

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

ENGENHARIA INFORMÁTICA

Tecnologias Multimédia

2022/2023

TP2
UNITY GAME – SNAKE 3D

Aluno:

Fernando – 22065

Docente:

Alexandre Silva

Beatriz Miranda

Marcelo Antunes Fernandes

Paula Rego



**Instituto Politécnico
de Viana do Castelo**

12 de junho de 2023

Índice

- 1. Introdução**
- 2. Gameplay**
- 3. Código**
- 4. Conclusão**
- 5. Links**

1. Introdução

Neste relatório, abordaremos o processo de desenvolvimento do Snake 3D, uma versão em três dimensões do icônico jogo da cobra, utilizando a plataforma Unity. O objetivo deste projeto foi criar uma experiência cativante e divertida para os jogadores, reimaginando um clássico dos arcades. O jogo foi estruturado em quatro cenas distintas: uma cena de menu, que oferece instruções e configurações do jogo, e duas cenas para os diferentes níveis de dificuldade.

Uma atenção especial foi dada aos detalhes audiovisuais para enriquecer a experiência do jogador. Foram incorporados efeitos sonoros personalizados, como o som da cobra ao comer a maçã, para aumentar a imersão. Além disso, modelos 3D foram utilizados para representar a maçã, adicionando uma dimensão visualmente atraente ao jogo.

Ao longo deste relatório, exploraremos as etapas-chave do desenvolvimento, incluindo a criação das cenas, a programação dos controles e a integração dos elementos audiovisuais. Também discutiremos os desafios enfrentados e as soluções encontradas durante o processo de desenvolvimento. No final, esperamos fornecer uma visão abrangente e informativa sobre a criação do Snake 3D em Unity.

2.Gameplay

O Snake 3D é um jogo simples e viciante que oferece uma experiência de gameplay intuitiva. O jogador controla uma cobra em um ambiente tridimensional e o objetivo é guiar a cobra para comer maçãs espalhadas pelo cenário. Ao comer as maçãs, a cobra cresce em comprimento, aumentando a dificuldade ao se movimentar. A jogabilidade é bastante direta, utilizando apenas as teclas A e D para controlar a direção da cobra. O desafio consiste em evitar colisões com as paredes e com o próprio corpo da cobra. O jogo oferece uma experiência simples, porém envolvente, que desafia a habilidade e a estratégia do jogador.

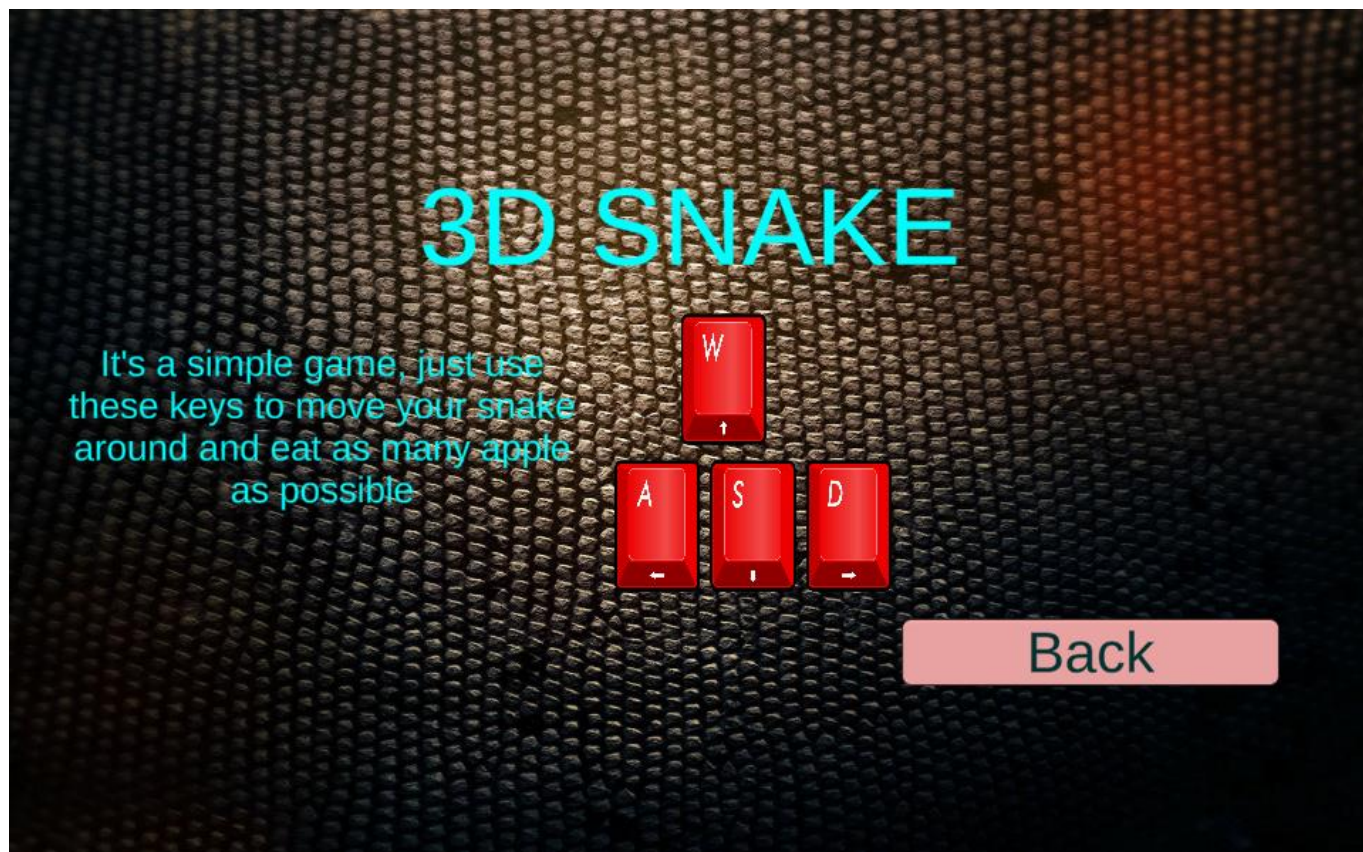


Figura 1 – Menu inicial, instruções

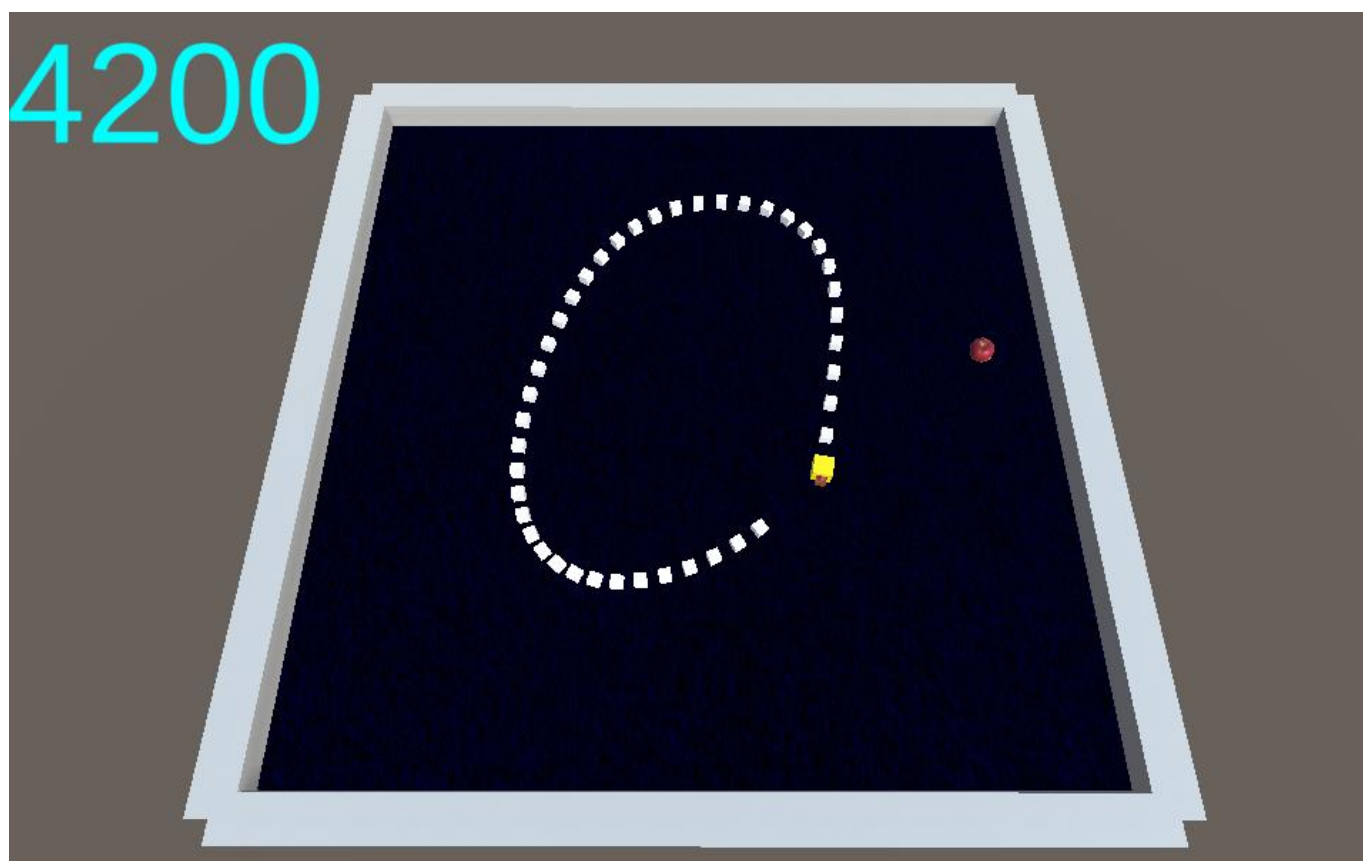


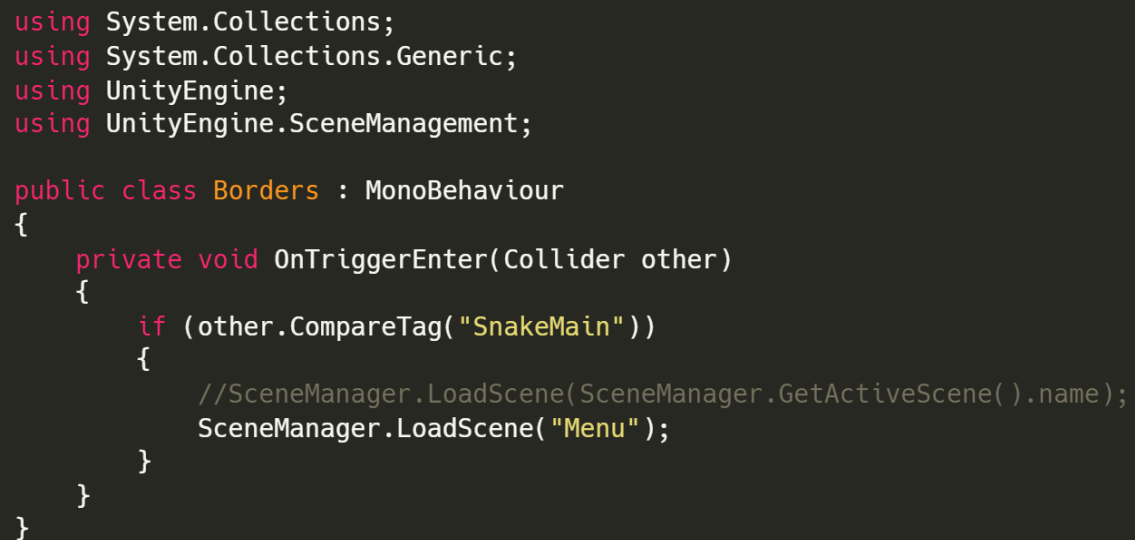
Figura 2 – Exemplo de gameplay

3.Código

Foram utilizados vários scrips:

- Borders
- Food
- FoodGeneration
- mmenu (Main Menu)
- SnakeMovement
- TailMovement

Borders script, se acontecer de bater na parede volta para o Menu, o mesmo que dizer que recomeça o jogo.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class Borders : MonoBehaviour
{
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("SnakeMain"))
        {
            //SceneManager.LoadScene(SceneManager.GetActiveScene().name);
            SceneManager.LoadScene("Menu");
        }
    }
}
```

Figura 3 – Borders script

Food script, trata das colisões com a maçã:

- Adiciona som na colisão, de uma mordida
- Adiciona uma Tail extra à cobra
- Destroi o objeto, já que foi consumido, pra ser gerado denovo no próximo script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Food : MonoBehaviour
{
    public AudioClip biteSound;
    private bool hasBeenConsumed = false;

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("SnakeMain") && !hasBeenConsumed)
        {
            hasBeenConsumed = true;

            // Play the bite sound
            if (biteSound != null)
            {
                AudioSource.PlayClipAtPoint(biteSound, transform.position);
            }

            other.GetComponent<SnakeMovment>().AddTail();
            Destroy(gameObject);
        }
    }
}
```

Figura 4 – Food script

Food Generation script, na falta de maçã, este script cria uma nova

curFood = currentFood, no caso do boolean ser false = 0, é criada uma nova maçã

Vector3 indica os limites onde a maçã pode ser criada, para não ser criada dentro das paredes ou fora do mapa.

```
using System.Collections;
using System.Collections.Generic;
using System.Security.Cryptography;
using UnityEngine;

public class FoodGeneration : MonoBehaviour
{
    #region variable
    public GameObject foodPrefab;
    public GameObject curFood;
    private Vector3 curPos;
    #endregion

    private void AddNewFood()
    {
        curPos = RandomPos();
        curFood = Instantiate(foodPrefab, curPos, Quaternion.identity);
    }

    private Vector3 RandomPos()
    {
        float randomX = Random.Range(-8.8f, 8.8f);
        float randomZ = Random.Range(-8.8f, 8.8f);

        return new Vector3(randomX, 0.3f, randomZ);
    }

    private void Update()
    {
        if(!curFood)
        {
            AddNewFood();
        }
    }
}
```

Figura 5 – Food Generation script

mmenu Script (ou Main Menu), trata apenas das funções dos botões do menu.

Foi utilizado ambos os numeros de building das cenas, como o nome das mesmas, ambas as funções fazem o mesmo.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class mmenu : MonoBehaviour
{
    public void Normal()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1); //numeros é mais facil
        pra ordem das scenes que nomes
    }
    public void Hard()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 2);
    }
    public void Instructions()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 3);
    }
    public void Back()
    {
        SceneManager.LoadScene("Menu");
    }
    public void Quit()
    {
        Application.Quit();
        Debug.Log("Player has quit the game!");
    }
}
```

Figura 6 – mmenu script

Snake Movement script, trata do movimento fluído da cobra e do corpo utilizando apenas A e D, para movimentar a cobra. Ao tratar da função de adicionar um novo corpo à cobra, a função aproveita e incrementa o resultado +100.

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Runtime.CompilerServices;
using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class SnakeMovment : MonoBehaviour
{
    #region variable
    public float speed = 3.0f;
    public float rotationSpeed = 100f;
    public float offset = 0.5f;
    public List<GameObject> tail = new();
    public GameObject tailPrefab;
    public TMP_Text scoreText;
    public int score = 0;
    #endregion

    public void Start()
    {
        tail.Add(gameObject);
    }
    public void Update()
    {
        scoreText.text = Convert.ToString(score);
        transform.Translate(speed * Time.deltaTime * Vector3.forward);

        if (Input.GetKey(KeyCode.A))
        {
            transform.Rotate(Vector3.up, -rotationSpeed * Time.deltaTime);
        }
        if (Input.GetKey(KeyCode.D))
        {
            transform.Rotate(Vector3.up, rotationSpeed * Time.deltaTime);
        }
    }
    public void AddTail()
    {
        score = score + 100;
        Vector3 newTailPos = tail[^1].transform.position;
        newTailPos.z -= offset;
        tail.Add(GameObject.Instantiate(tailPrefab, newTailPos, Quaternion.identity));
    }
}
```

Figura 7 – Snake Movement script

Tail Movement script, contém a função para acabar o jogo no caso da cabeça da cobra bater no corpo, e também dentro do update, as funções do movimento fluído da corpo.

```
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.SceneManagement;

public class TailMovment : MonoBehaviour
{
    #region variable
    public float speed;
    public Vector3 tailTarget;
    public SnakeMovment mainSnake;
    public GameObject tailTargetObj;
    public int id;
    #endregion
    void Start()
    {
        mainSnake = GameObject.FindGameObjectWithTag("SnakeMain").GetComponent<SnakeMovment>();
        speed = mainSnake.speed;
        tailTargetObj = mainSnake.tail[^2];
        id = mainSnake.tail.IndexOf(gameObject);
    }
    void Update()
    {
        tailTarget = tailTargetObj.transform.position;
        transform.LookAt(tailTarget);
        transform.position = Vector3.Lerp(transform.position, tailTarget, Time.deltaTime*speed);
    }
    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("SnakeMain"))
        {
            if(id>2)
            {
                //SceneManager.LoadScene(SceneManager.GetActiveScene().name);
                SceneManager.LoadScene("Menu");
            }
        }
    }
}
```

Figura 8 – Tail Movement script

4. Conclusão

Nesta jornada de desenvolvimento do jogo Snake 3D em Unity, alcançamos com sucesso todos os requisitos estabelecidos para o projeto. Através da implementação cuidadosa das quatro cenas do jogo, incluindo um menu com instruções e dois níveis de dificuldade, conseguimos oferecer aos jogadores uma experiência completa.

Ao utilizar modelos 3D para representar a maçã e efeitos sonoros personalizados para realçar a interação do jogador com o ambiente do jogo, pudemos aprimorar a imersão e proporcionar uma experiência audiovisual envolvente.

Além disso, implementamos a lógica do jogo de forma eficiente, garantindo que a cobra se movimentasse corretamente pelo cenário, crescesse ao comer as maçãs e evitasse colisões com as paredes e o próprio corpo.

Ao finalizar o desenvolvimento, a jogabilidade do Snake 3D ficou intuitiva e desafiadora, permitindo aos jogadores controlar a cobra com facilidade usando as teclas A e D.

Este projeto nos permitiu adquirir conhecimentos valiosos em termos de desenvolvimento de jogos em Unity, desde a criação de cenas e implementação de controles até a integração de elementos audiovisuais. Estamos satisfeitos com os resultados obtidos e esperamos que os jogadores desfrutem dessa versão moderna e imersiva do clássico jogo da cobra.

5.Links

GitHub - <https://github.com/frocha1012/TP2-22065>

Itch.io - <https://frocha2910.itch.io/snake-3d>