Game-Based Disaster Emergency Response Training
A Proposal to the CITRIS White Paper Competition
Kenneth Hullett, Ph.D. Student, Computer Science, UC Santa Cruz

Abstract
Nationwide, one rescue worker is lost for every two victims rescued in collapsed structure situations; following Sept. 11, 2001, this ratio fell to 1:1. Limited physical facilities are available for training rescue workers in collapsed structure rescue. Our idea is to develop a training game for teaching collapsed structure rescue skills to members of emergency response community. In order to provide the most effective training possible, we plan to develop a scenario generator using AI planning techniques. The trainee will specify which skills they want to train, and the game will create a unique scenario that containing the appropriate challenges.

Background
The idea for this project was developed after meeting with members of the Disaster Assistance and Rescue Team (DART) at NASA Ames Research center. They offer one of the few training in the world that specializes in collapsed structure rescue, and host an annual class with highly limited enrollment.

This limited access to training facilities is compounded by the fact that emergency response teams are largely composed of volunteers who respond only when their team is activated. They may be geographically dispersed and have day jobs in other fields. A training game would allow rescue workers to train on their own PC in their leisure time.

For a training game to be most effective, it would have to provide a large number of unique scenarios, insuring that the trainee is always provided with an unknown challenge. Creating a sufficiently large number of scenarios individually would be prohibitively time-consuming, so we plan to integrate a scenario generator into the game.

This project overlaps two research areas, both of which are relatively unexplored but do have a base of existing research. There have been other projects in the Serious Games area that relate to emergency rescue training. The closest work to this project is being done by Nathan Schurr and Milind Tambe at USC. Their DEFACTO system is a training tool for incident commanders for large scale disasters[^1]. While the domain is similar, their work differs from our proposal in several ways. The DEFACTO system also uses static scenarios instead of procedurally generated scenarios. The trainee can only train on the two provided scenarios. While separate training sessions may play out differently due to variability in the agent’s choices, it is still the same scenario. Our project will provide a unique scenario for every training session. Like real world rescue operations, the trainee will only have limited information about the situation they are entering. Further, DEFACTO focuses only on the high-level management of rescue operations, rather than the details of rescue operations.

Hazmat: Hotzone, developed at the Entertainment Technology Center at Carnegie-Mellon University, allows firefighters to train on hazardous materials scenarios in a multiplayer, 3D game environment[^2]. Like DEFACTO, Hazmat: Hotzone relies on static scenarios.
This requires an expert human trainer to design each scenario, placing all elements by hand. The result is that the trainees are only presented with only a small set of scenarios which provide little benefit after the first playing.

There is a small body of existing work in the area of procedural scenario generation, very little of which overlaps with the work we are proposing. Some commercial games have employed a limited form of procedural scenario (level) generation for 2D levels. One of the best known examples, Diablo 2, follows the same random level generation approaches originally developed in Rogue in the late 1970s. While this approach results in unique levels, they are not compelling and lack internal consistency. The additional requirements for spatial coherency imposed by 3D environments have prompted game developers to use pre-designed static levels. Current approaches to level generation for 3D environments in academic research projects tend to employ constrained-random placement of individual features.

The proposed Collapsed Structure Training Game would be unique in both fields. No existing Serious Game employs a procedural scenario generator to provide trainees with an unlimited number of training scenarios. Since rescue workers rarely know what to expect in real-world rescue situations, our proposed system would be a more effective training tool than any existing system. In the procedural scenario generation field, we are breaking new ground by applying the AI techniques of planning and knowledge representation to create plausible scenarios that provide the trainee with challenges specific to the desired training goals.

Collaboration with DART
On this project, we expect to work closely with Bob Dolci, chief of NASA Ames’ Disaster Assistance and Rescue Team (DART) and other members of DART. They are experts in collapsed structure rescue operations and will provide invaluable input to the content of the training game. They have also allowed us to observe and participate in their annual Collapsed Structure Rescue Class (CSRC). This is a 6 day, 60+ hour class held at DART’s training facility at Moffett Field. Participants learn proper techniques for rescuing victims trapped in collapsed structures. Allowing us to observe the CSRC provided us with considerable domain knowledge and a feel for the experience we plan to replicate in the training game.

Game Design
In real rescue operations, workers operate as part of a team, so we plan to include networked play into the game design. Multiple trainees will be able to coordinate efforts in the same manner they would in a real operation. Rescue team members in different locations will be able to participate in group training, provided they have a sufficiently fast internet connection. The game design has 2 different game play modes: the rescue worker’s view and the operation commander’s view.

The 3D, first-person aspect of the training game will provide a rescue worker’s view of an operation. This game play mode will superficially resemble a first-person shooter, a popular genre of commercial games. The player will have to plan a course of action based on their visual evaluation of the structure, and then use various tools to breach walls, shore up weaken areas, locate and extract victims.
The other mode of game play will be the operation commander’s view. Commanders are responsible for planning operations, assigning tasks, and allocating resources. The operation commander screens give the trainee an overview of the operation, allow assignment of personnel and resources, and display reports from workers in the field as they progress through the structure.

We plan to use the Torque Game Engine, a commercial game engine. The engine’s maker, Garage Games, has agreed to provide us with a free license for this project. Like all modern game engine, this is a large software system composed of many interconnected modules. While it provides much of the basic game functionality, many features needed for the training game must be implemented by us. This requires learning how the existing code base works and planning the best way to add the new features. We hope to engage undergrad computer science students in this project by allowing them to do some of the necessary work on the game engine.

By using the Torque Game Engine’s multiplayer functionality, the training game will allow trainees to simulate a complete operation. Commanders will assess the situation and assign other networked trainees acting in the role of rescue workers to specific tasks. These workers would then perform the tasks in the 3D first-person environment. As the situation evolves, the workers would report back to the commanders, requiring them to make further strategic decisions about the progress of the operation.

This integrated, multi-mode style of game play is a much more realistic presentation of the experience of a rescue operation. This would allow rescue teams to train as a cohesive unit, with each member carrying out the same tasks they would in a real rescue operation. Commanders would not only get experience managing personnel and equipment assignments, they would also be able to evaluate the performance of rescue workers in different situations, allowing them to better assess the strengths and weaknesses of their team members. The system could be used to test team members in a variety of roles to determine where they could contribute the most to the operation. Experienced rescue workers being considered for command positions could be evaluated in the commander role while less experienced workers can be exposed to a variety of different types of operations in a realistic environment.

A significant component of any modern computer game is its art assets. Game studios that develop top commercial game titles employ dozens of artist and animators over a 2+ year development cycle. For the training game to be compelling to its target audience, we must provide equivalent high quality art assets. These include highly detailed 3D models of the objects the trainee interacts with and the environment they are working in, as well as 2D art for planning and overview screens. Creation of these art assets is a specialized skill that requires talent and training. We have already contacted some UCSC art students who are interested in contributing their efforts to the project. We may also purchase some existing libraries of art assets.

**Scenario Generator**

In addition to the training game itself, the plan for this project includes a scenario generator. This by itself is a first-class research problem. To date, no game or simulation has included procedurally generated scenarios. Our design incorporates AI planning techniques to generate scenarios that satisfy given training goals. Furthermore, this
design is general enough that it can be modified to generate scenarios for training games in other emergency rescue domains.

Our planned approach begins with representing elements of a scenario in a graph. Each element and its associated properties are a node in the graph. Elements include parts of the structure itself, as well as potential hazards and victims. The scenario generator is given this graph and a set of training goals. The training goals define what conditions need to exist in the scenario. The scenario generator can apply operations to the graph that alter the properties of elements, causing the desired conditions to exist in the scenario. These operations and the relationships between nodes in the graph codify naïve physics knowledge, i.e., a column supports a ceiling, so an operation that causes a column to collapse would most likely cause the ceiling to collapse as well. The links in the graph propagate causal connections between building elements. This ensures that a generated collapsed structure level maintains realistic constraints between the parts of the building.

For example, if dealing with fire during collapsed structure rescue is a training goal, the scenario needs to contain a fire. For the scenario to contain fire, the conditions for a fire to start need to exist in the scenario. The scenario generator searches the graph of elements whose properties may be altered to create those conditions. If the scenario contains a source of natural gas, a leak could cause a fire, so the scenario generator will search for a way to change the properties of that source so that it is leaking. Changing this property, in turn, can only occur if properties of other nodes in the graph are changed, such as the wall containing the gas line being damaged. The scenario generator keeps searching for ways to apply operations to elements in the graph, altering their properties, to create a scenario that contains the features needed to test the goals.

**Extendibility**

While our initial focus is on collapsed structure rescue, we also hope to generalize the procedural scenario generator and the game infrastructure to support training for a wide class of emergency response operations. Other examples of emergency response domains include hazardous chemical spills, fires, floods, and terror attack response. While these domains differ from collapsed structure rescue in many ways, the method for generating scenarios can utilize the same operator-based search mechanism with graph-based constraint propagation. Certain conditions must exist in the scenario for training goals to be tested. Those conditions imply the existence of other conditions. Beginning with a representation of a structure and the stated goals, the scenario generator can search the space of possible interactions to create a unique scenario that tests the training goals.

Currently, training games for emergency response rescue are built as one-offs. There is no common infrastructure for creating such games for different emergency response domains; each such game must be slowly and painstakingly created from scratch. We want to change this situation, supporting the more rapid creation of emergency response games, and unlocking the true training potential of such games by making dynamic scenario generation, currently an exotic technology at the cutting edge of game research, a common technique. By creating this generalized framework for all types of emergency response training games, we hope to address a major societal challenge by increasing communities’ emergency preparation.
Bio
Kenneth Hullett is currently a 3rd year Ph.D. student in the Computer Science department at UC Santa Cruz. His research specialization is Game Studies, though he has interests in Simulation, Computer Graphics, and HCI as well. He has extensive experience with game development, having worked in the computer game industry as a programmer, designer, and producer from 1994 to 1999. He is credited on more than a dozen published games, including MechWarrior 2 and F-22 Raptor. He has also worked in the aerospace industry from 2002 to 2005, helping to develop an air traffic control simulation used by NASA. After completing his Ph.D., he hopes to become a professor at a leading university, helping to guide undergrads to become the next generation of game developers and working closely with graduate students to push research on the future of gaming and simulation technology.

Contact Information
Kenneth Hullett
401 Heller Dr. Box 4
Santa Cruz, CA 95064
831-239-5114
khullett@soe.ucsc.edu

We would consent to online public dissemination of our white paper.

Use of Funds
The main use of the funds would be as a scholarship for the primary researcher. Other uses include:

- Travel: We intend to publish our work in major conferences, such as AIIDE or the Game Developer’s Conference. Funding to cover travel expenses would allow us to present the work in person and receive feedback from peers in the field.
- Assets: Funds may be used to purchase commercially available content packs, as well as to hire student workers to create new content.
- Equipment: Funds may be used to purchase computer equipment and software as needed during the project development.

2 http://www.etc.cmu.edu/projects/hazmat_2005/
3 http://www.gamasutra.com/features/20001025/schaefer_02.htm