Validation Report

Chinese Chess

Team3

student xxxx

Content

Remark: The following part is a mixture of Chinese and English since it is hard to make sense if I translate terms in Chinese Chess directly to English.

T1.3.1 test“士”在九宫格内移动（沿九宫格内的斜线走一格）

T1.3.2 test “士”不能移动到九宫格外

T1.4 Movement of“相”

T1.4.1 test “相”在自己半场移动，没有相脚（走“田”字）

T1.4.2 test “相”在自己半场移动，有相脚（无法行棋）

T1.4.3 test “相”不能过河移动到对面半场

T1.5 Movement of“车”

T1.5.1 test “车”的移动，没有其他棋子阻挡（沿直线向任意方向）

T1.5.2 test “车”的移动，遇到友方棋子的阻挡

T1.5.3 test “车”的移动，遇到对方棋子的阻挡（即吃子）

T1.6 Movement of“马”

T1.6.1 test “马”的移动，没有遇到马脚（走“日”字）

T1.6.2 test “马”的移动，遇到马脚（无法行棋）

T1.7 Movement of“炮”

T1.7.1 test “炮”的移动

T1.7.2 test“炮”的吃子

T2 Functional test

Introduction of FunctionalTestOption.m

Introduction of FunctionalTestRecord.m

T2.1 Use Case “求和”

T2.1.1 test(在自己回合）先取消求和，接着确认求和，求和被拒绝，最后确认求和，求和成功

T2.1.2 test (在对方回合）先取消求和，接着确认求和，求和被拒绝，最后确认求和，求和成功

T2.2 Use Case “认输”

T2.2.1 test（在自己回合，没有进行过行棋）先取消认输，接着直接认输

T2.2.2 test（在自己回合，已经进行过行棋）认输

T2.2.3 test（在对方回合,没有进行过行棋）先取消认输，接着直接认输

T2.2.4 test（在对方回合，已经进行过行棋）认输

T2.3 Use Case “悔棋”

T2.3.1 test（在自己回合）先取消悔棋，接着确认悔棋，对方拒绝，最后确认悔棋，对方同意

T2.3.2 test（在对方回合）悔棋按钮不可用

T2.4 Use Case “观看回放”

T2.4.1 test（未使用快进按钮）播放行棋记录

T2.4.2 test（使用快进按钮）播放行棋记录

T3 Integration test

Introduction of IntegrationTest.m

T3.1 test“完整棋局 1”

T3.2 test“完整棋局 2”

T3.3 test“完整棋局 3”

T3.4 test“完整棋局 4”

T3.5 test“完整棋局 5”

T4 Risk management

T4.1 Invalid movement

T4.1.1 Move to the outside of the board

T4.2.1 Move chess to invalid place that against the rule

T4.2 Invalid“悔棋”

T4.3 mistakenly “认输”或“求和”

T4.4 handle disconnect

Model checking

Code Structure

Introdunction

There are altogether 4 files for testing. UnitTestMove.m is used for unit test. FunctionalTestOption.m and FunctionalTestRecord.m is used for functional test. IntegrationTest.m is use for integration test. Each testing file start with the declare of properties ,the definition of Methods TestClassSetup,methods TestMethodSetupand methodsTestMethodTeardown. The following code is testcase, which is different in each file.

Properties

```
properties
        timekeeper
        board
end
```

Methods TestClassSetup

TestClassSetup will be called only once when the test class is created

```
methods(TestClassSetup)
function Init(~)
            close all force;
            clear;
            clc;
            addpath("imgs/");
            addpath("Model/");
            addpath("ViewModel/");
            addpath("timer/");
            addpath("recordings/");
end
end
```

Methods TestMethodSetup

TestMethodSetup will be called whenever a testcase is created

```
methods(TestMethodSetup)
function createFigure(testcase)
            close all force;
            testcase.timekeeper=ChessTimer;
            testcase.board=ChessBoard(testcase.timekeeper);
end
end
```

MethodsTestMethodTeardown

TestMethodSetup will be called whenever a testcase is destroyed

```
methods(TestMethodTeardown)
function closeFigure(testcase)
if ~isempty(timerfind)
                stop(timerfind);
                delete(timerfind);
end
            close all force;
            clear;
            clc;
end
end
```

T1 Unit test

Introduction of UnitTestMove.m

Unit Test is Implemented in file UnitTestMove.m

This file mainly test if the movement of the chess is valid and correct.

The testcase is numbered according to classification.

case 10x 兵

case 20x 将

case 30x 士

case 40x 相

case 50x 车

case 60x 马

case 70x 炮

The test will done a sequence of movement of chess, some are valid ,while others are invalid. It will mainly use verifyEqual to test the if the result is right. It requires 2 click to complete the move. First select a chess that is being moved, then select a new position that want it to move to. If the movement is valid, the previous position will be blank and the new position will be filled with the moved chess. If the movement failed, the previous position will contain the chess we want to move , the new position will contain its original chess(or blank)..

The information of the position of chess is store in testcase.board.Red.Game, so

testcase.board.Red.Game.pos_x_y can get the chess currently in row x column y. Each chess has its corresponding number 14 for red“兵”，13 forred“炮”,12 for red“车” ,11 for red“马”, 10 for red“相”， 9 for red“士” , 8 for red“将” , 7 for black“兵”, 6 for black“炮”, 5 for black“车”, 4 for black“马”, 3 for black“相”, 2 for black “士”,1 for black“将” ,0 for blank space.
So testcase.verifyEqual(testcase.board.Content(5,7),int16(14))means to check if chess in row 5 column 7 is red “兵”

All the test will be ended with red“认输”to end the program, otherwise it will affect later testcase.

T1.1 Movement of“兵”

T1.1.1 test “兵”在过河前的移动（只能向前一格）

Test code:

```
function case101(testcase) %兵过河之前的移动,可以向前，不能向左右后
%HumanHuman Mode
           pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Invalid move,left
%red move bing 5-7->4-7
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_4_7);
           testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
           testcase.verifyEqual(testcase.board.Content(4,7),int16(0));
%Invalid move,right
%red move bing 5-7->6-7
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_6_7);
```

```
            testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
            testcase.verifyEqual(testcase.board.Content(6,7),int16(0));
%Invalid move,back
%red move bing 5-7->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
%Valid move,forward
%red move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Invalid move,left
%black move bing 5-7->4-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(6,4),int16(0));
%Invalid move,right
%black move bing 5-7->6-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_7);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(4,4),int16(0));

%Invalid move,back
%black move bing 5-7->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(0));


%Valid move,forward
```

```
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));



%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);

end
```

```
Test result:
  TestResult - 属性:

          Name: 'UnitTestMove/case101'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 23.2343
       Details: [1×1 struct]

总计:
   1 Passed, 0 Failed, 0 Incomplete.
```

23.2343 秒测试时间。

T1.1.2 test “兵”过河之后的移动（可以向前、左、右一格）

Test code:

```
function case102(testcase) %兵过河之后的移动，可以向左右前，不能向后
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(7,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(14));

%Valid move,forward
%black move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(3,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(7));

%Valid move,forward
%red move bing 7-6->7-5
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_5);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(14));

%Valid move,forward
%black move bing 7-6->7-5
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_5);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(3,6),int16(7));

%Invalid move,back
%red move bing 7-5->7-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(7,5),int16(14));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));

%Invalid move,back
%black move bing 7-5->7-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(3,6),int16(7));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));

%Valid move,left
%red move bing 7-5->6-5
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_5);
            testcase.verifyEqual(testcase.board.Content(7,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,5),int16(14));

%valid move,left
%black move bing 7-5->6-5
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_5);
            testcase.verifyEqual(testcase.board.Content(3,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,6),int16(7));

%Valid move,right
%red move bing 6-5->7-5
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_5);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_5);
            testcase.verifyEqual(testcase.board.Content(6,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(14));
```

```
%valid move,right
%black move bing 6-5->7-5
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_5);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_5);
            testcase.verifyEqual(testcase.board.Content(4,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,6),int16(7));

%Valid move,forward
%red move bing 7-5->7-4
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_4);
            testcase.verifyEqual(testcase.board.Content(7,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,4),int16(14));

%valid move,forward
%black move bing 7-5->7-4
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_5);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_4);
            testcase.verifyEqual(testcase.board.Content(3,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,7),int16(7));



%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'UnitTestMove/case102'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 47.1909

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

47.1909 秒测试时间。

T1.2 Movement of “将”

T1.2.1 test “将”在九宫格内移动（可以向前、后、左、右一格）

Test code:

```
function case201(testcase) %将在九宫格内移动
%HumanHuman Mode
           pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move jiang 5-10->5-9
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_10);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_9);
           testcase.verifyEqual(testcase.board.Content(5,10),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,9),int16(8));

%Valid move,forward
%black move jiang 5-10->5-9
           pause(0.5);
```

```
            testcase.press(testcase.board.Black.Game.pos_5_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(1));

%Valid move,left
%red move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,9),int16(8));

%Valid move,left
%black move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,2),int16(1));

%Valid move,forward
%red move jiang 4-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(4,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(8));

%Valid move,forawrd
%black move jiang 4-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(6,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(1));

%Valid move,right
%red move jiang 4-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(8));

%Valid move,right
%black move jiang 4-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(1));

%Valid move,right
%red move jiang 5-8->6-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,8),int16(8));

%Valid move,right
%black move jiang 5-8->6-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,3),int16(1));

%Valid move,right
%red move jiang 6-8->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(6,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(8));

%Valid move,right
%black move jiang 6-8->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(4,3),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(4,2),int16(1));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case201'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 32.5568
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

32.5568 秒测试时间。

T1.2.2 test “将”不能移动到九宫格外

Test code:

```
function case202(testcase) %将不能移动到九宫格外
```

```
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Valid move,forward
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));

%Valid move,forward
%red move jiang 5-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(8));

%Valid move,forward
%black move jiang 5-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(1));

%Valid move,forward
%red move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(4,9),int16(8));

%Valid move,left
%black move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,2),int16(1));

%Invalid move,left
%red move jiang 4-9->3-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(4,9),int16(8));
            testcase.verifyEqual(testcase.board.Content(3,9),int16(0));

%Invalid move,left
%black move jiang 4-9->3-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(6,2),int16(1));
            testcase.verifyEqual(testcase.board.Content(7,2),int16(0));

%Valid move,right
%red move jiang 4-9->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(8));

%Valid move,right
%black move jiang 4-9->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(1));
```

```
%Valid move
%red move jiang 5-9->5-8
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_9);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,8),int16(8));

%Valid move
%black move jiang 5-9->5-8
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_5_9);
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_5_8);
           testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,3),int16(1));

%InValid move
%red move jiang 5-8->5-7
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           testcase.verifyEqual(testcase.board.Content(5,8),int16(8));
           testcase.verifyEqual(testcase.board.Content(5,7),int16(0));

%Valid move
%black move jiang 5-8->5-7
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_5_8);
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_5_7);
           testcase.verifyEqual(testcase.board.Content(5,3),int16(1));
           testcase.verifyEqual(testcase.board.Content(5,4),int16(0));

%Valid move
%red move jiang 5-8->5-9
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_9);
           testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,9),int16(8));

%Valid move
%black move jiang 5-8->5-9
```

```
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(1));

%Valid move
%red move jiang 5-9->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(8));

%Valid move
%black move jiang 5-9->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,2),int16(1));

%InValid move
%red move jiang 6-9->7-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_9);
            testcase.verifyEqual(testcase.board.Content(6,9),int16(8));
            testcase.verifyEqual(testcase.board.Content(7,9),int16(0));

%InValid move
%black move jiang 6-9->7-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_9);
            testcase.verifyEqual(testcase.board.Content(4,2),int16(1));
            testcase.verifyEqual(testcase.board.Content(3,2),int16(0));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'UnitTestMove/case202'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 44.4953

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

44.4953 秒测试时间。

T1.3 Movement of “士”

T1.3.1 test “士”在九宫格内移动（沿九宫格内的斜线走一格）

Test code:

```
function case301(testcase) %士在九宫格内移动
%HumanHuman Mode
            pause(0.5);
```

```
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));

%Valid move
%black move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));

%Valid move
%black move shi 5-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));

%Valid move
%red move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));
```

```
%Valid move
%black move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,8),int16(9));

%Valid move
%black move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,3),int16(2));


%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case301'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 41.1851
       Details: [1×1 struct]
```

总计：

1 Passed, 0 Failed, 0 Incomplete.

41.1851 秒测试时间。

T1.3.2 test “士”不能移动到九宫格外

Test code:

```
function case302(testcase) %士不能移动到九宫格外
%HumanHuman Mode
           pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move bing 3-7->3-6
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_3_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_3_6);
           testcase.verifyEqual(testcase.board.Content(3,7),int16(0));
           testcase.verifyEqual(testcase.board.Content(3,6),int16(14));

%Valid move,forward
%black move bing 3-7->3-6
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_3_7);
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_3_6);
           testcase.verifyEqual(testcase.board.Content(7,4),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(7,5),int16(7));

%Valid move,forward
%red move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Valid move,forward
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));

%Valid move,forward
%red move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(7,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(14));

%Valid move,forward
%black move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(3,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(7));

%InValid move
%red move shi 4-10->3-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(9));
            testcase.verifyEqual(testcase.board.Content(3,9),int16(0));
```

```
%InValid move
%black move shi 4-10->3-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(2));
            testcase.verifyEqual(testcase.board.Content(7,2),int16(0));

%Valid move
%red move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));

%Valid move
%black move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));

%Valid move
%black move shi 5-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));

%Valid move
%red move shi 4-8->3-7
```

```
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_7);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(3,7),int16(0));

%Valid move
%black move shi 4-8->3-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_7);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));

%Valid move
%red move shi 4-8->5-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));

%Valid move
%black move shi 4-8->5-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));

%Valid move
%red move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));

%Valid move
%black move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
```

```
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,8),int16(9));

%Valid move
%black move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,3),int16(2));

%InValid move
%red move shi 6-8->7-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_7);
            testcase.verifyEqual(testcase.board.Content(6,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(7,7),int16(0));

%InValid move
%black move shi 6-8->7-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_7);
            testcase.verifyEqual(testcase.board.Content(4,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(3,4),int16(0));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case302'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 45.9677
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

45.9677 秒测试时间。

T1.4 Movement of “相”

T1.4.1 test “相”在自己半场移动，没有相脚（走“田”字）

Test code:

```
function case401(testcase) %相在自己半场，没有相脚
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
```

```
%red move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(7,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,8),int16(10));

%Valid move
%black move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(3,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,3),int16(3));

%Valid move
%red move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(9,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));

%Valid move
%black move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(1,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));

%Valid move
%red move xiang 7->6->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(10));

%Valid move
%black move xiang 7->6->5-8
            pause(0.5);
```

```
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(3));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

```
Test result:
  TestResult - 属性:

          Name: 'UnitTestMove/case401'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 21.6565
       Details: [1×1 struct]

总计:
   1 Passed, 0 Failed, 0 Incomplete.
   21.6565 秒测试时间。
```

T1.4.2 test “相”在自己半场移动，有相脚（无法行棋）

Test code:

```
function case402(testcase) %相在自己半场，有相脚
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move pao 8-8->8->9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_9);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,9),int16(13));

%Valid move
%black move pao 8-8->8->9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_9);
            testcase.verifyEqual(testcase.board.Content(2,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,2),int16(6));

%InValid move
%red move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(7,10),int16(10));
            testcase.verifyEqual(testcase.board.Content(9,8),int16(0));

%InValid move
%black move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(3,1),int16(3));
            testcase.verifyEqual(testcase.board.Content(1,3),int16(0));

%Valid move
%red move pao 8-9->8->8
```

```
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(8,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,8),int16(13));

%Valid move
%black move pao 8-9->8->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(2,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,3),int16(6));

%Valid move
%red move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(7,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,8),int16(10));

%Valid move
%black move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(3,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,3),int16(3));

%Valid move
%red move pao 8-8->8->7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_7);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,7),int16(13));

%Valid move
%black move pao 8-8->8->7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
```

```
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_7);
            testcase.verifyEqual(testcase.board.Content(2,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,4),int16(6));

%InValid move
%red move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(9,8),int16(10));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));

%InValid move
%black move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(1,3),int16(3));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));

%Valid move
%red move pao 8-7->8->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(8,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,8),int16(13));

%Valid move
%black move pao 8-7->8->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(2,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,3),int16(6));

%Valid move
%red move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
```

```
            testcase.verifyEqual(testcase.board.Content(9,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));

%Valid move
%black move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(1,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

```
Test result:
  TestResult - 属性:

          Name: 'UnitTestMove/case402'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 38.2244
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

38.2244 秒测试时间。

T1.4.3 test “相”不能过河移动到对面半场

Test code:

```
function case403(testcase) %相不能过河到对面半场
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move bing 1-7->1-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_1_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_1_6);
            testcase.verifyEqual(testcase.board.Content(1,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,6),int16(14));

%Valid move,forward
%black move bing 1-7->1-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_1_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_1_6);
            testcase.verifyEqual(testcase.board.Content(9,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,5),int16(7));

%Valid move,forward
%red move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Valid move,forward
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
```

```
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));

%Valid move
%red move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(7,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,8),int16(10));

%Valid move
%black move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(3,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,3),int16(3));

%Valid move
%red move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(9,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));

%Valid move
%black move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(1,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));

%InValid move
%red move xiang 7-6->9-4
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_4);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));
```

```
            testcase.verifyEqual(testcase.board.Content(9,4),int16(0));

%InValid move
%black move xiang 7-6->9-4
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_4);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));
            testcase.verifyEqual(testcase.board.Content(1,7),int16(0));

%InValid move
%red move xiang 7-6->5-4
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_4);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));

%InValid move
%black move xiang 7-6->5-4
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_4);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));

%Valid move
%red move xiang 7->6->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(10));

%Valid move
%black move xiang 7->6->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(3));
```

```
%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

```
Test result:
  TestResult - 属性:

          Name: 'UnitTestMove/case403'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 39.3739
       Details: [1×1 struct]
```

总计:

```
   1 Passed, 0 Failed, 0 Incomplete.
   39.3739 秒测试时间。
```

T1.5 Movement of “车”

T1.5.1 test “车”的移动，没有其他棋子阻挡（沿直线向任意方向）

Test code:

```
function case501(testcase) %车没有其他子阻挡
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move ju 9->10->9-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(9,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,9),int16(12));

%Valid move
%black move ju 9->10->9-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(1,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,2),int16(5));

%Valid move
%red move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(9,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(12));

%Valid move
%black move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(1,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,2),int16(5));


%Valid move
%red move ju 6->9->6-6
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(6,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,6),int16(12));

    %Valid move
    %black move ju 6->9->6-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(4,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,5),int16(5));

    %Valid move
    %red move ju 6->6->2-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_2_6);
            testcase.verifyEqual(testcase.board.Content(6,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,6),int16(12));

    %Valid move
    %black move ju 6->6->2-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_2_6);
            testcase.verifyEqual(testcase.board.Content(4,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,5),int16(5));

    %red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
    end
```

Test result:

TestResult - 属性:

```
        Name: 'UnitTestMove/case501'
      Passed: 1
      Failed: 0
  Incomplete: 0
    Duration: 24.3690
     Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

24.369 秒测试时间。

T1.5.2 test “车”的移动，遇到友方棋子的阻挡

Test code:

```
function case502(testcase) %车遇到友方棋子的阻挡
%HumanHuman Mode
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%InValid move
%red move ju 9->10->8-10
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_9_10);
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_8_10);
          testcase.verifyEqual(testcase.board.Content(9,10),int16(12));
          testcase.verifyEqual(testcase.board.Content(8,10),int16(11));

%Valid move
```

```
%black move ju  9->10->8-10
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_10);
            testcase.verifyEqual(testcase.board.Content(1,1),int16(5));
            testcase.verifyEqual(testcase.board.Content(2,1),int16(4));

%InValid move
%red move ju 9->10->9-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_7);
            testcase.verifyEqual(testcase.board.Content(9,10),int16(12));
            testcase.verifyEqual(testcase.board.Content(9,7),int16(14));

%Valid move
%black move ju  9->10->9-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_7);
            testcase.verifyEqual(testcase.board.Content(1,1),int16(5));
            testcase.verifyEqual(testcase.board.Content(1,4),int16(7));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case502'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 14.7462
       Details: [1×1 struct]
```

总计：

1 Passed, 0 Failed, 0 Incomplete.

14.7462 秒测试时间。

T1.5.3 test “车”的移动，遇到对方棋子的阻挡（即吃子）

Test code:

```
function case503(testcase) %车遇到对方棋子的阻挡（吃子）
%HumanHuman Mode
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move ju 9->10->9-9
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_9_10);
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_9_9);
          testcase.verifyEqual(testcase.board.Content(9,10),int16(0));
          testcase.verifyEqual(testcase.board.Content(9,9),int16(12));

%Valid move
%black move ju 9->10->9-9
          pause(0.5);
          testcase.press(testcase.board.Black.Game.pos_9_10);
          pause(0.5);
          testcase.press(testcase.board.Black.Game.pos_9_9);
          testcase.verifyEqual(testcase.board.Content(1,1),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(1,2),int16(5));

    %Valid move
    %red move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(9,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(12));

    %Valid move
    %black move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(1,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,2),int16(5));

    %Valid move
    %red move ju 6-9->6-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(6,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,1),int16(12));

    %Valid move
    %black move ju 6-9->6-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(4,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,10),int16(5));

    %Valid move
    %red move ju 6-1->7-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_1);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,1),int16(12));
```

```
%Valid move
%black move ju 6-1->7-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_1);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,10),int16(5));

%Valid move
%red move ju 7-1->7-4
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_4);
            testcase.verifyEqual(testcase.board.Content(7,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,4),int16(12));

%Valid move
%black move ju 7-1->7-4
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_4);
            testcase.verifyEqual(testcase.board.Content(3,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,7),int16(5));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case503'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 30.0935
       Details: [1×1 struct]
```

总计：

1 Passed, 0 Failed, 0 Incomplete.

30.0935 秒测试时间。

T1.6 Movement of “马”

T1.6.1 test “马”的移动，没有遇到马脚（走“日”字）

Test code:

```
function case601(testcase) %马没有遇到马脚
%HumanHuman Mode
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move ma 8-10->7-8
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_8_10);
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_7_8);
          testcase.verifyEqual(testcase.board.Content(8,10),int16(0));
          testcase.verifyEqual(testcase.board.Content(7,8),int16(11));

%Valid move
%black move ma 8-10->7-8
          pause(0.5);
          testcase.press(testcase.board.Black.Game.pos_8_10);
```

```
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            testcase.verifyEqual(testcase.board.Content(2,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));

%Valid move
%red move ma 7-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(7,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(11));

%Valid move
%black move ma 7-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(4));

%Valid move
%red move ma 5-9->4-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,7),int16(11));

%Valid move
%black move ma 5-9->4-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,4),int16(4));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case601'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 22.2228
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

22.2228 秒测试时间。

T1.6.2 test “马”的移动，遇到马脚（无法行棋）

Test code:

```
function case602(testcase) %马遇到马脚
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move ma 8-10->7-8
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.pos_8_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            testcase.verifyEqual(testcase.board.Content(8,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,8),int16(11));

%Valid move
%black move ma 8-10->7-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            testcase.verifyEqual(testcase.board.Content(2,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));

%InValid move
%red move ma 7-8->9-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(7,8),int16(11));
            testcase.verifyEqual(testcase.board.Content(9,9),int16(0));

%InValid move
%black move ma 7-8->9-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));
            testcase.verifyEqual(testcase.board.Content(1,2),int16(0));

%InValid move
%red move ma 7-8->9-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_7);
            testcase.verifyEqual(testcase.board.Content(7,8),int16(11));
            testcase.verifyEqual(testcase.board.Content(9,7),int16(14));

%InValid move
%black move ma 7-8->9-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
```

```
            testcase.press(testcase.board.Black.Game.pos_9_7);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));
            testcase.verifyEqual(testcase.board.Content(1,4),int16(7));

%InValid move
%red move ma 7-8->8-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_6);
            testcase.verifyEqual(testcase.board.Content(7,8),int16(11));
            testcase.verifyEqual(testcase.board.Content(8,6),int16(0));

%InValid move
%black move ma 7-8->8-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_6);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));
            testcase.verifyEqual(testcase.board.Content(2,5),int16(0));

%InValid move
%red move ma 7-8->6-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(7,8),int16(11));
            testcase.verifyEqual(testcase.board.Content(6,6),int16(0));

%InValid move
%black move ma 7-8->6-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));
            testcase.verifyEqual(testcase.board.Content(4,5),int16(0));


%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case602'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 27.0131
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

27.0131 秒测试时间。

T1.7 Movement of “炮”

T1.7.1 test “炮”的移动

Test code:

```
function case701(testcase) %炮的移动
%HumanHuman Mode
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);
```

```
%Valid move
%red move pao 8-8->4-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(13));

%Valid move
%black move ju 8-8->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(2,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(6));

%Valid move
%red move pao 4-8->4-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_6);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,6),int16(13));

%Valid move
%black move ju 4-8->4-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_6);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,5),int16(6));

%Valid move
%red move pao 4-6->8-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_6);
            testcase.verifyEqual(testcase.board.Content(4,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,6),int16(13));

%Valid move
```

```
%black move ju 4-6->8-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_6);
            testcase.verifyEqual(testcase.board.Content(6,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,5),int16(6));

%Valid move
%red move pao 8-6->8-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(8,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,8),int16(13));

%Valid move
%black move ju 8-6->8-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            testcase.verifyEqual(testcase.board.Content(2,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,3),int16(6));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

```
          Name: 'UnitTestMove/case701'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 27.0449
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

27.0449 秒测试时间。

T1.7.2 test “炮”的吃子

Test code:

```
function case702(testcase)
%HumanHuman Mode
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move pao 8-8->8-1
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_8_8);
          pause(0.5);
          testcase.press(testcase.board.Red.Game.pos_8_1);
          testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
          testcase.verifyEqual(testcase.board.Content(8,1),int16(13));

%Valid move
%black move pao 8-8->8-1
          pause(0.5);
          testcase.press(testcase.board.Black.Game.pos_8_8);
          pause(0.5);
          testcase.press(testcase.board.Black.Game.pos_8_1);
          testcase.verifyEqual(testcase.board.Content(2,3),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(2,10),int16(6));

    %Valid move
    %red move pao 8-1->6-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(8,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,1),int16(13));

    %Valid move
    %black move pao 8-1->6-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(2,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,10),int16(6));

    %Valid move
    %red move pao 6-1->9-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_1);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,1),int16(13));

    %Valid move
    %black move pao 6-1->9-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_1);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,10),int16(6));

    %red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
```

```
        pause(0.5);
end
```

```
Test result:
  TestResult - 属性:

          Name: 'UnitTestMove/case702'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 23.2087
       Details: [1×1 struct]

总计:
   1 Passed, 0 Failed, 0 Incomplete.
   23.2087 秒测试时间。
```

T2 Functional test

FunctionalTest isImplemented in file FunctionalTestOptional.m and FunctionalTestRecord.m.

Introduction of FunctionalTestOption.m

FunctionalTestOptional.m mainly test if the three option,“认输”、“求和”、“悔棋” canbehave correctly.

For simplicity, some testcase will only cover one color(black or red)of the chess. Thecorrectness is guaranteed since the black chess and red chess use the same code and behaveexactly the same.

The testcase is numbered according to classification

case 10x surrender（认输）

case 20x draw（求和）

case 30x regret （悔棋）

The test will done a sequence of request of “认输”、“求和”、“悔棋” and a sequence of movement as well. It will mainly use verifyEqual to test the if the result is right. In addition to the test of movement, which is stated clearly in Unit test, it also test the correct appearance of the message

boxes.testcase.board.Red.Game.Name.Text will get the message in message box called Name in board of red,testcase.board.Black.Game.Name.Text will get the message in message box called Name in board of black.So testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认输吗") will check if the text in message box called ConfirmMessage will show "您确定要认输吗". By checking the message, we can check whether the request of “认输”、“求和”、“悔棋”succeed or fail and then compare it to our desired result.

Some of the test in “悔棋” will be ended with red“认输”to end the program, otherwise it will affect later testcase.

Introduction of FunctionalTestRecord.m

FunctionalTestRecord.mmainly test if the function of "观看回放" will work well.

The test can only be partly automatic, the part of selecting recording file can only be done manually since it is controlled by the operating system,instead of Matlab.

There are only 2 testcase,but actually we have manually test all of the testcase in Presentatin.m,UnitTestMove.m,FunctionalTestOption.m and IntegrationTest.m. So actuallymany testcase has been performed.
The testcase is numbered according toclassification

case 10x record showed without fast forward

case 20x record showed with fast forward

T2.1 Use Case “求和”

T2.1.1 test (在自己回合）先取消求和，接着确认求和，求和被拒绝，最后确认求和，求和成功

Test code:

```
function case201(testcase)%(在自己回合）取消求和+确认求和，求和被拒绝+确认求和，求和成功
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Normal Move
%black move bing 3-7->3-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_6);
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(7));
%Draw
%red draw cancel
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmCancel);
```

```
%Draw
%black draw confirm
%reject by red
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了求和");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmCancel);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);

%Draw
%red draw confirm
%reject by black
            pause(2);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了求和");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"平局");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"平局");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(2);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case201'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 29.1789

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

29.1789 秒测试时间。

T2.1.2 test (在对方回合）先取消求和，接着确认求和，求和被拒绝，最后确认求和，求和成功

Test code:

```
function case202(testcase) %(在对方回合）取消求和+确认求和，求和被拒绝+确认求和，求和成功
            pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_8_8);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Draw
%red draw cancel
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmCancel);

%Draw
%black draw confirm
%reject by red
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了求和");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmCancel);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);

%Draw
%red draw confirm
%reject by black
            pause(2);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了求和");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
```

```
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"平局");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"平局");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(2);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case202'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 24.6049

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

24.6049 秒测试时间。

T2.2 Use Case “认输”

T2.2.1 test （在自己回合，没有进行过行棋）先取消认输，接着直接认输

Test code:

```
function case101(testcase)%取消投降+直接投降（在自己回合,不进行其他移动）
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%red surrendar canceled
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认
输吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmCancel);

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认
输吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您已认输
");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方已认
输");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case101'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 10.5602

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

10.5602 秒测试时间。

T2.2.2 test（在自己回合，已经进行过行棋）认输

Test code:

```
function case102(testcase)%在自己回合投降（已经进行过行棋）
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Normal Move
%black move bing 3-7->3-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_6);
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(7));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
```

```
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认
输吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您已认输
");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方已认
输");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case102'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 14.5779

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

14.5779 秒测试时间。

T2.2.3 test （在对方回合,没有进行过行棋）先取消认输，接着直接认输

Test code:

```
function case103(testcase) %取消投降+直接投降（在对方回合,不进行其他移动）
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%red surrendar canceled
            pause(0.5);
            testcase.press(testcase.board.Black.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"您确定要
认输吗");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmCancel);

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Black.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"您确定要
认输吗");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"您已认输
");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"对方已认输
");
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case103'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 25.4623

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

25.4623 秒测试时间。

T2.2.4 test（在对方回合，已经进行过行棋）认输

Test code:

```
function case104(testcase)%在对方回合投降（已经进行过行棋）
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认
输吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您已认输
");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方已认
输");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case104'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 10.7932

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

10.7932 秒测试时间。

T2.3 Use Case “悔棋”

T2.3.1 test （在自己回合）先取消悔棋，接着确认悔棋，对方拒绝，最后确认悔棋，对方同意

Test code:

```
function case301(testcase) %（在自己回合）取消悔棋+确认悔棋，对方拒绝+确认悔棋，对方同意

            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Regret
%red regret cancel
            pause(2);
            testcase.press(testcase.board.Red.Game.RegretRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要悔
棋吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmCancel);

%black move ma 8-10->7-8
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_8_10);
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_7_8);
            testcase.verifyEqual(testcase.board.Content(2,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,3),int16(4));

%Normal Move
%red move bing 5-7->5-6
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));
```

```
%Regret
%red regret confirm
%reject by black
            pause(2);
            testcase.press(testcase.board.Red.Game.RegretRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要悔
棋吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了悔棋");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmCancel);
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);

%Normal Move
%Black move pao 2-8->5-8
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_2_8);
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(6));

%Normal Move
%red move bing 1-7->1-6
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_1_7);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_1_6);
            testcase.verifyEqual(testcase.board.Content(1,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,6),int16(14));

%Regret
%red regret confirm
%accept by red
            pause(2);
            testcase.press(testcase.board.Red.Game.RegretRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要悔
棋吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
```

```
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了悔棋");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case301'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 56.1409

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

56.1409 秒测试时间。

T2.3.2 test （在对方回合）悔棋按钮不可用

Test code:

```
function case302(testcase) %（在对方回合）悔棋按钮不可用
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Regret
%black cannot regret,there will be no message box
            pause(2);
            testcase.press(testcase.board.Black.Game.RegretRequest);

testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Visible),"off");

testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Visible),"off");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Visible),"off");

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestOption/case302'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 15.9331

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

15.9331 秒测试时间。

T2.4 Use Case “观看回放”

T2.4.1 test（未使用快进按钮）播放行棋记录

Test code:

```
function case101(testcase) % record shoued without fast forward
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.PlayRecButton);
% faster
          testcase.press(testcase.board.log.player.fastforward);
          pause(60);
%exit
          testcase.press(testcase.board.log.player.exit);
          pause(0.5);

end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestRecord/case101'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 72.1365

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

72.1365 秒测试时间。

T2.4.2 test（使用快进按钮）播放行棋记录

Test code:

```
function case201(testcase) % record shoued with fast forward
          pause(0.5);
          testcase.press(testcase.board.Start.StartPageUI.PlayRecButton);
% faster
          testcase.press(testcase.board.log.player.fastforward);
          pause(20);
%exit
          testcase.press(testcase.board.log.player.exit);
          pause(0.5);
end
```

Test result:

TestResult - 属性:

Name: 'FunctionalTestRecord/case201'

Passed: 1

Failed: 0

Incomplete: 0

Duration: 31.1900

Details: [1×1 struct]

总计:

1 Passed, 0 Failed, 0 Incomplete.

31.19 秒测试时间。

T3 Integration test

Integration Test isImplemented in fileIntegrationTest.m.

Introduction of IntegrationTest.m

IntegrationTest.m. is used for Integration test

It test 5 whole complete game,each complete game is a combination of unit test and functional test.The 5 cases numbered from 1 to 5.Case 1 and case 2 test normally ended game,which means one playereated the other player's "将" and wined. Case 3 and case 4 is ended because of one player "认输",so the other player automatically win. Case 5 is ended with a draw,because one player has agreed the otherplayer's draw request.

T3.1 test “完整棋局 1”

Test code:

```
function case1(testcase)
           pause(2);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Normal Move
%red move pao 8-8->5-8
           pause(5);
           testcase.press(testcase.board.Red.Game.pos_8_8);
           pause(2);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,8),int16(13));

%Normal Move
%black move bing 3-7->3-6
           pause(2);
           testcase.press(testcase.board.Black.Game.pos_3_7);
           pause(2);
           testcase.press(testcase.board.Black.Game.pos_3_6);
```

```
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(7));

%Normal Move
%red move pao 5-8->5-4
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_4);
            testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,4),int16(13));

%Normal Move
%black move shi 6-10->5-9
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_6_10);
            pause(2);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Normal Move
%red move pao 5-4->5-1
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_4);
            pause(2);
            testcase.press(testcase.board.Red.Game.pos_5_1);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,1),int16(13));

%Red victory
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"胜利");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"失败");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);


end
```

Test result:

TestResult - 属性:

```
          Name: 'IntegrationTest/case1'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 37.1580
       Details: [1×1 struct]
```

总计:
   1 Passed, 0 Failed, 0 Incomplete.
   37.158 秒测试时间。

T3.2 test“完整棋局 2”

Test code:

```
function case2(testcase)
           pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Invalid move,left
%red move bing 5-7->4-7
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_4_7);
           testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
           testcase.verifyEqual(testcase.board.Content(4,7),int16(0));
%Invalid move,right
%red move bing 5-7->6-7
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_6_7);
           testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
           testcase.verifyEqual(testcase.board.Content(6,7),int16(0));
%Invalid move,back
%red move bing 5-7->5-8
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_7);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_8);
           testcase.verifyEqual(testcase.board.Content(5,7),int16(14));
           testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
%Valid move,forward
%red move bing 5-7->5-6
           pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Invalid move,left
%black move bing 5-7->4-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(6,4),int16(0));
%Invalid move,right
%black move bing 5-7->6-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_7);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(4,4),int16(0));

%Invalid move,back
%black move bing 5-7->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(7));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(0));


%Valid move,forward
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));
%Valid move
%red move pao 8-8->8-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_1);
```

```
            testcase.verifyEqual(testcase.board.Content(8,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,1),int16(13));

%Valid move
%black move pao 8-8->8-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_1);
            testcase.verifyEqual(testcase.board.Content(2,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,10),int16(6));

%Valid move
%red move pao 8-1->6-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_8_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(8,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,1),int16(13));

%Valid move
%black move pao 8-1->6-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_8_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            testcase.verifyEqual(testcase.board.Content(2,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,10),int16(6));

%Valid move
%red move pao 6-1->9-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_1);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,1),int16(13));

%Valid move
%black move pao 6-1->9-1
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_1);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_1);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,10),int16(6));
```

```
%Valid move
%red move pao 9-1->5-1
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_1);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_1);
            testcase.verifyEqual(testcase.board.Content(9,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,1),int16(13));

%Red victory
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"胜利");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"失败");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);

end
```

Test result:

```
          TestResult - 属性:

          Name: 'IntegrationTest/case2'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 35.4913
       Details: [1×1 struct]

总计:
   1 Passed, 0 Failed, 0 Incomplete.
   35.4913 秒测试时间。
```

T3.3 test“完整棋局 3”

Test code:

```
function case3(testcase)
%HumanHuman Mode
            pause(0.5);
```

```
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move,forward
%red move bing 3-7->3-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_6);
            testcase.verifyEqual(testcase.board.Content(3,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,6),int16(14));

%Valid move,forward
%black move bing 3-7->3-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_6);
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,5),int16(7));

%Valid move,forward
%red move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,6),int16(14));

%Valid move,forward
%black move bing 5-7->5-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_6);
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,5),int16(7));

%Valid move,forward
%red move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(7,7),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(14));
```

```
%Valid move,forward
%black move bing 7-7->7-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_7);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(3,4),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(7));

%InValid move
%red move shi 4-10->3-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(9));
            testcase.verifyEqual(testcase.board.Content(3,9),int16(0));

%InValid move
%black move shi 4-10->3-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_9);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(2));
            testcase.verifyEqual(testcase.board.Content(7,2),int16(0));

%Valid move
%red move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));

%Valid move
%black move shi 4-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->4-8
```

```
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));

    %Valid move
    %black move shi 5-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));

    %Valid move
    %red move shi 4-8->3-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_3_7);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(3,7),int16(0));

    %Valid move
    %black move shi 4-8->3-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_3_7);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(7,4),int16(0));

    %Valid move
    %red move shi 4-8->5-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_7);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(5,7),int16(0));

    %Valid move
    %black move shi 4-8->5-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
```

```
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_7);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(5,4),int16(0));

%Valid move
%red move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(9));

%Valid move
%black move shi 4-8->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(2));

%Valid move
%red move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,8),int16(9));

%Valid move
%black move shi 5-9->6-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,3),int16(2));

%InValid move
%red move shi 6-8->7-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_7);
```

```
            testcase.verifyEqual(testcase.board.Content(6,8),int16(9));
            testcase.verifyEqual(testcase.board.Content(7,7),int16(0));

%InValid move
%black move shi 6-8->7-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_7);
            testcase.verifyEqual(testcase.board.Content(4,3),int16(2));
            testcase.verifyEqual(testcase.board.Content(3,4),int16(0));


%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要认
输吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您已认输
");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方已认
输");
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(0.5);

end
```

Test result:

```
       ans =

  TestResult - 属性:

         Name: 'IntegrationTest/case3'
       Passed: 1
       Failed: 0
   Incomplete: 0
     Duration: 46.0298
      Details: [1×1 struct]
```

```
总计:
   1 Passed, 0 Failed, 0 Incomplete.
   46.0298 秒测试时间。
```

T3.4 test“完整棋局 4”

Test code:

```
function case4(testcase)
%HumanHuman Mode
           pause(0.5);
           testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move ma 8-10->7-8
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_8_10);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_7_8);
           testcase.verifyEqual(testcase.board.Content(8,10),int16(0));
           testcase.verifyEqual(testcase.board.Content(7,8),int16(11));

%Valid move
%black move ma 8-10->7-8
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_8_10);
           pause(0.5);
           testcase.press(testcase.board.Black.Game.pos_7_8);
           testcase.verifyEqual(testcase.board.Content(2,1),int16(0));
           testcase.verifyEqual(testcase.board.Content(3,3),int16(4));

%Valid move
%red move ma 7-8->5-9
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_7_8);
           pause(0.5);
           testcase.press(testcase.board.Red.Game.pos_5_9);
           testcase.verifyEqual(testcase.board.Content(7,8),int16(0));
           testcase.verifyEqual(testcase.board.Content(5,9),int16(11));

%Valid move
%black move ma 7-8->5-9
           pause(0.5);
```

```
            testcase.press(testcase.board.Black.Game.pos_7_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(3,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(4));

%Valid move
%red move ma 5-9->4-7
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,7),int16(11));

%Valid move
%black move ma 5-9->4-7
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_7);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,4),int16(4));

%Valid move,forward
%red move jiang 5-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,9),int16(8));

%Valid move,forward
%black move jiang 5-10->5-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            testcase.verifyEqual(testcase.board.Content(5,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,2),int16(1));

%Valid move,left
%red move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_9);
            pause(0.5);
```

```
            testcase.press(testcase.board.Red.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,9),int16(8));

%Valid move,left
%black move jiang 5-9->4-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            testcase.verifyEqual(testcase.board.Content(5,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,2),int16(1));

%Valid move,forward
%red move jiang 4-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(4,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,8),int16(8));

%Valid move,forawrd
%black move jiang 4-9->4-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            testcase.verifyEqual(testcase.board.Content(6,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,3),int16(1));

%Valid move,right
%red move jiang 4-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(4,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(8));

%Valid move,right
%black move jiang 4-8->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_4_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(6,3),int16(0));
```

```
            testcase.verifyEqual(testcase.board.Content(5,3),int16(1));

%Valid move,right
%red move jiang 5-8->6-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,8),int16(8));

%Valid move,right
%black move jiang 5-8->6-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            testcase.verifyEqual(testcase.board.Content(5,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,3),int16(1));

%Valid move,right
%red move jiang 6-8->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(6,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(8));

%Valid move,right
%black move jiang 6-8->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(4,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,2),int16(1));

%red surrender success
            pause(0.5);
            testcase.press(testcase.board.Red.Game.DefeatRequest);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
```

```
            pause(0.5);


end
```

Test result:

```
          TestResult - 属性:

          Name: 'IntegrationTest/case4'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 44.6754
       Details: [1×1 struct]

总计:
   1 Passed, 0 Failed, 0 Incomplete.
   44.6754 秒测试时间。
```

T3.5 test “完整棋局 5”

Test code:

```
function case5(testcase)
            pause(0.5);
            testcase.press(testcase.board.Start.StartPageUI.HumanButton);

%Valid move
%red move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            testcase.verifyEqual(testcase.board.Content(7,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,8),int16(10));

%Valid move
%black move xiang 7-10->9->8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
```

```
            testcase.verifyEqual(testcase.board.Content(3,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,3),int16(3));

%Valid move
%red move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(9,8),int16(0));
            testcase.verifyEqual(testcase.board.Content(7,6),int16(10));

%Valid move
%black move xiang 9-8->7->6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_8);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            testcase.verifyEqual(testcase.board.Content(1,3),int16(0));
            testcase.verifyEqual(testcase.board.Content(3,5),int16(3));

%Valid move
%red move xiang 7->6->5-8
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(7,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,8),int16(10));

%Valid move
%black move xiang 7->6->5-8
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_7_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_5_8);
            testcase.verifyEqual(testcase.board.Content(3,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(5,3),int16(3));

%Valid move
%red move ju 9->10->9-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(9,10),int16(0));
            testcase.verifyEqual(testcase.board.Content(9,9),int16(12));
```

```
%Valid move
%black move ju 9->10->9-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_10);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            testcase.verifyEqual(testcase.board.Content(1,1),int16(0));
            testcase.verifyEqual(testcase.board.Content(1,2),int16(5));

%Valid move
%red move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(9,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,9),int16(12));

%Valid move
%black move ju 9->9->6-9
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_9_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            testcase.verifyEqual(testcase.board.Content(1,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,2),int16(5));


%Valid move
%red move ju 6->9->6-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(6,9),int16(0));
            testcase.verifyEqual(testcase.board.Content(6,6),int16(12));

%Valid move
%black move ju 6->9->6-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_9);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_6);
            testcase.verifyEqual(testcase.board.Content(4,2),int16(0));
            testcase.verifyEqual(testcase.board.Content(4,5),int16(5));
```

```
%Valid move
%red move ju 6->6->2-6
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_6_6);
            pause(0.5);
            testcase.press(testcase.board.Red.Game.pos_2_6);
            testcase.verifyEqual(testcase.board.Content(6,6),int16(0));
            testcase.verifyEqual(testcase.board.Content(2,6),int16(12));

%Valid move
%black move ju 6->6->2-6
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_6_6);
            pause(0.5);
            testcase.press(testcase.board.Black.Game.pos_2_6);
            testcase.verifyEqual(testcase.board.Content(4,5),int16(0));
            testcase.verifyEqual(testcase.board.Content(8,5),int16(5));

            pause(2);
            testcase.press(testcase.board.Red.Game.DrawRequest);
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"您确定要求
和吗");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmConfirm.Text),"确定");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmCancel.Text),"取消");
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"对方提出
了求和");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmConfirm.Text),"同意");
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmCancel.Text),"拒绝");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            testcase.verifyEqual(string(testcase.board.Black.Game.ConfirmMessage.Text),"平局");
            testcase.verifyEqual(string(testcase.board.Red.Game.ConfirmMessage.Text),"平局");
            pause(2);
            testcase.press(testcase.board.Black.Game.ConfirmConfirm);
            pause(2);
            testcase.press(testcase.board.Red.Game.ConfirmConfirm);
            pause(2);
end
```

Test result:

TestResult - 属性:

```
          Name: 'IntegrationTest/case5'
        Passed: 1
        Failed: 0
    Incomplete: 0
      Duration: 44.7853
       Details: [1×1 struct]
```

总计:

1 Passed, 0 Failed, 0 Incomplete.

44.7853 秒测试时间。

T4 Risk management

T4.1 Invalid movement

T4.1.1 Move to the outside of the board

Point outside the chessboard is unable to click, so such movement is unable to happen. It is unable to test automatically since no position corresponding to area outside the board. But we have tested it manually.

T4.1.2 Move chess to invalid place that against the rule

Many unit test including T1.1.1,T1.1.2,T1.2.2,T1.3.2,T1.4.2,T1.4.3, T1.5.3,T1.6.2,T1.7.2 has cover the invalid movement of “将”“士”“相”“车”“马”“炮”.All of the testcase has passed indicates that the invalid move that against the law will not occur.

T4.2 Invalid “悔棋”

In oppnent’s turn “悔棋” will be invalid since it is your opponent turn to move chess. Which is verified using T2.3.2, in this testcase, the button will be disabled.

T4.3 mistakenly “认输”或“求和”

When pressing “认输”or“求和”, there will first be a message box to

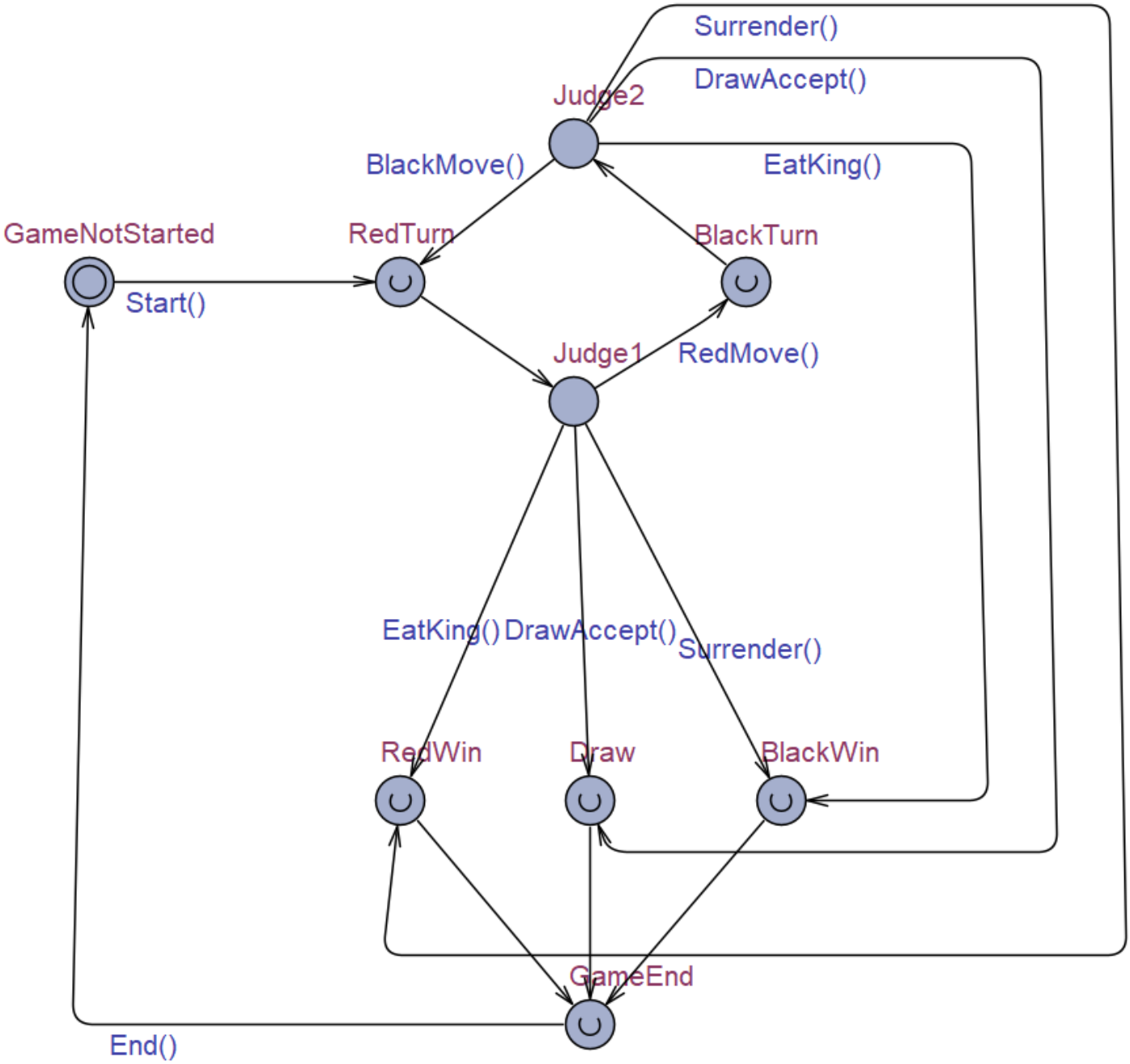
let you confirm whether you really want to “认输”or“求和”.So you can avoid mistakenly press“认输” or“求和”to lose game or draw. T2.1.1,T2.1.2,T2.2.1,T2.2.3 has veritied that.

T4.4 handle disconnect

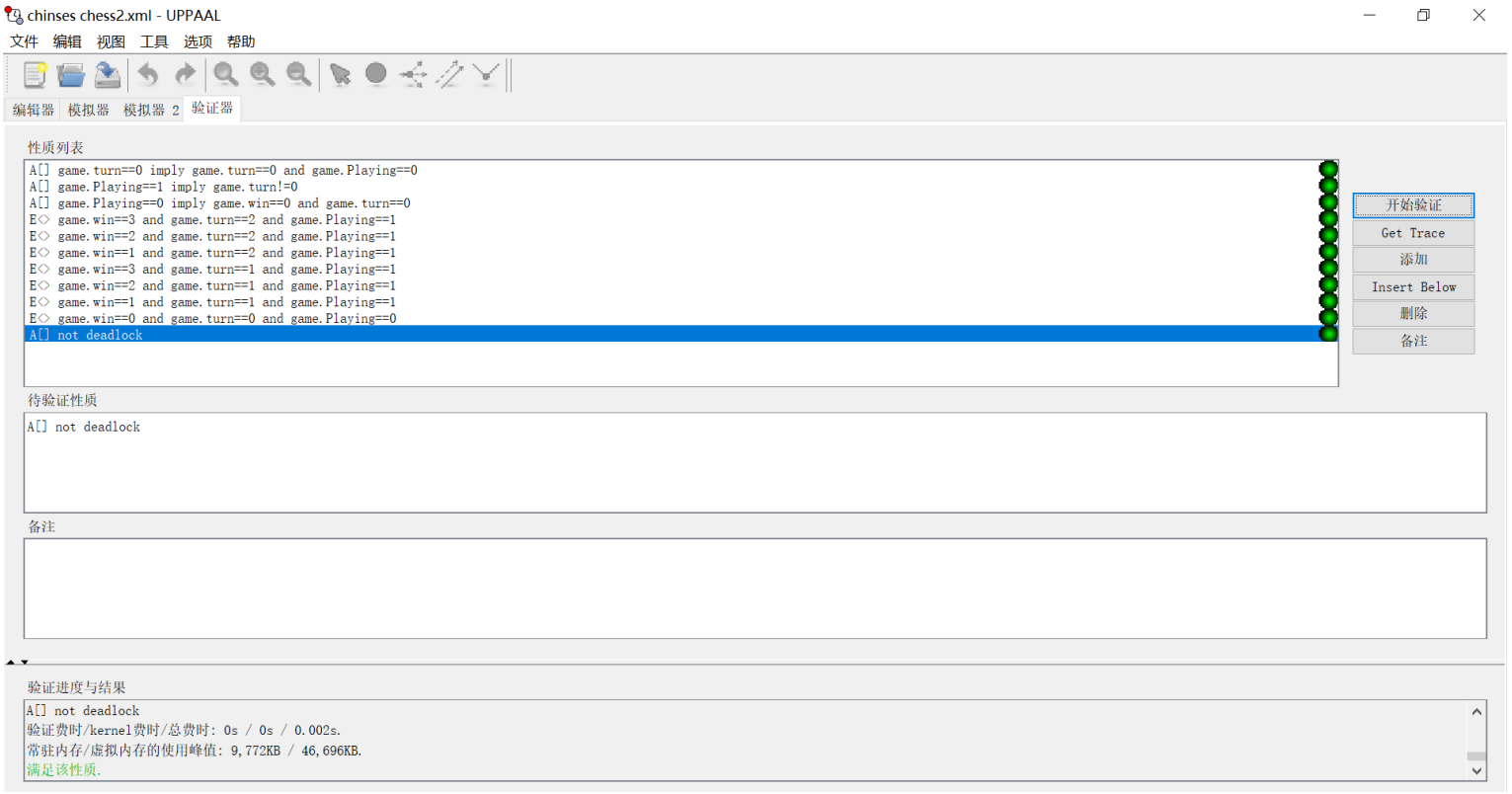
Disconnect may happen in online version. We only use manual test to test it since it envolved the testing of 2 devices, which is hard to implement using Matlab. We have set a time limit. If is unable to connect to the server, then the message box will throw an error to indicate the failture of connect.So that the connect will not last forever.

Model Checking

An uppaal model is built for Chinese chess.



Justify the properties



See more detail in Chinese Chess.xml.