

# 潜艇大战

软件 22 王华清 2012013317

软件 22 孙 浩 2012013315

软件 22 王雨婷 2012013328

# 目录

1 开发环境

2 游戏规则

3 设计与实现

4 参考资料

5 人员分工

# 1 开发环境

开发环境：Windows7

编辑器：Masm32 & Visual Studio 2012

## 2 游戏规则

### 2.1 游戏背景

潜艇大战这个游戏不知曾经出现在多少人的童年里。它总是和 Win98，Win2000，小时候的机房联系在一起。不过随着时间的推移，这款游戏似乎已经销声匿迹。经过讨论，我们小组本次决定用汇编实现它，找回童年的记忆。

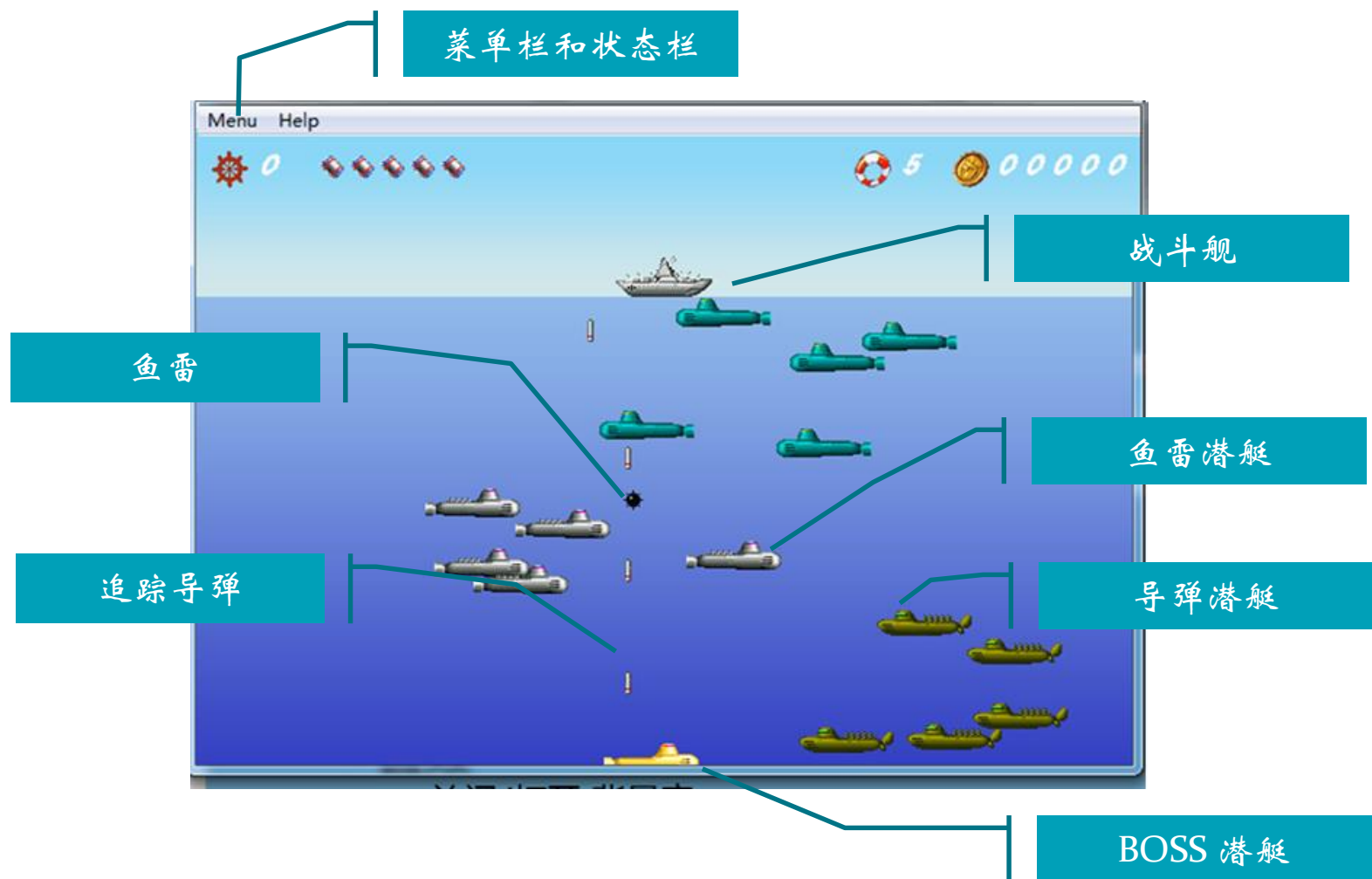
### 2.2 操作方式

它是一款战争类游戏。用户通过键盘左右键控制潜艇移动，通过 ZXC 按键控制战斗舰从左中右发射炮弹。敌方的潜艇会发射鱼雷和导弹来摧毁你的潜艇。

潜艇	能力
 战斗舰	 携带 5 枚炮弹
 潜艇 1	无攻击
 潜艇 2	 携带 1 个鱼雷
 潜艇 3	 携带一枚追踪导弹
 潜艇 boss	 携带多枚追踪导弹

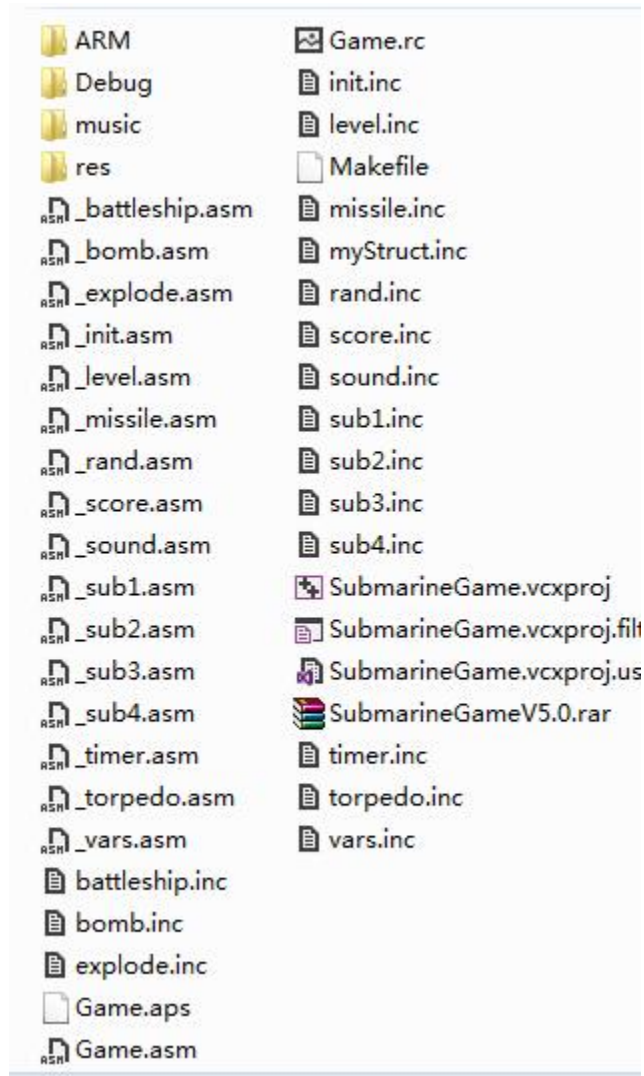
### 2.3 关卡设置

为了让游戏的娱乐性更强，我们精心设计了 10 个关卡，每个关卡中会出现不同类型不同数量的潜艇，到达一定分数后会进入下一关。关卡难度层层递进，让用户在游戏的过程中了解各种潜艇的能力，同时提高控制技巧。(level.asm)



## 3 设计与实现

### 3.1 代码结构



文件夹 res: 包含所有的图片资源文件

文件夹 music: 包含所有音乐资源

.asm 和.inc 文件：

本次开发过程中，我们选择按照对象分文件，如和 1 号潜艇相关的函数和变量（绘图和逻辑）都在 sub1.inc 和\_sub1.asm 中

## 3.2 界面绘制

### 3.2.1 图层蒙版

本游戏用到的图片资源均为 bmp 类型，所有图片在 Winmain 过程中载入，保存在位图句柄中。图片包括背景图，各种类型的潜艇图，数字图，爆炸效果图，炸弹图等。

首先载入蒙版图进行“或”操作（白色部分为 0，黑色部分为 1），保留蒙版图的黑色部分；之后载入前景图片，两个图片进行“与”操作，保留的黑色部分替换为彩色图片，以此实现图片的叠加。

### 3.2.2 双缓冲减少屏幕闪烁

使用一般的方法进行绘图，窗体在重绘时会由于过频的刷新而引起闪烁现象。为了防止绘制界面时屏幕出现闪烁，故采用双缓冲技术进行绘图。

双缓冲技术的原理是每次窗体在响应 WM\_PAINT 消息进行重绘时，先在内存中创建一张和窗体一样大的位图，然后在内存中的位图上进行绘制。所有的绘图工作完成后，一次性把内存中的这张位图复制到窗体上，这样就消除了闪烁。调用 CreateCompatibleDC 创建两个内存 DC，调用 CreateCompatibleBitmap 创建与 hDC 环境相关的设备兼容的位图，并将其放入 hMemDC 中。而 hImgDC 的作用是在绘图过程中将要画的位图选入，再在 hMemDC 上绘图。绘图完成后，调用 BitBlt，将 hMemDC 中的图一次性绘制到 hDC 上。

最后，需要注意的是，在绘制完成后要调用 DeleteDC 和 DeleteObject 把之前创建的两个内存 DC 和一个位图删掉，否则会造成内存泄露。

### 3.2.3 动画实现方法

动画实现主要是利用了时钟时间处理函数 Timer(timer.asm)。计时器在 WinMain 中进行定义，消息处理函数接收到 WM\_PAINT 时就会执行不同的动画。同时通过 SetTimer 和 KillTimer 设置 Timer 的开关和响应频率。本程序采用多个 Timer，分别控制屏幕刷新、生成潜艇和切换图片。

### 3.2.3 鱼雷导弹追踪效果

鱼雷和导弹均在母舰下方发射，这是通过将潜艇和母舰的 x 坐标进行对比判断的。其中鱼雷浮到水面后经过两次 Timer 计时自动消失。而导弹会在指定水平面下追踪母舰，通过对导弹和母舰的 x 坐标判断，确定导弹的移动方向并切换对应方向的图片。导弹斜着移动的速度与直行的移动速度大致相同，符合现实中的物理现象。

### 3.3 键盘控制

通过从缓冲区读按键码的方式可以对不同的按键响应进行判断。我们对不同按键进行了响应，包括移动和发炮弹。

原先我们只采用 keydown 模式，不过后来发现这样会有卡顿。调整后改成：在 keydown 时赋予潜艇不同方向移动速度，在 keyup 的时候将潜艇的移动速度置为零，这样可以保证潜艇左右移动非常流畅，避免卡顿。

### 3.4 碰撞检测

碰撞检测主要涉及 explode.asm, bomb.asm, missile.asm。其中 BombShip 和 BombSub 功能是判读炸弹是否炸到潜艇、战斗舰。打击的判定主要是把船和炮弹抽象成矩形，判断两个矩形的距离。为了让打击更加精确，我们特意调整了一些细微的参数。

除此以外，我们还实现了连续爆炸的效果：一个潜艇爆炸会波及到其他潜艇。这个使得游戏更加接近真实情况。文件中的 \_BoomSub 和 \_BoomShip 功能是判断爆炸效果是否炸到其他潜艇和战斗舰本身。

### 3.5 音乐处理

游戏配有循环播放的背景音以及音效（爆炸，导弹发射），所有音乐均为 wav 格式。音效的播放通过 playsound 函数实现，非常的流畅。背景音的播放则利用 MCI 的 API，从游戏开始一直循环播放。其中可以通过菜单键控制关闭（MCI\_PAUSE），打开（MCI\_PLAY）背景音和音效。背景音停止时利用（MCI\_STATUS）记录暂停位置，重新打开时会从上次停止的地方继续播放。

### 3.6 菜单项和快捷键

菜单栏的定义在 Game.rc 中，其中可以定义好菜单的结构，每一项的名称和对应的码。然后在 WinMain 中载入菜单。在 WM\_COMMAND 中处理对于菜单项的点击事件。同时调用 EnableMenuItem, ModifyMenu 来让菜单项失效或者更改。同时在 Game.rc 中可以进行快捷键的设置。

### 3.7 随机数生成

随机数\_iRand 用系统时间作为随机种子，然后利用公式 $(\text{randseed} * 23 + 7) \bmod (\text{second} - \text{first} + 1) + \text{first}$ 来产生 first 到 second 中的随机整数，其中 23 和 7 可以改为其他不同的质数。由于系统时间每 18 毫秒更新一次，故设置潜艇出现时每 100 毫秒增加一艘潜艇。

## 4 参考资料

Intel 汇编语言程序设计（第五版）

<http://www.feiesoft.com/win32asm/>

## 5 人员分工

王华清	双缓冲，碰撞，随机数，潜艇鱼雷导弹爆炸
孙浩	美工，音乐素材，展示，测试
王雨婷	母舰，菜单快捷，音乐，分数生命关卡