南开大学 信息科学与技术大类

姓名：朱泽帅

学号：2411560

2025年4月27日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 2](#_Toc196677184)

[**一.** **作业题目** 2](#_Toc196677185)

[**二.** **开发软件** 2](#_Toc196677186)

[**三.** **课题要求** 2](#_Toc196677187)

[**四.** **主要流程** 2](#_Toc196677188)

[**1． 整体流程** 2](#_Toc196677189)

[**2． 代码框架** 7](#_Toc196677190)

[封装主要人物：andy类 7](#_Toc196677191)

[封装背景环境的一些类： 7](#_Toc196677192)

[brick类，bullet类，diamond类，fire类，flagsystem类，flower类，master类，pipe类，weed类，mushroom类，star类 7](#_Toc196677193)

[游戏关卡： 7](#_Toc196677194)

[Level1类，Level2类，Level3类 7](#_Toc196677195)

[控制背景音：musicplayer类 7](#_Toc196677196)

[控制按钮：pushbutton类 7](#_Toc196677197)

[**3． 核心实现** 7](#_Toc196677198)

[代码框架已经给出，下面就第一关为例详细介绍一下实现逻辑： 7](#_Toc196677199)

[**（1） 首先是构造函数level1** 7](#_Toc196677200)

[**（2） Game\_Init** 7](#_Toc196677201)

[**（3） Pause\_Init** 8](#_Toc196677202)

[**（4） timer** 8](#_Toc196677203)

[**（5） 圆形** 8](#_Toc196677204)

[**五.** **收获** 8](#_Toc196677205)

高级语言程序设计大作业实验报告

1. **作业题目**

小恐龙历险记

1. **开发软件**

Visual Studio 2022以及QT

1. **课题要求**

学生自选题目，使用C++语言完成一个图形化的小程序。图形化平台不限，可以是MFC、QT等任何C++图形化平台，完成一个小游戏的制作。

1. **主要流程**
   1. **整体流程**
      1. 主界面：

主界面类：mainwindow

在主界面之中有4个按键：

Start Setting Help Exit

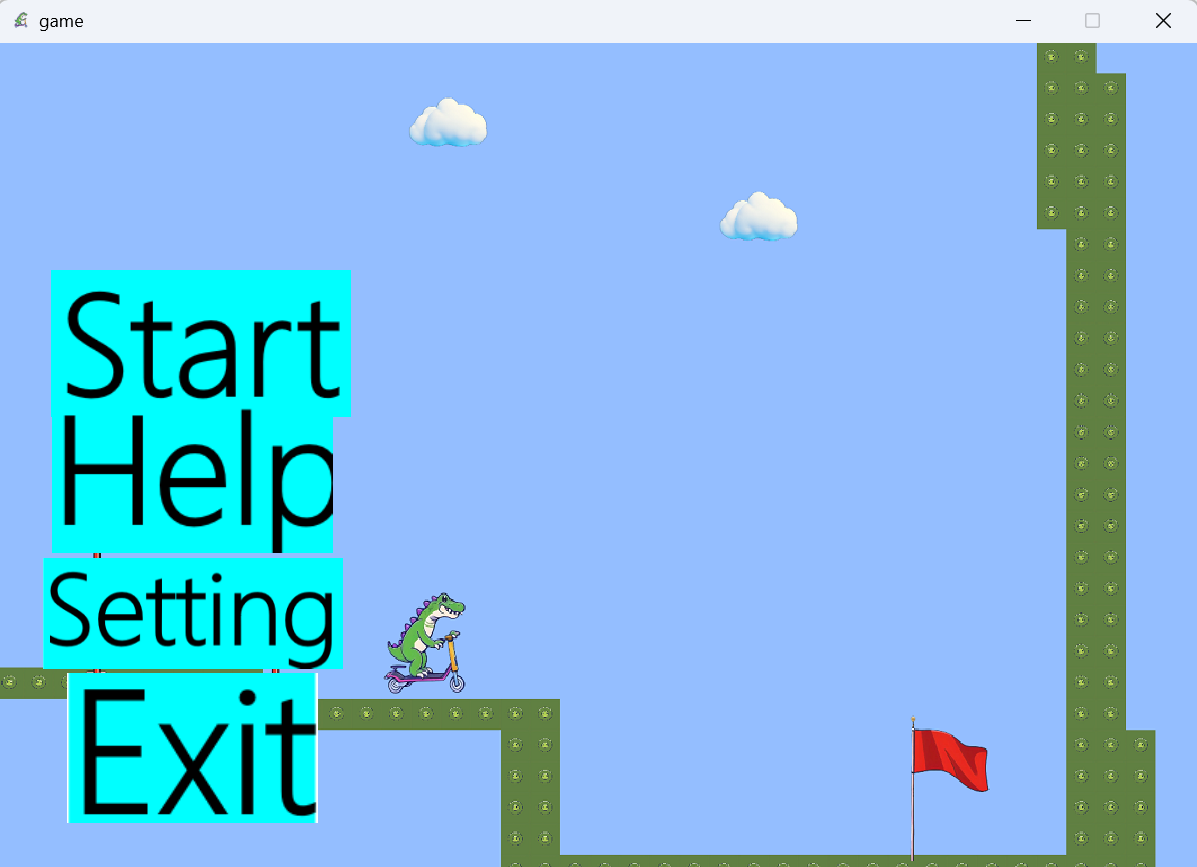
点击Start开始游戏

点击Setting可以选择打开音乐或者关闭音乐

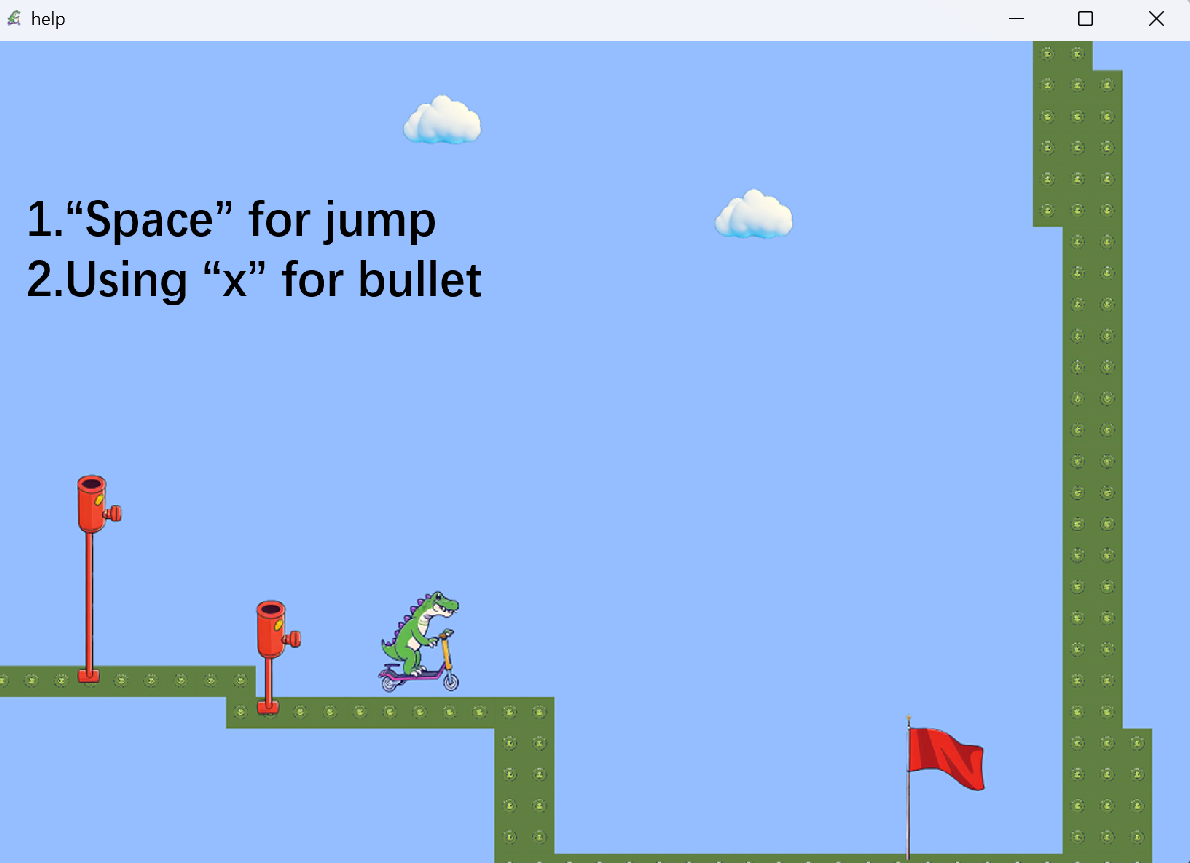
点击Help可以查看操作设置

点击Exit退出游戏

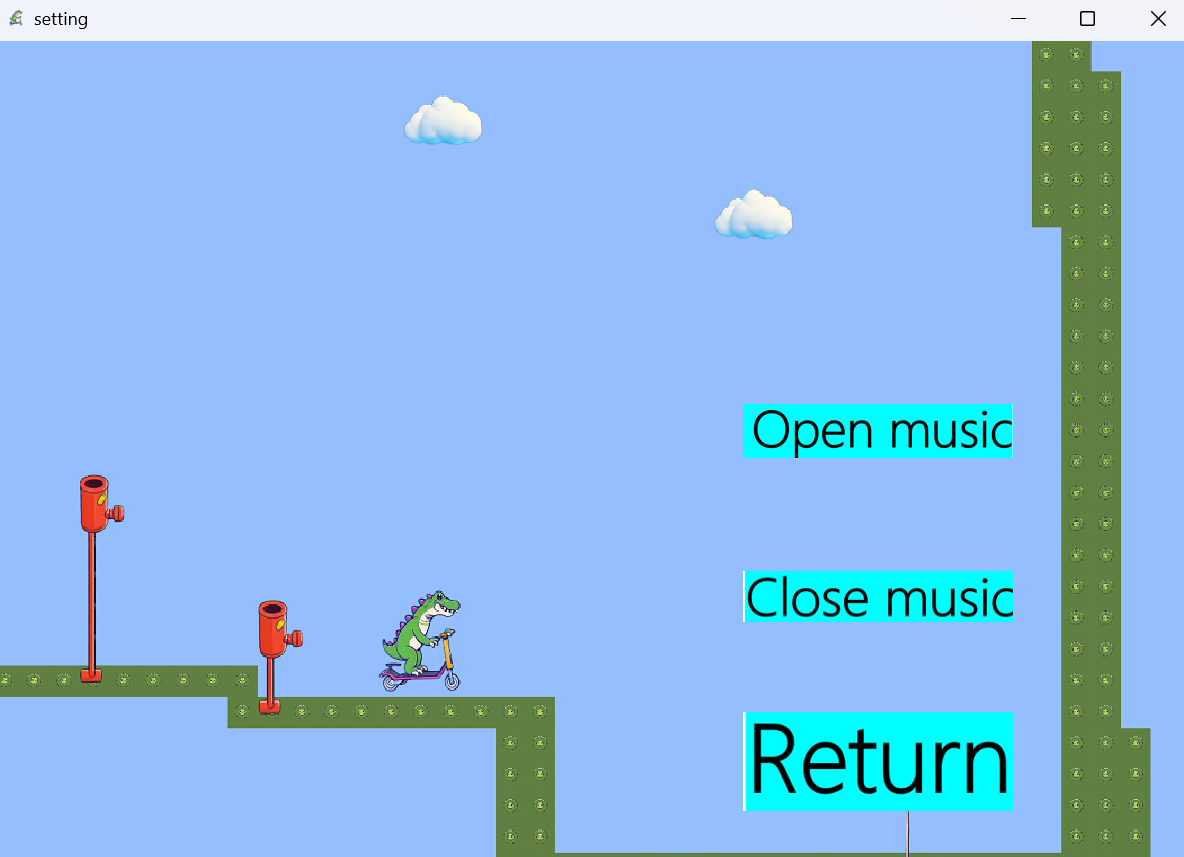
初始界面：



Help界面：



Setting界面：



* + 1. 关卡：

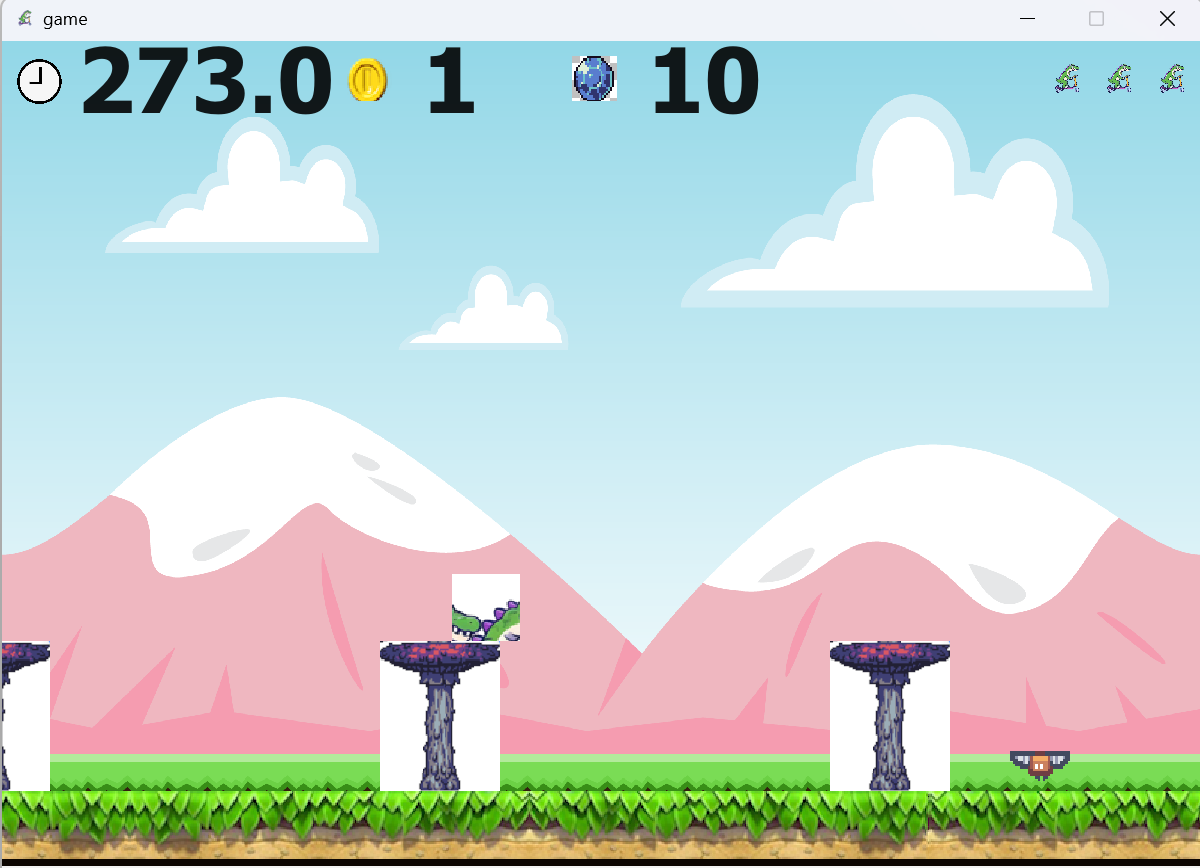
游戏共设三关，依次为 Level 1、Level 2、Level 3，需按顺序逐关挑战，完成前一关方可解锁下一关，整体难度随关卡等级逐步提升。

Level 1 限时 300 秒，玩家需在规定时间内完成通关。进入 Level 2 后，限时缩短至 200 秒，同时场景内怪物的出现频率显著提升，挑战难度较前一关明显增加。

来到最终的 Level 3，限时仅 100 秒，此关卡除怪物出现频率进一步提高外，还将新增随机生成的地刺类陷阱。当小恐龙不慎触碰地刺时，会直接触发扣血机制，玩家需在更紧凑的时间限制下，同时应对高密度的怪物群与致命的地刺陷阱。

游戏界面：

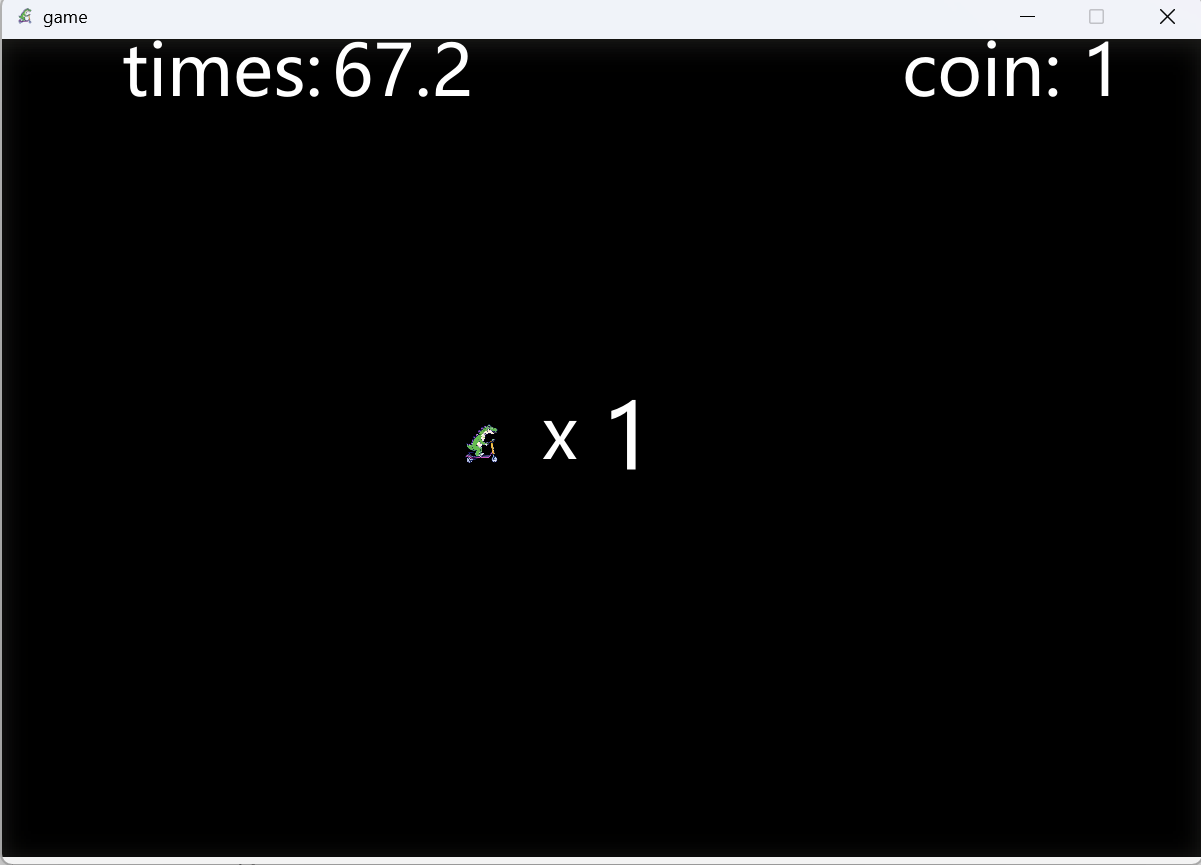
第一关：



第三关：



加载画面：



结束画面：



* + 1. 其它部分：

按钮以及播放音乐

* 1. **代码框架**

**封装主要人物：**andy类

**封装背景环境的一些类：**

brick类，bullet类，diamond类，fire类，flagsystem类，flower类，master类，pipe类，weed类，mushroom类，star类

**游戏关卡：**

Level1类，Level2类，Level3类

控制背景音：musicplayer类

控制按钮：pushbutton类

* 1. **核心实现**

代码框架已经给出，下面就第一关为例详细介绍一下实现逻辑：

* + 1. **首先是构造函数level1**

在level1中，先调用了Game\_Init()初始化游戏：初始化游戏逻辑，比如人物位置、敌人、障碍物、道具之类的初始数据。

timer1 = startTimer(15): 启动一个周期是15毫秒的定时器（高频率的更新，比如主角动作刷新）。

timer3 = startTimer(50): 启动一个周期是50毫秒的定时器（用于控制敌人移动、生成物体等中等频率事件）。

game\_start = true: 标记游戏正式开始，可以响应输入、更新画面了。

* + 1. **Game\_Init**

在游戏初始化的这个函数中，创建了andy对象，brick等环境对象，并且把他们初始化。

* + 1. **Pause\_Init**

它的作用是：当游戏处于暂停状态重新开始时，重新初始化游戏状态和元素。  
和刚才的 Game\_Init类似，但它不是完全重新创建对象，而是重置已有对象。

* + 1. **Timer**

如果触发 timer1，而且主角 andy 已经死亡，执行 Andy\_die()，处理死亡动画或状态更新，同时停止背景音乐，播放死亡音乐。然后死亡初始化，调用 Die\_Init，处理死亡后重生或者Game Over。timer3 是一个倒计时控制器，同时附带一些状态更新，每50ms触发一次。总的来说：

| **计时器** | **作用** |
| --- | --- |
| timer1 | 每15ms刷新整个游戏，包括移动、跳跃、碰撞检测等. |
| timer2 | 辅助处理移动。 |
| timer3 | 控制倒计时（time变量），每50ms减少0.05秒，同时更新敌人状态和砖块状态。 |

* + 1. **Key**

**键盘按键处理：**

右键：向右移动

左键：向左移动

Z：加速冲刺

Space：跳跃

C：加一条生命

X：发射火球

* + 1. **paint**

若游戏还没开始（暂停/开始界面）：画一个简单的静态背景+一些基础信息（比如生命、时间、金币数）。画完后就 return，不会继续往下画动态内容。

游戏开始时：画天空背景，小图标，数值，生命值，怪物，砖块等环境背景。

* + 1. **Jump\_Collision（跳跃检测）**

先判断andy跳跃动作是否结束，若跳跃没结束，遍历所有的砖块，判断是否撞到了某个普通砖块。水平方向上：判断距离是否在±30范围内，若满足，说明在水平方向上andy和砖块对齐。竖直方向上：判断和 andy 的头（andy的y坐标+40）是否在垂直方向上距离合适。碰到砖块，+5分，播放砖块碎裂声，然后重置跳跃高度防止继续往上。问号砖块同理。

总而言之：在 andy 跳跃时，检测是否顶到了砖块或者问号砖块，如果撞到了，就根据砖块类型执行不同的效果（碎砖、出金币、出蘑菇），并调整角色状态。

* + 1. **Move\_Collision（左右移动检测）**

遍历所有普通砖块，并且记录砖块的坐标和状态。如果 andy 向右移动，并且砖块在他右边35到40像素的地方，且垂直方向上位置差不多对齐。说明砖块是实体砖块，则禁止继续移动，马上退出函数。andy 向左移动同理。

如果andy靠近了终点，水平方向上：在一个大概 x方向-60到-20像素的范围内，且竖直方向差不多在andy上方100-200像素左右，说明andy到达了终点。

* + 1. **Fall\_Down（降落检测）**

andy高度减去andy已经跳过的距离小于0，说明andy正在下落。如果andy的 y 坐标（垂直位置）已经低于地面（设地面在455像素处），说明掉到了地上。把andy的y坐标固定在地面，然后结束跳跃状态。接着跳跃参数复原，准备下一次跳跃，然后返回，不继续执行后面代码。

然后判断是否与砖块碰撞：

水平方向上：砖块的x坐标在andy左右30像素以内，竖直方向上：砖块的y坐标在andy下方25~50像素之间，并且砖块是实体砖块。

如果满足这些条件，就让andy站到砖块上。判断是否落在管道，其它砖块也是同理。

* + 1. **Die\_Init（死亡初始化）**

核心：死亡检测，复活重置，无敌管理。

死亡检测：调用isdie检测是否死亡，若andy死亡，并且有按键按下过，那么把andy的走路状态清零，停止走动，然后把按键状态清空，防止继续输入。

复活管理：如果还有命，停掉3个timer，并且暂停游戏。然后重置死亡标志。

开启无敌状态，重新启动3个定时器，游戏重新开始。并且把死亡动画状态重置，播放重启音乐。

无敌管理：如果无敌时间计数达到了100帧，2秒左右，那么，取消无敌状态并且重置无敌时间，如果andy当前是无敌的，那么每次无敌时间+1

* + 1. **level1\_Win**

即玩家在 level1 中获胜，则进入以下的胜利处理流程：

设置一个延迟1秒的定时器，在1秒后停止游戏，调用 update() 函数来重新绘制窗口或UI，更新游戏状态。创建一个新的 level2 对象，并且隐藏当前的 level1 界面，显示 level2 界面，表示游戏从第一关 (level1) 过渡到第二关 (level2)。

1. **收获**

## 面向对象设计与模块化思维

将角色（Andy）、背景元素（砖块、管道、蘑菇等）、关卡逻辑、音效控制、按钮交互等拆分到各自的类里，提高了代码的可读性和可维护性。通过 Level1、Level2、Level3 等关卡类，直观地实现关卡切换和复用同一套框架逻辑。

## Qt 事件驱动与定时器机制

掌握了 QTimer::singleShot 和 startTimer/killTimer，在不同频下更新游戏逻辑、倒计时和特殊动作。

结合 keyPressEvent/keyReleaseEvent 与 paint，形成完整的“用户输入——状态更新——画面重绘”循环。

## 碰撞检测与物理交互

在 Jump\_Collision、Move\_Collision、Fall\_Down 中分别管理跳跃顶砖、左右阻挡、下落支撑，积累了“基于坐标区间判断”实现简单物理交互的经验。

通过分支控制不同类型砖块（如：砖块）的响应逻辑，初步掌握了游戏中“碰撞分发”的思路。

## 状态管理与复活逻辑

借助 Die\_Init 实现了“死亡检测—暂停—扣血—复活—无敌保护—恢复”流程，学会了如何在游戏中插入延迟与状态过渡。

## 5. 整体架构与可扩展性

学会了采用“Game\_Init vs Pause\_Game\_Init”区分“全新初始化”与“复用已有对象重置状态”的思路，为后续添加更多关卡或新道具提供了模式。未来如要加入新道具（比如星星等），只需按现有模板增添类并在初始化、碰撞、绘制中加入调用，就能快速扩展。