ПРИЛОЖЕНИЕ С

(обязательное)

Листинг программы

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class bloodyScreen : MonoBehaviour{

public GameObject healthMonstr;

public GameObject bloodyScree;

void Start(){

bloodyScree.SetActive(false);}

void Update(){

if (healthMonstr == false){

bloodyScree.SetActive(false);

}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class DeadPers : MonoBehaviour {

public void RestarT()

{Application.LoadLevel(3) ;

}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class DestroyPatron : MonoBehaviour{

private float lofeTime = 5;

private void OnCollisionEnter(Collision collision){

if (gameObject.name == "Potron(Clone)"){

Destroy(gameObject);}}

void Start(){

if (gameObject.name == "Potron(Clone)"){

Destroy(gameObject, lofeTime);}

}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.Collections;

public class DontDestr : MonoBehaviour {

private static DontDestr instance;

void Awake(){

if (!instance) // экземпляр{

instance = this;}else{

Destroy(this.gameObject);}

DontDestroyOnLoad(this.gameObject); // не разрушать перса на 2 уровн

}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

[ExecuteInEditMode]

public class EfectBlood : MonoBehaviour {

public Material ShaderTexture;

void InRenderImage(RenderTexture cameraViev, RenderTexture shaderView){

Graphics.Blit(cameraViev,shaderView,ShaderTexture);

}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class EnerjiPlayer : MonoBehaviour {

public Slider sliders;

public float stamina;// общая энерг

public GameObject stamiN

void Update(){

sliders.value = stamina;

if (Input.GetKey(KeyCode.R)) //восст энергии{

stamina += 1f;}

if (stamina > 100){

stamina = 100;}

if (stamina < 0){

stamina = 0;}

if (Input.GetKey(KeyCode.Space)){

stamina -= 6f \* Time.deltaTime;}

if (Input.GetKey(KeyCode.W)){

stamina -= 2f \* Time.deltaTime;}

if (Input.GetKey(KeyCode.S)){

stamina -= 2f \* Time.deltaTime;}

if (Input.GetKey(KeyCode.A)){

stamina -= 2f \* Time.deltaTime;}

if (Input.GetKey(KeyCode.D)){

stamina -= 2f \* Time.deltaTime;}

if (stamiN == true) {

if (stamina <= 0){

stamiN.GetComponent<Pers>().speed = 5;}

if (stamina >= 0){

stamiN.GetComponent<Pers>().speed=20;}}

else { }}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Fonary : MonoBehaviour {

void Update () {

if (Input.GetKeyDown(KeyCode.F))

GetComponent<Light>().enabled = !GetComponent<Light>().enabled;}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class GlavMenuCurs : MonoBehaviour {

void Start () {

Cursor.visible = true;}

void Update () {}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class helathPlayer : MonoBehaviour {

public int health;

public Slider slider;

public Text txt;

public GameObject endText;

public GameObject Camera2;

public GameObject pricel;

public GameObject bloodyscreen;

public GameObject bloodyscreen1;

void Start () {}

// После смерти

void Update () {

slider.value = health;

if (health <= 0){

Cursor.visible = true;

endText.SetActive(true);

Camera2.SetActive(true);

Destroy(gameObject);

pricel.SetActive(false);

Destroy(bloodyscreen1);

Destroy(bloodyscreen);}

txt.text=health+"%";}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class Konec : MonoBehaviour {

[Header("Загружаемая сцена")]

public int sceneID;

private bool up;

private float time;

void Start(){}

void Update(){

if (up == true){

time += 0.1f;

if (time >= 2){

Destroy(gameObject); //табл исчезает}}}

private void OnTriggerEnter(Collider col){

if (col.tag == "Player"){

SceneManager.LoadSceneAsync(sceneID);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.Runtime.InteropServices;

public class MenuActive : MonoBehaviour{

public GameObject menu;

public static bool GameIsPouse = false;

[SerializeField]

private GameObject player;

Pers controller;

void Start(){

Time.timeScale = 1f;

Cursor.visible = false;

controller = player.GetComponent<Pers>();}

void Update(){

if (Input.GetKeyDown(KeyCode.Escape)){

if (GameIsPouse) {

Resume(); }

else {

Pouse();}}}

void Resume(){

menu.SetActive(false);

Cursor.visible = false;

Time.timeScale = 1f;

GameIsPouse = false;

controller.enabled = true; // включение скрипта}

void Pouse(){

menu.SetActive(true);

Time.timeScale = 0f;

Cursor.visible = true;

GameIsPouse = true;

controller.enabled = false;}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

public class MenuButton : MonoBehaviour {

public GameObject settings;

public GameObject Player;

public GameObject instruction;

[System.Serializable]

public class Position{

public float x;

public float y;

public float z;}

public void StartGame(){

Application.LoadLevel(3);}

public void LoidGame(){

// загрузить файл по позициям

if (File.Exists(Application.dataPath + "/saves/save.sm")){

FileStream fs = new FileStream(Application.dataPath + "/saves/save.sm", FileMode.Open);

BinaryFormatter formatter = new BinaryFormatter();

try{

Position pos = (Position)formatter.Deserialize(fs);

Player.transform.position = new Vector3(pos.x, pos.y, pos.z);}

catch (System.Exception e){

Debug.Log(e.Message);}

finally{

fs.Close();}}

else{

Application.Quit();}

Player.GetComponent<helathPlayer>().health = PlayerPrefs.GetInt("health"); // сохр. здоровья}

public void OnSave(){

Position position = new Position();

position.x = Player.transform.position.x;

position.y = Player.transform.position.y;

position.z = Player.transform.position.z;

if (!Directory.Exists(Application.dataPath + "/saves"))

Directory.CreateDirectory(Application.dataPath + "/saves");

FileStream fs = new FileStream(Application.dataPath + "/saves/save.sm", FileMode.Create);

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, position);

fs.Close();

PlayerPrefs.SetInt("health", Player.GetComponent<helathPlayer>().health);}

public void Settings(){

settings.SetActive(!settings.activeSelf);}

public void Ixnstruction() {

instruction.SetActive(!instruction.activeSelf);}

public void ExitGame() {

Application.Quit();}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MiniMap : MonoBehaviour {

public GameObject myPlayer;

public float distance = 100;

void LateUpdate(){

if (myPlayer == true){{

transform.position = myPlayer.transform.position + Vector3.up \* distance;}}

else { }}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class MonsterDamegger : MonoBehaviour {

//дамаг ПО монстру

void Start () {}

void Update () {}

private void OnTriggerEnter(Collider col){

if (col.tag == "MonsterAI"){

col.GetComponent<Monstr>().healthMonster -= 10;}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Monstr : MonoBehaviour{

public float seeDistance = 5f;

public float attackDistance = 2f;

public float speed;

private Transform target; //цель

public int healthMonster;

public GameObject bloodyScreen;

[SerializeField]

private GameObject AI;

WayPoints controller;

private AudioSource source;

void Start(){

controller = AI.GetComponent<WayPoints>();

target = GameObject.FindWithTag("Player").transform; //прислед

source = GetComponent<AudioSource>();}

void Update(){

if (healthMonster == 0){

Destroy(this.gameObject);}

if (target == true){

if (Vector3.Distance(transform.position, target.transform.position) < seeDistance){

if(Vector3.Distance(transform.position, target.transform.position) > attackDistance){

controller.enabled = false; //ВайПоинтс

//движ

transform.LookAt(target.transform);

transform.Translate(new Vector3(0, 0, speed \* Time.deltaTime));

gameObject.GetComponent<Animator>().SetTrigger("Move");

bloodyScreen.SetActive(true);}

else{}

if (speed != 0.0f || speed != 0.0f){

if (!source.isPlaying){

source.Play();}}}

else { bloodyScreen.SetActive(false);

controller.enabled = true;}}

else { }}}

using System.Collections;

using System.Collections.Generic;

using System.Linq;

using UnityEngine;

using UnityEngine.UI;

public class Nastrouki : MonoBehaviour {

public GameObject QuestForLevel;

public Slider Soundslider;

public GameObject menu;

public Dropdown dropdown;

public Toggle toggle; //галочка

Resolution[] res;

void Start () {

Screen.fullScreen = true;

toggle.isOn = true;

Resolution[] resolution = Screen.resolutions; // врнт

res = resolution.Distinct().ToArray(); // без повтр

string[] strRes = new string[res.Length];

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

strRes[i] = res[i].width.ToString() + "x" + res[i].height.ToString(); //передача без частоты}

dropdown.ClearOptions();

dropdown.AddOptions(strRes.ToList());

Screen.SetResolution(res[res.Length - 1].width, res[res.Length - 1].height, Screen.fullScreen);//макс . разр}

public void SetRes(){

Screen.SetResolution(res[dropdown.value].width, res[dropdown.value].height, Screen.fullScreen);}

public void ScreenMode(){

Screen.fullScreen = !toggle.isOn;}

void Update () {

AudioListener.volume = Soundslider.value; // настр звука}

// настройки графики

public void LowQuality(){

QualitySettings.SetQualityLevel(0, true);}

public void MediumQuality(){

QualitySettings.SetQualityLevel(2, true);}

public void UltraQuality(){

QualitySettings.SetQualityLevel(4, true);}

public void ExitGame(){

Application.LoadLevel(1);}

public void QuestForLevel2() // квест{

Destroy(QuestForLevel);

Cursor.visible = false;}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class OtobrOruj : MonoBehaviour {

public GameObject Weapon1;

public GameObject Weapon2;

public GameObject Weapon3;

public GameObject Weapon4;

public GameObject Weapon5; //колич. паатронов

void Update()

{if (Input.GetKeyDown(KeyCode.Alpha1)){

Weapon1.SetActive(true);

Weapon2.SetActive(true);

Weapon3.SetActive(false);

Weapon4.SetActive(false);

Weapon5.SetActive(false);}

if (Input.GetKeyDown(KeyCode.Alpha2)){

Weapon1.SetActive(false);

Weapon2.SetActive(false);

Weapon3.SetActive(true);

Weapon4.SetActive(true);

Weapon5.SetActive(true);}

if (Input.GetKeyDown(KeyCode.Alpha0)){

Weapon1.SetActive(false);

Weapon2.SetActive(false);

Weapon3.SetActive(false);

Weapon4.SetActive(false);

Weapon5.SetActive(false);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Pers : MonoBehaviour{

public float speed = 2f;

public float sensitivty = 10f; //чувства

CharacterController player;

public GameObject eyes;

private float moveFB, moveLR, rotX, rotY, vertVelocty;

public float jumpForce = 4f;

private bool hasJumped, isCrouched;

private AudioSource source;

void Start(){

player = GetComponent<CharacterController>();

source = GetComponent<AudioSource>();}

void Update()

{ //пжок

Movement();

if (Input.GetButtonDown("Jump")){

hasJumped = true;}

if (Input.GetButtonDown("Crouch")) // пресмыкаться{

if (isCrouched == false){

player.height = player.height / 2;

isCrouched = true;}

else{

player.height = player.height \* 2;

isCrouched = false;}}

ApplyGravity();

if (transform.position.y < 0) { Application.LoadLevel(Application.loadedLevel); } //Огран. Выход за гран.}

public void Movement(){

moveFB = Input.GetAxis("Vertical") \* speed;

moveLR = Input.GetAxis("Horizontal") \* speed;

rotX = Input.GetAxis("Mouse X") \* sensitivty; //повороты, повороты с движем

rotY = Input.GetAxis("Mouse Y") \* sensitivty;

Vector3 movement = new Vector3(moveLR, vertVelocty, moveFB);

transform.Rotate(0, rotX, 0);

eyes.transform.Rotate(-rotY, 0, 0);

movement = transform.rotation \* movement;

player.Move(movement \* Time.deltaTime);

//звуки шагов

if (moveFB != 0.0f || moveFB != 0.0f)

{ // Палка вниз

if (!source.isPlaying){

source.Play();}}

if (moveLR != 0.0f || moveLR != 0.0f){

if (!source.isPlaying){

source.Play();}}}

private void ApplyGravity(){

if (player.isGrounded == true) // заземлен{

if (hasJumped == false){

vertVelocty = Physics.gravity.y;}

else{

vertVelocty = jumpForce; //можно прыгать}}

else

{ //скоость приземл

vertVelocty += Physics.gravity.y + Time.deltaTime;

vertVelocty = Mathf.Clamp(vertVelocty, -50f, jumpForce);

hasJumped = false;}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class PodborOruj : MonoBehaviour {

public GameObject axe;

private void OnTriggerEnter(Collider col){

if (col.tag == "Player")

{axe.SetActive(true);

Destroy(gameObject);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

public class Portal : MonoBehaviour {

public int LevelCount;

// порт перса на данную перееменную

void OnTriggerEnter(Collider col){

if (col.tag == "Player"){

SceneManager.LoadScene(LevelCount);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Video;

using UnityEngine.EventSystems;

public class Predistor : MonoBehaviour{

public int sceneID;

public VideoPlayer videop;

Slider taisn;

private void Start(){

taisn = GetComponent<Slider>();}

void Update(){

taisn.value = (float)videop.frame / (float)videop.frameCount;

if ((float)videop.frame / (float)videop.frameCount >= 1) { Application.LoadLevel(1); }

if (Input.GetKeyDown(KeyCode.Escape)){

Application.LoadLevel(1);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class QuestForLVL2 : MonoBehaviour {

public GameObject Quest;

// Use this for initialization

void Start () {}

// Update is called once per frame

void Update () {

if (Quest == true) { Cursor.visible = true; }}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

public class quitIngame : MonoBehaviour{

void Start()

public void ExitGame()

{// Application.Quit();

Application.LoadLevel(1);}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class RottateObj : MonoBehaviour {

public int x;

public int y;

public int z;

void Update () {

gameObject.transform.Rotate(new Vector3(x, y, z));}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

public class SaveGame : MonoBehaviour {

public GameObject player;

[System.Serializable]

public class Position{

public float x;

public float y;

public float z;

} // сохранения по позициям в папку

public void Save () {

Position position = new Position();

position.x = player.transform.position.x;

position.y = player.transform.position.x;

position.z = player.transform.position.x;

if (!Directory.Exists(Application.dataPath + "/saves"))

Directory.CreateDirectory(Application.dataPath+"/saves");

FileStream fs = new FileStream(Application.dataPath + "/saves/save.sv", FileMode.Create);

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, position);

fs.Close();}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System.IO;

using System.Runtime.Serialization.Formatters.Binary;

public class SaveLoad2level : MonoBehaviour {

public GameObject Player;

[System.Serializable]

public class Position{

public float x;

public float y;

public float z;}

public void StartGame(){

Application.LoadLevel(3);}

public void LoidGame(){

// загрузить файл по позициям

if (File.Exists(Application.dataPath + "/saves/save.sm")){

FileStream fs = new FileStream(Application.dataPath + "/saves/save.sm", FileMode.Open);

BinaryFormatter formatter = new BinaryFormatter();

try{

Position pos = (Position)formatter.Deserialize(fs);

Player.transform.position = new Vector3(pos.x, pos.y, pos.z);}

catch (System.Exception e){

Debug.Log(e.Message);}

finally{

fs.Close();}}

else{

Application.Quit();}

Player.GetComponent<helathPlayer>().health = PlayerPrefs.GetInt("health");}

public void OnSave(){

Position position = new Position();

position.x = Player.transform.position.x;

position.y = Player.transform.position.y;

position.z = Player.transform.position.z;

if (!Directory.Exists(Application.dataPath + "/saves"))

Directory.CreateDirectory(Application.dataPath + "/saves");

FileStream fs = new FileStream(Application.dataPath + "/saves/save.sm", FileMode.Create);

BinaryFormatter formatter = new BinaryFormatter();

formatter.Serialize(fs, position);

fs.Close();

PlayerPrefs.SetInt("health", Player.GetComponent<helathPlayer>().health);}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;

public class SceneLoding : MonoBehaviour {

[Header("Загружаемая сцена")]

public int sceneID;

[Header("Остальные объекты")]

public Image loadingImg;

public Text progressText;

void Start() {

StartCoroutine(AsyncLoad());}

IEnumerator AsyncLoad(){

AsyncOperation opertion = SceneManager.LoadSceneAsync(sceneID);

while (!opertion.isDone)

{ // вовремя загрузки проценты доходят до 100 и кружок

float progress = opertion.progress / 0.9f;

loadingImg.fillAmount = progress;

progressText.text = string.Format("{0:0}%", progress \* 100);

yield return null;}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SkeletAttack : MonoBehaviour {

public GameObject AI;

public GameObject triggerweapon;

void Start () {

triggerweapon.SetActive(false);}

private void OnTriggerEnter(Collider col){

if (col.tag == "Player"){

AI.GetComponent<Animator>().SetTrigger("Attack");

triggerweapon.SetActive(true);}}

//видит, атакует. бежит не атакует

private void OnTriggerExit(Collider col){

if (col.tag == "Player"){

AI.GetComponent<Animator>().SetTrigger("Move");

triggerweapon.SetActive(false);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SmenaOrij : MonoBehaviour{

public GameObject menu;

public GameObject Weapon1;

public GameObject Weapon2;

public GameObject qrabot;

void Start(){

qrabot.SetActive(false);}

void Update(){

if (Input.GetKeyDown(KeyCode.Alpha1)){

qrabot.SetActive(false);

Weapon1.SetActive(true);

Weapon2.SetActive(false)}

if (Input.GetKeyDown(KeyCode.Alpha2)){

qrabot.SetActive(true);

Weapon2.SetActive(true);

Weapon1.SetActive(false);}

if (Input.GetKeyDown(KeyCode.Alpha0)){

qrabot.SetActive(false);

Weapon2.SetActive(false);

Weapon1.SetActive(false);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SnenaSL : MonoBehaviour {

float hours = 5;

float minutses = 45;

public float speed;

public GameObject PivotLS;

private void FixedUpdate(){

PivotLS.transform.Rotate(Vector3.right \* 5 \* Time.deltaTime);

minutses = minutses + speed + Time.deltaTime;

if (minutses >= 60){

hours = hours + 1;

if (hours >= 24){

hours = 0;}

minutses = 0;}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class Strelyba : MonoBehaviour{

public Text text;

public int AmmoCount; //Патронов в обоймах

public int ColP; //Кол-во патроеов

public int ColPVO; //Кол-во патронов в 1ой обойме

public AudioClip Fire; // Звук выстрела

public float ShootSpeed; // Скорострельность

public float ReloadSpeed; // Скорость Перезарядки

public AudioClip Reload; // Звук перезарядки

public float ReloadTimer = 0.0f; // Время перезарядки

public float ShootTimer = 0.0f; // Время выстрелa

public Transform Patron; // Патрон

public GameObject avtomat;

void Update(){

// стрельба

if (Input.GetMouseButton(0) & ColP > 0 & ReloadTimer <= 0 & ShootTimer <= 0){

avtomat.GetComponent<Animator>().SetTrigger("Fire");

Transform BulletInstance = (Transform)Instantiate(Patron, GameObject.Find("Spawn").transform.position, Quaternion.identity);

BulletInstance.GetComponent<Rigidbody>().AddForce(transform.forward \* 5000);

ColP = ColP - 1;

GetComponent<AudioSource>().PlayOneShot(Fire);

ShootTimer = ShootSpeed;}

if (ShootTimer > 0){

ShootTimer -= Time.deltaTime;}

//перезарялка

if (Input.GetKeyDown(KeyCode.Q)){

avtomat.GetComponent<Animator>().SetTrigger("Perez");

ReloadTimer = ReloadSpeed;

ColP = ColPVO;

GetComponent<AudioSource>().PlayOneShot(Reload);{

if (ShootTimer > 0){

ShootTimer -= Time.deltaTime;}}}

if (ReloadTimer > 0){

ReloadTimer -= Time.deltaTime;}

text.text = ColP+"/10";}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class SwordAttack : MonoBehaviour{

public GameObject SliderEner;

public GameObject triggerDamage;

void Update(){

if (Input.GetButtonDown("Fire1")) //ЛКМ{

gameObject.GetComponent<Animator>().SetTrigger("Attack");

triggerDamage.SetActive(true);

SliderEner.GetComponent<EnerjiPlayer>().stamina -= 5; // вычитании энергии}

if (Input.GetButtonDown("Fire2")){

SliderEner.GetComponent<EnerjiPlayer>().stamina -= 10;

gameObject.GetComponent<Animator>().SetTrigger("Attack1");

triggerDamage.SetActive(true);}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class TabletUp : MonoBehaviour {

public int upTablet;

private float time;

private bool up;

void Update () {

if (up == true){

time += 0.1f;

if (time >= 1){

Destroy(gameObject); //табл исчезает}}}

private void OnTriggerEnter(Collider col){

if (col.tag == "Player"){

col.GetComponent<helathPlayer>().health += upTablet;

up = true;

gameObject.GetComponent<MeshRenderer>().enabled = false;}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class TimeAttack : MonoBehaviour {

public int timeStart;

public int timeEnd;

void Update () {

timeStart += 1;

if (timeStart >= timeEnd){

timeStart = 0;

gameObject.SetActive(false); //удар не проходит}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class TriggerOtnimJ : MonoBehaviour{

public int valueDown; //сколько отнимает

void OnTriggerEnter(Collider col)

{if (col.tag == "Player"){

col.GetComponent<helathPlayer>().health -= valueDown;}}}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class WayPoints : MonoBehaviour {

public GameObject[] waypoints;

public int num = 0;

public float minDist;

public float speed;

public bool rand = false;

public bool go = true;

private Transform target;

void Update(){

float dist = Vector3.Distance(gameObject.transform.position, waypoints[num].transform.position);

if (go){

if (dist > minDist){

move();}

else{

if (!rand){

if (num + 1 == waypoints.Length){

num = 0;}

else { num++; }}

else { num = Random.Range(0, waypoints.Length); }}}}

public void move(){

gameObject.GetComponent<Animator>().SetTrigger("Move");

gameObject.transform.LookAt(waypoints[num].transform.position);

gameObject.transform.position += gameObject.transform.forward \* speed \* Time.deltaTime;}}