Integrating Automated Play in Level Co-Creation

Andrew Hoyt[1], Matthew Guzdial[2], Yalini Kumar[1], Gillian Smith[3], and Mark O. Riedl[1]
[1]School of Interactive Computing, Georgia Institute of Technology
[2]Department of Computing Science, University of Alberta
[3]Computer Science Department, Worcester Polytechnic Institute
andrewhoyt@gatech.edu, guzdial@ualberta.ca, ysk3@gatech.edu, gmsmith@wpi.edu, riedl@cc.gatech.edu

Abstract

In level co-creation an AI and human work together to create a video game level. One open challenge in level co-creation is how to empower human users to ensure particular qualities of the final level, such as challenge. There has been significant prior research into automated pathing and automated playtesting for video game levels, but not in how to incorporate these into tools. In this demonstration we present an improvement of the *Morai Maker* mixed-initiative level editor for Super Mario Bros. that includes automated pathing and challenge approximation features.

Introduction

Morai Maker is a general video game level editor (Guzdial et al. 2017), which can load any set of sprites (game entities), which can then be used to construct new levels. However, it's development has focused on level design for the game Super Mario bros.. The editor has been augmented with a variety of machine learning-based AI level design partners who work with the human user in a turn-based fashion (Guzdial, Liao, and Riedl 2018). It has shown success in human subject studies with both novice and expert game designers (Guzdial et al. 2019).

We identify two concerns raised by human users of *Morai Maker*, based on the qualitative and quantitative results of prior human subject studies (Guzdial et al. 2019). First, that the AI partner can make additions to the level that are not guaranteed to be playable. This concern arises from the fact that none of the four AI agents built for *Morai Maker* has an explicit model of the physics in Super Mario Bros.. Thus there's no way for the AI partner to ensure that the additions to the level it creates allow for a fully playable level–a level that can be completed from beginning to end. However, this is not as simple as adding a playability check to any AI additions. Level design in *Morai Maker* is intentionally iterative. Therefore at any point in time the level may be unplayable because it is not yet complete.

The second concern is that there's no way in the editor to ensure level content is reachable without the user playing the level. While *Morai Maker* allows users to play through the current level, this can be challenging and frustrating given that it can break up the flow of level design. It also forces level creators to only design levels that they personally can play themselves, if they care about level playability.

To address these two concerns in this demonstration paper we present two new features for *Morai Maker*. The first is an A* Reachability Check, which allows a user to determine whether it is possible as Mario to go from any two arbitrary points. The second is an A*-based Survival Analysis (Isaksen, Gopstein, and Nealen 2015), giving users a rough approximation of the difficulty of navigating some section of the level. We briefly discuss related, prior work and then detail these two new features.

Related Work

Morai Maker can be considered a co-creative example of procedural level generation via machine learning (PCGML) (Summerville et al. 2018). Such tools have been previously discussed in the literature (Summerville et al. 2018; Zhu et al. 2018), but are still underexplored. Comparatively there exist many prior approaches to co-creative or mixed-initiative level design without machine learning (Yannakakis, Liapis, and Alexopoulos 2014; Deterding et al. 2017). These approaches instead rely upon search or grammar-based approaches (Smith, Whitehead, and Mateas 2011; Liapis, Yannakakis, and Togelius 2013; Shaker, Shaker, and Togelius 2013; Baldwin et al. 2017; Alvarez et al. 2018). However, none of these prior approaches include automated playtesting and challenge analysis systems like the features presented in this paper.

Cicero (Machado et al. 2018) is a mixed-initiative game development tool built within the General Video Game AI (GVGAI) framework (Perez-Liebana et al. 2018). Therefore, it also has access to a variety of general GVGAI AI agents for automated playtesting (Horn et al. 2016; Khalifa et al. 2016). However, because these agents are general they cannot be employed in the same way as the features we present, which rely on finding the optimal path.

Automated play has a rich history in the field of game AI. Our work specifically relates to the Mario AI Benchmark, Competitions, and Championships (Karakovskiy and Togelius 2012; Togelius et al. 2013). Specifically, the dis-

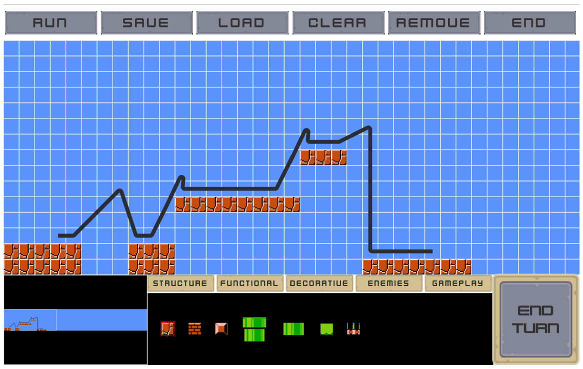

Figure 1: Screenshot of a calculated path displayed over a user-constructed level section.

covery that an A* agent could outperform all other automated game playing agents for Super Mario Bros. when given access to the game's rules. This directly lead to our choice to rely on A* as the basis for both new features. We draw on Isaksen et al.'s conception of survival analysis (Isaksen, Gopstein, and Nealen 2015; Isaksen and Nealen 2015) as an approach for determining game difficulty by simulating many agents through a game or level.

A* Reachability Check

The A* Reachability Check is the first of the features we introduce to address user concerns with level playability and challenge. We visualize an example output of the feature in Figure 1. You can see some example level geography created by a user of the system (the orange ground blocks). To employ the reachability check a user pressed the "P" key on the keyboard and then clicks somewhere with the mouse. A black line is drawn from where the first mouse click occurs until the mouse is clicked again. At this point, if it is possible given the mechanics of Super Mario Bros. for a player to traverse from the first point to the second the optimal path is drawn. Otherwise no line is drawn. This feature required the representation of Super Mario Bros. physics as A* path finding operators, which follows prior work in this area (Karakovskiy and Togelius 2012; Togelius et al. 2013).

This feature does not guarantee that the visualized path is the only possible path, but it does allow a user to check if any path is possible between two points. Thus users can check whether a level is possible to complete without having to or even having to be able to complete the level themselves. However, this feature alone does not reflect the potential difficulty of paths outside of the one visualized path.

A*-based Survival Analysis

The second feature is based upon Isaksen et al.'s Survival Analysis, in which many agents attempt to play through a level (Isaksen, Gopstein, and Nealen 2015; Isaksen and Nealen 2015). These agents differ in the optimality of their play, which means one can employs Survival Analysis to determine a rough notion of challenge in a level (e.g. how un-optimal can one play and still make it to the end). Our second feature presents a version of this for Super Mario Bros., which takes the A* path and sends a designer-chosen number of agents along this path (with a default of 100). Each agent has a randomly chosen threshold of how close they must be to a particular node on the A* path before attempting to reach the next one. This allows for the variance in optimality of these agents. To interact with the feature a user makes a path via the reachability check feature as seen on the top of Figure 2. If there is a current path, hitting run will send the agents along that path, as can be seen at the bottom of Figure 2. The top right corner of the screen will then count up the number of agents that make it to the final goal, as a rough approximation of challenge.

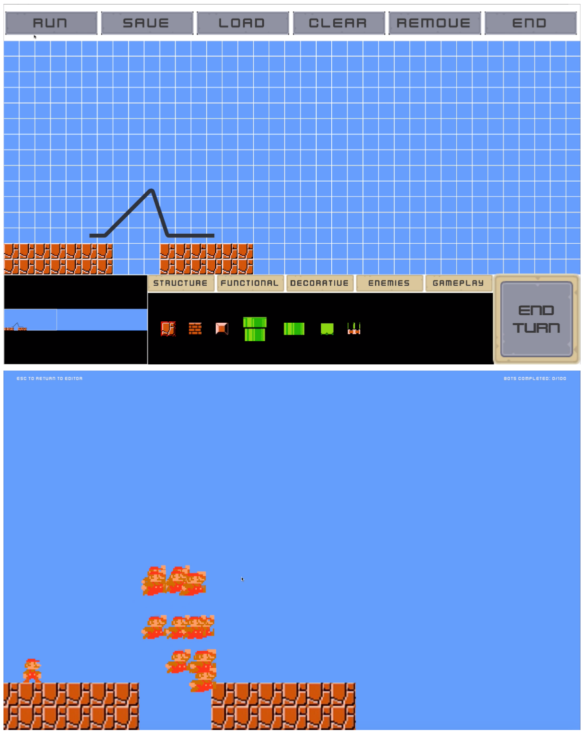

Figure 2: Example of the path (top) and corresponding in-progress survival analysis (bottom).

Due to the variability in assigning random thresholds to A* agents, the final proportion of successful agents can differ across new runs of the same path. In addition, this proportion should not be taken as equivalent to perceived human difficulty, as humans play differently than A* agents. However, it can still be a useful measure for comparing difficulty of different level sections built within the editor.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1525967. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Alvarez, A.; Dahlskog, S.; Font, J.; Holmberg, J.; Nolasco, C.; and Österman, A. 2018. Fostering creativity in the mixed-initiative evolutionary dungeon designer. In *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 50. ACM.

Baldwin, A.; Dahlskog, S.; Font, J. M.; and Holmberg, J. 2017. Mixed-initiative procedural generation of dungeons using game design patterns. In *Computational Intelligence and Games (CIG), 2017 IEEE Conference on*, 25–32. IEEE.

Deterding, C. S.; Hook, J. D.; Fiebrink, R.; Gow, J.; Akten, M.; Smith, G.; Liapis, A.; and Compton, K. 2017. Mixed-initiative creative interfaces. In *CHI EA'17: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM.

Guzdial, M. J.; Chen, J.; Chen, S.-Y.; and Riedl, M. 2017. A general level design editor for co-creative level design. In *Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Guzdial, M.; Liao, N.; Chen, J.; Chen, S.-Y.; Shah, S.; Shah, V.; Reno, J.; Smith, G.; and Riedl, M. O. 2019. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 624. ACM.

Guzdial, M.; Liao, N.; and Riedl, M. 2018. Co-creative level design via machine learning. *arXiv preprint arXiv:1809.09420*.

Horn, H.; Volz, V.; Pérez-Liébana, D.; and Preuss, M. 2016. Mcts/ea hybrid gvgai players and game difficulty estimation. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.

Isaksen, A., and Nealen, A. 2015. Comparing player skill, game variants, and learning rates using survival analysis. In *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*.

Isaksen, A.; Gopstein, D.; and Nealen, A. 2015. Exploring game space using survival analysis. In *FDG*.

Karakovskiy, S., and Togelius, J. 2012. The mario ai benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games* 4(1):55–67.

Khalifa, A.; Isaksen, A.; Togelius, J.; and Nealen, A. 2016. Modifying mcts for human-like general video game playing. In *IJCAI*, 2514–2520.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*, 213–220. FDG.

Machado, T.; Gopstein, D.; Nealen, A.; Nov, O.; and Togelius, J. 2018. Ai-assisted game debugging with cicero. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. IEEE.

Perez-Liebana, D.; Liu, J.; Khalifa, A.; Gaina, R. D.; Togelius, J.; and Lucas, S. M. 2018. General video game ai: a multi-track framework for evaluating agents, games and content generation algorithms. *arXiv preprint arXiv:1802.10363*.

Shaker, N.; Shaker, M.; and Togelius, J. 2013. Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels. In *Proceedings of the Ninth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):201–215.

Summerville, A.; Snodgrass, S.; Guzdial, M.; Holmgård, C.; Hoover, A. K.; Isaksen, A.; Nealen, A.; and Togelius, J. 2018. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games* 10(3):257–270.

Togelius, J.; Shaker, N.; Karakovskiy, S.; and Yannakakis, G. N. 2013. The mario ai championship 2009-2012. *AI Magazine* 34(3):89–92.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the 9th Con- ference on the Foundations of Digital Games*. FDG.

Zhu, J.; Liapis, A.; Risi, S.; Bidarra, R.; and Youngblood, G. M. 2018. Explainable ai for designers: A human-centered perspective on mixed-initiative co-creation. *Computational Intelligence in Games*.