



Link dos códigos e funções do jogo: <https://drive.google.com/drive/folders/1Ve-NiyAk1SEeivyAjhsZdNwS2ngj1Ts7?usp=sharing>

Relatório: Jogo de Tiro ao Alvo feito com a biblioteca OpenGL

Universidade Federal Rural do Rio de Janeiro - UFRRJ

Discente: Gabriel Lima Prisco Madureira | Sistemas de Informação

Docente: Nilton José Rizzo

Disciplina: IC836 – Introdução à Computação Gráfica

1. INTRODUÇÃO

O programa é uma simulação de tiro ao alvo em objetos esféricos que se movimentam (relacionado à velocidade) e surgem de forma aleatória, porém é possível fazer surgir mais alvos ou reduzir a quantidade atual, também existe uma mira, movimentada pelo mouse, que quando pressionado o botão esquerdo dispara os tiros que irão explodir as esferas. Além disso, o jogo possui recurso de zoom (para chegar mais perto dos alvos) e também retroceder, incluindo movimentação da câmera e um record final, mostrado ao sair do jogo.

2. TABELA DE FUNÇÕES PADRÕES UTILIZADAS

Principais funções padrões utilizadas na construção do programa:

gluLookAt	Cria uma matriz de visualização derivada de um ponto de olho, um ponto de referência indicando o centro da cena e um vetor para cima.
glTranslated	Multiplica a matriz atual por uma matriz transladada.
glRotatef	Multiplica a matriz atual por uma matriz rotacionada.

glBindTexture	<p>Permite criar ou utilizar uma textura nomeada. Ela é chamada com os parâmetros escolhidos: GL_TEXTURE_1D, GL_TEXTURE_2D, GL_TEXTURE_3D, GL_TEXTURE_1D_ARRAY, GL_TEXTURE_2D_ARRAY, GL_TEXTURE_RECTANGLE, GL_TEXTURE_CUBE_MAP, GL_TEXTURE_CUBE_MAP_ARRAY, GL_TEXTURE_BUFFER, GL_TEXTURE_2D_MULTISAMPLE, GL_TEXTURE_2D_MULTISAMPLE_ARRAY e a textura definida com o nome da nova textura vincula o nome da textura ao destino. Quando uma textura é vinculada a um destino, a vinculação anterior para esse destino é quebrada automaticamente. Enquanto uma textura está vinculada, as operações GL no destino ao qual ela está vinculada afetam a textura vinculada e as consultas do destino ao qual ela está vinculada retornam o estado da textura vinculada. Com efeito, os alvos de textura tornam-se aliases para as texturas atualmente vinculadas a eles, e o nome da textura zero refere-se às texturas padrão que foram vinculadas a eles na inicialização.</p>
gluBuild2DMipmaps	<p>Constrói um "minimapa" com duas dimensões. Para funcionar, requer que a textura, em formato de imagem, já esteja armazenada nos arquivos do jogo.</p>
glTexEnvf	<p>Especifica ambiente de textura. Um ambiente de textura especifica como os valores de textura são interpretados quando um fragmento é texturizado.</p>
glTexParameterf	<p>Atribui o valor ou valores em "params" ao parâmetro de textura especificado como pname.</p>

glTexCoord2d	define as coordenadas de textura atuais que fazem parte dos dados associados aos vértices de polígono. A função glTexCoord especifica coordenadas de textura em uma, duas, três ou quatro dimensões. A função glTexCoord1 define as coordenadas de textura atuais como (s, 0, 0, 1); uma chamada para glTexCoord2 os define como (s, t, 0, 1). Da mesma forma, glTexCoord3 especifica as coordenadas de textura como (s, t, r, 1) e glTexCoord4 define todos os quatro componentes explicitamente como (s, t, r, q).
glNormal	A normal atual é definida para as coordenadas fornecidas sempre que glNormal é emitido. Argumentos de byte, short ou int são convertidos para o formato de ponto flutuante com um mapeamento linear que mapeia o valor inteiro representável mais positivo para 1.0 e o valor inteiro representável mais negativo para -1.0.

3. ARQUIVOS FONTES INCLUÍDOS NA FUNÇÃO MAIN

A. carrega.h - responsável por inicializar o game, chamando a função CarregaJogo(int CJFlag)

B. jogo.h - responsável pelo esqueleto do jogo, com informações sobre altura e largura da tela, bem como profundidade do corredor. chama também funções que preparam o ambiente do game.

C. movimentos.c - responsável por toda movimentação do jogo, ele compõe funções que testam quando os objetos esféricos estão encostando na parede, simulando o

efeito de "rebater" na parede, com as funções "MoveDireta", "MoveEsquerda" (relacionadas com a distância entre o corredor, transladando e rotacionando a matriz da esfera para ir para a direção oposta), "MoveFrente", "MoveTras" (está relacionado com a mira do jogador, com o "zoom" que o jogador pode realizar no jogo, alterando a percepção dos objetos). Também possui outras funcionalidades, como a movimentação dos tiros, desenhar o tiro e principalmente uma função MoveAlvos responsável por toda a dinâmica lógica do game, em que os alvos ficam aleatoriamente no fundo para serem acertados, devendo possuir uma física: serem arremessadas para o sentido oposto ao coliderem com a parede ou umas com as outras, sendo inicializadas a partir de um looping FOR que irá se repetir de acordo com a quantidade de inimigos presentes no cenário. Por fim, ele contém uma função que movimenta o alvo e a câmera, inclusive quando o tiro intercepta o alvo, devendo contar pontos para o record e também fazer sumir a esfera.

D. record.c - código responsável por escrever e armazenar em um arquivo binário informações referentes ao record do jogador, ou seja, armazena a quantidade de pontos que o jogador já fez durante todas as suas jogatinas, criando um ranking em ordem decrescente, possuindo uma função que sempre lê o ranking já armazenado e outra que irá gravar a nova pontuação feita pelo usuário.

E. textura.c - responsável pela textura do jogo, possui funções capazes de gerarem o efeito visual para o jogo. Importantes funcionalidades invocadas no código: glBindTexture, gluBuild2DMipmaps, glTexEnvf, glTexParameterf. Nesse caso, as texturas são trazidas externamente no game, na pasta "texturas" do arquivo do game.

F. objetos.c - contém funções que controlam o surgimento e desaparecimento das esferas(alvos), função que inicializa o corredor, luminária, placar e mira. A maioria das funções utiliza a variável Total_Inimigos_SCR para controlar os fluxos de controle e laços de repetição.

G. desenha.c - Utiliza, principalmente, funções contendo glVertex3d para desenhar toda a composição do jogo, como mira, esferas, luminárias... Além de inicializar as cores.

H. mouse.c - contém função que controla a ação do click e movimentação do mouse.

I. trajetória.c - calcula a trajetória do tiro disparado pela mira.

J. teclado.c:

Tecla:	Ação:
S ou s	Desloca a câmera para a direita
W ou w	Move a mira para frente, direção ao fundo
Z ou z	Move a mira para trás, (só funciona se já tiver deslocado para frente, não é possível se afastar mais do fundo do que o padrão, ao iniciar o jogo)
R ou r	Inicializa os alvos, ou seja, “recarrega” os alvos iniciais por novos
i	Diminui a quantidade de alvos disponíveis para -4
I	Aumenta a quantidade de alvos disponíveis para +4
c	Flag da mira muda para 0, voltando a câmera ao padrão (câmera não desloca enquanto o jogador movimenta o mouse). Formato padrão ao iniciar o jogo
C	Flag da mira muda para 1, câmera do jogo desloca enquanto o jogador movimenta o mouse

t	flag do tiro muda para 0, fazendo com que o tiro seja atirado proporcionalmente em relação ao fundo, e não necessariamente na posição da mira. Formato padrão ao iniciar o jogo
T	flag do tiro muda para 1, fazendo com que o tiro agora seja na direção em que a mira (controlada pelo mouse) está apontando, independentemente da relação com o fundo.