

# Battle City: 游戏的算法设计与分析

徐贤达 2016060601018

## 摘要

在此课程设计中，本人基于 Java 开发平台，成功地开发了一款具有图像交互式界面的坦克大战游戏 Battle City。在此游戏中，玩家可以选择游戏难度，操控己方英雄坦克，与电脑控制的多辆敌方坦克进行战斗。此外，本游戏基于文件系统，具有存储恢复功能，提升了游戏体验。本游戏的数据结构采用了 Java 中的矢量队列，算法设计主要包括面向对象编程和多线程编程，IO 设计主要包括界面编程，绘图技术和文件操作。

## 关键词

Java 语言、坦克大战游戏设计、面向对象编程、多线程编程、界面编程、绘图技术、文件操作

# 目录

第一章 需求分析.....	3
1.1 产品介绍 .....	3
1.2 产品遵循的标准.....	3
1.3 产品的功能性需求.....	3
第二章 相关技术.....	5
2.1 JAVA 语言介绍 .....	5
2.2 矢量队列 VECTOR 介绍 .....	5
2.3 面向对象编程介绍.....	6
2.4 多线程编程介绍.....	6
2.5 界面编程介绍.....	7
2.6 绘图技术介绍 .....	8
2.7 文件操作介绍 .....	8
第三章 详细设计.....	10
3.1 面向对象编程设计.....	10
3.2 多线程编程设计 .....	10
3.3 界面编程设计 .....	10
3.4 绘图设计 .....	11
3.5 文件设计 .....	12
第四章 游戏测试.....	13
4.1 测试环境 .....	13
4.2 测试结果 .....	13
第五章 总结与展望 .....	13
5.1 总结.....	13
5.2 展望.....	13

# 第一章 需求分析

## 1.1 产品介绍

本人设计的游戏 Battle City 基于经典的坦克大战游戏。它具有图形交互式界面，玩家可以在菜单栏中选择“开始新游戏”、“保存游戏并退出”、“继续游戏”等等。在“开始新游戏”菜单中，具有“简单”、“中等”和“困难”三种游戏难度。游戏过程中，玩家可以操控己方的英雄坦克与电脑操控的敌方坦克进行战斗。在不同的困难模式下，敌方坦克的数量，敌方坦克的移动速度，玩家坦克一次发射子弹的数量，玩家坦克的移动速度都有所不同。玩家坦克有 3 个生命值，被敌方坦克击中后可以复活，直到 3 个生命值耗光，游戏结束，玩家失败。如果玩家将对面的敌方坦克全部击毁，玩家胜利。综上，Battle City 是一款游戏体验性好，操作简单易上手的游戏。

## 1.2 产品遵循的标准

本人设计的游戏 Battle City 基于经典的坦克大战游戏，具有以下的特点：

1. 操作界面美观大方
2. 功能齐全，玩家和电脑分别能够操纵坦克进行射击且不越界
3. 体验性强，游戏具有三种不同难度，满足熟悉程度不同的玩家
4. 持续性强，游戏具有文件读写操作，可以保存之后继续

## 1.3 产品的功能性需求

功能类别	功能名称、标识符	描述
菜单栏	开始游戏	选择不同难度新游戏
	退出游戏	退出游戏
	保存游戏	保存当前游戏

	读取游戏	返回最近一次退出的界面
玩家坦克	坦克出生	在指定的位置出生
	移动坦克	通过键盘的上下左右移动坦克
	发射子弹	通过键盘的空格键发射子弹
	被击中	被击中时产生爆炸效果，生命值减一
	复活坦克	被击中后，如果还有生命值，复活坦克
	输掉游戏	生命值为 0 时输掉游戏
	赢得游戏	击败所有敌方坦克赢得游戏
电脑坦克	坦克出生	在随机的位置出生
	移动坦克	电脑自动在规则内移动坦克
	发射子弹	电脑自动发射子弹
	被击中	被击中时产生爆炸效果，消失

## 第二章 相关技术

### 2.1 Java 语言介绍

本人在本次课程设计中采用了 Java 语言。Sun 公司对 Java 编程语言的解释是：Java 编程语言是个简单、面向对象、分布式、解释性、健壮、安全与系统无关、可移植、高性能、多线程和动态的语言。

Java 编程语言的风格与 C、C++ 语言十分类似。不同的是：Java 是一个纯的面向对象的程序设计语言，Java 舍弃了 C++ 语言中容易引起错误的指针(用引用来取代)、多重继承(用接口来取代)、运算符重载(operator overloading)等特性，Java 继承了 C++ 语言面向对象技术的核心，增加了垃圾回收器功能用于回收不再被引用的对象所占据的内存空间，使得程序员不用再为内存管理而担忧。

### 2.2 矢量队列 Vector 介绍

本人在本次课程设计中主要采用的数据结构是 Vector，即矢量队列。

Vector 继承了 AbstractList，实现了 List 和 RandomAccess 等接口，如下图所示。和 ArrayList 相似但是有所不同的是，Vector 中的操作是线程安全的。在本次课程设计中，考虑到需要多线程编程，所以本人采用了 Vector 作为主要使用的数据结构而不是通常所用的 ArrayList。

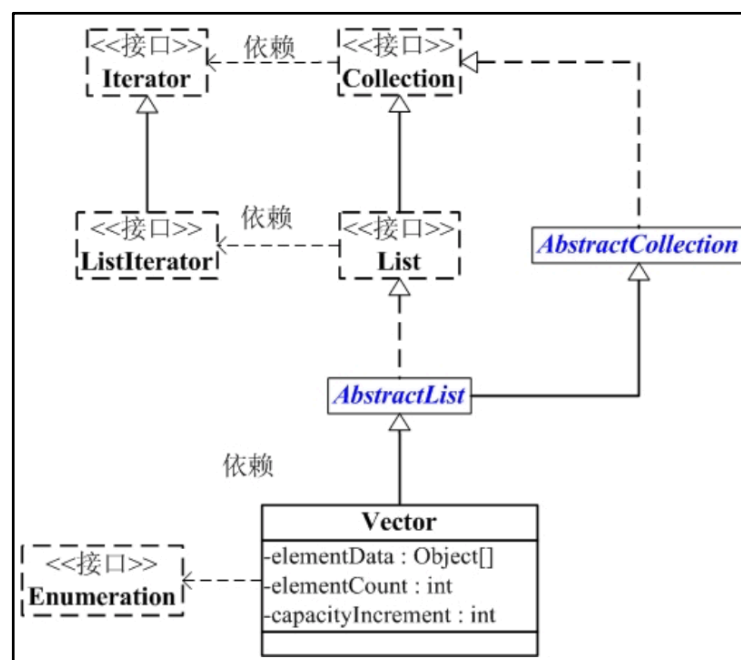


图 2.1 Vector 继承关系图

在此游戏的编写中，Vector 主要用来存放敌方坦克的所有对象，所有子弹的对象以及记录节点的对象。

## 2.3 面向对象编程介绍

面向对象编程是当今主流的程序设计思想，已经取代了过程化程序开发技术，Java 是完全面向对象编程语言。

面向对象具有以下特点：

1. 面向对象是一种常见的思想，比较符合人们的思考习惯
2. 面向对象可以将复杂的业务逻辑简单化，增强代码复用性
3. 面向对象具有抽象、封装、继承、多态等特性。

面向对象的基本思想是，从现实世界中客观存在的事物出发来构造软件系统，并在系统的构造中尽可能运用人类的自然思维方式。面向对象更加强调运用人类在日常生活的逻辑思维中经常采用的思想方法与原则，如抽象、分类、继承、聚合、多态等。

在此游戏的编写中，坦克、子弹、爆炸、画板等等，都是我们的类。

## 2.4 多线程编程介绍

Java 为多线程编程提供了内置的支持。一条线程指的是进程中一个单一顺序的控制流，一个进程中可以并发多个线程，每个线程并行执行不同的任务。目前绝大部分应用程序都会涉及到多并发的问题。只要应用程序涉及到并发，就离不开多线程编程。

Java 提供了三种创建线程的方法：

1. 通过实现 Runnable 接口
2. 通过继承 Thread 类本身
3. 通过 Callable 和 Future 创建线程

在本游戏编写中，我主要采用前两种方式进行线程创建。

它们都需要重写 run 方法，该方法是新进程的入口点，并且通过调用 start

方法创建并启动线程。

在此游戏的编写中，游戏程序是一个进程，下面有画板、坦克、子弹等许多小的线程。

下面我们讨论单线程中对于中断的响应与处理。

Java 对中断事件的处理采取的是“委派事件模型”。所谓“委派事件模型”是指当事件发生时，产生事件的对象(即事件源)，会把此“信息”传递给“事件的监听者”处理的一种方式，而这里所说的“信息”实际上就是 java.awt.event 事件类库里某个类所创建的对象，我们暂时把它称为“事件的对象”。



图 2.2 Java 委派事件模型

事件时承载事件源状态改变时的信息对象，常见的事件有：

事件类	说明
ActionEvent	通常在按下按钮，或双击一个列表项或选中某个菜单时发生
AdjustmentEvnet	当操作一个滚动条时发生
ComponentEvent	当一个组件隐藏、移动、改变大小时发送
ContainerEvent	当一个组件从容器中加入或者删除时发生
FocusEvent	当一个组件获得或是失去焦点时发生
ItemEvent	当一个复选框或是列表项被选中时，当一个选择框或选择菜单被选中
KeyEvent	当从键盘的按键被按下，松开时发生
MouseEvent	当鼠标被拖动、移动、点击、按下..
TextEvent	当文本区和文本域的文本发生改变时发生
WindowEvent	当一个窗口激活、关闭、失效、恢复、最小化...

在此游戏的编写中，我们会遇到 ActionEvent 事件，即按下菜单按钮的事件，KeyEvent，即按下键盘的事件，MouseEvent，即鼠标操作的事件以及 WindowEvent，即窗口操作的事件。而我们处理的方法，则是实现它们的接口。

## 2.5 界面编程介绍

Swing 组件是常用的菜单界面绘制组件，它是一个为 Java 设计的 GUI 工具

包，包括了图形用户界面器件如：文本框、按钮、分隔窗格和表。它的常用组间包括 JFrame, JPanel, JButton, JLabel, JTextField, JPasswordField, JCheckBox, JRadioButton, JComboBox, JScrollPane, JList 等等。

## 2.6 绘图技术介绍

在此游戏的编写中，绘制坦克等所用的类是 Graphics 类。它可以被理解为是画笔，为我们提供了各种绘制图形的方法：

- 1、画直线 drawLine(int x1,int y1,int x2,int y2);
- 2、画矩形边框 drawRect(int x,int y,int width,int height);
- 3、画椭圆边框 drawOval(int x,int y,int width,int height);
- 4、填充矩形 fillRect(int x,int y,int width,int height);
- 5、填充椭圆 fillOval(int x,int y,int width,int height);
- 6、画图片 drawImage(Image img,int x,int y,..);
- 7、画字符串 drawString(String str,int x,int y);
- 8、设置画笔的字体 setFont(Font font);
- 9、设置画笔的颜色 setColor(Color c);

Component 类提供了两个和绘图相关的重要方法：

- 1、paint ( Graphics g ) 绘制组件的外观
- 2、repaint ( ) 刷新组件的外观

利用此，我们便可以绘制出游戏要求的图形。

## 2.7 文件操作介绍

文件在 Java 中是以流的形式来操作的。



图 2.3 文件以流的方式操作



一个流可以理解为一个数据的序列。输入流表示从一个源读取数据，输出流表示向一个目标写数据。Java.io 包几乎包含了所有操作输入和输出需要的类。下面是一个描述输入流和输出流的类层次图：

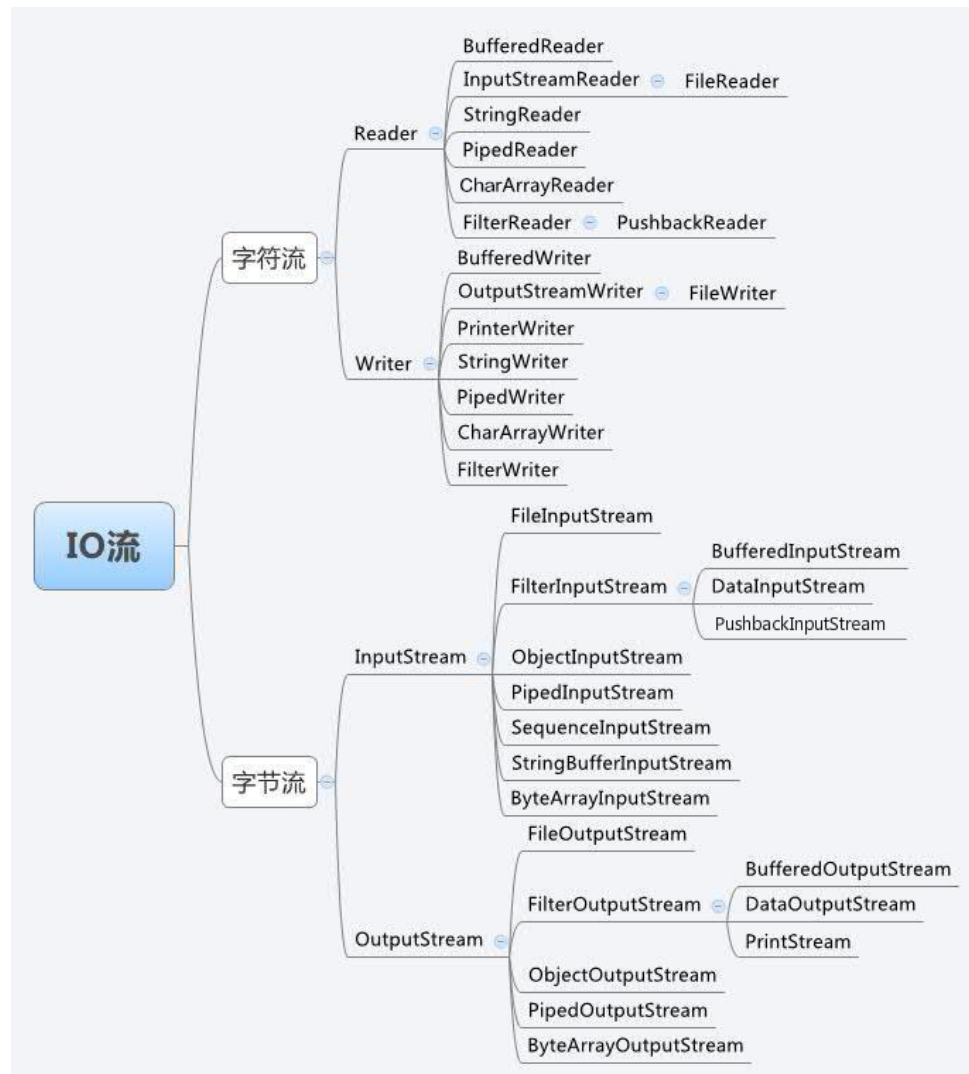


图 2.4 Java IO 流类层次图

## 第三章 详细设计

### 3.1 面向对象编程设计

本游戏所有类及其说明如下表所示:

类	说明
Tank	所有坦克的类
EnemyTank	敌方坦克的类, 继承 Tank
Hero	玩家坦克的类, 继承 Tank
Shot	子弹的类
Bomb	爆炸效果的类
Recorder	存档的类
Node	记录点的类
MyStartPanel	启动画板的类
MyPanel	游戏画板的类
MyTankGame	主程序的类

### 3.2 多线程编程设计

本游戏所有线程及其说明如下表所示:

线程中对象所属的类	说明
MyStartPanel	启动界面的线程
MyPanel	游戏界面的线程
Hero	玩家坦克的线程
EnemyTank	每个敌方坦克拥有一个线程
Shot	每发子弹拥有一个线程
Bomb	每次爆炸拥有一个线程

### 3.3 界面编程设计

本游戏菜单界面如下图所示:

Start: 开始新游戏 (有简单、中等、困难三种模式)

Exit: 退出游戏

Save and quit: 保存并退出游戏

Return: 返回上一次保存的游戏

界面设计具有简单美观的特点。

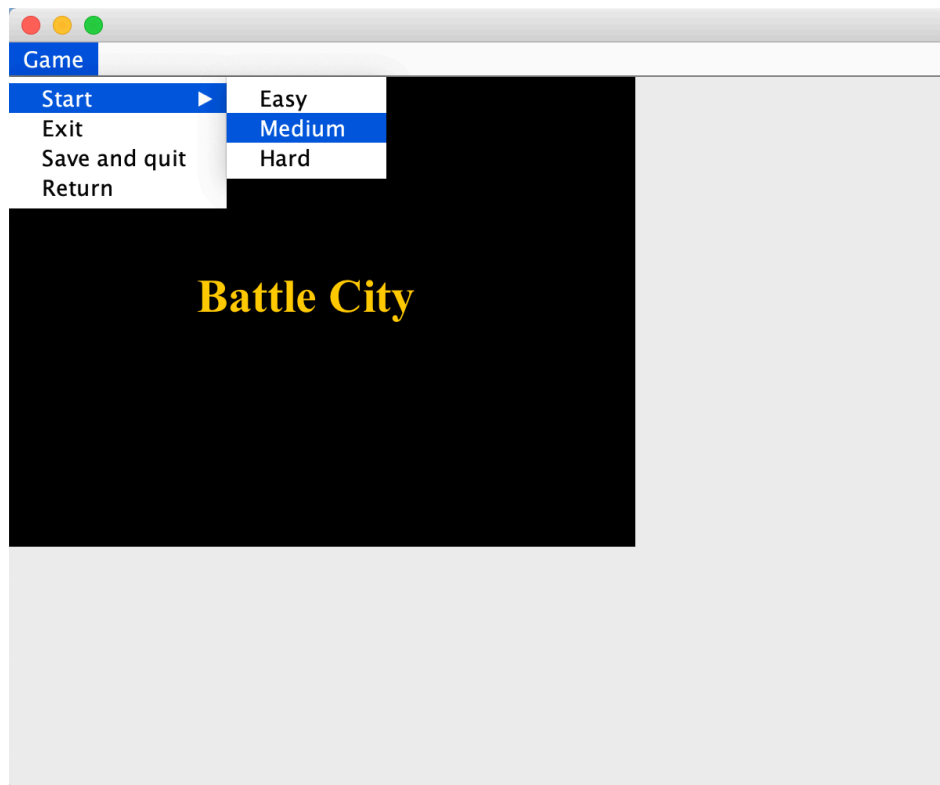


图 3.1 游戏菜单栏设计

### 3.4 绘图设计

本游戏坦克的绘图如下图所示:

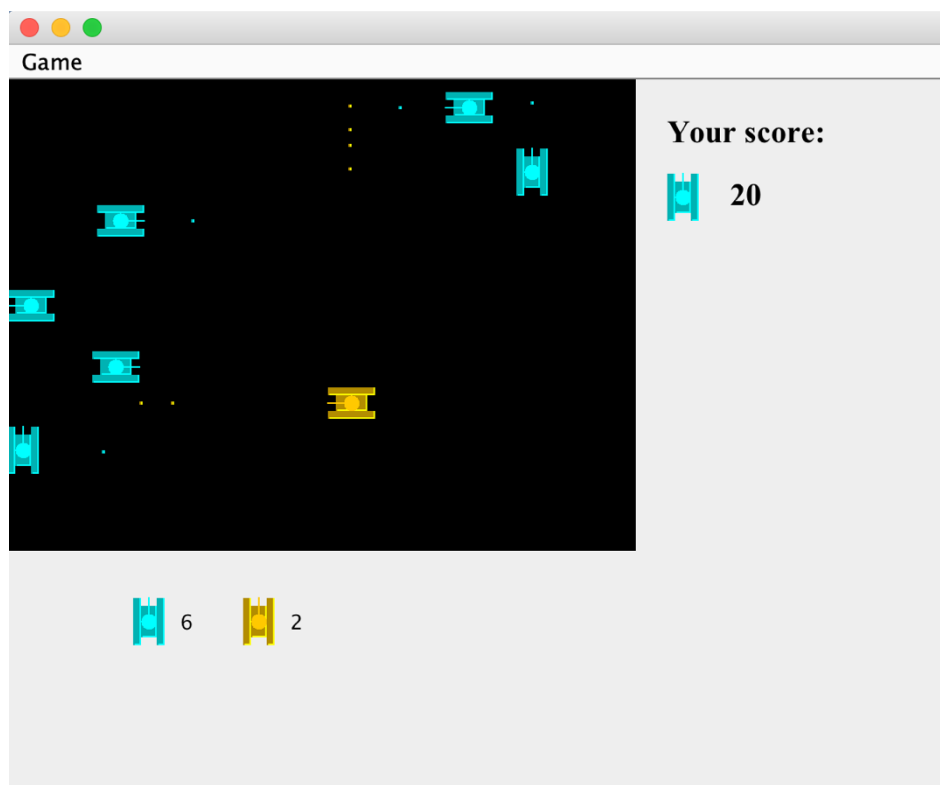


图 3.2 坦克绘图设计

其中，敌方坦克和玩家坦克的外形接近，只是颜色不同。

值得注意的是，在游戏过程中，我们需要有规范性的要求，下面我们逐一来讨论与解决。

### 1. 坦克的活动

坦克的活动，我们通过每一次画图（间隔 50 ms）对于坐标的修改完成，即加上或减去坦克的速度值。

坦克的活动，需要在指定的范围中进行，所以我们需要不断地检测坦克是否碰到墙壁或者是否碰到其它坦克。如果碰到了，我们不能让它们继续运动。

### 2. 电脑坦克的活动

电脑控制着坦克的活动，它的活动基本方式在上面讲述，但是，它和玩家操纵的坦克不同，它没有控制它转向的外部中断。在这里，我设置了每隔一定时间进行随机转向和发射子弹的函数，让电脑控制的坦克正常运行。

### 3. 爆炸效果的产生

一旦坦克被击中后，便会产生爆炸效果。爆炸效果的产生，实际上是几张图片的定时切换。我在网上找了几张爆炸不同阶段的图片，一旦触发爆炸，每隔一定时间切换图片，使得看上去是连续的爆炸。

## 3.5 文件设计

游戏使用了 Recorder 这个类，辅助以 Node 记录点，进行存档与读档操作。

一旦用户选择保存退出，Recorder 便记录当前的所有信息，包括存活坦克的信息，剩余坦克的数量，玩家的积分等等。存活坦克的信息记录，通过 Node 类完成，包括每个坦克的坐标，方向与速度。

记录完成后，Recorder 便将所有信息记录到特定路径下的文本文件中存放。之后读档的时候同理进行还原。

## 第四章 游戏测试

### 4.1 测试环境

操作系统: macOS Mojave 10.14.4

编译器: IntelliJ IDEA

处理器: Intel Core i7

内存: 16 GB

### 4.2 测试结果

请见视频

## 第五章 总结与展望

### 5.1 总结

本课程设计中, 本人编写了一款基于经典坦克大战的游戏 Battle City。在此游戏中, 玩家可以选择游戏难度, 操控己方英雄坦克, 与电脑控制的多辆敌方坦克进行战斗。此外, 本游戏基于文件系统, 具有存储恢复功能, 提升了游戏体验。

在本学期课程的学习中, 本人收获颇丰, 总结为“温故而知新”。本人重温了经典的数据结构, 包括队列、栈、二叉树、红黑树等等, 重温了诸如深度优先搜索、广度优先搜索、二分搜索等经典的算法, 通过课堂与作业加深了对它们的理解与应用。此外, 本人了解了新的东西, 包括作业中的字典树, Raptor 软件的使用等等。在大作业的编程中, 本人一方面熟练了 Java 的编程, 另一方面尝试了新的东西, 在这里是多线程编程。在此大作业的编写之后, 本人对于多线程编程有了更好地了解。

### 5.2 展望

随着人们的需求不断地在变更, 本人认为此游戏在以下方面有进一步发展与完善的空间:

1. 适当增加障碍物
2. 适当增加关卡
3. 实现多玩家的操作

我相信，通过进一步的发展与完善，本人目前编写的 Battle City 游戏能够满足更高的用户体验要求。

最后，感谢戴波老师一学期的课堂教学和助教们一学期的课后工作，谢谢。