

AGES 10-16

8 × 90 MIN

A\$480

AIRBOTIX FAMILY GUIDES · 澳洲家长成长指南

Build a Super Mario Game with AI

8 节精品视觉课程图鉴 · 看见创意怎样变成代码、游戏与正式作品

8 节课程，8 幅精品教学视觉：让家长看见孩子怎样从角色创意走向游戏逻辑、测试与正式发布。

2026 年 7 月精品视觉版

澳洲 10-16 岁孩子家庭



01

先把课程事实说清楚

如果您正在判断这门课是否适合孩子，下面几项比宣传口号更重要。

家长需要确认	当前课程安排
年龄	10-16 岁
基础	零代码基础可以；需要会打字并能阅读基础英文
形式	线下面授，8-15 人小班
时长	8 节，每节约 90 分钟
价格	A\$480，即每节 A\$60
核心环境	Kids OpenCode；AI 美术、音乐和声音工具通过课程环境使用
最终产出	完整可玩游戏、可分享链接、标题卡、短预告 GIF、Demo Day 展示

线下场地、开班日期与剩余名额会随班次变化。您不需要现在就确定所有安排；可以先联系 Rain，我们会和您一起确认孩子的基础、方便地点与当前班次。

本手册按 2026 年 7 月 24 日的线上课程 API 与产品课程源核实。旧版 9 周资料已经不再代表当前课程。

02

八节课的知识不是零散的，而是一层层连接起来

家长担心“每周做一点，最后拼不起来”很正常。这门课从第一节起就在同一个项目上持续增加功能，每节都有可运行成果，也都有下一节需要使用的**基础**。

节次	主要学习知识	看得见的理解证据
1	Art direction、角色一致性、迭代与项目结构	能说明为什么选择这一版角色，项目能稳定打开到原创标题画面
2	动画帧、键盘事件、速度、重力、跳跃参数与调试	英雄能走跑跳，孩子能指出关键代码并解释两个物理参数
3	Tilemap、坐标网格、碰撞边界、关卡节奏与试玩	能把纸上路线变成可完成关卡，并根据玩家反馈修改
4	Event、variable、score、音频触发与混音	能解释金币怎样让变量改变、声音为什么在正确事件出现
5	Condition、巡逻算法、碰撞方向与公平规则	能解释敌人何时转向、为什么踩踏与侧碰结果不同
6	State machine、状态切换、能力变化与反馈设计	能指出状态在哪里改变、哪些行为会读取这个状态
7	Game loop、HUD、胜负条件、场景切换与重置	能完整测试胜利、失败和重试三条路径
8	Playtest、bug report、triage、deployment与作品表达	能用证据修复问题，并让另一台设备打开最终作品

每节课都遵循“看懂目标 → 动手制作 → 运行测试 → 解释结果 → 当堂验收”。进度快的孩子会得到延伸挑战；遇到困难的孩子会先守住本节最关键的**可运行成果**。

03

最终作品包含什么，也有哪些边界

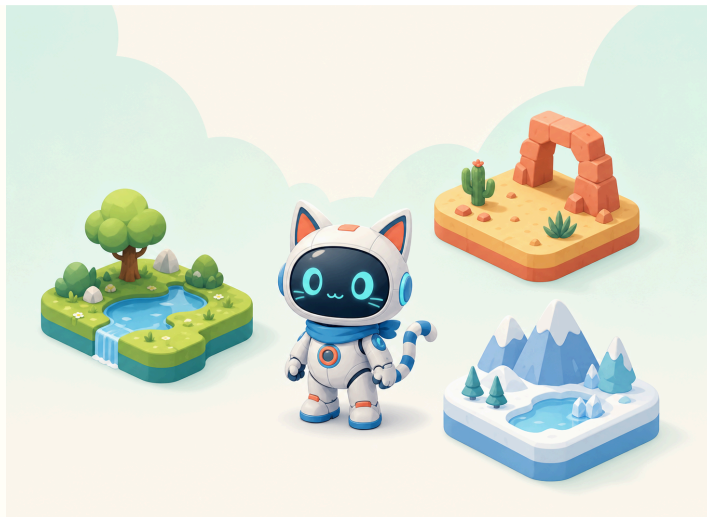
八节课结束时，孩子应完成一款有完整游戏循环的原创横版闯关游戏：

- 自己设计的主角、世界、敌人、道具、界面和关卡；
- 行走、跑动、跳跃、重力、碰撞与操作手感；
- 平台、空隙、路线、金币、得分、生命与终点；
- 原创主题音乐、跳跃/金币/Power-up/胜利等声音反馈；
- 敌人、受伤、失败、重试与完整胜负状态；
- 经过同学试玩、难度调整和最后修复的公开版本；
- 标题卡、短预告 GIF、作品链接与 Demo Day 讲解。

课程名保留 **Build a Super Mario Game with AI**，但孩子制作的是自己的原创平台闯关作品。课程不使用 Nintendo 官方角色、Logo、音乐或素材，也不表示与 Nintendo 存在授权或合作。公开分享还需要遵守平台规则与家长同意要求。

04

第1节：Design your hero & world



第1节视觉示例：从原创角色设定与世界方向开始。

第一节不是先让孩子复制代码，而是先建立“这是谁的作品”。孩子会观察平台游戏怎样用主角、世界和第一屏建立记忆点，再担任自己项目的美术指导。

本节一眼看懂

- **学习：**英雄、世界与标题画面怎样共同建立作品身份；
- **动手：**迭代正面、侧面与跳跃姿势，比较3个世界方向并搭建标题画面；
- **AI与孩子：**AI提供草案，孩子给方向、拒绝不一致结果并选择最终版本；
- **看得见的成果：**游戏重新启动后，仍能进入“原创英雄站在原创世界里”的标题画面。

这里练习的不是“生成一张漂亮图片”，而是持续表达、比较与修改。孩子保留第一张结果并不算完成；他需要说明自己为什么选择这一版，以及哪些细节让角色真正属于自己。

第1节精品视觉：从一个想法建立完整视觉方向

左侧看懂这张图

- **角色规则：** 颜色、比例和标志细节先保持一致；
- **世界选择：** 比较多个方向，再决定作品气质；
- **设计过程：** 草稿、反馈和修改都属于最终成果的一部分。

当堂证据： 孩子能说明为什么保留最终角色，并让项目稳定进入原创标题画面。

家长可以问：“换一个姿势后，哪些特征一定不能变？”



第1节精品视觉：角色一致性与世界选择

这节课怎样把知识变成能力

- **孩子会做：** 在多版草稿中比较颜色、轮廓、比例和配件，把选定规则继续用于侧面、跳跃姿势与标题画面，而不是每次重新接受一个随机结果；

- **老师会检查：** 请孩子说明保留与拒绝的理由，指出不同姿势中的不一致，再确认项目重新启动后仍能稳定显示同一个原创角色和世界；

- **最后能解释：** AI 可以快速提供选择，但作品方向来自孩子。角色一致性、世界气质和最终取舍，都需要孩子自己判断并承担决定。

第1节知识拆解：孩子怎样建立视觉方向

这张图在讲什么

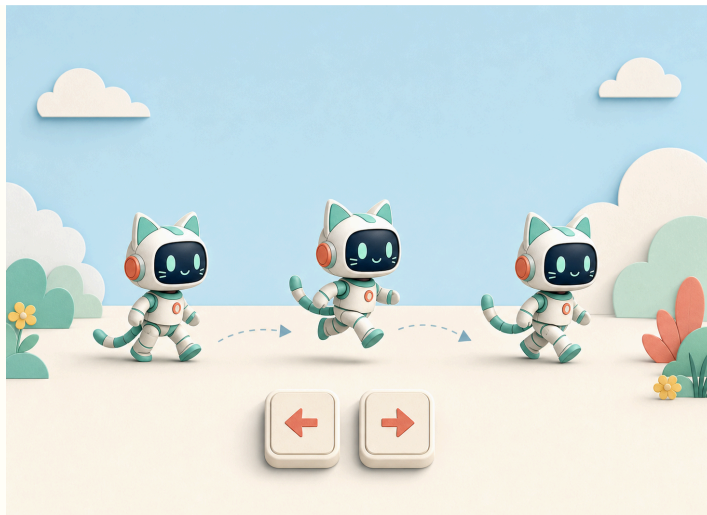
- 先定规则，再请 AI 帮忙。孩子先定义配色、性格和标志细节，AI 才不是随机出图；
- 一致性比单张漂亮更重要。正面、侧面和跳跃姿势都要像同一个角色；
- 专业创作需要迭代。第一版只是草稿，比较、指出问题、修改和选择才是设计过程。

孩子真正掌握的标准：他不只会说“AI 画了一个角色”，还能够解释自己给了什么方向、拒绝了什么、为什么最终版本更符合自己的世界。

家长课后可以问：“如果把角色换一个姿势，哪些特征一定不能变？”能说出颜色、脸型、比例或标志配件，就说明孩子理解了“角色一致性”。

07

第2节：Make it move & jump



第2节视觉示例：动画、键盘移动、重力与跳跃调参。

第二节是整门课最密集的一节：静态角色要在同一节里学会动画、走路、跑动和跳跃。老师会把“先读代码、再运行代码”建立为后续每节都要使用的习惯。

本节一眼看懂

- **学习：** 动画帧、键盘输入、速度、重力与跳跃力度；
- **动手：** 制作 idle/run 帧，逐行读移动代码，完成一次单变量调参实验；
- **AI 与孩子：** AI 起草和解释代码，孩子先预测、再运行、一次只改一个数字；
- **看得见的成果：** 英雄能稳定走、跑、跳和落地，孩子能解释两个物理参数。

“跳起来了”只是功能完成；能比较太飘、太重与刚刚好的差别，并说出是哪一个参数造成变化，才是这一节真正的学习成果。

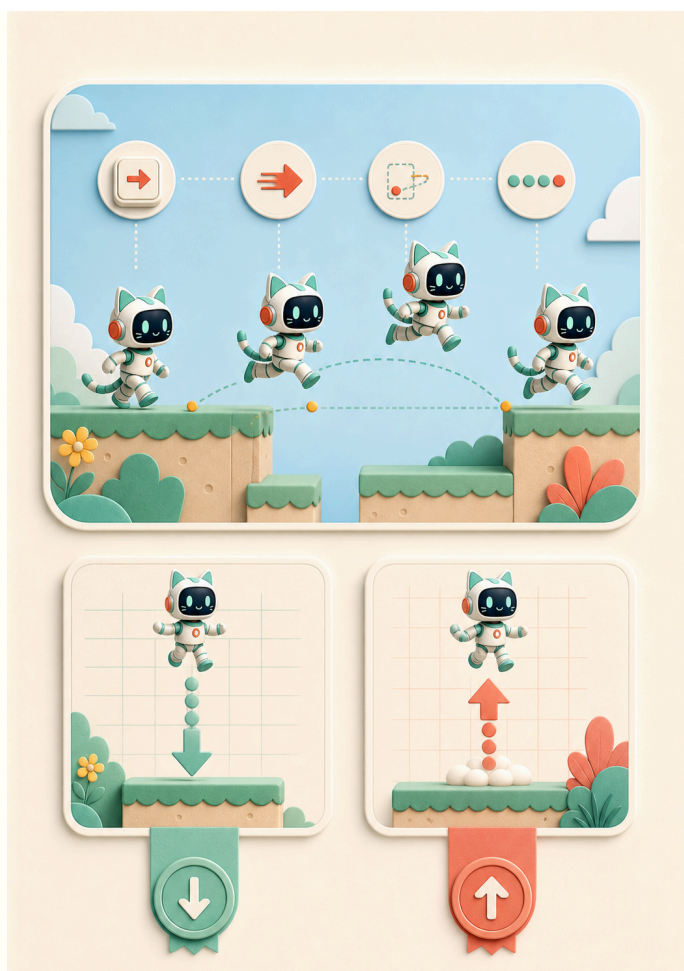
第2节精品视觉：动画、输入、速度与游戏物理

左侧看懂这张图

- **输入：** 按键先改变速度，速度再改变位置；
- **跳跃：** 向上的初速度与向下的重力共同作用；
- **调试：** 一次只改一个参数，才能解释结果。

当堂证据： 英雄能稳定走、跑、跳和落地，孩子能解释至少两个物理参数。

家长可以问：“如果只把重力调大，跳跃会发生什么变化？”



第2节精品视觉：角色运动与跳跃

这节课怎样把知识变成能力

- **孩子会做：** 先预测按键、水平速度、跳跃初速度和重力会怎样影响角色，再运行游戏；调试时一次只改一个数字，比较修改前后的手感；
- **老师会检查：** 角色是否能稳定走、跑、跳和落地，并请孩子指出监听键盘、改变速度和施加重力的关键位置，而不是只看画面动起来；
- **最后能解释：** 输入不会直接把角色搬到新位置。程序先改变速度，再持续更新位置；向上的初速度与向下的重力共同形成跳跃轨迹。

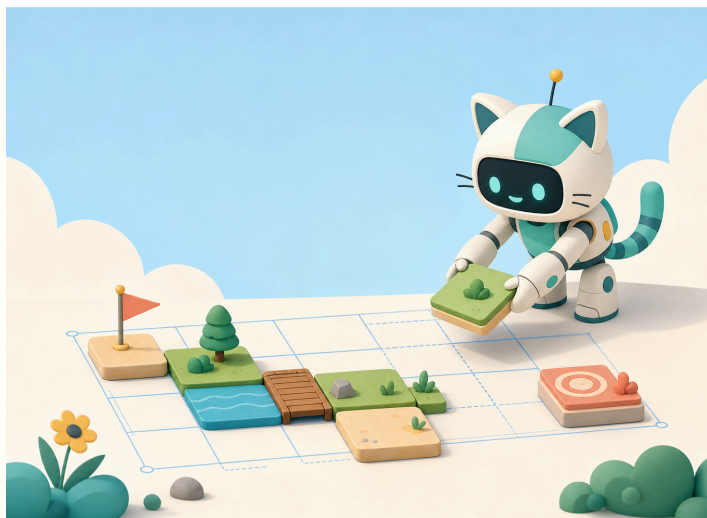
第2节知识拆解：角色为什么会动起来

这张图在讲什么

- 按键不会直接“搬动”角色。程序先改变速度，再持续更新位置；
- 跳跃来自两股力量。向上的初速度与不断向下的重力共同决定手感；
- 调试要控制变量。一次只改一个数值，才能知道结果为什么改变。

孩子真正掌握的标准：他能指出哪段代码监听键盘、哪段代码改变速度，并用自己的话解释至少两个物理参数如何影响跳跃。

家长课后可以问：“位置和速度有什么不同？如果只把重力调大，跳跃会怎样变化？”这能判断孩子是在理解运动模型，而不是只把AI给出的数字复制进去。



第3节视觉示例：先在纸上设计，再把地块组成可玩的路线。

第三节会先收起键盘，用格子纸设计。这样做是为了让孩子明白：关卡不是随意堆平台，而是要安排“先教会玩家、再测试玩家、最后奖励玩家”的体验节奏。

本节一眼看懂

- **学习：** 网格、碰撞、路线可读性与 teach → test → reward；
- **动手：** 在格子纸标出起点、空隙、高路线、奖励和终点，再按图搭建；
- **AI 与孩子：** AI 生成统一风格的地块，孩子决定路线、跳跃距离与难度；
- **看得见的成果：** 一段使用原创地块、能从起点完整跑到终点的关卡。

这一节留下两份证据：纸上设计与实际关卡。家长能看到孩子不是“边做边乱放”，而是在把一个设计意图转换为可运行结果。

第3节精品视觉：网格、碰撞与关卡设计

左侧看懂这张图

- **网格：** Tilemap 记录地面、平台和空隙的位置；
- **碰撞：** 看得见的平台还需要可以站立的边界；
- **节奏：** 好关卡会先教会、再测试、最后奖励玩家。

当堂证据：纸上路线能转换成一段从起点跑到终点的可玩关卡。

家长可以问：“第一次玩的玩家跳不过时，你会先改哪里？”



第3节精品视觉：从关卡草图到可玩世界

这节课怎样把知识变成能力

- **孩子会做：** 先在格子纸上标出起点、安全区、空隙、高路线、奖励与终点，再把这套路线转换成 Tilemap，而不是打开编辑器后随意堆放平台；

- **老师会检查：** 关卡能否从起点完整跑到终点，平台是否有正确碰撞边界，以及孩子能否根据第一次试玩者卡住的位置作出一次具体修改；

- **最后能解释：** 一张漂亮背景并不是关卡。可玩关卡需要网格数据、碰撞规则和体验节奏，同时照顾“看得懂、跳得到、愿意继续尝试”。

第3节知识拆解：好关卡怎样教会、测试并奖励玩家

这张图在讲什么

- 背景图不等于关卡。Tilemap 用网格数据说明哪里是地面、平台或空隙；
- 看得见还不够。平台还需要碰撞边界，角色才能真的站上去；
- 设计要为玩家服务。先安全教会，再增加挑战，最后用奖励确认成功。

孩子真正掌握的标准：他能把纸上路线转换为可运行关卡，并说明哪个位置负责“教”、哪个位置负责“考”、玩家成功后得到什么奖励。

家长课后可以问：“为什么这个空隙放在这里？如果第一次玩的玩家跳不过，你会改什么？”回答若能联系跳跃距离、提示和难度，就说明孩子开始具备玩家视角。

第4节：Coins, sound & music



第4节视觉示例：让收集、得分、音效和音乐在正确事件发生。

第四节把代码、音效和音乐拆成三个工作站。孩子会第一次使用“状态”：金币被收集后，数量要改变；声音也必须由正确事件触发，而不是一直播放。

本节一眼看懂

- **学习：** 事件、变量、得分、声音触发与混音；
- **动手：** 加入收集物和计数，制作金币/跳跃音效并创作短主题音乐；
- **AI与孩子：** AI 提供声音草案，孩子选择情绪、节奏、音量与触发时机；
- **看得见的成果：** 金币只加一次分，音效在正确时刻出现，音乐不会盖住反馈。

音乐生成会设定明确时间盒，避免孩子一直刷新版本却没有完成整合。专业制作不等于无限选择，而是能在限制内做决定并交付。

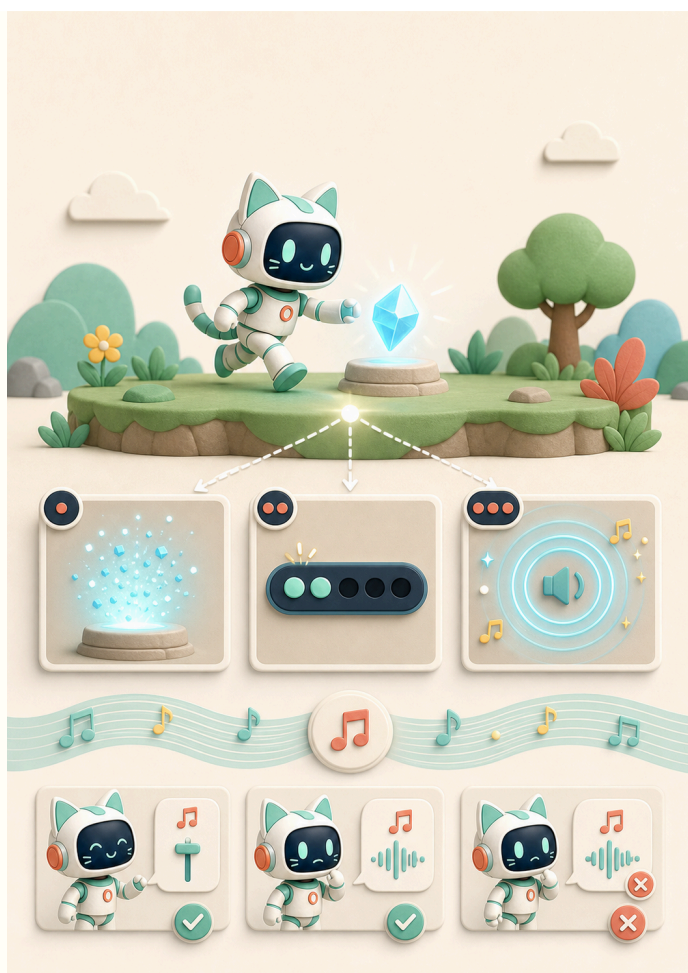
第4节精品视觉：事件、变量、计分与声音反馈

左侧看懂这张图

- **事件：**英雄碰到水晶，程序才开始执行动作；
- **变量：**score 让游戏记住不断变化的分数；
- **反馈：**消失、加分和音效需要在同一时刻发生。

当堂证据：水晶只加一次分，反馈声音在正确事件发生时播放。

家长可以问：“碰到水晶以后，程序依次做了哪几件事？”



第4节精品视觉：一枚收集物背后的事件链

这节课怎样把知识变成能力

- **孩子会做：**把“碰到水晶”拆成检测碰撞、隐藏收集物、增加分数和播放音效四个连续动作，并调整主题音乐与反馈声音的音量关系；

- **老师会检查：**同一枚水晶是否只加一次分，计分显示是否与变量同步，声音是否只在正确事件发生时出现，而且不会被背景音乐盖住；

- **最后能解释：**事件负责启动动作，变量负责保存变化，视觉和声音负责告诉玩家结果。三者连接起来，游戏才会对玩家的操作作出可信回应。

第4节知识拆解：一枚收集物背后的事件链

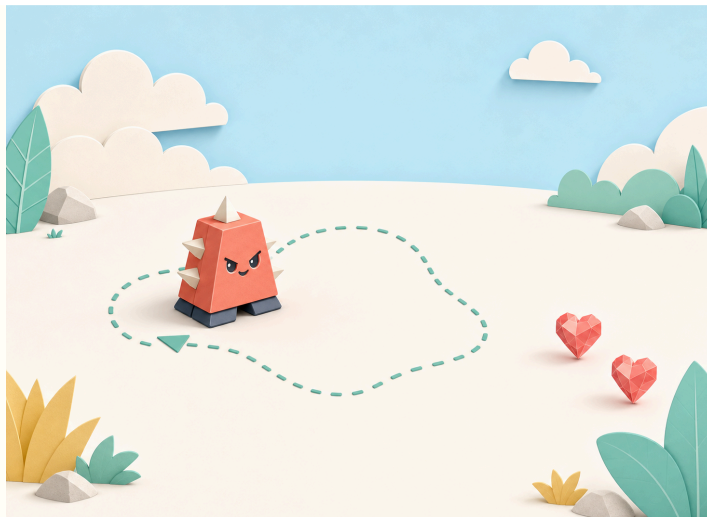
这张图在讲什么

- 事件是起点。英雄碰到金币，才触发后续一组动作；
- 变量让游戏有记忆。score 保存会变化的分数，而不是只显示一张数字图片；
- 反馈需要同步。金币消失、分数增加和“叮”的声音要让玩家感到是同一件事。

孩子真正掌握的标准：他能解释“金币被碰到”这个事件如何让分数变量改变，并能指出声音在哪一个事件发生时播放。

家长课后可以问：“金币被碰到以后，程序按什么顺序做了哪些事？”若孩子能说出检测碰撞、隐藏金币、增加分数、播放音效，说明他理解了事件链。

第5节：Enemies & danger



第5节视觉示例：敌人巡逻、转向、被踩踏与造成伤害。

第五节加入风险。孩子会先用身体演示巡逻规则，再逐行检查敌人代码，避免“AI 写好了，所以我不知道它为什么转向”。

本节一眼看懂

- **学习：** 条件判断、巡逻算法、碰撞方向与公平挑战；
- **动手：** 设计敌人，设定左右巡逻，实现踩踏、侧碰受伤和重新开始；
- **AI 与孩子：** AI 起草代码，孩子指出转向条件并测试上方、侧面与边缘碰撞；
- **看得见的成果：** 敌人会巡逻、能被踩败，也会在侧面碰撞时伤害角色。

“更难”不一定更好。孩子需要说明玩家是否看得见危险、是否有反应时间，以及失败后能否理解原因。

第5节精品视觉：条件判断、巡逻与碰撞规则

左侧看懂这张图

- **巡逻：**移动、检查边缘、转向，再继续检查；
- **条件：**墙壁、悬崖和碰撞方向决定下一步；
- **公平：**踩踏与侧碰需要给玩家不同而清楚的结果。

当堂证据：敌人会巡逻和转向，踩踏与侧碰能产生不同结果。

家长可以问：“敌人没有眼睛，程序怎样知道前面是悬崖？”



第5节精品视觉：敌人算法与碰撞分类

这节课怎样把知识变成能力

- **孩子会做：**为敌人设置移动方向，持续检查墙壁与平台边缘，在条件成立时反转速度；再分别实现从上方踩踏和从侧面碰撞的结果；
- **老师会检查：**敌人能否稳定巡逻、不从边缘掉落，并让孩子分别测试上方、侧面和平台边缘三种情况，说明程序为什么选择不同分支；
- **最后能解释：**敌人看似“会思考”，其实是在重复执行移动、检查和转向规则。公平挑战来自清楚条件、可预判危险和一致的碰撞结果。

第5节知识拆解：敌人会巡逻的秘密

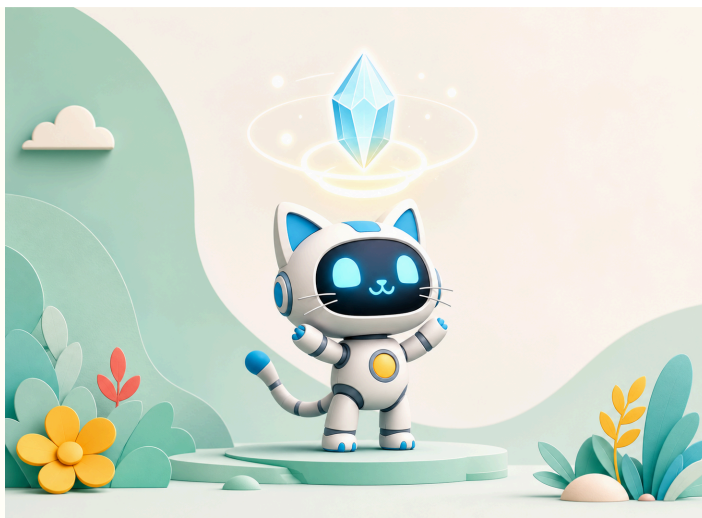
这张图在讲什么

- “会巡逻”来自重复规则。移动、检查墙或边缘、转向，然后再次检查；
- 转向本质是速度改变。水平速度由正变负，同时翻转动画；
- 碰撞也要分类。从上方接触可踩败敌人，侧面接触则让角色受伤。

孩子真正掌握的标准：他能用一句条件逻辑解释敌人为什么转向，并能说明程序怎样区分“从上方踩到”和“从侧面撞到”。

家长课后可以问：“敌人没有眼睛，它怎么知道前面是墙或悬崖？”孩子若能回答“程序不断检查某个条件”，就理解了行为背后的算法。

第6节：Power-ups



第6节视觉示例：拾取前后，角色能力与反馈都发生变化。

第六节把“状态”讲得更清楚：角色从 normal 变成 powered，行为才随之变化。孩子可选择成长、加速或二段跳，但必须找到“状态在哪里被改变、行为在哪里检查状态”这两个位置。

本节一眼看懂

- **学习：** 游戏状态、状态切换、能力边界与反馈设计；
- **动手：** 选择一种能力，设计道具与短音乐，加入粒子、闪光或缩放；
- **AI 与孩子：** AI 起草状态代码，孩子决定能力边界并指出“改变”和“读取”状态的位置；
- **看得见的成果：** Power-up 明显改变角色能力，并有同步的视觉与声音反馈。

这节课也会说明“polish（手感打磨）不是有空才做的装饰”。反馈清楚，玩家才知道自己的动作产生了什么结果。

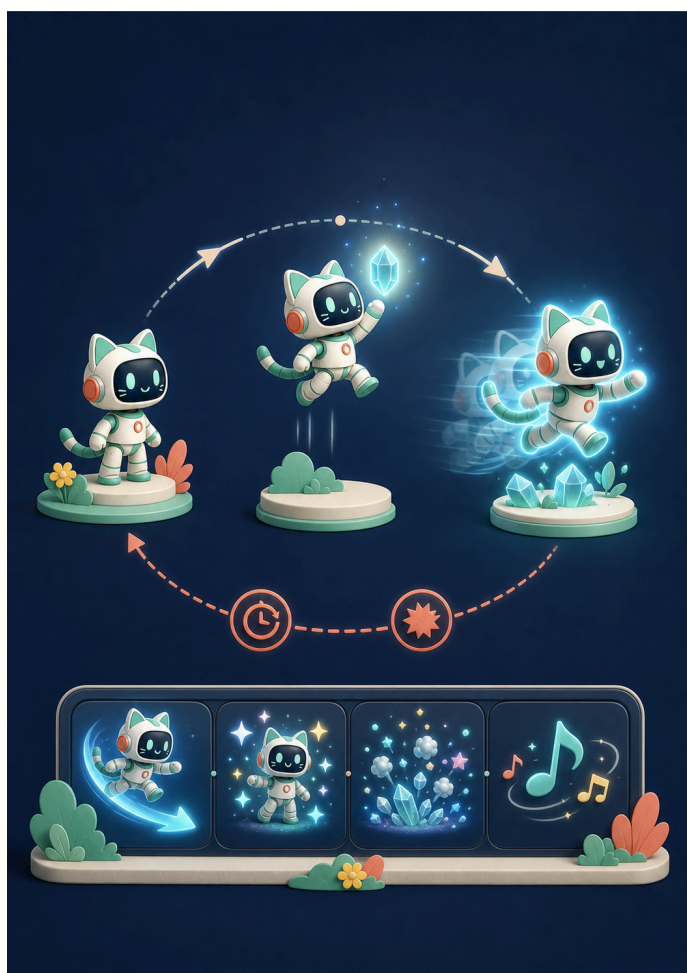
第6节精品视觉：状态机、能力变化与反馈设计

左侧看懂这张图

- **状态：** hero_state
记住角色当前是否处于强化状态；
- **切换：** 拾取时进入，
计时结束或受伤后退出；
- **反馈：** 能力、外观、
粒子和声音需要同步改变。

当堂证据： Power-up 会改变实际能力，而且强化状态能正确开始与结束。

家长可以问：“程序在哪里记住角色已经变强了？”



第6节精品视觉：能力状态切换

这节课怎样把知识变成能力

- **孩子会做：** 选择一种明确能力，把角色从 normal 切换到 powered，并让移动、跳跃或外观读取这个状态；计时结束或受伤后再恢复；

- **老师会检查：** 状态是否只有清楚的进入与退出路径，能力变化是否真实生效，以及粒子、颜色和声音是否与规则变化同时出现而不是只做装饰；

- **最后能解释：** 状态是程序对“现在发生什么”的记忆。改变状态、读取状态和恢复状态必须互相对应，否则角色可能永远强化或画面与能力不一致。

第 6 节知识拆解：角色怎样记住自己变强了

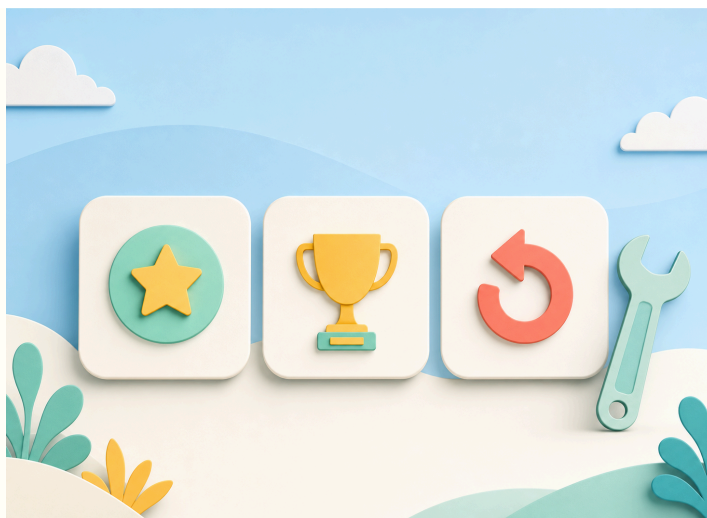
这张图在讲什么

- 状态是程序的记忆。hero_state 记录角色当前是 normal 还是 powered；
- 切换必须有清楚的起点和终点。拾取后变强，计时结束或受伤后恢复；
- 规则和反馈要一起变。能力、外观、粒子和声音同时改变，玩家才知道新规则生效。

孩子真正掌握的标准：他能指出哪一行改变状态、哪些代码读取状态，并解释为什么视觉变化必须和能力变化同时发生。

家长课后可以问：“角色变强以后，程序在哪里记住这件事？受伤以后怎样恢复？”这能判断孩子是否理解状态的生命周期，而不只是看见一段特效。

第7节：Win, lose & polish



第7节视觉示例：把开始、游玩、胜利、失败和重试连成完整循环。

第七节是整合周。旧功能在整合时可能被新功能影响，所以课程会预留明确的 Fix-it Buffer。孩子会学到：出现问题不是项目失败，而是工程计划中需要诊断的一部分。

本节一眼看懂

- **学习：**完整 game loop、HUD、胜负条件、重置与整合调试；
- **动手：**接入分数和生命，制作 HUD、终点、胜利/失败画面与重试；
- **AI 与孩子：**AI 提供连接和修复建议，孩子测试胜利、失败和第二次重玩；
- **看得见的成果：**游戏可通关、可失败、可重试，上一局状态不会留到下一局。

家长看到孩子“修 bug”时，不必把它理解成课堂没准备好。整合后发现冲突、缩小问题范围并验证修复，本身就是软件项目里非常真实的能力。

第7节精品视觉：游戏循环、界面、胜负与重置

左侧看懂这张图

- 闭环：title → play → win/lose → reset；
- 界面：HUD 把分数和生命等内部状态显示给玩家；
- 重置：角色、敌人、金币与能力都要回到初始值。

当堂证据：游戏可以通关、失败和重试，第二局不会残留上一局状态。

家长可以问：“重新开始时，除了角色位置，还要恢复什么？”



第7节精品视觉：完整游戏循环

这节课怎样把知识变成能力

- 孩子会做：把标题、游玩、胜利、失败和重新开始连接成闭环，并让HUD实时显示分数和生命；之后连续测试胜利、失败与第二次重玩；
- 老师会检查：每条路径能否独立完成，点击重试后角色、敌人、金币、分数、生命和Power-up是否全部恢复，不留下一局的隐藏状态；
- 最后能解释：完整游戏不只是“能操作”。系统需要开始、目标、失败、反馈与重置；任何一个环节断开，玩家都无法完成可靠的游戏循环。

第7节知识拆解：怎样把零散功能连成完整游戏

这张图在讲什么

- 游戏必须形成闭环。title → play → win/lose → reset → title；
- HUD 是系统状态的窗口。分数和生命要随游戏实时更新，但不能遮挡操作；
- 重置是一项系统工程。角色、分数、生命、敌人、金币与能力都要恢复。

孩子真正掌握的标准：他能画出完整游戏循环，主动测试胜利、失败和第二次重玩，并解释重置时哪些数据必须恢复。

家长课后可以问：“点击重新开始时，除了角色位置，还有哪些东西必须回到初始状态？”若孩子能想到分数、生命、敌人、金币和 Power-up，就理解了系统整合。

第 8 节：Ship it & Demo Day



第 8 节视觉示例：完成试玩、修复、发布和作品讲解。

最后一节几乎不再讲新知识，而是像一个小型工作室发布日。孩子先根据清单选择最重要的两项修复，尽早发布，再用他人的设备验证链接，最后进行作品讲解。

本节一眼看懂

- **学习：** 试玩证据、问题描述、优先级取舍、发布与跨设备验证；
- **动手：** 选择 Top-2 修复，完整测试，发布链接并制作标题卡和短预告；
- **AI 与孩子：** AI 作为 debugging partner，孩子决定先修什么并验证修改；
- **看得见的成果：** 可分享链接能在另一台设备打开，并完成一次 Demo Day 展示。

“发布”不代表作品以后不能再改，而是孩子学会在有限时间内决定什么必须完成、什么可以留到下一版。

第8节精品视觉：测试、调试、发布与作品表达

左侧看懂这张图

- **证据：**记录预期、实际结果和复现步骤；
- **取舍：**优先修复影响最大的Top-2问题；
- **发布：**公共链接还要在另一台设备上验证。

当堂证据：另一台设备能打开最终链接，孩子也能说明最后两项修复的理由。

家长可以问：“为什么先修这两个问题，而不是继续加功能？”



第8节精品视觉：从试玩证据到正式发布

这节课怎样把知识变成能力

- **孩子会做：**观察真实试玩者，记录预期结果、实际结果和复现步骤；再按影响范围与剩余时间选择最重要的Top-2问题进行修复；

- **老师会检查：**修复前后是否有可比较证据，公开链接能否在另一台设备正常打开，以及孩子能否清楚介绍角色、核心玩法和一个关键设计决定；

- **最后能解释：**发布不是按下一个按钮，而是完成取舍、验证与表达。能说清楚“为什么先修这个”，比在最后时刻继续添加新功能更接近真实项目工作。

第 8 节知识拆解：从我觉得可以到真正发布

这张图在讲什么

- “我觉得没问题”不是证据。需要观察真实玩家并写出预期、实际和复现步骤；
- 有限时间要会取舍。先修影响最大的 Top-2，发布前停止增加大功能；
- 本地运行不等于发布。公共链接必须在另一台设备上再次验证。

孩子真正掌握的标准：他能用证据选择修复优先级，写出清楚的问题描述，并让另一台设备从公共链接成功打开游戏。

家长课后可以问：“你最后为什么先修这两个问题，而没有增加新功能？”孩子能说明影响范围、时间与发布风险，就开始理解真实项目管理。

AI 到底做什么，孩子又必须做什么

家长担心“作品是不是 AI 代做”很合理。课程会把 AI 的帮助和孩子的责任分开检查。

创作线	AI 可以帮助	孩子必须负责
AI Art	提供主角、敌人、地块、道具、背景和 UI 草案	描述方向、比较结果、保持一致性、拒绝不合适版本
AI Music	提供主题曲、Power-up 与胜利短音乐草案	选择情绪、节奏和长度，判断是否适合自己的世界
AI Sound FX	提供跳跃、金币、踩踏等声音草案	筛选、调音量、接到正确事件并实际试玩
AI Coding	起草或解释移动、物理、敌人、状态和界面逻辑	先读再运行、预测效果、描述问题、验证修改、改成自己的版本

老师不会只看“画面有没有成功”。更重要的证据是孩子能否回答：哪一行控制这个行为？改了哪个参数？为什么保留这个版本？如果结果不对，下一步怎么查？

老师怎样判断孩子真的在学习

专业课程不应该只在最后看一个漂亮作品。八节课会留下多种过程证据：

- **启动检查：** 每节先确认上节成果仍能正常打开；
- **当堂 checkpoint：** 每个关键制作块结束时都有可运行检查；
- **口头解释：** 孩子需要用自己的话说明代码、参数、状态或设计选择；
- **对比测试：** 一次只改一个变量，比较修改前后差异；
- **同伴试玩：** 观察真实玩家在哪里卡住，而不是只由制作者自己判断；
- **两条路径测试：** 胜利和失败都要运行，不能只测试最顺利的一条；
- **跨设备验证：** 最终链接必须在另一台设备打开；
- **作品表达：** Demo Day 说明英雄是谁、危险是什么、自己做了哪一个关键决定。

不同孩子的美术风格、关卡难度与功能细节不会完全相同。共同标准是核心游戏循环能够运行，孩子能说明自己的关键选择，并能根据测试证据继续修改。

报名前怎样判断是否适合

如果孩子喜欢游戏，但您不确定他是否已经准备好，下面的判断会比“以前学过几年编程”更有用。

比较适合

- 10-16 岁，喜欢游戏、角色、音乐、世界观或创意项目；
- 会打字并能阅读基础英文，零代码基础也可以；
- 愿意先读一小段代码，再运行和修改；
- 遇到问题时愿意尝试第二版，而不是只等待正确答案；
- 希望完成一个能试玩、能讲解、能继续迭代的长期作品。

可能需要先聊一聊

- 还不能稳定使用键盘或阅读基础英文；
- 只想玩现成游戏，不愿意设计、测试或调试；
- 期待 AI 一次生成完整作品，不愿意检查修改；
- 已能独立完成完整多文件 2D 游戏，可能需要更高阶挑战；
- 线下场地或时间暂时不确定。

这些情况不代表孩子“不适合学 AI”。您可以先和我们说说孩子的年龄、兴趣、键盘使用情况和过去做过的项目；如果有些信息还不确定，也完全没关系，我们会一起判断当前课程是否合适。

家长常问与下一步

- **需要先会编程吗？** 不需要传统代码基础，但孩子需要能打字、阅读基础英文，并愿意阅读和修正 AI 辅助生成的代码。
- **是不是每个孩子最后都做出一样的游戏？** 核心机制一致，但英雄、世界、音乐、敌人、道具、关卡路线和难度由孩子自己决定。
- **为什么是线下小班？** 角色控制、调参、整合和发布时，老师需要观察孩子怎样读代码、描述问题和验证结果；8-15 人小班也保留同伴试玩与展示。
- **会不会只是 AI 代做？** AI 可以起草，但孩子必须选择、阅读、测试、解释和修改。课堂验收会检查过程证据，不只看最后画面。
- **最后真的有完整作品吗？** 课程目标是完成有原创角色、美术、音乐、音效、敌人、Power-up 与完整胜负循环的可玩项目，并在 Demo Day 展示。
- **是不是复制 Nintendo 游戏？** 不是。孩子制作原创英雄、世界和音乐，不使用 Nintendo 官方角色、Logo、音乐或素材。
- **如果缺课怎么办？** 不同班次的补课方式可能不同。请尽早联系，我们会根据孩子进度与当前安排一起确认最实际的补上方式。

当前课程为 A\$480，共 8 节，每节约 90 分钟。线下场地、开班日期和剩余名额会变化；扫描下一页 **Website** 查看当前课程页，或通过 **Sales** 联系 Rain，一起确认孩子是否适合以及当前报名安排。

NEXT STEP

看完逐课安排，再一起判断孩子是否适合

Website 查看当前课程页；Sales 联系 Rain 确认线下场地、开班日期、剩余名额与报名安排。

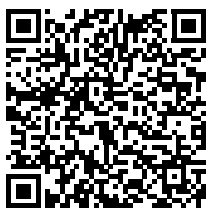
hello@airbotix.ai

Rain · Sales: 0421-672-555

WEBSITE

当前课程页

查看当前公开课程信息。场地、日期、名额或页面内容变化时以官网与报名确认信息为准。



airbotix.ai/programs/super-mario-game ↗

SALES

联系 Rain

咨询线下场地、开班日期、名额、付款与孩子是否适合本课程。



[Rain · Sales 微信](#) ↗