SoftwareX 27 (2024) 101763

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

Optimizing multimedia and gameplay data labeling: A web-based tool for Game-Based Assessment

Manuel J. Gomez *, José A. Ruipérez-Valiente, Félix J. García Clemente

Facultad de Informática, Universidad de Murcia, Murcia, Spain

ARTICLE INFO

Dataset link: https://github.com/CyberDataLab/gba-labeling-tool/blob/main/sampleData

Keywords:
Game-Based Assessment
Data annotation
Data labeling
Human labeling
Multimedia data

ABSTRACT

In this research, we introduce a novel open-source labeling tool, the Game-Based Assessment (GBA) Labeling Tool, specifically designed to address current challenges for data labeling in GBA scenarios. This web-based application facilitates the annotation of audio, video, and game event data, offering three different types of annotations – global, time instant, and time window annotations – to enhance accuracy in the labeling process. The tool also offers customizable labels and various types of visualizations to support different contexts and scenarios.

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-24-00172
Permanent link to Reproducible Capsule	–
Legal Code License	MITLicense
Code versioning system used	git
Software code languages, tools, and services used	CSS, JavaScript, HTML, Python, Django
Compilation requirements, operating environments & dependencies	Ubuntu/MacOS, Python ≥ 3.7.2
If available Link to developer documentation/manual	https://github.com/CyberDataLab/gba-labeling-tool/blob/main/readme.md
Support email for questions	manueljesus.gomezm@um.es

1. Motivation and significance

Modern technologies are having a significant impact on every industry, including gaming and education [1]. Serious Games (SGs), which are designed specifically for purposes other than or in addition to pure entertainment [2], have gained significant attention in recent years. In particular, SGs are being explored for their potential to provide more valid assessments compared to traditional assessment approaches. Game-Based Assessment (GBA) has been increasingly used in various domains such as education, health, military, and industry [3]. With the growing popularity and adoption of GBA, there has also been a significant increase in the quantity of data generated from user interaction.

Data collected for SGs and assessment research covers diverse measurements, including performance skills and behavioral factors relevant to both the process and outcomes [4]. Usually, various data types are collected, including audio recordings, video captures, and game event data from players' interaction with games. In this regard, labeled data plays a crucial role in the development of models and algorithms that enable researchers to gain deeper insights into learners' behaviors, engagement, and performance [5]. However, the process of data labeling is time-consuming and challenging. In many real-world scenarios, large-scale labeled datasets can be very costly to acquire, specially when expert annotators are required [6]. As datasets for training and testing of algorithms get increasingly larger, there is a need for efficient and user-friendly solutions to facilitate the annotation of multimedia data [7].

* Corresponding author.
E-mail address: manueljesus.gomezm@um.es (Manuel J. Gomez).

https://doi.org/10.1016/j.softx.2024.101763
Received 14 March 2024; Received in revised form 26 April 2024; Accepted 11 May 2024
Available online 20 May 2024

In the context of GBA, where large quantities of data are collected, the lack of standardized tools and methodologies for data labeling raises significant challenges. Researchers usually employ primitive methods such as Excel worksheets or manual annotations, and the existing multi-functional labeling tools are too general to be used with GBA data. Wang et al. [8] reported the very limited existence of labeled datasets in many real-world scenarios, and specially in education. They also noted that labels are often annotated by multiple workers with different expertise, leading to noise and inconsistency. Although we identified that many previous SGs studies have applied AI models/techniques [9,10], the majority of these studies heavily rely on automatically labeled data derived from in-game measures or statistics. Therefore, we note that human-labeled data is often underutilized, potentially due to the considerable time and economic resources required for its acquisition.

With this tool, we aim to address existing challenges associated with data annotation in GBA by developing a web-based open-source tool specifically designed for GBA data annotation. By integrating the possibility of labeling audio, video, and game-event data in the same tool, we provide researchers with a flexible and efficient solution for labeling various types of multimedia data. It provides different annotation types (global, time instant and time window annotations) and different ways of visualizing the same data. In addition, the tool's interface and workflow optimize the annotation process, potentially reducing inconsistencies between annotations and users' interpretations. By developing this tool, we contribute to the GBA literature and enhance the potential for leveraging multimedia data for educational research and assessment.

A comparative analysis with already existing tools in the literature reveals the unique features and enhancements of our tool. Zhang et al. [11] developed a conceptual framework for data labeling, and Da Silva et al. [7] presented an open source multipurpose tool for the annotation of multimedia datasets with collaborative annotation capabilities. While this multipurpose tool offers versatility in accepting images, audio, point cloud, and general signals, our tool stands out with its specialization in integrating game event data and providing an automatic feature report specifically designed for GBA scenarios. In addition, our tool allows to directly incorporate Unity projects into the environment, a capability challenging to achieve with generic annotation tools. Moreover, the *Videojot* tool [12] focuses on the annotation of video streams by combining zoom, drawing, and temporal social bookmarking, meanwhile Palotai's tool [13] emphasizes ML-based event recognition in video data to enable automated annotation. In contrast, our tool emphasizes manual annotation with customizable labels tailored for GBA scenarios, while also accepting multimedia data. Furthermore, Philbrick's *RIL-Contour* [14] is specifically designed for medical imaging datasets and promotes collaborative annotation by supporting concurrent multiuser workflows. While existing tools offer versatility, they lack specialized features required for effective labeling in GBA scenarios. Our tool addresses the specific needs of GBA research, enhancing the efficiency and accuracy of the data labeling process, and setting it apart from other generic or context-specific tools.

2. Software description

2.1. Software architecture

We have built the "GBA Labeling Tool" as a web application built using the Django framework. Our tool is developed as a server-side web application, providing robust and scalable capabilities. Users can access our tool using any web browser, making it easily accessible and compatible with any platform. In Fig. 1 we can see the complete platform's architecture.

Django is a web framework that uses Python for building very fast dynamic websites. It implements advanced security measures, including SQL injection prevention and data validation [15]. Django's architecture is inherently scalable, enabling web applications to expand without causing any disruptions to their functioning and the flow of traffic [15]. Django offers scalability and speed optimization capabilities such as caching, load balancing, or horizontal scaling, making it an ideal choice for addressing these difficulties. Considering that the GBA Labeling Tool is expected to be used by a relatively small numbers of users concurrently, Django's scalability features align perfectly with our needs.

We employ the SQLite embedded database, ideal for most low to medium traffic applications, supporting tens of thousands of transactions per second and offering faster blob data processing compared to file systems [16]. The combination of SQLite's efficiency and Django's capabilities for improving performance of database queries ensures our tool can efficiently handle large datasets while maintaining responsiveness. Next, we present the different modules that form the architecture.

2.2. Data input

Users can upload *audio*, *video*, and *game event data*. Each individual replay must be linked to a game, user group (e.g., a classroom), and a user. If these details are not included along with the replays, the platform will use the default game, group, and/or user.

Multimedia data (audio and video) are directly stored as replays in the database. Regarding gameplay data, the tool incorporates a *custom parser* to transform the raw data, which can be received in multiple formats such as CSV (Comma-Separated Values) or JSON (JavaScript Object Notation), into a different format for further processing and visualization. Our parser maps every individual event in the log file into an *event* instance in the database. This event model includes essential information such as the timestamp, event type, and user-related details. In addition, our backend includes a *feature computing* module that analyzes the data by iterating through each individual event and calculates a set of useful features to add context to the labeled replay. There are two types of features: *context features*, related to the user up to that point in the gameplay, and *attempt features*, which are specific to that particular attempt. These features include information like the active and inactive time (in seconds), the number of completed levels, or a specific count of events.

2.3. Customization

Gameplay data can be fully customized by the user. The first customization option is event naming. Typically, event names defined by the original event model are lengthy or hard to read. By customizing each event's name, annotators can create more readable replays and make the annotation process more efficient.

Annotators can also define new types of events by combining existing ones along with a set of regular expression operators. Fig. 2 shows the "Custom events definition" page. On the left column of the page we can find the "items list", which includes the original events from a specific game, as well as a wildcard event (*Any*). On the right column, three operators are available: *?* matches the preceding event zero or one time; * matches anything in place of the *; and + matches the preceding event one or more times. To use these operators effectively and avoid errors or potential infinite loops, users should have some knowledge of regular expressions. The user can drag any of the items on the columns into the center box and combine them to create new events to be shown in textual replays. For instance, Fig. 2 shows the custom events definition interface, where a user is defining an event called *NFLAPS* as "(FLAP)+". This means that any sequence of one or more *FLAP* events will be replaced by the new custom event. By using custom events, annotators can adapt the replays to their specific needs, ensuring they are more representative of the gameplay.

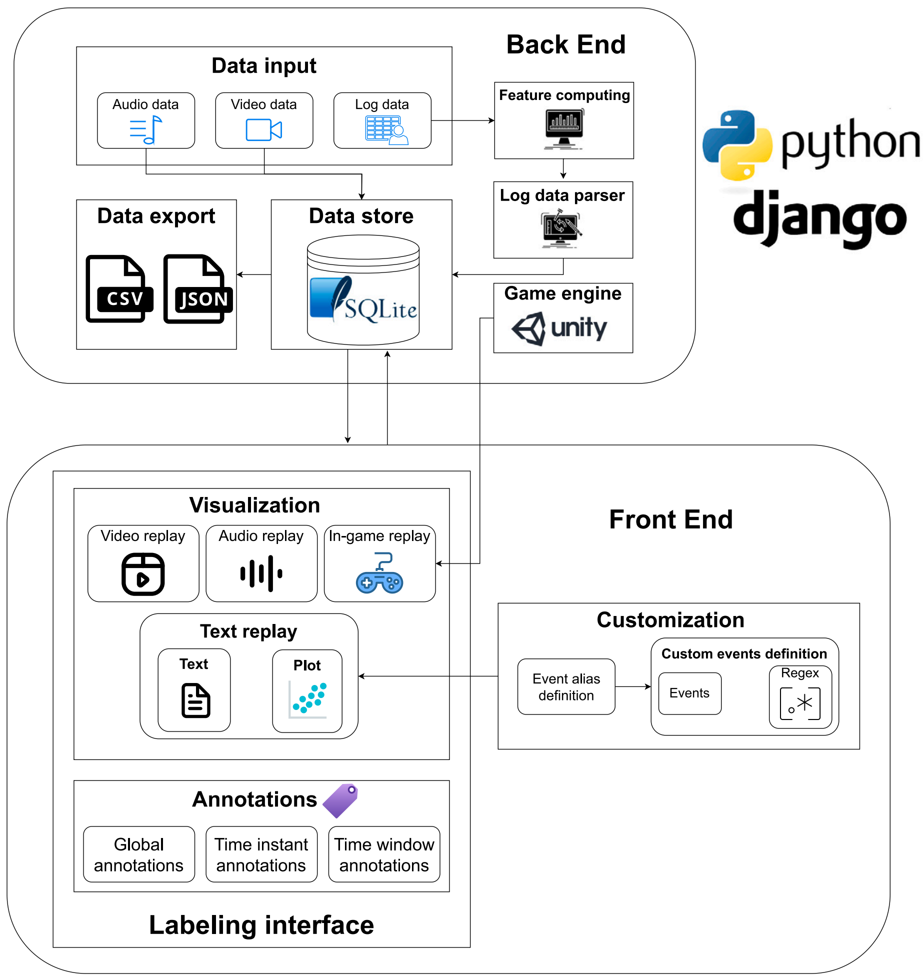


Fig. 1. Platform's architecture.

2.4. Visualization

The three different data formats result in four types of replays: *audio*, *video*, *in-game*, and *text* replays. When the user selects a specific replay, the tool loads the replay file to visualize it. For instance, in Fig. 3 we can see an *audio* replay, which is represented as waveforms.

We can visualize *game event data* in two different ways. Since log data is parsed and stored in the database, we can use the *game engine* itself to visualize the replay if the game allows it. In the example shown in 4(a), we use *Shadowspect*, a 3D geometry game designed as a formative assessment tool to measure math core standards [17]. The tool includes a set of templates that offer integration with *Unity WebGL* applications, simplifying the process for users to incorporate their projects into the tool environment.

Secondly, we can generate a textual ("pretty-printed") representation using the original events generated by the users' interaction with the games. In this example shown in Fig. 4(b), we can observe the game's start, the number of level attempts, each action's timestamp relative to the previous action, and the final outcome (completed or not). In both log data visualizations, features previously computed are incorporated in the right column. This data enrichment helps to provide a more informative and detailed understanding of replays.

In the text replay visualization interface, three different buttons located in the lower right corner offer annotators even more options to customize their data. The *Collapse/Uncollapse events* button allows merging consecutive identical actions, showing the number of times that the action has been performed consecutively. The *Replace custom events* button allows to replace the original events with their previously defined custom events. Finally, the *Text/Visual mode* button allows users to create a plot using the log data to visually represent the events. For each replay, two plots are available: one generated with the original events and another generated using custom events. In Fig. 5 we can see an example of a plot generated using custom events and the same replay as in Fig. 4(b).

2.5. Annotations

In our tool, an annotation (or tag) is defined as the relationship between a user (annotator), a replay, a label, a value, a time interval, and a type of annotation. First, **global annotations** refer to the entire duration of the replay, indicating that a specific label value has been detected throughout the entire replay. Second, **time instant annotations** refer to a single point in time, indicating that a specific label value has been detected at a particular moment during the replay. Finally, **time window annotations** refer to a time interval (start and end) between the replay's beginning and end.

When adding a time instant annotation to an audio or video replay, the tool will automatically assign the time instant to the moment displayed in the media player, as shown in Fig. 6(a). Moreover, when the user wants to add a time window annotation in an audio or video replay, they need to set the start and end of the time window using the respective buttons. For time instant annotations in a textual replay, a dropdown menu is available to select the event the user wants to annotate. Once selected, the tool assigns the global timestamp associated with that event to the annotation. Fig. 6(b) shows the annotation interface in a text replay when adding a time window annotation. We

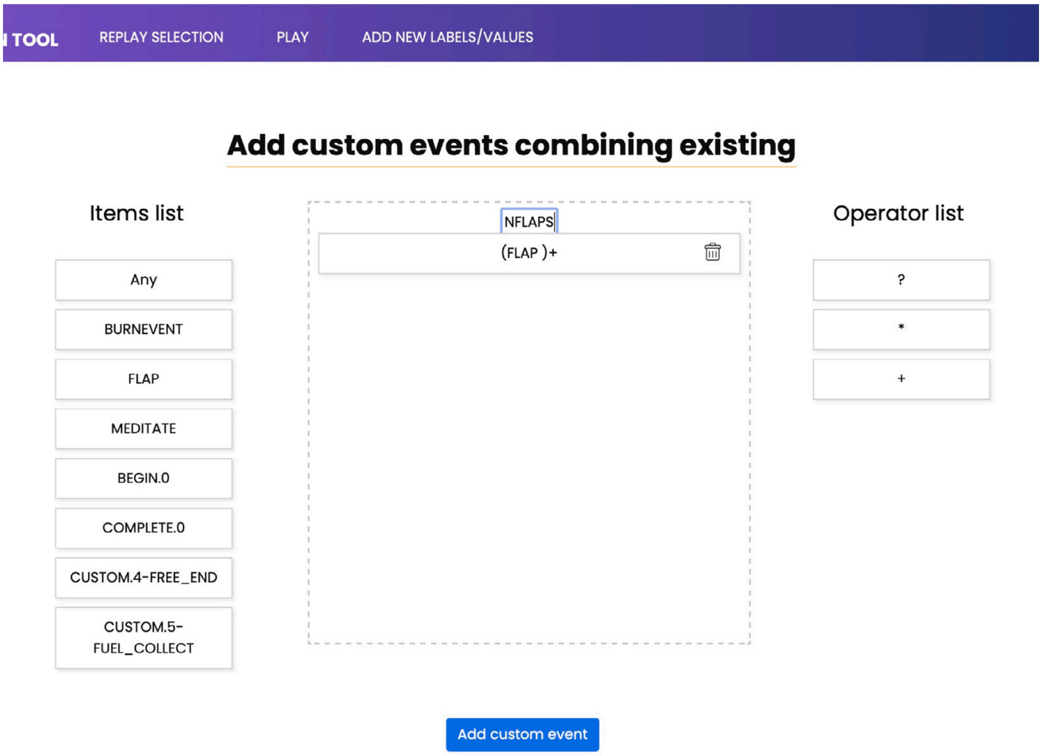


Fig. 2. Custom events definition page.

Fig. 3. Audio replay visualization.

can see a dropdown menu to select the type of annotation, event, label, and value, and two additional buttons to set the time window beginning and end.

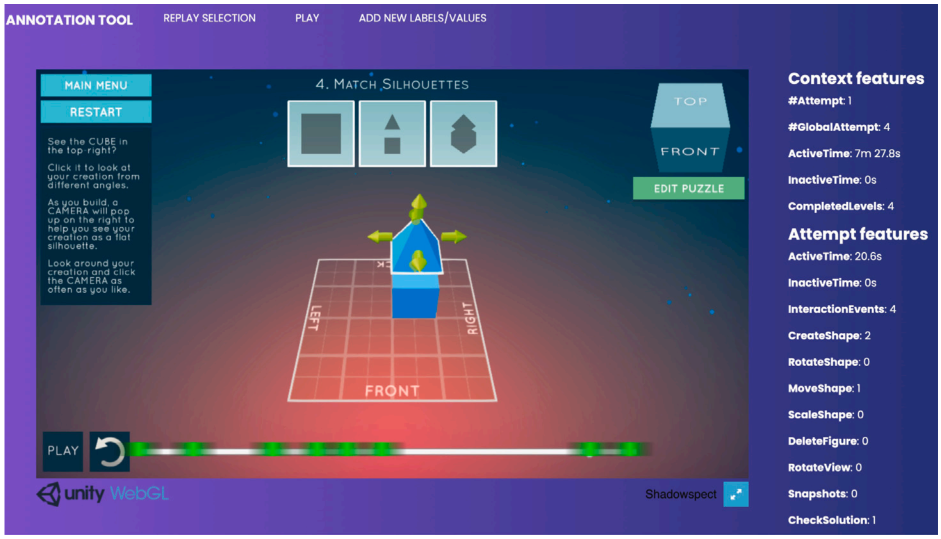
2.6. Data export

Users can export their annotated data at any time using the *Export* button available in the game, group, and user selection menus. The tool allows exporting data from all games or filtering data for specific games or groups. Two export formats are available: *JSON* and *CSV*, providing flexibility for users to choose the standard format that best suits their needs.

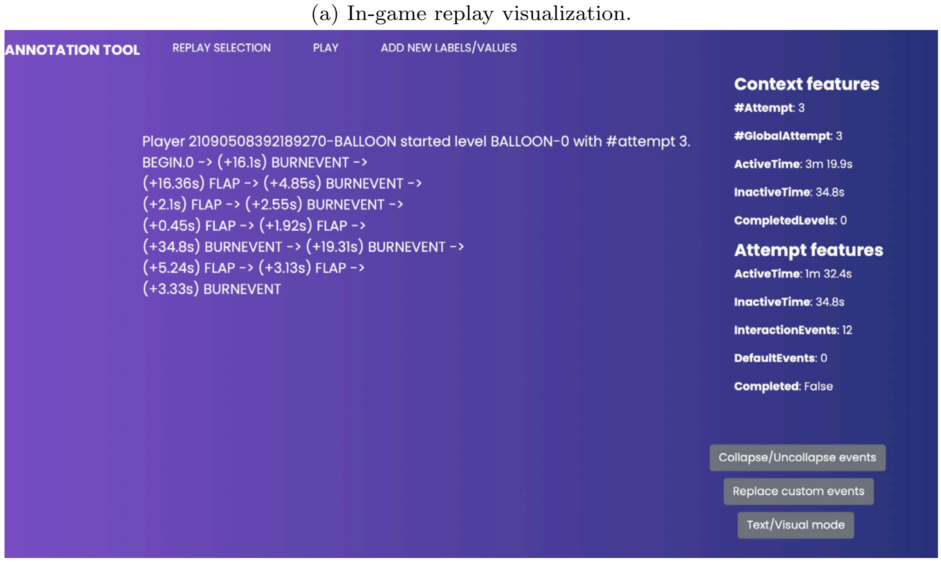
```
{
  "label": "Creativity",
  "typeTag": "TimeWindow",
  "value": "Creative",
  "annotationTime": 7.4,
  "finalAnnotationTime": 131.57,
  "game": "BALLOON",
  "user": 21090508420934984,
  "level": "BALLOON-0",
  "group":"MainGroup"
},...
```

Listing 1: Example of one of the JSON files generated by the export option.

Listing 1 shows a fragment of a JSON file generated by the export utility. As we can see, the JSON file contains all the information for each annotation added, including the type of tag, value, game, or level.



(a) In-game replay visualization.



(b) Text replay visualization.

Fig. 4. Log data visualizations.

3. Illustrative examples

To facilitate the reader's understanding and visualization, we have created a complementary video (https://youtu.be/szOU9HL1QB0) that demonstrates how a user would label game event data. Specifically, the user in this example aims to label the *persistence* competence by visualizing game replays from a game called *Crystal*.

First, the user begins by creating an alias for different game events and defining a custom event named "NRELEASES", defined as *one or more MOLECULERELEASE events*. Then, the user proceeds to label different replays by visualizing the game event data, collapsing events, replacing custom events, and plotting the data to obtain a better overview of each replay. Once the labeling process is complete, the video shows how the user exports the data into CSV format for downloading.

In addition, we have created another complementary video (https://youtu.be/XhGKdZfutOk) that provides a complete overview of the tool, including the use of audio, video, and game event data, as well as different annotation types.

4. Impact

The expected impacts of our tool primarily include the rise in adoption of multimedia and gameplay data labeling methodologies within GBA research. This tool is the first designed to meet specific requirements for annotating datasets in GBA scenarios, offering novel capabilities not found in existing annotation tools. We believe it can be applied in diverse GBA environments due to its versatility and adaptability to different types of multimedia data. This open-source tool contributes to the standardization of methodologies in GBA research, facilitating collaboration between researchers and the validation and comparison of results across studies.

One potential advantage is the reduction of annotation time, as the tool provides a user-friendly interface with customizable labeling options, helping practitioners to label data more efficiently. Moreover, the tool can contribute to higher consistency between annotations through predefined annotation types, labels, and values. Although using this labeling tool helps to optimize the annotation workflow, limitations still exist, as relying on manual annotations often results in noise and inconsistencies due to differences in annotators' interpretations of data and labeling criteria. However, by combining the strengths of our tool with training and consistency measures, researchers can mitigate

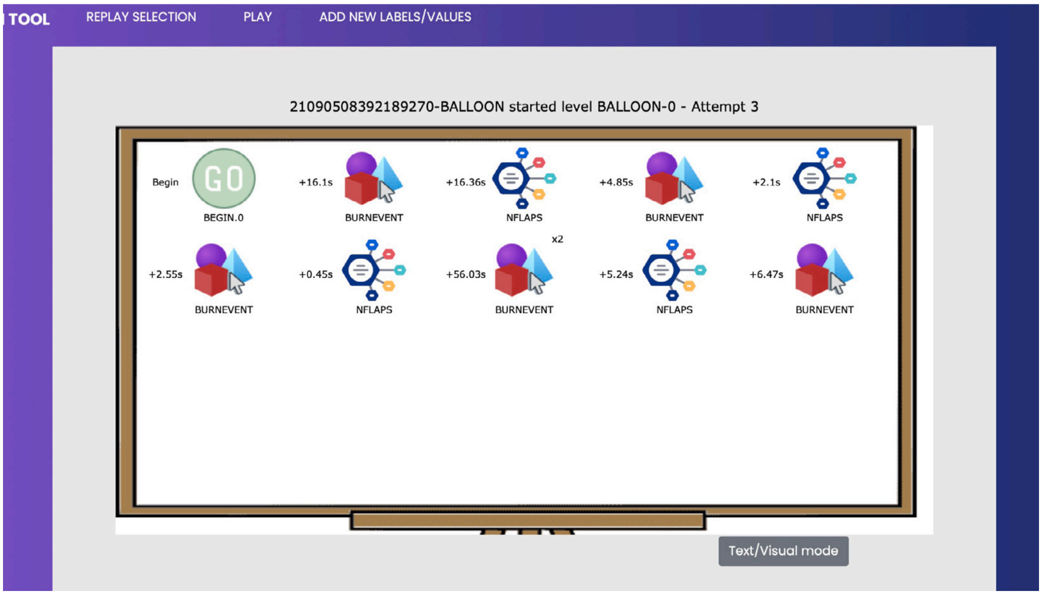
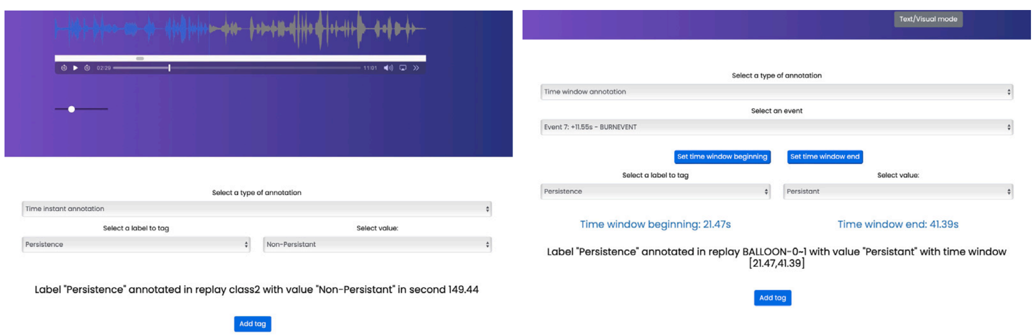


Fig. 5. Plot visualization using log data.



(a) Time Instant annotation in an audio replay. (b) Time window annotation in a textual replay.

Fig. 6. Screenshots of the annotation interface in the tool.

potential issues and ensure the reliability of the annotated data. Moreover, exploring AI-assisted labeling techniques can help enhance the efficiency and accuracy of the annotation process, as well as enable the annotation of more complex and larger datasets.

Furthermore, the integration of annotated datasets with common analysis tools in the field allow researchers to conduct more complex analysis and gain deeper insights. There are several tools that are often used for statistical analysis and data visualization in educational research, such as *RapidMiner*, *Orange*, *SPSS*, and different packages in Python like *Scikit-learn* or *NumPy* [18,19]. Researchers can benefit from these tools and explore different correlations and patterns between game event data and learning outcomes, as well as apply sophisticated algorithms for predictive modeling or clustering, helping to advance the understanding of the impact of game-based learning on educational outcomes.

5. Conclusions

In this research, we introduce an open-source tool to address existing challenges in GBA data labeling. Our tool supports labeling of audio, video, and game event data, with a custom parser that integrates game event data to facilitate the analysis of gameplay performance and patterns. This includes a summary of game event data, as well as the possibility to define custom events by combining existing ones along with a set of regular expression operators. Moreover, users can employ three annotation types and customize labels and values to meet the unique requirements of GBA scenarios. Additionally, the integration with Unity WebGL applications simplifies the process for users to incorporate their Unity projects into the labeling tool environment. Finally, users can export their labeled data in both CSV and JSON formats, facilitating data sharing and analysis.

The illustrative example demonstrates the practical potential of the proposed tool, and the open-source repository includes dataset samples that enable straightforward use of the tool. While our tool can help to optimize and standardize the annotation process, ensuring consistency among annotators may still require training and control. We plan to incorporate collaborative annotation features, such as calculating agreement between annotators, to reduce noisy data and improve annotations by identifying discrepancies. Furthermore, we plan to explore the integration of AI-assisted labeling techniques to enhance the efficiency and accuracy of the annotation process. This would help not only to reduce dependency on manual annotation but also to reduce noise between different annotators by providing automated assistance in the annotation process. Finally, to validate and enhance the tool's usability and accessibility, we plan to explore user feedback and conduct empirical usability studies, ensuring its accessibility to a wider audience and further validating its usability.

CRediT authorship contribution statement

Manuel J. Gomez: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **José A. Ruipérez-Valiente:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Formal analysis, Conceptualization. **Félix J. García Clemente:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Manuel J. Gomez reports financial support was provided by Fundación Séneca. Manuel J. Gomez reports financial support was provided by Cybersecurity National Institute. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Sample data has been provided in the repository at https://github.com/CyberDataLab/gba-labeling-tool/blob/main/sampleData.

Acknowledgments

This work was partially supported by a) grant 21795/FPI/22 - Séneca Foundation. Cofinanced by Innovatiio Global Educación. Region of Murcia (Spain), b) REASSESS project (grant 21948/JLI/22) and SEMANTIC proof of concept (grant 22238/PDC/23), funded by the Seneca Foundation, Science and Technology Agency of the Region of Murcia, and c) the strategic project CDL-TALENTUM from the Spanish National Institute of Cybersecurity (INCIBE) and by the Recovery, Transformation and Resilience Plan, Next Generation EU.

References

[1] Ullah M, Amin SU, Munsif M, Safaev U, Khan H, Khan S, et al. Serious games in science education. A systematic literature review. Virtual Real Intell Hardw 2022;4(3):189–209.
[2] Becker K. What's the difference between gamification, serious games, educational games, and game-based learning. Acad Lett 2021;209:1–4.
[3] Kato PM, de Klerk S. Serious games for assessment: Welcome to the jungle. J Appl Test Technol 2017;18(S1):1–6.
[4] Smith SP, Blackmore K, Nesbitt K. A meta-analysis of data collection in serious games research. In: Serious games analytics: Methodologies for performance measurement, assessment, and improvement. Springer; 2015, p. 31–55.
[5] Serrano Á, Marchiori EJ, del Blanco Á, Torrente J, Fernández-Manjón B. A framework to improve evaluation in educational games. In: Proceedings of the 2012 IEEE global engineering education conference. EDUCON, IEEE; 2012, p. 1–8.
[6] Gao M, Zhang Z, Yu G, Arık SÖ, Davis LS, Pfister T. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In: Computer vision–ECCV 2020: 16th European conference, glasgow, UK, August 23–28, 2020, proceedings, part x 16. Springer; 2020, p. 510–26.
[7] Da Silva JL, Tabata AN, Broto LC, Cocron MP, Zimmer A, Brandmeier T. Open source multipurpose multimedia annotation tool. In: Image analysis and recognition: 17th international conference, ICIAR 2020, póVoa de varzim, Portugal, June 24–26, 2020, proceedings, Part I 17. Springer; 2020, p. 356–67.
[8] Wang W, Xu G, Ding W, Huang GY, Li G, Tang J, et al. Representation learning from limited educational data with crowdsourced labels. IEEE Trans Knowl Data Eng 2020;34(6):2886–98.
[9] Auer EM, Mersy G, Marin S, Blaik J, Landers RN. Using machine learning to model trace behavioral data from a game-based assessment. Int J Sel Assess 2022;30(1):82–102.
[10] Chen F, Cui Y, Chu M-W. Utilizing game analytics to inform and validate digital game-based assessment with evidence-centered game design: A case study. Int J Artif Intell Educ 2020;30:481–503.
[11] Zhang Y, Wang Y, Zhang H, Zhu B, Chen S, Zhang D. Onelabeler: A flexible system for building data labeling tools. In: Proceedings of the 2022 CHI conference on human factors in computing systems. 2022, p. 1–22.
[12] Riegler M, Lux M, Charvillat V, Carlier A, Vliegendhart R, Larson M. Videojot: A multifunctional video annotation tool. In: Proceedings of international conference on multimedia retrieval. 2014, p. 534–7.
[13] Palotai Z, Láng M, Sárkány A, Tősér Z, Sonntag D, Toyama T, et al. Label-Movie: Semi-supervised machine annotation tool with quality assurance and crowd-sourcing options for videos. In: 2014 12th international workshop on content-based multimedia indexing. CBMI, IEEE; 2014, p. 1–4.
[14] Philbrick KA, Weston AD, Akkus Z, Kline TL, Korfiatis P, Sakinis T, et al. RIL-contour: A medical imaging dataset annotation tool for and with deep learning. J Digit Imaging 2019;32:571–81.
[15] Chen S, Ahmmed S, Lal K, Deming C. Django web development framework: Powering the modern web. Am J Trade Policy 2020;7(3):99–106.
[16] Gaffney KP, Prammer M, Brasfield L, Hipp DR, Kennedy D, Patel JM. Sqlite: past, present, and future. Proc VLDB Endow 2022;15(12).
[17] Arcade ME. Shadowspect. 2020, https://education.mit.edu/project/shadowspect/. [Online; Last Accessed 15 September 2023].
[18] Purwanto A. Education research quantitative analysis for little respondents: comparing of lisrel, tetrad, GSCA, amos, SmartPLS, WarpPLS, and SPSS. J Studi Guru Dan Pembelajaran 2021;4(2).
[19] Slater S, Joksimović S, Kovanovic V, Baker RS, Gasevic D. Tools for educational data mining: A review. J Educ Behav Stat 2017;42(1):85–106.