Towards a Lightweight Approach for Modding Serious Educational Games: Assisting Novice Designers

Jacob Dahleen, Alex Hunsberger,
Ryan Weber, Dennis Brylow
Department of Mathematics, Statistics
and Computer Science
Marquette University
Milwaukee, U.S.A.

C. Shaun Longstreet
Center for Teaching and Learning
Marquette University
Milwaukee, U.S.A.

Kendra M.L. Cooper
Department of Computer Science
The University of Texas at Dallas
Richardson, U.S.A.

***Abstract*— Serious educational games (SEGs) are a growing segment of the education community's pedagogical toolbox. Effectively creating such games remains challenging, as teachers and industry trainers are content experts; typically they are not game designers with the theoretical knowledge and practical experience needed to create a quality SEG. Here, a lightweight approach to interactively explore and modify existing SEGs is introduced, a toll that can be broadly adopted by educators for pedagogically sound SEGs. Novice game designers can rapidly explore the educational and traditional elements of a game, with a stress on tracking the SEG learning objectives, as well as allowing for reviewing and altering a variety of graphic and audio game elements.**

Keywords—Serious Educational Games, Visualization Environment, Novice Game Designers

I. Introduction

Serious educational games (SEGs) are a growing segment of the education community's pedagogical toolbox. There is increasing evidence for their efficacy in sustaining engaged learning [11][12][13][22]. SEGs allow for rapid feedback for individually customized educational experiences. Moreover, digital engagement is increasingly ubiquitous, with individuals of all ages growing adept at learning complex tasks online without the use of texts or self-contained tutorials. Nevertheless, the time, financial, and knowledge resources required to develop quality games are out of reach of most pedagogical stakeholders. This is all the more so for those who require a rapid role-out of games with up-to-date, just-in-time information for multiple groups, different grades, and varied subject matter.

The development of a semi-automated game development platform is important because the vast majority of educators do not have the knowledge or time to code their own educational gaming experiences. Semi-automation can assist with two critical functions. First, it can greatly reduce the complexity of game generation, leaving much of the educational content elements to the instructor. Teachers and industry trainers are content experts; typically they are not game designers with the theoretical knowledge and practical experience needed to create a quality SEG. Without game design experience, instructors are left to storyboard an SEG effectively, determine variants for levels of difficulty, identify opportunities for challenges and then establish how results of the challenges affect subsequent game play.

Second, semi-automated game development situates the learning experience for both instructor and students in a way that draws upon an established body of knowledge and structured modes of learning, thereby insuring an SEG experience that is directly tied to selected learning objectives. This latter part is where many SEG programs can fall short for educators. Instructors would need to outline, embed, and track learning objectives for each specific player choice and challenges for individual games. The quality of the SEG game play experience becomes overly dependent on instructors' abilities to develop a quality game and too often puts the burden of game immersion on the player.

In this paper, we introduce a lightweight modding approach, SEG$_{\text{Mod}}$, that can be broadly adopted by novice designers of SEGs. The approach is part of a larger, on-going project, SimSYS, which explores SEG research issues with a development platform for serious educational games, including Game Generation. To keep this proposed approach straightforward, the educational game designers are provided with a relatively modest set of functionality. They can load the game, browse the structure of the game, browse the learning outcomes of a game (from the top down to challenge levels of game play), modify visual and audio game elements, check for errors that may have been introduced, and save their tailored games. Our work is innovative in that the learning outcomes are consistently and explicitly present throughout the game design process and explicitly embedded in game play.

The SEG$_{\text{Mod}}$ approach has been investigated using a scenario based approach. First, a set of high level scenarios are defined. For example:

> *Scenario 1. Amy, a 4th grade teacher, has just finished generating a game to teach her students math. She wants to review the learning objectives and confirm that*

the game has the appropriate assets. Amy opens the game with the Interactive Environment and clicks through each of the screens to insure consistency with the prior week's lessons. As she clicks on the first screen of the opening scene in Act Two, she notices that the protagonist character is too small and out of place. She moves and resizes the character to a suitable location and saves the game.

A set of scenarios such as the one above is used to drive the requirements for the approach and the interface design. Our approach has been validated by developing a prototype tool. The design of the tool has two components and it has been programed using Java, open source tools, and standard libraries. Substantial testing (feature, performance) has been conducted as preliminary validation, demonstrating that the tool supports the outlined scenarios, thereby providing strong indicators the SEG$_{\text{Mod}}$ is a promising research direction.

The structure of the remainder of this paper is as follows. Section II presents background on the broader SimSYS project, to provide the context for this research. The proposed SEG$_{\text{Mod}}$ approach and the validation are presented in Section III and Section IV, respectively. Related Work is in Section V. Conclusions and Future Work are in Section VI.

II. SimSYS Project Background

SimSYS is an on-going project investigating the effective semi-automated development of SEGs [18]. This is an inter-disciplinary project, which integrates three research areas: traditional game design, pedagogical methodologies, and software engineering methodologies. Here, we provide an overview of the SimSYS Game Development Platform (GDP) to provide the context of the new results reported in this paper.

A. *SimSYS Game Development Platform*

The SimSYS GDP is being systematically engineered so that it is well-modularized, extensible, and established on a meta-model foundation [4][7][18]. The overall architecture, illustrated in Fig. 1, is based on the repository style, which consists of a repository that stores knowledge with controlled access, and a collection of knowledge sources that directly interact with the repository (i.e., clients of the repository). This organization has been adopted to support the rapid evolution of content; we recognize there may be redundant storage of assets that increases the storage overhead of this approach.

Here, the repository is organized into domain specific and domain independent sub-repositories. The domain specific sub-repository stores game models (scripts), game play data, and re-usable game assets such as collections of domain specific Challenges (e.g., quiz questions/answers/feedback about software engineering, mathematics, physics, and so on) that are used in the games. The challenge content also captures relationships to educational standards (e.g., SWEBOK [15], [26]), Taxonomy of Learning standards (e.g., Bloom's [1]) and the level of difficulty (defined by educators). Re-usable domain independent assets include the formal SimSYS game specification[17], defined in XSD [27], and a collection of game assets (e.g., sound effects, graphic images). A SimSYS game is organized into Acts, with learning objectives coupled to them. Each Act contains lower-level Scenes, Screens, and Challenges, marking advancing points in the game where learning objectives and game narrative play out. SimSYS games are represented in XML, as Game Scripts.

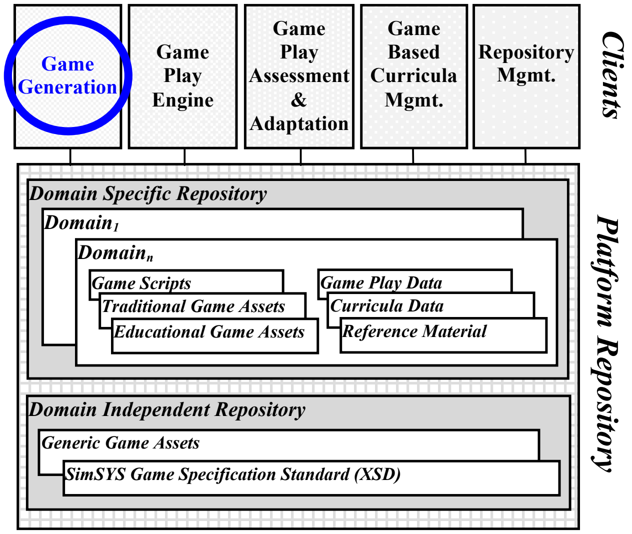


Fig. 1. SimSYS Game Development Platform

The knowledge sources for the platform are modules that interact indirectly via changes to the repository: Game Generation; Game Play Engine; Game Play Assessment and Adaptation; Game Based Curricula Management; and Repository Management. These modules embody knowledge of related standards, including body of knowledge, accreditation, and certification. The stakeholder interacting with a module can work with the content in the repository from a particular view.

The Game Generation module is being explored as a collection of three approaches: 1) a highly automated game generator [7]; 2) a semi-automated game generator (interactive wizard to create and modify a game, guided by an experienced game designer), and 3) a preview visualization tool to provide a lightweight, easy to use tool support for exploring an existing game and making relatively minor modifications to it. The preview tool is a particularly appealing prospect because computer games are an inherently visual medium, and because a textual representation of the complex information encoded in an SEG would likely overwhelm all but the most dedicated novices in a design environment. Introducing the interactive, preview visualization approach, hereafter referred to as the SEG$_{\text{Mod}}$, will be the focus for the rest of this paper.

III. SEG$_{\text{Mod}}$: Lightweight Interactive Modding

The purpose of the SEG$_{\text{Mod}}$ approach is to support designers who only need to review, affect small changes to an existing game, and to make simple edits. Games can be created using, for example, an interactive semi-automated game generation wizard; individual instructors can use our preview tool to contextualize and frame specific SEG scenarios by editing text in speech bubbles, adding information boxes, or modifying just-in-time hints and prompts. They can play or switch out sounds and background music, swap characters, and even tailor the organization of challenge components. An instructor can check and adjust relatively minor details within an SEG environment, or build new SEGs out of existing games stored in a file directory. For example, a fantasy-themed reading comprehension game with a castle backdrop for 6th grade language arts might not be appropriate for adult learners of English. In another situation, a community college instructor teaching a department's common U.S. History course might switch out game props that are contextually relevant to his specific class discussions. Likewise, there will be times when an instructor will want to add, delete, highlight, move, or resize game elements without having to build an entirely new game.

Experienced game designers can use the $\mathrm{SEG}_{\mathrm{Mod}}$ approach as part of the process for creating a game from scratch. These designers can use the interactive, semi-automated game generation wizard to create a game. The $\mathrm{SEG}_{\mathrm{Mod}}$ approach can be used to quickly visualize the overall structure and content of the game, in addition to making limited modifications to it, as the design progresses. The experienced designers can iteratively design, visualize, and make minor modifications to a game. The lightweight $\mathrm{SEG}_{\mathrm{Mod}}$ is envisioned to provide a complimentary alternative to loading and playing the game in the game play engine.

Error detection and a feature for automatically correcting some specific kinds of errors is part of the preview tool. For example, if one instructor drags an asset off-screen rather than deleting it and saves the game, subsequent users of that game would have difficulty addressing that asset, so the automated error correction would reset off-screen assets back onto the interactive display. Likewise, if the game engine did not find appropriate knowledge data associated with learning objectives or is unable to call items from the game asset repository, these errors would be flagged visually in an error report of the preview welcome display as well as on the relevant lower level screens in the interactive display.

The $\mathrm{SEG}_{\mathrm{Mod}}$ approach is not intended to be a full audio/visual SEG development solution, which would need to support major changes to learning objectives or game structure. For that, the SimSYS game development framework provides the game development wizard, wherein an instructor, lab coordinator, supervisor can input data, learning objectives, general assets and assessment parameters for the SEG. This approach differs from other researchers [8][20] in that alternative tools provide different levels of design engagement, which creates a more flexible game development platform. Moreover, the development of $\mathrm{SEG}_{\mathrm{Mod}}$ will allow for easier scalability and adaptations across institutions.

A. Scenarios

A collection of high level scenarios have been defined and refined requirements provide the scope of functionality in the lightweight approach. Several example scenarios are:

Scenario 1. We already have mentioned Amy, the 4th grade teacher above who uses the $\mathrm{SEG}_{\mathrm{Mod}}$ approach to move and resize characters to more suitable locations and saves the game (refer to Section I).

Scenario 2. George, a high school physics teacher, has generated a game to teach his students about kinematics and has opened the game with the $\mathrm{SEG}_{\mathrm{Mod}}$ approach to see what it will look like. Previewing the game, he decides, based on previous weeks' experience, that his students would benefit from the addition of more instruction. He then adds a new character with a speech bubble to provide more information to his students. He also adds hints that remind his students what they have been working on in class that will help them in the game. Further on, he sees an opportunity in character conversations to add some humour to the game, and make the game more personal for his students. He finds the rest of the game acceptable and saves the game.

Scenario 3. Pat, a corporate human resources educator, will use a pre-developed game in her training program for the company's ethics reporting procedure. She wants to preview the game first so that she can make references to it in her face-to-face session with her staff later in the week. After loading the game in the $\mathrm{SEG}_{\mathrm{Mod}}$ approach she runs through the screens in the game. She observes how the game plays out for participants, noting particular scenarios that she thinks will be good for a longer discussion with her class. Furthermore, she sees an opportunity to update some content that will align the game with more recent developments in corporate relations. She adjusts some text, but then also changes out visual assets like logos from the asset repository as well as adjusting elements of game characters' ethnography to be more representative of the company. She edits some of the text and saves the adjusted game.

B. Requirements

The scenarios have been refined into requirements. The core functional requirements are capabilities to:

- Browse the organization of the game.
  - Act – the largest organizational unit of the game, in which the parameters of the game context, characters and learning objectives are all contained.
  - Scene – a collection of screens that establish the narrative arc and pedagogical goals for the player.
  - Screen – an individual page that contains information for the player and may or may not require player input.
  - Challenge – the exercise by which a player demonstrates proficiency towards a learning objective.
- Browse game learning objectives at any point of game organization, (i.e. Act, Scene, Screen, and Challenge).
- Preview the content of the game (Screen and Challenge).
- Add and edit the text of speech bubbles, hints, and feedback.
- Add, Move, Delete, and Resize game elements (characters, props) in the game (Screen).
- Change the backdrop (Scene).
- Toggle the display of hints and sound effects associated with visual elements (Screen and Challenge).
- Add, preview, and change sound effects (Screen).
- Add, preview, and change background audio (Scene).
- Save changes to the current or new game.
- Detect, report, and recommend corrections to errors in the game.

C. Interface Design

A screenshot of the interface design is provided in Fig. 2. As one part of the Game Generation module (Fig. 1), the preview and visualization capabilities are provided under the

Preview tab with an **Interactive Display** area, **Browser**, and **Toolbar**. A main **Menu** provides common features, such as file management (load, save, save as, and error checking).

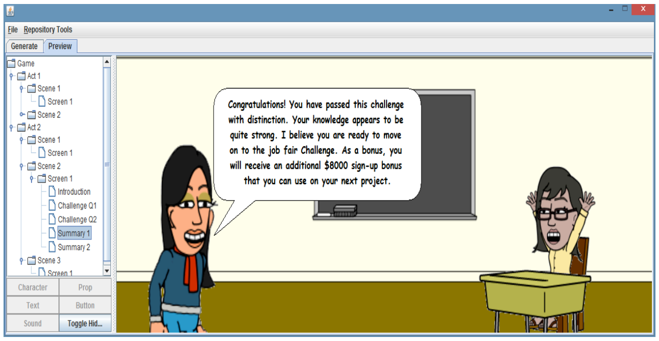

File Repository Tools
Generate Preview
Game
Act 1
Scene 1
Screen 1
Scene 2
Act 2
Scene 1
Screen 1
Scene 2
Screen 1
Introduction
Challenge Q1
Challenge Q2
Summary 1
Summary 2
Scene 3
Screen 1
Character Prop
Text Button
Sound Toggle Hid...
Congratulations! You have passed this challenge with distinction. Your knowledge appears to be quite strong. I believe you are ready to move on to the job fair Challenge. As a bonus, you will receive an additional $8000 sign-up bonus that you can use on your next project.

Fig. 2 Interface Design: Menu, Browser, Toolbar, and Interactive Display areas.

Using the **Toolbar**, in the lower left of the interface, a designer can add a character, a prop (e.g., furniture, coffee cup), or a user interface button. He or she can choose to create an information box, hint, conversation bubble, or replace the backdrop for a Scene. The toolbar allows one to show/hide the icons that indicate a sound effect is present for a game element, the learning objectives associated with a Scene or Challenge, or any hints that are present. For example, clicking the "Character" button in the Toolbar opens the Character Select Window, where the user can browse through a collection of character options and add one to a Screen or Challenge.

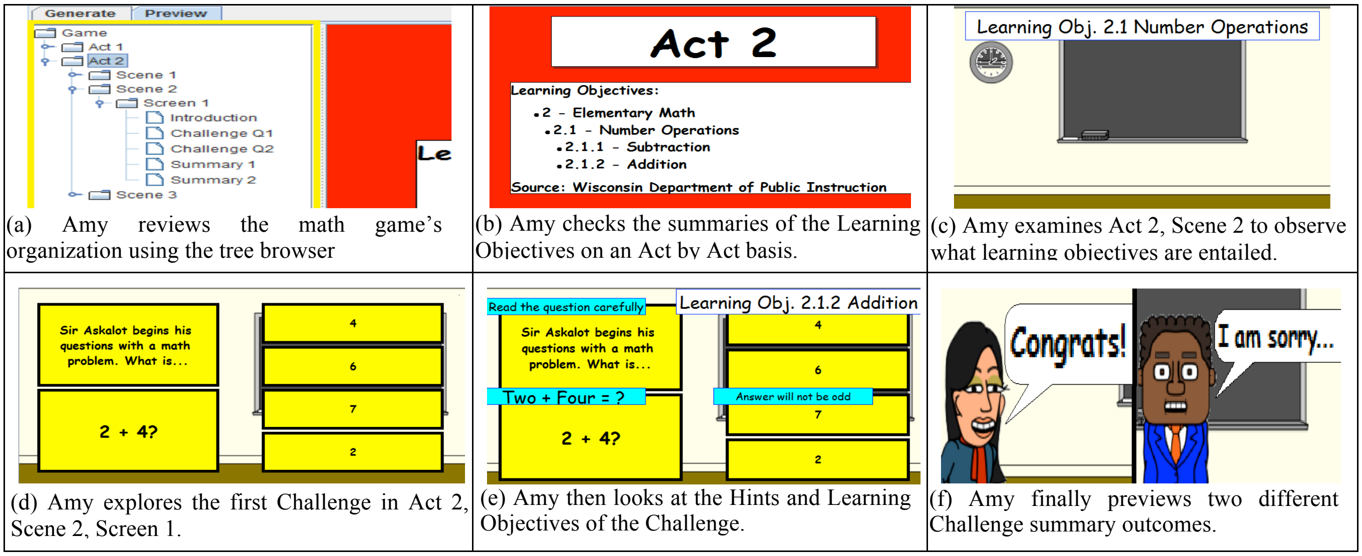
To guide the designer, access to the toolbar buttons changes with the level of game currently being viewed. For example, in a Scene, the designer can edit the background. At the Screen level, however, changing the background is not an option, so that button is disabled and greyed out. All buttons offer a similar selection wizard with varying categories for each asset type.

In the **Interactive Display** area, on the right side of the interface, a designer can use the mouse to select a game element (character, props) and move or resize it; these same selected elements can also be deleted using the delete key. A designer can edit the text of generic interaction elements, including speech bubbles, information boxes, and game hints. A designer can also add, delete, play sound effects. The browser is on the upper left of the interface. The designer can use the **Browser** to conveniently and quickly explore the game. The tree display allows the user to navigate between the Game, organized into Acts, Scenes, Screens, and Challenges. Each Act displays its title along with Learning Objectives that are included in the underlying Scenes/Screens. The challenge screen displays questions as the player would see them, as well as the view revealing hints and the intended learning objectives.

Using the main **Menu**, the designer can manage files (load, save, save as) and access the error checking feature. The game error checker runs following either loading a game, or clicking the "Check XML Errors" menu item. This feature checks the game for 63 kinds of errors and presents the results in the Interactive Display Panel. Each game error consists of a classification, a textual description, and its location in the game structure. Depending on its classification, an "Autocorrect" button may appear enabled to the right of the error text. Clicking this button saves the user from manually editing XML to make corrections.

IV. Validation

The preliminary validation of the SEG$_{\text{Mod}}$ approach includes a prototype tool, which has undergone substantial feature and performance testing. Here, we provide a summary of the validation effort using Scenario 1 (Amy the 4$^{\text{th}}$ grade teacher, Section I). The results are presented in two parts, in order to simplify the presentation. The first part focuses on interactions to explore the learning objectives (Fig. 3); the second to adjust some of the play aesthetics (Fig. 4). Both parts begin after the designer opens a game and is shown a splash screen with a high level summary of any errors found checking it (Fig. 2); the designer is offered the "autocorrect" option for errors that can be handled by a default operation. These illustrations are



(a) Amy reviews the math game's organization using the tree browser

(b) Amy checks the summaries of the Learning Objectives on an Act by Act basis.

(c) Amy examines Act 2, Scene 2 to observe what learning objectives are entailed.

(d) Amy explores the first Challenge in Act 2, Scene 2, Screen 1.

(e) Amy then looks at the Hints and Learning Objectives of the Challenge.

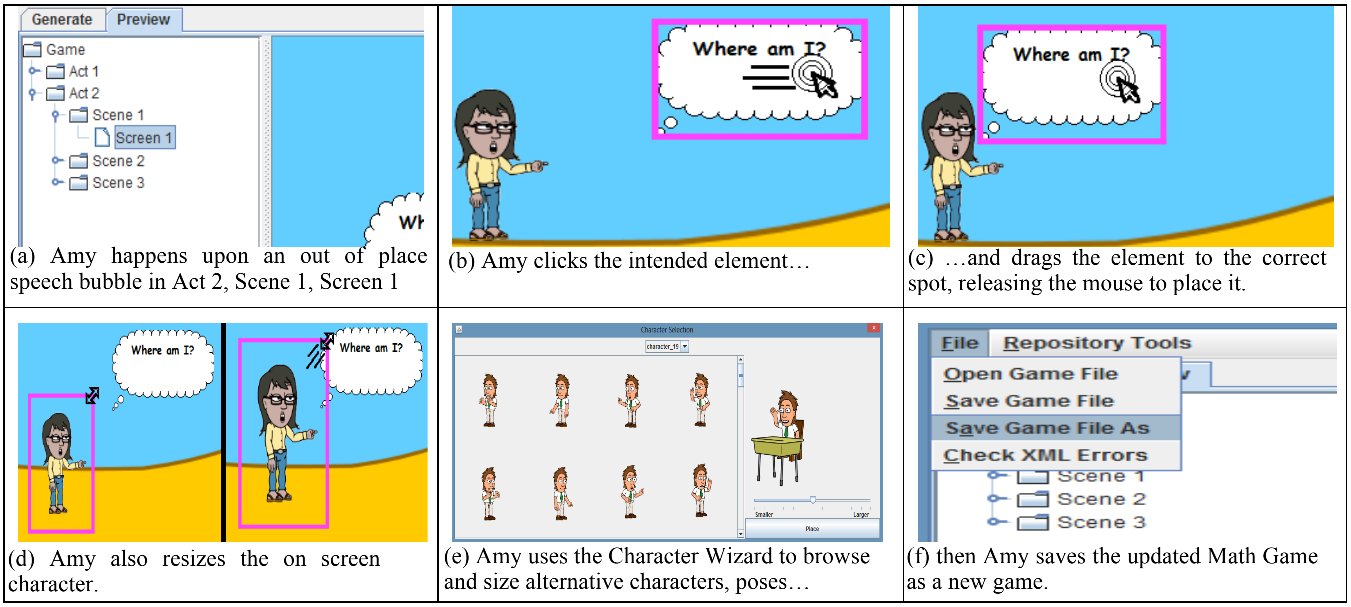
(f) Amy finally previews two different Challenge summary outcomes.

Fig. 3 Scenario 1 Amy Explores the Game's Learning Objectives

not exhaustive; they are intended to provide a sense of the approach and the prototype tool.

A. *Learning Objectives Focus*

In Fig. 3, Amy explores the overall organization of the game using the Browser (Fig. 3a), then progresses to explore the learning objectives covered in Act 2 (Fig. 3b) and scene level (Fig. 3c). She then decides to take a look at the first challenge (Fig. 3d), to see the description, question, and answer options. Amy then takes a more detailed look checking the hints and the learning objectives for the challenge, by hovering the mouse over the challenge in the Interactive Display area. (Fig. 3e), followed by the feedback options presented to the students when they are correct or incorrect (Fig. 3f).



(a) Amy happens upon an out of place speech bubble in Act 2, Scene 1, Screen 1

(b) Amy clicks the intended element…

(c) …and drags the element to the correct spot, releasing the mouse to place it.

(d) Amy also resizes the on screen character.

(e) Amy uses the Character Wizard to browse and size alternative characters, poses…

(f) then Amy saves the updated Math Game as a new game.

Fig. 4 Scenario 1 Amy Explores the Aesthetics and Traditional Game Content

B. *Traditional Aesthetics and Game Content Focus*

In Fig. 4, Amy explores the overall organization of the game using the Browser Fig. 4(a) and notices a speech bubble looks out of place. She mouse selects the speech bubble in the Interactive Display area and drags it closer to the character (Fig. 4b and Fig. 4c). Amy prefers the character to be bigger, so she mouse selects the character and resizes it in the Interactive Display area by dragging the corner of the image (Fig. 4d). She then decides to take a look at alternative character s and poses by selecting the Character button in the toolbar. A wizard is presented, where she has the option to select and add a new element (Fig. 4e). She decides to stay with the current character by cancelling; then continues on to save the work as a new game file (Fig. 4f).

V. Related Work

Recent research into the efficacy of SEGs has ranged across the spectrum of applications from corporate training [2] to human anatomy courses [19]; in some cases this prior work has also evaluated corresponding tools for SEG development in teaching non-computing students relevant concepts, or in using SEGs to monitor student learning [6]. In many contexts, SEGs have been shown to sustain engaged learning [12], and improve users' feelings of competence [24].

Tools such as ARGILE [10] can help domain experts to collaboratively develop and annotate the rules of knowledge-intensive SEGs. Recent Domain Specific Visual Languages (DSVL) have proven useful in programming abstract tasks ranging from algorithmic behavioral specifications (Vibes [14]) to home automation design (Pantagruel [9]). SimSYS combines advantages from both of these bodies of work to allow experts in other domains (besides SEG development) to work with an SEG in a visual programming language.

A wide variety of development environments for SEGs have been proposed [3][8][16][21][25], which focus on serious computer science education through the process of developing games. The eAdventures Project [20] has produced a DSVL for developing serious educational games in other fields of study. This is closest in spirit to our own work, targeting educators and domain-experts as primary users of the tool. Aspects of eAdventures that we consider to be predecessors of our own tool include the chapter/section structure for previewing story flow, the use of hints as game elements, and the ability to transform the visual description automatically into a playable game.

SimSYS differs from eAdventures in the incorporation of learning outcomes as an integral part of the game structure, rather than as optional annotations. The SimSYS tool generates and previews game variants based upon an existing, defined knowledge base with explicit learning outcomes and a repository of applicable game elements, rather than focusing on authoring complex narrative control flow choices from scratch. Our goal from the outset is to equip an educational tool with a DSVL, as opposed to making a game design tool become educational. As a result, assessment is more explicit, and learning standards are more easily tracked on an institutional basis.

VI. Conclusions and Future Work

In order to take advantage of the many instructional benefits of SEGs, instructors require powerful, flexible solutions that can help them to effectively navigate the complexity of game development. This then frees them to concentrate on the pedagogical goals of the game. The Game Generation module, part of the SimSYS Game Development Platform, aims to be such a solution, providing a collection of approaches to generate and preview SEGs. Unlike other tools, our system begins with an educational foundation, and helps the instructor to build a high-quality game upon it.

In this paper, we present $\mathrm{SEG}_{\mathrm{Mod},}$ a lightweight approach for interactively modding SEGs that allows a domain expert (but novice game designer) to explore, visualize, listen to, and tune specific elements of a game. For example, the designer can browse the existing organization and learning objectives of a game, in effect its infrastructure, but cannot modify them as this requires substantial expertise to maintain the consistency of the game. They can, however, modify the aesthetics and traditional game play elements, such as changing the backdrop or background music for a scene; add, modify, or delete the display of characters, props and their sound effects. It is not intended to be a broad, comprehensive solution that supports modifying every aspect of the game design. Instead, to keep it simple and easy to use, we have proposed an approach with a relatively modest subset of functionality.

A prototype tool has been developed as part of the validation work; its features and performance have been tested and the results are encouraging. A comprehensive usability study is needed to fully to evaluate the set of functionality proposed, quantifying its strengths and limitations.

References

[1] P. Airasian, K. Cruikshank, R. Mayer, P. Pintrich, J. Raths, M. Wittrock, *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives,* 2000. L. Anderson and D. Krathwohl, eds., Allyn and Bacon.

[2] A. Azadegan, J. C. K. H. Riedel, and J. Baalsrud Hauge. Serious games adoption in corporate training. In *Proceedings of the Third International Conference on Serious Games Development and Applications*, SGDA'12, pages 74-85, Berlin, Heidelberg, 2012. Springer-Verlag.

[3] V. E. Bennett, K. H. Koh, and A. Repenning. CS education re-kindles creativity in public schools. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ITiCSE '11, pages 183-187, 2011. ACM.

[4] K. Cooper and C. Longstreet, *Towards Model-driven Game Engineering for Serious Educational Games: Tailored Use Cases for Game Requirements*, in Proceedings of the IEEE 17th International Conference on Computer Games, 2012, pp. 208-212.

[5] K. Cooper, E. Nasr, E., and C. Longstreet, "Towards Model-Driven Requirements Engineering for Serious Educational Games: Informal, Semi-formal, and Formal Models". In *Proceedings of the 20th International Working Conference on Requirements Engineering: Foundation for Software Quality,* 2014, pp. 17-22.

[6] S. Cuomo, V. Fuccella, and A. Murano. Full formative assessment based on educational video games. In *Proceedings of DMS 2010: 16th International Conference on Distributed Multimedia Systems*, pages 228-231. Knowledge Systems Institute, 2010.

[7] R. Daconceicao, C. Locke, S. Longstreet, and K. Cooper, "Semi-automated Serious Educational Game Generation: A Component-based Game Engineering Approach", in Proc. of IEEE 18th Int. Conference on Computer Games, 2013, pp. 222-227.

[8] S. Dekhane and X. Xu. Engaging students in computing using GameSalad: A pilot study. *J. Comput. Sci. Coll.*, 28(2):117-123, Dec. 2012.

[9] Z. Drey and C. Consel. Taxonomy-driven prototyping of home automation applications: A novice-programmer visual language and its evaluation. *J. Vis. Lang. Comput.*, 23(6):311-326, 2012.

[10] N. El Mawas and J.-P. Cahier. Towards participative and knowledge-intensive serious games. In *Proceedings of the Third International Conference on Serious Games Development and Applications*, SGDA'12, pages 86-97, Berlin, Heidelberg, 2012. Springer-Verlag.

[11] R. Garris, R. Ahlers, and J. Driskell. Games, motivation, and learning: A research and practice model. *Simulation & Gaming*, 33(4): 441-467, 2002.

[12] J. P. Gee. What Video Games Have to Teach Us About Learning and Literacy. Palgrave Macmillan, 2nd edition, 2007.

[13] V. Guillén-Nieto and M. Aleson-Carbonell. Serious games and learning effectiveness: The case of *It's a Deal!*. *Computers and Education* 58: 435-448, 2012.

[14] G. Güleşir, L. Bergmans, M. Akşit, and K. van den Berg. Vibes: A visual language for specifying behavioral requirements of algorithms. *J. Vis. Lang. Comput.*, 24(5):350-364, 2013.

[15] IEEE Computer Society, SWEBOK 3.0 *Guide to the Software Engineering Body of Knowledge* V3.0, 2014, P. Bourque and R. Fairley eds., available at: http://www.computer.org/portal/web/swebok.

[16] Ioannidou, A. Repenning, and D. C. Webb. AgentCubes: Incremental 3D end-user development. *J. Vis. Lang. Comput.*, 20(4):236-251, Aug. 2009.

[17] H. Jayaraj, R. Raiker, K. Cooper, and C. Longstreet, *SimSYS Game Specification*, Technical Report UTDCS-06-14, The University of Texas at Dallas, 2014.

[18] C. Longstreet and K. Cooper, *A meta-model for developing simulation games in higher education and professional development training,* in Proceedings of the IEEE 17th International Conference on Computer Games, 2012, pp. 39-44.

[19] M. Ma, K. Bale, and P. Rea. Constructionist learning in anatomy education: What anatomy students can learn through serious games. In *Proceedings of the Third International Conference on Serious Games Development and Applications*, SGDA'12, pages 43-58, Berlin, Heidelberg, 2012. Springer-Verlag.

[20] E. J. Marchiori, Á. del Blanco, J. Torrente, I. Martinez-Ortiz, and B. Fernández-Manjón. A visual language for the creation of narrative educational games. *J. Vis. Lang. Comput.*, 22(6):443-452, 2011.

[21] M. Overmars. Teaching computer science through game design. *Computer*, 37(4):81-83, Apr. 2004.

[22] M. Papastergiou. Digital-Based Learning in High School Computer Science Education: impact on educational effectiveness and student motivation. *Computers and Education* 52: 1-12, 2009.

[23] SimSYS Game Development Platform for Serious Educational Games, project homepage www.utdallas.edu/~kcooper.

[24] E. D. Van Der Spek. Towards designing for competence and engagement in serious games. In *Proceedings of the Third International Conference on Serious Games Development and Applications*, SGDA'12, pages 98-109, Berlin, Heidelberg, 2012. Springer-Verlag.

[25] T. Wahner, M. Kartheuser, S. Sigl, J. Nolte, and A. Hoppe. Logical thinking by play using the example of the game "space goats". In *Proceedings of the Third International Conference on Serious Games Development and Applications*, SGDA'12, pages 174-182, Berlin, Heidelberg, 2012. Springer-Verlag.

[26] *Common Core State Standards for Mathematics* Wisconsin Department of Public Instruction, Madison, WI, 2011.

[27] World Wide Web Consortium (W3C), XML Schema Definition Language (XSD) 1.1 Part 1: Structures, April 2012, available at: http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405.