

Integrantes: Erik Monteiro e Lucca Iânez

Relatório do T2 – 2024/1

A) Principais Funcionalidades

A) 1. Definição e Renderização do Labirinto

O labirinto é carregado a partir de um arquivo, que define a estrutura do ambiente. Os diferentes tipos de elementos (paredes, janelas, portas, etc.) são representados por valores inteiros. As paredes podem ter diferentes orientações e espessuras.

A) 2. Funções principais

`void readMap(const char *filename):` Lê a configuração do labirinto a partir de um arquivo.

`void drawLabirinto():` Renderiza o labirinto na tela.

B) Projeção e Exibição

O programa permite alternar entre diferentes modos de projeção (paralela ortográfica e perspectiva) e modos de exibição (wireframe ou faces preenchidas).

B) 1. Funções principais:

`void PosicUser():` Define a posição da câmera e o tipo de projeção.

`void init(void):` Inicializa os parâmetros globais do OpenGL.

C) Colisão e Física

O sistema de colisão detecta interações entre o personagem e outros elementos do labirinto, como paredes e cápsulas de energia. A colisão é gerenciada por caixas delimitadoras (Bounding Boxes).

C) 1. Funções principais:

BoundingBox getBoundingBox(Ponto pos, float width, float height, float depth): Calcula a Bounding Box de um objeto.

bool detectCollision(BoundingBox a, BoundingBox b): Detecta colisão entre duas Bounding Boxes.

bool detectaColisaoParede(Ponto personagemPos): Detecta colisão entre o personagem e as paredes, lembrando que uma janela é uma parede com um 'furo' no meio, portanto, na função também comparamos com a janela.

void detectaColisaoCapsula(): Detecta colisão entre o personagem e as cápsulas de energia.

D) Personagem, Inimigos e Cápsulas de energia

O programa gerencia a posição e movimento do personagem e dos inimigos. O personagem é controlado pelo jogador, enquanto os inimigos se movem automaticamente em direção ao personagem. Há cápsulas de energia espalhadas pelo mapa de forma aleatória, caso o usuário colida, então sua energia é aumentada.

D) 1. Funções principais:

void desenhaPersonagem(): Renderiza o personagem na tela.

void desenhaInimigo(): Renderiza os inimigos na tela.

void moveInimigo(): Atualiza a posição dos inimigos, movendo-os em direção ao personagem.

O mapa do labirinto é gerado a partir da leitura de um arquivo .txt, que é formado por 50 linhas e 50 colunas no qual cada caractere representa:

0 -> Piso

1 -> Parede (tipo 1)

2 -> Parede (tipo 2)

3 -> Janela

4 -> Porta

5 -> Cadeira

6 -> Mesa

A função responsável por fazer a leitura desse arquivo é a **readMap(fileName)**, após chamar ela, guardamos na variável global **'labirinto'** cada caractere lido deste arquivo.

Na função **drawLabirinto()**, percorremos cada linha e coluna armazenados em **'labirinto'** e desta forma concluímos a criação do mapa.