

Geoblocks: geração de mundos virtuais baseados em blocos a partir de informações geográficas

Pedro Pontes, Rodrigo Rocha Gomes e Souza
Departamento de Ciência da Computação
Universidade Federal da Bahia - UFBA
Salvador, Brasil
pontescpedro@gmail.com, rodrigorgs@ufba.br

Resumo—Jogos como o Minecraft têm despertado o interesse de pessoas que querem construir mundos virtuais que reproduzem locais do mundo real. Construir esses mundos manualmente, bloco a bloco, requer muito esforço e tempo. Neste trabalho propomos o Geoblocks, um processo e um conjunto de ferramentas que permite construir mundos virtuais baseados em blocos a partir de informações geográficas. O processo foi aplicado com sucesso para reproduzir o campus de Universidade Federal da Bahia (UFBA) no jogo Minetest.

Keywords-mundos virtuais; informações geográficas; software;

I. INTRODUÇÃO

Minecraft é um dos jogos de maior sucesso comercial da atualidade, tendo vendido cerca de 176 milhões de cópias até maio de 2019. Ele permite construir e explorar mundos virtuais baseados em blocos em formato de cubo. Sua simplicidade e potencial criativo deram origem a diversas aplicações na educação, em áreas diversas como programação [1], produção literária [2] e ciências naturais [3].

Devido a sua popularidade e seu potencial, logo surgiram projetos de reconstrução de regiões do mundo real no ambiente do jogo, inclusive iniciativas governamentais [4], [5]. Essas iniciativas, no entanto, ou requerem a construção de mundos virtuais de forma manual — bloco a bloco, em um processo lento e tedioso — ou dependem de ferramentas proprietárias e caras.

Neste trabalho, propomos um processo, o Geoblocks, para sistematizar a construção de mundos virtuais a partir de informações geográficas e ferramentas disponíveis gratuitamente. Além disso, criamos novas ferramentas para automatizar algumas etapas do processo e para aumentar a produtividade das etapas manuais. Por fim, realizamos uma validação preliminar do processo através de sua aplicação na construção do *campus* da UFBA dentro do jogo Minetest, uma alternativa *open source* ao Minecraft.

II. TRABALHOS RELACIONADOS

Durante a busca por informações inicial, encontramos diversos trabalhos, isso é, iniciativas e ferramentas, relacionadas à recriação de cenários reais dentro de jogos.

Iniciativas. Uma das iniciativas foi a GB Minecraft [4], que, utilizando dados abertos de terreno da Ordnance Survey e com o auxílio de alguns estagiários construíram a região da Grã-Bretanha no Minecraft contendo mais de 22 bilhões de blocos, alegando, inclusive, a possibilidade de ser o maior mundo do Minecraft criado com dados reais de terreno. Além disso citaram, também, o potencial que o Minecraft possui de ser utilizado em escolas com o propósito de ensinar computação e geografia para alunos.

Essa iniciativa construiu um programa que utiliza, para cada região de 100 km por 100 km, dados de altura, obtidos a partir do OS Terrain 50, e dados cartográficos, utilizando imagens vetoriais do OS VectorMap District, como entradas para gerar um mundo do Minecraft. A principal diferença em relação ao nosso projeto se encontra nas fontes de dados utilizadas pela iniciativa, que no caso do GB Minecraft abrangem apenas o território da Grã-Bretanha.

Temos, também, a iniciativa chamada Blockholm [5], que se propõe a construir uma réplica exata da cidade de Estocolmo, Suécia, produzida no Minecraft sem nenhum prédio, apenas representando precisamente – na escala de 1 metro para 1 bloco – topografia, hidrografia, ruas, pontes, distritos e parques, deixando a parte de construção dos prédios para a comunidade, promovendo a criatividade. Para alcançar esse objetivo, similarmente ao nosso projeto, foram utilizados dados reais de terreno, que foram processados em uma ferramenta proprietária e paga, a FME (Safe Software) [5]. Em contraste, nosso projeto busca usar ferramentas *open source*.

Ferramentas. Uma das ferramentas encontradas foi o *geo-mapgen*¹, que se trata de um programa desenvolvido em Python que utiliza modelos de elevação digital (acrônimo em inglês: DEM), provenientes da Missão Topográfica Radar Shuttle (acrônimo em inglês: SRTM), no formato GeoTIFF, para gerar o terreno dentro do Minetest. Optamos por não utilizar essa ferramenta no projeto pela limitação de que não é possível utilizar dados de prédios e ruas com a ferramenta.

Outra ferramenta encontrada foi o Geocraft², software

¹<https://github.com/Gael-de-Sailly/geo-mapgen>

²<https://github.com/cgutteridge/geocraft>

desenvolvido em Perl que utiliza mapas urbanos do OpenStreetMap³, assim como o nosso projeto, e dados de elevação do UK LIDAR. A principal limitação da ferramenta é o fato de usar uma fonte de dados de elevação que abrange apenas o Reino Unido.

III. O PROCESSO GEOBLOCKS E SUA APLICAÇÃO

Para explicar o processo, é importante, primeiramente, definir alguns termos que serão utilizados durante as explicações. *Mundo* se refere ao espaço recriado, dentro do jogo, a partir de um modelo real; *mapa* é definido por um conjunto de dados de algum região existente; *bitmap* é um formato de arquivo de imagem que contém uma cor específica para cada pixel.

Como podemos perceber pela Figura 1, o processo de geração de um mundo a partir de mapas é realizado através de várias etapas. A primeira delas consiste em obter dados – mapas – de alguma região do planeta, contendo informações de prédios e ruas assim como de elevação. Depois, realizar a geração de *bitmaps* contendo as informações dos mapas obtidos anteriormente. Por fim, a partir dos *bitmaps* obtidos, gerar um mundo, utilizando uma ferramenta construída para algum jogo baseado em blocos.

Além disso, aplicamos o processo na construção da UFBA, utilizando apenas ferramentas gratuitas e amplamente disponíveis. Escolhemos o Minetest para ser o jogo que irá conter o mundo – por ser gratuito e *open source*, maduro, e bastante similar ao popular jogo Minecraft.

Selecionar Região. Deve-se, desde o início do processo, selecionar a região do mapa que será utilizada, de forma que não existam dúvidas no passo de extração de dados. Pensando nisso e visando abranger todos os *campi* da nossa universidade, durante a aplicação, selecionamos uma área de aproximadamente 3,5 km².

Definir escala. Dado que vários jogos baseados em blocos possuem um limite fixo de altura em seus mundos, é necessário encontrar uma escala apropriada, de forma que seja possível representar toda a dimensão do mapa dentro do mundo. Por exemplo, o Minecraft tem um limite de 256 blocos na vertical, fazendo com que seja impossível representar a Torre Eiffel – que possui aproximadamente 300 metros de altura – numa escala de 1 bloco para 1 metro, sendo necessário, então, reduzir a escala. Para o plano horizontal, em geral, os limites são muito maiores; no Minecraft, por exemplo, esse limite é de 60 milhões de blocos em cada dimensão.

Durante a aplicação, detectamos que a altura máxima do terreno da região selecionada era inferior a 256 metros. Por isso, decidimos utilizar a escala de 1 bloco para cada 1 metro em todos os eixos.

Extrair dados (mapa urbano). A partir de ferramentas adequadas, extraem-se dados de um mapa urbano, que

contém informações de prédios, ruas, pontos de ônibus, hospitais, dentre outros. Alguns exemplos de ferramentas que podem ser utilizadas para a extração de dados são o OpenStreetMap e o Google Maps.

Para a recriação da universidade utilizamos o OpenStreetMap pela maior facilidade para a extração de dados e pelo maior detalhamento dos prédios do nosso interesse. Para isso, primeiro selecionamos a região e, em seguida, exportamos os dados dessa região em um formato XML específico do OpenStreetMap, que foi utilizado posteriormente na filtragem dos dados.

Filtrar (mapa urbano). Ao extrair dados de mapas urbanos de diversas ferramentas, obtêm-se, simultaneamente, informações adicionais, tais como símbolos de ponto de ônibus, árvores, assim como nomes de edifícios e ruas. Visto que, posteriormente, precisaremos de um *bitmap*, tais símbolos não são interessantes, pois, para o exemplo de um símbolo de ponto de ônibus, ao invés de criar um ponto de ônibus no jogo, seria desenhado, na superfície daquela região, o símbolo em si, poluindo o mundo; portanto, todas as informações extras citadas devem ser removidas.

Após obtermos o mapa urbano da nossa região selecionada, usamos a ferramenta Maperitive⁴ para realizar a filtragem desses dados e conversão para algum formato gráfico.

No Maperitive, carregamos o XML do mapa urbano juntamente com um conjunto de regras que especificam como representar graficamente cada dado do XML. Como o conjunto de regras padrão não filtrava os dados da maneira desejada, criamos um conjunto de regras customizado para ignorar símbolos extras, como pontos de ônibus e árvores, e todo tipo de texto, como nomes de prédios e ruas. Também mapeamos cada elemento para uma cor específica, por exemplo, ruas para preto, prédios para cinza e grama para verde. Além disso, como as ruas são representadas por linhas sem espessura, definimos uma largura fixa para todas as ruas.

Por fim, exportamos o resultado no formato SVG, formato de ilustração vetorial, para ser utilizado na fase de alinhamento dos mapas. Optamos por um formato vetorial para possibilitar a transformação da ilustração (rotação, escala etc.) sem perda de qualidade.

Extrair dados (mapa de elevação). Definimos mapa de elevação como um mapa que especifica, para cada ponto, a altitude de tal ponto, na realidade. Um exemplo é um *bitmap* em escala de cinza onde a intensidade de cada pixel representa a altitude da área representada por esse pixel.

Para obter o mapa de altura da nossa região escolhida, utilizamos o EarthExplorer⁵, site que disponibiliza diversas informações de regiões do planeta, incluindo mapas de elevação. Por fim, selecionamos a região desejada e obtivemos as áreas, que contêm a região, no formato GeoTIFF,

³<https://www.openstreetmap.org/>

⁴<http://maperitive.net/>

⁵<https://earthexplorer.usgs.gov/>

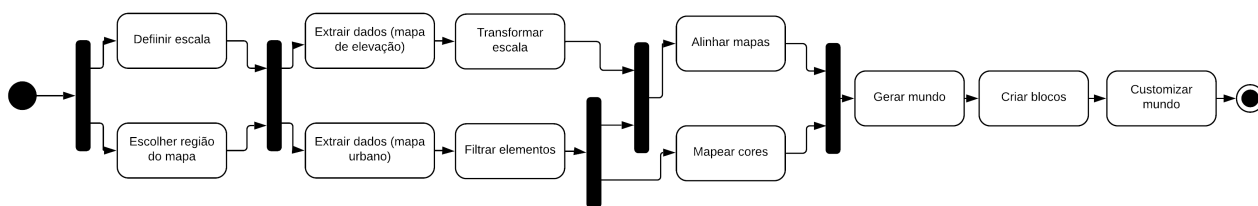


Figura 1. Diagrama do processo Geoblocks para geração de um mundo

isso é, uma forma de armazenar metadados de georreferenciamento, como projeção de mapa e sistema de coordenadas, em um arquivo TIFF, que é um formato de *bitmap*.

Transformar escala (mapa de elevação). Como dito anteriormente, muitos jogos baseados em blocos têm um limite pré-estabelecido para a altura, portanto, nesse passo, realizamos a transformação do mapa obtido para a altura que foi definida anteriormente.

Como já tínhamos conhecimento dos limites de altura da nossa região, foi preciso apenas utilizar a biblioteca GDAL⁶ (Biblioteca de Abstração de Dados Geoespaciais), mais especificamente a ferramenta `gdal-translate`, para converter os arquivos GeoTIFF obtidos anteriormente para PNG, um formato *bitmap* suportado por mais ferramentas, de forma que a variação de 1 metro no GeoTIFF correspondesse à variação de 1 unidade na luminosidade do pixel correspondente na imagem PNG.

Alinhar mapas. Em posse do *bitmap* do mapa de elevação e da ilustração vetorial (SVG) do mapa urbano, deve-se modificar as dimensões de cada um para que os dois tenham a mesma escala. Feito isso, é preciso alinhá-los, calculando suas coordenadas específicas, de forma que se sobreponham perfeitamente. Em seguida, pode ser necessário que se rotacione o mapa, para reduzir, ao máximo, o número de paredes diagonais em relação aos eixos ortogonais. Caso necessário, recortam-se os mapas para focar nas áreas de interesse. Por fim, deve-se converter ambos os mapas para arquivos de *bitmap*, pois serão utilizados posteriormente nesse formato durante a geração do mundo.

Para alinhar os mapas que obtivemos anteriormente, utilizamos a ferramenta Inkscape⁷, por trabalhar com SVGs e ser gratuito. Ao importar ambos os mapas para o programa, precisamos definir suas escalas e posições, em pixels, de um em relação ao outro. Para isso, desenvolvemos um *script* que, a partir das dimensões e coordenadas reais dos mapas, retorna as coordenadas e dimensões, em pixels, que devem ser utilizadas para alinhá-los.

Por fim, como o Inkscape não possibilitava exportar o resultado sem anti-serrilhamento – opção que evita o efeito pixelado da imagem, aproximando as cores de pixels

⁶<https://gdal.org/>

⁷<https://inkscape.org/>



Figura 2. PAF I - UFBA

próximos, o que atrapalha o mapeamento de cores, já que utilizamos um bloco para cada cor de pixel – utilizamos o editor gráfico GIMP⁸ apenas para finalizar o processo com a exportação.

Mapear cores. Após gerar o *bitmap* do mapa cartográfico, é necessário realizar um mapeamento de cores para blocos específicos do jogo. Assim, ao gerar o mundo, cada cor será transformada em um bloco específico ou um bloco padrão, caso a cor não seja mapeada. Também é possível definir uma cor específica para construções e árvores, isso é, para áreas do *bitmap* que possuam a cor atribuída a construções, será produzida uma estrutura com altura predefinida, simulando uma edificação; e, para árvores, de forma semelhante, para cada pixel do *bitmap* com a cor definida anteriormente, uma árvore será colocada na posição do pixel. Vale notar que na aplicação realizamos este processo juntamente com a filtragem do mapa urbano.

Gerar mundo. Como a criação de um mundo para um determinado jogo baseado em blocos é uma tarefa bem específica, é trabalho do usuário do processo construir tal ferramenta. Em posse de ambos os mapas, urbano e de elevação, assim como do mapeamento de cores para blocos, utiliza-se a ferramenta de criação construída anteriormente, fornecendo os arquivos *bitmap* e de mapeamento de cores para blocos do jogo, para gerar um mundo para o jogo escolhido.

Seguindo o processo, desenvolvemos a ferramenta de

⁸<https://www.gimp.org/>

geração utilizando o J2Blocks⁹ – biblioteca em Java que auxilia na criação de mundos do Minecraft –, por ter sido a biblioteca com maior documentação e facilidade de uso mesmo com a limitação de gerar mundos apenas para Minecraft. Dada tal limitação, foi necessário utilizar uma ferramenta para realizar a conversão para o Minetest; no nosso caso, utilizamos o mcimport¹⁰.

Nossa ferramenta, como descrito no processo, recebe os mapas e o mapeamento de cores e gera o mundo com terreno e altura especificada, além do adicional de prédios e árvores gerados automaticamente nas regiões com cores definidas como de prédio e árvore, respectivamente.

Criar blocos. Após gerar o mundo, durante a construção manual, pode ser necessário utilizar um bloco que não existe no jogo base. Diante de tal necessidade, a obtenção de blocos extras é de grande importância para a customização do jogo. É possível obtê-los de alguma fonte conhecida, isso é, criados por outras pessoas, ou criados pelo próprio usuário.

Durante a aplicação, utilizamos o GIMP para criar novas texturas – para blocos simples, como concreto e pilares – e o Node Box Editor¹¹, editor de blocos para Minetest, para criar blocos com formas irregulares, como grades e corrimão de escadas.

Customizar mundo. Após gerar o mundo, mesmo com grande parte construída automaticamente, ainda restam muitos detalhes a serem criados. Muitos desses detalhes, principalmente os mais específicos, como a decoração de espaços dentro do jogo, podem ser construídos de forma manual, enquanto criar um novo prédio pode ser facilmente realizado utilizando algum *mod* – isso é, uma modificação do jogo feita por algum usuário –, evitando que o jogador coloque, manualmente, uma grande quantidade de blocos.

Pensando nisso, durante a aplicação do processo, desenvolvemos ferramentas e *mods* para criar e aumentar o tamanho de prédios, pilares e paredes, para pintar faixas horizontais e verticais, e para corrigir erros utilizando uma ferramenta de *undo*. Além de utilizar *mods* já existentes como o WorldEdit¹² e o MTPaint¹³.

Por fim, disponibilizamos o mundo, juntamente com as ferramentas, *mods*¹⁴ e blocos criados, em um servidor. Convidamos pessoas para acessar o servidor, jogar e customizar o mundo. Com o *feedback* recebido, pudemos corrigir problemas e aprimorar os *mods* de construção.

IV. CONCLUSÃO

Durante a realização do projeto, encontramos diversas ferramentas para extrair e manipular dados urbanos e de

elevação de regiões do planeta, assim como pudemos criar uma ferramenta, utilizando bibliotecas já existentes, para, a partir desses dados, recriar de forma automática tais regiões dentro do jogo Minetest.

O desenvolvimento desse processo proporciona uma forma de automatizar trabalhos extensos que demandariam ações repetitivas ou tediosas, como construir alguma cidade colocando um bloco por vez no mundo, por exemplo. O uso desses recursos, por sua vez, possibilita que o usuário possa utilizar o seu tempo para trabalhos mais refinados, como a decoração e customização de algum ambiente dentro do jogo, manualmente ou com o uso dos *mods* e ferramentas desenvolvidos durante o projeto. Esses recursos são independentes do processo, ou seja, podem ser utilizados por qualquer pessoa que necessite.

Foi possível, também, ao utilizar o processo de criação de mundos, recriar UFBA no Minetest – resultado observado na Figura 2 –, disponibilizando para os estudantes, principalmente calouros, uma forma simples e divertida de explorar e conhecer a universidade, fazendo com que eles não se percam por não conhecer os caminhos que devem seguir para chegar aos seus destinos.

Para o futuro pensamos em continuar e refinar o trabalho, disponibilizando o servidor para que todos possam acessar. Além disso, pretendemos criar novas ferramentas e *mods* para o jogo, visando facilitar, sempre que possível, a construção. E, também, desejamos construir um jogo de exploração, baseado em missões dentro do jogo, para tornar a experiência do jogador ainda mais interessante e divertida.

Por fim, pensamos na possibilidade de utilizar da ferramenta e do processo para construir outras regiões do mundo, como, por exemplo, pontos turísticos e outros *campi* universitários.

REFERÊNCIAS

- [1] J. A. L. da Silva, F. C. S. Oliveira, and D. J. S. Martins, “Storytelling e gamificação como estratégia de motivação no ensino de programação com python e minecraft,” in *Proceedings of SBGames 2017*, 2017.
- [2] R. Custodio and E. Pozzebon, “Minecraft: um jogo? um mundo? uma estratégia de ensino?” in *Proceedings of SBGames 2016*, 2016.
- [3] T. Knittel, L. Santana, M. Pereira, and M. Menuzzi, “Minecraft: Experiências de sucesso dentro e fora da sala de aula,” in *Proceedings of SBGames 2017*, 2017.
- [4] Ordnance Survey, “Minecrafting with OS OpenData,” 2013. [Online]. Available: <https://www.ordnancesurvey.co.uk/innovate/developers/minecraft-map-britain.html>
- [5] Ulf Månsson, “The Real World and Minecraft Meet in Blockholm, Sweden,” 2013. [Online]. Available: <https://www.safe.com/blog/2013/10/real-world-minecraft-meet-blockholm-sweden/>

⁹<https://github.com/MorbZ/J2Blocks>

¹⁰<https://github.com/minetest-tools/mcimport>

¹¹<https://rubenwardy.com/NodeBoxEditor/>

¹²<https://github.com/Uberi/Minetest-WorldEdit>

¹³<https://forum.minetest.net/viewtopic.php?id=5674>

¹⁴<https://github.com/mineufba/minetest-mods>