Realizing a Mobile Multimodal Platform for Serious Games Analytics

Laila Shoukry, Stefan Göbel

KOM - Technical University of Darmstadt
firstname.lastname@kom.tu-darmstadt.de

Abstract

This paper presents the design and development of "StoryPlay Multimodal", a mobile multimodal analytics platform for the evaluation of Serious Games. It is intended to serve researchers, teachers and educational game developers as a means to assess their Serious Game Design. This is done by capturing, pre-processing, synchronizing and visualizing multimodal serious games analytics and mobile sensor data from playtesting sessions. By linking log data with multimodal data, it is possible to uncover relations between design elements, gameplay interactions, context parameters and affective and cognitive states. This is crucial for gaining full insight into a session, even if not present with the player at the same location. After discussing design requirements, the architecture of the software, the different modules, additional features, implementation challenges and solutions are presented. Testing settings, participants and results are also discussed to demonstrate how an evaluation procedure can be applied to deliver valuable outcomes for Serious Games Research.

Keywords: Serious Games, Evaluation, Game-Based Learning, Learning Analytics

1 Introduction and Motivation

Evaluating the design of a serious game needs investigating it from many different dimensions. It is true that the effectiveness of achieving a learning outcome is the major evaluation goal of any learning platform. However, specifically for educational games, the fun gaming aspect as well as the user experience play an equally important role in design assessment [1, 2]. This is due to the fact that the decision of creating an educational game instead of using traditional methods of delivery is based on an assumption that this would increase learner's engagement and motivation. Especially younger children cannot be forced to use such an application as part of a curriculum, but are usually playing such games at home as a means of practice for a certain subject. Thus the amount of time they spend using such a game, preferring it to other activities, will depend on the positive experience they have during interaction. The Reasons and Responses Model [3] describes the different aspects of a Serious Game which are to be evaluated and classifies the following four dimensions: Learning, Gaming, Using and Context. Under each of these dimensions, different requirements need to be met to increase the quality of a Serious Game. These evaluation aspects are further categorized into two categories: features that can be evaluated without real users and others which need testing sessions with participants to be investigated, as they depend on traits, abilities and preferences of different users. These include aspects which need capturing and analyzing affective and cognitive states as well as context parameters. Evaluating these latter aspects is difficult when relying only on logging mechanisms or traditional testing sessions, due to their multimodal nature along with the fast and highly interactive nature of gameplay [4–6]. This is why more and more studies are using multimodal data (video, audio, screen capture, physiological sensors, mobile sensors, ..) for evaluation [7].

Although remote testing in the field offers many chances for Serious Games evaluation which motivated this research, tests in real environments suffer from dynamic factors affecting recorded data quality such as variations of illumination, exposure, orientation, distance, stability and noise. Such factors are more difficult to control than in laboratory settings and result in missing or noisy data [6].

Serious Game researchers and developers (also teachers) with limited time and budget for their evaluation and who need a theoretical framework and user-friendly platform for multimodal evaluation are the main target audience of our platform.

2 Related Work

Capturing multimodal data unobtrusively in a naturalistic setting is essential for getting "real" interaction, affective and context data. As this is difficult to achieve in traditional lab settings, many tools supporting this process are nowadays used for testing sessions, especially remote testing of mobile applications. While there are a number of such tools available for testing in general (see a comparison of features in [7]), none of them is especially tailored for testing mobile educational games. This was the motivation behind the development of StoryPlay Multimodal, an evaluation platform for Serious Games which helps in capturing, preprocessing and visualizing multimodal gameplay data for researchers and educational game designers.

GLEANER or "Game LEarning Analytics for Educational Research" is an evluation tool intended to support serious games research [8] by helping define the data capturing and analysis mechanisms. Data collected in GLEANER include generic and game-specific traces. Another recent framework for game learning analytics built in the Unity environment is called Unity Logger [9].

In the field of entertainment games, there are some evlauation tools which support multimodal capturing [10, 11] including Microsoft's TRUE system [12]. The latter collects gameplay, video and self-report data and visualizes metrics to help developers improve game mechanics. An annotation tool described in [13] helps detect physiological reactions to gameplay by logging in-game events, annotating physiological responses and synchronizing their readings with gameplay session videos. A similar tool presented in [14] combines logging, video, physiological data and self-reports. Similarly, [4] describes an approach visualizing the player's path in the game along with physiological measurements. LAIF [15] is another tool for game user research which records eye tracking data and relates it with game-play logging.

Apart from the gaming field, there are tools enabling data collection and interaction analysis in general which support and synchronize more than one modality. Chronoviz [16] offers visualization and interactive navigation of multimodal data streams of interaction testing sessions. These include video and audio files, computer logs, sensor readings, paper notes, and transcriptions. Similarly, Replayer [17] is a cross-platform platform offering synchronization of recorded data from diverse sources.

As for mobile games, Playtestcloud (playtestcloud.com) is an online playtesting platform which offers a software which wraps around a mobile game to equip it with screen and touch recording features without the need to modify the game or to integrate an SDK. They offer acces to playtesters, who, before they start the game, will see screens with instructions that walk them through tasks they have to accomplish for the playtest and prevent them from launching the game after the playtest has concluded. The software will record the screen contents of the app, all touch gestures and the microphone input.

A recent research paper [18] which came to our attention after the initial comparison of tools in [7] describes a tool called Vixen for interaction visualization of gameplay experiences. This tool might be the closest approach to our platform but is intended for general games created with unity and not especially for Serious Games as there is no learning analytics element to it. In addition, it does not run on mobile devices.

Thus, to the best of our knowledge, there is no Serious Games evaluation tool which helps in synchronizing and/or analysing multimodal information.

The initial StoryPlay is a tool for collecting and visualizing learning and gameplay traces based on the Narrative Game-Based Learning Object (NGLOB) model [19] with which the Serious Games authoring environment StoryTec was created. This model represents a player's learning competencies, his/her player type as well as the narrative stage reached in the Serious Game to help online monitoring and adaptation of the three aspects. Thus a serious game (or a "story") created with StoryTec can be tested in real-time or offline using StoryPlay, whereby the researcher, instructor and/or game designer can view a replay of the gaming session alongside visualizations of any updates in the model and variables as well as aggregate traces of multiple players. In StoryPlay Multimodal (StoryPlayMM), we aimed to extend it to include all important features found in available platforms (discussed in more detail in [7]) into one Serious Games Analytics Platform.

3 Design Requirements and Architecture

The design goal of StoryPlay Multimodal (StoryPlayMM) is creating a non-invasive Serious Games research/evaluation tool supporting remote, asynchronous observational evaluation of mobile serious games. Main requirements for the framework were determined from literature and software review [20] as well as recommendations of Serious Games researchers.

3.1 Underlying Architecture

The StoryTec Authoring Environment is built upon an internal model considering updates in the learner and player model during play in addition to the storyline [21]. The original StoryPlay rapid prototyping tool is based on the same story engine (see Figure 1) and also displays updates in the internal models [22]. This data is gathered based on the Narrative Game-Based Learning Object (NGLOB) Model [19]. The player used for running games created with StoryTec, called StoryPublish, allows running the game interface on different platforms [23]. The story structure is formatted using the xml-based model description language ICML [19] and communicated between the authoring tool and the Story Engine. This same information is used for reconstructing the sessions in StoryPlay using StoryPublish.

3.2 Design Goals and Requirements

The goal of this work is to extend StoryPlay to support multimodal data and link it with event logging and internal model updates while minimizing invasiveness. This would be helpful for authors/researchers/instructors (here StoryTec users) to offer them a ready means for getting feedback to improve the design of their serious game. To minimize obtrusiveness, the prototypical implementation is developed and tested on smartphones where the sensing mechanism is regarded to be far less obtrusive than sensors which are worn on the body or fixed inside labs. This also allows carrying out evaluations by play-testers worldwide without having to be present in the same place.

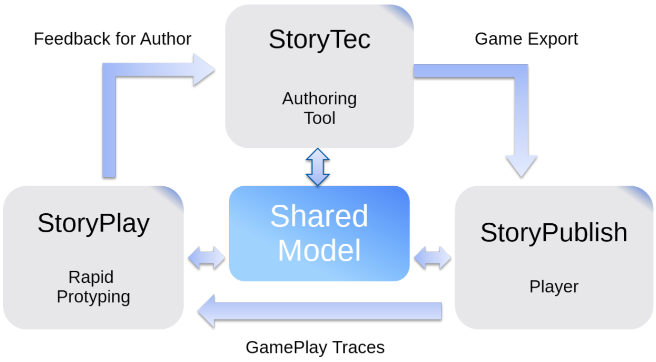


Figure 1: *StoryTec Architecture*

Using mobile sensors also gives the possibility of providing an insight into a wider set of context parameters as context plays an integral role in playtesting nowadays due to its influence on experience. The hypothesis is that an easy and goal-oriented navigation through multimodal data would help researchers disambiguate ambiguous actions in the event log. The ultimate goal is to allow researchers to better understand relationships among data and provide them with additional information from natural settings. For instance, finding out if a pause in gameplay activity is due to experiencing frustration, reflecting on playing strategy or learning content or getting distracted by the environment can be important for evaluation (Other examples of such ambiguities can be found in the applications of Reasons and Responses Model described in [3]).

Relying on data automatically captured during the gameplay experience should help make evaluation studies more objective and time-efficient than subjective observation and self-reports for uncovering aspects related to emotion and cognition. Combined with logging, this should help identifying advantages and problems with specific game elements with regard to fun, usability and effectiveness and how to improve the Serious Game at hand thus helping advance Serious Games research.

The requirements for the development of StoryPlayMM arose from a real practical need identified by the Serious Games research group in various research projects and not just on theoretical assumptions. As mentioned earlier, it was a development objective to integrate the most useful features from different platforms into one platform tailored for Serious Games. Features considered for design are gathered from available research and commercial tools discussed in [7] and adjusted to be used with scene-based Serious Games authored with StoryTec. The tool should be usable without prior programming skills to make it usable for all StoryTec target users. It should also have potential for integrating different recognition modules to act on the analysis of the captured data.

As in the PlaytestCloud tester app mentioned earlier, StoryPlayMM has a similar approach in recording log files on mobile devices, as well as front-facing camera video and/or microphone recordings. Audio recordings can be used for think-aloud, laughter detection etc. - and detecting the environment background noise.

The use of a session video view with adjustable speed is desirable for investigating in-game action in parallel with their responses [13]. In addition, allowing the user to quickly skip through a video to jump to a certain event or a certain reaction by clicking on this particular interest point was found to be a very useful feature [13]. In StoryPlayMM, this is achieved not by recording screen video which would be big in size along with the front-facing camera video and not easy to navigate to a certain event. It is achieved by a session replay tool which reconstructs gameplay from the log files using the game engine itsself, as will be discussed in next sections. A replay of the whole gameplay session is reconstructed from the logged information by embedding the StoryPublish player in addition to enabling adjustable speed and interactive navigation of the replay based on main events. The navigation feature helps in speeding up the process of evaluation as researchers can directly navigate to the specific event of interest without having to watch the whole session.

A close coupling with StoryTec authoring (.icml) files allows saving much space in the replay component files. This tight integration allows accessing and showing internal state information along with the interface itself to assist in correlating game activity with leaner experience. With this replay, there is no need for large and difficult to navigate files containing screen recordings.

The feature of recording and replaying face expressions of game players is presented as having many advantages for evaluation: It is an unobtrusive way of observing players' engagement and involuntary reactions during playtesting [18, 24]. Recognition and Analysis of these expressions can also be applied on the recordings as a quantitative assessment method [25].

For the logging, main recorded actions from the player side and events from the system side need to be distinguished and represented with the possibility to jump to related multimodal data where needed. In addition, mobile sensor information like illumination and movement can help determine context as discussed in [3]. Location sensors have been used in some multimodal interaction analysis tools supporting mobile deployment to track the location of the testers during a remote testing session [26]. This is useful when having playtesters from different places and a need to gather location data and was thus also included in StoryPlay Multimodal.

The synchronization and interactive navigation of multimodal data is applied in many multimodal data analysis tools like ChronoViz [16], Tatiana [27], Digital Replay System [28], Noldus Observer [29] and Mangold Interact (mangold-international.com). It enables researchers to jump to the point of interest and see all related multimodal data run simultaneously next to each other which saves time and effort of analyzing and annotating qualitative data.

The feature of filtering out uninteresting video frames using information from low power mobile sensors like the illumination sensor was described in [30]. They also investigated predicting whether a frame contains faces using the accelerometer and gyroscope sensors. These features have been included in StoryPlay as well and this was one of the reasons why mobile sensor data is used in the framework. The other reason is that it also provides data on how the player is holding the device which is also important for UX testing.

Built-in measures of self-reporting, testing instructions and after-game survey can be integrated to investigate correlations with observed interactions and reactions. Eye-tracking and Physiological data are not directly included in design considerations, but can be extended if there is a non-obtrusive way of monitoring using built-in mobile sensors. It is possible for users and developers to select data sources to avoid privacy issues. Player profiles can be created or, if permitted, can be collected from the phone.

Session summarizations include aggregations and a future feature would be detecting and highlighting relevant data such as correlations, repetitions and significant events and sequences. Detecting event sequences may be helpful for comparing the input stream with some target sequence (e.g. of an expert player) or emphasizing certain pre-defined player behavior.

Aggregations can be done across users for extensive data (e.g. scenes where most users had a sudden movement or smile or laughter etc.) or across modalities for one and the same user for intensive data (when more than one data source has significant change at the same time) [2]. Abstracting low-level details for investigators helps speed up the evaluation process and aggregations can help in rating subgames as well as identifying common paths and interaction patterns. The hybrid approach where analysis is not fully automated and not fully manual is chosen for our purpose as it is difficult to fully rely on automatic recognition where experts can more efficiently get better results [31], especially because of the heterogeneity of Serious Games. These experts, however, still need the pre-processing to save time and have objective measures. There is a potential for supporting the integration of off-the-shelf recognition modules, e.g. for facial features. A way for supporting user-defined annotations should also be included.

In summary, the most important features for the platform can be summarized into the following and described in [3]:

- supporting the dimensions of learning, gaming and interaction
- recording AND playback of multimodal data
- logging or screen recording of interaction
- supporting different modalities like video, audio, eye-tracking, physiological and mobile sensors
- synchronization of multimodal data
- ituitive interactive navigation
- filtering and preprocessing of data
- a means of annotating data
- data analytics and visualizations

An overview of the architecture and design goals is given in Figure 2.

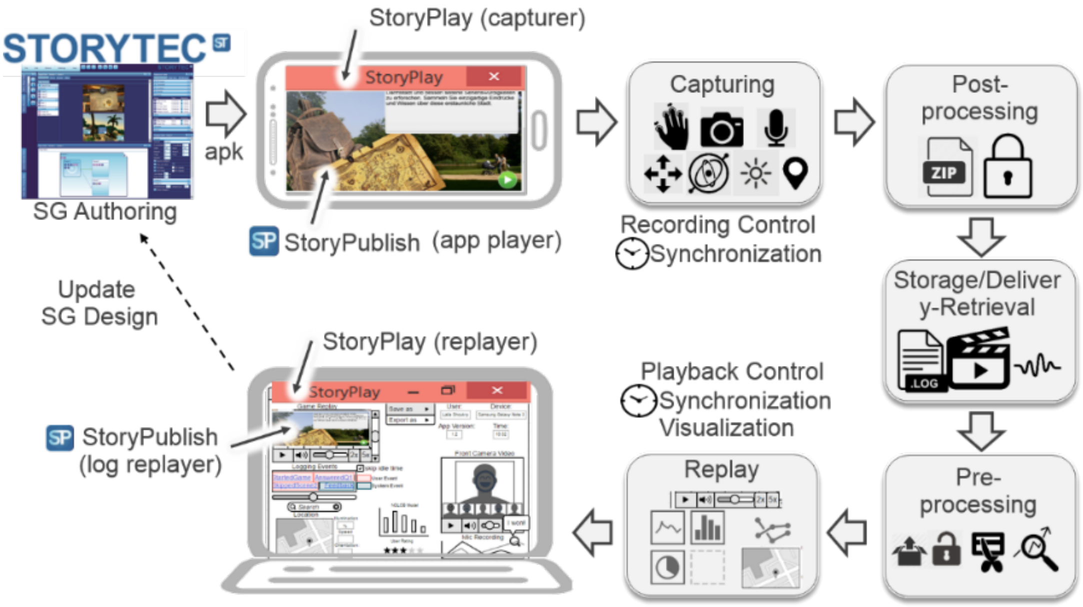


Figure 2: *Multimodal Evaluation Platform Design Architecture [7]*

Two different modules are needed for realizing this tool: a capturing app which is run with the application to capture logs and multimodal data (here a mobile app as a prototypical implementation of a naturalistic testing session in the field) and the main desktop analysis tool for replaying and analysing data.

4 Design and Implementation

StoryPlay Multimodal platform design consists of two main modules: The **Capturer** app and the desktop **Replayer** component. A mockup of the tool with the features described in the previous section is depicted in Figure 2. In addition to those two components, an **Observer** app was developed for testing sessions where an observer is present to help him record his/her session observations in real-time, which are then also shown later according to their timestamps in the session replay. In this section the basic features implemented in the different modules are described in more detail. Some interfaces of the modules are shown in Figures 3-9.

4.1 Capturer App

The Capturer mobile application was integrated into mobile educational games exported with StoryTec [21, 23], a Serious Games Authoring Environment used for creating scene-based educational games based on the NGLOB (Narrative Game-Based Learning Object) Model [19]. To combine the advantages of event data and the more rich observational data, it was a main requirement of the project to provide the investigator with a way to review all segments of the session quickly without having to manually search through the entire data as well as get help and summary statistics. All events can be used to navigate in the session by skipping to the timestamp where this event occurred. Interesting parts of the video are also highlighted as we will see later. Ways of seeing where a new scene started and which notes have been provided by the observer on a particular scene. Also mobile sensor data are visualized and their timelines are also synchronized with the timelines of the events, observer events, game and video, all playing together.

Based on design requirements identified in [3], the following are the major features supported on the capturer app:

- Synchronous recording of video (from mobile front-facing camera), event logs and mobile sensor events with flexible user configuration options.
- Integration inside StoryPublish Android software (used for running StoryTec games on different platforms), run on different Android devices
- Unobtrusive, not negatively affecting game experience
- Storing and retrieving user profiles on device
- Avoiding privacy, storage and bandwidth issues by giving user full control over sensor activation and over what and when to store and/or upload to the server
- Interface usable without programming or special background
- Providing a game rating option after gameplay

The capturer app (some screenshots are shown in Figure 3) is an extension around the mobile version of the StoryPublish engine which wraps the game to be tested after it is exported to Android from StoryTec Authoring Software. This wrapper is implemented in haxe based on kha engine. The capturer contains the options for recording user, log and multimodal data during interaction with the mobile edugame. When running the app, the player first enters his google drive log-in data if he wishes, if not it is set to the default firebase server [1] which offers cloud storage for app data. After setting the user profile for testing (name, age, gender

[1] http://www.firebase.google.com

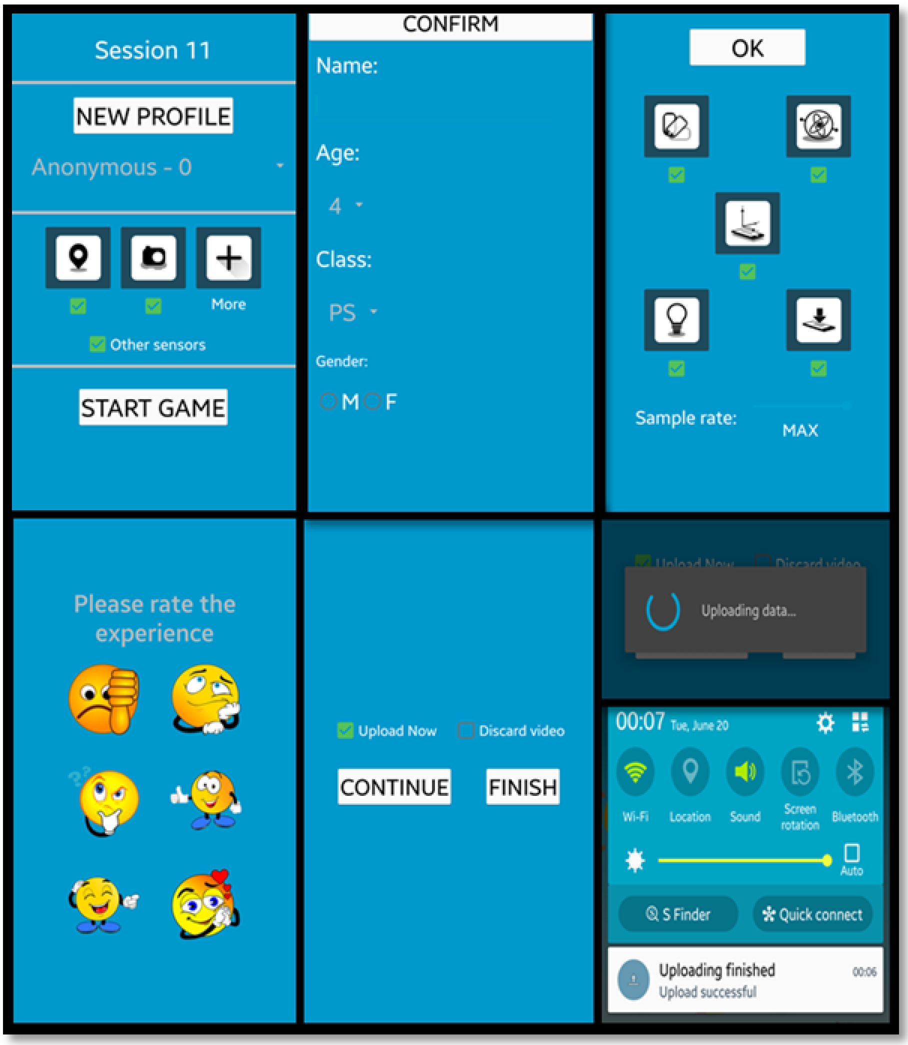
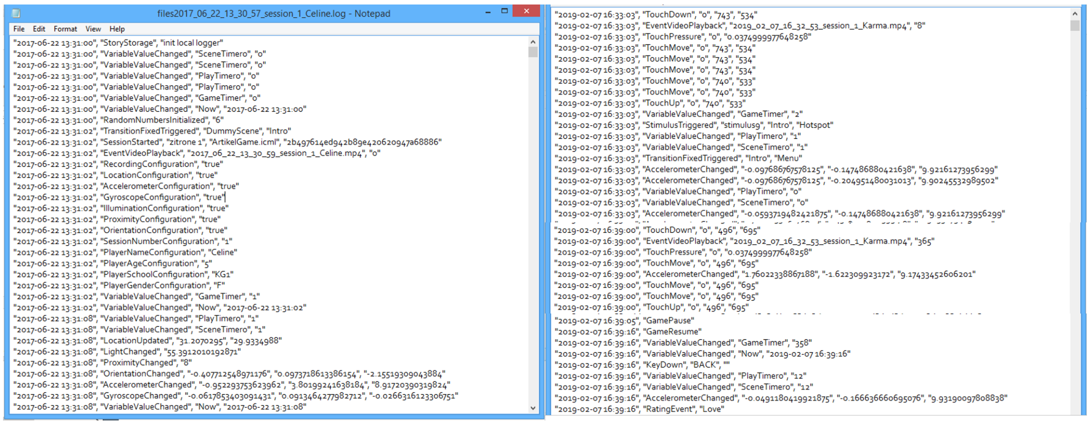


Figure 3: *Mobile Capturer Component Screens before and after GamePlay*

and class), the user chooses which sensor data are recorded and will have the option at the end to give permission on what is sent, what is saved on the device and what is deleted. S/he can check any or all of the following sensors: location, camera, accelerometer, gyroscope, proximity and illumination using icons and descriptions for the different sensors. S/he can also choose the sample rate of some sensors to be normal, high or very high using a slidebar. In the case that the researcher is co-located with the tester, these options can be chosen by the researcher. Also if it is a child testing remotely, this part can be done by his parent. During gameplay, the sensor data specified is logged and if given permission, the front-facing camera captures a video of the participant's face during play. The recording part is implemented mostly in java. At the end of the gaming session, a rating scene is shown where the player rates his experience.

Log files saved on the mobile phone (to be sent later to the server) contain timestamped logged events from the game like clicks, variable changes and scene names in addition to events needed for synchronization and statistics like timers as well as events recording sensor changes. In addition, some events were added for mobile gaming like game pause and game resume to account for interruptions in game play. Figure 4 shows some excerpts from generated game session log files from the capturer app.



files2017_06_22_13_30_57_session_1_Celine.log - Notepad
File Edit Format View Help

```
"2017-06-22 13:31:00", "StoryStorage", "init local logger"
"2017-06-22 13:31:00", "VariableValueChanged", "SceneTimer0", "0"
"2017-06-22 13:31:00", "VariableValueChanged", "SceneTimer0", "0"
"2017-06-22 13:31:00", "VariableValueChanged", "PlayTimer0", "0"
"2017-06-22 13:31:00", "VariableValueChanged", "PlayTimer0", "0"
"2017-06-22 13:31:00", "VariableValueChanged", "GameTimer", "0"
"2017-06-22 13:31:00", "VariableValueChanged", "Now", "2017-06-22 13:31:00"
"2017-06-22 13:31:00", "RandomNumbersInitialized", "6"
"2017-06-22 13:31:02", "TransitionFixedTriggered", "DummyScene", "Intro"
"2017-06-22 13:31:02", "SessionStarted", "zitrone 1", "ArtikelGame.icml", "2b497614ed942b89e420620947a68886"
"2017-06-22 13:31:02", "EventVideoPlayback", "2017_06_22_13_30_59_session_1_Celine.mp4", "0"
"2017-06-22 13:31:02", "RecordingConfiguration", "true"
"2017-06-22 13:31:02", "LocationConfiguration", "true"
"2017-06-22 13:31:02", "AccelerometerConfiguration", "true"
"2017-06-22 13:31:02", "GyroscopeConfiguration", "true"
"2017-06-22 13:31:02", "IlluminationConfiguration", "true"
"2017-06-22 13:31:02", "ProximityConfiguration", "true"
"2017-06-22 13:31:02", "OrientationConfiguration", "true"
"2017-06-22 13:31:02", "SessionNumberConfiguration", "1"
"2017-06-22 13:31:02", "PlayerNameConfiguration", "Celine"
"2017-06-22 13:31:02", "PlayerAgeConfiguration", "5"
"2017-06-22 13:31:02", "PlayerSchoolConfiguration", "KG1"
"2017-06-22 13:31:02", "PlayerGenderConfiguration", "F"
"2017-06-22 13:31:02", "VariableValueChanged", "GameTimer", "1"
"2017-06-22 13:31:02", "VariableValueChanged", "Now", "2017-06-22 13:31:02"
"2017-06-22 13:31:08", "VariableValueChanged", "PlayTimer0", "1"
"2017-06-22 13:31:08", "VariableValueChanged", "SceneTimer0", "1"
"2017-06-22 13:31:08", "LocationUpdated", "31.2070295", "29.9334988"
"2017-06-22 13:31:08", "LightChanged", "55.3912010192871"
"2017-06-22 13:31:08", "ProximityChanged", "8"
"2017-06-22 13:31:08", "OrientationChanged", "-0.407712548971176", "0.0973718613386154", "-2.15519309043884"
"2017-06-22 13:31:08", "AccelerometerChanged", "-0.952293753623962", "3.80199241638184", "8.91720390319824"
"2017-06-22 13:31:08", "GyroscopeChanged", "-0.0617853403091431", "0.0913464277982712", "-0.0266316123306751"
"2017-06-22 13:31:08", "VariableValueChanged", "Now", "2017-06-22 13:31:08"
```

```
"2019-02-07 16:33:03", "TouchDown", "0", "743", "534"
"2019-02-07 16:33:03", "EventVideoPlayback", "2019_02_07_16_32_53_session_1_Karma.mp4", "8"
"2019-02-07 16:33:03", "TouchPressure", "0", "0.0374999977648258"
"2019-02-07 16:33:03", "TouchMove", "0", "743", "534"
"2019-02-07 16:33:03", "TouchMove", "0", "743", "534"
"2019-02-07 16:33:03", "TouchMove", "0", "743", "534"
"2019-02-07 16:33:03", "TouchMove", "0", "740", "533"
"2019-02-07 16:33:03", "TouchMove", "0", "740", "533"
"2019-02-07 16:33:03", "TouchUp", "0", "740", "533"
"2019-02-07 16:33:03", "VariableValueChanged", "GameTimer", "2"
"2019-02-07 16:33:03", "StimulusTriggered", "stimulus9", "Intro", "Hotspot"
"2019-02-07 16:33:03", "VariableValueChanged", "PlayTimer0", "1"
"2019-02-07 16:33:03", "VariableValueChanged", "SceneTimer0", "1"
"2019-02-07 16:33:03", "TransitionFixedTriggered", "Intro", "Menu"
"2019-02-07 16:33:03", "AccelerometerChanged", "-0.09768676757578125", "-0.147486880421638", "9.92161273956299"
"2019-02-07 16:33:03", "AccelerometerChanged", "-0.09768676757578125", "-0.204951480031013", "9.90245532989502"
"2019-02-07 16:33:03", "VariableValueChanged", "PlayTimer0", "0"
"2019-02-07 16:33:03", "VariableValueChanged", "SceneTimer0", "0"
"2019-02-07 16:33:03", "AccelerometerChanged", "-0.0593719482421875", "-0.147486880421638", "9.92161273956299"
"2019-02-07 16:39:00", "TouchDown", "0", "496", "695"
"2019-02-07 16:39:00", "EventVideoPlayback", "2019_02_07_16_32_53_session_1_Karma.mp4", "365"
"2019-02-07 16:39:00", "TouchPressure", "0", "0.0374999977648258"
"2019-02-07 16:39:00", "TouchMove", "0", "496", "695"
"2019-02-07 16:39:00", "AccelerometerChanged", "1.76022338867188", "-1.622309923172", "9.17433452606201"
"2019-02-07 16:39:00", "TouchMove", "0", "496", "695"
"2019-02-07 16:39:00", "TouchMove", "0", "496", "695"
"2019-02-07 16:39:00", "TouchUp", "0", "496", "695"
"2019-02-07 16:39:05", "GamePause"
"2019-02-07 16:39:16", "GameResume"
"2019-02-07 16:39:16", "VariableValueChanged", "GameTimer", "358"
"2019-02-07 16:39:16", "VariableValueChanged", "Now", "2019-02-07 16:39:16"
"2019-02-07 16:39:16", "KeyDown", "BACK", ""
"2019-02-07 16:39:16", "VariableValueChanged", "PlayTimer0", "12"
"2019-02-07 16:39:16", "VariableValueChanged", "SceneTimer0", "12"
"2019-02-07 16:39:16", "AccelerometerChanged", "-0.0491180419921875", "-0.166636660695076", "9.9319009780838"
"2019-02-07 16:39:16", "RatingEvent", "Love"
```

Figure 4: *Excerpts from Log Files Generated by the Capturer App*

4.2 Observer App

The observer app was initially not part of the design but was then found to be important for several reasons: First, it was found to be a good addition to the evaluation suite, as it can be used in observational studies where the researcher is co-located. Secondly, it can help in training a machine learning algorithm for extracting features from multimodal data by serving as a ground truth. Lastly it helps investigating relationships between log events and affective, cognitive and context states assuming that we have already extracted the given features from multimodal data which can be a great help to advance Serious Games research.

Taking notes during a session by the observer can be so time consuming that an important observation can be missed while the observer is still writing. So the main goal of the interface is to make it very easy to record observations with just one click during the usually fast and unpredictable playtesting session. The design and choice of the observation recording buttons and types is based on the LeGUC Features for Evaluating Experience in Serious Games Playtesting described in [20] and depicted in Figure 5.

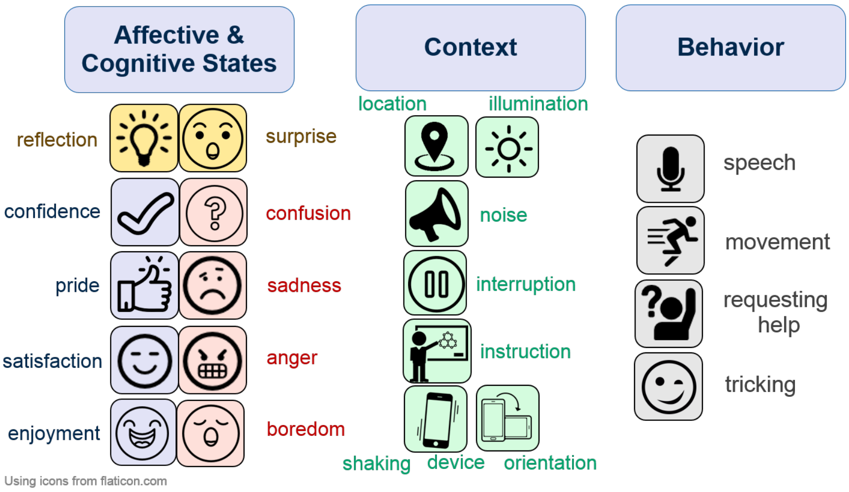
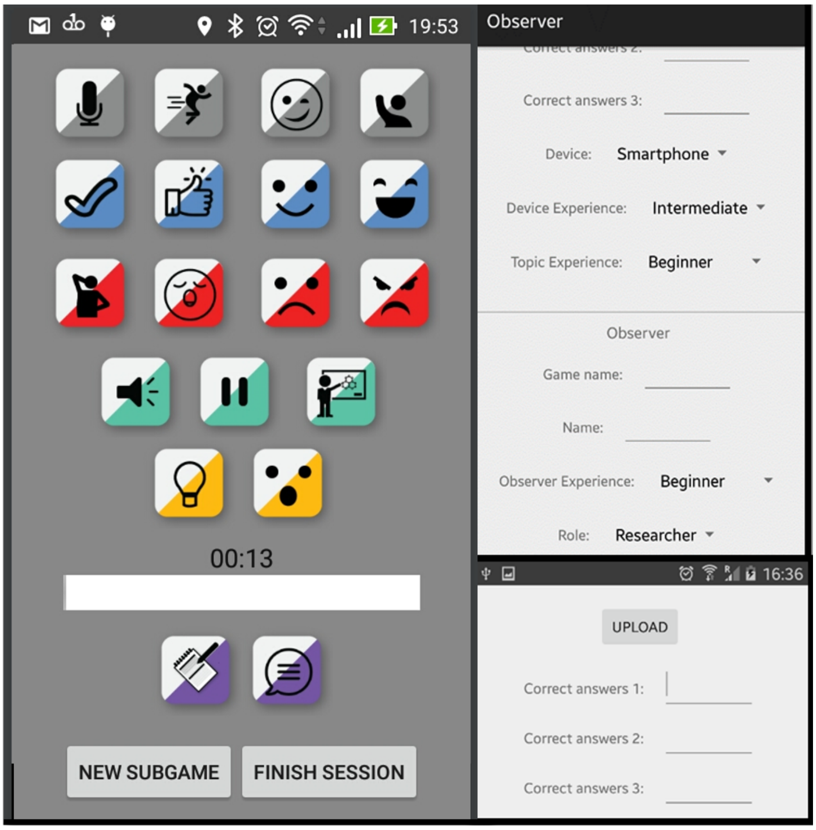


Figure 5: *LeGUC Features for Evaluating Experience in Serious Games Playtesting [20]*

To the best of our knowledge, this is the first observation app considering features related to Serious Games Evaluation. The current design after several iterations and tests can be seen in Figure 6.

The following are the main features of this component. It is implemented in Java for



Observer

Correct answers 2:

Correct answers 3:

Device: Smartphone

Device Experience: Intermediate

Topic Experience: Beginner

Observer

Game name:

Name:

Observer Experience: Beginner

Role: Researcher

Correct answers 1:

Correct answers 2:

Correct answers 3:

Figure 6: *StoryPlayMM Observer App Screens*

Android phones and tablets.

- Fast access with saved timestamps to reduce delay and facilitate synchronization
- Features based on LeGUC features described in [20]
- Can help in training machine learning modules for recognition from real MM data
- Help answer research questions by assuming features are already extracted from MM data

After starting a new session (numbered automatically on each device) and setting a server log-in account, information about the observer like his role (being a researcher, teacher or an educational game developer) and his experience with the observer app are entered. In addition, background information about the player and the session is entered, like the game name and the session ID generated on the other device which has the game running on the capturer app for later synchronization purposes. Some fields for pre-test correct answers numbers are also available if the observer wants to ask the player before starting the actual playing session to compare with post-tests, also administered in the same way. Once the session on the other device has started, the observer starts the session on his device and clicks on the different observations or writes notes which are all saved with timestamps relative to the start of the session. The time lag between starting the game session on one device and the observer session on the other device is later compensated in the desktop evaluation platform. This is realized by providing a slider for the researcher to adjust the offset at the beginning of the timeline until the data is aligned with the observations on the video or the game replay.

After starting the session, a screen is shown with many icons, a textfield and some buttons. The observer clicks on the icons when s/he observes a certain behavior, context or reaction which happens during the testing. For example, the grey icons stand for the tester doing one of those behaviors: speaking, moving, tricking, asking for help. The blue icons stand for his positive reactions like being confident while answering a question, being proud of answering correctly, smiling or laughing. The red icons stand for negative reactions of the tester like being confused, bored, sad or angry. The green icons are for context events like noise, interruption, or you offering help to the tester and explaining something in the game. The yellow icons stand for neutral reactions like reflection and surprise. The observer can use the textfield for two things: taking a note about something (e.g. a bug) and then clicking the left purple button, or writing something that the player said and then clicking the right purple button. The new sub-game button can be used when a player goes to the main menu and choose a new subgame. When the player wishes to finish the session, the observer click on Finish session enters data about post-test if s/he wishes and all the recorded data is saved in an observer log along with its timestamp on the device to be sent to the server.

4.3 Replayer Component

The Replayer desktop component is the main tool used by the researcher for evaluation. It runs on Windows operating system and its interface is divided into different tabs presenting different data about the sessions. These views are placed in tabs which can be toggled to allow for adjusting the level of detail for each analysis task by expanding the corresponding area. All tabs can be toggled based on researchers needs to avoid overloading the program and the screen when some parts are not needed.

The following are its main features:

- Synchronous playback of game events, user video, event log, mobile sensor data, observer logs, profile, session data and model updates (NGLOB narrative, gaming and learning models)
- Availability and coherence of replay controls
- Coordinated Interactive Navigation of all data based on scenes and events
- Summary Statistics, Pre-processing and Visualization
- Usable by Non-Programmers
- Modular design to accommodate off-the-shelf detectors or researcher-specified rules

The desktop analysis tool is implemented in C# and its interface is built using Windows Presentation Foundation (WPF). Some views of the StoryPlay interface are shown in Figures 7-11.

Importing Session Files After session files were uploaded from the researcher's mobile phone to their own google drive, s/he can log in with his/her account and download session files belonging to him/her. If the app was sent to be tested remotely from participants (after their agreement to send data and choosing which data to send), the data (log files, videos and/or mobile sensing data) is sent to the firebase server to the researcher's account. The observer app files are also sent from the researcher's device and collected on the server. The session ID helps to match observer files with game log files and videos. These files can be imported in the desktop evaluation component on the researcher's computer by clicking sync or by directly downloading and importing the files. Files belonging to the same session are identified by the session ID in the file name and grouped together for investigation.

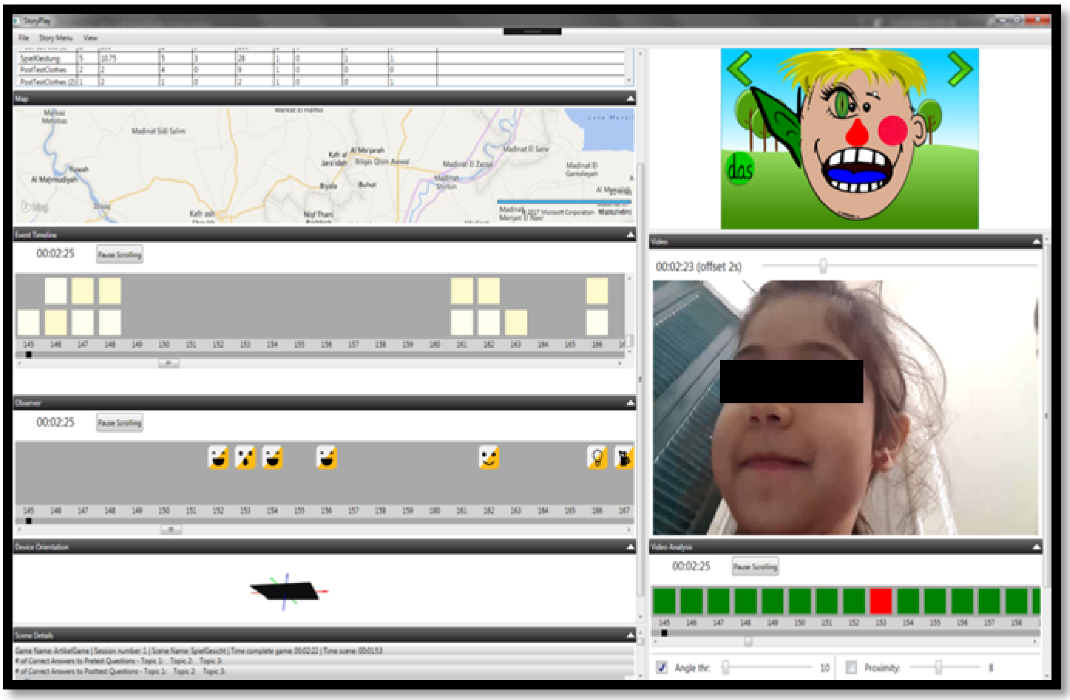
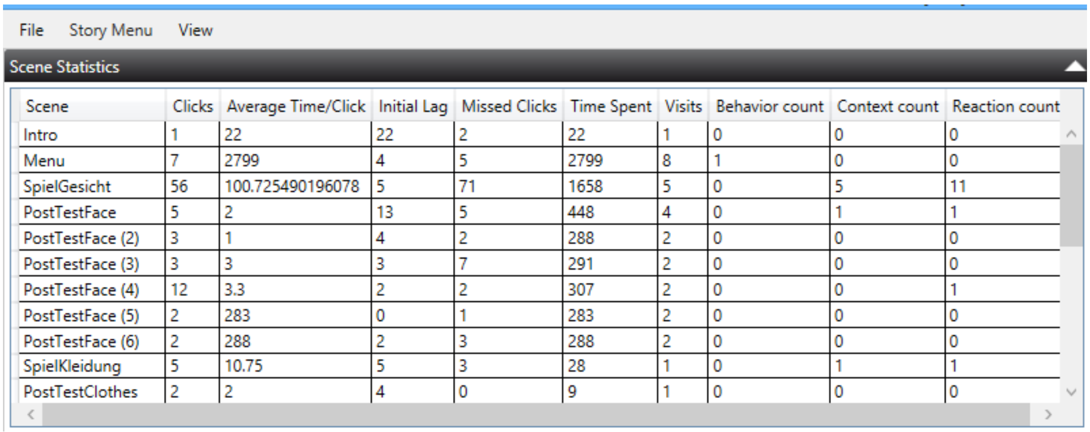


Figure 7: *StoryPlayMM Replayer Component*

Scene Information and Statistics After the session files are loaded, the different tabs show details about the session. In addition to a tab with general session information like game name, session number, total time on the game, current scene (during replay) and time on the current scene, there is a separate tab with more detailed statistics relevant for research. These are gathered from the log data and displayed in a separate tab with the option to export the data to .xls extension. These data include the game scenes visited in their respective order, the number of clicks in each scene, the average time per click in each scene, the initial lag before clicks, the missed clicks (clicking on parts of the screen which provide no action), the total time spent on each scene (all visits of the scene accumulated), the number of visits and the number of observations recorded by the observer app on this scene categorized in behavior, context and reaction observations. Examples of a session's gameplay statistics are depicted in Figure 8. Examples of how these statistics were used in session evaluation will be discussed in the next section.



Scene	Clicks	Average Time/Click	Initial Lag	Missed Clicks	Time Spent	Visits	Behavior count	Context count	Reaction count
Intro	1	22	22	2	22	1	0	0	0
Menu	7	2799	4	5	2799	8	1	0	0
SpielGesicht	56	100.725490196078	5	71	1658	5	0	5	11
PostTestFace	5	2	13	5	448	4	0	1	1
PostTestFace (2)	3	1	4	2	288	2	0	0	0
PostTestFace (3)	3	3	3	7	291	2	0	0	0
PostTestFace (4)	12	3.3	2	2	307	2	0	0	1
PostTestFace (5)	2	283	0	1	283	2	0	0	0
PostTestFace (6)	2	288	2	3	288	2	0	0	0
SpielKleidung	5	10.75	5	3	28	1	0	1	1
PostTestClothes	2	2	4	0	9	1	0	0	0

Figure 8: *Examples of Log Statistics*

Session Replay Tab In the Replay area, all interactions are reconstructed from parsed logged data using the StoryPublish serious game engine by parsing the .icml file of the tested game and combining both. This is why the path for the game files has to be chosen at the beginning of the replay. A mouse icon is being displayed at the position of the mouse calculated from mouse movement. This will be extended by changing the color of the mouse icon for different states like mouse clicks. The replay speed can be changed, affecting all other multimodal data replay which runs simultaneously with the in-game events in the different views. One can also choose to jump to a certain event or video frame at any time by clicking on the displayed event icons. As the original log contained all events, the events of interest with meaningful interaction during the game were first identified. The session timestamps where the mouse was just moving around without interaction while playing the game were not considered as meaningful interactions. Thus for skipping a log entry was considered only after a triggered stimuli, i.e. when the user clicked a button to interact with the game.

Events Timeline Tab In the events navigator, the user sees a list of significant events moving with time. This is improved by using color codes for different event types and can later also use icons. The events movement is synchronized with the game replay. These events include game generated events like starting a new game or transitioning between scenes, and user events like clicks, triggered stimulus, game pause and resume (when the application is interrupted by the phone, for example.). When hovering over the events on the timeline, the names of the events are displayed underneath. The user can also click on any of the colored squares representing events to jump to this part in all open views simultaneously for a closer investigation.

Observations Timeline Tab In this tab icon representations of all observer recordings which are parsed from the observer app log files of the chosen session are displayed on a timeline using the same icons of the observer app which corresponds to the LeGUC states depicted in Figure 5. These move in synchronization with the rest of the representations in open tabs according to the timestamps. When hovering over the “notes” icon, one can see the text written by the observer as a note at a certain instant. Other icons represent any logged reactions like smiling, laughing, showing confusion, boredom, sadness, resentfulness, confidence or pride when answering questions, reflecting or being surprised, behaviors like speaking, moving, tricking or asking for help or context events like noise, interruption or offering explanation by the observer.

Session Videos Video clips are displayed in the WPF GUI using an HTML5 tag called media element tag which supports a broad range of media elements to avoid using extra plugins. Microsoft Expression encoder is used to handle the video files. For synchronizing video with replay, video log events were added to the log-file at the appropriate positions. As the original log contained all events, the events of interest with meaningful interaction during the game were first identified. The session timestamps where the mouse was just moving around without interaction while playing the game were not considered as meaningful interactions. Thus, a log entry was added only after a triggered stimuli, i.e. when the user clicked a button to interact with the game. Initially, a special “EventVideoPlayback” log entry was added for replaying specific chunks of the log. This entry has the following format: DateTime (timestamp to write the video events), EventVideoPlayback, PathToSavedFile (path of the captured video file) Offset (for synchronizing the playback of video while playing a specific portion of the log file) and SpeedRatio (playback speed with 1 for normal playback). As in the original replay, a timer is started for all the events in the log to execute them accordingly. To replay a session part between two important events, all events prior to the event selected in the navigator are executed without timer and then the timer is started from this event to the next significant event. The aforementioned offset tag saved in the EventVideoPlayback entry was initially used to determine from where to start playback and when to stop it, later the synchronization was found to be better when using events like clicks for playback rather than these events.

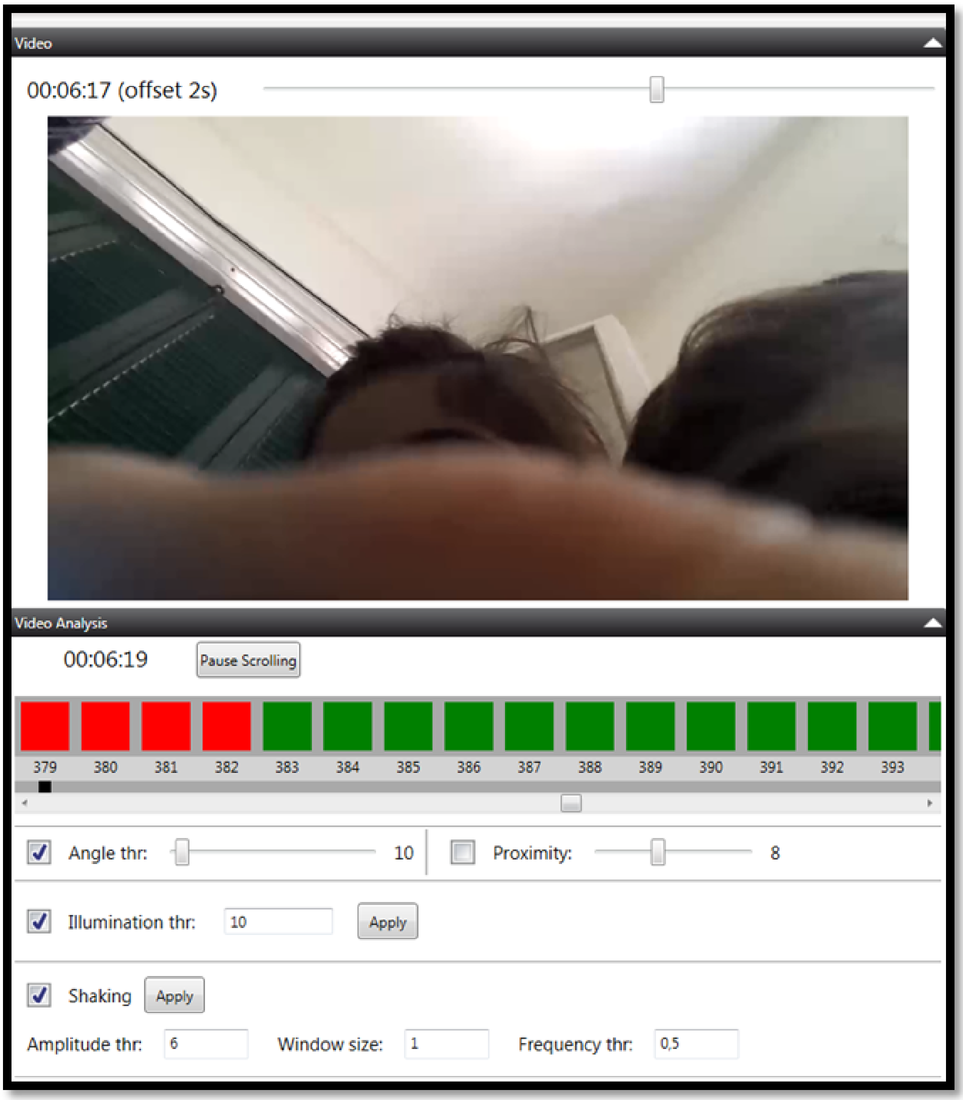
Video Analysis Tab As discussed earlier, some mobile sensors can give a good indicator when to look at multimodal data like video and when the quality might not be good enough or provide important information about the context of the player. By just using low-power sensors like illumination and gyroscope, bad video frames can be discarded without the need for complex recognition algorithms. The video analysis tab is dedicated to adjusting different thresholds of mobile sensor data. Based on these settings video frames are flagged as good or bad depending on context conditions as can be seen in Figure 9. In this figure the testing children put their finger on the smartphone camera and thus covered their faces. Usind the illumination sensor with a threshold of 10 these frames were automatically flagged with red squares meaning that they are not usable and can be discarded. However, these frames are not automatically discarded from the beginning as the objective of this tab is to allow researchers to experiment with the best threshold suitable for their particular experiment conditions. A similar method can be used for discarding video frames where the camera is pointing upwards to the ceiling using the angle threshold. For the shaking of the device, a frequency threshold (for shaking speed), an amplitude threshold (for shaking intensity) and a window size can be chosen to flag frames with considerable shaking. The formula used for using the shaking threshold is as follows:
$if((amp_X/window > thr)||(amp_Y/window > thr)||(amp_Z/window > thr))\, returnfalse;$

where amp is the number of times there was an acceleration bigger than the chosen amplitude in a certain window for each direction, respectively.

After some experiments, the default amplitude was set to 6 and the default window size was set to 1 which gave the best results for testing data. However, the user can freely adjust to his/her own conditions. The color used for good video frames which are more likely usable for evaluation is green. The timeline of these colored squares corresponding to video frames also moves with time in synchronization with the video.

Internal Model Changes Tabs The StoryTec Authoring tool offers game creators the ability to adapt their games by annotating every scene on three dimensions: learner, player and story model (see [32]). Individual learning skills are modeled based on the *Competence based Knowledge Space* [33]. Playing preferences of different players are modelled based on Bartle's four player models described in [34]: killer, achiever, socializer and explorer by numbers in the interval [0,1] to offer a percentage mapping as players usually show features of different models combined together. The story model is based on the Hero's Journey as modified in [19]. These three models are constantly updated during play based on the game author's annotations made in StoryTec to choose a scene which suits the player best.

In StoryPlay, the state changes of the underlying learner and player model as well as important information about the story path in addition to all current values of active variables are communicated using different visualizations with each event and can be used for various evaluation purposes. This is also possible due to the close coupling with the StoryTec authoring tool which is based on the same internal model (NGLOB) and thus can track data based on it. In Addition the History Tab shows the visited scenes of the game in a graphical representation (see Figure 10-11).



00:06:17 (offset 2s)

00:06:19

☑ Angle thr: 10

□ Proximity: 8

☑ Illumination thr: 10

☑ Shaking

Amplitude thr: 6

Window size: 1

Frequency thr: 0,5

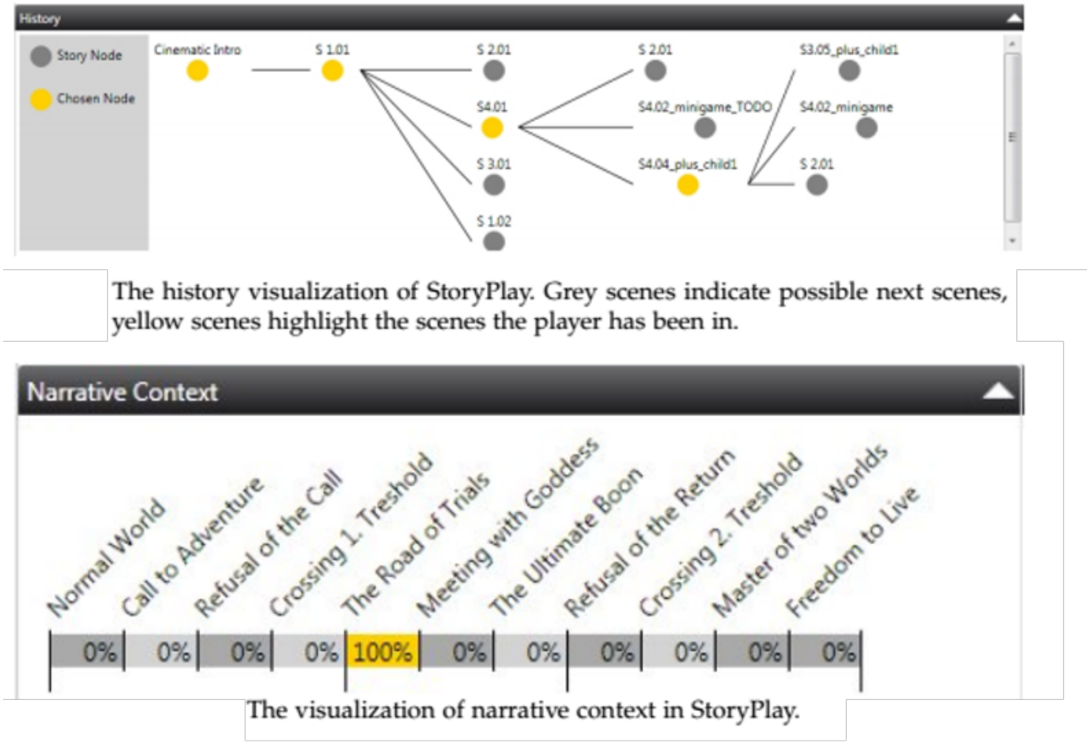
Figure 9: *Flagging Frames with degraded Video Quality based on Mobile Sensing Data*

Mobile Sensors Tabs A map tab visualizes the map and displays the GPS information captured during the session. Some technical difficulties were faced in using Google Maps so Bing Maps was used. Both required an account to generate a key to be inserted in the code for the application to work. The orientation of the device is represented on a separate tab with a dynamic 3D Model representation of the device. This model moves according to the movement which was recorded in the sensor log entries. Colored arrows show the different directions.

5 Evaluation

5.1 Playtesting with Children by the Researcher

After an initial technical evaluation where limitations were identified across multiple devices and data was collected using them, StoryPlay Multimodal was used to evaluate a game teaching children the German Artikel. To evaluate the application of the tool a special educational game was created with special requirements. The following were its main requirements (see Figure 12):



The history visualization of StoryPlay. Grey scenes indicate possible next scenes, yellow scenes highlight the scenes the player has been in.

The visualization of narrative context in StoryPlay.

Figure 10: *Scene History and Narrative Context Tabs in StoryPlay [35]*

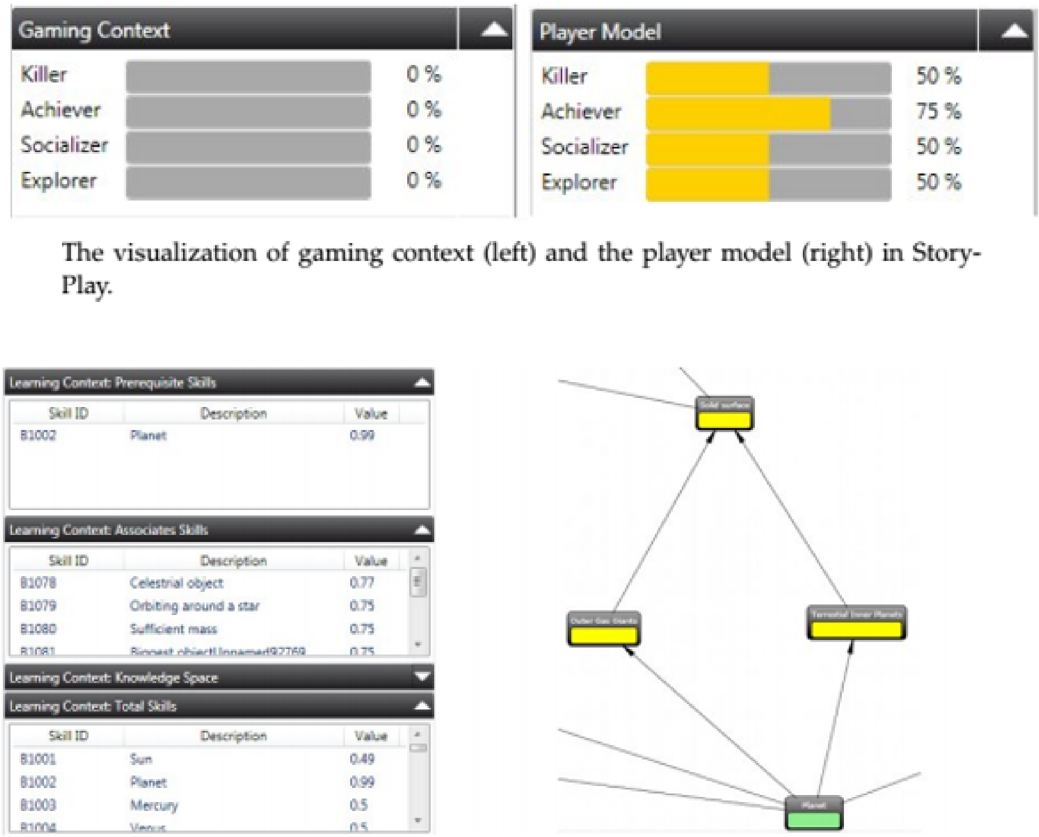
- Created with StoryTec
- Runing on different devices
- Diversity in mechanics and media
- Collecting high number of different gameplay-related events in a relatively short time.
- Analytics in mind: using tags for recognizable and quantifiable log events
- Evoking affect reactions: funny and motivating elements
- A minimum of three different subgames for comparison
- Integrating pre-& post-Test

The resulting game had three scene types for each subgame (game scenes, quiz scenes and video scene).

Three playtesting sessions were carried out in addition to a remote session which kids carried out with their parents at home. The data was automatically sent to a server from their mobile devices (if they chose to allow this). In total there were twenty unique children, aged from 4 to 10, but many kids played more than one session of the game. The data from the first session was used to improve the game for the second session, simulating iterative design and evaluation of games using information provided from the data.

Different devices were used in the evaluation a Laptop, a tablet and a smartphone, also different devices were used for the observer app: a tablet and a smartphone - in addition to the participants devices at home. In total, 22 log files were recorded (See Table 2).

A limitation of this first evaluation study is that it was a single case study with one Serious Game and carried out by one researcher. However, the varying types of scenes and subgames



Gaming Context

Killer	0 %
Achiever	0 %
Socializer	0 %
Explorer	0 %

Player Model

Killer	50 %
Achiever	75 %
Socializer	50 %
Explorer	50 %

The visualization of gaming context (left) and the player model (right) in StoryPlay.

Learning Context: Prerequisite Skills

Skill ID	Description	Value
B1002	Planet	0.99

Learning Context: Associates Skills

Skill ID	Description	Value
B1078	Celestrial object	0.77
B1079	Orbiting around a star	0.75
B1080	Sufficient mass	0.75
B1081	Biggest object|Unnamed92769	0.75

Learning Context: Knowledge Space

Learning Context: Total Skills

Skill ID	Description	Value
B1001	Sun	0.49
B1002	Planet	0.99
B1003	Mercury	0.5
B1004	Venus	0.5

: The visualizations for the skills of the player in textual form (left) and as a graph (right) in StoryPlay.

Figure 11: *Learning and Gaming Context Tabs in StoryPlay [35]*

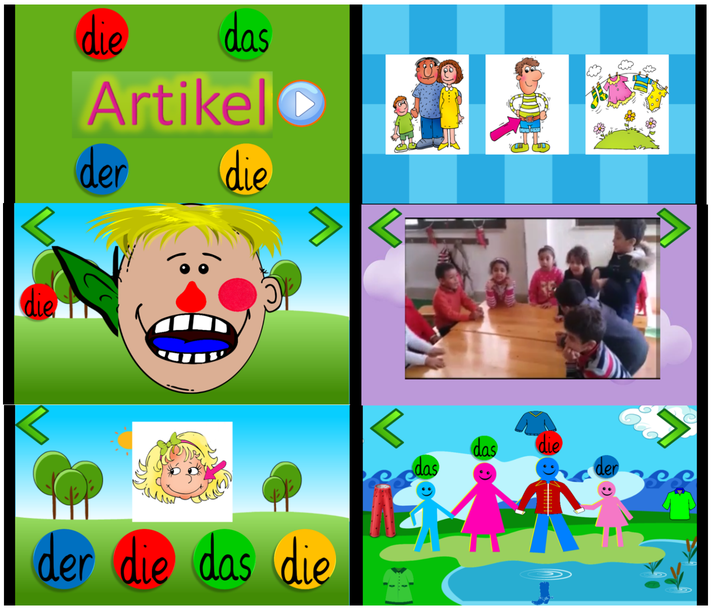


Figure 12: *Some Screens of the Mobile Game created with StoryTec and tested with StoryPlayMM [20]*

Table 1: *Evaluation Participants in Different Sessions*

Session	4y	5y	6y	7y	8y	9y	10y	Total
no.1	0	1	5	3	1	1	1	12
no.2	1	0	6	0	1	0	0	8
no.3	1	3	0	0	0	0	0	4
Remote	0	0	2	0	0	0	0	2
Total	0	0	2	0	1	1	1	26 (20 unique)

Table 2: *Devices and Software used in Evaluation and number of Log Files generated*

Session	Device	Game Version	Game Log Files	Observer Log Files	Device Videos	Observer Videos	Observer Notes
no.1	Laptop	1	0	2	2	8	12
no.2	Tablet	1	8	8	8	1	8
no.3	Smartphone	2	4	4	4	2	4
Remote	Tabl./Sm.	2	10	0	4	0	0
Total			22 (14 unique)	14	18	11	24

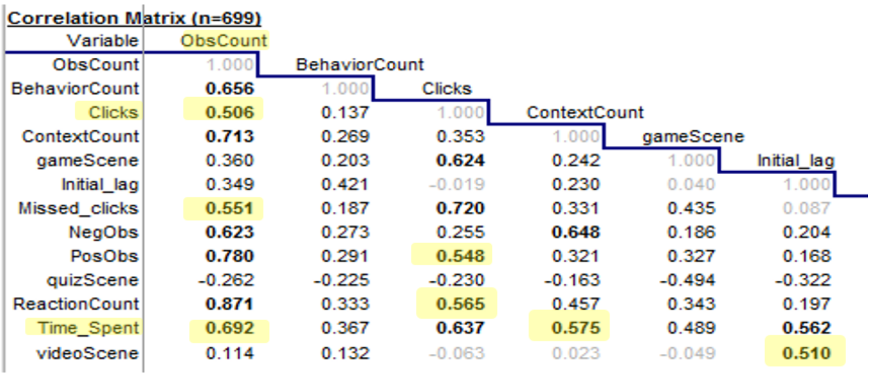
in the game offered variation and depth to the analysis. In addition it had to be fixed to allow for trying the different features on the same game, on multiple user sessions to have comparable results. Also, the observation approach is better to initially be used with testers on a relatively small scale, then it can be enhanced based on the data for larger scale evaluations.

The evaluation with children helped mainly identify some improvement aspects which were reported in [20] and improved in further development versions of all modules. In addition, the evaluation process was useful for improving the game itself in several iterations.

As a main benefit of the platform was enabling the linking of quantitative with qualitative data, the evaluation platform was used for exploration in this regard. The two main log statistics used for deeper investigation for each scene visited were the number of stimulus (e.g. response to a clickable object) in a scene and the time between stimulus. These were examined in relation to the observed states from the LeGUC states presented earlier. When interesting behaviors were found in the extracted statistics, corresponding observations in the observer file were investigated to see if they give additional information.

Different count measures were extracted from game logs and observer logs, aligned together and aggregated to count data frequencies of different reactions, behavior and context in each game scene (see Figure 13). Count measures extracted for each scene from the observer logs included reaction, context and behavior recordings count as well as recorded observer notes and participant utterances. From the game log, measures such as the number of visits for each scene, the average number of clicks per visit, the average number of missed clicks per visit, the average time per click, the initial delay (time to first click on first visit) and the time on scene were extracted.

As the game used for testing had different scene types (game, video, quiz), the number of clicks in a scene and the time between clicks had different meanings in different scene types: for a quiz scene it is better to finish the scene quickly whereas in the video scene this would mean that the kid was bored and wanted to skip the video. A high number of clicks in the game scene is a good sign meaning there is a lot of engaged interaction whereas in the quiz scene this means many wrong answer attempts. The number of states recorded using the observer app were measured in different scenes to see if they correlate with the logging events differently. Indeed, the number of clicks and time spent in the game scenes both lead to a



Correlation Matrix (n=699)

Variable	ObsCount	BehaviorCount	Clicks	ContextCount	gameScene	Initial_lag
ObsCount	1.000					
BehaviorCount	**0.656**	1.000				
Clicks	**0.506**	0.137	1.000			
ContextCount	**0.713**	0.269	0.353	1.000		
gameScene	0.360	0.203	**0.624**	0.242	1.000	
Initial_lag	0.349	0.421	-0.019	0.230	0.040	1.000
Missed_clicks	**0.551**	0.187	**0.720**	0.331	0.435	0.087
NegObs	**0.623**	0.273	0.255	**0.648**	0.186	0.204
PosObs	**0.780**	0.291	**0.548**	0.321	0.327	0.168
quizScene	-0.262	-0.225	-0.230	-0.163	-0.494	-0.322
ReactionCount	**0.871**	0.333	**0.565**	0.457	0.343	0.197
Time_Spent	**0.692**	0.367	**0.637**	**0.575**	0.489	**0.562**
videoScene	0.114	0.132	-0.063	0.023	-0.049	**0.510**

Figure 13: *Extracting and Aligning Count Measures for Each Scene from Observer and Game Logs*

higher number of recorded events. Whereas in the video and quiz scenes they had a different influence. For example in the video scene, when more time is spent, less behavior is observed. Observations were necessary to make sense of the logging data. This can be an indicator that multimodal and mobile sensing data will help disambiguate some logging data [36] and help determine context of the experience. See Figures 14 and 15.

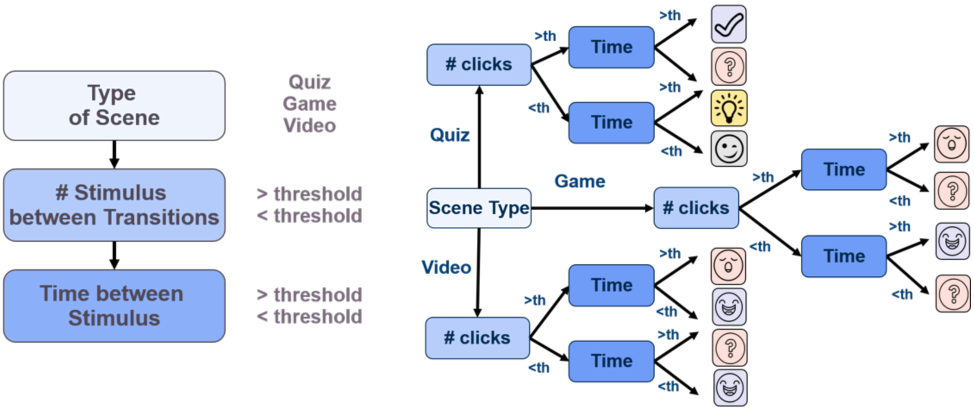
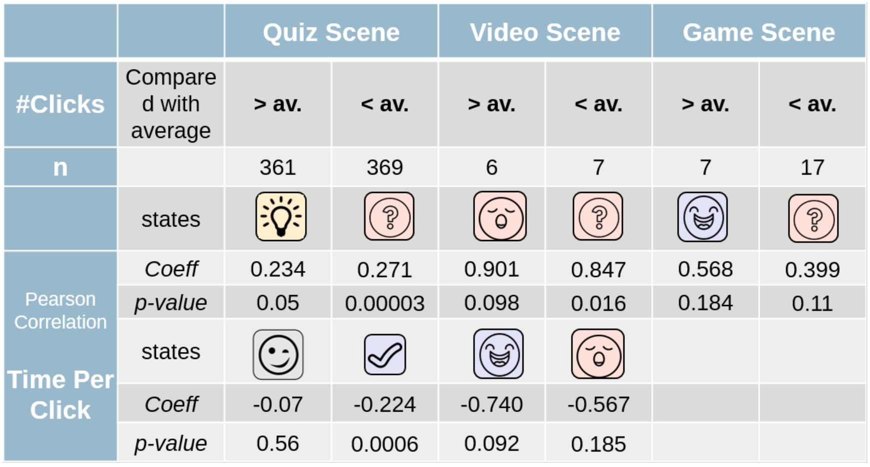


Figure 14: *Detecting Intersting Behavior in Logs and Investigating Corresponding Observer Data*

5.2 *Evaluation by Students*

This evaluation study included undergraduate Computer Science students taking a Game Design course in their last semester and was carried out at the end of the course as practical application on the topic of digital game prototyping. Two different classes took part in the sessions. From the first group, seven students actively took part, four female and three male students, and from the second group nine students were involved, five male and four female students. In some milestones, the students worked in groups, in others individually. The evaluation process consisted of the following steps carried out by the students:

1. Creating a game with Storytec and exporting it into an apk
2. Testing apks of friends on their mobile devices in the lab while another team member acts as the observer using the observer app on his/her device
3. Filling out an online form about their playtesting process



		Quiz Scene		Video Scene		Game Scene	
#Clicks	Compared with average	> av.	< av.	> av.	< av.	> av.	< av.
n		361	369	6	7	7	17
	states	💡	?	😮	?	😆	?
Pearson Correlation	*Coeff*	0.234	0.271	0.901	0.847	0.568	0.399
	p-value	0.05	0.00003	0.098	0.016	0.184	0.11
Time Per Click	states	☺	☑	😆	😮		
	Coeff	-0.07	-0.224	-0.740	-0.567		
	p-value	0.56	0.0006	0.092	0.185		

Figure 15: *Different Relations between Game and Observer Count Measures for Different Scene Types*

4. Letting another tester (preferably a younger family member at home) test their apk outside the lab while the student takes observational notes using the observer app.

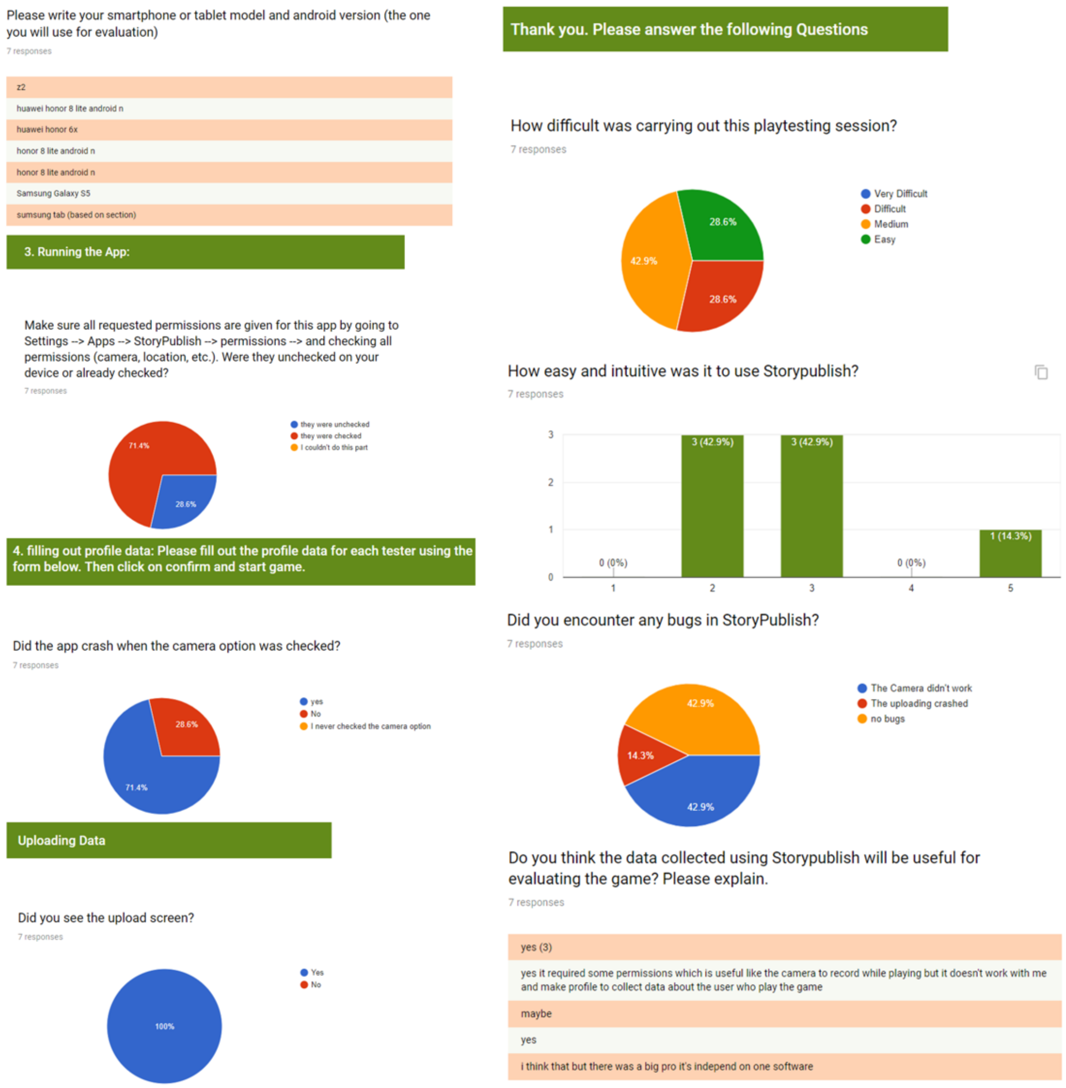
5. Filling out a form about using the capturer and observer app

6. The instructor checks the data sent to the server from the devices and assists in technical problems.

Figure 16: *Participants in Student Evaluation*

The main aim of the study was to test the mobile applications on different devices and by different playtesters and observers as well as test the smoothness of the whole process. An added value of this study to the first evaluation by the researcher is also to get remote observer files and test the procedure of their integration with remote playtesting log files from another device. The online form steps guided students through the evaluation process so that minimum interventions were needed and asked them questions useful for assessing the evaluation experience. The last step of testing the desktop replay component by students is still to be carried out in a further milestone but it was used by the instructor to test the whole integration and view the received files.

Figures 17 and 18 show some results from the online forms filled by the students in their evaluation study. It can be seen that one major problem was found with respect to switching on the camera on the devices which make the app crash on some devices as discussed in the implementation challenges in Section 6. Although this problem had been resolved on other devices during the initial technical evaluation, it was found that it was still present on other devices. In addition, some devices failed to send the data to the server because of restrictions imposed on apps not downloaded from the Google Play app store. Also, all permissions for the app like camera, location and data access had to be set manually by going to the settings (these steps were explained in detail to the students in the online form but some students failed to carry them out correctly). In general, the challenge of getting all features to work properly on all different devices was found to be a very difficult and time-consuming process and new android updates need further updates in some apps when they emerge.



Please write your smartphone or tablet model and android version (the one you will use for evaluation)

7 responses

z2
huawei honor 8 lite android n
huawei honor 6x
honor 8 lite android n
honor 8 lite android n
Samsung Galaxy S5
sumsung tab (based on section)

3. Running the App:

Make sure all requested permissions are given for this app by going to Settings --> Apps --> StoryPublish --> permissions --> and checking all permissions (camera, location, etc.). Were they unchecked on your device or already checked?

7 responses

4. filling out profile data: Please fill out the profile data for each tester using the form below. Then click on confirm and start game.

Did the app crash when the camera option was checked?

7 responses

Uploading Data

Did you see the upload screen?

7 responses

Thank you. Please answer the following Questions

How difficult was carrying out this playtesting session?

7 responses

How easy and intuitive was it to use Storypublish?

7 responses

Did you encounter any bugs in StoryPublish?

7 responses

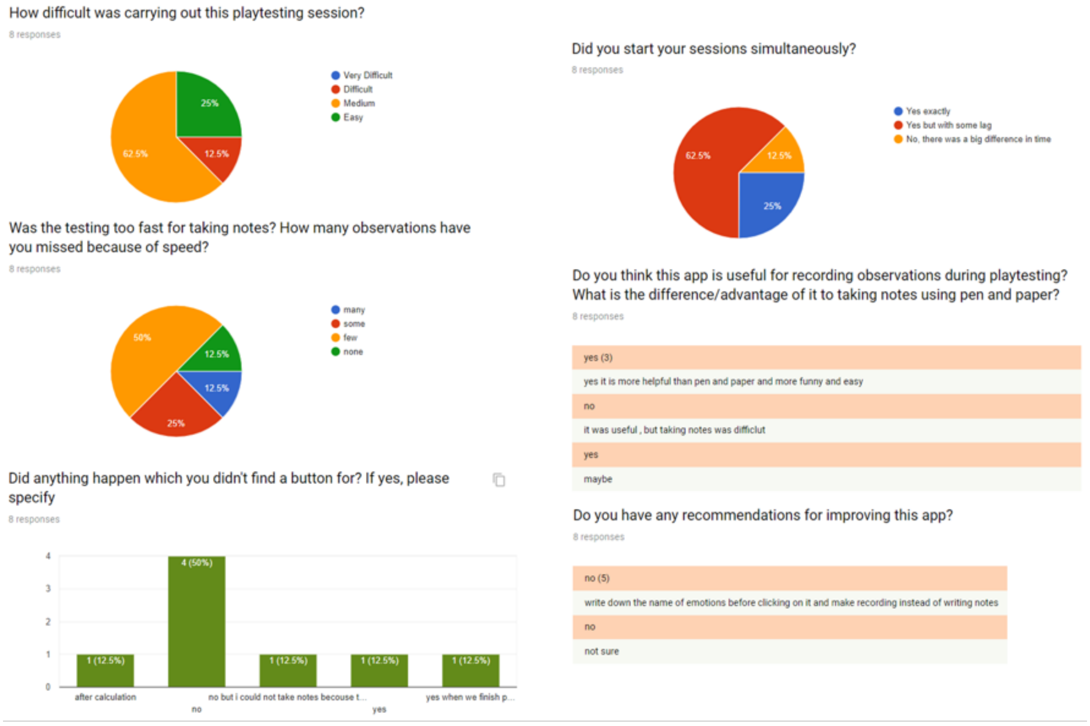
Do you think the data collected using Storypublish will be useful for evaluating the game? Please explain.

7 responses

yes (3)
yes it required some permissions which is useful like the camera to record while playing but it doesn't work with me and make profile to collect data about the user who play the game
maybe
yes
i think that but there was a big pro it's independ on one software

Figure 17: *Using the Capturer App - Some Evaluation Form Results*

As can be seen in Figures reffig:capturerwide and 18, the majority (80%) of students found the apps useful in testing their friends' games as well as improving their own game by observing others playtesting it. They also found most controls intuitive. A feature which was found to need improvement was that clicking the back button while uploading cancelled the uploading while only the home button allowed for continuing the upload in the background. In addition, in both evaluation procedures, the multimodal capturer app helped enable remote evaluation in more naturalistic and unobtrusive settings. Mobile sensing data helped in identifying bad frames from video recordings and give richer information about the session. Using the observer app for the observations also revealed valuable insights into Learner Experience aspects as demonstrated in [20]. According to occurrences of some behavioral, affective and context states recorded during the playtesting sessions, improvements for the next iteration of the observer app were identified and implemented (see more details in [20]).


How difficult was carrying out this playtesting session?

8 responses

Was the testing too fast for taking notes? How many observations have you missed because of speed?

8 responses

Did anything happen which you didn't find a button for? If yes, please specify

8 responses

Did you start your sessions simultaneously?

8 responses

Do you think this app is useful for recording observations during playtesting? What is the difference/advantage of it to taking notes using pen and paper?

8 responses

yes (3)

yes it is more helpful than pen and paper and more funny and easy

no

it was useful , but taking notes was difficlut

yes

maybe

Do you have any recommendations for improving this app?

8 responses

no (5)

write down the name of emotions before clicking on it and make recording instead of writing notes

no

not sure

Figure 18: *Using the Observer App - Some Evaluation Form Results*

5.3 Evaluation by a Teacher

The process of using the three components of StoryPlayMM to evaluate a mobile educational game (Artikelgame) was carried out by a German teacher holding a German language workshop for Egyptian kids in a learning center in Egypt. The sessions included seven boys and one girl between the age of four and seven. The following steps were included in the evaluation carried out by the teacher:

1. Running the game with the capturer app on one device and filling a profile at the beginning of every session for each child before the child starts playing on it
2. Running the observer app on another device which the teacher holds and filling a profile about the teacher as the observer
3. Letting each child test the game with the capturer app in the learning center and making observations using the observer app while the app is recording multimodal data
4. Opening session data on StoryPlayMM desktop component by the teacher after the sessions and examining them to give feedback on encountered problems and possible improvements in usability
5. Filling out an online form about using the capturer and observer app

6. Filling out an online form about the replayer software

From the device given to the children by the teacher, the videos were unfortunately recorded without sound. It is not clear why this particular device recorded videos without sound. Also, the teacher was not adviced to instruct students to hold the device in a certain way to make the sessions as natural as possible. However, the device was put on a table and thus was facing the ceiling all the time. These video frames were automatically flagged in the video analysis tab as bad video segments. By watching them, it was found that only one video showed part of a child's face. The feedback given by the teacher in the evaluation forms suggests a smooth evaluation session and an intuitive use of the observer app by the teacher. However, it was noted that the sessions were not properly terminated with the StoryPublish app which resulted in many session logs recorded into one file with many pauses in-between sessions, instead of creating a new log file for each session. This means the process of terminating the game between sessions was not intuitive for the teacher.

6 Challenges

6.1 Synchronization

The advantage in our approach of using built-in mobile sensors is that most of the recordings are done on the same device, so that there are no different hardware sensors requiring some hardware synchronization like in most similar software. However, synchronization was still a major challenge faced in the implementation phase as also reported in literature[2, 16, 27, 28] is multimodal data synchronization. Recordings have different frame-rates, start and end time as well as varying reaction time of the observer in case of observer logs. Other synchronization problems were related to StoryPublish which is used to replay the game from the log file. The replay here is an emulation of input and feedback which has its own lags, and the original implementation thus does not allow instantaneous skipping to an event, but rather replays all events quickly until it reaches the desired game state instance when jumping to an event. In addition, rendering the game has its own additional lag and the irregular frame-rate in wpf and decoupling of threads made an exact synchronization almost impossible. Jumping to the correct frame when clicking on an event was not an easy task, especially with the high frequency of events in a session. Also, cursor position and global time needed to be updated on all tabs when skipping. These calculations were found to be consuming more than 85 percent of the UI thread, making the skipping very slow. Some skip took up to 20 seconds which is why a loading icon was added in this case to avoid confusion. Many optimizations were done and others can still be added to the code to make this process faster. In addition to reducing the number and frequency of such calculation operations and reducing rendering, it can be useful to save the current status in order not to start calculating positions and sizes of all elements in every frame. However, the saving operation might also introduce additional delay.

To address the synchronization issues, an optimal affordable granularity of integration has to be defined for the specific case, then choosing a suitable frame-rate and letting all data obey the same timer. In our case, the timer of the game was chosen in order to introduce the same lags of processing into the other timelines to avoid getting out of sync. To compensate for different starting/ending/reaction times a user-defined offset is allowed using a sliding bar. This can also be improved to automatically align data using some recognition techniques. Event-driven synchronization could also be used to synchronize observer data with the log files.

Game logs and observer logs were also joined on the statistics tab to provide a mapping between quantitative and qualitative data in statistics and not just in replay. One issue in doing this was the time lag between both logs because of the reaction time of the observer. This was also handled in the replay synchronization by adding a manual offset. Here it is required to make sure the lag is not too big to give wrong results. For example, to count number of reaction observations in a certain scene or scene type there has to be a mapping between scenes and the observations made in them which when depending on time only could be inaccurate. Although a "new subgame" button was added in the observer to annotate the beginning of a new part, this information was not reliable as the observer needs to click on the button in time which is not always easy when the game advances fast. So the manual offset set in the replay by the researcher at analysis time need also to be used in the statistics.

6.2 *Interoperability and the Heterogenity of Devices*

The heterogenity of devices was also a very time-consuming challenge as applications have to be tested on different devices and different Android versions and updates. This renders it very difficult to make sure it will run smoothly on all possible user devices. Even sensor configurations can differ between devices as some give more useful detailed values than others. The possible values and thresholds for proximity, for instance, differ between devices. The screen resolution problem of devices also affects the replay of games and the calculations of game elements positions and click locations. Some changes may even affect the whole game replay as clicks may be missed when regenerated in replays. Furthermore, video and picture formats supported are different on devices which makes some game elements not run on certain phones. Dealing with permissions to start the camera or save log files was different from one device to another. Thus, it was difficult to give general instructions to all users on how to operate the application and various tests needed to be run to discover these differences. Even after enabling these permissions, some cameras would not work on some devices and the code needed to be debugged for those cases. And even after running the camera, the differences in encoding the data made some videos get saved without the audio, or with audio only without picture. Thus, the heterogenity of devices is considered one of the main challenges for the creation and maintenance of such multimodal applications for mobile devices. Not only this, but compiling the original Storypublish c++/haxe code for Android takes a very long time which makes updating the code with any new feature or adjustment and retesting on devices a considerably time-consuming task.

6.3 *Data Quality*

One main challenge of combining multimodal data for the evaluation was concerning the quality of recorded data, for example the quality of video recordings which is also discussed in [30, 37–39]. Videos from front-facing cameras on smartphones suffer from a dynamic environment which results in variations in illumination, stability, orientation, exposure and distance. To deal with this problem, the aforementioned feature of using data from low-power sensors like accelerometer, orientation, illumination and proximity and applying user-defined thresholds to determine if context conditions are suitable for obtaining good quality data from video or audio was implemented. The goal is to emphasize only useful data to reduce data size and/or reviewing time. Another good feature which could be added in the future would have the goal of enhancing the video using these sensor data, for example automatic illumination compensation in bad video segments. A plugin using ffmpeg, for instance, can be added to provide the user with the option to adjust adjust brightness, contrast and rotation. The recognition of bad orientation could also be done on-the-fly during recording to alert users to adjust the camera view to show their faces in the videos. Face detection could also be used in the beginning of the sessions for this purpose.

6.4 Data Granularity and the Heterogenity of Scenes and Games

One big advantage for this project was the similarity of game structure as all games are created with the same authoring environment StoryTec. Nevertheless, there were still differences between games and scenes. This is why considerable time needed to be invested in making data ready for analysis by the researcher. Pre-processing data so they can give meaningful clues and can help to distinguish significant events while at the same time staying general enough to accommodate different games is not a trivial task. One small example was calculating the time-per-click metric. Although it should be a straightforward task, some differences had to be taken into consideration. Some scenes have some audio instructions at the beginning where mostly the interactions were delayed. To account for this, the initial lag before the first click was not calculated in the average time per click, but as a separate metric. Other scenes contained only a video so there were no meaningful clicks (only missed clicks, i.e. touch down and up without a stimulus invoked). Here the time-per-click metric would be equal to the time-on-scene and would help the researcher only if it is crucial to know if kids skip videos early, for example. In some cases this points to the fact that they find the video boring, or just that they skipped by mistake, which in both cases need to be considered in design. In addition, it can help distinguish the taste of different genders or ages of players for certain content. Also, some scenes can have a transition without a click (e.g. based on a timer) or reach the end of the game, so here the last delta time calculated would be until the end of the scene. Many other similar examples of differences between scenes were encountered in this process.

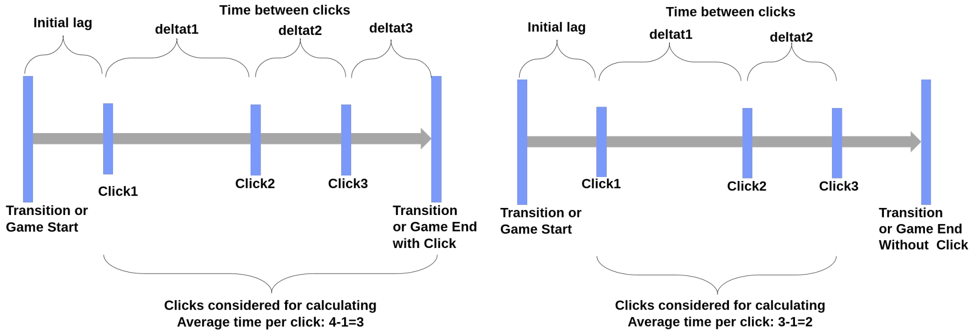

Initial lag
Time between clicks
deltat1
deltat2
deltat3
Click1
Click2
Click3
Transition or Game Start
Transition or Game End with Click
Clicks considered for calculating Average time per click: 4-1=3
Initial lag
Time between clicks
deltat1
deltat2
Click1
Click2
Click3
Transition or Game Start
Transition or Game End Without Click
Clicks considered for calculating Average time per click: 3-1=2

Figure 19: *Differences in Calculating Average Time per Click*

Another example of data granularity was the time-on-scene metric. It is desirable to know the average time spent on each scene (accumulated over all visits of the scene and divided over the number of visits) but also to know the time spent on a particular scene in a particular visit. Also, there is a difference between the time on the first visit and the time on further visits of one and the same scene. This depends on the type of the scene but generally the first time a scene is visited there is more time needed for discovery as in next visits the player has usually already "mastered" this scene. Thus aggregating data sometimes hides important information which is only uncovered when investigated separately. Both options need to be given to the researcher for different purposes.

]

6.5 Privacy

The privacy issue was addressed in the initial requirements by giving users full control over sensor activation and over what and when to store and/or upload to the server. However, some additional points concerning privacy needed to be handled. At the beginning, the only way offered for uploading data was uploaded to a firebase project created for gathering log data from this particular application (a key for each app is needed to be added to the project using the account created for the current research). The same process should be applied by any researcher before compiling each of his games so that the data is sent to his/her account which would be the most secure option. Another option which is provided to researchers in the current app is to create their own account inside the created firebase project and to use it for uploading (from capturer) and downloading (by replayer) their data privately without accessing other researchers' data. However, in this case the creators of the initial account (i.e. of this research project) still have access to all data sent to the server. This is why a google drive option was added to the application where users can log into their own google drive accounts and upload data there, so they can have full control and privacy. The problem with both log-in features is the complicated setup needed for them to run on the user end for newly created games as new keys need to be generated for each app and added before compilation. The final more secure option is to use an offline mechanism for transferring the saved files to the researcher's desktop. However, this needs expert users to deal with getting this app data from their devices and using them on their laptops for the replay or sending them to a remote researcher.

6.6 Other Challenges

In addition to the discussed main challenges, many other challenges were identified and addressed in the initial project requirements, like avoiding obtrusiveness. The problem of high battery consumption of the capturer app was not very concerning as the mobile edugame sessions are relatively short. This would need to be tested for prolonged sessions where a player repeats the same game several times over a long period of time. Another challenge was making the platform easily usable and intuitive for non-professional users. Many layout changes were carried out to accommodate for the many tabs which may need to be opened at once and to give the user the choice to change the layout to one which suits his/her needs and devices. A switch between horizontal and vertical layout and a flexible tab size in all directions was found to be very useful. Also, making sure the menus, buttons and icons on timelines are descriptive enough to make it easier to understand during analysis and providing easy means of skipping needed many iterations and tests. Other additions like muting videos when their tab is not expanded and muting game sounds while skipping were made to make the evaluation session run more smoothly.

7 Conclusion, Limitations and Future Work

In this paper an environment offering capturing, synchronization, replay, pre-processing and interactive navigation of multimodal data for Serious Games evaluation was introduced. This unified visualization of quantitative and qualitative playlearner data makes it possible to discover relations between game elements and playtester behaviors, affective and cognitive states as well as evaluation context. The steps described for creating this proof-of-concept software help describe how multimodal data can be captured unobtrusively for Serious Games evaluation, how it can be linked to recorded log events and what the associated challenges are. Many improvement possibilities as well as useful applications of the current project have been mentioned throughout this paper which can be summarized as follows:

- training machine learning modules to predict whether a frame contains faces using the accelerometer and gyroscope sensors
- using built-in smile detectors or facial feature recognition modules
- collecting and saving more user data and demographics automatically and tracking learning progress over time, also the possibility of using information for game rating or recommender systems
- detecting and highlighting other relevant data such as correlations, repetitions and significant events and sequences. Detecting event sequences may be helpful for comparing the input stream with some target sequence (e.g. of an expert player) or emphasizing certain pre-defined player behavior. Aggregations and Correlations can be done across users for extensive data for identifying common paths and interaction patterns or rating scenes(e.g. scenes where most users had a sudden movement or smile or laughter etc.) or across modalities for one and the same user for intensive data (when more than one data source has significant change at the same time) .
- Recorded observer reactions, behaviors and context events can help in training a machine learning algorithm for extracting features from multimodal data by serving as a ground truth.
- Improvements on the design of the desktop replayer app are mainly in optimizations in replay and synchronization mechanisms. Observer and interaction data can be automatically aligned together instead of the manual offset by training a machine learning algorithm to detects lags in observer session start time. The same can be applied to aligning statistic aggregations.
- Video quality can be enhanced using the sensor data monitored, for example automatic illumination compensation in bad video segments. A plugin using ffmpeg, for instance, can be added to provide the user with the option to adjust adjust brightness, contrast and rotation. The recognition of bad orientation could also be done on-the-fly during recording to alert users to adjust the camera view to show their faces in the videos. Face detection could also be used in the beginning of the sessions for this purpose.

The user studies presented were carried out to demonstrate the benefits of combining multimodal data with event logging for the evaluation of mobile learning games. Structuring and linking raw multimodal data for easier navigation was found to be very helpful in carrying out user studies of this type of software. Unified visualization of quantitative and qualitative playlearner data made it possible to discover relations between game elements and playtester behaviors, affective and cognitive states as well as evaluation context. Results showed some benefits of multimodal data for interpreting log events. However, the challenges discussed in Section 6 were the main challenges faced during the implementation as well as the evaluation process. Once these challenges have been overcome, a more thorough evaluation investigation can follow.

More research is needed to determine when exactly adding more data adds value for the evaluation of Serious Games and when it is a waste of resources. Some studies in mobile sensing address these problems to adaptively switch on different sensors according to environmental, device and user conditions. This research area is especially promising as it helps develop efficient and effective multimodal mechanisms which add richness to evaluation processes.

References

[1] L. Anolli, F. Mantovani, L. Confalonieri, A. Ascolese, and L. Peveri, "Emotions in serious games: From experience to assessment." *iJET*, vol. 5, no. SI3, pp. 7–16, 2010. doi: https://doi.org/10.3991/ijet.v5s3.1496

[2] L. Shoukry, S. Göbel, and R. Steinmetz, "Learning analytics and serious games: Trends and considerations," in *Proceedings of the 2014 ACM International Workshop on Serious Games*. ACM, 2014. doi: https://doi.org/10.1145/2656719.2656729 pp. 21–26.

[3] L. Shoukry and S. Göbel, "Reasons and responses: A multimodal serious games evaluation framework," *IEEE Transactions on Emerging Topics in Computing*, 2017. doi: https://doi.org/10.1109/TETC.2017.2737953

[4] P. Mirza-Babaei, G. Wallner, G. McAllister, and L. E. Nacke, "Unified Visualization of Quantitative and Qualitative Playtesting Data," *Proceedings of CHI EA 2014*, pp. 1363–1368, 2014. doi: https://doi.org/10.1145/2559206.2581224. [Online]. Available: http://dl.acm.org/citation.cfm?id=2581224

[5] L. Shoukry, S. Göbel, and R. Steinmetz, "Towards mobile multimodal learning analytics," in *Proceedings of the Learning Analytics for and in Serious Games Workshop at EC-TEL*, 2014. doi: https://doi.org/10.1145/2656719.2656729

[6] N. Bosch, F. Hall, and N. Dame, "Multimodal affect detection in the wild : Accuracy , availability , and generalizability," 2015. doi: https://doi.org/10.1145/2818346.2823316

[7] L. Shoukry and S. Göbel, "Storyplay multimodal: A research tool for the multimodal evaluation of serious games," in *Proceedings of the 11th European Conference on Games Based Learning*, D. J. G. Dr Maja Pivec, Ed. Academic Conferences and Publishing International Limited, Oct 2017. ISBN 978-1-911218-57-9 pp. 577–584.

[8] J. B. HAUGE, I. A. STÃNESCU, A. STEFAN, M. B. CARVALHO, T. LIM, S. LOUCHART, and S. ARNAB, "Serious game mechanics and opportunities for reuse." *eLearning & Software for Education*, no. 2, 2015.

[9] E. Harpstead, B. A. Myers, and V. Aleven, "In search of learning: facilitating data analysis in educational games," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2013. doi: https://doi.org/10.1145/2470654.2470667 pp. 79–88.

[10] V. Zammitto, P. Mirza-Babaei, I. Livingston, M. Kobayashi, and L. E. Nacke, "Player experience: mixed methods and reporting results," in *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2014. doi: https://doi.org/10.1145/2559206.2559239 pp. 147–150.

[11] C. T. Tan, P. Mirza-Babaei, V. Zammitto, A. Canossa, G. Conley, and G. Wallner, "Tool design jam: Designing tools for games user research," in *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. ACM, 2015. doi: https://doi.org/10.1145/2793107.2810263 pp. 827–831.

[12] J. H. Kim, D. V. Gunn, E. Schuh, B. Phillips, R. J. Pagulayan, and D. Wixon, "Tracking real-time user experience (true): a comprehensive instrumentation solution for complex systems," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 2008. doi: https://doi.org/10.1145/1357054.1357126 pp. 443–452.

[13] P. A. Nogueira, V. Torres, R. Rodrigues, and E. Oliveira, "An annotation tool for automatically triangulating individualsß psychophysiological emotional reactions to digital media stimuli," *Entertainment Computing*, vol. 9, pp. 19–27, 2015. doi: https://doi.org/10.1016/j.entcom.2015.06.003

[14] J. M. Kivikangas, L. Nacke, and N. Ravaja, "Developing a triangulation system for digital game events, observational video, and psychophysiological data to study emotional

responses to a virtual character," *Entertainment Computing*, vol. 2, no. 1, pp. 11–16, 2011. doi: https://doi.org/10.1016/j.entcom.2011.03.006

[15] L. E. Nacke, S. Stellmach, D. Sasse, J. Niesenhaus, and R. Dachselt, "Laif: A logging and interaction framework for gaze-based interfaces in virtual entertainment environments," *Entertainment Computing*, vol. 2, no. 4, pp. 265–273, 2011. doi: https://doi.org/10.1016/j.entcom.2010.09.004

[16] A. Fouse, N. Weibel, E. Hutchins, and J. D. Hollan, "Chronoviz: A system for supporting navigation of time-coded data," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '11. New York, NY, USA: ACM, 2011. doi: https://doi.org/10.1145/1979742.1979706. ISBN 978-1-4503-0268-5 pp. 299–304.

[17] A. Morrison, P. Tennent, and M. Chalmers, "Coordinated visualisation of video and system log data," in *Coordinated and Multiple Views in Exploratory Visualization, 2006. Proceedings. International Conference on*. IEEE, 2006, pp. 91–102.

[18] B. Drenikow and P. Mirza-Babaei, "Vixen: interactive visualization of gameplay experiences," in *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 2017. doi: https://doi.org/10.1145/3102071.3102089 p. 3.

[19] S. Göbel, A. de Carvalho Rodrigues, F. Mehm, and R. Steinmetz, "Narrative game-based learning objects for story-based digital educational games," *narrative*, vol. 14, p. 16, 2009. [Online]. Available: http://www.eightydays.eu/Paper/GdCMS09.pdf

[20] L. Shoukry, J. Konert, and S. Göbel, "The evaluation of learner experience in serious games," in *Learner Experience and Usability in Online Education*. IGI Global, 2018, pp. 122–148.

[21] F. Mehm, S. Göbel, and R. Steinmetz, "Authoring of serious adventure games in storytec." Springer, 2012, pp. 144–154.

[22] C. Reuter, F. Mehm, S. Göbel, and R. Steinmetz, "Evaluation of Adaptive Serious Games using Playtraces and Aggregated Play Data," *Proceedings of the 7th European Conference on Game Based Learning (ECGBL) 2013*, no. October, pp. 504–511, 2013.

[23] L. Shoukry, C. Reuter, and F. Mehm, "Storytec and storyplay as tools for adaptive game-based learning research," in *Games for Training, Education, Health and Sports*. Springer, 2014, pp. 195–198.

[24] P. N. Dixit and G. M. Youngblood, "Understanding information observation in interactive 3d environments," in *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*. ACM, 2008. doi: https://doi.org/10.1145/1401843.1401874 pp. 163–170.

[25] C. T. Tan, S. Bakkes, and Y. Pisan, "Inferring player experiences using facial expressions analysis." in *Proceedings of the 2014 Conference on Interactive Entertainment - IE2014*, 2014. doi: https://doi.org/10.1145/2677758.2677765 pp. 7–1.

[26] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay, "Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones," in *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, 2007. doi: https://doi.org/10.1145/1247660.1247670 pp. 57–70.

[27] G. Dyke, K. Lund, and J.-J. Girardot, "Tatiana: An environment to support the cscl analysis process," in *Proceedings of the 9th International Conference on Computer Supported Collaborative Learning - Volume 1*, ser. CSCL'09. International Society of the Learning Sciences, 2009. doi: https://doi.org/10.3115/1600053.1600062. ISBN 978-1-4092-8598-4 pp. 58–67. [Online]. Available: http://dl.acm.org/citation.cfm?id=1600053.1600062

[28] P. Brundell, P. Tennent, C. Greenhalgh, D. Knight, A. Crabtree, C. O'Malley, S. Ainsworth, D. Clarke, R. Carter, and S. Adolphs, "Digital replay system (drs)-a tool

for interaction analysis," in *Proceedings of the 2008 International Conference on Learning Sciences (Workshop on Interaction Analysis)*, 2008.

[29] P. H. Zimmerman, J. E. Bolhuis, A. Willemsen, E. S. Meyer, and L. P. Noldus, "The observer xt: A tool for the integration and synchronization of multimodal signals," *Behavior research methods*, vol. 41, no. 3, pp. 731–735, 2009. doi: https://doi.org/10.3758/BRM.41.3.731

[30] S. Han, R. Nandakumar, M. Philipose, A. Krishnamurthy, and D. Wetherall, "Glimpsedata: Towards continuous vision-based personal analytics," in *Proceedings of the 2014 workshop on physical analytics*. ACM, 2014. doi: https://doi.org/10.1145/2611264.2611269 pp. 31–36.

[31] J. Staiano, M. Menéndez, A. Battocchi, A. De Angeli, and N. Sebe, "Ux_mate: from facial expressions to ux evaluation," in *Proceedings of the Designing Interactive Systems Conference*. ACM, 2012. doi: https://doi.org/10.1145/2317956.2318068 pp. 741–750.

[32] S. Göbel, V. Wendel, C. Ritter, and R. Steinmetz, "Personalized, adaptive digital educational games using narrative game-based learning objects." Springer, 2010, pp. 438–445. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-14533-9_45

[33] K. Korossy, "Modeling knowledge as competence and performance," *Knowledge spaces: Theories, empirical research, and applications*, pp. 103–132, 1999.

[34] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit muds," *Journal of MUD research*, vol. 1, no. 1, p. 19, 1996.

[35] F. Mehm, V. Wendel, S. Göbel, and R. Steinmetz, "Bat cave: A testing and evaluation platform for digital educational games," in *Proceedings of the 3rd European Conference on Games Based Learning*, 2010, pp. 251–260.

[36] S. Oviatt, "Ten myths of multimodal interaction," *Communications of the ACM*, vol. 42, no. 11, pp. 74–81, 1999.

[37] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 71–84.

[38] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "Cenceme–injecting sensing presence into social networking applications," in *Smart Sensing and Context*. Springer, 2007, pp. 1–28.

[39] H. Lee, Y. S. Choi, S. Lee, and I. Park, "Towards unobtrusive emotion recognition for affective social communication," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 260–264.