Using Genetic Algorithms to Automate Scenario Generation and Enhance the Training Value of Serious Games for Adaptive Instruction

Robert Sottilare[1], Christopher Ballinger[1], MAJ Matthew Litvinas[2], Christopher McGroarty[3] & Shuowen Hu[4]

[1]Soar Technology LLC, [2]US Army, [3]US Army Soldier Center & [4]US Army Research Laboratory
bob.sottilare@soartech.com; christopher.ballinger@soartech.com; matthew.c.litvinas.mil@army.mil; christopher.j.mcgroarty.civ@army.mil; shuowen.hu.civ@army.mil

Abstract

This paper discusses an end-to-end process to enhance the variability of serious game scenario conditions to support adaptive instructional strategies that select changes in scenario difficulty to match learning objectives to trainee capabilities. This process can also be used to model or recognize human responses to serious game stimuli to support trainee feedback during after-action reviews (AARs). Adaptive instruction is a learning experience where interventions (content selection and feedback) are tailored for each individual trainee or group of trainees. Adaptive training has been shown to be as much as one to two standard deviations more effective than traditional classroom methods. Our process starts with a few parent scenarios. The population of parent scenarios is automatically expanded using a Novelty search (genetic algorithm) methodology to optimize feature variability in the game-based scenarios. Specifically, this paper focuses on the process of creating new training scenarios of varying difficulty and conditions based on changes in scenario environmental factors (i.e., light, precipitation, wind, and cloud cover). The resulting new scenarios may then be run within a serious game with embedded intelligent agents to represent realistic entity behaviors under the scenario-specified conditions. The ability to morph conditions and scenario difficulty during an adaptive training event is a significant benefit of this process.

Introduction

Adaptive training is a paradigm where the instructional content and interventions (e.g., feedback or direction) are tailored by an intelligent computer process facilitated by an adaptive instructional system (AIS; Sottilare & Brawner 2018). AISs optimize learning for each individual trainee or team of trainees by intelligently selecting/modifying content or conditions and by intervening with trainees (e.g., offering feedback, direction, or recommendations). A major tenet of adaptive training is the ability to accelerate learning by tracking trainee progress toward a set of defined learning objectives and focusing on concepts in areas where trainees need improvement (e.g., land navigation skills). This requires more content than traditional, one-size-fits-all approaches where everyone receives the same content.

Serious games refer to computer-based games that are designed with a primary purpose other than pure entertainment. These games are created to serve educational and training goals by simulating real-world situations defined by scenarios, plausible situations composed of a sequence of events, or a set of conditions. Scenarios can be used in various contexts such as business, situational analysis, environmental modeling, and training & education.

Serious games often incorporate elements of gameplay to engage users while delivering specific content that stimulate the growth of knowledge and skills. In this paper, we discuss how serious games may be enhanced through adaptive training processes. To optimize learning experiences, we developed automated scenario generation (ASG) processes to create a large volume of new training experiences for serious games. ASG increases the number of realistic conditions that trainees encounter within serious games, but without the cost of large amounts of additional game developer or scenario authoring labor. The result is a greater number of training sets and repetitions that are sufficiently different, but also representative of the operational conditions likely to be encountered in work environments. This research described in this paper enhanced trainee experiences and made serious games both more efficient and effective learning tools.

Serious Games in Adaptive Training

AISs tailor training experiences to support defined training objectives, but often require significantly more content (e.g., scenarios) to support an array of potential training paths which are a structured sequence of activities designed to help trainees acquire a specific set of skills (Folsom-Kovarik & Brawner, 2018). Training paths are often used in educational and professional development settings to guide trainees through a progressive series of lessons or training modules leading to the achievement of a credential. Credentials are documentation of skill proficiency and are often used to establish someone's eligibility or suitability for a particular role, job, or task.

In adaptive training using serious games, trainees attempt to achieve a set of training objectives as part of a simulation scenario intended to represent a real-world experience. Scenarios in serious games enable trainees to interact within a simulated environment over a sequence of events which exercise a set of needed skills related to the training objectives. Events may be adapted to change the training path or the difficulty level of the experience to optimize the effectiveness and efficiency of the training.

Importance of ASG to Adaptive Training

The goal of ASG in adaptive training using serious games is to enhance the experience by dynamically tailoring scenarios to the individual needs (e.g., skill gaps) and skill levels of each trainee. Serious games are composed of interactive and engaging simulation events driven by dynamic entities and phenomena that are designed to provide unique training experiences.

ASG involves the use of algorithms and artificial intelligence methods to create scenarios that can develop or exercise each trainee's needed skills. The adaptive training capability within the serious game tailors the scenario events and conditions to keep the training engaged and optimize their performance with respect to their assigned training objectives. This system adaptivity also helps ensure that the training experience is challenging yet achievable. Adaptive training within serious games supports personalization including selection of scenarios with tailored levels of difficult to match the capabilities of the trainee. Rapid skill progression, engagement, skill transfer, and accelerated learning are also benefits of adaptive training facilitated by a wide selection of serious game scenarios.

By modeling trainee performance relative to training objectives, AISs can tailor scenarios to stimulate the use of specific knowledge, skills, and abilities. ASG enables a variety of scenarios to be available to support tailored training experiences, and this level of personalization maintains the difficulty level to be consistent with the trainee’s capabilities, ensuring a balance between difficulty and motivation and thereby maintaining engagement (Vygotsky’s zone of proximal development, 1983).

Designing scenarios that align with the trainee's expected skill progression can gradually increase scenario complexity as their skills improve. ASG processes integrate real-world situations and challenges into scenarios to enhance the practical application of learned skills (transfer of learning; Perkins & Salomon 1992). ASG also maximizes training outcomes (e.g., enhanced knowledge and skills) by optimizing the use of time and resources through adaptive scenario generation. By incorporating ASG into serious games for adaptive training, organizations and educational institutions can provide a more tailored and effective learning experiences to their trainee population, ultimately improving the retention, transfer, and application of acquired skills.

Limitations of Manual Scenario Generation

In this section, we discuss the advantages of ASG by exploring the limitations of manual scenario generation processes including time/cost, expertise, subjectivity, limited coverage of scenario conditions, scalability, reproducibility, and maintainability. Creating scenarios manually can be a time-consuming and costly process that requires significant effort and expertise to carefully design scenarios that adequately cover the range of possible real-world situations. Manual scenario generation is subjective, as it heavily relies on the knowledge and biases of the individuals creating the scenarios. This can lead to scenarios that may not accurately represent all possible (or even all required) real-world conditions. There is a risk of overlooking certain edge cases or unusual conditions that an automated or more systematic approach might more easily capture.

Manual scenario generation may become impractical when dealing with complex systems or large datasets. As scenario complexity increases, it becomes challenging to manually create enough diverse scenarios to thoroughly represent the target operational environment. The reproducibility of manually generated scenarios can be a challenge. Different individuals may interpret requirements differently, may lead to inconsistencies in scenario design. Automated methods often provide more consistent and reproducible results. When training requirements change or evolve, updating manually generated scenarios can be cumbersome and time intensive. Automated methods may offer more flexibility in adapting to changes by standardizing scenario formats thereby reducing the costs to maintain scenario databases.

Prior Scenario Generation Approaches

In this section, we explore both manual and automated approaches to ASG.

Manual Approaches

Three widespread manual approaches to scenario generation include expert-crafted scenario generation, user-generated content, and storyboarding. For expert-crafted scenario generation (ECSG), domain experts or instructional designers manually create scenarios based on their expertise. They carefully design the context, challenges, and goals of the scenario to align with the learning objectives of the serious game. The advantages of ECSG methods are that they ensure scenarios are tailored to specific training goals and objectives. For user-generated content (UGC) approaches, users or trainees create their own scenarios within the serious game environment. Trainees design challenges, set objectives, and define the parameters of the scenario. The advantage of this approach is that it promotes user engagement and creativity and offers a diverse range of scenarios as users bring their unique perspectives and experiences. However, quality control can be an issue, and not all users may have the skills or motivation to create effective scenarios. Storyboarding is an approach where scenario content is visualized using sequential panels of pictures to define a narrative being implemented in the serious game. Storyboards outline the sequence of events, decision points, and interactions within the scenario. Storyboarding provides a visual representation of the scenario flow, helping in the early identification of potential issues, and it facilitates collaboration among instructional designers, game developers, and subject matter experts. A limitation is their static nature that might not capture dynamic aspects of gameplay.

Automated ASG Approaches

Rule-based approaches offer a structured and systematic way to automate scenario generation, allowing for a balance between scenario control and variability in serious games. The choice of method often depends on the specific requirements for a given training context and the desired learning outcomes. Rule-based ASG methods involve defining a set of explicit instructions and guidelines to create new serious game scenarios. These instructions can cover various aspects of scenario design, such as environment setup, character behavior, and trainee progression. Some common rule-based methods used in scenario generation include scripted events, decision trees, randomization methods or goal-based scenario generation.

Predefined scripts or sequences of events are created to guide the progression of the scenario. These scripts dictate the timing or frequency of events, triggers, and outcomes of events within the game. The advantage of scripted events is their fine control over the narrative and scenario tempo. The major challenge associated with this approach is limited variability and adaptability which may result in a linear gameplay experience. Decision trees are hierarchical structures where each node represents a decision point, and branches represent possible outcomes. Players' choices at decision points lead to different branches which helps shape the scenario. This approach offers a structured approach to scenario branching and allows for multiple paths and outcomes based on player choices or progress toward objectives. The limitation of this approach is the ability to design comprehensive decision trees can be complex and may not support a needed balance between meaningful choices and a clear, simple path to a plausible outcome.

Randomization may be used to introduce variability in scenarios while adhering to predefined constraints. This involves setting boundaries on random choices to ensure that generated scenarios remain meaningful and aligned with training objectives. Scenarios may also be generated based on predefined goals or training objectives. Rules determine how different elements within the scenario contribute to achieving these goals. The advantage of this approach is that it aligns scenario content with training objectives. Machine learning methods can be employed to automate scenario generation in serious games, offering adaptive and personalized experiences for users. Machine learning methods include but are not limited to Markov processes (Zook et al 2012), reinforcement learning (RL; Rowe et al 2018), neural networks (Luo, 2016), generative adversarial networks (GANs; Chen et al 2018), deep learning for natural language processing (Gee & Jenkins 2019), and evolutionary algorithms, such as genetic algorithms (GAs; Sottilare, 2018).

For genetic algorithms (GAs), parameters of scenarios are treated as genes, and the fittest scenarios survive and reproduce. This offers a nature-inspired approach to scenario generation that can optimize scenarios for specific criteria over time. A major challenge to this approach is the task of defining appropriate fitness functions and encoding scenarios as genotypes. Computational intensity may be a concern for very complex genomes. Novelty search (Lehman & Stanley 2011) is an evolutionary algorithm that emphasizes the exploration of diverse and previously unexplored areas of a solution space, rather than focusing solely on optimizing a specific objective function. Novelty search was used to generate new scenarios from a small parent population that included genes for the size and shape of the room, the number of armed enemy forces present in the room, the number of non-combatants present in the room, and obstacles in the room (Figure 1).

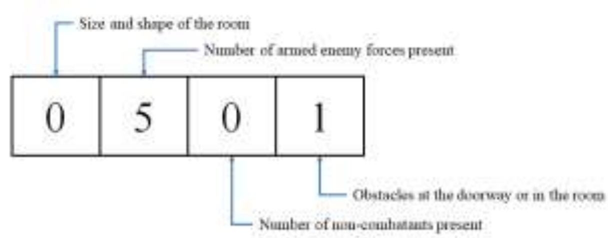


Figure 1. A genome for a room clearing task.

The combinatorial optimization element of the fitness function eliminated new scenarios that had too few or too many people and barriers present. Additional scenario diversity was achieved through random placement of people and barriers in the room.

Methodology for a GA-Driven ASG Approach

In this section, we describe novelty search methods used to generate a diverse set of relevant scenarios from a small population of parent scenarios. Novelty search was chosen to generate the broadest and most diverse population of environments possible for use in training complex military tasks. In general, novelty search fitness criteria focus on creating new unique scenarios and avoiding duplicates. The process used in our ASG approach is detailed in Figure 2.

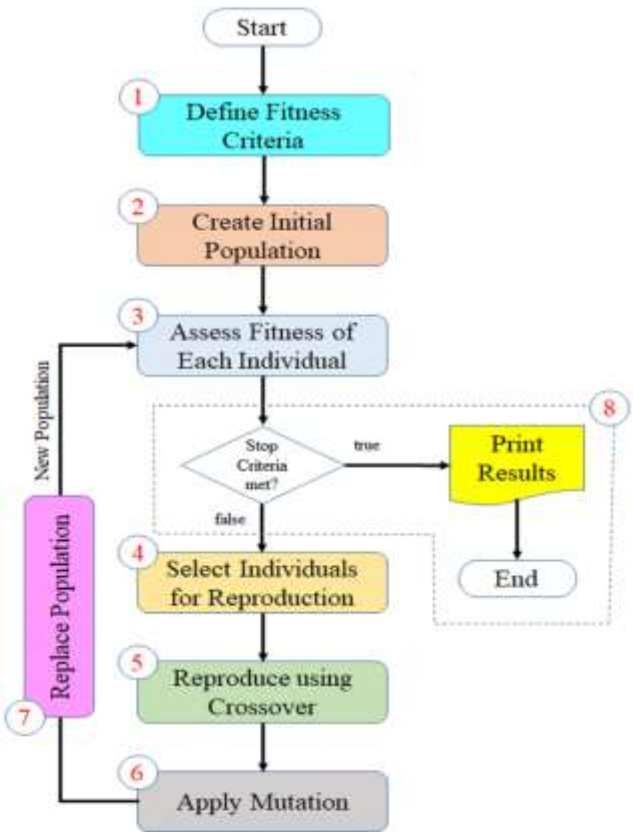


Figure 2. A novelty search approach to ASG

Our goal in selecting a genome centered on environmental conditions such as ambient light, cloud cover, wind and precipitation was to create a population of scenarios that varied in meaningful dimensions. Specifically, we chose factors that could impact the performance of a variety of individual and team tasks. Unlike the joint novelty search and combinatorial optimization approach used by Sottilare (2018), we chose to use parameters and their numerical limits (e.g., rates and intensity) to bound child scenarios to only those that were plausible. For example, we did not include scenarios that reflected negative precipitation.

GA Parameters & Gene Representation

For the training environment for serious games, we chose to represent the following environmental parameters in our genome:

- PRECIPITATION_TYPES = ['rain', 'sleet', 'snow']
- PRECIPITATION_INTENSITY = ['very light', 'light', 'moderate', 'heavy']
- CLOUD_COVER = ['none', 'scattered', 'partly cloudy', 'mostly cloudy', 'full cloud cover']
- WIND_DIRECTIONS = ['N', 'NNE', 'NE', 'ENE', 'E', 'ESE', 'SE', 'SSE', 'S', 'SSW', 'SW', 'WSW', 'W', 'WNW', 'NW', 'NNW']
- LIGHT_SOURCES = ['sun', 'moon']
- LIGHT_DIRECTIONS = ['night', 'dawn', 'day', 'dusk']
- precipitation rate = [0.0, 1.0]
- wind intensity = [0.0, 50.0]
- light intensity = [0.0, 1000.0]

Environmental parameters were chosen as universal variables that influence the complexity of all military training tasks by impacting sensor performance, visibility, trafficability (ability to maneuver on the terrain), and aviation performance. Military planners and commanders must consider these environmental factors when designing operational plans, selecting equipment, and conducting training. Adaptiveness within different environmental conditions are essential elements of military training and operations.

Fitness Function Design Considerations

The goal for our novelty search approach was to generate a child population that was representative of the entire search space represented by the selected environmental parameters (i.e., wind, precipitation, light intensity). We considered using elitism as a fitness function, but this caused convergence issues. In the context of novelty search, elitism refers to a mechanism that preserves only the most novel and diverse solutions within a population across generations. Issues arising from the use of elitism can be traced to three primary causes: loss of diversity, stagnation, overemphasis of local optima, and resistance to innovation.

Elitism is supposed to solve the problem of preserving the best individuals in each generation to expedite convergence and maintaining the potential for generating optimal solutions. Instead, it can cause premature convergence by favoring the most novel individuals too strongly. If the algorithm keeps selecting a small subset of highly novel solutions, it might ignore other potentially diverse regions of the search space, leading to a lack of exploration and innovation. Elitism might also cause the algorithm to focus too much on a specific set of solutions, preventing the introduction of new, possibly more innovative solutions. This can result in stagnation and hinder the discovery of novel and beneficial traits. Finally, elitism can lead the algorithm to favor individuals in local optima, even if those solutions are not globally optimal. This can hinder the ability of the algorithm to escape local optima and explore new, potentially more promising regions.

Based on these issues, we adopted a non-elitist approach in novelty search which aims to explore a broader range of possibilities and solutions without favoring any subset. Non-elitism enables the search algorithm to explore across a wider space, potentially uncovering more diverse and innovative solutions. Specifically, we implemented a comparison of solutions based on their Euclidean distance. Each solution in the population is represented as a point in the search space. This representation could be a vector of parameters, a neural network architecture, or any other suitable encoding for each unique problem space. Euclidean distance is computed between the feature vectors or representations of different solutions in the population. The Euclidean distance *d* between two solutions *A* and *B* in a multidimensional space is given by the formula:

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \qquad \text{(Eq. 1)}$$

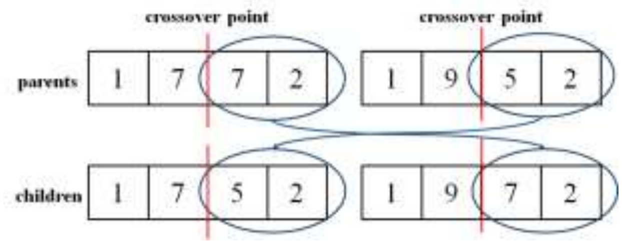
Where *n* is the number of dimensions or features in the solution representation. For features that are enumerative values, the distance between those values are considered 0 if they are the same and 1 – (1 / possible_enum_values). This means that the more possible enumerative values a feature has, the greater the novelty will be when features don’t match, slightly encouraging exploring features that have more possibilities. We also normalize the values of numerical features to values 0-1 when calculating Euclidean distance. While some numerical features are may have values in the range 0-10, others may have values I the range of 0-1000. Without normalization, features with large value ranges tended to dominate the novelty score.

The novelty score for a solution is then calculated based on its distance from its *k* nearest neighbors in the population. The idea is to balance exploration (novelty) with exploitation (optimization). This can be achieved through a weighted sum or other combination methods. Solutions are then selected for reproduction or survival based on their combined fitness values. By incorporating Euclidean distance into the fitness function, novelty search encourages the exploration of diverse regions in the solution space, leading to a more comprehensive exploration of the problem landscape. The use of distance metrics, such as Euclidean distance enables the algorithm to identify solutions that are dissimilar to those already present in the population, promoting the discovery of novel and potentially innovative solutions.

Crossover and Mutation Strategies

To maintain genetic diversity, which is essential for avoiding premature convergence and enabling a broader range of solutions, we adopted a random crossover. For each gene position in the offspring, the genetic material is randomly selected from either of the parent solutions.

Figure 3. A rule-based crossover strategy



Unlike the crossover strategy shown in Figure 3, a random crossover strategy means that there is no specific pattern or order in choosing genes from the parents. In a basic random crossover strategy, each parent has an equal probability of contributing genetic material to the offspring. This ensures a fair and unbiased combination of traits from both parents. The random selection process is repeated for each gene in the offspring's genome. This can result in a diverse combination of genetic material, as different genes may come from different parents. The randomness in crossover helps introduce diversity into the population. It prevents the algorithm from being overly deterministic, promoting exploration of the search space.

After the crossover operation is completed, a subset of individuals (or all individuals) in the population is selected for mutation. The decision of which individuals to mutate can be based on various criteria, such as a fixed probability for everyone. We adopted a random uniform mutation which is often applied independently to each gene selected for mutation. This allows for a diverse set of changes across the genome of an individual. For each selected individual, specific genes are chosen for mutation. This can be done by randomly selecting individual genes based on a mutation probability. The mutation probability determines the likelihood of a gene undergoing a mutation. Once the genes to be mutated are identified, random changes are introduced. For a random uniform mutation, the changes are typically drawn from a uniform distribution. This means that the new value for a gene is chosen randomly from a specified range around the current value. The range the value can mutate to is plus or minus a fixed percent (usually 1%) of the total range of values. So, a feature with range 0-100 mutates current values in a range -1 to 1, while a feature with a range of 100-1100 mutates in a range of -10 to 10. In addition to random uniform mutation in a range near the current value, we allow an even smaller probability that catastrophic mutation may occur, a random uniform mutation to any valid value for the feature, regardless of range or distance to the current value. The mutation probability is a crucial parameter that influences the extent of mutation in the population. A higher mutation probability results in more genes being mutated, leading to increased exploration of the solution space. The introduction of random mutations helps strike a balance between exploration (searching for novel solutions) and exploitation (focusing on promising solutions). It prevents the population from converging prematurely to a suboptimal region in the solution space. The entire process of crossover, mutation, and selection was repeated over multiple generations, allowing the population to evolve and adapt over time.

Results and Discussion

This research produced a GA process using novelty search that rapidly generated 300 plausible scenarios in a few minutes. The resulting scenarios can be used in a variety of training contexts with a broad spectrum of training objectives. The scenario genome is tied to the conditions under which tasks are conducted and measured, but not directly tied to a type of task or trainee behaviors. The ability to rapidly generate a large number of scenarios that influence task performance set the stage for effective adaptive instruction that can be tailored to enable trainees to experience task execution under a variety of dynamic conditions that foster training transfer and trainee flexibility (Cheng & Hampson, 2008).

While integrating the novelty search approach to ASG within serious game frameworks can be beneficial, there are several potential barriers and challenges that developers may encounter. These barriers include technical, design, and practical considerations such as algorithm complexity, computational resources, evaluation metrics, user acceptance, alignment with training objectives, and balancing exploration and exploitation.

Implementing and fine-tuning evolutionary algorithms, including novelty search, can be complex. Developers need a solid understanding of the algorithm's parameters, such as mutation rates, crossover strategies, and the weighting of novelty versus traditional fitness. To overcome this barrier, we recommend more user-friendly interfaces to simplify the configuration and adjustment of algorithm parameters to provide a greater understanding of their effect. Evolutionary algorithms, especially with large populations and complex scenarios, can be computationally intensive. This may pose challenges in terms of processing power, memory requirements, and runtime. We recommend optimizing the algorithm and scenario representations to reduce computational demands, and parallel processing or distributed computing for large-scale scenarios.

Defining appropriate evaluation metrics, especially for the novelty component, can also be challenging. Deciding how to quantify and measure the novelty of scenarios currently lacks clear guidelines. We recommend exploring ways to validate the effectiveness of the novelty metric in promoting diverse and engaging scenarios. This could include combinatorial optimization methods.

Users may have expectations based on traditional serious game design approaches. Novelty search may prioritize diverse scenarios over those directly aligned with training goals. We recommend guidelines for designers to develop composite fitness functions to balance novelty with traditional fitness and assess how well the scenarios align with training objectives through testing and user feedback.

Introducing novelty-focused scenarios might initially be met with resistance or confusion, especially if integration with serious games is complicated. The good news is that a mass of serious games is beginning to integrate dynamic weather phenomena into their entity performance models. According to Weideman (2024), "dynamic weather in games adds immersion and enhances the gaming experience by creating tension and peaceful moments". This tension adds to the realism of executing assigned tasks under a variety of conditions. While these are not serious games, Weideman lists the following games with realistic weather effects that influence game performance:

- No Man's Sky (fire storms and tornadoes)
- Sea of Thieves (wind direction/speed affect navigation; dynamic rain and thunderstorms)
- Last Train Home (cold temperatures cause player avatars to fall ill)

Conclusion

This paper discussed research that examined methods to automate scenario generation using genetic algorithms. A major finding of this research was the ability to use novelty search with fitness functions that enable exploration and exploitation. The methods described herein facilitated the ability to generate a diverse set of scenarios using novelty search and eliminate unrealistic scenarios using a fitness function with bounded variables to ensure plausibility.

A major contribution of this research is a paradigm shift in ASG from the creation of narrowly focused scenarios that are inclusive of entities and entity interactions to a broader base of scenarios that focuses on the conditions under which entities in serious games conduct training and operational tasks. The process described in this paper can be broadly applied to a variety of training tasks, simulations, and serious game environments.

Acknowledgements

The research described in this paper has been sponsored by the U.S. Army (contract W912CG21C0002). Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

References

Chen, Y., Wang, Y., Kirschen, D. and Zhang, B., 2018. Model-free renewable scenario generation using generative adversarial networks. IEEE Transactions on Power Systems, 33(3), pp.3265-3275.

Cheng, E. W., & Hampson, I. (2008). Transfer of training: A review and new insights. International journal of management reviews, 10(4), 327-341.

Folsom-Kovarik, J.T. & Brawner, K. (2018, May). Automating Variation in Training Content for Domain-general Pedagogical Tailoring. In Proceedings of the 6th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium. US Army Research Laboratory, Orlando, FL.

Gee, L.O. and Jenkins, I.R., 2019. An approach to automate accident scenario generation using recurrent neural networks.

Lehman, J. and Stanley, K.O., 2011. Novelty search and the problem with objectives. Genetic programming theory and practice IX, pp.37-56.

Luo, L., Yin, H., Cai, W., Zhong, J. and Lees, M., 2016. Design and evaluation of a data-driven scenario generation framework for game-based training. IEEE Transactions on Computational Intelligence and AI in Games, 9(3), pp.213-226.

Perkins, D.N. and Salomon, G., 1992. Transfer of learning. International encyclopedia of education, 2, pp.6452-6457.

Rowe, J., Smith, A., Pokorny, B., Mott, B. and Lester, J., 2018, May. Toward automated scenario generation with deep reinforcement learning in gift. In Proceedings of the Sixth Annual GIFT User Symposium (pp. 65-74).

Sottilare, R.A., 2018, September. A hybrid machine learning approach to automated scenario generation (ASG) to support adaptive instruction in virtual simulations and games. In The International Defense & Homeland Security Simulation Workshop of the I3M Conference, Budapest, Hungary.

Sottilare, R. and Brawner, K., 2018, June. Component interaction within the Generalized Intelligent Framework for Tutoring (GIFT) as a model for adaptive instructional system standards. In Proceedings of the 14th International Conference of the Intelligent Tutoring Systems (ITS) Conference, Montreal, Quebec, Canada.

Weideman, M. (2024). 10 Games Where Weather Impacts Gameplay. Game Rant.

Zook, A., Lee-Urban, S., Riedl, M.O., Holden, H.K., Sottilare, R.A. and Brawner, K.W., 2012, May. Automated scenario generation: toward tailored and optimized military training in virtual environments. In Proceedings of the international conference on the foundations of digital games (pp. 164-171).