

Flight Simulator - Parte II (30 pontos)

Dando continuidade ao trabalho anterior, nesta etapa focaremos na principal funcionalidade do nosso sistema que será aplicar o simulador a uma 'escola de voo'.

Escola de voo

Para adicionarmos os elementos necessários para implementar a escola de voo, será necessário adicionar três elementos: definir o caminho, os *checkpoints* e um sistema para marcar o tempo. O caminho poderá ser visualizado e, para tal, você deverá utilizar um gerador de curvas baseado em pontos (a sugestão é utilizar o algoritmos de [catmull-rom](#)). Os pontos do caminho serão os centros dos checkpoints.

Os checkpoints serão representados por [torus](#), e devem estar sempre com a orientação perpendicular ao caminho. Utilize transparência nos objetos. O caminho deverá ter entre 10 e 15 checkpoints e deve começar e terminar em locais próximos. Construa um caminho que não seja simplesmente circular. A complexidade do caminho será um dos critérios de avaliação deste item.

O sistema deverá identificar quando um checkpoint for atravessado removendo o objeto da cena. A forma (cálculo) para definir se o avião atravessou ou não um determinado checkpoint será um dos critérios de avaliação deste item. Ao concluir o caminho, deverá ser apresentado na tela o tempo gasto para concluí-lo. Esse tempo começará a ser contado quando o primeiro checkpoint for atravessado.

A simulação começará com o avião parado (pousado) sobre o plano base. O avião deverá decolar mas não será necessário pousá-lo ao final. Veja [este vídeo](#), entre os tempos 3:33 e 4:23, para ter uma noção geral do comportamento geral que deverá ser implementado.

O caminho que o avião deve seguir pode ser visualizado ou ocultado ao pressionarmos a tecla "Enter".

Ambiente e aspectos visuais

O cenário será composto por uma luz direcional que representará o Sol e uma luz ambiente do tipo [HemisphereLight](#). O grupo deverá pesquisar o funcionamento e vantagens em usar esse tipo de iluminação ambiente.

O grupo deverá criar no ambiente duas montanhas formadas por objetos do tipo *ConvexBufferGeometry*. As montanhas **não** terão geração randômica, então o grupo deverá definir todos os pontos de forma criteriosa. Serão criadas 3 montanhas. A montanha central será a maior e a mais alta. Para que as montanhas tenham formato mais irregular, utilize ao menos dois objetos do tipo *ConvexBufferGeometry* para as montanhas menores e três para a montanha mais alta. Cada objeto terá no mínimo 20 pontos. É importante que os pontos da base da montanha estejam posicionados sobre o plano, que nessa versão terá uma cor semelhante à ilustrada ao lado. Veja na ilustração ao lado qual deve ser o formato aproximado das montanhas. A sugestão é que as montanhas modeladas tenham um formato semelhante a esse. Outra possibilidade é construir as montanhas em um aplicativo externo e importá-la no sistema. Neste caso, na apresentação do trabalho, o aplicativo deverá estar aberto para que o grupo possa mostrar o que usou, número de polígonos utilizados, etc.



Fig 1 - Montanha

Também usando como base a ilustração supracitada, deverão ser criadas árvores que serão aleatoriamente posicionadas sobre o plano. Estas árvores projetarão suas respectivas sombras sobre o solo. É importante que o número de vértices utilizado para a criação da árvore não seja muito alto para não impactar no desempenho geral do sistema. As árvores podem ser criadas com primitivas básicas ou criadas externamente e posteriormente importadas. Ao menos 50 árvores deverão ser criadas no ambiente.

As montanhas e as árvores utilizarão material do tipo Gouraud. O avião utilizará material de phong na maioria de suas partes (eventualmente material do tipo Gouraud por ser aplicado em superfícies emborrachadas, por exemplo).

Modo de câmera

Nesta etapa adicionaremos o nosso último modo de câmera denominado "Modo Cockpit". Basicamente a ideia é posicionar a câmera dentro do avião, em um ponto de vista semelhante ao ilustrado na Figura 2. Então teremos um total de 3 modos de câmera: Modo de simulação, modo inspeção (ambos do T1) e modo cockpit. Para acessar o modo cockpit é necessário estar no modo de simulação e pressionar a tecla 'c'. Pressionando novamente volta para o modo de simulação do T1. Neste modo é necessário ser possível ver parte do avião (bico, hélice da frente ou algo assim). Não é necessário modelar a parte interna do avião, pois será avaliada apenas o posicionamento da câmera. O comportamento do avião neste modo de câmera será o mesmo do modo de simulação.



Fig 2 - Visão do modo cockpit

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

Grupo	Item	Pontos
Escola de voo (12 pontos)	Construção do caminho Qualidade e complexidade do caminho criado, considerando a posição inicial do avião (pousado).	4
	Visualização/ocultação do caminho Correta implementação da tecla 'enter' para visualizar/ocultar o caminho	1
	CheckPoints - Funcionalidade Sistema de contagem dos checkpoints atravessados e contagem de tempo	4
	CheckPoints - Qualidade visual Qualidade da visualização da geometria e posicionamento dos torus em relação ao caminho.	3
Ambiente (12 pontos)	Montanhas Correto uso do <i>ConvexBufferGeometry</i> ou importação das geometrias feitas externamente. Serão avaliadas a complexidade das geometrias e aparência final das montanhas e do plano base.	5
	Árvores Correta criação e/ou importação das geometrias e aparência	4
	Avião Utilização de materiais (prioritariamente phong) adequados no avião	2
	Iluminação Correta utilização da luz direcional e ambiente.	1
Modo Cockpit (6 pontos)	Modo Cockpit Correta visualização, comportamento e posicionamento da câmera neste modo	5
	Acesso ao Modo cockpit Correto mapeamento da tecla 'c' para acessar/sair deste modo	1

Foco na apresentação:

Cada aluno pode ser questionado sobre qualquer aspecto do desenvolvimento do trabalho mas minimamente um dos componentes do grupo deve ficar responsável por um (ou mais em caso de duplas e trabalhos individuais) dos tópicos abaixo:

- Escola de voo;
- Ambiente;
- Modo Cockpit.

O trabalho deve ser implementado em Three.js. Um dos aspectos mais importantes da implementação é a questão da clareza do código. O projeto deverá ser minimamente modelado antes de ser implementado.

O grupo será questionado a respeito de detalhes do código e a avaliação será individual, realizada em sala virtual separada no Google Meet no dia da apresentação.

Notas:

- Continue desenvolvendo seus códigos na pasta **works**. Com isso, ao enviar seu código para avaliação, só será necessário enviar o arquivo *.html* e os arquivos *.js* correspondentes pois os mesmos serão copiados para a mesma pasta na máquina onde a avaliação será realizada. Envie os arquivos **compactados** (zip, rar etc).
- Para o caso de optar por criar externamente o modelo de árvore que será utilizado, crie **dentro da pasta works** uma pasta chamada "assets" e copie lá o arquivo que foi importado.

Prazo para envio 1: 26/07/2021 (segunda - qualquer horário)
Apresentação 1: 28/07/2021 (aula de quarta)

Prazo para envio 2: 29/07/2021 (quinta - qualquer horário)
Apresentação 2: 02/08/2021 (aula de segunda)

Bom trabalho!