

J2ME潜艇大战游戏设计与实现-j2me PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/287/2021_2022_J2ME_E6_BD_9C_E8_89_87_c104_287510.htm 摘要 本文给出了一个飞机射击类(潜艇)游戏的设计过程以及相应的流程图，并提供了Demo版源代码下载地址，游戏在sun wtk v2.2 下调试通过。声明：本游戏所用的资源大多不属于作者，代码仅供非商业用途的学习参考。以下为游戏的几张截图(740)this.width=740" border=undefined>游戏设计1 游戏操作流程：游戏的初始画面是一个impulsive的List列表格式，选择new game以后就可以启动游戏，游戏是以全屏方式开启，但仍有用于暂停/开始和退出的命令响应功能键。在游戏当中启动暂停，海水和海洋生物仍将继续活动，但敌方潜艇和所有鱼雷都将静止下来，玩家的控制也将不起作用，直到游戏被再次启动为止。从游戏状态退出将回到菜单界面，再次选择退出将退出程序。以下是游戏的总体流程图：(740)this.width=740"

border=undefined>2 算法处理游戏自身是一个单态结构(singleton)，由一个Controller来处理目前应该显现的屏幕和状态。主要的显示类包括主画面，游戏画布，帮助画面和一些辅助的Alert，其中最主要是游戏画面。游戏是在MIDP 2的状态下运行的，所以画面继承了GameCanvas。游戏中的角色图层全部加载在一个远大于手机屏幕大小的图层管理器上。当玩家位置移动时，图层管理器同步计算出玩家的相对位置，移动当前图层的显示位置。这个游戏的本质就是在midlet启动后把整个画布作为一个线程，每隔25ms扫描一下用户按键操作，每隔50ms扫描一次敌人和鱼雷等物体的运动状态，

对于比较苛求的子弹射出实时响应，则继承了keyPressed方法。在扫描玩家移动操作时，按键编码被传入到玩家潜艇的响应方法。在扫描敌人和鱼雷运动时，分两步完成。第一步，执行各个物体画面的移动方法。程序并不为每个新增鱼雷或者敌人开启一个新线程，太多线程会造成程序的性能直线下降；而是为每一个物体类型保留一个Vector数组，每当一个新物体产生，他将首先被加入改类型所属的数组中，然后才在图层上画出；当物体消除时，除了在图层上被消除，也需要从响应数组中删除。这样就形成一种注册机制，所有游戏物体的产生和消除都需要向画布注册，画布拥有不同类型物体的所有“名单”，所以，当执行移动命令时，只需要遍历各个类型数组，依次执行其中每个元素的移动方法即可。第二步，就是一般意义上重画，这里使用flushGraphics()替代了传统的repaint()，由于本游戏是以导入图片为操作对象，图片位置的更改已经在第一步被执行，所以没有怎么涉及到像素级的重画，只是在游戏初始化状态时需要加载各个物体数组所对应的图层到新的图层管理器。(由于手上没有rose，只有暂时用EclipseUML的流程图画出草图，若干有问题的地方需要商榷，也希望擅长此到的朋友不吝指正)

740" border=undefined>3 类结构以下列出了主要组成类的含义结构，并以游戏画布为中心列出一个类关联图。Controller: 控制类，用以控制当前屏幕显示哪一个Displayable类。SubObject: 接口类型。玩家潜艇，敌人潜艇以及鱼雷等可移动图层类型皆扩展于此接口。Sub: 玩家潜艇。包含了位置，生命值等成员变量以及移动，开火等方法。EnemySub, MoveableSprite, Tinfish: 敌人潜艇，海洋生物，鱼雷。都是可移动图层类型，

一旦产生就在画布中注册，由时间触发器维护其运动和生命状态。SubCanvas: 游戏主画布。存储着游戏地图信息，游戏状态，玩家信息，同步维护着敌人潜艇，鱼雷，海洋生物的运动和生存状态。 SubCanvas关联类图：740)this.width=740" border=undefined>4 源代码本程序的源代码，包括jad，混淆过的jar和一个rar文件，全部可以在 <http://www.hyweb.net/BrowseFiles.aspx?Folder=Public/My Project> 上下载到，代码中包含了较为详细的注释。这份程序只是一个demo版本，有一些遗留问题（玩家生命值显示，性能优化）还需要进一步完善。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com