|  |
| --- |
| **Comments to author(s)** The paper presents an approach for modeling "provenance" in gameplay data. In this setting, provenance is viewed as a way of capturing cause-effect relationships between players, their actions, game processes, and in-game artifacts. A graph-based modeling approach is described.  The overall approach is to have a record gameplay data using this model, then use an graph data visualization tool called Proof (built on an existing tool called JUNG) to look at the resulting provenance graph. Players of the game can then visually examine the graph, and thereby develop an improved understanding of cause-effect relationships in their prior gameplay.  The paper provides a detailed description of the provenance data model and the visualization approach. Several example gameplay traces gathered from a software engineering training game (SDM) are presented in the paper.  Overall, the provenance idea is interesting. The goal of modeling gameplay traces using a more abstracted model seem promising, as it would seem to support a broader range of analysis of gameplay data. This, in turn, would allow more interesting insights to be gathered from this trace data.  The overall use model presented in the paper appears to be fundamentally flawed. The goal of the effort is to allow a player to better understand cause-effect relationships in their prior gameplay. However, in order to do this, the player must: (a) learn the graph-based provenance model and how it is mapped to specific elements in the game (b) learn how to use the Proof graph visualization tool (c) learn how to filter the data down to a more manageable set (d) be able to interpret the smaller set of data as shown in the Proof tool While step (a) might be reasonable to expect for Software Engineering students, in general it does not seem reasonable to expect a game player to learn a graph-based gameplay model plus supporting visualization tool.  Backing up a step, it is perhaps reasonable to ask why the SDM training game was created in the first place. The general idea is to create an experience that has an underlying software project simulation, but made playable in a way that is more enjoyable than, say, just plugging numbers into a large Excel spreadsheet. Most simulation games are essentially graphical window dressing over a numerical model that can be represented in a spreadsheet.  If the entire goal of the software engineering game is to make the interaction with a software project simulation more enjoyable, why, then, should such a player be required to interact directly with the provenance model via the Proof visualizer? It is somewhat analogous to putting the player "back into the spreadsheet".  What would seem to be the most promising approach is to record the provenance data, then provide \*in-game\* analysis and presentation of the provenance data. This would not take the form of a graph-based visualization, but, instead, elements that are consistent with the visual presentation and fictional world of the game. So, in the software engineering game, one could imagine a specific "employee morale" analysis that \*behind\* \*the\* \*scenes\* analyzes the provenance graph and then has a specific visual presentation of the reasons why players do or don't have high morale, and/or reasons for leaving the company.   Other criticisms of the current approach are that it doesn't scale well (mentioned in the paper), and it is unclear whether players really can reliably interpret the graphs in Proof. In the authors continue with the overall approach of showing results to players in Proof (not recommended), then these two issues need to be addressed. A user study with actual players interacting with Proof is an urgent next step. |
| **Summary of review** An interesting graph-based model is used to capture and represent "provenance" data (cause-effect relationships) in gameplay traces. Graphs are visualized in a system called Proof and shown to players. The overall approach has poor scalability characteristics, since graphs can grow quickly. No user study was performed to validate that players can actually perform filters to reduce graph size and then interpret the resulting visualized provenance graph data. It seems unlikely that game players from the general population would be able to use this visualization. |