

测试报告

version 1.0.0

修订历史

日期	版本	作者	描述
8月7日	1.0.0	黄嘉敏、 谢明、 许立子、 鲁升纲	实现 Jumper 类基本功能

目录

- 1 测试计划.....4
- 2 用例 1.....4
 - 2.1 测试用例.....4
 - 2.2 测试结果.....4
 - 2.3 结果分析.....4
- 3 用例 2.....5
 - 3.1 测试用例.....5
 - 3.2 测试结果.....5
 - 3.3 结果分析.....5

1 测试计划

/**

* 给出测试计划，说明需要进行哪些测试，并说明原因

*/

测试计划	原因
1.测试 Jumper 的基本跳转、转向（包括遇到障碍时的行为模式）功能	这是 Jumper 的基本功能
2.测试一个 Jumper 与其他 Actor 相向，并且下一步都想移动到同一个位置的情况	应当考虑特殊情况，在具体运行的时候不是多线程而是有先后运行顺序，在测试的时候也应当考虑

2 用例 1

/**

* 在此描述测试目的

*/

测试目的：测试 Jumper 的行为模式，测试 Jumper 的基本跳转、转向、在遇到石头障碍时的行为模式，是否能跳跃到 Flower 上等

2.1 测试用例

/**

* 在此描述测试用例和预期结果

*/

测试 Jumper 的基本跳转、转向、在遇到石头障碍时的行为模式，是否能跳跃到 Flower 上等



如图所示，有 a,b,c 三个 Jumper，r1,r2 两个 Rock，以及 f 一个 Flower。

a 的初始方向设定为西北，b 和 c 的初始方向设定为北方。

a 的初始位置是 (2, 0)

b 的初始位置是 (2, 1)

c 的初始位置是 (2, 2)

r1 的初始位置是 (0, 2)

r2 的初始位置是 (1, 3)

f 的初始位置是 (0, 4)

```
a.setDirection(Location.NORTHWEST);
b.setDirection(Location.NORTH);
c.setDirection(Location.NORTH);
world.add(new Location(2,0), a);
world.add(new Location(2,1), b);
world.add(new Location(2,2), c);
world.add(new Location(0,2), r1);
world.add(new Location(1,3), r2);
world.add(new Location(0,4), f);
```

a, b, c 运行一次之后的预期结果是：

a 的位置为 (2, 0)，方向为北方

b 的位置为 (0, 1)，方向为北方

c 的位置为 (2, 2)，方向为东北方

```
assert (gra != null);
assertEquals(2, a.getLocation().getRow());
assertEquals(0, a.getLocation().getCol());
assertEquals(Location.NORTH, a.getDirection());
assert (grb != null);
assertEquals(0, b.getLocation().getRow());
assertEquals(1, b.getLocation().getCol());
assertEquals(0, b.getDirection());
assert (grc != null);
assertEquals(2, c.getLocation().getRow());
assertEquals(2, c.getLocation().getCol());
assertEquals(45, c.getDirection());
```

然后 c 再运行三次（先跳到花的位置，再转向，最后离开）

预期结果是花消失了

```
int i = 0;
for(; i < 3; i++)
{
    c.act();
}
Grid<Actor> gr = world.getGrid();
Actor whatAFuck = gr.get(new Location(0,4));
assertEquals(true, whatAFuck == null);
```

2.2 测试结果

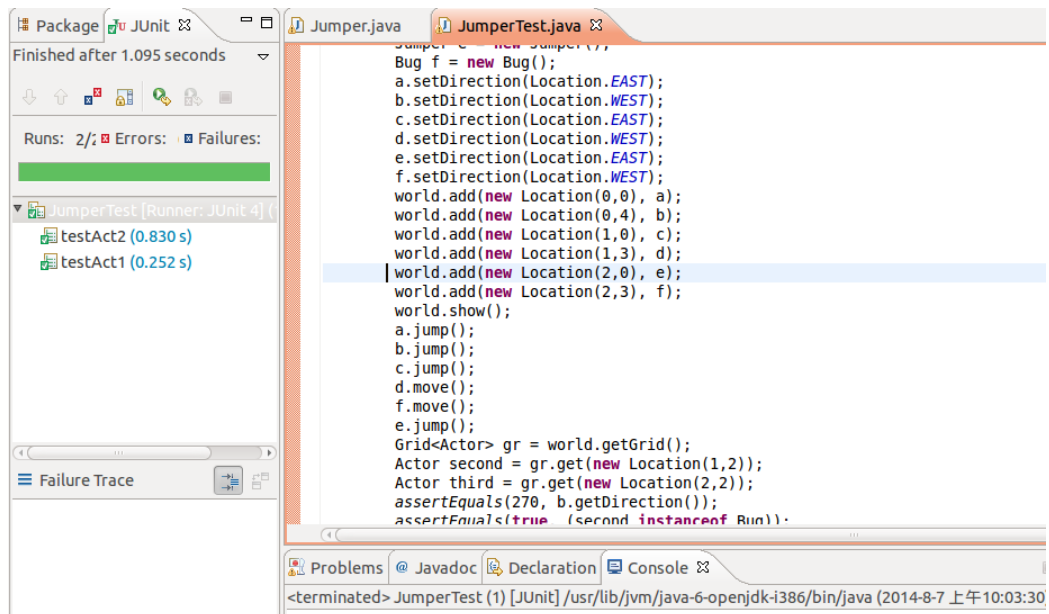
/**

* 在此描述测试结果，包括系统运行的截图

*/

测试结果与预期相符。

系统运行截图如下：



2.3 结果分析

由测试结果可知，结果与预期结果一致。

Jumper 可以正常跳跃，不会跳到无效的位置，不会跳出 world，可以跳到 flower 上面。

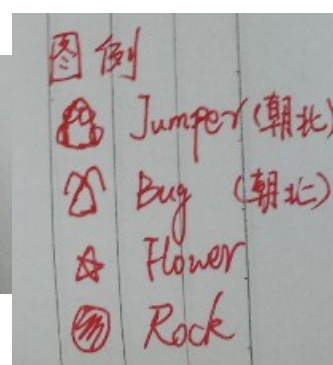
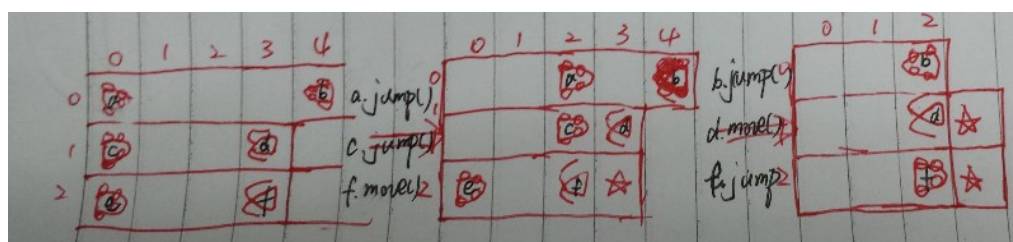
当 Jumper 跳到 flower 上再离开，原来位置的 flower 就会消失掉。

3 用例 2

测试目的：测试 Jumper 与其他 Actor 如 Bug 相对并且下一步都要移动到同一位置的行
为情况，实际情况有随机先后顺序，测试的时候也考虑不同的先后顺序

3.1 测试用例

测试用例：测试 Jumper 与其他 Actor 如 Bug 相对并且下一步都要移动到同一位置的
行为情况



如图所示，有 a,b,c,e 四个 Jumper，d,f 两个 Bug

a 的初始位置是 (0, 0)，方向为东

b 的初始位置是 (0, 4)，方向为西

c 的初始位置是 (1, 0)，方向为东

d 的初始位置是 (1, 3)，方向为西

e 的初始位置是 (2, 0)，方向为东

f 的初始位置是 (2, 3)，方向为西

进行以下操作后：

```
a.jump();
b.jump();
c.jump();
d.move();
f.move();
e.jump();
```

因为单独执行 jump 或者 move 后来的 actor 移动到对应位置会强制性覆盖前面的 actor

因此预期结果是：

a,b 抢，b 赢了。c,d 抢，d 赢了。e,f 抢，f 赢了。

```
a.setDirection(Location.EAST);
b.setDirection(Location.WEST);
c.setDirection(Location.EAST);
d.setDirection(Location.WEST);
e.setDirection(Location.EAST);
f.setDirection(Location.WEST);
world.add(new Location(0,0), a);
world.add(new Location(0,4), b);
world.add(new Location(1,0), c);
world.add(new Location(1,3), d);
world.add(new Location(2,0), e);
world.add(new Location(2,3), f);
```

```
Actor second = gr.get(new Location(1,2));
Actor third = gr.get(new Location(2,2));
assertEquals(270, b.getDirection());
assertEquals(true, (second instanceof Bug));
assertEquals(true, (third instanceof Jumper));
```

3.2 测试结果

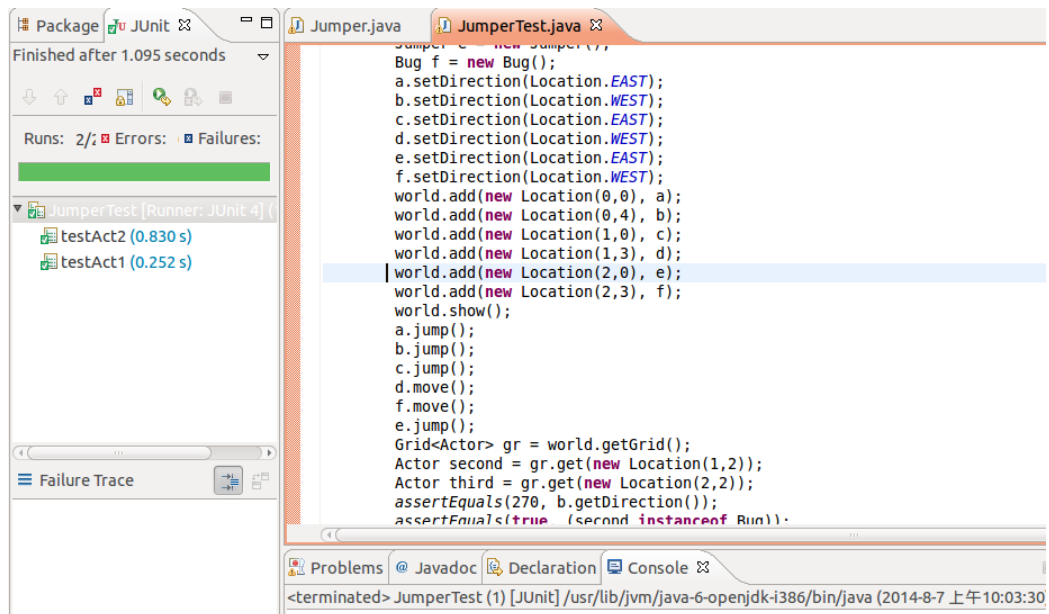
/**

* 在此描述测试结果，包括系统运行的截图

*/

测试结果与预期相符。

系统运行截图如下：



3.3 结果分析

由测试结果可知，结果与预期结果一致。