Procedural Content Generation for Computer Games

What is Procedural Content Generation?

- Procedural Content Generation is a field with no widely agreed definition:
  - “the application of computers to generate game content, distinguish interesting instances among the ones generated, and select entertaining instances on behalf of the players.” - Hendrikx et al. [1].
  - “the algorithmical creation of game content with limited or indirect user input.” - Togelius et al. [2].
- PCG techniques often make use of approaches from the related fields of dynamic difficulty adjustment, particle systems, evolutionary algorithms, cellular automata and AI.

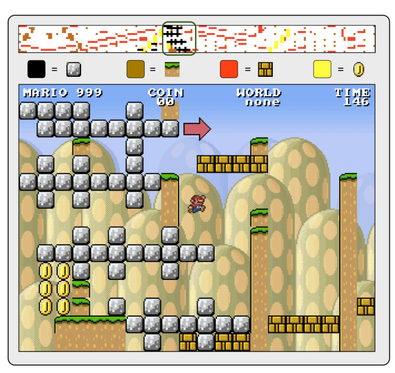

Figure 1: PCG Mario [3]

Online vs Offline

- “Offline” PCG
  - Performed at development or load time
- “Online” PCG
  - Performed during game execution
  - Can use runtime information

	Traditional	Offline	Online
Reduced Workload		✓	✓
Reduced Storage		✓	✓
More Variety		✓	✓
Responsive			✓

PCG for Artists

Artists use PCG techniques to make the production of multiple content types more efficient & varied:

- Textures
  - Perlin noise and PRNGs
  - Image Filtering techniques
- Models
  - Component-Assembly method using generative grammars
- Music
  - Responsive generated music
- Animations
  - Offline automatic completion
  - Online reactive generation
  - Player–driven generation
- Effects
  - Procedural particle systems
  - Procedural rendering effects (see figure 2)

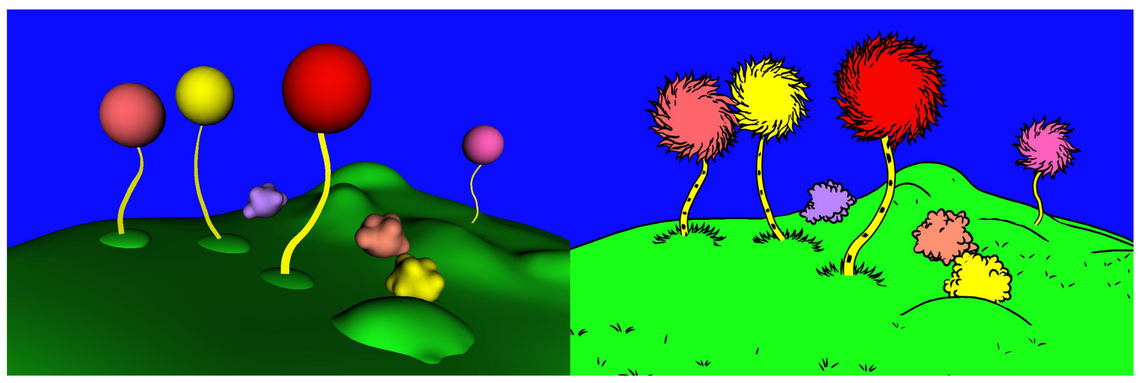

Figure 2: Procedural rendering of Seussian grass and trees, from [4].

PCG for Designers

The use of PCG methods bring several benefits to developers, enabling:

- Content Scale enhancements
  - PCG approaches can populate large handcrafted areas
  - Near–limitless areas can be generated from scratch
- Replay Value improvement
  - Generated content can vary between playthroughs
- Challenge Adjustment systems
  - Online algorithms can use player metrics to adjust the challenge of procedurally generated content

Two particularly active research areas are:

- Search-based PCG techniques
  - Fitness functions evaluate generated content
- Mixed-Initiative content creation
  - Developer and PCG algorithm work in tandem

PCG for Consumers

Some games use information about the player to tailor content at runtime, to improve:

- Experience of gameplay overall
  - Valve’s ‘AI Director’ – maintains emotional intensity
  - Bethesda’s ‘Radiant Story’ – encourages exploration
- Agency available to players
  - ‘Weapons Lab’ in Galactic Arms Race – custom weapons
  - ‘Creature Creator’ in Spore – procedural creatures

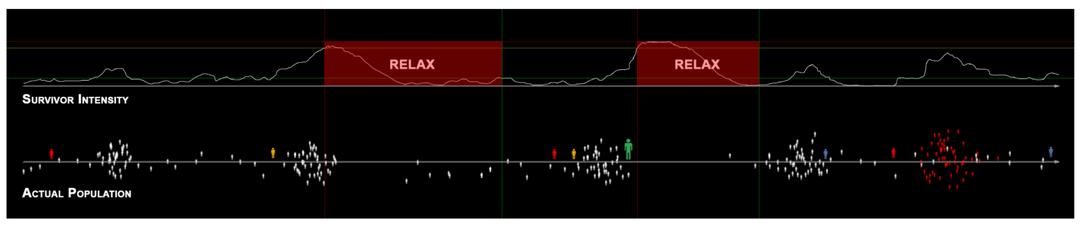

Figure 2: Generation of opponent population by AI Director [5]

Conclusions & Future Work

PCG techniques can improve the process of content creation, providing:

- Increased efficiency
- Greater variation
- Tailored content

Commercially successful games use PCG for many purposes. However:

- Implementations are bespoke
- No standard approaches exist

This hinders research into the use of PCG with other fields by dividing focus between incompatible methods. Further work is needed to:

- Develop domain-independant methods to standardise approaches
- Investigate combination with related research areas (DDA, AI, etc)

References:

[1] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: a survey. 2012.
[2] J. Togelius, E. Kastbjerg, D. Schedl, and G. Yannakakis. What is procedural content generation?: Mario on the borderline. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, page 3. ACM, 2011.
[3] M. Kerssemakers, J. Tuxen, J. Togelius and G. Yannakakis. A procedural procedural level generator generator. In Proceedings of the 2012 IEEE Conference on Computational Intelligence and Games (CIG’12), pages 335-341, IEEE, 2012 .
[4] M. A. Kowalski, L. Markosian, J. Northrup, L. Bourdev, R. Barzel, L. S. Holden, and J. F. Hughes. Art-based rendering of fur, grass, and trees. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 433–438. ACM, 1999. [5] M. Booth. The AI systems of Left 4 Dead. In *Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE ’09)*, October 14-16, 2009. [©] Battlefish © Maxis Entertainment, Spore promotional material.

Author: Tom Smith (taes1g09@ecs.soton.ac.uk)
Supervisor: Rikki Prince (rfp@ecs.soton.ac.uk)

Electronics and Computer Science
UNIVERSITY OF Southampton