

《八年级第五单元 第23课-记忆力训练游戏》学生学习手册

学校名称: _____ 班级: _____ 姓名: _____

模块一：课程导入

真实情境

随着全球老龄化的加剧，阿尔兹海默症（俗称老年痴呆）已成为影响老年人生活质量的一大杀手。医学研究表明，早期的认知干预与记忆力训练能有效减缓大脑退化。

你是否想过，利用物联网技术开发一款“记忆力挑战设备”？它不仅能帮助长辈进行日常脑力锻炼，还能通过数据上传功能，让远方的子女实时掌握长辈的认知状态，实现“科技助老”。

核心问题

我们能否利用物联网技术，设计一款既能让老人进行记忆力挑战，又能将数据同步到云端的系统？

学习目标

通过本课程的学习，你将能够：

1. 理解物联网在医疗健康领域的应用价值。
2. 掌握使用列表（数组）来存储和处理序列数据的方法。
3. 学会使用按键进行人机交互，并实现多模式选择。
4. 理解“状态机”编程思想，并用它来管理复杂的游戏流程。

5. 设计并实现一个具备随机出题、实时反馈、难度递增和云端同步功能的记忆力训练游戏。

模块二：概念学习

1. 核心数据结构：列表（List / Array）

想一想：如果老师让你记住一串随机的数字“8, 2, 5, 9, 1”，你是如何在大脑里储存它们的？

你可能会在脑海里形成一个有序的“格子”，把这些数字按顺序放进去。在编程中，这种可以按顺序存放多个数据的“格子”，就叫做**列表**（或数组）。

- **特点：**可以存放多个数据，并且每个数据都有自己的“门牌号”（索引），从 0 开始。
- **用途：**在我们的游戏中，电脑随机生成的题目序列（如“上, 下, 左, 右”）和玩家输入的答案序列，都需要用列表来存储。

2. 核心逻辑：状态机（State Machine）

想一想：一个游戏通常有几个不同的阶段？比如“等待开始”、“游戏中”、“游戏结束”等等。

我们的记忆力游戏也一样，它有几个清晰的状态：

1. **难度选择状态：**等待玩家选择游戏难度。
2. **题目显示状态：**电脑正在依次展示题目序列。
3. **玩家答题状态：**等待玩家根据记忆输入答案。
4. **游戏结束状态：**显示最终得分，并上传数据。

程序在不同状态下，做的事情是完全不同的。这种根据当前状态来决定下一步做什么的编程思想，就叫做**状态机**。

模块三：逻辑分析

1. 系统功能拆解

系统部分	具体功能	实现方式（图形化积木块）
输入	1. 接收玩家选择的难度 接收玩家输入的答案	1. 使用“当按键...按下”积木块 2. 将按键对应的数值存入“玩家答案列表”
处理	1. 据难度生成随机题目序列 2. 依次显示题目序列 3. 逐一比较“题目列表”和“玩家答案列表” 4. 判断游戏是进入下一关还是结束 5. 将得分上传到云平台	1. 使用“重复...次”和“随机数”积木块，将结果存入“题目列表” 2. 循环遍历“题目列表”，在点阵屏上显示 3. 循环比较两个列表在相同索引位置的值是否相等 4. 使用“如果...那么...否则...”积木块 5. 使用“发送数据到云平台”积木块
输出	1. 在点阵屏上显示题目序列 2. 显示答对（√）或答错（✗）的反馈 3. 显示最终得分	1. 使用“点阵屏显示图案”积木块 2. 使用“点阵屏显示图案”积木块 3. 使用“OLED 显示字符串”积木块

2. 逻辑可视化：列表比较

看一看：下面的表格模拟了程序是如何比较两个列表的。

索引（“门牌号”）	0	1	2	3

题目列表 “上” “下” “左” “右”

玩家答案列表 “上” “下” “右” “左”

比较结果 相等 相等 不相等 -

请回答：程序在比较到第几个（索引值为多少）时，发现玩家出错了？此时应该做什么？

模块四：算法设计

让我们用一个流程图来梳理“记忆力训练游戏”的完整逻辑。

读一读，理一理：请仔细观察下面的流程图，并尝试用自己的话描述从“开始”到“结束”的完整过程。

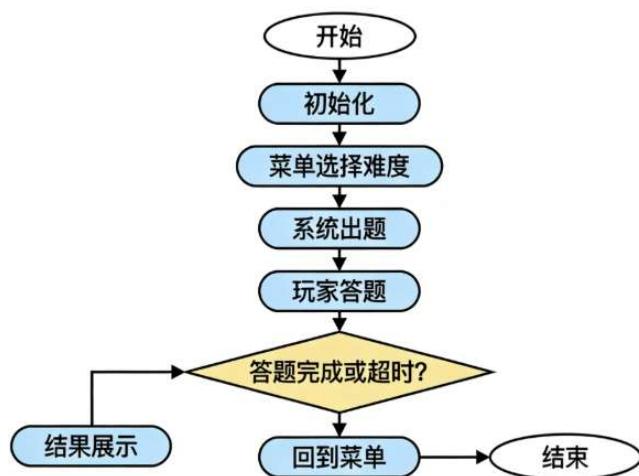


图1：游戏主流程图

模块五：搭建指导

这个项目的核心在于用一个“游戏状态”变量来控制程序的走向。

任务 1：搭建游戏主框架

目标：使用“如果...那么...”积木块，根据“游戏状态”变量的值，执行不同的操作。



动手试一试：

1. 创建一个名为“游戏状态”的变量。
2. 在“重复执行”积木块内部，放入多个“如果...那么...”积木块。 - 如果 游戏状态 == 1，执行难度选择的逻辑。 - 如果 游戏状态 == 2，执行题目显示的逻辑。 - 如果 游戏状态 == 3，执行玩家答题的逻辑。 - 如果 游戏状态 == 4，执行游戏结束和上传分数的逻辑。

任务 2：实现“随机序列生成”

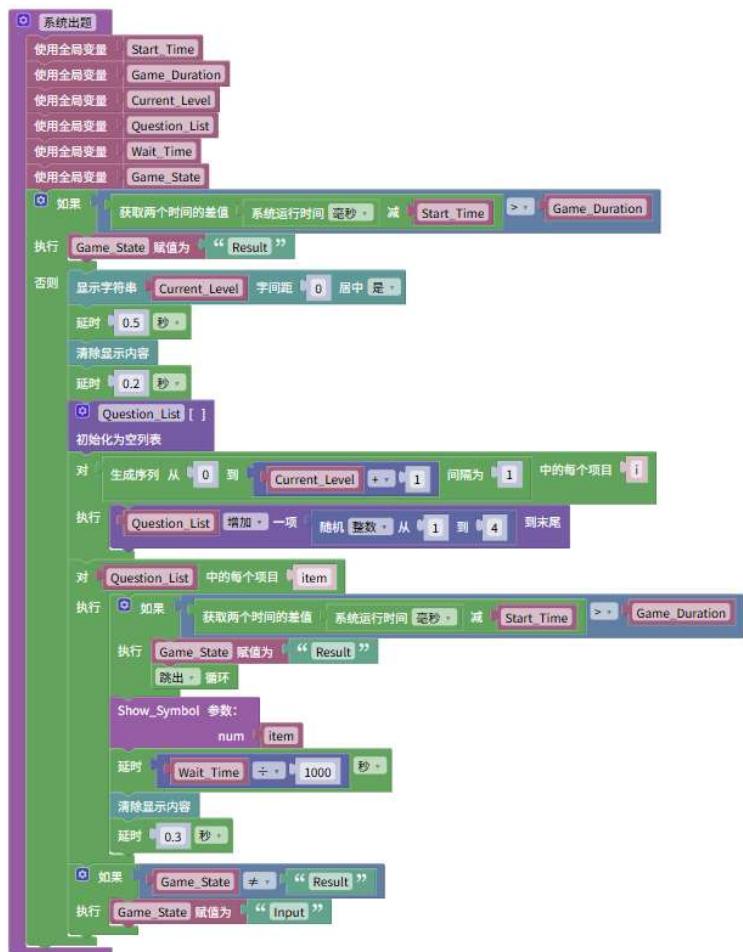
目标：创建一个长度随关卡增加的随机方向列表。

想一想：如何让列表的长度每一关都增加 1？

动手试一试：

1. 创建一个名为关卡等级的变量，初始值为 1。

2. 使用“重复关卡等级次”的循环积木块。
3. 在循环内部，生成一个1到4的随机数。
4. 用“如果...那么...”积木块，将1对应“上”，2对应“下”，3对应“左”，4对应“右”。
5. 将生成的方向添加到“题目列表”中。



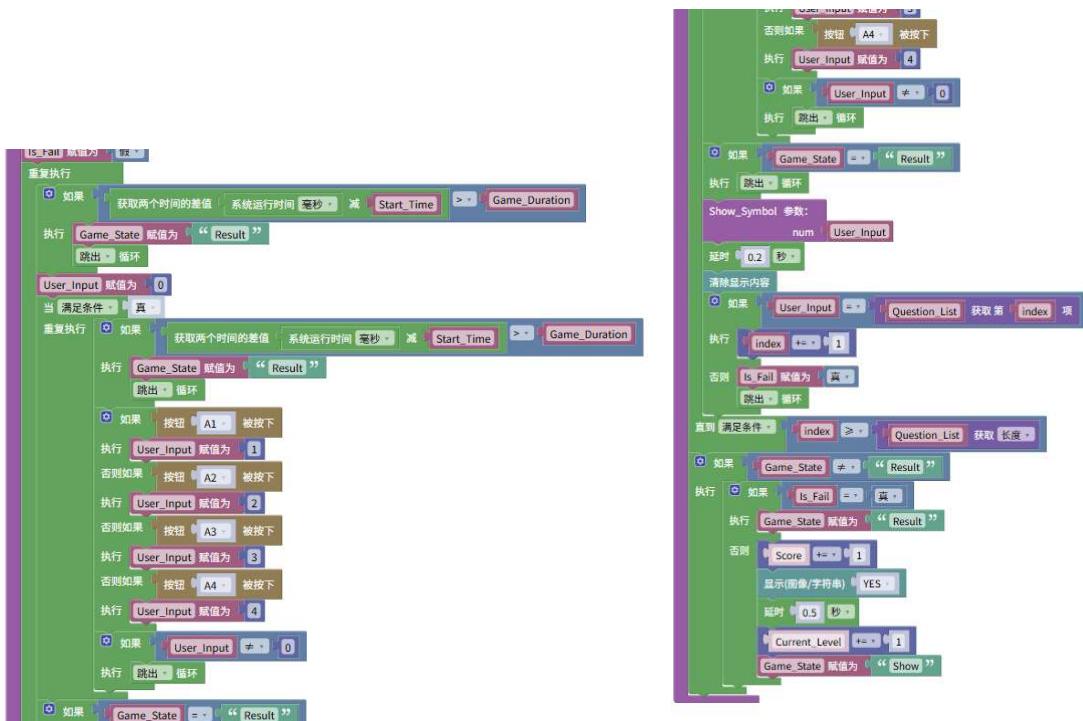
任务3：实现“玩家输入与比较”

目标：逐一比较玩家的输入和标准答案。

想一想：程序如何知道玩家当前正在回答的是第几个题目？

动手试一试：

1. 创建一个名为当前答案索引的变量，初始值为 0。
2. 当玩家按下按键时，比较他的输入是否等于“题目列表”中第当前答案索引个元素。
3. 如果正确，将当前答案索引加 1。然后检查当前答案索引是否等于关卡等级，如果相等，说明本关通过！
4. 如果错误，立刻将“游戏状态”变量设为 4（游戏结束）。



模块六：调试指导

1. 分段测试法

不要等所有积木块都搭完再测试！每完成一个核心功能，就测试一次。

第一步：只测试“难度选择”功能。按下按键，看看对应的难度变量是否变化。

第二步：测试“随机序列生成”功能。让程序生成一个序列，并用显示屏打印出来，看看是否符合预期。

第三步：测试“玩家答题”功能。手动设置一个固定的“题目列表”，然后输入答案，看看程序判断是否正确。

2. 常见问题排查清单

故障现象	可能原因	解决方法
游戏卡住 不动	“游戏状态”变量没有被正 确改变	使用显示屏或串口打印“游戏状态”变量的 值，看它停在了哪个状态。
答案总是 错误	列表比较的逻辑有误，或 索引变量没有正确增加	打印“题目列表”和“玩家答案列表”，以及 “当前答案索引”的值，进行对比。
难度没变 化	关卡等级变量没有在过关 后增加	检查过关后，是否执行了“将关卡等级加 1” 的积木块。

模块七：拓展挑战

- 增加“生命值”系统：**玩家不再是一次失误就结束游戏，而是有 3 次机会。你需要增加一个“生命值”变量，每次答错就减 1，减到 0 才游戏结束。
- 增加“双人对战”模式：**两名玩家轮流进行记忆挑战，看谁能达到更高的关卡。
- 数据可视化：**在物联网平台上，将每次上传的得分用折线图展示出来，观察记忆力训练的进步曲线。

模块八：学习反思

知识清单

请用你自己的话，描述一下你对以下概念的理解：

- **列表 (List) :**
- **状态机 (State Machine) :**
- **随机数 (Random Number) :**

“拦路虎”日志

我遇到的问题 我是如何解决的 我学到了什么

创意思考

除了记忆力训练，这项技术还能用在哪些帮助老年人的场景中？（例如：用药提醒、防走失预警、趣味社交等）