

ARTIFICIAL INTELLIGENCE FOR GAMES

IAN MILLINGTON



ELSEVIER

AMSTERDAM • BOSTON . HEIDELBERG
LONDON • NEW YORK * OXFORD
PARIS • SAN DIEGO . SAN FRANCISCO
SINGAPORE . SYDNEY • TOKYO

Morgan Kaufmann is an imprint of Elsevier

MORGAN KAUFMANN PUBLISHER

CONTENTS

ABOUT THE AUTHOR	VI
LIST OF FIGURES	XXI
ACKNOWLEDGMENTS	XXIX
PREFACE	XXXI
ABOUT THE CD-ROM	XXXIII

PART I

AI AND GAMES

CHAPTER

INTRODUCTION

- 1.1 WHAT IS AI?
 - 1.1.1 Academic AI
 - 1.1.2 Game AI
- 1.2 MY MODEL OF GAME AI
 - 1.2.1 Movement
 - 1.2.2 Decision Making
 - 1.2.3 Strategy
 - 1.2.4 Infrastructure
 - 1.2.5 Agent-Based AI
 - 1.2.6 In the Book
- 1.3 ALGORITHMS, DATA STRUCTURES, AND REPRESENTATIONS
 - 1.3.1 Algorithms
 - 1.3.2 Representations

VIII Contents

1.4 ON THE CD

1.4.1 Programs

1.4.2 Libraries

15 LAYOUT OF THE BOOK

CHAPTER

GAME AI

2.1 THE COMPLEXITY FALLACY

2.1.1 When Simple Things Look Good

2.1.2 When Complex Things Look Bad

2.1.3 The Perception Window

2.1.4 Changes of Behavior

2.2 THE KIND OF AI IN GAMES

2.2.1 Hacks

2.2.2 Heuristics

2.2.3 Algorithms

2.3 SPEED AND MEMORY

2.3.1 Processor Issues

2.3.2 Memory Concerns

2.3.3 PC Constraints

2.3.4 Console Constraints

2.4 THE AI ENGINE

2.4.1 Structure of an AI Engine

2.4.2 Toolchain Concerns

2.4.3 Putting It All Together

PART II

TECHNIQUES

CHAPTER

MOVEMENT

3.1 THE BASICS OF MOVEMENT ALGORITHMS

3.1.1 Two -Dimensional Movement

3.1.2 Statics

3.1.3 Kinematics

3.2 KINEMATIC MOVEMENT ALGORITHMS

3.2.1 Seek

3.2.2 Wandering

- 3.2.3 On the CD
- 3.3 STEERING BEHAVIORS**
 - 3.3.1 Steering Basics
 - 3.3.2 Variable Matching
 - 3.3.3 Seek and Flee
 - 3.3.4 Arrive
 - 3.3.5 Align
 - 3.3.6 Velocity Matching
 - 3.3.7 Delegated Behaviors
 - 3.3.8 Pursue and Evade
 - 3.3.9 Face
 - 3.3.10 Looking Where You're Going
 - 3.3.11 Wander
 - 3.3.12 Path Following
 - 3.3.13 Separation
 - 3.3.14 Collision Avoidance
 - 3.3.15 Obstacle and Wall Avoidance
 - 3.3.16 Summary
- 3.4 COMBINING STEERING BEHAVIO**
 - 3.4.1 Blending and Arbitration
 - 3.4.2 Weighted Blending
 - 3.4.3 Priorities
 - 3.4.4 Cooperative Arbitration
 - 3.4.5 Steering Pipeline
- 3.5 PREDICTING PHYSICS**
 - 3.5.1 Aiming and Shooting
 - 3.5.2 Projectile Trajectory
 - 3.5.3 The Firing Solution
 - 3.5.4 Projectiles with Drag
 - 3.5.5 Iterative Targeting
- 3.6 JUMPING
 - 3.6.1 Jump Points
 - 3.6.2 Landing Pads
 - 3.6.3 Hole Fillers
- 3.7 COORDINATED MOVEMENT**
 - 3.7.1 Fixed Formations
 - 3.7.2 Scalable Formations
 - 3.7.3 Emergent Formations
 - 3.7.4 Two-Level Formation Steering
 - 3.7.5 Implementation
 - 3.7.6 Extending to More than Two Levels
 - 3.7.7 Slot Roles and Better Assignment

X Contents

- 3.7.8 Slot Assignment
- 3.7.9 Dynamic Slots and Plays
- 3.7.10 Tactical Movement
- 38 MOTOR CONTROL
 - 3.8.1 Output Filtering
 - 3.8.2 Capability-Sensitive Steering
 - 3.8.3 Common Actuation Properties
- 3.9 MOVEMENT IN THE THIRD DIMENSION
 - 3.9.1 Rotation in Three Dimensions
 - 3.9.2 Converting Steering Behaviors to Three Dimensions
 - 3.9.3 Align
 - 3.9.4 Align to Vector
 - 3.9.5 Face
 - 3.9.6 Look Where You're Going
 - 3.9.7 Wander
 - 3.9.8 Faking Rotation Axes

CHAPTER

PATH FINDING

- 4.1 THE PATHFINDING GRAPH
 - 4.1.1 Graphs
 - 4.1.2 Weighted Graphs
 - 4.1.3 Directed Weighted Graphs
 - 4.1.4 Terminology
 - 4.1.5 Representation
- 4.2 DIJKSTRA
 - 4.2.1 The Problem
 - 4.2.2 The Algorithm
 - 4.2.3 Pseudo-Code
 - 4.2.4 Data Structures and Interfaces
 - 4.2.5 Performance of Dijkstra
 - 4.2.6 Weaknesses
- 4.3 A*
 - 4.3.1 The Problem
 - 4.3.2 The Algorithm
 - 4.3.3 Pseudo-Code
 - 4.3.4 Data Structures and Interfaces
 - 4.3.5 Implementation Notes
 - 4.3.6 Algorithm Performance
 - 4.3.7 Node Array A*
 - 4.3.8 Choosing a Heuristic

- 4.4 WORLD REPRESENTATIONS
 - 4.4.1 Tile Graphs
 - 4.4.2 Dirichlet Domains
 - 4.4.3 Points of Visibility
 - 4.4.4 Polygonal Meshes
 - 4.4.5 Non-Translational Problems
 - 4.4.6 Cost Functions
 - 4.4.7 Path Smoothing
- 4.5 IMPROVING ON A*
- 4.6 HIERARCHICAL PATHFINDING
 - 4.6.1 The Hierarchical Pathfinding Graph
 - 4.6.2 Pathfinding on the Hierarchical Graph
 - 4.6.3 Hierarchical Pathfinding on Exclusions
 - 4.6.4 Strange Effects of Hierarchies on Pathfinding
 - 4.6.5 Instanced Geometry
- 4.7 OTHER IDEAS IN PATHFINDING
 - 4.7.1 Open Goal Pathfinding
 - 4.7.2 Dynamic Pathfinding
 - 4.7.3 Other Kinds of Information Reuse
 - 4.7.4 Low Memory Algorithms
 - 4.7.5 Interruptible Pathfinding
 - 4.7.6 Pooling Planners
- 4.8 CONTINUOUS TIME PATHFINDING
 - 4.8.1 The Problem
 - 4.8.2 The Algorithm
 - 4.8.3 Implementation Notes
 - 4.8.4 Performance
 - 4.8.5 Weaknesses
- 4.9 MOVEMENT PLANNING
 - 4.9.1 Animations
 - 4.9.2 Movement Planning
 - 4.9.3 Example
 - 4.9.4 Footfalls

CHAPTER

DECISION MAKING

- 5.1 OVERVIEW OF DECISION MAKING
- 5.2 DECISION TREES
 - 5.2.1 The Problem
 - 5.2.2 The Algorithm

XII Contents

- 5.2.3 Pseudo-Code
- 5.2.4 On the CD
- 5.2.5 Knowledge Representation
- 5.2.6 Implementation Nodes
- 5.2.7 Performance of Decision Trees
- 5.2.8 Balancing the Tree
- 5.2.9 Beyond the Tree
- 5.2.10 Random Decision Trees
- 5.3 STATE MACHINES
 - 5.3.1 The Problem
 - 5.3.2 The Algorithm
 - 5.3.3 Pseudo-Code
 - 5.3.4 Data Structures and Interfaces
 - 5.3.5 On the CD
 - 5.3.6 Performance
 - 5.3.7 Implementation Notes
 - 5.3.8 Hard-Coded FSM
 - 5.3.9 Hierarchical State Machines
 - 5.3.10 Combining Decision Trees and State Machines
- 5.4 FUZZY LOGIC
 - 5.4.1 Introduction to Fuzzy Logic
 - 5.4.2 Fuzzy Logic Decision Making
 - 5.4.3 Fuzzy State Machines
- 5.5 MARKOV SYSTEMS
 - 5.5.1 Markov Processes
 - 5.5.2 Markov State Machine
- 5.6 GOAL-ORIENTED BEHAVIOR
 - 5.6.1 Goal-Oriented Behavior
 - 5.6.2 Simple Selection
 - 5.6.3 Overall Utility
 - 5.6.4 Timing
 - 5.6.5 Overall Utility GOAP
 - 5.6.6 GOAP with IDA*
 - 5.6.7 Smelly GOB
- 5.7 RULE-BASED SYSTEMS
 - 5.7.1 The Problem
 - 5.7.2 The Algorithm
 - 5.7.3 Pseudo-Code
 - 5.7.4 Data Structures and Interfaces
 - 5.7.5 Implementation Notes
 - 5.7.6 Rule Arbitration
 - 5.7.7 Unification

- 5.7.8 Rete
- 5.7.9 Extensions
- 5.7.10 Where Next
- 5.8 BLACKBOARD ARCHITECTURES
 - 5.8.1 The Problem
 - 5.8.2 The Algorithm
 - 5.8.3 Pseudo-Code
 - 5.8.4 Data Structures and Interfaces
 - 5.8.5 Performance
 - 5.8.6 Other Things Are Blackboard Systems
- 5.9 SCRIPTING
 - 5.9.1 Language Facilities
 - 5.9.2 Embedding
 - 5.9.3 Choosing a Language
 - 5.9.4 A Language Selection
 - 5.9.5 Rolling Your Own
 - 5.9.6 Scripting Languages and Other AI
- 5.10 ACTION EXECUTION
 - 5.10.1 Types of Action
 - 5.10.2 The Algorithm
 - 5.10.3 Pseudo-Code
 - 5.10.4 Data Structures and Interfaces
 - 5.10.5 Implementation Notes
 - 5.10.6 Performance
 - 5.10.7 Putting It All Together

CHAPTER

TACTICAL AND STRATEGIC AI

- 6.1 WAYPOINT TACTICS
 - 6.1.1 Tactical Locations
 - 6.1.2 Using Tactical Locations
 - 6.1.3 Generating the Tactical Properties of a Waypoint
 - 6.1.4 Automatically Generating the Waypoints
 - 6.1.5 The Condensation Algorithm
- 6.2 TACTICAL ANALYSES
 - 6.2.1 Representing the Game Level
 - 6.2.2 Simple Influence Maps
 - 6.2.3 Terrain Analysis
 - 6.2.4 Learning with Tactical Analyses
 - 6.2.5 A Structure for Tactical Analyses
 - 6.2.6 Map Flooding

- 6.2.7 Convolution Filters
- 6.2.8 Cellular Automata
- 6.3 TACTICAL PATHFINDING
 - 6.3.1 The Cost Function
 - 6.3.2 Tactic Weights and Concern Blending
 - 6.3.3 Modifying the Pathfinding Heuristic
 - 6.3.4 Tactical Graphs for Pathfinding
 - 6.3.5 Using Tactical Waypoints
- 6.4 COORDINATED ACTION
 - 6.4.1 Multi-Tier AI
 - 6.4.2 Emergent Cooperation
 - 6.4.3 Scripting Group Actions
 - 6.4.4 Military Tactics

CHAPTER

LEARNING

- 7.1 LEARNING BASICS
 - 7.1.1 Online or Offline Learning
 - 7.1.2 Intra-Behavior Learning
 - 7.1.3 Inter-Behavior Learning
 - 7.1.4 A Warning
 - 7.1.5 Over-learning
 - 7.1.6 The Zoo of Learning Algorithms
 - 7.1.7 The Balance of Effort
- 7.2 PARAMETER MODIFICATION
 - 7.2.1 The Parameter Landscape
 - 7.2.2 Hill Climbing
 - 7.2.3 Extensions to Basic Hill Climbing
 - 7.2.4 Annealing
- 7.3 ACTION PREDICTION
 - 7.3.1 Left or Right
 - 7.3.2 Raw Probability
 - 7.3.3 String Matching
 - 7.3.4 N-Grams
 - 7.3.5 Window Size
 - 7.3.6 Hierarchical N-Grams
 - 7.3.7 Application in Combat
- 7.4 DECISION LEARNING
 - 7.4.1 Structure of Decision Learning
 - 7.4.2 What Should You Learn?
 - 7.4.3 Three Techniques

- 7.5 DECISION TREE LEARNING
 - 7.5.1 ID3
 - 7.5.2 ID3 with Continuous Attributes
 - 7.5.3 Incremental Decision Tree Learning
- 7.6 REINFORCEMENT LEARNING
 - 7.6.1 The Problem
 - 7.6.2 The Algorithm
 - 7.6.3 Pseudo-Code
 - 7.6.4 Data Structures and Interfaces
 - 7.6.5 Implementation Notes
 - 7.6.6 Performance
 - 7.6.7 Tailoring Parameters
 - 7.6.8 Weaknesses and Realistic Applications
 - 7.6.9 Other Ideas in Reinforcement Learning
- 7.7 ARTIFICIAL NEURAL NETWORKS
 - 7.7.1 Overview
 - 7.7.2 The Problem
 - 7.7.3 The Algorithm
 - 7.7.4 Pseudo-Code
 - 7.7.5 Data Structures and Interfaces
 - 7.7.6 Implementation Caveats
 - 7.7.7 Performance
 - 7.7.8 Other Approaches

CHAPTER

8

BOARD GAMES

- 8.1 GAME THEORY
 - 8.1.1 Types of Games
 - 8.1.2 The Game Tree
- 8.2 MINIMAXING
 - 8.2.1 The Static Evaluation Function
 - 8.2.2 Minimaxing
 - 8.2.3 The Minimaxing Algorithm
 - 8.2.4 Negamaxing
 - 8.2.5 AB Pruning
 - 8.2.6 The AB Search Window
 - 8.2.7 Negascout
- 8.3 TRANSPOSITION TABLES AND MEMORY
 - 8.3.1 Hashing Game States
 - 8.3.2 What to Store in the Table
 - 8.3.3 Hash Table Implementation

XVI Contents

- 8.3.4 Replacement Strategies
- 8.3.5 A Complete Transposition Table
- 8.3.6 Transposition Table Issues
- 8.3.7 Using Opponent's Thinking Time
- 8.4 MEMORY-ENHANCED TEST ALGORITHMS
 - 8.4.1 Implementing Test
 - 8.4.2 The MTD Algorithm
 - 8.4.3 Pseudo-Code
- 8.5 OPENING BOOKS AND OTHER SET PLAYS
 - 8.5.1 Implementing an Opening Book
 - 8.5.2 Learning for Opening Books
 - 8.5.3 Set Play Books
- 8.6 FURTHER OPTIMIZATIONS
 - 8.6.1 Iterative Deepening
 - 8.6.2 Variable Depth Approaches
- 8.7 TURN-BASED STRATEGY GAMES
 - 8.7.1 Impossible Tree Size
 - 8.7.2 Real-Time AI in a Turn-Based Game

PART III

SUPPORTING TECHNOLOGIES

CHAPTER

EXECUTION MANAGEMENT

- 9.1 SCHEDULING
 - 9.1.1 The Scheduler
 - 9.1.2 Interruptible Processes
 - 9.1.3 Load-Balancing Scheduler
 - 9.1.4 Hierarchical Scheduling
 - 9.1.5 Priority Scheduling
- 9.2 ANYTIME ALGORITHMS
- 9.3 LEVEL OF DETAIL
 - 9.3.1 Graphics Level of Detail
 - 9.3.2 AILOD
 - 9.3.3 Scheduling LOD
 - 9.3.4 Behavioral LOD
 - 9.3.5 Group LOD
 - 9.3.6 In Summary

CHAPTER

10

WORLD INTERFACING

- 10.1 COMMUNICATION
- 10.2 GETTING KNOWLEDGE EFFICIENTLY
 - 10.2.1 Polling
 - 10.2.2 Events
 - 10.2.3 Determining What Approach to Use
- 10.3 EVENT MANAGERS
 - 10.3.1 Implementation
 - 10.3.2 Event Casting
 - 10.3.3 Inter-Agent Communication
- 10.4 POLLING STATIONS
 - 10.4.1 Pseudo-Code
 - 10.4.2 Performance
 - 10.4.3 Implementation Notes
 - 10.4.4 Abstract Polling
- 10.5 SENSE MANAGEMENT
 - 10.5.1 Faking It
 - 10.5.2 What Do I Know?
 - 10.5.3 Sensory Modalities
 - 10.5.4 Region Sense Manager
 - 10.5.5 Finite Element Model Sense Manager

CHAPTER

11

TOOLS AND CONTENT CREATION

- 11.0.1 Toolchains Limit AI
- 11.0.2 Where AI Knowledge Comes from
- 11.1 KNOWLEDGE FOR PATHFINDING AND WAYPOINT TACTICS
 - 11.1.1 Manually Creating Region Data
 - 11.1.2 Automatic Graph Creation
 - 11.1.3 Geometric Analysis
 - 11.1.4 Data Mining
- 11.2 KNOWLEDGE FOR MOVEMENT
 - 11.2.1 Obstacles
 - 11.2.2 High-Level Staging
- 11.3 KNOWLEDGE FOR DECISION MAKING
 - 11.3.1 Object Types
 - 11.3.2 Concrete Actions
- 11.4 THE TOOLCHAIN

- 11.4.1 Data-Driven Editors
- 11.4.2 AI Design Tools
- 11.4.3 Remote Debugging
- 11.4.4 Plug-Ins

PART IV

DESIGNING GAME AI

CHAPTER

12

DESIGNING GAME AI

- 12.1 THE DESIGN
 - 12.1.1 Example
 - 12.1.2 Evaluating the Behaviors
 - 12.1.3 Selecting Techniques
 - 12.1.4 The Scope of One Game
- 12.2 SHOOTERS
 - 12.2.1 Movement and Firing
 - 12.2.2 Decision Making
 - 12.2.3 Perception
 - 12.2.4 Pathfinding and Tactical AI
 - 12.2.5 Shooter-Like Games
- 12.3 DRIVING
 - 12.3.1 Movement
 - 12.3.2 Pathfinding and Tactical AI
 - 12.3.3 Driving-Like Games
- 12.4 REAL-TIME STRATEGY
 - 12.4.1 Pathfinding
 - 12.4.2 Group Movement
 - 12.4.3 Tactical and Strategic AI
 - 12.4.4 Decision Making
- 12.5 SPORTS
 - 12.5.1 Physics Prediction
 - 12.5.2 Playbooks and Content Creation
- 12.6 TURN-BASED STRATEGY GAMES
 - 12.6.1 Timing
 - 12.6.2 Helping the Player

CHAPTER

13

AI-BASED GAME GENRES

13.1	TEACHING CHARACTERS	81
13.1.1	Representing Actions	81
13.1.2	Representing the World	81
13.1.3	Learning Mechanism	81
13.1.4	Predictable Mental Models and Pathological States	82
13.2	FLOCKING AND HERDING GAMES	82
13.2.1	Making the Creatures	
13.2.2	Tuning Steering for Interactivity	
13.2.3	Steering Behavior Stability	
13.2.4	Ecosystem Design	

APPENDIX

REFERENCES

A.1 BOOKS, PERIODICALS, AND PAPERS

A.2 GAMES

INDEX