

# UM METODO AUTOMATICO PARA A GERAÇÃO DE AMBIENTES DIVERSIFICADOS EM JOGOS

## A METHOD FOR AUTOMATIC ENVIRONMENTS GENERATION GAMES DIVERSIFIED

Data de entrega dos originais à  
redação em: 18/02/2016.  
e recebido para diagramação em:  
30/11/2016

Héllen Caroline Salvato<sup>1</sup>  
Marcio Kassouf Crococomo<sup>2</sup>

*Esse trabalho propõe um método para a criação de diferentes cenários em jogos de computador e mostra o funcionamento da técnica proposta ao implementar um software capaz de criar tais ambientes. O método é dividido em duas etapas, sendo a primeira responsável pela seleção de diferentes terrenos que devem compor o ambiente, e a segunda a seleção e proliferação da vegetação nesses terrenos. Os resultados mostram a possibilidade da criação rápida de uma grande quantidade de ambientes que parecem apresentar uma maior naturalidade do que ambientes criados de forma totalmente aleatória, sendo esperada assim uma maior imersão do jogador. Uma continuação deste trabalho é a de adicionar, além da vegetação já presente no ambiente, simulações de vida animal no mesmo, colaborando ainda com a imersão do jogador nos ambientes gerados.*

**Palavras-chave:** Games. Desenvolvimento de Jogos. Geração Aleatória. Simulação.

*In this Project, we propose a method, which is able to create several different scenarios to be used in computer games. To show the use of the proposed technique, we developed a software capable of generating scenarios as proposed. The presented technique is composed of two different parts: i) the selection of different terrains, which together form the map, and ii) the selection and proliferation of plants, forming the environment vegetation. Results show that the technique, inspired by its simplicity, can generate satisfactory environments for games, positively influencing the immersion of the player. An expected continuation of this project, is the implementation of an additional step, responsible for generation artificial life forms to populate the environment, further collaborating to the player immersion.*

**Keywords:** Games. Games Development. Random Generation. Simulation.

### 1 INTRODUÇÃO

Atualmente com o crescimento do universo tecnológico, o ramo de jogos é o que mais vem se destacando na área de desenvolvimento de softwares para entretenimento (DELFINO; PAVARINI, 2011). Em (GEDIGAMES, 2014) é apontado como um dos maiores motivos do crescimento da importância e desenvolvimento dos jogos digitais a atual mudança do público alvo, que costumava ser formado, em sua maioria, por jovens do sexo masculino, e atualmente possui uma composição mais variada envolvendo homens, mulheres de várias faixas etárias. Além disso, seu uso passou as fronteiras do entretenimento com a criação dos jogos sérios (SUSI; JOHANNESSON; BACKLUND, 2007) que são usados na educação, reabilitação, pesquisas científicas, treinamentos entre outros.

<sup>1</sup>Aluna do curso de Análise e Desenvolvimento de Sistemas, IFSP campus Piracicaba.

<sup>2</sup>Docente do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus Piracicaba.

Os jogos vêm se tornando cada vez mais complexos, apresentando uma enorme riqueza em detalhes gráficos, exigindo grandes equipes e um longo tempo de trabalho para serem produzidos (CARLI, 2012).

O desenvolvimento de técnicas procedurais (CARLI, 2012) vem aumentando, essas são usadas como ferramentas para automatizar a criação de elementos do jogo, como cidades, prédios, vegetação entre outros, tendo como objetivo auxiliar o trabalho dos artistas, possibilitando a criação de jogos mais atrativos e com custo e tempo de desenvolvimento reduzido.

## 2 TRABALHOS RELACIONADOS

Trabalhos envolvendo técnicas procedurais para a criação de cenários em jogos foram estudados, dentre os quais podem ser citados (TEIXEIRA; COELHO; GUBITOSO, 2014), que propõe o desenvolvimento de uma ferramenta para construção de mapas, com o objetivo de economizar tempo dos desenvolvedores, e (DUARTE, 2012), que possui objetivo similar, no entanto, focando na criação de cenários para jogos de plataforma, especificando o gênero aventura. Além desses, dois outros trabalhos (CARLI, 2012; CHIONG; NERI; MCKAY, 2009) se baseiam em aspectos de ambientes reais para a criação dos cenários.

Em (CARLI, 2012), é proposto um método procedural que se baseia em técnicas de computação gráfica, visão computacional e busca em grafos, para a construção de cenários aleatórios para jogos digitais em 3D. No trabalho em questão, cenários são construídos a partir da reprodução de características obtidas analisando imagens de ambientes reais. Uma abordagem diferente é proposta em (CHIONG; NERI; MCKAY, 2009), na qual é realizada a modelagem de biodiversidade prevendo padrões de distribuição de plantas em um ecossistema, formando uma vegetação que pode crescer, se reproduzir e formar paisagens.

## 3 ALGORITMO PROPOSTO

Neste trabalho, é proposto um algoritmo para a construção automática de mapas e vegetações em jogos, com o objetivo de facilitar e agilizar o trabalho do desenvolvedor, e também de trazer maior realidade ao ambiente, aumentando a imersão do jogador. Para demonstrar o funcionamento do algoritmo proposto, foi criado um software utilizando o motor de jogos Unity 3D (UNITY, 2015).

A construção do cenário (ou mapa) é dividida em duas partes: *i*) seleção dos terrenos, explicada na Seção 3.1 e *ii*) seleção e proliferação da vegetação, explicada na Seção 3.2. É importante ressaltar que algumas variáveis do jogo, como as que controlam as quantidades de tipos de terrenos e de plantas, podem ser alteradas de acordo com a necessidade do desenvolvedor. No entanto, neste documento o algoritmo é explicado através de sua implementação no software desenvolvido (um protótipo responsável pela geração dos mapas), que fixa a quantidade de terrenos em cinco e de plantas em 16.

### 3.1 Seleção de terrenos

Para o software desenvolvido foram usados cinco tipos diferentes de terrenos, cada um representado por um código como mostrado na Tabela 1. Os modelos 3D dos mapas utilizados na simulação foram criados utilizando a ferramenta terrain do Unity, e podem ser vistos na Figura 1. Ao utilizar o software desenvolvido nesse projeto, cada desenvolvedor pode utilizar os próprios terrenos, no entanto todos devem possuir o mesmo tamanho.

Tabela 1 - Tipos de terrenos

| Código | Tipo de Terreno  |
|--------|------------------|
| 1      | Água             |
| 2      | Areia            |
| 3      | Terra            |
| 4      | Gramma “Seca”    |
| 5      | Gramma “Molhada” |

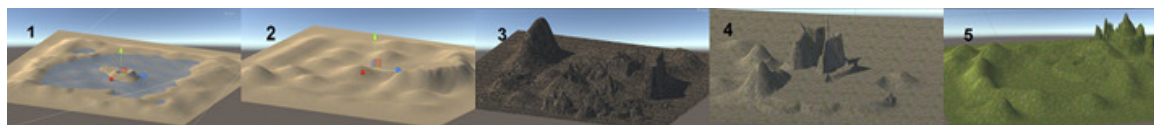


Figura 1- Modelos 3D dos terrenos utilizados

O código é responsável por montar um mapa a partir da combinação de terrenos como os mostrados na Figura 1. Este mapa é composto por um conjunto de terrenos vizinhos, sendo que o primeiro terreno a ser gerado é escolhido de forma aleatória entre todos os disponíveis. Cada terreno seguinte é, sequencialmente, selecionado com base na limitação imposta pelos terrenos vizinhos já definidos, por exemplo, o terreno “grama seca” (4) não pode ser vizinho do terreno “água” (1) e vice-versa. A Tabela 2 ilustra as possibilidades de vizinhança, e a Figura 1 mostra os mapas criados e utilizados na simulação.

Tabela 2 - Possíveis vizinhos

| Tipo | Vizinho A (1) | Vizinho B (2) | Vizinho C (3) | Vizinho D (4) | Vizinho E (5) |
|------|---------------|---------------|---------------|---------------|---------------|
| 1    | OK            | OK            | OK            | -             | OK            |
| 2    | OK            | OK            | OK            | OK            | -             |
| 3    | OK            | OK            | OK            | OK            | OK            |
| 4    | -             | OK            | OK            | OK            | OK            |
| 5    | OK            | -             | OK            | OK            | OK            |

### 3.2 Vegetação

A vegetação se baseia no tipo de terreno, sendo que cada terreno possui um conjunto de possíveis plantas. Com a finalidade de distribuir a vegetação no ambiente, foi criado um objeto 3D chamado de semente. Sementes são criadas por todo o mapa em uma determinada altura (deve-se garantir que essa altura seja mais alta do que o ponto mais alto do terreno para garantir a colisão durante a queda), quando entra em contato com o solo o algoritmo identifica o tipo do mesmo e o conjunto de possíveis plantas que podem nascer no local, das quais uma é selecionada aleatoriamente.

Foi criado também um método de reprodução para as plantas, no qual um parâmetro para cada tipo de planta determina o tempo de reprodução das mesmas. Por exemplo, o desenvolvedor pode ajustar esses parâmetros para que plantas do tipo “cactos” se reproduzam mais lentamente do que plantas do tipo “aquáticas”. No software desenvolvido o tempo varia de acordo com o tamanho da planta e o terreno em que se encontra. Quando o tempo de reprodução é alcançado, por exemplo, a cada minuto de jogo para um determinado tipo de planta, a mesma libera uma determinada quantidade de sementes ao seu redor para gerar novas plantas do mesmo tipo, esse método de reprodução pode ser muito útil em jogos onde o jogador interage com o cenário destruindo árvores e plantas, ou até mesmo com a interação com criaturas não controladas pelo jogador, que se alimentam de plantas.

## 4 RESULTADOS

O software produzido é capaz de criar uma grande quantidade de mapas diferentes, como esperado, sendo isso um bom resultado já que a ideia é criar uma grande variedade de mapas de maneira automática, o que facilita o trabalho do desenvolvedor e aumenta a

quantidade de experiências que o jogador pode ter ao jogar. Além disso, a técnica proposta também possui como objetivo garantir certa realidade aos ambientes criados. Para verificar essa capacidade, foi feito um teste utilizando um algoritmo para criação de mapas utilizando os mesmos recursos porém, de maneira totalmente aleatória (tipos de terrenos e plantas, assim como suas posições, determinadas de forma aleatória).

A Figura 2 ilustra quatro dos mapas criados, sendo os dois mais a esquerda resultados do algoritmo aleatório, e os dois mais a direita resultados do algoritmo proposto nesse trabalho. As duas primeiras imagens apresentam situações não desejadas como, por exemplo, plantas não aquáticas dentro de lagos, ou cactos em ambientes não apropriados. Situações similares não podem ser observadas nas imagens geradas pelo algoritmo proposto nesse trabalho, o que indica uma maior realidade dos mapas gerados, como desejado.



Figura 2 - a) Mapas gerados em uma simulação automática sem limitações e b) Mapas gerados pelo método proposto. Os modelos 3D utilizados nessas imagens foram obtidos dos sites Reiner's Tile Sets<sup>i</sup> e Unity Asset Store<sup>ii</sup>

## 5 CONSIDERAÇÕES FINAIS

O método criado funcionou conforme previsto, podendo ser uma ferramenta útil para os desenvolvedores de jogos. Embora trabalhos como (CARLI, 2012; CHIONG; NERI; MCKAY, 2009) apresentem bons resultados na criação de cenários com o mesmo propósito, o método proposto nesse trabalho deve ser capaz de gerar cenários bons o suficiente para uma grande variedade de jogos, no entanto, possuindo uma menor complexidade de implementação, podendo ser alterado facilmente conforme as necessidades do desenvolvedor.

Possíveis continuações desse trabalho envolvem a criação de técnicas procedurais para a geração de criaturas com a finalidade de aumentar a imersão do jogador, ou mesmo a evolução das criaturas no ambiente gerado ao utilizar técnicas da área de Vida Artificial (NETTO; RINALDI, 2011), ampliando as contribuições deste trabalho ao torna-lo um ecossistema artificial para o estudo de tais técnicas.

## AGRADECIMENTOS

Ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo pela bolsa institucional de iniciação científica fornecida.

## REFERÊNCIAS BIBLIOGRAFICAS

CARLI, D. M. DE. **Geração procedural de cenários 3D de cânions com foco em jogos digitais**. 2012.

CHIONG, R.; NERI, F.; MCKAY, R. An Artificial Life-Based Vegetation Modelling Approach for Biodiversity Research. **Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: Implications in business, Science and Engineering**, p. 1–24, 2009.

DELFINO, W. D. O.; PAVARINI, L. Leas online : protótipo de um jogo de mmorpg. 2011.

DUARTE, P. M. **Geração Procedural de Cenários Orientada a Objetivos** 2012.

**GEDIGAMES. Mapeamento da Indústria Brasileira e Global de Jogos Digitais.** [s.l: s.n.]. Disponível em: <[http://www.bndes.gov.br/SiteBNDES/bndes/bndes\\_pt/Galerias/Arquivos/conhecimento/seminario/seminario\\_mapeamento\\_industria\\_games042014\\_RelApoioCensoIndustriaBrasileiradeJogos.pdf](http://www.bndes.gov.br/SiteBNDES/bndes/bndes_pt/Galerias/Arquivos/conhecimento/seminario/seminario_mapeamento_industria_games042014_RelApoioCensoIndustriaBrasileiradeJogos.pdf)>.

NETTO, M. L.; RINALDI, L. C. A. **Vida artificial: conceitos e aplicações mX SBAI** – Simpósio Brasileiro de Automação Inteligente. **Anais...**São João del-Rei: 2011

SUSI, T.; JOHANNESON, M.; BACKLUND, P. Serious games: An overview. 2007.  
TEIXEIRA, G.; COELHO, C.; GUBITOSO, M. D. **Geração procedimental de ambientes para jogos eletrônicos.** 2014.

UNITY. **Unity - Game Engine.** Disponível em: <<http://unity3d.com/pt>>. Acesso em: 18 jul. 2014.

---

<sup>i</sup> Reiner's Tile sets, disponível em < <http://www.reinerstilesets.de/> > , último acesso em 31/08/2015.

<sup>ii</sup> Unity Asset Store, disponível em < <https://www.assetstore.unity3d.com>>, último acesso em 07/09/2015.