Telemetry for Arcade-Like Games: Model and Tool

Geraldo Xexéo*†, Eduardo Mangeli*, Allan David† and Miguel Péres†

*LUDES - Laboratório de Ludologia, Engenharia e Simulação
Programa de Engenharia de Sistemas e Computação - COPPE
Universidade Federal do Rio de Janeiro
email: [xexeo,mangeli]@cos.ufrj.br

†Departamento de Ciência da Computação
Instituto de Matemática
Universidade Federal do Rio de Janeiro
email: [allandavid,miguelperes]@ufrj.br

***Abstract*—As the game development gets more approachable, more individuals try to breakthrough in the independent scene, which is why launching a game with quality is essential to standout in the market. However, without the budget to invest in complex tools or large teams, it's not always simple to achieve a high level quality. This paper propose a model and an open source Unity extensions tool to aid developers in gathering and visualizing gameplay data to analyze player behavior, and with this information assert their design decisions, making the necessary tweaking to improve player experience.**

***Keywords*-game; development; telemetry; quality;**

I. Introduction

With low financial resources, independent game developers, known as *indies*, are betting on innovation to stand out in the saturated gaming market, even competing with *AAA* titles. However, in the quality aspect, unlike big companies that have dedicated testing teams and budget to invest in complex analytic tools, *indies* need to resort to other ways of ensuring the quality of their products. For example, the "early access" strategy, when costumers can buy an unfinished alpha version of the game, is one method to expose the product to players and obtain feedback faster and at the same time get paid during the development period. This way the developers can fund their products while ensuring its quality for the target audience.

This feedback can be acquired unobtrusively, by analyzing player behavior in-game and comparing it with the design intentions behind the game level. Kohwalter et al. [1] proposed a framework for provenance in games, inspired by the Open Provenance Model [2], to analyze the outcome of serious games.

This work defines a lean model for mapping game events into specific concepts of provenance. This model is used to implement an open source Unity extension that gathers data, through telemetry, and generate a visualization simple to interpret.

This visualization will aid independent developers in detecting and solving design flaws, to help them launch a quality product to stand out in the market and please their audience, without the need of a high budget or big QA teams.

The text is structured as follows: first it will delve into the required background knowledge [II] to clarify the approach taken, then it will propose a model [III] and briefly describe the implementation [IV], to finally show the application in an actual game [V] and state the conclusion [VI].

II. Background Knowledge

A. Quality in Video Games

There are academic and technical works about game quality. Those works often use different terms to refers to the same concept, but there are efforts to systematize quality concepts related to games. One can see an example in the systematic mapping [3] that relates quality concepts from the ISO 25010 [4] standard to those presented in works on quality of serious games.

Fullerton [5], for example, defined five factors that should be taken into account to ensure quality for a game. So, a game should be: internally complete, functional, balanced, acessible, and fun. Balance, accessibility and fun are related to Quality in Use. One should also remember that Quality in Use depends "not only on the software or computer system, but also on the particular context in which the product is being used" [6]. That rationale, including the context to understand quality, is present in other works in this area.

In this sense, Brunnström et al. set a series of concepts related to the quality of experience (QoE) [7].

Important to note that, in this paper, we assume that games *per se* are systems, as stated in [8] and [9]. So, we are interested in the quality of games and not only in the quality of software. Hence, gather information about a game execution in its full context is crucial to improve the analysis of its quality.

B. Telemetry

Telemetry is a process to gather measures and data remotely.

From companies with thousands of employees [10] to solo developers [11], game developers have been using telemetry to help the development of their designs. There are different telemetry tools in the market to help them, which offer many options, mostly paid or with premium features, such as: Telemetry[1] (from RAD Game Tools), Unity Analytics[2], and Unity Heatmaps[3].

C. Provenance

The term provenance refers to a kind of metadata that captures information about how a product was produced in a workflow context [12]. Therefore, data provenance explains how the data object of analysis was obtained, allowing verification and auditing as well as the re-acquisition or re-computation of the data if some error occurs in a previous step [13].

The Open Provenance Model [2] propose a causality graph that is composed of three types of nodes: *Artifacts*, *Processes* and *Agents* [1]. Artifacts perform or cause Processes, creating new artifacts. An Agents is a "contextual entity acting as a catalyst to the processes, enabling, facilitating, controlling or affecting their execution" [2]. The model uses a graph represention, and its notation can be seen in Figure 1.

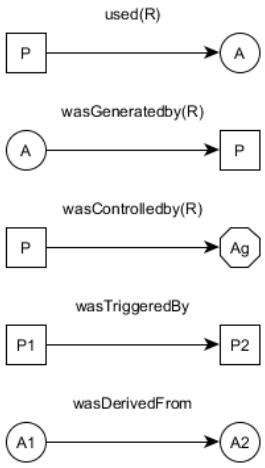


Figure 1. The OPM Model nodes and relationships [2]

D. Provenance in Games

Kohwalter et al. [1] defined a framework able to register data during game sessions. The intent was to register the behavior of the players while playing a serious game, to later allow for the discussion of how their decisions and actions affected the final result, what were their mistakes and what they got right, to help the learning process. With this goal, they adapted the Open Provenance Model to the video game context.

Later on, Kohwalter et al., following their previous work, also developed a framework, PinG [14], to track actions and events with focus in cause-and-effect relationship, being able to plot the events in a static top-down image, generated from the a game level map.

III. Model

The objective of the proposed model is to be able to map every situation of interest that happens during the play-through of a game session into one of some specified events. In this model, events are observable facts that happen involving one or more entities.

A. Entities

There are the two types of entities: Agents and Objects.

Agents are entities provided with intelligence, whether it is a player or NPC, i.e. any entity that can make decisions, or have a will on its own.

Objects are tools that take no decisions, which are manipulated by the agents. Some example of entities that can be mapped as objects in this model are: a sword in a adventure game, or a horse in an exploration game.

B. Events

Every observable situation that happens in the game will be mapped into one of the following category of events: Single Event, Chain Event and Tracking Event. Each one of them has its particular characteristics and usage.

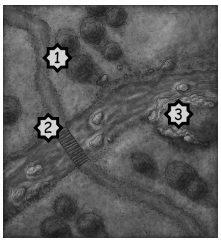
1) Single Event: Single Events represent instant changes in the state of the game made by an agent. Each interaction of an agent with objects that yields an immediate consequence, is a candidate for a Single Event. Figure 2a shows a view of a hypothetical game's map to depict the usage of single events. Each number represents a Single Event. Event 1, in the figure, could represent a character harvesting berries from a bush.

2) Chain Event: Chain Events also represent changes in the state of the game, but depicting two different events happening at different times: one depicting the cause and one the consequence of the actions taken by an agent. Thus expressing how the agent action affect the course of the game. Figure 2b shows a view of some hypothetical game's map to depict the usage of chain events, like a hunter, from the one side of the river, shooting it's prey, on the other side, with arrows .

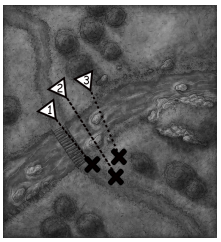
[1]http://www.radgametools.com/telemetry.htm

[2]https://docs.unity3d.com/Manual/UnityAnalytics.html

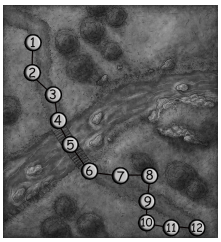
[3]https://assetstore.unity.com/packages/add-ons/services/analytics/unity-analytics-heatmaps-65801



(a) Example of Single Events to capture a variety of in-game events.



(b) Example of Chain Events to capture how a hunt came about.



(c) Example of Tracking Events to capture in-game player position.

Figure 2. Event Categories

3) Tracking Event: The purpose of the Tracking Event is to represent an event that has a small consequence in the final outcome of the game in the player's point of view. This type of event is used to keep track of an agent's position or an object of interest, e.g. Figure 2c shows an example where the avatar position is continuously registered while it moves, registering the chosen route.

IV. Implementation

The proposed model was implemented as a Unity tool.

The core of the project was to implement the event node. To be instantiated, an event needs a type (Single, Tracking or Chain), a name (to be identified in the inspector), a position, an optional linked event, and optional custom information of the developer's interest.

The developers must code the event triggers. These triggered events will be stored in a chosen database, to be later retrieved and rendered in the 3D interface of Unity.

V. Case Study: Heratoth

Heratoth is a stealth game where the player takes control of an archeologist exploring a tomb in search of an ancient relic. The player has to avoid the creatures that lie in the tomb, guarding the relic. Along the way, the player can find and read journals of other explorers that perished in the same quest, to learn the true story behind the relic. The game has two endings (a good and a bad one), depending on whether the player has found all the journals or not, before stumbling on the relic.

A. Events' Use in the Game

The Tracking Event was used to track player position, which was triggered each second, so the player's movement could be accurately analyzed. (Figure 3a)

Two in-game events were mapped into Single Events: the reading of a journal and the possession of the relic.

The reading of a journal is triggered when the player interacts with it, opening a UI that shows its contents. Since events can be customized with extra information, the time the player spent reading the journal was registered. (Figure 3b)

The other Single Event is registered when the player interacts with the relic, finally finishing the game. It also registers which of the two endings the player accomplished.

The event mapped as a Chain Event is when the player is spotted by one of the creatures in the tomb, losing the game. The event shows where the player and the create were at the moment of the sighting. With this event is possible to analyze the difficulty in the game, detecting parts of the level that players are struggling with.

B. Using Data to Improve

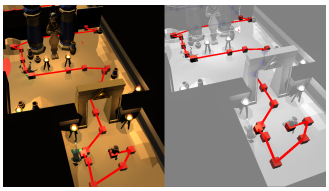
With these events mapped, and after collecting some data, it was possible to detect some points of improvement. Most of the players lost the game when going through one specific corridor, which was too long and had a creature patrolling it. The player could neither spot the enemy beforehand (because of the corridor length) nor outrun them (because of their speed), and there was no place to hide, so after entering the corridor the player was doomed. To solve this a room was added midway, so the player could get out of the enemy sight.

Another issue was that no one was finding all the journals, which means that no one reached the good ending of the game. The hypothesis was that no one was getting to the last journal because it was located beyond the relic, and every player went to the relic immediately after spotting it, without exploring the rest of the tomb. A possible solution to this problem was to show an indicator of the total amount of journals available to find and how many the player had already found, to incite the players to search for it.

VI. Conclusion

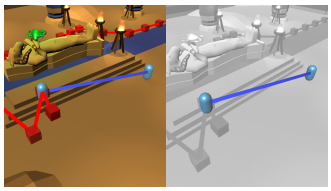
The telemetry process is very useful for developers to gather player's data and measure the quality in their products and adjust it to provide the best experience for their users. Some companies use complex and expensive tools as a solution, but for small companies or solo developers, a simple (and cheap/free) solution is enough to vastly improve their games.

In this work we have created a model and an open source and easy to use Unity extension tool, a game engine



(a) Visualization of the "Player Position" Tracking Event. The connected red cubes represent the player's path.

(b) Visualization of the "Journal Reading" Single Event. The green sphere represent a journal that has been read.



(c) Visualization of the "Player Spotted" Chain Event. The blue cylinders are the enemy and the player.

Figure 3. Heratoth visualization

widely used and known by independent developers. The tool generates an intuitive visualization in real-time in a Unity scene, allowing the user to accurately navigate even through complex 3D scenarios, analyzing the events registered during a play session to backtrack player's actions. We believe that this interactive three-dimensional visualization, available in real-time, is an improvement to previous works that, on the other hand, provided static two-dimensional visualization.

Triggering three kinds of predefined events, it is possible to map every situation in a play-through, regardless of any technical or aesthetic decision the developers may have taken for their game. The simplicity of the model make it easy to understand and even be extended. The only constraint being to respect the game format, which should be "arcade like": small independent maps that can be contained in one Unity Scene.

The tool can be found in this repository: https://github.com/allmonty/telemetry

Acknowledgments

This work was done with support from CNPq, National Council of Cientific and Technologic Development - Brasil - RESOLUÇÃO NORMATIVA RN-017/2006 and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] T. C. Kohwalter, E. W. G. Clua, and L. G. P. Murta, "Provenance in games," in *SBGames*, 2012.

[2] L. Moreau, B. C. amd Juliana Freire, J. Futrell, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. Bussche, "The open provenance model core specification (v1.1)," *Future Generation Computer Systems*, jun 2010.

[3] J. A. Vargas, L. García-Mundo, M. Genero, and M. Piattini, "A systematic mapping study on serious game quality," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2014, p. 15.

[4] ISO/IEC, "Iso/iec 25010 system and software quality models," ISO/IEC, Tech. Rep., 2010.

[5] T. Fullerton, *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*, 2nd ed. Morgan Kaufmann, 2008.

[6] ISO/IEC, "Systems and software engineering - systems and software quality requirements and evaluation (square) – measurement of quality in use," ISO/IEC, Tech. Rep. ISO/IEC 25022 ISO/IEC JTC 1/SC 7 N, 2012.

[7] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, "Qualinet white paper on definitions of quality of experience," 2013.

[8] K. Salen and E. Zimmerman, *The Game Design Reader: A Rules of Play Anthology*. The MIT Press, 2005.

[9] A. Järvinen, *Games without Frontiers: Theories and Methods for Game Studies and Design*. Tampere University Press, 2008.

[10] A. Drachen and A. Canossa, "Evaluating motion: spatial user behaviour in virtual environments," *International Journal of Arts and Technology*, 2013.

[11] C. Pruett, "Hot failure: Tuning gameplay with simple player metrics," *Game Developer Magazine*, sep 2010.

[12] Y. L. Simmhan, B. Plale, and D. Gannon, "A survey of data provenance in e-science," *SIGMOD Rec.*, vol. 34, no. 3, pp. 31–36, Sep. 2005.

[13] R. Ikeda and J. Widom, "Panda: A system for provenance and data," in *Proceedings of the 2Nd Conference on Theory and Practice of Provenance*, ser. TAPP'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 5–5.

[14] T. C. Kohwalter, L. G. P. Murta, and E. W. G. Clua, "Capturing game telemetry with provenance," *SBGames*, nov 2017.