Full Lifecycle Architecture for Serious Games: Integrating Game Learning Analytics and a Game Authoring Tool

Cristina Alonso-Fernandez[1], Dan C. Rotaru[1], Manuel Freire[1], Ivan Martinez-Ortiz[1] and Baltasar Fernandez-Manjon[1]

[1] Facultad de Informática, Complutense University of Madrid, C/ Profesor José García Santesmases 9, 28040 Madrid, Spain
{crisal03, drotaru}@ucm.es
{manuel.freire, imartinez, balta}@fdi.ucm.es

Abstract. The engaging and goal-oriented nature of serious games has been proven to increase student motivation. Games also allow learning assessment in a non-intrusive fashion. To increase adoption of serious games, their full lifecycle, including design, development, validation, deployment and iterative refinement must be made as simple and transparent as possible. Currently serious games impact analysis and validation is done on a case-by-case basis. In this paper, we describe a generic architecture that integrates a game authoring tool, uAdventure, with a standards-based Game Learning Analytics framework, providing a holistic approach to bring together development, validation, and analytics, that allows a systematic analysis and validation of serious games impact. This architecture allows game developers, teachers and students access to different analyses with minimal setup; and improves game development and evaluation by supporting an evidence-based approach to assess both games and learning. This system is currently being extended and used in two EU H2020 serious games projects.

Keywords: Learning Analytics, Serious Games, E-learning, xAPI, uAdventure

1 Introduction

Games have been applied in multiple fields such as medicine [1], science [2], arts [3] or military [4]. Their benefits, such as their goal-oriented, engaging nature, makes them especially adequate for education, where students' motivation is essential.

Serious Games (SGs) are videogames where the main purpose is not pure entertainment: it may be to teach, to change an attitude or behavior or to create awareness of a certain issue [5]. There are several examples of successfully applied SGs: *Aislados* helped teenagers to prevent drug addiction and other risk behaviors [6] while *Darfur is Dying* created awareness of the ongoing war in Sudan in 2006 [7].

Most games, however, follow the black box model when it comes to collecting players' interactions: they merely report final results, which are far less informative than access to real-time learning progress. In fact, the usual method to evaluate SGs effectiveness is through pre-post questionnaires [8]. This evaluation method requires significant investments of time and effort, and individual solutions have to be provided ad hoc for every particular game, severely impacting the scalability of the solution.

The pre-post evaluation method also fails to detect changes in learning as they occur. Learning concepts appear at different stages of the game for different players; and this learning process should be tracked in real-time through the observation of in-game interactions for optimal feedback regarding the effectiveness of the games' learning design.

In the entertainment games industry, data analysis has been long applied to capture players' interactions and to improve their user experience as well as the game design [9] in a discipline that is usually called Game Analytics (GA). Meanwhile,

In e-learning and different learning systems, such as learning management systems (LMS), Learning Analytics (LA) is commonly used to capture learners' actions to try to understand their learning process and prevent their failure. There are several definitions of LA; we could define it as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environment in which it occurs" [10]-[11].

To apply these analytics models to SGs, we have to break the black box model to gain insights of players' interactions as they take place. This information could be then related to each student's learning process. Game developers could also benefit from this information to determine areas of minor or greater difficulty for players, or even game bugs such as unreachable areas. If the game design is suitable and the relevant interaction data is captured by tracing players' interactions in the SG, it should be possible to trace the evolution of their knowledge, telling apart the areas where they struggle or shine.

In Section 2, we describe the lifecycle of evidence-based SGs' impact. In Section 3, we describe the proposed abstract architecture for applying game and learning analytics for SGs and the different steps it comprises in design, development and evaluation. In Section 4, we describe a reference implementation as part of two EU H2020 SG-related projects. Finally, in Section 5 we summarize the main contributions and future work.

2 Lifecycle of evidence-based serious games' impact

The combination of LA methods with the technologies long applied in GA allows players' interactions within SGs to be traced and analyzed, providing insight into their learning progress. We call this process Game Learning Analytics [5]. GLA allows an evidence-based approach to games' lifecycle (e.g. development, validation and evaluation).

The lifecycle of a serious game (see Fig. **1**) goes from initial conception to development, validation (which may require several iterations if design flaws are uncovered before widespread release), and exploitation, during which periodic evaluation of student progress and outcomes will take place, once the game is released to its target players. The role of integrated analytics is critical to collecting and analyzing interactions to generate actionable feedback.

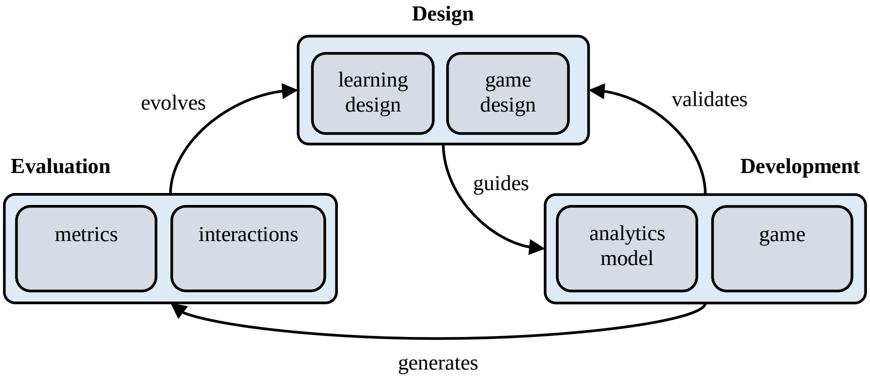

Fig. 1. Lifecycle of a serious game: from learning and game design, through development, validation and evaluation.

Both validation and evaluation require a strong integration of analytics to benefit from feedback and allow players to be evaluated meaningfully on their progress. The use of a unified analytics framework that can do GLA (that is, analyses both at the game level and the learning level) combining separate systems presents advantages for both. Integration of GLA into the development platform also presents significant benefits, comparable to those that test-driven design brings to programming: an early emphasis on choosing and measuring evidence of quality.

Both teachers and students can benefit from closer analytics integration. Analytics can provide real-time knowledge of what students are doing, but interpreting the data is difficult unless it is well presented. Dashboards that combine complementary visualizations appear to be an appropriate way of communicating data to stakeholders, who generally do not need to understand the details of the analysis performed underneath.

To achieve the most informative results, ad-hoc visualizations would be needed; however, providing meaningful default dashboards ensures that no setup is required to start enjoying advantages.

3 Proposed abstract architecture

We propose a complete and scalable analytics architecture, based on standards, that encompasses the whole process from game development to the analysis and visualization of results; a design guided game development where the interaction tracker sends players' interactions to the analytics platform composed of collector, analysis and dashboard; feedback will be sent back to the learning and game design (see Fig. **2**).

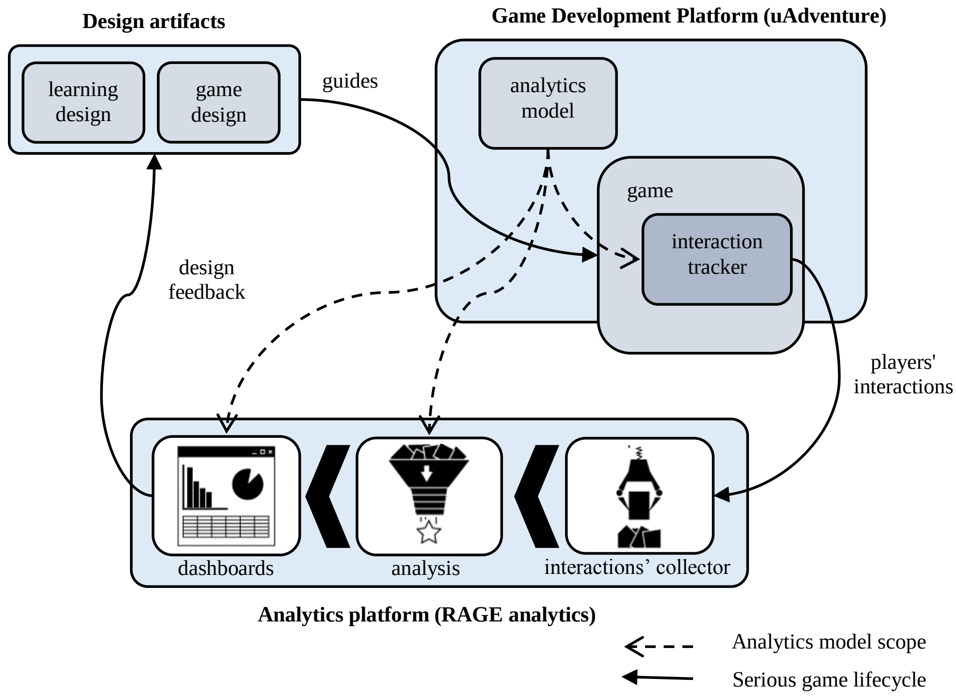

Fig. 2. Proposed architecture: design guides game development.

In this architecture, SGs send traces in a standard-based format to a server that analyzes the data and transforms it into useful information. This analyzed information is then displayed in dashboards for different stakeholders: teachers or instructors in charge of players, the players themselves, game developers or designers and researchers [12].

3.1 Analytics models

Essential questions when performing analytics are those of what to track, how to analyze it, and how to present the results. Another way to frame these questions is to attempt to create a list of visualizations that would evidence that the goals of the game mechanics and the learning design are being met, and work backwards to define the analyses and data-collection that would be required. As illustrated in Fig. **2**, analytics models inform all these decisions, and are an integral part of the game development.

The best analytics models are those that are designed together with the game itself, and are both influenced by the game's design and, where necessary, result in changes to the design that make the resulting game easier to analyze. However, there is a strong case for providing default analytics models whenever possible, to minimize the burden on game designers and developers (which could otherwise decide to forgo GLA altogether) and provide sane defaults on which more targeted analytics can be built.

3.2 Interaction tracking

Analytics requires collection of each player's interaction with the game prior to any analysis. A standard collection format is desirable to allow interoperability and avoid data lock-in. After analyzing the current state of data standards and SGs, in addition to previous experiences applying e-learning standards to SGs [13]-[14], a new interaction representation model has been defined and implemented based on the Experience API standard [15]-[16]: the xAPI Serious Games vocabulary, or xAPI-SG for short.

Experience API (xAPI) is a data format developed by a community led by the Advanced Distributed Learning Initiative (ADL) [17]. The standard derives from Activity Streams, which represent a series of statements regarding learning activities with three main attributes: an actor, a verb and an object. Additional attributes may be included such as the result of the action or a timestamp. Fig. **3** shows an xAPI-SG sample trace generated with [18] representing that the learner completed the SG with a score of seven.


```
{
    "actor": {
        "mbox": "mailto:learner@example.com",
        "name": "Example Learner",
        "objectType": "Agent"
    },
    "verb": {
        "id": "http://adlnet.gov/expapi/verbs/completed",
        "display": {
            "en-US": "completed"
        }
    },
    "object": {
        "id": "http://adlnet.gov/expapi/activities/serious-game",
        "definition": {
            "name": {
                "en-US": "Serious Game"
            },
            "description": {
                "en-US": "Serious game example"
            }
        },
        "objectType": "Activity"
    },
    "result": {
        "score": {
            "raw": 7
        }
    }
}
```


Fig. 3. Sample xAPI trace. The serious game activity was completed by Example Learner with result score of 7.

The interaction model comprises several concepts such as *completables* (e.g. levels, quests or the serious game), *alternatives* (e.g. options in questions or dialogs) and general *variables* to track interactions in the specific domain of SGs. If desired, custom interactions may also be defined to extend the information for a specific SG.

In the architecture illustrated in Fig. **2**, games contain an interaction tracker component that communicates players' in-game interactions via xAPI-SG to the analytics platform. The analytics model defines which interactions, events and targets are reported and how they are mapped to their corresponding xAPI-SG statement attributes, verbs and activity types.

3.3 Game Development Platform

Integrating tracking of GLA with a developed SG is typically performed ad hoc, and both the tracker and the analytics model are external to the chosen game development platform. However, a game development platform which follows the architecture illustrated in Fig. **2**, must include the tracking component in each game, and configure it with an analytics model that is fully integrated with the game's authoring environment. This integration greatly reduces the investment of time and effort required from game developers to benefit from analytics, and therefore increases the likelihood that they will be able, with some additional effort, to improve the game design, the analytics model, and most importantly the game itself in each successive iteration of its lifecycle.

In our reference implementation of the architecture this component is implemented using uAdventure [19]-[20], a complete rewrite of eAdventure, an authoring tool for point-and-click games written in Java and previously developed by the e-UCM Research Group [21]-[22]. As many platforms and devices no longer support Java, uAdventure is built on Unity3D.

3.4 Data analysis

Once the data is collected, the analytics server can begin to process it. Again, the analytics model must provide information on the metrics and KPIs that will be used to prove the effectiveness of the learning design. We distinguish two types of analysis:

1. Game-independent analysis that should be suitable for any SG that connects to the analytics server, as long as the game generates standards-compliant xAPI-SG traces.
2. Game-dependent analysis, which must be developed ad hoc for each game, but allow game and learning designers to create dashboards that perfectly match their game's goal and design.

The information obtained as result of the analysis should be stored for its later visualization; in the proposed architecture, analytics results are stored in a time series database (in our reference implementation of the architecture, Elasticsearch [23] is used), which can analyze and query large amounts of data in semi-real time, and is especially suited for later visualization.

3.5 Data visualization

To facilitate the understanding of the analysis results for a range of stakeholders (including teachers, students and developers), it is important to provide each with informative dashboards to display results. The need for easy to understand and informative visualizations is especially important in the case of teachers, which can greatly benefit from real-time information to monitor a class while students are playing a game, and to provide targeted feedback to students that get stuck. Students will see their own personal progress in real-time and a general ranking within the same class competing with other students. Visualizations about the overall usage of the games such as session's length and server loading are shown to the developers, though the student specific data is only shown to the teacher because of privacy concerns.

In our reference implementation of the architecture of this component, visualization dashboards have been developed using Kibana [24], an open source visualization engine which is directly connected with Elasticsearch. Kibana provides a browser-based interface to quickly develop analysis and visualizations with different predefined graphics (e.g. line chart, bar chart, pie chart). Two sample visualizations available: the left one shows number of correct (in green) and incorrect (in red) answers in each alternative; the right one, the progress (in range 0 to 1) in each of the three completables and in the complete serious game for each player (see Fig. **4**). New visualizations and dashboards may be configured in the system by selecting the required fields to be analyzed and displayed in the graphs.

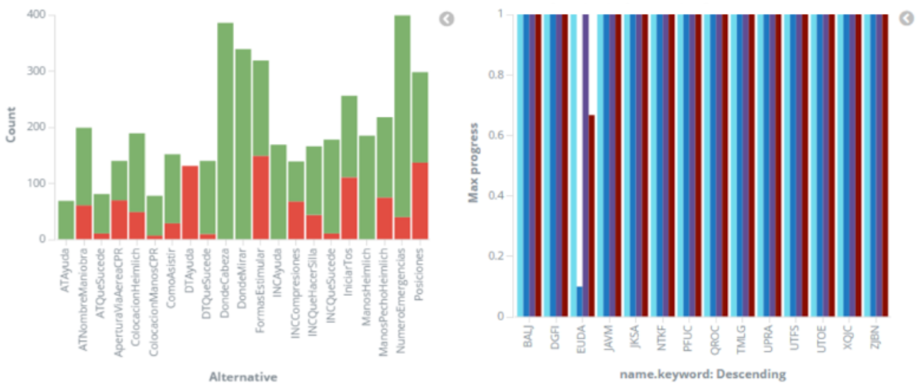

Fig. 4. Two sample visualizations containing errors made in *alternatives* and progress made in *completables*.

To extend the usefulness of these visualizations, recommended actions may be included to help teachers provide timely feedback. In our implementation, alerts (situations that require immediate action, e.g. "a student has made an important error") and warnings (less urgent actions, e.g. "a student has been inactive for two minutes") provide near real-time information to teachers. Fig. **5** shows the general view of alerts and warnings (on top); clicking on a specific user in the general view displays details on the selected user's alerts and warnings (bottom part).

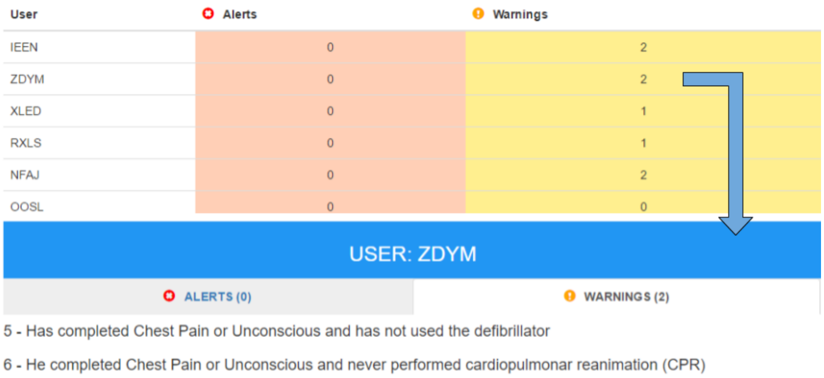

Fig. 5. General view of alerts and warnings for each anonymized user. Clicking on a specific user provides further details on the user's alerts and warnings.

4 GLA architecture reference implementation

A complete architecture to manage GLA requires handling several interlocking parts: data tracking, data analysis and results visualization. We proposed the following standard-based architecture, a combination of modules that work together to analyze and visualize information collected from SGs [12]. Fig. **6** shows a diagram of the GLA architecture: from learning and game design, the serious game is created. Its embedded tracker sends xAPI traces to the collector, which stores them in a LRS for batch analysis and sends them for real-time analysis. Visualizations developed from analytics provide feedback to come full circle improving the learning and game design and helping to assess students.

- The learning and game design determine the SG implementation. This includes the game mechanics, structure, goals and in-game items or characters. Both these designs also determine the elements that will contain the relevant information for learning (usually as game variables), that is, the elements that are essential to be tracked as they will tell if the game is helping players to learn or not.
- The SG itself will use a tracker component to send traces in xAPI format (called *statements*). The tracker provides an application programming interface designed to send data to the server without having to know the underlying xAPI format specification. Current tracker implementations include Unity C#, pure C# and JavaScript to facilitate their integration with different SGs.
- xAPI-SG statements are sent to a collector endpoint on the server-side. Then they are sent to a real-time analysis component which updates the information for each

player. xAPI statements are also sent to a Learning Record Store (LRS), a scalable database designed for storing xAPI learning activity data to perform a different analysis ("batch analysis").

- With the analysis results, visualizations in suitable dashboards show all metrics of interest for the stakeholders. If configured, personalized visualizations, alerts and warnings will also be displayed.
- Finally, the process is completed when the information obtained through analysis and visualizations provides feedback and improvement actions that can be reintroduced in the system for following iterations of the learning and game design. Additionally, this information may also help teachers in students' assessment.

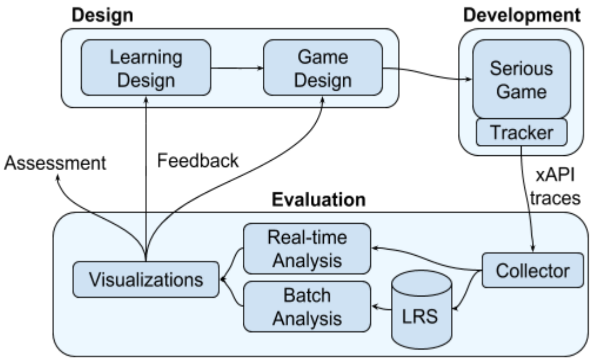

Fig. 6. GLA system: from design to development and evaluation.

4.1 Architecture implementation

The GLA architecture described above has been developed as part of an EU H2020 SG-related project. All components are open source and available online[1]. When deployed, the components are launched as Docker containers [25] which eases deployment by eliminating all dependencies except for Docker itself. The main components are:

- An authorization and authentication component (A2), which enforces access controls and allows integration with existing institutional single sign-on systems, as well as hiding the complexity of all other components behind a reverse proxy.
- A frontend that allows stakeholders (teacher, developer and student) to configure and/or view dashboards for which they have appropriate credentials.
- A backend that collects incoming traces, analyzes them (either in real-time from incoming data or on-demand from LRS queries), and exhibits results for the frontend.
- A xAPI Learning Record Store (LRS) which allows third-party systems to query xAPI traces collected by the backend.

[1] eUCM Research Group, RAGE Analytics, (2017). https://github.com/e-ucm/rage-analytics

Since teachers, students and other institutional stakeholders typically already enjoy single sign-on in institutional systems, the A2 component has been extended to interact with these institutional systems via login plugins for either SAML2 (Security Assertion Markup Language v 2.0) [26] or LTI (Learning Tools Interoperability) [27]. These plugins simplify deployment in applicable institutions, since no additional credentials need to be created. Additionally, in the case of LTI, certain setup tasks, such as registering students as belonging to a particular teacher's class, can be eliminated altogether. Further information about the RAGE Analytics System can be found in the e-UCM Research Group's GitHub wiki page [28].

4.2 System applications

The proposed system has been recently tested in an experiment with more than 200 students of a school in Madrid. With the goal of evaluating the whole GLA architecture, students played a SG to teach first aid maneuvers [29] while teachers obtained real-time dashboards about what students were doing, being able to control which students were progressing and which students were falling behind.

Students' anonymization was ensured via unique codes provided at the beginning of the experiment and required to access the game. Teachers were the only holders of the mapping between individuals and codes; all information collected in the system was only identified through this code.

During the experiments, it came to light that some visualizations were not easy to understand by teachers. As teachers are the only experts qualified in the evaluated field to know if students are learning or not, dashboards need to provide information in a clearer or more simplified manner for their easy comprehension.

5 Conclusions and future work

Although GLA is no longer an emerging field, it is still performed mostly through ad-hoc solutions, and therefore it could greatly benefit from a general standardized approach. Such an approach can increase adoption of SGs by promoting quality through evidence-based iterative improvement and better evaluation; while minimizing GLA deployment and development costs.

Our approach has three main pillars: first, the integration of analytics into the game authoring tool itself; second, the use of a standard xAPI-SG interaction model to standardize trace collection; and third, a default set of analysis and visualizations for the main SG stakeholders, including game developers, teachers and students.

Games created with the authoring tool uAdventure can effortlessly integrate tracking and analysis of results. Moreover, they can be deployed on a wide range of platforms, and can also support geolocalization [30].

Ad hoc analyses and visualizations can also be created by adding configuration files to the system or selecting the attributes to be visualized, respectively. Theses personalized analyses and visualizations could be useful if a particular game requires them; however, a moderate use of these is recommended as the more personalized configurations included, the less general the solution will be and the more effort it will require.

With these contributions, we have advanced towards a systematized standards-based system that helps to complete the full circle of GLA for SGs: learning and game design, SG development, tracking, analysis, visualization, and feedback, as depicted in Fig. **2**.

However, there is still work to do. Some areas for improvement include:

- Improved explanations to allow novice users to interpret dashboard visualizations; especially for users that may not have been involved in the game design process.
- Simplified creation for custom visualizations. We are developing a wrapper around Kibana's built-in authoring environment to ease the process for non-programmers.
- Bidirectional communication between the tracker and the server, allowing the tracker to be notified when certain conditions are fulfilled in order to adapt the game's learning design and/or provide in-game, real-time feedback to players.

The system will be tested in more experiments with serious games currently under development. Work will continue on these and other improvements as the system is going to be improved and extended as part of the H2020 SG-related projects RAGE and BEACONING.

Acknowledgements

The e-UCM research group 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. Evans, K.H., Daines, W., Tsui, J., Strehlow, M., Maggio, P., Shieh, L.: Septris: a novel, mobile, online, simulation game that improves sepsis recognition and management. Acad. Med. 90, 180–4 (2015).
2. Center for Game Science at the University of Washington: Treefrog Treasure, http://centerforgamescience.org/blog/portfolio/treefrog-treasure/.
3. Manero, B., Torrente, J., Serrano, Á., Martínez-Ortiz, I.: Can educational video games increase high school students' interest in theatre? Comput. 87, 182–191 (2015).
4. United States Army: America's Army, https://www.americasarmy.com/.
5. Freire, M., Serrano-Laguna, Á., Iglesias, B.M., Martínez-Ortiz, I., Moreno-Ger, P., Fernández-Manjón, B.: Game Learning Analytics: Learning Analytics for Serious Games. In: Learning, Design, and Technology. pp. 1–29. Springer International Publishing, Cham (2016).
6. Asociación Servicio Interdisciplinar de Atención a las Drogodependencias (SIAD): Aislados, http://www.aislados.es/zona-educadores/.
7. interFUEL, L.: Darfur is Dying, http://www.gamesforchange.org/play/darfur-is-dying/.

8. Calderón, A., Ruiz, M.: A systematic literature review on serious games evaluation: An application to software project management. Comput. Educ. 87, 396–422 (2015).
9. El-Nasr, M., Drachen, A., Canossa, A.: Game analytics maximizing the value of player data. Springer, London ;;New York : (2013).
10. Long, P., & Siemens, G.: Penetrating the Fog: Analytics in Learning and Education. Educ. Rev. 31–40 (2011).
11. Siemens, G., Dawson, S., & Lynch, G.: Improving the Quality and Productivity of the Higher Education Sector. Policy and Strategy for Systems-Level Deployment of Learning Analytics. (2013).
12. Alonso-Fernández, C., Calvo Morata, A., Freire, M., Martínez-Ortiz, I., Fernandez-Manjon, B.: Systematizing game learning analytics for serious games. In: IEEE Global Engineering Education Conference (EDUCON). pp. 1106–1113 (2017).
13. Serrano, A., Marchiori, E.J., del Blanco, A., Torrente, J., Fernandez-Manjon, B.: A framework to improve evaluation in educational games. In: Proceedings of the 2012 IEEE Global Engineering Education Conference (EDUCON). pp. 1–8. IEEE (2012).
14. del Blanco, Á., Marchiori, E.J., Torrente, J., Martínez-Ortiz, I., Fernández-Manjón, B.: Using e-learning standards in educational video games. Comput. Stand. Interfaces. 36, 178–187 (2013).
15. Serrano-Laguna, Á., Martínez-Ortiz, I., Haag, J., Regan, D., Johnson, A., Fernández-Manjón, B.: Applying standards to systematize learning analytics in serious games. Comput. Stand. Interfaces. 50, 116–123 (2017).
16. eUCM: xAPI Serious Games Profile, http://w3id.org/xapi/seriousgames.
17. ADL Initiative: xAPI Specification, https://github.com/adlnet/xAPI-Spec/blob/a752217060b83a2e15dfab69f8c257cd86a888e6/xAPI.md.
18. ADL Initiative: xAPI statement generator, http://experienceapi.com/statement-generator/.
19. Perez Colado, I., Perez Colado, V., Martínez-Ortiz, I., Freire, M., Fernandez-Manjon, B.: uAdventure: The eAdventure reboot - Combining the experience of commercial gaming tools and tailored educational tools. In: IEEE Global Engineering Education Conference (EDUCON). pp. 1754–1761 (2017).
20. Perez Colado, I.: uAdventure: desarrollo del intérprete y de un emulador de videojuegos de -Adventure sobre Unity3D, http://www.e-ucm.es/drafts/e-UCM_draft_294.pdf, (2016).
21. Torrente, J., Blanco, Á., Marchiori, E.J., Moreno-ger, P., Fernández-Manjón, B., Del Blanco, Á.: <e-Adventure> Introducing Educational Games in the Learning Process. IEEE Educ. Eng. EDUCON 2010 Conf. 127, 1121–1126 (2010).
22. e-ucm: eAdventure, http://e-adventure.e-ucm.es/.
23. Elastic: Elasticsearch, https://www.elastic.co/products/elasticsearch.
24. Elastic: Kibana, https://www.elastic.co/products/kibana.
25. Docker: Docker, https://www.docker.com/.
26. OASIS: Security Assertion Markup Language (SAML), https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.
27. IMS Global Learning Consortium: Learning Tools Interoperability, https://www.imsglobal.org/activity/learning-tools-interoperability.
28. e-ucm: RAGE Analytics GitHub Wiki Page, https://github.com/e-ucm/rage-analytics/wiki.
29. e-ucm: First Aid Game, http://first-aid-game.e-ucm.es/.
30. Pérez Colado, V.: Extendiendo uAdventure con funcionalidades de geoposicionamiento, http://www.e-ucm.es/drafts/e-UCM_draft_302.pdf, (2017).