# Abstract

Winning or losing a game session is the final consequence of a series of decisions and actions made during the game. The analysis and understanding of events, mistakes, and flows of a concrete game play may be useful for different reasons: understanding problems of gameplay, data mining of specific situations, and even understanding educational aspects in serious games. We introduce a novel approach based on provenance concepts in order to model and represent a game flow. We model the game data and map it to provenance in order to generate a provenance graph, used for analysis. As a proof of concept, we also instantiated our proposed framework and graph generation in a Software Engineering game, allowing players to identify their mistakes and learn through them by analyzing the generated provenance graph from collected gameplay data.

# Resumo

# – Introduction

## Motivation

## Goals

## Research Questions

The construction of the approach and the evaluation of results through experiments have as main objective to answer the following research questions:

* Does provenance analysis help to understand events that emerged during the game?
* Is provenance analysis faster than only watching a replay of the game session?
* Is provenance analysis more accurate than only watching a replay of the game session?

## Contributions

This work introduces new perspectives on gameplay data logging and visualization, to a level where the game provenance can produce and consolidate knowledge. This knowledge can help on (1) confirming the hypotheses formulated by players on how events affected the game, (2) supporting tutors for a better guidance when applying in a serious game, (3) motivating group dynamics around some case studies, and (4) extracting behavior patterns from individual sessions or groups of sessions. The provenance gathering also opens new research possibilities for: behavior pattern data mining, provenance in storytelling, detecting gameplay design issues, gameplay metrics, and gameplay refinement and balance.

The provenance visualization can occur both on-the-fly or in post-mortem sessions. It allows the discovery of issues that contributed to specific game flows and results achieved throughout the gaming session. This analysis can be used on games to improve understanding of the game flow and identifying actions that influenced the outcome, aiding the player to understand why they happened the way they did. It can also be used to analyze a game story development, how it was generated, and which events affected it.

Currently, we do not make inferences to the user, but let the user decide what he wants to infer. Studies in this area can be made in order to identify information that can be automatically omitted from the user without affecting the overall analysis.

## Organization

This work is organized in six other chapters, beside this introduction. Chapter 2 outlines part of the knowledge base necessary for understanding this work. It presents existing usages for gameplay data logging and analysis in the game industry. Then it describes known approaches for gameplay data logging and gameplay data visualization. However, these approaches are designed for aiding developers to identify player strategies and gameplay issues during testing phases.

Chapter 3 outlines the other part of the knowledge base necessary for this work. It describes the concepts of provenance in order to gather historical information about objects for further analysis. It also presents both existing provenance models (OPM and PROV) that can be used for provenance of digital information. Lastly, it presents a comparison between models, pointing out their similarities.

The proposed conceptual framework is presented at Chapter 4, denominated as *provenance in games*. Subsections describe how the gameplay data is gathered, in a game using the conceptual framework, and structured to be used in a provenance graph. Then it describes rules to interpret the gathered data for the provenance graph. Lastly, it describes features to distinguish information in the graph and existing visualization features to aid in the analysis.

Chapter 5 presents the materialization of the conceptual framework presented at Chapter 4, encompassing both the collection and visualization phases. The subsections provide an overview of serious game, describing the game and how the gameplay data is gathered, and implementation details for the provenance graph visualization tool developed in this work. It also presents a guiding game session example, which is the same game session used at Chapter 6 experiment. Lastly, it shows details about the provenance graph generation from the gathered information, the graph representations, and analysis features available at the developed tool by using the guiding example scenario.

After describing the approach and the graph visualization tool, Chapter 6 describes the evaluation performed on the usage of provenance analysis to understand game events. The planning and execution of the experiments are mentioned in subsections indicating how we obtained and processed the data for the assessment. The analysis is detailed through hypothesis testing. Finally, it presents the threats to validity of experiments.

Finally, Chapter 6 concludes this work, listing contributions, limitations and future work.

# – Conclusion

## Contributions

This work introduced a new approach for gameplay data logging and visualization, denominated *provenance in games*. This approach collects gameplay information from executed actions and events by logging related information, including the characters who executed the action and those affected by it. After the completion of the gaming session, the collected information, denominated as *game flow log*, is exported to an external provenance graph visualization tool, *Prov Viewer*. The *game flow log* contains the provenance information from the gaming session and is used by *Prov Viewer* to plot the game’s provenance graph.

The provenance graph allows post game analysis to discover issues that contributed to specific game flows and results achieved throughout the gaming session. This analysis can be used on games to improve understanding of the game flow and to identify actions that influenced the outcome, aiding the player to understand why they happened the way they did. It can also be used to analyze a game story development, how it was generated, and which events affected it.

The provenance graph visualization tool *Prov Viewer* utilizes graphic features to distinguish information for faster comprehension of the events. These features affect the displayed graph by transforming vertices and edges, changing their shapes and color according to the information type. Another important feature present is the information filter, which omit displayed information that is not relevant for the analysis. This filtering is important for analysis because it reduces the amount of displayed information to only those desired to be seen by the user. This allows for faster identification of the influences in the game, which is possible due how the provenance graph is structured.

The usage of provenance graph for analysis provides faster and more precise answers when determining the reasons of events outcomes in a game, in comparison with watching a replay of the game session. Both precision and agility were evaluated by an experiment where players watched a game session and answered a questionnaire with specific questions about certain events that occurred in the game.

While the main application of provenance in this work is over a serious game and is used to assist players in understanding how events affected the story, we believe that the concepts discussed are applicable to other kinds of games and useful to support advanced forms of analysis. The concepts presented may be useful for gameplay balancing and design, behaviors data mining, storytelling, gameplay metrics, and to detect gameplay issues.

## Limitations

One limitation in the *Prov Viewer* is related to scalability, which involves performance. Algorithms in *Prov Viewer* are not optimized for manipulating a graph with thousands vertices and edges. Thus, its performance may be compromised when dealing with such graph sizes.

Another current limitation is based on inferences, which *Prov Viewer* does not automatically make. All inferences must be done by the user. This might cause visualizations problems at a first glance due to the size of the graphs at their full extension. However *Prov Viewer* supports inference rules and provides the necessary features for implementing them.

Currently, *Prov Viewer’s* is not compatible with other provenance applications due to the input file format. The current input format uses an unspecified and undefined structure from the SDM game. Nevertheless, it can be adapted for known formats, such as XML, by modifying how *Prov Viewer* reads a file.

## Future Work

After developing the conceptual framework for data logging and analysis, along with the provenance graph viewer tool *Prov Viewer*, it is possible to describe new research possibilities for the proposed approach. The following sections describe possible researches and improvements in the *provenance in games* conceptual framework and the *Prov Viewer* provenance graph visualization tool.

### ACCESSIBILITY

In relation to accessibility, we propose to make *Prov Viewer* less context sensitive, allowing the user to customize filters (edge filters and attribute status visualization) without the need of hard coding it in the application. For example, allowing the user to provide a configuration file that specifies the type of each filter. Thus, this would make *Prov Viewer* compatible with other games or provenance applications without the need of tinkering with the source code.

Another change would be related to *Prov Viewer’s* input file format. Currently, the *game flow log* is a simple tab separated value text file. However, there are plans to modify the structure to use some semi-structured format such as JSON or XML for greater compatibility with other applications. Thus, it would allow *Prov Viewer* to be more accessible by other applications due to the usage of semi-structured format.

One hypothesis for improving accessibility for *Prov Viewer* would be in creating an extension for existing game engines (i.e. Unity3D). This extension, when enabled, would automatically capture gameplay information when executing events, thus generating a *game flow log* without the need of changing the game’s source code.

### Inferences

One of the current draw-backs of *Prov Viewer* is related to inferences. Currently, we do not make inferences to the user, but let the user or developers decide what needs to be inferred. However, we provide the necessary tools to create inference rules, like filters and collapses (both for vertices and edges). Studies in this area can be made in order to identify information that can be omitted from the user without affecting the overall analysis, while at the same time not being context dependable. Thus it would provide generic inferences rules that can be used in any game.

### Scalability

Another area to be worked upon is related to the graph scalability. Depending on the game style, a game session might take several hours to complete, or even days in case of RPGs. This makes the size of the provenance graph to be overwhelming to the user, even when removing unnecessary information during the generation of the *game flow log* to reduce the log size. One way to avoid such situations is to show the provenance graph with some filters selected instead of its full extension. For example, before showing the graph to the user, it is possible to use collapses to reduce the graph’s size. Combats can be identified and collapsed into a single vertex for each instance. Places visited in the game can also be collapsed into a single vertex, containing all interactions made in that location, even combats. It is also possible to have collapses inside collapses. In this case, a collapsed combat inside a collapsed area visited by the player may contain other actions aside from the combat, such as interactions with the ambient. This gives an impression of a map from the player’s journey, showing vertices for each location visited by the player, while allowing the player to expand only the situations he desires to analyze. It is similar to *google maps*, where it shows the entire world and allows the user to zoom into specific locations. However in this case, it shows instances of the journey taken by the player.

It is also possible to go beyond that. Instead of collapsing all combats and locations, filters can be used to decide which combats or locations were not relevant to the story, or had no noticeable impact in the player’s journey, while keeping important events visible to the player. This is possible because provenance is analyzed from the present to the past. This way, combats outcomes are known and can be used to decide if it they are relevant or not. If the player was victorious with minor challenge, did not suffer severe wounds, or barely used any resources at his disposal, then the entire combat can be simplified into just one vertex representing the combat with the enemy. However, if the combat was challenging or the player lost, it is interesting to display all actions in it for analysis, allowing the player to identify important facts that influenced the combat outcome.

## Possibilities

Although the proposed approach was used to assist the players to understand game events, it can also be used to open new researches in the field of behavior patterns data mining, gameplay design, detecting gameplay issues, and adjusting gameplay features. It might also be possible to modify the proposed approach to be used for artificial intelligence (AI). The AI would collect gameplay information, according to its perception limitations (if any), and analyze provenance information to improve its performance and adapt through past experiences.