	<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 3D

1 Introdução

Este trabalho tem como objetivo fixar as técnicas de computação gráfica 3D adaptando o trabalho anterior, T2D, para 3 coordenadas. Ele deverá ser implementado em C++ (ou C) usando as bibliotecas gráficas OpenGL e GLUT (freeglut).

O aluno deverá implementar um jogo de ataque de barris. O jogo será composto por um jogador com uma arma para atirar nos barris que rolam em sua direção. O personagem do jogador será controlado pelo teclado e, poderá atirar nos barris com uma arma controlada pelo mouse. Cada barril terá uma quantidade de vezes que poderá ser atingido, sendo destruído quando essa quantidade extrapolar. Nesta versão, o jogo será 3D (ver exemplo de inspiração para o jogo na Figura 1). Os barris iniciam em uma posição horizontal aleatória no topo da arena e rolam em linha reta para baixo. O jogador inicia no centro da base da arena e pode mover-se de um lado para o outro e para cima e para baixo sem sair da arena e sem ultrapassar o centro da arena na direção vertical. Caso o barril encoste no jogador, ele morre. Cada barril poderá ter (decisão aleatória com 50% de chance) um inimigo em cima dele que poderá atirar no jogador. Caso o barril seja destruído, o inimigo em cima dele morre. Caso o inimigo seja atingido, somente ele morre e o barril continua. Caso o tiro do inimigo acerte o jogador, o jogador morre e perde o jogo. O jogo acaba com vitória após o jogador destruir N barris. O número de barris destruídos deverá ser mostrado no topo da arena.

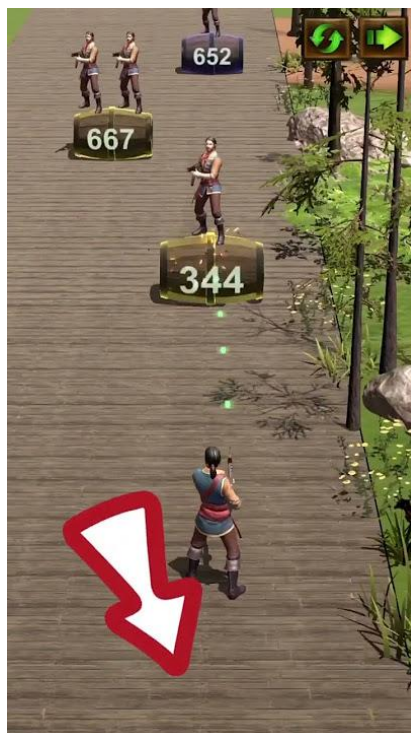



Figura 1: A figura apresenta a inspiração para este jogo.

2 Especificação das Funcionalidades

O programa deverá ser configurável por meio de um arquivo de configurações colocado no mesmo diretório do executável. Deve-se utilizar um arquivo em formato XML (denominado configuracoes.xml).

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

A leitura do XML poderá ser feita utilizando-se um parser para XML. A biblioteca Tinyxml é uma opção simples para lidar com XML e pode ser enviada juntamente com o código para ser compilada (lembre-se que não se pode demandar instalação de bibliotecas adicionais). O arquivo de configurações deverá permitir a alteração da largura e da altura da arena, do raio da cabeça dos jogadores e inimigos, da altura (diâmetro) e largura do barril, do número de vezes que um barril deve ser atingido para ser destruído, do número N (máximo de barris destruídos para ganhar o jogo), do número de tiros por segundo, da velocidade do tiro e da velocidade do barril, velocidade do jogador. Outros parâmetros podem ser usados pelo aluno para ajudar a demonstração do jogo.

Após ler as informações do arquivo de configurações (equivalentes ao T2D), o programa deverá carregar os elementos da arena e iniciar o jogo. A janela de visualização deverá ter 500x500 pixels. O jogo deverá ter as mesmas funcionalidades do T2D, exceto as que forem redefinidas abaixo.

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 de configurações (assim como o trabalho anterior).

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 de colisão representará o corpo do jogador quando visto de cima. 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 com a junta do joelho e com a do quadril. Portanto, ao movimentar os jogadores, as pernas deverão simular o movimento de andar de um humano. É necessário fazer um movimento lateral quando o jogador andar para o lado e um frontal para quando andar para frente ou para trás. Use a criatividade!

Atirar


O jogador poderá atirar com o botão esquerdo do mouse, cada clique será um tiro que continuará sua trajetória independente do jogador. A arma do jogador deverá girar, em relação ao jogador visto de cima, no sentido anti-horário ao mover o mouse para a esquerda e horário ao mover para a direita (ver Figura 2). O movimento análogo deverá ocorrer ao mover o mouse para cima e para baixo (ou seja, deverá fazer a arma apontar para cima e para baixo). O ângulo limite tanto no movimento lateral quanto no vertical será de 60 graus e isso formará um cone de possibilidades de tiro com a arma. Para quem estiver usando modelos avançados (explicado adiante), pode colocar uma arma (representada por um cilindro) no ombro do jogador.



Figura 2: Ilustração do movimento de giro da arma.

Jogador Oponente

O jogador oponente deverá se movimentar como no trabalho anterior, ou seja, andando em cima do barril.

	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 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 uma lanterna na ponta da arma (representada por uma iluminação spot) fará a iluminação do ambiente. A arena deverá ser texturizada, assim como os jogadores e barris. 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 na arma, ou seja, uma câmera perspectiva posicionada no cano da arma e olhando para frente e paralelamente ao cano (up apontando para o teto), equivalente a uma visão de uma mira. Com essa visão, deve ser possível ver parte do cano da arma, assim como o que estiver a sua frente. 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 dessa câmera.

Mini mapa

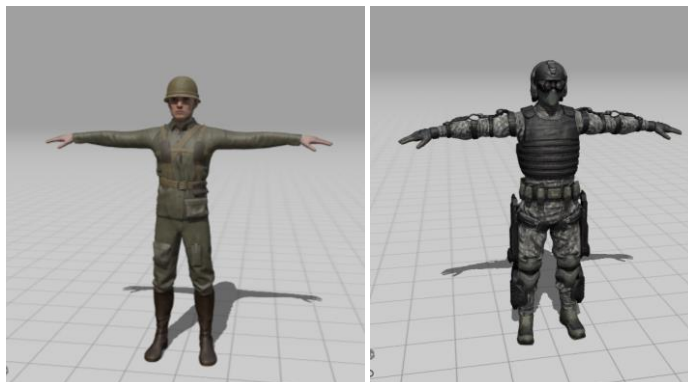
Mapa de posição, dos jogadores. Utilizar uma câmera ortogonal para desenhar um minimapa da arena descrevendo a sua posição (verde) e a posição dos oponentes (vermelho) e barris. O chão desse mapa deve ser transparente para não ofuscar a visão original do jogo (ou seja, utilizar apenas linha para representar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

Implementar uma visão do oponente, ou seja, ver o jogo se tivesse uma câmera nos olhos de um dos oponentes e olhando para o centro do jogador. Quando o oponente morrer ou sumir, pode mudar a câmera para outro. 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.



	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. 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!
- 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*.

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 de outra avaliação que não tenha atingido a nota máxima 10.

Funcionalidade	Pontuação
Base do jogo	2
Jogador 3D	2,5
Aparência do jogo (iluminação e textura)	1,5
Câmeras	3
Mini mapa	1
Bônus 1	1
Bônus 2	1

4.1 Apresentação do Trabalho

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

O grupo terá 30 minutos para apresentar seu trabalho para a turma. A apresentação será feita no laboratório, 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. **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	
	Atirar			1,5	
Aparência do jogo	Iluminação 1			0,5	
	Holofote			0,5	
	Textura			0,5	
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	