

GERAÇÃO PROCEDURAL DE MONTANHAS POR MEIO DE DADOS GEOLÓGICOS

Eric Gomes Gregory (IC) e Ismar Frango Silveira (Orientador)

Apoio: PIVIC Mackenzie

RESUMO

Neste artigo de iniciação científica, buscou-se estudar o comportamento de placas tectônicas, por meio de modelos matemáticos, para se desenvolver um programa capaz de gerar proceduralmente terrenos através da simulação das interações entre placas tectônicas. O projeto é composto por duas componentes, uma teórica em que foi feita uma pesquisa de caráter bibliográfico sobre técnicas de geração procedural de conteúdo, comportamento de placas tectônicas, e diagramas de repartição de espaço. Da componente teórica foi produzido um algoritmo que pode ser aplicado na plataforma de desenvolvimento de jogos Unity, utilizando a linguagem de programação C#. Já na componente prática do projeto o programa obtido cria uma área plana que é subdividida em um número de placas tectônicas, utilizando diagramas de Voronoi, que são movimentadas em espaço de textura e suas interações, que consistem em interações entre placas e interações entre placas e o manto. O resultado dessas interações é armazenado em uma nova textura (mapa de altitude). Finalmente o programa aplica os valores do mapa de altitude em uma malha de vértices como uma representação visual. Quanto aos resultados obtidos, o algoritmo possui um tempo de execução de caráter exponencial e baixa variedade no formato das placas tectônicas, devido à implementação das regiões de Voronoi. Porém o algoritmo possui uma sequência de operações baseadas em princípios físicos com variáveis mais intuitivas se comparadas às variáveis que compõem o Perlin noise, além disso o programa gera terrenos acidentados como de esperado.

Palavras-chave: Geração procedural de conteúdo, terreno, tectônica

ABSTRACT

In this scientific initiation article, we sought to study the behavior of tectonic plates, through mathematical models, to develop a program capable of procedurally generating terrains through the simulation of interactions between tectonic plates. The project consists of two components, a theoretical one in which a bibliographic research was carried out on techniques of procedural content generation, behavior of tectonic plates, and space distribution diagrams. From the theoretical component, an algorithm was produced that can be applied to the Unity game development platform, using the C# programming language. In the practical component of the project, the program obtained creates a flat area that is subdivided into several tectonic plates, using Voronoi diagrams, which are moved in texture space and their interactions, which consist of interactions between plates and interactions between plates and the cloak. The result of these interactions is stored in a new texture (altitude map). Finally, the program

applies the altitude map values to a mesh of vertices as a visual representation. As for the results obtained, the algorithm has an exponential sum runtime and low variety in the shape of the tectonic plates, due to the implementation of Voronoi regions. However, the algorithm has a sequence of operations based on physical principles with more intuitive variables compared to the variables that make up Perlin noise, in addition the program generates rough terrain as expected.

Keywords: Procedural content generation, terrain, tectonics

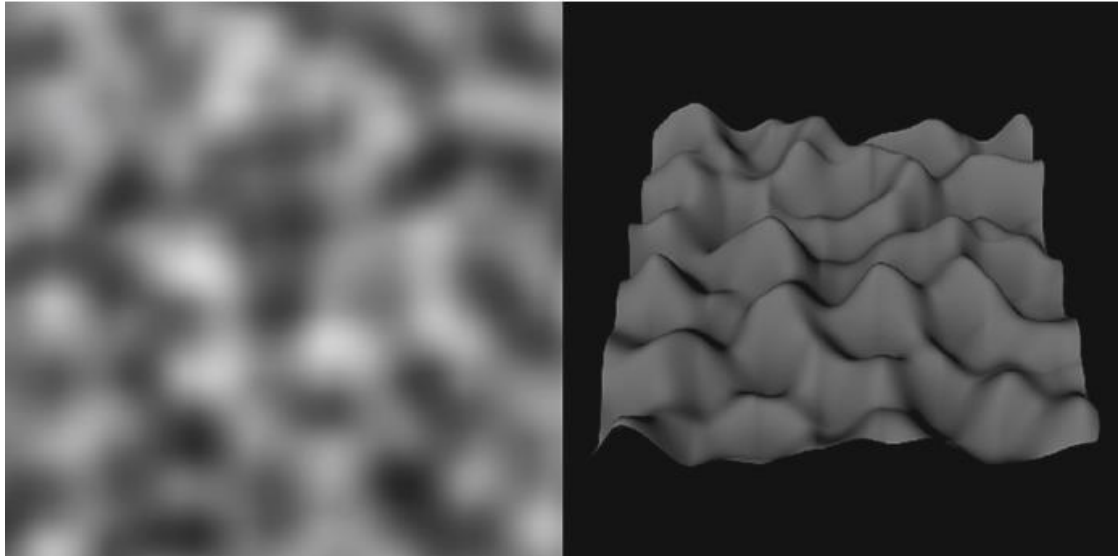
1. INTRODUÇÃO

Os jogos eletrônicos são uma das maiores indústrias no ramo de entretenimento, tendo um rendimento estimado de 159,3 bilhões de dólares americanos em 2020, segundo a plataforma de estatística Statista. Dentre os jogos mais populares das últimas décadas, Minecraft (www.minecraft.net/) e No Mans Sky (www.nomanssky.com/) são os jogos que fazem parte de um subgênero de jogos de exploração conhecidos como jogos proceduralmente gerados.

Geração procedural é a aplicação de um ou mais algoritmos formais para gerar conteúdo de jogos que normalmente seriam produzidos por humanos (SMITH, 2015) como itens, personagens e especificamente para este artigo terrenos. Na geração procedural de terreno inúmeras técnicas computacionais podem ser utilizadas, em isolamento ou mais comumente em conjunto, para se obter o resultado desejado.

Dentre as técnicas mais comuns o uso de Perlin noise (Ken Perlin 1985), uma função computacional capaz de gerar ruído que interpola entre pontos de alta e baixa que gera uma textura em escalas de cinzas. Devido à essas características o Perlin noise é um dos métodos mais comuns de se gerar terreno proceduralmente uma vez que uma textura de Perlin noise pode ser traduzida em elevação apenas amostrando os pixels da textura, como demonstrado abaixo.

Figura 1: *Textura de Perlin noise em escala de cinzas à esquerda e malha tridimensional como terreno baseado na textura à direita*



Fonte: Van Eck e Lamers

Como se pode observar na Figura 1, terrenos gerados utilizando Perlin noise são limitados à aleatoriedade do ruído e sua natureza suavizada que requer inúmeras técnicas de pós processamento do ruído para se obter terrenos mais realistas.

Outra técnica utilizada para a geração procedural de conteúdo são os Autômatos celulares, sistemas de agentes temporais compostos por regras simples capazes de gerar comportamentos complexos a partir da interação entre múltiplos agentes, popularizados pelo Jogo da Vida de Jhon Conway.

Os autômatos celulares são muito utilizados na geração procedural de conteúdo para gerar formas orgânicas, como sistemas de cavernas, como demonstrado por Sebastian Lague em sua série de tutoriais Procedural Cave Generation.

Figura 2. Caverna gerada a partir de autômatos celulares



Fonte: Sebastian Laque

Porém, autômatos celulares são sistemas complexos que requerem uma de duas coisas, ou a meticulosa consideração de todas as regras estabelecidas para determinar o comportamento final do algoritmo; ou um processo extenso e técnico de experimentação de inúmeras regras de geração para se encontrar um resultado satisfatório.

Em anos mais recentes, com a popularização do uso de inteligências artificiais e aprendizagem de máquina, alguns pesquisadores têm utilizado IA para criar terrenos, como no artigo *Controlled Procedural Terrain Generation Using Software Agents* de J. Doran e I. Parberry.

Porém os terrenos gerados demandam uma altíssima quantidade de dados para o treinamento das IAs e devido à recente popularização ainda está muito limitado ao campo da pesquisa.

Dado o contexto da indústria e das técnicas utilizadas na geração procedural de terreno ponderando suas vantagens e desvantagens, foram então estabelecidos os seguintes objetivos:

Objetivo Principal

Desenvolver um programa, dentro da plataforma de desenvolvimentos de jogos Unity, capaz de gerar proceduralmente terreno montanhoso utilizando uma simulação de placas tectônicas.

Objetivos Secundários

- Fazer um levantamento bibliográfico sobre geração procedural de conteúdo
- Fazer um levantamento bibliográfico sobre a formação de montanhas e movimentação de placas tectônicas.

Com os objetivos estabelecidos, um algoritmo foi pré-concebido e serviu tanto como uma estrutura geral para o programa a ser construído como um norte para indicar as lacunas que deveriam ser preenchidas:

- Primeiro define-se um número de placas N e número de iterações da simulação T .
- Em seguida cria-se uma malha de pontos M plana e quadrada de largura L .
- Então cria-se uma textura de mesmo tamanho que a malha M , a textura então é dividida em N regiões de Voronoi V .
- Então, para cada região V_i ($0 \leq i < N$) estabelecem as propriedades físicas da placa, segundo a teoria contemporânea de tectonia.
- Então itera-se pela simulação até se atingir T iterações, executando o movimento de cada placa na textura. Quando uma interação física decorrente da simulação resulte em uma força de deslocamento, este deslocamento é aplicado no ponto M_{ij} ($0 \leq i < L$) e ($0 \leq j < L$) da malha correspondente ao ponto da textura.

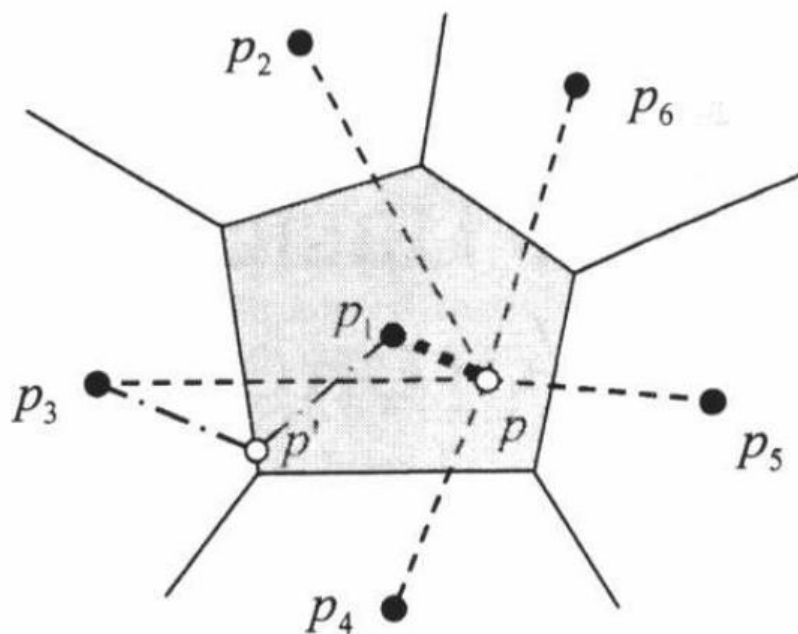
2. DESENVOLVIMENTO DO ARGUMENTO

O algoritmo descrito anteriormente serviu como um norte para o desenvolvimento do projeto, primeiro revelando que a principal lacuna a ser preenchida deveria ser o comportamento físico das placas tectônicas e segundo indicando aspectos técnicos como diagramas de Voronoi e manipulação de texturas como tecnologias essenciais para o desenvolvimento deste projeto.

Foi feita uma pesquisa sobre o algoritmo de partição de espaço conhecido como diagrama de Voronoi. A técnica consiste em dividir pontos em uma área de forma que os pontos contidos em uma região estejam sempre mais próximos do centro de

sua região do que do centro de qualquer outra região. Como exemplificado pela figura abaixo.

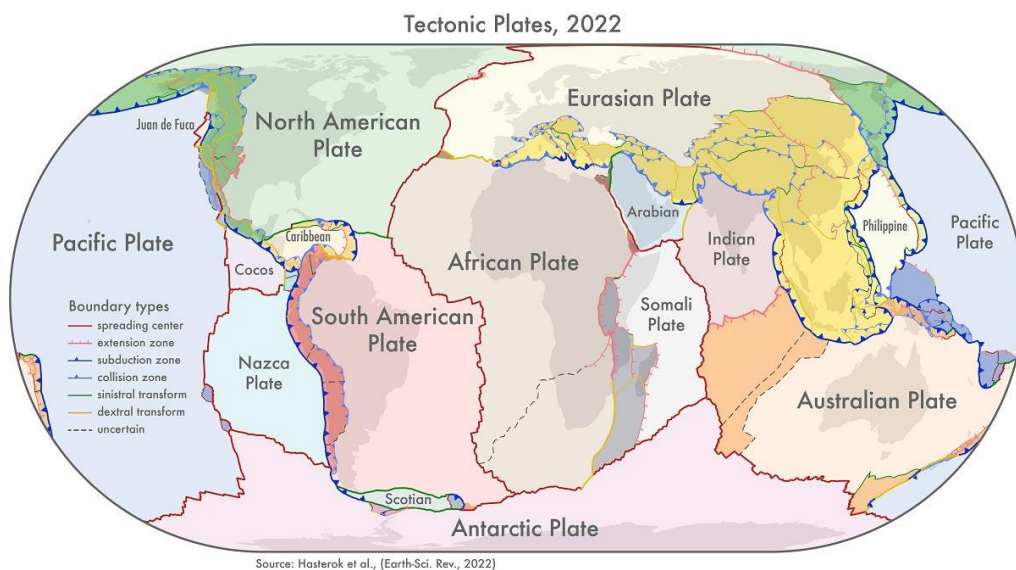
Figura 3. Diagrama de Voronoi simples



Fonte: NOLAN, 2020

Essa técnica é extensamente utilizada nesta pesquisa uma vez que ela gera regiões contínuas, que se aproximam do formato esperado das placas tectônicas. Como exemplificado pela figura 4 abaixo.

Figura 4. Novo modelo de placas tectônicas



Fonte: Derrick Hasterok

Em seguida, com base nas lacunas de conhecimento detectadas anteriormente, a pesquisa verteu para um aspecto bibliográfico não computacional em que buscou-se entender o comportamento físico das placas tectônicas e as formas de interação possíveis.

O artigo de Pechersky, Pirogov, Sadowski e Yambartsev “The Dynamics of Tectonic Plates”, se tornou então uma peça seminal para a pesquisa. Nele os autores detalham um modelo matemático estocástico baseado em cadeias de markov para sistemas de movimentação e interação entre placas tectônicas.

No decorrer do artigo os autores definem modelos matemáticos para tanto a micro quanto para a macro escala da movimentação das placas. Além disso os modelos também foram extrapolados para generalizar o comportamento de forma a operar até em espaços infinitos. A pesquisa desenvolvida se limitou a estudar apenas os modelos em espaço finito.

No modelo utilizado, primeiro assume-se uma situação em que um espaço estático denominado substrato é sobreposto por uma ou mais placas tectônicas. Tanto as placas quanto o substrato possuem asperezas, pontos de possível contato entre placa e substrato.

Então, para cada placa, aplica-se uma força F constante que resulta no movimento de cada placa. No decorrer desse movimento as asperezas de cada placa podem interagir com as asperezas do substrato causando uma força G no ponto de contato ω que faz parte do conjunto de colisões Ω . O comportamento dessas interações é evidenciado pela fórmula abaixo.

$$G = [F - k \sum_{\omega \in \Omega} \min \{x_{\omega}, x_{\omega}^{\alpha}\}] , \quad (1)$$

Fórmula (3.1) retirada de “The Dynamics of Tectonic Plates”, Capítulo 3 “The Model Description. Finite Volume”, página 6

Nela os autores descrevem o comportamento da interação entre as asperezas como uma função elástica com constante k que é aplicada à aspereza de menor intensidade, seja ela da placa ou do substrato, de forma a gerar uma força G a partir da diferença do resultado da função elástica com o vetor F .

A força G , obtida na fórmula anterior, pode facilmente ser traduzida em uma velocidade v utilizando a fórmula abaixo em que v é uma proporção γ de G . Segundo o artigo, esse comportamento não newtoniano ocorre uma vez que as placas estão

parcialmente imersas em um meio viscoso, o manto, que age de forma a gerar inércia apesar da tremenda massa das placas.

$$\mathbf{v} = \gamma \mathbf{G} \quad (2)$$

Fórmula (3.2) retirada de “The Dynamics of Tectonic Plates”, Capítulo 3 “The Model Description. Finite Volume”, página 6

Esse comportamento proporcional parece abstrato e antitético à simulação minuciosa descrita pelo artigo, porém quando se trata de placas tectônicas as distâncias utilizadas como unidades de simulação são enormes e, portanto, a descrição proporcional é uma boa simplificação da realidade.

Sabemos também que a velocidade pode ser traduzida em deslocamento utilizando a fórmula de deslocamento, como descrita abaixo:

$$\mathbf{s} = \mathbf{s}_0 + \mathbf{v}t \quad (3)$$

Fórmula de deslocamento

Utilizando a fórmula de deslocamento como base, pode-se substituir a velocidade $\mathbf{v}$ pela fórmula 3.2 de The Dynamics of Tectonic Plates para se obter o seguinte desenvolvimento.

$$\mathbf{s} = \mathbf{s}_0 + (\gamma \mathbf{G})t \quad (4)$$

$$\mathbf{s} = \mathbf{s}_0 + \left(\gamma \left[F - k \sum_{\omega \in \Omega} \min \{x_{\omega}, x_{\omega}^{\alpha}\} \right] \right) t \quad (5)$$

Fórmula resultante (Autoral)

Essa nova fórmula foi utilizada no programa para obter o valor de deslocamento para todos os contatos entre as placas e o substrato de forma a solucionar a principal lacuna do projeto.

Além disso a fórmula obtida possui três constantes $[\gamma, k, \alpha]$, que respectivamente controlam a inércia do manto, a elasticidade da superfície e a intensidade das asperezas, que podem ser utilizadas por artistas e designers para controlar o terreno gerado sem necessitar o conhecimento técnico do código.

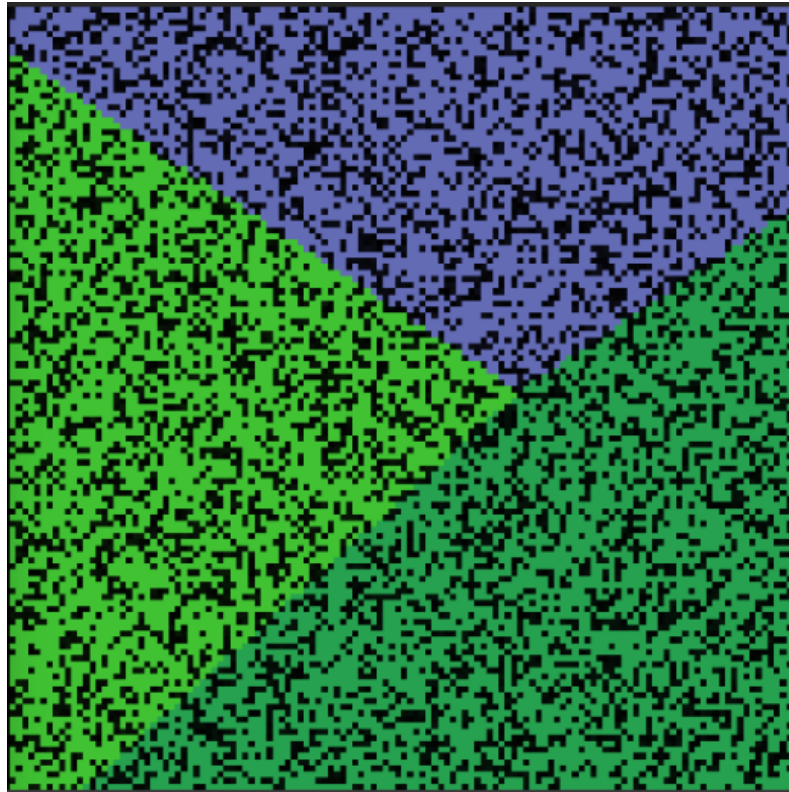
Com base nos modelos estudados e da fórmula obtida, um novo algoritmo foi projetado, preenchendo as lacunas com os modelos matemáticos do artigo discutido anteriormente.

- Primeiro define-se um número de placas N e número de iterações da simulação T .
- Em seguida cria-se uma malha de pontos M plana e quadrada de largura L .
- Então cria-se duas texturas R e O de mesmo tamanho que a malha M , uma para as placas e outra para armazenar o deslocamento.
- Então cria-se um mapa de asperezas, pontos P_{ij} ($0 \leq i < L$) e ($0 \leq j < L$) com um valor A de aspereza ($A > 0$).
- A textura das regiões é então dividida em N regiões de Voronoi V e cada ponto da textura é dado um valor A de aspereza ($A > 0$).
- Então, para cada região V_i ($0 \leq i < N$) estabelece o vetor V de movimento da placa e sua densidade D .
- Então itera-se pela simulação, movendo os pixels da textura R de acordo com o vetor V_i de cada placa, até se atingir T iterações:
 - Quando um ponto R_{ij} está sobreposto a um ponto do mapa de asperezas, uma interação tectônica é registrada na textura O .
 - Quando um ponto R_{ij} está sobreposto a um ponto de outra placa, verifica-se a densidade de cada placa para se determinar o “vencedor” e registra-se o pixel da placa menos densa no ponto R_{ij} e registra-se também uma interação sísmica na textura O .
- Finalmente aplica-se os valores, na escala de cinzas, da textura O em M , obtendo um terreno gerado a partir de princípios tectônicos.

Em seguida o projeto verteu para uma componente prática em que se foi utilizada a plataforma de desenvolvimento de jogos Unity para implementar o algoritmo projetado.

O programa ao inicializar cria uma textura T_1 de tamanho pré-determinado que é subdividida entre o número de placas tectônicas. Essas são inicializadas com um centro e vetor de movimento F aleatórios. Durante a subdivisão de T_1 o canal alpha (canal de transparência) é utilizado para armazenar o valor das asperezas e é criada uma lista de asperezas para o substrato.

Figura 5. Textura que representa a região simulada subdividida em placas tectônicas.



Fonte: Autoral

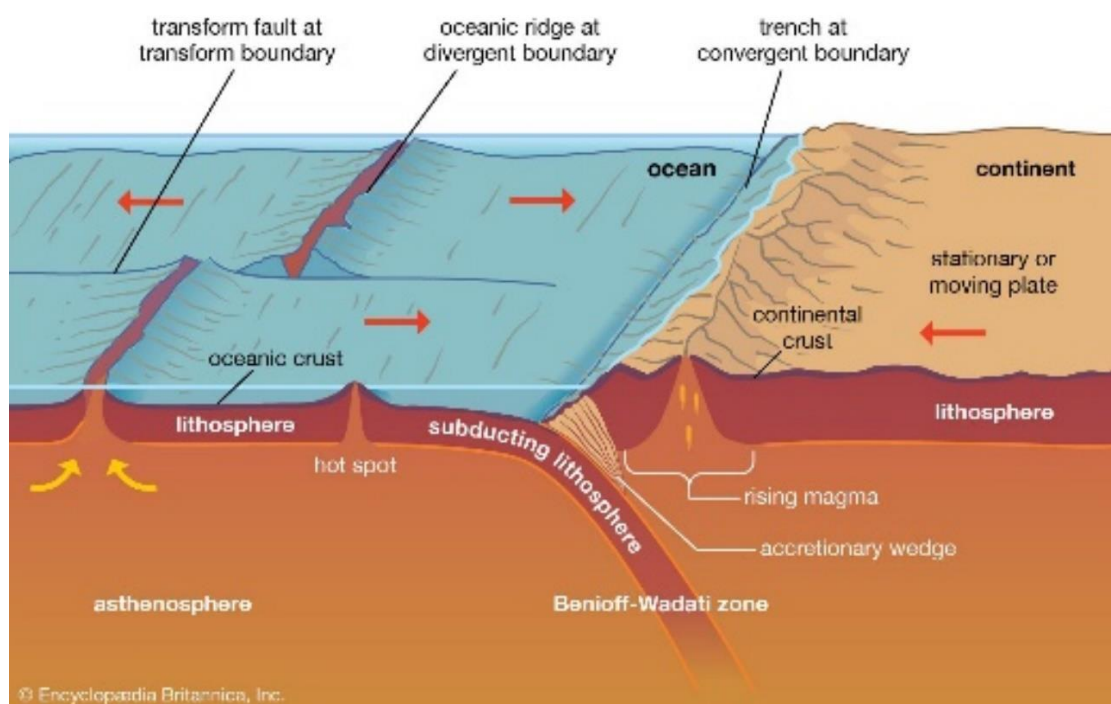
Feita essa inicialização o programa então itera pelo tempo de simulação movimentando as placas e armazena o resultado da interação entre as asperezas do substrato e das placas, utilizando a fórmula (1.0), em uma nova textura T_2 que representa altitude.

O sistema executa a simulação a cada iteração utilizando a função **SimulateStep**, que recebe três mapas como entradas (Anterior, Atual e Próximo). Na primeira iteração tanto o Anterior quanto o Atual são idênticos e o Próximo serve como uma referência para o resultado da iteração.

Então para cada iteração da simulação a função executa a movimentação dos pontos das placas e compara a posição final de cada ponto com o mesmo ponto no passo anterior. A cada ponto que sobreponha uma placa distinta da atual, verifica-se qual delas é menos densa e registra-se a menos densa no mapa próximo.

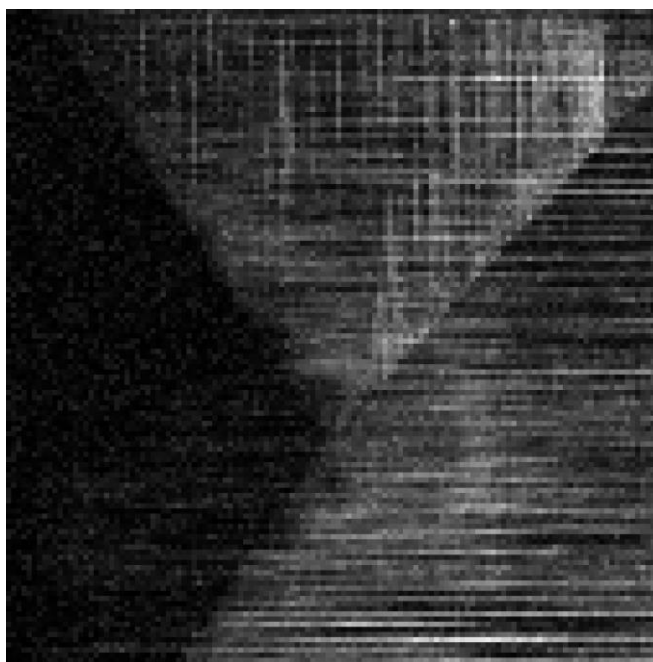
Assim o sistema simula o fenômeno da subducção em que durante o choque entre placas tectônicas, a placa de menor densidade tende a se sobrepôr sobre a placa de maior densidade, forçando-a para dentro do manto. Durante esse processo essa sobreposição causa o deslocamento vertical da região da placa menos densa, causando a formação de montanhas.

Figura 6. Processo de subducção e formação de montanhas por colisão entre placas de densidades distintas.



Fonte: Enciclopédia Britânica Inc. Acesso em <https://www.britannica.com/science/plate-tectonics/Seafloor-spreading>

Figura 7. Mapa de altitude resultante da simulação

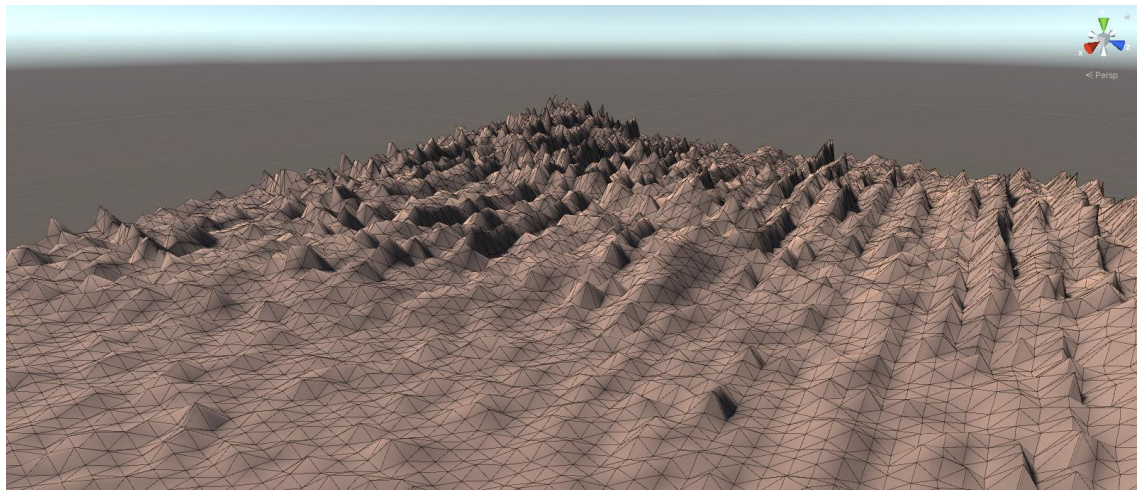


Fonte: Autoral

Como se pode observar na Figura 6 (mapa de altitude T_2) as placas obtidas na Figura 5 foram movimentadas por 200 iterações e interagiram tanto com o substrato e entre si para resultar em uma imagem de escala de cinzas.

Com o mapa de altitude T_2 obtido (Figura 6) o programa então traduz o valor em escala de cinzas para altitude aplicando-a em uma malha geométrica (Terreno) de mesmo tamanho que a textura. Assim obtendo a imagem abaixo.

Figura 8. Terreno gerado a partir do mapa de altitude T_2



Fonte: Autoral

Uma das principais vantagens de armazenar o resultado da simulação tectônica em uma textura, ao invés de aplicar os valores diretamente à malha, é a dissociação entre a lógica e o visual do algoritmo. Essa separação permite que a malha possa ser substituída por outras técnicas de representação geométrica como Voxels ou Marching Cubes, de forma a se adequar às necessidades de cada projeto.

Além disso a questão de armazenamento também se torna importante, uma vez que é tanto computacionalmente mais barato armazenar uma imagem e reconstruir o terreno a partir dela, do que armazenar uma malha inteira no disco.

O código do projeto está disponível na plataforma de armazenamento de código e versionamento GitHub (<https://github.com/EricGomesGregory/IC-TCC-PlateTectonics>) sobe a licença de software MIT.

3. CONSIDERAÇÕES FINAIS

Neste projeto de iniciação científica foi feito uma pesquisa que resultou em um programa de geração de terreno montanhoso por meio da simulação da movimentação de placas tectônicas.

Além disso o projeto, através da pesquisa bibliográfica, possibilitou a intersecção de conhecimentos díspares como a modelagem matemática do comportamento das placas tectônicas, técnicas de repartição do espaço como diagramas de Voronoi e geração procedural de conteúdo.

Esse processo de aprendizado resultou em um algoritmo de simulação que pode ser aplicado como uma base para a construção de mundos digitais para jogos ou outras simulações no âmbito científico.

Do ponto de vista técnico o programa obtido gera terrenos acidentados com um grau elevado de ruído se comparado com Perlin noise, porém seu comportamento físico é mais simples de se compreender do que o sistema matemático abstrato de Perlin.

Contudo o programa sofre uma principal limitação, seu tempo de execução tende a crescer de forma exponencial devido à natureza da simulação resultando em minutos de espera para espaços muito grandes (maiores que 1024×1024) ou para simulações com mais de 1000 iterações. Porém quando se trata de geração procedural de terreno uma demora de alguns minutos não é tão inesperada além disso o projeto pode ser agilizado utilizando algumas técnicas de otimização, como a utilização de compute shaders para permitir que placas de vídeo executem algumas operações do programa.

Como trabalhos futuros, o programa pode ser aprimorado das seguintes três maneiras, apresentadas em ordem crescente de complexidade:

Poderia se modificar a geração inicial das placas, tornando as fronteiras entre placas mais irregulares. Para isso um passo adicional seria adicionado ao processo de inicialização das placas em que repartiria o espaço de simulação em inúmeras regiões de Voronoi e essas seriam divididas entre as placas baseado na distância entre o centro de cada subdivisão e o centro de cada placa.

Figura 9. Repartição

Fonte: Autoral

Essa mudança tornaria o terreno mais variado e daria mais controle a artistas e designers uma vez que as subdivisões da malha poderiam ser modificadas para se obter mais variedade na irregularidade das fronteiras entre as placas.

Outra adição ao programa que o tornaria mais variado seria a utilização de texturas de Perlin noise para determinar características da simulação, como a constante elástica, de forma a aumentar a variedade no terreno de forma suavizada uma vez que a elasticidade da placa se tornaria não uniforme e mais realista.

Uma última opção para melhorar, dessa vez o programa em si, seria a utilização de compute shaders, uma tecnologia que permite a utilização de placas de vídeo (GPU) para executar operações simples de forma a aliviar a quantidade de processos da CPU. Se aplicada à simulação, em específico na movimentação das placas, o programa se tornaria mais rápido e eficiente.

Finalmente, uma modificação futura para este projeto pode ser a modificação do código fonte para que o projeto se torne um plugin ou ferramenta. Essa modificação tornaria o conhecimento obtido nesta iniciação científica livre para a utilização nas indústrias criativas como o cinema e principalmente jogos eletrônicos, a indústria que instigou esta pesquisa.

Para isso será necessário um trabalho de otimização do código, que envolveria restringir a aleatoriedade por meio de técnicas de seeding. Além do desenvolvimento de uma interface de usuário mais intuitiva, com um foco em uma experiência de usuário UX autoexplicativa e clara sobre os procedimentos em ocorrência.

Essa futura produção possibilitaria que artistas e designers criassem mundos digitais com maior verossimilitude e com menor esforço.

4. REFERÊNCIAS

E. Pechersky, S. Pirogov, G. Sadowski e A. Yambartsev. (2015) "The Dynamics of Tectonic Plates", IME USP.
https://www.ime.usp.br/~yambar/files/MovingPlatePrepr_JIP.pdf

DOBRIN, Adam. A review of properties and variations of Voronoi diagrams. Whitman College (2005): 1949-3053. Whitman College, [s. l.], 2005.

SMITH, Gillian (2015) "An Analog History of Procedural Content Generation", Northeastern University, Playable Innovative Technologies Lab.
http://www.fdg2015.org/papers/fdg2015_paper_19.pdf

FREIKNECHT, J.; EFFELSBURG, W. A Survey on the Procedural Generation of Virtual Worlds. *Multimodal Technologies and Interaction*, 1(4), 27, 2017.
<https://www.mdpi.com/2414-4088/1/4/27>.

GÉNEVAUX, J. D.; GALIN, E.; GUÉRIN, E.; PEYTAVIE, A.; BENES, B. (2013). Terrain Generation Using Procedural Models Based on Hydrology. *ACM Transactions on Graphics (TOG)*, 32(4), 1-13. <https://dl.acm.org/doi/abs/10.1145/2461912.2461996>. Acesso em 23 de abril de 2021.

GUÉRIN, Éric; DIGNE, Julie; GALIN, Éric; PEYTAVIE, Adrien; WOLF, Christian; BENES, Bedrich; MARTINEZ, Beniît. Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks. Association for Computing Machinery, ACM Transactions on Graphics, 2017.

LAGUE, Sebastian. Procedural Terrain Generation. [S. l.: s. n.], 2019. 21 vídeos (~25 min). Disponível em:
https://www.youtube.com/playlist?list=PLFt_AvWsXI0eBW2EiBtl_sxmDtSqZBxB3.

NOLAN, Ma. Voronoi Diagram. Independente, [s. l.], 2020.

PARBERRY, Ian. "Designer Worlds: Procedural Generation of Infinite Terrain from Real-World Elevation data." *Journal of Computer Graphics Techniques* 3.1 (2014).

PERLIN, Ken. An Image Synthesizer. *ACM SIGGRAPH : Computer Graphics*, [s. l.], 1985 <https://dl.acm.org/doi/10.1145/325165.325247>

PETROVAS, A.; BAUŠYS, R. Automated Digital Terrain Elevation Modification By Procedural Generation Approach. In *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)* (pp. 1-5). IEEE, 2019. <https://ieeexplore.ieee.org/abstract/document/8732171/>.

SMELIK, Ruben M., et al. "A survey of procedural methods for terrain modelling." *Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*. 2009.

VAN ECK, Wim; LAMERS, Maarten H. "Biological content generation: evolving game terrains through living organisms." *International Conference on Evolutionary and Biologically Inspired Music and Art*. Springer, Cham, 2015.

J. Doran e I. Parberry, "Controlled Procedural Terrain Generation Using Software Agents," em *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 2, pp. 111-119, June 2010, doi: 10.1109/TCIAIG.2010.2049020.

LAGUE, Sebastian. Procedural Cave Generation. [S. l.: s. n.], 2015. 9 vídeos (~20 min). Disponível em: https://www.youtube.com/playlist?list=PLFt_AvWsXI0eZgMK_DT5_biRkWXftAO9

HASTEROK Derrick, HALPIN Jacqueline A., COLLINS Alan S., HAND Martin, KREEMER Corné, GARD Matthew G., GLORIE Stijn, "New Maps of Global Geological Provinces and Tectonic Plates", em *Earth-Science Reviews*, Volume 231, 2022, 104069, ISSN 0012-8252, em <https://doi.org/10.1016/j.earscirev.2022.104069>.