	<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, TC4, para 3 coordenadas.

O objetivo geral do jogo será parecido com o do trabalho curto 4 e, também, terá os mesmos elementos de jogo, porém com algumas funcionalidades específicas do ambiente 3D.

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 (outros exemplos de caminhos possíveis “../Test1/”, “./../Test1/”, etc.). As informações contidas nesse arquivo servirão para ler o arquivo SVG contendo as informações da arena.

Após ler as informações do arquivo de configurações (equivalentes ao TC4), o programa deverá carregar os elementos da arena do arquivo do tipo SVG respectivo e colocar um jogador verde ao invés de um círculo verde, jogadores vermelhos ao invés de círculos vermelhos, além dos outros elementos da arena como definidos nos trabalhos anteriores. A janela de visualização deverá ter 500x500 pixels. O jogo deverá ter as mesmas funcionalidades do TC4, exceto as que forem redefinidas abaixo.

### *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 8 vezes a altura do jogador (i.e., diâmetro do círculo do jogador). A base inimiga deverá ser representada por uma esfera com metade dentro do chão da arena. Pode usar objetos mais complexos para representar esses conceitos. Use a criatividade!

### *Jogador*

O jogador deverá ter os mesmos componentes dos trabalhos anteriores (parte principal, asas, hélices, etc.), porém agora em 3D. Utilize a criatividade para construir o jogador! Ele será delimitado por uma esfera com o mesmo raio do círculo definido no “svg”. Perceba que a esfera é “virtual”, ela serve apenas para calcular a colisão e não deve ser mostrada na tela. A subida e descida do avião serão controladas agora pelas teclas “w” e “s” respectivamente. Ao apertar “s” o avião deverá descer enquanto ela estiver sendo pressionada e ao apertar “w” o avião deverá subir enquanto ela estiver sendo pressionada. O tiro de frente do avião deverá sair de um canhão que pode ser controlado pelo mouse. 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 canhão atirar. O movimento do canhão deve ser limitado a uma angulação de no máximo 30 graus para cima, para baixo, para a esquerda e para a direita.

### *Jogadores Inimigos*

Os jogadores não devem colidir entre si, ou seja, eles podem passar através do outro. Cada jogador adversário deverá ficar se movendo aleatoriamente, incluindo movimento de subida/descida, (algoritmo de livre escolha do aluno, mas respeitando as propriedades de movimento definidas no TC3 e TC4) e de tempos em tempos atirar.

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

### *Jogo em geral*

O jogo em geral deverá seguir as funcionalidades do TC4, e.g., ganhar, perder, mostrar mensagens, etc. Contudo, agora as funcionalidades devem ser adaptadas para o 3D. Por exemplo, o tiro agora sai em 3D, portanto, pode atingir os outros em qualquer posição da arena, desde que o tiro não colida antes com alguma coisa bloqueando o caminho (ex. com um obstáculo). A bomba continua destruindo as bases inimigas.

### *Aparência do Jogo*

Deverão ser utilizados conceitos de iluminação e textura. O jogo deverá conter pelo menos um modelo de luz na arena (pontual ou direcional). Além disso, o jogo deverá ter um modo noturno (fazer a troca de modos com a tecla “n”) em que todas as luzes da arena são apagadas, exceto o farol (representada por uma iluminação spot) fixado na frente do avião e apontando para a frente dele. As paredes, o chão e o teto da arena deverão ser texturizados, assim como o jogador. O aluno está livre para escolher as texturas e utilizar luzes adicionais. O aluno pode colocar objetos adicionais no mapa, como árvores, prédios, casas, etc. Use a criatividade!

### *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 avião e olhando para frente. O botão 2 deverá acionar uma câmera perspectiva posicionada em cima do canhão e olhando paralelamente ao canhão e na mesma direção que ele (up apontando para o céu). Com essa visão, seria possível ver parte do “cano” do canhão (ou do que for usado para atirar no seu modelo), como se fosse a mira de uma arma. O botão 3 deverá acionar uma câmera perspectiva posicionada inicialmente atrás do jogador (a uma distância grande suficiente para ver todo o jogador por uma terceira pessoa) e a uma altura superior à do jogador, e olhando para o centro do jogador (up apontando para o teto). Essa última câmera poderá ser rotacionada em torno do jogador ( $-180^\circ$  a  $+180^\circ$  de um lado para o outro e  $-60^\circ$  a  $+60^\circ$  de cima para baixo) quando pressionado o botão direito do mouse em cima da arena.

### *Mini mapa*


Mapa de posição, do jogador, dos adversários e das bases inimigas. Utilizar uma câmera ortogonal para desenhar um mini mapa da arena descrevendo a sua posição (verde), a posição dos adversários (vermelho), e das bases inimigas (amarelo), como no trabalho 4. O chão desse mapa deve ser transparente para não ofuscar a visão original do jogo (ou seja, utilizar apenas linha para representar o círculo da arena). 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.

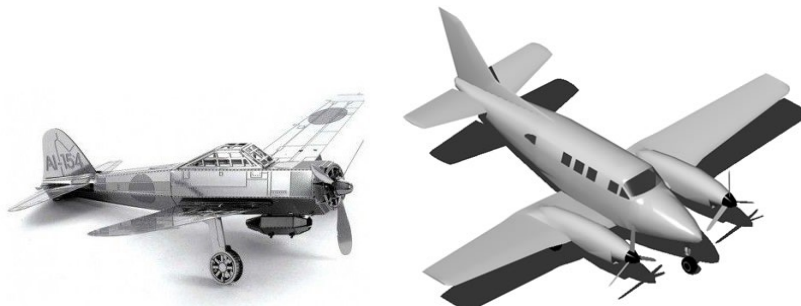
### *Bônus 1 - Visão da bomba*

Implementar uma visão permanente da bomba. Ela deverá usar uma câmera perspectiva posicionada no centro da bomba e olhando para baixo (up na direção do movimento). Utilizar uma janela com 200 pixels a mais em y para mostrar essa visão constantemente durante o jogo (isto é, a janela inicial de 500x500 será 500 por 700 se essa funcionalidade for implementada). DICA: É necessário dividir o viewport! Enquanto a bomba não for lançada, essa parte da janela reservada para ele deverá mostrar preto. Quando ele for lançado a visão da câmera deve começar e continuar até que ela seja destruída, ou seja, até que ele atinja o chão.

### *Bônus 2 – Modelos Avançados*

Utilizar modelos avançados de jogador e suas partes (ver exemplos abaixo). O aluno está livre para utilizar modelos 3D e suas partes feitos no Blender ou baixados da internet. Não pode haver grupos com modelos repetidos. A qualidade dos modelos será julgada caso a caso. Atenção, modelos muito pesados podem deixar o jogo muito lento e isso não é desejável.

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



**OBSERVAÇÕES:** O aluno poderá incluir (e deverá, se for a única maneira de mostrar uma funcionalidade) parâmetros e teclas 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. As funcionalidades só serão pontuadas se elas forem vistas durante a apresentação, isto é, falar que colocou a luz não basta, é necessário mostrar o seu efeito e explicar coerentemente. O aluno deverá utilizar os mesmos conceitos já exigidos nos trabalhos anteriores. Arquivos exemplo serão distribuídos juntamente com essa especificação. Inclua um README.txt explicando os atalhos e funcionalidades adicionais.

### 3 Regras Gerais

O trabalho poderá ser feito em dupla, exceto para os alunos das pós-graduação. Trabalhos identificados como fraudulentos serão punidos com nota zero. Casos típicos de fraude incluem, mas não se restringem às cópias de trabalhos, ou de parte dele, assim como trabalhos feitos por terceiros. Caso seja necessário confirmar o conhecimento do aluno a respeito do código entregue, o professor poderá pedir ao aluno para apresentar o trabalho oralmente, durante a apresentação, ou em um momento posterior. A nota da apresentação servirá para ponderar a nota obtida no trabalho. Cada membro da dupla deverá obrigatoriamente conhecer todo o conteúdo e código do trabalho.

#### 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-2019-2] <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 autor, ex. FulanoDaSilva.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. Seja cuidadoso ao testar o seu programa, isto é, não teste com o arquivo no diretório do programa, pois você pode esquecer de testá-lo em outro lugar posteriormente.

### 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

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

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	2
Jogador 3D	2
Aparência do jogo (iluminação e textura)	2
Câmeras	3
Mini mapa	1
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 o tempo de apresentação, o aluno deverá mostrar e testar todas as funcionalidades requeridas do trabalho. O trabalho (arquivos) a ser utilizado na apresentação deverá ser o mesmo enviado para o professor, e será fornecido pelo professor na hora da apresentação. É responsabilidade do aluno testar o trabalho na máquina de apresentação do Labgrad (i.e., máquina do professor no 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 apresentações. 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.

## 5 Erratas

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

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

**Tabela de pontos**

Nome do aluno:	
Nome do aluno:	

Itens	Sub-Itens	Feito	Observações	Pontos	Nota
Base do jogo	Mover inimigos			1,0	
	Jogo (morrer, ganhar, mensagem, etc.)			1,0	
Jogador 3D	Movimentos e suas partes			1,0	
	Canhão e tiro			1,0	
Aparência do jogo	Iluminação 1			0,5	
	Farol			0,5	
	Textura			1,0	
Câmeras	Câmera 1			1,0	
	Câmera 2			1,0	
	Câmera 3			1,0	
Mini mapa	Mapa			1,0	
Bônus 1	Visão da bomba			1,0	
Bônus 2	Modelos avançados			1,0	