

# GenTP – Uma Ferramenta Interactiva para a Geração Artificial de Terrenos

Miguel Frade<sup>1</sup>, F. Fernandez de Vega<sup>2</sup>, Carlos Cotta<sup>3</sup>

[mfrade@estg.ipleiria.pt](mailto:mfrade@estg.ipleiria.pt), [fcofdez@unex.es](mailto:fcofdez@unex.es), [ccottap@lcc.uma.es](mailto:ccottap@lcc.uma.es)

<sup>1</sup> ESTG – Instituto Politécnico de Leiria, Alto do Vieiro, Apartado 4163, 2411-901 Leiria, Portugal

<sup>2</sup> Centro Universitario de Merida, UNEX, C/Sta Teresa de Jornet 38, 06800 Mérida, Espanha

<sup>3</sup> ETSI Informática, Campus de Teatinos, Universidad de Málaga, 29071 - Málaga, Espanha

**Resumo:** Actualmente existe uma grande variedade de técnicas de geração artificial de terrenos. No entanto todas elas estão focadas em fornecer terrenos de aspecto realista, negligenciando por vezes a estética e a criatividade. Este artigo apresenta uma ferramenta interactiva para a geração artificial de terrenos, designada GenTP, inspirada nos sistemas de arte evolutiva com programação genética. Esta ferramenta permite aos *designers* evoluir TPs (*Terrain Programs*) de acordo com o seu sentido estético ou características desejadas. Os TPs, o produto desta ferramenta, geram terrenos diferentes a cada execução, mas sempre coerentes nas suas características morfológicas.

**Palavras-chave:** Geração de terrenos; *design* evolutivo; programação genética.

## 1. Introdução

As técnicas de geração artificial de terrenos são usadas em diversas áreas, como a educação, planeamento do território, colocação de transmissores de telecomunicações e análise de sinal, entretenimento, simulação e treino militar. No entanto, é nos vídeo jogos que o seu uso é mais comum.

Actualmente existem várias técnicas de geração artificial de terrenos, mas todas elas apresentam limitações. As técnicas mais flexíveis requerem muito tempo, esforço e perícia por parte dos *designers* para obter resultados aceitáveis. Outra desvantagem é a impossibilidade de serem usadas de forma automatizada para gerarem terrenos. Por sua vez os métodos mais simples permitem gerar pouca diversidade de terrenos e não possibilitam o controlo sobre as suas características localizadas, como por exemplo a presença de um lago ou de uma montanha. No entanto, todas as técnicas tradicionais de geração artificial de terrenos apresentam

uma característica em comum, estão focadas na geração de terrenos com aspecto real. Esta característica é importante, mas existem situações, em especial na indústria dos vídeo jogos, onde o sentido estético do *designer* pode ser mais relevante do que o realismo.

Com o intuito de colmatar as limitações das técnicas tradicionais (Frade, Vega, & Cotta, 2008) propuseram uma técnica evolutiva designada *Genetic Terrain Programming* (Programação Genética de Terrenos). Neste artigo é apresentada a ferramenta que foi desenvolvida para demonstrar a validade desta nova técnica de geração artificial de terrenos, o GenTP (*Generator of Terrain Programs*).

A Secção 2 apresenta as técnicas tradicionais e evolutivas de geração artificial de terrenos. De seguida, na Secção 3, expõem-se os detalhes de implementação do sistema evolutivo do GenTP, a sua interface gráfica e alguns exemplos de terrenos gerados a partir desta ferramenta. Por fim, na Secção 4, constam as conclusões e trabalho futuro.

## **2. Geração Artificial de Terrenos**

Existem diversos métodos para representar um terreno, mas um dos mais comuns é o mapa de elevações. Formalmente, um mapa de elevações é uma função escalar de duas variáveis a que corresponde uma altura  $h$  para cada par de coordenadas  $(x,y)$ . Os mapas de elevações não permitem representar estruturas com múltiplas alturas para o mesmo par de coordenadas, como por exemplo grutas. Esta limitação não é crítica para a maioria das utilizações e além disso esta representação permite elevados níveis de optimização para geração de imagens e detecção de colisões (Duchaineau et al., 1997) .

### **2.1. Técnicas Tradicionais**

As técnicas tradicionais de geração de terrenos podem ser divididas em três grupos: medição, modulação e procedimentais. As técnicas de medição consistem na recolha das elevações de um terreno real através de levantamentos topográficos. Estas técnicas têm a vantagem de produzir terrenos extremamente realistas, mas comprometem a possibilidade de controlo por parte do *designer*. Se este pretender obter um terreno com determinadas características poderá ter que efectuar pesquisas exaustivas para encontrar um mapa real que corresponda aos seus objectivos.

De todas as técnicas tradicionais a modulação é a mais flexível. Um *designer*, ou artista, esculpe a morfologia do terreno usando um programa de modelação 3D como por exemplo o 3D Studio<sup>1</sup> ou Blender<sup>2</sup>. Esta abordagem permite um controlo

<sup>1</sup> <http://www.autodesk.com/fo-products>

<sup>2</sup> <http://www.blender.org>

ilimitado sobre a topologia e características dos terrenos. No entanto, isto também pode ser uma desvantagem devido aos elevados requisitos que exige ao *designer* em termos de conhecimento sobre a ferramenta de modelação, esforço e tempo.

As técnicas procedimentais podem ser ainda subdivididas em: simuladores físicos, síntese espectral e fractais. Os simuladores físicos baseiam-se em modelos matemáticos dos diversos fenómenos associados à evolução dos terrenos como a erosão (Kelley, Malin, & Nielson, 1988), ou movimentos das placas tectónicas. Esta técnica gera terrenos muito semelhantes aos reais, mas requerem um conhecimento profundo sobre as leis físicas para ser implementada e usada.

A síntese espectral consiste no cálculo de valores aleatórios no domínio da frequência e depois convertê-los para alturas no domínio do espaço através da transformada inversa de Fourier. Os terrenos gerados a partir desta técnica são estatisticamente homogêneos e isotrópicos, características que os terrenos reais não possuem (Olsen, 2004). Além disso permite muito pouco controlo sobre o terreno resultante.

As técnicas baseadas em fractais exploram a auto-semelhança que os terrenos apresentam – até certo ponto (Peitgen, Jürgens, & Saupe, 2004). Os algoritmos de geração de terrenos baseados em fractais são bastante populares entre os programadores de vídeo jogos devido à sua simplicidade e velocidade. Existem diversas ferramentas baseadas em fractais como o Terragen<sup>3</sup> e o GenSurf<sup>4</sup>. No entanto, os terrenos gerados por esta técnica são facilmente reconhecidos devido ao padrão de auto-semelhança característico dos fractais. Além disso não é possível o controlo das características finais do terreno, embora existam alguns parâmetros que permitem especificar, por exemplo, a rugosidade do terreno.

## **2.2. Técnicas Evolutivas**

As técnicas evolutivas baseiam-se em algoritmos bio-inspirados que aplicam a teoria de (Darwin, 1859). Esta defende a evolução natural das espécies onde os organismos vivos são recompensados, através da sobrevivência e da propagação dos seus próprios genes aos sucessores. Actualmente existem quatro classes principais de algoritmos evolutivos: Algoritmos Genéticos (AG) (Holland, 1975), Estratégias Evolutivas (Bäck, Hoffmeister, & Schwefel, 1991), Programação Genética (GP) (Koza, 1992) e Programação Evolutiva (Fogel, Owens, & Walsh, 1966). Todos os algoritmos evolutivos mantêm uma população de soluções candidatas sobre a qual efectua uma pesquisa para determinar os indivíduos mais fracos. De acordo com um determinado critério, estes são substituídos por outros gerados através de operadores aplicados aos melhores indivíduos da população, criando assim uma nova geração. Este processo é repetido sobre sucessivas gerações até se encontrar uma boa solução, que pode não ser a óptima.

<sup>3</sup> <http://www.planetside.co.uk/terrigen>

<sup>4</sup> <http://tarot.telefragged.com/gensurf>

Os primeiros autores a proporem o uso de algoritmos evolutivos para a geração artificial de terrenos foram (Ong, Saunders, Keyser, & Leggett, 2005). Na sua abordagem, baseada em AG, a geração do terrenos processa-se em duas fases. Primeiro geram a silhueta do terreno onde podem ser definidas zonas de acordo com as características desejadas. A segunda fase está encarregue de gerar o mapa de elevações. A geração do mapa de elevações consiste na pesquisa de uma base de dados, com amostras de mapas reais pré-seleccionados, que permitam o melhor arranjo de acordo com as características expressas na primeira fase. Tal como as técnicas tradicionais, também esta está focada apenas na geração de terrenos com aspecto realista. Não permite a geração de terrenos “extra-terrestres” ou de acordo com o sentido estético do *designer*.

(Frade, Vega, & Cotta, 2008) propuseram a *Genetic Terrain Programming*, que consiste na combinação de sistemas de arte evolutiva com PG para gerar terrenos. Os sistemas de arte evolutiva têm muitas semelhanças entre si (Bentley, 1999): todos eles são baseados em algoritmos bio-inspirados; geram novas formas a partir de uma população inicial aleatória; dependem de um utilizador para avaliar os indivíduos da população; o tamanho das populações é pequeno para evitar a fadiga do utilizador e permitir avaliações rápidas; e normalmente as interfaces dispõem em forma de grelha os indivíduos a avaliar. A principal diferença destes sistemas reside no fenótipo – a forma como os genes são mapeados para o resultado final.

Os sistemas de arte evolutiva, onde os fenótipos mais comuns são as imagens, têm sido usados com sucesso, há alguns anos, por vários artistas. Nesta área a PG tem sido o algoritmo bio-inspirado mais profícuo para a evolução interactiva de imagens. Karl Sims usou PG para criar e evoluir imagens através de equações matemáticas. As equações são usadas para determinar o valor de cada pixel (Sims, 1991), ou criar filmes adicionando uma variável temporal a equações diferenciais dinâmicas (Sims, 1992). Ele criou várias obras de arte gráficas incluindo *Panspermia* e *Primordial Dance*. Permiteu aos visitantes de galerias de arte uma forma de criarem a sua própria arte gráfica através da interacção com o seu sistema designado *Galapagos*<sup>5</sup>. Desde então vários investigadores tem expandido o seu trabalho. Alguns dos trabalhos mais proeminentes nesta área são: (Unemi, 2000) que desenvolveu o SBART (*Simulated Breeding ART*), um sistema que permite realizar colagens de imagens e filmes; e o NEvAr (*Neuro Evolutionary Art*) de (Machado & Cardoso, 2000) que focou na reutilização dos melhores indivíduos, armazenados numa base de dados, para criar procedimentos automáticos de inseminação.

Na *Genetic Terrain Programming* os fenótipos são mapas de elevações, em vez de imagens. Esta técnica permite obter terrenos com aspecto realista (sem a necessidade de recorrer a base de dados com terrenos reais), ou de acordo com o sentido estético do *designer*. Esta ferramenta produz TPs (*Terrain Programs*) que, a cada execução, geram um terreno diferente. Essa diferença deve-se aos valores

<sup>5</sup> <http://www.genarts.com/galapagos>

aleatórios incorporadas no próprio TP. Apesar disso, os terrenos criados pelo mesmo TP apresentam sempre as mesmas características. Esta propriedade permite o controlo localizado das características de um terreno através da sua composição por diversos TPs, eliminando assim umas das principais desvantagens das técnicas procedimentais. Esta ferramenta permite ainda diminuir o tempo e o esforço dos *designers*, relativamente às técnicas de modelação, para criarem terrenos complexos e o resultado final não está dependente da sua perícia.

### 3. GenTP

O GenTP assenta na PG como algoritmo evolutivo, que se distingue dos outros pela forma como representa os indivíduos de uma população. Enquanto os AG representam um indivíduo como uma sequência de genes, a PG representa-os como árvores compostas por nós intermédios, que contêm funções, e nós finais que são designados por terminais. Cada indivíduo do GenTP é uma árvore composta por funções, descritas na Tabela 1, e pelos terminais apresentados na Tabela 2. Contrariamente aos sistemas de arte evolutiva revistos na Secção 2.2, no GenTP os terminais não são valores escalares, mas matrizes bidimensionais que representam mapas de elevações. A Figura 1 ilustra o exemplo de um indivíduo, que no contexto do GenTP, também é designado por TP.

Tabela 1 – Funções para os nós intermédios da árvore

Nome	Descrição
<i>plus(a,b)</i>	soma: $a + b$
<i>minus(a,b)</i>	subtração: $a - b$
<i>multiply(a,b)</i>	multiplicação: $a \times b$
<i>sin(a), cos(a), tan(a), atan(a)</i>	funções trigonométricas
<i>myLog(a)</i>	devolve $0$ , se $a=0$ e $\log( a )$ caso contrário
<i>myPower(a,b)</i>	devolve $0$ se $a^b$ for <i>NaN</i> , <i>Inf</i> , ou tiver parte imaginária, caso contrario devolve $a^b$
<i>myDivide(a,b)</i>	devolve $a$ se $b=0$ , senão devolve $a/b$
<i>myMod(a,b)</i>	é $0$ se $b=0$ , senão devolve o resto da divisão inteira $a/b$
<i>mySqrt(a)</i>	devolve a raiz quadrada de $ a $
<i>negative(a)</i>	devolve $-a$
<i>FFT(a)</i>	calcula a transformada discreta de Fourier 2-D
<i>smooth(a)</i>	aplica um filtro circular passa baixo de raio 5

Tabela 2 – Terminais da árvore

Nome	Descrição
<i>rand</i>	mapa com elevações aleatórias entre 0 e 1.
<i>fftGen</i>	mapa de elevações baseado na síntese espectral. O espectro depende de um valor aleatório.
<i>gauss</i>	mapa de elevações em forma de sino cujo largura depende de um valor aleatório.
<i>plane</i>	mapa de elevações que representa um plano inclinado, a direcção da inclinação tem 8 valores possíveis e é escolhido de forma aleatória.
<i>step</i>	mapa de elevações que representa um degrau, tanto a orientação como o centro do degrau estão dependentes de valores aleatórios.
<i>sphere</i>	mapa de elevações com uma semi-esfera cujos centro e raios dependem de valores aleatórios.

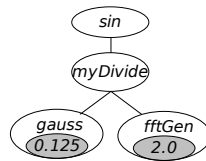


Figura 1 – Exemplo de um TP representado em árvore

Os terminais representam mapas de elevações que são sempre proporcionais às dimensões do próprio mapa. Isto permite-nos que um TP gere terrenos sempre com as mesmas características independente do tamanho escolhido. Além disso, todos os terminais dependem de valores aleatórios, pelo que sucessivas execuções do mesmo TP geram terrenos diferentes. Alguns desses valores são designados por Constantes Efémeras Aleatórias (CEF). As CEFs são determinadas aleatoriamente e mantêm-se constantes durante a vida de um determinado indivíduo. A Figura 1 apresenta um exemplo onde as CEF's estão representadas dentro de elipses cinzentas. Os cálculos para gerar os terrenos são efectuados sobre matrizes que representam, internamente, os mapas de elevações.

O GenTP foi programado em Matlab<sup>6</sup> com recurso ao GPLab<sup>7</sup>, uma *toolbox* de código aberto para PG. A população inicial é criada de forma aleatória com as árvores inicialmente limitadas a 20 nós de profundidade e uma população fixa de 12 indivíduos. O número de gerações é decidido pelo *designer*, que pode parar a aplicação a qualquer momento. As gerações seguintes à inicial são criadas com base num ou dois indivíduos, conforme o desejo do *designer*. Caso seja seleccionado

<sup>6</sup> <http://www.mathworks.com>

<sup>7</sup> <http://gplab.sourceforge.net>

apenas um indivíduo só o operador de mutação será usado. Se forem seleccionados 2 indivíduos serão aplicados os operadores mutação e de reprodução sexual (*crossover*) (Koza, 1992). Tal como acontece noutros sistemas de arte evolutiva, a avaliação dos indivíduos está exclusivamente dependente da escolha do *designer*. A sua decisão pode ser baseada no sentido estético ou nas características morfológicas pretendidas.

### 3.1. Interface Gráfica

Como pode ser observado na Figura 2, o GenTP apresenta uma interface gráfica com 12 indivíduos dispostos em grelha representados como superfícies 3D. A partir desta interface o *designer* pode seleccionar, de acordo com o seu critério, um ou dois TPs para criar a próxima geração. Se seleccionar um só TP a geração seguinte apresentará poucas diferenças em relação ao TP seleccionado. Se forem seleccionados dois TPs a geração seguinte será mais diversificada e com características de ambos os TPs.

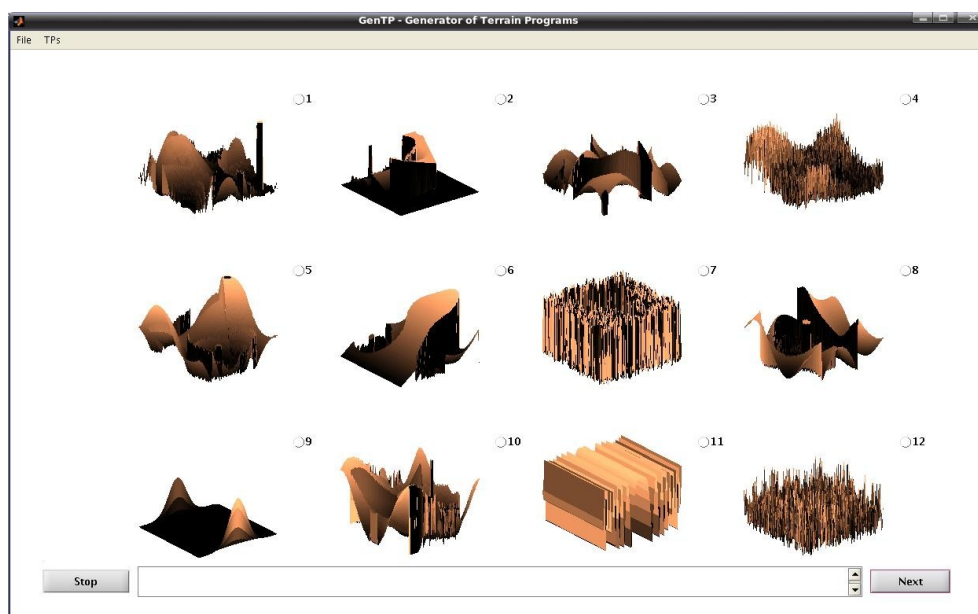


Figura 2 – Interface principal

Apesar desta interface servir o seu propósito, por vezes algumas características dos TPs ficam escondidas devido ao ângulo em que são apresentados. A inspecção de pequenos detalhes também não é possível, o que dificulta a escolha dos TPs para criar a geração seguinte. Além disso essa escolha teria de ser efectuada com base numa só execução do TP, não sendo possível averiguar o seu comportamento em

execuções sucessivas. Para resolver esta limitação foi adicionada uma função de análise. Esta função abre uma nova janela com 8 execuções consecutivas do TP seleccionado, conforme apresentado na Figura 3. Aí é possível verificar a persistência das características produzidas por um TP, rodá-lo sobre os 3 eixos, fazer *zoom* e seleccionar o esquema de cores para representar os terrenos. Desta forma o *designer* pode efectuar uma análise mais detalhada de um TP antes de tomar uma decisão.

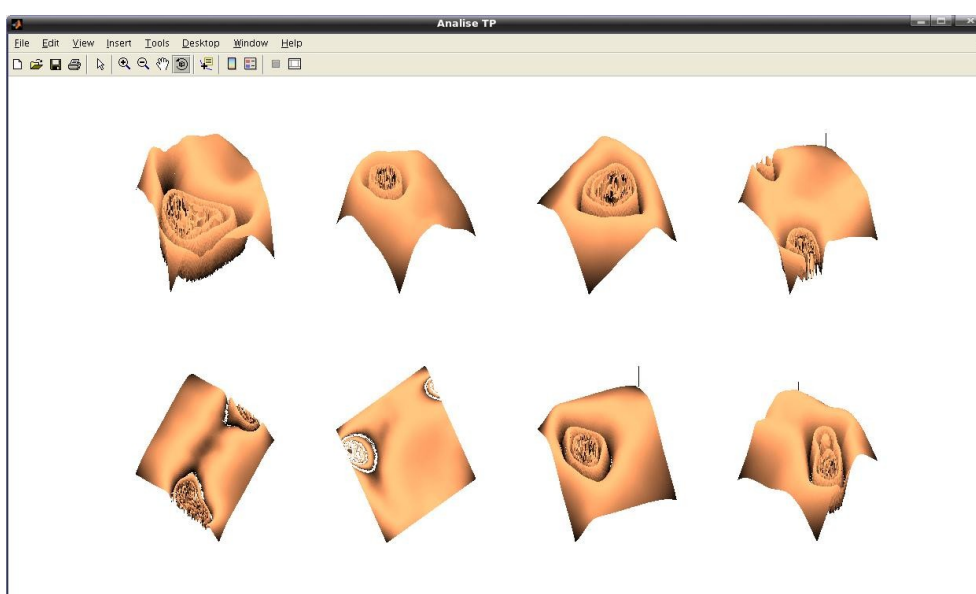


Figura 3 – Interface de análise

O GenTP também permite ver a equação correspondente ao TP seleccionado e gravá-lo num ficheiro de texto, ou numa base de dados. O mapa de elevações resultante da execução do TP escolhido pode ser gravado como um ficheiro VRML 2.0, o que permite a sua importação por outras aplicações (para processamento adicional), ou para um vídeo jogo. Na Figura 4 apresenta-se o esquema funcional do GenTP.

### 3.2. Exemplos de TPs

Nas Eq. 1 e 2 apresentam-se dois exemplos de TPs que foram seleccionados pelo apelo estético dos terrenos que produzem (Figuras 5 e 6). As Eq. 3 e 4 apresentam dois TPs que foram evoluídos com objectivos pré-definidos, a cratera de um vulcão (Figura 7) e os meandros de um rio (Figura 8). As Figuras 5 a 8 apresentam o *rendering*, realizado no 3D Studio, dos terrenos criados a partir de três execuções de cada TP.

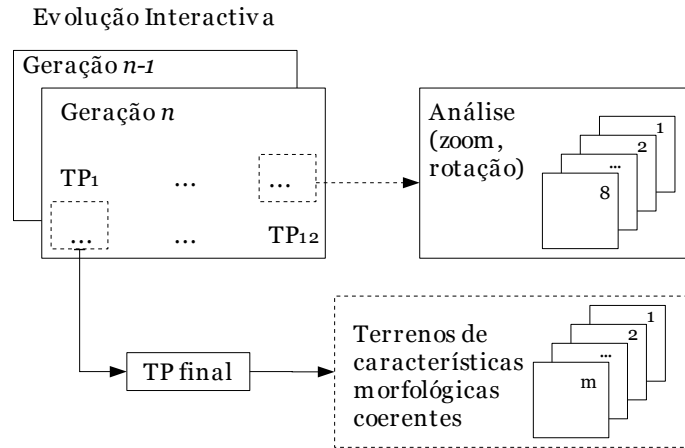


Figura 4 – Esquema funcional do GenTP

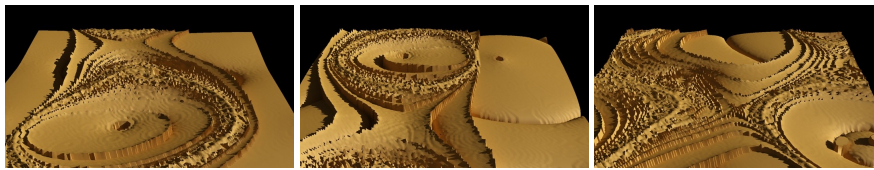


Figura 5 – Terrenos gerados pelo TP da Eq. (1), evoluído pelo apelo estético

$$TP = \text{myLog}(\text{myLog}(\text{myMod}(\text{myLog}(\text{fftGen}(3.75)), \text{myLog}(\text{myLog}(\text{fftGen}(4.25)))))) \quad \text{Eq. (1)}$$

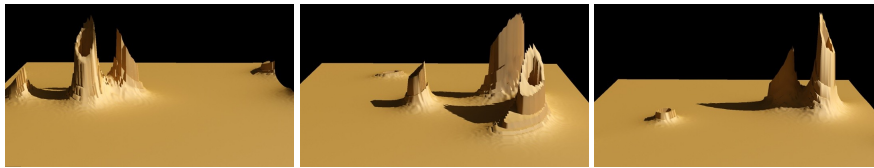


Figura 6 – Terrenos gerados pelo TP da Eq. (2), evoluído pelo apelo estético

$$TP = \text{myDivide}(\text{myDivide}(\text{myDivide}(\text{myDivide}(\text{plane}(1), \text{fftGen}(4.00)), \text{fftGen}(4.00)), \text{fftGen}(4.00)), \text{fftGen}(4.00)) \quad \text{Eq. (2)}$$

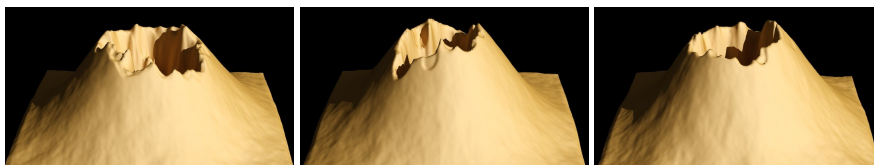


Figura 7 – Terrenos gerados pelo TP da Eq. (3), evoluído com o objectivo de obter a cratera de um vulcão

$$TP=plus(fftGen(2.00),smooth(myMod(gauss(0.75),cos(fftGen(1.00)))))) \quad \text{Eq. (3)}$$

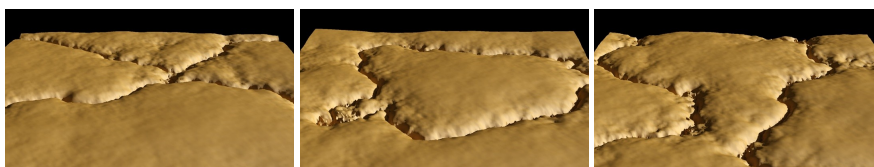


Figura 8 – Terrenos gerados pelo TP da Eq. (4), evoluído com o objectivo de obter os meandros de um rio

$$TP=myLog(myLog(cos(minus(fftGen(2.00),fftGen(3.75)))))) \quad \text{Eq. (4)}$$

#### 4. Conclusões

Neste artigo foi apresentado o GenTP, uma ferramenta interactiva para criar terrenos artificiais. Esta ferramenta implementa uma abordagem evolutiva baseada em PG que permite gerar e evoluir TPs. A evolução é influenciada pela quantidade de TPs seleccionados. Se for seleccionado apenas um a evolução será lenta e se forem seleccionados dois TPs a próxima geração apresentará mais diversidade. Com esta ferramenta um *designer* poderá inspeccionar detalhadamente as características de um determinado TP através da função de análise. Nessa inspecção poderão ser tidos em conta os aspectos morfológicos, a persistência e o grau de variação das características morfológicas dos terrenos gerados. Através de uma série de experiências mostrou-se que várias execuções de um TP geram sempre terrenos diferentes, mas de características morfológicas coerentes. Além disso os TPs podem ser usados de forma procedimental e, por isso, ser facilmente embudados num vídeo jogo. O GenTP permitiu demonstrar a validade da técnica *Genetic Terrain Programming* estando previsto a sua incorporação num vídeo jogo.

O potencial mostrado pelo GenTP aponta-nos novas linhas de pesquisa. Uma delas será efectuar a evolução de terrenos tendo em conta o tempo de execução de cada TP. O desempenho é um aspecto importante se pretendermos gerar terrenos em

tempo real, por exemplo num vídeo jogo. Outro aspecto importante será estudar os terminais e funções que mais influenciam as características de um TP com o objectivo de minimizar o número de gerações necessárias para obter TPs com as características desejadas. Outros elementos poderão ser integrados no GenTP de forma a evoluir TPs que permitam a construção de uma paisagem completa, incluindo vegetação, água e edifícios.

## Referências

- Bentley, P. (1999). *Evolutionary design by computers* (P. J. Bentley, Ed.). CA, USA: Morgan Kaufmann Publishers, Inc.
- Bäck, T., Hoffmeister, F., & Schwefel, H.-P. (1991). A survey of evolution strategies. *Proceedings of the Fourth International Conference on Genetic Algorithms*, 2-9.
- Darwin, C. (1859). *On the origin of species by means of natural selection*. John Murray.
- Duchaineau, M., Wolinsky, M., Sigeti, D., Millery, M., Aldrich, C., & Mineev-Weinstein, M. (1997). ROAMing terrain: Real-time optimally adapting meshes. In *Vis '97: Proceedings of the 8th conference on visualization '97* (pp. 81-88). Los Alamitos, CA, USA: IEEE Computer Society Press.
- Fogel, L., Owens, A., & Walsh, M. (1966). *Artificial intelligence through simulated evolution*. Wiley.
- Frade, M., Vega, F. F. de, & Cotta, C. (2008). Modelling video games' landscapes by means of genetic terrain programming - a new approach for improving users' experience. In M. Giacobini & et al. (Ed.), *EvoWorkshops 2008* (Vol. 4974, pp. 485-490). Napoli, Italy: Springer.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.
- Kelley, A., Malin, M., & Nielson, G. (1988). Terrain simulation using a model of stream erosion. In *Siggraph '88: Proceedings of the 15th annual conference on computer graphics and interactive techniques* (pp. 263-268). NY, USA: ACM.
- Koza, J. R. (1992). Genetic programming. on the programming of computers by means of natural selection. *Cambridge MA: The MIT Press*.
- Machado, P., & Cardoso, A. (2000). NEvAr - the assessment of an evolutionary art tool. In G. Wiggins (Ed.), *Proceedings of the aisb'00 symposium on creative & cultural aspects and applications of ai & cognitive science 2000*. Birmingham, UK.

- Olsen, J. (2004). Realtime procedural terrain generation - realtime synthesis of eroded fractal terrain for use in computer games. *Department of Mathematics And Computer Science (IMADA), University of Southern Denmark*.
- Ong, T. J., Saunders, R., Keyser, J., & Leggett, J. J. (2005). Terrain generation using genetic algorithms. In *Gecco '05: Proceedings of the 2005 conference on genetic and evolutionary computation* (pp. 1463-1470). NY, USA: ACM.
- Peitgen, H.-O., Jürgens, H., & Saupe, D. (2004). *Chaos and fractals - new frontiers of science* (2nd ed.). Springer.
- Sims, K. (1991). Artificial evolution for computer graphics. In *Siggraph '91: Proceedings of the 18th annual conference on computer graphics and interactive techniques* (pp. 319-328). NY, USA: ACM.
- Sims, K. (1992). Interactive evolution of dynamical systems. In F. Varela & P. Bourguine (Eds.), *Toward a practice of autonomous systems: Proceedings of the first European conference on artificial life* (p. 171-178). Paris, FR: MIT Press.
- Unemi, T. (2000). SBART 2.4: an IEC tool for creating 2D images, movies, and collage. In *Proceedings of 2000 genetic and evolutionary computational conference* (p. 153). NV, USA.