An Infrastructure for Gameplay Gathering and Analysis with Provenance

Troy C. Kohwalter, Leonardo G. P. Murta (*Co-Advisor*), Esteban W. G. Clua *(Advisor)*
Programa de Pós-Graduação em Computação
Universidade Federal Fluminense
Niterói-RJ, Brasil
{tkohwalter, leomurta, esteban}@ic.uff.br

***Abstract*— The outcome of a game session is the consequence of a series of events and decisions that are made by the player. Understanding and extracting the session information is important in Game Analytics. This thesis proposes a framework to capture data from game sessions and generate interactive provenance graphs for exploratory analysis, allowing analysts to understand the results obtained. We present an unprecedented way to represent and capture causal relationships in game sessions, resulting in four major contributions: (1) The *PinG for Unity* (*PinGU*) framework, for tracking provenance data, (2) *Prov Viewer*, for interactive visualization of provenance graphs, (3) three distinct summarization techniques, for reducing the overall provenance graph size with minimal information loss, and (4) a comparison technique for provenance graphs, to discover underlying issues that might have affected the result.**

Keywords- provenance; graph; games; metrics; telemetry; analytics, data mining

I. Introduction

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

The game industry has developed many proprietary tools and techniques to track and store data from a gaming session. This increasing demand for understanding player behavior resulted in the Game Analytics field, which aims at improving game quality and enhances the player experience. However, the current methods for analytics are not enough to capture the underlying cause-and-effect influences that shape the outcome of a game session and, therefore, allowing deeper understanding and interpretation of the game features.

During my Masters, we proposed a novel approach [7] based on provenance to track and record these causal relationships, providing the necessary groundwork to use provenance information in game analytics. The provenance data is a structured and directed acyclic graph, also known as a provenance graph. In the game context, the resulting provenance graph shows actions performed by characters (player or non-player) and events that occurred during game sessions, and the causal dependencies among these actions or events. The information collected during the game is used for the generation of the provenance graph.

This thesis extends our original approach by providing a concrete framework for tracking, managing, and visualizing provenance data during the game. Through this work, we can plot the provenance data in an interactive graph for exploratory analysis, allowing developers and analysts to understand the events and outcomes better.

Game telemetry has the general challenge of working with *Big Data* [1]. Thus, we also propose automatic summarization techniques to reduce the provenance data without losing information by clustering similar sequential events that alone were not enough to generate any meaningful change in the game. Furthermore, we take the provenance analysis to a new level, allowing the analysis of multiple provenance graphs simultaneously by generating a summarized provenance graph. This summarized graph is useful for game designers, to aid in the detection of patterns in player's behaviors, to identify issues not reported by game testers, to confirm hypotheses formulated by the development team, and even testing monetization issues.

II. PinG for Unity

In this thesis, we introduce a generic component capable of gathering provenance during a game session, leading to a domain-independent and low-coupling solution. Our solution, named PinGU[1](PinG for Unity), provides easier provenance extraction, requiring minimal coding in the game's existing components. PinGU has three different types of classes: eight *Core* classes, one *Interface* class, and five *Auxiliary* classes.

Figure 1 illustrates a simplified class diagram for PinGU. *Core* classes are in yellow, *Interface* class is in light blue, and *Auxiliary* classes are in orange. The *Core* classes represent an evolution of the infrastructure of PinG [7] and are responsible for provenance information management, making everything transparent to the game designer. Analogously, it can be referenced as the provenance

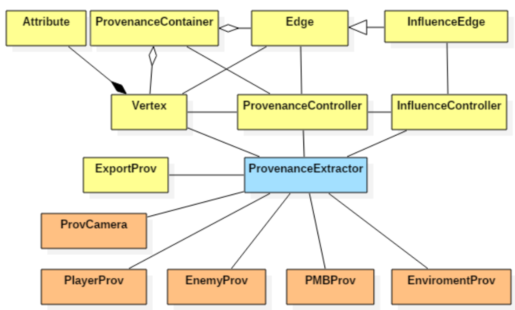

Figure 1: PinG for Unity class diagram

[1] PinGU is available at: https://github.com/gems-uff/ping

"server." PinGU is game independent but is currently only available for the game engine Unity.

PinGU has one class (*Provenance Controller*) responsible for storing all the provenance data. Furthermore, it uses two structures for storing provenance data: one list for storing actions, events, characters, and objects, which are the vertices of a provenance graph, and another list for relationships that appear as edges in a provenance graph. These two data structures (*Vertex* and *Edge*) store the provenance data in the format used by the graph. Meanwhile, the *Influence Controller* class manages the cause-and-effect relationships (influence edges), dealing with possible influences and passing it to the *Provenance Controller* class as they occur in the game. The *Provenance Container* class exports the provenance data from the game.

The *Interface* class (*Provenance Extractor)* is the gateway between the game and the *Core* classes. While the *Core* classes can be seen as the server, the *Interface* class can be seen as the client application and is responsible for tracking all provenance information and passing it to the *Provenance Controller* for storage.

Lastly, the *Auxiliary* classes contain domain-specific functions customized for a specific behavior or games and are responsible for gathering domain-data. These classes represent the provenance tracking functions that are used during provenance gathering and are required to be inserted in the existing game classes to capture telemetry data. Nevertheless, the existing templates can be used as a guiding example in cases that the desired action is not already implemented. These classes are *PBMProv*, *PlayerProv*, *EnemyProv*, and *EnviromentProv*, and each is customized for the genre of the game they represent (Car-related movements, Player, Enemy, and Environment). The Thesis and the SBGames 2017's Best Paper [8] describe in more details how to integrate PinGU in existing games.

A. *Provenance Gathering*

The PinGU approach is responsible for gathering all provenance data through its *Provenance Extractor* class and passing them to the *Provenance Controller* when the data is ready to be stored. Thus, instead of each character having a list of actions, now the character only needs to notify the *Provenance Extractor* class the action executed together with any additional desired information (*i.e.*, attributes), as illustrated by Figure 2.

After receiving the notification, the *Provenance Extractor* class packs the details of the execution, creates the corresponding vertex (Activity, Agent, or Entity), and sends it to *Provenance Controller*, which in turn stores the new vertex in the list for vertices. PinGU automatically generates default provenance relationships through the *Provenance Controller*, which are the relationships between activities and the agents that executed them, along with the chronological order of events.

B. *Tracking Influences*

Whenever an agent executes an action and notifies it to *Provenance Extractor*, the game should also notify if the action can generate any possible influences. If it does, then the *Provenance Extractor* passes the information to *Influence Controller*. After receiving the notification,

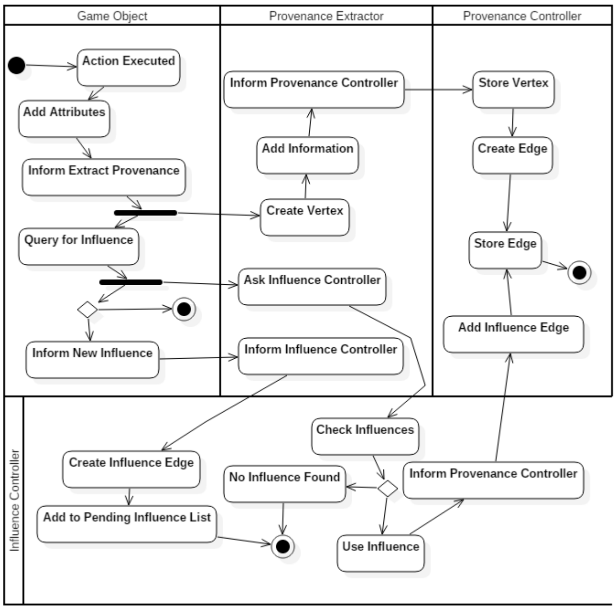

Figure 2: PinGU tracking and storing provenance data

Influence Controller creates a special *Influence Edge* and stores it in a pending influence list awaiting further instructions. Furthermore, *Influence Controller* registers the action that generated the influence and the triggers for the influence, along with other relevant information for the influence. Therefore, it is necessary to notify possible influences that the action generated during its execution. Figure 2 also illustrates how PinGU track influences.

The required information related to influences normally is present in the game design document (GDD). The GDD explicitly says the effects of each action and how they can interact in the game. In other words, if an action can affect other actions, such as the effect of a spell that will affect the player that used it.

III. Provenance Visualization

Displaying game data is also an issue in present times, bringing problems related to scalability when dealing with long game sessions or by having too many actors/players. Using provenance as a method of gathering game data escalates this issue due to the richness and highly detailed data, generating huge quantities of data consequently. Although there are some tools in the literature for graph analysis [9]–[11], they are based on simple node-link diagrams. However, using these simple node-link diagrams to represent provenance data can also harden the graph understanding when dealing with the wealth of information contained in a single provenance node, even when using the different shapes to distinguish the information.

Therefore, another contribution of this thesis is the *Prov Viewer*[2] tool, a graph-based visualization tool for interactive exploration of provenance data. *Prov Viewer* processes the collected provenance data to generate a provenance graph to provide advanced visualization features for identifying steps and contributors to a given result.

Prov Viewer is the result of several extensions and new techniques we developed to address issues encountered in

[2] *Prov Viewer* is available at https://github.com/gems-uff/prov-viewer

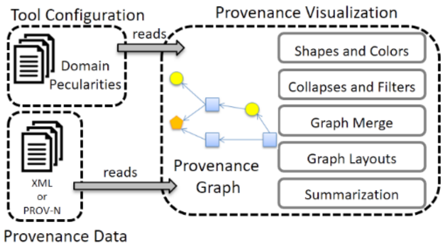

Figure 3: *Prov Viewer's* high-level architecture

different scenarios. We designed new visual representations and interaction mechanisms that address many of the aforementioned challenges: (1) domain configuration, customizing the visualization for specific needs; (2) interoperability, supporting PROV-N [12] for importing provenance data; (3) shapes, sizes, and colors, supporting a clear distinction of information types; (4) collapsing, highlighting the relevant information in the graph; (5) filtering, removing information that is not relevant for a given analysis; (6) graph merge, integrating the analysis of multiple trials; and (7) specialized layouts, organizing the graph in a more understandable way and using the game map as a background. Figure 3 shows the high-level architecture of our tool, illustrating some of its features that allow users to interact with the provenance data and identify relevant actions that impacted the results.

Figure 4 illustrates one of the possible visualizations of the provenance data gathered by PinGU in a game scene, showing the vertex visualization scheme for the player's health value (vertex color using a traffic light scheme) and the edges that influence in it (green and red edges) as the game progresses. Blue vertices represent other characters in the game (enemies), blue edges represent the chronological order of events, and green edges represent the player's health generation due to his passive regeneration ability. By analyzing Figure 4, we can see the chronology of events, regions visited by the player, sections where more action happened, places where the player engaged in battle, and when the player suffered heavy health loss. The thesis describes such analysis in more detail as well as provide other case studies. The Thesis and IPAW Paper [13] describe Prov Viewer in more details.

IV. Provenance Analysis

Although our visualization tool offers many features to interact with the provenance graph, there was still a problem when visualizing provenance graphs from long game sessions. Thus, we also proposed three automatic data filtering techniques [14] for provenance graphs, based on *DBSCAN* [15]. Our algorithms take into consideration the temporal sequence of information of a provenance graph to summarize tracked data to a more manageable size through the usage of collapse strategies. In the game domain, vertices that were not collapsed tend to represent drastic changes in the game state and might be worth displaying during the analysis, while all collapsed vertices tend to represent minor variances around the state.

We evaluated our approach for collapsing provenance graphs through two different experiments using automatic evaluation techniques and experts. The experimental results showed that at least one of our algorithms (*IC* variant) provided better results than the *DBSCAN* algorithm for collapsing similar segments in the provenance graph.

Figure 4: A fragment of the provenance graph

DBSCAN only surpassed our *IC* variant in one case. The Thesis and FGCS Paper [14] provides more details.

Lastly, a key usage for provenance is to understand how an outcome was reached. As a final contribution, we proposed a provenance graph merge and comparison approach. Our proposed approach can identify possible reasons and discrepancies in a provenance graph that might have led a player to fail to reach the goal. To do so, we contrast the provenance of the game session that failed with the combined provenance of all successful game sessions. We integrated our solution in *Prov Viewer* and provided an experimental study. Our experimental results showed that our comparison approach could detect underlying issues that could have led to failure by comparing with another provenance graph that was known to reach the goal. Furthermore, the results showed that using the optimal *similarity distance metric* results in a 55% *accuracy* with a 40% *retention* rate of the original parameters and actions. However, it can reach as far as 100% *accuracy* when having a *retention* rate close to 5%.

V. Conclusion

This thesis presents a novel approach for tracking and displaying the game provenance data, opening a new research area in Game Analytics for tracking cause-and-effect telemetry data and extracting knowledge from it. The richness and completeness of the provenance data allow for more abstract analysis over tracked game data, broadening the possibilities for different types of analysis and applications for tracked data. This graph can be used for data mining techniques to extract knowledge or used for exploratory analysis by generating a visual representation of the data through a dynamic and interactive node-link diagram. The proposed approach also supports the analysis of multiple game sessions by merging different provenance data to generate a summarized provenance graph that can be used to understand the reasons that led to different outcomes by detecting the differences between game sessions and displaying the probable causes.

The advances presented in the thesis can be divided into **four major contributions**: **(1)** The *PinG for Unity* (*PinGU*) framework, for tracking provenance data, **(2)** *Prov Viewer*, for interactive display and visualization of provenance graphs, **(3)** three distinct summarization techniques, for reducing the overall provenance graph size with minimal information loss, and **(4)** a comparison technique for provenance graphs, to discover underlying issues that might have affected the result.

While the main application of provenance in this work is for games, we believe that the concepts discussed throughout the thesis apply to other domains and might be useful to support advanced forms of analysis. The proposed concepts could be applied in scientific experiments to visualize experiment provenance, to debug the experiment to identify issues, and to understand the obtained results.

VI. Contributions and Publications

The work presented in the thesis resulted in the following publications: Two journal articles [14], [16] and four conference papers [8], [13], [17], [18], including the **SBGames 2017 Best Paper award** for the Computing Track. Moreover, this is a topic that resulted in multiple studies and contributions since when we introduced it in 2012: [7], [19]–[24]. We also have a journal article submission with a requested revision for Entertainment Computing that is directly based on this work using Machine Learning techniques to detect and capture influences during a game session.

Finally, the result of this work created a new line of research relevant to the area of games with the use of provenance. The richness of data obtained through the provenance capture and the structure of the provenance graph allows for a deeper analysis of the data collected and offers more complex visualizations than the simple use of metrics. For example, the provenance graph allows one to easily make inferences because of the structuring of the data in a directed graph. The advantages offered by this new line of research have already culminated in new studies based on the work proposed in this thesis. The result of this thesis is being applied in research projects related to (1) dynamic content balancing in games; (2) reproducibility of sessions; (3) use of the provenance graph in machine learning to generate autonomous agents and detect patterns; (4) views and summaries of the data collected to enable exploratory analysis; (5) debugging of players' performance; and (6) predictive and probabilistic analyzes in games.

References

[1] M. El-Nasr, A. Drachen, and A. Canossa, Eds., *Game Analytics - Maximizing the Value of Player Data*. In: Springer Science & Business Media, 2013.

[2] G. Zoeller, "Development telemetry in video games projects," *Game Dev. Conf. GDC*, 2010.

[3] R. Hunicke, "The Case for Dynamic Difficulty Adjustment in Games," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, New York, NY, USA, 2005, pp. 429–433.

[4] A. Drachen, R. Sifa, C. Bauckhage, and C. Thurau, "Guns, swords and data: Clustering of player behavior in computer games in the wild," *Conf. Comput. Intell. Games CIG*, pp. 163–170, 2012.

[5] C. Pedersen, J. Togelius, and G. N. Yannakakis, "Modeling Player Experience for Content Creation," *Trans. Comput. Intell. AI Games T-CIAIG*, vol. 2, no. 1, pp. 54–67, Mar. 2010.

[6] B. G. Weber, M. John, M. Mateas, and A. Jhala, "Modeling Player Retention in Madden NFL 11," *Innov. Appl. Artif. Intell. Conf. IAAI*, 2011.

[7] T. Kohwalter, E. Clua, and L. Murta, "Provenance in Games," *Braz. Symp. Games Digit. Entertain. SBGAMES*, pp. 162–171, 2012.

[8] T. C. Kohwalter, L. G. P. Murta, and E. W. G. Clua, "Capturing Game Telemetry with Provenance," *Braz. Symp. Games Digit. Entertain. SBGAMES*, 2017.

[9] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," in *Third International AAAI Conference on Weblogs and Social Media*, 2009.

[10] J. Ellson, E. R. Gansner, E. Koutsofios, S. C. North, and G. Woodhull, "Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools," in *Graph Drawing Software*, M. Jünger and P. Mutzel, Eds. Springer Berlin Heidelberg, 2004, pp. 127–148.

[11] M. I. Seltzer and P. Macko, "Provenance Map Orbiter: Interactive Exploration of Large Provenance Graphs," *TaPP*, 2011.

[12] L. Moreau and P. Missier, "PROV-N: The Provenance Notation," 2010. [Online]. Available: http://www.w3.org/TR/prov-n/. [Accessed: 21-Mar-2013].

[13] T. Kohwalter, T. Oliveira, J. Freire, E. Clua, and L. Murta, "Prov Viewer: A Graph-Based Visualization Tool for Interactive Exploration of Provenance Data," in *Proceedings of the 6th International Provenance and Annotation Workshop on Provenance and Annotation of Data and Processes - Volume 9672*, New York, NY, USA, 2016, pp. 71–82.

[14] T. Kohwalter, L. Murta, and E. Clua, "Filtering irrelevant sequential data out of game session telemetry though similarity collapses," *Future Gener. Comput. Syst.*, vol. 84, pp. 108–122, Jul. 2018.

[15] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," *Data Min Knowl Discov*, vol. 2, no. 2, pp. 169–194, Jun. 1998.

[16] T. Costa Kohwalter, F. M. de Azeredo Figueira, E. A. de Lima Serdeiro, J. R. da Silva Junior, L. Gresta Paulino Murta, and E. Walter Gonzalez Clua, "Understanding game sessions through provenance," *Entertain. Comput.*, vol. 27, pp. 110–127, Aug. 2018.

[17] F. Figueira, L. Nascimento, T. Kohwalter, J. R. Silva Junior, L. Murta, and E. Clua, "BinG: A Framework for Dynamic Game Balancing using Provenance," in *Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES)*, 2018.

[18] F. Pinto, T. Kohwalter, V. Braganholo, and L. Murta, "An Infrastructure for Software Release Analysis through Provenance Graphs," presented at the VI Workshop on Software Visualization, Evolution and Maintenance (VEM), 2018.

[19] T. Kohwalter, E. Clua, and L. Murta, "Reinforcing Software Engineering Learning through Provenance," *2014 Braz. Symp. Softw. Eng. SBES*, pp. 131–140, Sep. 2014.

[20] L. Jacob, T. Kohwalter, E. Clua, D. De Oliveira, and A. Machado, "A Non-intrusive Approach for 2D Platform Game Design Analysis Based on Provenance Data Extracted from Game Streaming," *2014 Braz. Symp. Comput. Games Digit. Entertain. SBGAMES*, pp. 41–50, Nov. 2014.

[21] B. Bivar, L. Santos, T. Kohwalter, A. Marinho, M. Mattoso, and V. Braganholo, "Uma Comparação entre os Modelos de Proveniência OPM e PROV," *Braz. E-Sci. Workshop*, 2013.

[22] T. Kohwalter, E. Clua, and L. Murta, "Game Flux Analysis with Provenance," *Adv. Comput. Entertain. ACE*, pp. 320–331, 2013.

[23] L. B. Jacob, T. Kohwalter, A. Machado, and E. W. G. Clua, "A Game Design Analytic System Based on Data Provenance," *Entertain. Comput. – ICEC 2013*, pp. 114–119, 2013.

[24] L. Jacob, E. Clua, and D. de Oliveira, "Oh Gosh!! Why is this game so hard? Identifying cycle patterns in 2D platform games using provenance data," *Entertain. Comput.*, vol. 19, pp. 65–81, Mar. 2017.