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

Trabalho 3D

1 Introdução

Este trabalho tem como objetivo fixar as técnicas de computação gráfica 3D adaptando o trabalho anterior, T2D, para 3 coordenadas.

O objetivo geral do jogo será parecido com o do trabalho 2D 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 2D em 3D. O trabalho deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freelut).

2 Especificação das Funcionalidades

Ao rodar, o programa deverá ler e interpretar os elementos da arena do arquivo do tipo SVG informado pela linha de comando. A arena será composta por uma série de elementos (ver Figura 1): uma arena representada pelo retângulo azul; um círculo verde representando o personagem do jogador e um círculo vermelho representando o oponente (controlado pelo computador). Um arquivo SVG será fornecido como exemplo juntamente com a descrição do trabalho, porém é responsabilidade do aluno testar outros arquivos com configurações (posições e tamanhos) diferentes para os elementos. A leitura do SVG poderá ser feita utilizando-se um parser para XML. Sugiro utilizar a Tinyxml que é simples e pode ser enviada juntamente com o código para ser compilada.

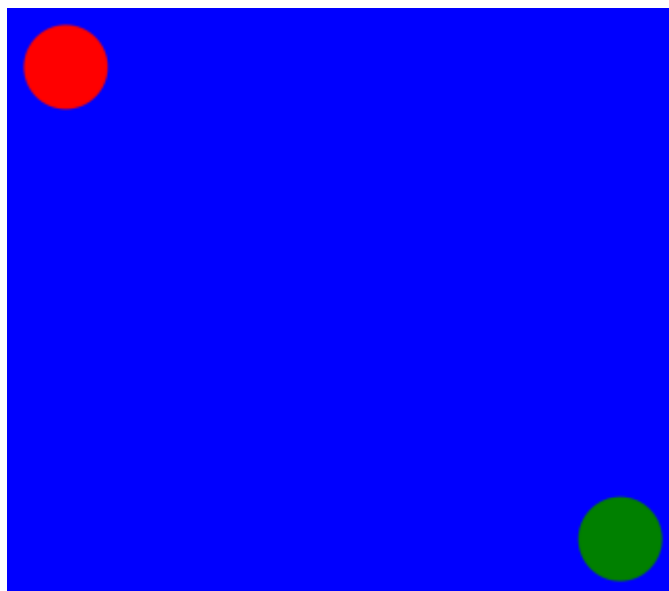



Figura 1: Visualização do arquivo SVG fornecido “arena_2.svg”, representando a arena em azul, jogador em verde e oponente em vermelho.

Após ler as informações do arquivo de configurações (equivalentes ao T2D), 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, um jogador vermelho ao invés de um círculo vermelho, além da arena como definidos no trabalho anterior. A janela de visualização deverá ter 500x500 pixels. O jogo deverá ter as mesmas funcionalidades do T2D, exceto as que forem redefinidas abaixo.

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

Arena

Assim como no trabalho curto 2D, 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 2 vezes a altura do jogador escolhido.

Jogador

O jogador deverá ter cabeça, braços, corpo e pernas representados em 3D. Utilize a criatividade para construir o jogador! O sistema de colisão será como no trabalho anterior, porém agora o círculo do "svg" representará o corpo do jogador quando visto de cima. Assim como no trabalho anterior, para os jogadores não ocuparem o mesmo espaço, a colisão deverá ser calculada considerando um círculo imaginário envolvendo o jogador. Perceba que o círculo é "virtual", ele serve apenas para calcular a colisão e não deve ser mostrado na tela. Os controles do jogador também permanecerão como no trabalho anterior.

Andar

O controle do movimento de andar será como no trabalho anterior, porém agora veremos também o efeito das pernas se movendo. Portanto, ao movimentar os jogadores, as pernas deverão simular o movimento de andar de um humano. Use a criatividade!

Dar Socos

O jogador poderá dar socos como no trabalho anterior e a colisão será medida com a esfera virtual englobando a cabeça do jogador. Ou seja, assumir que a cabeça está dentro de uma esfera para permitir o cálculo da colisão.

Jogador Oponente

O jogador oponente deverá se movimentar como no trabalho anterior, ou seja, de forma "aleatória" e buscando a luta.

Jogo em geral

O jogo em geral deverá seguir as funcionalidades do T2D, e.g., ganhar, perder, mostrar mensagens, etc.

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 e cada um dos jogadores é iluminado por um holofote individual (representada por uma iluminação spot) vindo do teto e apontando para o jogador. 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. 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 olho do jogador e olhando para frente. O botão 2 deverá acionar uma câmera relógio, ou seja, uma câmera perspectiva posicionada no pulso e olhando para frente e paralelamente ao braço do jogador (up apontando para o teto). Com essa visão, seria possível ver parte do braço e da luva, assim como o que estiver a frente da luva. 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 (360 graus em torno do jogador e ± 60 graus para cima e para baixo) quando pressionado o botão direito do mouse em cima da arena. As teclas de + e - controlarão o seu zoom.

Mini mapa

Mapa de posição, dos jogadores. Utilizar uma câmera ortogonal para desenhar um mini mapa da arena descrevendo a sua posição (verde) e a posição do oponente (vermelho). O chão desse mapa deve ser transparente para não ofuscar a visão original do jogo (ou seja, utilizar apenas linha para representar a

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

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 do Oponente

Implementar uma visão do oponente, ou seja, ver o jogo com os olhos do oponente. 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!

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, ou editado com ambos. 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.



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

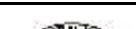
O trabalho poderá ser feito em dupla, exceto para os alunos das pós-graduação. Trabalhos identificados como fraudulentos serão punidos com nota zero. Casos típicos de fraude incluem, mas não se restringem às cópias de trabalhos, ou de parte dele, assim como trabalhos feitos por terceiros. Cada membro da dupla deverá obrigatoriamente conhecer todo o conteúdo e código do trabalho.

3.1 Entrega do Trabalho

O código deverá ser entregue pelo Google Classroom dentro do prazo definido. Trabalhos entregues após a data estabelecida não serão avaliados.

A entrega do trabalho deverá seguir estritamente as regras a seguir. O não cumprimento **inviabilizará a correção do trabalho** que, por sua vez, receberá nota zero.

- Arquivo zippado (com o nome do(s) autor(es), ex. FulanoDaSilva.zip ou FulanoDaSilva_CiclanoSantos.zip) contendo todos os arquivos necessários para a compilação do trabalho;
- **Não enviar** arquivos já compilados, inclusive bibliotecas!

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

- O arquivo zip 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 da arena 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 pontos extras. Os pontos dessas questões bônus serão utilizados para completar a nota desse trabalho ou do trabalho anterior 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 online com compartilhamento de tela, sendo que todos os componentes do grupo devem estar preparados para apresentar o trabalho. 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 (mostrando o funcionamento) 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. 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 obtidos 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	Movimentos do oponente			1,0	
	Jogo (morrer, ganhar, mensagem, etc.)			1,0	
Jogador 3D	Andar			1,0	
	Soco			1,0	
Aparência do jogo	Iluminação 1			0,5	
	Holofote			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 do oponente			1,0	
Bônus 2	Modelos avançados			1,0	