Entertainment Computing 27 (2018) 110–127

Contents lists available at ScienceDirect

Entertainment Computing

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

Understanding game sessions through provenance

Troy Costa Kohwalter[a,*,1], Felipe Machado de Azeredo Figueira[b,2], Eduardo Assis de Lima Serdeiro[b,2], Jose Ricardo da Silva Junior[b,2], Leonardo Gresta Paulino Murta[a,1], Esteban Walter Gonzalez Clua[a,1]

[a] *Universidade Federal Fluminense, Brazil*
[b] *Instituto Federal do Rio de Janeiro, Brazil*

ARTICLE INFO

Keywords:
Game analytics
Tracked game data
Provenance graph

ABSTRACT

The outcome of a gameplay session is derived from a series of events, decisions, and interactions made during the game. Many techniques have been developed by the game industry to understand a gameplay session. A successful technique is game analytics, which aims at understanding behavior patterns to improve game quality. However, current methods are not sufficient to capture underlying cause-and-effect relationships that occur during a gameplay session, which would allow designers to better identify possible mistakes in the mechanics or fine-tune their game. Recently, it was proposed a conceptual framework based on provenance to capture these relationships. In this paper, we present a concrete framework to capture provenance data, allowing developers to add provenance gathering capabilities to their games. We instantiated our framework in two games, showing how it can be used in practice, and we developed a new game to demonstrate how provenance could be employed in early stages of game development to assist balancing the difficulty. We conducted an experiment with twelve volunteers and used the gathered provenance data to answer designers' frequent questions when trying to understand game sessions and balancing the difficulty of their games. This supports the relevance of collecting provenance data from games.

1. Introduction

The analysis of tracked game data, also known as game telemetry, has become an important stage of game design and production in the last few years [1]. This gathered data brings relevant information and possibilities, such as measuring the game stability [2], dynamically adjusting the difficulty of the game [3], performing behavioral analysis [4], understanding common behaviors [5], improving the monetization process [1], and balancing the game experience [6]. Moreover, game telemetry allows game developers to collect player interactions in the game inconspicuously over extended time periods, during production and after deployment.

Tracking game data and making it understandable is challenging due to the complexity of the games, leading to huge amounts of information. Additionally, deciding which information should be tracked and recorded is another challenge. One of the most common types of telemetry data is through states changes [7–9]. Even though state data is easier to examine, they typically lack contextual information and provides only a high-level view of what happened in the game. In contrast, telemetry data that captures events [10,11], can provide more low-level and fine-grained information, capturing and describing the player activity and relating it more closely to the game session. Furthermore, since the data is collected at fine-grain, developers can use aggregating techniques to summarize the data by giving an overview of the game sessions and only digging through the fine-grained data when necessary.

However, no known approaches for game analytics take into consideration the cause-and-effect relationships between events during a game session, which may be an important factor for determining the reasons that led to a certain outcome. In a recent work, Kohwalter et al. [12] introduced the usage of digital provenance [3] in games in order to detect these cause-and-effect relationships through a conceptual framework, named *Provenance in Games* (PinG), that can collect information during a game session and maps the data to provenance terms,

* Corresponding author.
E-mail addresses: tkohwalter@ic.uff.br (T. Costa Kohwalter), felchado@gmail.com (F.M. de Azeredo Figueira), eduardoassislima@gmail.com (E.A. de Lima Serdeiro), jose.junior@ifrj.edu.br (J.R. da Silva Junior), leomurta@ic.uff.br (L. Gresta Paulino Murta), esteban@ic.uff.br (E. Walter Gonzalez Clua).
[1] Address: Av. Gal. Milton Tavares de Souza, s/nº, Niterói, RJ, Brazil.
[2] Address: Av. Maria Luiza, s/nº, Eng. Paulo de Frontin, RJ, Brazil.
[3] Provenance refers to the documented history of an object's life cycle and is generally used in the context of art, digital data, and science [13].

https://doi.org/10.1016/j.entcom.2018.05.001
Received 1 March 2018; Received in revised form 26 April 2018; Accepted 6 May 2018

providing the means for a post-game analysis. That conceptual framework was manually instantiated over a game named SDM [14], which focuses on teaching Software Engineering concepts. The provenance support in that game allowed for a broader range of analysis by using collected provenance information to generate provenance graphs [15]. Even more recently, Kohwalter et al. [16] also demonstrated the benefits of using their PinG approach during game analysis of serious games, helping students to understand the underlying reasons for an outcome.

In this paper, we present a concrete framework for capturing provenance data for the game engine Unity, allowing developers to add provenance gathering capabilities to their games. We detail how our proposed framework can be instantiated in an existing game and show how the generated provenance graph can be visualized using the provenance visualization tool *Prov Viewer* [17], which is a visualization tool that supports multiple features for visual data analysis, including spatial-referencing the graph in the game level map.

We also provide an evaluation on how our framework can be used for the analysis of the cause-and-effect relationships by instantiating it over two open source games released by the Unity team. Additionally, we also demonstrate that our concrete framework can be used at early stages of game design and development by instantiating it over an inhouse game to aid the game balancing process. Instead of only relying on beta testers feedback, we collect and analyze the cause-and-effect relationships emerged during game sessions played by twelve invited subjects. The analysis is used to answer a set of questions game designers normally ask [18][4] (e.g., "How does the level of challenge increase as the player succeeds?").

This paper extends a prior conference paper [19] by demonstrating how provenance can be used in early stages of game development to aid the game design in the process of balancing a game. This analysis is mapped to questions game designs normally ask during the balancing process. The steps discussed here could be used with minor adjustments for other games.

This paper is organized as follows. The Section 2 presents related work and Section 3 provides background information as well as our proposed PinG framework. Section 4 presents the PinG framework usage and analysis over two existing games. Section 5 discuss the execution of the experiment, subjects' characteristics, and the design of the developed game. The results and discussion of the data game provenance analysis are also presented in the same section. Finally, Section 6 concludes this work, pointing out future works.

2. Related work

The literature adopts different terms for **tracked game data**, such as gameplay data, logged data, play traces, and telemetry data. Moreover, the process of analyzing such data, referenced here as **game analytics**, is also named in different ways, such as gameplay visualization, visual data mining, and game session analysis. In this section, we kept the original terms of each work, as they are usually reflected in the approaches' names.

Joslin [10] proposed the *Gameplay Visualization Manifesto* (GVM), which is a framework for gameplay data logging that uncovers gameplay events by attaching logging methods in the game objects responsible for generating relevant events during the game. The event model is the basis for the game data logging framework. It encapsulates the information that is desired by users and classifies the events into three groups: immersion, quest, and social. The immersion group represents events related to increasing the player's sensation of being involved in the game flux. The quest group represents events related to quest creation, execution, and analysis. Lastly, the social group represents events related to social factors in the game, such as group meeting or interaction with other characters.

The main application of GVM is for collecting game metrics, such as player deaths, position, time spent in available features (*e.g.*, crafting and fighting), item usage (*e.g.*, equipment), actions performed, and player enjoyment. Therefore, GVM does not track cause-and-effect relationships. It tracks only the executed actions along with their timestamp and location, in addition to character attributes and equipment.

Kim et al. [11] proposed the *Tracking Real-Time User Experience* (TRUE) approach that combines human–computer interaction (HCI) instrumentation, which collects *user initiated events* (UIEs), and log file analysis techniques in order to automatically record user interactions with games. Thus, TRUE can capture behavioral data and the attitudinal information behind the decisions made by the player in order to obtain better understanding of the context of each captured behavior.

Nevertheless, the designer still needs to infer the reasons behind the elements that led to an outcome. This occurs because the contextual information is only extra attributes that were tracked during the execution of the action and not actual relationships between events. Thus it does not capture cause-and-effect relationships. The cause-and-effect relationships must be inferred by the designer when analyzing the logged data. Moreover, TRUE was designed for the industry and is not easily available for indie companies. Even though we did not explore attitudinal data with PinG, it can be trivially incorporated in our approach as attributes for the player's actions or by creating specific activity vertices only for the attitudinal data when they are captured.

Playtracer [8], which is a visual tool designed to illustrate how groups of players move through the game space, aids the designer by tracking game states and showing common pathways and alternatives that players used to succeed or fail in their tasks, identifying pitfalls and anomalies in the scene. Nonetheless, *Playtracer* does not consider temporal information and does not preserve the order of the states visited by players when he/she revisits the same state. Moreover, incorporating *Playtracer* in the game design is challenging because it requires designers to define a state distance metric and identify relevant states.

Play-Graph [7] captures and illustrates the sequence of states and the actions that caused the player's state changes over the course of the game. In the Play-Graph context, a game state describes a certain configuration of the game or an entity, while actions consist on player interactions within the game, such as shooting, jumping, or using an object. In this concept, a game is viewed as a finite state machine with a finite number of states and transitions between them. The states are composed of a set of attributes from the game and players trigger actions at some specific points in the game. However, due to the nature of how the data is structured in Play-Graph, the understanding of player behavior is guided by the player progression in the game (*e.g.*, killed a boss), and not by how he/she interacted with the world (*e.g.*, combat rounds from the battle against the boss). From the available documentation, there is no way to determine interactions or influences. Only the changes from one state to another, caused by an action executed by the player, can be identified. Conversely, influences in the player's action, such as an influence from another character that affected the transition of one state to another, are not present in the graph (there are no edges linking edges).

According to Fernandez-Vara [20], different analyses need to be performed in order to increase the video game quality. Such analyses include understanding the game balancing in order to better attune for the vast majority of players. This process can be facilitated by using the collected game data. Furthermore, balancing also impacts how the player perceives the game difficulty. The analysis presented by Fernandez-Vara can be performed automatically to adapt the game difficulty to match the current player's skills. According to Black and Hickey [21], player's profile can change progressively or suffer immediate changes. The former is referred as evolutionary adaptation while the latter is referred as revolutionary adaptation. There are a plenty of approaches designed for performing dynamic difficulty adjustments

[4] In the book, Lens #31 (The Lens of Challenge), Schell defines a set of questions advised to be used while performing game balancing.