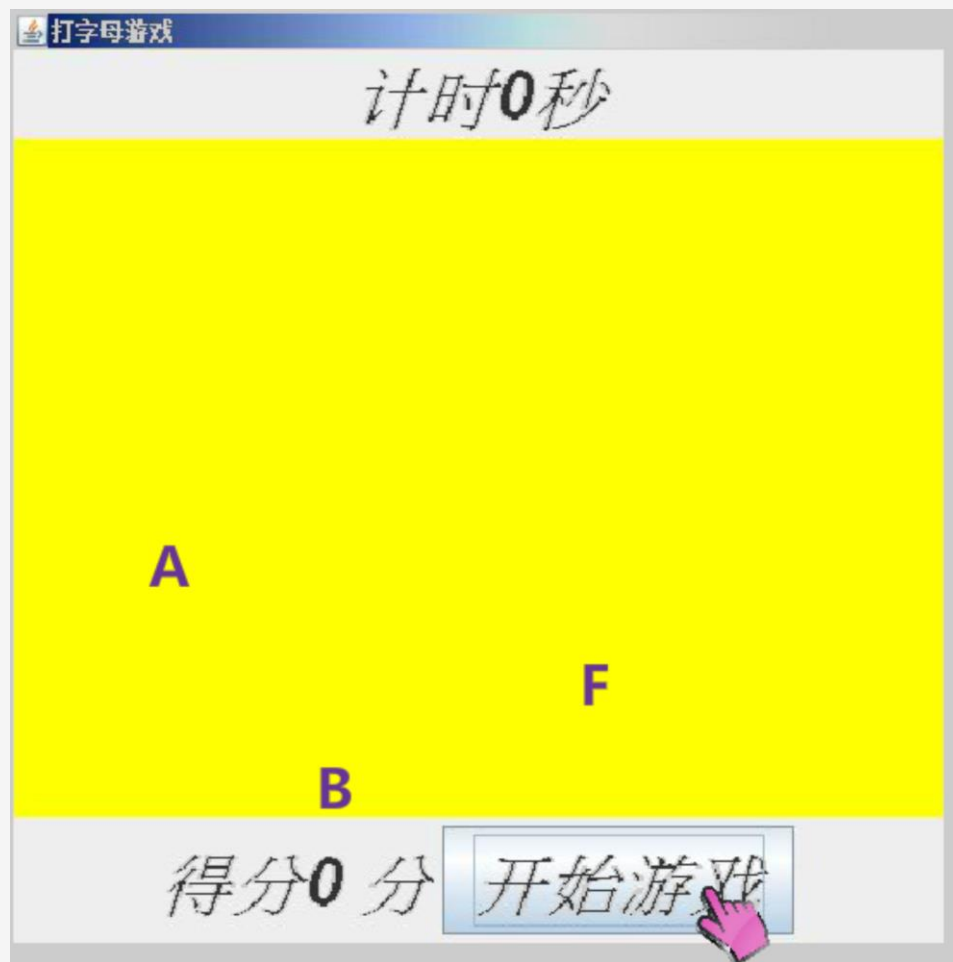


学习任务二 开发小游戏

子任务2.5 游戏分析与设计

游戏需求分析：画类图

理解游戏设计需求，设计类图



在这个游戏中：

- 1、有几个角色？
- 2、每一种角色有什么属性？
- 3、每一种角色有什么方法？

游戏需求分析：画类图

理解游戏设计需求，设计类图

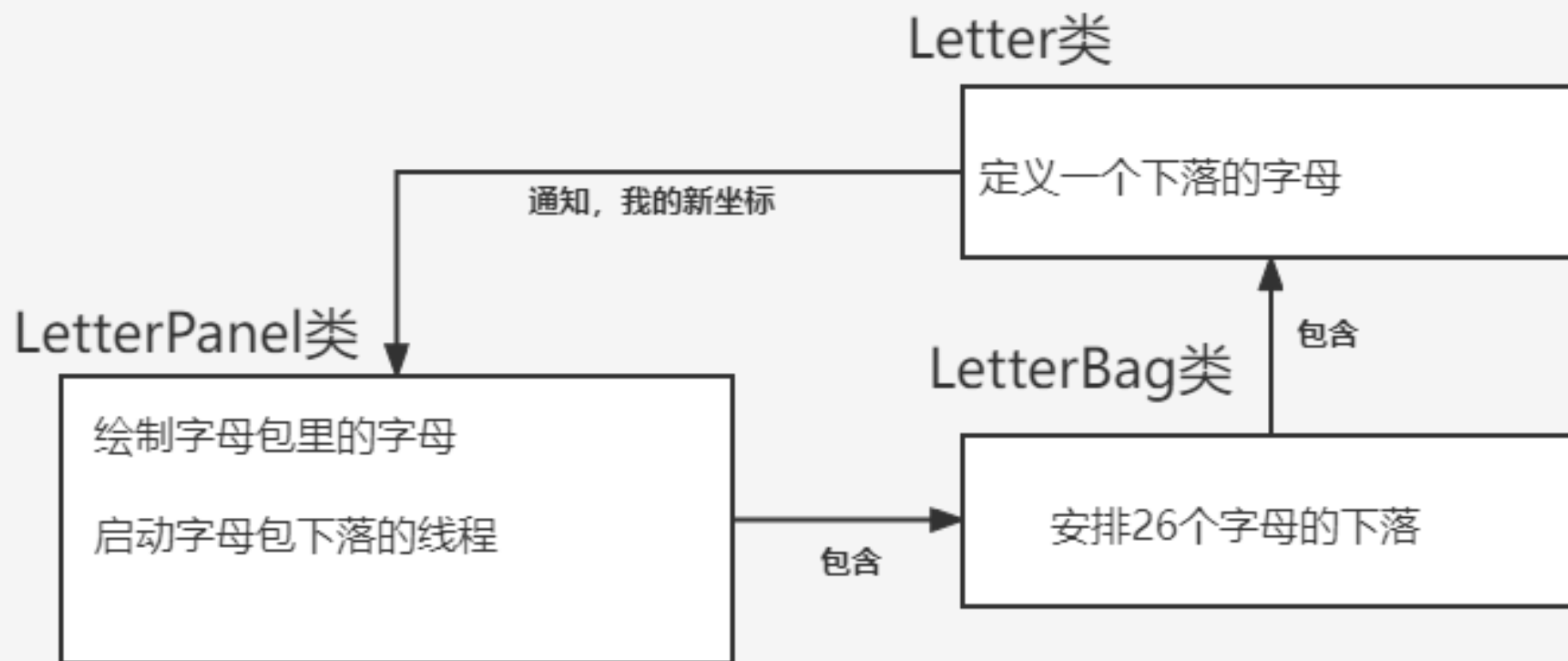


在这个游戏中：

- 1、界面类角色：主面板、字母面板、计时面板
- 2、字母包角色——一个装26个字母的集合
- 3、字母角色——从天而降的26个字母，依次随机的下落

游戏需求分析：画类图

理解游戏设计需求，设计类图



游戏需求分析：画类图

理解游戏设计需求，设计类图

Letter类

▲ h	字母运动的高度范围
▲ letter	字母
▲ status	字母的状态：四种状态
▲ step	字母运动的步长
▲ w	字母运动的宽度范围
▲ x	字母运动中的x坐标
▲ y	字母运动中的y坐标

- move() : void
- resize(int, int) : void
- start() : void

字母下移一次的动作

重置字母的运动范围，重新定义x坐标

启动下落线程

在Letter类中：

- 1、属性都是和单个字母相关的一些数据和对象
- 2、方法都是和一个字母的活动轨迹相关的行为

move确定一个字母移动一次要做什么

resize确定当屏幕的尺寸被调整一个字母要做什么

改变

start用来启动一个字母的下落

游戏需求分析：画类图

理解游戏设计需求，设计类图

LetterBag类

- LetterBag(int, int, Callable) 构造函数
- falling() : boolean 字母下落一次要做什么
- hasFalingLetter() : ArrayList<Letter> 获得正在下落的所有字母
- hasReadyLetter() : ArrayList<Letter> 获得准备好的字母
- resize(int, int) : void 如何调整所有字母的运动范围

在LetterBag类中：

- 1、他的主要责任是管理要下落的26个字母，所以他是一个集合
- 2、提供的方法，用来处理26个字母的动作

falling确定一次下移，字母包需要放一个字母出来

两个hasxxx方法为falling服务，用来查询字母包中下落和准备好的字母

resize用来处理屏幕尺寸如果调整之后，26个字母要共同做什么调整

游戏需求分析：画类图

理解游戏设计需求，设计类图

LetterPanel类

 Callable	回调类
 gHidth	面板的高度
 gWidth	面板的宽度
 call	回调器对象
 gameover	游戏是否结束
 list	字母包
 score	得分
 hit(char) : int	判断用户是否敲中字母
 paintComponent(Graphics) : void	把字母画到面板上
 resize() : void	处理当界面的大小改变时，字母的位置如何调整
 start() : void	启动一个线程，每隔一段时间让字母包处理下落一次

在LetterPanel类中：

- 1、Callable是回调类，沟通字母和面板
- 2、其他属性和字母面板的状态控制有关
- 3、hit用来处理如果用户敲击了一个字母，字母包如何判断是否加分
- 4、paintCompment主要是绘制字母
- 5、start用来启动线程，每隔一段时间启动新的字母下落

游戏的实现：完成代码的编写和调试

完成四个类的设计与调试：

- 1、Letter类**
- 2、LetterBag类**
- 3、LetterPanel类**
- 4、Main类**