

Testing Report: GUI command and PC side communication testing report

Prepared by JianQiu Li , ID : 1635717

1 Introduction

In the project, the GUICommands class is used to transmit the Commands which is obtained from the mouse actions of operator. We define these commands as int values so that computer and the robot can communicate easily and it is easy for testing. This testing report is used to test the commands before sending to robot, which can ensure the robot could get the right action commands that we want. The testing of these GUI commands is very important, because with this testing report, we could analyze where the communicating problems are happen. For example, if this testing is passed, if there is a communicating problem, it could be happen in robot side. This is why this testing report is important. Thus, in order to give the right commands to the robot, this testing report will cover some commands testing cases.

In our project, the PC side will execute the methods in GUICommands class and it will send the commands to robot, so in this testing report, we predict the value of these commands, and then, we compare the predicted value of command with the values we get. If the code is right, the predicted value will corresponding with the mouse actions. This is the process of testing for the GUICommands class.

2 Testing Cases

2.1 Test Case: Emergency Stop Command in GUICommands class

Functionality: This testing case is to test the function of Emergency Stop. This testing case ensures the robot could get right command from PC side, so that the robot could have the ability of Emergency Stop at at anytime.

```
@Test
public void testStop() {
    GUICommands t1= new GUICommands();
    int stop = t1.stop();
    Assert.assertEquals(0,stop);
}
```

Testing Output: In the code of GUICommands class, The value of Emergency Stop is equal to 0, so, the predicted value is equal to 0. The output value from Eclipse testing is 0, the testing is passed.

2.2 Test Case: Movement Commands in GUICommands class

Functionality: This testing case is to test the function of movement control commands on the GUI. It is used to ensure that operators can control the robot by using commands on the GUI.

```
@Test
public void testForward() {
    GUICommands t2= new GUICommands();
    int forward = t2.forward();
    Assert.assertEquals(1,forward);
}

@Test
public void testBackward() {
    GUICommands t3= new GUICommands();
    int backward = t3.backward();
    Assert.assertEquals(2,backward);
}

@Test
public void testTurnleft() {
    GUICommands t4= new GUICommands();
    int left = t4.turnleft();
    Assert.assertEquals(3,left);
}

@Test
public void testTurnright() {
    GUICommands t5= new GUICommands();
    int right = t5.turnright();
    Assert.assertEquals(4,right);
}
```

Testing Output: In the code of GUICommands class, the number of 1,2,3,4 respectively match FORWARD, BACKWARD, TURNLEFT, TURNRIGHT. The predicted value is respectively equal to 1,2,3,4. The output value from Eclipse testing is matched, the testing is passed

2.3 Test Case: Marking closures Command in GUICommands class

Functionality: This testing case is to test the function of marking closure commands on the GUI. It is used to ensure that operators can control the robot to mark the closure.

```
@Test
public void testmarkClosure() {
    GUICommands t7= new GUICommands();
    int mark = t7.markClosure();
    Assert.assertEquals(20,mark);
}
```

Testing Output: In the code of GUICommands class, the value of marking is 20, so the predicted value is equal to 20. The output value from Eclipse testing is matched, the testing is passed.

2.4 Test Case: Disconnect Command in GUICommands class

Functionality: This testing case is to test the function of disconnect commands on the GUI. This is used to disconnect the link between PC side and robot.

```
@Test
public void testDisconnct() {
    GUICommands t8= new GUICommands();
    int disconnect = t8.disconnect();
    Assert.assertEquals(-1,disconnect);
}
```

Testing Output:In the code of GUICommands class, the value of disconnect commands is -1, so the predicted value is equal to -1. The output value from Eclipse testing is matched, the testing is passed.

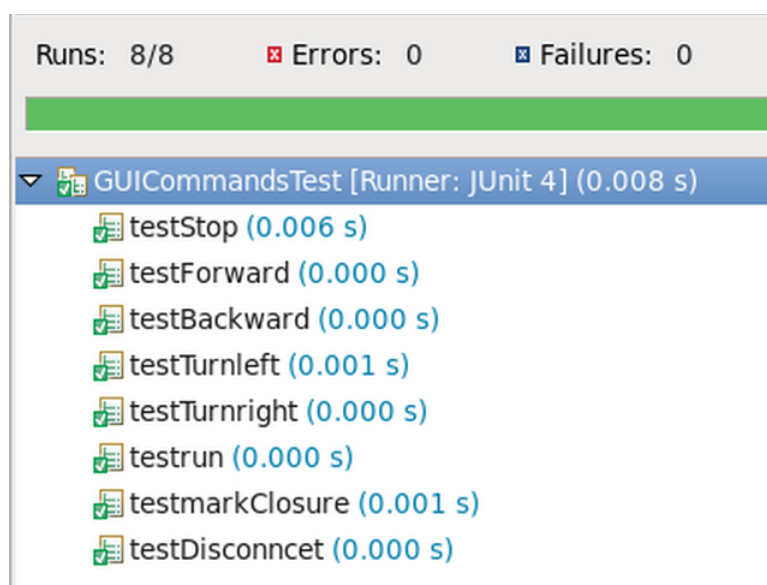
2.5 Test Case: BatteryInfo in GUICommands class

Functionality: BatteryInfo is defined as a value which is used to be transmitted between robot and PC side. This testing ensures that the transmitted value is correct.

```
@Test
public void testrun() {
    GUICommands t6= new GUICommands();
    int batteryinfo = t6.run();
    Assert.assertEquals(12,batteryinfo);
}
```

Testing Output:In the code of GUICommands class, the value of BatteryInfo commands is 12, so the predicted value is equal to 12. The output value from Eclipse testing is matched, the testing is passed.

3 Testing results from Junit



All the testing cases are passed.