

程序说明（171 组）

version 0.0

修订历史

日期	版本	作者	描述
2015/8/21	0.0	张振文	本程序为 Jumper，Jumper 是一只可以跳跃的虫子，它每次可以向前移动两格，可在网格世界内跳跃任何障碍物
		孙圣	
		彭勃	

目录

1	程序功能.....	3
2	实现过程.....	3
3	总结体会.....	4

1 程序功能

本程序为 Jumper, Jumper 是一只可以跳跃的虫子, 它每次可以向前移动两格, 可在网格世界内跳跃任何障碍物

1. 如果 Jumper 前面第二格有石头或者花, 直接把石头或花移除, 跳到该位置
2. 如果 Jumper 前面第二格在网格边界外, 则右转 90°
3. 如果 Jumper 前面面对边界, 则右转 90°
4. 如果遇到了其它的 actor, 则右转 90°
5. 如果遇到了其它的 jumper, 则右转 90°
6. 它能跳过任何东西, 除了石头和花以外, 还能跳过其它 Actor 和 Jumper

2 实现过程

1. 首先我们的 Jumper 需要继承 Bug 类, 所以先 `import info.gridworld.actor.Bug`。接着就让 `jumper extends Bug` 继承了 Bug 类

```
import info.gridworld.actor.Bug;
```

```
public class Jumper extends Bug {
```

2. 由于我们的设定是遇到不能跳跃的情况右转 90°, 所以重写了 `turn` 方法, 改成了

```
public void turn()
{
    setDirection(getDirection() + Location.RIGHT);
}
```

3. 由于 Jumper 一次要跳两格, 所以 `move` 方法里面的 `next` 要变成 `next` 的 `next`, 我们命名为 `nextTwo`, 这里调用了父类的 `moveTo` 方法, `moveTo(nextTwo)` 来实现跳跃

```
Location loc = getLocation();
Location next = loc.getAdjacentLocation(getDirection());
Location nextTwo = next.getAdjacentLocation(getDirection());
if (gr.isValid(nextTwo))
{
    moveTo(nextTwo);
}
else
{
    removeSelfFromGrid();
}
```

4. 对于 canMove 方法的重写

①首先判断 jumper 是否在网格的边缘，面向无效格子，如果是，则返回判断结果为 false;

```
Location nextOne = loc.getAdjacentLocation(getDirection());
if (!gr.isValid(nextOne))
{
    return false;
}
```

②再判断 jumper 前面第二格是否在网格外，为无效格子，如果是，则返回判断结果为 false;

```
Location nextTwo = nextOne.getAdjacentLocation(getDirection());
if (!gr.isValid(nextTwo))
{
    return false;
}
```

③最后判断前面第二格是否为空或者石头或者花，如果是，则返回判断结果为 true; 如果不是，则是其它 Actor 或 Jumper，则返回结果为 false

```
Actor neighbor = gr.get(nextTwo);
return (neighbor == null) || (neighbor instanceof Flower) || (neighbor instanceof Rock);
```

3 总结体会

本次 Jumper 程序的编写，让我们小组有了第一次的相对较集体交流沟通和讨论，增强了我们的表达能力以及相互之间协同合作的能力。我们在一开始就讨论了该如何定义 Jumper 类的功能和函数。然后就把任务以“函数”为单位分给每位组员，高效地“多线程”同时进行编程，最后再整合代码。让我们增长了些许团队合作的小经验。

同时，这个程序还让我们进一步地学习了 Java，也增强了对程序项目的分析能力。比如这个 Jumper，如果我们要把所有的函数都重头写的话，在这么短时间内肯定是不可能完成的。所以这里学会了 Java 中的继承，通过分析，我们认为继承 Bug 类会给我们带来比较直接的收益，最后就只需要重写 turn, move, canMove 这三个函数就可以了。

另外，我们的学习能力也得到了提升，通过学习 Bug 类中的函数，写出了我们自己定义功能的 Jumper。