

FUNDAÇÃO VALEPARAIBANA DE ENSINO
COLÉGIOS UNIVAP – UNIDADE CENTRO

CURSO TÉCNICO EM INFORMÁTICA

LEONARDO MARTINELLI DE OLIVEIRA LIMA

LISTA DE EXERCÍCIOS 2º BIMESTRE

DESENVOLVIMENTO DE JOGOS

Lista apresentada ao Curso Técnico de informática
como composição de nota.
Prof. Me. Hélio Lourenço Esperidião Ferreira

SÃO JOSÉ DOS CAMPOS
2023

1)Quais as formas de colisão que existem e podem ser representadas no unity?

De acordo com os dados da Unity os tipos de colisores são:

Composite Collider, Box Collider, Circle Collider, Polygon Collider, Capsule Collider , Edge Collider.

2)Crie um pequeno jogo contendo partes do chão coberta com gelo. Aumente a velocidade do jogador quando personagem passar pelo gelo. Ao sair do gelo retorne a velocidade normal.

```
using UnityEngine;
```

```
public class PlayerController : MonoBehaviour {
    public float playerSpeed = 10f;
    public float iceSpeedMultiplier = 1.5f; // valor maior que 1 para aumentar a velocidade no gelo
    private float originalSpeed;

    private void Start() {
        originalSpeed = playerSpeed;
    }

    private void OnTriggerEnter(Collider collider) {
        if (collider.gameObject.tag == "Ice") {
            playerSpeed = originalSpeed * iceSpeedMultiplier;
        }
    }

    private void OnTriggerExit(Collider collider) {
        if (collider.gameObject.tag == "Ice") {
            playerSpeed = originalSpeed;
        }
    }

    private void FixedUpdate() {
        float moveHorizontal = Input.GetAxis("Horizontal");
        float moveVertical = Input.GetAxis("Vertical");

        Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);

        GetComponent<Rigidbody>().AddForce(movement * playerSpeed * Time.deltaTime);
    }
}
```

3)Crie um pequeno jogo contendo lama no chão. Diminua a velocidade do jogador quando o mesmo permanece na lama.

```
using UnityEngine;
```

```
public class PlayerController : MonoBehaviour {  
    public float playerSpeed = 10f;  
    public float mudSpeedMultiplier = 0.5f; // valor menor que 1 para diminuir a velocidade na  
    lama  
    private float originalSpeed;
```

```
    private void Start() {  
        originalSpeed = playerSpeed;  
    }
```

```
    private void OnTriggerEnter(Collider collider) {  
        if (collider.gameObject.tag == "Lama") {  
            playerSpeed = originalSpeed * mudSpeedMultiplier;  
        }  
    }
```

```
    private void OnTriggerExit(Collider collider) {  
        if (collider.gameObject.tag == "Lama") {  
            playerSpeed = originalSpeed;  
        }  
    }
```

```
    private void FixedUpdate() {  
        float moveHorizontal = Input.GetAxis("Horizontal");  
        float moveVertical = Input.GetAxis("Vertical");
```

```
        Vector3 movement = new Vector3(moveHorizontal, 0.0f, moveVertical);
```

```
        GetComponent<Rigidbody>().AddForce(movement * playerSpeed * Time.deltaTime);  
    }  
}
```

4)Faça um pulo triplo.

```
public class PlayerController : MonoBehaviour  
{  
    public float jumpForce = 10f;  
    public int maxJumps = 3;  
    private int jumpsLeft;
```

```

private void Start()
{
    jumpsLeft = maxJumps;
}

private void Update()
{
    if (jumpsLeft > 0 && Input.GetKeyDown(KeyCode.Space))
    {
        GetComponent<Rigidbody2D>().velocity = Vector2.up * jumpForce;
        jumpsLeft--;
    }
}

private void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Ground"))
    {
        jumpsLeft = maxJumps;
    }
}

```

5) Faça que o personagem ao tocar em uma banana permita pulo duplo e ao tocar em uma maçã permita pulo triplo.

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float jumpForce = 10f;
    public int maxJumps = 1;
    private int jumpsLeft = 0;

    private void Update()
    {
        // Verificar se o jogador ainda tem pulos restantes
        if (jumpsLeft > 0 && Input.GetKeyDown(KeyCode.Space))
        {
            GetComponent<Rigidbody2D>().velocity = Vector2.up * jumpForce;
            jumpsLeft--;
        }
    }

    private void OnTriggerEnter2D(Collider2D other)

```

```

{
if (other.CompareTag("Banana"))
{
maxJumps = 2; // Permitir pulo duplo
Destroy(other.gameObject);
}
else if (other.CompareTag("Apple"))
{
maxJumps = 3; // Permitir pulo triplo
Destroy(other.gameObject);
}
}

private void OnCollisionEnter2D(Collision2D other)
{
// Resetar os pulos restantes ao tocar no chão
if (other.gameObject.CompareTag("Ground"))
{
jumpsLeft = maxJumps - 1;
}
}
}

```

6)Faça com que o jogador possa coletar maçãs verdes e vermelhas. As vermelhas aumentam a velocidade de movimento. A verdes diminuem.

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
public float moveSpeed = 5f;
public float speedBoostDuration = 5f;
public float speedBoostMultiplier = 2f;
public float speedDecreaseDuration = 5f;
public float speedDecreaseMultiplier = 0.5f;
private float originalMoveSpeed;
private bool isSpeedBoosted = false;
private bool isSpeedDecreased = false;

private void Start()
{
originalMoveSpeed = moveSpeed;
}

private void Update()

```

```

{
// Movimentação horizontal
float horizontalInput = Input.GetAxis("Horizontal");
Vector3 movement = new Vector3(horizontalInput, 0f, 0f);
transform.position += movement * Time.deltaTime * moveSpeed;
}

private void OnTriggerEnter(Collider other)
{
if (other.gameObject.CompareTag("AppleGreen"))
{
moveSpeed *= speedDecreaseMultiplier;
isSpeedDecreased = true;
Destroy(other.gameObject);
}
else if (other.gameObject.CompareTag("AppleRed"))
{
moveSpeed *= speedBoostMultiplier;
isSpeedBoosted = true;
Destroy(other.gameObject);
}
}

private void Update()
{
// Verificar se o jogador ainda está afetado pelo aumento ou diminuição da velocidade
if (isSpeedBoosted)
{
speedBoostDuration -= Time.deltaTime;
if (speedBoostDuration <= 0f)
{
isSpeedBoosted = false;
moveSpeed = originalMoveSpeed;
}
}
if (isSpeedDecreased)
{
speedDecreaseDuration -= Time.deltaTime;
if (speedDecreaseDuration <= 0f)
{
isSpeedDecreased = false;
moveSpeed = originalMoveSpeed;
}
}

// Movimentação horizontal

```

```

float horizontalInput = Input.GetAxis("Horizontal");
Vector3 movement = new Vector3(horizontalInput, 0f, 0f);
transform.position += movement * Time.deltaTime * moveSpeed;
}
}

```

7)Faça um pulo triplo onde a intensidade do pulo diminui do pulo primário para o duplo e do duplo para o triplo.

```
using UnityEngine;
```

```

public class PlayerController : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float jumpForcePrimary = 10f;
    public float jumpForceDouble = 7f;
    public float jumpForceTriple = 4f;
    private int jumpsRemaining = 3;
    private bool isJumping = false;
    private Rigidbody rb;

    private void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    private void Update()
    {
        // Movimentação horizontal
        float horizontalInput = Input.GetAxis("Horizontal");
        Vector3 movement = new Vector3(horizontalInput, 0f, 0f);
        transform.position += movement * Time.deltaTime * moveSpeed;

        // Verificar se o jogador pode pular
        if (Input.GetKeyDown(KeyCode.Space) && jumpsRemaining > 0)
        {
            isJumping = true;

            // Determinar a força do pulo com base na quantidade de pulos restantes
            float jumpForce = jumpForcePrimary;
            if (jumpsRemaining == 2)
            {
                jumpForce = jumpForceDouble;
            }
            else if (jumpsRemaining == 1)

```

```

{
jumpForce = jumpForceTriple;
}

// Aplicar a força do pulo à Rigidbody
rb.AddForce(new Vector3(0f, jumpForce, 0f), ForceMode.Impulse);
jumpsRemaining--;
}
}

private void OnCollisionEnter(Collision collision)
{
if (collision.gameObject.CompareTag("Ground"))
{
isJumping = false;
jumpsRemaining = 3;
}
}
}

```

8)Crie um pequeno jogo onde o personagem caminha por uma superfície. Em um determinado momento o personagem deve ser arremessado para o alto. (Utilize tags para marcar parte do chão, quando o jogador permanecer sobre uma determinada tag ele recebe um vetor de movimento com componente em y).

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
public float moveSpeed = 5f;
public float jumpForce = 10f;
private bool isJumping = false;
private Rigidbody rb;

private void Start()
{
rb = GetComponent<Rigidbody>();
}

private void Update()
{
// Movimentação horizontal
float horizontalInput = Input.GetAxis("Horizontal");
Vector3 movement = new Vector3(horizontalInput, 0f, 0f);
transform.position += movement * Time.deltaTime * moveSpeed;
}
}

```



```

// Verificar se está em cima da tag "Jump"
RaycastHit hit;
if (Physics.Raycast(transform.position, Vector3.down, out hit, 0.5f))
{
    if (hit.collider.CompareTag("Jump"))
    {
        Jump();
    }
}

private void Jump()
{
    if (!isJumping)
    {
        isJumping = true;
        rb.AddForce(new Vector3(0f, jumpForce, 0f), ForceMode.Impulse);
    }
}

private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        isJumping = false;
    }
}

```

9) Crie um pequeno jogo onde o personagem caminha por uma superfície. Em um determinado momento o personagem deve ser arremessado para o alto em um ângulo de 45°.

```
using UnityEngine;
```

```

public class PlayerController : MonoBehaviour
{
    public float speed = 5f;
    public float jumpForce = 10f;
    private bool isJumping = false;

    private void Update()
    {
        // Movimentação horizontal
    }
}

```

```

float horizontalInput = Input.GetAxis("Horizontal");
Vector3 movement = new Vector3(horizontalInput, 0f, 0f);
transform.position += movement * Time.deltaTime * speed;

// Pulo
if (Input.GetKeyDown(KeyCode.Space) && !isJumping)
{
    isJumping = true;
    GetComponent<Rigidbody>().AddForce(new Vector3(1f, 1f, 0f).normalized * jumpForce,
    ForceMode.Impulse);
}
}

private void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        isJumping = false;
    }
}
}

```

10) Faça com que o jogador possa coletar cogumelos que permitam que o mesmo possa executar pulo triplo por 10 vezes.

```

using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float speed = 5f; // Velocidade de movimento
    public float jumpForce = 10f; // Força do pulo
    public float jumpDuration = 1f; // Duração do pulo triplo
    public int maxJumps = 10; // Número máximo de pulos triplos
    public AudioClip jumpSound; // Som do pulo
    public AudioClip mushroomSound; // Som do cogumelo

    private int jumps; // Número de pulos triplos restantes
    private bool canJump; // Pode realizar pulo triplo?
    private float jumpTime; // Tempo decorrido do pulo triplo
    private bool isJumping; // Está realizando pulo triplo?
    private Rigidbody2D rb;
    private AudioSource audioSource;

    private void Start()
    {

```

```

// Inicializa as variáveis
jumps = maxJumps;
canJump = true;
jumpTime = 0f;
isJumping = false;
rb = GetComponent<Rigidbody2D>();
audioSource = GetComponent<AudioSource>();
}

private void Update()
{
    // Movimenta o jogador
    float moveInput = Input.GetAxis("Horizontal");
    rb.velocity = new Vector2(moveInput * speed, rb.velocity.y);

    // Verifica se o jogador pode realizar pulo triplo
    if (canJump && Input.GetButtonDown("Jump"))
    {
        // Executa o pulo triplo
        jumps--;
        if (jumps >= 0)
        {
            rb.velocity = new Vector2(rb.velocity.x, jumpForce);
            jumpTime = 0f;
            isJumping = true;
            audioSource.PlayOneShot(jumpSound);
        }
    }

    // Verifica se está realizando pulo triplo
    if (isJumping)
    {
        jumpTime += Time.deltaTime;

        if (jumpTime >= jumpDuration)
        {
            isJumping = false;
            canJump = false;
        }
    }

    // Verifica se o jogador colidiu com um cogumelo
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.CompareTag("Mushroom"))
        {
            // Executa o som do cogumelo

```

```
audioSource.PlayOneShot(mushroomSound);
```

```
// Aumenta o número de pulos triplos restantes  
jumps += 10;
```

```
// Desativa o cogumelo  
other.gameObject.SetActive(false);  
}  
}  
}  
}
```

11) Faça um jogo que caiam coisas do céu, caso o jogador seja atingido por algo verde sua velocidade aumenta 2x. caso seja atingido por algo azul sua velocidade aumenta 3x. caso seja atingido por uma pedra sua velocidade diminui 4 vezes.
using UnityEngine;

```
public class PlayerController : MonoBehaviour  
{  
    public float moveSpeed = 5f; // Velocidade inicial do jogador  
    private int velocityMultiplier = 1; // Multiplicador de velocidade do jogador  
  
    private void Update()  
    {  
        // Movimentação do jogador  
        float horizontalInput = Input.GetAxis("Horizontal");  
        float verticalInput = Input.GetAxis("Vertical");  
        transform.Translate(new Vector3(horizontalInput, verticalInput, 0f) * moveSpeed *  
velocityMultiplier * Time.deltaTime);  
    }  
  
    private void OnTriggerEnter2D(Collider2D collision)  
    {  
        // Verifica se colidiu com objeto verde  
        if (collision.gameObject.CompareTag("Green"))  
        {  
            velocityMultiplier = 2;  
        }  
        // Verifica se colidiu com objeto azul  
        else if (collision.gameObject.CompareTag("Blue"))  
        {  
            velocityMultiplier = 3;  
        }  
        // Verifica se colidiu com pedra  
        else if (collision.gameObject.CompareTag("Rock"))
```

```

        {
            velocityMultiplier = 0.25f;
        }
    }

    private void OnTriggerExit2D(Collider2D collision)
    {
        // Volta ao multiplicador de velocidade inicial
        velocityMultiplier = 1;
    }
}

using UnityEngine;

public class SpawnerController : MonoBehaviour
{
    public GameObject[] objectsToSpawn; // Objetos a serem gerados
    public float spawnRate = 1f; // Taxa de geração
    public float minX = -5f; // Limite

    public float maxX = 5f; // Limites
    public float minHeight = -5f; // Altura mínima
    public float maxHeight = 5f; // Altura máxima

    private void Start()
    {
        // Inicia a geração de objetos
        InvokeRepeating("SpawnObject", 0f, spawnRate);
    }

    private void SpawnObject()
    {
        // Gera um objeto aleatório na posição aleatória
        int randomIndex = Random.Range(0, objectsToSpawn.Length);
        GameObject objectToSpawn = objectsToSpawn[randomIndex];
        float randomX = Random.Range(minX, maxX);
        float randomY = Random.Range(minHeight, maxHeight);
        Vector3 spawnPosition = new Vector3(randomX, randomY, 0f);
        Instantiate(objectToSpawn, spawnPosition, Quaternion.identity);
    }
}

```

12) Faça o flappy Bird tendo você, uma figura histórica ou o professor como personagem.

```

using UnityEngine;
using System.Collections;

```

```
public class FlappyBird : MonoBehaviour {

    public float jumpForce = 200f;
    public GameObject gameOverText;

    private Rigidbody2D rb2d;
    private bool isDead = false;

    void Start () {
        rb2d = GetComponent<Rigidbody2D>();
    }

    void Update () {
        if (Input.GetButtonDown("Jump") && !isDead) {
            rb2d.velocity = Vector2.zero;
            rb2d.AddForce(new Vector2(0, jumpForce));
        }
    }

    void OnCollisionEnter2D(Collision2D other) {
        if (other.collider.CompareTag("Obstacle")) {
            isDead = true;
            gameOverText.SetActive(true);
        }
    }
}
```