

Preface

Why would anyone want to make a computer game intelligent? What motivation is there for putting **intelligence** into perfectly good games? After all, humans have played games since prehistory and we have never had the ability to create intelligence in our games until now. This, of course, overlooks the fact that we typically played games against other human beings (most board games and sports) or against ourselves (often memory based games), sometimes with a statistical element involved (such as the solo card games of ‘patience’). Most people would agree that the most satisfying is playing games against other people, particularly when the other people are of the same standard as ourselves. There is something inherently satisfying about competition with our peers. We do not get the same satisfaction competing against, for example, children who do not yet have our capabilities. Similarly we get no fun at all playing against an opponent who is perceived to be much better than us at a particular game. Thus we want to create games in which the opponent is, in some way, a worthy opponent for us. The computer promises to liberate us from the need to find such opponents by providing us with artificial opponents of a suitable ability. Thus our computer games commonly come not just with the game itself but with built-in artificial opponents who aspire to being worthy opponents for us.

Even in the realm of online multiplayer gaming, nonplayer characters (NPCs) are a required staple of many games, such as in most massively-multiplayer online games (MMOG), where NPCs may be required to provide missions or combat encounters. Here the contrast between intelligent players and nonintelligent computer controlled characters is even sharper, and clearly an area where more machine intelligence could be applied to improve the game for all the players.

This book is concerned with the application of modern artificial intelligence (AI) techniques to computer games. We suggest that the current generation of computer games is largely free of this technology. You may well protest that AI is already well established in such games; after all the computer opponent in such games not always known as the ‘AI.’ But we highlight the use of the word ‘modern.’ Most, though by no means all, games of recent and current generations use fairly old and certainly rather standard forms of artificial intelligence. In this book we will concentrate on technologies which are still the subject of large scale effort in the research field, such as artificial neural networks, genetic algorithms, or artificial immune systems. We shall not discuss, for example, case based reasoning since we concentrate solely on techniques which are based on those thought to dominate in real life.

So what is it that these technologies bring to computer games that finite state machines, expert systems, and the rest do not bring? The unifying theme of these techniques is adaptation, whether as a means to learn how to play a game or even while the game is being played. In artificial neural networks we are changing the network weights—corresponding to the synaptic efficiencies—so that the response from the artificial neural network changes from game to game. In evolutionary algorithms, the artificial chromosomes are being selected base on their fitness so that better solutions are being found as the game proceeds; in artificial immune systems, our artificial antibodies are being produced in response to attacks by artificial antigens which allows the artificial immune system to adapt to varying threats.

Why do we consider adaptation to be so central to creating intelligence in computer games? We feel that most people playing against an AI wish to feel that they are playing against another intelligent person. When two people play a competitive game, each uses a continually changing strategy: if you know your opponent's strategy in advance, you can prepare a defence specific to such a strategy. Even in completely nonrandom games such as chess, no player will use exactly the same opening or defence in each game. To do so would leave an opponent with a very simplified problem and would certainly result in long-term defeat. This is not to say that we will not have a set of preferred strategies in a game but the emphasis must be on having a nonsingular set, not just a single strategy.

There is one other point which may be made with respect to these biologically inspired AI techniques. Competition (for food, for shelter, for mates) is built into life itself and so we consider that using analogies in software of real biological processes means that we are basing our techniques on methodologies which have already proved successful in real life. Thus we may anticipate that the resulting techniques match in some way the experiences we already have as human beings in our existing environments.

By focussing on biologically-inspired AI techniques, rather than just those methods commonly used in the games industry, some sections of this book may appear to be more theoretical than applied. A number of case-studies are scattered throughout—either as chapters in their own right, or included into larger chapters—to offset this. These vary from simple illustrative examples, to case studies in contemporary action and racing games.

Origin of this Book

This book arose from our experience of teaching on a range of modules related to computer game development, including modules specifically focussed on game AI. We found that there was no existing text which satisfied our desired pedagogical aims. Many textbooks on game AI tend to be devoted to state space methods and efficient searches. Most present algorithms and solutions, but in concentrating on showing how to achieve results, they provide little in the way of theoretical underpinnings. A notable exception is that field which is devoted to evolutionary game theory; however, this field is much more based on mathematical game theory which is outside the interests of most programmers who are investigating computer games.

The research background of several of the authors is in the broad area of artificial neural networks (ANN), and this accordingly features in several of the chapters. It was our intention, however, to present a book which presented a more complete, yet still academic, view

of game AI. We have also been able to draw upon our experience from research in a range of related areas, from artificial life and genetic algorithms to artificial immune systems and ant-colony optimisation.

Composition of this Book

In Chapter I, we present a brief review the history of computer games, from the 1950s to the present day. Needless to say the rate of change over the last decade has been much greater than that seen in earlier times and this acceleration seems likely to continue but we have tried to do justice to the major trends in existence today. The goal of this chapter is to more clearly set out what the common goals of game AI are, and to explain why much of modern game AI is not of academic interest.

In Chapter II, we provide a broad introduction to natural and artificial neural networks, including notes on the history of the development of modern artificial neural networks. Then, in Chapter III, we introduce the technique which, to many people, *is* artificial neural networks, the multilayered perceptron, or MLP. This well known technique uses the back-propagation of errors to learn. We give the theory behind the learning and some heuristics about its performance before illustrating its use on some general data problems. We then illustrate how it may be used in a game context. In Chapter IV, we provide a short case study on the use of artificial neural networks—particularly the multilayered perceptron—in digital games, using the Robocode competition framework as a suitable example.

The artificial neural networks in Chapters III and IV use supervised learning to adapt; the training set must include a set of answers appropriate to the problems in hand. However, a lot of human and animal learning occurs without a teacher. In particular, when playing a game we may learn by creating internal models of the game as it is being played, without a teacher taking on the role of a ‘learning supervisor.’ Thus in Chapter V, we discuss two types of artificial neural networks which organise according to unsupervised learning principles. We differentiate between competitive learning, a variation of which leads to self-organising maps, and Hebbian learning; the former tends to be used for clustering a data set while the latter leads to low dimensional projections of a data set. The self-organising map is used to develop AI opponents in the rather old game of Pong while the projection methods are illustrated in a high dimensional hide-and-seek game.

We stated that, for many people, artificial neural networks are equated with the multilayered perceptron, however the multilayered perceptron is actually only one network in a rather large field of supervised learning methods. Therefore in Chapter VI, we introduce a further alternative which can be used instead of the multilayered perceptron. We pay particular emphasis on the radial basis network since its speed of learning makes it extremely attractive in the context of computer games. Returning to Pong, we are then able to compare the in-game performance of different artificial neural networks in creating players for this simple game.

In Chapter VII, we switch to evolutionary algorithms, the most famous of which is the genetic algorithm. We spend some time examining issues of representation before illustrating its use on a computer game. The case study presented in this chapter, a motocross game with realistic physics, is a particularly challenging problem for AI to solve; the solution

presented combining the power of both evolutionary and artificial neural net approaches. The games industry's traditional solution to this problem has been to use designer intelligence rather than machine intelligence and have the developers implement the paths to be followed. This work clearly demonstrates that contemporary games can potentially benefit from biologically inspired AI.

In Chapter VIII we explore just some of the many ways in which the standard genetic algorithm can be extended, and look at a few alternative evolutionary methods. Deceptive problems are introduced before we consider the first extension of the structured GA. Other alternatives introduced include probability based incremental learning (PBIL), which is also demonstrated through an example of the N-Persons Iterated Prisoners' Dilemma.

Chapter XI introduces the basic concept of multiobjective problems, with particular relation to the genetic algorithm. It then considers how genetic algorithms and evolutionary methods might be applied in the real-time strategy genre of computer games. In such games, AI has to work at a number of distinct levels, roughly speaking covering the strategic goals and the low level tactics. Exploring the full set of strategies and finding all the good solutions in such situations is far from straightforward, and this chapter illustrates some of the issues.

In Chapter X, we discuss artificial immune systems and show how our resulting models can be used in a variety of computer games. Since this topic is rather newer than the other AI techniques, we spend some time developing the artificial immune systems in a variety of game situations.

Two further, very distinct, biologically inspired AI techniques are introduced in the following two chapters. In Chapter XI ant-colony optimisation (ACO) is demonstrated as an alternative method of path-planning. An example application, the classic game of Combat is presented.

The following chapter, Chapter XII, outlines the methods used in reinforcement learning. Although the term 'reinforcement learning' can be used more generally to describe the learning methods used in a variety of AI techniques, it is also the name of a particular approach and set of algorithms for AI learning. The reinforcement learning approach is one based on the payment of rewards for actions leading to good results, such as winning a game. This simple notion forms the basis for a group of methods which at their core resemble human game playing approaches in a number of ways, although which have yet to be used to any extent in the commercial world of game development.

We close with two chapters which focus less on particular AI methods, but on issues relating to their use and misuse. Chapter XIII discusses adaptivity in games more generally. Current approaches to adaptivity in commercial games are discussed, along with issues facing developers wishing to introduce and develop adaptive solutions. It is our feeling that most attempts to include adaptivity in games have been (often deliberately) limited, and that there is a great deal of potential in making games more adaptive in a range of areas.

Finally in Chapter XIV, we assess how successful our attempt to install human-like intelligence in computer games has been using methods suggested by the famous Turing Test. It is of interest that Turing expected computers to pass his test by the year 2000. We believe that success in the game version of the Turing Test is more reachable in the near future than success in the standard Turing Test.

The book need not be read in any particular order and the chapters are intended to be self-contained, the exception being that Chapter III should be read before any of the other chapters on artificial neural networks. Thus we have also provided references at the end of each

separate chapter rather than a single section at the end of the book. We hope that you will enjoy the book, that it will stimulate you to experiment with the AI techniques discussed, and that you will have as much fun playing with these techniques in your games as we have had while writing this book.

Trademarks

Naturally, in a book which discusses game AI, mention is made of a large number of commercial computer games, as well as a few films and other media outputs. All such trademarks and copyrights are property of their respective owners.