

THOMSON

• 游戏开发经典丛书 •



本书配光盘

(美) Brian Schwab 著
林龙信 张波涛 译

AI 游戏 引擎程序设计

AI Game Engine Programming

CENGAGE
Learning™

清华大学出版社

Brian Schwab

AI Game Engine Programming

EISBN: 1-58450-344-0

Copyright © 2004 by Thomson, a division of Thomson Learning

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd).

All rights reserved.

Tsinghua University Press is authorized by Thomson Learning to publish and distribute exclusively this Simplified Chinese edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本中文简体字翻译版由汤姆森学习出版集团授权清华大学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾地区)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

北京市版权局著作权合同登记号 图字: 01-2006-7223

本书封面贴有 **Thomson** 防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

AI 游戏引擎程序设计/(美)施瓦布(Schwab, B.)著; 林龙信, 张波涛译. —北京: 清华大学出版社, 2007.11
(游戏开发经典丛书)

书名原文: AI Game Engine Programming

ISBN 978-7-302-16312-1

I. A… II. ①施… ②林… ③张… III. 游戏—软件开发 IV. TP311.5

中国版本图书馆 CIP 数据核字(2007)第 159567 号

责任编辑: 王 军 梁卫红

装帧设计: 康 博

责任校对: 胡雁翎

责任印制: 何 芊

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 北京鑫丰华彩印有限公司

装 订 者: 三河市李旗庄少明装订厂

经 销: 全国新华书店

开 本: 185×260 印 张: 29.75 字 数: 724 千字

附光盘 1 张

版 次: 2007 年 11 月第 1 版 印 次: 2007 年 11 月第 1 次印刷

印 数: 1~4000

定 价: 59.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系
调换。联系电话: (010)62770177 转 3103 产品编号: 023671-01



译者序

如今电脑游戏正变得越来越普及，与此相应，游戏编程也成了计算机行业内的一个热门领域。特别是随着人工智能理论研究的进展，许多现代游戏都具有了很多的智能成分。如何理解智能？游戏人工智能是什么？有哪些人工智能技术可以应用到游戏编程之中？它们又该如何实现？这些都是摆在智能游戏开发者面前的问题。

但是，现在市面上关于一般的游戏编程的书还不多，而关于“游戏人工智能”编程的书则更少。Brian Schwab 先生的这本 *AI Game Engine Programming* 可谓是给我们这些智能游戏开发者带来了一阵及时雨，它以一种独特深邃的视角，从众多游戏编程书籍中脱颖而出，是一本非常难得的佳作。

本书以严谨的理论说明、丰富的游戏示例以及翔实的程序代码对人工智能游戏引擎编程知识进行了很好的归纳和阐述。尽管本书是为职业的游戏 AI 程序员，以及有志于将其兴趣领域扩展到 AI 的程序员编写的，但其对各个层次的游戏编程人员都有很好的启发作用。希望通过本书的学习，读者能够快速地掌握人工智能游戏引擎的编程技巧，将人工智能真正地应用到所开发的游戏之中。

本书涉及到了相当多的游戏，有些可能是我们相当熟悉的，有些则可能非常陌生。为了防止翻译过来的游戏名称产生歧义，我们仍然采用游戏的英文原名(某些非常熟悉的游戏则在英文后标注了中文游戏名)。另外，由于本书中具有大量的游戏专业词汇，尽管借助了强大的网络搜索工具，但对于某些具体词汇的译法仍然比较模糊，同行的翻译也不是很一致，故在译稿中仅仅按实际含义给出。

本书由林龙信、张波涛翻译完成，但翻译工作是集体智慧的结晶。除了译者外，还得到了翻译工作的组织者以及许多网络游戏玩家的大量指导和帮助，在此表示衷心的感谢。

由于译者水平有限，书中难免会存在不妥之处，敬请广大读者批评指正。

译者

2007 年 4 月于湖南长沙



前 言

现在市面上关于一般游戏编程的书还不多，而关于“游戏人工智能(Game Artificial Intelligence, 游戏 AI)”编程的书则更少。本书将为读者介绍 4 个方面的内容，这些内容都是目前市面上的游戏编程书籍较少涉及到的。

(1) 对“游戏 AI”进行清楚的定义。很多书本使用一个概括的或非常宽泛的概念来对 AI 这个术语进行解释，这样不仅会使读者对解答感到不满意，而且还可能加深对 AI 的“神秘”感，而 AI 在一般大众和行业人士的常识中是一个非常普遍的词汇。这里，将游戏 AI 问题划分成确实符合真实 AI 解决方案的若干部分来考虑。

(2) 对 AI 元素和解决方案按游戏类型进行分解。很多书本仅仅依赖于一种类型的游戏，或一个有限的示例程序。本书将大多数现代游戏类型进行分解，并以实际发布的名称给出 AI 使用技巧的具体示例。通过了解不同游戏类型的 AI 范例选择背后的基本思想，读者将会对范例本身有更多理解。

(3) 对大多数通用的 AI 范例进行代码实现。在本书的后几章中，对于每种 AI 技术都会给出真实代码，这包括以框架的形式和作为真实示例应用的一部分。本书将会对代码进行分解，并对其进行充分的讨论，以帮助说明系统的实际处理过程。

(4) 对未来改进方向进行讨论。对于每种游戏类型和 AI 技术，本书都会给出系统如何扩展的示例。这将通过两种方式来完成：指出当前游戏和经典游戏中的一般的 AI 缺陷；详细给出在空间、速度或其他限制条件下实现系统最优的方式。

本书总揽

本书第 I 部分是对游戏 AI 的一个概述，它涵盖了贯穿于本书的基本术语，对游戏 AI 中的几个基本概念进行了探讨，并详细分析了游戏 AI 引擎的基本系统。第 II 部分对具体的游戏类型以及它们如何使用不同的 AI 范例进行了讨论。尽管本书并不是包罗万象的(即详细论述每种游戏是如何实现的)，但它将对各种类型游戏提出的问题的更一般解决方案进行讨论。第 III 部分给出了基本 AI 技术的真实代码实现。而第 IV 部分则对高级的 AI 技术进行了讨论。第 V 部分对许多概念和关注点进行了分解，着重处理“真正游戏 AI 开发”的各种事项，包括一般的设计与开发问题、作为一个综合范例的“分层式设计 AI”(有助于几乎所有 AI 引擎的组织性)、对 AI 系统的调试以及 AI 的未来。

本书读者

本书旨在向游戏开发人员提供设计现代游戏 AI 引擎所必需的工具，并对当前一些 AI 引擎上应用的技术进行分析。AI 编程是游戏开发过程中一个非常具有挑战性的方向，尽管已经有很多书对一般的游戏相关的数据结构和编码风格进行了阐述，但很少对“游戏 AI”这个重要而且技术性又强的学科领域进行专门论述。

本书是为职业的游戏 AI 程序员，以及有志于将其兴趣领域扩展到 AI 的程序员专门编写的。如果在决定使用哪种技术上存在困难，或者对一个特定游戏最适合的引擎的工作代码存在问题，或者需要这样的运行代码，那么可以从本书中获取答案。本书将为多种有用的游戏 AI 技术提供一个简洁而可用的接口。本书将强调主要的决策范例，因此不会对路径搜索(至少不会直接深入研究，事实上很多本书提到的技术都可用于运行一个路径搜索器)或感知等重要领域进行深入研究——尽管也会对它们进行讨论。

本书假定读者具有 C++ 语言、经典数据结构以及面向对象编程基本知识的应用经验。本书的示例程序是在 Windows[®] 平台下用 Microsoft Visual C++[®] 编写的，但只有渲染是与具体平台相关的，并且所使用的渲染 API 是扩展到 OpenGL 中的 GLUT。因此，如果需要，可以很容易地移植到另外的系统。可参见光盘中关于 GLUT 和 OpenGL 的信息。

读过本书之后，读者将对作为一个游戏 AI 程序员必备的知识结构有个充分的了解。游戏类型的介绍将使得程序员能够了解在给定产品和时间表的情况下，如何从头至尾地构造一个 AI 系统。本书的代码涵盖了非常宽广的游戏类型，几乎可以构造任意类型的 AI 系统，并将详细说明如何将这些技术综合应用到更复杂和可用的具体游戏 AI 引擎中。

目 录

第 I 部分 概述

第 1 章 基本定义与概念	3
1.1 什么是智能	3
1.2 什么是游戏 AI	3
1.3 什么不是游戏 AI	6
1.4 该定义与人工智能理论 定义的区别	8
1.5 可应用的大脑科学与心理学 理论	9
1.5.1 大脑的组织结构	9
1.5.2 知识库与学习	10
1.5.3 认知	12
1.5.4 心智理论	13
1.5.5 有限最优	19
1.5.6 来自机器人技术的启发	20
1.6 小结	22
第 2 章 AI 引擎的基本组成与设计	23
2.1 决策与推理	23
2.1.1 解决方案的类型	24
2.1.2 智能体的反应能力	24
2.1.3 系统的真实性	24
2.1.4 游戏类型	25
2.1.5 游戏内容	25
2.1.6 游戏平台	25
2.1.7 开发限制	27
2.1.8 娱乐限制	28
2.2 输入处理机与感知	29

2.2.1 感知类型	29
2.2.2 更新规则	29
2.2.3 反应时间	30
2.2.4 门限	30
2.2.5 负荷平衡	30
2.2.6 计算代价与预处理	30
2.3 导航	31
2.3.1 基于网格	31
2.3.2 简单避免与位势场	32
2.3.3 地图节点网络	33
2.3.4 导航网格	33
2.3.5 组合系统	34
2.4 综合考虑	35
2.5 小结	36
第 3 章 Alsteroids: AI 试验平台	37
3.1 GameObj 类	38
3.2 GameObj 类的 Update()函数	39
3.3 Ship 对象	40
3.4 其他游戏对象	41
3.5 GameSession 类	42
3.5.1 主逻辑与碰撞检测	43
3.5.2 对象清除	45
3.5.3 主飞船和宝物的产生	45
3.5.4 奖励生命	46
3.5.5 级别和游戏的结束	46
3.6 Control 类	47
3.7 AI 系统钩子	47
3.8 游戏主循环	48
3.9 小结	48

第 II 部分 游戏类型

第 4 章 角色扮演类游戏	53
4.1 通用 AI 元素	57
4.1.1 敌人	57
4.1.2 头目	58
4.1.3 非玩家角色	58
4.1.4 店员	59
4.1.5 队员	59
4.2 有用的 AI 技术	60
4.2.1 脚本	60
4.2.2 有限状态机	61
4.2.3 消息	62
4.3 示例	62
4.4 例外	63
4.5 需要改进的具体游戏元素	64
4.5.1 角色扮演不等于战斗	64
4.5.2 语法机器	64
4.5.3 任务发生器	65
4.5.4 更好的队员 AI	65
4.5.5 更好的敌人	66
4.5.6 完全真实的市镇	67
4.6 小结	68
第 5 章 冒险类游戏	69
5.1 通用 AI 元素	70
5.1.1 敌人 AI	70
5.1.2 非玩家角色	70
5.1.3 协作元素	71
5.1.4 感知系统	71
5.1.5 摄像机	71
5.2 有用的 AI 技术	71
5.2.1 有限状态机	71
5.2.2 脚本系统	72
5.2.3 消息系统	72
5.2.4 模糊逻辑	72
5.3 示例	73
5.4 需要改进的领域	74
5.4.1 潜行目标的附加类型	74

5.4.2 传统冒险根源的回归	74
5.4.3 更好的 NPC 通信	74
5.4.4 用户界面	74
5.5 小结	75
第 6 章 即时策略游戏	77
6.1 通用 AI 元素	77
6.1.1 个体单元	77
6.1.2 雇佣个体单元	78
6.1.3 指挥官与中级战略性元素	78
6.1.4 高层战略性 AI	78
6.1.5 市镇构建	79
6.1.6 本土生活	79
6.1.7 路径搜索	79
6.1.8 战术与战略支撑系统	79
6.2 有用的 AI 技术	81
6.2.1 消息	81
6.2.2 有限状态机	81
6.2.3 模糊状态机	82
6.2.4 层次化 AI	82
6.2.5 规划	82
6.2.6 脚本	82
6.2.7 数据驱动 AI	83
6.3 示例	84
6.4 需要改进的领域	85
6.4.1 学习	85
6.4.2 确定 AI 元素何时受困	85
6.4.3 AI 助手	86
6.4.4 对抗人物	86
6.4.5 多战略少战术	87
6.5 小结	88
第 7 章 第一人称/第三人称射击游戏	89
7.1 通用 AI 元素	91
7.1.1 敌人	91
7.1.2 敌人头目	92
7.1.3 死亡竞赛对手	92
7.1.4 武器	92
7.1.5 协作智能体	93
7.1.6 分队成员	93

7.1.7 路径搜索	93	9.2 有用的 AI 技术	119
7.1.8 空间推理	94	9.2.1 有限状态机	119
7.2 有用的 AI 技术	94	9.2.2 脚本系统	119
7.2.1 有限状态机	94	9.2.3 数据驱动系统	119
7.2.2 模糊状态机	97	9.3 例外	120
7.2.3 消息系统	98	9.4 示例	120
7.2.4 脚本系统	98	9.5 需要改进的领域	121
7.3 示例	98	9.6 小结	122
7.4 需要改进的领域	99	第 10 章 运动类游戏	123
7.4.1 学习与对手建模	99	10.1 通用 AI 元素	124
7.4.2 个性	100	10.1.1 教练或团队级别 AI	124
7.4.3 创造力	100	10.1.2 玩家级别 AI	124
7.4.4 预测	100	10.1.3 路径搜索	125
7.4.5 更好的会话引擎	101	10.1.4 摄像机	125
7.4.6 动机	101	10.1.5 混杂元素	126
7.4.7 更好的分队 AI	101	10.2 有用的 AI 技术	127
7.5 小结	101	10.2.1 有限状态机与模糊 状态机	127
第 8 章 平台游戏	103	10.2.2 数据驱动系统	133
8.1 通用 AI 元素	108	10.2.3 消息系统	134
8.1.1 敌人	108	10.3 示例	134
8.1.2 敌人头目	108	10.4 需要改进的领域	135
8.1.3 协作元素	108	10.4.1 学习	135
8.1.4 摄像机	109	10.4.2 游戏平衡	135
8.2 有用的 AI 技术	110	10.4.3 玩法创新	135
8.2.1 有限状态机	110	10.5 小结	136
8.2.2 消息系统	110	第 11 章 赛车游戏	139
8.2.3 脚本系统	110	11.1 通用 AI 元素	140
8.2.4 数据驱动系统	110	11.1.1 轨迹 AI	140
8.3 示例	110	11.1.2 交通	141
8.4 需要改进的领域	111	11.1.3 行人	142
8.4.1 摄影技巧	111	11.1.4 敌人与战斗	142
8.4.2 帮助系统	111	11.1.5 非玩家角色	142
8.5 小结	112	11.1.6 其他竞争行为	142
第 9 章 射击类游戏	113	11.2 有用的 AI 技术	143
9.1 通用 AI 元素	118	11.2.1 有限状态机	143
9.1.1 敌人	118	11.2.2 脚本系统	143
9.1.2 敌人头目	118	11.2.3 消息系统	143
9.1.3 协作元素	119		

11.2.4 遗传算法.....	143	第 14 章 著名的混杂游戏类型.....	167
11.3 示例.....	143	14.1 文明游戏.....	167
11.4 需要改进的领域.....	144	14.2 天神游戏.....	174
11.4.1 除犯罪以外的其他感 兴趣领域.....	144	14.3 战争游戏.....	177
11.4.2 更多的智能 AI 敌人.....	144	14.4 飞行模拟游戏.....	181
11.4.3 永不落幕的游戏世界.....	144	14.5 音乐游戏.....	186
11.5 小结.....	145	14.6 益智游戏.....	186
第 12 章 经典策略游戏.....	147	14.7 人工生命游戏.....	187
12.1 通用 AI 元素.....	156	第 III 部分 基本的 AI 引擎技术	
12.1.1 对手 AI.....	156	第 15 章 有限状态机.....	193
12.1.2 AI 助手.....	157	15.1 FSM 概述.....	193
12.2 有用的 AI 技术.....	157	15.2 FSM 骨架代码.....	197
12.2.1 有限状态机.....	157	15.2.1 FSMState 类.....	197
12.2.2 Alpha-Beta 搜索.....	157	15.2.2 FSMMachine 类.....	198
12.2.3 神经网络.....	158	15.2.3 FSMAIControl 类.....	199
12.2.4 遗传算法.....	158	15.3 在试验平台上实现 FSM 控制的飞船.....	200
12.3 例外.....	158	15.4 示例实现.....	201
12.4 示例.....	158	15.4.1 Control 类编码.....	202
12.5 需要改进的领域.....	159	15.4.2 状态编码.....	204
12.5.1 创造力.....	159	15.5 使用该系统的 AI 的性能.....	213
12.5.2 速度.....	159	15.5.1 基于 FSM 系统的优势.....	213
12.6 小结.....	159	15.5.2 基于 FSM 系统的劣势.....	214
第 13 章 格斗类游戏.....	161	15.6 范例扩展.....	215
13.1 通用 AI 元素.....	162	15.6.1 层次化 FSM.....	215
13.1.1 敌人.....	162	15.6.2 基于消息和事件的 FSM.....	216
13.1.2 碰撞系统.....	163	15.6.3 具有模糊转换的 FSM.....	216
13.1.3 敌人头目.....	163	15.6.4 基于堆栈的 FSM.....	216
13.1.4 摄像机.....	163	15.6.5 多重并发 FSM.....	217
13.1.5 动作和冒险元素.....	164	15.6.6 数据驱动 FSM.....	217
13.2 有用的 AI 技术.....	164	15.6.7 惯性 FSM.....	218
13.2.1 有限状态机.....	164	15.7 最优化.....	219
13.2.2 数据驱动系统.....	164	15.7.1 FSM 和感知的负荷平衡.....	219
13.2.3 脚本系统.....	164	15.7.2 LOD AI 系统.....	219
13.3 示例.....	165	15.7.3 共享数据结构.....	220
13.4 需要改进的领域.....	165	15.8 设计上考虑的因素.....	220
13.5 小结.....	166	15.8.1 解决方案的类型.....	221

15.8.2	智能体的反应能力	221	16.9.4	游戏类型	246
15.8.3	系统的真实性	221	16.9.5	游戏平台	246
15.8.4	游戏类型	221	16.9.6	开发限制	246
15.8.5	游戏内容	222	16.9.7	娱乐限制	246
15.8.6	游戏平台	222	16.10	小结	246
15.8.7	开发限制	222	第 17 章	基于消息的系统	249
15.8.8	娱乐限制	222	17.1	消息概述	249
15.9	小结	223	17.2	消息的骨架代码	250
第 16 章	模糊状态机	225	17.2.1	Message 对象	250
16.1	FuSM 概述	225	17.2.2	MessagePump 类	251
16.2	FuSM 骨架代码	228	17.3	客户端句柄	255
16.2.1	FuSMState 类	228	17.4	在 AIsteroids 试验平台上的 示例实现	256
16.2.2	FuSMMachine 类	230	17.4.1	MessState 类	256
16.2.3	FuSMAIControl 类	231	17.4.2	MessMachine 类	257
16.3	在试验平台上实现 FuSM 控制的飞船	232	17.4.3	MessAIControl 类	258
16.4	示例实现	232	17.5	状态编码	262
16.4.1	添加 Saucer	232	17.6	使用该系统的 AI 的性能	265
16.4.2	其他的游戏修改	233	17.6.1	消息系统的优势	265
16.4.3	FuSM 系统	233	17.6.2	消息系统的劣势	266
16.5	控制类编码	234	17.7	范例扩展	266
16.6	使用该系统的 AI 的性能	241	17.7.1	消息优先级	266
16.6.1	基于 FuSM 系统的优势	241	17.7.2	消息仲裁	267
16.6.2	基于 FuSM 系统的劣势	242	17.7.3	自动和扩展的消息类型	267
16.7	范例扩展	243	17.8	最优化	268
16.7.1	有限数量当前状态的 FuSM	243	17.9	设计上考虑的因素	268
16.7.2	作为角色支持系统的 FuSM	243	17.9.1	解决方案的类型	268
16.7.3	在较大 FSM 中作为 单一状态的 FuSM	244	17.9.2	智能体的反应能力	268
16.7.4	层次化 FuSM	244	17.9.3	系统的真实性	268
16.7.5	数据驱动 FuSM	244	17.9.4	游戏类型和平台	268
16.8	最优化	245	17.9.5	开发限制	269
16.9	设计上考虑的因素	245	17.9.6	娱乐限制	269
16.9.1	解决方案的类型	245	17.10	小结	269
16.9.2	智能体的反应能力	245	第 18 章	脚本系统	271
16.9.3	系统的真实性	245	18.1	脚本概述	271
			18.2	AIsteroids 测试平台中的 脚本实现	272
			18.2.1	一种配置脚本语言	273

18.2.2	配置脚本系统的 AI 性能分析	278	19.1.2	智能地形技术(Smart Terrain)	300
18.2.3	游戏中 Lua 的嵌入	278	19.1.3	地形分析技术(Terrain Analysis, TA)	301
18.3	Lua 在 AIsteroids 测试平台中的实现	286	19.2	各种技术的使用方法	301
18.4	Lua 脚本系统的 AI 性能分析	290	19.2.1	占用数据	301
18.5	脚本系统的优点	290	19.2.2	场地控制	301
18.5.1	快速原型开发	290	19.2.3	探路系统的辅助数据	302
18.5.2	更低的门槛	291	19.2.4	危险预警	302
18.5.3	更快的 AI 调试速度	291	19.2.5	初步战场计划	303
18.5.4	更多的用户扩展手段	291	19.2.6	简单战场分析	303
18.5.5	更广的适用范围	291	19.2.7	高级战场分析	303
18.6	脚本系统的缺点	291	19.3	影响图框架代码及测试平台实现	305
18.6.1	执行速度	292	19.3.1	占用影响图	310
18.6.2	调试难度	292	19.3.2	占用 IM 测试平台的使用	314
18.6.3	脚本作用	292	19.3.3	控制影响图	315
18.6.4	宿主代码和脚本的功能划分	293	19.3.4	控制 IM 测试平台的使用	318
18.6.5	需维护的系统数量	293	19.3.5	逐位影响图	319
18.7	范例扩展	294	19.3.6	逐位 IM 测试平台的使用	324
18.7.1	自定义语言	294	19.3.7	其他实现	324
18.7.2	内建调试工具	294	19.4	基于位置的信息系统的优点	326
18.7.3	智能脚本 IDE	294	19.5	基于位置的信息系统的缺点	326
18.7.4	游戏脚本自动集成	295	19.6	范例扩展	326
18.7.5	自主修改脚本	295	19.7	优化	326
18.8	优化	295	19.8	设计上考虑的因素	327
18.9	设计上考虑的因素	296	19.8.1	解决方案的类型	327
18.9.1	解决方案的类型	296	19.8.2	智能体的反应能力	327
18.9.2	智能体的反应能力	296	19.8.3	系统的真实性	327
18.9.3	系统的真实性	296	19.8.4	游戏类型和平台	328
18.9.4	游戏类型和平台	297	19.8.5	开发限制	328
18.9.5	开发限制	297	19.8.6	娱乐限制	328
18.9.6	娱乐限制	297	19.9	小结	328
18.10	小结	297			
第 19 章	基于位置的信息系统	299			
19.1	基于位置的信息系统概述	299			
19.1.1	影响图技术(IM)	299			

第IV部分 高级 AI 引擎技术

第 20 章	遗传算法	331
20.1	遗传算法概述	331
20.1.1	自然进化规律	331
20.1.2	游戏中的进化	332
20.1.3	遗传算法基本过程	333
20.2	问题的表示	334
20.2.1	基因和基因组	335
20.2.2	适应度函数	336
20.2.3	繁殖	337
20.3	AIsteroids 测试平台中遗传算法的实现	342
20.4	遗传算法在测试平台中的性能	354
20.5	基于遗传算法的系统的优点	356
20.6	基于遗传算法的系统的缺点	357
20.6.1	时间代价较大	357
20.6.2	算法性能随机性大	357
20.6.3	结果成败定义模糊	358
20.6.4	最优解不能保证	358
20.6.5	参数调试和扩展难度大	358
20.7	范例扩展	358
20.7.1	蚁群算法	358
20.7.2	协同进化	359
20.7.3	自适应遗传算法	359
20.7.4	遗传程序设计	359
20.8	设计上考虑的因素	360
20.8.1	解决方案的类型	360
20.8.2	智能体的反应能力	360
20.8.3	系统的真实性	360
20.8.4	游戏类型	360
20.8.5	平台	361
20.8.6	开发限制	361
20.8.7	娱乐限制	361
20.9	小结	361

第 21 章	神经网络	363
21.1	自然中的神经网络	363
21.2	人工神经网络概述	364
21.3	神经网络的使用	366
21.3.1	结构	366
21.3.2	学习机制	367
21.3.3	创建训练数据	368
21.4	神经网络活动	368
21.5	在 AIsteroids 测试平台上实现神经网络	371
21.5.1	NeuralNet 类	371
21.5.2	NLayer 类	375
21.5.3	NNAIControl 类	379
21.6	测试平台的性能	384
21.7	优化	385
21.8	基于神经网络的系统的优点	386
21.9	基于神经网络的系统的缺点	386
21.10	范例扩展	387
21.10.1	其他类型的神经网络	388
21.10.2	神经网络学习的其他类型	388
21.11	设计上考虑的因素	389
21.11.1	解决方案的类型	389
21.11.2	智能体的反应能力	389
21.11.3	系统的真实性	389
21.11.4	游戏类型和平台	390
21.11.5	开发限制	390
21.11.6	娱乐限制	390
21.12	小结	390
第 22 章	其他技术备忘录	393
22.1	人工生命	393
22.1.1	人工生命在游戏中的应用	394
22.1.2	人工生命学科	394
22.1.3	优点	395
22.1.4	缺点	396

22.1.5	游戏设计可以开发的领域	396
22.2	规划算法	396
22.2.1	当前在游戏中的使用状况	397
22.2.2	优点	398
22.2.3	缺点	399
22.2.4	游戏设计可以开发的领域	399
22.3	产生式系统	400
22.3.1	优点	401
22.3.2	缺点	401
22.3.3	游戏设计可以开发的领域	401
22.4	决策树	402
22.4.1	优点	403
22.4.2	缺点	403
22.4.3	游戏设计可以开发的领域	404
22.5	模糊逻辑	404
22.5.1	优点	406
22.5.2	缺点	406
22.5.3	游戏设计可以开发的领域	406
22.6	小结	406

第V部分 AI 实战游戏开发

第23章	分层式 AI 设计	411
23.1	基本回顾	411
23.2	分层式层结构	412
23.2.1	重现前述示例	413
23.2.2	感知和事件层	414
23.2.3	行为层	414
23.2.4	动画层	415
23.2.5	运动层	417
23.2.6	短期决策层(ST)	418
23.2.7	长期决策层(LT)	418
23.2.8	基于位置的信息层	418

23.3	BROOKS 包容式体系结构	419
23.4	游戏层次分解	419
23.4.1	目标	419
23.4.2	分层式超级玛莉	420
23.4.3	AI 怪物的实现	420
23.4.4	AI 玩家的实现	424
23.5	小结	427

第24章 AI 开发中普遍关心的问题 429

24.1	有关设计的问题	429
24.1.1	数据驱动系统设计时需考虑的问题	429
24.1.2	“一根筋”(OTM)综合症	431
24.1.3	多细节层次(LOD)AI	432
24.1.4	支持 AI	434
24.1.5	通用 AI 设计思想	435
24.2	有关娱乐的问题	436
24.2.1	所有重要的趣味性因素	436
24.2.2	随机感	437
24.2.3	一些令 AI 系统看上去非常愚蠢的因素	438
24.3	有关产品的问题	439
24.3.1	保持 AI 行为的一致性	439
24.3.2	提前思考游戏参数的调试问题	440
24.3.3	预防 AI 系统的未知行为	440
24.3.4	注意设计人员工具使用方式的差异性	441
24.4	小结	441

第25章 调试 443

25.1	AI 系统的通用调试	443
25.2	可视化调试	443
25.2.1	提供各种信息	443
25.2.2	有助于调试	444
25.2.3	时序信息	444
25.2.4	监视状态转变	444
25.2.5	有助于控制台调试	444

25.2.6	调试脚本语言	444	25.3.7	程序集成	452
25.2.7	双功能影响图	444	25.4	小结	456
25.3	Widget	445	第 26 章	总结与展望	457
25.3.1	实现	445	26.1	AI 引擎设计总结	457
25.3.2	BasicButton	448	26.2	AI 游戏的未来展望	457
25.3.3	Watcher	449	附录	有关 CD-ROM 的说明	459
25.3.4	RadioButton	449			
25.3.5	OnOffButton	450			
25.3.6	ScrubberWidget	451			

第 I 部分 概述

第 1 章将对“游戏人工智能(Game Artificial Intelligence, 游戏 AI)”进行定义, 并用一个合适的术语来代替这个含糊的表述。也将从心理学界和人工智能理论界的角度出发讨论一些适当可用的 AI 理论。

第 2 章将介绍构成一般的游戏 AI 引擎的基本系统, 以及设计一个新的 AI 引擎时需要考虑的各个要素。

第 3 章将引用并讨论一个基本的应用示例。该示例在本书后面各部分中将作为一个 AI 实现的试验平台。

1

基本定义与概念

欢迎阅读《AI 游戏引擎程序设计》这本书。本书旨在向游戏 AI 的程序员提供开发面向现代商业游戏的 AI 引擎所需要的知识和工具。但究竟什么是“游戏 AI”？AI 是一门相对年轻的科学，其早期的一些工作完成于 19 世纪 50 年代初期。受早期游戏机计算能力和存储空间的限制，游戏采用真正 AI 技术的历史则更短。由于 AI 在游戏中是一个崭新的概念，所以关于游戏 AI 的定义对于大多数人，甚至对那些从事游戏开发的人来说也都还不清楚。本章将对“游戏 AI”这个术语进行定义，区分游戏 AI 中常常混用的实践和技术问题，并对其未来可能的扩展领域进行讨论。在本章的后面，还会围绕游戏 AI 对认知科学、心理学和机器人技术等其他领域的相关概念进行讨论。

1.1 什么是智能

“智能”这个术语是相当模糊的。从字典里可以查知，它是指“获取和应用知识的能力”，但这个解释太笼统。从字面上解释，该定义可能意味着自动调温器是智能的。因为它可以获取房间太冷这个知识，从而应用它所学到的知识去调节加热器。字典接着表明，智能体现了“思考和推理的能力”。尽管这个定义稍微好些(有更多的限制，从而把自动调温器排除在外)，但它也仅仅是通过引入两个甚至更不清晰的术语“思考”和“推理”来增加我们定义的困难。事实上，对智能的“真正”定义是一个古老而又折磨人的争论，它远远超出了本文的范围。值得庆幸的是，设计一个好的游戏并不需要这个定义。实际上，本文赞同字典里的第一个定义，因为它与我们期望在开发的游戏系统中看到的被认为是智能的行为非常吻合。出于我们自己的目的，一个智能的游戏智能体(agent)是能够获取关于这个世界的知识，并对该知识做出反应的智能体。这种反应的质量和效果就是游戏需要权衡和设计的问题。

1.2 什么是游戏 AI

让我们先对 AI 作一个理论上的定义。在 AI 的经典著作 *Artificial Intelligence: A Modern Approach*(人工智能：一种现代方法)中，Russel 和 Norvig 指出，AI 就是设计计算机程序，