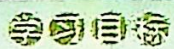




4.5 小松鼠找食物



- (1) 用键盘控制角色行走。
- (2) “观看此角色”指令的使用。*对对对*
- (3) “如果 那么”指令的使用。
- (4) 碰撞与反弹效果的实现。

小松鼠找食物的制作效果如图 4.5.1 所示。

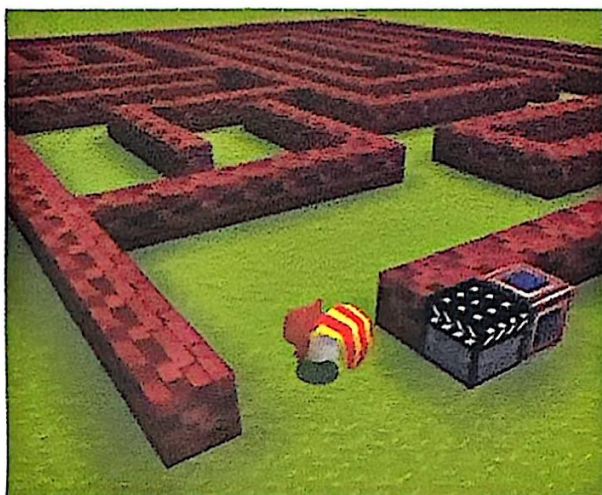


图 4.5.1 小松鼠找食物

4.5.1 打开世界

打开 Paracraft, 在图 4.5.2 所示位置输入“4914”, 进入迷宫的场景。

场景内已经有搭建好的迷宫, 迷宫的终点位置还放置了蘑菇; 另外还有一个“电影方块”和“代码方块”, “电影方块”里也已经放置了今天要用到的角色“松鼠”。



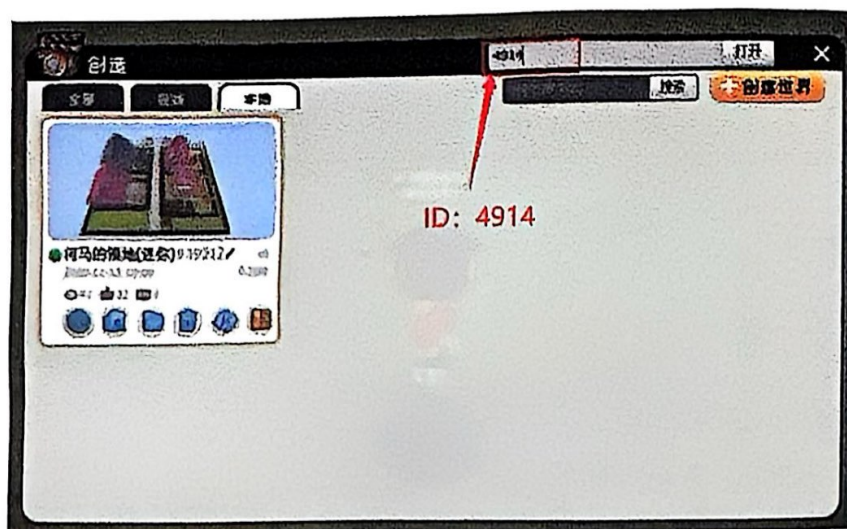


图 4.5.2 打开 4914 场景



进入世界之后,打开的是只读文件,需要将文件“保存”到本地,快捷方式为 Ctrl+S 键。

4.5.2 控制角色移动

(1) 单击“代码方块”,切换到“图块”编辑界面。

(2) 在“事件”分类下找到“当空格键按下时”,并将它拖曳到编辑区,将触发条件修改为当 W 键按下时。

(3) 当按下键盘的 W 键时,小松鼠应该往前移动 1 步。在“运动”分类下找到“前进 1 格在 0.5 秒内”,并将其拖曳到编辑区,单击运行按钮▶测试一下,如图 4.5.3 所示。

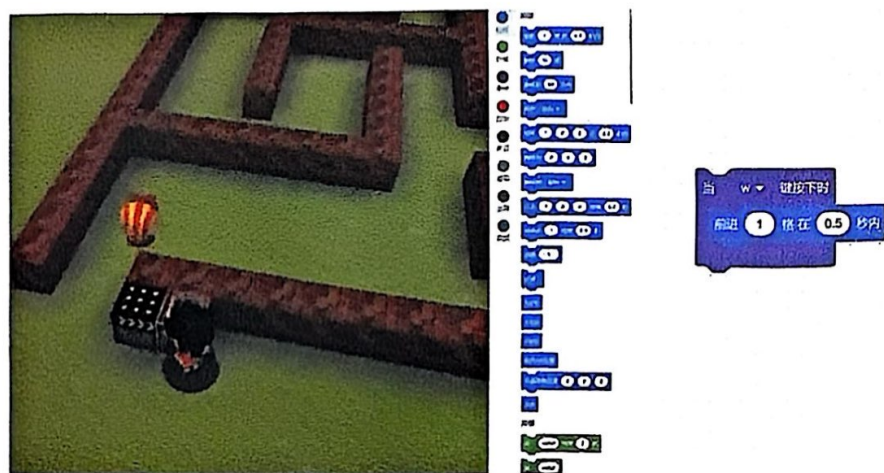


图 4.5.3 设置小松鼠向前走 1 格



(4) 当希望角色朝某个方向旋转并移动时,需要用到“旋转到_方向”这个指令。那么,“旋转到_方向”指令的方向应该怎么判断呢?

在 Paracraft 中,方向可以用世界坐标的 X 轴来判断,如图 4.5.4 和图 4.5.5 所示。



图 4.5.4 三维坐标轴

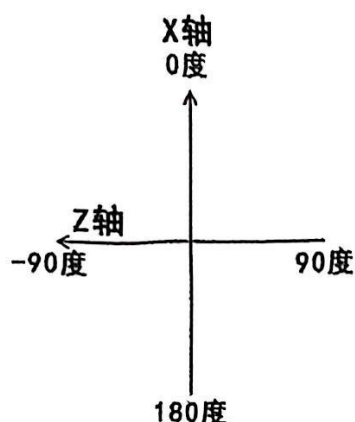


图 4.5.5 轴向和角度

跟慧编程的
过程一致

红色为 X 轴,蓝色为 Y 轴,绿色为 Z 轴;X 轴的箭头方向是 0 度;它左边的是一 90 度,右边是 90 度,后面就是 180 度。

因此,键盘控制角色行走的程序编写思路如图 4.5.6 所示。



图 4.5.6 控制角色移动的代码指令






按 W 键的时候,小松鼠应该“旋转到 0 度方向”,然后再“前进 1 格”。

按 S 键的时候,小松鼠应该“旋转到 180 度方向”,然后再“前进 1 格”。

按 A 键的时候,小松鼠应该“旋转到 -90 度方向”,然后再“前进 1 格”。

按 D 键的时候,小松鼠应该“旋转到 90 度方向”,然后再“前进 1 格”。

4.5.3 观看此角色 (演员与“我”分开, 我作为主视角, 观看演员运动)

单击运行按钮测试代码会发现,用 W、S、A、D 键来控制松鼠移动的时候,世界的主角也跟着在动,同时摄影机镜头也跟着一起动。这时候,我们需要用到“外观”分类下的“观看此角色”指令,将它拖曳到编辑区,并放置在所有代码的最上方,如图 4.5.7 所示。

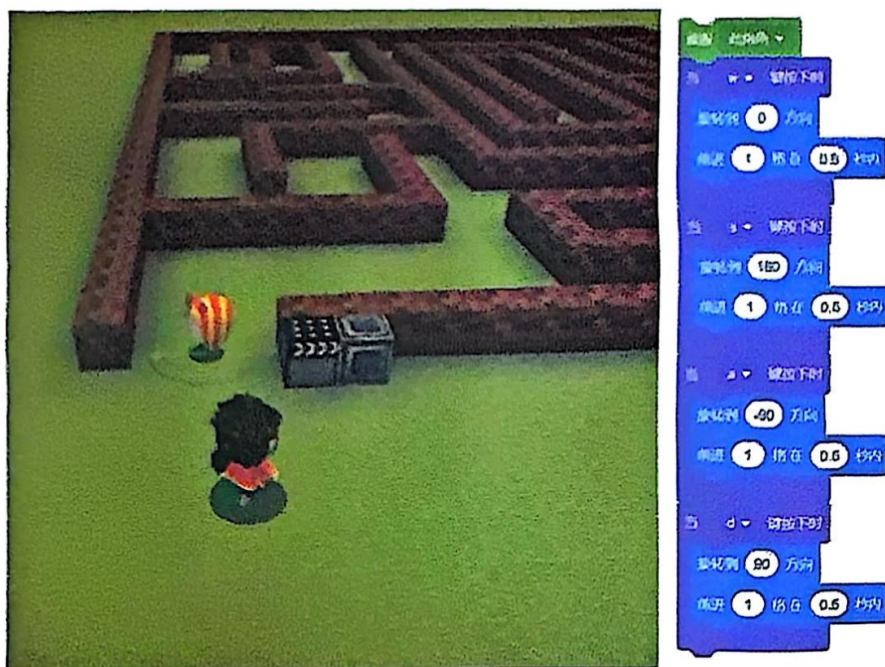



图 4.5.7 观看小松鼠的角色

再次单击运行按钮测试一下,摄影机就固定跟随着小松鼠了,同时, W、S、A、D 键也只对小松鼠起作用了。



扫一扫: 控制演员的创建方法:



4.5.4 条件限制

当角色触碰到墙的时候,我们不希望它穿过去,那要如何实现呢?

(1) 在“控制”分类下找到“如果 那么”指令。在“感知”分类下找到“是否碰到 block”,如图 4.5.8 所示。



任何材料
都可以有
圆形的
id

(2) 确定参数。按下键盘的 F3 键,然后将光标移动到墙面上,会发现数值框里显示“砖块 70”,70 这个数字就是砖块的 ID,将它填入“是否碰到”的参数框中。

(3) 条件响应。如果碰到墙壁了,小松鼠会怎么做呢? 我们可以使用“前进 -1 格在 0.5 秒内”实现小松鼠的后退。

(4) 重复检测。因为需要在程序运行过程中持续监测是否碰到墙壁,因此这一过程需要永远重复,如图 4.5.9 所示。

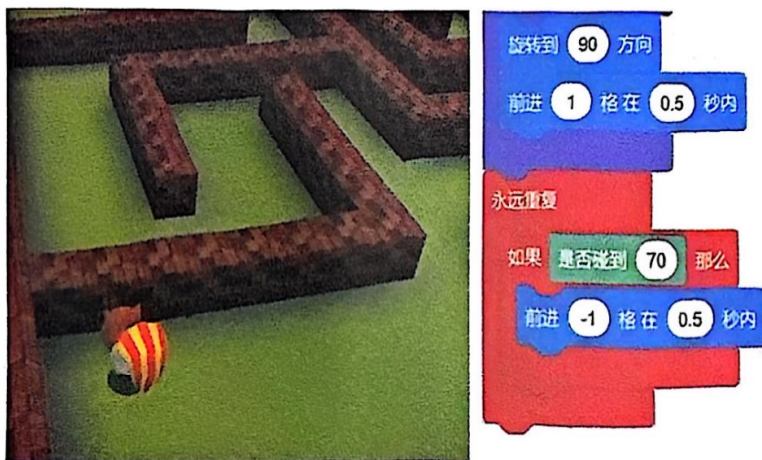


图 4.5.9 完整的碰撞反弹指令





扫一扫：条件限制的设置方法：



想一想：当小松鼠最终走到迷宫终点，遇到蘑菇，又会有怎样的反馈呢？你可以使用“外观”分类下的“提示文字 Start Game!”命令，自己去试试吧！

4.5.5 程序完善

松鼠吃到食物之后，游戏并没有结束，而是停留在了终点的位置。

我们可以使用“控制”分类下的“结束程序”或“重新开始”指令，使程序更加完整，快点试一试吧！

你挑战成功了吗？记得按 Ctrl+S 键保存作品哦！

学习心得：

