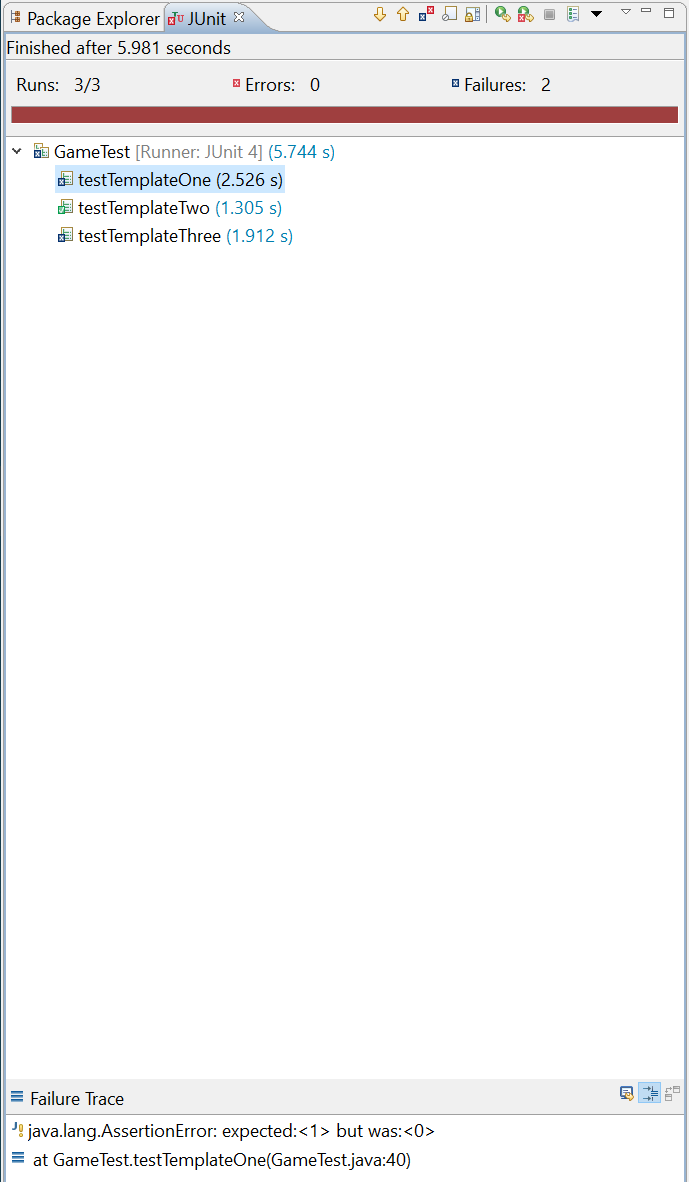
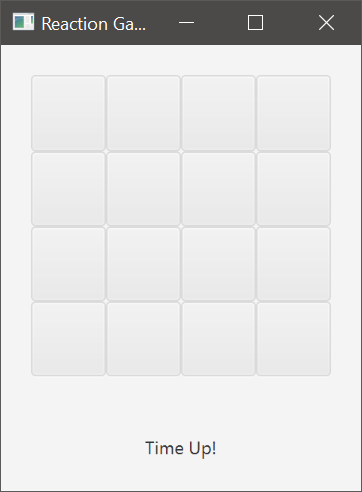
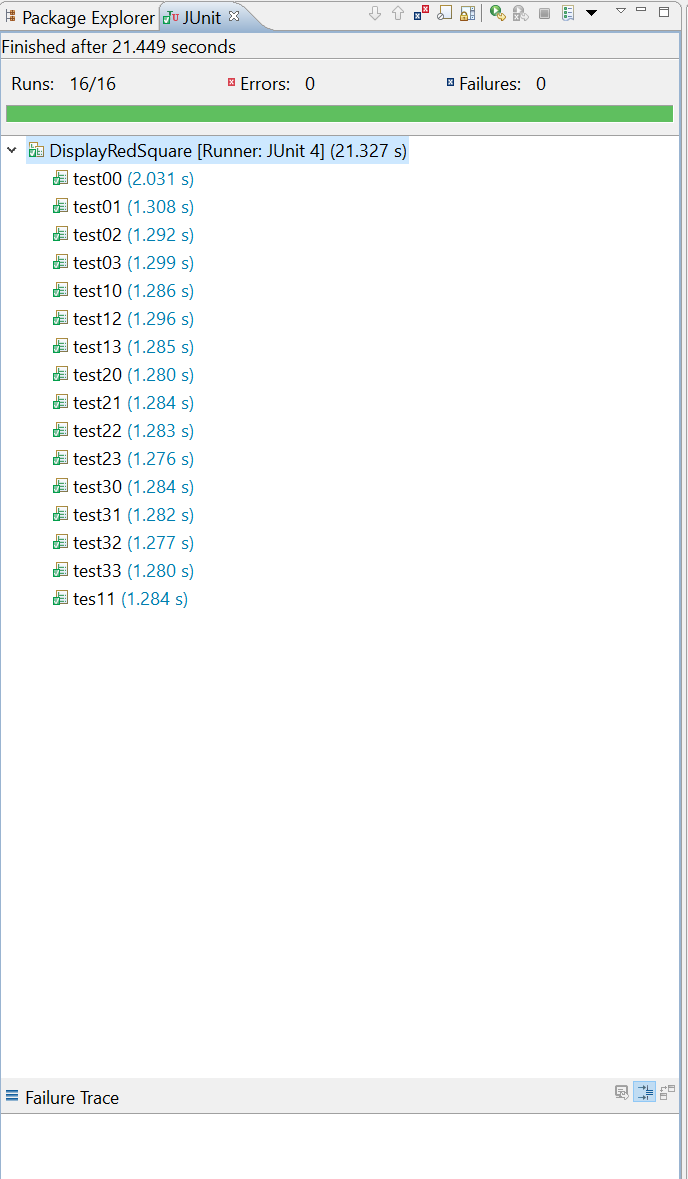
# Task 8.1



# Task 8.2



## DisplayRedSquare.java

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.fail;

import org.junit.Test;

import org.testfx.framework.junit.ApplicationTest;

import javafx.scene.control.Button;

import javafx.scene.control.Labeled;

import javafx.stage.Stage;

public class DisplayRedSquare extends ApplicationTest {

private GameApplication game;

public void start(Stage stage) throws Exception {

// Create the JavaFX application by instantiating it and

// manually calling the JavaFX start method.

game = new GameApplication();

game.start(stage);

stage.toFront();

}

public int getNumberReds() {

int numberOfReds = 0;

for (int i = 0; i < GameApplication.GRID\_SIZE \* GameApplication.GRID\_SIZE; i++) {

Button b = lookup("#button" + i).queryButton();

if(b.getStyle().equals(GameApplication.BUTTON\_STYLE\_RED)) {

numberOfReds++;

}

}

return numberOfReds;

}

@Test

public void test00() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 0;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test01() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 1;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test02() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 2;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test03() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 3;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test10() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 0;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void tes11() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 1;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test12() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 2;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test13() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 3;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test20() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 0;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test21() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 1;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test22() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 2;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test23() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 3;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test30() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 0;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test31() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 1;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test32() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 2;

game.displaySquare(x, y);

Thread.sleep(500);

assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

@Test

public void test33() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 3;

game.displaySquare(x, y);

Thread.sleep(500);

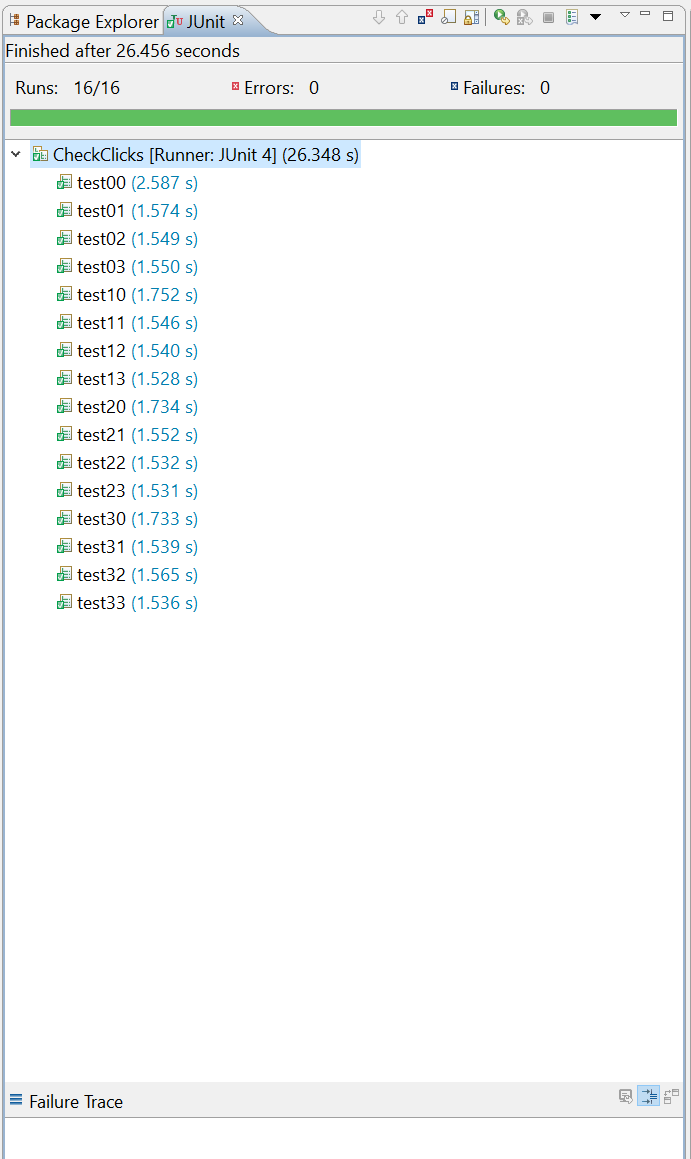
assertEquals(x, game.getDisplayedSquareX());

assertEquals(y, game.getDisplayedSquareY());

}

}

# Task 8.3



## CheckClicks.java

import static org.junit.Assert.\*;

import org.junit.Test;

import org.testfx.framework.junit.ApplicationTest;

import javafx.scene.control.Button;

import javafx.scene.control.Labeled;

import javafx.stage.Stage;

public class CheckClicks extends ApplicationTest {

private GameApplication game;

public void start(Stage stage) throws Exception {

// Create the JavaFX application by instantiating it and

// manually calling the JavaFX start method.

game = new GameApplication();

game.start(stage);

stage.toFront();

}

@Test

public void test00() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 0;

game.displaySquare(x, y);

clickOn("#button0");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test01() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 1;

game.displaySquare(x, y);

clickOn("#button4");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test02() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 2;

game.displaySquare(x, y);

clickOn("#button8");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test03() throws InterruptedException {

Thread.sleep(500);

int x = 0;

int y = 3;

game.displaySquare(x, y);

clickOn("#button12");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test10() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 0;

game.displaySquare(x, y);

clickOn("#button1");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test11() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 1;

game.displaySquare(x, y);

clickOn("#button5");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test12() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 2;

game.displaySquare(x, y);

clickOn("#button9");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test13() throws InterruptedException {

Thread.sleep(500);

int x = 1;

int y = 3;

game.displaySquare(x, y);

clickOn("#button13");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test20() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 0;

game.displaySquare(x, y);

clickOn("#button2");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test21() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 1;

game.displaySquare(x, y);

clickOn("#button6");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test22() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 2;

game.displaySquare(x, y);

clickOn("#button10");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test23() throws InterruptedException {

Thread.sleep(500);

int x = 2;

int y = 3;

game.displaySquare(x, y);

clickOn("#button14");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test30() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 0;

game.displaySquare(x, y);

clickOn("#button3");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test31() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 1;

game.displaySquare(x, y);

clickOn("#button7");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test32() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 2;

game.displaySquare(x, y);

clickOn("#button11");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

@Test

public void test33() throws InterruptedException {

Thread.sleep(500);

int x = 3;

int y = 3;

game.displaySquare(x, y);

clickOn("#button15");

Thread.sleep(500);

assertEquals(x, game.getLastClickedSquareX());

assertEquals(y, game.getLastClickedSquareY());

}

}