



2009-01-01

Agent Based Modeling in Computer Graphics and Games

Brian MacNamee

Dublin Institute of Technology, brian.macnamee@dit.ie

Follow this and additional works at: <http://arrow.dit.ie/scschcombk>

 Part of the [Computational Engineering Commons](#), and the [Robotics Commons](#)

Recommended Citation

Mac Namee, B. (2009) Agent Based Modeling in Computer Graphics and Games. In R.A.Meyers (ed.) *Encyclopedia of Complexity and Systems Science*, Springer. doi:10.1007/978-1-4614-1800-9_39

This Book Chapter is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Books/Book Chapters by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.



This work is licensed under a [Creative Commons Attribution-NonCommercial-Share Alike 3.0 License](#)



School of Computing

Other resources

Dublin Institute of Technology

Year 2009

Agent Based Modeling in Computer Graphics and Games

Brian Mac Namee
Dublin Institute of Technology, brian.macnamee@dit.ie

This paper is posted at ARROW@DIT.
<http://arrow.dit.ie/scschcomoth/1>

— Use Licence —

Attribution-NonCommercial-ShareAlike 1.0

You are free:

- to copy, distribute, display, and perform the work
- to make derivative works

Under the following conditions:

- Attribution.
You must give the original author credit.
- Non-Commercial.
You may not use this work for commercial purposes.
- Share Alike.
If you alter, transform, or build upon this work, you may distribute the resulting work only under a license identical to this one.

For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the author.

Your fair use and other rights are in no way affected by the above.

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit:

- URL (human-readable summary):
<http://creativecommons.org/licenses/by-nc-sa/1.0/>
 - URL (legal code):
<http://creativecommons.org/worldwide/uk/translated-license>
-

Agent-Based Modelling in Computer Graphics and Games

1 BRIAN MAC NAMEE
2 School of Computing, Dublin Institute of Technology,
3 Dublin, Ireland

4 Article Outline

5 Glossary
6 Definition of the Subject
7 Introduction
8 Agent-Based Modelling in Computer Graphics
9 Agent-Based Modelling in CGI for Movies
10 Agent-Based Modelling in Games
11 Future Directions
12 Bibliography

13 Glossary

14 **Computer generated imagery (CGI)** The use of com-
15 puter generated images for special effects purposes in
16 film production.

17 **Intelligent agent** A hardware or (more usually) software-
18 based computer system that enjoys the properties au-
19 tonomy, social ability, reactivity and pro-activeness.

20 **Non-player character (NPC)** A computer controlled
21 character in a computer game – as opposed to a player
22 controlled character.

23 **Virtual character** A computer generated character that
24 populates a virtual world.

25 **Virtual world** A computer generated world in which
26 places, objects and people are represented as graphical
27 (typically three dimensional) models.

28 Definition of the Subject

29 As graphics technology has improved in recent years,
30 more and more importance has been placed on the behav-
31 ior of virtual characters in applications set in virtual worlds
32 in areas such as games, movies and simulations. The behav-
33 ior of virtual characters should be believable in order
34 to create the illusion that these virtual worlds are popu-
35 lated with living characters. This has led to the applica-
36 tion of agent-based modeling to the control of these vir-
37 tual characters. There are a number of advantages of using
38 agent-based modeling techniques which include the fact
39 that they remove the requirement for hand controlling all
40 agents in a virtual environment, and allow agents in games
41 to respond to unexpected actions by players.

Introduction

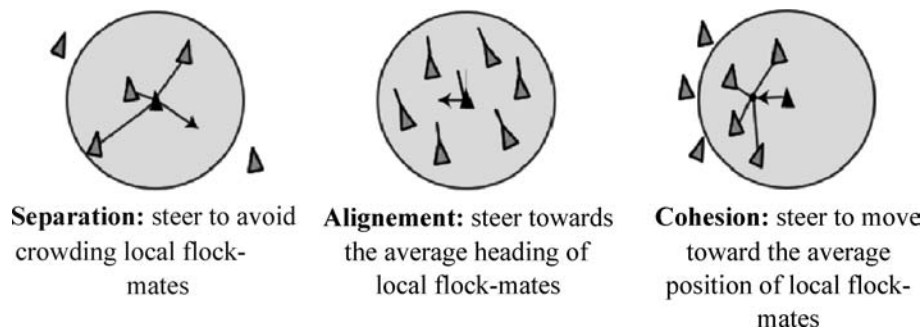
Advances in computer graphics technology in recent years
have allowed the creation of realistic and believable vir-
tual worlds. However, as such virtual worlds have been de-
veloped for applications spanning games, education and
movies it has become apparent that in order to achieve
real believability, virtual worlds must be populated with
life-like virtual characters. This is where the application of
agent-based modeling has found a niche in the areas of
computer graphics and, in a huge way, computer games.
Agent-based modeling is a perfect solution to the prob-
lem of controlling the behaviors of the virtual characters
that populate a virtual world. In fact, because virtual char-
acters are embodied autonomous agents these applications
require an even stronger notion of agency than many other
areas in which agent-based modeling is employed.

Before proceeding any further, and because there are
so many competing alternatives, it is worth explicitly stat-
ing the definition of an intelligent agent that will inform
the remainder of this article. Taken from [83] an intelli-
gent agent is defined as “... a hardware or (more usually)
software-based computer system that enjoys the following
properties:

- **autonomy:** agents operate without the direct interven-
tion of humans or others, and have some kind of control
over their actions and internal state;
- **social ability:** agents interact with other agents (and
possibly humans) via some kind of agent-communication
language;
- **reactivity:** agents perceive their environment, (which
may be the physical world, a user via a graphical user
interface, a collection of other agents, the INTERNET, or
perhaps all of these combined), and respond in a timely
fashion to changes that occur in it;
- **pro-activeness:** agents do not simply act in response to
their environment, they are able to exhibit goal-directed
behavior by taking the initiative.”

Virtual characters implemented using agent-based mod-
eling techniques satisfy all of these properties. The char-
acters that populate virtual worlds should be fully au-
tonomous and drive their own behaviors (albeit some-
times following the orders of a director or player). Vir-
tual characters should be able to interact believably with
other characters and human participants. This property is
particularly strong in the case of virtual characters used
in games which by their nature are particularly interac-
tive. It is also imperative that virtual characters appear to
perceive their environments and react to events that oc-
cur in that environment, especially the actions of other

Please note that the pagination is not final; in the print version an entry will in general not start on a new page.



Agent-Based Modelling in Computer Graphics and Games, Figure 1

The three rules used by Reynolds' original Boids system to simulate flocking behaviors

91 characters or human participants. Finally virtual characters should be pro-active in their behaviors and not always require prompting from a human participant in order to take action.

95 The remainder of this article will proceed as follows. 96 Firstly, a broad overview of the use of agent-based modeling in computer graphics will be given, focusing in particular on the genesis of the field. Following on from this, the focus will switch to the use of agent-based modeling techniques in two particular application areas: computer generated imagery (CGI) for movies, and computer games. CGI has been used to astounding effect in movies for decades, and in recent times has become heavily reliant on agent-based modeling techniques in order to generate CGI scenes containing large numbers of computer generated extras. Computer games developers have also been using agent-based modeling techniques effectively for some time now for the control of non-player characters in games. There is a particularly fine match between the requirements of computer games and agent-based modeling due to the high levels of interactivity required.

112 Finally, the article will conclude with some suggestions for the future directions in which agent-based modeling technology in computer graphics and games is expected to move.

116 Agent-Based Modelling in Computer Graphics

117 The serious use of agent-based modeling in computer graphics first arose in the creation of autonomous groups and crowds – for example, crowds of people in a town square or hotel foyer, or flocks of birds in an outdoor scene. While initially this work was driven by visually unappealing simulation applications such as fire safety testing for buildings [75] focus soon turned to the creation of visually realistic and believable crowds for applications such as movies, games and architectural walk-

throughs. Computer graphics researchers realized that creating scenes featuring large virtual crowds by hand (a task that was becoming important for the applications already mentioned) was laborious and time-consuming and that agent-based modeling techniques could remove some of the animator's burden. Rather than requiring that animators hand-craft all of the movements of a crowd, agent-based systems could be created in which each character in a crowd (or flock, or swarm) could drive its own behavior. In this way the behavior of a crowd would emerge from the individual actions of the members of that crowd.

Two of the earliest, and seminal, examples of such systems are Craig Reynolds' Boids system [64] and Tu & Terzopoulos' animations of virtual fish [76]. The Boids system simulates the flocking behaviors exhibited in nature by schools of fish, or flocks of birds. The system was first presented at the prestigious SIGGRAPH conference (www.siggraph.org) in 1987 and was accompanied by the short movie "Stanley and Stella in: Breaking the Ice". Taking influence from the area of artificial life (or aLife) [52], Reynolds postulated that the individual members of a flock would not be capable of complex reasoning, and so flocking behavior must emerge from simple decisions made by individual flock members. This notion of emergent behavior is one of the key characteristics of aLife systems.

In the original Boids system, each virtual agent (represented as a simple particle and known as a *boi*d) used just three rules to control its movement. These were separation, alignment and cohesion and are illustrated in Fig. 1. Based on just these three simple rules extremely realistic flocking behaviors emerged. This freed animators from the laborious task of hand-scripting the behavior of each creature within the flock and perfectly demonstrates the advantage offered by agent-based modeling techniques for this kind of application.

The system created by Tu and Terzopoulos took a more complex approach in that they created complex models of biological fish. Their models took into account fish physiology, with a complex model of fish muscular structure, along with a perceptual model of fish vision. Using these they created sophisticated simulations in which properties such as schooling and predator avoidance were displayed. The advantage of this approach was that it was possible to create unique, unscripted, realistic simulations without the intervention of human animators. Terzopoulos has since gone on to apply similar techniques to the control of virtual humans [68].

Moving from animals to crowds of virtual humans, the Virtual Reality Lab at the Ecole Polytechnique Fédérale de Lausanne in Switzerland (vrlab.epfl.ch) led by Daniel Thalmann has been at the forefront of this work for many years. They group currently has a highly evolved system, ViCrowd, for the animation of virtual crowds [62] which they model as a hierarchy which moves from individuals to groups to crowds. This hierarchy is used to avoid some of the complications which arise from trying to model large crowds in real time – one of the key goals of ViCrowd.

Each of the levels in the ViCrowd hierarchy can be modeled as an agent and this is done based on *beliefs*, *desires* and *intentions*. The beliefs of an agent represent the information that the agent possesses about the world, including information about places, objects and other agents. An agent's desires represent the motivations of the agent regarding objectives it would like to achieve. Finally, the intentions of an agent represent the actions that an agent has chosen to pursue. The belief-desire-intention (BDI) model of agency was proposed by Rao and Georgeff [61] and has been used in many other application areas of agent-based modeling.

ViCrowd has been used in ambitious applications including the simulation of a virtual city comprised of, amongst other things, a train station a park and a theater [22]. In all of these environments the system was capable of driving the believable behaviors of large groups of characters in real-time.

It should be apparent to readers from the examples given thus that the use of agent-based modeling techniques to control virtual characters gives rise to a range of unique requirements when compared to the use of agent modeling in other application areas. The key to understanding these is to realize that the goal in designing agents for the control of virtual characters is typically not to design the most efficient or effective agent, but rather to design the most interesting or believable character. Outside of very practical applications such as evacuation simulations, when creating virtual characters, designers are con-

cerned with maintaining what Disney, experts in this field, refer to the *illusion of life* [36].

This refers to the fact that the user of a system must believe that virtual characters are living, breathing creatures with goals, beliefs, desires, and, essentially, lives of their own. Thus, it is not so important for a virtual human to always choose the most efficient or cost effective option available to it, but rather to always choose reasonable actions and respond realistically to the success or failure of these actions. With this in mind, and following a similar discussion given in [32], some of the foremost researchers in virtual character research have the following to say about the requirements of agents as virtual characters.

Loyall writes [46] that “*Believable agents are personality-rich autonomous agents with the powerful properties of characters from the arts.*” Coming from a dramatic background it is not surprising that Loyall's requirements reflect this. Agents should have strong personality and be capable of showing emotion and engaging in meaningful social relationships.

According to Blumberg [11], “... *an autonomous animated creature is an animated object capable of goal-directed and time-varying behavior*”. The work of Blumberg and his group is very much concerned with virtual creatures, rather than humans in particular, and his requirements reflect this. Creatures must appear to make choices which improve their situation and display sophisticated and individualistic movements.

Hayes–Roth and Doyle focus on the differences between “*animate characters*” and traditional agents [27]. With this in mind they indicate that agents' behaviors must be “*variable rather than reliable*”, “*idiosyncratic instead of predictable*”, “*appropriate rather than correct*”, “*effective instead of complete*”, “*interesting rather than efficient*”, and “*distinctively individual as opposed to optimal*”.

Perlin and Goldberg [59] concern themselves with building believable characters “*that respond to users and to each other in real-time, with consistent personalities, properly changing moods and without mechanical repetition, while always maintaining an author's goals and intentions*”.

Finally, in characterizing believable agents, Bates [7] is quite forgiving requiring “*only that they not be clearly stupid or unreal*”. Such broad, shallow agents must “*exhibit some signs of internal goals, reactivity, emotion, natural language ability, and knowledge of agents ... as well as of the ... micro-world*”.

Considering these definitions [32] identifies the fact that the consistent themes which run through all of the requirements given above match the general goals of

264 agency – virtual humans must display autonomy, reactiv- 312
 265 ity, goal driven behavior and social ability – and again sup- 313
 266 ports the use of agent-based modeling to drive the behav- 314
 267 ior of virtual characters. 315

268 The Spectrum of Agents 316

269 The differences between the systems mentioned in the pre- 317
 270 vious discussion are captured particularly well on the *spec-* 318
 271 *trum of agents* presented by Aylett and Luck [5]. This 319
 272 positions agent systems on a spectrum based on their ca- 320
 273 pabilities, and serves as a useful tool in differentiating be- 321
 274 tween the various systems available. One end of this spec- 322
 275 trum focuses on *physical agents* which are mainly con- 323
 276 cerned with simulation of believable physical behavior, 324
 277 including sophisticated physiological models of muscle 325
 278 and skeleton systems, and of sensory systems. Interesting 326
 279 work at this end of the spectrum includes Terzopoulos’ 327
 280 highly realistic simulation of fish [76] and his virtual stunt- 328
 281 man project [21] which creates virtual actors capable of re- 329
 282 alistically synthesizing a broad repertoire of lifelike motor 330
 283 skills.

284 *Cognitive agents* inhabit the other end of the agent 331
 285 spectrum and are mainly concerned with issues such as 332
 286 reasoning, decision making, planning and learning. Sys- 333
 287 tems at this end of the spectrum include Funge’s cogni- 334
 288 tive modeling approach [26] which uses the situation cal- 335
 289 culus to control the behavior of virtual characters, and 336
 290 Nareyek’s work on planning agents for simulation [55], 337
 291 both of which will be described later in this article. 338

292 While the systems mentioned so far sit comfortably 339
 293 at either end of the agent spectrum, many of the most 340
 294 effective inhabit the middle ground. Amongst these are 341
 295 *c4* [13], used to great effect to simulate a virtual sheep dog 342
 296 with the ability to learn new behaviors, *Improv* [59] which 343
 297 augments sophisticated physical human animation with 344
 298 scripted behaviors and the *ViCrowd* system [62] which sits 345
 299 on top of a realistic virtual human animation system and 346
 300 uses planning to control agents’ behavior. 347

301 Virtual Fidelity 348

302 The fact that so many different agent-based modeling sys- 349
 303 tems, all for the control of virtual humans exist gives rise to 350
 304 the question why? The answer to this lies in the notion of 351
 305 *virtual fidelity*, as described by Badler [6]. Virtual fidelity 352
 306 refers to the fact that virtual reality systems need only re- 353
 307 main true to actual reality in so much as this is required 354
 308 by, and improves, the system. 355

309 In [47] the point is illustrated extremely effectively. 356
 310 The article explains that when game designers are archi- 357
 311 tecting the environments in which games are set, the scale 358

312 to which these environments are created is not kept true to 313
 314 reality. Rather, to ease players’ movement in these worlds, 315
 316 areas are designed to a much larger scale, compared to 317
 318 character sizes, than in the real world. However, game 319
 320 players do not notice this digression from reality, and in 321
 322 fact have a negative response to environments that are de- 322
 323 signed to be more true to life finding them cramped. This 323
 324 is a perfect example of how, although designers stay true 324
 325 to reality for many aspects of environment design, the par- 325
 326 ticular blend of virtual fidelity required by an application 326
 327 can dictate certain real world restrictions can be ignored 327
 328 in virtual worlds. 328

329 With regard to virtual characters, virtual fidelity dic- 329
 330 tates that the set of capabilities which these characters 330
 331 should display is determined by the application which they 331
 332 are to inhabit. So, the requirements of an agent-based 332
 333 modeling system for CGI in movies would be very differ- 333
 334 ent to those of a agent-based modeling system for control- 334
 335 ling the behaviors of game characters. 335

Agent-Based Modelling in CGI for Movies 336

336 With the success of agent-based modeling techniques 337
 337 in graphics firmly established there was something of 338
 338 a search for application areas to which they could be ap- 339
 339 plied. Fortunately, the success of agent-based modeling 340
 340 techniques in computer graphics was paralleled with an 341
 341 increase in the use of CGI in the movie industry, which 342
 342 offered the perfect opportunity. In many cases CGI tech- 343
 343 niques were being used to replace traditional methods for 344
 344 creating expensive, or difficult to film scenes. In particular, 345
 345 scenes involving large numbers of people or animals were 346
 346 deemed no longer financially viable when set in the real 347
 347 world. Creating these scenes using CGI involved painstaking 348
 348 hand animation of each character within a scene, 349
 349 which again was not financially viable. 350

350 The solution that agent-based modeling offers is to 351
 351 make each character within a scene an intelligent agent 352
 352 that drives its own behavior. In this way, as long as the in- 353
 353 itial situation is set up correctly scenes will play out without 354
 354 the intervention of animators. The facts that animating for 355
 355 movies does not need to be performed in real-time, and is 356
 356 in no way interactive (there are no human users involved 357
 357 in the scene), make the use of agent-based modeling a par- 358
 358 ticularly fine match for this application area. 359

359 Craig Reynolds’ Boids system [64] which simulates the 360
 360 flocking behaviors exhibited in nature by schools of fish, 361
 361 or flocks of birds and was discussed previously is one of 362
 362 the seminal early examples of agent-based modeling tech- 363
 363 niques being used in movie CGI. Reynold’s approach was 364
 364 first used for CGI in the 1999 film “Batman Returns” [14] 365

361 to simulate swarms of bats. Reynold's technologies have
 362 been used in "The Lion King" [4] and "From Dusk 'Till
 363 Dawn" [65] amongst others. Reynolds' approach was so
 364 successful, in fact, that he was awarded an Academy Award
 365 for his work in 1998.

366 Similar techniques to those utilized in the Boids sys-
 367 tem have been used in many other films to animate such
 368 diverse characters as ants, people and stampeding wilde-
 369 beest. Two productions which were released in the same
 370 year, "Antz" [17] by Dreamworks and "A Bug's Life" [44]
 371 by Pixar took great steps in animating large crowds for
 372 CGI effects. For "Antz" systems were developed which
 373 allowed animators easily create scenes containing large
 374 numbers of virtual characters modeling each as an intel-
 375 ligent agent capable of obstacle avoidance, flocking and
 376 other behaviors. Similarly, the creators of "A Bug's Life"
 377 created tools which allowed animators easily combine pre-
 378 defined motions (known as alibis) to create behaviors
 379 which could easily be applied to individual agents in scenes
 380 composed of hundreds of virtual characters.

381 However, the largest jump in the use of agent-based
 382 modeling in movie CGI was made in the recent Lord of
 383 the Rings trilogy [33,34,35]. In these films the bar was
 384 raised markedly in terms of the sophistication of the vir-
 385 tual characters displayed and the sheer number of char-
 386 acters populating each scene. To achieve the special ef-
 387 fects shots required by the makers of these films, the
 388 Massive software system was developed by Massive Soft-
 389 ware (www.massivesoftware.com). This system [2,39] uses
 390 agent-based modeling techniques, again inspired by aLife,
 391 to create virtual extras that control their own behaviors.
 392 This system was put to particularly good use in the large
 393 scale battle sequences that feature in all three of the Lord
 394 of the Rings films. Some of the sequences in the final film
 395 of the trilogy, the Return of the King, contain over 200,000
 396 digital characters.

397 In order to create a large battle scene using the Massive
 398 software, each virtual extra is represented as an intelligent
 399 agent, making its own decisions about which actions it will
 400 perform based on its perceptions of the world around it.
 401 Agent control is achieved through the use of fuzzy logic
 402 based controllers in which the state of an agent's brain is
 403 represented as a series of motivations, and knowledge it
 404 has about the world – such as the state of the terrain it
 405 finds itself on, what kinds of other agents are around it and
 406 what these other agents are doing. This knowledge about
 407 the world is perceived through simple simulated visual, au-
 408 ditory and tactile senses. Based on the information they
 409 perceive agents decide on a best course of action. Design-
 410 ing the brains of these agents is made easier that it might
 411 seem at first by the fact that agents are developed for short

412 sequences, and so a short range of possible tasks. So for ex-
 413 ample, separate agent models would be used for a fighting
 414 scene and celebration scene.

415 In order to create a large crown scene using Massive
 416 animators initially set up an environment populating it
 417 with an appropriate cast of virtual characters where the
 418 brains of each character are slight variations (based on
 419 physical and personality attribute) of a small number of
 420 archetypes. The scene will then play itself out with each
 421 character making it's own decisions. Therefore there is no
 422 need for any hand animation of virtual characters. How-
 423 ever, directors can view the created scenes and by tweak-
 424 ing the parameters of the brains of the virtual characters
 425 have a scene play out in the exact way that they require.

426 Since being used to such impressive effect in the Lord
 427 of the Rings trilogy (the developers of the Massive sys-
 428 tem were awarded an academy award for their work), the
 429 Massive software system has been used in numerous other
 430 films such as "I, Robot" [60], "The Chronicles of Narnia:
 431 The Lion, the Witch and the Wardrobe" [1] and "Rata-
 432 touille" [10] along with numerous television commercials
 433 and music videos.

434 While the achievements of using agent-based model-
 435 ing for movie CGI are extremely impressive, it is worth
 436 noting that none of these systems run in real-time. Rather,
 437 scenes are rendered by banks of high powered comput-
 438 ers, a process that can take hours for relatively simple
 439 scenes. For example, the famous Prologue battle sequence
 440 in the "Lord of the Rings: The Fellowship of the Ring" took
 441 a week to render. When agent-based modeling is applied
 442 to the real-time world of computer games, things are very
 443 different.

444 Agent-Based Modelling in Games

445 Even more so than in movies, agent-based modeling tech-
 446 niques have been used to drive the behaviors of virtual
 447 characters in computer games. As games have become
 448 graphically more realistic (and in recent years they have
 449 become extremely so) game-players have come to expect
 450 that games are set in hugely realistic believable virtual
 451 worlds. This is particularly evident in the widespread use
 452 of realistic physics modeling which is now commonplace
 453 in games [67]. In games that make strong use of physics
 454 modeling objects in the game world topple over when
 455 pushed, float realistically when dropped in water and gen-
 456 erally respond as one would expect them to. Players expect
 457 the same to be true of the virtual characters that populate
 458 virtual game worlds. This can be best achieved by mod-
 459 eling virtual characters as embodied virtual agents. How-
 460 ever, there are a number of constraints which have a major

461 influence on the use of agent-based modeling techniques
462 in games.

463 The first of these constraints stems from the fact that
464 modern games are so highly interactive. Players expect to
465 be able to interact with all of the characters they encounter
466 within a game world. These interactions can be as simple
467 as having something to shoot at or having someone to race
468 against; or involve much more sophisticated interactions
469 in which a player is expected to converse with a virtual
470 character to find out specific information or to cooperate
471 with a virtual character in order to accomplish some
472 task that is key to the plot of a game. Interactivity raises
473 a massive challenge for practitioners as there is very little
474 restriction in terms of what the player might do. Virtual
475 characters should respond in a believable way at all times
476 regardless of how bizarre and unexpected the actions of
477 the player might be.

478 The second challenge comes from the fact that the
479 vast majority of video games should run in real time. This
480 means that the computational complexity must be kept to
481 a reasonable level as there are only a finite number of processor
482 cycles available for AI processing. This problem is
483 magnified by the fact that an enormous amount of CPU
484 power is usually dedicated to graphics processing. When
485 compared to the techniques that can be used for controlling
486 virtual characters in films some of the techniques used
487 in games are rudimentary due to this real-time constraint.

488 Finally, modern games resemble films in the fact that
489 creators go to great lengths to include intricate storylines
490 and control the building of tension in much the way that
491 film script writers do. This means that games are tested
492 heavily in order to ensure that the game proceeds smoothly
493 and that the level of difficulty is finely tuned so as to
494 always hold the interest of a player. In fact, this testing
495 of games has become something of a science in itself [77].
496 Using autonomous agents gives game characters the ability
497 to do things that are unexpected by the game designers
498 and so upset their well laid plans. This can often be a barrier
499 to the use of sophisticated techniques such as learning.

500 Unfortunately there is also a barrier to the discussion
501 of agent-based modeling techniques used in commercial
502 games. Because of the very competitive nature of the
503 games industry, game development houses often consider
504 the details of how their games work as valuable trade secrets
505 to be kept well guarded. This can make it difficult to
506 uncover the details of how particularly interesting features
507 of a game are implemented. While this situation is improving
508 – more commercial game developers are speaking at
509 games conferences about how their games are developed
510 and the release of game systems development kits for the
511 development of game modifications (or *mods*) allows re-

searchers to plumb the depths of game code – it is still often
512 impossible to find out the implementation details of
513 very new games. 514

515 Game Genres

516 Before discussing the use of agent-based modeling in
517 games any further, it is worth making a short clarification
518 on the kinds of computer games that this article refers
519 to. When discussing modern computer games, or video
520 games, this article does not refer to computer implemen-
521 tations of traditional games such as chess, backgammon
522 or card games such as solitaire. Although these games are
523 of considerable research interest (chess in particular has
524 been the subject of extremely successful research [23]) they
525 are typically not approached using agent-based modeling
526 techniques. Typically, artificial intelligence approaches to
527 games such as these rely largely on sophisticated searching
528 techniques which allow the computer player to search
529 through a multitude of possible future situations dictated
530 by the moves it will make and the moves it expects its
531 opponent to make in response. Based on this search,
532 and some clever heuristics that indicate what constitutes
533 a good game position for the computer player, the best
534 sequence of moves can be chosen. This searching technique
535 relies on the fact that there are usually a relatively small
536 number of moves that a player can make at any one time
537 in a game. However, the fact that the ancient Chinese game
538 of Go-Moku has not, to date, been mastered by computer
539 players [80] illustrates the restrictions of such techniques.

540 The common thread linking together the kinds of
541 games that this article focuses on is that they all contain
542 computer controlled virtual characters that possess a
543 strong notion of agency. Efforts are often made to separate
544 the many different kinds of modern video games that
545 are the focus of this article into a small set of descriptive
546 genres. Unfortunately, much like in music, film and literature,
547 no categorization can hope to perfectly capture the
548 nuances of all of the available titles. However, a brief
549 mention of some of the more important game genres is worth
550 while (a more detailed description of game genres, and
551 artificial intelligence requirements of each is given in [41]).

552 The most popular game genre is without doubt the
553 *action game* in which the player must defeat waves of
554 demoted foes, typically (for increasingly bizarre motivations)
555 bent upon global destruction. Illustrative examples
556 of the genre include *Half-Life 2* (www.half-life2.com) and
557 the *Halo* series (www.halo3.com). A screenshot of the
558 upcoming action game *Rogue Warrior* (www.bethsoft.com)
559 is shown in Fig. 2.



Agent-Based Modelling in Computer Graphics and Games, Figure 2

A screenshot of the upcoming action game *Rogue Warrior* from Bethesda Softworks (image courtesy of Bethesda Softworks)

560 *Strategy games* allow players to control large armies
561 in battle with other people, or computer controlled op-
562 ponents. Players do not have direct control over their
563 armies, but rather issue orders which are carried out
564 by agent-based artificial soldiers. Well regarded exam-
565 ples of the genre include the *Age of Empires* (www.ageofempires.com) and *Command & Conquer* (www.commandandconquer.com) series.

566
567
568 *Role playing games* (such as the *Elder Scrolls* (www.elderscrolls.com) series) place game players in expansive
569 virtual worlds across which they must embark on fantasti-
570 cal quests which typically involve a mixture of solving puz-
571 zles, fighting opponents and interacting with non-player
572 characters in order to gain information. Figure 3 shows
573 a screenshot of the aforementioned role-playing game *The*
574 *Elder Scrolls IV: Oblivion*.

575
576 Almost every sport imaginable has at this stage been
577 turned into a computer based *sports game*. The challenges
578 in developing these games are creating computer con-
579 trolled opponents and team mates that play the games
580 at a level suitable to the human player. *FIFA Soccer*
581 *08* (www.fifa08.ea.com) and *Forza Motorsport 2* (www.forzamotorsport.net).

582
583 Finally, many people expected that the rise of mas-
584 sively multi-player online games (MMOGs), in which
585 hundreds of human players can play together in an online
586 world, would sound the death knell for the use of virtual
587 non-player characters in games. Examples of MMOGs in-
588 clude *World of Warcraft* (www.worldofwarcraft.com) and
589 *Battlefield 2142* (www.battlefield.ea.com). However, this
590 has not turned out to be the case as there are still large

591 numbers of single player games being produced and even
592 MMOGs need computer controlled characters for roles
593 that players do not wish to play.

594 Of course there are many games that simply do not
595 fit into any of these categorizations, but that are still rel-
596 evant for a discussion of the use of agent-based tech-
597 niques – for example *The Sims* (www.thesims.ea.com) and
598 the *Microsoft Flight Simulator* series (www.microsoft.com/games/flightsimulatorx). However the categorization still
599 serves to introduce those unfamiliar with the subject to the
600 kinds of games up for discussion.
601

602 Implementing Agent-Based Modelling Techniques 603 in Games

604 One of the earliest examples of using agent-based model-
605 ing techniques in video games was its application to path
606 planning. The ability of non-player characters (NPCs) to
607 manoeuvre around a game world is one of the most ba-
608 sic competencies required in games. While in very early
609 games it was sufficient to have NPCs move along pre-
610 scribed paths, this soon become unacceptable. Games
611 programmers soon began to turn to AI techniques which
612 might be applied to solve some of the problems that were
613 arising. The A* path planning algorithm [74] was the first
614 example of such a technique to find wide-spread use in
615 the games industry. Using the A* algorithm NPCs can be
616 given the ability to find their own way around an envi-
617 ronment. This was put to particularly fine effect early on
618 in real-time strategy games where the units controlled by
619 players are semi-autonomous and are given orders rather



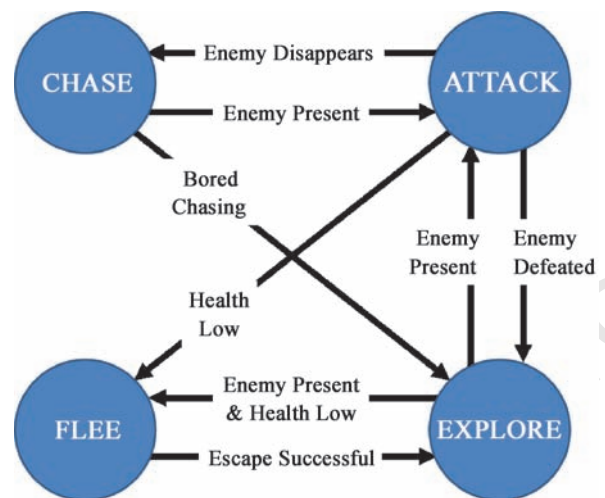
Agent-Based Modelling in Computer Graphics and Games, Figure 3

A screenshot from Bethesda Softwork's role playing game The Elder Scrolls IV: Oblivion (image courtesy of Bethesda Softworks)

620 than directly controlled. In order to use the A* algorithm
 621 a game world must be divided into a series of cells each
 622 of which is given a rating in terms of the effort that must
 623 be expended to cross it. The A* algorithm then performs
 624 a search across these cells in order to find the shortest path
 625 that will take a game agent from a start position to a goal.

626 Since becoming widely understood amongst the game
 627 development community many interesting additions have
 628 been made to the basic A* algorithm. It was not long be-
 629 fore three dimensional versions of the algorithm became
 630 commonly used [71]. The basic notion of storing the en-
 631 ergy required to cross a cell within a game world has also
 632 been extended to augment cells with a wide range of other
 633 useful information (such as the level of danger in crossing
 634 a cell) that can be used in the search process [63].

635 The next advance in the kind of techniques being used
 636 to achieve agent-based modeling in games was the finite
 637 state machine (FSM) [30]. An FSM is a simple system in
 638 which a finite number of *states* are connected in a directed
 639 graph by *transitions* between these states. When used for
 640 the control of NPCs, the nodes of an FSM indicate the
 641 possible actions within a game world that an agent can
 642 perform. Transitions indicate how changes in the state of
 643 the game world or the character's own attributes (such as
 644 health, tiredness etc) can move the agent from one state to
 645 another.



Agent-Based Modelling in Computer Graphics and Games, Figure 4

A simple finite state machine for a soldier NPC in an action game

646 Figure 4 shows a sample FSM for the control of an
 647 NPC in a typical action game. In this example the behav-
 648 iors of the character are determined by just four states –
 649 CHASE, ATTACK, FLEE and EXPLORE. Each of these states
 650 provides an action that the agent should take. For exam-

651 ple, when in the EXPLORE state the character should wan- 702
 652 der randomly around the world, or while in the FLEE state 703
 653 the character should determine a direction to move in that 704
 654 will take it away from its current enemy and move in that
 655 direction. The links between the states show how the be-
 656 haviors of the character should move between the various
 657 available states. So, for example, if while in the ATTACK
 658 state the agent's health measure becomes low, they will
 659 move to the FLEE state and run away from their enemy.

660 FSMs are widely used because they are so simple, well
 661 understood and extremely efficient both in terms of pro-
 662 cessing cycles required and memory usage. There have
 663 also been a number of highly successful augmentations
 664 to the basic state machine model to make them more ef-
 665 fective, such as the introduction of layers of parallel state
 666 machines [3], the use of fuzzy logic in finite state ma-
 667 chines [19] and the implementation of cooperative group
 668 behaviors through state machines [72].

669 The action game *Halo 2* is recognized as having a par-
 670 ticularly good implementation of state machine based
 671 NPC control [79]. At any time an agent could be in any
 672 one of the four states *Idle*, *Guard/Patrol*, *Attack/Defend*,
 673 and *Retreat*. Within each of these states a set of rules was
 674 used in order to select from a small set of appropriate ac-
 675 tions for that state – for example a number of different
 676 ways to attack the player. The decisions made by NPCs
 677 were influenced by a number of character attributes in-
 678 cluding strength, speed and cowardliness. Transition be-
 679 tween states was triggered by perceptions made by charac-
 680 ters simulated senses of vision and hearing and by internal
 681 attributes such as health. The system implemented also al-
 682 lowed for group behaviors allowing NPCs to hold conver-
 683 sations and cooperate to drive vehicles.

684 However, FSMs are not without their drawbacks.
 685 When designing FSMs developers must envisage every
 686 possible situation that might confront an NPC over the
 687 course of a game. While this is quite possible for many
 688 games, if NPCs are required to move between many dif-
 689 ferent situations this task can become overwhelming. Sim-
 690 ilarly, as more and more states are added to an FSM de-
 691 signing the links between these states can become a mam-
 692 moth undertaking.

693 From [31] the definition of rule based systems states
 694 that they are "... comprised of a database of associated
 695 rules. Rules are conditional program statements with con-
 696 sequent actions that are performed if the specified condi-
 697 tions are satisfied". Rule based systems have been applied
 698 extensively to control NPCs in games [16], in particular
 699 for the control of NPCs in role-playing games. NPCs be-
 700 haviors are scripted using a set of rules which typically in-
 701 dicate how an NPC should respond to particular events

702 within the game world. Borrowed from [82], the listing be-
 703 low shows a snippet of the rules used to control a warrior
 704 character in the RPG *Baldur's Gate* (www.bioware.com).

```

705 IF
706 // If my nearest enemy is not within 3
707 !Range(NearestEnemyOf(Myself),3)
708 // and is within 8
709 Range(NearestEnemyOf(Myself),8)
710 THEN
711 // 1/3 of the time
712 RESPONSE #40
713 // Equip my best melee weapon
714 EquipMostDamagingMelee()
715 // and attack my nearest enemy, checking every 60
716 // ticks to make sure he is still the nearest
717 AttackReevaluate(NearestEnemyOf(Myself),60)
718 // 2/3 of the time
719 RESPONSE #60
720 // Equip a ranged weapon
721 EquipRanged()
722 // and attack my nearest enemy, checking every 30
723 // ticks to make sure he is still the nearest
724 AttackReevaluate(NearestEnemyOf(Myself), 30)
  
```

725 The implementation of an NPC using a rule-based system
 726 would consist of a large set of such rules, a small set of
 727 which would fire based on the conditions in the world at
 728 any given time. Rule based systems are favored by game
 729 developers as they are relatively simple to use and can be
 730 exhaustively tested. Rule based systems also have the ad-
 731 vantage that rule sets can be written using simple propri-
 732 etary scripting systems [9], rather than full programming
 733 languages, making them easy to implement. Development
 734 companies have also gone so far as to make these scripting
 735 languages available to the general public, enabling them to
 736 author their own rule sets.

737 Rule based systems, however, are not without their
 738 drawbacks. Authoring extensive rule sets is not a trivial
 739 task, and they are usually restricted to simple situations.
 740 Also, rule based systems can be restrictive in that they
 741 don't allow sophisticated interplay between NPCs motiva-
 742 tions, and require that rule set authors foresee every situa-
 743 tion that the NPC might find itself in.

744 Some of the disadvantages of simple rule based systems
 745 can be alleviated by using more sophisticated inference
 746 engines. One example uses Dempster Schafer theory [43]
 747 which allows rules to be evaluated by combining multi-
 748 ple sources of (often incomplete) evidence to determine
 749 actions. This goes some way towards supporting the use
 750 of rule based systems in situations where complete knowl-
 751 edge is not available.

752 ALife techniques have also been applied extensively in
 753 the control of game NPCs, as much as a philosophy as in
 754 any particular techniques. However, the outstanding ex-
 755 ample of this is *The Sims* (thesims.ea.com) a surprise hit
 756 of 2000 which has gone on to become the best selling PC
 757 game of all time. Created by games guru Will Wright the
 758 Sims puts the player in control of the lives of a virtual fam-
 759 ily in their virtual home. Inspired by aLife the characters
 760 in the game have a set of motivations, such as hunger, fa-
 761 tigue and boredom and seek out items within the game
 762 world that can satisfy these desires. Virtual characters also
 763 develop sophisticated social relationships with each other
 764 based on common interest, attraction and the amount of
 765 time spent together. The original system in the Sims has
 766 gone on to be improved in the sequel *The Sims 2* and a se-
 767 ries of expansion packs.

768 Some of the more interesting work in developing tech-
 769 niques for the control of game characters (particularly in
 770 action games) has been focused on developing interesting
 771 sensing and memory models for game characters. Play-
 772 ers expect when playing action games that computer con-
 773 trolled opponents should suffer from the same problems
 774 that they do when perceiving the world. So, for exam-
 775 ple, computer controlled characters should not be able to
 776 see through walls or from one floor to the next. Similarly,
 777 though, players expect computer controlled characters to
 778 be capable of perceiving events that occur in a world and
 779 so NPCs should respond appropriately to sound events or
 780 on seeing the player.

781 One particularly fine example of a sensing model was
 782 in the game *Thief: The Dark Project* where players are re-
 783 quired to sneak around an environment without alerting
 784 guards to their presence [45]. The developers of *Thief 2*
 785 produced a relatively sophisticated sensing model that was
 786 used by non-player characters which modeled visual ef-
 787 fects such as not being able to see the player if they were in
 788 shadows, and moving some way towards modeling acous-
 789 tics so that non-player characters could respond reason-
 790 ably to sound events.

791 2004's *Fable* (fable.lionhead.com) took the idea of
 792 adding memory to a game to new heights. In this adven-
 793 ture game the player took on the role of a hero from boy-
 794 hood to manhood. However, every action the player took
 795 had an impact on the way in which the game world's pop-
 796 ulation would react to him or her as they would remember
 797 every action the next time they met the player. This notion
 798 of long-term consequences added an extra layer of believ-
 799 ability to the game-playing experience.

Serious Games & Academia

800
 801 It will probably have become apparent to most readers of
 802 the previous section that much of the work done in imple-
 803 menting agent-based techniques for the control of NPCs
 804 in commercial games is relatively simplistic when com-
 805 pared to the application of these techniques in other ar-
 806 eas of more academic focus, such as robotics [54]. The
 807 reasons for this have been discussed already and briefly
 808 relate to the lack of available processing resources and
 809 the requirements of commercial quality control. However,
 810 a large amount of very interesting work is taking place in
 811 the application of agent-based technologies in academic
 812 research, and in particular the field of serious games. This
 813 section will begin by introducing the area of serious games
 814 and then discuss interesting academic projects looking at
 815 agent-based technologies in games.

816 The term serious games [53] refers to games designed
 817 to do more than just entertain. Rather, serious games,
 818 while having many features in common with conven-
 819 tional games, have ulterior motives such as teaching, train-
 820 ing, and marketing. Although games have been used for
 821 ends apart from entertainment, in particular education,
 822 for a long time, the modern serious games movement is set
 823 apart from these by the level of sophistication of the games
 824 it creates. The current generation of serious games is com-
 825 parable with main-stream games in terms of the quality
 826 of production and sophistication of their design. Serious
 827 games offer particularly interesting opportunities for the
 828 use of agent-based modeling techniques due to the facts
 829 that they often do not have to live up to the rigorous testing
 830 of commercial games, can have the requirement of special-
 831 ized hardware rather than being restricted to commercial
 832 games hardware and often, by the nature of their applica-
 833 tion domains, require more in-depth interactions between
 834 players and NPCs.

835 The modern serious games movement can be said to
 836 have begun with the release of *America's Army* (www.
 837 americasarmy.com) in 2002 [57]. Inspired by the real-
 838 ism of commercial games such as the *Rainbow 6* series
 839 (www.rainbow6.com), the United States military devel-
 840 oped *America's Army* and released it free of charge in or-
 841 der to give potential recruits a flavor of army life. The game
 842 was hugely successful and is still being used today as both
 843 a recruitment tool and as an internal army training tool.

844 Spurred on by the success of *America's Army* the seri-
 845 ous games movement began to grow, particularly within
 846 academia. A number of conferences sprung up and no-
 847 tably the Serious Games Summit became a part of the
 848 influential Game Developer's Conference (www.gdconf.
 849 com) in 2004.

850 Some other notable offerings in the serious games field
851 include *Food Force* (www.food-force.com) [18], a game
852 developed by the United Nations World Food Programme
853 in order to promote awareness of the issues surrounding
854 emergency food aid; *Hazmat Hotzone* [15], a game devel-
855 oped by the Entertainment Technology Centre at Carnegie
856 Mellon University to train fire-fighters to deal with chem-
857 ical and hazardous materials emergencies; *Yoursel!Fitness*
858 (www.yoursel!fitness.com) [53] an interactive virtual per-
859 sonal trainer developed for modern games consoles; and
860 *Serious Gordon* (www.seriousgames.ie) [50] a game devel-
861 oped to aid in teaching food safety in kitchens. A screen
862 shot of *Serious Gordon* is shown in Fig. 5.

863 Over the past decade, interest in academic research
864 that is directly focused on artificial intelligence, and
865 in particular agent-based, techniques and their applica-
866 tion to games (as opposed to the general virtual char-
867 acter/computer graphics work discussed previously) has
868 grown dramatically. One of the first major academic re-
869 search projects into the area of Game-AI was lead by John
870 Laird at the University of Michigan, in the United States.
871 The SOAR architecture was developed in the early nine-
872 teen eighties in an attempt to “develop and apply a unified
873 theory of human and artificial intelligence” [66]. SOAR is
874 essentially a rule based inference system which takes the
875 current state of a problem and matches this to production
876 rules which lead to actions.

877 After initial applications into the kind of simple puz-
878 zle worlds which characterized early AI research [42], the
879 SOAR architecture was applied to the task of controlling
880 computer generated forces [37]. This work lead to an ob-
881 vious transfer to the new research area of game-AI [40].

882 Initially the work of Laird’s group focused on apply-
883 ing the SOAR architecture to the task of controlling NPC
884 opponents in the action game *Quake* (www.idsoftware.com) [40]. This proved quite successful leading to oppo-
885 nents which could successfully play against human play-
886 ers, and even begin to plan based on anticipation of what
887 the player was about to do. More recently Laird’s group
888 have focused on the development of a game which re-
889 quires more involved interactions between the player and
890 the NPCs. Named *Haunt 2*, this game casts the player in
891 the role of a ghost that must attempt to influence the ac-
892 tions of a group of computer controlled characters inhab-
893 iting the ghost’s haunted house [51]. The main issue that
894 arises with the use the SOAR architecture is that it is enor-
895 mously resource hungry, with the NPC controllers run-
896 ning on a separate machine to the actual game.

898 At Trinity College in Dublin in Ireland, the author
899 of this article worked on an intelligent agent architec-
900 ture, the Proactive Persistent Agent (PPA) architecture,

901 for the control of background characters (or support char-
902 acters) in character-centric games (games that focus on
903 character interactions rather than action, e. g. role-play-
904 ing games) [48,49]. The key contributions of this work was
905 that it made possible the creation of NPCs that were capa-
906 ble of behaving believably in a wide range of situations and
907 allowed for the creation of game environments which it
908 appeared had an existence beyond their interactions with
909 players. Agent behaviors in this work were based on mod-
910 els of personality, emotion, relationships to other charac-
911 ters and behavioral models that changed according to the
912 current role of an agent. This system was used to develop
913 a stand alone game and as apart of a simulation of parts of
914 Trinity College. A screenshot of this second application is
915 shown in Fig. 6.

916 At Northwestern University in Chicago the Interac-
917 tive Entertainment group has also applied approaches
918 from more traditional research areas to the problems fac-
919 ing game-AI. Ian Horswill has lead a team which are
920 attempting to use architectures traditionally associated
921 with robotics for the control of NPCs. In [29] Horswill
922 and Zubek consider how perfectly matched the behavior
923 based architectures often used in robotics are with the re-
924 quirements of NPC control architectures. The group have
925 demonstrated some of their ideas in a test-bed environ-
926 ment built on top of the game *Half-Life* [38]. The group
927 also looks at issues around character interaction [85] and
928 the many psychological issues associated with creating vir-
929 tual characters asking how we can create virtual game
930 agents that display all of the foibles that make us relate to
931 characters in human stories [28].

932 Within the same research group a team led by Ken For-
933 bus have extended research previously undertaken in con-
934 junction with the military [24] and applied it to the prob-
935 lem of terrain analysis in computer strategy games [25].
936 Their goal is to create strategic opponents which are ca-
937 pable of performing sophisticated reasoning about the ter-
938 rain in a game world and using this knowledge to iden-
939 tify complex features such as ambush points. This kind
940 of high level reasoning would allow AI opponents play
941 a much more realistic game, and even surprise human
942 players from time to time, something that is sorely miss-
943 ing from current strategy games.

944 As well as this work which has spring-boarded from
945 existing applications, a number of projects began expressly
946 to tackle problems in game-AI. Two which particularly
947 stand out are the *Excalibur Project*, led by Alexander
948 Nareyek [55] and work by John Funge [26]. Both of these
949 projects have attempted to applying sophisticated plan-
950 ning techniques to the control of game characters.



Agent-Based Modelling in Computer Graphics and Games, Figure 5
A screenshot of Serious Gordon a serious game developed to aid in the teaching of food safety in kitchens



Agent-Based Modelling in Computer Graphics and Games, Figure 6
Screenshots of the PPA system simulating parts of a college

951 Nareyek uses constraint based planning to allow game
952 agents reason about their world. By using techniques such
953 as local search Nareyek has attempted to allow these so-
954 phisticated agents perform resource intensive planning
955 within the constraints of a typical computer game envi-
956 ronment. Following on from this work, the term *anytime*
957 *agent* was coined to describe the process by which agents
958 actively refine original plans based on changing world con-
959 ditions. In [56] Nareyek describes the directions in which
960 he intends to take this work in future.

961 Funge uses the situational calculus to allow agents rea-
962 son about their world. Similarly to Nareyek he has ad-
963 dressed the problems of a dynamic, ever changing world,
964 plan refining and incomplete information. Funge's work
965 uses an extension to the situational calculus which allows
966 the expression of uncertainty. Since completing this work
967 Funge has gone on to be one of the founders of AiLive
968 (www.ailive.net), a middleware company specializing in
969 AI for games.

970 While the approaches of both of these projects have
971 shown promise within the constrained environments to
972 which they have been applied during research, and work
973 continues on them it remains to be seen whether such
974 techniques can be successfully applied to a commercial
975 game environment and all of the resource constraints that
976 such an environment entails.

977 One of the most interesting recent examples of agent-
978 based work in the field of serious games is that under-
979 taken by Barry Silverman and his group at the Univer-
980 sity of Pennsylvania in the United States [69,70]. Silver-
981 man models the protagonists in military simulations for
982 use in training programmes and has taken a very interest-
983 ing approach in his agent models are based on established
984 cognitive science and behavioral science research. While
985 Silverman admits that many of the models described in the
986 cognitive science and behavioral science literature are not
987 well quantified enough to be directly implemented, he has
988 adapted a number of well respected models for his pur-
989 poses. Silverman's work is an excellent example of the ca-
990 pabilities that can be explored in a serious games setting
991 rather than a commercial game setting, and as such merits
992 an in depth discussion. A high-level schematic diagram of
993 Silverman's approach is shown in Fig. 7 showing the agent
994 architecture used by Silverman's system, PMFserv.

995 The first important component of the PMFserv system
996 is the biology module which controls biological needs us-
997 ing a metaphor based on the flow of water through a sys-
998 tem. Biological concepts such as hunger and fatigue are
999 simulated using a series of reservoirs, tanks and valves
1000 which model the way in which resources are consumed
1001 by the system. This biological model is used in part to

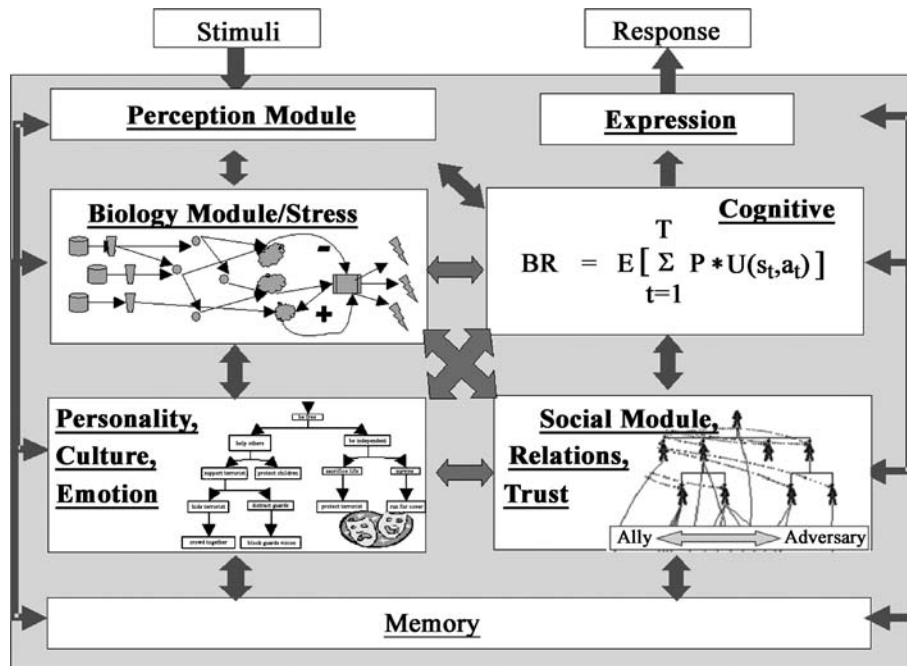
1002 model stress which has an important impact on the way in
1003 which agents make decisions. To model the way in which
1004 agent performance changes under pressure Silverman uses
1005 *performance moderator functions* (PMFs). An example of
1006 one of the earliest PMFs used is the Yerkes–Dodson “in-
1007 verted-u” curve [84] which illustrates that as a stimulus is
1008 increased performance initially improves, peaks and then
1009 trails off again. In PMFserv a range of PMFs are used to
1010 model the way in which behavior should change depend-
1011 ing on stress levels and biological conditions.

1012 The second important module of PMFserv attempts
1013 to model how personality culture and emotion affect the
1014 behavior of an agent. In keeping with the rest of their
1015 system PMFserv uses models inspired by cognitive sci-
1016 ence to model emotions. In this case the well known OCC
1017 model [58], which has been used in agent-based applica-
1018 tions before [8], is used. The OCC model provides for 11
1019 pairs of opposite emotions such as pride and shame, and
1020 hope and fear. The emotional state of an agent with regard
1021 to past, current and future actions heavily influences the
1022 decisions that the agent makes.

1023 The second portion of the Personality, Culture, Emo-
1024 tion module uses a value tree in order to capture the val-
1025 ues of an agent. These values are divided into a Preference
1026 Tree which captures long term desired states for the world,
1027 a Standards Tree which relates to the actions that an agent
1028 believes it can or cannot follow in order to achieve these
1029 desired states and a Goal Tree which captures short term
1030 goals.

1031 The PMFserv also models the relationships between
1032 agents (Social Model, Relations, Trust in Fig. 7). The re-
1033 lationship of one agent to another is modeled in terms of
1034 three axes. The first is the degree to which the other agent
1035 is thought of as a human rather than an inanimate object
1036 relationship – locals tend to view American soldiers as ob-
1037 jects rather than people. The second axis is the cognitive
1038 grouping (ally, foe etc) to which the other agent belongs
1039 and whether this is also a group to which the first agent has
1040 an affinity. Finally, the valence, or strength, of the rela-
1041 tionship is stored. Relationships continually change based on
1042 actions that occur within the game world. Like the other
1043 modules of the system this model is also based on psycho-
1044 logical research [58].

1045 The final important module of the PMFserv architec-
1046 ture is the Cognitive module which is used to decide on
1047 particular actions that agents will undertake. This module
1048 uses inputs from all of the other modules to make these de-
1049 cisions and so the behavior of PMFserv agents is driven by
1050 their stress levels, relationships to other agents and objects
1051 within the game world, personality, culture and emotions.
1052 The details of the PMFserv cognitive process are beyond



Agent-Based Modelling in Computer Graphics and Games, Figure 7

A schematic diagram of the main components of the PMFserv system (with kind permission of Barry Silverman)



Agent-Based Modelling in Computer Graphics and Games, Figure 8

A screenshot of the PMFserv system being used to simulate the Black Hawk Down scenario (with kind permission of Barry Silverman)

1053 the scope of this article, so it will suffice to say that action
1054 selection is based on a calculation of the utility of a partic-
1055 ular action to an agent, with this calculation modified by
1056 the factors listed above.

1057 The most highly developed example using the PMF-
1058 serv model is a simulation of the 1993 event in Mogadishu,
1059 Somalia in which a United States military Black Hawk he-
1060 licopter crashed, as made famous by the book and film
1061 “Black Hawk Down” [12]. In this example, which was
1062 developed as a military training aid as part of a larger
1063 project looking at agent implementations within such sys-
1064 tems [78,81] the player took on the role of a US army
1065 ranger on a mission to secure the helicopter wreck in
1066 a modification (or “mod”) of the game *Unreal Tourna-*
1067 *ment* (www.unreal.com). A screenshot of this simulation
1068 is shown in Fig. 8.

1069 The PMFserv system was used to control the behav-
1070 iors of characters within the game world such as Somali
1071 militia, and Somali civilians. These characters were im-
1072 bued with physical attributes, a value system and relation-
1073 ships with other characters and objects within the game
1074 environment. The sophistication of PMFserv was appar-
1075 ent in many of the behaviors of the simulations NPCs. One
1076 particularly good example was the fact that Somali women
1077 would offer themselves as human shields for militia fight-
1078 ers. This behavior was never directly programmed into the
1079 agents make-up, but rather emerged as a result of their
1080 values and assessment of their situation. PMFserv remains
1081 one of the most sophisticated current agent implementa-
1082 tions and shows the possibilities when the shackles of com-
1083 mercial game constraints are thrown off.

1084 Future Directions

1085 There is no doubt that with the increase in the amount of
1086 work being focused on the use of agent-based modeling
1087 in computer graphics and games that there will be large
1088 developments in the near future. This final section will at-
1089 tempt to predict what some of these might be.

1090 The main development that might be expected in all
1091 of the areas that have been discussed in this article is an
1092 increase in the depth of simulation. The primary driver of
1093 this increase in depth will be the development of more so-
1094 phisticated agent models which can be used to drive ever
1095 more sophisticated agent behavior. The PMFserv system
1096 described earlier is one example of the kinds of deeper sys-
1097 tems that are currently being developed. In general com-
1098 puter graphics applications this will allow for the creation
1099 of more interesting simulations including previously pro-
1100 hibitive features such as automatic realistic facial expres-
1101 sions and other physical expressions of agents’ internal

1102 states. This would be particularly use in CGI for movies
1103 in which, although agent based modeling techniques are
1104 commonly used for crowd scenes and background charac-
1105 ters, main characters are still animated almost entirely by
1106 hand.

1107 In the area of computer games it can be expected that
1108 many of the techniques being used in movie CGI will fil-
1109 ter over to real-time game applications as the process-
1110 ing power of game hardware increases – this is a pattern
1111 that has been evident for the past number of years. In
1112 terms of depth that might be added to the control of game
1113 characters one feature that has mainly been conspicuous
1114 by its absence in modern games is genuine learning by
1115 game agents. 2000’s *Black & White* and its sequel *Black &*
1116 *White 2* (www.lionhead.com) featured some learning by
1117 one of the game’s main characters that the player could
1118 teach in a reinforcement manner [20]. While this was
1119 particular successful in the game, such techniques have
1120 not been more widely applied. One interesting academic
1121 project in this area is NERO project (www.nerogame.org)
1122 which allows a player to train an evolving army of soldiers
1123 and have them battle the armies of other players [73]. It is
1124 expected that these kinds of capabilities will become more
1125 and more common in commercial games.

1126 One new feature of the field of virtual character con-
1127 trol in games is the emergence of specialized middle-
1128 ware. Middleware has had a massive impact in other
1129 areas of game development including character mod-
1130 eling (for example Maya available from www.autodesk.
1131 com) and physics modeling (for example Havok avail-
1132 able from www.havok.com). AI focused middleware for
1133 games is now becoming more common with notable of-
1134 ferings including AI-Implant (www.ai-implant.com) and
1135 Kynogon (www.kynogon.com) which perform path find-
1136 ing and state machine based control of characters. It is ex-
1137 pected that more sophisticated techniques will over time
1138 find their way into such software.

1139 To conclude the great hope for the future is that more
1140 and more sophisticated agent-based modeling techniques
1141 from other application areas and other branches of AI will
1142 find their way into the control of virtual characters.

1143 Bibliography

1144 Primary Literature

- 1145 1. Adamson A (Director) (2005) *The Chronicles of Narnia: The*
1146 *Lion, the Witch and the Wardrobe*. Motion Picture. [http://](http://adisney.go.com/disneypictures/narnia/lb_main.html)
1147 adisney.go.com/disneypictures/narnia/lb_main.html **TS2**
- 1148 2. Aitken M, Butler G, Lemmon D, Saindon E, Peters D, Williams G
1149 (2004) *The Lord of the Rings: the visual effects that brought*
1150 *middle earth to the screen*. International Conference on

TS2 Please provide access date.

CE3 Please provide date and location of conference/proceeding.

- 1151 Computer Graphics and Interactive Techniques (SIGGRAPH),
1152 Course Notes **CE3**
- 1153 3. Alexander T (2003) Parallel-State Machines for Believable Char-
1154 acters. In: **CE4** Massively Multiplayer Game Development.
1155 Charles River Media
- 1156 4. Allers R, Minkoff R (Directors) (1994) The Lion King. Motion
1157 Picture. [http://disney.go.com/disneyvideos/animatedfilms/
1158 lionking/](http://disney.go.com/disneyvideos/animatedfilms/lionking/) **TS2**
- 1159 5. Aylett R, Luck M (2000) Applying Artificial Intelligence to Vir-
1160 tual Reality: Intelligent Virtual Environments. *Appl Artif Intell*
1161 14(1):3–32
- 1162 6. Badler N, Bindiganavale R, Bourne J, Allbeck J, Shi J, Palmer M
1163 (1999) Real Time Virtual Humans. In: Proceedings of the Inter-
1164 national conference on Digital Media Futures. **CE3**
- 1165 7. Bates J (1992) The Nature of Characters in Interactive Worlds
1166 and the Oz Project. Technical Report CMU-CS-92–200. School
1167 of Computer Science, Carnegie Mellon University
- 1168 8. Bates J (1992) Virtual reality, art, and entertainment. *Presence:*
1169 *J Teleoper Virtual Environ* 1(1):133–138
- 1170 9. Berger L (2002) Scripting: Overview and Code Generation. In:
1171 Rabin S (ed) *AI Game Programming wisdom*. Charles River
1172 Media **CE5**
- 1173 10. Bird B, Pinkava J (Directors) (2007) Ratatouille. Motion Pic-
1174 ture. [http://disney.go.com/disneyvideos/animatedfilms/
1175 ratatouille/](http://disney.go.com/disneyvideos/animatedfilms/ratatouille/) **TS2**
- 1176 11. Blumberg B (1996) Old Tricks, New Dogs: Ethology and Interac-
1177 tive Creatures. PhD Thesis, Media Lab, Massachusetts Institute
1178 of Technology **CE5**
- 1179 12. Bowden M (2000) Black Hawk Down. *Corgi Adult* **CE5**
- 1180 13. Burke R, Isla D, Downie M, Ivanov Y, Blumberg B (2002) Cre-
1181 ature Smarts: The Art and Architecture of a Virtual Brain. In: Pro-
1182 ceedings of Game-On 2002: the 3rd International Conference
1183 on Intelligent Games and Simulation, pp 89–93 **CE3**
- 1184 14. Burton T (Director) (1992) Batman Returns. Motion Picture.
1185 <http://www.warnervideo.com/batmanmoviesondvd/> **TS2**
- 1186 15. Carless S (2005) Postcard From SGS 2005: Hazmat: Hot-
1187 zone – First-Person First Responder Gaming. Retrieved Oc-
1188 tober 2007, from Gamasutra: [www.gamasutra.com/features/
1189 20051102/carless_01b.shtml](http://www.gamasutra.com/features/20051102/carless_01b.shtml)
- 1190 16. Christian M (2002) A Simple Inference Engine for a Rule Based
1191 Architecture. In: Rabin S (ed) *AI Game Programming Wisdom*.
1192 Charles River Media **CE5**
- 1193 17. Darnell E, Johnson T (Directors) (1998) Antz. Motion Picture.
1194 <http://www.dreamworksanimation.com/> **TS2**
- 1195 18. DeMaria R (2005) Postcard from the Serious Games Summit:
1196 How the United Nations Fights Hunger with Food Force. Re-
1197 trieved October 2007, from Gamasutra: [www.gamasutra.com/
1198 features/20051104/demaria_01.shtml](http://www.gamasutra.com/features/20051104/demaria_01.shtml)
- 1199 19. Dybsand E (2001) A Generic Fuzzy State Machine in C++.
1200 In: Rabin S (ed) *Game Programming Gems 2*. Charles River
1201 Media **CE5**
- 1202 20. Evans R (2002) Varieties of Learning. In: Rabin S (ed) *AI Game*
1203 *Programming Wisdom*. Charles River Media **CE5**
- 1204 21. Faloutsos P, van de Panne M, Terzopoulos D (2001) The Vir-
1205 tual Stuntman: Dynamic Characters with a Repertoire of Au-
1206 tonomous Motor Skills. *Comput Graph* 25(6):933–953
- 1207 22. Farenc N, Musse S, Schweiss E, Kallmann M, Aune O, Boulic R
1208 et al (2000) A Paradigm for Controlling Virtual Humans in Ur-
1209 ban Environment Simulations. *Appl Artif Intell J Special Issue*
1210 *Intell Virtual Environ* 14(1):69–91
23. Feng-Hsiung H (2002) Behind Deep Blue: Building the Com-
1211 puter that Defeated the World Chess Champion. Princeton
1212 University Press **CE5**
24. Forbus K, Nielsen P, Faltings B (1991) Qualitative Spatial Rea-
1214 soning: The CLOCK Project. *Artif Intell* 51:1–3
25. Forbus K, Mahoney J, Dill K (2001) How Qualitative Spa-
1216 tial Reasoning Can Improve Strategy Game AIs. In: Proceed-
1217 ings of the AAAI Spring Symposium on AI and Interactive
1218 Entertainment **CE3**
26. Funge J (1999) AI for Games and Animation: A Cognitive Mod-
1220 eling Approach. A.K. Peters **CE5**
27. Hayes-Roth B, Doyle P (1998) Animate Characters. *Auton*
1222 *Agents Multi-Agent Syst* 1(2):195–230
28. Horswill I (2007) Psychopathology, narrative, and cognitive ar-
1224 chitecture (or: why NPCs should be just as screwed up as we
1225 are). In: Proceedings of AAAI Fall Symposium on Intelligent
1226 Narrative Technologies, **CE3**
29. Horswill I, Zubek R (1999) Robot Architectures for Believable
1228 Game Agents. In: Proceedings of the 1999 AAAI Spring Sym-
1229 posium on Artificial Intelligence and Computer Games. **CE3**
30. Houlette R, Fu D (2003) The Ultimate Guide to FSMs in Games.
1231 In: Rabin S (ed) *AI Game Programming Wisdom 2*. Charles River
1232 Media **CE5**
31. IGDA (2003) Working Group on Rule-Based Systems Report. In-
1234 ternational Games Development Association **CE5**
32. Isbister K, Doyle P (2002) Design and Evaluation of Embodied
1236 Conversational Agents: A Proposed Taxonomy. In: Proceed-
1237 ings of the AA- MAS02 Workshop on Embodied Conversational
1238 Agents: Lets Specify and Compare Them! Bologna, Italy
33. Jackson P (Director) (2001) The Lord of the Rings: The Fellow-
1240 ship of the Ring. Motion Picture. [http://www.lordoftherings.
1241 net/](http://www.lordoftherings.net/) **TS2**
34. Jackson P (Director) (2002) The Lord of the Rings: The Two
1243 Towers. Motion Picture. <http://www.lordoftherings.net/> **TS2**
35. Jackson P (Director) (2003) The Lord of the Rings: The Return
1245 of the King. Motion Picture. <http://www.lordoftherings.net/> **TS2**
36. Johnston O, Thomas F (1995) The Illusion of Life: Disney Ani-
1247 mation. Disney Editions **CE5**
37. Jones R, Laird J, Neilsen P, Coulter K, Kenny P, Koss F (1999)
1249 Automated Intelligent Pilots for Combat Flight Simulation. *AI*
1250 *Mag* 20(1):27–42
38. Khoo A, Zubek R (2002) Applying Inexpensive AI Techniques to
1252 Computer Games. *IEE Intell Syst Spec Issue Interact Entertain*
1253 17(4):48–53
39. Koeppl D (2002) Massive Attack. [http://www.popsi.com/
1255 popsci/science/d726359b9fa84010vgnvcm1000004eebcddr.
1256 html](http://www.popsi.com/popsci/science/d726359b9fa84010vgnvcm1000004eebcddr.html). Accessed Oct 2007
40. Laird J (2000) An Exploration into Computer Games and Com-
1258 puter Generated Forces. The 8th Conference on Computer
1259 Generated Forces and Behavior Representation. **CE3**
41. Laird J, van Lent M (2000) Human-Level AI's Killer Application:
1261 Interactive Computer Games. In: Proceedings of the 17th Na-
1262 tional Conference on Artificial Intelligence, **CE3**
42. Laird J, Rosenbloom P, Newell A (1984) Towards Chunking as
1264 a General Learning Mechanism. The 1984 National Conference
1265 on Artificial Intelligence (AAAI), pp 188–192 **CE3**
43. Laramée F (2002) A Rule Based Architecture Using Dempster-
1267 Schafer theory. In: Rabin S (ed) *AI Game Programming Wisdom*.
1268 Charles River Media **CE5**
44. Lasseter J, Stanton A (Directors) (1998) A Bug's Life; Motion Pic-
1270 ture. <http://www.pixar.com/featurefilms/abl/> **TS2**
- 1271

CE4 Please provide the editors.

CE5 Please provide the publisher location.

- 1272 45. Leonard T (2003) Building an AI Sensory System: Examining the
1273 Deign of Thief: The Dark Project. In: Proceedings of the 2003
1274 Game Developers' Conference, San Jose
- 1275 46. Loyall B (1997) Believable Agents: Building Interactive Person-
1276 alities. PhD Thesis, Carnegie Melon University, **CE5**
- 1277 47. Määta A (2002) Realistic Level Design for Max Payne. In: Pro-
1278 ceedings of the 2002 Game Developer's conference, GDC
1279 2002, **CE3**
- 1280 48. Mac Namee B, Cunningham P (2003) Creating Socially Interac-
1281 tive Non Player Characters: The μ -SIC System. Int J Intell Games
1282 Simul 2(1):**TS6**
- 1283 49. Mac Namee B, Dobbyn S, Cunningham P, O'Sullivan C (2003)
1284 Simulating Virtual Humans Across Diverse Situations. In: Pro-
1285 ceedings of Intelligent Virtual Agents '03, pp 159–163 **CE3**
- 1286 50. Mac Namee B, Rooney P, Lindstrom P, Ritchie A, Boylan F,
1287 Burke G (2006) Serious Gordon: Using Serious Games to Teach
1288 Food Safety in the Kitchen. The 9th International Conference
1289 on Computer Games: AI, Animation, Mobile, Educational & Se-
1290 rious Games CGAMES06, Dublin
- 1291 51. Magerko B, Laird JE, Assanie M, Kerfoot A, Stokes D (2004) AI
1292 Characters and Directors for Interactive Computer Games. The
1293 2004 Innovative Applications of Artificial Intelligence Confer-
1294 ence. AAAI Press, San Jose
- 1295 52. Thalmann MN, Thalmann D (1994) Artificial Life and Virtual Re-
1296 ality. Wiley **CE5**
- 1297 53. Michael D, Chen S (2005) Serious Games: Games That Educate,
1298 Train, and Inform. Course Technology PTR **CE5**
- 1299 54. Muller J (1996) The Design of Intelligent Agents: A Layered Ap-
1300 proach. Springer **CE5**
- 1301 55. Nareyek A (2001) Constraint Based Agents. Springer **CE5**
- 1302 56. Nareyek A (2007) Game AI is Dead. Long Live Game AI! IEEE
1303 Intell Syst 22(1):9–11
- 1304 57. Nieborg D (2004) America's Army: More Than a Game. Bridging
1305 the Gap; Transforming Knowledge into Action through Gam-
1306 ing and Simulation. Proceedings of the 35th Conference of the
1307 International Simulation and Gaming Association (ISAGA), Mu-
1308 nich
- 1309 58. Ortony A, Clore GL, Collins A (1988) The cognitive structure of
1310 emotions. Cambridge University Press, Cambridge
- 1311 59. Perlin K, Goldberg A (1996) Improv: A System for Scripting In-
1312 teractive Actors in Virtual Worlds. In: Proceedings of the ACM
1313 Computer Graphics Annual Conference, pp 205–216 **CE3**
- 1314 60. Proyas A (Director) (2004) I, Robot. Motion Picture. [http://www.
1315 irobotmovie.com](http://www.irobotmovie.com) **TS2**
- 1316 61. Rao AS, Georgeff MP (1991) Modeling rational agents within
1317 a BDI-architecture. In: Proceedings of Knowledge Repre-
1318 sentation and Reasoning (KR&R-91). Morgan Kaufmann **CE3**,
1319 pp 473–484
- 1320 62. Musse RS, Thalmann D (2001) A Behavioral Model for Real Time
1321 Simulation of Virtual Human Crowds. IEEE Trans Vis Comput
1322 Graph 7(2):152–164
- 1323 63. Reed C, Geisler B (2003) Jumping, Climbing, and Tactical Rea-
1324 soning: How to Get More Out of a Navigation System. In: Ra-
1325 bin S (ed) AI Game Programming Wisdom 2. Charles River
1326 Media **CE5**
- 1327 64. Reynolds C (1987) Flocks, Herds and Schools: A Distributed Be-
1328 havioral Model. Comput Graph 21(4):25–34
- 1329 65. Rodriguez R (Director) (1996) From Dusk 'Till Dawn. Motion Pic-
1330 ture
- 1331 66. Rosenbloom P, Laird J, Newell A (1993) The SOAR Papers: Read-
1332 ings on Integrated Intelligence. MIT Press, **CE5**
67. Sánchez-Crespo D (2006) GDC: Physical Gameplay in Half-Life
1333 2. Retrieved October 2007, from gamasutra.com: [http://www.
1334 gamasutra.com/features/20060329/sanchez_01.shtml](http://www.gamasutra.com/features/20060329/sanchez_01.shtml)
1335
68. Shao W, Terzopoulos D (2005) Autonomous Pedestrians. In:
1336 Proceedings of SIGGRAPH/EG Symposium on Computer Ani-
1337 mation, SCA'05, pp 19–28 **CE3**
1338
69. Silverman BG, Bharathy G, O'Brien K, Cornwell J (2006) Human
1339 Behavior Models for Agents in Simulators and Games: Part II:
1340 Gamebot Engineering with PMFserv. Presence Teleoper Virtual
1341 Worlds 15(2):163–185
1342
70. Silverman BG, Johns M, Cornwell J, O'Brien K (2006) Human Be-
1343 havior Models for Agents in Simulators and Games: Part I: En-
1344 abling Science with PMFserv. Presence Teleoper Virtual Envi-
1345 ron 15(2):139–162
1346
71. Smith P (2002) Polygon Soup for the Programmer's Soul: 3D
1347 Path Finding. In: Proceedings of the Game Developer's Confer-
1348 ence 2002, GDC2002, **CE3**
1349
72. Snavelly P (2002) Agent Cooperation in FSMs for Baseball. In:
1350 Rabin S (ed) AI Game Programming Wisdom. Charles River
1351 Media **CE5**
1352
73. Stanley KO, Bryant BD, Karpov I, Miikkulainen R (2006) Real-
1353 Time Evolution of Neural Networks in the NERO Video Game.
1354 In: Proceedings of the Twenty-First National Conference on Ar-
1355 tificial Intelligence, AAAI-2006. AAAI Press, pp 1671–1674 **CE3**
1356
74. Stout B (1996) Smart Moves: Intelligent Path-Finding. Game
1357 Dev Mag Oct **TS7**
1358
75. Takahashi TS (1992) Behavior Simulation by Network Model.
1359 Memoirs of Kougakuin University 73, pp 213–220 **CE3**
1360
76. Terzopoulos D, Tu X, Grzeszczuk R (1994) Artificial Fishes with
1361 Autonomous Locomotion, Perception, Behavior and Learning,
1362 in a Physical World. In: Proceedings of the Artificial Life IV Work-
1363 shop. MIT Press **CE5**
1364
77. Thompson C (2007) Halo 3: How Microsoft Labs Invented a
1365 New Science of Play. Retrieved October 2007, from wired.com:
1366 [http://www.wired.com/gaming/virtualworlds/magazine/
1367 15-09/ff_halo](http://www.wired.com/gaming/virtualworlds/magazine/15-09/ff_halo)
1368
78. Toth J, Graham N, van Lent M (2003) Leveraging gaming in
1369 DOD modelling and simulation: Integrating performance and
1370 behavior moderator functions into a general cognitive archi-
1371 tecture of playing and non-playing characters. Twelfth Confer-
1372 ence on Behavior Representation in Modeling and Simulation
1373 (BRIMS, formerly CGF), Scottsdale, Arizona
1374
79. Valdes R (2004) In the Mind of the Enemy: The Artificial
1375 Intelligence of Halo 2. Retrieved October 2007, from How-
1376 StuffWorks.com: [http://entertainment.howstuffworks.com/
1377 halo2-ai.htm](http://entertainment.howstuffworks.com/halo2-ai.htm)
1378
80. van der Werf E, Uiterwijk J, van den Herik J (2002) Program-
1379 ming a Computer to Play and Solve Ponnuki-Go. In: Proceed-
1380 ings of Game-On 2002: The 3rd International Conference on
1381 Intelligent Games and Simulation, pp 173–177 **CE3**
1382
81. van Lent M, McAlinden R, Brobst P (2004) Enhancing the be-
1383 havioral fidelity of synthetic entities with human behavior
1384 models. Thirteenth Conference on Behavior Representation in
1385 Modeling and Simulation (BRIMS) **CE3**
1386
82. Woodcock S (2000) AI Roundtable Moderator's Report. In:
1387 Proceedings of the Game Developer's Conference 2000
1388 (GDC2000), **CE3**
1389
83. Wooldridge M, Jennings N (1995) Intelligent Agents: Theory
1390 and Practice. Know Eng Rev 10(2):115–152
1391

TS6 Please provide page number.

TS7 Please provide page numbers

- 1392 84. Yerkes RW, Dodson JD (1908) The relation of strength of stim-
1393 ulus to rapidity of habit formation. *J Comp Neurol Psychol*
1394 18:459–482
- 1395 85. Zubek R, Horswill I (2005) Hierarchical Parallel Markov Models
1396 of Interaction. In: *Proceedings of the Artificial Intelligence and*
1397 *Interactive Digital Entertainment Conference, AIIDE 2005*, [CES](#)

1398 **Books and Reviews**

- 1399 DeLoura M (ed) (2000) *Game Programming Gems*. Charles River
1400 Media [CES](#)
- 1401 DeLoura M (ed) (2001) *Game Programming Gems 2*. Charles River
1402 Media [CES](#)
- 1403 Dickheiser M (ed) (2006) *Game Programming Gems 6*. Charles River
1404 Media [CES](#)
- 1405 Kirmse A (ed) (2004) *Game Programming Gems 4*. Charles River
1406 Media [CES](#)
- 1407 Pallister K (ed) (2005) *Game Programming Gems 5*. Charles River
1408 Media [CES](#)
- 1409 Rabin S (ed) (2002) *Game AI Wisdom*. Charles River Media [CES](#)
- 1410 Rabin S (ed) (2003) *Game AI Wisdom 2*. Charles River Media [CES](#)
- 1411 Rabin S (ed) (2006) *Game AI Wisdom 3*. Charles River Media [CES](#)
- 1412 Russell S, Norvig P (2002) *Artificial Intelligence: A Modern Ap-*
1413 *proach*. Prentice Hall [CES](#)
- 1414 Treglia D (ed) (2002) *Game Programming Gems 3*. Charles River
1415 Media [CES](#)

Uncorrected Proof
2008-07-25