

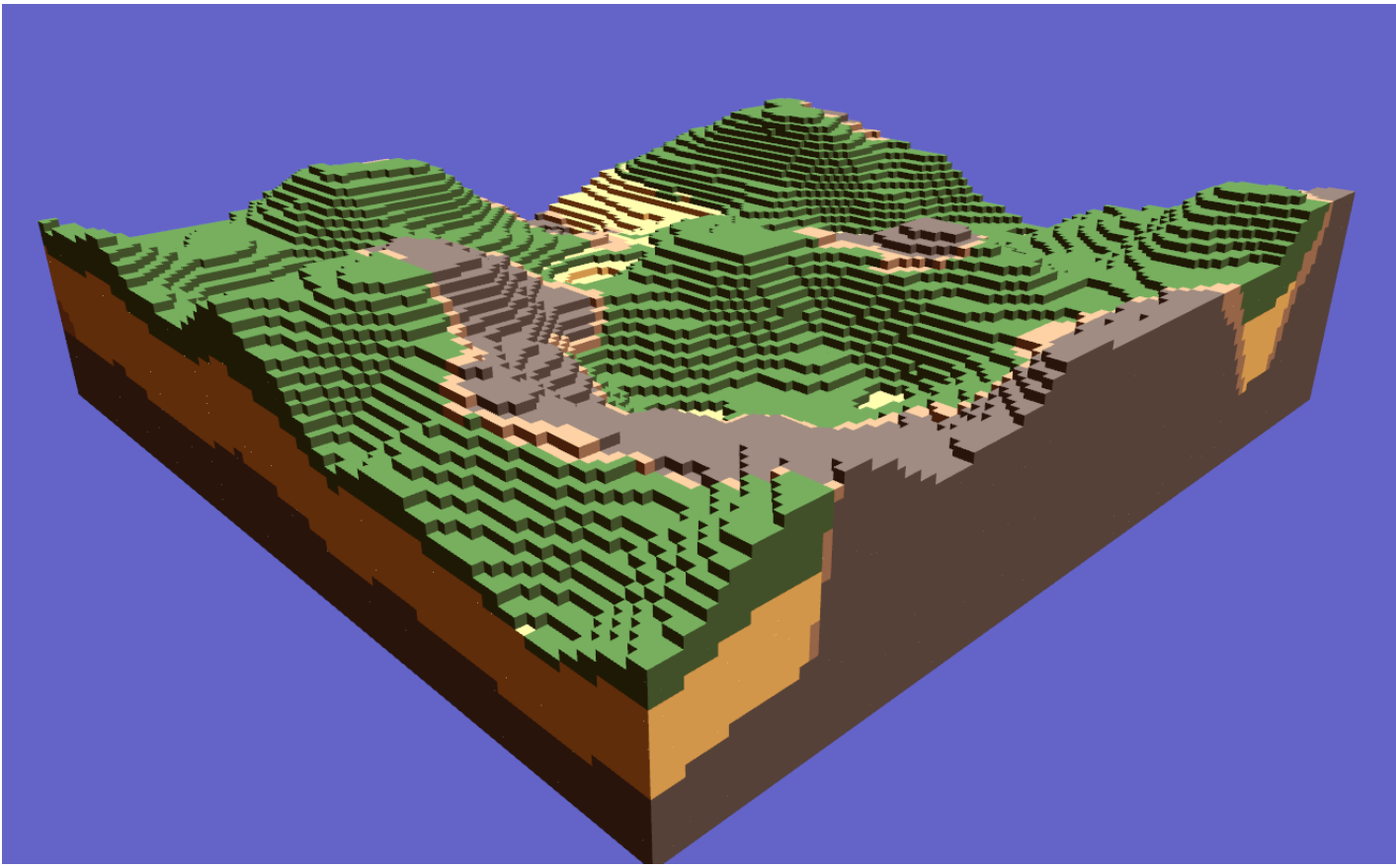
Voxel-based environments - 35 pontos

Dando continuidade ao trabalho anterior, nesta etapa expandiremos nosso ambiente baseado em voxels com modelagem procedural e câmera em terceira pessoa.

Geração Procedural

Nesta etapa o ambiente será criado por [geração procedural](#). Existem vários algoritmos que podem ser utilizados para este fim como ilustrado [neste link](#). Contudo, diferente do que é demonstrado no link, seguiremos com a construção do ambiente através de voxels. A altura máxima das estruturas que compõem o ambiente será de 20 unidades. Observe na imagem abaixo um exemplo de ambiente gerado com este tipo de técnica. As cores do ambiente gerado podem utilizar essa imagem como base. O grupo deve pesquisar por algoritmos clássicos ou empregar [uma função de ruído apropriada](#) para gerar o ambiente.

Nota: no momento da apresentação é importante que os grupos indiquem qual algoritmo utilizaram e porquê.



O ambiente deve ser suficientemente amplo para que seja possível andar por um certo tempo sem chegar à sua borda. Para evitar problemas de desempenho, empregaremos um sistema de névoa (*fog*) para que apenas uma parte do ambiente seja visível. Um slider para aumentar/diminuir a distância do efeito de fog deve ser incluído (como [neste exemplo](#)). No valor máximo, deve ser possível ver todo o ambiente. Deve ser incluído também um contador de FPS (*frames per second*) na interface do ambiente (exemplo [aqui](#)).

As árvores criadas no T1 devem ser inseridas aleatoriamente no ambiente gerado, levando-se em consideração a altura do bloco onde cada árvore será inserida. Informações sobre os materiais a serem utilizados no terreno e árvores serão detalhadas mais adiante. Deve-se incluir no mínimo 20 árvores.

Nota: o ambiente de modelagem será utilizado novamente no último trabalho.

Personagem em terceira pessoa

Neste trabalho utilizaremos um dos personagens do Minecraft em uma visão em terceira pessoa (caso alguém não esteja familiarizado com esse tipo de visão, veja exemplos [aqui](#)). Os grupos poderão baixar *assets* para este fim ou utilizar [este modelo](#). Os movimentos a serem implementados serão os de andar e pular. Importante ressaltar que o modelo disponibilizado tem animação de andar embutida. Se o grupo optar por baixar seu próprio modelo, o mesmo deve ter no mínimo esse tipo de animação.

Em relação à jogabilidade e interação, a movimentação será semelhante a do primeiro trabalho, com rotação da câmera feita através do mouse e movimentação via teclado. Deve-se mapear tanto as teclas WASD quanto as setas do teclado para fazer a movimentação. Para pular, deve-se mapear a tecla espaço e o botão direito do mouse (ambos devem funcionar). Na rotação, a orientação em Y pode ser a convencional ou a invertida. Para modificar entre os dois modos, deve-se pressionar a tecla 'Y'.

Novamente teremos uma câmera de inspeção e uma câmera em terceira pessoa, sendo alternadas ao pressionar a tecla 'C'.

Colisão

Para interagir com o ambiente um sistema de colisão deve ser implementado. A ideia básica por trás de um sistema de colisão em *threejs* pode ser analisada por [este exemplo](#), mas considerando a natureza do sistema proposto, é possível implementar sistemas *ad hoc* mais eficientes.

Essencialmente o personagem deve colidir com todos os blocos do ambiente, podendo subir nos níveis mais altos do terreno através de saltos. Ao descer de um bloco, o personagem deve deslocar para o primeiro bloco abaixo. Toda a movimentação deve ser suave, conforme ocorre em qualquer jogo semelhante ao que está sendo implementado. Deve ser possível, por exemplo, pular de um bloco mais alto e cair em um bloco mais baixo. O sistema de colisão deve impedir que o personagem atravessasse um bloco. Neste caso, ao colidir com um bloco sem pular, o personagem deve deslizar pelo bloco, mantendo a animação de movimento.

Iluminação e materiais

Para este trabalho teremos uma luz direcional principal que projetará sombra no ambiente. Para uma melhor qualidade visual, deve-se utilizar uma luz ambiente ou criar uma segunda luz direcional de menor intensidade posicionada na direção oposta à principal, que não projetará sombras, para melhor iluminar o ambiente. A luz deve estar posicionada de forma a projetar uma sombra não muito alongada (considere um Sol às 11h da manhã no verão). Todos os itens (personagem, blocos e árvores) devem projetar sombra sobre os blocos que compõem o ambiente.

O *target* da(s) luz(es) deve ser o personagem. Desta forma, a luz acompanhará o personagem ao longo do ambiente, respeitando a movimentação (translação), mas não a rotação. Deve-se criar um helper para que o volume de visualização da sombra seja analisado (deve-se pressionar 'H' para habilitar/desabilitar este helper). O comportamento esperado é que as sombras sejam projetadas sempre na mesma direção. Deve-se utilizar um tamanho adequado de mapa de sombra para que tenhamos qualidade com um bom desempenho. Uma nota importante é que a área do volume da câmera virtual para geração de sombras deve aumentar/diminuir junto com a distância escolhida para o fog. Com isso teremos sempre o menor volume possível com todos os elementos projetando sombra.

Para todos os materiais, deve-se usar LambertMaterial.

Considere para efeito de avaliação os seguintes critérios de pontuação geral:

Grupo	Item	Pontos
Geração procedural (15 pontos)	Criação do ambiente por geração procedural Ambiente será avaliado pela qualidade visual e atendimento dos requisitos (cor, altura máxima etc) <i>Nota: os três itens abaixo (fog, árvores e fps) só serão avaliados se ESTE ITEM for minimamente implementado</i>	9
	Fog Criação do efeito de fog, com slider para controlar o efeito	3
	Inclusão de árvores Árvores devem ser incluídas em posições aleatórias respeitando a altura onde será inserida (mínimo de 20)	2
	Inclusão do contador de frame Contador de frame deve ser incluído e funcionar corretamente	1
Personagem em terceira pessoa (7 pontos)	Inclusão do personagem em terceira pessoa Personagem deve ser corretamente incluído, considerando a escala de todo ambiente <i>Nota: os dois itens abaixo só serão avaliados se ESTE ITEM for implementado</i>	3
	Animação do personagem ao andar/pular Ao andar/pular, uma animação correspondente será utilizada. O movimento pode ser o mesmo para as duas ações	2
	Mapeamento de mouse/teclado Mouse e teclado devem ser corretamente mapeados conforme descrito, incluindo a possibilidade de inverter o Y	2
Sistema de Colisão (6 pontos) <i>* este item, só será avaliado se a geração procedural e personagem em 3ª pessoa estiverem implementados</i>	Sistema base de colisão Análise do item será feita pela análise da acurácia do sistema de colisão	3
	Correta movimentação ao subir ou descer dos blocos A análise do item será realizada pela suavidade e acurácia dos movimentos ao subir/descer os blocos.	3
Iluminação e materiais (7 pontos) <i>* este item, só será integralmente avaliado se a geração procedural e personagem em 3ª pessoa estiverem implementados</i>	Inclusão do sistema de iluminação com mapeamento de sombras Inclusão de uma luz direcional e luzes acessórias. Item será avaliado considerando a qualidade visual do resultado	2
	Luzes devem seguir o personagem Além da qualidade do sistema de iluminação, deve haver um helper para analisar o volume da câmera do <i>ShadowMap</i>	3
	Volume do <i>ShadowMap</i> deve ser alterado junto com a distância do fog A análise do item será visual e através do helper do <i>ShadowMap</i> .	2
Nota 1: Informações adicionais e/ou correções a este enunciado podem ser adicionadas na forma de comentários no Google Meet.		
Nota 2: O trabalho pode ter uma penalização de até 30% do total se forem encontrados problemas de usabilidade não mapeados na tabela acima.		
Nota 3: Itens essenciais do trabalho anterior que prejudiquem a utilização deste trabalho podem impactar em sua avaliação (isto é, você pode perder pontos por pendências do T1 que ainda façam sentido neste trabalho)		
Nota 4: O envio incorreto dos arquivos (falta de arquivo, arquivo incorreto etc) ou problemas gerais na chamada dos <i>assets</i> (caminhos ou referências incorretas) podem acarretar em uma penalização de até 10% da nota total. Lembrem-se que o linux é <i>case sensitive</i> .		
Nota 5: Se forem identificadas cópias parciais ou totais de código, a nota será dividida pelos grupos (exemplo: para um trabalho cujo grupo tenha tirado 24 pontos, se identificada a cópia, cada grupo envolvido ficará com 12 pontos. Se forem três grupos envolvidos, serão 8 pontos para cada grupo e assim por diante).		

Foco na apresentação

Um dos aspectos mais importantes da implementação é a questão da clareza do código. O projeto deve ser minimamente **modelado** antes de ser implementado. O grupo será questionado a respeito de detalhes do código e a avaliação será individual.

Qualquer componente do grupo poderá ser questionado por qualquer parte da implementação, mas é importante que cada um esteja devidamente preparado para apresentar um tópico específico.

Nota importante: no [mesmo nível](#) da pasta *examples* do nosso repositório, crie uma pasta **T2** e desenvolva seus códigos nesta pasta. Para enviar seu trabalho, compacte esta pasta (*zip*, *rar* etc) e envie via Google Classroom. Os *assets* utilizados devem estar dentro desta pasta. **TESTE SEU SISTEMA NO LABORATÓRIO ou EM ALGUMA MÁQUINA LINUX ANTES de enviá-lo.**

Prazo para envio do trabalho:	02/02* (domingo - até 23:59)
Datas de apresentação do trabalho:	03/02 (segunda) OU 06/02 (quinta)

** Será aplicado um desconto de 10% na nota final para cada hora de atraso na entrega.*