

2D PixelArt Endless Runner

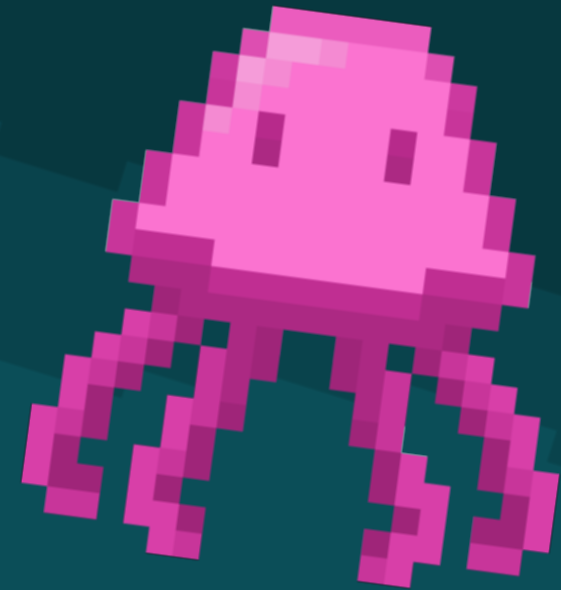
KP9_Ema Stolicna

Goals:

- Basics of pixel art
- Pixel art animation

- Basic 2D Unity game dev
- Basic programming in C#
- Endless Runner prototyping

- Vertical slice- simple Endless Runner



```
private float timeBtwSpawn;
public float startTimeBtwSpawn;
public float decreaseTime;
public float minTime = 0.65f;

private void Update()
{
    if (timeBtwSpawn <= 0)
    {
        int rand = Random.Range(0, obstaclePatterns.Length);
        Instantiate(obstaclePatterns[rand], transform.position, Quaternion.identity);
        timeBtwSpawn = startTimeBtwSpawn;
        if (startTimeBtwSpawn > minTime) {
            startTimeBtwSpawn -= decreaseTime;
        }
    }
    else {
        timeBtwSpawn -= Time.deltaTime;
    }
}
}
```

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

public class Restart : MonoBehaviour {

    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.R)) {
            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
        }
    }
}
```

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

public class Obstacle : MonoBehaviour {

    public int damage = 1;
    public float speed;

    public GameObject effect;
    public GameObject explosionSound;

    private void Update()
    {
        transform.Translate(Vector2.left * speed * Time.deltaTime);

        private void OnTriggerEnter2D(Collider2D other)
        {
            if (other.CompareTag("Player")) {
                Instantiate(explosionSound, transform.position, Quaternion.identity);

                Instantiate(effect, transform.position, Quaternion.identity);

                other.GetComponent<Player>().health -= damage;
                Debug.Log(other.GetComponent<Player>().health);
                Destroy(gameObject);
            }
        }
    }
}
```

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

public class Destroyer : MonoBehaviour {

    public float lifetime;

    private void Start()
    {
        Destroy(gameObject, lifetime);
    }
}
```

```
using UnityEngine;

public class RepeatingBG : MonoBehaviour {

    public float speed;

    public float endx;
    public float startx;

    private void Update()
    {
        transform.Translate(Vector2.left * speed * Time.deltaTime);

        if (transform.position.x <= endx) {
            Vector2 pos = new Vector2(startx, transform.position.y);
            transform.position = pos;
        }
    }
}
```

```
using UnityEngine.UI;

public class Player : MonoBehaviour {

    private Vector2 targetPos;
    public float Yincrement;

    public int health = 3;

    public GameObject effect;
    public Text healthDisplay;
    public GameObject gameOver;
    public GameObject jellySound;

    private void Update()
    {
        healthDisplay.text = health.ToString();

        if (health <= 0) {
            gameOver.SetActive(true);
            Destroy(gameObject);
        }

        transform.position = Vector2.MoveTowards(transform.position, targetPos, speed * Time.deltaTime);

        if (Input.GetKeyDown(KeyCode.UpArrow) && transform.position.y < maxHeight) {
            Instantiate(jellySound, transform.position, Quaternion.identity);
            Instantiate(effect, transform.position, Quaternion.identity);
            targetPos = new Vector2(transform.position.x, transform.position.y + Yincrement);
        }
        else if (Input.GetKeyDown(KeyCode.DownArrow) && transform.position.y > minHeight) {
            Instantiate(jellySound, transform.position, Quaternion.identity);
            Instantiate(effect, transform.position, Quaternion.identity);
            targetPos = new Vector2(transform.position.x, transform.position.y - Yincrement);
        }
    }
}
```

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

public class SpawnPoint : MonoBehaviour {

    public GameObject obstacle;

    private void Start()
    {
        Instantiate(obstacle, transform.position, Quaternion.identity);
    }
}
```

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

public class ScoreManager : MonoBehaviour {

    public int score;
    public Text scoreDisplay;

    private void Update()
    {
        scoreDisplay.text = score.ToString();
    }

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Obstacle")) {
            score++;
            Debug.Log(score);
        }
    }
}
```

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

public class ScoreManager : MonoBehaviour {

    public int score;
    public Text scoreDisplay;

    private void Update()
    {
        scoreDisplay.text = score.ToString();
    }

    private void OnTriggerEnter2D(Collider2D other)
    {
        if (other.CompareTag("Obstacle")) {
            score++;
            Debug.Log(score);
        }
    }
}
```

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

public class SpawnPoint : MonoBehaviour {

    public GameObject obstacle;

    private void Start()
    {
        Instantiate(obstacle, transform.position, Quaternion.identity);
    }
}
```

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

public class Destroyer : MonoBehaviour {

    public float lifetime;

    private void Start()
    {
        Destroy(gameObject, lifetime);
    }
}
```

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

public class RepeatingBG : MonoBehaviour {

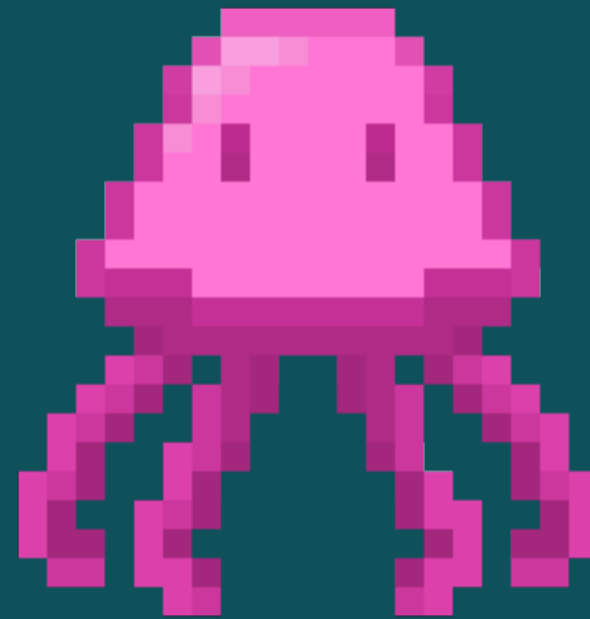
    public float speed;

    public float endx;
    public float startx;

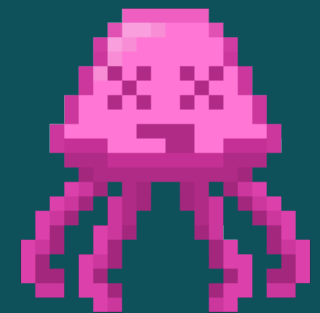
    private void Update()
    {
        transform.Translate(Vector2.left * speed * Time.deltaTime);

        if (transform.position.x <= endx) {
            Vector2 pos = new Vector2(startx, transform.position.y);
            transform.position = pos;
        }
    }
}
```

SCRIPTS



- 1 parallax enviro
- 1 character
- 1 animation
- 1 obstacle
- 1 death screen
- 1 UI element



2

5



<https://vimeo.com/314507658>

1

3



https://emiepinkbunny.itch.io/_jelly



TY GG WF