Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-20)

A Demonstration of Anhinga: A Mixed-Initiative EPCG Tool for Snakebird

Nathan R. Sturtevant,[1] Nicolas Decroocq,[2] Aaron Tripodi,[3] Carolyn Yang,[3] Matthew Guzdial[1]
[1]Department of Computing Science, Alberta Machine Intelligence Institute (Amii), University of Alberta, Canada
[2]ESM Saint-Cyr, 1er Bataillon de France, F-56380 GUER Cedex, France
[3]Department of Computing Science, University of Alberta, Canada
{nathanst, tripodi, carolyn4, guzdial}@ualberta.ca, nicolasdecroocq1@gmail.com

Abstract

Mixed-initiative procedural content generation (PCG) refers to systems where a human and AI cooperate in some way to produce content. While there has been increasing interest in research on these systems, there are still many domains and PCG approaches that have not yet been explored. In this demonstration we introduce a novel mixed-initiative tool that employs Exhaustive PCG for puzzle level design.

Introduction

Mixed-initiative procedural content generation (PCG) includes "a broad range of generators, algorithms, and tools" that all "require human input in order to be of any use" (Liapis, Smith, and Shaker 2016). This has been an active area of research, however the majority of research has focused on evolutionary search (Liapis, Yannakakis, and Togelius 2013; Baldwin et al. 2017) and/or on platformer games (Smith, Whitehead, and Mateas 2010; Guzdial et al. 2019; Liapis 2020). Thus, we still lack a breadth of understanding of mixed-initiative systems. Towards the goal of increasing this breadth of understanding, we introduce a mixed-initiative system employing an underexplored PCG approach and game domain. Specifically, a mixed-initiative exhaustive PCG (EPCG) (Sturtevant and Ota 2018) approach for the puzzle game *Snakebird*.

The most common approach for search-based mixed-initiative PCG (SBPCG) (Togelius et al. 2011) is to present a user with random variations of a current state. However, *Snakebird* is a complex puzzle game, and thus there is a high likelihood that random changes to a level would lead to an unplayable level. For a similar reasons, it would be inappropriate to draw on PCG via Machine Learning (Summerville et al. 2018), as any noise in the machine learned model could lead to an unplayable level. Comparatively, EPCG has shown success with similarly complex puzzle games (Sturtevant 2019). Thus, we employ it to find a single change to a level by exhaustively searching all possible changes and selecting the one that most maximizes some evaluation function.

In this demonstration we present a mixed-initiative clone of *Snakebird* that we call Anhinga. In Anhinga a user is able to play a subset of levels from the original *Snakebird* game and query an EPCG backend system to find the single best change to the current level according to the evaluation function (Sturtevant et al. 2020). We note that while the system is an example of mixed-initiative PCG, it is not yet co-creative (Yannakakis, Liapis, and Alexopoulos 2014) due to the limitations of the human-AI interaction. This is, however, our intention for future work. In this abstract we cover related prior work, an overview of the demonstration, and discuss the demonstration and future directions.

Related Work

Mixed-initiative PCG tools are those in which a human and an AI work together to produce game content (Liapis, Smith, and Shaker 2016). One benefit of these tools is that they allow a user to offload some cognition to the tool, whether that be translation of an initial idea into a piece of concept (Smith, Whitehead, and Mateas 2010), filling in gaps in a piece of content in development (Guzdial et al. 2019), or generation of variations on the content (Liapis, Yannakakis, and Togelius 2013). Our tool targets the last of these tasks. Mixed-initiative tools based on producing variations on content are relatively common, with the majority making use of interactive evolution (Baldwin et al. 2017; Schrum et al. 2020; Charity, Khalifa, and Togelius 2020). In these approaches, a user acts as an evaluator, repeatedly choosing between sets of random variations on a current piece of level content. In comparison, our tool allows a user to query for the one change out of all possible changes that maximize or minimize an evaluation function.

Demo Overview

Snakebird is a 2015 game by Noumenon Games, with a 2019 follow-up called Snakebird Primer. The goal of each level is to have one or more snakebirds eat all of the fruit in the level and then for all snakebirds to leave via the exit. Our research clone of Snakebird is called Anhinga, named after a bird which, when swimming, can look like a "snake moving through the water" (Henderson 2010).

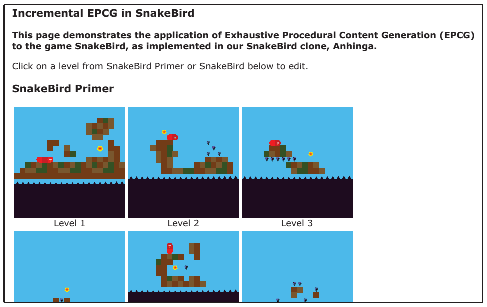


Figure 1: Anhinga landing page.

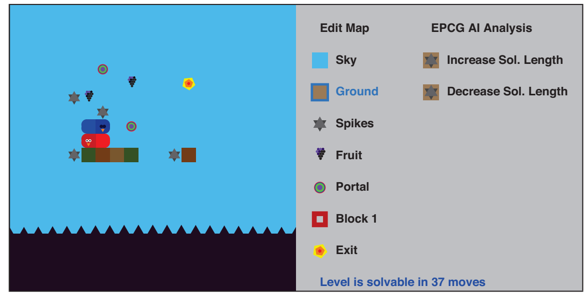


Figure 2: Anhinga gameplay and editing screens.

Anhinga is written in C++ and compiled to JavaScript for use on the web using emscripten. It uses a custom graphics library to provide cross-platform support to several different platforms; this demo is rendered using the HTML5 canvas element. The current landing page for the demo, shown in Figure 1 and available from https://www.movingai.com/snakebird-editor.html, provides a list of levels, which can be loaded into a separate window to edit and play. The play window contains a link to detailed instructions for playing and buttons for controlling the snakebirds. There are also buttons for editing a level, solving a level, undoing actions, and resetting the level.

The mixed-initiative EPCG code can be used from within the level editor. A level with the current version of the edit screen is shown in Figure 2. This screen appears adjacent to the level, and allows editing gameplay elements or applying the EPCG analysis. The EPCG analysis is used to suggest a change to the level that maximally increases or decreases the optimal solution length. When making changes, the editor reports the difference in solution length. Note that memory limits restrict which levels can be solved in the browser, but we choose to use the browser for this demo to maximize the accessibility of the demo.[1]

Figure 3(a) and 3(b) show one example of the before and after results when using EPCG to increase the solution length. This is the result of asking the EPCG system to increase the optimal solution length in SnakeBird Primer Level 2. The system adds a single spike between the fruit which, in this case, increases the solution length from 28 to 41. In our informal tests on this level, this single change makes the level far more difficult and requires a deeper understanding of level mechanics to solve.

If a level is too difficult for a user to solve, there is a solve button which solves the level and then animates the solution. The editor currently displays the optimal solution length of a level and also offers simple editing of other gameplay elements. The capabilities of the editor are still under active development and will continue to change and be updated as we implement new capabilities.

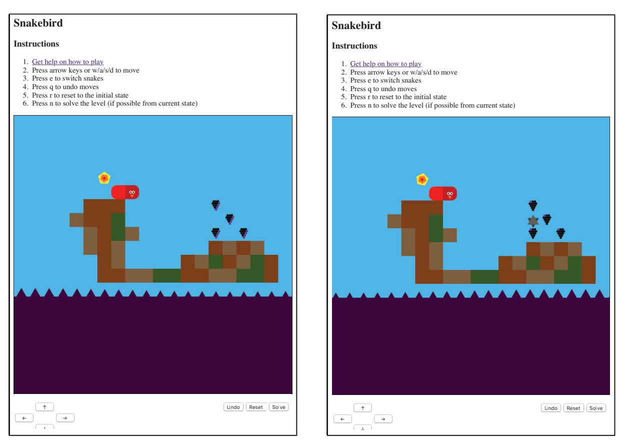
[1]Native executables which are more performant can be compiled for different platforms from the freely available source: https://github.com/nathansttt/hog2/

Backend EPCG

Exhaustive PCG requires a generator and an evaluator (Sturtevant and Ota 2018). The task of the generator is to create all possible content in a given scenario. The evaluator is applied to all content and the content with the maximum evaluation is returned. In our tool, the generator produces all single-tile changes in a level, where a single tile can be either sky, ground, or spikes. The evaluator uses a breadth-first search to find the shortest solution to each level given all possible modifications proposed by the generator (Sturtevant et al. 2020). We can either select the change that maximizes or minimizes the optimal solution length. Choosing the change that maximizes the solution length generally makes a level more difficult, where minimizing solution length makes levels easier.

Discussion

The premise of our research in this area (Sturtevant et al. 2020) is that EPCG can be used as part of a mixed-initiative tool to explore interesting design changes to game levels. This demonstration of our mixed-initiative EPCG *Snakebird* tool, Anhinga, allows users to interact with Anhinga and experience the process of working with an EPCG system. In the future, we plan to expand Anhinga into a full co-creative system (Yannakakis, Liapis, and Alexopoulos 2014), allow-



(a) (b)

Figure 3: Anhinga EPCG examples.

ing users to use EPCG in the level design process. To evaluate this future system, we plan on running a formal human subject study. Specifically, our plans are to draw on the study design used in (Guzdial et al. 2019), having a population of level design practitioners interact with the tool in a thinkaloud framework in order to analyze the impact of EPCG on user experience more formally.

Mixed-initiative systems for puzzle games are still underexplored. Anhinga represents one approach to this task, but it is not the only possible approach (Charity, Khalifa, and Togelius 2020). Still, we anticipate that the basic EPCG approach should generalize to compact puzzle games like *Snakebird*, along with *Fling* and *The Witness*, given that EPCG has previously been applied in these domains (Sturtevant and Ota 2018; Sturtevant 2019). We hope to more formally compare EPCG and other PCG methods for puzzle design in a mixed-initiative context in future work.

Conclusions

In this demonstration we introduce Anhinga, a clone of Snakebird that includes a mixed-initiative system drawing on exhaustive PCG (EPCG) for puzzle design. This demonstration is the first example of applying EPCG to mixed-initiative design. We hope that this will help expand the breadth of understanding of mixed-initiative systems.

Acknowledgements

This work was funded by the Canada CIFAR AI Chairs Program. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). This work was supported by the Ross and Verna Tate HIP program.

References

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)*, 25–32. IEEE.

Charity, M.; Khalifa, A.; and Togelius, J. 2020. Baba is y'all: Collaborative mixed-initiative level design. *arXiv preprint arXiv:2003.14294*.

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*, 1–13.

Henderson, C. L. 2010. *Birds of Costa Rica a field guide / Carrol L. Henderson ; with photographs by the author ; illustrations by Steve Adams ; foreword by Alexander F. Skutch.* Corrie Herring Hooks series ; no. 64. Austin: University of Texas Press.

Liapis, A.; Smith, G.; and Shaker, N. 2016. Mixed-initiative content creation. In *Procedural content generation in games*. Springer. 195–214.

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

Liapis, A. 2020. 10 years of the pcg workshop: Past and future trends. In *FDG Workshop on Procedural Content Generation*.

Schrum, J.; Gutierrez, J.; Volz, V.; Liu, J.; Lucas, S.; and Risi, S. 2020. Interactive evolution and exploration within latent level-design space of generative adversarial networks. *arXiv preprint arXiv:2004.00151*.

Smith, G.; Whitehead, J.; and Mateas, M. 2010. Tanagra: A mixed-initiative level design tool. In *Foundations of Digital Games (FDG)*, 209–216.

Sturtevant, N. R., and Ota, M. J. 2018. Exhaustive and semi-exhaustive procedural content generation. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 109–115.

Sturtevant, N. R.; Decroocq, N.; Tripodi, A.; and Guzdial, M. 2020. The unexpected consequence of incremental design changes. In *Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*.

Sturtevant, N. 2019. Exploring epcg in the witness. In *AAAI Knowledge Extraction from Games Workshop*.

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). *Transactions on Games* 10(3):257–270.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *Transactions on Computational Intelligence and AI in Games* 3(3):172–186.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Foundations of Digital Games (FDG)*.