

```

using UnityEngine;
using System.Collections;
using UnityEngine.UI;

public enum Stage{
    /* en inicio/fin aplicar daños de veneno/etc, revisar si alguno esta muerto */
    inicio, elegir, entrenador, atq1, atq2, atq3, resultado, victoria, derrota
}

public class Battle : MonoBehaviour {

    //catchRate se carga en la escena, 0 para entrenadores, N para salvajes
    public int catchRate = 0;

    private int battleStage,battleStageOp;
    private bool waiting = false;
    [SerializeField] GameObject actionPanel;
    [SerializeField] GameObject atkPanel;
    [SerializeField] GameObject chgPanel;
    private Accion act1;
    private Accion act2;
    public Monstruo userMon, opoMon;

    [SerializeField] GameObject projectile1;
    [SerializeField] GameObject projectile2;

    public void Shoot(direccion i, string img, Monstruo m){
        GameObject p;
        if(((i == direccion.otros)&&(m==userMon))||((i == direccion.mismo)&&(m==opoMon)))
            p = Instantiate(projectile1);
        else
            p = Instantiate(projectile2);
        p.GetComponent<ProjectileScr>().imgName = img;
        p.transform.SetParent(GameObject.Find("Console").transform);
    }

    public ClaseJugador user;
    private string opoN;
    public Entrenador oponent;

    [SerializeField] Slider userCauge;
    [SerializeField] Slider opoCauge;
    [SerializeField] Text userName;
    [SerializeField] Text opoName;

    void Start () {
        user = (ClaseJugador)Entrenador.CreateTrainer("ClaseJugador","PEPE");
        opoN = PlayerPrefs.GetString("Entrenador");
        oponent = (Entrenador)Entrenador.CreateTrainer(opoN,opoN);
        PlayerPrefs.DeleteKey("Entrenador");
        battleStageOp = battleStage = (int)Stage.elegir;
        userMon = user.equipo[user.activo];
        opoMon = oponent.equipo[oponent.activo];
        act1 = user.accionEntrenador();
        act2 = Accion.CreateAccion("Elegir");

        GameObject.Find("OponentIm").GetComponent<Image>().sprite = Resources.Load(opoMon.imgDir, typeof(Sprite)
    ) as Sprite;
        GameObject.Find("UserIm").GetComponent<Image>().sprite = Resources.Load(userMon.imgDir, typeof(Sprite)
    ) as Sprite;
        actionPanel.SetActive(true);
        if(oponent.catchRate <= 0){
            GameObject.Find("ItemBtn").SetActive(false);
        }
    }
}

```

```

void Update () {
    if(userMon.estado.statActual.vida == 0){
        SaveMonster.AddMonster(userMon,false);
        act1.stg = act2.stg = Stage.elegir;
        user.clicks = accionesEntrenador.nula;
        user.menuActivo = menus.capa1;
        if( user.Change() < 0){
            battleStage = (int) Stage.derrota;
        }
    }
    if(opoMon.estado.statActual.vida == 0){
        userMon.AddExp(opoMon.exp);
        act1.stg = act2.stg = Stage.elegir;
        if( oponent.Change() < 0){
            battleStage = (int) Stage.victoria;
        }
    }
    if((battleStage == (int)Stage.resultado) || (battleStage == (int)Stage.derrota) || (battleStage == (int)
)Stage.victoria)){
        Salir(battleStage);
    }

    InitPanels();

    if(act1.stg == Stage.elegir){
        battleStage = (int)Stage.elegir;
        act1 = user.accionEntrenador();
    }
    if(userMon.estado.statActual.vida == 0){
        if( user.Change() < 0){
            battleStage = (int) Stage.resultado;
        }
    }
    if(act2.stg == Stage.elegir){
        battleStageOp = (int)Stage.elegir;
        act2 = oponent.accionEntrenador();
    }
    if(opoMon.estado.statActual.vida == 0){
    }
    if(act1.stg != Stage.elegir){
        if(battleStage == (int)act1.stg){
            act1.ac();
            act1.stg = Stage.elegir;
        }else{
            if(!waiting){
                waiting = true;
                StartCoroutine(Wait(0));
            }
        }
    }
    if(act2.stg != Stage.elegir){
        if(battleStageOp == (int)act2.stg){
            act2.ac();
            act2.stg = Stage.elegir;
        }else{
            if(!waiting){
                waiting = true;
                StartCoroutine(Wait(1));
            }
        }
    }
}

void OnGUI(){
    userCauge.maxValue = userMon.GetStats().vida;
    opoCauge.maxValue = opoMon.GetStats().vida;
}

```

```

opoName.text = opoMon.nombre+" LV:"+opoMon.lv;
userName.text = userMon.nombre+" LV:"+userMon.lv;
if(userCauge.value > userMon.estado.statActual.vida)
    userCauge.value-=0.5f;
if(userCauge.value < userMon.estado.statActual.vida)
    userCauge.value+=0.5f;
if(opoCauge.value > opoMon.estado.statActual.vida)
    opoCauge.value-=0.5f;
if(opoCauge.value < opoMon.estado.statActual.vida)
    opoCauge.value+=0.5f;

}

IEnumerator Wait(int i){
    float t;
    if(i==0){
        t = opoMon.estado.statActual.velocidad/userMon.estado.statActual.velocidad;
        if(t > 1f){
            t = 1f;
        }
        if(t < 0.5f){
            t = 0.5f;
        }
        yield return new WaitForSeconds(t);
        battleStage++;
    }else {
        t = userMon.estado.statActual.velocidad/opoMon.estado.statActual.velocidad;
        if(t > 1f){
            t = 1f;
        }
        if(t < 0.5f){
            t = 0.5f;
        }
        yield return new WaitForSeconds(t);
        battleStageOp++;
    }
    waiting = false;
}

private void InitPanels(){
    if(userMon != user.equipo[user.activo]){
        SaveMonster.AddMonster(userMon,false);
        userMon = user.equipo[user.activo];
        Log.AddLine(user.nombre+": 'Ve "+userMon.nombre+"!'");
        act1.stg = Stage.elegir;
    }
    if(opoMon != oponent.equipo[oponent.activo]){
        opoMon = oponent.equipo[oponent.activo];
        Log.AddLine(oponent.nombre+": 'Ve "+opoMon.nombre+"!'");
        act2.stg = Stage.elegir;
    }
    GameObject.Find("OponentIm").GetComponent<Image>().sprite = Resources.Load(opoMon.imgDir, typeof(Sprite)
)) as Sprite;
    GameObject.Find("UserIm").GetComponent<Image>().sprite = Resources.Load(userMon.imgDir, typeof(Sprite)
) as Sprite;

    user.target = opoMon;
    user.source = userMon;
    oponent.target = userMon;
    oponent.source = opoMon;
    if(user.menuActivo == menus.capa1){
        actionPanel.SetActive(true);
    }else{
        actionPanel.SetActive(false);
    }
    if(user.menuActivo == menus.capaAtaque){

```

```
        atkPanel.SetActive(true);
    }else{
        atkPanel.SetActive(false);
    }
    if(user.menuActivo == menus.capaCambio){
        chgPanel.SetActive(true);
    }else{
        chgPanel.SetActive(false);
    }
}

private void Salir(int res){
    string result = "?";
    switch(res){
        case (int)Stage.victoria:
            Log.AddLine("Has salido victoriosos!");
            result = "Victoria";
            break;
        case (int)Stage.derrota:
            Log.AddLine("Has sido derrotado!");
            result = "Derrota";
            break;
    }
    SaveMonster.AddMonster(userMon, false);
    PlayerPrefs.SetString("Result", result);
    Application.LoadLevel("battleResult");
}
}
```