Integrating learning analytics into a game authoring tool

Ivan J. Perez-Colado, Victor M. Perez-Colado, Manuel Freire-Moran, Ivan Martinez-Ortiz, Baltasar Fernandez-Manjon

Dept. of Software Engineering and Artificial Intelligence, Facultad de Informática, Universidad Complutense de Madrid Madrid, Spain
{ivanjper, victormp}@ucm.es, {manuel.freire, imartinez, balta}@fdi.ucm.es

Abstract. Educational games can greatly benefit from integrating support for learning analytics. Game authoring tools that make this integration as easy as possible are therefore an important step towards improving adoption of educational games. We describe the process of integrating full support for game learning analytics into uAdventure, a serious game authoring tool. We argue that such integrations greatly systematize, simplify and reduce both the cost and the knowledge required to apply analytics to serious games. In uAdventure, we have used an analytics model for serious games and its supporting implementation as a xAPI application. We describe how player interactions are automatically traced, and provide an interaction-model-trace table with the general game traces that are generated by the editor. Also, we describe the custom editors that simplify the task of authoring game-dependant analytics. Thanks to these integrated analytics, games developed with uAdventure provide detailed tracking information that can be sent to a cloud analytics server, to be analyzed and visualized with dashboards that provide game developers and educators with insights into how games are being played.

Keywords: game learning analytics, analytics, serious games, games authoring, xAPI

1. Introduction

Game analytics (GA, also called telemetry) is the process of collecting and analyzing videogame user interactions to generate a better insight of the game experience for game designers and developers to take decisions in the next project iterations [1]. For example, such an analysis can reveal which levels are too hard for the average user, or provide insights on how to increase monetization. Similarly, learning analytics (LA) is the analysis of user's interactions with educational purposes [2], for instance providing information that allows educators to better understand how the learners are applying domain knowledge in an e-learning system, and improve the educational experience in some way (e.g. evaluate student progress). Game learning analytics (GLA) is the combination of both GA and LA to allow both game designers and educators to analyze player/learner interactions to improve the use of the games in education [3].

Although there are many platforms that provide both LA and GA, however GLA is still a complex process and there is no generally-accepted approach to apply it to serious games. For example, games must typically provide data to each analytics system (e.g. GA, LA) separately, and frequently in different formats. In addition, once the games are deployed, the GLA results are only available through each specific analytics system's proprietary analysis, reports and dashboards. As of this writing, GLA is a complex ad-hoc process specific for each game and each analytics system. However, we believe that GLA should play a critical role in the lifecycle of serious games (see Fig. 1), as it is key to allow both game and learning designers to validate their designs; and can also be used to provide formative and summative evaluation.

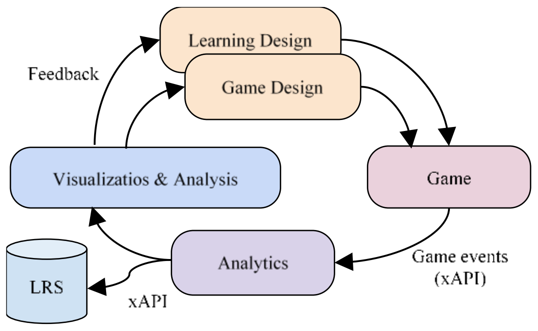


Fig. 1: Use of Game Learning Analytics in a serious game lifecycle. Use of xAPI provides a standards-based format for analysis and archival.

The process of applying GLA to educational games would be greatly simplified by having in-built support within the authoring tools, so that the resulting games can communicate with analytics services using well-known standards. This integration would systematize and simplify the usage of analytics by automatically linking the game model with the analytics model, reducing both the cost and the know-how barriers that have hampered GLA adoption in small deployments. In the following sections, we first briefly introduce uAdventure, a game authoring tool into which we have integrated GLA support. We then describe the analytics model and the user-interfaces that determine and configure how user gameplay is mapped into analytics traces. Finally, we present a discussion and conclusions that summarize the main results of this work, together with future directions for improvement.

2. An Analytics model for uAdventure

uAdventure (uA) is a serious game authoring tool built on top of the Unity game authoring platform that supports the full development lifecycle for adventure "point and click" games [4]. It is a reimplementation of eAdventure (eA) [5], which was built using Java. The use of the Unity platform allows uA games to support a much larger range of devices and platforms. The goal of uA is to allow non-expert developers to create “point and click” educational games, including features such as scenarios, characters, dialogs, and assessment. While in eA the assessment system was based on e-learning models (e.g. SCORM), uA now extends it with support for GLA.

uA GLA support is based on a general game analytics model that is implemented according to the emergent specification promoted by ADL and called xAPI (eXperience API). A game analytics model describes how in-game interactions are reported to an analytics server, typically as a stream of events, but also sometimes as fully serialized game-states. The serious games xAPI profile [6] (SG-xAPI for short) is a general event-based analytics model that builds on the xAPI activity stream standard [7], and is therefore event-based. The main event categories found in SG-xAPI, strongly inspired by [8], are

- Completables, which describe progress along a particular level or task with a start and an end. Completables can be nested. Examples could be game, game session, level, quest, or race.
- Alternatives, which reflect in-game choices made by the player. Menus, questions, paths and dialogue choices are examples of alternatives.
- Meaningful variables, which directly echo in-game state changes such as score or character health. Note that these variables are a very small subset of the total game-state.
- Custom interactions, intended as extension points for actions not covered in the above categories.

xAPI traces are composed of a subject, a verb, an object, and optional context; for example, an xAPI trace describing that “player Alice used a key” in a given game would include Alices’ identity as subject, “Used” as xAPI verb, the specific in-game key identifier as an object, and the specific door as context. This output is then passed on into existing GLA services, such as the open-source RAGE Analytics[1] system which is used in the H2020 RAGE and BEACONING projects.

Integrating GLA into a game requires both the analytics model and the definition of the server-side analyses that process this data into a format suitable for visualization within dashboards [3]. However, since uA relies on the SG-xAPI default analyses and visualizations performed in RAGE Analytics and described in [9], this work focuses exclusively on the mapping between in-game actions and the SG-xAPI traces that are sent to the server.

Point and click games are typically composed of multiple scenes woven together via a supporting narrative, where players interact on scene items or characters via mouse clicks to advance the plot. The following subsections explore the areas in which events have been analyzed and traced, starting from the lowest to the highest level of abstraction. The following subsections describe i) session management events; ii) user interactions with game elements inside the scenes; iii) scene changes; iv) meaningful variable changes; and finally, v) completables, which are high-level tasks related to in-game progress.

[1] https://github.com/e-ucm/rage-analytics

2.1. Session Events

In order to process traces sent from games into sessions, the analytics servers need to identify players. Three options are available: fully logged-in players, pseudonymous players, and fully anonymous players. When players are fully logged in, the analytics system can be integrated with an LMS to provide detailed evaluation that is tied to the player's true identity [10]. This, however, presents certain privacy and confidentiality concerns, and may not be feasible in an analytics-as-a-service setting. With pseudonymous login, players are assigned a random pseudonym when they first connect, which they will continue to use in later sessions. This allows evaluation to be carried out, without the possibility of tracing the actual identities of the players. Finally, in a fully anonymous setting, all sessions are independent of each other, and no evaluation is possible. This option is still interesting during initial pilots to gather feedback to improve learning and game design (as shown in Figure 1).

While necessary for analytics, authentication and authorization are not part of the analytics model itself, as they are expected to be handled by the GLA servers before any actual gameplay data is sent. uAdventure supports login (for non-anonymous use) by displaying a small login form on game startup, and transparently handles session startup with the GLA servers specified during game development.

2.2. User Interaction Events

Raw input events such as mouse movements, clicks, and keyboard inputs constitute the lowest level of abstraction in analytics. These events provide little information on their own, even when grouped to create higher-level abstractions such as dragging or double-clicking. For GLA purposes, interactions that are directly linked to game content and/or game progress, such as interactions with scene elements, are much more valuable. Elements inside scenes include items, characters or areas; and players can typically interact with them with in-game actions such as "examine" or "use".

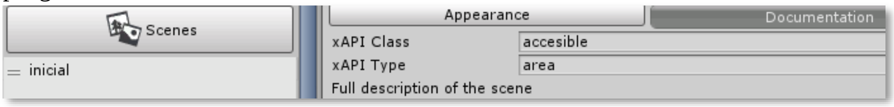
uAdventure automatically sends "Interacted" SG-xAPI traces whenever a player interacts with an in-game element such as an item or character. The trace will identify both the object of the interaction and, if applicable, the type of interaction (such as "examine" or "eat"). Therefore, tracing such low-level interactions requires zero effort on the part of the game developer, and provides a free stream of information to perform GA (e.g. interaction heat maps). On the other hand, these events are of little help to better understand the in-game learning process; using Completables (see section 2.6), which do require a certain configuration, yields much better results.

2.3. Scene and Cutscene changes

Scenes in uAdventure are modeled after their theater counterparts: they provide backdrops within which the player moves and interacts with items and other characters. Certain mini-games are also modeled as scenes. Cutscenes are special scenes that usually imply the visualization of non-interactive content, such as slides or videos. Tracking scene changes and whether cutscenes are skipped or not provides valuable context to understand other events that are being reported by a game. Additionally, the game always knows which scene is currently open, so that information regarding both is readily available.

uAdventure uses the SG-xAPI "Accessed" verb to track scene changes, and also allows developers to further describe the scene for analytics purposes (choices include Screen, Zone, Area and Cutscene). Skipping cutscenes, such as videos, might reduce the learning value, as the player has only been exposed to a part of the potentially informative content. uAdventure uses "Skipped" verbs to signal this fact.

While access events are valuable for GA, they are not (with the exception of Skipped) so useful for LA: game developers can freely connect scenes, and there is no game-independent way to determine the game progress from these changes [11]. To infer progress, game developers can specify that scenes should be treated, in addition to Accessibles, as Alternatives (see section 2.5), where by entering or leaving the scene the player is making a choice, or as a Completable (see section 2.6) task such as a minigame, where entering or exiting the scene should be interpreted as actual progress.



xAPI Class: accesible
xAPI Type: area
Full description of the scene

Fig. 1. Editor shows scene documentation where game designer can change scene specification, default values are, as shown: class Accessible with type Area

2.4. Meaningful Variables

Game state in uA is represented mainly by variables and flags. uA condition system uses their values to control content display and behavior execution. Therefore, changes in variables are usually good candidates for analytics.

However, a game-specific analysis is required to identify the relevance of a variable (or a set of variables) for the learning process. Even if not all variables are relevant to LA, to reduce game authoring complexity, all changes in variables and flags are automatically reported. Once the data is collected, it can be mined to distinguish the variables with a larger impact on the learning process.

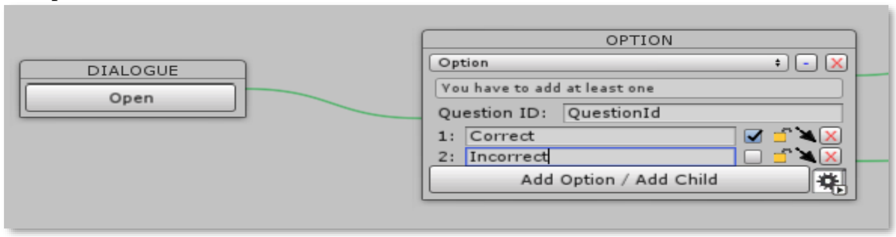
When a player interaction triggers several in-game consequences, so, to preserve the change context (which would allow for a more precise analysis), it is better to associate all of them together as the result of an interaction event. This is possible as, instead of sending an event per variable change, all the variable changes are stacked waiting for the next non-variable event. When the next interaction occurs, the trace generated will carry all the extensions inside.

2.5. Alternatives

Alternative selections represent a higher level of abstraction, and provide a direct measure of the student’s knowledge. An alternative can be represented in many ways (e.g. text response selection, image selection, path selection); but in general, constitutes a choice where some options can be considered correct and others less so. Therefore, alternative selection is very valuable for LA and assessment, as well as to test and verify game design.

The most common types of alternatives in point-and-click games are dialogue responses for use with non-player characters [11]. uA provides a graph-based dialogue editor where questions can be identified with a unique id and, using a checkbox, marked as correct or incorrect (see Fig. 3). In uA questions without a question-id are considered not relevant for analytics purposes.

To track these events, the SG-xAPI “Selected” verb is used, where question types include menu, path or question. A result field describes the selected option identifier as response and the whether it was correct or not.


DIALOGUE

Open

OPTION

Option

You have to add at least one

Question ID: QuestionId

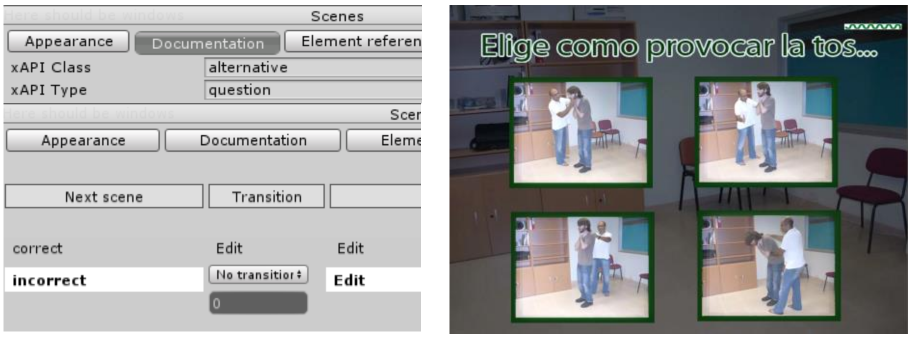
1: Correct ✓

2: Incorrect

Add Option / Add Child

Fig. 2. Dialog Question editor allowing a game designer to specify a question id and which of the answers of that question should be considered correct or incorrect (Checkbox)

In addition to dialog responses, as mentioned in subsection 2.3, it is also possible to interpret scene selection as type of Alternative. In this case, as depicted in Fig. 4, each available scene exit is considered a response using the arriving scene name as response identifier. If the player passes through the exit successfully, the answer was correct, but otherwise, it was a wrong answer.


Scenes

Appearance | Documentation | Element referen

xAPI Class: alternative

xAPI Type: question

Scer

Appearance | Documentation | Eleme

Next scene | Transition

correct | Edit | Edit

incorrect | No transition | Edit

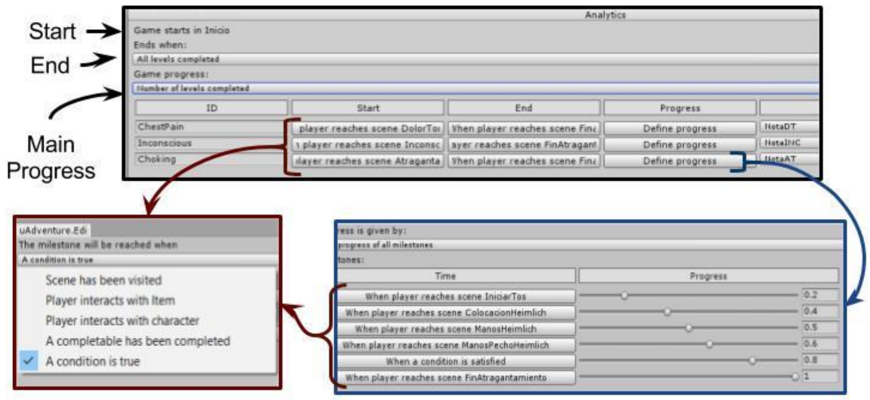
0

Fig. 3. Left: An Alternative scene type Question with correct and incorrect exits. Right: runtime Alternative Type Question scene from the SG First Aid Game.

2.6. Completables

Completables provide the highest level of abstraction and describe player progress along the abstract tasks that the user has to complete throughout the game. Tasks are not limited to representing game progress: they can be used to represent learning process itself, for instance by establishing a correspondence between progress and score in a given task (game design) and learning the concepts that it exercises (learning design). Completables are available in the uA editor's Analytics tab. To track completables, SG-xAPI includes the verbs "Started", "Progressed" and "Finished".

Inside of a game there could be multiple completables and they may be active at any time, even in parallel. A game with correctly configured completables is much easier to analyze than one where the underlying triggers have to be decoded from the underlying game traces (in a server-side game-specific analysis). In order to track them, the completable system lets the developer freely configure starting milestone and ending milestones. These milestones can be determined by five different triggers: i) access to specific scene, ii) interaction with specific item, iii) interaction with specific character, iv) accomplishment of another completable and v) a boolean condition as determined by a combination of in-game variables.



Analytics

Game starts in Inicio

Ends when:

All levels completed

Game progress:

Number of levels completed

ID	Start	End	Progress	
ChestPain	player reaches scene DolorTo	Vhen player reaches scene Fin	Define progress	NotaDT
Inconscious	player reaches scene Inconsc	ayer reaches scene FinAtragant	Define progress	NotaINC
Choking	layer reaches scene Atraganta	Vhen player reaches scene Fin	Define progress	NotaAT

uAdventure.Edi

The milestone will be reached when

A condition is true

- Scene has been visited
- Player interacts with Item
- Player interacts with character
- A completable has been completed
- ✓ A condition is true

ess is given by:

progress of all milestones

tones:

Time	Progress	
When player reaches scene IniciarTos		0.2
When player reaches scene ColocacionHeimlich		0.4
When player reaches scene ManosHeimlich		0.5
When player reaches scene ManosPechoHeimlich		0.6
When a condition is satisfied		0.8
When player reaches scene FinAtragantamiento		1

Fig. 4. uA Progress editor. i) Top shows main completable editor, defining three cases of the serious game FirstAidGame. Progress is defined by the number of cases completed. ii) Right shows individual completable editors, with a list of milestones to be satisfied. Progress is determined by the max of the milestones reached. iii) Left shows milestone editor options, letting game designer to choose between a list of options to define what reach this milestone

Completables can be composed of several milestones; the accomplishment of each milestone defines the progress. To define them, the uA editor provides a progress editor that lets the user specify all milestones that compose the completable. Progress can either be calculated as the ratio of completed-to-total milestones, or configured for each specific completable using sliders as illustrated in Fig. 5. Finally, to associate the score, the user can select the variable that will hold its value.

In addition to the level-specific completables, there is a global game completable, identified using the "Game" type. When the game starts, it generates a "Started" event; whenever a completable is completed a "Progressed" trace is launched tracking the game general progress; finally, the condition associated with game end will generate the "Completed" event with the average score of all completables.

3 Mapping game events to SG-xAPI

Table 1 contains a summary of the mapping from game events to xAPI traces presented in the previous subsections, in increasing order of abstraction; and therefore, utility from a LA point of view: events closer to the interaction with game elements are less meaningful than the ones that are connected to game progression, such as alternatives and completables.

Event	Cause	xAPI Type xAPI Verb	Target	Result R: response, S: success, Ext: extensions
NPC Interaction	Player opens NPC actions menu	Character Interacted	NPC name	Ext: Action name
Item Interaction	Player opens item actions menu	Item Interacted	Item name	Ext: Action name
Scene access	Player enters a scene	Accessible Accessed	Scene id	
Cutscene start	Player starts a cutscene	Cutscene Accessed	Cutscene id	
Cutscene skip	Player presses skip	Cutscene Skipped	Cutscene id	Ext: Percent watched
Exit selection in alternative type scenes	Player selects an exit in current scene, for menus or visual choices	(Alternative, Question Menu or Path) Selected	Exit Id	R: Arriving scene S: Based on exit conditions
Dialog choice	Player selects one dialog option	Alternative Selected	Question Id	R: Response S: Correctness
Task start	Player reaches a milestone	Completable Started	Completable Id	
Task progress	Player reaches one of the milestones	Completable Progressed	Completable Id	Ext: Milestone progress value
Task finish	Player reaches a milestone or completes all the steps	Completable Completed	Completable Id	R: Score from variable S: Based on conditions Ext: Time
Game start	Player visits title	Game Started	Game name	
Game progress	Accomplishment of any of the levels	Game Progressed	Game name	Ext: Progress as percent of levels (tasks) completed
Game end	Milestone or all levels completed	Game Completed	Game name	R: Avg. score of all levels S: Based on conditions Ext: Time

Table 1. Summary of game events traced inside the uAadventure authoring tool

4 Discussion and Conclusions

Serious games have frequently been evaluated by relying exclusively on (paper-based) learning assessment. In traditional e-learning, different methods have been used to connect the serious game outcome (e.g. score) to different learning systems allowing the use of serious games as any other assessment tool such as online tests; but this integration focuses on simple outcomes and provide few or no insights into the process. However, if a game fails to work for a particular set of players, knowing the exact steps they have followed inside the game is crucial to determine the reasons. Tackling this lack of flexibility requires a transformation from the traditional assessment model to the new evidence-driven model; and mapping in-game actions to events that can be used to analyze and gain insight not only on the results, but also on the process.

The uAdventure analytics model describes how player interactions are automatically captured and transformed into events that the LA system can collect and analyze. The events generated cover a wide range of situations such as scene access, element interaction and options selection. We also describe how higher-level events can be authored from within uA. Other tools that wish to integrate GLA into their games should find this analysis useful when developing their own analytics models and user interfaces.

The integration of uAdventure with GLA presents multiple benefits to game developers. First, all games developed with the tool automatically integrate free support for LA requiring minimum developer effort. This greatly reduce the GLA cost. In addition, thanks to the use of standardized events, developers do not need handle event encoding, and can automatically take advantage of several generic analyses and visualizations in xAPI-aware GLA platforms such as the (open-source and freely available) RAGE Analytics. For no added effort, uAdventure developers and users can enjoy basic analytics and assessment information, which can then be enriched by generating game-dependent events that provide richer information that links to the relevant learning situations identified in the game learning design.

During the integration, we have also identified multiple areas that can benefit from further work. For example, changes to game variables are only sent to the server as a part of future non-variable updates, instead of when they actually occur. This is part of the standard, but may confuse analyses that expect traces to be sent in the order they were generated. The uAdventure editor can also benefit from numerous usability enhancements, such as displaying information about completables directly inside of the element editors that are used as triggers in the completables' definition; or integrating game mechanics such as quests or missions that are directly tied to completables and provide explicit in-game feedback to allow users to track their own progress.

We believe that the integration between game authoring tools and game learning analytics, as described in the present work, is an is an important step towards wider usage of GLA in serious games.

Acknowledgement

This work has been partially funded by Regional Government of Madrid (eMadrid S2013/ICE-2715), by the Ministry of Education (TIN2013-46149-C2-1-R) and by the European Commission (RAGE H2020-ICT-2014-1-644187, BEACONING H2020-ICT-2015-687676).

References

[1] M. Seif El-Nasr, A. Drachen, and A. Canossa, *Game Analytics*. London: Springer London, 2013.

[2] T. Elias, "Learning Analytics : Definitions , Processes and Potential," *Learning*, vol. 23, pp. 134–148, 2011.

[3] M. Freire, Á. Serrano-Laguna, B. M. Iglesias, I. Martínez-Ortiz, P. Moreno-Ger, and B. Fernández-Manjón, "Game Learning Analytics: Learning Analytics for Serious Games," in *Learning, Design, and Technology*, Cham: Springer International Publishing, 2016, pp. 1–29.

[4] I. Perez Colado, V. Perez Colado, I. Martínez-Ortiz, M. Freire, and B. Fernandez-Manjon, "uAdventure: The eAdventure reboot - Combining the experience of commercial gaming tools and tailored educational tools," *IEEE Glob. Eng. Educ. Conf.*, no. April, pp. 1754–1761, 2017.

[5] J. Torrente, A. del Blanco, E. J. Marchiori, P. Moreno-Ger, and B. Fernandez-Manjon, "<e-Adventure>: Introducing educational games in the learning process," in *IEEE EDUCON 2010 Conference*, 2010, pp. 1121–1126.

[6] A. Serrano-Laguna, I. Martinez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernandez-Manjon, "Applying standards to systematize learning analytics in serious games," *Comput. Stand. Interfaces*, vol. 50, no. September, pp. 116–123, 2016.

[7] "xAPI - ADL Net @ www.adlnet.gov." .

[8] Á. Serrano-laguna, J. Torrente, P. Moreno-ger, and B. Fernández-manjón, "Tracing a little for big Improvements : Application of Learning Analytics and Videogames for Student Assessment," *Procedia Comput. Sci.*, 2012.

[9] C. Alonso-Fernandez, A. Calvo, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon, "Systematizing game learning analytics for serious games," *2017 IEEE Glob. Eng. Educ. Conf.*, no. April, pp. 1106–1113, 2017.

[10] M. Bienkowski, M. Feng, and B. Means, "Enhancing teaching and learning through educational data mining and learning analytics: An issue brief," *Washington, DC SRI Int.*, pp. 1–57, 2012.

[11] F. Mehm, S. Göbel, and R. Steinmetz, "An Authoring Tool for Educational Adventure Games," *Int. J. Game-Based Learn.*, vol. 3, no. 1, pp. 63–79, 2013.