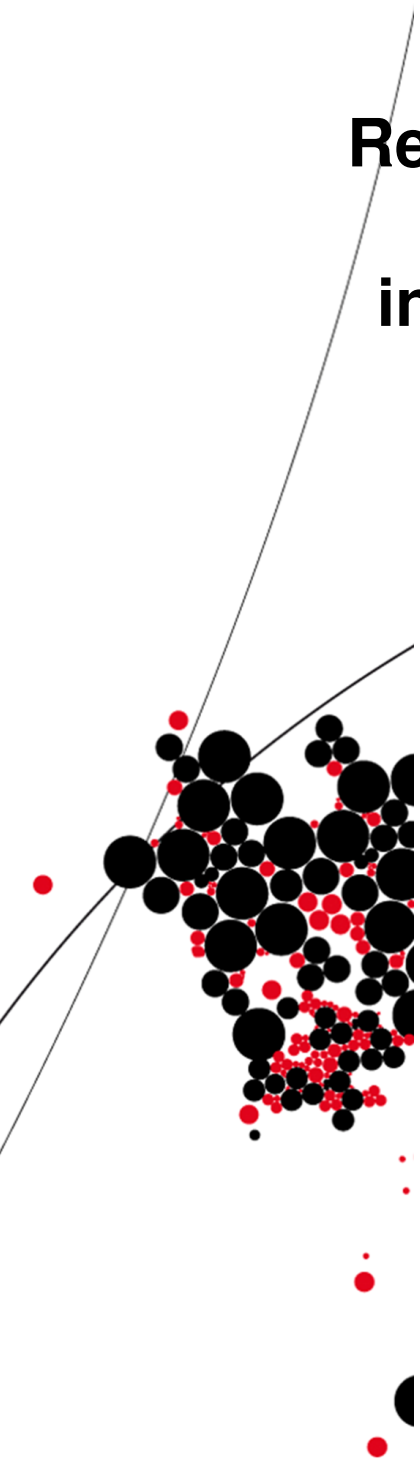




# UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering,  
Mathematics & Computer Science



## Evaluation of a Reusable In-Game Analytics Tool for Serious Games in Emotional and Mental Health

Alberto Vaccari  
Final Project  
August 2023

---

**Supervisors:**

dr.ir. R. Klaassen  
dr.ir. A.A.M. T. Spil

Human Media Interaction  
Faculty of Electrical Engineering,  
Mathematics and Computer Science  
University of Twente  
P.O. Box 217  
7500 AE Enschede  
The Netherlands

---



# Contents

<b>List of acronyms</b>	<b>vii</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Context . . . . .	4
1.2 Problem Statement . . . . .	4
1.3 Research Questions . . . . .	5
1.4 Scope . . . . .	6
1.5 Report organisation . . . . .	6
<b>2 Background</b>	<b>7</b>
2.1 Research Topics . . . . .	7
2.2 GEMH Lab . . . . .	11
2.2.1 Scroll Quest 2.0 . . . . .	11
2.2.2 Hit-n-Run . . . . .	13
2.3 Summary . . . . .	14
<b>3 Studies</b>	<b>15</b>
3.1 Study #1 - Game Metrics Validation . . . . .	15
3.1.1 Methods . . . . .	15
3.1.2 Materials . . . . .	17
3.1.3 Procedure . . . . .	18
3.1.4 Results . . . . .	18
3.2 Study #2 - Prototype Development . . . . .	23
3.2.1 Methods . . . . .	23
3.2.2 Materials . . . . .	24
3.2.3 Procedure . . . . .	24
3.2.4 Results . . . . .	25
3.3 Study #3 - Usability Evaluation of Prototype . . . . .	38
3.3.1 Methods . . . . .	38
3.3.2 Materials . . . . .	38
3.3.3 Procedure . . . . .	39

3.3.4	Results . . . . .	39
3.3.5	Discussion . . . . .	43
<b>4</b>	<b>Discussion</b>	<b>45</b>
4.1	Research Question #1: How compatible are in-game metrics collected from the learning and physical health domain with metrics from Serious Games for emotional and mental health? . . . . .	45
4.1.1	Limitations . . . . .	46
4.1.2	Future Work . . . . .	46
4.2	Research Question #2: What are usability characteristics of such a tool deemed important from a game developer’s perspective? . . . . .	47
4.2.1	Impact of Skill Level . . . . .	48
4.2.2	Limitations . . . . .	48
4.2.3	Future Work . . . . .	48
4.3	Main Research Question: How can a generic in-game analytics software tool for Serious Games for emotional and mental health be created? . . . . .	50
4.3.1	Implications . . . . .	51
4.3.2	Limitations . . . . .	52
4.3.3	Future Research . . . . .	52
<b>5</b>	<b>Conclusions</b>	<b>53</b>
	<b>Appendices</b>	<b>55</b>
A.1	Data Analysts - Interview Questions . . . . .	55
A.2	Data Analysts - Interview First Coding . . . . .	57
A.3	Data Analysts - Interview Generating Themes . . . . .	59
A.4	Game Developer - Feedback Interview . . . . .	63
A.5	Round 1 - Game Developer Test Results . . . . .	67
A.6	Round 1 - Game Developer Feedback Answers . . . . .	68
A.6.1	User 1 . . . . .	68
A.6.2	User 2 . . . . .	69
A.6.3	User 3 . . . . .	70
A.7	Round 2 - Game Developer Test Results . . . . .	77
A.8	Round 2 - Game Developer Feedback Answers . . . . .	78
A.8.1	User 4 . . . . .	78
A.8.2	User 5 . . . . .	79
A.8.3	User 6 . . . . .	80

A.9 Round 3 - User Experience Questionnaire . . . . .	87
A.10 Round 3 - System Usability Survey . . . . .	88
A.11 Round 3 - Game Developer Feedback Answers . . . . .	89
A.11.1 User 7 . . . . .	89
A.11.2 User 8 . . . . .	91
A.11.3 User 9 . . . . .	93
A.12 Programmer's Experience Heuristics . . . . .	95

<b>References</b>	<b>97</b>
-------------------	-----------



# List of acronyms

<b>SG</b>	Serious Game
<b>LA</b>	Learning Analytics
<b>GA</b>	Game Analytics
<b>SEG</b>	Serious Educational Game
<b>NPC</b>	Non-Playing Character
<b>SQ2.0</b>	Scroll Quest 2.0
<b>HnR</b>	Hit-n-Run
<b>SUS</b>	System Usability Survey
<b>UEQ</b>	User Experience Questionnaire





## **Abstract**

Serious Games for emotional and mental health have gained increasing attention due to their potential in testing new techniques for improving mental health and well-being. Collecting in-game analytics for player's behaviour can provide invaluable insights and data to support the results of interventions. However, the development of such games and data collection often involves challenges in communication and collaboration between game developers and data analysts. This study addresses these challenges by developing a reusable in-game analytics software tool that standardises the collection of in-game metrics related to player behaviour. Through an iterative process involving feedback from game developers and a senior data analyst, the study identifies usability characteristics deemed important from a developer's perspective. The study also contributes to the standardisation of metric collection across various Serious Games projects by defining and validating a set of in-game metrics spanning different domains. The study resulted in the successful development of a reusable in-game analytics software tool through an iterative development process involving feedback from game developers and a senior data analyst. The feedback identified key usability characteristics from a developer's perspective, highlighting the importance of understandability, ease of use, clarity, internal consistency and comprehensive documentation.

**Keywords:** *serious games, in-game analytics, emotional health, mental health, usability, game developers, user experience*



## Introduction

Serious Game (SG)s for emotional and mental health have gained significant attention over the past few years, as they provide an interactive and engaging way to test theories and new techniques for improving mental health and well-being [1] [2]. GEMH Lab (Gaming for Emotional and Mental Health) is a research and development lab that aims to establish the scientific foundations for effective mental health games<sup>1</sup>. Currently, each new SG the game lab develops requires the design and creation of a brand new and ad-hoc system for tracking in-game behaviour of the players, which can be time consuming and limiting. The in-game metrics are often used in conjunction with pre/post tests to validate, verify and predict the behaviour change of a given player, making them invaluable for the evaluation of the impact of a SG.

This study has the aim to better understand the process of creating a reusable in-game analytics software tool for evaluating an SG focused on emotional and mental health which is both easy to use and integrate and is flexible enough to be reused across different games. GEMH Lab mainly uses the Unity 3D engine <sup>2</sup> for the development of their video games, for this reason the software tool from this study will be a Unity plug-in, which will be created and tested strictly in Unity. The software tool will consist of two main components: the backend/database and the Unity plug-in. The backend/database will be responsible for the storage and management of the collected data, while the Unity plug-in will be responsible for the integration of the tool into the game.

The metrics supported by the tool are going to be defined by comparing and analysing the in-game metrics used by other games in the learning and physical health domains with metrics used in SGs for emotional and mental health from GEMH Lab. The comparison might be useful in order to understand whether these different domains share similarities on the type of metrics commonly collected. A

---

<sup>1</sup><https://gemhlab.com/>

<sup>2</sup><https://unity3d.com>

data analyst from the lab will be interviewed to confirm the usefulness of the identified metrics as well as to get feedback on the last version of the tool.

The usability characteristics needed for such a tool will be investigated during rounds of testings and interviews with game developers and stakeholders while using first prototypes of the analytics tool, and a final round of tests with game developers will be used to evaluate the extent to which the final prototype of the tool is meeting the previously identified usability characteristics.

## **1.1 Context**

Generally, the development of SGs for emotional and mental health involves multiple stakeholders with distinct roles and expertise. In our case, we will focus primarily on game developers and data analysts. Game developers are typically focused on the technical aspects of creating engaging and functional games. They excel in coding, design, and the technical implementation of game mechanics but may lack a deep understanding of the specific data requirements and the context in which the data will be used. On the other hand, data analysts are experts in data interpretation and statistical analysis. They possess a thorough understanding of the context and importance of the data but often lack technical proficiency in game development.

This divide in expertise can lead to significant challenges in the development and utilisation of in-game analytics. Miscommunication and a lack of shared understanding between game developers and data analysts can result in several issues. Game developers might collect excessive amounts of data, store it in inappropriate formats, or use naming conventions that are not intuitive for data analysts. Consequently, the data may not be fully utilised, leading to incomplete studies and wasted resources. The non-full utilisation of valuable collected player behaviour data also means that new insights might be overlooked, slowing down the research in the SG domain.

## **1.2 Problem Statement**

GEMH Lab (Gaming for Emotional and Mental Health) has developed a framework for the creation of a serious game. The framework is a set of steps/processes to go from a concept to a finalised and tested game. One of the steps involves evaluating whether the game is actually meeting its predefined targets (e.g. helping the player quit smoking, helping the player coping with rejection, etc.). This evaluation is conducted outside of the game (through pre and post tests), as the behaviour is itself exhibited outside of the virtual world. In-game analytics have been found to be a

good predictor for behaviour change outside the game in the education and learning domain [3], as well as in the mental and emotional health domains [4].

The primary goal of the project is to facilitate GEMH Lab and its game developers in the collection of in-game analytics pertaining to player behaviour within their games. Typically, the development of a new analytics system entails considerable time and effort. However, the aim here is to move towards the creation of a comprehensive and adaptable in-game analytics tool that can be seamlessly integrated into any game. By achieving this objective, the process of gathering crucial data regarding player behaviour will be streamlined, providing game developers with a generalised and reusable solution. This advancement will alleviate the need for extensive time and energy expenditure in creating a new analytics system for each game.

### 1.3 Research Questions

This study has the aim to answer the following research questions:

- **How can a generic in-game analytics software tool for Serious Games for emotional and mental health be created?**

This question will guide the whole study and it entails the two main aspects of this project: the type of in-game metrics collected by the tool and the usability of the tool from a game developer's perspective.

1. **How compatible are in-game metrics collected from the learning and physical health domain with metrics from Serious Games for emotional and mental health?**

This sub-question will help explore metrics found in studies of SGs in other domains and whether they are compatible with the metrics used in two of the games created by GEMH Lab. Given the high number of studies solely focused on learning and physical health as opposed to emotional and mental health, answering this question could help demonstrate whether the metrics used in other domains could be generalised and used in games in the emotional and mental health domain.

2. **What are usability characteristics of such a tool deemed important from a game developer's perspective?**

This sub-question will lead to the definition and evaluation of the usability characteristics of the different versions of the prototype created for this study. By first identifying the important usability characteristics and then

testing for their presence, the study will try to get closer to an in-game analytics tool which is both useful and usable by game developers.

## **1.4 Scope**

This study focuses on easing and improving the workflow needed to integrate and collect in-game metrics of a SG, as well as making sure that the collected metrics are useful to a data analyst. On the other hand, this study is limited to the actual collection of the in-game metrics in Unity and thus will not focus on improving the workflow for the data analysts after the players' data is collected. This study will solely focus on the usability of the Unity plug-in from the perspective of game developers, as that is the first technical step of the data collection workflow, where the in-game data is collected. The reason is that the technical implementation is both the crucial part of the data collection, as well as the point where the data analysts have to interact with game developers. Starting from such a critical part of the workflow can bring better understanding and help with improving the following steps too. The possible behaviour change resulting from the use of in-game analytics or of the tool created for this research is outside of the scope of the report and will be left for future work.

## **1.5 Report organisation**

The remainder of this report is organised as follows: in Chapter 2, the relevant theory and background of GEMH Lab and the games used for this research are presented. Chapter 3 is organised into three distinct studies, each with its own methods, materials, procedure, and results. Study #1 has the goal to define a set of metrics to be used in the in-game analytics tool. Study #2 involves the development of the prototype of the tool. Study #3 evaluates the usability of the prototype through user testing and feedback from game developers. The discussion of the main research question and its two sub-questions is contained in Chapter 4. And finally, the conclusions of the report are given in Chapter 5.

# Background

## 2.1 Research Topics

The study builds upon the results of a prior "Research Topics" report [5], which investigated the in-game metrics encountered through a literature review and which created a proposal for a set of generalised metrics that could be applied to a wider range of SGs. The following sections are taken from the report to provide context on the origin of the metrics used in the current study:

### Learning Domain

In the Learning domain, 5 papers have been found. The games goal range from learning spelling to weather prediction and environmental policies inquiry.

In their paper, Cariaga et al. [6], propose an analytics model for Game-Based Learning that captures game data to be presented in the Apple's Game Center through achievements and leader-boards. The paper stated that the model will be implemented in an educational game for mobile called Kinespell<sup>1</sup>, which has been modified to integrate the learning analytics process.

Ifenthaler et al. [7] discuss the development and validation of a Learning Analytics (LA) framework for analysing educational data to improve teaching and learning. The framework defines a range of parameters such as interest, motivation, response to reactive inventories, time spent on learning, task completion rate, assessment outcome, and support access. The authors have conducted case studies to validate the predictive accuracy of the learning profile. The predictive accuracy ranged between 58.63% and 81.69% depending on the classifier model used.

---

<sup>1</sup>Android version available at: <https://apkpure.com/en/kinespell/com.Kinespell>

The paper by Westera et al. [8] explores log files of an environmental policy game to identify relevant player behaviours and performance patterns. They found that there was substantial behavioural variability across students and that switching behaviour, or the rate of switching between different assets in the game, was a consistent and inherent personal trait. They established a model, a combination of real-time user data extraction and analysis during gameplay for personalised responses and an offline posterior logging analysis for quality assessment and game improvement, that uses switching indicators as predictors for the efficiency of learning and found that students who display increased switching behaviours need more time to complete the games. However, the authors noted that they did not have access to background profiles of students and could not make use of matched pre-tests and post-tests, questionnaires, direct observations, or a randomised trial with experimental groups and a control group. As such, the authors acknowledged that their interpretation of the findings is somewhat speculative and provisional, and further research is needed to corroborate their results.

The paper by Chen et al. [9] aimed to validate and inform the evidence-centred game design of a digital game-based assessment through game analytics of students' evidence trace files. The game is about storm chasing, where students play the role of storm chasers and use weather instruments to collect information on six storm features: cloud types, precipitation types, precipitation amount, wind speed, wind directions, and type of air movement. Based on this information, students are required to identify the type of storm they are chasing. The authors extracted 27 behavioural features as indicators of students' gameplay activities and used machine learning algorithms to identify the key features for prediction of students' mastery of the overall skill required by the game. The results showed that retry attempts on two assessment tasks were most influential for prediction, while performance features were not found to be influential. In addition, the authors used long short-term memory networks to model students' time-series behavioural features across multiple learning opportunities and found that five learning opportunities were sufficient for evaluating students' mastery of the overall skill.

Another paper by Westera et al. [10] presents an exploratory analysis of existing log files of the VIBOA environmental policy games. The games are inquiry-based, and the authors have identified "switching behaviour," defined as the number of game objects accessed per unit time, as a relevant behavioural pattern. Multiple regression analysis showed that switching rates of videos and locations explain 54% of the variance of learning efficiency. Additionally, switching behaviour as based on video access rates and location access rates is a predictor of learning efficiency,



and can account for 45% of the variance of total time spent.

In the Learning domain, emphasis is given to the performance of the player in completing activities, including activity duration and correct answers, and to their pre-test scores.

## **Health Domain**

In the Health Domain, 2 papers have been found. The games goals are weight-loss and physical health through walking.

The paper by Alamri et al. [11] describes a cloud-based serious game that uses physical exercise to promote weight loss in obese individuals. The game uses sensors to monitor various health and exercise-related parameters such as heart rate, weight, step count and calorie burn, which are accessible to therapists/caregivers who can provide real-time recommendations to adjust the intensity of the physical exercises.

The paper by Henriksson [12] presents the results of a research project about physical health that aimed to create a pirate-themed location-based mobile game and study how it was played using Game Analytics (GA). The goal was to validate the game's design in terms of healthiness and investigate whether game design could have a positive influence on a player's health. The paper describes the data gathering process, which involved developing three different game telemetry systems, and focuses on the third system that records most of the players' actions during the game, including time and location information. The paper also presents the findings from the analysis of the telemetry data, which showed that players could increase their physical activity by playing the game, particularly by walking between locations.

In the Health Domain, data about the players is either directly synced into the game, such as weight and steps count, or is inferred from their activities, such as calories burnt from step count and physical activity from distance between real-world locations.

## **Literature Reviews**

Two literature reviews have been found. One provided a list of metrics used by Serious Educational Game (SEG)s, while the other proposed a set of categories for

in-game metrics.

The paper by Daoudi et al. [13] presents a systematic literature review on the application of LA to SEG. The aim of the review was to identify the main features of an efficient use of SEGs in terms of success factors and learning outcomes, and to propose a multidimensional taxonomy for categorising these features. The authors also discussed the benefits and challenges of integrating LA approaches into these environments. The paper highlights the beneficial effects of SEGs on students' behaviour, cognition, and emotion but suggests that more empirical studies investigating data science techniques are needed to improve the usability of educational games.

The paper by Serrano-Laguna et al. [14] addresses the challenge of standardising the data collected from SG for LA. They propose an interaction model that tracks players' interactions and metrics commonly distilled from them. The authors analyse the current state of learning analytics, data standards, and specifications used in the field. They present an implementation of the model with the xAPI (eXperience API) specification and develop the Serious Games xAPI Profile to align with the most common use cases in the serious games domain. The paper reviews 14 serious games tracking players' in-game interactions and identifies the most commonly tracked interactions.

From the literature review, the in-game metrics encountered during the research have led to the creation of a set of metrics which integrates and combines them. Categories have been created from the 116 collected metrics presented in the previous section. Each of the individual metrics have been compared to the other ones for similarities. Any overlapping metric, for example having exactly the same name and goal, have been merged into one. With the new list, each element has been grouped to a metric with a higher degree of generalisation. Any duplicate groups have been merged once more. This process has been repeated until all the groups were distinct, and the metrics were found to be game-agnostic and usable by a wider amount of games.

The metrics are presented in Table 2.1, in conjunction with the reference of the studies that used them. Combining the metrics from studies in different SG domains, such as learning, physical health, and training, allowed us to create a set of generalised metrics that can be applied to a wider range of serious games. This approach makes the metrics less ad-hoc and more suitable for a larger set of domains, providing a better understanding of the game's impact on the player. The results are mostly in line with the systematic literature review by Daoudi et al. [13], as most of

Metric	Used By
Game Started	[6] [7] [8] [13] [10]
Game Stopped	[6] [7] [8] [13] [10]
Level Started	[6] [7] [8] [9] [13] [10]
Level Finished	[6] [7] [8] [9] [13] [10] [12]
Score Adjusted	[8] [9] [13] [10] [12]
Activity Performed	[6] [7] [8] [13] [10] [12] [14]
Alternative Selected	[6] [9] [14]
User Data Updated	[6] [8] [9] [13] [10] [11]
NPC Interaction	[7] [13]
Inter-Player Interaction	[7] [13]

**Table 2.1:** Overview Game Metrics from Research Topics

the new proposed metrics are used by games presented in their study.

For the evaluation of the effects of a SG, different methods to gather in-game data have been found. Out of the 9 papers analysed, 6 papers included gathering session data, such as logins, levels, and scores, as well as asking the player questions before, during, and after the game. Recording deeper interaction behaviour with other players and/or Non-Playing Character (NPC) has also been found to provide valuable data, whenever the game has that kind of interactions available, often through the collection of chat data.

## 2.2 GEMH Lab

GEMH Lab has developed over the years several games in the emotional and mental health domain. Among these, 2 have been made available by the company for the purpose of this study, namely Scroll Quest 2.0 (SQ2.0) and Hit-n-Run (HnR). They will be analysed with the purpose to provide a sample for understanding how in-game analytics are generally developed by the lab in the domain of emotional and mental health.

### 2.2.1 Scroll Quest 2.0

SQ2.0<sup>2</sup> is a game developed for research purposes, aimed at exploring the ways in which young individuals cope with rejection. Rejection Sensitivity, characterised by

<sup>2</sup><https://gemhlab.com/games/scrollquest2/>

an anxious anticipation, heightened perception, and exaggerated response to rejection, can contribute to a detrimental cycle of aggressive or withdrawn behaviours, eventually leading to the development of severe mental health issues such as depression [15]. The primary design objective of SQ2.0 was to develop a rejection experience that is more immersive and distinctive compared to the commonly employed, standardised task for assessing rejection and isolation known as Cyberball.



Figure 2.1: Scroll Quest 2.0 - Screenshots

## 2.2.2 Hit-n-Run

HnR<sup>3</sup> is a mobile game which offers an opportunity to enhance impulse control and promote team collaboration while aiding individuals in their efforts to quit addictive behaviours. The game and related studies focus on identifying potential key targets for intervention in addiction among adolescents. Specifically, the Smoking Cessation project aims to address impulsivity by targeting inhibitory control, stimulus valuation, and the influence of social networks [16]. The Go/NoGo task, a commonly employed method, is utilised to assess the level of inhibitory control. Participants are instructed to press a button when a Go stimulus is presented and withhold the response when a NoGo stimulus is displayed. The primary objective of HnR is to investigate and foster alternative behavioural practices among adolescents prior to the development of habitual smoking habits.

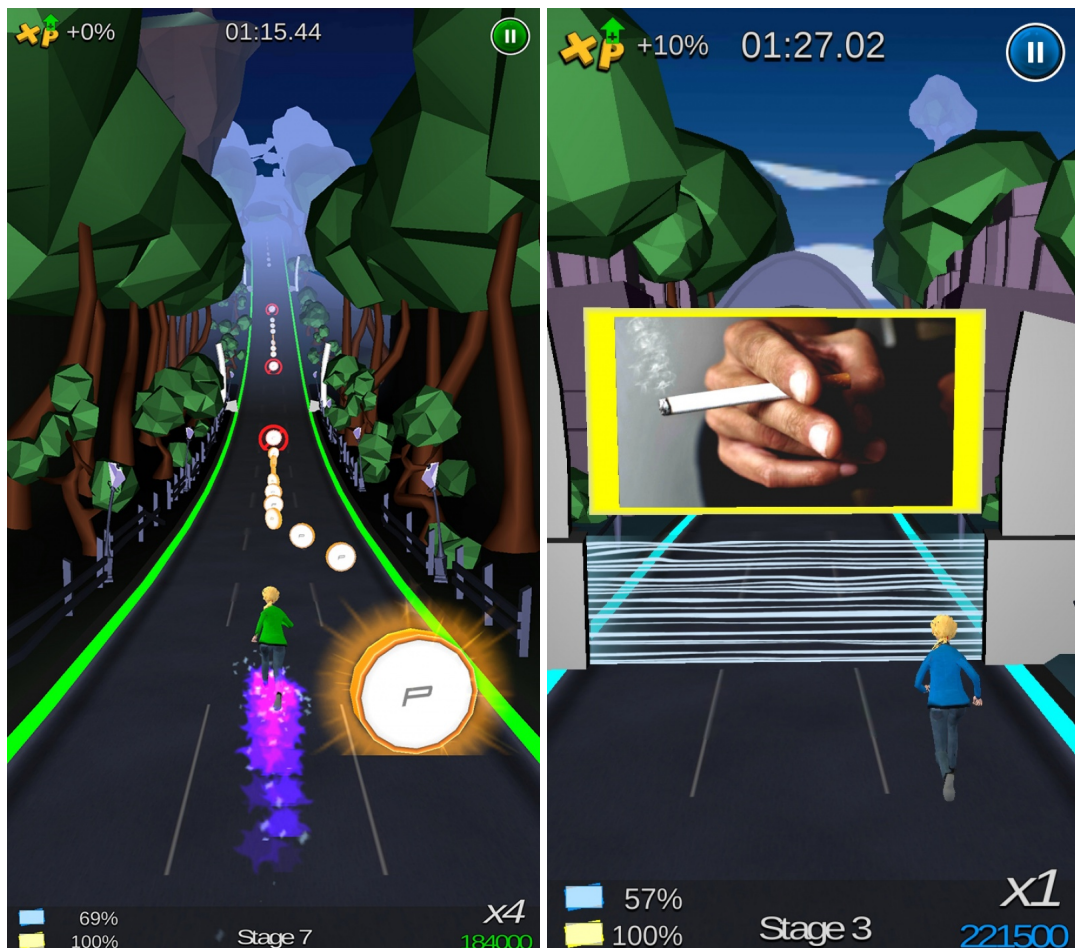


Figure 2.2: Hit-n-Run - Screenshots

<sup>3</sup><https://gemhlab.com/games/hit-n-run/>

## 2.3 Summary

This chapter reviewed previous research on in-game metrics across learning and health domains. It presented a set of generalised metrics derived from the literature, applicable to various serious games. The chapter introduced two games developed by GEMH Lab: Scroll Quest 2.0, which investigates young individuals' responses to rejection, and Hit-n-Run, which targets impulse control and team collaboration in addiction prevention. These games exemplify the application of in-game analytics in emotional and mental health research, providing context for the current study's analysis and the base for the Study #1 presented in the following chapter (in Section 3.1).

The chapter provides a foundation for the research, as the in-game analytics metrics previously presented will become the starting point of the studies in this report; this is based on the assumption that in-game analytics metrics across different SGs in different domains have at least some similarities which can be leveraged when attempting to create a reusable in-game analytics collection tool. This assumption spawns from the work done in the Research Topics report [5], as it showed that although games had different genres, domains and goals their metrics were all compatible.

# Studies

This study is divided into three smaller studies, each helping towards the goal of creating a prototype for a reusable in-game analytics tool for Unity. Study #1 (Section 3.1) will focus on identifying the most suitable metrics for the tool to be developed in Study #2 (Section 3.2), where a prototype of the in-game analytics tool will be iterated upon with game developers. Study #3 (Section 3.3) has the goal to evaluate the usability of the final version of the tool.

## 3.1 Study #1 - Game Metrics Validation

The first study will be conducted in partnership with GEMH Lab who will offer insights on selected past games through their senior data analyst, along with granting access to the source code of one of these titles.

This study serves to answer *"How compatible are in-game metrics collected from the learning and physical health domain with metrics from Serious Games for emotional and mental health?"* (Research Question #1 1), and it has the practical goal of defining the set of validated metrics to be made available within the tool.

### 3.1.1 Methods

#### 1. Metrics Collection

The study will begin with the list of metrics identified in the related Research Topics study [5], presented in Materials (Section 3.1.2), which will be used to represent the learning and physical health domains.

The metrics used in Scroll Quest 2.0 (SQ2.0) and Hit-n-Run (HnR) from GEMH Lab, presented and used in Results (Section 3.1.4), will be used to represent the emotional and mental health domain.

The metrics will be extracted from the projects by using a mix of automated text search tool within the whole Unity codebase and a manual search of each script file contained within the projects.

## **2. Metrics Comparison Across Domains**

The 3 sets of metrics will be compared in order to identify similarities and differences between the different domains and, if needed, a new list of metrics will be created to include any metric deemed missing. Afterwards, the finalised list of in-game metrics will be validated by conducting an interview with a data analysts from GEMH Lab, in order to confirm the usefulness and completeness of the metrics from the studio's own business context.

Each game's metric will be compared to the already existing list for compatibility: if the metric could still be collected, nothing is changed, else a new metric will be created to account for it.

## **3. Metrics Validation with Data Analyst through Interview**

The data analyst will participate in a single semi-structured interview (Interview Questions can be found at Appendix A.1) aimed at eliciting insights into their experience and perspectives regarding the selection of metrics and their role in the workflow surrounding the analysis of players' data of a game for emotional and mental health. During the interview (Questions in Appendix A.1), the data analyst will be encouraged to express concerns, offer opinions, and propose suggestions regarding the selected game metrics for the prototype.

In the creation of the questions, the following a priori themes have been outlined:

### **A Priori Themes:**

1. Importance of In-Game Analytics
2. Use of Game-Specific Metrics
3. Ethical Data Collection
4. Data Processing Challenges

These themes show the interest in better understanding the role of in-game analytics for SGs in the emotional and mental health, how metrics are selected and change across different projects with different goals. A focus on ethics as the topics of these games tend to be on a deeper level and might be important for the game development team to address concerns of privacy and safety. And finally, exploring



any challenges encountered in previous games concerning with the processing of the data by the data analyst would be useful to understand and be aware of common pitfalls when developing the reusable plug-in.

Following the interview phase, the procedure of Thematic Analysis will be employed to analyse the collected data. The process will involve the following steps:

1. **Transcription:** Transcribe the interviews verbatim, ensuring accurate representation of the responses provided by the data analyst.
2. **Coding:** Thoroughly analyse the transcribed interviews, identifying key concepts, patterns, and recurring themes within the dataset.
3. **Analysis of Themes (A Priori and Emergent):** Based on the coded data, assign them to the a priori themes which encapsulate the main ideas, perspectives, and insights shared by the data analyst. These themes will serve as the foundation for extracting meaningful conclusions and actionable recommendations. Any data not fitting the a priori themes will be the base of emergent themes.

Once the themes and recommendations have been analysed, any actionable requirement coming from the data analyst perspective will be included as to make sure the resulting prototype takes into consideration aspects related to the person using the game data for the next steps.

### 3.1.2 Materials

The metrics to be extracted will be compared to the following metrics list, which contains the metrics collected during the previous Research Topics report [5]:

- Game Started
- Game Stopped
- Level Started
- Level Finished
- Score Adjusted
- Activity Performed
- Alternative Selected
- User Data Updated
- NPC Interaction
- Inter-Player Interaction

After the metrics have been compared and a new metrics list finalised (if necessary), a single semi-structured interview using an audio/video call with the data analyst from GEMH Lab has been used (Full interview questions in Appendix A.1).

### 3.1.3 Procedure

The codebases for SQ2.0 and HnR were accessed to retrieve all unique logging events collected within the games. Initially, a text searching tool was utilised to scan the entire project codebase for relevant logging events. Subsequently, a manual verification step was performed, involving the examination of each script available within the Unity project, to ensure that no logging events were overlooked. The manual verification process also facilitated a comprehensive understanding of the context in which the metrics were employed within each game. Once all the metrics for the two games were listed and comprehended, each individual metric was compared against the metrics list presented in the Materials section (Section 3.1.2) to determine whether they were compatible with any previously defined metric or represented a new metric. With the resulting finalised metrics list, a semi-structured interview was conducted with the data analyst from the GEMH Lab to gather additional information and further verify the metrics list; the interview was performed fully online and recorded upon participant's consent.

### 3.1.4 Results

#### 1. Metrics Collection

For each of the games provided by GEMH Lab, the metrics used during the game have been analysed and collected to be compared:

##### **Scroll Quest 2.0 (SQ2.0)**

The analytics data recorded in SQ2.0 has been analysed and collected to better understand what metrics are important for such a game. The events collected are subdivided in the following 4 categories:

##### **Categories:**

##### **1. General**

| Category for system-wide metrics (e.g. game started and game ended)

##### **2. Sync Marker**

| Event for multiplayer synchronisation events

##### **3. Combat**

| Category for all events related to combat, they include the initiating player, the target player and other contextual information of the combat. Item collection, healing, death and entering safe zones are also part of this category.

#### 4. **Interaction** (e.g. trading and voting)

| Category for all social interactions, including trading, player upgrades and voting against other players.

Each category type, except for #2, has a set of metrics to collect granular information of the player's behaviour in-game.

The list of specific individual events is the following:

##### 1. **# General**

- Init
- Quit
- Disconnect

##### 2. **# Sync Marker**

##### 3. **# Combat**

- |                 |                  |              |
|-----------------|------------------|--------------|
| • Start         | • Attack         | • Collect    |
| • End           | • Heal           | • Emote      |
| • Objective     | • TakeDamage     | • AloneStart |
| • SafeZoneEnter | • Death          | • AloneEnd   |
| • SafeZoneExit  | • ItemPickup     | • Spectate   |
| • ZoneEnter     | • ItemDrop       | • IdleStart  |
| • ZoneExit      | • HealerDepleted | • IdleEnd    |

##### 4. **# Interaction**

- |              |                 |             |
|--------------|-----------------|-------------|
| • Start      | • TradeRequest  | • VoteEnd   |
| • End        | • TradeResponse | • Vote      |
| • TradeStart | • Upgrade       | • VoteLoser |
| • TradeEnd   | • Downgrade     |             |
| • TradeSent  | • VoteStart     |             |

#### **Hit-n-Run (HnR)**

The same procedure has been done for HnR. The game contained only the following analytics events:

1. Player Profile Updated
2. Level Completed

Event #1 was sent whenever the player updates its profile, including the modified data, and metric #2 triggered whenever a level was completed, containing information regarding the timing, score, etc.

## 2. Metrics Comparison Across Domains

The procedure employed to determine the game metrics for the initial version of the prototype involved a synthesis of metrics derived from both the Research Topics [5] and the two aforementioned games from GEMH Lab.

These metrics were examined and commonalities were identified, resulting in the consolidation of overlapping metrics into a unified list.

The updated game metrics chosen for the prototype development are the following:

- Game Started
- Game Stopped
- Level Started
- Level Finished
- Score Adjusted
- Activity Performed
- Alternative Selected
- User Data Updated
- NPC Interaction
- Inter-Player Interaction

It is to be noted that the list of metrics derived from the RT already covered the specific metrics obtained from the two games from GEMH Lab.

## 3. Metrics Validation with Data Analyst through Interview

As a first step, the full interview has been transcribed to be able to better analyse it. Following transcription, coding was conducted to systematically organise the information into chunks which could be further processed (Codes found in Appendix A.2). From the coded data, several new themes emerged (Themes generation found in Appendix A.3):

### **Emergent Themes Found:**

1. Behavioural Analysis Goals
2. Impact Evaluation Approaches

### 3. Toolbox for Researchers and Designers

For better analysis, the topics discussed during the interview are divided into their related theme:

#### **Importance of In-Game Analytics**

During the interview the data analyst highlighted the need for these metrics to be sophisticated and specifically tailored to the game's objectives. For instance, in the game HnR, where accuracy was crucial, reaction time data was initially captured. However, reliability issues led them to prioritise accuracy metrics. Conversely, Mind-Lights, aimed at reducing anxiety, focused on measuring approach and avoidance behaviour through player actions within the game, directly linking to the core game-play loop.

#### **Ethical Data Collection**

While capturing basic metrics like points and leaderboards was straightforward, the analyst highlighted the ethical considerations of protecting player data beyond the game itself; in MindLights they chose to focus on analysing anonymised leaderboard data downloaded from the game's social platform (Google Hangouts). The analyst also noted the potential value of capturing and analysing textual data like player comments for deeper insights, but acknowledged the complexities involved.

#### **Data Processing Challenges**

Regarding the data handling, previously, the analyst had encountered difficulties with exporting accuracy and reaction time data into various tools due to a lack of a participant-centric data structure. For example, issues with GameSparks data output made participant identification challenging. While Unity dashboards were also used, their formatting and comprehensibility sometimes presented problems, which meant that it often necessitated manual data conversion, a time-consuming process that could be avoided with a more user-friendly system and the use of macros.

#### **Behavioural Analysis Goals**

Beyond data collection and handling, the analyst discussed the goals of behavioural analysis and expressed interest in tracking changes over time, observing trends like reduced mistakes and faster reaction times in players. In MindLights this translated to monitoring a shift in "approach" versus "avoidance" behaviour as players progressed. These observations resonated with existing research that linked positive language use in online communities with better smoking cessation outcomes, suggesting a potential meta-view for categorising such behaviours across different games. This finding further reinforced the notion that in-game behaviour can indeed

correlate with real-life actions.

### **Impact Evaluation Approaches**

Limited budgets often necessitated shorter intervention periods, making long-term impact assessment difficult. The challenge lies in transferring the learning gained within the game to real-world situations. While statistical analysis provided valuable insights, the analyst acknowledged the complexity of unaccounted variables that could influence outcomes.

### **Toolbox for Researchers and Designers**

The ideal data analytics system for SGs, according to the data analyst, would serve a dual purpose – as a data analysis tool and a guide for game design. They emphasised the need for user-friendly explanations/documentation to bridge the knowledge gap between researchers and game designers. Clear communication of the backend functionalities and limitations from researchers is deemed crucial for game designers to fully leverage the potential of in-game data.

### **Use of Game-Specific Metrics**

Additionally, the interview highlighted concerns from the data analyst regarding the need for flexibility in metric integration, regardless of the chosen metrics. To address this concern, a **Custom event** will be incorporated into the prototype. This feature allows game developers to integrate additional metrics not initially included in the predetermined list, catering to the evolving needs of game designers and data analysts.

Below is presented the full list of in-game metrics which will be used in the prototype for the following studies:

### **In-Game Metrics**

- Game Started
- Game Stopped
- Level Started
- Level Finished
- Score Adjusted
- Activity Performed
- Alternative Selected
- User Data Updated
- NPC Interaction
- Inter-Player Interaction
- **Custom Event**

## 3.2 Study #2 - Prototype Development

After having defined and validated the set of in-game metrics to be used within the prototype, the prototype can now be developed.

The second study is focused on the development of a prototype for a reusable in-game analytics plug-in for Unity. The overall goal of these iterations is to create a first minimum viable product, referred as "prototype", that can be used, tested and evaluated to help answering *"How can a generic in-game analytics software tool for Serious Games for emotional and mental health be created?"* (Main Research Question 1.3), which seeks to better uncover usability criteria for the creation of a generic in-game analytics plug-in, through the feedback of game developers with different background experience. In summary, this study will produce a prototype improved through two rounds of testing with game developers.

### 3.2.1 Methods

The development of the prototype will be divided into 3 iterations: the first two will serve as a way to gather feedback about the usability of the plug-in from a game developer's perspective, each round influencing the development of the following one, while the last iteration will be used to collect remarks and comments about usability aspects of an analytics plug-in deemed important by a game developer with experience in Serious Games focused on emotional and mental well-being (presented in Study #3 in Section 3.3).

After the first iteration of Study #2 has been completed, target users (e.g. game developers) will be asked to participate to rounds of testing and evaluations of the tool. The target users will be given instructions on the goals of the test, which will require them to integrate and use the tool in a given project.

Before starting the testing, the participants will be given introductory tasks to get used to the format and the environment. These include setting up a new Unity project with the provided template, importing the files and all the basic data. Only once everything has been correctly setup, the evaluation will begin.

These formative sessions will be carried out in the form of usability tests, where they will be asked to fully integrate the plug-in in a given Unity project, as well as interviews aimed at discovering crucial pain points and important usability features. At the end of each session a structured interview will be conducted with each participant to gather their feedback (Questions in Appendix A.4). User feedback and field observations, collected by the researcher during the tests, will then be used to

identify the usability characteristics they deem most important.

### 3.2.2 Materials

For the evaluation of the plug-in with game developers, a standard and straightforward Unity tutorial named "Roll-a-Ball"<sup>1</sup> was selected as the foundational project. This choice aimed to reduce potential friction associated with the project itself, considering its inclusion in Unity's official "Basic Scripting"<sup>2</sup> course. By opting for a widely recognised and accessible tutorial, the intention was to create a familiar and approachable environment for developers engaging with the plug-in. This decision was made in recognition of the importance of minimising barriers to entry and ensuring that the evaluation process was conducted in a context conducive to meaningful feedback and assessment.

### 3.2.3 Procedure

The participants were selected based on their experience in developing games using the Unity game engine. Six game developers of varying experience, nationalities, genders and ages have been chosen to provide feedback on the prototype. The participants were distributed across different iterations to maintain a diverse mix of age groups and experience levels, thereby minimising the potential impact of these factors on the results.

Each development phase was carried out based on the feedback obtained from the previous feedback session. The iterations are presented as follows:

- **First Iteration:** Based on interview with GEMH Lab data analyst
- **Second Iteration:** Based on first user feedback session with game developers
- **Third Iteration:** Based on second user feedback session with game developers

For each user feedback session the following structure has been used:

1. User Testing with Testing Tasks A.3
2. Structured Interview A.4

---

<sup>1</sup>Roll a Ball: <https://assetstore.unity.com/packages/essentials/tutorial-projects/roll-a-ball-tutorial-complete-urp-77198>

<sup>2</sup>Beginner Scripting course: <https://learn.unity.com/course/beginner-scripting>



## Tasks

The testing tasks assigned to the game developers are the following (original document available A.3):

1. Verify that plug-in is installed correctly
2. Add Level Started event with current level (e.g. "Level 1") in GameManager.cs
3. Add Level Finished event with current level (e.g. "Level 2") in GameManager.cs
4. Add Score Adjusted event every time the score is updated (e.g. "10")
5. Add Activity Performed event when using booster ("Booster Used")
6. Add Custom event when the game ends with current score ("High Score")
7. Add Custom event with total time played when the game ends ("Time Played")

Each session has been fully performed online via an audio/video call with the participant's screen shared to the researcher. Thinking aloud was used to allow the participants to share their thoughts as they were performing the tasks, field notes and other observations were collected during the session by the researcher. Each comment, note and feedback was then collected into a single document and used as a base for detecting usability issues and suggesting possible improvements.

## 3.2.4 Results

### 3.2.4.1 First Iteration

In the first iteration, the prototype encompasses five components aimed at establishing the foundational infrastructure for our envisioned plug-in. The following elements are the key parts of the initial iteration:

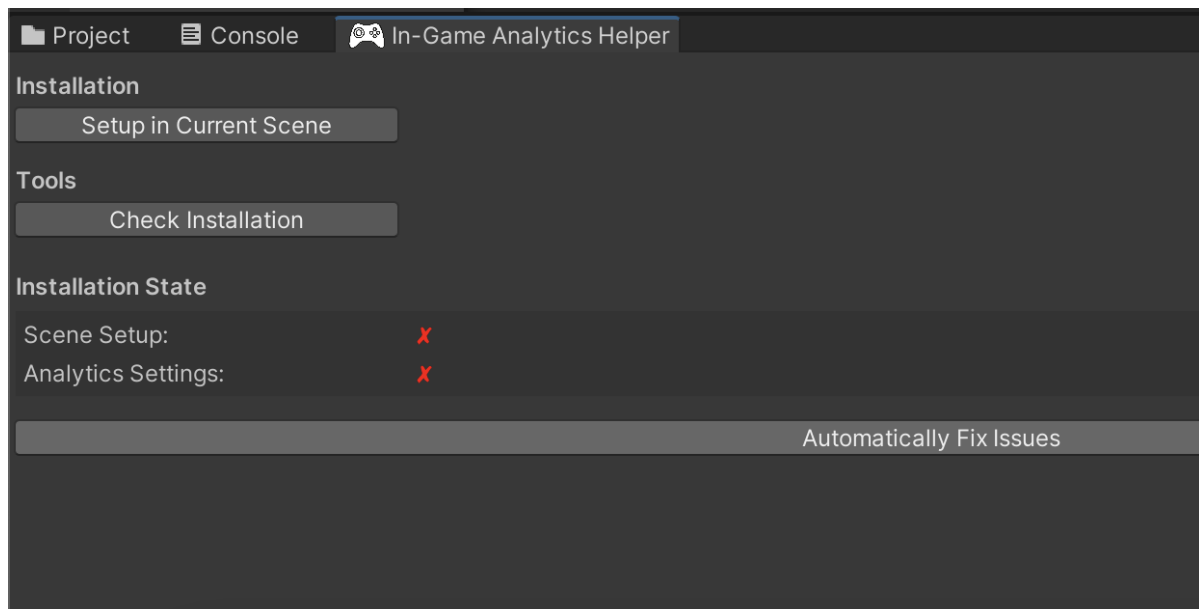
**1. User Interface (UI) Window:** The prototype features a simple UI window designed to facilitate the automatic installation and validation process of the plug-in. Its primary function is to visually alert users in case any crucial component necessary for the plug-in's functionality is missing. This window serves as the entry point for users to get started with the plug-in (see Figure 3.1).

**2. Console:** Real-time feedback is provided to the game developer as the game is running via the built-in Unity console. It shows whenever a game event is triggered.

**3. Game Metrics Logging Library:** Accompanying the window is a library created to facilitate the logging of game metrics, as outlined in Section 3.1.4. The user can directly start adding logs into the game as all the setup is automatically done for them.

**4. Automated Event Logging:** The prototype also incorporates automatic logging functionality for essential events within the game environment. Basic events such as 'Game Started' and 'Game Stopped' are logged, along with timestamps. This automated logging mechanism offers users to get started collecting important player data without manual setup.

**5. README File:** Included in this iteration is the initial version of a README file (see in Appendix A.4) aimed at guiding users through the setup and utilisation of the plug-in. This document serves as a comprehensive reference point, offering step-by-step instructions and essential information regarding the installation, configuration, and usage of the plug-in. It is intended to provide users with clear and concise guidance, as well as a knowledge base to explore all the available game metrics and how to use them.



**Figure 3.1:** First Iteration - Unity Plug-in (User Interface - Zoomed In)

## 1. Prototype Evaluations

The evaluation of the first iteration of the prototype was conducted with three game developers. The age range of the participants spanned from 22 to 38 years, and their experience in game development varied from less than one year to over ten years. Three of the participants were native English speakers, while the other three spoke English as their second or third language. The complete data and feedback from the evaluations are available in Appendix A.5.

The table below 3.1 shows the summary of the results of the participants; for each task (full list available at Section 3.2.3), the task number ("**#**"), the number of incorrect actions performed ("**Errors**"), the time needed to successfully complete the task ("**Time**") and the time needed for the participant to be sure that the task has been performed correctly ("**Verify**"):

Summary					
#	Errors	Average		Median	
		Time	Verify	Time	Verify
1	0	00:17	00:00	00:15	00:00
2	4	02:35	01:36	01:47	01:00
3	0	01:22	00:30	01:21	00:31
4	0	01:13	00:22	01:04	00:28
5	0	00:47	00:16	00:47	00:17
6	2	03:28	00:49	02:47	00:45
7	0	01:30	00:21	01:19	00:21

**Table 3.1:** Round 1 - Summary [Time format mm:ss]

The table shows that errors were committed during task 2 and 6, increasing the overall time needed to perform the task and to verify it. For context, task 2 is the first practical use of the library in code and task 6 is related to the setup of a Custom event.

## 2. User Feedback

In the feedback interviews with game developers, several comments and queries surfaced regarding the functionality and usability of the custom event system within the game development environment (Feedback interview answers: A.6).

The most interesting comments and quote from the feedback and interviews are presented below for discussion:

- **User 1**

1. Shouldn't the value [of custom event] be shown here [console]?

• **User 2**

2. Why am I seeing additional events? Where are they coming from?

3. I feel like I should be seeing Level 1 in the console.

• **User 3**

4. I expected to see something in the Helper viewer for the new event.

5. I was expecting that the value of the event would be shown in the console to make sure it is correctly set up.

6. **Feedback:** If it was possible, which events were already available (in that scene)

7. **Feedback:** I would like to know what scene is being selected or which scene the plugin in being loaded.

A recurring point within the comments centred on the need for increased visibility and user control over custom event data within the console. User 1 specifically requested that the value of custom events be directly displayed in the console. Similarly, User 3 expressed a desire to see the value of custom events displayed in the console; the participant placed emphasis on the value this information holds for verification purposes, which highlights the importance of the console as a central hub for monitoring and verifying the behaviour of the plug-in. User 2 was surprised by automated events appearing in the console, suggesting a lack of understanding of the existence of automated events coming from the tool itself. They also indicated an expectation of seeing "Level 1" displayed in the console, suggesting a need for improved clarity and formatting of the events displayed in the console, as the value was present, just not obviously visible to them. User 3 also provided explicit feedback related to the integration between the development tool and the existing UI window: they expected to be provided with information about the integrated events, in order to get clear feedback that their additions are being correctly implemented. Moreover, they also expected feedback regarding the currently selected scene to make them at ease on where the plugin was being installed.

In conjunction with the console, the UI window seems to be a valuable and key component for providing information and timely feedback to the game developers, in order to make them at ease regarding their usage of the plug-in.

### 3. Discussion

The first round of evaluations provided valuable insights into the strengths and areas for improvement of the prototype. Key learnings acquired from the user feedback and observed user behaviour include:

1. **Surprise in Noticing Automated Events:** A user was surprised upon encountering automated events in the console. The nature of these events highlighted a need for improved communication and transparency regarding the plug-in's functionality and automated processes (Quote: #2 | 3.2.4).
2. **Errors Performed When First Using the Library:** Several users committed errors when initially using the library for logging game metrics. These errors underscored the importance of refining the library's documentation and providing clearer instructions for seamless integration and usage (Evaluation Results - Task #2 | 3.1).
3. **Errors Performed When First Using a Custom Event:** Users faced challenges and committed errors when attempting to implement custom events within the game. The complexity of the event implementation process revealed a need for enhanced guidance and support to facilitate smoother integration and customisation (Evaluation Results - Task #6 | 3.1).
4. **Lack of Clear Feedback in the Console:** Users expressed dissatisfaction with the clarity of feedback provided in the console. Insufficient information in the console output hindered users' ability to interpret and understand logged events effectively (Quotes: #1, #3, #5 | 3.2.4).
5. **Lack of Event Information in the UI Window:** A user noted a deficiency in event information presented within the UI window, more precisely the lack of feedback whenever a new event was added and on which scene the tool will be installed. This would allow developers to not have to run the game to see if the events have been correctly created, as well as being certain that they are installing the plug-in in the right location (Quotes: #4, #6, #7 | 3.2.4).

From these findings, the following changes are proposed to address the issues and improve the usability of the tool for the next iteration.

#### **Proposed Changes:**

1. **README:** Add an explanation of automated events.
2. **README:** Add an explanation of metrics and their possible uses.

3. Improve feedback of events to the console (improve styling for clarity, add event values).
4. Add a list and count of analytics events to track how many got added (if possible granularity per file).
5. Show the name of the "current scene" in the plug-in window for clarity.

### 3.2.4.2 Second Iteration

In the second iteration, the prototype undergoes enhancements and updates to further solidify its functionality and user-friendliness. The following components have been refined and improved based on feedback and proposed changes:

**README File:** The README file (see in Appendix A.6.3) receives several updates, now containing even more detailed information on the potential uses and examples of each game metric. Specific details regarding Automated Events have been incorporated, providing comprehensive insights for game developers regarding their existence and implementation. This expanded documentation aims to address proposed changes 1 and 2, offering users a comprehensive resource for understanding and utilising the plug-in effectively.

**Console Logging:** Many improvements have been made to the console functionality (Figure 3.2). Now, whenever an event is triggered, the console displays more comprehensive information regarding the event, including both the triggered game event and its corresponding values. Additionally, the formatting of the console output has been optimised to enhance readability, particularly when dealing with complex data structures. This update addresses proposed change 3, enhancing the user experience by providing clearer and more informative feedback.

**UI Window:** A new feature has been introduced to the UI window, presenting users with a real-time count of analytics events present in the project (Figure 3.3). This addition offers game developers a convenient means to obtain a quick overview of the entire analytics setup without the need to consistently run the game. By providing instant visibility into the analytics infrastructure, this feature streamlines the development process and facilitates efficient monitoring of game metrics. The inclusion of this feature addresses proposed change 4, enhancing the accessibility and usability of the plug-in. The current selected scene name has also been added to help the game developers know where the tool will set up, in order to address proposed change 5.

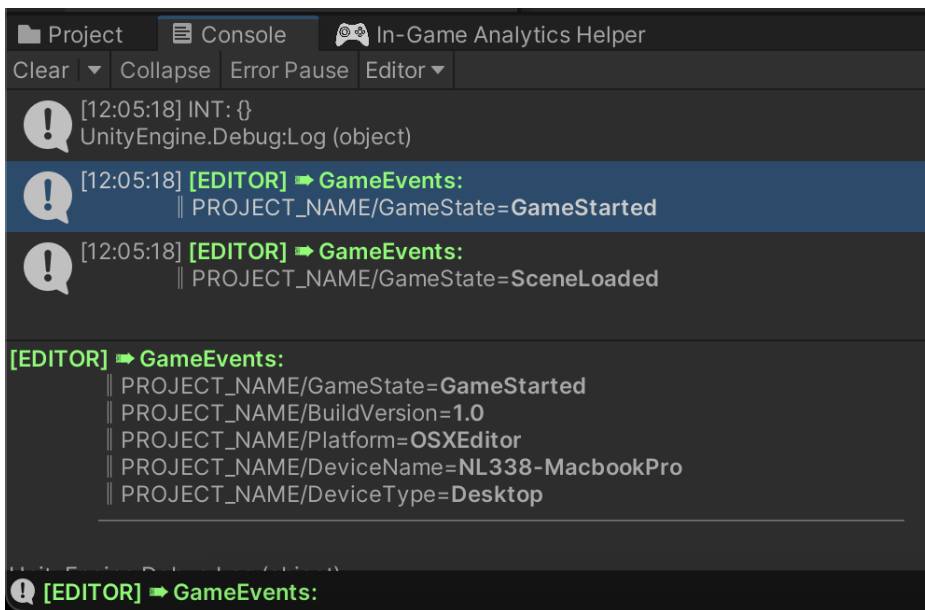


Figure 3.2: Second Iteration - Updated Tool Console

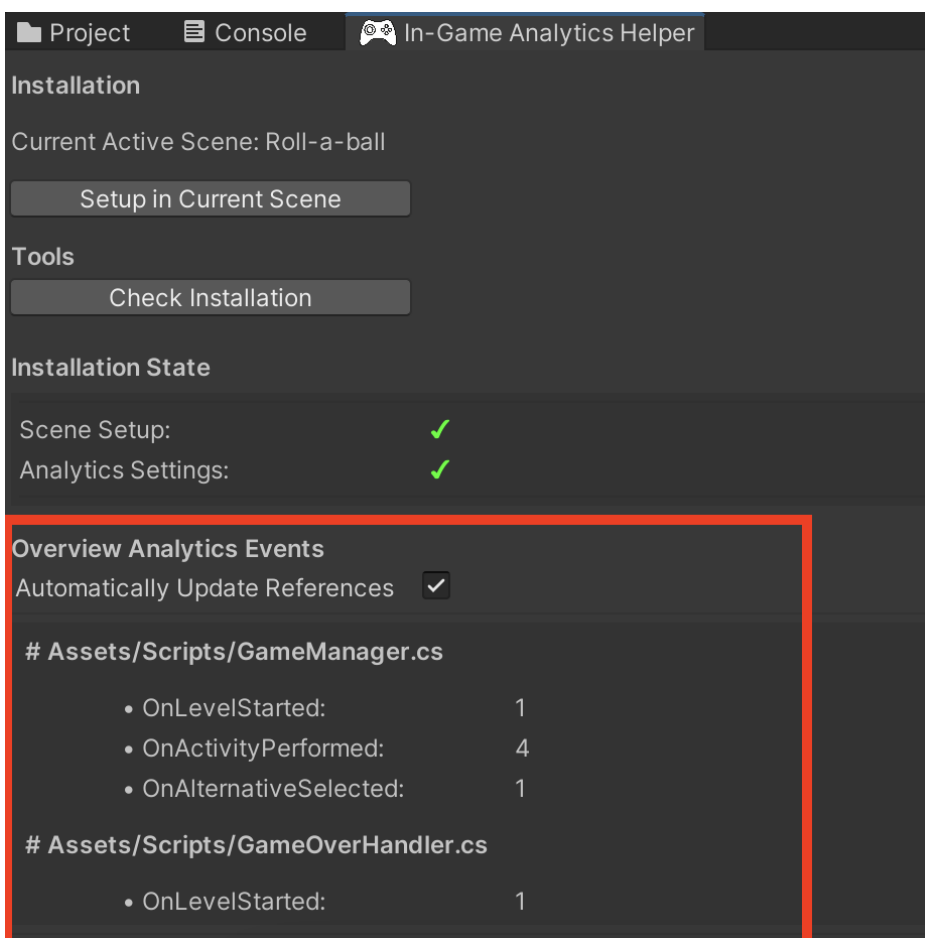


Figure 3.3: Second Iteration - Updated Tool Window



## 1. Prototype Evaluations

The evaluation of the second iteration of the prototype followed a similar methodology and procedure as the previous iteration, engaging a new set of three game developers with varying degrees of experience in Unity development, spanning from novice to seasoned professional. Detailed results of this evaluation can be found in the appendix (see in Appendix A.7). The summary of participants' results is presented in Table 3.2, outlining the performance metrics for each task, including the number of errors, time taken for successful completion, and verification time. Notably, in this iteration, no errors were recorded across all tasks, irrespective of participants' expertise levels in Unity and game development:

Summary					
#	Errors	Average		Median	
		Time	Verify	Time	Verify
1	0	00:21	00:03	00:20	00:02
2	0	01:04	00:24	01:05	00:25
3	0	00:37	00:17	00:38	00:20
4	0	00:42	00:13	00:39	00:14
5	0	00:35	00:09	00:41	00:08
6	0	00:29	00:16	00:29	00:17
7	0	00:48	00:19	00:21	00:18

**Table 3.2:** Round 2 - Summary [Time format mm:ss]

A comparison between the results of Round 1 (see Table 3.1) and Round 2 is provided in Table 3.3, negative differences mean that the task was better performed by participants in Round 2 (i.e. the task took less time to be completed), while a positive number shows an increase of time needed to complete the task and thus a decrease in speed:

It is noteworthy that Task #2 and Task #6 exhibited flawless execution in both rounds, with overall improvements observed in timing metrics across tasks. Notably, the median time for Task #1 showed a marginal increase of 5 seconds in the current round, while all other metrics demonstrated either improvements, more or less pronounced, or remained consistent.

## 2. User Feedback

In the second round of feedback interviews with game developers (Feedback interview questions in Appendix A.4), additional insights were gathered regarding user expectations and the learnability of the custom event system (Feedback interview

Summary					
#	Errors	Average		Median	
		Time	Verify	Time	Verify
1	0	-00:04	00:00	<b>+00:05</b>	00:00
2	-4	-01:31	-01:12	-00:42	-00:35
3	0	-00:45	-00:13	-01:09	-00:40
4	0	-00:31	-00:09	-00:25	-00:14
5	0	-00:12	-00:07	-00:06	-00:09
6	-2	-02:59	-00:33	-02:18	-00:28
7	0	-00:42	-00:02	-00:58	-00:03

**Table 3.3:** Round 2 - Differences from Round 1 [Time format mm:ss]

answers in Appendix A.8). The comments and quotes provided by the developers shed light on both positive aspects and areas for improvement within the system:

- **User 4**

1. I thought that the tool would automatically pick up when to call the events and I shouldn't need to put them in the correct place.
2. I wasn't sure if the levelId was optional or if they both needed to be set.

- **User 5**

3. The system seems quite neat and the code is nicely available to understand the inner workings of the system, no magical black box.

- **User 6**

4. I like that the console is colourful and showing the events.
5. I can see that the event is there [Tool Window], I wish I could see the parameters of the events too.
6. I was expecting a key-pair for each, so I could mentally match them but I guess it's fine like this too.

User 4 vocalised their assumption that the tool would automate event placement within the code, highlighting a potential gap between user expectation and the current functionality. Additionally, confusion arose regarding the optional nature of certain parameters, underscoring the need for clearer documentation and guidance to prevent confusion with developers not experienced with the plug-in and programming in general.

On the other hand, User 5 offered positive feedback on the overall organisation and accessibility of the system. They praised the system's transparency, noting the availability of code for understanding its inner workings. This feedback highlights the importance of providing experienced developers with access into the underlying code and architecture of the tool to facilitate a deeper comprehension and trust.

User 6 provided a mixture of positive and constructive feedback regarding the system's user interface: they appreciated the colourful and informative display of events within the console, indicating a positive experience with the visual feedback provided. However, they also expressed a wish to see event parameters displayed alongside the event itself in the tool window, highlighting a potential need for an even more comprehensive view of event data within the UI. The participant also expressed a preference for a key-pair format for events; while the current format might be functional, a key-value pair layout could improve mental model alignment for users expecting a more structured and direct presentation.

### 3. Discussion

The second round of evaluations provided further insights into the strengths and areas for improvement of the prototype, as well as feedback on changes applied in this current iteration. Key learnings acquired from the user feedback and observed user behaviour include:

1. **Mismatch in Mental Model for Custom Events Data Structure:** A user expressed a preference for a key-pair format for Custom Events over using arrays, indicating that while the current format might be functional, a key-value pair layout could improve mental model alignment for users expecting a more structured and direct presentation. This presents a possible decrease in usability when using Custom Events with a larger amount of data to be logged (Quote: #6 | 3.2.4).
2. **Ambiguity in Optional Argument:** Uncertainty regarding the necessity of arguments like "levelId" arose, indicating a need for clearer documentation on optional parameters within the system (Quote: #2 | 3.2.4).
3. **Unclear Game Developer's Task:** Users assumed the tool would automatically detect and place events within the code, highlighting a discrepancy between user expectations and the current functionality (Quote: #1 | 3.2.4).
4. **Lack of Event Parameters in the UI Window:** Users expressed a wish to see the event parameters displayed alongside the events in the tool window, highlighting a potential need for a more comprehensive view and feedback of event data within the user interface to the game developers (Quote: #5 | 3.2.4).

From these findings, the following changes are proposed to address the issues and improve the usability of the tool for the next iteration.

**Proposed Changes:**

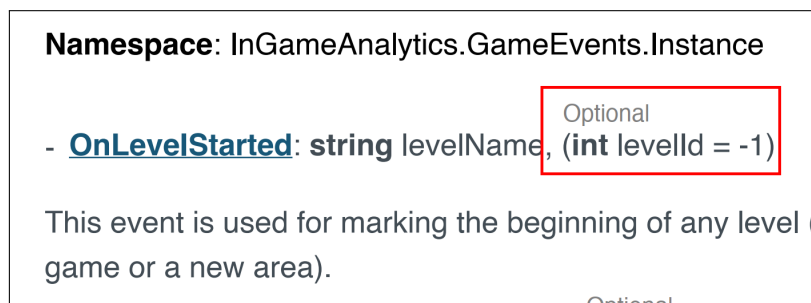
1. Add support for key-pairs when sending multiple Custom Events.
2. **README:** Better explain that certain arguments are optional.
3. **README:** Clearly explain that the events need to be placed in the code by the game developer.
4. While users expressed a desire for event parameters alongside event names in the UI window (4), addressing this comprehensively is complex. Unlike static event names, event parameter values are dynamic, making a one-size-fits-all solution difficult, and as such, it has been deemed not viable to be implemented for this prototype.

### 3.2.4.3 Third Iteration

The work for this iteration focused on improving the clarity of the README file and improving the usability of Custom Events, as highlighted out by the previous round of testing, in preparation for the study on Usability Evaluation of the prototype (Section 3.3). The following aspects have been updated based on feedback and proposed changes from the previous iteration:

**Custom Events Data:** The proposed change regarding key-pairs for custom events 1 has been implemented, introducing a new Dictionary-based approach alongside the existing ordered array implementation, which enables game developers to choose between the two options based on their preference. The Dictionary implementation allows for the use of key-value associations, providing more flexibility and context compared to the ordered array structure.

**README File:** The README file A.8.3 got further updated to visually inform that certain parameters are optional 2 (Figure 3.4). The text has been edited to state that C# programming knowledge and editing code are required to be able to make full use of the Unity plug-in 3. These modifications should reduce misconceptions and present the target audience of this tool to non-technical readers.



**Figure 3.4:** Third Iteration - Updated README file showing optional parameters

### **3.3 Study #3 - Usability Evaluation of Prototype**

As the final step, now that the prototype has been developed, it is necessary to evaluate its usability from a game developer's point of view, as the goal is to have a tool which can be used and understood by game developers.

This study is specifically aimed at evaluating the prototype at its current state, after the various iterations, and answering "*What are usability characteristics of such a tool deemed important from a game developer's perspective?*" (Research Question #2 2).

#### **3.3.1 Methods**

The final study will be a set of summative evaluations conducted on the same Unity project with the goal to assess the overall usability of the tool from a game developer's perspective.

The evaluation process encompasses both the functionality of the plug-in and feedback concerning the analytics metrics integrated within the tool and it is carried out in the form of a semi-structured interview (Interview questions can be found in Appendix A.4).

Utilising a User Experience Questionnaire (UEQ) and an System Usability Survey (SUS), they serve as the foundation for collecting qualitative data from each participant. Given the limited number of participants (three) in this phase, statistical significance may not be attainable. However, leveraging qualitative data collection methods allows for a nuanced understanding of participants' perceptions and experiences, enabling meaningful insights into the tool's usability. This approach acknowledges the inherent constraints of a small sample size while maximising the value derived from participants' qualitative feedback to inform iterative improvements in usability.

#### **3.3.2 Materials**

For this study, the prototype developed in the previous study (Section 3.2) will be used as a base for the evaluations.

The UEQ (see in Appendix A.10) and the SUS (see in Appendix A.9) are used to elicit answers and comments from the participants, in order to better understand their feelings and opinions on the usability of the prototype.

### **3.3.3 Procedure**

The usability evaluation has been performed to gather qualitative data from three game developers of varying experience, nationalities, genders and ages, in order to evaluate the current state of the prototype. One of the participants was a native English speaker, while the other two spoke English as their second or third language. Each evaluation has been fully performed online via an audio/video call with the participant's screen shared to the researcher.

The participants were provided with the same set of tasks as the previous two iterations to complete, which allowed for a consistent evaluation process. Upon finishing the assigned tasks, they were asked to fill out a UEQ and an SUS. These tools, normally useful to gather quantitative data, were utilised instead as a way to gather qualitative data regarding the perceived usability and overall user experience of the prototype. The low amount of participants available for this round was not sufficiently high to be statistically significant [17]. For this reason, follow-up questions, based on their responses, were asked to the participants to gather more in-depth comments and opinions on the reason behind their evaluations.

For each evaluation the following structure has been used:

1. User Testing with Testing Tasks A.3
2. Independent filling of UEQ
3. Analysis of UEQ answers and follow-up questions
4. Assisted filling of SUS with follow-up questions regarding responses

### **3.3.4 Results**

#### **1. Usability Evaluations**

Even though not directly used, given the low countable number of participants, for completeness and information, the summary of the participants' SUS scores are presented below, as well as the average and standard deviation:

**SUS**

	User 7	User 8	User 9
<b>SUS</b>	73	84	82
<b>Average</b>	80		
<b>Std. dev.</b>	5.74		

**Table 3.4:** SUS Summary Scores

## UEQ

The responses to the UEQ for each participant are available in the appendix A.11, as well as in the following subsection.

## 2. User Feedback

In this study, questionnaires have been utilised to gather structured feedback, paired with further clarifying questions concerning their choices has been conducted (Responses to both questionnaires are available in Appendix A.11). Below, a subset of comments and quotes provided by each participant has been included for reference and discussion:

### • User 7

1. Once you do one or two it's easy.
2. It might need to be more accessible to non-developers.
3. The README was just a long list of terms.
4. After reading the README file it was very straightforward, so I had no difficulties.
5. It's useful, it has a very clear purpose. I need to have analytics for my games, so why not?
6. Very straightforward.
7. Simple instructions, simple procedure of how to use the tool.
8. Everything is visually laid out for you and I liked how you can see the events in real time in the window.
9. Everything follows the same structure for the events.
10. I enjoyed that I got feedback when I added the functions was cool.
11. It was a smooth experience.

### • User 8

12. (...) I see where this tool is trying to add value.



13. Install it and then go.
14. From the logs, I am not sure if the data will arrive or not.
15. It helps motivating by clearing the mind for other stuff.
16. Good thought put into the UI to make it clear.
17. I would definitely see myself using this in projects that need it. I would need to take a look at the data storage stage.
18. I think it's very straightforward, as soon as I saw it's event-based I knew how it would do. It's setup in the same exact way as I would do it myself.
19. Especially with the README, it does everything. It gives you steps and it even gives you code examples. Maybe if you are not familiar with events but maybe a simple link to the Unity docs to explain that would fix it.
20. I found the UI window to be really clear and helpful to understand what's going on.
21. I think testing it out and immediately seeing in the console when it was working (or not), seeing this that it correctly tracks the logs makes it clear.
22. It frees up time to do the fun developer stuff.
23. I am wondering about more complex data. When I have to store game store, I might have to track more complex tracking.

• **User 9**

24. I expect to have events for game starting/ending and all was consistent. I felt I understood how it was working.
25. All of the functions had the same requirements for the parameters to pass and all usable in the same way. Nothing strange in the logic.
26. I don't know much about how it handling the data.
27. Makes me more efficient.
28. Everything was in clear places and easy to find everything.
29. No, after a first explanation. If the document was less of a list of events. More examples, more screenshots. Some use cases. Now it's very dry, basically.
30. I felt medium confident. I thought I had some ideas what I needed to do. I felt I needed to verify that it did the thing that I wanted to do.
31. It was nice to see that it worked pretty predictably.

In this round, users acknowledged the clear purpose and straightforward nature of the plug-in, contributing to an easy learning process. The visual layout, particularly the real-time event visualisation and the UI of the tool, was praised for its clarity and helpfulness. Participants appreciated the consistent structure of events and the immediate feedback provided, facilitating error prevention and a smooth experience.

On the other hand, areas for improvement were identified, mainly concerning the documentation, which could benefit from more examples, screenshots, and use cases to enhance accessibility and engagement, especially for unfamiliar concepts to less experienced developers. Concerns were raised regarding the tool's capability to handle complex data structures and the need for clarification on data storage and safety measures.

### 3. Discussion

The feedback from evaluations helped establishing the overall usability of the prototype at its current state, as well as understanding what usability metrics are important to game developers in regards to a reusable in-game analytics plug-in. From participants' comments, the following insights can be found:

- **Clear Purpose and Utility:** The tool's purpose and usefulness for game analytics are evident to users, making them likely to adopt it. The simple instructions and easy-to-follow procedure contribute to a seamless user experience that aligns well with expectations (Quotes: #5, #12, #15, #17, #22, #27 | 3.3.4).
- **Straightforward, Usable and Understandable:** Users perceive the tool as straightforward and simple to use, with clear instructions and procedures. In particular, some appreciated the clear visual layout of the UI window and the real-time feedback provided during event addition and log tracking, which contributed to a user-friendly experience (Quotes: #1, #4, #6, #7, #8, #9, #10, #11, #13, #16, #18, #19, #20, #21, #24, #25, #28, #31 | 3.3.4).
- **Room for Improvement in Documentation:** While the provided documentation was found to generally clear and comprehensive, a user suggested including more examples, screenshots, and use cases could make it more engaging and accessible to a wider audience, especially for users unfamiliar with certain concepts like events and analytics (Quotes: #2, #3, #29 | 3.3.4).
- **Uncertainty in Data Safety:** Users showed lack of confidence and understanding of what happens to the data after it is logged by the tool. There were also concerns on what would happen to the data if something went wrong (Quotes: #14, #23, #26, #30 | 3.3.4).

Further examining the feedback, the participants have shown to value the following usability metrics (based on Nielsen's Usability Heuristics [18]):

1. Efficiency
2. Perceived Usefulness
3. Learnability
4. Ease of Use
5. Satisfaction
6. Error Handling and Feedback
7. Help/Documentation
8. Data Safety

The SUS scores received (73, 84, and 82) indicate a generally positive usability assessment, suggesting that users found the plugin relatively easy to use and learn. The participants' feedback supports the results: User 7 described the plugin as "very straightforward", and having a "clear purpose". User 8 mentioned that "it's very straightforward" and "does everything," while User 9 found it "consistent" and working "predictably." The feedback from the UEQ suggests a positive user experience. Users appreciated the visual layout (User 7: "Everything is visually laid out for you"), the real-time feedback (User 7: "I enjoyed that I got feedback when I added the functions"), and the efficiency gains (User 9: "Makes me more efficient").

The evaluation process revealed that the prototype effectively addresses several critical usability metrics, including perceived usefulness, learnability, ease of use, and user satisfaction. By incorporating the insights gained from user feedback, particularly regarding clarity of the documentation as a whole, data safety concerns, and handling complex data scenarios, the tool can be refined to even better meet the expectations and requirements of game developers wanting to use such a plug-in in their own projects.

### **3.3.5 Discussion**

In this chapter, the studies performed during the research have been presented. Each study had the aim to answer a specific research questions (Research Questions in Section 1.3) and each achieved a different goal to help with the following study: Study #1 (in Section 3.1) helped validating the assumption that in-game metrics selected for different SGs in different domains (i.e. learning, health, and mental

and emotional health) are at compatible enough to be grouped into a generalised and reusable list of metrics (full list available in Section 3.1.4). This finding was somewhat surprising as one might expect such different domains to have very different data being collected, although the differences seemed to be more pronounced when compared across different genres of games rather than across different domains.

Study #2 (in Section 3.2) focused on the creation of the prototype of the in-game analytics collection tool and the feedback provided by game developers on its features and usability. The study highlighted the importance of working with the target users from the very beginning to better understand their needs and concerns, as even two iterations were able to pinpoint key usability issues. The inclusion of game developers with varying level of skills and domain knowledge allowed the feedback to cover many different aspects that a single profile of users might not have found focused on: beginner users focused on their initial experience, clear and understandable feedback, and the quality of the documentation provided, while experienced developers had internal consistency, data safety and efficiency in mind. These differences in focus allowed the feedback to be comprehensive and to explore many ways the tools could be used in real life. These findings are expected and support the current praxis in the field of Human Computer Interaction Design of involving users from the very beginning and making sure all profiles are involved in the process.

Study #3 (in Section 3.3) helped evaluating the prototype created in the previous study and also to uncover which usability characteristics are deemed important by a game developer when using an in-game analytics collection tool. The resulting list of characteristics (in Section 3.3.4) showed which usability heuristics are most important for game developers working on such a tool. While not surprising in and of itself, the characteristics showed that feeling at ease during the operation, understanding the usefulness of the tool and being able to understand its purpose are key aspects for developers. Beginner developers reinforced the previously noted idea that providing clear feedback and a comprehensive documentation is crucial for their experience and the outcome of their work.

# Discussion

## 4.1 Research Question #1:

### **How compatible are in-game metrics collected from the learning and physical health domain with metrics from Serious Games for emotional and mental health?**

The discussion of this question is based on Study #1, Game Metrics Validation (Section 3.1).

The interview with the senior data analyst from GEMH Lab provided valuable insights into the compatibility of in-game metrics across different domains like learning, physical health, and emotional/mental health SG's.

The data analyst highlighted that while basic metrics like game start/stop, level metrics, and scoring are essentially universal across games, the core gameplay metrics need to be closely aligned with the specific goals and objectives of each game. For example, in HnR where accuracy was paramount, they initially tracked reaction time but later prioritised accuracy metrics due to reliability issues. In contrast, for MindLights, aimed at reducing anxiety, they measured approach/avoidance behaviour through player actions mapping to the core gameplay loop. This suggests that the metrics from the learning and physical health domains may have some overlap with emotional/mental health games in terms of basic event tracking. Additionally, the data analyst found the proposed list of metrics 3.1.4 derived from the learning and physical health domains to be sufficiently generic to be applicable across a wide variety of projects from GEMH Lab, while still being specific enough to provide useful insights. This supports the claim that the metrics identified fully from the learning and physical health domains are indeed compatible and extensible to SG's focused on emotional and mental health interventions, helping to answer the

research question.

To note is that the "Custom event" metric was mainly included as a future-proofing measure to allow game developers to integrate any additional metrics not initially covered, rather than due to any specific lacking in the proposed metrics list. This, according to the data analyst, might provide flexibility to cater to evolving needs across different future game design goals.

Serrano et al. [14] stated in their paper that "One of the key open issues in learning analytics is the standardization of the data collected. This is a particularly challenging issue in serious games, which generate a diverse range of data", which supports the idea that standardisation across domain might be a difficult task. On the other hand, both Stănescu et al. [19] and Wiemeyer [20] in their papers propose standardised frameworks that can work across different domains, in order to create interoperability between different SGs. The findings of Study #1 in this report support the claim from Section 2.3 that there are shared commonalities between different domains of SG regarding in-game analytics, despite their inherent differences. Although at first sight different domains seem very different, the major differentiators are often the game genres chosen for such SGs rather than the goal of the games.

In conclusion, this study finds that there are shared commonalities between in-game analytics metrics collected in the physical and learning domains and the emotional and mental health ones, opening the road for analytics standardisation among these widespread domains of SGs.

#### **4.1.1 Limitations**

One key limitation of this study is the relatively small sample of SGs analysed - with source code access provided for only two games from a single studio. A broader analysis spanning multiple games and development studios focusing on emotional and mental health interventions could yield more generalised insights. Additionally, the study relied primarily on interviews with a single data analyst. While this provided an in-depth perspective, further interviews with game designers, developers and subject matter experts could uncover other important considerations when defining metrics for emotional/mental health SGs.

#### **4.1.2 Future Work**

More work could be done to evaluate even more projects in the emotional and mental health domains, as well as utilising the proposed set of metrics in real SGs to practically evaluate whether they are exhaustive enough for those domains.

## 4.2 Research Question #2:

### **What are usability characteristics of such a tool deemed important from a game developer's perspective?**

The discussion of this question is based on Study #3, Usability Evaluation of Prototype (Section 3.3).

The final evaluation round with game developers revealed several key usability characteristics that they deemed important for an in-game analytics system. Based on the feedback and insights gathered, developers valued the analytics tool as its purpose and usefulness was evident to them, making them more likely to adopt it. The simple instructions and easy-to-follow procedures contributed to a seamless user experience that aligned well with their expectations. Participants perceived the tool as straightforward, usable, and understandable, with clear instructions and procedures. The clear visual layout of the UI window and the real-time feedback provided during event addition and log tracking were also particularly appreciated, contributing to a more user-friendly experience. Developers also valued the tool's ease of learning, as demonstrated by their comments on the consistency of the tool, as well as simplicity and intuitive design, which contributed to an easy learning process.

Participants expressed satisfaction with the tool's functionality and user experience, indicating that it met their expectations and needs. While the provided documentation was generally clear and comprehensive, some less experienced participants suggested including more examples, screenshots, and use cases to enhance accessibility, especially for users unfamiliar with certain concepts like events and analytics. Concerns were also raised regarding the tool's capability to handle complex data structures and further clarifications on data storage and safety measures, showing a need for additional low-level system information to be included in the README file, as well as more clarity on data safety methods employed by the tool.

In their system literature review, Morales et al. [21] conducted a systematic literature review on the concept of Programmer eXperience (PX). The PX heuristics found in their study align with the heuristics deemed important by game developers in Study #2 of this report, as the papers they have reviewed include learnability, usefulness, clarity, ease of use and value. Furthermore, Morales et al. [22] in their following paper, adapting from Nielsen's usability heuristics [18], developed "a set of heuristics to evaluate programming environments, which includes aspects of usability/UX". Their work resulted in "a new set of 12 specific heuristics that incorporate concepts of UX and usability of programming environments," which highlight the im-

portance of feedback, consistency and help for programmers (full list of heuristics in Appendix A.12).

In conclusion, the usability characteristics deemed important by game developers found in this study are supported by the existing literature on general users as well as programmer-specific. While the results are not surprising, as game developers are still users and their tasks are generally not dissimilar from the ones of other programmers, they do add to the body of knowledge of usability from a game developer's perspective, especially regarding the usability of an in-game analytics tool for SGs.

### **4.2.1 Impact of Skill Level**

The feedback provided by the participants and their related skill level with game development and Unity can provide some insights on how that affects the usability characteristics deemed important for them. For instance, less experienced developers placed greater emphasis on the clarity of documentation, examples, and use cases, as well as the tool's ease of learning and visibility of the state of the system (through a cleared UI and rapid feedback). More experienced developers, on the other hand, prioritised efficiency, consistency with industry standards, advanced features for handling complex data structures, and robust data safety measures.

Kolling et al. [23] in their paper, focused on defining heuristics for novice programmers and developed a similar set of heuristics for developers, which included consistency, feedback and clarity in their list. As seen in this study, the skill level of the game developers seemed to have more influence in determining the relative importance of specific usability features, rather than in identifying which features were considered important overall.

### **4.2.2 Limitations**

As mentioned in the related study (Section 3.3), the limited number of participants in the final evaluation round is not deemed sufficient to achieve statistical significance. While this constraint was mitigated by leveraging qualitative data collection methods to gain meaningful insights into participants' perceptions and experiences, a larger sample size would be desirable to further validate and generalise the findings.

### **4.2.3 Future Work**

To address the limitations mentioned above, future work should focus on expanding the evaluation process to include a larger and more diverse sample of game devel-



operators; a larger sample size would provide more robust and generalisable insights into the usability characteristics. Additionally, increasing the age range of participants and making sure to still include developers with varying levels of experience would further refine the understanding of how skill level impacts the prioritisation of different usability metrics.

More work on providing support for data loss due to bad connectivity and offline usage of the prototype could be done to increase the overall robustness of the tool in many more contexts.

### **4.3 Main Research Question:**

## **How can a generic in-game analytics software tool for Serious Games for emotional and mental health be created?**

The discussion of this question is based on Study #2, Usability Evaluation of Prototype (Section 3.2).

The development of a generic in-game analytics software tool for SGs focused on emotional and mental health is a complex task that requires a thorough understanding of the needs and expectations of game developers, the first point of contact of the users of such a tool. Central to this development process is the continuous engagement with game developers, whose feedback is crucial in refining it.

In the prototype created for this study, the 3 iterations provided insights into how the developers interact with the tool, highlighting the importance of adaptability and responsiveness to user feedback when developing such a software tool. The feedback collected in the initial stages identified key aspects to keep in mind during development, such as the need for clear communication regarding actions not directly performed or setup by the user (e.g. automated events), the importance of providing documentation with examples, as well as effectively giving visual feedback to the user. Another critical aspect to bear in mind is the necessity for a simple, easy-to-install, and straightforward system; this reduces initial friction and allows developers to quickly start using the tool. This ease of use was consistently highlighted as a positive aspect by participants, emphasising that reducing the complexity of initial setup can crucially impact the tool's overall usability and acceptance. Consistency within the tool, particularly in how events are triggered, emerged as another important factor; developers rely on predictable patterns and behaviours to form mental models that guide their interactions with the tool. Inconsistencies might lead to errors and misunderstandings, thereby reducing the tool's effectiveness and the user experience.

These findings are not surprising as Kujala stated that "User involvement is a widely accepted principle in development of usable systems" [24], in fact the positive impact of user involvement on their satisfaction has been noticed by Baroudi in their paper [25] saying that "The results demonstrate that user involvement in the development of information systems will enhance both system usage and the user's satisfaction with the system". This is in line with the results of this study as user involvement helped improving the usability and satisfaction of the tool by other game developers.

In conclusion, someone creating a generic in-game analytics software tool must

first and foremost take into consideration its users and stakeholders from its inception, guiding them with clear feedback, documentation and consistency. Game developers and data analysts have different needs and goals, therefore such a tool much not only help accomplish the project goals but also bridge any existing gap between the two roles. A software tool developed this way can become an excellent tool not only for collecting in-game player data but also for creating a shared standardised language across the team, fostering a better understanding of the crucial step of in-game analytics collection for behaviour change.

### **4.3.1 Implications**

SGs offer a promising avenue for achieving behaviour change outside of a game through play. However, evaluating their actual effectiveness and impact can be challenging given the lack of established and generalised evaluation methodologies. In this regard, in-game analytics can play a crucial role in improving the evaluation of SGs and understanding their impact on players. As Fendandez et al. stated in their paper [26] that "the application of analytics is a complex and costly process that is not yet generalized in serious games", showing that a generalised tool for in-game analytics for SGs is currently missing and that such a tool could have an impact in decreasing the complexity and costs of such games.

A generalised tool would have impact on game developers who could become more and more acquainted with a single tool and method of in-game data collection for behaviour change, as well as data analysts who can start creating and using tools that do not change every project, allowing them to more effectively extract insights from the collected data.

This research contributes to the field of SGs by addressing a critical need: the creation of a reusable in-game analytics tool for SGs with the secondary goal of streamlining and standardising the crucial step of in-game data collection for behaviour change. A standardised and reusable tool for collecting in-game metrics has the possibility to bridge the gap between the roles, fostering a shared understanding and streamlining the process of in-game analytics. By ensuring the tool's usability and ease of integration from a developer's standpoint, the subsequent steps in the data analysis process can be facilitated more effectively, especially as a shared language can be created within the team around in-game analytics for behaviour change.

The approach used to defining and validating a comprehensive set of in-game metrics, spanning different domains such as learning, physical health, and emotional/mental health, contributes to the standardisation of metric collection across various SG projects. This standardisation can also facilitate cross-project compar-

isons and possibly enable more robust data analysis, which could lead to more effective interventions.

### **4.3.2 Limitations**

One of the limitations of the research was that this study focused mainly on the perspective of game developers, which, although intentional, may have resulted in a narrow view of the tool's usability. Including more professionals from other relevant roles, such as game designers and data analysts, could have yielded further insights into the tool's accessibility and the comprehensibility of the documentation surrounding the metrics. By involving a broader range of stakeholders, the research could have ensured that the documentation was understandable not only for developers but also for those responsible for analysing and interpreting the collected data, thereby enhancing the tool's overall usability and effectiveness in facilitating collaboration across different roles within the SGs development process.

### **4.3.3 Future Research**

Future work on the development and refinement of the in-game analytics software tool for SGs should focus on several key areas. First, expanding the user testing pool to include not only game developers but also game designers and data analysts is crucial. Additionally, applying the tool in at least a real-world emotional and mental health project aimed at behaviour change will certainly provide valuable insights into its practical application and impact. Such an implementation would allow for the assessment of the tool's efficacy in a live environment, offering concrete data on its ability to facilitate behaviour change through SGs. This practical application will help identify any further adjustments needed to optimise the tool's functionality and effectiveness. Furthermore, longitudinal studies measuring the impact of using this tool on both the game development process and the outcomes of emotional and mental health interventions would be beneficial, providing deeper insights into its long-term value and potential for widespread adoption in the SG field.

# Conclusions

This research aimed to develop a reusable in-game analytics software tool for SGs focused on emotional and mental health. The primary goal was to facilitate the collection of in-game metrics related to player behaviour, streamlining the process for game developers and data analysts involved in the development and evaluation of such games. Throughout the iterative development and evaluation process, comments and feedback were gathered from game developers. This feedback played a crucial role in refining the prototype and ensuring that it meets the usability characteristics deemed important from a game developer's perspective.

The work highlighted the importance of clear and comprehensive documentation in promoting the tool's usability and adoption. While the provided documentation was generally well-received, suggestions were made to include more examples, screenshots, and use cases to enhance accessibility, particularly for users less familiar with concepts like events and analytics. This research contributes to the field of SGs by addressing a critical need: improving communication and collaboration between game developers and data analysts. By providing a standardised and reusable tool for collecting in-game metrics, the study aims to bridge the gap between these two distinct roles, fostering a shared understanding and streamlining the process of in-game analytics.

The focus of the study on the game developer's perspective was intentional, as they are often the first point of contact in the data collection workflow. By ensuring the tool's usability and ease of integration from a developer's standpoint, the subsequent steps in the data analysis process can be facilitated more effectively. The approach used to defining and validating a comprehensive set of in-game metrics, spanning different domains such as learning, physical health, and emotional/mental health, contributes to the standardisation of metric collection across various SG projects. This standardisation can facilitate cross-project comparisons and enable more robust data analysis, ultimately possibly leading to more effective interventions.

The studies started with the assumption (in Section 2.3) that in-game metrics

selected for different SGs from learning and health domains were at least somewhat compatible with games from the emotional and mental health domain. The work and interview conducted in Study #1 (in Section 3.1) supported such a statement as the finalised list of in-game metrics that was used in the tool (list in Section 3.1.4) did not change when the two games from GEMH Lab were included and analysed ("Custom Event" was indeed added to the initial list but more as a precaution to future-proof the tool rather than due to a specific incompatibility).

This work marks the starting point for a truly holistic in-game analytics framework for SGs regardless of their domain: it allows for researchers, game developers, game designers and data analytics to be able to have a shared language around in-game data collection, to create tools which can be reused across projects and possibly shared within the field as a whole. Starting with GEMH Lab, game studios will not have to spend resources on how to collect data on player behaviour but rather on investing in the creation of reusable data processing and visualisation tools. This will have the consequence of not requiring data analysts to create single-use tools, macros or scripts for a specific project but to be able to create a robust arsenal of tools for their research. Future games will be impacted as, thanks to the standardisation of the in-game analytics metrics, from the very beginning a new SG will be able to leverage all the previously created tools to provide insights and thus allowing more time for the studios to focus on improving the game for its players.

# Appendices

## A.1 Data Analysts - Interview Questions

1. Normally, how are the in-game metrics to be used in a project decided?
2. What are the key goals GEMH Lab aims to achieve by collecting in-game metrics?
3. How is the data collected from the projects generally analysed?
4. Is there any workflow, tool or process for processing the collected data?
5. Are there any concerns or challenges related to the collection or analysis of in-game metrics that you have encountered in previous projects?
6. Have you experienced any issues in analyzing and interpreting the collected data in the past?
7. Have you encountered any issues related to data quality, such as duplicated data, inconsistent naming, etc.?
8. Are there any major workflow differences between projects when it comes to analysing and processing the collected data?
9. Are there any considerations to keep in mind when collecting in-game data?
10. Have you come across any instances where the data collected did not align with the intended metrics or objectives?
11. What are the most important aspects when analysing the collected data?
12. What would you change or improve to make the work of a data analyst better or easier?
13. What improvements can be made to the data analysis workflow or tools?
14. Are there any specific ethical considerations or guidelines followed when collecting and analyzing in-game metrics?

15. Has the collected data ever been used to predict the impact of the game on a player's behaviour outside the game? If so, when and how?
16. Are there any metrics that you think should be reconsidered or revised in terms of their relevance or effectiveness from a data analyst's perspective?
17. In your opinion, are there any metrics or categories that are missing from the list?
18. Do you have any comments or concerns with creating a generalized tool for collecting in-game metrics?
19. Do you have any further comments or final remarks?



## A.2 Data Analysts - Interview First Coding

1. In-game metrics crucial but often considered too late; need sophistication.
2. Hit-n-Run focused on accuracy, skipped reaction time for reliability issues.
3. Captured participant points, high scores, leaderboards, and basic metrics.
4. MindLights aimed at reducing anxiety; measured approach and avoidance behaviour.
5. Metrics (accuracy, reaction times, approach-avoid) based on literature and tasks.
6. No personalised data in-game; ethics prioritise protection outside the game.
7. Downloaded leaderboards, but not crucial for analysis; focused on Hangouts.
8. Captured textual data for analysis, decisions based on literature, research questions.
9. Exported accuracy/reaction time data to various tools; coupled with survey data.
10. Interested in change over time; observe reduced mistakes, faster reaction times.
11. In MindLights observed approach vs. avoid behaviour change over time.
12. Analysed Hangouts data semantics for team connections, focused on pronouns.
13. Connected, positive language users showed better results in quitting smoking.
14. Collecting and analysing chat data for valuable insights.
15. No fixed syntax; Excel macros for specific project data, not universal.
16. Standard thinking: Accuracy, reaction times, approach-avoidance in multiple projects.
17. Insufficient time spent on data structure; required manual formatting.
18. Unity dashboards used, but not always comprehensible; formatting challenges.
19. Conversion done manually; preference for control over using macros.

20. Issues with GameSparks data output; challenging participant identification and matching.
21. GameSparks data hard to understand starting/ending times; limited personalisation.
22. MindLight had custom-made data but backend issues.
23. Metrics vary among games; challenging standardisation in data collection.
24. Underlying psychological mechanisms similar; consider meta view for broader categorisation.
25. Data handling depends on expertise; computer scientists find game formatting easier.
26. Consider target group expertise; for social scientists, focus on R or SPSS.
27. Prefer participant-centric data structure; struggle with dispersed participant data.
28. Person-focused approach in social sciences; emphasise participant-centric presentation.
29. Social sciences focus on participants; game design on events.
30. Suggest making data more person-focused; combines all events for easier comprehension.
31. Consider target audience, social scientists may struggle with raw JSON.
32. Unexpected issue with unreliable reaction time data. Prioritised game experience.
33. Difficulty capturing accurate timestamps during gameplay.
34. Ethical considerations: GDPR, data protection; avoid personal data in global servers.
35. Collected personal data outside game; in-game, only anonymised, non-traceable metrics.
36. Hit-n-Run accuracy linked to better smoking outcomes outside the game.
37. MindLight behavioural data linked to anxiety levels outside the game.

38. Evaluation process: Pre-test, post-test, and follow-ups with questionnaires and tasks.
39. Connect in-game data to long-term outcomes; link accuracy improvement to smoking status.
40. Limited budgets; games designed for shorter intervention periods.
41. Assessing long-term impact, transfer to daily life is open-ended.
42. Statistical analysis provides insights; complexity and potential impact of unaccounted variables.
43. In-game behaviour correlates well with real-life behaviour; less influenced by social pressure.
44. Personalise gameplay for targeted improvement.
45. Generic metrics apply broadly; consider sub-metrics for activity performed.
46. Generic metrics align with varied game experiences.
47. System for data analysis or also guiding backend and game metric design?
48. Consider dual purpose: data analysis tool and guide for game design. Include sub-metrics and examples for better understanding across disciplines.
49. Add explanation of each metric for game designers.
50. More detailed explanation for broader utilisation in researchers' and designers' toolbox
51. Need for user-friendly explanations to bridge understanding between researchers and game designers.
52. Importance of guidance; researchers may lack clarity on backend possibilities.

### **A.3 Data Analysts - Interview Generating Themes**

1. Importance of In-Game Metrics:
  - Recognition of the crucial role of in-game metrics.
  - Emphasis on considering metrics early in the game development process.
  - Need for sophistication in handling in-game metrics.

## 2. Focus on Game-Specific Metrics:

- Hit-n-Run focused on accuracy metrics; skipped reaction time due to reliability issues.
- MindLights aimed at reducing anxiety; measured approach and avoidance behaviour.
- Metrics (accuracy, reaction times, approach-avoidance) based on literature and tasks.
- Standard thinking in terms of accuracy, reaction times, approach-avoidance.

## 3. Data Collection and Ethical Considerations:

- Captured participant points, high scores, leaderboards, and basic metrics.
- Ethical considerations prioritise protecting personal data outside the game.
- Downloaded leaderboards; focus on Hangouts data for analysis.
- Capture and analysis of textual data for valuable insights.

## 4. Challenges with Data Handling:

- Exported accuracy/reaction time data to various tools.
- Preference for a participant-centric data structure.
- Issues with GameSparks data output; challenging participant identification.
- Unity dashboards used, with challenges in comprehensibility and formatting.
- Insufficient time spent on data structure; required manual formatting.
- Unity dashboards used, but not always comprehensible; formatting challenges.
- Conversion done manually; preference for control over using macros.

## 5. Behavioural Analysis Goal:

- Interest in change over time; observing reduced mistakes, faster reaction times.
- In MindLights, observed approach vs. avoid behaviour change over time.
- Connected, positive language users showed better results in quitting smoking.
- Underlying psychological mechanisms considered similar; suggest a meta view for categorisation.

- In-game behaviour correlates well with real-life behaviour.

#### 6. Impact Evaluation Approaches:

- Evaluation process: Pre-test, post-test, and follow-ups with questionnaires and tasks.
- Limited budgets; games designed for shorter intervention periods.
- Assessing long-term impact; challenges in transferring game learning to daily life.
- Statistical analysis provides insights; acknowledgement of the complexity of unaccounted variables.

#### 7. Toolbox for Researchers and Designers:

- Consideration of the dual purpose of the system: data analysis tool and guide for game design.
- Need for user-friendly explanations to bridge understanding between researchers and game designers.
- Importance of guidance; researchers may lack clarity on backend possibilities.

# Game Developers Testing Tasks

## Tasks Definition

### Overview

This document contains the list of tasks to be performed for the evaluation of the **In-Game Analytics Plugin for Unity**.

For this experiment, *thinking aloud* will help with understanding how the tool is being experienced. To think aloud is simply to voice anything that comes to mind when performing the tasks.

### Tasks

#### Preparation:

- 1: Import tutorial project in Unity
- 2: Play the game a few times (WASD to move and "P" to skip levels)
- 3: Import plug-in from [.unitypackage](#)

#### Test:

- 1: Read README
- 2: Install plug-in
- 3: Verify that plug-in is installed correctly
- 4: Add **Level Started** event with current level (e.g. "Level 1") in *GameManager.cs* - [\[verify in the console that it is triggered\]](#)
- 5: Add **Level Finished** event with current level (e.g. "Level 2") in *GameManager.cs* - [\[verify in the console that it is triggered\]](#)
- 6: Add **Score Adjusted** event every time the score is updated (e.g. "10")
- 7: Add **Activity Performed** event when using booster ("Booster Used")
- 8: Add **Custom** event when the game ends with current score ("High Score")
- 9: Add **Custom** event with total time played when the game ends ("Time Played")

## A.4 Game Developer - Feedback Interview

1. How many years of experience do you have with Unity?
2. Have you previously used any in-game analytics plug-ins in Unity?
3. If Yes, please briefly describe your experience with the previous plug-in(s) (e.g., name of the plug-in, likes, and dislikes)
4. The initial setup process for the analytics plug-in was straightforward. (1-5)
5. Learning how to use the analytics plug-in was easy. (1-5)
6. The provided documentation (README) was helpful in setting up and using the plug-in. (1-5)
7. Navigating the analytics system was intuitive. (1-5)
8. Understanding the analytics system was intuitive. (1-5)
9. I encountered no major issues or bugs during the testing session. (1-5)
10. What features or functionalities did you find particularly useful in the analytics plug-in?
11. Were there any aspects of the plug-in that you found confusing or difficult to use? If yes, please describe.
12. Are there any specific analytics metrics or data points you feel are missing and should be added in future versions of the plug-in?
13. How do you envision the analytics plug-in improving your game development process further?
14. Do you have any other comments, suggestions, or feedback related to the Unity analytics plug-in?

## Round 1 - Plug-in README

# In-Game Analytics Plugin for Unity

## Overview

Welcome to the **In-Game Analytics Plugin for Unity**! This plugin provides an easy-to-use solution for integrating analytics into your Unity game to track user behaviour and gather valuable insights.

With this plugin, you can monitor various game events, player interactions, and more to improve your game.

## Installation

To get started with the In-Game Analytics Plugin, follow these steps:

### 1. Import the Package into Your Project:

- Open your Unity project.
- Double-click `.unitypackage` file.
- Click "Import" to add the plugin to your project.

### 2. Setup Analytics:

- Locate "Tools" from the top-bar
- Click on it and hover on "In-Game Analytics"
- Click "Show Helper" to open the In-Game Analytics Helper
- In the first scene of the project, click "Setup in Current Scene"
- Let the tool create the needed "In-Game Analytics Manager" gameobject
- Click "Check Installation" and make sure all tests pass ✓



# Round 1 - Plug-in README

## Default Events

- **OnLevelStarted**: **string** levelName, (**int** levelId = -1)
- **OnLevelFinished**: **string** levelName, (**int** levelId = -1)
- **OnScoreAdjusted**: **float** newScore
- **OnActivityPerformed**: **string** activityName, (**string** activityData = "")
- **OnAlternativeSelected**: **string** alternativeName, **string** chosenAlternative, **string[]** possibleAlternatives
- **OnUserDataUpdated**: **string** userDataName, **string** newValue
- **OnNPCInteraction**: **string** npcName, **string** interactionType, (**string** interactionData = "")
- **OnInterPlayerInteraction**: **string** fromPlayer, **string** toPlayer, **string** interactionType, (**string** interactionData = "")

## How to Send Default Events

### Example 1:

Sending event for new score achieved:

```
score = score + 100;  
InGameAnalytics.GameEvents.Instance.OnScoreAdjusted(score);
```

### Example 2:

Sending event for NPC interaction:

```
InGameAnalytics.GameEvents.Instance.OnNPCInteraction("Merchant",  
"Steal", "Apple (1)");
```

# Round 1 - Plug-in README

## Custom Events

- [OnCustomEvent](#): string key, string value
- [OnCustomEvents](#): string[] key, string[] value

## How to Send Custom Events

### Example 1:

Sending single custom event:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvent("Key", "Value");
```

### Example 2:

Sending multiple custom events at once:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvents(new string[]  
{ "Key1", "Key2", "Key3" }, new string[] { "Value1", "Value2", "Value3" });
```

## A.5 Round 1 - Game Developer Test Results

Results									
#	User 1			User 2			User 3		
	Errors	Time	Verify	Errors	Time	Verify	Errors	Time	Verify
3	0	00:05	00:00	0	00:15	00:00	0	00:36	00:00
4	3	04:55	01:00	0	01:47	00:20	1	01:03	03:30
5	0	01:21	00:31	0	01:32	00:27	0	01:13	00:32
6	0	01:04	00:30	0	01:30	00:28	0	01:04	00:07
7	0	00:51	00:20	0	00:44	00:17	0	00:47	00:11
8	1	04:52	01:10	0	02:47	00:45	1	02:45	00:32
9	0	02:20	00:26	0	01:19	00:22	0	00:50	00:16

**Table 1:** Round 1 - User Results [Time format mm:ss]

## A.6 Round 1 - Game Developer Feedback Answers

### A.6.1 User 1

#	Answer
1	1-2 years
2	No
3	N/A
4	Straightforward - 5
5	Learnability - 4
6	Documentation - 4
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	<ul style="list-style-type: none"><li>- Easy to use.</li><li>- Debugging is good.</li><li>- Setting up was easy.</li></ul>
11	Order of the console messages should reflect reality; Confusing.
12	Nothing to say.
13	I would use it a lot for debugging. I made something similar before for a project. I would use it for playtesting games and getting data on it.
14	All is good!

**Table 2:** User 1 - Feedback Responses

## A.6.2 User 2

#	Answer
1	Less than 1 year
2	No
3	N/A
4	Straightforward - 5
5	Learnability - 5
6	Documentation - 5
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	<ul style="list-style-type: none"><li>– Default functions are generic to every game, don't need to think about it. Just check what's available.</li><li>– On custom event is cool.</li><li>– Feels like a complete product.</li><li>– I would like to know how much memory it uses. I worry if there were millions of events it could have an impact.</li></ul>
11	No. All was easy to use.
12	<ul style="list-style-type: none"><li>– Level, game, score... they are all already there.</li><li>– Thinking to self: Larger data structures could be stored as a JSON...</li><li>– I would like to see the performance impact of the plugin.</li></ul>
13	<ul style="list-style-type: none"><li>– It puts my mind at ease, no need to think about naming.</li><li>– Generic solution to any logging I need. That's what I would use to do debug logs but then I would have the final analytics solution already in the game.</li></ul>
14	I said everything, maybe ways to automatically handle complex objects.

**Table 3:** User 2 - Feedback Responses

### A.6.3 User 3

#	Answer
1	3-5 years
2	No
3	N/A
4	Straightforward - 4
5	Learnability - 5
6	Documentation - 5
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	<ul style="list-style-type: none"><li>– I like that it was so easy to install and the feedback that it was installed correctly.</li><li>– I liked that there were some default events that would be used a lot.</li></ul>
11	Aside from what I said before, I would say nothing else was confusing.
12	<ul style="list-style-type: none"><li>– OnGameStarted and OnGameFinished events would be nice to have.</li><li>– I would like to send more complicated events, not just strings. Like an array or a complex object.</li></ul>
13	It would make it much easier to have game analytics setup from the get-go, so that would be nice.
14	<ul style="list-style-type: none"><li>– I would like to know what scene is being selected or which scene the plugin in being loaded.</li><li>– If it was possible, which events were already available (in that scene)</li></ul>

**Table 4:** User 3 - Feedback Responses

## Round 2 - Plug-in README

# In-Game Analytics Plugin for Unity

## Overview

Welcome to the **In-Game Analytics Plugin for Unity**! This plugin provides an easy-to-use solution for integrating analytics into your Unity game to track user behaviour and gather valuable insights.

With this plugin, you can monitor various game events, player interactions, and more to improve your game.

## Installation

To get started with the In-Game Analytics Plugin, follow these steps:

### 1. Import the Package into Your Project:

- Open your Unity project.
- Double-click `.unitypackage` file.
- Click "Import" to add the plugin to your project.

### 2. Install Analytics:

- Locate "Tools" from the top-bar
- Click on it and hover on "In-Game Analytics"
- Click "Show Helper" to open the In-Game Analytics Helper
- In the first scene of the project, click "Setup in Current Scene"
- Let the tool create the needed "In-Game Analytics Manager" gameobject
- Click "Check Installation" and make sure all tests pass ✓

## Round 2 - Plug-in README

# Overview Analytics Events

List of all events available in the plug-in and their description for reference.

**Namespace:** InGameAnalytics.GameEvents.Instance

Name	Type	Description
<b>OnGameStarted</b>	Automatic	Triggered when a game is initiated.
<b>OnGameStopped</b>	Automatic	Activated when the game application is halted.
<b>OnSceneChanged</b>	Automatic	Fired when the game transitions from one scene to another.
<b>OnLevelStarted</b>	Default	Used for the logging the beginning of a new level.
<b>OnLevelFinished</b>	Default	Used for the end of a game level.
<b>OnScoreAdjusted</b>	Default	When the player's score/currency change.
<b>OnActivityPerformed</b>	Default	When a specific in-game important activity/task is performed.
<b>OnAlternativeSelected</b>	Default	When the player chooses an alternative path or option.
<b>OnUserDataUpdated</b>	Default	When player-specific data is modified or updated, such as player profile, user settings, etc.
<b>OnNPCInteraction</b>	Default	Used for interactions between the player and a non-player character (NPC).
<b>OnInterPlayerInteraction</b>	Default	When there is an interaction between multiple players in a multiplayer environment.
<b>OnCustomEvent</b>	Custom	A generic event that can be customized based on specific game requirements.
<b>OnCustomEvents</b>	Custom	Same as OnCustomEvents but allows to send multiple events in one single request (reduces network calls).



# Round 2 - Plug-in README

## Automated Events

Automated Events are events that are handled by the plug-in automatically for you. They are activated by default, they can be disabled if needed.

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnGameStarted**: void

This event logs the start of a gaming session, the moment when the game is first launched.

- **OnGameStopped**: void

This event logs the end of a gaming session, the moment when the game is stopped completely and quits.

- **OnSceneChanged**: **Scene** current, **Scene** next

This event logs the change of a Unity scene from one to another. Useful to keep track of which locations the player is going to.

---

- **SetAutomatedEventsEnabled**: bool status

Allows to enable/disable the automated collection of game events.

# Round 2 - Plug-in README

## Default Events

Default Events are commonly used events which can be invoked anytime they are needed during the game. *(Optional parameters are shown in brackets.)*

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnLevelStarted**: **string** levelName, (**int** levelId = -1)

This event is used for marking the beginning of any level (e.g. new stage, new phase of the game or a new area).

- **OnLevelFinished**: **string** levelName, (**int** levelId = -1)

This event marks the conclusion of a previously started level.

- **OnScoreAdjusted**: **float** newScore

This event is used for any sort of numerical score that is changed by player's actions (e.g. collecting coins, spending energy or setting a new best time in a race).

- **OnActivityPerformed**: **string** activityName, (**string** activityData = "")

This is a general event which allows to mark any event that is performed by the player. Examples could be saving the game, opening a chest, unlocking a secret area. The event can be extended to provide any additional data for even more customization.

- **OnAlternativeSelected**: **string** alternativeName, **string** chosenAlternative, **string[]** possibleAlternatives

This event is used to mark any choice the player has had to make. Useful to log not only the choice but also the not chosen alternatives. Examples include a selected choice in a visual novel dialogue, path choices inside a dungeon in an RPG or doing an action over another.

- **OnUserDataUpdated**: **string** userDataName, **string** newValue

Event for logging updated data related to the user. This data can be about the players themselves or about their character(s). This could be used for logging the update of a user's nickname or changing the appearance of their main character, etc.

## Round 2 - Plug-in README

- **OnNPCInteraction**: **string** npcName, **string** interactionType, (**string** interactionData = "")

Event to log interaction with NPCs. It can be used for both Player-NPC interactions, as well as NPC-NPC interactions if needed. The additional data allows for further customization.

- **OnInterPlayerInteraction**: **string** fromPlayer, **string** toPlayer, **string** interactionType, (**string** interactionData = "")

Similar to *OnNPCInteraction* event but solely for Player-Player interactions.

## How to Send Default Events

### Example 1:

Sending event for new score achieved:

```
score = score + 100;
InGameAnalytics.GameEvents.Instance.OnScoreAdjusted(score);
```

### Example 2:

Sending event for NPC interaction:

```
InGameAnalytics.GameEvents.Instance.OnNPCInteraction("Merchant",
"Steal", "Apple (1)");
```

### Example 3:

Sending event whenever a choice is made:

```
InGameAnalytics.GameEvents.Instance.OnAlternativeSelected("L1_Companion_Choice_1", "Yes", new string[] {"Yes", "No", "Let me think"});
```

## Round 2 - Plug-in README

### Custom Events

Custom Events allows you to save any type of data. Useful for project-specific events not handled by the Default events or sending complex data for later analysis.

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnCustomEvent**: **string** key, **string** value

Flexible and customizable event which can be implemented for any project-specific logging needs. **Tip:** Complex objects can be encoded to a JSON string before sending.

- **OnCustomEvents**: **string[]** key, **string[]** value

Same as **OnCustomEvent** but allows to send multiple values in the same request. This reduces the total number of network requests needed to store multiple data. Useful when the data is complex and should be clumped into one request.

### How to Send Custom Events

**Example 1:**

Sending single custom event:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvent("Key", "Value");
```

**Example 2:**

Sending multiple custom events at once:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvents(new string[]  
{ "Key1", "Key2", "Key3" }, new string[] { "Value1", "Value2", "Value3" });
```

## A.7 Round 2 - Game Developer Test Results

Results									
#	User 4			User 5			User 6		
	Errors	Time	Verify	Errors	Time	Verify	Errors	Time	Verify
3	0	00:25	00:02	0	00:18	00:01	0	00:20	00:05
4	0	01:22	00:30	0	01:05	00:25	0	00:45	00:16
5	0	00:25	00:20	0	00:47	00:20	0	00:38	00:10
6	0	00:39	00:14	0	00:32	00:16	0	00:55	00:08
7	0	00:43	00:08	0	00:21	00:15	0	00:41	00:05
8	0	00:26	00:25	0	00:33	00:17	0	00:29	00:06
9	0	00:18	00:22	0	00:21	00:18	0	01:45	00:16

**Table 5:** Round 2 - User Results [Time format mm:ss]

## A.8 Round 2 - Game Developer Feedback Answers

### A.8.1 User 4

#	Answer
1	1-2 years
2	No
3	N/A
4	Straightforward - 5
5	Learnability - 4
6	Documentation - 5
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	Custom vs Default events is very useful.
11	The only thing was to find were to place the functions precisely.
12	No, it looks pretty flexible. Activity performed could be anything, so it's ok.
13	<ul style="list-style-type: none"><li>- I would love to see which levels are played (since in my game the levels can be played in any order)</li><li>- Which actions did the player take?</li><li>- Which puzzles they solved or did they get stuck somewhere.</li></ul>
14	No, all was good.

**Table 6:** User 4 - Feedback Responses

## A.8.2 User 5

#	Answer
1	6+ years
2	Yes
3	I used Unity Analytics system for various projects in the past. I liked that it was plug and play but it was really slow to update and sometimes clunky to use. Another major issue was to become tied to the Unity ecosystem, making it hard to migrate if needed. Also it might get expensive with bigger projects.
4	Straightforward - 5
5	Learnability - 5
6	Documentation - 5
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	The editor window with the real-time updated events list was extremely useful, I don't think I've seen that before.
11	All was pretty easy and smooth.
12	I think all seems there. Perhaps I would want a bit more visuals on the dashboard, like a graph showing which events are fired the most or something like that.
13	<ul style="list-style-type: none"> <li>– The system seems quite neat and the code is nicely available to understand the inner workings of the system, no magical black box.</li> <li>– I like that it's actually not sending anything when in development mode, as it would be quite annoying to spam the database when working.</li> <li>– Custom events are nice to have, as one can decide what to use.</li> </ul>
14	I usually make something similar when I start a project, this would save me time for sure.

**Table 7:** User 5 - Feedback Responses

### A.8.3 User 6

#	Answer
1	3-5 years
2	Yes
3	<ul style="list-style-type: none"><li>– I used a custom-made package for a game we did in a previous company.</li><li>– It had an unnecessarily confusing setup (lots of different imports and settings to configure)</li><li>– Once setup it worked fine although it was still pretty rudimentary.</li><li>– I liked that it had a cool dashboard (on the backend) but it wasn't related to the plugin, I guess</li></ul>
4	Straightforward - 5
5	Learnability - 5
6	Documentation - 4
7	Navigating Intuitive - 5
8	Understanding Intuitive - 5
9	No Bugs - 5
10	I particularly liked that you could see the events used in each file.
11	Nothing comes to mind.
12	All seems pretty good to me.
13	It would allow me to make sure each scene has full analytics coverage, which I think it's important. So, I can get good feedback from my players.
14	All is good

**Table 8:** User 6 - Feedback Responses



## Round 3 - Plug-in README

# In-Game Analytics Plugin for Unity

## Overview

Welcome to the **In-Game Analytics Plugin for Unity**! This plugin provides an easy-to-use solution for integrating analytics into your Unity game to track user behaviour and gather valuable insights.

With this plugin, you can monitor various game events, player interactions, and more to improve your game.

[Basic programming experience in C# required for using the events in-game.](#)

## Installation

To get started with the In-Game Analytics Plugin, follow these steps:

### 1. Import the Package into Your Project:

- Open your Unity project.
- Double-click `.unitypackage`` file.
- Click "Import" to add the plugin to your project.

### 2. Install Analytics:

- Locate "Tools" from the top-bar
- Click on it and hover on "In-Game Analytics"
- Click "Show Helper" to open the In-Game Analytics Helper
- In the first scene of the project, click "Setup in Current Scene"
- Let the tool create the needed "In-Game Analytics Manager" gameobject
- Click "Check Installation" and make sure all tests pass ✓

## Round 3 - Plug-in README

### Overview Analytics Events

List of all events available in the plug-in and their description for reference.

**Note:** *Events need to be placed in the code, wherever the event should be triggered.*

**Namespace:** InGameAnalytics.GameEvents.Instance

Name	Type	Description
<b>OnGameStarted</b>	Automatic	Triggered when a game is initiated.
<b>OnGameStopped</b>	Automatic	Activated when the game application is halted.
<b>OnSceneChanged</b>	Automatic	Fired when the game transitions from one scene to another.
<b>OnLevelStarted</b>	Default	Used for the logging the beginning of a new level.
<b>OnLevelFinished</b>	Default	Used for the end of a game level.
<b>OnScoreAdjusted</b>	Default	When the player's score/currency change.
<b>OnActivityPerformed</b>	Default	When a specific in-game important activity/task is performed.
<b>OnAlternativeSelected</b>	Default	When the player chooses an alternative path or option.
<b>OnUserDataUpdated</b>	Default	When player-specific data is modified or updated, such as player profile, user settings, etc.
<b>OnNPCInteraction</b>	Default	Used for interactions between the player and a non-player character (NPC).
<b>OnInterPlayerInteraction</b>	Default	When there is an interaction between multiple players in a multiplayer environment.
<b>OnCustomEvent</b>	Custom	A generic event that can be customized based on specific game requirements.
<b>OnCustomEvents</b>	Custom	Same as OnCustomEvents but allows to send multiple events in one single request (reduces network calls).

## Round 3 - Plug-in README

### Automated Events

Automated Events are events that are handled by the plug-in automatically for you. They are activated by default, they can be disabled if needed.

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnGameStarted**: void

This event logs the start of a gaming session, the moment when the game is first launched.

- **OnGameStopped**: void

This event logs the end of a gaming session, the moment when the game is stopped completely and quits.

- **OnSceneChanged**: **Scene** current, **Scene** next

This event logs the change of a Unity scene from one to another. Useful to keep track of which locations the player is going to.

---

- **SetAutomatedEventsEnabled**: bool status

Allows to enable/disable the automated collection of game events.

# Round 3 - Plug-in README

## Default Events

Default Events are commonly used events which can be invoked anytime they are needed during the game. *(Optional parameters are shown in brackets.)*

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnLevelStarted**: **string** levelName, Optional (**int** levelId = -1)

This event is used for marking the beginning of any level (e.g. new stage, new phase of the game or a new area).

- **OnLevelFinished**: **string** levelName, Optional (**int** levelId = -1)

This event marks the conclusion of a previously started level.

- **OnScoreAdjusted**: **float** newScore

This event is used for any sort of numerical score that is changed by player's actions (e.g. collecting coins, spending energy or setting a new best time in a race).

- **OnActivityPerformed**: **string** activityName, Optional (**string** activityData = "")

This is a general event which allows to mark any event that is performed by the player. Examples could be saving the game, opening a chest, unlocking a secret area. The event can be extended to provide any additional data for even more customization.

- **OnAlternativeSelected**: **string** alternativeName, **string** chosenAlternative, **string[]** possibleAlternatives

This event is used to mark any choice the player has had to make. Useful to log not only the choice but also the not chosen alternatives. Examples include a selected choice in a visual novel dialogue, path choices inside a dungeon in an RPG or doing an action over another.

- **OnUserDataUpdated**: **string** userDataName, **string** newValue

Event for logging updated data related to the user. This data can be about the players themselves or about their character(s). This could be used for logging the update of a user's nickname or changing the appearance of their main character, etc.

## Round 3 - Plug-in README

- **OnNPCInteraction**: `string` npcName, `string` interactionType, Optional (`string` interactionData = "")

Event to log interaction with NPCs. It can be used for both Player-NPC interactions, as well as NPC-NPC interactions if needed. The additional data allows for further customization.

- **OnInterPlayerInteraction**: `string` fromPlayer, `string` toPlayer, `string` interactionType,

Optional  
(`string` interactionData = "")

Similar to *OnNPCInteraction* event but solely for Player-Player interactions.

## How to Send Default Events

### Example 1:

Sending event for new score achieved:

```
score = score + 100;
InGameAnalytics.GameEvents.Instance.OnScoreAdjusted(score);
```

### Example 2:

Sending event for NPC interaction:

```
InGameAnalytics.GameEvents.Instance.OnNPCInteraction("Merchant",
"Steal", "Apple (1)");
```

### Example 3:

Sending event whenever a choice is made:

```
InGameAnalytics.GameEvents.Instance.OnAlternativeSelected("L1_Companion_Choice_1", "Yes", new string[] {"Yes", "No", "Let me think"});
```

## Round 3 - Plug-in README

### Custom Events

Custom Events allows you to save any type of data. Useful for project-specific events not handled by the Default events or sending complex data for later analysis.

---

**Namespace:** InGameAnalytics.GameEvents.Instance

- **OnCustomEvent**: **string** key, **string** value

Flexible and customizable event which can be implemented for any project-specific logging needs. **Tip:** Complex objects can be encoded to a JSON string before sending.

- **OnCustomEvents**: **string[]** key, **string[]** value

- **OnCustomEvents**: **Dictionary<string, string>** events

Same as **OnCustomEvent** but allows to send multiple values in the same request. This reduces the total number of network requests needed to store multiple data. Useful when the data is complex and should be clumped into one request. Can be used with separate ordered arrays or a dictionary, where the order of the events does not matter.

### How to Send Custom Events

**Example 1:**

Sending single custom event:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvent("Key", "Value");
```

**Example 2:**

Sending multiple custom events at once:

```
InGameAnalytics.GameEvents.Instance.OnCustomEvents(new string[]  
{ "Key1", "Key2", "Key3" }, new string[] { "Value1", "Value2", "Value3" });
```

## A.9 Round 3 - User Experience Questionnaire

Please assess the product now by ticking one circle per line.

	1	2	3	4	5	6	7		
annoying	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	enjoyable	1
not understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	understandable	2
creative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	dull	3
easy to learn	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	difficult to learn	4
valuable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	inferior	5
boring	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	exciting	6
not interesting	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	interesting	7
unpredictable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	predictable	8
fast	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	slow	9
inventive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	conventional	10
obstructive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	supportive	11
good	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	bad	12
complicated	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	easy	13
unlikable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasing	14
usual	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	leading edge	15
unpleasant	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	pleasant	16
secure	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	not secure	17
motivating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	demotivating	18
meets expectations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	does not meet expectations	19
inefficient	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	efficient	20
clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	confusing	21
impractical	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	practical	22
organized	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	cluttered	23
attractive	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unattractive	24
friendly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	unfriendly	25
conservative	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	innovative	26

## **A.10 Round 3 - System Usability Survey**

1. I think that I would like to use this tool frequently
2. I found the tool unnecessarily complex
3. I thought the tool was easy to use
4. I think that I would need the support of a technical person to be able to use this tool
5. I found the various functions in this tool were well integrated
6. I thought there was too much inconsistency in this tool
7. I would imagine that most people would learn to use this tool very quickly
8. I found the tool very cumbersome to use
9. I felt very confident using the tool
10. I found this tool boring
11. I had fun while using this tool
12. I did not find the tool interactive enough
13. I would play this tool again
14. I did not enjoy the tool



## A.11 Round 3 - Game Developer Feedback Answers

### A.11.1 User 7

<b>Experience</b>
1 to 2 years

#### UEQ

annoying	4	enjoyable	It felt like it did an analytics tool.
not understandable	4	understand.	Maybe just the README.
creative	4	dull	-
easy to learn	2	diff. to learn	Once you do one or two it's easy.
valuable	3	inferior	-
boring	3	exciting	-
not interesting	5	interesting	If the output gives something that a researcher can use.
unpredictable	4	predictable	It's consistent.
fast	4	slow	-
inventive	4	conventional	-
obstructive	4	supportive	-
good	3	bad	-
complicated	4	easy	-
unlikable	4	pleasing	-
usual	4	bleeding edge	-
unpleasant	3	pleasant	-
secure	2	not secure	I imagine it is more secure than sending data in public.
motivating	4	demotivating	-
meets expectations	3	d.n.m. expect.	-
inefficient	4	efficient	-
clear	4	confusing	-
impractical	4	practical	It might need to be more accessible to non-developers.
organized	5	cluttered	The README was just a long list of terms.
attractive	4	unattractive	-
friendly	3	unfriendly	After you get more insights into it. Then I understood it.
conservative	5	innovative	It depends on what it does with this data.

**Table 9:** User 7 - UEQ Responses

## SUS

#	Sentence	Value	Comment
1	I think that I would like to use this tool frequently	4	-
2	I found the tool unnecessarily complex	3	It was just very practical to use. Not necessarily easier than other tools though. Just felt normal, not harder, not easier.
3	I thought the tool was easy to use	4	Yes, eventually.
4	I think that I would need the support of a technical person to be able to use this tool	1	No, after a first explanation. If the document was less of a list of events. More examples, more screenshots. Some use cases. Now it's very dry, basically.
5	I found the various functions in this tool to be well integrated	4	-
6	I thought there was too much inconsistency in this tool	2	No, not necessarily.
7	I would imagine that most people would learn to use this tool very quickly	5	Developers definitely yeah. The data analysts would need a developer to integrate.
8	I found the tool to be very cumbersome to use	1	-
9	I felt very confident using the tool	3	I felt medium confident. I thought I had some ideas what I needed to do. I felt I needed to verify that it did the thing that I wanted to do.
10	I found this tool boring	3	I thought it was practical, it's not more boring than other similar tool.
11	I had fun while using this tool	4	It was nice to see that it worked pretty predictably.
12	I did not find the tool interactive enough	3	It's code. I don't expect it to be more interactive than this.
13	I would use this tool again	4	I would need to know what it actually outputs. Whether it is more useful.
14	I did not enjoy the tool	2	-

**Table 10:** User 7 - SUS Responses

## A.11.2 User 8

<b>Experience</b>
10+ years

### UEQ

annoying	5	enjoyable	-
not understandable	7	understandable	All made sense in the README.
creative	3	dull	-
easy to learn	1	difficult to learn	-
valuable	1	inferior	Standardised way to store the data has multiple benefits: 1) Stops to have to reinvent the wheel. 2) The researcher can expect how the data will look like. 3) Only one solution so the plugin can be tied into a larger system (e.g. dashboard). I see where this tool is trying to add value.
boring	5	exciting	-
not interesting	6	interesting	-
unpredictable	2	predictable	It's what I expected.
fast	2	slow	Install it and then go.
inventive	2	conventional	-
obstructive	6	supportive	-
good	2	bad	Just some QoL stuff but other than that it.
complicated	6	easy	-
unlikable	5	pleasing	-
usual	5	bleeding edge	-
unpleasant	6	pleasant	-
secure	4	not secure	From the logs, I am not sure if the data will arrive or not.
motivating	3	demotivating	It helps motivating by clearing the mind for other stuff.
meets expectations	2	d.n.m. expect.	-
inefficient	6	efficient	-
clear	2	confusing	Good thought put into the UI to make it clear.
impractical	7	practical	-
organized	2	cluttered	-
attractive	2	unattractive	-
friendly	2	unfriendly	-
conservative	5	innovative	-

**Table 11:** User 8 - UEQ Responses

**SUS**

#	Sentence	Value	Comment
1	I think that I would like to use this tool frequently	4	I would definitely see myself using this in projects that need it. I would need to take a look at the data storage stage.
2	I found the tool unnecessarily complex	1	I think it's very straightforward, as soon as I saw it's event-based I knew how it would do. It's setup in the same exact way as I would do it myself.
3	I thought the tool was easy to use	4	-
4	I think that I would need the support of a technical person to be able to use this tool	1	Especially with the README, it does everything. It gives you steps and it even gives you code examples. Maybe if you are not familiar with events but maybe a simple link to the Unity docs to explain that would fix it.
5	I found the various func. in this tool to be well integrated	5	I found the UI window to be really clear and helpful to understand what's going on.
6	I thought there was too much inconsistency in this tool	1	The way the custom events are shown, I would expect to tell me it's a custom event, otherwise it's consistent.
7	I would imagine that most people would learn to use this tool very quickly	5	I think it would take 1 look at the docs. As soon as you call that line once, it's a matter of picking the right event.
8	I found the tool to be very cumbersome to use	1	Straightforward.
9	I felt very confident using the tool	5	I think testing it out and immediately seeing in the console when it was working (or not), seeing this that it correctly tracks the logs makes it clear.
10	I found this tool boring	3	It's data analytics, it's not super exciting but it does the job.
11	I had fun while using this tool	3	It frees up time to do the fun developer stuff.
12	I did not find the tool interactive enough	2	There are some things that I'd like to see but overall it does the job. Leaving the custom data very open is definitely very helpful to customize for any scenario.
13	I would use this tool again	4	I am wondering about more complex data. When I have to store game store, I might have to track more complex tracking.
14	I did not enjoy the tool	2	-

**Table 12:** User 8 - SUS Responses

### A.11.3 User 9

<b>Experience</b>
5 years

#### UEQ

annoying	4	enjoyable	I guess it's as enjoyable as a tool can be. And not annoying as it does the job.
not understandable	7	understandable	I expect to have events for game starting/ending and all was consistent. I felt I understood how it was working.
creative	3	dull	-
easy to learn	1	difficult to learn	Same as understandable.
valuable	2	inferior	-
boring	4	exciting	-
not interesting	5	interesting	-
unpredictable	7	predictable	All of the functions had the same requirements for the parameters to pass and all usable in the same way. Nothing strange in the logic.
fast	4	slow	It's as fast as you use it, it doesn't have an inherent speed.
inventive	4	conventional	Very specifically solves a problem, it meets all of the requirements but it's not groundbreaking.
obstructive	6	supportive	I can imagine it would support me with analytics.
good	2	bad	Well made, easy to use. And it's good.
complicated	7	easy	-
unlikable	6	pleasing	Taking away a task that I would have to do otherwise, and that's pleasing.
usual	4	bleeding edge	Same as before, it's neither new nor tired.
unpleasant	5	pleasant	-
secure	4	not secure	I don't know much about how it handling the data.
motivating	3	demotivating	Makes it easier, it's a bit motivating.
meets expectations	1	d.n.m. expect.	-
inefficient	6	efficient	Makes me more efficient.
clear	1	confusing	-
impractical	7	practical	-
organized	1	cluttered	Everything was in clear places and easy to find everything.
attractive	3	unattractive	-
friendly	4	unfriendly	It's a utility, it would be weird if it was friendly.
conservative	5	innovative	-

**Table 13:** User 9 - UEQ Responses

## SUS

#	Sentence	Value	Comment
1	I think that I would like to use this tool frequently	4	It's useful, it has a very clear purpose. I need to have analytics for my games, so why not?
2	I found the tool unnecessarily complex	2	Very straightforward.
3	I thought the tool was easy to use	5	Simple instructions, simple procedure of how to use the tool.
4	I think that I would need the support of a technical person to be able to use this tool	1	No, it's a developer tool, so you need to be a somewhat technical person to use this tool.
5	I found the various functions in this tool to be well integrated	5	Everything is visually laid out for you and I liked how you can see the events in real time in the window.
6	I thought there was too much inconsistency in this tool	1	Everything follows the same structure for the events.
7	I would imagine that most people would learn to use this tool very quickly	4	It's simpler than most plug-ins.
8	I found the tool to be very cumbersome to use	1	It wasn't cumbersome.
9	I felt very confident using the tool	5	After reading the README file it was very straightforward, so I had no difficulties.
10	I found this tool boring	3	It's just a tool, it doesn't bring me joy or boredom, it's just useful.
11	I had fun while using this tool	3	Same as before.
12	I did not find the tool interactive enough	2	I enjoyed that I got feedback when I added the functions was cool.
13	I would use this tool again	4	When making another game, I would.
14	I did not enjoy the tool	2	It was a smooth experience.

**Table 14:** User 9 - SUS Responses

## A.12 Programmer's Experience Heuristics

ID	Heuristic	Description
PE-1	Visibility	The programming environment must keep the user informed about its status. In addition, it must deliver timely information
PE-2	Programmer Language	The programming environment must speak the programmer language without ambiguous terminology
PE-3	Control	The programming environments must give control to the programmer on his/her project, in a safe way
PE-4	Consistency	The programming environments must be consistent in appearance and behaviour
PE-5	Error-Prevention	The programming environments should favour the prevention of errors over their reporting. If the system can prevent or workaround an error, then it should do it
PE-6	Recognition	Programming environments should minimise the memory load of the programmer, favouring recognition
PE-7	Flexibility of Use	The programming environments must allow different ways of use
PE-8	Minimalist Design	The programming environments must show only important information orderly and clear
PE-9	Error Handling and Recovery	The programming environments should offer clear error messages and facilities to recovery
PE-10	Help	The programming environments must provide aid for programmers in relation to the way they should be used
PE-11	Configurable Interface	The programming environments must allow the programmer to customise its interface
PE-12	Automatic feedback	The programming environment should offer the programmer automatic feedback that facilitates his/her work

**Table 15:** Programmer's Experience Heuristics [22]





# Bibliography

- [1] T. M. Fleming, L. Bavin, K. Stasiak, E. Hermansson-Webb, S. N. Merry, C. Cheek, M. Lucassen, H. M. Lau, B. Pollmuller, and S. Hetrick, "Serious games and gamification for mental health: Current status and promising directions," *Frontiers in Psychiatry*, vol. 7, p. 215, Jan 2017.
- [2] A. Dewhirst, R. Laugharne, and R. Shankar, "Therapeutic use of serious games in mental health: scoping review," *BJPsych Open*, vol. 8, no. 2, p. e37, 2022.
- [3] C. Alonso-Fernandez, I. Martinez-Ortiz, R. Caballero, M. Freire, and B. Fernández-Manjón, "Predicting students' knowledge after playing a serious game based on learning analytics data: A case study," *Journal of Computer Assisted Learning*, vol. 36, 06 2020.
- [4] A. Wols, A. Lichtwarck-Aschoff, E. A. Schoneveld, and I. Granic, "In-game play behaviours during an applied video game for anxiety prevention predict successful intervention outcomes," *Journal of Psychopathology and Behavioral Assessment*, vol. 40, pp. 655–668, 12 2018.
- [5] A. Vaccari, "In-game metrics for evaluating the impact of a serious game for behaviour change," *Research Topics*, 2023.
- [6] A. A. Cariaga and R. Feria, "Learning analytics through a digital game-based learning environment," in *2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2015, pp. 1–3.
- [7] D. Ifenthaler and C. Widanapathirana, "Development and validation of a learning analytics framework: Two case studies using support vector machines," *Technology, Knowledge and Learning*, vol. 19, no. 1, pp. 221–240, 2014. [Online]. Available: <https://doi.org/10.1007/s10758-014-9226-4>
- [8] W. Westera, R. Nadolski, and H. Hummel, "Serious gaming analytics: What students' log files tell us about gaming and learning," *International Journal of Serious Games*, vol. 1, no. 2, Jun. 2014.

- [9] F. Chen, Y. Cui, and M.-W. Chu, "Utilizing game analytics to inform and validate digital game-based assessment with evidence-centered game design: A case study," *International Journal of Artificial Intelligence in Education*, vol. 30, 2020.
- [10] N. Westera, Wim and Hans, "Learning analytics in serious gaming," *Games and Learning Alliance*, vol. 8605, 2014, publication status: Published - 26 Oct 2014.
- [11] A. Alamri, M. M. Hassan, M. A. Hossain, M. Al-Qurishi, Y. Aldukhayyil, and M. S. Hossain, "Evaluating the impact of a cloud-based serious game on obese people," *Comput. Hum. Behav.*, vol. 30, p. 468–475, jan 2014.
- [12] J. Henriksson, "Validating game design with game analytics in a location-based game," Master's Thesis, University of Turku, Department of Future Technologies, Turku, Finland, Dec. 2018, 42 p., 7 app. p.
- [13] I. Daoudi, "Learning analytics for enhancing the usability of serious games in formal education: A systematic literature review and research agenda," *Educational Technology & Society*, vol. 27, no. 8, pp. 11 237–11 266, 2022, epub 2022 May 3.
- [14] Ángel Serrano-Laguna, I. Martínez-Ortiz, J. Haag, D. Regan, A. Johnson, and B. Fernández-Manjón, "Applying standards to systematize learning analytics in serious games," *Computer Standards & Interfaces*, vol. 50, pp. 116–123, 2017.
- [15] A. Tuijnman, I. Granic, J. Whitkin, and R. C. Engels, "Work in progress: Developing and testing scrollquest: A video game targeting rejection sensitivity in adolescents," in *CHI PLAY '17 Extended Abstracts: Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play*, 2017, pp. 213–221.
- [16] H. Scholten and I. Granic, "Use of the principles of design thinking to address limitations of digital mental health interventions for youth: Viewpoint," *Journal of Medical Internet Research*, vol. 21, no. 1, p. e11528, 2019. [Online]. Available: <https://www.jmir.org/2019/1/e11528>
- [17] J. Lewis and J. Sauro, "Sample sizes for a sus score," <https://measuringu.com/sample-sizes-for-sus-ci/>, 1 2022.
- [18] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '94. New York, NY, USA: Association for Computing Machinery, 1994, p. 152–158. [Online]. Available: <https://doi.org/10.1145/191666.191729>

- [19] I. A. Stănescu, A. Stefan, M. Kravcik, T. Lim, and R. Bidarra, "Interoperability strategies for serious games development," *Internet Learning*, vol. 2, no. 1, 3 2013, spring 2013.
- [20] J. Wiemeyer, "Towards a generic framework for serious games," in *Proceedings of the Conference*. Darmstadt, Germany: Technische Universität Darmstadt, n.d., institute for Sport Science.
- [21] J. Morales, C. Rusu, F. Botella, and D. Quiñones, "Programmer experience: A systematic literature review," *IEEE Access*, vol. 7, pp. 71 740–71 752, 6 2019, received April 14, 2019; accepted May 14, 2019; date of publication May 31, 2019; date of current version June 11, 2019.
- [22] —, "Programmer experience: A set of heuristics for programming environments," in *Proceedings of the Conference*. Talca, Chile: Universidad Autónoma de Chile, 2023.
- [23] M. Kölling and F. McKay, "Heuristic evaluation for novice programming systems," *ACM Transactions on Computing Education*, vol. 16, no. 3, pp. 1–30, 6 2016.
- [24] S. Kujala, "User involvement: A review of the benefits and challenges," *Behaviour & Information Technology*, vol. 22, no. 1, pp. 1–16, 2003.
- [25] J. J. Baroudi, M. H. Olson, and B. Ives, "An empirical study of the impact of user involvement on system usage and information satisfaction," *Commun. ACM*, vol. 29, no. 3, p. 232–238, mar 1986. [Online]. Available: <https://doi.org/10.1145/5666.5669>
- [26] C. Alonso-Fernández, A. Calvo-Morata, M. Freire, I. Martínez-Ortiz, and B. Fernández-Manjón, "Game learning analytics: Blending visual and data mining techniques to improve serious games and to better understand player learning," *Journal of Learning Analytics*, vol. 9, no. 3, pp. 32–49, 2022. [Online]. Available: <https://doi.org/10.18608/jla.2022.7633>