

<<游戏编程中的人工智能技术>>

图书基本信息

书名：<<游戏编程中的人工智能技术>>

13位ISBN编号：9787302125990

10位ISBN编号：7302125996

出版时间：2006-5

出版时间：清华大学出版社

作者：布克兰德

页数：296

字数：447000

译者：吴祖增

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<游戏编程中的人工智能技术>>

前言

智能和人工智能. 本书介绍游戏编程中的人工智能, 简称“游戏人工智能”。所谓人工智能, 就是由人工建立的硬件或软件系统的智能, 是无生命系统的智能。例如, 一个机器人的智能, 一个能与人下棋的软件的智能。

智能是人类(大脑)智力活动的能力。

它的表现形式非常广泛, 例如, 能利用各种知觉系统接受来自环境的各种信息, 能形象思维, 能逻辑推理, 能理解事物之间的内在联系, 能作归纳和推广来发现规律; 能承认真理, 能使自己适应环境, 能主动学习, 能根据以往经验改进提高自己, 不至于犯同样的错误; 能进行创新思维来从事各种创新等。

智能是一个抽象的概念, 一个软件或硬件系统是否有智能, 只能根据它所表现出来的行为..

<<游戏编程中的人工智能技术>>

内容概要

本书是人工智能游戏编程的一本指南性读物，介绍在游戏开发中怎样应用遗传算法和人工神经网络来创建电脑游戏中所需要的人工智能。

书中包含了许多实用例子，所有例子的完整源码和可执行程序都能在随书附带的光盘上找到。

光盘中还有不少其他方面的游戏开发资料和一个赛车游戏演示软件。

本书讲解的原理通俗易懂，介绍程序详细周到，很适用于游戏编程者自学之用，也可以作为游戏培训教材使用（本书实际已被国内外许多游戏培训单位用作指定教材）。

对于任何希望了解遗传算法和人工神经网络等人工智能技术的各行业人员，特别是要实际动手做应用开发的人员，都是一本值得一读的好书。

<<游戏编程中的人工智能技术>>

作者简介

Mat Buckland在伦敦大学学完计算机科学后，做过多年的风险管理咨询顾问。终于他开始厌倦了所有的金钱游戏和公司制度，就一把火烧掉了他原有的公司礼服，跑到一家为Gremlin software开发游戏的公司工作。虽然薪水少得多，但是却很有趣，而且他可以每天穿牛仔裤去上班了!现在Mat同时是一个自由程序员和人工智能咨询顾问。自从20世纪80年代初第一次接触到这些技术，他就对进化计算和AI尤感兴趣。他是ai-iunkie.com网站（www.ai-junkie.com）的作者，该网站提供一些有关进化算法的教程和建议。

<<游戏编程中的人工智能技术>>

书籍目录

第1篇 Windows编程 第1章 Windows概述 1.1 历史一瞥 (A Little Bit of History) 1.2 Hello World! 1.3 第一个Windows程序 第2章 Windows编程进阶 2.1 Windows图形设备接口 2.2 文本 2.3 如何创建后备缓冲 2.4 使用资源 2.5 对话框 2.6 正确定时 2.7 总结第2篇 遗传算法 第3章 遗传算法入门 3.1 鸟和蜜蜂 3.2 二进制数速成 3.3 计算机内的进化 3.4 帮助Bob回家 3.5 练习 第4章 置换码与巡回销售员问题 4.1 巡回销售员问题 4.2 置换杂交操作 4.3 交换变异操作 4.4 选择一个适应性函数 4.5 选择 4.6 把一切组合在一起 4.7 总结 4.8 练习 第5章 遗传算法优化 5.1 TSP用的各种算子 5.2 各种处理工具 5.3 总结 5.4 练习 第6章 登月也不难 6.1 创建和处理矢量图形 6.2 矢量 6.3 相关的物理知识 6.4 人工控制的登月工程 6.5 遗传算法控制的登月飞船 6.6 总结 6.7 习题第3篇 神经网络 第7章 神经网络概述 7.1 神经网络介绍 7.2 一个生物学的神经网络——大脑 7.3 数字版的神经网络 7.4 扫雷机游戏 7.5 总结 7.6 练习 第8章 为机器人提供知觉 8.1 回避障碍物 8.2 为机器人提供记忆器 8.3 总结 8.4 练习 第9章 有监督的训练方法 9.1 异或函数 反向传播的工作原理 9.2 RecognizeIt——鼠标手势的识别 9.3 一些有用的技术和技巧 9.4 监督学习的应用 9.5 一个现代寓言 9.6 练习 第10章 实时演化 10.1 外星人游戏 10.2 练习 第11章 演化神经网络的拓扑 11.1 竞争约定问题 11.2 直接编码 11.3 间接编码 11.4 拓扑扩张的神经演化 11.5 总结 11.6 练习 附录A WEB资源 附录B 参考书目及推荐读物 附录C 光盘中的内容 后记

<<游戏编程中的人工智能技术>>

章节摘录

欢迎阅读《游戏编程中的人工智能技术》。

我想你会发现，这正是你所读到的有关游戏编程的最有用的图书之一。

Mat最先引起我的注意大概是在2000年的时候，那时，他开始在GameDev论坛(www.gamedev.net)发布各类关于游戏人工智能的帖子，并回答网友的各种相关问题。

他很快就获得了大家的关注，在跟贴中也不乏赞扬与认同，尤其是在贴出他为公众消遣而制作的有关神经网络和遗传算法的两个教程之后。

Mat发现游戏开发者需要获得AI技术方面的知识以期运用在游戏制作中，而他的两个教程以及在GameDev论坛里对提问的耐心答复无疑是满足这一需要的一个途径。

我对能为这样一个专题的书写前言而感到荣幸，希望以后能..

<<游戏编程中的人工智能技术>>

媒体关注与评论

引言 考虑一下有多少傻瓜能作微积分演算，要让任何其他傻瓜去掌握看来是更困难或更繁琐的微积分技巧就令人惊奇了。

有些微积分技巧是相当容易的，但有些则非常难。

那些编写了高等数学教课书的傻瓜——他们都是些最聪明的傻瓜——很少会劳神向你说明简单的微积分有多么简单。

相反地，他们采用了最艰难的方法来处理相关内容，好像就是刻意要让你牢记，他们那无比的智慧。

由于我是一个出名的傻家伙，我始终不学那些困难的技巧，我现在也要求为我的傻瓜读者提供那些不难的部分。

彻底地掌握这些以后，其余的也会随之而来（follow）。

我相信，一个傻瓜能干的，其他傻瓜也一定能干。

Silvanus P. Thompson，《简易微积分》导言，1910年初版 家用电脑从Sinclair ZX80演变到现在已经历了很长的一段路程。

硬件速度越来越快，而电脑组件的价格则越来越低。

在短短几年时间里，我们所看到的游戏的图形质量有了飞速的提高。

不过，到目前为止，这几乎就是游戏开发所有努力要解决的重点——提供更好的视觉享受，而在我们最喜爱的电脑对手的AI方面，却改进甚微。

然而，时代在改变。

计算机的硬件现已到了这样的转折点，它已能为游戏开发者创建AI提供更多的时钟周期。

同时，游戏的玩家们的品位也愈来愈复杂。

人们已经不再希望在游戏中看到诸如Doom和Quake这些曾经是最心爱的早期游戏中所看到的那种蠢笨怪兽了。

他们也不希望再看到计算机控制的游戏角色在那里盲目地蹒跚着寻找根本不存在的路径，不时地被卡在那些拐弯处，或是在没有资源的地方开发矿藏，傻头傻脑地撞在周围的树上。

游戏玩家想要从游戏中获得更多的乐趣。

他们希望从电脑生成的对手（或盟友）那里看到合情合理的、有智能的行为。

基于这些原因，我坚定地认为，在未来的几年里，AI技术将会大幅度起飞。

像Black & White 和 Halo这样的游戏已经让我们为其AI技术而倾心，游戏玩家们正期待更多的此类游戏。

此外，基于AI技术和人造生命的全新游戏流派也在最近几年里开始出现，如Steve Grand的Creatures，这个游戏的销售量超过100万份的事实让他本人和其他所有人都吃惊。

但如果你认为这是很多了，那么你再看看由Electronic Arts制作的Sims的销售，到目前为止，Sims及其服务器软件（add-on packs）的销售已经高达1300万份！

这是一个巨大的销售收入，而同时也是一个重要指示：玩家对这类技术有多么浓厚的兴趣。

这个趋势只可能继续。

创造智能幻影（illusion）的技术有多种，本书主要探讨两个方面：遗传算法和人工神经网络。

这两项技术讨论的地方很多，它们无疑正是当前的热点论题，但是被误解的地方也相当多。

就拿神经网络来说，经常会发现游戏开发者误认为神经网络是极其复杂的东西，因而会占用太多的处理器时间，导致游戏速度降低。

或者相反，他们可能过度地追求神经网络的能力，企图去创造一个有感觉的类似HAL的人工生命，其结果必然是以失败而告终。

我希望本书能帮助他们减少类似的曲解。

上面我引用Silvanus Thompson所著的受到人们喝彩的《简易微积分导言》中的那一段话可以成为本书的完美的开端（谢谢，Silvanus！

），因为，神经网络和遗传算法和微积分一样，对初学者来说，可能是非常难入门的，尤其对那些没

<<游戏编程中的人工智能技术>>

有接受过正规专业教育的人来说更是如此。

而已有的所有此类著作都是由学院式专家为学院式读者所写，其中充斥着（非本专业人）看不习惯的数学公式和难于理解的术语。

因此，我写了一本是我刚开始对这些课题感兴趣时所想要读到的那种书：一本由傻瓜写给傻瓜看的书。

相信我，如果我刚开始涉足该领域时能够得到这样一本书，我就用不着为搞清那些学究们到底是在讲什么而花那么多时间、受那么多挫折了！

经过这些年，我已经读了有关这个课题方面的很多书和论文，它们几乎都没有能给出真实的例子，没有东西可以让你牢靠地掌握并对自己说“啊！

这就是我能拿来运用的东西！

比如，关于遗传算法的书差不多总是向你提出这样的问题：最小化下面的函数 $f(x_1, \dots, x_5) = x_1 \sin x_1 + 1.7x_2 \sin x_1 - 1.5x_3 - 0.1x_4 \cos(x_4 + x_5 - x_1) + 0.2x_5^2 - x_2 - 1$ 其中 $-100 \leq x_1, \dots, x_5 \leq 100$ 我的意思是说，这是一个完全可以用遗传算法来解决的问题，但是上面的提法却让我们这些凡人实在难于理解其中的意义。

除非你有一个很好的数学基础，这样的问题很可能会显得过于抽象，会立即让你感觉到不舒服，对继续学下去也就没有趣味。

但如果给你的问题是这样：请允许我向你介绍Bob。

今天对Bob来说可不是什么好日子，他已深陷到一个迷宫中，而他的妻子正期待他能早点回家一起分享她花了整个下午制作的晚餐。

我来告诉你如何使用遗传算法帮助他找到迷宫出口，以挽救他的婚姻。

你的大脑就有一个可以与之联系的着落点（anchor point）。

你也立即会对这样的题目感到舒适。

不仅如此，这也是一道有趣的题目。

你一定很想知道我们怎样来帮他呢？

由此你就会打开书本，继续学下去，在学习中感受到乐趣。

这就是我在本书中用来阐明概念所使用的一种类型的题目。

如果我的做法正确有效，如何把你的想法运用到你的游戏和项目里就会变得显而易见。

对于我的读者，我只作一个假设，那就是读者了解如何编程。

我不了解会有些什么样的读者，但是有时，当我买了一本书而发现其中仅有一部分我不了解，以至于必须去买另一本书来解释第一本书里我所不懂的内容，就会让我感到失望。

为了防止这样的事情发生在读者身上，我努力确保这本书能解释代码所说明的一切——从Windows GDI、矩阵和矢量数学的使用，直到物理学和2D图像学。

我知道事情通常都有其两面性，很可能一些读者已经掌握了这些图形学、物理学和GDI的相关知识，那么，你们可以跳过这部分而只阅读那些你们更感兴趣的部分。

注：创建演示程序 演示程序是很容易编译生成的。

首先把你所需的源程序复制到你的硬盘里（译注：因编译一般需要用其他的一些文件，你在复制源文件时，应把源程序所在文件夹中所有的文件整个地从光盘复制到硬盘）。

如果你用微软的Visual Studio来编译，只要点击工程的workspace文件（译注：即以.ws为后缀的那个文件）就可以开始进行编译并最后生成可执行程序。

如果使用其他的编译器，你应该创建一个新的Win32工程（确保winmm.lib在你的工程设置里），并在按编译按钮之前，先在工程文件夹里加上相关的源程序和资源文件。

所有要做的就是这些。

用不到添加额外的路径、DirectX或OpenGL。

在所有的例子里，我尽量使代码编写得尽可能简单。

使用的是C++语言，但是我希望C语言程序员也能理解我的代码。

所以，出于这方面的考虑，我没有使用C++所固有的诸如继承、多态一类的东西（stuff）。

<<游戏编程中的人工智能技术>>

我使用了很简单的标准模板库（standard Template Library，sTL）功能，但是在我引用sTL功能的地方，会在边框里进行解释。

使用简单代码的一个出发点是不使我所要阐述的原理显得晦涩。

相信我，本书所涵盖的有些内容一开始并不容易掌握，所以我不想把因为在例子中使用了高级代码而把事情弄得更复杂。

我已尽我所能恪守老管理顾问的信条：K.I.S.S（Keep It Stupidly Simple），让一切尽可能地简单。

好了，不再罗嗦了，让我们来开始冒险吧.....

<<游戏编程中的人工智能技术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>