Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS) – Escola Politécnica
Disciplina de Computação Gráfica
Professor: Márcio S. Pinho
Alunos: Ricardo B. Süffert e Vinícius C. Turani

RELATÓRIO DO TRABALHO II

O VÍDEO DE DEMONSTRAÇÃO PODE SER ACESSADO AQUI.

NOTA: O trabalho descrito neste relatório consiste na implementação de um jogo com a biblioteca OpenGL, consistindo no segundo (e último) trabalho para a disciplina de Computação Gráfica. Informações sobre o jogo podem ser encontradas na página com a descrição do trabalho.

1. **COMANDOS (TECLAS) DO JOGO:**
   a. A **movimentação do canhão** é dada pelas teclas W, A, S e D, conforme o padrão para jogos de computador na indústria;
   b. O **ângulo do cano do canhão** é controlado pelas teclas ↑ (*arrow up*, para subir) e ↓ (*arrow down*, para descer);
   c. O **controle da força do tiro** é definido utilizando as teclas → (*arrow right*, para aumentar) e ← (*arrow left*, para diminuir);
   d. O **disparo dos tiros** é feito por meio da tecla espaço.
2. **DESENHO DO PAREDÃO:**

A estratégia utilizada para o desenho do paredão foi a aplicação de transformações geométricas sobre o piso disponibilizado, além do redimensionamento para atender ao tamanho especificado no enunciado do trabalho. Além disso, também foram aplicadas cores no lugar das texturas, que eram um requisito inicial do trabalho, mas foram removidos pelo professor. O resultado final foi o cenário a seguir:

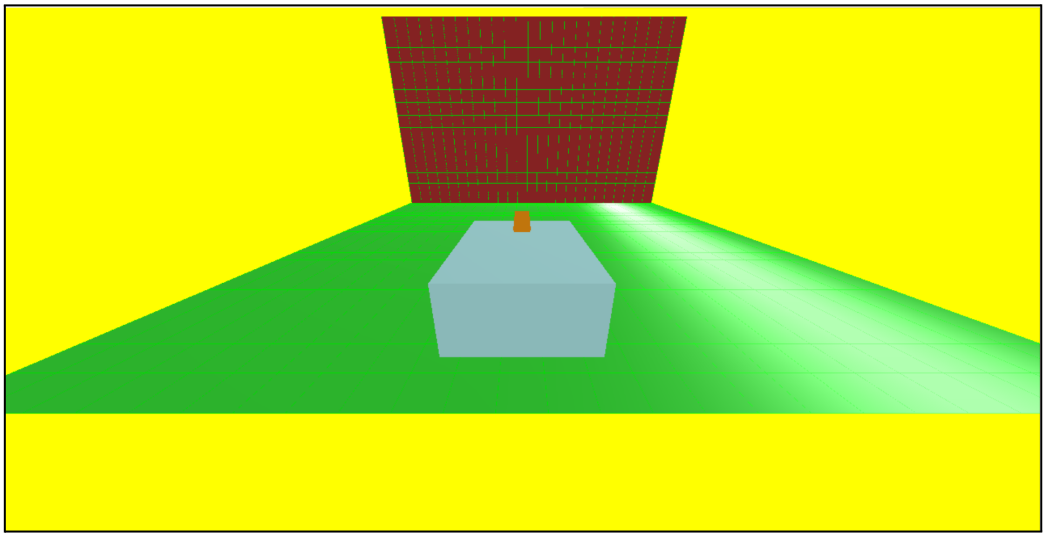

Figura 1 – visão geral do cenário modelado (com o paredão)

---

3. **COLISÃO COM OS AMIGOS/INIMIGOS, COM O PAREDÃO E COM O CHÃO, E RECONFIGURAÇÃO DO PAREDÃO APÓS COLISÃO:**

Conforme o paredão é atingido, são destruídos o bloco do paredão que foi atingido, bem como os oito (8) blocos ao redor, e o sistema concede cinco pontos ao usuário, em conformidade com a especificação do trabalho. O cálculo da colisão com o paredão é feito calculando o deslocamento do tiro a partir do ponto localizado no canto superior esquerdo do paredão. Uma demonstração da funcionalidade acima pode ser vista na imagem abaixo:

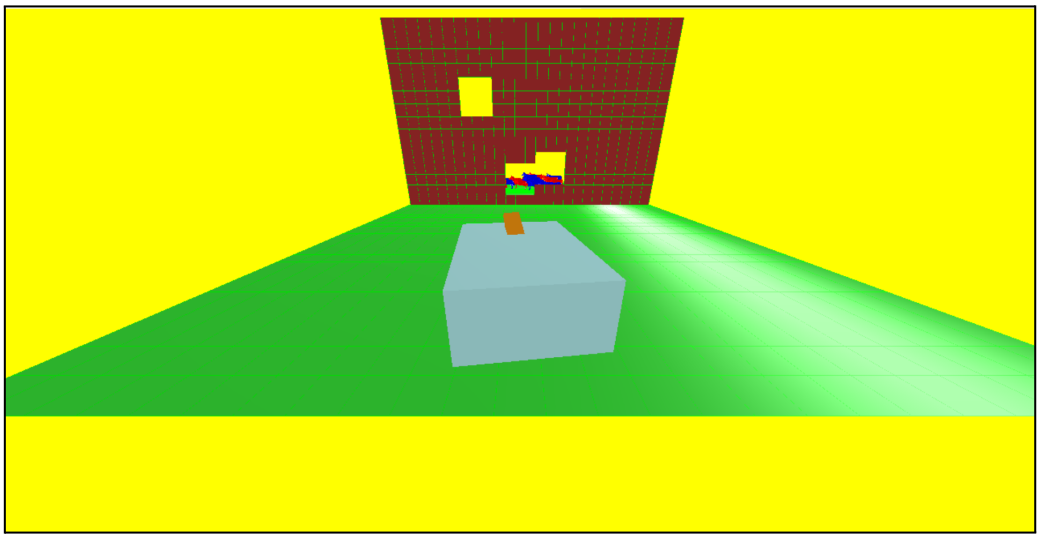

Figura 2 – paredão com buracos após ser atingido pelos disparos do canhão

Na imagem, o paredão foi atingido cinco vezes, resultando em 15 pontos sendo concedidos ao usuário, conforme o *output* de terminal abaixo:

```
Colisao com o muro, voce ganhou 5 pontos
Voce tem 5 pontos

Colisao com o muro, voce ganhou 5 pontos
Voce tem 10 pontos

Colisao com o muro, voce ganhou 5 pontos
Voce tem 15 pontos
```

Figura 3 – saída padrão demonstrando a detecção de colisão com o paredão

Além disso, os objetos do cenário que devem ser atingidos estão representados por vacas. Há 20 vacas no cenário, posicionadas atrás do paredão, sendo metade delas “amigas” (azuis) e metade delas “inimigas” (vermelhas). Ao atingir uma vaca inimiga, são concedidos 10 pontos; por outro lado, acertar uma vaga amiga resulta na perda de 10 pontos e, em ambos os casos, a vaca atingida desaparece. Ademais, a colisão com o solo, calculada verificando se a coordenada *y* do projétil está abaixo da coordenada *y* do canto superior esquerdo do chão, resulta em uma perda de 5 pontos ao usuário. A dinâmica de atingir aos amigos/inimigos é demonstrada na sequência de imagens abaixo, em que uma vaca inimiga é atingida e, então, ela desaparece e 10 pontos são concedidos:

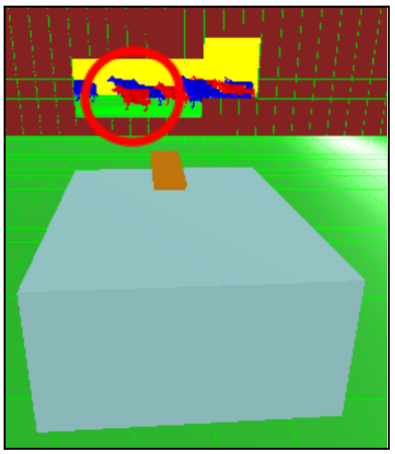

Figura 4 – demonstração de uma vaca inimiga no cenário

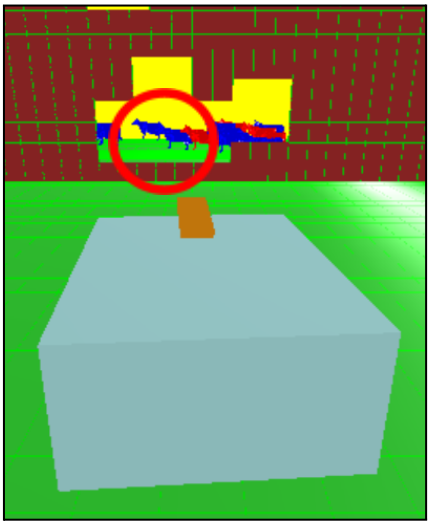

Figura 5 – demonstração de que, ao ser atingida, a vaca inimiga desapareceu

```
Vaca inimiga atingida, Voce ganhou 10 pontos
Voce tem 20 pontos
```

Figura 6 – saída padrão demonstrando que 10 pontos foram concedidos ao atingir a vaca inimiga

De forma análoga, ao atingir uma vaca amiga, o usuário é penalizado com a perda de 10 pontos. Isso é demonstrado no *output* abaixo:

```
Vaca amiga atingida, Voce perdeu 10 pontos
Voce tem 10 pontos
```

Figura 7 – saída padrão demonstrando que 10 pontos foram subtraídos ao atingir vaca amiga

---

4. MODELAGEM DO VEÍCULO:

O veículo foi modelado utilizando dois paralelepípedos, sendo um cinza (representando a base do canhão) e outro marrom (representando o cano do canhão). Os paralelepípedos foram redimensionados por meio da aplicação de escala para estarem em conformidade com as dimensões estabelecidas no enunciado do trabalho. Nas imagens abaixo, é possível ter uma visão traseira, uma visão lateral e uma visão frontal do canhão modelado:

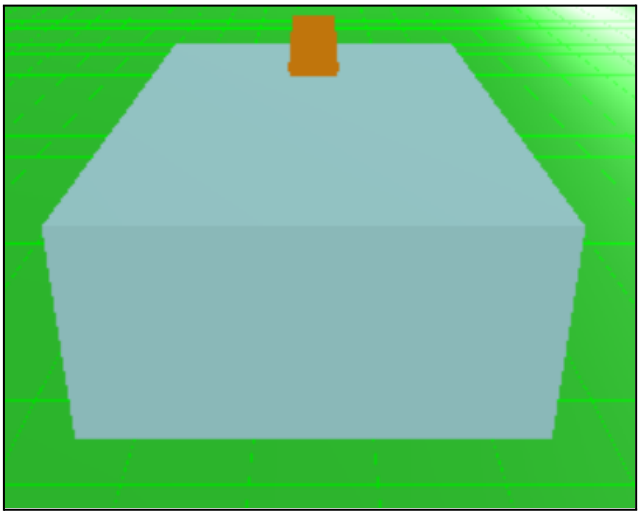

Figura 8 – vista traseira do canhão modelado

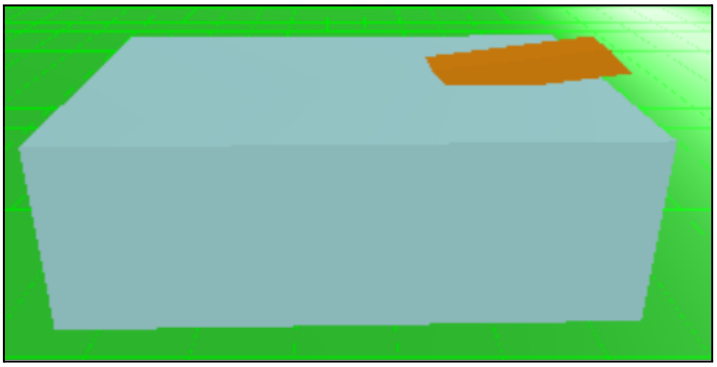

Figura 9 – vista lateral do canhão modelado

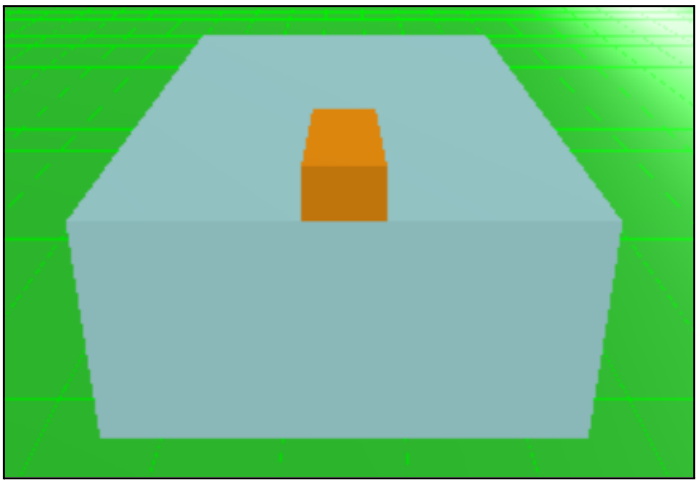

Figura 10 – vista frontal do canhão modelado

---

5. MOVIMENTAÇÃO DO VEÍCULO:

Por meio das teclas W, A, S e D, o canhão apresentado no item anterior pode ser deslocado ao redor do cenário. Além disso, o ângulo do cano do canhão com relação à base pode ser aumentado e diminuído utilizando as teclas *arrow up* e *arrow down*, respectivamente. Para a movimentação do canhão para frente e para trás no cenário, foi utilizado um ponto que se desloca juntamente com o canhão conforme este anda para frente e para trás. Quando a base do canhão é rotacionada, o ponto que representa o alvo também é, de modo que este esteja sempre exatamente à frente do canhão e sempre à mesma distância dele. Além disso, para movimentar o cano do canhão, é aplicada uma rotação no paralelepípedo marrom. Nas imagens abaixo, é possível visualizar o deslocamento do canhão no cenário e a movimentação do cano do canhão.

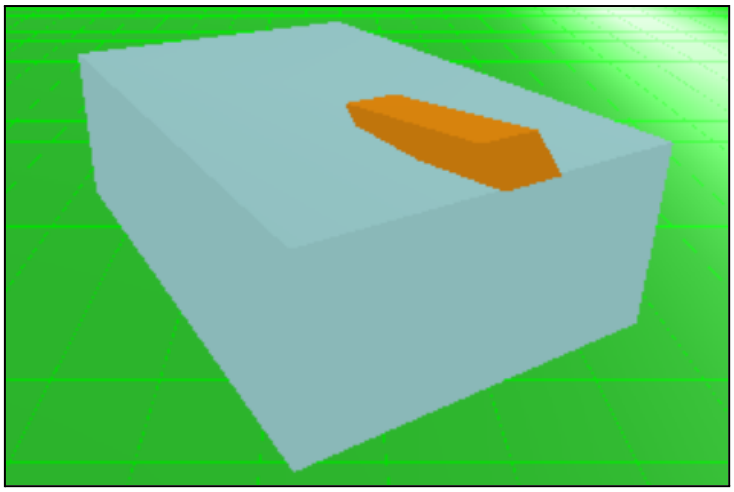

Figura 11 – cano do canhão na posição inicial

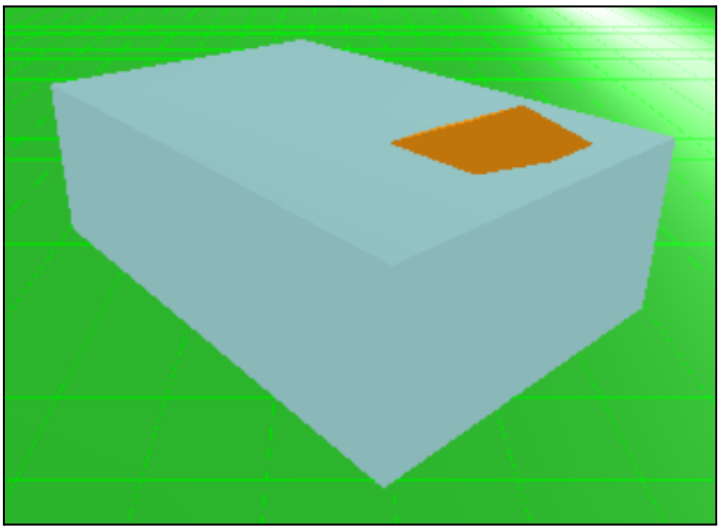

Figura 12 – cano do canhão rotacionado para cima

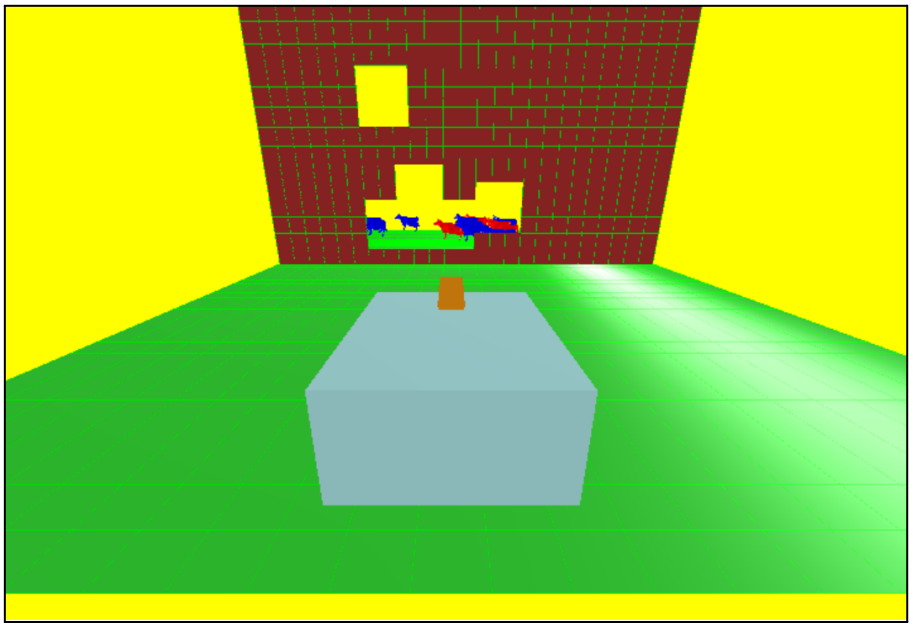

Figura 13 – canhão na posição inicial

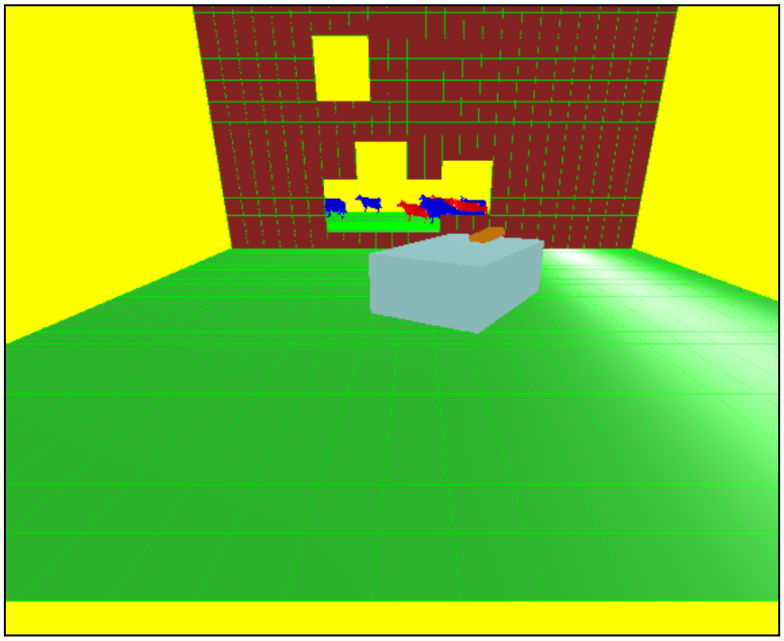

Figura 14 – canhão rotacionado e movimentado para perto do paredão

6. TRAJETÓRIA DO PROJÉTIL:

Para o cálculo da trajetória do projétil, foram gerados três pontos de controle no espaço de modo a traçar uma curva de Bézier descrevendo o movimento do projétil. O ponto inicial da trajetória está localizado na posição canhão, o ponto médio é calculado de acordo com o ângulo de rotação do cano do canhão no eixo *x* e o ponto final da trajetória indica a posição final do tiro — ou seja, o local onde ele irá colidir com o chão, caso não haja nenhuma colisão no caminho. A trajetória do tiro pode ser descrita de forma aproximada pelas seguintes linhas:

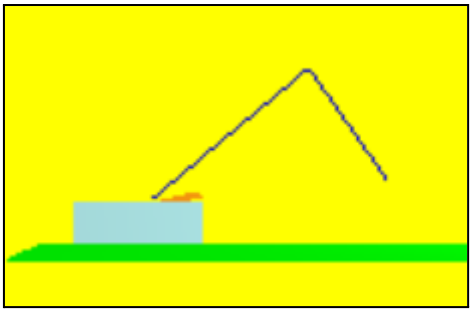

Figura 15 – descrição aproximada da trajetória do tiro

Percebe-se, ainda, que, conforme o ângulo de rotação do cano do canhão aumenta, os pontos de controle se adaptam para refletir a trajetória da curva corretamente. Tal comportamento pode ser observado na imagem abaixo:

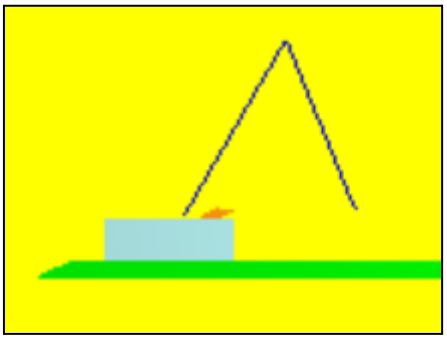

Figura 16 – demonstração de que a curva se adapta conforme o ângulo de rotação do canhão aumenta

7. EXIBIÇÃO DOS OBJETOS (FORMATO TRI):

Por fim, para a leitura do arquivo TRI, foi utilizado como biblioteca um módulo visto e elaborado em um exercício de aula, presente no código com o nome `LeitorObjeto3D`. Foi implementado, também, um método chamado `ExibeObjeto`, capaz de, após a leitura do arquivo, renderizar o objeto na tela, iterando sobre seus triângulos, calculando seus vetores normais e os exibindo na tela. Esse processo resultou em objetos como este sendo exibidos no cenário:

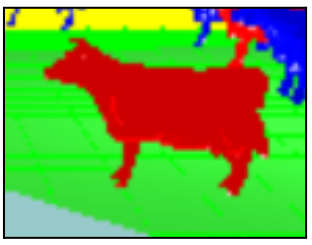

Figura 17 – exemplo de vaca renderizada no cenário