TAFE Invaders

Test Document

Version 1.0

12/06/2020

TABLE OF CONTENTS

[1 Test Data 1](#_Toc43457224)

[2 Test Data Evidence 2](#_Toc43457225)

[3 Unit Testing 9](#_Toc43457226)

[3.1 Screenshots 9](#_Toc43457227)

[4 Encountered Issues 10](#_Toc43457228)

[5 Evaluation 11](#_Toc43457229)

[5.1 Requirements Traceability Matrix 11](#_Toc43457230)

[5.2 Conclusion 11](#_Toc43457231)

# Test Data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Module** | **Function** | **Input** | **Expected Result** | **Pass / Fail** |
| 1 | Music Player | Play | Press play button, music is stopped. | Music begins playing current track. | PASS |
| 2 | Press play button, music is playing. | Music continues to play current track. | PASS |
| 3 | Stop | Press stop button, music is stopped. | Music playback remains stopped. | PASS |
| 4 | Press stop button, music is playing. | Music stops playback. | PASS |
| 5 | Previous | Press previous button, playlist has a previous track. | The previous track becomes the current track. | PASS |
| 6 | Press previous button, playlist has no previous track. | The current track remains the same. | PASS |
| 7 | Next | Press next button, playlist has a next track. | The next track becomes the current track. | PASS |
| 8 | Press next button, playlist does not next track. | The first track becomes the next track. | PASS |
| 9 | Search | Playlist contains track with search query. | Searched track is selected and begins playing. | PASS |
| 10 | Playlist does not contain track with search query. | Error message is shown. | PASS |
| 11 | High-scores | Add a high-score | Name entered is empty. | High-score dialogue box is shown again to user. | PASS |
| 12 | Name entered is not empty. | High-score is added to the high-score table. | PASS |
| 13 | Sort by Name | Click ‘Sort by Name’ button. | High-scores are sorted in alphabetically order of names, disregarding capitalization. | PASS |
| 14 | Sort by Date | Click ‘Sort by Date’ button. | High-scores are sorted in chronological order of dates. | PASS |
| 15 | Sort by Score | Click ‘Sort by Score’ button. | High-scores are sorted in descending order of scores. | PASS |
| 16 | Gameplay | Movement | Press left arrow | Player moves left. | PASS |
| 17 | Press right arrow | Player moves right. | PASS |
| 18 | Shoot | Press spacebar | Player shoots. | PASS |
| 19 | GUI | Help Files | Click Help -> Help | Help launches in default browser. | PASS |
| 20 | Exit game | Click File -> Close | Game closes. | PASS |

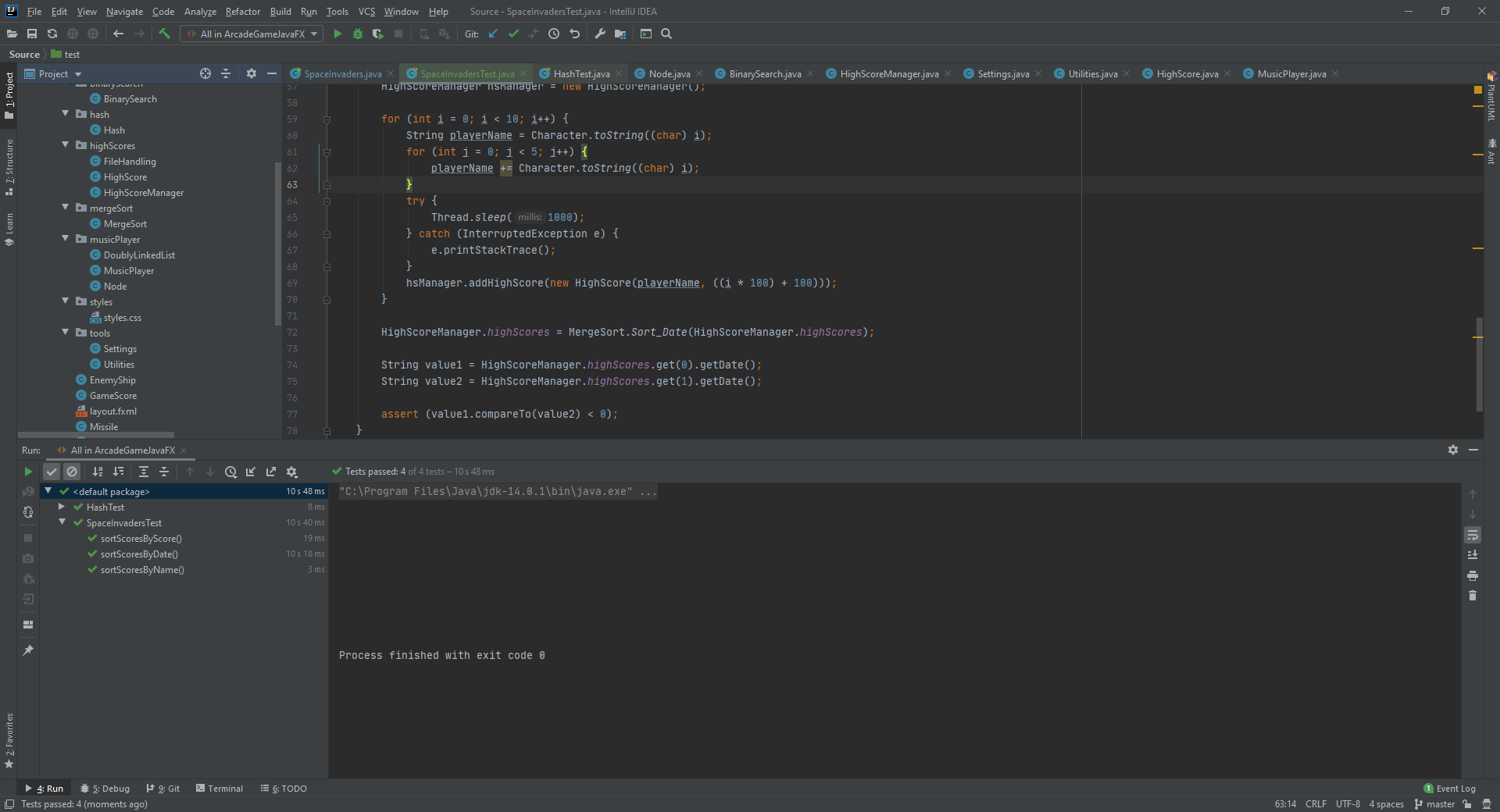
# Test Data Evidence

|  |  |
| --- | --- |
| **#** | **Screenshot** |
| 1 | No screenshot available. |
| 2 | No screenshot available. |
| 3 | No screenshot available. |
| 4 | No screenshot available. |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |
| 19 |  |
| 20 | No Screenshot available. |

# Unit Testing

## Screenshots

Below is a screenshot of all the test methods in the test package using Junit 5.4 completing successfully. The tests are found within the project source.



# Encountered Issues

The search algorithm encountered a problem during testing, where a fault was found searching for the first track in the playlist.

After inspection the code surrounding this functionality the following correction was made:

Original:

// Determine if song is found  
if (foundIndex > 0)  
{

Fixed:

// Determine if song is found  
if (foundIndex >= 0)  
{

Due to the use of the operators in the first statement, the title would be treated the same as a not found result unless the index was 1 or higher. But changing this to the >= operator, the first value is now returned as found by the method.

# Evaluation

## Requirements Traceability Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Case #** | **Defects** | **Status** |
| Must contain dynamic data structures | 1 – 10 | - | Satisfied |
| Must contain hashing techniques | 11, 12 | - | Satisfied |
| Must contain sorting algorithm | 13, 14, 15 | - | Satisfied |
| Must contain searching technique | 9, 10 | - | Satisfied |
| Must contain 3rd party library | 9, 10 | - | Satisfied |
| Must have a GUI | 1 – 18 | - | Satisfied |
| Must adhere to coding standards | 1 – 18 | - | Satisfied |
| Must have help files | 19 | - | Satisfied |

## Conclusion

After completing the test results, I have concluded this project performs is functionalities as intended, and satisfies all requirements as stated by JMC.

The requirements traceability matrix below tracks all the requirements and their relevant test cases, showing this project meets all requirements.