	<p>Centro Tecnológico Departamento de Informática</p>
<p>Disciplina: Computação Gráfica</p>	<p>Código: INF09282 e INF09284</p>
<p>Prof. Thiago Oliveira dos Santos</p>	

Trabalho Final

1 Introdução

Esse trabalho tem como objetivo fixar as técnicas de computação gráfica 3D adaptando o trabalho anterior para 3 coordenadas.

O objetivo geral do jogo será parecido com o do trabalho curto 4, porém com algumas funcionalidades específicas do ambiente 3D. O jogo terá como objetivo fazer o resgate dos objetos sem tomar um tiro de seus adversários ou deixar o combustível acabar. Para isso você poderá matar seus adversários com tiro e deverá reabastecer antes de acabar o combustível. O jogo acaba quando você resgata todos os objetos, ou quando você morre.

O aluno deverá implementar um programa que transforme o trabalho curto 4 em 3D. O trabalho deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freeglut)..

2 Especificação das Funcionalidades

Ao rodar, o programa deverá ler, de um arquivo de configurações (denominado “config.xml”), as configurações necessárias para suas tarefas. O arquivo de configurações deverá estar no formato xml e será fornecido juntamente com a especificação do trabalho. A localização do arquivo “config.xml” será fornecida pela linha de comando ao chamar o programa. Por exemplo, se o arquivo estiver dentro de uma pasta chamada “Test1” localizada na raiz, basta chamar o programa com “/Test1/” como argumento. As informações contidas nesse arquivo servirão para ler o arquivo SVG contendo as informações da arena. Esse trabalho utilizará as tags já especificadas nos trabalhos anteriores.

Após ler as informações do arquivo de configurações, o programa deverá carregar os elementos da arena do arquivo do tipo SVG respectivo e um helicóptero verde ao invés de um círculo verde, helicópteros vermelhos ao invés de círculos vermelhos, esferas azuis, ou outros objetos a seu gosto, ao invés dos círculos azuis, e um retângulo cinza. A janela de visualização deverá ter 500x500 pixels.

Arena


Assim como no trabalho curto 4, o programa deverá criar uma arena virtual, porém desta vez em 3D. O plano x e y terá informações idênticas às lidas do arquivo "svg" (assim como os trabalhos anteriores). A altura da arena, z, deverá ser 5 vezes a altura do helicóptero (a ser definido adiante).

Helicóptero

O helicóptero deverá ter os mesmos componentes dos trabalhos anteriores (base, hélice e canhão). Utilize a criatividade para construir o helicóptero! Ele continuará sendo delimitado pelo círculo definido no "svg", porém agora ao invés de ser envolvido por um círculo, ele será envolvido por uma esfera de raio igual ao do círculo. Portanto, o helicóptero deverá estar contido nessa esfera. Perceba que essa esfera é “virtual”, ela serve apenas para calcular a colisão e não deve ser mostrado na tela. A hélice deverá girar seguindo os mesmos princípios dos trabalhos anteriores. O canhão também deverá seguir os mesmos princípios dos trabalhos anteriores, porém ele poderá se mover em 3D. Isto é, ele poderá ir de -45° a +45°, de um lado para o outro, e de 0° a -45°, horizontal para baixo, do helicóptero. Mexer o mouse na vertical para cima, move o canhão para cima. Mexer o mouse na vertical para baixo, move o canhão para baixo. Mexer o mouse na horizontal para esquerda, gira o canhão no sentido anti-horário quando visto de cima. Mexer o mouse na horizontal para direita, gira o canhão no sentido horário quando visto de cima. O botão esquerdo faz o tanque atirar.

Pousar e Decolar

Utilize agora as teclas “+” e “-” para subir e descer o helicóptero respectivamente. Ao encostar no chão o helicóptero passará para o estado pousado e ao sair do chão passará para o estado voando.

	<p>Centro Tecnológico Departamento de Informática</p>
<p>Disciplina: Computação Gráfica</p>	<p>Código: INF09282 e INF09284</p>
<p>Prof. Thiago Oliveira dos Santos</p>	

Helicópteros Inimigos

Cada helicóptero adversário deverá ficar se movendo aleatoriamente (algoritmo de livre escolha do aluno, e pode ser o mesmo do trabalho anterior) e de tempos em tempos (utilizar o parâmetro “freqTiro”) apontar para a direção do jogador e atirar. Cada helicóptero inimigo deverá ser inicializado em uma altura de voo diferente do outro (aleatória e de escolha do aluno).

Aparência do Jogo

Deverão ser utilizados os conceitos de iluminação e textura. O jogo deverá conter pelo menos 2 modelos diferentes de luz (pontual, direcional ou spot). As paredes, o chão e o teto da arena deverão ser texturizados, assim como o corpo do helicóptero. O aluno está livre para escolher as texturas e também poderá utilizar luzes adicionais. Usa a criatividade!

Reabastecimento

O reabastecimento deverá seguir os conceitos do trabalho anterior.

Resgate de Objetos

Os objetos para resgate deverão estar posicionados estaticamente no chão da arena. Para resgatar os objetos, basta verificar se a esfera “virtual” representando o jogador está tocando a esfera “virtual” representando o objeto a ser resgatado.

Câmeras

O jogo deverá implementar 3 tipos de visões que poderão ser trocadas com os botões numéricos do teclado (1, 2 e 3). O botão 1 (opção padrão) deverá acionar uma câmera perspectiva posicionada no cockpit do helicóptero e olhando para frente (up apontando para o teto). O botão 2 deverá acionar uma câmera perspectiva posicionada em cima do canhão e olhando na direção do canhão (up apontando para o teto). O botão 3 deverá acionar uma câmera perspectiva posicionada inicialmente atrás do helicóptero (a uma distância grande suficiente para ver todo o helicóptero por uma terceira pessoa) e a uma altura superior à do helicóptero, e olhando para o centro do helicóptero (up apontando para o teto). Essa última câmera poderá ser rotacionada em torno do helicóptero quando pressionado o botão direito do mouse (-180° a +180° de um lado para o outro e -90° a +90° de cima para baixo).

Bônus 1

Implementar uma visão permanente do cockpit do helicóptero. Utilizar uma janela com 200 pixels a mais em y para mostrar a visão do cockpit constantemente durante o jogo. É necessário dividir o viewport!


Bônus 2

Mapa de posição, sua, dos seus adversários e dos objetos a serem resgatados. Utilizar uma câmera ortogonal para desenhar um mini mapa da arena descrevendo a sua posição (verde), a posição dos adversários (vermelho), e dos objetos a serem resgatados (azul). O chão desse mapa deve ser transparente para não ofuscar a visão original do jogo. Utilizar o mesmo conceito da impressão de texto no canto da tela. O mapa deve ficar fixo no canto inferior direito e ocupar 1/4 da largura da janela

OBSERVAÇÕES: O aluno poderá incluir parâmetros adicionais para facilitar a apresentação do trabalho. Por exemplo, o aluno pode criar uma tecla para habilitar e desabilitar uma determinada funcionalidade, para mostrar que ela funciona. O aluno deverá utilizar os mesmos conceitos já exigidos nos trabalhos anteriores. Arquivos exemplo serão distribuídos juntamente com essa especificação.

3 Regras Gerais

O trabalho poderá ser feito em dupla. Trabalhos identificados como fraudulentos serão punidos com nota zero. Casos típicos de fraude incluem, mas não se restringem a cópias de trabalhos, ou parte dele, assim como trabalhos feitos por terceiros. Caso seja necessário confirmar o conhecimento do aluno, individualmente, a respeito do código entregue, o professor poderá pedir ao aluno para apresentar o trabalho oralmente em um momento posterior. A nota da apresentação individual servirá para ponderar a nota obtida no trabalho.

	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos		

3.1 Entrega do Trabalho

O código deverá ser entregue por email (para: todosantos@inf.ufes.br) dentro do prazo definido no portal do aluno. Trabalhos entregues após a data estabelecida não serão corrigidos.

A entrega do trabalho deverá seguir estritamente as regras a seguir. O não cumprimento acarretará na **não correção do trabalho** e respectivamente na atribuição da nota zero.

- Assunto da mensagem: [CG-2015-2] <tipo do trabalho>. Onde, <tipo do trabalho> pode ser TC1, TC2, TC3 e representa respectivamente trabalho curto 1, 2, 3, etc , ou TF para o trabalho final.
- Anexo da mensagem: arquivo zippado (com o nome do(s) autor(es), ex. FulanoDaSilva_CiclanoMaluco.zip) contendo todos os arquivos necessários para a compilação do trabalho;
- Não enviar arquivos já compilados, inclusive bibliotecas!
- O diretório deverá necessariamente conter um makefile que implemente as seguintes diretivas "make clean" para limpar arquivos já compilados, "make all" para compilar e gerar o executável. O executável deverá ser chamado *trabalhocg*.

Lembre-se que a localização do arquivo config.xml será passada via linha de comando e portanto não se deve assumir que haverá um arquivo desses na pasta do executável.


4 Pontuação

O trabalho será pontuado conforme a tabela dada na última folha desse documento e resumida abaixo. Bugs serão descontados caso a caso. Observe que existem duas funções bônus no trabalho, ou seja, 2 pontos extras. Os pontos dessas questões bônus serão utilizados para completar a nota desse trabalho, da prova, ou dos trabalhos curtos que não tenham atingido a nota máxima 10.

Funcionalidade	Pontuação
Base do jogo	3
Helicóptero 3D	2
Aparência do jogo (iluminação e textura)	2
Câmeras	3
Bônus 1	1
Bônus 2	1


4.1 Apresentação do Trabalho

O grupo terá 20 minutos para apresentar seu trabalho para a turma. A apresentação será feita no laboratório, portanto o aluno poderá utilizar um computador do laboratório e o projetor. As apresentações ocorrerão no horário da aula e em uma data posterior à de entrega. Durante esses 20 minutos, o aluno deverá apresentar e testar todas as funcionalidades requeridas do trabalho. O trabalho (arquivos) a ser utilizado na apresentação deverá ser o mesmo enviado para o professor. É responsabilidade do aluno testar o trabalho na máquina de apresentação do Labgrad antes do dia da apresentação. A ordem de apresentação será sorteada durante a aula, portanto, todos os alunos devem estar preparados para apresentar o trabalho durante o período de apresentação. Os alunos devem estar preparados para responder possíveis perguntas sobre o trabalho. Prepare-se para fazer a apresentação dentro do seu tempo (20 min.). **Pontos só serão ganhos por funcionalidades apresentadas**, isto é, a audiência deverá ser capaz de ver e perceber o resultado produzido pela funcionalidade implementada no jogo. Cabe aos alunos, portanto, criar atalhos no trabalho para facilitar a apresentação das funcionalidades.

	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos		

5 Erratas

Qualquer alteração nas regras do trabalho será comunicada em sala e no portal do aluno. É de responsabilidade do aluno freqüentar as aulas e manter-se atualizado.

	Centro Tecnológico Departamento de Informática	
Disciplina: Computação Gráfica		Código: INF09282 e INF09284
Prof. Thiago Oliveira dos Santos		

Nome do aluno:	
Nome do aluno:	

Itens	Sub-Itens	Feito	Observações	Pontos	Nota
Base do jogo	Pousar e Decolar			0,5	
	Movimento Helicóptero			0,5	
	Colisão helicóptero			0,5	
	Reabastecimento			0,5	
	Acertar e Levar tiros			0,5	
	Resgate			0,5	
Helicóptero 3D	Hélice			1,0	
	Canhão			1,0	
Aparência do jogo	Iluminação 1			0,5	
	Iluminação 2			0,5	
	Textura Arena			0,5	
	Textura Helicóptero			0,5	
Câmeras	Câmera 1			1,0	
	Câmera 2			1,0	
	Câmera 3			1,0	
Bônus 1	Segundo Viewport			1,0	
Bônus 2	Câmera paralela			1,0	