using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class enemyPathfinding : MonoBehaviour

{

public int enemySpeed;

public int detectDistance;

float deltaPosX;

float deltaPosZ;

// Start is called before the first frame update

void Start()

{

}

// Update is called once per frame

void Update()

{

// dit zorgt dat de vijanden geen colission hebben met muren zodat ze

// niet vast komen te zitten bij een hoek van een muur dit is een bug

// die ik later ga oplossen

int layerMask = 1 << 8;

// zoekt naar een object met de naam speler en berekent de afstand

// tussen de speler en de vijand

GameObject Player = GameObject.Find("player");

deltaPosX = transform.position.x - Player.transform.position.x;

deltaPosZ = transform.position.z - Player.transform.position.z;

// bij de code van pickles was een read me die zij hoe je de code kon

// gebruiken hieronder staat hoe ik denk dat het werkt

if (deltaPosX > -detectDistance && deltaPosX < detectDistance && deltaPosZ + 5 > -detectDistance && deltaPosZ < detectDistance && float.IsNaN(FlowFieldProvider.GetVector(transform.position).x) == false && float.IsNaN(FlowFieldProvider.GetVector(transform.position).y) == false && float.IsNaN(FlowFieldProvider.GetVector(transform.position).z) == false)

{

// gebruikt de positie van de vijand om een vector 3 van de

// flowfieldprovider in de variabele direction te laden

Vector3 direction = FlowFieldProvider.GetVector(transform.position);

// controleert op colissions hij negeert de colissions met de muren

// (anders is er een bug)

// en zorgt ervoor dat vijanden, pets en de speler niet op dezelfde

// locatie staan zodat je kan zien dat er meerdere vijanden staan

if (!Physics.Raycast(transform.position, direction, enemySpeed \* Time.deltaTime + 5, layerMask))

{

// vermenigvuldigt de waarde van directie met de snelheid en telt dat

// bij de positie van de vijand op

transform.position = transform.position + direction \* enemySpeed / 10;

}

}

}

}