

Mexican Train Game Testing Document

Ignas Kamugisha

December 2, 2023

1. Introduction

This document presents a comprehensive testing report for the Mexican Train Game, a Java application I implemented to model the Mexican Domino game. The program consists of several classes, including `MexicanTrainGame`, `DominoTrain`, `Domino` and other classes imported from the Labs i.e the `LinkedList` and `LLNode`. To properly do the testing, I extensively used a JUnit class to test the modelled behaviours of the game and I also used normal observation of the Javafx elements to justify their functionality.

2. Testing Objectives

The primary objectives of the testing process were to:

- Verify the correctness of the `MexicanTrainGame` class and its associated methods.

3. Testing MexicanTrainGame Class

3.1 Setup Method

Objective: To verify that the setup method initializes the game parameters correctly.

- **Test Case 1:** Setting up a game with valid parameters.
- **Test Case 2:** Attempting to set up a game with an insufficient deck size.
- **Test Case 3:** Verifying correct initialization of players, trains, and decks.

3.2 Deal Method

Objective: To ensure that the deal method distributes dominos to players correctly.

- **Test Case 1:** Verifying each player receives the correct number of dominos.
- **Test Case 2:** Checking that dominos are removed from the game deck after dealing.

3.3 addToMexicanTrain Method

Objective: To confirm that dominos can be added to the Mexican Train correctly.

- **Test Case 1:** Adding a valid domino to the Mexican Train.
- **Test Case 2:** Attempting to add an invalid domino to the Mexican Train.

3.4 Player Actions

Objective: Testing various player actions, including drawing from the pile and playing dominos.

- **Test Case 1:** Ensuring correct behavior when a player draws from the pile.
- **Test Case 2:** Verifying that players can play dominos on their trains.
- **Test Case 3:** Testing the "Add to Mexican" button functionality.

4. Testing DominoTrain Class

4.1 Construction and Initialization

Objective: To validate the construction and initialization of the DominoTrain class.

- **Test Case 1:** Creating a DominoTrain with a valid starting value.
- **Test Case 2:** Ensuring correct initialization and rotation of dominos in the train.

4.2 addToFront Method

Objective: Verifying the behavior of the addToFront method in adding dominos to the train.

- **Test Case 1:** Adding a valid domino to the front of the train.
- **Test Case 2:** Attempting to add an invalid domino to the front of the train.

4.3 canAdd Method

Objective: Testing the canAdd method to ensure compatibility with adding dominos.

- **Test Case 1:** Verifying that a domino can be added to the train.
- **Test Case 2:** Checking if an incompatible domino can be added.

5. Testing Domino Class

5.1 Construction and Initialization

Objective: Validating the construction and initialization of the Domino class.

- **Test Case 1:** Creating a domino with valid back and front values.
- **Test Case 2:** Ensuring correct initialization and rotation of dominos.

5.2 toString Method

Objective: Verifying the correctness of the toString method for string representation.

- **Test Case 1:** Checking the string representation of a standard domino.
- **Test Case 2:** Testing the string representation after domino rotation.

5.3 equals Method

Objective: Confirming the correctness of the equals method for domino equality.

- **Test Case 1:** Comparing two equal dominos.
- **Test Case 2:** Verifying the inequality of two different dominos.

6. Conclusion

The testing process has thoroughly examined the functionality of the Mexican Train game and its associated classes. The implemented test cases cover various scenarios, ensuring the robustness and correctness of the code. The absence of critical issues indicates the reliability of the game implementation, Although there were a few failed test cases which did not significantly affect the game.

7. References

1. JUnit Documentation: <https://junit.org/junit4/>
2. JavaFX Documentation: <https://openjfx.io/javadoc/>
3. JavaFX Documentation: <https://google.com>