Sina MohammadiNiyaki
Khai Duy Nguyen

Code Review

In this report, we conducted an in-depth examination of our game's code, focusing specifically on the UI.java and TileManager.java classes. These classes are pivotal to our game's operation and offer substantial scope for enhancement. Our objective was to refine the code to enhance its clarity, flexibility, and resilience, preparing it for future updates and expansions.

1. The TileManager class needed was very unorganized in the method getTileImage(). Therefore, I decided to make another method called setup that makes the process much smoother and results in fewer lines of code as well. We commented out the old code just to have it as a reference but decided to remove the comments now that we're sure the setup method works. Commit f38ab70
2. checkPlayer method in CollisionChekcer.java class was used at first when I started writing the code to check collision for the player. However, going forward, we needed more methods for different entities and objects, so I removed the checkPlayer method. Commit ee6a83d
3. In our UI class, there were many fields that were never used, so we safely removed them from the class. Also, for the images we have to be sure they're not null, so I changed the way we get our paths. Commit 1e22c3ce
4. The other refactoring In the UI class is simplifying our condition for branching. Before refactoring, we were using many unnecessary conditions in our branching. Now, it should be simpler and easier to read. Commit 232667c
5. Added a null check to UI.java: g2 variable is being checked for null before attempting to perform any operations on it. If g2 is null, methods or properties are accessed on it (such as g2.setFont() or g2.drawImage()), it would result in a NullPointerException, causing the program to crash. Commit 0e9549b
6. Removed commented-out code in GamePanel class: Some commented-out code that was used for debugging purposes has been removed. Commit 6160daf
7. Added a key input to go back to the title after winning the game: after some testing, we realized that there wasn't an option to play again after beating the game, the refactoring was done to the GamePanel, UI and KeyHandler classes. Commit 4cddb74
8. Added drawWinScreen methods in UI class: All code used to draw the winning messages and the final time have been encapsulated in one method called drawWinScreen to ensure consistency with other screen drawing methods. Commit 4cddb74

9. Simplifying branching conditions in our EventHandler.java in order to have easier to read code. Also, we cleaned up the import statement. Commit 54e194c

In conclusion, the comprehensive code review of the game's code, particularly focusing on the UI.java and TileManager.java classes, has led to significant improvements in the game's structure and functionality. Key enhancements include the reorganization of the TileManager class for greater efficiency, the removal of redundant code in the CollisionChecker.java and GamePanel classes, and the optimization of the UI class by eliminating unused fields and simplifying branching conditions. Additionally, the implementation of null checks and the introduction of new features, such as the ability to replay the game after winning and the encapsulation of screen drawing methods, have notably increased the code's robustness and user experience. These refinements not only improve the current state of the game but also lay a solid foundation for future updates and expansions, ensuring that the code remains clear, flexible, and resilient.