Entertainment R&D for Defense

Michael Zyda, John Hiles, Alex Mayberry, Casey Wardynski, Michael Capps, Brian Osborn, Russell Shilling, Martin Robaszewski, and Margaret Davis
The Moves Institute

We discuss *America's Army,* an Internet-based PC game suite developed by the Moves Institute under the auspices of the US Army.

In response to the US National Research Council's 1997 report that specified a research agenda for defense and entertainment, the Modeling, Virtual Environments, and Simulation (Moves) Institute proposed two games—*America's Army: Operations* and *America's Army: Soldiers*—to recreate the US Army for the benefit of young civilians. (For more information on the council's report and how this idea came about, see the sidebar "*America's Army*: A Background.")

Our goal within *America's Army: Operations* was to demonstrate life in the infantry. The idea for the game took shape as a first-person mission experience that starts with training. For example, a player can't use a particular weapon without first qualifying on the appropriate range.

We conceived *America's Army: Soldiers* as a realistic look at army personal and career opportunities via sophisticated role-playing. The player creates a character with which to identify, much like the popular game *The Sims*, and moves through an interactive story constructed spontaneously in response to the character's goals, resources, and values. (The character's values are based on the Army's core values: loyalty, duty, respect, service, honor, integrity, and courage.)

Moves named the overall project the Army Game Project (AGP). For the project, Moves envisioned a fully 3D, accurate gaming environment with technological efforts more complex than previous attempts. Our goal was deep immersion in stories that were sensible-yet-surprising, interactive, and nonrepeating. We also wanted to ascertain whether we could compute the aptitude of users by their proficiency in play.

The two games were built in secret for 24 months until we obtained results, secured approvals from the Army, and debuted the suite at the Electronics Entertainment Expo (E3) in May 2002.

Building the R&D teams

The first funding for AGP arrived in May 2000. At that time, a core group at Moves had constructed the large-scale networked virtual environment, NPSNET,[1] and we had brought in John Hiles (who had worked at Maxis on *SimCity Supreme*, *SimAnt*, and *SimFarm*).

Moves began building a development team by networking with friends and consultants. We recruited veteran artists, designers, and programmers from industry giants such as Maxis, Electronic Arts, Sony, and Kalisto. Between them, their expertise had yielded some 31 commercial games. On the research side, graduate students from all branches of the US military and a number of allied countries worked with Moves faculty on the tough technical issues underlying the project. Practical support from the Army included unprecedented access to posts, equipment, and subject-matter experts.

Twelve months from the May 2000 start, Mike Capps and Alex Mayberry led the *Operations* team, while John Hiles led the *Soldiers* team. The *Operations* team consisted of 18 developers (level designers, artists, and game programmers) and the *Soldiers* team consisted of 10.

Moves Institute researchers and developers visited some 19 army posts and videotaped, photographed, and recorded audio of everything that moved—and didn't. The team digitized video of soldiers in training, equipment and weapons, texture details such as chipping paint, and realistic minutiae such as the appearance of sand at Fort Benning, Georgia.

The team learned to shoot M-16s and sniper rifles, hurl hand grenades, and fire mortars. They went on night parachute jumps with the troops and fed themselves to the K-9 corps (wearing padded suits). The post visits created an evermore motivated and informed group of developers, eager to attain higher levels both literally and figuratively.

Operations

America's Army: Operations begins in single-player mode as a new recruit ready to train. The player embarks on basic rifle marksmanship and combat train-

Published by the IEEE Computer Society 0272-1716/03/$17.00 © 2003 IEEE

ing (BCT) through a representation of the ranges at Fort Benning, Georgia.

The player's range score determines his advancement. If he scores poorly anywhere in training, he won't proceed to another task until his score improves. If he scores well, the player may advance to M-24 rifle qualification or US Army sniper school, where he will learn, among other things, to breathe at the right moment in the firing sequence, and get the most from an M-24 by using it on a bipod in a stationary position. (Although the M-24 can be used while moving, as is common in other games, the penalty applied to the player's accuracy is severe.)

Weaponry is represented as precisely as possible—for example, weapons must be loaded and cleared as in real life, and the load is finite (Figure 1). The BCT weapons familiarization includes the M-249 squad automatic weapon (SAW).

We achieved a high level of verisimilitude—soldiers who know Fort Benning easily recognize it within the game. For example, the *Operations*' BCT obstacle course is timed and sequenced as in real life.

Single-player training features the McKenna Military *Operations* in Urban Terrain (MOUT) course at Fort Benning, including use of the flash-bang as the player clears a dark labyrinthine building of terrorist pop-up targets. Training progresses to the US Army's airborne school 250-foot jump tower.

Hearing is believing

Operations is rich aurally as well as visually. Based on Moves' research indicating that complex, multilayered sound magnifies the sense of immersion in a simulation, the opulent sound created for *Operations* by audio designer Russell Shilling pulls you inexorably into the game.

In filmmaking, the rule is that if you see a sound, you should hear a sound. We scrupulously observed this dictum in *Operations*. Sound effects, weapons foley, and ambiences were custom recorded or obtained from professional libraries. Weapons animations, for example, are accompanied by detailed and accurate audio representations that focus the players' attention on the weapons and heighten their emotional impact.

For added realism, footsteps, bullet impacts, particle effects, grenades, and shell casings are accorded texture-specific impact noises. A flying shell casing clinks differently on concrete, wood, or metal, for instance, and the distinction is clearly heard in the game. Likewise, footsteps on dirt, mud, wood, concrete, grass, and metal are sounded correctly.

In a typical *Operations* firefight, bullets whiz and crack by the player's ear, slam into the wall behind, and tinkle concrete and glass fragments at his feet. The player hears his own shell casings thunk off the wooden door behind him and ping the concrete floor. Meanwhile, to the clatter of a nearby reload, the enemy creaks across a steel catwalk overhead. The player hears a flash-bang grenade scud off the floor behind him just before being incapacitated by the roar and ring of tinnitus in his ears. We simulated the acoustical effects using OpenAL with EAX 3.0 extensions from Creative Labs.

America's Army: A Background

In 1997, the US National Research Council (NRC) issued a report specifying a joint research agenda for defense and entertainment modeling and simulation.[1] The report provided a guide to the R&D necessary to build such systems. Included was an agenda treating immersive technologies, networked virtual environments, computer-generated autonomy, standards for interoperability, and tools for creating simulated environments.

After the report's publication, the Moves Institute realigned its research directions with the agenda. The NRC report states that games and interactive entertainment—not defense research expenditures—have become the main technology drivers for networked virtual environments.

According to the report, to keep up with evolving modeling, virtual environment, and simulation technologies, the US Department of Defense (DoD) needed to examine networked entertainment to ascertain the potential for joint investment or collaboration. In the fall of 1999, the Moves Institute's Army Game Project (AGP) emerged as one such potential investment.

The AGP effort originated with a discussion between the directors of the Moves Institute and of the US Army's Office of Economic and Manpower Assessment (OEMA).

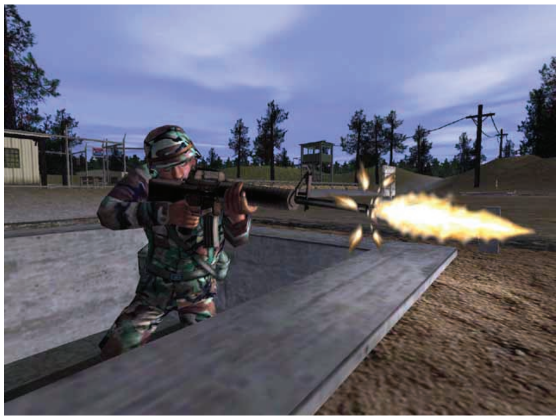
The Army was concerned about falling recruitment and perceived the need for new initiatives aimed at computer-literate recruits for today's high-tech Army. The discussion turned to the medium of the PC game.

The Army had previous success using popular entertainment media—they piggybacked advertisements onto newsreels in movie theaters in the 1930s and 1940s and employed trailers in theaters and Super Bowl TV advertisements in recent years. The emerging question, then, was Could the Army use PC games for strategic communication?

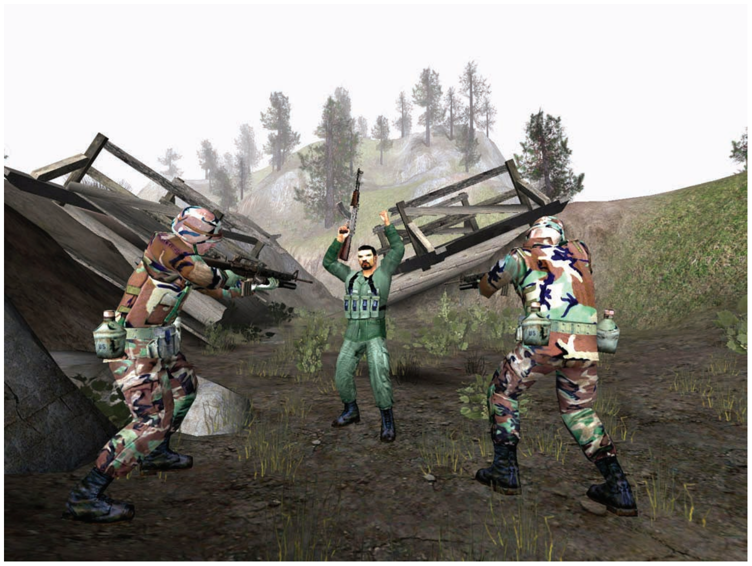
A dwindling number of young Americans have a veteran in the family with whom to discuss army life. The directors imagined PC games as a vehicle for communicating what an army career might entail. They also acknowledged that, since no one had tried such a thing, the only way to assess the idea was to actually plan and build a game.

Reference

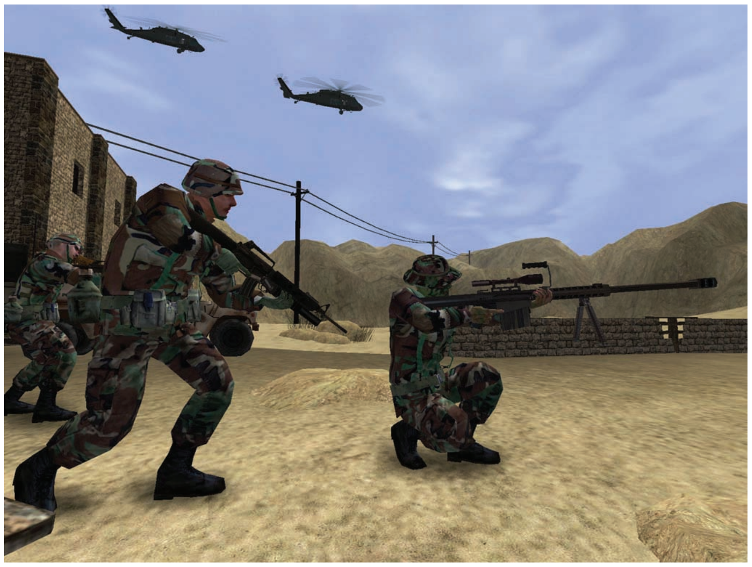
1. M. Zyda and J. Sheehan, eds., *Modeling and Simulation: Linking Entertainment & Defense,* National Academy Press, 1997.



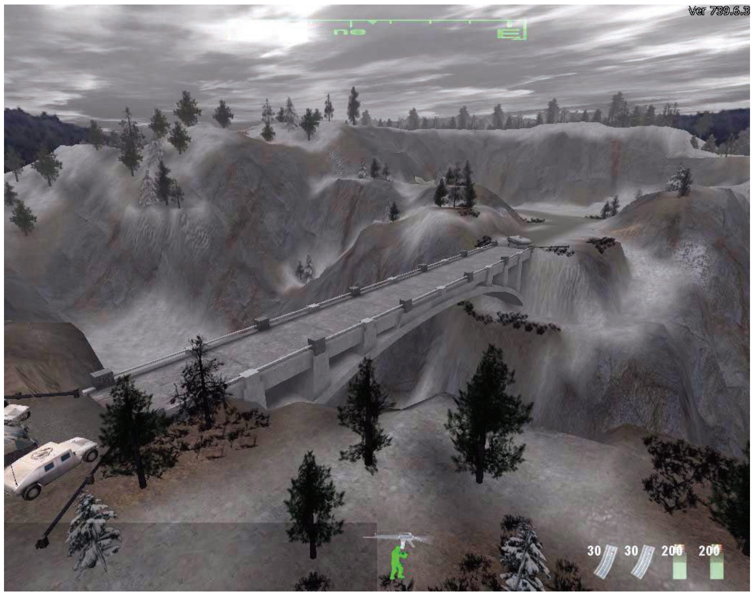
1 ***Operations*** **firing range. Weapons are modeled and handled authentically.**



2 ***Operations*** **players see themselves with US uniform and weaponry, and opponents as villains.**



3 ***Operations*** **desert mission.**



4 ***Operations*** **winter mission.**

Operations benefited from the advice of many talented people in the entertainment industry, including Gary Rydstrom at Skywalker Sound, who provided helpful insights into the design of the weapons' audio and proper use of ambiences.

The game's audio also benefited greatly from interactions with Dolby Laboratories and received Dolby Digital 5.1 Surround Certification. It's one of the first PC-based videogames to be released with this designation.

Operations team play

In *Operations*, no one ever plays a villain fighting the US. Both teams always see themselves as part of the US Army and perceive the other team as the opposition (Figure 2).

Operations offers a variety of combat scenarios, from desert to arctic, jungle to swamp, to traverse in mission fashion (Figures 3 and 4). The goal isn't to blast everyone in sight, but to cooperate as a team intent on a purpose, which might be to identify a weapons cache, rescue a prisoner of war, or perhaps assault an airfield. We designed scenarios so that mission goals and objectives make sense to both teams, requiring one group to assault and the other to defend.

All players abide by rules of warfare. If a player violates the Uniform Code of Military Justice, rules of engagement, or laws of land warfare, reprisal is instant. He will find himself in a cell at Fort Leavenworth, accompanied by a mournful harmonica playing the blues. Continued violation of the rules may cause a player to be eliminated from the game. To rejoin, he must create a new ID and restart.

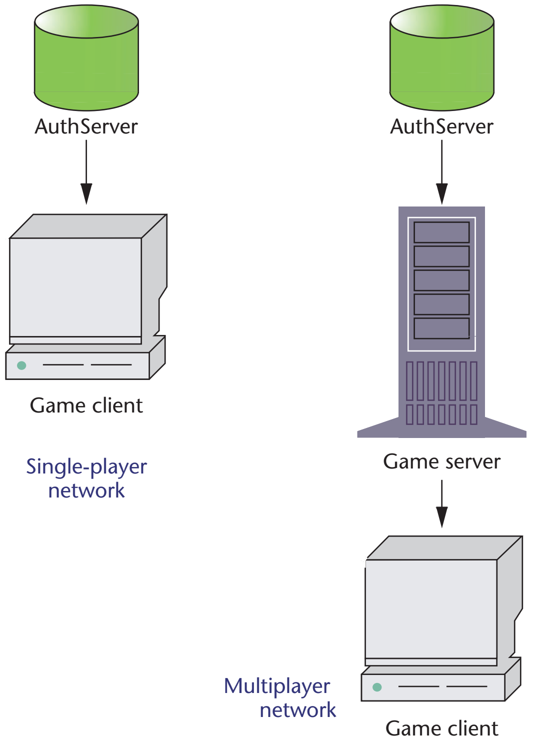
The game insists on the mission orientation of the US Army. Above all, soldiers must be team players, following army values and rules.

Network specs

Built on Epic Games' latest Unreal engine, *Operations* can support up to 32 players on a wide area network and perhaps up to 64 on a local area network. Nevertheless, as with many other first-person action games, the maps in *Operations* have a limit of 26 players to keep teams balanced and the bandwidth reasonable.

Besides the standard client–server component, *Operations* includes an online database to store player records. Through a Web page, the player creates an online account on the authorization server database. The game contacts the AuthServer at various times during play to update information online. For example, when a single-player training mission is completed, the client uploads the score to the database. When clients connect to an official game server, the server contacts the authorization server for verification of the client's mission qualifications and weapons training to make the appropriate weapons available. Online storage lets players download their characters to any machine running the game (Figure 5).

Plans are underway to link *Soldiers* and *Operations* through the database, so that players can unlock new abilities within *Operations* by completing tasks within *Soldiers*.

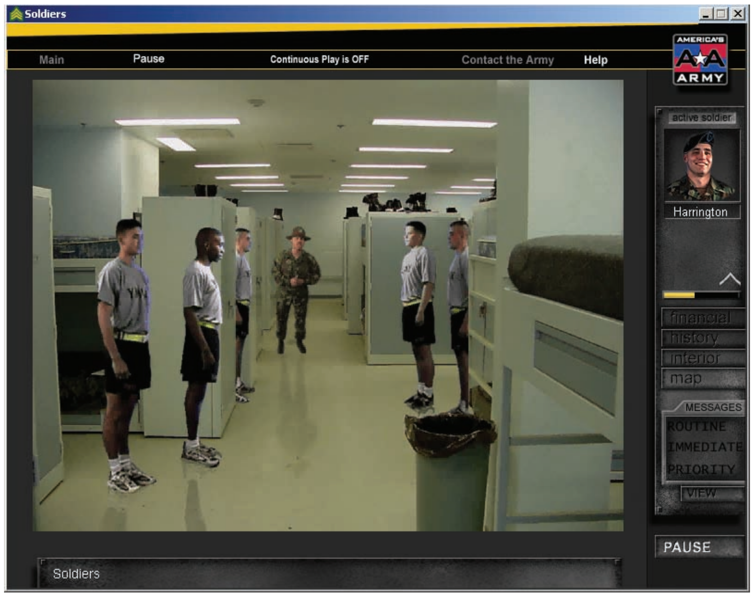


5 Networking configuration.

Computing needs

Operations runs on a PC under Microsoft Windows 98, ME, XP, or 2000, with a preferred minimum of a Pentium 3 processor running at 700 MHz, 128 Mbytes of RAM, 1.5 Gbytes of disk space, and a 56-Kbyte modem or digital subscriber line (DSL) connection.

Because of its complex scenes and characters, *Operations* requires a graphics card with hardware support for transformation and lighting, such as Nvidia's GeForce 2 chip set. No peripherals except a keyboard and mouse are required.



6 *Soldiers* screen shot: daybreak in the barracks.

Soldiers

While the combat training and missions on which *Operations* focuses are critical, they represent only a small part of army life and opportunities.

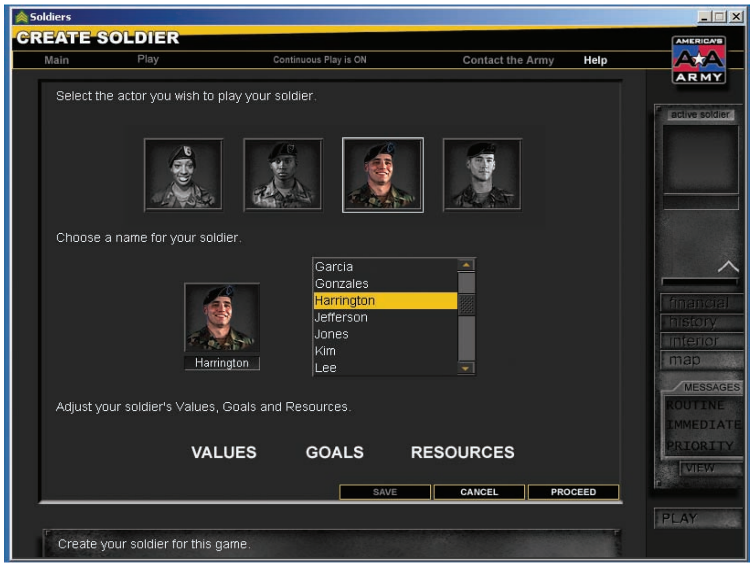
The *Soldiers* development team needed to present an unvarnished view of a wide range of army career fields. The rest of the army that *Soldiers* depicts includes basic training (drill instructors and all), advanced specialty training, on- and off-duty life, and enjoyment of the facilities available to army personnel and their families.

We did everything as accurately as possible, not only in portraying what army life is all about (Figure 6), but in the appearance of bases, offices, barracks, and facilities. To complement *Operations* and depict army locations with maximal realism, we employed digital imagery as the display medium.

Home-grown story engine

Unlike *Operations*, for which Moves was able to buy the latest commercial game engine, no existing engine satisfied the peculiar requirements of *Soldiers*. We had to build and design the entire system from scratch. *Soldiers* consists of four main components: a story engine, location generator, animation engine, and text-to-voice system.

Underlying *Soldiers* is a story engine that constructs and sequences scenes to create interactive storytelling. The player creates a character with a set of personality traits (based on the Army's seven core values) and



Select the actor you wish to play your soldier.

Choose a name for your soldier.

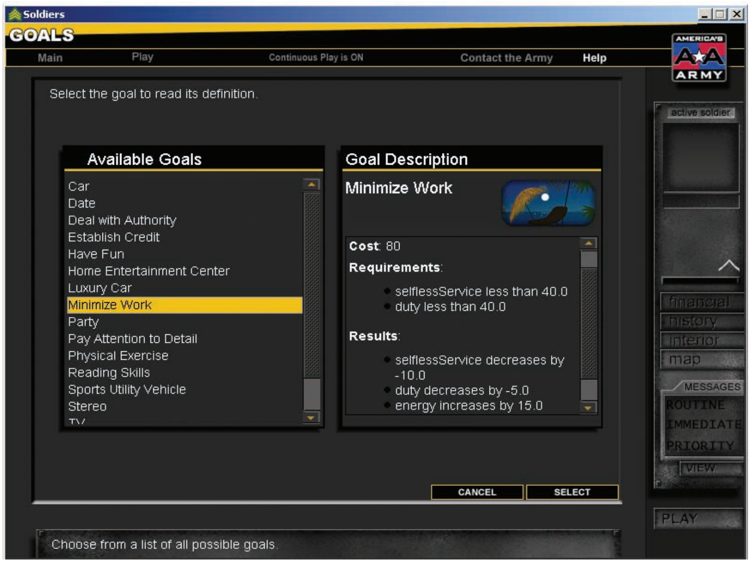
Harrington

Garcia
Gonzales
Harrington
Jefferson
Jones
Kim
Lee

Adjust your soldier's Values, Goals and Resources.

Create your soldier for this game.

7 ***Soliders*** **screen used to pick a character for play.**



Select the goal to read its definition.

Available Goals

Car
Date
Deal with Authority
Establish Credit
Have Fun
Home Entertainment Center
Luxury Car
Minimize Work
Party
Pay Attention to Detail
Physical Exercise
Reading Skills
Sports Utility Vehicle
Stereo
TV

Goal Description

Minimize Work

Cost: 80

Requirements:

- selflessService less than 40.0
- duty less than 40.0

Results:

- selflessService decreases by -10.0
- duty decreases by -5.0
- energy increases by 15.0

Choose from a list of all possible goals.

8 ***Soldiers*** **screen for tweaking character goals.**

guides it through a career, managing its resources and deciding its fate (Figure 7).

The player wins his character's trust through good decisions that give him increasing access to the character's interior, where he can tweak its goals and values (Figure 8). These manipulations propel the character through a career in a number of specialties. Along the way, the player gets a glimpse of what army life offers and demands.

Under the hood

For the past three years, the Moves Institute's computer-generated autonomy group has been exploring multiagent system simulation architectures that make the development of complex, adaptive behavior easier to achieve and control. Our research has produced four agent-based simulation design concepts for modeling multiagent systems and implementing the models in software simulations: composite agents, goal management, tickets, and connectors.[2]

These concepts evolved under the guidance of John Hiles and through many hours of discussion between principal investigators Michael VanPutte and Brian Osborn. A novel simulation methodology emerged, capable of generating dynamic plans and interactive stories. Called connector-based multiagent simulation (CMAS), the architecture serves as the underlying model for the story engine.

Soldiers' story engine is actually a general-purpose simulation engine that generates dynamic plans. Stories evolve as a byproduct of observing constraints previously defined on agents within the simulation.

In other words, the stories are plots generated through discovery rather than fixed plans defined beforehand. The engine produces them through a simulation process called *connecting*, whereby agents are bound together according to a best-fit axiom. The result of a successful connection is the next step in the plan (or, to put it another way, the next visualization of a story scene). By this means, dynamic plans—or *stories*—evolve as the simulation runs.

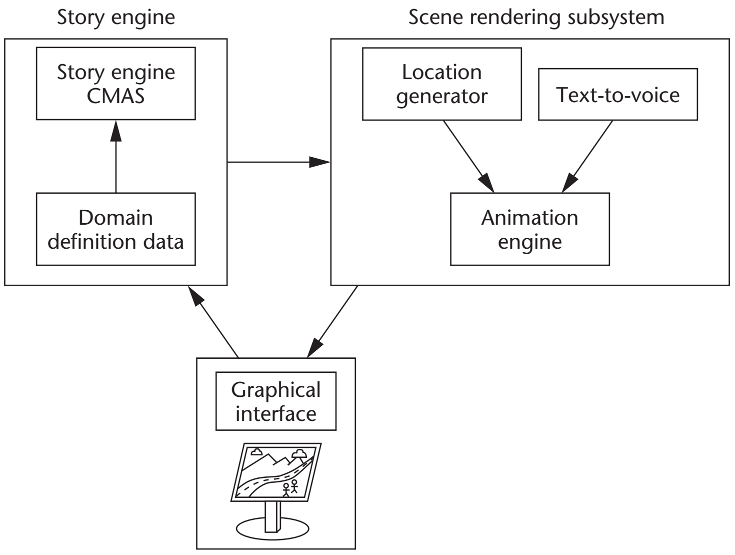
Figure 9 provides a diagram of how the system works. The scene-rendering subsystem of the CMAS takes streams of scenes produced by the story engine and sends them to a desktop computer after adding multimedia (graphics, video, and speech) to each scene as it generates (the scene-rendering subsystem is described elsewhere[3,4]). The story engine takes domain-definition data as its input (consisting of descriptions of the characters, goals, procedures, and objects in the simulation) and outputs a stream of scenes. The user can update the domain-definition data via the graphical interface.

In a CMAS simulation, agents communicate, and thereby advance the simulation through time, via connections. Agents extend connectors to signal their readiness to participate, and retract them when they aren't ready. Part of the story engine's job is to seek agents with extended connectors and match them.

Internal behavioral procedures (called *tickets*) and the current state of the environment dictate whether an agent's connectors are extended or retracted. Tickets are input to the story engine via the domain-definition data set. They specify the actions of individual agents, which in turn are dictated by the phenomenon being modeled. For example, if a predator were being modeled as an agent, then its hungry connector would be extended when the predator was in the hungry state.

The predator would seek a connection with agents that are edible—that is, prey. The story engine manages the task of matching predator to prey and updating the internal states of both. In this way, it remains independent of the particular story underway. Then the input data, not the simulator, drives the story.

Think of a CMAS simulation as a collection of agents that evolve their states, and thereby the state of the entire simulation, by making and breaking connections. The state of an agent is updated only when connected to one or more others. Thus the story engine is a simulation engine that manages the evolution of agents over time by matching them with one another, combining them in connections, updating their internal-state variables, and disconnecting. A stream of scenes is the byproduct of the repeated connect-update-disconnect process.



9 Embedding the story engine in an interactive story-generation system.

Soldiers' story engine uses tickets and connectors extensively to generate interactive, dynamic stories. Because the intention of *Soldiers* isn't to tell a predetermined story, but to cobble a credible plot on the fly, a typical story consists of goal-driven autonomous characters (such as a protagonist, drill instructor, buddy, and supporting characters), a narrative structure closely aligned with the main character, and a collection of potential scenes. Of course, we also use media, dialogue, and interactions to populate the scenes. Combined dynamically at runtime, these elements produce storylines that reflect the characters' actions, relationships, and personalities.

Two primary classes of agents exist in the story engine: characters and scenes. Through a process of connecting, the main character binds to a scene, and the scene in turn binds to resources such as supporting characters and media elements. Once resource requirements have been met, the scene executes based on the goals, values, and resources of the characters.

During scene execution, the story engine interfaces with the location generator, animation engine, and text-to-voice system to manufacture and present a scene onscreen. As the scene unrolls, the game updates each character's internal state. For instance, in a training scene a character's skill may increase and capabilities grow. When the scene is over, the process begins anew with the main character binding to another logically consequent scene.

With scenes dynamically constructed at runtime, the story engine generates a plot adapted to the user's interventions. As the story plays out, the player is free to adjust his character's goals and values. The result is a story personalized to the player's will.

Location generator

It goes without saying that we had significant obstacles to overcome to deliver a compelling product on CD. For both *Operations* and *Soldiers*, we needed to show real army posts through digital imagery, present a breadth of career fields, and reference the Army's depth of training and resources.

Many of these challenges centered around squeezing enough media on a CD to support the hundreds of thousands of scenes the story engine was capable of generating. For example, each scene is played on a 640×480 background or *location*. At approximately 960 Kbytes per background (uncompressed), it was impossible to store enough data on a 650-Kbyte CD to accommodate the story engine's range of scenes.

The answer was to use recombinable pieces. We made every bit of imagery, animation frame, and sound recyclable.

The location generator constructs backgrounds appropriate to a given scene and character by managing a database of chopped-up media pieces with which to construct locations at runtime. A typical location consists of 6 to 10 media pieces, including a sky, midground, foreground, and context-specific props. If we station the character at Fort Bragg, for example, a scene on the parade field will depict Fort Bragg buildings. The same scene played out at Fort Jackson would show the buildings there.

Animation engine

The reusability principle also applies to animation of character actions. We constructed an animation engine to build photo animations from individual frames of digital imagery. Army soldiers enacting a range of motions and gestures were first filmed against a blue screen. We processed the raw media and a portion used to construct a database of animation frames.

Just as a cartoon animator flips through a set of images to produce movement, the animation engine selects a set of prescribed frames from the database and plays them sequentially to create animation. Each character's database takes about 40 Mbytes of space.

Once we establish and catalog the frame set, however, the cost of adding animations built from the data-

E3 and the Fourth of July Release

Developing the *America's Army* suite silently for two years from first funding (May 2000 to product announcement at the 22 May 2002 E3 conference) was difficult for our large team of collaborators, but we managed to arrive at E3 with the story mostly under wraps. Our first major press exposure was on the front page of the *Los Angeles Times,* replete with a color shot from the game.[1] The suite won several press awards at E3 (http://www.movesinstitute.org/aapress.html).

The official *America's Army* Web site, http://www.AmericasArmy.com, contained little more than game images and a location where visitors could register for an email notification of the release date. By 24 May, the Web site was receiving 180,000 unique visitors per hour, with 18,000 pages served every five seconds.

Our booth at E3 had giant display screens high above the convention floor. Every two hours an army bugler called in an armed company of men simulating an air insertion, including soldiers scrambling down ropes hung from the ceiling of the Los Angeles Convention Center (Figure A).

Like hotcakes

We were pleased with the outcome of the conference, but when it came to posting *America's Army: Operations* onto the Internet, army program managers could only guess how many servers they'd need on 4 July, the official launch date. The Army budgeted for 140 servers to seed the game's launch.

On 4 July at 12:01 am, the first 10 levels of *America's Army: Operations* were posted to the Internet, and by noon the next day, 500,000 downloads of the 211-Mbyte game had been made. The Army's 140 servers were swamped, and we rushed to complete and post the community server kit the same week.

By 15 July, we were seeing approximately 1,900 servers, each capable of serving 26 players, for a total potential of around 49,400 players. By 30 August, *Operations* was downloaded 2.5 million times.

A Rappellers descend from the ceiling of the *America's Army* booth at the E3 Convention, May 2002.

Capture the record

Gratifying statistics roll in daily. Game use as of 16 November 2002 saw 1,007,000 registered accounts, 614,000 graduates of BCT, and more than 32 million missions completed (averaging 6 to 10 minutes). Missions per day average 338,380, with players typically accomplishing 21 missions after BCT.

Assuming 10 minutes per mission, we calculate gamers racked up a combined 263 years of nonstop play in the first 58 days alone (one avid player enacted over 3,600 missions during this period).

To put it another way, if these hours were payable at

base is minute. These 40 Mbytes of individual frames can produce hundreds of onscreen actions.

Creating a spoken dialogue system to match the story engine's staggering diversity presented a daunting task. Given the photorealistic backgrounds and animations, synthetic vocalization was out of the question. Instead, Moves constructed dialogue on a basis similar to the technique used in constructing locations.

In locations, a template guides the selection of images comprising the background. In our text-to-voice system, we used a formal grammar to define sentence structures, including phrases that vary not only according to the needs of the script but also to the speaker's identity. For example, a morning greeting by a sergeant and a buddy should differ.

We recorded actors speaking full sentences, then deconstructed the sentences into elemental fragments and collected them in a database. The text-to-voice system's sentence generator parses sentence definitions and constructs sentences from the database.

While our solution isn't as flexible as a text-to-voice program, we felt the gain in realism was worth the trade-off.

Instrumentation

With the help of the Army Research Institute, we looked into whether a game player's aptitude for an army career could be computed. The work from ARI minimum wage ($6.75 an hour), the bill would hit $15,590,367 for 58 days. And if we project the 4.6 years of play per day to 1,679 years of play per annuum, we're looking at $99,279,270 of intensive effort donated gratis by America's youth.

Though the Army has used http://www.goarmy.com as a recruitment site for years, traffic is way up—about 28 percent of the hits now originate from the game.

Hard copies

Mass production and distribution of the *America's Army: Operations* CD began 1 September. In midautumn, army recruiters received CDs directly from the factory, in initial quantities of 50 units apiece (for a total of 300,000 to all stations).

Approximately 100,000 CDs are slated for army events and 100,000 for fulfillment by mail. A million disks will be inserted in popular gaming magazines. In addition, *Operations* will be featured at this year's college immersion tour, available for a test drive via eight interactive kiosks in the tour's 53-foot trailer. To complement the exhibit's parachute simulators, *Operations* will be configured to highlight its single-player Airborne School missions.

Reference

1. A. Pham, "Army's New Message to Young Recruits: Uncle 'Sim' Wants You," *Los Angeles Times,* 22 May 2002, p. 1.

looks promising and may appear in a later version of the game. Meanwhile, the conclusion is that the Army won't receive aptitude data unless the player willingly forwards it to the Army.

If the player requests information about an army career, we'll ask if we can forward the player's scoring information to a recruiter. If the player approves, the game forwards the information. If not, it goes nowhere—cookies aren't used in this process.

To see how the release of the software turned out, read the sidebar "E3 and the Fourth of July Release."

Next up

Having a successful online game inside the Moves Institute is kind of like having your own particle accelerator. Lots of proposed applications and interesting research are coming in the door.

Many related training applications using the America's Army code base as a starting point are being considered. We have funding from one project that's using *Operations* for treaty verification preplanning, and an Air Force group is looking at funding a training level within the game that will deal with force protection. Infantry soldiers at Fort Benning are using *Operations* in their training regimen to gain familiarity with range procedures before setting foot on the real range. Also, the Army's Objective Force is looking at integrating prototypes of their new weapons systems into *Operations* to evaluate their potential utility.

One extraordinary possibility, raised by the Under Secretary of Defense's office, is massively multiplayer (MMP) gaming, and the America's Army project is being looked at both as a model of how such a development effort could be carried out within government and as a possible starting point for an MMP project. Work involved might include the procurement (or development) of a government-owned game engine capable of full-spectrum combat modeling and large-scale interoperability integration, as well as a programming interface for modeling human and organizational behaviors and stories à la *Soldiers*. An additional goal would be a rapid prototyping interface to the MMP that would allow any mission to be put together nearly overnight. ■

Acknowledgments

This work would have been impossible without the support of our sponsor, the Office of the Deputy Assistant Secretary of the Army for Manpower and Reserve Affairs.

We salute the excellent work of program manager Lieutenant Colonel Casey Wardynski, as well as Major Chris Chambers, of the Office of Economic and Manpower Assessment. The full development team of America's Army appears on the Web at http://movesinstitute.org/team.html.

References

1. S. Singhal and M. Zyda, *Networked Virtual Environments—Design and Implementation,* ACM Press, 1999.
2. J. Hiles et al., *Innovations in Computer Generated Autonomy,* Naval Postgraduate School technical report NPS-MV-02-002, Monterey, Calif., 2002.
3. N. Elzenga, "The Recruits—Media Architecture System Description," *Army Game Project* (working document), *Moves Institute*, 2001.
4. B.A. Osborn, *An Agent-Based Architecture for Generating Interactive Stories*, PhD thesis, Dept. of Computer Science, Naval Postgraduate School, 2002.

Michael Zyda *is the director of the Moves Institute at the Naval Postgraduate School (NPS), Monterey, California. His research interests include computer graphics, large-scale, networked 3D virtual environments, agent-based simulation, modeling human and organizational behavior, interactive computer-generated stories, computer-generated characters, video production, entertainment/defense collaboration, and modeling and simulation. He's the prin-*

cipal investigator of the America's Army *PC game. He received his BA in bioengineering from the University of California, San Diego, La Jolla, his MS in computer science from the University of Massachusetts, Amherst, and his DSc in computer science from Washington University, St. Louis.*

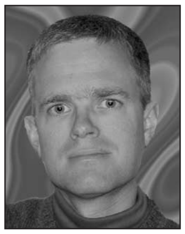
John Hiles *is the technical director for computer-generated autonomy at the Moves Institute, a research professor at NPS, and program manager for the* America's Army *suite. His main work is in using agent-based simulation and entertainment technology to model human and organizational behavior. He has a BA in creative writing from the University of California, Santa Barbara.*

Alex Mayberry *is creative director for the Moves Institute and the executive producer of* America's Army: Operations*.*

Casey Wardynski *is the originator and director of the* America's Army *game suite, and director of the Army Office of Economic and Manpower Analysis at the US Military Academy, where he is an associate professor of economics. He has an MS in public policy from Harvard and a PhD in policy analysis from Rand Graduate School.*

Michael Capps *is a producer with Epic Games in Raleigh, North Carolina. He is a former research assistant professor at NPS, as well as former producer of the* America's Army: Operations *game. He received a BS degree in mathematical sciences and an MS in computer science from the University of North Carolina, Chapel Hill, an MS in electrical engineering and computer science from MIT, and a PhD in computer science from the Naval Postgraduate School.*

Brian Osborn *is an active duty naval aviator and researcher in the Moves Institute. He's the principle implementor of the Moves Story Engine. His concentrations include agent-based modeling and simulation, computer-generated autonomous behavior, and interactive narrative. He has an MS in operations research and a PhD in computer science from NPS.*



Russell Shilling *is the technical director for immersive technologies at the Moves Institute, an associate professor at NPS, and the lead audio scientist and designer for the* America's Army: Operations *game. He received his PhD in experimental psychology (neuroscience) from the University of North Carolina, Greensboro.*

Martin Robaszewski *is a programmer on the* America's Army: Operations *video game. His main areas on the project are user interface, networking, and mission-scripting. He's primarily self-taught and has worked on numerous platforms using many different programming languages.*

Margaret Davis *is the writer and Webmaster for the Moves Institute. She has an MA in British literature from California Polytechnic State University, San Luis Obispo.*

Readers may contact Michael Zyda, the director of the Moves Institute, at zyda@movesinstitute.org.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.