

# 测试报告

## version 1.0.0

修订历史

日期	版本	作者	描述
2015.8.21	1.0.0	GA207	完成 Jumper 类大部分方法的测试

--	--	--	--

# 目录

1	测试计划.....	3
2	用例 1.....	3
	2.1 测试用例.....	3
	2.2 测试结果.....	3
	2.3 结果分析.....	3
3	用例 2.....	4
	3.1 测试用例.....	4
	3.2 测试结果.....	4
	3.3 结果分析.....	4
4	用例 3.....	3
	2.1 测试用例.....	3
	2.2 测试结果.....	3
	2.3 结果分析.....	3
5	用例 4.....	4
	3.1 测试用例.....	4
	3.2 测试结果.....	4
	3.3 结果分析.....	4
6	用例 5.....	3
	2.1 测试用例.....	3
	2.2 测试结果.....	3
	2.3 结果分析.....	3
7	用例 6.....	4
	3.1 测试用例.....	4
	3.2 测试结果.....	4
	3.3 结果分析.....	4

## 1 测试计划

```
public ActorWorld w = new ActorWorld();  
//public UnboundedGrid<Actor> w.getGrid() = new UnboundedGrid<Actor>();  
public Jumper a = new Jumper();  
public Jumper b = new Jumper();  
public Rock r = new Rock();  
public Flower f = new Flower();  
public Actor x = new Actor();
```

```
//Init the location and direction.  
@Before  
public void setUp() throws Exception {  
    w.add(new Location(1, 1), a);  
    a.setDirection(Location.EAST);  
}
```

预置 Jumper a 到(1, 1)位置，方向为东。

## 2 用例 1

测试目的：测试 Jumper 基本的跳方法，是否跳到制定为位置，方向是否正确，对间隔跳中间的石头和鲜花碰撞是否存在，无法跳时是否转向。

## 2.1 测试用例

测试代码如下：

```
public void testJump() {
    a.act();
    assertEquals(a.getLocation(), new Location(1, 3));
    assertEquals(a.getDirection(), Location.EAST);

    w.add(new Location(1, 4), f);
    a.act();
    assertEquals(a.getLocation(), new Location(1, 5));
    assertEquals(f, w.getGrid().get(new Location(1, 4)));
    assertEquals(a.getDirection(), Location.EAST);

    w.add(new Location(1, 6), r);
    a.act();
    assertEquals(a.getLocation(), new Location(1, 7));
    assertEquals(r, w.getGrid().get(new Location(1, 6)));
    assertEquals(a.getDirection(), Location.EAST);

    a.act();
    a.act();
    assertEquals(a.getLocation(), new Location(1, 9));
    assertEquals(a.getDirection(), Location.SOUTH);

    f.removeSelfFromGrid();
    r.removeSelfFromGrid();
}
```

预期结果：

Jumper 按预期 2 格一跳，鲜花石头都不会被碰撞，无法跳跃时转向

## 2.2 测试结果

（测试结果截图在最后给出）

## 2.3 结果分析

结果与预期一致

## 3 用例 2

测试目的：是否会碰撞石头和鲜花

### 3.1 测试用例

测试代码如下：

3.2

测

```
public void testJumpToFlowerOrRock() {
    w.add(new Location(1, 3), f);
    a.act();
    assertEquals(a.getLocation(), new Location(1, 3));
    assertEquals(null, f.getGrid());
    assertEquals(a.getDirection(), Location.EAST);

    w.add(new Location(1, 5), r);
    int oldHp = a.getHp();
    a.act();
    assertEquals(a.getLocation(), new Location(1, 5));
    assertEquals(a.getDirection(), Location.EAST);
    assertEquals(oldHp - 1, a.getHp());
    assertEquals(new Location(1, 6), r.getLocation());

    r.moveTo(new Location(1, 9));
    a.act();
    a.act();
    assertEquals(null, r.getGrid());
}
```

### 测试结果

(测试结果截图在最后给出)

### 3.3 结果分析

石头会被碰撞并往碰撞方向“推动”一步，且 Jumper 掉 1 点血。  
鲜花不会被碰撞，会消失。

## 4 用例 3

测试目的：测试 Jumper 是否在即将跳出边界时转向。

### 4.1 测试用例

测试代码如下：

```
4.2      public void testJumpToOutside() {  
          a.setDirection(Location.WEST);  
          a.act();  
          assertEquals(new Location(1, 1), a.getLocation());  
          assertEquals(Location.NORTH, a.getDirection());  
      }
```

### 测试结果

（测试结果截图在最后给出）

### 4.3 结果分析

Jumper 会想顺时针转向 90 度。

## 5 用例 4

测试目的：测试在边界的跳跃。

### 5.1 测试用例

测试代码如下：

```
public void testFacingEdge() {  
    a.moveTo(new Location(0, 0));  
    a.setDirection(Location.WEST);  
    a.act();  
    assertEquals(new Location(0, 0), a.getLocation());  
    assertEquals(Location.NORTH, a.getDirection());  
}
```

### 5.2 测试结果

(测试结果截图在最后给出)

### 5.3 结果分析

在边界跳跃也会转向。



## 6 用例 5

测试目的：跳向一般 Actor 对象的结果

### 6.1 测试用例

测试代码如下：

```
public void testJumpToActor() {
    w.add(new Location(1, 3), x);
    int oldHp = a.getHp();
    a.act();
    assertEquals(a.getLocation(), new Location(1, 3));
    assertEquals(a.getDirection(), Location.EAST);
    assertEquals(oldHp - 1, a.getHp());
    assertEquals(new Location(1, 4), x.getLocation());

    a.moveTo(new Location(1, 7));
    x.moveTo(new Location(1, 9));
    oldHp = a.getHp();
    a.act();
    assertEquals(a.getLocation(), new Location(1, 9));
    assertEquals(a.getDirection(), Location.EAST);
    assertEquals(oldHp - 1, a.getHp());
    assertEquals(null, x.getGrid());

    a.setDirection(Location.SOUTH);
    w.add(new Location(3, 9), b);
    oldHp = b.getHp();
    a.act();
    assertEquals(a, w.getGrid().get(new Location(3, 9)));
    assertEquals(b, w.getGrid().get(new Location(4, 9)));
    assertEquals(oldHp - 1, b.getHp());

    b.removeSelfFromGrid();
}
```

## 6.2 测试结果

(测试结果截图在最后给出)

## 6.3 结果分析

预期为除了 flower 以外的对象都会被碰撞并且被推开一步。

# 7 用例 6

测试目的：两个 Jumper 跳跃路径交叉的影响

## 7.1 测试用例

测试代码如下：

```
public void testEncounterAnother() {
    w.add(new Location(1, 2), b);
    int oldHpOfA = a.getHp();
    int oldHpOfB = b.getHp();
    a.act();
    assertEquals(new Location(1, 3), a.getLocation());
    assertEquals(new Location(1, 2), b.getLocation());
    assertEquals(Location.EAST, a.getDirection());
    assertEquals(Location.NORTH, b.getDirection());
    assertEquals(oldHpOfA, a.getHp());
    assertEquals(oldHpOfB, b.getHp());

    b.removeSelfFromGrid();
}
```

## 7.2 测试结果

(测试结果截图在最后给出)

## 7.3 结果分析

预期是互不影响

### 最终结果截图

```
test:
[junit] Running JumperTest
[junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.05
2 sec
```

由图可见，六个测试全部成功。