Proceedings of the Twenty-First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2025)

State Space Visualization for Strong Story Experience Management Design

Justus Robertson[1], **Valentina Genoese-Zerbi**[1], **Rogelio E. Cardona-Rivera**[2]

[1]School of Interactive Games and Media, Rochester Institute of Technology, Rochester, NY, USA
[2]The Utah Division of Games, University of Utah, Salt Lake City, UT, USA
jjrigm@rit.edu, vg1566@rit.edu, r.cardona.rivera@utah.edu

Abstract

This paper presents a software library that enumerates the space of a state transition system specified by an action language, visualizes the states and action connections as a graph, and modifies the visualization based on underlying features determined through state and graph analysis. The library is intended as a tool for strong story interactive narrative design.

Introduction

Strong story experience managers (Riedl and Bulitko 2013) control the design and characters in interactive story games to balance player autonomy with author requirements that infuse narrative structure into game events. An open problem in these systems is how to avoid *dead-ends*, states in the story game world's transition system that softlock the player into story sequences that do not contain the narrative structure specified by the game author. The standard approach for avoiding dead-ends is to navigate around these areas through NPC actions (Robertson and Young 2018; Ware et al. 2022) or more direct interventions (Young et al. 2004; Robertson, Cardona-Rivera, and Young 2020), decreasing player autonomy or simulation cohesion to preserve narrative structure. We propose a second approach that more closely mirrors level design in other game genres: change the game design to decrease or eliminate the accessible softlock states.

Software Tool

As a first step towards an integrated development environment (IDE) to help human authors identify and prevent narrative softlocks prior to runtime, we present a tool that automatically analyzes the state space of a formally specified strong story experience management system and help human authors understand the game's state space and its features. The tool provides the following in its analysis: 1.) An enumeration of all unique states in the state space; 2.) An enumeration of all actions that lead from one unique state to another; 3.) The Strongly Connected Components (SCCs) within the graph; 4.) Classification of nodes into categories based on how many goals are achieved or if it is a dead-end; and 5.) A visualization of all these features in a graph generated by the Python Netgraph library (Brodersen 2023).

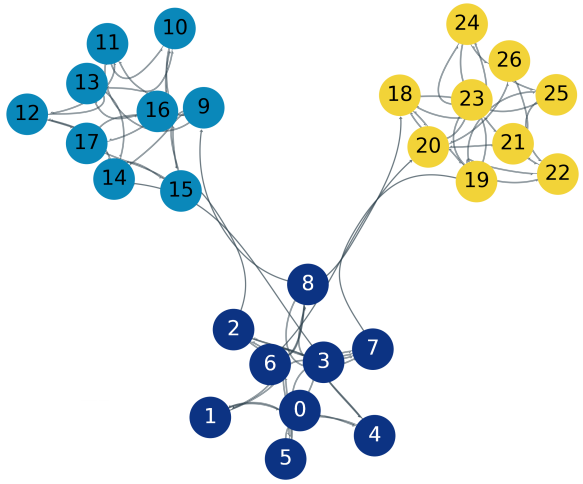

Figure 1: Visualization of a game with 26 unique states and associated action connections, grouped by strongly connected components (SCC). Nodes are colored according to the • Dead-End Region, • Neutral Region, or • Goal Region.

The underlying formal symbolic systems used are specified in an action language similar to typed Planning Domain Definition Language (PDDL) (McDermott 2000) that can be written directly in Python code. These formal worlds are meant to be played like simple text adventures. The examples are compiled and run to build the state space and identify strongly connected components. The Netgraph Python library (Brodersen 2023) is then used to visualize the results. It creates an image file that visualizes possible states as nodes, clustered based on the strongly connected components, and colored based on the number of goals achieved and whether it's a dead-end. This graph image is paired with a text file that enumerates state, connection, SCC, and state classification information as a reference for the visualization. SCCs are a helpful visualization tool because any node that is in an SCC with a dead-end is a dead-end itself. Example output is shown in Figure 1, with three regions: • softlocked dead-ends, • gameplay, and • goal regions. By viewing the graph topography at design-time, designers can alter or eliminate dead-end regions prior to gameplay.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. #2303650.

References

Brodersen, P. J. N. 2023. Netgraph: Publication-quality Network Visualisations in Python. *Journal of Open Source Software*, 8(87): 5372.

McDermott, D. M. 2000. The 1998 AI Planning Systems Competition. *AI Magazine*, 21(2): 35.

Riedl, M. O.; and Bulitko, V. 2013. Interactive narrative: An intelligent systems approach. *Ai Magazine*, 34(1): 67–67.

Robertson, J.; Cardona-Rivera, R. E.; and Young, R. M. 2020. Invisible dynamic mechanic adjustment in virtual reality games. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, 282–289. IEEE.

Robertson, J.; and Young, R. M. 2018. Perceptual experience management. *IEEE Transactions on Games*, 11(1): 15–24.

Ware, S.; Garcia, E. T.; Fisher, M.; Shirvani, A.; and Farrell, R. 2022. Multiagent narrative experience management as story graph pruning. *IEEE Transactions on Games*, 15(3): 378–387.

Young, R. M.; Riedl, M. O.; Branly, M.; Jhala, A.; Martin, R.; and Saretto, C. 2004. An architecture for integrating plan-based behavior generation with interactive game environments. *J. Game Dev.*, 1(1): 1–29.