

Rail shooter - 30 pontos

Este trabalho consiste na criação de um protótipo de jogo tipo *rail shooter*. A inspiração serão os jogos tipo *plane shooters* desta categoria como os clássicos *Star Fox* e o *Star wars Rogue Squadron*. Nesta primeira etapa nos preocuparemos com os elementos básicos do protótipo como posicionamento de câmera em relação ao avião e a movimentação e modelagem da primeira versão do avião.

Movimentação e câmera

A posição da câmera e sua movimentação básica será semelhante à maioria dos jogos desta categoria. Considere como exemplo os jogos citados anteriormente ([1] e [2]). Repare que a câmera tem um movimento contínuo, sempre para frente, e o movimento do avião é feito indiretamente, produzido através da movimentação do alvo (*target*).

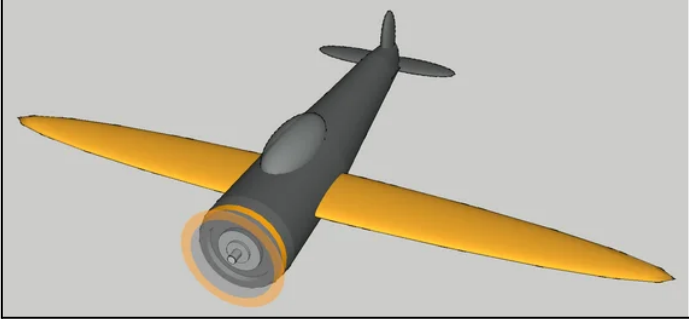
A movimentação do alvo (que por enquanto é tão somente o cursor do mouse) será realizada com o mouse, **sem a necessidade de clicar em qualquer um dos botões**. Como visto nos vídeos de exemplo e considerando o sistema de coordenadas visto em aula, o alvo será movido no plano XY em relação ao avião.

A movimentação do avião se dará no eixo Z. Neste primeiro trabalho será necessário implementar apenas a rotação em Z ao movimentar o avião. Demais rotações serão exploradas nos trabalhos seguintes, mas nada impede que os grupos já as implementem de uma vez essas rotações.

Dica: Neste vídeo vocês encontrarão algumas ideias que podem ajudá-los a modelar esta parte do trabalho.

Modelagem do avião

Utilizaremos por enquanto apenas geometrias básicas para criar o avião. Nos demais trabalhos utilizaremos outros objetos/modelos mais elaborados.



O avião nesta etapa será representado por um conjunto de primitivas básicas agrupadas com transformações geométricas adequadas, gerando um resultado que, a nível de complexidade, seja semelhante ao ilustrado ao lado.

Modelagem do ambiente

O ambiente será modelado com a utilização de um plano retangular composto por um número **aleatório** de árvores representadas por "objetos compostos" como o ilustrado ao lado. A ideia é que sejam utilizados múltiplos planos que serão alternados de acordo com a movimentação do avião (darei detalhes na apresentação em sala desta parte do trabalho). Cada novo plano receberá um novo número aleatório de árvores.



Para criarmos nossos planos retangulares que comporão o ambiente será necessário desenvolver um plano que exiba sua malha interna (*wireframe*). Utilize como base a função *createGroundPlaneWired* (disponível em *../Libs/util/util.js*) para criar estes planos. Ele será especialmente útil para identificarmos o movimento do avião e da câmera. Lembre-se que o sistema será executado indefinidamente, e não é possível criar um plano de tamanho infinito. Pensem em uma forma de reaproveitar um plano de tamanho mediano durante a execução do sistema.

O grupo deverá implementar um esquema de *fade*, no qual objetos mais distantes (árvores e planos distantes) não serão vistos e, à medida que forem se aproximando do avião, começam a aparecer gradativamente. A forma mais fácil de fazer isso é através da propriedade de transparência/opacidade dos objetos. O grupo deve pesquisar como isso é feito em nossa biblioteca.

Nota: Vale ressaltar que a movimentação proposta pode ser interpretada de duas formas distintas mas que produzirão o mesmo resultado. Na primeira forma, o avião estaria parado em Z, movendo apenas no plano XY. Neste caso, todos os demais elementos se movem. Na segunda forma, todos os elementos são criados em suas posições definitivas e apenas o avião (e por consequência a câmera) moverá em todos os eixos.

Outros

Para esta versão, utilize como material do avião e demais objetos o comando *setDefaultMaterial*, passando as cores como parâmetro. Para iluminar o ambiente, utilize o comando *initDefaultBasicLight(scene)*. O sistema definitivo de iluminação da simulação será definido em detalhes nos próximos trabalhos. Essas duas funções foram utilizadas em nosso primeiro exemplo (*basicScene.js*)

Considere para efeito de avaliação os seguintes critérios de pontuação geral (30 pontos no total):

Grupo	Item	Pontos
Movimentação e câmera (12 pontos)	Câmera Virtual - Criação e movimentação Correta criação e posicionamento da câmera. Movimentação deve ser constante, sempre para frente.	2
	Interação via mouse A movimentação do avião será realizada com o mouse, sem a necessidade de acionar qualquer um dos botões.	4
	Movimentação do avião O avião deve se mover no plano XY e a velocidade deve ser constante. Deve ser implementada apenas a rotação em Z (isto é, rotações laterais). A avaliação será por semelhança com a movimentação dos aviões mostradas nos vídeos de exemplo.	6
Modelagem do avião (5 pontos)	Modelagem do avião O avião será modelado com geometrias básicas devidamente agrupadas. Utilize um mínimo de 7 geometrias básicas para construir seu avião.	3
	Material do avião Utilize a função indicada para definir os materiais do seu avião. Ao menos 3 cores devem ser utilizadas.	2
Modelagem e dinâmica do ambiente (12 pontos)	Plano base Crie um plano wireframe de base. Esse plano será importante para dar a noção de movimento ao sistema (será nosso ponto de referência).	1
	Modelagem das árvores As árvores serão modeladas com geometrias básicas devidamente agrupadas. Utilize um mínimo de 2 tipos de geometrias básicas para construir cada árvore.	2
	Dinâmica do ambiente - movimentação do plano Como nesta etapa o sistema será executado indefinidamente, deve-se pensar uma forma de reaproveitar um plano de tamanho finito (a forma de reaproveitamento deste objeto será um dos critérios de avaliação deste item).	5
	Dinâmica do ambiente - <i>fade in</i> Objetos mais distantes (árvores e planos distantes) começam transparentes e começam a ser vistos à medida que forem se aproximando do avião.	4
Outros (1 ponto)	Iluminação direcional Crie um sistema de iluminação direcional para iluminar o ambiente.	1
Nota 1: Informações adicionais e/ou correções a este enunciado podem ser adicionadas na forma de comentário 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: 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. Estejam preparados de acordo.

Nota importante: no mesmo nível da pasta *examples* do nosso repositório, crie uma pasta T1 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 LABORATÓRIO ANTES de enviá-lo.**

Prazo para envio do trabalho:	11/04* (terça-feira - até 23:59)
Datas de apresentação do trabalho:	14/04 (sexta) 17/04 (segunda)

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

