

PROFESSOR: Adriano Felix Valente	
CURSO: Ciência da Computação	
DISCIPLINA: Algoritmos e Lógica de Programação	
TURMA: 1º Matutino	DATA: 22/04/2025
ALUNO(A): Felipe Mathey, Guilherme Monteiro, Laís Navarro, Raphael Santos e Pedro Costa.	

## 1. Introdução

Neste documento, mostramos como aplicamos estruturas fundamentais da programação — decisão, repetição, vetores e matrizes — no desenvolvimento de um projeto feito na Unity. A ideia é demonstrar como essas estruturas foram usadas de maneira clara, funcional e sem atalhos, conforme os exemplos discutidos em aula.

## 2. Estruturas de Decisão

Exemplo: Controle da HUD (HUDController.cs)

```
if (popularity >= 70)
    happinessEmojiText.text = "😊";
else if (popularity >= 40)
    happinessEmojiText.text = "😐";
else
    happinessEmojiText.text = "😞";
```

**Resultado no Projeto:** A HUD exibe um emoji que representa o nível de satisfação da população com o prefeito (Jogador), baseado na variável popularity. Essa estrutura condicional permite representar visualmente o impacto das decisões do jogador na felicidade da cidade. Se o jogador tomar decisões corretas a felicidade da população aumentará, escolhendo a decisão incorreta a população ficará cada vez mais irritada.

## 3. Estruturas de Repetição

Exemplo: Atualização Contínua da HUD

```
void Update()
{
    UpdateHUD();
}
```

**Resultado no Projeto:** A função Update() roda a cada frame. Nela, chamamos UpdateHUD() para manter os dados visuais sempre sincronizados com os valores internos da simulação. Ele garante que os dados da HUD sejam atualizados em tempo real, refletindo dinamicamente as mudanças nos indicadores da cidade.

## 4. Vetores

Exemplo: Direção de Movimento da Câmera (CityCameraController.cs)

```
Vector3 direction = new Vector3();
if (Input.GetKey(KeyCode.W)) direction += transform.forward;
if (Input.GetKey(KeyCode.S)) direction -= transform.forward;
if (Input.GetKey(KeyCode.D)) direction += transform.right;
if (Input.GetKey(KeyCode.A)) direction -= transform.right;
```

**Resultado no Projeto:** A direção da câmera é determinada com base em vetores tridimensionais, possibilitando movimentação fluida e controlada em um ambiente 3D. A manipulação de vetores é essencial para a navegação precisa pelo mapa da cidade. O jogador pode navegar pelo cenário usando as teclas de direção.

## 5. Matrizes

No contexto dos códigos apresentados, não há implementação direta de matrizes (como arrays bidimensionais). A lógica do jogo não exigiu esse tipo de estrutura até agora.

## 6. Conclusão

Foram aplicadas estruturas de decisão, repetição e vetores que são essenciais na programação de jogos, com exemplos reais extraídos da HUD e do controle de câmera. Os exemplos práticos apresentados mostram como esses conceitos ganham vida no desenvolvimento de uma simulação interativa na Unity.