****

**详细设计说明书**

**组名：减压工作室-微信小程序开发**

**组号：17**

**组员：16020520040 郭宣威**

**16020520041 方振轩**

**16020520042 韦欧睿**

**16020520043 吴 彬**

**一、代码总体结构设计**

根据我们之前做出的概要设计说明书，其中的游戏运行模块是关键因素。因此，我们组从这一模块为出发点，再结合其他模块的概要设计对代码的结构做出了总体设计。

下图即为我们的代码总体设计图：

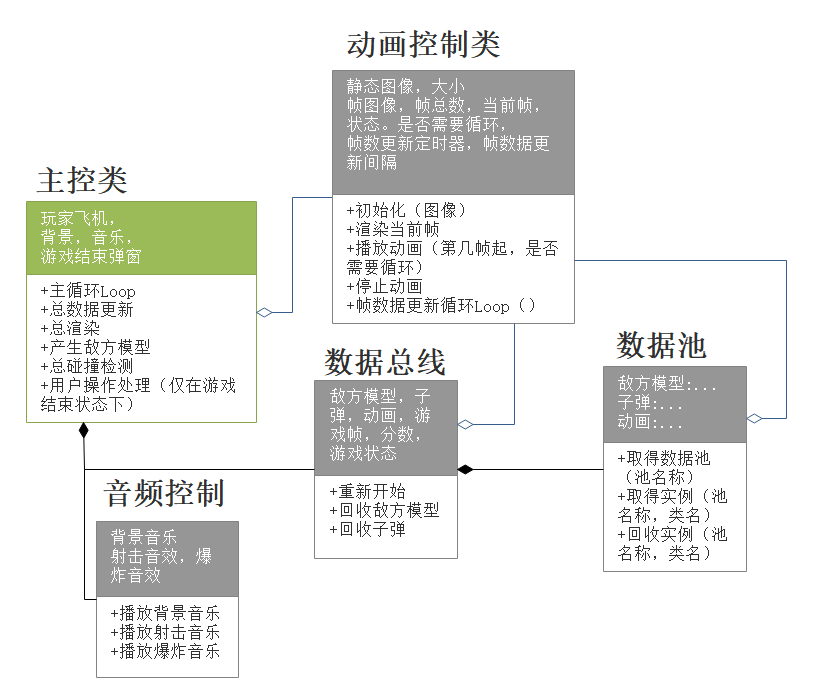


图1 静态类图1

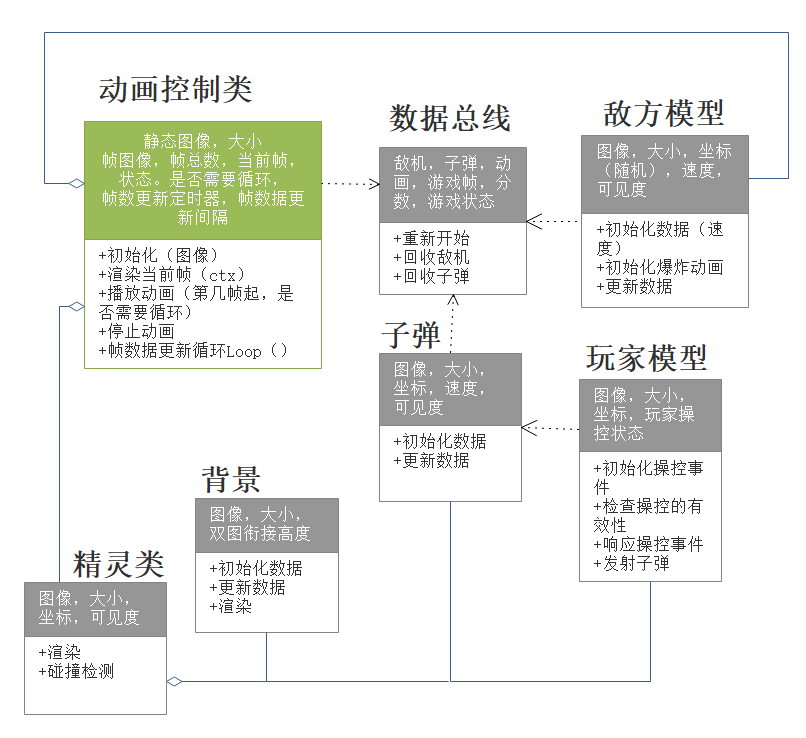


图 2 静态类图2

**二、代码设计的具体化阐释**

由上图可知，我们组将游戏运行模块的组成部分主要分为以下这几个类：

1.**主控类**：真正的入口与主控类，包括了总数据更新、总数据渲染、总玩家操控处理。

a.总数据更新：包括了调用背景、子弹和敌机的数据更新（主要也就是坐标更新），以及敌机模型的生成、总碰撞检测。

b.总数据渲染：包括了调用背景、子弹和敌机的数据渲染，还有玩家飞机、敌机爆炸动画的渲染，以及“游戏结束”的弹窗的渲染。

c.总玩家操控处理：仅负责处理对游戏界面层的操控（目前只有游戏结束弹窗），而把对玩家飞机操控事件的处理留给了玩家飞机类。

注：在这里，因为玩家飞机的数据(目前只有坐标)，是由用户操控处理逻辑在决定着，因此玩家飞机只需渲染不需数据更新。

2. **玩家模型类:** 包括了用户操控处理（包含数据更新）、数据渲染，以及子弹的生成。

a. 玩家操控处理：主要是飞机周边30像素内的触摸都算作有效操控、改变坐标，及阻止超越地图边界的移动。

b. 数据渲染：在当前坐标上渲染图片。通过继承精灵类来实现。

3.**敌方模型类:** 包括了数据更新、数据渲染，以及初始化敌机爆炸动画。

a. 数据更新：y坐标由上至下改变、及越过边界时回收自己。

b. 数据渲染：通过继承精灵类实现。

4. **子弹类：**包括了数据更新、数据渲染。

a. 数据更新：y坐标由下至上改变、及越过边界时回收自己。

b. 数据渲染：通过继承精灵类实现。

5.**背景类：**采用一张无缝衔接背景图实现无限滚动，包括了数据更新、数据渲染。这是比较次要的一部分。

a. 数据更新：将图片衔接位置的y坐标由上至下改变。

b. 数据渲染：重载了精灵类的渲染方法，使得图片能沿着衔接位置上下渲染两次。

6.**精灵类：**所有功能层面实体类的基本类。它维护着图片、大小、坐标，包括了数据更新、碰撞检测。

a. 数据渲染：将图片按照大小、坐标，渲染到给定的画布。

b. 碰撞检测：根据大小、坐标，判断两个精灵是否碰撞。

7.**动画类：**它继承了精灵类，为此维护着一张静态图片、大小、坐标，同时还维护着动画所需的图片数组、当前帧、动画播放状态等数据。

8.**数据总线类与数据池类：**我们组考虑到为了避免不必要的实例创建或者销毁的操作，设置敌机、子弹的所有实例将只会存在于数据总线或数据池队列的任何一个中。（换句话说就是：已经失效的实例，将会被回收到数据池，而不是真正引发系统的垃圾回收。再次需要实例时，会优先从数据池中取，而不是引发内存申请。除此之外，数据总线还维护着其他游戏全局数据，如游戏状态、得分、动画数组、甚至是当前游戏帧。

a. (数据池)申请实例：从相关队列头部取出，如已为空则创建实例。

b. (数据池)回收实例：实例排回到相关队列尾部。

c. (数据总线)回收子弹、敌机：通过调用自己的数据池实现。

**三、一些小改进**

经过之前的具体化设计步骤后，我们小组认为游戏的基本模型已经产生，并且在经过许多次模拟演示后，在之前设计的代码基础之上增加了一些功能，使得游戏的趣味性提高。

下面就是我们的功能增加具体改进：

**1. 游戏设定类、数据更新主循环与渲染主循环的改造：**

a. 游戏设定类：为方便调试各种功能，例如切换高速弹、调整数据更新频率，我们把相关变量交给一个设定类统一管理，使用静态变量即可。

b. 数据更新主循环与渲染主循环的改造：我们通过改进，使得实体类不再需要通过自己激活来更新循环，而是在主控类中，就有一套数据更新主循环与一套渲染主循环，调用各游戏实体的更新与渲染。

**2. 玩家操控处理的改造:**

在之前的设计中，我们组认为主画面与玩家飞机的操控处理层次较为散乱。因此，在改进过程中，我们将操控处理层设计成界面层、实体层、背景层三层事件响应。

**3.** **游戏设定界面：**

这个游戏设定界面是一种简单的交互界面并且能在游戏中实时修改各种设定。

a. 界面激活：在主界面增加一个设定图标。

b. 界面显示：通过按钮列表即可。

4. **子弹的增强：**

我们增加了新的子弹类型（比如：高速弹、双排弹、三排弹等等）可以通过设定界面来切换，也可以通过拾取漂浮物来更新子弹。

**5.敌方模型的改造：**

我们组认为，敌方模型应该依赖于动画类，被击毁时生成一个爆炸动画的效果。动画类改为在主控类中启动总数据更新循环

**6.漂浮物：**

这是一种新的实体，它具体可以表现为弹药包等等。它与玩家飞机碰撞后触发设定值的变更，并且按随机概率产生漂浮物。