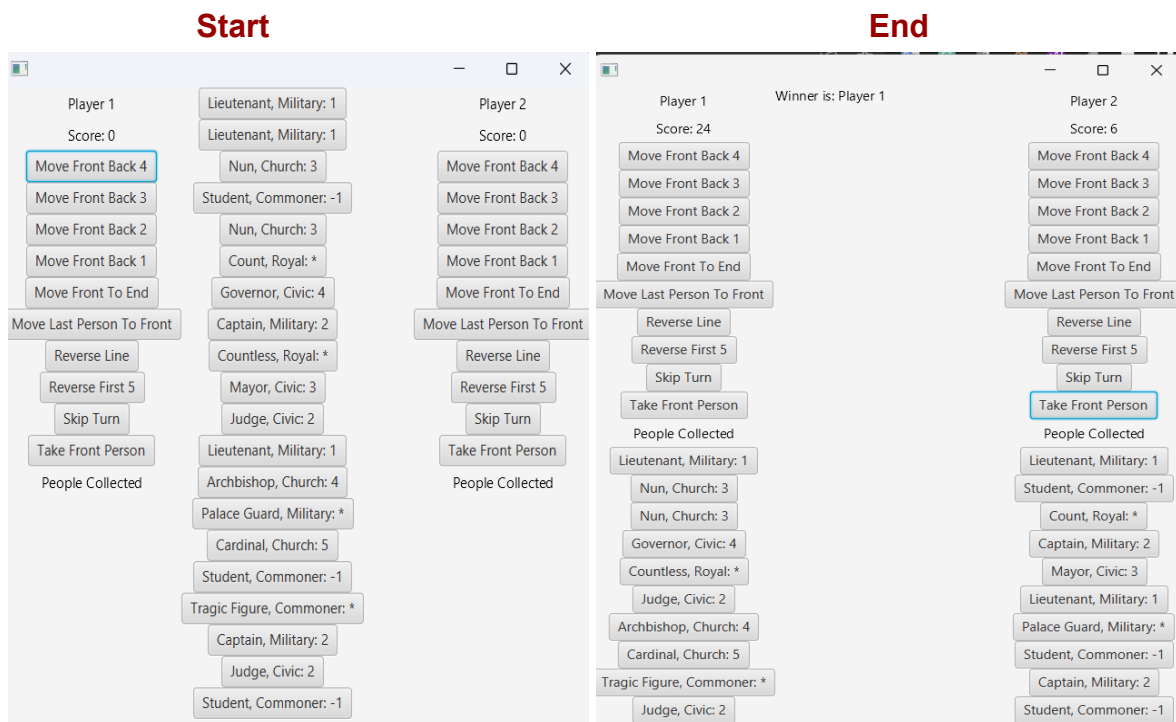


Project 4 Testing Report

A. Guillotine Class (Javafx display Test)

- ★ In this testing report, Images of the game have been used to verify that the score points work the way they should be based on the instruction and each scenario is meant to test specific special cards and contexts.
- ★ For the classes provided from the instruction, the new methods are tested.

Scenario 1: Testing score **without** using any Action cards.



Expected player 1 score = 24

Expected player 2 score = 6

Player 1 score found = 24

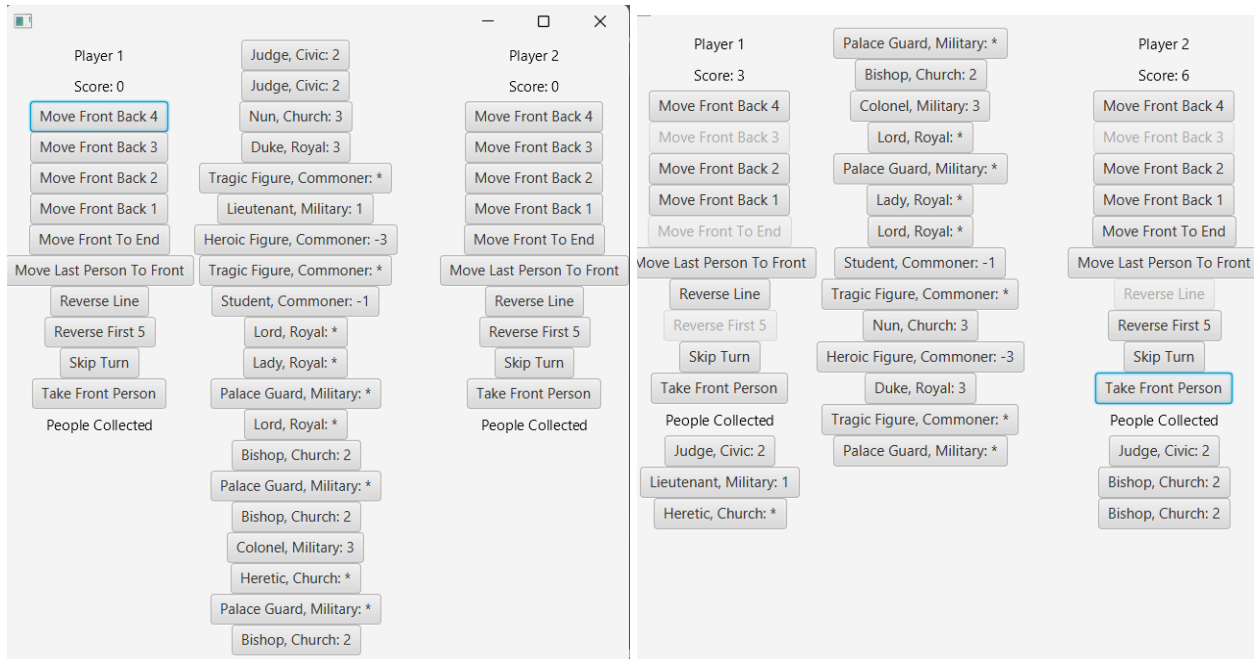
Player 2 score found = 6

Winner: player 1 (Test for winner method)

Scenario 2: Testing with Action cards

Start

End



Actions taken: take card(p1) > take card(p2) > reverse first 5, take card (p1) > reverse line, take card(p2) > move to end, take card (p1) > move back 3 , take card (p2)

Expected player 1 score = 3

Expected player 2 score = 6

Player 1 score found = 3

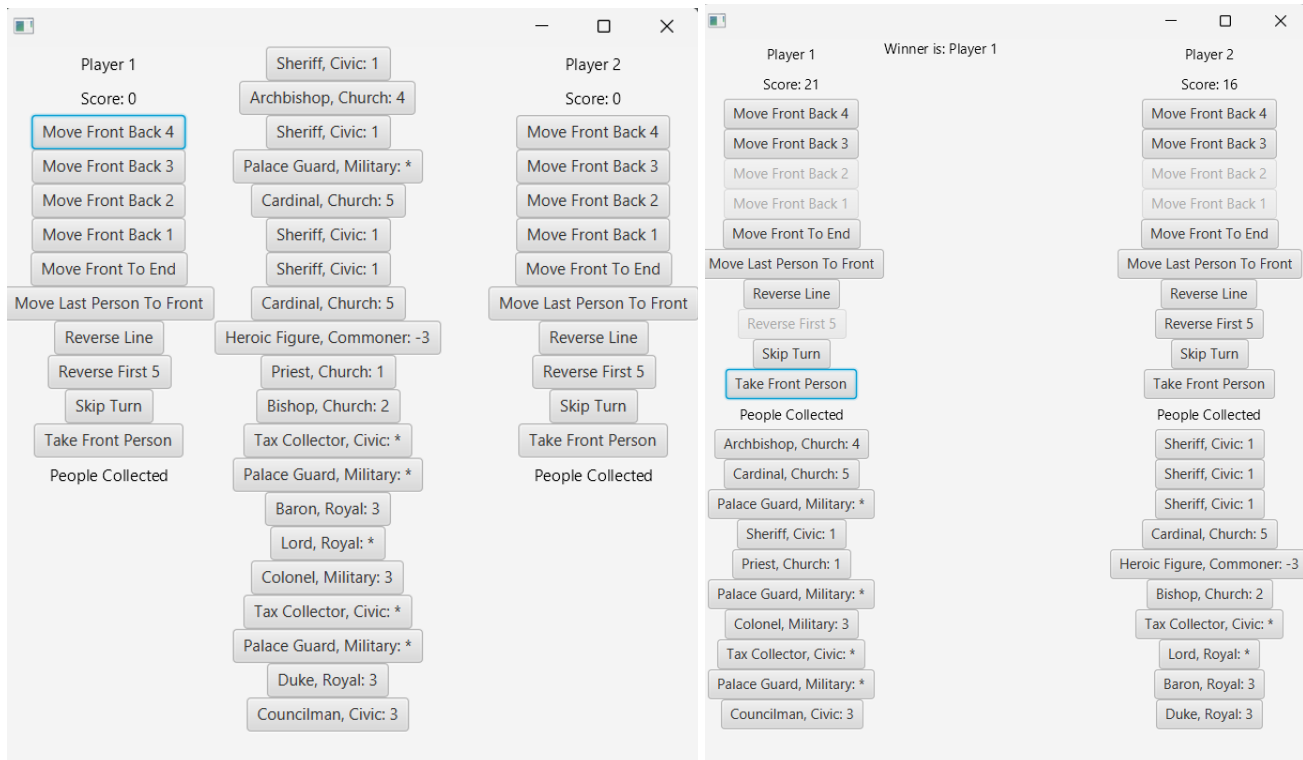
Player 2 score found = 6

Scenario 3: Testing for score points for special cards

- **Test case 1:** Testing for Palace Guard Military and Tax Collector

Start

End



Palace Guard militray: 3 palace guard military (have 3 points total for **player 1**)

Tax collector civic: 1 civic person before card collected (have 1 point for **player 1**) and 3 civics collected before tax collector card (have 3 points for **player 2**)

Expected player 1 score = 21

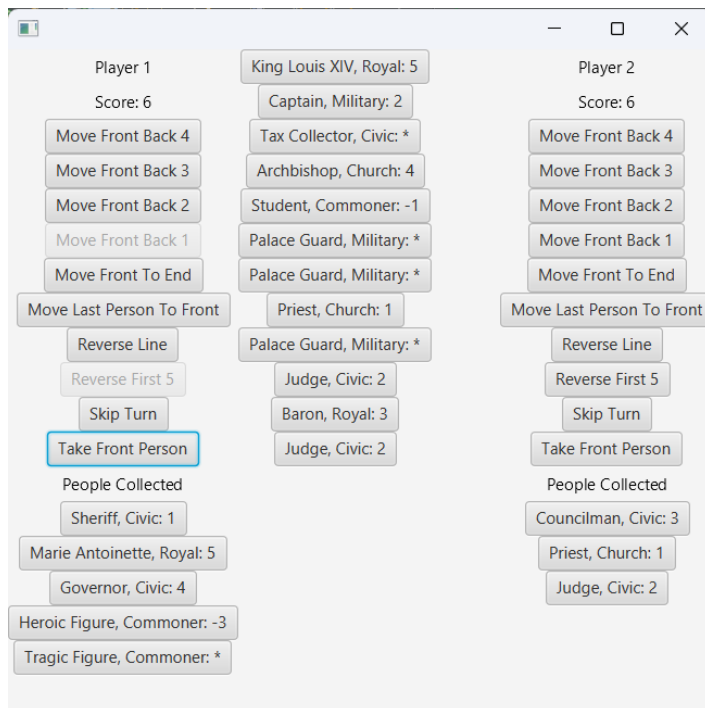
Expected player 2 score = 16

Player 1 score found = 21

Player 2 score found = 16

- **Test case 2:** Testing for **Tragic Figure** Commoner (-1 number of commoner collected)

End



Heroic Figure commoner: 1 in **player 1** (- 1 > deducted from 1+5+4-3 (-1))

Expected player 1 score = 6

Player 1 score found = 6

- **Test case 3:** Testing for Lord Loyal, Lady Royal, and Heretic Church.

Start



Lord Loyal & Lady Royal: 1 each in **player 1** (+ 4 points when the second one is collected)
Heretic Church: one nun church in **player 1** (+ 1 point added)
Expected player 1 score = 19
Expected player 1 score = 16
Player 1 score found = 16

B. LinkedList Class with JUnit test file

★ For testing this class test first middle and last have not been used since it does not have a loop and has no meaning in the case of a linked list.

1. MoveBackTester(): method which uses assertEquals tester to test moving back nodes a number of times in a linked list.

Test 0 : Tests an empty linked list

Test 1: Tests linkedlist with one node.

Test many: Tests linked list with many nodes.

2. moveFirstToLastTester(): method which uses assertEquals tester to test moving first node to last node in a linked list.

Test 0 : Tests an empty linked list

Test 1: Tests linkedlist with one node.

Test many: Tests linked list with many nodes.

3. moveLastToFirstTester(): method which uses assertEquals tester to test moving last node to first node in a linked list.

Test 0 : Tests an empty linked list

Test 1: Tests linkedlist with one node.

Test many: Tests linked list with many nodes.

3. reverseListTester(): method which uses assertEquals tester to test reversing nodes in a linked list.

Test 0 : Tests an empty linked list

Test 1: Tests linkedlist with one node.

Test many: Tests linked list with many nodes.

3. reverseListKTester(): method which uses assertEquals tester to test reversing the first k nodes in a linked list.

Test 0 : Tests an empty linked list

Test 1: Tests linkedlist with one node.

Test many: Tests linked list with many nodes.

C. Player Class with JUnit test file

playerClassTester(): method which uses assertEquals tester to test the setters and getters of the player class.

Test getName : tests the getName method

Test getPoint: tests the get point method.

Test isPlayerTurn: tests isplayerturn method.

D. Card Class with JUnit test file

cardClassTester(): method which uses assertEquals tester to test the setters and getters of the card class.

Test getName : tests the getName method

Test getPoint: tests the getPoint method.

Test haveSpecialPoint: tests the haveSpecialPoint method.

Test getGroup: tests the getGroup method.