes • Refactored code calls Button's constructor and can be found in LosePanel, WinPanel, TitlePanel, and RulePanel
6	Commit ID: e8121539 Commit message: "Refactored Character class to follow the singleton design structure"	• Applied the singleton design pattern to the Character class to ensure only one instance of the character exists throughout the entire game • Used a getInstance() method to call the Character's constructor • Refactored code can be found in the Character class and in ScreenPanel where getInstance() is called instead of directly calling Character's constructor