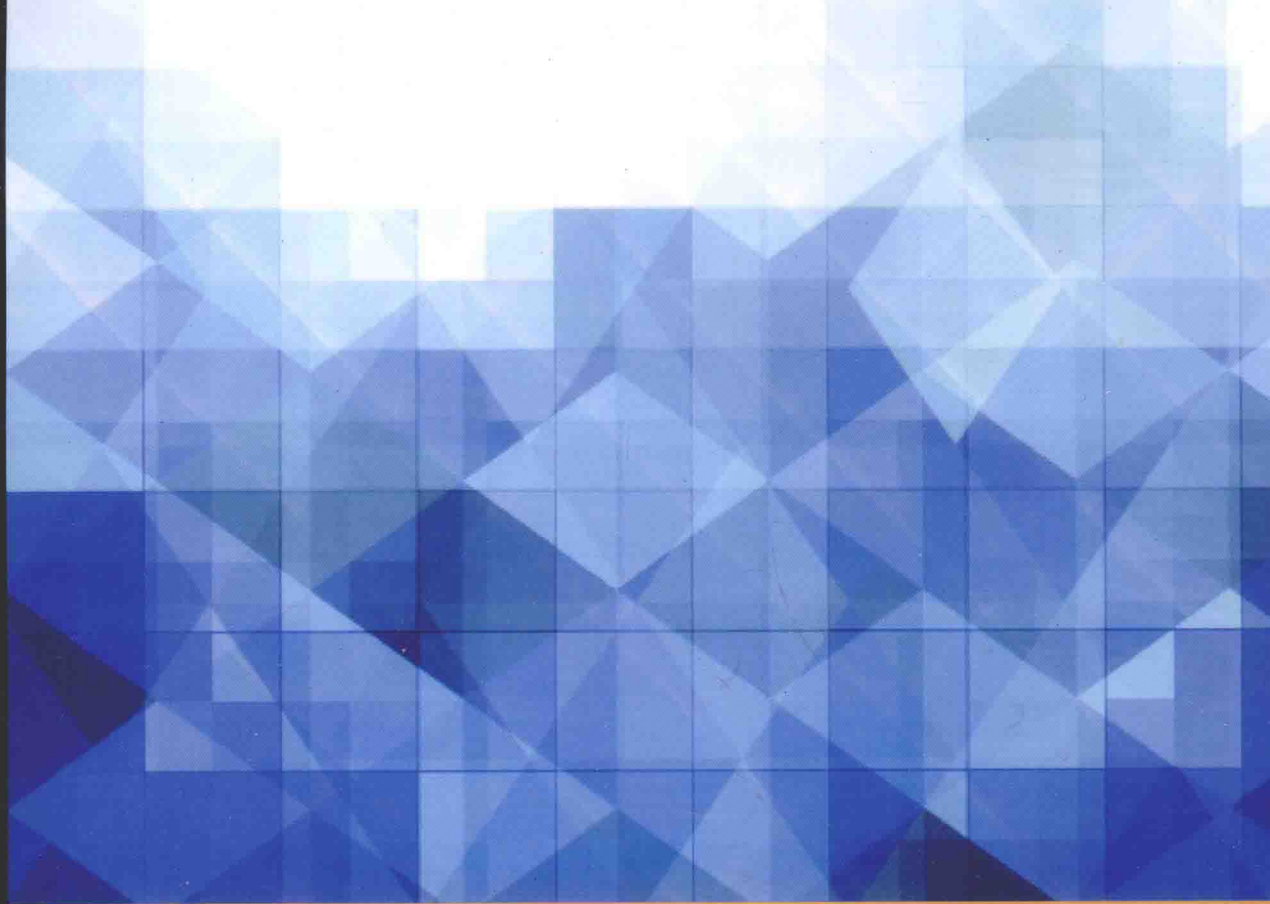

[美]Ray Barrera等 著 张颖 译

Unity人工智能游戏开发（第2版）

Unity AI Game Programming, Second Edition

清华大学出版社

Unity 人工智能游戏开发（第 2 版）

[美] Ray Barrera 等著

张 颖 译

清华大学出版社

北 京

内 容 简 介

本书详细阐述了与 Unity 游戏人工智能相关的基本解决方案，主要包括游戏 AI 的基础知识、有限状态机、实现感知系统、寻路方案、群集行为、行为树、模糊逻辑等内容。此外，本书还提供了相应的示例、代码，以帮助读者进一步理解相关方案的实现过程。

本书适合作为高等院校计算机及相关专业的教材和教学参考书，也可作为相关开发人员的自学教材和参考手册。

北京市版权局著作权合同登记号 图字：01-2016-5195

图书在版编目（CIP）数据

Unity 人工智能游戏开发：第 2 版/（美）雷·巴雷拉（Ray Barrera）等著；张颖译. —北京：清华大学出版社，2016
书名原文：Unity AI Game Programming-Second Edition
ISBN 978-7-302-44690-3

I. ①U…　II. ①雷…　②张…　III. ①游戏程序-程序设计　IV. ①TP311.5

中国版本图书馆 CIP 数据核字（2016）第 184628 号

责任编辑：贾小红
封面设计：刘　超
版式设计：魏　远
责任校对：王　云
责任印制：何　芊

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：三河市君旺印务有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　张**：12　　**字　数**：226 千字
版　　次：2016 年 9 月第 1 版　　**印　次**：2016 年 9 月第1 次印刷
印　　数：1～3000
定　　价：49.00 元

产品编号：068954-01

译 者 序

Unity 是近几年非常流行的一个 3D 游戏开发引擎（特别是移动平台），它的特点是跨平台能力强，支持 PC、Mac、Linux、网页、iOS、Android 等几乎所有的平台，移植便捷，3D 图形性能出众，为众多游戏开发者所喜爱。在手机平台，Unity 几乎成为 3D 游戏开发的标准工具。

Unity 向开发人员提供了多种工具，以实现具有人工智能的游戏体验。Unity 的内建 API 以及特性可有效地实现多种可能性，并构建游戏场景和角色对象。无论开发何种游戏，理解并应用人工智能特性可视为游戏设计的基本因素之一。本书将人工智能技术划分为多个简单概念，进而有助于读者理解这一话题的基础内容。本书通过大量实例，详细分析所涉及的概念，并对核心概念予以实现。

在此基础上，本书还将引领读者实现自己的状态机模式，其中涉及了 AI 的基础型传感器系统，并将其整合至有限状态机中。随后，读者还将领略 Unity 内建的 NavMesh 特性，并实现自己的 A*寻路系统。本书还讲解了群集行为这一核心 AI 概念的实现，并了解行为树的工作和实现方式。最后，状态机系统中还将进一步加入模糊逻辑，并最终实现一个完整的小型游戏项目。

本书将 Unity 与人工智能进行有机的结合，并讨论较为高级的开发技术和解决方案。

在本书的翻译过程中，除张颖外，程聪、解宝香、景秀红、李保金、李莉、李亚楠、梁洪娇、林芮、刘鹤等人也参与了部分翻译工作，在此一并表示感谢。

限于译者的水平，译文中难免有错误和不妥之处，恳请广大读者批评指正。

译　　者

前　言

本书讨论与游戏开发相关的人工智能（AI）技术，并展示了游戏开发过程中的多种应用，或许其处理方式超出了我们的想象。

如果读者想成为 AI 领域的专家，须经历漫长的学习过程。本书提供了与 AI 相关的技术知识和开发工具，读者可在自己的游戏项目中实现 AI，并在此基础上对其进行扩展或创新。

本书示例均配备了相应的代码和项目文件，各章涵盖了与相关概念相符的背景知识以及示例。读者理解后可将其应用于自己的开发项目中。

本书内容

考虑 AI 的广泛性和重要性，第 1 章阐述了与其相关的基本概念。

第 2 章探讨 AI 中应用十分广泛的概念，即有限状态机。

第 3 章介绍游戏 AI 主体的重要实现方式，进而对周边场景予以感知。AI 主体的真实性直接关系到与周围环境间的响应方式。

第 4 章讨论游戏 AI 主体中的路径搜索模式。游戏中的 AI 需要遍历游戏关卡区域，并躲避途中的障碍物。

第 5 章介绍群集模拟算法，以使读者可处理游戏中主体间的整体运动，而非各独立体的单一逻辑。

第 6 章实现了自定义行为树，这也是游戏中复杂 AI 行为中较为常见的处理方式。

第 7 章根据多种因素探讨了游戏 AI 主体的决策制定方式。其中，模糊逻辑可用于模拟人类的决策方式。

第 8 章讨论单目标游戏模板中各类系统间的整合方式，并可方便地对其进行扩展。

背景知识

当运行本书提供的示例项目时，读者需要获取一份 Unity 5 副本。对此，可访问 https://unity3d.com/get-unity 并免费下载 Unity 5 的最新版本。另外，关于 Unity 的系统需求条件，读者可访问 https://unity3d.com/get-unity。

Unity 5 绑定了 MonoDevelop，但本书并不推荐使用该软件，常见的 IDE 即可满足本

书的文本编辑要求。然而，MonoDevelop 涵盖了编写、调试代码的一切内容，包括自动补齐功能，且无须使用任何插件和扩展。

适用读者

本书假定读者已基本了解 C#语言以及 Unity 编辑器，但希望能够编写首款属于自己的游戏，并扩展个人的游戏开发知识。本书提供了大量的与游戏 AI 相关的示例，且不要求读者具备与游戏 AI 相关的任何技术背景知识。

本书约定

本书涵盖了多种文本风格，进而对不同类型的信息加以区分。下列内容展示了对应示例及其具体含义。

文本中的代码、数据库表名称、文件夹名称、文件名、文件扩展名、路径名、伪 URL、用户输入以及推特用户名采用如下方式表示：

" We'll name it TankFsm"。

代码块则通过下列方式设置：

```
using UnityEngine;
using System.Collections;

public class TankPatrolState : StateMachineBehaviour {

  //OnStateEnter is called when a transition starts and the state
 machine starts to evaluate this state
  //override public void OnStateEnter(Animator animator,
AnimatorStateInfo stateInfo, int layerIndex) {
  //
  //}

  //OnStateUpdate is called on each Update frame between
OnStateEnter and OnStateExit callbacks
  //override public void OnStateUpdate(Animator animator,
AnimatorStateInfo stateInfo, int layerIndex) {
  //
```

```
    //}

    //OnStateExit is called when a transition ends and the state
  machine finishes evaluating this state
    //override public void OnStateExit(Animator animator,
  AnimatorStateInfo stateInfo, int layerIndex) {
    //
    //}

    //OnStateMove is called right after Animator.OnAnimatorMove().
  Code that processes and affects root motion should be implemented here
    //override public void OnStateMove(Animator animator,
  AnimatorStateInfo stateInfo, int layerIndex) {
    //
    //}

    //OnStateIK is called right after Animator.OnAnimatorIK(). Code
  that sets up animation IK (inverse kinematics) should be implemented
  here.
    //override public void OnStateIK(Animator animator,
  AnimatorStateInfo stateInfo, int layerIndex) {
    //
    //}
  }
```

最后，“提示”表示一些较为重要的提示；“技巧”则表示相关的操作技巧。

读者反馈和客户支持

欢迎读者对本书的建议或意见予以反馈，以进一步了解读者的阅读喜好。反馈意见对于我们来说十分重要，以便改进我们日后的工作。

对此，读者可向 feedback@packtpub.com 发送邮件，并以书名作为邮件标题。

若读者针对某项技术具有专家级的见解，抑或计划撰写书籍或完善某部著作的出版工作，则可阅读 www.packtpub.com/authors 中的 author guide 一栏。

作为本书的读者支持，我们将对每一名用户提供竭诚的服务。

资源下载

读者可访问 http://www.packtpub.com 下载本书中的示例代码文件；或者访问 http://www.packtpub.com/support，经注册后可直接通过邮件方式获取相关文件。

另外，我们还以 PDF 文件方式提供了本书中截图、图表的彩色图像，以帮助读者进一步理解输出结果中的变化，读者可访问 https://www.packtpub.com/sites/default/files/downloads/8272OT_ColorImages.pdf 下载该 PDF 文件。

勘误表

尽管我们在最大程度上做到尽善尽美，但错误依然在所难免。如果读者发现谬误之处，无论是文字错误抑或是代码错误，还望不吝赐教。对于其他读者以及本书的再版工作，这将具有十分重要的意义。对此，读者可访问 http://www.packtpub.com/submit-errata，选取对应书籍，单击 ErrataSubmissionForm 超链接，并输入相关问题的详细内容。经确认后，填写内容将被提交至网站，或添加至现有勘误表中（位于该书籍的 Errata 部分）。

另外，读者还可访问 http://www.packtpub.com/books/content/support 查看之前的勘误表。在搜索框中输入书名后，所需信息将显示于 Errata 项中。

版权须知

一直以来，互联网上的版权问题从未间断，Packt 出版社对此类问题异常重视。若读者在互联网上发现本书任意形式的副本，请告知网络地址或网站名称，我们将对此予以处理。

关于盗版问题，读者可发送邮件至 copyright@packtpub.com。

对于作者的爱护，我们表示衷心的感谢，并于日后向读者呈现更为精彩的作品。

问题解答

若读者对本书有任何疑问，均可发送邮件至 questions@packtpub.com，我们将竭诚为您服务。

目　录

第 1 章　游戏 AI 的基础知识

一般来讲，人工智能（AI）是一个庞杂的话题，其内容较为艰深，但应用范围却十分规范，例如机器人学、统计学、娱乐业（近期展示出了强大的势头）以及视频游戏。本章目标即是将 AI 应用划分为多种彼此关联同时切实可行的方案，对应示例进一步阐述了相关概念，并直奔主题。

本章还将介绍某些在学术领域、传统领域和游戏领域内与 AI 相关的背景知识，其中包括：

- AI 在游戏中的应用与实现有别于其他领域。
- 考察 AI 在游戏中的特定需求条件。
- 考察游戏中基本的 AI 模式。

本章讨论了 Unity 中 AI 的实现模式，并可作为其他章节的参考内容。

1.1　创建生活幻象

生物体通常包含一定的智能，例如动物和人类，进而可制定某种执行决策。人类的大脑可通过声音、触觉、气味或者视觉对刺激行为予以反馈，并于随后将此类数据转换为可处理的信息。另外一方面，作为一类电子设备，计算机可接收二进制信息，高速执行逻辑和数学计算，并输出最终结果。因此，AI 实际上使得计算机设备具有某种思考能力，并像有机生物一样执行特定操作。

AI 及其相关知识涵盖了大量的内容，在深入讨论这一主题之前，读者应理解 AI 在不同领域内的基本应用。作为一类通用术语，AI 的实现和应用针对不同领域具有不同的处理方式，进而求解不同的问题集。

在考察特定的游戏技术之前，下面首先讨论某些与 AI 相关的科研领域，这些领域在近些年来取得了重大的进步。某些在科幻小说中出现的场景现已成为科学事实，例如自主机器人。AI 的发展也体现在日常的生活中，例如，智能手机即包含了数字辅助特征，并与某些最新出现的 AI 技术有关。下列科研领域对 AI 技术的发展均起到了

推动作用。

- 计算机视觉：从视频和摄像头中获取输入，对其进行分析后执行特定的操作，例如面部识别、物体识别以及光电字符识别。
- 自然语言处理（NLP）：机器能够像人类那样正常地读写并理解自然语言。该问题可描述为，机器通常难以理解人类语言，相同事物存在不同的阐述方式；根据上下文，同一句子也包含了不同的含义。在处理并做出响应之前，由于需要理解人类的语言和表达方式，因而 NLP 是机器处理过程中的一个重要步骤。对此，Web 中存在大量的数据集，以帮助科研工作者进行语言的自动分析处理。
- 常识推理：即使在不熟悉的领域，人类的大脑也可从中获取某种答案。鉴于大脑可在上下文、背景知识以及语言能力间混合、交互信息，因而常识性知识对于人类而言并不是问题。然而，机器处理过程则异常复杂，对于科研工作者而言依然是一项艰巨的挑战。
- 机器学习：这一话题听起来像是取材于科幻电影，但现实与科幻间的差距并不遥远。计算机程序一般包含静态指令集，接受输入数据并提供输出结果。机器学习强调算法和程序，并可从程序所处理的数据中进行学习。

1.2　利用 AI 进一步完善游戏

游戏中的 AI 可追溯到早期作品，例如 Namco 发布的“吃豆人”游戏，其 AI 尚处于初级阶段。但即使如此，敌方角色（Blinky、Pinky、Inky 和 Clyde）均具有各自的行为模式，并通过多种方式向玩家发起挑战。游戏难度的增加使得玩家乐此不疲，自其发布 30 多年来，玩家们依然对此津津乐道。

游戏设计师负责掌控游戏的难度，并制定游戏与玩家的平衡性。最终，AI 可视为一类强大的工具，并有助于实现游戏实体行为模式的抽象化操作，进而使其更具真实感。类似于动画设计师进行逐帧设计，或者艺术家利用手中的画笔进行艺术创作，设计人员或程序员同样可发挥其创造力，并巧妙地利用 AI 技术对游戏予以完善。

AI 在游戏中扮演的角色旨在增加游戏的趣味性，即提供具有一定挑战性的内容，以及相对有趣的非玩家控制角色（即 NPC），并在游戏场景中呈现较为真实的运动行为。当前目标并非是复制人类或动物的全部思考过程，仅是呈现某种幻象以使 NPC 具

有一定的智能，并通过玩家眼中相对正确的方式与游戏场景中变化的环境进行互动。

尽管技术上可设计、构建复杂的模式和行为，但游戏 AI 尚不能模拟真实的人类行为。功能强大的微芯片、大容量内存，甚至是分布式计算，大大提升了 AI 的计算能力。总体上看，数据资源依然会在其他操作间被共享，例如图形渲染、物理模拟、音频处理以及动画等，全部内容均以实时方式呈现。同时，系统间须实现完美整合，并在游戏中提供稳定的帧速率。与游戏开发过程中的其他准则类似，优化 AI 计算对于开发人员而言仍然颇具挑战性。

1.3　在 Unity 使用 AI

本节引领读者大致浏览不同游戏类型中所用的 AI 技术，后续章节将逐一展现 Unity 中的各项特性。Unity 引擎具有强大的灵活性，并提供了多种 AI 模式的实现方法。其中，某些方法可直接使用，而另一些方案则需要从头开始加以构建。本书主要讨论 Unity 中基础的 AI 模式，以使游戏中的 AI 实体正常运行。AI 领域涵盖了大量的内容，本书仅是万里长征的第一步。

1.4　定 义 主 体

在讨论具体技术之前，首先需要明晰本书所采用的一个关键术语，即主体(agent)。考虑到与 AI 紧密相关，因而主体表示为具有人工智能的实体。当谈及 AI 时，并非特指某一具体角色，而是指呈现复杂行为模式的某一实体，且具有非随机性或智能特征。此类实体可以是一个角色、生物体、车辆或者其他事物。主体定义为一类自治实体，并执行所制定的模式和行为。后续章节将据此展开讨论。

1.5　有限状态机概述

有限状态机（FSM）可视为最为简单的 AI 模型之一，且常出现于游戏中。状态机通常包含了一组状态，并通过图中的转换予以连接。一般情况下，游戏实体始于某一初始状态，并搜索触发状态间转换的事件和规则。在任意既定时刻，游戏实体仅处

于一种状态之下。

例如，在典型的射击类游戏中，AI 防御角色通常具有巡逻模式、追逐模式和射击模式，如图 1.1 所示。

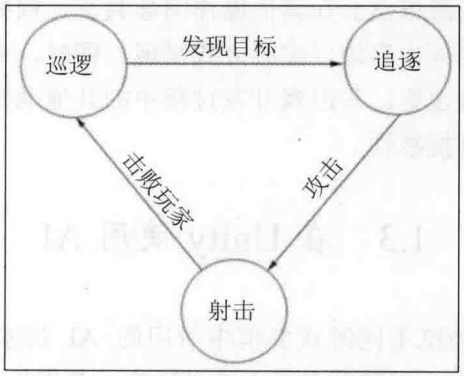

图 1.1

简单的 FSM 中一般包含下列 4 个组件。

- 状态：该组件定义了一组游戏实体或 NPC 可选择的一组唯一状态（例如巡逻、追逐和射击）。
- 转换：该组件定义了不同状态间的关系。
- 规则：该组件用于触发某一状态转换（例如玩家巡视、位于射杀范围以及玩家被摧毁）。
- 事件：该组件负责触发规则检测操作（守卫的可见区域、与玩家间的距离等）。

鉴于实现的简单性、可视化特征以及易于理解，FSM 常用于游戏开发过程中的搜寻 AI 模式。对此，使用简单的 if/else 语句或 switch 语句，即可实现 FSM。如果开始阶段即包含诸多状态和转换，则事态将会变得较为复杂。第 2 章将对 FSM 的管理方式加以深入讨论。

1.6　通过主体视角查看场景

为了保证 AI 行为确实可信，主体对象需要与其周边事件、环境、玩家，甚至是其他主体进行反馈。类似于真实的有机生物，此类主体依赖于视线、声音以及其他物理“刺激”。然而，与真实生物体感知其周围环境相比，主体对象可访问游戏中的大

量数据，例如玩家的位置、距离、库存量、道具的位置，以及在代码中暴露于该主体对象的任意变量。

在图 1.2 中，当前主体对象的视野通过其前方锥体表示，而听觉范围则采用周围的灰色圆形表示。

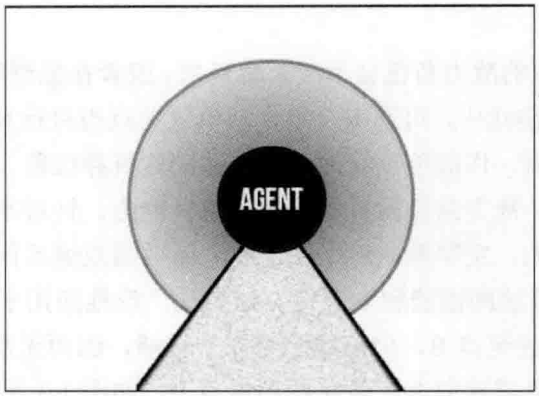

图 1.2

基本上讲，视觉、声音以及其他感觉均可视为数据。其中，视觉表示为光线粒子，声音则表示为一种振动方式等。当然，本书并不会讨论光线粒子反射过程中的复杂行为及其与主体间的作用方式，而是通过某种方式对数据进行建模，进而获得类似的结果。

可以想象，还可采用类似方式对其他感知系统进行建模，其范围并不仅限于生物体（包括视觉、声音或气味），甚至还可以是数字和机械系统，例如敌方的机器人、炮塔、声呐或雷达系统。

1.7　路 径 跟 踪

某些时候，AI 角色需要在游戏场景中按照大致路线或既定路线行进。例如，在赛车类游戏中，AI 车辆需要在公路上进行导航；在 RTS 游戏中，其他作战单位须知晓玩家发出的作战位置，并实现整体前进。

为了进一步展示智能特征，主体对象需要确定目标位置，判断是否可到达该点，选取最佳路线，并在遇到障碍物时调整路线。在后续章节中将会讨论到，各条路径均