**Appendix**

using UnityEngine;

using UnityEngine.UI;

using System.Collections;

using UnityEngine.AI;

public class intothelight : MonoBehaviour {

    bool yell = false;

public Transform target;

public Transform target2;

    public Transform player;//set target from inspector instead of looking in Update

public float speed = 3f;

public bool follow=false;

public bool trust=false;

    public float chestlikely=50;

    public float chestpercent;

    public float buttonlikely=50;

    public float buttonpercent;

    public float traplikely = 50;

    public float trappercent;

    public int choice;

public int xp=0;

public int want=0;

public bool attack=false;

public int once=0;

public int once2=1;

public int goodtoo=1;

private int percent=21;

    private int timer = 0;

    private Random r = new Random();

    public int knowchest = 0;

    public int knowbutton=0;

    public static bool busy = false;

    public static int health = 100;

    public Slider healthbar;

   public static int newthing = 0;

    Vector3 a = new Vector3();

    [SerializeField]

    NavMeshAgent agent;

void Start () {

        a = new Vector3(transform.rotation.x, 0, 0);

        agent = this.GetComponent<NavMeshAgent>();

        setdestination();

}

    void setdestination()

    {

    }

void Update(){

        healthbar.value = health;

        Vector3 targetvector;

        if (goodtoo==1)

trust = false;

if (goodtoo==0)

trust = true;

//rotate to look at the player

if(follow==true)

{

transform.LookAt(target.position-new Vector3(0,target.position.y-transform.position.y,0));

transform.Rotate(new Vector3(0,0,0),Space.Self);//correcting the original rotation

            targetvector = target.transform.position;

            agent.SetDestination(targetvector);

            //move towards the player

        }

        if(follow== false&&busy==false)

        {

            transform.LookAt(target2.position - new Vector3(0, target2.position.y - transform.position.y, 0));

            transform.Rotate(new Vector3(transform.rotation.x, 0, 0), Space.Self);//correcting the original rotation

            targetvector = target2.transform.position;

            agent.SetDestination(targetvector);

            //move towards the player

        }

       /\* if (Input.GetKeyDown(KeyCode.Q) && trust == false)

        {

            while (timer <= 10)

            {

                if (timer == 10) {

                    goodtoo = Random.Range(0, 1);

                    timer = 0;

                }

                timer++;

            }

            timer++;

        }\*/

if (Input.GetMouseButtonDown(1) && trust==true)

        {

follow=true;

}

if (Input.GetMouseButtonUp(1)&&trust==true) {

follow = false;

            goodtoo = 0;

}

        if (Input.GetKeyDown(KeyCode.E) && trust == true)

        {

            yell=true;

        }

        if (Input.GetKeyUp(KeyCode.E) && trust == true)

        {

            yell = false; ;

            goodtoo = 0;

        }

    }

    void OnTriggerStay(Collider other)

    {

        if (other.tag == "enemy")

        {

            busy = true;

            transform.LookAt(other.transform.position- new Vector3(0, other.transform.position.y - transform.position.y, 0));

            transform.Rotate(new Vector3(0, 0, 0), Space.Self);//correcting the original rotation

            Vector3 targetvector = other.transform.position;

            agent.SetDestination(targetvector);

            //move towards the player

            if (Vector3.Distance(transform.position, other.transform.position) > 1.0f)

            {//move if distance from target is greater than 1

                transform.Translate(0.0f, 0.0f, speed \* 1.0f \* Time.deltaTime);

            }

            else

            {

                busy = false;

                newthing = 0;

            }

        }

        if (other.tag == "goodchest")

        {

            if(knowchest==1)

            {

                busy = true;

                chestpercent =Random.Range(1,100);

                knowchest = 0;

            }

            if (chestpercent > chestlikely && busy==true&&newthing==0||follow==true&&newthing==0)

            {

                transform.LookAt(other.transform.position- new Vector3(0, other.transform.position.y - transform.position.y, 0));

                transform.Rotate(new Vector3(0, 0, 0), Space.Self);//correcting the original rotation

                Vector3 targetvector = other.transform.position;

                agent.SetDestination(targetvector);

                //move towards the player

                if (Vector3.Distance(transform.position, other.transform.position) > 4.0f)

                {//move if distance from target is greater than 1

                    transform.Translate(0.0f, 0.0f, speed \* 1.0f \* Time.deltaTime);

                }

                else

                {

                    chestpercent = -1;

                    chestlikely -= chestlikely \* .05f;

                    knowchest = 1;

                    busy = false;

                    newthing = 1;

                }

            }

            else if(yell==true&& chestpercent > chestlikely && busy == true && newthing == 0)

            {

                chestpercent = -1;

                chestlikely += chestlikely \* .1f;

                knowchest = 1;

                busy = false;

                newthing = 1;

            }

            else

            { busy = false;

                knowchest = 1;

                newthing = 1;

            }

        }

        if (other.tag == "goodbutton")

        {

            if (knowbutton == 1)

            {

                buttonpercent = Random.Range(1,100);

                knowbutton = 0;

                busy = true;

            }

            if (buttonpercent> buttonlikely&&busy==true&&newthing==0||follow==true)

            {

                transform.LookAt(other.transform.position- new Vector3(0, other.transform.position.y - transform.position.y, 0));

                transform.Rotate(new Vector3(0, 0, 0), Space.Self);//correcting the original rotation

                Vector3 targetvector = other.transform.position;

                agent.SetDestination(targetvector);

                //move towards the player

                if (Vector3.Distance(transform.position, other.transform.position) >= 4.0f)

                {//move if distance from target is greater than 1

                    transform.Translate(0.0f, 0.0f, speed \* 1.0f \* Time.deltaTime);

                }

                else

                {

                    buttonpercent = -1;

                    buttonlikely -= buttonlikely \* .05f;

                    knowbutton = 1;

                    busy = false;

                    newthing = 1;

                }

            }

            else if(yell==true && buttonpercent > buttonlikely && busy == true && newthing == 0)

            {

                buttonpercent = -1;

                buttonlikely += buttonlikely \* .1f;

                knowbutton = 1;

                busy = false;

                newthing = 1;

            }

            else

            {

                busy = false;

                knowbutton = 1;

                newthing = 1;

            }

        }

        if (other.tag == "trap")

        {

                trappercent = Random.Range(1, 100);

                busy = true;

            if (trappercent < traplikely && busy == true && newthing == 0 || follow == true)

            {

                transform.LookAt(other.transform.position - new Vector3(0, other.transform.position.y - transform.position.y, 0));

                transform.Rotate(new Vector3(0, 0, 0), Space.Self);//correcting the original rotation

                Vector3 targetvector = other.transform.position;

                agent.SetDestination(targetvector);

                //move towards the player

                if (Vector3.Distance(transform.position, other.transform.position) >= 10.0f)

                {//move if distance from target is greater than 1

                    transform.Translate(0.0f, 0.0f, speed \* 1.0f \* Time.deltaTime);

                }

                else

                {

                    trappercent = -1;

                    traplikely += traplikely \* .05f;

                    health = 0;

                    busy = false;

                    newthing = 1;

                }

            }

            else if (yell == true || trappercent > traplikely && busy == true && newthing == 0)

            {

                trappercent = -1;

                traplikely -= traplikely \* .1f;

                busy = false;

                newthing = 1;

                Vector3 targetvector = player.position;

                agent.SetDestination(targetvector);

            }

            else

            {

                busy = false;

                knowbutton = 1;

                newthing = 1;

            }

        }

}

    void OnTriggerExit(Collider other)

    {

        newthing = 0;

        busy = false;

    }

}

scene transition

using UnityEngine;

using UnityEngine.SceneManagement;

public class transition : MonoBehaviour {

    public string scenetarget;

    public Transform player;

    // Use this for initialization

    void Start () {

}

// Update is called once per frame

void Update () {

        if (Vector3.Distance(player.position, this.transform.position) < 4.5)

        {

            SceneManager.LoadScene(scenetarget);

        }

    }

}

player movement

using UnityEngine;

using System.Collections;

using UnityEngine.SceneManagement;

public class playermovement : MonoBehaviour {

    public int speed = 1;

    // Use this for initialization

    void Start () {

}

// Update is called once per frame

void Update () {

if(Input.GetKey (KeyCode.W))

{

transform.Translate (0.0f,0.0f,-1.0f\*speed\*Time.deltaTime);

}

if(Input.GetKey (KeyCode.S))

{

transform.Translate (0.0f,0.0f,1.0f\*speed\*Time.deltaTime);

}

if(Input.GetKey (KeyCode.D))

{

transform.Rotate (0.0f,1.0f,0.0f\*Time.deltaTime);

}

if(Input.GetKey (KeyCode.A))

{

transform.Rotate (0.0f,-1.0f,0.0f\*Time.deltaTime);

}

if(Input.GetKey (KeyCode.Space))

{

transform.Translate (0.0f,1.0f,0.0f\*Time.deltaTime);

}

if(Input.GetKey (KeyCode.LeftShift))

{

transform.Translate (0.0f,-1.0f,0.0f\*Time.deltaTime);

}

    }

}

spot light toggle

using UnityEngine;

using System.Collections;

public class lighttoggle : MonoBehaviour {

// Use this for initialization

void Start () {

}

// Update is called once per frame

void Update () {

if (Input.GetMouseButtonDown (1)) {

GetComponent<Light>().enabled=true;

}

if (Input.GetMouseButtonUp (1)) {

GetComponent<Light>().enabled = false;

}

}

}

enemy script

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class enemy : MonoBehaviour {

    public Transform player;

     Animator anim;

    public static int health = 1000;

// Use this for initialization

void Start () {

        anim = GetComponent<Animator>();

    }

    // Update is called once per frame

    void Update ()

    {

        if(health<=0)

        {

            intothelight.busy = false;

            intothelight.newthing = 0;

            Destroy(this.gameObject);

        }

        Vector3 direction = player.position - this.transform.position;

        float angle = Vector3.Angle(direction, this.transform.forward);

        if (Vector3.Distance(player.position, this.transform.position)<10&&angle<30)

        {

            this.transform.rotation = Quaternion.Slerp(this.transform.rotation, Quaternion.LookRotation(direction), 0.05f);

            anim.SetBool("isidle", false);

            if (direction.magnitude > 5)

            {

                this.transform.Translate(0, 0, 0.05f);

                anim.SetBool("iswalking", true);

                anim.SetBool("isatk", false);

            }

            else

            {

                anim.SetBool("isatk", true);

                anim.SetBool("iswalking", false);

            }

        }

        else

        {

            anim.SetBool("isidle", true);

            anim.SetBool("iswalking", false);

            anim.SetBool("isatk", false);

        }

}

    void OnTriggerStay (Collider other)

    {

        if (other.tag=="playerhit")

        {

            health -= 5;

        }

    }

}