For the testing of my game I mostly used the Debug.Log() to print out to see if a method or function was called when it needed to be or if it was being called when it wasn't supposed to be and I would also use it if I have a variable that I wanted to change and I wanted to make sure that the value of it was being changed when I wanted it to be and not before.

Here is a example where you will see after all my testing that I wont delete my Logs to show myself what the difficulties were with these sections so I comment them out and these Logs were to help show the change it the healthPercentage variable at the end of the method and I would show myself the amount of damage that was being passed from the AI script to this class and Log it out to make sure it came across and that it was the right value and I would also check to make sure that the hit variable was being set to true at the right time and I would have a Log telling me when the attack function was being called.

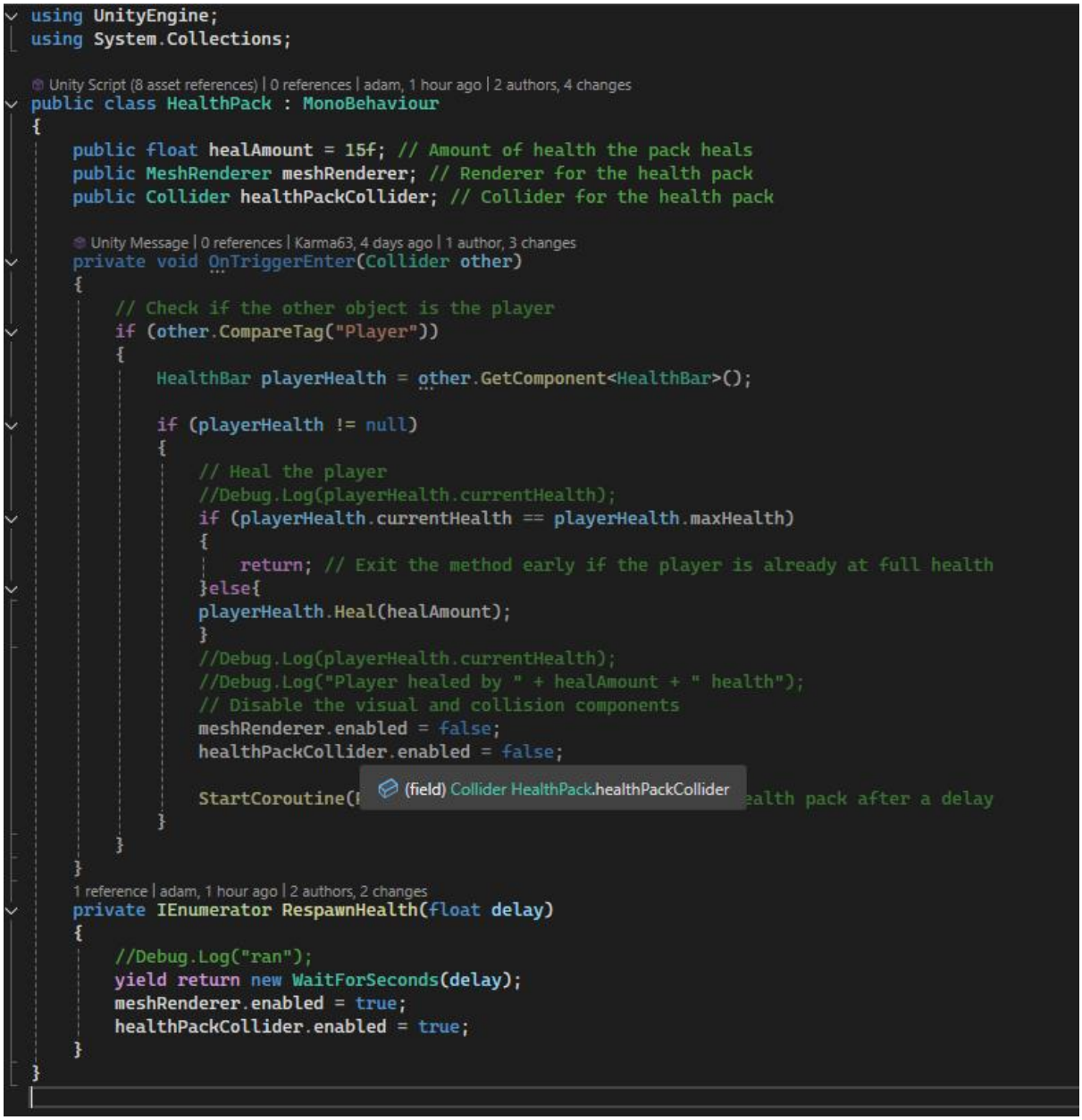
```
    //Debug.Log("Health Percentage: " + healthPercentage);
    // Adjust the width of the health bar based on the health percentage
    healthBarImage.sizeDelta = new Vector2(originalWidth * healthPercentage, healthBarImage.sizeDelta.y);
    Audio.PlayHealSound();
}

1 reference | adam, 58 minutes ago | 2 authors, 6 changes
public void TakeDamage(float amount)
{
    amount = PlayerDamage;
    //Debug.Log("Player took damage: " + amount + " health");
    // Decrease health by the given amount
    currentHealth -= amount;
    if (currentHealth <= 0){
        currentHealth = 0;
        dead();
    };

    Hit=true;
    if(Hit){
        //Debug.Log("Hit");
        StartCoroutine(Attack());
        Hit=false;
    }
    Sword.rectTransform.anchoredPosition = new Vector2(0,-400); // Adjust as needed
    HitSword.rectTransform.anchoredPosit
                                          Vector2 RectTransform.anchoredPosition { get; set; }
                                          The position of the pivot of this RectTransform relative to the anchor reference point.
    // Update the health bar
    UpdateHealthBar();
}

1 reference | adam, 58 minutes ago | 2 authors, 3 changes
private IEnumerator Attack()
{
//Debug.Log("Attack");
```

This is my script for my health packs that when the player runs over it with under 100% health they will be healed 15 points of health and how I tested this called was the same as before where I made sure that all variables that can be changed will be checked and all function or method called will tell the developer that that function ran.



But with unity there are time when you cant use a Debug.log() to help with testing and will instead have to use trial and error such as with a image in a canvas since the x and y for it is the x and y for the game field and the x and y in the code base is the x and y inside the canvas so the positioning of the image is trail and error of trying to get it in the position you want.