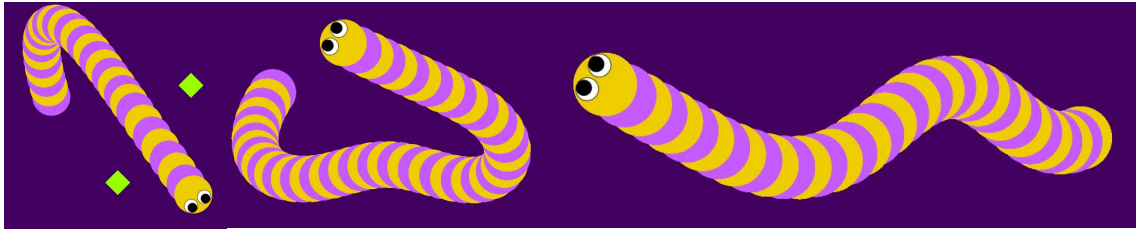


Trabalho Prático – Fundamentos de Computação Gráfica

SIMULANDO A COBRINHA DO SLITHER.IO

Entrega: até **19/10/2024**, via Moodle

Grupo: Individual ou em grupos de até 3 participantes



Fonte da imagem: Autoral

Introdução

O jogo slither.io, lançado em 2016 por Steve Howse, é uma releitura do clássico *Snake* (1976). Ele se tornou popular por sua jogabilidade casual e multiplayer, onde jogadores controlam uma cobra composta de segmentos, competindo por espaço ao evitar colisões e coletar "comidinhas" para crescer. Neste trabalho, vamos simplificar essa ideia para focar em transformações gráficas e controle de movimentação, explorando conceitos importantes no pipeline gráfico programável com a API OpenGL.

Objetivo

Desenvolver uma aplicação interativa, utilizando a API OpenGL, que simula a movimentação de uma cobrinha em um ambiente 2D. O objetivo é aplicar corretamente as transformações gráficas (model matrix e projection matrix) para movimentação e controle da cobrinha, além de implementar colisões básicas e a coleta de itens. Esse trabalho deve demonstrar domínio do pipeline gráfico programável, com ênfase em buffers de geometria e shaders.

Descrição do Problema

Sua tarefa é implementar uma versão simplificada de *slither.io*, onde o jogador controla uma cobrinha composta de segmentos. O jogador deverá movimentar a cobrinha por meio do mouse ou teclado, rotacionando a cabeça e fazendo com que os segmentos do corpo sigam o movimento. A cobrinha poderá coletar "comidinhas", representadas por quadrados na tela, e crescer conforme se alimenta. As transformações aplicadas aos segmentos e itens da cena devem ser gerenciadas pela matriz de projeção ortográfica 2D (projection matrix) e a matriz de transformação dos objetos (model matrix).

Requisitos

1. **API OpenGL 3.3 ou superior:** O programa deve utilizar shaders para o pipeline gráfico, implementando tanto o vertex shader quanto o fragment shader.
2. **Buffers de Geometria:**
 - Corretamente criar e utilizar **VAO**, **VBO(s)** e, opcionalmente, EBO para armazenar e gerenciar os dados dos vértices e polígonos.
3. **Transformações:**

- Implementar corretamente a matriz de projeção ortográfica 2D para mapeamento da cena.
- Aplicar e atualizar as transformações nos objetos da cena (model matrix) para permitir a movimentação fluida da cobrinha e dos itens.

4. Movimentação:

- Implementar o controle da movimentação da cobrinha com input via teclado e/ou mouse.
- Incluir a funcionalidade de rotacionar a cabeça da cobrinha com base no movimento do cursor ou teclas.

5. Colisões e Coleta de Itens:

- Implementar colisões básicas utilizando hitboxes para a detecção de colisão entre a cobrinha e as "comidinhas".
- A cobrinha deve crescer conforme coleta itens (aumento no número de segmentos).

Saída do Programa

O programa deve exibir uma cobrinha composta de segmentos, com movimentação suave e interativa. A tela deve mostrar a cobrinha coletando itens e crescendo à medida que se alimenta, utilizando as transformações corretas. A movimentação da cobrinha deve ser natural, com os segmentos seguindo a cabeça, e as colisões com os itens devem ser detectadas corretamente.

Instruções para Envio

A entrega do trabalho será feita em duas partes:

1. Código-fonte:

Apenas um integrante do grupo deve enviar o link para o repositório do projeto no Moodle até as 23h59min do dia 19/10/2024. O diretório do projeto deve conter:

- O código-fonte completo do trabalho.
- Arquivo **LEIAME.md** com:
 - Nome completo dos integrantes.
 - Instruções para executar o programa.
 - Qualquer explicação adicional necessária sobre o funcionamento do código.

2. Apresentação em Vídeo:

Além do código, o grupo deve enviar uma gravação de tela com narração, apresentando o projeto e explicando os principais aspectos do desenvolvimento. Pontos a serem abordados no vídeo:

- Apresentação do grupo (nomes dos integrantes).
- Estrutura geral do código (classes e/ou funções).

- Estrutura dos buffers e shaders (como os vértices e seus atributos foram especificados e enviados para os shaders).
- Controle do personagem (teclado e/ou mouse).
- Verificação de colisões.
- Movimentação de outros objetos do cenário (se houver).
- Demonstração do programa em funcionamento (sugestão: mostre o programa logo após a apresentação do grupo, mas pode retornar ao programa em outros momentos, conforme necessário).

Bom trabalho! 😊

Lembre-se: problemas grandes podem ser resolvidos quebrando-os em problemas menores!