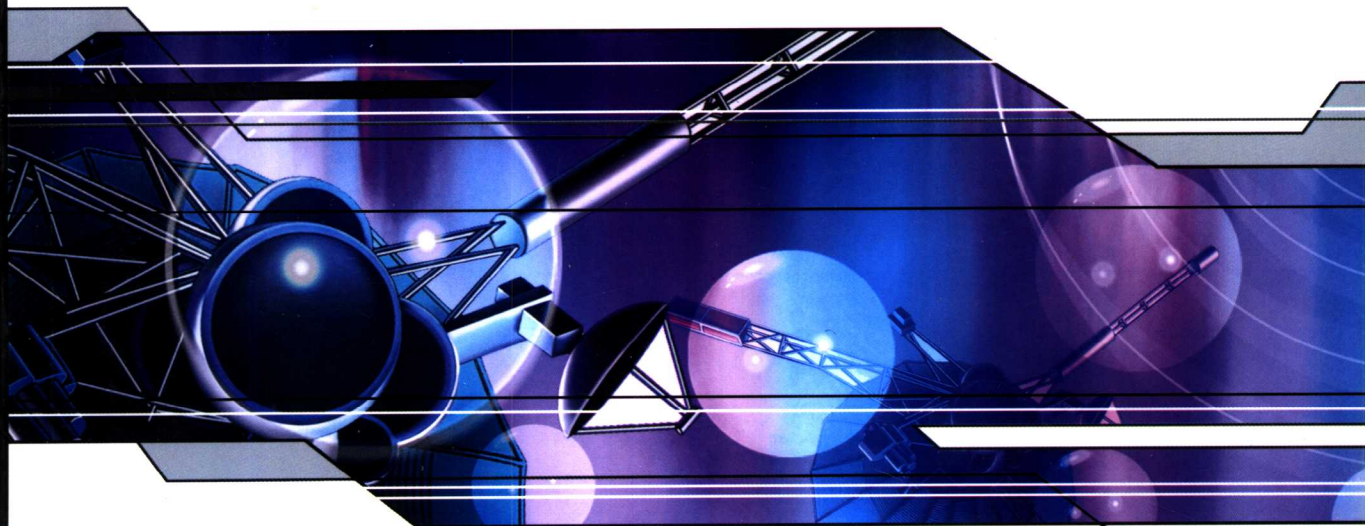




游戏开发与编程系列

游戏编程中的数理应用

Beginning
Math and Physics
for Game Programmers



北京希望电子出版社 总策划

[美] Wendy Stahler 著

冯宝坤 曹英 译

爱亿尔[北京]国际游戏开发院 审校

New
Riders

红旗出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhep.com.cn

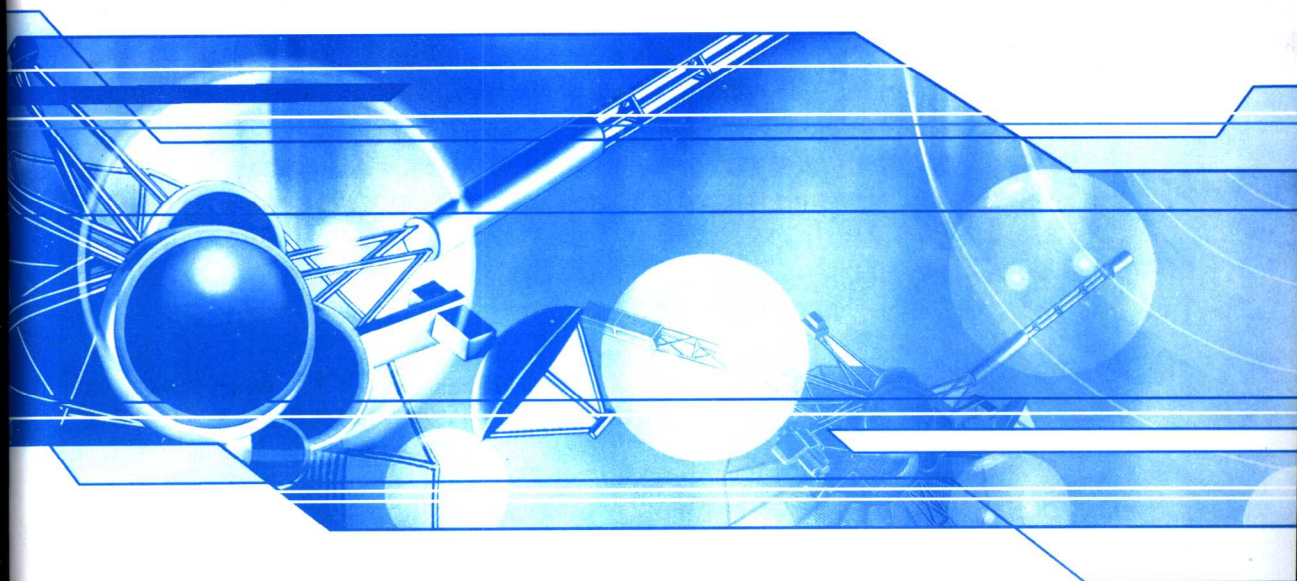
NRG



游戏开发与编程系列

游戏编程中的数理应用

Beginning
Math and Physics
for Game Programmers



北京希望电子出版社 总策划

[美] Wendy Stahler 著

冯宝坤 曹英 译

爱亿尔[北京]国际游戏开发院 审校

New
Riders

红旗出版社



北京希望电子出版社
Beijing Hope Electronic Press
www.bhep.com.cn



图书在版编目 (CIP) 数据

游戏编程中的数理应用 / (美) 斯塔勒 (Stahler, W.)

著; 冯宝坤等译. —北京: 红旗出版社, 2005.3

书名原文: Beginning Math and Physics for Game Programmers

ISBN 7-5051-10739-X

I. 游... II. ①斯... ②冯... III. 游戏—软件设计...

IV. TP311.5

中国版本图书馆 CIP 数据核字 (2004) 第 102827 号

内 容 简 介

本书从高等数学和物理知识着手, 讲述 2D 图形在迪卡尔坐标中的构形与计算, 同时涉及一些 3D 图形图像及其处理方面的知识。对于物理知识部分, 主要讲述力与物体运动的关系, 速度、加速度、牛顿三定律、动量、冲量和能量守恒定律, 以及这些知识在三维空间游戏建模中的应用。

本书所讲知识主要是为游戏开发做基础, 并配以丰富的 C++ 实例代码, 对所讲知识做实践, 使本书的价值提高到一个新的位置。虽然书中没有具体讲解 3D 游戏设计的范例, 但它却是在解决 3D 游戏设计中关于物理和数学问题的重要工具。

本书适合于所有游戏设计与开发者, 对高校计算机专业师生也是一本难得的好书。

光盘内容为书中部分 C++ 实例源代码和演示程序。

版 权 声 明

本书英文版书名为 Beginning Math and Physics for Game Programmers, 作者是 Wendy Stahler, 由 New Riders 出版, 版权归 New Riders 所有。未经出版者书面许可, 本书的任何部分不得以任何形式或手段复制、传播。

系 列 名	游戏开发与编程系列
书 名	游戏编程中的数理应用
编 著	[美] Wendy Stahler
总 策 划	北京希望电子出版社
翻 译	冯宝坤 曹英
责 任 编 辑	但明天 雷铎
出 版 发 行	红旗出版社 北京希望电子出版社
地 址	红旗出版社 北京市沙滩北街 2 号 (100727) 电话: (010) 64037138 北京希望电子出版社 北京海淀区上地三街 9 号金隅嘉华大厦 C 座 610
经 销	各地新华书店、软件连锁店
排 版	希望图书输出中心
印 刷	保定列电印刷厂
版 次 / 印 次	2005 年 3 月第 1 版 2005 年 3 月第 1 次印刷
开 本 / 印 张	787 毫米×1092 毫米 1/16 22.25 印张 510 千字
印 数	0001~5000 册
书 号	ISBN 7-5051-1039-X
定 价	45.00 元 (配 1 张光盘)

从游戏中学会生存（总序）

——献给游戏创造者

游戏为什么受到重视

今天，越来越多的人不再把游戏看作小儿科，也不仅仅视其为一门艺术了。这大概要归功于全球每天数亿美元的市场业绩，抑或在金融危机中使一个国家经济转危为安的力挽狂澜作用。作为一种生存途径，日本、韩国深得游戏产业振兴经济的非凡要领。无数名不见经传的小人物靠游戏制作成为百万富翁，游戏经典之作的原创者则受到了超级明星般的待遇而被全世界的游戏玩家追捧着。

当游戏被视为艺术时，人们常常拿他与传统艺术形式作比较。认为游戏作为一门娱乐艺术形式，不同于诗歌、小说、戏剧、电影、电视等等传统艺术形式的根本点是“交互”，是“参与”。不错，游戏作为一个产业勃然兴起，确实离不开计算机和互联网。但就其对人类的根本作用而言，要比交互、参与这类技术层面的问题远为原始得多。

游戏比艺术起源更早

诗歌、小说、戏剧、电影、电视等等艺术的形成，依赖于语言文字的产生，因此其历史见于壁画与甲骨等人类考古学证据的发现，大约可以上溯到距今 5000 年前。

游戏却不仅仅属于人类。小猫玩线球，人们看到的只是小猫的兴趣盎然和乐此不疲，却不见在这种练习中小猫捕捉能力的提高：通过追逐线球，小猫强壮了自己的利爪，敏捷了全身的器官和肌腱。这种对后代的训练形式普遍存在于动物界，其历史至少可以追溯到对哺乳动物的考古学发现，并非最早的证据距今也已经有 6000 万年了。

从生存描述到生存模拟

作为对生存过程的一种描述，诗歌、小说、戏剧、电影、电视等等艺术，是把受体放在媒体之外，通过媒体的艺术感染力，使受体产生共鸣，令其产生沉浸感。

作为对生存过程的一种模拟，游戏则是创造一种虚拟环境，把受体放在媒体之中，使受体自己判断、决策和动作，通过反复操作媒体获得某种能力。数千万年以来，游戏使用着更直接的沉浸手段，让后代在沉浸中得到更实际的生存锻炼。

假如游戏不仅以娱乐为目的

由此可见，作为功能的界定，与游戏更接近的社会部类是教育，而不仅仅或不主要是娱乐。无怪乎国际游戏开发者协会（IGDA）活动负责人杰森·德拉·罗卡（Jason Della Rocca）在 2003 CHINAJOY 大会上的主题发言是《“严肃”游戏：游戏对社会经济的潜在影响》。他把“严肃”游戏（Serious Game）定义为“不以娱乐为主要目的游戏”，例举了训练市长的《模拟城市》、训练董事长的《虚拟领导》、训练员工的《直言者》、训练海军陆战队员的《DOOM》等等经典游戏作品。在这一方面，美国确实走在了全世界的前面。早在 1994 年，美国海军陆战队就成立了世界上第一个 Game 军事训练机构；1995 年，美国空军和陆军紧随其后，把游戏作为军队训练的有效辅助手段。总结这些经验，罗卡先生列出了游戏

的适用领域依次为：

- 教育
- 国家政策
- 卫生保健
- 企业培训
- 其他

游戏可以创造未来

随着计算机的更加移动化、个性化，虚拟社会与现实社会的界限愈加模糊了。如今网游交易市场上的商业链，与现实市场上的买卖规则何其一致！试想，当我们的手机、腰带、手表甚至纽扣全都可以无线上网之后，谁还能够分清现实世界与虚拟世界的截然界限呢？

在虚拟世界与现实世界的直接联系中，虚拟世界的创造者可以把游戏里发展起来的生存新规则，通过游戏渠道传播到更大的范围，影响更多的人。换言之，游戏的创造者可以创造更加理想的未来。这就是：

- 改变工作生活方式
- 影响现有科学技术
- 增强综合国力
- 造福于社会

创造者在游戏中永生

回首 12 年前，中国最早的一批拓荒者进入了游戏开发领域。那时他们相信凭借坚定的信念和努力付出就可以确保成功。然而，游戏也是一种商业，他像其他行业一样受到财务、市场和管理等等条件的制约。在软件盗版和决策失误、管理失措的多重夹击下，许多人付出了惨重代价。然而这些付出换来的是生存经验的积累和生存能力的提高。在经过最初的游戏狂热到现在国际游戏市场的供不应求之后，更多的游戏开发从业者学会了在激烈竞争和快速变化的市场中生存。在游戏产业中，似乎并不遵守“胜者王侯败者寇”的法则，美酒的甘甜和血汗的腥涩都可以收入博物馆，正因为他们是历史的拓荒者。

今天，互联网给游戏带来了蓬勃发展的春天。大量资金裹挟着人才和技术进入游戏开发领域。然而，仿效、重复也跟着多起来了，类同题材和近似玩法的游戏产品充斥市场。巨大的风险潜藏随之，二八铁律决定了绝大多数仿效者的时空有限。因此，要想在游戏中获得永生，唯有创造，别无他途。

希望这套出自世界游戏创造高手的丛书能对梦想创造者有所帮助。

杨南征

爱亿尔（北京）国际游戏软件开发院院长
工商管理博士、作战模拟研究员

前言

6年前，当有人问我有关游戏设计课程中物理和数学的发展状况时，事实上，当时还完全没有任何关于该主题的参考书。

尽管在产业杂志上有许多文章以及相关的书籍，但都只是很零碎地谈到有关游戏发展的一般状况，几乎没有一本书集中讨论过在游戏应用程序中关于数学和物理的问题。我非常热烈地从多方面收集并了解有关这个主题的信息，之后我从传统的数学和物理课本中整理这些资料，并收集有关游戏设计书籍，摘录并撰写成文，最终成为本书内容的核心部分。

在对游戏设计和发展计划的材料进行仔细研究和精加工之后，本书的编写工作起锚了。开始，很多出版商与我联系。最终，关于该领域的几本书出现在书架上，但是其中的大部分的市场反响并不强烈，这个问题在我的大脑中不断回旋。

为了给热心的程序员准备一本普遍适用的书，所以我决定把本书的作用放到一个基础的位置，尽量避免艰难晦涩。

本书也许不是你从未读过的关于游戏设计中的数学和物理主题的第一本，我希望它能够为你提供一些帮助，并成为你学习程序设计的启蒙读物。

本书的前一部分讨论了一些关于数学方面的基础知识，后一部分则讲述了一些有关物理方面的基础知识。

第1章从点和线开始，同时讲述了在直角平面和三维空间程序设计中，有关点、直线和平面的知识。

第2章主要介绍了游戏设计中，有关几何学方面的知识，并对勾股定理、两点之间的距离公式以及中点公式进行了回顾。给出了一些常用的公式，例如圆、抛物线和球，并对它们进行了详细讨论。

第3章主要介绍了游戏设计中有关三角函数方面的知识，分别对6种三角函数做了讨论。在附录里，还给出了测量角和圆弧的C++函数。

第4章对向量和极坐标做了介绍。首先介绍了不同的向量表示方法，然后介绍了向量的所有数学运算，包括向量加法、点乘、叉乘、相交以及标准向量表示法。

第5章讨论数学中的矩阵和复数，它们的运算包括矩阵相加、相乘、复数相乘以及转置矩阵等。

第6章描述了所有的复变换。复数在游戏中能控制对象的动作，例如物体平移、缩

放和旋转。

第 7 章介绍了游戏设计中需要处理的有关物理量。在编码中，给出了所有游戏单元的量度转换方法。

第 8 章主要讲述了物理学方面的知识，包括速度、位移以及它们对时间的关系，位移的定义和度量。

第 9 章主要介绍了有关速度和加速度方面的引申概念，同时对微积分知识也做了简单叙述。

第 10 章对前面两章进行展开讨论，介绍了有关三维空间的知识。在本章里，主要对抛射运动做了讲解。

第 11 章主要讲述了作用力与物体运动的知识。对广为人知的牛顿（艾萨克·牛顿）三定律做了介绍。本章与前面章节相结合，主要讨论物体运动的规律。

第 12 章对两种不同的能量类型做了描述。详细介绍了能量在游戏中对物体运动的状态影响、变化以及对象建模。

第 13 章对娱乐性游戏中的主要内容——冲突进行了介绍。没有冲突也就没有联系，本章描述了物体的运动方式，其结果就是它们如何彼此攻击。

第 14 章主要讨论了物体的转动。从早期的直线运动转化成角运动和旋转运动的条件进行了重定义。

附录 A 是本书中涉及的数学和物理公式。

附录 B 是一些数学和物理术语。

每一章都包含了若干实例以及问题的解答（自我测试），以便检查你的回答是否正确。每章都包含了许多知识点，部分内容在随书光盘中有收集。几个章节针对重要概念做了举例、简单图解，并给出了演示程序。光盘中还提供了每个实例的 C++ 源代码，以便你能理解它们的作用。

作者简介

Wendy Stahler 是位于佛罗里达州文特公园的周游现实世界教育机构的游戏设计和编程发展的首席执行官。在那里的 6 年时间里，她致力于数学、物理编程的教学研究。同时，作为格林大学 IT 工程系的副教授，她刚决定负责公司的 IT 培训。Wendy 毕业于洛林大学，在那里她获得了数学学士学位，并集中精力研究计算机科学，同时她还获得了企业技术和交流的硕士学位。

技术顾问

这些顾问都曾对本书的编写做过大量工作。在本书的编写过程中，他们直接参与了有关技术内容的指导、文章的组织 and 流程安排。正是他们的指导才使本书能够成为可以教给读者最优技术知识的好书。

Roberto Colnaghi, Jr. 曾获得计算机科学学士学位和微软技术认证。他致力于复杂的网络系统和网络服务开发，有利用 Microsoft .NET Framework 技术进行网络游戏的编写。Roberto 在微软技术认证学校任过教，并有一个涵盖广泛的个人图书馆，书目内容从多媒体游戏到游戏起源，包罗万象。他参与过游戏学术交流并收集相关人士的 E-mail，还在学校和网络上为应用于 Windows 系统中的游戏编程提供素材。

Kevin Mack 是一位游戏设计师，近期致力于电子艺术公司的《荣誉勋章：太平洋攻击》。在加入 EA 之前，他为迪斯尼、凯迪拉克和科罗诺数字娱乐设计游戏。作为一名程序员，Kevin 曾经编写过实时冒险类 3D 游戏，并为 Razorfish LA 公司充当技术顾问。他的研究范围很广，同时在游戏发展上的建树，使他足以做一些纪实电影的制作工作，并且参与了纽约资本市场的数据库建立。他在纽约大学学习期间，就获得了美国电影机构的电影硕士学位。Kevin 现居住于加利福尼亚州的帕萨德纳。

编外评论

下面是编外人员对本书的一些评论。

“这本书写得很好，作者详尽地解释了数学和物理中的许多概念，并将如何运用这些术语 and 知识进行 3D 编程介绍给读者。”

——Bill Galbreath，数字媒体和周游现实世界教育机构的项目经理

“我们在书店里能够找到许多其他关于游戏编程的物理和数学方面的书籍，而那些书更偏重于介绍高级知识，这些知识对于一个初学者来说太复杂了。本书所介绍的知识正是其他书籍所忽略的，它们的作者自以为读者早已了解了这些知识。”

——David Astle，GameDev.net 的网管

“我用这本书教我的学生进行游戏设计和编程。本书的内容组织得很好，它清晰、透彻，而且具有目的性，非常适合那些仍缺乏物理和数学知识的具体运用的初学者。”

——Harvey Duff, 加拿大阿尔波塔加斯伯中学计算机技术系主任

“数学和物理的初学者会发现，这本书是一个很好的开始。这本书很容易理解，所以你能够有一个好的开端。习题和实例更能帮助你很好地学习。”

——Gianfranco Berardi, 德保罗大学计算机科学系学生

“这本书是对游戏编程中有关物理、数学知识的一个很好的总结。作者对三角函数和矩阵都进行了生动的图形展示，让读者更加乐于阅读。另外，里边还包含了很多理论的综合应用和有用的建议。”

——Richard Jones, 莫吉斯有限公司主管 <http://www.annexia.org>

读者对象

本书适合于所有游戏设计爱好者。书中大部分实例的源代码都放在了附录光盘上，如果你已经对 C++ 非常熟悉，那么你可以对这些关于游戏设计中的数学和物理问题的代码进行深入研究。不过，程序设计经验不是必须的，假如你仅仅将精力集中在数学和物理领域，你可能会对自己产生质疑，这时你也可以回顾一下这些代码，你可能会发现它们和你学习 C++ 语言的时候有什么不同。

本质上本书不是讲 3D 游戏设计，但它却是在解决 3D 游戏设计中由于数学和物理问题的得力工具。在本书的附录部分，作者还对有关 3D 方面的数学和物理概念、公式等分章进行了归纳，这对你解决 3D 仿真中的实际问题很有帮助。阅读本书，扎实的代数知识是必须的，同时能具有良好的几何与三角知识则更加有效。

从本书里你将获得什么？

如果你能做完每一章里的所有实例，并且能够完全理解其中的数学原理，那么对于创建和实现一般的 3D 模型并确保它们能在现有系统上运行已经很有把握。这样，你就可以摆脱点与线的束缚，投入并运用这些基本原理到 3D 游戏引擎设计中。从本书中，你将获从任何高级教材中所没有的更多知识。

记住，这仅仅是一个开始。运用这些基础知识到更深入领域的研究中。附录 B 中给出了一些数学和物理方面的概念，其中一些概念比较新。运用本书的知识，开始你长期的、疯狂的在 3D 世界的物理仿真和数学实现之旅吧。你能做到吗？

致 谢

爸爸、妈妈——谢谢你们鼓励！

Hap Aziz——谢谢你把我领入疯狂的游戏世界。

Gene Woolcock 和 Tony Judice——感谢你们在中学教给我的数学和计算机知识。

Douglas Child 和 Donald Griffin——感谢你们给我这些前导的建议。

Stephanie Wall 和 Lisa Thibault——谢谢你们对我的细心帮助。

Marc Mencher——谢谢你鼓励我写这本书。

Cody Kahrizi 和 Dustin Clingman——谢谢你们的专业编程知识和对本书的贡献。另外，感谢我的程序员们——Dustin Henry, Mike Wiggand 和 Mike Fawcett。

此外，特别感谢我所有的学生，谢谢你们的热情和鼓舞。

目 录

第1章 点和线.....1	自我测试.....29
1.1 点的定义.....1	2.2 抛物线.....29
示例 1-1 在2D场景中确定位置..2	示例 2-6 画抛物线草图.....30
示例 1-2 画出3D位置.....4	示例 2-7 画出另一支抛物线的 草图.....30
自我测试.....5	自我测试.....31
1.2 线的定义.....6	2.3 圆和球.....31
示例 1-3 画线.....6	示例 2-8 画出圆的草图.....32
示例 1-4 画水平直线.....7	示例 2-9 画出另一个圆的草图..33
自我测试.....8	示例 2-10 写出圆的方程.....34
1.3 直线的属性.....8	示例 2-11 球体的圆心和半径...34
示例 1-5 两点之间的斜率.....9	自我测试.....35
示例 1-6 直线的斜率.....9	2.4 碰撞检测的应用.....36
示例 1-7 计算直线的斜率.....11	示例 2-12 两圆是否重叠.....37
示例 1-8 写出直线方程.....11	示例 2-13 两球是否重叠.....38
示例 1-9 算出垂直直线方程.....12	自我测试.....40
示例 1-10 找出3D直线.....13	2.5 视觉体验：相撞检测.....40
自我测试.....13	2.6 自我测试答案.....41
1.4 相交线检测的应用.....13	两点间的距离.....41
示例 1-11 一个直线方程组.....15	抛物线.....41
示例 1-12 直线交点的检测.....15	圆和球.....42
示例 1-13 用联合法求交点.....17	相撞检测的应用.....42
示例 1-14 用代入法求交点.....17	第3章 三角函数.....43
自我测试.....19	3.1 角度与弧度.....43
1.5 自我测试答案.....20	示例 3-1 标准位置的正角.....44
点的定义.....20	示例 3-2 标准位置的负角.....44
线的定义.....21	示例 3-3 把角度转换成弧度.....45
直线的属性.....22	示例 3-4 把弧度转换成角度.....45
相交线检测的应用.....22	自我测试.....45
第2章 一些几何知识.....23	3.2 三角函数.....46
2.1 两点间的距离.....23	示例 3-5 说明正弦余弦和正切..47
勾股定理.....23	示例 3-6 利用余弦.....49
示例 2-1 屏幕上两点间的距离..25	示例 3-7 用反正切计算.....49
示例 2-2 判断直角三角形.....25	示例 3-8 增大正弦曲线的频率..51
示例 2-3 3D空间中的两点距离..26	示例 3-9 减小正弦曲线的频率..52
示例 2-4 屏幕上两点的中点.....27	示例 3-10 扩大正弦曲线的振幅..53
示例 2-5 3D空间中的中点坐标..28	

示例 3-11 缩小正弦曲线的振幅	53	标量的乘积	74
自我测试	55	示例 4-12 将一个向量单位化	75
3.3 三角函数的性质	55	自我测试	75
示例 3-12 验证正负角的性质	57	4.5 点 乘	76
示例 3-13 $\sin(90^\circ + \alpha)$	57	示例 4-13 检测物体是否在	
示例 3-14 $\sin(180^\circ - \alpha)$	57	视野中	76
示例 3-15 $\cos(180^\circ + \alpha)$	58	示例 4-14 两向量之间的角度	77
示例 3-16 $\cos(90^\circ - \alpha)$	58	自我测试	78
自我测试	58	4.6 叉 乘	78
3.4 利用 C++ 中的数学运算库	58	示例 4-15 叉乘	79
3.5 自我测试答案	60	示例 4-16 面垂直单位	82
角度和弧度	60	示例 4-17 两向量之间的角	83
三角函数	61	自我测试	84
三角函数的性质	61	4.7 视觉体验	84
第 4 章 向量的功能	62	4.8 自我测试答案	85
4.1 向量和标量	62	向量和标量	85
示例 4-1 正方向位移	63	极坐标和直角坐标	85
示例 4-2 反向位移	63	向量的加减法	86
示例 4-3 路程和位移	64	标量与向量的乘法	86
自我测试	65	点 乘	86
4.2 极坐标和直角坐标	65	叉 乘	86
示例 4-4 把极坐标转化为直角		第 5 章 矩阵运算	87
坐标	67	5.1 全等向量	87
示例 4-5 把直角坐标转化为		示例 5-1 说明一个矩阵	88
极坐标	67	示例 5-2 是否是全等矩阵	89
自我测试	69	示例 5-3 是否是全等矩阵	89
4.3 向量的加减	69	自我测试	90
示例 4-6 画出新向量	70	5.2 矩阵的加减	91
示例 4-7 用数字进行向量加法		示例 5-4 矩阵加法	91
运算	71	示例 5-5 矩阵减法	92
示例 4-8 用数字进行 3D 向量		自我测试	93
加法运算	72	5.3 矩阵与标量的乘法	94
示例 4-9 用数字进行 3D 向量		示例 5-6 标量*矩阵	94
减法运算	73	示例 5-7 矩阵方程	95
自我测试	73	自我测试	96
4.4 向量与标量的乘积	74	5.4 矩阵之间的乘法	96
示例 4-10 用极坐标求向量与		示例 5-8 计算两个 2x2 矩阵	
标量的乘积	74	间的乘法	96
示例 4-11 用直角坐标求向量与		示例 5-9 计算两个不同大小	

矩阵间的乘法	98	扩展	136
自我测试	100	旋转	136
5.5 转置矩阵	101	串联矩阵	136
示例 5-10 3×3 矩阵的转置	101	第 7 章 单位换算	138
示例 5-11 转置一个向量	102	7.1 公制	138
示例 5-12 转置一个矩阵	102	示例 7-1 把米转换成公里。	139
自我测试	103	示例 7-2 把公里转换成米	139
5.6 视觉体验	104	自我测试	139
5.7 自我测试答案	104	7.2 在不同单位制之中转换	139
全等矩阵	104	示例 7-3 把 2 周转换成秒	140
矩阵加减法	105	示例 7-4 距离转换	141
标量与矩阵的乘法	105	示例 7-5 速度转换	142
矩阵之间的乘法	105	示例 7-6 加速度转换	142
求转置矩阵	106	自我测试	143
第 6 章 变换	107	7.3 计算机转换	143
6.1 平 移	107	示例 7-7 把二进制转换成	
示例 6-1 用加法进行 2d 平移 ..	108	十进制	144
示例 6-2 用加法进行 3D 平移 ..	109	示例 7-8 把十进制转换为	
示例 6-3 用乘法进行 2D 平移 ..	110	二进制	144
示例 6-4 用乘法进行 3D 平移 ..	112	二进制与阿拉伯数字之间的	
自我测试	113	转换	145
6.2 缩 放	113	示例 7-9 计算机(单位)转换 ..	147
示例 6-5 2D 均衡缩放	114	自我测试	148
示例 6-6 2D 非均衡扩缩	116	7.4 自我测试答案	148
示例 6-7 3D 均衡扩缩	117	公制	148
示例 6-8 3D 非均衡扩缩	118	在不同单位制之中转换	148
自我测试	119	计算机转换	148
6.3 旋 转	119	第 8 章 一维空间运动	149
示例 6-9 2D 旋转	120	8.1 速率与速度	149
示例 6-10 绕 Y 轴做 3D 旋转 ..	126	示例 8-1 计算匀速运动的路程 ..	149
自我测试	126	示例 8-2 计算匀速运动物体的	
6.4 串 联	127	新位置	150
示例 6-11 根据中心对 3D 物体		示例 8-3 计算平均速度	151
进行扩缩	129	示例 8-4 几帧间的平均速度	152
示例 6-12 3D 旋转串联	129	自我测试	152
自我测试	133	8.2 加速度	152
6.5 视觉体验	134	示例 8-5 计算加速度	153
6.6 自我测试答案	136	示例 8-6 计算减速或加速度	153
平移	136	自我测试	154

8.3 运动方程	154	示例 10-9 从已知高度跳下	184
示例 8-7 赛车	157	示例 10-10 弹射球	186
示例 8-8 再看赛车	157	自我测试	188
示例 8-9 用多个方程	158	10.3 可视化检测	188
示例 8-10 竖直运动	159	10.4 自我测试答案	190
自我测试	160	利用向量	190
8.4 可视化检测	160	抛物运动	190
8.5 自我测试答案	161	第 11 章 牛顿定律	191
速度与速率	161	11.1 力	191
加速度	161	示例 11-1 计算重量	192
运动方程	161	示例 11-2 计算质量	192
第 9 章 一维空间运动导数	162	示例 11-3 垂直于面的压力	193
9.1 速度图像和导数	162	示例 11-4 斜面压力	193
示例 9-1 计算平均速度	163	示例 11-5 计算摩擦力	195
示例 9-2 计算瞬时速度	165	示例 11-6 计算 2D 合力	198
示例 9-3 计算倒数	165	自我测试	199
自我测试	166	11.2 牛顿三定律	200
9.2 加速度图像和导数	166	示例 11-7 牛顿第一定律应用	
示例 9-4 计算平均加速度	168	于冰球	200
示例 9-5 计算瞬时加速度	169	示例 11-8 牛顿第二定律	200
示例 9-6 计算二阶倒数	170	示例 11-9 牛顿第二定律和 5 个	
自我测试	171	运动方程	201
9.3 自我测试答案	172	自我测试	202
速度图像和导数	172	11.3 自我测试答案	203
加速度图像和导数	172	力	203
第 10 章 二维和三维空间运动	173	用牛顿定律计算力对物体运动的	
10.1 使用向量	173	影响	203
示例 10-1 求 2d 位移	175	第 12 章 能 量	205
示例 10-2 求 3D 位移	175	12.1 功和动能	205
示例 10-3 计算 3D 平均速度	176	示例 12-1 计算功	206
示例 10-4 在 2D 中使用方程	178	示例 12-2 计算有偏角的力所	
示例 10-5 在 3D 中使用方程	178	作的功	207
自我测试	179	示例 12-3 计算力和位移角度	
10.2 抛物运动	179	不同所作的功	208
示例 10-6 抛物运动的竖直分量		示例 12-4 计算动能	210
.....	180	示例 12-5 功能转换定理	211
示例 10-7 抛物运动的水平分量		自我测试	212
.....	181	12.2 势能和机械能守恒定理	212
示例 10-8 从已知高度下落	182	示例 12-6 重力势能	213

示例 12-7 机械能守恒定律	214	示例 14-1 CD-ROM 的平均	
示例 12-8 机械能守恒定律	215	角速度	240
自我测试	215	示例 14-2 计算轮子的平均角	
12.3 自我测试答案	216	加速度	241
功和动能	216	示例 14-3 轮子的命运	242
势能和机械能守恒定律	216	示例 14-4 投棒球	244
第 13 章 动量和碰撞	217	示例 14-5 再次研究轮子的	
13.1 和静止物体的碰撞	217	命运	244
示例 13-1 向量的轴平行与非轴		自我测试	245
平行反弹	218	14.2 旋转力学	245
示例 13-2 向量的非轴平行 2D		示例 14-6 撞车	246
反弹	220	示例 14-7 滚动的球	247
示例 13-3 向量的非轴平行 3D		自我测试	249
反弹	222	14.3 自我测试答案	250
自我测试	223	圆周运动	250
13.2 动量和冲量	223	旋转力学	250
示例 13-4 计算 1D 动量	224	附录 A 一些物理和数学公式	251
示例 13-5 计算 3D 动量	224	点和直线	251
示例 13-6 动量定理	226	几何基础	252
自我测试	226	三角形	253
13.3 研究碰撞	227	向量操作	254
示例 13-7 动量定理的变形	227	矩阵操作	257
示例 13-8 完全非弹性碰撞	228	矩阵变换	259
示例 13-9 弹性碰撞	230	单位换算	261
自我测试	233	运动测量	262
13.4 可视化检测	233	对物体运动的进一步研究	262
清单 13-1 台球碰撞的基类	234	空间几何运动	263
13.5 自我测试答案	238	牛顿定律	264
和静止物体的碰撞	238	能量守恒	264
动量和冲量的介绍	238	动量与碰撞	265
研究碰撞	238	第 14 章 转动	266
第 14 章 旋转运动	239	附录 B 术语表	251
14.1 圆周运动	239		

第 1 章 点和线

本章主题

- 点的定义
- 直线的定义
- 直线的属性
- 相交线的检测与应用

你是否想过编一个游戏，而不知从何入手？怎样才能让电脑明白你要把一个物体放在何处？又怎样让物体移动？这一章的内容就是关于这些问题的，同时也为后面的章节打下基础。

本章首先定义点，就可以把物体放在 3D 或 2D 场景的指定位置，然后确立了直角坐标系，并将在以后的章节里一直沿用。到此，你就可以开始研究 3D 或 2D 场景中的线。点让我们能够把物体放在指定位置，而线却可以确定物体的移动轨迹。在研究线的所有特性之后，再谈一谈两条直线相交问题的检测规则。

1.1 点的定义

在游戏编程中，需要处理许多不同的函数。简单地说，函数是用来把输入值变成输出值的一条规则。

在编程过程中，函数输入值可以是不同类型的，比如变量、文本或常数，它们同样可以输出不同类型的信息，也可以是一种功能。在一个成功的程序中，可以看到里面的重复执行部分是被写成函数的，这样就可以通过调用而不是重复编程，使得你的程序变得简单而又整洁。让我们先来看一个好而有用的函数：

```
// purpose: determine whether a number is a power of 2
// input: the number being checked
// output: true is it's a power of 2, else false
Bool powoftwo(int num) {
    Return !(num & (num - 1));
}
```

这个函数的功能是检测输入的整型数值是否为以 2 为底数的乘方，输出值为真假布尔值 (True, False)，比如 16, 32, 64……

在数学中，函数仅仅应用于数值：输入数值通过指定的规则进行运算，然后输出新的数值。有两种方法可以定义数学函数：一是可以用列举的方法把每个输入值和其对应的输出值一一列举出来。

例如，下面的函数就是采用了把两个值对应列举的方法：

(0, 1), (1, 2), (2, 4), (3, 6)和(4, 8)

也可以把它们写成表格的形式，如表 1-1 所示。

表 1-1 将对应的值进行列表

X 坐标	Y 坐标
0	0
1	2
2	4
3	6
4	8

尽管做一个表格可以简单而清楚地查阅，这种方法还是存在弊端。假如你输入的数值在表格中并不存在，那么就会得到错误信息。我们来看表 1-1，假如你输入数值 5，又会发生什么呢？表格并没有列出输入值为 5 的情况，所以电脑并不知道该怎样处理这个输入值。表格的另一个问题是，必须列出输入值的所有情况，这样就必须建立一个巨大的表格来储存所有数据。于是我们想到另一个更好的方法：可以建立一个公式或一个方程来表达两个数值之间的关系。可以把函数简单地写成 $y=2x$ ， x 是输入值， y 是输出值。尽量使函数适用于所有的输入和输出值。

这会有用么？首先应该看到，这两种方法实际上定义的是同一个函数。可是，使用第二种方法却适用于每一个输入的数值，而不仅仅是在表格中列出的 5 个数值。由于这种方法可以适用于任何输入数，比起表格它大大地节约了存储空间。让我们再来看一个函数的常用功能：把物体放在屏幕的指定位置上。如果你编写一个游戏程序，就需要让电脑把你指定的物体显示到屏幕上。

电脑如何知道该把物体放到什么位置呢？一般地，利用直角坐标系来确定物体在屏幕上的位置。直角坐标系包含一个 x 轴和一个 y 轴。每个点都可以用两个轴上的坐标表示。写成 (x, y) 。原点是两轴相交的地方，它的坐标是 $(0, 0)$ 。从原点出发，向右是 x 轴的正方向，向左是 x 轴的反方向；同样，向上是 y 轴的正方向，向下是 y 轴的反方向。

示例 1-1 在 2D 场景中确定位置

方法：如果需要把 6 个物体 (A-F) 画到屏幕上，用直角坐标系来确定它们的位置： $A(0, 0)$ ， $B(1, 2)$ ， $C(2, 4)$ ， $D(-1, 2)$ ， $E(-2, -1)$ 和 $F(3, -2)$ ，结果如图 1-1 所示。

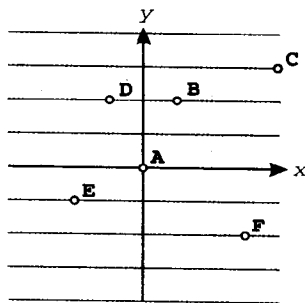


图 1-1 点 A-F 在平面直角坐标系中的位置