Universidade Federal de Juiz de Fora
Faculdade de Engenharia & Instituto de Ciências Exatas
Bacharelado em Engenharia Computacional

Incorporação de Dados Geográficos de Altimetria na Geração Procedural de Terrenos com Funções de Ruído

Leonardo Gregório de Andrade

JUIZ DE FORA
JULHO, 2023

Incorporação de Dados Geográficos de Altimetria na Geração Procedural de Terrenos com Funções de Ruído

Leonardo Gregório de Andrade

Universidade Federal de Juiz de Fora
Faculdade de Engenharia & Instituto de Ciências Exatas
Mecânica Aplicada e Computacional
Bacharelado em Engenharia Computacional

Orientador: Marcelo Caniato Renhe

JUIZ DE FORA
JULHO, 2023

Incorporação de Dados Geográficos de Altimetria na Geração Procedural de Terrenos com Funções de Ruído

Leonardo Gregório de Andrade

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO FACULDADE DE ENGENHARIA & INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA COMPUTACIONAL.

Aprovada por:

Marcelo Caniato Renhe
Doutor em Engenharia de Sistemas e Computação

Igor de Oliveira Knop
Doutor em Modelagem Computacional

Luiz Maurílio da Silva Maciel
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA
10 DE JULHO, 2023

Agradecimentos

Agradeço imensamente à minha família e amigos por serem, simultaneamente, minha espada e minha armadura.

Também expresso meu agradecimento ao meu orientador e a todos os professores que me acompanharam ao longo desta jornada acadêmica.

Além disso, gostaria de estender meus agradecimentos a todos os funcionários da UFJF, que também são parte integral do funcionamento dessa instituição que, apesar de todos os desafios, permanece pública, acessível, gratuita e de qualidade, como sempre.

"My heart is fire, my mind is clear, my spirit is a roaring sea.".

Gavin Aung Than

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DCC	Departamento de Ciência da Computução
UFJF	Universidade Federal de Juiz de Fora
DEM	*Digital Elevation Model*
INPE	Instituto Nacional de Pesquisas Espaciais
MDE	Modelo Digital de Elevação
MFB	Movimento Fractal Browniano
NASA	*National Aeronautics and Space Administration*
NIMA	*National Imagery and Mapping Agency*
SRTM	*Shuttle Radar Topography Mission*

Resumo

A geração procedural de terrenos é uma aplicação bastante comum de algoritmos procedurais de ruído, principalmente no contexto de jogos. Esses algoritmos, de natureza pseudo aleatória, permitem criar terrenos realistas e podem ter seus parâmetros ajustados para produzir terrenos com muitas características diferentes, como lagos, cavernas, planaltos, planícies, montanhas rochosas e picos nevados. Dependendo do ajuste, podem ser criados terrenos com diferentes escalas e granularidades. Nessas aplicações, tais algoritmos permitem criar terrenos relativamente complexos de forma automática e eficiente, reduzindo custos no processo de criação. No entanto, as características do terreno gerado são, normalmente, configuradas manualmente pelo desenvolvedor, tornando a criação de terrenos autênticos um desafio. O presente trabalho explora a utilização de dados geográficos de altimetria integrados ao algoritmo de ruído procedural, com o objetivo de conferir características topológicas de determinada região ao terreno gerado. Para tal, foram coletados dados de altimetria de um Modelo Digital de Elevação (MDE), que oferece valores de altitude para todo o território nacional. Estes dados foram utilizados em conjunto com o Ruído Simplex para a geração de terrenos pseudo-aleatórios, de forma a enviesar o algoritmo para a produção de terrenos com características topográficas com o mesmo perfil estatístico da localidade real. Os experimentos realizados indicam que a técnica utilizada é promissora no refinamento da geração procedural de terrenos, sendo possível observar características das regiões desejadas no terreno produzido algoritmicamente.

Palavras-chave: geração procedural, terrenos, funções de ruído, altimetria.

Abstract

Procedural terrain generation is a widely used application of procedural noise algorithms, especially in the realm of video games. These pseudo-random algorithms enable the creation of realistic terrain with adjustable parameters to produce various features such as lakes, caves, plateaus, plains, rocky mountains and snowy peaks. Depending on the settings, it is possible to create terrains with different scales and granularity. In these applications, the algorithms allow the creation of relatively complex terrain automatically and efficiently, reducing costs in the creation process. However, the characteristics of the generated terrain are usually manually configured by developers, posing a challenge in achieving authentic terrains. This study explores the integration of altitude data into procedural noise generation algorithm, aiming towards conferring topological characteristics of specific regions to the generated terrains. Altitude data were collected from a Digital Elevation Model (DEM) that provides altitude vales for the entire Brazilian territory. These data were then used in conjunction with Simplex noise algorithm to generate pseudo-random terrains, biasing the algorithm for producing terrains with topographic characteristics matching the statistical profile of the real location. Experimental results indicate the promising nature of this technique in refining the standard procedural generation of terrains, allowing the observation of features from the desired region in the algorithmically produced terrains.

Keywords: procedural generation, terrains, noise algorithms, altitude data.

1 Introdução

Nos últimos anos, mídias digitais de entretenimento, como jogos e animações, têm se tornado cada vez mais populares, impulsionando a demanda pela disponibilização de conteúdo variado nestes veículos. No entanto, a produção desse conteúdo exige das empresas responsáveis pelo desenvolvimento um investimento maior em recursos humanos e financeiros. Nesse contexto, a geração procedural surge como alternativa para a criação de conteúdo de forma econômica e escalável.

A abordagem da geração procedural se destaca ao gerar conteúdo automaticamente por meio de algoritmos, sendo mais adequada quando deseja-se criar conteúdo padronizado ou que pode ser gerado com base em regras bem definidas. No contexto de jogos, por exemplo, a geração procedural pode ser empregada para criar personagens, níveis, texturas, comportamentos, narrativa e, em particular, ambientes, incluindo a geração procedural de terrenos. Essa técnica de geração de conteúdo pode ser um recurso potente para estúdios de menor porte produzirem conteúdo em larga escala.

A criação procedural de terrenos em jogos oferece vantagens significativas, principalmente em jogos de mundo aberto, nos quais os jogadores têm ampla liberdade de exploração. A alta variabilidade de cenários, devida à natureza dos algoritmos, causa no jogador uma sensação de maior liberdade e imersão.

Como exemplo, pode-se citar o jogo *Minecraft*[1] (Mojang, 2009), mostrado na Figura 1.1, um jogo baseado em voxels que utiliza uma combinação de diferentes algoritmos de ruído procedural para gerar um terreno com diversos biomas e ambientes exploráveis pelo jogador. Outro exemplo é *No Man's Sky* (Hello Games, 2016)[2], mostrado na Figura 1.2, que é um jogo no qual o jogador tem uma gama aparentemente infinita de planetas e galáxias para explorar, sendo que cada planeta possui sua própria geografia, fauna e clima. Dessa forma, algoritmos de ruído procedural têm papel vital em conferir aos ambientes realismo e variedade.

[1]Disponível para download em: ⟨https://www.minecraft.net/⟩
[2]Disponível para download em: ⟨https://www.nomanssky.com/⟩

Figura 1.1: Minecraft (Mojang, 2009) (Fonte: Capturado pelo Autor)

Figura 1.2: No Man's Sky (Hello Games, 2016) (Fonte: Capturado pelo Autor)

Dentre os algoritmos utilizados na geração procedural de terrenos, destacam-se os algoritmos de ruído, sendo uma ferramenta versátil e poderosa. Esses algoritmos permitem criar terrenos com muitas características diferentes, como lagos, cavernas, planaltos, planícies, montanhas rochosas e picos nevados e, dependendo do ajuste, podem ser criados terrenos com diferentes escalas e granularidades.

Esses algoritmos têm sido amplamente estudados tanto pela indústria quanto pela academia. Eles se baseiam na adição controlada de características aleatórias ao terreno, conferindo a este uma aparência natural e orgânica. Existem diversas funções de ruído, e dentre as mais utilizadas, destacam-se algumas como Ruído de Valor, Ruído de Perlin e Ruído Simplex.

Apesar de serem ferramentas poderosas para a geração procedural de terrenos, o

resultado das funções de ruído depende do ajuste fino de parâmetros, que usualmente é feito manualmente pelo desenvolvedor. Encontrar a combinação ideal de parâmetros para obter um terreno com as características desejadas pode se mostrar uma tarefa difícil. Além disso, devido à natureza das funções de ruído, que geram valores uniformemente distribuídos, é necessário investir tempo em pós-processamento para alcançar um terreno realista e condizente com alguma localidade específica.

1.1 Objetivos

Este trabalho tem como objetivo principal integrar dados de terrenos reais diretamente no processo de geração procedural, para incorporar as características geográficas de um local em terrenos gerados por meio de função de ruído.

Para atingir o objetivo proposto, pretende-se comparar os algoritmos de Ruído de Valor, Ruído de Perlin e Ruído Simplex aplicados à geração procedural de terrenos. Adicionalmente, busca-se avaliar a possibilidade de se incorporar dados geográficos específicos de regiões de Minas Gerais no algoritmo do Ruído Simplex, utilizando uma variação do método proposto em Parberry (2014).

1.2 Organização do trabalho

Este trabalho está estruturado conforme delineado abaixo.

No Capítulo 2 é realizada uma revisão da literatura sobre os temas relacionados a funções de ruído, geração procedural de terrenos e geração procedural de terrenos utilizando dados reais.

O Capítulo 3 discute todos os conceitos fundamentais para a compreensão do estudo. São introduzidos conceitos relacionados a mapas de alturas, além de detalhar os algoritmos de ruído procedural Ruído de Valor, Ruído de Perlin e Ruído Simplex. Também é feita uma explicação sobre altimetria.

No Capítulo 4, é abordada uma técnica para geração de terrenos através de funções de ruído. Além disso, é proposta uma abordagem para a incorporação de dados de altimetria no algoritmo Ruído Simplex.

No Capítulo 5, são caracterizados os experimentos realizados com o objetivo de aplicar os algoritmos de ruído em geração procedural de terrenos e incorporar dados de altimetria no Ruído Simplex. Além disso, são discutidos e analisados os resultados dos experimentos.

O Capítulo 6 discute os resultados e limitações dos métodos, além de propor modificações para trabalhos futuros.

2 Revisão bibliográfica

A modelagem de terrenos representa um desafio, com mais de quatro décadas de pesquisas e desenvolvimento de métodos para sua execução. Dentre as abordagens utilizadas, destaca-se o uso de ruído procedural como uma técnica eficaz e amplamente adotada.

Ken Perlin em Perlin (1985) apresentou pela primeira vez um método de ruído procedural, a fim de abordar o problema de gerar proceduralmente representações gráficas que possuem características suaves e naturais. Nesse trabalho, o autor apresenta um algoritmo eficiente para geração de ruído e o aplica na texturização de diversos objetos. A técnica introduzida por Ken Perlin se tornou um elemento fundamental tanto na indústria quanto na academia.

Em Perlin (2002), o autor aborda deficiências em seu algoritmo original (PERLIN, 1985), propondo melhorias pontuais para o método. Já no artigo Perlin (2001), Ken Perlin aborda uma solução aprimorada ao Ruído de Perlin, o Ruído Simplex. Todavia, a fonte original do algoritmo de Ruído Simplex não oferecia uma explicação detalhada do método, o que levou Stefan Gustavson a revisitar e explicar mais profundamente esse algoritmo em Gustavson (2005).

Além dos citados, outros algoritmos de ruído procedural foram propostos, como em Worley (1996), que propõe o Ruído de Worley, uma função de ruído baseada em células, muito utilizada na geração de estruturas com aspecto natural. Outro método relevante é o denominado Flow Noise, proposto em Perlin e Neyret (2001), que consiste em uma extensão do conhecido Ruído de Perlin, adicionando um componente de movimento ao processo de geração, para criar uma aparência de fluxo contínuo. No livro Ebert et al. (2003), são descritos detalhadamente diversos algoritmos de ruído procedural, juntamente com suas aplicações em texturização e modelagem, incluindo sua utilização na geração de terrenos.

Em relação à geração procedural de terrenos, no artigo Musgrave, Kolb e Mace (1989) é apresentado um método matemático para geração de terrenos fractais através

de ruído. Esse método difere da abordagem proposta por Ken Perlin ao utilizar a interpretação da transformada de Fourier inversa por um mapa de alturas. Posteriormente, Musgrave (1993) revisitou o método proposto, adicionando erosão, técnica que tambem foi explorada em Šťava et al. (2008). Contudo, as técnicas citadas são mais custosas computacionalmente e possuem implementação muito mais complexa quando comparadas à abordagem de Ken Perlin, além de apresentarem discontinuidades em seus contornos.

Existe um interesse notável em personalizar o resultado das funções de ruído na geração procedural de terrenos. No trabalho de Smelik et al. (2010), são demonstradas técnicas para integrar a geração procedural de terrenos com edições manuais realizadas pelo usuário. No entanto, os *designers* necessitam de um nível mais refinado de controle, tornando a tarefa de produzir mundos virtuais consistentes difícil e trabalhosa.

Em Santamaría-Ibirika et al. (2014), é proposta uma abordagem para criar terrenos volumétricos utilizando o ruído de Perlin para as camadas de terreno, que consistem em um conjunto de materiais combinados. Já em Li et al. (2015) é apresentado um método de paralelização em GPU do algoritmo do ruído de Perlin para geração de terrenos, com o objetivo de melhorar o desempenho do algoritmo. O trabalho Gasch et al. (2020) busca aprimorar o controle sobre as funções de ruído que, pela sua natureza aleatória, tornam o ajuste dos resultados difícil. Para superar essa dificuldade, o estudo estabelece restrições tais como elevações máximas, pontos de interesse e áreas de interesse para permitir um controle mais preciso sobre o processo de geração. Diversas outras abordagens sobre geração de ambientes digitais podem ser encontradas em Galin et al. (2019).

Uma abordagem promissora envolve utilizar dados provenientes de Modelos Digitais de Elevação (MDE). Nesse sentido, o estudo realizado em Zhou et al. (2007) apresenta uma técnica para sintetizar terrenos utilizando MDEs através de geração baseada em exemplos, onde características marcantes são extraídas de uma seção de terreno e reproduzidas para o terreno criado.

Nos trabalhos Parberry (2014) e Parberry (2015), o autor demonstrou que funções de ruído podem ser melhoradas ao serem combinadas com dados reais. Nestes trabalhos, foram utilizados valores de altitude de Utah para gerar terreno fidedigno através da integração com o algoritmo de Ruído de Valor, obtendo bons resultados.

Em Guérin et al. (2016), é proposto um método para criação de terreno através da análise de dados provenientes de MDEs e utilização desses dados para a criação de dicionários com as características do terreno, combinando os elementos desses dicionários para formar o terreno final. Já em Guérin et al. (2017), é proposto um método para gerar terrenos através da criação de uma Rede Adversária Generativa, que é treinada utilizando os dados de terrenos reais contidos em MDEs. O trabalho Vitacion e Liu (2019) explora a utilização das funções de ruído para produzir superfície de planetas, além de comparar a performance e escalibilidade de diferentes algoritmos de ruído para este fim.

Seguindo a linha de utilizar dados geográficos para gerar terrenos proceduralmente, Lamontagne (2022) apresentou um método para gerar terrenos acoplando orometria a ruídos celulares e gradientes, obtendo resultados extremamente realistas. Em Jain, Sharma e Rajan (2022), é apresentado um método para gerar terrenos que combina uma Rede Adversária Generativa, assim como em Guérin et al. (2017), porém melhora a geração através de mesclagem entre os terrenos gerados utilizando o ruído de Perlin, possibilitando a geração de terrenos infinitos com características realistas.

O presente trabalho foca no método apresentado por Parberry (2014) e busca aprimorá-lo através da introdução de uma técnica complementar. A abordagem mencionada será detalhada no Capítulo 4.

3 Fundamentação teórica

O objetivo deste trabalho é investigar a geração de terrenos através de funções de ruído incrementadas por dados reais. Para isso, nas seções seguintes será estabelecido o arcabouço teórico que embasa o método, passando pelos conceitos envolvendo funções de ruído, geração procedural de terrenos e altimetria.

3.1 Mapas de altura

Em computação gráfica, um mapa de altura é uma imagem de duas dimensões que é utilizada para representar uma superfície ou terreno em um espaço tridimensional. Dessa forma, cada pixel da imagem representa uma região no espaço, ao qual é atribuído um valor, frequentemente entre 0 e 1, correspondente à altitude daquela região. Dessa forma, são definidos valores entre 0 e 1 para os pixels, sendo o valor 0 representado pela cor preta na tela, o valor 1 representado pela cor branca e os valores intermediários em escala de cinza. Os mapas de altura são largamente utilizados para criar cenas em jogos e outras aplicações gráficas como, por exemplo, animações.

Os valores para cada pixel do mapa de altura podem ser obtidos manualmente ou gerados automaticamente através de funções matemáticas e/ou algoritmos, como é o caso das funções de ruído. Essas funções criam valores que podem ser empregados para obter uma vasta gama de terrenos ou superfícies (LAMONTAGNE, 2022), incluindo montanhas, vales e platôs. A Figura 3.1 apresenta um mapa de alturas à esquerda, com a superfície tridimensional referente a este mapa à direita.

Além disso, o mapa de altura pode ser pós-processado através da aplicação de filtros, do uso de operações morfológicas (como erosão) (MUSGRAVE, 1993), ou da combinação de dados reais para criar uma aparência ainda mais natural e orgânica, como em Parberry (2014).

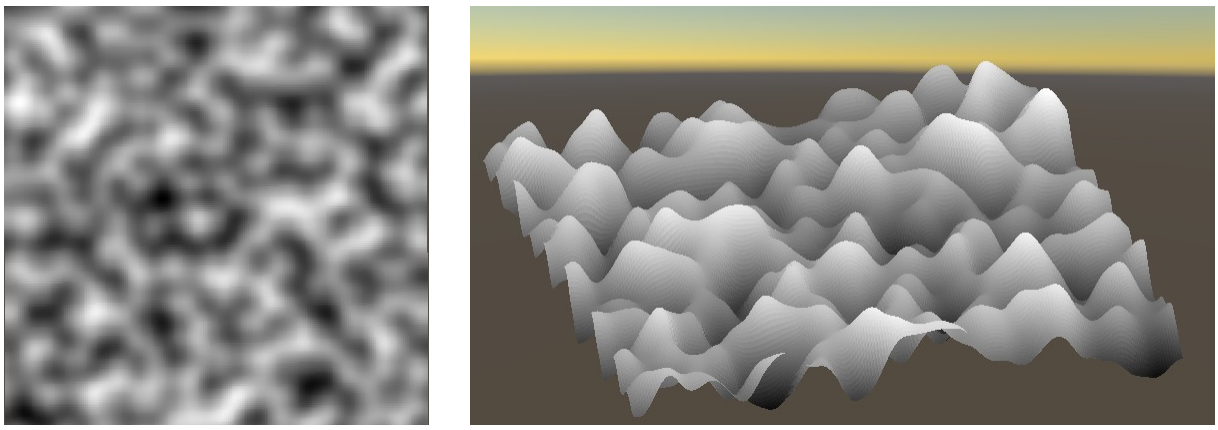

Figura 3.1: Mapa de altura (esquerda) que representa a superfície tridimensional à direita (Fonte: Gerado pelo autor)

3.2 Funções de ruído

De acordo com Perlin (1985), ruído é uma função escalar que recebe como argumento um vetor de n dimensões e possui como propriedades invariância estatística sob rotação e translação e um limite estreito de passagem de frequência. Em outras palavras, um ruído é incapaz de transmitir sinais fora de uma faixa de frequência predeterminada. Funções de ruído produzem agrupamentos pseudo-aleatórios que podem ser utilizados em diversas aplicações, pela característica de aparente imprevisibilidade, sendo adequadas para reproduzir padrões orgânicos, superfícies, texturas, dentre outras.

De acordo com Lagae et al. (2010), as funções de geração de ruído procedural possuem vantagens em relação a abordagens não-procedurais, devido ao fato de serem compactas em relação a imagens ou texturas pré-criadas, utilizando apenas poucos *kilobytes* de espaço comparadas aos *megabytes* de espaço em disco necessários para armazenar essas representações pré-existentes. Isso se deve ao fato de o ruído procedural ser gerado algoritmicamente, tornando-o mais eficiente e flexível. Além disso, as funções de ruído procedural são, de maneira geral, aperiódicas, podendo se estender indefinidamente e cobrir uma área de dimensões arbitrárias, sem emendas ou repetições aparentes. Outra vantagem é que as funções podem ser parametrizadas, tornando possível gerar uma classe de ruído. Dessa forma, o ruído pode ser controlado por um conjunto de parâmetros, que podem ser ajustados para criar diferentes variações do mesmo padrão de ruído. Ademais, funções de ruído procedural têm acesso aleatório, podendo ser calculadas em qualquer ponto do espaço, e o tempo de processamento é constante e independe da localização

do ponto a ser calculado. Ou seja, a função pode ser avaliada em qualquer ponto sem requerer o cálculo de outros pontos.

Existem diversos tipos de ruídos procedurais utilizados em computação gráfica, como ruído gaussiano, ruído de Perlin, ruído de Worley (WORLEY, 1996), ruído Simplex (PERLIN, 2001), dentre outros. Neste trabalho, serão abordados três dos diversos tipos de ruídos procedurais citados: Ruído de Valor, Ruído de Perlin e Ruído Simplex.

3.2.1 Ruído de Valor

Ruído de Valor, ou *Value Noise*, é um tipo de ruído procedural que é calculado através da atribuição de valores pseudo-aleatórios entre 0 e 1 nos vértices de uma grade regular. Os valores aleatórios podem ser gerados por vários métodos, porém o mais adotado é através de uma tabela de permutação que oferece um valor para cada vértice.

A partir da grade, o valor do ruído, ou seja, o valor no interior de cada célula é obtido através de interpolação bilinear dos quatro valores dos vértices em torno da célula. Na Figura 3.2, podemos observar os valores aleatórios em cada vértice e os valores interpolados para os pontos localizados no centro de cada célula, entre os respectivos vértices.

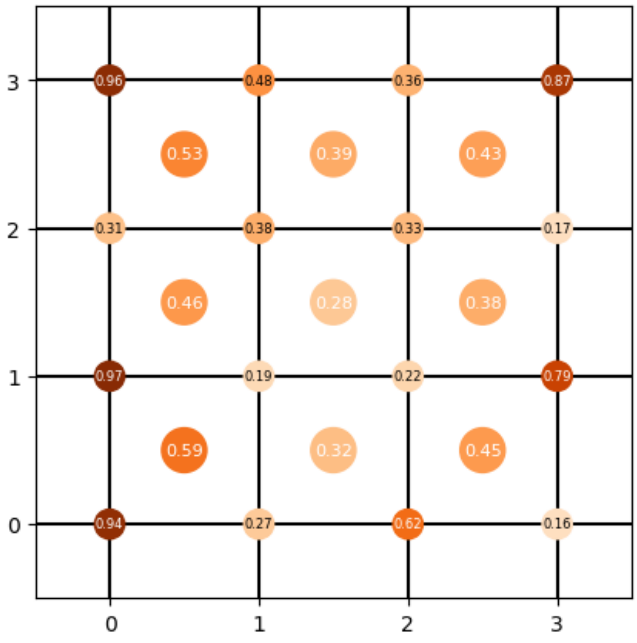

Figura 3.2: Grade com valores pseudo-aleatórios (Fonte: Elaborado pelo autor)

Por mais que o Ruído de Valor seja aparentemente aleatório, este tem uma aparência

com muitos artefatos e alta granularidade, como pode-se verificar na Figura 3.3, deixando clara a organização em grade utilizada. Assim, este ruído normalmente não é utilizado sozinho para fins de geração de mapas de altura ou texturas.

Figura 3.3: Mapa de altura gerado através do Ruído de Valor (Fonte: Gerado pelo autor)

3.2.2 Ruído de Perlin

O Ruído de Perlin, ou *Perlin Noise*, criado em 1985 por Ken Perlin (PERLIN, 1985), foi a primeira função de ruído gradiente. Além disso, essa técnica tem sido amplamente utilizada em diversas áreas, como em Computação Gráfica, simulações físicas, simulações em medicina (DUSTLER et al., 2015), visualização de dados (CONINX et al., 2011) e até mesmo em sintetização de músicas (POPOV, 2018). Dois mapas de alturas gerados através do Ruído de Perlin são mostrados na Figura 3.4.

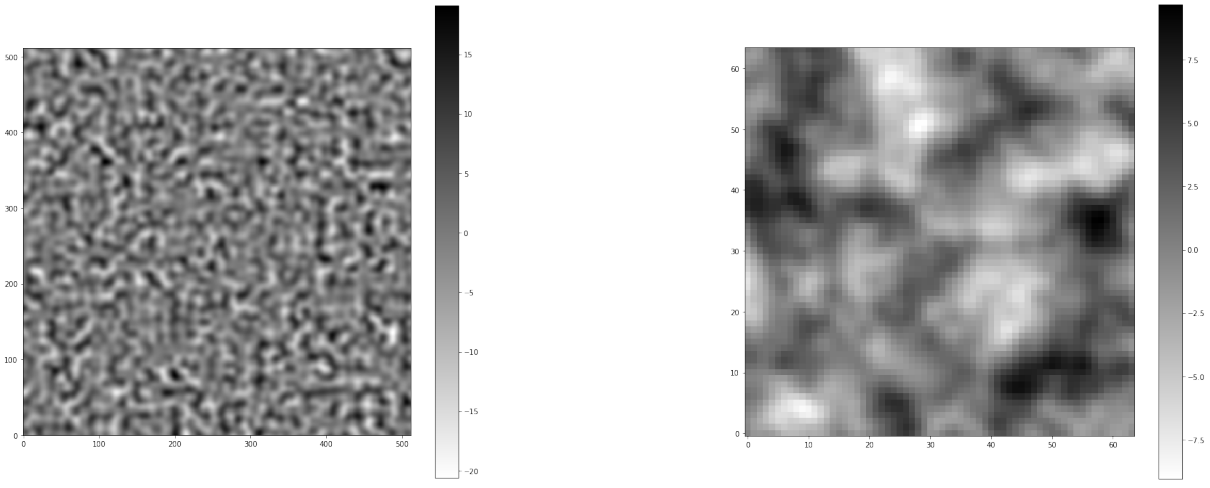

Figura 3.4: Mapas de altura gerados por Ruído de Perlin (Fonte: Gerado pelo autor)

De maneira geral, assim como no Ruído de Valor, o Ruído de Perlin envolve a geração de valores para os quatro vértices de cada célula que engloba um pixel da grade. Através de interpolações desses valores, determina-se o valor do ruído para cada ponto. Entretanto, o Ruído de Perlin se utiliza de vetores gradientes, em contraste com o Ruído de Valor, que usa valores escalares. O método pode ser dividido em três etapas, que são: a determinação da grade, o cálculo dos produtos escalares e a realização da interpolação.

Para a criação da grade, primeiro define-se a sua resolução. Para cada vértice, determina-se um vetor bidimensional aleatório. Para tal é utilizado um vetor de permutações, que consiste em uma lista aleatoriamente permutada de valores inteiros de 0 a 255 que é utilizada para selecionar os vetores gradiente na lista. No caso bidimensional, os vetores possíveis são: $(1,0)$, $(-1,0)$, $(0,1)$, $(0,-1)$, $(\sqrt{2},\sqrt{2})$, $(-\sqrt{2},\sqrt{2})$, $(\sqrt{2},-\sqrt{2})$ e $(-\sqrt{2},-\sqrt{2})$. A vantagem dessa abordagem em relação ao Ruído de Valor é uma maior variedade de valores para o ruído em cada ponto, com menos artefatos visíveis.

A determinação do vetor gradiente associado aos quatro vértices em torno do ponto segue o seguinte processo: primeiro determina-se as coordenadas do vértice inferior esquerdo da célula em que o ponto se encontra. Sendo (x, y) o ponto em análise, as coordenadas (X, Y) do vértice inferior esquerdo da célula são dadas pelo menor inteiro mais próximo às coordenadas do ponto, ou seja, $X = \lfloor x \rfloor$ e $Y = \lfloor y \rfloor$. Dessa forma, assumindo cada célula com dimensão 1, é possível obter as coordenadas dos outros vértices: $(X + 1, Y)$ (direito inferior), $(X, Y + 1)$ (esquerdo superior) e $(X + 1, Y + 1)$ (direito superior).

O vetor gradiente é, então, mapeado através da tabela de permutações. No Algoritmo 1, que mostra o processo de mapeamento, `grad` representa a lista de vetores gradientes, `gradMask` o tamanho da lista de gradientes (usualmente, 8), `perm` a tabela de permutações, `permMask` o tamanho da tabela de permutações (usualmente, 256) e g_{00}, g_{01}, g_{10} e g_{11} os vetores gradientes atribuídos a cada vértice da célula.

Algoritmo 1 Processo de atribuição dos vetores gradientes

$g_{00} \leftarrow \texttt{grad}[(X + \texttt{perm}[Y])\%\texttt{gradMask}]$
$g_{01} \leftarrow \texttt{grad}[(X + \texttt{perm}[Y + 1])\%\texttt{gradMask}]$
$g_{10} \leftarrow \texttt{grad}[(X + 1 + \texttt{perm}[Y])\%\texttt{gradMask}]$
$g_{11} \leftarrow \texttt{grad}[(X + 1 + \texttt{perm}[Y + 1])\%\texttt{gradMask}]$

Para determinar os produtos escalares, primeiro determina-se os vetores distância do ponto em questão até os quatro vértices mais próximos, ilustrados na Figura 3.5. Em seguida, calcula-se o produto escalar entre os vetores distância e os vetores gradientes mapeados aos vértices correspondentes. Esses produtos escalares atrelados a cada vértice correspondente serão, então, interpolados.

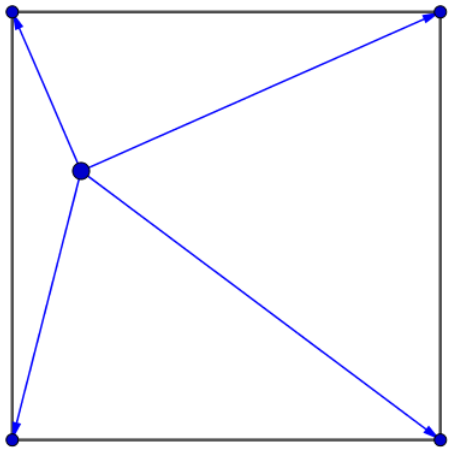

Figura 3.5: Vetores distância de uma célula (Fonte: Elaborado pelo autor)

Com os valores parciais para cada vértice da célula que contém o ponto estudado, realiza-se a interpolação entre os valores obtidos. Para tal, utiliza-se alguma função de uma família denominada *smoothstep* (EBERT et al., 2003), que possua primeira derivada nula em seus pontos extremos ($x = 0$ e $x = 1$) e ofereça um perfil que possibilite uma transição gradual entre valores. Na implementação original (PERLIN, 1985), Ken Perlin utiliza a seguinte equação para o cálculo da interpolação:

$$smoothstep(x) = 3x^2 - 2x^3$$

Porém, em um estudo posterior (PERLIN, 2002), a função *smoothstep* utilizada na interpolação é alterada para uma com grau 5, que apresenta menos artefatos visuais:

$$smoothstep(x) = 6x^5 - 15x^4 + 10x^3.$$

Como podemos visualizar na Figura 3.6, a função *smoothstep* possui um perfil em S, proporcionando uma interpolação mais suave quando comparada à interpolação linear.

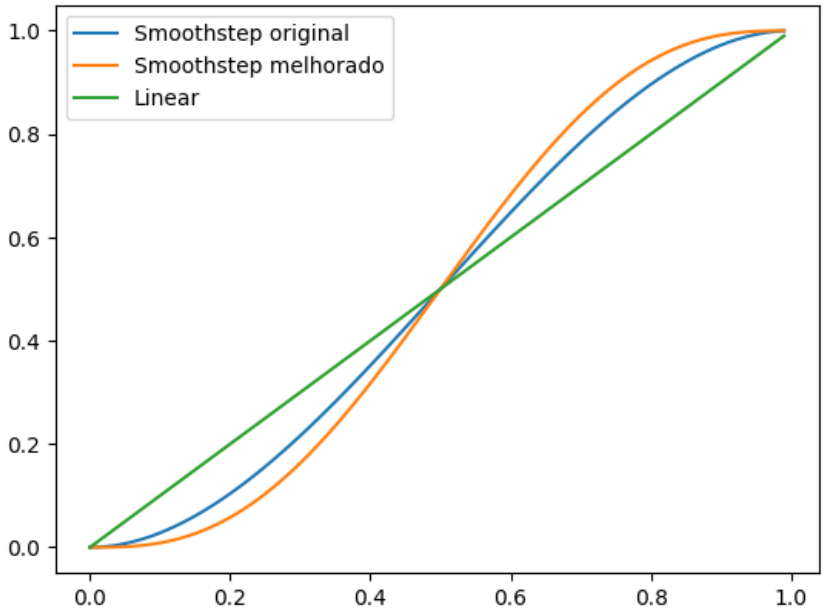

Figura 3.6: Funções de interpolação *smoothstep* e linear (Fonte: Elaborado pelo autor)

A função *smoothstep* tem como argumento um valor x em um intervalo e retorna um valor dentro deste mesmo intervalo. De forma a obter o valor do pixel, é realizada uma interpolação semelhante à bilinear. Para tal, primeiro são realizadas duas interpolações utilizando *smoothstep*. Essas interpolações ocorrem em uma dimensão entre dois pares de vértices que compartilham a mesma coordenada em algum eixo. A distância entre os vértices e o ponto em questão é utilizada como peso. Por fim, esses dois valores obtidos são combinados com outra interpolação, também utilizando *smoothstep*, obtendo o valor do Ruído de Perlin para o ponto de interesse.

A Figura 3.7 representa uma célula e demonstra o mecanismo das duas primeiras interpolações para o ponto P: os valores correspondentes de produto escalar nos pontos A e B são interpolados utilizando *smoothstep* para o ponto X_1, utilizando i_A como peso para A e i_B como peso para B. De maneira análoga, os valores correspondentes de produto escalar nos pontos C e D são interpolados utilizando *smoothstep* para o ponto X_2, utilizando i_A como peso para C e i_B como peso para D.

Após obter os valores interpolados em X_1 e X_2, realiza-se uma nova interpolação entre esses dois pontos, utilizando as distâncias deles até o ponto P como pesos. Como

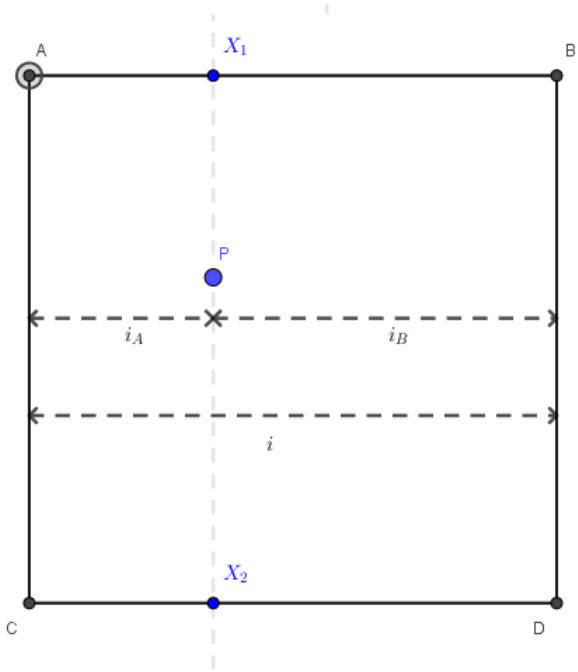

Figura 3.7: As duas primeiras interpolações em uma célula de Ruído de Perlin (Fonte: Elaborado pelo autor)

mostrado na Figura 3.8, essas distâncias são representadas por j_A para o ponto X_1 e j_B para o ponto X_2. Essa segunda interpolação tem como objetivo calcular o valor final do ruído para o ponto P, levando em consideração a influência dos pontos vizinhos.

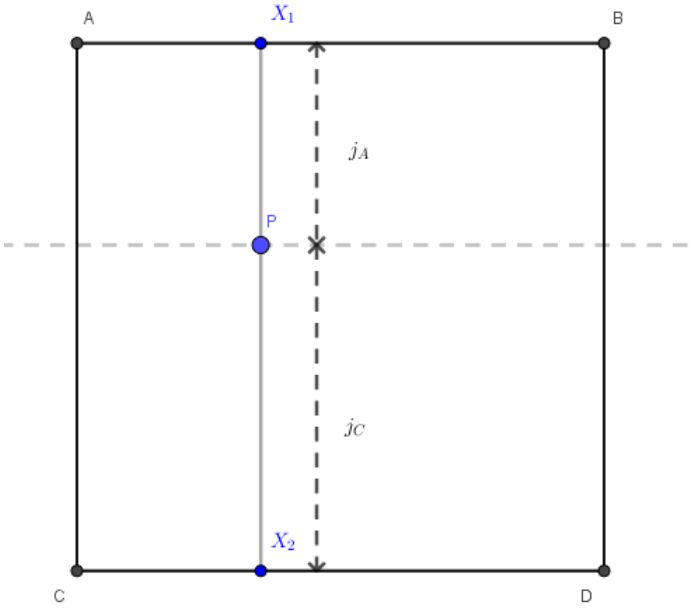

Figura 3.8: A terceira interpolação em uma célula de Ruído de Perlin (Fonte: Elaborado pelo autor)

Apesar de sua popularidade, existem limitações neste algoritmo. Uma delas é a

tendência de produzir padrões regulares e repetitivos. Para lidar com essa limitação, diversas variações do Ruído de Perlin foram desenvolvidas, como o Ruído de Worley e o Ruído Simplex. Essas variações se utilizam de estruturas e métodos de interpolação mais complexas, que mascaram esses padrões visíveis.

3.2.3 Ruído Simplex

Ruído Simplex, ou *Simplex Noise*, é uma função de ruído procedural criada por Ken Perlin em Perlin (2001) como um aperfeiçoamento do Ruído de Perlin, gerando resultados similares, como pode-se observar na Figura 3.9.

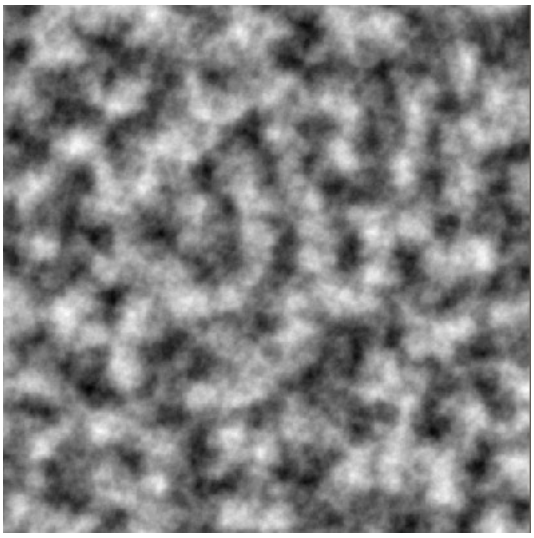

Figura 3.9: Mapa de altura gerado pelo Ruído Simplex (Fonte: Gerado pelo autor)

Ao comparar com o Ruído de Perlin, o Ruído Simplex possui menor complexidade computacional, em virtude de dividir o espaço em formas otimizadas para menos vértices (triângulos em duas dimensões, tetraedros em três dimensões), realizando menos multiplicações e permitindo escalar para dimensões maiores com um custo computacional muito menor. Além disso, o Ruído Simplex possui menos artefatos direcionais, ou seja, não apresenta padrões visíveis, linhas ou divisões em seu resultado final, tendo aparência mais suave. Apesar de todas as vantagens explicitadas, o Ruído Simplex não é tão largamente utilizado como o Ruído de Perlin, principalmente pela sua implementação complexa e pela patente que existia até o ano de 2022.

Assim como no Ruído de Perlin, o primeiro passo para gerar o Ruído Simplex é a criação de uma grade. Porém, o Ruído Simplex não utiliza uma grade feita de quadrados,

mas uma grade composta por simplexos. Um simplexo é uma generalização do triângulo para qualquer dimensão. Essa escolha de formato de grade traz algumas vantagens em relação à grade quadrada, preenchendo o espaço de maneira mais eficiente. Isso resulta em uma menor quantidade de vértices na grade, o que representa um ponto positivo, já que a cada vértice uma operação é realizada.

Para uma dimensão, a única forma possível é uma linha que preenche um intervalo de tamanho igual por todo o espaço. Em duas dimensões, a forma utilizada na grade simplexo para cobrir a superfície é um triângulo equilátero que, ao ser justaposto com outros, forma um padrão que pode se repetir infinitamente em todas as direções, como demonstrado na Figura 3.10. Ao unir dois destes triângulos, um losango é formado, podendo ser interpretado como um quadrado que foi encolhido ao longo de sua diagonal principal. De maneira análoga, em três dimensões, a forma geométrica espacial é um tetraedro, que quando é combinado a outros cinco tetraedros, forma um cubo encolhido em sua diagonal.

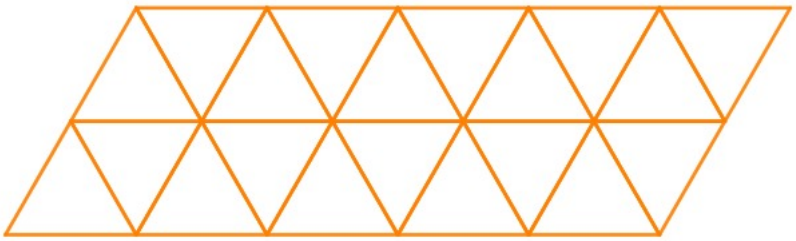

Figura 3.10: Espaço preenchido por um padrão de triângulos justapostos (Fonte: Elaborado pelo autor)

Essa ideia pode ser expandida para dimensões superiores, sendo que, quanto maior a dimensão, maior a vantagem da utilização da tesselação por simplexos quando comparada a hipercubos, pois, em um espaço n-dimensional, formas simplexo possuem $n + 1$ vértices, enquanto hipercubos possuem 2^n vértices. Dessa forma, com o crescimento de n, o problema de realizar a interpolação em hipercubos se torna muito mais custoso computacionalmente, por ter ordem de complexidade computacional $O(2^n)$, enquanto na grade simplexo, essa tarefa tem ordem de complexidade computacional $O(n^2)$ (PERLIN, 2001).

No caso 2D, cada ponto P que se encontra em um triângulo recebe contribuições

para o valor do ruído apenas das três extremidades mais próximas. Ao contrário do Ruído de Perlin, que utiliza interpolações sucessivas para o cálculo do ruído em um ponto, o Ruído Simplex realiza apenas uma soma direta da contribuição dos vértices mais próximos, que são atenuados de acordo com a distância utilizando uma função de decaimento radial independente entre os vértices.

Uma grade de simplexos pode ser obtida ao transformar uma grade de quadrados por um processo que pode ser visto como uma deformação nos dois eixos. Esse processo consiste em dividir cada quadrado em dois triângulos retângulos isóceles. Com isso, procura-se obter as coordenadas relativas aos triângulos equiláteros obtidos, o que é realizado através da translação dos pontos paralelamente, na direção da diagonal principal de cada quadrado.

De acordo com a Figura 3.11, e assumindo a coordenada do ponto fixo sendo $(0, 0)$ e dos que não pertencem à diagonal $(0, 1)$ e $(1, 0)$, temos que, para um deslocamento em cada eixo de S dos pontos B e D, ocorre um deslocamento de $2S$ do ponto C. Dessa forma, os pontos B e D são deslocados para $B' = (1-S, -S)$ e $D' = (-S, 1-S)$, respectivamente, enquanto o ponto C é deslocado para $C' = (1 - 2S, 1 - 2S)$.

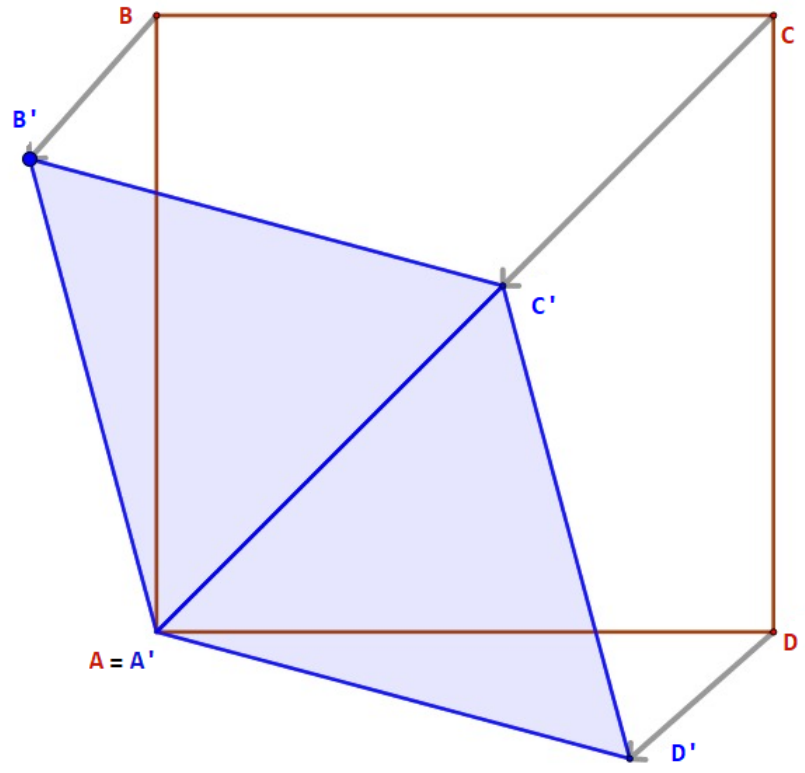

Figura 3.11: Transformação de malha quadrada para simplexo através de deformação (Fonte: Elaborado pelo autor)

Considerando o fato de o triângulo formado ser equilátero, é possível construir uma equação a partir das novas coordenadas obtidas:

$$d(B', C') = d(A', C'). \tag{3.1}$$

Ao resolver a Equação 3.1, encontra-se $S = \dfrac{3 - \sqrt{3}}{6}$ para o fator de conversão de malha quadrada para a malha triangular. Analogamente, no caso da conversão da malha triangular para a malha quadrada, utiliza-se a constante $U = \dfrac{\sqrt{3} - 1}{2}$. Esse valor é utilizado para determinar os pontos na grade quadrada correspondentes aos pontos de amostra na grade triangular.

Como uma célula quadrada em coordenadas cartesianas dá origem a duas células triangulares (Figura 3.11), deve-se primeiro definir em qual triângulo o ponto se encontra (superior ou inferior). Com esse objetivo, determina-se as coordenadas do ponto $P(x, y)$ (o ponto em que se deseja calcular o valor do ruído) nas coordenadas relativas à malha simplicial. Assim, calcula-se um fator de conversão da malha quadrada para simplexo $s = (x + y) \cdot S$. As coordenadas do ponto na nova malha são dadas por:

$$x_s = x + s, \tag{3.2}$$

$$y_s = y + s. \tag{3.3}$$

Em seguida, define-se a origem de um simplexo como o ponto com coordenadas $(0, 0)$ fixo na transformação entre célula quadrada para célula simplexo. Na Figura 3.11, a origem do simplexo é representada pelo ponto A, que é congruente ao ponto A'.

Dessa forma, podemos determinar as coordenadas da origem do simplexo no qual o ponto está contido, também em coordenadas relativas à malha simplex. Sejam essas coordenadas dadas por (i, j), temos:

$$i = \lfloor x_s \rfloor, \tag{3.4}$$

$$j = \lfloor y_s \rfloor. \tag{3.5}$$

De posse das coordenadas da origem da célula simplexo que contém o ponto P, pode-se

calcular as distâncias até a origem nos dois eixos e compará-las, a fim de determinar em qual triângulo o ponto P se encontra. Porém, para se utilizar do método tradicional de cálculo de distâncias, as coordenadas devem ser cartesianas. Por esse motivo, a coordenada da origem da célula simplexo é convertida para coordenadas cartesianas através de um fator de conversão $t = (i + j) \cdot U$. Aplicando o fator de conversão, a origem da célula simplexo equivale, em coordenadas cartesianas, ao ponto (X_0, Y_0) dado por

$$X_0 = i - t,$$
$$Y_0 = j - t.$$

Agora, calcula-se as distâncias da origem da célula simplexo até o ponto P, em coordenadas cartesianas:

$$d_{x_0} = x - X_0,$$
$$d_{y_0} = y - Y_0. \tag{3.6}$$

Conforme representado na Figura 3.11, caso d_{x_0} tenha valor maior que d_{y_0}, ou seja, o ponto esteja mais próximo do ponto D' que do ponto B', este se encontra no triângulo inferior, caso contrário, o ponto se encontra no triângulo superior.

Em coordenadas da malha simplex e em relação à origem da célula, os triângulos superior e inferior compartilham os vértices de coordenadas $(0, 0)$ (origem) e $(1, 1)$, como pode ser visualizado na Figura 3.12. Seja o vértice restante o ponto com coordenadas (i_1, j_1), temos que se o triângulo que contém o ponto for o triângulo inferior,

$$i_1 = 1, \tag{3.7}$$
$$j_1 = 0, \tag{3.8}$$

e se o triângulo que contém o ponto for o triângulo superior,

$$i_1 = 0, \tag{3.9}$$
$$j_1 = 1. \tag{3.10}$$

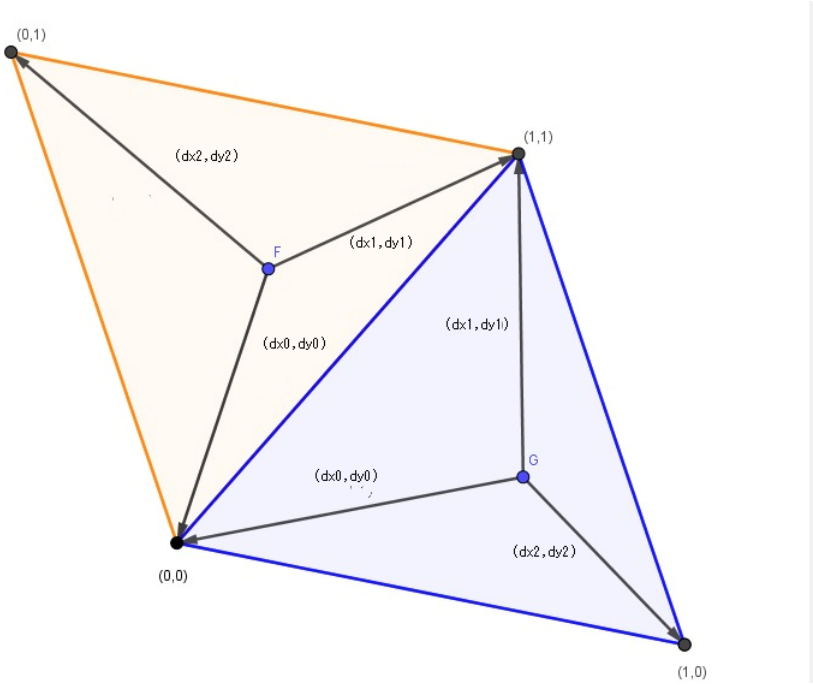

Figura 3.12: Simplexos provenientes do mesmo quadrado e distâncias aos respectivos vértices (Fonte: Elaborado pelo autor)

Como a contribuição de cada vértice do triângulo para o ruído final depende da distância até o ponto P, calcula-se essas distâncias com base nas coordenadas cartesianas. Define-se as distâncias do ponto dentro do simplexo até os pontos $(0, 0)$ (origem), $(1, 1)$ e (i_1, j_1) por (d_{x_0}, d_{y_0}) (já definidos na Equação 3.6), (d_{x_1}, d_{y_1}) e (d_{x_2}, d_{y_2}), respectivamente, assim como se observa na Figura 3.12. Portanto, temos:

$$d_{x_1} = d_{x_0} - i_1 + U \tag{3.11}$$

$$d_{y_1} = d_{y_0} - j_1 + U \tag{3.12}$$

$$d_{x_2} = d_{x_0} - 1 + 2U \tag{3.13}$$

$$d_{y_2} = d_{y_0} - 1 + 2U. \tag{3.14}$$

O Ruído Simplex também é denominado um ruído gradiente pois realiza uma operação de produto escalar com um vetor gradiente unitário para diminuir a previsibilidade direcional do resultado final. Esse vetor é usualmente escolhido dentre um conjunto de vetores unitários.

Uma implementação comum para a escolha do vetor gradiente leva em conta a pré-existência de uma tabela de permutação, que é comumente composta de 256 ou 512

números inteiros diferentes em uma ordem aleatória. Cada vértice da malha simplex é mapeado em um gradiente, normalmente por uma função de *hashing*, da mesma maneira que é descrita no Algoritimo 1. Portanto, para os três vértices de uma célula, três vetores unitários são selecionados: $\vec{G_0}, \vec{G_1}, \vec{G_2}$.

Assim, pode-se obter a magnitude da contribuição de cada vértice do triângulo no qual o ponto está através de:

$$v_0 = (0.5 - d_{x_0} - d_{y_0})^4.((d_{x_0}, d_{y_0}) \cdot \vec{G_0}) \tag{3.15}$$

$$v_1 = (0.5 - d_{x_1} - d_{y_1})^4.((d_{x_1}, d_{y_1}) \cdot \vec{G_1}) \tag{3.16}$$

$$v_2 = (0.5 - d_{x_2} - d_{y_2})^4.((d_{x_2}, d_{y_2}) \cdot \vec{G_2}). \tag{3.17}$$

Por fim, soma-se a contribuição de cada vértice para determinar o valor do ruído no ponto que está sendo analisado, obtendo-se o valor do ruído final:

$$v = v_0 + v_1 + v_2. \tag{3.18}$$

3.2.4 Movimento Browniano Fractal

Em Computação Gráfica, é prática comum a utilização da técnica do Movimento Browniano Fractal (MBF), que consiste na sobreposição de várias camadas de ruído com o objetivo de obter um ruído com maior detalhamento. Através da utilização dessa técnica, é possível obter resultados extremamente detalhados, que seriam difíceis ou impossíveis de se obter através do ruído convencional.

O MBF é altamente personalizável, permitindo que a aparência final do ruído seja adequada ao interesse do usuário. Essa personalização é dada através do ajuste de parâmetros que controlam a mudança de aparência entre as camadas sucessivas do ruído, que são somadas. Essas camadas são denominadas oitavas e possuem valores diferentes entre si de frequência e amplitude. A frequência determina o quão rápido o ruído varia, e a amplitude o quão significantes os valores de ruído da a oitava são para a composição do ruído final. A mudança de frequência e amplitude entre as oitavas é controlada por parâmetros denominados persistência e lacunaridade.

A persistência é utilizada para modular a magnitude da variação de amplitude entre as oitavas. De maneira visual, a persistência controla o contraste apresentado no mapa de altura. Por exemplo, se o valor de persistência é $0,5$, cada oitava sucessiva terá amplitude correspondente à metade da anterior. De maneira análoga, a lacunaridade determina a variação de frequência entre oitavas sucessivas. Dessa forma, quanto maior o valor da lacunaridade, maior a granularidade do ruído, deixando mais evidente a estrutura base deste.

A Figura 3.13 apresenta mapas de altura gerados através de 4 oitavas de Ruído Simplex com o parâmetro de persistência fixado no valor $0,5$, e para diferentes valores de lacunaridade: $0,25$, $0,5$ e $1,0$, da esquerda para a direita. É possível observar que, para o mesmo valor de persistência, com a diminuição do valor de lacunaridade o mapa de altura apresenta maior granularidade.

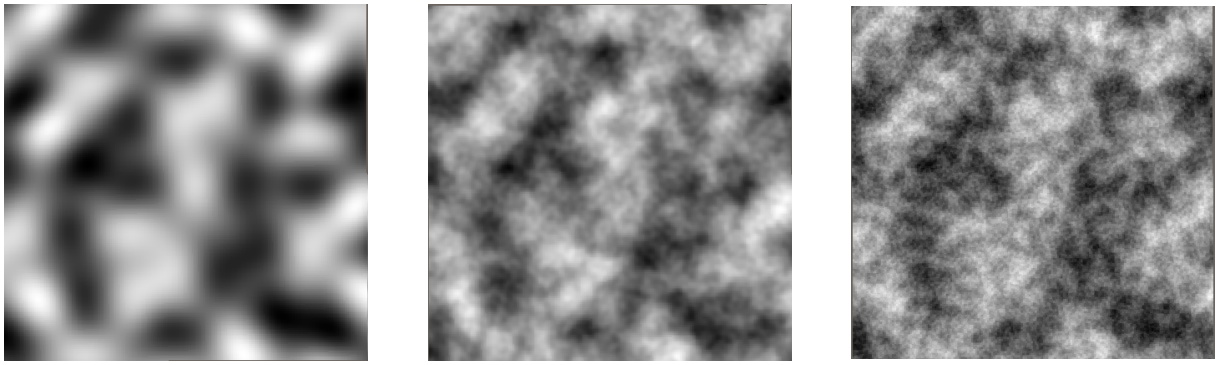

Figura 3.13: Mapas de altura gerados através de 4 oitavas de Ruído Simplex com $0,5$ de persistência e $0,25$, $0,5$ e $1,0$ de lacunaridade, da esquerda para a direita. (Fonte: Gerado pelo autor)

Por outro lado, a Figura 3.14 apresenta mapas de altura gerados através de 4 oitavas de Ruído Simplex com o parâmetro de lacunaridade fixado em $2,6$ e o parâmetro de persistência assumindo os valores $0,25$, $0,5$ e $0,75$, da esquerda para a direita. É possível observar que, para o mesmo valor de lacunaridade, a diminuição do valor de persistência acarreta maior suavidade no mapa de alturas, pelo fato das oitavas posteriores contribuírem pouco para a aparência final do ruído.

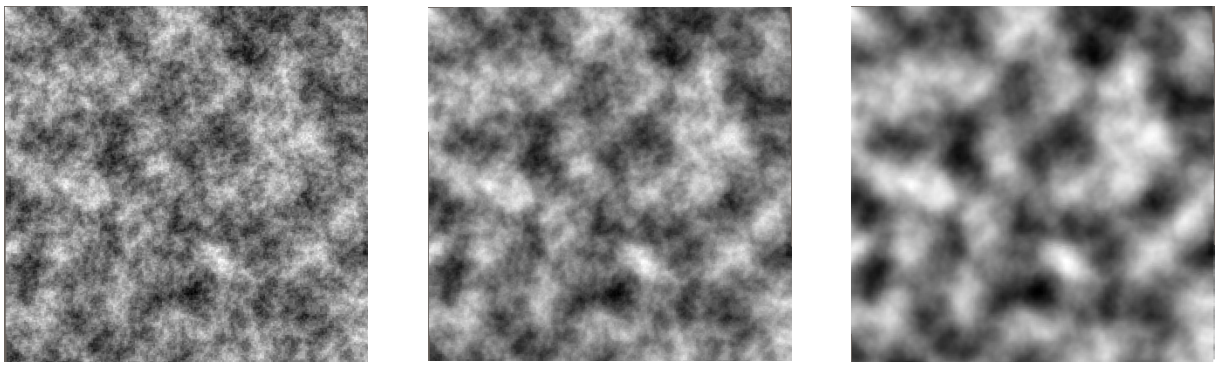

Figura 3.14: Mapas de altura gerados através de 4 oitavas de Ruído Simplex com $2,6$ de lacunaridade e persistência com os valores $0,25$, $0,5$ e $0,75$, da esquerda para a direita (Fonte: Gerado pelo autor)

3.3 Altimetria

Em Geografia, Topografia é o estudo da forma e característica das superfícies terrestres. De acordo com ESPARTEL (1987), o objetivo principal da Topografia é estabelecer a configuração, tamanho e posição relativa de uma área especifica da superfície terrestre, considerando apenas sua porção limitada, sem levar em consideração a curvatura da Terra. A Topografia se divide em Topometria e Topologia, sendo a primeira dedicada a estudar processos de medição de distâncias, ângulos e desníveis, e a última dedicada ao estudo das formas exteriores e das regras que regem sua forma, assim como explicitado em Veiga, Zanetti e Faggion (2012). Por sua vez, a Topometria também possui duas subdivisões: planimetria e altimetria, sendo aquela dedicada a definir a posição bidimensional dos pontos terrestres, e esta com enfoque em determinar a altitude de um ponto na superfície terrestre.

As informações sobre altimetria e planimetria usualmente são condensadas em um artefato denominado carta topográfica. De maneira tradicional, utiliza-se sensoriamento remoto para a produção destas. De acordo com Oliveira (2005), na segunda metade do século XX, essa produção era feita principalmente através de fotos aéreas, sendo substituídas quase completamente na última década do século XX pela utilização de sensores orbitais na extração de dados planimétricos. Apenas a partir de 1999, com a utilização de radares de abertura sintética (SAR), os instrumentos de medição começaram a possuir tecnologia para obter dados tridimensionais, sendo utilizados para extração de dados de

altimetria. Em 22 de fevereiro de 2000, a *National Aeronautics and Space Administration* (NASA), em conjunto com a *National Imagery and Mapping Agency* (NIMA), realizou a missão espacial denominada *Shuttle Radar Topography Mission* (SRTM) com objetivo de obter dados topográficos precisos da superfície da Terra (FARR; KOBRICK, 2000). Desde então, os dados obtidos pela SRTM são os mais utilizados para derivar modelos de elevação.

Um Modelo Digital de Elevações (MDE) é uma representação matricial numérica do relevo de certa localidade. Essa matriz é composta pelos valores de elevação de pontos igualmente espaçados na superfície terrestre. Outra definição para o MDE é apresentada por Paradella et al. (2001) como uma representação digital de uma parte da superfície, por meio de uma matriz de pixels com coordenadas planimétricas (x, y) e valor de intensidade de pixel correspondendo à elevação. Essa definição é muito semelhante à definição de mapa de altura tratada na Seção 3.1. Dessa forma, podemos entender o MDE como um mapa de altura para representar a topometria de uma seção da superfície terrestre.

Com base nos dados do SRTP, em 2008, foi lançado pelo Instituto Nacional de Pesquisas Espaciais (INPE) o Topodata, que consiste em um MDE para toda a cobertura do território brasileiro (VALERIANO, 2008). De acordo com Valeriano (2008), o Topodata foi concebido com o objetivo de calcular e disponibilizar variáveis geomorfométricas locais, como declividade, orientação de vertentes, curvatura horizontal, curvatura vertical e insumos para o delineamento da estrutura de drenagem, visando facilitar o uso dessas informações pela comunidade científica.

Dados de altimetria podem ser um recurso valioso para geração procedural de terrenos pois oferecem subterfúgios para a análise da topologia de um local, que pode ser parametrizada para gerar computacionalmente terrenos realistas.

3.4 Utilização da altimetria na Geração Procedural de Terrenos

Os desenvolvedores de jogos, por exemplo, frequentemente têm o desejo de criar terrenos baseados em locais reais. Tradicionalmente, esse processo envolve ajustes ma-

nuais para tentar aproximar o terreno ao visual desejado ou o uso de parâmetros para a geração procedural, que nem sempre garantem precisão em reproduzir fielmente as características desejadas. No entanto, uma maneira mais eficiente é usar dados reais, como altimetria, para conferir ao terreno características de um local específico, possibilitando aos jogadores explorar e interagir com ambientes familiares e realistas.

Em Parberry (2014) é apresentado um método para enviesar o resultado do Ruído de Valor de modo que este tome características da altimetria[3]. Para tal, é criado um histograma dos valores normalizados de altitude encontrados no local estudado. Esse histograma representa a frequência de ocorrência de diferentes faixas de altitude. Com base nesse histograma, é gerada uma tabela de frequências relacionada, na qual os grupos de frequência são ajustados para que a soma total seja igual a 256. A partir dessa tabela de frequências, é criada uma lista com 256 posições preenchidas com valores crescentes aleatórios entre -1 e 1. Essa lista é gerada de forma que os elementos mais frequentes no histograma possuam mais ocorrências relacionadas na lista.

O Algoritmo 2 apresenta o processo para gerar a lista de números aleatórios, denominada `listaValores`. O algoritmo recebe como parâmetros `POINTCOUNT`, que é a quantidade de casas utilizada na tabela de frequências, além da tabela de frequência em si, que é denominada `distribuicao`. Além disso, o método utiliza uma função para gerar valores aleatórios entre 0 e 1, exibida com a assinatura `gerarNumeroAleatorio`.

Algoritmo 2 Processo de geração dos números aleatórios crescentes

```
δ ← 2.0/(POINTCOUNT − 1)
min ← −1.0
k ← 0
for i ← 0 to POINTCOUNT − 1 do
    for j ← 0 to distribuicao[i] − 1 do
        listaValores[k] ← min + δ × gerarNumeroAleatorio()
        k ← k + 1
    end for
    min ← min + δ
end for
```

No método, é utilizado o Ruído de Valor, no qual os vértices da grade são mapeados para esse vetor de 256 posições, que efetivamente oferece os valores aleatórios necessários para o cálculo do ruído que incorpora as características de altimetria específicas da região.

[3]Código-fonte disponível em ⟨https://github.com/Ian-Parberry/DesignerWorlds/⟩

4 Desenvolvimento

Este capítulo tem como objetivo apresentar um método empregado na geração procedural de terrenos utilizando funções de ruído, que permite criar uma paisagem complexa de forma controlada. É introduzida uma abordagem baseada na técnica proposta por Parberry (2014) para gerar terrenos utilizando dados de MDEs. Esse método combina a análise de características do MDE com a aplicação de operadores específicos para simular a formação de terrenos complexos a fim de emular características de relevo real. Ao compreender esses métodos, é possível aplicá-los para criar terrenos virtuais convincentes.

4.1 Geração Procedural de Terrenos

A geração procedural de terrenos é uma das aplicações mais comuns e mais úteis dos algoritmos de ruído, principalmente no contexto de jogos. Esses algoritmos permitem criar terrenos realistas, detalhados, com aparência orgânica e que podem ser utilizados em vários tipos de jogos, como por exemplo de aventura ou jogos de estratégia.

Nessas aplicações, os algoritmos de geração procedural se sobressaem em relação às técnicas de geração manual de terrenos, uma vez que possibilitam a criação de terrenos complexos de forma automática e eficiente.

Neste trabalho, são usados mapas de altura gerados por algoritmos de ruído para a geração de ambientes tridimensionais, a fim de aplicar as técnicas de ruído procedural abordadas. A geração pode ser dividida em três partes: geração do mapa de altura, definição das regiões e criação da malha.

4.1.1 Geração do mapa de altura

Como primeiro passo, deve-se gerar um mapa de altura nas dimensões desejadas. O mapa de altura tem seus parâmetros de persistência, lacunaridade, escala e quantidade de oitavas para formar um mapa de altura final. Como exemplo, temos o mapa de altura da Figura 4.1, que foi gerado utilizando o Ruído de Perlin.

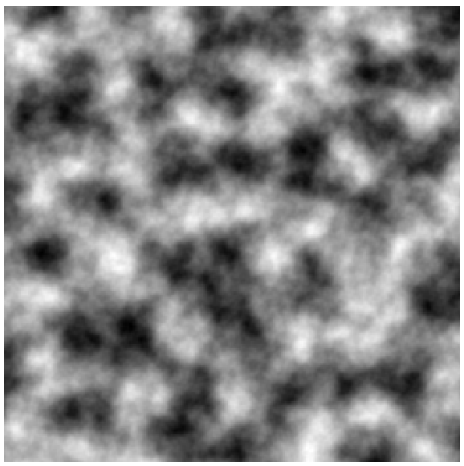

Figura 4.1: Exemplo de mapa de altura utilizado para gerar um terreno (Fonte: Gerado pelo autor)

4.1.2 Geração do mapa de cores

Após a geração do mapa de alturas, esse mapa é utilizado para gerar um mapa de cores, sendo que cada cor representa uma região do mapa a ser representada, por exemplo: a cor azul representa uma região de água, a cor verde uma região de vegetação, a cor marrom uma região de rochas e a cor branca uma região nevada. Esse mapa de cores adiciona diversidade visual e melhora a representação de características do terreno.

No caso da implementação realizada, o único critério para determinar a região de cor que cada ponto ocupa é o valor do mapa de altura naquele ponto. Estes intervalos são fixos para qualquer terreno gerado. A Figura 4.2 apresenta um mapa de cores gerado a partir do mapa de altura mostrado na Figura 4.1.

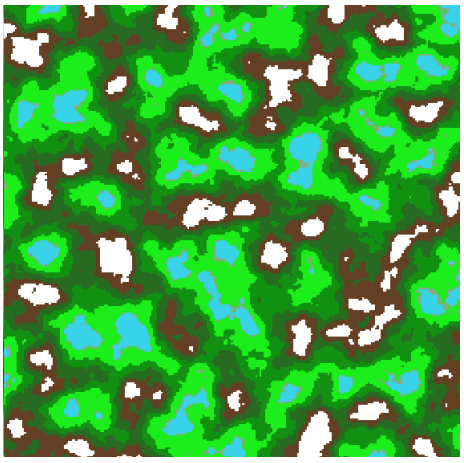

Figura 4.2: Mapa de cores gerado a partir de um mapa de altura (Fonte: Gerado pelo autor)

4.1.3 Criação da malha

Depois da geração do mapa de altura e do mapa de cores, a malha é criada através de ferramentas presentes no motor de jogos Unity. Uma malha consiste em uma coleção de vértices e faces, que no caso são triangulares, que definem a topologia do terreno a ser gerado. Em suma, partindo-se do plano obtido anteriormente, a malha que representa o terreno é obtida através da iteração em cada ponto definindo a altura de cada vértice como o valor do pixel no mapa de altura naquele ponto.

Como primero passo, deve-se converter de um mapa bidimensional para uma malha tridimensional. É utilizada uma malha triangular simples, como apresentado em Ebert et al. (2003), que consiste em mapear todo o plano com triângulos retângulos equiláteros justapostos, assim como mostrado na Figura 4.3. A utilização de uma malha triangular traz várias vantagens, além da praticidade de implementação: os triângulos garantem a coplanaridade de todos os vértices, característica que é computacionalmente custosa para se verificar.

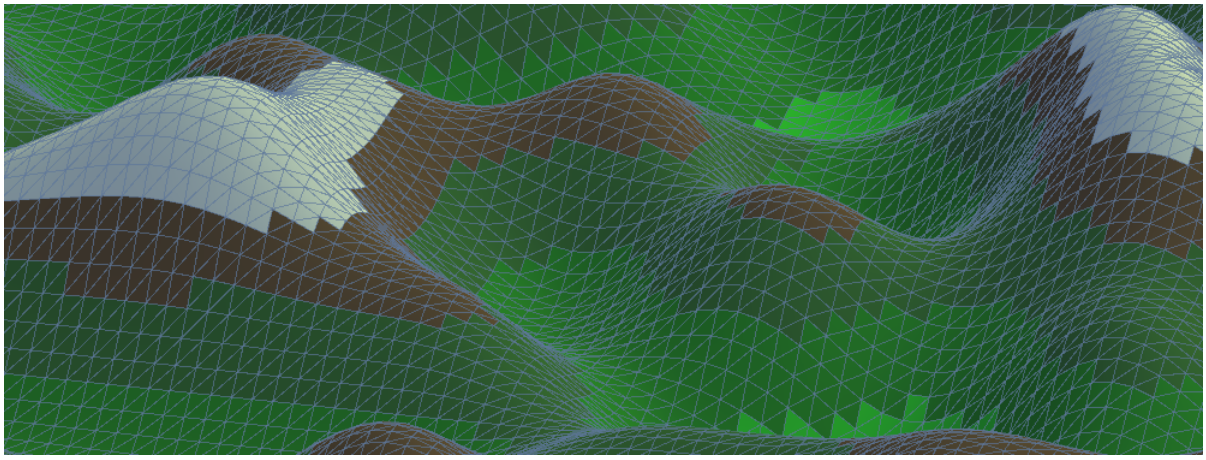

Figura 4.3: Malha triangular sendo utilizada para formar terreno 3D (Fonte: Gerado pelo autor)

Para realizar a conversão, calcula-se a posição de cada vértice, oferecendo as três coordenadas espaciais como entrada, sendo duas delas as mesmas coordenadas correspondentes à superfície do mapa de cores, com a terceira coordenada sendo equivalente ao valor do ruído calculado naquele ponto. Dessa forma, obtém-se uma superfície tridimensional com alturas entre 0 e 1. Como o terreno gerado costuma ter dimensões no plano xy pelo menos na ordem de 10^2 e o valor de ruído gerado tem valores entre 0 e 1, torna-se

necessário escalar o resultado final, multiplicando-se o valor da altura em cada ponto por um parâmetro fixo hm, com o intuito de o terreno adquirir um aspecto mais realista.

Assim, um ponto da malha pode ser definido por:

$$V = (x, y, Noise(x, y).hm). \tag{4.1}$$

Após esse processo, são calculadas as normais de cada triângulo da malha, que consistem no vetor normal à superfície que indica o lado no qual a textura deve ser aplicada. Com as normais calculadas, aplica-se a cor correspondente à coordenada (x, y) ao triângulo da malha. A Figura 4.4 apresenta a superfície 3D gerada através da combinação do mapa de alturas mostrado na Figura 4.1 e do mapa de cores da Figura 4.2.

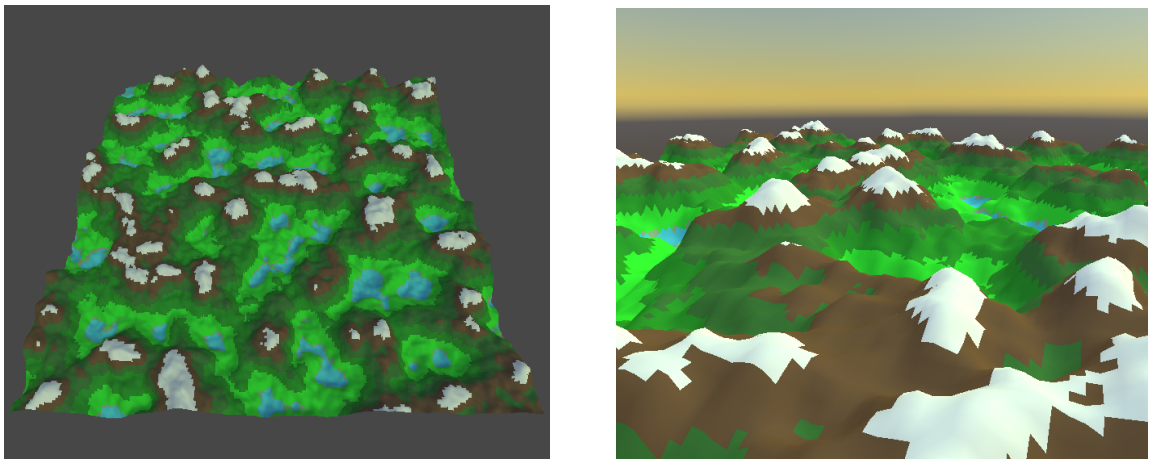

Figura 4.4: Terreno procedural gerado a partir de função de ruído (Fonte: Gerado pelo autor)

4.2 Incorporação da Altimetria no Ruído Simplex

O Ruído Simplex, assim como o Ruído de Perlin, tem como característica a geração de superfícies relativamente uniformes e indistintas, como apontado em Parberry (2014). Pode ser de interesse a geração de terrenos com características de alguma localidade e, pelos métodos tradicionais, como mostrado em Smelik et al. (2010) e na Seção 4.1, o ajuste manual dos parâmetros de ruído pode ser uma tarefa dispendiosa e complexa.

Com o propósito de explorar um método para influenciar intencionalmente a saída do algoritmo Ruído Simplex, esta seção busca apresentar um método para utilizar dados de altimetria da área que se pretende simular. Dessa maneira, procura-se obter resultados

mais representativos da topografia do local em questão. O método aqui descrito pode ser adaptado também para o Ruído de Perlin.

Essa abordagem permite criar um terreno procedural que se baseia nas características do histograma original. A utilização do Ruído de Valor, conforme feito em Parberry (2014) no entanto, nem sempre gera os melhores resultados. O presente trabalho apresenta a integração dos dados do MDE ao algoritmo do Ruído Simplex, que é um ruído gradiente, que apresenta resultados melhores na geração de terrenos.

Primeiramente, obteve-se os dados de altimetria do MDE Topodata[4]. No modelo, o território nacional é dividido em quadrículas de 1^{o} de latitude e $1,5^{\text{o}}$ de longitude. Os dados de altimetria de cada quadrícula estão disponibilizados através de um arquivo *.txt*. Internamente, os arquivos estão organizados em linhas e colunas. Cada linha representa uma coordenada geográfica específica, onde a primeira coluna contém a latitude, a segunda coluna a longitude, e a terceira a altitude. Após a escolha da região de interesse, os dados de altimetria foram processados em um *Dataset* da biblioteca de análise de dados *pandas*, que permitiu a análise e a manipulação dos dados de maneira eficiente. Gerou-se o histograma das altitudes com as informações obtidas sendo que, para o método escolhido, optou-se por dividir os dados em 100 grupos de frequências.

Ao adotar a região de Juiz de Fora como a região de análise, obteve-se a distribuição de altitudes explicitada na Figura 4.5. É possível verificar, apenas pelo perfil do histograma, o formato característico do relevo da região, que é composto principalmente por cumes arredondados e pouco bruscos, denominados "mares de morros" (AB'SÁBER, 2003). Essa característica pode ser verificada pela maior quantidade de valores inferiores e médios, indicando maior presença de vales e elevações levemente inclinadas.

De maneira análoga ao método de Parberry (2014), é criada uma tabela de frequências para as alturas. Essa tabela é em seguida transformada em uma lista sequencial normalizada para que a soma de todos seus elementos seja 1024, com objetivo de posteriormente criar uma lista de 1024 elementos. Desse modo, é criada uma tabela de números crescentes aleatórios, assim como no método original. Aqui, em contraste com a implementação realizada em Parberry (2014), os números estão no intervalo de 0 a 2, pois assim têm a

[4]Disponível em: ⟨http://www.dsr.inpe.br/topodata/index.php⟩

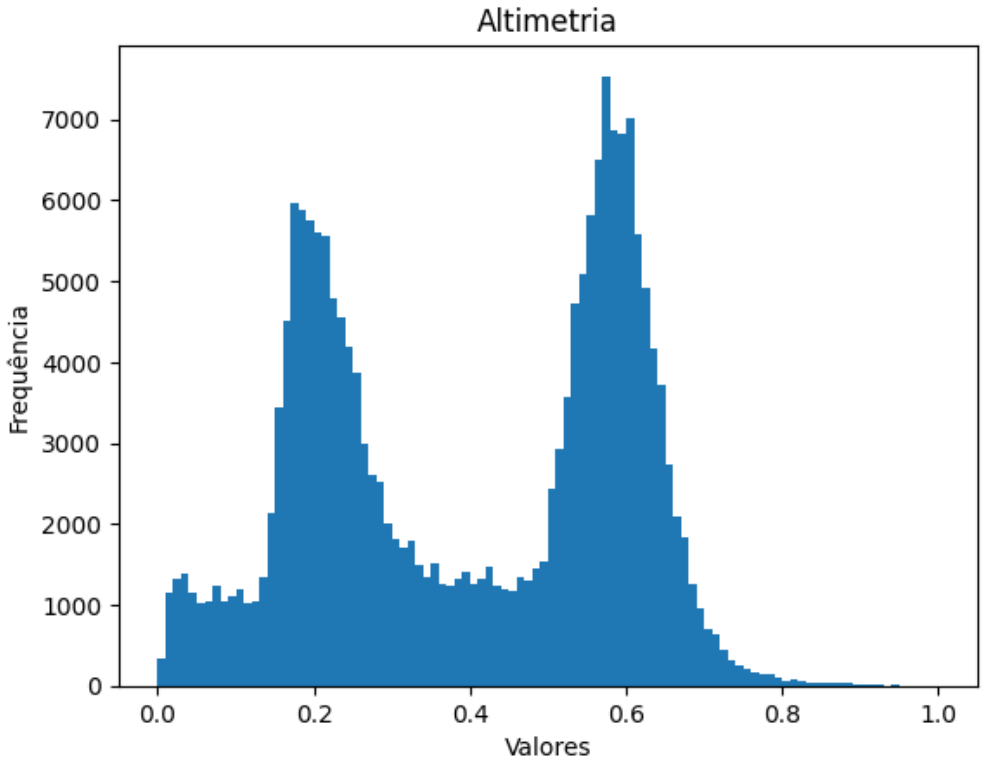

Figura 4.5: Distribuição de altitudes no relevo da região de Juiz de Fora (Fonte: Gerado pelo autor)

capacidade de diminuir ou aumentar a contribuição de cada vértice para o ruído final.

Essa tabela é indexada aos vértices da malha Simplex por meio de um processo semelhante ao utilizado para indexar os vetores gradientes na implementação do Ruído Simplex tradicional, como descrito no Capítulo 3. No entanto, pelo fato de possuir 1024 elementos, é utilizado um vetor de permutações com números entre 0 e 1023.

No passo final do Ruído Simplex, quando as contribuições dos vértices são adicinadas, o valor da contribuição de cada vértice é multiplicado pelo valor correspondente do vetor aleatório gerado com base nos dados altimétricos.

Além disso, é utilizado um multiplicador nas alturas para gerar um perfil com aparência mais realista, assim como na Seção 4.1. Porém, neste método esse multiplicador é definido com base na seguinte expressão:

$$m_{altitude} = \frac{(a_{maxima} - a_{minima})}{a_{maximaBrasil}} * c, \tag{4.2}$$

sendo a_{maxima} a maior altitude no conjunto estudado, a_{minima} a menor altitude no conjunto estudado, $a_{maximaBrasil}$ a altura correspondente à maior altitude do território brasileiro,

que é o Pico da Neblina, na fronteira entre Brasil e Venezuela, com 2994 metros de altitude, e c uma constante para regular a aparência do relevo, que foi ajustada pra o valor de 200.

Como resultado, o valor do ruído final é influenciado pela frequência das altitudes, uma vez que há mais registros relacionados a essas altitudes na tabela de valores aleatórios.

5 Resultados e Discussões

Nesse capítulo, serão apresentados resultados obtidos aplicando os métodos explanados. Os resultados foram divididos em duas partes: a primeira dedica-se a aplicar o método de geração de terrenos, utilizando os três algoritmos abordados detalhadamente no Capítulo de Fundamentação (Capítulo 3), bem como o método desenvolvido conforme descrito no Capítulo de Desenvolvimento (Capítulo 4). A segunda parte dos resultados concentra-se na apresentação e discussão sobre os terrenos gerados pelo método aprimorado apresentado no Capítulo 4, baseado na técnica proposta por Parberry (2014).

5.1 Tecnologias utilizadas

Neste trabalho, foram conduzidos experimentos utilizando a linguagem de programação C# e o motor de jogos Unity3D. A escolha pelo Unity3D 2021.2.13f1 Personal se deu devido à sua capacidade de facilitar a manipulação e renderização de objetos tridimensionais, o que era fundamental para a implementação e visualização dos experimentos. Para a análise e tratamento dos dados obtidos, foram desenvolvidos códigos em Python, utilizando principalmente a biblioteca *pandas*. Em relação ao ambiente, foi utilizado o *laptop* Samsung Expert X40 Intel(R) Core(TM) i7-5500U 2.40GHz, com 8GB RAM e sistema operacional Windows 10 Home.

5.2 Comparação entre algoritmos de ruído em geração procedural de terrenos

Nesta seção, descrevem-se os experimentos realizados com a utilização de cada algoritmo de ruído procedural apresentados na Seção 3.2 para criar terrenos empregando o processo descrito na Seção 4.1. Para cada experimento, foi criado um plano de dimensões 240×240, além de luz direcional para melhor visualização. Para a criação do mapa de cores, foram escolhidas 7 regiões, cada uma com sua cor correspondente, como explicitado

Tabela 5.1: Cores atribuídas para cada intervalo de valor de ruído

Limite inferior	Limite superior	Cor (Hexadecimal)	Cor (Visualização)
0,0	0,15	#38d1e9	
0,15	0,21	#82ad7f	
0.21	0,39	#1cee1c	
0,39	0,54	#129012	
0,51	0,68	#276a21	
0,68	0,78	#633d23	
0,78	1,0	#ffffff	

na Tabela 5.1.

Com as cores geradas, seria possível aplicar texturas adequadas para cada região, levando em consideração a forma do terreno. No entanto, no contexto deste trabalho, o foco principal está na representação da forma do terreno em si, e, portanto, a aplicação de texturas não foi implementada.

5.2.1 Aplicação do Ruído de Valor

O objetivo deste experimento é explorar a capacidade do Ruído de Valor de gerar terrenos proceduralmente.

Foi gerado um mapa de alturas aplicando 8 oitavas do Ruído de Valor. Foi utilizada uma persistência de valor $0,65$ e uma lacunaridade de valor $0,60$ para controlar o nível de detalhe e suavidade do mapa. Com base no mapa de alturas criado, foi produzido o mapa de cores e uma representação tridimensional da superfície, sendo o processo ilustrado na Figura 5.1.

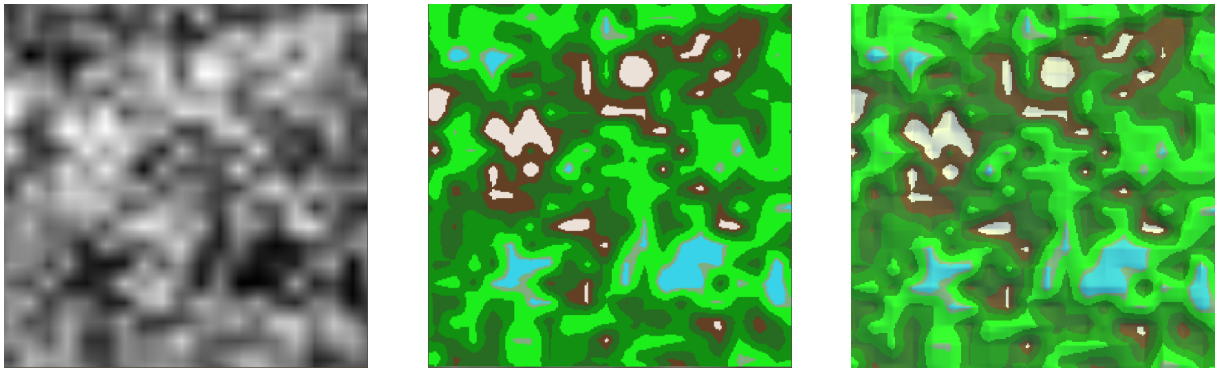

Figura 5.1: Processo de criação de terreno 3D utilizando Ruído de Valor,com persistência $0,65$ e uma lacunaridade $0,60$ (Fonte: Gerado pelo autor)

A Figura 5.2 mostra que a superfície tridimensional gerada pelo Ruído de Valor

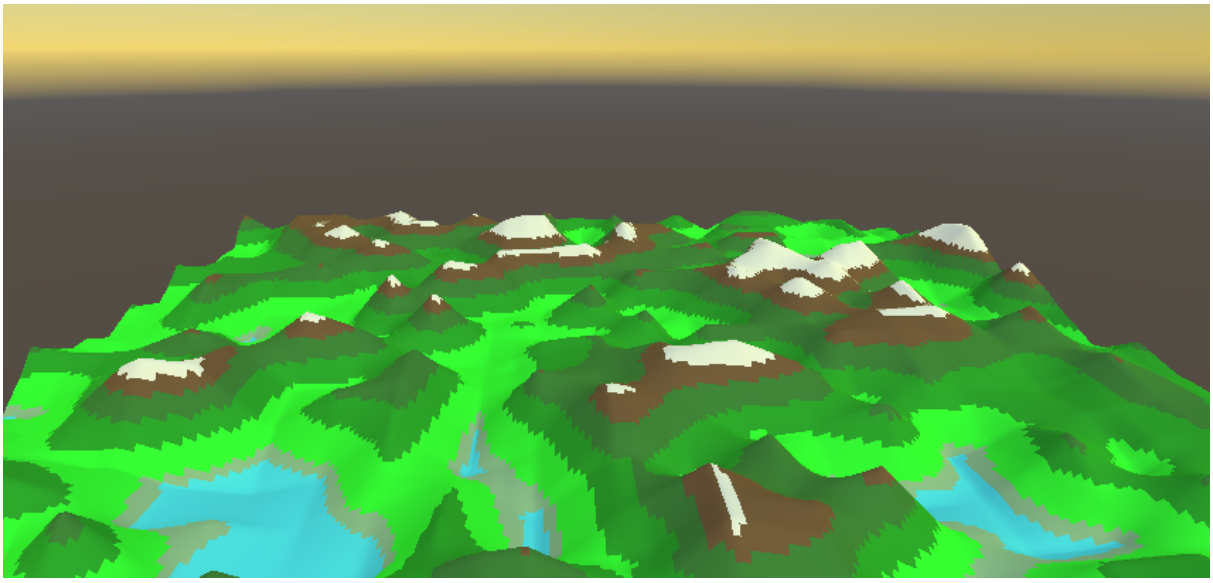

Figura 5.2: Terreno gerado através de 8 iterações de Ruído de Valor,utilizando persistência $0,65$ e uma lacunaridade de $0,60$ (Fonte: Gerado pelo autor)

apresenta deformações e transições abruptas que deixam evidente a estrutura de dados subjacente. Já a Figura 5.3 apresenta outro mapa de alturas gerado através de Ruído de Valor, com 8 oitavas utilizando $0,5$ de persistência e $0,6$ de lacunaridade. Nas regiões salientadas em vermelho pode-se observar claramente algumas das distorções que transparecem a estrutura de grade empregada no Ruído de Valor.

Figura 5.3: Terreno gerado através de 8 oitavas de Ruído de Valor, com $0,5$ de persistência e $0,6$ de lacunaridade. As regiões em vermelho demonstram artefatos visuais apresentados pelo Ruído de Valor. (Fonte: Gerado pelo autor)

Essas distorções mostradas nas Figuras 5.2 e 5.3 comprometem a aparência natural e suave do terreno gerado, limitando a capacidade do Ruído de Valor de produzir terrenos com transições suaves e superfícies realistas. Apesar disso, é possivel fazer um ajuste fino nos parâmetros do algoritmo para obter um terreno mais suave e diverso do que nos exemplos anteriores. A Figura 5.4 apresenta um terreno gerado por 12 oitavas de Ruído de Valor com 0,5 de persistência e 1,7 de lacunaridade. O terreno é bem mais suave que os anteriores, porém apresenta regiões com transições mais bruscas, destacadas em vermelho.

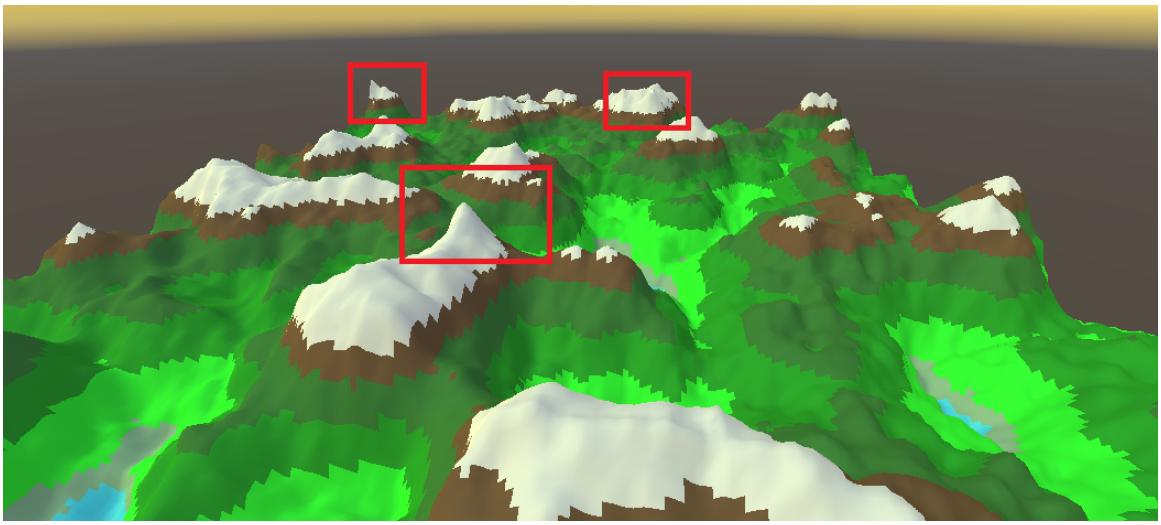

Figura 5.4: Terreno gerado através de 12 oitavas de Ruído de Valor, com 0,5 de persistência e 1,7 de lacunaridade.As regiões em vermelho demonstram artefatos visuais apresentados pelo Ruído de Valor. (Fonte: Gerado pelo autor)

5.2.2 Aplicação do Ruído de Perlin

Inicialmente, foi gerado um mapa de alturas utilizando 8 oitavas do Ruído de Perlin. Foi adotada uma persistência de valor $0,5$ e uma lacunaridade de valor $0,75$ para controlar o nível de detalhe e suavidade do mapa. Com base no mapa de alturas criado, foi produzido o mapa de cores e uma representação tridimensional da superfície, sendo o processo mostrado na Figura 5.5.

A Figura 5.6 apresenta um terreno criado através do Ruído de Perlin. Pode-se observar a aparência suave do terreno gerado, em contraste com o Ruído de Valor.

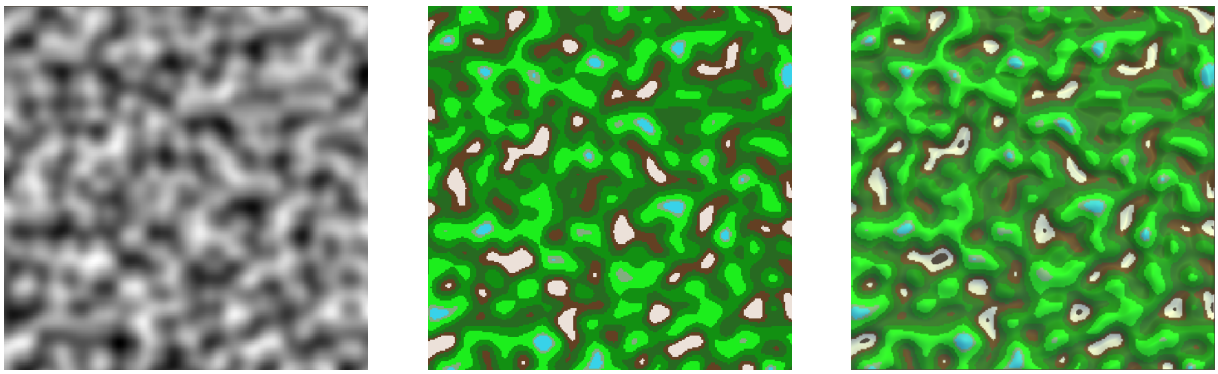

Figura 5.5: Criação de terreno 3D utilizando o Ruído de Perlin com 0,5 de persistência e 0,75 de lacunaridade (Fonte: Gerado pelo autor)

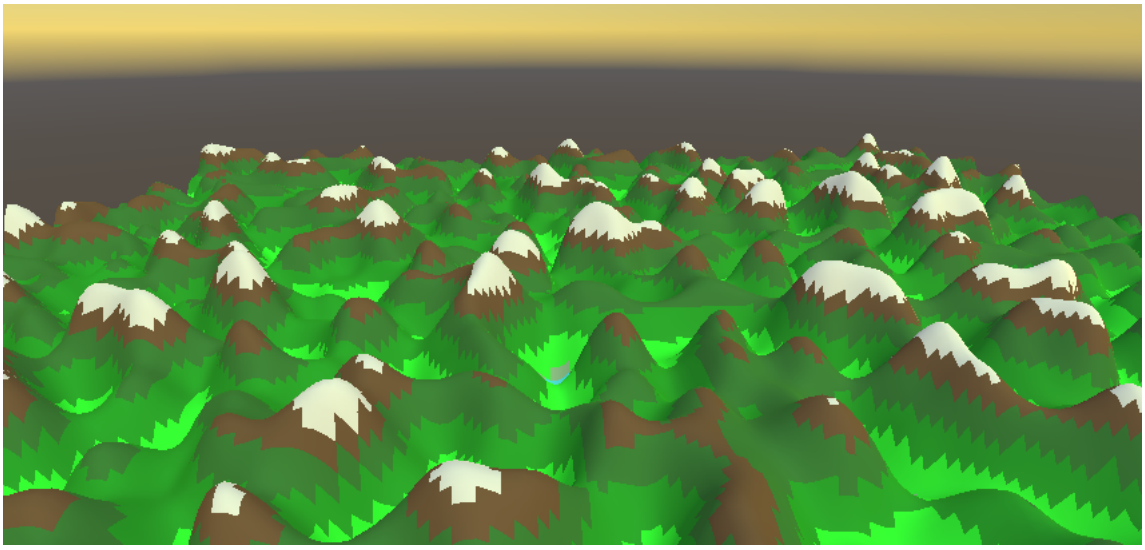

Figura 5.6: Terreno gerado através de 8 oitavas do Ruído de Perlin (Fonte: Gerado pelo autor)

Ao variar os parâmetros de lacunaridade e persistência, é possível criar superfícies com diferentes características. A Figura 5.7 apresenta uma superfície gerada através de 8 oitavas do Ruído de Perlin com lacunaridade 0,5 e persistência 0,5. Este resultado, quando comparado ao da Figura 5.8, tem aparência mais suave e menos complexa, o que pôde ser alcançado apenas com o ajuste manual de parâmetros.

Como explicitado, o Ruído de Perlin pode ser uma boa alternativa para ser aplicado em geração procedural de terrenos, por gerar superfícies suaves e contínuas além de ser muito customizável através do ajuste de parâmetros.

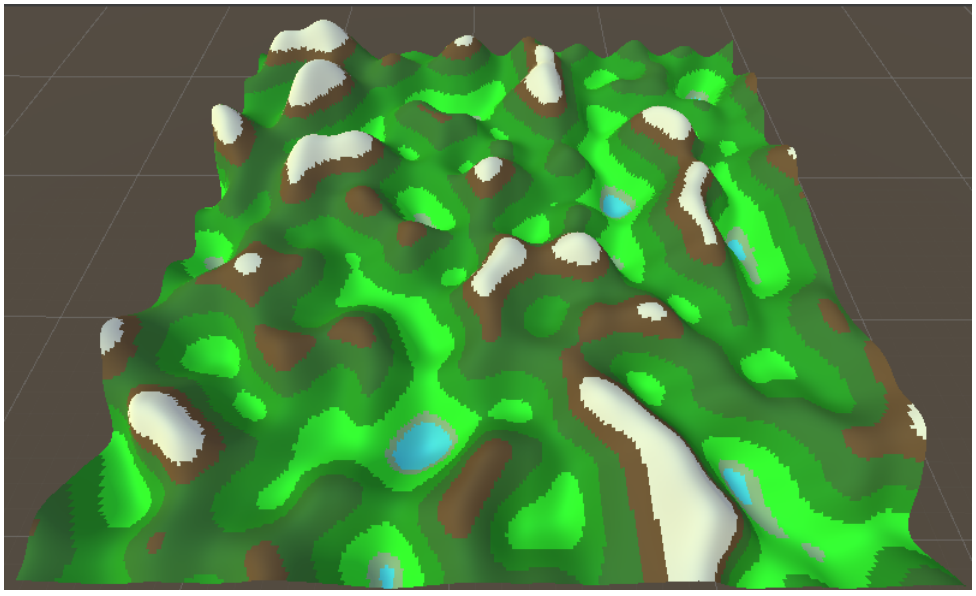

Figura 5.7: Terreno gerado através de 8 oitavas do Ruído de Perlin com lacunaridade 0,5 e persistência 0,5 (Fonte: Gerado pelo autor)

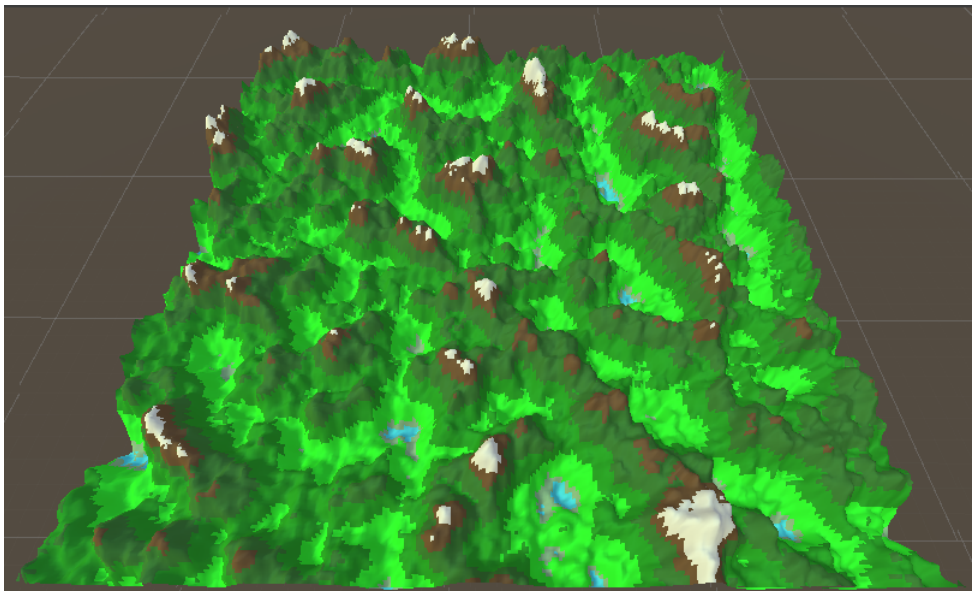

Figura 5.8: Terreno gerado através de 8 iterações do Ruído de Perlin com lacunaridade 1,5 e persistência 0,6 (Fonte: Gerado pelo autor)

5.2.3 Aplicação do Ruído Simplex

Para o Ruído Simplex, primeiramente, um mapa de alturas foi gerado utilizando 8 iterações do ruído. Para controlar o nível de detalhe e suavidade do mapa, foram utilizados persistência de valor $0,6$ e lacunaridade com valor $1,4$. Com base no mapa de alturas criado, foi produzido o mapa de cores e uma representação tridimensional da superfície. Todo o processo está ilustrado na Figura 5.9.

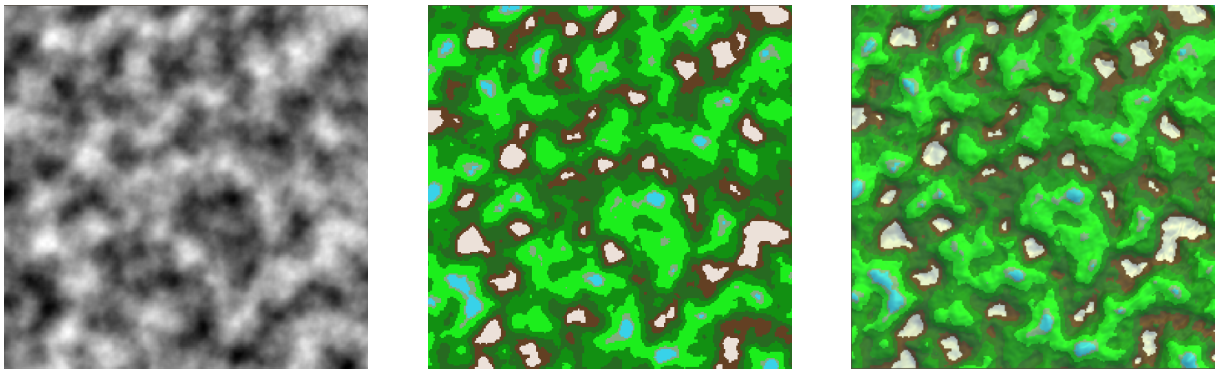

Figura 5.9: Criação de terreno 3D utilizando o Ruído Simplex com 1,4 de lacunaridade e 0,6 de persistência (Fonte: Gerado pelo autor)

Como evidenciado pela Figura 5.10, podemos aferir que, assim como o Ruído de Perlin, o Ruído Simplex é adequado para a geração de terrenos, oferecendo resultados análogos. As diferenças que podem ser observadas podem ser reduzidas através de ajuste mais fino de parâmetro.

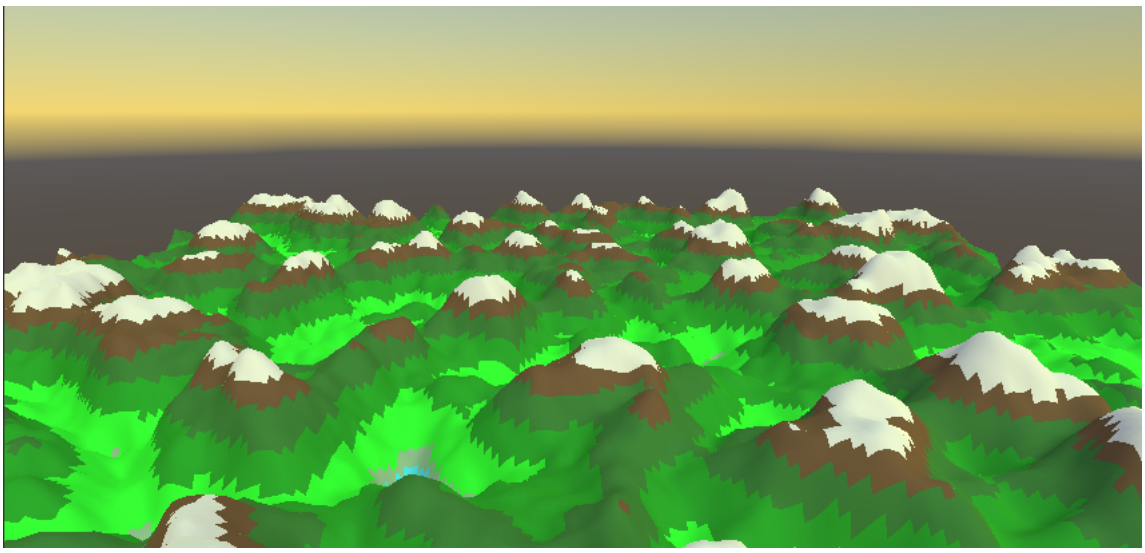

Figura 5.10: Terreno gerado através de 8 iterações do Ruído Simplex (Fonte: Gerado pelo autor)

A Figura 5.11 apresenta uma superfície gerada através de 8 oitavas do Ruído

Simplex com lacunaridade 1,66 e persistência 0,5. Este resultado, quando comparado ao da Figura 5.12, que tem lacunaridade 0,83 e persistência 0,6, possui aparência mais suave e menos complexa, o que pôde ser alcançado apenas com o ajuste de parâmetros.

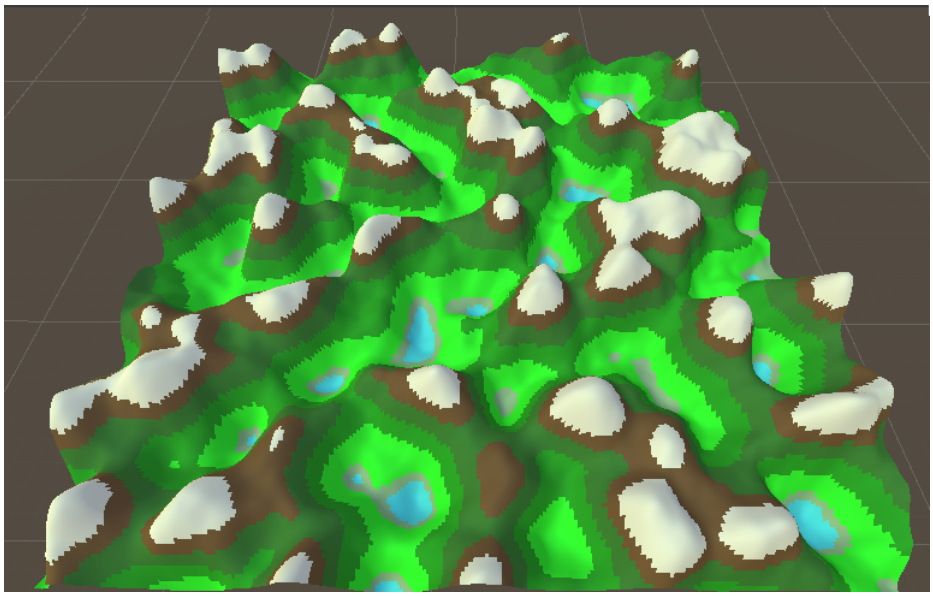

Figura 5.11: Terreno gerado através de 8 iterações do Ruído Simplex com lacunaridade 1,66 e persistência 0,5 (Fonte: Gerado pelo autor)

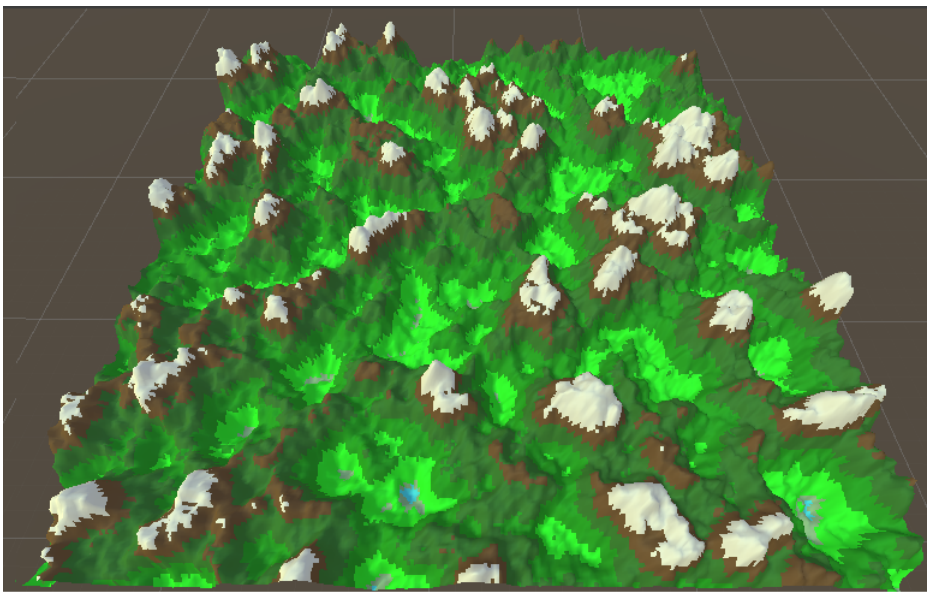

Figura 5.12: Terreno gerado através de 8 iterações do Ruído Simplex com lacunaridade 0,83 e persistência 0,6 (Fonte: Gerado pelo autor)

Ao alterar os parâmetros do ruído, foi possível obter superfícies bastante distintas. Assim, como no Ruído de Perlin, no Ruído Simplex é possível ajustar os parâmetros para controlar as características do terreno criado. Dessa forma, tanto o Ruído de Perlin quanto o Ruído Simplex fornecem ferramentas poderosas para a geração procedural de

terrenos, permitindo que os desenvolvedores e artistas ajustem os parâmetros. No entanto, pequenas alterações nestes causam notáveis efeitos no ruído final.

Além disso, embora tanto o Ruído Simplex quanto o Ruído de Perlin proporcionem resultados suaves e com poucos artefatos visuais, é importante considerar sua aplicação na geração de terrenos, onde podem surgir superfícies com muitas depressões. Isso pode resultar em paisagens que não refletem fielmente uma aparência realista, comprometendo a qualidade das representações geradas.

Por mais que o Ruído Simplex tenha sido desenvolvido como uma melhoria sobre o Ruído de Perlin, este não apresenta, em geral, características visuais necessariamente melhores que aquele. Dessa forma, o Ruído de Perlin pode ainda ser a melhor alternativa em relação a geração de ruído para superfícies suaves, pelo fato de sua implementação ser mais simples que a do Ruído Simplex, produzindo resultados semelhantes.

Ademais, apesar de o Ruído de Valor ser capaz de gerar superfícies suaves e naturais, este requer um ajuste mais fino de parâmetros além de, normalmente, precisar de uma quantidade maior de oitavas para apresentar resultados suaves.

5.3 Incorporação da Altimetria no Ruído Simplex

Como visualizado nos experimentos da seção anterior, o Ruído de Valor não é o algoritmo que gera os melhores resultados em relação a criação de terrenos. Portanto, para o presente estudo, desenvolveu-se e testou-se um método baseado no descrito em Parberry (2014) para incorporar dados de Altimetria no Ruído Simplex. Embora o Ruído de Perlin seja mais simples e mais utilizado na prática, optou-se por utilizar o Ruído Simplex por este não gerar artefatos visuais, conforme discutido em Perlin (2001), e pelo fato de este ruído não ser tão frequentemente explorado na literatura. Mas, conforme já dito, o ruído de Perlin poderia ser igualmente utilizado para a incorporação dos dados altimétricos.

O método proposto visa superar as limitações apresentadas pelo Ruído de Valor. Para tal foram empregadas algumas variações e acréscimos ao algoritmo proposto por Parberry (2014). A Figura 5.13 apresenta um terreno formado através da implementação original do Ruído Simplex, enquanto a Figura 5.14 apresenta o terreno formado através

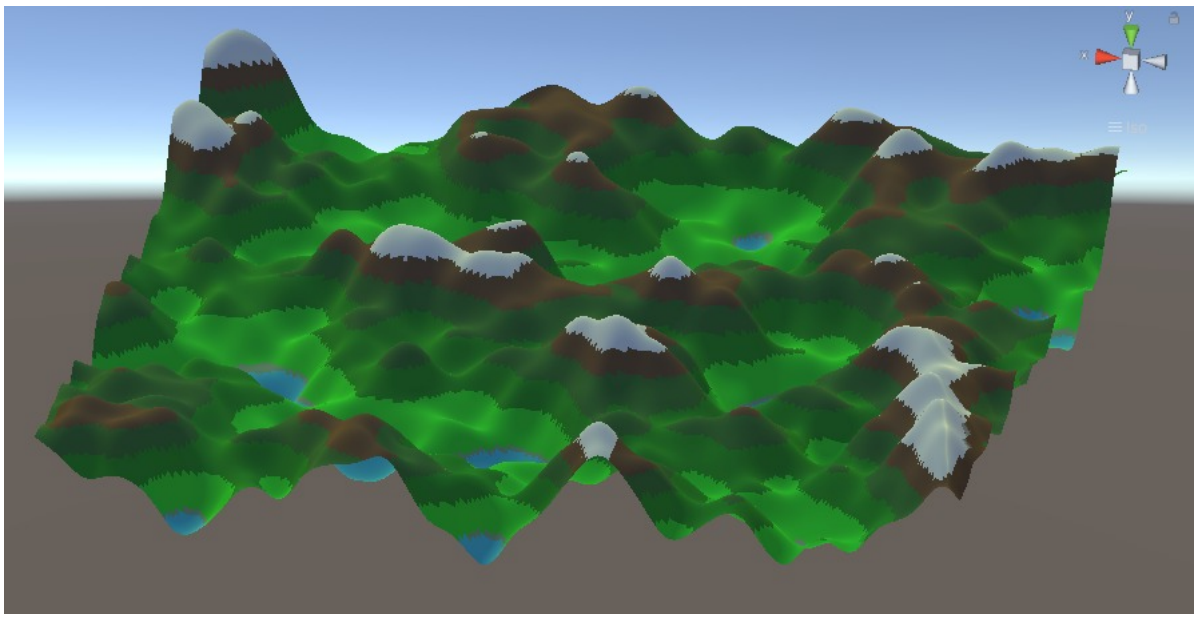

Figura 5.13: Terreno gerado através do Ruído Simplex tradicional com 0,65 de persistência e 1,64 de lacunaridade (Fonte: Gerado pelo autor)

da implementação modificada do Ruído Simplex, incorporando dados geográficos. Foram empregadas 4 oitavas de ruído com valor de $0,65$ para a persistência, $1,64$ para a lacunaridade, e as alturas foram todas multiplicadas por um fator constante de 44.

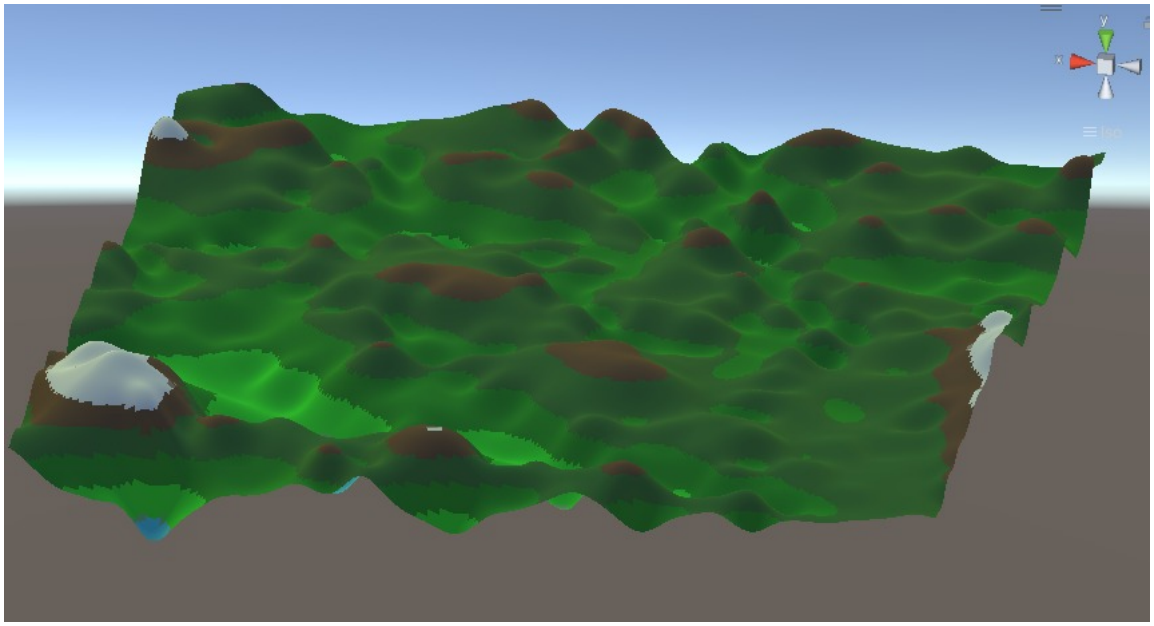

Figura 5.14: Terreno gerado através do Ruído Simplex com 0,65 de persistência e 1,64 de lacunaridade, influenciado pelos dados de altimetria da região de Juiz de Fora, Minas Gerais (Fonte: Gerado pelo autor)

Para o terreno apresentado na Figura 5.14, foram aplicados os dados de altimetria provenientes do Topodata, especificamente da quadrícula 21s435. Foi extraída uma

parcela com 100000 coordenadas, pois cada quadrícula do Topodata representa uma área muito extensa, que pode ter características muito diferentes entre si. É possível observar a atenuação das formas íngremes do terreno, além de uma diminuíção da altura da maioria dos trechos mais elevados, o que está de acordo com o esperado para o terreno da região próxima a Juiz de Fora.

A Figura 5.15 apresenta duas visões aproximadas da mesma região dos terrenos mostrados nas Figuras 5.13 e 5.14. Na imagem da esquerda, que corresponde a uma parte do terreno de Figura 5.13, pode-se observar tanto elevações como depressões com alta inclinação. Por outro lado, na imagem da direita, que retrata a mesma parte do terreno na Figura 5.14, é perceptível uma diminuição das inclinações e uma maior planificação das regiões de depressão.

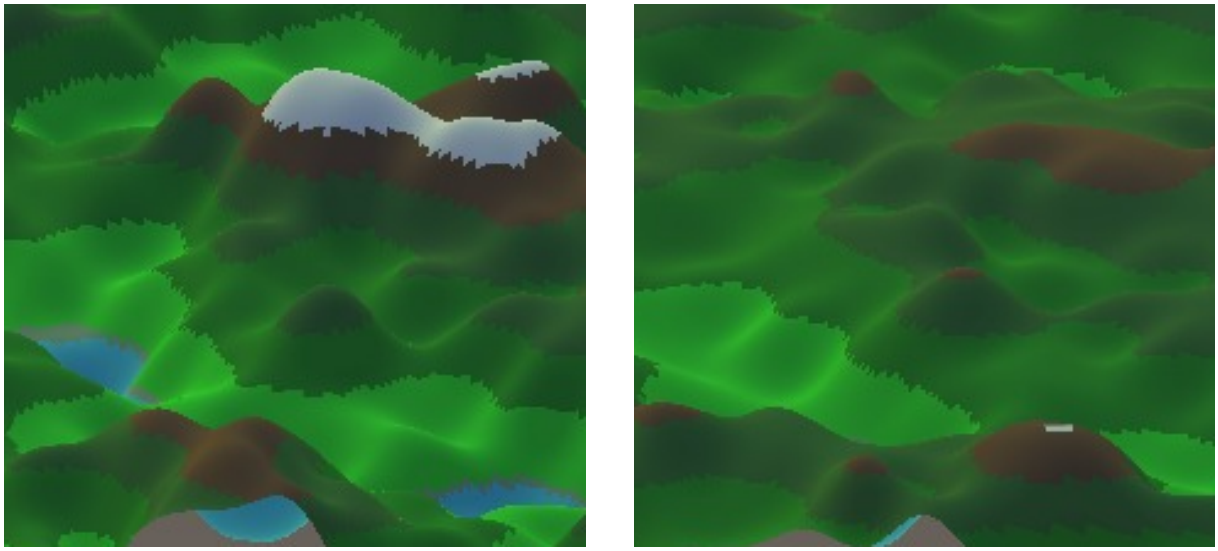

Figura 5.15: Comparação de região sem e com o método (Fonte: Gerado pelo autor)

Foi realizado outro experimento, aplicando-se um deslocamento nos pontos utilizados no cálculo do ruído, a fim de gerar um terreno diferente do anterior. O deslocamento em x foi de 318 e em y foi de 50,7. O parâmetro de persistência possuiu valor de 0,65, e a lacunaridade foi definida em 1,64. A Figura 5.16 apresenta um terreno gerado utilizando Ruído Simplex sem utilizar os dados altimétricos. Pode-se notar as estruturas normalmente apresentadas por esse método, como depressões e elevações íngremes. Por outro lado, a Figura 5.17 mostra um terreno gerado através do Ruído Simplex utilizando dados de altimetria da região de Juiz de Fora. Assim como na Figura 5.14, o terreno gerado possui maior suavidade nos morros gerados, além de menos depressões quando comparado

ao método original, sem usar os dados.

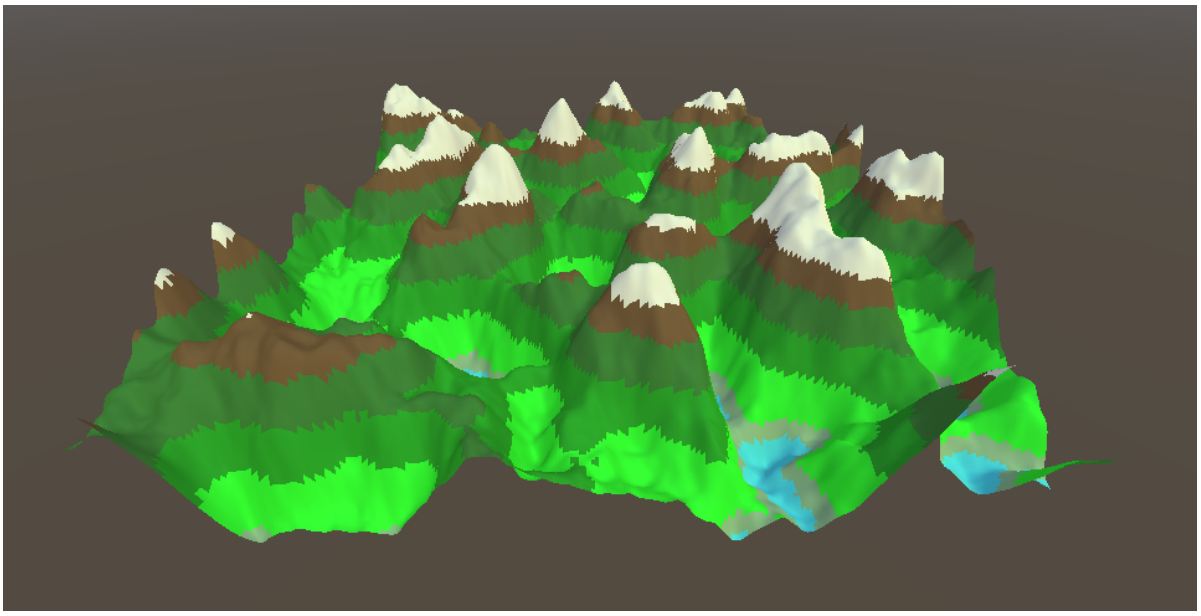

Figura 5.16: Terreno gerado através do Ruído Simplex com 0,65 de persistência e 1,64 de lacunaridade, deslocado de 318 em x e de 50,7 em y (Fonte: Gerado pelo autor)

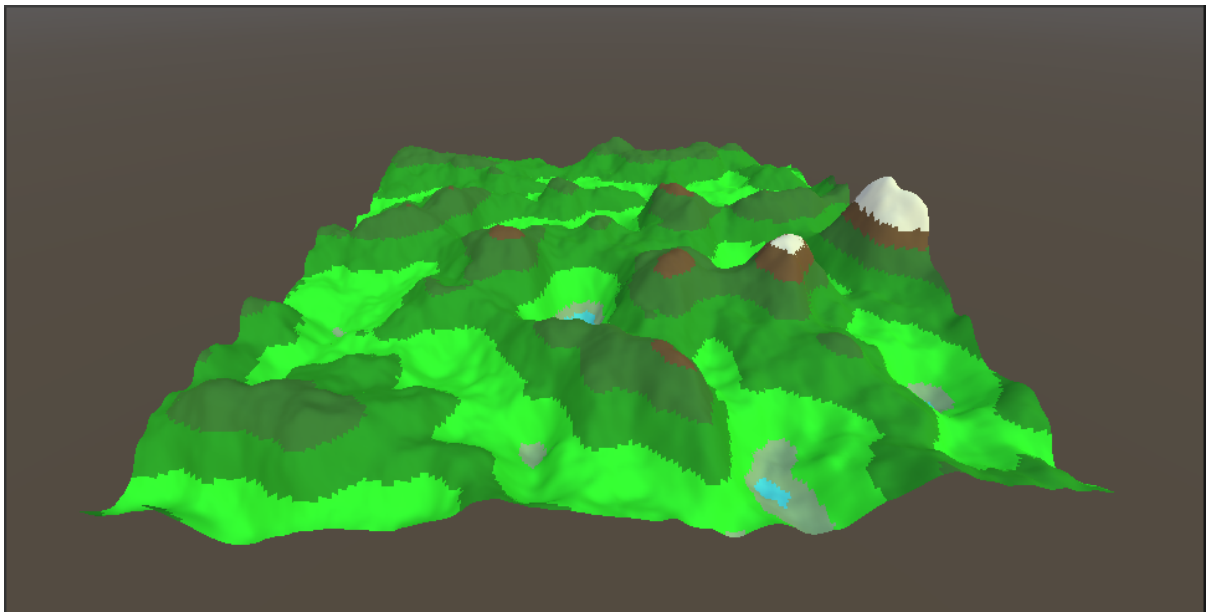

Figura 5.17: Terreno gerado através do Ruído Simplex com 0,65 de persistência e 1,64 de lacunaridade,deslocado de 318 em x e de 50,7 em y, influenciado pelos dados de altimetria da região de Juiz de Fora, Minas Gerais (Fonte: Gerado pelo autor)

Por fim, também foram utilizados dados de relevo da Serra do Espinhaço, uma cadeia montanhosa no norte de Minas Gerais. A serra do Espinhaço é considerada a única cordilheira do Brasil, sendo uma região com relevo mais íngreme, quando comparado ao relevo da região do domínio de Mares de Morros. Dessa forma utilizou-se a quadrícula

17s435 do Topodata e obteve-se um valor de 76 para o multiplicador de altitude. A Figura 5.18 apresenta o histograma para as altitudes nessa região, enquanto a Figura 5.19 mostra um terreno gerado por meio do Ruído Simplex utilizando dados de altimetria da região da Serra do Espinhaço.

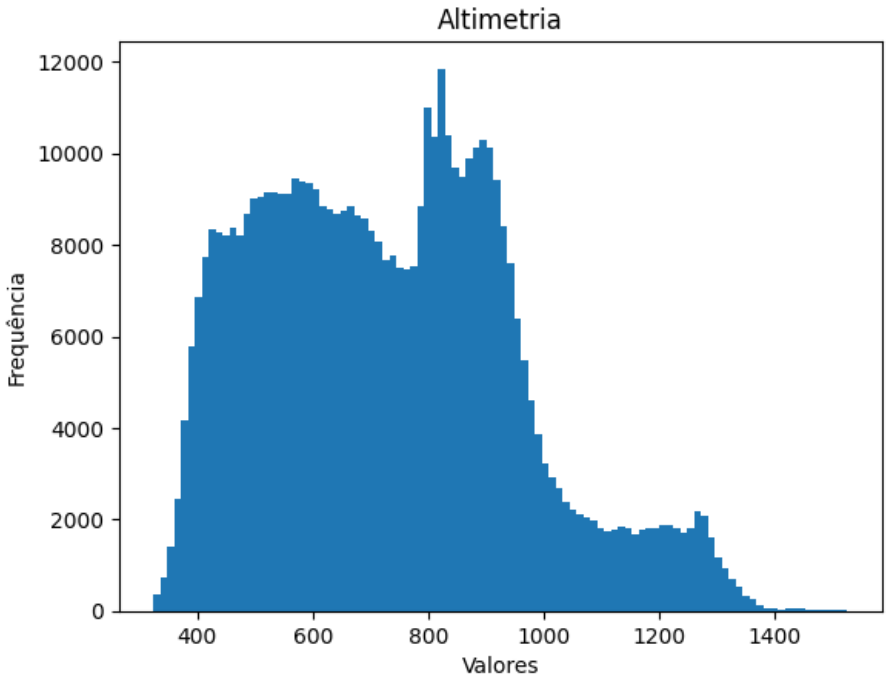

Figura 5.18: Distribuição de altitudes no relevo da região da Serra do Espinhaço (Fonte: Gerado pelo autor)

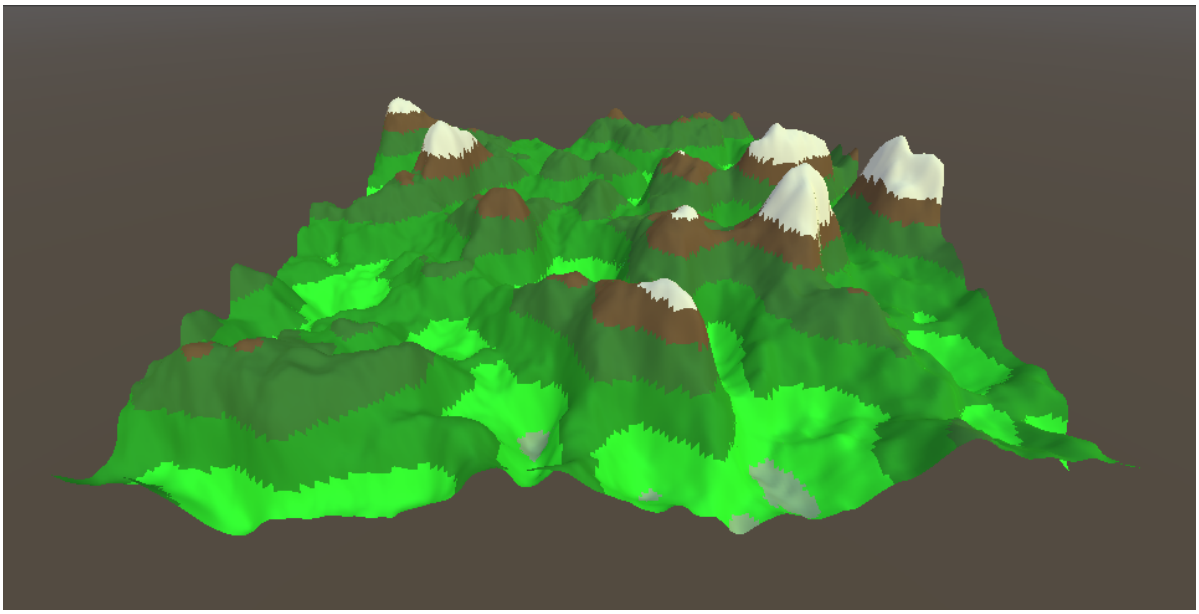

Figura 5.19: Terreno gerado através do Ruído Simplex com 0,65 de persistência e 1,64 de lacunaridade,deslocado de 318 em x e de 50,7 em y, influenciado pelos dados de altimetria da região da Serra do Espinhaço (Fonte: Gerado pelo autor)

Observando a Figura 5.19, nota-se que o relevo é mais íngreme quando comparado ao relevo gerado com base na região de Juiz de Fora, conforme esperado. Visualiza-se, também, uma menor prevalência de depressões para alturas mais baixas, o que se assemelha à realidade.

Os resultados dos experimentos com dados de altimetria sugerem que o método pode ajustar de forma adequada o perfil de valores gerados pelo Ruído Simplex para corresponder ao perfil altimétrico do local estudado. Essa adaptação é evidente, especialmente na transição de declives entre as simulações, que se alinham ao terreno real, assim como na redução significativa na ocorrência de depressões em altitudes mais baixas. Por mais que a incorporação de dados altimétricos melhore o resultado da geração de terrenos, o ajuste dos parâmetros de persistência, lacunaridade e multiplicador de alturas ainda é de suma importância para configurar a aparência do terreno para a apliação desejada.

6 Considerações finais

Este trabalho explorou a geração procedural de terrenos utilizando funções de ruído. Além disso, investigou a viabilidade da incorporação de dados reais de altimetria no Ruído Simplex para gerar terrenos com as características específicas do terreno analisado. Com essa abordagem, o estudo procurou obter terrenos mais realistas e com as características geográficas reais, o que pode ter grande relevância em diversas aplicações, como jogos, animações e visualizadores de ambientes virtuais.

Foi apresentada a teoria referente a funções de ruído, sendo descritos em detalhes os algoritmos Ruído de Valor, Ruído de Perlin e Ruído Simplex. Foram discutidos, também, conceitos relacionados a altimetria, bem como Modelos Digitais de Elevação e Movimento Browniano Fractal. Um método para incorporação de dados reais de altimetria diretamente no algoritmo de Ruído Simplex foi usado para gerar terrenos proceduralmente com maior fidelidade à topografia local, cujos resultados foram apresentados no Capítulo 5, juntamente com os experimentos envolvendo os três algoritmos de ruído abordados. Observou-se que os três ruídos são adequados para geração de terrenos suaves e com características realistas. No entanto, o Ruído de Valor requer um ajuste de parâmetros ainda mais fino quando comparado com os outros, apresentando um resultado com qualidade inferior. Também foi verificado que, para os experimentos realizados, com diferentes combinações de parâmetros, os algoritmos Ruído de Perlin e Ruído Simplex são aparentemente equivalentes no que tange a geração de terrenos procedurais.

Além do mencionado, foi demonstrado que a incorporação de dados de altimetria na geração procedural de terrenos pode ser uma boa alternativa para conferir realismo ao terreno gerado. Essa abordagem permite capturar características de uma localização real, gerando resultados mais autênticos, ao contrário dos métodos tradicionais que, devido a sua natureza aleatória, geram resultados mais genéricos.

Entretanto, mesmo com a incorporação de dados reais, o desafio de encontrar um conjunto de parâmetros adequados ainda persiste. No entanto, essa técnica representa um passo significativo em direção à criação de terrenos mais realistas e autênticos.

Como trabalhos futuros, uma possibilidade natural é a aplicação do método de incorporação de dados de altimetria ao Ruído de Perlin, considerando a semelhança de natureza com o Ruído Simplex. Em princípio, outros ruídos também poderiam ser adaptados para o uso desse tipo de informação. Além disso, combinar dados de altimetria com informações sobre a orometria do terreno, assim como abordado em Lamontagne (2022), é outra abordagem interessante para obter terrenos com uma aparência ainda mais realista.

Bibliografia

AB'SÁBER, A. N. *Os domínios de natureza no Brasil: potencialidades paisagísticas*. [S.l.]: Ateliê editorial, 2003. v. 1.

CONINX, A. et al. Visualization of uncertain scalar data fields using color scales and perceptually adapted noise. In: *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*. [S.l.: s.n.], 2011.

DUSTLER, M. et al. Application of the fractal perlin noise algorithm for the generation of simulated breast tissue. In: *Medical Imaging 2015: Physics of Medical Imaging*. [S.l.]: SPIE, 2015. v. 9412, p. 94122L.

EBERT, D. S. et al. *Texturing & modeling: a procedural approach*. [S.l.]: Morgan Kaufmann, 2003.

ESPARTEL, L. Curso de topografia, editora globo. *Rio de Janeiro*, 1987.

FARR, T. G.; KOBRICK, M. Shuttle radar topography mission produces a wealth of data. *EOS (Transactions, American Geophysical Union)*, v. 81, p. 583–585, 2000.

GALIN, E. et al. A review of digital terrain modeling. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 2019. v. 38, n. 2, p. 553–577.

GASCH, C. et al. Procedural modelling of terrains with constraints. *Multimedia Tools and Applications*, Springer, v. 79, p. 31125–31146, 2020.

GUÉRIN, E. et al. Sparse representation of terrains for procedural modeling. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 2016. v. 35, n. 2, p. 177–187.

GUÉRIN, É. et al. Interactive example-based terrain authoring with conditional generative adversarial networks. *ACM Trans. Graph.*, v. 36, n. 6, p. 228–1, 2017.

GUSTAVSON, S. *Simplex noise demystified*. 2005. Self-published tutorial. Disponível em: ⟨https://www.researchgate.net/publication/216813608_Simplex_noise_demystified⟩.

JAIN, A.; SHARMA, A.; RAJAN. Adaptive & multi-resolution procedural infinite terrain generation with diffusion models and perlin noise. In: *Proceedings of the Thirteenth Indian Conference on Computer Vision, Graphics and Image Processing*. [S.l.: s.n.], 2022. p. 1–9.

LAGAE, A. et al. A survey of procedural noise functions. In: WILEY ONLINE LIBRARY. *Computer Graphics Forum*. [S.l.], 2010. v. 29, n. 8, p. 2579–2600.

LAMONTAGNE, E. Orometric noise for procedural terrain generation. In: *ICETIS 2022; 7th International Conference on Electronic Technology and Information Science*. Harbin, China: [s.n.], 2022. p. 1–8.

LI, H. et al. A parallel algorithm using perlin noise superposition method for terrain generation based on cuda architecture. In: ATLANTIS PRESS. *International Conference on Materials Engineering and Information Technology Applications (MEITA 2015)*. [S.l.], 2015. p. 967–974.

MUSGRAVE, F. K. *Methods for realistic landscape imaging*. [S.l.]: Yale University, 1993.

MUSGRAVE, F. K.; KOLB, C. E.; MACE, R. S. The synthesis and rendering of eroded fractal terrains. *ACM Siggraph Computer Graphics*, ACM New York, NY, USA, v. 23, n. 3, p. 41–50, 1989.

OLIVEIRA, C. G. *Avaliação de modelos digitais de elevação gerados a partir de sensores remotos orbitais óptico (ASTER) e radar (RADARSAT-1, SRTM): um estudo para a região da Serra dos Carajás (PA)*. 184 p. Tese (Doutorado) — INPE, São José dos Campos, 2005. (INPE-13168-TDI/1027).

PARADELLA, W. R. et al. A geração de modelos digitais de elevação pela estereoscopia de radar: conhecimento atual e resultados com imagens radarsat-1 na amazônia. In: *Simpósio Brasileiro de Sensoriamento Remoto*. [S.l.: s.n.], 2001. v. 10, p. –.

PARBERRY, I. Designer worlds: Procedural generation of infinite terrain from real-world elevation data. *Journal of Computer Graphics Techniques*, v. 3, n. 1, 2014.

PARBERRY, I. Modeling real-world terrain with exponentially distributed noise. *Journal of Computer Graphics Techniques*, v. 4, n. 2, p. 1–9, 2015.

PERLIN, K. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, v. 19, n. 3, p. 287–296, 1985.

PERLIN, K. Noise hardware. *Real-Time Shading SIGGRAPH Course Notes*, v. 8, p. 19, 2001.

PERLIN, K. Improving noise. *ACM Transactions on Graphics (TOG)*, v. 21, n. 3, p. 681–682, 2002.

PERLIN, K.; NEYRET, F. Flow noise. In: SIGGRAPH. *28th International Conference on Computer Graphics and Interactive Techniques (Technical Sketches and Applications)*. [S.l.], 2001. p. 187.

POPOV, A. Using perlin noise in sound synthesis. In: *Proceedings of the Linux Audio Conference*. [S.l.: s.n.], 2018.

SANTAMARÍA-IBIRIKA, A. et al. Procedural approach to volumetric terrain generation. *The Visual Computer*, Springer, v. 30, p. 997–1007, 2014.

SMELIK, R. et al. Integrating procedural generation and manual editing of virtual worlds. In: *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*. [S.l.: s.n.], 2010. p. 1–8.

ŠT'AVA, O. et al. Interactive terrain modeling using hydraulic erosion. In: *Proceedings of the 2008 acm siggraph/eurographics symposium on computer animation*. [S.l.: s.n.], 2008. p. 201–210.

VALERIANO, M. d. M. Topodata: guia para utilização de dados geomorfológicos locais. *São José dos Campos: INPE*, v. 72, 2008.

VEIGA, L. A. K.; ZANETTI, M. A. Z.; FAGGION, P. L. *Fundamentos de Topografia*. Paraná: Universidade Federal do Paraná, 2012.

VITACION, R. J.; LIU, L. Procedural generation of 3d planetary-scale terrains. In: IEEE. *2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*. [S.l.], 2019. p. 70–77.

WORLEY, S. A cellular texture basis function. In: *Proceedings of SIGGRAPH*. [S.l.: s.n.], 1996. p. 291–294.

ZHOU, H. et al. Terrain synthesis from digital elevation models. *IEEE transactions on visualization and computer graphics*, IEEE, v. 13, n. 4, p. 834–848, 2007.