Multiplayer Computer Games: A Team Performance Assessment Research and Development Tool

Elizabeth M. Biddle, Ph.D.
Michael L. Keller
The Boeing Company
13501 Ingenuity Drive
Suite 204
Orlando, FL 32826
407-736-8732, 407-306-9468
elizabeth.m.biddle@boeing.com, michael.l.keller@boeing.com

Keywords: performance assessment, games, team performance

ABSTRACT: Commercial computer games are being widely investigated by the military modeling, simulation, and training communities. The goal is to provide realistic operational environments for training, decision-making, and human behavior representation development. This approach has been enabled by the exploitation of the powerful graphics (almost photorealistic), high-fidelity physics and behaviors that are embedded in several of these low-cost games. Additionally, many of the more sophisticated computer games come bundled with editing and scripting tools offering the user the ability to access/modify game features and additional internal workings. This paper describes an effort to utilize Unreal Tournament 2004, a commercial computer game, as a vehicle to develop a testbed supporting the research and development of automated performance assessment technologies in collective training environments. The rationale, method, and evaluation of this proof-of-concept testbed are described. The paper concludes with a description of future activities planned in this area.

1. Introduction

Commercial computer games have become a popular topic in the modeling and simulation (M&S), training, and education communities. Their affordability, the use of high-quality graphics and physical functionality provide operationally realistic environments for military operations that can be conducted in both single player and multiplayer modes (Herz & Macedonia, 2002). Additionally, many commercial computer games include powerful scripting languages that enable the individual user to modify the characters, scenarios, and synthetic environment (the result is termed a “mod”).

The ease of customizing these games has led to the emergence of a “mod” community. The mod community uses the Internet to share information on how to create mods as well as the mods themselves (Chick, 2002). In many cases, the original game developers have leveraged the products of the mod community in the production of newer versions of their games (Lenoir, 2003).

Given their low-cost, commercial availability, and ease of transforming the simulation characteristics, training and human behavior researchers have been using the games as testbed environments for investigating the development of engaging training methods (Herz & Macedonia, 2002; Zyda et al., 2003) and advanced human behavior representation technologies (Back, 2002; Wray, Laird, Nuxell, & Jones, 2002).

The authors of this paper have been studying the development of automated performance measures for implementation in team and collective training devices. To conduct this research, a testbed for collecting data on the performance of team and collective tasks was needed. Attack helicopter operations were of particular interest to the authors. A proof-of-concept collective training testbed was implemented using the computer game Unreal Tournament 2004, developed by Epic Games, as the basis for the virtual team-training environment. This paper describes the development of a prototypical collective training testbed and how it can be used to develop performance measures for collective training missions.

2. Background

The need for an automated performance assessment capability to improve the evaluation of collective training exercises has been advocated (Cardinal et al., 2004; Meliza et al., 1992; Watz et al., 2003). To address this need, the design and development of an automated performance assessment capability was pursued. A simulation environment for implementing team/collective training scenarios and recording performance data automatically was needed to support this endeavor.
Based on the results from prior research efforts that employed computer games to: (1) provide realistic military operational environments (Laird, 2001; Manojlovich, Prasithsangaree, Hughes, Chen, & Lewis, 2003) and (2) study team processes (Proctor, Panko, & Donovan, 2003; Stahl & Loughran, 2002), the development of a computer game-based team/collective training testbed was initiated. The goal was to evaluate the feasibility of using a low-cost, computer game to develop a collective/team training testbed. Specifically, a proof-of-concept collective/team training testbed was launched to verify the feasibility of creating scenarios from which performance data could be automatically logged. If successful, further enhancements would be made to the testbed to meet the requirements needed to support future automated performance assessment research activities.

2.1. Collective Performance Assessment Investigation

The requirements for an automated performance assessment capability were identified through: (1) literature and technology reviews and (2) interviews with military instructors involved with collective training. Collective training was defined as team-of-team training, which involves the performance of task-related and team-related processes. The assessment of individual task performance is generally understood. However, teamwork performance is more complicated and has primarily addressed teams at the crew level (tactical teams – see Salas, Stagl, & Burke, 2004 for a comprehensive overview of this research) and more recently at the command and control level (decision-making teams – see Warner & Wroblewski, 2004). However, the collective level has been less studied, and currently there are no standard collective assessment methodologies or technologies in existence.

Performance assessment refers to the analytical process of making inferences regarding a trainee’s (individual or unit) mastery of a training objective (Department of the Army, Training and Doctrine Command, 1999). It is anticipated that performance assessment results can be used to diagnose performance and provide feedback during debriefs or After Action Review (AAR) sessions. Collective performance assessment is typically performed by a human observer/trainer who assesses unit proficiency as T (trained), P (needs practice), or U (untrained) (Headquarters, Department of the Army, 2002). This method of assessment is instructor intensive and challenging. The task becomes even more daunting for collective missions, which involve many individuals and teams simultaneously performing actions in a dynamic environment.

A consistent and reoccurring result is that accuracy and reliability of such assessments are poor. This is due in most cases to human error during the monitoring and recording of performance, and the subsequent application of subjectivity and bias during this type of evaluation (Cardinal et al., 2004; Holden, Throne, & Sterling, 2001; Watz, Keck, & Schreiber, 2004). Additionally, the use of TPU assessment does not provide for detailed diagnosis of the underlying causes of good or poor performance. Methods for automatically recording data from collective training environments, analyzing the data, and presenting it to the instructor in an understandable format to improve the evaluation of collective training events were investigated. The development of these technologies required access to a simulated training environment that can be used to:

- Implement operationally plausible collective or team military training scenarios
- Record individual (e.g., button presses) and collective processes (e.g., maintaining formation, providing situational updates)
- Create training scenarios
- Specify the data to be collected
- Support various domain tasks (e.g., Rotary or fixed wing operations)

To meet these objectives, the use of readily available commercial computer games to provide a simulated, collective training environment was investigated.

2.2. Computer Games

State-of-the-art commercial computer games provide the user with a sense of realism and an engaging story (scenario). Synthetic characters can serve as interactive, computerized allies or enemies. The Department of Defense (DoD) has supported the development of several game-based training applications such as the Marine Corps’ family of Tactical Decision Making Simulations and the Army’s Full Spectrum Warrior, developed by the Institute for Creative Technologies.

Massive, multi-player (MMP) games provide an infrastructure to support distributed large- or small-scale collective training exercises (Defense Science Board, 2001). The United States (US) Army Research and Development Command (RDECOM) is exploiting the benefits of MMP games (Miller, unknown) to provide a realistic, distributed, training environment, known as the Massively Multiplayer Simulation for Asymmetric Warfare, used to train asymmetric military missions.

In addition to employing gaming technologies to provide engaging training applications, the military M&S and training communities have been leveraging commercial games as simulation tools. There are two areas that commercial games have primarily been used to support:
(1) human behavior representation research and development (R&D) and (2) 3-D visualization.

Dr. John Laird pioneered the leveraging of commercial games to serve as testbeds for developing advanced human behavior representation techniques. Dr. Laird’s approach is to implement the games such that the human behavior technology, Soar in this case, controls the behavior of one or more of the game’s synthetic characters. This initial work (Laird, 2000) employed the computer game *Quake*, developed by Id, a first-person shooter game. *Quake* has a scripting language that enables the user to modify the visual and behavioral characteristics of the game’s characters, environment, and objects.

More recently, versions of Epic’s *Unreal Tournament* (UT) commercial game (Laird, Wray, Nuxell, & Jones, 2002) were used to develop realistic behaviors for opposing forces for Military Operations in Urban Terrain (MOUT). *UT* is also a multi-player, first-person shooter game set in a futuristic environment. Similar to *Quake*, *UT*’s scripting and editing tools enable the user to customize the game. These activities include: importing or creating terrain, modifying the visual and behavior properties of the synthetic characters, and enabling the user’s ability to access the game physics and user interface actions. Additionally, there is a large user community that is continuously making tools and player entities available to the public at no charge. Best, Lebiere, and Scarpinatto (2002) have also utilized *UT* to emulate a MOUT training environment to support the development of credible synthetic opponents for MOUT training simulations.

*UT* and Epic’s *Unreal* game engine (set of common code used to produce new games) have been used to provide a 3-D viewer for 2-D military simulation applications. Manojlvich *et al.* (2003) used *UT* as a 3-D viewer for the constructive simulation: OneSAF (One Semi-Automated Force). They developed an architecture, Unreal Tournament Semi-Automated Force (UTSAF), that enabled the entities represented in OneSAF to be viewed in a modified version of *UT*. Again, *UT*’s editing tools enabled the terrain and synthetic entities to appear as a realistic military environment and platforms. Another effort (Doris, Larkin, Zieniewicz, & Szymanski, 2004) employed the *Unreal* engine in a similar manner to support the development of an application interface (API) prototype to enable military simulations to use computer games as 3-D viewers.

3. Team Training Testbed Development

*UT* 2004 was used to evaluate the feasibility of commercial computer games to provide a collective/team training environment to support the development of automated performance assessment technologies. *UT2004* was selected because the editing tools that are bundled with the game provide a means of recording scenario events and user interactions (*e.g.*, keyboard strokes). Additionally, *UT2004* offered a new feature enabling one to six players to simultaneously board and operate ground and air vehicles. This feature enables the conduct of a collective (team of team) scenario. For example, a collective helicopter scenario can be implemented by having two or more people work as a helicopter crew in a scenario that involves two or more helicopters. Further, *UT2004* has a voice chat feature that allows players to talk to each other over a local network or the Internet. As communications between crewmembers and teams is an important process in team and collective operations, the voice chat feature adds another level of realism to a team or collective training scenario.

First, an attack helicopter scenario was specified. This scenario involved two helicopters whose mission was to destroy a set of targets that were being protected by ground forces. Figure 3.0-1 provides a description of the scenario. A map of the scenario is displayed in Figure 3.0-2.

```
• Friendly forces are 2 helicopters (or comparable aircraft)
  – 1 lead
  – 1 wingmen
• The mission objective is to destroy an enemy HQ camp, which is comprised of 5 towers.
• There will be a few land-based threats that the helicopters will encounter on their mission.
  – Ground forces using rifles and surface-to-air missiles
• The helicopters will have planned waypoints that they are supposed to arrive at on the way to the initial point (the point from which they engage the enemy).
• After destroying the target, the helicopters will return to their Point of Departure.
```

Figure 3.0-1. Scenario Description

Next, performance measures were developed to evaluate the scenario. These measures were: time-on-target, accuracy of weapon fire, and number of targets destroyed. Additionally, the ability to evaluate ownship status (fuel consumption, damage) and communications were identified as behaviors of interest, but no specific metrics were defined. The automated data collection requirements to enable the evaluation of these metrics were:

- Ability to time-stamp data
- Position of ownship (altitude and coordinates) for both helicopters
• Coordinates of: waypoints, refueling station, Point of Departure, Holding Area, all threats
• Speed of ownship
• Player keyboard input (when they fire a weapon, steering input for the aircrafts, communications sent)
• Knowledge of weapon success/failure (e.g., indication that a missile was fired, indication of whether the missile killed the threat or not —or if it killed a friendly)
• Status of ownship (both helicopters)—e.g., fuel consumption, damage

The UT2004 testbed (Figure 3.0-3) was launched employing three Dell Precision 670 desktop computer stations. Each station had a keyboard and mouse to interact with the environment and an Altec Lansing headset with microphone to communicate and hear the game’s audio environment. Two of the stations were used as player (trainee) stations, and the third station was used to simulate an instructor operating station (IOS).

An UT2004 mod (FragOps) was used to implement the attack helicopter scenario. UT2004 users employed the UT Editor to create a modified version of this game and have made it available for free on the Internet (http://www.frag-ops.com/). FragOps is set in the very near future (2006-2007) and simulates real-world soldiers and vehicle platforms (Comanche helicopter, jeep, MIG style aircraft). The programmers decided to use FragOps to implement the scenario since it had a realistic helicopter platform (Comanche) and combat environment as there was a limited amount of time allocated for this effort. The selection of FragOps reduced the development effort by eliminating the need to employ the UT Editor to develop realistic helicopter platforms, ground forces, and terrain.
The scenario implemented with *FragOps* involved two players (can accommodate more than two players), who each assumed the role of a Comanche pilot. Their mission was to board and fly their own Comanche and destroy five lookout towers that were being defended by enemy ground forces. The two player-stations acted as the helicopter cockpits. The IOS station was used to monitor and run the exercise. The IOS station was enabled using the *UT2004* spectator role feature that allows the user to fly to various viewpoints in the *UT* game environment.

Figure 3.0-3. *UT2004* Testbed

Although the use of *FragOps* eliminated the need for the physical development of platforms and terrain, there is little supporting documentation regarding the development of the game since it was created for purely entertainment purposes. Therefore, there was some difficulty encountered with modifying aspects of the scenario, such as target types. Due to the critical time constraints for the development of the testbed, the targets and waypoints depicted in Figure 3.0-2 were modified. Additionally, the data collection requirements were reduced.

The automated data logging capability was achieved by modifying the output of the network log file that *FragOps* automatically generates. In order to amend the log file to record a specific variable, the naming convention that *FragOps* utilized for that variable was needed. However, the lack of available documentation for the game and time limitations made it impossible to meet the previously stated data collection requirements. As the goal of this project was to verify that modification of the scenario and collection of data was feasible, a reduction in the data requirements would not prohibit the feasibility evaluation. The script for the *Frag Ops* log file was modified to include scenario time and update the player positions at one-second intervals. Additionally, the log file was also updated to modify the format of the position data so that it was usable for data analysis. Therefore, the data collection capabilities for this proof-of-concept implementation became:

- Ability to time-stamp data
- Position of ownship (x, y, and z coordinates) for both helicopters

Additionally, the coordinates for the Point of Departure and threats were obtained from the *FrapOps* map file as they were fixed variables.

4. Proof-of-Concept Evaluation

As stated previously, the goal of this effort was to determine whether the *UT2004* testbed would provide the capabilities needed to develop automated performance assessment methods and technologies for a collective training application. These capabilities, which were identified in Section 2.1, are:

- Implement plausible collective or team military training scenarios
- Record individual (e.g., button presses) and collective processes (e.g., maintaining formation, providing situation updates)
- Modify the training scenario
- Specify the data to be collected
- Support various domain tasks (e.g., helicopter or fixed wing operations)

Figure 4.0-1 Track History from Team Scenario

The evaluation involved the performance of the attack helicopter scenario described in Section 3 and analysis of data recorded from the scenario. Two trainees from the Orlando Boeing office participated in the scenario. As the goal was to verify that the game-based testbed could automatically log data that could be used to implement automated measures of performance, only one measure was used: time-to-complete-scenario -- the length of time it took the team to destroy all five targets. Additionally, an individual training scenario was performed in which
one player assumed the role of helicopter pilot and destroyed all the targets single-handedly. Time-to-complete-scenario was evaluated for both the team and individual versions of the scenario, and they were: 4.9 and 5.4 minutes, respectively. As there was no intent to draw conclusions from the data, such as learning, the data were not analyzed further.

Future use of UT2004 as the simulation for the testbed involves the creation of a mod from the original UT2004 games. This will enable the specification of terrain features (including the threats and targets), data to be logged automatically, and the platform type for the scenario participants.

6. Conclusion

Commercial computer games offer high-fidelity visualizations, physical fidelity, multi-player simulation environments at a reasonable cost. Many of these games include editing and scripting tools to provide the user with a means of modifying the game features and inputting and outputting data to and from the game. Consequently, commercial computer games are being used by researchers in the M&S and training communities to develop advanced applications.

A testbed environment that enables the performance of team/collective scenarios and the automated logging of data was needed to support automated performance assessment research. Given the capabilities of the UT2004 computer game, it was selected to implement a proof-of-concept game-based, team training testbed. The evaluation of the UT2004 testbed demonstrated that it provides the capabilities needed for this effort. However, there were limitations to the testbed prototype that will need to be addressed prior to its use as part of the research and development activities.

7. Reference


In, Proceedings to the 2004 Spring Simulation Interoperability Workshop.


8. Acknowledgements

The authors would like to acknowledge the efforts of: James Durtchy, Edward Kimble, Mark Kohm, Matthew
Skaggs, and H. Mark Taylor in the development and setup of the UT2004 testbed.

Author Biographies

**DR. ELIZABETH BIDDLE** is an Engineer/Scientist at The Boeing Company located in Orlando, Florida. She is a Principal Investigator for advanced instructional and training research and development projects. Dr. Biddle earned a Ph.D. in Industrial Engineering and Management Systems from the University of Central Florida in 2001. She received the Modeling & Simulation Professional Certification by the Modeling & Simulation Professional Certification Commission in 2002 and the Modeling & Simulation Award in Training by the Defense Modeling & Simulation Office in 2001.

**MICHAEL L. KELLER** is Senior Systems Analysts at The Boeing Company in Orlando, Florida. Mr. Keller is currently the Training Support Contract II Lot I Lead Engineer and is supporting the automated performance assessment research and development activities. Mr. Keller has been with The Boeing Company for fifteen years and has been the Instructional Design Lead for many DoD and FMS efforts. Mr. Keller received an B.A. in Industrial Psychology from Michigan State University in 1966 and has graduate work at Ball State University 1970-71 and the University of Northern Colorado in 1974-75 in Psychology. Mr. Keller retired from the United States Air Force where he dual rated as a Pilot/Navigator.