Unity WebGL snake game project

Ѓорѓи Галевски, 196041

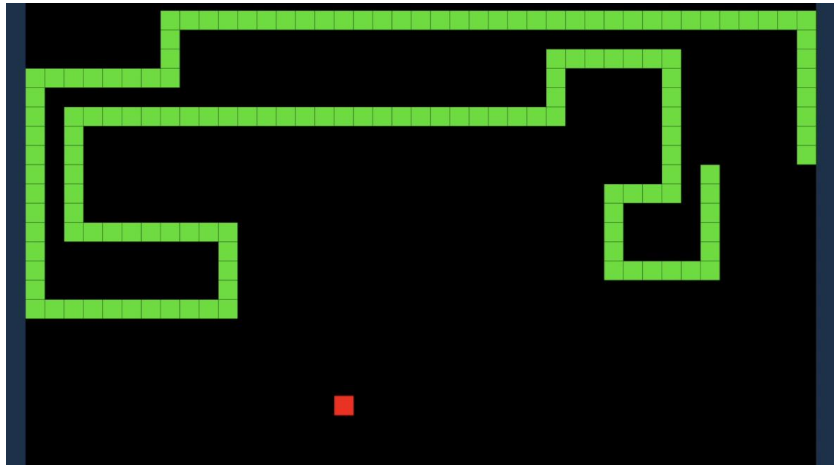
Спасе Костадински, 196116

Introduction

Main goals

Objectives

Snake game and it's history



Project goal

Develep a functional Snake game with classic gameplay

Implement scoring mechanics and high score tracking

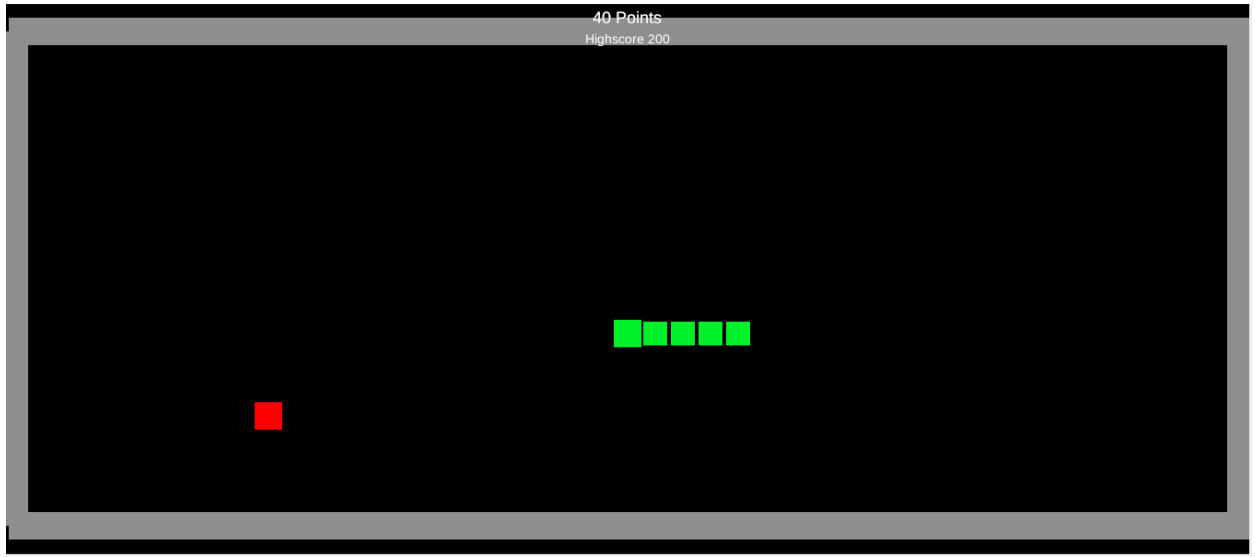
Explore the Unity WebGL platform for web deployment

Game overview

Key features of the Snake game

Design choices made for the game

Importance of scoring and high score tracking



Technical implementation

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

public class ScoreManager : MonoBehaviour
{
    public static ScoreManager instance;
    public Text scoreText;
    public Text highscoreText;

    public int score = 0;
    int highscore = 0;

    private void Awake()
    {
        instance = this;
    }

    void Start()
    {
        highscore = PlayerPrefs.GetInt("highscore", 0);
        UpdateHighscoreText();
    }

    public void AddPoints()
    {
        score += 10;
        scoreText.text = score.ToString() + " Points";
        if (score > highscore)
        {
            highscore = score;
            PlayerPrefs.SetInt("highscore", highscore);
            UpdateHighscoreText();
        }
    }

    public void ResetScore()
    {
        score = 0;
        scoreText.text = score.ToString() + " Points";
    }

   public void UpdateHighscoreText()
    {
        highscoreText.text = "Highscore " + highscore.ToString();
    }
}
```

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

public class Food : MonoBehaviour
{
   public BoxCollider2D gridArea;


    private void Start(){
        RandomizePosition();
    }
    private void RandomizePosition(){
     Bounds bounds = this.gridArea.bounds;

     float x=Random.Range(bounds.min.x,bounds.max.x);
     float y=Random.Range(bounds.min.y,bounds.max.y);

     this.transform.position=new Vector3(Mathf.Round(x),Mathf.Round(y),0.0f);

    }

    private void OnTriggerEnter2D(Collider2D other){
     if(other.tag == "Player")
     {
         RandomizePosition();
     }
    }
}
```

Technical implementation

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


public class Snake : MonoBehaviour
{
    private Vector2 direction = Vector2.right;
    private List<Transform> segments;

    public Transform segmentPrefab;

    private void Start()
    {
        segments = new List<Transform>();
        segments.Add(this.transform);
    }

    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.W) && direction != Vector2.down)
        {
            direction = Vector2.up;
        }
        else if (Input.GetKeyDown(KeyCode.S) && direction != Vector2.up)
        {
            direction = Vector2.down;
        }
        else if (Input.GetKeyDown(KeyCode.A) && direction != Vector2.right)
        {
            direction = Vector2.left;
        }
        else if (Input.GetKeyDown(KeyCode.D) && direction != Vector2.left)
        {
            direction = Vector2.right;
        }
    }
```

```

    private void FixedUpdate()
    {

        for(int i=segments.Count - 1; i > 0; i--){
            segments[i].position=segments[i-1].position;
        }

        this.transform.position = new Vector3(
            (float)(Math.Round(this.transform.position.x) + direction.x),
            (float)(Math.Round(this.transform.position.y) + direction.y),
            0.0f
        );
    }


    private void Grow()
    {
        Transform segment = Instantiate(this.segmentPrefab);
        segment.position = segments[segments.Count - 1].position;

        segments.Add(segment);
        ScoreManager.instance.AddPoints();
    }


```

```
    private void ResetState(){
        for(int i=1; i<segments.Count; i++){
            Destroy(segments[i].gameObject);
        }

        segments.Clear();
        segments.Add(this.transform);

        this.transform.position=Vector3.zero;
        ScoreManager.instance.ResetScore();
        ScoreManager.instance.UpdateHighscoreText();
    }

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.tag == "Food")
        {
            Grow();
        }
        else if(other.tag == "Obstacle"){
            ResetState();
        }
    }
}
```

Challenges and solutions

Challenges faced during deployment

Online Hosting

Version control for collaboration

Future Development

Scoreboard
Leaderboard

Difficulty
levels

Level mode

CONCLUSION