**HealCollect.cs**

using UnityEngine;

public class HealCollect : MonoBehaviour

{

[SerializeField] private int healthValue;

private void OnTriggerEnter2D(Collider2D collision)

{

Debug.Log("collide with object");

if (collision.tag == "Player")

{

Debug.Log("collide with player");

collision.GetComponent<Health>().AddHealth(healthValue);

gameObject.SetActive(false);

}

}

}

Đoạn mã này quản lý hành vi của một vật phẩm hồi máu trong trò chơi Unity. Khi người chơi va chạm với vật phẩm hồi máu, máu của người chơi sẽ được tăng lên và vật phẩm hồi máu sẽ biến mất khỏi trò chơi.

**HealthBar.cs**

using UnityEngine;

using UnityEngine.UI;

public class HealthBar : MonoBehaviour

{

[SerializeField] private Health playerHealth;

[SerializeField] private Image currentHealthBar;

private void Update()

{

currentHealthBar.fillAmount = playerHealth.currentHealth / playerHealth.startingHealth;

}

}

Đoạn mã này cập nhật máu hiện tại của người chơi lên thanh máu trong UI.



vật phẩm hồi máu và thanh máu

**Camera controller.cs**

using UnityEngine;

public class Cameracontroller : MonoBehaviour

{

[SerializeField] private float Camera\_speed = 2f;

public Transform player;

private int left\_right;

[SerializeField] private float offset\_x = 2f;

[SerializeField] private float offset\_y = 2f;

Vector3 velocity = Vector3.zero;

private void FixedUpdate()

{

if ((player.position.x > -8f && player.position.x < 6f) && (player.position.y > -43f && player.position.y < -36f))

{

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(-1.2f, -39.3f, -10f), ref velocity, Camera\_speed);

//this.transform.position = new Vector3(-1.2f, -39.3f, -10f);

}

else

{

if (Input.GetAxis("Horizontal") > 0f)

{

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(player.position.x + offset\_x, player.position.y + offset\_y, -10f), ref velocity, Camera\_speed);

left\_right = 1;

}

if (Input.GetAxis("Horizontal") < 0f)

{

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(player.position.x - offset\_x, player.position.y + offset\_y, -10f), ref velocity, Camera\_speed);

left\_right = -1;

}

if (Input.GetAxis("Vertical") < 0f)

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(player.position.x, player.position.y - 2 \* offset\_y, -10f), ref velocity, Camera\_speed);

else

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(player.position.x + left\_right\*offset\_x, player.position.y + offset\_y, -10f), ref velocity, Camera\_speed);

}

}

}

Đoạn mã này quản lý chuyển động của camera trong trò chơi Unity, theo dõi vị trí của người chơi và di chuyển mượt mà giữa các vị trí dựa trên chuyển động của người chơi và các giới hạn cụ thể.

Nếu người chơi ở vị trí khu vực spawn thì camera sẽ không di chuyển theo nhân vật mà chỉ đứng yên tại một vị trí cố định. Đoạn code sau thực hiện điều này:

if ((player.position.x > -8f && player.position.x < 6f) && (player.position.y > -43f && player.position.y < -36f))

{

transform.position = Vector3.SmoothDamp(transform.position, new Vector3(-1.2f, -39.3f, -10f), ref velocity, Camera\_speed);

//this.transform.position = new Vector3(-1.2f, -39.3f, -10f);

}

Nếu người chơi di chuyển sang phải (Input.GetAxis("Horizontal") > 0f), camera sẽ theo sau với một khoảng cách bù trừ theo trục x và y.

Nếu người chơi di chuyển sang trái (Input.GetAxis("Horizontal") < 0f), camera sẽ theo sau với một khoảng cách bù trừ ngược lại.

Nếu người chơi di chuyển xuống (Input.GetAxis("Vertical") < 0f), camera sẽ theo sau với một khoảng cách bù trừ theo trục y lớn hơn.

Trong các trường hợp khác, camera sẽ theo sau người chơi với khoảng cách bù trừ đã lưu trong biến left\_right.



player spawn