Available online at www.sciencedirect.com

SCIENCE DIRECT®

BioSystems 85 (2006) 72–83

Bio Systems

www.elsevier.com/locate/biosystems

A platform for evolving intelligently interactive adversaries

David B. Fogel[a,*], Timothy J. Hays[a], Douglas R. Johnson[b,1]

[a] Natural Selection, Inc., 3333 N. Torrey Pines Ct., Suite 200, La Jolla, CA 92037, United States

[b] Digenetics, Inc., 3333 N. Torrey Pines Ct., Suite 200, La Jolla, CA 92037, United States

Received 28 February 2006; accepted 28 February 2006

Abstract

Entertainment software developers face significant challenges in designing games with broad appeal. One of the challenges concerns creating nonplayer (computer-controlled) characters that can adapt their behavior in light of the current and prospective situation, possibly emulating human behaviors. This adaptation should be inherently novel, unrepeatable, yet within the bounds of realism. Evolutionary algorithms provide a suitable method for generating such behaviors. This paper provides background on the entertainment software industry, and details a prior and current effort to create a platform for evolving nonplayer characters with genetic and behavioral traits within a World War I combat flight simulator.

Keywords: Evolutionary algorithms; Evolutionary computation; Neural networks; Object neural networks; Video games; Agent-based; Nonplayer characters; Flight simulator

1. Introduction

Entertainment software developers face two significant challenges in designing games with broad appeal. One challenge is to reduce the time and cost for development. More time spent in development translates into higher costs for programmers, software testers, and time lost in production instead of time spent on store shelves. Another challenge is to ensure that each game has a long "half-life," meaning that players do not quickly lose interest in the game. Software publishers know that over 30% of their sales come from word-of-mouth. Only a few years ago, Credit Suisse First Boston (2002) noted that 45% of surveyed respondents indicated that recommendations from friends and relatives was the most effective means of information flow regarding games, scoring highest among all forms of communication, including television, Internet, and radio. Boring games do not garner word-of-mouth and are discounted for cheaper sale, making way for new products. Software developers need to increase their games' longevity, extending their useful lifespan to years, ensuring higher product prices and also maintaining customer loyalty.

Neither of these challenges is being met adequately. The current costs for game development now top $6 million for consoles (e.g., PlayStation2, as found in editorials in PC Week (2002) and Global Media (2003)), and typically run from $1 million to $4 million or more for PC titles, with additional similar costs for conversions to other consoles (e.g., Xbox and the new Xbox 360). For comparison, Shiny Entertainment's "Enter the Matrix" cost in excess of $20 million to develop (Loftos, 2003), but companies can expect to spend from $2 million to $7.5 million to develop a game on multiple platforms, often over a 2-year time frame. Companies attempt

* Corresponding author. Tel.: +1 858 455 6449; fax: +1 858 455 1560.

E-mail addresses: dfogel@natural-selection.com (D.B. Fogel), timhays@natural-selection.com (T.J. Hays), drjohnson@digenetics.com (D.R. Johnson).

[1] Tel.: +1 858 455 6449, fax: +1 858 455 1560.

0303-2647/$ – see front matter
doi:10.1016/j.biosystems.2006.02.010

to reduce the production time by employing reusable graphics engines that support three-dimensional character movement, as well as backgrounds, and other artwork. But game-play design and testing remain a weak point, as these efforts are performed by hand, requiring a team of programmers and play-testers.

For example, the popular game *Age of Empires II* (which led to other games such as *Age of Mythology* and *Age of Empires III*), published by Ensemble Studios through Microsoft, offers the player different choices of “civilizations.” The game developers wanted to ensure that no civilization had an inherent advantage. They therefore play tested, by hand, all different possible combinations (about 90) of civilizations, making adjustments to the parameters of the types of units in the civilizations (e.g., strength of a man-at-arms or accuracy of a crossbowman). A handbook published with the game indicated that a team of at least five people spent nine months in this play testing. This is a very costly approach – likely in excess of $250,000 (perhaps much more) and a missed opportunity of earlier sales – and indicates a significant opportunity to address a market need.

Software developers attempt to create characters that will entertain by being persistently competitive and by generating novel behaviors. Characters are programmed to obey specific rules that govern their behavior. Thus the quality of this programmed behavior depends entirely on the skill of the programmers in both novelty and acumen. Despite efforts to write rules to cover every imaginable circumstance, the result is typically weak. The failure of artificial intelligence methods to really create novel high-quality behaviors is legendary in the entertainment software business (Nakamoto, 2001). Programmers cannot conceive of everything that might happen in a game, and when unanticipated situations arise, programmed characters often act in ways that no human opponent would consider rational. The main way that software developers have achieved any measure of success in overcoming this is by iterative play testing, finding these weaknesses, and writing more rules to cover them. Ultimately, this patchwork method fails, and games are released even though the developer knows that the game characters will act contrived and will not maintain the player’s interest for very long.

In parallel to game development, research into machine learning in games has been advancing rapidly. Starting from perhaps the earliest efforts of Samuel (1959) with checkers (also known as draughts), research has been conducted in various forms of computational intelligence (e.g., neural networks and evolutionary computation) for a wide variety of *n*-player zero-sum and nonzero-sum games. A complete review of these efforts is beyond the scope of this paper; however the breadth of study can be appreciated by reviewing the numerous contributions to games research using the iterated prisoner’s dilemma (e.g., Axelrod, 1987; Fogel, 1993; Fogel, 1995; Harrald and Fogel, 1996; Darwen and Yao, 2000; Chong and Yao, 2004; Chong and Yao, 2005; Ishibuchi and Namikawa, 2005; Franken and Engelbrecht, 2005 and others), general game theory and evolutionary stable strategies (Fogel et al., 1997; Fogel et al., 1998; Fogel and Beyer, 2000; Ficici et al., 2005 and others), board games such as checkers (Chellapilla and Fogel, 1999a,b, 2001; Fogel, 2002; Hughes, 2003; Franken and Engelbrecht, 2003; Kim and Cho, 2003, 2005; Hughes, 2005), chess (Kendall and Whitwell, 2001; Fogel et al., 2004a,b), Othello (Moriarty and Miikkulainen, 1995), backgammon (Pollack and Blair, 1998; Darwen, 2001), versions of Go (Richards et al., 1998, 2001; Kendall et al., 2004; Runnarson and Lucas, 2005; Lubberts and Miikkulainen, 2001), RISK (Vaccaro and Guest, 2005), Monopoly (Frayn, 2005), and other games such as core wars (Corno et al., 2003, 2004; Corno et al., 2005a,b), card games (Kendall and Smith, 2003; Fogel, 2004), combat and other video games (Gallagher and Ryan, 2003; Stanley et al., 2005; Louis and Miles, 2005; Hong and Cho, 2005; Tanev et al., 2005; Togelius and Lucas, 2005; Lucas, 2005; Parker et al., 2005), and many others (see also Kendall and Lucas, 2005; Fogel et al., 2005).

Within the last 5 years, attention has been given increasingly to games in which the player controls characters that develop during play. Some of these games provide players with characters that have both observable traits and underlying genetic characteristics. As such, these games are well suited for evolutionary computing to provide a means for generating intelligently interactive behaviors, as well as allowing players to clone, mutate, and breed characters. Some specific examples are discussed in the next section.

2. Background

There are two main target markets for applying evolutionary computing to commercial games: the individual consumer of video games and the software developer/publisher. Each has unique desires and clearly identified needs.

Game players want continually novel and fun gaming experiences. They spend an average of 3.7 days per week playing games with an average of 2.01 h per day (Credit Suisse First Boston, 2002). Maintaining this level of involvement requires continual novelty. In 2002, U.S. households were polled to assess the appeal of alternative game features. The greatest response (50%) was seen in