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

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, porém com algumas funcionalidades específicas do ambiente 3D. O jogo terá como objetivo completar uma volta na pista no sentido anti-horário sem tomar um tiro de seus adversários. Para isso você poderá matar seus adversários com tiro. O joga acaba quando você completa a volta, 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.

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 carro verde ao invés de um círculo verde, carros 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.

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 4 vezes a altura do carro (a ser definido adiante).

Carro

O carro deverá ter os mesmos componentes dos trabalhos anteriores (base, rodas e canhão), porém agora em 3D. A roda deverá girar e virar conforme os comandos. Utilize a criatividade para construir o carro! Ele continuará sendo delimitado pelo círculo definido no ”svg” e o carro deverá estar contido nele. Perceba que o círculo é “virtual”, ele serve apenas para calcular a colisão e não deve ser mostrado na tela. 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 cima, do carro. 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 carro atirar.

Carros Inimigos

O movimento dos carros inimigos deverá ficar restrito a região da pista, ou seja, eles não poderão sair da pista. Os carros não devem colidir entre si, ou seja, dois carros não podem ocupar o mesmo espaço. Cada carro adversário deverá ficar se movendo aleatoriamente (algoritmo de livre escolha do aluno, mas respeitando as propriedades de movimentos definidas no TC3) e de tempos em tempos atirar. Exemplos de movimentos, seguir a direção da pista para dar voltas; andar em zig zag; de tempos em tempos escolher uma direção e ir, etc. Ao colidir, pode dar ré e tentar outro caminho.

Tempo

O jogo deverá exibir um cronometro continuamente no canto direito superior da janela. Esse cronômetro deverá mostrar a quantidade de tempo que a volta está durando em segundos. Ele deverá inicializar no momento em que o carro começar a se mover, e deverá para quando ele cruzar a linha de chegada propriamente.

Finalização do Jogo

No final do jogo, uma mensagem deverá ser impressa na tela dizendo se você ganhou ou perdeu. O jogador ganha se completar a volta, e perde se levar um tiro.

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") e que todas as luzes da arena são apagadas e o farol do carro (representado por uma iluminação spot) é aceso. As paredes, o chão e o teto da arena deverão ser texturizados, assim como o corpo do carro. O aluno está livre para escolher as texturas e utilizar luzes adicionais. Usa 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 carro 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 carro (a uma distância grande suficiente para ver todo o carro por uma terceira pessoa) e a uma altura superior à do carro, e olhando para o centro do carro (up apontando para o teto). Essa última câmera poderá ser rotacionada em torno do carro quando pressionado o botão direito do mouse (-180° a +180° de um lado para o outro e -90° a +90° de cima para baixo).

Retrovisor

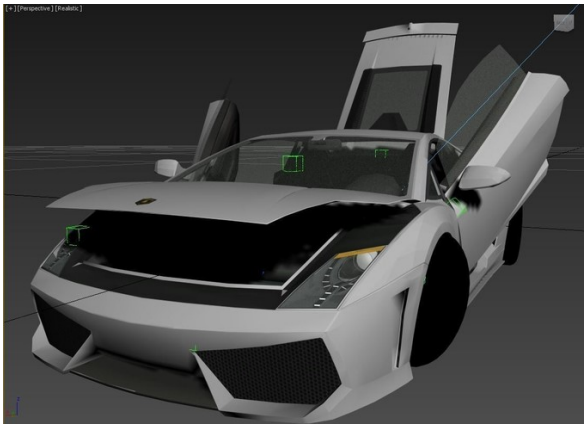
Implementar uma visão permanente do cockpit do carro. Utilizar uma janela com 200 pixels a mais em *y* para mostrar a visão do cockpit constantemente durante o jogo (isto é, a janela inicial de 500x500 será 500 por 700 se essa funcionalidade for implementada). É necessário dividir o viewport!

Bônus 1

Mapa de posição, sua e dos seus adversários. 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 (Ou seja, utilizar apenas linha para representar a pista). 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 2

Utilizar modelos avançados de carros e suas partes (ver exemplos abaixo). O aluno está livre para utilizar modelos de carro 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.

OBSERVAÇÔES: 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 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 em um momento posterior. A nota da apresentação servirá para ponderar a nota obtida no trabalho.

3.1 Entrega do Trabalho

O código deverá ser entregue por email (para: todsantos@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-2016-2] <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 testa-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 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
Carro 3D	3
Aparência do jogo (iluminação e textura)	2
Câmeras	3
Bônus 1	1
Bônus 2	1

4.1 Apresentação do Trabalho

O grupo terá 25 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 25 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, e será fornecido pelo professor na hora da apresentação. 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 apresentações. Os alunos devem estar preparados para responder possíveis perguntas sobre o trabalho. Prepare-se para fazer a apresentação dentro do seu tempo (25 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	Volta			0,5	
	Mover carros			1,0	
	Jogo (morrer, ganhar, tempo)			0,5	
Carro 3D	Movimento (rodas girando, curva, etc.)			1,0	
	Canhão e tiro			1,0	
	Retrovisor			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	
Bônus 1	Câmera paralela			1,0	
Bônus 2	Modelos avançados			1,0	