AI Game Development

GEORGIOS N. YANNAKAKIS, University of Edinburgh, U.K.

This is a review of the book*: Synthetic Creatures with Learning and Reactive Behaviors*. Alex J. Champandard, New Riders Publishing, 2004, 768 pp.

This book is the first attempt to bridge the current gap between artificial intelligence (AI) research in academia and computer-game development in industry. The book bridges the gap successfully. The author uses FEAR (Flexible Embodied Animat 'Rchitecture), which is an open-source project integrated with a commercial first-person shooter (FPS) game, to analyze, implement, test, and evaluate various AI approaches to various behavioral tasks for nonplayer characters (NPCs). The tasks start with low-level obstacle-avoidance and progress to high-level character emotions; each task with its own part in the book.**]** Each part is structured into chapters and follows a standard schema: the problem is analyzed, potential mechanisms (including AI) are discussed informally, details of the selected approach are outlined formally, an implementation is provided, the approach is evaluated (primarily visually) through simulation, and finally the advantages and disadvantages of the proposed approach are discussed.

The book's content and structure make it primarily suitable for game development, but it can also be used for didactic purposes, owing more to its practicality than to the theoretical background it covers. Most of the chapters end with demonstrations of the proposed prototypes, which are available from the book's web site. Source code implementation and demonstrations may very well provide suggestions for lab assignments. The quantity of mechanisms, the detailed description of open-source material, and the steps through the game development process are all helpful for game developers at all levels. However, researchers focusing on applications of AI techniques in computer games may find this book merely an extended overview of the area, since the quantity of AI methodologies described and implemented are at the expense of theoretical depth and extended literature review expected by researchers. In addition, this book is not suitable as a complete course text because there are too few chapters where the theoretical aspect of the AI approaches is detailed to the level required by lecturers.

Overall, the book's strong points are its careful consideration of problems that may arise in any phase of AI game development and the suggested solutions, which are outlined. Moreover, the game environment, the problem, and the case study assumptions in each part are analyzed in detail. On the other hand, weaknesses can be found in the endeavor to generalize AI methods on the basis of the platform of a single genre of computer games (i.e., first-person shooter). Case studies in this book cover a large set of

Author's address: School of Informatics, Centre for Intelligent Systems and their Applications, The University of Edinburgh, EH8 9LE, UK; email: g.yannakakis@sms.ed.ac.uk

nonplayer character tasks in computer games; but the applicability and efficiency of the proposed methodology to other genres of games are not clearly justified.

1. OVERVIEW

The debate about the general aspects of computer games and AI is presented In the first part of the book. It is noted that the graphic and sound aspects of computer games have now reached impressive levels, and therefore more attention is being focused on AI. After natural and artificial intelligence are defined, it is suggested that intelligent nonplayer characters are absolutely necessary to create the illusion of playing with or against other intelligent players. The fact that the current state of game AI is mostly A* pathfinding is also stressed, and questions of why and where in game development AI is necessary are raised. Even though the "AI revolution" is about to emerge due to increasing computational power and AI experience, various hurdles still appear. For instance, intelligent NPCs may challenge the authority of the game designer. As stressed in this part of the book, AI game development is about providing control to an NPC in order to produce entertainment, and hence the engineering perspective into AI game design is outlined in this part of the book also. The importance of the *animat* approach and embodiment in games is noted and a realistic (partial) *animat* perception of the game's virtual environment is suggested. At this point in the book, the reader has his or her first contact with so-called "nouvelle AI" techniques, like learning, including its types (offline, online) and its benefits when applied to computer games. The discussion of reactive versus planning techniques covers the next chapter. According to the author, reactive techniques are more appropriate for AI game development because they have the advantages of standard game AI (i.e., scripts, rule-based systems, finite-state machines) and embodiment and learning for animats. The last chapter of this part of the book attempts to combine all aforementioned techniques and approaches in a game platform. The AI testbed for this book is called the FEAR (Flexible Embodied Animat 'Rchitecture) project, which is an open-source project integrated with a commercial first-person shooter game. A comprehensive overview of the project, details about advanced topics, demonstrations, tools, and source code used in this book are available online from the website http://AiGameDev.com/.

2. MOVING AROUND

As noted in the second part of the book, the game world plays a very important role in the design of AI. More specifically,

- features such as dimensionality and precision in time and space (continuous or discrete) determine the type of the game environment. Navigating within such an environment constitutes a primitive behavior that is primarily based on the animat's perception of the environment. Partial sensing, which follows the rules of embodiment, may cause erroneous motion, notably in dynamic environments.

The book provides chapters that deal with fundamental phases of problem-solving and AI design. In the second part, the importance of the first two phases of *analysis* and *understanding* are outlined and adjusted to support AI game development. Given this theoretical background and the task being investigated (i.e., motion), the formulation of motion is discussed. Various options are sketched about the context (relative or absolute coordinates of discrete or continuous values), actions (discrete or continuous moves and turns), and senses (point content, lines traces, collision detection).

The proposed approach for obstacle-avoidance is based on ALife, and more specifically on the steering behavior approach of Reynolds [1987]. The advantages of this approach include simplicity, reliability, and efficiency, while the disadvantages are local trap behavior, lack of realism, and scalability. A proposed solution to overcome these disadvantages is the use of more knowledge through a rule-based system (RBS), and so an extensive overview of the rule-based systems' components and the details of the rule-based wall-following approach are presented. An evaluation of the approach demonstrates that RBSs are not particularly well suited to low-level control such as movement.

3. LEARN TO SHOOT!

This part of the book explores AI learning techniques and methodologies to generate efficient shooting skills for the NPCs. According to the author, shooting (in games) is a straightforward process enriched by noise (to appear more realistic if not more accurate), whereas gravity is added to only a few projectiles, such as grenades. First there is an overview of the basic features of the various weapon types that a player will meet in most FPS computer games, and then, judging from the way weapons are used in combat, a hierarchy is drawn of the skills required for shooting a moving target effectively.

At the top level of the hierarchy, as a consequence of *anticipation* and *ballistics,* we meet *prediction*. By using fundamental nongravity ballistics theory for the projectile and three observations, a moving object's position can be predicted by calculating its speed and its acceleration. Prediction of movement is given by an algorithm based on forward integration of Euler kinematics. Using this algorithm, game bots perform quite well, although they lack capabilities to predict patterns of enemy motion.

Additionally, at the top level of the shooting skills hierarchy we meet *aiming,* which involves *target selection*. For aiming skill, a perceptron trained offline is used to turn the animat's body smoothly towards a target, whereas online training, by gathering training examples, is used to increase the animat's effectiveness. An overview of perceptrons, the fundamentals of optimization in general, and perceptron- training algorithms is presented before the learning approach above is proposed. For the target-selection skill, a multi-layered neural network (NN) is trained offline (the training of sample data is based on enemy distance and damage caused by a rocket). A preceding chapter outlines the multi-layered NN theoretical background, which includes the NN biological correlation, training algorithms (backprop, Quickprop, Rprop), as well as the advantages and disadvantages of their use in general.

This part of the book concludes with a theoretical chapter on "Knowledge of the problem," a fundamental principle of AI development. This chapter also introduces in more detail ways to better understand black box problems and the informal, theoretical, statistical, and analytical levels of such a cognitive process**.**

4. CHOOSE YOUR WEAPON

The fourth part of this book commences with coverage of the different attributes of the weapons in computer games as well as a presentation of the environmental aspects that affect weapon selection. Weapon selection constitutes a decision-making problem, which humans solve based on either deduction or experience or a hybrid of the two. The features and criteria for making decisions about weapons are presented, as well as a comprehensive discussion of the appropriate inputs (environment state; player state; enemy damage, obtained indirectly from sound or visual effects) and outputs (actions to take). The first weapon-selection approach proposed is a voting system implemented in a scripting language; both the required background on scripting languages and a detailed description of voting systems are covered. However, the voting system approach uses theoretical assumptions and lots of expert knowledge, which can lead to undesirable behaviors. In an attempt to overcome this drawback, a variety of decision trees (DTs) are investigated. More specifically, the focus is on both classification and regression trees, their representation, how they are trained, validated, and tested. A DT is used as a second approach for learning weapon selection. The proposed methodology is based on learning a fitness function assigned for each weapon. The animats generated are quite efficient and well behaved but not optimal because they have no sense of survival or desire to score points or even any idea of the purpose of weapon selection itself. This section also concludes with a theoretical, or "lesson," chapter. Topics covered in this chapter — which provides insights for understanding the solution — include information theory for the description of AI solutions, the problem's search space, and various approaches to obtain a solution (expert, expert guidance, exhaustive, random, hybrid.

5. USING ITEMS AND OBJECTS

Part five of the book devotes a chapter to describing objects (items and contraptions) in the game environment and discusses the human approach to collecting items (almost sub-consciously) and using contraptions. According to the author, an AI solution for this problem should be based primarily on criteria such as effectiveness and justifiability. At this point, the reader is introduced to fuzzy set theory and how fuzzy expert systems are used to make decisions.

Fuzzy control is the AI approach proposed for animats to deal with doors, ladders, and elevators (contraptions). A specification of the fuzzy variables and rules for solving the problems of climbing ladders, using doors, and taking platforms is presented. An evaluation of this approach shows smoothness in the animats' movements and effectiveness in their overall behavior. However, fuzzy rules involve heavy specification of behavior (the rules are usually better at dealing with decision-making than with control).

Methodologies based on Darwinian evolution can overcome the aforementioned disadvantage of fuzzy approaches. There is an extended overview of genetic algorithms (including, biological evolution, representation, genetic operators, and the basic steps of the algorithm) and learning classifier systems. For the latter, a modified Wilson algorithm [Wilson 1994] is presented. This material is used as background knowledge for the proposed approach to learning dodgingfire and rocket-jumping behaviors via use of a genetic algorithm. The behaviors due to this approach proved quite successful.

As in the rest of the book, this part concludes with a theoretical chapter on AI learning. Sections of this chapter cover the reasons for learning, approaches to learning (offline — batch; online — incremental), learning types (supervised, reinforcement, evolutionary, unsupervised), and methodologies for learning (training, imitation, trial and error, shaping). As stressed in this last chapter, the learning methodology recommended for computer games is supervised offline, since this produces the most predictable results.

6. EMOTIONS

This part is devoted to emotions and how to represent them artificially, in order to improve the believability of the characters. Definitions of emotions, sensations, and moods (by means of expressions, gestures, behaviors, voice, and language), and the role of emotions in research on human-computer interaction and the development of games are provided

The primary argument for using emotions in games is that they define humans, and are therefore a key factor in providing realism and believability in computer games. The proposed approach to generate emotions is based on finite state machines (FSMs). A chapter is devoted to covering the fundamentals of finite state automata and machines (Mealy and Moore) for AI game development and their positive aspects (they are simple; easy to use, implement, visualize, design, debug, and work with) and their negative ones (complexity; no online modification; very predictable output, and unsuitability for creating behaviors). According to the emotional approach, an animat is designed to respond according to its moods via an FSM. Although the emotional model proposed is simple to adjust, it generates predictable emotions that remain the same throughout the game. An outline of nondeterministic, probabilistic (Markov model), and fuzzy state machines and their pluses and minuses in the development of computer games is covered next. Hierarchical state machines (HSMs) are presented as well, and their interaction semantics are discussed in detail. As emphasized in the book, hierarchical finite state machines, due primarily to their flexibility, push FSMs to give their best performance. However, graphical tools are required for the design of the hierarchy, and the interaction between states must be carefully defined.

Given the background on state machines mentioned above, the following chapter provides an improved emotional system based on fuzzy, probabilistic, and nondeterministic FSMs. According to this system, memories are obtained by statistics; feelings are modeled with a nondeterministic automaton; sensations are represented as fuzzy automata; emotions are fuzzy state machines; mannerisms are picked by nested probabilistic automata; and moods are modeled with nested states. The evaluation of the system demonstrates a significant improvement in the animat's realistic representation, which is primarily affected by the addition of mannerisms in the emotional model.

The last chapter in this part of the book covers the issue of emergence, how to achieve it (e.g., through evolutionary design and agile development practices) and how computer games can benefit from emergent behaviors.

7. ACTION SELECTION

The focus of this part of the book is on analyzing death-match situations in the context of decision making and on attempts to develop learning algorithms capable of producing adaptive behaviors. The first prototype introduced to create death-match behaviors was designed manually, based on knowledge of tactical behavior gained through decision--making analysis. This prototype follows Brooks' subsumption architecture [Brooks 1991], and even though it performs quite well, it has obvious limitations, such as predictability, a narrow set of behaviors, and a lack of learning capability.

The theory behind reinforcement learning (RL) is introduced, various algorithms (dynamic programming, Monte Carlo, temporal difference) appropriate for game development are described, and finally the advantages and disadvantages of RL in the context of AI in games are discussed. The problem of the animat's tactical behavior is split into components and RL is applied to each one of them separately. More analytically, learning gathering behaviors is achieved via a statistical estimate of the reward values where decision depends on the animat's mood. *Q*learning is used for effective decision making on the type of movement, with rewards based on the animat's moods and killing performance. Finally, an episodic (i.e., the duration of a fight) learning algorithm (i.e., Monte Carlo) is employed for adaptive weapon selection where the reward is primarily determined by the moods of the animat. Other behaviors are not learned, since there are simpler programming techniques in the book that can generate them effectively.

Overall, the emergent animat behaviors demonstrate the positive impact of learning on performance, and the proposed learning mechanisms were able to adapt to the trends in the game. However, as noted in the book, adaptation may give rise to many problems, so learning mechanisms should be designed carefully, especially when it comes to online learning (see also Yannakakis and Hallam [2004]). The final chapter in this part of the book focuses on these problems and proposes potential solutions regarding adaptive behaviors in software engineering, system modeling, and adaptation management.

8. SUMMARY

The book's last part summarizes the ways in which games may be developed more efficiently, like hints for faster implementation, solutions to common problems concerning AI techniques and their applicability, and finally hints for choosing the appropriate learning mechanisms and feedback functions.

The book concludes with three main directions for AI in games: (a) further experimentation on combining numerous AI techniques and various behavioral tasks; (b) creation of game-world models based on memory; and (c) planning techniques as the alternative to the reactive approaches discussed extensively in this book.

9. CONCLUSIONS

AI Game Development is easy to read; only a shallow background in computer science and C++ programming is needed for the open-source code experimentation. Since it is the first book released on the use of advanced AI tools in computer games, it certainly constitutes an ideal script for game developers, practitioners, and students interested in learning about this exciting new field.

REFERENCES

BROOKS, R. A. 1991. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (San Mateo, CA, 1991). 569–595.

REYNOLDS, C. W. 1987. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics 21*, 4 (1987), 25–34.

WILSON, S. W. 1994. ZCS: A zeroth level classifier system. In *Evolutionary Computation.* 1994.

YANNAKAKIS, G. N. AND HALLAM, J. 2004. Evolving opponents for interesting interactive computer games. In *From Animals to Animats 8: Proceedings of the 8th International Conference on Simulation of Adaptive Behavior* (SAB04). S. Schaal et al., eds. 2004. 499–508.

Received January 2005; accepted January 2005