

GUI Testing:

To test the GUI of the application, we utilized inline-block CSS and ran the application to observe how the GUI appeared. For each button created, we first displayed the buttons and then added spacing, color, background, and size to them. We noticed that the size of the Grid did not fit the screen display, so we scaled the screen display to fit the Grid and enhance its aesthetics. We apply this process to any new features (buttons) in the new releases.

The testing approach for the GUI includes the following steps:

1. Add the features.
2. Apply styling.

Model and Controller Testing:

We conducted test suites for both the Model and Controller, using different testing approaches. For the Model, we employed unit testing, while for the Controller, we used Mocking Objects with unit testing since it integrates external sources like the database. The unit testing we used was JUnit4, and the Mocking framework was Mockito. Please note that using Mockito can be challenging based on Java versions, Gradle, and dependencies of the application. Therefore, we recommend testing for compatibility issues.

Below are the steps we followed for the Model and Controller testing:

1. Test Model with unit testing.
2. Test the Controller using Mockito.

Test Summary

16	0	0	0.320s
tests	failures	ignored	duration

100%
successful

Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.example.kami.ControllerTest	1	0	0	0.314s	100%
com.example.kami.ModelTest	15	0	0	0.006s	100%

Generated by [Gradle 8.0](#) at Mar 11, 2023, 12:01:28 AM

Test Summary

16	0	0	0.320s
tests	failures	ignored	duration

100%
successful

Packages

Classes

Package	Tests	Failures	Ignored	Duration	Success rate
com.example.kami.ControllerTest	1	0	0	0.314s	100%
com.example.kami.ModelTest	15	0	0	0.006s	100%

Generated by [Gradle 8.0](#) at Mar 11, 2023, 12:01:28 AM

Package com.example.kami.ControllerTest

all > com.example.kami.ControllerTest

1
tests

0
failures

0
ignored

0.314s
duration

100%
successful

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
GamePageTest	1	0	0	0.314s	100%

Generated by Gradle 8.0 at Mar 11, 2023, 12:01:28 AM

Class com.example.kami.ControllerTest.GamePageTest

all > com.example.kami.ControllerTest > GamePageTest

1
tests

0
failures

0
ignored

0.314s
duration

100%
successful

Tests

Test	Duration	Result
testUpdateCurrentColor	0.314s	passed

Generated by Gradle 8.0 at Mar 11, 2023, 12:01:28 AM

Package com.example.kami.ModelTest

all > com.example.kami.ModelTest

15
tests

0
failures

0
ignored

0.006s
duration

100%
successful

Classes

Class	Tests	Failures	Ignored	Duration	Success rate
AbstractBoardAndCellTest	7	0	0	0.003s	100%
RectangleCellTest	4	0	0	0.002s	100%
TriangleCellTest	4	0	0	0.001s	100%

Generated by Gradle 8.0 at Mar 11, 2023, 12:01:28 AM

Class com.example.kami.ModelTest.AbstractBoardAndCellTest

all > com.example.kami.ModelTest > AbstractBoardAndCellTest

7
tests

0
failures

0
ignored

0.003s
duration

100%
successful

Tests

Test	Duration	Result
testAbstractBoard	0s	passed
testAbstractCell	0s	passed
testGetBoardType	0s	passed
testGetCell	0.001s	passed
testGetCells	0s	passed
testGetStringColors	0.001s	passed
testMassColoring	0.001s	passed

Generated by Gradle 8.0 at Mar 11, 2023, 12:01:28 AM

Class **com.example.kami.ModelTest.RectangleCellTest**

all > [com.example.kami.ModelTest](#) > RectangleCellTest

4

tests

0

failures

0

ignored

0.002s

duration

100%

successful

Tests

Test	Duration	Result
testSetColor	0s	passed
testSetColumn	0.001s	passed
testSetPoints	0s	passed
testSetRow	0.001s	passed

Generated by [Gradle 8.0](#) at Mar 11, 2023, 12:01:28 AM

Class **com.example.kami.ModelTest.TriangleCellTest**

all > [com.example.kami.ModelTest](#) > TriangleCellTest

4

tests

0

failures

0

ignored

0.001s

duration

100%

successful

Tests

Test	Duration	Result
setPointsShouldAddPointsToList	0s	passed
testGetOrientation	0s	passed
testSetColor	0s	passed
testSetPoints	0.001s	passed

Generated by [Gradle 8.0](#) at Mar 11, 2023, 12:01:28 AM