

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.

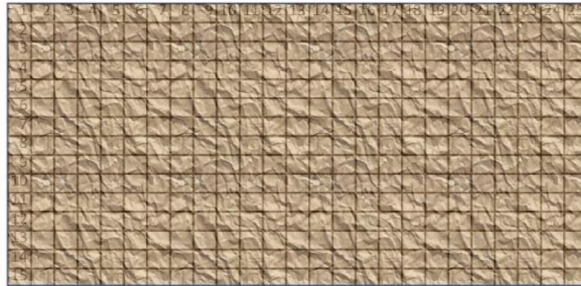


Figura - Vista Frontal do Paredão

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

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.

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.

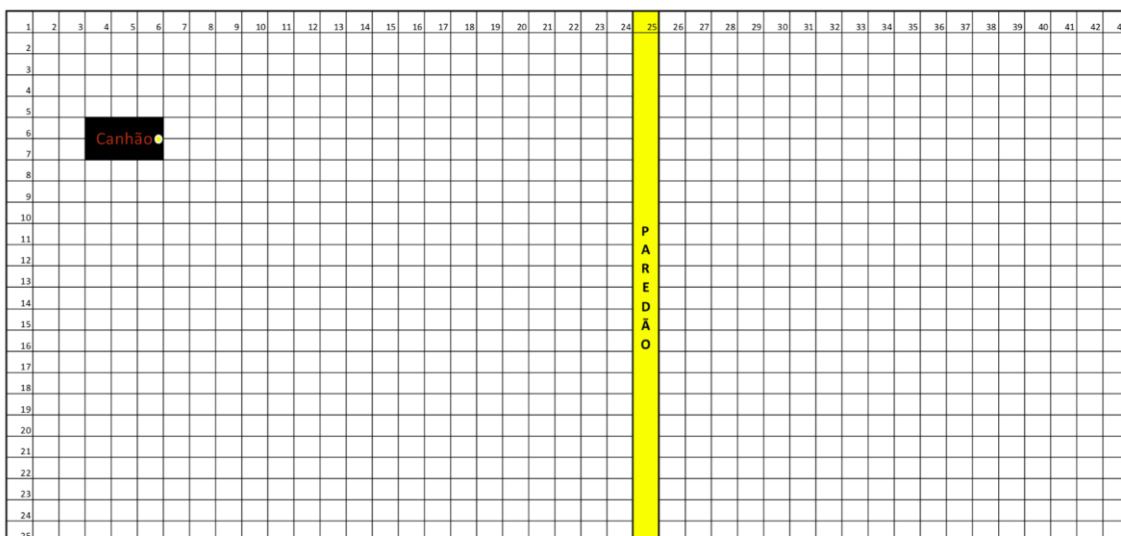


Figura - Vista Superior do Cenário

Veículo

O veículo possui um lançador de projéteis articulado, formado, por uma base (paralelepípedo) e dois **segmentos** cilíndricos. A base do veículo deve ter comprimento de 3m x largura de 2m.

A figura a seguir apresenta uma vista ortográfica do veículo.

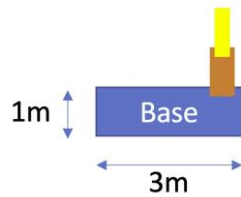


Figura - Vista Lateral do Veículo

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

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 força máxima deve ser tal que com um ângulo de inclinação de 45 graus, deve ser possível passar sobre o paredão, estando a 12 metros dele.

Projéteis

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

O movimento dos projéteis devem seguir a formulação da física de **lançamento de projéteis**. A direção inicial do projétil é dada pela direção do lançador.

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.

A velocidade do projétil é de 10m/s. **Como isto, o projétil atravessa o cenário, em sua dimensão maior, 5 segundos.**

Objetos do Cenário

O cenário deve conter pelo menos 40 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.