Universidade Católica de Pernambuco

Recife 06 de junho de 2016

Cadeira: Programação de jogos para dispositivos móveis

Alunos: Gabriel Carvalho e Rany Ferraz

Códigos do jogo King of Runners

using UnityEngine;

using System.Collections;

public class TouchPulo : MonoBehaviour

{

PlayerController link;

void Awake()

{

link = GameObject.Find("Personagem").GetComponent<PlayerController>();

}

void Update()

{

if (Input.GetMouseButtonDown(0))

{

if (link.grounded)

{

link.myRigidbody.velocity = new Vector2(link.myRigidbody.velocity.x, link.jumpForce);

link.stoppedJumping = false;

link.jumpSound.Play();

}

if (!link.grounded && link.canDoubleJump)

{

link.myRigidbody.velocity = new Vector2(link.myRigidbody.velocity.x, link.jumpForce);

//jumpTimeCounter = jumpTime;

link.stoppedJumping = false;

link.canDoubleJump = false;

link.jumpSound.Play();

}

}

if ((Input.GetMouseButton(0) && !link.stoppedJumping))

{

if (link.jumpTimeCounter > 0)

{

link.myRigidbody.velocity = new Vector2(link.myRigidbody.velocity.x, link.jumpForce);

link.jumpTimeCounter -= Time.deltaTime;

}

}

if (Input.GetMouseButtonUp(0))

{

link.jumpTimeCounter = 0;

link.stoppedJumping = true;

}

}

}

using UnityEngine;

using System.Collections;

using UnityEngine.UI;

public class ScoreManager : MonoBehaviour {

public Text scoreText;

public float scoreCount;

public float pointsPerSecond;

public bool scoreIncreasing;

// Update is called once per frame

void Update () {

if (scoreIncreasing) {

scoreCount += pointsPerSecond \* Time.deltaTime;

}

scoreText.text = "Score: " + Mathf.Round (scoreCount);

}

public void AddScore(int pointsToAdd)

{

scoreCount += pointsToAdd;

}

}

using UnityEngine;

using System.Collections;

public class prediosGenerator : MonoBehaviour {

public Transform backGroundGenerator;

public GameObject backGround;

public Transform backGroundGenerationPoint;

private float larguraDoBG = 17.5f;

public ObjectPooler theObjectPooler;

void Update () {

if (transform.position.x < backGroundGenerationPoint.position.x) {

transform.position = new Vector3 (transform.position.x + larguraDoBG, backGroundGenerationPoint.position.y, transform.position.z);

GameObject newpredio = theObjectPooler.GetPooledObject();

newpredio.transform.position = transform.position;

newpredio.transform.rotation = transform.rotation;

newpredio.SetActive (true);

}

}

}

using UnityEngine;

using System.Collections;

public class PlayerController : MonoBehaviour

{

public float moveSpeed;

private float moveSpeedStore;

public float speedMultiplier;

public float speedIncreaseMilestone;

private float speedIncreasMilestoneStore;

private float speedMilestoneCount;

private float speedMilestoneCountStore;

public float jumpForce;

public float jumpTime;

public float jumpTimeCounter;

public bool stoppedJumping;

public bool canDoubleJump;

public bool grounded;

public LayerMask whatIsGround;

public Transform groundCheck;

public float groundCheckRadius;

public Rigidbody2D myRigidbody;

public Animator myAnimator;

public GameManager theGameManager;

public AudioSource jumpSound;

public AudioSource deathSound;

public IEnumerator run()

{

yield return new WaitForSeconds(1);

myRigidbody.velocity = new Vector2(moveSpeed, myRigidbody.velocity.y);

}

// Use this for initialization

void Start()

{

myRigidbody = GetComponent<Rigidbody2D>();

myAnimator = GetComponent<Animator>();

jumpTimeCounter = jumpTime;

speedMilestoneCount = speedIncreaseMilestone;

moveSpeedStore = moveSpeed;

speedMilestoneCountStore = speedMilestoneCount;

speedIncreasMilestoneStore = speedIncreaseMilestone;

}

// Update is called once per frame

void Update()

{

grounded = Physics2D.OverlapCircle(groundCheck.position, groundCheckRadius, whatIsGround);

if (transform.position.x > speedMilestoneCount)

{

speedMilestoneCount += speedIncreaseMilestone;

speedIncreaseMilestone = speedIncreaseMilestone \* speedMultiplier;

moveSpeed = moveSpeed \* speedMultiplier;

}

StartCoroutine("run");

if (grounded)

{

jumpTimeCounter = jumpTime;

canDoubleJump = true;

}

myAnimator.SetBool("Grounded", grounded);

}

void OnCollisionEnter2D(Collision2D other)

{

if (other.gameObject.tag == "killbox")

{

theGameManager.RestartGame();

moveSpeed = moveSpeedStore;

speedMilestoneCount = speedMilestoneCountStore;

speedIncreaseMilestone = speedIncreasMilestoneStore;

deathSound.Play();

}

}

}

using UnityEngine;

using System.Collections;

public class PlataformGenerator : MonoBehaviour

{

public GameObject thePlataform;

public Transform generationPoint;

private float plataformWidth;

public ObjectPooler theObjectPool;

private CoinGenerator theCoinGenerator;

public float randomCoinTreshold;

public float randomObstaculo;

public ObjectPooler obstaculo;

void Start()

{

plataformWidth = thePlataform.GetComponent<BoxCollider2D>().size.x;

theCoinGenerator = FindObjectOfType<CoinGenerator>();

}

void Update()

{

if (transform.position.x < generationPoint.position.x)

{

transform.position = new Vector3(transform.position.x +1.15f, transform.position.y, transform.position.z);

GameObject newPlataform = theObjectPool.GetPooledObject();

newPlataform.transform.position = transform.position;

newPlataform.transform.rotation = transform.rotation;

newPlataform.SetActive (true);

if (Random.Range(0f, 100f) < randomCoinTreshold)

{

theCoinGenerator.SpawnCoins(new Vector3(transform.position.x, transform.position.y + 1.5f, transform.position.x));

}

if (Random.Range(0f, 100f) < randomObstaculo)

{

GameObject newObstaculo = obstaculo.GetPooledObject();

newObstaculo.transform.position = new Vector3(transform.position.x + Random.RandomRange(-5f, 5f), -2.371852f, 38f);

newObstaculo.transform.rotation = transform.rotation;

newObstaculo.SetActive (true);

}

transform.position = new Vector3(transform.position.x + (plataformWidth /2), transform.position.y, transform.position.z);

}

}

}

using UnityEngine;

using System.Collections;

public class PlataformDestroyer : MonoBehaviour {

public GameObject plataformDestructionPoint;

void Start () {

plataformDestructionPoint = GameObject.Find ("PlataformDestructionPoint");

}

void Update () {

if (transform.position.x < plataformDestructionPoint.transform.position.x) {

gameObject.SetActive(false);

}

}

}

using UnityEngine;

using System.Collections;

public class pickUpPoints : MonoBehaviour {

public int scoreToGive;

private ScoreManager theScoreManager;

private AudioSource coinSound;

// Use this for initialization

void Start () {

theScoreManager = FindObjectOfType<ScoreManager> ();

coinSound = GameObject.Find ("CoinSound").GetComponent<AudioSource>();

}

// Update is called once per frame

void Update () {

}

void OnTriggerEnter2D(Collider2D other){

if (other.gameObject.name == "Personagem") {

theScoreManager.AddScore (scoreToGive);

gameObject.SetActive(false);

if(coinSound.isPlaying)

{

coinSound.Stop();

coinSound.Play ();

}else{

coinSound.Play();

}

}

}

}

using UnityEngine;

using System.Collections;

public class PauseMenu : MonoBehaviour {

public string mainMenuLevel;

public GameObject pauseMenu;

public AudioSource clickSom;

public Musica musicalink;

void Awake()

{

musicalink = GameObject.Find("Menu Principal - Smoth Labyrinth (1)").GetComponent<Musica>();

}

public void PauseGame (){

clickSom.Play();

Time.timeScale = 0f;

pauseMenu.SetActive(true);

}

public void continuar(){

clickSom.Play();

Time.timeScale = 1f;

pauseMenu.SetActive(false);

}

public void RestartGame(){

clickSom.Play();

pauseMenu.SetActive(false);

Time.timeScale = 1f;

FindObjectOfType<GameManager> ().Reset ();

}

public void QuitToMain(){

clickSom.Play();

musicalink.bula = true;

Time.timeScale = 1f;

Application.LoadLevel (mainMenuLevel);

}

void Update()

{

DontDestroyOnLoad(clickSom);

}

}

using UnityEngine;

using System.Collections;

using System.Collections.Generic;

public class ObjectPooler : MonoBehaviour {

public GameObject pooledObject;

public int pooledAmount;

List<GameObject> pooledObjects;

// Use this for initialization

void Start () {

pooledObjects = new List<GameObject>();

for (int i = 0; i < pooledAmount; i++)

{

GameObject obj = (GameObject)Instantiate (pooledObject);

obj.SetActive (false);

pooledObjects.Add (obj);

}

}

public GameObject GetPooledObject ()

{

for (int i = 0; i < pooledObjects.Count; i++)

{

if (!pooledObjects[i].activeInHierarchy)

{

return pooledObjects [i];

}

}

GameObject obj = (GameObject)Instantiate(pooledObject);

obj.SetActive (false);

pooledObjects.Add (obj);

return obj;

}

}

using UnityEngine;

using System.Collections;

public class Musica : MonoBehaviour

{

public AudioSource trilha;

public bool bula;

private static Musica instance = null;

public static Musica Instance

{

get { return instance; }

}

void Start()

{

bula = true;

}

void Update()

{

if (bula == false)

{

trilha.mute = true;

}

else

{

trilha.mute = false;

}

}

void Awake()

{

trilha = GetComponent<AudioSource>();

if (instance != null && instance != this)

{

if (instance.trilha.clip != trilha.clip)

{

instance.trilha.clip = trilha.clip;

instance.trilha.volume = trilha.volume;

instance.trilha.Play();

}

Destroy(this.gameObject);

return;

}

instance = this;

trilha.Play();

DontDestroyOnLoad(this.gameObject);

}

}

using UnityEngine;

using System.Collections;

public class MenuPrincipal : MonoBehaviour {

public string playGameLevel;

public string playGameLevel2;

public string playGameLevel3;

public string playGameLevel4;

public string playGameLevel5;

public Musica musicalink;

public AudioSource clickSound;

void Awake()

{

musicalink = GameObject.Find("Menu Principal - Smoth Labyrinth (1)").GetComponent<Musica>();

}

public void JogarNivel1()

{

clickSound.Play();

musicalink.bula = false;

Application.LoadLevel (playGameLevel);

}

public void JogarNivel2()

{

clickSound.Play();

musicalink.bula = false;

Application.LoadLevel (playGameLevel5);

}

public void Jogar()

{

clickSound.Play();

Application.LoadLevel (playGameLevel4);

}

public void Instrucoes()

{

clickSound.Play();

Application.LoadLevel (playGameLevel2);

}

public void Creditos()

{

clickSound.Play();

Application.LoadLevel (playGameLevel3);

}

public void Sair()

{

clickSound.Play();

Application.Quit ();

}

void Update()

{

DontDestroyOnLoad(clickSound);

}

}

using UnityEngine;

using System.Collections;

public class GameManager : MonoBehaviour {

public Transform plataformGenerator;

public Vector3 plataformStartPoint;

public PlayerController thePlayer;

public Vector3 playerStartPoint;

public Transform gramaGenerator;

public Vector3 gramaGeneratorOriginPoint;

private ScoreManager theScoreManager;

public DeathMenu theDeathScreen;

public PlataformDestroyer[] destroyerList;

void Start () {

plataformStartPoint = plataformGenerator.position;

playerStartPoint = thePlayer.transform.position;

gramaGeneratorOriginPoint = gramaGenerator.transform.position;

theScoreManager = FindObjectOfType<ScoreManager> ();

}

void Update () {

}

public void RestartGame()

{

theScoreManager.scoreIncreasing = false;

thePlayer.gameObject.SetActive(false);

theDeathScreen.gameObject.SetActive (true);

}

public void Reset(){

theDeathScreen.gameObject.SetActive (false);

destroyerList = FindObjectsOfType<PlataformDestroyer>();

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

{

destroyerList[i].gameObject.SetActive(false);

}

thePlayer.transform.position = playerStartPoint;

plataformGenerator.position = plataformStartPoint;

gramaGenerator.position = gramaGeneratorOriginPoint;

thePlayer.gameObject.SetActive (true);

theScoreManager.scoreCount = 0;

theScoreManager.scoreIncreasing = true;

}

}

using UnityEngine;

using System.Collections;

public class DeathMenu : MonoBehaviour {

public AudioSource clickSoun;

public string mainMenuLevel;

public Musica musicalink;

void Awake()

{

musicalink = GameObject.Find("Menu Principal - Smoth Labyrinth (1)").GetComponent<Musica>();

}

public void RestartGame(){

clickSoun.Play();

FindObjectOfType<GameManager> ().Reset ();

}

public void QuitToMain(){

clickSoun.Play();

musicalink.bula = true;

Application.LoadLevel (mainMenuLevel);

}

}

using UnityEngine;

using System.Collections;

public class CoinGenerator : MonoBehaviour {

public ObjectPooler coinPool;

public float distanceBetweenCoins;

private float distancey = Random.Range(1,8);

private float distancex = Random.Range(5, 10);

public void Update()

{

distancey = Random.Range(1, 7);

distancex = Random.Range(5, 10);

}

public void SpawnCoins (Vector3 startPosition){

GameObject coin1 = coinPool.GetPooledObject();

coin1.transform.position = new Vector3(startPosition.x + distancex, startPosition.y + distancey, 35);

coin1.SetActive (true);

GameObject coin2 = coinPool.GetPooledObject();

coin2.transform.position = new Vector3(startPosition.x - distanceBetweenCoins + distancex, startPosition.y + distancey, 35);

coin2.SetActive (true);

GameObject coin3 = coinPool.GetPooledObject();

coin3.transform.position = new Vector3(startPosition.x + distanceBetweenCoins + distancex, startPosition.y + distancey, 35 );

coin3.SetActive (true);

}

}

using UnityEngine;

using System.Collections;

public class CameraController : MonoBehaviour {

public PlayerController thePlayer;

private Vector3 lastPlayerPosition;

private float distanceToMove;

// Use this for initialization

void Start () {

thePlayer = FindObjectOfType<PlayerController>();

lastPlayerPosition = thePlayer.transform.position;

}

// Update is called once per frame

void Update () {

distanceToMove = thePlayer.transform.position.x - lastPlayerPosition.x;

transform.position = new Vector3(transform.position.x + distanceToMove, transform.position.y, transform.position.z);

lastPlayerPosition = thePlayer.transform.position;

}

}