The HEAT 3D Engine

Current 3D engines have multiple functionalities that are used to create powerful games and simulations. However, educational developers may not have the means to gain access to all of the engine’s functionalities. Commercial engines can be expensive to license. Open access engines do not offer many of the instructional components needed to leverage the best options for training situations. Creating a personalized 3D engine can ensure access to the latest in 3D rendering and physics, while integrating “special” functions that implement instructional design and learning theory attributes. A personalized 3D engine allows designers/developers to control attributes of the applications made from it, removing many of the constraints that educational game and simulation designers normally face.

With these ideas in mind, the HEAT engine was created for the specific purpose of having access to features that do not exist in other 3D engines. The HEAT engine contains “standard” functionality similar to that of other popular engines (e.g., Unreal 3, Quake 3). Standard functionalities consist of 3D graphics within 3D environments, physics for any force resolution that exists within the environments, networking for multi-player experiences, and 3D audio. Most commonly, the Unity and Caspian Learning engines are being used for creating applications with educational purposes, but cost, distribution and design constraints still persist.

Built-in Features

Saving: The Utah State University group led by Dr. Brett Shelton created the HEAT 3D engine and is responsible for the IP and modifications behind it. Consequently, they can “plug-in” any feature they envision for a variety of instructional purposes. For example, a feature was derived that consists of saving, replaying, and the possible regeneration of any scenario. That is, each scenario can be saved and played back for review during a debrief session to aid in learning. The data that is saved for play back is kept separate from the engine, which can be transferred and viewed between users (teacher and student). The transfer and distributions of the same pre-built or saved scenarios can be viewed and experienced by a potentially unlimited number of people. Distributing saved simulations scenarios that require “fixing” would “standardize” the assessment for all learners. In essence, giving students distributed across time and distance the same “saved” scenario and letting each student fix, or alter the scenario in the way they think is best, allows instructors to assess performance in open-ended complex simulation environments that previously was not an option.

Regen: A saved scenario can be regenerated (regen for short) at any point along the saved timeline. The regen can be used to create multiple outcomes based on any single scenario. To illustrate, consider Figure 1 in which a key decision was made during the simulation at 2 minutes, represented at point "C". During the playback review of a saved scenario, the facilitator (instructor) can start recording a new
scenario at the key decision “C” time creating a new saved scenario, leading to a new outcome. All the while, the new scenario 2 contains exactly the same saved actions in scenario 1 from the time 00:00 to 2:00. Said another way, scenario 1 at time A – C contains the same saved actions as in scenario 2 at A – C. To further illustrate how regen works, during play back of scenario 2, another key decision was made at point E along the timeline now named scenario 2. A new regen from point E could generate yet another completely new scenario, designated scenario 3. Note that all saved behaviors and actions in the 3D environment are identical from points A – C – E within scenarios 2 and 3. All three scenarios look the exact same between points A-C. Yet all three scenarios diverge from each other, having completely different outcomes. Points B, D, and F are the arbitrary "end" points in the simulation, dictated by completion of the activities or when the facilitator stops the recording of the scenario. This regen feature is unique because 1) it gives learners the opportunity to immediately learn from their mistakes without having to repeat already correctly performed actions and 2) it allows the key learning aspects of review and replay, controlled by an instructor or a student, to assist in identifying alternatives during critical points of a simulation. No other engine employs this functionality within open ended 3D learning environments.

Decision Tree: Another key attribute implements the use of a decision tree for tracking complex decision-making by a user, allowing for further capabilities such as automated assessment and feedback. When the user focuses on a point of instruction within a simulation, the user is presented with different choices or decisions that should be made. After each made decision, the user is presented with new choices that are related to the previously made decision. After a given user action, the choice or decision is noted within the application. Having the means to easily modify the way complex decision trees are implemented within a simulation offers the instructional flexibility for multiple scenarios,
changes in the way decisions are presented or manifested, and be modified along the spectrum for novice-to-expert education. The stored decisions can be exported to a text log file and that file can then be emailed to a facilitator or teacher for assessment.

**Automatic Assessment:** A key feature for instructional purposes within open ended 3D environments is the opportunity of automatic assessment. User actions and decisions are tracked during the simulation which then are assessed on accuracy, completeness, and timing. This assessment can be used to train users during scenarios through heads-up feedback displayed as an overlay on the simulation scene. In addition, the feedback can be “hidden” to allow the user participation without the added scaffolding. Hiding the heads-up assessment offers the instructor the option for testing strategies, especially when combined with the standardized scenarios previously described. Working with the decision tree feature, the assessment feature stores the saved data into a separate log file (text file) that can be exported and viewed / annotated on its own. This log file can be emailed to a facilitator or teacher for assessment, or printed and copied by a student.

The HEAT 3D engine is designed to create high-fidelity instructional simulations. In particular, the engine allows the learner to experience cause-effect elements presented within the system. The automated assessment is required to record and analyze these learner movements. The integrity of tasks that are dependent or independent on the completion of another task must be maintained. The decision tree, embedded in the engine, is utilized to determine dependencies and scores are awarded within the decision tree by the engine. For example, a (dependent) learner has to do A before B (accuracy, completeness) or (independent) learner can do A or B at any time to earn the completeness points. Figure 2 illustrates an example of the operation through a decision tree.
Currently, automated assessment is a feature under research and is not implemented with other simulation assessment techniques during after action reviews (AARs) or debrief sessions. The key to leveraging asynchronous assessments is how the HEAT engine allows saved scenarios to be experienced from a 1st person perspective by the instructor. The scenario files created by the HEAT engine can be electronically sent to an instructor just like the summary (potentially annotated) log files can. The assessment of students’ saved scenarios can be completed at a time and place that different than when created. The scenarios can be practiced, recorded, and re-recorded as many times as the student wishes, sent to the instructor, annotated and graded, sent back to student with requests for regens from any point along the timeline.

**Implementations**

The HEAT engine, with a potential for use in any virtual environment imagined, has been used to create two different simulation applications thus far. The first, called the HEAT first-responder training application (Hazard, Emergency and Accident Training), is a multi-player simulation designed to train the Incident Commander (IC) on the protocols when arriving at the scene of a house fire. The second simulation, FIT (Forensics Investigation Training), is a single-player simulation designed to give forensic investigators the opportunity to ascertain the cause of a fire. Both the HEAT and FIT simulations implement the features described above.
HEAT Simulation

Dr. Brett Shelton of Utah State University was contacted by the Utah Institute for Emergency Services and Homeland Security for the purpose of building a 3D simulation application that trains first responders in emergency situations. The simulation created is a 3D networked multiplayer game-like environment in which players have the opportunity to not only learn and practice tactical response skills in a controlled, operations based real-time instructional environment, but also to practice and improve skills dealing with incident command or an incident commander.

The HEAT simulation has two parts, the first part is designed so a facilitator can create and start a scenario. In the second part, learners interact and participate in the open ended 3D environment. As seen in Figure 3, section (1) is where the facilitator creates the scenario and (2) is where the saved scenarios are loaded that are used for playback. (3) The overview is an outline of the house where the facilitator can see the location of each participant. The orange circles indicate that participants are on the first level of the house and the silver circles indicate a participant is on the second level. This outline view can be watched while the simulation is being recorded as well as during the review of the playback. (4) The player views are the first-person perspective view of each participant within the simulation. The facilitator can see what each participant is looking at and what they are doing. During the playback, all participants can watch this view to see what everyone was doing during the simulation. (5) After a saved scenario is loaded, the facilitator uses these controls to view the playback much like a VCR (play, pause, fast-forward). (6) The facilitator types in a new name to regen a scenario from the currently loaded scenario. As seen in Figure 4, the facilitator can also place an emphasis on a player by enlarging the player view, which in effect switches the player view with the top-down, 2D overview.
Figure 3. Screenshot of facilitator view within the HEAT simulation.
Figure 4. Screenshot of player view switched with overview for added emphasis.

The HEAT simulation is a multiplayer simulation meaning that more than one person can participate in the training scenario at a time. Each participant has a first-person view of the 3D environment. Figure 5 shows the perspective of two participants watching another participant spray the fire in an upstairs bedroom.
Figure 5. First-person perspective on the left and the right shows multiple players within the HEAT simulation.

FIT Simulation

The FIT simulation is meant for a single person to use while learning about, or being assessed in their knowledge of, fire investigation. The simulation contains many of the same features as the HEAT simulation, such as first-person perspective and saving, and replaying scenarios. However, the FIT simulation implements the decision tree and automatic assessment. When the learner focuses on the source of the fire, as seen in Figure 6, the user is presented with different choices or decisions that should be made in the investigation process (see Figures 7 and 8).
Figure 6. Screenshot of the FIT simulation where a user can investigate the source of a fire.
Figure 7. The user is presented with the choice to investigate the burned area.
After each decision is made, the user is presented with new choices that are related to the previously made decision. After the user makes a key decision, each choice or decision is stored. The stored decisions can be exported to a text file and that file can then be emailed to a facilitator or teacher, with or without annotations, and with or without automated assessments present. The saved scenarios themselves may also be exported to the instructor, for first-person replay and review of the student’s exact actions within the scenario.

Contact Information

To find out more about the HEAT engine, or the H.E.A.T or FIT applications, please contact:

Brett Shelton, PhD (Project Director): brett.shelton@usu.edu
Alan Hashimoto, MFA (Project co-Director): alanhashimoto@gmail.com
Mary Ann Parlin, PhD (Project Manager): maryann.parlin@gmail.com
Related References


Under Review


Software Release

Shelton, B.E. (2010). Forensics Investigation Training (2.1) [computer program]. Location: Utah State University.

Shelton, B.E. (2009). Hazard Emergency and Accident Training (1.52) [computer program]. Location: Utah State University.