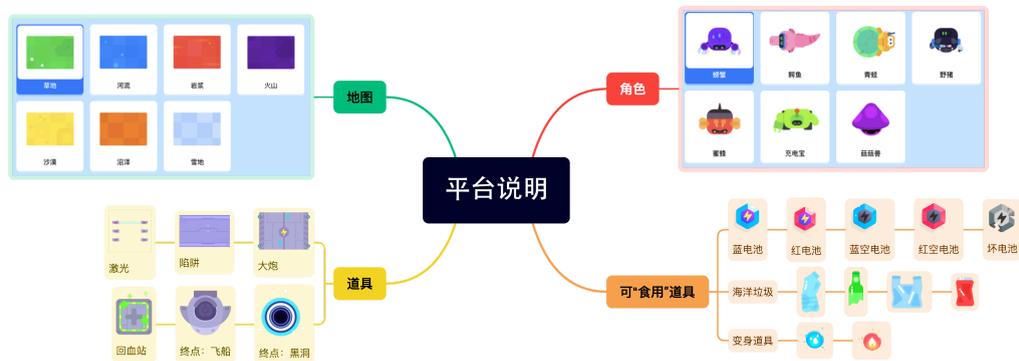
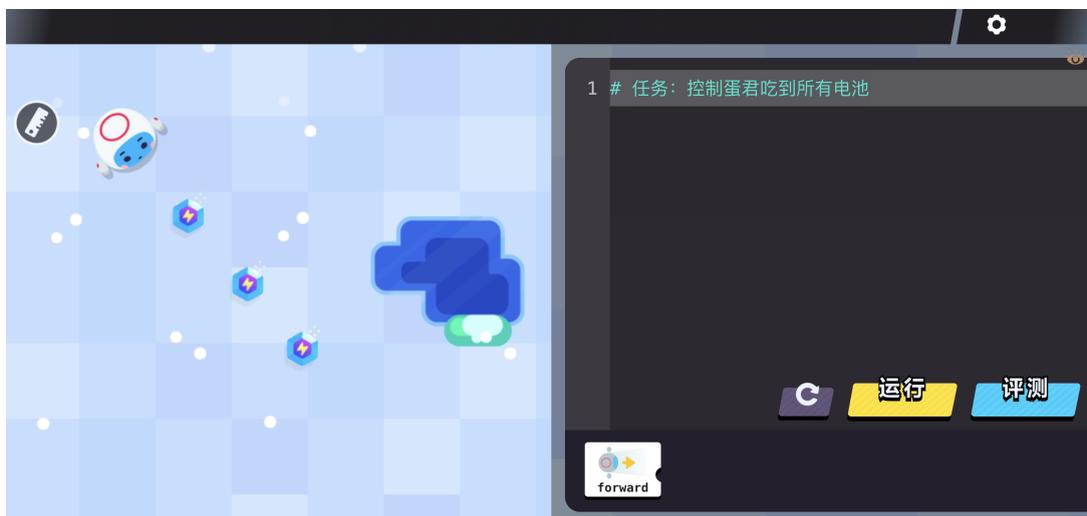


【信息素养大赛-编程闯关竞技赛】平台说...



一、基础移动指令



1.代码指令

知识	描述	示例
forward()	让角色向前走一段距离	forward(10)
left()	让角色左转一定角度	left(90)
right()	让角色右转一定角度	right(90)
对象.指令	让蛋君以外的角色也行动起来	crab.left(90) crab.forward(8)

2.尺子

用于测量任意两个角色之间的距离（步数）。



3.快捷移动指令

- goto():

控制蛋君前往某个角色的位置，示例：goto(battery)

- turn_to():

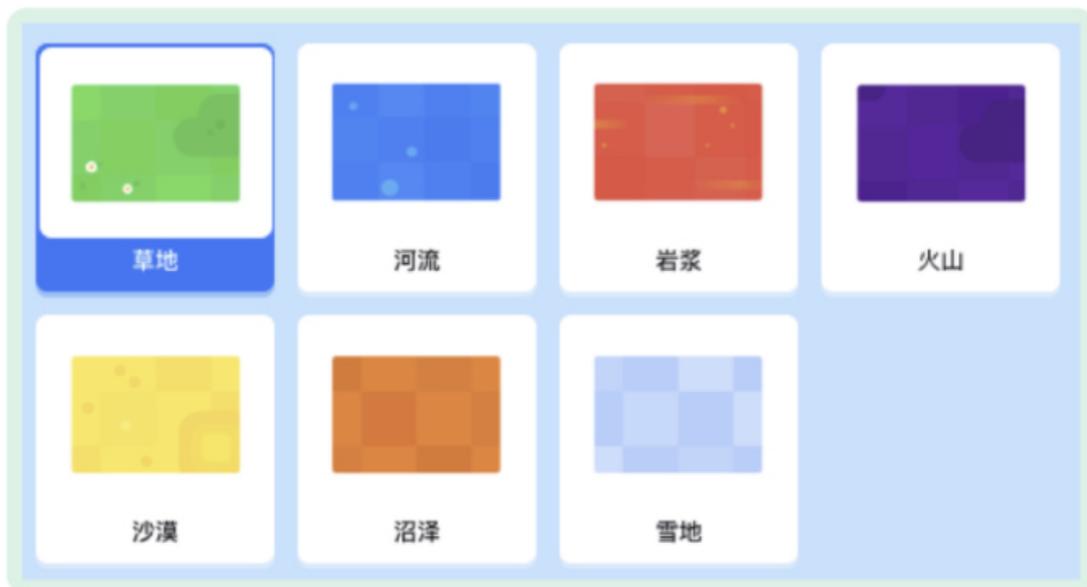
控制蛋君转向，正对某个角色，示例：turn_to(battery)

- distance_to():

计算蛋君到某个角色的距离，示例：distance_to(battery)

通常结合forward()语句指令。

二、地图

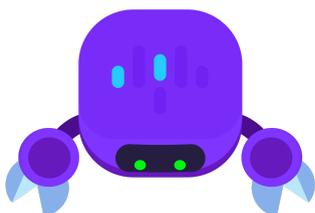


在草地、火山、沙漠、雪地上，蛋君可以踩在地图的任意位置。

在河流、岩浆、沼泽上，蛋君无法直接通过，需要借助小岛、木筏、桥梁、螃蟹等元素才能安全通过。

三、角色

1.螃蟹 crab



水上选手，可以在水中前进、后退、旋转，蛋君可以踩在他坚硬的壳上。

对象.指令

- 鼠标点击螃蟹，会出现螃蟹对象的名字crab
- 蛋君有主角光环，所以行动时直接用指令就行。而如果想要除了蛋君以外的角色动起来，需要使用【对象.指令】。
- 例如：

```
1 crab.left(90) # 让螃蟹向左转90度
2 crab.forward(8) # 让螃蟹前进8步
```

2. 鳄鱼crocodile



鳄鱼 (crocodile) 和青蛙一样，是我们可以控制的水中角色。鳄鱼的身体修长，体力充沛，蛋君可以踩在鳄鱼的身体上过河。

一般情况下，我们会请求鳄鱼旋转，用身体帮助蛋君搭建两座礁石之间的桥梁，方便蛋君跨越湖水。

鳄鱼由于不喜欢运动，所以他不会听从 forward() 指令，不能前进或后退。他只会在原地进行转圈。换句话说就是，鳄鱼只能执行 left() 和 right() 指令。

```
1 crocodile.left(90)
2 crocodile.right(90)
```

3. 青蛙frog



青蛙(frog)是我们可以控制的水中角色，青蛙的身体和荷叶都可以承受重量。

与螃蟹相比：①青蛙只能前进和后退，但不能转动。②可以发挥桥的作用帮助蛋君通行。

```
1 frog.forward(10)
2 frog.forward(-10)
```

4. 充电宝cube



充电宝 (cube) 有神奇的能力，它的机械手臂可以给没电的电池充电。我们要如何指挥充电宝呢？它可以前进、旋转。因此我们可以使用forward、left、right指令控制它。例如：

```
1 cube.forward(10)
2 cube.left(90)
```

如果在关卡中遇到有多个不同种类电池时，需要智能识别电池。使用指令red()可以判断是否为红电池。

例如：



```
1 battery.red() # 成立
```

5.野猪enemy



野猪(enemy)就像他的英文名字一样，他是我们的敌人，但是他的视力不是特别好，正常情况下野猪会站在固定的位置，一旦蛋君靠近了他，他就开始攻击蛋君。如果跑的够快，野猪会回到原来的位置站立。

如果蛋君吃了火之石或水之石变身后，可通过attack()指令消灭野猪。

```
1 attack(enemy)
```

6.蜜蜂bee



蜜蜂(bee)也是属于敌人的一类，他和野猪的区别在于，他会遵循固定的圆形轨迹进行巡逻。一旦靠近蜜蜂，他就会主动攻击你。如果蛋君吃了水之石这类道具后，可通过attack()指令消灭野猪。

天敌：水蛋君

```
1 attack(bee)
```

7. 菇菇兽mushroom



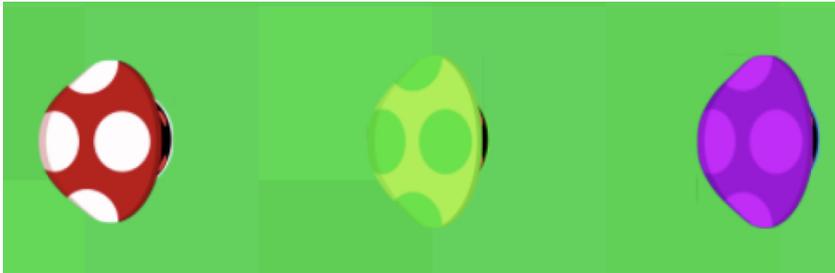
菇菇兽(mushroom)，看表情就知道它也是蛋君的敌人。

定位：坏角色，会攻击蛋君/充电宝；

行为：有多重形态；只会在配置好的长度范围内来回直行，不会主动攻击，蛋君只会在碰到它时掉血。

天敌：火蛋君

多种形态：



四、可食用道具

1. 电池 battery



- 蓝电池：通关条件，蛋君能吃；
- 红电池：通关条件，充电宝cube能吃；
- 坏电池：不能被充电，不能接触蛋君，否则蛋君会死掉；
- 蓝空电池：需要先被充电宝进行充电操作后，变成蓝电池。

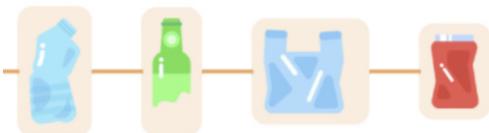
```
1 cube.goto(battery)
2 cube.charge()
```

- 红空电池：同上，充电后变成红电池。

识别电池的相关指令：得到的结果为成立(True) 或 不成立(False)

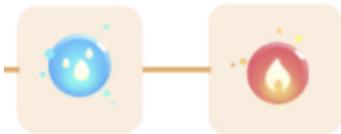
- 识别电池是否没电 battery.low_power()
- 识别电池是否为红电池 battery.red()
- 识别电池是否损坏 battery.broken()

2. 海洋垃圾 garbages



海上的收集物，类似于电池，收集关卡所有海洋垃圾才算通关。

3. 变身道具 stone



水之石 stone

吃到水之石后，蛋君会变身为水蛋君。

水蛋君能安全在水上行走，也能对敌人或某些障碍物发动攻击。



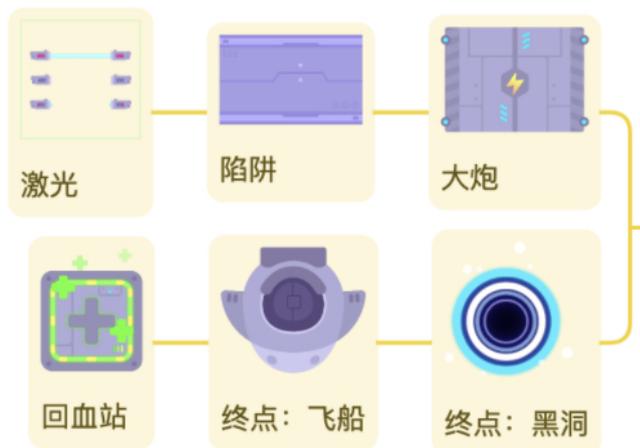
火之石 stone

吃到火之石后，蛋君会变身为火蛋君。

火蛋君能安全在岩浆上行走，也能对敌人或某些障碍物发动攻击。

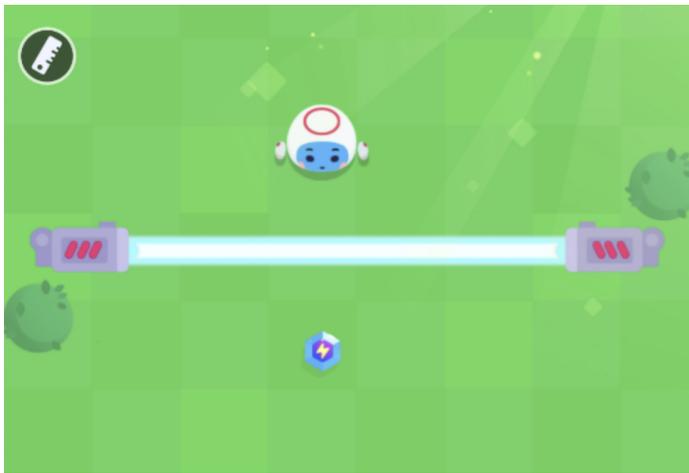


五、其他道具



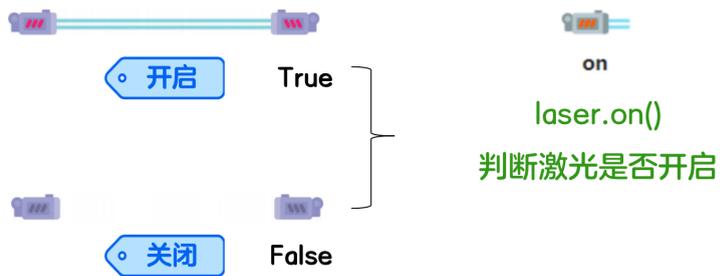
1. 激光 laser

激光时而开启，时而关闭。只有关闭状态时，蛋君才能安全通过哦！



激光laser的两种状态

电脑判断指令



通过机关的方式：



2. 陷阱 trap



在ailld中，同激光的使用方式，只不过对象名字不同。

```

1 # 示例程序
2 while trap.on():
3     wait()
  
```

3. 大炮 salvo



大炮的下面藏着很多的电池，当我们运行程序后，这些电池会被大炮一并发射出来，电池们会掉落在地图中的随机的位置，一般结合for遍历进行电池收集

```
1 for b in batteries:  
2     goto(b)
```

4. 回血站 health_zone



恢复体力区域，蛋君走到这里会补充体力，想要多待会记得wait()。

5. 终点：飞船 endpoint



在蛋君收集完所有的电池之后，前往飞船即可通关关卡。

5. 终点：黑洞 endpoint



同终点：飞船。

六、多个角色&道具

当角色&道具在关卡中存在有多个时，会以列表形式存储，例如本关卡的batteries电池列表。

