

Disciplina: Computação Gráfica
Professor: Marcio Sarroglia Pinho

Trabalho 02 – Robô Articulado Atirador

Felipe Freitas e Lucas Wolschick

Porto Alegre

2023

Sumário

1. Introdução.....	4
2. Critérios de Avaliação	5
2.1 Desenho do Paredão.....	5
2.2 Detecção da colisão do projétil com o paredão e com os objetos do cenário	5
2.3 Reconfiguração do Paredão após a colisão com o projétil.....	5
2.4 Modelagem do Veículo com as articulações móveis.....	5
2.5 Movimentação do Veículo - girar e andar para frente	6
2.6 Lançamento do Projétil, com base no ângulo do canhão e do veículo	6
2.7 Exibição correta dos objetos no formato TRI	6
3. Imagens do Desenvolvimento	8
3.1 Menu Terminal de Boas-Vindas.....	8
3.2 Vista Inicial (Lateral) do Cenário	8
3.3 Vista Inicial (3ª pessoa) do Cenário	9
3.4 Movimentação do veículo e canhão.....	9
3.5 Paredão após receber 3 tiros centrais	10
3.6 Espaço aberto para o veículo passar	10
3.7 Canhão do outro lado da parede, mirando em vacas	11
3.8 Veículo atropelando vacas.....	11
3.9 Visão lateral dos pontos de controle e curva de bézier	12
3.10 Terminal após uso simulado	12

1. Introdução

Neste trabalho, foi desenvolvido um “minijogo” no qual o jogador é um veículo articulado capaz de lançar projéteis com o objetivo de destruir todos os “adversários”, sendo estes objetos 3D formados por triângulos, localizados atrás de um grande paredão. O jogador poderá destruir o paredão para então ter mais certeza de acertar apenas os adversários, ou mesmo arriscar lançar o projétil por cima, com o risco de eliminar um amigo. O propósito deste relatório é demonstrar e analisar a implementação de algumas especificações de entrega do trabalho, cada uma especificada a seguir.

Um vídeo demonstrando uma iteração sobre o cenário e algumas funcionalidades do jogo está disponível em <https://www.youtube.com/watch?v=2jJOvFsd2k>.

2. Critérios de Avaliação

2.1 Desenho do Paredão

No processo de construção do paredão, utilizamos o método 'desenhaLadrilho', ajustando o ângulo para -90° no eixo Y, permitindo criar uma malha como o chão, porém na vertical. Introduzimos uma matriz booleana para representar o paredão em uma 'grade', que indica se cada ladrilho está destruído ou não. Ao mapear os pontos dos ladrilhos, adicionamos a textura correspondente à divisão da imagem durante a criação de cada um. Conforme especificações, o paredão foi criado com dimensões 25x15 e possui uma textura semelhante a tijolos, na qual cada ladrilho compõe parte da imagem.

2.2 Detecção da colisão do projétil com o paredão e com os objetos do cenário

O cálculo do projétil foi feito por meio de uma curva Bézier, e foi definido que o projétil seria desenhado no máximo 50 vezes. A cada desenho, testamos se: o projétil colide primeiramente com o veículo, o que acontece se o ângulo do canhão for de 90° ; se ele colide com o paredão, o que acontece caso a esfera possua um z (profundidade) igual ao do paredão (metade do terreno) e, claro, o ladrilho do paredão na posição x, y da esfera esteja intacto, do contrário o tiro atravessa; há colisão com o chão (e aqui incluímos também além do cenário, para isso nós diminuimos o limite do ponto final (alvo) da Bézier em 10 unidades no Y, para que o tiro sempre termine no chão); se existe algum objeto no caminho, neste caso vemos se o tiro está entre a coordenada central do objeto com alguma margem, por exemplo, de valor 2, então se o projétil tiver estiver a um raio de 2 unidades para fora do centro do objeto, destruimos o objeto; caso não haja nenhuma colisão, aumentamos o parâmetro da curva em 1/50 e repetimos o cálculo para a próxima esfera.

2.3 Reconfiguração do Paredão após a colisão com o projétil

No caso de haver sido constatada colisão com o paredão (ver seção 2.2), partimos para a remoção do ladrilho em questão e seus vizinhos. Com o ponto de colisão separado, pegamos suas coordenadas X e Y e utilizamos a função *round()* para determinar qual o ladrilho central que foi atingido. Voltamos à matriz booleana (citada na seção 2.1) e assinalamos este ladrilho como estando destruído, para que ele não seja exibido no próximo frame. Agora, basta realizar matemáticas simples e validações de limite, isto é, para cada um dos 8 vizinhos do ladrilho central verificamos que ele está dentro dos limites de desenho (para não destruir sempre 8 ladrilhos, e sim apenas aqueles adjacentes ao central) e aumentamos/diminuímos a coordenada X e Y do próximo ponto, marcando-o também como destruído na grade.

2.4 Modelagem do Veículo com as articulações móveis

Para modelar o veículo com duas articulações móveis, desenhamos 2 paralelepípedos, um em cima do outro, a base do veículo e o canhão (ver imagem 3.3). Ao pressionar as teclas A e D, alteramos o ângulo de rotação do veículo no eixo Y e redesenhamos o veículo no frame seguinte com a rotação adequada (ver imagem 3.4). O mesmo acontece para o cano, porém com algumas validações a mais, ou seja, o ângulo não pode ser superior a 90° (atirar para trás) ou inferior a 45° (atirar para baixo). Esta modelagem inclui também a texturização dos veículos e cálculo das normais para iluminação.

2.5 Movimentação do Veículo - girar e andar para frente

Após o ângulo da base do veículo (articulação 1) ter sido alterado (ver seção 2.4), criamos um ponto que aponta para a frente do canhão (ponta do cano), e rotacionamos este ponto no ângulo definido. Este ponto rotacionado é o alvo do veículo e vamos multiplicá-lo pela distância de movimento do veículo para determinar qual o destino desejado do veículo. Verificamos que ele não está indo para além dos limites do cenário e, principalmente, verificamos se ele não está colidindo com o paredão. Se ele for colidir com o paredão ou sair do cenário ele não se move até ajustar o ângulo para alguma movimentação possível, ou então, andar para trás. Ainda, sobre a colisão com objetos, adicionamos a opção de “atropelar” objetos (amigos ou inimigos), ou seja, se houver colisão entre o veículo e um objeto, retiramos o objeto de cena e somamos/decrementamos os pontos adequados.

2.6 Lançamento do Projétil, com base no ângulo do canhão e do veículo

Conforme especificado e demonstrado na imagem 3.9, a trajetória é definida de acordo com a direção do veículo, inclinação do canhão e força do disparo. Todos estes parâmetros são dinâmicos e foram baseados nos códigos do professor tanto para cálculo dos pontos na curva Bézier quanto para definição dos pontos de controle (força e distância). Infelizmente, existe um “bug” no código que não nos permite visualizar o projétil pois ele é muito rápido, então ao pressionar a tecla ESPACO o projétil é atirado na direção apontada, porém, é possível que ele apenas de um feedback através do paredão, inimigo destruído ou informação de pontuação no terminal. Caso fique pressionada a tecla espaço, apresenta todos os pontos da trajetória.

2.7 Exibição correta dos objetos no formato TRI

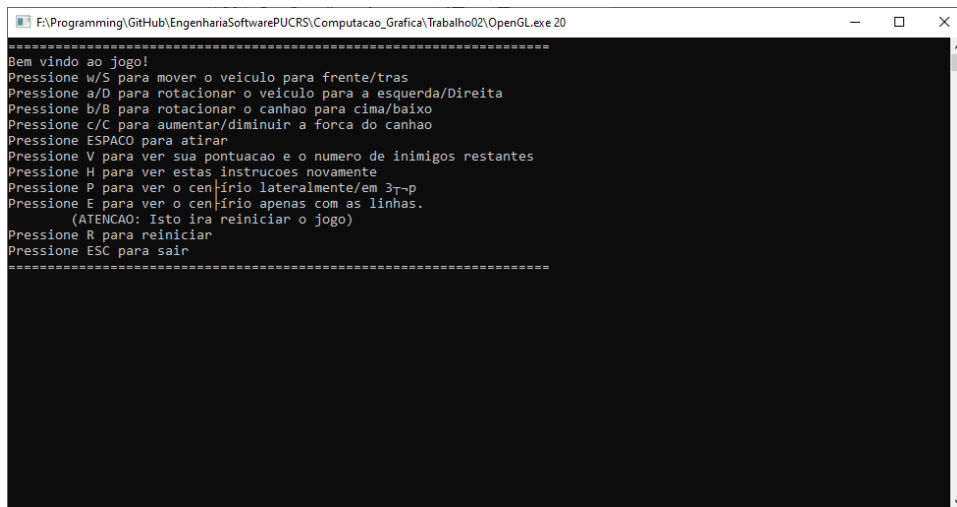
Para exibir os arquivos no formato .TRI, conforme especificado, utilizamos o código disponibilizado pelo professor que continha alguns objetos de exemplo. Todos esses objetos, a exceção da vaca, possuíam uma cor atrelada, para o qual a classe de leitura de Objeto não estava preparada. Como estávamos preocupados com outras funcionalidades, optamos por não adaptar a classe ou remover a cor e utilizar apenas a Vaca.tri como objetos amigos e inimigos, e representamos cada uma por uma cor, sendo verde amigo e vermelho inimigo.

Conforme especificado, a normal foi calculada utilizando o produto vetorial para cada

face do objeto e salva no Objeto3D. Ainda, foram renderizados 20 objetos, 10 amigos e 10 inimigos, sendo que suas posições iniciais eram geradas aleatoriamente do lado de trás da parede.

3. Imagens do Desenvolvimento

3.1 Menu Terminal de Boas-Vindas

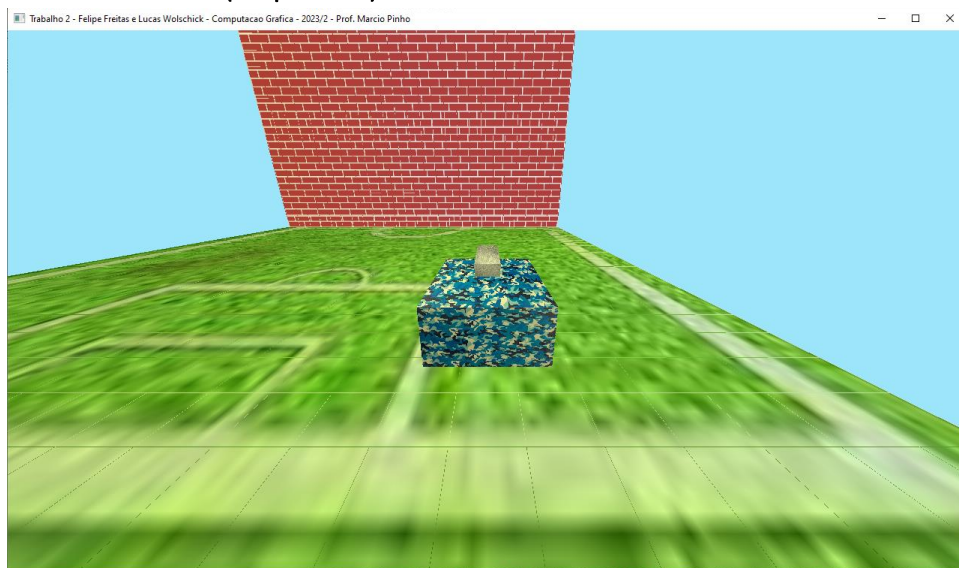


```
F:\Programming\GitHub\EngenhariaSoftwarePUCRS\Computacao_Grafica\Trabalho02\OpenGL.exe 20
=====
Bem vindo ao jogo!
Pressione w/S para mover o veiculo para frente/tras
Pressione a/D para rotacionar o veiculo para a esquerda/Direita
Pressione b/B para rotacionar o canhao para cima/baixo
Pressione c/C para aumentar/diminuir a forca do canhao
Pressione ESPACO para atirar
Pressione V para ver sua pontuacao e o numero de inimigos restantes
Pressione H para ver estas instrucoes novamente
Pressione P para ver o cen-irio lateralmente/em 3~p
Pressione E para ver o cen-irio apenas com as linhas.
      (ATENCAO: Isto ira reiniciar o jogo)
Pressione R para reiniciar
Pressione ESC para sair
=====
```

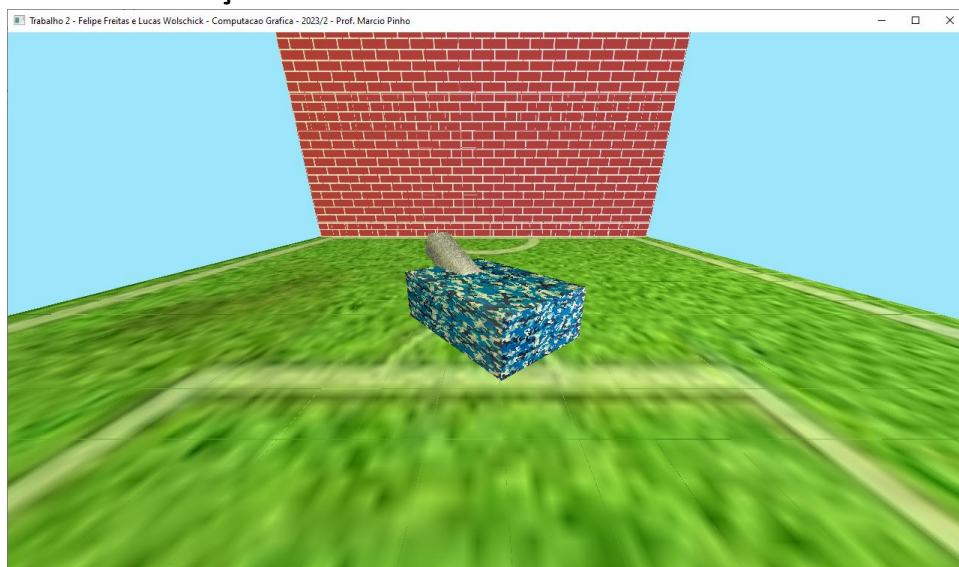
3.2 Vista Inicial (Lateral) do Cenário



3.3 Vista Inicial (3ª pessoa) do Cenário



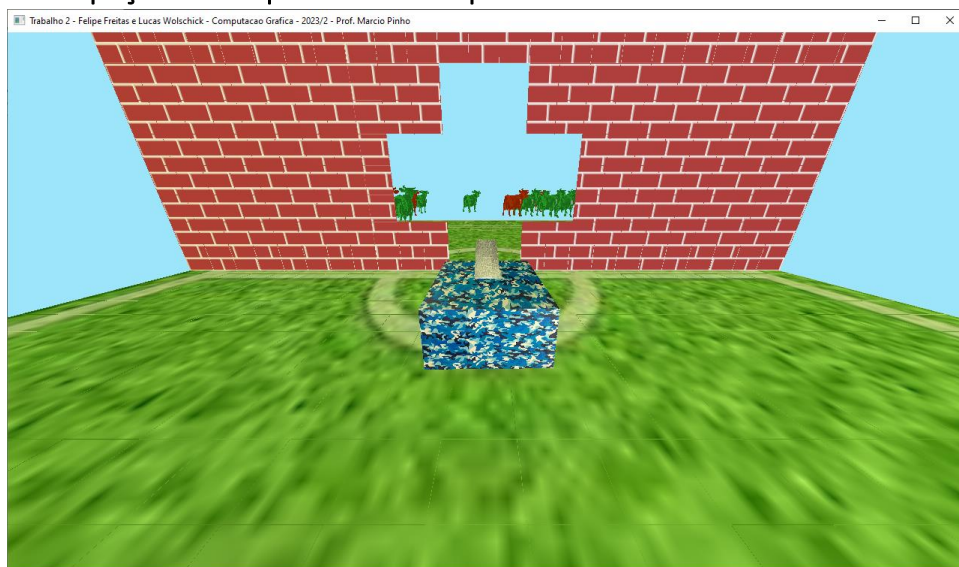
3.4 Movimentação do veículo e canhão



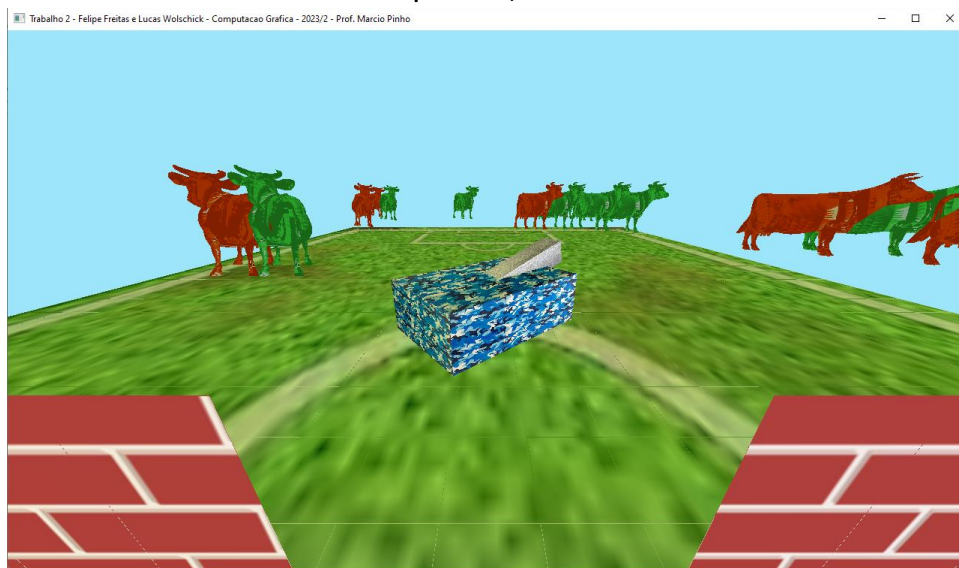
3.5 Paredão após receber 3 tiros centrais



3.6 Espaço aberto para o veículo passar



3.7 Canhão do outro lado da parede, mirando em vacas



3.8 Veículo atropelando vacas



3.9 Visão lateral dos pontos de controle e curva de bézier



3.10 Terminal após uso simulado

```
F:\Programming\GitHub\EngenhariaSoftwarePUCRS\Computacao_Grafica\Trabalho02\OpenGL.exe 20
=====
Bem vindo ao jogo!
Pressione w/S para mover o veiculo para frente/tras
Pressione a/D para rotacionar o veiculo para a esquerda/Direita
Pressione b/B para rotacionar o canhao para cima/baixo
Pressione c/C para aumentar/diminuir a forza do canhao
Pressione ESPACO para atirar
Pressione V para ver sua pontuacao e o numero de inimigos restantes
Pressione H para ver estas instrucoes novamente
Pressione P para ver o cen-irio lateralmente/em 3~p
Pressione E para ver o cen-irio apenas com as linhas.
(ATENCAO: Isto ira reiniciar o jogo)
Pressione R para reiniciar
Pressione ESC para sair
=====
Voce acertou no paredao : )      +5 pontos
Voce atropelou um inimigo : )    +10 pontos
Voce atropelou um amigo : (      -10 pontos
Voce atropelou um inimigo : )    +10 pontos
Voce matou um inimigo : )        +10 pontos
Voce atropelou um inimigo : )    +10 pontos
Voce matou um inimigo : )        +10 pontos
Voce matou um inimigo : )        +10 pontos
Voce matou um inimigo : )        +10 pontos
Voce matou um amigo : ( -10 pontos
Voce matou um inimigo : )        +10 pontos
Voce matou um amigo : ( -10 pontos
Voce matou um amigo : ( -10 pontos
Voce matou um inimigo : )        +10 pontos
Voce acertou no paredao : )      +5 pontos
Voce matou um inimigo : )        +10 pontos
Voce matou todos os inimigos : )
Game Over
Pontuacao final: 70
Pressione ESC para sair
Pressione P para ver o cen-irio lateralmente
Pressione E para ver o cen-irio apenas com as linhas
Pressione R para reiniciar
```