In *Proceedings of the IEEE 2005 Symposium on Computational Intelligence and Games (CIG'05)*.
Piscataway, NJ: IEEE.

Academic AI and Video games: a case study of incorporating innovative academic research into a video game prototype

Aliza Gold
Digital Media Collaboratory, IC[2] Institute
University of Texas at Austin
2815 San Gabriel Street
Austin, TX 78705
aliza@icc.utexas.edu

Abstract- Artificial intelligence research and video games are a natural match, and academia is a fertile place to blend game production and academic research. Game development tools and processes are valuable for applied AI research projects, and university departments can create opportunities for student-led, team-based project work that draws on students' interest in video games. The Digital Media Collaboratory at the University of Texas at Austin has developed a project in which academic AI research was incorporated into a video game production process that is repeatable in other universities. This process has yielded results that advance the field of machine learning as well as the state of the art in video games. This is a case study of the process and the project that originated it, outlining methods, results, and benefits in order to encourage the use of the model elsewhere.

1 Introduction

Video game technology can provide a rich platform for validating and advancing theoretical AI research (Adobbati et al. 2001, Laird 2001, Isla & Blumberg 2002, Fogel 2003). Therefore, projects that can bring the areas of artificial intelligence research and video game technology together can uniquely benefit academia. These benefits can sometimes extend to industry as well. A research and development team at the Digital Media Collaboratory (DMC) in the IC[2] Institute at the University of Texas at Austin has developed a process for utilizing video game production techniques to both apply and advance academic AI research. The process has yielded successful results and demonstrates one way to bring the two domains together. In the NeuroEvolving Robotic Operatives (NERO) project, a team of student artists and programmers has created a prototype game based on a Ph.D. candidate's novel neuroevolution (i.e. the evolution of neural networks with a genetic algorithm) method. Although this ongoing project has not been funded by an industry group, it has incorporated industry needs into its goals. It is possible to blend game development and research, and building 'killer apps' for industry is within reach for academia, without sacrificing research results. In fact, this project produced dissertation-level research that could not have been accomplished without the use of an off-the-shelf game engine as a test-bed.

Projects like NERO are repeatable. Currently there is a unique confluence of the ideas, people, tools, and technology required to implement similar projects. Many AI researchers today have grown up playing video games and are motivated to improve them, and at the same time many university departments are beginning to see video games as a legitimate medium, worthy of study and research (Laird & van Lent 2000). These schools are bringing people from the game industry to their programs so that game industry veterans and academic research students can work alongside each other. Simultaneously, some game development tool companies have forged new business models that allow them to price their products in ways that university programs can afford. And finally, exponential increases in processor power have enabled advanced AI to run in real time on personal computers.

Thus there is an opportunity to create projects in academia that yield potential benefits for both research and commercial interests. Current university projects reflect this interest in using computer games as a test-bed for AI research (Adobbati et al. 2001, Geisler 2002, McLean 2003). These articles report results in applied AI research, however few articles document the development processes used in obtaining the results. Discussing project development processes can aid groups in other universities to conduct applied artificial intelligence research projects with video game test beds successfully. This article outlines the process the DMC has employed to develop such a project. It describes the approaches used, challenges encountered, and makes suggestions for implementing projects such as these in other academic environments.

2 Background

When initiating NERO, the project team was motivated by simultaneous goals: 1) achieve tangible academic research results, and 2) create an application that could demonstrate and advance the state of the art of artificial intelligence in the video game industry. The project leaders hypothesized that video game production techniques combined with academic research methods could achieve these goals.

Computer and video game entertainment software originally evolved from both action-oriented digital games played in arcades and from fantasy role-playing gamers who used the computer as a digital storytelling medium (Garriott 2002, King 2003). These games were small and could be programmed by one person or a team of a few people. Over the past 20 years, computer and video games have gained enough popularity to be considered a mainstream entertainment medium provided for by an industry with revenues of 7 billion US dollars in 2003 (ESA 2004). The most popular video games today are complex systems, and developing them requires large project teams of up to 100 people (Bethke 2003) with the associated project management infrastructure. Large teams producing complex projects require management processes in order to reliably produce results within the constraints of a schedule and budget. Formal software engineering processes have been applied as useful paradigms for projects in the commercial sector, including video game development.

In the NERO project, the team attempted to use standard software development models utilized in video game production with varying success as discussed in Section 3.2.2. The examples defined here provide clarity in the discussion of NERO:

- Waterfall method. This model was first introduced by Royce (1970) as software projects grew in size and complexity. It stipulates that projects follow distinct phases of requirements gathering, preliminary program design, analysis, coding, and testing, with documentation produced at the end of each phase. Each phase is completed before the next phase begins.
- Spiral model. Boehm (1988) developed the spiral model as a way to manage the risk that arises from uncertainties in a project. The spiral begins with well-defined exploratory phases in which risks are assessed and mitigations determined. As more definition for the software is achieved, a series of prototypes may be developed iteratively or it may be appropriate to transition to a waterfall or other model.
- Incremental method. This is a parallel process method used primarily on large projects. Rather than an entire project team following the waterfall method, the coding of separate segments of an application's system are begun according to when requirements can be defined for them.

Additional concepts from management science for the commercial sector are also useful in understanding the circumstances in which the NERO project was developed:

- Interdisciplinary teams. Also known as cross-functional teams, these are comprised of members with expertise from a variety of domains relevant to the project.
- Skunkworks. This term refers to a process engineering method innovation pioneered in the 1940s at Lockheed Martin (Brown 2001). It is characterised by the use of a small group completing a prototype or project from beginning to end (Wolff 1987). The team is given its own independent space and autonomy from the bureaucratic norms of the larger organization.

The waterfall method has been a useful tool for the game industry as game projects coordinate the efforts of designers, programmers, and artists to create products that are both functional and engaging for game-playing audiences. All groups contribute to the design of the product and group understanding of the final design is crucial to the success of the project. However, a video game can consist of numerous sub-systems within the larger product, and risk management strategies like those in the spiral method are sometimes utilized, as well as incremental strategies allowing different systems modules to develop within their own time cycles inside the larger project (Gattis & Waldrip 2004). The NERO project leadership experimented with all of these software development models. Some proved useful, while others presented unique challenges in an academic environment.

3 Project NERO

3.1 From NEAT to NERO

Laird (2000) listed three reasons that the closer ties and working relationships between the video game industry and the artificial intelligence research community that many parties identify as mutually beneficial has recently seen an increase in activity: 1) advancing technology with better processor power, 2) student gamers now in universities are doing AI research, and 3) players are demanding better AI in video games.

This change was the impetus for a workshop held by the DMC in August of 2003 on the topic of AI in Video Games, where University of Texas at Austin researchers and game industry developers presented their research in a single conference track. One of the researchers was then Ph.D. candidate Kenneth O. Stanley (completed August, 2004), who presented his method for evolving neural networks called NeuroEvolution of Augmenting Topologies (NEAT; Stanley & Miikkulainen 2002). Mat Buckland, in his 2002 book *AI Techniques for Game Programming* posited the potential for NEAT to improve AI in video games. By the time of the workshop Stanley had conceived of a game that would require machine learning capabilities that intelligent agents trained with the NEAT method could provide. In breakout sessions for game design he proposed the game idea and then presented it to the conference. After the conference, the DMC decided to create a prototype of Stanley's game idea, instantiating an applied research project that would build on Stanley's basic NEAT research and create a real-time implementation of it (Stanley et al. 2004). This project came to be called NERO, based on the game concept of training and battling robotic armies: NeuroEvolving Robotic Operatives.

Video game prototypes or demos often function as a demonstration of the fully developed game and as a proof of concept for the technology behind the game (Bethke

Figure 1: A screenshot from a training scenario in the NERO prototype in which AI agents learn to find their way around obstacles to a goal

2003). The NERO project was envisioned similarly, and by conducting the implementation of the AI method into a game software environment, the project generated tangible research, real-time NEAT (rtNEAT), a new derivative of NEAT which was reported in Stanley's dissertation (Stanley 2004).

3.2 Development Methods

3.2.1 Environment/Context

The skunkworks construct (Section 2) has many similarities to the project circumstances and environment at the DMC. The physical location of the project has always been the DMC lab, where team members meet and work individually and collaboratively. The DMC is part of a university research unit, not a department in a college, and thus does not need to follow departmental norms. The DMC director has given few directives to the project staff except for requesting a target milestone date to have a functional prototype completed. The core leadership of the NERO team has been the same since the inception, as in Wolff (1987), described in Section 2. The team's work has been design and prototyping.

3.2.2 Process

The DMC team's original hypothesis was that applying game development team structures and typical game software development methods (Bethke 2003) to an applied AI research project would yield constructive results. Thus the leadership began with the waterfall method, as most video game industry projects do, to confirm requirements and create a preliminary design. However, the team immediately encountered difficulty with the waterfall method, since it was unclear who would be the ultimate customer of this technology, and whether the DMC would release NERO as a game eventually. Similar problems arose when attempting high-level game design because at the beginning the team did not have enough knowledge about how NEAT would work in the game engine, or if it could be made to work in real time. Thus starting with the typical waterfall phases of requirements and preliminary design, as game projects normally would, was not appropriate for the NERO project.

As noted by McCormack and Verganti (2003) in their discussion of uncertainty in projects and how they affect the development process, "different types of projects carried out in different environments are likely to require quite different development processes if they are to be successful." This is echoed by game developers who acknowledge that video games, for which custom software systems are often built, frequently require a custom mix of development approaches (Gattis & Waldrip 2004).

The NERO leadership team evolved an approach suited to the research and development challenges the project has faced. The team adapted the spiral method, defining specific intermediate goals for research and production, and choosing target completion dates. Since the university semester schedule imposes long breaks and team turnover among student participants, semesters provide natural milestones for the project in which the team can evaluate progress, identify the next round of goals and tasks, and evaluate which tasks to tackle immediately and which to postpone based on current resources.

Implementing the NEAT AI technique in a 3D game engine required crafting a series of experiments that would both validate and advance the capabilities of NEAT. At the same time, choosing the commercial engine Torque (Garage Games) as a development platform and research test-bed required many modifications to accommodate rtNEAT and implement the NERO game design around it. NERO has incorporated academic research goals, alongside concurrent production involving engine coding

Experiments -
E1 – Learn to move to approach some stationary target.
For convenience, just use an enemy for the target.
E2 – Learn to acquire and shoot to hit a stationary target.
May not require a factory.
b – Ditto, except with a mobile target.
E3 – Learn to approach a stationary turret that is firing in an oscillating pattern, without getting killed.
b – Ditto, except against two turrets.
E4 – Learn basic assault tactics: move across the map into firing range and destroy a stationary turret that is firing in an oscillating pattern, without getting killed.
b – Ditto, except against two turrets.
E5 – Deploy two teams trained as in E4b in a combat area, and show that they display appropriate behavior.
No evolution involved.
Use teams trained to different behaviors, e.g. "move a lot" vs. "don't move much", "disperse a lot" vs. "stay bunched up", or "approach the enemy" vs. "avoid the enemy".

Figure 2: An excerpt from the first plan of experiments

and art creation. The team leadership also has adapted an incremental approach to the project by having some programmers work individually or in small groups on different parts of the prototype, some conducting research experiments, while others work on the user interface or other game-engine related tasks. The research aspect of the project has taken a large percentage of the total effort and often drives the other development tasks. As the project progressed, the application increasingly took on the appearance of a game while research simultaneously advanced (Figure 1).

3.2.3 Team Structure

The game industry relies on interdisciplinary teams (Section 2) to develop complex entertainment software products. The NERO project has used team structure and development methods borrowed from the game industry, but on a much smaller scale and in an academic environment. Thus for NERO a team-based structure has been employed, with subteams and associated team leads for programming, art, and design. Almost all of the team members have been undergraduate student volunteers with no professional experience in industry. Of those who are not volunteers, the project producer and the lead designer are staff members of the DMC, and the AI researchers Kenneth Stanley and Bobby D. Bryant, Ph.D. candidate, are both on fellowships. The project was not prohibitively costly to implement since volunteers and staff members completed much of the work.

For the art team and programming team leadership positions, which sometimes change, the producer tapped willing volunteers. They usually had only classroom experience in project work and leadership. However the result of giving undergraduates positions of responsibility has been that they perform beyond expectations, carrying out research and solving difficult problems (Kanter 1985).

The programming and art leads make up half of the project leadership team. Stanley and Bryant, the producer, and the designer make up the rest of the leadership team, who meet weekly to make decisions and coordinate all aspects of the project. These meetings are crucial for directing the production work, staying abreast of the research progress, and continually evaluating how the advancing research impacts the design and production. The DMC lab has been the center of operations for all meetings and most of the project work.

3.2.4 Design Considerations

For the NERO project, there have been two distinct but related areas of design – the game design aspect of the prototype and the research design. The game designer has created a context for Stanley's game concept, which included background fiction, a visual design concept, and some game design elements. This vision is a motivator for the volunteer team, who are excited to work toward a real game. The two AI researchers have had the responsibility of designing the series of experiments that document and validate the success or lack thereof of implementing NEAT into the Torque game environment and exhibiting the desired capabilities (Figure 2 shows an example of an experiment series). The game design and the research design have been intertwined throughout development because what has been possible in the game depended on the capabilities of the artificial intelligence. As research and production have continued, new possibilities have presented themselves, based on what has been accomplished in the research. Sometimes new research directions have appeared, which implied accompanying shifts in the game design. This has contributed to the usefulness of the spiral development method for this project, where design choices are made iteratively, prototypes are produced to reflect the new choices, and risks are evaluated at each turn (Boehm 1998).

3.2.5 Results

The NERO project has resulted in a playable video game prototype, now its second version. An exciting result is that the novel aspect of rtNEAT has yielded novel gameplay in the prototype (NERO will be presented in the Experimental Gameplay Workshop at the 2005 Game Developers Conference).

In NERO, a player trains a group of ignorant robot soldiers by setting learning objectives for the group through an interface. After the objective is set, the robots learn in real time to achieve their goal. The player can incrementally increase the challenge for the robot soldiers though the learning objective interface and by customizing the training arena, thus creating increasingly sophisticated behavior in the group. Once the real-time training is completed to the player's satisfaction, the player can save the team. Then the player can battle the team against the computer's or a human opponent's team, observing the results of the training in a battle situation, where learning is no longer taking place. The real-time training in NERO, powered by rtNEAT, is a type of gameplay that currently does not exist in commercial games. This new style of gameplay could lead to new video game genres.

NERO has been a successful project because it has resulted in innovations in terms of both research and what is possible in games. NEAT first had to be reengineered to work in real time, producing one of the first genetic algorithm systems to do so. Stanley created rtNEAT, an innovation that he included in his doctoral dissertation (Stanley 2004), and with the undergraduate lead programmer implemented the method into the Torque 3D engine. Sensors had to be engineered to enable the agents to perceive elements of the environment such as enemies or walls. The agents perceive the world differently than typical non-player characters in a video game that are fed information about the environment through the game system. NERO agents sense the environment egocentrically, more like robots in the physical world, and thus had to be equipped with egocentric sensors.

Another essential element of the training function of the game is the interface where players, from their perspective, assign objectives to the agents during training, such as approaching the on-field enemy or avoiding getting hit by fire. This interface gives players access, without their explicit knowledge, to the fitness functions of the agents, allowing them to interact with the evolution of the networks in real time, another innovation enabled by rtNEAT (Stanley et al. 2004).

Additionally, art students have created robot models to embody the agents, and game level environments to more aptly demonstrate the agents' capabilities. Thus at this point NERO is a successful demonstration of NEAT, and as such has generated considerable interest from both companies in the game industry and groups in the military. A provisional patent has been filed on rtNEAT, an SDK is being developed, and licensing options are being investigated. NERO has been an experiment on multiple levels: artificial intelligence research, process management, and video game design. At each level interesting results have emerged.

4 Discussion

4.1 Project-Based Considerations

Over the yearlong course of developing the NERO project, there has been no road map to tell the leadership how to direct a project of this kind. Some of the solutions the leadership has developed may be generalizable to other projects.

4.1.1 Industry Input

Other projects of this kind would benefit from getting input from industry early in the project. Whether through casual conversations or formal presentations with industry professionals, input concerning what would be desirable or innovative is essential if a goal is to eventually commercialize the research. It would be undesirable for academic groups who are unfamiliar with video games to attempt to solve problems that are considered already solved in the game industry. Posing a question such as "If our group had a technology that could do [X technology functionality], what would you expect it to be capable of in a game environment?" would yield valuable information that a team could use to guide the development of the project by comparing industry interests to the group's own research interests.

4.1.2 The Research and Production Mix

In a project where the goal is to create a technology that interests industry while at the same time advancing research in an academic field, maintaining awareness of those sometimes opposing goals is important. In the NERO project the leadership team has experienced the tension between those goals, and that has been an occasional source of conflict. In any project phase, there is a limit to the resources available to apply to project tasks, and choices must be made and priority given to some tasks over others. Transition points between phases are the best time to evaluate the meta-goals of the project for the upcoming phase.

4.1.3 Technology Considerations

Using an existing game engine (Torque) as a platform for applying new artificial intelligence methods has been very helpful for the NERO team. It has allowed the team to develop a functional prototype fairly quickly after the effort of making NEAT workable in Torque. Additionally, using a commercial engine gives our undergraduate programmers the experience of working with an industry quality engine, which is valuable for industry hopefuls. However, it is also important to be aware of the limitations of the chosen engine or platform. The Torque engine is designed for the first-person shooter game genre, a very different kind of game than NERO is intended to be; thus some functionality that would ideally already exist does not. For instance, the training aspect of NERO requires the player to build training environments, and an engine similar to the one used in the video game SimCity would have that functionality built in. Currently there is not an affordable engine for every existing game genre, and thus much functionality may need to be built from scratch.

Alongside functionality considerations are other technical concerns that should be understood when evaluating engine platforms. In Torque, the STL (Standard Template Library) is not supported, which required that the NEAT code be rewritten so as not to use that library. This was a surmountable challenge, but non-trivial, and has kept some researchers from wanting to use the engine.

4.1.4 Process

When dealing with research experiments it is much more difficult to create a reliable schedule based on the estimated level of effort, as would be done with a commercial project. Encountering unexpected difficulties when conducting experiments can extend the time to complete them far beyond what a team lead might imagine or plan. The NERO leadership has found it more productive to create a milestone list of experiments and production work based on the semester, without assigning specific completion dates. Together the leads create an agreed-upon list of goals and track the completion of them throughout the semester.

4.1.5 Team

The primary difficulties encountered when working with an undergraduate student volunteer team are inexperience and turnover. The NERO project leadership has developed processes and techniques for bringing on new students and retaining them:

1. Recruiting process. It is most efficient to bring on new programmers and train them as a group rather than individually. That also gives them camaraderie as new members of the team.
2. High expectations from the outset. Too many uncommitted people on the team are an administrative overload for the lead. By broadcasting the need for commitment to the project, people know what to expect and the lead can limit participation to members who are likely to stay on the project and contribute significantly (Botkin 1985).
3. Weekly meetings. These are important for each subteam to keep people on track and help them understand their contributions in a group context. Meetings are especially important for team members who lack experience working in a group toward a common goal.
4. Mentorship. Create an environment where the more senior members can advise newer ones; with new members joining the team each semester, it can take some of the burden off the lead to establish times where most members are in the work space, thus enabling mentoring opportunities.
5. Communication. Project leaders improve morale by communicating the meta-goals for the project phase and gathering team members' input. As implementers, team members have valuable insights and perspectives of which the project leadership needs to be aware.
6. Fun. Project leaders can demonstrate their appreciation to the team by hosting game nights, bringing in industry speakers, and recognizing accomplishments whenever possible.

While some might be concerned that having volunteers on the project is exploitative, it is apparent that the experience on the project has proved beneficial for undergraduate team members. Game projects appeal to undergraduates, and many students hope to find employment in the game industry, which is notoriously difficult to enter. Working on a game engine with a team provides valuable experience toward that goal. Other students who are less driven by game industry goals value the opportunity to work with Ph.D.-candidate-level graduate students doing innovative research, which many undergraduates find inspiring. Undergraduates are a frequently overlooked group; however, they have proved to be crucial contributors to the NERO project team. NERO project veterans have gone on to create their own lines of research based on the project or, following their graduation, have found employment with local game development studios. The lessons learned with NERO are serving the project team well as it goes forward into a second year of development, reducing the time spent on administrative necessities such as student recruiting, and allowing more time for research and game production work.

4.2 Institutional Considerations

Accompanying the project-based considerations in conducting a project like NERO, there are also issues at the institutional level. There is an increasing movement in academia, in the US, Europe, and elsewhere, to commercialize university research, a movement that has created controversy in the academic community. Proponents cite the important benefits for society, such as bringing technology improvements to the public, job creation, and funding for universities. However, there are serious concerns among academic leaders that market-driven influences will corrupt academic research processes with conflicts of interest and manipulated research agendas (see McMaster 2002, Bok 2003). Nevertheless, NERO has shown that it is possible to create university/industry collaborations that can provide educational benefits to students as well as financial benefits to departments, without compromising basic research.

Dr. George Kozmetsky, the founder of the University of Texas at Austin's IC^2 Institute, promoted an agenda of capitalism for the public good. The Institute has a long history of technology commercialization with groups under its umbrella such as the Austin Technology Incubator and the Masters of Science in Science and Technology Commercialization degree program. The DMC is the newest organization within the IC^2 Institute, created in part to apply basic research with the University of Texas' faculty and students and bring it to the public/commercial sector. The DMC is an obvious organization to implement projects of this kind, and the NERO project has been an initial implementation of Dr. Kozmetsky's vision.

The DMC has provided a fertile ground for implementing NERO as an academic applied research project that supplies some of the project's research goals by observing industry needs. University departments anywhere can develop similar projects by creating the necessary infrastructure to implement them. Departments can create opportunities for applied research projects outside of students' normal classwork, supplementing their knowledge acquisition with collaborative, project-based educational experiences. Creating this infrastructure does not require a large investment of departmental resources.

The purpose of having a structure in which projects occur outside of classes is to provide an avenue for the creation of projects that develop for longer than a semester. It also allows students to be driven by motivations other than external class-related measures, such as receiving grades.

In order to create such an infrastructure, university departments could institute processes such as these:

- Create a process to invite applied research project ideas, requiring a group implementation, from graduate students and faculty, based on their research.
- Have a committee that includes faculty and industry members to choose the best ideas based on industry interests and advancements in the relevant academic field.
- Choose a project manager to create teams based on the chosen project ideas. This coordinator could be a student in the department, a faculty member, or a student from another major such as business. This role requires commitment, deep interest in the

research area, and organizational and communication skills. It may be advisable to assign a departmental staff member the job of supervising these project coordinators to provide mentorship and guidance.

- Give students autonomy in how they develop their projects by allowing project leaders to set goals in collaboration with the other team leads.
- Have a review process at appropriate milestones with team members, faculty members, and industry members in order to assess project progress.

To broaden the participation of students from other disciplines, invite students from departments such as communications, art, business, design, and information sciences. All of these departments represent roles in game industry projects, which can increase the real world impact of the project.

5 Implications and Future Work

The NERO project is an example of the advantages of leveraging existing technology (e.g. off-the-shelf game engine applications) for advancing academic research. Such technologies, which have become increasingly more affordable for university department budgets, can supply one researcher or a team with the foundation to relatively quickly implement research technologies such as machine learning, robotics, databases, or other areas. These technologies provide platforms for visualization and validation, and because they support high-quality demonstrations, these platforms can be a bridge to commercialization of research technologies.

After implementing rtNEAT in the NERO game, other potential applications have become apparent, even within the realm of video game technology. The rtNEAT method could be implemented in a persistent online 3D environment. Massively multiplayer online games (MMOGs) are one possibility for such an implementation. In a multiplayer persistent virtual world, the agents, as non-player characters, could evolve new behaviors by reacting to players' choices. Non-player-characters with true adaptive, not scripted, behavior are very desirable for game developers and players alike (Ilsa & Blumberg 2002, Fogel 2003). Non-player-characters in current MMOs have predictable behaviors with a limited ability to provide players with new challenges. To address game designer needs, a back-end interface to the evolution similar to the NERO interface would ensure that designers had the necessary control over the agents' behavior, which is an important industry consideration.

There are also implications for non-game environments. By modeling the agents as animals or other biological entities and giving them appropriate animations, the adaptation ability shown in NERO might result in artificial life agents who could evolve for weeks, months, or perhaps years, based on the concept of open-ended evolution (Taylor 1999). This sort of artificial ecology experiment would be an exciting next step that could provide a test-bed for exploring the limits of rtNEAT's complexification characteristics.

Another conclusion of this case study is that closer collaboration with the video game industry could bring significant benefits to the academic community. One of these benefits is in the area of applied research problems. Most AI machine learning researchers who have worked with games until recently have used board games such as checkers or Go (Furnkranz 2000) as research areas. In contrast, the video game industry offers complex and novel application challenges, which have "real world" status that can motivate student teams who are interested in advancing the state of the art in game technology.

Game companies might be persuaded to provide funding to departments engaged in projects that could produce usable technologies if they understood the benefit of having highly specialized academic research applied to the video game domain on problems they need solved. Again, departments must always weigh industry interests against the interests of advancing the research in the relevant academic field, and attempt to find areas of commonality.

6 Conclusions

Video game technology has given the rtNEAT method a platform to demonstrate its capabilities, which has proved invaluable for introducing it to the world. Projects such as NERO can be repeated in other universities where there is support for innovative ways of conducting applied research in computer science or interdisciplinary departments. Universities can use video game technology to accelerate research advancement as well as reveal commercialization opportunities. These projects provide motivating challenges for students who are highly interested in doing innovative work, even as undergraduates.

University departments can draw on students' intense interest in video games. Research conducted by the DMC in 2003 shows that, among middle school students in Texas, 75% of boys and 50% of girls were found to play video games (Gold et al. 2005). With so many university students engaged as users of this medium, there is a large pool of potential team members for game-related projects.

Ultimately, as innovative artificial intelligence techniques are implemented in commercial video games and AI students and professors leverage video game technology as applied research platforms, everyone benefits.

Acknowledgments

The author thanks Risto Miikkulainen for his crucial initial support and helpful suggestions for the analysis. Bobby D. Bryant, former NERO lead programmer, Kenneth O. Stanley, and Damon Waldrip provided insightful comments on the manuscript and editorial assistance. The success of the project has depended on the contributions of Michael Chrien, NERO lead programmer, and the entire NERO team, past and present. The project has only been possible with the support of the DMC staff: Alex Cavalli,

Melinda Jackson, Aaron Thibault, and Matt Patterson, NERO designer.

Bibliography

Adobbati, R., Marshall, A., Scholer A., Tejada, S., Kaminka, G., Schaffer, S., and Sollitto, C. (2001). Gamebots: A 3D virtual world test-bed for multi-agent research. *Proceedings of the 2nd Workshop on Infrastructure for Agents, MAS, and Scalable MAS*. New York, NY: ACM Press.

Botkin, J.W. (1985) Transforming creativity into innovation: Processes, prospects, and problems. In *Frontiers in Creative and Innovative Management,* Kuhn, R. Ed. Cambridge, MA: Ballinger Publishing Company.

Bethke, E. (2003). *Game Development and Production*. Plano, TX: Worldware Publishing.

Boehm, B. W. (1988) A spiral model of software development and enhancement. *IEEE Computer*: 21.

Bok, D. (2003). *Universities in the Marketplace*. Princeton, NJ: Princeton University Press.

Brown, T. (2001). Skunk works: A sign of failure, a sign of hope? For the *Future of Innovation Studies Conference*. The Netherlands.

Buckland, Mat. (2002) AI techniques for game programming. Boston, MA: Premier Press.

Entertainment Software Association. (2004). *Essential Facts about the Computer and Video game Industry*.

Fogel, D. (2003). Evolutionary entertainment with intelligent agents. *IEEE Computer:* 36.

Furnkranz, J. (2000). Machine learning in games: A survey. Austrian Research Institute for Artificial Intelligence. Technical Report OEFAI-TR-2000-31.

Garriott, R. (2002). The origins of Ultima. Presentation at the *Digital Media Collaboratory GameDev 2002 Workshop*.

Gattis, G., and Waldrip, D. Interviewed by author, October 10, 2004.

Geisler, B. (2002). An empirical study of machine learning algorithms applied to modeling player behavior in a "first-person shooter" video game. Masters thesis, Department of Computer Science, University of Wisconsin-Madison.

Gold, A., Durden, E. Alluah, S., Kase, M. Career aspirations and their influences among central texas middle school youth. To be presented at the *American Educational Research Association 2005 Conference*.

Isla, D. & Blumberg, B., (2002). New challenges for character-based AI for games. *AAAI Spring Symposium on AI and Interactive Entertainment*, 2002

Kanter, R.M. (1985). Change-Master skills: What it takes to be creative. In *Handbook for Creative and Innovative Managers*. Kuhn, R. Ed. New York, NY: McGraw-Hill.

King, B. and Borland, J. (2003). *Dungeons and Dreamers: The Rise of Video Game Culture*. Emeryville, CA: Mcgraw-Hill/ Osborne.

Laird, J. E. (2000). Better AI development: Bridging the gap. *Game Developer Magazine*. August 2000.

Laird, J. E. (2001). Using a computer game to develop advanced AI. *IEEE Computer:* 34.

Laird, J. E., and van Lent, M. (2000). Human-level AI's killer application: Interactive computer games. In *Proceedings of the 17th National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press.

MacCormack, A., and Verganti, R. (2003). Managing the sources of uncertainty: Matching process and context in software development. *Journal of Product Innovation Management, 20: 217-232*.

McLean, S. (2003). Multi-agent cooperation using trickle-down utility. Ph.D. thesis, Division of Informatics, University of Edinburgh.

McMaster, G. (2002) The balancing act of commercializing research. *Atlanta Express News,* February 8.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings, IEEE WESCON,* August: 1-9.

Stanley K. O. (2004). Efficient evolution of neural networks through complexification. Department of Computer Sciences, University of Texas at Austin. Technical Report AI-TR-04-39.

Stanley, K. O., Bryant, B. D., and Miikkulainen, R. (2004). The NERO real-time video game. To appear in: *IEEE Transactions on Evolutionary Computation*, special issue on *Evolutionary Computation and Games*.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation,* 10(2).

Taylor, T. (1999) From Artificial Evolution to Artificial Life. Ph.D. thesis, Division of Informatics, University of Edinburgh.