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

public class Widget_AnimalScratcher : MonoBehaviour
{
    public GameObject animalObject;
    public AnimalValues animalValues;
    public soAnimal sOAnimal;
    public float happinessIncreaseRate = 5f;
    public float bonusPoints = 0.0050f;
    public float bonusPointsRange = .2f;
    public Slider Happiness_Bar;
    Canvas_ReceptionRoom canvas_ReceptionRoom;

    //public Image bonusPointsBar;
    public Image bonusPointsPrefab;
    public RectTransform canvasRectTransform;
    //public CareRoomManager careRoomManager;


    private bool isScratching = false;
    public float currentHappiness = 0f;
    private float currentBonusPoints = 0f;
    [SerializeField] GameObject vfx;
    Vector3 mouse;
    public void Init(soAnimal _sOAnimal)
    {
        //animalValues = GetComponent<AnimalValues>();
        //careRoomManager = gameObject.AddComponent<CareRoomManager>();
        sOAnimal = _sOAnimal;
    }
    private void Start()
    {
        currentHappiness = sOAnimal.f_happiness;
        vfx.SetActive(false);
        // Get the RectTransform of the canvas
        canvasRectTransform =
GetComponentInParent<Canvas>().GetComponent<RectTransform>();
    }

    private void Update()
    {
```

```
        sOAnimal.f_happiness = currentHappiness;
        //Add check value for bar here so the bar updates while you look at the animal.
        Debug.Log(currentHappiness);
        Happiness_Bar.value = currentHappiness;
        if (Input.GetMouseButtonDown(0))
        {
          // Mouse();

            isScratching = true;

            // Check if the mouse is hovering over a bonusPointsPrefab
            RaycastHit2D hit =
Physics2D.Raycast(Camera.main.ScreenToWorldPoint(Input.mousePosition), Vector2.zero);
            if (hit.collider != null && hit.collider.gameObject == bonusPointsPrefab.gameObject)
            {
                CollectBonusPoints();
            }
        }
        else if (Input.GetMouseButtonUp(0))
        {
            isScratching = false;
        }

        if (isScratching)
        {
            ScratchAnimal();
            CheckForBonusPoints();

            // Update the happiness Slider
            UpdateHappyBar();

        }
    }

    private void ScratchAnimal()
    {
        // Increase happiness
        currentHappiness += happinessIncreaseRate * Time.deltaTime;


        //currentHappiness = Mathf.Clamp(currentHappiness, 0f, 100f);
        //happinessBar. = currentHappiness / 100f;

        // Update the animal's appearance or behavior based on happiness
```

```
        // Example: change animation, play sound, etc.

        // Emit particle effects from the animal
        // Example: Instantiate particle system at the animal's position
        //ParticleSystem particleSystem = Instantiate(particlePrefab,
animalObject.transform.position, Quaternion.identity);
        //particleSystem.Play();

        // Emit particle effects from the mouse position
        // Example: Instantiate particle system at the mouse position
        //Vector3 mousePosition = Camera.main.ScreenToWorldPoint(Input.mousePosition);
        //ParticleSystem particleSystem = Instantiate(particlePrefab, mousePosition,
Quaternion.identity);
        //particleSystem.Play();
    }

    private void CheckForBonusPoints()
    {
        // Increase bonus points
        //currentBonusPoints += bonusPoints; ========================
        //bonusPointsBar.fillAmount = currentBonusPoints / 100f;

        // Check if currentHappiness reached max
        if (currentHappiness >= 100f)
        {
            return;
        }

        // Delete the bonusPointsPrefab object
        Destroy(bonusPointsPrefab.gameObject);

        // Calculate random position within the canvas boundaries
        float minX = canvasRectTransform.rect.xMin;
        float maxX = canvasRectTransform.rect.xMax;
        float minY = canvasRectTransform.rect.yMin;
        float maxY = canvasRectTransform.rect.yMax;

        Vector3 randomPosition = new Vector3(Random.Range(minX, maxX),
Random.Range(minY, maxY), 0f);

        // Create the bonus points object in a new random location
        bonusPointsPrefab = Instantiate(bonusPointsPrefab, randomPosition, Quaternion.identity);
        bonusPointsPrefab.transform.SetParent(canvasRectTransform);
```

```
        // Increase currentHappiness
        //currentHappiness += bonusPoints; ========================
        //currentHappiness = Mathf.Clamp(currentHappiness, 0f, 100f);
========================


    }

    private void UpdateHappyBar()
    {
        float ratio = currentHappiness; // sOAnimal.f_max;

            // Calculate the happiness ratio between 0 and 1
            // float ratio = currentHappiness / 100f;

        // Update the value of the happiness Slider
        Happiness_Bar.value = ratio;

       // Happiness_Bar.value = ratio;
      //  t_Happy.text = (ratio * 100).ToString("0") + "%";
    }
    public void CollectBonusPoints()
    {
       // Increase bonus points
        currentBonusPoints += bonusPoints;
     //   bonusPointsBar.fillAmount = currentBonusPoints / 100f;
    }

   public void OnbackPressed()
    {
        GameManager.gm.careRoomManager.animalValues.isBeingPet = false;
        GameManager.gm.careRoomManager.animalValues.isPetting = false;
        GameManager.gm.careRoomManager.isWaiting = true;
        //canvas_ReceptionRoom.canvas_hud[0].SetActive(true);

        GameManager.gm.cameraManager.MoveCamera(2);
        Destroy(Happiness_Bar.gameObject);
        Destroy(this.gameObject);
    }

   /* private void Mouse()
    {
        if (Input.GetMouseButtonDown(0))
        {
```

```
            mouse = Camera.current.ScreenToWorldPoint(Input.mousePosition);
            vfx.SetActive(true);
            vfx.transform.position = new Vector3(mouse.x, mouse.y, 0f);
        }
        if (Input.GetMouseButtonDown(0))
        {
            vfx.SetActive(false);
        }
    }
    */
}
```