

计算机游戏中的智能技术

李睿凡 左申正 李卫

北京邮电大学信息工程学院 310 信箱 100876

E-mail: liruifan@tom.com zuoshenzheng@163.com

摘要: 计算机游戏是计算机应用的一个热点领域。而人工智能技术则是被认为是计算机图形学在计算机游戏中发展到极致后极具潜在推动力的一个技术。传统认为游戏 AI 与学术 AI 有较大差别。但近来的发展表明, 学术人工智能技术将为游戏产业的发展注入新的活力。本文从游戏设计元素出发, 讨论它们与人工智能之间的关系。然后介绍游戏中相对成熟技术, 如 A* 算法、有限状态机、群体寻径, 再讨论近来游戏开发者关注的问题, 如适应与学习。

关键词: 人工智能 有限状态机 群体智能 神经网络 遗传算法

An Overview of Artificial Intelligence in Computer Games

Ruifan LI Shenzheng ZUO Wei LI

School of Information, Beijing University of Posts and Telecommunications, mailbox 310, 100876

E-mail: Liruifan@tom.com Zuoshenzheng@tom.com liw@nlu.caai.cn

Abstract: Computer game is an active field in computer science. Compared with computer graphics, artificial intelligence (AI) is taken as potential locomotion for computer games. In general, academic AI is thought as different from game AI. However, recent progress has shown that academic AI will spark new elements in computer games. In this paper, we discussed the relationship between elements in computer games and artificial intelligence. Then some techniques, including A* algorithm, finite state machine, path-finding based on swarm, and some problems attracting game-developer's interest, such as adaptivity and learning are discussed.

Key Words: Artificial Intelligence Finite State Machine Swarm Intelligence Neural Network Genetic Algorithms

1. 引言

游戏具有相当长的发展史。广义上, 可以认为游戏的产生是伴随着人类的产生、发展而不断进步的。古人类之间的嬉戏就是一种朴素的游戏活动。据史料记载, 早在公元前 4000 年时, 上古的尧发明了一种流传至今的游戏——围棋[1]。而计算机游戏是现代产物, 它是在电子技术、信息技术的发展中诞生的。1950 年, 信息论之父 Shannon 就提出用

里。正如在物理学发展史上出现的“两朵乌云”一样, 计算机游戏也正面临一次新的考验。也就是, 业界普遍认为, 在图形技术给游戏玩家带来的感官极致之后, 人工智能技术将取而代之成为下一个推动整个游戏产业的支柱技术[5, 6]。

2. 游戏元素与人工智能

目前, 市面上的游戏不计其数, 种类繁多, 形式多样。但考察不同的游戏可以发现它们具有一般

论文发表在: 中国人工智能学会 2007 年全国学术年会 (CAAI'12)

计算机编程方式实现人机对弈的构想, 并发起了棋牌游戏的研究[2]。这是计算机游戏的发端。1984 年, Cris Crawford 出版了名为《The Art of Computer Game Design》的第一本关于计算机游戏的书籍[3], 标志着计算机游戏时代的来临, 是计算机游戏史上的一个重要里程碑。

目前, 在芯片技术、网络技术和图形技术的催化下, 计算机游戏已经在全球范围内发展成为一个具有每年数十亿美元的产业。据《2003 年度中国游戏产业报告》, 中国网络游戏的市场预计在 2007 年将达到 67 亿元人民币[4]。计算机游戏已成为一种普遍的娱乐方式。但是, 游戏的发展也并非晴空万

化的形式元素。Crawford 的建议如下[3]: 1) 表现: 游戏是设计者根据现实而主观抽象出的一个虚拟世界; 2) 交互: 游戏是玩家与计算机互动的过程; 3) 冲突: 玩家为完成游戏设定的目标而努力, 同时也受到来自游戏设计的阻力; 4) 安全: 危险源于游戏冲突所产生的心灵体验; 游戏是体验现实世界的一种安全方式。

这些形式元素表达了游戏的特点, 当然也可以有其他方式表现游戏的特征, 它们都是殊途同归的。形式元素是设计游戏时候必须考虑的。不论游戏的形式如何, 本质上是人参与的一种活动, 其最终目的是为人娱乐, 或者是教育目的。

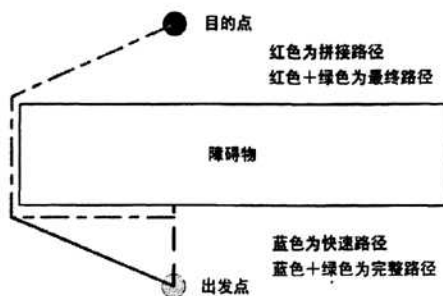


图1 三段路径体系

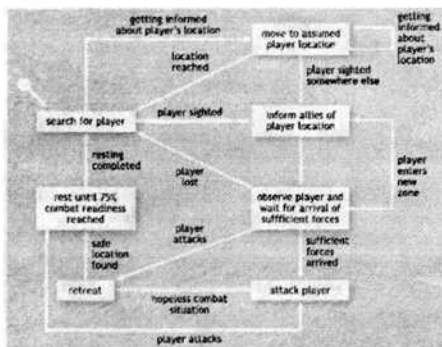


图2 游戏中有限状态机实例[5]

关于游戏的娱乐目的是显然的，而游戏的教育目的可能有些迷惑性。事实上，除了棋牌类游戏之外，游戏的教育功能可能是间接的、潜移默化的。比如，日本光荣公司出品的“大航海时代”系列游戏[7]就是一款以16世纪的航海为背景的寓教于乐的好游戏。

2.1 游戏 AI 与学术 AI

Crawford 认为计算机游戏是可以与电影并列的一种新的艺术创作形式，这种观点也逐渐为认可。而游戏中人工智能的水平很大程度上决定了游戏的是否能够被玩家和市场所接受。

学术人工智能的一个普遍定义如下[8, 9]：即一种能有效地通过图灵测试的系统。这意味着被测试的计算机能够就相同问题提供与人同样的答案。而游戏开发者则将用于控制非玩家角色的代码认为是人工智能，即游戏 AI[10]。它确定了游戏对于玩家输入动作的反应。这种反应可能是符合逻辑的，也可能是随机的，甚至是人为的愚蠢，但它们都是游戏 AI。造成游戏 AI 与学术 AI 差异的根本原因在于前者是为游戏的根本目的——娱乐服务的，而后者目的在于探讨智能本身。

游戏 AI 的设计与实现是游戏设计中的一个重要环节。事实上，如果没有游戏 AI，游戏也就失去对抗性而不会成为令人愉悦的产品。游戏 AI 的工作包含设计与实现两个过程：首先依据游戏本身的主题和目标确定游戏 AI 完成什么工作；然后针对完成的工作确定实现的算法、程序。

不同的游戏类型所要求的 AI 设计大相径庭。但可以大体将游戏 AI 的目标划分如下[10]：1)

挑战玩家：给玩家提供恰到好处的挑战是游戏 AI 设计的最基本目标；2) 建立基本真实感：通俗地说，就是避免一些显而易见的错误，如避免 NPC 绕树行走的明显错误；3) 增加不确定性：令玩家无法准确预测游戏中 NPC 的行为；4) 辅助情节的展开：让游戏 AI 变化游戏故事情节的发展；5) 创造逼真的世界：游戏世界虽然是虚拟世界，但某些情节上还需要相对的真实性，如天空中翱翔

的飞鸟，草地上正在吃草的羊群。这些在一定程度上可以看成是游戏的装饰物。

即使学术 AI 与游戏 AI 有着明显的差异，但仍可以看到学术 AI 的某些思想、算法是可以用于游戏 AI 的设计。同时，仍然需要注意的是，游戏 AI 不是学术 AI 的测试对象而在于增强玩家体验。但长期以来，人工智能技术在游戏产业与学术界的发展是互相独立的。一方面，学术界认为游戏只是人工智能技术的一个应用；而游戏产业界则认为学术界的理论知识一厢情愿的想法。有幸的是，人们已经注意到 Game AI 与学术 AI 之间的差别。

3. 一般的游戏 AI 技术

上节中已经谈及游戏 AI 与学术 AI 的差异，这种情形在 2001 年以前尤为突出。Woodcock 在 2001 年的游戏开发者大会 (Conference on Game Developer) 上指出工程应用中的简单化原则 (KISS, i.e. Keep It Simple and Stupid) 体现在游戏 AI 中则是应用基于规则的系统与模糊状态机处理几乎所有的问题[11]。可见，它们在游戏编程中的地位。

3.1 走步：定向与寻路算法

在游戏中，通常需要让 NPC 能够聪明地从地图上某点移动到另一点。这里的“聪明”，基本的含义是能够避开障碍物，能都找到最近的路。如果至少存在一条路径，在不同的路径搜索算法中，A* 算法能够找到最短路径，而且算法速度快。下面，简要介绍基本 A* 算法[12]。

首先为每个节点定义 g , h 和 f 三个属性：(1) g (goal) 是从起始节点到该节点的代价。虽然从起始节点到该节点的位置有很多不同路径，但该节点仅表示一条单一路径。(2) h (heuristic) 是该节点到目标节点估计代价。(3) f (fitness) 是 g 与 h 之和。它表示对该路径的最佳估计，其中的 f 值越小则认为该路径越好。另外，A* 算法维护两个表：Open 表和 Closed 表。Open 表由未考察的节点组成，而 Closed 表由已考察的节点组成。当算法已经检查过

与某个节点相连的所有节点, 计算出它们的 g , h 和 f , 并把它们放入 Open 表, 以待考察, 则称这个节点为“已考察的”。

如何高效地运用 A*算法关键在于代价 g 与启发 h 的计算函数。问题的关键在不同的情境下, A*算法的差异也将发生变化。比如, 地形分析(Terrain analysis), 瓶颈探测等问题。对于 A*算法, 在不同的情形下有不同的变种。主要包括 A*算法对不同问题的通用性、负载优化、速度和美学优化等。针对 A*算法的通用性问题, Dan Higgins 提出了通用 A*路径搜索引擎的设计[13], 命名为 A*机, 将存储、目标、地图、引擎作为一个通过构架, 完成算法的重用设计。对于多个 NPC (上百, 甚至更多) 同时寻找路径给计算机中央处理器带来过载而可能死机的问题, Dan Higgins 提出了路径搜索的体系结构的方法[14]。文中采用了三段路径体系: 快速路经、完整路径、拼接路径, 并通过队列路径管理器完成路径的选择与更新。其思路的示意如图 1 所示。

在实现速度上, A*算法优化强调如何高效地存储及搜索 Open 表和 Closed 表[15, 16]。文献另外, 文献[17]考虑了如何约束 A*算法, 约束算法何时搜索、如何搜索、以及充分运用局部路径, 从而显著降低搜索开销的方法。在美学优化上, 文献[18]主要考虑了使得路径更笔直、更平滑、更直接的问题, 这在复杂环境下尤为突出。对于笔直路径问题, 主要的策略有两种: 其一是促进 A*算法中笔直路径的构造; 二是在路径计算完毕后再重新清理。路径平滑则采用 Catmull-Rom 样条算法。对于分级寻径(Hierarchical path-finding)导致的路径破坏, 则可以采用改进的分级寻径方法。除 A*算法之外, 为适应手机等处理能力不足的情形, 也存在不同的替代算法。文献[19]提出了一种基于加权节点的低耗、高速的路径搜索算法。

3.2 有限状态机

有限状态机(Finite State Machine, i.e. FSM)也是计算机游戏中应用普遍的人工智能模型。通常, 这类系统描述了一个事件驱动系统模型, 易于规则表达。一般的, 状态机可以用状态转移图的方式描绘。其中的节点表示不同的状态, 不同状态之间由于触发条件而发生转换, 用弧线表示。状态转换的条件写在弧线上方, 旁边是当转换发生时的可能输出。图 2 为有限状态机的实例。

有限状态机在游戏中可以构成管理游戏世界(Game world)的基础, 模拟 NPC 的情绪状态[6], 维持游戏的状态, 分析玩家的输入, 或者管理对象的状态。

就有限状态机的实现上, 文献[20]给出了一个有限状态机的 C++类实现。此外, 由于状态机的滥用, 可能导致程序可读性差, 调试困难, 性能下降。针对这个问题, Steve Rabin[21]提出一种以简单语

言来组织有限状态机的鲁棒方法。继而, [22]将该方法拓展, 用消息机制考虑了人工智能对象间通信的问题, 包括状态机之间的切换、同步等。

作为一般有限状态机的扩展, 模糊状态机(Fuzzy State Machine, i.e. FuSM)也是较为常用的模型[23]。这种模型可以实现不可预测的行为, 即游戏每次都有可能产生不同的行为结果, 从而为玩家提供更多乐趣, 也具有更大的重玩可能性。这也是智能行为的一种表现。

模糊状态机的典型运用包含 NPC 或玩家角色的生命值定义, 赛车游戏中汽车的加速与制动[23], NPC 情绪的模糊状态等。文献[24]给出了一个通用的 C++模糊状态机设计。与一般状态机类似, 模糊状态机也存在模糊状态增加而导致的组合爆炸。文献[25]提出了 Combs 方法用于解决组合爆炸问题。传统系统中的规则通常考虑输入集合的组合同输出之间的关系, 而 Combs 则考虑各个集合同输出之间的关系, 每次为一个变量创建规则。

3.3 群体智能的模拟

群智能(Swarm Intelligence)是计算智能的一个研究热点。其典型算法包括: 粒子群算法(Particle Swarm Optimization, i.e. PSO)[26]、蚁群算法(Ant Colony Optimization, i.e. ACO)[27]。而粒子群算法来源于 Reynolds 所谓的 flocking 的生物群体行为, 如鸟群的飞翔、鱼群的游弋的模拟。Reynolds[28]提出的 Boids 模型用于模仿群体的定向行为(Steering Behaviors)。该模型认为, 在群体的个体之间保持统一定向行为存在三个规则: (1) Separation(分离): 避免临近个体间的碰撞; (2) Alignment(对齐): 匹配临近同伴间的平均速度方向; (3) Cohesion(内聚): 保持与临近同伴间的相对稳定。

如果定义 Boids 群体行为的期望速度为 \mathbf{v}_d , 同时表达对齐、内聚与分离分别为 \mathbf{v}_a , \mathbf{v}_c , \mathbf{v}_s , 则期望速度 \mathbf{v}_d 是三个有权值分量之和: $\mathbf{v}_d = w_0 \hat{\mathbf{v}}_a + w_1 \hat{\mathbf{v}}_c + w_2 \hat{\mathbf{v}}_s$ 。通过不断计算个体的速度, 可以获得整个 Boids 群体的动态行为。具体的, 个体在每一次的更新中重新计算, 而并不存储其它信息。这令群体表现出了突现行为(emergent behaviors)。即, 群体中个体成员无法预知下一步的动向, 而群体则以动态方式保持步调一致。文献[29]用 C++语言实现了 flocking 类。Reynolds 的群体算法的缺点在于计算开销大, 这在个体数目的情形下更为突出。为此, 文献[30]提出了一种简化策略用于大规模的实时操作。

3.4 学习与适应性

学习与适应是智能的重要特点。而近年来, 智能游戏中也有采用学习与适应技术的趋势[31]。其

有两点较为明显的、宏观上的好处：一是如果“真正”具有不断学习与适应的人工智能，这就使得玩家需要不断尝试新的办法以对付不断改变的游戏程序；二是让游戏设计者从预先估计玩家的策略中解脱出来。但是，就目前而言，采用学习、适应机制的游戏还不多，即使使用，其水平也不高。

大体上，可以将游戏开发中采用的学习和适用方法分成两类。第一类方法称为间接方法。即通过提取游戏运行中的统计数据，控制 NPC 的行为。而所采用的数据和数据与行为间的关系由游戏设计者事先确定。第二类方法称为直接方法。它将学习算法应用于 NPC 的行为本身，由游戏开发者指定需要调整行为的哪些方面。实际中，通常用某种方法将 NPC 的行为参数化，然后使用优化算法寻找合适的参数，从而获得良好的性能。

神经网络(Neural Network, i.e. NN)是人工智能中典型的可用于学习的模型。有关神经网络的详尽描述见[32]。文献[33]给出了一个在游戏中使用多层感知机(Multi-Layer Perception, i.e. MLP)的例子。问题是如何设置坦克炮筒的倾斜角，以命中敌方坦克。文章采用坦克间的距离、风速大小和方向作为 3 层 MLP 的输入。网络训练不是采用通常的梯度下降法，而是采用了称为扰动搜索(Perturbation Search)的方法。该方法具有易于编码、确保稳定的优点。具体细节见原文。另一个例子采用多层感知机模拟 NPC 的随机变化的方法文献[34]。在上面的坦克例子中，尽管神经网络可以模拟游戏者的行为，但只能重现游戏者行为的确切方面，而不能重现它的随机变化。文章将差错的范围分割成互不重叠的区间，称为 bin；而用多层感知机计算概率如何随分布的输入变化而变化。

遗传算法(Genetic Algorithm, i.e. GA)是随机优化的方法，导引性的著作可参考[35]。文献[36]给出如何将遗传算法用于游戏编程的例子。例子描述如何进化出具有偷羊本领的洞穴人。通常而言，在处理很多的变量、复杂的环境交互的情形下，可以通过模拟自然界适者生存的进化过程，可以探索出不同寻常的融合策略，在游戏中将有助于设计出有趣的 NPC。

N-gram 模型通常被用于语言建模。简言之，该问题就是在—篇文本中猜测下一个字母，而用马尔科夫(Markov)表达。文献[37]将 N 元文法(N-gram)模型用于预测玩家行为。其基本假设在于玩家的行为与特征具有局部结构。其局部结构包括：(1)游戏的限制和约束：游戏中的某些动作只有在特定的情形下才是合法的；(2)玩家风格：如在模拟飞行与格斗游戏中，玩家通常有其特有的操作技术与动作序列；(3)激励：玩家有时在完成一系列动作后会受到奖励；(4)游戏控制：通过按键控制游戏很容易得到动作序列。通过训练，N-gram 模型可以识别这些结构化的元素，能够控制 NPC 的行为。

不同的学习方法都与其优势与不足。所以需要

将不同的方法融合而构建更为合理的智能行为。游戏 Black & White[38]中的怪兽采用了不同的学习方法。文献[39]提出了混合策略下学习的思路。其中的学习技巧在于：学习事事情况；学习哪些愿望具有支配性，以及不同愿望下的敏感度；学习如何评价最大程度地满足智能愿望的方法。学习过程包含三种方式：学习过程可以发生在玩家给出反馈之后；学习过程可以发生在观察其他 NPC 的行为之后；学习过程可以发生在给出命令后。其设计框架是基于 Multi-Agent System 的信念-愿望-意图结构(Belief Desire Intension, i.e. BDI)。信念是一种数据结构，它存储了单个对象的信息；愿望是智能体(Agent)追求的目标；智能体使用愿望、信念和判断产生行为意图。在此基础上，提出增加判断(Opinion)结构。文章中饥饿愿望采用感知器模拟，表达同情愿望采用决策树模拟。

4. 前景与展望

本文描述了当前计算机游戏中人工智能技术应用的现状，主要针对 NPC 行为而忽略附加功能，如自然语言处理、语音识别等应用，更为深入的讨论已经超出了本文的所及。关于游戏人工智能技术的最新书籍因条件所限而未能涉猎[53-57]。

计算机游戏中人工智能无疑将是取代图形学而成为下一个游戏产业的强大推动力。这为人工智能的研究与应用提供了更为广阔的舞台。人工智能的研究成果已经运用到游戏开发的实践之中。学术界与产业界不断达成共识，涌现出对于人工智能的更大兴趣。可以预见，游戏中人工智能将成为游戏的大卖点，学术人工智能人员也将创造出与人匹敌的博弈系统。

参考文献

- [1] 何云波, *围棋与中国文化*. 北京: 人民出版社, 2001.
- [2] C. E. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, vol. 314, pp. 256-275, 1950.
- [3] C. Crawford, *The Art of Computer Game Design*. Berkeley, CA: McGraw-Hill, 1984.
- [4] <http://weiqi.ourgame.com/info/data/history/wqds.html>, "2003 年度中国游戏产业报告."
- [5] A. Nareyek, "AI in Computer Games," *ACM Queue*, vol. 1, pp. 58-65, 2004.
- [6] D. J. Funge, *Artificial Intelligence for Computer Games*. Washington: AK Peters, Ltd., 2004.
- [7] <http://koukai.ewewind.com/index.htm>, "大航海时代中文网."
- [8] R. Rouse, *Game Design: Theory and Practice*. Texas: Wordware Publishing, Inc, 2001.

- [9] W. Steven, "Game AI: The state of the Industry 2000-2001: It's Not Just Art, It's Engineering," in *Game Developer Magazine*, 2001.
- [10] B. Stout, "The Basics of A* for Path Planning," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 254-263.
- [11] D. Higgins, "Generic A* Pathfinding," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 114-121.
- [12] D. Higgins, "Pathfinding Design Architecture," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 122-132.
- [13] S. Rabin, "A* Speed Optimization," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 272-287.
- [14] D. Higgins, "How to Achieve Lightning-Fast A*," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 133-145.
- [15] T. Cain, "Practical Optimization for A* Path Generation," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 146-152.
- [16] S. Rabin, "A* Aesthetic Optimization," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 264-271.
- [17] M. Mika and C. Charla, "Simple, Cheap Pathfinding," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 155-160.
- [18] E. Dybsand, "A Finite-State Machine Class," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 237-248.
- [19] S. Rabin, "Implementing a State Machine Language," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 314-320.
- [20] S. Rabin, "Enhancing a State Machine Language through Messaging," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 321-330.
- [21] M. McCuskey, "Fuzzy Logic for Video Games," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 319-329.
- [22] E. Dybsand, "A Generic Fuzzy State Machine in C++," in *Game Programming Gems 2*, M. DeLoura, Ed.: Charles River Media, Inc., 2001, pp. 337-341.
- [23] M. Zarozinski, "Explosion in a Fuzzy System," in *Game Programming Gems 2*, M. DeLoura, Ed.: Charles River Media, Inc., 2001, pp. 342-350.
- [24] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics," *IEEE Transaction on Antennas and Propagation*, vol. 52, pp. 397-407, 2004.
- [25] M. Dorigo, V. Maniezzo, and A. Colomi, "The Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Transactions on Systems Man and Cybernetics - Part B*, vol. 26, pp. 29-41, 1996.
- [26] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, vol. 4, pp. 25-34, 1987.
- [27] S. Woodcock, "Flocking: A Simple Technique for Simulating Group Behavior," in *Game Programming Gems 1*, M. Deloura, Ed.: Charles River Media, 2000, pp. 305-318.
- [28] T. Scutt, "Simple Swarms as an Alternative to Flocking," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 202-208.
- [29] J. Manslow, "Learning and Adaptation," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 557-566.
- [30] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Beijing: Tsinghua University Press, 2001.
- [31] J. Manslow, "Using a Neural Network in a Game: A Concrete Example," in *Game Programming Gems 2*, M. DeLoura, Ed.: Charles River Media, Inc., 2001, pp. 351-357.
- [32] J. Manslow, "Imitating Random Variations in Behaviors Using a Neural Network," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 624-628.
- [33] M. Mitchell, *An Introduction to Genetic Algorithms* MIT Press, 1996.
- [34] F. D. Laramée, "Genetic Algorithms: Evolving the Perfect Troll," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 629-639.
- [35] F. D. Laramée, "Genetic Algorithms: Evolving the Perfect Troll," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 596-601.
- [36] <http://www.lionhead.com/bw2/>, "Black & White Official Web Site."
- [37] R. Evans, "Varieties of Learning," in *AI Game Programming Wisdom 1*, S. Rabin, Ed.: Charles River Media, 2002, pp. 567-578.
- [38] A. Watt and F. Policarpo, *3D Games Real-time Rendering and Software Technology, Vol 1* (中译本). 北京: ACM

Press (机械工业出版社), 2001 (2005).

- [39] J. D. Funge, *AI for Games and Animation: A Cognitive Modeling Approach*: A. K. Peters, Ltd., 1999.
- [40] S.-J. Yen, J. Yen, and S.-C. Hsu, "Move Strategies in Middle Game of Computer Go," presented at 7th TAAL, Taiwan, 2002.
- [41] S. Rabin, *AI Game Programming Wisdom 2*: Thomson Learning, 2005.
- [42] A. Kirmse, *Game Programming Gems 4*: Charles River Media, Inc., 2004.
- [43] K. Pallister, *Game Programming Gems 5*: Charles River Media, Inc., 2005.
- [44] H. J. Herik, H. Iida, and E. A. Heinz, *Advances in Computer Games: Many Games, Many Challenges*: Kluwer Academic Publishers, 2003.
- [45] H. J. Herik, H. Iida, and E. A. Heinz, *Advances in Computer Games*: Kluwer Academic Publishers, 2005.