APLICAÇÃO COM DE ALGORITIMOS QUE UTILIZEM ESTRUTURAS DE FUNÇÕES E PROCEDIMENTOS

Movimento, puzzles, reputação do personagem.

MOVIMENTAÇÃO DO PERSONAGEM:

Movimentos nos botões A e D e botão espaço para pular.

SCRIPT:

using Unity.VisualScripting;

using UnityEngine;

public class Movimento : MonoBehaviour

{

private float lastStepTime = 0;

public float delayBetweenSteps = 0.5f;

public CharacterController character;

private Animator animator;

private Vector2 input;

private float velocidade = 4f;

private float forcaY;

private bool podePular = true;

public bool estaNoChao;

[SerializeField] private LayerMask colisaoLayer;

[SerializeField] public Transform peDoPersonagem;

[SerializeField] private AudioClip passos;

private AudioSource audioSource;

private float gravidade = -20f;

private float forcaPulo = 10f;

void Start()

{

character = GetComponent<CharacterController>();

animator = GetComponent<Animator>();

}

void Update()

{

// Movimento lateral no eixo X (2.5D)

input.Set(Input.GetAxisRaw("Horizontal"), 0);

Vector3 move = new Vector3(input.x \* velocidade, 0, 0);

// Verifica se est� no ch�o

estaNoChao = Physics.CheckSphere(peDoPersonagem.position, 0.2f, colisaoLayer);

animator.SetBool("estatNoChao", estaNoChao);

if (estaNoChao && forcaY < 0)

{

forcaY = -2f;

podePular = true;

}

// Pulo

if (Input.GetKeyDown(KeyCode.Space) && estaNoChao && podePular)

{

forcaY = forcaPulo;

animator.SetTrigger("pular");

podePular = false;

}

// Gravidade

forcaY += gravidade \* Time.deltaTime;

move.y = forcaY;

// Move personagem

character.Move(move \* Time.deltaTime);

// Espelha o personagem na dire��o correta (lado)

if (input.x > 0)

{

transform.rotation = Quaternion.Euler(0, 90, 0); // Direita

}

else if (input.x < 0)

{

transform.rotation = Quaternion.Euler(0, -90, 0); // Esquerda

}

// Anima��o de corrida

if (Mathf.Abs(input.x) > 0.1f)

{

animator.SetBool("correndo", true);

}

else

{

animator.SetBool("correndo", false);

}

if ((Mathf.Abs(input.x) > 0.1f) && (estaNoChao) && (Time.time - lastStepTime >= delayBetweenSteps)) //se estiver andando, estiver no ch�o e se faz algum tempo desde que o som de passos tocou

{

AudioSource.PlayClipAtPoint(passos, transform.position, 1f); // toca som de passos

lastStepTime = Time.time; // atualiza tempo em que som passos tocou

}

}

}

PUZZLE DE CONSERTAR HIDRANTE:

HIDRANTE ATIRA BOLAS DE AGUA NO PERSONAGEM, ENTÃO O PERSONAGEM TEM QUE CHEGAR E FAZER UMA PUZZLE ONDE O CONSERTA.

SCRIPT:

using UnityEngine;

using static UnityEditor.ShaderData;

public class TriggerCanos1 : MonoBehaviour

{

[SerializeField] private AudioClip puzzleAtivo; //audio de ativa��o do puzzle

public GameObject PuzzleCanos1; // canvas do puzzle

public GameObject Canvas1; // texto que pede input do jogador

public GameObject Win1; // canvas de quando jogador ganha

private bool PlayerTrigger = false;

private AudioSource audioSource;

void Start()

{

PuzzleCanos1.gameObject.SetActive(false); //desativa o canvas do puzzle

Canvas1.gameObject.SetActive(false); // desativa o texto de input

}

void Update()

{

if (Input.GetKeyDown(KeyCode.E) && PlayerTrigger) //se o player estiver dentro do trigger e apertar "E"...

{

AudioSource.PlayClipAtPoint(puzzleAtivo, transform.position, 1f); // toca som da ativa��o de puzzles

Canvas1.SetActive(false); // desativa canvas1

PuzzleCanos1.SetActive(true); //ativa canvas do puzzle

Win1.SetActive(false);// desativa canvas de vitoria

Time.timeScale = 0f; // pausa o tempo

}

}

private void OnTriggerEnter(Collider other)

{

if (other.gameObject.tag == "Player") //se player entrar no trigger

{

Canvas1.SetActive(true); // ativa o texto de input

PlayerTrigger = true; // faz com que PlayerTrigger seja true

}

}

private void OnTriggerExit(Collider other)

{

if (other.gameObject.tag == "Player") //se player sair do trigger

{

Canvas1.SetActive(false); //desativa texto de input

PlayerTrigger = false; // faz com que Playertrigger seja false

}

}

}

PUZZLE DA PLACA SOLAR

PLACAS SOLARES COM PROBLEMAS, LEO PRECISA CONECTAR OS FIOS PARA FAZER ELA FUNCIONAR NOVAMENTE.

SCRIPT:

using UnityEngine;

using UnityEngine.UI;

public class PainelSolarPuzzle : MonoBehaviour

{

public Button[] fios; // 3 fios

public Button[] terminais; // 3 terminais

public Button botaoConcluir;

public Button botaoSair; // Bot�o para sair do puzzle

public GameObject trigger;

public GameObject maisReputacao;

private float aumentoReputacao = 10f;

public Slider barraReputacao;

private int fioSelecionado = -1;

private bool[] conectado = new bool[3];

void Start()

{

for (int i = 0; i < fios.Length; i++)

{

int index = i;

fios[i].onClick.AddListener(() => SelecionarFio(index));

}

for (int i = 0; i < terminais.Length; i++)

{

int index = i;

terminais[i].onClick.AddListener(() => Conectar(index));

}

botaoConcluir.onClick.AddListener(FinalizarPuzzle);

botaoSair.onClick.AddListener(SairDoPuzzle);

}

void SelecionarFio(int index)

{

fioSelecionado = index;

Debug.Log("Fio selecionado: " + index);

}

void Conectar(int terminalIndex)

{

if (fioSelecionado == -1)

{

Debug.Log("Selecione um fio primeiro!");

return;

}

if (fioSelecionado == terminalIndex)

{

Debug.Log("Conectado corretamente!");

fios[fioSelecionado].interactable = false;

terminais[terminalIndex].interactable = false;

conectado[terminalIndex] = true;

}

else

{

Debug.Log("Conex�o errada!");

}

fioSelecionado = -1;

}

void FinalizarPuzzle()

{

if (conectado[0] && conectado[1] && conectado[2])

{

trigger.SetActive(false);

Debug.Log("Puzzle resolvido!");

gameObject.SetActive(false);

maisReputacao.SetActive(true);

Time.timeScale = 1f;

barraReputacao.value = Mathf.Clamp(barraReputacao.value + aumentoReputacao, 0, barraReputacao.maxValue);

}

else

{

Debug.Log("Ainda falta ligar algum fio!");

}

}

void SairDoPuzzle()

{

Debug.Log("Saindo do puzzle...");

// Aqui voc� pode decidir o que fazer quando o jogador sair do puzzle:

// 1. Desativa o painel do puzzle.

// 2. Retorna para o menu principal ou cena anterior.

gameObject.SetActive(false); // Desativa o painel do puzzle

Time.timeScale = 1f; // Se o jogo foi pausado, retoma o tempo normal

}

}

BARRA DE REPUTAÇÃO

O JOGADOR CONTA COM UMA BARRA DE REPUTAÇÃO QUE AUMENTA A CADA PUZZLE FEITA.

SCRIPT:

using UnityEngine;

using System.Collections;

public class MaisReputacãoTxt : MonoBehaviour

{

void Update()

{

if (gameObject.activeSelf) //se o objeto estiver ativo

{

StartCoroutine(wait()); //começa coroutine

}

}

public IEnumerator wait()

{

yield return new WaitForSeconds(2f); // espera 2 segundos

gameObject.SetActive(false); // se desativa

}

}