



Procedural Content Generation for Computer Games

What is Procedural Content Generation?

- ▶ Procedural Content Generation (PCG) is a broad 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].
- ▶ Applications of PCG techniques often make use of approaches from the fields of dynamic difficulty adjustment, particle systems, evolutionary algorithms, cellular automata and AI.

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 fig. 1)

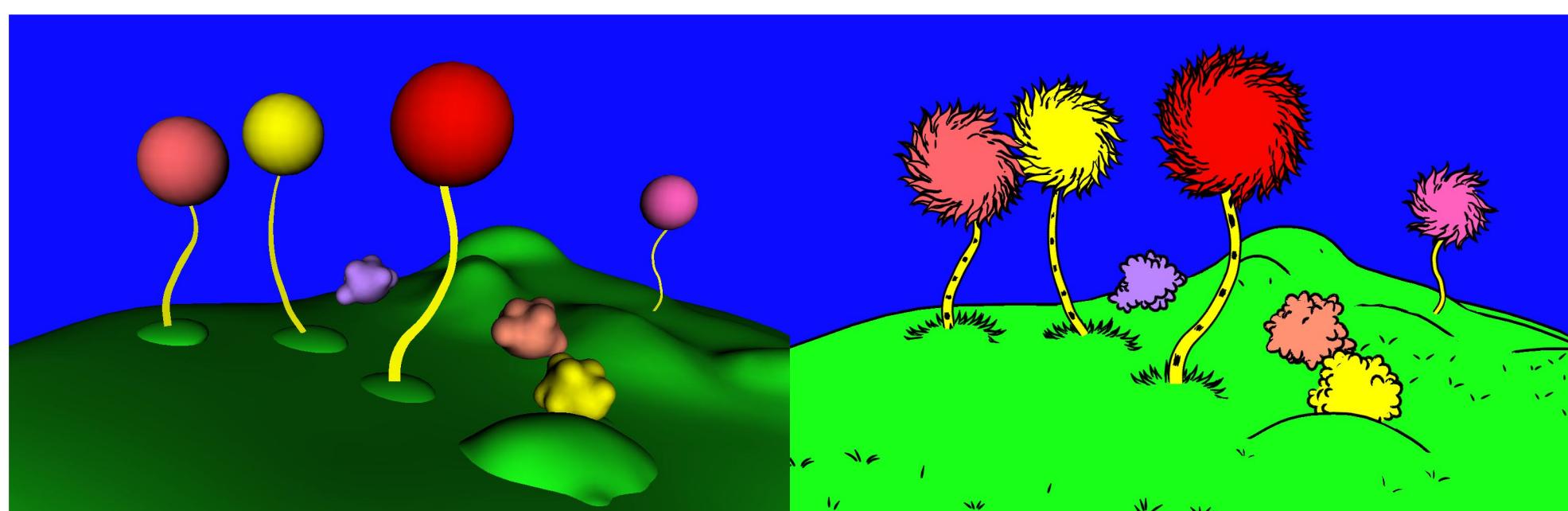


Figure 1: Procedural rendering of grass and trees, from [3].

Conclusions & Future Work

PCG techniques can greatly improve the efficiency of content creation, and lead to varied and tailored content. Many commercially successful games have used a range of PCG methods for a number of purposes. However, many existing implementations are bespoke, and the lack of standard approaches hinders research in the area by dividing focus between incompatible methods. Further work is needed to develop domain-independent methods, and there is scope for investigation into the combination of PCG with related research areas.

PCG for Designers

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

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

Two of the active research areas are:

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

PCG for Users

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

- ▶ Experience
 - ▷ Valve's AI Director – maintains emotional intensity
 - ▷ Bethesda's Radiant Story – encourages exploration
- ▶ Agency
 - ▷ Weapons Lab in Galactic Arms Race – custom weapons
 - ▷ Creature Creator in Spore – procedural creatures

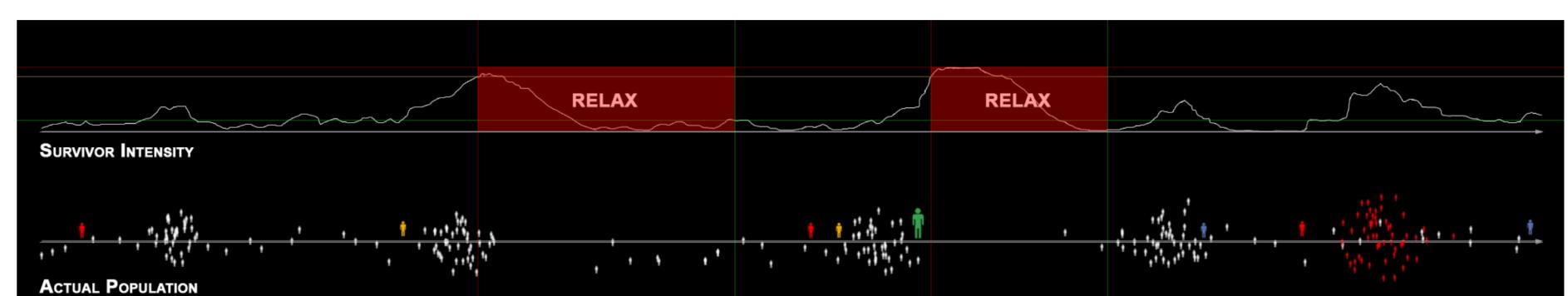


Figure 2: Generation of opponent population by AI Director [4]

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. Kastberg, 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. 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 Press/Addison-Wesley Publishing Co., 1999.
- [4] M. Booth. The AI systems of Left 4 Dead. In *Keynote, Fifth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE '09)*, Stanford, CA, October 14-16, 2009. Battlefish © Maxis Entertainment, distributed as Spore promotional material