

Simulador de Treinamento de Tênis

Nome: Renan Ryu Kajihara, NUSP: 14605762

25 de junho de 2025

Resumo

O presente relatório irá descrever as atividades realizadas no Projeto Final da disciplina "Introdução a Computação Gráfica (MAC0420)", que consistiu em um simulador 3D interativo de tênis, desenvolvido utilizando a API do WebGL2, que combina várias técnicas aprendidas ao longo do semestre como: modelagem de objetos, texturização, iluminação de Phong, interatividade em tempo real e sombreado. O sistema permite ao usuário controlar um personagem, rebater uma bola, mirar em alvos e interagir com elementos realistas de uma quadra de tênis.

Conteúdo

1	Introdução	3
2	O que o sistema faz	3
2.1	Mini Manual do Usuário	3
2.2	Como executar o código	3
3	Como o sistema foi desenvolvido	4
3.1	Modelagem dos objetos	4
3.2	Implementação das interações do personagem com o usuário	4
3.3	Implementação da interação da bola com os outros objetos	5
3.4	Implementação da luz seguindo o modelo de Phong	5
3.5	Implementação do sombreado	5
3.6	Implementação de texturas	5
3.7	Alterações em relação a proposta de 15/05	6
4	Descrição das atividades realizadas por cada membro	6
5	Bugs e Problemas Conhecidos	6

1 Introdução

O trabalho consistiu na criação de um jogo que modela um ambiente 3D que simula um treinamento de tênis, na qual o usuário deve acertar o alvo o máximo de vezes possível para acumular pontos, sem que ele deixe a bola impossível de ser golpeada.

Este trabalho foi desenvolvido utilizando a API do WebGL2 e combinou diversas técnicas apresentadas nas aulas de Introdução à Computação Gráfica como: interatividade do usuário, modelagem de objetos, iluminação de Phong, movimentação da câmera, texturização e sombreamento.

2 O que o sistema faz

Primeiramente, o sistema simula um jogador em uma quadra de tênis 3D, onde há uma parede de tijolos e um alvo posicionado à frente dessa parede. O usuário controla o boneco, podendo movimentá-lo lateralmente e realizar golpes de raquete para rebater a bola. O principal objetivo é acertar o alvo o maior número de vezes possível, utilizando a bola de tênis. Cada vez que a bola atinge o alvo, o jogador marca um ponto e o alvo muda de posição, aumentando o desafio. A bola pode quicar no chão, rebater na parede de tijolos (que escurecem e desaparecem com o impacto), e interagir fisicamente com a raquete e o ambiente, proporcionando uma experiência dinâmica e realista. O sistema também apresenta animações suaves, texturização diferenciada para cada objeto e feedback visual para as ações do usuário.

2.1 Mini Manual do Usuário

O objetivo do jogador é acertar o alvo o maior número de vezes possível, marcando pontos. Cada vez que o alvo é atingido, ele se move para uma nova posição aleatória na parede, tornando o desafio mais dinâmico. Ao acertar o muro de tijolos, os tijolos atingidos escurecem e enfraquecem, podendo desaparecer após algumas colisões.

Para controlar o boneco, é possível utilizar as teclas:

- j : move o personagem para a esquerda;
- l : move o personagem para a direita;
- espaço : faz com que o personagem faça o movimento de um forehand.

O usuário pode aumentar ou diminuir a velocidade do boneco por meio do slider disponível abaixo da tela do jogo.

A posição em que a raquete colide com a bola é fundamental para a jogabilidade: quanto mais próximo da parede a bola for rebatida, mais para a esquerda ela seguirá; quanto mais distante da parede, mais para a direita ela irá.

O usuário perde se a bola passar para trás do jogador ou atravessar para o outro lado da parede. Para reiniciar o jogo, basta clicar na tela. A pontuação atual pode ser acompanhada no canto superior esquerdo da tela.

2.2 Como executar o código

- 1- Primeiramente, certifique-se de ter um navegador moderno com suporte a WebGL2 e HTML5.
- 2- Caso não seja possível utilizar as texturas disponível no .zip, é necessário apagar as texturas presentes no arquivo zip e baixá-las diretamente na mesma pasta que os arquivos index.html e pf.js:
 - baixe a textura disponível no link: <https://br.pinterest.com/pin/197947346115374138/> e renomeie o arquivo para palmeiras.jpg;
 - baixe a textura disponível no link: <https://stock.adobe.com/images/red-black-circles-focus-target-style-concept-pattern-colorful-design-structure-decoration-abstract-geometric-background-illustration-fashion-look-backdrop-wallpaper-abstract-decoration-graphic/243362076> e renomeie o arquivo para textura_alvo.jpg;
- 3- Abra o arquivo 'index.html' da pasta 'Projeto_Final' em seu navegador.
- 4- O sistema será carregado automaticamente e estará pronto para uso.

3 Como o sistema foi desenvolvido

O sistema foi desenvolvido em várias etapas, sendo elas: modelagem dos objetos que compõem a cena, implementação das interações do personagem com o usuário, implementação das interações da bola com os outros objetos, adição de luzes de Phong, adição de sombreadamento e adição de texturas.

3.1 Modelagem dos objetos

A cena é composta por 6 elementos principais: parede de tijolos, alvo, chão, boneco, bola e raquete. Todos os elementos da cena foram modelados manualmente utilizando primitivas básicas, como paralelepípedos e esferas. A composição de cada um desses elementos é feita da seguinte forma:

- parede de tijolos: formada por 1200 tijolos cúbicos;
- alvo: formada por uma esfera achatada;
- chão: formada por um paralelepípedo de grande área e pequena altura;
- boneco: composto por 6 paralelepípedos, que representam a cabeça, as duas pernas, os dois braços e o torso;
- bola: modelada como uma esfera;
- raquete: composta por um paralelepípedo para representar o cabo e uma esfera achatada para representar a cabeça da raquete.

Os objetos que compõe a cena podem ser vistos na seguinte imagem:

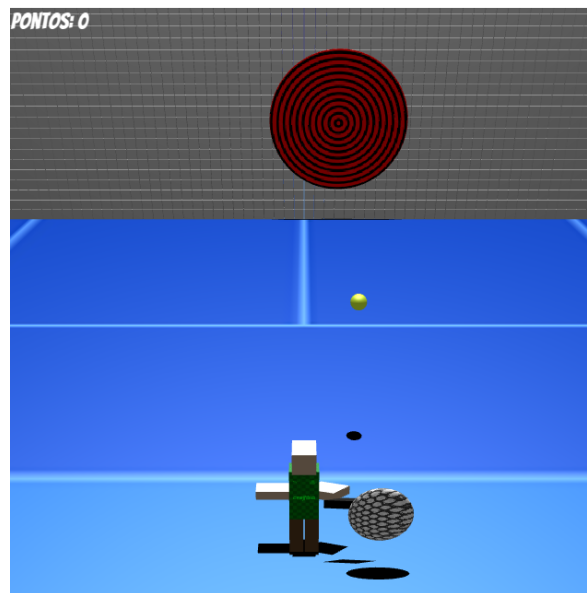


Figura 1: Modelagem dos objetos da cena

3.2 Implementação das interações do personagem com o usuário

As interações que o usuário possuem com o personagem são: movimentação lateral e animação da raquetada.

A movimentação lateral do boneco foi feita alterando a posição X de todas as partes do boneco, mantendo a coesão do movimento. Além disso, a câmera, que foi modelada como uma esfera que não é renderizada, assim como no EP3, se movimenta junto do personagem, modificando sua posição X de forma igual. Ademais, o usuário pode aumentar a velocidade do boneco por meio de um slider, modificando o tamanho do passo no eixo X que o boneco fará a cada interação.

A animação da raquetada foi feita modificando o ângulo do braço direito do personagem e da raquete no eixo Y. Nesse sentido, o ângulo do braço é interpolado de -30° até 90° em torno do eixo Y,

simulando o movimento real do forehand. Após o golpe, o braço retorna suavemente à posição inicial, realizando a rotação reversa também ao redor do eixo Y.

Além disso, o jogador perde se a bola passar para trás do boneco (posição Z maior que a posição Z do boneco + threshold) ou atravessar a parede de tijolos (posição Z menor que 0).

3.3 Implementação da interação da bola com os outros objetos

A movimentação da bola é feita por meio de vetores de velocidade e aceleração (gravidade). Nesse sentido, a atualização de sua posição a cada frame é feita considerando a velocidade X e Y da bola, a aceleração da gravidade e as colisões da bola com os mais diferentes objetos.

A bola pode colidir com o chão, a raquete, os tijolos e o alvo.

Se a bola atinge o chão, sua posição Y é corrigida e a velocidade vertical é invertida e reduzida, simulando o quique da bola.

Já a colisão com a raquete é detectada pelo cálculo da distância entre o centro da bola e da cabeça da raquete. Se houver contato e não houve contato recente, a bola inverte a direção Z, recebe impulso lateral proporcional à posição do impacto, simulando efeito realista. Um cooldown entre as raquetadas impede múltiplos contatos em sequência em uma mesma raquetada.

Ao atingir a parede, a posição da bola é usada para identificar os tijolos afetados. O tijolo escurece a cada impacto e é removido se ficar totalmente escuro (cor (0,0,0)). O impacto também afeta tijolos vizinhos, com escurecimento proporcional à proximidade.

Por fim, quando a bola colide com o alvo (verificação por distância e sobreposição em Z), o jogador ganha ponto, o alvo muda de posição aleatória e a bola rebate.

3.4 Implementação da luz seguindo o modelo de Phong

O sistema utiliza o modelo de iluminação Phong, que calcula componentes ambiente, difusa e especular para cada fragmento, proporcionando realismo visual com brilho e sombreamento adequados. Nesse sentido, a cena conta com uma fonte de luz, que possui posição (0,800,10), componente ambiente (0.47,0.47,0.47), componente difusa (0.68, 0.68, 0.68) e componente especular (0.39, 0.39, 0.39). Além disso, cada objeto renderizado possui seu próprio material, com componentes que buscam criar realismo com o material original do objeto.

3.5 Implementação do sombreamento

Nesse projeto, o sombreamento foi aplicado para reforçar o efeito de profundidade e realismo, principalmente da bola. Nesse sentido, foi utilizado a matriz de projeção apresentada no capítulo 21 do livro da disciplina, em que as sombras dos objetos são projetadas no chão, acompanhando a posição da fonte de luz, e são desenhadas com cor preta para simular o efeito de sombra.

3.6 Implementação de texturas

Nesse projeto, 4 objetos possuem texturas: o alvo, a cabeça da raquete, o chão da quadra e o torso do boneco. As texturas são aplicadas via coordenadas UV e múltiplas unidades de textura (GL_TEXTURE0, GL_TEXTURE1, etc). O mapeamento UV foi ajustado para cada modelo, e as texturas são carregadas dinamicamente. Isso permite diferenciar visualmente cada elemento e aumentar a imersão.

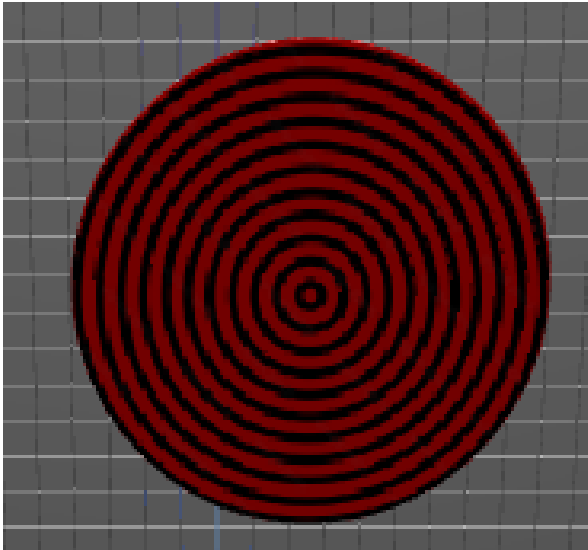


Figura 2: Textura do alvo



Figura 3: Textura do torso e da cabeça da raquete

3.7 Alterações em relação a proposta de 15/05

O projeto sofreu algumas alterações em relação a proposta enviada no dia 15/05, sendo elas: ausência do movimento para frente e para trás do jogador; implementação de sombras; e implementação de objetos além da quadra.

O movimento do jogador para frente e para trás não foi implementado intencionalmente para preservar a jogabilidade. Caso o personagem pudesse avançar ou recuar livremente, haveria situações em que a bola passaria acima ou abaixo da raquete, pois não haveria tempo suficiente para a gravidade ajustar a altura da bola em relação à raquete. Isso comprometeria a experiência do usuário, tornando os golpes imprecisos e dificultando o controle da partida. Dessa forma, a movimentação foi restrita ao eixo lateral, garantindo que o jogador esteja sempre alinhado com a trajetória da bola, favorecendo uma jogabilidade mais fluida e intuitiva.

A implementação de sombras e de texturas em mais objetos (torso, alvo e cabeça da raquete) tiveram como objetivo aumentar o realismo e a imersão usuário no jogo.

4 Descrição das atividades realizadas por cada membro

Por se tratar de um projeto individual, todas as atividades foram realizadas exclusivamente por Renan Ryu Kajihara.

5 Bugs e Problemas Conhecidos

Os bugs conhecidos são: podem ocorrer pequenas imprecisões na detecção de colisão em ângulos extremos e raramente a detecção de colisão da bola com a raquete pode apresentar atrasos ou parecer menos responsiva, devido à sobrecarga de processamento.