Constrained Videogame Content Generation with Answer Set Programming

Adam M. Smith (amsmith@soe.ucsc.edu)
Expressive Intelligence Studio, UC Santa Cruz

ABSTRACT

Automatically generating content for videogames (often called procedural content generation or PCG) requires developing systems that can synthesize artifacts that are both deeply relevant to a designer's vision and appropriate for a player's experience in a game so far. PCG work has historically proceeded by developing algorithms for each combination of content type and design requirements. As ill-defined requirements on generated content are refined, the design of generative procedures increasingly requires expertise in combinatorial search that exceeds any reasonable bound on a game developer's skillset. To address this problem, recent work [3] shows how to declaratively model videogame content *design spaces* (specifying what to generate instead of how to generate it) as very concise answer set programs (ASP is a modern development from the logic programming community [1]). Powered by domain-independent algorithms capable of efficiently solving NP-hard problems, ASP offers a way to rapidly produce systems that are able to much more efficiently and expressively generate artifacts from tightly-constrained design spaces than a fair comparison with bespoke algorithm design by competent developers who are not experts in combinatorial search [2].

BODY

Answer set programming allows the rapid development of tightly-constrained videogame content generators without inventing new algorithms.

REFERENCES

[1] M. Gebser, B. Kaufmann, R. Kaminski, M. Ostrowski, T. Schaub, and M. T. Schneider. Potassco: The Potsdam answer set solving collection. *AI Commun.*, 24(2):107–124, 2011.

[2] A. M. Smith, E. Andersen, M. Mateas, and Z. Popović. A case study of expressively constrainable level design automation tools for a puzzle game. In *Proceedings of the 7th International Conference on the Foundations of Digital Games*, 2012.

[3] A. M. Smith and M. Mateas. Answer set programming for procedural content generation: A design space approach. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):187–200, 2011.