ISGG

Based on the related work and our own experiences building a using a prototype that has similarities with the <e-learning> project approach, we have identified a collection of goals, or high level requirements, for an Intelligent Semi-automated Game Generation (ISGG) approach.

* The ISGG should support the generation of SEGs with diverse learning objectives (e.g., SE topics, both technical and soft-skills), diverse player audiences (K-12, post-secondary, industry training, continuing education), and diverse delivery options (augment traditional lectures, game as an e-learning training module). This will help to encourage broad adoption, as games can be provided to meet the needs of educators with different backgrounds, teaching different SE topics in different organizations. This mass customization can be achieved with a product-line [13] approach. Re-usable SEG patterns can be defined as core assets with variation points and composed into customized solutions.
* The ISGG should apply intelligent algorithms to search for, select, tailor, and instantiate re-usable game assets. The trade-offs among alternative algorithms will be analyzed; the most promising candidates will be explored with prototypes using existing libraries. While considering the alternative algorithms, questions to consider include how to represent our search problem. For example, in evolutionary algorithms, what would be a “good” fitness function? How should mutations be created? How can variability in the ranking be parameterized? How does the choice of algorithm affect the representation of the search space (direct, indirect)? As the game generation is performed off-line, the performance of the algorithms is not a high priority concern.
* The ISGG should support variable degrees of user interaction (from push-button game generation to a more user guided approach). This will help educators generate games to meet different needs. For example, a push-button approach may be very desirable to create a very short game to augment a lecture. Educators may want to provide a more hands-on approach, perhaps even to guide the generation of a specific part of a game.
* The ISGG should generate collections of games with variations (not always the same game generated). This can quickly provide new games for students and help keep them engaged.
* The ISGG should provide analysis and reporting features. For example, an automated report summarizing the learning objective coverage for a collection of one or more games would relieve the burden on educators manually collecting data, analyzing data, and preparing reports.
* The ISGG should generate games (scripts) in alternative formats that can be loaded/played in different engines. Although originally considered as one module in the SimSYS game development platform [7], this approach could be utilized to generate games for a collection of game play engines.
* The ISGG support the straightforward maintenance of game core assets. Well modularized repositories are needed to quickly support, for example, updates to standards and new topics. Repositories of learning objectives (derived from standards, for example bodies of knowledge); challenges for learning objectives (quiz items, quizzes, exercises across diverse topics, educational levels); and more traditional game content (images, audio - music, sound effects), story-lines, themes, established characters; styles) will be needed.
* The ISGG tool support should be broadly accessible (web-based), support the generation and preview of games, in addition to load/edit/save features. The overall architecture is Model-View-Controller; the Model is further structured using a repository style. The repository is organized as a collection of partitioned repositories (one for learning objectives, one for challenges, and so on); the clients of the repository in the Model include the search-based algorithm (modularized for potential changes).
* The ISGG approach should be evaluated with a study to quantitatively and qualitatively compare the new approach (with prototype tool) with a manual approach (XML text editing). From the game designers’, data will be collected and analyzed, including the amount of time to generate the game; the number of iterations to finalize the game; the number and kinds of adjustments/modifications made to the game; and the satisfaction level of the game designer for the final game produced. Although we anticipate the semi-automated approach to take less time, it will be important to determine the value of the approach by considering, for example, the amount of rework game designers deem necessary to accept the generated game and their perception of the game quality.