using System.Collections;

using UnityEngine;

using UnityEngine.AI;

using UnityEngine.UI;

public class AIController : MonoBehaviour {

[Header("//Stats//")]

public int level = 20;

public float startHealth = 250.0f;

[Header("//Gun//")]

public Transform muzzle;

public GameObject bullet;

public float RPM;

public bool isAware = false;

[Header("UI")]

public Image healthBar;

public GameObject canvas;

[Header("//Path//")]

public GameObject player;

public float patrolDist = 20.0f;

public Transform eyes;

private float wait = 0f;

public float alertRadius = 20f;

[Header("//Audio//")]

public AudioClip[] walkSoundClips;

public AudioSource spotNoise;

[Header("//Components//")]

public NavMeshAgent nav;

public Animator anim;

public CapsuleCollider cc;

[Header("Particle System")]

public ParticleSystem headPS;

public ParticleSystem rightPS;

public ParticleSystem leftPS;

public ParticleSystem muzzleFlash;

[Header("GameObjects")]

public GameObject rightUpperArm;

public GameObject rightLowerArm;

public GameObject leftUpperArm;

public GameObject leftLowerArm;

public GameObject head;

private float currentHealth;

private AudioSource aSource;

public string state = "Idle";

private bool isAlive = true;

private bool highAlert = false;

private Rigidbody[] bodyParts;

private float fireRate;

private bool isPlaying = false;

void Start()

{

aSource = GetComponent<AudioSource>();

bodyParts = transform.GetComponentsInChildren<Rigidbody>();

EnableRagdoll(false);

currentHealth = startHealth;

fireRate = RPM;

}

void Awake()

{

if (player == null)

{

player = GameObject.FindGameObjectWithTag("Player");

}

}

void Update()

{

if (currentHealth <= 0)

{

Die();

currentHealth = 0f;

canvas.SetActive(false);

}

Shoot();

if (isAlive)

{

anim.SetFloat("Velocity", nav.velocity.magnitude);

if (state == "Idle")

{

isAware = false;

//pick a random place to walk to.

Vector3 randomPos = Random.insideUnitSphere \* alertRadius;

NavMeshHit navHit;

NavMesh.SamplePosition(transform.position + randomPos, out navHit, patrolDist, NavMesh.AllAreas);

if (highAlert)

{

NavMesh.SamplePosition(player.transform.position + randomPos, out navHit, patrolDist, NavMesh.AllAreas);

alertRadius += 5f;

if (alertRadius > 20f)

{

highAlert = false;

nav.speed = 1.5f;

anim.speed = 1f;

}

}

nav.SetDestination(navHit.position);

state = "Walk";

}

//Walk

if (state == "Walk")

{

isAware = false;

if (nav.remainingDistance <= nav.stoppingDistance && !nav.pathPending)

{

state = "Search";

wait = 5f;

}

}

//Search

if (state == "Search")

{

isAware = false;

if (wait > 0f)

{

wait -= Time.deltaTime;

}

else

{

state = "Idle";

}

}

//Chase

if (state == "Chase")

{

isAware = true;

nav.destination = player.transform.position;

float distance = Vector3.Distance(transform.position, player.transform.position);

if (distance > 50f)

{

//StartCoroutine(HuntDelay());

state = "Hunt";

}

}

//Hunt

if (state == "Hunt")

{

isAware = false;

if (nav.remainingDistance <= nav.stoppingDistance && !nav.pathPending)

{

state = "Search";

wait = 5f;

highAlert = true;

alertRadius = 5f;

CheckSight();

}

}

}

}

public void FootStep(int \_num)

{

aSource.clip = walkSoundClips[\_num];

aSource.Play();

}

public void CheckSight()

{

if (isAlive)

{

RaycastHit rayHit;

if (Physics.Linecast(eyes.position, player.transform.position, out rayHit))

{

if (rayHit.collider.gameObject.tag == "Player")

{

if (state != "Kill")

{

state = "Chase";

nav.speed = 3.0f;

anim.speed = 2.0f;

spotNoise.pitch = 1.1f;

spotNoise.Play();

}

}

}

}

}

void Shoot()

{

fireRate -= Time.deltaTime;

if (fireRate <= 0 && isAlive && isAware)

{

Instantiate(bullet, muzzle.position, muzzle.rotation);

fireRate = RPM;

muzzleFlash.Play();

}

}

public void TakeDamage(float damage)

{

currentHealth -= damage;

if (state == "Walk" || state == "Idle")

{

state = "Search";

}

if (state == "Search" || state == "Hunt")

{

state = "Chase";

}

healthBar.fillAmount = currentHealth / startHealth;

}

public void Die()

{

isAlive = false;

nav.isStopped = true;

isAware = false;

cc.enabled = false;

rightUpperArm.transform.parent = null;

rightLowerArm.transform.parent = null;

leftUpperArm.transform.parent = null;

leftLowerArm.transform.parent = null;

head.transform.parent = null;

StartCoroutine(Despawn());

if (!isPlaying)

{

headPS.Play();

rightPS.Play();

leftPS.Play();

isPlaying = true;

}

EnableRagdoll(true);

}

public void EnableRagdoll(bool value)

{

anim.enabled = !value;

for (int i = 0; i < bodyParts.Length; i++)

{

bodyParts[i].isKinematic = !value;

}

}

IEnumerator Despawn()

{

yield return new WaitForSeconds(3.0f);

Destroy(this.gameObject);

}

}