

Geração Procedural de Conteúdo por Busca Inovativa em Nichos

Alexandre Santos Melotti*

Carlos Henrique Valério de Moraes

Universidade Federal de Itajubá, IESTI, Brasil

RESUMO

Com a crescente demanda de ambientes virtuais diferenciados surge a necessidade da geração dinâmica de conteúdo, agilizando o desenvolvimento, reduzindo custos de produção e ampliando a vida útil do produto. Em termos gerais, mundos procedurais geram uma gama maior de detalhamento no ambiente e personalizam a experiência, sem prejudicar a performance do sistema, mas isso só é garantido quando a qualidade do conteúdo obtido satisfaz o usuário em qualidade e diversidade. Desta forma, é necessária a utilização de algoritmos que sejam capazes de garantir a qualidade da experiência gerada sem faltar com a diversidade. Este trabalho apresenta uma técnica que pode ser utilizada no desenvolvimento de ambientes virtuais tanto em sua produção, oferecendo ao desenvolvedor possibilidades distintas de design para serem elaboradas e refinadas, como também durante a execução, construindo simulações únicas enquanto sendo utilizada pelo usuário. Assim, a técnica desenvolvida utiliza nichos morfológicos para garantir diversidade de soluções, produzindo resultados que apresentam as melhores soluções dentro de cada grupo, superando em qualidade as técnicas existentes como a busca inovativa e competição local.

Palavras-chave: Computação evolutiva, jogos, algoritmos genéticos, busca inovativa, geração procedural de conteúdo, geração procedural de fases.

1 INTRODUÇÃO

Com o crescimento do mercado de jogos também aumenta a demanda por conteúdos novos, cada vez mais sofisticados, diferenciados e muitas vezes personalizados. Infelizmente o custo e escalabilidade para criação manual de novos conteúdos variados, é alta e não escalável. Com isso, muita atenção tem sido dada para técnicas de Geração Procedural de Conteúdo (PCG), tanto no meio comercial quanto acadêmico. PCG é o termo geral para o processo onde métodos algorítmicos são utilizados para geração de conteúdo, seja em jogos ou outros contextos [1]. O conteúdo gerado pode incluir, mas não se limita a, terrenos, mapas, itens, recompensas, posicionamento de inimigos, distribuição de recursos, áudio, árvores de diálogo, objetivos de missões, padrões de comportamentos, e muitos outros. Estudos recentes, como [5], [12] e [15], avaliam e relacionam diversas metodologias e áreas onde a PCG pode ser aplicada em jogos.

2 O PROBLEMA

Como técnicas de *online* PCG são integradas e aplicadas durante a execução dos programas, alguns requisitos se fazem necessários na maioria dos casos: o algoritmo precisa ser executado de forma rápida, necessita ter um tempo de execução previsível e seus resultados devem ter qualidade esperada. Os dois primeiros requisitos, ambos relacionados ao tempo de execução do algoritmo, se devem ao fato de que o uso da *online* PCG ocorrer, muitas vezes, durante

*e-mail: contato@lexmelotti.com

a ação do jogo. Em jogos de que exigem reflexos rápidos e precisos, gargalos de processamento são inaceitáveis, exigindo que tais técnicas sejam processadas de forma rápida e eficiente.

Quanto a qualidade do conteúdo gerado, o problema surge quando alguns itens são criados de forma incompatível ou defeituosa. Enquanto alguns jogos como *Spelunky* [16] se beneficiam pela geração de conteúdos incrivelmente difíceis, a alta dificuldade de solução nestes casos é constante e faz parte de seu *design*. Em contraste, conteúdos mal formados podem influenciar a dinâmica do jogo de forma muito negativa. Se algum desafio gerado se torna impossível ou há um grande declive entre os níveis gerados, o equilíbrio entre desafio e tédio é rompido e o jogo deixa de ser divertido.

Analisando estes problemas, seria interessante um método que consiga gerar, em uma só execução, conteúdo bom e diversificado, garantindo pelo menos um mínimo de qualidade funcional e diferenciação entre si. Todas estas qualidades em um só método poderiam não só evitar frustração e tédio ao jogador, como poupar tempo com execuções desnecessárias gerando conteúdos semelhantes e não funcionais.

3 BACKGROUND

Esta seção apresenta metodologias relacionadas à busca por inovação que inspiraram e permitiram a principal contribuição deste trabalho (Niche Search), fazendo uma breve análise em relação ao potencial de suas aplicações no problema proposto.

3.1 Novelty Search

Algoritmos evolutivos são algoritmos de otimização que convergem soluções em direção à uma função de avaliação global. Em contraste, a Busca Inovativa (NS) [7] [9] é uma técnica de evolução divergente baseada na característica de diversificação da evolução natural, a qual é capaz de criar e manter várias espécies que dão suporte à vida. Desta forma a NS recompensa comportamentos inovadores ao invés do progresso em direção à um objetivo fixo, explorando eficientemente o espaço de busca e eventualmente encontrando indivíduos que solucionem o problema, mesmo sem promover um gradiente rumo à nota de avaliação.

3.2 Minimal Criteria Novelty Search

A Busca Inovativa com Critério Mínimo (MCNS) [10] é uma extensão da NS que leva em consideração outra característica importante da evolução natural: a função crítica para a evolução de qualquer organismo é a habilidade de sobreviver até ser capaz de se reproduzir.

3.3 Progressive Minimal Criteria Novelty Search

A Busca Inovativa com Critério Mínimo Progresso (PMCNS) [4] é uma extensão da MCNS que visa aproveitar as restrições do espaço de busca comportamental gerado pela técnica, mas sem a necessidade de redefinir um critério mínimo dependente de domínio.

3.4 Diversificação Morfológica

Algoritmos evolutivos convergem toda a sua população em direção à função de avaliação global, normalmente isso implica em uma população final com indivíduos extremamente semelhantes em algumas de suas características fenotípicas. Esse comportamento ocorre por consequência da natureza do algoritmo evolutivo, que recompensa fortemente nichos onde a nota da função de avaliação é boa. As estratégias a seguir, introduzidas em [10], combatem este comportamento de convergência fenotípica e evoluem uma diversidade de criaturas virtuais com somente uma busca.

3.5 Balanceando Funcionalidade e Inovação

A estratégia utiliza as características de exploração por inovação da NS para buscar novos nichos de indivíduos, definindo uma métrica para quantificar a diferença morfológica entre novos indivíduos e indivíduos anteriores.

Com uma métrica de distância puramente morfológica, não é garantido que os indivíduos evoluídos somente com a NS sejam funcionais, ou seja, recompensar somente diferenciação morfológica não visa a melhoria em prol da função de avaliação, garantindo somente a diversidade das características avaliadas em questão e nada mais. Desta forma, uma extensão ao método de NS é proposta, visando contornar a falta de busca objetiva e balanceando duas frentes concorrentes (inovação e funcionalidade) com a utilização de um Algoritmo Evolutivo Multiobjetivo (MOEA) baseado em eficiência à Pareto. Esta estratégia pode ser vista como uma busca objetiva que encoraja inovação a fim de manter uma diversidade morfológica na população.

Entretanto, a simples combinação de uma função de avaliação global e inovação morfológica em um MOEA não dá suporte ao fato de que: diferentes nichos suportam diferentes níveis de nota de avaliação. Indivíduos em um MOEA baseado em eficiência à Pareto são recompensados com base nas frentes não dominadas. Com isso, um indivíduo que tenha uma excelente nota de avaliação dentro de seu nicho, mesmo com a nota de avaliação medíocre em relação ao total da população, não será recompensado da forma como deveria.

3.6 Competição Local

A solução proposta para solucionar o problema de diferentes capacidades dos nichos é limitar a competição entre indivíduos localmente em relação a seus nichos, ou seja, indivíduos competem entre si somente com outros indivíduos morfológicamente próximos. A ideia é explorar a capacidade dentro de cada nicho ao invés de explorar gananciosamente somente os melhores nichos.

A principal mudança é que a competição local dentro de um nicho altera a nota de avaliação, que passa de uma medida global para uma medida relativa somente aos vizinhos próximos em seu nicho. Desta forma, o gradiente da busca passa de um balanço entre diversidade e performance global. Assim beneficia somente alguns nichos, para um balanço entre diversidade e performance dentro de um nicho local.

O valor de competição local é combinado com o valor de dispersão em um MOEA. Esta nova estratégia pode ser vista como uma busca objetiva limitada a nichos com encorajamento a inovação em busca de novos nichos. Graças às características da eficiência à Pareto indivíduos com boa nota de avaliação dentro de seus nichos tendem a ser mantidos junto com indivíduos com altas notas de inovação, motivando assim a exploração e eventualmente encontrando novos nichos a serem otimizados.

4 NICHE SEARCH

A seguir serão abordados os conceitos básicos utilizados para o desenvolvimento da Niche Search, cuja finalidade é dividir a população em nichos distintos e buscar os melhores indivíduos em cada um deles, explorando bem o espaço de busca e evitando a

concentração em somente um máximo ou mínimo local. A estratégia apresentada a seguir tenta resolver o problema evoluindo somente uma população, sem a necessidade de nenhum controle direto sobre a separação dos nichos ou evolução do problema em questão a parte das funções de avaliação e distância.

A combinação da NS com a estratégia de competição local em um MOEA com eficiência à Pareto, como apresentada em [10], provou ser eficiente na busca de diferentes morfologias em um problema de evolução de criaturas virtuais. Nesta estratégia entretanto, não há nenhuma consideração real em relação à nota de avaliação do indivíduo, sendo utilizada somente a sua nota de competição local e seu valor de dispersão para seleção. Isso implica que nichos que possuem baixas capacidades de performance são mantidos durante a busca, competindo diretamente com indivíduos em nichos com capacidades superiores. No contexto de exploração de nichos esta estratégia é boa, pois mantém os melhores indivíduos de todos os possíveis nichos encontrados; em um contexto onde deseja-se encontrar somente os melhores nichos, contudo, estes indivíduos apenas ocupam espaço na população durante a busca, sem contribuir diretamente para a busca por novos nichos.

4.1 Critério Mínimo Global

A proposta de melhoria oferecida neste trabalho é, portanto, eliminar nichos que possuam uma capacidade funcional muito baixa, evitando que indivíduos destes nichos sejam mantidos por suas notas de competição local. Enquanto possa parecer trivial apenas adicionar a nota de avaliação global como um terceiro objetivo no MOEA, as implicações dessa alteração certamente influenciariam o gradiente de busca, afetando desta forma a procura por novos nichos. O objetivo da melhoria é somente eliminar nichos com notas de avaliação muito baixas, sem influenciar a forma como a busca se espalha pelo espaço das morfologias.

Para endereçar este problema de nichos com baixa capacidade funcional, considerou-se a premissa já elaborada em [8] para a MCNS, de que a função crítica para a evolução de qualquer organismo é a habilidade de sobreviver até ser capaz de se reproduzir. Desta forma, esta pesquisa sugere a utilização de um critério mínimo relacionado à função de avaliação global durante o processo de seleção.

Para a introdução deste critério mínimo a implementação escolhida como base foi a PMCNS, pois não necessita da definição de um parâmetro para critério mínimo dependente de domínio e por ser capaz de simular a NS e a MCNS com simples alterações em seus parâmetros de controle. Porém, na implementação aqui proposta, a PMCNS utiliza o critério mínimo de forma diferente, atribuindo zero para a nota de competição local de indivíduos que não atingem o critério mínimo em relação a nota de avaliação global, nunca alterando o valor de inovação calculado. Com isto, devido às características do MOEA, a busca continua a explorar os diversos espaços de morfologias, selecionando e mantendo em sua população somente indivíduos de nichos que atinjam o critério mínimo.

4.2 Estratégia de Seleção

Como MOEA para seleção, o algoritmo escolhido para esta pesquisa foi o NSGA-II [3]. Além de rápido e eficiente este também foi o algoritmo utilizado em [10], onde teve seu *operador de distância de aglomeração* alterado em prol da diversidade morfológica. A escolha deste algoritmo não só permite uma melhor comparação com os estudos realizados, como também possui este *operador de distância de aglomeração* que pode ser utilizado em favor da busca por diversidade.

4.3 Conjunto de Soluções

Com o objetivo de fornecer diversos indivíduos diferenciados e de qualidade como solução, o conjunto de soluções final C_f é definido

como o conjunto de indivíduos que possuem nota local máxima ao final das execuções. O conjunto C_f representa os indivíduos com maior capacidade funcional dentro de seus próprios nichos, garantindo assim uma boa diversidade entre os resultados, assim como qualidade superior em relação aos indivíduos em sua proximidade.

5 EVOLUINDO MAPAS PARA UM JOGO ROGUELIKE

Para implementação e testes da técnica proposta, o problema de geração de mapas diferenciados para um jogo *roguelike* foi o escolhido por ser uma das grandes áreas de sucesso de aplicação de PCG [2] [14] [13]. Ao gerar novos mapas, buscou-se criar opções que sejam ao mesmo tempo inovadores e interessantes; assim sendo, a estratégia que busca os melhores indivíduos em nichos diferentes se torna promissora para a solução deste tipo de problema.

5.1 Função de Avaliação

A função de avaliação é montada de forma a incentivar a exploração e aventura do jogador, e por este motivo as posições de entrada e saída são de extrema importância. Os mapas são examinados utilizando uma variação de uma das funções de avaliação proposta em [2] e revisada em [14], relativa ao comprimento do menor caminho entre a entrada e saída; quanto mais comprido for o menor caminho existente entre estes, maior o esforço que o jogador precisará realizar a fim de passar para a fase seguinte, contribuindo assim para o fator de exploração e aventura do mapa e fazendo com que ele se torne mais interessante.

5.2 Função Distância

Outra função importante a ser definida é a função de distância, esta função é fortemente relacionada ao contexto do problema, e normalmente é utilizada na NS para encontrar indivíduos com comportamentos diferenciados.

O cálculo de distância fenotípica é a comparação direta entre o fenótipo dos mapas em questão, ou seja, compara-se diretamente cada uma das 4096 variáveis do fenótipo de dois mapas e tem-se como distância final o número de comparações diferentes entre eles. Desta forma, quando mapas muito parecidos são comparados, um baixo valor de distância é obtido, uma vez que muitas de suas variáveis coincidem; de forma oposta, quando mapas com áreas abertas em locais diferentes são comparados o valor de distância calculado é grande.

6 EXPERIMENTOS E RESULTADOS

Os experimentos realizados nesta pesquisa evoluem mapas para um jogo *roguelike* utilizando os métodos previamente elaborados nos capítulos anteriores. O objetivo dos experimentos é testar e comprovar a eficiência do método proposto em obter mapas diversificados e com boas notas de avaliação em comparação à outras técnicas.

Quatro configurações de execuções foram consideradas para os testes: **genetic algorithm (GA)**, **novelty search (NS)**, **local competition (LC)** e **niche search (Niche)**.

6.1 Métricas

As métricas utilizadas são a Nota Local, Quantidade de Indivíduos em C_f , Nota de Avaliação e Nota de Dispersão. Os valores foram obtidos com base na população final P_f após 200 gerações e em relação ao conjunto C_f de indivíduos que obtiveram a nota local máxima na última geração, como descrito na seção 4.3.

6.2 Resultados Obtidos

A seguir são apresentados e comparados os valores obtidos para cada uma das métricas em cada uma das configurações. Para cada uma das 4 configurações, os testes foram executados até que a variação da média acumulada para cada métrica testada em todas as execuções se mantivesse abaixo de 5% por duas execuções seguidas, um número mínimo de 5 execuções foi utilizado. Todas as

métricas foram calculadas com base na população final obtida, e o valor obtido para cada métrica é o resultado da média entre todas as execuções realizadas. Para as configurações **GA**, **NS**, **LC** e **Niche**, o número de execuções necessárias para a estabilização da média em todas as métricas foram respectivamente 21, 8, 5 e 7. A Tabela 1 contém os resultados obtidos em todas as métricas, para cada uma das configurações, ao final de suas execuções.

7 DISCUSSÃO DAS ANÁLISES E TESTES

Observando-se os resultados obtidos durante os testes é possível verificar a alta capacidade de diversificação introduzida pela NS, com as demais configurações baseadas em NS obtendo boas notas nas métrica de Nota Local, Quantidade de indivíduos em C_f e Dispersão. Mesmo a configuração baseada em AG tendo obtido os melhores resultados em relação a Nota de Avaliação, esta não se mostrou capaz de gerar resultados bons o suficiente para o objetivo desta pesquisa, resultando em indivíduos muito semelhantes em sua população final.

Os resultados analisados suportam a hipótese de que, no contexto de exploração de indivíduos inovadores, a introdução de um critério mínimo global para restrição da busca, em adição à estratégia de Competição Local apresentada por [10], contribui para a melhoria da qualidade funcional dos indivíduos obtidos. Mesmo havendo uma relação de troca em relação à Nota de Avaliação e Dispersão, esta representa um ganho de 18,80 e 58,09 % para Nota de Avaliação contra uma queda de apenas 3,09 e 3,56 % para Dispersão em relação as configurações **NS** e **LC** respectivamente, comprovando a eficiência da Niche Search em obter resultados melhores.

8 CONCLUSÃO

Este trabalho apresentou e implementou o Niche Search, um algoritmo evolutivo capaz de obter indivíduos diversificados e com alta capacidade funcional em uma única execução. O mesmo, com base em uma função de distância fenotípica, é capaz de manter diferentes nichos morfológicos durante a busca ao mesmo tempo que seleciona os melhores indivíduos dentro de cada um deles. Além disso, o algoritmo é eficaz em eliminar nichos com baixa capacidade funcional da busca, obtendo um conjunto final de indivíduos com melhor nota de avaliação quando comparado às técnicas nas quais se baseia.

Os resultados obtidos permitiram confirmar os benefícios da introdução da NS no contexto de busca por inovação, com todas as configurações baseadas na NS obtendo valores de dispersão muito superiores aos valores obtidos pela configuração baseada em GA, demonstrando que a introdução da mesma contribui significativamente para a obtenção de indivíduos diferenciados. Embora haja um grande ganho em diversificação, os testes revelam que a utilização da NS sem uma função de distância que se relacione de algum modo com a funcionalidade, não contribui com a busca por boas notas de avaliação, selecionando indivíduos somente com base na morfologia para a qual foi elaborada.

Os resultados mostram ainda, que as modificações realizadas no algoritmo apresentado em [10], que foram aqui introduzidas e dão suporte a ideia de exclusão de nichos de baixa capacidade, obtiveram bons resultados em relação à qualidade dos indivíduos encontrados. Quando comparados, os resultados obtidos pela configuração **LC** e **Niche**, revelam um ganho de 18,80% em relação à nota de avaliação média do conjunto C_f obtido pela segunda, ao mesmo tempo que mantém o valor de dispersão com uma variação de apenas -3,09%. Estes valores comprovam a eficiência da remoção de nichos ruins da busca para a melhoria da qualidade funcional obtida, dando prioridade a nichos com alta capacidade durante a seleção, sem que a diversidade introduzida pela NS seja prejudicada.

Tabela 1: Valores obtidos por cada configuração em cada uma das métricas.

Métricas	Algoritmo Genético	Novelty Search	Competição Local	Niche Search
Número de Execuções	21	8	5	7
Nota Local Média	0.3776	7.9075	9.6340	9.4607
Nota Local Máxima	9.0000	15.0000	15.0000	15.0000
Quantidade de indivíduos em C_f	0.4762	12.2500	22.6000	18.4286
Nota de Avaliação Média - P_f	243.4019	63.0031	102.2420	133.7679
Nota de Avaliação Média - C_f	N/A	100.7976	134.1249	159.3472
Dispersão Média - P_f	35.8831	1692.0211	1681.9223	1550.1303
Dispersão Média - C_f	N/A	2044.7265	2034.8144	1971.9663
Tempo Total em segundos	875	1634	1008	1401
Tempo Médio de Execução	41.6667	204.2500	201.6000	200.1429
Tempo Médio por Solução	87.4982	16.6735	8.9203	10.8604

Nos testes realizados, a configuração **Niche** foi capaz de obter, de forma fidedigna, uma quantidade significativa de mapas como solução em todas as execuções. Além disso, se comparados com os mapas obtidos pela AG, a configuração Proposta demonstra ser capaz de, em uma única execução, encontrar mapas visualmente e estruturalmente bem diversificados. Outra consequência disso, é que mesmo obtendo um tempo total de execução maior que a AG, o tempo médio por solução obtida pela proposta também se torna um avanço em relação a capacidade computacional da solução do problema proposto.

Com isso, as técnicas e algoritmos aqui propostos, evidenciam ser viáveis para o contexto de geração procedural de conteúdo onde se deseja evitar a baixa qualidade dos itens gerado, assim como repetição ou inconsistência nos padrões dos mesmos.

Fazem parte da contribuição gerada por este trabalho: o pacote computacional genérico da ferramenta proposta; assim como as extensões para utilização da NS, MOEAs e NSGA-II para a *framework* AForge.NET [6], com código fonte disponível online em [11] sob licença GNU LGPLv3.

9 TRABALHOS FUTUROS

Como sugestão de estudos de continuidade, propõe-se uma extensão do algoritmo apresentado que aprimore a busca dentro dos nichos obtidos, efetuando uma pesquisa por indivíduos com altas notas de avaliação limitada por barreiras, evitando assim a convergência global da busca e mantendo a diversidade obtida pelo algoritmo aqui apresentado. Para tal, sugere-se a utilização de um algoritmo genético com barreiras. Utilizando os indivíduos encontrados em C_f como base para o particionamento, e técnicas de agrupamento como *k-means* e *k-medoids*, a busca pode ser delimitada por regiões definidas pelas medianas dos centróides e medóides obtidos. Outra opção possível, uma vez que o algoritmo aqui apresentado já realiza uma segregação por nichos morfológicos, é limitar a busca ao redor dos indivíduos de C_f com o cálculos de medianas entre os mesmos.

Sugere-se ainda a utilização da estrutura de dados *QuadTrees* em trabalhos futuros para otimização do cálculo da busca por vizinhos mais próximos, assim como a possibilidade da utilização de outras métricas para o cálculo de distância entre mapas, podendo considerar a utilização de técnicas de similaridade de imagens, comparação entre os caminhos gerados, e outros cálculos utilizando o genótipo para simplificação e otimização.

REFERÊNCIAS

[1] M. C. Angelides and H. Agius. Procedural content generation. In *Handbook of Digital Games*, pages 62–91. Wiley-IEEE Press, 2014.

- [2] D. Ashlock, C. Lee, and C. McGuinness. Search-based procedural generation of maze-like levels. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):260–273, 2011.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [4] J. Gomes, P. Urbano, and A. L. Christensen. Progressive minimal criteria novelty search. In *Ibero-American Conference on Artificial Intelligence*, pages 281–290. Springer, 2012.
- [5] M. Hendriks, S. Meijer, J. Van Der Velden, and A. Iosup. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 9(1):1, 2013.
- [6] A. Kirillov. Aforge.net framework, 2013.
- [7] J. Lehman and K. O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *ALIFE*, pages 329–336, 2008.
- [8] J. Lehman and K. O. Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 103–110. ACM, 2010.
- [9] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223, 2011.
- [10] J. Lehman and K. O. Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218. ACM, 2011.
- [11] A. Melotti. Niche Search extension for AForge.net. <https://github.com/lexmelotti/nichesearch>, 2016. [Online; acessado 23-Setembro-2017].
- [12] J. Togelius, G. N. Yannakakis, K. O. Stanley, and C. Browne. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):172–186, 2011.
- [13] V. Valtchanov and J. A. Brown. Evolving dungeon crawler levels with relative placement. In *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering*, pages 27–35. ACM, 2012.
- [14] R. van der Linden, R. Lopes, and R. Bidarra. Procedural generation of dungeons. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1):78–89, 2014.
- [15] G. N. Yannakakis and J. Togelius. Experience-driven procedural content generation. *IEEE Transactions on Affective Computing*, 2(3):147–161, 2011.
- [16] D. Yu. Spelunky. Mossmouth, 2012.