



Framework Para a Modelação Procedimental Integrada de Terrenos Completos

HUGO FILIPE ALVES PINTO

Outubro de 2017

Framework Para a Modelação Procedimental Integrada de Terrenos Completos

Hugo Filipe Alves Pinto

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: Doutor João Miguel Queirós Magno Leitão

Co-orientador: Doutor João Paulo Jorge Pereira

Supervisor: Doutor Carlos José Ribeiro Campos

Resumo

Desde o início dos anos 80 que a modelação procedimental tem sido abordada na criação de terrenos. Inicialmente com o objetivo de criar mapas de elevação revelou, mais tarde, ser um tema com bastante potencial para a modelação de outros componentes de um terreno. Com os sucessivos contributos e o aparecimento de novas técnicas, o estudo da modelação procedimental passou também a abordar outras características, como as redes fluviais, redes de estradas, vegetação, e mais recentemente centros urbanos. Com o objetivo de aumentar o realismo dos modelos obtidos, técnicas baseadas em fenómenos físicos foram concebidas.

Apesar do potencial comprovado e da utilização de técnicas de modelação procedimental na criação de terrenos em videojogos e simuladores, existem ainda dificuldades na sua aplicação. A existência de diversas soluções, o facto de isoladamente focarem num conjunto limitado de características do terreno e a falta de controlo do processo de criação têm constituído um entrave na criação de *frameworks* que permitam gerar terrenos completos e utilizáveis.

Neste trabalho foi efetuado o estudo das diferentes técnicas de modelação procedimental para cada característica constituinte de um terreno e foi concebida e desenvolvida uma *framework* de modelação procedimental. A *framework* desenvolvida permite a integração dessas técnicas de forma a modelar terrenos completos, com ferramentas que facilitam a interação do utilizador e a alteração de parâmetros no processo de criação.

Para demonstrar o potencial da solução desenvolvida, técnicas de modelação procedimental foram implementadas e integradas na *framework*. Os diferentes meios de parametrização foram identificados para cada técnica implementada e os resultados da parametrização foram analisados nos modelos gerados.

A solução desenvolvida no âmbito deste projeto permitiu solucionar o problema da integração e da controlabilidade de diferentes camadas que constituem a modelação de um terreno. Foi efetuada a análise de valor que permitiu identificar os benefícios que a solução introduz na adoção de técnicas de modelação procedimental e na investigação e desenvolvimento de novas técnicas.

Palavras-chave: *framework*, modelação procedimental, modelação de terrenos, Unity3D

Abstract

Since the beginning of the 80s, procedural modeling have been applied in the creation of terrains. Being initially used to create height maps, lately, procedural modeling revealed to have great potential modeling other terrain components. With further contributes and the development of new techniques, other terrain characteristics such as water bodies, road networks, vegetation, and recently, urban centers began to be considered for the application of procedural modeling. To further enhance realism, techniques inspired by physics phenomena have been developed.

Despite of proven potential in the use of procedural modeling techniques to create terrains for videogames and simulators, there are still some difficulties in its application. The existence of different solutions, the fact that when isolated they focus in a subset of terrain characteristics and the lack of control over the generation process turn out to be a difficulty in the development of frameworks that could generate complete and usable terrains.

In this work, existing procedural modeling techniques were studied for each terrain characteristic and a procedural modeling framework was developed. This framework integrates different procedural modeling techniques, allowing them to generate a complete terrain, and delivering easy tools that give control over the modeling process.

To demonstrate the potential of the developed solution, procedural modeling techniques were implemented and integrated inside the framework. Different methods of parametrization were identified for each implemented technique and parametrization results were analyzed in generated models.

The solution developed in this project solved the integration problem and the controllability of different layers that form a complete terrain modeling. A value analysis were made to identify the benefits that the solution brings in the adoption of procedural modeling techniques and in the investigation and development of new techniques.

Keywords: *framework*, procedural modeling, terrain modeling, Unity3D

Agradecimentos

Quero agradecer, em primeiro lugar, aos meus orientadores, o Professor Miguel Leitão, o Professor João Paulo Pereira e o Professor Carlos Campos pelo seu apoio incondicional e pela partilha dos seus conhecimentos no tema deste trabalho e vasta experiência de orientação.

Aos meus pais, pelo esforço de uma vida, graças à qual eu tive a oportunidade de me formar. À minha irmã, que sempre foi uma inspiração e um exemplo de dedicação e disciplina.

Aos meus amigos, que me têm acompanhado e apoiado nos bons e maus momentos.

Por fim, agradeço à minha esposa, pela paciência, companhia e pela ajuda que me tem dado, motivando-me na continuação do desenvolvimento deste trabalho.

Índice

1	Introdução	1
1.1	O Que é Uma Técnica Procedimental?.....	1
1.2	Descrição do Problema	1
1.3	Objetivos.....	2
1.4	Estrutura do Documento	3
2	Estado da Arte	5
2.1	Superfície do Terreno	5
2.1.1	Mapa de Elevação	5
2.1.2	Representação Multicamada	8
2.2	Rios, Oceanos e Lagos	8
2.3	Vegetação.....	9
2.4	Redes de Estradas	12
2.5	Ambiente Urbano.....	14
2.6	Integração e <i>Frameworks</i>	14
2.7	Erosão	15
2.8	Resumo do Capítulo.....	16
3	Análise do Problema	17
3.1	Pressupostos.....	17
3.2	Limitações	18
3.3	Contributos Importantes	18
3.3.1	<i>Framework</i> de Modelação Procedimental.....	18
3.3.2	Controlabilidade.....	19
3.4	<i>Workflow</i> do Processo de Criação	19
3.5	Estruturas de Dados.....	19
3.6	<i>Framework</i> de Modelação Procedimental.....	20
3.7	<i>Frameworks</i> e Linguagens de Programação a Utilizar	21
3.8	Análise e Escolha de Técnicas de Modelação Procedimental	21
3.8.1	Critérios e Matriz de Comparação.....	22
3.8.2	Comparação de Alternativas na Modelação de Mapas de Elevação	25
3.8.3	Comparação de Alternativas na Modelação de Rios, Oceanos e Lagos	27
3.8.4	Comparação de Alternativas na Modelação de Vegetação	29
3.8.5	Comparação de Alternativas na Modelação de Redes de Estradas	31
3.8.6	Comparação de Alternativas na Modelação de Centros Urbanos	33
3.9	Resumo do Capítulo.....	34

4	Análise de Valor	37
4.1	Contexto	37
4.1.1	Descrição da Solução	37
4.1.2	Clientes	38
4.1.3	Intervenientes	38
4.1.4	Principais Interações.....	39
4.1.5	Lucro.....	39
4.2	Processo de Inovação	40
4.2.1	Identificação de Oportunidade	40
4.2.2	Análise de Oportunidade.....	40
4.2.3	Geração de Ideias e Enriquecimento	41
4.2.4	Seleção de Ideias	42
4.2.5	Definição de Conceito	42
4.3	Valor Para o Cliente.....	43
4.3.1	Benefícios e Sacrifícios para Artistas e Desenvolvedores de Modelos 3D.....	43
4.3.2	Benefícios e Sacrifícios para Investigadores e Programadores de <i>Software</i> ...	44
4.4	Modelo de Negócio de <i>Canvas</i>	45
4.5	Rede de valor	46
4.6	Proposta de Valor	47
4.7	Resumo do Capítulo.....	48
5	<i>Design</i> da Solução	49
5.1	Solução Modular	49
5.2	Arquitetura da Aplicação.....	49
5.2.1	<i>Plugins</i> de Modelação Procedimental.....	49
5.2.2	Núcleo da <i>Framework</i>	50
5.2.3	Interface para o Utilizador	51
5.3	Controlabilidade e Interface para o Utilizador.....	51
5.4	Interface com as Técnicas de Modelação	52
5.5	Edição do <i>Workflow</i> e Escolha de <i>Plugins</i>	53
5.6	Fluxo de Dados.....	54
5.7	Principais Estruturas de Dados	56
5.8	Resumo do Capítulo	58
6	Implementação.....	59
6.1	<i>Framework</i>	59
6.1.1	Sistema de <i>Plugins</i>	59
6.1.2	Projeto de Terreno.....	60
6.1.3	Configuração do <i>Workflow</i> de Modelação.....	62
6.1.4	Geração de Terrenos	64
6.1.5	Ferramentas de Visualização e Edição	66
6.2	Técnicas de Modelação Procedimental	68

6.2.1	Algoritmo <i>Diamond Square</i>	69
6.2.2	Algoritmo <i>Midpoint Displacement's Inverse</i>	71
6.2.3	Geração de Linhas de Serra	74
6.2.4	Algoritmos de Erosão	76
6.2.5	Algoritmo de Erosão Térmica	76
6.2.6	Algoritmo de Erosão Hidráulica	78
6.2.7	Algoritmo de Colonização Espacial.....	81
6.2.8	Modelação de Mapas de Elevação	83
6.2.9	Modelação de Vegetação.....	85
6.2.10	Integração com a <i>Framework</i>	88
6.3	Resumo do Capítulo	89
7	Avaliação de Resultados	91
7.1	Resultados da Parametrização dos Mapas de Elevação	91
7.2	Resultados da Parametrização da Vegetação	98
7.3	Resumo do Capítulo	100
8	Conclusão	103
8.1	Objetivos Atingidos	103
8.2	Dificuldades Encontradas	105
8.3	Sugestões e Melhorias	105
8.4	Trabalhos Futuros	106

Lista de Figuras

Figura 1 – Representação de duas iterações do algoritmo <i>diamond square</i> (Olsen 2004, p. 3) .	6
Figura 2 – Representação do algoritmo (MDI) (Belhadj & Audibert 2005, p. 3).....	7
Figura 3 – Os vários passos do algoritmo de colonização espacial (Runions et al. 2007, p. 2)..	11
Figura 4 – Logótipo da framework a desenvolver.....	37
Figura 5 – Modelo de negócio de <i>canvas</i>	45
Figura 6 – Rede de valor do modelo de negócio.....	47
Figura 7 – Diagrama de componentes da solução proposta.....	50
Figura 8 – Protótipo da configuração do <i>workflow</i> de modelação.....	53
Figura 9 – Diagrama de classes da biblioteca <i>TerrainLayerBase</i>	57
Figura 10 – Vista de escolha de um projeto de terreno.....	60
Figura 11 – Vista de projeto de terreno.....	61
Figura 12 – Edição de parâmetros na vista de projeto de terreno.....	62
Figura 13 – Vista de configuração do <i>workflow</i> de modelação.....	63
Figura 14 – Lista de <i>plugins</i> disponíveis e respetivas restrições.....	64
Figura 15 – Detalhe do botão para iniciar a modelação e a seleção de camadas de terreno ...	65
Figura 16 – Diferentes ferramentas de edição de parâmetros e visualização de resultados....	67
Figura 17 – Grelha sem elevações definidas.....	69
Figura 18 – Atribuição dos valores de elevação iniciais.....	70
Figura 19 – Cálculo do valor da elevação da célula central.....	70
Figura 20 – Cálculo do valor da elevação das células centrais das laterais das subgrelhas.....	71
Figura 21 – Iterações de subdivisão para obter os valores de elevação.....	71
Figura 22 – A verde, as células que dão origem à elevação da célula vermelha no algoritmo <i>diamond square</i>	72
Figura 23 – Processo de geração de linhas de serra.....	75
Figura 24 – Vizinhança de Moore, a vermelho as células adjacentes à célula verde.....	77
Figura 25 – Quando a diferença de elevação entre células adjacentes é maior que T , parte dessa diferença é movida.....	77
Figura 26 – Algoritmo de colonização espacial.....	82
Figura 27 – Distribuição dos pontos de atração.....	86
Figura 28 – Geração do modelo 3D de uma árvore.....	87
Figura 29 – Terreno resultante da combinação do mapa de elevação com a vegetação.....	88
Figura 30 – Mapa de elevação resultante de zero linhas de serra.....	91
Figura 31 – Mapa de elevação resultante de uma linha de serra.....	92
Figura 32 – Mapa de elevação resultante de cinco linhas de serra.....	92
Figura 33 – Mapa de elevação resultante da modificação do esboço de linhas de serra com duas linhas curvas.....	94
Figura 34 – Mapa de elevação resultante da modificação do esboço de linhas de serra com uma circunferência.....	94
Figura 35 – Mapa de elevação resultante da modificação do esboço de linhas de serra com uma estrela.....	95

Figura 36 – Mapa de elevação resultante da modificação de um esboço de rugosidade	95
Figura 37 – Simulação de erosão no mapa de elevação com cinco iterações	96
Figura 38 – Simulação de erosão no mapa de elevação com dez iterações	97
Figura 39 – Simulação de erosão no mapa de elevação com vinte iterações.....	97
Figura 40 – Ramos de árvores gerados com diferentes constantes	99
Figura 41 – Árvores geradas e respectivos esboços	100

Lista de Tabelas

Tabela 1 – Matriz de comparação de critérios.....	24
Tabela 2 – Matriz de comparação de critérios normalizada.....	24
Tabela 3 – Vetor de prioridades relativas dos critérios	24
Tabela 4 – Valores de IR para matrizes quadradas de ordem n	25
Tabela 5 – Matriz de comparação paritária do critério de facilidade de integração na modelação de mapas de elevação	26
Tabela 6 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de mapas de elevação	26
Tabela 7 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de mapas de elevação	26
Tabela 8 – Matriz de comparação paritária do critério de relevância histórica na modelação de mapas de elevação.....	27
Tabela 9 – Matriz de prioridades na modelação de mapas de elevação.....	27
Tabela 10 – Vetor do <i>ranking</i> das alternativas na modelação de mapas de elevação	27
Tabela 11 – Matriz de comparação paritária do critério de facilidade de integração na modelação de rios, oceanos e lagos	28
Tabela 12 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de rios, oceanos e lagos	28
Tabela 13 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de rios, oceanos e lagos	28
Tabela 14 – Matriz de comparação paritária do critério de relevância histórica na modelação de rios, oceanos e lagos	29
Tabela 15 – Matriz de prioridades na modelação de rios, oceanos e lagos	29
Tabela 16 – Vetor do <i>ranking</i> das alternativas na modelação de rios, oceanos e lagos	29
Tabela 17 – Matriz de comparação paritária do critério de facilidade de integração na modelação de vegetação	30
Tabela 18 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de vegetação	30
Tabela 19 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de vegetação	30
Tabela 20 – Matriz de comparação paritária do critério de relevância histórica na modelação de vegetação	30
Tabela 21 – Matriz de prioridades na modelação de vegetação	31
Tabela 22 – Vetor do <i>ranking</i> das alternativas na modelação de vegetação	31
Tabela 23 – Matriz de comparação paritária do critério de facilidade de integração na modelação de redes de estradas	32
Tabela 24 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de redes de estradas	32
Tabela 25 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de redes de estradas	32

Tabela 26 – Matriz de comparação paritária do critério de relevância histórica na modelação de redes de estradas	32
Tabela 27 – Matriz de prioridades na modelação de redes de estradas	32
Tabela 28 – Vetor do <i>ranking</i> das alternativas na modelação de redes de estradas	33
Tabela 29 – Matriz de comparação paritária do critério de facilidade de integração na modelação de centros urbanos.....	33
Tabela 30 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de centros urbanos.....	33
Tabela 31 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de centros urbanos.....	34
Tabela 32 – Matriz de comparação paritária do critério de relevância histórica na modelação de centros urbanos.....	34
Tabela 33 – Matriz de prioridades na modelação de centros urbanos.....	34
Tabela 34 – Vetor do <i>ranking</i> das alternativas na modelação de centros urbanos.....	34
Tabela 35 – Benefícios e sacrifícios da solução para artistas e desenvolvedores de modelos 3D	43
Tabela 36 – Benefícios e sacrifícios da solução para investigadores e programadores de <i>Software</i>	44
Tabela 37 – Dados de descrição de mapas de elevação	55
Tabela 38 – Dados de descrição de rios, oceanos e lagos.....	55
Tabela 39 – Dados de descrição de vegetação.....	55
Tabela 40 – Dados de descrição de redes de estradas.....	55
Tabela 41 – Dados de descrição de centros urbanos.....	55
Tabela 42 – Valores médios de elevação de mapas de elevação com diferentes números de linha de serra (em metros).....	93
Tabela 43 – Valores médios da variação da elevação entre células adjacentes para diferentes níveis de rugosidade (em metros).....	96
Tabela 44 – Valores médios da variação da elevação entre células adjacentes para diferentes passagens dos algoritmos de erosão (em metros).....	98
Tabela 45 – Constantes usadas nos ramos das árvores geradas	98

Acrónimos

Lista de Acrónimos

.NET	<i>Framework de software</i> desenvolvida pela Microsoft
2D	Duas dimensões
3D	Três dimensões
AHP	<i>Analytic hierarchy process</i>
C#	C Sharp, linguagem de programação desenvolvida pela Microsoft
CPU	<i>Central processing unit</i>
GPU	<i>Graphics processing unit</i>
Unity3D	Motor de jogo multiplataforma
WYSIWYG	<i>What you see is what you get</i>

1 Introdução

1.1 O Que é Uma Técnica Procedimental?

“Técnicas procedimentais são segmentos de código ou algoritmos que especificam algumas características de um modelo ou efeito gerado por computador. Por exemplo, a textura procedimental para uma superfície em mármore não usa uma imagem digitalizada para definir os valores da cor. Em vez disso, usa algoritmos e funções matemáticas para determinar a cor.” (Ebert 2003, p. 1, tradução nossa)

1.2 Descrição do Problema

A modelação procedimental tem sido um tema bastante estudado desde o início da década de 80. Vários algoritmos e técnicas (capítulo 2) têm sido desenvolvidos com o objetivo de replicar elementos da natureza ou de influência do homem na criação de paisagens e terrenos realistas. De forma a reduzir o custo da modelação manual dos ambientes 3D, cada vez mais exigentes para aplicação em videogames ou simuladores, a modelação procedimental tem surgido como uma solução aplicada com sucesso. No entanto, apesar da sua aplicação já comprovada, a utilização de técnicas de modelação procedimental de terrenos é ainda um tema a explorar e existe alguma resistência ao uso de modelos gerados procedimentalmente, o que se deve a algumas dificuldades relacionadas com as soluções existentes a nível individual ou no conjunto.

Um dos problemas conhecidos é a quantidade de soluções existentes para atender às diferentes características de terrenos, algumas delas bastante complexas, ou do seu foco na resolução de um conjunto de problemas muito concretos ou de uma característica isolada no processo de geração.

A falta de controlo sobre o resultado final constitui um outro problema conhecido. As soluções geralmente carecem de uma parametrização ou mesmo um meio de edição direta de

características, de forma a influenciar o modelo gerado e atendendo à criatividade do artista ou do modelo onde se integra o terreno.

Muitas das técnicas existentes carecem de uma noção de dimensão, além de serem destinadas à criação de áreas relativamente pequenas, podendo tornar-se bastante pesadas a nível de utilização de memória e CPU com o aumento do tamanho ou do detalhe do resultado final pretendido.

A necessidade de considerar aspetos importantes na distribuição de componentes, como a existência de fronteiras e proximidade de recursos naturais que são importantes na criação de uma paisagem coerente e realista, constitui também um problema que urge resolver.

O problema mais relevante é a carência de uma solução integradora que permita a junção das várias técnicas existentes para atender às diferentes características na criação de um terreno.

1.3 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de uma *framework* de modelação procedimental que permita a criação de um terreno completo e coerente de forma autónoma. Adicionalmente o utilizador poderá, com o auxílio de ferramentas intuitivas, interagir no processo de modelação, personalizando o resultado final de modo a obedecer aos requisitos impostos. A *framework* deve ser uma ferramenta que integre a modelação das diferentes características de um terreno e abstrair o utilizador do funcionamento interno dos algoritmos. A *framework* deve estimular a utilização de técnicas de modelação procedimental por utilizadores finais e facilitar o desenvolvimento de novos contributos por investigadores e programadores de *software*.

Para atingir este objetivo será necessário estudar as técnicas de modelação procedimental mais relevantes e reconhecidas na comunidade científica, para a criação de terrenos, paisagens entre outras, escolhendo as soluções que melhor se aplicam no processo de integração. Estudar uma forma de organizar as diferentes técnicas de modelação procedimental e encontrar um método para auxiliar a análise, a escolha e a parametrização das diferentes abordagens. Reduzir a complexidade envolvida no estudo e apreciação de técnicas de modelação procedimental.

Ainda no âmbito deste trabalho, avaliar e considerar automaticamente aspetos essenciais do terreno que influenciam o posicionamento de características (por exemplo, o posicionamento de uma cidade na proximidade de recursos naturais, o desvio de estradas para reduzir os custos associados à criação de pontes e túneis na proximidade de rios e montanhas, entre outros).

1.4 Estrutura do Documento

Para auxiliar a leitura deste documento, é feita uma breve referência aos diferentes capítulos que o constituem.

No capítulo 1 é efetuada uma introdução ao conceito da modelação procedimental e quais as dificuldades que motivaram a abordagem deste tema. Os objetivos, com base nas dificuldades estabelecidas, são também expostos para delinear os estudos e as soluções desenvolvidas ao longo deste projeto.

Para fazer uma contextualização sobre o tema da modelação procedimental, são apresentados no capítulo 2 vários contributos que descrevem diferentes abordagens na modelação procedimental de terrenos.

Uma análise sobre os contributos estudados é efetuada no capítulo 3, de forma a encontrar abordagens que fundamentem o *design* da solução proposta. Neste capítulo é também delimitado o âmbito da solução, pela apresentação dos pressupostos assumidos, limitações e tecnologias a utilizar. A aplicação de um método analítico para auxiliar na escolha de contributos na modelação procedimental de terrenos é apresentada.

Contextualizando a *framework* desenvolvida num modelo de negócio, é efetuada a análise de valor obtida da solução a desenvolver, no capítulo 4.

No capítulo 5 é concebido o *design* da solução utilizada na implementação da *framework* de modelação procedimental. Neste capítulo são explicados os vários componentes, estruturas de dados, fluxo de informação e funcionamento da *framework*.

A implementação da *framework* de modelação procedimental é apresentada no capítulo 6, demonstrando as diferentes vistas da interface para o utilizador, configuração e processo de modelação. É também descrito o funcionamento das técnicas de modelação procedimental implementadas e quais os meios de parametrização utilizados.

Para fundamentar a solução desenvolvida, no capítulo 7, é efetuada uma avaliação dos resultados das técnicas de modelação procedimental implementadas e o impacto que a parametrização proposta tem nos modelos gerados.

No capítulo de conclusão, capítulo 8, são avaliados os objetivos atingidos pelas soluções desenvolvidas em contraste com os problemas e dificuldades descritos no capítulo de introdução. São também abordadas as limitações encontradas durante o desenvolvimento da *framework* e é efetuado um levantamento de trabalhos futuros que podem derivar deste projeto.

2 Estado da Arte

Neste capítulo é feita uma abordagem sobre os contributos mais relevantes para o tema da modelação procedimental de terrenos. Seguindo a separação definida noutros estudos sobre o tema (Smelik et al. 2009; Smelik et al. 2014), os contributos podem ser agrupados nas seguintes características: superfície do terreno, rios, oceanos e lagos, vegetação, redes de estradas e ambientes urbanos. São também abordados contributos na simulação de erosão pela especial relevância no tema deste trabalho. Por fim, são estudadas as *frameworks* para criação de terrenos existentes bem como as propostas para soluções de integração e criação de *frameworks*.

2.1 Superfície do Terreno

A criação de superfícies do terreno foi um dos temas inicialmente abordados e com importantes contributos nesta área desde o início dos anos 80.

2.1.1 Mapa de Elevação

Um mapa de elevação é uma grelha bidimensional onde cada célula indica um valor de altitude e está entre as representações mais utilizadas para descrever um terreno base. Numa escala grosseira, esta representação é pouco exigente a nível computacional e não apresenta grandes desvantagens ao nível do realismo do terreno gerado.

Entre as técnicas clássicas para gerar mapas de elevação, são de salientar as técnicas baseadas em subdivisão e as técnicas baseadas em geradores de ruído.

Dos algoritmos mais antigos para gerar um mapa de elevação simples, os algoritmos de subdivisão, baseados no algoritmo *midpoint displacement*, são descritos nas suas diferentes variações por Miller (1986): *triangular edge*, *diamond square* e *square square*. Destes três, destaca-se o *diamond square*, inicialmente introduzido por Fournier et al. (1982). Trata-se de um algoritmo de subdivisão fractal e estocástico que permite, a cada iteração, aumentar o nível de detalhe do terreno. Tendo como base o algoritmo *midpoint displacement*, o *diamond square* foi adaptado com o principal objetivo de criar terrenos 3D e de forma a resolver o efeito de xadrez, problema conhecido do algoritmo em que se baseia quando aplicado em ambientes 3D. Este algoritmo funciona a partir de uma grelha de lado $2n+1$ onde inicialmente são atribuídos valores arbitrários ou aleatórios nos quatro cantos da grelha (Figura 1). No início de uma iteração, onde por interpolação dos valores dos cantos e acrescentando um desfasamento aleatório, o valor do ponto central é determinado. Novamente, por interpolação acrescida de desfasamento aleatório, são obtidos os valores centrais dos lados da grelha utilizando o valor

central anteriormente calculado e o valor dos cantos adjacentes. Uma nova iteração é iniciada partindo dos valores calculados na iteração anterior e subdividindo a grelha.

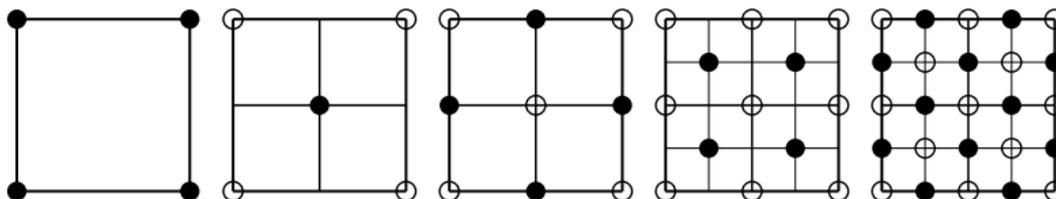


Figura 1 – Representação de duas iterações do algoritmo *diamond square* (Olsen 2004, p. 3)

Poisson faulting é uma técnica baseada em geradores de ruído que gera terreno em aproximação ao movimento *browniano* fracionário. Apresenta a vantagem de ser bastante adequado para a aplicação em esferas. Esta técnica consiste em “[...] aplicar um deslocamento aleatório Gaussiano (falhas ou funções escalonadas) a esferas ou planos com uma distribuição de *Poisson* de intervalos” (Musgrave et al. 1989, p. 42, tradução nossa). O resultado é uma superfície *browniana* que pode ser usada como mapa de elevação. Este algoritmo tem uma complexidade $O(n^3)$ o que o torna demasiado lento para aplicações em tempo real.

Um outro método estocástico de geração de mapas de elevação é o método desenvolvido por Musgrave et al. (1989), baseado em geradores de ruído. Este método consiste no “dimensionamento e soma de sucessivas oitavas de ruído de frequência cada vez maior num mapa de elevação” (Smelik et al. 2014, p. 3, tradução nossa). O resultado é um terreno montanhoso de aspeto natural. Um exemplo de gerador de ruído utilizado, é o ruído de Perlin (Perlin 1985) que foi desenvolvido com o objetivo de criar texturas realistas.

Uma grande lacuna nestes métodos de geração de terreno é o facto de o processo de geração ser pouco controlável, estando o controlo reservado à definição de variáveis iniciais pouco intuitivas. Por esse motivo, contributos mais recentes têm-se baseado em técnicas clássicas não só para melhorar o seu resultado mas para fornecer meios para aumentar a controlabilidade. Também é de salientar a grande homogeneidade dos terrenos gerados, tornando os resultados menos realistas em terrenos de grandes dimensões, daí um grande foco na introdução de características discretas, como a introdução de zonas mais acidentadas e na deformação do terreno para as acomodar.

Para resolver o problema do posicionamento de características num terreno, Stachniak e Stuerzlinger (2005) propuseram um método onde restrições no processo de criação do terreno são indicadas através de máscaras. Essas restrições dão origem a operações de deformação a aplicar no terreno até este atender a todas as restrições.

Schneider et al. (2006) propuseram um editor interativo com uma “[...] interface WYSIWYG [onde] o utilizador pode literalmente pintar as funções base [do gerador de ruído]

representadas por imagens em escala de cinza” (Schneider et al. 2006, p. 4, tradução nossa). Essas imagens passam a substituir grelhas de ruído de Perlin na geração de terreno.

Belhadj e Audibert (2005) desenvolveram uma técnica para a criação de um terreno com serras e rios usando o algoritmo de subdivisão *triangular edge*. Nesta técnica, as linhas que representam as serras são criadas com recurso a “partículas de serra [colocadas aleatoriamente no mapa de altitude e que são sujeitas a] impactos laterais de forma a descrever um movimento *Browniano* fracionário” (Belhadj & Audibert 2005, p. 1, tradução nossa). O mapa de elevação é preenchido com sucessivas curvas de Gauss centradas e perpendiculares em cada ponto do percurso das linhas de serra. Com base neste terreno temporário, partículas de rio são colocadas aleatoriamente nas linhas de serra onde, por ação de um método físico simulando a ação da gravidade, descrevem um percurso de rio. Partindo dos valores de altitude das linhas de serra e do percurso de rios calculados, um novo algoritmo introduzido, o *midpoint displacement's inverse* (MDI), é utilizado de forma a calcular as altitudes de forma inversa ao *triangular edge* (Figura 2). Estes pontos de elevação calculados de forma inversa permitem a utilização de um algoritmo de subdivisão resolvendo problemas de continuidade no terreno junto das linhas de serra e dos percursos dos rios.

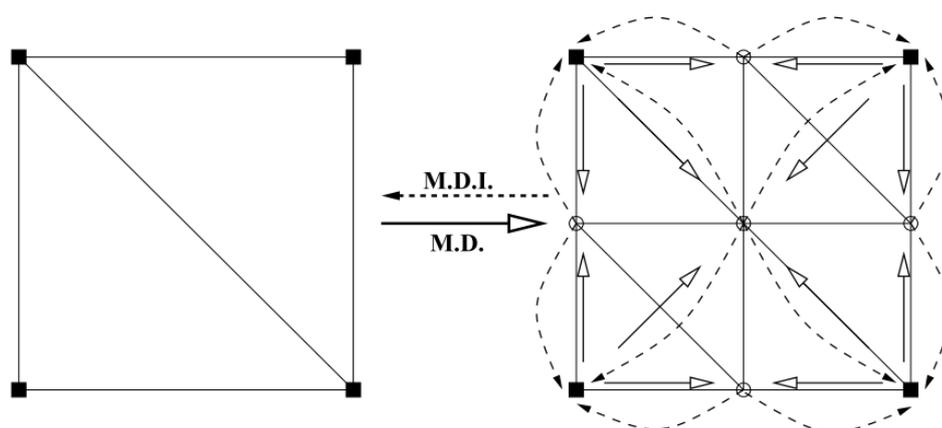


Figura 2 – Representação do algoritmo (MDI) (Belhadj & Audibert 2005, p. 3)

Uma técnica desenvolvida por Zhou et al. (2007) parte de um mapa de elevação de um terreno real e de um esboço com linhas representando características do terreno, como linhas de serra e vales, para gerar um novo terreno. O algoritmo combina de forma suavizada características extraídas do exemplo do mapa de elevação e do esboço de linhas de características num novo mapa de elevação.

Doran e Parberry (2010) apresentaram uma prova de conceito onde são usados agentes de *software* para gerar determinadas características de um terreno: linhas costeiras, praias, montanhas, rios e suavização. Para melhorar a controlabilidade da criação de terrenos, foram introduzidos parâmetros que permitem influenciar o funcionamento dos agentes de *software* e abstraído os processos envolvidos para permitir que o controlo seja efetuado por utilizadores sem conhecimentos técnicos.

Génevaux et al. (2013) apresentaram uma *framework* que permite a criação de terrenos através de conceitos baseados na hidrologia. O processo de geração tem como base dois esboços que contêm parâmetros fornecidos pelo utilizador, o primeiro esboço indica o tipo de rios principais e o seu percurso, o segundo esboço indica as zonas montanhosas e as zonas mais regulares. Durante o processo de criação os algoritmos guardam a informação em “árvores de construção” em que os nós internos da árvore definem combinações de operações e os nós folha definem características do terreno.

2.1.2 Representação Multicamada

Representações multicamada de mapas de elevação ou por *voxels* têm a vantagem de simular características de terreno que as representações de mapas de elevação simples não permitem, como a formação de grutas, arcos e beirais rochosos. Nesta representação, a altitude de cada célula é constituída pela altitude de camadas de terreno com diferentes materiais, como por exemplo, rocha, areia, terra ou mesmo ar (como no caso de grutas, arcos e beirais rochosos). Este tipo de descrição de terreno pode melhorar o realismo na aplicação de algoritmos de erosão, como abordado na secção 2.7. Devido à quantidade de informação mantida por esta representação, a maior utilização de memória constitui uma desvantagem quando comparado com a representação simples. No entanto, essa desvantagem é compensada quando um resultado com maior detalhe e em menor escala é necessário.

Uma referência neste tipo de abordagem de mapa de altitude é a *framework* Arches, apresentada por Peytavie et al. (2009). Nesta *framework*, é utilizada uma estrutura de dados compacta que contém informação volumétrica dos vários materiais que constituem o terreno. A estrutura de dados é representada por uma grelha bidimensional em que cada célula é uma *stack* de camadas de materiais. As camadas podem ser ar, água, areia, rocha-mãe e pedras. Nesta *framework* é descrito um processo de estabilização das camadas de terreno onde o volume de diferentes materiais pode ser reordenado ou movido para células adjacentes tendo em conta o ângulo de repouso do material.

2.2 Rios, Oceanos e Lagos

Na modelação procedimental de rios e massas de água, existem dois tipos de abordagem: ou o processo de criação é integrado na geração dos mapas de elevação, ou então, é iniciado tendo em conta um mapa de elevação já criado, alterando-o. Comparado com as contribuições para a criação de mapas de elevação, que apresentam um grande conjunto de soluções existentes, o tema da geração de rios ainda carece de contributos significativos, tal com é observado por Smelik et al. (2014). Além do mais, soluções para gerar oceanos e lagos são escassos ou inexistentes, sendo muitas vezes, no caso de oceanos, definida uma elevação a partir da qual se considera que altitudes inferiores representam oceano.

Num dos primeiros contributos neste tema, Kelley et al. (1988) descrevem um método onde uma rede de rios é criada partindo de uma linha inicial que é subdividida recursivamente. Usando esse “[...] esqueleto de mapa de elevação [, o resto das elevações é] preenchido usando uma função de interpolação de dados dispersos [, o formato da rede de rios é influenciado] pelo tipo de clima e o tipo de material do solo” (Smelik et al. 2009, p. 3, tradução nossa).

Na técnica desenvolvida por Belhadj e Audibert (2005), abordado na secção 2.1.1, quanto à geração de mapas de elevação, partículas de rio são colocadas aleatoriamente numa linha de serra de altitude elevada. Essas partículas definem o percurso de rios quando sujeitas a um método físico, simulando a ação da gravidade num mapa de elevação preenchido por funções de Gauss centradas nas linhas de serra. Essas linhas de rios podem colidir entre si, desencadeando mecanismos de *backtracking* que influenciam a largura e a profundidade dos rios. Os rios e as linhas de serra servem de base para a geração dos restantes pontos de altitude do terreno.

Na solução apresentada por Smelik et al. (2011), onde o processo de criação de terreno é influenciado por “esboços procedimentais”, é resolvido o problema da controlabilidade da criação de rios. O “*designer* pode introduzir uma sequência de pontos de controlo no terreno indicando o percurso pretendido [e um algoritmo altera o terreno de forma a criar um percurso] viável e plausível” (Smelik et al. 2014, p. 5, tradução nossa).

Na *framework* apresentada por Génévaux et al. (2013), referida quanto à criação de terrenos na secção 2.1.1, uma rede de rios é criada inicialmente, tendo como base dois esboços que indicam o tipo de rios principais e o seu percurso e as zonas mais acidentadas ou regulares. Seguindo modelos baseados na hidrologia, o percurso dos rios é traçado partindo da foz em direção às nascentes, criando um grafo geométrico. O processo começa por selecionar os nós iniciais no contorno do terreno ou no início do percurso indicado pelo esboço dos rios. O grafo é sucessivamente incrementado e elevado usando um método probabilístico em que nós candidatos são selecionados segundo heurísticas que têm em conta a elevação do nó e um índice de prioridade.

2.3 Vegetação

O facto de a estrutura de uma planta poder ser “vista como uma estrutura modular que consiste em unidades relativamente independentes e repetidas, tal como folhas, flores, rebentos e segmentos de ramificações” (Prusinkiewicz 2000, p. 86, tradução nossa) justifica que a modelação procedimental de plantas recorra vulgarmente à utilização de *L-systems*. *Lindenmeyer’s system* ou *L-system*, nome atribuído em homenagem a Lindenmayer e seu trabalho (Lindenmayer 1968), é um sistema de reescrita paralela constituída por um conjunto de regras que definem uma construção numa linguagem formal.

“Um modelo de *L-System* é especificado pela listagem dos três componentes seguintes:

- Um conjunto de tipos de módulos, ou alfabeto, do *L-System*;
- Um conjunto de regras de reescrita, ou produções, que capturam o comportamento dos módulos individuais dentro de determinado intervalo de tempo;
- A estrutura inicial, ou axioma;” (Prusinkiewicz 2000, p. 86, tradução nossa).

Prusinkiewicz et al. (1988) apresentaram um método para a modelação de plantas herbáceas usando *L-systems* que depois são processados e convertidos na forma geométrica.

Weber e Penn (1995) apresentaram um método para criar e renderizar árvores. As árvores são geradas partindo de ramos de forma aproximadamente cônica que são subdivididos e rotações são aplicadas, formando curvas. Clones dos ramos são iniciados a meio de ramos existentes, adotando as propriedades do ramo de origem, formando ramificações recursivamente. Folhas são adicionadas a partir de um nível de recursividade parametrizada. É também introduzida uma solução para restringir o crescimento da árvore para um determinado espaço e para orientar as folhas tendo em conta a otimização da captação de luz. Esta solução inclui um meio para indicar parâmetros que influenciam vários fatores durante o processo de criação.

Uma abordagem, tendo em conta a criação de um ecossistema de plantas usando *L-systems* foi proposta por Prusinkiewicz (2000) em que três tipos de comunicação são identificados durante o processo de criação: comunicação por linhagem, interação endógena e interação exógena. Comunicação por linhagem é a comunicação mais simples e independente do contexto, diz respeito à influência do módulo antecessor na construção da planta. Interação endógena pode, na presença de parâmetros, influenciar a construção da planta para que, por exemplo, seja considerado o fluxo e existência de nutrientes num determinado módulo ou mesmo a produção de fotossíntese. Interação exógena tem em conta o contexto onde a planta é integrada que pode influenciar a construção da planta ou a planta pode influenciar a construção de outras plantas, originando um *loop* de *feedback*. Um exemplo de interação exógena pode ser a influência da luz e da sombra com origem noutros ramos da planta que influenciam a posição de um novo ramo durante o processo e construção. Outro exemplo de interação exógena é a competição por espaço de plantas ou ramos adjacentes.

Um método descrito por Rodkaew (2003) propõe a construção de árvores recorrendo a sistemas de partículas. Neste método, partículas são distribuídas num volume delimitado por uma fronteira externa e interna perto do limite que define o formato da copa da árvore. As partículas são depois movidas, em iterações, convergindo-as para um ponto central representando a base do tronco principal. A proximidade de outras partículas aplica um efeito de atração que altera a sua trajetória. Duas partículas, a partir de um determinado valor de proximidade, podem fundir-se criando uma nova partícula com um movimento resultante da combinação dos seus movimentos e simulando a ramificação. As trajetórias descritas pelo

movimento das partículas são usadas na construção dos ramos da representação gráfica da árvore. Neste trabalho é abordado, ainda, o efeito da menor incidência da luz em determinadas zonas da árvore, resultante da sombra provocada pela proximidade de outras árvores e posição da luz. A intensidade da luz é calculada usando *ray-tracing* e é aplicada no cálculo da densidade das partículas em determinada área da copa. A influência da luz constitui uma interação exógena e permite obter um modelo de árvore mais coerente com o que acontece na natureza.

Um outro contributo influenciado por sistemas de partículas, o algoritmo de colonização espacial (Runions et al. 2007), permite a criação de árvores com controlo sobre o resultado final. Pontos de atração são distribuídos numa área 3D representando a copa da árvore. Partindo da base da árvore, nós são gerados na direção dos pontos de atração mais próximos e removendo-os iterativamente (Figura 3). Esses nós representam segmentos dos ramos da árvore. Este método apresenta bastante controlabilidade no processo de criação através da indicação do formato da copa da árvore, onde os pontos de atração são distribuídos, e da indicação de dois parâmetros: a distância de influência e a distância de remoção dos pontos de atração. O primeiro parâmetro indica qual a distância mínima entre um nó e um ponto de atração a partir da qual um nó é atraído, o segundo parâmetro, indica qual a distância a partir da qual o ponto de atração deve ser removido.

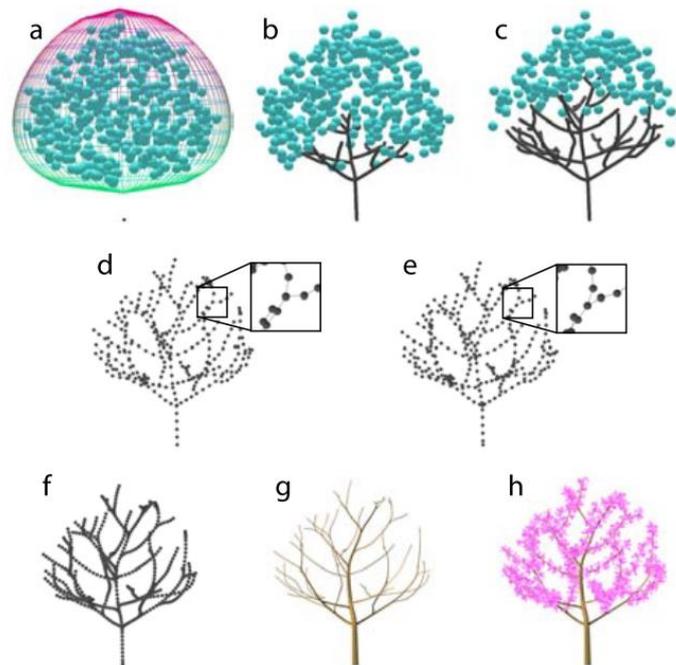


Figura 3 – Os vários passos do algoritmo de colonização espacial (Runions et al. 2007, p. 2)

Em contributos mais recentes, como os modelos apresentados por Palubicki et al. (2009), interações exógenas avançadas são abordadas, de forma a criar mecanismos para a organização de árvores e arbustos. Nestes modelos, ramos e brotos gerados são fortemente influenciados pela competição pela luz e pelo espaço.

Plastic trees, uma técnica abordando interações exógenas (Pirk et al. 2012), permite a modelação de árvores que se adaptam às condições de espaço, obstáculos sólidos e exposição de luz.

2.4 Redes de Estradas

A modelação procedimental de estradas tem sido abordada maioritariamente no contexto da modelação procedimental de ambientes urbanos, por esse motivo, o foco desta secção será no tema da modelação de estradas secundárias e autoestradas, ficando o tema da modelação de estradas em centros urbanos reservado à secção 2.5.

Drive (McCrae & Singh 2009) é um sistema para criação de estradas que permite a edição por intermédio do esboço de um mapa. As estradas são definidas por curvas desenhadas que são ajustadas e adaptadas ao mapa de elevação. É possível redesenhar o mapa de forma a estender as estradas existentes. Ferramentas são disponibilizadas para remover porções de estrada. Um conjunto de regras de desenho permitem indicar o modo como duas estradas se cruzam, de forma a indicar se as estradas devem cruzar por intermédio de uma ponte ou juntar criando um cruzamento.

Um método apresentado por Bruneton & Neyret (2008) permite popular um terreno com estradas, rios, lagos e prados. Os elementos a adicionar são definidos vectorialmente e são combinados de forma coerente no terreno numa estrutura de dados baseada em *quadtrees*. Uma técnica para a rápida renderização do terreno é proposta usando o GPU.

Galin et al. (2010) propuseram um método para gerar automaticamente estradas tendo em conta funções de custo. Estas funções são combinadas com o algoritmo A* para obter o percurso otimizado e têm em conta vários parâmetros do terreno, como o declive, lagos, rios, montanhas e florestas. Noutro contributo (Galin et al. 2011), um método procedimental para criar redes de estradas hierárquicas para unir cidades num terreno foi criado. A hierarquia é constituída por três tipos de estradas: autoestradas, estradas principais e estradas secundárias. Funções de custo têm em conta o tipo de estrada de forma a evitar interseções desnecessárias e estimulando a utilização de segmentos de estradas superiores hierarquicamente sempre que possível.

Um método para a geração de traçados rodoviários apresentado por Campos et al. (2014) segue um modelo hierárquico em níveis onde traçados de estradas são iterativamente refinados. O método parte da definição de uma ligação entre dois pontos, onde um corredor do traçado de estrada é identificado com uma largura dependente da extensão da ligação ou, no último nível, também dependente da velocidade base do projeto. Nesse corredor é adicionado um vértice que subdivide a ligação inicial e permite a identificação de dois novos corredores de largura inferior. Este processo é repetido iterativamente para cada nova ligação até gerar uma “[...] definição com nível de detalhe adequado em alinhamentos retos” (Campos et al. 2014, p. 3). Em cada nível, várias alternativas no posicionamento dos vértices são

identificadas e escolhidas tendo em conta a futura implantação de curvas que cumpram as normas de projeto e tendo em conta funções de custo que consideram os seguintes fatores: zona classificada, construção em ponte, construção em túnel, movimento de terras, construção de via e custo de utilização. Posteriormente, um traçado em planta com curvas compostas é desenvolvido tendo como base a definição em alinhamentos retos. Por fim, o traçado em perfil longitudinal (traçado em altimetria) é desenvolvido de forma a “[...] adequar a cota da via ao relevo do terreno” (Campos et al. 2014, p. 6). Este método foi desenvolvido com o intuito de gerar ambientes rodoviários realistas para simulação de condução.

Noutro contributo (Campos, Leitão, Pereira, et al. 2015), um método para definição de uma rede topológica de estradas que permite a interação do utilizador no processo de criação foi descrito. Neste método, dois módulos são identificados, o módulo de nós e o módulo de estradas topológicas. O módulo de nós permite a criação e distribuição dos nós e recebe como parâmetro a definição do terreno onde vão ser distribuídos os nós e o número de nós a gerar. Os nós são distribuídos tendo em conta fatores naturais do terreno, tais como, a elevação a proximidade de zonas costeiras e redes de fluviais. Restrições na distribuição de nós também são consideradas, tais como, redes de estradas previamente criadas, edifícios e áreas de cultivo ou florestas. O módulo de estradas topológicas recebe como parâmetro de entrada a definição dos nós gerados no módulo de nós e permite gerar a rede topológica de estradas. Dependendo do nível de importância, os nós são organizados nas seguintes tipologias: cidades, vilas e aldeias. Estas tipologias permitem a definição da topologia das redes de estradas. A rede de autoestradas é criada da ligação de cidades, a rede de estradas secundárias é obtida da ligação das vilas e a rede rural é obtida da ligação das aldeias. O utilizador pode alterar, em cada fase, o processo de geração, criando ou alterando nós e ligações ou alterando parâmetros que influenciam a classificação.

Foi proposta, noutro contributo (Campos, Leitão & Coelho 2015), uma metodologia hierárquica por camadas que integra a geração dos modelos 3D de estradas com a definição dos modelos semânticos. São identificados cinco módulos que constituem a metodologia proposta, o módulo de nós, o módulo de estradas topológicas, o módulo de *centerlines*, o módulo de ambiente rodoviário e o módulo de geração de modelos. O módulo de nós gera nós de interligação com uma localização e nível de importância. Os nós são usados pelo módulo de estradas topológicas, onde são organizados por tipologia tendo em conta o seu nível de importância, dando origem a uma definição topológica da rede de estradas (Campos, Leitão, Pereira, et al. 2015). O módulo de *centerlines* gera a definição dos percursos das estradas (Campos et al. 2014). O módulo de ambiente rodoviário adapta o modelo do terreno à rede de estradas gerada, simulando a movimentação de solo necessária à construção de rodovias. Por fim, o módulo de gerador de modelos cria os modelos 3D das estradas e do terreno.

2.5 Ambiente Urbano

A modelação procedimental de ambientes urbanos tem sido um tema bastante estudado nos últimos 20 anos. As cidades são geralmente criadas partindo de uma rede de estradas e preenchendo os espaços com edifícios. Para a criação de estradas, várias técnicas são abordadas, desde criação de simples ruas paralelas (Greuter et al. 2003), utilização de *L-systems* (Parish & Müller 2001; Coelho et al. 2007) ou utilização de campos tensoriais (Chen et al. 2008).

CityEngine, um sistema que permite a criação de cidades (Parish & Müller 2001), baseia-se em dados recebidos por máscaras indicando a localização da água, mapa de elevação e densidade populacional. Dois *L-systems* diferentes são usados na geração de cidades completas, sendo responsáveis pela geração de ruas e edifícios.

No método apresentado por Lechner et al. (2006), um terreno é gerado dado um mapa desenhado com a descrição do terreno contendo zonas bem definidas, tais como, zonas residenciais, zonas comerciais, zonas industriais e parques.

Num contributo de Coelho et al. (2007), foi apresentado uma extensão de *L-system* que considera a informação geoespacial de objetos, incluindo-a na definição das regras de produção para a modelação de ambientes urbanos.

A *framework* introduzida por Chen et al. (2008), define um *pipeline* de modelação constituído por três camadas: gerador de campo tensorial, gerador de grafo de ruas e gerador de geometria 3D. O processo é iterativo dando a possibilidade de influenciar as ruas geradas utilizando ferramentas, tais como, pinceis e aplicação de ruído, edição do grafo de ruas ou mesmo movendo e adicionando ruas diretamente. Máscaras são utilizadas para influenciar o aumento da densidade de ruas perto do centro da cidade. Resultados apresentados mostram semelhanças consideráveis entre mapas reais de cidades e mapas gerados com a *framework*.

Emilien et al. (2012) introduziram um método para gerar pequenas vilas. O método começa por gerar uma rede de ruas e vai iterativamente construindo edifícios tendo em consideração um conjunto de critérios: elevação do terreno, inclinação, acesso à rua, distância aos vizinhos, e edifícios especiais (igreja e fortes).

2.6 Integração e *Frameworks*

Apesar de existirem bastantes contributos no tema da modelação procedimental de terrenos, as técnicas e abordagens existentes são vulgarmente focadas num aspeto isolado no processo de criação. Essa lacuna cria uma área promissora no tema da modelação procedimental que já apresenta algumas soluções.

Smelik et al. (2008) propuseram um modelo para a criação de uma *framework* de modelação procedimental. Neste modelo, técnicas de modelação procedimental fazem parte

de camadas que são responsáveis por um conjunto de características de terreno e são organizadas num fluxo de execução. O utilizador define o terreno que pretende através de esboços e pode fazer correções em pontos intermédios do fluxo de execução. Ações corretivas no terreno criado podem originar a geração de todo o terreno ou apenas das camadas envolvidas na alteração.

Baseado na proposta anterior, Smelik et al. (2010; 2011) apresentaram uma *framework* de criação de terrenos para jogos de treino militar. Nesta *framework*, uma interface para a criação do esboço do terreno é utilizada para indicar a localização de elementos importantes, como: florestas, montanhas, cidades e vilas. Um conjunto de técnicas de modelação procedimental funcionam de forma integrada para modelar o terreno. A interface de esboço é constituída por uma grelha, a “grelha de ecótipos”, em que a cada célula pode ser atribuída uma cor representando o tipo de terreno e quais os limites de elevação. Polígonos podem ser desenhados sobre essa grelha para indicar a localização de praias, vegetação, oceanos, prados e colinas.

2.7 Erosão

Não sendo uma técnica de modelação procedimental, a simulação de erosão permite melhorar de forma significativa o realismo dos mapas de elevação. Estas técnicas são vulgarmente divididas em dois tipos (Musgrave et al. 1989; Benes & Forsbach 2002): erosão térmica e erosão hidráulica.

A erosão térmica simula a deposição de sedimentos em pontos de menor altitude com origem no desprendimento de rochas em altitudes mais elevadas com os ciclos de congelamento e degelo e variação de temperaturas. Os sedimentos são movidos para pontos adjacentes segundo um *threshold* de inclinação, o ângulo de talus, em várias iterações.

A erosão hidráulica simula a deposição de sedimentos transportados nas águas pluviais, sendo estes sedimentos depositados de acordo com uma constante de saturação da água, saturação essa que pode ser afetada pela constante de evaporação e pela constante de precipitação.

Quando aplicado em mapas de elevação com várias camadas (Benes & Forsbach 2001), o realismo do resultado dos algoritmos de erosão pode ser elevado. Os materiais que constituem o terreno têm diferentes propriedades. Materiais mais moles, como a terra ou areia, dissolvem-se ou separam-se mais facilmente, sofrendo maior erosão do que materiais mais duros, como a rocha.

2.8 Resumo do Capítulo

Neste capítulo foi efetuado o estudo de contributos existentes no tema da modelação procedimental de terrenos, servindo de introdução ao tema deste trabalho. Os diferentes contributos foram separados segundo a organização proposta por Smelik et al. (2009; 2014) e referidos por ordem cronológica. As técnicas de modelação podem ser separadas pelas características de terreno que abordam, tais como superfície do terreno, rios, oceanos e lagos, vegetação, redes de estradas e ambiente urbano.

Deste estudo, foi possível observar que o tema da modelação procedimental tem um número de contribuições extensa, tendo evoluído ao longo de mais de quatro décadas. Tal como evidenciado na separação efetuada por característica de terreno, é possível observar que algumas destas características apresentam um menor número de contributos e constituem uma área a explorar. Quanto à criação de soluções integradoras de técnicas de modelação procedimental, apesar de existirem abordagens a esse tema, as soluções são insuficientes, limitadoras e não apresentam um método universal que permita a comunicação entre diferentes técnicas de modelação procedimental.

3 Análise do Problema

Neste capítulo é feita uma análise do problema de forma a preparar o leitor e justificar a solução proposta no capítulo 5. Os trabalhos abordados no capítulo 2, bem como os pressupostos e limitações são fatores que influenciam o *design* da solução.

3.1 Pressupostos

Ao longo deste trabalho é utilizada com frequência a expressão “técnica de modelação procedimental”. Esta expressão tem como objetivo simplificar a referência a um trabalho, algoritmo ou técnica que aplique o conceito de modelação procedimental com o objetivo de criar uma ou mais características de um terreno. É considerada, portanto, uma técnica de modelação procedimental cada contributo ou conjunto de contributos abordados no capítulo 2 que descrevem um procedimento semelhante e geram as mesmas características de terreno.

O objetivo deste trabalho resume-se sucintamente a resolver dois problemas na modelação procedimental de terrenos: o reduzido controlo e a dispersão de técnicas de modelação existentes. Dessa forma, o foco principal é o de obter o melhor e o mais completo resultado na integração das técnicas que melhor se adequem, permitindo a intervenção do utilizador no processo. Alguns aspetos, tal como a velocidade de execução dos algoritmos, são considerados na escolha das técnicas de modelação mas não são determinantes. Além disso, e apesar de algumas soluções apresentadas no capítulo 2 terem um resultado bastante realista, a sua complexidade pode não justificar o trabalho envolvido na implementação e integração. Estes e outros fatores são descritos na secção 3.8, onde o método usado para apoiar a decisão da escolha das técnicas de modelação estudadas é descrito.

A *framework* integradora deve permitir e facilitar a sua utilização fora do âmbito deste trabalho. Por esse motivo, a *framework* não deverá impor nenhuma restrição quanto às tecnologias usadas pelas técnicas de modelação procedimental. Estas técnicas devem permitir a sua implementação de forma a utilizar os recursos de *hardware* existentes. Os métodos de modelação procedimental devem ser tratados como módulos independentes, executados pelo núcleo da *framework* e que seguem protocolos de comunicação bem definidos para permitir a sua integração. Este pressuposto pretende facilitar a contribuição de técnicas de modelação procedimental, onde um determinado método é tratado como um *plugin* que recebe informação de entrada bem definida e exporta os resultados. As tecnologias a utilizar na implementação das técnicas de modelação procedimental, no âmbito deste trabalho, são abordados na secção 3.7.

A *framework* integradora deve ser independente da interface com o utilizador. São definidos protocolos para que as técnicas de modelação procedimental possam descrever o

tipo de *input* esperado do utilizador. A especificação deste protocolo deve descrever ferramentas tais como: máscaras, introdução de parâmetros e opções de escolha. O módulo de interface com o utilizador deve interpretar esta informação e preparar as ferramentas necessárias. Da mesma forma, o protocolo deve descrever o tipo de *output* de dados de pré-visualização intermédios, no processo de criação.

A *framework* deve permitir uma fácil escolha entre técnicas de modelação procedimental alternativas para que o utilizador final possa utilizar a técnica que melhor se adequa ao resultado pretendido.

3.2 Limitações

Tendo em conta a quantidade de técnicas de modelação procedimental de terrenos existente e o tempo para a elaboração deste trabalho, só um subconjunto de técnicas poderão ser implementadas e integradas em algumas das características de terreno. Apesar dessa limitação, todas as características do processo de criação de um terreno serão estudadas. Na secção 3.8 é descrito o método que será utilizado para apoiar a decisão na escolha de técnicas de modelação procedimental.

3.3 Contributos Importantes

Alguns dos contributos apresentados no capítulo 2 referem e solucionam, parcialmente, problemas que motivaram o tema deste trabalho. Por esse motivo é importante a inclusão desses trabalhos nesta análise e devem também servir de fundamento para a solução apresentada.

3.3.1 *Framework* de Modelação Procedimental

As *frameworks* de modelação procedimental abordadas na secção 2.6 (Smelik et al. 2010; Smelik et al. 2011) integram várias técnicas existentes e permitem a construção de um terreno completo com a utilização de ferramentas declarativas para resolver o problema da controlabilidade. Apesar destas *frameworks* terem um elevado potencial, outras abordagens podem trazer um valor significativo para o tema da modelação procedimental de terrenos.

O modelo que serviu de base para as *frameworks* referidas no parágrafo anterior (R M Smelik et al. 2008) enquadra-se com a natureza modular da solução pretendida para este projeto. A arquitetura apresentada potencia a que se mantenha uma separação bem definida entre a implementação das diferentes técnicas de modelação procedimental, já que um *workflow* do processo de criação é definido e constituído por várias camadas responsáveis pela criação de uma determinada característica do terreno. Sendo assim, este modelo pode servir de base para a definição da arquitetura da *framework* a desenvolver.

3.3.2 Controlabilidade

Algumas técnicas de modelação procedimental apresentam soluções para a questão da controlabilidade, como a indicação de parâmetros bem definidos (Weber & Penn 1995), utilização de máscaras (Stachniak & Stuerzlinger 2005), esboços do terreno (Génevaux et al. 2013) ou mesmo interfaces com o utilizador completas (Schneider et al. 2006). Estes exemplos podem servir de modelos para a especificação de ferramentas de edição e outras técnicas podem ser adaptadas para que também possam ser parametrizadas, melhorando a controlabilidade do processo de criação.

3.4 *Workflow* do Processo de Criação

Um *workflow* no processo de criação tal como apresentado por R M Smelik et al. (2008), surge como uma solução eficaz para organizar o processo de criação. Nesse caso, o *workflow* é definido por camadas, cada uma representando apenas a criação de uma característica de terreno e com uma ordem específica: mapa de elevação, rios oceanos e lagos, vegetação, rede de estradas e ambiente urbano. Dois trabalhos mais recentes baseados nesta proposta (Smelik et al. 2010; Smelik et al. 2011) descrevem uma *framework* para a modelação procedimental de terrenos em que as técnicas de modelação utilizadas são bem definidas para cada camada do *workflow*.

Neste trabalho, e seguindo os pressupostos identificados na secção 3.1, as técnicas de modelação procedimental devem ser implementadas de forma independente do resto da solução para que possam ser adicionadas, removidas e escolhidas pelo utilizador final para cada característica do terreno. No entanto, essa flexibilidade constitui uma dificuldade na aplicação do modelo apresentado por R M Smelik et al. (2008), já que algumas das técnicas de modelação procedimental apresentadas no capítulo 2 seguem uma ordem diferente das características de terreno (Génevaux et al. 2013), modelam mais do que uma característica (Belhadj & Audibert 2005; Doran & Parberry 2010; Bruneton & Neyret 2008) e utilizam diferentes estruturas de entrada e saída de dados.

Desta forma, torna-se necessário desenvolver uma solução que permita a flexibilidade pretendida quanto à escolha das técnicas de modelação procedimental mantendo a organização e a abrangência de um *workflow*.

3.5 Estruturas de Dados

A existência de um grande número de técnicas de modelação procedimental, utilizando diferentes meios para a modelação de característica de terreno, dificulta a comunicação dos dados gerados entre si. Essas técnicas de modelação procedimental criam e recebem dados que podem ser de diferentes tipos e ter diferentes estruturas. As técnicas de modelação

procedimental, para funcionarem em conjunto, necessitam de informação base como *input* que pode ser o resultado de *output* de uma técnica de modelação procedimental, responsável por uma outra característica de terreno. Essa comunicação entre as diferentes técnicas (camadas) constitui um fluxo de informação bem definido. Quando integradas num *workflow* (R M Smelik et al. 2008), com a flexibilidade pretendida (ver secção 3.4), implicam a definição de estruturas de dados reconhecidas no âmbito deste trabalho. Estas estruturas de dados podem depois ser utilizadas pelas técnicas de modelação procedimental como entrada e saída de dados.

3.6 **Framework de Modelação Procedimental**

Este documento aborda a construção de uma *framework* para a modelação procedimental de terrenos. No entanto, torna-se necessária a análise da utilização do termo *framework* para descrever a solução proposta, já que este pode assumir diferentes significados de acordo com o tipo de utilização.

No contexto de engenharia de *software*, o termo *framework* descreve, sucintamente, módulos, ambientes de desenvolvimento ou bibliotecas reutilizáveis que facilitam a criação de *software*. Alguns dos problemas descritos na secção 1.2, quanto à modelação procedimental, são o facto de os contributos para este tema abordarem maioritariamente uma característica ou um pequeno conjunto de características de terreno, a falta de controlo no processo de criação e a inexistência de uma solução integradora (Smelik et al. 2009; Smelik et al. 2014). Estes problemas aumentam a complexidade no desenvolvimento de uma técnica de modelação procedimental. A solução proposta pretende solucionar estes problemas, permitindo a um investigador ou programador testar, desenvolver e implementar uma técnica de modelação procedimental, utilizando as ferramentas fornecidas e as interfaces de *software* disponíveis para definir parametrização e integração com outras técnicas. Dessa forma, o foco no desenvolvimento de uma técnica de modelação procedimental mantém-se exclusiva ao desenvolvimento dos algoritmos, podendo a interface para o utilizador, entrada e saída de dados e terreno base de suporte à técnica em desenvolvimento, ser controlado pela solução proposta. Neste contexto podemos considerar que a solução constitui uma *framework* de *software*.

Num outro contexto de utilização, e como abordado no capítulo 1, apesar do potencial do tema da modelação procedimental na criação de terrenos, o grande número existente de técnicas para atender a cada característica de terreno e a dificuldade em parametrizar e combinar essas técnicas constituem uma das maiores resistências à sua adoção por artistas e na sua aplicação em jogos e simuladores (Smelik et al. 2009; Smelik et al. 2014). Ainda existe a “falta de uma *framework* de modelação unificada que combine estas técnicas de uma forma usável” (Smelik et al. 2008, p. 1, tradução nossa). A solução proposta impõe uma organização das técnicas de modelação procedimental, pelas características de terreno que abordam, dando a possibilidade de escolha entre técnicas alternativas. O utilizador pode, desta forma, criar o seu próprio gerador de terrenos ao testar e escolher diferentes combinações de técnicas e parametrizar, usando as ferramentas de edição disponibilizadas e impostas por cada técnica,

de forma a obter o resultado mais adequado à sua utilização. A solução constitui assim, uma *framework* de modelação procedimental de terrenos.

3.7 **Frameworks e Linguagens de Programação a Utilizar**

A *framework* de modelação procedimental será uma aplicação modular constituída por três partes essenciais: o núcleo, as técnicas de modelação procedimental e a interface para o utilizador.

A interface para o utilizador será construída com o Unity3D. O Unity3D é um motor de jogo e *framework* multiplataforma para a criação de jogos, disponibilizando todos os meios necessários para a criação de ferramentas de edição 2D e 3D bem como a pré-visualização de resultados.

O núcleo e as técnicas de modelação procedimental serão desenvolvidos e implementados na linguagem de programação C# com a *framework* .NET. Esta combinação de tecnologias permite facilitar a integração com o Unity3D e manter a coerência entre os módulos. Outro fator importante na escolha desta *framework* e linguagem é o suporte a *reflection*, o que permite a implementação e a separação de código por *plugins* que podem ser ativadas e enumeradas em *runtime*. O suporte a interoperabilidade constitui um outro fator importante, permitindo que as técnicas de modelação procedimental possam usar código desenvolvido noutras linguagens de programação, facilitando a sua integração, e facilitando a utilização de recursos de *hardware*, tal como o uso do GPU (Schneider et al. 2006; Bruneton & Neyret 2008).

3.8 **Análise e Escolha de Técnicas de Modelação Procedimental**

O objetivo deste trabalho pressupõe o estudo de várias técnicas de modelação procedimental quanto à criação de características de um terreno. Dessas técnicas, a que melhor se adequar, deverá ser escolhida para cada característica de forma a integrar, por defeito, a *framework* desenvolvida. Essa escolha é fundamentada pela utilização da técnica estruturada para lidar com decisões complexas multicritério, *analytic hierarchy process* (AHP).

AHP é uma técnica analítica de tratamento de informação que permite estruturar um problema, quantificando os seus elementos e relacionando-os de forma a avaliar soluções alternativas. Nesta técnica deve ser identificado qual o problema, quais os critérios a considerar, quais as alternativas e qual a matriz de comparação de critérios:

1. Qual o problema?
 - Escolha da técnica de modelação procedimental que melhor se adequa na criação de uma característica de terreno.
2. Quais os critérios?
 - Facilidade de integração
 - Facilidade de implementação
 - Qualidade dos resultados
 - Relevância histórica
3. Quais as alternativas?
 - Técnicas e abordagens ao tema da modelação procedimental de terrenos estudados (ver capítulo 2).
4. Qual a matriz de comparação de critérios?
 - Ver a secção 3.8.1

Nas seguintes secções é descrita e aplicada a técnica AHP em cada característica de terreno. A identificação das alternativas e o peso dos critérios são justificados tendo em conta os contributos estudados no capítulo 2.

Na secção de análise e aplicação do método AHP nas alternativas de modelação de mapas de elevação (secção 3.8.2), a técnica AHP é explicada com maior detalhe. Por simplicidade, nas restantes características de terreno é apenas efetuada uma simples análise, é mostrada a comparação de alternativas e é apresentado o resultado.

3.8.1 Critérios e Matriz de Comparação

Dada a impossibilidade da experimentação de todas as técnicas de modelação procedimental, a análise dos critérios é feita com base nos contributos que as abordam, sendo feita uma análise quanto aos seguintes critérios:

- Facilidade de integração - Sendo a criação de uma *framework* integradora o principal objetivo deste trabalho, este fator torna-se importante na escolha de técnicas de modelação procedimental a implementar. Este critério pesa a recetividade da técnica de modelação procedimental a estruturas de dados geradas noutras camadas do *pipeline* de modelação procedimental. Também conta para o peso deste critério a abordagem ou a facilidade de adaptação de ferramentas de parametrização. Como

pode ser observado na matriz de comparação de critérios, é atribuído a este critério a maior importância na escolha das alternativas.

- Facilidade de implementação - Dado o estudo de técnicas de modelação procedimental estar integrado no *design* e desenvolvimento de uma *framework* integradora, o foco deste projeto tem maior incidência na escolha de técnicas que permitam uma maior facilidade de integração e uma maior facilidade na apresentação de resultados. Desta forma, várias técnicas com menor complexidade de implementação podem ser implementadas para permitir demonstrar e testar a *framework* desenvolvida, libertando recursos que podem ser aplicados no desenvolvimento da própria *framework*. Alguns dos contributos estudados, dado a sua complexidade, podem constituir por si só, um projeto independente.
- Qualidade dos resultados – Sendo um critério importante para o utilizador final da *framework* de modelação procedimental, este critério é considerado na escolha das técnicas mais adequadas. No entanto, e atendendo aos objetivos deste projeto, a tentativa de obter um resultado mais realista ou mais detalhado na modelação de uma característica de terreno não deve comprometer significativamente a escolha de uma técnica de modelação procedimental menos complexa e que demonstre um maior potencial de integração e de parametrização.
- Relevância histórica – Algumas das técnicas estudadas são consideradas, são constituídas, ou são fortemente baseadas em técnicas clássicas e recorrentes no tema da modelação procedimental de terrenos. Por esse motivo, torna-se importante para demonstração do potencial da *framework* desenvolvida a implementação e a integração destas técnicas. Este critério é, no entanto, o critério com menor importância na escolha das técnicas de modelação procedimental.

A aplicação deste método na escolha das técnicas de modelação procedimental pretende demonstrar a utilidade de métodos analíticos como auxílio a escolhas complexas. Os critérios e a sua importância têm em conta o âmbito deste projeto e os seus objetivos. Na implementação de uma técnica de modelação procedimental para integrar uma característica de terreno a incluir, por exemplo, na *framework* final ou no âmbito de um outro projeto, os critérios considerados bem como os pesos atribuídos podem assumir diferentes combinações.

Estando identificado o problema e os critérios, na técnica AHP, é necessário criar inicialmente uma matriz de comparação de critérios, tal como representado na Tabela 1. O critério da linha da matriz é comparado com o critério da coluna. Os valores usados representam o nível de importância do critério e assumem valores entre 1 e 9 (sendo 1 igual importância e 9 importância absoluta) quando a sua importância é superior ao critério comparado. Se o critério tiver uma importância inferior ao critério com o qual é comparado, o valor definido é o inverso da comparação contrária. Na comparação do mesmo critério, o valor usado é 1 (igual importância).

Tabela 1 – Matriz de comparação de critérios

	Facilidade de integração	Complexidade de implementação	Qualidade dos resultados	Relevância histórica
Facilidade de integração	1	2	4	5
Facilidade de implementação	1/2	1	3	5
Qualidade dos resultados	1/4	1/3	1	2
Relevância histórica	1/5	1/5	1/2	1

A matriz de comparação de critérios é normalizada na Tabela 2, dividindo cada elemento pela soma da respetiva coluna, de forma a igualar todos os critérios a uma mesma unidade.

Tabela 2 – Matriz de comparação de critérios normalizada

	Facilidade de integração	Complexidade de implementação	Qualidade dos resultados	Relevância histórica
Facilidade de integração	0.51	0.57	0.47	0.38
Facilidade de implementação	0.26	0.28	0.35	0.38
Qualidade dos resultados	0.13	0.09	0.12	0.15
Relevância histórica	0.10	0.06	0.06	0.08

O vetor de prioridades relativas dos critérios, Tabela 3, é posteriormente calculado de forma a identificar a ordem de importância de cada critério. Este vetor é obtido da matriz de comparação normalizada, calculando a média de cada linha.

Tabela 3 – Vetor de prioridades relativas dos critérios

Facilidade de integração	0.48
Facilidade de implementação	0.32
Qualidade dos resultados	0.12
Relevância histórica	0.07

Para avaliar a consistência das prioridades relativas, é calculada a razão de consistência (RC). A razão de consistência é obtida em relação a grandes amostras de juízos aleatórios e permite verificar que a atribuição da importância dos critérios foi racional, ou seja, no método AHP é

pressuposto que caso um critério A seja preferido a B e um critério B seja preferido a C então o critério A é preferido a C. RC é calculado aplicando as seguintes fórmulas (Equações 1 e 2):

$$RC = \frac{IC}{IR} \quad (1)$$

$$IC = \frac{\lambda_{max} - n}{n - 1} \quad (2)$$

O índice de consistência (IC) é obtido do maior valor próprio (λ_{max}) da matriz de comparação de critérios e da ordem da matriz (n). O índice aleatório (IR) é obtido de uma tabela com diferentes valores para matrizes quadradas de ordem n (Tabela 4). Esses índices foram calculados pelo Laboratório Nacional de Oak Ridge, nos EUA, e é referente a um grande número de comparações par a par. Neste caso o valor de IR a utilizar é 0.90.

Tabela 4 – Valores de IR para matrizes quadradas de ordem n

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
IR	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

O valor de RC para o vetor de prioridades relativas representado na Tabela 3 foi calculado em aproximadamente 0.02. Como o valor de RC é inferior a 0.1, podemos concluir que os valores utilizados são consistentes.

3.8.2 Comparação de Alternativas na Modelação de Mapas de Elevação

Do estudo dos contributos de modelação procedimental focados na modelação de mapas de elevação, foram identificadas quatro possíveis alternativas:

- Alternativa A - Algoritmo *diamond square* (Fournier et al. 1982)
- Alternativa B - Algoritmos baseados em geradores de ruído (Musgrave et al. 1989)
- Alternativa C - Algoritmo *triangular edge* com linha de serra e MDI (Belhadj & Audibert 2005)
- Alternativa D - Geração de terrenos através de conceitos baseados na hidrologia (Génevaux et al. 2013)

O algoritmo *diamond square* e os algoritmos baseados em geradores de ruído são soluções relativamente simples e clássicas na modelação procedimental de mapas de elevação. Estas

técnicas estão entre as mais utilizadas na geração de um terreno base para a modelação de outras características de terreno. No entanto, a sua utilização é pouco controlável, gera mapas de elevação excessivamente uniformes e por consequência pouco realistas. Em contraste, a abordagem recente, proposta por Génevaux et al. (2013), apresenta resultados realistas integrando a modelação de rios, contudo, o procedimento é significativamente mais complexo. A solução proposta por Belhadj & Audibert (2005) combina uma técnica clássica de subdivisão, como é o caso do *diamond square*, com a definição de linhas de serra e traçados de rios, dando origem a um mapa de elevação com um realismo satisfatório e apresentando uma solução para a controlabilidade.

Com base nesta análise, para cada critério, é feita a comparação entre as alternativas. De forma semelhante ao cálculo do vetor das prioridades relativas dos critérios, é calculado o vetor de prioridades das alternativas. A Tabela 5, Tabela 6, Tabela 7 e Tabela 8 representam as tabelas paritárias de comparação das alternativas para cada critério e respetivos vetores de prioridades.

Tabela 5 – Matriz de comparação paritária do critério de facilidade de integração na modelação de mapas de elevação

	Alternativa A	Alternativa B	Alternativa C	Alternativa D	Vetor prioridade
Alternativa A	1	3	1/2	1/2	0.20
Alternativa B	1/3	1	1/4	1/4	0.08
Alternativa C	2	4	1	1/2	0.30
Alternativa D	2	4	2	1	0.42

Tabela 6 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de mapas de elevação

	Alternativa A	Alternativa B	Alternativa C	Alternativa D	Vetor prioridade
Alternativa A	1	3	2	3	0.44
Alternativa B	1/3	1	1/3	1	0.12
Alternativa C	1/2	3	1	3	0.31
Alternativa D	1/3	1	1/3	1	0.12

Tabela 7 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de mapas de elevação

	Alternativa A	Alternativa B	Alternativa C	Alternativa D	Vetor prioridade
Alternativa A	1	2	1/5	1/5	0.11
Alternativa B	1/2	1	1/5	1/5	0.07
Alternativa C	5	5	1	1	0.41
Alternativa D	5	5	1	1	0.41

Tabela 8 – Matriz de comparação paritária do critério de relevância histórica na modelação de mapas de elevação

	Alternativa A	Alternativa B	Alternativa C	Alternativa D	Vetor prioridade
Alternativa A	1	1	4	7	0.41
Alternativa B	1	1	4	7	0.41
Alternativa C	1/4	1/4	1	4	0.13
Alternativa D	1/7	1/7	1/4	1	0.05

Usando os vetores de prioridades de alternativas e seguindo a ordem dos critérios descrita na Tabela 1, é formada uma nova matriz, a matriz de prioridades (Tabela 9).

Tabela 9 – Matriz de prioridades na modelação de mapas de elevação

	Facilidade de integração	Facilidade de implementação	Qualidade dos resultados	Relevância histórica
Alternativa A	0.20	0.44	0.11	0.41
Alternativa B	0.08	0.12	0.07	0.41
Alternativa C	0.30	0.31	0.41	0.13
Alternativa D	0.42	0.12	0.41	0.05

A matriz de prioridades é multiplicada pelo vetor de prioridades relativas dos critérios (Tabela 3), sendo obtido o vetor do *ranking* das alternativas, tal como representado na Tabela 10.

Tabela 10 – Vetor do *ranking* das alternativas na modelação de mapas de elevação

Alternativa A	0.28
Alternativa B	0.12
Alternativa C	0.31
Alternativa D	0.30

Deste vetor, pode ser tirada a conclusão que a alternativa C, ou seja, o algoritmo *triangular edge* com linha de serra e MDI, é a melhor alternativa a ser implementada e integrada na *framework* de modelação procedimental.

3.8.3 Comparação de Alternativas na Modelação de Rios, Oceanos e Lagos

- Alternativa A – Modelação de uma rede de rios através de subdivisão recursiva de linhas e posterior geração de mapa de elevação (Kelley et al. 1988)

- Alternativa B – Partículas de rio sujeitas a um método físico (Belhadj & Audibert 2005)
- Alternativa C – Geração de rios por influência de esboços e seguindo modelos baseados na hidrologia (Génevaux et al. 2013).

Todas as alternativas geram rios como meio para a modelação de mapas de elevação. Neste caso, destaca-se a técnica descrita por Génevaux et al. (2013) por ser a mais recente e por apresentar um resultado bastante realista. A abordagem de Belhadj & Audibert (2005), apesar de modelar um terreno menos realista, tem um resultado satisfatório com uma complexidade inferior e é baseada numa técnica clássica. O contributo de Kelley et al. (1988) gera rios por subdivisão recursiva de linhas e considerando o clima e tipo de solo. Esta solução, no entanto, não é facilmente parametrizável e modela rios mais adequados a zonas acidentadas. Nas seguintes tabelas (Tabela 11, Tabela 12, Tabela 13, Tabela 14 e Tabela 15) é efetuada a comparação de critérios destas alternativas:

Tabela 11 – Matriz de comparação paritária do critério de facilidade de integração na modelação de rios, oceanos e lagos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/5	1/5	0.08
Alternativa B	5	1	2	0.61
Alternativa C	5	1/2	1	0.32

Tabela 12 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de rios, oceanos e lagos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/3	5	0.33
Alternativa B	3	1	3	0.55
Alternativa C	1/5	1/3	1	0.12

Tabela 13 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de rios, oceanos e lagos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/5	1/4	0.10
Alternativa B	5	1	2	0.57
Alternativa C	4	1/2	1	0.33

Tabela 14 – Matriz de comparação paritária do critério de relevância histórica na modelação de rios, oceanos e lagos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	3	5	0.65
Alternativa B	1/3	1	2	0.23
Alternativa C	1/5	1/2	1	0.12

Tabela 15 – Matriz de prioridades na modelação de rios, oceanos e lagos

	Facilidade de integração	Facilidade de implementação	Qualidade dos resultados	Relevância histórica
Alternativa A	0.08	0.33	0.10	0.65
Alternativa B	0.61	0.55	0.57	0.23
Alternativa C	0.32	0.12	0.33	0.12

Tabela 16 – Vetor do *ranking* das alternativas na modelação de rios, oceanos e lagos

Alternativa A	0.20
Alternativa B	0.56
Alternativa C	0.24

Deste vetor (Tabela 16), pode ser tirada a conclusão que a alternativa B, ou seja, partículas de rio sujeitas a um método físico, é a melhor alternativa a ser implementada e integrada na *framework* de modelação procedimental.

3.8.4 Comparação de Alternativas na Modelação de Vegetação

- Alternativa A – Modelação de plantas herbáceas usando *L-systems* (Prusinkiewicz et al. 1988)
- Alternativa B – Modelação procedimental de árvores por subdivisão e clonagem de ramos com transformações (Weber & Penn 1995)
- Alternativa C – Criação de árvores com o algoritmo de colonização espacial (Runions et al. 2007)

As técnicas de modelação procedimental baseadas em *L-systems* são simples de implementar e geram resultados que podem replicar de forma satisfatória o formato de plantas reais. No entanto, a utilização de *L-systems* requer a introdução de regras e axiomas que descrevem os modelos de plantas a gerar, o que limita o automatismo da técnica de modelação procedimental. O procedimento apresentado por Weber & Penn (1995) para a modelação de árvores é pouco complexo e tem potencial de parametrização. No entanto, e apesar do

algoritmo de colonização espacial ter um maior nível de complexidade de implementação, permite um elevado nível de parametrização e os modelos resultantes são realistas. Com base na análise das alternativas, foi efetuado nas seguintes tabelas (Tabela 17, Tabela 18, Tabela 19, Tabela 20 e Tabela 21) a comparação dos critérios:

Tabela 17 – Matriz de comparação paritária do critério de facilidade de integração na modelação de vegetação

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/3	1/6	0.10
Alternativa B	3	1	1/3	0.25
Alternativa C	6	3	1	0.65

Tabela 18 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de vegetação

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	3	4	0.62
Alternativa B	1/3	1	2	0.24
Alternativa C	1/4	1/2	1	0.14

Tabela 19 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de vegetação

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/2	1/6	0.10
Alternativa B	2	1	1/5	0.17
Alternativa C	6	5	1	0.72

Tabela 20 – Matriz de comparação paritária do critério de relevância histórica na modelação de vegetação

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	4	4	0.66
Alternativa B	1/4	1	2	0.21
Alternativa C	1/4	1/2	1	0.13

Tabela 21 – Matriz de prioridades na modelação de vegetação

	Facilidade de integração	Facilidade de implementação	Qualidade dos resultados	Relevância histórica
Alternativa A	0.10	0.62	0.10	0.66
Alternativa B	0.25	0.24	0.17	0.21
Alternativa C	0.65	0.14	0.72	0.13

Tabela 22 – Vetor do *ranking* das alternativas na modelação de vegetação

Alternativa A	0.31
Alternativa B	0.23
Alternativa C	0.46

Deste vetor (Tabela 22), pode ser tirada a conclusão que a alternativa C, ou seja, criação de árvores com o algoritmo de colonização espacial, é a melhor alternativa a ser implementada e integrada na *framework* de modelação procedimental.

3.8.5 Comparação de Alternativas na Modelação de Redes de Estradas

- Alternativa A – Modelação procedimental de estradas por adaptação de curvas ao mapa de elevação e edição de um esboço de mapa (McCrae & Singh 2009)
- Alternativa B - Modelação de estradas recorrendo a funções de custo e algoritmo A* (Galin et al. 2010)
- Alternativa C - Geração de traçados rodoviários seguindo um modelo hierárquico em níveis e definição de uma rede topológica de estradas (Campos et al. 2014; Campos, Leitão, Pereira, et al. 2015; Campos, Leitão & Coelho 2015)

A abordagem de McCrae & Singh (2009) apresenta meios para a parametrização da modelação de traçados rodoviários, no entanto, e apesar do método introduzido para a definição de curvas, o resultado não é o mais coerente numa perspetiva de realismo. Já nos contributos que apresentam a alternativa C, além de abordagem a métodos que permitem a controlabilidade, os modelos de traçados rodoviários gerados são organizados por tipologia, avaliam funções de custo e seguem conceitos de engenharia civil que contribuem para o realismo dos resultados. O procedimento de modelação de estradas recorrendo a funções de custo e algoritmo A*, recorre à utilização de um algoritmo clássico na obtenção otimizada de percursos, com resultados satisfatórios. Da análise efetuada às alternativas consideradas quanto à modelação de redes de estradas, é apresentada nas seguintes tabelas (Tabela 23, Tabela 24, Tabela 25, Tabela 26 e Tabela 27) a comparação dos critérios:

Tabela 23 – Matriz de comparação paritária do critério de facilidade de integração na modelação de redes de estradas

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	2	1/4	0.20
Alternativa B	1/2	1	1/5	0.12
Alternativa C	4	5	1	0.68

Tabela 24 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de redes de estradas

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/2	2	0.31
Alternativa B	2	1	2	0.49
Alternativa C	1/2	1/2	1	0.20

Tabela 25 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de redes de estradas

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/3	1/4	0.12
Alternativa B	3	1	1/3	0.27
Alternativa C	4	3	1	0.61

Tabela 26 – Matriz de comparação paritária do critério de relevância histórica na modelação de redes de estradas

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/3	1/3	0.14
Alternativa B	3	1	1	0.43
Alternativa C	3	1	1	0.43

Tabela 27 – Matriz de prioridades na modelação de redes de estradas

	Facilidade de integração	Facilidade de implementação	Qualidade dos resultados	Relevância histórica
Alternativa A	0.20	0.31	0.12	0.14
Alternativa B	0.12	0.49	0.27	0.43
Alternativa C	0.68	0.20	0.61	0.43

Tabela 28 – Vetor do *ranking* das alternativas na modelação de redes de estradas

Alternativa A	0.22
Alternativa B	0.28
Alternativa C	0.50

Deste vetor (Tabela 28), pode ser tirada a conclusão que a alternativa C, ou seja, geração de traçados rodoviários seguindo um modelo hierárquico em níveis e definição de uma rede topológica de estradas, é a melhor alternativa a ser implementada e integrada na *framework* de modelação procedimental.

3.8.6 Comparação de Alternativas na Modelação de Centros Urbanos

- Alternativa A – Modelação através da criação e ruas paralelas (Greuter et al. 2003)
- Alternativa B – Modelação com recurso a *L-systems* (Parish & Müller 2001; Coelho et al. 2007)
- Alternativa C – Modelação utilizando campos tensoriais (Chen et al. 2008)

A solução de modelação através da criação de ruas paralelas gera o resultado menos realista e mais limitado, contudo é a técnica mais simples para a modelação procedimental de centros urbanos. As alternativas que recorrem a *L-systems* e à utilização de campos tensoriais apresentam um elevado nível de parametrização e de realismo de resultados. São apresentadas nas tabelas (Tabela 29, Tabela 30, Tabela 31, Tabela 32 e Tabela 33) seguintes a comparação dos critérios entre as alternativas:

Tabela 29 – Matriz de comparação paritária do critério de facilidade de integração na modelação de centros urbanos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	2	1/3	0.25
Alternativa B	1/2	1	1/3	0.16
Alternativa C	3	3	1	0.59

Tabela 30 – Matriz de comparação paritária do critério de facilidade da implementação na modelação de centros urbanos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	2	3	0.54
Alternativa B	1/2	1	2	0.30
Alternativa C	1/3	1/2	1	0.16

Tabela 31 – Matriz de comparação paritária do critério de qualidade dos resultados na modelação de centros urbanos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/2	1/3	0.16
Alternativa B	2	1	1/2	0.30
Alternativa C	3	2	1	0.54

Tabela 32 – Matriz de comparação paritária do critério de relevância histórica na modelação de centros urbanos

	Alternativa A	Alternativa B	Alternativa C	Vetor prioridade
Alternativa A	1	1/2	2	0.30
Alternativa B	2	1	3	0.54
Alternativa C	1/2	1/3	1	0.16

Tabela 33 – Matriz de prioridades na modelação de centros urbanos

	Facilidade de integração	Facilidade de implementação	Qualidade dos resultados	Relevância histórica
Alternativa A	0.25	0.54	0.16	0.30
Alternativa B	0.16	0.30	0.30	0.54
Alternativa C	0.59	0.16	0.54	0.16

Tabela 34 – Vetor do *ranking* das alternativas na modelação de centros urbanos

Alternativa A	0.34
Alternativa B	0.25
Alternativa C	0.42

Deste vetor (Tabela 34), pode ser tirada a conclusão que a alternativa C, ou seja, modelação utilizando campos tensoriais, é a melhor alternativa a ser implementada e integrada na *framework* de modelação procedimental.

3.9 Resumo do Capítulo

Foi efetuada, neste capítulo, uma análise dos contributos referidos no estado da arte. Desses contributos, foram identificadas soluções relevantes que resolvem parcialmente os objetivos quanto à construção de uma *framework* integradora e quanto à controlabilidade do processo de modelação.

Foram estabelecidos os pressupostos e as limitações que definem o âmbito do tema deste trabalho e da solução a desenvolver. As linguagens de programação e *frameworks* de desenvolvimento mais adequadas para atender à solução pretendida foram identificadas.

Por fim, para cada característica de terreno, foram levantadas abordagens alternativas do estado da arte e analisadas quanto a um conjunto de critérios. Com base nessa análise, foi aplicado um método analítico de apoio a decisões complexas, o AHP, para selecionar as alternativas mais adequadas para a integração na *framework* a desenvolver.

4 Análise de Valor

4.1 Contexto

Antes de abordar a análise de valor, é necessário fornecer o contexto e o tipo de modelo de negócio no qual a solução proposta se insere bem como os seus intervenientes.

4.1.1 Descrição da Solução

A solução proposta no âmbito deste projeto é *Terraframe* (Figura 4), uma *framework* de modelação procedimental, que permite resolver a carência de soluções para a criação de modelos 3D de terrenos completos de forma automática e inclui ferramentas simples de parametrização. Pretende ser uma alternativa aos meios tradicionais de criação de modelos 3D em editores, que requerem um esforço, trabalho, tempo e custo elevados. Pretende ser, também, uma solução que facilite o desenvolvimento de novos contributos na área da modelação procedimental, ao fornecer um ambiente com interface para o utilizador e ferramentas de edição já implementadas, permitindo que um investigador ou programador de *software* se foque apenas na criação e teste de algoritmos.



Figura 4 – Logótipo da framework a desenvolver

Dessa forma, a solução proposta é uma *framework* de modelação procedimental que gera modelos 3D de terrenos. Essa *framework* permite a gestão de projetos de terreno, sendo o processo de modelação organizado num *workflow* constituído por camadas que representam características de terreno. Técnicas de modelação procedimental podem ser adicionadas ou escolhidas, através de *plugins*, e integrar o *workflow* de modelação na camada correspondente à característica de terreno que modela.

4.1.2 Clientes

Dois tipos de clientes são identificados na utilização da *framework* de modelação procedimental: artistas ou desenvolvedores de modelos 3D e investigadores ou programadores de *software*.

Artistas e desenvolvedores de modelos 3D são os principais utilizadores da *framework* de modelação procedimental. Estes utilizadores tem como objetivo a obtenção de modelos 3D de terrenos para serem aplicados em jogos e simuladores. Este utilizador tem como principais vantagens na utilização da *framework* proposta a rapidez e facilidade na modelação de terrenos 3D quando comparado com os meios tradicionais, dado que o processo de modelação é essencialmente automático e intuitivamente parametrizado.

Os investigadores e programadores de *software* utilizam a solução proposta como uma plataforma que lhes permite testar e criar algoritmos de modelação procedimental sem a necessidade de desenvolver a interface para o utilizador ou de integrar dados de entrada, sendo esse processo regulado pelo núcleo da *framework*. Desta forma, o utilizador será mais focado na implementação da técnica que desenvolve.

4.1.3 Intervenientes

Além dos clientes, mais dois intervenientes são identificados neste modelo de negócio:

- A comunidade da *framework* de modelação procedimental de terrenos
 - Esta comunidade tem como objetivo representar a organização que explora a *framework* proposta, fornece os *plugins* de modelação procedimental, dá suporte, mantém o *website*, disponibiliza documentação, oferece formações e organiza seminários.
- Universidades e Núcleos de Investigação
 - O tema da modelação procedimental tem-se revelado muito promissor, existindo muitos contributos nessa área. Tendo em conta que a *framework* proposta apresenta soluções para a integração e disponibiliza ferramentas de parametrização facilitando a implementação de novas técnicas de modelação procedimental de terrenos, a sua adoção como *sandbox* em projetos de investigação torna-se viável. Além disso, a *framework* impõe uma organização das técnicas de modelação procedimental, obrigando a sua implementação, em *plugins*, a fornecer metadados e indicar quais as características de terreno que modelam. Estas imposições dão origem a que contributos futuros gerem novos *plugins* de modelação procedimental prontos a integrar na *framework*.

4.1.4 Principais Interações

Nesta secção são descritas as principais interações entre os intervenientes neste modelo de negócio:

- Disponibilização de documentos
 - Estes documentos podem ser manuais de utilizador ou de integração. Podem estar disponíveis no *website* informativo da *framework* proposta.
- Formações
 - Formações podem ser feitas, quer no contexto de utilização por artistas e desenvolvedores de modelos 3D quer no contexto de investigadores e programadores de *software*.
- Organização de seminários
 - Seminários com o tema da modelação procedimental, onde a *framework* proposta pode ser apresentada e demonstrada.
- Troca de *plugins* de técnicas de modelação procedimental de terrenos
 - *Plugins* de modelação procedimental podem ser disponibilizadas no *website* da *framework*. Os clientes que necessitem de integrar uma determinada característica de terreno podem consultar quais os *plugins* disponíveis e transferir os que melhor se adequam.

4.1.5 Lucro

O lucro extraído do modelo de negócio proposto é obtido essencialmente de duas formas:

- Uma subscrição para a utilização da *framework* por parte dos artistas e desenvolvedores de modelos 3D. Uma licença gratuita é disponibilizada para Universidades e Núcleos de investigação.
- Compra de *plugins* de modelação procedimental. De forma a incentivar a criação e desenvolvimento de técnicas de modelação procedimental, é oferecida a possibilidade ao programador de atribuir um preço ao *plugin* de modelação procedimental.

4.2 Processo de Inovação

O tema deste trabalho pressupõe a criação de uma *framework* de modelação procedimental de terrenos. Esta *framework*, sendo um produto inovador na área da computação gráfica, é enquadrado e analisado quanto ao processo de inovação.

“O processo de inovação pode ser dividido em três áreas: o *fuzzi front end* (FFE), o processo *new product development* (NPD) e a comercialização [...]” (Belliveau et al. 2004, p. 5, tradução nossa).

Neste capítulo é abordado apenas o FFE, já que esta é a fase inicial do processo de inovação e é “definida pelas atividades que antecedem o formal e bem estruturado [...]” (Koen et al. 2001, p. 1, tradução nossa) NPD.

O modelo no qual se baseia esta análise é o *new concept development* (NCD) (Koen et al. 2001). Este modelo define cinco elementos chave: identificação de oportunidade, análise de oportunidade, geração de ideias e enriquecimento, seleção de ideias e definição de conceito. Os elementos chave para este projeto foram identificados tendo apenas como base o tema do projeto, os estudos efetuados, a análise e a solução proposta. Métodos efetivos, ferramentas e técnicas poderão ser utilizados para desenvolver melhor cada um dos elementos chave (Belliveau et al. 2004) e serão referidos.

4.2.1 Identificação de Oportunidade

Na análise do tema da dissertação foi identificada uma dificuldade na adoção de técnicas de modelação procedimental de terrenos por artistas e na sua aplicação em simuladores e videojogos, apesar do potencial comprovado (secção 1.2). Foi também identificada a carência de ferramentas que permitam integrar a variedade de técnicas existente de forma a criar um terreno completo e a dificuldade em controlar o seu funcionamento. As dificuldades identificadas constituem problemas que podem ser resolvidos ou melhorados e revelam uma lacuna tecnológica, seguindo, desta forma, a definição de oportunidade (Belliveau et al. 2004).

Os principais métodos para a análise deste elemento chave são focados na avaliação de fatores incertos e de previsão: criação de *roadmaps*, análise e previsão de tendências tecnológicas, análise de conhecimento competitivo, análise de tendências dos clientes, estudos de mercado e planificação de cenário.

4.2.2 Análise de Oportunidade

Um estudo sobre contributos na área da modelação procedimental foi efetuado (capítulo 2). Soluções propostas ou construídas foram encontradas de forma a resolver o problema da controlabilidade e integração, no entanto, essas soluções são escassas ou têm lacunas, deixando margem para melhoria ou criação de alternativas.

A utilização de técnicas de modelação procedimental revela ter potencial ao permitir reduzir o tempo e o custo da criação de conteúdos 3D quando comparado com a utilização de ferramentas de modelação tradicionais.

Os mesmos métodos, técnicas e ferramentas usados na identificação de oportunidade podem ser usados neste elemento chave do NCD, contudo, com maior detalhe. Em projetos de maior dimensão, pode ser criada uma pequena equipa multidisciplinar para trabalhar a tempo inteiro na análise de oportunidade.

4.2.3 Geração de Ideias e Enriquecimento

Como resultado da análise das contribuições neste tema, dos objetivos e dos pressupostos (secção 3.1), várias ideias a utilizar na solução final foram identificadas:

- Criação de uma *framework* que permita a gestão e modelação de projetos de terrenos gerados de forma procedimental.
- Organização das técnicas de modelação procedimental por características de terreno que abordam.
- Definição de um *workflow* de modelação com uma ordem específica de características de terreno.
- Implementação de técnicas de modelação procedimental específicas para cada característica de terreno.
- Implementação de técnicas de modelação em *plugins* de forma a poderem ser adicionadas novas técnicas e permitir escolha.
- Definição de um *workflow* com uma ordem de características de terreno configurável.
- Implementação de ferramentas de parametrização por técnica de modelação procedimental.
- Implementação de um conjunto de ferramentas de parametrização genérico que pode ser utilizado pelas técnicas de modelação procedimental.

Entre os métodos, técnicas e ferramentas para avaliar este elemento chave, destacam-se: métodos para identificar as necessidades dos clientes, descobrir o arquétipo do cliente, intercalar continuamente as necessidades e dificuldades do mercado e tipo de negócio com os avanços tecnológicos, identificar novas soluções tecnológicas, impor uma cultura organizacional que encoraje os colaboradores a testar e validar as suas próprias ideias ou as ideias de outros, aplicação de diversos incentivos de forma a estimular a criação de ideias,

criação de um banco de ideias *online* e inclusão de pessoas com diferentes estilos cognitivos na equipa.

4.2.4 Seleção de Ideias

Das ideias identificadas, foram selecionadas as que melhor se encaixavam na solução pretendida, tendo em conta a diversidade de técnicas de modelação procedimental existentes, e foram selecionadas as que impunham menores limitações. Desta forma foram selecionadas as seguintes ideias:

- Criação de uma *framework* que permita a gestão e modelação de projetos de terrenos gerados de forma procedimental.
- Organização das técnicas de modelação procedimental por características de terreno que abordam.
- Implementação de técnicas de modelação em *plugin* de forma a poderem ser adicionadas novas técnicas e permitir escolha.
- Definição de um *workflow* com uma ordem de características de terreno configurável.
- Implementação de um conjunto de ferramentas de parametrização genérico que pode ser utilizado pelas técnicas de modelação procedimental.

Os métodos, técnicas e ferramentas para auxiliar este elemento chave do NCD podem ser: metodologias de portfólio baseadas em múltiplos fatores, processo de seleção formal de ideias com o pedido de *feedback* dos colaboradores de ideias e usar a teoria de opções de forma a avaliar projetos.

4.2.5 Definição de Conceito

A criação de uma *framework* de modelação procedimental. A *framework* organiza as técnicas de modelação procedimental pelas características de terreno constituindo um *workflow* de modelação configurável. A *framework* permite a escolha de técnicas alternativas para cada característica. Ferramentas de parametrização e visualização são disponibilizadas para que as técnicas possam definir o tipo de interação com o utilizador. Da combinação de diferentes técnicas de modelação procedimental é gerado um terreno completo e contendo as características geradas pelas respetivas técnicas.

Alguns métodos, técnicas e ferramentas para auxiliar esta análise são: abordagens de deliberação de objetivos, avaliação rápida de inovações com maior potencial, uso rigoroso de um processo de *technology stage-gate* (TSG) em projetos de elevado risco, envolvimento de clientes em testes de produto e perseguir abordagens científicas alternativas.

4.3 Valor Para o Cliente

O valor para o cliente é a percepção de vantagem que o cliente pode ter em associação com o que o produto oferece. O valor para o cliente pode “[...] ocorrer como a redução de sacrifícios, presença de benefícios [,] o resultado da combinação pesada de sacrifícios e benefícios ou a agregação, ao longo do tempo, destes três fatores [...]” (Woodall 2003, p. 2, tradução nossa).

Tendo como base esta definição, os benefícios e sacrifícios são identificados e explicados para cada cliente alvo de forma a permitir avaliar o valor para o cliente (Tabela 35 e Tabela 36). Nestes dois casos, a existência de um maior número de benefícios em relação a sacrifícios constitui o valor para o cliente da solução proposta.

4.3.1 Benefícios e Sacrifícios para Artistas e Desenvolvedores de Modelos 3D

Tabela 35 – Benefícios e sacrifícios da solução para artistas e desenvolvedores de modelos 3D

Benefícios	Sacrifícios
Facilidade de utilização	Distribuição
Inovação	Instalação
Customização	
Preço	
Custo	
Rapidez	

- Benefícios
 - Facilidade de utilização
 - A aplicação apresenta uma interface intuitiva que disponibiliza ferramentas de fácil utilização para parametrizar a modelação de terrenos. É imposta a organização do processo de modelação num *workflow*.
 - Inovação
 - A integração de técnicas de modelação procedimental e a sua parametrização é ainda um tema pouco explorado. A *framework* proposta segue uma arquitetura inovadora neste tema, na medida em que permite ligar facilmente técnicas de modelação alternativas e organiza-las num *workflow* de modelação.
 - Customização
 - Técnicas de modelação procedimental são organizadas e implementadas em *plugins*, o que permite a ligação e escolha de novas técnicas. O *workflow* de modelação é customizável de forma atender às necessidades das técnicas de modelação procedimental.

- Preço
 - Ferramentas tradicionais de modelação apresentam custos de compra e licenciamento em contrapartida à solução proposta.
- Custo
 - O custo no desenvolvimento de modelos 3D é reduzido, já que o processo de modelação procedimental é essencialmente automático, reduzindo gastos em recursos humanos.
- Rapidez
 - O processo é essencialmente automático, necessitando apenas da parametrização do terreno pretendido. Desta forma, o processo de modelação é rápido em alternativa ao processo tradicional que é moroso.
- Sacrifícios
 - Distribuição
 - A existência de um grande número de técnicas de modelação procedimental pode causar confusão no utilizador e dificultar a sua distribuição.
 - Instalação
 - Para um tipo de utilizador com menores conhecimentos informáticos, a instalação de *plugins* de modelação procedimental pode constituir uma dificuldade.

4.3.2 Benefícios e Sacrifícios para Investigadores e Programadores de *Software*

Tabela 36 – Benefícios e sacrifícios da solução para investigadores e programadores de *Software*

Benefícios	Sacrifícios
Facilidade de utilização	Performance
Facilidade de integração	
Inovação	
Rapidez	

- Benefícios
 - Facilidade de utilização
 - A *framework* define interfaces de *software* bem definidas e documentação bem estruturada. A interface para o utilizador é intuitiva e disponibiliza ferramentas de fácil utilização para parametrizar o processo de modelação.
 - Facilidade de integração
 - As técnicas de modelação são implementada como *plugins* que são integrados no *workflow* de modelação. A interface de *software* é bem documentada e permite definir as características de terreno que a técnica modela e as ferramentas de parametrização a utilizar.

- Inovação
 - A integração de técnicas de modelação procedimental e a sua parametrização é ainda um tema pouco explorado. A *framework* proposta segue uma arquitetura inovadora neste tema, na medida em que permite ligar facilmente técnicas de modelação alternativas e organiza-las num *workflow* de modelação.
- Rapidez
 - A implementação de uma técnica de modelação é mais focada nos algoritmos procedimentais. A interface para o utilizador, fluxo de dados e gestão do processo de modelação são tratados no núcleo da *framework*. Desta forma o processo de teste e desenvolvimento de técnicas de modelação procedimental torna-se mais rápido.
- Sacrifícios
 - Performance
 - A utilização e integração da técnica de modelação procedimental na *framework* pode reduzir ligeiramente a performance comparativamente com a implementação isolada.

4.4 Modelo de Negócio de *Canvas*

Para auxiliar a descrição do modelo de negócio no qual a *framework* é inserida (secção 4.1), é apresentado na Figura 5 o modelo de negócio de *canvas* (Osterwalder & Pigneur 2010).



Figura 5 – Modelo de negócio de *canvas*

Este modelo permite efetuar uma definição formal de um modelo de negócio através da identificação de nove elementos chave: segmentos de cliente, proposta de valor, canais, relacionamento com clientes, fontes de receita, recursos-chave, atividades-chave, parcerias-chave e estrutura de custos.

4.5 Rede de valor

A análise de valor de negócio pode recorrer à utilização de modelos para analisar a criação de valor. Dois desses modelos são a cadeia de valor de Porter (Porter 1985) e a rede de valor de Verna Allee (Allee 2002a). No entanto, a cadeia de valor é um “modelo linear e mecanizado de negócio que é baseado na linha de produção da era industrial” (Allee 2002b, p. 1, tradução nossa). Desta forma, a rede de valor apresenta-se como um modelo mais indicado para analisar a criação de valor do modelo de negócio apresentado, já que o seu funcionamento é bastante centrado na troca de valor entre diferentes intervenientes.

A rede de valor é um modelo representado por uma “[...] teia de relações que geram valor económico e outros benefícios através de trocas dinâmicas complexas entre dois ou mais indivíduos, grupos ou organizações” (Allee 2002, p.1, tradução nossa). As trocas de valor podem ser de dois tipos: trocas de valor tangíveis e trocas de valor intangíveis.

As trocas de valor tangíveis dizem respeito a trocas de bens, serviços, lucros, transações envolvendo contratos e faturas, recibos de encomendas, pedidos de propostas, confirmações ou pagamentos, relatórios, fluxo de bens, fluxo de serviços e fluxo de lucro.

As trocas intangíveis podem ser separadas em conhecimento e benefícios. Como trocas de conhecimento intangíveis entende-se informação estratégica, planeamento de conhecimento, conhecimento de processo, conhecimento tecnológico, *design* colaborativo e desenvolvimento de políticas. As trocas intangíveis de benefícios podem ser vantagens ou favores oferecidos por uma pessoa, grupo ou organização.

O objetivo da rede de valor é gerar benefícios para todos os seus participantes. “Numa rede de valor bem-sucedida todos os atores ou participantes contribuem e recebem valor de formas que sustentam o seu sucesso e o sucesso da rede de valor como um todo” (Allee 2002b, p. 2, tradução nossa). Dessa forma, a construção de uma rede de valor permite reconhecer falhas e otimizar a criação de valor.

A Figura 6 mostra a rede de valor do modelo de negócio apresentado. Salientam-se as trocas intangíveis de conhecimento, resultantes da investigação de técnicas de modelação procedimental pelos investigadores, a avaliação resultante do feedback sobre as técnicas de modelação procedimental da parte dos artistas e desenvolvedores de modelos 3D e do reconhecimento dado à *framework* de modelação procedimental.

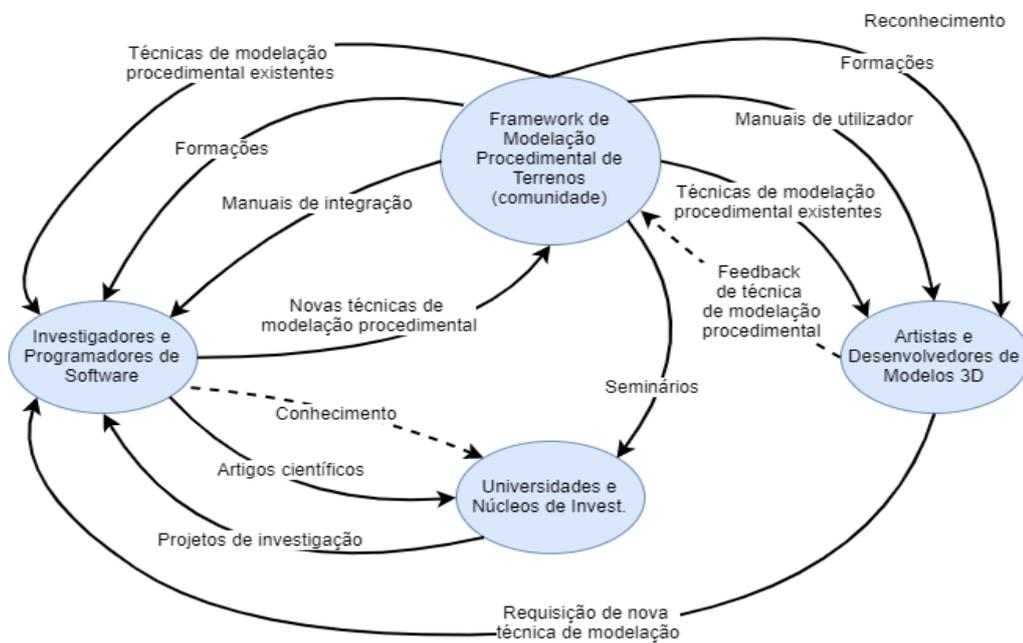


Figura 6 – Rede de valor do modelo de negócio

4.6 Proposta de Valor

A proposta de valor é uma mensagem simples, clara, curta e eficaz e deve transmitir a informação necessária ao público-alvo. A proposta de valor é um resumo geral do negócio onde são identificados os produtos e serviços prestados, qual o respetivo valor e quem são os consumidores. Seguindo esta definição é enunciada a proposta de valor da *framework* de modelação de terrenos:

Terraframe é uma framework de modelação procedimental que permite a artistas e desenvolvedores gerar modelos 3D de terrenos de forma rápida, simples e eficaz, para integrar nos seus projetos de software. Oferece ferramentas que facilitam a investigação e o desenvolvimento de novas técnicas de modelação.

As quatro questões que uma proposta de valor deve responder são enumeradas e explicadas com base na proposta de valor enunciada:

- Quais são os produtos?
 - A *framework* de modelação procedimental.
- Quem é o cliente alvo?
 - Artistas, programadores e investigadores.

- Qual o valor que o negócio disponibiliza?
 - Gera modelos 3D de terrenos para integrar nos projetos de *software* e disponibiliza ferramentas que facilitam a investigação e o desenvolvimento de técnicas de modelação.
- Por que razão o produto é único?
 - Permite gerar modelos 3D de terrenos de forma rápida, simples e eficaz e disponibiliza ferramentas que facilitam a investigação.

4.7 Resumo do Capítulo

Neste capítulo foi contextualizada a *framework* de modelação procedimental desenvolvida num modelo de negócio. Desta forma, foi possível efetuar uma análise de valor e explorar o possível potencial que a solução desenvolvida introduz.

Primeiramente, a *framework* desenvolvida foi enquadrada no processo de inovação, identificando os cinco elementos chave do modelo NCD. O valor para o cliente foi identificado, tabelando e explicando os benefícios e sacrifícios para cada cliente alvo da *framework* desenvolvida. A criação de valor foi analisada através da representação do modelo de negócio em *canvas* e do modelo de rede de valor de Verna Allee. Por fim, respondendo às quatro questões essenciais, foi construída a proposta de valor da *framework* desenvolvida.

5 *Design* da Solução

Neste capítulo é apresentada a solução que foi concebida atendendo aos objetivos propostos para o projeto e tendo como base a análise do problema apresentada no capítulo 3.

5.1 Solução Modular

A grande variedade de soluções existentes e os sucessivos contributos que vão surgindo faz com que seja de enorme importância que a solução integradora permita a fácil ligação a novas técnicas de modelação procedimental. Além disso, a escolha entre técnicas alternativas para a construção de uma determinada característica de terreno tem muito potencial ao permitir a construção de terrenos com características diferentes e permitindo que o resultado seja o mais adequado à utilização pretendida. Desta forma, a *framework* pode servir de *sandbox* para o teste de novas contribuições no tema da modelação procedimental de terrenos.

Por este motivo, uma solução com recurso a *plugins* é proposta. Os métodos de modelação procedimental são implementados em *plugins* que são detetados pelo núcleo da *framework* e que podem ser ativados ou desativados pelo utilizador. Esta abordagem permite uma separação bem definida entre as técnicas de modelação procedimental e as restantes partes da *framework*. Os *plugins* obedecem a uma interface que especifica a entrada e saída de informação e inclui metadados que o núcleo da *framework* utiliza para disponibilizar informação (nome, descrição, data e autores) ao utilizador e identificar as partes funcionais da criação de terrenos aos quais se destina.

É também proposta uma separação da interface para o utilizador de forma a permitir a fácil integração de outros tipos de interface com o núcleo da *framework*. Permitindo que, por exemplo, uma interface *web* possa ser ligada com a *framework* de forma a gerar uma aplicação *web* em *cloud computing*.

5.2 Arquitetura da Aplicação

A arquitetura da *framework* proposta é constituída por três partes essenciais (Figura 7): *plugins* de modelação procedimental, núcleo e interface para o utilizador.

5.2.1 *Plugins* de Modelação Procedimental

Os *plugins* de modelação procedimental são os módulos responsáveis pela implementação das técnicas que permitem criar determinada característica ou conjunto de características de terreno. Os *plugins* são implementados de forma a facilitar a sua integração, podendo ser

adicionados e ativados sem necessidade de compilar a *framework*. Versões alternativas para a criação de características de terreno podem existir e ser escolhidas na interface para o utilizador. Interfaces de aplicação são definidas e devem ser implementadas pelo *plugin*, de forma a compatibilizar a sua integração.

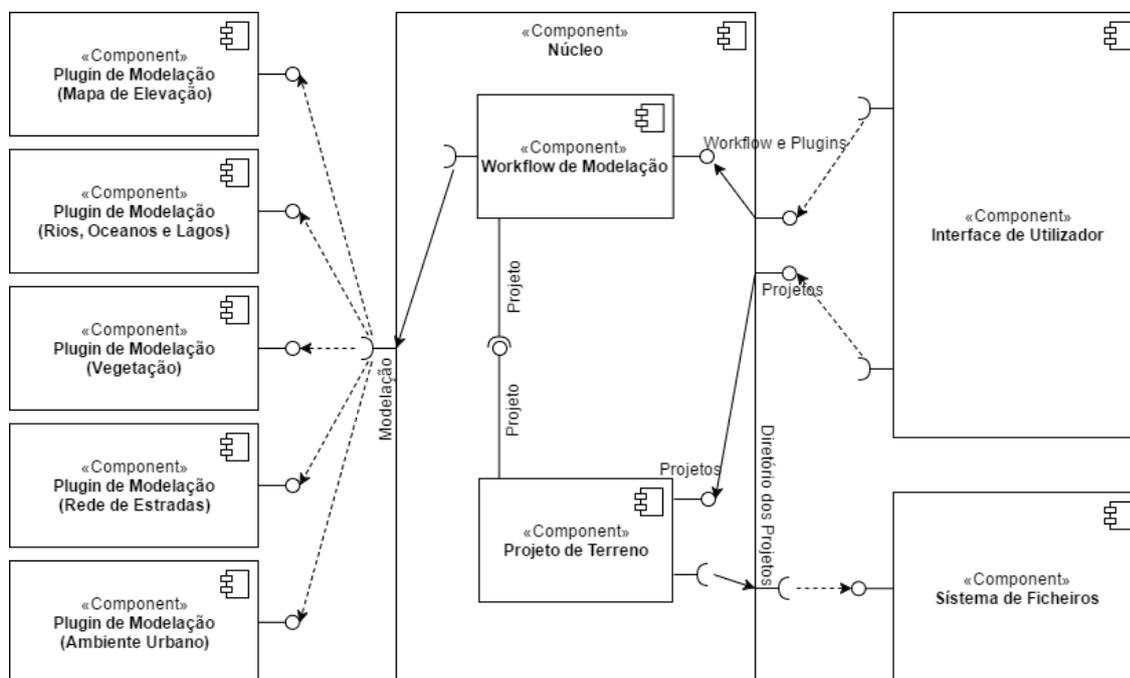


Figura 7 – Diagrama de componentes da solução proposta

5.2.2 Núcleo da *Framework*

O núcleo da *framework* é o módulo central onde o processo de criação do terreno é controlado. O funcionamento baseia-se num *workflow* de criação (R M Smelik et al. 2008) constituído por várias camadas responsáveis pelas características de terreno. Essas características são: mapa de elevação, rios, oceanos e lagos, vegetação, redes de estradas e ambiente urbano. As técnicas de modelação procedimental de terrenos são implementadas de forma a constituir uma ou mais camadas no *workflow* de criação num *plugin*. Dessa forma, os *plugins* de modelação procedimental são invocados pelas respetivas camadas do *workflow* de criação para gerar as características do terreno.

A especificação das interfaces de *software* que permitem a ligação dos *plugins* e que permitem a ligação com a interface para o utilizador é efetuada neste componente. Além disso, são definidas classes base para as técnicas de modelação procedimental indicarem ferramentas de visualização e edição de parâmetros e especifica estruturas de dados que integram o fluxo de informação na execução do *pipeline*.

A gestão de projetos de terreno e de *plugins* de modelação é efetuada neste componente. As interfaces de *software* para a ligação à interface para o utilizador disponibilizam meios para a sua manipulação.

5.2.3 Interface para o Utilizador

A interface para o utilizador disponibiliza ferramentas para permitir o desenho e a parametrização do processo de criação de modelos. Ferramentas, tais como, máscaras, esboços e parâmetros pré definidos, são implementadas e a informação recolhida é utilizada pelo núcleo e *plugins* de modelação procedimental. A interface implementa ferramentas de pré-visualização de resultados. As ferramentas de edição e pré-visualização que devem ser mostradas são especificadas pelo núcleo da *framework* e são listadas pelos *plugins* de modelação procedimental. A interface para o utilizador disponibiliza configurações de forma a permitir gerir os *plugins* de modelação procedimental atualmente disponíveis.

5.3 Controlabilidade e Interface para o Utilizador

A interface para o utilizador baseia-se em contributos de soluções de integração existentes e baseia-se em ideias para a melhoria da controlabilidade das técnicas de modelação procedimental. É uma solução flexível, que se ajusta às necessidades das técnicas de modelação procedimental ligadas ao núcleo da *framework*.

A interface segue o *workflow* definido para o processo de modelação de um terreno. O utilizador consegue parametrizar, utilizando as diferentes ferramentas disponibilizadas, cada camada responsável por uma característica do terreno. Esse processo pode ser efetuado na íntegra, antes de iniciar o processo de criação completo, ou permitir a paragem a meio do processo para que ajustes possam ser efetuados para as camadas seguintes do *workflow*.

A qualquer momento, o utilizador pode parar o processo de criação em curso ou então partir de um terreno gerado para editar a parametrização de uma determinada camada do *workflow*. Essa alteração de parâmetros já processados implica o reinício do processo de criação, partindo da camada editada, para que o terreno se ajuste às alterações impostas. Ferramentas de pré-visualização do terreno auxiliam o processo corretivo. Desta forma, o utilizador pode criar um terreno, visualizá-lo e ajustar as características iterativamente até obter o resultado pretendido.

As ferramentas de edição disponibilizadas são definidas pelo núcleo da *framework*. O módulo da interface para o utilizador recebe uma lista, para cada camada do processo de criação, das ferramentas a mostrar e ajusta a interface convenientemente.

As ferramentas disponibilizadas podem ser parâmetros, escolhas, máscaras e esboços.

As ferramentas mais simples, a edição de parâmetros e escolhas, permitem a definição de constantes que influenciam a técnica de modelação procedimental. *Widgets* utilizados podem ser caixas de texto, *checkboxes*, *radio buttons*, *combo box* ou *sliders*.

Máscaras e esboços dizem respeito à edição de *bitmaps*. A edição tem semelhanças a aplicações de edição de imagem e permite a importação de imagens externas.

As ferramentas de pré-visualização são visualizadores 3D, que mostram o estado ou atributos do terreno, visualizadores de *bitmaps*, listagem de valores e gráficos.

As diferentes ferramentas são acompanhadas de metadados que influenciam a forma como a ferramenta é apresentada. Podem, por exemplo, indicar o título associado ao parâmetro, as opções (no caso de uma escolha), a resolução (no caso de uma máscara), entre outros.

5.4 Interface com as Técnicas de Modelação

As técnicas de modelação procedimental são implementadas de forma isolada constituindo um *plugin*. Desta forma, a troca de técnicas de modelação procedimental pode ser efetuada sem ser necessário compilar toda a *framework*. No entanto, esta abordagem implica a definição de uma interface que disponibiliza todos os meios para a comunicação entre o núcleo e o *plugin*.

Um dos aspetos importantes é a passagem de metadados que identifiquem o *plugin* e que o enquadrem no *workflow* de criação de terrenos. Desta forma, o núcleo pode determinar quais as características de terreno que o *plugin* consegue gerar e a interface para o utilizador pode listar informação dos *plugins* disponíveis.

As ferramentas de parametrização e de pré-visualização disponíveis são definidas pelo núcleo da *framework*. As ferramentas, necessárias para a parametrização da técnica de modelação procedimental implementada, são listadas pelo *plugin*, com os respetivos metadados. Um *plugin* pode listar várias ferramentas de parametrização e pré-visualização em simultâneo. A ordem e os metadados das ferramentas listadas pelo *plugin* influenciam a interface para o utilizador para a respetiva camada de terreno.

As ferramentas de parametrização, depois de editadas na interface para o utilizador, dão origem a tipos de dados bem definidos que servem de entrada na execução da técnica de modelação procedimental implementada no respetivo *plugin*. Esses tipos de dados podem ser uma matriz de valores, *arrays*, variáveis, entre outros. Os resultados da execução da técnica devem respeitar o tipo de dados esperado pelas ferramentas de pré-visualização.

Um objeto global do núcleo da *framework* será sempre um parâmetro de entrada na execução da técnica implementada pelo *plugin*. Esse objeto transporta a informação do terreno, acumulada pelo *workflow* de criação, para que as técnicas de modelação procedimental possam considerar as características já criadas pelas camadas anteriores. No fim da execução, o *plugin*

altera o objeto com os resultados para que a próxima camada do *workflow* dê continuidade ao processo de criação.

5.5 Edição do *Workflow* e Escolha de Plugins

Como identificado na secção 5.1, o *workflow* do processo de modelação é constituído pelos *plugins* que implementam as técnicas de modelação procedimental. Na análise efetuada na secção 3.4, foram identificadas algumas dificuldades na implementação de um modelo baseado num *workflow* com diferentes técnicas de modelação procedimental. Para solucionar estas dificuldades, é proposto um *workflow* configurável (Figura 8).

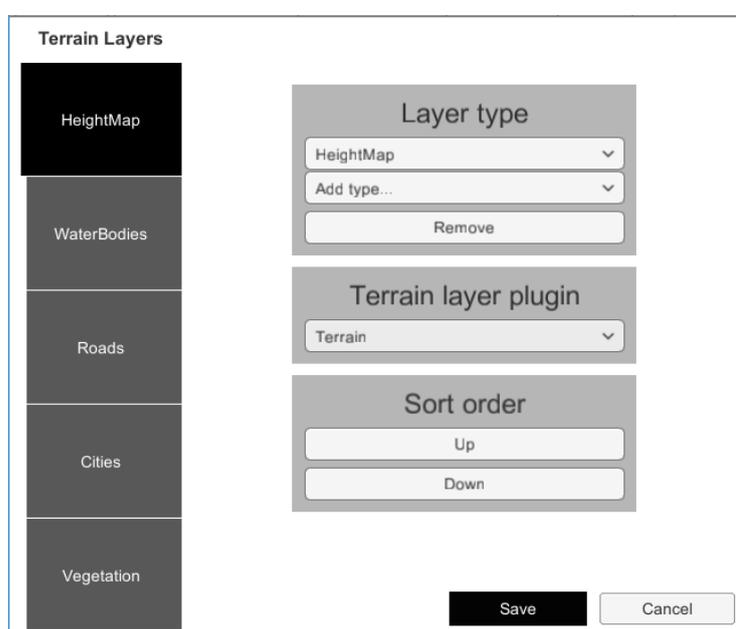


Figura 8 – Protótipo da configuração do *workflow* de modelação

Ao contrário do contributo no qual se baseia este modelo (R M Smelik et al. 2008), a ordem das características de terreno no *workflow* do processo de modelação não deve ser fixa, mas sim permitir a sua reordenação. Além disso, a configuração do *workflow* deve permitir que várias características de terreno possam ser tratadas numa mesma camada. Desta forma, uma técnica de modelação procedimental que aborde uma característica de terreno com uma ordem diferente ou que modele mais do que uma característica pode ser integrada no *workflow*.

Este tipo de configuração implica a adição de metadados e de restrições na configuração dos *plugins* de modelação procedimental que constituem o *workflow*, mantendo a coerência. Sendo assim, os *plugins* devem identificar três listas que enumeram tipos de camadas: as camadas que modela, as camadas que necessita para efetuar a modelação e as camadas que não podem estar definidas. Com estas restrições, o *plugin* consegue controlar de que forma pode ser posicionado num *workflow* de modelação, para que tenha a informação necessária aos algoritmos que o constitui, para que os dados que modela sejam os esperados pelo

utilizador e para que não entre em conflito com a modelação das camadas anteriores não previstas. Um *plugin*, por exemplo, que modela as características de terreno do mapa de elevação e de rios, oceanos e lagos, só pode ser atribuído a uma camada do *workflow* configurada com essas características. Se um *plugin* indicar que necessita dos dados obtidos da camada de vegetação, só pode ser configurado no *workflow* após a camada de vegetação. Num outro exemplo, se um *plugin* não permitir que a modelação prévia de estradas seja efetuada, só permitirá a sua configuração antes da camada do *workflow* que modela estradas. Da mesma forma, um *plugin* que não permita a modelação prévia de qualquer outra característica, só poderá ser atribuído à primeira camada do *workflow*.

5.6 Fluxo de Dados

Para permitir a comunicação entre técnicas de modelação procedimental que implementam diferentes características de terreno, é necessário a definição de uma estrutura onde os dados gerados são colocados. Esta estrutura é fornecida sempre que uma técnica de modelação procedimental é executada, para que tenha acesso e considere os dados obtidos das camadas do *workflow* já processadas e para que acrescente os seus dados gerados. Este fluxo permite que o *workflow* dê origem a um modelo coerente, já que cada técnica de modelação procedimental integra a característica de terreno que modela, acrescentando, adaptando ou mesmo modificando as características de terreno processadas em camadas do *workflow* anteriores.

Para que o fluxo de dados funcione e cada característica de terreno consiga interpretar a informação obtida das camadas anteriores, a estrutura de dados assemelha-se a um conjunto de *slots*, que são preenchidos à medida que as características de terreno vão sendo processadas, por cada camada do *workflow*. Cada um desses *slots* é um objeto que contém os dados que descrevem o modelo obtido de uma característica de terreno. Dada a particularidade de cada característica de terreno, existe um objeto específico para cada uma dessas características. Cada objeto contém dados que permitem a descrição da característica de terreno que representa para permitir a interpretação e integração pelas camadas seguintes no *workflow*. Uma camada do *workflow*, além de interpretar os dados obtidos de outras características de terreno, pode modificar essa informação para atender à modelação que acrescenta e evitar possíveis sobreposições ou erros de coerência nas camadas que a sucedem.

Na secção 5.5 foi descrita a configuração de um *workflow* de modelação procedimental e das restrições quanto às características de terreno que uma camada do *workflow* modela, camadas necessárias à sua execução e camadas que não podem ser modeladas previamente. Essas restrições estão relacionadas com os objetos do fluxo de dados que uma determinada técnica de modelação procedimental espera receber, deverá acrescentar ou não considera no seu processamento.

Para cada característica de terreno foi efetuado o levantamento dos dados necessários para a definição dos objetos que constituem o fluxo de dados do *workflow*. Na Tabela 37, Tabela 38,

Tabela 39, Tabela 40 e Tabela 41 são descritos os dados que permitem definir os modelos gerados pelas diferentes camadas de terreno.

Tabela 37 – Dados de descrição de mapas de elevação

Nome	Descrição
heightMap : float[][]	Grelha onde cada célula contém um valor de elevação entre 0 e 1
cellSize : float	Tamanho, em metros, do lado de uma célula do mapa de elevação
minHeight : float	Elevação mínima, em metros, representada pelo mapa de elevação
maxHeight : float	Elevação máxima, em metros, representada pelo mapa de elevação

Tabela 38 – Dados de descrição de rios, oceanos e lagos

Nome	Descrição
waterPaths : RiverPath[]	Descrição da geometria dos contornos dos rios, oceanos e lagos
waterModels : TerrainOutputMesh[]	Modelos 3D gerados dos rios, oceanos e lagos

Tabela 39 – Dados de descrição de vegetação

Nome	Descrição
vegetationModels : TerrainOutputMesh[]	Modelos 3D gerados da vegetação
vegetationAreas : ObjectArea[]	Descrição da posição de cada elemento de vegetação e do seu limite

Tabela 40 – Dados de descrição de redes de estradas

Nome	Descrição
roadsPath : RoadPath[]	Descrição do percurso, largura e tipo das estradas
roadModels : TerrainOutputMesh[]	Modelos 3D das estradas geradas

Tabela 41 – Dados de descrição de centros urbanos

Nome	Descrição
streetsPath : RoadPath[]	Descrição do percurso, largura e tipo de ruas
buildingAreas : ObjectArea[]	Descrição da posição de cada edifício ou objeto urbano e o seu limite
streetsModel : TerrainOutputMesh[]	Modelos 3D das ruas
buildingModels : TerrainOutputMesh[]	Modelos 3D de edifícios e objetos urbanos

Quando uma camada do *workflow* é executada, a técnica de modelação procedimental reconhece o objeto que necessita obter do fluxo de dados e, no final da modelação, acrescenta, ao fluxo de dados do *workflow*, os objetos correspondentes às camadas de terreno que modelou.

5.7 Principais Estruturas de Dados

As estruturas de dados utilizadas num projeto de terreno, para a definição de ferramentas de edição e visualização de dados, fluxo de dados do *workflow* e comunicação com os *plugins* de modelação procedimental, fazem parte do núcleo da *framework*. No entanto, para que essas estruturas possam ser utilizadas de forma independente pela interface para o utilizador e pelos *plugins* de modelação procedimental, a sua implementação é efetuada numa biblioteca independente. Desta forma, a biblioteca *TerrainLayerBase* é parte integral do núcleo da *framework* e é disponibilizada para construção da interface para o utilizador e para permitir a implementação dos *plugins* de modelação. Na Figura 9 é apresentado o diagrama de classes desta biblioteca, tendo sido omissas algumas classes por simplicidade.

Para que um *plugin* de modelação procedimental possa ser identificado e permitir a comunicação com o núcleo da *framework*, a biblioteca *TerrainLayerBase* deve ser importada pelo *plugin* e a interface *TerrainPlugin* deve ser implementada. O núcleo da *framework* usa esta interface para procurar a biblioteca que constitui o *plugin* de modelação procedimental usando *reflection*. A interface *TerrainPlugin* impõe a indicação de informação por parte do *plugin* que identifica o título, autor, descrição, ferramentas de parametrização e restrições. As restrições, tal como descrito na secção 5.5, são indicadas por três listas de tipos de camada do *workflow*. Na interface *TerrainPlugin*, *outputLayerTypes* corresponde às características de terreno que o *plugin* modela, *inputLayerTypes* corresponde às características de terreno que necessita e *forbiddenLayerTypes* são as características que não são suportadas pelo *plugin*. A informação imposta pela interface, além de permitir a gestão dos *plugins* pelo núcleo da *framework*, disponibiliza informação para que o módulo da interface para o utilizador crie a configuração do *workflow* e escolha de *plugins*. A interface impõe a implementação do método *startProcess* que dá início à execução dos algoritmos de modelação procedimental. Num projeto de terreno, o elemento principal do *workflow* de modelação são os *plugins* de modelação. Quando é iniciado o processamento de um *workflow* de modelação, os *plugins* são executados, isoladamente e na sequência com que foram configurados.

As interfaces *TerrainInput* e *TerrainOutput* permitem a listagem, por parte dos *plugins* de modelação procedimental, das implementações das classes que descrevem as ferramentas de parametrização e visualização de resultados. Os objetos que descrevem as ferramentas de parametrização contém um atributo *value* onde a interface para o utilizador coloca os dados introduzidos pelo utilizador. As instâncias que descrevem a ferramenta de parametrização são usadas como parâmetro de entrada no método *startProcess*, sendo este o meio com que os dados introduzidos pelo utilizador são recebidos pelo *plugin*.

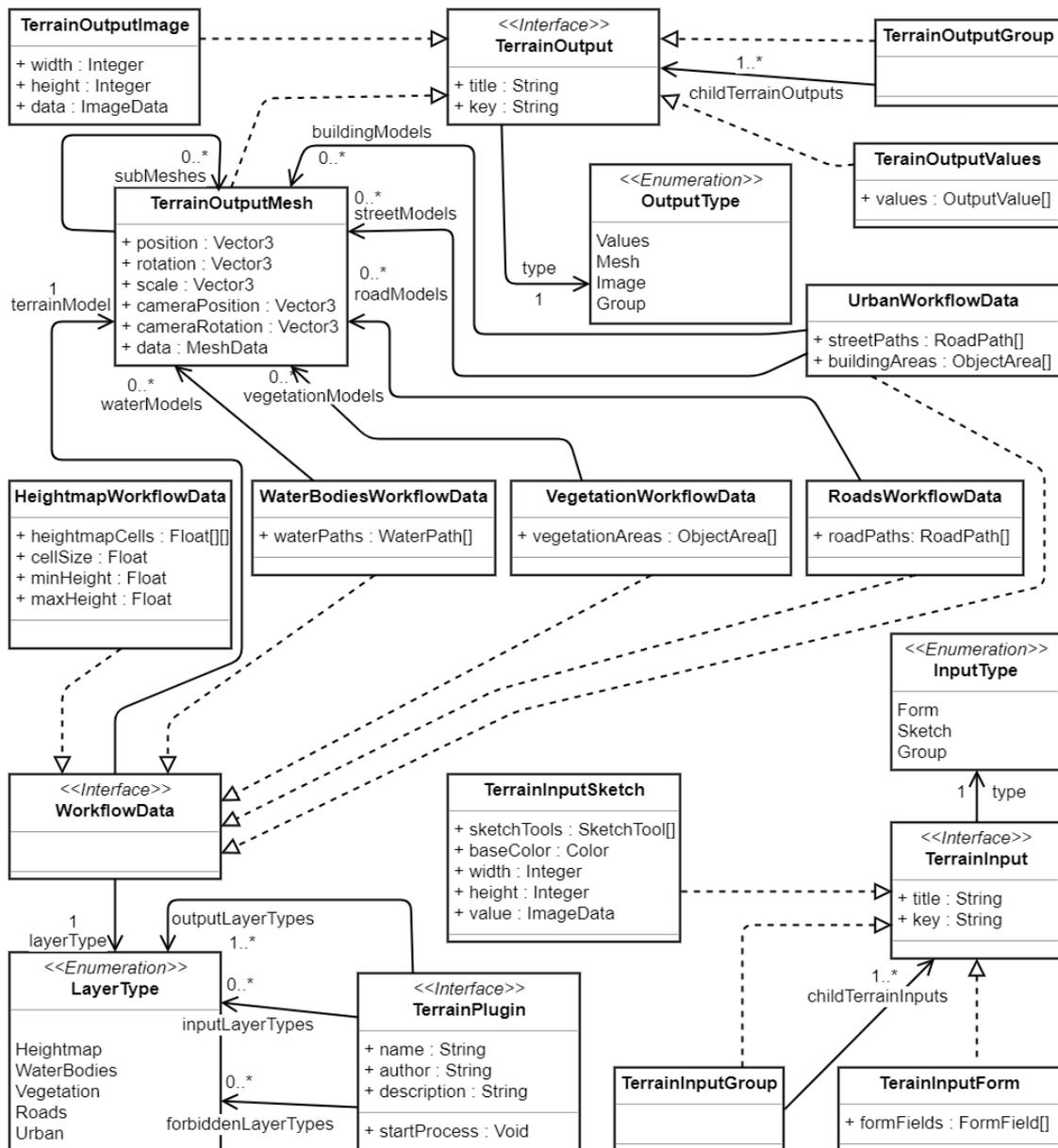


Figura 9 – Diagrama de classes da biblioteca *TerrainLayerBase*

A interface *WorkflowData* agrupa os objetos que contém os dados que descrevem o resultado das técnicas de modelação procedimental para cada característica de terreno. Quando o processo de modelação é iniciado, o projeto de terreno cria um novo fluxo de dados, que é fornecido aos *plugins* de modelação procedimental que constituem o *workflow*, e que vai sendo preenchido com instâncias de objetos que implementam a interface *WorkflowData*. O fluxo de dados é uma tabela *hash* onde as chaves são os tipos de característica de terreno e o valor são instâncias de *WorkflowData*. Se o processo de modelação for parcial, ou seja, se já existirem camadas do *workflow* que já tenham sido processadas e que não tenham sido escolhidas para serem processadas novamente (ver secção 5.3), o novo fluxo de dados é inicialmente preenchido com as instâncias de *WorkflowData* das camadas que não vão ser processadas.

Um projeto de terreno é constituído pela configuração de um *workflow* de modelação, pelas ferramentas de parametrização e visualização (se as respetivas camadas já tiverem sido processadas) das camadas de terreno configuradas e do fluxo de dados com os dados processados. Esses dados são serializados e guardados no sistema de ficheiros.

5.8 Resumo do Capítulo

Neste capítulo, foi apresentada a arquitetura da solução concebida, o seu funcionamento e as estruturas de apoio ao processo de modelação.

A *framework* desenvolvida é constituída por três módulos principais que podem existir de forma independente, os *plugins* que implementam as técnicas de modelação procedimental, o núcleo que organiza os projetos de terreno e controla o processo de modelação e a interface para o utilizador.

O processo de modelação foi definido como a execução sequencial dos *plugins* de modelação procedimental pela ordem com que foram pré-configurados, definindo um *workflow*. Essa organização obedece a regras relacionadas com o tipo de dados que os *plugins* de modelação esperam receber de outras camadas ou retornar às camadas seguintes, para que os algoritmos de modelação procedimental implementados disponham da informação necessária para a sua execução.

Por fim foram apresentadas as principais estruturas de dados disponibilizadas pelo núcleo da *framework* e identificados os tipos de dados resultantes do processo de modelação para cada característica de terreno.

6 Implementação

Considerando o *design* da solução proposta e o estudo efetuado das diferentes técnicas de modelação procedimental de terrenos, é apresentada e descrita, neste capítulo, a implementação da *framework* e dos *plugins* de modelação procedimental.

6.1 Framework

Terraframe, a *framework* de modelação procedimental desenvolvida (secção 4.1, Figura 4), cria um sistema e convenções para a comunicação entre os procedimentos que modelam diferentes características de terreno. Desta forma, é disponibilizado um ambiente onde programadores de *software* e investigadores podem integrar, disponibilizar e desenvolver novas técnicas de modelação procedimental. Assim, o desenvolvimento é mais eficiente, utilizando outras técnicas para produzir dados de entrada e utilizando ferramentas disponibilizadas pela *framework* para parametrização dos processos e visualização de modelos.

Na geração de modelos 3D de terrenos por artistas, a *framework* permite a escolha e a combinação de diferentes técnicas de modelação procedimental integradas, de forma a obter os resultados mais adequados à utilização pretendida. As ferramentas de visualização de resultados e de parametrização garantem a controlabilidade do processo. O processo de modelação é efetuado pela combinação das diferentes técnicas que são executadas pela ordem com que foram pré configuradas e/ou restrições que impõem. O processo pode ser interrompido pelo utilizador numa determinada fase e parcialmente executado, produzindo resultados intermédios. Os parâmetros podem ser editados e o processo reiniciado partindo de uma determinada técnica intermédia.

Nas seguintes secções é descrito o funcionamento dos diferentes componentes que constituem a *framework*.

6.1.1 Sistema de Plugins

A *framework* desenvolvida permite a implementação de técnicas de modelação procedimental como módulos separados da aplicação principal. Esses módulos implementam uma interface que os identifica e que define a comunicação com a *framework*. Desta forma, cada módulo que implementa uma técnica de modelação procedimental constitui um *plugin* adicionado a um diretório onde é identificado e acoplado dinamicamente pela *framework*. Uma interface de *software* impõe a implementação de métodos que retornam informação necessária para a identificação do *plugin*, tal como o nome, as camadas de terreno que modela, restrições, autores e descrição.

Além de retornar informação para a identificação do *plugin*, a *framework* impõe a implementação de um método que dá início ao processo de modelação procedimental deste módulo. Este método recebe a estrutura de dados partilhada e os parâmetros de entrada e retorna os resultados. Um *handler* é também recebido por este método para comunicar o progresso do processo de modelação.

Os *plugins* são instanciados recorrendo à utilização de *reflection*, através das classes do *namespace System.Reflection* da *framework*. NET e carregados dinamicamente. A leitura da informação que identifica o *plugin* é usada para criar a interface de escolha de *plugins* e impor as restrições que influenciam a ordem com que os *plugins* são organizadas no *workflow* de modelação. A escolha de *plugins* e configuração do *workflow* são descritas na secção 6.1.3.

Para implementar um novo *plugin* de modelação procedimental, a *framework* disponibiliza uma biblioteca *TerrainLayerBase* para auxiliar a integração. Essa biblioteca contém a interface que deve ser implementada bem como os objetos descritores das ferramentas de parametrização dos processos de modelação e visualização dos resultados e do fluxo de dados do *workflow*.

6.1.2 Projeto de Terreno

Um projeto de terreno é constituído pela configuração de um *workflow* de modelação procedimental, parâmetros e resultados das diferentes camadas processadas do terreno.



Figura 10 – Vista de escolha de um projeto de terreno

A vista de entrada da *framework* permite escolher um projeto existente ou criar um novo projeto de terreno. Os projetos são apresentados com um *thumbnail* do modelo do terreno obtido da última modelação, título e uma barra contendo ícones que representam as camadas de terreno configuradas e processadas (Figura 10).

Os ícones das camadas processadas são apresentados com cor e os ícones brancos representam as camadas configuradas que ainda não foram processadas. Esta forma de sinalização permite informar o utilizador do estado do processo de modelação configurado ou se o processo já foi concluído.

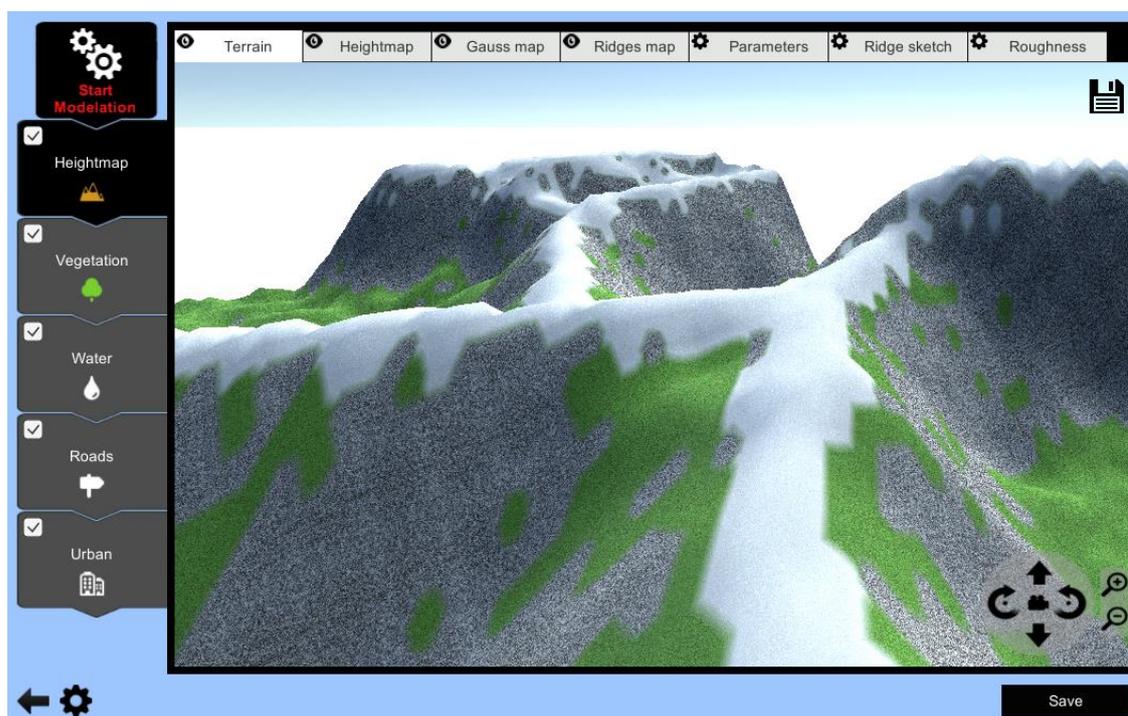


Figura 11 – Vista de projeto de terreno

Ao criar um novo projeto, o *workflow* de modelação do último projeto de terreno aberto é utilizado. Se não existir nenhum projeto, é aberta a configuração do *workflow*. Num projeto com um *workflow* configurado, o *workflow* pode ser alterado utilizando o botão com o ícone de uma roda dentada no canto inferior esquerdo na vista de projeto (Figura 11).

A vista de projeto é organizada em *tabs* verticais e horizontais. Existem *tabs* verticais, do lado esquerdo, por cada camada, ou conjunto de camadas de terreno. No conjunto, os *tabs* verticais representam o *workflow* de modelação, com a ordem do topo para parte inferior da vista. De forma semelhante à vista de escolha de projeto, ícones são utilizados para auxiliar na identificação das camadas de terreno que cada *tab* representa. Nas camadas processadas o ícone é colorido e nas camadas configuradas e por processar, o ícone é branco. Selecionando um *tab* vertical de uma camada de terreno, são exibidos os *tabs* horizontais que permitem selecionar as ferramentas de parametrização e as ferramentas de visualização de resultados (Figura 12).

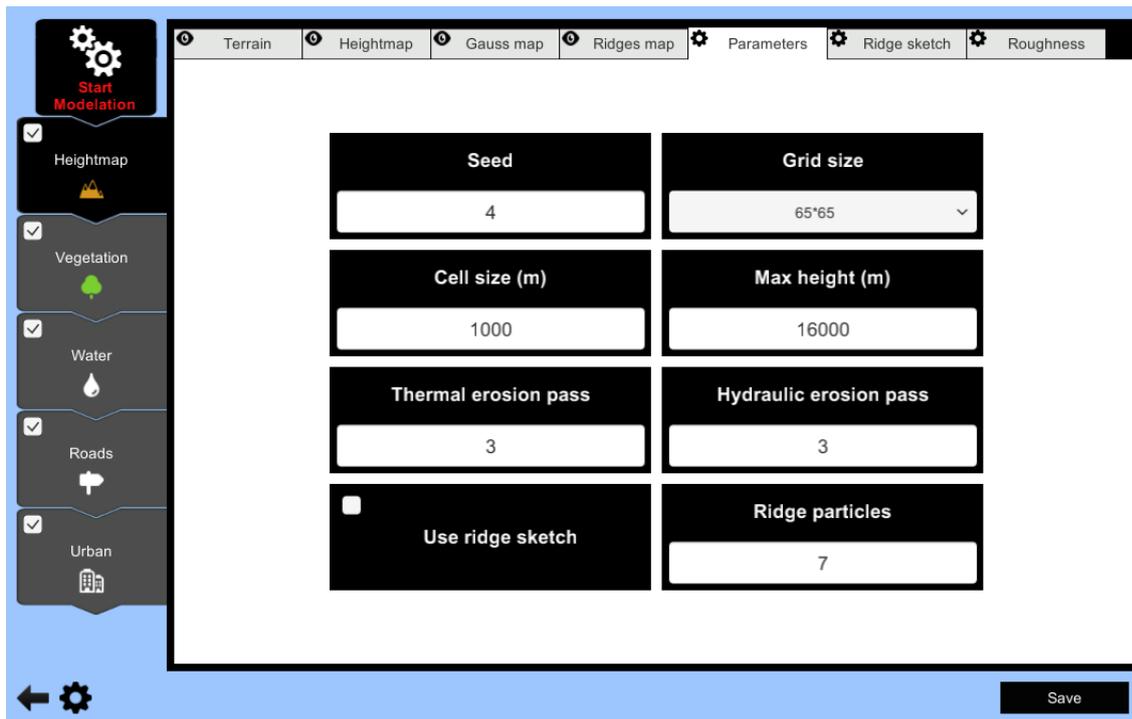


Figura 12 – Edição de parâmetros na vista de projeto de terreno

Os *tabs* de parametrização são identificados pelo ícone de uma roda dentada e os *tabs* de visualização de resultados são identificadas por um ícone de visualização. Se a camada ou conjunto de camadas de terreno ainda não tiver sido processada, os *tabs* de visualização de resultados não são apresentados.

6.1.3 Configuração do *Workflow* de Modelação

O *workflow* de modelação é constituído pelos *plugins* de modelação procedimental e a ordem com a qual estão organizados. A sua organização atende às restrições impostas pelas estruturas necessárias para a execução de cada camada. Assim, um *plugin* só pode ser incluído no *workflow* se as estruturas de dados necessárias para a sua execução forem fornecidas pelos *plugins* anteriores do *workflow*.

O *workflow* é configurado pela ordenação e junção de camadas funcionais no processo de modelação e escolha de *plugins* de modelação compatíveis quanto às camadas funcionais que implementam.

A vista de configuração do *workflow* de modelação é constituída por *tabs* verticais (Figura 13). Tal como na vista de projeto, o conjunto dos *tabs* verticais e a sua ordem representam o *workflow* de modelação procedimental. Selecionado um *tab* são exibidos controlos que permitem escolher as camadas funcionais do processo de modelação de terrenos e mudar a ordem com a qual são executados no *workflow*.

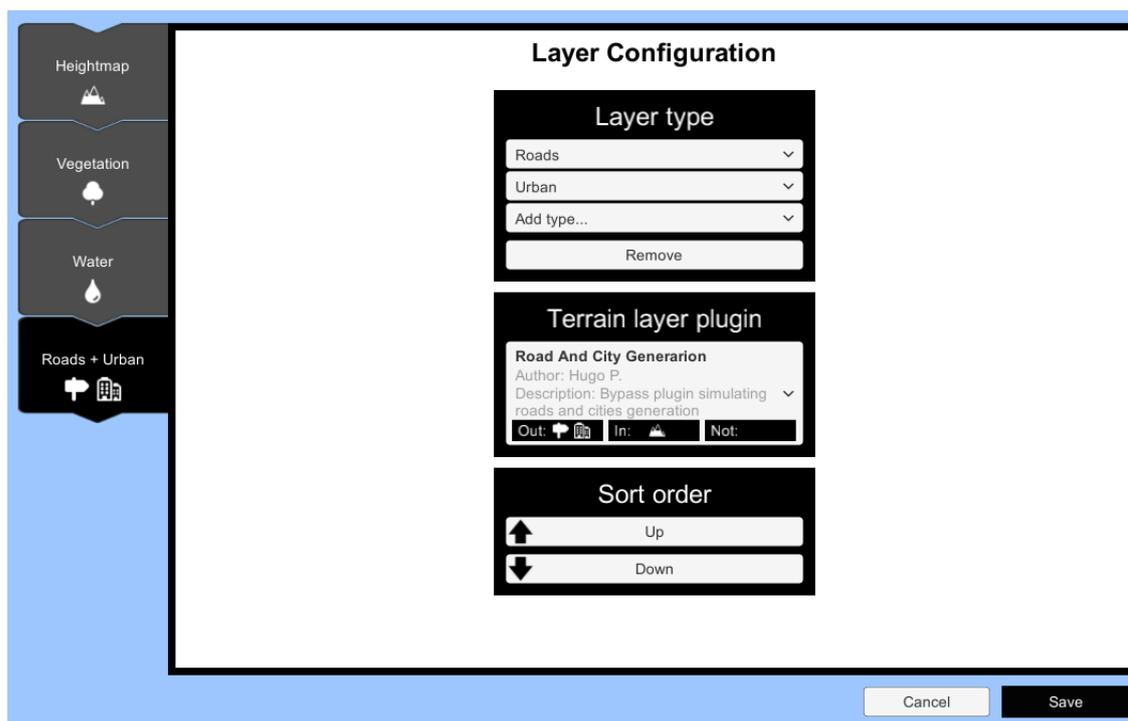


Figura 13 – Vista de configuração do *workflow* de modelação

Um outro controlo permite escolher de uma lista o *plugin* de modelação procedimental compatível com a camada do *workflow* configurada (Figura 14). A ordem e junção das camadas de terreno não apresentam qualquer restrição na sua organização, no entanto, a escolha dos *plugins* apresentados é restringida pelas camadas funcionais que modelam. Ou seja, num *tab* que represente a modelação da camada de mapas de elevação e da camada de rios, lagos e oceanos, só serão apresentados para escolher os *plugins* que modelem simultaneamente os mapas de elevação e os rios, oceanos e lagos. Uma outra restrição na escolha de *plugins* são as camadas anteriores do *workflow*. Um *plugin* pode exigir que uma determinada camada do *workflow* seja, ou não, processada previamente. Esta restrição prende-se com o facto de o *plugin* poder necessitar dos dados gerados numa outra camada para efetuar a sua modelação ou o seu processamento não ser compatível, ou não considerar os dados de outras camadas.

Para que o utilizador possa identificar as restrições dos *plugins* e ajustar as camadas para os compatibilizar, todos os *plugins* são apresentados na lista, sendo identificados os *plugins* incompatíveis com um sombreamento vermelho. Cada *plugin* é identificado na lista, pelo título, autor, descrição e restrições. As restrições são separadas em três barras de ícones. A barra *Out* contém os ícones das camadas de modelação de terreno que modela. A barra *In* os ícones das camadas que devem ser processadas previamente no *workflow*. Por último, a barra *Not* contém os ícones das camadas que não podem ser processadas previamente. Quando um *plugin* incompatível é listado, além do sombreamento, a barra de restrição com conflito é apresentada com um fundo vermelho.

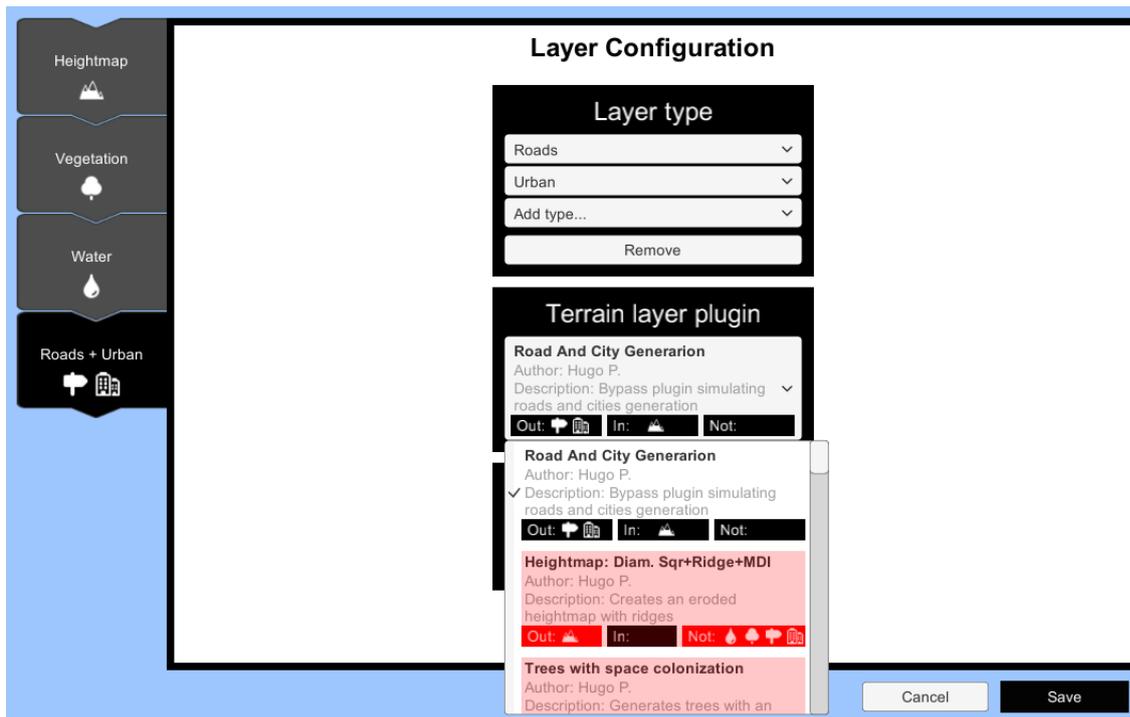


Figura 14 – Lista de *plugins* disponíveis e respetivas restrições

No exemplo apresentado na Figura 14, é visível um *plugin* listado no qual as restrições das camadas de terreno que modela e das camadas de terreno que não podem estar configuradas falham. Tal como é indicado pelo fundo vermelho das barras de restrição *Out* e *Not*. A camada do *workflow* foi configurada como última camada do workflow e de forma a modelar a rede de estradas e centros urbanos, no entanto, o *plugin* incompatível, modela mapas de elevação e não aceita nenhuma camada de terreno anterior. Por restringir a configuração de qualquer outra camada anterior, este *plugin* só pode ser utilizado numa primeira camada do *workflow* que modele mapas de elevação.

A configuração de um *workflow* só pode ser guardada se não existir nenhum erro de validação e se não existirem camadas intermédias por configurar. Ou seja, o *workflow* não obriga a configuração de todas as camadas funcionais na modelação de terrenos, mas as camadas que tenham um *plugin* definido devem ser contíguas e devem estar no início do *workflow*. Depois de guardar a configuração do *workflow* de modelação, na vista de projeto, só são exibidos os *tabs* que representam as camadas configuradas.

6.1.4 Geração de Terrenos

Tendo um *workflow* de modelação configurado, o processo de modelação é efetuado seguindo a ordem com a qual as camadas de terreno e os respetivos *plugins* que as implementam estão organizados.

Para iniciar o processo de modelação, é necessário escolher quais as camadas do *workflow* que devem ser processadas, selecionando-as através da *checkbox* no *tab* vertical. O processo é iniciado pressionando o botão de iniciar modelação no topo dos *tabs* das camadas de terreno (Figura 15).



Figura 15 – Detalhe do botão para iniciar a modelação e a seleção de camadas de terreno

Depois de terminado o processo de modelação, *tabs* horizontais com ferramentas de visualização são acrescentadas em cada camada do *workflow* para que o utilizador possa avaliar os resultados.

Para processar uma camada de terreno isolada, é necessário que as camadas anteriores tenham sido previamente processadas, caso contrário, é apresentada uma mensagem de erro. Esta restrição prende-se com a necessidade das camadas intermédias num *workflow* de modelação necessitarem dos dados resultantes das camadas anteriores para poderem ser executadas.

A escolha das camadas de terreno a processar permite que o processo de modelação possa correr de forma parcial, modelando apenas um conjunto das características de terreno. Um outro objetivo da modelação parcial é a possibilidade de permitir alterar os parâmetros de uma camada de modelação intermédia para ajustar o modelo resultante da combinação das camadas. Desta forma, o utilizador pode ajustar os parâmetros de uma camada intermédia e iniciar o processamento da camada ajustada e das camadas seguintes até obter o resultado pretendido, de uma forma iterativa.

Como a modelação das camadas seguintes do *workflow* depende do resultado das camadas anteriores, sempre que uma camada intermédia é isoladamente processada, os resultados previamente processados das camadas seguintes são eliminados. Desta forma é garantido que os resultados da modelação das diferentes camadas são o resultado da combinação das

camadas anteriores, e que a alteração dos parâmetros de uma camada anterior pode influenciar o modelo gerado pela camada seguinte.

6.1.5 Ferramentas de Visualização e Edição

Sendo uma das dificuldades identificadas na adoção de técnicas de modelação procedimental na geração de terrenos, a carência de métodos de parametrização constitui uma importante motivação no desenvolvimento da *framework*. Por esse motivo, as classes de suporte à definição das ferramentas de parametrização e visualização de dados são incluídas na biblioteca utilizada na implementação dos *plugins* de modelação procedimental. Os *plugins* de modelação procedimental definem, utilizando essas classes base, o tipo de ferramentas e a sua configuração. O núcleo da *framework* usa essa definição para obter e guardar os parâmetros introduzidos pelo utilizador, associado a um projeto de terreno. A interface para o utilizador usa a definição para apresentar as ferramentas editáveis e de visualização com as características configuradas pelo *plugin*.

As classes base de definição das ferramentas de edição de parâmetros e visualização de dados são também utilizadas como meio de comunicação dos dados de entrada e saída dos *plugins*. Os parâmetros editados pelo utilizador nas ferramentas de parametrização são transportados na própria definição da ferramenta de parametrização, que é depois utilizada pelo núcleo ao guardar os parâmetros e serve como dados de entrada para iniciar a modelação no *plugin*. De igual modo, os dados resultantes do procedimento de modelação são guardados na definição da ferramenta de visualização para que os resultados sejam guardados pelo núcleo e para que a interface para o utilizador possa apresentar a informação ao utilizador.

O mecanismo de definição de ferramentas de parametrização e visualização foi implementado para permitir a expansão de novas ferramentas. Os descritores de editores de parâmetros implementam uma mesma interface, contendo o título, a chave e o tipo de editor. De igual modo, os visualizadores de resultados implementam uma mesma interface contendo os mesmos atributos.

Foram essencialmente implementadas três ferramentas de parametrização, o editor de valores, o editor de esboços e a ferramenta de agrupar (Figura 16).

O editor de atributos (Figura 16 *b*) permite a edição de valores numéricos, texto e opções. O editor é configurado como uma lista de variáveis. Essas variáveis têm a sua própria configuração, como o tipo, o título e a chave e são derivadas de uma mesma interface. Dependendo da implementação, outros atributos permitem configurar a variável, tal como o limite mínimo e máximo, opções disponíveis entre outros.

O editor de esboços (Figura 16 *a* e *c*) é uma ferramenta semelhante a um editor de imagem, mas com ferramentas de desenho bem definidas e sem grande precisão. É usado para a parametrização através de uma imagem. A configuração deste meio de parametrização é efetuada pela indicação do tamanho do esboço pretendido, cor base do desenho e a listagem

de ferramentas de esboço. As ferramentas de esboço herdam a mesma interface e todas contêm um título. Dependendo da implementação da ferramenta, outros atributos podem ser definidos, tal como a cor e a espessura da linha entre outros. Esta ferramenta permite a importação de uma imagem externa, que é processada e adaptada para atender à sua configuração.

Apesar de ser uma ferramenta de organização e de não transportar dados de parametrização, a ferramenta de agrupar (Figura 16 c) permite juntar um conjunto de ferramentas de parametrização para que possam ser selecionadas numa lista e facilitar a sua organização. É útil nos casos em que existe a parametrização de múltiplos elementos semelhantes no processo de modelação procedimental.

Quanto às ferramentas de visualização, foram implementadas a visualização de imagem, a visualização de *mesh*, a visualização de valores e ferramenta de grupo.

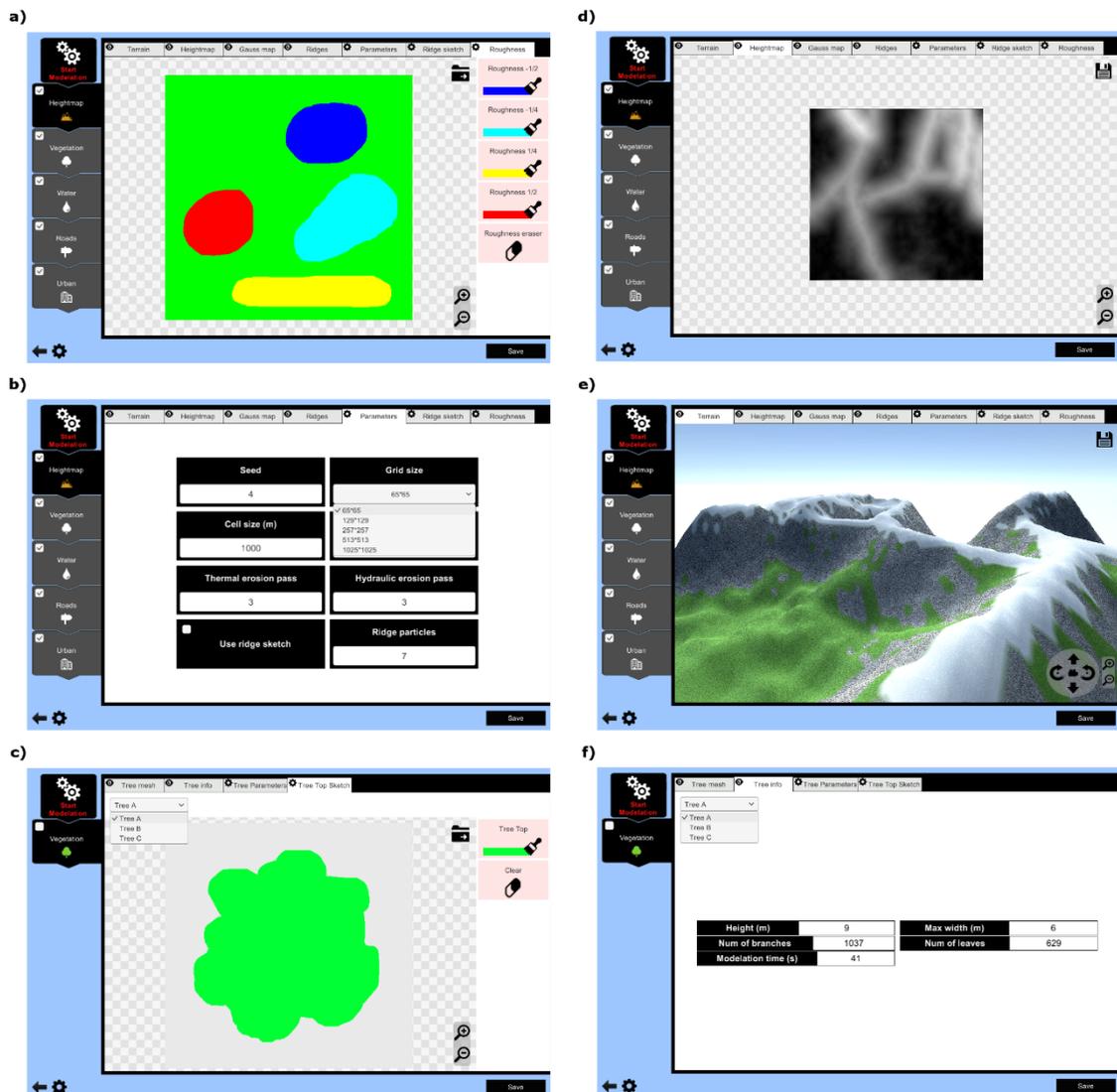


Figura 16 – Diferentes ferramentas de edição de parâmetros e visualização de resultados

A visualização de imagem (Figura 16 d) permite visualizar um esboço ou um mapa resultante do processo de modelação. A ferramenta permite mover e aumentar a imagem, bem como exportar o resultado para um ficheiro externo.

A ferramenta de *mesh* (Figura 16 e) permite a visualização dos modelos 3D, através da definição de vértices, vértices de textura, vetores normais, faces e materiais. O visualizador pode ser controlado com o rato do computador para alterar o ângulo, posição e *zoom* na visualização do modelo. Esta ferramenta permite a exportação de um modelo 3D.

O visualizador de valores (Figura 16 f) permite a consulta de resultados numéricos e de texto.

Tal como nas ferramentas de parametrização, também foi implementada uma ferramenta para agrupar os visualizadores de resultados (Figura 16 f).

6.2 Técnicas de Modelação Procedimental

Na secção 3.8, foram analisadas as técnicas de modelação procedimental identificadas no estado da arte (capítulo 2). Recorrendo à utilização da técnica AHP, as técnicas de modelação que melhor se adequam na integração na *framework* desenvolvida neste projeto foram escolhidas considerando os critérios de facilidade de integração, facilidade de implementação, qualidade dos resultados e relevância histórica. Nesta secção é descrito o funcionamento e a implementação dos algoritmos e procedimentos que constituem as técnicas de modelação procedimental implementadas, quais as características de terreno que constituem e de que forma foram integradas na *framework* desenvolvida. Das técnicas escolhidas pelo método AHP, foram implementadas duas técnicas que modelam respetivamente mapas de elevação e vegetação.

A técnica escolhida para a modelação de mapas de elevação foi a técnica apresentada por Belhadj & Audibert (2005), o algoritmo *triangular edge* com linha de serra e MDI. Esta técnica resulta da combinação de três algoritmos: o *triangular edge*, o MDI e o algoritmo de geração de linhas de serra. Nesta implementação, o algoritmo *triangular edge* foi substituído pelo algoritmo *diamond square*, sendo também um algoritmo de subdivisão. Para melhorar o realismo do mapa de elevação gerado, foram acrescentados dois algoritmos de simulação de erosão, a erosão térmica e erosão hidráulica. Os algoritmos que foram implementados e que constituem a modelação dos mapas de elevação são descritos separadamente, por simplicidade, nas secções 6.2.1, 6.2.2, 6.2.3, 6.2.5 e 6.2.6. Na secção 6.2.8 é explicada a combinação dos algoritmos de forma a constituir a técnica de modelação procedimental e quais os meios e ferramentas usados na parametrização e exportação de resultados.

A modelação de vegetação foi implementada com base na abordagem apresentada por Runions et al. (2007), a criação de árvores com o algoritmo de colonização espacial. A

implementação do algoritmo de colonização espacial é descrita na secção 6.2.7, sendo explicada na secção 6.2.9 a parametrização e a exportação dos resultados.

6.2.1 Algoritmo *Diamond Square*

O algoritmo *diamond square* é um algoritmo de subdivisão recursiva, gerando fractais que podem ser utilizados como mapas de elevação (secção 2.1.1, Figura 1). Desta forma, este algoritmo pode, isoladamente, constituir a camada do *workflow* na geração de mapas de elevação, seguindo a organização do processo de modelação estabelecido na *framework* desenvolvida. Tendo sido descrito pela primeira vez por Fournier et al. (1982) e denominado de *diamond square* por Miller (1986) ao compara-lo com dois outros algoritmos de subdivisão, o *triangular edge* e o *square square*, tem sido utilizado e referido em estudos mais recentes devido à simplicidade, resultados heterogéneos e reduzida complexidade computacional.

Este algoritmo gera mapas de elevação representados por uma grelha bidimensional quadrada, de lado 2^n+1 . Cada célula da grelha contém um valor de elevação entre zero e um, permitindo que o valor da elevação possa ser facilmente adaptado para qualquer gama de valores, de acordo com uma escala, aproximando o modelo gerado de um terreno real.

O algoritmo pode ser descrito nos seguintes passos:

1. Inicialmente, é criada uma grelha vazia (Figura 17), onde as células sem elevação definida não contêm valor.

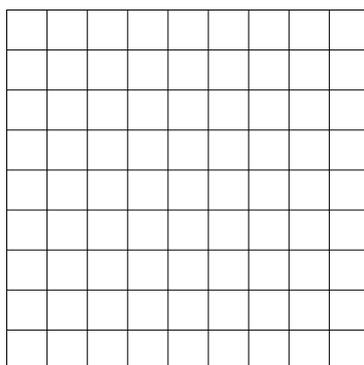


Figura 17 – Grelha sem elevações definidas

2. São atribuídos valores de elevação iniciais aleatórios aos cantos da grelha (Figura 18), ou seja, nas posições $[0, 0]$, $[0, n-1]$, $[n-1, 0]$ e $[n-1, n-1]$ considerando uma grelha de lado n .

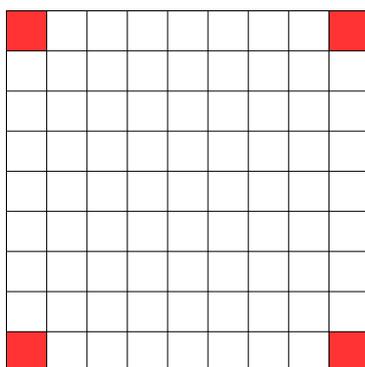


Figura 18 – Atribuição dos valores de elevação iniciais

3. É iniciada uma iteração de subdivisão da grade. Cada subgrade é delimitada, nos cantos, pelas células definidas na iteração anterior. Na primeira iteração, a subgrade é igual à grade inicial.
4. Para cada subgrade, é identificada a célula central (Figura 19), ou seja, a célula $[(n-1)/2, (n-1)/2]$ numa subgrade de lado n . A cada célula central é atribuído um valor obtido da média do valor das células dos cantos da respectiva subgrade. A esse valor é efetuado um desfasamento aleatório.

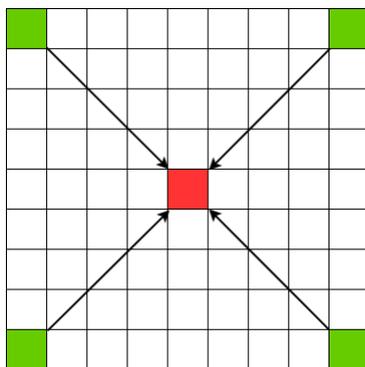


Figura 19 – Cálculo do valor da elevação da célula central

5. São identificadas as células centrais das laterais das subgrades (Figura 20), ou seja, as células $[0, (n-1)/2]$, $[(n-1)/2, 0]$, $[(n-1), (n-1)/2]$ e $[(n-1)/2, (n-1)]$ tendo uma subgrade de lado n . Estas células são partilhadas com subgrades adjacentes. Para cada uma dessas células é atribuído um valor obtido da média do valor das células mais próximas calculadas no passo 4). A esse valor é efetuado um desfasamento aleatório.
6. Uma nova iteração é iniciada, partindo do passo 3) até a grade ficar totalmente preenchida. A cada iteração, a grade é progressivamente subdividida, apresentando quatro vezes mais subgrades que na iteração anterior (Figura 21).

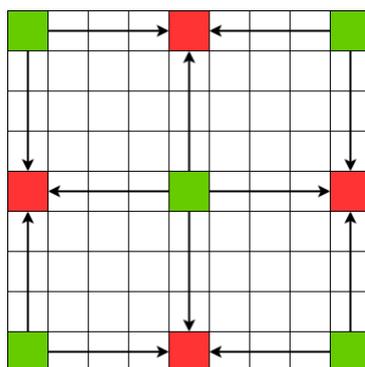


Figura 20 – Cálculo do valor da elevação das células centrais das laterais das subgradeiras

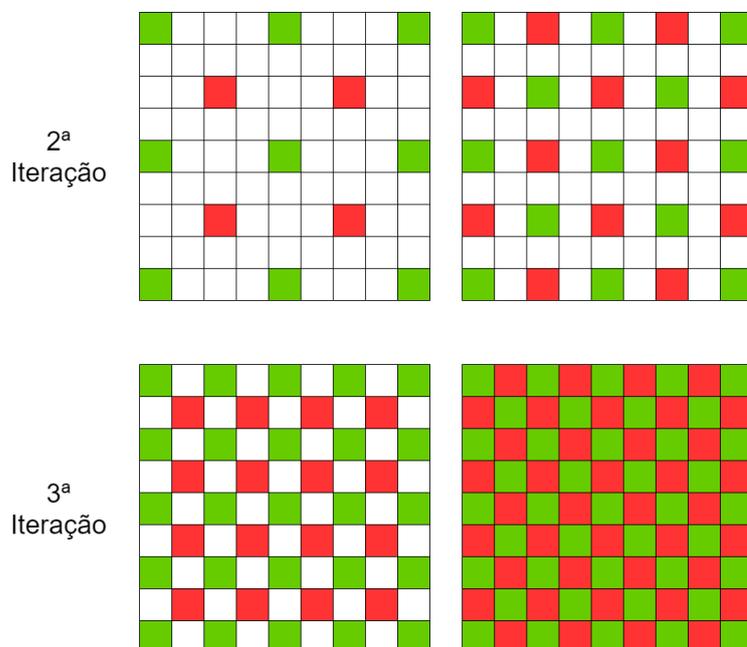


Figura 21 – Iterações de subdivisão para obter os valores de elevação

O desfasamento aleatório aplicado em cada média tem a magnitude reduzida a cada iteração, esta redução permite controlar a rugosidade do terreno gerado. Um fator de redução da magnitude do valor aleatório pode ser editado de forma a aumentar ou reduzir a rugosidade do mapa de elevação.

6.2.2 Algoritmo *Midpoint Displacement's Inverse*

Uma das dificuldades na controlabilidade dos algoritmos de subdivisão, como o *diamond square*, é a dificuldade em influenciar ou definir uma elevação específica num determinado ponto do mapa de elevação a ser gerado. O algoritmo parte da definição aleatória da elevação dos cantos de uma grade, sendo estes os únicos valores que podem ser influenciados ou

definidos arbitrariamente. Outros tipos de controlo possível nos algoritmos de subdivisão são a influência na obtenção de valores aleatórios ou na interpolação dos valores de elevação existentes.

A definição da elevação em determinados pontos do mapa de elevação, antes da execução do algoritmo de subdivisão, gera diferenças de elevação abruptas e descontinuidades nesses mesmos pontos. Este problema deve-se ao facto da elevação atribuída a qualquer célula na grelha que representa o mapa de elevação ter origem no valor de diferentes células que foram calculadas em diferentes iterações.

Um tipo de controlo com uma importância considerável na modelação de um terreno é a definição de áreas com elevações específicas que podem ter um propósito na utilização pretendida do terreno final. Além da controlabilidade, outras influências na elevação de determinados pontos do mapa de elevação, como a definição de linhas de serra ou a consideração de outras características de terreno, podem melhorar a qualidade dos resultados.

Para resolver o problema de continuidade, o algoritmo *midpoint displacement's inverse (MDI)* foi apresentado por Belhadj & Audibert (2005) para permitir a influência de linhas de serra e traçados de rio na aplicação do algoritmo de subdivisão *triangular edge*. Este algoritmo é aplicado antes do algoritmo de subdivisão e calcula a elevação dos pontos do mapa de elevação que influenciam as elevações previamente definidas. O algoritmo de subdivisão pode, desta forma, ser executado mantendo os valores obtidos do algoritmo MDI e mantendo os valores definidos garantindo a continuidade do mapa de elevação.

Assumindo o mapa de elevação a ser gerado, representado por uma grelha quadrada de lado 2^n+1 , com as mesmas características de uma grelha utilizada na aplicação de um algoritmo *midpoint displacement*. O algoritmo MDI pode ser explicado nos seguintes passos:

1. Para cada célula da grelha, são pré-calculadas as células que influenciam o cálculo da sua elevação na aplicação do algoritmo de subdivisão (Figura 22) e a respetiva distância. Esses dados são guardados num mapa indexado pela respetiva célula.

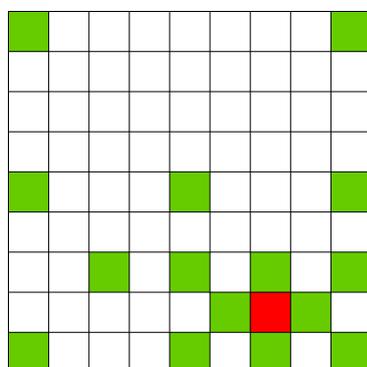


Figura 22 – A verde, as células que dão origem à elevação da célula vermelha no algoritmo *diamond square*

2. Numa grelha vazia (sem nenhum valor de elevação definido) são atribuídas as elevações dos pontos já conhecidos ou definidos pelo utilizador.
3. Todas as células da grelha são percorridas.
4. Quando uma célula é encontrada com o valor de elevação definido, o percurso da grelha termina e é iniciado o processo inverso de cálculo de elevação.
5. O processo inverso obtém, do mapa previamente calculado, as células que dão origem à célula encontrada. Para cada uma dessas células é calculada a elevação com base na elevação da célula encontrada e da sua distância.
6. Terminado o processo inverso, o algoritmo continua o passo 2), partindo da célula de paragem.

O cálculo da elevação das células que influenciam uma célula conhecida tem como base a sua distância e elevação. No estudo, no qual se baseia esta implementação, é efetuada uma interpolação com um desfasamento aleatório para a obtenção das elevações, no entanto, nesta implementação é utilizada a função de Gauss. Esta abordagem tem origem na observação do mapa de elevação resultante da aplicação da função de Gauss para a interpolação de pontos de elevação entre as linhas de serra e do potencial de controlabilidade.

A função de Gauss tem várias aplicações no ramo da matemática e física, sendo a sua utilização em estatística e probabilidade a mais conhecida. A característica que torna esta função relevante na sua utilização neste algoritmo é a curva simétrica em forma de “sino” do gráfico da função, a curva de Gauss. A função pode ser descrita da seguinte forma (Equação 3):

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{2c^2}} \quad (3)$$

Sendo a a altura do pico da curva, b a posição do centro do pico, c a largura do sino e e o número de Euler.

Dado que o algoritmo MDI é utilizado, neste trabalho, na integração dos pontos de elevação que representam as linhas de serra com a aplicação do algoritmo *diamond square*, a curva de Gauss pode ser utilizada para influenciar o formato das montanhas e serras geradas no mapa de elevação. Ou seja, o cálculo da elevação dos pontos que dão origem à elevação de um determinado ponto numa linha de serra, aplicando o algoritmo MDI, é facilmente efetuado, alimentando uma função de Gauss. Essa função é centrada no ponto da linha de serra (b igual a zero), em que a é igual à elevação desse ponto e x é igual à distância ao ponto cuja elevação se pretende obter. Na função, o valor c pode ser utilizado para influenciar a largura da montanha e a intensidade do declive.

6.2.3 Geração de Linhas de Serra

Os algoritmos de subdivisão geram mapas de elevação com reduzida complexidade computacional e com resultados satisfatórios que podem servir de base na aplicação de outras características de terreno. No entanto, é observada uma uniformidade excessiva que compromete o realismo dos mapas de elevação gerados. A definição da elevação dos cantos do mapa constitui, apenas, os valores de elevação iniciais que serão interpolados e acrescidos de valores aleatórios, progressivamente, em cada iteração. Essa particularidade dos algoritmos de subdivisão, limita a controlabilidade e reduz a formação de características bem definidas no mapa de elevação.

Num mapa de elevação obtido de modelos reais, é identificável a ocorrência de serras. As serras são conjuntos de montanhas elevadas em que os cumes vão formando uma linha com extensões que podem ter entre algumas dezenas a centenas de quilómetros. Visto de alguma distância, a diferença das elevações dos cumes das montanhas assemelha-se a uma serra, daí o nome desta formação geológica.

Como observado por Belhadj & Audibert (2005), a integração de linhas de serra num mapa de elevação gerado através de um algoritmo de subdivisão, pode melhorar significativamente o seu realismo, tendo proposto um algoritmo que permite gerar linhas de serra, baseado no movimento browniano fracionário. Para descrever o algoritmo é considerado um mapa de elevação representado por uma grelha, semelhante à utilizada no algoritmo *diamond square*, onde cada célula representa o valor de elevação. Os seguintes passos representam o procedimento para gerar linhas de serra:

1. Para cada linha de serra a gerar, são criadas duas partículas de serra. As partículas de serra são constituídas por: posição na grelha, elevação e orientação.
2. As partículas de serra são posicionadas e iniciadas aleatoriamente na mesma posição da grelha e com uma elevação igual relativamente elevada. A orientação é igualmente iniciada de forma aleatória mas oposta entre as duas partículas (Figura 23 a).
3. É iniciada uma iteração onde são aplicadas forças de impacto laterais nas partículas (Figura 23 b).
4. A orientação de cada partícula é alterada de acordo com os impactos laterais e a partícula é deslocada (Figura 23 c e d).
5. A deslocação da partícula é registada na grelha, definindo a elevação da célula onde se posiciona com a sua elevação (Figura 23 k).
6. A elevação da partícula é reduzida ao longo do seu deslocamento de acordo com uma função de Gauss centrada na posição inicial.

7. Uma nova iteração é iniciada, partindo do ponto 2), até as partículas ultrapassarem o limite da grelha, até colidirem com o percurso de uma outra partícula ou até a sua elevação aproximar-se de zero (Figura 23 de e a j).

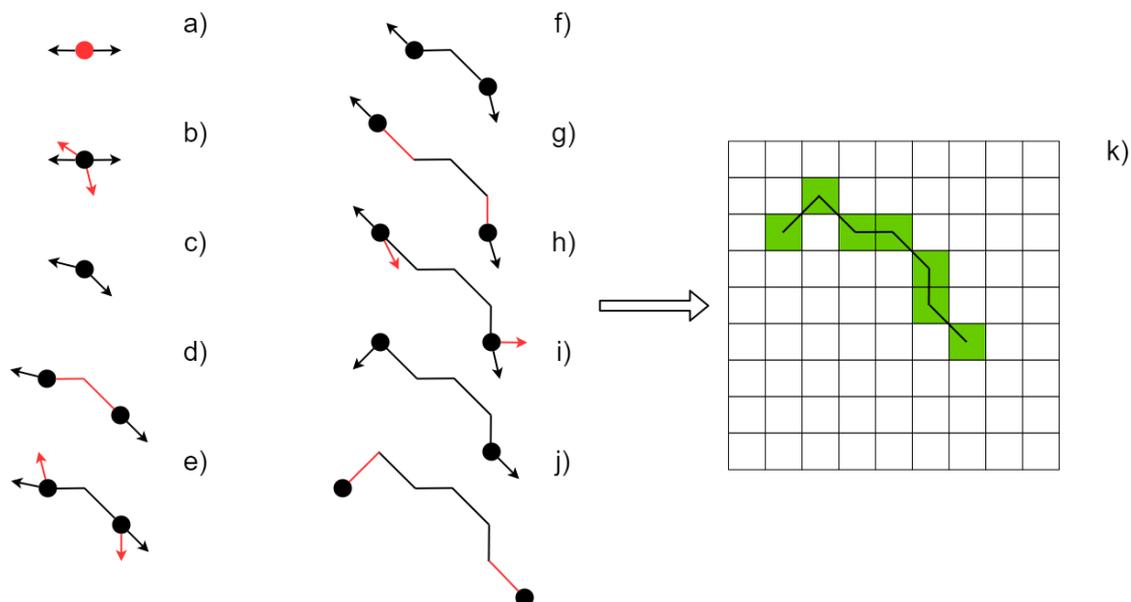


Figura 23 – Processo de geração de linhas de serra

A elevação das partículas é obtida da aplicação da função de Gauss, de forma semelhante ao procedimento descrito na secção 6.2.2, no cálculo de elevações no processo inverso ao algoritmo de subdivisão (Equação 4):

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{2c^2}} \quad (4)$$

Neste caso a função é centrada na posição inicial das partículas de serra, ou seja, b igual a zero e a igual à elevação inicial das partículas. A função é alimentada pela distância deslocada da partícula desde a posição inicial. A constante c influencia a extensão da linha de serra gerada, aumentando ou reduzindo a tendência das partículas em perder elevação durante a deslocação e constituindo um fator importante na parametrização do algoritmo.

Além da influência da constante c na função de Gauss, outros parâmetros podem influenciar a geração das linhas de serra. Um *threshold* pode determinar o intervalo de elevações que podem ser atribuídos inicialmente às partículas. De igual forma, um *threshold* de elevação mínimo pode ser usado como critério de paragem na criação da linha de serra. Uma linha de serra pode, também, ser definida manualmente pelo utilizador, caso queira atribuir uma característica específica no mapa de elevação gerado.

O algoritmo *midpoint displacement's inverse*, descrito na secção 6.2.2, é aplicado após a criação ou definição das linhas de serra. Este algoritmo permite obter as elevações que dariam origem às células que constituem as linhas de serra se o algoritmo *diamond square* fosse

aplicado. A elevação dessas células são utilizadas na aplicação do algoritmo *diamond square* de forma a obter um mapa de elevação contínuo contendo as linhas de serra.

6.2.4 Algoritmos de Erosão

Os algoritmos de subdivisão geram mapas de elevação rugosos, com variações bruscas de elevação que contrastam com o perfil erodido de modelos reais de mapas de elevação. Por influência dos fenômenos físicos, aos quais as formações geológicas estão sujeitas, tal como a chuva, neve, formação de gelo, degelo, vento e variações de temperatura dão origem à formação e movimentação de sedimentos que alteram a paisagem e em maior escala, suavizam as variações de elevação. Por esse motivo, a erosão tem sido abordada quanto à geração de mapas de elevação, existindo algumas abordagens que partem da erosão como meio de modelação de mapas de elevação.

A abordagem na qual se baseia esta implementação de erosão, parte de um mapa de elevação gerado, movendo a elevação de acordo com a simulação de dois tipos de erosão: a erosão térmica e a erosão hidráulica (Musgrave et al. 1989; Benes & Forsbach 2002). Estes algoritmos suavizam o mapa de elevação mantendo as características que o constituem e pode ser controlado quanto à sua intensidade pela modificação de constantes e número de passagens ou iterações.

6.2.5 Algoritmo de Erosão Térmica

A erosão térmica simula a deposição de material na base de inclinações. Esse material resulta da erosão e conseqüente desprendimento de sedimentos que desliza pela inclinação. A erosão é provocada pelas diferenças de temperatura resultantes da ocorrência de chuva, sol, congelamento e degelo que expandem e contraem as rochas, criando fendas e quebrando-as. A implementação efetuada deste algoritmo é inspirada nos procedimentos descritos por Olsen (2004).

Num mapa de elevação descrito numa grelha em que cada célula contém um valor de elevação, o algoritmo de erosão térmica pode ser descrito nos seguintes passos:

1. Todas as células são percorridas, iniciando uma iteração.
2. Para cada célula, a elevação é comparada com a elevação das células adjacentes.
3. Se a diferença de elevação com uma determinada célula adjacente (d_i) for superior a um *threshold* (ângulo de *talus*, T), parte da elevação da célula original é movida para a célula adjacente. As novas elevações são guardadas numa grelha temporária.
4. No fim da iteração, as elevações guardadas na grelha temporária são aplicadas definitivamente na grelha original.

5. É iniciada uma nova iteração partindo do passo 1) até completar um número de iterações.

Para cada célula da grelha, obtendo a lista de células adjacentes segundo a vizinhança de Moore (Figura 24), a diferença de elevação entre a célula atual e uma célula adjacente i , é obtida com a seguinte equação (Equação 5):

$$d_i = h - h_i \quad (5)$$

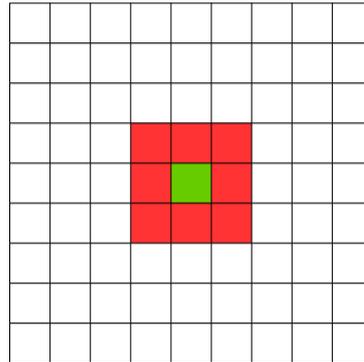


Figura 24 – Vizinhança de Moore, a vermelho as células adjacentes à célula verde

Parte da elevação de uma célula é movida para uma célula adjacente se a elevação da célula adjacente for inferior numa determinada magnitude, ou seja, se d_i for superior a T , o ângulo de talus. A elevação de uma célula adjacente, h_i pode ser obtida da seguinte forma (Equação 6):

$$h_i = \begin{cases} d_i > T: & h_i + c(d_i - T) \\ d_i \leq T: & h_i \end{cases} \quad (6)$$

O valor de T determina a diferença de elevação a partir da qual a elevação da célula adjacente deve ser modificada. Se a diferença de elevação for superior a T , parte da diferença de elevação é acrescentada à elevação da célula adjacente (Figura 25). A constante c indica a percentagem da diferença de elevação a mover.

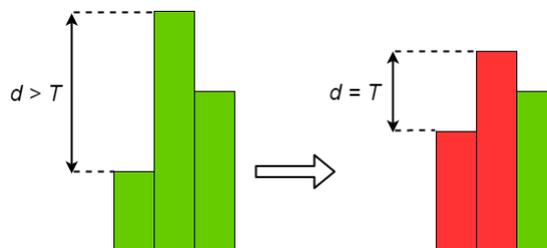


Figura 25 – Quando a diferença de elevação entre células adjacentes é maior que T , parte dessa diferença é movida.

Quando várias células adjacentes apresentam diferenças de elevação superior a T , a elevação a ser movida da célula original deve ser distribuída. A elevação, h_i , para cada uma dessas células é obtida pela seguinte equação (Equação 7):

$$h_i = h_i + c(d_{max} - T) \times \frac{d_i}{d_{total}} \quad (7)$$

Onde d_{max} é o maior d_i e d_{total} o somatório de todos os d_i maiores que T .

As elevações calculadas são mantidas numa grelha temporária durante a iteração para que as elevações previamente alteradas ao percorrer as células do mapa de elevação não influenciem as restantes células. Desta forma a erosão é simulada, numa passagem, como se fosse aplicada ao mesmo tempo em todas as células.

Este algoritmo de erosão pode ser controlado quanto à sua intensidade, indicando o número de passagens, ou iterações, que são efetuadas sobre um mapa de elevação. O valor de T e o valor de c permitem controlar o comportamento do algoritmo, aumentando ou reduzindo o limite de inclinação para a movimentação de sedimentos ou determinando a quantidade de material a ser movido. Uma possível abordagem é a influência dos valores de T e de c de acordo com um esboço, de forma a indicar o tipo de erosão que acontece em determinadas zona do mapa de elevação, simulando diferentes climas e tipo de solo.

6.2.6 Algoritmo de Erosão Hidráulica

A erosão hidráulica simula a movimentação de sedimentos por ação da água. Parte do solo é diluído na água, que flui ao longo do terreno até zonas de menor elevação, depositando os sedimentos. A diluição e deposição de sedimentos depende do nível de saturação da água. À medida que os sedimentos são diluídos a água satura. Por efeito da evaporação, o nível de saturação da água também aumenta, até atingir o limite de saturação e originar a deposição dos sedimentos excedentes. Tal como o algoritmo de erosão térmica, a seguinte implementação é baseada no procedimento descrito por Olsen (2004).

Assumindo uma grelha em que cada célula contém, além do valor de elevação do solo, a elevação de água e o volume de sedimentos, o algoritmo pode ser descrito nos seguintes passos:

1. É iniciada uma iteração de erosão, percorrendo todas as células e introduzido um valor de água de forma a simular a queda de chuva.
2. São percorridas, novamente, todas as células convertendo parte da elevação em sedimento de acordo com a elevação de água existente, simulando a diluição do solo na água.
3. É iniciado o transporte de água. Para cada célula, a elevação de água é movida para as células adjacentes com elevação total (elevação do solo mais elevação da água) inferior de forma a nivelar as elevações totais. Os novos valores são guardados

numa grelha temporária e aplicados no fim deste passo, para não influenciar células que ainda não tenham sido calculadas.

4. Os sedimentos são movidos em quantidade proporcional à elevação de água movida.
5. Percorrendo todas as células, a elevação da água é parcialmente removida, simulando a evaporação.
6. É calculado o valor da saturação da água tendo em conta a elevação da água e a quantidade de sedimentos para determinar o volume de sedimentos diluídos na água que deve ser convertido para elevação do solo.
7. Uma nova passagem do algoritmo é iniciada partindo do passo 1) até completar um número de iterações.

A quantidade de água a acrescentar em cada célula é regulada pela constante K_r , que representa a quantidade de água da chuva a ser introduzida por cada célula em cada passagem do algoritmo. A elevação de água w de uma determinada célula é calculada da seguinte forma (Equação 8):

$$w = w + K_r \quad (8)$$

Os sedimentos que são diluídos na água contida na célula são retirados da elevação do solo. Desta forma o valor de sedimentos a ser acrescentado na célula é removido em igual proporção da elevação. Este cálculo é efetuado tendo em conta a constante K_s , o coeficiente de solubilidade do solo, e a elevação de água w . Desta forma, a quantidade de sedimentos Δs a retirar da elevação, bem como o novo valor de elevação h e o novo valor de sedimentos s da célula são calculados da seguinte forma (Equações 9, 10 e 11):

$$\Delta s = K_s \times w \quad (9)$$

$$h = h - \Delta s \quad (10)$$

$$s = s + \Delta s \quad (11)$$

O deslocamento da água é simulado, distribuindo a elevação de água contida na célula atual para as células adjacentes com menor elevação total a . As células adjacentes são obtidas segundo a vizinhança de Moore (Figura 24 – Vizinhança de Moore, a vermelho as células adjacentes à célula verde) e o valor da elevação total é obtido pela soma da elevação do solo com a elevação da água para cada célula (Equação 12):

$$a = h + w \quad (12)$$

A quantidade de água distribuída para cada célula adjacente com menor elevação total (a) é calculada de forma a equalizar as elevações totais com a elevação total da célula atual. Desta

forma, a quantidade de água a ser movida Δw_i para uma célula adjacente i com menor elevação total é efetuada pelas seguintes equações (Equações 13, 14 e 15):

$$\Delta a = a - \bar{a} \quad (13)$$

$$d_i = a - a_i \quad (14)$$

$$\Delta w_i = \begin{cases} \Delta a < w: & \Delta a \times \frac{d_i}{d_{total}} \\ \Delta a \geq w: & w \times \frac{d_i}{d_{total}} \end{cases} \quad (15)$$

Onde \bar{a} é a média das elevações totais a e d_{total} é a soma das diferenças de elevação total d_i das células adjacentes envolvidas no cálculo.

A quantidade de sedimentos a ser movida Δs_i para cada uma das células adjacentes i com menor elevação total é obtida em proporção à água transportada (Equação 16):

$$\Delta s_i = s \times \frac{\Delta w_i}{w} \quad (16)$$

A água removida a cada célula por evaporação é simulada considerando uma constante de evaporação K_e . Desta forma, a elevação de água final w pode ser obtida pela equação (Equação 17):

$$w = w \times (1 - K_e) \quad (17)$$

Obtida a elevação da água final para uma passagem do algoritmo de erosão, é necessário verificar se a quantidade de sedimentos presentes em cada célula excede o limite de saturação da água s_{max} . Esse valor é calculado tendo em conta o coeficiente de capacidade de sedimentos da água K_c . Se o limite de saturação for excedido, o excedente de sedimentos Δs é acrescentado à elevação h da célula e é removido do volume de sedimentos s (Equações 18, 19, 20 e 21):

$$s_{max} = K_c \times w \quad (18)$$

$$\Delta s = \begin{cases} s > s_{max}: & s - s_{max} \\ s \leq s_{max}: & 0 \end{cases} \quad (19)$$

$$s = s - \Delta s \quad (20)$$

$$h = h + \Delta s \quad (21)$$

O número de passagens do algoritmo de erosão permite controlar a intensidade das alterações efetuadas no mapa de elevação. Tal como o algoritmo de erosão térmica, este algoritmo pode tirar partido da parametrização através de esboços para influenciar as constantes envolvidas nos cálculos K_r , K_s e K_e , simulando respetivamente, a pluviosidade, o tipo de solo e a temperatura de diferentes regiões do mapa de elevação.

6.2.7 Algoritmo de Colonização Espacial

Apresentado por Runions et al. (2007), o algoritmo de colonização espacial permite a geração de modelos 3D do tronco e ramos de árvores e arbustos com resultados realistas. Apesar de não simular interações endógenas e exógenas, nem abordar outras variedades de vegetação, este algoritmo permite, em grande escala, obter resultados satisfatórios. Outra característica importante deste algoritmo é o potencial de parametrização de forma a influenciar os modelos gerados.

O algoritmo pode ser explicado de forma simplificada nos seguintes passos:

1. É distribuído um número de pontos (pontos de atração) dentro de uma área delimitada no espaço tridimensional. Essa área determina o tamanho e o formato da árvore ou arbusto pretendido (os pontos verdes no esquema *a* da Figura 26 – Algoritmo de colonização espacial).
2. Assumindo um ponto, representando a base da árvore, e uma direção, é criado um ramo inicial (o ponto vermelho no esquema *a* da Figura 26). Um ramo é representado pela origem (um ponto no espaço tridimensional), um vetor direção e a referência para um ramo antecessor. O primeiro ramo é a origem da árvore e, por essa razão, não contém antecessor.
3. É iniciado o processo iterativo do algoritmo de colonização espacial, percorrendo todos os pontos de atração.
4. Se existirem ramos com a origem a uma distância ao ponto de atração inferior à distância de influência d_i (Figura 26 esquema *b*), a direção do ramo com a menor distância é alterada na direção do ponto de atração.
5. Se existirem ramos com a origem a uma distância ao ponto de atração inferior à distância de exclusão d_k , o ponto de atração é removido (Figura 26 esquema *c*).
6. Para cada ramo influenciado por pontos de atração, é criado um novo ramo que o refere como antecessor e herda a sua direção. O ponto de origem do novo ramo é obtido da deslocação do ponto de origem na direção herdada do antecessor.
7. Uma nova iteração é iniciada, partindo do passo 3. até todos os pontos de atração serem removidos (Figura 26 esquemas *d*, *e*, *f* e *g*).
8. No fim, a estrutura da árvore é obtida da ligação dos pontos de origem dos ramos com o ponto de origem do ramo antecessor (Figura 26 esquema *h*).

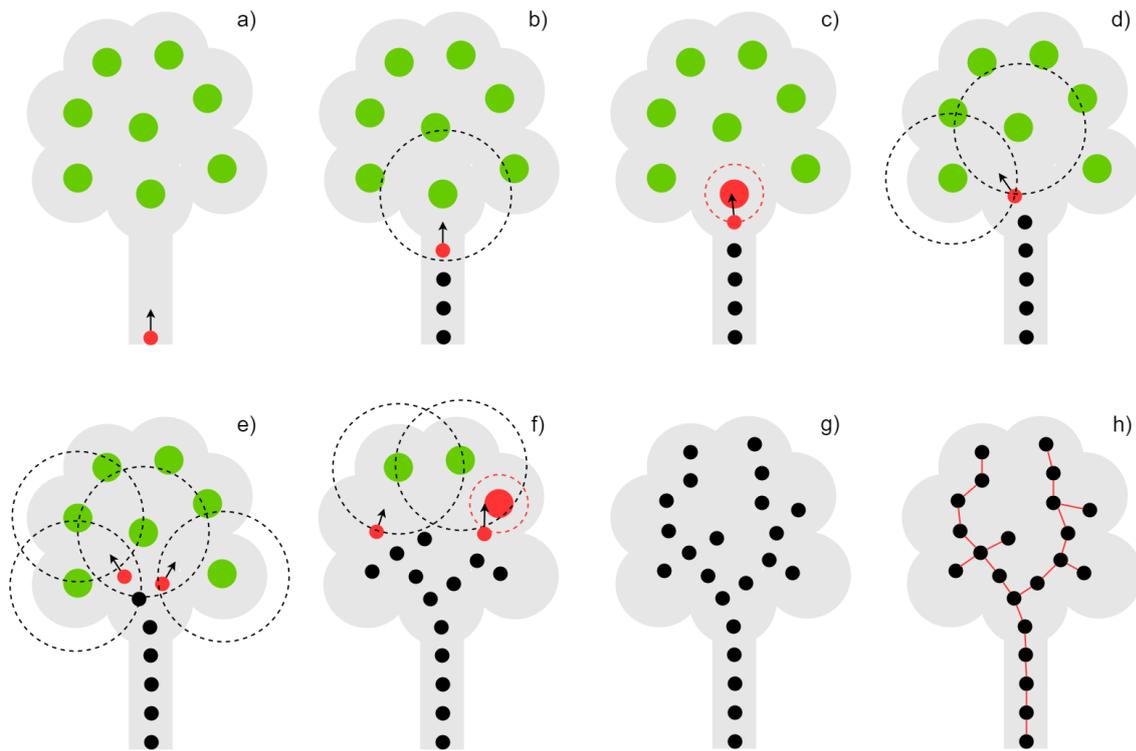


Figura 26 – Algoritmo de colonização espacial

A direção associada a cada ramo é representada por um vetor normalizado. No ramo inicial, a origem é arbitrária, sendo a direção definida tendo em conta a localização dos pontos de atração. No início do algoritmo, se nenhum ramo estiver no alcance da distância de influência de um ponto de atração, ramos são gerados sucessivamente, na direção inicial até se aproximarem de um ponto de atração, para que o processo iterativo principal possa ser iniciado.

Quando um ponto de atração encontra vários ramos com o ponto de origem dentro da distância de influência d_i , é considerado o ramo mais próximo, e é modificada a sua direção \vec{r}_d da seguinte forma (Equação 22):

$$\vec{r}_d = \|\vec{r}_d + (P - R)\| \quad (22)$$

Sendo P e R , respetivamente, a posição do ponto de atração e a origem do ramo. Desta forma, os ramos influenciados mantêm o vetor normalizado para que as deslocações sejam calculadas uniformemente.

Em cada iteração, sempre que um ramo é influenciado, é gerado um novo ramo com as mesmas características, que o sucede. Sempre que um novo ramo é criado, o ponto de origem é deslocado de acordo com a sua direção \vec{r}_d . Esta deslocação é influenciada por uma constante d_m que controla a sua magnitude. A nova posição da origem R de um ramo pode ser obtida segundo a Equação 23:

$$R = R + \vec{r}_d \times d_m \quad (23)$$

Este algoritmo pode ser influenciado através das constantes, do número de pontos de atração gerados, da posição e direção do ramo inicial e da definição dos limites onde os pontos de atração são distribuídos. As constantes d_k , d_i e d_m e o número de pontos de atração influenciam a forma e densidade dos ramos gerados. Para maiores valores de d_k e reduzido número de pontos de atração, os ramos gerados são mais dispersos e em menor quantidade. Quanto à constante d_i , quanto menor o seu valor, mais sinuosos e irregulares são os ramos gerados. A constante d_m aumenta o detalhe das ramificações quanto menor for o seu valor, já que permite controlar o número de secções que descrevem os ramos. A área onde os pontos de atração são distribuídos, permite definir o formato da árvore ou arbustos gerados. Com a combinação destes parâmetros, este algoritmo permite gerar e simular as ramificações de um vasto conjunto de espécies reais.

6.2.8 Modelação de Mapas de Elevação

A técnica de modelação procedimental de mapas de elevação implementada resulta da combinação de vários algoritmos descritos nas secções anteriores. O mapa de elevação resultante é representado por uma grelha onde cada célula contém um valor de elevação.

Como parâmetros gerais, que influenciam os diferentes algoritmos utilizados nesta técnica, são indicados valores que definem a dimensão do mapa de elevação a gerar. É indicado o número de células da grelha na razão $(2^n + 1)^2$ e o comprimento do lado de cada célula, em metros. A área geográfica representada pelo mapa de elevação gerado, em metros, é obtida da seguinte forma (Equação 24):

$$g = (\sqrt{d_g} \times d_c)^2 \quad (24)$$

Sendo d_g o número de células da grelha e d_c o lado de cada célula, em metros. Um outro parâmetro geral definido é um valor da elevação máxima possível, em metros. É também indicado um número inteiro que serve de semente para o gerador de números aleatórios, para que um mesmo mapa de elevação possa voltar a ser gerado com os mesmos parâmetros, ou para gerar outras variantes modificando esse mesmo valor.

O procedimento para modelar mapas de elevação é iniciado pela geração de linhas de serra usando o algoritmo descrito na secção 6.2.3. A geração de linhas de serra é controlada por um parâmetro que indica o número de partículas de serra e indiretamente pela parametrização da dimensão do mapa de elevação. O número de partículas de serra determina o número de linhas de serra geradas. A elevação da partícula influencia a constante c no cálculo da função de Gauss da seguinte forma (Equação 25):

$$c = (R \times h_p) \div d_c \quad (25)$$

Sendo R uma constante de decréscimo da elevação, que assume o valor 15 nesta implementação. O valor de h_p é a elevação inicial da partícula de serra. A constante c aumenta

ou reduz a largura do gráfico da curva de Gauss, o que por sua vez altera a tendência da linha de serra em perder elevação. Desta forma, num mapa de menor dimensão geográfica, a linha de serra tem uma redução da elevação menor à medida que a linha de serra é traçada. Já num mapa de maior dimensão, a linha de serra tende a perder maior elevação. Este comportamento pretende adaptar o traçado de uma linha de serra à dimensão geográfica do mapa de elevação, evitando declives excessivos num mapa de elevação de menor dimensão geográfica.

O algoritmo de geração de linhas de serra pode ser completamente desativado e substituído por um esboço de linhas de serra parametrizado pelo utilizador.

Este editor de esboços é configurado com duas ferramentas. A ferramenta de linha de serra, que desenha uma linha onde a elevação é indicada entre a cor preta e a cor branca, sendo a cor preta sem elevação e a cor branca elevação máxima. A ferramenta de linha de serra é sensível à velocidade, sendo maior a elevação representada pelo traçado da linha quanto maior for a velocidade com que o utilizador a desenha. A ferramenta para apagar a linha de serra, permite corrigir e eliminar traçados previamente definidos.

Após a geração de linhas de serra e antes da execução do algoritmo de subdivisão, é necessário obter os valores de elevação das células da grelha que dariam origem à elevação das células já calculadas. Esta abordagem permite resolver o problema da continuidade das elevações das linhas de serra com as elevações obtidas do método de subdivisão. Para obter essas elevações é utilizado o algoritmo *midpoint displacement's inverse*, tal como descrito na secção 6.2.2. Este algoritmo não recebe qualquer parametrização, no entanto, tal como o algoritmo para a geração de linhas de serra, a função de Gauss utilizada neste algoritmo é indiretamente influenciada pela dimensão geográfica do mapa de elevação. Neste caso, a constante R assume um valor superior, 150, já que a queda de elevações das células perpendiculares às linhas de serra tem maior magnitude.

Resolvido o problema da continuidade das elevações das linhas de serra, é executado o algoritmo *diamond square*, tal como descrito na secção 6.2.1, obtendo as elevações restantes do mapa de elevação. A parametrização deste algoritmo é efetuada recorrendo a um esboço de rugosidade do terreno.

Este editor de esboço é configurado com uma cor base que define a rugosidade intermédia e com cinco ferramentas que permitem alterar a rugosidade em cinco níveis. Quatro dessas ferramentas permitem desenhar áreas com rugosidade diferente, duas ferramentas para rugosidade superior à rugosidade intermédia e duas com rugosidade inferior. As cores usadas, da rugosidade superior para a inferior são, respetivamente: vermelho, amarelo, verde, azul-escuro e azul claro. Sendo a cor verde, rugosidade intermédia, a cor base do esboço. A quinta ferramenta permite apagar áreas de rugosidade alterada. Este esboço influencia os desfasamentos aleatórios que são adicionados no cálculo da elevação de uma nova célula em cada subdivisão. Quanto maior a rugosidade, maior o intervalo de valores aleatórios possíveis.

Por fim, para suavizar e melhorar o realismo do mapa de elevação gerado, são aplicados os algoritmos de erosão térmica e hidráulica, tal como apresentado nas secções 6.2.5 e 6.2.6. Cada um desses algoritmos são parametrizados pela indicação do número de iterações e por esboços.

No algoritmo de erosão térmica, o utilizador cria um esboço usando um editor com duas ferramentas de edição, uma para indicar maior volume de sedimentos que se desprendem por efeito da erosão e outra para indicar menor volume. Outra ferramenta permite remover as áreas editadas, mantendo a cor base do esboço que indica o valor intermédio do volume de sedimentos que se desprendem. As cores no esboço influenciam diretamente o valor da constante c da função que calcula o volume de sedimentos (Equação 7) que é movido para as células adjacentes com declive maior que o ângulo de talus (T).

No algoritmo de erosão hidráulica, o esboço permite indicar regiões com três tipos de clima. O clima moderado é indicado pela cor base do esboço e duas ferramentas de esboço permitem indicar zonas frias com maior pluviosidade ou zonas quentes e secas. Este esboço permite influenciar as constantes K_r (quantidade de água introduzida em cada iteração) e K_e (constante de evaporação) que têm impacto nas equações de cálculo de água introduzida em cada célula e a quantidade de água que é reduzida por evaporação.

Os principais meios para a visualização do mapa de elevação resultante é uma imagem mostrando as elevações em escala de cinza e o modelo 3D do terreno formado pelo mapa de elevação.

A imagem da representação do mapa de elevação tem um número de *pixels* igual ao número de células. Cada valor de elevação (valores entre zero e um) é convertido num *pixel* da imagem entre a cor preta e a cor branca, representando a cor preta a elevação zero e a cor branca a elevação máxima.

O modelo 3D do terreno é obtido pelo percurso da grelha, criando um vértice por cada célula. As faces são obtidas percorrendo novamente cada célula da grelha e formando triângulos com os vértices gerados. São aplicadas três texturas diferentes, relva, rocha e neve. A textura da relva é aplicada em zonas pouco inclinadas e de baixa elevação, enquanto a textura da neve é aplicada em zonas com maior elevação. Em zonas com maior declive independentemente da elevação é aplicada a textura de rocha.

6.2.9 Modelação de Vegetação

A modelação de árvores e arbustos utiliza o algoritmo de colonização espacial, explicado na secção 6.2.7. Neste algoritmo, foram identificadas várias constantes que têm impacto nas ramificações resultantes e que podem ser parametrizados. Desta forma, são parametrizados o número de pontos de atração gerados e as constantes d_k , d_i e d_m representando, respetivamente, a distância de remoção, a distância de atração e o tamanho do segmento do ramo.

Um aspecto importante na parametrização do algoritmo de colonização espacial é a definição da área onde os pontos de atração são distribuídos. A distribuição dos pontos de atração é o fator com maior peso na definição do formato da árvore ou arbusto a modelar. Por esse motivo a área da distribuição dos pontos de atração é definida pelo utilizador através de um esboço. Esse editor de esboço é constituído por duas ferramentas, um pincel que permite desenhar um perfil da área onde os pontos são distribuídos e uma ferramenta que permite apagar as áreas definidas para efetuar correções.

O perfil definido no esboço permite obter a área onde são distribuídos os pontos de atração. Assumindo o esboço como um conjunto de *pixels* num sistema de coordenadas XY, a área é obtida dividindo o esboço por linhas onde cada linha tem o mesmo valor da coordenada y. Por cada linha é identificado o *pixel* definido com o menor valor de x e o *pixel* definido com o maior valor de x. Esses pontos em cada linha são usados para criar um círculo tridimensional com diâmetro igual ao módulo da diferença entre o menor e o maior valor de x, centrado no valor de x intermédio. Esse círculo representa uma secção da área obtida do esboço onde são distribuídos os pontos de atração (Figura 27).

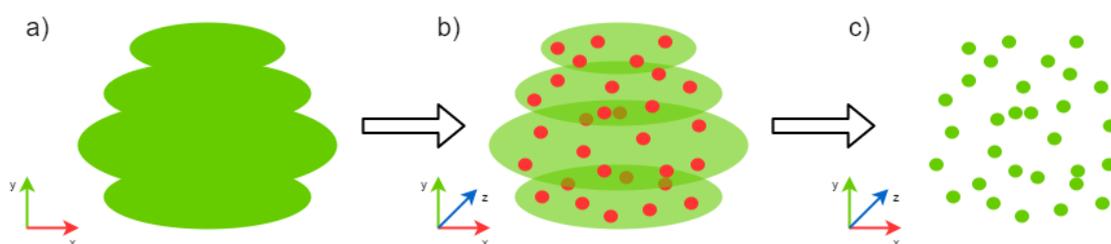


Figura 27 – Distribuição dos pontos de atração

Sendo a área de distribuição definida por secções em forma de círculo, os pontos são distribuídos aleatoriamente nas secções. A distribuição é efetuada tendo em conta a área dos círculos, ou seja, o número de pontos a distribuir por cada secção é proporcional com área delimitada pelos círculos. Dentro de cada círculo, as coordenadas dos pontos de atração a distribuir é obtida pela equação cartesiana do círculo, gerando aleatoriamente por cada ponto o valor de um ângulo entre 0 e 2π e o valor de um raio entre 0 e o raio do círculo. Para evitar uma maior distribuição de pontos perto do centro do círculo, é criada uma tabela *lookup* com valores de raio. Valores de raio iguais são colocados na tabela *lookup* em quantidade proporcional ao valor do perímetro desse raio no círculo. Para obter um valor do raio aleatório da circunferência para calcular as coordenadas de um ponto, é gerado um valor inteiro aleatório entre 0 e o número de elementos da tabela *lookup*.

Os principais meios de visualização de resultados são a visualização do modelo 3D das árvores ou arbustos gerados e do modelo 3D do terreno formado pelo mapa de elevação combinado com a variedade de vegetação modelada.

O modelo 3D da árvore é obtido, inicialmente, pelo modelo da ramificação da árvore. Cada setor de ramo é formado por um cilindro, que tem um diâmetro mínimo nas terminações dos ramos e tem o maior diâmetro perto da raiz. Por questões de performance, na implementação efetuada, o cilindro foi substituído por prismas triangulares, reduzindo a complexidade do modelo gerado sem comprometer significativamente o resultado. Uma textura de madeira é aplicada nas faces dos prismas que constituem as secções dos ramos. As ramificações resultantes têm um nível de detalhe e realismo consideráveis, no entanto, o resultado pode ser melhorado pela modelação de folhagem.

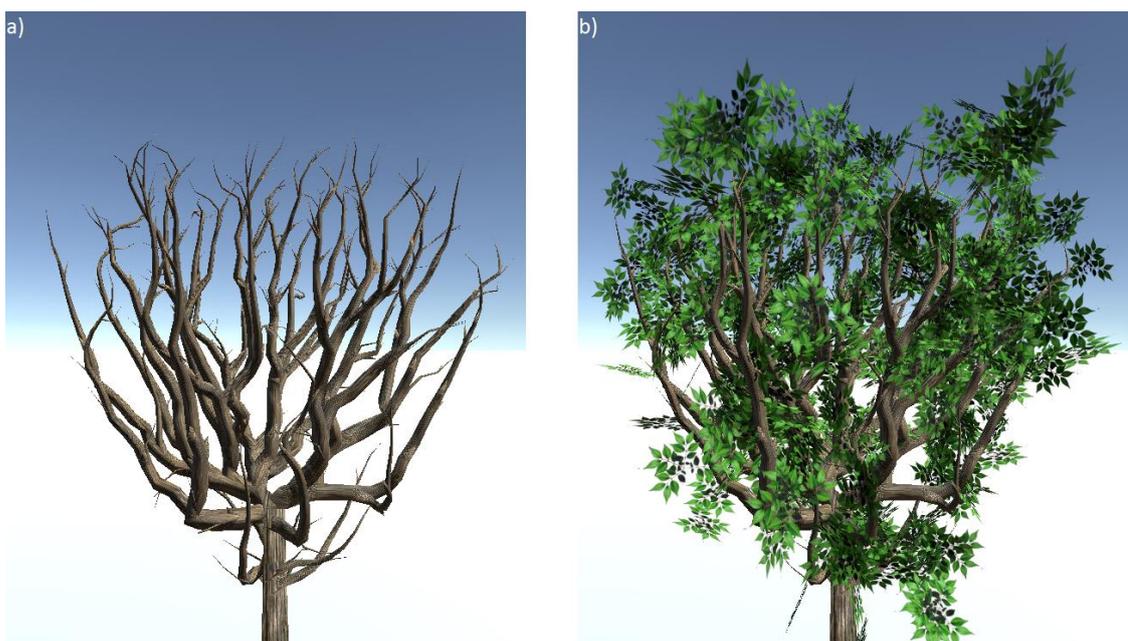


Figura 28 – Geração do modelo 3D de uma árvore

A modelação de folhagem é obtida pela distribuição de faces de dimensão parametrizável contendo uma textura de folhas (Figura 28). Essas faces são distribuídas nas terminações dos ramos, aleatoriamente, dentro de uma distância parametrizada à terminação dos ramos, sendo a orientação das faces, também atribuída aleatoriamente. Esta abordagem permite salientar o formato da copa da árvore ou arbusto e gerar árvores com uma densidade de folhas elevada, com um reduzido número de polígonos.

O modelo 3D do terreno (Figura 29) resulta da distribuição das árvores e arbustos pelo modelo de terreno gerado a partir do mapa de elevação. As árvores e arbustos são distribuídos aleatoriamente obedecendo a algumas regras. Toda a vegetação tem limitação na sua distribuição quanto à inclinação do terreno, ou seja, as árvores e arbustos não podem ser colocadas em inclinações elevadas. Essa inclinação limite é parametrizada, tendo a vegetação maior tolerância à inclinação quanto menor for o seu porte. Esta tolerância à inclinação simula o facto da vegetação de menor dimensão poder surgir em locais de maior inclinação, dado o seu menor peso. A vegetação de menor porte tem menor probabilidade de ser distribuída perto

de vegetação de maior porte e vice-versa, simulando a concorrência da vegetação por espaço e luz solar.

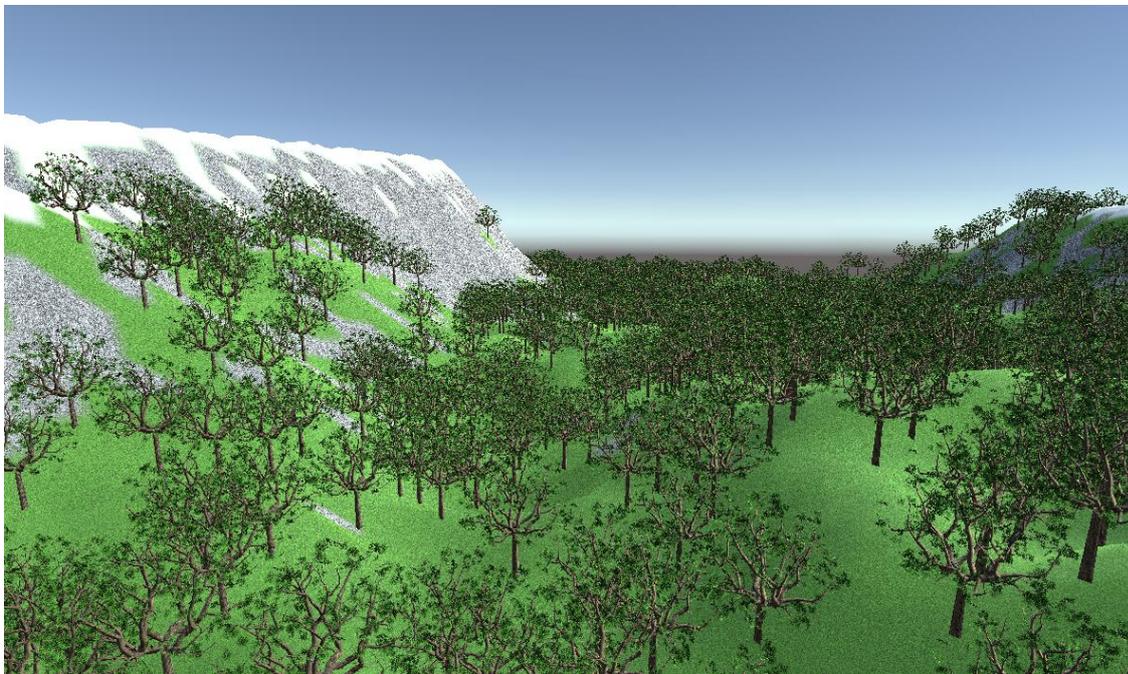


Figura 29 – Terreno resultante da combinação do mapa de elevação com a vegetação

6.2.10 Integração com a *Framework*

As técnicas de modelação foram implementadas de forma a constituir dois *plugins* de modelação procedimental, um *plugin* de modelação procedimental de mapas de elevação e um *plugin* de modelação de vegetação. Foram também usadas ferramentas de parametrização e visualização disponibilizadas pela *framework*, tal como descrito nos detalhes de implementação nas secções 6.2.8 e 6.2.9, e implementada a interface de comunicação para permitir a integração.

O *plugin* de modelação de mapas de elevação não requer quaisquer dados de entrada do *workflow* de modelação procedimental e não permite a existência de dados já processados, por esse motivo, este *plugin* só pode ser configurado como primeira camada no *workflow* de modelação. Esta restrição prende-se com o facto da técnica de modelação procedimental implementada não necessitar e não considerar outras características de terreno existentes, evitando a perda dos dados gerados nessas camadas já processadas ou gerar o modelo de um terreno incoerente, contendo características sobrepostas. Os dados de saída deste *plugin*, tal como é imposto pela camada de modelação que modela, são a grelha de elevações e respetivos parâmetros (elevação máxima e mínima e distância das laterais em metros) e o modelo 3D do terreno formado pelo mapa de elevação.

Quanto ao *plugin* de modelação de vegetação, é obrigatória a modelação do mapa de elevação e não é permitida a modelação prévia da rede de estradas e centros urbanos. Este conjunto de restrições permite que, opcionalmente, seja configurada a modelação de rios, oceanos e lagos, numa camada anterior do *workflow*, já que a técnica de modelação procedimental implementada pode considerar o posicionamento da água na distribuição da vegetação. Já a existência de um mapa de elevação é obrigatória porque a distribuição da vegetação depende das elevações do terreno e requer a definição da dimensão do terreno obtida dessa camada do *workflow*. Implementando a camada de vegetação do *workflow* de modelação, os dados impostos que devem ser acrescentados à estrutura comum do *workflow* são os modelos 3D das árvores e arbustos gerados, a lista de posições desses modelos no terreno e o modelo 3D do terreno contendo a vegetação.

6.3 Resumo do Capítulo

Neste capítulo foi apresentada a *framework* de modelação procedimental implementada bem como o seu funcionamento. Foi também descrita a implementação dos algoritmos de modelação procedimental usados para a demonstração da *framework* desenvolvida.

Foram demonstradas as diferentes vistas da interface para o utilizador que permitem a gestão dos projetos de terreno, a configuração do *workflow* de modelação, a edição de parâmetros através das ferramentas de edição e visualização e o controlo do processo de modelação. A modelação de um terreno é tratada como um projeto onde o *workflow* de modelação é configurado pela organização de *plugins* que modelam uma característica ou um conjunto de características de terreno. Os dados e modelos resultantes, bem como os parâmetros alterados, são guardados juntamente com o projeto para que o terreno modelado possa ser visualizado novamente e o processo de modelação alterado e reiniciado. A configuração do *workflow* é auxiliada por ícones e códigos de cores que ajudam a identificar quais as características ou conjunto de características de terreno de um *plugin* e a identificar as restrições que influenciam a sua organização.

Quanto aos algoritmos de modelação procedimental, foram implementadas duas características de terreno, mapas de elevação e vegetação. A modelação de mapas de elevação foi implementada com base num algoritmo de subdivisão associado a algoritmos para gerar linhas de serra e simular erosão. A modelação de vegetação recorreu à utilização de um algoritmo de colonização espacial. Para cada algoritmo e *plugins* de modelação implementados, foram descritos os meios de parametrização possíveis e quais os seus efeitos no modelo gerado. Por fim foram descritas as ferramentas de parametrização utilizadas.

7 Avaliação de Resultados

No capítulo 6 foi descrita a implementação da *framework* de modelação procedimental. A *framework* desenvolvida apresenta várias inovações que permitem facilitar a integração e parametrização de técnicas de modelação procedimental. Como exemplo, duas técnicas de modelação procedimental foram implementadas e integradas utilizando a *framework* desenvolvida. A implementação das técnicas de modelação procedimental teve um foco significativo em criar meios de parametrização que utilizassem as ferramentas de parametrização. Neste capítulo é efetuada a análise e demonstração dos diferentes resultados que se podem obter pela utilização das ferramentas desenvolvidas.

7.1 Resultados da Parametrização dos Mapas de Elevação

Os mapas de elevação são parametrizados pela alteração de variáveis e pela modificação de esboços. Um valor parametrizável que tem um impacto significativo na irregularidade do terreno é o número de linhas de serra geradas automaticamente. Na Figura 30, Figura 31 e Figura 32, é visível o acréscimo da densidade de montanhas à medida que o número de linhas de serra geradas é aumentado, apresentando zero, uma e cinco linhas de serra, respetivamente.

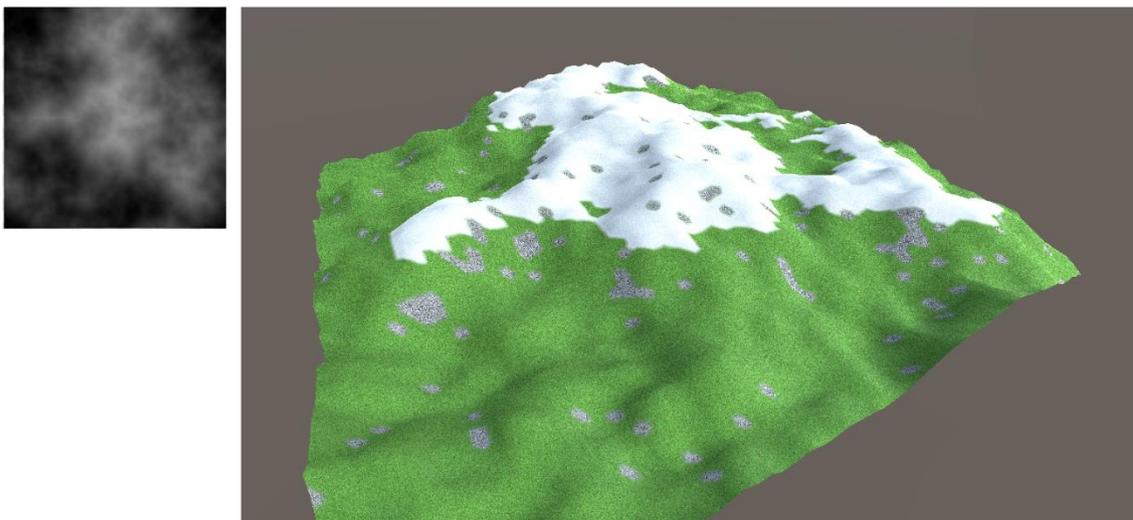


Figura 30 – Mapa de elevação resultante de zero linhas de serra

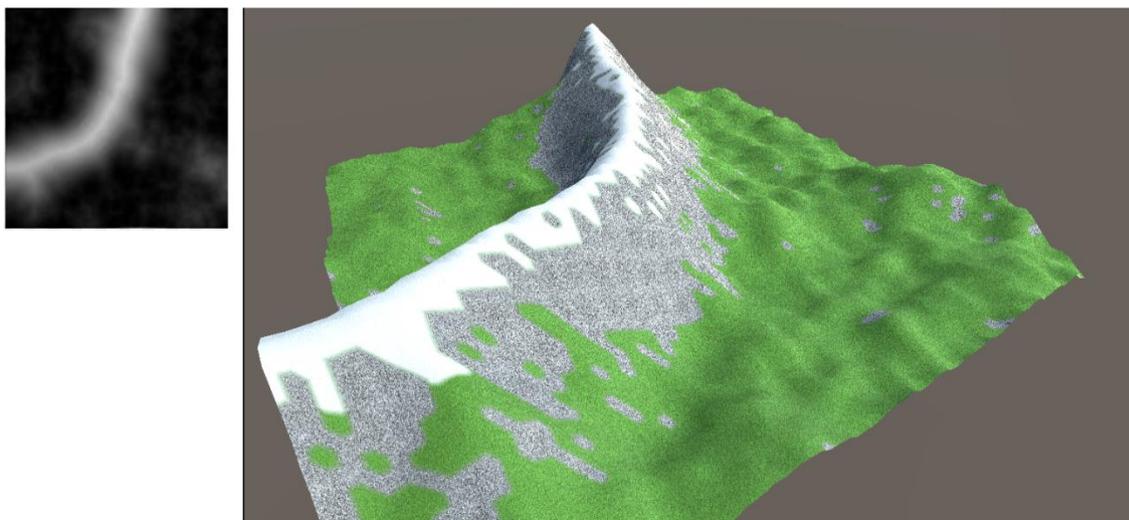


Figura 31 – Mapa de elevação resultante de uma linha de serra

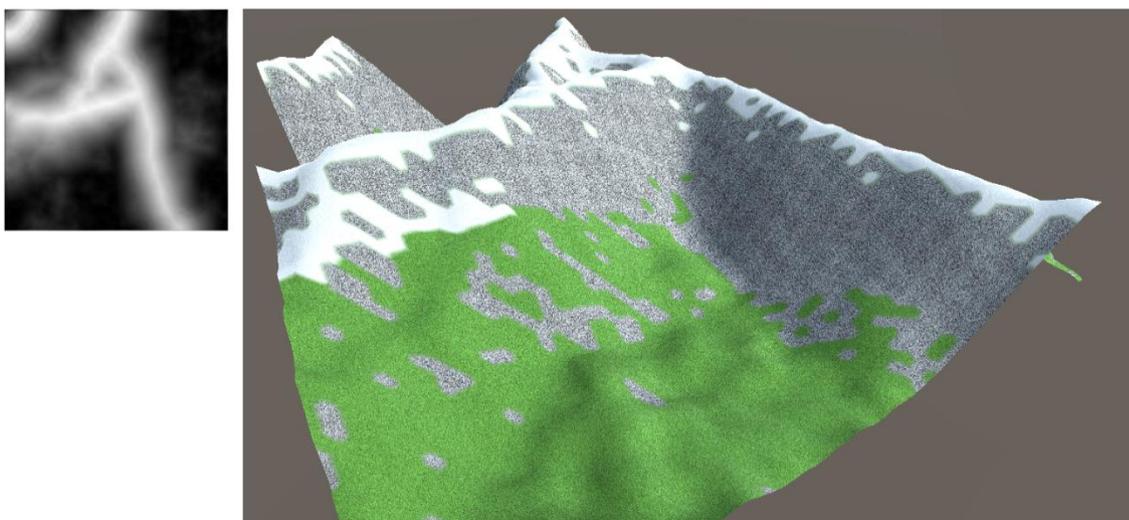


Figura 32 – Mapa de elevação resultante de cinco linhas de serra

Uma outra observação efetuada da modificação do número de linhas de serra é a de que, para um elevado número de linhas de serra, o mapa de elevação aparenta ter uma elevação média superior. Para demonstrar este comportamento, foram recolhidas 100 amostras da média da elevação de mapas de elevação aleatórios gerados para zero, uma, cinco e dez linhas de serra. A cada dez amostras, foi calculada a média dos valores obtidos. Estes valores são apresentados na Tabela 42:

Tabela 42 – Valores médios de elevação de mapas de elevação com diferentes números de linha de serra (em metros)

	0 Linhas de Serra	1 Linha de Serra	5 Linhas de Serra	10 Linhas de Serra
10	429,7744	350,8212	718,0368	938,506
Amostras				
20	444,2095	363,0317	737,6362	956,0749
Amostras				
30	450,649	381,2847	742,7725	954,5202
Amostras				
40	450,8926	383,4416	738,8398	961,8601
Amostras				
50	453,5907	391,6071	738,8323	955,305
Amostras				
60	457,1824	390,1007	739,6608	954,386
Amostras				
70	457,4506	393,8293	748,9785	957,1749
Amostras				
80	464,5875	393,8319	747,6964	952,3099
Amostras				
90	452,726	389,2541	747,218	947,8563
Amostras				
100	459,2506	390,003	744,1483	955,2238
Amostras				

Dos valores da tabela, pode ser concluído que a média das elevações tende a ser maior quanto maior for o número de linhas de serra geradas. Este facto está relacionado com o posicionamento de partículas de serra em elevações relativamente elevadas e que vão perdendo elevação segundo uma função de Gauss à medida que o traçado das linhas de serra são descritos. A aplicação do algoritmo MDI é também influenciado por uma função de Gauss na obtenção das elevações no processo inverso. Quando um elevado número de linhas de serra é gerado, a proximidade das linhas de serra não permite que as elevações tenham uma redução significativa, já que os sopés das montanhas se apresentam a uma distância relativamente próxima dos respetivos cumes. Por outro lado, quando o mapa de elevação apresenta um reduzido número de linhas de serra, a elevação das células tem uma rápida redução quanto mais afastadas estiverem das linhas de serra. Quando não existem linhas de serra, o algoritmo MDI não é aplicado e o terreno tende a apresentar elevações uniformemente distribuídas. Por essa razão, os mapas de elevação sem linhas de serra tendem a apresentar uma elevação média superior a mapas de elevação com uma linha de serra.

Um outro método para gerar linhas de serra é a edição manual através de um esboço. Na Figura 33, Figura 34 e Figura 35 são mostrados os esboços de linhas de serra com contornos artificiais (no quadrado inferior esquerdo) e os respetivos mapas de elevação gerados. Estes

mapas de elevação permitem demonstrar a eficácia do algoritmo MDI na junção de características definidas pelo utilizador no modelo gerado.

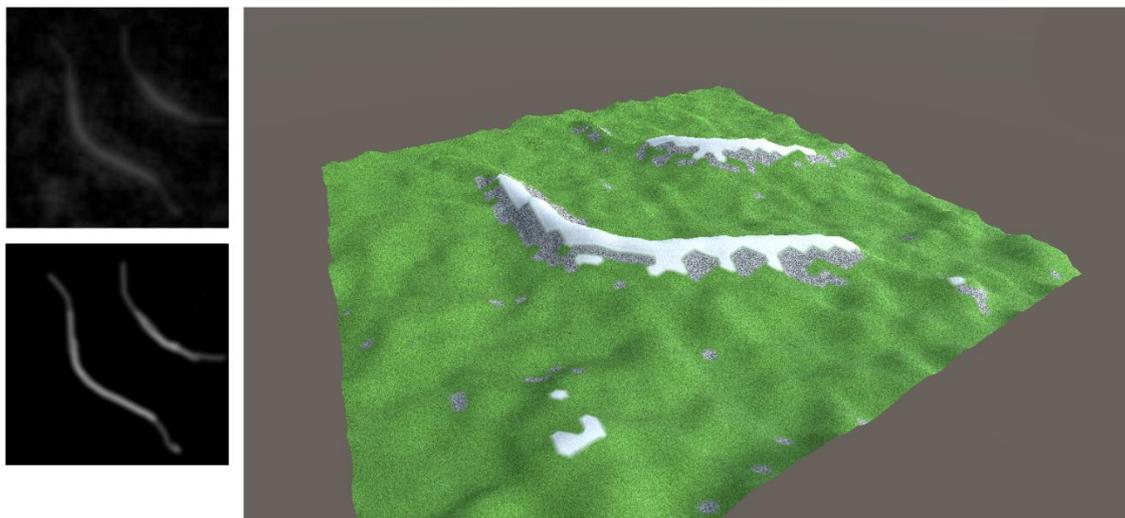


Figura 33 – Mapa de elevação resultante da modificação do esboço de linhas de serra com duas linhas curvas

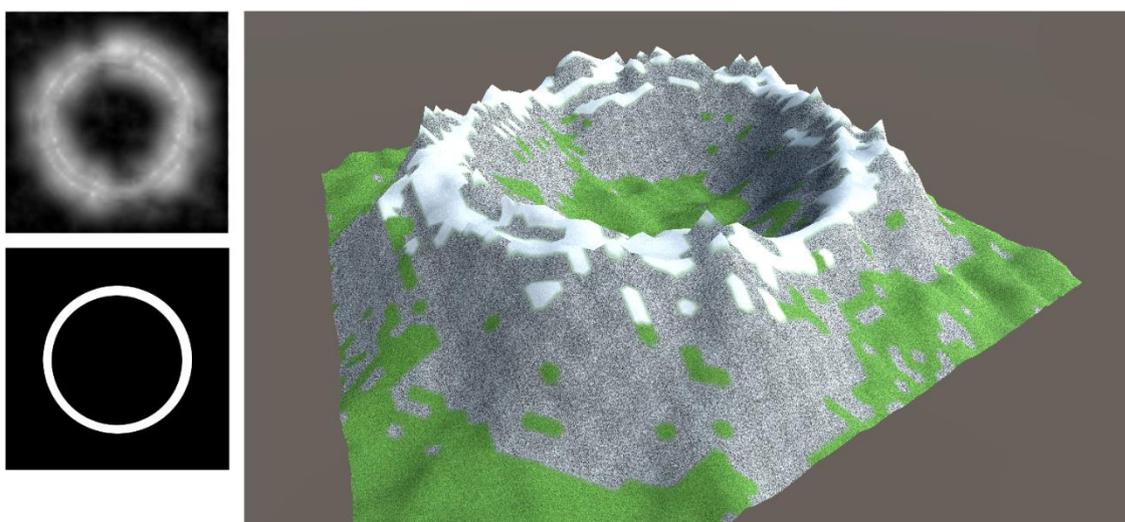


Figura 34 – Mapa de elevação resultante da modificação do esboço de linhas de serra com uma circunferência

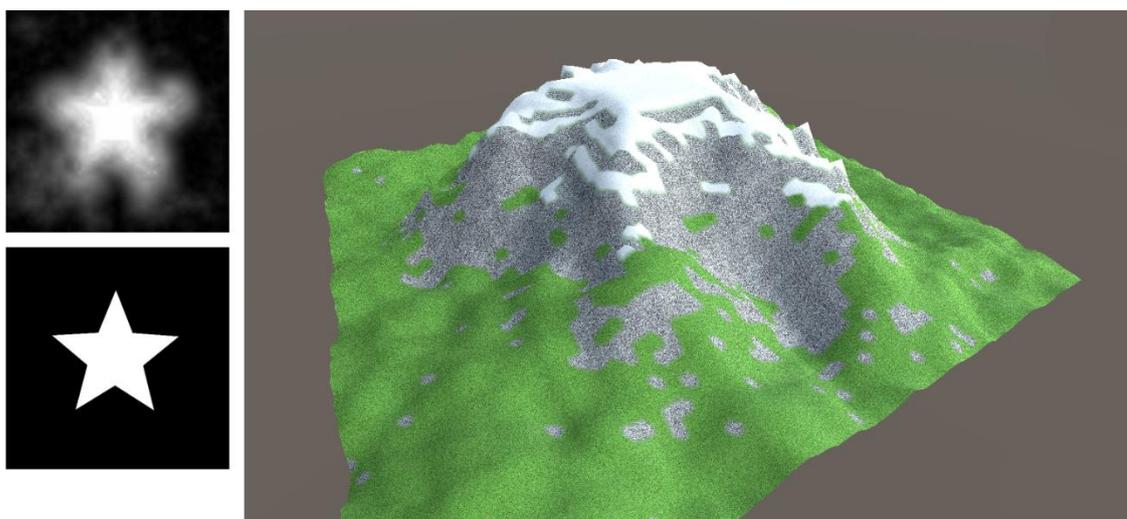


Figura 35 – Mapa de elevação resultante da modificação do esboço de linhas de serra com uma estrela

Os desfasamentos aleatórios do algoritmo *midpoint displacement* podem ser controlados através de um esboço de rugosidade. No modelo de terreno na Figura 36, é apresentado um mapa de elevações gerado sem linhas de serra, influenciado por um esboço de rugosidade que o divide numa zona de planícies e numa zona montanhosa.

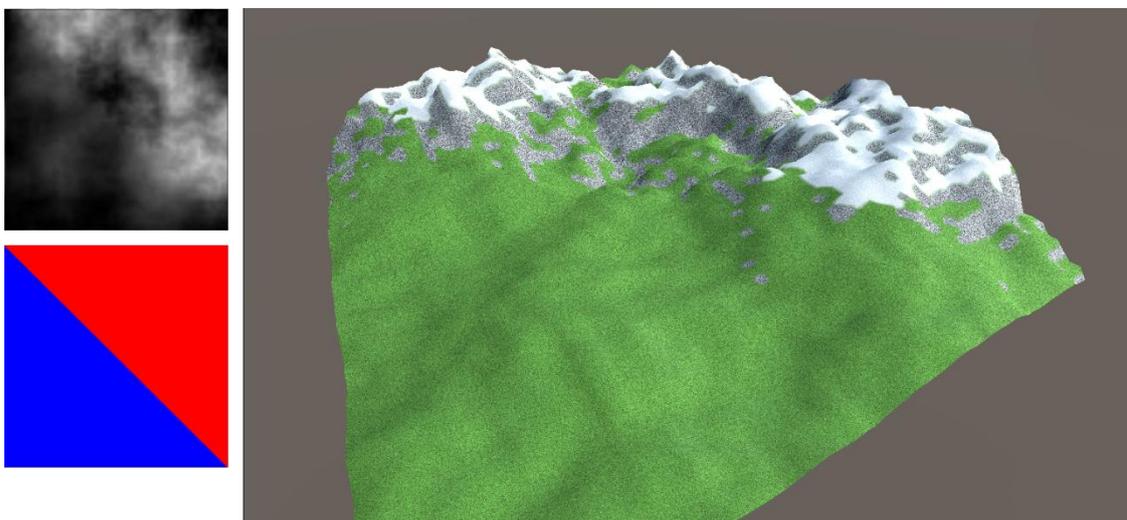


Figura 36 – Mapa de elevação resultante da modificação de um esboço de rugosidade

Nessa figura é observável que o mapa de elevação é mais acidentado nas zonas correspondentes à área vermelha do esboço e mais uniforme nas áreas azul-escuro.

Para melhor demonstrar o impacto do esboço de rugosidade foi recolhida a média da diferença de elevação entre células adjacentes de 100 mapas de elevação aleatórios gerados com os mesmos parâmetros mas para três esboços diferentes, um esboço de rugosidade mínima, um esboço de rugosidade intermédia e um esboço de rugosidade elevada. As médias dos valores foram calculados a cada dez amostras obtidas e são apresentados na Tabela 43:

Tabela 43 – Valores médios da variação da elevação entre células adjacentes para diferentes níveis de rugosidade (em metros)

	Baixa Rugosidade	Média Rugosidade	Alta Rugosidade
10 Amostras	14,4546	28,9529	43,3106
20 Amostras	14,4447	28,8769	43,1382
30 Amostras	14,4477	28,8993	43,2762
40 Amostras	14,4719	28,8639	43,3227
50 Amostras	14,4518	28,8282	43,3344
60 Amostras	14,4608	28,8255	43,2996
70 Amostras	14,4751	28,8677	43,2728
80 Amostras	14,4700	28,8655	43,2926
90 Amostras	14,4800	28,8812	43,3294
100 Amostras	14,4681	28,8696	43,3480

Como pode ser concluído na tabela, a variação entre células adjacentes tende a ser maior para o esboço de maior rugosidade e menor para o de menor rugosidade. Este comportamento deve-se ao facto do esboço de rugosidade influenciar diretamente o alcance do valor aleatório adicionado a cada subdivisão do algoritmo *diamond square*. O esboço de menor rugosidade reduz o alcance para metade, enquanto o esboço de maior rugosidade aumenta para o dobro do valor normal.

Os algoritmos de erosão têm um importante impacto na suavização do mapa de elevação gerado, tal como pode ser observado na Figura 37, Figura 38 e Figura 39.

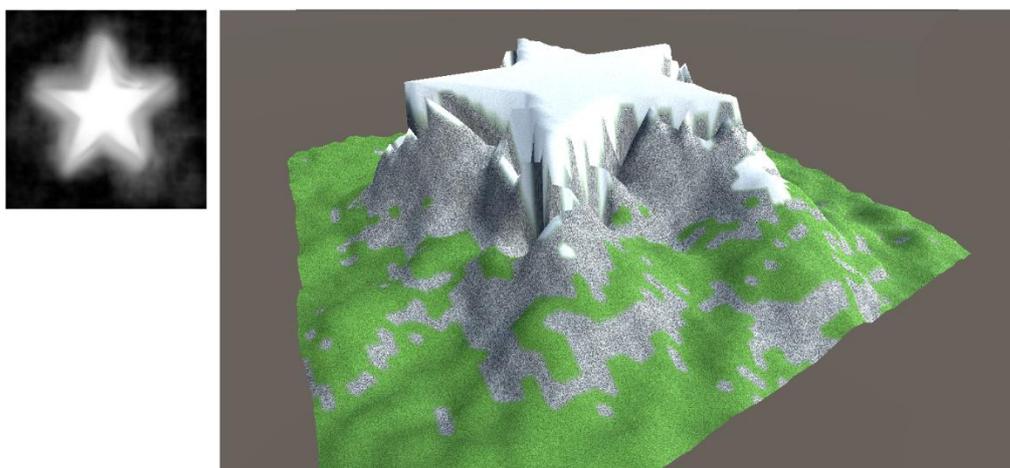


Figura 37 – Simulação de erosão no mapa de elevação com cinco iterações

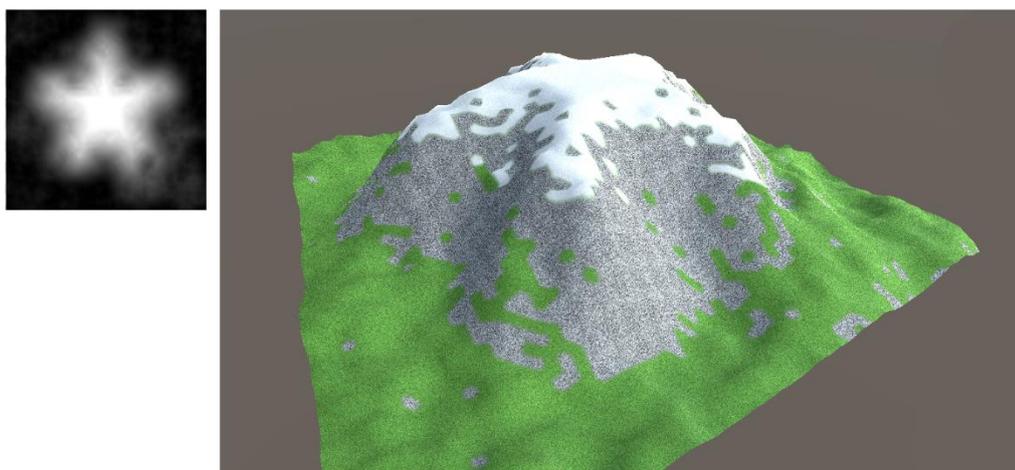


Figura 38 – Simulação de erosão no mapa de elevação com dez iterações

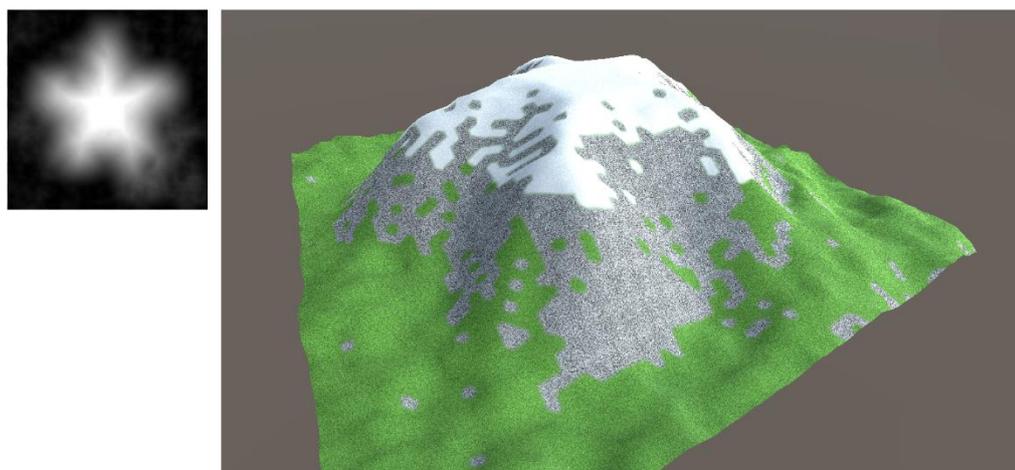


Figura 39 – Simulação de erosão no mapa de elevação com vinte iterações

Para testar a influência dos algoritmos de erosão nos mapas de elevação gerados foram recolhidas 100 amostras da média da variação de elevação entre células adjacentes de mapas de elevação aleatórios com os mesmos parâmetros para zero, cinco e dez passagens dos algoritmos de erosão térmica e hidráulica. Foi efetuada uma média a cada dez valores recolhidos. Os valores recolhidos são apresentados na seguinte tabela (Tabela 44):

Tabela 44 – Valores médios da variação da elevação entre células adjacentes para diferentes passagens dos algoritmos de erosão (em metros)

	0 Passagens	5 Passagens	10 Passagens
10 Amostras	43,31635	20,52426	17,47077
20 Amostras	43,30511	20,44501	17,3018
30 Amostras	43,24758	20,61087	17,31955
40 Amostras	43,27248	20,5186	17,319
50 Amostras	43,27123	20,61366	17,3005
60 Amostras	43,29766	20,49186	17,20982
70 Amostras	43,27942	20,48198	17,17932
80 Amostras	43,27129	20,44968	17,16884
90 Amostras	43,28488	20,44238	17,18209
100 Amostras	43,301	20,38937	17,176

Como observado na tabela, a simulação do desprendimento e transporte de sedimentos tende a reduzir as diferenças de elevação entre células adjacentes. Este comportamento tem como resultado a suavização do mapa de elevação gerado.

Para além dos parâmetros que se pretendiam testar, todas os mapas de elevação gerados nos ensaios efetuados usaram a mesma parametrização. Os mapas de elevação foram gerados com grelhas de 65*65 células onde cada célula tinha 100 metros de lado, elevação máxima de 2000 metros, zero passagens dos algoritmos de erosão, esboço de linhas de serra vazio e esboço de rugosidade intermédio. Para intensificar o impacto dos ensaios para os testes aos algoritmos de erosão, foi definido um esboço de elevada rugosidade

7.2 Resultados da Parametrização da Vegetação

A parametrização do processo de modelação da vegetação é essencialmente efetuada pela modificação do formato da copa da árvore a gerar. Outra parametrização possível é a influência do número de pontos de atração, comprimento dos segmentos de ramo (d_m), distância de atração (d_i) e distância de remoção (d_k) dos pontos de atração. Essas constantes têm impacto no formato dos ramos gerados, tal como demonstrado na Figura 40. As árvores representadas na figura usaram as seguintes constantes (Tabela 45):

Tabela 45 – Constantes usadas nos ramos das árvores geradas

	a)	b)	c)	d)	e)
Número de pontos	2000	1000	1000	1000	1000
d_m	5	5	5	30	5
d_i	70	70	150	100	70
d_k	10	10	50	50	50

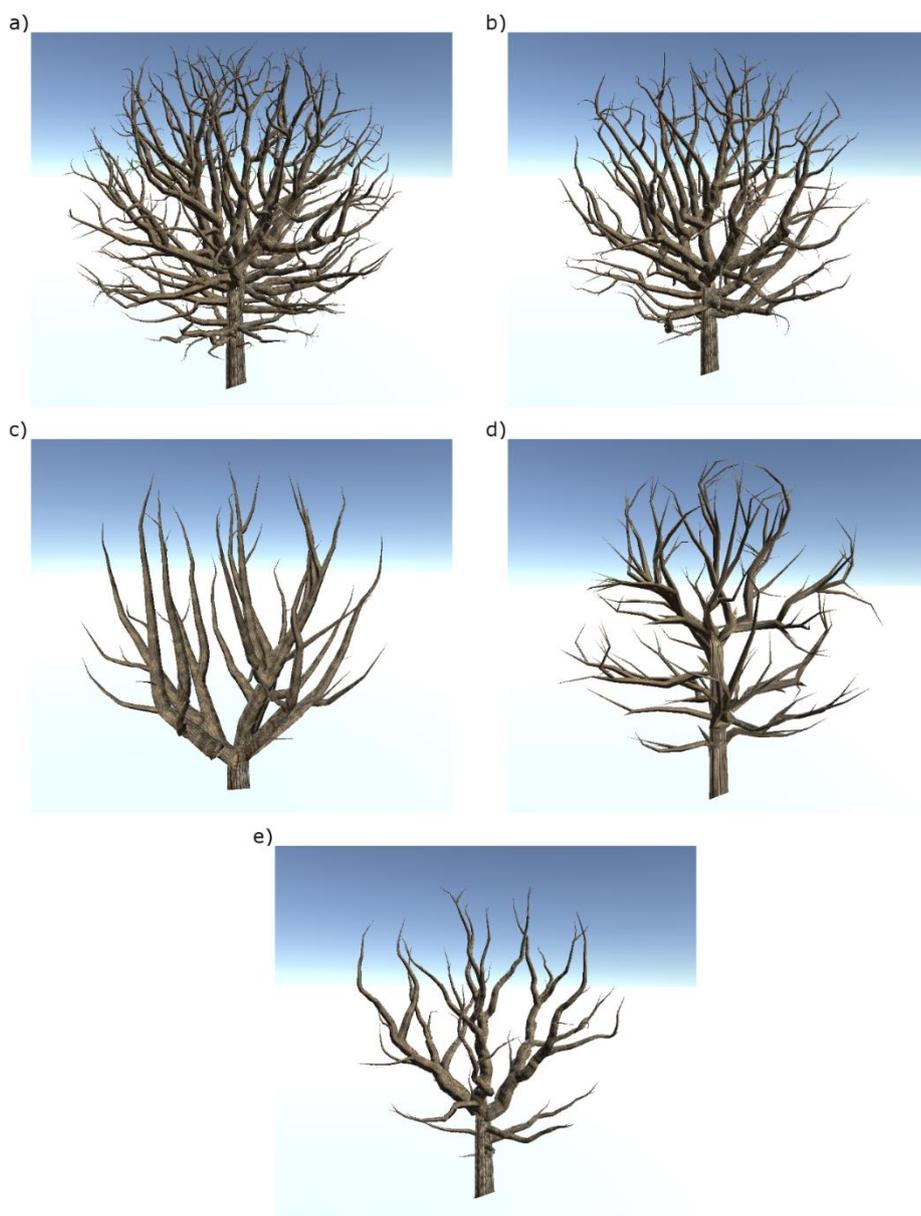


Figura 40 – Ramos de árvores gerados com diferentes constantes

Pelo que se pode observar, a ramificação das árvores a) e b) é mais fragmentada. Esta característica está relacionada com o menor valor da distância de remoção (d_k) que força os ramos gerados a uma maior subdivisão para se aproximar dos pontos de atração. O número de pontos de atração tem também um impacto visível na densidade dos ramos gerados.

Na ramificação c) é verificado o impacto de um elevado valor da distância de atração (d_i). Os ramos são gerados partindo da proximidade da base da árvore e com uma direção exata no sentido dos pontos de atração, o que faz com que os pontos de atração sejam removidos em menos iterações, resultando numa menor fragmentação.

A ramificação d) demonstra o impacto de um elevado comprimento dos segmentos de ramos (d_m). É notório o menor detalhe dos ramos gerados, que apresentam um formato reto.

A ramificação e) apresenta uma estrutura detalhada mas pouco densa devido à maior distância de remoção.

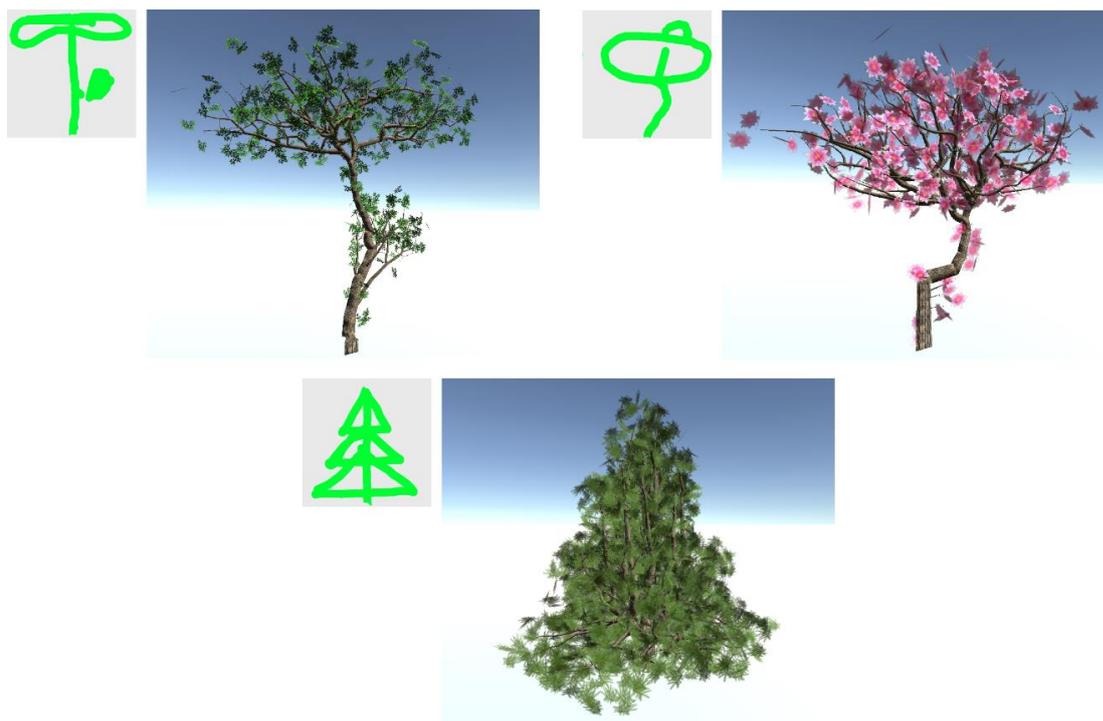


Figura 41 – Árvores geradas e respetivos esboços

Para demonstrar o potencial desta técnica de modelação procedimental aliada a meios de parametrização, são mostradas, na Figura 41, diferentes árvores geradas pela combinação de esboços do formato da árvore, constantes e texturas dos ramos e folhas.

7.3 Resumo do Capítulo

Neste capítulo foram apresentados diferentes modelos gerados proceduralmente pelos *plugins* implementados, através da *framework* desenvolvida e recorrendo às ferramentas de parametrização disponibilizadas.

Nos modelos gerados foi possível verificar o impacto que a modificação de parâmetros tem no resultado final. Na modelação de mapas de elevação a alteração de linhas de serra, rugosidade e o efeito da simulação da erosão alteram significativamente o tipo de terreno

gerado. Na modelação de vegetação é possível verificar a geração de árvores com diferentes características, permitindo uma aproximação a espécies reais. O impacto das ferramentas de parametrização é realçado pelos resultados apresentados, onde a definição de esboços e a manipulação de constantes permitem modificar o modelo gerado de forma a atender ao tipo de utilização pretendido.

8 Conclusão

Neste capítulo, são apresentadas as principais conclusões retiradas da realização desta dissertação, fazendo um balanço do estudo do tema deste trabalho, da solução desenvolvida, da implementação, dos resultados e contrapondo com os objetivos estabelecidos. São também apresentadas algumas sugestões de melhoria assim como apontados os desenvolvimentos futuros.

8.1 Objetivos Atingidos

No capítulo 1, foram identificadas várias dificuldades que constituem um entrave na adoção de técnicas de modelação procedimental para gerar modelos 3D de terrenos completos. Tendo em conta as vantagens na utilização deste tipo de técnicas na geração de modelos 3D face aos meios tradicionais, tornou-se evidente a importância do estudo e procura de soluções que reduzissem ou resolvessem essas dificuldades de integração.

Uma das dificuldades mencionadas foi a complexidade da escolha entre diferentes abordagens para a modelação procedimental de terrenos, devido ao número elevado de soluções. Para reduzir esta dificuldade foi proposta a catalogação de técnicas de modelação procedimental por características bem definidas do terreno. Esta separação foi evidenciada no capítulo do estado da arte (capítulo 2), onde foi estudada uma organização das técnicas de modelação procedimental no âmbito deste projeto. Sendo assim, as características de terreno consideradas foram: mapas de elevação, rios, oceanos e lagos, vegetação, redes de estradas e centros urbanos. Com esta separação, as técnicas de modelação podem ser comparadas e escolhidas de forma mais adequada, tendo em conta que modelam a mesma característica de terreno. Para essa comparação, foi proposta e explicada a utilização de uma técnica de apoio à tomada de decisões, a técnica AHP. A eficácia da técnica AHP foi demonstrada na escolha das técnicas mais adequadas para o tema deste projeto.

Outros problemas identificados estão relacionados com a dificuldade na adoção das técnicas de modelação procedimental na obtenção de modelos 3D por artistas e modeladores. Essas dificuldades devem-se à fragmentação das técnicas de modelação procedimental que dificultam a modelação de terrenos completos, e a falta de meios de parametrização que permitam influenciar facilmente o modelo resultante. Para resolver estas dificuldades foi desenvolvida e apresentada uma *framework* de modelação procedimental.

Esta *framework* permite que as técnicas de modelação específicas sejam integradas, considerando as diferentes camadas na modelação de um terreno completo. Permite que as técnicas de modelação procedimental sejam usadas por utilizadores que pretendam modelar terrenos 3D sem que sejam necessários conhecimentos aprofundados do funcionamento do

processo de modelação. Essa facilidade tem origem no *design* da *framework*, que estabelece regras e convenções na organização das técnicas de modelação, que constituem um *workflow* de modelação procedimental. As técnicas de modelação procedimental são implementadas como *plugins* que podem ser adicionados e removidos do *workflow*, permitindo a escolha de soluções alternativas para cada característica de terreno e de acordo com o resultado pretendido. Ferramentas para facilitar a parametrização e a visualização dos resultados foram também implementadas na *framework*.

A *framework* foi desenvolvida não só para utilizadores finais, mas também para facilitar o desenvolvimento e a distribuição de técnicas de modelação procedimental por investigadores e programadores. A organização proposta do processo de modelação procedimental permite que o desenvolvimento tenha um maior foco nos algoritmos a desenvolver. Os processos de modelação intermédios podem utilizar os resultados de outras camadas de modelação para produzirem os seus resultados e também usar as ferramentas de parametrização e visualização de modelos.

A facilidade da distribuição de técnicas de modelação é garantida pela arquitetura modular da *framework*, já que as técnicas constituem *plugins* que podem ser adicionados e removidos do *workflow* e contêm metadados que facilitam a sua identificação.

A eficácia da *framework* de modelação procedimental desenvolvida foi comprovada pela implementação de duas técnicas de modelação. As técnicas implementadas foram seleccionadas com base na técnica AHP e adaptadas de forma a permitir a fácil parametrização dos dados de entrada e a visualização dos resultados. A implementação demonstrou a utilização das ferramentas de parametrização e visualização e a integração através do *workflow* de modelação procedimental. Os modelos resultantes das técnicas de modelação consideradas foram analisados quanto ao impacto das diferentes parametrizações estudadas.

Considerando o potencial da *framework* de modelação desenvolvida, e sendo um produto inovador, foi efetuada a análise de valor da solução apresentada. Nesta análise foram identificados os elementos chave do modelo NCD numa primeira fase do processo de inovação. Um possível modelo de negócio foi descrito pela identificação dos clientes, intervenientes, principais interações e obtenção de lucro. Nesse modelo de negócio foi analisado o valor para o cliente, a rede de valor e foi construída a proposta de valor.

Tendo em conta o problema exposto na introdução, este trabalho conseguiu oferecer soluções que contribuem para melhorar a adoção de técnicas de modelação procedimental, resolvendo o problema da sua integração e controlabilidade do processo de modelação, atingindo com sucesso os objetivos inicialmente estabelecidos. A *framework* desenvolvida demonstrou ser um produto com potencial para a sua expansão e distribuição, podendo ser utilizado como elemento central num modelo de negócio que potencia a criação de valor.

8.2 Dificuldades Encontradas

A modelação procedimental de terrenos é um tema muito abrangente e com um número elevado de estudos efetuados. Todos os anos surgem novos contributos que complementam a base de conhecimento desta área da computação gráfica. Por esse motivo, o conjunto das técnicas de modelação abordadas no estado da arte teve de ser limitado, tentando selecionar diferentes contributos pela inovação da solução que contemplam e pela dispersão temporal em que surgiram.

As técnicas de modelação procedimental estudadas aceitam e geram dados de diferentes formatos, de acordo com a sua implementação. Esses dados podem, por exemplo, ser vetores, grelhas de valores, *arrays*, entre outros. Foi necessário definir, para cada característica de terreno, as estruturas de dados que permitissem de uma forma global descrever os modelos gerados para que outros *plugins* que implementam técnicas de modelação os possam considerar ou modificar. Esta imposição permite que cada *plugin* de modelação conheça os dados que irá receber e exportar, no entanto, esses dados podem ter de ser adaptados para o formato utilizado pelos algoritmos que implementam.

A flexibilidade da *framework* de modelação procedimental desenvolvida tem consequência na performance dos algoritmos de modelação. Todos os *plugins* de modelação têm de gerar modelos 3D de terrenos. Esta imposição deve-se ao facto de os *plugins* poderem ser configurados como a última camada do *workflow*, a necessidade de visualizar o terreno resultante em cada camada e do modelo gerado poder ser considerado na modelação da camada seguinte. A imposição do modelo 3D do terreno acrescenta complexidade na técnica de modelação procedimental e aumenta o tempo de execução do processo de modelação. De igual modo, a adaptação dos dados de entrada e saída dos *plugins* de modelação procedimental, também aumenta a complexidade tendo impacto na performance.

Devido ao âmbito deste trabalho e da complexidade envolvida na implementação de cada técnica de modelação procedimental, só foram implementadas duas técnicas. Apesar dessa limitação, foi possível aplicar e demonstrar detalhadamente a integração, parametrização e a inclusão no *workflow* de modelação. Também pelo mesmo motivo, algumas funcionalidades da *framework* de modelação não puderam ser implementadas por completo.

8.3 Sugestões e Melhorias

Os objetivos estabelecidos para este projeto foram atingidos com sucesso, no entanto, existem alguns melhoramentos que, devido à complexidade que introduzem, não puderam ser contemplados na solução final.

As técnicas de modelação procedimental foram consideradas como a combinação dos algoritmos, a geração do modelo 3D e a lógica de distribuição. Por essa razão, todas estas funções são implementadas num mesmo *plugin* que é utilizado na *framework*. No entanto,

muitas das contribuições estudadas no tema da modelação procedimental não abordam a distribuição ou a geração da geometria da característica de terreno que modelam, deixando essas funções à responsabilidade da implementação. Uma possível melhoria no *design* da *framework* seria a separação da geração dos modelos 3D e da distribuição em componentes isolados. Porém, esta alteração no *design* implicaria a existência de um maior número de *plugins* a configurar, o que poderia aumentar a complexidade na configuração.

Uma observação efetuada na implementação das técnicas de modelação procedimental foi o facto de existir um conjunto de algoritmos para gerar modelos 3D que podem ser reutilizados. Por exemplo, na execução de um algoritmo de modelação, um mapa de elevação previamente gerado pode ser modificado e implicar a criação do modelo 3D partindo do mapa de elevação. Neste caso, o algoritmo para a geração do modelo 3D a partir de um mapa de elevação é implementado no *plugin* de modelação de mapas de elevação e no *plugin* que utiliza o mapa de elevação gerado e que o altera. Dada esta observação, poderia ser feito um levantamento dos algoritmos reutilizáveis e adicioná-los a classes de utilitários disponíveis para os *plugins* de modelação procedimental.

8.4 Trabalhos Futuros

Para complementar este trabalho e preencher lacunas deixadas pelas limitações encontradas no desenvolvimento das soluções, foram identificados trabalhos que poderiam derivar deste projeto tornando-o ainda melhor.

Um possível trabalho, consistiria no estudo e implementação de outras técnicas de modelação procedimental de forma a criar os restantes *plugins* para as camadas de terreno que não foram implementadas, ou criar *plugins* com técnicas alternativas das camadas de terreno abordadas neste trabalho. Desta forma, configurando todas as camadas do *workflow* de modelação, seria estudada a modelação completa de um terreno contendo todas as características. Desse estudo das técnicas de modelação procedimental poderiam ser analisados novos meios de parametrização, o que poderia resultar no melhoramento das ferramentas de visualização e edição existentes e desenvolvimento de novas ferramentas.

Outro trabalho poderia aprofundar a utilização do método AHP, fazendo uma comparação mais exaustiva de diferentes abordagens ao tema da modelação procedimental e considerando outros critérios de comparação. Conjuntos de critérios poderiam ser agrupados para diferentes tipos de utilização da *framework*.

Referências

- Allee, V., 2002a. A Value Network Approach for Modeling and Measuring Intangibles. , (November).
- Allee, V., 2002b. Understanding Value Networks.
- Belhadj, F. & Audibert, P., 2005. Modeling Landscapes with Ridges and Rivers: bottom up approach. *Proceedings of the 3rd international conference ...*, 1, pp.1–4.
- Belliveau, P., Griffin, A. & Somermeyer, S., 2004. *The PDMA ToolBook 1 for New Product Development*, Wiley..
- Benes, B. & Forsbach, R., 2001. Layered data representation for visual simulation of terrain erosion. *Proceedings Spring Conference on Computer Graphics*.
- Benes, B. & Forsbach, R., 2002. Visual simulation of hydraulic erosion. *Wscg'2002, Vols I and II, Conference Proceedings*, pp.79–86.
- Bruneton, E. & Neyret, F., 2008. Real-time rendering and editing of vector-based terrains. *Computer Graphics Forum*, 27(2), pp.311–320.
- Campos, C., Leitão, J.M., Pereira, J.P., et al., 2015. Procedural generation of topologic road networks for driving simulation. *2015 10th Iberian Conference on Information Systems and Technologies, CISTI 2015*, (1).
- Campos, C., Leitão, J.M. & Coelho, A.F., 2014. Geração Procedimental de Traçados Rodoviários para Simulação de Condução Geração Procedimental de Traçados Rodoviários para Simulação de Condução. , (February 2016).
- Campos, C., Leitão, J.M. & Coelho, A.F., 2015. Integrated Modeling of Road Environments for Driving Simulation. *Visigrapp*, (February 2016), pp.70–80.
- Chen, G. et al., 2008. Interactive procedural street modeling. *ACM Transactions on Graphics*, 27(3), p.1.
- Coelho, A. et al., 2007. Expeditious Modelling of Virtual Urban Environments with. , 26(4), pp.769–782.
- Doran, J. & Parberry, I., 2010. Controlled procedural terrain generation using software agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 2(2), pp.111–119.
- Ebert, D.S., 2003. *Texturing & Modeling: A Procedural Approach*, Morgan Kaufmann..
- Emilien, A. et al., 2012. Procedural generation of villages on arbitrary terrains. *Visual Computer*, 28(6–8), pp.809–818.
- Fournier, A., Fussell, D. & Carpenter, L., 1982. Computer rendering of stochastic models. *Communications of the ACM*, 25(6), pp.371–384..
- Galin, E. et al., 2011. Authoring hierarchical road networks. *Computer Graphics Forum*, 30(7), pp.2021–2030.
- Galin, E. et al., 2010. Procedural generation of roads. *Computer Graphics Forum*, 29(2), pp.429–438.
- Génevaux, J.-D. et al., 2013. Terrain generation using procedural models based on hydrology. *ACM*

Transactions on Graphics, 32(4), p.1.

Greuter, S. et al., 2003. Real-time procedural generation of 'pseudo infinite' cities. *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pp.87–95.

Kelley, A.D., Malin, M.C. & Nielson, G.M., 1988. Terrain simulation using a model of stream erosion. *ACM SIGGRAPH Computer Graphics*, 22(4), pp.263–268.

Koen, P. et al., 2001. Providing Clarity and a Common Language To the «Fuzzy Front End.» *Research Technology Management*, 44(2), pp.46–55.

Lechner, T. et al., 2006. Procedural modeling of urban land use. *ACM SIGGRAPH 2006 Research posters on - SIGGRAPH '06*, p.135.

Lindenmayer, a, 1968. Mathematical models for cellular interactions in development. II. Simple and branching filaments with two-sided inputs. *Journal of theoretical biology*, 18(3), pp.300–315.

McCrae, J. & Singh, K., 2009. Sketch-based path design. *Proceedings - Graphics Interface*, pp.95–102.

Miller, G.S.P., 1986. The definition and rendering of terrain maps. *ACM SIGGRAPH Computer Graphics*, 20(4), pp.39–48.

Musgrave, F.K., Kolb, C.E. & Mace, R.S., 1989. The synthesis and rendering of eroded fractal terrains. *ACM SIGGRAPH Computer Graphics*, 23(3), pp.41–50.

Olsen, J., 2004. Realtime procedural terrain generation. *Department of Mathematics And Computer Science (...)*, p.20.

Osterwalder, A. & Pigneur, Y., 2010. *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers* 1.^a ed., Wiley.

Palubicki, W. et al., 2009. Self-organizing tree models for image synthesis. *SIGGRAPH - Special Interest Group on GRAPHics and Interactive Techniques*, 28(3), p.1.

Parish, Y.I.H. & Müller, P., 2001. Procedural Modeling of Cities. *28th annual conference on Computer graphics and interactive techniques*, (August), pp.301–308.

Perlin, K., 1985. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3), pp.287–296.

Peytavie, A. et al., 2009. Arches: A framework for modeling complex terrains. *Computer Graphics Forum*, 28(2), pp.457–467.

Pirk, S. et al., 2012. Plastic trees. *ACM Transactions on Graphics*, 31, pp.1–10.

Porter, M.E., 1985. *Competitive Advantage - Creating and Sustaining Superior Performance*. New York: FreePress, p.580.

Prusinkiewicz, P., 2000. Simulation modeling of plants and plant ecosystems. *Communications of the ACM*, 43(7), pp.84–93.

Prusinkiewicz, P., Lindenmayer, A. & Hanan, J., 1988. Development models of herbaceous plants for computer imagery purposes. *ACM SIGGRAPH Computer Graphics*, 22, pp.141–150.

Rodkaew, Y. et al., 2003. Particle systems for plant modeling. *Plant growth modeling and applications*

Proceedings of PMA03, 2(November).

Runions, A., Lane, B. & Prusinkiewicz, P., 2007. Modeling trees with a space colonization algorithm. *Natural Phenomena*, pp.63–70.

Schneider, J., Boldte, T. & Westermann, R., 2006. Real-time editing, synthesis, and rendering of infinite landscapes on GPUs. *Vision, modeling, and visualization 2006: proceedings*, p.145.

Smelik, R.M. et al., 2011. A declarative approach to procedural modeling of virtual worlds. *Computers and Graphics (Pergamon)*, 35(2), pp.352–363.

Smelik, R.M. et al., 2008. A Proposal for a Procedural Terrain Modelling Framework. *EGVE Symposium*, pp.1–4.

Smelik, R.M. et al., 2008. A Proposal for a Procedural Terrain Modelling Framework. *EGVE Symposium*.

Smelik, R.M. et al., 2009. A survey of procedural methods for terrain modelling. *3AMIGAS - 3D Advanced Media In Gaming And Simulation*, (June 2015), pp.25–34.

Smelik, R.M. et al., 2014. A survey on procedural modelling for virtual worlds. *Computer Graphics Forum*, 33(6), pp.31–50.

Smelik, R.M. et al., 2010. Declarative terrain modeling for military training games. *International Journal of Computer Games Technology*, 360458(11).

Stachniak, S. & Stuerzlinger, W., 2005. An Algorithm for Automated Fractal Terrain Deformation. In *Proceedings of Computer Graphics and Artificial Intelligence*, pp.64–76.

Weber, J. & Penn, J., 1995. Creation and rendering of realistic trees. *SIGGRAPH '95 - Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pp.119–128.

Woodall, T., 2003. Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis. *Academy of Marketing Science Review*, 12(5), pp.1–42.

Zhou, H. et al., 2007. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, 13(4), pp.834–848.