



智能科学与技术本科专业系列教材

HINENG KEXUE YU JISHU BENKE ZHUANYE XILIE JIAOCAI

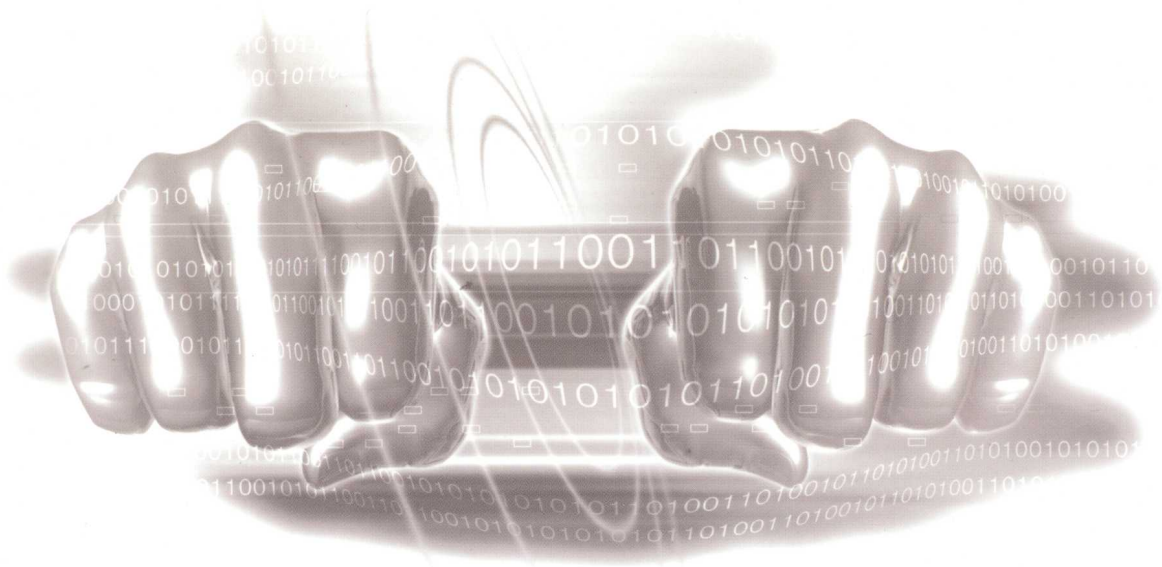
游戏人工智能

——计算机游戏中的人工智能

〔美〕方约翰(John David Funge) 著

李睿凡 郭燕慧 吴昕 译

涂晓媛 校



北京邮电大学出版社
www.buptpress.com

智能科学与技术本科专业系列教材

游戏人工智能

——计算机游戏中的人工智能

Artificial Intelligence for Computer Games

[美] 方约翰 (John David Funge) 著

李睿凡 郭燕慧 吴昕 译

涂晓媛 校

北京邮电大学出版社

· 北京 ·

元00.81

ISBN 978-7-5632-1402-2/P · 281

speech (语音) 北京邮电大学出版社

Artificial Intelligence for Computer Games: An Introduction

Copyright © 2004 by A K Peters, Ltd.

Original English edition: Artificial Intelligence for Computer Games: An Introduction

© 2004 A K Peters, Ltd. ALL RIGHTS RESERVED.

英文原版: 游戏人工智能——计算机游戏中的人工智能由A K Peters, Ltd. 出版。

© 2004 A K Peters, Ltd. 版权所有。

内 容 简 介

游戏角色的智能水平是一款游戏可玩性的决定因素之一, 因而也是游戏开发中需要考虑的重要问题之一。本书作者获加拿大多伦多大学博士学位, 现任职于美国一家著名的游戏开发公司, 具有深厚的理论基础与丰富的开发经验。本书以游戏角色表现的智能行为为主线, 对游戏人工智能的设计与实现进行了导论性的介绍。书中图表丰富, 并附加对关键技术的代码实现, 易于理解。本书可供高年级大学生、研究生以及广大游戏开发者参阅。

北京市版权局著作权合同登记号 图字: 01-2006-0849

图书在版编目(CIP)数据

游戏人工智能: 计算机游戏中的人工智能/(美)方约翰(Funge, J. D.)著; 李睿凡, 郭燕慧, 吴昕译. —北京: 北京邮电大学出版社, 2007

ISBN 978-7-5635-1405-2

I. 游… II. ①方… ②李… ③郭… ④吴… III. 人工智能—应用—游戏—软件设计 IV. TP3115

中国版本图书馆CIP数据核字(2007)第060069号

书 名: 游戏人工智能——计算机游戏中的人工智能

作 者: [美] 方约翰

译 者: 李睿凡 郭燕慧 吴 昕

责任编辑: 李欣一

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路10号(邮编: 100876)

北方营销中心: 电话: 010-62282185 传真: 010-62283578

南方营销中心: 电话: 010-62282902 传真: 010-62282735

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm × 960 mm 1/16

印 张: 8.75

字 数: 145千字

印 数: 1-3 000册

版 次: 2007年6月第1版 2007年6月第1次印刷

ISBN 978-7-5635-1405-2/TP · 281

定 价: 18.00元

· 如有印装质量问题, 请与北京邮电大学出版社营销中心联系 ·

中文版序

智能游戏和动画领域的知名学者方约翰博士 (Dr. John David Funge) 的第一本专著《人工智能在游戏动画中的应用——认识建模方法》中译本已于2004年出版, 受到读者欢迎。

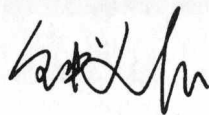
现在, 他的新书《游戏人工智能——计算机游戏中的人工智能》的中译本又将和我国读者见面。

“游戏人工智能”(Game AI) 是人工智能 (AI) 科学技术领域一个新兴的、活跃的学科分支, 是计算机游戏和人工智能相结合的产物。一方面, 研究如何将人工智能的理论、方法和技术应用于设计智能游戏, 提高角色的智能水平; 另一方面, 研究如何开发适用于计算机游戏的人工智能新理论、新方法、新技术。

《游戏人工智能》系统地论述了“游戏人工智能”的基本概念和学科内容, 包括: 智能游戏角色的行动、感知、反应、记忆、搜索与学习的理论方法和技术, 重点研究如何提高“非玩家角色”的智能水平问题。深入浅出, 简明易学, 适应广大计算机游戏设计人员和爱好者的需要。

我相信, 《游戏人工智能》新书出版, 将对我国“游戏人工智能”的学科发展和实际应用作出重要的贡献。

中国人工智能学会理事长
北京邮电大学教授



2007年3月18日

前 言

本书是我第二本关于人工智能与计算机游戏的书籍。第一本书源于我的博士论文。虽然该书难免艰深，但收效良好。一些知名游戏开发人员所表达的阅读兴趣与受益令我深感欣慰。本书较之前者更易介入，适应读者广泛。我相信本书对于有志成为游戏人工智能的程序设计人员，以及那些已在此领域工作的人们有所裨益。

我也希望本书能够引发人工智能学术界对于计算机游戏研究的新兴趣。有一种倾向认为，游戏只不过是人工智能的一个应用领域，因而无须特别的关注。虽然许多通用的人工智能算法可用于游戏开发，但是其中仍蕴涵大量的研究机遇。

目前，已有几本专门探讨或部分涉及人工智能与游戏的书籍，试图以纲要形式列出人工智能相关的文章。与之相比，本书则从统一的、整体的视角对游戏人工智能进行系统的论述。更确切地说，本书将围绕“非玩家角色与如何赋予其智能”这一主题展开讨论。

同时，现在已出版了大量关于人工智能的通用书籍，可为人们提供坚实的基础知识，也就无须重复那些相同的概念与算法。因而本书专注于“人工智能在游戏中的应用”，并为读者提供一些容易获得的导引性参考资料。另一方面，本书还指出了人工智能与计算机游戏相关的某些前沿方向。

当今时代，互联网为我们提供了计算机游戏与人工智能的丰富资源，包含了大量算法与技术的细节。当然，本书也提供了相应的参考文献（包括网络资源）。由于传统参考文献的形式已不合时宜，为此，本书提供了网站www.ai4games.org作为本书的额外信息资源。

我的第一本书，使我投身于游戏产业的人工智能技术开发。这种经历使我更深刻地理解人工智能对于游戏的作用，而这正是大多数学术研究人员所欠缺的。

比如，学术研究人员谈到人工智能对于游戏的意义时可能滔滔不绝，但是，从商业的角度看，人工智能程序并非总是游戏开发的重中之重，新技术可能导致的风险也需要考虑。因此，游戏产业中的商务人士真正关心的是新技术所带来的效益能否为他们提供有助于市场竞争的新产品。

通常，这种新效益来自计算机图形学，如三维技术、纹理映射以及最新的实时程序渲染语言。这样，游戏不断推动计算机图形学的发展而冷落了人工智能。直到人工智能技术能够为主流玩家提供一些令人兴奋的效果，这种情况才会改变。

终有一天，计算机图形学将不再成为游戏技术的驱动力。效用递减规律已使普通玩家无法区分昔日与今朝图形技术的差别。人工智能有潜力成为游戏革新的新驱动力，新兴的技术能够带来新风格的游戏。我希望本书有助于读者思考人工智能对新游戏的效益，激发游戏设计者与玩家对人工智能的热情。

John Funge
方约翰

致 谢

现在，我作为合伙人在一家新成立的公司工作，致力于娱乐产业中人工智能技术的研发。我希望本书能够借助它所建立的一般框架与术语，为游戏开发人员与人工智能中间件公司建立良好的沟通。在本书的写作过程中，我们对所开发的技术负有保密的义务，因此，我非常谨慎以避免涉及保密信息。然而，我对于游戏人工智能一般问题的许多思想都受到我的工作经历与周围同事的影响。所以，我对现在和以往的同事深表感激，他们是：Brian Cabral、Wolff Dobson、Nigel Duffy、Michael McNally、Ron Musick、Stuart Reynolds、Xiaoyuan Tu、Ian Eright、Wei Yen。

Wei Yen CEO是一位成功人士，感谢他提供如此有教益、有趣味的环境，给我与他合作的机会。Ian、Michael和Wolff都有成功开发商用游戏软件的经验；他们的经验对于我理解人工智能如何应用于游戏编程是非常宝贵的。Ian校对了全书，并在本书的写作过程中就风格与内容提出了很多有益的建议，提高了本书的水平。Michael在编程和游戏软件结构的组织方面给我很多指教。Ian和Brian还促进我对于编程技术的认识。Nigel和Ron与我讨论了大量人工智能的问题；Stuart特别在强化学习方面给了我许多指教。Xiaoyuan是我们核心技术开发的中心，她帮助我更深入地理解人工智能。Dale Schuurmans和Stuart Russell曾帮助我从更为现代的观点理解人工智能。感谢Benjamin Funge做了本书的部分校对。

多年来，我同游戏产业人员的交谈中受益匪浅，对于他们我深表感激。本书中的不少概念引用于其他人，如有任何的疏忽与错误，显然责任在我自己。

Xiaoyuan是我的同事，也是我的妻子。我衷心感谢她的真爱、善良和支持。

最后，我要感谢Alice Peters及所有A K Peters出版社的工作人员在本书的撰写过程中对我的支持、理解和鼓励。特别感谢Jonathan Peters及Garage Game公司的有关人员，他们花费时间和精力进行了艺术创作，本书才有如此美丽的封面。

John Funge

方约翰

美国加州Sunnyvale

目 录

第 1 章 概述	1
1.1 计算机游戏的角色	1
1.2 角色行为控制	2
1.3 游戏系统结构	4
1.4 人工智能	6
1.5 游戏人工智能	8
1.5.1 游戏设计	10
1.5.2 计算机游戏与人工智能研究	11
第 2 章 行动	13
2.1 追捕游戏	13
2.2 游戏状态	14
2.3 仿真器	18
2.3.1 行动	18
2.3.2 动画	19
2.3.3 牛顿物理学	19
2.3.4 追捕游戏物理学	20
2.3.5 时间的推移	21
2.3.6 碰撞	21
2.3.7 定时步仿真	22
2.3.8 离散事件仿真	23
2.4 控制器	24
2.4.1 分级控制	25
2.4.2 分级行动	26
2.4.3 细节级别	28
2.4.4 游戏行动的相容性	28

2.5 可行的行动	29
第 3 章 感知	31
3.1 渲染器	31
3.2 仿真感知	33
3.3 针对角色的感知特征	35
3.3.1 “我”的取值	35
3.3.2 其他重要角色	35
3.3.3 提示	36
3.3.4 相对值	36
3.4 部分可观性	37
3.4.1 可见性	38
3.4.2 模仿有噪声的传感器	39
3.4.3 离散化	39
3.5 预测器感知特征	40
第 4 章 反应	45
4.1 反应式控制器的定义	45
4.2 简单的感知特征函数	48
4.3 反应型产生式规则	50
4.4 决策树	51
4.5 逻辑推理	54
第 5 章 记忆	57
5.1 控制器定义	57
5.2 记忆感知特征	58
5.3 信念维护	62
5.3.1 弱化记忆感知特征	62
5.3.2 作弊	64
5.4 精神状态变量	65
5.4.1 情感	66
5.4.2 有限状态机	67
5.4.3 产生式规则	68
5.4.4 逻辑推理	68
5.5 通信	68

5.6 随机化控制器	70
第 6 章 搜索	73
6.1 离散追捕游戏	73
6.2 采用搜索技术的控制器	74
6.3 多步向前搜索	77
6.4 连续域搜索	80
6.4.1 启发式搜索	81
6.4.2 重新规划	83
6.5 路标	83
6.6 对抗搜索	86
6.7 搜索的渲染	91
6.8 广义目标行动规划	92
第 7 章 学习	95
7.1 仿真学习	95
7.2 定义	96
7.3 奖励	99
7.4 记忆	101
7.5 强化学习	103
7.5.1 罗盘方向	104
7.5.2 路标	104
7.5.3 转换模型	105
7.5.4 收敛性	108
7.5.5 函数逼近	109
附录 A 选择	110
A.1 最可能选择	110
A.2 随机选择	111
附录 B 编程	113
参考文献	116
索引	124

第1章 概述

“计算机游戏” (Computer Game), 或更准确地称为“视频游戏” (Video Game), 始于1958年名为“两人网球” (Tennis for Two) 的游戏。直到20世纪70年代, Atari公司成功推出了名为Pong的控制台游戏, 才使更多的人开始关注计算机游戏。此后, 每年都不断有许多计算机游戏产品推出, 形成了一个数十亿美元的产业。尽管游戏的名目众多, 但游戏的类型却十分有限。游戏类型的精确划分还存在争议, 如同电影那样, 一款游戏属于哪种类型, 人们也是各执己见。本书网站 (www.ai4games.org) 提供了一些网站链接, 其中, 某些网站对游戏类型进行了系统的分类, 列举了每种类型的代表作品; 某些网站对计算机游戏的历史和现状做了有趣的介绍。

1.1 计算机游戏的角色

计算机游戏“古墓丽影”中的Lara Croft、“超级马里奥”中的Mario、“吃豆先生”中的Pac-Man都是众所周知的计算机游戏角色。他们也是“玩家角色” (Player Character, PC) 的典型代表。所谓“玩家角色”是指: 行为由玩家通过操纵杆等输入设备控制的游戏角色。例如, 玩家按“A”键, 让角色跳跃, 按“B”键, 让角色出拳, 向前按住拇指操纵杆, 让角色向前走等。通常, 玩家角色在游戏中扮演英雄, 而“非玩家角色” (Non-Player Character, NPC) 扮演其他角色, 比如恶棍、随从 (side-kicks) 和无名小兵 (cannon-fodder)。Wario (Mario的邪

恶兄弟)、Blink、Pinky、Inky和Clyde (Pac-Man中的4个幽灵)是几个著名非玩家角色。

玩家角色和非玩家角色并不总是截然分开的。例如,许多游戏允许玩家在不同的时间控制不同的角色。在这种情况下,玩家角色和非玩家角色会频繁地变换。在玩家控制多个角色(一队或一组)的游戏中,玩家角色和非玩家角色之间的转换可以非常流畅,以至于它们的界限已经变得模糊。也就是说,非玩家角色在接受玩家控制指令前,一直在自主地行动。执行完玩家控制指令后,他们又回到自主行动的状态。

在游戏中,非玩家角色也充当摄影师、灯光师、解说员,甚至导演。当然,通常游戏场景中并不实际存在手持摄像机、移动照明灯、提供解说,或指导其他人的人物角色。但是,用相应的程序代码可以实现诸如:控制摄像机、照明、解说、导演等功能。将这些代码想象成非玩家角色是很直观的,因为它们同屏幕上呈现出的非玩家角色拥有很多相同的属性。因此,本书中讨论的大部分技术不仅适用于台前的“非玩家角色”,也适用于幕后的“非玩家角色”。

1.2 角色行为控制

游戏中的每一个角色至少有一个与它关联的“控制器”(Controller),而且控制器可以在不同的角色间共享。控制器如同角色的大脑,其输入是游戏世界的状态信息,其输出是影响游戏世界并导致非玩家角色相应行为的动作选择^①。

在其他文献中,术语“控制器”有时用来指玩家的输入设备。而在本书中,“控制器”专指角色的大脑,“游戏操纵杆”(Joystick)指玩家的输入设备。

玩家角色的控制器包含解释玩家操纵杆各种操作的机制。当然,玩家的大脑也是玩家角色控制器的一部分,因为它是决定角色如何行动的意图来源。

非玩家角色的控制器可以表现为多种形式,并具有不同的功能,这些形式和功能将是本书重点描述的内容。

^① 在本书中,我们将经常把“NPC的控制器”简称为“NPC”。例如,“NPC选择跳跃行为”应被理解为“NPC的控制器选择跳跃行为”。

大部分玩家并不关心是何种内在机制产生了非玩家角色的外表、运动以及行为，他们仅仅关心最终的结果。人们关注非玩家角色的行为，在于考虑非玩家角色行为是否表现“合理”。这里所定义的“合理性”是指：非玩家角色的行为是否与其目的相吻合，而与其目的本身的合理性无关。从更大的视角看，非玩家角色的目的有可能是不合理的。例如，对一个非玩家角色而言，从隐藏的地方跑出来，攻击刚刚消灭了所有同伴的强大玩家角色，它的行为就不太合理。

非玩家角色的最终目的是为了玩家取得娱乐效果，但这个目的一般是无法直接写入非玩家角色的控制器程序中的。因而，非玩家角色的控制器大多被设定去执行一些简单的任务。例如，试图不惜一切代价去攻击玩家角色。游戏设计者为非玩家角色设定这些任务，目的是使玩家玩得痛快。

判定什么是“合理的”行为，这个问题依赖于游戏本身，包括：游戏的困难级别、游戏环境，乃至玩家本人的期望。但也有一些常见的、易于识别的不合理行为是由控制器的程序错误引起的。比如，两个非玩家角色都想同时上一个梯子，结果就不断地互相冲撞；或者，非玩家角色成一列纵队冲过一个战场，以至于轻易地就被一个机枪手消灭了。

目前，期望非玩家角色具备同人那样的智能在实现技术上还存在许多限制。但是，游戏开发者有很大的选择余地，以确定控制器应该具备哪些智能去实现非玩家角色的目标行为。两个看上去相似的行为可以通过不同的控制方式实现。让我们考虑这样一个问题：“怎样使非玩家角色从一个地方移动到另一个地方？”这就是所谓的“路径规划”（Path Planning）问题。一种实现路径规划的方法是让控制器包含一个地图以及一个路径规划算法。这种方法将在第6章中详细描述。另一种方法是让游戏“层次设计师”（Level Designer）对游戏世界中的所有路径用“路标”（Signpost）标注。这些路标对玩家是隐蔽的，但非玩家角色却能够使用它们。一个非玩家角色想要到达某地，它的控制器只需追随路标即可。从玩家的角度看，非玩家角色在这两种控制方式下的行为大体是相似的。

在地图中加路标（取决于地图的大小和复杂程度）可能是一个费力、耗时、易错的工作。此外，如果地图发生改变，或有新地图，部分或全部的地图标注就要重新生成。与之相比，路径规划方法在不同的地图上都适用，它甚至能够运行

在由玩家创建的地图上。然而，在简便性及行为的控制精度方面，路标方式具备一定优势。而且，它对硬件和编程技巧的要求也显著降低。因而，两种方法都是可用的，各有各的优缺点。由此，读者可以想象，关于非玩家角色究竟需要哪些智能的问题，人们经常是各执己见。

一般地，非玩家角色的控制器究竟必须、应当、可能有哪些智能是由具体情况来决定的。游戏设计时要考虑硬件平台的性能、可获得的软件、技术水平、对技术的熟悉程度以及个人偏好。

1.3 游戏系统结构

图1.1给出了一个典型游戏软件的主要模块及其相互关系的系统结构图。这个系统结构只是一种可能的软件架构。但其他的架构将拥有大致相同的模块。每个模块的作用可简要叙述如下。

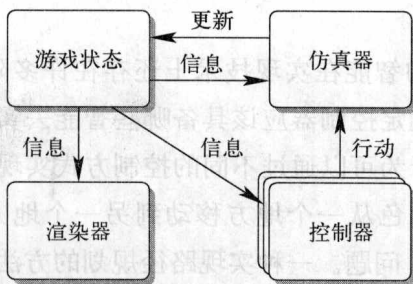


图1.1 体系结构

游戏状态 (Game-state): 游戏状态代表当前游戏世界的状态，它描述游戏世界中所有物体及其状态信息。其他模块可以通过“游戏状态”模块查询游戏世界中任何物体的当前状态。

仿真器 (Simulator): 仿真器制定游戏状态如何发生改变的规则，即“游戏物理学” (Game Physics)。具体说，

游戏角色控制器所选择的所有行动都是通过仿真器调动相应的动画实现的。

渲染器 (Renderer): 渲染器利用游戏中物体的几何关系及纹理提供对游戏状态的描绘，其输出结果通常包括图形和音响。

控制器 (Controllers): 每个游戏角色至少有一个相关的控制器，控制器负责选择角色的行动。对于玩家角色而言，控制器解释玩家游戏杆的按键。对于非玩家角色，控制器就是它的“大脑”，即它的“智能”所在。

非玩家角色需要在游戏中有具体表现才能通过其行动选择对游戏世界产生影响。因此，第2章将描述游戏状态、仿真器和控制器如何相互配合，以实现非玩家角色在游戏世界中的行为表现。

在游戏中，非玩家角色除了具备“行动”这种基本功能，还应具备“感知”世界的的能力。否则，其相应的控制器无法得知哪个行动是合适的、什么行动会产生什么效果。缺乏感知能力将使游戏角色茫然不知所措。对玩家角色而言，玩家可以直接观看渲染器，来获得关于游戏世界状态变化的各种反馈信息。非玩家角色则可以通过对游戏状态的查询，找到它们需要的信息。但是，从“人工智能”的角度看，如果能将游戏状态中不重要的细节简化或删除，编写控制器程序就会容易很多。另外，游戏状态所代表的信息越精炼，越能突出不同游戏状态之间的主要相似之处，控制器的编程就越简便、通用。因此，第3章将描述常用、有效的游戏状态信息转换，并解释如何通过游戏状态信息的选取，使非玩家角色的感知能力与一般玩家的预期相符。

当非玩家角色能感知它的周围世界，它就可以做出相应的行动决策，这些决策取决于它所感知的信息。因此，第4章将介绍如何设计无记忆的“反应式控制器”（Reactive Controllers）。这一类控制器所进行的行动选择仅仅建立在对当前游戏世界状态感知的基础上。

反应式控制器能够用来创建很多有效的行为。但是，如果没有记忆能力，控制器就无法知道它以前有过的类似经历，从而可能会造成行为的循环。因此，第5章将介绍如何给非玩家角色控制器加入内部状态，以便记住以前的经历。这样，非玩家角色的决策不仅可以基于当前游戏世界的状态，而且可以基于它过去的经历。理论上，具有记忆的非玩家角色控制器可以实现任何可计算的行为^①。

一种简单的控制器设计方法是直接遵循一组预先编制的行动规则，这些规则规定非玩家角色怎样达到它的行为目标。但是，在第6章讨论了另一种设计方法。如果控制器被给予明确的行为目标，它可以自动搜索能成功达到给定目标的行动序列。为了实现自动搜索，控制器需要可预测任意行动序列结果的世界模型，这可通过控制器对仿真器的访问来获得。通常，控制器需要的只是粗略的世界模型，

^① 这是因为它具备图灵完备性，也就是说，在理论上它与图灵机等价。而图灵机则是用来定义可计算性的数学模型。

这样搜索速度也会更快。虽然搜索是一种高效能的技术，但是，不可低估它耗费的CPU运算时间和内存资源。

除了简单的记忆能力，我们还可以使非玩家角色的控制器具备学习能力，从过去经历中增长见识。这种学习能力意味着控制器能将过去的经验推广到新环境，用于新的事件的处理。这完全不同于简单地存储然后不断反复重演过去的经历。控制器的“自学习”可以有很多方式。第7章集中地介绍控制器如何通过“反复试错”的学习方式自动生成所需要的各种知识。

读者阅读完这本书时，应能识别在所玩的计算机游戏中哪些技术产生了非玩家角色的行为；也应能够判断在所创作的游戏非玩家角色应具备哪些适用的智能。利用本书的文献以及本书网站上可获得的资源，读者应能根据自己的判断来设计并实现相应的非玩家角色控制器。

还有许多非玩家角色控制器可能需要具备的智能在本书中没有涉及。例如，给非玩家角色“装备”一个摄像头，摄取玩家所在环境的影像，我们就可能赋予角色“看”的能力。在游戏中，利用麦克风，使非玩家角色具有理解和产生语音的能力也越来越重要。虽然在目前游戏中利用摄像头和麦克风的范围尚小，但是有不断增长的趋势，见[Mar01, LDG03]。

1.4 人工智能

许多非玩家角色控制器潜在能力的开发来源于“人工智能”(Artificial Intelligence)领域的学术成果。最早的人工智能研究可以追溯到20世纪40年代。一些早期的成功导致人们过于乐观，认为计算机很快就会变得像人一样智能。但是，科学家们很快意识到，许多人工智能问题远比他们初始时的想象要困难得多。20世纪60年代，专家系统的出现为人工智能带来了基于专业知识的应用技术。专家系统的设计思想是将人类专家的知识编码存储于计算机中的知识库，利用知识库进行知识推理，计算机就能够回答和解决问题。

虽然专家系统在学术和商业领域取得了不少成就，但是，它的严重困难在于