

Universidade de São Paulo  
Instituto de Matemática e Estatística  
IME

**Relatório do projeto final de MAC0420**  
**”Garden Gaze”**  
**(Introdução à Computação Gráfica)**  
Clara Yuki Sano  
Júlia Melo Teixeira dos Santos  
Luísa Menezes da Costa

Junho  
2024

# 1 Breve descrição do projeto

O projeto consiste em um simulador de jardim com pontos de vista (POVs) de animais diferentes se movimentando por meios diferentes (terrestre, aquático e aéreo).

## 2 Introdução

Garden Gaze é um projeto desenvolvido para a disciplina MAC0420 (Introdução à Computação Gráfica) utilizando WebGL2. O objetivo do projeto é implementar diversas técnicas aprendidas durante a disciplina, tais como texturas, iluminação, sombras, etc.

## 3 O que o sistema faz?

Garden Gaze permite ao usuário passear por um jardim povoado com diversos animais e elementos cenográficos. Ao carregar o projeto, o usuário controla uma câmera que olha o jardim inteiro. É possível sobrevoar o jardim a partir dessa câmera em 1ª pessoa, bem como alterar o ângulo da câmera em todos os eixos.

O usuário também é capaz de trocar a posição da câmera de forma a enxergar o POV em 3ª pessoa de algum dos animais. Ao mudar para o POV de um animal, ele para seu movimento automático para que o usuário possa controlar seu movimento.

### 3.1 Mini manual do usuário

#### 3.1.1 Botões e slider da interface

- **botão Executar/Pausar:** ao clicar em "Executar", o simulador é iniciado e os animais começam a se movimentar. Se a câmera possuir velocidade, ela também se movimenta. Ao clicar em Pausar, o simulador pausa e os animais e a câmera não se movem. Note que ainda é possível rotacionar a câmera e os animais pausados, se você estiver nos respectivos POVs.
- **slider Dia/Noite:** o slider começa na posição Dia, que simula o nascer do Sol. O usuário pode alterar os valores do slide para simular a movimentação natural do Sol e alterar as sombras dos objetos no chão. Além disso, mudar o slider também escurece a cena, para simular um dia claro e uma noite escura.
- **botão Jardim:** ao clicar no botão "Jardim", a câmera volta para sua posição inicial olhando o jardim. O usuário pode controlar a câmera a partir das teclas abaixo.

- **botão Abelha:** ao clicar no botão "Abelha", o usuário vai para o POV da abelha e acompanha ela a partir de uma câmera em 3ª pessoa. O movimento automático da abelha para e o usuário pode controla-la a partir das teclas abaixo.
- **botão Caracol:** ao clicar no botão "Caracol", o usuário vai para o POV do caracol e acompanha ele a partir de uma câmera em 3ª pessoa. O movimento automático do caracol para e o usuário pode controlá-lo a partir das teclas abaixo.
- **botão Peixe:** ao clicar no botão "Peixe", o usuário vai para o POV do peixe e acompanha ele a partir de uma câmera em 3ª pessoa. O movimento automático do peixe para e o usuário pode controla-lo a partir das teclas abaixo.

### 3.1.2 Teclas de controle de movimento

Os controles da câmera geral e da abelha funcionam da seguinte forma:

- **tecla J:** diminui velocidade de translação, fazendo com que a câmera ande para trás;
- **tecla K:** zera velocidade de translação, fazendo com que a câmera pare;
- **tecla L:** aumenta velocidade de translação, fazendo com que a câmera ande para frente;
- **teclas W e X:** W faz a câmera subir (apontar para cima) e X faz a câmera descer (apontar para baixo);
- **teclas A e D:** A faz a câmera virar para à esquerda e D para à direita;
- **teclas Z e C:** Z faz a câmera girar no sentido anti-horário e C no sentido horário.

Os controles do peixe e do caracol funcionam da seguinte forma:

- **tecla J:** diminui velocidade de translação, mas não move o animal;
- **tecla K:** zera velocidade de translação, fazendo com que a câmera pare;
- **tecla L:** aumenta velocidade de translação, mas não move o animal;
- **teclas W e S:** se o animal tiver alguma velocidade (setada com os botões JKL), W faz ele andar para frente e S faz ele andar de ré;
- **teclas A e D:** A faz a câmera virar para à esquerda e D para à direita;

### 3.2 Como rodar o código?

Dentro de seu VSCode, instale a extensão LiveServer. Após instalar, vá para o arquivo .html do projeto. No canto inferior direito, clique na opção "Go Live". O projeto será automaticamente aberto no local host do seu computador.

**ATENÇÃO:** Se você rodar o .html diretamente do diretório, as texturas não serão carregadas, por isso você deve obrigatoriamente rodar com o auxílio do LiveServer.

## 4 Desenvolvimento do sistema

### 4.1 Modelos

#### 4.1.1 Animais

- **Caracol:** O caracol foi modelado como dois cilindros conectados em 90° graus para o corpo e uma esfera para a concha. Além disso, a esfera possui uma textura que simula uma concha e os cilindros tem textura que simula uma pele de caracol. Na cena geral, o caracol faz trajetória elíptica sobre a terra enquanto sua cabeça rotaciona. Quando o usuário mudar para o POV do caracol, ele poderá controlar seu movimento pelo chão utilizando as teclas mapeadas.
- **Peixe:** O peixe foi modelado como uma esfera alongada para o corpo e uma pirâmide ligada a uma das pontas dessa esfera para a cauda. Além disso, a esfera e a pirâmide possuem uma textura que simulam escamas de peixe. Na cena geral, o peixe irá se movimentar em formato de infinito dentro da água enquanto sua cauda rotaciona. Quando o usuário mudar para o POV do peixe, ele poderá controlar seu movimento pelo água utilizando as teclas mapeadas.
- **Abelha:** A abelha foi modelada como uma esfera para o corpo, duas pirâmides estreitas ligadas à lateral dessa esfera para as asas e dois cilindros ligados à ponta da esfera para as antenas. Na cena geral, a abelha faz trajetória circular no ar enquanto suas asas rotacionam (para cima e para baixo). Quando o usuário mudar para o POV do caracol, ele poderá controlar seu movimento livre pelo ar utilizando as teclas mapeadas.

#### 4.1.2 Elementos cenográficos

- **Pedras:** As pedras foram modeladas como esferas escaladas de formas específicas, permitindo uma variedade maior entre seus formatos. Além disso, houve a utilização de uma textura própria de pedra em sua criação.
- **Árvore:** As árvores foram modeladas a partir de um cilindro alongado simulando o tronco, e uma esfera simulando a copa. Não houve a utilização de texturas nas árvores, apenas a atribuição de cores (marrom e verde).

- **Horta:** A horta é composta por dois elementos essenciais: a terra, modelada por um cubo escalado de forma a parecer um retângulo sobre o chão do jardim, e os brotos, modelados como pequenas esferas verdes. Houve a utilização de textura apenas para a terra de cultivo.

## 4.2 Interatividade

O usuário pode interagir com a cena através de mudanças na câmera geral (que olha todo o jardim) e mudanças de movimentos dos animais. Essas mudanças de câmera são feitas pelas teclas especificadas na seção 3.1.

Na interface, há um slider de Dia/Noite que permite controlar a iluminação da cena e testar diversos tipos de sombra. Além disso, há botões para escolher qual animal (POV) o usuário gostaria de controlar, incluindo um botão para voltar a ver o jardim pela câmera geral.

## 4.3 Algoritmos e métodos implementados

- Iluminação de Phong;
- Implementação de texturas para concha e pele do caracol, escamas do peixe, asas e corpo da abelha, grama, pedras, terra, folhas e tronco das árvores;
- Implementação de sombras projetadas que mudam de posição e tamanho de acordo com a posição do Sol;

## 4.4 Alterações que o projeto sofreu a partir da proposta

Não foram implementadas texturas da água, do rastro de caracol e nem da visão em hexágonos da abelha. Além disso, não foi utilizada a biblioteca `babylon.js`

## 5 Atividades realizadas por cada integrante

- **Julia:** classes Shader (com iluminação e sombra), Cilindro, Caracol e Camera.
- **Yuki:** classes Esfera e Abelha; texturas e implementação dos efeitos do slider Dia/Noite.
- **Luísa:** classes Cubo, Pirâmide e Peixe.
- **Todas:** implementação da cena geral, implementação do POV dos animais, escrita do relatório.

## 6 Bugs e problemas conhecidos

- Ao rodar o código diretamente pelo .html, as texturas não são carregadas, pois são locais. Para evitar isso, rode o código com a extensão LiveServer do VSCode.
- Não foram implementados limites para a posição dos animais, então eles podem atravessar o chão e andar para uma posição em que não há mais chão.
- A câmera tem suas rotações ao redor do eixo Y e Z confundidas quando está rotacionada em 90 graus, ou seja, olhando para o norte, apesar do mapeamento de seu sistema de coordenadas ter seguido as instruções discutidas em aula.

## 7 Considerações finais

O projeto permitiu com que colocássemos em prática os aprendizados de implementação de elementos visuais gráficos "do zero", sem auxílio de biblioteca. Construímos shaders mais complexos unindo os diversos algoritmos vistos em aula em uma só aplicação, buscando alcançar o maior nível de realismo possível dentro das limitações e escopo da disciplina. Como foco do projeto, tivemos a manipulação de câmeras em diversas configurações (POV, cena geral, etc.) controladas pelo usuário.