Disciplina: Algoritmos e Lógica de Programação

Integrantes: Enzo Henrique, Enzo Sakita, Katiê Prado

Curso: Ciência da computação

**Estruturas do jogo “Beefed Up!”**

**1. Estrutura de Decisão (if)**

Usada para verificar o estado da vida do jogador:

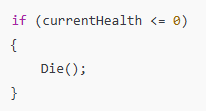
csharp

if (currentHealth <= 0)

{

Die();

}



* **Significado lógico:** Se a vida atual for menor ou igual a 0, o personagem morre.
* Esse tipo de estrutura é essencial em jogos para controlar condições críticas como "Game Over".

**✅ 3. Estrutura de Repetição (comentada no Update)**

O trecho abaixo está comentado, mas pode ser usado como exemplo didático:

csharp

// void Update()

// {

// if (Input.GetKeyDown(KeyCode.Space))

// {

// TakeDamage(10f);

// }

// }

Texto

O conteúdo gerado por IA pode estar incorreto.

* **Descrição:** Um loop implícito, pois o método Update() é chamado a cada frame pelo Unity.
* **Aplicação:** Detecta a tecla **espaço** sendo pressionada para testar dano repetidamente durante a execução do jogo.

✅ **1. Estrutura de Decisão (if)**

No método AddCoins():

if (totalCoins >= limiteParaTrocarCena)

{

TrocarDeCena();

}

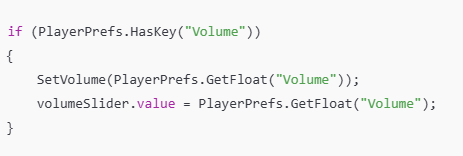
Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

* **Lógica aplicada:** Verifica se o total de moedas atingiu o limite para trocar de cena.
* **Tipo:** **Decisão condicional simples**.

**Estrutura de Decisão (if)**

No método Awake()



* **Explicação:** Verifica se existe um valor salvo anteriormente para o volume do jogo.
* **Finalidade:** Recuperar a configuração de som do jogador ao reiniciar o jogo.
* **Tipo:** **Decisão com chave de persistência (salvamento local)**.

**Após jogar todos os códigos para uma análise**

**Parte 2: Algoritmos – Onde há lógica computacional relevante**

**🔁 1. Sistema de diálogo com digitação letra por letra (InteractwithMan.cs)**

r

IEnumerator TypeSentence(string sentence)

{

foreach (char letter in sentence)

{

dialogueText.text += letter;

yield return new WaitForSeconds(typingSpeed);

}

}

* **Algoritmo de iteração por caractere** com delays — simula uma máquina de escrever.
* Baseado em **corrotinas**, que são uma forma de "pausar" a execução sem travar o jogo.

**🧠 2. Sistema de pontuação e verificação de vitória (wifis.cs)**

csharp

public void AddPoint()

{

score++;

if (score >= maxScore)

{

SceneManager.LoadScene("Beefed Up! Cena Final");

}

}

* Algoritmo simples de **contador** + **verificação de condição final**.
* Se atingir o número máximo de pontos (13), **troca de cena**.

**🚶 3. Controle de movimentação baseado na câmera**

movimento = myCamera.TransformDirection(movimento);

* Converte o movimento do jogador com base no eixo da câmera.
* Isso exige compreensão de **transformações vetoriais** e lógica espacial 3D.

**🧠 4. Interação via interface**

if (hitInfo.collider.TryGetComponent(out IInteractable interactableObj))

{

interactableObj.Interact();

}

* Algoritmo de **identificação e execução de comportamento baseado em interface**.
* Usa **polimorfismo** para permitir que vários objetos respondam a Interact() de forma diferente.