

“Interactive games have the potential to become much more expansive than even the most epic film. If games are to reach this potential then humans can no longer continue to function as the predominant content creators. [1]” Presently, most video game content is created with hundreds of thousands of man hours by artists painstakingly creating most interactions and every texture, model, animation, and sound. The product, while truly a work of art, is very brittle and usually offers very little replay. Moreover, the product can take years to develop. There is, however, an alternative to the traditional approach of content¹ and asset² creation. Procedural synthesis yields a very rigid product with potentially infinite replay value which can be developed within a fraction of the time frame allocated to a traditionally developed game. Further, utilizing procedural synthesis for video game content and asset creation allows for the final product to require generally less storage space and scale with hardware improvements. Additionally procedural synthesizing content and assets creates a game that provides unique experiences and real, dynamic interactions.

Today, many AAA³ games start off like movies. There is a script, concept art, story boarding, etc. Before anything manifests itself in the virtual game world, artists paint and draw their vision for the game. Then these drawings and paintings are laboriously turned into their virtual counterparts using many different 2D and 3D art applications. The output of these applications is then processed by tools specialized for the particular game engine used by the developer. Finally, the game engine can use the newly created asset and render it in the virtual world.

Every time the game engine changes-- be it a customization, overhaul, or complete game engine change-- some, most, or all of the assets need to be re-created as well. Procedural synthesis, sometimes

1 Content is story, unique interactions, etc.

2 An asset is a model, sound, texture, or animation which can be re-used throughout the game.

3 The term AAA is not formally defined for the field. However, it refers to top quality interactive titles which often are highly anticipated by the community.

referred to as procedural generation, methods offer a much more agile¹, rigid², and potentially faster approach to game development. Procedural synthesis methods generate content on demand utilizing algorithms, executing at run-time, rather than manually creating them by a human artist. This allows for quick migration between different game engines. In addition, these procedural methods have the potential to allow the game assets to graphically scale with improvements to hardware with just code modification.

While artists generally command a lower salary than programmers or engineers, the amount of artists on a development team can sometimes dwarf the size of the programming team. If the majority of a game's assets are created manually by artists the costs soon begin to sky rocket. Additional levels, details, music, textures, etc. require either a greater number of artists [2] or longer development times. Average development costs for current-generation³ AAA titles are closing in on the production costs for modern blockbuster movies. For example, *Grand Theft Auto IV* cost \$100m to make while *The Dark Knight* had a budget of \$180m [17-18]. With games becoming larger and boasting longer game play development costs have the potential to surpass movie production costs and many studios may not be able to afford to compete with larger, big-name studios. However, the cost of utilizing procedural content can be much lower, especially so on larger content. Moreover, the results have the potential to be superior in quality to the results of the artist using traditional tools.

For example, take a very re-usable asset like a tree or fire-hydrant. Walking around a forest many trees share similarities, but they also have glaring differences. The same holds for fire-hydrants in the city. However, in most games today an artist will create one or a handful of trees or fire-hydrants and quickly the player will begin to see the same object over and over again [3]. Today's gamers demand an immersive experience. In real life every tree and every fire-hydrant, while looking similar,

-
- 1 Agility in software development refers to quick development cycles and promotes development iterations, open collaboration, and process adaptability
 - 2 Rigidness refers to the code's ability to withstand changes to other parts of the code without causing any errors.
 - 3 At the time of writing, current generation titles are those developed for Playstation 3, Wii, and Xbox 360.

exhibit stark differences. A traditionally created game can break immersion if too many objects which should be different look exactly the same. On the other hand, a game with procedurally synthesized assets can have these objects look different every time.

The assets can look different every time, but it is not necessary that they do. For example, *.kkrieger*, a showcase for procedural synthesis in games, looks virtually the same every time it is played. The game is entirely procedural and takes up a mere 96kB of space on a storage device while procedurally generating hundreds of megabytes of data on the fly and in real-time [4]. The entire game essentially uses less storage space than a JPEG¹ compressed screenshot of the game. This is made possible utilizing *.werkzeug3*, a procedural content creation and management tool, to create all the textures, materials, levels, characters, animations, and meshes [5]. *.kkrieger* proves video games can be entirely procedurally generated without losing the artists touch. According to Introversion Software, “procedural content generation [is not new] – in fact, most of the early video games used procedural generation as it was the only way to fit the vast amounts of data onto the relatively small mediums available at the time. [6]” So why are procedural generation methods not more widespread in the industry?

Perhaps the best response is the need for procedural synthesis methods is not readily apparent. After some further analysis the difficulty of shifting industry norms materializes. It is simply much easier to follow current standards than to do something quasi-radically different. Moreover, it is a lot more difficult to procure algorithms to generate textures, models, animations, and sound. Additionally developing new software, intended for use by artists and designers, in order to create these algorithms can be a colossal undertaking. However, utilizing procedural synthesis can create a significantly better product and developers are left no choice but to use procedural synthesis.

Games like *Mario Bros.*, *The Legend of Zelda*, *Elite*, and *Frontier* made heavy use of procedural

¹ JPEG, developed by Joint Photographic Experts Group, is a commonly used method of compression for photographic images.

synthesis in order to fit their rather expansive games on the limited distribution mediums available.

While space limitations may not be much of a concern today, Natural Motion affirms the modern gaming audience demands “[true] next-generation experiences to justify the cash outlay for hardware and software. [7]” These next-generation experiences contain unique experiences that are “not pre-produced, scripted, or necessarily planned. [7]” Essentially, next-generation content needs to provide the player with experiences generated utilizing procedural synthesis.

Many modern games already use some procedural synthesis: *Diablo* and *Diablo II* utilize procedural synthesis for level and dungeon generation [8]; *Grand Theft Auto IV* utilizes Natural Motion's *euphoria* system for procedural character animation and interaction; *Darwinia* creates its maps with procedural synthesis [6]; *Oblivion* utilizes procedural generation for tree and forest generation; *Spore* moderately utilizes procedural generation for the creation of models, textures, environments, and animations[6]; and plethora of games utilize procedural synthesis for motion and animation with regards to physics. Even some hardware, like the Xbox 360, utilizes procedural generation techniques [3]. The pattern is pretty clear: game developers have resorted to pigeon hole procedural synthesis to throw-away¹ content and environment, map, or terrain generation.

The reasoning for using procedural synthesis exclusively for throw-away content or terrain generation is eluding; especially when procedural synthesis can be utilized to generate virtually everything in the game world. Cities, containing buildings with navigable floor-plans, roads with deformations like pot holes and erosion, etc., can be generated on demand and in real-time [9-11]. The traffic on these roads can be redirected dynamically when obstacles or other restrictions arise [11]. The city's historic brown-stone buildings can be generated with realistic looking vine growing on them [13]. Even a realistic sky, day or night, can be generated on-the-fly and in real-time [12]. Outside of the city can be a realistic mountain range complete with rivers [14]. Rivers can even be added to random terrain

¹ Throw-away content or assets is something the player sees only once or twice throughout the entire game.

or throughout the city at the whim of the developer [15]. All of this is possible on-the-fly and mostly in real-time.

Utilizing procedural synthesis, as opposed to traditional techniques, more complex scenes can be created with the same amount of raw hardware power. For example, the Xbox 360 hardware allows for procedural generations of models and textures as well as deformations and interactions of those models and textures from simple meta data. This method of generation and deformation allows for less data to be sent across the bus¹ between the CPU², GPU³, and memory [3]. Since there is less data sent across the bus, more bandwidth is available for other assets allowing much more complex scenes. Using the ubiquitous forest example: a forest created with traditional methods will be rendered with a hundred or so trees, while a procedurally synthesized forest can be rendered with hundreds or thousands of trees utilizing the same amount of bus bandwidth. Procedurally synthesis techniques allow for much more complex, realistic, and visually appealing scenery while using generally the same hardware as traditional techniques. This is especially possible with architecture improvements like CUDA⁴ on NVidia cards allowing generally arbitrary software to run directly on the GPU instead of on the CPU.

Procedural synthesis is not without downsides. Procedurally generated content can be predictable, lack uniqueness, and games using this content may take much longer to load than traditionally created games [6]. However, utilizing tools like *.werkzeug3*, the tools created by Natural Motion for its animation engine *euphoria*, and the tools used by M.Ashikhmin in his project for synthesizing natural textures; predictability and lack of uniqueness can easily be averted. Moreover, with processors becoming more and more powerful by the quarter and multi-core CPUs quickly

1 A subsystem that transfers data between computer components inside a computer or between computers.

2 The Central Processing Unit provides virtually all the processing for any computer.

3 The Graphics Processing Unit takes care of rendering 2-D and 3-D images instead of the CPU.

4 Compute Unified Device Architecture is a C compiler and set of development tools that allow programmers to use the C programming language to code algorithms for execution on the GPU.

becoming the norm both in consoles and PCs long load times and gameplay smoothness interruptions can be minimized as well. Introversion Software argues that procedurally generated environments can provide gameplay difficulties. Some may simply not be fun at all so utilizing more traditional approaches may be necessary [6]. For instance, a naïve algorithm for procedurally generating a map or level may create similar spaces on multiple levels, place obstacles in places which do not make any sense to a designer or player, or even generate unreachable places. However, *.kkrieger* seems to disprove the notion that procedurally synthesized game spaces cannot be fun as the game space in it is very similar to any other FPS on the market (alluding that the game space is just as fun to play in as any other FPS game space), but entirely procedurally generated. Procedural synthesis does not have to be limited to game spaces, textures, models, and animations; virtually everything can be procedurally synthesized, including sound.

While procedural synthesis can be used for generating music and sound, generating realistic sound and composer quality music seems to be currently unattainable. Many games today use procedurally generated sound and music like *Spore*, *.kkrieger*, *Asheron's Call*, and *Dungeons and Dragons: Online* [20]. However, the music pales in comparison to the soundtracks of *World of Warcraft*, *Halo*, *Diablo*, and many other games whose music is composed and orchestrated. For games such as *Halo 3* studios have sound engineers on staff to create sound effects or record real-life sound effects for their games [16]. Such sound quality has yet to be achieved with procedural generation effects. Further, realistic and fluid sounding speech has yet to be achieved by even the top researchers. This is clearly seen by using any modern text-to-speech software. Until software algorithms are developed which can accurately generate sound and music on par with the quality of their human counterparts, it is, perhaps, better to continue allowing humans to be the primary creators of this type of content.

Despite all the drawback to utilize procedural synthesis methods as primary content and game

asset creators, it is still very feasible. procedural synthesis methods can be used to generate immense and unique worlds, characters, sounds, and animations. Further, the utilization of these methods decrease the overall storage footprint of games and allow for scaling of games with newer hardware by only code modification instead of code and asset modification. Moreover, procedurally generating content and assets allows for faster and more agile development cycles as well as much more rigid software while allowing for lower development costs. Not only are procedural synthesis methods for generating assets and content feasible, their heavy use is almost inevitable.

Bibliography

- [1] T. Roden, I. Parberry, "From Artistry to Automation: A Structured Methodology for Procedural Content Creation," in *Lecture Notes in Computer Science: Entertainment Computing – ICEC 2004*, Vol. 3166/2004. Springer, 2004, pp. 151-156.
- [2] Danc, "Content is Bad," *Lost Garden*, Feb. 10, 2007. [Online]. Available: <http://lostgarden.com/2007/02/content-is-bad.html>. [Accessed: Jul. 16, 2008].
- [3] J. Stokes, "Inside the Xbox 360, part I: procedural synthesis and dynamic worlds," *Ars Technica*, May 24, 2005. [Online]. Available: <http://arstechnica.com/articles/paedia/cpu/xbox360-1.ars/2>. [Accessed: Jul. 25, 2008].
- [4] ".kkrieger faq," *theprodukt.com*. [Online]. Available: <http://www.theprodukt.com/faq>. [Accessed: Jul. 18, 2008]
- [5] ".thprodukt," *theprodukt.com*, sec. technology. [Online]. Available: <http://www.theprodukt.com/theprodukt#10>. [Accessed: Jul. 21, 2008]
- [6] Introversion Software, "Procedural Content Generation," *gamecareerguide.com*, Jun. 02, 2007. [Online]. Available: http://www.gamecareerguide.com/features/336/procedural_content_.php?Page=2. [Accessed: Jul. 19, 2008]
- [7] Natural Motion Ltd. "Unique Game Moments," *Natural Motion Ltd.*, Mar. 2006 [Online] Available: http://www.naturalmotion.com/files/white_paper_ugm.pdf. [Accessed : Jul. 19, 2008]
- [8] M. Nitsche, M. Ashmore, C. Hankinson, W. Fitzpatrick, R. Kelly, K. Margenau. "Designing Procedural Game Spaces: A Case Study." *Proceedings of FuturePlay2006*. Oct 2006. [Online] Available: http://www.lcc.gatech.edu/~nitsche/download/Nitsche_DesigningProcedural_06.pdf [Access: Jul 19, 2008]
- [9] S. Grueter, J. Parker, N. Stewart, and G. Leach. "Real-time Procedural Generation of 'Pseudo Infinite' Cities." *Computer Graphics and Interactive Techniques in Australasia and South East Asia 2003*. pp. 87-94. [Online] Available: <http://0-delivery.acm.org.ilsprod.lib.neu.10.1145/610000/604490/p87-greuter.pdf?key1=604490&key2=5089526121&coll=portal&dl=ACM&CFID=78224579&CFTOKEN=91174020>. [Accessed: Jul 19, 2008]
- [10] A. Martinet, E. Galin, B. Desbenoit, and S. Akkouche. "Procedural Modeling of Cracks and Fractures." *Shape Modeling Applications, 2004. Proceedings. 2004*. pp. 346-349. [Online] Available: <http://0-ieeeexplore.ieee.org.ilsprod.lib.neu.edu/iel5/9186/29137/01314524.pdf>

- tp=&arnumber=1314524&isnumber=29137. [Accessed: Jul 19, 2008]
- [11] J. Sung, X. Yu, G. Baciuc, and M. Green. "Template-Based Generation of Road Networks for Virtual City Modeling." *Virtual Reality Software and Technology*. 2002. pp.33-40. [Online] Available: <http://0-delivery.acm.org.ilsprod.lib.neu.edu/10.1145/590000/585747/p33-sun.pdf?key1=585747&key2=6569526121&coll=Portal&dl=GUIDE&CFID=78255082&CFTOKEN=26695671>. [Accessed: Jul. 19 2008]
- [12] T. Roden and I. Parberry. "Clouds and Stars: Efficient Real-Time Procedural Sky Rendering Using 3D Hardware." *ACM International Conference Proceeding Series*. Vol. 265. 2005 pp. 434-437. [Online]. Available: <http://0-delivery.acm.org.ilsprod.lib.neu.edu/10.1145/1180000/1178574/p434-roden.pdf?key1=1178574&key2=2069526121&coll=portal&dl=ACM&CFID=78254649&CFTOKEN=99824709>. [Accessed: Jul. 19 2008]
- [13] M. Ashikhmin. "Synthesizing Natural Textures." *Symposium on Interactive 3D Graphics*. [Online]. Available: <http://0-delivery.acm.org.ilsprod.lib.neu.edu/10.1145/370000/364405/p217-ashikhmin.pdf?key1=364405&key2=9989526121&coll=portal&dl=ACM&CFID=78224579&CFTOKEN=91174020>. [Accessed: Jul. 19 2008]
- [14] P. Prusinkiewicz and M. Hammel. "A Fractal Model of Mountains with Rivers." *Proceeding of Graphics Interface '93*, pp. 174-180, May 1993. [Online]. Available: <http://algorithmicbotany.org/papers/mountains.gi93.pdf>. [Accessed: Jul. 26 2008]
- [15] D. Clyde. "Adding Realistic Rivers to Random Terrain." *gamedev.net*. Mar. 15 2004. [Online]. Available: <http://www.gamedev.net/reference/articles/article2065.asp>. [Accessed: Jul. 26 2008]
- [16] *Making of Halo 3*. [Videorecording]. Bungie, Inc. 2007.
- [17] D. Halbfinger. "A Director Confronts Darkness and Death", *The New York Times*, nytimes.com. Mar. 09 2008. [Online] Available: <http://www.nytimes.com/2008/03/09/movies/09halb.html>. [Accessed: Jul. 29 2008]
- [18] C. Berardini. "Grand Theft Auto IV Cost Around \$100 Million to Make," *Team Xbox*, [teamxbox.com](http://news.teamxbox.com/xbox/16384/Grand-Theft-Auto-IV-Cost-Around-100-Million-to-Make/). Apr. 30 2008. [Online] Available: <http://news.teamxbox.com/xbox/16384/Grand-Theft-Auto-IV-Cost-Around-100-Million-to-Make/>. [Accessed: Jul 29 2008]
- [19] *Behind the Scense of Unreal Tournament 3*. [Videorecording]. Epic Games, Inc. 2007.
- [20] Jason Booth. *Game Design and Development*. [Videorecording]. Northeastern University Association of Computing Machinery. 2006