

Realidade Virtual em cenários de Incêndios Florestais

Marta Carlos*

Orientadores:

Professor Fernando Birra†

Professor João Moura Pires ‡

Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa



Figura 1: Floresta de Eucalipto, 3 de Julho 2011, Ilha da Madeira, Portugal [anagh 2011]

Resumo

Os incêndios florestais presentes no períodos quentes e secos afetam o meio rural em Portugal, tendo em 2017 atingindo o maior número de hectares e perda de vidas humanas de toda história do país. A evidência deste fenómeno extremo trouxe uma necessidade urgente de investir e explorar novas formas de apoio à decisão no combate a incêndios.

As operações de combate aos incêndios florestais requerem uma complexa coordenação de vários meios com base na melhor informação disponível em cada momento. Neste contexto, é essencial que a tomada de decisão assente num retrato tão fiel quanto possível da realidade e dos fatores imediatos que afetam a sua evolução. Quem toma as decisões deve, além do mais, ter a percepção do que os operacionais no terreno estão a enfrentar.

A Realidade Virtual (RV) é uma tecnologia que, ao longo dos anos, tem marcado cada vez mais a sua presença em estudos científicos e utilizada como ferramenta de simulação e treino. Neste sentido, é proposta a utilização de RV como ferramenta para permitir obter uma melhor percepção dos principais fatores que contribuem para a progressão de um incêndio florestal como o declive, exposição e vegetação.

Um primeiro passo necessário para representar a frente de fogo é a representação do próprio terreno com elevados níveis de realismo. Neste trabalho é proposta e avaliada uma metodologia para o uso de uma plataforma de jogos em RV para a navegação num modelo de terreno tridimensional construído a partir de modelos reais de elevação de terreno (DEM), ortofotos e informação cartográfica de precisão.

Keywords: Realidade Virtual, Incêndios Florestais, HTC Vive,

Unreal Engine 4, Modelação de Terreno GIS

1 Introdução

Em 2003 e 2005, ocorreram as maiores catástrofes ambientais alguma vez registados na história de Portugal, sendo que nestes dois anos, ardeu um total de cerca de 750 mil hectares¹.

Posteriormente, viveu-se uma calma aparente durante doze anos onde se acreditou que incêndios de igual escala não voltariam a assombrar as florestas de Portugal. Contudo, após a perda de mais de 100 vidas humanas e 442.418 hectares de área ardida no ano 2017 [de Gestão de Áreas Públicas e de Proteção Florestal 2017], os incêndios florestais provaram continuar a ser uma catástrofe que Portugal ainda não consegue minimizar.

O apoio à decisão no combate a estes incêndios é um fator determinante no impacto dos mesmos no país e na população. Procurando fornecer um melhor suporte a esse combate, pretende-se representar a frente de fogo, utilizando a tecnologia Realidade Virtual (RV). Esta representação será uma ferramenta de auxílio numa avaliação do estado e das diversas características do terreno presente na frente de fogo.

*e-mail:m.carlos@campus.fct.unl.pt

†e-mail: fpb@fct.unl.pt

‡e-mail: jmp@fct.unl.pt

¹Incêndios florestais e área ardida – Continente (2018, 2 de Maio), visitado em [https://www.pordata.pt/Portugal/Incêndios florestais e área ardida – Continente-1192](https://www.pordata.pt/Portugal/Incêndios%20florestais%20e%20área%20ardida%20-%20Continente-1192)

Profissionais na gestão de recursos e meios operacionais têm acesso a um número vasto de informações sobre a área onde decorre o incêndio, no entanto, estas informações apenas oferecem uma visualização limitada sob gráficos e imagens 2D.

O uso de RV, poderá permitir ao responsável da tomada de decisões ter a percepção real da evolução do incêndio, sem estar fisicamente presente no local. As características imersivas e interativas associadas à RV permitirão oferecer leituras da frente de fogo, das características do terreno e do posicionamento dos meios no combate ao incêndio que hoje não são possíveis através de informações apresentadas em gráficos e imagens 2D.

De forma a obter um maior nível de imersão, é necessária uma reprodução realista do terreno no mundo virtual, replicação essa que deverá respeitar as condições do terreno no momento imediatamente anterior ao incêndio.

Nas secções seguintes são identificados os aspetos em que a RV já foi empregue como apoio à decisão em contexto de emergência; a razão da escolha de Unreal Engine 4 (UE4) como plataforma de trabalho; a abordagem escolhida; a sua implementação; a avaliação do trabalho produzido; conclusão do processo de trabalho, terminando com possíveis abordagens para trabalhos futuros.

2 Estado da Arte

A RV cresceu principalmente devido ao seu desenvolvimento na área de jogos e multimédia. Contudo, esta tecnologia não está restrita à indústria do entretenimento e tem sido aplicada em outras áreas temáticas em diferentes estudos científicos.

Na área da biomédica, foram desenvolvidas aplicações em RV como auxílio à compreensão da correlação entre a personalidade de cada indivíduo e as suas decisões perante situações moralmente ambíguas [Ju et al. 2016]. Igualmente na área da saúde, onde a falha humana é a causa de 80% das falhas cirúrgicas, a RV poderá servir como uma ferramenta útil para o treino de futuros profissionais [Wei and Wen-qí 2010]. Permitiu também simular o fluxo sanguíneo em redes cardiovasculares complexas, onde a visualização proporcionada pela RV contribui para a melhor compreensão dos passos a seguir por parte dos médicos [Holland et al. 2017].

Com vista a diminuir os custos e a eliminar o fator de risco humano, foi desenvolvida uma aplicação em RV num ambiente metropolitano [Sharma et al. 2014] e num ambiente citadino [Sharma et al. 2017] para simular uma evacuação em caso de incêndio. Os utilizadores podem interagir entre si no mundo simulado e decidir como seria realizada a evacuação. Há ainda referências a estudos focados na simulação de evacuações de multidões em aviões [Sharma et al. 2011] e estádios [Yi 2011].

Uma simulação em RV foi também realizada para o treino à tomada de decisão de civis e profissionais de Gestão de Emergência. Esta simulação coloca o participante num situação de emergência, em que o mesmo toma as decisões necessárias para a resolver. Nesta simulação estão presentes diversos *avatars* de diferentes equipas de emergência, podendo cada equipa ter associado as suas respetivas funções. Restrições de tempo e fatores de stress estão também presentes nesta simulação.

Os resultados mostram que o nível de realismo e complexidade necessário para uma boa simulação estão relacionados com a experiência prévia do participante: profissionais exigem um maior nível de realismo. Contudo, provou ser uma simulação que providencia um bom método de aprendizagem para não-profissionais [Prasolova-Forland et al. 2017].

Para a avaliação do comportamento humano em situações de incêndio, a RV tem como principais vantagens a sua validade interna e externa: o controlo e manipulação dos estímulos numa simulação de incêndio e os resultados do estudo poderão ser generalizados para outras abordagens. A sua maior desvantagem é a dificuldade de demonstração da validade ecológica [Kinadeder et al. 2014]: a validade é alta, mas não absoluta. Ambientes virtuais não substituem a realidade, são abstrações desta, logo haverá perda de validade ecológica. Participantes de um estudo terão sempre o conhecimento que se trata de um estudo, pela presença de aparelhos não intuitivos como HMD (Head Mounted Display) e controles externos, que tornam difícil a completa abstração do utilizador num mundo virtual. A RV continua a ser uma ferramenta importante para o estudo do comportamento humano, que só através de mais dados e simulações poderá afirmar a sua contribuição para a replicação do mundo real em ambientes controlados [Kinadeder et al. 2014].

Sendo RV uma tecnologia reconhecida por distintas áreas científicas, e com bases já sustentadas, tendo-se revelado útil para simulações de situações de emergência, sugere-se a possibilidade do alcance de bons resultados como ferramenta de apoio à decisão da gestão dos recursos e meios operacionais no combate a incêndios.

3 Avaliação das plataformas de Realidade Virtual

Um motor de jogos é um ambiente de desenvolvimento de software cujo principal objetivo é simplificar a construção de um jogo. São plataformas que oferecem diversas ferramentas para o desenvolvimento de jogos, assim como um conjunto de funcionalidades sofisticadas e essenciais nesse desenvolvimento, evitando a necessidade da sua implementação. Com o rápido interesse de jogos em supor RV por parte do público alvo, houve a necessidade de garantir compatibilidade dos motores com este avanço tecnológico, razão pela qual motores como Unity3D e UE4 dispõem já suporte para RV.

Apesar do principal objetivo destas plataformas ser o desenvolvimento de videojogos, são também utilizadas para outros tipos de aplicações gráficas 3D interativas. Tendo em consideração o objetivo principal deste projeto, o uso de um motor de jogos era a escolha natural. É então necessário realizar uma avaliação dos motores de jogo disponíveis, para a realização de uma escolha informada.

Apesar da grande diversidade de motores de jogos na indústria, optou-se por fazer apenas uma escolha entre Unity3D e UE4, devido a estes serem os motores de jogos mais conhecidos e proclamados no mercado atual.

Unity3D², desenvolvido por Unity Technologies, é atualmente o motor mais usado para a elaboração de jogos com cerca de 34% dos 1000 jogos móveis grátis *mainstream* a serem criados neste motor³. É um motor especialmente conhecido como o melhor para jogos *mobile* e pelo seu suporte em diversas plataformas (iOS, Android, Linux, Windows, etc).

Unreal Engine 4⁴, desenvolvido por Epic Games, é conhecido para a criação de jogos com maiores orçamentos, a sua qualidade gráfica, *filmmaking* e visualização foto-realista.

²Unity, visitado em <https://unity3d.com>

³Company Facts, visitado em <https://unity3d.com/public-relations>

⁴What is Unreal Engine 4, visitado em <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>

Para ser feita uma escolha acertada e consciente entre estes dois motores de jogos, foram analisados os seguintes aspectos:

- **Modelação de terrenos usando dados GIS:** de modo a garantir o maior nível de realismo, é necessário trabalhar com informações que representam realmente o terreno que se pretende gerar. Terrenos virtuais gerados com base principal em dados GIS serão terrenos fieis e realistas.
- **Diferentes níveis de detalhe:** uma boa compreensão, gestão e funcionamento de diferentes níveis de detalhe é uma característica importante quando se pretende trabalhar com grandes quantidades de malhas, por questões de otimização.
- **Carregamento dinâmico de terrenos:** ao trabalhar com grandes áreas de terreno, o carregamento total deste em memória implicaria grandes custos no desempenho, de tal forma, que uma boa implementação de carregamento dinâmico é necessária.
- **Opinião de developers:** o sucesso de um produto é determinado principalmente pela classificação dada pelos seus consumidores e quem desenvolve esse mesmo produto. Uma boa percepção de como as opiniões sobre estes dois motores se distinguem é fundamental para uma escolha acertada.

Nas secções seguintes segue-se uma análise detalhada destes quatro aspectos. Esta análise conduziu à escolha de UE4 como o motor de jogos a usar, tendo o motor disponíveis várias ferramentas úteis para a realização de um protótipo na temática deste projeto.

3.1 Modelação de terrenos usando dados GIS

Ambos os motores oferecem a opção de carregar imagens externas ao motor para serem usadas como texturas, sendo assim possível carregar ortofotos nos motores. Cada um dos motores disponibilizam ferramentas para a manipulação manual de terrenos, mas uma recriação realista dificilmente seria alcançável utilizando apenas este género de ferramentas. Em alternativa, ambos dispõem do possível uso de *Heightmaps*, ou seja, Data Elevation Models (DEMs).

Um DEM é uma imagem raster bidimensional de valores, tais como a elevação de uma superfície. Estas imagens poderão ser utilizadas pelos motores de jogos para uma posterior reprodução do terreno em gráficos 3D. Imagens DEMs estão disponíveis em diversos formatos, havendo a necessidade de conversões de ficheiros antes destas imagens serem importadas por um motor de jogos.

Nenhum dos motores oferecem ferramentas para a manipulação direta de informação cartográfica como Shapefiles, sendo necessária a implementação de código suplementar C# (Unity3D) e C++ (UE4).

3.2 Diferentes níveis de detalhe

Num cenário de simulação de incêndio, existem vários componentes que necessitam de ser visualizados, como os diversos tipos de vegetação do local. Não é necessário, por questões de otimização, que um componente do ambiente, que se encontre a uma distância considerável do utilizador no mundo, seja completamente renderizado com todos os seus detalhes.

A técnica de otimização utilizada para evitar este tipo de situações é designada Nível de Detalhe⁵ (LOD) e permite reduzir o número de triângulos desenhados dependendo da distância da câmara ao objeto em questão.

No caso do Unity3D, a técnica é implementada através de Grupos de Componentes LOD (LOD Group Component). Estes grupos são utilizados para gerir os diferentes LODs associados aos objetos. Este componente funciona à base de atribuição de níveis: é possível manipular propriedades deste componente e do objeto consoante o nível de detalhe, como por exemplo, que malha de polígono usar num determinado nível.

O sistema LOD do UE4 seleciona um nível de detalhe de um objeto num dado momento consoante a percentagem do espaço que o objeto ocupa no ecrã. Usando o mecanismo da Computação Automática de Distâncias dos LODs (Auto Compute LOD Distances) é possível ao utilizador decidir quando trocar um LOD por outro mais adequado, através da divisão equitativa da percentagem do tamanho do ecrã pelo número de LODs disponíveis.

Em alternativa a este mecanismo, é possível a personalização de qual percentagem em que se quer realizar a troca de LOD. UE4 também dispõe da Geração Automática de LODs (Automatic LOD Generation) onde apenas é necessária a atribuição da malha mais detalhada. As malhas respetivas aos restantes níveis de LODs são computadas pelo próprio motor.

O sistema de geração automática de LODs do motor UE4, dado os recursos disponíveis, apresenta ser um sistema apelativo e útil para a realização de um protótipo.

3.3 Carregamento dinâmico de terrenos

Ao trabalhar com grandes áreas de terreno, o carregamento total deste em memória causaria grandes perdas no desempenho. É então necessário recorrer a técnicas de carregamento dinâmico, de forma a reduzir o número de componentes carregados em memória, mas criando a ilusão de se estar perante um terreno completamente contínuo.

3.3.1 Unity3D

Unity3D não dispõe de nenhuma forma direta para lidar com o carregamento dinâmico de terrenos, sendo necessário realizar a sua implementação.

O tema do carregamento dinâmico de terrenos é um tema fértil para discussão, sendo até publicado em 2013, um vídeo por Unity Technologies sobre o assunto. Nesse vídeo, intitulado "Unite 2013 - GIS Terrain & Unity"⁶, é dedicado à integração dos dados GIS e gestão de quantidades arbitrárias de terreno em simulações e jogos *openworld*. A solução apresentada é resumida em:

- Divisão do terreno em *Terrain Mosaics* com dimensões $n \times n$ (ou seja, os *Terrain Mosaics* deverão ser quadrados)
- Cada *terrain tile* terá os seus valores de latitude e longitude, como também a sua origem e posição
- A informação sobre cada *terrain tile* terá de ser armazenada numa base de dados (ou outra forma de armazenamento externo)
- Para se obter os *terrain tiles* quando é necessário, são feitas *queries* à base de dados, sendo essas *queries* baseadas nos valores de longitude e latitude associados a cada *terrain tile*

De modo a enriquecer esta solução, é possível recorrer à utilização de *Asset Bundles*. São ficheiros completamente separados do arquivo principal do projeto que contêm *assets* prontos a serem utilizados. Estes ficheiros agem como coleções de recursos (*assets*)

⁶Unite 2013 - GIS Terrain & Unity (2013, 12 de Dezembro) visitado em <https://www.youtube.com/watch?v=VKWvAuTGVrQ>

⁵Level of Detail em inglês

e existem fora do editor, normalmente armazenados num servidor web para poderem ser acedidos dinamicamente.

Carregamento dinâmico não é então uma característica de raiz do motor mas através de código C# é possível ser alcançado.

3.3.2 Unreal Engine 4

O Streaming de Níveis (LS⁷) é uma ferramenta do motor utilizada para o carregamento dinâmico de mapas para a memória ao dividir o mundo virtual em várias partes. Esta ferramenta permite um maior controlo sobre a quantidade de recursos e malhas de polígonos a serem renderizadas.

LS oferece a possibilidade da criação de grandes mundos, para tal ser efetuado com êxito é necessária a criação de um Nível Persistente (PL⁸). Este nível poderá ser considerado como um nível "mestre" cujo principal objetivo é coordenar quais os níveis que serão carregados. Os níveis poderão ser definidos como Sempre Carregados (*Always Loaded*), mas sendo que o projeto a ser implementado envolve grandes terrenos, esta opção não é relevante.

O terreno do mundo a ser criado necessaria de ser dividido em diversos *tiles* para tirar o maior proveito deste sistema: cada nível representaria um *tile* no mundo.

Em acréscimo a este sistema, World Composition é um sistema específico de LS frequentemente utilizado para a criação de grandes mundos. Os diferentes níveis podem ser organizados numa grelha planar e carregados enquanto o utilizador se aproxima. Um dos maiores propósitos desta ferramenta é evitar problemas de *bottleneck* no PL. Desta forma, o Nível Persistente não armazena qualquer informação de *streaming* e como alternativa explora as pastas e lida todos os níveis encontrados como níveis de streaming.

A necessidade de realizar a implementação do carregamento dinâmico no Unity3D surge como uma desvantagem na futura utilização deste motor, uma vez que, UE4 disponibiliza à partida o seu próprio mecanismo. Caso as prioridades do projeto valorizassem um maior controlo sobre as técnicas do carregamento dinâmico (maior controlo poderia permitir a garantia da menor perda possível de desempenho), a decisão entre os dois motores dependeria de outros fatores. No entanto, a compreensão e o conhecimento de qual a melhor metodologia a seguir para tornar um protótipo escalável é uma maior prioridade, ou seja, o sistema de LS do UE4 prova ser uma vantagem.

3.4 Opinião de *developers*

A opinião de *developers* é indiscutivelmente das mais importantes em plataformas de desenvolvimento como Unity3D e UE4. As várias opiniões são uma reflexão direta do impacto da plataforma, do seu desempenho, da sua interface e do quanto intuitivo o uso desta é.

Quando se está a avaliar a interface, a preferência baseia-se, geralmente, em aspectos subjetivos. Uma conclusão sobre qual dos motores oferece a melhor interface não deverá ser feita com base nestas opiniões.

De seguida estão listados os vários prós e contras de cada motor baseados nas diferentes opiniões encontradas.

Unity3D apresenta como vantagens:

- Permitir a implementação em diversas plataformas, como *mobile*, RV, Android, PC, iOS, Linux, entre outros

- Ser melhor escolha para aplicações móveis
- Oferecer acesso a uma grande variedade de plugins através da sua *Asset Store*
- Apresentar uma curva de aprendizagem mais adequada a principiantes - a plataforma é intuitiva
- Devido à sua grande comunidade, é mais fácil encontrar respostas a problemas encontrados durante o desenvolvimento.

Desvantagens:

- Não oferecer a mesma qualidade gráfica que UE4
- Apenas disponibilizar o código fonte em condições específicas, sobre forma de pagamento
- O processo de *renderização* poderá vir a ser lento
- Apesar da sua curva de aprendizagem inicial se mostrar agradável, para projetos de dificuldade intermédia ou superior, a falta de certas ferramentas prova ser uma desvantagem

As opiniões a favor de UE4 são:

- Em termos de qualidade gráfica, aparenta ser das melhores no mercado
- O seu sistema de Blueprint Visual Scripting permite a utilizadores com pouca compreensão/conhecimento de código usar o motor para questões mais iniciais/básicas. É considerado um bom sistema para fazer protótipos
- Disponibiliza o código fonte, algo que poderá facilitar o desenvolvimento e melhorar a sua eficiência
- Várias ferramentas já disponibilizadas, sem ser necessário recorrer à utilização de plugins
- Oferece várias opções de otimização

Tendo como contras:

- Não é aconselhável para projetos pequenos
- O editor é exigente na memória e no disco
- A sua *Asset Store* não oferece tantas opções

4 Abordagem

A integração dos dados geográficos no motor de jogos é a temática abordada neste relatório. Estes dados podem ser representados por imagens *raster* georreferenciadas ou informação vetorial (informação cartográfica).

Numa imagem *raster* georreferenciada, cada pixel da imagem tem a ele uma localização geográfica associada, sendo armazenada informação no valor do pixel, como por exemplo, um tom de cinzento para armazenar um valor de elevação (DEM) ou um valor RGB (ortofotos).

A informação vetorial provém de ficheiros do tipo Shapefile. Um Shapefile é um formato vetorial de representação de informação geográfica, ou seja, consegue representar informação vetorial através de pontos, linhas e polígonos. Esta informação vetorial pode codificar os mais variados temas, tais como redes viárias, cursos de rios, reservas de água, perímetros urbanos e industriais, zonas classificadas, linhas de alta tensão, marcos geodésicos, etc.

Será avaliada e discutida uma metodologia que descreva um processo, extensível e não sujeito ao erro subjetivo, para a integração deste tipo de dados escalável.

⁷Level Streaming

⁸Persistent Level



Figura 2: Diagrama de fases para o desenvolvimento do protótipo

Esta metodologia será dividida em cinco fases principais (figura 2). A primeira fase irá consistir na seleção de uma área de interesse de modo a realizar-se um protótipo sobre essa mesma área. A área de interesse escolhida deverá conter composição florestal característica de áreas com alta probabilidade de incêndio. Deverá também apresentar uma diferença considerável nos dados de elevação de terreno.

É importante salientar que estes critérios são apenas necessários para a implementação de um protótipo capaz de generalizar o processo para qualquer área pretendida.

Para a recriação da área selecionada, é necessário o acesso aos dados do terreno. Estes dados deverão corresponder a dados de elevação do terreno (ficheiros DEM), imagens áreas (ortofotos) e ficheiros de informação geográfica (ficheiros Shapefile) da área .

Os dados DEM são os que permitem a criação de um modelo 3D do terreno que respeite as suas dimensões e diferentes valores de altitude. Logo, uma segunda fase consistirá na preparação dos dados DEM para serem integrados no motor.

Tendo o terreno tridimensional que respeite os dados DEM, este estará pronto para a adição da vegetação respetiva. Para esta adição de vegetação, vai-se recorrer ao uso de ortofotos. De modo a poupar recursos, o mapeamento de ortofotos como textura no terreno será realizado para zonas do terreno consideradas a uma distância tão longe do utilizador que não será necessário a colocação de malhas individuais para a representação da vegetação. As ortofotos serão também utilizadas para obter máscaras do terreno que indicam onde está presente a sua vegetação. Serão então colocadas malhas de vegetação no terreno consoante a informação obtida através das máscaras, mas apenas em regiões perto da câmara.

A colocação de estradas e trilhos no mundo virtual irá necessitar da leitura de Shapefiles. Obtendo a informação respetiva aos Shapefiles da área, irá realizar-se a conversão das coordenadas geográficas, realizar a sua conversão para coordenadas no mundo virtual e mapear as malhas respetivas no terreno consoantes as coordenadas calculadas.

Dispondo de acesso ao equipamento RV HTC Vive, irá ser implementada uma navegação RV que usufrui dos controlos de movimento deste equipamento. Com o HMD, o utilizador entrará no ambiente virtual e através dos controlos de movimento, poderá usar o seu teleporte para navegar no mundo virtual.

5 Implementação

5.1 Fase 1: Área de interesse

Devido à disponibilidade e acesso imediato dos dados disponibilizados por USGS National Map⁹, foi realizado um protótipo sobre terreno americano, nomeadamente terreno da região Twin Mountain, New Hampshire, EUA. Os dados respetivos e os seus detalhes estão enumerados na Tabela 1.

Dados	Detalhes
DEM	<ul style="list-style-type: none"> - dimensões de 10012 x 10012 pixels - 1m² por pixel - projeção geográfica EPSG:26919
Ortofotos	<ul style="list-style-type: none"> - NAIP Plus - resolução entre 1m e 1ft¹⁰ de superfície terrestre - projeção geográfica EPSG:3857
Shapefiles	<ul style="list-style-type: none"> - informação referente a estradas - informação referente a trilhos

Tabela 1: Informação detalha dos dados obtidos

Para facilitar a manipulação dos dados respetivos à área de interesse, foi feita a junção dos ficheiros (do mesmo tipo) obtidos.

Tendo acesso a estes dados, foi selecionada uma área de interesse com cerca de 5 km x 5 km de dimensão. A área selecionada corresponde a um total de 2500 hectares, ou seja, 0,5% da área ardida em 2017.

A área selecionada permite explorar as possíveis dificuldades na modelação de um terreno virtual com dimensões consideráveis, como a integração dos vários dados geográficos correspondentes. É uma área com uma composição florestal significativa e com dados de elevação análogos à realidade que se pretende simular.

Para a recriação de um terreno virtual com estes dados, foi avaliado o uso de programas externos, como a plataforma interativa QGIS e Adobe Photoshop, no entanto, o seu uso exclusivo para uma manipulação total dos dados não é escalável.

De modo a se apenas obter os dados respetivos à área de interesse, é necessário realizar um corte sobre os ficheiros junção mencio-

⁹USGS National Map Viewer visitado em <https://viewer.nationalmap.gov/basic/>

¹⁰Um pixel poderá cobrir 1m² ou 1ft²

nada anteriormente. Este corte deverá ter em consideração as coordenadas geográficas da área de interesse e as diferentes projeções geográficas dos ficheiros de modo a garantir a coerência dos dados.

Sendo GDAL uma biblioteca disponibilizada para a manipulação e leitura de ficheiros *raster* e outros dados geoespaciais, o corte dos ficheiros é efetuado com comandos desta biblioteca. O corte respeitivo à área de interesse é efetuado com os comandos *gdal_translate* (para os ficheiros DEM e ortofoto) e *ogr2ogr* (para os ficheiros Shapfile).

5.2 Fase 2: Modelo digital do terreno

5.2.1 Divisão do terreno em tiles

O carregamento dinâmico do terreno exige o corte deste em diversos *tiles*, por forma a que apenas os *tiles* em uso estejam carregados em memória. O corte em vários *tiles* será feito sobre o ficheiro DEM. Através dos diferentes valores de elevação presentes num ficheiro DEM é possível a modelação dum terreno tridimensional fiel ao terreno real.

A divisão em *tiles* surge por questões de otimização no desempenho. Para perceber as implicações envolvidas na geração de terreno dinâmico no UE4, é necessário compreender parte da arquitetura do motor.

Os objetos que representam os vários *tiles* são *Landscape Streaming Proxies*, que herdam as propriedades de objetos atores do tipo *Landscape*.

*Landscape*s estão divididos em múltiplos *Components*(componentes) (figura 3). Os componentes permitem o cálculo de visibilidade colisões e são a unidade base de desenho do motor. Todos os componentes têm o mesmo tamanho e são sempre quadrados, sendo o tamanho destes decidido na criação do ator *Landscape*¹¹. Os dados de elevação de um componente estão armazenados numa textura quadrada de dimensões $n * n$, sendo n uma potência de dois (*power of two texture*).

Por si, um componente é composto por uma ou 4 (2x2) secções. Por fim, cada secção é então composta por *quads*. O número de vértices de uma secção deve também corresponder a uma potência de dois.

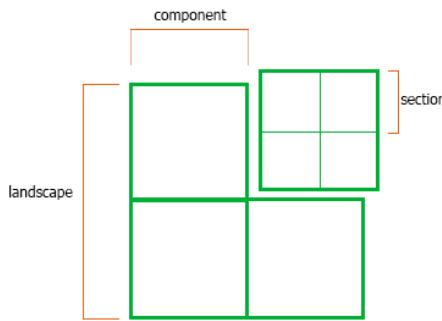


Figura 3: Hierarquia de um *Landscape* com 4 componentes e cada componente dividido em 4 secções.¹²

¹¹*Landscape Technical Guide*, visitado em <https://docs.unrealengine.com/en-us/Engine/Landscape/TechnicalGuide>

¹²Adaptado de <https://docs.unrealengine.com/en-us/Engine/Landscape/TechnicalGuide>

Tendo estas restrições, nem todas dimensões de *Landscape* são válidas e cada componente tem um custo de uma *thread* de renderização no CPU. As dimensões recomendadas por UE4 variam entre 127x127 a 1024x1024 vértices¹³. Para efeitos deste protótipo, foi considerada uma dimensão válida intermédia de 505 x 505 (sendo equivalente a 504 m x 504 m) para os *tiles* do terreno.

No entanto, as dimensões de cada *tile* deverá ser um tema explorado e estudado, sendo o ideal o teste do desempenho com *tiles* de diferentes tamanhos antes de poder tomar uma decisão definitiva sobre o tamanho ideal.

Os nomes dos *tiles* devem seguir a convenção estipulada pelo motor, ou seja, NOME_xN_yM, sendo N e M números inteiros que respeitam o sistema de coordenadas do motor (sendo que x0_y0 corresponde ao canto superior esquerdo do mundo).

O corte dos ficheiros em *tiles* é feito através de código bash script, implementado por Morgan Hite [Hite 2014], e código python. O código não se limita a realizar o corte do ficheiro, como também permite escolher quantos pixeis de sobreposição os *tiles* partilham entre si. Um pixel de uma imagem DEM, quando integrada no motor, vai dar origem a um vértice na malha do terreno. Assim sendo, a partilha de um pixel de sobreposição em *tiles* adjacentes, garante a criação de um terreno sem falhas. A figura 4 ilustra o fenómeno presenciado quando não existe um pixel de sobreposição.

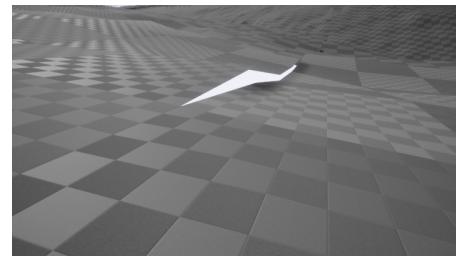


Figura 4: Falha em terreno gerado por tiles sem vértices partilhados

Foi implementado código bash script suplementar a este que trata de nomear os *tiles* criados no formato estipulado, como também a sua conversão de ficheiros .tif em PNG de 16 bits. Este código recebe todos os ficheiros .tif, convertendo em PNG de 16 bits associada a uma escala recebida como parâmetro. Esta escala deverá representar o intervalo $[-maxE; maxE]$, sendo $maxE$ o valor correspondente ao valor máximo de elevação da área.

Com sistema de World Composition do UE4, a geração do terreno e o carregamento dinâmico deste é fácil de alcançar. Utilizando a opção Import Tiled Landscape, onde são selecionados os ficheiros PNG obtidos, o terreno virtual é gerado. Neste menu (figura 5), é logo de inicio definido o valor de Z_{scale} , X_{scale} e Y_{scale} .

Os valores X_{scale} e Y_{scale} são responsáveis pela definição da largura e comprimento do terreno. Como as unidades pré-definidas do UE4 é cm (SI), os valores de X_{scale} e Y_{scale} predefinidos (100 para ambos) garantem que a distância entre dois vértices é equivalente a 1 metro no mundo virtual. Este é precisamente o valor que traduz a resolução do DEM utilizado. Para um DEM com uma resolução diferente seria necessário ajustar os valores de X_{scale} e Y_{scale} .

Como os *tiles* correspondem a ficheiros PNG de 16bits, a gama de altitudes que se pode representar varia dentro do intervalo

¹³Estes valores podem ser verificados na secção Recommended Landscape Sizes de <https://docs.unrealengine.com/en-us/Engine/Landscape/TechnicalGuide>

[-32768; 32767]. O cálculo efetuado pelo motor de modo a obter o valor da altitude no mundo virtual é:

$$h_{MV} = (h_{DEM}/128) * Z_{scale}$$

Com um valor de $Z_{scale} = 100$ é possível modelar terrenos com altitudes que variam entre [-256; 256]. Conhecendo este processo executado pelo motor, é possível otimizar o valor de Z_{scale} para uma determinada gama de altitudes que tenha os valores de elevação desejados. De modo a obter um intervalo de altitudes $[-maxE; maxE]$, o valor de Z_{scale} será obtido através de:

$$Z_{scale} = (maxE * 100 * 128/32767)$$

5.3 Fase 3: Vegetação

A aplicação da ortofoto como textura é conseguida com a criação de um *Material Landscape*. *Materials* são um tipo de *asset* do motor que está responsável por calcular como a luz interage com a superfície a qual o *Material* é aplicada. São definidos através da construção de grafos, cujos seus nós contêm código HLSL. Estes nós definidos como *Material Expressions*.

A aplicação da ortofoto é realizada com a criação de um *Material* que irá tratar do mapeamento da desta como textura consoante a posição no mundo virtual.

O nó *Texture Sample* tem a ele associado o ficheiro respetivo à ortofoto do terreno. Este é ligado ao nó *Landscape Layer Coords*, que trata de gerar as coordenadas UV necessárias para aplicar a textura em todo o terreno, consoante o *input* introduzido (figura 5). Este valor de *input* deverá ser coerente com o tamanho do terreno gerado no total.

Sendo o objetivo principal da aplicação de ortofotos o uso destas para zonas do terreno virtual a uma distância considerável, é necessário certificar que o mapeamento da textura da ortofoto apenas acontece no cenário especificado. Isto é alcançável através dos nós *Absolute World Position* e *Camera Position* no grafo do *Material* (figura 6).

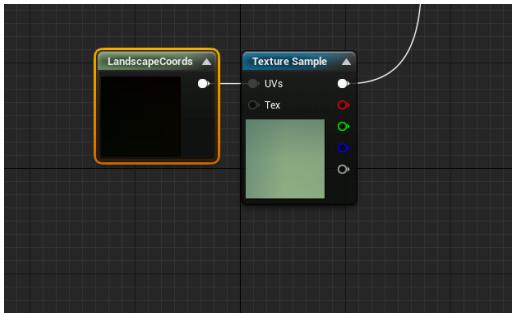


Figura 5: Nós responsáveis pela escala da textura da ortofoto

Com o grafo do Material, é também possível a aplicação da técnica de *splatting* de uma textura. Esta técnica permite adicionar um maior nível de detalhe às texturas mapeadas no terreno através da combinação de diferentes texturas de forma harmonizada.

Através da ortofoto do terreno, são obtidas máscaras a ser atribuídas ao Material: com uma máscara a preto e branco, apenas o que está a branco é pintado/povoado. Esta atribuição, em conjunto com o sistema de *Procedual Foliage* do UE4 permite o mapeamento de árvores e relvado no terreno, consoante as máscaras utilizadas.

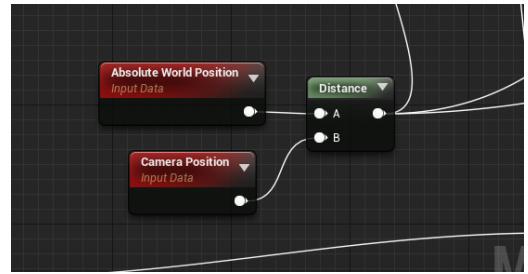


Figura 6: Nós responsáveis pelo mapeamento da ortofoto no terreno consoante a distância



(a) Sem ortofoto

(b) Com ortofoto

Figura 7: A comparação de um terreno virtual sem e com a ortofoto mapeada

Neste protótipo, as máscaras foram obtidas através da manipulação de imagens (através de programas de edição de imagem): considerou-se que uma determinada cor na imagem correspondia a um tipo de árvore (por exemplo) e todos os pixels com essa cor foram pintados a branco e os outros restantes a preto.



Figura 8: Terreno com árvores e relvado genérico com mapeamento baseado em máscaras

5.4 Fase 4: Informação cartográfica

Os ficheiros Shapefile respetivos à área de interesse contêm informação rodoviária sob a forma de linhas, sendo estas constituídas por um conjunto de pontos.

5.4.1 Biblioteca GDAL e UE4

Para integrar esta informação no mundo virtual, é possível recorrer ao uso da biblioteca GDAL¹⁴. Esta biblioteca permite uma fácil leitura e escrita de ficheiros raster e dados geo-espaciais vetoriais (como Shapefiles). GDAL dispõe de uma API para projetos em C++, logo é uma biblioteca de interesse para a criação deste protótipo.

¹⁴GDAL - Geospatial Data Abstraction Library visitado em <http://www.gdal.org>

Componentes do tipo *Splines* (SC¹⁵) permitem a construção de caminhos baseados em pontos (que representam posições) no mundo virtual. Estes caminhos poderão ser utilizados para controlar o movimento ou colocação de atores no mundo. A este tipo de componente pode ser atribuído uma malha a ser mapeada consoante os pontos definidos da *Spline* (figura 9).

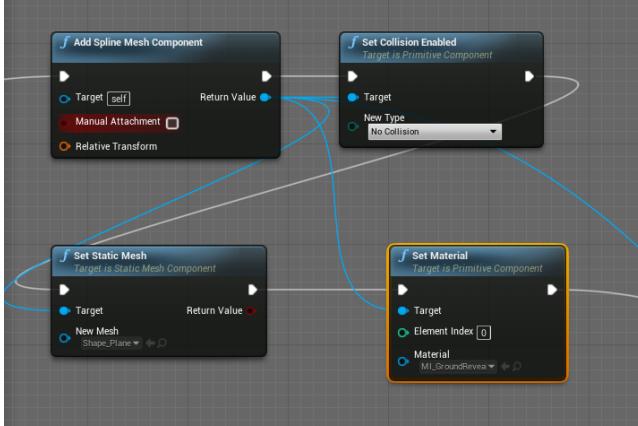


Figura 9: Nós responsáveis pela atribuição de malha(s) à *Spline*

Este mecanismo poderá ser utilizado para a construção de estradas no mundo virtual baseados em dados provenientes de Shapefiles: cada ponto em coordenadas geográficas terá um equivalente no mundo virtual, que servirá como ponto de controlo dos *Splines*.

Uma *Spline* pode ser construída através de uma classe Blueprint (BP). O sistema de BP permite uma implementação no seu código de construção (*Construction Script*), sendo assim possível a construção de uma *Spline* de um modo automatizado.

A atribuição dos pontos a uma *Spline* necessita, num primeiro passo, de ler os respetivos pontos em coordenadas geográficas. Como a biblioteca GDAL não tem uma versão específica para o sistema de BP, é necessária a implementação de uma classe C++. Esta classe deverá tratar da declaração e implementação de funções visíveis a BP, que realizem a leitura do ficheiro Shapefile em questão. Contudo, no desenvolvimento deste protótipo, até à data, não se conseguiu integrar a biblioteca GDAL no projeto UE4 onde foi implementado o protótipo.

Apesar deste contratempo, de seguida segue-se uma possível abordagem para a integração de Shapefiles com informações rodoviárias¹⁶.

5.4.2 Leitura de Shapefiles: abordagem

Um Shapefile tem mais do que uma linha, contudo, estas não deverão estar todas ligadas entre si. Deverá então existir um ciclo principal que percorra todas as linhas poligonais presentes no Shapefile. Cada iteração deste ciclo deverá seguir o seguinte protocolo:

Estrutura de dados: vetor de vetor *allCoords*. Cada casa do vetor *allCoords* armazenará um vetor (*lineCoords*) de pontos (x, y, z) em coordenadas do mundo virtual correspondentes a uma linha poligonal do ficheiro.

Ciclo: para todos os pontos presentes no ficheiro Shapefile

1. Obter as coordenadas geográficas do ponto (geo_x, geo_y)

¹⁵Spline Component

¹⁶Não se aplica a outro tipo de informação, como rios e lagos.

2. Obter as coordenadas do pixel $P(x_p, y_p)$ no ficheiro DEM associadas às coordenadas (geo_x, geo_y)
 - (a) Obter o valor de elevação Z associado ao pixel $P(x_p, y_p)$
3. Transformar as coordenadas geográficas para coordenadas do mundo virtual

$$P_{MV}(x_{MV}, y_{MV}, z_{MV})$$

=

$$P_{MV}(x_p * X_{scale}, y_p * Y_{scale}, (Z/128) * Z_{scale})$$

4. Guardar o ponto P_{MV} em *lineCoords*

Depois de percorrer todos os pontos da linha poligonal em questão, o vetor *lineCoords* deverá ser adicionado a *allCoords*.

A classe BP deverá receber o vetor retornado pela função implementada em C++ e cada ponto armazenado no vetor *allCoords* deverá ser adicionado à *Spline*.

5.5 Fase 5: Navegação no mundo

O UE4 fornece logo à partida projetos exemplo para uma melhor compreensão das diferentes funcionalidades disponíveis. Um destes projetos tem como foco principal como implementar RV no motor. Foi com base neste projeto que se introduziu a navegação em RV no terreno gerado.

A arquitetura do UE4 permite o uso de Peões (*Pawns*) para a representação física do utilizador no mundo virtual. Um Peão foi configurado de modo a receber os *inputs* do utilizador RV, como os controladores de movimento (*Motion Controllers* e HMD do equipamento RV HTC Vive).

O utilizador pode navegar pelo mundo através do seu tele-porte, sendo este conseguido com os controlos de movimento.

6 Avaliação

O principal objetivo deste trabalho era avaliar o uso de um motor de jogos de RV para a modelação digital de terreno de uma área florestal, com um nível de realismo aceitável. O modelo digital do terreno da área deveria ser de dimensão apropriada num contexto de incêndios florestais.

Os dois principais contributos deste trabalho são: uma metodologia capaz de integrar os dados reais de áreas florestais numa plataforma de jogos de RV; um protótipo demonstrativo da metodologia proposta, aplicado a um terreno com cerca de 25 km².

A metodologia indicada é independente da área selecionada e apresenta um certo nível de automatismo, apesar de não ser directamente aplicável a grandes dimensões, devido ao facto de se ter optado por fazer a junção dos ficheiros do mesmo tipo. Contudo, nunca foi o objectivo deste estudo garantir, à partida, a replicação da metodologia para grandes extensões territoriais.

É também necessário perceber os pontos onde esta metodologia poderá induzir em erro. Recorrendo a máscaras do terreno, é feita uma filtragem da área a ser mapeada com vegetação. A criação destas máscaras através de valores RGB não garante uma modelação completamente fiel aos dados reais: a presença de sombras e diferentes tonalidades de cor associados ao mesmo tipo de vegetação poderão levar a uma recriação incorreta da mesma. Com este procedimento é possível obter um mundo virtual realista mas que não represente a realidade no total.

O protótipo desenvolvido permite uma exploração interativa virtual do terreno modelado, possibilitando deste modo avaliar o realismo da visualização gerada em diferentes pontos de vista, num contexto de navegação à escala real. Apesar de ser necessário alguma aprendizagem inicial com o equipamento de RV, a navegação no mundo com a RV permite ao utilizador um bom nível de imersão e interação.

Isto posto, uma avaliação mais completa do protótipo seria realizada por participantes com experiência prévia na vigilância de terreno florestal e participantes com experiência na análise da progressão de um incêndio e área ardida. No entanto, esta avaliação encontra-se fora do âmbito deste projeto devido à necessidade de estabelecer um contacto prévio com participantes que cumprem os critérios descritos acima e de um protótipo mais completo. Seria igualmente necessário modelar territórios conhecidos pelos participantes.

7 Conclusão e Trabalho Futuro

Foi implementado um protótipo que, através do uso de um motor de jogos, permite a navegação RV num mundo virtual baseado em dados reais. Através da integração de dados GIS foi possível uma recriação de uma área previamente selecionada podendo ser a metodologia extensível para outra área distinta, desde que se tenha acesso aos dados necessários.

Foi implementado um protótipo que, através do uso de um motor de jogos, permite a navegação num mundo virtual baseado em dados reais. Através da integração de dados GIS foi possível a recriação de uma área previamente selecionada podendo ser a metodologia aplicável para outra área distinta, desde que se tenha acesso aos dados necessários. O projeto correspondente ao protótipo desenvolvido está publicado no repositório git https://bitbucket.org/st_visualanalytics/unreal-vr-gis-terrain

Com a realização do protótipo e a proposta da metodologia, conseguimos avaliar positivamente o uso da RV e de um motor de jogos para a recriação de uma grande quantidade de terreno de forma realista, permitindo uma navegação imersiva. Conseguimos igualmente identificar as limitações das tecnologias, das ferramentas utilizadas e da abordagem proposta.

Assim sendo, uma das maiores dificuldades encontradas passou pela compreensão da arquitetura do motor UE4. Mesmo com a documentação facilmente acessível, realmente compreender as diversas configurações das várias ferramentas disponíveis provou ser o maior consumo de tempo. Apesar da curva de aprendizagem se mostrar pouco apelativa, depois de se estar familiarizado com o motor e às suas facetas, a produtividade aumenta consideravelmente.

Como já mencionado na secção anterior, a metodologia deverá ser adaptada de modo a se tornar o mais automatizada, nomeadamente: - na criação de máscaras para o mapeamento da vegetação. Sugere-se uma futura implementação de um algoritmo capaz de ler ficheiros com informação respetiva ao tipo de vegetação no terreno e que com esses dados produza uma máscara para os tipos de vegetação a serem integrados no mundo virtual. - acrescentar a capacidade de lidar com áreas muito extensas, por exemplo, dividindo o terreno em regiões de 5 por 5 km (ou menores), assegurando uma gestão automática das regiões, do seu pre-processamento e do seu carregamento dinâmico. - melhorar a integração de dados de cartografia, nomeadamente, após a identificação dos diferentes tipos de elementos cartográficos a integrar considerando a motivação original de combate aos incêndios.

Outro aspeto que pode ser considerado para trabalho futuro é a avaliação experimental de um protótipo com participantes ligados

à gestão e vigilância da floresta. Para tal será interessante alargar a interações possíveis, nomeadamente a possibilidade de voar sobre o terreno modelado de maneira a ter uma perspetiva mais geral, porventura necessária para suporte à decisão em contexto de emergência.

Um grande nível de realismo pode vir a requerer a incorporação de informação da hora do dia e da meteorologia.

Referências

- ANAGH, 2011. Eucalyptus forest fire, madeira, portugal, 3 july 2011.
- DE GESTÃO DE ÁREAS PÚBLICAS E DE PROTEÇÃO FLORESTAL, D. 2017. 10.º relatório provisório de incêndios florestais - 2017 01 de janeiro a 31 de outubro. Tech. rep., Departamento de Gestão de Áreas Públicas e de Proteção Florestal, 11.
- HITE, M., 2014. Cutting large dems into square, partially overlapping tiles, Dec.
- HOLLAND, M. I., POP, S. R., AND JOHN, N. W. 2017. VR cardiovascular blood simulation as decision support for the future cyber hospital. In *2017 International Conference on Cyberworlds (CW)*, 233–236.
- JU, U., KANG, J., AND WALLRAVEN, C. 2016. Personality differences predict decision-making in an accident situation in virtual driving. In *2016 IEEE Virtual Reality (VR)*, 77–82.
- KINATEDER, M., RONCHI, E., NILSSON, D., KOBES, M., MÜLLER, M., PAULI, P., AND MÜHLBERGER, A. 2014. Virtual reality for fire evacuation research. In *2014 Federated Conference on Computer Science and Information Systems*, 313–321.
- PRASOLOVA-FORLAND, E., MOLKA-DANIELSEN, J., FOMINYKH, M., AND LAMB, K. 2017. Active learning modules for multi-professional emergency management training in virtual reality. In *2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 461–468.
- SHARMA, S., OTUNBA, S., AND HAN, J. 2011. Crowd simulation in emergency aircraft evacuation using virtual reality. In *2011 16th International Conference on Computer Games (CGAMES)*, 12–17.
- SHARMA, S., JERRIPOTHULA, S., MACKEY, S., AND SOUMARE, O. 2014. Immersive virtual reality environment of a subway evacuation on a cloud for disaster preparedness and response training. In *2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI)*, 1–6.
- SHARMA, S., DEVREAUX, P., SCRIBNER, D., GRYNOVICKI, J., AND GRAZAITIS, P. 2017. Megacity: A collaborative virtual reality environment for emergency response, training, and decision making. *Electronic Imaging 2017*, 1, 70–77.
- WEI, S., AND WEN-QI, Z. 2010. Virtual reality technology in modern medicine. In *2010 International Conference on Audio, Language and Image Processing*, 557–561.
- YI, X. 2011. The application research of virtual reality technology in emergency evacuation simulation of sports stadium. In *2011 Seventh International Conference on Natural Computation*, vol. 4, 2300–2303.

A Ambiente em Realidade Virtual



Figura 10: Navegação no mundo virtual usando o equipamento HTC Vive

B Imagens do Mundo Virtual modelado

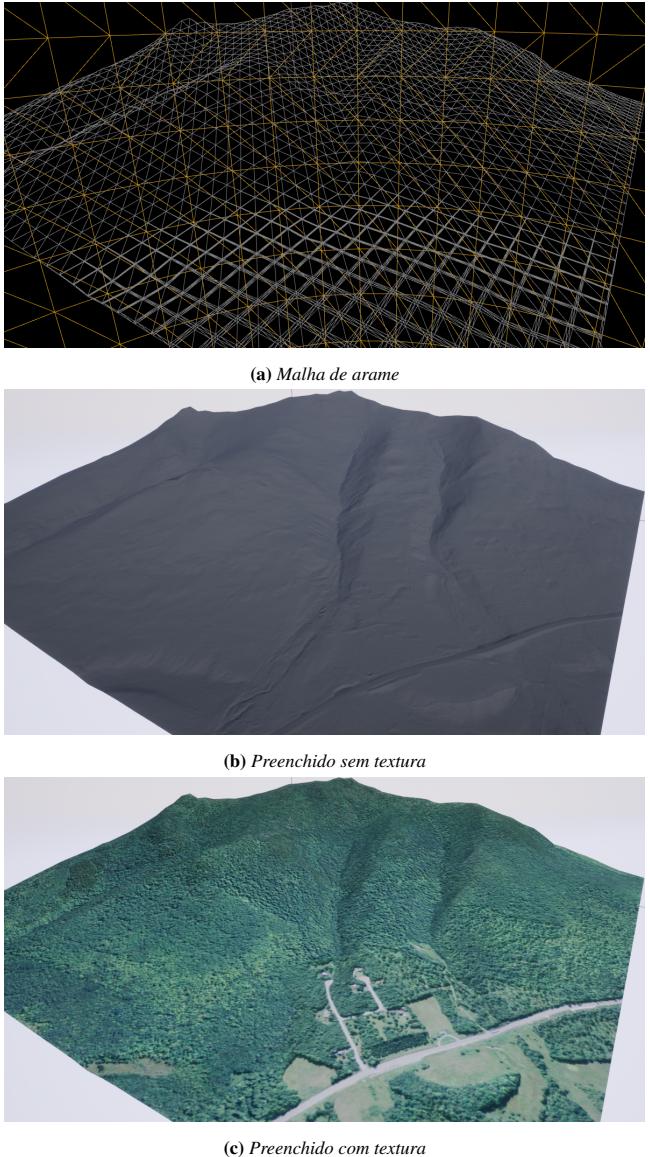
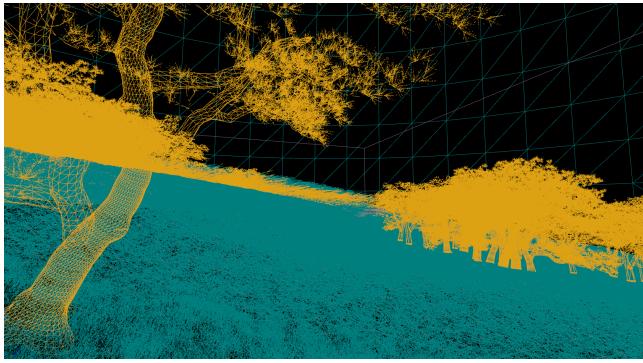


Figura 11: Diferentes modos de visualização do terreno



(a) Malha de arame



(b)

Figura 12: Terreno modelado com vegetação

C Comparação com imagens de Google Earth

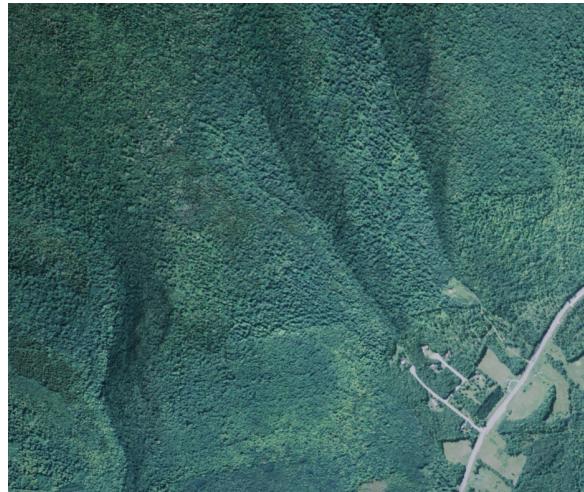


(a) Terreno modelado



(b) Google Earth

Figura 13: Comparação do terreno modelado com Google Earth: visão de perspetiva



(a) Terreno modelado



(b) Google Earth

Figura 14: Comparação do terreno modelado com Google Earth: vista de topo