

Computação Gráfica

Trabalho II

Prof. Márcio Sarroglia Pinho

Este trabalho pode ser desenvolvido em duplas ou individualmente.

Objetivo

Criar jogo, em 3D, no qual o usuário controla um **veículo** na forma de um robô articulado. O objetivo do usuário é destruir **inimigos** que estão no cenário, posicionados atrás de um **paredão**, lançando **projéteis** a partir da posição do canhão. Caso um projétil atinja o paredão, parte dele deve ser destruído, bem como o projétil. O mesmo deve ocorrer com o **piso**. No cenário também existem objetos **amigos**, que não devem ser destruídos.

Piso e Paredão

O piso deve representar um terreno de 50m x 25m. Este terreno deve ser modelado por polígonos quadrados de 1m x 1m.

O paredão tem altura de 15m e deve ficar posicionado no meio do cenário dividindo a dimensão maior em duas partes iguais. Este paredão deve ser modelado por polígonos quadrados de 1m x 1m.

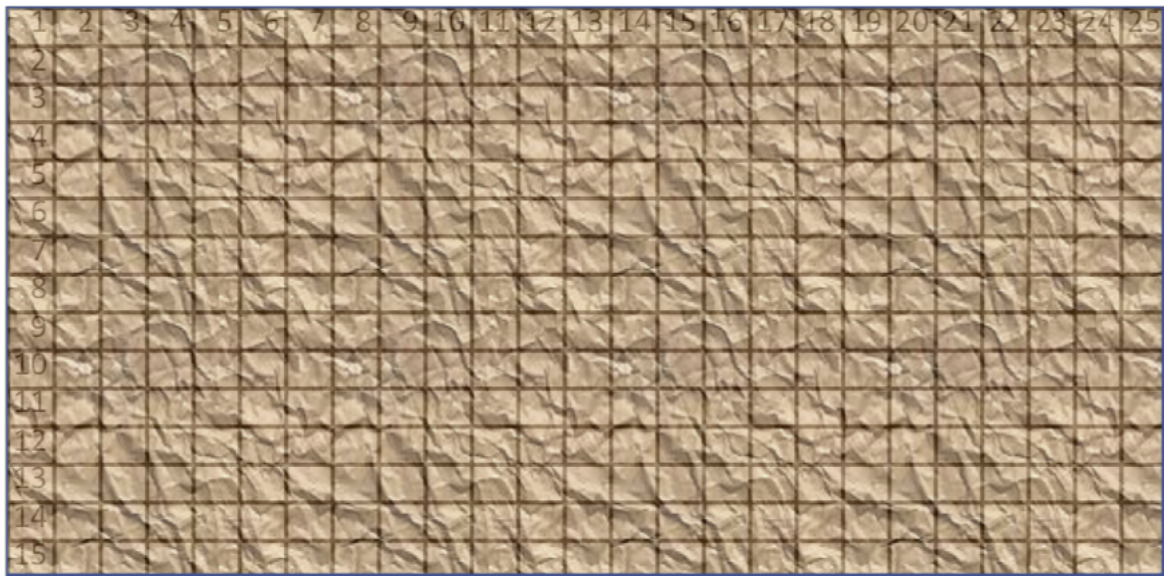


Figura - Vista Frontal do Paredão

O piso deve receber a aplicação de uma textura semelhante a grama. O paredão deve receber a aplicação de uma textura semelhante a tijolos.

Para maiores informações sobre o uso de texturas em OpenGL, consulte o link a seguir:

Cada uma destas texturas deve ser uma única imagem. Desta forma, cada polígono que compõe cada um dos objetos deve receber a aplicação de **uma parte da imagem**.

No caso do paredão, ao ser atingido por um projétil, o polígono que compõe o paredão deve deixar de ser renderizado e não gera mais colisão com o projétil, nem com o veículo.

A figura a seguir exemplifica uma vista superior do cenário. Nesta figura, cada célula tem 1m x 1m.

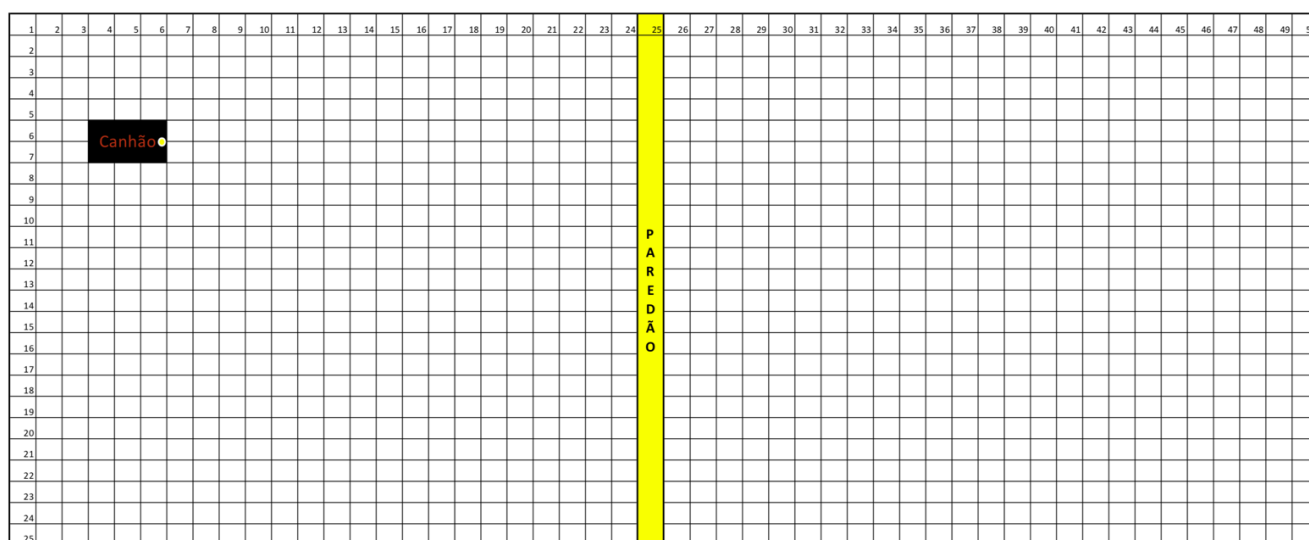


Figura - Vista Superior do Cenário

Veículo

O veículo deve ser representado por um lançador de projéteis articulado, formado, por uma base(paralelepípedo) e um canhão representado por um outro paralelepípedo.

A base do veículo deve ter comprimento de 3m x largura de 2m.

A figura a seguir apresenta uma vista ortográfica do veículo.

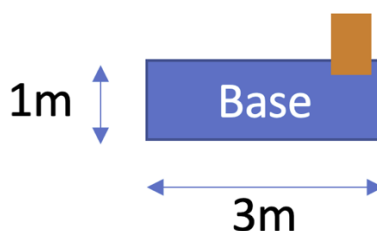


Figura - Vista Lateral do Veículo

Os segmentos são conectados por articulações. Cada articulação deve ser controlada por uma tecla distinta, utilizando, por exemplo, a seguinte convenção:

- A/a - Move articulação 1 no sentido horário/anti-horário
- B/b - Move articulação 2 no sentido horário/anti-horário

O primeiro segmento (conectado na base) deve girar no eixo Y e o segundo deve girar de forma a se inclinarem para a frente do robô, quando a rotação do primeiro segmento estiver em 0 graus.

Deve ser possível aumentar e diminuir a força de lançamento do projétil com teclas, usando a mesma convenção das articulações.

A posição do veículo no plano deve ser controlável por teclas, de forma que ele possa andar para frente, para trás e girar para esquerda e direita. Veja detalhes sobre navegação em 3D [nesta página](#).

A velocidade do veículo é de 2.5m/s. **Como isto, a travessia do cenário, em sua dimensão menor, deve levar 10 segundos.**

Projéteis

Os projéteis devem ser representados por esferas compatíveis com o tamanho do canhão.

O movimento dos projéteis deve seguir a formulação da física de **lançamento de projéteis** ou utilizar uma curva Bèzier. A direção inicial do projétil é dada pela direção do lançador. Para realizar o cálculo da trajetória dos projéteis, pode ser usada a formulação sugerida [neste material](#).

Ao atingir um objeto do cenário, este objeto deve desaparecer, contabilizando uma pontuação, utilizando a seguinte convenção:

- Objeto inimigo: +10 pontos
- Objeto amigo: -10 pontos
- Piso: -5 pontos
- Paredão: +5 pontos
- Canhão: fim do jogo

No caso de colisão com o paredão, devem ser destruídos os 8 (oito) polígonos ao redor do polígono que foi atingido.

Objetos do Cenário

O cenário deve conter pelo menos 20 objetos 3D lidos de arquivo no [formato TRI](#). Metade deles deve ser de inimigos e metade de amigos.

Estes objetos devem vir de arquivos em algum formato 3D. Podem ser lidos, por exemplo de arquivos no [formato TRI](#) ou OBJ ou [FBX](#).

Para obter objetos no formato TRI, pode ser usado o conversor i3dConvert <http://3doc.i3dconverter.com/index.html>. Para converter objetos de FBX para OBJ pode ser usado o site <http://www.greentoken.de/onlineconv>. Depois de converter para OBJ, este pode ser convertido TRI com o i3dConvert já citado (as dicas sobre as conversões são do aluno Vinicius C. Teixeira).

Cálculo da Coordenada de um Ponto

Para obter as coordenadas de um ponto de um objeto, após as transformações geométricas, uma alternativa é utilizar a função

Ponto CalculaPonto(Ponto p),
que foi disponibilizada nos códigos de exemplo.

A ideia é chamar a função logo após o desenho de um objeto como no exemplo a seguir.

```
# Desenha um cubo amarelo à direita
glColor3f(0.5,0.5,0.0) # Amarelo
glPushMatrix()
glTranslatef(2,0,0)
glRotatef(-Angulo,0,1,0)
DesenhaCubo()
P = CalculaPonto(Ponto(0,0,0))
P.imprime("Centro do Cubo da Direita:")
glPopMatrix()
```

Neste código, obtém-se a posição do centro do cubo (0,0,0) após as translações e rotações aplicadas ao objeto.

Esta estratégia pode ser usada, por exemplo, para calcular a direção do canhão, tomando-se dois pontos na superfície do objeto que define o canhão.

Rotação ao Redor de um Ponto

Para aplicar uma rotação ao redor de um ponto específico, pode-se usar o código a seguir

```
void RotacionaAoRedorDeUmPonto(float alfa, Ponto P)
{
    glTranslatef(P.x, P.y, P.z);
    glRotatef(alfa, 0,0,1);
    glTranslatef(-P.x, -P.y, -P.z);
}
```

Tabela com critérios de avaliação

Critérios	Pesos	Observações
Desenho do Paredão	2,0	O desenho deve ser feito com cada célula sendo gerada individualmente, com uma textura aplicada a cada célula. A textura deve vir da mesma imagem, mas em cada célula, uma parte diferente da imagem deve ser mapeada.
Deteção da colisão do projétil com o paredão e com os objetos do cenário	1,5	Este item só será considerado 100% correto se for possível mover o veículo corretamente e fazer os disparos conforme a especificação
Reconfiguração do Paredão após a colisão com o projétil	1,0	O paredão deve apresentar buracos nos pontos em que os projéteis atingirem. Devem ser destruídos os 8 (oito) polígonos ao redor do polígono que foi atingido.
Modelagem do Veículo com as articulações móveis	1,0	
Movimentação do Veículo - girar e andar para frente	1,5	
Lançamento do Projétil, com base no ângulo do canhão e do veículo	1,5	A trajetória deve ser definida de acordo com a direção do veículo, a inclinação do canhão e a força definida para o disparo.
Exibição correta dos objetos no formato TRI	1,5	A exibição deve levar em conta o cálculo dos vetores normais das faces.

Entrega

Data de entrega no *Moodle* e apresentação: **26/11/2023** até o horário de início da aula.

Para a entrega deverá ser criado um vídeo, de até 2 minutos, mostrando o funcionamento do programa e um relatório demonstrando que o trabalho atende aos requisitos da especificação e contendo o link para o vídeo.

Os trabalhos podem ser desenvolvidos em duplas. Os arquivos, contendo os programas-fontes do programa, devem ser compactados e submetidos pelo *Moodle* até a data e hora especificadas. **ENVIE APENAS ARQUIVOS .ZIP, ou seja, não envie 7z, rar, tar.gz, tgz, tar.bz2, etc.**

Poderá ser solicitada também a apresentação do trabalho de forma síncrona. Durante a apresentação será avaliado o domínio da resolução do problema,

podendo inclusive ser possível invalidar o trabalho quando constatada a falta de conhecimento sobre o código implementado.

A cópia parcial ou completa do trabalho terá como consequência a atribuição de nota ZERO ao trabalho dos alunos envolvidos.

FIM.