

Rail shooter - 35 pontos

Dando continuidade ao trabalho anterior, nesta etapa focaremos em aprimoramentos visuais e novos recursos de jogabilidade do sistema.

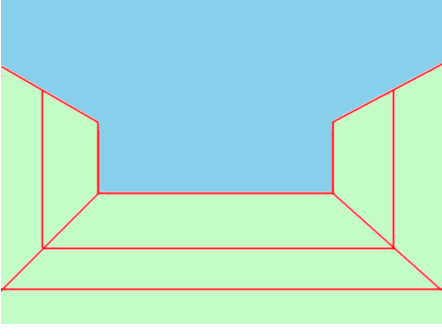
Iluminação

O grupo deve criar um sistema de iluminação direcional corretamente posicionado de forma a fazer com que todos os objetos presentes no sistema sejam corretamente iluminados com suas respectivas sombras projetadas. A depender da implementação, a luz tem que acompanhar o movimento da câmera (desnecessário se o grupo optou por mover o chão ao invés do avião). A luz estará posicionada em X e Y positivo fazendo com que a sombra seja projetada à esquerda do avião (darei detalhes em sala).

O grupo deve se preocupar em fazer o correto balanço entre a qualidade das sombras projetadas sobre o chão e o desempenho geral do sistema. Deve-se encontrar uma resolução de sombra e um volume de projeção (da câmera virtual necessária) que garanta isso. Para análise de desempenho, nosso ambiente de teste serão as máquinas do laboratório.

Ambiente

Neste trabalho o ambiente será um pouco modificado pois ao invés de estarmos em um plano simples, estaremos em um ambiente semelhante ao da imagem ao lado (cores e quantidade de planos visíveis meramente ilustrativas). As sombras dos objetos que comporão o ambiente serão projetadas **tanto no plano inferior quanto nas laterais**.



A lateral do ambiente será formada por cubos e todos os cubos devem ter cor e borda (arestas) visíveis. A lógica para a renderização dos cubos é a mesma dos planos do T1, isto é, o mesmo efeito de *fade* utilizado nos planos será também utilizado nas laterais. Vocês poderão, inclusive, substituir o plano base por cubos e não há problema se as arestas “internas” dos cubos ficarem visíveis (internamente cubos são criados com faces triangulares como pode ser visto [aqui](#)).
Nota: há várias formas de visualizar a aresta de uma geometria. Vale pesquisar estas formas antes de tentar implementar uma solução própria.

Movimentação do avião e da câmera

Nesta versão a movimentação do avião será via *Raycaster*. Veja em nosso ambiente online os novos exemplos que disponibilizei. Dos exemplos disponíveis, o mais interessante para este trabalho é o *RaycasterLayers*, uma vez que a implementação a ser desenvolvida está baseada na interseção do mouse com um plano invisível que ficará à frente. Considerando que o avião também está se movimentando em um plano, a ideia é alinhar a posição XY do avião com a posição XY capturada no plano invisível via Raycaster e interpolar o movimento de um ponto a outro.

O cursor do mouse será substituído por um target semelhante ao visto [aqui](#). O grupo deve pesquisar como ocultar o cursor durante a execução da simulação. Esse target terá a mesma coordenada Z do plano invisível descrito no parágrafo anterior. A posição relativa do target em relação ao avião (e por consequência, do plano invisível) deve ser feita por semelhança em relação [ao vídeo de exemplo](#). Não é necessário implementar o target duplo ilustrado no vídeo.

Nesta versão do trabalho o avião deverá rotacionar em X, Y e Z ao movimentar de um ponto A para um ponto B e todas as movimentações devem ser suaves, semelhantes àquelas vistas nos vídeos apresentados no trabalho anterior ([\[1\]](#) e [\[2\]](#)). As bordas laterais servirão como limitadores da movimentação do avião, mesmo que a altura do mesmo seja superior à altura da borda.

Além dos movimentos supracitados, a câmera também terá pequenos movimentos no plano XY quando o avião estiver mais próximo das bordas da tela. Tente entender como esse movimento é feito analisando o início [deste vídeo](#). A avaliação deste item será feita por semelhança.

Importação de objetos

Para esta versão teremos dois elementos *importados*, o avião e as [torretas](#) (*turrets*), que comporão nosso ambiente em conjunto com as árvores do T1. Em relação às torretas, teremos no máximo 3 visíveis simultaneamente durante a execução da simulação. As torretas não precisam ser criadas aleatoriamente e o grupo deve ter o cuidado de não posicioná-las sobre as árvores, isto é, não deve haver interseção entre as torretas e as árvores do ambiente.

Tanto avião e *torretas* podem ou não ter texturas (esse tema será tratado em detalhes no T3).
Nota: Não utilize o avião disponível em nosso repositório.

Jogabilidade e interação

Em relação a jogabilidade do sistema, teremos algumas adições importantes.

Deve ser possível **ATIRAR** com o avião nas *torretas* criadas anteriormente. O tiro deve ser disparado na direção apontada pelo target. Nesta versão do trabalho as torretas não atirarão de volta. Ao ser atingida, a torreta deve ter algum tipo de animação (mudança de opacidade, diminuição de escala etc) antes de desaparecer. Neste item, o grupo deve se preocupar com formas de disparar a partir do avião e estudar também como tratar a colisão do tiro com a torreta. O tiro pode ter qualquer geometria, mas paralelepípedos, cilindros e esferas são objetos que facilitarão a colisão. Se o tiro chegar em uma região não visível (bateu no chão, nas paredes ou em uma região transparente no fundo) ele deve ser removido do sistema. O tiro será realizado clicando com o botão esquerdo do mouse.
Dica 1: Criar uma estrutura de dados para gerir os tiros disparados na cena facilitará a implementação deste item.
Dica 2: há em nosso repositório exemplos de colisão. Vale investigar antes de tentar implementar outras soluções.

O grupo deve mapear a tecla ESC para pausar a simulação. Ao fazer isso, o cursor do mouse voltará a aparecer. Clicando na tela do sistema novamente a execução deve ser retomada de onde parou.

O sistema deve ter 3 modos de velocidade e os modos devem ser modificados ao pressionar as teclas 1, 2 ou 3. As mudanças nas três velocidades devem ser claramente perceptíveis e coerentes com a proposta do sistema.

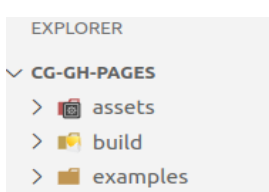
Considere para efeito de avaliação os seguintes critérios de pontuação geral:

| Grupo | Item | Pontos |
|--|--|--------|
| Iluminação (xx pontos) | Luz Direcional com sombras Atentar para o balanço entre qualidade e desempenho e a movimentação da fonte de luz (se necessário) | |
| | Correto posicionamento da luz direcional para geração de sombras Todos os objetos presentes na cena devem projetar sombras | |
| Ambiente (xx pontos) | Novo ambiente composto por planos e cubos na lateral Atentar para a forma como os objetos aparecerão na cena (com <i>fade</i>) e na projeção de sombras tanto nos planos quanto nas laterais. | |
| | Visualização das arestas dos cubos Cubos das laterais tem ter arestas visíveis | |
| Movimentação do avião e da câmera (xx pontos) | Implementação da movimentação via Raycaster Avaliação será feita através da qualidade do alinhamento entre o ponto de interseção com o plano invisível e o objeto | |
| | Substituição do cursor do mouse pelo target O cursor do mouse deve ficar oculto enquanto o target estiver visível. A forma do target deve ser semelhante à apresentada no vídeo de exemplo e a avaliação será por semelhança com este vídeo. | |
| | Rotação do avião em todos os eixos Ao movimentar o avião pela cena, deve-se fazer a rotação adequada em todos os eixos. A avaliação será por semelhança com o vídeo de exemplo. As laterais da cena servirão como delimitador do movimento do avião. | |
| | Movimentação da câmera A câmera terá pequenos movimentos quando o avião estiver próximo às bordas, com avaliação por semelhança com o vídeo de exemplo. | |
| Importação de objetos (xx pontos) | Importação do Avião O objeto que representará o avião será importado e deve ser corretamente posicionado/escalado na cena. | |
| | Importação das torretas As torretas serão importadas e devem existir no máximo 3 torretas simultâneas durante a execução do sistema. Os objetos devem ser corretamente escalados e posicionados na cena. | |
| Jogabilidade e interação (xx pontos) | Sistema de tiro do avião Avião deve atirar projéteis contra as torretas ao clicarmos com o botão esquerdo do mouse na direção indicada pelo target. Tiros que não estão visíveis (ver descrição) devem ser removidos da cena. Os tiros devem ter sua origem em alguma parte do avião. | |
| | Colisão dos tiros contra as torretas e animação Deve-se criar um eficiente sistema de colisão quando uma torreta for atingida por um tiro. Ao ser atingida, a torreta deve sofrer uma animação antes de desaparecer. | |
| | Mapeamento da tecla ESC A tecla ESC deve ser utilizada para pausar a simulação, voltando com o cursor do mouse. Clicando novamente na simulação o target deve voltar a aparecer. | |
| | Modos de velocidade A mudança nas três velocidades devem ser claramente perceptíveis e coerentes com a proposta do sistema e deve ser realizada ao pressionarmos as teclas 1, 2 e 3 do teclado. | |
| 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 impactem na utilização deste trabalho podem impactar em sua avaliação (isto é, você pode perder pontos por pendências do T1) | | |
| 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. **TESTE SEU SISTEMA NO LINUX ANTES** de enviá-lo.

| | |
|------------------------------------|--|
| Prazo para envio do trabalho: | <u>20/05</u> * (sábado - até 23:59) |
| Datas de apresentação do trabalho: | <u>22/05</u> (segunda) ou <u>26/05</u> (sexta) |

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