

# 游戏与人工智能

杜小勤

武汉纺织大学数学与计算机学院

2018/12/16

# 主要内容

- 游戏;
- 人工智能;
- 游戏与人工智能:
  - Game for AI;
  - AI for Game;

# 人工智能与游戏的关系

长期以来，人工智能与游戏的关系异常密切：两者的历史一样悠久，共同演化，无论是学术界还是工业界，无论是深度还是广度；

游戏（Board Game 和 Video Game）是人工智能的完美测试环境或测试床（Test-bed）；

# 主要研究内容

- 对弈程序——DeepBlue、AlphaGo、AlphaGo Zero、AlphaZero 等；
- 人机游戏（Video Game）对战程序——StarCraft 等；
- 游戏内容的自动生成；
- 游戏 NPCs（非玩家角色 Non-Player Characters, NPCs）中 AI 的设计；
- 玩家模型与玩家情感模型；
- 虚拟现实与增强虚拟现实（VR 与 AR）；
- ...

它是一个新领域：

- 2005, 1st IEEE Symposium(Conference) on Computational Intelligence and Games(CIG)
- 2005, 1st AAAI AIIDE Conference(Artificial Intelligence In Digital Entertainment)
- IEEE Transactions On Computational Intelligence And AI In Games(TCIAIG)  
——IEEE Transactions On Games(ToG),  
renamed from January 2018.

# 相关成果

在 GAME AI 领域，形成了一些新的理论与技术，现在已经成为主流：

- Monte Carlo Tree Search(MCTS):  
CrazeStone, 后来成为 AlphaGo 的三大技术之一;
- Procedural Content Generation: 使用 Neuroevolution 生成游戏内容等;

# 相关成果

- Playing Games based on Screen Capture:  
DeepMind 的 Atari 平台、StarCraft 人机大战平台等。目前，流行的技术有深度强化学习 (Deep Reinforcement Learning)、神经演化 (Neuroevolution) 等；
- Automated Game Design;
- Animation Synthesis;
- ...

# Board Game for AI

自 20 世纪 50 年代 AI 创立以来，Board Game 一直是 AI 研究的重点领域：搜索技术、推理与实时决策等；

甚至在 AI 领域正式形成之前，Board Game 就被作为测试对象——检测程序是否具有智能；

A. S. Douglas: 1952 年, 第 1 个解决 TicTacToe 的对弈程序。采用的技术与方法作为其剑桥大学博士论文的一部分；



# Board Game for AI

Alan Turing: 1953 年，发明了 Minimax 算法，用于 Chess 对弈程序<sup>1</sup>；

Arthur Samuel: 1959 年，Checker 对弈程序，通过 Self-play 方式进行学习——一种强化学习<sup>2</sup>方法<sup>3</sup>；

---

<sup>1</sup>Alan M. Turing. Digital computers applied to games. Faster than thought, 101, 1953.

<sup>2</sup>虽然强化学习在 20 世纪 80 年代作为一个领域才正式形成。

<sup>3</sup>Arthur L. Samuel. Some studies in machine learning using the game of Checkers. IBM Journal of research and development, 3(3):210-229, 1959.

# Board Game for AI

1992 年，Gerald Tesauro 开发的 Backgammon 对弈程序 TD-Gammon，达到了人类顶尖棋手的水平<sup>4</sup>；

它使用了人工神经网络（Artificial Neural Network, ANN）和 Temporal Difference Learning（TD-Learning）算法，采用 Self-Play 方式<sup>5</sup>；

<sup>4</sup>Gerald Tesauro. Practical issues in temporal difference learning. *Machine learning*, 8(3- 4):257-277, 1992.

<sup>5</sup>Gerald Tesauro. Temporal difference learning and TD-Gammon. *Communications of the ACM*, 38(3):58–68, 1995.

# Board Game for AI

经过 30 多年的研究，博弈树的研究取得了突破。

1994 年，Chinook Checker 对弈程序击败了  
Checker 世界冠军 Marion Tinsley<sup>6</sup>；

2007 年，Checker 被完全解决<sup>7</sup>；

---

<sup>6</sup>Jonathan Schaeffer, Robert Lake, Paul Lu, and Martin Bryant.  
Chinook: the world man-machine Checkers champion. AI Magazine,  
17(1):21, 1996.

<sup>7</sup>Jonathan Schaeffer, Neil Burch, Yngvi Bjornsson, Akihiro  
Kishimoto, Martin Míuller, Robert Lake, Paul Lu, and Steve  
Sutphen. Checkers is solved. Science, 317(5844):1518-1522, 2007.

# Board Game for AI

Chess 一度被认为是 AI 界的“果蝇”，许多搜索技术被应用<sup>8</sup>；

1997 年，IBM's Deep Blue 击败了国际象棋世界冠军 Garry Kasparov<sup>9</sup>—— $\alpha - \beta$  算法 + 专家级评估函数 + 超级计算机<sup>10</sup>；

---

<sup>8</sup>Nathan Ensmenger. Is Chess the Drosophila of AI? A Social History of an Algorithm. *Social Studies of Science*, 42(1):5-30, 2012.

<sup>9</sup>Murray Campbell, A. Joseph Hoane, and Feng-hsiung Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57-83, 2002.

<sup>10</sup>Feng-Hsiung Hsu. Behind Deep Blue: Building the computer that defeated the world chess champion. Princeton University Press, 2002.

# Board Game for AI

在计算机国际象棋取得成功之后，计算机围棋 (Computer Go) 成为 AI 界的新“果蝇”——分支因子大约为 250，远远高于国际象棋的分支因子 35；

2016 年，Google DeepMind's AlphaGo 击败韩国 9 段棋手李世石<sup>11</sup>；

---

<sup>11</sup>David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587):484-489, 2016.

# Board Game for AI

2017 年，AlphaGo 的升级版击败世界排名第 1 的中国 9 段棋手柯洁（单机版本）；

AlphaGo 使用了三大技术：Monte Carlo 树搜索、强化学习与深度学习；

# Board Game for AI

AlphaGo 的后续版本 AlphaGo Zero、AlphaZero 已经成为超级棋手，远超人类的顶尖水平；

至此，围棋成为最后一个“值得 AI 研究”<sup>12</sup>且也被解决的 Board Game 问题；

---


<sup>12</sup>计算机围棋也曾经一度被认为是 AI 界的圣杯，人们普遍认为，如果它被解决的话，那么人类离真正的 AI 也就不远了，然而事实并非如此。

# Video Game for AI

在 2001 年之后，Video Game 成为许多 AI 新算法的新测试平台<sup>13</sup>，尤其是 2016 年 AlphaGo 大获成功以后，可应用的技术有：深度学习、深度强化学习、神经演化等；

Laird 和 van Lent 的文章，可以被认为 GAME AI 领域开端的标志；

---

<sup>13</sup>John Laird and Michael van Lent. Human-level AI' s killer application: Interactive computer games. AI Magazine, 22(2):15, 2001. 



# Video Game for AI

Video Game 更具有挑战性：NPCs 的 AI 设计、自动生成游戏 2D/3D 内容、建立玩家模型（例如，难度自适应调节，使游戏更加有趣）、玩家大数据分析、优化游戏设计过程、游戏自动设计、人类级别的人机自动对话等；

# Video Game for AI

早期的 Video Game for AI 研究，主要集中于开发能够 Playing 游戏的 AI 技术及 NPCs AI 的设计；

2014 年，Google DeepMind 开发了能够玩经典的 Atari 2600 视频游戏的程序，达到了顶尖人类玩家的水平——将屏幕像素点作为输入<sup>14</sup>；

---

<sup>14</sup>Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dhharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. Nature, 518(7540):529-533, 2015.


# Video Game for AI

2017 年，Microsoft Maluuba 团队解决了视频游戏 Ms Pac-Man<sup>15</sup>；

Video Game 的解决都离不开一种技术——深度强化学习；

实际上，另一种技术 Neuroevolution 也能够很好地解决 Video Game，它被认为是目前模拟人类大脑进化的最佳技术，也被认为是另一种形式的深度学习；

---

<sup>15</sup>Harm van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. Hybrid Reward Architecture for Reinforcement Learning. arXiv preprint arXiv:1706.04208, 2017. 

# Industry: AI for Video Game

1996, Creatures 游戏 (Millennium Interactive),  
人工神经网络;

1998, Thief 游戏 (EIDOS), 高级感知系统;

2011-2017, Halo 系列游戏 (Microsoft Studios),  
团队战术与真实的战争场景;

Halo 2, 引入行为树技术;

# Industry: AI for Video Game

1997, Blade Runner 游戏 (Virgin Interactive), 基于行为的 AI 技术;

1998, Half-Life 游戏 (Valve), 对手的高级战术;

2000, Black and White 游戏 (EA), 强化学习、Belief-Desire-Intention 模型 (Perceptrons、Decision Trees 等);

2000-2017, The Sims 系列游戏 (EA), 可信智能体;

# Industry: AI for Video Game

2005, Forza Motorsport (MS Game Studios), 模仿学习;

2005, FEAR (Sierra Entertainment), 面向目标的动作规划;

1991-2016, Civilization 系列游戏 (MicroProse, Activision, Infogrames Entertainment, SA and 2K Games), 过程式生成的世界;

2006, Dwarf Fortress (Bay 12 Games), 过程式生成的世界;

# Industry: AI for Video Game

2008, Left 4 Dead (Valve), AI director;

2010, Red Dead Redemption (Rockstar Games), 真实的枪战;

2010, Silent Hill: Shattered Memories (Konami), 个性化的自适应性;

2010, Heavy Rain (Quantic Dream), 基于情感的多镜头表现;

# Industry: AI for Video Game

2010, Supreme Commander 2 (Square Enix),  
使用 Neuroevolution 训练野战排;

2013, The Last of Us (Sony Computer  
Entertainment), AI 搭档;

2013, BioShock Infinite (2K Games), AI 搭档;

2014, Blood & Laurels (Emily Short), 交互式  
旁白;



# Industry: AI for Video Game

2014, Alien: Isolation (Sega), 对手的自适应行为;

2013, Spelunky (Mossmouth, LLC), 过程式生成的世界;

2016, No Man's Sky (Hello Games), 过程式生成的世界;

# 游戏的过程式内容生成

1980 年，出现了过程式内容生成（Procedural Content Generation）技术——利用算法动态地生成游戏内容；

游戏：Rogue、Elite、Diablo III、No Man's Sky、Bloodborne；

# 玩家模型的构建

利用 AI 技术对玩家进行各种分析，可以为创建吸引各种玩家的游戏提供有用的决策；

2009 年，Facebook 的游戏 FarmVille，支持玩家分析及游戏的半自动调整；

2016 年，游戏 Nevermind 可以跟踪玩家的情绪变化并做出相应的调整；

# 游戏图灵测试

2012 年，Turing 诞辰 100 周年，在 Unreal Tournament 2004 游戏中，2 个 AI 控制的 Bots 通过了 Turing 测试<sup>16</sup>；

---

<sup>16</sup>Jacob Schrum, Igor V. Karpov, and Risto Miikkulainen. UT2: Human-like behavior via neuroevolution of combat behavior and replay of human traces. In Computational Intelligence and Games (CIG), 2011 IEEE Conference on, pages 329-336. IEEE, 2011.

# GAMEAI 的前景

游戏和人工智能的融合，具有广阔的前景——科学研究、工程应用、社会与经济等价值，可以很好地与教育、培训、影视、动画、娱乐等领域深度融合；

# GAMEAI 的产生原因

为什么游戏与人工智能的关系如此的密切？

## ■ 游戏

对于游戏而言，需要提高 NPCs 的智能程度，需要提升游戏的有趣性与可玩性，需要建立玩家模型，需要建立情感模型，需要提高游戏设计与开发的效率，需要增加游戏场景与内容的不确定性与丰富性，...

# GAMEAI 的产生原因

为什么游戏与人工智能的关系如此的密切？

## ■ 人工智能

人工智能的研究主题非常丰富，离强人工智能还有相当的距离，人工智能与人类在智能的难易程度上有着很大的差别——对人类而言，极其简单的事情，人工智能不是很擅长；反之，人工智能很擅长的事情，人类却很难做到。

# GAMEAI 的产生原因

特别是，对于那些人类依靠潜意识或意识执行的任务，人类自己却无法进行有效的形式化、知识符号化与程序化，以教会人工智能算法进行有效的执行。例如，图象识别、语言翻译与语音识别等；

游戏的设计、实现、2D/3D 内容的生成、玩家模型、NPCs 的设计及玩游戏本身等正是这样一种任务，游戏是人工智能的理想测试平台！



# GAMEAI 的新测试平台

计算机围棋 (Computer Go) 大约有  $10^{170}$  个状态，可以认为已经被解决；

StarCraft 和 StarCraft II，实时策略游戏，目前最好的 StarCraft Bots 只达到了业余玩家的水平，大约有  $10^{1685}$  个状态<sup>17</sup>——多目标任务、部分信息、大量的对象，成为 AI 技术的新测试平台；


---

<sup>17</sup>Nicolas Usunier, Gabriel Synnaeve, Zeming Lin, and Soumith Chintala. Episodic Exploration for Deep Deterministic Policies: An Application to StarCraft Micromanagement Tasks. arXiv preprint arXiv:1609.02993, 2016.

# GAMEAI 的新测试平台

而在可观察的宇宙中，质子的个数<sup>18</sup>大约是  $10^{80}$ 。

---

<sup>18</sup>Arthur S. Eddington. The Constants of Nature. In The World of Mathematics 2, pages 1074-1093. Simon & Schuster, 1956. 

# GAMEAI 的新测试平台

继推出 Arcade Learning Environment (Playing Atari 游戏) 之后, Google DeepMind 又推出了 StarCraft II 测试平台;

Facebook AI Research 开发了 TorchCraft, 在深度学习库 Torch 和 StarCraft 之间建立了联系;

Alibaba 也开启了 StarCraft 的 AI 之旅<sup>19</sup>;

---

<sup>19</sup>Peng Peng, Quan Yuan, Ying Wen, Yaodong Yang, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent Bidirectionally-Coordinated Nets for Learning to Play StarCraft Combat Games. arXiv preprint arXiv:1703.10069, 2017.

# 游戏类型 I

- 动作: 格斗, 射击 (FPS, TPS), ...;
- 动作冒险: 潜行类, 恐怖类, ...;
- 冒险: 文字冒险, 图形冒险 (密室逃脱), 互动式电影, ...;
- 大型多人在线 (MMO): 大型多人在线第一人称射击大型多人在线角色扮演大型多人在线即时战略 (MMOFPS, MMORPG, MMORTS) ;
- 角色扮演: 动作角色扮演, 迷宫探索, 多人角色扮演 (MUD, Multi-User Dungeon), 战术角色扮演, ...;

# 游戏类型 II

- 策略: 即时策略 (多人线上战斗竞技场-塔防, 大型多人在线即时策略-对战平台), 即时策略/战术, 回合制策略/战术, 战争, ...;
- 模拟: 建造与经营 (商业, 城市, 政府), 生活模拟 (宠物, 上帝模拟, 社会模拟), 体育, 交通工具模拟 (飞行模拟, 竞速模拟, 训练模拟, 水下模拟), ...;
- 其它类型: 益智, 音乐, 教育, 广告, 听觉, ...;

# 游戏类型

每种游戏类型所需要的技术不一样，它们都有自己鲜明的特点；

# Action Games: FPS 游戏

典型游戏：Doom, Quake, Half Life, Unreal or Goldeneye;

# 早期的 FPS 游戏



图 2-1: Wolfenstein



# 早期的 FPS 游戏



图 2-2: Doom

# 早期的 FPS 游戏



图 2-3: Quake

# 早期的 FPS 游戏



图 2-4: Quake III Arena

# 早期的 FPS 游戏

它们的图形质量越来越好，ID Software 是早期 FPS 游戏的开拓者之一，John D. Carmack 是其代表人物。



图 2-5: John D. Carmack

# Strategy Games

- Real Time Strategy (RTS);
- Turn Based Strategy (TBS);

典型游戏：Age of Empires, Command & Conquer, Civilization, StarCraft;

# Strategy Games

策略游戏是一种广泛存在于图板游戏、电视游戏和电脑游戏的游戏形式。它要求游戏的参与者“拥有”做出决策的能力。在策略游戏中，决策对游戏的结果产生至关重要的影响。

# 分类

按照对现实世界的依赖程度，可以分为抽象策略游戏和模拟策略游戏。

抽象策略游戏与现实世界的关系不大，是一种基于逻辑的游戏，这种游戏通常有游戏参与者必须遵守的简明的游戏规则。著名的抽象策略游戏有中国象棋、国际象棋和围棋等，它们流传千百年，至今经久不衰。

# 分类

模拟策略游戏则将游戏的重点放在对现实世界策略的模拟上。在这类游戏中，参与者的每一个决策，都将影射到模拟世界中。



# 分类

依照按排决策进行顺序的方式，可以分为即时策略游戏和回合制策略游戏。

# 分类

在即时策略游戏，所有的决策都是即时进行的，即：游戏是连续的，你可以在游戏进行中的任何时间做出并完成决策。

而回合制策略游戏则相反，游戏是基于回合的。在回合制策略游戏中，参与者要依照游戏规则轮流做出决策，只有当一方完成决策后，其他参与者才能进行决策。大部分非电脑游戏都是回合制策略游戏，然而也有极少数的非电脑策略游戏是即时策略的，如 Icehouse。

# RTS

即时策略游戏 RTS 通常包含了各种类型的单元 (Units)，每个单元有自己的优势和劣势，有不同的移动速度、健康值、攻击能力，可以用于战斗、巡逻，可以担负巡逻任务，许多 RTS 允许构造新的单元。

RTS 允许玩家对许多单元进行非直接的控制。



图 2-6: Globulation 2

# RTS

此外，大多数 RTS 要求玩家收集各种资源，以用于建造新的单元，允许玩家通过消耗资源来获得生产设备和单元的升级。

许多 RTS 游戏给玩家提供一个鹰眼窗口，让玩家对整个场景有一个有效的操控。另外，地图的探索也是一个有趣的体验——玩家尚未探索的区域是黑色的。

# Freeciv



图 2-7: Freeciv is an open source implementation of the Civilization series

# Path Finding

Path Finding 是策略游戏的关键技术之一；

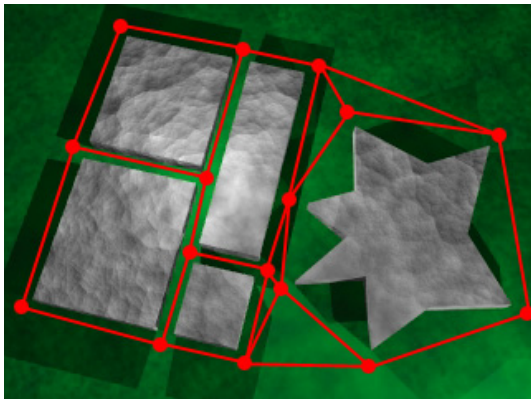


图 2-8: Node、Graph and Path Finding

# Path Finding

计算机游戏中使用的主要技术：A\* 算法。

A\* 算法 DEMO、RTS 游戏中的 Path Finding DEMO（文件：Pathfinding Demonstration for Real Time Strategy game. - YouTube [360p].mp4）



# Simulation Games

典型游戏：Microsoft Flight Simulator、Need for Speed, NASCAR Racing;

# FlightGear



图 2-9: FlightGear, a flight simulator video game

# Simulation Games

模拟游戏包括的游戏种类非常多，它是对现实世界和虚构世界在某种程度上的一种模拟。

它致力于模拟各种实际活动，用于各种目的，例如娱乐、训练、分析、预测等。

# Simulation Games

模拟游戏涉及到的技术非常多，它要对现实世界的若干活动进行有效和有趣的建模。

# Vehicle Simulation Games

VSG 给玩家提供了操控各种交通工具的真实体验，这些交通工具包括：automobiles, aircraft, watercraft, spacecraft, military vehicles, and a variety of other vehicles。

VSG 给玩家提供 pilot 和 driver 视角的体验，它的主要挑战在于玩家需要掌握驾驶和操控技巧，另外一些游戏还增加了另一种挑战——竞速和格斗（飞机和坦克等）。

有些 VSG 还给玩家提供了更加真实的物理环境和挑战，例如油箱管理等。

# Vehicle Simulation Games



图 2-10: Microsoft Flight Simulator focuses solely on the experience of flying an aircraft, in contrast to other games with missions and goals.

# Vehicle Simulation Games

VSG 的玩家可以分为 2 大类：

- Purists;
- Casual players;

前者要求 VSG 高度精确，后者则不要求那么精确。

例如，对于 Flight simulator 来讲，前者要求跟实际情况一致，而后者则要求简化飞行流程，期望开飞机如同开汽车那般简单。这些都对 VGS 的实现提出了不同的要求。

# 人工智能的方法 I

## ■ Ad-Hoc Behavior Authoring

- Finite State Machine;
- Behavior Tree;
- Utility-Based AI;

## ■ Tree Search

- Uninformed Search;
- Best-First Search;
- Minimax、 $\alpha - \beta$ ;
- Monte Carlo Tree Search;

## ■ Evolutionary Computation

- Genetic Algorithm;



# 人工智能的方法 II

- Neuroevolution;
- Supervised Learning
  - Artificial Neural Network;
  - Support Vector Machine;
  - Decision Tree;
- Reinforcement Learning & Deep Reinforcement Learning
- Unsupervised Learning
  - Clustering;
  - Frequent Pattern Mining;

# Finite State Machines

2000 年代中期, FSMs<sup>20</sup>和 Hierarchical FSMs, 一直作为设计 NPCs AI 的主流技术;

FSMs 使用图来表示:

- States;
- Transitions;
- Actions;

---

<sup>20</sup>Arthur Gill. Introduction to the theory of Finite-State Machines. McGraw-Hill, 1962.

# Finite State Machines

FSMs 易于设计、实现、可视化与调试；

但是，不易扩展，复杂度高时，不易维护，缺少灵活性与动态性；

可以引入 Fuzzy Rules<sup>21</sup>和 Probabilities<sup>22</sup>，部分解决对应的问题；

---

<sup>21</sup>Michele Pirovano. The use of Fuzzy Logic for Artificial Intelligence in Games. Technical report, University of Milano, Milano, 2012.

<sup>22</sup>Alex J. Champandard. AI game development: Synthetic creatures with learning and reactive behaviors. New Riders, 2003.

# Finite State Machines

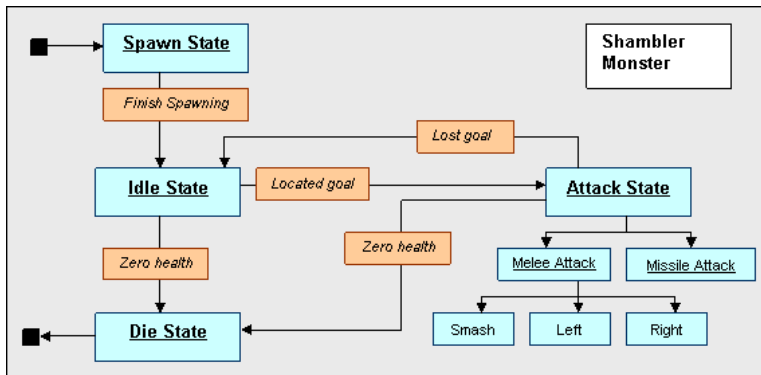


图 2-11: Shambler Monster 的层次有限状态机

# Behavior Trees

BT<sup>23</sup>与 FSMs 类似，但是模块性更好——BT 是由行为构成的，而不是状态。

与 FSMs 一样，BT 易于设计、测试与调试；

在成功应用于 Halo 2(Microsoft Game Studios, 2004)<sup>24</sup>与 Bioshock(2K Games, 2007) 之后，成为主流技术之一；

---

<sup>23</sup>Alex J. Champanhard. Behavior trees for next-gen game AI. In Game Developers Conference, Audio Lecture, 2007.

<sup>24</sup>Damian Isla. Handling complexity in the Halo 2 AI. In Game Developers Conference, 2005.

# Behavior Trees

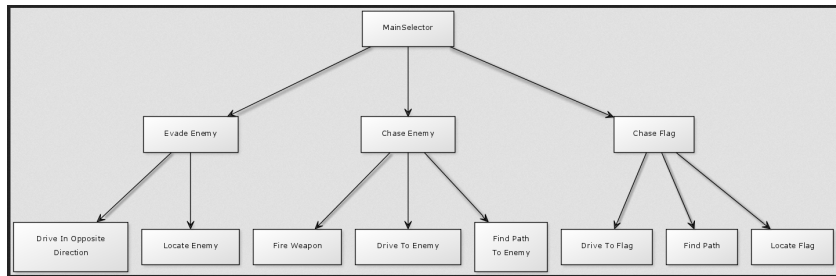


图 2-12: 行为树示例

# Behavior Trees

一般地，BT 由 3 种类型的节点组成：

- Sequence;
- Selector;
- Decorator;

与 FSMs 一样，缺乏灵活性与动态性，已经提出了一些改进方法；

# Utility-Based AI

在游戏开发领域，无论是 NPCs AI 的设计，还是其它功能的设计，都需要高度可信的模块化代码；

FSMs 和 BTs 存在一定的模块化问题<sup>25</sup>，  
Utility-Based AI (UBAI) 于是应运而生<sup>26</sup>；

---

<sup>25</sup>Jakob Rasmussen. Are Behavior Trees a Thing of the Past?  
Gamasutra, 2016.

<sup>26</sup>Dave Mark. Behavioral Mathematics for game AI. Charles River  
Media, 2009.



# Utility-Based AI

每个 AI 对象，有一个可供选择的行为集合，每个行为被赋予一个 Utility Value，AI 对象选择具有最高 Utility 的行为<sup>27</sup>；

这种方式，使得 AI 行为的设计与实现可以实现高度的模块化，并且易于扩展<sup>28</sup>；

---

<sup>27</sup>Bob Alexander. The beauty of response curves. AI Game Programming Wisdom, page 78, 2002.

<sup>28</sup>Kevin Dill. A pattern-based approach to modular AI for Games. Game Programming Gems, 8:232-243, 2010.

# MS Pac-Man



图 2-13: MS. Pac-Man

# MS Pac-Man

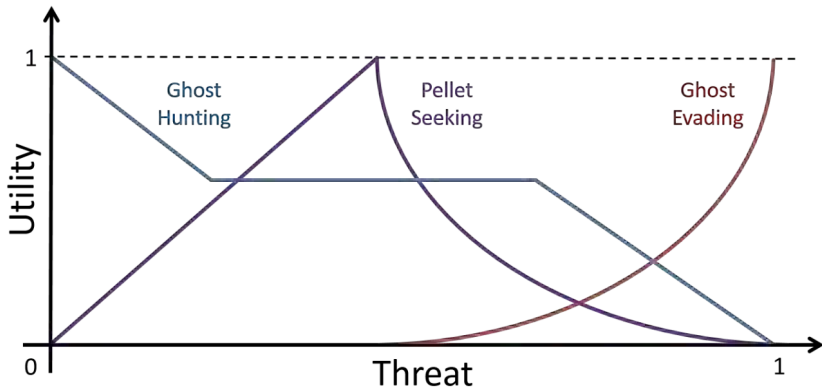


图 2-14: MS. Pac-Man 的效用函数

# Utility-Based AI

效用函数：任一可用事件的量化，例如角色和敌人的健康值、弹药数量等；

函数可以是线性的，也可以是非线性的；

设计时，可以将 NPCs AI 需要考虑的每个因素，赋予一个衡量重要程度的权值；

这些因素可以根据实际场景进行增减，动态变化，非常方便；

# Weapon Selection System<sup>29</sup>

需要考虑的因素: Range, Inertia, Random Noise, Ammo and Indoors;

NPCs 定期检测每个武器的效用, 选择具有最高效用的武器;

---

<sup>29</sup>Dave Mark and Kevin Dill. Improving AI decision modeling through utility theory. In Game Developers Conference, 2010.

# Ad-Hoc Behavior Authoring

虽然上述几种方法属于基本的 AI 设计方法，但是通过引入模糊与概率技术，可以设计出动态的 AI 行为；

无论是 FSMs、BTs，还是基于效用的 AI，都可以作为 NPCs AI 的基本框架，并且可以引入任何其它更复杂、更高级的 AI 算法来取代其中的小模块，从而形成所谓的 AI 混合框架；


# Tree Search

搜索是 AI 领域基本的问题，曾经也是最重要的问题<sup>30</sup>。

许多 AI 问题可以看作是搜索问题——找到最优解，例如最短路径、最佳下法等；

Tree Search 是常见的状态空间表示法；

---

<sup>30</sup>Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs, 1995. 

# Uninformed Search

无信息搜索，没有利用任何与搜索目标相关的信息，包括 Breadth-First Search(BFS) 和 Depth-First Search(DFS) 算法；

大部分情况下，被看作是计算机科学中的算法，而不认为是 AI 中的算法；

DFS 与回溯法相结合是最常见的形式，用途广泛，已经成为一种基本的算法框架；



# Uninformed Search

在游戏中的应用：Iterative Width Search（论文<sup>31</sup>）和 BFS（论文<sup>32</sup>）

---

<sup>31</sup>Blai Bonet and Hector Geffner. Planning as heuristic search. *Artificial Intelligence*, 129(1- 2):5-33, 2001.

<sup>32</sup>Antonios Liapis, Georgios N. Yannakakis, and Julian Togelius. Sentient Sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*, pages 213-220, 2013.

# Best-First Search

最佳优先搜索算法是一类算法的统称，在形式上可以特别化为无信息搜索和启发式搜索，虽然在一般情况下，认为最佳优先搜索属于启发式搜索算法；

A\* 算法是最佳优先搜索的典型代表，游戏中被广泛应用于 NPCs 的路径规划等任务中；

The winner of the 2009 Mario AI Competition  
是一个 A\* Planner<sup>33</sup>；

<sup>33</sup>Julian Togelius, Sergey Karakovskiy, and Robin Baumgarten. The 2009 Mario AI competition. In Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, 2010.

# 博弈树搜索

Minimax 算法可以用来设计棋类对弈程序<sup>34</sup>，例如 TicTacToe、国际象棋和 Checker 等；

$\alpha - \beta$  算法是 Minimax 算法的改进，通过剪枝技术有效地增强了搜索效率，例如 IBM 的 Deep Blue 国际象棋对弈程序等；

---

<sup>34</sup>Alan M. Turing. Digital computers applied to games. Faster than thought, 101, 1953.

# 博弈树搜索

Monte Carlo Tree Search (MCTS) 于 2007 年提出，最早应用于计算机围棋程序 CrazeStone，在计算机围棋比赛中取得了令人瞩目的成绩；

AlphaGo 的三大技术之一，便是 MCTS<sup>35</sup>；

---

<sup>35</sup>另两项技术是强化学习与深度学习。

# MS. Pac-Man

Maastricht 是一个 MCTS 实时角色控制器，在 2012 年 IEEE Computational Intelligence and Games 举办的 MS. Pac-Man 竞赛中取得第 1 名<sup>36</sup>；

---

<sup>36</sup>Tom Pepels, Mark H. M. Winands, and Marc Lanctot. Real-time Monte Carlo tree search in Ms Pac-Man. IEEE Transactions on Computational Intelligence and AI in Games, 6(3):245-257, 2014. ▶

# Evolutionary Computation

基于生物进化理论，适用 8 字原则“优胜劣汰，适者生存”；

模拟生物进化进程：建立种群；通过复制/交叉/变异产生后代；依照适应度函数评价个体。上述过程，一直迭代下去，直到满足应用需求为止；

最新的研究领域是 Neuroevolution，即神经演化。典型的框架是 NEAT/HyperNEAT；

Neuroevolution 取得了许多重要的成果，被认为是另一种形式的深度学习技术；

# Supervised Learning

典型代表：Artificial Neural Networks、Support Vector Machines、Decision Tree；

典型的应用：分类与回归。几乎可以解决所有的问题，应用非常广泛，尤其是近几年兴起的深度神经网络技术；

# Reinforcement Learning

强化学习是一种通过与环境的交互并得到环境的反馈进行学习的技术；

应用非常广泛，可以与 Neuroevolution、ANN 等技术结合，可以解决许多问题——从 Playing Board game 和 Video game 到机器人学习与控制等；

最新的方法是深度强化学习与神经演化强化学习；



# 主要学习内容 I

- 掌握程序设计语言：  
C/C++/Python/MATLAB/OCTAVE 等；
- 掌握数据结构与计算机算法；
- 学好高等数学、离散数学、线性代数、概率论等；
- 人工智能与机器学习；
- 学好数电、模电、DSP、图像处理、计算机图形学、机器学习、AI 等；
- 学习使用 FPGA、GPU、Arduino、Raspberry PI 等；

# 主要学习内容 II

- 学习制作四旋翼无人机、各型机器人等；
- 学会使用 Ubuntu 环境；
- 学会使用 Emacs、L<sup>A</sup>T<sub>E</sub>X 来编写程序与编辑文档；
- ...

作为计算机专业的学生，要充分合理地利用各种软硬件资源、工具来解决领域问题，要具有扎实的程序设计、算法与数学背景。这些对于自己的职业发展与学术深造是非常重要的。当前，各种技术与方法的融合趋势已经相当明显，期望运用单一的技能，是不现实的...

# 学习方法举例

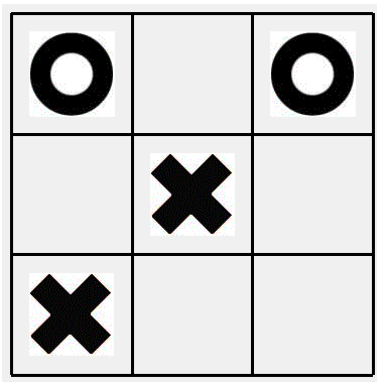


图 2-15: TicTacToe

# 学习方法举例

针对 TicTacToe 问题，可以应用不同的算法来解决：

- Minimax、 $\alpha - \beta$  算法；
- Monte Carlo Tree Search 算法；
- Reinforcement Learning；
- Genetic Algorithm；
- Deep Reinforcement Learning；
- Neuroevolution；
- ...

作用：可以比较不同的方法，可以透过现象发现更一般性的原理与本质——实际上，这也是科学发现的一般规律；

# 学习方法举例

在人工智能与机器学习领域，任何成功的智能体（Agent）都需要做好利用与探索（Exploitation & Exploration, EE）的平衡——DRL 算法、MCTS 算法、ANN 算法等，无一例外；

作为学习者的我们，也常常需要离开自己的学习“舒适区”，对各种未知领域保持一颗好奇的心，维持知识结构的广度与深度的平衡，才能不断地提升自己，让自己成为一名合格的研究与工程人员！

# 参考文献

1. The Use of Mathematics in Computer Games. Dan Goodman.
2. John D. Carmack.
3. Mathematics for Computer Games Technology.
4. 策略游戏.
5. Strategy video game.
6. Vehicle Simulation Games.
7. Georgios N. Yannakakis and Julian Togelius. Artificial Intelligence and Games. Springer, January 26, 2018.