游戏概况：

STG游戏；

1. 战斗系统
2. 道具系统 x
3. 信息显示系统 x

战斗系统架构：

（实体数组：敌人弹幕数组[1000]、玩家弹幕数组[1000]、敌人数组[50]）

// eb\_index = 10;

// eb\_index++; enemy\_b[eb\_index % 1000] = bullet\_A;

// bullet\_A.clean() : enemy\_b[this.index] = null;

//clean; b[

1. 玩家角色（实体）
   1. 位置
   2. 速度
   3. 出生为止时间
   4. 释放的玩家弹幕类型+弹幕初速（时间）
   5. 动作图片帧
   6. 弹幕开关
   7. 碰撞体积
   8. 血量
   9. next函数：
      1. time++
      2. 位置=位置+速度
      3. if (b\_type==0):
         1. if (time%5==0) player\_b[get\_player\_b\_index()] = PB\_TYPE\_B(v, …)
      4. pic\_index = play\_pic[time%10]
2. 敌人
   1. 普通小怪
      1. 实体
         1. 位置
         2. 出生为止时间
         3. 怪物类型
         4. 速度
         5. 索引
         6. （行动逻辑编号）
         7. 碰撞体积
         8. 血量
      2. 类型
         1. 释放的敌人弹幕类型+弹幕初速（时间）
         2. 动作图片帧
         3. 行动逻辑
         4. （掉落道具） x
   2. boss x
3. 弹幕
   1. 敌人的弹幕
      1. 实体
         1. 位置
         2. 出生为止时间
         3. 速度
         4. 弹幕类型
         5. （行动逻辑编号）
         6. 从属的敌人发弹幕时的位置
         7. 从属敌人的索引
         8. 索引
         9. 碰撞体积
      2. 类型
         1. 图片
         2. 行动逻辑
   2. 玩家的弹幕
      1. 实体
         1. ..
4. 结构：
   1. 实体class
      1. next函数
   2. 类型class

战斗流程（while time++）：

* + - 1. 读取键盘（四个keycode，…）
      2. case (keycode): 执行命令
      3. 所有实体next
      4. 判断碰撞（对比数组）：
         1. 人物是否被碰撞：

扫描所有敌人弹幕数组，敌人弹幕位置-人物

…

人物死亡：命-1；所有子弹消失

子弹是否出界：enemy\_b[i]=null;

* + - * 1. 敌人是否被碰撞：

扫描所有玩家弹幕数，弹幕位置-敌人

…

敌人死亡：（掉落道具）；（分数增加）

碰撞：血量减少

1. 画图
   1. 背景设置
   2. 扫描敌人位置+图片
   3. 扫描玩家和敌人弹幕位置+图片
   4. 扫描人物位置+图片
2. 写入sram

底层